



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS14/15

Harald Lang (harald.lang@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1415/grundlagen/>

Blatt Nr. 4

Tool zum Üben von SQL-Anfragen: <http://hyper-db.com/interface.html>.

Hausaufgabe 1

Holen Sie ggf. Aufgabe 1 von Blatt 3 nach.

Hausaufgabe 2

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

- 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.

- 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';
```

- 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 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
...	...

a) Ermitteln Sie die Goldmedaillengewinner in SQL. (Eine Goldmedaille bekommen alle Athleten, für die gilt: es gibt niemand besseren (also mit mehr Punkten).)

b) Ermitteln Sie die Silbermedaillengewinner in SQL. (Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/n bessere/n.)

HINWEIS: Beachten Sie, dass die Relation *Zehnkampf* in der oben genannten Webschnittstelle nicht existiert. Verwenden Sie die folgende Syntax um temporäre Relationen zu erzeugen:

```

with zehnkampf(name,punkte) as (
  values
    ('Eaton', 8869),
    ('Suarez', 8523),
    ('Behrenbruch', 8126),
    ('Hardee', 8671),
    ('Sebrle', 8869)
)
select * from zehnkampf order by punkte desc

```

Gold

```

select name
  from zehnkampf
 where punkte = (select max(punkte) from zehnkampf)

```

Silber

```

select name
  from zehnkampf z
 where (select count(*)
        from zehnkampf
        where punkte > z.punkte) = 1

```

Zusatzaufgabe 1 (wird nicht in der Übung besprochen)

- Übertragen Sie das ER-Modell der Zusatzaufgabe 1 von Blatt 3 in ein relationales Schema.
- Verfeinern Sie das relationale Schema durch Elimination von Relationen.
- Formulieren Sie folgende Anfrage in relationaler Algebra jeweils für die Schemas aus den Teilaufgaben a) und b): Finden Sie alle Auswirkungen des Ereignisses mit ID=10.

a)

$$\begin{aligned} \text{Ereignisse} &: \{[\underline{\text{ID}} : \text{Integer}, \text{Bezeichnung} : \text{String}]\} \\ \text{folgt} &: \{[\text{UrsacheID} : \text{Integer}, \underline{\text{WirkungID}} : \text{Integer}]\} \end{aligned}$$

b)

$$\text{Ereignisse} : \{[\underline{\text{ID}} : \text{Integer}, \text{Bezeichnung} : \text{String}, \text{UrsacheID} : \text{Integer}]\}$$

c)

$$\begin{aligned} &\Pi_{\text{Bezeichnung}}(\sigma_{\text{UrsacheID}=10}(\text{folgt}) \bowtie_{\text{WirkungID}=\text{ID}} \text{Ereignisse}) \\ &\Pi_{\text{Bezeichnung}}(\sigma_{\text{UrsacheID}=10}(\text{Ereignisse})) \end{aligned}$$