MOBI-DBS-B: Datenbanksysteme Relationale Algebra

Vorlesung Sommersemester 2019

Tanya Braun, Universität zu Lübeck

Lehrauftrag SoSe 19, Universität Bamberg

Die Phasen des DB-Entwurfs

Informations-sammlung → **Semantische** Datenmodellier. → Logische Daten-modellierung → **Datenbank-** Installation

Anforderungs-**analyse** | **Grobe** Modellierung | Genaue **Modellierung** | **Konzeptuelles** Schema | *t*

- Interview
- Begriffsanalyse
- Brainstorming
- Dokumenten-analyse

…

- **ERM**
- NIAM
- EXPRESS-G
- IDEF1X
- STEP
- UML
- …

- hierarchisch
- netzwerk
- **relational**
- objekt-orientiert
- XML

…

- DB2
- Informix
- ORACLE
- Postgres
- mySQL
- …

← DBMS unabhängig | DBMS abhängig →

konzeptueller DB Schema Entwurf | **Logischer Schemaentwurf** | Physischer **Schemaentwurf**

Beispiel eines relationalen 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 von Relationenzuständen

STUDENT

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

COURSE

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

Das Relationale Datenmodell

Inhalte

- Relationales Datenmodell
  - DB = Sammlung von Relationen
  - Relationen
  - Relationale Datenbanken und -schemata
  - Referentielle Integrität
- Vom ER-Modell zum RM
- Relationale Algebra: Anfragen
  - Insert, delete, update
  - $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
  - Minimalität
  - Aggregieren, gruppieren

Kompetenzen

- ER-Modelle in ein relationales Schema überführen
- Mengenorientierte Verarbeitung verstehen und anwenden
- Unterschied deklarative und prozedurale Sprachen verstehen
- Publikationen aus der Forschung verstehen

Bezug zu Phasen des DB-Entwurfs

Relationale Algebra

Änderungen von Relationenzuständen und Verarbeitung von Anfragen

Notation (Wiederholung)

- $R(A_1, \ldots, A_n)$
  - Relationenschema n-ten Grades
- $t = <v_1, \ldots, v_n>$
  - 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
  - $t[A_u, \ldots, A_z]$ und $t.(A_u, \ldots, A_z)$ beziehen sich auf Werte $<v_u, \ldots, v_z>$ von Subtupeln von t, die den Attributen $A_u, \ldots, A_z$ von R entsprechen.

Attribut A

Relationenschema R

Relation r

Grad 4

Tupel t

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

Anfragen an Relationen

- Relationenzustände ändern
  - Einfügen, löschen, aktualisieren
- 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

Einfügen, Löschen, Aktualisieren

Änderung von Relationenzuständen

Einfügen von Tupeln

- INSERT INTO R(A_1, …, A_n) // oder: INSERT INTO R
  VALUES <v_1, …, v_n> :
  - Eingabe:
    eine Liste von Attributwerten für ein neues Tupel t = <v_1, …, v_n>, das in die Relation R bzw. r(R) eingefügt werden soll
  - 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

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 Mathmatics	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 Mathmatics	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)

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 Mathmatics	MATH2410	3	MATH
Databases	CS3380	3	CS

Löschen von Tupeln

- DELETE FROM R
  [WHERE <bedingung>]
  - Löscht eine Menge von Tupeln { t_j }$_j$ aus einer Relation r(R)
  - Spezifiziert über Bedingungen; Beispiele:
    - Bestimmtes Tupel über Wert v des Primärschlüssels A (löscht ein Tupel!)
    - DELETE FROM COURSE
      WHERE COURSE.CourseNumber = CS3380
    - Alle Tupel, bei denen z.B. ein Attribut B größer einem Wert w ist (löscht mehrere Tupel!)
    - DELETE FROM COURSE
      WHERE COURSE.CreditHours >= 4

  → Löscht ein oder mehrere Tupel (mengenorientierte Verarbeitung)
- Was kann nur schief gehen?

COURSE

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

wird gelöscht

wird gelöscht

Löschen von Tupeln: Fehlersituation

- DELETE
  - Löscht eine Menge von Tupeln $\{ t_j \}_j$ aus einer Relation r(R)
- 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_j
  - DELETE FROM COURSE
    WHERE COURSE.CourseNumber=CS3380
  - Ist überhaupt ein Kurs ohne Weiteres löschbar?

Primärschlüssel

COURSE

CourseName	CourseNumber
Introduction to CS	CS1310
Data Structures	CS3320
Discrete Mathmatics	MATH2410
Databases	CS3380

zu löschen

Fremdschlüssel

SECTION

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

Löschen von Tupeln: Fehlersituation

- Fehlersituation: Referenzielle Integrität
  - DELETE FROM COURSE
    WHERE COURSE.CourseNumber= CS3380
- 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

Primärschlüssel

COURSE

CourseName	CourseNumber
Introduction to CS	CS1310
Data Structures	CS3320
Discrete Mathmatics	MATH2410
Databases	CS3380

zu löschen

Fremdschlüssel

SECTION

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

Aktualisieren von Tupeln

- UPDATE R
  SET ...
  [WHERE <bedingung>]
  - Aktualisiert/ändert Tupel 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

  → Ändert ein oder mehrere Tupel (mengenorientierte Verarbeitung)

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathmatics	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 Mathmatics	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

Primärschlüssel

COURSE

CourseName	CourseNumber
Introduction to CS	CS1310
Data Structures	CS3320
Discrete Mathmatics	MATH2410
Databases	CS3380

Fremdschlüssel

SECTION

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

zu aktualisieren

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

Selektion und Projektion

Entfernende Operatoren*
Relationale Algebra

Selektion σ

- Bildet eine Teilmenge von Tupeln einer Relation, die (jeweils) eine bestimmte Auswahlbedingung erfüllen:
  - $R' = \sigma_{\text{<Auswahlbedingung>}}(R)$
  - Unär: wird auf genau eine Relation angewendet
  - Grad von R' = Grad von R
    - d.h., alle Attribute bleiben erhalten
- Beispiele:
  - $\sigma_{Department=CS}(COURSE)$

COURSE

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

Selektion σ

- Bildet eine Teilmenge von Tupeln einer Relation, die (jeweils) eine bestimmte Auswahlbedingung erfüllen:
  - $R' = \sigma_{<Auswahlbedingung>}(R)$
  - Unär: wird auf genau eine Relation angewendet
  - Grad von R' = Grad von R
    - d.h., alle Attribute bleiben erhalten
- Beispiele:
  - $\sigma_{Department=CS}(COURSE)$
  - $\sigma_{CreditHours>=4}(COURSE)$

COURSE

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

Selektion σ

- Bildet eine Teilmenge von Tupeln einer Relation, die (jeweils) eine bestimmte Auswahlbedingung erfüllen:
  - $R' = \sigma_{<Auswahlbedingung>}(R)$
  - Unär: wird auf genau eine Relation angewendet
  - Grad von R' = Grad von R
    - d.h., alle Attribute bleiben erhalten
- Beispiele:
  - $\sigma_{Department=CS}(COURSE)$
  - $\sigma_{CreditHours>=4}(COURSE)$
  - $\sigma_{(CreditHours<=3 \text{ AND } Department=CS) \text{ OR } (Department=MATH)}(COURSE)$

COURSE

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

Selektion σ

- $R' = \sigma_{\text{<Auswahlbedingung>}}(R)$
- Kommutativ:

$$\sigma_{\text{<Bedingung1>}}(\sigma_{\text{<Bedingung2>}}(R)) = \sigma_{\text{<Bedingung2>}}(\sigma_{\text{<Bedingung1>}}(R))$$

  - $\sigma_{(\text{CreditHours<=3})}(\sigma_{(\text{Department=CS})}(\text{COURSE})) = \sigma_{(\text{Department=CS})}(\sigma_{(\text{CreditHours<=3})}(\text{COURSE}))$

    3 Tupel nach erster σ 2 Tupel nach erster σ
  - Reihenfolge hat eine Auswirkung auf Größe des Zwischenergebnisses!
- Kaskade von σ = AND-Verknüpfung der Bedingungen:

$$\sigma_{\text{<Bedingung1>}}(\sigma_{\text{<Bedingung2>}}(\ldots(\sigma_{\text{<Bedingung}n\text{>}}(R))\ldots)) =$$
$$\sigma_{\text{<Bedingung1> AND <Bedingung2> AND … AND <Bedingung}n\text{>}}(R)$$

  - $\sigma_{\text{CreditHours<=3}}(\sigma_{(\text{Department=CS})}(\text{COURSE})) = \sigma_{\text{CreditHours<=3 AND Department=CS}}(\text{COURSE})$

COURSE

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

Projektion π

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

COURSE

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

Projektion π

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

COURSE

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

Projektion π

- Wählt aus einer Relation bestimmte Attribute aus und verwirft die anderen:
  - $R' = \pi_{\text{<Attributliste>}}(R)$
  - Unär: wird auf genau eine Relation angewendet
  - Grad R' ≤ Grad R
    - idR fehlen nach π Attribute
- Beispiele:
  - $\pi_{\text{CourseName, CourseNumber}}(\text{COURSE})$
  - $\pi_{\text{CourseName}}(\text{COURSE})$
- Es gilt:
  - Wenn <Liste1> $\subseteq$ <Liste2>: $\pi_{\text{<Liste1>}}(\pi_{\text{<Liste2>}}(R)) = \pi_{\text{<Liste1>}}(R)$
    - $\pi_{\text{CourseName}}(\pi_{\text{CourseName, CourseNumber}}(\text{COURSE})) = \pi_{\text{CourseName}}(\text{COURSE})$

COURSE

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

Projektion π

- $R' = \pi_{<\text{Attributliste}>}(R)$
- Anzahl der Tupel kann sich verringern
  - Mengeneigenschaft entfernt Duplikate
- Beispiele:
  - $\pi_{\text{CreditHours, Department}}(\text{COURSE})$
  - Projektionen der vorherigen Folie
    - Keine doppelten Einträge
    - CourseName, CourseNumber eindeutig (Primärschlüssel bzw. Schlüsselkandidaten involviert)

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 Mathmatics	MATH2410	3	MATH
Databases	CS3380	3	CS

Operationssequenzen und Renaming

Relationale Algebra

Sequenzen von Operationen

- Im Allg. werden mehrere Operationen nacheinander ausgeführt
  - Einzelner Ausdruck oder Sequenz mit explizit benanntem Zwischenergebnis
- Beispiel
  - $\pi_{\text{CreditHours, Department}}(\sigma_{\text{CreditHours>=4}}(\text{COURSE}))$
  - $\text{MIN\_4} \leftarrow \sigma_{\text{CreditHours>=4}}(\text{COURSE})$
    $\text{RESULTAT} \leftarrow \pi_{\text{CreditHours,Department}}(\text{MIN\_4})$
- Relationen umbenennen möglich:
  - $\text{MIN\_4} \leftarrow \sigma_{\text{CreditHours>=4}}(\text{COURSE})$
    $\text{COURSE4(Hours, Department)} \leftarrow$
    $\pi_{\text{CreditHours, Department}}(\text{MIN\_4})$

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 Mathmatics	MATH2410	3	MATH
Databases	CS3380	3	CS

Umbenennung ρ

- Erlaubt die explizite Umbenennung von Relationen und Attributen
  - Verändert nicht Tupel, sondern Schemata
  - Auch bekannt als RENAME
- Gegeben Ausgangsrelation $R(A_1, \ldots, A_n)$:
  - Umbenennung von R in S und $A_1, \ldots, A_n$ in $B_1, \ldots, B_n$
    - $\rho_{S(B_1, \ldots, B_n)}(R)$
    - $\rho_{KURS(KursName,\ KursNummer,\ SWS,\ Dekanat)}(COURSE)$

KURS

KursName	KursNummer	SWS	Dekanat
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathmatics	MATH2410	3	MATH
Databases	CS3380	3	CS

COURSE

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

Umbenennung ρ

- Erlaubt die explizite Umbenennung von Relationen und Attributen
  - Verändert nicht Tupel, sondern Schemata
  - Auch bekannt als RENAME
- Gegeben Ausgangsrelation $R(A_1, \ldots, A_n)$:
  - Umbenennung von R in S
    - $\rho_S(R)$
  - Umbenennung von $A_1, \ldots, A_n$ in $B_1, \ldots, B_n$
    - $\rho_{(B_1, \ldots, B_n)}(R)$
- Häufig Hilfsoperation in Operationssequenzen
  - Mengenoperationen, Join, ... (mehr dazu gleich)

COURSE

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

Vereinigung, Schnitt, Differenz

Mengenoperatoren
Relationale Algebra

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

Vereinigung: Beispiel

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

RESULT

CreditHours	Department
3	MATH
3	CS

$\cup$

RESULT1

CreditHours	Department
3	MATH

RESULT2

CreditHours	Department
3	MATH
3	CS

COURSE

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

Schnitt: Beispiel

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

RESULT

CreditHours	Department
3	MATH

$\cap$

RESULT1

CreditHours	Department
3	MATH

RESULT2

CreditHours	Department
3	MATH
3	CS

COURSE

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

Differenz: Beispiel

- RESULT1 $\leftarrow \pi_{\text{CreditHours, Department}}(\sigma_{\text{Department=MATH}}(\text{COURSE}))$
- RESULT2 $\leftarrow \pi_{\text{CreditHours, Department}}(\sigma_{\text{CreditHours<=3}}(\text{COURSE}))$
- RESULT $\leftarrow$ RESULT2 – RESULT1
  - RESULT $\leftarrow$ RESULT1 – RESULT2? (Tupelmenge leer!)

RESULT

CreditHours	Department
3	CS

–

RESULT1

CreditHours	Department
3	MATH

RESULT2

CreditHours	Department
3	MATH
3	CS

COURSE

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

Mengenoperatoren: Eigenschaften

- Können nur auf UNION-kompatible Relationen angewendet werden:
  - R und S haben gleichen Grad
  - Attribute haben gleiche Wertebereiche
    - $dom(A_i) = dom(B_i)$ für alle $1 \leq i \leq n$
  - Attribute müssen aber nicht gleich heißen → 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 Allg. nicht kommutativ: $R - S \neq S - R$
  - Im Allg. 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") miteinander verbunden
  - Gegeben $R(A_1, \ldots, A_n)$ und $S(B_1, \ldots, B_m)$
  - $R \times S = Q(A_1, \ldots, A_n, B_1, \ldots, B_m)$
  - Binär
  - R und S müssen nicht UNION-kompatibel sein
- Resultat
  - Keine Duplikate
  - R hat n Spalten, S hat m Spalten → $R \times S$ hat (n+m) Spalten
  - R hat k Zeilen, S hat l Zeilen → $R \times S$ hat $(k \cdot l)$ Zeilen
  - Um eindeutige Attributbezeichnungen in der Ergebnisrelation zu gewährleisten, müssen Attribute, die in R und S gleich bezeichnet sind, vor der Bildung des kartesischen Produkts umbenannt werden

Kartesisches Produkt: Beispiel

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

COURSE

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

RES1

CourseNumber
CS1310
CS3320
CS3380

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

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

- RES1 ← $\pi_{\text{CourseNumber}}(\sigma_{\text{Department=CS}}(\text{COURSE}))$
- RES2 ← $\rho_{\text{Course, Instructor}}(\pi_{\text{CourseNumber, Instructor}}(\sigma_{\text{Year=19}}(\text{SECTION})))$
- CROSS ← RES1 × RES2
- RESULT ← $\pi_{\text{CourseNumber, Instructor}}(\sigma_{\text{CourseNumber=Course}}(\text{CROSS}))$

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?
Wir wissen ja:
CourseNumber=Course

Join ⋈: Intuition

- RES1 ← $\pi_{CourseNumber>}(\sigma_{Department=CS}(COURSE))$
- RES2 ← $\rho_{Course,\ Instructor}(\pi_{CourseNumber,\ Instructor}(\ SECTION))$
- 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~~	~~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 ⋈: Intuition

- RES1 ← $\pi_{CourseNumber>}(\sigma_{Department=CS}(COURSE))$
- RES2 ← $\rho_{Course,\ Instructor}(\pi_{CourseNumber,\ Instructor}(\ SECTION))$
- Die Tupel miteinander verbinden, wo
  RES1.CourseNumber = RES2.Course

$RES1 \bowtie_{CourseNumber=Course} RES2$

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

Join ⋈

- Verbindet die Tupel zweier Relationen, die Join-Bedingung erfüllen
  - Gegeben $R(A_1, \ldots, A_n)$ und $S(B_1, \ldots, B_m)$
  - $R \bowtie_{\text{<Bedingung>}} S = Q(A_1, \ldots, A_n, B_1, \ldots, B_m)$
  - Q enthält alle Kombinationen von Tupeln, die der Bedingung entsprechen
- Beispiel
  - $RES1 \leftarrow \pi_{CourseNumber}(\sigma_{Department=CS}(COURSE))$
  - $RES2 \leftarrow \rho_{Course,\ Instructor}(\pi_{CourseNumber,\ Instructor}(SECTION))$
  - $JOINRES \leftarrow RES1 \bowtie_{CourseNumber=Course} RES2$
  - $RES \leftarrow \pi_{CourseNumber,\ Instructor}(JOINRES)$
- Äquivalent zu kartesischem Produkt mit anschließender Selektion
  - $R \bowtie_{\text{<Bedingung>}} S = \sigma_{\text{<Bedingung>}} (R \times S)$
  - Vgl. Beispiel zum kartesischen Produkt

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 θ B_j der Join-Bedingung
  - basiert auf einem θ aus {=, <, ≤, ≥, >, ≠}
- Equi-Join (Spezialfall)
  - θ ist "=" : es gibt nur eine Join-Bedingung, und sie prüft auf Gleichheit
- Beispiel von vorher: Equi-Join
  - RES1 ← $\pi_{\text{CourseNumber}}(\sigma_{\text{Department=CS}}(\text{COURSE}))$
  - RES2 ← $\rho_{\text{Course, Instructor}}(\pi_{\text{CourseNumber, Instructor}}(\text{ SECTION}))$
  - JOINRES ← $\text{RES1} \bowtie_{\text{CourseNumber=Course}} \text{RES2}$
  - RES ← $\pi_{\text{CourseNumber, Instructor}}(\text{JOINRES})$

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 $\leftarrow \pi_{\text{CourseNumber}}(\sigma_{\text{Department=CS}}(\text{COURSE}))$
  - RES2 $\leftarrow \cancel{\rho_{\text{Course, Instructor}}}(\pi_{\text{CourseNumber, Instructor}}(\text{ SECTION}))$
  - JOINRES $\leftarrow$ RES1 * RES2
  - ~~RES~~ $\cancel{\leftarrow \pi_{\text{CourseNumber, Instructor}}(\text{JOINRES})}$

RES1

CourseNumber
CS1310
CS3320
CS3380

RES2

CourseNumber	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
  - Left Outer Join: alle Tupel von R
  - Right Outer Join: alle Tupel von S
  - Full Outer Join: alle Tupel von R, S

- Beispiele:
  - LEFT ← RESN ⟕$_{\text{CourseNumber=Course}}$RES2
  - RIGHT ← RESN ⟖$_{\text{CourseNumber=Course}}$RES2
  - FULL ← RESN ⟗$_{\text{CourseNumber=Course}}$RES2

RESN

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
- NULL-Werte für Datenfelder, die dadurch für ein Tupel neu entstehen
- Beispiel:
  - ALL_COURSES $\leftarrow$ COURSE $\uplus$ $\rho_{\text{CourseName,CreditHours,Department,Module}}$(SEMINAR)

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathmatics	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 $\uplus$

- Vereinigung von Tupeln, deren Relationen nicht UNION-kompatibel bzw. nur partiell UNION-kompatibel sind
- NULL-Werte für Datenfelder, die dadurch für ein Tupel neu entstehen
- Beispiel:
  - ALL_COURSES $\leftarrow$ COURSE $\uplus$ $\rho_{CourseName,CreditHours,Department,Module}$(SEMINAR)

ALL_COURSES

CourseName	CourseNumber	CreditHours	Department	Module
Introduction to CS	CS1310	4	CS	NULL
Data Structures	CS3320	4	CS	NULL
Discrete Mathmatics	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 $X \subseteq Z$
  - Binäre Operation
  - Nicht sehr intuitiv, deswegen selten verwendet
- Sei $Y = Z - X$
  - Y ist die Menge der Attribute von R, die keine Attribute in S sind
- T(Y) enthält ein Tupel t, wenn ein Tupel t_R in R existiert, so dass für jedes Tupel t_S in S: $t_R[Y]=t$ 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
  - CS_COURSE ← $\sigma_{Department=CS}$(COURSE)

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathmatics	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

- 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)

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

- Ermittle alle Namen von Studenten, die in allen Kursen, die das Department CS anbietet, im Jahr 19 Prüfungen abgelegt haben
  - CS_COURSE ← $\sigma_{\text{Department=CS}}$(COURSE)
  - 19_SECTION ← $\sigma_{\text{Year=19}}$(SECTION)
  - CS19_SECTION ← $\pi_{\text{SectionIdentifier}}$(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

- Ermittle alle Namen von Studenten, die in allen Kursen, die das Department CS anbietet, im Jahr 19 Prüfungen abgelegt haben
  - CS_COURSE ← $\sigma_{Department=CS}$(COURSE)
  - 19_SECTION ← $\sigma_{Year=19}$(SECTION)
  - CS19_SECTION ← $\pi_{SectionIdentifier}$(19_SECTION * CS_COURSE)
  - STN_SID ← $\pi_{StudentNumber,\ SectionIdentifier}$(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

- Ermittle alle Namen von Studenten, die in allen Kursen, die das Department CS anbietet, im Jahr 19 Prüfungen abgelegt haben
  - CS_COURSE ← $\sigma_{Department=CS}$(COURSE)
  - 19_SECTION ← $\sigma_{Year=19}$(SECTION)
  - CS19_SECTION ← $\pi_{SectionIdentifier}$(19_SECTION * CS_COURSE)
  - STN_SID ← $\pi_{StudentNumber,\ SectionIdentifier}$(GRADE_REPORT)
  - CS19_STUDENT(StudentNumber) ← STN_SID ÷ CS19_SECTION

CS19_SECTION

SectionIdentifier
102
119
135

STN_SID

StudentNumber	SectionIdentifier
17	112
17	119
8	85
8	119
8	102
8	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 ← $\sigma_{Department=CS}$(COURSE)
  - 19_SECTION ← $\sigma_{Year=19}$(SECTION)
  - CS19_SECTION ← $\pi_{SectionIdentifier}$(19_SECTION * CS_COURSE)
  - STN_SID ← $\pi_{StudentNumber,\ SectionIdentifier}$(GRADE_REPORT)
  - CS19_STUDENT(StudentNumber) ← STN_SID ÷ CS19_SECTION
  - RESULTAT ← π_{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 ($\bowtie$): $\sigma_{<\ldots>}(A \times B)$
- 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
    Daher wird auf ρ im Folgenden nicht weiter eingegangen
  - Es verbleiben also die Operatoren σ, π, $\cup$, –, x bzgl. der Untersuchung, ob man welche davon ohne Verlust der Ausdrucksmöglichkeiten streichen kann

Untersuchung der verbliebenen Operatoren

- Kartesisches Produkt (×):
  - Kann nicht simuliert werden, da σ, π, ∪ und – ein Schema nicht erweitern können.
- Projektion (π):
  - Analoges Argument: keiner der Operatoren σ, ∪, – und × 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 (∪)
  - Kann nicht durch durch die Operatoren σ, π, x und – dargestellt werden.

Aggregatfunktion und Gruppierung

Relationale Algebra

Aggregatsfunktionen

- Aggregiert mehrere Tupel zu einem bzgl. eines Attributes A
  - $\mathcal{F}_{\text{<Liste von <Funktion Attribut A> Paaren>}}(R)$
  - A Attribut von 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 ≠ NULL in r(R)
  - Setzt voraus, dass im Wertebereich des aggregierten Attributs eine Ordnung (bei MIN, MAX) bzw. Rechenoperationen (bei AVG, SUM) definiert sind

Aggregatsfunktionen: Beispiele

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

COURSE

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

RMIN

Min
3

RMAX

Max
4

RAVG

Avg
3.5

RSUM

Sum
14

RCNT

Cnt
4

Aggregatsfunktionen: Beispiele

- RMIN(Min) ← $\mathcal{F}_{\text{MIN CreditHours}}$ (COURSE)
- RMAX(Max) ← $\mathcal{F}_{\text{MAX CreditHours}}$ (COURSE)
- RAVG(Avg) ← $\mathcal{F}_{\text{AVG CreditHours}}$ (COURSE)
- RSUM(Sum) ← $\mathcal{F}_{\text{SUM CreditHours}}$ (COURSE)
- RCNT(Cnt) ← $\mathcal{F}_{\text{COUNT CreditHours}}$ (COURSE)
  - $\mathcal{F}_{\text{COUNT Department}}$ (COURSE) = 4 (!)
  - $\mathcal{F}_{\text{COUNT *}}$ (COURSE) ohne Attribut (*) = Kardinalität von COURSE = 5
  - Häufig im Anschluss an eine Selektion: Anzahl an Kursen vom Department CS

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathmatics	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
  - $<B_1, \ldots, B_m>\ \mathcal{F}_{\text{<Liste von <Funktion Attribut> Paaren>}}(R)$
- Beispiel:
  - Bestimme die durchschnittliche Stundenzahl der Kurse der Departments
    $GAVG(Department, Cnt, Avg) \leftarrow$
    $Department\ \mathcal{F}_{\text{COUNT CourseNumber, AVG CreditHours}}(COURSE)$

COURSE

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

GAVG

Department	Cnt	Avg
CS	3	3.7
MATH	1	3

Unterschiede

- Beispiel:
  - $\mathcal{F}_{\text{COUNT CourseNumbers, AVG CreditHours}}(\text{COURSE})$
  - Department $\mathcal{F}_{\text{COUNT CourseNumbers, AVG CreditHours}}(\text{COURSE})$
  - GAVG(Department, KursAnzahl, StundenSchnitt) $\leftarrow$
    Department $\mathcal{F}_{\text{COUNT CourseNumbers, AVG CreditHours}}(\text{COURSE})$

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

Rückblick

- Relationale Algebra als Anfragesprache an Relationen
- Relationenzustände ändern
  - Einfügen, löschen, aktualisieren
- Entfernende Operatoren
  - Selektion σ, Projektion π
- Umbenennung ρ
- Klassische Mengenoperatoren (kombinieren Relationen)
  - Vereinigung $\cup$, Schnitt $\cap$, Differenz $-$
- Kombinierende Operatoren
  - Kartesisches Produkt $\times$, Join $\bowtie$ und weitere Join-Arten, Outer Union, Division
- Minimalität der relationalen Algebra
- Aggregieren, gruppieren