cs3743 Pgm 3 JDBC (40 pts) – due 4/23

In this assignment, you will use MySQL and JDBC:

* Use mysql comand line utility:
  + Connect to your database
  + Create the Flight, Customer, and Reservation tables in your database using SQL scripts.
  + Create an additional index on the Reservation table.
  + Populate each of those tables using SQL scripts
* Use JDBC to connect your Java program to your database and perform a variety of select, insert, and delete operations

**MySql User Id, Password, and Database**

Your user id: ***abc123***

Your password: *lastThreeDigitsBannerId***pw**

Your database name: *abc123***db**

mysql server address: 10.100.1.81

mysql serve name: db01

**Source Directory**

* Create a directory for this assignment
* Copy the sql Script directory and cs3743 package directory to that directory

$ cp -r /usr/local/courses/clark/cs3743/2018Sp/P3/\* .

Note: once you have started coding, don't do this cp again since it will overwrite your code.

* You should notice that it created a "**cs3743**" directory which should contain:

**MySqlUtility.java** java code which can print the result set and meta data. You should not have to change this code.

**P3Main.java** contains a static main method for invoking your P3Program passing your ID and password. You need to change those.

**P3Program.java** contains your java source code which uses jdbc. Initially, it will have some constants. This is where you will do the bulk of your work.

Note that this directory contains the java package named cs3743 which is why the directory is named that.

* You should notice that it created a "**Script**" directory which should contain:

**createFlight.sql** SQL script to drop **Flight** if it exists and create the **Fight** table.

**createCustomer.sql** SQL script to drop **Customer** if it exists and create the **Customer** table.

**createReservation.sql** SQL script to drop **Reservation** if it exists and create the **Reservation** table.

**insertFlight.sql** SQL script to insert rows into the **Flight** table.

**insertCustomer.sql** SQL script to insert rows into the **Customer** table.

**insertReservation.sql** SQL script to insert rows into the **Reservation** table.

**Using MySQL to connect to flightdb**

1. Login to either a fox server or one of the Linux workstations in the CS main lab.
2. To launch the mysql command line utility (use your abc123 ID and password

$ mysql –h **db01** –u ***abc123*** –p

* It will prompt you to enter a password.
* It should then give you the **mysql>** prompt.

1. Tell mysql that you want to use **your database**:

mysql> **use *abc123*db;**

* + It should state "**Database changed**".

1. Your database doesn't have any tables.
2. Use the mysql **source** command to run **each of the sql scripts to create your tables and then populate them.**

mysql> **source** Script/*filename*.sql;

Note that the create scripts first do a **drop table if exists** which will show a warning if it doesn't already exist. That is expected.

1. Confirm that those tables are populated:

mysql> **select \* from Flight;**

mysql> **select \* from Customer;**

mysql> **select \* from Reservation;**

1. The **Reservation** Table has a primary key index on `custNr` then `flightId`. Execute an SQL statement to create another unique index by `flightId` then `custNr` on that table. You can name that index `flightIdx`.
2. To exit the mysql command line utility:

mysql> **exit;**

**Java Programs**

On a fox server or CS Main Lab Linux workstations, you will create a Java program. Before proceeding, make certain you set the CLASSPATH in your .cshrc file as stated in the course notes.

**P3Main.java**

Change it to use your id and password.

**P3Program.java**

1. Change the constructor method to use your database.
2. Create a **printCustomers**(title, resultSet) instance method which prints the title, a column heading, and the data for each tuple. Make certain you check for null values where appropriate and print "---". This must be code in a manner similar to the **printSections** method in the MySQLandJDBC notes; however, you should indent the column headings and column data four spaces to improve readability.
3. Create a **runProgram** method which does the following:
   1. Use **statement** to execute this select statement:

select c.\* from Customer c

* 1. Use **printCustomers**("Beginning Customers", resultSet) to print those customers.
  2. Use **statement** to execute this select statement:

select f.\* from Flight f

* 1. Use **MySqlUtility.printUtility**("Beginning Flights", resultSet) to print those flights.
  2. Use statement.**executeUpdate** to insert a row into Customer which uses:
     + 1999 for the custNr
     + your name for the name
     + your favorite airline for the airline
     + your birthdate (or use NULL if you would rather not provide it) for birthdate
     + your gender

Note: place the executeUpdate in a special **try… catch** to print an error if there was a duplicate as was done in our notes Example #14.

* 1. Use **statement** to get the customers (as was done in step a.) and **printCustomers**("Customers after I was added", resultSet).
  2. Create a **prepareStatement** which does an **insert** into Reservation using subsitution parameters for each attribute. This should be done outside the **for** statement of the next step.
  3. Use a Java **for** statement to iterate through the **strFlightIdM** array until a value of "END" is encountered. Set the substitution parameters as follows:
* Parameter 1 to 1999.
* Parameter 2 to strFlightIdM[i]. Assuming i is your counter variable in the **for.**
* Parameter 3 to 2.

Execute the insertion using preparedStatement.**executeUpdate**().   
Note: place the executeUpdate in a special **try… catch** to print an error if there was a duplicate as was done in our notes Example #14.

* 1. Create a **prepareStatement** to **select** reservations for a custNr that is provided as a subsitution parameter. Provide 1999 for the custNr parameter and use preparedStatement**.executeQuery()** to execute it. Print the result using **MySqlUtility.printUtility**("My reservations", resultSet).
  2. Use **statement** to execute a select statement which returns the flightId, custNr, and customer name for customers who are on flights that customer 1999 is on. Do not include 1999 in the returned rows. Print them using   
     **MySqlUtility.printUtility**("Other customers on my flights", resultSet).
  3. Use **statement** to execute a select statement which returns flightId and count(\*) for flights having more than two reservations. (See example #15 in the SQL DML notes.) Print the results using  
     **MySqlUtility.printUtility**("Flights Having more than 2 reservations", resultSet).
  4. **Delete** the reservations for customer 1999 from the **Reservation** Table using **statement**.
  5. Repeat step J, but use this for the title "Other customers on my flights after mine were deleted". Of course, no results are printed.

**To compile your code from the directory above cs3743 (package directory):**

$ javac cs3743/P3Main.java # should only do this once

$ javac cs3743/MySqlUtility.java # should only do this once

$ javac cs3743/P3Program.java

**To execute your main:**

$ java cs3743/P3Main

**What to submit via BlackBoard?**

Submit a zip file named *LastnameFirstName*.zip containing:

* Your **P3Program.java**
* Your **P3Main.java**
* Your output generated by your program. Place it in a file named **p3Out.txt**