

EXPERIMENT – 5

CODE:

```
package name;

import java.sql.*;

public class Mydb {

    private static final String URL = "jdbc:mysql://localhost:3306/amitdb";

    private static final String USER = "root";

    private static final String PASSWORD = "Deep9262@J";

    private Connection connection;

    public Mydb() {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            connection = DriverManager.getConnection(URL, USER, PASSWORD);

            System.out.println(" Database Connected Successfully!");

        } catch (ClassNotFoundException e) {

            System.err.println("JDBC Driver not found! Add MySQL Connector JAR.");

        } catch (SQLException e) {

            System.err.println(" Connection failed! Check credentials. Error: " + e.getMessage());

        }

    }

    public void getAllEmployees() {

        String query = "SELECT * FROM employee";

        try (Statement stmt = connection.createStatement();

            ResultSet rs = stmt.executeQuery(query)) {

            System.out.println("\n Employee List:");

            while (rs.next()) {
```

```
        System.out.println("ID: " + rs.getInt("id") +
            ", Name: " + rs.getString("name") +
            ", Salary: " + rs.getDouble("salary"));
    }
} catch (SQLException e) {
    System.err.println(" Error fetching employees: " + e.getMessage());
}
}

public void addEmployee(int id,String name, double salary) {
    String query = "INSERT INTO employee (id,name, salary) VALUES (?,?, ?)";
    try (PreparedStatement stmt = connection.prepareStatement(query)) {
        stmt.setInt(1, id);
        stmt.setString(2, name);
        stmt.setDouble(3, salary);
        stmt.executeUpdate();
        System.out.println("Employee added successfully: " + name);
    } catch (SQLException e) {
        System.err.println(" Error adding employee: " + e.getMessage());
    }
}

public void removeDuplicateEmployees() {
    String query = "DELETE e1 FROM employee e1 " +
        "INNER JOIN employee e2 " +
        "ON e1.name = e2.name AND e1.salary = e2.salary " +
        "WHERE e1.id > e2.id";
    try (Statement stmt = connection.createStatement()) {
        int rowsDeleted = stmt.executeUpdate(query);
        System.out.println(" " + rowsDeleted + " duplicate records removed.");
    }
}
```

```
        } catch (SQLException e) {
            System.err.println(" Error removing duplicates: " + e.getMessage());
        }
    }

    public void closeConnection() {
        if (connection != null) {
            try {
                connection.close();

                System.out.println(" Database Connection Closed.");
            } catch (SQLException e) {
                System.err.println(" Error closing connection: " + e.getMessage());
            }
        }
    }

    public static void main(String[] args) {
        Mydb db = new Mydb();
        db.addEmployee(109,"James", 5000);
        db.addEmployee(110,"Harry Potter", 6000);
        db.addEmployee(111,"John cena", 5500);
        db.addEmployee(112,"Rahul", 8000);
        db.addEmployee(113,"Ram", 5200);
        db.addEmployee(140,"Devi", 6000);
        db.getAllEmployees();
        db.removeDuplicateEmployees();
        db.getAllEmployees();
        db.closeConnection();
    }
}
```

DATABASE:

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' pane on the left displays the 'amitdb' database with tables 'employee' and 'employee2'. The 'Query' pane shows a SQL query: `SELECT * FROM amitdb.employee;`. The 'Result Grid' displays the following data:

id	name	salary
109	Harsh1	5500.00
110	Sartha1k	8000.00
111	John cena	5500.00
112	Rahul	8000.00
113	Ram	5200.00
140	Devi	6000.00
160	Aman1	5200.00

The 'Output' pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
12	09:59:19	select * from employee2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
13	09:59:22	select * from employee2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
14	09:59:36	SELECT * FROM amitdb.employee2 LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec
15	10:03:10	SELECT * FROM amitdb.employee2 LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec
16	10:03:13	SELECT * FROM amitdb.employee LIMIT 0, 1000	12 row(s) returned	0.015 sec / 0.000 sec
17	14:35:23	SELECT * FROM amitdb.employee LIMIT 0, 1000	14 row(s) returned	0.000 sec / 0.000 sec

OUTPUT :

The screenshot shows an IDE with a Java file named 'input.jsp'. The code is as follows:

```
71 connection.close();
72 System.out.println("Connection closed.");
73 } catch (SQLException e) {
74     System.err.println("SQLException: " + e.getMessage());
75 }
76 }
77 }
78 }
79 }
```

The 'Console' pane shows the following output:

```
<terminated> Mydb [Java Application] C:\Users\DELL\p2\p...
ID: 108, Name: Krishna1, Salary: 6000.0
ID: 109, Name: Harsh1, Salary: 5500.0
ID: 110, Name: Sartha1k, Salary: 8000.0
ID: 111, Name: John cena, Salary: 5500.0
ID: 112, Name: Rahul, Salary: 8000.0
ID: 113, Name: Ram, Salary: 5200.0
ID: 140, Name: Devi, Salary: 6000.0
ID: 160, Name: Aman1, Salary: 5200.0
0 duplicate records removed.

Employee List:
ID: 101, Name: John Doe, Salary: 50000.0
ID: 102, Name: Krishna, Salary: 60000.0
ID: 103, Name: Harsh, Salary: 55000.0
ID: 104, Name: Sarthak, Salary: 58000.0
ID: 105, Name: Aman, Salary: 52000.0
ID: 107, Name: John1 Doe, Salary: 5000.0
ID: 108, Name: Krishna1, Salary: 6000.0
ID: 109, Name: Harsh1, Salary: 5500.0
ID: 110, Name: Sartha1k, Salary: 8000.0
ID: 111, Name: John cena, Salary: 5500.0
ID: 112, Name: Rahul, Salary: 8000.0
ID: 113, Name: Ram, Salary: 5200.0
ID: 140, Name: Devi, Salary: 6000.0
ID: 160, Name: Aman1, Salary: 5200.0
Database Connection Closed
```