Dt : 5/5/2022

Ex_Application :

JDBC application to display the details of Product based on prodCode.

Program : DBCon3.java

```
package test;

import java.sql.*;

import java.util.*;;

public class DBCon3 {

          public static void main(String[] args) {

      try {

          Scanner s = new Scanner(System.in);

          System.out.println("Enter the ProdCode:");

          String pCode = s.nextLine();

          Connection con = DriverManager.getConnection

         ("jdbc:oracle:thin:@localhost:1521:xe","system","manager");

          Statement stm = con.createStatement();

          ResultSet rs = stm.executeQuery

           ("select * from Product45 where pcode='"+pCode+"'");

          if(rs.next()) {

                    System.out.println(rs.getString(1)+"\t"+

              rs.getString(2)+"\t"+rs.getFloat(3)+

              "\t"+rs.getInt(4));

          }else {
```

```
                    System.out.println("Invalid ProdCode...");
            }
            con.close();
            s.close();
        }catch(Exception e) {e.printStackTrace();}
            }
}
```

o/p:

```
Enter the ProdCode:
A222
A222    KB       1300.0 13
```

==

*imp

2.PreparedStatement:

=>PreparedStatement is an interface from java.sql package and which is used to execute normal queries with IN parameters.

=>The following are some important methods of PreparedStatement:

(i)executeQuery()

(ii)executeUpdate()

(i)executeQuery():

=>executeQuery() method is used to execute select queries.

Method Signature:

public abstract java.sql.ResultSet executeQuery() throws java.sql.SQLException;

(ii)executeUpdate():

=>executeUpdate() method is used to execute Non-Select queries.

Method Signature:

public abstract int executeUpdate() throws java.sql.SQLException;

=>we use prepareStatement() method from 'Connection' interface to create the implementation object of 'PreparedStatement' interface.

Method Signature of prepareStatement():

public abstract java.sql.PreparedStatement prepareStatement(java.sql.String)

throws java.sql.SQLException;

syntax:

PreparedStatement ps = con.prepareStatement("query-structure");

===

Prog-1 : JDBC Application to read employee details from Console and insert into DB table Employee45 using PreparedStatement

Note:

Calculate totSal based on bSal using the following formula:

totSal = bSal+HRA+DA;

HRA = 93% of bSal

DA = 63% of bSal

DB Table : Employee45(eid,ename,edesg,bsal,totsal)

```
create table Employee45(eid varchar2(10),ename varchar2(15),edesg varchar2(10),

bsal number(10),totsal number(10,2),primary key(eid));
```

Program : DBCon4.java

```
package test;

import java.util.*;

import java.sql.*;

public class DBCon4 {

  public static void main(String[] args) {

           try {

                   Scanner s = new Scanner(System.in);

                   System.out.println("Enter the EmpId:");

                   String eId = s.nextLine();

                   System.out.println("Enter the EmpName:");

                   String eName = s.nextLine();

                   System.out.println("Enter the EmpDesg:");

                   String eDesg = s.nextLine();

                   System.out.println("Enter the BSal");

                   int bSal = s.nextInt();

                   float totSal = bSal+(0.93F*bSal)+(0.63F*bSal);


                   Connection con = DriverManager.getConnection

                   ("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
```

```
        PreparedStatement ps = con.prepareStatement
                ("insert into Employee45 values(?,?,?,?,?)");//Compilation
        ps.setString(1,eId);
        ps.setString(2,eName);
        ps.setString(3,eDesg);
        ps.setInt(4, bSal);
        ps.setFloat(5, totSal);

        int k = ps.executeUpdate();//Execution process

        if(k>0) {
            System.out.println("Employee Details inserted Successfully...");
        }
        con.close();
        s.close();
    }catch(Exception e) {e.printStackTrace();}
  }
}
```

o/p:

```
Enter the EmpId:
A121
Enter the EmpName:
Raj
```

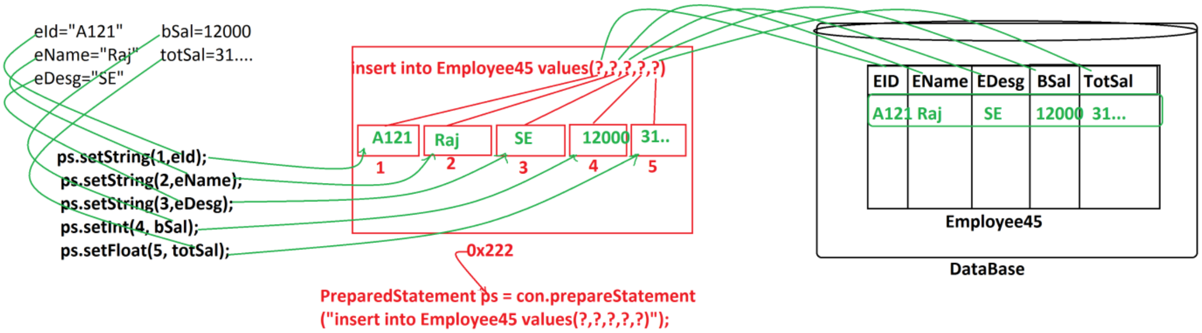
Enter the EmpDesg:

SE

Enter the BSal

12000

Employee Details inserted Successfully...



===

faq:

wt is the execution behaviour of prepareStatement() method?

=>prepareStatement() method will compile the query Structure.

=>If the compilation is successfully then prepareStatement() method will create fields equal to the parameter indexes.

Note:

=>we use setter methods to set the data to the fields based on parameter index and then executed.

===

faq:

wt is the advantage of PreparedStatement?

=>In PreparedStatement the query is compiled once and can be executed any

number of times by setting the values.

===

Assignment:

DB Table : UserReg45(uname,pword,fname,lname,addr,mid,phno)

Primary Key : uname and pword

Construct JDBC Application to perform the following operations based on User Choice:(Using PreparedStatement)

1.Register

2.Login

1.Register:

=>read user details from console and insert in to DBtable UserReg45.

2.Login:

=>read uname and pword from the console and display the following msg:

Login Successfull...

Welcome User : fName

==