TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

Database System Concepts for Non-Computer Scientist - WiSe 24/25

Alice Rey (rey@in.tum.de)
http://db.in.tum.de/teaching/ws2425/DBSandere/?lang=en

Sheet 07

Repetition 1

Find those students who have attended all lectures that they wrote a test in.

Exercise 2

„Busy Students“: Find all students that have more weekly hours in total than the average student. Try to simplify the query using the with construct. (Also consider students that do not attend any lecture).

Exercise 3

Create SQL DML statements for the following tasks:

a) “Professor meeting”: Move all professors to room 419.

b) “Lazy students”: Remove all students from the database who have ever failed a test (grade worse than 4.0).

Exercise 4

Write a SQL statement to create a view that gives an overview of the difficulty of each lecture. The difficulty of a lecture is defined as the sum of the weekly hours of that lecture and its direct predecessors. In our example instantiation of the university schema, the following query on your view should yield the result (only partially shown):

```
select * from LectureDifficulties;
```

lectureNr	title	difficulty
5216	Bioethik	6
4630	Die 3 Kritiken	4
...	...	...

Optional 5

Considering the following table definitions:

1)
```
create table A(a int primary key);
create table B(b int);
```

2)
```
create table A(a int primary key);
create table B(b int references A(a));
```

Assuming the cardinalities (number of tuples) of the relation A and B are $|A|$ and $|B|$, respectively. How many tuples are produced by the following queries. If no exact estimate is possible, give a range. Alternatively you can use mathematical set operations.

a) `select * from A, B;`

b) `select * from A join B on A.a = B.b;`

c) `select * from A left outer join B on A.a = B.b;`

d) `select * from A right outer join B on A.a = B.b;`

e) `select * from A full outer join B on A.a = B.b;`