**JAVA SWINGS BASED- DRINKSBUCKS STORE**

**- SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

# INFORMATION TECHNOLOGY

By

Chemudu Sreeya <1602-19-737-110 >

Under the guidance of Ms B. Leelavathy



Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

# BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘**DRINKSBUCKS STORE’** is a project work of Ms. Chemudu Sreeya bearing roll no. 1602-19-737-110 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature Signature

Internal Examiner External Examiner

# ABSTRACT

Drinksbucks is a store which is designed to ease customers life. sometimes you feel like having some liquids or doesn’t feel like to make your own drink or coffee or juice. So, drinksbucks is here to help you. In this drinksbucks, customers need to select their drink from the menu and then make an order. employees in the store will provide the order which the customer have ordered and make the bill payments accordingly. Drinks can be made within the store or raw materials needed can be supplied by a supplier. Here, every person involving has their separate id’s and information which is stored separately in the database named drinksbucks.

# INTRODUCTION

## REQUIREMENTS

|  |  |
| --- | --- |
| Table name | Attributes |
| Employee | employee\_id number empname varchar2(25)  age number  contact number |
| Customer | customer\_id number  customer\_name varchar2(25)  gender varchar2(10)  contactno number |
| Supplier | supplier\_id number  supplier\_name varchar2(20)  contactno number  mailaddress varchar2(20) |
| Drinks | drinks\_id number  drinks\_name varchar2(25)  quantity number  price number |
| Torder | order\_id number  customer\_id number  drinks\_id number  qty number  price\_perqty number |
| Payments | payment\_id number  order\_id number  customer\_id number  ttlquantity number  ttlprice number |
| Transaction | transaction\_id number  customer\_id number  employee\_id number  order\_id number  drinks\_id number  payment\_id number |

## AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based drinksbucks store which takes the values like: customer id, drinks id etc. from the customer who orders their choice from the given drinks, it also stores information about the employees, suppliers, payments and orders. These values are to be updated in the database using JDBC connectivity .

## ARCHITECTURE AND TECHNOLOGY

**Software used:** Java Eclipse, Oracle 11g Database, Java SE version 13, SQL\*Plus.

**Java AWT:**

**Java AWT**

**SWING** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

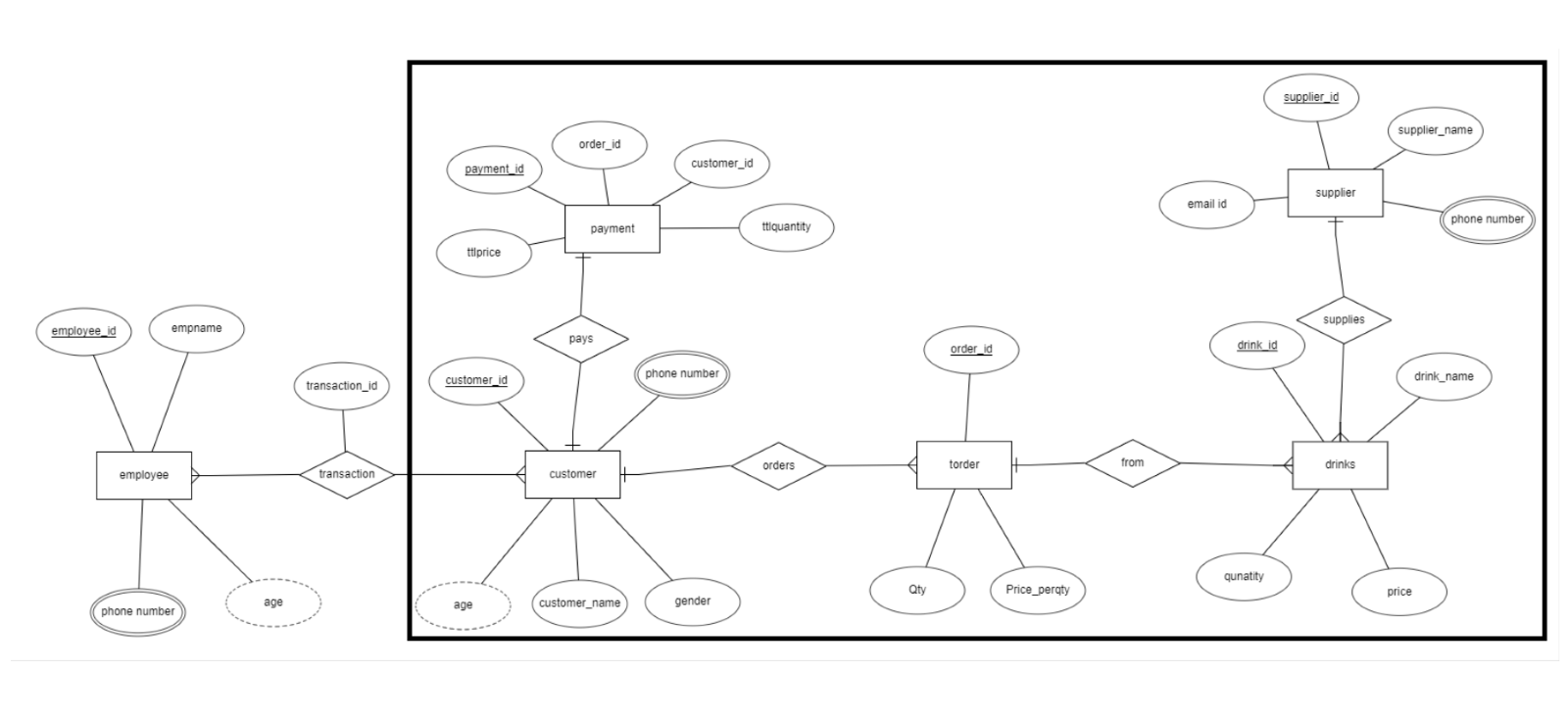
Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

**SQL:**

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

## DESIGN

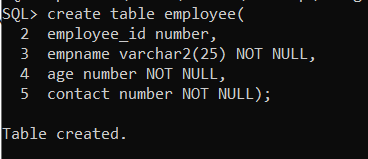
**Entity relationship diagram**

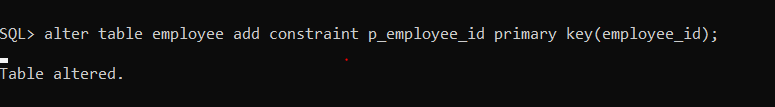


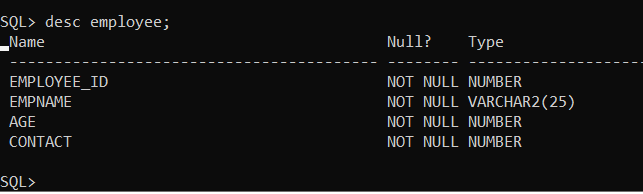
**Database Design:**

DDL Operations

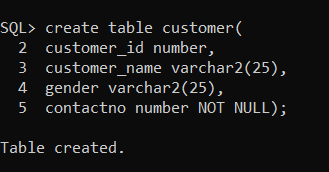
Employee table:

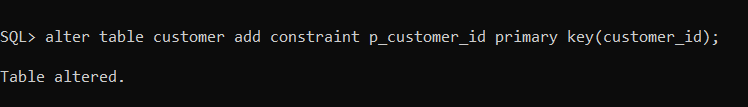


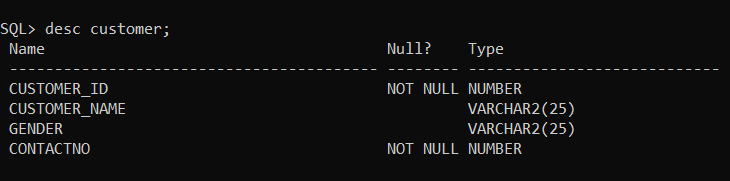




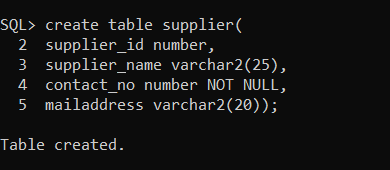
Customer table:

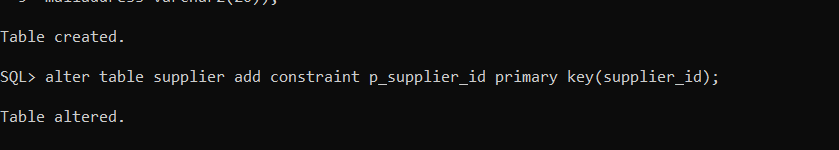


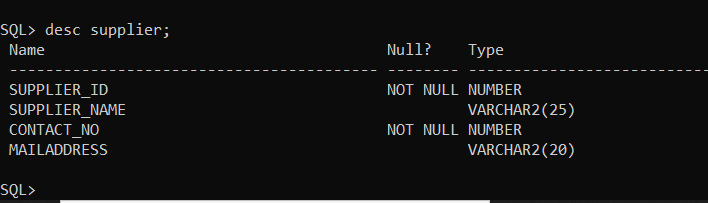




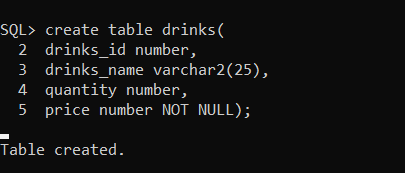
Supplier table:

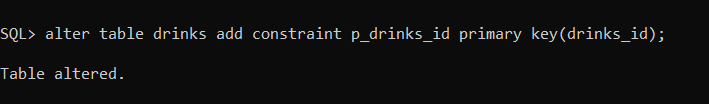


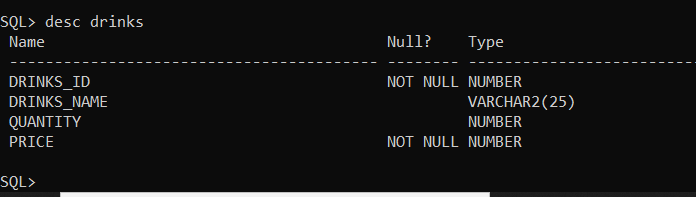




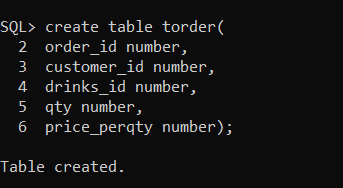
Drinks table:

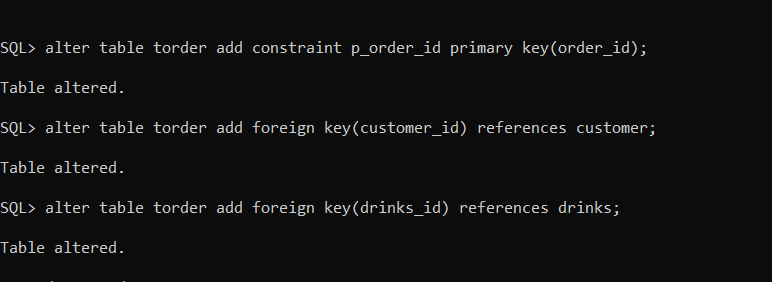


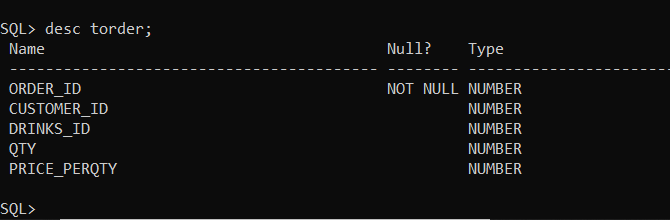




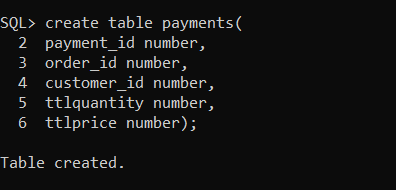
Order table:

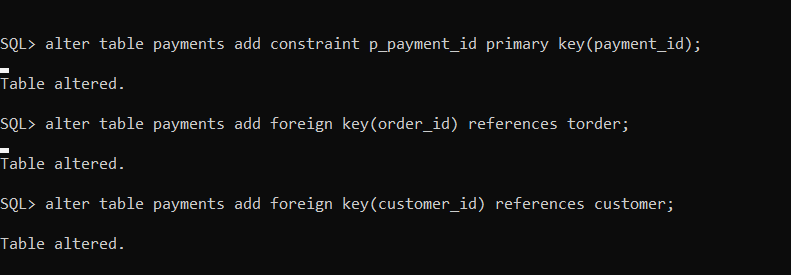


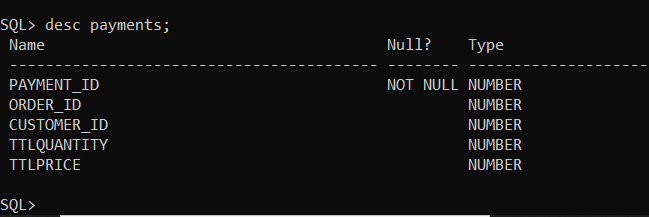




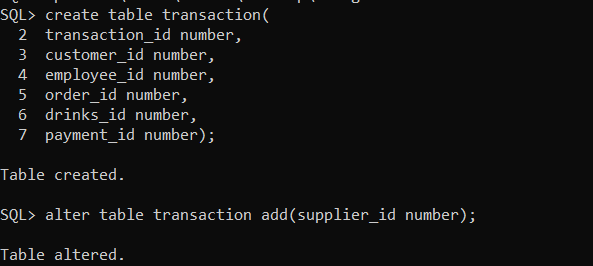
Payment table:

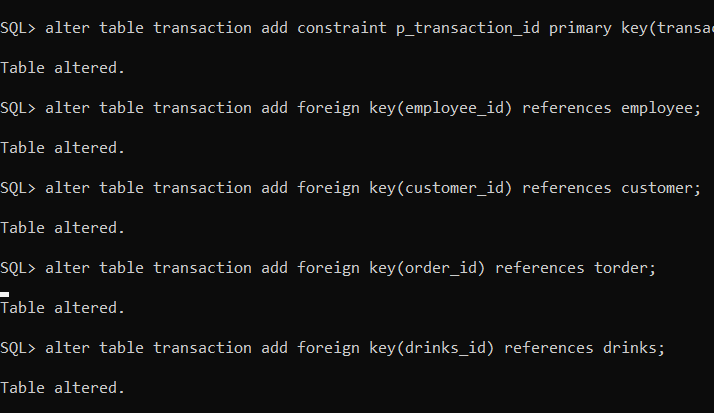


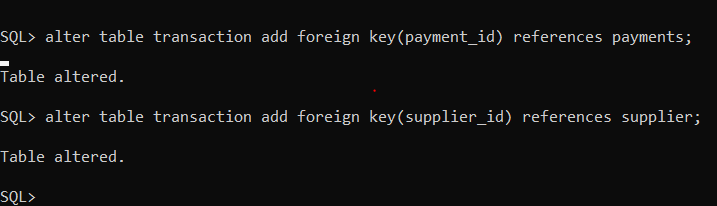




Transaction table:

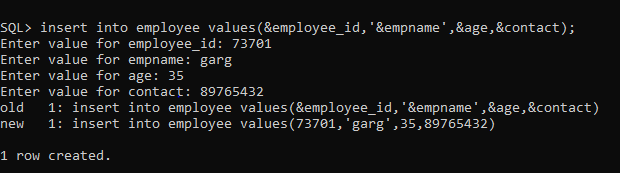


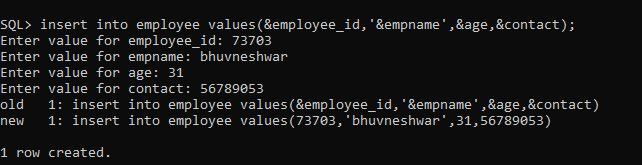


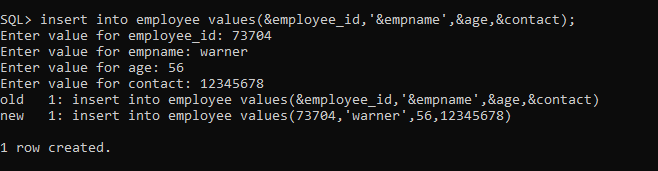


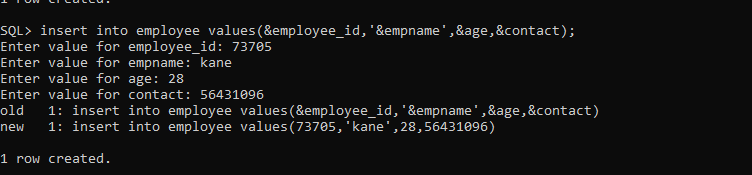
DML Operations

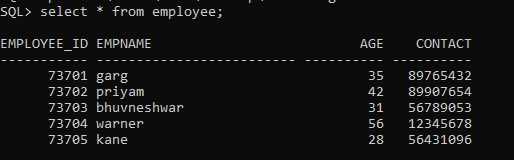
Employee table:



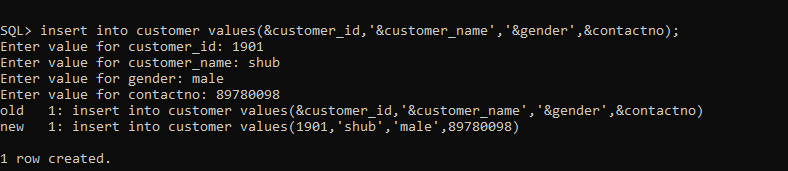


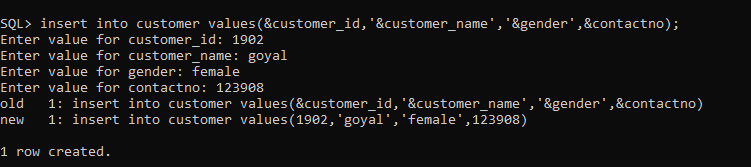


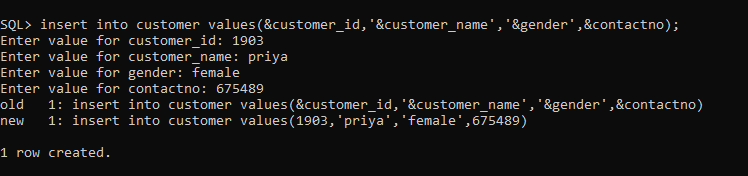


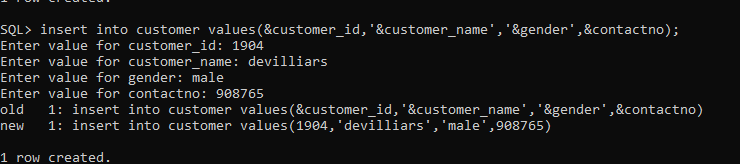


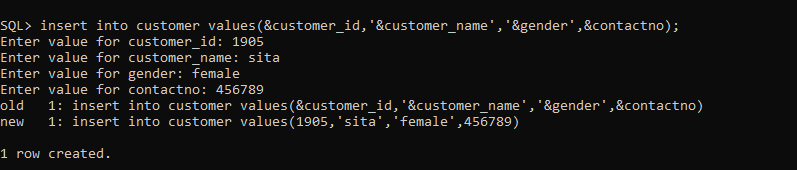
Customer table:

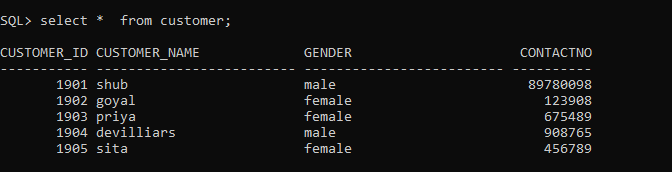




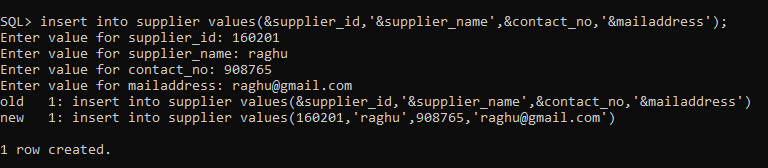


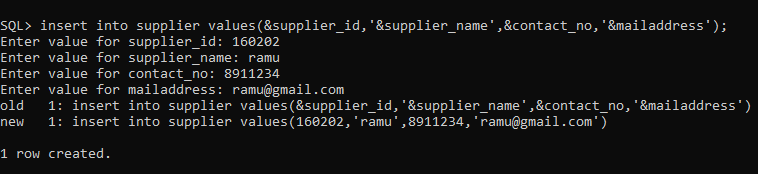


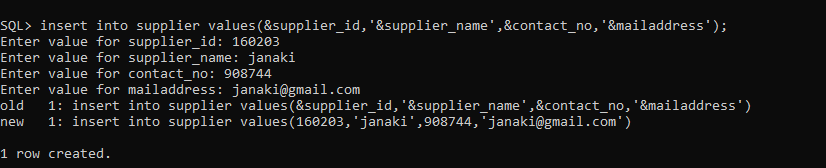


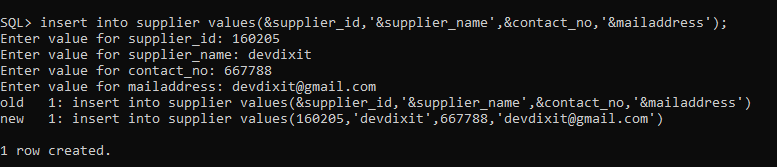
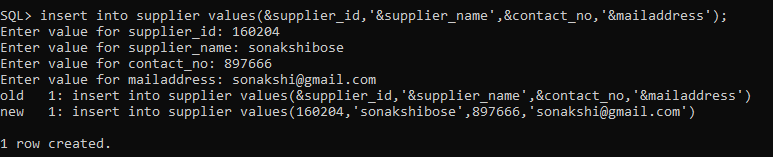


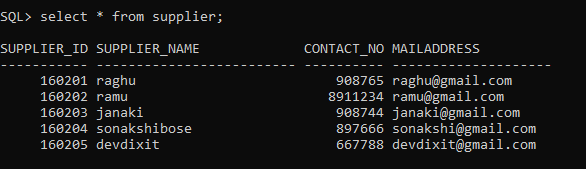
Supplier table:





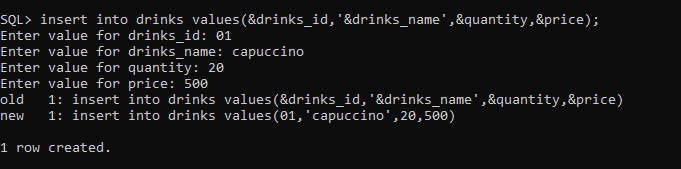


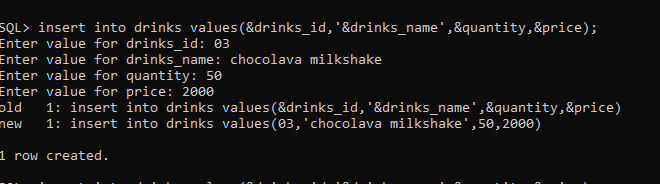
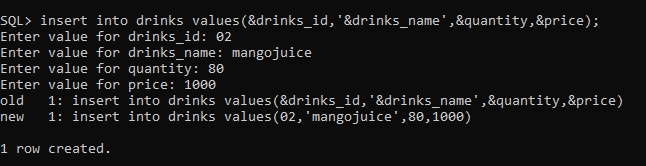


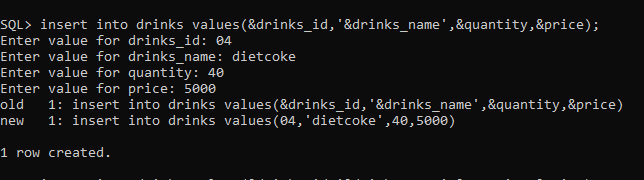


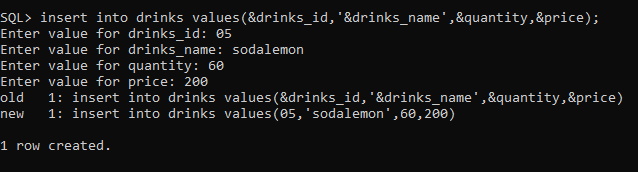
INFORMATION RETRIEVAL OF GOOGLE QUERIES OF POSITIVE AND NEGATIVE FEEDBACK

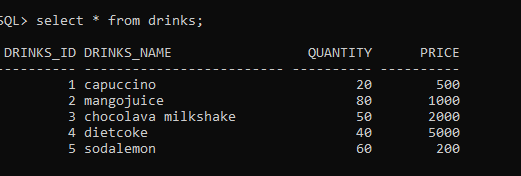
Drinks table:



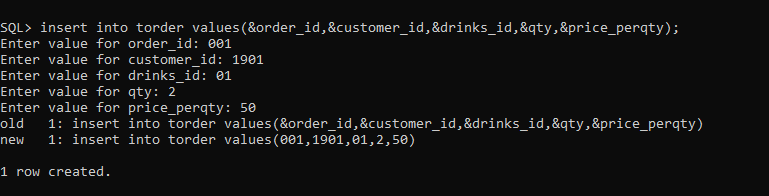


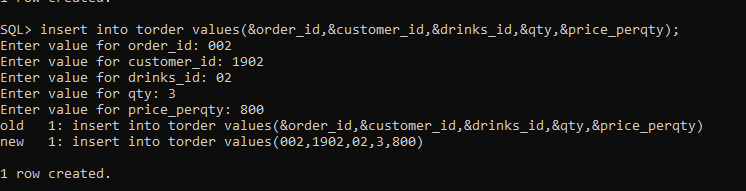


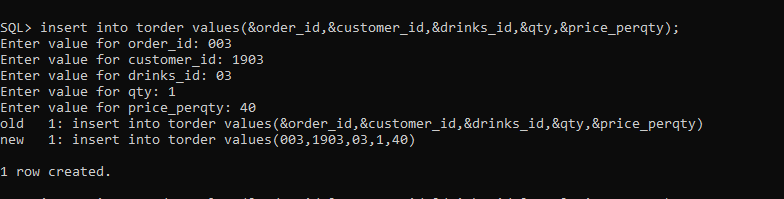


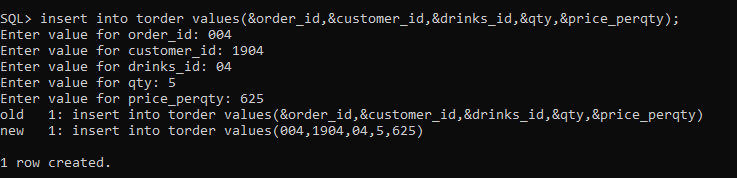


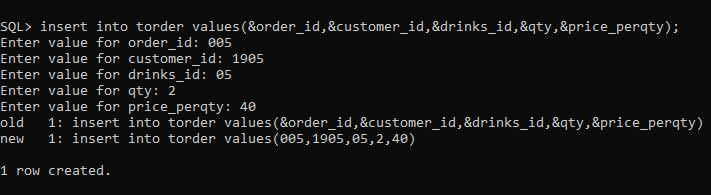
Order table:

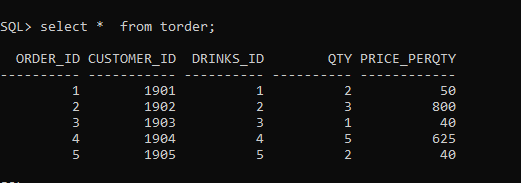




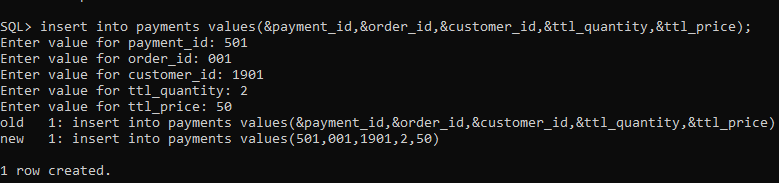


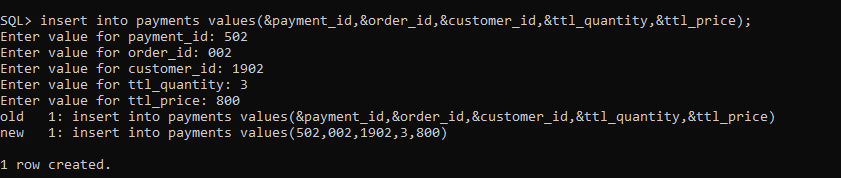


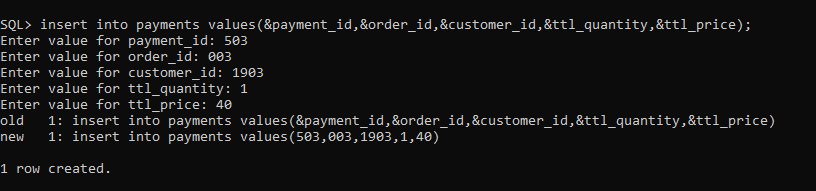


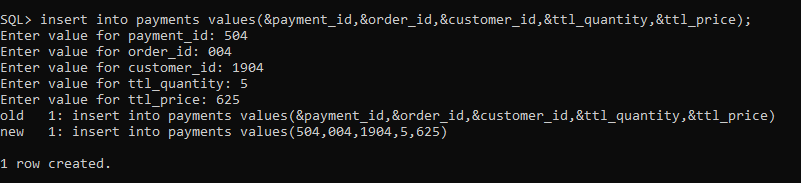


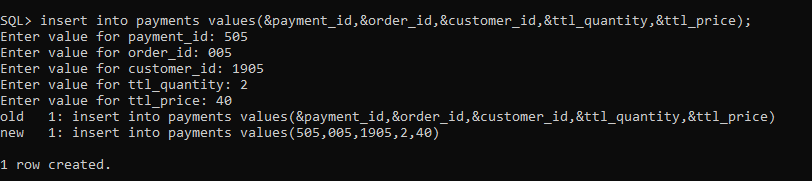
Payment table:

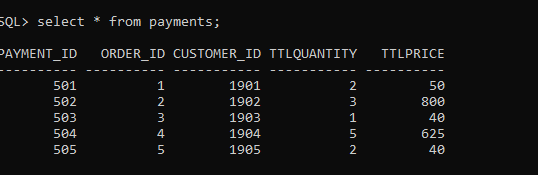




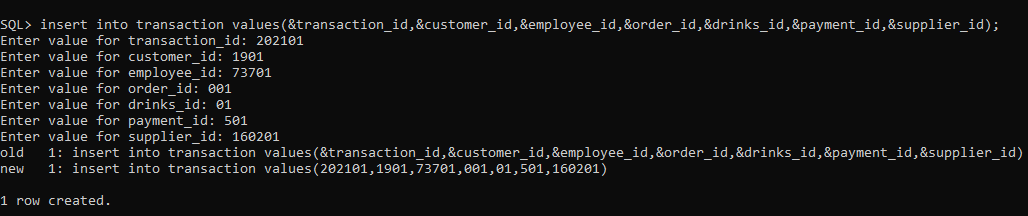


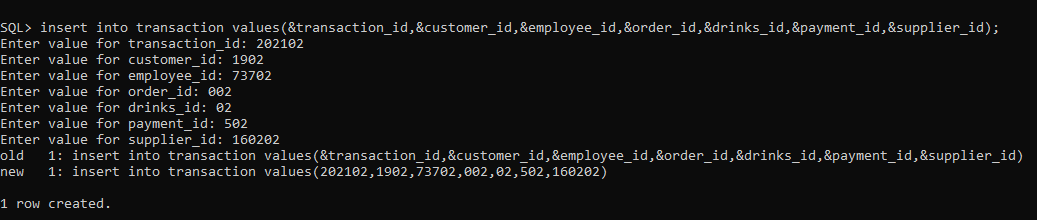


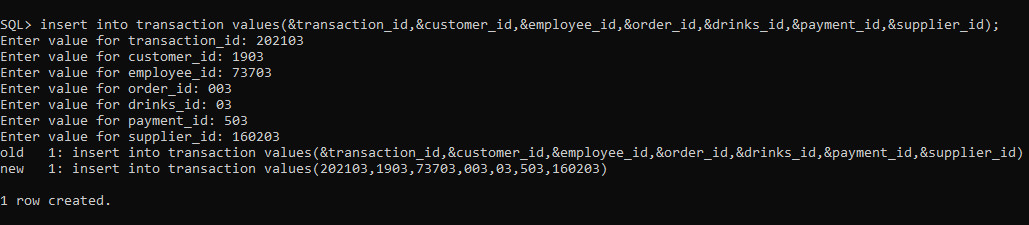


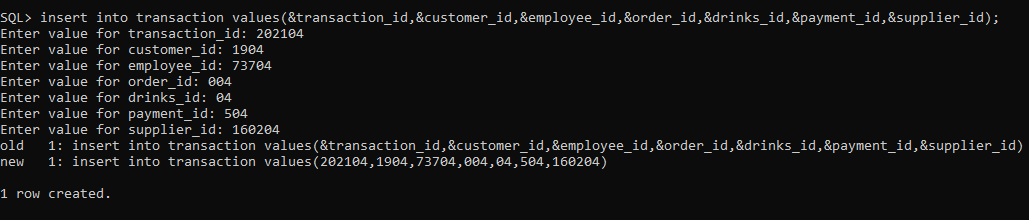


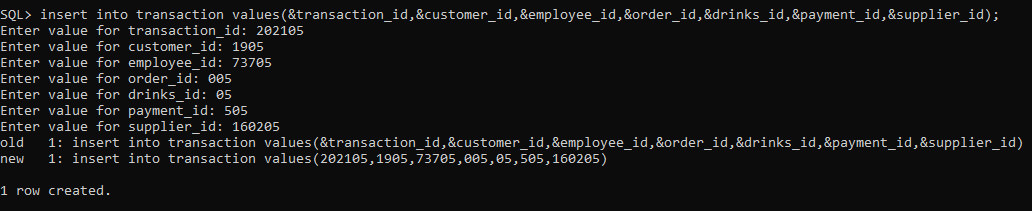
Transaction table:

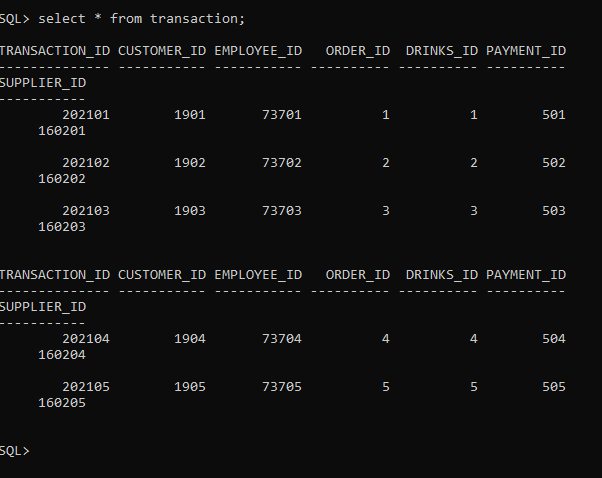












## IMPLEMENTATION

**Front end programs and its connectivity**

**Java Database Connectivity** (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

**public void** connectToDB()

{

**try**

{

connection =

DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi"); statement = connection.createStatement();

}

**catch** (SQLException connectException)

{

System.***out***.println(connectException.getMessage())

;

System.***out***.println(connectException.getSQLState()

);

System.***out***.println(connectException.getErrorCode());

System.*exit*(1);

}

}

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

PROGRAMS:

Employee:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class Emp{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

private JLabel lblemployee\_id,lblempname,lblage,lblcontact;

private JTextField txtemployee\_id,txtempname,txtage,txtcontact;

private List MIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,update,view;

public Emp(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

this.update=update;

this.view=view;

lblemployee\_id=new JLabel("employee\_id");

lblempname=new JLabel("empname");

lblage=new JLabel("age");

lblcontact=new JLabel("contact");

txtemployee\_id=new JTextField(15);

txtempname=new JTextField(15);

txtage=new JTextField(8);

txtcontact=new JTextField(15);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loademployees() {

try {

MIDList.removeAll();

rs=statement.executeQuery("select \* from employee");

while(rs.next()) {

MIDList.add(rs.getString("employee\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

txtemployee\_id.setText(null);

txtempname.setText(null);

txtage.setText(null);

txtcontact.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblemployee\_id);

p1.add(txtemployee\_id);

p1.add(lblempname);

p1.add(txtempname);

p1.add(lblage);

p1.add(txtage);

p1.add(lblcontact);

p1.add(txtcontact);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

MIDList=new List(10);

loademployees();

p2.add(MIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO employee VALUES("+txtemployee\_id.getText()+",'"+txtempname.getText()+"',"+txtage.getText()+",'"+txtcontact.getText()+"')";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");loademployees();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txtemployee\_id.setText(null);

txtempname.setText(null);

txtage.setText(null);

txtcontact.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblemployee\_id);

p1.add(txtemployee\_id);

p1.add(lblempname);

p1.add(txtempname);

p1.add(lblage);

p1.add(txtage);

p1.add(lblcontact);

p1.add(txtcontact);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

MIDList=new List(10);

loademployees();

p2.add(MIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

MIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from employee");

StringTokenizer st=new StringTokenizer(MIDList.getSelectedItem(),"->");

String p=st.nextToken();

while (rs.next())

{

if (rs.getString("employee\_id").equals(p))

break;

}

if (!rs.isAfterLast())

{

txtemployee\_id.setText(rs.getString("employee\_id"));

txtempname.setText(rs.getString("empname"));

txtage.setText(rs.getString("age"));

txtcontact.setText(rs.getString("contact"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

StringTokenizer st=new StringTokenizer(MIDList.getSelectedItem(),"->");

String query="DELETE FROM employee WHERE employee\_id="+st.nextToken();;

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loademployees();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("modify");

txtemployee\_id.setText(null);

txtempname.setText(null);

txtage.setText(null);

txtcontact.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblemployee\_id);

p1.add(txtemployee\_id);

p1.add(lblempname);

p1.add(txtempname);

p1.add(lblage);

p1.add(txtage);

p1.add(lblcontact);

p1.add(txtcontact);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

MIDList=new List(10);

loademployees();

p2.add(MIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

MIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from employee");

while (rs.next())

{

if (rs.getString("employee\_id").equals(MIDList.getSelectedItem()))

break;

}

if (!rs.isAfterLast())

{

txtemployee\_id.setText(rs.getString("employee\_id"));

txtempname.setText(rs.getString("empname"));

txtage.setText(rs.getString("age"));

txtcontact.setText(rs.getString("contact"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loademployees();

String age=JOptionPane.showInputDialog(p,"enter the age");

txtage.setText(age);

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

String query="update employee set age='"+age+"'where employee\_id="+txtemployee\_id.getText();

// int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

// if(a==JOptionPane.YES\_OPTION){

//String query="update employee set empname='"+txtempname.getText()+"',age="+txtage.getText()+",contact='"+txtcontact.getText()+"' WHERE employee\_id="+MIDList.getSelectedItem();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loademployees();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("employees view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2); p.setLayout(new FlowLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("Medicine details");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("employee id");

model.addColumn("empname");

model.addColumn("age");

model.addColumn("contact");

try {

rs=statement.executeQuery("select \* from employee");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("employee\_id"), rs.getString("empname"),rs.getString("age"),rs.getString("contact")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 300, 300);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(800, 400);

f.setVisible(true);

}

});

}

});

}

}

Customer:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class Cust{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

private JLabel lblcustomer\_id,lblcustomer\_name,lblgender,lblcontactno;

private JTextField txtcustomer\_id,txtcustomer\_name,txtgender,txtcontactno;

private List OrderIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,update,view;

public Cust(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

this.update=update;

this.view=view;

lblcustomer\_id=new JLabel("customer id");

lblcustomer\_name=new JLabel("customer name");

lblgender=new JLabel("gender");

lblcontactno=new JLabel("contactno");

txtcustomer\_id=new JTextField(15);

txtcustomer\_name=new JTextField(15);

txtgender=new JTextField(8);

txtcontactno=new JTextField(8);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loadcustomers() {

try {

OrderIDList.removeAll();

rs=statement.executeQuery("select \* from customer");

while(rs.next()) {

OrderIDList.add(rs.getString("customer\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

txtcustomer\_id.setText(null);

txtcustomer\_name.setText(null);

txtgender.setText(null);

txtcontactno.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblcustomer\_name);

p1.add(txtcustomer\_name);

p1.add(lblgender);

p1.add(txtgender);

p1.add(lblcontactno);

p1.add(txtcontactno);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p3.setBounds(200,350,75,35); p1.setBounds(115,80,300,250);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

OrderIDList=new List(10);

loadcustomers();

p2.add(OrderIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO customer VALUES("+txtcustomer\_id.getText()+",'"+txtcustomer\_name.getText()+"','"+txtgender.getText()+"',"+txtcontactno.getText()+")";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

loadcustomers();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txtcustomer\_id.setText(null);

txtcustomer\_name.setText(null);

txtgender.setText(null);

txtcontactno.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblcustomer\_name);

p1.add(txtcustomer\_name);

p1.add(lblgender);

p1.add(txtgender);

p1.add(lblcontactno);

p1.add(txtcontactno);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p3.setBounds(200,350,75,35); p1.setBounds(115,80,300,250);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

OrderIDList=new List(10);

loadcustomers();

p2.add(OrderIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

OrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from customer");

StringTokenizer st=new StringTokenizer(OrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

while (rs.next())

{

if (rs.getString("customer\_id").equals(p))

break;

}

if (!rs.isAfterLast())

{

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtcustomer\_name.setText(rs.getString("customer\_name"));

txtgender.setText(rs.getString("gender"));

txtcontactno.setText(rs.getString("contactno"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

StringTokenizer st=new StringTokenizer( OrderIDList.getSelectedItem(),"->");

String query="DELETE FROM customer WHERE customer\_id="+st.nextToken();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loadcustomers();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("modify");

txtcustomer\_id.setText(null);

txtcustomer\_name.setText(null);

txtgender.setText(null);

txtcontactno.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblcustomer\_name);

p1.add(txtcustomer\_name);

p1.add(lblgender);

p1.add(txtgender);

p1.add(lblcontactno);

p1.add(txtcontactno);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p3.setBounds(200,350,75,35); p1.setBounds(115,80,300,250);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

OrderIDList=new List(10);

loadcustomers();

p2.add(OrderIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

OrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from customer");

while (rs.next())

{

if (rs.getString("customer\_id").equals(OrderIDList.getSelectedItem()))

break;

}

if (!rs.isAfterLast())

{

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtcustomer\_name.setText(rs.getString("customer\_name"));

txtgender.setText(rs.getString("gender"));

txtcontactno.setText(rs.getString("contactno"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loadcustomers();

String contactno=JOptionPane.showInputDialog(p,"enter the contactno");

txtcontactno.setText(contactno);

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

String query="update customer set contactno='"+contactno+"'where customer\_id="+txtcustomer\_id.getText();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadcustomers();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("customer view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2);

p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("Customer Details ");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("customer id");

model.addColumn("customer name");

model.addColumn("gender");

model.addColumn("contactno");

try {

rs=statement.executeQuery("select \* from customer");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("customer\_id"), rs.getString("customer\_name"),rs.getString("gender"),rs.getString("contactno")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 300, 300);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(800, 400);

f.setVisible(true);

}

});

}

});

}

}

Supplier:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class Supp{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

//private JLabel lblorder\_id,lbltot,lblstatus;

private JLabel lblsupplier\_id,lblsupplier\_name,lblcontact\_no,lblmailaddress;

//private JTextField txtorder\_id,txttot,txtstatus;

private JTextField txtsupplier\_id,txtsupplier\_name,txtcontact\_no,txtmailaddress;

private List OrderIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,update,view;

public Supp(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

this.update=update;

this.view=view;

lblsupplier\_id=new JLabel("supplier id");

lblsupplier\_name=new JLabel("supplier name");

lblcontact\_no=new JLabel("contact no");

lblmailaddress=new JLabel("mail address");

txtsupplier\_id=new JTextField(15);

txtsupplier\_name=new JTextField(15);

txtcontact\_no=new JTextField(8);

txtmailaddress=new JTextField(8);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loadsuppliers() {

try {

OrderIDList.removeAll();

rs=statement.executeQuery("select \* from supplier");

while(rs.next()) {

OrderIDList.add(rs.getString("supplier\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

txtsupplier\_id.setText(null);

txtsupplier\_name.setText(null);

txtcontact\_no.setText(null);

txtmailaddress.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblsupplier\_id);

p1.add(txtsupplier\_id);

p1.add(lblsupplier\_name);

p1.add(txtsupplier\_name);

p1.add(lblcontact\_no);

p1.add(txtcontact\_no);

p1.add(lblmailaddress);

p1.add(txtmailaddress);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p3.setBounds(200,350,75,35); p1.setBounds(115,80,300,250);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

OrderIDList=new List(10);

loadsuppliers();

p2.add(OrderIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO supplier VALUES("+txtsupplier\_id.getText()+",'"+txtsupplier\_name.getText()+"',"+txtcontact\_no.getText()+",'"+txtmailaddress.getText()+"')";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

loadsuppliers();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txtsupplier\_id.setText(null);

txtsupplier\_name.setText(null);

txtcontact\_no.setText(null);

txtmailaddress.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblsupplier\_id);

p1.add(txtsupplier\_id);

p1.add(lblsupplier\_name);

p1.add(txtsupplier\_name);

p1.add(lblcontact\_no);

p1.add(txtcontact\_no);

p1.add(lblmailaddress);

p1.add(txtmailaddress);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p3.setBounds(200,350,75,35); p1.setBounds(115,80,300,250);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

OrderIDList=new List(10);

loadsuppliers();

p2.add(OrderIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

OrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from supplier");

StringTokenizer st=new StringTokenizer(OrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

while (rs.next())

{

if (rs.getString("supplier\_id").equals(p))

break;

}

if (!rs.isAfterLast())

{

txtsupplier\_id.setText(rs.getString("supplier\_id"));

txtsupplier\_name.setText(rs.getString("supplier\_name"));

txtcontact\_no.setText(rs.getString("contact\_no"));

txtmailaddress.setText(rs.getString("mailaddress"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

StringTokenizer st=new StringTokenizer(OrderIDList.getSelectedItem(),"->");

String query="DELETE FROM supplier WHERE supplier\_id="+st.nextToken();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loadsuppliers();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("modify");

txtsupplier\_id.setText(null);

txtsupplier\_name.setText(null);

txtcontact\_no.setText(null);

txtmailaddress.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lblsupplier\_id);

p1.add(txtsupplier\_id);

p1.add(lblsupplier\_name);

p1.add(txtsupplier\_name);

p1.add(lblcontact\_no);

p1.add(txtcontact\_no);

p1.add(lblmailaddress);

p1.add(txtmailaddress);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p3.setBounds(200,350,75,35); p1.setBounds(115,80,300,250);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

OrderIDList=new List(10);

loadsuppliers();

p2.add(OrderIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

OrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from supplier");

while (rs.next())

{

if (rs.getString("supplier\_id").equals(OrderIDList.getSelectedItem()))

break;

}

if (!rs.isAfterLast())

{

txtsupplier\_id.setText(rs.getString("supplier\_id"));

txtsupplier\_name.setText(rs.getString("supplier\_name"));

txtcontact\_no.setText(rs.getString("contact\_no"));

txtmailaddress.setText(rs.getString("mailaddress"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loadsuppliers();

String contact\_no=JOptionPane.showInputDialog(p,"enter the contactno");

txtcontact\_no.setText(contact\_no);

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

String query="update supplier set contact\_no='"+contact\_no+"'where supplier\_id="+txtsupplier\_id.getText();

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

//String query="update supplier set supplier\_name="+txtsupplier\_name.getText()+",contact\_no="+txtcontact\_no.getText()+",mailaddress='"+txtmailaddress.getText()+"' WHERE supplier\_id="+OrderIDList.getSelectedItem();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadsuppliers();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("suppliers view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2);

p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("orderDetails ");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("supplier id");

model.addColumn("supplier name");

model.addColumn("contact no");

model.addColumn("mail address");

try {

rs=statement.executeQuery("select \* from supplier");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("supplier\_id"), rs.getString("supplier\_name"),rs.getString("contact\_no"),rs.getString("mailaddress")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 300, 300);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(800, 400);

f.setVisible(true);

}

});

}

});

}

}

Drinks:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class Drinks{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

//private JLabel lblmid,lblmname,lblprice,lblinstock;

private JLabel lbldrinksid,lbldrinksname,lblquantity,lblprice;

//private JTextField txtmid,txtmname,txtprice,txtinstock;

private JTextField txtdrinksid,txtdrinksname,txtquantity,txtprice;

private List MIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,update,view;

public Drinks(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

this.update=update;

this.view=view;

lbldrinksid=new JLabel("Drinks id");

lbldrinksname=new JLabel("Drinks Name");

lblquantity=new JLabel("quantity");

lblprice=new JLabel("price");

txtdrinksid=new JTextField(15);

txtdrinksname=new JTextField(15);

txtquantity=new JTextField(15);

txtprice=new JTextField(8);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loaddrinks() {

try {

MIDList.removeAll();

rs=statement.executeQuery("select \* from drinks ");

while(rs.next()) {

MIDList.add(rs.getString("drinks\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

txtdrinksid.setText(null);

txtdrinksname.setText(null);

txtquantity.setText(null);

txtprice.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lbldrinksid);

p1.add(txtdrinksid);

p1.add(lbldrinksname);

p1.add(txtdrinksname);

p1.add(lblquantity);

p1.add(txtquantity);

p1.add(lblprice);

p1.add(txtprice);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

MIDList=new List(10);

loaddrinks();

p2.add(MIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO drinks VALUES("+txtdrinksid.getText()+",'"+txtdrinksname.getText()+"',"+txtquantity.getText()+",'"+txtprice.getText()+"')";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");loaddrinks();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txtdrinksid.setText(null);

txtdrinksname.setText(null);

txtquantity.setText(null);

txtprice.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lbldrinksid);

p1.add(txtdrinksid);

p1.add(lbldrinksname);

p1.add(txtdrinksname);

p1.add(lblquantity);

p1.add(txtquantity);

p1.add(lblprice);

p1.add(txtprice);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

MIDList=new List(10);

loaddrinks();

p2.add(MIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

MIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from drinks");

StringTokenizer st=new StringTokenizer(MIDList.getSelectedItem(),"->");

String p=st.nextToken();

while (rs.next())

{

if (rs.getString("drinks\_id").equals(p))

break;

}

if (!rs.isAfterLast())

{

txtdrinksid.setText(rs.getString("drinks\_id"));

txtdrinksname.setText(rs.getString("drinks\_name"));

txtquantity.setText(rs.getString("quantity"));

txtprice.setText(rs.getString("price"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

StringTokenizer st=new StringTokenizer(MIDList.getSelectedItem(),"->");

String query="DELETE FROM DRINKS WHERE drinks\_id="+st.nextToken();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loaddrinks();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("modify");

txtdrinksid.setText(null);

txtdrinksname.setText(null);

txtquantity.setText(null);

txtprice.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(4,2));

p1.add(lbldrinksid);

p1.add(txtdrinksid);

p1.add(lbldrinksname);

p1.add(txtdrinksname);

p1.add(lblquantity);

p1.add(txtquantity);

p1.add(lblprice);

p1.add(txtprice);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

MIDList=new List(10);

loaddrinks();

p2.add(MIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);

p. add(p2);

MIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from drinks");

while (rs.next())

{

if (rs.getString("drinks\_id").equals(MIDList.getSelectedItem()))

break;

}

if (!rs.isAfterLast())

{

txtdrinksid.setText(rs.getString("drinks\_id"));

txtdrinksname.setText(rs.getString("drinks\_name"));

txtquantity.setText(rs.getString("quantity"));

txtprice.setText(rs.getString("price"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loaddrinks();

String price=JOptionPane.showInputDialog(p,"enter the new price");

txtprice.setText(price);

// int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

// if(a==JOptionPane.YES\_OPTION){

String query="update drinks set price='"+price+"'where drinks\_id="+txtdrinksid.getText();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loaddrinks();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("Drinks view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2); p.setLayout(new FlowLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("Drinks details");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("Drinks id");

model.addColumn("Drinks Name");

model.addColumn("quantity");

model.addColumn("price");

try {

rs=statement.executeQuery("select \* from drinks");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("drinks\_id"), rs.getString("drinks\_name"),rs.getString("quantity"),rs.getString("price")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 300, 300);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(800, 400);

f.setVisible(true);

}

});

}

});

}

}

Torder:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Choice;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class Order{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

//private JLabel lbluser\_id,lblmid,lblorder\_id,lblday;

private JLabel lblorder\_id,lblcustomer\_id,lbldrinks\_id,lblqty,lblprice\_perqty;

//private JTextField txtuser\_id,txtmid,txtday,txtorder\_id;

//private Choice user\_id,order\_id,mid;

private JTextField txtorder\_id,txtcustomer\_id,txtdrinks\_id,txtqty,txtprice\_perqty;

private Choice customer\_id,drinks\_id;

private List UserOrderIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,update,view;

public Order(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

this.update=update;

this.view=view;

lblorder\_id=new JLabel("Order ID");

lblcustomer\_id=new JLabel("Customer ID");

lbldrinks\_id=new JLabel("Drinks ID");

lblqty=new JLabel("quantity");

lblprice\_perqty=new JLabel("priceperquantity");

//order\_id=new Choice();

customer\_id=new Choice();

drinks\_id=new Choice();

txtorder\_id=new JTextField(15);

txtcustomer\_id=new JTextField(15);

txtdrinks\_id=new JTextField(15);

txtqty=new JTextField(8);

txtprice\_perqty=new JTextField(8);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loadorders() {

try {

UserOrderIDList.removeAll();

rs=statement.executeQuery("select \* from torder");

while(rs.next()) {

UserOrderIDList.add(rs.getString("order\_id")+"->"+ rs.getString("customer\_id")+"->"+rs.getString("drinks\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadcustomers() {

try {

customer\_id.removeAll();

rs=statement.executeQuery("select \* from customer");

while(rs.next()) {

customer\_id.add(rs.getString("customer\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loaddrinks() {

try {

drinks\_id.removeAll();

rs=statement.executeQuery("select \* from drinks");

while(rs.next()) {

drinks\_id.add(rs.getString("drinks\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

loadcustomers();

loaddrinks();

txtorder\_id.setText(null);

//txtcustomer\_id.setText(null);

//txtdrinks\_id.setText(null);

txtqty.setText(null);

txtprice\_perqty.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

// loadorders();

// loadcustomers();

// loaddrinks();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lblcustomer\_id);

p1.add(customer\_id);

p1.add(lbldrinks\_id);

p1.add(drinks\_id);

p1.add(lblqty);

p1.add(txtqty);

p1.add(lblprice\_perqty);

p1.add(txtprice\_perqty);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loadorders();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO TORDER VALUES("+txtorder\_id.getText()+","+customer\_id.getSelectedItem()+","+drinks\_id.getSelectedItem()+","+txtqty.getText()+","+txtprice\_perqty.getText()+")";

//System.out.println(query);

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

loadorders();

//loadorders();

loadcustomers();

loaddrinks();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txtorder\_id.setText(null);

txtcustomer\_id.setText(null);

txtdrinks\_id.setText(null);

txtqty.setText(null);

txtprice\_perqty.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lbldrinks\_id);

p1.add(txtdrinks\_id);

p1.add(lblqty);

p1.add(txtqty);

p1.add(lblprice\_perqty);

p1.add(txtprice\_perqty);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,400,300);

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loadorders();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

UserOrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from torder");

StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

String q=st.nextToken();

String r=st.nextToken();

while (rs.next())

{

if (rs.getString("order\_id").equals(p) && rs.getString("customer\_id").equals(q) && rs.getString("drinks\_id").equals(r))

break;

}

if (!rs.isAfterLast())

{

txtorder\_id.setText(rs.getString("order\_id"));

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtdrinks\_id.setText(rs.getString("drinks\_id"));

txtqty.setText(rs.getString("qty"));

txtprice\_perqty.setText(rs.getString("price\_perqty"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

//StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String query="DELETE FROM TORDER WHERE order\_id="+txtorder\_id.getText()+"";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loadorders();

// loadorders();

loadcustomers();

loaddrinks();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("modify");

txtorder\_id.setText(null);

txtcustomer\_id.setText(null);

txtdrinks\_id.setText(null);

txtqty.setText(null);

txtprice\_perqty.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lbldrinks\_id);

p1.add(txtdrinks\_id);

p1.add(lblqty);

p1.add(txtqty);

p1.add(lblprice\_perqty);

p1.add(txtprice\_perqty);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loadorders();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

UserOrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from torder");

StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

String q=st.nextToken();

String r=st.nextToken();

while (rs.next())

{

if (rs.getString("order\_id").equals(p) && rs.getString("customer\_id").equals(q) && rs.getString("drinks\_id").equals(r))

break;

}

if (!rs.isAfterLast())

{

txtorder\_id.setText(rs.getString("order\_id"));

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtdrinks\_id.setText(rs.getString("drinks\_id"));

txtqty.setText(rs.getString("qty"));

txtprice\_perqty.setText(rs.getString("price\_perqty"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loadorders();

String qty=JOptionPane.showInputDialog(p,"enter the qty");

txtqty.setText(qty);

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

//StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

//String query="update torder set order="+txtorder.getText()+",customer\_id="+txtcustomer\_id.getText()+",day='"+txtday.getText()+"' WHERE ORDER\_ID="+st.nextToken()+"and user\_id="+st.nextToken();

String query="update torder set qty='"+qty+"'where order\_id="+txtorder\_id.getText();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadorders();

// loadorders();

loadcustomers();

loaddrinks();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("ORDER view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2);

p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("order Details ");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("Order ID");

model.addColumn("Customer ID");

model.addColumn("Drinks ID");

model.addColumn("quantity");

model.addColumn("priceperquantity");

try {

rs=statement.executeQuery("select \* from torder");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("order\_id"), rs.getString("customer\_id"),rs.getString("drinks\_id"),rs.getString("qty"),rs.getString("price\_perqty")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 300, 300);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(800, 400);

f.setVisible(true);

}

});

}

});

}

}

Payments:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Choice;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class pay{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

private JLabel lblpayment\_id,lblorder\_id,lblcustomer\_id,lblttlquantity,lblttlprice;

private JTextField txtpayment\_id,txtorder\_id,txtcustomer\_id,txtttlquantity,txtttlprice;

private Choice order\_id,customer\_id;

private List UserOrderIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,update,view;

public pay(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

this.update=update;

this.view=view;

lblpayment\_id=new JLabel("payment\_id");

lblorder\_id=new JLabel("order\_id");

lblcustomer\_id=new JLabel("customer\_id");

lblttlquantity=new JLabel("ttlquantity");

lblttlprice=new JLabel("ttlprice");

//payment\_id=new Choice();

order\_id=new Choice();

customer\_id=new Choice();

txtpayment\_id=new JTextField(15);

txtorder\_id=new JTextField(15);

txtcustomer\_id=new JTextField(15);

txtttlquantity=new JTextField(8);

txtttlprice=new JTextField(8);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loadpayments() {

try {

UserOrderIDList.removeAll();

rs=statement.executeQuery("select \* from payments");

while(rs.next()) {

UserOrderIDList.add(rs.getString("payment\_id")+"->"+rs.getString("order\_id")+"->"+rs.getString("customer\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadorders() {

try {

order\_id.removeAll();

rs=statement.executeQuery("select \* from torder");

while(rs.next()) {

order\_id.add(rs.getString("order\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadcustomers() {

try {

customer\_id.removeAll();

rs=statement.executeQuery("select \* from customer");

while(rs.next()) {

customer\_id.add(rs.getString("customer\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

txtpayment\_id.setText(null);

// txtorder\_id.setText(null);

//txtcustomer\_id.setText(null);

txtttlquantity.setText(null);

txtttlprice.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

loadpayments();

loadorders();

loadcustomers();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lblpayment\_id);

p1.add(txtpayment\_id);

p1.add(lblorder\_id);

p1.add(order\_id);

p1.add(lblcustomer\_id);

p1.add(customer\_id);

p1.add(lblttlquantity);

p1.add(txtttlquantity);

p1.add(lblttlprice);

p1.add(txtttlprice);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loadpayments();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO payments VALUES("+txtpayment\_id.getText()+","+order\_id.getSelectedItem()+","+customer\_id.getSelectedItem()+","+txtttlquantity.getText()+","+txtttlprice.getText()+")";

//System.out.println(query);

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

loadpayments();

//loadpayment();

loadorders();

loadcustomers();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txtpayment\_id.setText(null);

txtorder\_id.setText(null);

txtcustomer\_id.setText(null);

txtttlquantity.setText(null);

txtttlprice.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lblpayment\_id);

p1.add(txtpayment\_id);

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblttlquantity);

p1.add(txtttlquantity);

p1.add(lblttlprice);

p1.add(txtttlprice);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,400,300);

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loadpayments();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

UserOrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from payments");

StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

String q=st.nextToken();

String r=st.nextToken();

while (rs.next())

{

if (rs.getString("payment\_id").equals(p) && rs.getString("order\_id").equals(q) && rs.getString("customer\_id").equals(r))

break;

}

if (!rs.isAfterLast())

{

txtpayment\_id.setText(rs.getString("payment\_id"));

txtorder\_id.setText(rs.getString("order\_id"));

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtttlquantity.setText(rs.getString("ttlquantity"));

txtttlprice.setText(rs.getString("ttlprice"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

//StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String query="DELETE FROM payments WHERE payment\_id="+txtpayment\_id.getText()+"";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");

loadpayments();

// loadpayment();

loadorders();

loadcustomers();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("modify");

txtpayment\_id.setText(null);

txtorder\_id.setText(null);

txtcustomer\_id.setText(null);

txtttlquantity.setText(null);

txtttlprice.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lblpayment\_id);

p1.add(txtpayment\_id);

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblttlquantity);

p1.add(txtttlquantity);

p1.add(lblttlprice);

p1.add(txtttlprice);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loadpayments();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

UserOrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from payments");

StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

String q=st.nextToken();

String r=st.nextToken();

while (rs.next())

{

if (rs.getString("payment\_id").equals(p) && rs.getString("order\_id").equals(q) && rs.getString("customer\_id").equals(r))

break;

}

if (!rs.isAfterLast())

{

txtpayment\_id.setText(rs.getString("payment\_id"));

txtorder\_id.setText(rs.getString("order\_id"));

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtttlquantity.setText(rs.getString("ttlquantity"));

txtttlprice.setText(rs.getString("ttlprice"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loadpayments();

String ttlquantity=JOptionPane.showInputDialog(p,"enter the qty");

txtttlquantity.setText(ttlquantity);

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

//StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String query="update payments set ttlquantity='"+ttlquantity+"'where payment\_id="+txtpayment\_id.getText();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadpayments();

//loadpayment();

loadorders();

loadcustomers();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("payments view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2);

p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("payments Details ");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("payment\_id");

model.addColumn("order\_id");

model.addColumn("customer\_id");

model.addColumn("ttlquantity");

model.addColumn("ttlprice");

try {

rs=statement.executeQuery("select \* from payments");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("payment\_id"), rs.getString("order\_id"),rs.getString("customer\_id"),rs.getString("ttlquantity"),rs.getString("ttlprice")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 180, 150);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(500, 400);

f.setVisible(true);

}

});

}

});

}

}

Transaction:

package DrinkB;

import java.awt.BorderLayout;

import java.awt.Choice;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.\*;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class trans{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JButton insertButton,deleteButton,updateButton,viewButton;

private JPanel p1,p2,p3,p;

private JLabel lbltransaction\_id,lblcustomer\_id,lblemployee\_id,lblorder\_id,lbldrinks\_id,lblpayment\_id,lblsupplier\_id;

private JTextField txttransaction\_id,txtpayment\_id,txtemployee\_id,txtcustomer\_id,txtdrinks\_id,txtorder\_id,txtsupplier\_id;

private Choice employee\_id,drinks\_id,payment\_id,order\_id,customer\_id,supplier\_id;

private List UserOrderIDList;

Connection con;ResultSet rs;

Statement statement;

private JFrame frame;

private JMenuItem insert,delete,view;

public trans(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem view)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

this.frame=frame;

this.insert=insert;

this.delete=delete;

//this.update=update;

this.view=view;

lbltransaction\_id=new JLabel("transaction\_id");

lblcustomer\_id=new JLabel("customer\_id");

lblemployee\_id=new JLabel("employee\_id");

lblorder\_id=new JLabel("order\_id");

lbldrinks\_id=new JLabel("drinks\_id");

lblpayment\_id=new JLabel("payment\_id");

lblsupplier\_id=new JLabel("supplier\_id");

//payment\_id=new Choice();

customer\_id=new Choice();

employee\_id=new Choice();

order\_id=new Choice();

drinks\_id=new Choice();

payment\_id=new Choice();

supplier\_id=new Choice();

txttransaction\_id=new JTextField(15);

txtcustomer\_id=new JTextField(15);

txtemployee\_id=new JTextField(15);

txtorder\_id=new JTextField(8);

txtdrinks\_id=new JTextField(8);

txtpayment\_id=new JTextField(8);

txtsupplier\_id=new JTextField(8);

this.p=p;

}

public void connectToDB()

{

try {

Connection con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","it19737110","vasavi");

statement=con.createStatement();

statement.executeUpdate("commit");

}

catch (SQLException connectException)

{

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());

System.exit(1);

}

}

private void displaySQLErrors(SQLException e)

{

JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " + e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loaduserOrderIDs() {

try {

UserOrderIDList.removeAll();

rs=statement.executeQuery("select \* from transaction");

while(rs.next()) {

UserOrderIDList.add(rs.getString("customer\_id")+"->"+rs.getString("employee\_id")+"->"+rs.getString("order\_id")+"->"+rs.getString("drinks\_id")+"->"+rs.getString("payment\_id")+"->"+rs.getString("supplier\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadcustomers() {

try {

customer\_id.removeAll();

rs=statement.executeQuery("select \* from customer");

while(rs.next()) {

customer\_id.add(rs.getString("customer\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loademployees() {

try {

employee\_id.removeAll();

rs=statement.executeQuery("select \* from employee");

while(rs.next()) {

employee\_id.add(rs.getString("employee\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadorders() {

try {

order\_id.removeAll();

rs=statement.executeQuery("select \* from torder");

while(rs.next()) {

order\_id.add(rs.getString("order\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loaddrinks() {

try {

drinks\_id.removeAll();

rs=statement.executeQuery("select \* from drinks");

while(rs.next()) {

drinks\_id.add(rs.getString("drinks\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadpayments() {

try {

payment\_id.removeAll();

rs=statement.executeQuery("select \* from payments");

while(rs.next()) {

payment\_id.add(rs.getString("payment\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadsuppliers() {

try {

supplier\_id.removeAll();

rs=statement.executeQuery("select \* from supplier");

while(rs.next()) {

supplier\_id.add(rs.getString("supplier\_id"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI() {

insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

insertButton=new JButton("submit");

loadcustomers();

loademployees();

loadorders();

loaddrinks();

loadpayments();

loadsuppliers();

txttransaction\_id.setText(null);

// txtcustomer\_id.setText(null);

// txtemployee\_id.setText(null);

// txtorder\_id.setText(null);

// txtdrinks\_id.setText(null);

// txtpayment\_id.setText(null);

// txtsupplier\_id.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

//loadpayment();

p1=new JPanel();

p1.setLayout(new GridLayout(7,1));

p1.add(lbltransaction\_id);

p1.add(txttransaction\_id);

p1.add(lblcustomer\_id);

p1.add(customer\_id);

p1.add(lblemployee\_id);

p1.add(employee\_id);

p1.add(lblorder\_id);

p1.add(order\_id);

p1.add(lbldrinks\_id);

p1.add(drinks\_id);

p1.add(lblpayment\_id);

p1.add(payment\_id);

p1.add(lblsupplier\_id);

p1.add(supplier\_id);

p3=new JPanel(new FlowLayout());

p3.add(insertButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loaduserOrderIDs();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

insertButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

String query="INSERT INTO transaction VALUES("+txttransaction\_id.getText()+","+customer\_id.getSelectedItem()+","+employee\_id.getSelectedItem()+","+order\_id.getSelectedItem()+","+drinks\_id.getSelectedItem()+","+payment\_id.getSelectedItem()+","+supplier\_id.getSelectedItem()+")";

//System.out.println(query);

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

loaduserOrderIDs();

loadcustomers();

loademployees();

loadorders();

loaddrinks();

loadpayments();

loadsuppliers();

}

catch(SQLException insertException){

displaySQLErrors(insertException);

}

}

});

}

});

delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

deleteButton=new JButton("delete");

txttransaction\_id.setText(null);

txtcustomer\_id.setText(null);

txtemployee\_id.setText(null);

txtorder\_id.setText(null);

txtdrinks\_id.setText(null);

txtpayment\_id.setText(null);

txtsupplier\_id.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(7,1));

p1.add(lbltransaction\_id);

p1.add(txttransaction\_id);

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblemployee\_id);

p1.add(txtemployee\_id);

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lbldrinks\_id);

p1.add(txtdrinks\_id);

p1.add(lblpayment\_id);

p1.add(txtpayment\_id);

p1.add(lblsupplier\_id);

p1.add(txtsupplier\_id);

p3=new JPanel(new FlowLayout());

p3.add(deleteButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,400,300);

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loaduserOrderIDs();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

UserOrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from transaction");

StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

String q=st.nextToken();

String r=st.nextToken();

String s=st.nextToken();

String t=st.nextToken();

String u=st.nextToken();

while (rs.next())

{

if (rs.getString("customer\_id").equals(p) && rs.getString("employee\_id").equals(q) && rs.getString("order\_id").equals(r) && rs.getString("drinks\_id").equals(s) && rs.getString("payment\_id").equals(t) && rs.getString("supplier\_id").equals(u))

break;

}

if (!rs.isAfterLast())

{

txttransaction\_id.setText(rs.getString("transaction\_id"));

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtemployee\_id.setText(rs.getString("employee\_id"));

txtorder\_id.setText(rs.getString("order\_id"));

txtdrinks\_id.setText(rs.getString("drinks\_id"));

txtpayment\_id.setText(rs.getString("payment\_id"));

txtsupplier\_id.setText(rs.getString("supplier\_id"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");

if(a==JOptionPane.YES\_OPTION){

//StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String query="DELETE FROM transaction WHERE transaction\_id="+txttransaction\_id.getText()+"";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loaduserOrderIDs();

// loadpayment();

loadcustomers();

loademployees();

loadorders();

loaddrinks();

loadpayments();

loadsuppliers();

}

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});

/\*update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

JButton updateButton = new JButton("update");

txttransaction\_id.setText(null);

txtcustomer\_id.setText(null);

txtemployee\_id.setText(null);

txtorder\_id.setText(null);

txtdrinks\_id.setText(null);

txtpayment\_id.setText(null);

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(5,1));

p1.add(lbltransaction\_id);

p1.add(txttransaction\_id);

p1.add(lblcustomer\_id);

p1.add(txtcustomer\_id);

p1.add(lblemployee\_id);

p1.add(txtemployee\_id);

p1.add(lblorder\_id);

p1.add(txtorder\_id);

p1.add(lbldrinks\_id);

p1.add(txtdrinks\_id);

p1.add(lblpayment\_id);

p1.add(txtpayment\_id);

p3=new JPanel(new FlowLayout());

p3.add(updateButton);

//p1.add(txtf1);

p3.setBackground(Color.yellow);

p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);

p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

UserOrderIDList=new List(10);

loaduserOrderIDs();

p2.add(UserOrderIDList);

p2.setBounds(450,150,350,180); p2.setBackground(Color.cyan) ;

p. add(p1);p.add(p3);

p. add(p2);

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

UserOrderIDList.addItemListener(new ItemListener()

{

public void itemStateChanged(ItemEvent e)

{

try

{

rs=statement.executeQuery("select \* from transaction");

StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String p=st.nextToken();

String q=st.nextToken();

String r=st.nextToken();

String s=st.nextToken();

String t=st.nextToken();

while (rs.next())

{

if (rs.getString("customer\_id").equals(p) && rs.getString("employee\_id").equals(q) && rs.getString("order\_id").equals(r) && rs.getString("drinks\_id").equals(s) && rs.getString("payment\_id").equals(t))

break;

}

if (!rs.isAfterLast())

{

txttransaction\_id.setText(rs.getString("transaction\_id"));

txtcustomer\_id.setText(rs.getString("customer\_id"));

txtemployee\_id.setText(rs.getString("employee\_id"));

txtorder\_id.setText(rs.getString("order\_id"));

txtdrinks\_id.setText(rs.getString("drinks\_id"));

txtpayment\_id.setText(rs.getString("payment\_id"));

}

}

catch (SQLException selectException)

{

displaySQLErrors(selectException);

}

}

});

p.setLayout(new BorderLayout());

frame.add(p);

frame.setSize(800,800);

frame.validate();

updateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

try {

loaduserOrderIDs();

String ttlquantity=JOptionPane.showInputDialog(p,"enter the qty");

txtttlquantity.setText(ttlquantity);

//int a=JOptionPane.showConfirmDialog(p,"Are you sure want to update:");

//if(a==JOptionPane.YES\_OPTION){

//StringTokenizer st=new StringTokenizer(UserOrderIDList.getSelectedItem(),"->");

String query="update payments set ttlquantity='"+ttlquantity+"'where payment\_id="+txtpayment\_id.getText();

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loaduserOrderIDs();

//loadpayment();

loadorders();

loadcustomer();

}

catch(SQLException deleteException){

displaySQLErrors(deleteException);

}

}

});

}

});\*/

view.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

p.removeAll();

frame.invalidate();

frame.validate();

frame.repaint();

Label view1=new Label("transaction view");

//view1.setAlignment(Label.CENTER);

Font myFont = new Font("Serif",Font.BOLD,50);

view1.setFont((myFont));

viewButton=new JButton("View");

p1=new JPanel();

p2=new JPanel();

p1.add(view1);

p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;

p.add(p1);p.add(p2);

p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);

frame.add(p);

frame.setSize(800,800);

frame.validate();

viewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JFrame f;

JTable j;

f = new JFrame();

f.setTitle("transaction Details ");

DefaultTableModel model = new DefaultTableModel();

j = new JTable(model);

model.addColumn("transaction\_id");

model.addColumn("customer\_id");

model.addColumn("employee\_id");

model.addColumn("order\_id");

model.addColumn("drinks\_id");

model.addColumn("payment\_id");

try {

rs=statement.executeQuery("select \* from transaction");

while(rs.next()) {

model.addRow(new Object[]{rs.getString("transaction\_id"), rs.getString("customer\_id"),rs.getString("employee\_id"),rs.getString("order\_id"),rs.getString("drinks\_id"),rs.getString("payment\_id")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);

j.setBounds(30, 40, 180, 150);

JScrollPane sp = new JScrollPane(j);

f.add(sp);

f.setSize(500, 400);

f.setVisible(true);

}

});

}

});

}

}

DBUI:

package DrinkB;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

public class DBUI extends JFrame{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JMenuBar mnu;

private JMenu mnuemployee;

private JMenu mnucustomer;

private JMenu mnusupplier;

private JMenu mnudrinks;

private JMenu mnutorder;

private JMenu mnupayments;

private JMenu mnutransaction;

private JMenuItem insert1,update1,delete1,view1;

private JMenuItem insert2,update2,delete2,view2;

private JMenuItem insert3,update3,delete3,view3;

private JMenuItem insert4,update4,delete4,view4;

private JMenuItem insert5,update5,delete5,view5;

private JMenuItem insert6,update6,delete6,view6;

private JMenuItem insert7,delete7,view7;

private JLabel labelName;

private static JPanel p0,p1;

void initialize() {

mnu=new JMenuBar();

mnuemployee= new JMenu("Employee");

mnucustomer= new JMenu("Supplier");

mnusupplier= new JMenu("Customer");

mnudrinks= new JMenu("Drink");

mnutorder= new JMenu("Orders");

mnupayments= new JMenu("Payments");

mnutransaction= new JMenu("Transactions");

labelName=new JLabel("DRINK BUCKS");

p1=new JPanel();

p0=new JPanel();

insert1=new JMenuItem("Insert");

update1=new JMenuItem("Update");

delete1=new JMenuItem("Delete");

view1=new JMenuItem("View");

insert2=new JMenuItem("Insert");

update2=new JMenuItem("Update");

delete2=new JMenuItem("Delete");

view2=new JMenuItem("View");

insert3=new JMenuItem("Insert");

update3=new JMenuItem("Update");

delete3=new JMenuItem("Delete");

view3=new JMenuItem("View");

insert4=new JMenuItem("Insert");

update4=new JMenuItem("Update");

delete4=new JMenuItem("Delete");

view4=new JMenuItem("View");

insert5=new JMenuItem("Insert");

update5=new JMenuItem("Update");

delete5=new JMenuItem("Delete");

view5=new JMenuItem("View");

insert6=new JMenuItem("Insert");

update6=new JMenuItem("Update");

delete6=new JMenuItem("Delete");

view6=new JMenuItem("View");

insert7=new JMenuItem("Insert");

// update7=new JMenuItem("Update");

delete7=new JMenuItem("Delete");

view7=new JMenuItem("View");

insert1=new JMenuItem("Insert");

update1=new JMenuItem("Update");

delete1=new JMenuItem("Delete");

view1=new JMenuItem("View");

}

void addComponentsToFrame() {

mnuemployee.add(insert1);

mnuemployee.add(delete1);

mnuemployee.add(update1);

mnuemployee.add(view1);

mnucustomer.add(insert2);

mnucustomer.add(delete2);

mnucustomer.add(update2);

mnucustomer.add(view2);

mnusupplier.add(insert3);

mnusupplier.add(delete3);

mnusupplier.add(update3);

mnusupplier.add(view3);

mnudrinks.add(insert4);

mnudrinks.add(delete4);

mnudrinks.add(update4);

mnudrinks.add(view4);

mnutorder.add(insert5);

mnutorder.add(delete5);

mnutorder.add(update5);

mnutorder.add(view5);

mnupayments.add(insert6);

mnupayments.add(delete6);

mnupayments.add(update6);

mnupayments.add(view6);

mnutransaction.add(insert7);

mnutransaction.add(delete7);

// mnutransaction.add(update7);

mnutransaction.add(view7);

mnu.add(mnuemployee);

mnu.add(mnucustomer);

mnu.add(mnusupplier);

mnu.add(mnudrinks);

mnu.add(mnutorder);

mnu.add(mnupayments);

mnu.add(mnutransaction);

setJMenuBar(mnu);

p1.add(labelName);p1.setAlignmentY(CENTER\_ALIGNMENT);

p1.setBounds(500,500,800,100);

p0.add(p1);

p0.setBackground(Color.CYAN);

add(p0);

}

void closeWindow(){

try {

int a=JOptionPane.showConfirmDialog(this,"Are you sure want to Quit DRINK BUCKS:");

if(a==JOptionPane.YES\_OPTION){

JOptionPane.showMessageDialog(this,

"Thank you!\nExiting DRINK BUCKS","Quit",

JOptionPane.WARNING\_MESSAGE);

System.exit(0);

}

else if (a== JOptionPane.NO\_OPTION) {

setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

else if (a== JOptionPane.CANCEL\_OPTION) {

setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

}

catch(Exception e) {

System.out.println(e);

}

}

void register() {

Emp log=new Emp(p0,DBUI.this,insert1,delete1,update1,view1);

log.buildGUI();

Cust user=new Cust(p0,DBUI.this,insert3,delete3,update3,view3);

user.buildGUI();

Supp user1=new Supp(p0,DBUI.this,insert2,delete2,update2,view2);

user1.buildGUI();

Drinks d=new Drinks(p0,DBUI.this,insert4,delete4,update4,view4);

d.buildGUI();

Order u=new Order(p0,DBUI.this,insert5,delete5,update5,view5);

u.buildGUI();

pay p=new pay(p0,DBUI.this,insert6,delete6,update6,view6);

p.buildGUI();

trans u\_orders=new trans(p0,DBUI.this,insert7,delete7,view7);

u\_orders.buildGUI();

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we)

{

closeWindow();

}

});

}

public DBUI() {

initialize();

addComponentsToFrame();

register();

// setBackground(Color.RED);

setTitle("DRINKSBUCKS");

setSize(800,800);

setVisible(true);

}

}

MAIN:

package DrinkB;

public class main {

public static void main(String[] args) {

// TODO Auto-generated method stub

new DBUI();

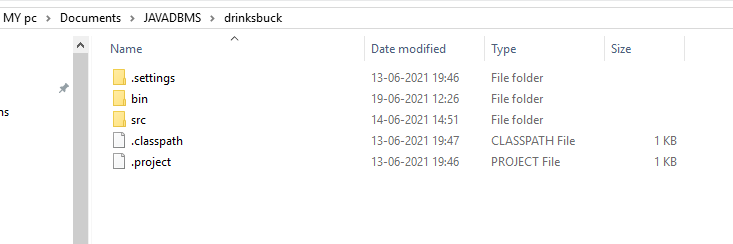
}

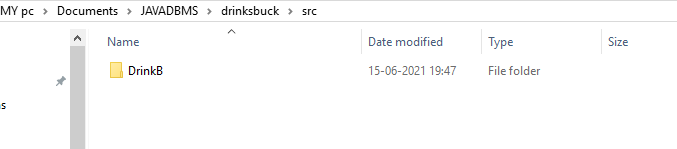
}

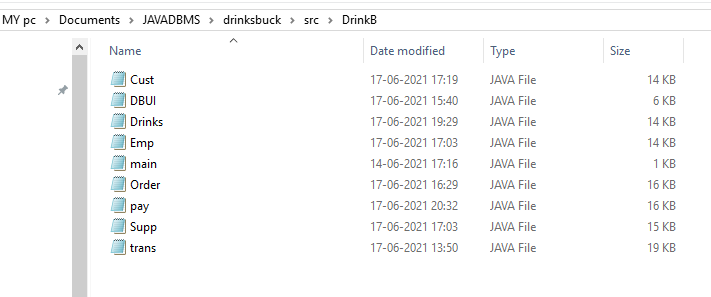
**GITHUB LINK**:

https://github.com/sreeya-c/DRINKS-BUCKS

FOLDER STRUCTURE:

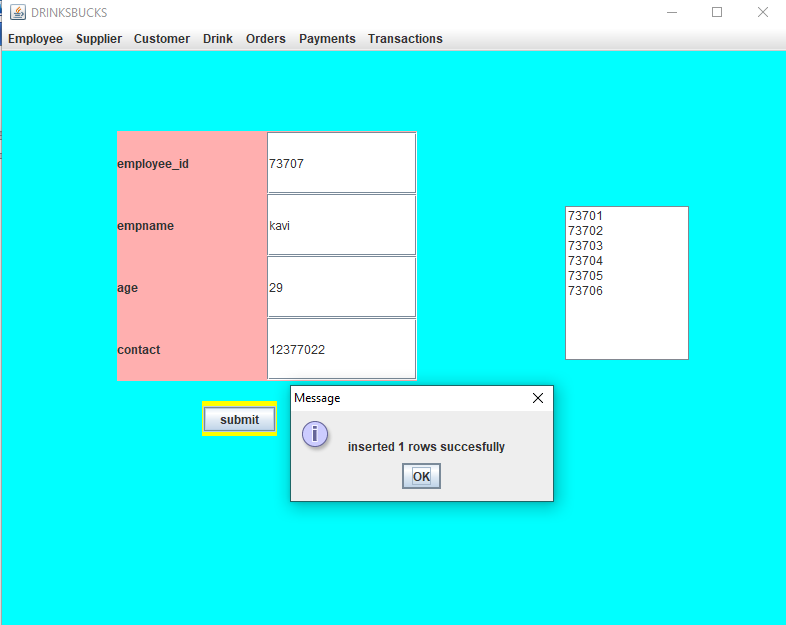


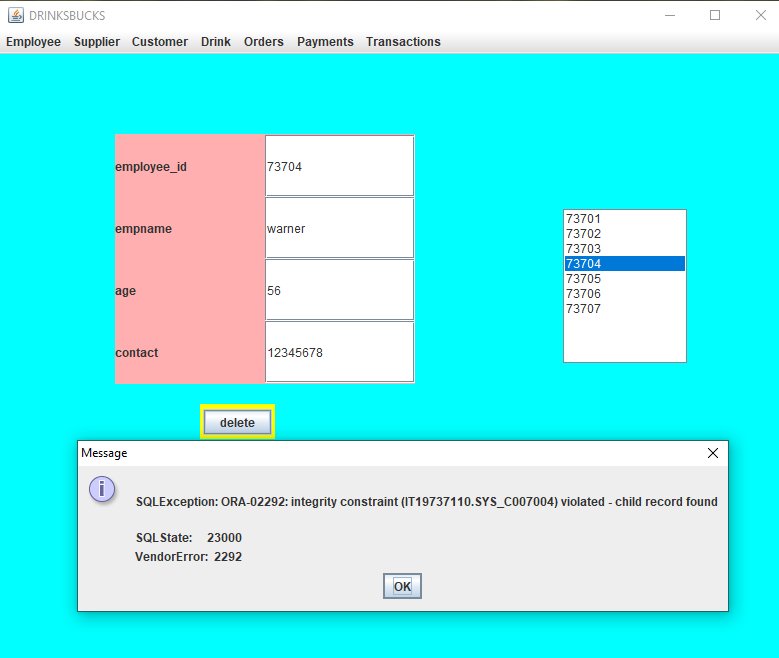


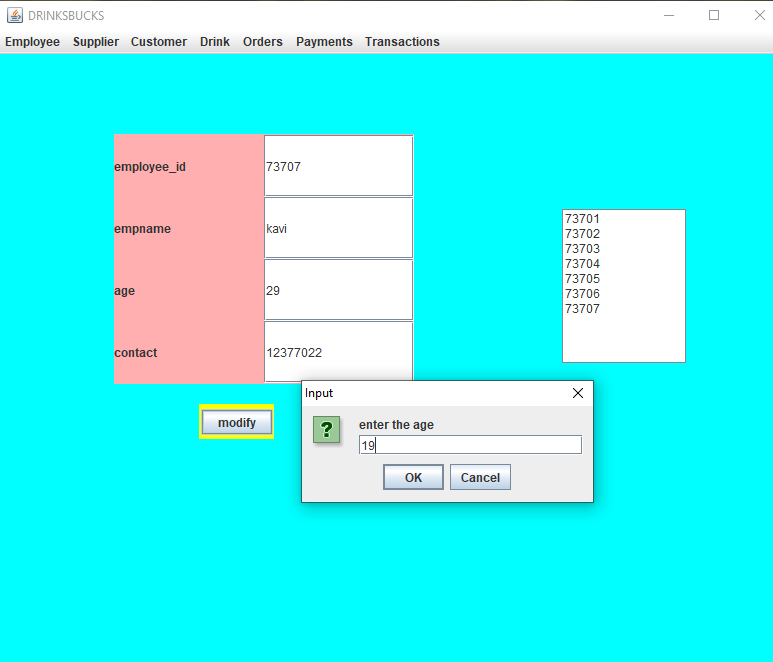


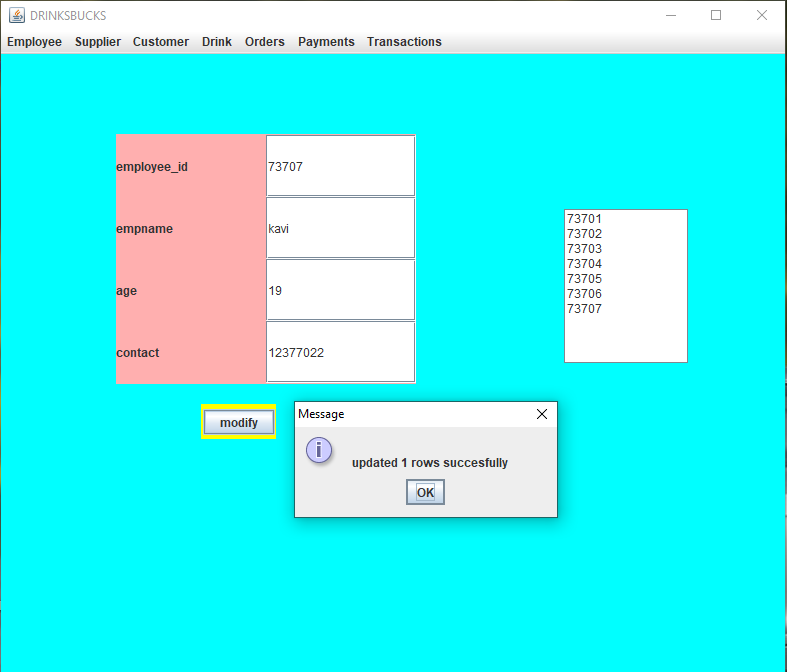
## TESTING

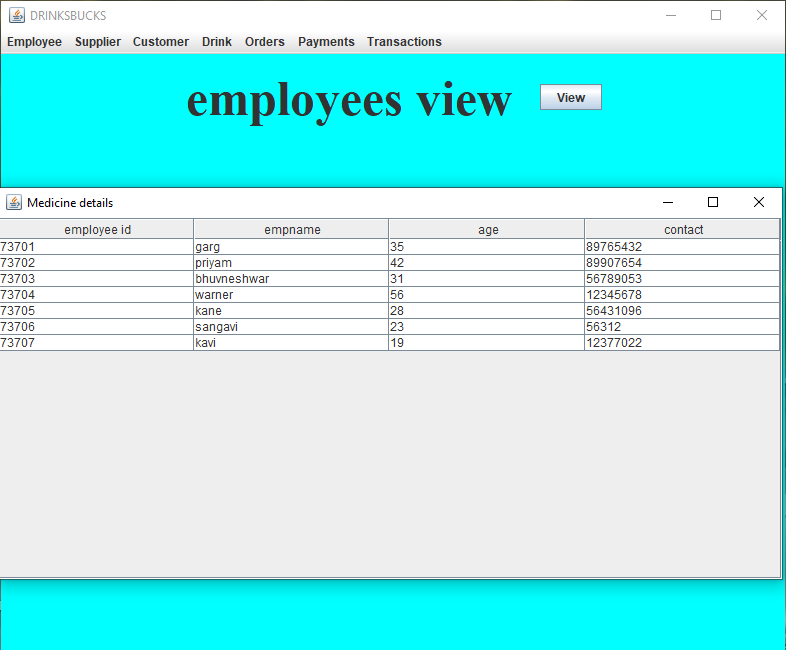
OUTPUT SCREENSHOTS:

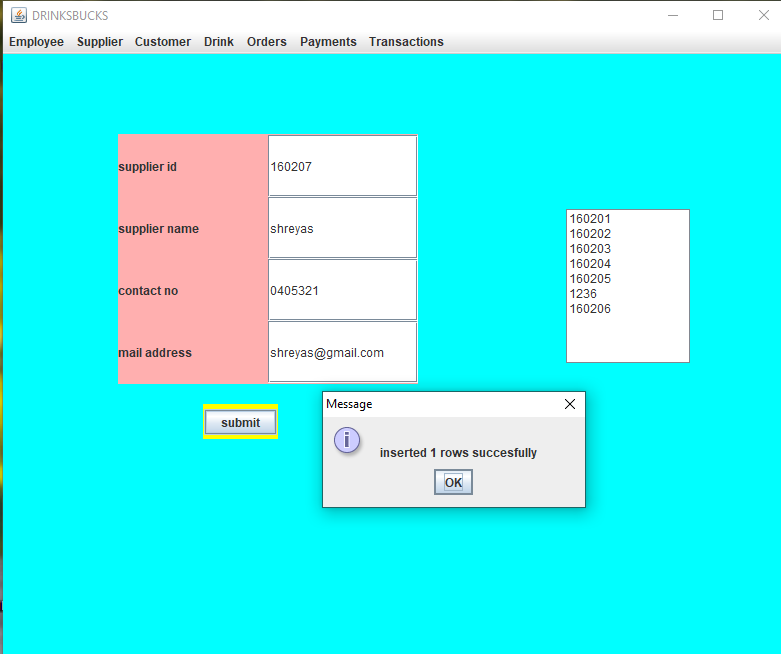


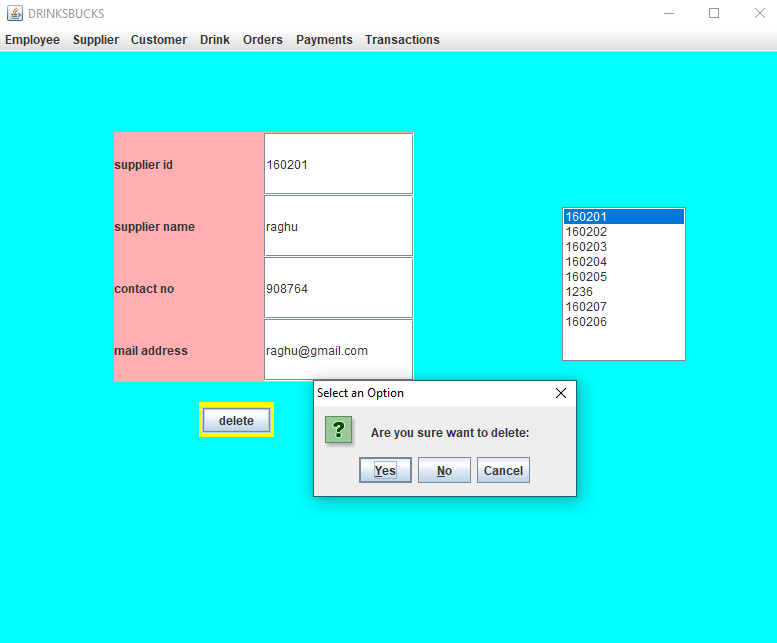


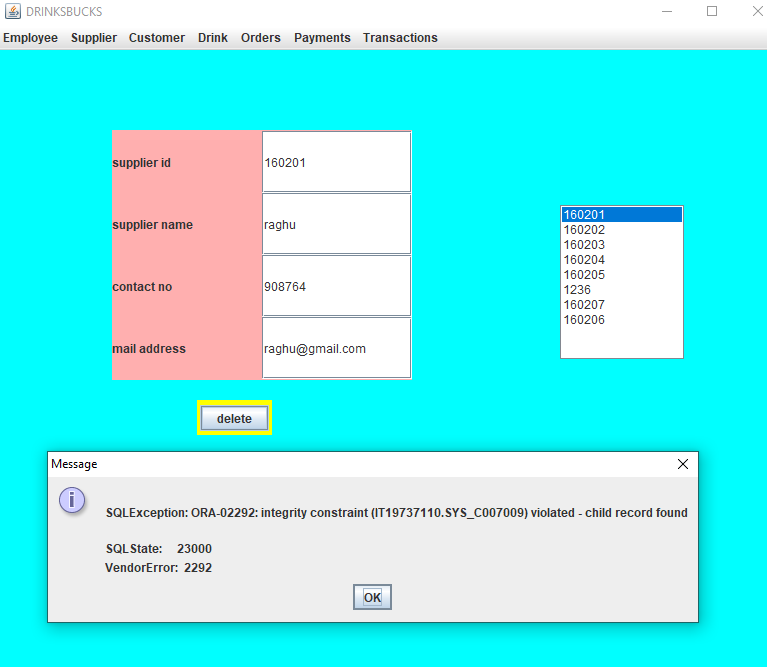


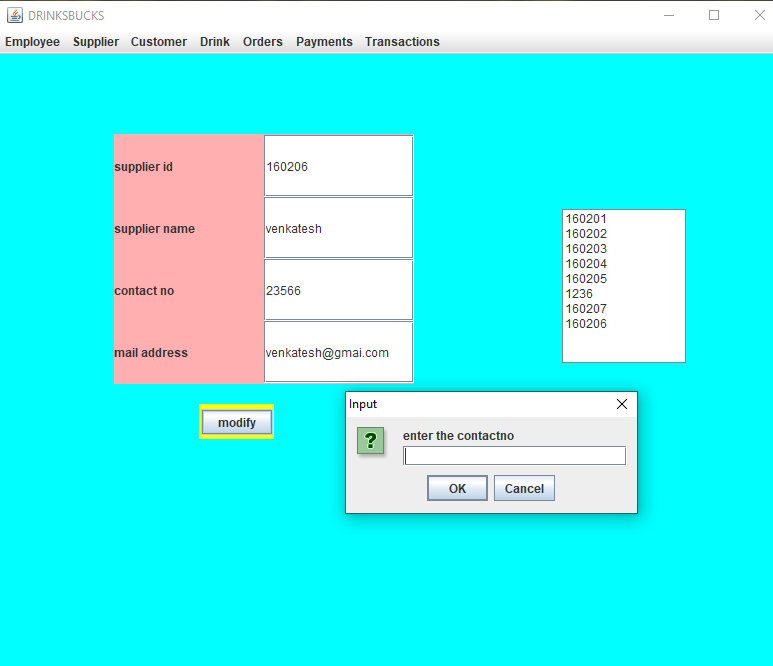


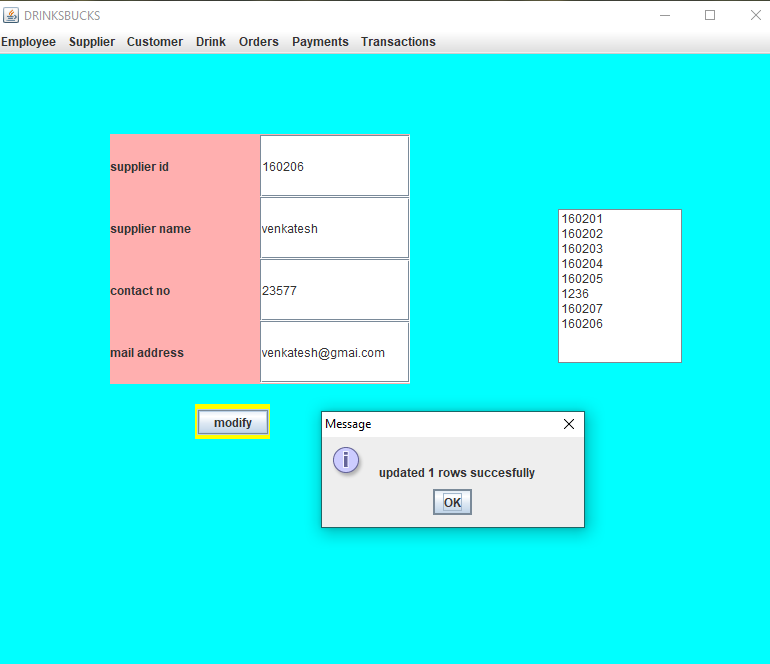


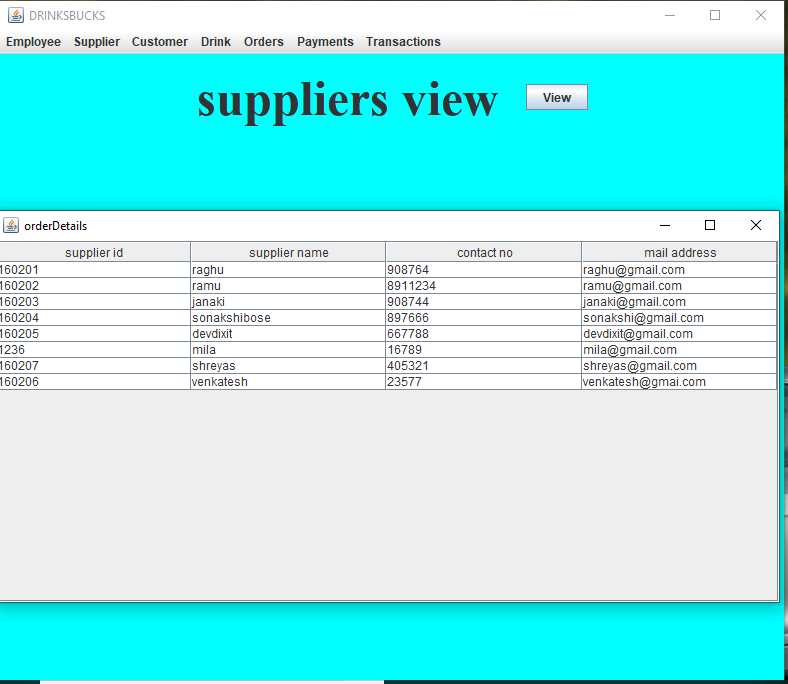


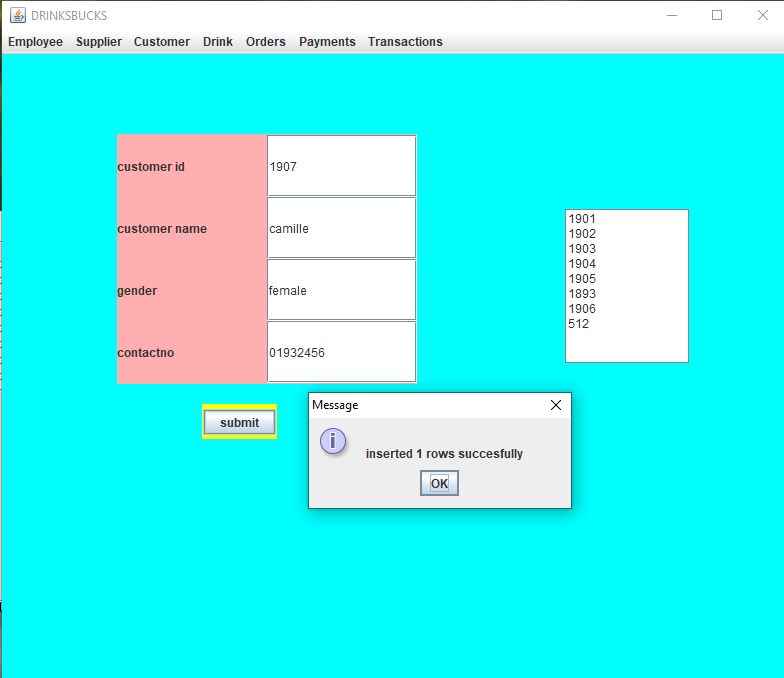


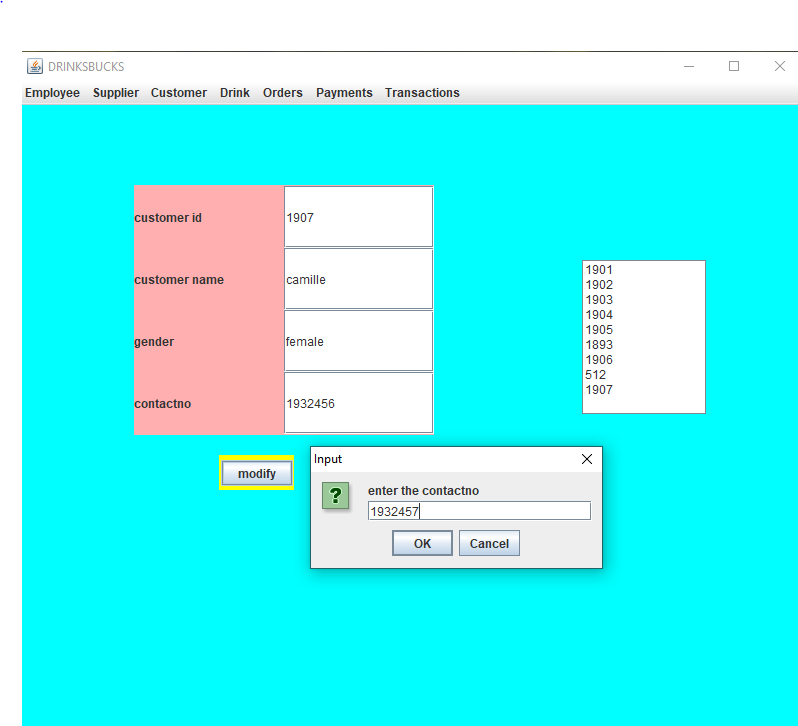


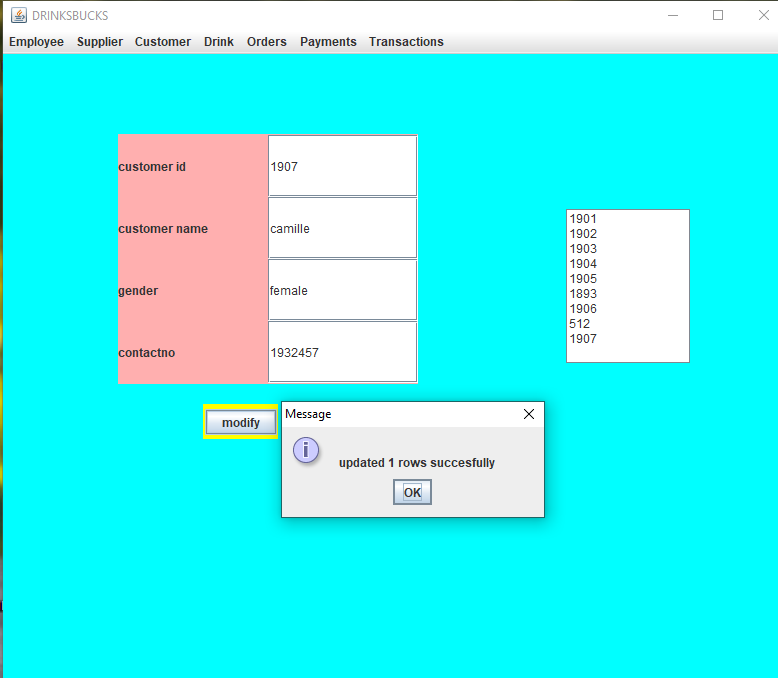


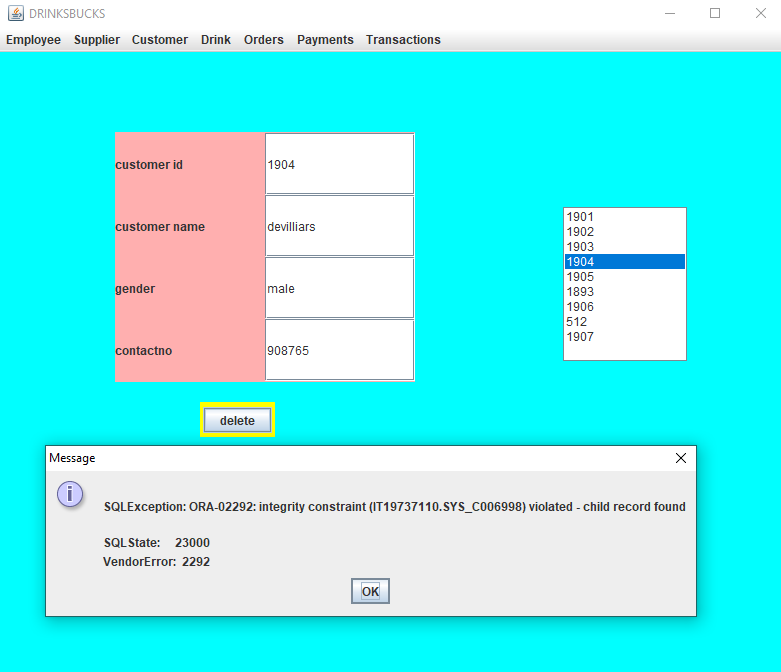


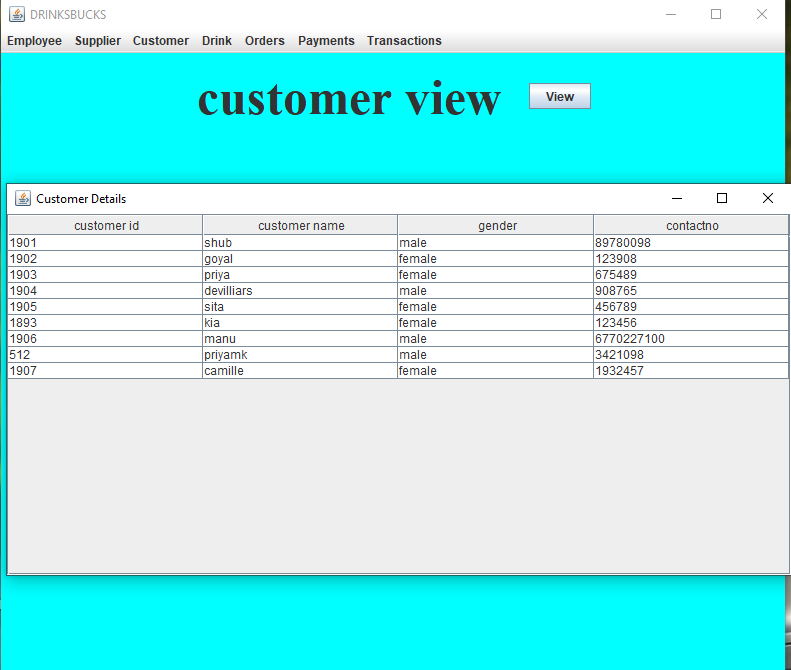


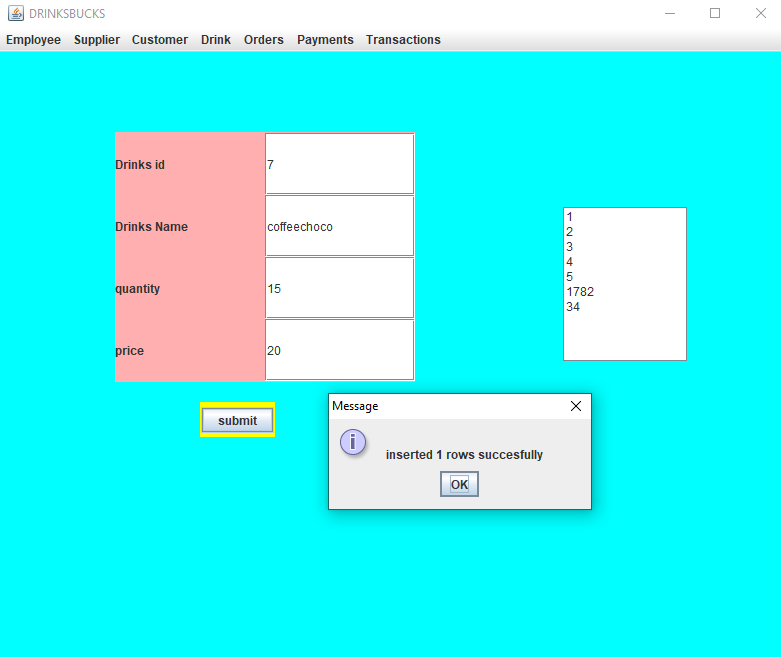


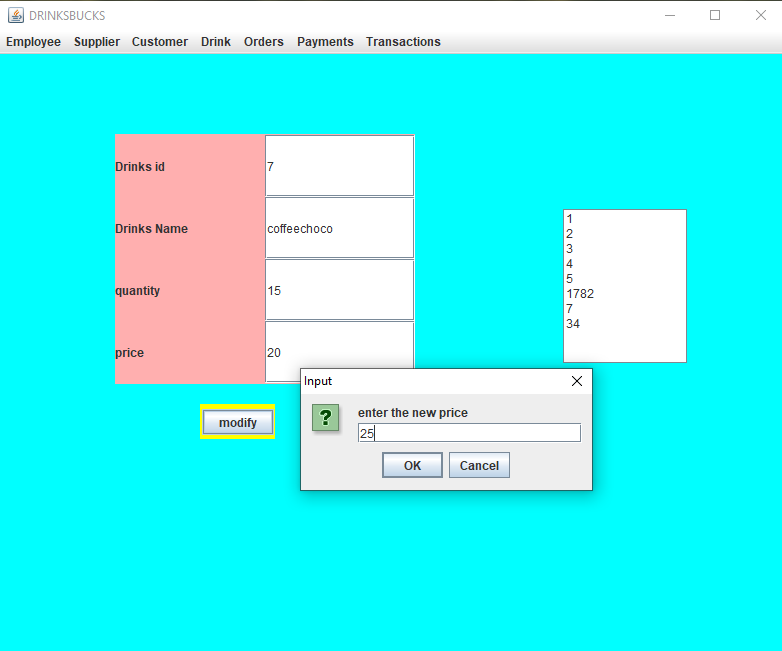


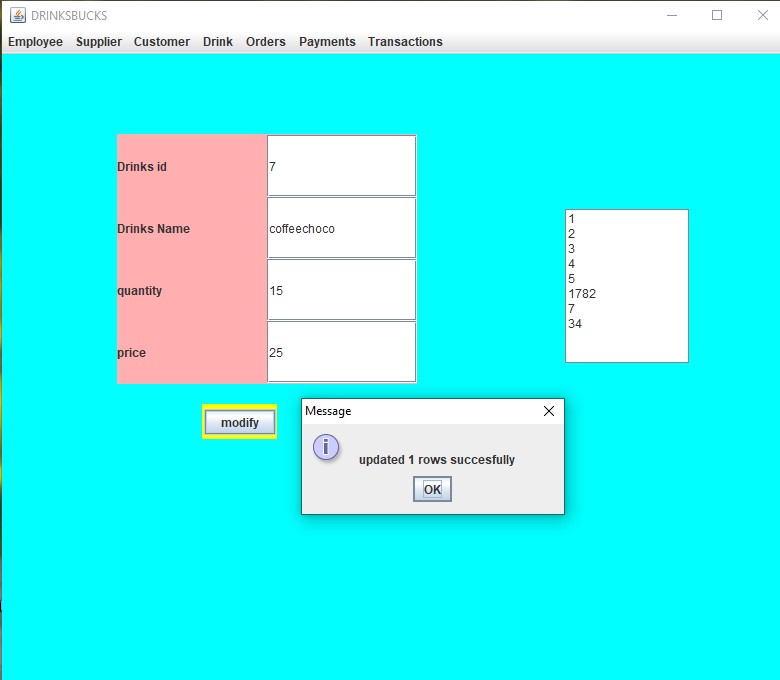


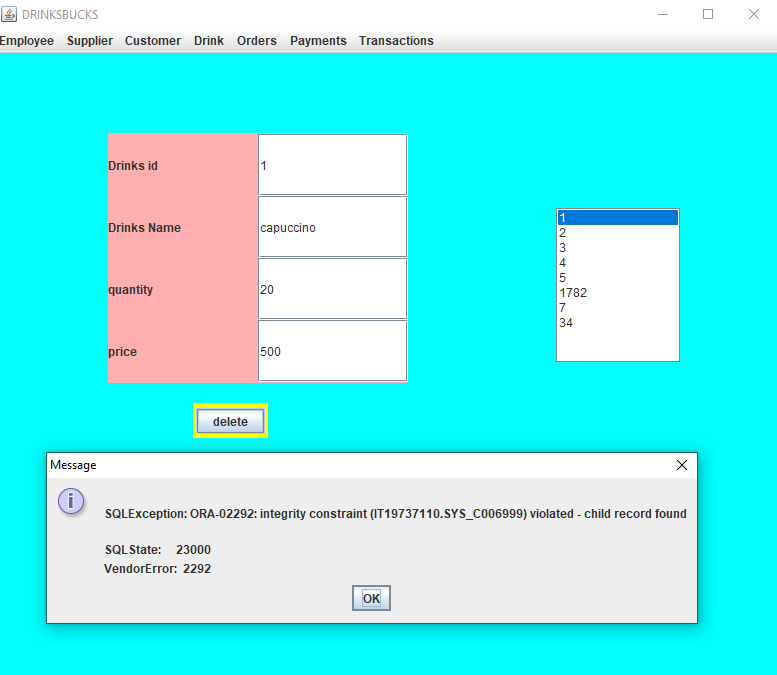


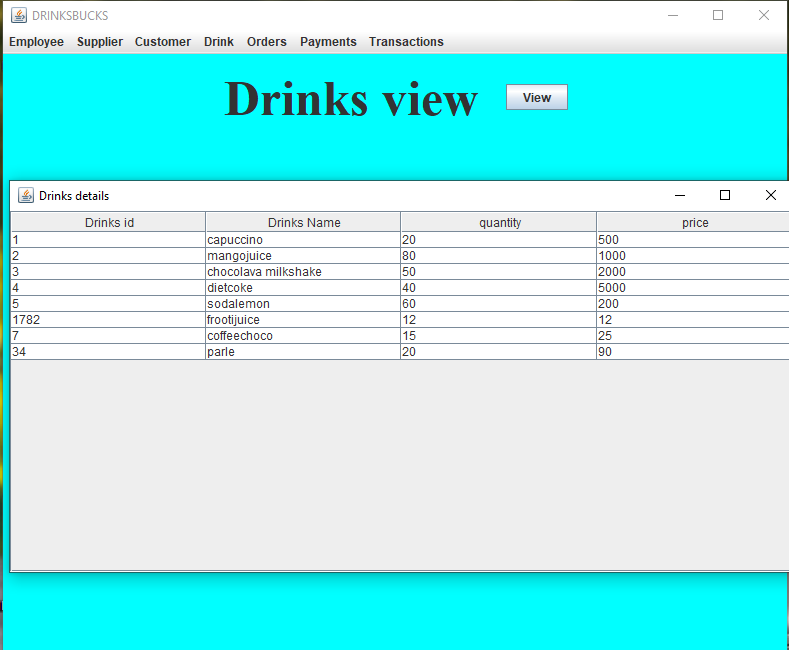


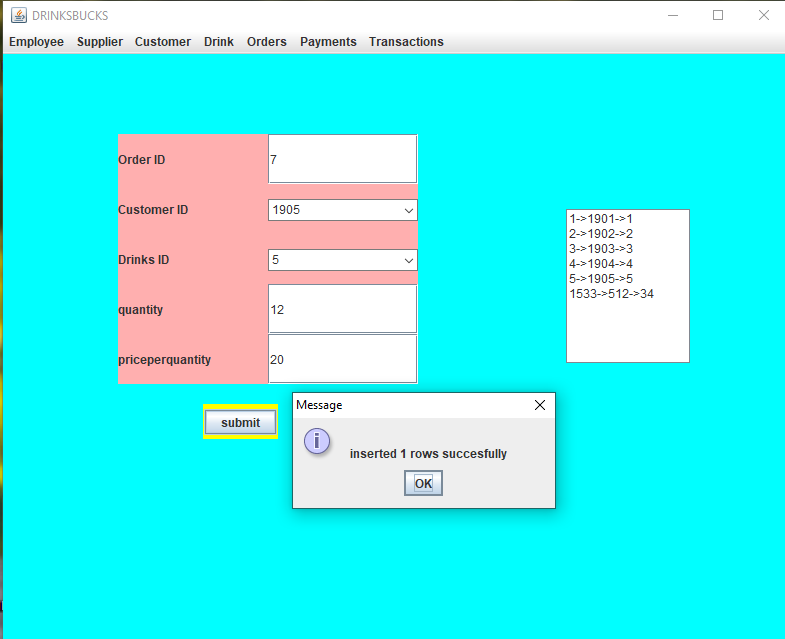


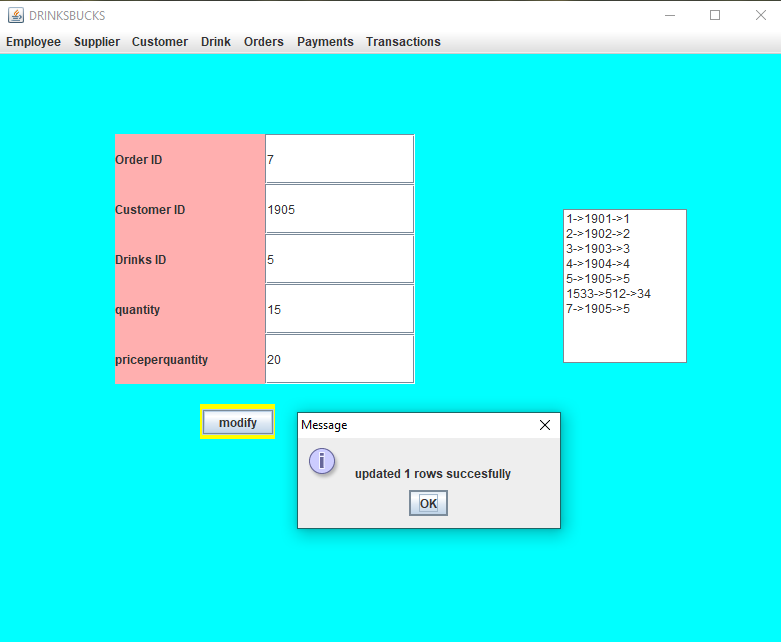


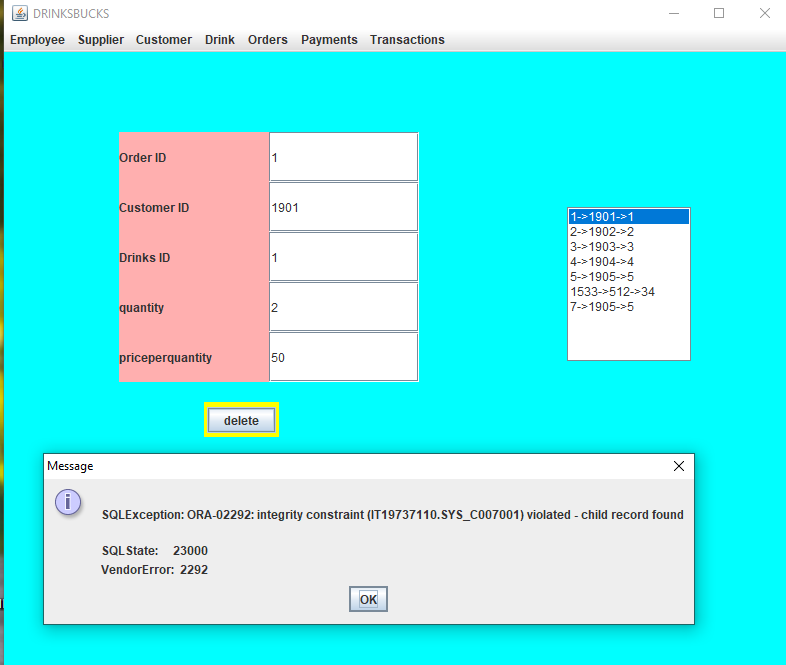


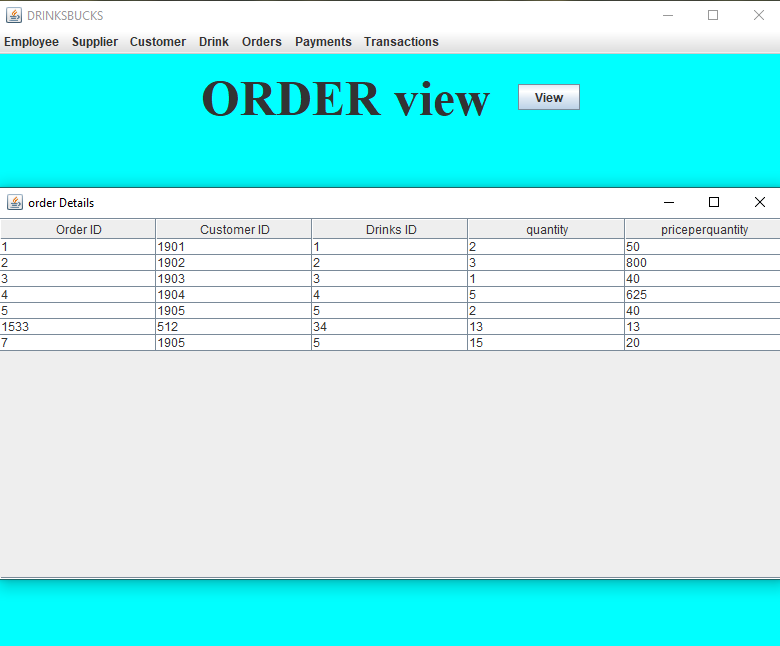


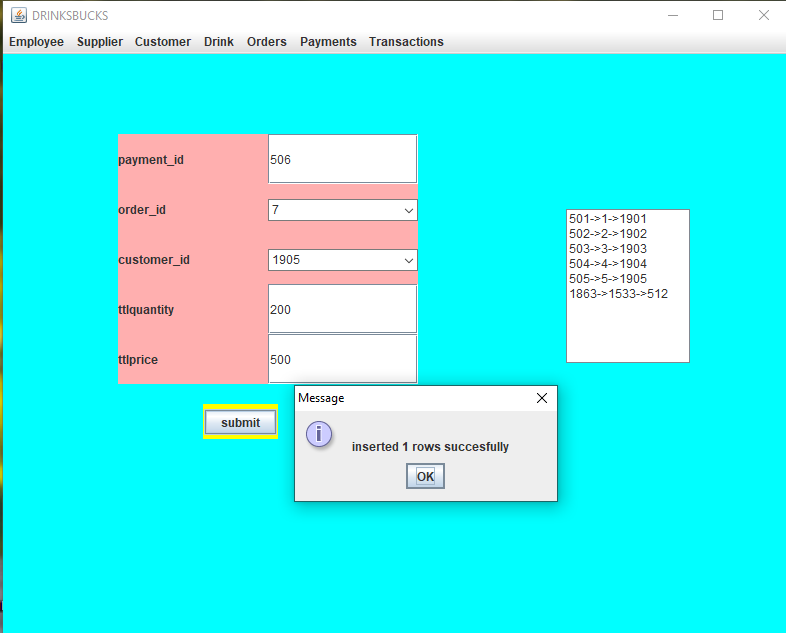


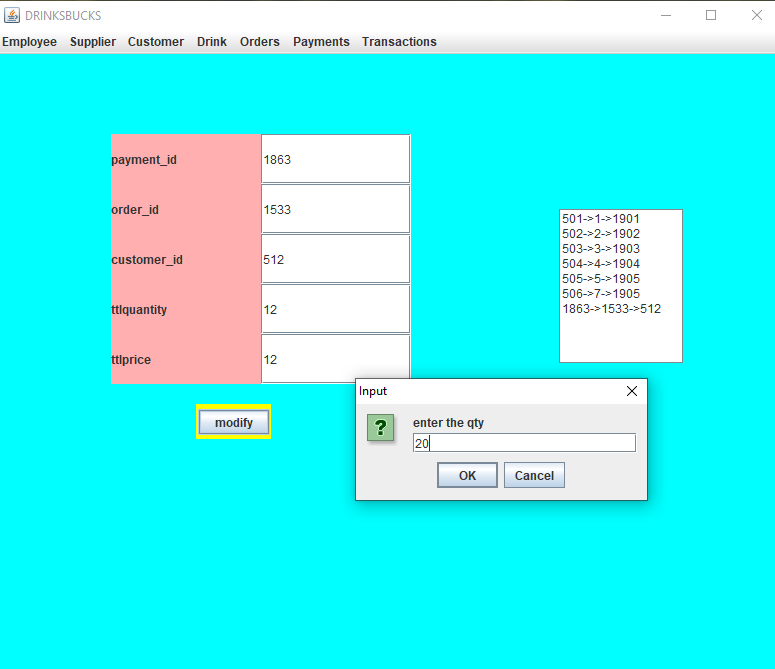


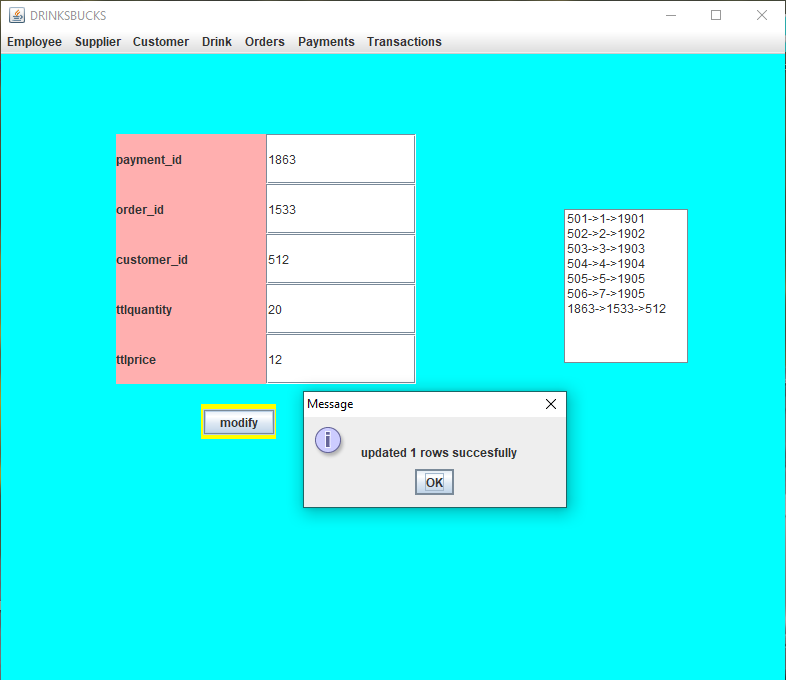


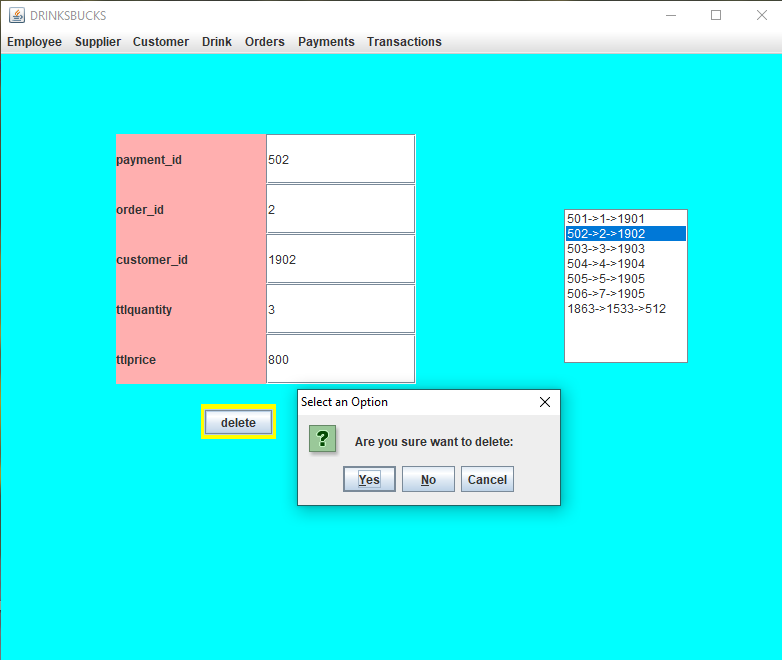


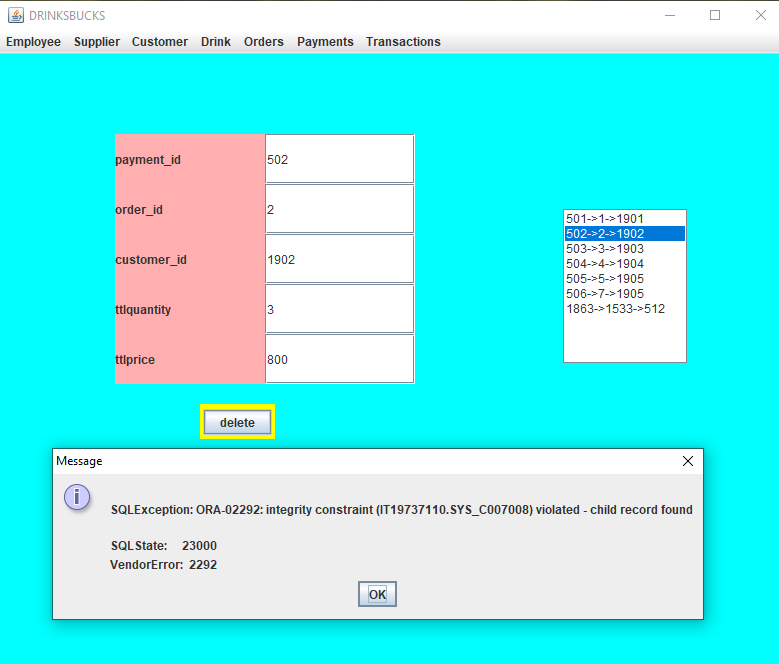


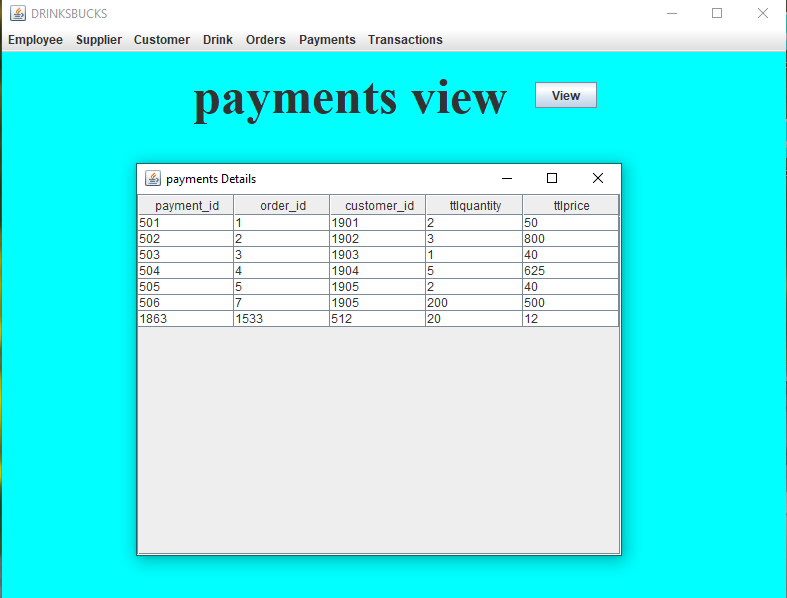


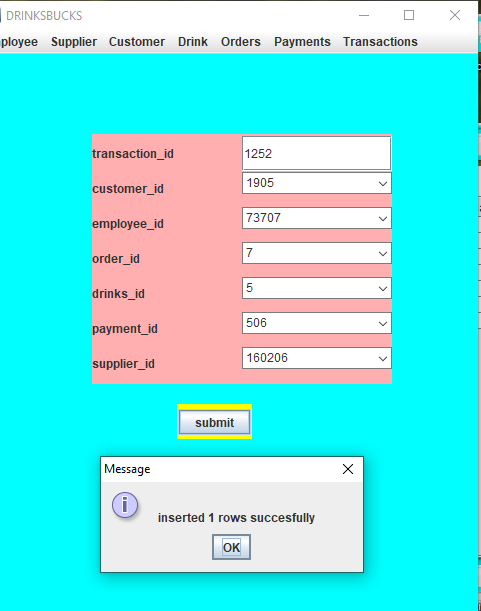


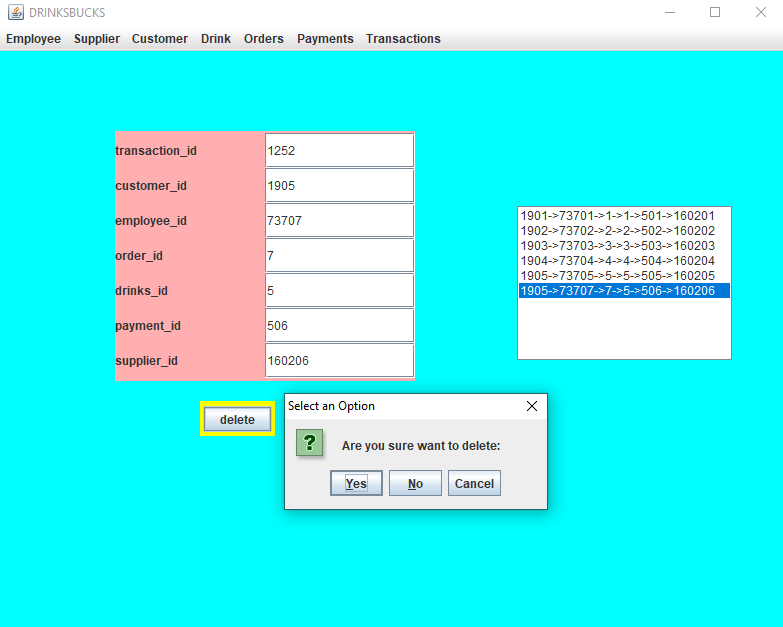


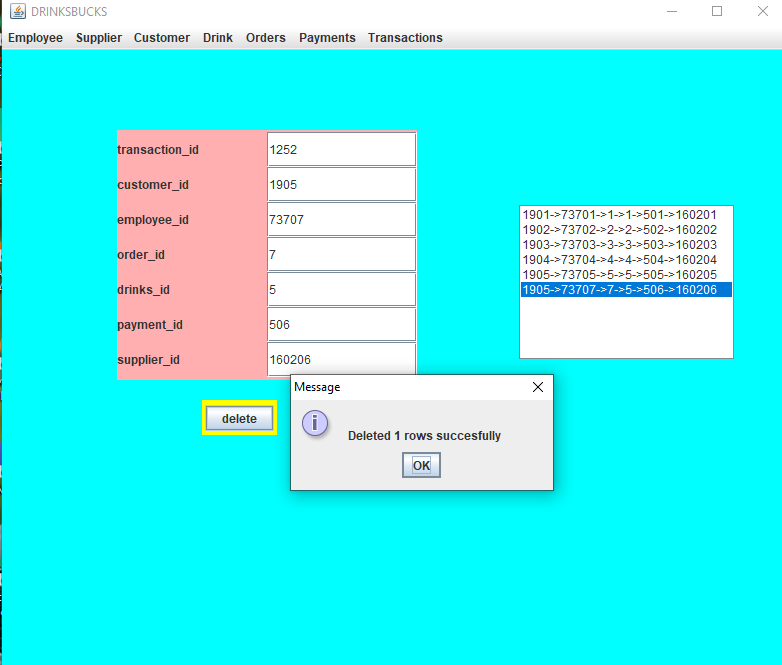


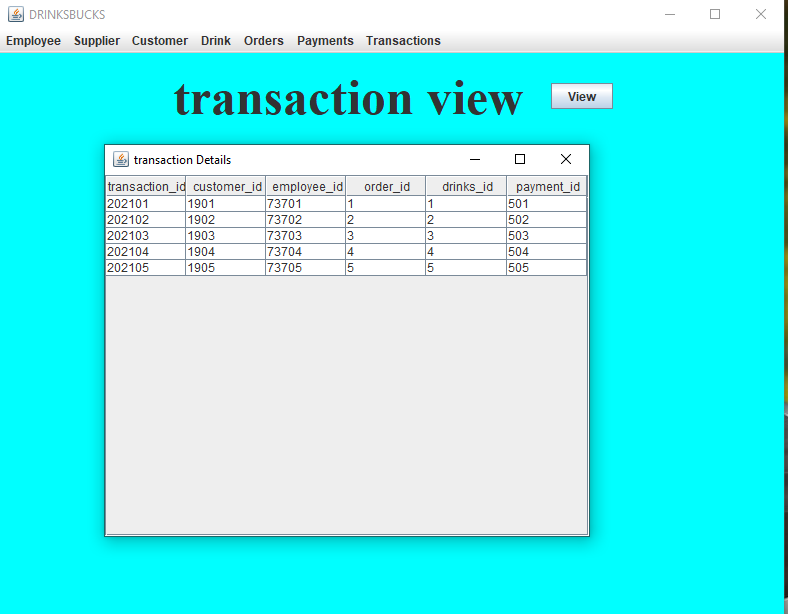


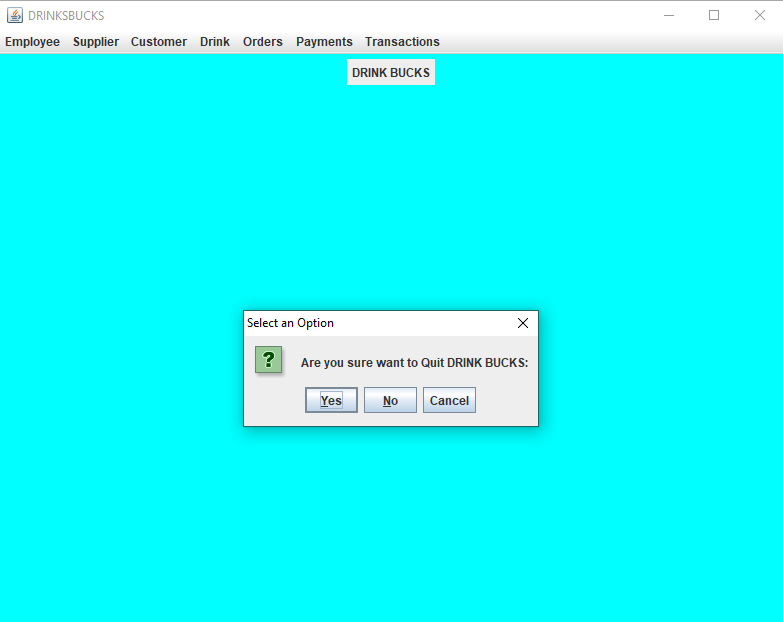


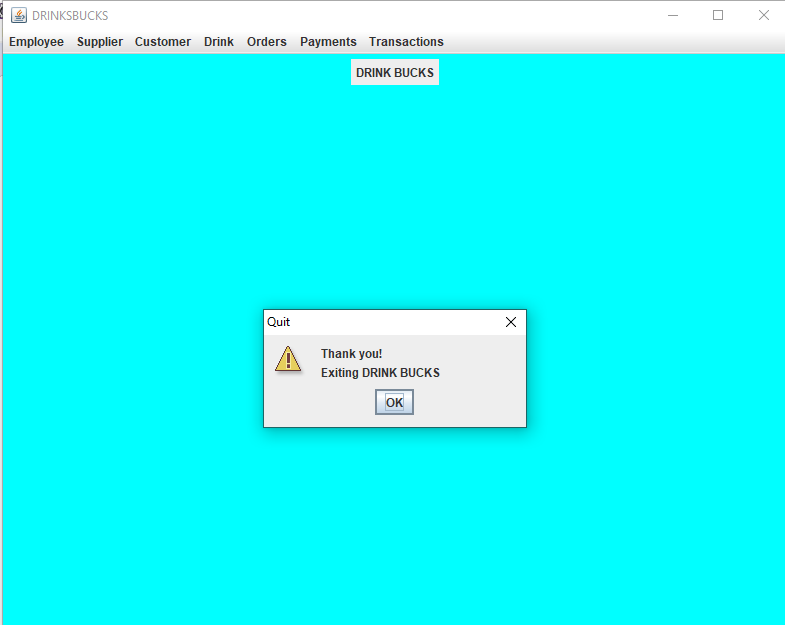




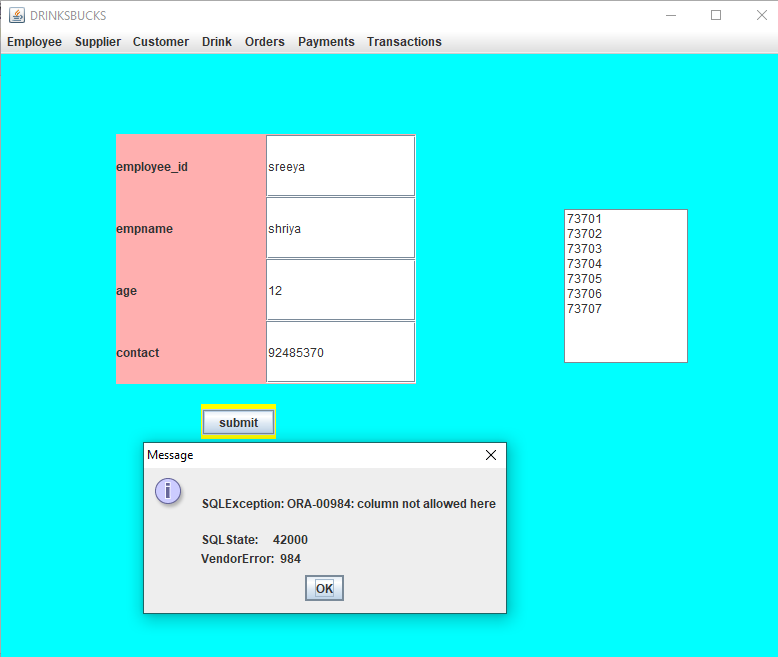


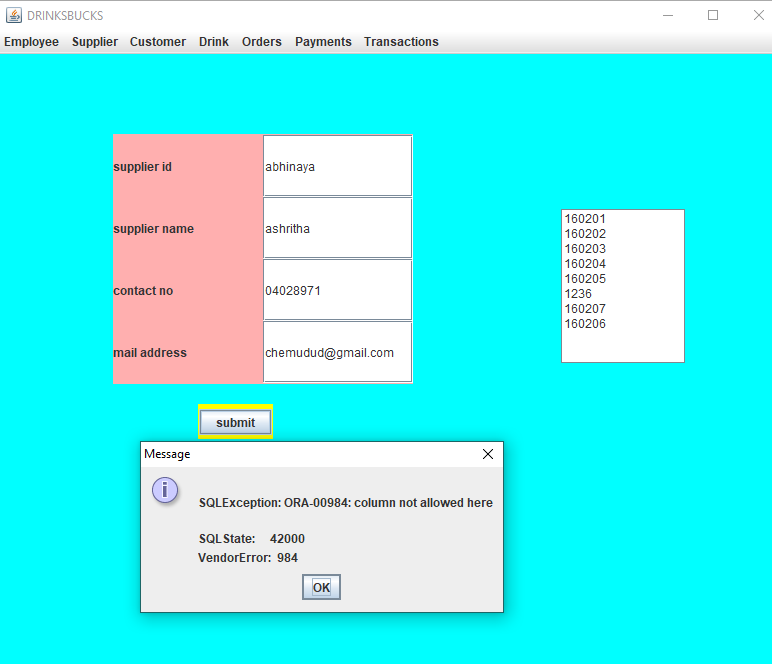




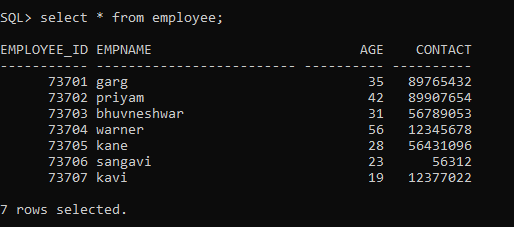


ADDITIONAL TESTCASES:

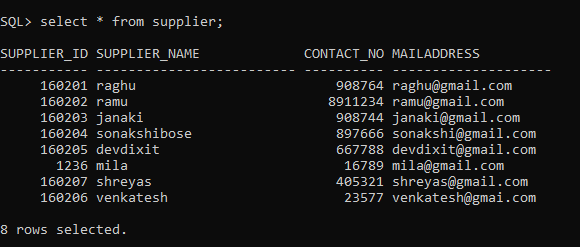




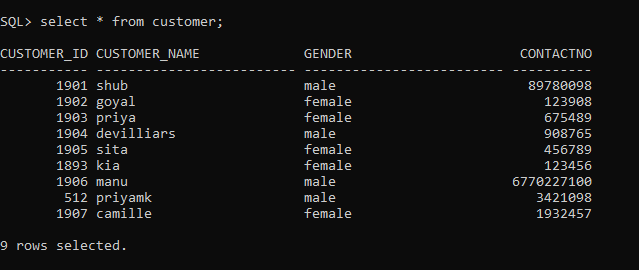
**The data entered in the above form is updated in the “employee” table of the Oracle database 11g as:**



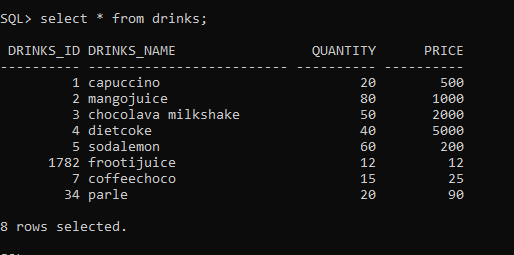
**The data entered in the above form is updated in the “supplier” table of the Oracle database 11g as:**



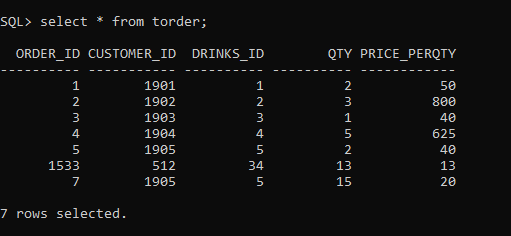
**The data entered in the above form is updated in the “customer” table of the Oracle database 11g as:**



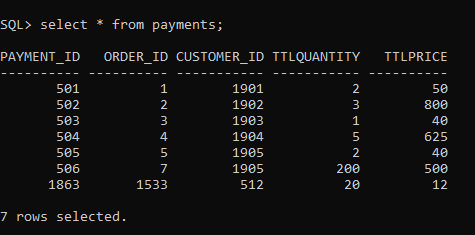
**The data entered in the above form is updated in the “drinks” table of the Oracle database 11g as:**



**The data entered in the above form is updated in the “torder” table of the Oracle database 11g as:**



**The data entered in the above form is updated in the “payments” table of the Oracle database 11g as:**



**The data entered in the above form is updated in the “transaction” table of the Oracle database 11g as:**



#### RESULTS:

I had successfully completed PROJECT on “DRINKSBUCKS”.

## DISCUSSION AND FUTURE WORK

So far this project has helped us in easing the customers life. The Customers order eventually is the crucial aspect here as it is the main parameter for which the project is designed for. It helps you to select your own drink without wasting your precious time and involves less human intervention.

In future the most probable aspects could be improvising payment options like adding paypal, paytm, Gift Cards etc and provide the customer with a visual graphical order status bar. Also it can be refined by adding a feature of sending an order ready notification to customer. Lastly, there could be one more additional feature added like providing deals and promotional offer details to home page.

#### CONCLUSION:

Thus, a Java SWING based DRINKSBUCKS is created which is connected to the Oracle 11g database. Therefore, all the entries and details are directly updated on their respective tables created in the database.

REFERENCES:

* <https://docs.oracle.com/javase/7/docs/api/>
* <https://www.javatpoint.com/dbms-tutorial>
* <https://en.wikipedia.org/wiki/Online_pharmacy>