Datenbanksysteme I

WS 2019/20

Torsten Grust, Christian Duta

Assignment #6

Submission Deadline: December 3, 2019 - 10:00

Exercise 1: About SQL (15 Points)

Please answer the following questions **briefly**:

1. Consider the following schema and query:

```
CREATE TABLE r(d real, e int, f int);
CREATE TABLE s(x int, y int);
CREATE TABLE t(a real, b text, c text);

SELECT *
FROM (SELECT r.*, t.a, t.b
      FROM r r, t t
      WHERE r.d < t.a) r1,
     s r2
WHERE r2.x <> r1.f;
```

   What is the *row type* of row variable `r1`?

2. Are the following queries valid? Explain.

   (a)

```
CREATE TABLE r(a int, b int, c int);
CREATE TABLE s(x int, y int);

SELECT r1.a, r1.b, s.x
FROM s s,
     (SELECT a, b
      FROM r r
      WHERE s.x = r.a) r1
WHERE s.y = 42;
```

   (b)

```
CREATE TABLE r(a int, b int);

SELECT r1.a FROM (SELECT a, b AS a FROM r r) r1;
```

3. Simplify the following SQL queries as much as possible.

   (a)

   ```
   CREATE TABLE r(a int, b int, c int, d int);

   SELECT r1.a, r1.b, r1.c
   FROM (TABLE r) r1
   WHERE true;
   ```

   (b)

   ```
   CREATE TABLE r(a int, b int, c int, d int);

   SELECT r1.a, r1.b, r1.c, r1.d
   FROM (SELECT t.* FROM r t) r1;
   ```

4. Consider the following query: **SELECT** r.*,s.* **FROM** r, s **WHERE** r.a = s.x. Tables r and s contain |r| and |s| tuples, respectively.

   (a) Without further knowledge, what can you say about the size of the join result?

   (b) Now, assume that x is the primary key in s. What can you now say about the size of the join result.

Exercise 2: SQL University (15 Points)

We provided you with an archive uni.zip which contains schemata and data about students, courses and lectures at a fictional university:

Student(studentid, name, major, pursueddegree, age)
Staff(staffid, name, deptid, age)
Class(classid, name, meetsat, room, staffid)
Enrolled(studentid, classid)
Department(deptid, name)

Import the schema from uni-schema.sql and then load the data into the tables with \copy <table> FROM <file> CSV; for each .csv file. Write the following SQL queries using only constructs of the SQL language which have been introduced up until the end of slide set 6 (A Diversion into SQL). The result of your SQL queries are described as **result**(c_1, c_2, ...) where c_i corresponds to a column in one of the tables described above.

1. List names of all BSc students whose name begins with "Mar". Use LIKE[1] to formulate this predicate.

   ```
   result(student_name)
   ```

2. For each class, its class name and the name of the teaching staff member.

   ```
   result(class_name, staff_name)
   ```

3. List all students enrolled in classes of the "Computer Science" department. The result should not contain any duplicates. Is DISTINCT necessary to ensure this? Explain.

   ```
   result(student_id, student_name)
   ```

[1] https://www.postgresql.org/docs/12/functions-matching.html#FUNCTIONS-LIKE

4. Find names of BSc students which are enrolled in classes taught by "Ivana Teach". Draw the join graph for your query as well.

   ```
   result(student_name)
   ```

5. List names of staff members who are at least twice as old as some student enrolled in one of their classes.

   ```
   result(staff_name)
   ```

6. Which classes have both BSc and MSc students enrolled? Draw the join graph for your query as well.

   ```
   result(class_name)
   ```

7. For each class, list its class name, the name of the teaching staff member and the name of their department. Do not forget to list teaching staff members without a designated department.
   Note: Missing departments have their names represented as `NULL`. Think about using a correlated subquery to determine the department name.

   ```
   result(class_name, staff_name, department_name)
   ```