

CNT 4714 – Programming Assignment #3 – Spring 2014

Title: “Program Assignment 3: Two-Tier Client-Server Application Development With MySQL and JDBC”

Points: 100 points

Due Date: Tuesday March 11, 2014 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 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 the code page 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 we have set the database server up to allow up to 10 concurrent connections.

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 this project we will assume that the only clients are the root user. Since the root user is assumed to have all permissions on the database, any command they issue will be executed. We'll create additional clients with varying permissions in the next project.

References for this assignment:

Notes: Lecture Notes for MySQL and JDBC.

Input Specification: Create your database using the script named `prog3dbscript.sql` which you can download from the assignments page. Your application does not need to support the execution of scripts (you can execute the script using the MySQL Workbench tool). We will expect the user to enter each SQL command in the command window and click the execute button for each command entered.

Output Specification: 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 **prog3userscript.sql** available on the course website.

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 the **prog3userscript.sql** file via WebCourses no later than 11:59pm Tuesday March 11th . Be sure to clearly label each screen shot.

Details:

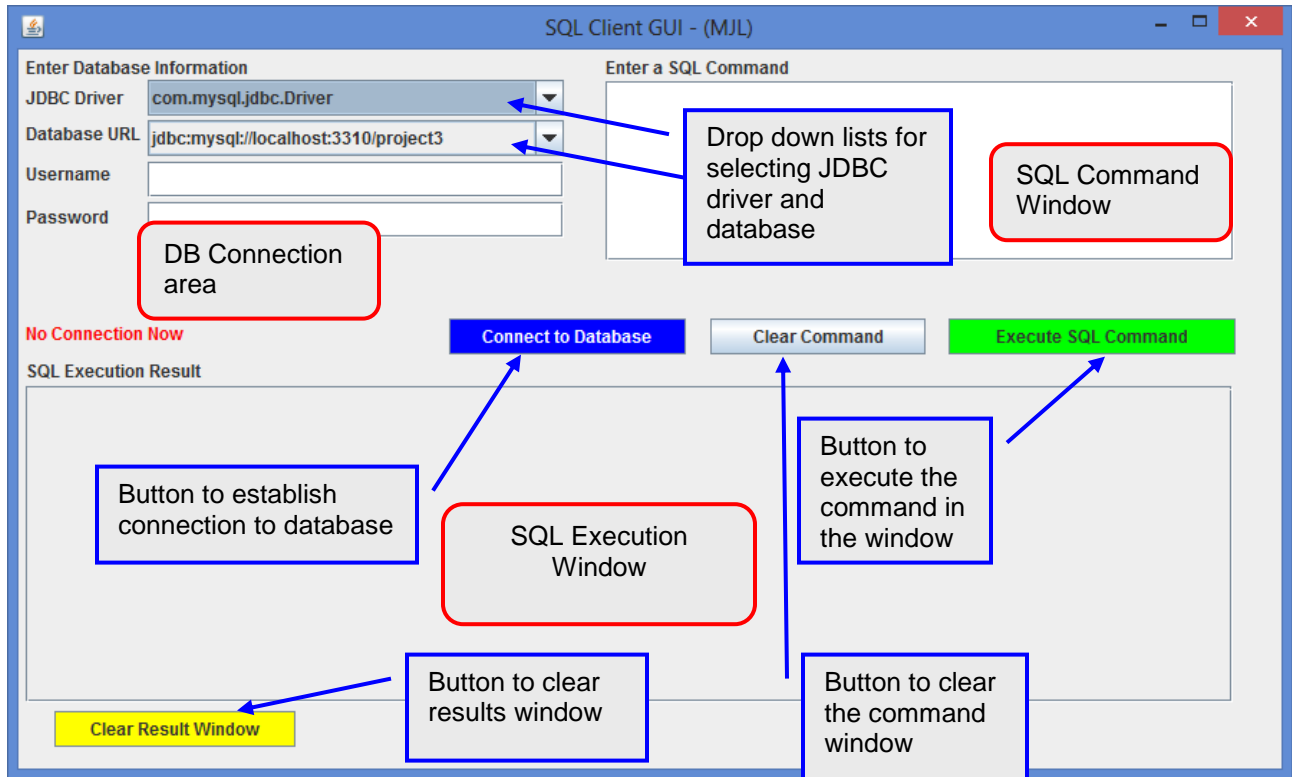
Shown below is a screen shot of the initial GUI. Notice that there are drop-down lists for selecting the JDBC driver and database URL that the client must select. The client must also specify a username and password (MySQL option) before connecting to the database.

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.

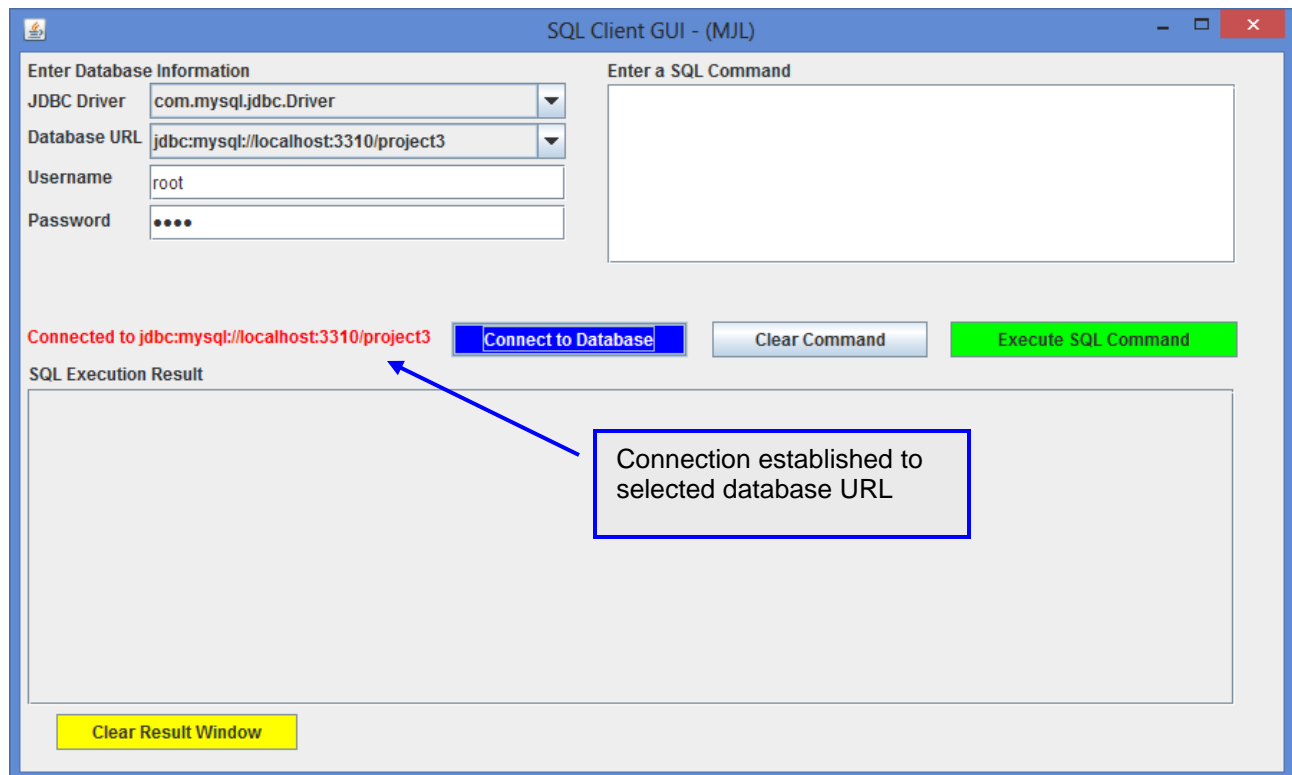
The output of all SQL commands should be returned to the SQL Execution Result window. Please note that only SQL commands can be executed via this application, we will not go to the effort of making the application display the results of MySQL-specific commands. (When a MySQL-specific command is executed, 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.

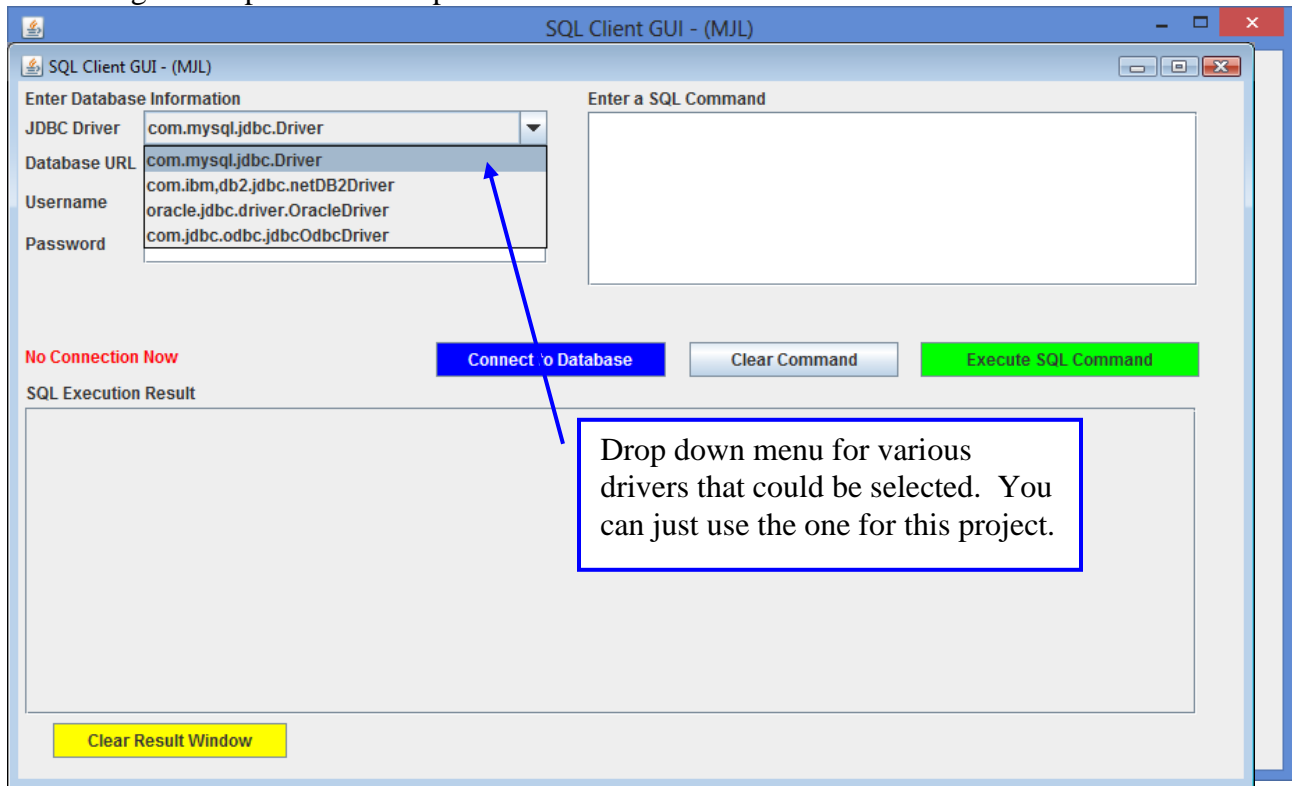
The GUI areas defined.



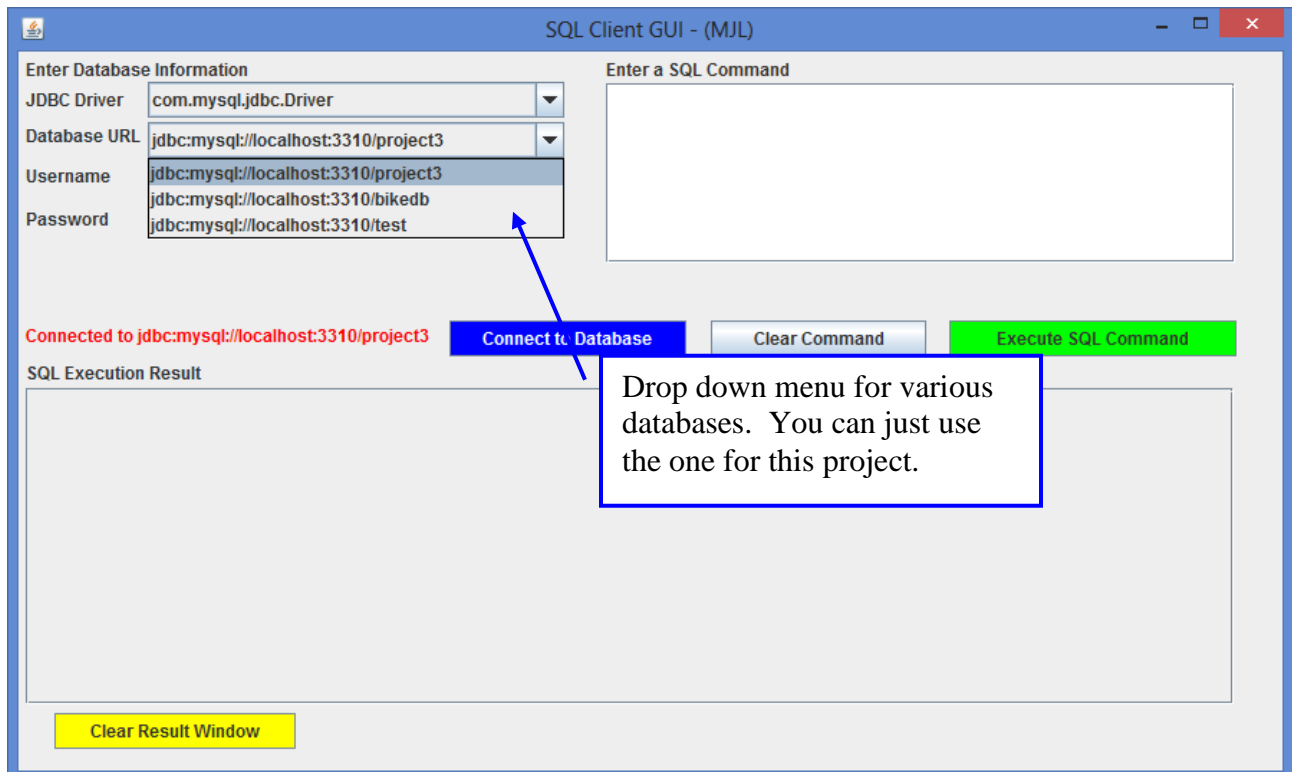
Screen shot illustrating an initial client connection.



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.jdbc.Driver; Database URL: jdbc:mysql://localhost:3310/project3; Username: root; Password: masked.
- Enter a SQL Command:** select * from riders
- Buttons:** Connect to Database (blue), Clear Command (grey), Execute SQL Command (green).
- SQL Execution Result:** A table with columns: ridername, teamname, nationality, num_pro_wins. The table contains 14 rows of data.
- Callout Box:** A red-bordered box with a red arrow pointing to the 'nationality' column header, containing the text: "Note the metadata. Your application must print this for the user."

ridername	teamname	nationality	num_pro_wins
Alberto Contador	Astana	Spain	21
Allesandro Ballan	Lampre	Italy	21
Andy Schleck	Leopard-Trek	Luxemborg	35
Bradley Wiggins	Ti-Raleigh	Great Britain	13
Dietrich Thurau	Ti-Raleigh	Germany	78
Fabian Cancellara	SaxoBank	Switzerland	58
Fedor den Hertog	Acqua & Sapone	Netherlands	20
Frank Schleck	Leopard-Trek	Luxemborg	28
George Hincapie	BMC	USA	22
Jens Voigt	SaxoBank	Germany	38
Johan Museeuw	Quick-Step	Belgium	120
Mario Cipolini	Acqua & Sapone	Italy	130

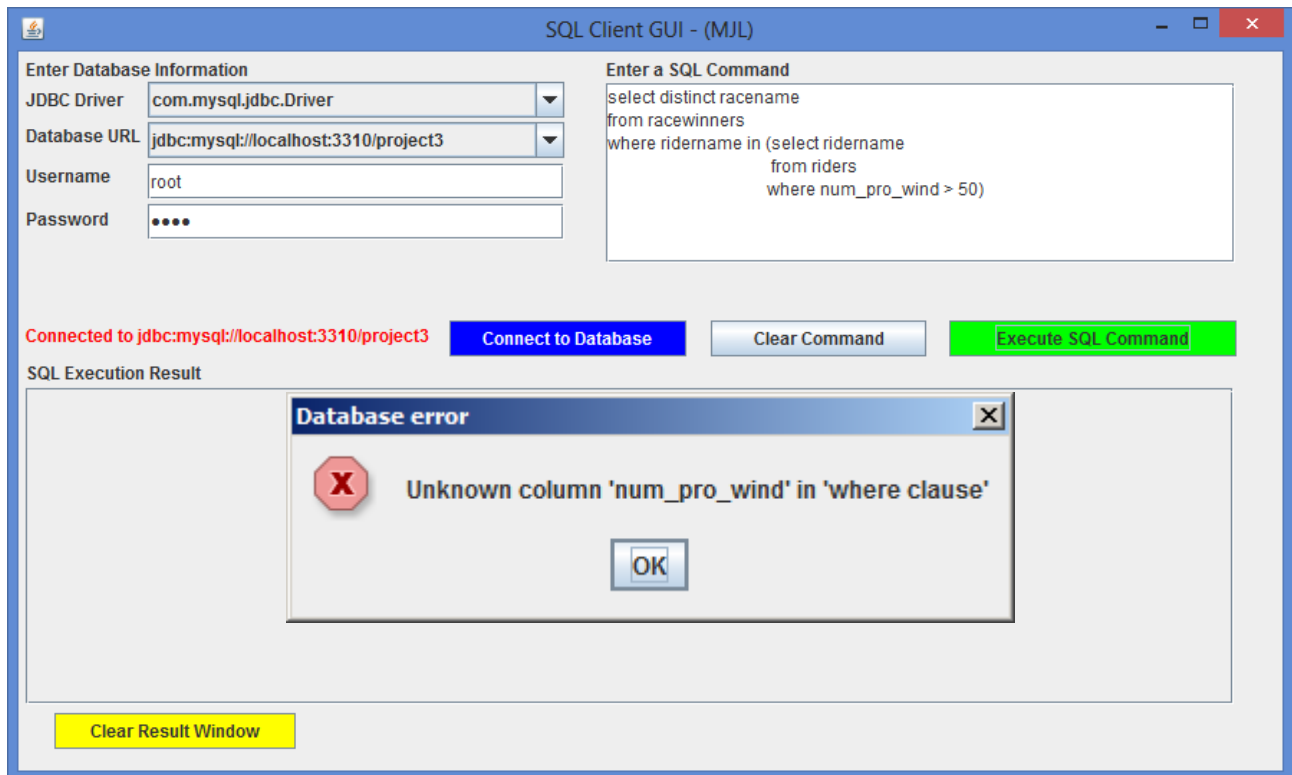
A more complicated query:

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

- Enter Database Information:** JDBC Driver: com.mysql.jdbc.Driver; Database URL: jdbc:mysql://localhost:3310/project3; Username: root; Password: masked.
- Enter a SQL Command:** select distinct racename from racewinners where ridername in (select ridername from riders where num_pro_wins > 50)
- Buttons:** Connect to Database (blue), Clear Command (grey), Execute SQL Command (green).
- SQL Execution Result:** A table with column: racename. The table contains 5 rows of data.

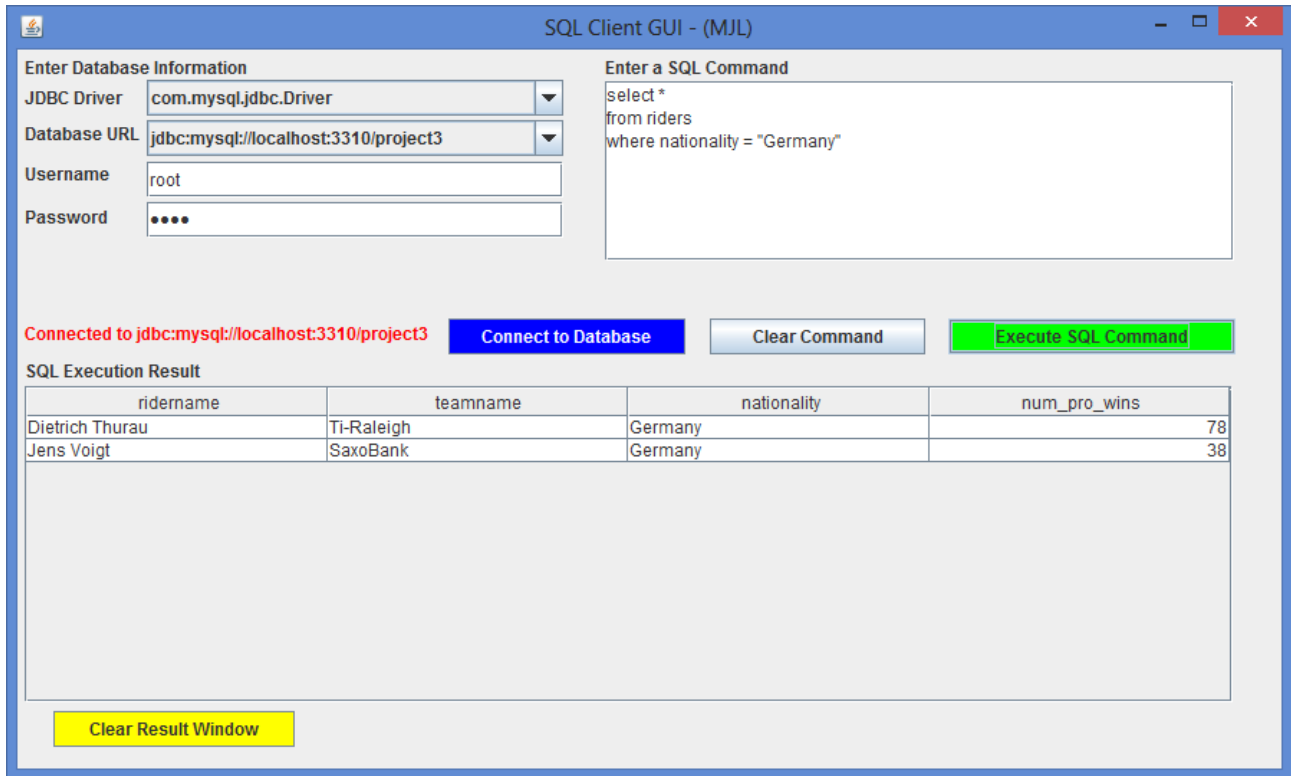
racename
Amstel Gold
GP-E3
Liege-Bastogne-Liege
Paris-Roubaix
Rund de Flandren

When the user makes a mistake entering a SQL command:

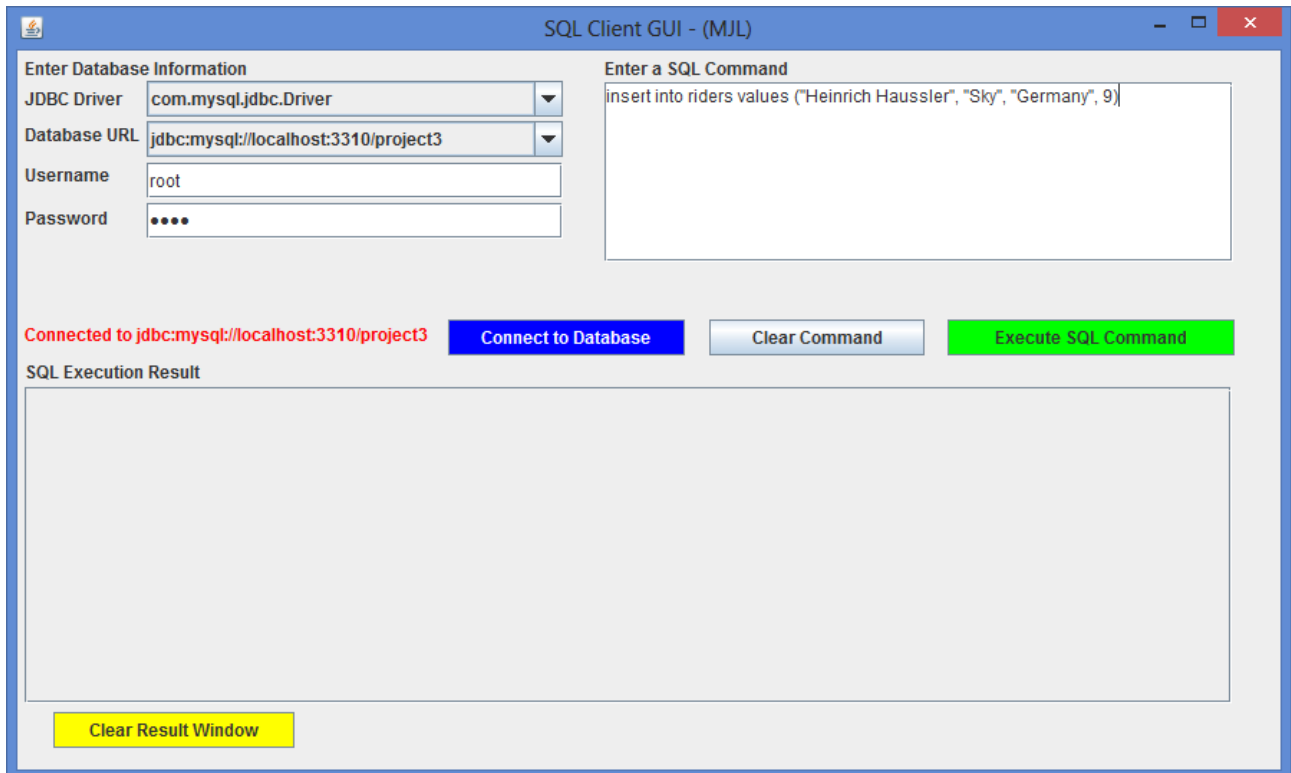


The following two screen shots illustrate that your application should be able to handle non-query commands from the users.

Before screen shot of a subset of the riders relation:



Insert command issued:



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

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

- Enter Database Information:**
 - JDBC Driver: com.mysql.jdbc.Driver
 - Database URL: jdbc:mysql://localhost:3310/project3
 - Username: root
 - Password: ****
- Enter a SQL Command:**

```
select *  
from riders  
where nationality = "Germany"
```
- Buttons:** Connect to Database (blue), Clear Command (light blue), Execute SQL Command (green)
- Connected to:** jdbc:mysql://localhost:3310/project3
- SQL Execution Result:**

ridername	teamname	nationality	num_pro_wins
Dietrich Thurau	Ti-Raleigh	Germany	78
Heinrich Haussler	Sky	Germany	9
Jens Voigt	SaxoBank	Germany	38
- Clear Result Window:** (yellow button)