

CNT 4714 – Project Three – Spring 2019

Title: “Project Three: Two-Tier Client-Server Application Development With MySQL and JDBC”

Points: 100 points

Due Date: Sunday March 10, 2019 by 11:59 pm (WebCourses Time)

Objectives: To develop a two-tier Java based client-server application interacting with a MySQL database utilizing JDBC for the connectivity. This project is designed to give you some experience using the various features of JDBC and its interaction with a MySQL DB Server environment.

Description: In this assignment you will develop a Java-based GUI front-end (client-side) application that will connect to your MySQL server via JDBC.

You are to develop a Java application that will allow any client (the end-user) to execute commands against the database. You will create a Java GUI-based application front-end that will accept any MySQL DDL or DML command, pass this through a JDBC connection to the MySQL database server, execute the statement and return the results to the client. Note that while technically your application must be able to handle any DDL or DML command, we won't actually use all of the commands available in these sublanguages. For one thing, it would be quite rare to allow a client to create a database or a table within a database. Note too, that the only DML command that uses the `executeQuery()` method of JDBC is the Select command, all other DML and DDL commands utilize `executeUpdate()`. Some screen shots of what your Java GUI front-end should look like are shown below. Basically, this GUI is an extension of the GUI that was developed in the lecture notes and is available on WebCourses as `DisplayQueryResults.java`. Your Java application must give the user the ability to execute any SQL DDL or DML command for which the user has the correct permissions. Note also, that if the user wishes to change databases in the middle of a session, they must reconnect to the new database. Their user information can remain in the proper window, but you must click the reconnect button to establish a connection to the new database. You do not need to support simultaneous connections from your application to more than one database in this assignment. However, you will be able to start multiple instances of your Java application and allow different clients to connect simultaneously to the MySQL server, since the default number of connections is set at 151 (see your Workbench options file under the networking tab).

Once you've created your application, you will execute a sequence of DML and DDL commands and illustrate the output from each in your GUI for two different users. For this project you will create, in addition to the root user, a client user with limited permissions on the database (see below). The root user is assumed to have all permissions on the database, any command they issue will be executed. The client user will be far more restricted.

References for this assignment:

Notes: Lecture Notes for MySQL and JDBC.

Input Specification:

The **first step** in this assignment is to login to the MySQL Workbench as the root user and execute/run the script to create and populate the backend database. This script is available on the assignment page and is named “`project3dbscript.sql`”. This script creates a database named **project3**. You can use the MySQL Workbench for this step, or the command line whichever you prefer.

The **second step** is to create authorizations for a client user (in addition to the root user) named `client`. By default your root user has all permissions on the **project3** database. Use either SQL Grant statements from the command line or the MySQL Workbench (see separate document for details on how to accomplish this task) to check and set permissions for the client as follows:

Register the new user named **client** (assign them the password *client* – ignore the MySQL warning on weak password setting) and assign to this user only selection privileges on the **project3** schema.

Output Specification: There are two parts for the output for this project. Part 1 is to provide screen shots from your application which clearly show the complete query/command expression and results for each of the commands that appear in the script named: **project3rootuserscript.sql** available on the course website. There are eight different commands in this script and some of the commands will have more than one output capture (see below). Part 2 is to provide screen shots from your application which clearly show the complete query/command expression and results for each of the commands that appear in the script named: **project3clientuserscript.sql** available on the course website. There are three different commands in this script and some of the commands will have more than one output capture (see below). To produce your final output, first recreate the database, then run the root user commands followed by the client commands.

Deliverables:

Zip up all of the .java files associated with your application as well as the screen shots from each of the commands specified in both the **project3rootuserscript.sql** and **project3clientuserscript.sql** files via WebCourses no later than 11:59pm Sunday March 10, 2019. Be sure to clearly label each screen shot.

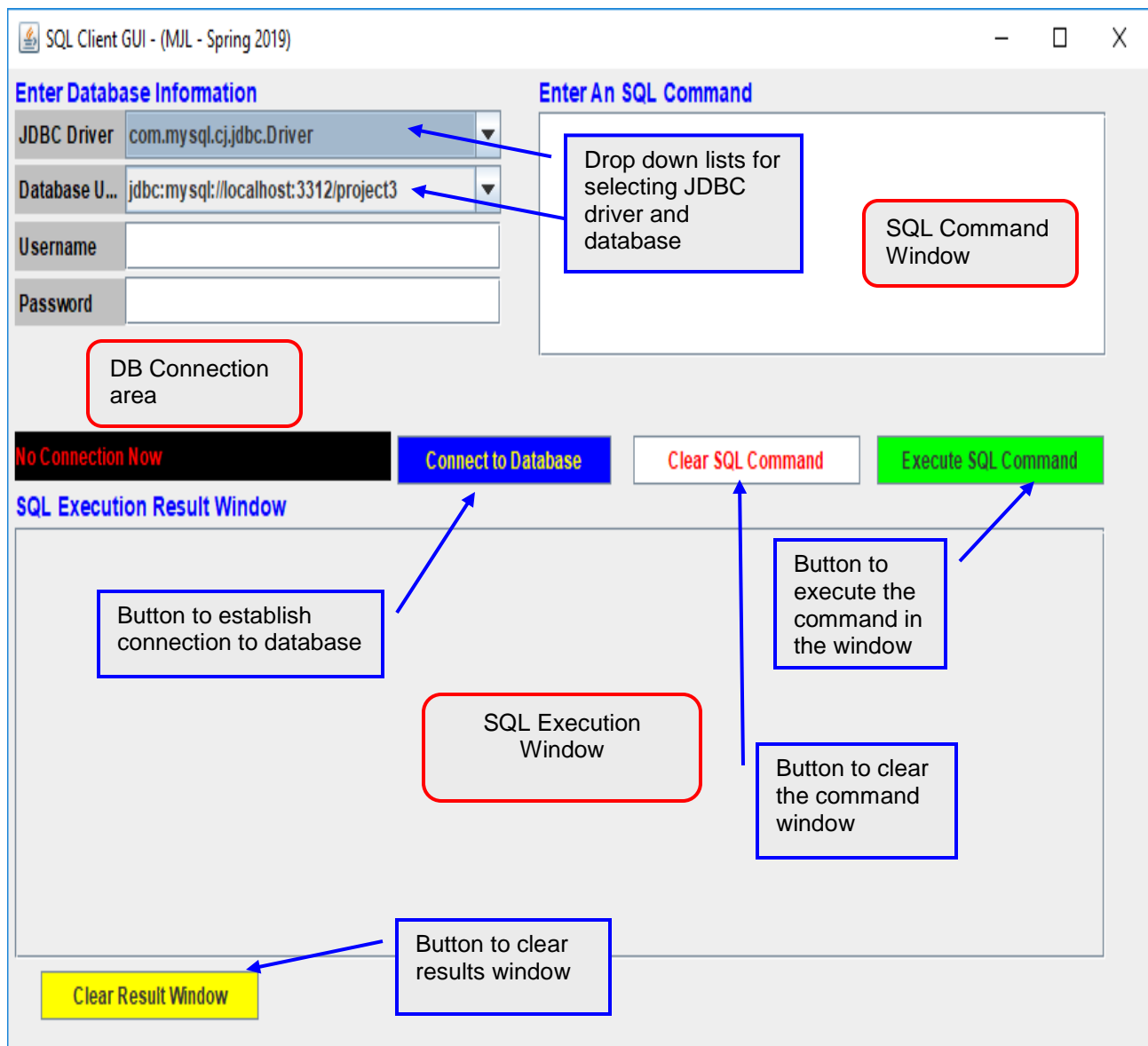
Details:

Shown below is a screen shot of the initial GUI. Notice that there are drop-down lists for selecting the JDBC driver and database URL that the client must select. The client must also specify a username and password (MySQL option) before connecting to the database.

You should provide buttons for the user to clear the command window as well as the result window. The status of the connection should be returned to the GUI and displayed in the connection area.

The output of all SQL commands should be returned to the SQL Execution Result window. Please note that only SQL commands can be executed via this application, we will not go to the effort of making the application display the results of MySQL-specific commands. (When a MySQL-specific command is executed, the SQL Execution Result window does not need to display any results, if you wanted to you could display the line “MySQL command executed” in the results window, but this is not required.)

Note that for non-query DML and DDL commands, before and after screen shots must be taken to illustrate the basic effect of the command. See pages 7-8 for an illustration of this.



The GUI areas defined.

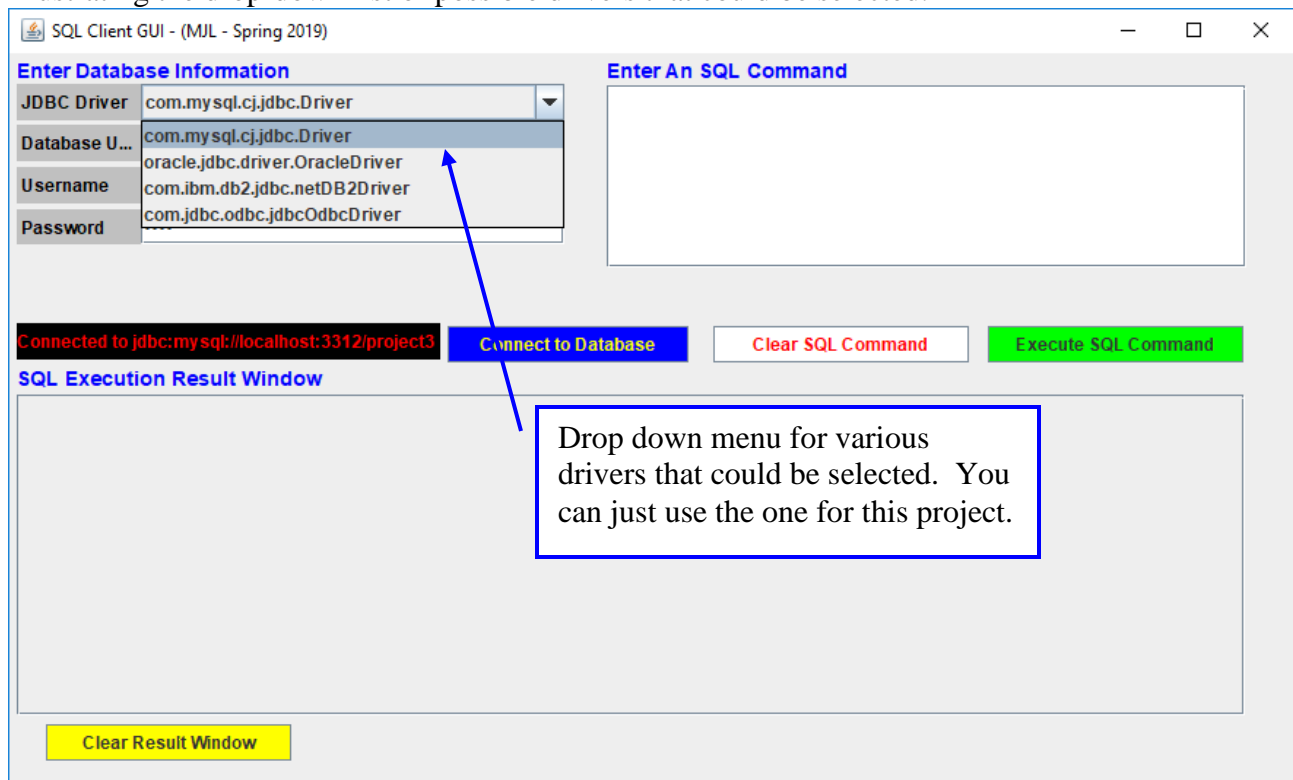
Screen shot illustrating an initial client connection.

The screenshot displays the 'SQL Client GUI - (MJL - Spring 2019)' window. It is divided into several sections:

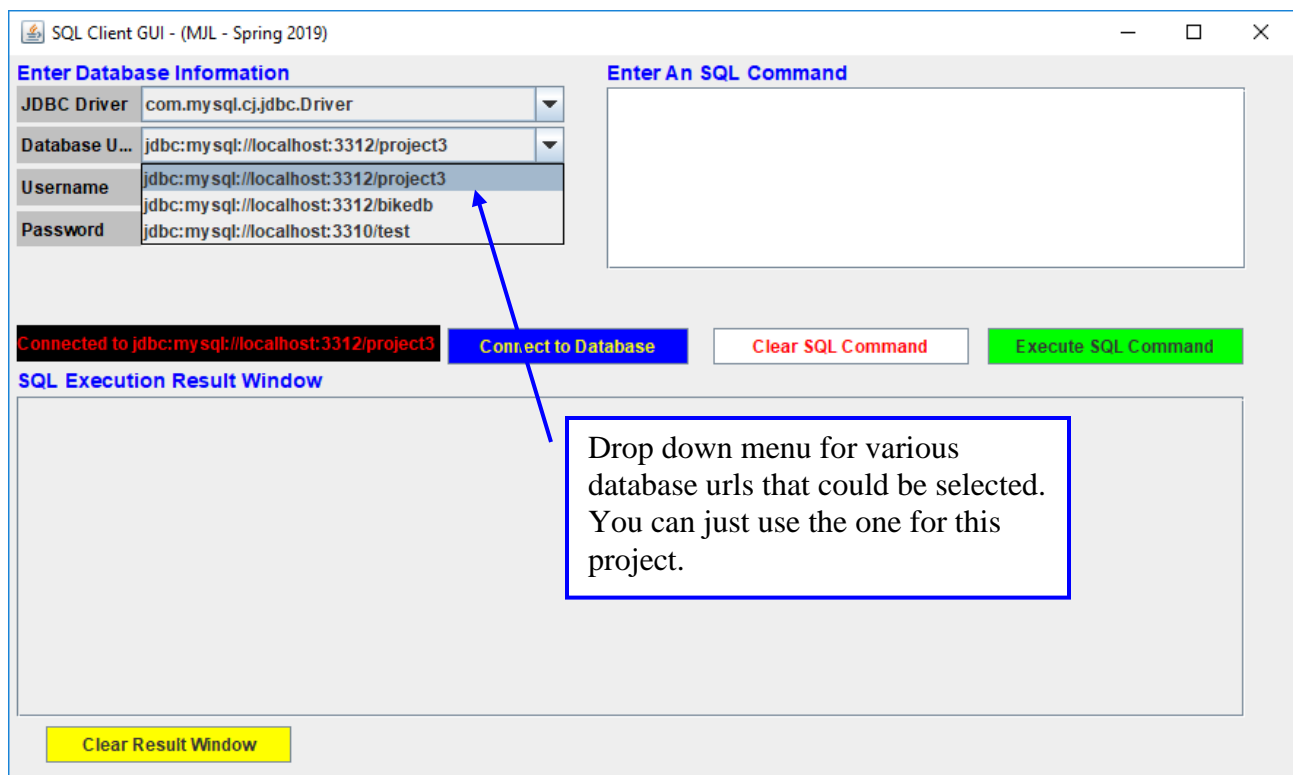
- Enter Database Information:** Contains four input fields: 'JDBC Driver' (set to 'com.mysql.cj.jdbc.Driver'), 'Database U...' (set to 'jdbc:mysql://localhost:3312/project3'), 'Username' (set to 'root'), and 'Password' (masked with '****').
- Enter An SQL Command:** A large text area for entering SQL queries.
- Connection Status:** A black box displays 'Connected to jdbc:mysql://localhost:3312/project3' in red text.
- Buttons:** Includes 'Connect to Database' (blue), 'Clear SQL Command' (white with red text), 'Execute SQL Command' (green), and 'Clear Result Window' (yellow).
- SQL Execution Result Window:** A large area for displaying query results, currently empty.

A blue arrow points from a text box to the connection status message. The text box contains the text: 'Connection established to selected database URL'.

Illustrating the drop-down list of possible drivers that could be selected.



Illustrating the drop-down list of possible database URLs available.



User has connected to a database and issued a select command. Results are displayed in the SQL Execution window.

The screenshot shows the SQL Client GUI with the following details:

- Enter Database Information:**
 - JDBC Driver: com.mysql.cj.jdbc.Driver
 - Database U...: jdbc:mysql://localhost:3312/project3
 - Username: root
 - Password:
- Enter An SQL Command:** select * from riders
- Status:** Connected to jdbc:mysql://localhost:3312/project3
- Buttons:** Connect to Database, Clear SQL Command, Execute SQL Command
- SQL Execution Result Window:**

ridername	teamname	nationality	num_pro_wins	gender
Alberto Contador	Astana	Spain	21	M
Alessandro Ballan	Lampre	Italy	21	M
Andy Schleck	Leopard-Trek	Luxembourg	35	M
Bradley Wiggins	Ti-Raleigh	Great Britain	13	M
Chris Froome	Sky	Great Britain	23	M
Dietrich Thurau	Ti-Raleigh	Germany	78	M
Elisa Borghini	Schenger	Italy	34	F
Fabian Cancellara	SaxoBank	Switzerland	58	M
Fedor den Hertog	Acqua & Sapone	Netherlands	20	M
Frank Schleck	Leopard-Trek	Luxembourg	28	M
George Hincapie	BMC	USA	22	M
Jens Voigt	SaxoBank	Germany	38	M

A red arrow points to the 'num_pro_wins' column header in the result table. A red-bordered box contains the text: "Note the metadata. Your application must print this for the user."

A more complicated query:

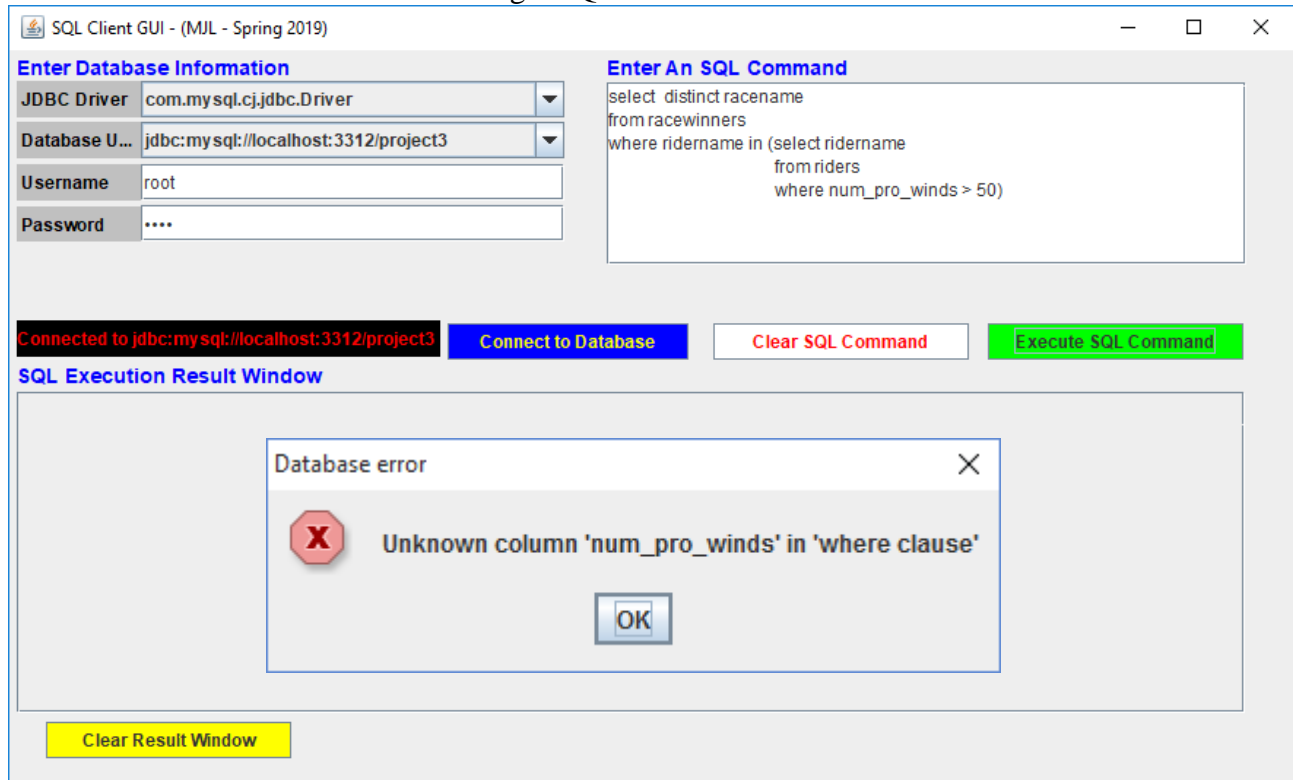
The screenshot shows the SQL Client GUI with the following details:

- Enter Database Information:**
 - JDBC Driver: com.mysql.cj.jdbc.Driver
 - Database U...: jdbc:mysql://localhost:3312/project3
 - Username: root
 - Password:
- Enter An SQL Command:**

```
select distinct racename
from racewinners
where ridername in (select ridername
                    from riders
                    where num_pro_wins > 50)
```
- Status:** Connected to jdbc:mysql://localhost:3312/project3
- Buttons:** Connect to Database, Clear SQL Command, Execute SQL Command
- SQL Execution Result Window:**

racename
Amstel Gold
Fleche Wallone - Feminine
GP-E3
Liege-Bastogne-Liege
Paris-Roubaix
Rund de Flandren
World Championship - Elite Women

When the user makes a mistake entering a SQL command:



The following two screen shots illustrate that your application should be able to handle non-query commands from the users.

Before screen shot of a subset of the riders relation:

The screenshot shows a window titled "SQL Client GUI - (MJL - Spring 2019)". It is divided into several sections:

- Enter Database Information:** Contains four fields: "JDBC Driver" (com.mysql.cj.jdbc.Driver), "Database U..." (jdbc:mysql://localhost:3312/project3), "Username" (root), and "Password" (masked with four dots).
- Enter An SQL Command:** A text area containing the SQL query:

```
select *  
from riders  
where nationality = "Germany"
```
- Buttons:** A row of four buttons: "Connected to jdbc:mysql://localhost:3312/project3" (black with red text), "Connect to Database" (blue), "Clear SQL Command" (white with red text), and "Execute SQL Command" (green).
- SQL Execution Result Window:** A table displaying the results of the query. It has five columns: "ridername", "teamname", "nationality", "num_pro_wins", and "gender".
- Clear Result Window:** A yellow button at the bottom left of the result window.

ridername	teamname	nationality	num_pro_wins	gender
Dietrich Thurau	Ti-Raleigh	Germany	78	M
Jens Voigt	SaxoBank	Germany	38	M

Insert command issued:

SQL Client GUI - (MJL - Spring 2019)

Enter Database Information

JDBC Driver: com.mysql.cj.jdbc.Driver
Database U...: jdbc:mysql://localhost:3312/project3
Username: root
Password: ****

Enter An SQL Command

```
insert into riders  
values ("Heinrich Haussler", "Sky", "Germany", 12, "M")
```

Connected to jdbc:mysql://localhost:3312/project3

Connect to Database Clear SQL Command Execute SQL Command

SQL Execution Result Window

Clear Result Window

After screen shot of subset of riders relation after insert command was issued:

SQL Client GUI - (MJL - Spring 2019)

Enter Database Information

JDBC Driver: com.mysql.cj.jdbc.Driver
Database U...: jdbc:mysql://localhost:3312/project3
Username: root
Password: ****

Enter An SQL Command

```
select *  
from riders  
where nationality = "Germany"
```

Connected to jdbc:mysql://localhost:3312/project3

Connect to Database Clear SQL Command Execute SQL Command

SQL Execution Result Window

ridename	teamname	nationality	num_pro_wins	gender
Dietrich Thurau	Ti-Raleigh	Germany	78	M
Heinrich Haussler	Sky	Germany	12	M
Jens Voigt	SaxoBank	Germany	38	M

Clear Result Window

Screen shot illustrating the client user issuing a command for which they do not have permission:

