



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS23/24

Christoph Anneser, Michael Jungmair, Stefan Lehner, Moritz Sichert, Lukas Vogel
(gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws2324/grundlagen/>

Blatt Nr. 04

Die Vorlesung am Mittwoch, den 15.11.2023, beginnt aufgrund der SVV erst ab 11:30.

Unter <https://dbis-uibk.github.io/relax/calc/local/uibk/local/1> finden Sie ein Tool zum Üben von Relationaler Algebra.

Hausaufgabe 1

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema im Tupel- und Domänenkalkül:

- Finden Sie die *Vorlesungen*, die keine Hörer haben.
- Finden Sie die *Studenten*, die alle *Vorlesungen* hören.

Lösung:

- Finden Sie die *Vorlesungen*, die keine Hörer haben.

Formulierung im Tupelkalkül

$$\{v \mid v \in \text{Vorlesungen} \wedge \neg \exists h \in \text{hören}(v.\text{VorlNr} = h.\text{VorlNr})\}$$

oder

$$\{v \mid v \in \text{Vorlesungen} \wedge \forall h \in \text{hören}(v.\text{VorlNr} \neq h.\text{VorlNr})\}$$

Formulierung im Domänenkalkül

$$\{[v,t,s,g] \mid [v,t,s,g] \in \text{Vorlesungen} \wedge \neg \exists m([m,v] \in \text{hören})\}$$

- Finden Sie die *Studenten*, die alle *Vorlesungen* hören.

Formulierung im Tupelkalkül

$$\{s \mid s \in \text{Studenten} \wedge \forall v \in \text{Vorlesungen}(\exists h \in \text{hoeren}(v.\text{VorlNr} = h.\text{VorlNr} \wedge s.\text{MatrNr} = h.\text{MatrNr}))\}$$

Formulierung im Domänenkalkül

$$\{[m,n,s] \mid [m,n,s] \in \text{Studenten} \wedge \\ \forall v,t,sws,g([v,t,sws,g] \in \text{Vorlesungen} \Rightarrow [m,v] \in \text{hören})\}$$

Hausaufgabe 2

Gegeben sei folgende Anfrage auf dem bekannten Universitätschema:

Gesucht sind die *Professoren*, deren sämtliche *Vorlesungen* nur auf selbst gelesenen (direkten) Vorgängern aufbauen.

Formulieren Sie die Anfrage

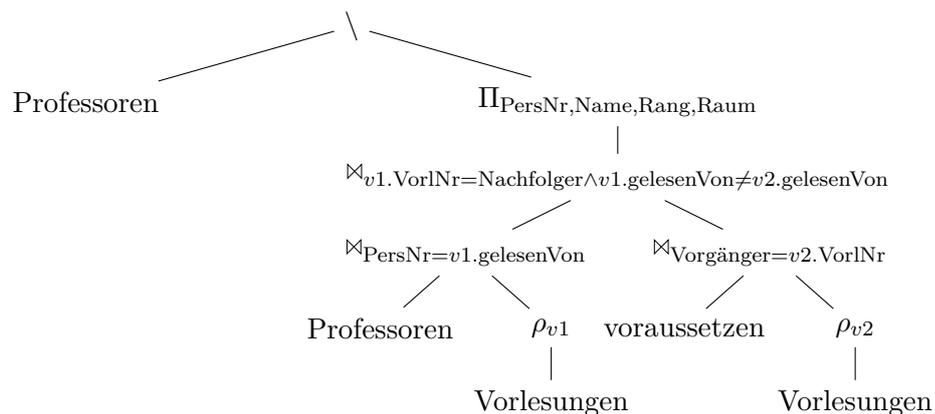
- in der Relationenalgebra,
- im Tupelkalkül und
- im Domänenkalkül.

Lösung:

Achtung: Auch Professoren, die nur Vorlesungen ohne Vorgänger lesen, sowie Professoren die überhaupt keine Vorlesung lesen sind Teil der Ergebnismenge.

Denn für alle Vorgänger (hier: keine) gilt: Sie werden vom Professor gehalten.

Formulierung in relationaler Algebra



Formulierung im Tupelkalkül

Die folgende Lösung verwendet Allquantoren, um die geforderten Bedingungen für alle Vorlesungen und deren Vorgänger zu überprüfen:

$$\{ p \mid p \in \text{Professoren} \wedge \\ \forall v_1 \in \text{Vorlesungen} (v_1.\text{gelesenVon} = p.\text{PersNr} \Rightarrow \\ \forall o \in \text{voraussetzen} (\\ o.\text{Nachfolger} = v_1.\text{VorlNr} \Rightarrow \\ \exists v_2 \in \text{Vorlesungen} (\\ o.\text{Vorgaenger} = v_2.\text{VorlNr} \wedge \\ v_2.\text{gelesenVon} = p.\text{PersNr} \\))) \}$$

Es ist auch möglich, diese Anfrage ohne Allquantoren zu formulieren:

$$\{ p \mid p \in \text{Professoren} \\ \wedge \neg \exists v_1 \in \text{Vorlesungen} (v_1.\text{gelesenVon} = p.\text{PersNr} \\ \wedge \exists o \in \text{voraussetzen} (v_1.\text{VorlNr} = o.\text{Nachfolger} \\ \wedge \exists v_2 \in \text{Vorlesungen} (o.\text{Vorgaenger} = v_2.\text{VorlNr} \\ \wedge v_2.\text{gelesenVon} \neq p.\text{PersNr}))) \}$$

Durch Äquivalenzumformungen lassen sich aus den oberen Lösungen auch weitere bilden, z.B. die folgende, die nur einen Allquantor verwendet:

$$\{ p \mid p \in \text{Professoren} \wedge \\ \forall v_1 \in \text{Vorlesungen} (v_1.\text{gelesenVon} = p.\text{PersNr} \Rightarrow \\ \neg \exists o \in \text{voraussetzen} (\\ o.\text{Nachfolger} = v_1.\text{VorlNr} \wedge \\ \exists v_2 \in \text{Vorlesungen} (\\ o.\text{Vorgaenger} = v_2.\text{VorlNr} \wedge \\ v_2.\text{gelesenVon} \neq p.\text{PersNr} \\))) \}$$

Formulierung im Domänenkalkül

Genau wie beim Tupelkalkül, kann die Anfrage im Domänenkalkül auch auf verschiedene Weisen formuliert werden. Die folgende Lösung verwendet wieder zwei Allquantoren:

$$\{ [p, n] \mid \exists rg, ra ([p, n, rg, ra] \in \text{Professoren} \wedge \\ \forall na, t_1, sws_1 ([na, t_1, sws_1, p] \in \text{Vorlesungen} \Rightarrow \\ \forall vo ([vo, na] \in \text{voraussetzen} \Rightarrow \\ \exists t_2, sws_2 ([vo, t_2, sws_2, p] \in \text{Vorlesungen} \\))) \}$$

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
...	...

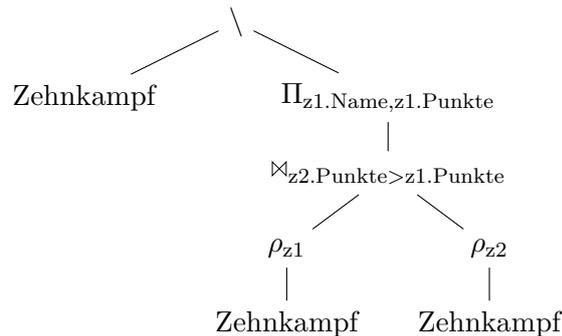
- Ermitteln Sie die Goldmedaillengewinner in relationaler Algebra. Eine Goldmedaille bekommen alle, für die gilt: es gibt niemand besseren (also mit mehr Punkten).
- Ermitteln Sie die Silbermedaillengewinner im Tupelkalkül. Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/n bessere/n.

Lösung:

- Goldmedaillengewinner in relationaler Algebra:

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

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

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in SQL:

- (a) Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.
- (b) Finden Sie die *Studenten*, die *Vorlesungen* hören, die auch Fichte hört.
- (c) 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;
```