

CNT 4714 – Project Two – Summer 2016

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

Points: 100 points

Due Date: Sunday June 19, 2016 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 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 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 “`project2dbscript.sql`”. This script creates a database named **project2**. 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 **project2** 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 **client2** (assign them the password *client* – ignore the MySQL warning on weak password setting) and assign to this user only selection privileges on the **project2** 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: **project2rootuserscript.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: **project2client2userscript.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 **project2rootuserscript.sql** and **project2client2userscript.sql** files via WebCourses no later than 11:59pm Sunday June 19, 2016. Be sure to clearly label each screen shot.

Details:

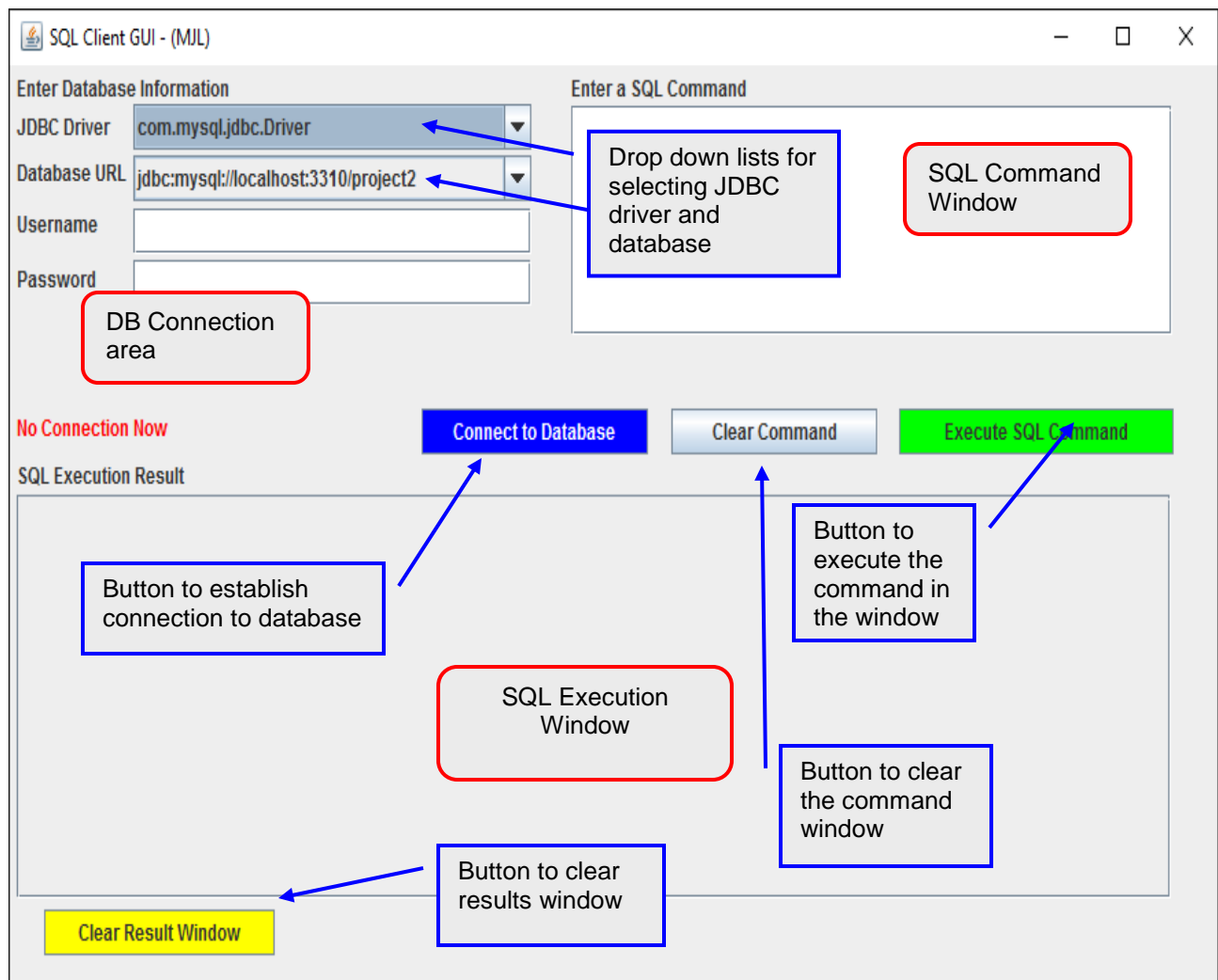
Shown below is a screen shot of the initial GUI. Notice that there are drop-down lists for selecting the JDBC driver and database URL that the client must select. The client must also specify a username and password (MySQL option) before connecting to the database.

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.

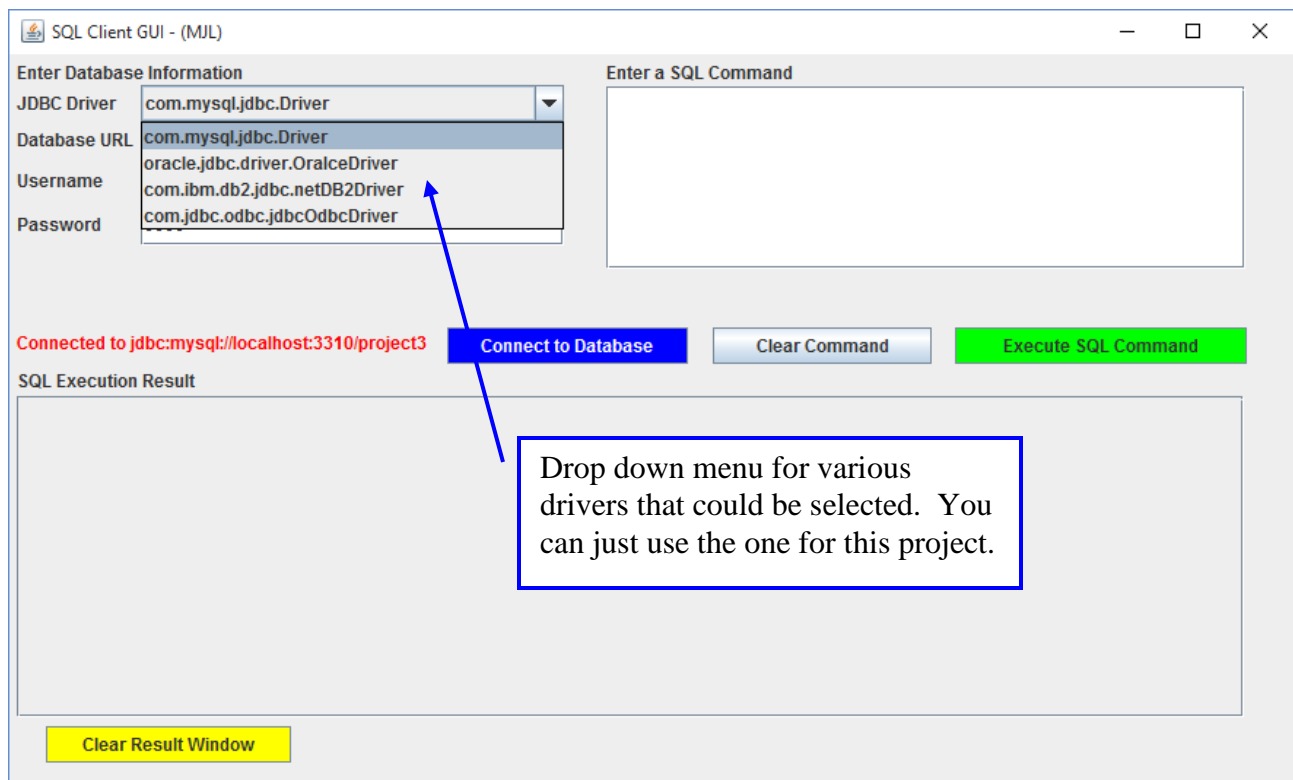
The GUI areas defined.



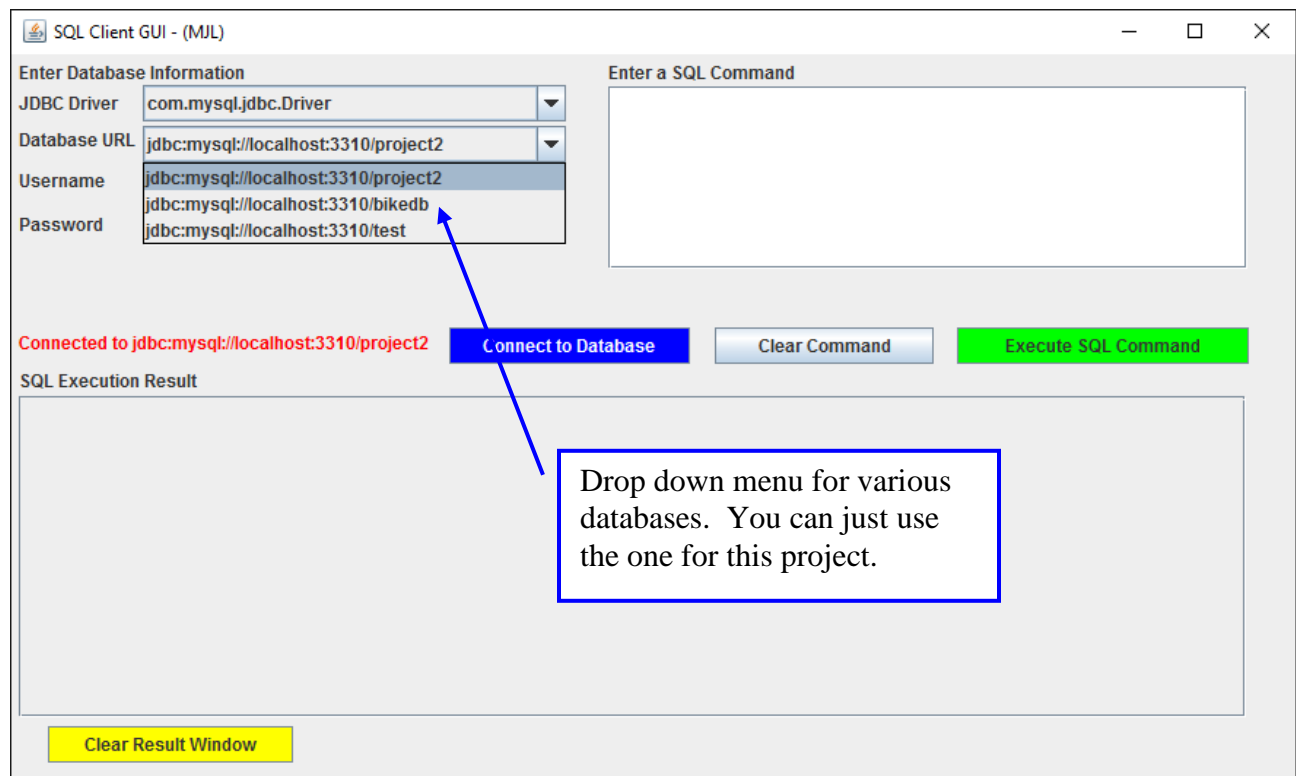
Screen shot illustrating an initial client connection.

The screenshot displays the 'SQL Client GUI - (MJL)' window. It features a 'Enter Database Information' section with fields for 'JDBC Driver' (set to 'com.mysql.jdbc.Driver'), 'Database URL' (set to 'jdbc:mysql://localhost:3310/project2'), 'Username' (set to 'root'), and 'Password' (masked with four dots). To the right is a large text area for 'Enter a SQL Command'. Below these fields, a red status message reads 'Connected to jdbc:mysql://localhost:3310/project2'. Three buttons are present: 'Connect to Database' (blue), 'Clear Command' (light blue), and 'Execute SQL Command' (green). A blue arrow points from a callout box containing the text 'Connection established to selected database URL' to the red status message. At the bottom, there is a 'SQL Execution Result' section with a large empty text area and a 'Clear Result Window' button (yellow).

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 - (MJL)' window. It has a title bar with standard window controls. The main area is divided into several sections:

- Enter Database Information:** Contains four fields: 'JDBC Driver' (set to 'com.mysql.jdbc.Driver'), 'Database URL' (set to 'jdbc:mysql://localhost:3310/project2'), 'Username' (set to 'root'), and 'Password' (masked with dots).
- Enter a SQL Command:** A text area containing the command 'select * from riders'.
- Buttons:** Below the command area are three buttons: 'Connect to Database' (blue), 'Clear Command' (light blue), and 'Execute SQL Command' (green).
- SQL Execution Result:** A table displaying the results of the query. The table has four columns: 'ridername', 'teamname', 'nationality', and 'num_pro_wins'. It lists 15 rows of data. A red arrow points from a text box to the 'nationality' column header.
- Clear Result Window:** A yellow button located at the bottom left of the results section.

A red-bordered text box with the following text is overlaid on the right side of the results table:

Note the metadata. Your application must print this for the user.

ridername	teamname	nationality	num_pro_wins
Alberto Contador	Astana	Spain	21
Alessandro 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
Michele Bartoli	UTS-Selle	Italy	81

A more complicated query:

The screenshot shows the 'SQL Client GUI - (MJL)' window. On the left, under 'Enter Database Information', the fields are: JDBC Driver: 'com.mysql.jdbc.Driver', Database URL: 'jdbc:mysql://localhost:3310/project2', Username: 'root', and Password: '....'. On the right, under 'Enter a SQL Command', the query is: 'select distinct racename from racewinners where ridername in (select ridername from riders where num_pro_wins > 50)'. Below these fields are three buttons: 'Connect to Database' (blue), 'Clear Command' (light blue), and 'Execute SQL Command' (green). The 'Execute SQL Command' button is highlighted. Below the buttons, the status bar shows 'Connected to jdbc:mysql://localhost:3310/project2'. Under 'SQL Execution Result', there is a table with one column 'racename' and five rows: 'Amstel Gold', 'GP-E3', 'Liege-Bastogne-Liege', 'Paris-Roubaix', and 'Rund de Flandren'. At the bottom left is a 'Clear Result Window' button.

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

When the user makes a mistake entering a SQL command:

The screenshot shows the same 'SQL Client GUI - (MJL)' window as before, but with an error message displayed in the 'SQL Execution Result' area. The error message is in a dialog box titled 'Database error' with a red 'X' icon. The text of the error is: 'Unknown column 'num_pro_winds' in 'where clause''. Below the error text is an 'OK' button. The rest of the GUI, including the database information and the SQL command, remains the same as in the previous screenshot.

Database error

Unknown column 'num_pro_winds' in 'where clause'

OK

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)". It contains two main sections: "Enter Database Information" and "Enter a SQL Command".

Enter Database Information:

- JDBC Driver: `com.mysql.jdbc.Driver`
- Database URL: `jdbc:mysql://localhost:3310/project2`
- 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)

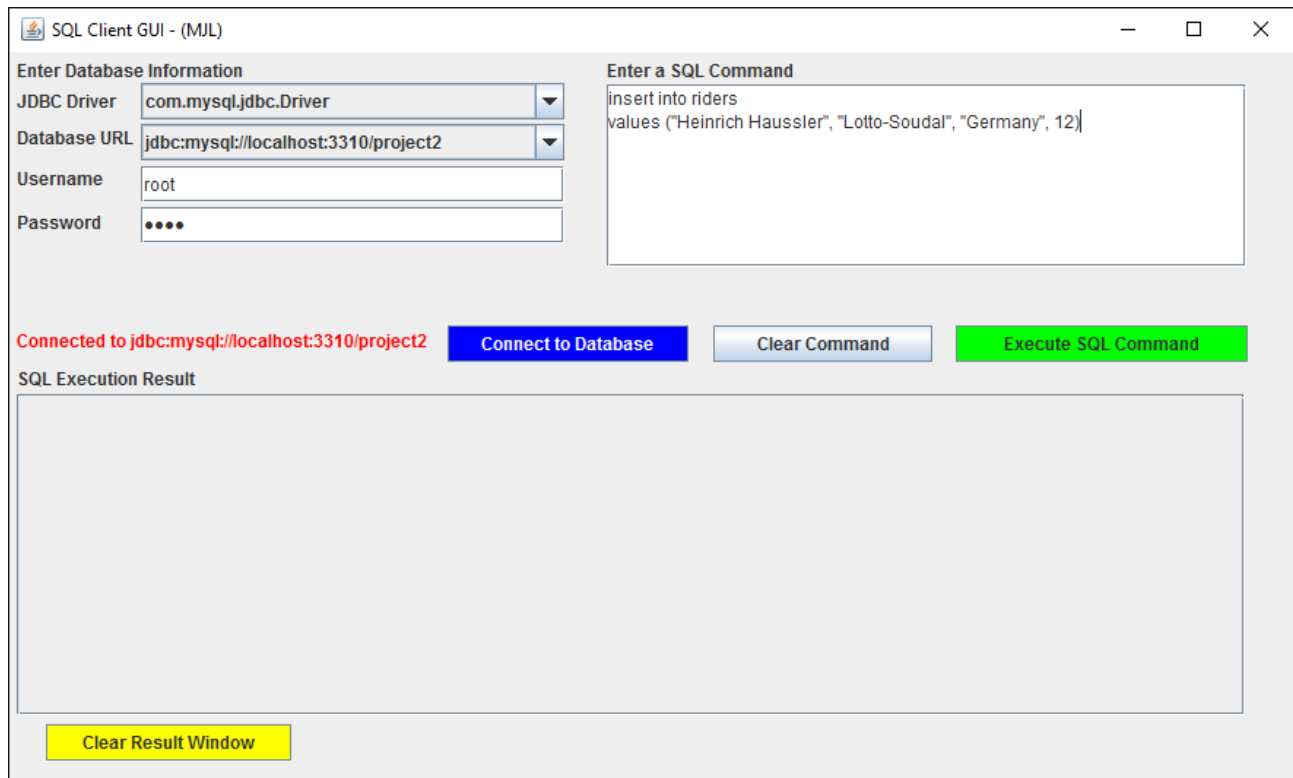
Connected to: `jdbc:mysql://localhost:3310/project2`

SQL Execution Result:

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

Clear Result Window (yellow button)

Insert command issued:

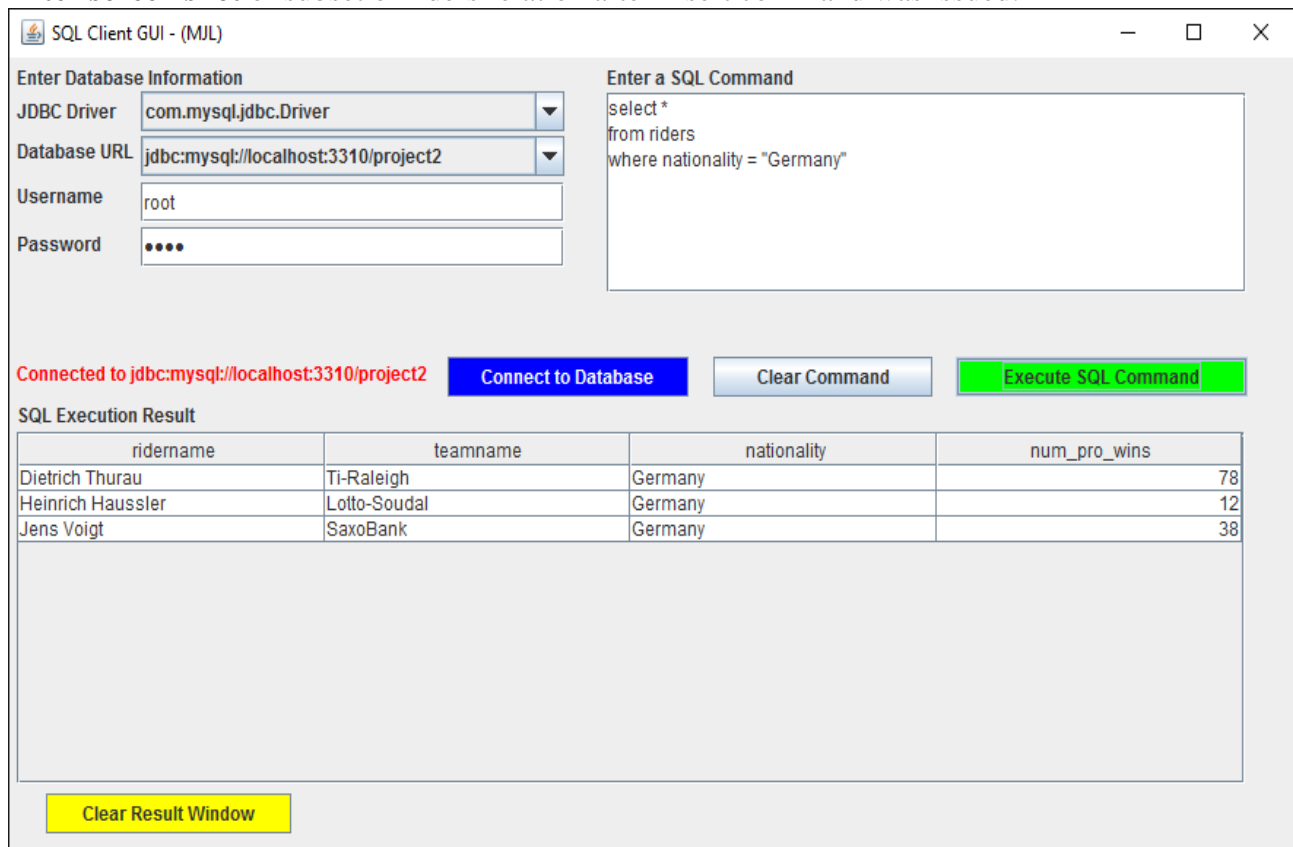


The screenshot shows the SQL Client GUI with the following details:

- Enter Database Information:**
 - JDBC Driver: `com.mysql.jdbc.Driver`
 - Database URL: `jdbc:mysql://localhost:3310/project2`
 - Username: `root`
 - Password: `....`
- Enter a SQL Command:**

```
insert into riders
values ("Heinrich Haussler", "Lotto-Soudal", "Germany", 12)
```
- Status:** Connected to `jdbc:mysql://localhost:3310/project2`
- Buttons:** Connect to Database, Clear Command, Execute SQL Command
- SQL Execution Result:** (Empty area)
- Clear Result Window** button at the bottom.

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/project2`
 - Username: `root`
 - Password: `....`
- Enter a SQL Command:**

```
select *
from riders
where nationality = "Germany"
```
- Status:** Connected to `jdbc:mysql://localhost:3310/project2`
- Buttons:** Connect to Database, Clear Command, Execute SQL Command
- SQL Execution Result:**

ridername	teamname	nationality	num_pro_wins
Dietrich Thurau	Ti-Raleigh	Germany	78
Heinrich Haussler	Lotto-Soudal	Germany	12
Jens Voigt	SaxoBank	Germany	38
- Clear Result Window** button at the bottom.

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

