4. Relationale Anfragesprachen

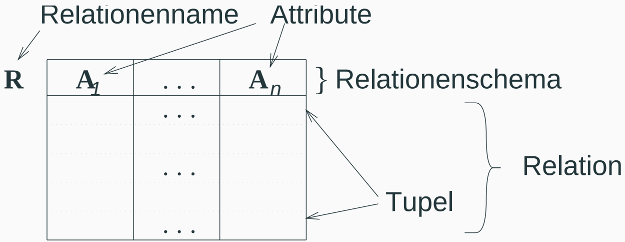
Einordnung in den Vorlesungsverlauf

- ER-Modell
- Relationenmodell
- **relationale Anfragesprachen**
- SQL
- Entwurfstheorie
- Transaktionen

Sprachen, mit denen man Informationen aus einer Datenbank im Relationenmodell extrahieren kann

Wiederholung Relationenmodell

- $R \subset REL(X)$ ist **Relation** über einem Relationenschema X.
- Das **Relationenschema** $X = \{A_1, \ldots, A_n\}$ ist Menge von Attributen mit bestimmten Wertebereichen (genauer $X = \{A_1 : D_1, \ldots, A_n : D_n\}$)
- Notation: $R(A_1, \ldots, A_n)$



Für ein Attribut A in der Relation R ist $R.A$ der **qualifizierte** Attributname.

Voraussetzungen für dieses Kapitel

Für die Relationen gelten folgende Bedingungen:

- keine `null`-Werte
- Eine Relation ist eine Menge von Tupel, bei der es keine Duplikate gibt.
- Im Relationenschema sind die Namen der Attribute relevant, nicht ihre Reihenfolge.
- Die Relation befindet sich in 1NF

Eine Relation ist in **erster Normalform** (1NF), wenn die Wertebereiche aller Attribute *atomar* sind. Das bedeutet, dass keine zusammengesetzten, geschachtelten oder mengenwertigen Einträge erlaubt sind.

Anfragen

Studi(MatrNr, Vorname, Nachname, Fach, Semester)   hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von)   Prof(ID, Vorname, Nachname, Lehrstuhl)

Beispiele für Anfragen

- Geben Sie alle Studis aus, die im ersten Semester sind
- Geben Sie den Titel alle Vorlesungen aus, die von Prof XY gehalten werden
- Geben Sie den Namen aller Profs aus, die *ausschließlich* Vorlesungen mit 5 CP halten.
- Geben Sie die Studis aus, die *alle* Vorlesungen von Prof XY gehört haben.

Ziel: Solche Anfragen mit Hilfe von relationalen Anfragesprachen formulieren.

Relationale Anfragesprachen

Anfrage: Folge von Operationen, die aus den Basisrelationen eine Ergebnisrelation berechnet

- interaktiv auf Bildschirm angezeigt oder per Programm weiterverarbeitet
- ggf. unter anderem Namen abgespeichert (Sichtrelation/Snapshot)

2 klassische Sprachen: **Relationenalgebra** und Relationenkalkül (Tupelkalkül, Domänenkalkül)

Eigenschaften

- Vollständigkeit: Eine Sprache heißt *relational vollständig*, wenn sie mindestens so mächtig ist wie die Relationenalgebra bzw. das Relationenkalkül.
- Abgeschlossenheit: Das Ergebnis einer Anfrage ist wieder eine Relation.

Exkurs: Algebren in der Mathematik

Eine **Algebra** besteht aus einer Menge A mit einer Familie von Operationen $(f_i)_{i \in I}$.

- Beispiel 1: Natürliche Zahlen mit Addition und Multiplikation
- Beispiel 2: Vektoren mit Addition, skalarer Multiplikation und Vektorkreuzprodukt
- Operationen erfüllen bestimmte Rechenregeln
  (z.B. Assoziativgesetz, Kommutativgesetz, ...)

Bei der **Relationenalgebra**: Relationen mit Operationen auf Relationen.

Die **minimale** Relationenalgebra besteht aus der kleinsten Menge von Operatoren, sodass die Vollständigkeit erhalten bleibt.

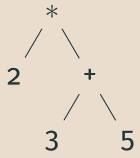
- Die Operatoren $\Omega = \{\pi, \sigma, \cup, -, \times, \rho\}$ bilden eine minimale Relationenalgebra

Ein **Ausdruck der Relationenalgebra** besteht i.A. aus einer Hintereinander-Ausführung/Verkettung von Operationen auf Relationen.

- Die Relationenalgebra ist eine *prozeduale* Sprache, also die Reihenfolge der Operationen ist relevant.
- Für Veranschaulichung von komplexen Anfragen: Operatorbäume

Exkurs: Operatorbaum

$2 * (3 + 5)$ entspricht



Operatorbaum Relationenalgebra

$\pi_A(R \bowtie S)$ entspricht

Überblick

1. Operatoren der minimalen Relationenalgebra
    - Projektion π
    - Selektion σ
    - Vereinigung $\cup$
    - Differenz $-$
    - Kreuzprodukt $\times$
    - Umbenennung ρ
2. Weitere Operatoren
    - Schnitt $\cap$
    - Joins $\bowtie$
    - Division $\div$
3. Erweiterungen der Relationenalgebra

Projektion

Sei $R \subset REL(X)$ eine Relation über dem Schema X.
Sei Y eine Teilmenge von X.

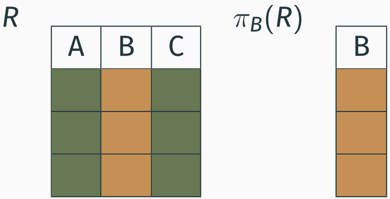
Die **Projektion** $\pi_Y(R)$ gibt nur die Spalten der Relation R zurück, die in der Projektionsliste $Y \subseteq X$ enthalten sind.

Syntax: $\pi_{<Attributmenge>}(< Relation >)$

Projektion formal

$$\pi_Y(R) = \{t[Y] \mid t \in R\}$$

wobei $t[Y]$ die Einschränkung eines Tupels t auf die Attribute in Y ist.



Projektion auf ein Attribut

$$\pi_A\left(\begin{array}{|c|c|c|}\hline A & B & C \\ \hline 1 & a & 7 \\ 2 & b & 6 \\ 1 & c & 7 \\ \hline\end{array}\right) = \begin{array}{|c|}\hline A \\ \hline 1 \\ 2 \\ \hline\end{array}$$

⚠ Duplikate werden entfernt!

Projektion auf eine Attributmenge

$$\pi_{A,B}\left(\begin{array}{|c|c|c|}\hline A & B & C \\ \hline 1 & a & 7 \\ 2 & b & 6 \\ 1 & c & 7 \\ \hline\end{array}\right) = \begin{array}{|c|c|}\hline A & B \\ \hline 1 & a \\ 2 & b \\ 1 & c \\ \hline\end{array}$$

- mehrere Attribute mit Komma getrennt
- keine Mengenklammern

Projektion Anfragen

Studi(MatrNr, Vorname, Nachname, Fach, Semester)   hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von)   Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Titel aller Vorlesungen aus.

$$\pi_{Titel}(Vorlesung)$$

Anfrage: Geben Sie die Matrikelnummern und die vollständigen Namen aller Studierenden aus.

$$\pi_{MatrNr,Vorname,Nachname}(Studi)$$

Sei R eine Relation und F ein Selektionsprädikat.

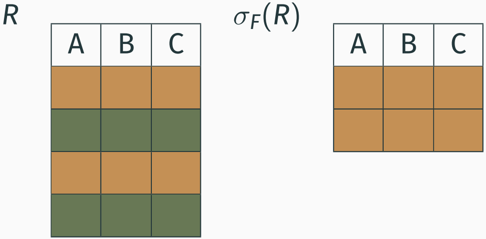
Die **Selektion** $\sigma_F(R)$ gibt alle Tupel der Relation R zurück, welche die Bedingung der Formel F erfüllen.

Syntax: $\sigma_{<Bedingung>}(< Relation >)$

Prädikat: logische Formel, die entweder $\mathtt{true}$ oder $\mathtt{false}$ ausgibt

Selektion formal

$$\sigma_F(R) = \{t \in R \mid F(t) = \mathtt{true}\}.$$



Welche Bedingungen sind erlaubt?

Bei einer Selektion $\sigma_F(R)$ wird die Formel F **Selektionsprädikat** genannt und ist aufgebaut aus:

- Attributen der Relation R oder Konstanten als Operanden
- arithmetische Vergleichoperatoren: $=, <, \leq, \geq, >, \neq$
- logische Operatoren: $\wedge$ (und), $\vee$ (oder), $\neg$ (nicht)

- größer/kleiner-Vergleiche funktionieren für alle Datentypen mit einer Ordnung
- bei Strings wird lexikographische Ordnung verwendet

Selektion

Konstanten-Selektion

$$\sigma_{C>2}\left(\begin{array}{|c|c|c|}\hline A & B & C \\ \hline 1 & a & 1 \\ 2 & b & 5 \\ 1 & c & 3 \\ \hline \end{array}\right) = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline 2 & b & 5 \\ 1 & c & 3 \\ \hline \end{array}$$

$$\sigma_{B=\text{'b'}}\left(\begin{array}{|c|c|c|}\hline A & B & C \\ \hline 1 & a & 1 \\ 2 & b & 5 \\ 1 & c & 3 \\ \hline \end{array}\right) = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline 2 & b & 5 \\ \hline \end{array}$$

⚠ Bei Zeichenketten an Anführungszeichen denken!

Selektion

Attribut-Selektion

$$\sigma_{C>A}\left(\begin{array}{|c|c|c|}\hline A & B & C \\ \hline 1 & a & 1 \\ 2 & b & 5 \\ 1 & c & 3 \\ \hline \end{array}\right) = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline 2 & b & 5 \\ 1 & c & 3 \\ \hline \end{array}$$

Verknüpfte Selektion

$$\sigma_{C>A \wedge B=\text{'b'}}\left(\begin{array}{|c|c|c|}\hline A & B & C \\ \hline 1 & a & 1 \\ 2 & b & 5 \\ 1 & c & 3 \\ \hline \end{array}\right) = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline 2 & b & 5 \\ \hline \end{array}$$

Selektion Anfragen

Studi(<u>MatrNr</u>, Vorname, Nachname, Fach, Semester) hört(MatrNr, VID)
Vorlesung(<u>ID</u>, Titel, CP, gehalten_von) Prof(<u>ID</u>, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie alle Studis aus, die mindestens im dritten Semester sind.

$\sigma_{Semester \geq 3}(Studi)$

Anfrage: Geben Sie alle Studis mit Nachnamen ab dem Anfangsbuchstaben K aus.

$\sigma_{Nachname \geq 'K'}(Studi)$

Anmerkungen:

- Bei Strings an Anführungszeichen denken
- Strings haben lexikographische Ordnung

Studi(<u>MatrNr</u>, Vorname, Nachname, Fach, Semester) hört(<u>MatrNr</u>, <u>VID</u>)
Vorlesung(<u>ID</u>, Titel, CP, gehalten_von) Prof(<u>ID</u>, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie alle BWL-Studis im ersten Semester aus.

$$\sigma_{Fach='\text{BWL}' \wedge Semester=1}(Studi)$$
$$= \sigma_{Fach='\text{BWL}'}(\sigma_{Semester=1}(Studi))$$
$$= \sigma_{Semester=1}(\sigma_{Fach='\text{BWL}'}(Studi))$$

Rechenregeln

- $\sigma_{F_1 \wedge F_2}(R) = \sigma_{F_1}(\sigma_{F_2}(R))$
- $\sigma_{F_2}(\sigma_{F_1}(R)) = \sigma_{F_1}(\sigma_{F_2}(R))$

Rechenregeln - Selektion/Projektion

Studi($\underline{\text{MatrNr}}$, Vorname, Nachname, Fach, Semester) hört($\overline{\underline{\text{MatrNr}, \text{VID}}}$)
Vorlesung($\underline{\text{ID}}$, Titel, CP, $\overline{\text{gehalten\_von}}$) Prof($\underline{\text{ID}}$, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie den Titel alle Vorlesungen aus, die mindestens 5 CP bringen.

$$\pi_{Titel}(\sigma_{CP\geq 5}(Vorlesung))$$

Anfrage: Geben Sie den Titel und die CP-Anzahl von allen Vorlesungen aus, die mindestens 5 CP bringen.

$$\pi_{Titel,CP}(\sigma_{CP\geq 5}(Vorlesung)) = \sigma_{CP\geq 5}(\pi_{Titel,CP}(Vorlesung))$$

Rechenregel

Wenn F *ausschließlich* Attribute aus X enthält, dann gilt: $\pi_X(\sigma_F(R)) = \sigma_F(\pi_X(R))$

Mengenoperationen

⚠ Für Mengenoperationen müssen die beiden Relationen das gleiche Schema haben!

- Die Namen der Attribute müssen gleich sein
- Wertebereich / Domäne der jeweiligen Attribute müssen gleich sein
- Die Reihenfolge spielt für uns keine Rolle.
- Ggf. sind Projektionen oder Umbenennungen (später im Kapitel → Slide 28) nötig, um das gleiche Schema zu erhalten.

Mengenoperationen

R

A	B
1	a
2	b
1	c

S

A	B
1	a
3	d
2	e

⚠ Keine Duplikate!

$R \cup S$

A	B
1	a
2	b
1	c
3	d
2	e

$R - S$

A	B
2	b
1	c

$R \cap S$

A	B
1	a

Vereinigung

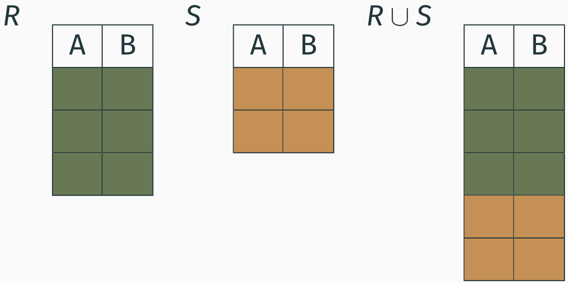
Seien $R \subset REL(X)$ und $S \subset REL(X)$ zwei Relationen über demselben Schema X.

Die **Vereinigung** $R \cup S$ gibt die Tupel zurück, die in der Relation R oder in der Relation S enthalten sind.

Syntax: $< \mathit{Relation1} > \cup < \mathit{Relation2} >$

Vereinigung formal

$$R \cup S = \{t \mid t \in R \vee t \in S\}.$$



Studi(MatrNr, Vorname, Nachname, Fach, Semester) hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von) Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Nachnamen aller Studis und aller Profs aus.

$$\pi_{Nachname}(Studi) \cup \pi_{Nachname}(Prof)$$

Anfrage: Geben Sie alle Studis aus, die entweder Informatik oder Mathematik studieren.

$$\sigma_{Fach=\text{'Informatik'}}(Studi) \cup \sigma_{Fach=\text{'Mathematik'}}(Studi)$$
$$= \sigma_{Fach=\text{'Informatik'} \vee Fach=\text{'Mathematik'}}(Studi)$$

Rechenregel

$$\sigma_{F_1}(R) \cup \sigma_{F_2}(R) = \sigma_{F_1 \vee F_2}(R)$$

Differenz

Seien $R \subset REL(X)$ und $S \subset REL(X)$ zwei Relationen über demselben Schema X.

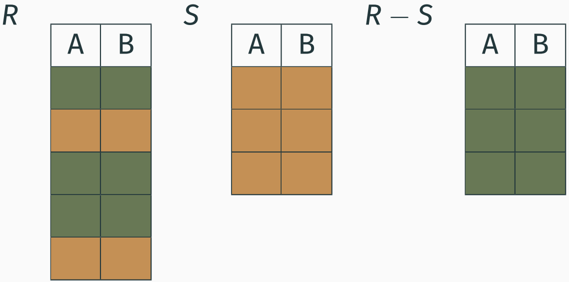
Die **Differenz** $R - S$ gibt die Tupel zurück, die in der Relation R, aber nicht in der Relation S enthalten sind.

Syntax: $< Relation1 > - < Relation2 >$

Differenz formal

$$R - S = \{t \in R \mid t \notin S\}.$$

- Einzige *nicht-monotone* Operation in der minimalen Relationenalgebra.



Differenz Anfragen

Studi(<u>MatrNr</u>, Vorname, Nachname, Fach, Semester)   hört($\overline{\text{MatrNr}}$, $\overline{\text{VID}}$)
Vorlesung(<u>ID</u>, Titel, CP, $\overline{\text{gehalten\_von}}$)   Prof(<u>ID</u>, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die MatrNr aller Studis aus, die *keine* Vorlesung hören.

$$\pi_{MatrNr}(Studi) - \pi_{MatrNr}(hört)$$

Kreuzprodukt

Seien $R \subset REL(X)$ und $S \subset REL(Y)$ zwei Relationen.

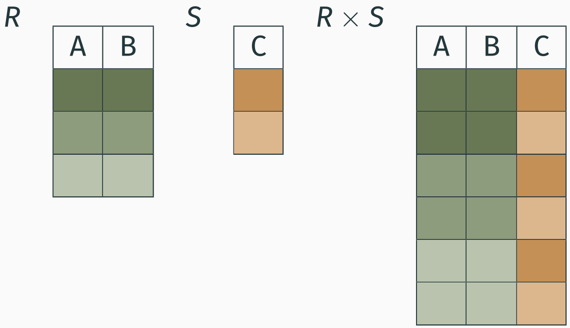
Das **Kreuzprodukt** $R \times S$ kombiniert alle Tupel von R mit den Tupeln von S.

Syntax: <Relation1> $\times$ <Relation2>

Kreuzprodukt formal

$$R \times S = \{r \circ s \mid r \in R, s \in S\}.$$

- Mit $\circ$ wird die *Konkatenation* von zwei Tupeln bezeichnet.
- Beispiel: $(a_1, a_2) \circ (b_1, b_2, b_3) = (a_1, a_2, b_1, b_2, b_3)$.
- $R \times S = REL(X \cup Y)$.



Kreuzprodukt

- $R \times S = REL(X \cup Y)$.
- R hat n Attribute und S hat m Attribute
  $\Rightarrow R \times S$ hat $n + m$ Attribute
- R hat k Tupel und S hat l Tupel
  $\Rightarrow R \times S$ hat $k \cdot l$ Tupel

Was, wenn R und S Attribute mit demselben Namen haben?

- ggf. *qualifizierte* Attributnamen verwenden
- alternativ Umbenennung verwenden
- Umbenennung nötig, falls beide Relationen gleich heißen

R

A	B
1	a
2	b

S

C	D
a	15
c	23

$R \times S$

A	B	C	D
1	a	a	15
1	a	c	23
2	b	a	15
2	b	c	23

R

A	B

S

A	C

$R \times S$

R.A	R.B	S.A	S.C

Kreuzprodukt Anfragen

Studi($\underline{\text{MatrNr}}$, Vorname, Nachname, Fach, Semester)   hört($\overline{\underline{\text{MatrNr}, \text{VID}}}$)
Vorlesung($\underline{\text{ID}}$, Titel, CP, $\overline{\text{gehalten\_von}}$)   Prof($\underline{\text{ID}}$, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie alle Informationen zu Vorlesungen zusammen mit den Profs aus, die sie halten.

$$\sigma_{gehalten\_von=Prof.ID}(Prof \times Vorlesung)$$

Anfrage: Geben Sie für jede Vorlesung jeweils die Vorlesungs-ID sowie den Nachnamen des Profs aus, der sie hält.

$$\pi_{Vorlesung.ID,Nachname}(\sigma_{gehalten\_von=Prof.ID}(Prof \times Vorlesung))$$

Umbenennung

Mit der **Umbenennung** $\rho_\alpha(R)$ können entweder die Relation selbst oder Attribute in der Relation umbenannt werden.

- Keine Auswirkung auf die Tupel in der Relation
- Umbenennung nur innerhalb der Anfrage, keine Änderung der Basisrelationen
- wird in der Regel eingesetzt, um Namenskonflikte aufzulösen
  - bei Mengenoperationen:
    unterschiedliche Attribute werden gleich benannt
  - beim Kreuzprodukt:
    gleiche Attribute werden unterschiedlich benannt

Umbenennung

Umbenennung von Attributen

- Syntax: $\rho_{<alterName> \rightarrow <neuerName>}(< Relation >)$
- Mit $\rho_{A \rightarrow B}(R)$ wird das Attribut A in R zu B umbenannt.
- Umbenennung mehrere Attribute ist gleichzeitig möglich: $\rho_{A_1 \rightarrow B_1, A_2 \rightarrow B_2, \ldots}(R)$

Umbenennung einer Relation

- Syntax: $\rho_{<neuerName>}(Relation)$
- Mit $\rho_S(R)$ wird die Relation R zu S umbenannt.
- meist nötig, wenn eine Relation mehrfach in einer Anfrage verwendet wird.

Umbenennung Anfrage

Studi(MatrNr, Vorname, Nachname, Fach, Semester)   hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von)   Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage Geben Sie die IDs aller Profs aus, die keine Vorlesung halten.

$$\pi_{ID}(Prof) - \rho_{gehalten\_von \to ID}(\pi_{gehalten\_von}(Vorlesung))$$

Umbenennung Anfrage

Studi(<u>MatrNr</u>, Vorname, Nachname, Fach, Semester)   hört($\overline{\text{MatrNr}}$, $\overline{\text{VID}}$)
Vorlesung(<u>ID</u>, Titel, CP, $\overline{\text{gehalten\_von}}$)   Prof(<u>ID</u>, Vorname, Nachname, Lehrstuhl)

Anfrage Geben Sie die Nachnamen aller Studis aus, die mehrfach vorkommen.

$$\pi_{S1.Nachname}(\sigma_{S1.Nachname=S2.Nachname \wedge S1.MatrNr \neq S2.MatrNr}(\rho_{S1}(Studi) \times \rho_{S2}(Studi)))$$

Zusammenfassung

- Operatoren einer (minimalen) Relationenalgebra:
  - Projektion π
  - Selektion σ
  - Vereinigung $\cup$
  - Differenz $-$
  - Kreuzprodukt $\times$
  - Umbenennung ρ

Die Operatoren $\Omega = \{\pi, \sigma, \cup, -, \times, \rho\}$ bilden eine **minimale Relationenalgebra**.

- Es gibt noch weitere Operatoren, die im weiteren Verlauf besprochen werden
- Auch andere Mengen von Operatoren können ebenfalls eine minimale Relationenalgebra bilden.

Komplexe Anfragen

Studi(MatrNr, Vorname, Nachname, Fach, Semester)   hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von)   Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie den Nachnamen und die Semesteranzahl aller Informatik-Studierenden aus, die mindestens im 3. Semester sind.

$$\pi_{Nachname,Semester}(\sigma_{Semester \geq 3 \wedge Fach = \text{'Informatik'}}(Studi))$$
$$= \sigma_{Semester \geq 3}(\pi_{Name,Semester}(\sigma_{Fach = \text{'Informatik'}}(Studi)))$$

Komplexe Anfragen

Studi(MatrNr, Vorname, Nachname, Fach, Semester) hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von) Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die MatrNr aller Studis aus, die eine Vorlesung mit dem Titel Datenbanken hören.

$$\pi_{MatrNr}(\sigma_{Titel='\text{Datenbanken}' \land ID=VID}(Vorlesung \times hört))$$
$$= \pi_{MatrNr}(\sigma_{ID=VID}(hört \times \sigma_{Titel='\text{Datenbanken}'}(Vorlesung)))$$

Komplexe Anfragen

Studi(MatrNr, Vorname, Nachname, Fach, Semester) hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von) Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die IDs aller Profs aus, die *ausschließlich* Vorlesungen halten, die 5 CPs bringen.
$\hat{=}$ *Geben Sie die IDs aller Profs aus, die Vorlesungen halten, aber keine Vorlesung halten, die nicht 5 CP bringt.*

$$\pi_{gehalten\_von}(Vorlesung) - \pi_{gehalten\_von}(\sigma_{CP \neq 5}(Vorlesung))$$

Anmerkung: Eine Bedingung, *ausschließlich* gelten soll, wird i.d.R. mit Differenz und negierter Bedingung umgesetzt.

Überblick

1. Operatoren der minimalen Relationenalgebra
   - Projektion π
   - Selektion σ
   - Vereinigung $\cup$
   - Differenz $-$
   - Kreuzprodukt $\times$
   - Umbenennung ρ
2. **Weitere Operatoren**
   - Schnitt $\cap$
   - Joins $\bowtie$
   - Division $\div$
3. Erweiterungen der Relationenalgebra

Schnitt

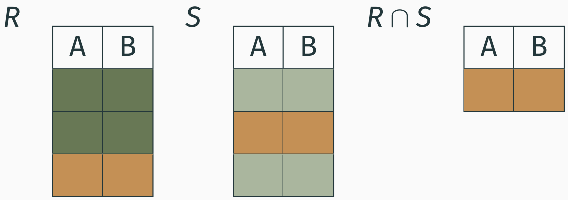
Seien $R \subset REL(X)$ und $S \subset REL(X)$ zwei Relationen über demselben Schema X.

Der **Schnitt** $R \cap S$ gibt die Tupel zurück, die sowohl in der Relation R, als auch der Relation S enthalten sind.

Syntax: $< Relation1 > \cap < Relation2 >$

Schnitt formal

$$R \cap S = \{t \mid t \in R \wedge t \in S\} = R - (R - S)$$



Schnitt Anfrage

Studi(MatrNr, Vorname, Nachname, Fach, Semester) hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von) Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage Geben Sie die Nachnamen aller Studis aus, bei denen ein Prof mit demselben Nachnamen existiert.

$$\pi_{Nachname}(Studi) \cap \pi_{Nachname}(Prof)$$

Erinnerung: Kreuzprodukt und Selektion

Kreuzprodukt wird i.d.R. zusammen mit einer Selektionsbedingung verwendet.

Studi($\underline{\text{MatrNr}}$, Vorname, Nachname, Fach, Semester)   hört($\overline{\underline{\text{MatrNr}, \text{VID}}}$)
Vorlesung($\underline{\text{ID}}$, Titel, CP, $\overline{\text{gehalten\_von}}$)   Prof($\underline{\text{ID}}$, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie alle Informationen zu Vorlesung zusammen mit den Profs aus, die sie halten.

$$\sigma_{gehalten\_von=Prof.ID}(Prof \times Vorlesung)$$

Theta-Join

Der **Theta-Join** $R \bowtie_\theta S$ enthält die Kombination von allen Tupeln von R mit Tupeln von S, die zusammen die Bedingung θ erfüllen.

Syntax: $< Relation1 > \bowtie_{<Bedingung>} < Relation2 >$

Theta-Join formal

$$R \bowtie_\theta S = \sigma_\theta(R \times S).$$

- θ ein Prädikat über die Attribut von R und S.
- Mehrere Vergleiche können mit $\wedge$ (und) verknüpft werden.

Beim **Equi-Join** enthält die Bedingung θ nur „$=$“-Vergleiche zwischen Attributen der beiden Relationen.

Theta-Join/Equi-Join Anfrage

Studi($\underline{\text{MatrNr}}$, Vorname, Nachname, Fach, Semester)   hört($\overline{\underline{\text{MatrNr}}}$, $\overline{\underline{\text{VID}}}$)
Vorlesung($\underline{\text{ID}}$, Titel, CP, $\overline{\text{gehalten\_von}}$)   Prof($\underline{\text{ID}}$, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie alle Informationen zu Vorlesung zusammen mit den Profs aus, die sie halten.

$$\sigma_{gehalten\_von=Prof.ID}(Prof \times Vorlesung) = Prof \bowtie_{gehalten\_von=Prof.ID} Vorlesung$$

Theta-Join/Equi-Join Anfrage

Studi($\underline{\text{MatrNr}}$, Vorname, Nachname, Fach, Semester) hört($\overline{\underline{\text{MatrNr}, \text{VID}}}$)
Vorlesung($\underline{\text{ID}}$, Titel, CP, $\overline{\text{gehalten\_von}}$) Prof($\underline{\text{ID}}$, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Fächer der Studis aus, die die Vorlesung mit der ID 123 hören.

$$\pi_{Fach}(\sigma_{VID=123}(\textit{hört}) \bowtie_{\textit{hört}.MatrNr=Studi.MatrNr} Studi)$$

Natural Join

Seien $R \subset REL(X)$ und $S \subset REL(Y)$ zwei Relationen.

Der **natural Join** $R \bowtie S$ kombiniert zwei Relationen über gleiche Werte in gleichnamigen Attributen.

Syntax: $< Relation1 > \bowtie < Relation2 >$

- Das Resultat von $R \bowtie S$ ist eine Relation mit einem Schema, das alle Attribute von R enthält und alle Attribute von S, die nicht in R vorkommen.
- Wenn $X \cap Y = \emptyset$, dann gilt $R \bowtie S = R \times S$.

Natural Join Anfrage

Studi(MatrNr, Vorname, Nachname, Fach, Semester) hört(MatrNr, VID)
Vorlesung(ID, Titel, CP, gehalten_von) Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Fächer der Studis aus, die die Vorlesung mit der ID 123 hören.

$$\pi_{Fach}(\sigma_{VID=123}(\textit{hört}) \bowtie_{\textit{hört}.MatrNr=Studi.MatrNr} Studi)$$
$$= \pi_{Fach}(\sigma_{VID=123}(\textit{hört}) \bowtie Studi)$$

Natural Join Anfrage

⚠ Natural Join verwendet **alle** gleichnamigen Attribute.

Studi(MatrNr, Vorname, Nachname, Fach, **Semester**)   hört(MatrNr, VID, **Semester**)
Vorlesung(ID, Titel, CP, gehalten_von)   Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Fächer der Studis aus, die die Vorlesung mit der ID 123 hören.

$\pi_{Fach}(\sigma_{VID=123}(hört) \bowtie Studi)$ ergibt nicht mehr das gewünschte Ergebnis.

Alternative 1: Join-Attribute mit Equi-Join festlegen

$$\pi_{Fach}(\sigma_{VID=123}(hört) \bowtie_{Studi.MatrNr=hört.MatrNr} Studi)$$

Natural Join Anfrage

⚠ Natural Join verwendet **alle** gleichnamigen Attribute.

Studi(MatrNr, Vorname, Nachname, Fach, **Semester**)   hört(MatrNr, VID, **Semester**)
Vorlesung(ID, Titel, CP, gehalten_von)   Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Fächer der Studis aus, die die Vorlesung mit der ID 123 hören.

$\pi_{Fach}(\sigma_{VID=123}(hört) \bowtie Studi)$ ergibt nicht mehr das gewünschte Ergebnis.

Alternative 2: Attribut vor Natural Join umbenennen

$$\pi_{Fach}(\sigma_{VID=123}(hört) \bowtie \rho_{Semester \rightarrow Fachsemester}(Studi))$$

Natural Join Anfrage

⚠ Natural Join verwendet **alle** gleichnamigen Attribute.

Studi(MatrNr, Vorname, Nachname, Fach, **Semester**) hört(MatrNr, VID, **Semester**)
Vorlesung(ID, Titel, CP, gehalten_von) Prof(ID, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die Fächer der Studis aus, die die Vorlesung mit der ID 123 hören.

$\pi_{Fach}(\sigma_{VID=123}(hört) \bowtie Studi)$ ergibt nicht mehr das gewünschte Ergebnis.

Alternative 3: Attribut durch Projektion entfernen (falls nicht benötigt)

$$\pi_{Fach}(\sigma_{VID=123}(hört) \bowtie \pi_{MatrNr,Fach}(Studi))$$

Division

Seien $R_1 \subset REL(X_1)$ und $R_2 \subset REL(X_2)$ zwei Relationen mit $X_2 \subset X_1$.

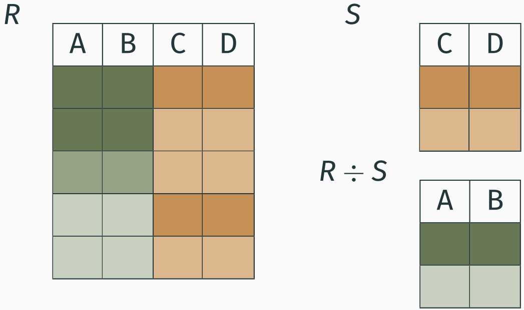
Das Ergebnis der **Division** $R_1 \div R_2$ besteht aus genau den Tupeln t, sodass t in Kombination mit allen Tupeln s aus R_2 in der Relation R_1 vorkommt.

$$t \in R_1 \div R_2 \;\Leftrightarrow\; t \circ s \in R_1 \; \forall s \in R_2$$

Syntax: $< Relation1 > \div < Relation2 >$

Division formal

$$R_1 \div R_2 = \pi_{X'}(R_1) - \pi_{X'}((\pi_{X'}(R_1) \times R_2) - R_1)$$
$$\text{mit } X' = X_1 \setminus X_2$$



Division

- $X_2 \subsetneq X_1$ muss erfüllt sein
- Schema von $R_1 \div R_2$ ist $X' = X_1 \setminus X_2$.
- Operator, mit dem ein Allquantor ausgedrückt werden kann.
- nicht-monotone Operation

R_1

MatrNr	VID
1	123
1	42
2	42
3	200
3	42
3	123
4	200

R_2

VID
123
42

$R_1 \div R_2$

MatrNr
1
3

R_3

VID
123
42
200

$R_1 \div R_3$

MatrNr
3

Division Anfrage

Studi(<u>MatrNr</u>, Vorname, Nachname, Fach, Semester) hört($\overline{\text{MatrNr}}$, $\overline{\text{VID}}$)
Vorlesung(<u>ID</u>, Titel, CP, $\overline{\text{gehalten\_von}}$) Prof(<u>ID</u>, Vorname, Nachname, Lehrstuhl)

Anfrage: Geben Sie die MatrikelNr aller Studis aus, die *alle* Vorlesungen vom Prof mit der ID 123 gehört haben.

$$hört \div \rho_{ID \rightarrow VID}(\pi_{ID}(\sigma_{gehalten\_von=123}(Vorlesung)))$$

weitere Operatoren

- Semi-Joins: $\ltimes$, $\rtimes$
- Anti-Join: $\triangleright$
- Outer Joins: $⟕$, $⟖$, $⟗$

(nicht klausurrelevant)

Einschränkungen und Erweiterungen

Einschränkungen

- Keine `null`-Werte
- Sortierung
- Zählen
- Berechnungen
- Teilstring-Vergleiche
- Existenzbedingungen

Erweiterungen

- NF^2-Algebra, Gruppierung
- Umgang mit `null`-Werte
- Operationen auf Werten, Arithmetische Funktionen

Relationenalgebra vs. Relationenkalkül

$\text{hört}(\underline{\overline{\text{MatrNr}}, \overline{\text{VID}}}) \quad \text{Vorlesung}(\underline{\text{ID}}, \text{Titel}, \text{CP}, \overline{\text{gehalten\_von}})$

Relationenalgebra: prozeduale Sprache

$$\pi_{MatrNr}(\mathit{hört} \bowtie_{VID=ID} \sigma_{Titel=\text{'Datenbanken'}}(\mathit{Vorlesung}))$$

Relationenkalkül: deklarative Sprache

Tupelkalkül

$\{t.MatrNr \mid \mathit{hört}(t) \wedge \mathit{Vorlesung}(s) \wedge t.VID = s.ID \wedge s.Titel = \text{'Datenbanken'}\}$ (mit Namen)

$\{t[1] \mid \mathit{hört}(t) \wedge \mathit{Vorlesung}(s) \wedge t[2] = s[1] \wedge s[2] = \text{'Datenbanken'}\}$ (mit Reihenfolge)

Domänenkalkül

$\{a \mid \mathit{hört}(a, b) \wedge \mathit{Vorlesung}(b, x, y, z) \wedge x = \text{'Datenbanken'}\}$