

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.
- Code Editor:** The main window displays the Java code for **MysqlDBConnection**. The code includes a static method **myDBgetConnection()** that uses **DriverManager.getConnection()** to establish a connection. It also contains a **main** method that calls **myDBgetConnection()** and prints "Connection Successfull..!" to the console.
- Console Tab:** Shows the output of the application's execution. It displays the message "Connection Successfull..!" followed by the Java runtime information: "terminated> MysqlDBConnection(1) [Java Application] C:\Users\madha\p2\pool\plugins\org.eclipse.jst\openjdk\hotspot\ire\full\win32\x86_64_21.0.9.v20251105-0741\ire\bin\javaw.exe (08-Jan-2026, 9:06:02 pm - 9:06:03 pm elapsed 00000) [pid: 11444]".
- 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 several Java projects and source files under the "com.acc.task2" package, including AbstractClassDemo, BookManagementSystem, ClonableExampleDemo, CollectionExampleDemo, CoreJavaProject, DataBaseConnectivity, EmpMgtSystem, ExceptionHandlingDemo, FruitSystem, HelloDemo, InterfaceExampleDemo, OExampleDemo, UnitTesting, MarriageBeuron, RSAssignment, Rpractice, and StudentRecordManagement.
- TotalStudent.java:** The active editor displays the following Java code:

```
15 int age;
16 int marks;
17 public static List<TotalStudent> displayAllStudents() {
18     Connection con=null;
19     con=MysqlDBConnection.myDBgetConnection();/* call method having create connection code */
20     Statement stmt=null;
21     ResultSet rs=null;
22     List<TotalStudent> stu=new ArrayList<TotalStudent>(); /* List reference stu points to ArrayList*/
23     try {
24         stmt=con.createStatement();
25         rs=stmt.executeQuery("select * from Students");
26         /*stores result of select query into ResultSet rs*/
27         while(rs.next()) {
28             stu.add(new TotalStudent(/*get the column record from ResultSet*/rs.getInt("id"),rs.getString("name"),rs
29             /* add the object into ArrayList having reference stu*/
30         }
31     }
32 }
```

- Console:** The terminal window shows the output of the program:

```
Total Students
| 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`, `TotalStudents`, `InsertStudentRecord`, `UpdateStudent`, `DBOperation`, `CollectionExample`, `*Student.java`, `DB properties`, and `PropertyLoader`.
- 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`, `CloneableExampleDemo`, `CollectionExampleDemo`, `JavaProject`, `DatabaseConnectivity`, `EmpMgtSystem`, `ExceptionHandlingDemo`, `FruitSystem`, `HelloDemo`, `InterfaceExampleDemo`, `IoExampleDemo`, `JUnitTesting`, `ManageLeave`, `ri`, `riAssignment`, `Rpractice`, `StudentRecordManagement`, `IRE System Library [JRE]`, and `src` (containing `com.acc.task1`, `com.acc.task2`, `com.acc.task3`, `com.acc.task4`, `com.acc.task5`, and `com.loader` packages).
- Code Editor:** Displays Java code for updating student marks. The code uses `PreparedStatement` to execute an update query. It includes exception handling for `SQLException`.
- Console:** Shows the command-line output of the application's execution. It prompts for a student ID, receives input '75', and then displays the 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 prompts the user for a student ID, prepares and executes a delete SQL statement, and then closes the connection. The code uses **Scanner** for input and **PreparedStatement** for the delete query. The execution output window shows the command **java DeleteStudentRecord** being run, followed by the user input **Enter Student id for deleting Student record**, the ID **3**, and the confirmation 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     12●     public static void stuInfoDeletion() {  
13         System.out.println("Enter Student id for deleting Student record");  
14         Scanner sc=new Scanner(System.in);  
15         int id =sc.nextInt();  
16         Connection con=null;  
17         con=MysqlDBConnection.myDBgetConnection();  
18         PreparedStatement psmt=null;  
19●         try {  
20             psmt=con.prepareStatement("Delete from students where id=?");  
21             psmt.setInt(1, id);  
22             psmt.executeUpdate();  
23●             } catch (SQLException e) {  
24                 e.printStackTrace();  
25●             } finally {  
26                 try {  
27                     if(con!=null)  
28                         {  
29                             con.close();  
30                         }  
31             } catch (SQLException e) {  
32                 e.printStackTrace();  
33             }  
34         }  
35     }  
36 }
```

Javadoc Declaration Console × Debug Call Hierarchy
terminated: DeleteStudentRecord (1) [Java Application] C:\Users\mada\p2\pool\plugins\org.eclipse.jdt.core\openjdk\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