Institut für Informationssysteme
Technische Universität Braunschweig

Technische Universität Braunschweig
Institut für Informationssysteme
http://www.ifis.cs.tu-bs.de

Wolf-Tilo Balke, Christoph Lofi, and Joachim Selke

SQL Lab: Assignment 5

(due to 11.2.2010)

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 L4 driver from our website.

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. 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). 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.

Task

1. Exercise 1

a) Write a Java method establishing a connection to the DB2 database (running at jdbc:db2://is54.idb.cs.tu-bs.de:50000, port 50000, database DBLAB). Use your own username and password to log in. The connection object is to be returned by the method.
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 SQL driver

b) Write another method that reads and prints some metadata of the database. The following information should be printed (try to format it so that it can be read without getting an headache):
   - Database product name, including major and minor version
   - Number of available schemas and a list of all schema names
   - For the schema matching the current user name, a list of all available tables and for each table, a list of columns and their data type. Also, if the table has any indexes, they should be listed with their name and the column they are indexing.

   The method's signature should look like this:
   `public void printMetaData(Connection conn) throws SQLException`

c) Create a new program (which, of course, may use the connect method of exercise a) doing the following: After being started, the program asks the user to enter a year. Then, the number of movies which have been released in that year and the average number of actors and the average number of actresses (two averages, one for each gender) of those movies is returned. The program then asks for a new year and repeats until the user wants to quit. Use the IMDB schema. Keep it simple. Plain console input/output is enough. Hint: You can read console input using the class java.util.Scanner.

d) Write a small application that connects to your copy of the IMDB dataset (in your schema; see the previous assignment) and inserts new movies and actors as follows: In order to inform the wide public of the heroic deeds of all the various super hero teams, documentaries about these hero teams are supposed to be filmed. Each documentary carries the name of the team it features (e.g. "The Incredible Mutant Defenders"). Each documentary has a director and a producer. Also, each hero character (e.g. "The marvelous Captain SQL") is portrayed by an actor (e.g. "Dan Chamballan"). Usually, a super team involves between two and ten heroes.
   Your application should insert 15 of those documentaries. The names of persons (actors, directors, producers), heroes (the roles/characters portrayed in the movie), and super teams (i.e. movie titles) may be randomly generated using our wonderful super hero generator (downloadable from the SQL lab web page, or directly from http://www.ifis.cs.tu-bs.de/webfm_send/358). Add the jar file to your Java project and, for using it, call ifis.heroGenerator.SuperGeneratorFactory.getNewXXXGenerator (XXX may be replaced by Hero, Villain, HeroTeam, VillainTeam, FirstName, LastName, Power;

use the syntax completion of your IDE) to obtain a generator instance. The resulting SuperGenerator can generate any specialized names by calling generateNext().

Example for generating Hero names:

```
SuperGenerator heroGen = ifis.heroGenerator.SuperGeneratorFactory.getNewHeroGenerator();
String heroName1 = heroGen.generateNext();
String heroName2 = heroGen.generateNext();
```

Important note: Each single movie and all related persons should be inserted within a single **transaction** (i.e., all data related to movie is inserted or none, even if the system crashes in between)! By checking for returned values and handling exceptions, make sure that each transaction has been executed successfully. In case of any problems, print out an informative(!) message.