



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS13/14

Henrik Mühe (muehe@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1314/dbsys/exercises/>

Blatt Nr. 6

Tool zum Üben von SQL <http://www-db.in.tum.de/~muehe/sql/>.

Hausaufgabe 1

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 SQL. (Eine Goldmedaille bekommen alle, für die gilt: es gibt niemand besseren (also mit mehr Punkten).)
- Ermitteln Sie die Silbermedaillengewinner in SQL. (Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/nbessere/n.)

Gold

```
select name from decathlon where score=(select max(score)
from decathlon)
```

Silber

```
select name from decathlon d where(select count(*)from
decathlon where score>d.score)=1
```

Hausaufgabe 2

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

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

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

- 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*1.0) 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*1.0) from
    (select count(*) as hcount from hoeren) h,
    (select count(*) as scount from studenten) s

```

Hausaufgabe 3

„Bekanntheitsgrad“: Formulieren Sie eine SQL-Anfrage, um den Bekanntheitsgrad von Studenten zu ermitteln. Gehen Sie dabei davon aus, dass Studenten sich aus gemeinsam besuchten Vorlesungen kennen. Sortieren Sie das Ergebnis absteigend nach Bekanntheitsgrad!

Zunächst definieren wir eine View, die für jeden Studenten alle seine Bekannten auflistet. Anschließend müssen wir diese Bekannten nur noch zählen, um den Bekanntheitsgrad der Studenten zu ermitteln.

```

create view Bekannte as
(select distinct h1.MatrNr as Student, h2.MatrNr as
    Bekannter
    from hoeren h1, hoeren h2
    where h1.VorlNr = h2.VorlNr
        and h2.MatrNr != h1.MatrNr
);

select s.MatrNr, s.Name, count(*) as AnzBekannter
from Studenten s, Bekannte b
where s.MatrNr = b.Student
group by s.MatrNr, s.Name
order by AnzBekannter desc;

```

Ohne View sieht die Anfrage entsprechend komplexer aus:

```

select s.MatrNr, s.Name, count(*) as AnzBekannter
from Studenten s,

```

```
(select distinct h1.MatrNr as Student, h2.MatrNr as
  Bekannter
  from hoeren h1, hoeren h2
  where h1.VorlNr = h2.VorlNr
    and h2.MatrNr != h1.MatrNr
) b
where s.MatrNr = b.Student
group by s.MatrNr, s.Name
order by AnzBekannter desc;
```