

1. Easy Level: Create a Java program to connect to a MySQL database and fetch data from a single table.

Code: import java.sql.\*;

```
public class DatabaseConnect {  
    public static void main(String[] args) {  
        String url = "jdbc:mysql://localhost:3306/test";  
        String user = "root";  
        String password = "password";  
  
        try {  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            Connection con = DriverManager.getConnection(url, user,  
password);  
            System.out.println("Connected to the database!");  
  
            Statement stmt = con.createStatement();  
            ResultSet rs = stmt.executeQuery("SELECT * FROM  
students");  
  
            System.out.println("ID\tName\tMarks");  
            while (rs.next()) {
```

```

int id = rs.getInt("id");
String name = rs.getString("name");
int marks = rs.getInt("marks");
System.out.println(id + "\t" + name + "\t" + marks);
}

con.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

## 2. Medium Level: Build a program to perform CRUD operation

Code: import java.sql.\*;  
import java.util.Scanner;

```

public class StudentCRUD {
    static final String URL =
"jdbc:mysql://localhost:3306/test";
    static final String USER = "root";
    static final String PASSWORD = "password";

    public static void main(String[] args) {

```

```

Scanner sc = new Scanner(System.in);
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection con =
DriverManager.getConnection(URL, USER, PASSWORD);

    while (true) {
        System.out.println("\n1. Insert\n2. Display\n3.
Update\n4. Delete\n5. Exit");
        System.out.print("Choose: ");
        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter ID: ");
                int id = sc.nextInt();
                sc.nextLine();
                System.out.print("Enter Name: ");
                String name = sc.nextLine();
                System.out.print("Enter Marks: ");
                int marks = sc.nextInt();

                PreparedStatement insert =
con.prepareStatement("INSERT INTO students VALUES
(?, ?, ?)");
                insert.setInt(1, id);
                insert.setString(2, name);
                insert.setInt(3, marks);
                insert.executeUpdate();

```

```
System.out.println("Inserted successfully!");  
break;
```

case 2:

```
Statement stmt = con.createStatement();  
ResultSet rs = stmt.executeQuery("SELECT *  
FROM students");  
System.out.println("ID\tName\tMarks");  
while (rs.next()) {  
    System.out.println(rs.getInt(1) + "\t" +  
rs.getString(2) + "\t" + rs.getInt(3));  
}  
break;
```

case 3:

```
System.out.print("Enter ID to update: ");  
int uid = sc.nextInt();  
sc.nextLine();  
System.out.print("Enter new Name: ");  
String uname = sc.nextLine();  
System.out.print("Enter new Marks: ");  
int umarks = sc.nextInt();
```

```
PreparedStatement update =  
con.prepareStatement("UPDATE students SET name=?,  
marks=? WHERE id=?");  
update.setString(1, uname);  
update.setInt(2, umarks);  
update.setInt(3, uid);
```

```
update.executeUpdate();  
System.out.println("Updated successfully!");  
break;
```

case 4:

```
System.out.print("Enter ID to delete: ");  
int did = sc.nextInt();  
PreparedStatement delete =  
con.prepareStatement("DELETE FROM students WHERE  
id=?");  
delete.setInt(1, did);  
delete.executeUpdate();  
System.out.println("Deleted successfully!");  
break;
```

case 5:

```
con.close();  
sc.close();  
System.exit(0);
```

default:

```
System.out.println("Invalid choice!");
```

```
}
```

```
}
```

```
} catch (Exception e) {  
    e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

### 3. Hard Level: Develop a Java application using JDBC and MVC architecture to manage student data

Code:

```
import java.util.*;

public class StudentController {
    private StudentDAO dao;
    private StudentView view;

    public StudentController(StudentDAO dao,
StudentView view) {
        this.dao = dao;
        this.view = view;
    }

    public void start() throws Exception {
        while (true) {
            int choice = view.showMenu();

            switch (choice) {
                case 1:
                    Student s1 = view.getStudentDetails();
                    dao.insertStudent(s1);
                    view.showMessage("Student added.");
                    break;

                case 2:
                    List<Student> list = dao.getAllStudents();
```

```
view.displayStudents(list);  
break;
```

case 3:

```
int id = view.getId();  
view.showMessage("Enter new details:");  
Student s2 = view.getStudentDetails();  
s2 = new Student(id, s2.getName(), s2.getMarks());  
dao.updateStudent(s2);  
view.showMessage("Student updated.");  
break;
```

case 4:

```
int delId = view.getId();  
dao.deleteStudent(delId);  
view.showMessage("Student deleted.");  
break;
```

case 5:

```
dao.close();  
System.exit(0);  
break;
```

default:

```
view.showMessage("Invalid choice!");
```

```
}
```

```
}
```

```
}
```

```
}
```