



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS16/17

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

<http://www-db.in.tum.de/teaching/ws1617/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>

### Aufwärmübung (wird nicht in der Übung besprochen)

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

### 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 Ausgabere Relation 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 Ausgabere Relation 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 Theta-Joins 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)  $Q_4 := R \triangleleft_{a_1=b_1} S$

### Formulierung im Tupelkalkül

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

### Formulierung im Domänenkalkül

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

### Hausaufgabe 1

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

### Lösung:

- 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

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

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

### Formulierung im Tupelkalkül

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

### Formulierung im Domänenkalkül

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

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

### Formulierung im Tupelkalkül

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

### Formulierung im Domänenkalkül

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

### Hausaufgabe 2

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 Silbermedaillengewinner im Tupelkalkül. (Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/n bessere/n.)

### Lösung:

Silbermedaillengewinner im Tupelkalkül:

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

### Hausaufgabe 3

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in SQL:

- Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.
- Finden Sie die *Studenten*, die *Vorlesungen* hören, die auch Fichte hört.
- Finden Sie die *Assistenten* von *Professoren*, die den *Studenten* Fichte unterrichtet haben – z.B. als potentielle Betreuer seiner Diplomarbeit.

- (d) Geben Sie die Namen der *Professoren* an, die Xenokrates aus *Vorlesungen* kennt.
- (e) Welche *Vorlesungen* werden von *Studenten* im Grundstudium (1.-4. Semester) gehört? Geben Sie die Titel dieser *Vorlesungen* an.

**Lösung:**

- (a) Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.

```
select s.Name, s.MatrNr
from Studenten s, hoeren h, Vorlesungen v,
     Professoren p
where s.MatrNr = h.MatrNr
     and h.VorlNr = v.VorlNr
     and v.gelesenVon = p.PersNr
     and p.Name = 'Sokrates';
```

DISTINCT wäre nett, um Duplikate zu unterdrücken ist aber nicht explizit in der Aufgabe gefordert.

- (b) Finden Sie die *Studenten*, die *Vorlesungen* hören, die auch Fichte hört.

```
select distinct s1.Name, s1.MatrNr
from Studenten s1, Studenten s2, hoeren h1, hoeren h2
where s1.MatrNr = h1.MatrNr
     and s1.MatrNr != s2.MatrNr
     and s2.MatrNr = h2.MatrNr
     and h1.VorlNr = h2.VorlNr
     and s2.Name = 'Fichte';
```

- (c) Finden Sie die *Assistenten* von *Professoren*, die den *Studenten* Fichte unterrichtet haben – z.B. als potentielle Betreuer seiner Diplomarbeit.

```
select a.Name, a.PersNr
from Assistenten a, Professoren p, Vorlesungen v,
     hoeren h, Studenten s
where a.Boss = p.PersNr
     and p.PersNr = v.gelesenVon
     and v.VorlNr = h.VorlNr
     and h.MatrNr = s.MatrNr
     and s.Name = 'Fichte';
```

- (d) Geben Sie die Namen der *Professoren* an, die Xenokrates aus *Vorlesungen* kennt.

```
select p.PersNr, p.Name
from Professoren p, hoeren h, Vorlesungen v,
     Studenten s
where p.PersNr = v.gelesenVon
     and v.VorlNr = h.VorlNr
     and h.MatrNr = s.MatrNr
     and s.Name = 'Xenokrates';
```

- (e) Welche *Vorlesungen* werden von *Studenten* im Grundstudium (1.-4. Semester) gehört? Geben Sie die Titel dieser *Vorlesungen* an.

```
select v.Titel
from Vorlesungen v, hoeren h, Studenten s
where v.VorlNr = h.VorlNr
     and h.MatrNr = s.MatrNr
     and s.Semester between 1 and 4;
```

#### **Hausaufgabe 4**

Formulieren Sie die folgenden Anfragen auf dem bekannten Universitätsschema in SQL:

- a) Bestimmen Sie das durchschnittliche Semester der Studenten der Universität.
- b) Bestimmen Sie das durchschnittliche Semester der Studenten, die mindestens eine Vorlesung bei Sokrates hören.
- c) Bestimmen Sie, wie viele Vorlesungen im Schnitt pro Student gehört werden. Beachten Sie, dass Studenten, die keine Vorlesung hören, in das Ergebnis einfließen müssen.

## Lösung:

- a) Bestimmen Sie das durchschnittliche Semester der Studenten der Universität.

```
select avg(semester*1.0) from studenten;
```

- b) Bestimmen Sie das durchschnittliche Semester der Studenten, die mindestens eine Vorlesung bei Sokrates hören. Beachten Sie, dass Sie das Semester von Studenten, die mehr als eine Vorlesung bei Sokrates hören, nicht doppelt zählen dürfen.

```
with
vorlesungen_von_sokrates as (
  select *
  from vorlesungen v, professoren p
  where v.gelesenVon = p.persnr and p.name = '
    Sokrates'
),
studenten_von_sokrates as (
  select *
  from studenten s
  where exists (
    select *
    from hoeren h, vorlesungen_von_sokrates v
    where h.matrnr = s.matrnr and v.vorlnr = h.vorlnr
  )
)
select avg(semester) from studenten_von_sokrates
```

Man beachte, dass die Formulierung mittels WHERE EXISTS für die Elimination von Duplikaten sorgt, d.h. ein Student, der 3 Vorlesungen von Sokrates hört kommt nur einmal in Studenten\_von\_sokrates vor, was gewünscht ist. Alternativ kann man studenten\_von\_sokrates formulieren als:

```
select DISTINCT s.*
from studenten s, hoeren h, vorlesungen_von_sokrates
  v
where h.matrnr = s.matrnr and v.vorlnr = h.vorlnr
```

- c) Bestimmen Sie, wie viele Vorlesungen im Schnitt pro Student gehört werden. Beachten Sie, dass Studenten, die keine Vorlesung hören, in das Ergebnis einfließen müssen.

```
select hcount/(scount*1.000)
from (select count(*) as hcount from hoeren) h,
     (select count(*) as scount from studenten) s

select hcount/(cast scount as decimal(10,4))
from (select count(*) as hcount from hoeren) h,
     (select count(*) as scount from studenten) s
```