**PRACTICAL 3.1**

**AIM:**

Write a JDBC Program to insert 3 records into student table using Statement. (Assume Student Table with Attributes Name, RollNo and Branch Field) with Batch Processing. (use Statement Interface)

**DESCRIPTION:**

JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database. JDBC works with Java on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

JDBC stands for **J**ava **D**ata**b**ase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

* Making a connection to a database.
* Creating SQL or MySQL statements.
* Executing SQL or MySQL queries in the database.
* Viewing & Modifying the resulting records.

Fundamentally, JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as −

* Java Applications
* Java Applets
* Java Servlets
* Java Server Pages (JSPs)
* Enterprise JavaBeans (EJBs).

All of these different executables are able to use a JDBC driver to access a database, and take advantage of the stored data. JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

**CODE:**

**Practical3\_1.java**

//STEP 1: Import required packages

import java.sql.\*;

public class Practical3\_1{

//JDBC Driver Name and Database URL

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost:3306/s2b130050131525";

//Database Credentials

static final String USER = "root";

static final String PASS = "mysql";

public static void main(String[] args) {

Connection conn = null;

Statement stmt = null;

try{

//STEP 2: Register JDBC Driver

Class.forName(JDBC\_DRIVER);

//STEP 3: Open a Connection

System.out.println("Connecting to selected database");

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

System.out.println("Connected to database successful");

//STEP 4: Creating Statement

stmt = conn.createStatement();

String query= "INSERT INTO student VALUES (20, 'ABC', 'CSE')";

stmt.addBatch(query);

stmt.addBatch("INSERT INTO student VALUES (68, 'PQR', 'ME')");

stmt.addBatch("INSERT INTO student VALUES (11, 'XYZ', 'Civil')");

//STEP 5: Execute Batch

System.out.println("Inserting records into the table");

stmt.executeBatch();

System.out.println("Records inserted");

} catch(SQLException se){

//Handle errors for JDBC

se.printStackTrace();

} catch(Exception e){

//Handle errors for Class.forName

e.printStackTrace();

} finally{

//finally block used to close resources

try{

if(stmt!=null)

conn.close();

} catch(SQLException se){

}

try{

if(conn!=null)

conn.close();

} catch(SQLException se){

se.printStackTrace();

}

}

System.out.println("Goodbye!");

}

}

**OUTPUT:**



