Dt : 7/5/2022

DB Table : UserReg45

```
create table UserReg45(uname varchar2(15),pword varchar2(15),
fname varchar2(15),lname varchar2(15),addr varchar2(15),
mid varchar2(25),phno number(15),primary key(uname,pword));
```

Program : DBCon5.java

```
package test;
import java.sql.Connection;
import java.sql.*;
import java.util.*;
public class DBCon5 {
        public static void main(String[] args) {
      try {
         Scanner s = new Scanner(System.in);
         Connection con = DriverManager.getConnection
        ("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
         PreparedStatement ps1 = con.prepareStatement
                        ("insert into UserReg45 values(?,?,?,?,?,?,?)");
         PreparedStatement ps2 = con.prepareStatement
                        ("select * from UserReg45 where uname=? and pword=?");
         while(true) {
                System.out.println("====Choice====");
```

```
System.out.println("1.Register\n2.Login\n3.exit");
System.out.println("Enter the Choice:");
int choice = Integer.parseInt(s.nextLine());
switch(choice)
{
case 1:
        System.out.println("Enter the UserName:");
        String uName = s.nextLine();
        System.out.println("Enter the PassWord:");
        String pWord = s.nextLine();
        System.out.println("Enter the FirstName:");
        String fName = s.nextLine();
        System.out.println("Enter the LastName:");
        String lName = s.nextLine();
        System.out.println("Enter the Address:");
        String addr = s.nextLine();
        System.out.println("Enter the MailId:");
        String mId = s.nextLine();
        System.out.println("Enter the PhoneNo:");
        long phNo = Long.parseLong(s.nextLine());

        ps1.setString(1,uName);
        ps1.setString(2,pWord);
        ps1.setString(3,fName);
```

```
            ps1.setString(4,lName);
            ps1.setString(5,addr);
            ps1.setString(6,mId);
            ps1.setLong(7,phNo);

            int k = ps1.executeUpdate();
            if(k>0) {
                  System.out.println("User Registered Successfully");
            }
            break;
      case 2:
            System.out.println("Enter the UserName:");
            String uN = s.nextLine();
            System.out.println("Enter the PassWord:");
            String pW = s.nextLine();

            ps2.setString(1,uN);
            ps2.setString(2, pW);

            ResultSet rs = ps2.executeQuery();
            if(rs.next()) {
                  System.out.println("Login Successfull...");
                  System.out.println("Welcome User : "+rs.getString(3));
            }else {
```

```
                    System.out.println("Invalid Login Process...");
                }
                break;
            case 3:
                System.out.println("Program terminated...");
                System.exit(0);
            default:
                System.out.println("Invalid Choice...");
            }//end of switch;
        }//end of loop
        //s.close();
    }catch(Exception e) {e.printStackTrace();}
        }
}
```

==========

Ex_Application:

JDBC Application to update Product Price and qty based on pCode?

Program : DBCon6.java

```
package test;
import java.sql.*;
import java.util.*;
public class DBCon6 {
        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");
    PreparedStatement ps1 = con.prepareStatement
                    ("select * from Product45 where pcode=?");
    PreparedStatement ps2 = con.prepareStatement
   ("update Product45 set pprice=?,pqty=pqty+? where pcode=?");

    ps1.setString(1,pCode);
    ResultSet rs = ps1.executeQuery();
    if(rs.next()) {
            System.out.println("Old price:"+rs.getFloat(3));
            System.out.println("Enter the New Price:");
            float newPrice = s.nextFloat();
            System.out.println("Available qty:"+rs.getInt(4));
            System.out.println("Enter the new qty:");
            int newQty = s.nextInt();

            ps2.setFloat(1,newPrice);
            ps2.setInt(2,newQty);
            ps2.setString(3, pCode);
```

```
                int k = ps2.executeUpdate();
                if(k>0) {
                        System.out.println("Product details updated...");
                }
        }else {
                System.out.println("Invalid ProdCode...");
        }
        s.close();
    }catch(Exception e) {e.printStackTrace();}
        }
}
```

==========

Ex_Application:

JDBC Application to delete Product on pCode?

Program : DBCon7.java

```
package test;
import java.sql.*;
import java.util.*;
public class DBCon7 {
        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");
        PreparedStatement ps1 = con.prepareStatement
                        ("select * from Product45 where pcode=?");
        PreparedStatement ps2 = con.prepareStatement
      ("delete from Product45 where pcode=?");

        ps1.setString(1,pCode);
        ResultSet rs = ps1.executeQuery();
        if(rs.next()) {
                ps2.setString(1,pCode);
                int k = ps2.executeUpdate();
                if(k>0) {
                        System.out.println("Product details deleted...");
                }
        }else {
                System.out.println("Invalid ProdCode...");
        }
        s.close();
    }catch(Exception e) {e.printStackTrace();}
        }
}
==================================================
```

Assignment:

JDBC application to perform the following operations on Book45 based

on user choice:(Choice is repeated until user terminates)

1.AddBook

2.ViewAllBooks

3.ViewBookByBookCode

4.UpdateBookByBookCode(price and qty)

5.DeleteBookByBookCode

Dt : 9/5/2020

**imp*

3.CallableStatement:

=>CallableStatement is an interface from java.sql package and

which is used to execute Procedures and Functions on DB-Product.

=>we use prepareCall() method from 'Connection' interface to

create the implementation object of CallableStatement interface.

Method Signature of prepareCall():

public abstract java.sql.CallableStatement prepareCall

(java.lang.String)throws java.sql.SQLException;

syntax

CallableStatement cs = con.prepareCall("Proc/Func");

define Procedure?

=>procedure is a set-of-queries executed on DataBase product

and after execution it will not return any value.

(Procedure means Non-return_type)

structure of procedure:

```
create or replace procedure Proc_name
(para_list) is
begin
  query1;
  query2;
  ...
end;
/
```

define Function?

=>Function is a set-of-queries executed on DataBase product and after execution it will return the value.

Note:

=>Functions in SQL will use 'return' statement to return the value after execution.

structure of Function:

```
create or replace function Func_name
```

(para_list)return data_type as var data_type;

begin

query1;

query2;

....

return var;

end;

/

==========

**imp*

Create and execute Procedure on DB-Product:

step-1 : Create the following DB-tables using SQL-CommandLine

Bank45(accno,custname,balance,acctype)

CustDetails45(accno,addr,mid,phno)

create table Bank45(accno number(15),custname varchar2(15),

balance number(10,2),acctype varchar2(15),primary key(accno));

create table CustDetails45(accno number(15),addr varchar2(15),

mid varchar2(25),phno number(15),primary key(accno));

step-2 : Construct the Procedure to createAccount

```
create or replace procedure CreateAccount45
(accno number,custN varchar2,bal number,accT varchar2,
addr varchar2,mid varchar2,phno number) is
begin
  insert into Bank45 values(accno,custN,bal,accT);
  insert into CustDetails45 values(accno,addr,mid,phno);
end;
/
```

step-3 : construct JDBC application to execute procedure

Program : DBCon8.java

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon8 {
        public static void main(String[] args) {
      try {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the accNo:");
        long accNo = Long.parseLong(s.nextLine());
        System.out.println("Enter the CustName:");
        String custName = s.nextLine();
```

```
System.out.println("Enter the Balance:");
float bal = Float.parseFloat(s.nextLine());
System.out.println("Enter the AccType:");
String accType = s.nextLine();
System.out.println("Enter the Address:");
String addr = s.nextLine();
System.out.println("Enter the MailId:");
String mId = s.nextLine();
System.out.println("Enter the PhoneNo:");
long phNo = s.nextLong();

Connection con = DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
CallableStatement cs = con.prepareCall
                ("{call CreateAccount45(?,?,?,?,?,?,?)}");
cs.setLong(1,accNo);
cs.setString(2,custName);
cs.setFloat(3,bal);
cs.setString(4,accType);
cs.setString(5,addr);
cs.setString(6, mId);
cs.setLong(7, phNo);

cs.execute();
```

```
            System.out.println("Procedure executed Successfully...");
            con.close();;
            s.close();
        }catch(Exception e) {e.printStackTrace();}
            }

}
```

o/p:

Enter the accNo:

6123456

Enter the CustName:

Raj

Enter the Balance:

12000

Enter the AccType:

savings

Enter the Address:

SrNagar

Enter the MailId:

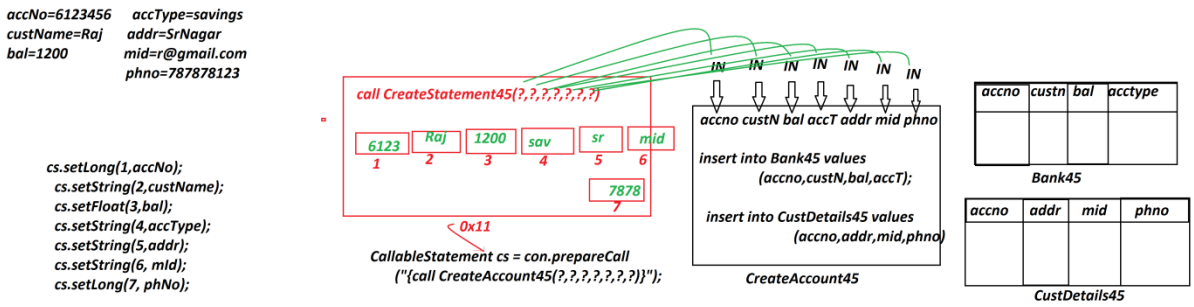
r@gmail.com

Enter the PhoneNo:

7878781234

Procedure executed Successfully...

Diagram:



==========

Assignment:

Create and execute procedure

Step-1 : Construct DB Tables

EmpData45(eid,ename,edesg)

EmpAddress45(eid,hno,sname,city,pincode)

EmpContact45(eid,mid,phno)

EmpSalary45(eid,bsal,totsal)

step-2 : Jdbc Application to execute procedure to update EmpDetails

==========

Dt : 10/5/2022

Types of Procedures:

=>Procedures are categorized into two types:

(i)IN-Parameter Procedures

(ii)OUT-Parameter Procedures

(i)IN-Parameter Procedures:

=>The procedures which take the data from JavaProgram and update DB-Tables are known as IN-Parameter procedures.

Ex:

above program(DBCon8.java)

(ii)OUT-Parameter Procedures:

=>The procedures which take the data from DB-Tables and sent to JavaProgram are known as OUT-Parameter Procedures.

Ex:

Construct Procedure to display the Complete details of Customer based on accno.

step-1 : Construct procedure to retrieve Complete details based on accno.

```
create or replace procedure RetrieveDetails45
(ano number,cname OUT varchar2,bal OUT number,atype OUT varchar2,
adr OUT varchar2,md OUT varchar2,pno OUT number) is
begin
 select custname,balance,acctype into cname,bal,atype from Bank45 where accno=ano;
 select addr,mid,phno into adr,md,pno from CustDetails45 where accno=ano;
```

```
end;
/
```

step-2 : JDBC application to execute procedure.

Program : DBCon9.java

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon9 {
        public static void main(String[] args) {
       try {
          Scanner s = new Scanner(System.in);
          System.out.println("Enter the AccNo:");
          long accNo = s.nextLong();
          Connection con = DriverManager.getConnection
          ("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
          CallableStatement cs = con.prepareCall
                        ("{call RetrieveDetails45(?,?,?,?,?,?,?)}");
          cs.setLong(1, accNo);
          cs.registerOutParameter(2,Types.VARCHAR);
          cs.registerOutParameter(3,Types.FLOAT);
          cs.registerOutParameter(4,Types.VARCHAR);
          cs.registerOutParameter(5,Types.VARCHAR);
```

```
            cs.registerOutParameter(6,Types.VARCHAR);
            cs.registerOutParameter(7,Types.BIGINT);
            cs.execute();
            System.out.println("AccNo:"+accNo);
            System.out.println("CustName:"+cs.getString(2));
            System.out.println("Balance:"+cs.getFloat(3));
            System.out.println("AccType:"+cs.getString(4));
            System.out.println("Address:"+cs.getString(5));
            System.out.println("MailId:"+cs.getString(6));
            System.out.println("PhoneNO:"+cs.getLong(7));;
            s.close();
        }catch(Exception e) {e.printStackTrace();}
          }

}
```

o/p:

Enter the AccNo:

6123456

AccNo:6123456

CustName:Raj

Balance:12000.0

AccType:savings

Address:SrNagar

MailId:r@gmail.com

PhoneNO:7878781234

---

faq:

define registerOutParameter() method?

=>This registerOutParameter() method specify the type of value to be recorded to the fields of CallableStatement object in OUT-Parameter procedure.

---

faq:

wt is the advantage of Procedures?

=>Using Procedures we can execute multiple queries at-a-time and which saves the execution time and generate HighPerformance of an application.

Assignment:

Construct Procedure to display the complete details of Employee based on eId.

==========

**imp*

Create and execute Functon:

step-1 : Construct function to retrieve balance based on accno.

create or replace function RetrieveBalance45

```
    (ano number)return number as bal number;
begin
  select balance into bal from Bank45 where accno=ano;
  return bal;
end;
/
```

step-2 : JDBC Application to execute function

Program : DBCon10.java

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon10 {
        public static void main(String[] args) {
      try {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the AccNo:");
        long accNo = s.nextLong();
        Connection con = DriverManager.getConnection
       ("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
        CallableStatement cs = con.prepareCall
                        ("{call ? := RetrieveBalance45(?)}");
        cs.registerOutParameter(1,Types.FLOAT);
```

```
            cs.setLong(2,accNo);
            cs.execute();
            System.out.println("AccNo:"+accNo);
            System.out.println("Balance:"+cs.getFloat(1));
            con.close();
            s.close();
        }catch(Exception e) {e.printStackTrace();}
            }
}
```

o/p:

Enter the AccNo:

6123456

AccNo:6123456

Balance:12000.0

==========

Assignment:

Construct Function to display totSal of an employee based on eId.

==========

==========