Dt : 28/1/2027

DB Table : Bank70(accno,name,balance,acctype)

Primary Key : accno

```
create table Bank70(accno number(15),name varchar2(15),balance number(10,2),
acctype varchar2(15),primary key(accno));
```

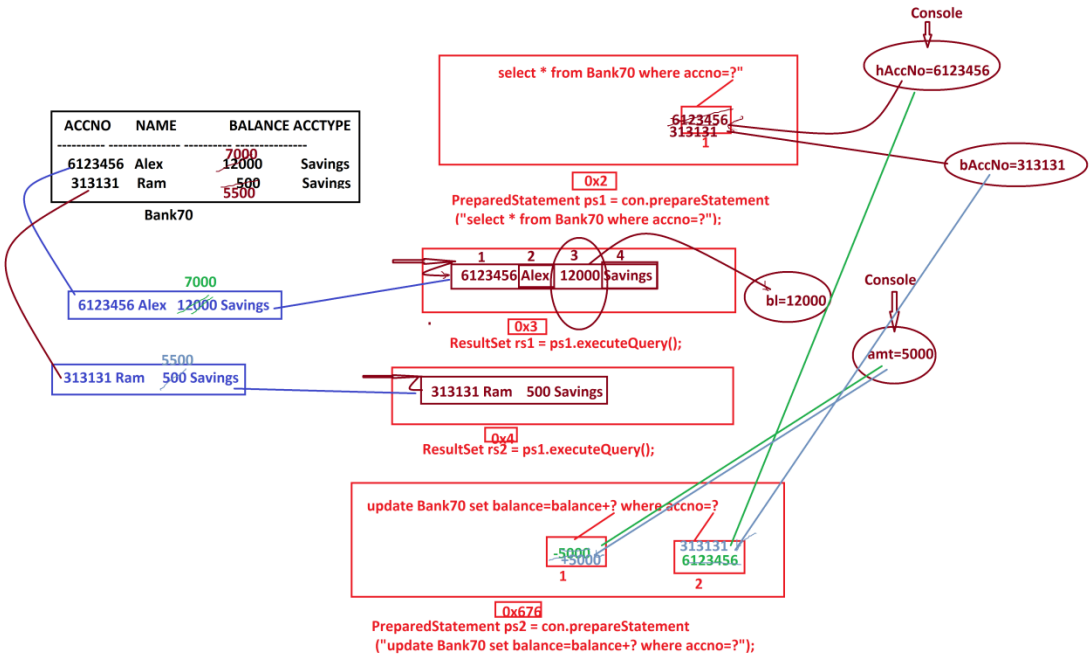
```
insert into Bank70 values(6123456,'Alex',12000,'Savings');
insert into Bank70 values(313131,'Ram',500,'Savings');
```

```
SQL> select * from Bank70;

    ACCNO NAME            BALANCE ACCTYPE
---------- --------------- ---------- ---------------
  6123456 Alex             12000 Savings
   313131 Ram                500 Savings

SQL>
```

Transaction : Transfer amt:5000/- from accNo:6123456 to accNo:313131

Statement-1 : Subtract amt:5000/- from accNo:6123456

Statement-2 : Add amt:5000/- to accNo:313131



Program : DBCon11.java

```
package test;

import java.util.*;

import java.sql.*;

public class DBCon11 {

        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 Bank70 where accno=?");
PreparedStatement ps2 = con.prepareStatement
                ("update Bank70 set balance=balance+? where accno=?");
System.out.println("commit status : "+con.getAutoCommit());
con.setAutoCommit(false);
System.out.println("commit status : "+con.getAutoCommit());
Savepoint sp = con.setSavepoint();
System.out.println("Enter the HomeAccNo:");
long hAccNo = s.nextLong();
ps1.setLong(1, hAccNo);
ResultSet rs1 = ps1.executeQuery();
if(rs1.next()) {
        float bl = rs1.getFloat(3);
        System.out.println("Enter the benificieryAccNo:");
        long bAccNo = s.nextLong();
        ps1.setLong(1, bAccNo);
        ResultSet rs2 = ps1.executeQuery();
        if(rs2.next()) {
                System.out.println("Enter the amt to be transferred:");
                float amt = s.nextFloat();
                if(amt<=bl) {
                        //Statement-1 : Subtract amt:5000/- from accNo:6123456
```

```
                    ps2.setFloat(1,-amt);
                    ps2.setLong(2, hAccNo);
                    int p = ps2.executeUpdate();//Buffer Updated

                    //Statement-2 : Add amt:5000/- to accNo:313131
                    ps2.setFloat(1, +amt);
                    ps2.setLong(2, bAccNo);
                    int q = ps2.executeUpdate();//Buffer Updated

                    if(p==1 && q==1) {
                        con.commit();//Updating Database
                        System.out.println("Transaction Successfull...");
                    }else {
                        System.out.println("Transaction failed...");
                    }
                }else {
                    con.rollback(sp);
                    System.out.println("InSufficient Fund...");
                }
            }else {
                System.out.println("Invalid bAccNo...");
            }
        }else {
            System.out.println("Invalid hAccNo...");
        }
```

```
        }catch(Exception e) {
          e.printStackTrace();
    }
        }
}
```

o/p:

commit status : true

commit status : false

Enter the HomeAccNo:

6123456

Enter the benificieryAccNo:

313131

Enter the amt to be transferred:

5000

Transaction Successfull...

Assignment:

(i)Update above program by generated else-block-messages from 'catch-block'

(ii)Create Table : TransLog70(hccno,baccno,amt,datetime)

Primary Key : haccno

when Transaction Successfull,then record the details.

**imp*

Batch Processing in JDBC:

=>The process of collecting multiple queries as batch and executing at-a-time is known as Batch Processing or Batch Update Processing.

=>In Batch Processing we cannot perform select-operation.

=>We use 'Statement' to perform batch processing,because we can update multiple db tables at-a-time

=>We use the following methods to perform Batch Processing:

1.addBatch()

2.executeBatch()

3.clearBatch()

1.addBatch():

=>addBatch()-method is used to add queries to batch.

Method Signature:

public abstract void addBatch(java.lang.String) throws java.sql.SQLException;

2.executeBatch():

=>executeBatch()-method is used to execute all queries from the batch.

Method Signature:

public abstract int[] executeBatch() throws java.sql.SQLException;

3.clearBatch():

=>clearBatch()-method will delete all queries from the batch and destroy the batch

Method Signature:

public abstract void clearBatch() throws java.sql.SQLException;