

# **JAVA AWT BASED- CAFETERIA MANAGEMENT SYSTEM- 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

Kalthi Reddy Gayathri <1602-19-737-013>

Maram Sindhuja <1602-19-737-043>

Under the guidance of Ms S.Sreelakshmi



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 '**Cafeteria Management System**' is a project work of Ms. Kalthi Reddy Gayathri bearing roll no. 1602-19-737-013 and Ms. Maram Sindhuja bearing roll no. 1602-19-737-043, who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

Internal Examiner

Signature

External Examiner

## **ABSTRACT**

Cafeteria management system is a simple application which is designed for the purpose of calculating total bill of a customer in a café. This system elaborates basic concept of storing and generating ordered item's details. This application will have an account for every customer. The features of this application are displaying menu of the food items, orders placed by the customers, details of the customers. The customer can register or unregister an account, also can view his/her details and modify his/her details.

## INTRODUCTION REQUIREMENTS

<b>TABLES</b>	<b>ATTRIBUTES</b>
customer	id            number(5) firstname    varchar2(15) lastname     varchar2(15) houseno       varchar2(30) street        varchar2(20) city            varchar2(20) state          varchar2(20) pincode       number(6)
customer_phone	id            number(5) phone         number(10)
staff	name         varchar2(20) id             number(5) salary        number(5) age            number(5)
menu	category_name    varchar2(30) category_id       number(5) food_name         varchar2(30) food_quantity     number(5) cost                number(5,2)
bill	itemname        varchar2(20) price             number(5,2) quantity         number(20) bill_id            number customer_id       number(5)

## AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based CAFETERIA MANAGEMENT SYSTEM which enables to perform DML sql operations on all the tables in the database. These values are to be updated in the database using JDBC connectivity.

## ARCHITECTURE AND TECHNOLOGY

### Software used:

1. Operating system: windows 10
2. Frontend software: java Eclipse, java.version=16.0.1, java.vendor=Oracle Corporation
3. Backend software: sqlplus (instant\_client\_19\_8)

### Java AWT:

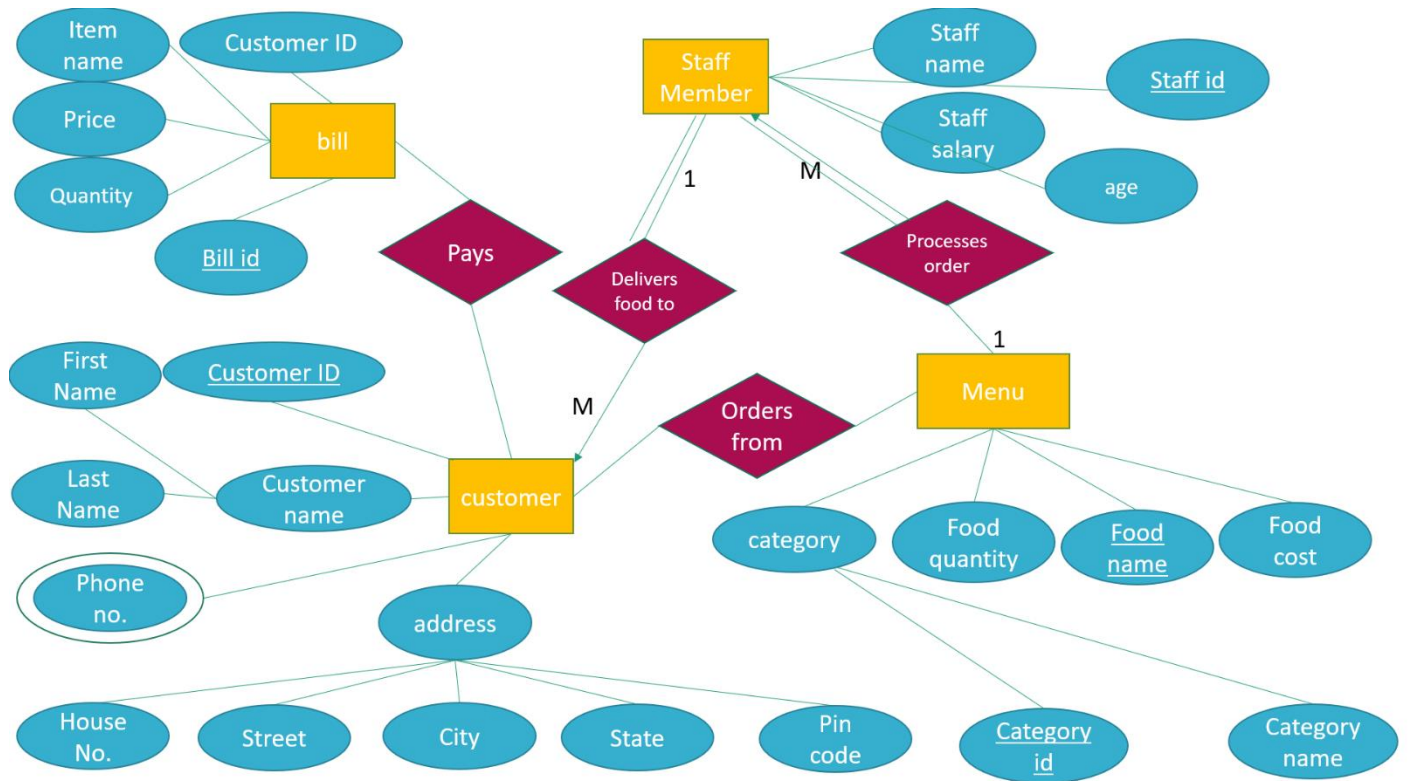
Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java. Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS. The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

### 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

Customer Table :

```
SQL> CREATE TABLE customer(  
  2  id          NUMBER(5),  
  3  firstname   VARCHAR2(15),  
  4  lastname    VARCHAR2(15),  
  5  houseno     VARCHAR2(30),  
  6  street      VARCHAR2(20),  
  7  city        VARCHAR(20),  
  8  state       VARCHAR2(20),  
  9  pincode     NUMBER(6)  
 10 );
```

Table created.

Customer\_phone table :

```
SQL> CREATE TABLE customer_phone(  
  2  id          NUMBER(5),  
  3  phone       NUMBER(10)  
  4 );
```

Table created.

Staff table :

```
SQL> CREATE TABLE staff(  
  2  name        VARCHAR2(20),  
  3  id          number(5),  
  4  salary      number(5),  
  5  age         number(5)  
  6 );
```

Table created.

Menu Table :

```
SQL> CREATE TABLE menu(  
 2  category_name      varchar2(30),  
 3  category_id        number(5),  
 4  food_name          varchar2(30),  
 5  food_quantity      number(5),  
 6  cost               number(5,2)  
 7 );
```

Table created.

Bill Table :

```
SQL> CREATE TABLE bill(  
 2  itemname           varchar2(20),  
 3  price              number(5,2),  
 4  quantity           number(20),  
 5  bill_id            number,  
 6  customer_id        number(5)  
 7 );
```

Table created.



## DML Operations

### Customer table :

```
SQL> INSERT INTO customer VALUES(43,'sindhuja','maram','F8','M. J. COLONY','Hyd','Telangana',500040);
1 row created.
SQL> INSERT INTO customer VALUES(13,'gayathri','kalthi','201','DR. CHARI','Hyd','Telangana',500044);
1 row created.
SQL> INSERT INTO customer VALUES(1029,'Tom','Fernandes','9-9/0','saraswathi','Tirupathi','Andhra Pradesh',517620);
1 row created.
SQL> INSERT INTO customer VALUES(941,'Hussai','Kishrit','G34','Lavani','Andheri','mumbai',400047);
1 row created.
SQL> INSERT INTO customer VALUES(113,'Guman','Dilrath','2-3/6/Q','kalis','Alipore','Kolkata',700005);
1 row created.
SQL> INSERT INTO customer VALUES(4410,'Sandhya','Rathi','GY41','Hanuman Galli','Ajmer','Rajasthan',305005);
1 row created.
SQL> INSERT INTO customer VALUES(789,'Alankritha','Elikshina','109','Jayanki','Mysore','Karnataka',570004);
1 row created.
SQL> INSERT INTO customer VALUES(9101,'Ishita','Bhalla','1-7-98/E','Maruthi','Erode','Tamil Nadu',638112);
1 row created.
SQL> commit;
Commit complete.
```

### Customer\_phone table :

```
SQL> INSERT INTO customer_phone VALUES(13,9874563210);
1 row created.
SQL> INSERT INTO customer_phone VALUES(43,2013654789);
1 row created.
SQL> INSERT INTO customer_phone VALUES(43,5632147890);
1 row created.
SQL> commit;
Commit complete.
```

Staff table :

```
SQL> INSERT INTO staff VALUES('kittu', 1031, 4000, 26);  
1 row created.  
SQL> INSERT INTO staff VALUES('Gamy', 2904, 4000, 33);  
1 row created.  
SQL> INSERT INTO staff VALUES('kiara', 1927, 4000, 19);  
1 row created.  
SQL> INSERT INTO staff VALUES('Buntu', 1722, 4000, 22);  
1 row created.  
SQL> INSERT INTO staff VALUES('Zahir', 1198, 4000, 31);  
1 row created.  
SQL> commit;  
Commit complete.
```

Menu Table :

```
SQL> INSERT INTO menu values('coffee', 1113, 'cold coffee', 150, 30);  
1 row created.  
SQL> INSERT INTO menu values('Fast Food', 4301, 'veg pizza', 100, 30);  
1 row created.  
SQL> INSERT INTO menu VALUES('Dessert', 1320, 'chocolate mouse', 40, 58.88);  
1 row created.  
SQL> INSERT INTO menu VALUES('coffee', 1945, 'Americano', 118, 50.95);  
1 row created.  
SQL> INSERT INTO menu VALUES('Dessert', 1229, 'Mango icecream', 78, 30.56);  
1 row created.  
SQL> INSERT INTO menu VALUES('Fast Food', 9207, 'chicken burger', 100, 70.79);  
1 row created.  
SQL> INSERT INTO menu VALUES('fast food', 501, 'french fries', 104, 55.21);  
1 row created.  
SQL> commit;  
Commit complete.
```

Bill table :

```
SQL> INSERT INTO bill VALUES('Americano', 50.95, 3, 1343, 43);  
1 row created.  
SQL> INSERT INTO bill VALUES('french fries', 55.21, 5, 891, 13);  
1 row created.  
SQL> INSERT INTO bill VALUES('Mango icecream', 30.56, 2, 7104, 1029);  
1 row created.  
SQL> INSERT INTO bill VALUES('chocolate mouse', 58.88, 7, 1610, 789);  
1 row created.  
SQL> INSERT INTO bill VALUES('chicken burger', 70.79, 1, 1002, 113);  
1 row created.  
SQL> commit;  
Commit complete.
```

## 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:

```
String dbURL = "jdbc:oracle:thin:@218.248.0.7:1521:rdbms";
String username = "it19737043";
String password = "vasavi";
Connection connection = DriverManager.getConnection(dbURL,username,password);
Statement st=connection.createStatement();
```

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

### Insert Queries :

Customer :

```
micInsert.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        getContentPane().removeAll();
        repaint();
        getContentPane().setBackground(Color.yellow);
        lbl= new JLabel("Insert into Customer table ");
        lbl.setFont(new Font("Verdana",Font.BOLD,20));
        lbl.setBounds(30,20,340,50);
        add(lbl);
        try {
            //label and textfield for customer ID
            lblcustId=new JLabel("Enter customer ID:");
            lblcustId.setBounds(50,80,110,30);
```

```

add(lblcustId);
tfcustId=new JTextField(10);
tfcustId.setBounds(220,80,190,30);
add(tfcustId);
//label and textfield for customer first name
lblcustfn=new JLabel("Enter customer's FirstName:");
lblcustfn.setBounds(50,130,190,30);
add(lblcustfn);
tfcustfn=new JTextField(10);
tfcustfn.setBounds(220,130,190,30);
add(tfcustfn);
//label and textfield for customer last name
lblcustln=new JLabel("Enter customer's LastName:");
lblcustln.setBounds(50,180,190,30);
add(lblcustln);
tfcustln=new JTextField(10);
tfcustln.setBounds(220,180,190,30);
add(tfcustln);
//label and textfield for customer HouseNo
lblcusthn=new JLabel("Enter customer's HouseNo:");
lblcusthn.setBounds(50,230,190,30);
add(lblcusthn);
tfcusthn=new JTextField(10);
tfcusthn.setBounds(220,230,190,30);
add(tfcusthn);
//label and textfield for customer streetname
lblcuststreet=new JLabel("Enter customer's streetName:");
lblcuststreet.setBounds(50,280,190,30);
add(lblcuststreet);
tfcuststreet=new JTextField(10);
tfcuststreet.setBounds(220,280,190,30);
add(tfcuststreet);
//label and textfield for customer city
lblcustcity=new JLabel("Enter customer's city:");
lblcustcity.setBounds(50,330,190,30);
add(lblcustcity);
tfcustcity=new JTextField(10);
tfcustcity.setBounds(220,330,190,30);
add(tfcustcity);
//label and textfield for customer state
lblcuststate=new JLabel("Enter customer's state:");
lblcuststate.setBounds(50,380,190,30);
add(lblcuststate);
tfcuststate=new JTextField(10);
tfcuststate.setBounds(220,380,190,30);
add(tfcuststate);
//label and textfield for customer pincode
lblcustpin=new JLabel("Enter customer's pincode:");
lblcustpin.setBounds(50,430,190,30);
add(lblcustpin);
tfcustpin=new JTextField(10);
tfcustpin.setBounds(220,430,190,30);
add(tfcustpin);
//button for inserting a customer
btncustSubmit=new JButton("Submit");
btncustSubmit.setBounds(120,500,100,30);
btncustSubmit.setBackground(Color.green);
add(btncustSubmit);

```

```

        btncustSubmit.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                try {
                    String dbURL =
"jdbc:oracle:thin:@218.248.0.7:1521:rdbms";
                    String username = "it19737043";
                    String password = "vasavi";
                    Connection connection =
DriverManager.getConnection(dbURL,username,password);
                    String id=tfcustId.getText();
                    String fn=tfcustfn.getText();
                    String ln=tfcustln.getText();
                    String hn=tfcusthn.getText();
                    String street=tfcuststreet.getText();
                    String city=tfcustcity.getText();
                    String state=tfcuststate.getText();
                    String pin=tfcustpin.getText();
                    String sql="INSERT INTO CUSTOMER
VALUES(?,?,?,?,?,?,?,?)";
                    PreparedStatement
stmt=connection.prepareStatement(sql);

                    stmt.setString(1,id);
                    stmt.setString(2,fn);
                    stmt.setString(3,ln);
                    stmt.setString(4,hn);
                    stmt.setString(5,street);
                    stmt.setString(6,city);
                    stmt.setString(7,state);
                    stmt.setString(8,pin);
                    int rows=stmt.executeUpdate();
                    stmt.executeUpdate("commit");
                    if(rows>0) {

JOptionPane.showMessageDialog(cafeteriaGUI.this,"Inserted succesfully.");
                    }
                    connection.close();
                }catch(SQLException se) {

JOptionPane.showMessageDialog(cafeteriaGUI.this,"Sorry, something went wrong couldn't
insert");

                    System.out.println("could not insert"+se);

                }

            }
        });

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

```

Update Queries :

Staff :

```

misModify.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        getContentPane().removeAll();
        repaint();
        getContentPane().setBackground(Color.pink);
        lbl= new JLabel("Modify Staff table ");
        lbl.setFont(new Font("Verdana",Font.BOLD,20));
        lbl.setBounds(30,15,340,50);
        add(lbl);
        try {
            String dbURL =
"jdbc:oracle:thin:@218.248.0.7:1521:rdbms";
            String username = "it19737043";
            String password = "vasavi";
            Connection connection =
DriverManager.getConnection(dbURL,username,password);
            String sql="select id from staff";
            PreparedStatement
stmt=connection.prepareStatement(sql);

            ResultSet rs= stmt.executeQuery(sql);

            String ids[] =new String[20];
            int i=0;
            while(rs.next())
            {
                ids[i]=rs.getString("id");
                i++;
            }
            int j=0,count=0;
            while(ids[j]!=null)
            {
                count++;j++;
            }
            String id[]=new String[count];
            for(j=0;j<count;j++) id[j]=ids[j];

            //label and jcombobox for selecting an id to be deleted
            lblmisDeleteStaffId=new JLabel("Select
StaffId:");

            lblmisDeleteStaffId.setBounds(50,100,170,30);
            lblmisDeleteStaffId.setFont(new
Font("Verdana",Font.BOLD,12));

            add(lblmisDeleteStaffId);
            cstaffId=new JComboBox(id);
            cstaffId.setBounds(220,100,170,30);
            cstaffId.setFont(new
Font("Verdana",Font.BOLD,12));

            add(cstaffId);

            //button for show details a staff
            btnstaffShow=new JButton("Show Staff Details");
            btnstaffShow.setBounds(120,150,170,30);

```

```

ID");

btnstaffShow.setBackground(Color.orange);
add(btnstaffShow);
//label and textfield for staff Id
lblstaffId=new JLabel("    You cannot modify Staff

lblstaffId.setBounds(50,200,190,30);
add(lblstaffId);
//label and textfield for staff Name
lblstaffName=new JLabel("Enter Staff Name:");
lblstaffName.setBounds(50,250,110,30);
add(lblstaffName);
tfstaffName=new JTextField(10);
tfstaffName.setBounds(220,250,190,30);
add(tfstaffName);
//label and textfield for staff salary
lblstaffSal=new JLabel("Enter staff salary:");
lblstaffSal.setBounds(50,300,190,30);
add(lblstaffSal);
tfstaffSal=new JTextField(10);
tfstaffSal.setBounds(220,300,190,30);
add(tfstaffSal);
//label and textfield for staff age
lblstaffAge=new JLabel("Enter staff age:");
lblstaffAge.setBounds(50,350,190,30);
add(lblstaffAge);
tfstaffAge=new JTextField(10);
tfstaffAge.setBounds(220,350,190,30);
add(tfstaffAge);

//button for updating a staff
btnstaffModify=new JButton("Modify");
btnstaffModify.setBounds(120,400,100,30);
btnstaffModify.setBackground(Color.green);
add(btnstaffModify);

btnstaffShow.addActionListener(new

ActionListener() {

    public void actionPerformed(ActionEvent e)

        try {
            String sid1=(String)

            String sql2="select

            PreparedStatement

            stmt2.executeUpdate();
            ResultSet rs2=

            while(rs2.next()) {

                tfstaffName.setText(rs2.getString(1));

                tfstaffSal.setText(String.valueOf(rs2.getInt(3)));

                tfstaffAge.setText(String.valueOf(rs2.getInt(4)));

        }

```



```

        }catch(Exception ae) {
            System.out.println("error");
        }

    }

});
btnstaffModify.addActionListener(new

ActionListener() {

    public void actionPerformed(ActionEvent e)

    {

        try {

            String sid=(String)

            String

            String

            String

            String sqlUpdate="update staff

            PreparedStatement

            int

            stmt.executeUpdate("commit");
            if(rows>0) {

                JOptionPane.showMessageDialog(cafeteriaGUI.this,"Modified succesfully.");
            }
            connection.close();
        }catch(SQLException se) {

            JOptionPane.showMessageDialog(cafeteriaGUI.this,"Sorry, something went wrong couldn't
            modify");

            System.out.println("could not

            update"+se);

        }

    }

});
//connection.close();
}catch(SQLException se) {
    se.printStackTrace();
}

}
});

```

Delete Table :

Menu :

```

mimDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        getContentPane().removeAll();
        repaint();
        getContentPane().setBackground(Color.cyan);
        lbl= new JLabel("Delete from Food Menu table ");
        lbl.setFont(new Font("Verdana",Font.BOLD,20));
        lbl.setBounds(30,15,340,50);
        add(lbl);
        try {
            String dbURL =
"jdbc:oracle:thin:@218.248.0.7:1521:rdbms";
            String username = "it19737043";
            String password = "vasavi";
            Connection connection =
DriverManager.getConnection(dbURL,username,password);
            String sql="select category_id from menu";
            PreparedStatement
stmt=connection.prepareStatement(sql);

            ResultSet rs= stmt.executeQuery(sql);

            String ids[] =new String[20];
            int i=0;
            while(rs.next())
            {
                ids[i]=rs.getString("category_id");
                i++;
            }
            int j=0,count=0;
            while(ids[j]!=null)
            {
                count++;j++;
            }
            String id[]=new String[count];
            for(j=0;j<count;j++) id[j]=ids[j];

            //label and jcombobox for selecting an id to be deleted
            lblmimDeleteCategoryId=new JLabel("Select
Category Id:");

            lblmimDeleteCategoryId.setBounds(50,100,170,30);
            lblmimDeleteCategoryId.setFont(new
Font("Verdana",Font.BOLD,12));

            add(lblmimDeleteCategoryId);
            cCategoryId=new JComboBox(id);
            cCategoryId.setBounds(220,100,170,30);
            cCategoryId.setFont(new
Font("Verdana",Font.BOLD,12));

            add(cCategoryId);

            //button for show details a menu
            btnMenuShow=new JButton("Show FoodMenu Details");
            btnMenuShow.setBounds(120,150,200,30);

```

Name:");

```

btnMenuShow.setBackground(Color.orange);
add(btnMenuShow);
//label and textfield for category name
lblcategoryName=new JLabel("Enter category
Name:");

lblcategoryName.setBounds(50,200,190,30);
add(lblcategoryName);
tfcategoryName=new JTextField(10);
tfcategoryName.setBounds(220,200,190,30);
add(tfcategoryName);
//label and textfield for category ID
lblcategoryId=new JLabel("Enter category ID:");
lblcategoryId.setBounds(50,250,110,30);
add(lblcategoryId);
tfcategoryId=new JTextField(10);
tfcategoryId.setBounds(220,250,190,30);
add(tfcategoryId);
//label and textfield for food name
lblfoodName=new JLabel("Enter Food Name:");
lblfoodName.setBounds(50,300,190,30);
add(lblfoodName);
tffoodName=new JTextField(10);
tffoodName.setBounds(220,300,190,30);
add(tffoodName);
//label and textfield for food quantity
lblfoodQ=new JLabel("Enter Food quantity:");
lblfoodQ.setBounds(50,350,190,30);
add(lblfoodQ);
tffoodQ=new JTextField(10);
tffoodQ.setBounds(220,350,190,30);
add(tffoodQ);
//label and textfield for cost
lblcost=new JLabel("Enter cost:");
lblcost.setBounds(50,400,190,30);
add(lblcost);
tfcost=new JTextField(10);
tfcost.setBounds(220,400,190,30);
add(tfcost);
//button for inserting a menu item
btnmenuDelete=new JButton("Delete");
btnmenuDelete.setBounds(120,450,100,30);
btnmenuDelete.setBackground(Color.green);
add(btnmenuDelete);

```

```

ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        try {
            String sid1=(String)
            cCategoryId.getSelectedItemAt();
            String sql2="select
            category_name,category_id,food_name,food_quantity,cost from menu where CATEGORY_ID="+sid1;
            PreparedStatement
            stmt2=connection.prepareStatement(sql2);
            stmt2.executeUpdate();
        }
    }
}

```

```

stmt2.executeQuery(sql2);

    tfcategoryName.setText(rs2.getString(1));
tfcategoryId.setText(rs2.getString(2));
    tffoodName.setText(rs2.getString(3));
    tffoodQ.setText(rs2.getString(4));
    tfcost.setText(rs2.getString(5));

    }

} catch (Exception ae) {
    System.out.println("error");
}

});
btnmenuDelete.addActionListener(new

    public void actionPerformed(ActionEvent e)

        try {
            String sid=(String)

            String sqlDelete="delete from

            PreparedStatement

            stmt3.setString(1,sid);
            int

            stmt.executeUpdate("commit");
            if(rows>0) {

                tfcategoryId.setText("");
                tffoodName.setText("");
                tffoodQ.setText("");
                tfcost.setText("");

                JOptionPane.showMessageDialog(cafeteriaGUI.this,"Deleted succesfully.");
            }
            connection.close();
        } catch (SQLException se) {

            JOptionPane.showMessageDialog(cafeteriaGUI.this,"Sorry, something went wrong couldn't
delete");

            System.out.println("could not
delete"+se);

        }

    });

```

```

        //connection.close();
    }catch(SQLException se) {
        se.printStackTrace();
    }
}
});

```

## Display Queries :

### Bill :

```

mibDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        getContentPane().removeAll();
        repaint();
        getContentPane().setBackground(Color.GREEN);
        lbl= new JLabel("Display Bill table ");
        lbl.setFont(new Font("Verdana",Font.BOLD,20));
        lbl.setBounds(30,20,340,50);
        add(lbl);
        try {
            JTextArea mibDisplayContent=new JTextArea(40,70);
            mibDisplayContent.setBounds(100,100,800,400);
            mibDisplayContent.setFont(new
Font("Verdana",Font.BOLD,12));

            add(mibDisplayContent);
            mibDisplayContent.setEditable(false);
            mibDisplayContent.setText("ITEMNAME \tPRICE
\tQUANTITY \tBILL_ID \tCUSTOMER_ID\n\n");

            String dbURL =
"jdbc:oracle:thin:@218.248.0.7:1521:rdbms";

            DriverManager.getConnection(dbURL,username,password);
            Statement st=connection.createStatement();
            ResultSet rs=st.executeQuery("select * from
bill");

            while(rs.next()) {

                mibDisplayContent.append(rs.getString(1).trim()+"\t"+rs.getDouble(2)+"\t"+rs.getInt(3)+"
\t"+rs.getInt(4)+"\t"+rs.getInt(5)+"\n");
            }
            connection.close();
        }catch(SQLException se) {
            se.printStackTrace();
        }
    }
});
//Add ActionListener to insert button in bill
mibInsert.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {

```

```

getContentPane().removeAll();
repaint();
getContentPane().setBackground(Color.yellow);
lbl= new JLabel("Insert into Bill table ");
lbl.setFont(new Font("Verdana",Font.BOLD,20));
lbl.setBounds(30,20,340,50);
add(lbl);
try {
    //label and textfield for item name
    lblitemName=new JLabel("Enter item Name:");
    lblitemName.setBounds(50,80,190,30);
    add(lblitemName);
    tfitemName=new JTextField(10);
    tfitemName.setBounds(220,80,190,30);
    add(tfitemName);
    //label and textfield for price
    lblprice=new JLabel("Enter price:");
    lblprice.setBounds(50,130,110,30);
    add(lblprice);
    tfprice=new JTextField(10);
    tfprice.setBounds(220,130,190,30);
    add(tfprice);
    //label and textfield for food name
    lblquantity=new JLabel("Enter Quantity:");
    lblquantity.setBounds(50,180,190,30);
    add(lblquantity);
    tfquantity=new JTextField(10);
    tfquantity.setBounds(220,180,190,30);
    add(tfquantity);
    //label and textfield for billid
    lblbid=new JLabel("Enter Bill Id:");
    lblbid.setBounds(50,230,190,30);
    add(lblbid);
    tfbid=new JTextField(10);
    tfbid.setBounds(220,230,190,30);
    add(tfbid);
    //label and textfield for customer id
    lblcustomerId=new JLabel("Enter customer ID:");
    lblcustomerId.setBounds(50,280,190,30);
    add(lblcustomerId);
    tfcustomerId=new JTextField(10);
    tfcustomerId.setBounds(220,280,190,30);
    add(tfcustomerId);
    //button for inserting a menu item
    btnbillSubmit=new JButton("Submit");
    btnbillSubmit.setBounds(120,330,100,30);
    btnbillSubmit.setBackground(Color.green);
    add(btnbillSubmit);

    btnbillSubmit.addActionListener(new

ActionListener(){

    public void actionPerformed(ActionEvent e){
        try {
            String dbURL =
                "jdbc:oracle:thin:@218.248.0.7:1521:rdbms";
            String username =
                "it19737043";
            String password = "vasavi";

```

```
}
}
});
```

**MAIN PROGRAM:**

```

package dbmsAssignment;

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;

public class cafeteriaGUI extends JFrame{
JMenu mnuCustomer,mnuStaff,mnuMenu,mnuBill;
JMenuItem micInsert,micDelete,micModify,micDisplay,
misInsert,misDelete,misModify,misDisplay,
mimInsert,mimDelete,mimModify,mimDisplay,
mibInsert,mibDelete,mibModify,mibDisplay;
JTextArea micDisplayContent,misDisplayContent,mimDisplayContent,mibDisplayContent;
JLabel lblcustId,lblcustfn,lblcustln,lblcusthn,lblcuststreet,lblcustcity,lblcuststate,lblcustpin,lbl,
lblstaffName,lblstaffId,lblstaffSal,lblstaffAge,
lblcategoryId,lblcategoryName,lblfoodName,lblfoodQ,lblcost,
lblitemName,lblquantity,lblprice,lblbid,lblcustomerId,
lblmisDeleteStaffId,lblmicDeleteCustId,lblmimDeleteCategoryId,lblmibDeletebillId;
JTextField tfcustId,tfcustfn,tfcustln,tfcusthn,tfcuststreet,tfcustcity,tfcuststate,tfcustpin,
tfstaffName,tfstaffId,tfstaffSal,tfstaffAge,
tfcategoryId,tfcategoryName,tffoodName,tffoodQ,tfcost,
tfitemName,tfquantity,tfprice,tfbid,tfcustomerId;
JButton btnstaffDelete,btnstaffShow,btnstaffModify,btnstaffSubmit,
btnCustDelete,btnCustShow,btnCustModify,btncustSubmit,
btnMenuShow, btnmenuDelete,btnmenuModify,btnmenuSubmit,
btnBillShow,btnBillDelete,btnBillModify,btnbillSubmit;
JComboBox cstaffId,cCustId,cCategoryId,cBillId;

public cafeteriaGUI(){
getContentPane().setBackground(Color.lightGray);
setSize(1200,800);
//menu
JMenuBar mb=new JMenuBar();
mnuCustomer = new JMenu("Customer");
mnuStaff = new JMenu("Staff");
mnuMenu = new JMenu("Food Menu");
mnuBill = new JMenu("Bill");
micInsert = new JMenuItem("Insert");
micDelete = new JMenuItem("Delete");
micModify = new JMenuItem("Modify");
micDisplay= new JMenuItem("Display");

```



```
misInsert = new JMenuItem("Insert");
misDelete = new JMenuItem("Delete");
misModify = new JMenuItem("Modify");
misDisplay= new JMenuItem("Display");
mimInsert = new JMenuItem("Insert");
mimDelete = new JMenuItem("Delete");
mimModify = new JMenuItem("Modify");
mimDisplay= new JMenuItem("Display");
mibInsert = new JMenuItem("Insert");
mibDelete = new JMenuItem("Delete");
mibModify = new JMenuItem("Modify");
mibDisplay= new JMenuItem("Display");
mnuCustomer.add(micInsert);
mnuCustomer.add(micDelete);
mnuCustomer.add(micModify);
mnuCustomer.add(micDisplay);
mnuStaff.add(misInsert);
mnuStaff.add(misDelete);
mnuStaff.add(misModify);
mnuStaff.add(misDisplay);
mnuMenu.add(mimInsert);
mnuMenu.add(mimDelete);
mnuMenu.add(mimModify);
mnuMenu.add(mimDisplay);
mnuBill.add(mibInsert);
mnuBill.add(mibDelete);
mnuBill.add(mibModify);
mnuBill.add(mibDisplay);
mb.add(mnuCustomer);
mb.add(mnuStaff);
mb.add(mnuMenu);
mb.add(mnuBill);
setJMenuBar(mb);

getContentPane().removeAll();
repaint();
JLabel welcome=new JLabel(" WELCOME TO CAFETERIA MANAGEMENT SYSTEM");
welcome.setFont(new Font("Verdana",Font.BOLD,28));
add(welcome);

//Add ActionListener for display button in customer
//Add ActionListener to insert button in customer
//add ActionListener to delete button in customer
//add ActionListener to modify button in customer
//Add ActionListener for display button in staff
//Add ActionListener to insert button in staff
//add ActionListener to delete button in staff
//add ActionListener to delete button in staff
```

```
//Add ActionListener for display button in food menu
//Add ActionListener to insert button in food menu
//add ActionListener to delete button in food menu
//add ActionListener to modify button in food menu
//add ActionListener to display all rows in bill table
//Add ActionListener to insert button in bill
//Add ActionListener to delete button in bill
//Add ActionListener to modify button in bill
```

```
setVisible(true);
setLayout(null);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
```

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

```
    cafeteriaGUI h=new cafeteriaGUI();
    h.setTitle("Cafeteria Management System");
```

```
}
```

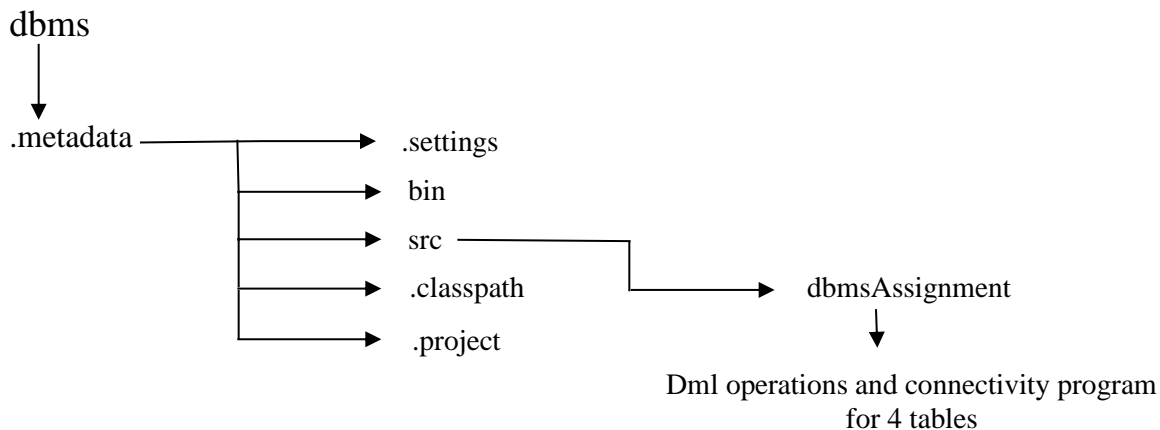
```
}
```

## **Github links and folder structure:**

### **Link :**

1. <https://github.com/gayathri0124/Database-Management-Systems/blob/main/cafeteriaGUI.java>
2. <https://github.com/sindhuja8520/Cafeteria-database-management-system.git>

### **Folder Structure :**



## TESTING

Cafeteria Management System

Customer Staff Food Menu Bill

### Display Customer table

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
941	Hussai	Kishrit	G34	Lavani	Andheri	mumbail	400047
62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharastra	600040
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Sandhya	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharastra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070

If we try to insert a customer with id=67, it shows error because ID is a primary key

Cafeteria Management System

Customer Staff Food Menu Bill

### Insert into Customer table

Enter customer ID:

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:

Enter customer's city:

Enter customer's state:

Enter customer's pincode:

Submit


Message



Sorry, couldn't insert. java.sql.SQLIntegrityConstraintViolationException: ORA-00001: unique constraint (IT19737043.PK\_ID) violated

OK

If we enter pincode greater than 6, Then it shows an error.

 Cafeteria Management System

Customer Staff Food Menu Bill

### Insert into Customer table

Enter customer ID:

67

Enter customer's FirstName:

Dinesh

Enter customer's LastName:

Agarwal

Enter customer's HouseNo:

41-I/56

Enter customer's streetName:

Riverpal nagar

Enter customer's city:

Banglore

Enter customer's state:


Hyderabad

Enter customer's pincode:

5000020


Submit

Message

 Sorry, couldn't insert, java.sql.SQLException: ORA-01438: value larger than specified precision allowed for this column

OK

## Triggers

 Cafeteria Management System

Customer Staff Food Menu Bill

### Insert into staff table

Enter Staff Name:

hari

Enter Staff ID:

1214

Enter staff salary:


4000

Enter staff age:

31

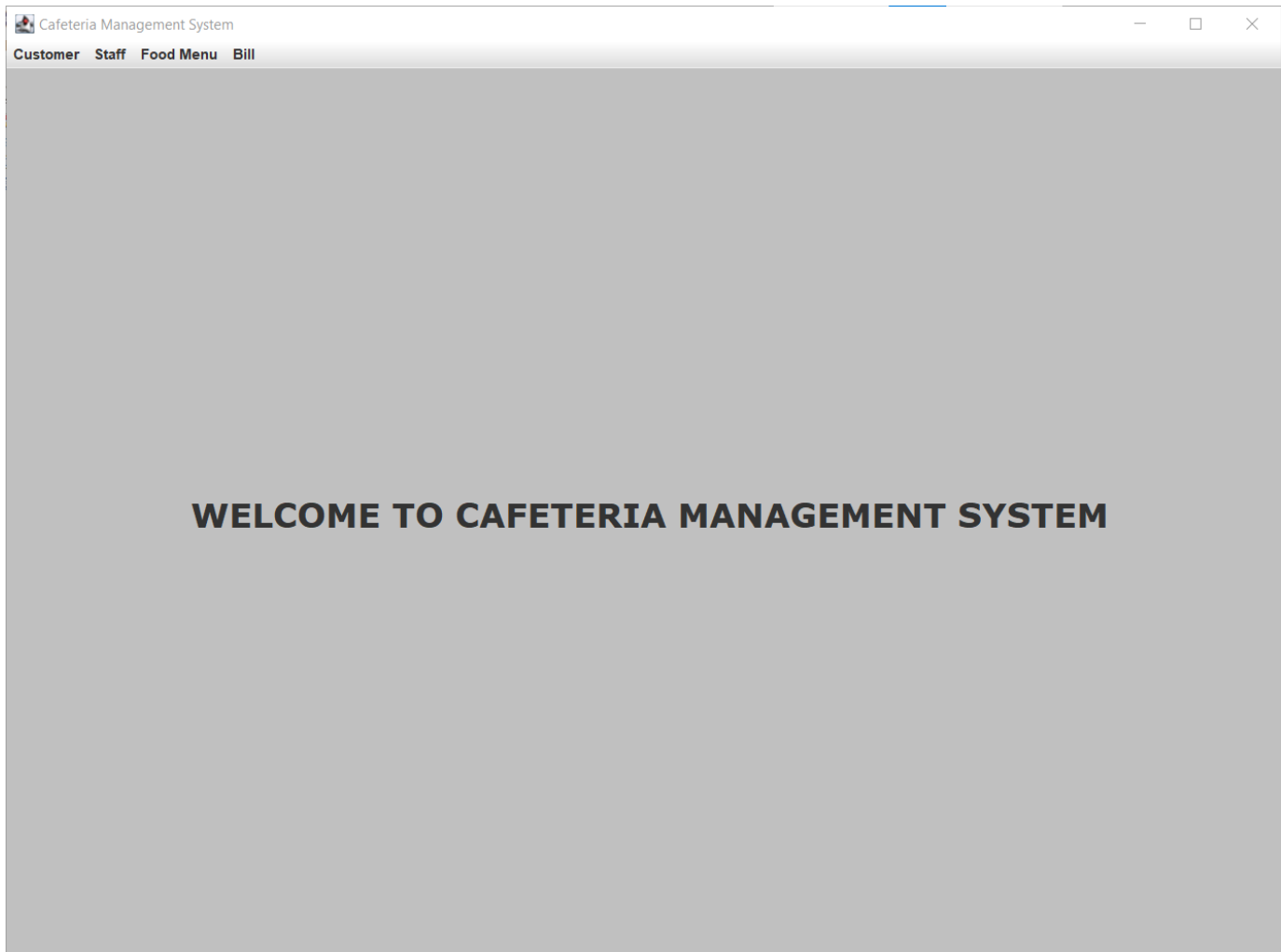
Submit

Message

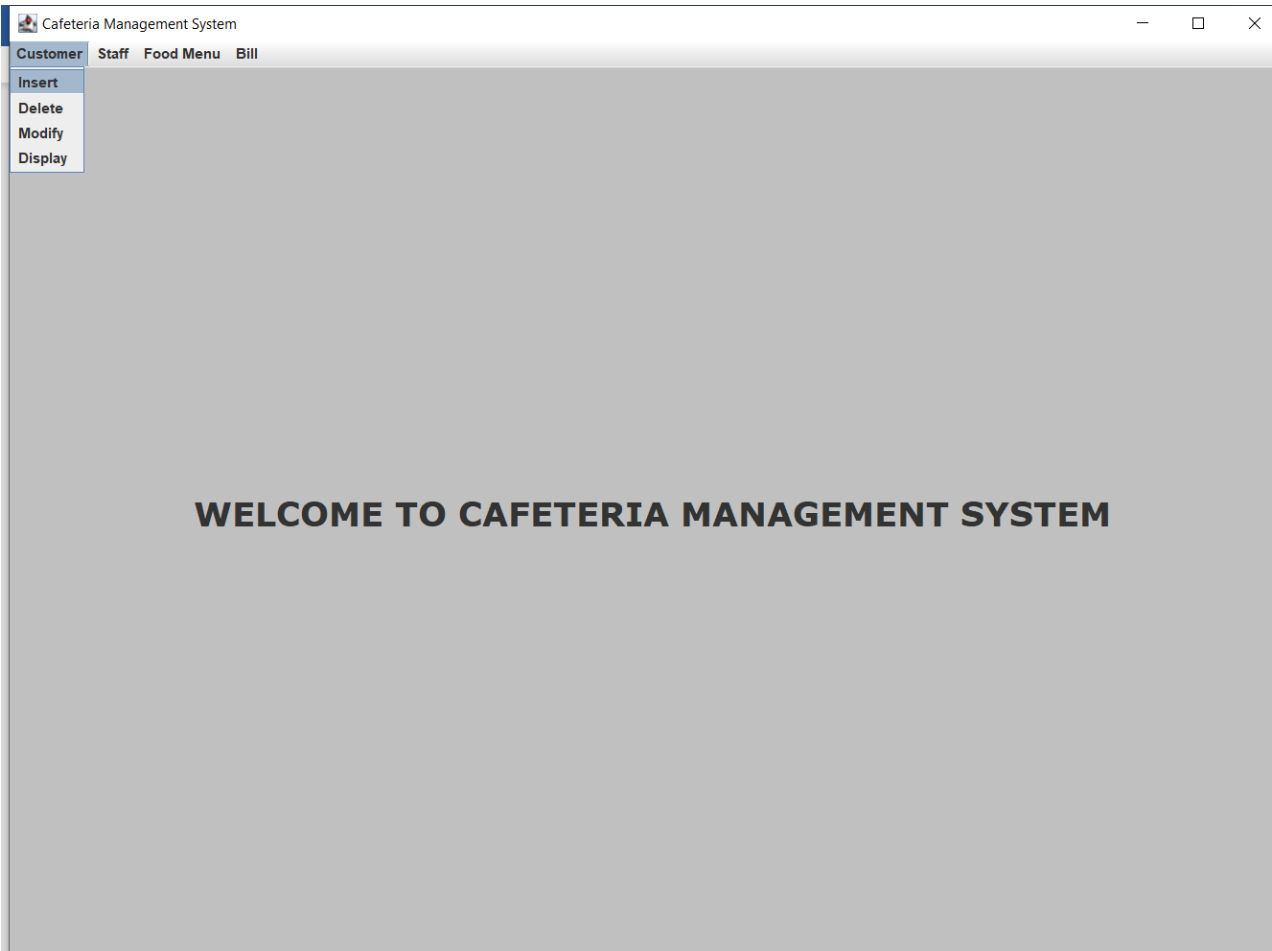
 Sorry, couldn't insert, java.sql.SQLException: ORA-20013: Lower case is not allowed  
ORA-06512: at "IT19737043.NOLOWERCASESTAFF", line 3  
ORA-04088: error during execution of trigger 'IT19737043.NOLOWERCASESTAFF'

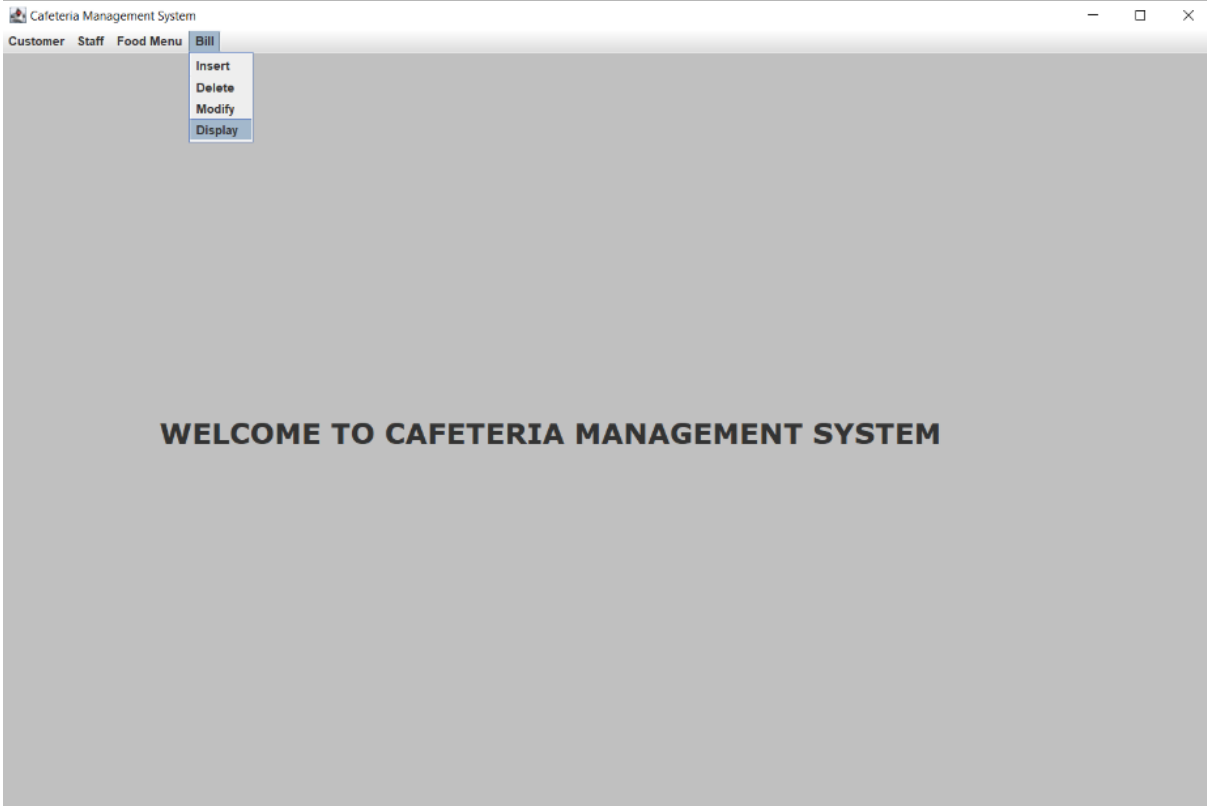
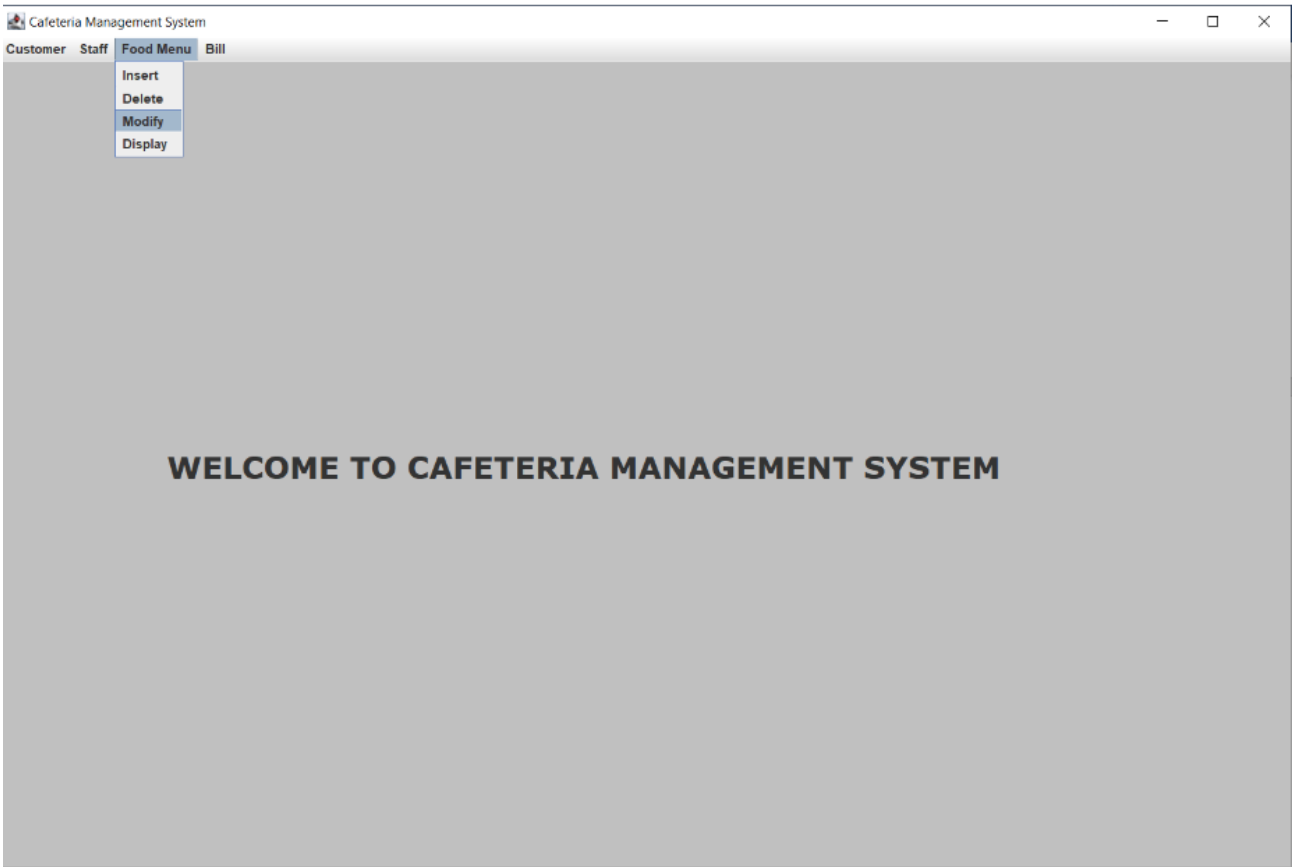
OK

## RESULTS




CAFETERIA MANAGEMENT SYSTEM





## Details of customer in database

 C:\Users\User\instantclient\_19\_8\sqlplus.exe

SQL&gt; select \*from customer;

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040

13	Gayathri	kalathi	201	DR.CHARI	Hyd	Telangana	500044
----	----------	---------	-----	----------	-----	-----------	--------

57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
----	-------	--------	-----	-------------	-------	-------	--------

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
941	Hussai	Kishrit	G34	Lavani	Andheri	mumbail	400047

62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharastra	600040
----	---------	-------	-------	--------------	--------	------------	--------

113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
-----	-------	---------	---------	-------	---------	---------	--------

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
4410	Sandhya	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005

56	Divyanka	tripati	388	mayur vihar	mumbai	maharastra	600070
----	----------	---------	-----	-------------	--------	------------	--------

67	shreya	goshal	890	udipi	chennai	tamil nadu	800070
----	--------	--------	-----	-------	---------	------------	--------



## Inserting into customer table

 Cafeteria Management System

Customer Staff Food Menu Bill

### Insert into Customer table

Enter customer ID:

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:


Enter customer's city:

Enter customer's state:

Enter customer's pincode:

Submit


Before Inserting, the customer table has

 Cafeteria Management System

CustomerStaffFood MenuBill

### Display Customer table

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
941	Hussai	Kishrit	G34	Lavani	Andheri	mumbai	400047
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Sandhya	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharastra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070

 Cafeteria Management System

CustomerStaffFood MenuBill

### Insert into Customer table

Enter customer ID:

62

Enter customer's FirstName:

Mallika

Enter customer's LastName:

Singh

Enter customer's HouseNo:

45G-9

Enter customer's streetName:

Kamala Nagar

Enter customer's city:

Mumbai

Enter customer's state:


Maharastra

Enter customer's pincode:

600040

Submit

Message

 Inserted succesfully.

OK

After Inserting a row into customer table

Cafeteria Management System

CustomerStaffFood MenuBill

Display Customer table

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharashtra	600040
941	Hussai	Kishrit	G34	Lavani	Andheri	mumbai	400047
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Sandhya	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharashtra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070



## CAFETERIA MANAGEMENT SYSTEM

```
SQL> select *from customer;
```

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
941	Hussai	Kishrit	G34	Lavani	Andheri	mumbail	400047
62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharastra	600040
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Sandhya	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharastra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070

```
9 rows selected.
```

```
SQL>
```

## Deleting a row from customer table

 Cafeteria Management System[Customer](#) [Staff](#) [Food Menu](#) [Bill](#)

### Delete from Customer table

Select Customer Id:

13

[Show Customer Details](#)

Enter customer ID:

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:


Enter customer's city:

Enter customer's state:

Enter customer's pincode:

[Delete](#)

Selecting an id which is to be deleted

 Cafeteria Management System

Customer   Staff   Food Menu   Bill

### Delete from Customer table

Select Customer Id:

941

13

43

56

57

62

67

113

941

Show Customer

Enter customer ID:

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:


Enter customer's city:

Enter customer's state:

Enter customer's pincode:

Delete

## Showing the details of the selected customer id

 Cafeteria Management System

Customer Staff Food Menu Bill

## Delete from Customer table

Select Customer Id: 941 ▼

Show Customer Details

Enter customer ID:

941

Enter customer's FirstName:

Hussai

Enter customer's LastName:

Kishrit

Enter customer's HouseNo:

G34

Enter customer's streetName:

Lavani

Enter customer's city:

Andheri


Enter customer's state:

mumbai

Enter customer's pincode:

400047

Delete

 Cafeteria Management System

Customer   Staff   Food Menu   Bill

### Delete from Customer table

Select Customer Id: 

941

Show Customer Details

Enter customer ID:

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:

Enter customer's city:

Enter customer's state:

Enter customer's pincode:

Delete


Message

Deleted succesfully.

OK



After deleting, table does not have 941 customer


 Cafeteria Management System

CustomerStaffFood MenuBill

Display Customer table

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharastra	600040
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Sandhya	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharastra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070

## Selecting an id to be modified

 Cafeteria Management System

Customer Staff Food Menu Bill

### Modify Customer table

Select Customer Id:

Show Customer Details

You cannot modify customer ID

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:


Enter customer's streetName:

Enter customer's city:

Enter customer's state:

Enter customer's pincode:

Modify

 Cafeteria Management System

Customer   Staff   Food Menu   Bill

### Modify Customer table

Select Customer Id:

4410

13

43

56

57

62

67

113

4410

Show Customer Id

You cannot modify customer Id

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:

Enter customer's city:

Enter customer's state:

Enter customer's pincode:

Modify



## Modify Customer table

Select Customer Id:

4410

**Show Customer Details**

You cannot modify customer ID

Enter customer's FirstName: Sandhya

Enter customer's LastName: Rathi

Enter customer's HouseNo: GY41

Enter customer's streetName: Hanuman Galli


Enter customer's city: Ajmer

Enter customer's state: Rajasthan

Enter customer's pincode: 305005

**Modify**

## After modifying

 Cafeteria Management System

Customer Staff Food Menu Bill

## Modify Customer table

Select Customer Id:

Show Customer Details

You cannot modify customer ID

Enter customer's FirstName:

Enter customer's LastName:

Enter customer's HouseNo:

Enter customer's streetName:


Enter customer's city:

Enter customer's state:


Enter customer's pincode:

Modify

Message

 Modified succesfully.

OK

 Cafeteria Management System

Customer   Staff   Food Menu   Bill

Display Customer table

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharastra	600040
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Deepika	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharastra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070

## CAFETERIA MANAGEMENT SYSTEM

C:\Users\User\instantclient\_19\_8\sqlplus.exe

SQL&gt; select \*from customer;

ID	FIRSTNAME	LASTNAME	HOUSENO	STREET	CITY	STATE	PINCODE
43	Sindhuja	maram	F8	M.J.COLONY	Hyd	Telangana	500040
13	Gayathri	kalthi	201	DR.CHARI	Hyd	Telangana	500044
57	Raman	Bhalla	768	mayur vihar	delhi	Delhi	700030
62	Mallika	Singh	45G-9	Kamala Nagar	Mumbai	Maharashtra	600040
113	Guman	Dilrath	2-3/6/Q	kalis	Alipore	Kolkata	700005
4410	Deepika	Rathi	GY41	Hanuman Galli	Ajmer	Rajasthan	305005
56	Divyanka	tripati	388	mayur vihar	mumbai	maharashtra	600070
67	shreya	goshal	890	udipi	chennai	tamil nadu	800070

## Displaying staff table

SQL&gt; select \*from staff;

NAME	ID	SALARY	AGE
Bunty	1722	4000	29
Kiara	1927	4000	19
kittu	1031	4000	26
Gamyra	2904	4000	33

Cafeteria Management System

CustomerStaffFood MenuBill

Display Staff table

NAME	ID	SALARY	AGE
Bunty	1722	4000	29
Kiara	1927	4000	19
kittu	1031	4000	26
Gamya	2904	4000	33



CAFETERIA MANAGEMENT SYSTEM

Cafeteria Management System

CustomerStaffFood MenuBill

Insert into staff table

Enter Staff Name:

Bittu

Enter Staff ID:

1986

Enter staff salary:

4000

Enter staff age:

27

Submit

Message

Inserted succesfully.

OK

Cafeteria Management System

CustomerStaffFood MenuBill

Display Staff table

NAME	ID	SALARY	AGE
Buntty	1722	4000	29
Bittu	1986	4000	27
Kiara	1927	4000	19
kittu	1031	4000	26
Gamya	2904	4000	33

```
SQL> select *from staff;

NAME                ID      SALARY      AGE
-----
Bunty                1722     4000        29
Bittu                1986     4000        27
Kiara                1927     4000        19
kittu                1031     4000        26
Gamy                 2904     4000        33

SQL>
```

Cafeteria Management System

CustomerStaffFood MenuBill

### Delete from Staff table

Select StaffId:

1927

1031

1722

1927

1986

2904

Show Staff

Enter Staff Name:

Enter Staff ID:

Enter staff salary:

Enter staff age:

Delete

Cafeteria Management System

CustomerStaffFood MenuBill

Delete from Staff table

Select StaffId:1927

Show Staff Details

Enter Staff Name:Kiara

Enter Staff ID:1927

Enter staff salary:4000

Enter staff age:19

Delete

Cafeteria Management System

CustomerStaffFood MenuBill

Delete from Staff table

Select StaffId:1927

Show Staff Details

Enter Staff Name:

Enter Staff ID:

Enter staff salary:

Enter staff age:

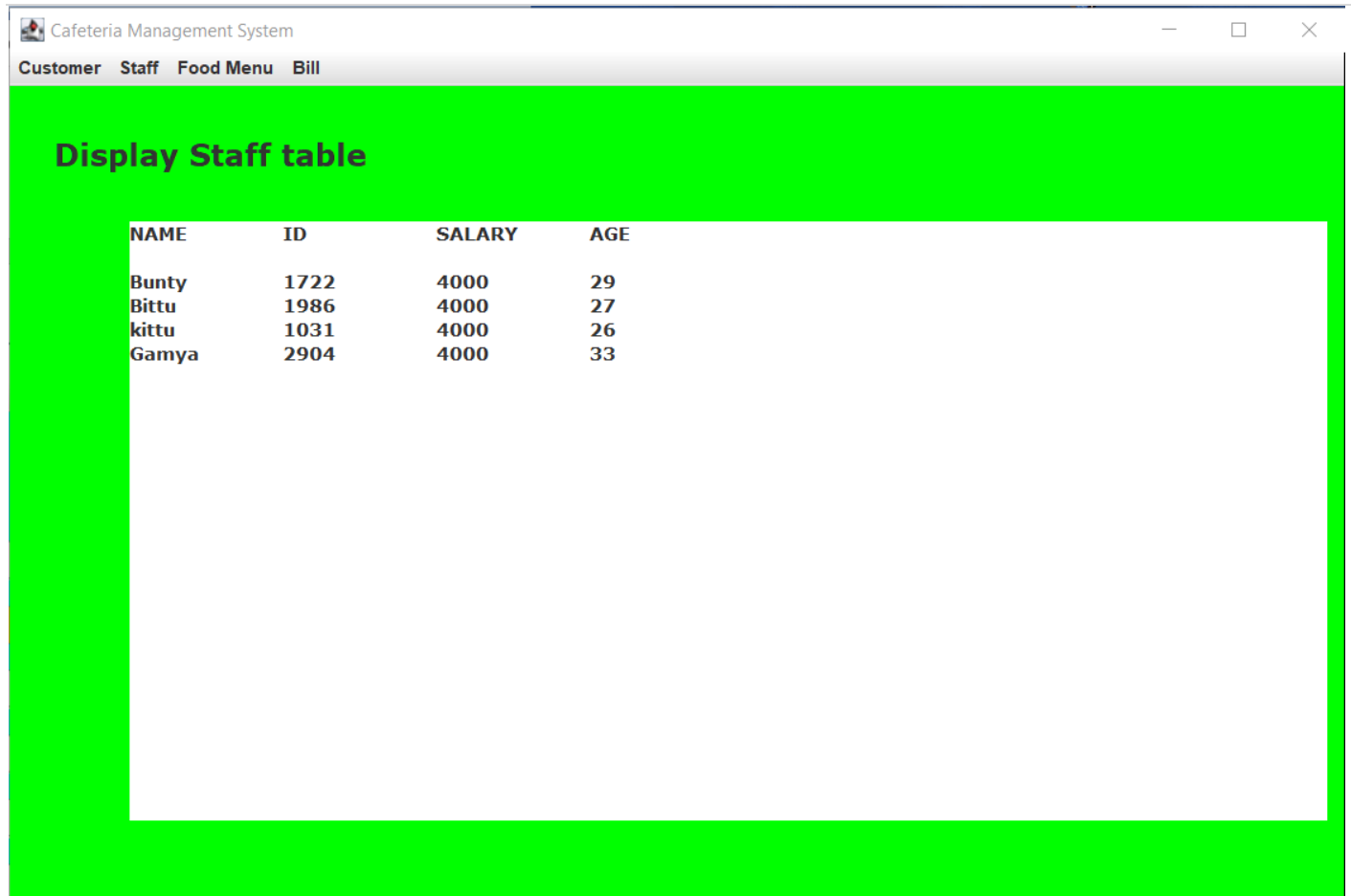
Delete

Message

Deleted succesfully.

OK


## CAFETERIA MANAGEMENT SYSTEM



```
SQL> select *from staff;
```

NAME	ID	SALARY	AGE
Bunty	1722	4000	29
Bittu	1986	4000	27
kittu	1031	4000	26
Gamyra	2904	4000	33

```
SQL>
```

 Cafeteria Management System

Customer Staff Food Menu Bill

## Modify Staff table

Select StaffId:

Show Staff Details

You cannot modify Staff ID


Enter Staff Name:

Enter staff salary:

Enter staff age:

Modify

## CAFETERIA MANAGEMENT SYSTEM

 Cafeteria Management System

Customer Staff Food Menu Bill

## Modify Staff table

Select StaffId:

Show Staff Details

You cannot modify Staff ID

Enter Staff Name:

Enter staff salary:

Enter staff age:

Modify

Cafeteria Management System

CustomerStaffFood MenuBill

### Modify Staff table

Select StaffId:1031

Show Staff Details

You cannot modify Staff ID

Enter Staff Name:kitty

Enter staff salary:4001

Enter staff age:27

Modify

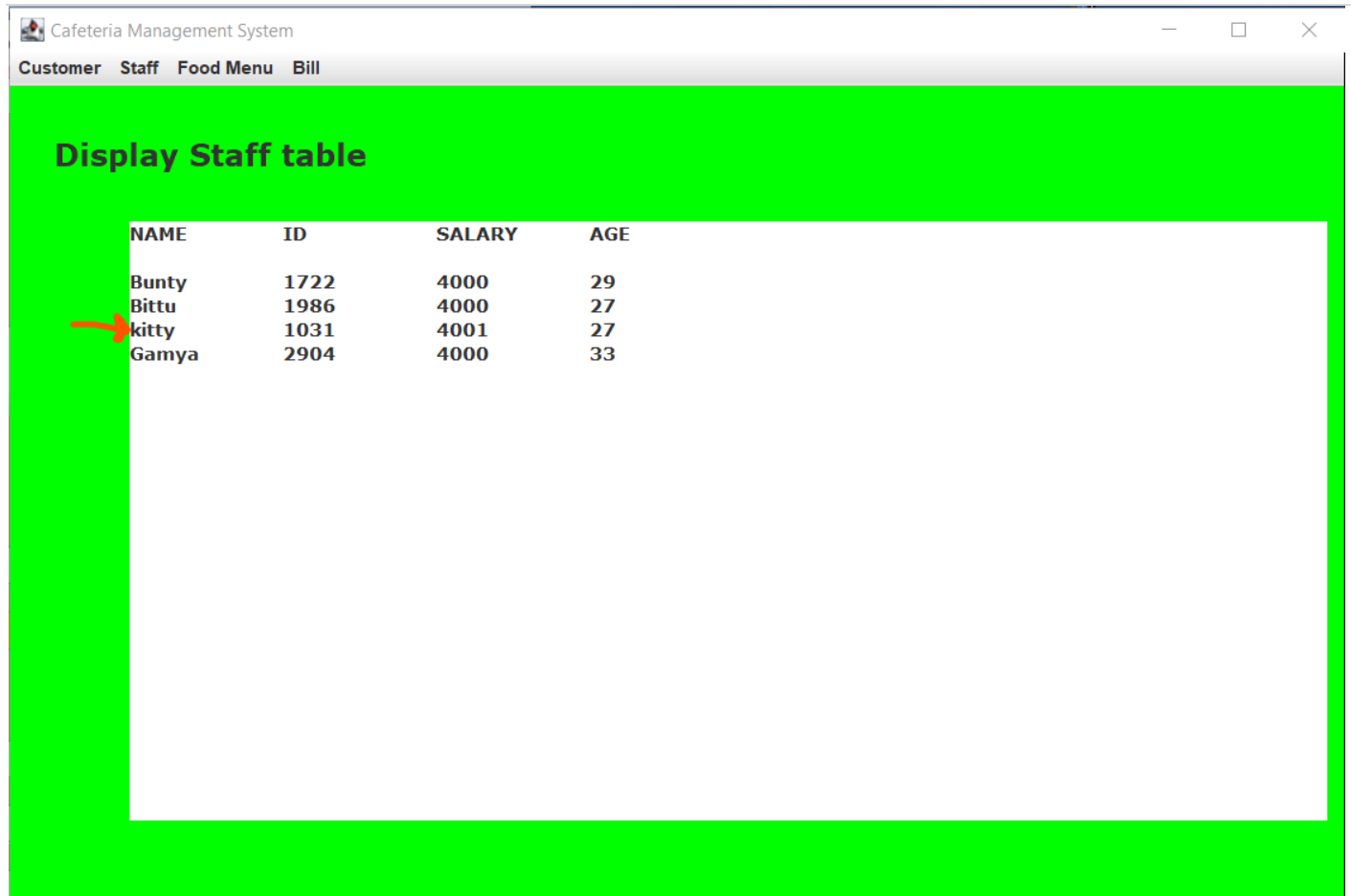
Message

Modified succesfully.

OK



## CAFETERIA MANAGEMENT SYSTEM



NAME	ID	SALARY	AGE
Bunty	1722	4000	29
Bittu	1986	4000	27
kitty	1031	4001	27
Gamya	2904	4000	33

```
SQL> select *from staff;
```

NAME	ID	SALARY	AGE
Bunty	1722	4000	29
Bittu	1986	4000	27
kitty	1031	4001	27
Gamya	2904	4000	33

```
SQL> 
```

Displaying menu table

C:\Users\User\instantclient\_19\_8\sqlplus.exe

```
SQL> select *from menu;
```

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
coffee	150	1113	cold coffee
Fast Food	100	4301	veg pizza
Dessert	40	1320	tiramisu

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
coffee	118	1945	Americano
fast food	100	9207	chicken burger
fast food	104	501	french fries

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
fast food	150	1231	Pasta

CAFETERIA MANAGEMENT SYSTEM

Cafeteria Management System

CustomerStaffFood MenuBill

Display Food Menu table

CATEGORY_NAME	CATEGORY_ID	FOOD_NAME	FOOD_QUANTITY	COST
coffee	1113	cold coffee	150	30.0
Fast Food	4301	veg pizza	100	30.0
Dessert	1320	tiramisu	40	58.88
coffee	1945	Americano	118	50.95
fast food	9207	chicken burger	100	70.79
fast food	501	french fries	104	55.21
fast food	1231	Pasta	150	70.0

Inserting into menu table

Cafeteria Management System

Customer

Staff

Food Menu

Bill

### Insert into Food Menu table

Enter category Name:

Fast food

Enter category ID:

1230

Enter Food Name:

Quesadilla

Enter Food quantity:

30

Enter cost:

299.9

Submit

Message

Inserted succesfully.

OK

Cafeteria Management System

Customer

Staff

Food Menu

Bill

### Display Food Menu table

CATEGORY_NAME	CATEGORY_ID	FOOD_NAME	FOOD_QUANTITY	COST
coffee	1113	cold coffee	150	30.0
Fast Food	4301	veg pizza	100	30.0
Dessert	1320	tiramisu	40	58.88
coffee	1945	Americano	118	50.95
fast food	9207	chicken burger	100	70.79
fast food	501	french fries	104	55.21
fast food	1231	Pasta	150	70.0
Fast food	1230	Quesadilla	30	299.9

## CAFETERIA MANAGEMENT SYSTEM


C:\Users\User\instantclient\_19\_8\sqlplus.exe

SQL&gt; select \*from menu;

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
coffee	150	1113	cold coffee
Fast Food	100	4301	veg pizza
Dessert	40	1320	tiramisu

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
coffee	118	1945	Americano
fast food	100	9207	chicken burger
fast food	104	501	french fries

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
fast food	150	1231	Pasta
Fast food	30	1230	Quesadilla



Deleting from food menu

Cafeteria Management System

Customer

Staff

Food Menu

Bill

Delete from Food Menu table

Select Category Id:

9207

Show FoodMenu Details

Enter category Name:

Enter category ID:

Enter Food Name:

Enter Food quantity:

Enter cost:

Delete

Cafeteria Management System

Customer

Staff

Food Menu

Bill

Delete from Food Menu table

Select Category Id:

9207

Show FoodMenu Details

Enter category Name:

fast food

Enter category ID:

9207

Enter Food Name:

chicken burger

Enter Food quantity:

100

Enter cost:

70.79

Delete

Cafeteria Management System

CustomerStaffFood MenuBill

Delete from Food Menu table

Select Category Id:

9207

Show FoodMenu Details

Enter category Name:

Enter category ID:

Enter Food Name:

Enter Food quantity:

Enter cost:

Delete

Message

Deleted succesfully.

OK

Cafeteria Management System

CustomerStaffFood MenuBill

Display Food Menu table

CATEGORY_NAME	CATEGORY_ID	FOOD_NAME	FOOD_QUANTITY	COST
coffee	1113	cold coffee	150	30.0
Fast Food	4301	veg pizza	100	30.0
Dessert	1320	tiramisu	40	58.88
coffee	1945	Americano	118	50.95
fast food	501	french fries	104	55.21
fast food	1231	Pasta	150	70.0
Fast food	1230	Quesadilla	30	299.9



## CAFETERIA MANAGEMENT SYSTEM

```
SQL> select *from menu;
```

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
coffee	150	30	1113 cold coffee
Fast Food	100	30	4301 veg pizza
Dessert	40	58.88	1320 tiramisu

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
coffee	118	50.95	1945 Americano
fast food	104	55.21	501 french fries
fast food	150	70	1231 Pasta


CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
Fast food	30	299.9	1230 Quesadilla

```
7 rows selected.
```

```
SQL>
```

## Modifying food menu table

### Before modifying

 Cafeteria Management System

Customer Staff Food Menu Bill

### Modify Food Menu table

Select Category Id:

Show FoodMenu Details

Enter category Name:

You cannot modify category ID

Enter Food Name:

Enter Food quantity:

Enter cost:

Modify

Cafeteria Management System

CustomerStaffFood MenuBill

### Modify Food Menu table

Select Category Id:1113

Show FoodMenu Details

Enter category Name:coffee

You cannot modify category ID

Enter Food Name:cold coffee

Enter Food quantity:180

Enter cost:70

Modify

Message

Modified succesfully.

OK

Cafeteria Management System

Customer

Staff

Food Menu

Bill

Display Food Menu table

CATEGORY_NAME	CATEGORY_ID	FOOD_NAME	FOOD_QUANTITY	COST
coffee	1113	cold coffee	180	70.0
Fast Food	4301	veg pizza	100	30.0
Dessert	1320	tiramisu	40	58.88
coffee	1945	Americano	118	50.95
fast food	501	french fries	104	55.21
fast food	1231	Pasta	150	70.0
Fast food	1230	Quesadilla	30	299.9

## CAFETERIA MANAGEMENT SYSTEM

```
SQL> select *from menu;
```

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
FOOD_QUANTITY	COST		
coffee	180	70	1113 cold coffee
Fast Food	100	30	4301 veg pizza
Dessert	40	58.88	1320 tiramisu

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
FOOD_QUANTITY	COST		
coffee	118	50.95	1945 Americano
fast food	104	55.21	501 french fries
fast food	150	70	1231 Pasta

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
FOOD_QUANTITY	COST		
Fast food	30	299.9	1230 Quesadilla

```
7 rows selected.
```

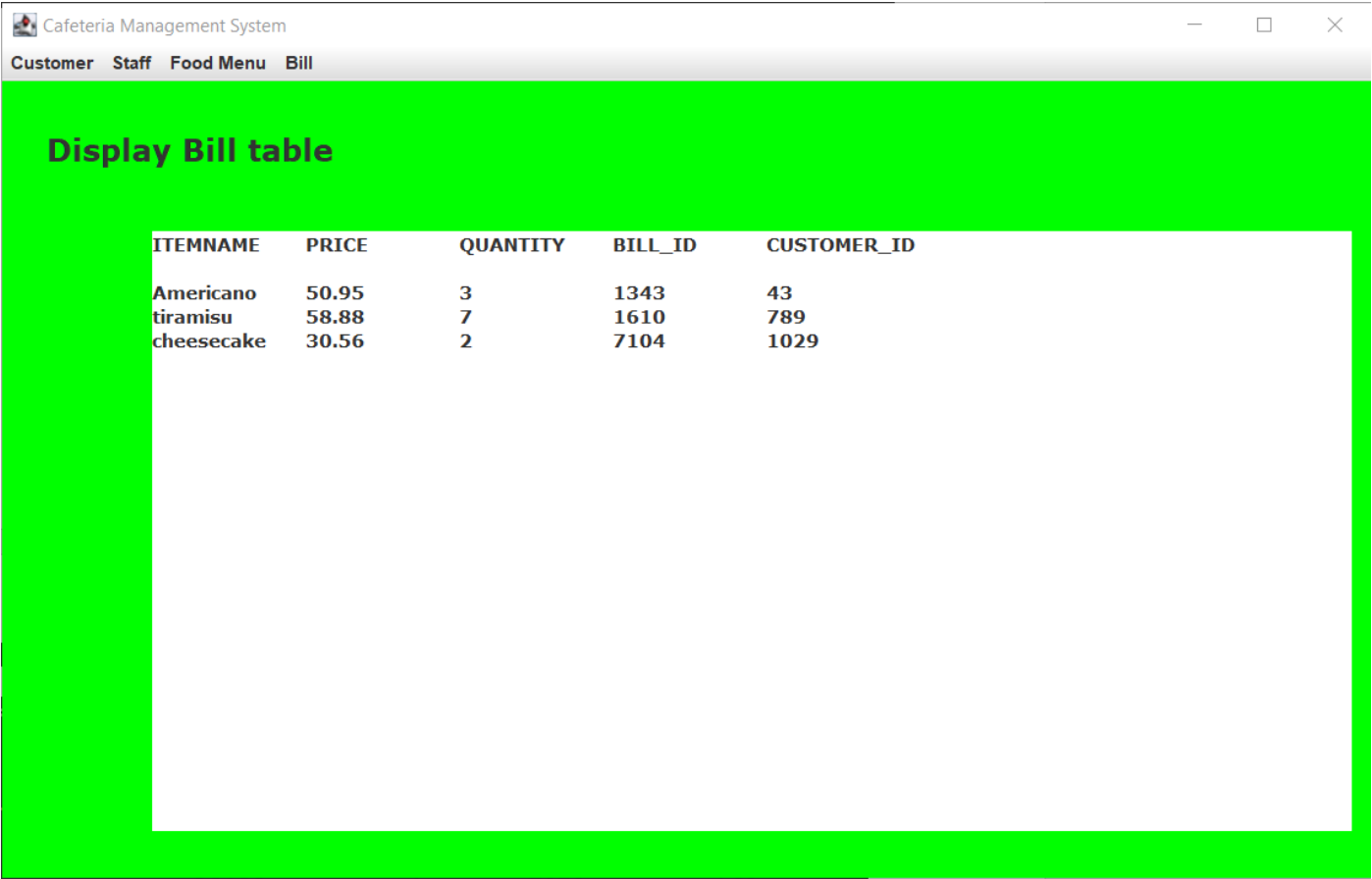
```
SQL>
```

Displaying bill table

```
SQL> select *from bill;

ITEMNAME          PRICE  QUANTITY  BILL_ID  CUSTOMER_ID
-----
Americano         50.95      3      1343      43
tiramisu          58.88      7      1610      789
cheesecake        30.56      2      7104     1029

SQL>
```



CAFETERIA MANAGEMENT SYSTEM

Cafeteria Management System

CustomerStaffFood MenuBill

Insert into Bill table

Enter item Name:

veg pizza

Enter price:

299.9

Enter Quantity:

5

Enter Bill Id:

4590

Enter customer ID:

13

Submit

Message

Inserted succesfully.

OK

Cafeteria Management System

CustomerStaffFood MenuBill

Display Bill table

ITEMNAME	PRICE	QUANTITY	BILL_ID	CUSTOMER_ID
Americano	50.95	3	1343	43
tiramisu	58.88	7	1610	789
cheesecake	30.56	2	7104	1029
veg pizza	299.9	5	4590	13

```
SQL> select *from bill;
```

ITEMNAME	PRICE	QUANTITY	BILL_ID	CUSTOMER_ID
Americano	50.95	3	1343	43
tiramisu	58.88	7	1610	789
cheesecake	30.56	2	7104	1029
veg pizza	299.9	5	4590	13

```
SQL>
```

## Deleting a row from bill table

Cafeteria Management System

Customer Staff Food Menu Bill

### Delete from Bill table

Select Bill Id:

Show Bill Details

Enter item Name:

Enter price:

Enter Quantity:

Enter Bill Id:

Enter customer ID:

Delete



CAFETERIA MANAGEMENT SYSTEM

Cafeteria Management System

CustomerStaffFood MenuBill

### Delete from Bill table

Select Bill Id:

1343

Show Bill Details

Enter item Name:

Enter price:

Enter Quantity:

Enter Bill Id:

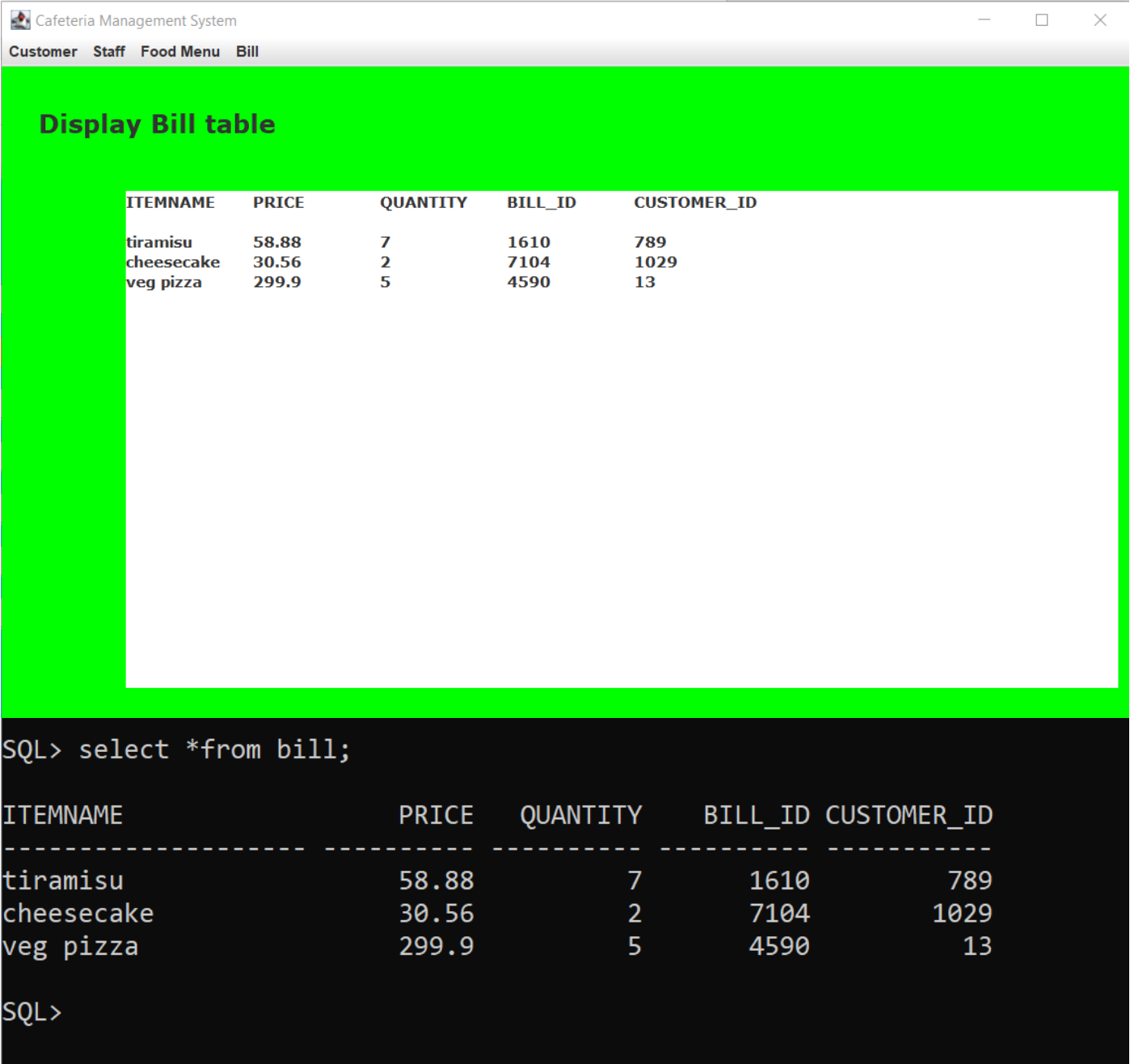
Enter customer ID:

Delete

Message

Deleted succesfully.

OK



The screenshot shows a web application window titled "Cafeteria Management System" with a navigation bar containing "Customer", "Staff", "Food Menu", and "Bill". The main content area has a green header with the text "Display Bill table". Below this is a table with the following data:

ITEMNAME	PRICE	QUANTITY	BILL_ID	CUSTOMER_ID
tiramisu	58.88	7	1610	789
cheesecake	30.56	2	7104	1029
veg pizza	299.9	5	4590	13

Below the table is a terminal window with the following content:


```
SQL> select *from bill;

ITEMNAME          PRICE    QUANTITY    BILL_ID  CUSTOMER_ID
-----
tiramisu          58.88         7         1610         789
cheesecake        30.56         2         7104        1029
veg pizza         299.9         5         4590         13

SQL>
```

Modifying a row from bill table

## CAFETERIA MANAGEMENT SYSTEM

 Cafeteria Management System

Customer Staff Food Menu Bill

## Modify Bill table

Select Bill Id:


Enter item Name:

Enter price:

Enter Quantity:

You cannot modify Bill Id

Enter customer ID:

 Cafeteria Management System

Customer Staff Food Menu Bill

## Modify Bill table

Select Bill Id:

Enter item Name:


Enter price:

Enter Quantity:

You cannot modify Bill Id

Enter customer ID:

Message

 **Modified succesfully.**

Cafeteria Management System

CustomerStaffFood MenuBill

Display Bill table

ITEMNAME	PRICE	QUANTITY	BILL_ID	CUSTOMER_ID
tiramisu	58.88	7	1610	789
cheesecake	300.0	2	7104	1029
veg pizza	299.9	5	4590	13

```
SQL> select *from bill;
```

ITEMNAME	PRICE	QUANTITY	BILL_ID	CUSTOMER_ID
tiramisu	58.88	7	1610	789
cheesecake	300	2	7104	1029
veg pizza	299.9	5	4590	13

```
SQL>
```

## **DISCUSSION AND FUTURE WORK**

So far, we have only implemented DML commands through this java GUI application. In future, we will try to include login ID, give authorities to admin only. Also we will try to make the use of html web pages and including food icons and will try to make it attractive and efficient.

## **REFERENCES**

<https://youtu.be/ea0HhETxOIs>

<https://docs.oracle.com/javase/7/docs/api/>

## CASE STUDY RELATED LAB WORK (SQL QUERIES)

### LAB 1 – DDL Operations

```
SQL> CREATE TABLE customer(  
 2   id          NUMBER(5),  
 3   firstname   VARCHAR2(15),  
 4   lastname    VARCHAR2(15),  
 5   houseno     VARCHAR2(30),  
 6   street      VARCHAR2(20),  
 7   city        VARCHAR(20),  
 8   state       VARCHAR2(20),  
 9   pincode     NUMBER(6)  
10  );
```

Table created.

```
SQL> CREATE TABLE customer_phone(  
 2   id          NUMBER(5),  
 3   phone       NUMBER(10)  
 4  );
```

Table created.

```
SQL> CREATE TABLE staff(  
 2   name        VARCHAR2(20),  
 3   id          number(5),  
 4   salary      number(5),  
 5   age         number(5)  
 6  );
```

Table created.

```
SQL> CREATE TABLE menu(  
 2  category_name      varchar2(30),  
 3  category_id        number(5),  
 4  food_name          varchar2(30),  
 5  food_quantity      number(5),  
 6  cost               number(5,2)  
 7 );
```

Table created.

```
SQL> CREATE TABLE bill(  
 2  itemname           varchar2(20),  
 3  price              number(5,2),  
 4  quantity           number(20),  
 5  bill_id            number,  
 6  customer_id        number(5)  
 7 );
```

Table created.

## LAB 2 – DML Operations

```
SQL> INSERT INTO customer VALUES(43,'sindhuja','maram','F8','M. J. COLONY','Hyd','Telangana',500040);  
1 row created.  
SQL> INSERT INTO customer VALUES(13,'gayathri','kalthi','201','DR. CHARI','Hyd','Telangana',500044);  
1 row created.  
SQL> INSERT INTO customer VALUES(1029,'Tom','Fernandes','9-9/0','saraswathi','Tirupathi','Andhra Pradesh',517620);  
1 row created.  
SQL> INSERT INTO customer VALUES(941,'Hussai','Kishrit','G34','Lavani','Andheri','mumbai',400047);  
1 row created.  
SQL> INSERT INTO customer VALUES(113,'Guman','Dilrath','2-3/6/Q','kalis','Alipore','Kolkata',700005);  
1 row created.  
SQL> INSERT INTO customer VALUES(4410,'Sandhya','Rathi','GY41','Hanuman Galli','Ajmer','Rajasthan',305005);  
1 row created.  
SQL> INSERT INTO customer VALUES(789,'Alankritha','Elikshina','109','Jayanki','Mysore','Karnataka',570004);  
1 row created.  
SQL> INSERT INTO customer VALUES(9101,'Ishita','Bhalla','1-7-98/E','Maruthi','Erode','Tamil Nadu',638112);  
1 row created.  
SQL> commit;  
Commit complete.
```

```
SQL> INSERT INTO customer_phone VALUES (13, 9874563210) ;  
1 row created.  
SQL> INSERT INTO customer_phone VALUES (43, 2013654789) ;  
1 row created.  
SQL> INSERT INTO customer_phone VALUES (43, 5632147890) ;  
1 row created.  
SQL> commit;  
Commit complete.
```

```
SQL> INSERT INTO staff VALUES (' kittu', 1031, 4000, 26) ;  
1 row created.  
SQL> INSERT INTO staff VALUES (' Gamya', 2904, 4000, 33) ;  
1 row created.  
SQL> INSERT INTO staff VALUES (' kiara', 1927, 4000, 19) ;  
1 row created.  
SQL> INSERT INTO staff VALUES (' Buntu', 1722, 4000, 22) ;  
1 row created.  
SQL> INSERT INTO staff VALUES (' Zahir', 1198, 4000, 31) ;  
1 row created.  
SQL> commit;  
Commit complete.
```



CAFETERIA MANAGEMENT SYSTEM

---

```
SQL> INSERT INTO menu values('coffee',1113,'cold coffee',150,30);
1 row created.
SQL> INSERT INTO menu values('Fast Food',4301,'veg pizza',100,30);
1 row created.
SQL> INSERT INTO menu VALUES('Dessert',1320,'chocolate mouse',40,58.88);
1 row created.
SQL> INSERT INTO menu VALUES('coffee',1945,'Americano',118,50.95);
1 row created.
SQL> INSERT INTO menu VALUES('Dessert',1229,'Mango icecream',78,30.56);
1 row created.
SQL> INSERT INTO menu VALUES('Fast Food',9207,'chicken burger',100,70.79);
1 row created.
SQL> INSERT INTO menu VALUES('fast food',501,'french fries',104,55.21);
1 row created.
SQL> commit;
Commit complete.
```

```
SQL> INSERT INTO bill VALUES('Americano',50.95,3,1343,43);
1 row created.
SQL> INSERT INTO bill VALUES('french fries',55.21,5,891,13);
1 row created.
SQL> INSERT INTO bill VALUES('Mango icecream',30.56,2,7104,1029);
1 row created.
SQL> INSERT INTO bill VALUES('chocolate mouse',58.88,7,1610,789);
1 row created.
SQL> INSERT INTO bill VALUES('chicken burger',70.79,1,1002,113);
1 row created.
SQL> commit;
Commit complete.
```

**LAB 3 – Integrity Constraints**

```
SQL> ALTER TABLE customer ADD CONSTRAINT pk_id PRIMARY KEY(id);
```

```
Table altered.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> desc customer;
```

Name	Null?	Type
ID	NOT NULL	NUMBER(5)
FIRSTNAME		VARCHAR2(15)
LASTNAME		VARCHAR2(15)
HOUSENO		VARCHAR2(30)
STREET		VARCHAR2(20)
CITY		VARCHAR2(20)
STATE		VARCHAR2(20)
PINCODE		NUMBER(6)

```
SQL> ALTER TABLE customer MODIFY firstname VARCHAR2(15) NOT NULL;
```

```
Table altered.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> ALTER TABLE customer_phone ADD CONSTRAINT ck_phone CHECK(LENGTH(phone) = 10);
```

```
Table altered.
```

## CAFETERIA MANAGEMENT SYSTEM

```
SQL> ALTER TABLE staff ADD CONSTRAINT pk_stid PRIMARY KEY(id);
```

```
Table altered.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> desc staff;
```

Name	Null?	Type
NAME		VARCHAR2(20)
ID	NOT NULL	NUMBER(5)
SALARY		NUMBER(5)
AGE		NUMBER(5)

```
SQL> ALTER TABLE staff MODIFY name VARCHAR2(20) NOT NULL;
```

```
Table altered.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> desc staff;
```

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(20)
ID	NOT NULL	NUMBER(5)
SALARY		NUMBER(5)
AGE		NUMBER(5)

```
SQL> ALTER TABLE bill ADD CONSTRAINT pk_billid PRIMARY KEY(bill_id);
```

```
Table altered.
```

```
SQL> ALTER TABLE bill MODIFY itemname VARCHAR2(20) NOT NULL;
```

```
Table altered.
```

```
SQL> ALTER TABLE bill MODIFY price NUMBER(5,2) NOT NULL;
```

```
Table altered.
```

```
SQL> ALTER TABLE bill MODIFY quantity NUMBER(20) NOT NULL;
```

```
Table altered.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> desc bill;
```

Name	Null?	Type
ITEMNAME	NOT NULL	VARCHAR2 (20)
PRICE	NOT NULL	NUMBER (5, 2)
QUANTITY	NOT NULL	NUMBER (20)
BILL_ID	NOT NULL	NUMBER
CUSTOMER_ID		NUMBER (5)

```
SQL> ALTER TABLE menu MODIFY cost NUMBER(5,2) NOT NULL;
```

```
Table altered.
```

```
SQL> commit;
```

```
SQL> ALTER TABLE menu ADD CONSTRAINT pk_cat_id_name PRIMARY KEY(food_name, category_id, category_name);
```

```
Table altered.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> desc menu;
```

Name	Null?	Type
CATEGORY_NAME	NOT NULL	VARCHAR2 (30)
CATEGORY_ID	NOT NULL	NUMBER (5)
FOOD_NAME	NOT NULL	VARCHAR2 (30)
FOOD_QUANTITY		NUMBER (5)
COST	NOT NULL	NUMBER (5, 2)

**LAB 4 – InBuilt Functions**

```
SQL> SELECT UPPER(FIRSTNAME) FROM customer where id = 13;
```

```
UPPER(FIRSTNAME)
```

```
-----  
GAYATHRI
```

```
SQL> select length(LASTNAME) from customer where id = 43;
```

```
LENGTH(LASTNAME)
```

```
-----  
5
```

```
SQL> SELECT CONCAT(LASTNAME, FIRSTNAME) FROM customer WHERE id = 13;
```

```
CONCAT(LASTNAME, FIRSTNAME)
```

```
-----  
kalthigayathri
```

```
SQL> SELECT SUBSTR(CONCAT(LASTNAME, FIRSTNAME), 13, 22) FROM customer WHERE ID = 13;
```

```
SUBSTR(CONCAT(LASTNAME, FIRSTNAME), 13, 22)
```

```
-----  
ri
```

```
SQL> select replace(food_name, 'izz', 'itt') from menu where category_id=4301;
```

```
REPLACE(FOOD_NAME, ' IZZ', ' ITT')
```

```
-----  
veg pitta
```

```
SQL>
```

**LAB 5 – SQL Operators, queries**

```
SQL> SELECT id "Customer ID", HouseNO "Address" FROM customer;
```

Customer ID	Address
43	F8
13	201
1029	9-9/0
941	G34
113	2-3/6/Q
4410	GY41
789	109
9101	1-7-98/E

8 rows selected.

```
SQL> select food_name, cost from menu where food_quantity=100
2 union
3 select food_name, cost from menu where food_quantity=118;
```

FOOD_NAME	COST
Americano	50.95
chicken burger	70.79
veg pizza	30

```
SQL> select category_name, food_name, cost from menu
2 minus
3 select category_name, food_name, cost from menu where cost=70.79;
```

CATEGORY_NAME	FOOD_NAME	COST
Dessert	Mango icecream	30.56
Dessert	chocolate mouse	58.88
Fast Food	veg pizza	30
coffee	Americano	50.95
coffee	cold coffee	30
fast food	french fries	55.21

6 rows selected.

## CAFETERIA MANAGEMENT SYSTEM

```
SQL> SELECT *FROM staff WHERE age BETWEEN 18 AND 25;
```

NAME	ID	SALARY	AGE
kiara	1927	4000	19
Buntu	1722	4000	22

```
SQL> SELECT *FROM staff WHERE age IN(26,57);
```

NAME	ID	SALARY	AGE
kittu	1031	4000	26

```
SQL> SELECT *FROM staff WHERE age IN(26,33);
```

NAME	ID	SALARY	AGE
kittu	1031	4000	26
Gamya	2904	4000	33

```
SQL> SELECT UNIQUE name FROM staff;
```

NAME
kiara
Buntu
kittu
Gamya
Zahir

```
SQL> select S.name,S.id FROM staff S WHERE S.age> ALL(SELECT S1.age FROM staff S1 WHERE id=1927);
```

NAME	ID
Buntu	1722
kittu	1031
Zahir	1198
Gamya	2904

```
SQL> SELECT COUNT(DISTINCT name) FROM staff;
```

```
COUNT(DISTINCTNAME)
```

```
-----
5
```

```
SQL> select AVG(cost) from menu;
```

```
AVG(COST)
```

```
-----
46.6271429
```

```
SQL> select SUM(food_quantity) from menu;
```

```
SUM(FOOD_QUANTITY)
```

```
-----
690
```

```
SQL> select MAX(cost) from menu;
```

```
MAX(COST)
```

```
-----
70.79
```

```
SQL> select *from menu WHERE cost<=(select MAX(cost) from menu);
```

```
CATEGORY_NAME CATEGORY_ID FOOD_NAME
```

```
-----
```

```
FOOD_QUANTITY COST
```

```
-----
```

```
coffee 150 30 1113 cold coffee
```

```
Fast Food 100 30 4301 veg pizza
```

```
Dessert 40 58.88 1320 chocolate mouse
```



## CAFETERIA MANAGEMENT SYSTEM

```

CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
FOOD_QUANTITY          COST
-----
coffee                1945 Americano
      118          50.95

Dessert                1229 Mango icecream
      78          30.56

Fast Food              9207 chicken burger
      100          70.79

CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
FOOD_QUANTITY          COST
-----
fast food              501 french fries
      104          55.21

7 rows selected.

SQL> select *from menu where food_quantity<(select MAX(M.food_quantity) from menu M where M.cost=30);

CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
FOOD_QUANTITY          COST
-----
Fast Food              4301 veg pizza
      100          30

Dessert                1320 chocolate mouse
      40          58.88

coffee                1945 Americano
      118          50.95

```

```

CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
FOOD_QUANTITY          COST
-----
Dessert                1229 Mango icecream
      78          30.56

Fast Food              9207 chicken burger
      100          70.79

fast food              501 french fries
      104          55.21

6 rows selected.

```

**LAB 6 – JOINS**

```
SQL> SELECT C.id,C.firstname,B.bill_id,B.price FROM customer C INNER JOIN bill B ON C.id = B.customer_id;
```

ID	FIRSTNAME	BILL_ID	PRICE
13	gayathri	891	55.21
43	sindhuja	1343	50.95
113	Guman	1002	70.79
789	Alankritha	1610	58.88
1029	Tom	7104	30.56

```
SQL> SELECT C.id,C.firstname,B.bill_id,B.price FROM customer C LEFT OUTER JOIN bill B ON C.id = B.customer_id;
```

ID	FIRSTNAME	BILL_ID	PRICE
13	gayathri	891	55.21
43	sindhuja	1343	50.95
113	Guman	1002	70.79
789	Alankritha	1610	58.88
941	Hussai		
1029	Tom	7104	30.56
4410	Sandhya		
9101	Ishita		

8 rows selected.

```
SQL> SELECT C.id,C.firstname,B.bill_id,B.price FROM customer C RIGHT OUTER JOIN bill B ON C.id = B.customer_id;
```

ID	FIRSTNAME	BILL_ID	PRICE
43	sindhuja	1343	50.95
13	gayathri	891	55.21
1029	Tom	7104	30.56
113	Guman	1002	70.79
789	Alankritha	1610	58.88

## CAFETERIA MANAGEMENT SYSTEM

```
SQL> SELECT C.id,C.firstname,B.bill_id,B.price FROM customer C NATURAL JOIN bill B ;
```

ID	FIRSTNAME	BILL_ID	PRICE
43	sindhuja	1343	50.95
13	gayathri	1343	50.95
1029	Tom	1343	50.95
941	Hussai	1343	50.95
113	Guman	1343	50.95
4410	Sandhya	1343	50.95
789	Alankritha	1343	50.95
9101	Ishita	1343	50.95
43	sindhuja	891	55.21
13	gayathri	891	55.21
1029	Tom	891	55.21
941	Hussai	891	55.21
113	Guman	891	55.21
4410	Sandhya	891	55.21
789	Alankritha	891	55.21
9101	Ishita	891	55.21
43	sindhuja	7104	30.56
13	gayathri	7104	30.56
1029	Tom	7104	30.56
941	Hussai	7104	30.56
113	Guman	7104	30.56
4410	Sandhya	7104	30.56
789	Alankritha	7104	30.56
9101	Ishita	7104	30.56
43	sindhuja	1610	58.88
13	gayathri	1610	58.88
1029	Tom	1610	58.88
941	Hussai	1610	58.88
113	Guman	1610	58.88
4410	Sandhya	1610	58.88

## CAFETERIA MANAGEMENT SYSTEM

```

      43 sindhuja          1610      58.88
      13 gayathri         1610      58.88
1029 Tom                 1610      58.88
      941 Hussai          1610      58.88
      113 Guman           1610      58.88
      4410 Sandhya        1610      58.88
      789 Alankritha      1610      58.88
      9101 Ishita         1610      58.88
      43 sindhuja          1002      70.79

      ID FIRSTNAME          BILL_ID    PRICE
-----
      13 gayathri          1002      70.79
1029 Tom                 1002      70.79
      941 Hussai          1002      70.79
      113 Guman           1002      70.79
      4410 Sandhya        1002      70.79
      789 Alankritha      1002      70.79
      9101 Ishita         1002      70.79

40 rows selected.

SQL> SELECT C.id,C.firstname,B.bill_id,B.price FROM customer C FULL OUTER JOIN bill B ON C.id = B.customer_id ;

      ID FIRSTNAME          BILL_ID    PRICE
-----
      43 sindhuja          1343      50.95
      13 gayathri          891       55.21
1029 Tom                 7104      30.56
      941 Hussai          1002      70.79
      113 Guman           1002      70.79
      4410 Sandhya        1610      58.88
      789 Alankritha      1610      58.88
      9101 Ishita         1610      58.88

8 rows selected.

```

```

SQL> SELECT B.bill_id,AVG(B.price) FROM bill B WHERE B.price>=28.9 GROUP BY B.bill_id;

      BILL_ID  AVG(B.PRICE)
-----
      891       55.21
      1002       70.79
      1343       50.95
      1610       58.88
      7104       30.56

```

```
SQL> SELECT *FROM bill ORDER BY price;
```

ITEMNAME	PRICE	QUANTITY	BILL_ID	CUSTOMER_ID
Mango icecream	30.56	2	7104	1029
Americano	50.95	3	1343	43
french fries	55.21	5	891	13
chocolate mouse	58.88	7	1610	789
chicken burger	70.79	1	1002	113

**LAB 7 – Views**

```
SQL> CREATE VIEW food_menu as select *from menu;
```

```
View created.
```

```
SQL> select *from food_menu;
```

CATEGORY_NAME			CATEGORY_ID	FOOD_NAME
-----				
FOOD_QUANTITY		COST		
-----				
coffee			1113	cold coffee
150		30		
Fast Food			4301	veg pizza
100		30		
Dessert			1320	chocolate mouse
40		58.88		
CATEGORY_NAME			CATEGORY_ID	FOOD_NAME
-----				
FOOD_QUANTITY		COST		
-----				
coffee			1945	Americano
118		50.95		
Dessert			1229	Mango icecream
78		30.56		
Fast Food			9207	chicken burger
100		70.79		
CATEGORY_NAME			CATEGORY_ID	FOOD_NAME
-----				
FOOD_QUANTITY		COST		
-----				
fast food			501	french fries
104		55.21		

```

101      30.21
7 rows selected.
SQL> insert into food_menu values('coffee',1521,'Espresso',101,40.43);
1 row created.
SQL> select *from food_menu;

```

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
FOOD_QUANTITY	COST		
coffee		1113	cold coffee
150	30		
Fast Food		4301	veg pizza
100	30		
Dessert		1320	chocolate mouse
40	58.88		

CATEGORY_NAME		CATEGORY_ID	FOOD_NAME
FOOD_QUANTITY	COST		
coffee		1945	Americano
118	50.95		
Dessert		1229	Mango icecream
78	30.56		
Fast Food		9207	chicken burger
100	70.79		

## CAFETERIA MANAGEMENT SYSTEM

```
Fast Food          9207 chicken burger
      100          70.79
```

```
CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
```

```
FOOD_QUANTITY          COST
-----
```

```
fast food          501 french fries
      104          55.21
```

```
coffee          1521 Espresso
      101          40.43
```

```
8 rows selected.
```

```
SQL> select *from menu;
```

```
CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
```

```
FOOD_QUANTITY          COST
-----
```

```
coffee          1113 cold coffee
      150          30
```

```
Fast Food          4301 veg pizza
      100          30
```

```
Dessert          1320 chocolate mouse
      40          58.88
```

```
CATEGORY_NAME          CATEGORY_ID FOOD_NAME
-----
```

```
FOOD_QUANTITY          COST
-----
```

```
coffee          1945 Americano
      118          50.95
```

```

coffee          118      50.95      1945 Americano
Dessert          78      30.56      1229 Mango icecream
Fast Food       100      70.79      9207 chicken burger

```

```

CATEGORY_NAME          CATEGORY_ID  FOOD_NAME
-----

```

```

FOOD_QUANTITY      COST
-----

```

```

fast food          501 french fries
      104      55.21
coffee           1521 Espresso
      101      40.43

```

```

8 rows selected.

```

```

SQL> create view cust as select *from customer WITH READ ONLY;

```

```

View created.

```

```

SQL> update cust set lastname='Maris' where id = 113;

```

```

update cust set lastname='Maris' where id = 113

```

```

      *

```

```

ERROR at line 1:

```

```

ORA-42399: cannot perform a DML operation on a read-only view

```

```

SQL> create view ordered as select c.id,c.firstname,b.bill_id,b.price from customer c, bill b where c.id = b.customer_id;

```

```

View created.

```

```

SQL> select *from ordered;

```

```

      ID  FIRSTNAME      BILL_ID      PRICE
-----
      13  gayathri          891        55.21
      43  sindhuja          1343        50.95
     113  Guman            1002        70.79
     789  Alankritha        1610        58.88
    1029  Tom              7104        30.56

```



**LAB-8&9 plsql, cursors, procedures**

```
SQL> Declare
  2  Cursor staff_cur
  3  is
  4  select name,id,age from staff;
  5  begin
  6  for rec_staff in staff_cur
  7  loop
  8  dbms_output.put_line(staff_cur%rowcount||','||rec_staff.name||','||rec_staff.id||','||rec_staff.age);
  9  end loop;
 10  end;
 11  /
1,Bunty,1722,29
2,Bittu,1986,27
3,kitty,1031,27
4,Gamya,2904,33

PL/SQL procedure successfully completed.

SQL>
```

```
SQL> select *from bill;

ITEMNAME          PRICE    QUANTITY    BILL_ID  CUSTOMER_ID
-----
tiramisu           58.88         7         1610         789
cheesecake         300           2         7104        1029
veg pizza          299.9         5         4590         13

SQL> begin update bill
  2  set price=100
  3  where itemname='tiramisu';
  4  if SQL%FOUND
  5  THEN
  6  dbms_output.put_line(SQL%ROWCOUNT|| 'ROWS UPDATED');
  7  ELSE
  8  dbms_output.put_line('itemname not found');
  9  end if;
 10  end;
 11  /
1ROWS UPDATED

PL/SQL procedure successfully completed.

SQL> select *from bill;

ITEMNAME          PRICE    QUANTITY    BILL_ID  CUSTOMER_ID
-----
tiramisu           100           7         1610         789
cheesecake         300           2         7104        1029
veg pizza          299.9         5         4590         13

SQL>
```

```
SQL> DECLARE
  2  staffID number;
  3  PROCEDURE staffUpdate(s in number) IS
  4  BEGIN
  5  update staff S
  6  set S.salary=S.salary+500
  7  where S.age>25 and S.id=s;
  8  IF SQL%FOUND
  9  THEN
 10  dbms_output.put_line('rows updated');
 11  ELSE
 12  dbms_output.put_line('age not found');
 13  END IF;
 14  END;
 15  BEGIN
 16  staffID:=&staffID;
 17  staffUpdate(staffID);
 18  END;
 19  /
```

Enter value for staffid: 1031

old 16: staffID:=&staffID;

new 16: staffID:=1031;

rows updated

PL/SQL procedure successfully completed.

```
SQL> /
```

Enter value for staffid: 1896

old 16: staffID:=&staffID;

new 16: staffID:=1896;

age not found

PL/SQL procedure successfully completed.

```
SQL> select *from staff;
```

NAME	ID	SALARY	AGE
Bunty	1722	4000	29
Bittu	1986	4000	22
kitty	1031	4501	27
Gamya	2904	4000	33

**LAB-10 Triggers**

```
SQL> set serveroutput on;
SQL> CREATE OR REPLACE TRIGGER customer_update
  2 AFTER UPDATE ON customer
  3 begin
  4 dbms_output.put_line('Customer details updated successfully');
  5 end;
  6 /
```

Trigger created.

```
SQL> update customer set firstname='Sindhujha' where id=43;
Customer details updated successfully
```

1 row updated.

SQL>

```
SQL> CREATE OR REPLACE TRIGGER restrictcost47
  2 BEFORE DELETE ON menu
  3 FOR EACH ROW
  4 begin
  5 IF :old.cost > 47 THEN
  6   raise_application_error(-20333,'Insufficient Privileges');
  7 END IF;
  8 end;
  9 /
```

Trigger created.

```
SQL> delete from menu where cost>47;
delete from menu where cost>47
      *
```

ERROR at line 1:

ORA-20333: Insufficient Privileges

ORA-06512: at "IT19737043.RESTRICTCOST47", line 3

ORA-04088: error during execution of trigger 'IT19737043.RESTRICTCOST47'

SQL>

```
SQL> CREATE OR REPLACE TRIGGER nolowercase
  2 BEFORE INSERT OR UPDATE on staff
  3 FOR EACH ROW
  4 begin
  5 IF trim(:new.name)!=upper(trim(:new.name))
  6 then
  7 raise_application_error(-20013,'Lower case is not allowed');
  8 end if;
  9 end;
 10 /
```

Trigger created.

```
SQL> Insert into staff values('hari',1214,4000,31);
Insert into staff values('hari',1214,4000,31)
      *
ERROR at line 1:
ORA-20013: Lower case is not allowed
ORA-06512: at "IT19737043.NOLOWERCASE", line 4
ORA-04088: error during execution of trigger 'IT19737043.NOLOWERCASE'
```

```
SQL>
```

**THANK YOU**