



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS17/18

Harald Lang, Linnea Passing (gdb@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1718/grundlagen/>

Blatt Nr. 04

Tool zum Üben der relationalen Algebra:

<http://db.in.tum.de/people/sites/muehe/ira/>

Tool zum Üben von SQL-Anfragen:

<http://hyper-db.com/interface.html>

Hausaufgabe 1

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in der Relationalen Algebra:

- Finden Sie die *Assistenten* von *Professoren*, die den Studenten Fichte unterrichtet haben – z.B. als potentielle Betreuer seiner Diplomarbeit.
- Finden Sie die *Studenten*, die *Vorlesungen* hören (bzw. gehört haben), für die ihnen die direkten Voraussetzungen fehlen.

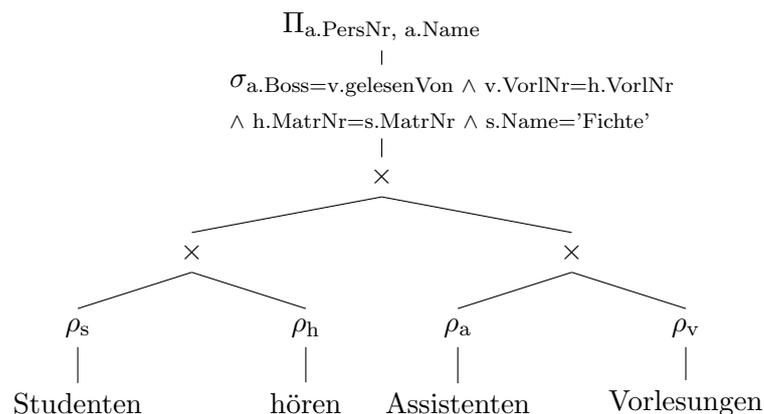
Lösung:

Die Anfragen sehen in relationaler Algebra wie folgt aus:

- Folgende Abfrage bildet zuerst das Kreuzprodukt über alle beteiligten Relationen, d.h. *Studenten*, *Vorlesungen*, *Assistenten* und *hören*. Anschließend erfolgt eine umfangreiche Selektion, die die auf Fichte zugeschnittenen Tupel extrahiert.

$$\Pi_{a.PersNr, a.Name}(\sigma_{a.Boss=v.gelesenVon \wedge v.VorlNr=h.VorlNr \wedge h.MatrNr=s.MatrNr \wedge s.Name='Fichte'}(\rho_a(\text{Assistenten}) \times \rho_s(\text{Studenten}) \times \rho_v(\text{Vorlesungen}) \times \rho_h(\text{hören})))$$

In Operatorbaumdarstellung:

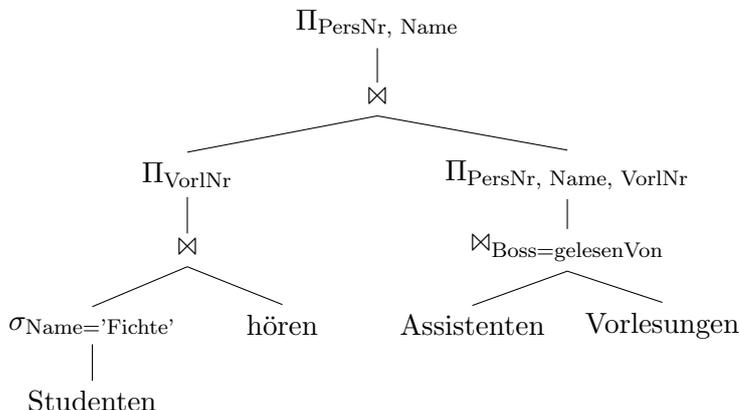


Die Bildung des Kreuzprodukts gilt es nach Möglichkeit zu vermeiden, da dadurch mitunter sehr große Zwischenergebnisse entstehen. Dies kann zu spürbaren Leistungseinbußen während der Anfragebearbeitung führen. Folgende Anfrage berechnet dieselbe Ergebnismenge, setzt jedoch bereits Optimierungstechniken, wie frühe Selektion

und den (natürlichen) Verbundoperator ein.

$$\Pi_{\text{PersNr, Name}}((\Pi_{\text{PersNr, Name, VorlNr}}(\text{Assistenten} \bowtie_{\text{Boss=gelesenVon}} \text{Vorlesungen})) \bowtie (\Pi_{\text{VorlNr}}(\sigma_{\text{Name='Fichte'}}(\text{Studenten}) \bowtie \text{hören})))$$

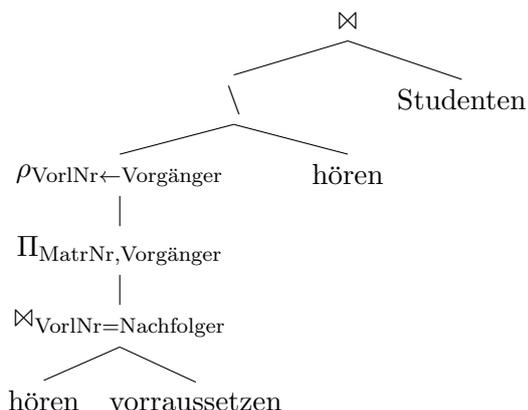
In Operatorbaumdarstellung:



- b) Wir konstruieren eine hypothetische Ausprägung der Relation *hören*, die gelten müsste, wenn alle Studenten alle benötigten Vorgängervorlesungen hören. Von dieser Menge ziehen wir die tatsächliche Ausprägung von *hören* ab, so dass diejenigen Einträge übrig bleiben, bei denen ein Student die Vorgängervorlesung nicht hört (bzw. gehört hat).

$$R := (\rho_{\text{VorlNr} \leftarrow \text{Vorgänger}}(\Pi_{\text{MatrNr, Vorgänger}}(\text{hören} \bowtie_{\text{VorlNr=Nachfolger}} \text{voraussetzen})) - \text{hören}) \bowtie \text{Studenten}$$

In Operatorbaumdarstellung:



Hausaufgabe 2

Formulieren Sie die folgenden Anfragen auf dem bekannten Universitätsschema im relationalen **Tupelkalkül** und im relationalen **Domänenkalkül**:

- Geben Sie alle *Vorlesungen* an, die der *Student* Xenokrates gehört hat.
- Geben Sie die Titel der direkten Voraussetzungen für die *Vorlesung* Wissenschaftstheorie an.

- c) Geben Sie Paare von *Studenten*(-Namen) an, die sich aus der *Vorlesung* Grundzüge kennen.

Lösung:

- a) Geben Sie alle *Vorlesungen* an, die der *Student* Xenokrates gehört hat.

Formulierung im Tupelkalkül

$$\{v \mid v \in \text{Vorlesungen} \wedge \exists h \in \text{hören}(v.\text{VorlNr} = h.\text{VorlNr} \wedge \exists s \in \text{Studenten}(s.\text{MatrNr} = h.\text{MatrNr} \wedge s.\text{Name} = \text{'Xenokrates'}))\}$$

Formulierung im Domänenkalkül

$$\{[v,t] \mid \exists s,g([v,t,s,g] \in \text{Vorlesungen} \wedge \exists m([m,v] \in \text{hören} \wedge \exists \text{sem}([m,\text{'Xenokrates'},\text{sem}] \in \text{Studenten})))\}$$

- b) Geben Sie die Titel der direkten Voraussetzungen für die *Vorlesung* Wissenschaftstheorie an.

Formulierung im Tupelkalkül

$$\{[v.\text{Titel}] \mid v \in \text{Vorlesungen} \wedge \exists \text{vor} \in \text{voraussetzen}(v.\text{VorlNr} = \text{vor.Vorgänger} \wedge \exists v2 \in \text{Vorlesungen}(v2.\text{VorlNr} = \text{vor.Nachfolger} \wedge v2.\text{Titel} = \text{'Wissenschaftstheorie'}))\}$$

Formulierung im Domänenkalkül

$$\{[t] \mid \exists v,s,g([v,t,s,g] \in \text{Vorlesungen} \wedge \exists v2([v,v2] \in \text{voraussetzen} \wedge \exists s2,g2([v2,\text{'Wissenschaftstheorie'},s2,g2] \in \text{Vorlesungen})))\}$$

- c) Geben Sie Paare von *Studenten*(-Namen) an, die sich aus der *Vorlesung* Grundzüge kennen.

Formulierung im Tupelkalkül

$$\{[s1.\text{Name}, s2.\text{Name}] \mid s1,s2 \in \text{Studenten} \wedge \exists h1,h2 \in \text{hören} (s1.\text{MatrNr} = h1.\text{MatrNr} \wedge s2.\text{MatrNr} = h2.\text{MatrNr} \wedge h1.\text{VorlNr} = h2.\text{VorlNr} \wedge s1.\text{MatrNr} \neq s2.\text{MatrNr} \wedge \exists v \in \text{Vorlesungen}(h1.\text{VorlNr} = v.\text{VorlNr} \wedge v.\text{Titel} = \text{'Grundzüge'}))\}$$

Formulierung im Domänenkalkül

$$\{[n1,n2] \mid \exists m1,m2,s1,s2(m1 \neq m2 \wedge [m1,n1,s1] \in \text{Studenten} \\ \wedge [m2,n2,s2] \in \text{Studenten} \wedge \exists v([m1,v] \in \text{hören} \\ \wedge [m2,v] \in \text{hören} \wedge \exists s,g([v, 'Grundzüge',s,g] \in \text{Vorlesungen})))\}$$

Hausaufgabe 3

Gegeben sei die folgende Relation **Zehnkampf** mit Athletennamen und den von ihnen erreichten Punkten im Zehnkampf:

Name	Punkte
Eaton	8869
Suarez	8523
Behrenbruch	8126
Hardee	8671
...	...

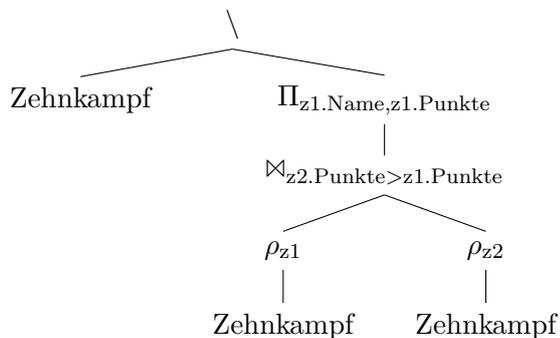
- Ermitteln Sie die Goldmedaillengewinner in relationaler Algebra. (Eine Goldmedaille bekommen alle, für die gilt: es gibt niemand besseren (also mit mehr Punkten).)
- Ermitteln Sie die Silbermedaillengewinner im Tupelkalkül. (Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/n bessere/n.)

Lösung:

- Goldmedaillengewinner in relationaler Algebra:

$$(\text{Zehnkampf} \setminus (\Pi_{\text{Name,Punkte}}(\rho_{z1}\text{Zehnkampf} \bowtie_{z2.\text{Punkte} > z1.\text{Punkte}} \rho_{z2}\text{Zehnkampf})))$$

In Operatorbaumdarstellung:



- Silbermedaillengewinner im Tupelkalkül:

$$\{k \mid k \in \text{Zehnkampf} \wedge \\ \exists k_{gold} \in \text{Zehnkampf}(\\ k_{gold}.\text{Punkte} > k.\text{Punkte} \wedge \forall k_{andere} \in \text{Zehnkampf}(\\ k_{andere}.\text{Punkte} \geq k_{gold}.\text{Punkte} \Rightarrow k_{andere}.\text{Name} = k_{gold}.\text{Name}) \wedge \\ \neg \exists k_{zwischen} \in \text{Zehnkampf}(k_{zwischen}.\text{Punkte} > k.\text{Punkte} \wedge \\ k_{zwischen}.\text{Punkte} < k_{gold}.\text{Punkte}))\}$$

Hausaufgabe 4

Gegeben seien die beiden Relationen $R : \{[a_1, \dots, a_n]\}$ und $S : \{[b_1, \dots, b_m]\}$. Geben Sie die folgenden Ausdrücke im Tupel- und Domänenkalkül an:

- a) $Q_1 := R \bowtie_{a_1=b_1} S$
- b) $Q_2 := R \bowtie_{a_1=b_1} S$
- c) $Q_3 := R \bowtie_{a_1=b_1} S$
- d) $Q_4 := R \natural_{a_1=b_1} S$

Lösung:

Bitte beachten Sie, dass in dieser Aufgabe ausschließlich allgemeine *Theta*joins ($\bowtie_{\Theta}, \bowtie_{\Theta}, \dots$) verwendet werden. Gemäß Definition werden somit alle Attribute der beiden Eingaberelationen in die Ausgaberelation projiziert, einschließlich der Attribute, welche in der Joinbedingung enthalten sind. Lediglich bei *natürlichen* Joins, wo implizit eine Gleichheitsbedingung für alle gleichnamigen Attribute erfüllt sein muss, werden gleichnamige Attribute nicht doppelt in die Ausgaberelation projiziert. Siehe hierzu auch Folie Kapitel 3, "Andere Join-Arten".

- a) $Q_1 := R \bowtie_{a_1=b_1} S$

Formulierung im Tupelkalkül

$$Q_1 := \{[r.a_1, \dots, r.a_n, s.b_1, \dots, s.b_m] \mid r \in R \wedge s \in S \wedge r.a_1 = s.b_1\}$$

Da der Joinoperator Tupel aus verschiedenen Relationen verbindet, müssen für die Ergebnismenge neue Tupel mithilfe des Tupelkonstruktors konstruiert werden: $[\text{attribut}_1 : \text{wert}_1, \dots, \text{attribut}_n : \text{wert}_n]$. Die oben verwendete Tupelkonstruktion $[r.a_1, \dots]$ ist eine verkürzte Schreibweise für $[a_1 : r.a_1, \dots]$ und kann verwendet werden, wenn der Attributname im neuen Tupel unverändert bleibt.

Im Falle des Thetajoins kann auch die Tupelkonkatenation $t_1 \circ t_2$ verwendet werden:

$$Q_1 := \{r \circ s \mid r \in R \wedge s \in S \wedge r.a_1 = s.b_1\}$$

Formulierung im Domänenkalkül

$$Q_1 := \{[a_1, \dots, a_n, b_1, \dots, b_m] \mid [a_1, \dots, a_n] \in R \wedge [b_1, \dots, b_m] \in S \wedge a_1 = b_1\}$$

oder

$$Q_1 := \{[a_1, \dots, a_n, b_1 : a_1, b_2, \dots, b_m] \mid [a_1, \dots, a_n] \in R \wedge [a_1, b_2, \dots, b_m] \in S\}$$

- b) $Q_2 := R \bowtie_{a_1=b_1} S$

Formulierung im Tupelkalkül

$$Q_2 := Q_1 \cup \{[r.a_1, \dots, r.a_n, b_1 : \text{null}, \dots, b_m : \text{null}] \mid r \in R \wedge \exists s \in S(r.a_1 = s.b_1)\}$$

Formulierung im Domänenkalkül

$$Q_2 := Q_1 \cup \{[a_1, \dots, a_n, b_1 : \text{null}, \dots, b_m : \text{null}] \mid [a_1, \dots, a_n] \in R \wedge \exists c_2, \dots, c_m([a_1, c_2, \dots, c_m] \in S)\}$$

c) $Q_3 := R \bowtie_{a_1=b_1} S$

Formulierung im Tupelkalkül

$$Q_3 := \{s \mid s \in S \wedge \exists r \in R(r.a_1 = s.b_1)\}$$

Formulierung im Domänenkalkül

$$Q_3 := \{[b_1, \dots, b_m] \mid [b_1, \dots, b_m] \in S \wedge \exists a_2, \dots, a_n([b_1, a_2, \dots, a_n] \in R)\}$$

d) $Q_4 := R \ltimes_{a_1=b_1} S$

Formulierung im Tupelkalkül

$$Q_4 := \{s \mid s \in S \wedge \nexists r \in R(r.a_1 = s.b_1)\}$$

Formulierung im Domänenkalkül

$$Q_4 := \{[b_1, \dots, b_m] \mid [b_1, \dots, b_m] \in S \wedge \nexists a_2, \dots, a_n([b_1, a_2, \dots, a_n] \in R)\}$$

Quiz 1

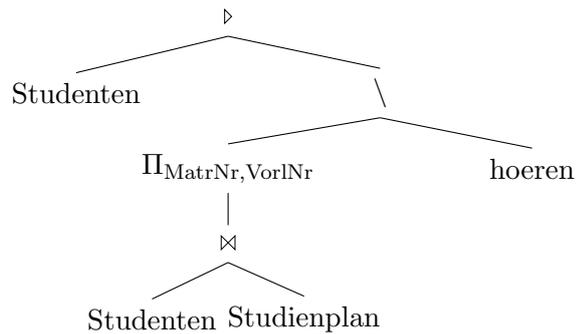
Erweitern Sie die Uni-Datenbank um die Relation

$$\text{StudienPlan} : \{[\text{Semester}, \text{VorlesungsNr}]\}$$

- Bestimmen Sie in relationaler Algebra die Studenten, die alle für ihr Semester vorgesehenen Vorlesungen hören.
- Bestimmen Sie in SQL die Studenten, die nur Vorlesungen ihres Semesters hören (nicht notwendigerweise alle).

Lösung:

- In Operatorbaumdarstellung:



```

b) -- Test Daten
with studienplan (semester, vorlnr) as (
  values (2, 5001), (2, 5049), (3, 5041), (3, 5052),
         (3, 5216), (3, 5259), (6, 4052), (12, 5022)
)

-- Loesung
select *
  from studenten s
 where not exists (select *
                  from hoeren h, studienplan sp
                  where s.matrnr = h.matrnr
                     and h.vorlnr = sp.vorlnr
                     and s.semester <> sp.semester)
    and exists (select *
               from hoeren h
               where h.matrnr = s.matrnr);

-- Alternative Loesung mit 'all'
select *
  from studenten s
 where s.semester = all(select sp.semester
                       from hoeren h, studienplan sp
                       where s.matrnr = h.matrnr
                          and h.vorlnr = sp.vorlnr)
    and exists (select *
               from hoeren h
               where h.matrnr = s.matrnr);

```