ITL 3 PRACTICAL - 01

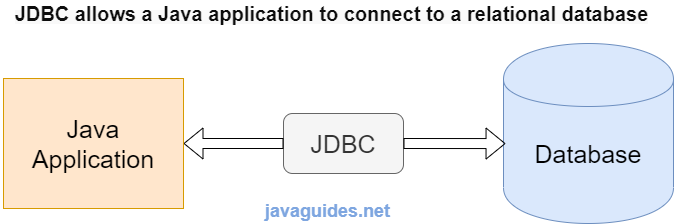
**AIM**

**Maintain record of students and perform CRUD functionality.**

**JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:**

* **JDBC-ODBC Bridge Driver,**
* **Native Driver,**
* **Network Protocol Driver, and**
* **Thin Driver**

**We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database. It is like Open Database Connectivity (ODBC) provided by Microsoft.**



**Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).**

**import java.io.\*;**

**import java.sql.\*;**

**class StudentJdbcMenu**

**{**

**public static void main(String args[])**

**{**

**Connection con = null;**

**Statement stmt = null;**

**ResultSet rs = null;**

**PreparedStatement ps1 = null, ps2 = null;**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String name;**

**int rno, ch;**

**float per;**

**try**

**{**

**/\* \* Class.forName("org.postgresql.Driver"); \* con=DriverManager.getConnection \* ("jdbc:postgresql:5333","root","redhat2"); \*/**

**Class.forName("com.mysql.jdbc.Driver");**

**con = DriverManager.getConnection("jdbc:mysql:///ashu", "root", "");**

**stmt = con.createStatement();**

**ps1 = con.prepareStatement("insert into student values(?,?,?)");**

**ps2 = con .prepareStatement("update student set name=?,percentage=? where roll\_no=?");**

**if (con != null)**

**{**

**do { System.out .println("1.Insert \n 2.Modify\n 3.Delete \n 4.Search \n 5.View All \n 6.Exit\n");**

**System.out.println("Enter the choice:");**

**ch = Integer.parseInt(br.readLine());**

**switch (ch) { case 5: rs = stmt.executeQuery("select \* from student");**

**System.out .println("Roll Number \t Name \t Percentage\n");**

**while (rs.next())**

**{**

**System.out.println(rs.getInt(1) + "\t\t" + rs.getString(2) + "\t\t" + rs.getFloat(3) + "\n"); }**

**break;**

**case 1: System.out.println("Enter the roll number:");**

**rno = Integer.parseInt(br.readLine());**

**if (rno < 0) throw new Exception("Enter positive number");**

**System.out.println("Enter the name:");**

**name = br.readLine();**

**System.out.println("Enter the percentage:");**

**per = Float.parseFloat(br.readLine());**

**if (per < 0) throw new Exception("Enter positive percentage");**

**ps1.setInt(1, rno);**

**ps1.setString(2, name);**

**ps1.setFloat(3, per);**

**ps1.executeUpdate();**

**break;**

**case 3: System.out .println("Enter the roll number to be deleted:");**

**rno = Integer.parseInt(br.readLine());**

**if (rno < 0) throw new Exception("Enter positive number");**

**rs = stmt .executeQuery("select \* from student where roll\_no=" + rno);**

**if (rs.next())**

**{**

**stmt.executeUpdate("delete from student where roll\_no=" + rno);**

**System.out.println("Record Deleted");**

**}**

**else**

**// System.out.println("Record not found");**

**throw new Exception("Record Not In Table");**

**break;**

**case 2: System.out.println("Enter the roll number to modify");**

**rno = Integer.parseInt(br.readLine());**

**if (rno < 0) throw new Exception("Enter positive number");**

**rs = stmt .executeQuery("select \* from student where roll\_no=" + rno);**

**if (rs.next())**

**{**

**System.out.println("Enter the name:");**

**name = br.readLine(); System.out.println("Enter the percentage:");**

**per = Float.parseFloat(br.readLine());**

**if (per < 0) throw new Exception("Enter positive percentage");**

**ps2.setString(1, name);**

**ps2.setInt(3, rno);**

**ps2.setFloat(2, per);**

**ps2.executeUpdate();**

**}**

**else throw new Exception("Record not found in table");**

**break;**

**case 4: System.out.println("Enter the roll number to search:");**

**rno = Integer.parseInt(br.readLine());**

**if (rno < 0) throw new Exception("Enter positive number");**

**rs = stmt .executeQuery("select \* from student where roll\_no=" + rno);**

**if (rs.next())**

**{**

**System.out.println("Record Found"); System.out.println("Roll number=" + rs.getInt(1) + "\tName=" + rs.getString(2) + "\tPercentage=" + rs.getFloat(3));**

**}**

**else**

**System.out .println("Sorry! Record not Found in table"); case 6: System.exit(0);**

**default: System.out.println("Wrong choice:");**

**}**

**}**

**while (ch != 6);**

**rs.close();**

**stmt.close();**

**con.close();**

**}**

**}**

**catch (NumberFormatException ae)**

**{**

**System.out.println("Wrong value Entered");**

**}**

**catch (Exception e)**

**{**

**System.out.println(e.getMessage());**

**}**

**}// main**

**}// class**

**OUTPUT:**

**A picture containing graphical user interface

Description automatically generated**

**Graphical user interface

Description automatically generated with low confidence**

**Graphical user interface

Description automatically generated with medium confidence**

**Table

Description automatically generated with low confidence**

**Graphical user interface

Description automatically generated with medium confidence**