

UCS-1412 DATABASE LAB

MINI PROJECT

WHOLESALE MANAGEMENT SYSTEM

Connectivity Code:

```
public void connect()
{try{
    Class.forName("oracle.jdbc.OracleDriver");
    con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
    JOptionPane.showMessageDialog(this, "Connected to oracle database!!");
}
catch(SQLException ex){
    Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
    JOptionPane.showMessageDialog(this,ex.getMessage());
}

catch(ClassNotFoundException | NullPointerException ex){
    Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
    JOptionPane.showMessageDialog(this, ex.getMessage());
}
}
```

INITIALIZING THE VALUES:

```
private String selectevalue;
Connection con;
PreparedStatement ps;
DefaultTableModel df;
```

VIEWING OF CONTENTS IN TABLE:

FOR CUSTOMER TABLE:

```

public void table()
{
    int s;
    try {
        ps = con.prepareStatement("select * from vendor");
        ResultSet rs = ps.executeQuery();
        ResultSetMetaData rd = rs.getMetaData();
        s = rd.getColumnCount();
        df = (DefaultTableModel)jTable1.getModel();
        df.setRowCount(0);
        while(rs.next())
        {
            Vector v2 = new Vector();
            for(int i=1; i<=s; i++)
            {
                v2.add(rs.getString("vendor_id"));
                v2.add(rs.getString("vendor_name"));
                v2.add(rs.getString("phoneno"));
                v2.add(rs.getString("address"));
                v2.add(rs.getString("email"));
            }
            df.addRow(v2);
        }

    }
    catch(SQLException ex){
        Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
    }

    catch(NullPointerException ex){
        Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

```

FOR PROFIT TABLE:

```

public void upload()
{
    try {

```

```

con = javaconnect.connect();
String h = "select * from pro";
ps = con.prepareStatement(h);
ResultSet rs = ps.executeQuery();

```

```

df = (DefaultTableModel)profit.getModel();
df.setRowCount(0);
while(rs.next())
{
    Vector v2 = new Vector();
    v2.add(rs.getDate(1));
    v2.add(rs.getDate(2));
    v2.add(rs.getInt(3));
    df.addRow(v2);
}

```

```

    } catch (SQLException ex) {
        Logger.getLogger(whole.customer.class.getName()).log(Level.SEVERE, null,
ex);
    }

}

```

FOR STOCK TABLE:

```

public void showTable() throws SQLException{
    Statement stmt = null;
    Connection con = null;
    con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
    stmt=con.createStatement();
    DefaultTableModel model = (DefaultTableModel)jTable2.getModel();
    model.setRowCount(0);
    ResultSet rs=null;
    rs=stmt.executeQuery("select stock_id,st_name,unit_price,exp_date from stocks");
    try {

```

```

        while(rs.next()){
            model.addRow(new
Object[]{rs.getString("stock_id"),rs.getString("st_name"),rs.getInt("unit_price"),rs.getDat
e("exp_date")});
        }
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

FOR EXPIRED TABLE:

```

try {
    stmt=con.createStatement();
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
ResultSet rs=null;
try {
    rs=stmt.executeQuery("select stock_id,st_name,unit_price,exp_date from stocks
where exp_date<=(select sysdate from dual)");
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
try {
    while(rs.next()){
        model.addRow(new
Object[]{rs.getString("stock_id"),rs.getString("st_name"),rs.getInt("unit_price"),rs.getDat
e("exp_date")});
    }
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
try {
    rs=stmt.executeQuery("select count(stock_id) from stocks where
exp_date<=(select sysdate from dual)");
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
try {

```

```

        while(rs.next()){
            jTextField5.setText(rs.getString("count(stock_id)"));
        }
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

LOGIN CODE:(ADMIN)

```

try {
    // TODO add your handling code here:
    Connection con;
    con = javaconnect.connect();
    String sql = "select * from login where username=? and password =?";
    ps = con.prepareStatement(sql);
    ps.setString(1, username.getText());
    ps.setString(2, password.getText());

    String user=username.getText();
    String pass=password.getText();

    if(user.equals("admin") && pass.equals("admin") )
    {
        //setVisible(false);
        // new customer().setVisible(true);
        JOptionPane.showMessageDialog(null, "Login Succesfull!");
        switchPanels(jbuyerpanel);
    }

} catch (SQLException ex) {
    Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
}

```

EMAIL VERIFICATION CODE:

```

if (!(Pattern.matches("[a-zA-Z0-9]+@[1]+[a-zA-Z0-9]+[.]{1}+[a-zA-Z0-9]+$",
jemailtxt.getText()))
{

```

```

        JOptionPane.showMessageDialog(null, "Please enter a valid email", "Error",
JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(null, "The email is valid", "Good!",
JOptionPane.INFORMATION_MESSAGE);

    }
}

```

CUSTOMER INSERT:

```

try {
    String vname = jvendornametxt.getText();
    String phone = jphonetxt.getText();
    String address =jaddresstxt.getText();
    String email =jemailtxt.getText();

    ps = con.prepareStatement("insert into
vendor(vendor_name,phoneno,address,email)values(?,?,?,?)");
    ps.setString(1, vname);
    ps.setString(2, phone);
    ps.setString(3, address);
    ps.setString(4, email);
    ps.executeUpdate();
    JOptionPane.showMessageDialog(this,"Vendor added succesfully");
    table();

    clearText();
} catch (SQLException ex) {
    Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
}

```

CUSTOMER UPDATE:

```

df = (DefaultTableModel)jTable1.getModel();
int selected = jTable1.getSelectedRow();
int id = Integer.parseInt(df.getValueAt(selected, 0).toString());

String vname = jvendornametxt.getText();
String phone = jphonetxt.getText();
String address = jaddresstxt.getText();
String email = jemailtxt.getText();
try {

    ps = con.prepareStatement("update vendor set vendor_name = ?,phoneno =
?,address = ?,email = ?");
    ps.setString(1, vname);
    ps.setString(2, phone);
    ps.setString(3, address);
    ps.setString(4, email);
    ps.executeUpdate();
    JOptionPane.showMessageDialog(this,"Vendor updated succesfully");

    clearText();
    table();
    jaddbtm.setEnabled(true);

} catch (SQLException ex) {
    Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
}

```

CUSTOMER DELETE:

```

df = (DefaultTableModel)jTable1.getModel();
int selected = jTable1.getSelectedRow();
int id = Integer.parseInt(df.getValueAt(selected, 0).toString());
try{
    ps = con.prepareStatement("delete from vendor where id = ?");
    ps.setInt(1,id);
    ps.executeUpdate();
    JOptionPane.showMessageDialog(this, "Vendor deleted ");
    clearText();
}

```

```

        table();
        jaddbtm.setEnabled(true);
    } catch (SQLException ex) {
        Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

STOCK INSERT:

```

Statement stmt = null;
Connection con = null;
int rows_aff=0;
try {
    con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        stmt=con.createStatement();
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }

    try {
        rows_aff = stmt.executeUpdate("insert into stocks
values('"+stkid_in.getText()+"','"+stknm_in.getText()+"','"+up_in.getText()+"','"+jFormatte
dTextField1.getText()+"')");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    if (rows_aff==1){
        try {
            showTable();
        } catch (SQLException ex) {
            Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
        }
        Icon icon=new javax.swing.ImageIcon(getClass().getResource("/icons/bsf2.png"));
        JOptionPane.showMessageDialog(this,"values successfully inserted
!!","SUCCESS !",JOptionPane.INFORMATION_MESSAGE,icon);
    }
}

```



```

    }
    else{
        Icon icon=new
        javax.swing.ImageIcon(getClass().getResource("/icons/warning_blue.png"));
        JOptionPane.showMessageDialog(this,"Provided id not found !!!","ERROR
        !",JOptionPane.INFORMATION_MESSAGE,icon);
    }

```

STOCK UPDATE:

```

Statement stmt = null;
    Connection con = null;
    int rows_aff=0;
    try {
        con =
        DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
        ystem","harish24");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        stmt=con.createStatement();
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        rows_aff = stmt.executeUpdate("update stocks set
        st_name='"+stknm_in.getText()+"',unit_price='"+up_in.getText()+"',exp_date='"+jFormatt
        edTextField1.getText()+"' where stock_id='"+stkid_in.getText()+"'");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    if (rows_aff==1){
        Icon icon=new
        javax.swing.ImageIcon(getClass().getResource("/icons/bsf2.png"));
        JOptionPane.showMessageDialog(this,"values successfully inserted
        !!!","SUCCESS !",JOptionPane.INFORMATION_MESSAGE,icon);
        try {
            showTable();
        } catch (SQLException ex) {

```

```

        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
}
else{
    Icon icon=new
javax.swing.ImageIcon(getClass().getResource("/icons/warning_blue.png"));
    JOptionPane.showMessageDialog(this,"Provided id not found !!!","ERROR
!",JOptionPane.INFORMATION_MESSAGE,icon);
}

```

STOCK REMOVE:

```

Statement stmt = null;
Connection con = null;
int rows_aff=0;
try {
    con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
try {
    stmt=con.createStatement();
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
try {
    rows_aff = stmt.executeUpdate("delete from stocks where stock_id
='"+stkid_in.getText()+"'");
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
if (rows_aff==1){
    Icon icon=new
javax.swing.ImageIcon(getClass().getResource("/icons/bsf2.png"));
    JOptionPane.showMessageDialog(this,"values successfully inserted
!!!","SUCCESS !",JOptionPane.INFORMATION_MESSAGE,icon);
    try {

```

```

        showTable();
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
}
else{
    Icon icon=new
javax.swing.ImageIcon(getClass().getResource("/icons/warning_blue.png"));
    JOptionPane.showMessageDialog(this,"Provided id not found !!!","ERROR
!",JOptionPane.INFORMATION_MESSAGE,icon);
}
}

private void insertbtnActionPerformed(java.awt.event.ActionEvent evt) {
    Statement stmt = null;
    Connection con = null;
    int rows_aff=0;
    try {
        con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        stmt=con.createStatement();
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }

    try {
        rows_aff = stmt.executeUpdate("insert into stocks
values('"+stkid_in.getText()+"','"+stknm_in.getText()+"','"+up_in.getText()+"','"+jFormatte
dTextField1.getText()+"')");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    if (rows_aff==1){
        try {

```

```

        showTable();
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    Icon icon=new javax.swing.ImageIcon(getClass().getResource("/icons/bsf2.png"));
    JOptionPane.showMessageDialog(this,"values successfully inserted
!!","SUCCESS !",JOptionPane.INFORMATION_MESSAGE,icon);
}
else{
    Icon icon=new
javax.swing.ImageIcon(getClass().getResource("/icons/warning_blue.png"));
    JOptionPane.showMessageDialog(this,"Provided id not found !!!","ERROR
!",JOptionPane.INFORMATION_MESSAGE,icon);
}
}
}

```

EXPIRED STOCKS:(SHOW)

```

DefaultTableModel model = (DefaultTableModel)jTable3.getModel();
model.setRowCount(0);
Statement stmt = null;
Connection con = null;
try {
    con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
try {
    stmt=con.createStatement();
} catch (SQLException ex) {
    Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
}
ResultSet rs=null;
try {
    rs=stmt.executeQuery("select stock_id,st_name,unit_price,exp_date from stocks
where exp_date<=(select sysdate from dual)");

```

```

    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        while(rs.next()){
            model.addRow(new
Object[]{rs.getString("stock_id"),rs.getString("st_name"),rs.getInt("unit_price"),rs.getDat
e("exp_date")});
        }
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        rs=stmt.executeQuery("select count(stock_id) from stocks where
exp_date<=(select sysdate from dual)");
        } catch (SQLException ex) {
            Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
        }
    try {
        while(rs.next()){
            jTextField5.setText(rs.getString("count(stock_id)"));
        }
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

EXPIRED STOCKS:(REMOVE)

```

Connection con = null;
CallableStatement stmt2 = null;
try {
    con =
DriverManager.getConnection("jdbc:oracle:thin:@//LAPTOP-RG4AKAFH:1521/orcl2","s
ystem","harish24");
    } catch (SQLException ex) {
        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        stmt2=con.prepareCall("call delete_exp()");
    } catch (SQLException ex) {

```

```

        Logger.getLogger(baseframe.class.getName()).log(Level.SEVERE, null, ex);
    }
    Icon icon=new javax.swing.ImageIcon(getClass().getResource("/icons/bsf2.png"));
    JOptionPane.showMessageDialog(this,"values successfully removed
!!","REMOVED !",JOptionPane2.INFORMATION_MESSAGE,icon);
}

```

MONTHLY PROFIT:(GENERATE)

```

try {
    Connection con1 = javaconnect.connect();
    String sql = "call w_orders(?,?)";
    ps = con.prepareStatement(sql);
    ps.setString(1, startdate.getText());
    ps.setString(2, enddate.getText());
    ResultSet rs = ps.executeQuery();
    if(rs!=null)
    {

        JOptionPane.showMessageDialog(null, "Profit calculated");
        upload();
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Error");

    }

} catch (SQLException ex) {
    Logger.getLogger(customer.class.getName()).log(Level.SEVERE, null, ex);
}

}

```

BACKEND:ORACLE 10g

SQL CODE:

```
SQL> @ D:miniproject.sql;
SQL> set echo on;
SQL> set linesize 150;
SQL> set pagesize 200;
SQL>
SQL>
SQL> rem:CUSTOMER PANEL:
SQL> rem:dropping table customer
SQL> drop table vendor;
```

Table dropped.

```
SQL>
SQL> rem:creating table of customer
SQL> create table vendor(vendor_id varchar2(7) constraint venid_pk primary key,
2          vendor_name char(100),
3          phoneno number(10),
4          address varchar2(20),
5          email varchar2(25));
```

Table created.

```
SQL>
SQL> rem:SEQUENCE CREATION
SQL>
SQL> DROP SEQUENCE seq_vendor;
```

Sequence dropped.

```
SQL>
SQL> create sequence seq_vendor
2 MINVALUE 1
3 START WITH 001
4 INCREMENT BY 1
5 CACHE 10;
```

Sequence created.

```
SQL>
```

```

SQL> rem:TRIGGERS:
SQL> rem:creating triigers
SQL> CREATE OR REPLACE TRIGGER trig_id
  2 BEFORE INSERT ON vendor
  3 FOR EACH ROW
  4 WHEN (new.vendor_id IS NULL)
  5 BEGIN
  6 SELECT seq_vendor.NEXTVAL
  7 INTO :new.vendor_id
  8 FROM dual;
  9 END;
10 /

```

Trigger created.

```

SQL>
SQL> rem:STOCKS PANEL:
SQL> rem:Cretaing table for stocks
SQL>
SQL> drop table stocks;

```

Table dropped.

```

SQL>
SQL> create table stocks(stock_id varchar2(5) constraint stk_pk primary key,
  2 st_name varchar2(15),
  3 unit_price number(4),exp_date date);

```

Table created.

```

SQL>
SQL> REM:PROCEDURE:
SQL> rem:procedure for expired stocks
SQL>
SQL> create or replace procedure delete_exp as
  2 cursor c1 is
  3 select stock_id from stocks where exp_date<=(select sysdate from dual);
  4 stkid stocks.stock_id%type;
  5 begin

```



```

6 open c1;
7 loop
8 fetch c1 into stkid;
9 if c1%notfound then
10 exit;
11 end if;
12 delete from stocks where stock_id=stkid;
13 end loop;
14 close c1;
15 end;
16 /

```

Procedure created.

SQL>

SQL> rem:**MONTHLY PROFIT PANEL:**

SQL>

SQL> drop table wholeorders;

Table dropped.

SQL>

SQL> rem:-- CREATION OF WHOLEORDERS TABLE (cus_id should be declared as foreign rem:key as per the s

```

SQL> create table wholeorders( ord_id varchar2(5) constraint order_pk primary key,
2          cus_id varchar2(5),
3          ord_date date,
4          amount number(10) );

```

Table created.

SQL>

SQL> rem:-- VALUES FOR WHOLEORDERS TABLE

SQL>

SQL> insert into wholeorders values('OR001','C001','22-FEB-2020', 2500);

1 row created.

SQL> insert into wholeorders values('OR002','C002','25-FEB-2020', 1500);

1 row created.

```
SQL> insert into wholeorders values('OR003','C003','27-FEB-2020', 1000);
```

1 row created.

```
SQL>
```

```
SQL> insert into wholeorders values('OR004','C004','10-MAR-2020', 25000);
```

1 row created.

```
SQL> insert into wholeorders values('OR005','C001','18-MAR-2020', 500);
```

1 row created.

```
SQL> insert into wholeorders values('OR006','C005','20-MAR-2020', 1500);
```

1 row created.

```
SQL>
```

```
SQL> insert into wholeorders values('OR007','C002','20-APR-2020', 100);
```

1 row created.

```
SQL> insert into wholeorders values('OR008','C006','25-APR-2020', 500);
```

1 row created.

```
SQL> insert into wholeorders values('OR009','C007','27-APR-2020', 100);
```

1 row created.

```
SQL>
```

```
SQL> rem:-- CREATION OF PROFIT TABLE
```

```
SQL>
```

```
SQL> drop table pro;
```

Table dropped.

SQL>

```
SQL> create table pro(startdate date,  
2          enddate date,  
3          amount number(5));
```

Table created.

SQL>

SQL>

SQL> rem:-- PROCEDURE FOR CALCULATING MONTHLY INCOME

SQL>

```
SQL> CREATE or REPLACE procedure w_orders(startdate IN varchar2,enddate IN  
varchar2) IS  
2 total int;  
3 BEGIN  
4 total:=0;  
5 select sum(amount) INTO total from wholeorders where ord_date between  
startdate and enddate;  
6 INSERT INTO pro VALUES (startdate,enddate,total);  
7 END;  
8 /
```

Procedure created.

SQL>