Dt : 11/5/2022

**imp*

Transaction Management in JDBC:

faq:

define Transaction?

=>The set-of-statements executed on a resource or resources using ACID properties is known as Transaction.

A - Atomicity

C - Consistency

I - Isolation

D - Durability

A - Atomicity:

=>The process completing the transaction to completed state or not(Transaction comes to zero state),is known as Atomicity.

C - Consistency:

=>The process in which the selected resources state remains same until the transaction completed,is known as Consistency

I - Isolation:

=>The process in which multiple users executing independently is known as Isolation.

D - Durability:

=>Once the transaction is successfull then we have to record the state of transaction is known as Durability.

---

faq:

define Transaction Management?

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

=>The following methods are used in Transaction Management process:

1. *setAutoCommit()*
2. *getAutoCommit()*
3. *setSavepoint()*
4. *removeSavepoint()*
5. *commit()*
6. *rollback()*

1.setAutoCommit():

=>This method is used to set auto-commit operation to 'false', because the JavaPrograms perform commit operations automatically.

2.getAutoCommit():

=>This method is used to known the state of commit operation.

3.setSavepoint():

=>setSavepoint() method is used to set the savepoint for rollback operation.

4.removeSavepoint():

=>This removeSavepoint() method is used to delete the savepoint.

5.commit():

=>commit() method is used to perform commit operation after the transaction is successfull.

6.rollback():

=>rollback() method is used to perform rollback operation when the transaction is failed.

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

SubT1 :Subtract amt:3000/- from accNo:6123456

SubT2 :Add amt:3000/- to accNo:313131

Ex_Program : DBCon11.java

```
package test;
import java.sql.*;
```

```
import java.util.*;;
public class DBCon11 {
        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");
         System.out.println("Commit status : "+con.getAutoCommit());
         con.setAutoCommit(false);
         System.out.println("Commit status : "+con.getAutoCommit());
         Savepoint sp = con.setSavepoint();
         PreparedStatement ps1 = con.prepareStatement
          ("select * from Bank45 where accno=?");
         PreparedStatement ps2 = con.prepareStatement
          ("update Bank45 set balance=balance+? where accno=?");
         System.out.println("Enter homeAccNo:");
         long hAccNo = s.nextLong();
         ps1.setLong(1, hAccNo);
         ResultSet rs1 = ps1.executeQuery();
         if(rs1.next())
         {
                  float bal = rs1.getFloat(3);
                  System.out.println("Enter benefiecieryAccNo:");
                  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<=bal)
        {
                ps2.setInt(1,-amt);
                ps2.setLong(2,hAccNo);
                int i = ps2.executeUpdate();//Buffer updated

                ps2.setInt(1,amt);
                ps2.setLong(2,bAccNo);
                int j = ps2.executeUpdate();//Buffer updated

                if(i==1 && j==1)
                {
                        System.out.println("Transaction Successfull");
                        con.commit();//DataBase Updated
                }//end of if
                else
                {
                        System.out.println("Transaction Failed...");
```

```
                    con.rollback(sp);
                }

            }//end of if
            else
            {
                System.out.println("Insufficient fund...");
}
        }//end of if
        else
        {
            System.out.println("Invalid bAccNo...");
        }
    }//end of if
    else
    {
        System.out.println("Invalid hAccno...");
    }
  }catch(Exception e) {e.printStackTrace();}
    }
}
```

o/p:

Commit status : true

Commit status : false

Enter homeAccNo:

6123456

Enter benefiecieryAccNo:

313131

Enter the amt to be transferred:

3000

Transaction Successfull

----

*SQL> select * from Bank45;*

ACCNO	CUSTNAME	BALANCE	ACCTYPE
6123456	Raj	12000	savings
313131	Ram	500	Savings

*SQL> select * from Bank45;*

ACCNO	CUSTNAME	BALANCE	ACCTYPE
6123456	Raj	9000	savings
313131	Ram	3500	Savings

SQL>