12.Write a program for Transaction.

Theory:

Transaction represents a single unit of work.

The ACID properties describes the transaction management well. ACID stands for Atomicity, Consistency, isolation and durability.

Atomicity: means either all successful or none.

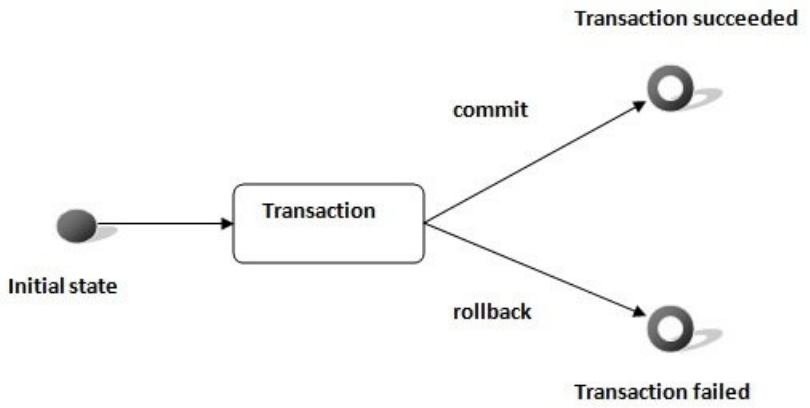
Consistency: ensures bringing the database from one consistent state to another consistent state.

Isolation: ensures that transaction is isolated from other transaction.

Durability: means once a transaction has been committed, it will remain so, even in the event of errors, power loss etc.

Advantage of Transaction Management

fast performance. It makes the performance fast because database is hit at the time of commit.



Source Code:

```
package q10;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class TransactionDemo {
        public static Connection con;
        public static ResultSet rsltset;
        public static Statement statement;

        public static void main(String[] args) throws SQLException {
```

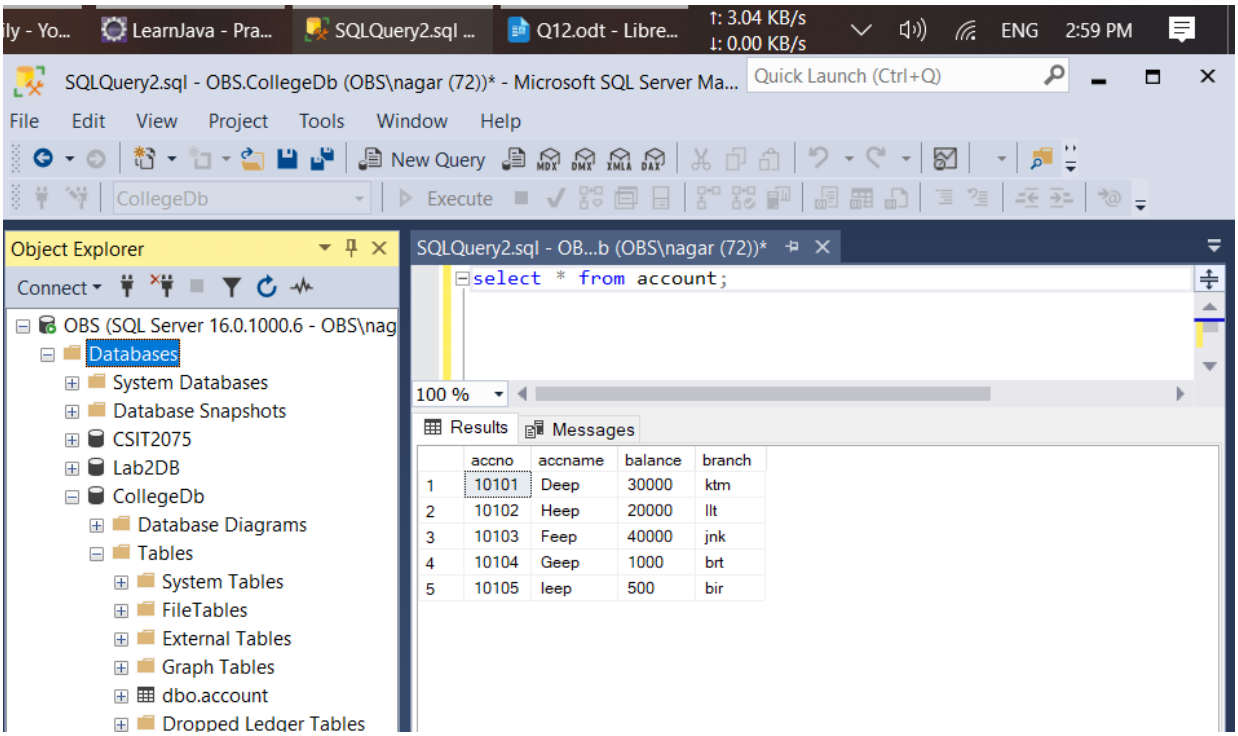
```
		try {
			Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
			System.out.println("Driver loaded");
			String server = "//Obs";
			String database = "CollegeDb";
			int port = 1433;

			String jdbcUrl = "jdbc:sqlserver:" + server + ":" + port + ";databaseName=" +
database
					+ ";integratedSecurity=true";
			con = DriverManager.getConnection(jdbcUrl);
			System.out.println("Connection obtained");
			statement = con.createStatement();
			System.out.println("Statement is created");
			con.setAutoCommit(false);
			String sql1 = "UPDATE account SET "+ "balance=balance-10000 "+" WHERE
accname='Deep'";
			String sql2="UPDATE account SET "+ "balance=balance+10000 "+" WHERE
accname='Geep'";

			statement.executeUpdate(sql1);
			statement.executeUpdate(sql2);
			con.commit();// Explicit Method for executing transactions.
			ResultSet rsltset;
			rsltset = statement.executeQuery("SELECT * from account;");
			System.out.println("After Transaction Complete");
			while (rsltset.next()) {
				int ano = rsltset.getInt("accno");
				String aname = rsltset.getString("accname");
				float bal = rsltset.getFloat("balance");
				String bran = rsltset.getString("branch");
				System.out.print("Account Number: " + ano);
				System.out.print("," + " ");
				System.out.print("Account Name: " + aname);
				System.out.print("," + " ");
				System.out.print("Account Balance: " + bal);
				System.out.print("," + " ");
				System.out.print("Account Branch: " + bran);
				System.out.println(",");
			}
			rsltset.close();
			statement.close();
		} catch (Exception e) {
			con.rollback();
		}
	}
}
```

Output:

Before Code run:



After Code run: