Database Systems (CPH) – Fall 2025
Teacher: Tiantian Liu
Exercise Sheet: Lecture 3: SQL Part 1

AALBORG UNIVERSITY
DENMARK

1 Review

Review the following terms.

1. Create table

2. Updates to table
   insert
   delete
   drop
   alter

3. SQL query structure
   select clause
   from clause
   where clause

4. **as** clause

5. **like** clause

6. **order by** clause

7. Set operations
   union
   intersect
   except

8. Aggregate functions
   avg, min, max, sum, count
   group by
   having

AALBORG UNIVERSITY
DENMARK

2 Querying Tables

Consider the following relational schema, which obviously does not describe the standard situation at Aalborg University.

We assume that tutors are responsible for one or multiple study groups, students individually (not per group) hand in solutions for exercise sheets and receive individual grades in terms of the number of achieved points per sheet. Some of the tutors are more experienced (senior) than others.

student: {[sid: int, firstname: string, lastname: string, semester: int, birthdate: date]}
tutor: {[tid: int, firstname: string, lastname: string, issenior: boolean]}
studygroup: {[gid: int, tid → tutor, weekday: string, room: string, starttime: time]}
exercisesheet: {[eid: int, maxpoints: int]}
handsin: {[sid → student, eid → exercisesheet, achievedpoints: int]}
member: {[sid → student, gid → studygroup]}

(The DDL can be found in Moodle)

Translate the following queries into equivalent SQL statements that run on the tables created above.

1. Find the different last names of the students whose first name is ‘Helle’.

2. Find all the diffierent last names of students that end with 'sen'.

3. List the first and last names of the tutors that are senior.

4. Find the number of students in each semester.

5. Find the semesters which the number of students is larger than 5.

6. Output the names of all students and their achieved points for exercise sheet 1 (eid = 1) in descending order by the number of achieved points (achievedpoints in handsin).

AALBORG UNIVERSITY
DENMARK

3 Manipulating Tables

1. Insert at least one valid tuple into student table.

4 Using PostgreSQL

Test your solutions to the previous exercises using PostgreSQL.
(The instruction and the database schema with instances have uploaded in Moodle.)