

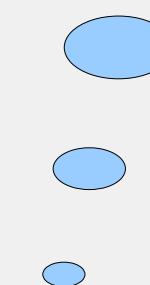
## Teil II

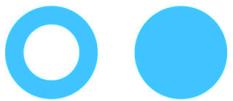
Von Daten und ihren Modellen

Robert Hartmann (SoSe 2024)

basierend auf Folien von  
Prof. Dr. Harm Knolle

Fachbereich Informatik  
Hochschule Bonn-Rhein-Sieg





## - Kapitel 6 - Datenanfrage und Datenänderung -

### Inhalt

- 0 - Vorbemerkungen

**Teil I** - Von EDV-Anwendungen und Ihren Anforderungen

- 1 - Einführung

**Teil II** - Von Daten und ihren Modellen

- 2 - Prozess des Datenbankentwurfs

- 3 - Semantische Datenmodelle

- 4 - Logische Datenmodelle

- 5 - Datenbankmodelle

- 6 - Datenanfrage und Datenänderung**

**Teil III** - Von Datenbanken und ihren Systemen

- 7 - Datenbanksysteme

- 8 - Speicherstrukturen

- 9 - Ausblick

### Inhalt

- ◆ Einführung
- ◆ Theoretische Grundlagen relationaler Sprachen
- ◆ Relationenalgebra
- ◆ Datenbanksprache SQL



## - Datenanfrage und Datenänderung -

### Ziel

- ◆ Formulierung von Datenanfragen auf der Grundlage relationaler Technologien
- ◆ Formulierung von Datenänderungen auf der Grundlage relationaler Technologien

### Hilfsmittel

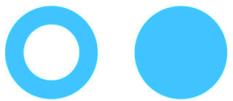
- ◆ Theoretische Grundlagen der Relationenalgebra und des Relationenkalküls
- ◆ relationale Datenbanksprache SQL
- ◆ Datenmanipulationsanweisungen (DML) einer Datenbanksprache

### Inhalt

- ◆ Einführung
- ◆ Theoretische Grundlagen relationaler Sprachen
- ◆ Relationenalgebra
- ◆ Datenbanksprache SQL

### Literatur

- ◆ KeEi15
  - Kapitel 3 „Das relationale Modell“
    - ab 3.4
  - Kapitel 4 „Relationale Anfragesprachen“
    - ab 4.6 bis einschl. 4.18
- ◆ Ku15
  - Kapitel 3: „Relationales Datenmodell“
    - ab 3.5
  - Kapitel 4: „Die Datenbanksprache SQL“
    - 4 bis einschl. 4.6.4
- ◆ SSH18
  - Kapitel 7: „Das relationale Datenmodell“
    - ab 7.3
  - Kapitel 8: „SQL - der relationale Datenbankstandard“
    - ab 8.2



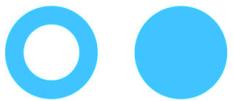
## - Einführung -

### Inhalt

- ◆ **Einführung**
- ◆ Theoretische Grundlagen relationaler Sprachen
- ◆ Relationenalgebra
- ◆ Datenbanksprache SQL

### Überblick

- ◆ Kriterien von Datenanfragesprachen
- ◆ Hierarchisches Datenmodell (in Kapitel 4)
- ◆ Netzwerk Datenmodell (in Kapitel 4)
- ◆ Heutige Anfragesprachen



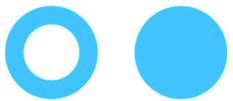
## - Kriterien von Datenanfragesprachen (I) -

### Grundlagen

- ◆ Abgeschlossenheit
  - Ergebnis ist wieder Relation und somit mögliche Basisrelation als Eingabe für weitere Anfragen
- ◆ Adäquatheit
  - alle Konzepte des zugrundeliegenden Datenmodells werden unterstützt
- ◆ Orthogonalität
  - Sprachkonstrukte sind in ähnlichen Situationen auch ähnlich anwendbar
- ◆ Mengenorientiertheit
  - jede Operation soll auf Mengen von Daten gleichzeitig arbeiten können, nicht nur navigierend auf einzelnen Elementen

### Schnittstelle zum System

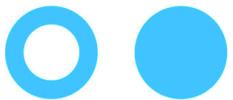
- ◆ Optimierbarkeit
  - Sprache besteht aus wenigen, formal definierten Operationen, für die es Optimierungsregeln gibt
- ◆ Effizienz
  - Komplexität der Operation steigt nicht exponentiell zur Anzahl der Tupel an
- ◆ Sicherheit (der Laufzeit)
  - keine (syntaktisch korrekte) Anfrage darf ein unendliches Ergebnis liefern oder unendlich lange ausgewertet werden
- ◆ Datenbanksprachen zur Datendefinition



## - Kriterien von Datenanfragesprachen (II) -

### Schnittstelle zum Anwender

- 
- ◆ Vollständigkeit
    - Anfragesprache muss mindestens die Anfragen einer Standardsprache ausdrücken können
    - im Relationenmodell
      - Relationenalgebra
      - Relationenkalkül
  - ◆ Eingeschränktheit
    - Anfragesprache sollte keine komplette Programmiersprache sein
  - ◆ Deskriptivität
    - Anwender soll formulieren, was er haben will, nicht welche Speicherstrukturen er ansprechen soll, um das zu bekommen, was er haben will
  - ◆ Ad-Hoc-Formulierung
    - Anwender soll Anfrage formulieren können, ohne vollständiges Programm schreiben zu müssen



## - Heutige Anfragesprachen -

### Relationale Systeme

- ♦ Standard fast aller Systemanbieter

### Sprache

- ♦ SEQUEL (System R)
- ♦ QUEL (INGRES)
- ♦ SQL (INFORMIX, ORACLE, MS SQL-Server, PostgreSQL, MySQL, ...)
- ♦ QBE (MS-ACCESS)

### Datendefinition

- ♦ basiert auf zweidimensionalen Tabellen
- ♦ Beziehungen über Fremdschlüssel und spezielle "Beziehungs-Tabellen"
- ♦ basiert auf Relationenalgebra und -kalkül
- ♦ mengenorientiert

### Objektorientierte Systeme

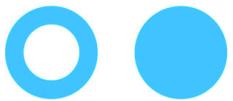
- ♦ Objektrelationale Systeme (z.B Oracle seit Version 8 (1997), ...)
- ♦ Objektorientierte Systeme (O<sup>2</sup>, POET, ...)

### Sprache

- ♦ ODMG-93 : Erweiterung von C++ (Objekt Data Management Group)
- ♦ ab SQL-99 : objekt-relationale SQL-Erweiterung
- ♦ Transact/, PL/SQL: Definition von anwendungsspezifischen Operationen auf Objekten (Programming Language/SQL)

### Datendefinition

- ♦ direkte Definition semantisch komplex strukturierter (relationaler) Objekte
- ♦ Erweiterungen von C++ oder Java
- ♦ Erweiterungen relationaler Sprachen



## - Theoretische Grundlagen relationaler Sprachen -

### Inhalt

- ◆ Einführung
- ◆ Theoretische Grundlagen relationaler Sprachen
- ◆ Relationenalgebra
- ◆ Datenbanksprache SQL

### Überblick

- ◆ Allgemeines
- ◆ Relationenkalkül vs. Relationenalgebra



## - Allgemeines -

### Datenbanksprachen zur Bearbeitung von Mengen in Form von Relationen

- ◆ Definition von Relationen
- ◆ Herleitung neuer Relationen aus vorhandenen
- ◆ Veränderung von vorhandenen Relationen
- ◆ Löschen von Relationen

### Aufbau von Datenbanksprachen wie

- ◆ SQL, QBE, QUEL, SEQUEL
- ◆ stützen sich auf mathematisch / logische Grundlagen wie
  - Relationenalgebra
  - Relationenkalkül

Student

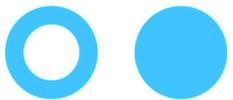
Matr-Nr	Name
13478	Dilloo
12481	Wilke
12496	Schmidt
13477	Grau
12462	Schmidt

### Anfrage über Relationenalgebra

- ◆ Spezifikation von gewünschten Relationen durch Angabe einer Operationsfolge
- ◆ z.B. "Suche die Zeilen der Relation Studenten, die als Matr.-Nr den Wert 12496 haben: Dargestellt werden soll nur die Spalte Name."

### Anfrage über Relationenkalkül

- ◆ Interessant für Abfrage-Optimierung sowie theoretische Betrachtung relationaler Technologien
- ◆ Spezifikation von gewünschten Relationen in deskriptiver Form ohne Angabe von Operatoren
- ◆ z.B. "Gesucht ist eine Menge von Namen von Studenten für die gilt, dass die Matr.-Nr dieser Studenten 12496 lauten muss."



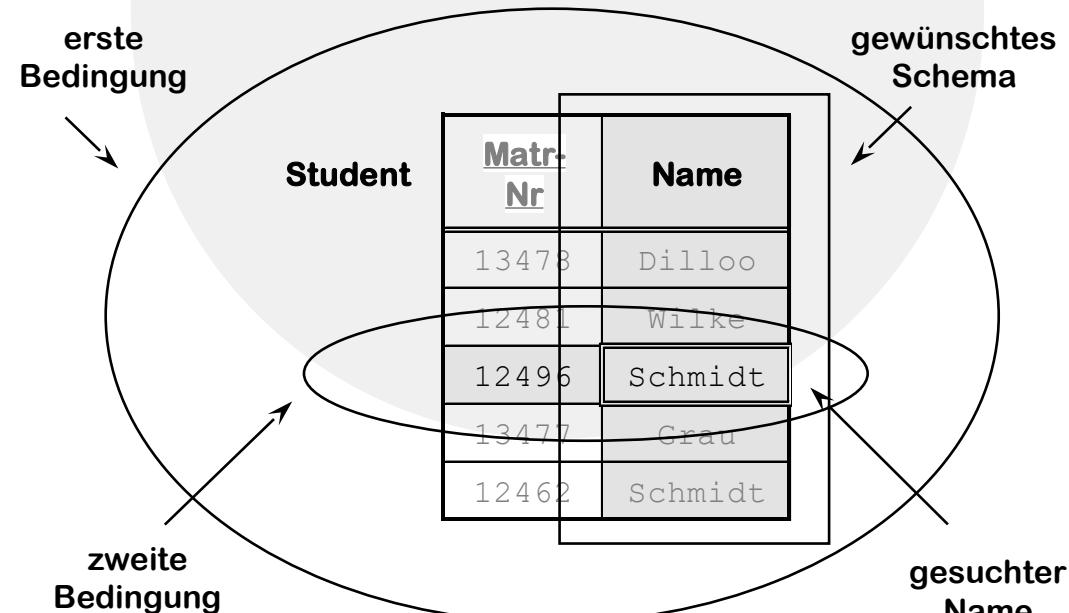
## - Relationenkalkül vs. Relationenalgebra (I) -

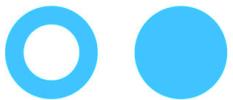
### Relationenkalkül

- ◆ Logik / Mathematik
  - Kalkül ist eine formale logische Sprache zur Formulierung von Aussagen
  - Prädikatenkalkül: Aussage als Anfrage mit Antwort "wahr" oder "falsch"
  - Aussagen zur Konstruktion von Mengen
  - z.B.  $\{ x^2 \mid x \in \mathbb{N} \wedge x^3 > 0 \wedge x^3 < 1.000 \}$
- ◆ Datenbank
  - Relationen als Mengen von Tupel
  - Datenanfrage als Konstruktion von Tupelmengen über Bedingungen
- ◆ Relationenkalkül
  - Angabe eines Relationenschemas mit Bedingungen (Prädikate), die den Werten dieses Schemas genügen
  - $\{ \text{Schema} \mid \text{Bedingung} \}$

- ◆ Nicht-prozeduraler Ansatz
  - lediglich Beschreibung, was man möchte
- ◆ Beispiel
  - Name des Studenten der Relation Student mit der Matr.-Nr 12496

$\{t.\text{name} \mid \text{Student}(t) \wedge t.\text{Matr.-Nr} = 12496\}$



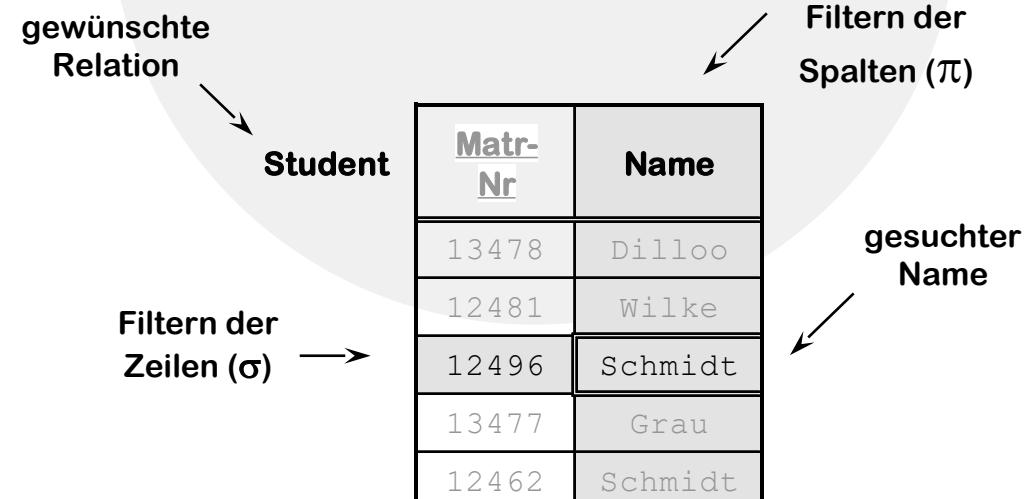


## - Relationenkalkül vs. Relationenalgebra (II) -

### Relationale Algebra

- ◆ Mathematik
  - Algebra definiert durch Wertebereich und auf diesem definierte Operatoren
- ◆ Datenbank
  - Inhalte der Datenbank sind Werte und Operatoren definieren Funktionen zum Berechnen von Anfrageergebnissen
- ◆ Relationenalgebra
  - vollständig und einsortig
  - Tabellen als Werte
  - Operatoren auf Tabellen wie
    - Spalten ausblenden
    - Zeilen heraussuchen
    - Tabellen verknüpfen
    - Tabellen vereinigen
    - Tabellen voneinander abziehen
    - Spalten umbenennen

- ◆ Prozeduraler Ansatz
  - Beschreibung, was man möchte
  - Beschreibung, wie man es bekommt
  - z.B. durch heraussuchen von Zeilen und Ausblenden von Spalten
- ◆ Beispiel
  - Name des Studenten der Relation Student mit der Matr.-Nr 12496





## - Relationenalgebra -

### Inhalt

- ◆ Einführung
- ◆ Theoretische Grundlagen relationaler Sprachen
- ◆ Relationenalgebra
  - Grundoperationen der Relationenalgebra
  - Zusammengesetzte Operationen
  - Datenmanipulation
- ◆ Datenbanksprache SQL

### Überblick

- ◆ Datenanfrage
- ◆ Umbenennung - Attribute Umbenennen
- ◆ Selektion - Zeilen heraussuchen
- ◆ Projektion - Spalten ausblenden
- ◆ Union - Vereinigung von Relationen
- ◆ Differenz - Differenz zwischen Relationen
- ◆ Produkt - Kartesisches Produkt zwischen Relationen



## - Datenanfrage -

### Grundoperationen

- ◆ Zeilen heraussuchen (Selektion)
- ◆ Spalten ausblenden (Projektion)
- ◆ Vereinigung von Relationen (Union)
- ◆ Differenz von Relationen
- ◆ Produkt von Relationen

### Hilfsoperationen

- ◆ Umbenennen von Spalten


Selektion


Projektion


Vereinigung

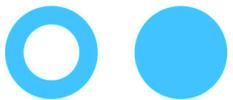

Differenz

a	x
b	
c	
	x
	y

×

x
y

Produkt



## - Umbenennung - Attribute Umbenennen -

$$R2 = \beta [ \text{Attribut neu} \leftarrow \text{Attribut alt} ] R1$$

- der Umbenennungsoperator  $\beta$  bewirkt, dass ein Attribut "alt" der Relation R in der Ergebnisrelation R2 in Attribut "neu" umbenannt wird

### Anwendung

- Bildung synonymer Attribute
- Vorbereitung zur Anwendung von Operatoren auf mehreren Relationen mit namensgleichen Attribut

### Beispiel

- Haeuser
- $= \beta [ \text{Haus} \leftarrow \text{Gebaeude} ] \text{ Gebaeude}$

Das hier verwendete Zeichen heißt „Beta“.

Gebaeude

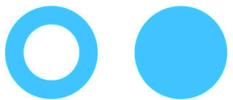
$$\beta \downarrow$$

Ge-baeude	Strasse
B	Am Schwimmbad
C	Blechhammer
D	Blechhammer

Haeuser

Haus	Strasse
B	Am Schwimmbad
C	Blechhammer
D	Blechhammer

Statt  $\beta$  wird anderen Orts auch  $\rho$  (sprich Rho) verwendet



## - Selektion - Zeilen heraussuchen -

$$R2 = \sigma [ \text{Bedingung} ] R1$$

- der Selektionsoperator  $\sigma$  selektiert aus der Relation R1 alle diejenigen Tupel, die einer bestimmten Bedingung  $\theta$  genügen und fasst diese zur Ergebnisrelation R2 zusammen
- Mehrere Bedingungen werden mittels logischen Operatoren zu einem Bedingungsausdruck verbunden

### Bedingung

- $A1 \theta A2$  oder  $A1 \theta$  Konstante
  - A1 und A2 sind Attribute
  - $\theta$  (Theta) steht für einen arithmetischen Vergleichsoperator ( $=, >, <, \text{usw.}$ )

### Anwendung

- Beschränkung der Tabelle auf wesentliche Informationsträger
- Selektieren von Zeilen

### Beispiel

- $\text{Einige\_Orte}$   
 $= \sigma [\text{Gebaeude} = 'B'] \text{ Veranstaltungsort}$

Das hier verwendete Zeichen heißt „Sigma“.

Veranstaltungs-  
ort

$\sigma$

Einige\_Orte

LV-Nr	Ge-baeude	Raum
1238	B	431
1153	D	123
1543	B	23
1352	C	231
1421	C	250

LV-Nr	Ge-baeude	Raum
1238	B	431
1543	B	23



## - Projektion - Spalten ausblenden -

$$R2 = \pi [ \text{Attributliste} ] R1$$

- der Projektionsoperator  $\pi$  selektiert bestimmte Attribute einer Relation R1 und fasst diese zur Ergebnisrelation R2 zusammen

### Anwendung

- Ausblendung uninteressanter oder datengeschützter Attributwerte

### Beispiel

- Veranstaltungsgebäude  
 $= \pi [LV\text{-}Nr, Gebaeude] Veranstaltungsort$

Veranstaltungs-  
ort

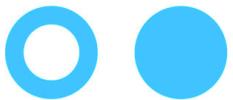
$\pi$

Veranstaltungs-  
gebäude

LV- Nr	Ge- baeude	Raum
1238	B	431
1153	D	123
1543	B	23
1352	C	231
1421	C	250

LV- Nr	Ge- baeude
1238	B
1153	D
1543	B
1352	C
1421	C

Das hier verwendete  
Zeichen heißt „Pi“.



## - Union - Vereinigung von Relationen -

$$R3 = R1 + R2$$

- der Union-Operator  $+$ , alternativ  $\cup$ , vereinigt die Zeilen zweier Relationen  $R1$  und  $R2$  zu einer Relation  $R3$
- Voraussetzung ist, dass  $R1$  und  $R2$  den gleichen Grad (Anzahl der Attribute) die gleichen Namen, gleiche Wertebereich und die gleiche Reihenfolge der jeweiligen Attribute besitzen
- bei ungleichen Namen der Attribute ist vorher eine Umbenennung ( $\beta$ ) erforderlich

### Anwendung

- Zusammenführung strukturgleicher Informationen

### Beispiel

- Veranstaltungsort  
 $= \text{Einige\_Orte} + \text{Andere\_Orte}$

Einige\_Orte

+

Andere\_Orte

Veranstaltungs-  
ort

LV-Nr	Ge-baeude	Raum
1238	B	431
1543	B	23

LV-Nr	Ge-baeude	Raum
1153	D	123
1352	C	231
1421	C	250

LV-Nr	Ge-baeude	Raum
1238	B	431
1543	B	23
1153	D	123
1352	C	231
1421	C	250



## - Differenz - Differenz zwischen Relationen -

$R3 = R1 - R2$

- die Ergebnisrelation  $R3$  einer Differenz-Operation - zwischen den Relationen  $R1$  und  $R2$  enthält lediglich solche Zeilen aus  $R1$ , die nicht in  $R2$  enthalten sind
- Voraussetzung ist, dass  $R1$  und  $R2$  den gleichen Grad (Anzahl der Attribute) die gleiche Wertebereich und die gleiche Reihenfolge der jeweiligen Attribute besitzen
- bei ungleichen Namen der Attribute ist vorher eine Umbenennung ( $\beta$ ) erforderlich

Anwendung

- Ausschluss von bestimmten Informationsträgern aus einer Menge

Beispiel

- Andere\_Orte  
= Veranstaltungsort - Einige\_Orte

Veranstaltungs-  
ort

LV- Nr	Ge- baeude	Raum
1238	B	431
1543	B	23
1153	D	123
1352	C	231
1421	C	250

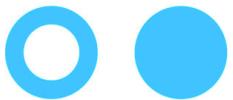
LV- Nr	Ge- baeude	Raum
1238	B	431
1543	B	23

Einige\_Orte



Andere\_Orte

LV- Nr	Ge- baeude	Raum
1153	D	123
1352	C	231



## - Produkt - Kartesisches Produkt zwischen Relationen -

$$R3 = R1 \times R2$$

- die Ergebnisrelation  $R3$  einer Produkt-Operation  $\times$  zwischen den Relationen  $R1$  und  $R2$  entspricht der Verknüpfung jeder Zeile von  $R1$  mit jeder Zeile von  $R2$
- die Zeilenanzahl von  $R3$  ist somit gleich dem Produkt der jeweiligen Zeilenanzahl von  $R1$  und  $R2$
- das Relationenschema von  $R3$  setzt sich aus der Verknüpfung der jeweiligen Attributlisten von  $R1$  und  $R2$  zusammen
- Umbenennung  $\beta$  erforderlich bei Namensgleichheiten

### Anwendung

- generelle Paarbildung, jeder mit jedem

### Beispiel

- Spielpaarung**  
 $= \text{Einige\_Studenten} \times \text{Gegner\_Studenten}$

Spielpaarung

Einige  
Studenten

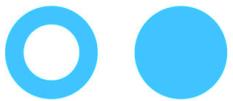
$\beta, \times$

Gegner  
Studenten

Matr-Nr	Name
13478	Dilloo
12481	Wilke
12496	Schmidt

Matr-Nr	Name
13477	Grau
12462	Schmidt

Matr-Nr	Name	Gegner-Nr	Gegner-Name
13478	Dilloo	13477	Grau
13478	Dilloo	12462	Schmidt
12481	Wilke	13477	Grau
12481	Wilke	12462	Schmidt
12496	Schmidt	13477	Grau
12496	Schmidt	12462	Schmidt



## - Relationenalgebra -

### Inhalt

- ◆ Einführung
- ◆ Theoretische Grundlagen relationaler Sprachen
- ◆ Relationenalgebra
  - Grundoperationen der Relationenalgebra
  - Zusammengesetzte Operationen
  - Datenmanipulation
- ◆ Datenbanksprache SQL

### Überblick

- ◆ Datenanfrage
- ◆ Theta Join / Equi Join - Verbund von Relationen
- ◆ Herleitung des Theta Join
- ◆ Natural Join - Natürlicher Verbund von Relationen
- ◆ Herleitung des Natural Join
- ◆ Division - Teilen von Relationen



## - Datenanfrage -

### Schnitt von Relationen (intersection)

- ◆  $R_1 \cap R_2$


Schnittmenge

### Verbund von Relationen (Join)

- ◆ Equi-Join:  $R_1 \bowtie_{(a=b)} R_2$
- ◆ Theta-Join:  $R_1 \bowtie_{(a \theta b)} R_2$
- ◆ natürlicher Join:  $R_1 \otimes R_2$

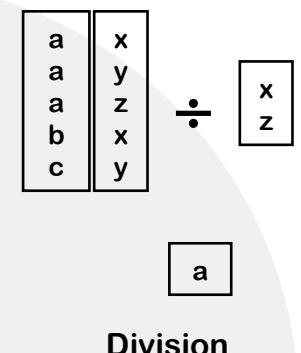
a	1	⊗	1	x
b	2		2	y
c	3		3	z

a	1	x
b	2	y
c	3	z

natürlicher  
Verbund

### Division von Relationen

- ◆  $R_1 \div R_2$





## - Theta Join / Equi Join - Verbund von Relationen -

$$R3 = R1 \bowtie(\theta) R2 = R1 \bowtie(\text{Bedingung}) R2$$

- im Gegensatz zum kartesischen Produkt × werden beim allgemeinen Join nur solche Zeilen der Relation R1 mit solchen Zeilen der Relation R2 verknüpft, die der Bedingung eines Vergleichoperators  $\theta$  (Theta) genügen
- Equi Join :  $R_1 \bowtie_{(a=b)} R_2$  wenn Bedingung Gleichheit „=“ ist

### Bedingung

- $A_1 \theta A_2$  oder  $A_1 \theta <\text{Konstante}>$
- $A_1$  und  $A_2$  sind Attribute aus R1 oder R2

### Anwendung

- bedingte Gegenüberstellungen von Daten

### Beispiel

- z.B. gegen ältere Gegner
- Vorteilpaarung  
= Einige\_Studenten  $\bowtie [ \text{Matr-Nr} > \text{Gegner-Nr} ]$  Gegner\_Studenten

Vorteil-paarung

Matr-Nr	Name	Gegner-Nr	Gegner-Name
13478	Dilloo	13477	Grau
13478	Dilloo	12462	Schmidt
12481	Wilke	12462	Schmidt
12496	Schmidt	12462	Schmidt

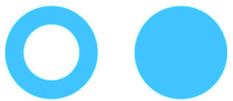
$\theta (>)$

Einige Studenten

Matr-Nr	Name
13478	Dilloo
12481	Wilke
12496	Schmidt

Gegner Studenten

Matr-Nr	Name
13477	Grau
12462	Schmidt



## - Herleitung des Theta Join / Equi Join -

Einige  
Studenten

Matr-Nr	Name
13478	Dilloo
12481	Wilke
12496	Schmidt

Gegner  
Studenten

Matr-Nr	Name
13477	Grau
12462	Schmidt

Spieldpaarung

Schritte 1 und 2:  $\beta, \times$



Matr-Nr	Name	Gegner-Nr	Gegner-Name
13478	Dilloo	13477	Grau
13478	Dilloo	12462	Schmidt
12481	Wilke	13477	Grau
12481	Wilke	12462	Schmidt
12496	Schmidt	13477	Grau
12496	Schmidt	12462	Schmidt

### Schritte 1, 2: Umbenennung, Produkt

- ♦ Spieldpaarung  
= Einige\_Studenten
- ×  $\beta[ \text{Gegner-Nr} \leftarrow \text{Matr-Nr}, \text{Gegner-Name} \leftarrow \text{Name} ] \text{Gegner_Studenten}$

### Schritt 3: Selektion

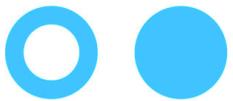
- ♦ Vorteilpaarung  
=  $\sigma [ \text{Matr.-Nr} > \text{Gegner-Nr} ] \text{ Spieldpaarung}$

Vorteil-  
paarung

Schritt 3:

$\sigma$

Matr-Nr	Name	Gegner-Nr	Gegner-Name
13478	Dilloo	13477	Grau
13478	Dilloo	12462	Schmidt
12481	Wilke	12462	Schmidt
12496	Schmidt	12462	Schmidt



## - Natural Join - Natürlicher Verbund von Relationen -

$$R3 = R1 \otimes R2$$

- beim natürlichen Verbund  $\otimes$  werden zwei Relationen R1 und R2 mit jeweils synonymen Attributen A1 und A2 vorausgesetzt
- im Gegensatz zum kartesischen Produkt  $\times$  enthält die Ergebnisrelation R3 beim natürlichen Join nur solche Zeilenkombinationen von R1 und R2, bei denen die Werte von A1 und A2 gleich sind
- außerdem wird eine der beiden synonymen Spalten von A1 oder A2 ausgeblendet

Veranstaltungsort	LV-Nr	Gebaeude	Raum
	1238	B	431
	1543	B	23



Gebaeude	Strasse
B	Am Schwimmbad
C	Blechhammer
D	Blechhammer

### Anwendung

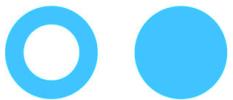
- Paarbildung und Verschmelzung von Informationen auf der Basis der Gleichheit bestimmter Merkmale (Denormalisierung)

Veranstaltungs-  
Adressen

LV-Nr	Raum	Gebaeude	Strasse
1238	431	B	Am Schwimmbad
1543	23	B	Am Schwimmbad

**Beispiel:** Veranstaltungsadressen = Veranstaltungsort  $\otimes$  Gebaeude

Veranstaltungsadressen = Veranstaltungsort  $\otimes_{(\text{Gebaeude})}$  Gebaeude



## - Herleitung des Natural Join -

Einige\_Veranstaltungsorte

LV-Nr	Ge-baeude	Raum
1238	B	431
1543	B	23

Temp\_1

LV-Nr	Haus	Raum	Ge-baeude	Strasse
1238	<b>B</b>	431	<b>B</b>	Am Schwimmbad
1238	B	431	C	Blechhammer
1238	B	431	D	Blechhammer
1543	<b>B</b>	23	<b>B</b>	Am Schwimmbad
1543	B	23	C	Blechhammer
1543	B	23	D	Blechhammer

Gebaeude

Ge-baeude	Strasse
B	Am Schwimmbad
C	Blechhammer
D	Blechhammer

### Schritte 1, 2: Umbenennung, Produkt

- Temp\_1 = Einige\_Veranstaltungsorte  $\times$   $\beta$  [ Haus  $\leftarrow$  Gebaeude ] Gebaeude

Veranstaltungs-adressen

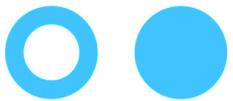
LV-Nr	Raum	Ge-baeude	Strasse
1238	431	B	Am Schwimmbad
1543	23	B	Am Schwimmbad

### Schritt 3: Selektion

- Temp\_2 =  $\sigma$  [ Gebaeude = Haus ] Temp\_1

### Schritt 4: Projektion

- Veranstaltungsadressen =  $\pi$  [ LV-NR, Raum, Ge-baeude, Strasse ] Temp\_2



## - Division - Teilen von Relationen -

$$R3 = R1 \div R2$$

- die Division  $\div$  setzt zwei Relationen R1 und R2 voraus, wobei das Attributenschema von R2 eine echte Teilmenge von R1 ist.
- die Ergebnisrelation R3 erhält nur solche Tupel aus R1 zugewiesen, die alle Eigenschaften der Tupel von R2 erfüllen
- das Schema von R3 enthält nur solche Attribute, die nicht in R2 enthalten sind

### Anwendung

- Ermittlung von Informationsträgern, die alle Eigenschaften einer Menge von Bedingungen erfüllen

### Beispiel:

- wer hört alle ausgewählten Lehrveranstaltungen?
- Ausgewählte\_Studenten  
 $= Belegungsplan \div Ausgewählte\_LV$

Belegungs-  
plan

Matr- Nr	LV- Nr
13478	1238
13478	1153
12481	1352
12496	1543
13477	1238
13477	1352
12462	1421

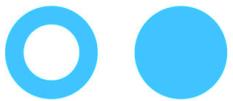
$\div$

Ausgewählte  
\_LV

LV- Nr
1238
1352

Ausgewählte  
Studenten

Matr- Nr
13477



## - Datenmanipulation -

### Änderung von Werten einer Relation

- ◆ Einfügen von Tupel (Vereinigung)
  - $R1 = R1 + R2$
  - $R1 = R1 + \{ \text{Liste mit Werten} \}$
- ◆ Löschen von Tupel (Differenz)
  - $R1 = R1 - R2$
  - $R1 = R1 - \{ \text{Liste mit Werten} \}$
- ◆ Ersetzen von Tupel (Zusammengesetzt)
  - $R1 = R1 - R2 + R3$
  - $R1 = R1 - \{ \text{Liste mit alten Werten} \} + \{ \text{Liste mit neuen Werten} \}$

### Möglichen Probleme

- ◆ Veränderung redundanter Informationen
  - müssen auch noch synonyme Objekte verändert werden?

- ◆ inkonsistente Zwischenzustände
  - andere dürfen die Zwischenergebnisse zusammengesetzter Veränderungen nicht sehen
- ◆ Einhaltung der referentiellen Integrität
  - werden die Beziehungen zwischen den Objekten nach einer Veränderung noch eingehalten?

### Beispiel

Veranstaltungs-  
ort

LV-Nr	Ge-baeude	Raum
1238	B	431
1543	B	23

- ◆  $\text{Veranstaltungsort} = \text{Veranstaltungsort} - \{ (1238, 'B', 431) \} + \{ (1238, 'B', 221) \}$
- ◆  $\text{Veranstaltungsort} = \text{Veranstaltungsort} - \{ (1543, 'B', 23) \}$