**PRACTICAL 3.3**

**AIM:**

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

**Description:**

The *PreparedStatement* interface extends the Statement interface, which gives you added functionality with a couple of advantages over a generic Statement object. This statement gives you the flexibility of supplying arguments dynamically. All parameters in JDBC are represented by the **?** symbol, which is known as the parameter marker. You must supply values for every parameter before executing the SQL statement.

**CODE:**

**Practical3\_3.java**

//STEP 1: Import required packages

import java.sql.\*;

public class Practical3\_3 {

//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: Preparing Statement

stmt = conn.prepareStatement("INSERT INTO student(rollNo, name, branch) VALUES(?, ?, ?)");

stmt.setInt(1,104);

stmt.setString(2,"Nishant");

stmt.setString(3,"CSE");

stmt.addBatch();

stmt.setInt(1,105);

stmt.setString(2,"Soham");

stmt.setString(3,"ME");

stmt.addBatch();

stmt.setInt(1,106);

stmt.setString(2,"Puja");

stmt.setString(3,"EC");

stmt.addBatch();

//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:**



