Certificate

This is to certify Mr./ Ms. Bhuvaneswari Yennapusala

Studying in Semester 5 ,Class STP2-A ,Having Roll No. 92200151009

has satisfactorily completed his / her term work in the subject (Name & Code)

Advanced Java Programming - 01AI0501 for the term

ending in the month of _____, Academic year 20 24 - 20 25

Date : _____

Faculty In-Charge

Name : Prof. Tarannum Bloch

Signature: _____

Head of Department : _____

Scanned with OKEN Scanner

Index

Name,	Bhuvaneswari Yennapwala	Year,	2024-2025
Subject,	AJP	Class,	STP2-A
Semester,	5	Roll No,	92200151069

No.	Date	Experiment Description / Subject	Page	Marks	Faculty's Signature
1.1		Five basics steps to establish a JDBC connection.	1		
1.2		Explore mySQL IDE & create database & drop it	2-3		
1.3		create student table in mySQL	4-6		
1.4		Develop a JDBC application which retrive & display all records.	7		
1.5		Develop JDBC from user input in html & insert records	8-9		
1.6		Develop JDBC application which asks roll no of student & update	10-11		
1.7		Develop JDBC application which asks roll no of student & deletes it	12		

Scanned with OKEN Scanner

Ex. No: 1_1	Write down Five Basic steps to establish a JDBC connection from a Java Application. Also, mention the sample code for each step.
Date:	

Objective: Establish a connection to a database from a Java application using JDBC, allowing the application to interact with the database for querying, updating, and managing data.

Using JDBC, there are 5 steps to connect any Java application with the database. These steps are as follows:

Register the Driver class: The forName() Class method is used to register the driver class. This method is used to load the driver class dynamically.

1). **Create connection:** The getConnection() method of the DriverManager class is used to establish a connection with the database.

2). **Create statement:** The createStatement() method of the Connection interface is used to create a statement. The object of the statement is responsible for executing queries with the database.

3). **Execute queries:** The executeQuery() method of the Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

4). **Close connection:** By closing the connection object statement and ResultSet will be closed automatically. The close() method of the Connection interface is used to close the connection.

Code:

Step 1: Register the Driver class (Mysql)

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

Step 2 : Create connection

```
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mefgi","root","");
```

Step 3 : create statement

```
Statement stmt=con.createStatement();
```

Step 4 : Execute Queries

```
ResultSet rs=stmt.executeQuery("select * from Student")
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
```

Step 5 : Close Connection

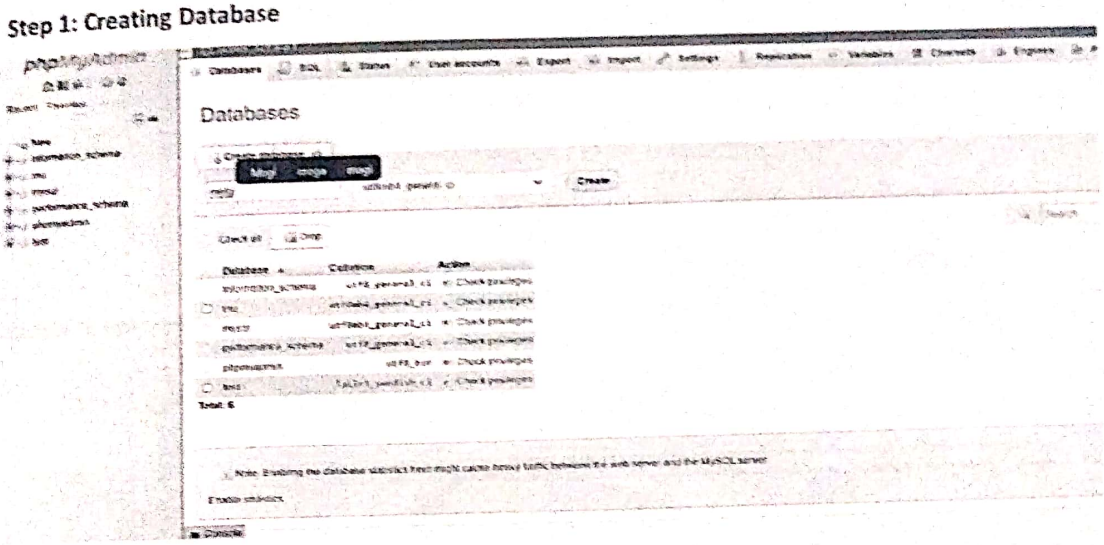
```
con.close();
```


Scanned with OKEN Scanner

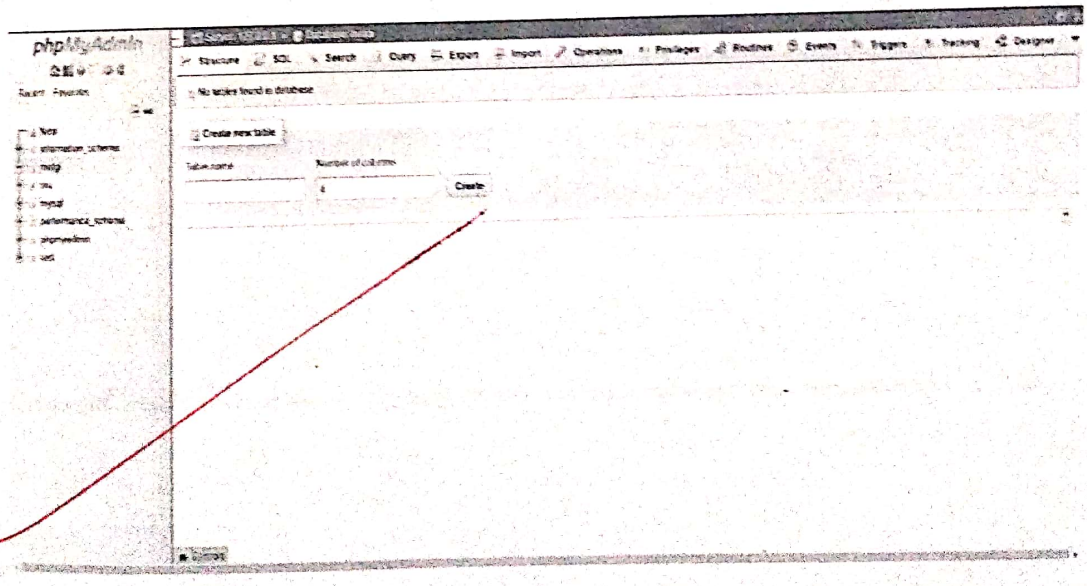
Ex. No: 1_2	Explore MySQL IDE and create a database and drop the database in the MySQL server using the SQL editor window.
Date:	

Objective: Understand the interface and functionalities of a MySQL IDE

Step 1: Creating Database

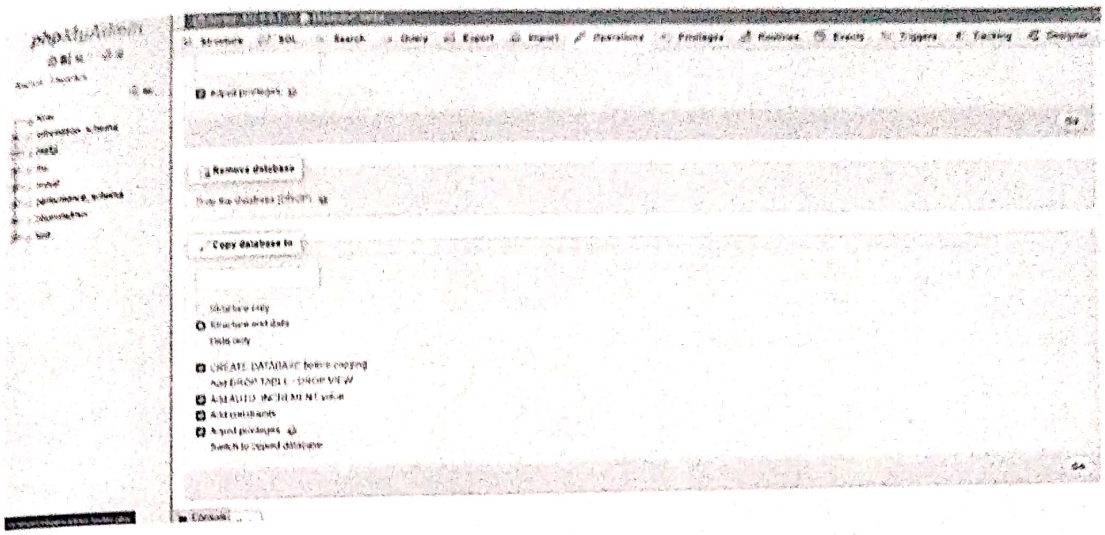


Step 2: Database created

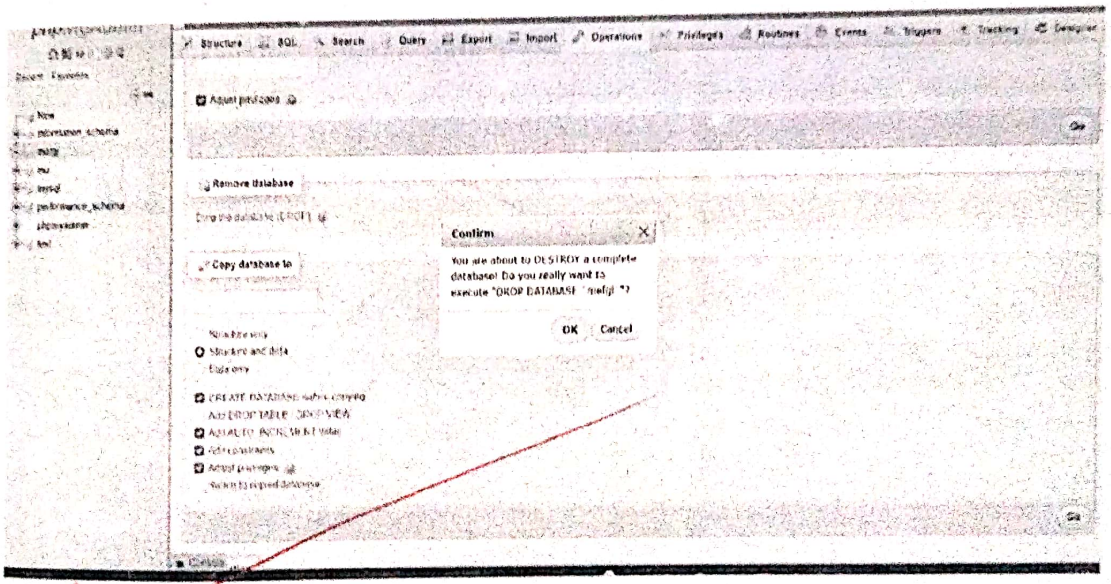


Scanned with OKEN Scanner

Step 3: Deleting the database



Step 4: Confirming delete database

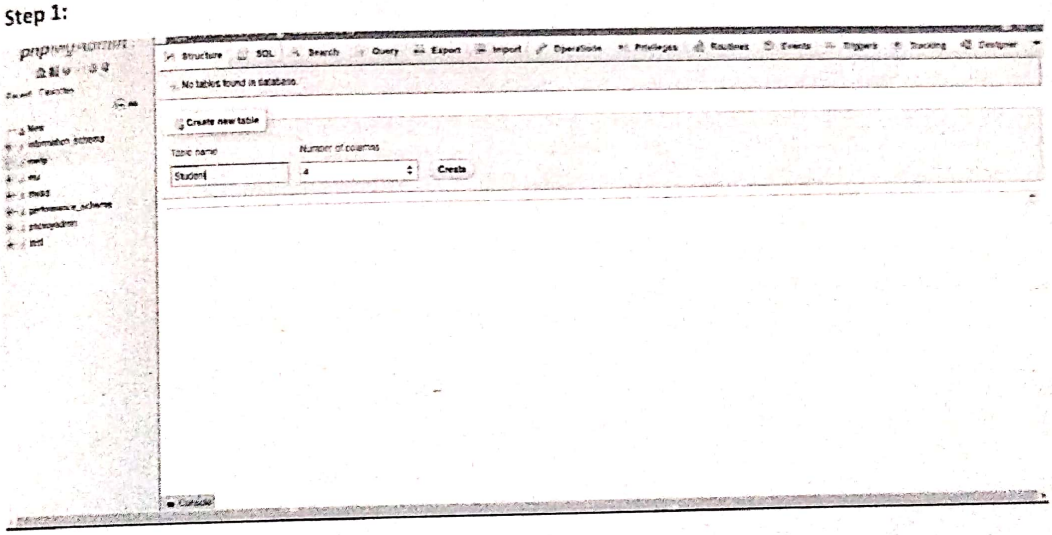


Scanned with OKEN Scanner

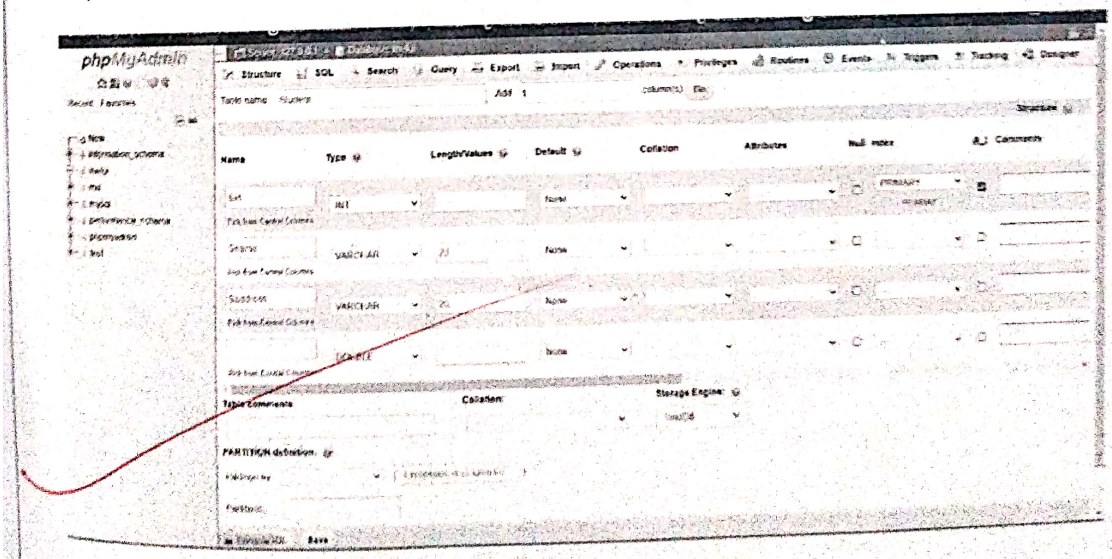
Ex. No: 1_3	Create student table in MySQL database management system having columns rollno, name, and address field and insert few records.
Date:	

Objective: Learn how to create and manage databases in MySQL and perform basic CRUD operations in MySQL.

Step 1:

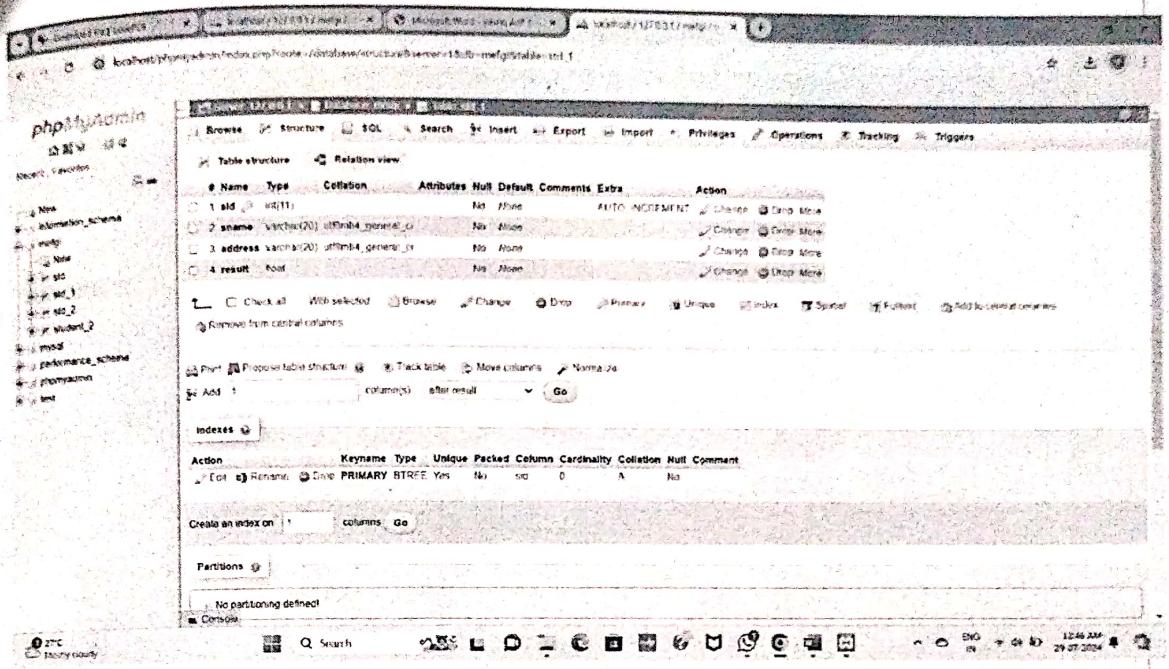


Step 2:

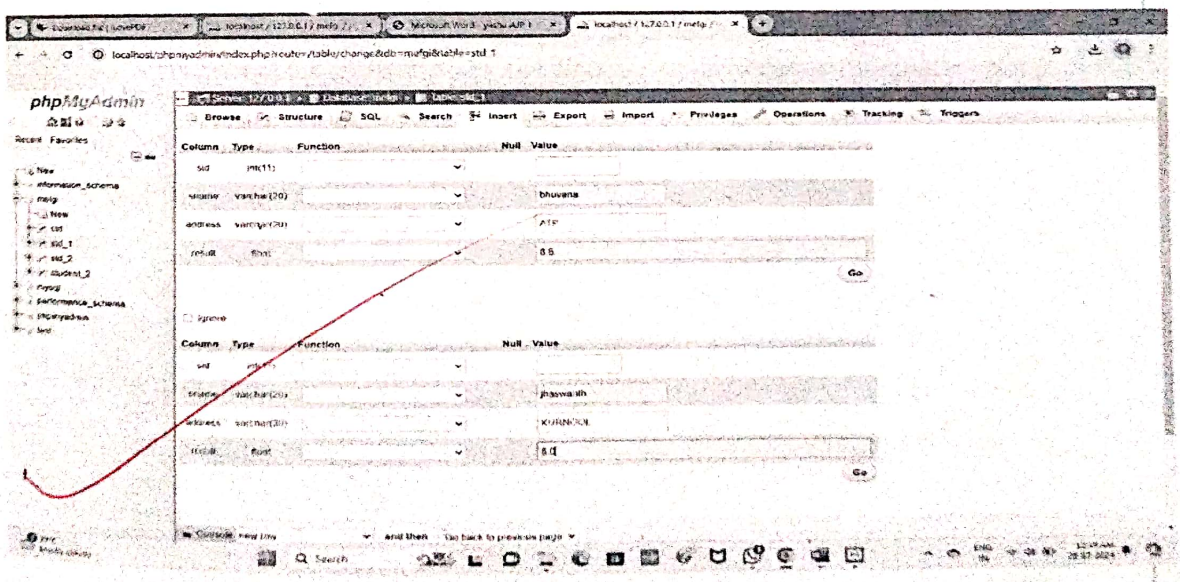


Scanned with OKEN Scanner

Step 3:

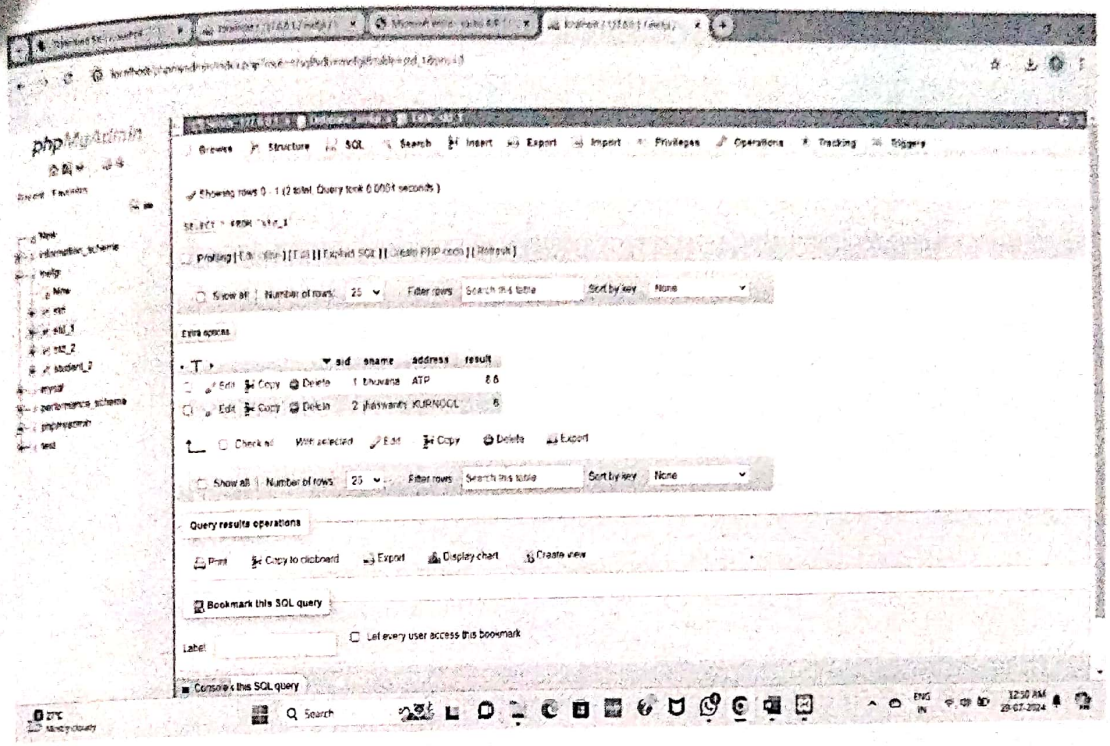


Step 4:



Scanned with OKEN Scanner

Step 5 :



Conclusion: Hence student table in MySQL database management system having columns rollno, name, and address field and insert few records is created

Scanned with OKEN Scanner

Ex. No: 1_4	Develop a JDBC application which retrieve and display all the records from student table created in previous tutorial using a statement object.
Date:	

Objective: Develop a Java application to interact with MySQL databases

using JDBC.

Code :

```
package practical_1_4_bhuna;
import java.sql.*;
public class Practical_1_4_Bhuna {
  public static void main(String[] args) {
try{
  Class.forName("com.mysql.cj.jdbc.Driver");
  Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mefgi","root","");
  Statement stmt = con.createStatement();
  ResultSet rs = stmt.executeQuery("select * from std_2");
  while (rs.next())
    System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
  con.close();
 }
 catch(Exception e){System.out.println(e);}
   }

 }
```

Output :

```
Output - Practical_1_4_Bhuna (run)
run:
3 bhuvana ATP 8.6
4 Sravani bellary 7.8
5 jhaswanth KURNOOL 8.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

7/3/26

Scanned with OKEN Scanner

Ex. No: 1_5	Develop a JDBC application which take input from user in a HTML file and insert record using prepared statement in student table created in table previous tutorial.
Date:	

Objective: Developing a JDBC application which take input from user in a HTML file and inserting record using prepared statement in student table created in table previous tutorial.

Code:

```
package practical1_5_bhuna;
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.util.Scanner;
class Practical1_5_bhuna
{
  public static void main (String[] args){
    try{
  Class.forName("com.mysql.cj.jdbc.Driver");
  Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mefgi","root","");
  PreparedStatement PS=con.prepareStatement("insert into student(sname,address,result values
(?,?,?)");

   Scanner sc = new Scanner(System.in);
   System.out.println("Enter student name: ");
   String nm = sc.next();
   System.out.println("Enter student address: ");
   String addr = sc.next();
   System.out.println("Enter student result: ");
   float res = sc.nextFloat();

   PS.setString(1, nm);
   PS.setString(2, addr);
   PS.setFloat(3, res);

   int result = PS.executeUpdate();

   if(result > 0)
     System.out.println("Record inserted");
   else
     System.out.println("Problem in insertion");


   con.close();
```


Scanned with OKEN Scanner

```
    catch(Exception e ){
        System.out.println(e);
    }

    }
}
```

Output:

```
Output - practical1_5_bhuna (run) #2
run:
Enter student name:
Taru
Enter student address:
Rajkot
Enter student result:
6.7
Record inserted
BUILD SUCCESSFUL (total time: 58 seconds)
```


Scanned with OKEN Scanner

No: 1_6	Develop a JDBC application which asks roll number of student and update records in student table created in table previous tutorial. Use prepared statement object to update the record.

Objective: Developing a JDBC application which asks roll number of student and update records in student table created in table previous tutorial ,Using prepared statement object to update the record.

Code:

```
package practical1_5_bhuna;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.util.Scanner;

public class Practical1_6_bhuna {
    public static void main(String[] args) {
        try {
            // Load MySQL JDBC Driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish connection to the database
            try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mefgi",
"root", "");
                Scanner sc = new Scanner(System.in);
                PreparedStatement ps = con.prepareStatement("UPDATE student SET sname=?,
address=?, result=? WHERE sid=?")) {

                // Get user input
                System.out.println("Enter student name: ");
                String sname = sc.nextLine();

                System.out.println("Enter address: ");
                String addrs = sc.nextLine();

                System.out.println("Enter result: ");
                float result = sc.nextFloat();

                System.out.println("Enter sid: ");
                int sid = sc.nextInt();

                // Set parameters for the SQL query
                ps.setString(1, sname);
                ps.setString(2, addrs);
                ps.setFloat(3, result);
                ps.setInt(4, sid);
```


Scanned with OKEN Scanner

```
            // Execute the update
            int myresult = ps.executeUpdate();

            // Check if the update was successful
            if (myresult > 0) {
                System.out.println("Record Updated");
            } else {
                System.out.println("Problem in updating record");
            }
        }
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
  }
}
```

Output:

practical1_5_bhuna.Practical1_6_bhuna > main > try > catch Exception e >

Output - practical1_5_bhuna (run) #2

```
Enter student name:
nayan
Enter address:
mumbai
Enter result:
7
Enter sid:
9
Record Updated
BUILD SUCCESSFUL (total time: 19 seconds)
```


Scanned with OKEN Scanner

Ex. No: 1_7	Develop a JDBC application which asks roll number of student and delete record using callable statement object in student table created in table previous tutorial.
Date:	

Objective: Developing a JDBC application which asks roll number of student and delete record using callable statement object in student table created in table previous tutorial.

Code:

```
package practical_1_7_bhuna;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.CallableStatement;
import java.util.Scanner;

public class Practical_1_7_bhuna {
  public static void main(String[] args) {
    try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mefgi", "root",
"");
        CallableStatement cs = con.prepareCall("{call dl_student(?)}");
        Scanner sc = new Scanner(System.in)) {
      System.out.println("Enter rollno: ");
      int rollno = sc.nextInt();
      cs.setInt(1, rollno);
      int result = cs.executeUpdate();
      if (result > 0) {
        System.out.println("Record deleted");
      } else {
        System.out.println("Problem in deletion");
      }
    } catch (Exception e) {
      System.out.println("Error: " + e.getMessage());
    }
  }
}
```

Output:

```
Output - practical_1_7_bhuna (run)
run:
Enter rollno:
13
Record deleted
BUILD SUCCESSFUL (total time: 0 seconds)
```