

Exercise 5

Date of Issue: 05.04.2021

Date of Issue Solutions: 12.04.2021

The exercise is based on the *Mondial* database.

1 Advanced SQL

For the following queries, (a) formulate and write down the query in SQL, and (b) execute it on your *Mondial* database. Make sure your expressions are correct for all *Mondial* database instances.

1. The name of the third longest river.

The SQL query returns Hwangho when executed on the *Mondial* database.

2. The names of cities exactly once which have the same name as a province. Solve this task in three different ways: once using **exists**, once using **in** and once using **join**.

The SQL query returns exactly 366 rows when executed on the *Mondial* database.

3. The names of the countries together with the number of rivers in that country and the number of lakes in that country.

The SQL query returns exactly 238 rows when executed on the *Mondial* database.

2 Views

We are interested in the names of all islands that are not reachable by sea when starting in Great Britain. Only consider seas - do not consider rivers and lakes. (a) Formulate and write down the query in SQL, and (b) execute it on your *Mondial* database.

1. The **mergesWith** relation contains information about neighboring seas and is not symmetric. Create a view called **symMergesWith** that returns the symmetric version of **mergesWith**.

2. Use the constant 'Great Britain' together with view **symMergesWith** to finish the task.

The SQL query returns exactly 5 rows when executed on the *Mondial* database.

3 Triggers and Constraints

Consider the two relations **company** and **collaboratesWith** that were created in a database with the following two statements:

```
create table company (  
    name VARCHAR PRIMARY KEY,  
    country VARCHAR NOT NULL,  
    budget INTEGER  
);  
  
create table collaboratesWith (  
    company1 VARCHAR,  
    company2 VARCHAR,  
    PRIMARY KEY (company1, company2),  
    FOREIGN KEY (company1) REFERENCES company (name),  
    FOREIGN KEY (company2) REFERENCES company (name)  
);
```

1. **Constraint: A company can collaborate with at most one company.**

This constraint is not enforced in relation **collaboratesWith**. Assume that the already existing tuples in relation **collaboratesWith** do not violate the constraint and that relation **collaboratesWith** is not symmetric, i.e., for each pair of collaborating companies (A,B) only one tuple is given.

- (a) Which modification statements (insert, update, delete) can result in violating this constraint? Explain your answer for each modification statement briefly, but precisely.
- (b) Finish the following sql statement so that the trigger enforces this constraint. (Please note that it is impossible to create triggers without **execute procedure** in postgresql. PostgreSQL only allows the execution of a user-defined function for the triggered action.)

```
create trigger singleCollaboration
_____  
for each statement  
when ( _____  
    ) > 1  
signal sqlstate '...' ('Single collaboration constraint violated!');
```