Lab 7 Amended set up and Exercise 1 - 29/11/2018

The aim of this lab is to introduce JDBC, the Java API for connecting to databases.

Copy the files **Lab7.java**, **orderdb.sql**, **custs.txt**, and **orders.txt** into a new folder on your **M:** drive. [Note that **orderdb.sql** will arrive as a text file since the course materials server will not allow retrieval of files with a **.sql** extension - you will need to rename it after saving it.]

For this exercise you will need mysql:jdbc driver,

Use The following driver available on Moodle: JDBC Driver 5.1.47

Add it to your InteliJ module:

- right click on your module name and select **module settings**
- go to **platform settings** and then **global libraries**
- click the plus sign on top name the library "mysql"
- click **attach jar directories** and add the path of your jar file (the one you have just downloaded)

EXERCISE 1

Lab7.java contains an incomplete program to query a database. Try compiling and running it (Note the comment in the Java file regarding the MySQL password). You should get an error message since the database does not exist. Check that the program produced output to say that it successfully loaded the JDBC driver (if it failed to do so you'll have to adjust your class path - but this should not be necessary as long as you're using the standard lab set-up).

In order to create a database we need to start up the mySQL server so open a command prompt window (from the **Programs-Accessories** menu)

Unless you already have a mySQL area on your M drive give the command from the M prompt:

```
C:\mysql\copy.bat
```

(This needs to be done once only - there's no need to do it again next time you want to use mySQL.)

Now start up the mySQL server by typing the command

```
mysqld –standalone --local-infile
```

Nothing should happen - the server is running so another prompt will not appear.

Leave this window open for the duration of the lab (you can minimise it to an icon but if you close it the server will stop running)

Open another command prompt window and change directory to the folder containing your copies of the supplied files.

The file **orderdb.sql** contains an SQL script to create a database and two tables.

Examine it using TextPad and then give the command

```
mysql -u root --local-infile < orderdb.sql
```

to run it.

(Note: You might need to use the -p flag together with your password if you have previously set the MySQL password. If you have set it and do not remember it, then you might need to reset it, to do that have a look at the instructions.)

Run the java program again - you should not get an error message this time.

In the command prompt window give the command

```
mysql -u root --local-infile
```

to enter an interactive mySQL session.

You should see an introductory message and a `mysql>` prompt.

Type the command

```
USE orderdb
```

and the query

```
SELECT * FROM customers;
```

You will find that there are no entries in the customer table - the script did not populate the tables.

The files **custs.txt** and **orders.txt** contain some data to be entered into the tables. Examine them in TextPad.

Type the commands

```
 LOAD DATA LOCAL INFILE 'custs.txt' INTO TABLE
customers;
```

and

```
 LOAD DATA LOCAL INFILE 'orders.txt' INTO TABLE orders;
```

to load the data from the files into the database then repeat the `SELECT` query above. Use a similar `SELECT` query to examine the orders table

If there is still an issue with loading the data to the database, then type the following command at the mysql prompt:

```
mysql> SET GLOBAL local_infile = 'ON';
Query OK, 0 rows affected (0.06 sec)
```

Check it is set to on using the following command:

```
mysql> SHOW GLOBAL VARIABLES LIKE 'local_infile';

+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | ON    |
+---------------+-------+
1 row in set (0.00 sec)
```

Close the mySQL session by typing

```
 EXIT
```

You may now close this command prompt window (but not the one in which the server is running).