Database System I
University of Auburn
Summer 2022
Professor Yang Zhou

Course Project

In this term project, you are required to do the followings:

1. Design and setup your database with the provided data.
2. Create a Java application with GUI that is capable of interacting with the database.
3. Correctly Implement the queries in SQL.

Design and Setting Up the Database

30pts

MariaDB, which is open-source variant of MySQL, is going to be used in this project. You would need to first install it on your computer.

Installation

Windows Installation package can be found here `https://mariadb.org/download/`. Things are pretty easy on Windows. Simply run the installation program and follow its instructions will do the job. Remember to tick the check box for "Allow remote root access" as well as taking a note of your root password and server port number. After installation, run the "Command Prompt (Mariadb)" program to access your database.

Linux You can install it via apt-get if your system does not have it already. first do

```
sudo apt update
```

Then

```
sudo apt install mariadb-server
```

Finally

```
sudo mysql_secure_installation
```

You will be greeted by a few questions related to configurations of the database. First following the instruction and set a root password. Then proceed following the settings below.

```
New password:
Re-enter new password:
Password updated successfully! Reloading
privilege tables..
 ...Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation go
a bit smoother.     You should remove them before moving into a production
environment.

Remove anonymous users?[Y/n] Y
 ...Success!

Normally, root should only be allowed to connect from 'localhost'.    This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely?[Y/n] n
 ...skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it?[Y/n] Y
 -Dropping test database...
 ...Success!
 -Removing privileges on test database...
 ...Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now?[Y/n] Y
 ...Success!

Cleaning up...

All done!If you  've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

After installation, run the following command to access your database.

```
    sudo mysql -u root -p
```

MacOS MacOS is not recommended for this project. If you insist on doing it with a Mac, here a link that might be useful for you. `https://mariadb.com/kb/en/installing-mariadb-on-macos-using-homebrew/`

Creating and Loading data into Your Database

Designing your Database You should be design the schema of your database using ER-diagram. Primary keys, Foreign Keys and other constraints should be properly applied to tables during this phase.

Creating the Database After you log into your database server for the first time, what you need to do is creating a database. You can use the CREATE DATABASE statement to do so. After doing so, select your database with the USE statement. Now you are ready for creating actual4 tables and filling them with data in this database.

Additional Information CSV files can be loaded into the database using the LOAD DATA Statement. Be sure to add the "LOCAL" keyword otherwise there might be permission problems. See details here. `https://dev.mysql.com/doc/refman/8.0/en/load-data.html`

Interface

40pts

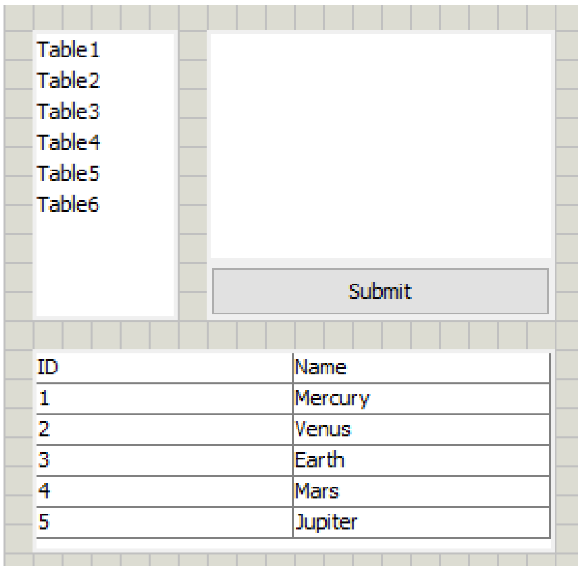
Once your work with the database is done, you can move on to coding the Java application that deals with it. Three UI components are required, all other things are optional.

As shown in the figure right:

1. Table list area, where the user should be able to click on one of themcheck the corresponding contents in the result area below.

2. input form, where the user could input SQL statement and submit it to the serverto execute. Note: your input form should NOT accept "DROP" operation.

3. result area, where either error message or the results show here.



ID	Name
1	Mercury
2	Venus
3	Earth
4	Mars
5	Jupiter

The layout of the UI is all up to you and you don't have to make it nice looking.

Data and Queries

30pts

In data.zip (which can be found in the "Files" section on Canvas), there are csv and txt files. The csv files are the data you need to put into your database, while the query.txt contains 20 queries you need to implement in SQL language, which I will test through your Java app.

Deliverable

What you need to turn in is a zip file, named like **your_auburn_username.zip** which should contain:

1. All source files for your Java app.
2. An sql.txt contains all your sql statements which solves the questions in query.txt.
3. The ER-diagram of your database.

Coding Hints

Sample Code

Below is piece of sample code that I wrote to show you how you can interact with MariaDB in Java.

```
packagesqlSample;
importjava.sql.Connection;
importjava.sql.DriverManager;
importjava.sql.SQLException;
importjava.sql.Statement;
importjava.sql.ResultSet;

publicclassexec{

    publicstaticvoidmain(String[]args){
        try{
            // The newInstance() call is a work around for some
            // broken Java implementations
            Class.forName("com.mysql.cj.jdbc.Driver").getDeclaredConstructor()
                                                            .newInstance();
            String dbName="Your Database Name"; String
            port="Your Database Server Port"; String
            pwd="Your root Password";
            Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:"
                                    +port+"/"+dbName+"?"+
                                   "user=root&password="+pwd);
            Statement stmt=conn.createStatement();
            ResultSet rs=stmt.executeQuery("SELECT * FROM Planets");

            while(rs.next()){
                intid=rs.getInt("ID");
                String name=rs.getString("name");
                System.out.println(id+"---"+name);
            }


        }catch(SQLException ex){
            // handle the error
            System.out.println("SQLException: "+ex.getMessage());
            System.out.println("SQLState: "+ex.getSQLState());
            System.out.println("VendorError: "+ex.getErrorCode());
        }
        catch(Exception e)
        {
            System.out.println("Unkown Error:"+e.getMessage());
        }
    }
}
```

Working with the JDBC driver

The following piece of code requires the help of a JDBC driver to work. It's jar file which you can download from Canvas via the link in the project details. To run your java code with the driver, first compile your code into classes via javac, then put the driver jar into your current working folder and run:

```
java -cp".;mysql-connector-java-8.0.16.jar"YourPackage.YourMainClass
```

Or you can add the location of the driver jar into your classpath and run your code directly.