

Dt : 8/3/2025

Ex:

Construct JDBC Application to perform the following operations on Choice based on AccNo

1.UpdateBankCustomer

2.DeleteBankCustomer

Program : DBCon7.java

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon7 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try(s){
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection
                ("jdbc:oracle:thin:@localhost:1521:xe","system","tiger");
            PreparedStatement ps1 = con.prepareStatement
                ("select * from BankCustomer72 where accno=?");
                //Compilation process
            PreparedStatement ps2 = con.prepareStatement
                ("update BankCustomer72 set balance=? where
accno=?");
                //Compilation Process
            PreparedStatement ps3 = con.prepareStatement
                ("delete from BankCustomer72 where accno=?");
                //Compilation Process
            System.out.println("Enter the Cust-AccNo to perform Update/Delete
operation:");
            Long accNo = s.nextLong();
            ps1.setLong(1, accNo);
            ResultSet rs = ps1.executeQuery();
            if(rs.next()) {
                System.out.println("*****Operation Choice*****");
                System.out.println("\t1.UpdateBankCustomer"
                    + "\n\t2.DeleteBankCustomer");
                System.out.println("Enter your Choice:");
                int choice = s.nextInt();
                switch(choice) {
                    case 1:
                        System.out.println("Existing
balance:"+rs.getFloat(4));
```

```

        System.out.println("Enter the new balance:");
        float nBal = s.nextFloat();
        ps2.setFloat(1, nBal);
        ps2.setLong(2, accNo);
        int k1 = ps2.executeUpdate();
        if(k1>0) {
            System.out.println("Customer Updated
Successfully...");
        }
        break;
    case 2:
        ps3.setLong(1, accNo);
        int k2 = ps3.executeUpdate();
        if(k2>0) {
            System.out.println("Customer deleted
Successfully....");
        }
        break;
    default:
        System.out.println("Invalid Choice....");
    } //end of switch
} else {
    System.out.println("Invalid accNo....");
}
con.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

o/p:(Update)

Enter the Cust-AccNo to perform Update/Delete operation:

454541234

*******Operation Choice*******

1.UpdateBankCustomer

2.DeleteBankCustomer

Enter your Choice:

1

Existing balance:16000.0

Enter the new balance:

20000

Customer Updated Successfully...

o/p:(Delete)

Enter the Cust-AccNo to perform Update/Delete operation:

454541234

*****Operation Choice*****

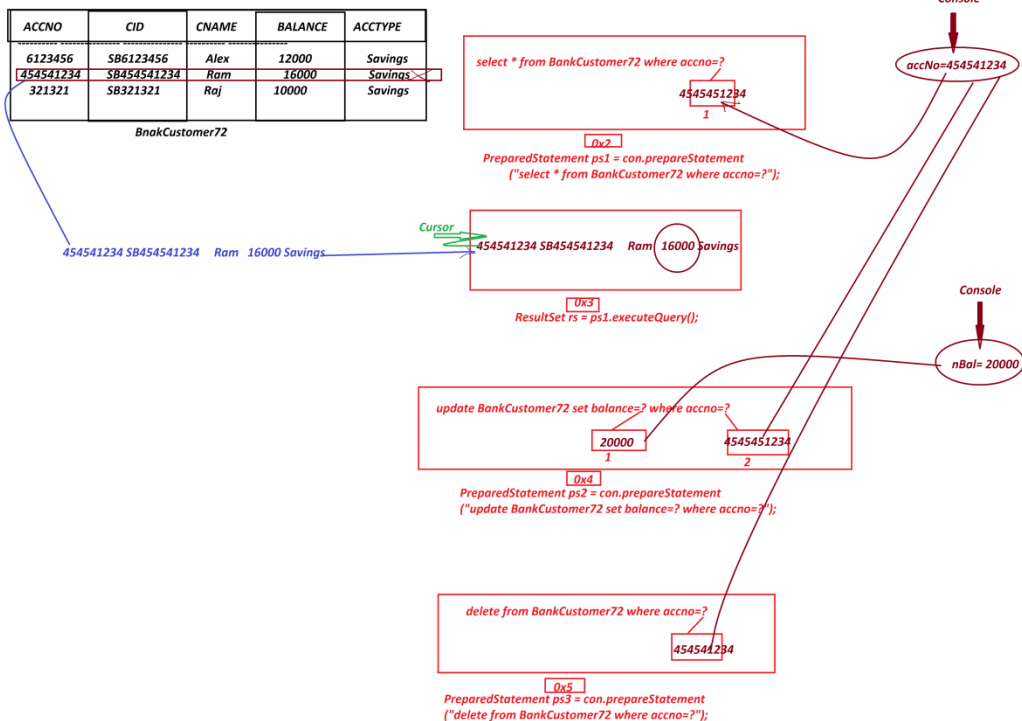
1.UpdateBankCustomer

2.DeleteBankCustomer

Enter your Choice:

2

Customer deleted Successfully....



***imp**

'ResultSet' in JDBC:

=>'ResultSet' is an interface from java.sql package and which is instantiated to hold the result generated from select-queries.

=>'ResultSet' Objects are categorized into two types:

1.NonScrollable ResultSet Objects

2.Scrollable ResultSet Objects

1.NonScrollable ResultSet Objects:

=>In NonScrollable ResultSet Objects the cursor can be moved only in one direction,from top-of-table-data to bottom-of-table-data,which means only in forward direction.

=>we use the following syntax to create NonScrollable ResultSet Object:

syntax-1 : Using 'Statement'

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery("select-query");

syntax-2 : Using 'PreparedStatement'

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

ResultSet rs = ps.executeQuery();

2.Scrollable ResultSet Objects:

=>In Scrollable ResultSet Objects the cursor can be moved in both directions,which means can be moved in foward and backward directions.

=>we use the following syntax to create Scrollable ResultSet Object:

syntax-1 : Using 'Statement'

```
Statement stm = con.createStatement(type,mode);
```

```
ResultSet rs = stm.executeQuery("select-query");
```

syntax-2 : Using 'PreparedStatement'

```
PreparedStatement ps = con.prepareStatement(type,mode,"select-query-structure");
```

```
ResultSet rs = ps.executeQuery();
```
