

Please consult the list of references p. 1, and then answer the following. Quiz #7, on Wednesday 25th April, will consist of questions taken from or inspired Parts I and II of this homework.

Part I — Questions

1. What are the technologies that makes it possible for a Java application to communicate with a DBMS?
2. What JDBC method do you call to get a connection to a database?
3. Briefly explain what the `next()` method from the `ResultSet` class does, and what is its return type.
4. How do you submit a **SELECT** statement to the DBMS?
5. Where is a `ResultSet` object's cursor initially pointing? How do you move the cursor forward in the result set?
6. Give three navigation methods provided by `ResultSet`.
7. Explain this JDBC URL format:
`jdbc:mysql://localhost:3306/HW_NewDB?createDatabaseIfNotExist=true&useSSL=true`
8. In what class is the `getColumnName()` method?
9. What is a prepared statement?

Part II — Problems

This part will mainly hands on, and ask you to read code. It is important that you learn how to get a working environment for a new technology quickly, for your understanding of this lecture, for the exam, and as a CS major. I'll assume that you will have successfully completed those tasks by Wednesday 25th April, so don't wait and let me know if you had difficulties solving them.

References

- This second part takes some inspiration from https://www.ntu.edu.sg/home/ehchua/programming/java/JDBC_Basic.html. If you experience troubles, <https://www.ntu.edu.sg/home/ehchua/programming/howto/ErrorMessage.html#JDBCErrors> might be a good read.
- Section 13.3.2 of your textbook is a condensed, but good read. Many textbook on Java includes a part on Databases, cf. for instance the Chapter 16 of *Starting Out with Java: Early Objects* (5th Edition) by Tony Gaddis.

Problem 1

In the archive, navigate to `code/sql/`, open and read `HW_ebookshop.sql`.

Then, open a terminal (or command-line interpreter), navigate to the folder where you stored that file (using `cd`), and type

```
mysql -u testuser -p < HW_ebookshop.sql
```

for linux, or (something like)

"C:\Program Files\MySQL\MySQL Server 5.7\bin\mysql.exe" -u testuser -p < HW_ebookshop.sql
for Windows.

You just discovered MySQL's batch mode, that perform *series* of instructions from a file. You can easily make sure that the database and the table were indeed created, and the values inserted.

Problem 2

This exercise supposes you successfully completed Problem 1. We will compile and execute your first database application, using Java and MySQL.

- I will assume that you have MySQL installed and set-up as indicated in Homeworks #1 and #2.
- I will assume that you have Java installed. If not, please refer to <http://spots.augusta.edu/caubert/teaching/general/java/> for a simple program and the instructions to compile and execute it.
- We need to set up the *driver* (or *connector*) to make the java sql API and MySQL communicate. To do so,
 - Go to <https://dev.mysql.com/downloads/connector/j/>
 - Click on "Download" in front of "Platform Independent (Architecture Independent), ZIP Archive"
 - Look for the (somewhat hidden) "No thanks, just start my download."
 - You will download a file named "mysql-connector-java-***.zip", where *** is the version number.
 - Upon completion of the download, unzip the file, and locate the "mysql-connector-java-***-bin.jar" file.
 - Copy that file in code/java/.

- Open a terminal in that same folder, and compile `FirstProg.java`, using

```
javac FirstProg.java
```

(or an equivalent command for windows). Normally, nothing will be printed, but a `FirstProg.class` file will be created.

- Now, execute that program, using

```
java -cp .:mysql-connector-java-***-bin.jar FirstProg
```

in Linux, or

```
java -cp .;mysql-connector-java-***-bin.jar FirstProg
```

in Windows. The `-cp` option lists the places where java should look for the class used in the program: we are explicitly asking java to use the `mysql-connector-java-***-bin.jar` executable to execute our `FirstProg` executable. Try to execute `FirstProg` without that flag, and see what happens.

Problem 3

There are three other programs in that archive:

Variation.java is the program we looked at during class.

NullProg.java is an attempt to answer the question whenever SQL's **NULL** is JAVA's **NULL** when it comes to strings.

AdvancedProg.java is a long and commented program that introduces multiple new tools and ideas.

Read, execute, break, edit, compile, patch, hack and (most importantly) understand those three programs, with, of course, a focus on the last one.

