WWU MÜNSTER

Das relationale Modell: Relationale Algebra

Datenbanken

RES1

CourseNumber
CS1310
CS3320
CS3380

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

$$RES \leftarrow RES1 * RES2$$

RES

CourseNumber	Instructor
CS1310	Anderson
CS3320	Knuth
CS3380	Stone

wissen.leben

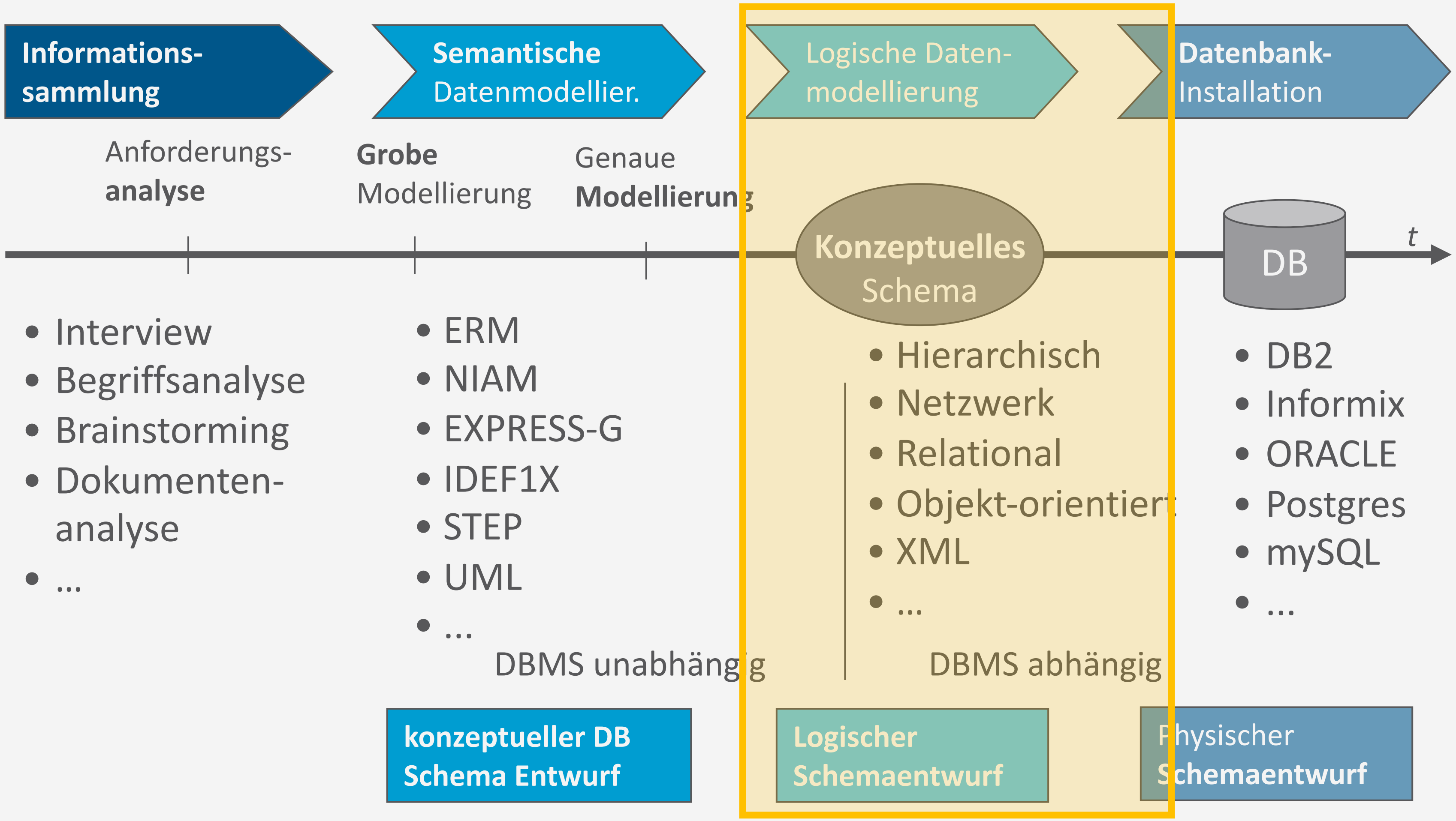
Tanya Braun
Arbeitsgruppe Data Science, Institut für Informatik

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. **Datenbank-Entwurf**
   - Funktionale Abhängigkeiten
   - Normalformen
5. **Structured Query Language (SQL)**
   - Datendefinition
   - Datenmanipulation
6. **Anfrageverarbeitung**
   - Architektur
   - Indexierung
   - Anfragepläne, Optimierung
7. **Transaktionen**
   - Transaktionsverarbeitung, Schedules, Sperren
   - Wiederherstellung
8. **Erweiterung**
   - Noch offen: verteilte DBs, deduktive DBs (DataLog → Logik-Verbindung), XML, Graph-DBs

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 $r(R)$
  - 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

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	StudentNumber	Class	Major	
COURSE	CourseName	CourseNumber	CreditHours	Department	
SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
GRADE_REPORT	StudentNumber	SectionIdentifier	Grade		
PREREQUISITE	CourseNumber	PrerequisiteNumber			

Beispiel eines DB-Zustands

Was für Fragen würde man stellen wollen?

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

STUDENT

Name	StudentNumber	Class	Major
Smith	17	1	CS
Brown	8	2	CS

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

GRADE_REPORT

StudentNumber	SectionIdentifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

CourseNumber	PrerequisiteNumber
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

Selektion und Projektion

Entfernende Operatoren*

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

Selektion σ

> Selektion = Auswahl
> Selektion → sigma (σ)

- Bildet eine Teilmenge von Tupeln einer Relation, die (jeweils) eine bestimmte Auswahlbedingung erfüllen:
  - $R' = \sigma_{\langle Auswahlbedingung \rangle}(R)$
  - Unär: wird auf genau eine Relation angewendet
  - Grad von R' = Grad von R
    - D.h., alle Attribute bleiben erhalten
  - Kardinalität wird möglicherweise kleiner
- Beispiele:
  - $\sigma_{Department=CS}(COURSE)$
  - $\sigma_{CreditHours \geq 4}(COURSE)$
  - $\sigma_{CreditHours \leq 3 \, \vee Deparment=MATH}(COURSE)$

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:

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

- Beispiel:

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

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:

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

  - 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

> Projektion = Abbildung
> Projektion → pi (π)

- Wählt aus einer Relation bestimmte Attribute aus und verwirft die anderen:
  - $R' = \pi_{\langle Attributliste \rangle}(R)$
  - Unär: wird auf genau eine Relation angewendet
  - Grad R' ≤ Grad R
    - idR fehlen nach π Attribute
- Beispiele:
  - $\pi_{CourseName, CourseNumber}(COURSE)$
  - $\boxed{\pi_{CourseName}(COURSE)}$

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:

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

  - Beispiel:

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

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

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

Operationssequenzen und Renaming

Relationale Algebra

Sequenzen von Operationen

- Im Allgemeinen werden mehrere Operationen nacheinander ausgeführt
  - Einzelner Ausdruck oder Sequenz mit explizit benanntem Zwischenergebnis
- Beispiel
  - $\pi_{CreditHours,Department}\left(\sigma_{CreditHours \geq 4}(COURSE)\right)$
  - $MIN4 \leftarrow \sigma_{CreditHours \geq 4}(COURSE)$
    $RESULTAT \leftarrow \pi_{CreditHours,Department}(MIN4)$
- Relationen umbenennen möglich:
  - $MIN4 \leftarrow \sigma_{CreditHours \geq 4}(COURSE)$
    $COURSE4(Hours, Department)$
    $\leftarrow \pi_{CreditHours,Department}(MIN4)$

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 Ausgangsrelation $R(A_1, \dots, A_n)$:
- Umbenennung von R in S und $A_1, \dots, A_n$ in $B_1, \dots, B_n$
  - $\rho_{S(B_1,\dots,B_n)}(R)$
    - Verändert nicht Tupel, sondern Schemata
    - Unär: wird auf ein Schema angewendet
    - Häufig Hilfsoperation in Operationssequenzen
    - Auch bekannt als RENAME
  - Beispiel:
    - $\rho_{KURS(KursName,KursNr,SWS,Institut)}(COURSE)$

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

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 dieselbe Relation sein, d.h., $R = S$
    - Dann auch durch UND, ODER, NICHT Verknüpfungen darstellbar

Vereinigung: Beispiel

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

RESULT

CreditHours	Department
3	MATH
3	CS

∪

RESULT2

CreditHours	Department
3	MATH
3	CS

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

Schnitt: Beispiel

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

RESULT

CreditHours	Department
3	MATH

$\cap$

RESULT2

CreditHours	Department
3	MATH
3	CS

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

Differenz: Beispiel

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

$RESULT \leftarrow RESULT1 - RESULT2?$

RESULT

CreditHours	Department
3	CS

$-$

RESULT2

CreditHours	Department
3	MATH
3	CS

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 Relationen angewendet werden:
  - R und S haben gleichen Grad n
  - Attribute haben gleiche Wertebereiche
    - $\mathrm{dom}(A_i) = \mathrm{dom}(B_i)$ für alle $1 \leq i \leq n$
  - Attribute müssen aber nicht gleich heißen
    $\rightarrow$ Umbenennung ρ
    - Konvention:
      Namen aus der ersten Relation 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:
    $$(R - S) - T \neq S - (R - T)$$

Kartesisches Produkt, Join, Outer Union, Division

Kombinierende Operatoren
Relationale Algebra

Kartesisches Produkt

- Alle Tupel zweier Relationen R und S werden kombinatorisch („vollständig“) 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 → $R \times S$ hat $(n + m)$ Spalten
  - Kardinalität: R hat k Zeilen, S hat l Zeilen → $R \times S$ 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

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

RES1

CourseNumber
CS1310
CS3320
CS3380

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

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

Kartesisches Produkt: Beispiel

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

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

Geht das auch einfacher?

RES2

CourseNumber	Instructor
CS1310	Anderson
CS3320	Knuth
CS3380	Stone

Join ⋈: 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 $RES1.CourseNumber = RES2.Course$

RES1

CourseNumber
CS1310
CS3320
CS3380

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

CROSS'

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

$RES1 \bowtie_{CourseNumber=Course} RES2$

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

Join $\bowtie$

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

Join ⋈: Beispiel

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

RES1

CourseNumber
CS1310
CS3320
CS3380

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

JOINRES

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

RES

CourseNumber	Instructor
CS1310	Anderson
CS3320	Knuth
CS3380	Stone

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 ($*$)
  - Join-Bedingung muss nicht angegeben werden: entspricht einem Equi-Join mit mehreren Attributen, die in beiden Relationen gleich heißen (→RENAME)
  - Doppelte Spalten werden entfernt
  - Beispiel von vorher als Natural Join
    - $RES1 \leftarrow \pi_{CourseNumber}\left(\sigma_{Department=CS}(COURSE)\right)$
    - $RES2 \leftarrow \cancel{\rho_{Course,Instructor}}\left(\pi_{CourseNumber,Instructor}\left(\sigma_{Year=19}(SECTION)\right)\right)$
    - $JOINRES \leftarrow RES1 * RES2$
    - ~~$RES \leftarrow \pi_{CourseNumber,Instructor}(JOINRES)$~~

Join-Arten (Forts.)

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

RES1

CourseNumber
CS1310
CS3320
CS3380

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

$*$

RES

CourseNumber	Instructor
CS1310	Anderson
CS3320	Knuth
CS3380	Stone

Join-Arten (Forts.)

- Left-/Right-/Full-Outer-Join ($R ⟕ S$, $R ⟖ S$, $R ⟗ S$)
  - 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:
  - $LEFT \leftarrow RES' ⟕_{CourseNumber=Course} RES2$
  - $RIGHT \leftarrow RES' ⟖_{CourseNumber=Course} RES2$
  - $FULL \leftarrow RES' ⟗_{CourseNumber=Course} RES2$

RES'

CourseNumber
CS1310
CS3320
CS3380
CS3390

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

Join-Arten (Forts.)

- $LEFT \leftarrow RES' ⟕_{CourseNumber=Course} RES2$
- $RIGHT \leftarrow RES' ⟖_{CourseNumber=Course} RES2$
- $FULL \leftarrow RES' ⟗_{CourseNumber=Course} RES2$

RES'

CourseNumber
CS1310
CS3320
CS3380
CS3390

RES2

Course	Instructor
CS3320	Knuth
MATH2410	Chang
CS1310	Anderson
CS3380	Stone

LEFT

CourseNumber	Course	Instructor
CS1310	CS1310	Anderson
CS3320	CS3320	Knuth
CS3380	CS3380	Stone
CS3390	NULL	NULL

RIGHT

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

FULL

CourseNumber	Course	Instructor
CS1310	CS1310	Anderson
CS3320	CS3320	Knuth
CS3380	CS3380	Stone
CS3390	NULL	NULL
NULL	MATH2410	Chang

Outer Union $\uplus$

- 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 → 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 $\uplus$

- 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 $\uplus$

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

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

„Sammle 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. $CS\text{--}COURSE \leftarrow \sigma_{Department=CS}(COURSE)$
  2. $19\text{--}SECTION \leftarrow \sigma_{Year=19}(SECTION)$
  3. $CS19\text{--}SECTION \leftarrow \pi_{SectionIdentifier}(19\text{--}SECTION * CS\text{--}COURSE)$
  4. $STN\text{--}SID \leftarrow \pi_{StudentNumber,SectionIdentifier}(GRADE\text{--}REPORT)$
  5. $CS19\text{--}STUDENT(StudentNumber) \leftarrow STN\text{--}SID \div CS19\text{--}SECTION$
  6. $RESULTAT \leftarrow \pi_{Name}(CS19\text{--}STUDENT * STUDENT)$

Beispiel

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

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

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. $CS19{-}SECTION \leftarrow \pi_{SectionIdentifier}(19{-}SECTION * CS{-}COURSE)$

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

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. $CS19{-}SECTION \leftarrow \pi_{SectionIdentifier}(19{-}SECTION * CS{-}COURSE)$
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
  1. $CS{-}COURSE \leftarrow \sigma_{Department=CS}(COURSE)$
  2. $19{-}SECTION \leftarrow \sigma_{Year=19}(SECTION)$
  3. $CS19{-}SECTION \leftarrow \pi_{SectionIdentifier}(19{-}SECTION * CS{-}COURSE)$
  4. $STN{-}SID \leftarrow \pi_{StudentNumber,SectionIdentifier}(GRADE{-}REPORT)$
  5. $CS19{-}STUDENT(StudentNumber) \leftarrow STN{-}SID \div CS19{-}SECTION$
  6. $RESULTAT \leftarrow \pi_{Name}(CS19{-}STUDENT * STUDENT)$

CS19-STUDENT

StudentNumber
8

STUDENT

Name	StudentNumber	Class	Major
Smith	17	1	CS
Brown	8	2	CS

RESULTAT

Name
Brown

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 ($\cap$): $A - (A - B)$
  - Join ($\bowtie$): $\sigma_{\langle \dots \rangle}(A \times B)$

Minimalität der relationalen Algebra

- Argumentation für die Minimalität – kein Beweis
  - Die 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
  - Es 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 $-$ 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 durch die Operatoren $\{\sigma, \pi, -, \times\}$ dargestellt werden.

Aggregatfunktion und Gruppierung

Relationale Algebra

Aggregatfunktionen

- Aggregiert mehrere Tupel zu einem Tupel bzgl. eines Attributes A einer Relation R
  - $\mathcal{F}_{\langle Liste\ von\ (Funktion, Attribut\ A)\ Paaren \rangle}(R)$
  - Abbildung in den Wertebereich von A
- Standard-Aggregationsfunkionen:
  - $\mathcal{F}_{\text{MIN}\ A}(R)$ Minimaler Wert, den A in $r(R)$ annimmt
  - $\mathcal{F}_{\text{MAX}\ A}(R)$ Maximaler Wert, den A in $r(R)$ annimmt
  - $\mathcal{F}_{\text{AVG}\ A}(R)$ Durchschnittlicher Wert von A über alle Tupel in $r(R)$
  - $\mathcal{F}_{\text{SUM}\ A}(R)$ Summe der Werte von A über alle Tupel in $r(R)$
  - $\mathcal{F}_{\text{COUNT}\ A}(R)$ Anzahl der Tupel, bei denen $A \neq$ NULL in $r(R)$
    - $\mathcal{F}_{\text{COUNT}*}(R)$ 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

Aggregatfunktionen: Beispiele

- $RMIN(Min) \leftarrow \mathcal{F}_{\text{MIN } CreditHours}(COURSE)$
- $RMAX(Max) \leftarrow \mathcal{F}_{\text{MAX } CreditHours}(COURSE)$
- $RAVG(Avg) \leftarrow \mathcal{F}_{\text{AVG } CreditHours}(COURSE)$
- $RSUM(Sum) \leftarrow \mathcal{F}_{\text{SUM } CreditHours}(COURSE)$
- $RCNT(Cnt) \leftarrow \mathcal{F}_{\text{COUNT } CreditHours}(COURSE)$
  - $\mathcal{F}_{\text{COUNT}*}(COURSE)$?
    - Mit Projektion:
      Anzahl an Kursen vom Department CS
      $\mathcal{F}_{\text{COUNT}*}\left(\sigma_{Department=CS}(COURSE)\right)$
  - $\mathcal{F}_{\text{COUNT } Department}(COURSE)$?

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
3

RMAX

Max
4

RAVG

Avg
3.5

RSUM

Sum
14

RCNT

Cnt
4

Gruppierung

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

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

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

Gruppierung: Unterschiede

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

COUNT_CourseNumbers	AVG_CreditHours
4	3.5

Department	COUNT_CourseNumbers	AVG_CreditHours
CS	3	3.7
MATH	1	3

GAVG

Department	KursAnzahl	StundenSchnitt
CS	3	3.7
MATH	1	3

Einfügen, Löschen, Aktualisieren

Änderung von Relationenzuständen

Mengenorientierte Spezifizierung von Änderungsoperatoren

- Änderungen am Relationenzustand über Mengenoperationen realisierbar
  - Gegeben 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

Was kann schief gehen?

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

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

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. dom(CreditHours) = 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

Was kann schief gehen?

- **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

wird gelöscht

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

Fremdschlüssel

SECTION

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

kaskadierend löschen

auf NULL setzen

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

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

Was kann schief gehen?

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
- Entfernende Operatoren
  - Selektion σ, Projektion π
- Umbenennung ρ
- Kombinierende 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