**PROJECT Bank record loan generation report 100 points**

**Objective** To write a program that creates a Loan analysis report from class objects created in lab #2/3.

***PROJECT DESCRIPTION***

Bank of IIT now needs your help in deciphering whom from its records should be exlusive to premium loans versus those offered micro or less premium loans.

Use a database to store then present Loan analysis information from your data BankRecords objects you worked on in prior labs.

***Project Details***

For this lab you will continue using your current project folder created for labs 2 & 3 and create a new file called **Dao** where Dao stands for Data Access Object. This will allow for CRUD database connectivity and operations. Create also a file called **LoanProcessing** as well which will act as a driver file (i.e., includes main function) to call your database CRUD methods and create some resulting output.

To start working with a database you will need a JDBC driver to allow for any connectivity within your app.

To include a driver for your MySQL database, you need to create a folder called **libs** within your project. The MySQL driver for JDBC connections packages in the form of a jar file which you can download here:

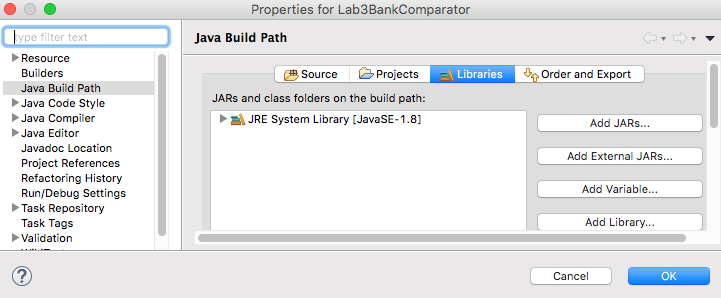
You will find a JDBC driver for connecting to a MySQL database located here:

<https://dev.mysql.com/downloads/connector/j/>

Depending on your OS you will need to download a windows zip file or a tar file if you are a Mac/Linux user. Unzip or untar the DOWNLOADED file and locate the needed .jar file and copy it into into your **libs** folder.

Make sure to include the jar file in your Build Path by right clicking on your project folder and then choose Build Path > Configure Build Path....

Once the dialog box opens make sure to click on the Libraries tab then click Add JARs…



Drill down to the libs folder within your project and then choose the jar file to add.

Click OK to commit.

Start coding your Dao class file. Include the following methods for your file.

-A method to create a database table called **createTable**. createTable merely creates a table when called. Include an **id** field, an **income** field and a **pep** field when building your table setup.

[ Note when creating a table it is IMPERATIVE to include the following name:

**yourFirstinitial\_First4LettersOfYourLastName\_tab** ]

-A method to insert records called **insertRecords()**.

-A method to retrieve records for display called **retrieveRecords()**.

In order to communicate or ‘connect’ with your database (which will be located on the papaserver) for CRUD operations, you’ll need to include certain credentials. Use the data below for establishing a JDBC connection to the papaserver.

url = "jdbc:mysql://www.papademas.net:3306/411labs";

username = "db411";

password = "411";

**Methods breakdown**

insertRecords(BankRecords [] arrayName) will allow for the array of BankRecord objects, to be passed to your method which will allow for the insertion of all the **id**, **income** and **pep** data from your BankRecords array (or *whatever* you named it) into your database table when called. Make sure then to ***extend*** your Dao class to include your **BankRecords** class to allow data to be obtained.

retrieveRecords() will return a **ResultSet** object used for creating output. Have the result set contain the record data consisting of the id, income and pep fields. Sort the pep field in *descending* order to allow for premium loan candidates to appear first in the record set for reporting purposes (i.e., those with data values of “YES”). The resultset query string to build can be something like:

String sql =

"select id,income, pep from your\_table\_name order by pep desc";

Make sure to always close out of your connections and any statements when through with any processing!

Make sure to include error trapping using SQLException handling for all your database operations and connection logic.

Also include messages to the console when your methods trigger. Ex. Table created, Inserting records into database, etc.

A super great resource to assist you with all your CRUD operations for your methods can be found at this site: <http://www.tutorialspoint.com/jdbc/>

For your LoanProcessing source file also *extend* **BankRecords**. Include in main() your readData() method call which will process your BankRecord objects. Then instantiate a Dao object and trigger your createTable(), insertRecords(your BankRecords array object) and retrieveRecords() methods in that order.

Once you have retrieved a recordset, print out all the records from the recordset to the console in a nice columnar format included with heading names for id, income and pep. Doesn’t hurt to have a title too, like Loan Analysis Report.

Extra Credit options

-Include SQL Prepared statements when inserting records (+5 points)

-Include a JTable GUI output display of your recordset data (+10 points)

For a JTable to trigger code build a frame with Javax Swing components which will display your loan data into a JTable. Column headings must be included as well as a scrollbar for proper appearance of your display within a GUI frame showing your table component.

Include all your project source code (new and old) in an orderly fashion and a snapshot of your app at runtime which must include a table creation message, an insertion message and the first few rows of your record results showing the Loan Analysis Report data into a doc file as well your entire app in a zip file format into BB for credit.