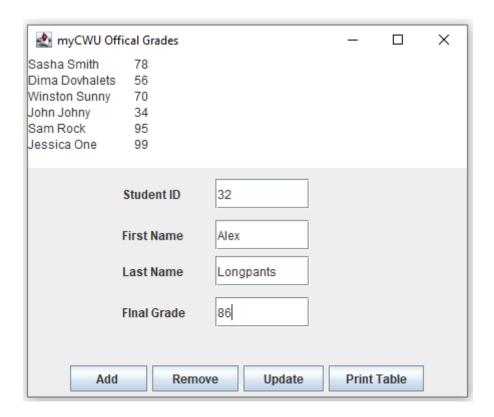# Lab Assignment #4
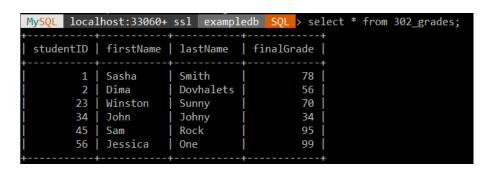
MySQL and Java GUI

Description

For this lab you will make a GUI that looks similar to the one below. It should function as an application that can be used to enter final grades and store them into a MySQL database. The table must be named the same and have the same columns with the same names as the one below.

Your application needs to be able to add, remove and update records in the database using Java statements. It also needs to be able to print out the first name, last name and the final grade for all entries in the table (button labeled "Print Table").

Details

You have to create a database and table using MySQL Shell or MySQL Workbench. The table must be named the "**302_grades**" and it must have the same number of columns. It is also important that your column names are spelled exactly like the picture above (studentID, firstName, ..etc), otherwise your application will not be able to connect to the testing database, when grading your submission.

Java has several Interfaces that allow us to connect and make queries with a MySQL database. In order to connect to the database we need to make use of Java Database Connectivity(JDBC) API. You will need to download "MySQLConnector/J 8.0". it is a jar file that you will link with your application.

A sample method that you can replicate to connect to the database can be seen below. In the example below you would have to edit the "url", "userName" and "pass". In my example "/exampledb" is the name of the database.

```java
public void connect() {

    String url = "jdbc:mysql://localhost:3306/exampledb";
    String userName = "dovhalets";
    String pass = "password";

    try{
        Connection con = DriverManager.getConnection(url, userName, pass);
        System.out.println("connected");

    }catch(Exception e) {
        System.out.println("exception "+ e.getMessage());
    }

}
```

To make a query using Java statements and retrieve data from the database you will need to make use of two more Java interfaces; Statement and ResultSet. A query can be passed in as a string into a method named executeQuery, which is called on an instance of Statement. A database result set can be stored in a ResultSet object, which maintains a cursor that can be used to navigate through the table.

Below is an example of query which retrieves everything from the table "302_grades". A while loop is used to move the cursor across the table. "result.getString("firstName")" will return an entry from the table from the row where the cursor is currently at , from the column named "firstName". Essentially the below code will print out all of the first names from the table.

```
String query = "select * from 302_grades";

Statement statement = con.createStatement();
ResultSet result =  statement.executeQuery(query);


while(result.next()) {
    System.out.println(result.getString("firstName"));
}
```

You will need to check the Java API for other methods in Connection, ResultSet and Statement in order to complete this lab. There are plenty of examples and tutorials online.

Submission

You need to submit to Canvas a screenshot of your table named "302_grades" this can be done by running the command "select * from 302_grades;".

You also need to submit a zip file of your entire project.

Rubric

100 points – A functional GUI that interacts with a MySQL database and is able to add, remove, update and print the entire table. When testing we only should have to modify the "url", "userName" and "pass".