Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Datenbanksysteme · Prof. Dr. Grust

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Datenbanksysteme I

WS 2021/22

Torsten Grust, Christian Duta, Tim Fischer

Assignment #6

Submission Deadline: December 8, 2021 - 10:00

Exercise 1: About SQL (15 Points)

Please answer the following questions **briefly**:

1. Consider the following schemata 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 AS r, t AS t
      WHERE  r.d < t.a) AS r1,
     s AS r2
WHERE r2.x <> r1.f;
```

   What is the *row type* of row variable `r1`? What is the *row type* of the overall query result?

2. Explain briefly, why the following queries are invalid.

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

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

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

SELECT r.*, t.y
FROM   r AS r, (SELECT s.y
                FROM   s AS s
                WHERE  s.x = r.a) AS t;
```

   (c)
```
CREATE TABLE r(a int, b int, c int);
CREATE TABLE s(x int, y int);
ALTER TABLE s ADD PRIMARY KEY (x);

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

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

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

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

   (c)
   ```
   SELECT ROW(v.*) :: t FROM t AS v;
   ```

4. Consider the following query: `SELECT r.*,s.* FROM r AS r, s AS s WHERE r.a = s.x`. Tables `r` and `s` contain `|r|` and `|s|` rows, 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(<u>student_id</u>, student_name, major, pursued_degree, age)
Staff(<u>staff_id</u>, staff_name, deptartment_id, age)
Class(<u>class_id</u>, class_name, meets_at, room, staff_id)
Enrolled(<u>enrolled_student_id, enrolled_class_id</u>)
Department(<u>department_id</u>, department_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 Chapter 6 (A Diversion into SQL). The resulting schemata of your SQL queries are described as **result**(c_1, c_2, ...).

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/current/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 department information shall be represented by `NULL`. Think about using a correlated subquery to determine the department name.

   result(class_name, staff_name, department_name)