

Prof. Torsten Grust, Louisa Lambrecht, Tim Fischer
WSI — Database Systems Research Group

# Assignment 6

Hand in this assignment until Thursday, 15 June 2023, 12:00 at the latest.

#### ▲ Lecture Evaluation

In the near future you will have the opportunity to **evaluate lectures** you are attending. We kindly ask you to make use of this opportunity to provide us with **your** by providing precious feedback. Check your Mailbox on .

### **E**xam-style Exercises

Exercises marked with (E) are similar in style to those you will find in the exam. You can use these to hone your expectations and gauge your skills.

### Running out of ideas?

Are you hitting a roadblock? Are some of the exercises unclear? Do you just need that one hint to get the ball rolling? Refer to the #forum channel on our Discord server and check the tag for this assignment—maybe you'll find just the help you need.

Task 1: About SQL (1 credit)

Please answer the following questions briefly:

(a) (E) 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?

(b) Explain briefly, why the following queries are invalid.

```
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;
```

```
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;
```

```
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;
```

(c) © Simplify the following SQL queries as far as possible.

```
i.
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;

ii.
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;

iii.
select r0w(v.*) :: t FROM t AS v;
```

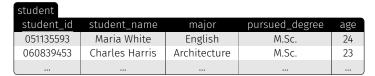
(d) (E) Consider the following query:

```
1 | 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.

- i. Without further knowledge, what can you say about the number of rows in the join result?
- ii. Now, assume that **x** is the primary key in **s**. What can you now say about the size of the join result?

In this task you will take a look at a database representing a fictional university and write queries over it to answer some questions.





enrolled	
enrolled_student_id	enrolled_class_id
112348546	2
115987938	2

department	
department_id	department_name
20	Computer Science
68	Engineering

staff			
staff_id	staff_name	staff_deptartment	age
142519864	Ivana Teach	20	54
242518965	James Smith	68	45

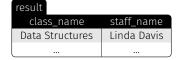
To get started, import the schema from the supplied uni-schema.sql and then load the data into the tables by running

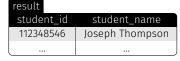
### \copy (table) FROM (file) CSV

for each .csv file. Write the following *SQL* queries using only the *SQL* constructs discussed up to Chapter 6 (A *Diversion into SQL*) of the lecture. The resulting schemas of your *SQL* queries should follow the examples on the right.

- (a) List names of all BSc students whose name begins with "Mar". Use LIKE<sup>1</sup> to formulate this predicate.
- (b) For each class, its class name and the name of the teaching staff member.
- (c) 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.
- (d) Find names of B.Sc. students which are enrolled in classes taught by "Ivana Teach". Draw the join graph for your query as well.
- (e) (E) List names of staff members who are at least twice as old as some student enrolled in one of their classes.
- (f) Which classes have both B.Sc. and M.Sc. students enrolled? Draw the join graph for your query as well.













## Note

Do not list classes for which only one category of students is enrolled.

(g) **(**E) 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.

result		
class_name	staff_name	department_name
Data Structures	Linda Davis	Computer Science
	•••	

### Note

Missing department information shall be represented by **NULL**. Think about using a correlated subquery to determine the department name.