



Exercise for *Database System Concepts for Non-Computer Scientist* im
WiSe 19/20

Alexander van Renen (renen@in.tum.de)
<http://db.in.tum.de/teaching/ws1920/DBSandere/?lang=en>

Sheet 08

Exercise 1

(Sub queries) Answer the following questions on our university database using SQL:

- a) Find all foundation lectures (lectures that don't depend on another lecture).
- b) "Lonely Students" Are there any students that are attending a lecture on their own?
Try solving it without the use of **group by**.
- c) "Industrial Students" List all students that are attending all lectures. Hint: ¹ ²

¹The task can be rephrased as: List all students, where there is no lecture with no attend record of the student.

²Alternatively, you could also try counting how many lectures a student is attending.

Solution:

```
a) select *
   from lectures l
   where not exists (
       select *
       from require r
       where l.lecturenr = r.successor)
```

b) Solution using a sub query:

```
select distinct s.studNr, name
from Students s, attend a
where s.studNr = a.studNr
and not exists(
    select *
    from attend a2
    where a2.lecturenr = a.lecturenr
    and a2.studnr <> a.studnr
)
```

Solution using group by and having:

```
select distinct min(s.studNr), min(name)
from Students s, attend a
where s.studNr = a.studNr
group by a.lectureNr
having count(*) = 1
```

c) Solution using a double negative: “Find all students where there is no lecture that they did not attend.” or “Find all students where there is no lecture, for which there is no entry in attend.”

```
select *
from Students s
where not exists
    (select *
     from lectures l
     where not exists(
         select *
         from attend a
         where l.lecturenr = a.lecturenr
         and a.studnr = s.studnr
     )
    )
```

Alternatively, we can simply count how many lectures there are in one sub query and then use a second sub query to count how many lectures a student is attending. If the two numbers match (e.g., if there are 5 lectures at a university and a student is attending 5 lectures) then the student is attending all lectures.

```
select *
from Students s
where (select count(*)
      from lectures)
      = (select count(*)
        from attend a
        where a.studNr = s.studNr)
```

Exercise 2

How many *Assistants* does each *Professor* have? Print out a list with the name of the professor and the number of her assistants. Think about why professors that don't have any assistants don't show up. How could you include them in the list. Hint: ³

Solution: The intention of this exercise was to play around with `group by`. We can write a rather short query to get the number of assistants for each professor. However, the professors without any assistants are missing, because these tuples are dropped during the join, because they don't find a join partner.

```
select p.persNr, p.name, count(*) as numberOfAssistants
from Professors p, Assistants a
where p.persNr = a.boss
group by p.persNr, p.name
```

To solve this we could use union to add the missing tuples:

```
select p.persNr, p.name, count(*) as numberOfAssistants
from Professors p, Assistants a
where p.persNr = a.boss
group by p.persNr, p.name
```

```
union
```

```
select p.persNr, p.name, 0 as numberOfAssistants
from Professors p
where not exists (select *
                  from Assistants a
                  where a.boss = p.persNr)
```

However, the simplest way to solve this query would be by using a sub query:

```
select p.persNr, p.name, (select count(*)
                          from Assistants a
                          where a.boss = p.persNr)
from Professors p
```

Exercise 3

Calculate the weighted average grade for the exams of each student. The grades should be weighted by the *weeklyhours* of the respective lecture. This is similar to the formula used for your degree. Students who did not take any test yet, do not have to be reflected in the result.

Solution:

```
select s.studnr, s.name, sum(t.grade * l.weeklyhours) /
       sum(l.weeklyhours)
from test t, students s, lectures l
where t.studnr = s.studnr
     and t.lecturenr = l.lecturenr
group by s.studnr, s.name
```

³Using union, we can combine the result of two queries with the same schema: `select a.name from (select 'Kant' as name union select 'Socrates' as name) a;`