Dt : 16/11/2023

faq:

define Transaction Management?

=>The process of controling the transaction from starting to ending is known as Transaction Management.

Note:

=>JDBC Applications will perform auto-commit operation,which means Commit-operation is performed automatically

=>Part of Transaction Management the auto-commit operation must be stopped,which means we perform commit-operation manually.

=>To perform Transaction Management,we use the following methods:

(a)getAutoCommit()

(b)setAutoCommit()

(c)setSavepoint()

(d)releaseSavepoint()

(e)commit()

(f)rollback()

(a)getAutoCommit():

=>getAutoCommit() method is used know the status of commit-operation

syntax:

boolean b1 = con.getAutoCommit();

(b)setAutoCommit():

=>setAutoCommit() method is used to set auto-commit-operation to 'false'

syntax:

con.setAutoCommit(false);

(c)setSavepoint():

=>setSavepoint() method is used to take one status-point to perform rollback-operation when transaction-failed

syntax:

Savepoint sp = con.setSavepoint();

(d)releaseSavepoint():

=>releaseSavepoint() method is used to delete savepoints.

syntax:

con.releaseSavepoint(sp);

(e)commit():

=>commit() method is used to save the data from temporary buffer to DB

product permanently.

syntax:

con.commit();

(f)rollback():

=>rollback() method is used to perform rollback-operation.

syntax:

con.rollback(sp);

Ex-program:(Demonstrating Transaction Management)

DB Table : BankCutomer57(accno,cname,bal,acctype)

primary key : accno

```
create table BankCustomer57(accno number(15),cname varchar2(15),
bal number(10,2),acctype varchar2(15),primary key(accno));

insert into BankCustomer57 values(6123456,'Alex',12000,'savings');
insert into BankCustomer57 values(313131,'Raj',500,'savings');
```

```
   ACCNO CNAME               BAL ACCTYPE
---------- --------------- ---------- ---------------
```

6123456 Alex 12000 savings

313131 Raj 500 savings

Transaction : Transfer 3000/- from accNo:6123456 to accNo:313131

Statement-1 : Subtract 3000/- from accNo:6123456

Statement-2 : Add 3000/- to accNo:313131

Program : DemoCon11.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;){
        try {

Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con =
DriverManager.getConnection
              ("jdbc:oracle:thin:@localhost:1521:xe",
                      "system","manager");
                System.out.println("Commit-status :
"+con.getAutoCommit());
                con.setAutoCommit(false);
                System.out.println("Commit-status :
"+con.getAutoCommit());
```

```
                PreparedStatement ps1 =
con.prepareStatement
                ("select * from BankCustomer57 where
accno=?");
                PreparedStatement ps2 =
con.prepareStatement
                ("update BankCustomer57 set bal=bal+?
where accno=?");
                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
benifiecieryAccNo:");
                long bAccNo = s.nextLong();
                ps1.setLong(1, bAccNo);
                ResultSet rs2 = ps1.executeQuery();
                if(rs2.next()) {
                    System.out.println("Enter the amt to
be transferred:");
                    int amt = s.nextInt();
                    if(amt<=bl) {
                        ps2.setFloat(1, -amt);
                        ps2.setLong(2, hAccNo);
                        int k =
ps2.executeUpdate();//Buffer Updated

                        ps2.setFloat(1, +amt);
                        ps2.setLong(2, bAccNo);
                        int z =
ps2.executeUpdate();//Buffer Updated

                        if(k==1 && z==1) {
                            con.commit();//Database
Updated
```

```
System.out.println("Transaction Successfull..");
                        }else {
                             con.rollback(sp);

System.out.println("Transaction failed..");
                        }

                    }else {
                        System.out.println("Insuffiecient
fund...");
                    }
                }else {
                     System.out.println("Invalid
bAccNo...");
                }
                }else {
                 System.out.println("Invalid
homeAccNo...");
                }
        }catch(Exception e) {e.printStackTrace();}
        }//end of try with resource
    }
}
```

o/p:

Commit-status : true

Commit-status : false

Enter the HomeAccNo:

6123456

Enter benifiecieryAccNo:

313131

Enter the amt to be transferred:

3000

Transaction Successfull..

ACCNO	CNAME	BAL	ACCTYPE
6123456	Alex	9000	savings
313131	Raj	3500	savings

==========

Note:

step-1 : set auto-commit-operation to false

step-2 : Take savepoint to perform rollback

step-3 : Execute all sub-statements in Transaction

step-4 : Check all sub-statements are executed successfully or not

step-5 : If all sub-statements are executed successfully then perform commit-operation,else perform rollback-operation

==========

Assignment:

Update above application by converting else-block messages to catch-block messages(exception-messages)

==========
==