

Name : Kalyani Kangle
Branch: CSE 3rd Year

Task 1: Database Connection Code

Description: In this task , I have performed the database connectivity with java application.This task includes connection code for connecting java application to database here the database used is MySQL.

For Connecting Java Application to Database , we have to follow two most important and mandatory steps:

1.Register Database Driver:

For Registering database driver , In JDBC the `forName()` method of `Class` type is used.

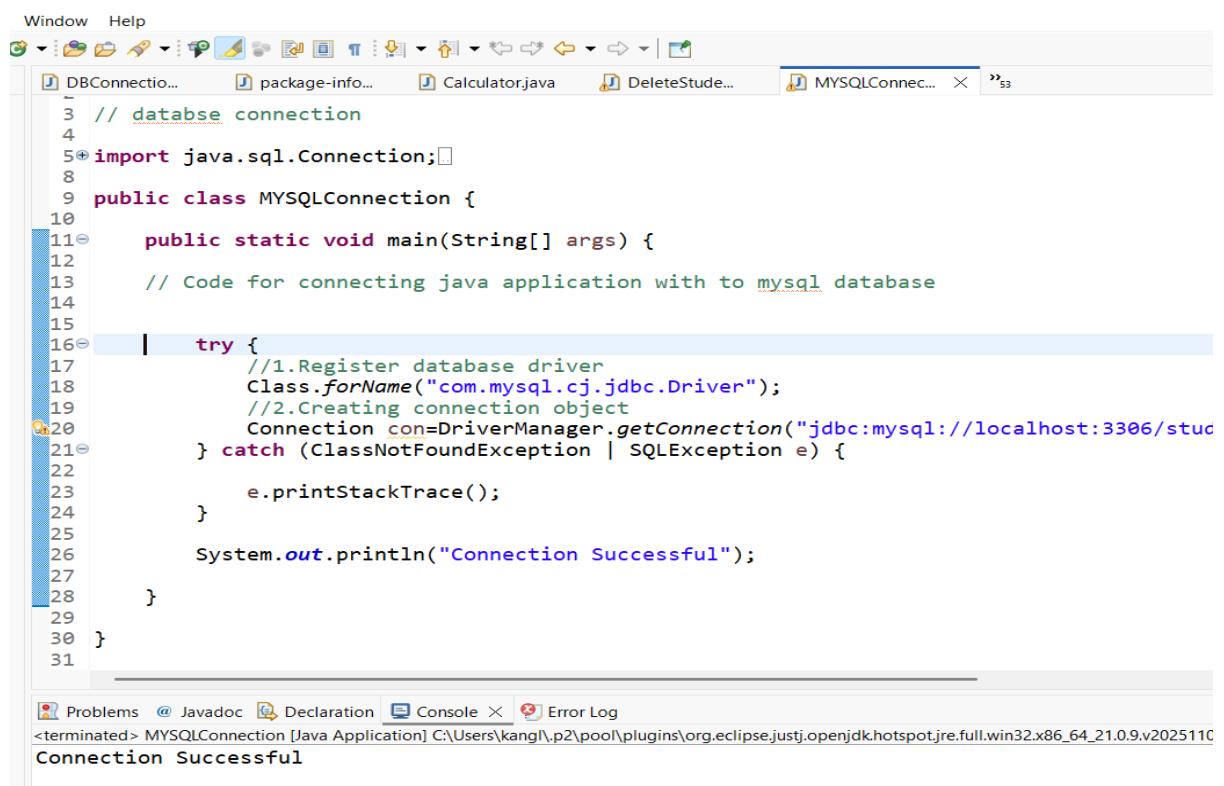
Syntax: `Class.forName("DriverName");`

2.Create Connection Object:

In this step , Connection reference is created , where `Connection` is an Interface.

Syntax: `Connection con=DriverManager.getConnection("URL","USERNAME","PASSWORD");`

Here , the `getconnection()` method of `DriverManager` class takes url , username and password as arguments to create reference on `Connection` , which is we need to use further steps.



The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Window, Help.
- Toolbar:** Standard Eclipse icons.
- Project Explorer:** Shows files like DBConnectio..., package-info..., Calculator.java, DeleteStude..., and MYSQLConnec... (the active file).
- Code Editor:** Displays Java code for a `MySQLConnection` class. The code includes imports for `java.sql.Connection` and `java.sql.DriverManager`. It contains a `main` method that attempts to connect to a MySQL database using the URL `"jdbc:mysql://localhost:3306/stud"`. A try-catch block handles potential `ClassNotFoundException` and `SQLException`.
- Console Tab:** Shows the output "Connection Successful".
- Status Bar:** Shows the path "C:\Users\kangl\p2\pool\plugins\org.eclipse.jdt.core\hotspot\jre\full\win32\x86_64_21.0.9.v2025110" and the message "terminated> MYSQLConnection [Java Application]".

Name : Kalyani Kangle
Branch: CSE 3rd Year

Task 2: Display All Students

Description: This tasks focuses on displaying the data from database to the console through java application.

In this task , First I have created a database name as students where student table is present in which student data id , name , marks are stored . This tasks mainly performed to display all the records from student table. For displaying data from database using java application , the java application should be connected with database.

I have completed this task by performing following 5 steps:

- 1.Registering database driver using Class.forName("DriverName")
- 2.Creating Connection reference that stores url , username and password through getConnection() method of DriverManager class.
- 3.Statement reference is created using Connection reference and use of createStatement() here createStatement() do not hold any query.
- 4.In this step , select query is hold and execute using exceuteQuery() , the result return by executeQuery() is stored in ResultSet rs , now next step is to extract data from rs for that rs.next() function is used to check whether there is next element is present in rs or not , if element or data is present rs.next() returns true otherwise returns false.
- 5.At last getter functions from ResultSet are used to get value from database (table) and display it.
- 6.Closing the connection using close() method.



```
DBConnectio... package-info... DeleteStude... MYSQLConnec... DisplayStude... > s3
1 package com.task2;
2
3 // Displaying students from database
4
5+import java.sql.Connection;□
10
11 public class DisplayStudents {
12
13+    public static void main(String[] args) □
14
15+        try {
16            //1.Register driver
17            Class.forName("com.mysql.cj.jdbc.Driver");
18            //2.Create connection object
19            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student");
20            //3.Create statement object : to hold the query
21            Statement stmt=con.createStatement();
22            //4.Execute the hold query
23            ResultSet rs=stmt.executeQuery("select * from student");
24            System.out.println("ID"+ "\t"+ "NAME"+ "\t"+ "AGE");
25+            while(rs.next()) {
26                System.out.print(rs.getInt("id") + "\t");
27                System.out.print(rs.getString("name") + "\t");
28                System.out.print(rs.getInt("age") + "\n");
29            }
30            //closing connections: freeing up resources
31            stmt.close();
32        }
```

Name : Kalyani Kangle

Branch: CSE 3rd Year

```
17     Class.forName("com.mysql.cj.jdbc.Driver");
18     //2.Create connection object
19     Connection con=DriverManager.getConnection("jdbc:mysql://localhost:
20     //3.Create statement object : to hold the query
21     Statement stmt=con.createStatement();
22     //4.Execute the hold query
23     ResultSet rs=stmt.executeQuery("select * from student");
24     System.out.println("ID"+ "\t" + "NAME" + "\t" + "AGE");
25     while(rs.next()) {
26         System.out.print(rs.getInt("id") + "\t");
27         System.out.print(rs.getString("name") + "\t");
28         System.out.print(rs.getInt("age") + "\n");
29     }
30     //closing connections: freeing up resources
31     stmt.close();
32     rs.close();
33     con.close();
34
35 } catch (ClassNotFoundException | SQLException e) {
36     // TODO Auto-generated catch block
37     e.printStackTrace();
38 }
39
40 }
41
42 }
43 }
```

Problems @ Javadoc Declaration Console × Error Log

<terminated> DisplayStudents [Java Application] C:\Users\kangl\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21

ID	NAME	AGE
1	Kalyani	20
2	Madhavi	20
3	Vivek	21
4	Aakash	21
5	Aarya	23

Name : Kalyani Kangle
Branch: CSE 3rd Year

Task 3: Inserting Student Records

Description: In this task students records are inserted into table student , using PreparedStatement and Parameterized query. The parameterised query is used to take values from users.

This task is performed by implementing following steps:

- 1.The most basic and primary steps is to register database and establishing connection with database.
- 2.Scanner class object is used to take input from users
- 3.Insert query with parameters is hold by prepareStatement() , where the parameters values are passed by user and store into that parameteres using setter methods , that setter methods are used or call using PreparedStatement reference.
- 4.As Insert query is a DML statement , executeUpdate() is used for exceuting it , it returns int value that is number of rows affected.
- 5.At last Connection is closed to free resources

The screenshot shows the Eclipse IDE interface with a Java code editor and a terminal window. The code in the editor is as follows:

```
1 package com.task3;
2
3④ import java.sql.Connection;□
4
5 //Inserting students details in student table by taking input from user through console
6 //Insertinsg student record
7
8 public class InsertStudents {
9
10    public static void main(String[] args) {
11        Scanner sc=new Scanner(System.in);
12        System.out.println("Enter Student id: ");
13        int id=sc.nextInt();
14        System.out.println("Enter Student name: ");
15        String name=sc.next();
16        System.out.println("Enter Student age: ");
17        int age=sc.nextInt();
18        System.out.println("Enter Student marks: ");
19        int marks=sc.nextInt();
20
21        try {
22            //1. Register database driver
23
24            Class.forName("com.mysql.cj.jdbc.Driver");
25
26            //2.Creating connection object
27
28            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/stude
29
30
31 }
```

The terminal window below shows the execution of the program:

```
Problems @ Javadoc Declaration Console × Error Log
<terminated> InsertStudents [Java Application] C:\Users\kangl\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-074
Enter Student id:
11
Enter Student name:
Aashish
Enter Student age:
20
```

Name : Kalyani Kangle

Branch: CSE 3rd Year

The screenshot shows the Eclipse IDE interface with the following details:

- Code Editor:** Displays Java code for inserting student data into a MySQL database. The code uses JDBC to connect to the database, prepare a statement, and execute it. It includes error handling for ClassNotFoundException and SQLException.
- Console Output:** Shows the following terminal output:

```
Enter Student age:  
20  
Enter Student marks:  
90  
1 Rows inserted successfully
```
- Toolbars and Menus:** Standard Eclipse toolbars and menus are visible at the top.
- Bottom Status Bar:** Shows the path to the project and the Java version used.

Name : Kalyani Kangle
Branch: CSE 3rd Year

Task 4: Update Student Records

Description: This task is to update students marks using student id which is to be inserted by user and also takes new marks from the user.

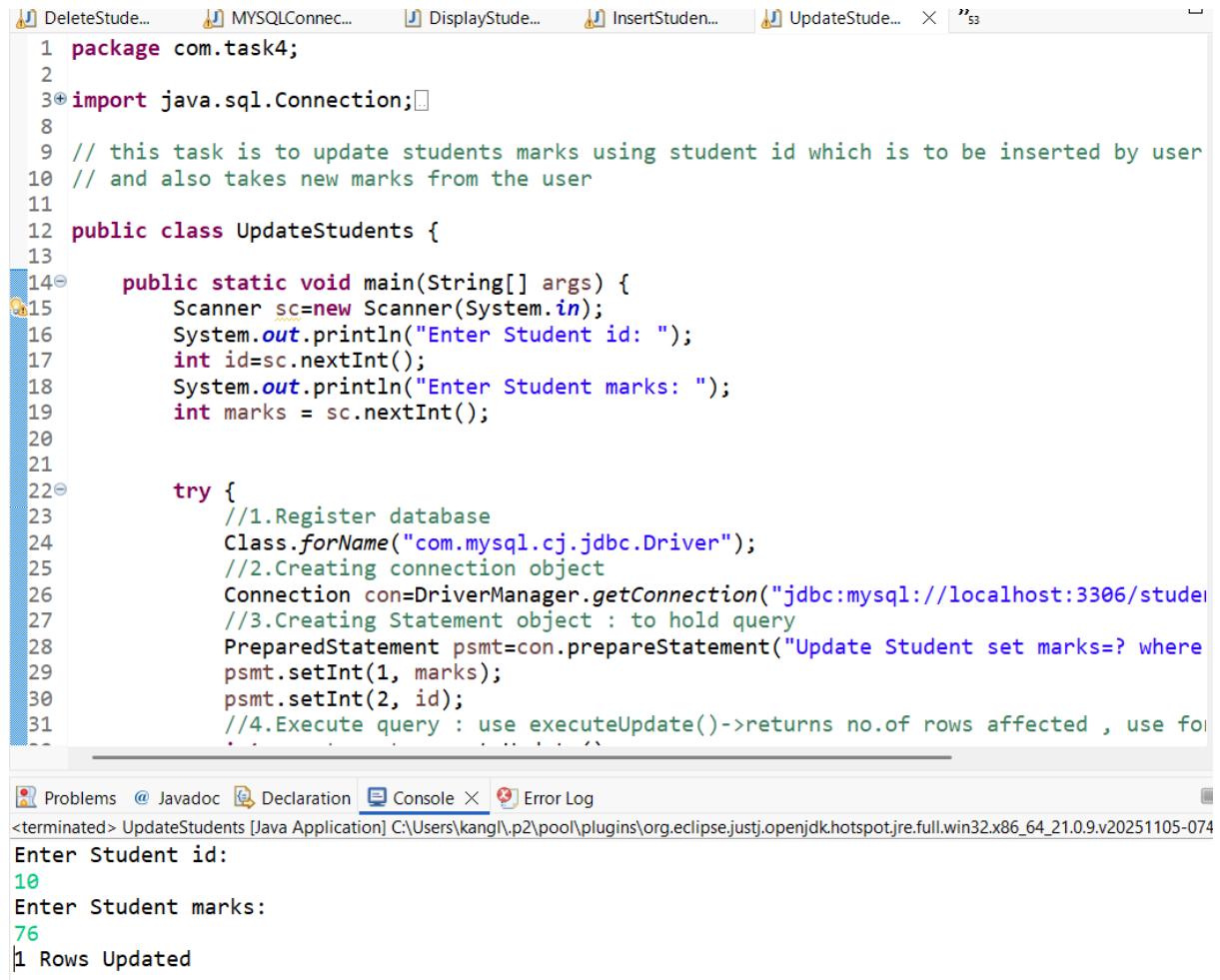
In this task , student marks are updated using Update query which is a parameterised query here that takes input student id at place of parameter from user and also takes new marks from the user.

1. Here , Firstly database driver is registered and Connection is established between java application and Database (MySQL).

2. Input id and marks are taken from user using Scanner Class object , that inputs are passed at the place of parameters to the query using setter methods called on PreparedStatement reference.

3. executeUpdate() method is used to execute query and returns number of rows affected (updated here) .

4. At last , Connections are closed for freeing up the resources by calling close() method.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows several Java files: DeleteStude..., MYSQLConnec..., DisplayStude..., InsertStuden..., UpdateStude..., and a file named "53".
- Code Editor:** Displays the Java code for the `UpdateStudents` class. The code uses JDBC to update student marks based on their ID.
- Console Tab:** Shows the output of the application's execution. It prompts for a student ID (10), a student mark (76), and then displays the message "1 Rows Updated".
- Status Bar:** Shows the path as <terminated> UpdateStudents [Java Application] C:\Users\kangl\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-074.

```
1 package com.task4;
2
3 import java.sql.Connection;
4
5 // this task is to update students marks using student id which is to be inserted by user
6 // and also takes new marks from the user
7
8 public class UpdateStudents {
9
10    public static void main(String[] args) {
11        Scanner sc=new Scanner(System.in);
12        System.out.println("Enter Student id: ");
13        int id=sc.nextInt();
14        System.out.println("Enter Student marks: ");
15        int marks = sc.nextInt();
16
17        try {
18            //1.Register database
19            Class.forName("com.mysql.cj.jdbc.Driver");
20            //2.Creating connection object
21            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/stude");
22            //3.Creating Statement object : to hold query
23            PreparedStatement psmt=con.prepareStatement("Update Student set marks=? where");
24            psmt.setInt(1, marks);
25            psmt.setInt(2, id);
26            //4.Execute query : use executeUpdate()->returns no.of rows affected , use fo
27
28        } catch (Exception e) {
29            e.printStackTrace();
30        }
31    }
32}
```

Name : Kalyani Kangle

Branch: CSE 3rd Year

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows tabs for DeleteStude..., MYSQLConnec..., DisplayStude..., InsertStuden..., UpdateStude..., and a search bar with "53".
- Code Editor:** Displays Java code for updating student marks in a MySQL database. The code uses JDBC to connect to a local MySQL server, prepare a statement, and execute an update query.
- Console Tab:** Active tab, showing the command-line interface output.
- Output:** The console shows:
 - Enter Student id: 10
 - Enter Student marks: 76
- Bottom Status Bar:** Shows the application name "UpdateStudents [Java Application]", the file path "C:\Users\kangl\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-07", and the status "<terminated>".

Name : Kalyani Kangle
Branch: CSE 3rd Year

Task 5: Delete Student Record

Description: In this task , id is taken from user as input , and the record corresponding to that input id is deleted.

- 1.Database Driver is registered and Connection is established between java application and database.
2. id is taken as input from user using Scanner class object and passed it to the Parameterized query using setter method called on Statement reference.
- 3.prepareStatement() is used to hold Delete query.
- 4.executeUpdate() is used to execute query and it returns number of rows deleted.
- 5.At last Connections are closed for freeing up the resources.

The screenshot shows the Eclipse IDE interface with the Java code for deleting a student record. The code uses JDBC to connect to a MySQL database and execute a delete query. The output in the Console tab shows the execution of the program and the confirmation of one row being deleted.

```
8
9 public class DeleteStudents {
10
11    public static void main(String[] args) {
12        Scanner sc=new Scanner(System.in);
13        System.out.println("Enter Student id: ");
14        int id=sc.nextInt();
15
16        try {
17            //1.Register Database Driver
18            Class.forName("com.mysql.cj.jdbc.Driver");
19            //2.Create Connection Object
20            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student");
21            //3.Create statement object
22            PreparedStatement psmt=con.prepareStatement("Delete from Student where id=?");
23            psmt.setInt(1, id);
24            //4.Execute Query
25            int count=psmt.executeUpdate();
26            System.out.println(count+" "+"Rows deleted");
27            //5.Closing Connection
28            con.close();
29        } catch (ClassNotFoundException | SQLException e) {
30            e.printStackTrace();
31        }
32    }
33
34 }
```

Problems @ Javadoc Declaration Console <terminated> DeleteStudents [Java Application] C:\Users\kangl\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v20251105-0741|
Enter Student id:
11
1 Rows deleted