

CNT 4714 – Project Two – Fall 2023

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

Points: 100 points

Due Date: Sunday October 29, 2023 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 clients (the end-user) to execute commands against a remote 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. User information for connections will be maintained in properties files, but each user must supply their username and password (for their MySQL server account) via the GUI for verification purposes (more later). You will be able to start multiple instances of your Java application and allow different clients to connect simultaneously to the MySQL DB sever, since the default number of connections is set at 151 (See your Workbench options file under the networking tab). In addition to the client interactions with your application, a background (business logic) transaction logging operation will occur which keeps a running total of the number of queries and the number of updates that have occurred via each user application (aggregate over all separate connections per user). This is a separate database (i.e., a completely different database than any to which a client user can connect), to which the application will connect, using special application-levels privileges in a separate properties file. This separate properties file is not accessible by any end user. Each user operation will cause the application to make this connection and update the operational logging database table. More details on this aspect of the application are shown below and will be covered in the Q&A sessions.

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 three different users. For this project you will create, in addition to the root user, two client users with limited permissions on the databases (see below). The root user is assumed to have all permissions on the databases, any command they issue will be executed. The client users will be far more restricted.

Restrictions:

1. Your source files should begin with comments containing the following information:

```
/*
```

Name: <your name goes here>

Course: CNT 4714 Fall 2023

Assignment title: Project 3 – A Two-tier Client-Server Application

Date: October 29, 2023

Class: <name of class goes here>

```
*/
```

2. Your application must provide a user interface, similar to the one shown below, that will allow any user the ability to connect to any database via properties files. Your application must verify that the user credentials (username and password) entered via the interface match with the user credentials found in the properties file that was selected via the interface. If the credentials do not match, then no connection is established.
3. Non-query commands should display a message to the user regarding the status of the executed command (see below).

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. You will also be using the **bikedb** from the notes. The script to create this database is already on Webcourses and if you have gone through the examples from the Module 3 notes (as you should have done), then you will have already created this database. If not, create it now.

The **second step** is to create a client-level user and assign authorizations for the user named **client1**. This client-level user is a new user with specific permissions. By default your root user has all permissions on the **project3** and **bikedb** databases. 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 set the specific permissions for the client as follows:

Register the new user named **client1** (assign them the password **client1** – ignore the MySQL warning on weak password setting) and assign to this user only **select** privileges on the **project3** and **bikedb** schemas.

The **third step** is to create another client-level user and assign authorizations for the user named **client2**. This client-level user is a new user with specific permissions. By default your root user has all permissions on the **project3** and **bikedb** databases. 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 set the specific permissions for the client as follows:

Register the new user named **client2** (assign them the password *client2* – ignore the MySQL warning on weak password setting) and assign to this user only **select** and **update** privileges on the **project3** and **bikedb** schemas.

The **fourth step** is to create the **operationslog** database using the `project3operationslog.sql` script. This script file is also available on WebCourses. This database will only be used and accessed by the project 3 application, it is not intended to be accessed by end users. You will need to create a specific user for this database. We will call this user **project3app**. Both the username and password should be *project3app* for this user. The **project3app** user will need to be assigned **select**, **insert**, and **update** privileges on the **operationslog** database.

Output Specification:

There are four 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 ten 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: **project3client1userscript.sql** available on the course website. There are seven different commands in this script and some of the commands will have more than one output capture (see below).
- Part 3 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: **project3client2userscript.sql** available on the course website. There are seven different commands in this script and some of the commands will have more than one output capture (see below).
- Part 4 is to provide a screenshot, taken from the MySQL Workbench perspective, that shows the final instance of the **operationscount** table in the **operationslog** database.

To produce your final output, first recreate the database, then run the root user commands followed by the client commands in script order within each script file.

Deliverables:

1. All of the .java files associated with your application.
2. All 17 screenshots from the execution of the commands specified in the **project3rootuserscript.sql** script.
3. All 11 screenshots from the execution of the commands specified in the **project3client1userscript.sql** script.
4. All 11 screenshots from the execution of the commands specified in the **project3client2userscript.sql** script.
5. A screenshot showing the final state of the **operationscount** table after executing the command `select * from operationscount;` once the **root** user and **client1** user, and **client2** user command script files have been completely executed. See page 15 in this document.

6. A screenshot showing a mismatch between the user-entered credentials and the selected properties file resulting in no connection to the database being established. See page 13 in this document.

All should be uploaded to WebCourses no later than 11:59pm Sunday October 29, 2023. Be sure to clearly label each screen shot. Use the convention: RootCommand1, RootCommand2A, RootCommand2B, and so on. Similarly for Client1Command1, Client1Command2A, and so on.

Details:

Beginning on page 5 are screen shots of the initial GUI and subsequent uses of the UI. Notice that there are drop-down lists for selecting the various properties file that will be used to make the user connection. There will be properties files that maintain database driver and URL information as well as properties files that maintain user details for the connections. The driver and database URL will be maintained in properties files using the naming convention **databaseName.properties**. The user credentials will be maintained in properties files using the naming convention **username.properties**. user credentials along with the JDBC driver and database URL will be specified in these files. The client must enter only their user credentials (username and password) through the GUI. Your application must verify that the user-entered credentials match those in the specified properties file before making a connection to the database. If the user entered credentials do not match those in the specified properties file, a message will be displayed to the user and no connection to the database will be established.

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.

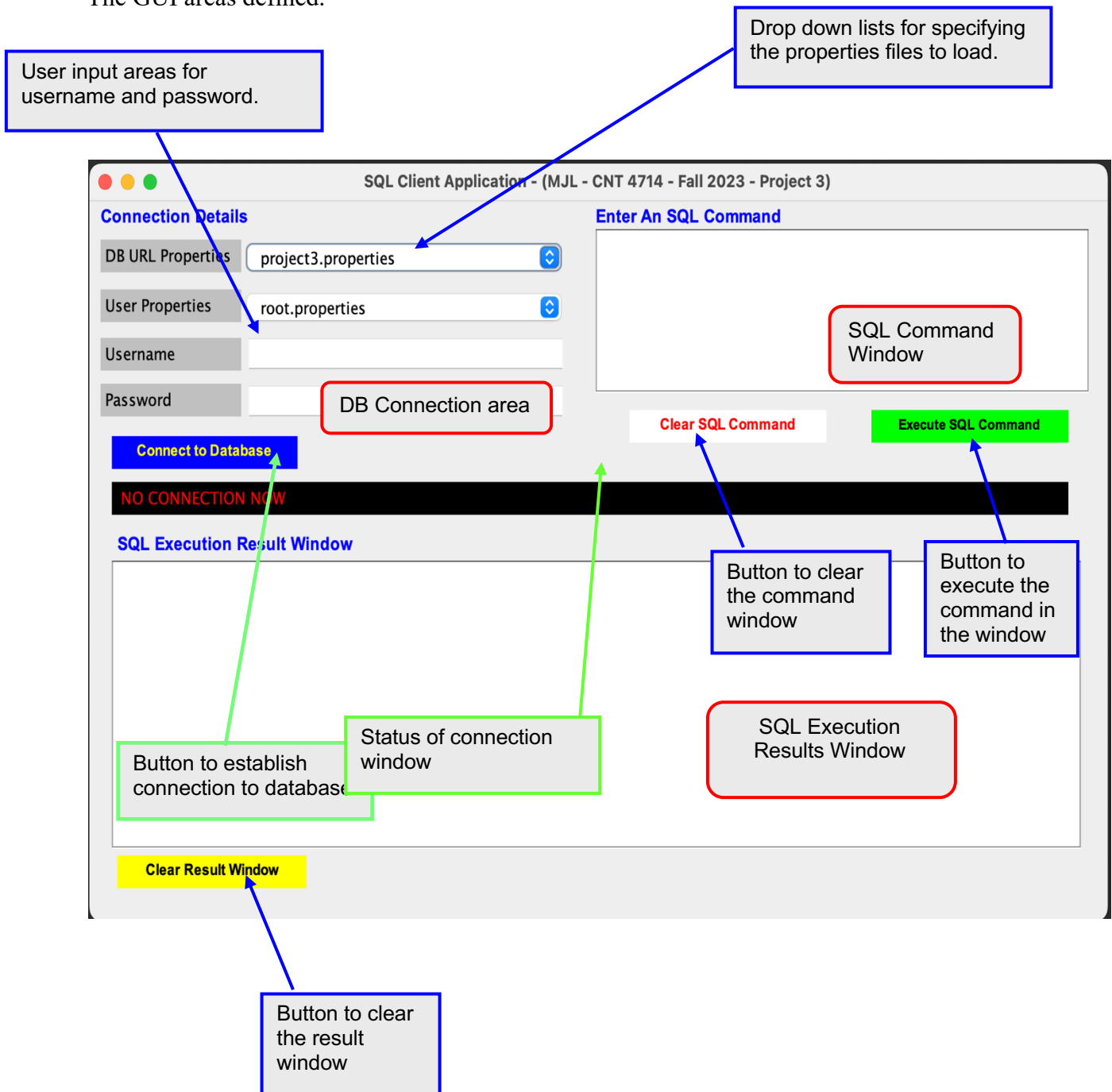
The output of all SQL commands should be returned to the SQL Execution Result window. Please note that only single SQL commands can be executed via this application (we will not execute scripts of commands). We will also not go to the effort of making the application display the results of MySQL-specific 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.)

As each command in the various user scripts is executed (only successful commands – some of the commands in the various scripts will not be successful) the **operationscount** table in the **operationslog** database must be updated by your application. Note that the **operationscount** table is initially empty. Each user’s queries and updates will be logged (counted) separately. Your application must obtain a connection to the **operationslog** database and perform the updates with **project3app** user credentials. Only successful operations will be logged – any transaction that errors will not increment any counter. These operations are invisible to the end user (regardless of who the user is, including root users). The application must connect to the **operationslog** database using a properties file which contains all necessary connection information.

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 11-12 for an illustration of this.

The remainder of the document illustrates the application at various phases during execution.

The GUI areas defined.



Screen shot illustrating an initial connection.

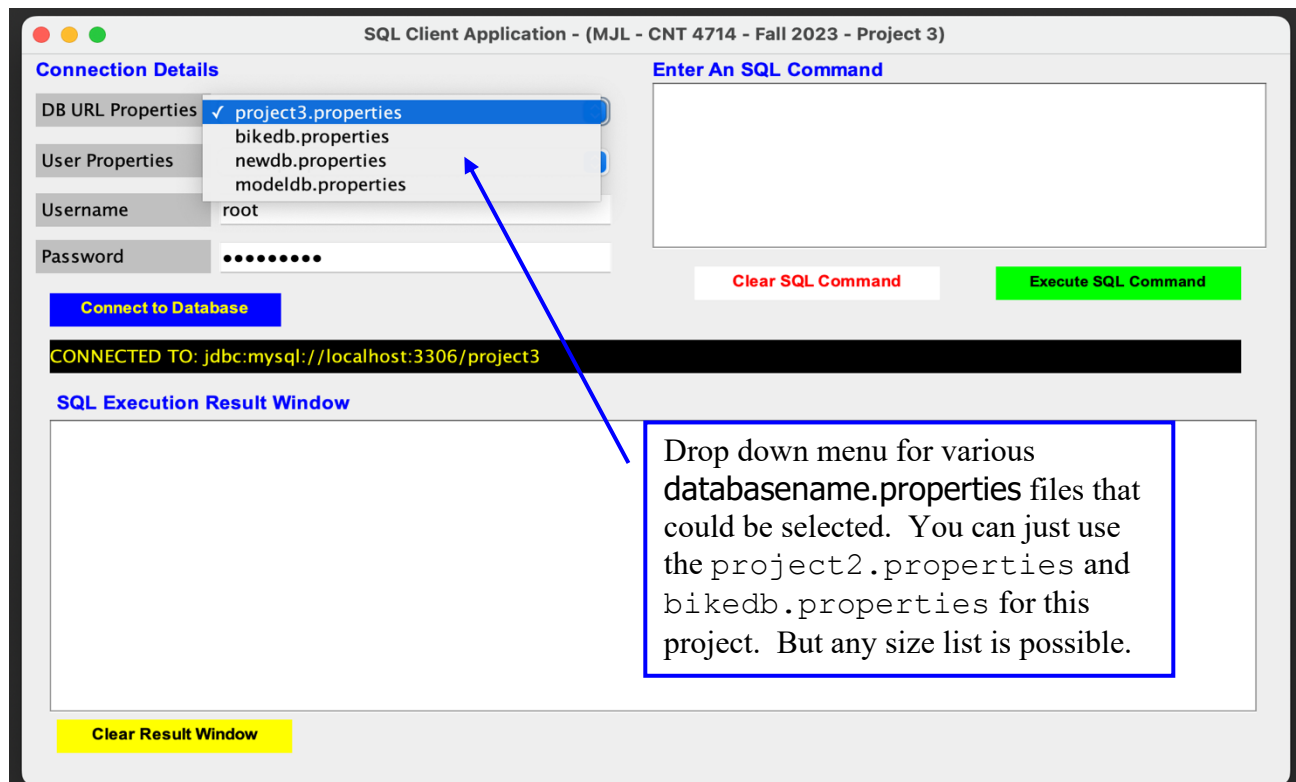
The screenshot shows a window titled "SQL Client Application - (MJL - CNT 4714 - Fall 2023 - Project 3)". The window is divided into several sections:

- Connection Details:** This section contains four input fields: "DB URL Properties" with the value "project3.properties", "User Properties" with the value "root.properties", "Username" with the value "root", and "Password" with masked characters "••••••••". Below these fields is a blue button labeled "Connect to Database".
- Enter An SQL Command:** This section features a large text area for entering SQL commands. Below the text area are two buttons: "Clear SQL Command" (red) and "Execute SQL Command" (green).
- Connection Status:** A black bar with yellow text displays the message "CONNECTED TO: jdbc:mysql://localhost:3306/project3".
- SQL Execution Result Window:** This section is labeled "SQL Execution Result Window" and contains a large empty area for displaying query results. A blue arrow points from a text box to this section.

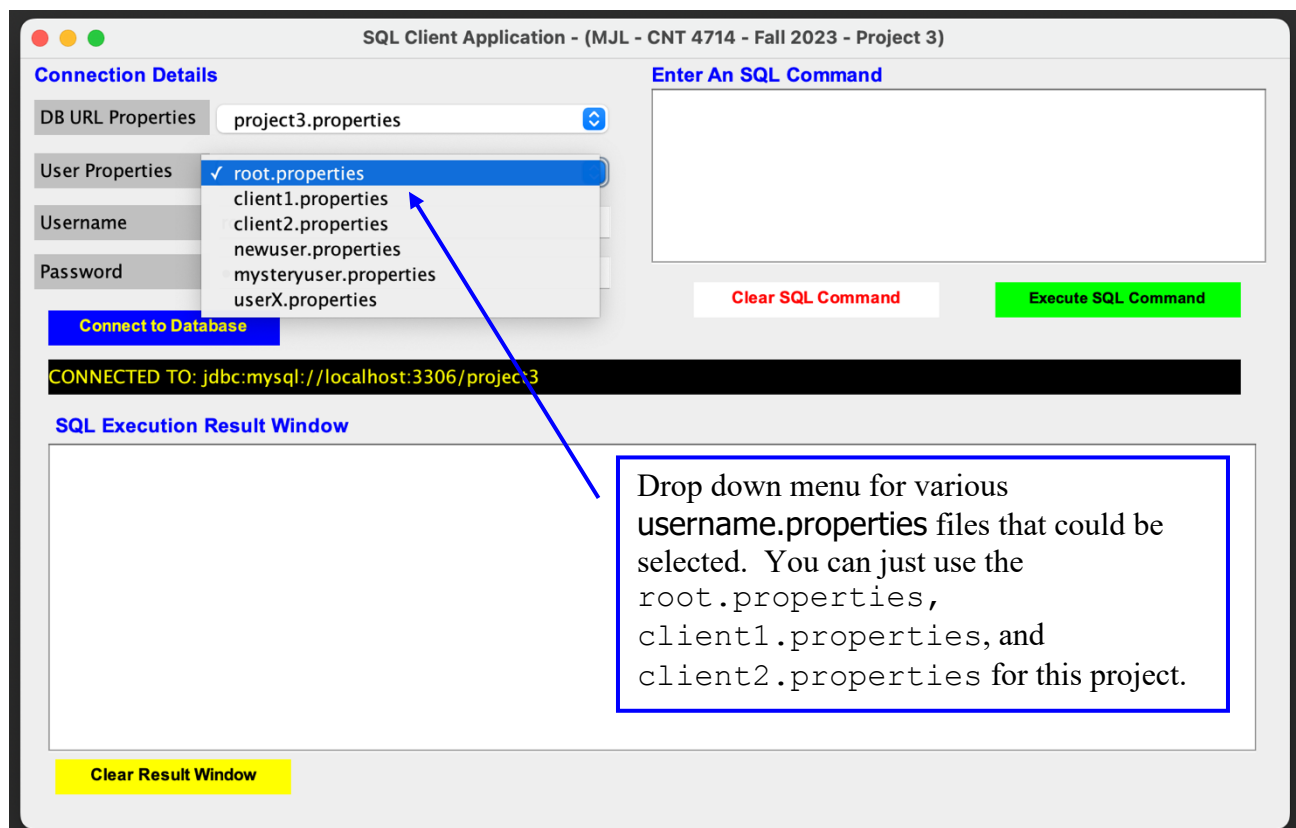
A text box with a blue border contains the message "Connection established to selected database URL".

At the bottom left of the window is a yellow button labeled "Clear Result Window".

Illustrating the drop-down list of possible `databasename.properties` files that could be selected.



Illustrating the drop-down list of possible `username.properties` files that could be selected.



The **root** user has connected to the **project3** database and issued a select command. Results are displayed in the SQL Execution window.

The screenshot shows an SQL Client Application window titled "SQL Client Application - (MJL - CNT 4714 - Fall 2023 - Project 3)". The interface is divided into several sections:

- Connection Details:** Includes fields for DB URL Properties (project3.properties), User Properties (root.properties), Username (root), and Password (masked with dots). A blue "Connect to Database" button is located below these fields.
- Enter An SQL Command:** A text area containing the command "select * from riders". Below it are two buttons: "Clear SQL Command" (red) and "Execute SQL Command" (green).
- Execution Status:** A black bar with yellow text stating "CONNECTED TO: jdbc:mysql://localhost:3306/project3". A red arrow points from a callout box to this status bar.
- SQL Execution Result Window:** Displays a table of results with columns: ridername, teamname, nationality, and gender. The table contains 15 rows of data.
- Callout Box:** A red-bordered box with the text: "Note the metadata. Your application must print this for the user." It points to the "CONNECTED TO" status bar.
- Clear Result Window:** A yellow button located at the bottom left of the results window.

ridername	teamname	nationality	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
Ceylin del Carmen Alvarado	Alpecin-Fenix	Netherlands	88 F
Chris Froome	Sky	Great Britain	23 M
Davide van der Poel	Jumbo-Visma	Alpecin	0 M
Dietrich Thurau	Ti-Raleigh	Germany	78 M
Elisa Balsamo	Trek-Segafredo	Italy	20 F
Elisa Borghini	Schenger	Italy	34 F
Fabian Cancellara	SaxoBank	Switzerland	58 M
Fedor den Hertog	Acqua & Sapone	Netherlands	20 M
Filippo Ganna	Ineos	Italy	25 M

A more complicated query:

SQL Client Application - (MJL - CNT 4714 - Fall 2023 - Project 3)

Connection Details

DB URL Properties

project3.properties

User Properties

root.properties

Username

root

Password

••••••••

Connect to Database

Enter An SQL Command

```
select distinct racename
from racewinners
where ridername in (select ridername
                    from riders
                    where num_pro_wins > 50)
```

Clear SQL Command

Execute SQL Command

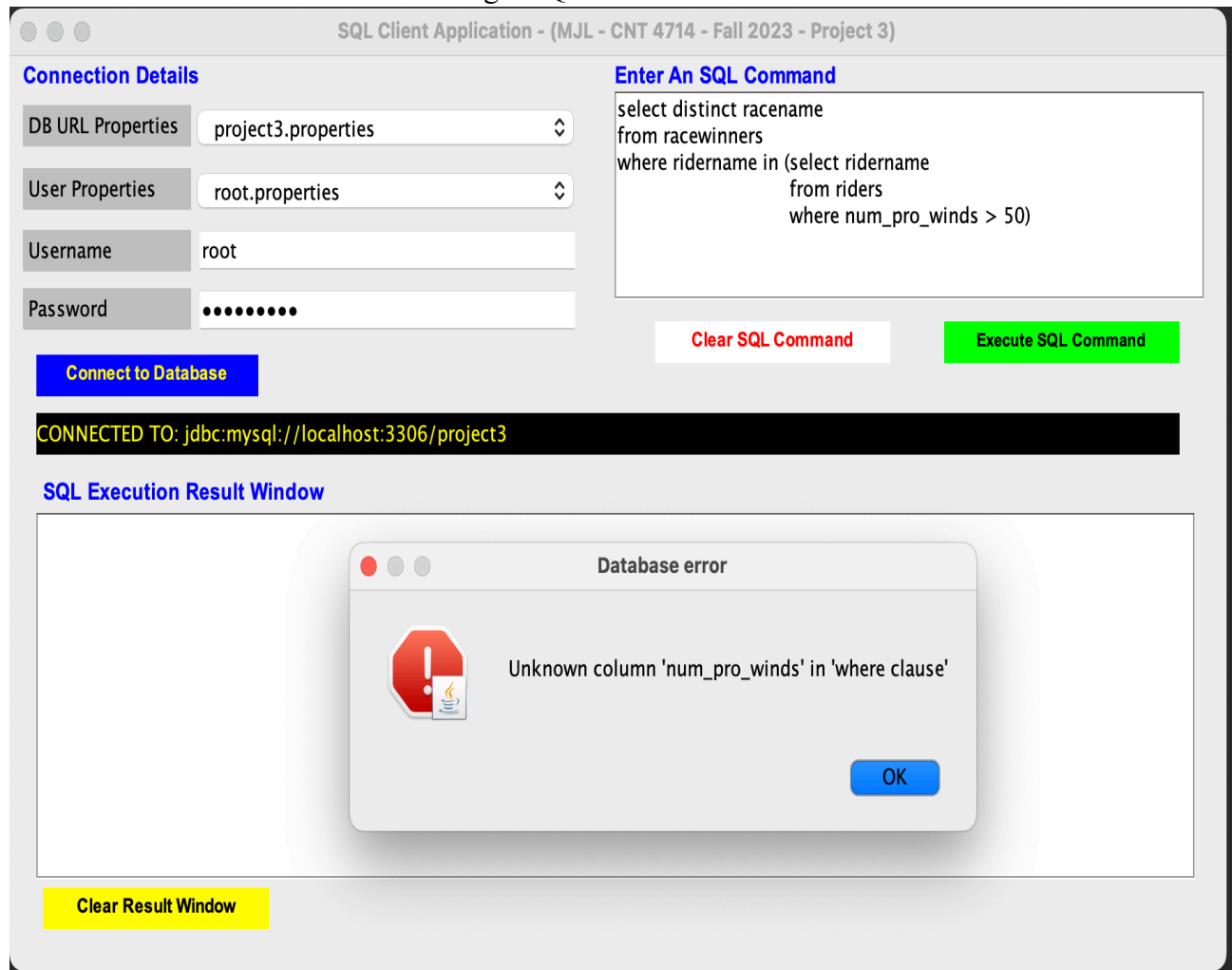
CONNECTED TO: jdbc:mysql://localhost:3306/project3

SQL Execution Result Window

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

Clear Result Window

When the user makes a mistake entering a SQL command:



The following three 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 an application window titled "SQL Client Application - (MJL - CNT 4714 - Fall 2023 - Project 3)". It is divided into several sections:

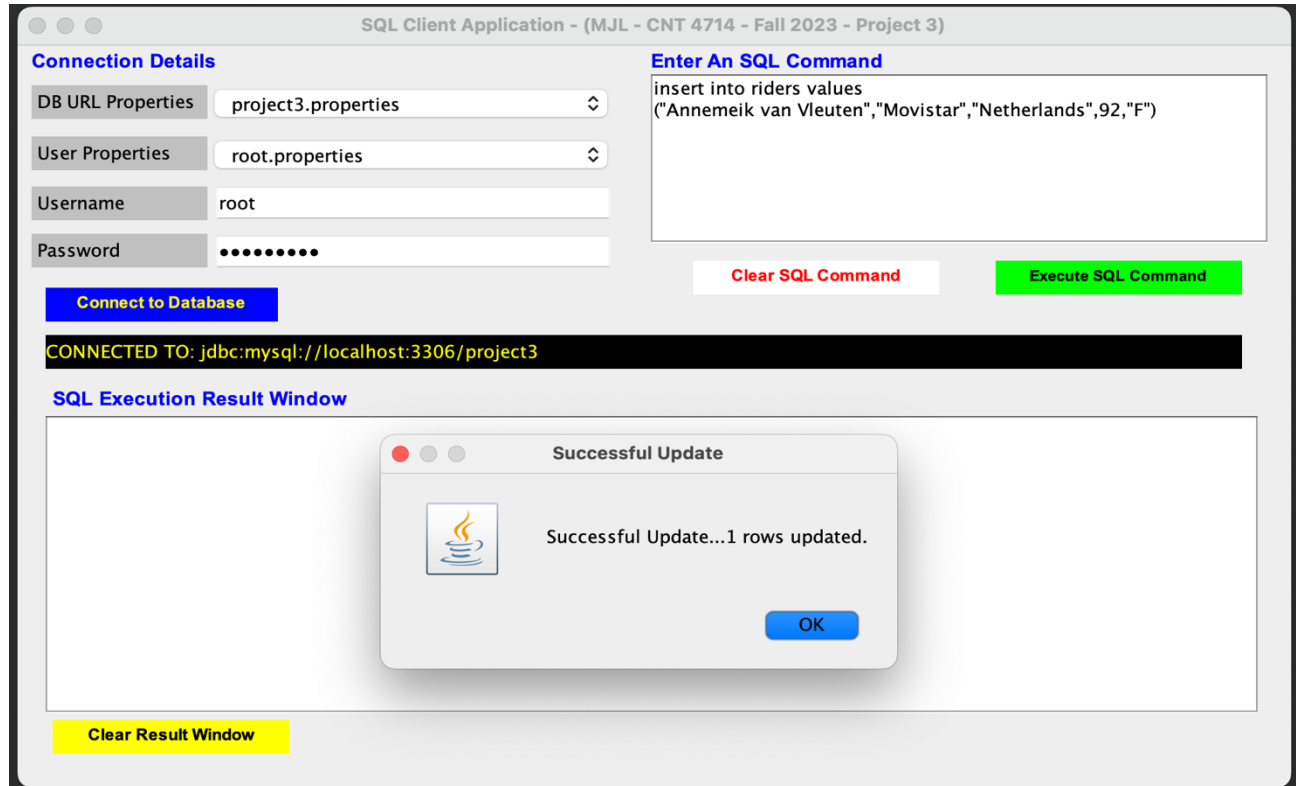
- Connection Details:** Contains fields for DB URL Properties (project3.properties), User Properties (root.properties), Username (root), and Password (masked with dots). A blue "Connect to Database" button is at the bottom.
- Enter An SQL Command:** A text area containing the query:

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

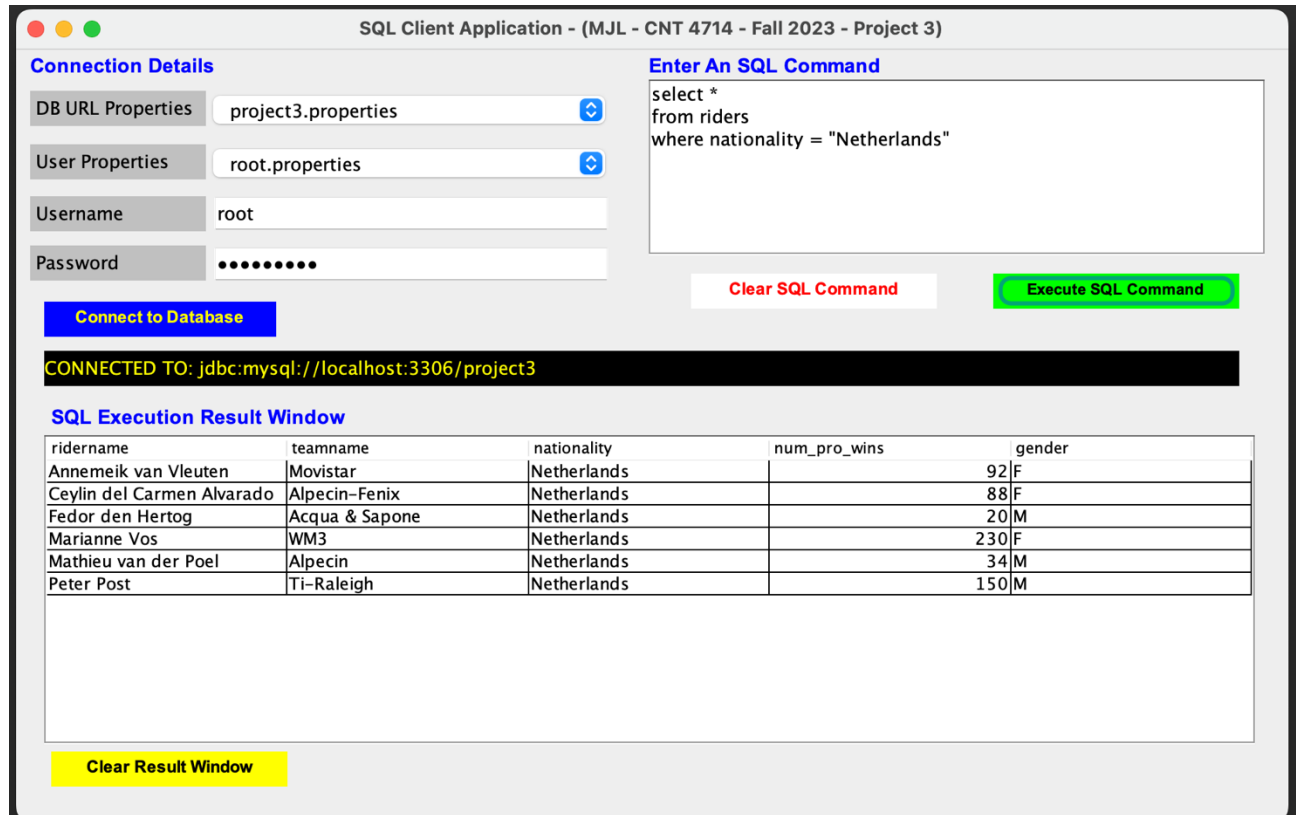
 Below it are two buttons: "Clear SQL Command" (red) and "Execute SQL Command" (green).
- Execution Status:** A black bar with yellow text stating "CONNECTED TO: jdbc:mysql://localhost:3306/project3".
- SQL Execution Result Window:** A table displaying the results of the query.
- Clear Result Window:** A yellow button at the bottom left of the result window.

ridername	teamname	nationality	num_pro_wins	gender
Ceylin del Carmen Alvarado	Alpecin-Fenix	Netherlands	88	F
Fedor den Hertog	Acqua & Sapone	Netherlands	20	M
Marianne Vos	WM3	Netherlands	230	F
Mathieu van der Poel	Alpecin	Netherlands	34	M
Peter Post	Ti-Raleigh	Netherlands	150	M

Insert command issued:



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



Screen shot illustrating the client1 user issuing a select command.

SQL Client Application - (MJL - CNT 4714 - Fall 2023 - Project 3)

Connection Details

DB URL Properties: project3.properties

User Properties: client1.properties

Username: client1

Password:

Enter An SQL Command

select * from riders

Clear SQL Command **Execute SQL Command**

Connect to Database

CONNECTED TO: jdbc:mysql://localhost:3306/project3

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
Annemeik van Vleuten	Movistar	Netherlands	92	F
Bradley Wiggins	Ti-Raleigh	Great Britain	13	M
Ceylin del Carmen Alvarado	Alpecin-Fenix	Netherlands	88	F
Chris Froome	Sky	Great Britain	23	M
Davide van der Poel	Jumbo-Visma	Alpecin	0	M
Dietrich Thurau	Ti-Raleigh	Germany	78	M
Elisa Balsamo	Trek-Segafredo	Italy	20	F
Elisa Borghini	Schenger	Italy	34	F
Fabian Cancellara	SaxoBank	Switzerland	58	M
Federico Bertoni	Acqua & Sapone	Netherlands	20	M

Clear Result Window

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

SQL Client Application - (MJL - CNT 4714 - Fall 2023 - Project 3)

Connection Details

DB URL Properties: project3.properties

User Properties: client1.properties

Username: client1

Password:

Enter An SQL Command

update riders
set num_pro_wins = 98
where ridername = "Annemeik van Vleuten"

Clear SQL Command **Execute SQL Command**

Connect to Database

CONNECTED TO: jdbc:mysql://localhost:3306/project3

SQL Execution Result Window

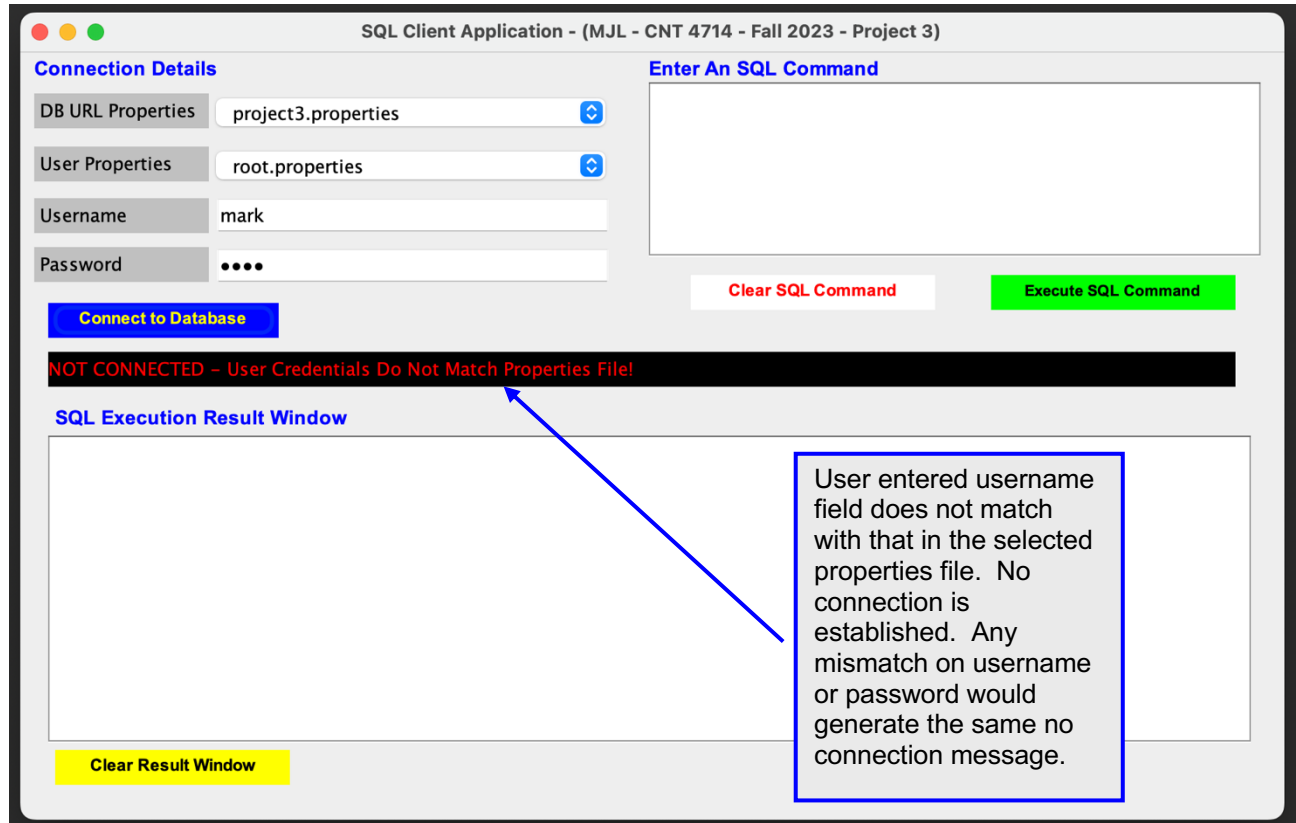
Database error

UPDATE command denied to user 'client1'@'localhost' for table 'riders'

OK

Clear Result Window

The following screenshot illustrates how the user-entered credentials must match those in the selected properties file in order to establish a connection to the database specified in the properties file.



The following screenshot illustrates the **operationscount** table values after various operations have been completed. This screenshot is taken from the **project3app** user account in the MySQL Workbench using the **operationslog** database. Note that the numbers shown in this screenshot are not the correct numbers that you will see after executing the different user command scripts in the project. This is just an example.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'operationslog' expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main query editor contains the query 'select * from operationscount'. Below the editor, the 'Result Grid' shows the following data:

login_username	num_queries	num_updates
client1@localhost	1	0
client2@localhost	3	1
root@localhost	7	5
NULL	NULL	NULL

Below the result grid, the 'Action Output' tab shows the execution details for 'operationscount 1':

	Time	Action	Response	Duration / Fetch Time
1	19:04:22	select * from operationscount LIMIT 0, 1000	3 row(s) returned	0.00037 sec / 0.0000...