



Exercise Sheet 12: Application Programming II (until Thursday, 02.02.2017) (30 points)

Please note: you need **50%** of all exercise points to receive the *Studienleistung* for this lecture. In order to pass the RDB I Module, you need both the *Studienleistung* **and** you need to pass the exam. Exercises have to be turned in until **Thursday before the lecture** either in the lecture hall or into our mailbox at the IFIS floor (Mühlenpfordtstraße 23, 2nd floor). Please do not forget your **Matrikelnummer** and your **tutorial group number** on your solutions. **If you forget** to write your Matrikelnummer and/or your tutorial group number, you get **automatically 0 points**. Your solutions may be in German or English. Unless otherwise specified: **Always use your own words!**

Notes

This week we will be using Java JDBC API to create a database and implement some methods to perform some queries. The main goal of the exercise is to learn the JDBC API. Therefore, you should search the information and practice to learn it! Moreover, on the Web you can find several tutorials where you can find some working examples that might help you to complete this assignment.

1. Use PostgreSQL to setup your database. Use “qa_posts” as a database name, “ifisapp” as username and “rdbeinsoderzwei” as password.
2. We advise you to use a Java IDE (e.g. Eclipse, NetBeans), but feel free to use any other tool you might be familiar with.
3. After you finish the assignment you must send your source code by email to your tutor. Please note that your tutor must be able to inspect your source code easily and run it, so make sure you do the following:
 - a) Send everything needed to compile and run your program in one zip file.
 - b) Document your code using Java comments.
 - c) If there is something special that has to be done in order to run your program, please document it.

Exercise 12.1 JDBC Programming (30 points)

1. Download the csv files from http://www.ifis.cs.tu-bs.de/webfm_send/2143/app_prog_2.zip. Please manually inspect the three files to understand what the data is about.
2. Download the JDBC driver for PostgreSQL from <https://jdbc.postgresql.org/download.html> and then write a `getConnection()` **method** that returns a java **Connection** (2 points)
3. Write a method called `CreateDatabaseSchema` that executes all **CREATE TABLE** statements to create the tables that correspond with the csv files of Exercise 1. Please consider at least **PRIMARY** and **FOREIGN** key constraints. (4 points)

4. Write a method called **FillDatabase** that inserts all data from the csv files that you download from Exercise 1 into the database. It is expected that each csv file will correspond to a table in the database. (4 points)
5. Write a method called **showTitlesFromUserID**. The method should receive as a parameter the user Id. The method should print the list of the titles of all the posts of the given user and at the end of the list, the number of titles found. (4 points)
6. Write a method called **showStatisticsByCountry**. The method should print a list of tuples of the following form: (**country_name**, **numberOfUsers**). The value of **country_name** is the name of the country. The value of **numberOfUsers** is the total number of users in the given country. Given this list, your method should do the following: first, calculate its average and second, at the end of the list, print how many countries have more users than this average (4 points)
7. Write a method called **insertUser** that inserts a new user. The method should receive as parameters all the attributes of the users table. (4 points).
8. Write a method called **createUserLogTable**. The method should create a table called **users_log**. This table will store the id of a user and the timestamp when a new user is created. Write another version of the method **insertUserWithLog** that adds the following functionality: whenever you insert a new user your method will also perform the insertion of the id and timestamp in the table **users_log**. (4 points)
9. Write a method called **showMedianAgeperCountry**. The method should print a list with the name of each country and the median age of the users of that country. (4 points).

Exercise 12.2 SQL Trainer (0 points)

To practice SQL, students have programmed a SQL trainer. This can be tried under the following link:
<http://sql.academy.ifis.cs.tu-bs.de/>