EXPERIMENT: 07

```
import java.sql.*;

import java.util.Scanner;
```

Easy Level: Connect to MySQL and fetch employee data

```
class EasyJDBC {

   public static void main(String[] args) {

      try {

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

         Statement stmt = con.createStatement();

         ResultSet rs = stmt.executeQuery("SELECT * FROM Employee");

         while (rs.next()) {

            System.out.println(rs.getInt("EmpID") + " " + rs.getString("Name") + " " +
rs.getDouble("Salary"));

         }

         con.close();

      } catch (Exception e) {

         e.printStackTrace();

      }

   }

}
```

Medium Level: CRUD operations on Product table

```
class MediumJDBC {
   static Connection getConnection() throws SQLException {
      return DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase",
"root", "password");
   }

   public static void main(String[] args) {
      try (Connection con = getConnection(); Scanner sc = new Scanner(System.in)) {
         while (true) {
            System.out.println("1. Insert 2. Read 3. Update 4. Delete 5. Exit");
            int choice = sc.nextInt();
            switch (choice) {
               case 1:
                  System.out.print("Enter ProductID, Name, Price, Quantity: ");
                  int id = sc.nextInt();
                  String name = sc.next();
                  double price = sc.nextDouble();
                  int qty = sc.nextInt();
                  PreparedStatement ps = con.prepareStatement("INSERT INTO Product VALUES
(?, ?, ?, ?)");
                  ps.setInt(1, id);
                  ps.setString(2, name);
                  ps.setDouble(3, price);
                  ps.setInt(4, qty);
                  ps.executeUpdate();
                  break;
```

```
                case 2:
                    Statement stmt = con.createStatement();
                    ResultSet rs = stmt.executeQuery("SELECT * FROM Product");
                    while (rs.next()) {
                        System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getDouble(3) +
" " + rs.getInt(4));
                    }
                    break;
                case 3:
                    System.out.print("Enter ProductID to update Price: ");
                    id = sc.nextInt();
                    System.out.print("Enter new Price: ");
                    price = sc.nextDouble();
                    ps = con.prepareStatement("UPDATE Product SET Price = ? WHERE ProductID =
?");
                    ps.setDouble(1, price);
                    ps.setInt(2, id);
                    ps.executeUpdate();
                    break;
                case 4:
                    System.out.print("Enter ProductID to delete: ");
                    id = sc.nextInt();
                    ps = con.prepareStatement("DELETE FROM Product WHERE ProductID = ?");
                    ps.setInt(1, id);
                    ps.executeUpdate();
                    break;
                case 5:
                    return;
            }
```

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

Hard Level: MVC-based Student Management System

```
class Student {
    int studentID;
    String name;
    String department;
    int marks;
}


class StudentDAO {
    static Connection getConnection() throws SQLException {
        return DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase",
"root", "password");
    }


    void insertStudent(Student s) throws SQLException {
        Connection con = getConnection();
        PreparedStatement ps = con.prepareStatement("INSERT INTO Student VALUES (?, ?, ?,
?)");
        ps.setInt(1, s.studentID);
        ps.setString(2, s.name);
        ps.setString(3, s.department);
```

```
        ps.setInt(4, s.marks);
        ps.executeUpdate();
        con.close();
    }
}

class StudentController {
    public static void main(String[] args) {
        StudentDAO dao = new StudentDAO();
        Student s = new Student();
        s.studentID = 1;
        s.name = "John";
        s.department = "CSE";
        s.marks = 85;
        try {
            dao.insertStudent(s);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```