Exercise 3
Date of issue: 14th March 2017 Deadline: 21st March 2017

The exercise is based on the *Mondial* database. Its schema can be accessed at [https://files.ifi.uzh.ch/dbtg/dbs/mondial/schema.pdf](https://files.ifi.uzh.ch/dbtg/dbs/mondial/schema.pdf)

1 Connection to PostgreSQL Server

1. Install pgAdmin following the tutorial provided on the course website[1]
2. Use the credentials of your PostgreSQL account that you received via email to connect to the server via pgAdmin
3. Execute first the create statements[2]and then the insert statements[3]that can be found on the course website

2 SQL – DDL

Consider the following three relation instances: table1, table2, and table3. The attributes that make up primary keys are underlined. In table2, attribute A is a foreign key to attribute A in table1; and attributes {C,D} in table2 form a foreign key to attributes {C,D} in table3.

<table>
<thead>
<tr>
<th>table1</th>
<th>table2</th>
<th>table3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>'a'</td>
<td>'a'</td>
<td>'x'</td>
</tr>
<tr>
<td>'b'</td>
<td>'x'</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>'x'</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'x'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'w'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'x'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'z'</td>
</tr>
</tbody>
</table>

1. Write `CREATE TABLE` statements to create the three tables. Infer suitable data types from the example instances. Include definitions of primary keys and foreign keys.

2. Does the order of the `CREATE TABLE` statements matter in your solution? Explain in one sentence!

3. Write `INSERT` statements to fill the tables with the example data.

4. Does the order of the `INSERT` statements matter in your solution? Explain in one sentence!

5. Given the three tables filled with the data from the example and given the mentioned key constraints,
   
   (a) is it allowed to insert tuple (‘a’, ‘x’, 2) in table2? Explain in one sentence!
   
   (b) is it allowed to insert tuple (‘b’, ‘x’, 2) in table2? Explain in one sentence!

[1] [https://files.ifi.uzh.ch/dbtg/dbs/pgAdmin/tutorial_pgAdminIII.pdf](https://files.ifi.uzh.ch/dbtg/dbs/pgAdmin/tutorial_pgAdminIII.pdf)
[2] [https://files.ifi.uzh.ch/dbtg/dbs/mondial/schema.sql](https://files.ifi.uzh.ch/dbtg/dbs/mondial/schema.sql)
[3] [https://files.ifi.uzh.ch/dbtg/dbs/mondial/inserts.sql](https://files.ifi.uzh.ch/dbtg/dbs/mondial/inserts.sql)
3 SQL – DML

For the following queries: (a) formulate and write down the query in SQL, (b) execute them on your Mondial database, and (c) write down the number of rows returned. Make sure your expressions are correct for all Mondial database instances.

1. The codes of all countries for which (a) the GDP is at least 90% composed of the Service and Industry sectors together, or (b) the inflation is lower than 2%. In Mondial, percentages are stored as NUMERICs in the range from 0 to 100.

2. The names and codes of all countries that gained independence in 20th century (i.e. when the year of independence is in between 1901 and 2000, bounds included)

3. The names of all cities with non-unique names. Return each city name only once. No aggregation allowed.

4. The names of all provinces with at least one million inhabitants through which exactly five rivers flow.

4 SQL – Understanding Queries

For each of the following queries explain briefly but precisely in natural language what they return independently of the actual data stored in the Mondial database. Include the result’s size and order in your description. In a second step, briefly explain the difference between the queries.

1. The LIMIT 1 clause in the following query limits the number of result rows to at most 1. See also PostgreSQL’s documentation.

   ```sql
   SELECT Name, Area
   FROM desert
   ORDER BY Area DESC, Name ASC
   LIMIT 1;
   ```

2. 

   ```sql
   SELECT Name, Area
   FROM desert
   WHERE Area = (SELECT MAX(Area) FROM desert)
   ORDER BY Name ASC;
   ```

3. 

   ```sql
   SELECT d1.Name, d1.Area
   FROM desert d1
   JOIN (SELECT Area FROM desert EXCEPT
           SELECT d3.Area
           FROM desert d3 JOIN desert d4 ON d3.Area < d4.Area
           ) d2 ON d1.Area = d2.Area
   ORDER BY d1.Name ASC;
   ```

   [https://www.postgresql.org/docs/current/static/queries-limit.html](https://www.postgresql.org/docs/current/static/queries-limit.html)