***--Db510***

**PROJECT Bank record loan generations**

**Objective** To write a program with a MVC *simulated* approach that performs a Loan analysis from class objects created in lab #2.

***PROJECT DESCRIPTION***

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

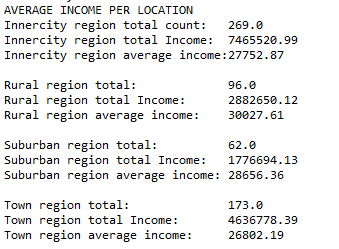
Use JDBC and a GUI to store and present Loan analysis data contained in a MySQL database. Perform the loan analysis as follows for this lab.

***Testcases Screenshot:***

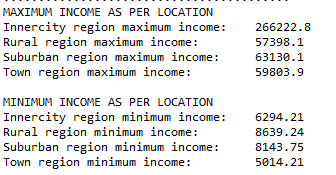
**Program Execution Begins:**



**Displaying Average Income per Location:**

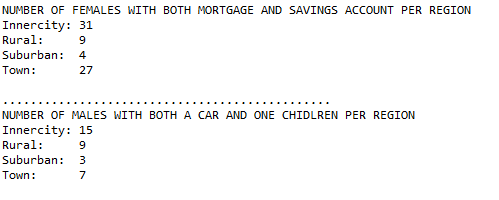


**Displaying Minimum and Maximum per Location and also Age:**

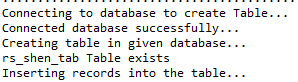


**Displaying Number of Females with a Mortgage and Savings per location**

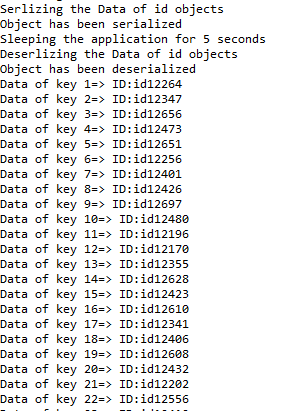
**Displaying Number of Males with both Car and 1 Child per location**



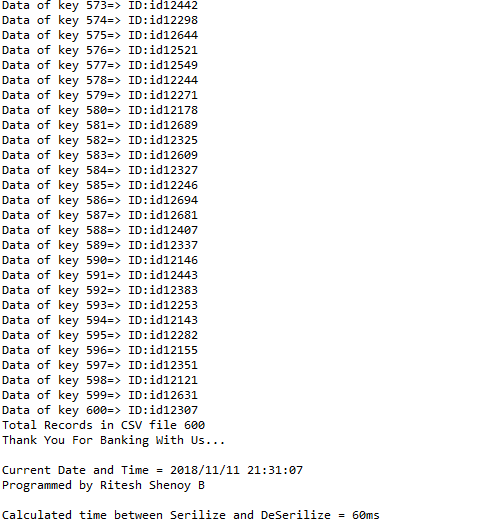
**Records Inserted into Papa Server and DB table**



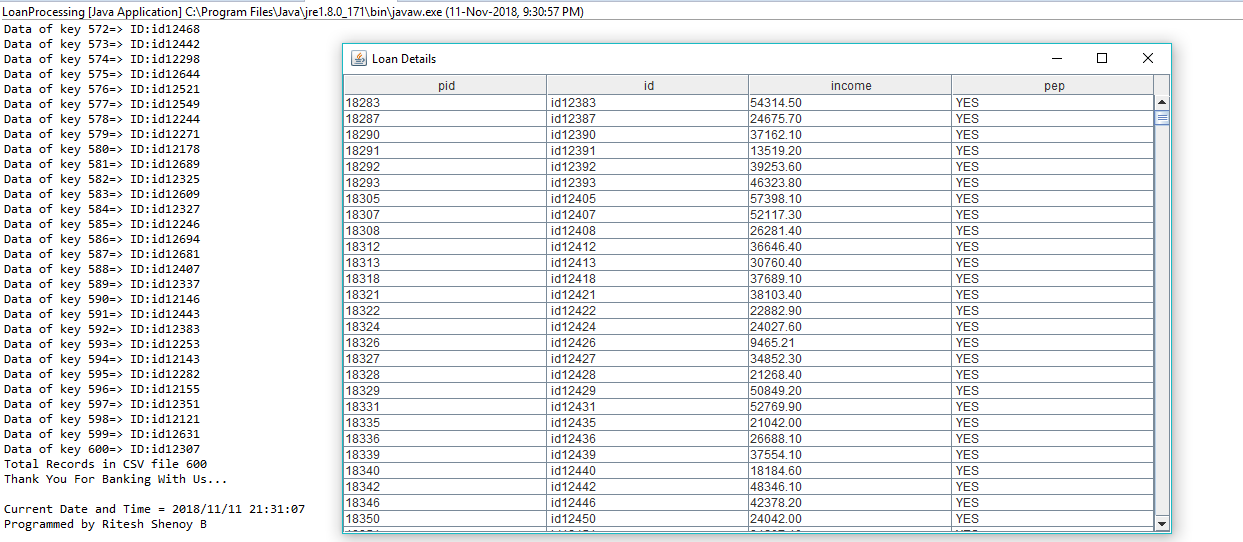
**Serialization and Map value for each ID:**



**Map for all 600 ID**



**JFrame To Display the records**



**Source Code:**

**LocalProcessing.java**

**package** controllers;

**import** java.sql.ResultSet;

**import** BankRecords.BankRecords;

**import** models.DaoModel;

**import** views.LoanView;

**import** BankRecords.Records;

**public** **class** LoanProcessing **extends** BankRecords {

**public** **static** **void** main(String[] args) {

BankRecords br = **new** BankRecords();

br.readDataFromFile();

Records rec = **new** Records();

rec.*RegionAverageComp*(); // analyze average income per loc

rec.*MaxMinIncomeComp*(); //compare max and min incomes per loc

rec.*FemalesWithMoratgeAndSavingsAccountComp*(); // analyze females w. mort/savings accounts per loc

rec.*MalesWithCarAndChildComp*(); // analyze male count w. car and 1 child per loc

rec.*MaxMinAge*();

DaoModel dao = **new** DaoModel();

dao.createTable();

dao.insertRecords(*arrayOfBankRecordObjects*); // perform inserts

ResultSet rs = dao.retrieveRecords();

**new** LoanView().runView(rs);

Serialize serialize = **new** Serialize();

System.***out***.print("Serlizing the Data of id objects\n");

serialize.Serlize();

System.***out***.print("Sleeping the application for 5 seconds\n");

**try** {

Thread.*sleep*(5);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.print("Deserlizing the Data of id objects\n");

serialize.deSerilize();

//Calculate the time

serialize.calculateTime();

}

}

**Serialize.java**

package controllers;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.HashMap;

import java.util.Map;

import BankRecords.BankRecords;

public class Serialize extends BankRecords {

String filename = "bankrecords.ser";

Map<Integer,String> dataToSerlize = new HashMap<Integer, String>();

//To get the current timestamp

String timeStamp = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(Calendar.getInstance().getTime());

private long endTime;

private long startTime;

public void Serlize() {

for (int i = 0; i <arrayOfBankRecordObjects.length; i++) {

dataToSerlize.put(i, arrayOfBankRecordObjects[i].getId());

}

// Serialization

try{

//Saving of object in a file

FileOutputStream file = new FileOutputStream(filename);

ObjectOutputStream out = new ObjectOutputStream(file);

// Method for serialization of object

out.writeObject(dataToSerlize);

out.close();

file.close();

startTime = System.nanoTime();

System.out.println("Object has been serialized");

}

catch(IOException ex) {

System.out.println("IOException is caught");

}

}

public void deSerilize() {

int len =1;

//Deserialization

try

{

// Reading the object from a file

FileInputStream file = new FileInputStream(filename);

ObjectInputStream in = new ObjectInputStream(file);

// Method for deserialization of object

@SuppressWarnings("rawtypes")

Map dataDeserilize = (Map)in.readObject();

in.close();

file.close();

endTime = System.nanoTime();

System.out.println("Object has been deserialized ");

for (int i = 0; i <arrayOfBankRecordObjects.length; i++) {

System.out.println("Data of key " + len + "=> ID:" + dataDeserilize.get(i));

len++;

}

//To display the total records

System.out.println("Total Records in CSV file " +arrayOfBankRecordObjects.length);

//End Statement

System.out.println("Thank You For Banking With Us... \n");

//To display the current timestamp and the Programmer info

System.out.println("Current Date and Time = " + timeStamp + "\nProgrammed by Ritesh Shenoy B\n");

}

catch(IOException ex) {

System.out.println("IOException is caught");

}

catch(ClassNotFoundException ex) {

System.out.println("ClassNotFoundException is caught");

}

}

public void calculateTime() {

long totalTime = endTime - startTime;

System.out.print("Calculated time between Serilize and DeSerilize = " +totalTime/1000000 + "ms");

}

}

**DBConnect.java**

package models;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnect {

// Code database URL

static final String DB\_URL = "jdbc:mysql://www.papademas.net:3307/510labs?autoReconnect=true&useSSL=false";

// Database credentials

static final String USER = "db510", PASS = "510";

public Connection connect() throws SQLException {

return DriverManager.getConnection(DB\_URL, USER, PASS);

}

}

**DaoModel.java**

package models;

import java.sql.DatabaseMetaData;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import BankRecords.BankRecords;

public class DaoModel {

//Declare DB objects

DBConnect conn = null;

Statement stmt = null;

// constructor

public DaoModel() { //create db object instance

conn = new DBConnect();

}

// CREATE TABLE METHOD

public void createTable() {

try {

// Open a connection

System.out.println("Connecting to database to create Table...");

System.out.println("Connected database successfully...");

// Execute create query

System.out.println("Creating table in given database...");

//To Check whether the table exists or not

DatabaseMetaData dbm = conn.connect().getMetaData();

ResultSet rs = dbm.getTables(null, null, "rs\_shen\_tab", null);

if (rs.next()) {

System.out.println("rs\_shen\_tab Table exists");

} else {

System.out.println("Table does not exist");

stmt = conn.connect().createStatement();

String sql = "CREATE TABLE rs\_shen\_tab " +

"(pid INTEGER not NULL AUTO\_INCREMENT, " +

" id VARCHAR(10), " +

" income numeric(8,2), " +

" pep VARCHAR(4), " +

" PRIMARY KEY ( pid ))";

stmt.executeUpdate(sql);

System.out.println("Created table in given database...");

conn.connect().close(); //close db connection

}

}catch (SQLException se) { // Handle errors for JDBC

se.printStackTrace();

}

}

// INSERT INTO METHOD

public void insertRecords(BankRecords[] robjs) {

try {

// Execute a query

System.out.println("Inserting records into the table...");

stmt = conn.connect().createStatement();

String sql = null;

// Include all object data to the database table

for (int i = 0; i < robjs.length; ++i) {

// finish string assignment to insert all object data

// (id, income, pep) into your database table

sql = "INSERT INTO rs\_shen\_tab(id, income, pep) " +

"VALUES (' "+robjs[i].getId()+" ', ' "+

robjs[i].getIncome()+" ', ' "+

robjs[i].getPep()+" ' )";

stmt.executeUpdate(sql);

}

conn.connect().close();

} catch (SQLException se) { se.printStackTrace(); }

}// INSERT INTO METHOD

public ResultSet retrieveRecords() {

ResultSet rs = null;

try {

stmt = conn.connect().createStatement();

String sql = "SELECT \* from rs\_shen\_tab order by pep desc";

rs = stmt.executeQuery(sql);

conn.connect().close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return rs;

}

}

**LoanView.java**

package views;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.util.Vector;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

public class LoanView {

public void runView(ResultSet rs) {

// instantiate vector objects to hold column/row data for JTable

Vector<Vector<Object>> data = new Vector<Vector<Object>>();

Vector<String> column = new Vector<String>();

try {

ResultSetMetaData metaData = rs.getMetaData();

int columns = metaData.getColumnCount();

// get column names from table!

String cols = "";

for (int i = 1; i <= columns; i++) {

cols = metaData.getColumnName(i);

column.add(cols);

}

// get row data from table!

while (rs.next()) {

Vector<Object> row = new Vector<Object>(columns);

for (int i = 1; i <= columns; i++)

row.addElement(rs.getObject(i));

data.addElement(row);

}

DefaultTableModel model = new DefaultTableModel(data, column);

JTable table = new JTable(model);

JFrame frame = new JFrame("Loan Details");

frame.setSize(700, 200);

frame.add(new JScrollPane(table));

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.pack();

frame.setVisible(true);

rs.close(); //close ResultSet instance

} catch (SQLException e) { e.printStackTrace(); }

}

}

**MVC Design Pattern:**

