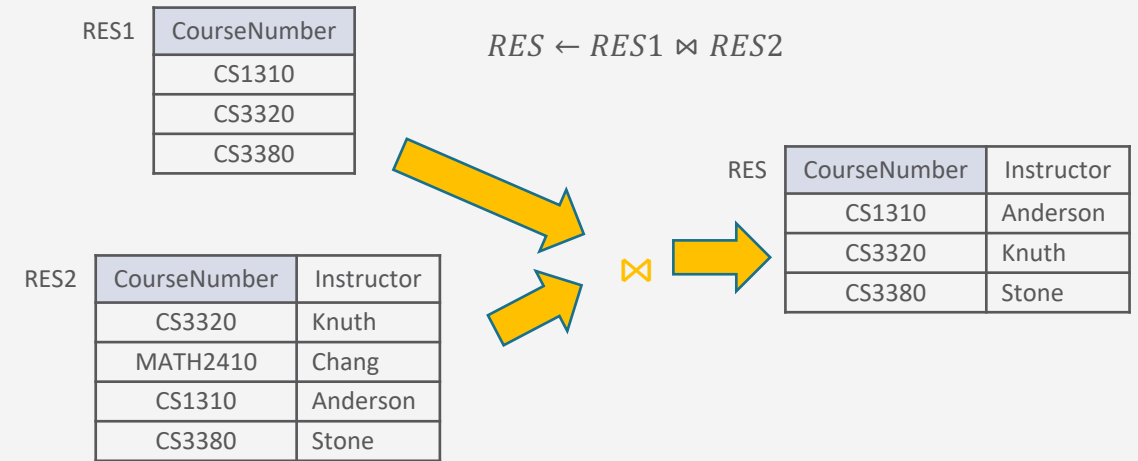


Das relationale Modell: Relationale Algebra

Datenbanken



Inhalte: Datenbanken (DBs)

1. Einführung

- Anwendungen
- Datenbankmanagementsysteme

2. Datenbank-Modellierung

- Entity-Relationship-Modell (ER-Modell)
- Beziehung zwischen ER und UML

3. Das relationale Modell

- Relationales Datenmodell (RM)
- Vom ER-Modell zum RM
- Relationale Algebra als Anfragesprache

4. Relationale Entwurfstheorie

- Funktionale Abhängigkeiten
- Normalformen

5. Structured Query Language (SQL)

- Datendefinition
- Datenmanipulation

6. Anfrageverarbeitung

- Architektur
- Indexierung
- Anfragepläne, Optimierung

7. Transaktionen

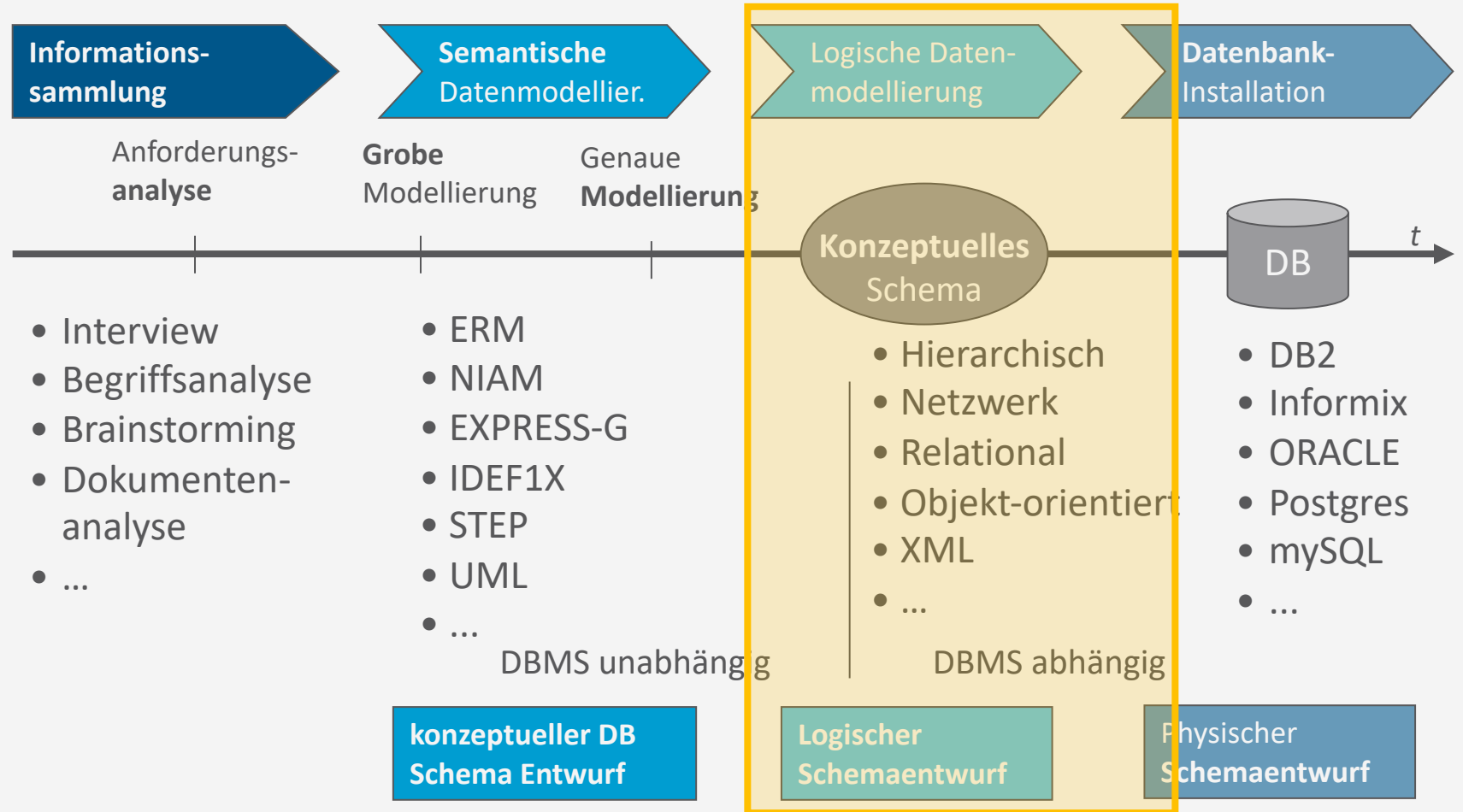
- Transaktionsverarbeitung, Schedules, Sperren
- Wiederherstellung

8. Verteilte Datenbanken

- Fragmentierung, Replikation, Allokation; CAP
- Anfragebeantwortung, föderierte Systeme

Phasen des DB-Entwurfs

- Ausblick: Von der Anwendung her
 - Teil von 2. DB-Modellierung
 - Methode: ERM
 - Teil von 3. Das relationale Datenmodell
 - Methode: relationale Modellierung
 - Teil von 4. DB-Entwurf
 - Teil von 5. SQL & Übergang zu „Hinter den Kulissen“



Übersicht: 3. Das Relationale Datenmodell

A. *Relationales Datenmodell*

- Relationen, Attribute, relationale Datenbanken und –schemata
- Schlüssel: Primärschlüssel, Fremdschlüssel, referentielle Integrität

B. *Entwurf relationaler Schemata*

- Vom ER-Diagramm zum relationalen Modell

C. **Relationale Algebra**

- $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
- Minimalität
- Aggregieren, gruppieren
- Einfügen, löschen, aktualisieren

Relationenschemata und Relationen

- $R(A_1, \dots, A_n)$: Relationenschema n -ten Grades
- $t = \langle v_1, \dots, v_n \rangle$: ein Tupel der Relation $r(R)$ (kurz: r)
 - v_i ist der Wert, der im Tupel t dem Attribut A_i entspricht
- $R.A$: ein Attribut A des Relationenschemas R
- Für einzelne Komponentenwerte von einem Tupel t gilt:
 - $t[A_i]$ bzw. $t.A_i$ beziehen sich auf den Wert v_i in t für Attribut A_i
 - $t[A_u, \dots, A_z]$ bzw. $t.(A_u, \dots, A_z)$ beziehen sich auf Werte $\langle v_u, \dots, v_z \rangle$ von Subtupeln von t , die den Attributen A_u, \dots, A_z von R entsprechen

Eigentlich sind Strings als solche zu markieren. Zur Übersicht lassen wir Anführungszeichen in dieser Vorlesung weg.

Relation r

Relationenschema R
Grad 4

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Beispiel eines DB-Schemas

STUDENT

Name	<u>StudentNumber</u>	Class	Major
------	----------------------	-------	-------

COURSE

CourseName	<u>CourseNumber</u>	CreditHours	Department
------------	---------------------	-------------	------------

SECTION

<u>SectionIdentifier</u>	<u>CourseNumber</u>	Semester	Year	Instructor
--------------------------	---------------------	----------	------	------------

GRADE_REPORT

<u>StudentNumber</u>	<u>SectionIdentifier</u>	Grade
----------------------	--------------------------	-------

PREREQUISITE

<u>CourseNumber</u>	<u>PrerequisiteNumber</u>
---------------------	---------------------------

Beispiel eines DB-Zustands

Was für Fragen würde man stellen wollen?

COURSE

CourseName	<u>CourseNumber</u>	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

STUDENT

Name	<u>StudentNumber</u>	Class	Major
Smith	17	1	CS
Brown	8	2	CS

SECTION

<u>SectionIdentifier</u>	<u>CourseNumber</u>	Semester	Year	Instructor
85	MATH2410	Fall	18	King
92	CS1310	Fall	18	Anderson
102	CS3320	Spring	19	Knuth
112	MATH2410	Fall	19	Chang
119	CS1310	Fall	19	Anderson
135	CS3380	Fall	19	Stone

GRADE_REPORT

<u>StudentNumber</u>	<u>SectionIdentifier</u>	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

<u>CourseNumber</u>	<u>PrerequisiteNumber</u>
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Anfragen an Relationen

- Entfernende Operatoren
 - Selektion σ
 - Projektion π
- Umbenennung ρ
- Klassische Mengenoperatoren (kombinieren Relationen)
 - Vereinigung \cup
 - Schnitt \cap
 - Differenz $-$
- Weitere kombinierende Operatoren
 - Kartesisches Produkt \times
 - Join \bowtie und weitere Join-Arten
 - Outer Union
 - Division
- Aggregieren, gruppieren
 - Über die klassische relationale Algebra hinaus
- *Spotlight*: Relationenzustände ändern
 - Einfügen, löschen, aktualisieren

Reihenfolge von Attributen

- Bemerke: Relationale Algebra ist eine Abstraktion, der Implementierungsdetail egal sind
- Effekt: Reihenfolge von Attributen in Relationen irrelevant
 - Implizite Zuordnung von Werten zu Attributen, basierend auf denen Vergleiche passieren

Selektion und Projektion

Entfernende Operatoren*

* Entfernen in dem Sinne, dass in einem Zwischenergebnis weniger Tupel oder Attribute vorkommen

Selektion σ

- Bildet eine Teilmenge von Tupeln einer Relation $r(R)$, die (jeweils) eine bestimmte Auswahlbedingung erfüllen:

Selektion = Auswahl
 Selektion \rightarrow σ (σ)

- $r' = \sigma_{\langle \text{Auswahlbedingung} \rangle}(r)$
- Unär: wird auf genau eine Relation angewendet
- Grad von $R' = \text{Grad von } R$
 - D.h., alle Attribute bleiben erhalten
- Kardinalität wird möglicherweise kleiner

Achtung: Die Eingabe in den Beispielen (hier und auf allen folgenden Folien) müsste die Menge der Tupel in der jeweiligen Tabelle sein. Wir nutzen den Namen der Relation als Platzhalter aus Platzgründen und der Übersichtlichkeit wegen.

- Beispiele:

- $\sigma_{\text{Department}=\text{CS}}(\text{COURSE})$
- $\sigma_{\text{CreditHours} \geq 4}(\text{COURSE})$
- $\sigma_{\text{CreditHours} \leq 3 \vee \text{Department}=\text{MATH}}(\text{COURSE})$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Selektion σ

- Kommutativ:

$$\sigma_{\langle \text{Bedingung1} \rangle} \left(\sigma_{\langle \text{Bedingung2} \rangle}(r) \right) = \sigma_{\langle \text{Bedingung2} \rangle} \left(\sigma_{\langle \text{Bedingung1} \rangle}(r) \right)$$

- Beispiel:

$$\sigma_{\text{CreditHours} \leq 3} \left(\sigma_{\text{Department} = \text{CS}}(\text{COURSE}) \right) = \sigma_{\text{Department} = \text{CS}} \left(\sigma_{\text{CreditHours} \leq 3}(\text{COURSE}) \right)$$

3 Tupel nach erster σ

2 Tupel nach erster σ

- Aber: Reihenfolge hat eine Auswirkung auf Größe des Zwischenergebnisses

- Zu nutze machen für effiziente Beantwortung ([→ Kapitel 6: Anfragebeantwortung](#))

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Selektion σ

- Kaskade von σ = Und-Verknüpfung der Bedingungen:

$$\sigma_{\langle Bed1 \rangle} \left(\sigma_{\langle Bed2 \rangle} \left(\dots \sigma_{\langle Bedn \rangle} (r) \right) \right) = \sigma_{\langle Bed1 \wedge Bed2 \wedge \dots \wedge Bedn \rangle} (r)$$

- Beispiel:

$$\sigma_{CreditHours \leq 3} \left(\sigma_{Department=CS}(COURSE) \right) = \sigma_{Department=CS \wedge CreditHours \leq 3}(COURSE)$$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Projektion π

- Wählt aus einer Relation r bestimmte Attribute aus und verwirft die anderen:

Projektion = Abbildung
 Projektion $\rightarrow \pi$ (π)

- $r' = \pi_{\langle \text{Attributliste} \rangle}(r)$
- Unär: wird auf genau eine Relation angewendet
- Grad $r' \leq \text{Grad } r$
 - idR fehlen nach π Attribute

- Beispiele:

$\pi_{\text{CourseName, CourseNumber}}(\text{COURSE})$

$\pi_{\text{CourseName}}(\text{COURSE})$

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Projektion π

- Es gilt:

$$\text{Wenn } \langle \text{Liste1} \rangle \subseteq \langle \text{Liste2} \rangle: \pi_{\langle \text{Liste1} \rangle} \left(\pi_{\langle \text{Liste2} \rangle}(r) \right) = \pi_{\langle \text{Liste1} \rangle}(r)$$

- Beispiel:

$$\pi_{\text{CourseName}} \left(\pi_{\text{CourseName, CourseNumber}}(\text{COURSE}) \right) = \pi_{\text{CourseName}}(\text{COURSE})$$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Projektion π

- Anzahl der Tupel (Kardinalität) kann sich verringern
 - **Mengeneigenschaft** entfernt Duplikate
- Beispiele:
 - $\pi_{CreditHours, Department}(COURSE)$
 - Projektionen der vorherigen Folien
 - $\pi_{CourseName, CourseNumber}(COURSE)$
 - $\pi_{CourseName}(COURSE)$
 - Keine doppelten Einträge
 - CourseName, CourseNumber eindeutig (Primärschlüssel bzw. Schlüsselkandidaten)

CreditHours	Department
4	CS
3	MATH
3	CS



CreditHours	Department
4	CS
4	CS
3	MATH
3	CS



CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Operationssequenzen und Renaming

Relationale Algebra

Sequenzen von Operationen

- Im Allgemeinen werden mehrere Operationen nacheinander ausgeführt
 - Einzelner Ausdruck oder Sequenz mit explizit benanntem Zwischenergebnis
- Beispiel

$$\bullet \pi_{CreditHours, Department}(\sigma_{CreditHours \geq 4}(COURSE))$$

$$\bullet MIN4 \leftarrow \sigma_{CreditHours \geq 4}(COURSE)$$

$$RESULTAT \leftarrow \pi_{CreditHours, Department}(MIN4)$$

- Relationen umbenennen möglich:

$$\bullet MIN4 \leftarrow \sigma_{CreditHours \geq 4}(COURSE)$$

$$COURSE4(Hours, Department)$$

$$\leftarrow \pi_{CreditHours, Department}(MIN4)$$

COURSE4

Hours	Department
4	CS



CreditHours	Department
4	CS



COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Umbenennung ρ

- Erlaubt die explizite Umbenennung von Relationen und Attributen
- Gegeben Ausgangsschema $R(A_1, \dots, A_n)$:
- Umbenennung von R in S und A_1, \dots, A_n in B_1, \dots, B_n

KURS

KursName	KursNr	SWS	Institut
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS



COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

- $\rho_{S(B_1, \dots, B_n)}(R)$
 - Verändert nicht Tupel, sondern Schemata
 - Unär: wird auf ein Schema angewendet
 - Häufig Hilfsoperation in Operationssequenzen
 - Auch bekannt als RENAME
- Beispiel:
 - $\rho_{KURS(KursName, KursNr, SWS, Institut)}(COURSE)$

Vereinigung, Schnitt, Differenz

Mengenoperatoren

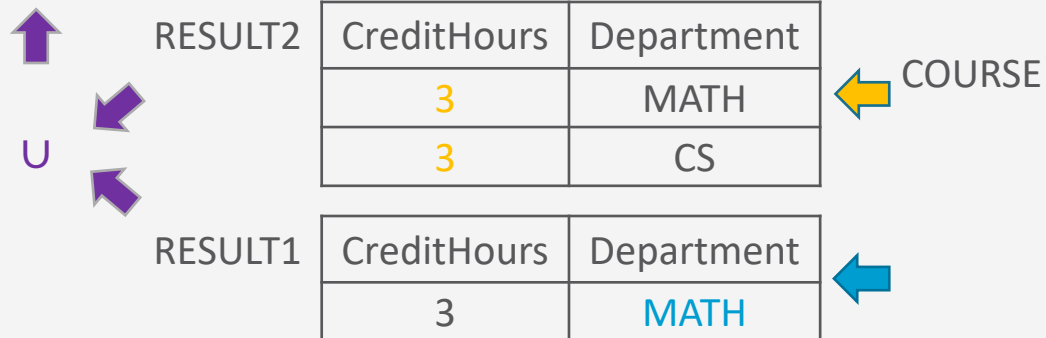
Mengenoperationen auf Relationen

- Vereinigung:
 - $r \cup s$ enthält alle Tupel, die in r , in s oder in beiden Relationen auftauchen
 - Duplikat-Tupel werden eliminiert
- Schnitt:
 - $r \cap s$ enthält nur Tupel, die in r und in s auftauchen
- Differenz:
 - $r - s$ enthält alle Tupel, die in r , jedoch nicht in s enthalten sind
- Auch bekannt als UNION, INTERSECTION, DIFFERENCE
- Binär: werden auf zwei Relationen angewendet
 - r und s können z.B. Ergebnisrelationen verschiedener Selektion auf derselben Ausgangsrelation sein → Dann auch durch UND, ODER, NICHT Verknüpfungen darstellbar

Vereinigung: Beispiel

- $RESULT1 \leftarrow \pi_{CreditHours, Department} \left(\sigma_{Department=MATH}(COURSE) \right)$
- $RESULT2 \leftarrow \pi_{CreditHours, Department} \left(\sigma_{CreditHours \leq 3}(COURSE) \right)$
- $RESULT \leftarrow RESULT2 \cup RESULT1$

RESULT	CreditHours	Department
	3	MATH
	3	CS

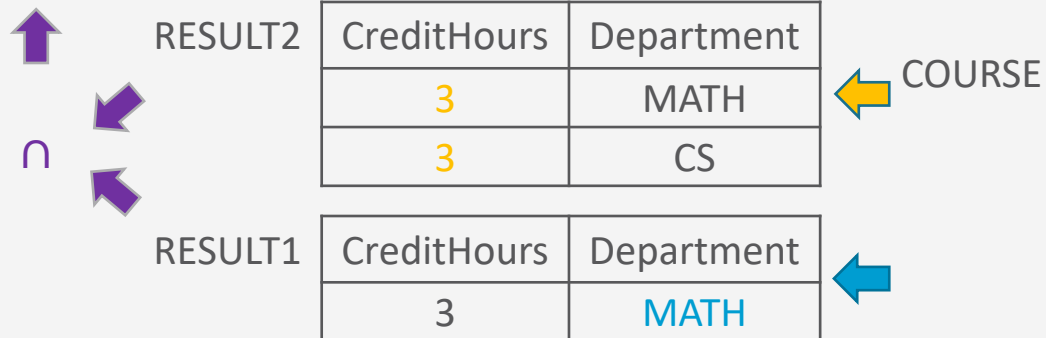


CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Schnitt: Beispiel

- $RESULT1 \leftarrow \pi_{CreditHours, Department} \left(\sigma_{Department=MATH}(COURSE) \right)$
- $RESULT2 \leftarrow \pi_{CreditHours, Department} \left(\sigma_{CreditHours \leq 3}(COURSE) \right)$
- $RESULT \leftarrow RESULT2 \cap RESULT1$

RESULT	CreditHours	Department
	3	MATH



CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Differenz: Beispiel

- $RESULT1 \leftarrow \pi_{CreditHours, Department} \left(\sigma_{Department=MATH}(COURSE) \right)$
- $RESULT2 \leftarrow \pi_{CreditHours, Department} \left(\sigma_{CreditHours \leq 3}(COURSE) \right)$
- $RESULT \leftarrow RESULT2 - RESULT1$

RESULT ← *RESULT1* − *RESULT2*?

RESULT	CreditHours	Department
	3	CS

RESULT2	CreditHours	Department
	3	MATH
	3	CS

← COURSE

RESULT1	CreditHours	Department
	3	MATH

← COURSE

↑
−
←

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Mengenoperatoren: Eigenschaften

- Können nur auf **UNION-kompatible** Relationenschemata angewendet werden:
 - R und S haben gleichen Grad n
 - Attribute haben gleiche Wertebereiche
 - $\text{dom}(A_i) = \text{dom}(B_i)$ für alle $1 \leq i \leq n$
 - Attribute müssen aber nicht gleich heißen
→ Umbenennung ρ
 - Konvention:
Namen aus dem ersten Schema R

- Vereinigung und Schnitt sind **kommutativ**

$$r \cup s = s \cup r$$

$$r \cap s = s \cap r$$

- Vereinigung und Schnitt sind **assoziativ**

$$(r \cup s) \cup t = s \cup (r \cup t)$$

$$(r \cap s) \cap t = s \cap (r \cap t)$$

- **Differenz?**

- Im Allgemeinen nicht kommutativ:

$$r - s \neq s - r$$

- Im Allgemeinen nicht assoziativ:

$$(r - s) - t \neq r - (s - t)$$

Kartesisches Produkt, Join, Outer Union, Division

Kombinierende Operatoren
Relationale Algebra

Kartesisches Produkt

- Alle Tupel zweier Relationen r und s werden kombinatorisch (vollständig: jedes Tupel aus r mit jedem Tupel aus s) miteinander verbunden
 - Gegeben $R(A_1, \dots, A_n)$ und $S(B_1, \dots, B_m)$
 - $r \times s = q(A_1, \dots, A_n, B_1, \dots, B_m)$
 - Binär: wird auf zwei Relationen angewendet
 - R und S müssen nicht UNION-kompatibel sein
 - Um eindeutige Attributbezeichnungen in der Ergebnisrelation zu gewährleisten, müssen Attribute, die in r und s gleich bezeichnet sind, vorher umbenannt werden
- Resultat
 - Grad: r hat n Spalten, s hat m Spalten $\rightarrow r \times s$ hat $(n + m)$ Spalten
 - Kardinalität: r hat k Zeilen, s hat l Zeilen $\rightarrow r \times s$ hat $(k \cdot l)$ Zeilen

Kartesisches Produkt: Beispiel

- $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=CS}(COURSE) \right)$
- $RES2 \leftarrow \rho_{Course,Instructor} \left(\pi_{CourseNumber,Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
- $CROSS \leftarrow RES1 \times RES2$
- $RESULT \leftarrow \pi_{CourseNumber,Instructor} \left(\sigma_{CourseNumber=Course}(CROSS) \right)$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS



RES1	CourseNumber
	CS1310
	CS3320
	CS3380

Kartesisches Produkt: Beispiel

- $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=cs}(COURSE) \right)$
- $RES2 \leftarrow \rho_{Course,Instructor} \left(\pi_{CourseNumber,Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
- $CROSS \leftarrow RES1 \times RES2$
- $RESULT \leftarrow \pi_{CourseNumber,Instructor} \left(\sigma_{CourseNumber=Course}(CROSS) \right)$

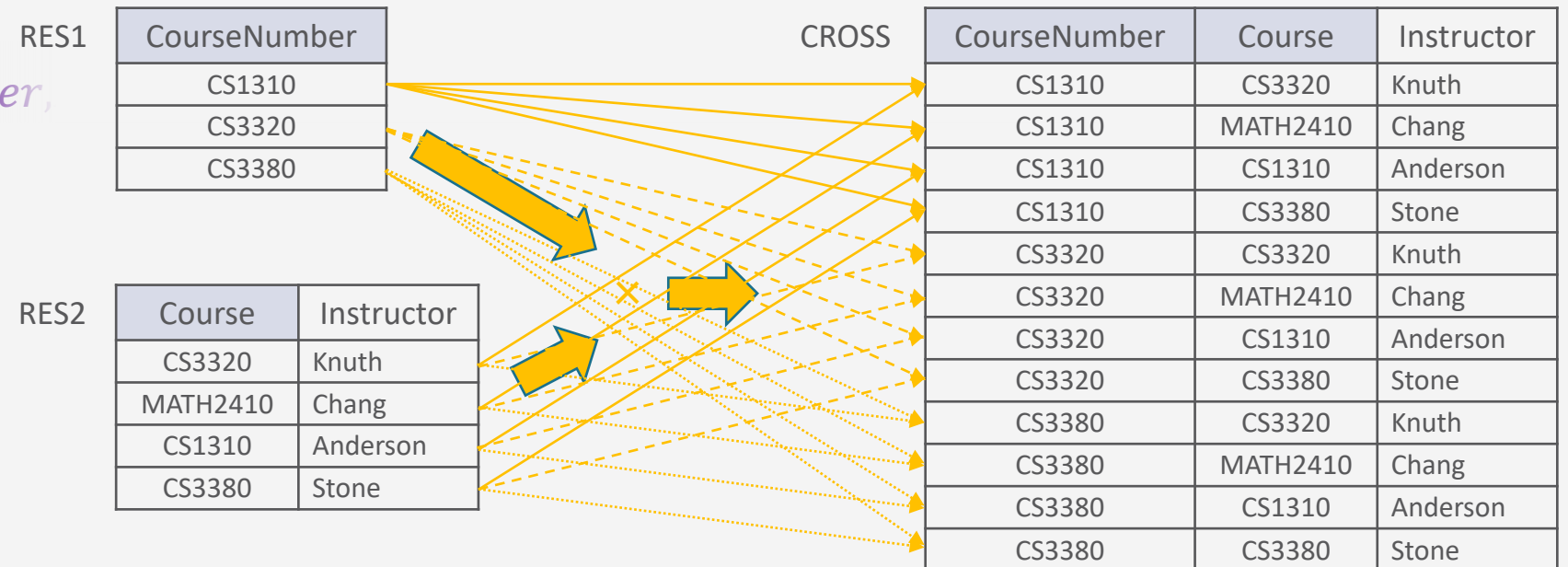
SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	18	King
	92	CS1310	Fall	18	Anderson
	102	CS3320	Spring	19	Knuth
	112	MATH2410	Fall	19	Chang
	119	CS1310	Fall	19	Anderson
	135	CS3380	Fall	19	Stone

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

Kartesisches Produkt: Beispiel

- $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=CS}(COURSE) \right)$
- $RES2 \leftarrow \rho_{Course, Instructor} \left(\pi_{CourseNumber, Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
- $CROSS \leftarrow RES1 \times RES2$
- $RESULT \leftarrow \pi_{CourseNumber, Course, Instructor}(CROSS)$



Kartesisches Produkt: Beispiel

- ...
- $CROSS \leftarrow RES1 \times RES2$
- $RESULT \leftarrow \pi_{CourseNumber, Instructor}(\sigma_{CourseNumber=Course}(CROSS))$

CROSS

CourseNumber	Course	Instructor
CS1310	CS3320	Knuth
CS1310	MATH2410	Chang
CS1310	CS1310	Anderson
CS1310	CS3380	Stone
CS3320	CS3320	Knuth
CS3320	MATH2410	Chang
CS3320	CS1310	Anderson
CS3320	CS3380	Stone
CS3380	CS3320	Knuth
CS3380	MATH2410	Chang
CS3380	CS1310	Anderson
CS3380	CS3380	Stone



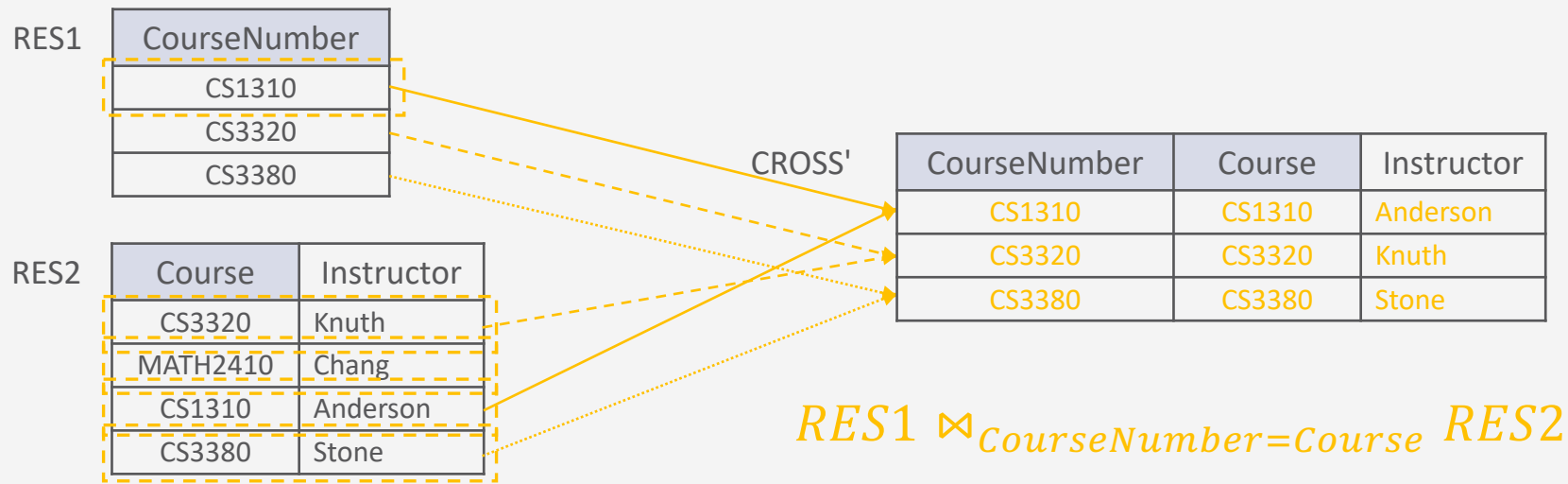
RES2

CourseNumber	Instructor
CS1310	Anderson
CS3320	Knuth
CS3380	Stone

Geht das auch einfacher?

Join \bowtie : Intuition

- $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=CS}(COURSE) \right)$
- $RES2 \leftarrow \rho_{Course,Instructor} \left(\pi_{CourseNumber,Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
- Die Tupel miteinander verbinden, wo
 $RES1.CourseNumber = RES2.Course$



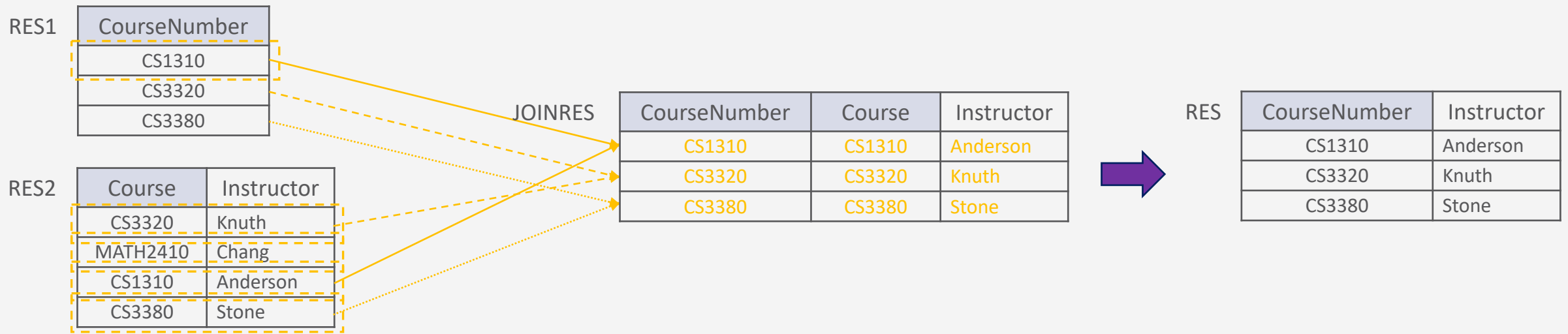
CourseNumber	Course	Instructor
CS1310	CS3320	Knuth
CS1310	MATH2410	Chang
CS1310	CS1310	Anderson
CS1310	CS3380	Stone
CS3320	CS3320	Knuth
CS3320	MATH2410	Chang
CS3320	CS1310	Anderson
CS3320	CS3380	Stone
CS3380	CS3320	Knuth
CS3380	MATH2410	Chang
CS3380	CS1310	Anderson
CS3380	CS3380	Stone

Join \bowtie

- Verbindet die Tupel zweier Relationen, die die **Join-Bedingung** erfüllen
 - Gegeben $R(A_1, \dots, A_n)$ und $S(B_1, \dots, B_m)$
 - $r \bowtie_{\langle \text{Bedingung} \rangle} s = q(A_1, \dots, A_n, B_1, \dots, B_m)$
 - q enthält alle Kombinationen von Tupeln, die der Bedingung entsprechen
 - Äquivalent zu kartesischem Produkt mit anschließender Selektion
 - $r \bowtie_{\langle \text{Bedingung} \rangle} s = \sigma_{\langle \text{Bedingung} \rangle}(r \times s)$
 - Zwischenergebnis kleiner bei Join
- Binär: werden auf zwei Relationen angewendet

Join ⋈: Beispiel

- $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=cs}(COURSE) \right)$
- $RES2 \leftarrow \rho_{Course,Instructor} \left(\pi_{CourseNumber,Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
- $JOINRES \leftarrow RES1 \bowtie_{CourseNumber=Course} RES2$
- $RES \leftarrow \pi_{CourseNumber,Instructor}(JOINRES)$



Join-Arten

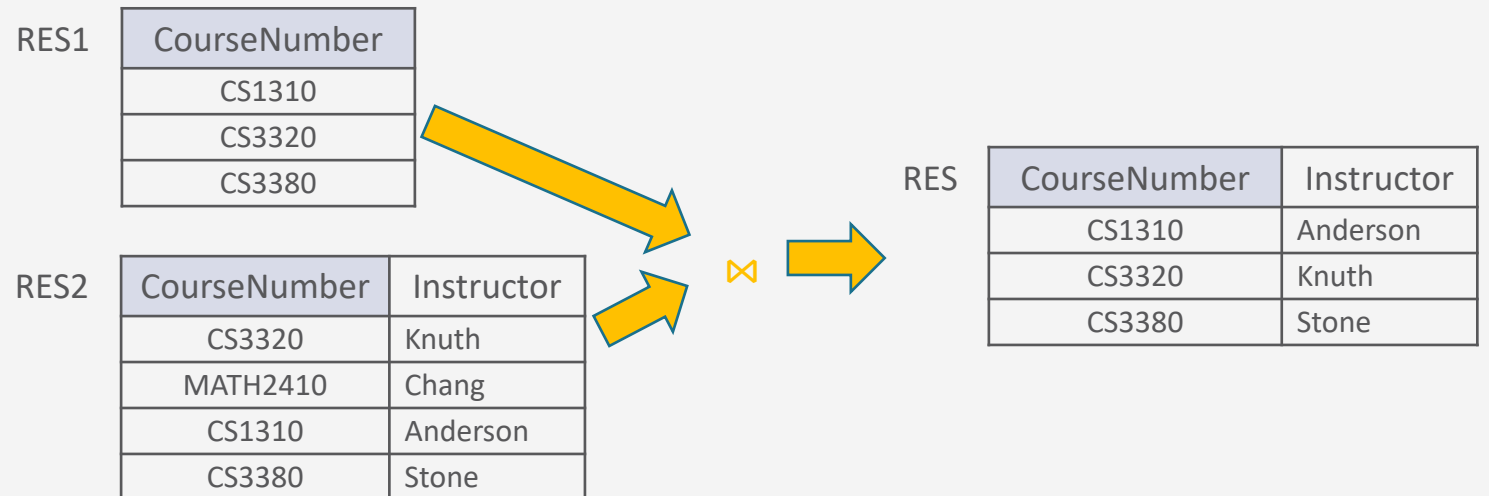
- Theta-Join
 - Jede (Teil-)Bedingung $A_i \theta B_j$ der Join-Bedingung
 - basiert auf einem θ aus $\{=, <, \leq, \geq, >, \neq\}$
- Equi-Join (Spezialfall)
 - θ ist $\{=\}$: es gibt nur eine Join-Bedingung, und sie prüft auf Gleichheit
 - Beispiel von vorher: Equi-Join
 - $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=CS}(COURSE) \right)$
 - $RES2 \leftarrow \rho_{Course,Instructor} \left(\pi_{CourseNumber,Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
 - $JOINRES \leftarrow RES1 \bowtie_{CourseNumber=Course} RES2$
 - $RES \leftarrow \pi_{CourseNumber,Instructor}(JOINRES)$

Join-Arten (Forts.)

- Natural Join (\bowtie ; ohne Bedingung im Subskript)
 - Join-Bedingung muss nicht angegeben werden: entspricht einem Equi-Join mit mehreren Attributen, die in beiden Relationen gleich heißen (\rightarrow RENAME)
 - Test auf Gleichheit bei allen Spaltenpaaren, die gleich heißen
 - **Doppelte Spalten werden entfernt**
 - Beispiel von vorher als Natural Join
 - $RES1 \leftarrow \pi_{CourseNumber} \left(\sigma_{Department=CS}(COURSE) \right)$
 - $RES2 \leftarrow \rho_{\overline{Course, Instructor}} \left(\pi_{CourseNumber, Instructor} \left(\sigma_{Year=19}(SECTION) \right) \right)$
 - $JOINRES \leftarrow RES1 \bowtie RES2$
 - ~~$RES \leftarrow \pi_{\overline{CourseNumber, Instructor}}(JOINRES)$~~

Join-Arten (Forts.)

- Beispiel von vorher als Natural Join
 - $RES1 \leftarrow \pi_{CourseNumber} (\sigma_{Department=CS}(COURSE))$
 - $RES2 \leftarrow \pi_{CourseNumber,Instructor} (\sigma_{Year=19}(SECTION))$
 - $JOINRES \leftarrow RES1 \bowtie RES2$



Join-Arten (Forts.)

- Left-/Right-/Full-Outer-Join
 $(r \bowtie s, r \ltimes s, r \bowtie\!\!\!\bowtie s)$
 - Tupel ohne Join-Partner kommen trotzdem ins Ergebnis
 - Fehlende Werte werden mit NULL aufgefüllt
 - **Left** Outer Join: alle Tupel von r
 - **Right** Outer Join: alle Tupel von s
 - **Full** Outer Join: alle Tupel von r, s
- Beispiele:
 - $LEFT \leftarrow RES' \bowtie_{CourseNumber=Course} RES2$
 - $RIGHT \leftarrow RES' \ltimes_{CourseNumber=Course} RES2$
 - $FULL \leftarrow RES' \bowtie\!\!\!\bowtie_{CourseNumber=Course} RES2$

RES'

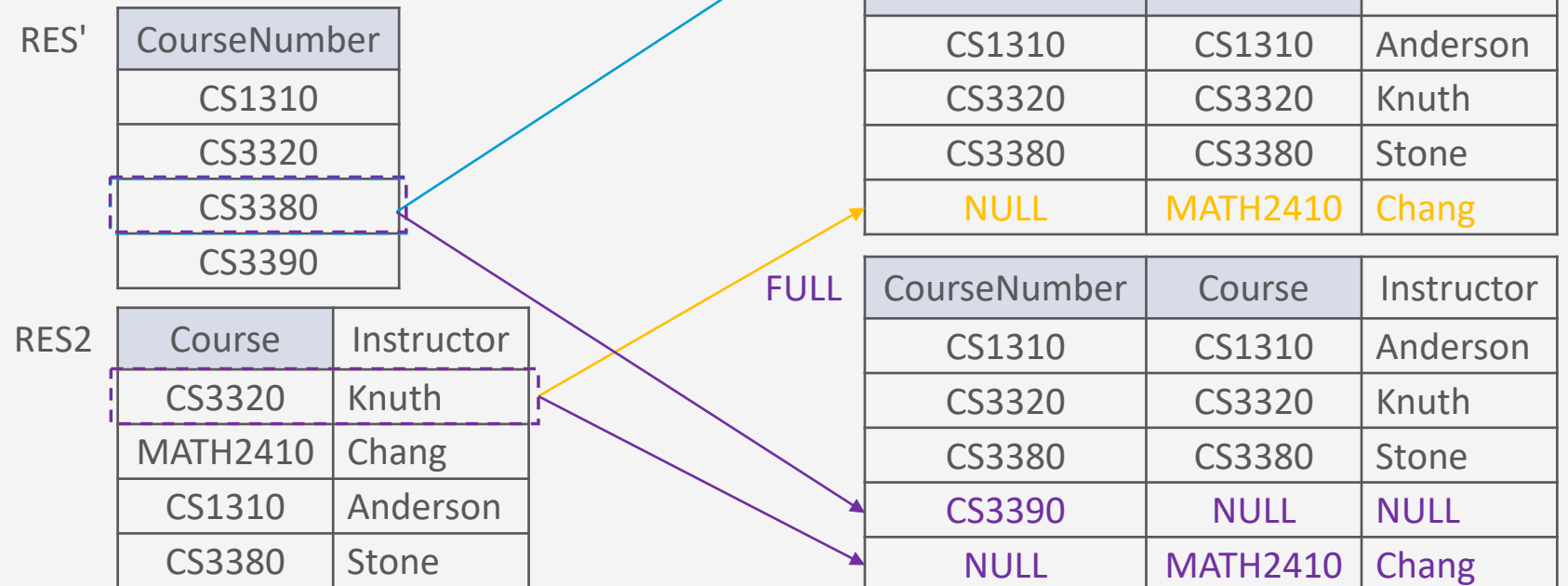
CourseNumber
CS1310
CS3320
CS3380
CS3390

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

Join-Arten (Forts.)

- $LEFT \leftarrow RES' \bowtie_{CourseNumber=Course} RES2$
- $RIGHT \leftarrow RES' \bowtie_{CourseNumber=Course} RES2$
- $FULL \leftarrow RES' \bowtie_{CourseNumber=Course} RES2$



Outer Union \cup

- Vereinigung von Tupeln, deren Relationen nicht UNION-kompatibel bzw. nur partiell UNION-kompatibel sind
 - Gegeben $R(A_1, \dots, A_n)$ und $S(B_1, \dots, B_m)$
 - $r \cup s = q(C_1, \dots, C_k)$
 - C_1, \dots, C_k beinhaltet die kompatiblen Attribute sowie die verbliebenen Attribute in r und s
 - Kompatible Attribute müssen nicht gleich heißen \rightarrow Umbenennung ρ
 - Konvention: Namen aus der ersten Relation r
 - Binär: wird auf zwei Relationen angewendet
 - **NULL**-Werte für Datenfelder, die dadurch für ein Tupel neu entstehen

Outer Union \cup

- Beispiel:

$$ALL-COURSES \leftarrow COURSE \cup \rho_{CourseName, CreditHours, Department, Module}(SEMINAR)$$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

SEMINAR	SeminarName	CreditHours	Department	Module
	Sorting	2	CS	A1
	Indexes	3	CS	A2
	Hashing	2	CS	A1

ALL-COURSES	CourseName	CourseNumber	CreditHours	Department	Module
-------------	------------	--------------	-------------	------------	--------

Outer Union \cup

- Beispiel:
 - $ALL-COURSES \leftarrow COURSE \cup \rho_{CourseName, CreditHours, Department, Module}(SEMINAR)$

ALL-COURSES	CourseName	CourseNumber	CreditHours	Department	Module
	Introduction to CS	CS1310	4	CS	NULL
	Data Structures	CS3320	4	CS	NULL
	Discrete Mathematics	MATH2410	3	MATH	NULL
	Databases	CS3380	3	CS	NULL
	Sorting	NULL	2	CS	A1
	Indexes	NULL	3	CS	A2
	Hashing	NULL	2	CS	A1

Division ÷

- $T(Y) = R(Z) \div S(X)$
 - Geht nur, wenn gilt: Attributmengen $X \subseteq Z$
 - Binär: auf zwei Relationen angewendet
 - Nicht sehr intuitiv → selten verwendet
- Sei $Y = Z - X$
- $T(Y)$ enthält ein Tupel t , wenn für jedes Tupel t_S in S ein Tupel t_R in R existiert, so dass gilt:

$$t_R[Y] = t \text{ und } t_R[X] = t_S$$
 - Jedes Ergebnistupel t muss mit jedem Tupel t_S aus S ein Tupel t_R in R erzeugen

- Beispiel
 - $R(A, B), Z = \{A, B\}, S(A), X = \{A\}$
 - $Y = \{B\} \rightarrow T(B)$

R	A	B
	a1	b1
	a2	b1
	a3	b1
	a4	b1
	a1	b2
	a3	b2
	a2	b3
	a3	b3
	a4	b3
	a1	b4
	a2	b4
	a3	b4

S	A
	a1
	a2
	a3

T	B
	b1
	b4

„Sammlle die B 's ein, die in R mit allen A 's auftreten, die in S vorkommen (a1, a2, a3).“

- b1: taucht mit a1, a2, a3 auf: ✓
 - b2: taucht mit a1, a3 auf: ✗
 - b3: taucht mit a2, a3 auf: ✗
 - b4: taucht mit a1, a2, a3 auf: ✓
- a4 irrelevant, da nicht in S.

Beispiel

- Ermittle alle Namen von Studenten, die in allen Kursen, die das Department CS anbietet, im Jahr 19 Prüfungen abgelegt haben
 1. $CS-COURSE \leftarrow \sigma_{Department=CS}(COURSE)$
 2. $19-SECTION \leftarrow \sigma_{Year=19}(SECTION)$
 3. $CS19-SECTION \leftarrow \pi_{SectionIdentifier}(19-SECTION \bowtie CS-COURSE)$
 4. $STN-SID \leftarrow \pi_{StudentNumber,SectionIdentifier}(GRADE-REPORT)$
 5. $CS19-STUDENT(StudentNumber) \leftarrow STN-SID \div CS19-SECTION$
 6. $RESULTAT \leftarrow \pi_{Name}(CS19-STUDENT \bowtie STUDENT)$

Beispiel

$$1. \quad CS-COURSE \leftarrow \sigma_{Department=CS}(COURSE)$$

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS



CS-COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Databases	CS3380	3	CS

Beispiel

2. $19\text{-SECTION} \leftarrow \sigma_{Year=19}(SECTION)$

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor
85	MATH2410	Fall	18	King
92	CS1310	Fall	18	Anderson
102	CS3320	Spring	19	Knuth
112	MATH2410	Fall	19	Chang
119	CS1310	Fall	19	Anderson
135	CS3380	Fall	19	Stone



19-SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor
102	CS3320	Spring	19	Knuth
112	MATH2410	Fall	19	Chang
119	CS1310	Fall	19	Anderson
135	CS3380	Fall	19	Stone

Beispiel

1. $CS-COURSE \leftarrow \sigma_{Department=CS}(COURSE)$
2. $19-SECTION \leftarrow \sigma_{Year=19}(SECTION)$
3. $CS19-SECTION \leftarrow \pi_{SectionIdentifier}(19-SECTION \bowtie CS-COURSE)$

CS-COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Databases	CS3380	3	CS

19-SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	102	CS3320	Spring	19	Knuth
	112	MATH2410	Fall	19	Chang
	119	CS1310	Fall	19	Anderson
	135	CS3380	Fall	19	Stone



CS19-SECTION	SectionIdentifier
	102
	119
	135

Beispiel

$$4. \quad STN-SID \leftarrow \pi_{StudentNumber, SectionIdentifier}(GRADE-REPORT)$$

GRADE-REPORT

StudentNumber	SectionIdentifier	Grade
17	112	B
17	119	C
8	85	A
8	119	A
8	102	B
8	135	A



STN-SID

StudentNumber	SectionIdentifier
17	112
17	119
8	85
8	119
8	102
8	135

Beispiel

3. $CS19-SECTION \leftarrow \pi_{SectionIdentifier}(19-SECTION \bowtie CS-COURSE)$
4. $STN-SID \leftarrow \pi_{StudentNumber, SectionIdentifier}(GRADE-REPORT)$
5. $CS19-STUDENT(StudentNumber) \leftarrow STN-SID \div CS19-SECTION$

STN-SID	StudentNumber	SectionIdentifier
	17	112
	17	119
	8	85
	8	119
	8	102
	8	135

CS19-SECTION	SectionIdentifier
	102
	119
	135

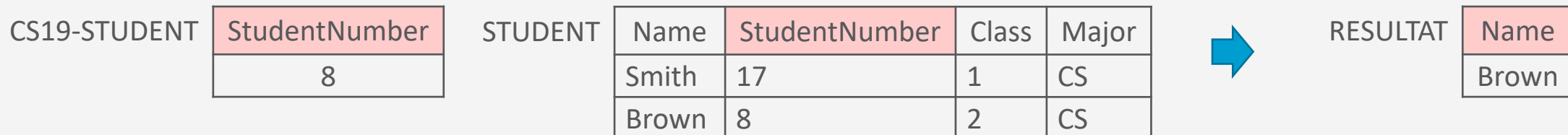


CS19-STUDENT	StudentNumber
	8

Beispiel

- Ermittle alle Namen von Studenten, die in allen Kursen, die das Department CS anbietet, im Jahr 19 Prüfungen abgelegt haben

- $CS-COURSE \leftarrow \sigma_{Department=CS}(COURSE)$
- $19-SECTION \leftarrow \sigma_{Year=19}(SECTION)$
- $CS19-SECTION \leftarrow \pi_{SectionIdentifier}(19-SECTION \bowtie CS-COURSE)$
- $STN-SID \leftarrow \pi_{StudentNumber,SectionIdentifier}(GRADE-REPORT)$
- $CS19-STUDENT(StudentNumber) \leftarrow STN-SID \div CS19-SECTION$
- $RESULTAT \leftarrow \pi_{Name}(CS19-STUDENT \bowtie STUDENT)$



Minimalität

Relationale Algebra

Minimalität der relationalen Algebra

- Minimale Operatormenge
 - Selektion (σ) und Projektion (π)
 - Umbenennung (ρ)
 - Vereinigung (\cup) und Differenz ($-$)
 - Kartesisches Produkt (\times)
- Weitere Operatoren durch minimale Operatormenge ausdrückbar
 - Schnitt (\cap): $r - (s - t)$
 - Join (\bowtie): $\sigma_{\langle \dots \rangle}(r \times s)$

Minimalität der relationalen Algebra

- Argumentation für die Minimalität
 - Umbenennung (ρ) kann nicht durch eine der anderen fünf Operatoren ersetzt werden
 - ρ hat zudem keinen Einfluss auf die Darstellung eines der anderen Operatoren mithilfe der noch verbleibenden fünf Operatoren ($\sigma, \pi, \cup, -, \times$)
Daher wird auf ρ im Folgenden nicht weiter eingegangen
 - Verbleiben also die Operatoren $\sigma, \pi, \cup, -, \times$ bzgl. der Untersuchung, ob man welche davon ohne Verlust der Ausdrucksmöglichkeiten streichen kann

Untersuchung der verbliebenen Operatoren

- Kartesisches Produkt (\times):
 - Kann nicht simuliert werden, da $\{\sigma, \pi, \cup, -\}$ ein Schema nicht erweitern können
- Projektion (π):
 - Analoges Argument: keiner der Operatoren $\{\sigma, \cup, -, \times\}$ kann ein Schema reduzieren
- Selektion (σ):
 - Kann höchstens durch Differenz ($-$) simuliert werden
 - Differenz testet jedoch nur auf Gleichheit ganzer Tupel und nicht auf beliebige Vergleiche durch Formeln, die sich auf Komponenten von Tupeln beziehen
- Differenz ($-$)
 - Kann die Selektion nicht simulieren, da σ die „Negation“ auf Relationen nicht darstellen kann.
- Vereinigung (\cup)
 - Kann nicht durch die Operatoren $\{\sigma, \pi, -, \times\}$ dargestellt werden.

Aggregatfunktion und Gruppierung

Relationale Algebra

Aggregatsfunktionen

- Aggregiert mehrere Tupel zu einem Tupel bzgl. eines Attributes A einer Relation r
 - $\mathcal{F}_{\langle \text{Liste von (Funktion, Attribut } A \text{) Paaren} \rangle}(r)$
 - Abbildung in den Wertebereich von A
- Standard-Aggregationsfunktionen:
 - $\mathcal{F}_{\text{MIN } A}(r)$ Minimaler Wert, den A in r annimmt
 - $\mathcal{F}_{\text{MAX } A}(r)$ Maximaler Wert, den A in r annimmt
 - $\mathcal{F}_{\text{AVG } A}(r)$ Durchschnittlicher Wert von A über alle Tupel in r
 - $\mathcal{F}_{\text{SUM } A}(r)$ Summe der Werte von A über alle Tupel in r
 - $\mathcal{F}_{\text{COUNT } A}(r)$ Anzahl der Tupel, bei denen $A \neq \text{NULL}$ in r
 - $\mathcal{F}_{\text{COUNT}^*}(r)$ ohne Attribut $(*)$ = Kardinalität von r Häufig im Anschluss an eine Selektion
 - Setzt voraus, dass im Wertebereich des aggregierten Attributs eine Ordnung (bei MIN, MAX) bzw. Rechenoperationen (bei AVG, SUM) definiert sind

Aggregatsfunktionen: Beispiele

- $RMIN(Min) \leftarrow \mathcal{F}_{MIN\ CreditHours}(COURSE)$
- $RMAX(Max) \leftarrow \mathcal{F}_{MAX\ CreditHours}(COURSE)$
- $RAVG(Avg) \leftarrow \mathcal{F}_{AVG\ CreditHours}(COURSE)$
- $RSUM(Sum) \leftarrow \mathcal{F}_{SUM\ CreditHours}(COURSE)$
- $RCNT(Cnt) \leftarrow \mathcal{F}_{COUNT\ CreditHours}(COURSE)$

- $\mathcal{F}_{COUNT*}(COURSE)?$

- Mit Projektion:

Anzahl an Kursen vom Department CS

$$\mathcal{F}_{COUNT*}(\sigma_{Department=CS}(COURSE))$$

- $\mathcal{F}_{COUNT\ Department}(COURSE)?$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS
	Algorithms	CS3390	4	NULL

RMIN	Min	RMAX	Max	RAVG	Avg	RSUM	Sum	RCNT	Cnt
	3		4		3.5		14		4

Gruppierung

- Bildet Gruppen von Tupeln, die in einer Attributmengende die gleichen Werte haben
 - Häufig zur Vorbereitung einer Aggregation
 - Notation: Attributliste vor den Ausdruck
 - $\langle B_1, \dots, B_m \rangle \mathcal{F}_{\langle \text{Liste von (Funktion, Attribut) Paaren} \rangle} (r)$
 - Beispiel:
 - Bestimme die durchschnittliche Stundenzahl der Kurse der Departments

$GAVG(\text{Department}, \text{Cnt}, \text{Avg})$

$\leftarrow \text{Department } \mathcal{F}_{\text{COUNT CourseNumber, AVG CreditHours}}(\text{COURSE})$

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS



GAVG	Department	Cnt	Avg
	CS	3	3.7
	MATH	1	3

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Gruppierung: Unterschiede

- Beispiel:
 - $\mathcal{F}_{\text{COUNT CourseNumber,AVG CreditHours}}(\text{COURSE})$
 - *Department* $\mathcal{F}_{\text{COUNT CourseNumber,AVG CreditHours}}(\text{COURSE})$
 - *GAVG(Department, Cnt, Avg)*
 ← *Department* $\mathcal{F}_{\text{COUNT CourseNumber,AVG CreditHours}}(\text{COURSE})$

COUNT_CourseNumbers	AVG_CreditHours
4	3.5

Department	COUNT_CourseNumbers	AVG_CreditHours
<i>CS</i>	3	3.7
<i>MATH</i>	1	3

GAVG

Department	Cnt	Avg
CS	3	3.7
MATH	1	3

Einfügen, Löschen, Aktualisieren

Änderung von Relationenzuständen

Mengenorientierte Spezifizierung von Änderungsoperatoren

- Änderungen am Relationenzustand über Mengenoperationen realisierbar
 - Gegeben Relationen r, s, t über die gleichen Attribute
 - s, t kann eine Menge von Tupeln sein oder ein komplexer relationaler Ausdruck, der in die gleichen Attribute endet
 - *Einfügen*: Vereinigung $r \leftarrow r \cup s$
 - *Löschen*: Differenz $r \leftarrow r - s$
 - *Aktualisieren*: $r \leftarrow (r - s) \cup t$ oder ein relationaler Ausdruck, der Tupel aktualisiert: $r \leftarrow \xi(r)$
- Änderungen
 - Einfügen, löschen: ganze Tupel betroffen
 - Aktualisieren: Werte einzelner Attribute ändern, Attribute hinzufügen (z.B. über Funktionen)
- Deklarative Spezifizierung auch mittels INSERT, DELETE, UPDATE bekannt

Einfügen von Tupeln

- **INSERT INTO** $R(A_1, \dots, A_n)$ // oder: **INSERT INTO** R
VALUES $\langle v_1, \dots, v_n \rangle$

- Eingabe:
eine Liste von Attributwerten für ein neues Tupel $t = \langle v_1, \dots, v_n \rangle$,
das in die Relation r bzw. $r(R)$ eingefügt werden soll

- Beispiel:

- **INSERT INTO** COURSE
VALUES <Algorithms, CS3390, 4, CS>

- Primärschlüssel noch nicht vorhanden
- Werte der Attribute liegen in den Domänen der Attribute
- Kein zu prüfender Fremdschlüssel in Tupel

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS
Algorithms	CS3390	4	CS


 Was sollte man vorher checken?

Einfügen von Tupeln: Fehlersituationen

- **INSERT** t erzeugt **Fehler**
→ Führt zur Abweisung der INSERT Operation (Konsistenz bewahren!)
- Fehlersituationen
 - *Wertebereichseinschränkungen*: v_i entspricht nicht dem für A_i festgelegten Wertebereich
 - **INSERT** ... <Algorithms, CS3390, **four**, CS> bei z.B. $\text{dom}(\text{CreditHours}) = \text{Integer}$
 - *Schlüsseleinschränkungen*: Primärschlüsselwert in t existiert schon in $r(R)$
 - **INSERT** ... <Algorithms, **CS3380**, 4, CS>
 - *Entitätsintegrität*: Primärschlüssel / Teil des Primärschlüssels in t hat den Wert NULL
 - **INSERT** ... <Algorithms, **NULL**, 4, CS>

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Einfügen von Tupeln: Fehlersituationen

- **INSERT** t erzeugt **Fehler**
→ Führt zur Abweisung der INSERT Operation (Konsistenz bewahren!)
- Fehlersituationen (Forts.)
 - *Referenzielle Integrität:*
Wert eines Fremdschlüssels in t referenziert ein Tupel s in einer Relation S , welches dort gar nicht existiert
 - **INSERT INTO SECTION VALUES** (142, **CS3390**, Fall, 19, Anderson)
 - Vorherige Fehler können also spätere Fehler nach sich ziehen

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor
85	MATH2410	Fall	18	King
92	CS1310	Fall	18	Anderson
102	CS3320	Spring	19	Knuth
142	CS3390	Fall	19	Anderson

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Löschen von Tupeln

- **DELETE FROM R**
[WHERE <Bedingung>]

- Löscht eine Menge von Tupeln $\{t_k\}_{k=1}^K$ aus einer Relation $r(R)$
- Spezifiziert über Bedingungen
- Beispiele:
 - Bestimmtes Tupel über Wert v des Primärschlüssels P referenziert (löscht ein Tupel)
 - **DELETE FROM COURSE**
WHERE COURSE.CourseNumber = CS3380
 - Alle Tupel, bei denen z.B. ein Attribut A größer einem Wert w ist (löscht mehrere Tupel)
 - **DELETE FROM COURSE**
WHERE COURSE.CreditHours >= 4



COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

wird gelöscht (points to CS1310)

wird gelöscht (points to MATH2410)

Löschen von Tupeln: Fehlersituation

- Fehlersituation: *Referenzielle Integrität*
 - In einer anderen Relation S gibt es einen Fremdschlüssel auf R und ein Tupel s in S referenziert das zu löschende Tupel t_k
 - **DELETE FROM COURSE**
WHERE COURSE.CourseNumber=CS3380

Was können wir machen um die Konsistenz zu bewahren?

Fremdschlüssel

SECTION

SectionIdentifier	CourseNumber
85	MATH2410
92	CS1310
102	CS3320
112	MATH2410
119	CS1310
135	CS3380

Primärschlüssel

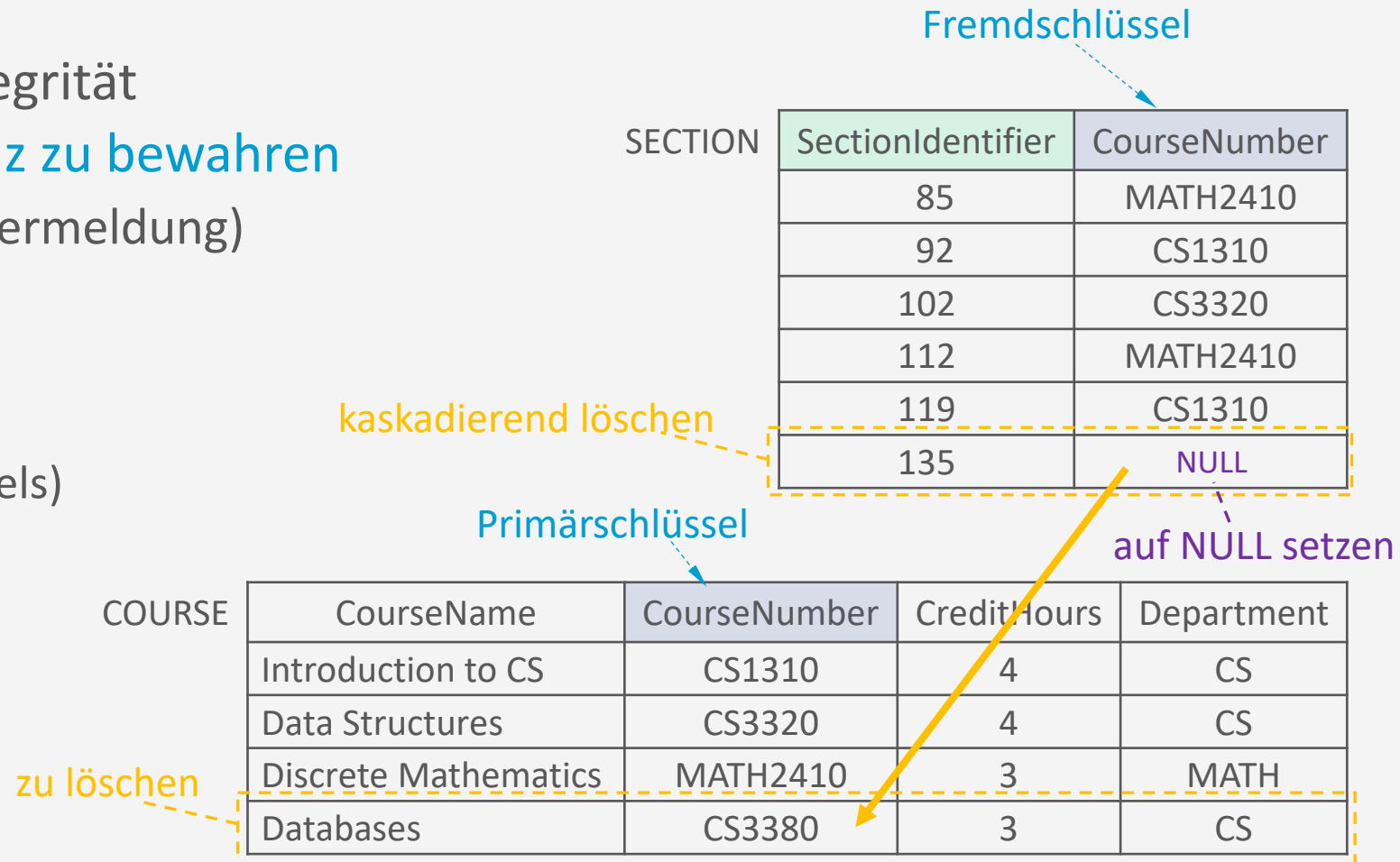
COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

zu löschen

Löschen von Tupeln: Fehlersituation

- Fehlersituation: Referenzielle Integrität
- Lösungsansätze um die Konsistenz zu bewahren
 - DELETE Operation abweisen (Fehlermeldung)
 - Kaskadierend löschen
 - Betroffene Tupel korrigieren
 - Fremdschlüssel auf NULL setzen (wenn nicht Teil des Primärschlüssels)
 - Fremdschlüssel auf Default-Wert setzen (wenn vorhanden)
 - Fremdschlüssel auf existierenden Schlüsselwert setzen



Aktualisieren von Tupeln

- **UPDATE R**
SET ...

[**WHERE** <Bedingung>]

- Aktualisiert / ändert eine Menge von Tupeln aus einer Relation
- Identifikation bestimmter Tupel über Schlüsselwerte/Bedingungen
 - **UPDATE COURSE**
SET Course.CreditHours = 4
WHERE Course.CourseNumber = CS3380
 - **UPDATE COURSE**
SET Course.CreditHours = 4
WHERE Course.CreditHours = 3



COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete MathEmatics	MATH2410	4	MATH
Databases	CS3380	4	CS

Aktualisieren von Tupeln

- Fehlersituationen
 - Wenn keine Primär- oder Fremdschlüssel geändert werden:
 - Nur Wertebereichseinschränkungen
 - **UPDATE COURSE**
SET Course.CreditHours = **four**
WHERE Course.CourseNumber = CS3380
 - Kein Tupel durch Bedingung angesprochen (keine Auswirkung!)
 - **UPDATE COURSE**
SET Course.CreditHours = 4
WHERE Course.CourseNumber = **CS3390**
- Ansonsten alle Probleme von **INSERT** und **DELETE**

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Aktualisieren von Tupeln

- Fehlersituationen
 - Ansonsten alle Probleme von **INSERT** und **DELETE**; Beispiele
 - Primärschlüsseländerungen: Neuer Schlüsselwert schon belegt?
 - **UPDATE COURSE SET** Course.CourseNumber = **CS1310** ...
 - Referenz auf alten Schlüssel in anderer Relation vorhanden? → Kaskadierend aktualisieren
 - **UPDATE COURSE SET** Course.CourseNumber = **CS3390** ...
 - Fremdschlüsseländerungen: Existiert neuer Fremdschlüsselwert in referenzierter Relation?
 - **UPDATE** Section.CourseNumber = **CS3390** ...

- **Lösungen:**
Prinzipiell gleiche Optionen wie bei **DELETE**

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Zwischenzusammenfassung

- Relationale Algebra als Anfragesprache an Relationen
- Entfernde Operatoren
 - Selektion σ , Projektion π
- Umbenennung ρ
- Kombinerende Operatoren
 - Klassisch: Vereinigung \cup , Schnitt \cap , Differenz $-$
 - Kartesisches Produkt \times , Join \bowtie und weitere Join-Arten, Outer Union, Division
- Minimalität der relationalen Algebra
- Aggregieren, gruppieren
- Relationenzustände ändern
 - Einfügen, löschen, aktualisieren

Übersicht: 3. Das Relationale Datenmodell

A. *Relationales Datenmodell*

- Relationen, Attribute, relationale Datenbanken und -schemata
- Schlüssel: Primärschlüssel, Fremdschlüssel, referentielle Integrität

B. *Entwurf relationaler Schemata*

- Vom ER-Diagramm zum relationalen Modell

C. *Relationale Algebra*

- $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
- Minimalität
- Aggregieren, gruppieren
- Einfügen, löschen, aktualisieren

→ Datenbank-Entwurf