SQL Lab: Assignment 5
(due to 07. / 08.02.2012)

General Information

This week, we will do some Java programming. For that task, we again recommend that you use a Java IDE (e.g Netbeans or Eclipse). However, you are free to use any tools for programming you prefer. Besides an IDE, you obviously will need a Java SE JDK (use version 6 or newer). After you got your IDE up and running, you should get the DB2 JDBC driver.

During the course of the exercise, you are to write a little Java program. Please send the .java files of your program via **e-mail** to your tutor. Please note that your tutor must be able to inspect your program easily – so take care that you package everything needed to **compile** and **run** it. If something special has to be done to compile or run your code, please **document** this. The resulting program should run on any platform! Also, you are required to **comment** sections of your program code (using Java comments) so that it is easy to understand by others (in particular by your tutor). In addition please hand in a short user manual for your program. If your code cannot be compiled or run easily or is not well documented/commented, you will receive the grade −1.

It is part of this exercise to look up all necessary commands and methods the lecture did not cover in depth. The easiest way for this probably is referring to the Java API documentation which can be found online. Moreover, many Java and JDBC tutorials can be found on the Web.

Tasks

To achieve the following tasks do the following steps:

- Use prepared statements wherever possible!
- Use the watch list tables to store your favorite documents in your database. **Always use transactions when inserting or removing data!** *Hint: Remember what we have learned about auto-commit in lecture 11.*
- For all **query tasks**, use the **DBLAB77** schema. Here you can find all information you should have imported last week. You can read this schema using your default group login.

- For all tasks, where you have to **store data**, please use **your schema**. **Remark: You do NOT have to change any of the tables you filled with data last week. Only empty tables will be touched!**
- Write a Java method establishing a connection to our DB2 database. The Connection object is to be returned by the method. The signature should look like this:

  public Connection *getDBLABConnection() **throws** SQLException*

  Hint: Set the retrieveMessagesFromServerOnGetMessage property to true during connection initialization in order to obtain verbose exceptions from the JDBC driver.

a) Create a program doing the following: After being started the user is asked to enter an author name. Then, all documents and the corresponding publication details of this author are returned showing the newest documents first, like the following example (plain console input/output is enough).

<div align="center">

DOCUMENT TITLE

by: AUTHOR NAMES

published in: JOURNAL NAME (PUBLICATION DATE)

</div>

b) Extended this program by the management of your favorite documents. The program has to provide the following features:
- Provide a login for the program to identify the current user. Use the user account information you inserted in the last assignment. For this task you have to use the useraccount table of your schema, not the DBLAB77 schema.
- Search and retrieve documents in the DBLAB77 schema by entering an author name, a document title or part of the title. Create a view including all needed information of your favorite documents. Use this view for the "show all favorite documents" feature.
- Declare documents from the result set as favorite document and store them in your watch list. Each favorite document should also contain a rank between 0 and 1.
- Remove documents from your favorite list.

c) Extend the program by the functionality of assigning tags to documents The program has to provide the following features:
- Login to the program to identify the current user.
- Search and retrieve documents like in b).

- Assign and remove a list of tags to documents within the result set. A tag can be private and public which mean private tags can only be seen by the user and public ones by everyone.
- Extend the output of a search by the assigned tags:

<div align="center">

DOCUMENT TITLE

by: AUTHOR NAMES

published in: JOURNAL NAME (PUBLICATION DATE)

private Tags: TAGS

public Tags: TAGS

</div>

Take care of public and private tags!

Do the following task to test your program:

- *Search for all document written by the author with the lastname "Katritzky" (16 documents should be found).*
- *Add the document "C-Cyanation with 1-cyanobenzotriazole" and "Acyl anion synthons: benzotriazole stabilized compared to classical" to your favorite list.*
- *Show your favorite list (2 entries should be included)*
- *Delete the document "Acyl anion synthons: benzotriazole stabilized compared to classical" from your favorite list.*
- *Add two private tags and two public tags to the remaining document.*
- *List your favorite documents (you should see 4 tags).*
- *Logout and login as tutor.*
- *Search for the remaining document and you should see only two tags in the result output.*

Important note: It is not necessary to build a GUI for your program. A text interface is also fine. You can use the Scanner class to read keyboard input. In all cases provide a small hand book how to use your program!