



Übung zur Vorlesung *Einführung in die Informatik 2 für Ingenieure (MSE)*

Christoph Anneser (anneser@in.tum.de)

<http://db.in.tum.de/teaching/ss21/ei2/>

Blatt Nr. 10

Dieses Blatt wird am Montag, den 28. Juni 2021 besprochen.

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

Aufgabe 1: SQL Gruppierung

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

- (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.

```
select avg(semester*1.0)
from studenten s
where exists
    (select *
     from hoeren h, vorlesungen v, professoren p
     where v.gelesenVon = p.persnr
     and p.name = 'Sokrates'
     and 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
```

Aufgabe 2: SQL Exists

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

- (a) Finden Sie die Namen der *Studenten*, die in keiner *Prüfung* eine bessere Note als 3.0 hatten.

```

select s.name, s.matrnr
from studenten s
where not exists (select *
                  from pruefen p
                  where p.matrnr = s.matrnr
                  and p.note < 3.0);

```

(b) Finden Sie alle Professoren die einen Assistenten haben, aber dennoch keine Vorlesung geben.

```

select *
from professoren p
where exists (select *
             from assistenten a
             where a.boss = p.persnr)
and not exists (select *
               from vorlesungen v
               where v.gelesenvon = p.persnr);

```

Aufgabe 3: Bekanntheitsgrad

Formulieren Sie eine SQL-Anfrage, um den Bekanntheitsgrad von Studenten zu ermitteln. Der Bekanntheitsgrad eines Studenten ist definiert als die Anzahl an Studenten die ihn kennen. Gehen Sie dabei davon aus, dass Studenten sich aus gemeinsam besuchten Vorlesungen kennen. Sortieren Sie das Ergebnis absteigend nach Bekanntheitsgrad!

Bonus: Überlegen Sie sich warum Studenten die niemanden kennen nicht im Ergebnis auftauchen. Wie könnte man dieses Problem lösen.

```

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;

```

Aufgabe 4: Lokales DBMS [Optional, aber empfohlen!]

Installieren Sie sich lokal ihr eigenes Postgres-Datenbanksystem. Richten Sie sich im ersten Schritt eine eigene Datenbank via Command Line ein. Verbinden Sie sich dann mit einer Integrierten Entwicklungsumgebung (z.B. DataGrip) mit ihrem Datenbankserver. Beachten Sie hierzu das Zusatzblatt auf der Vorlesungswebsite.