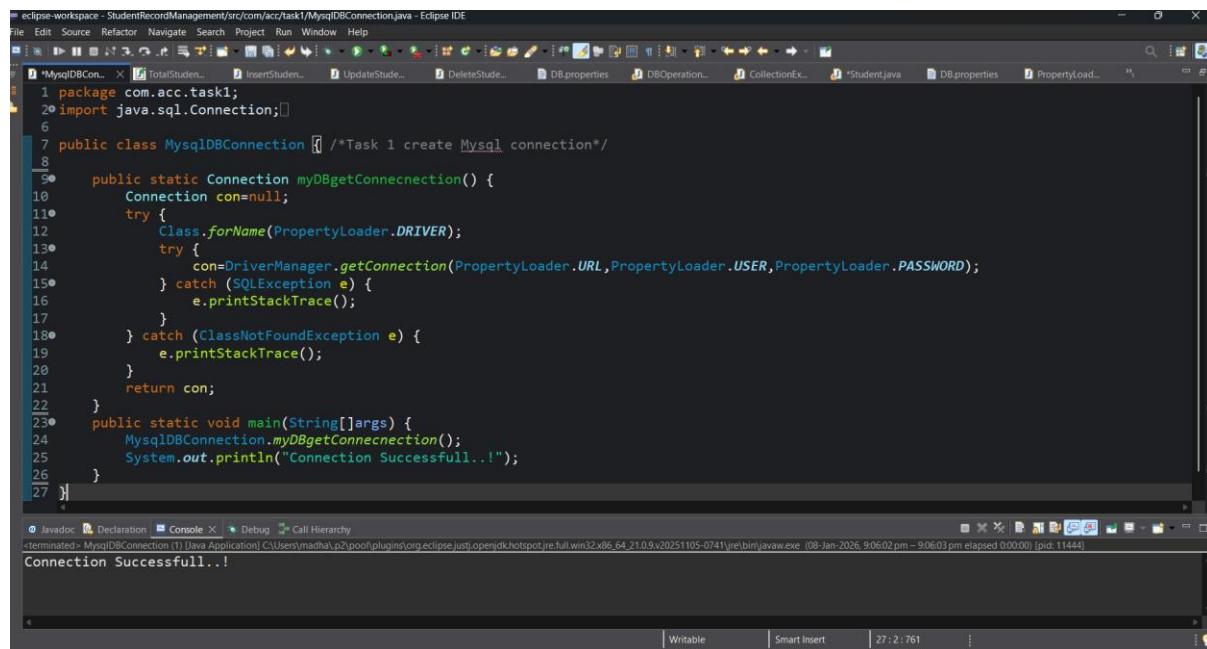


Task 1: Connect to Mysql Database

In this task I have created one class **MysqlDBConnection** in which i created **myDBgetConnection()** method. In this methos I write Database Connection code as following.

1. Created **Connection con** reference variable
2. Registered Driver using **DriverManager.getConnection()** method and pass the url,username,password
3. Call the **myDBgetConnection()** method in main method
4. Run the program
5. Output- printed message “**Connection successful..!**”.



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

- Title Bar:** eclipse-workspace - StudentRecordManagement/src/com/acc/task1/MysqlDBConnection.java - Eclipse IDE
- Toolbar:** Standard Eclipse toolbar with various icons for file operations, search, and project management.
- Left Margin:** Shows line numbers from 1 to 27.
- Code Editor:** Displays the Java code for **MysqlDBConnection**. The code includes a static method **myDBgetConnection()** that uses **DriverManager.getConnection()** to establish a connection. It also includes a **main** method that calls **myDBgetConnection()** and prints "Connection Successful..!" to the console.
- Console Tab:** Shows the output of the application's execution: "Connection Successful..!"
- Status Bar:** Shows the status bar with "Writable", "Smart Insert", and the current line number "27:2:761".

Task 2: Display All Students

In this task I have Display all the record from database using “Select Query” through java application.I creaded one class **TotalStudent** ,in which I created **displayAllStudent()** method In this method I write the logic for printing Student records from database .

- 1.Created connection by calling method **myDBgetConnection()** which contains connection code.
- 2.Used **Statement Interface** for creating statement
- 3 execucute the query using **executeQuery()**;
- 4.storeed record in **resultSet rs**
- 5.Retrieve records from **rs** using **getInt/getString** method and stored in **object** of type **TotalStudent**.
- 6.Created the reference of **List<Total Student> stu** and created **Object of ArrayList** --(Used Generic Collection)
- 7.Stored the **TotalStudent** type **object** into **ArrayList stu** using **add()** method of collection.
- 8.closed connection using **close()** method.
9. Returned **ArrayList stu**.
- 10.Call **displayAllStudent()** in main method.
- 11.Print record from **ArrayList stu** using **Inhance for Loop**

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

- Project Explorer:** Shows various Java projects and source files under the "com.acc.task2" package.
- TotalStudent.java:** The active file contains Java code for a **TotalStudent** class, specifically the **displayAllStudents()** method. The code uses JDBC to connect to a MySQL database, execute a select query, and store the results in an **ArrayList** of **TotalStudent** objects.
- Terminal:** The "Console" tab shows the output of the application running. It displays the heading "Total Students" followed by a table with two rows of student data.
- Table Data:**

ID	NAME	AGE	MARKS
1	Madhavi	20	82
2	Anushka	15	95

Task 3: Insert Student Record

In this task I have insert the student Details into the database. through the java application

Using “insert Query”.I created one class **InsertStudentRecord** in this class I created one method **stuInfoInsertion()** containing the logic of Inert record into database as follow.

- 1.Created connection by calling method **myDBgetConnection()** which contains connection code.
- 2.create statement using **PreparedStatement psmt**
- 3.call **preparedStatement()** and pass insert query as argument.
- 4.take value from user using **Scanner** object
- 5.stored that value using **setInt()** method into **psmt** and pass to insert query .
- 6.execute the statement using **executeUpdate()** .
7. closed connection using **close()** method.
- 8.call **stuInfoInsertion()** in main method and print Insetion successfull message.

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

- Project Explorer:** Shows various Java projects and files, including `MysqlDBConnection.java`, `TotalStudents.java`, `InsertStudentRecord.java`, `UpdateStudent.java`, `DBOperation.java`, `CollectionExample.java`, `*Student.java`, `DB.properties`, and `PropertyLoader.java`.
- Code Editor:** Displays the `InsertStudentRecord.java` file with the following code:14 int age;
15 int marks;
16 public static void stuInfoInsertion() {
17 Connection con=null;
18 con=MysqlDBConnection.myDBgetConnection();
19 PreparedStatement psmt=null;
20 InsertStudentRecord isr=new InsertStudentRecord();
21 try {
22 psmt=con.prepareStatement("insert into Students(id,name,age,marks)values(?, ?, ?, ?)");
23 psmt.setInt(1, isr.id);
24 psmt.setString(2, isr.name);
25 psmt.setInt(3, isr.age);
26 psmt.setInt(4, isr.marks);
27 psmt.executeUpdate();
28 } catch (SQLException e) {
29 e.printStackTrace();
}
- Console Output:** Shows the execution of the code and the resulting output:

```
Insert Student Details
Enter Student id
3
Enter Student Name
Himanshu
Enter Student Age
21
Enter Student marks
63
Student Details Inserted Successfully
```

Task 4: Update Student Marks

In this task I have update the student **marks** by student **id** using “update query” through java application. I have created one class **UpdateStudentRecord** in which I created **stuInfoupdation()** method containing logic for updating record of database as follow.

- 1.Created connection by calling method **myDBgetConnection()** which contains connection code.
- 2.create statement using **PreparedStatement psmt**
- 3.take value of id from user using **Scanner** object
- 4.call **preparedStatement()** and pass update query as argument.
- 5.execute the statement using **executeUpdate()** .
6. closed connection using **close()** method.
- 8.call **stuInfoupdation()** in main method and print updatation successfull message.

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

- Project Explorer:** Shows various Java projects and files, including `AbstractClassDemo`, `BookManagementSystem`, `CloneableExampleDemo`, `CollectionExampleDemo`, `ExceptionJavaProject`, `EmployeeManagementSystem`, `ExceptionHandlingDemo`, `FruitSystem`, `HelloDemo`, `InterfaceExampleDemo`, `IOExampleDemo`, `JUnitTesting`, `ManageLeaveRow`, `RI`, `RIAssignment`, `Rispractice`, `StudentRecordManagement`, and `IRE System Library [JRE]`.
- Code Editor:** Displays the Java code for the `UpdateStudentRecord` class, specifically the `stuInfoupdation()` method. The code uses a `PreparedStatement` to update student marks based on their ID.
- Console Output:** Shows the terminal window output where the user enters the student ID (75) and the updated marks (75), followed by the confirmation message "Student Details Updated Successfully".

Task 5: Delete Student Record

In this task I have delete the student record by student **id** using “delete query” through java application. I have created one class **DeleteStudentRecord** in which I created **stuInfoDeletion()** method containing logic for deleting record from database as follow.

- 1.Created connection by calling method **myDBgetConnection()** which contains connection code.
- 2.create statement using **PreparedStatement psmt**
- 3.take value of id from user using **Scanner** object
- 4.call **preparedStatement()** and pass delete query as argument.
- 5.execute the statement using **executeUpdate()** .
6. closed connection using **close()** method.
- 8.call **stuInfoDeletion()** in main method and print updation successfull message.

The screenshot shows the Eclipse IDE interface with the Java code for the **DeleteStudentRecord** class. The code implements a static method **stuInfoDeletion()** that takes a student ID as input, prepares and executes a DELETE query, and then closes the connection. The code is annotated with line numbers. Below the code editor, the Java console displays the execution of the program, showing the prompt "Enter Student id for deleting Student record", the user input "3", and the success message "Student Details Deleted Successfully".

```
File Edit Source Refactor Navigate Search Project Run Window Help
MysqlDBConne... TotalStuden... InsertStuden... UpdateStude... DeleteStude... DB properties DBOperation... DB.properties PropertyLoad... PropertyLoad...
10 public class DeleteStudentRecord {  
11     public static void stuInfoDeletion() {  
12         System.out.println("Enter Student id for deleting Student record");  
13         Scanner sc=new Scanner(System.in);  
14         int id =sc.nextInt();  
15         Connection con=null;  
16         con=MysqlDBConnection.myDBgetConnection();  
17         PreparedStatement psmt=null;  
18         try {  
19             psmt=con.prepareStatement("Delete from students where id=?");  
20             psmt.setInt(1, id);  
21             psmt.executeUpdate();  
22         } catch (SQLException e) {  
23             e.printStackTrace();  
24         } finally {  
25             try {  
26                 if(con!=null)  
27                 {  
28                     con.close();  
29                 }  
30             } catch (SQLException e) {  
31                 e.printStackTrace();  
32             }  
33         }  
34     }  
35 }
```

Javadoc Declaration Console × Debug Call Hierarchy
terminated: DeleteStudentRecord (1) [Java Application] C:\Users\madha\p2\jpoof\plugins\org.eclipse.jdt.core\open\jdk\hotspot\jre\full\win32\x86_64_21.0.9.v20251105-0741\jre\bin\javaw.exe (08-Jan-2026, 9:26:05 pm - 9:26:12 pm elapsed 0:00:07) [pid: 1]
Enter Student id for deleting Student record
3
Student Details Deleted Successfully