Main.java

```
package jdbcexamplea;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class main {
public static void main(String[] args) {
// Add employees
employeeDAO.addEmployee("Alice Cooper", "Developer", 70000);
employeeDAO.addEmployee("Bob Marley", "Manager", 80000);

// Update employee

employeeDAO.updateEmployee( 1,"John Doe", "Senior Software Engineer", 90000);
// Get all employees
List<Employee> employees = employeeDAO.getAllEmployees();
employees.forEach(System.out::println);
// Delete employee
employeeDAO.deleteEmployee(2);
}
}
```

employeeDAO.java

```
package jdbcexamplea;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class employeeDAO {
   // Create an employee
public static void addEmployee(String name, String position, double salary) {
String sql = "INSERT INTO employees (name, position, salary) VALUES(?, ?, ?)";
try (Connection conn = databaseconnec.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql)) {
stmt.setString(1, name);
stmt.setString(2, position);
stmt.setDouble(3, salary);
int rowsAffected = stmt.executeUpdate();
System.out.println("Employee added successfully. Rows affected: "+ rowsAffected);
} catch (SQLException e) {
e.printStackTrace();
}
}
public static List <Employee> getAllEmployees() {
List<Employee> employees = new ArrayList<>();
String sql =("SELECT * FROM employees") ;
try (Connection conn = databaseconnec.getConnection();
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(sql)) {
while (rs.next()) {
Employee employee = new Employee(
rs.getInt("id"),
```

```
rs.getString("name"),
rs.getString("position"),
rs.getDouble("salary")
);
employees.add(employee);
}
} catch (SQLException e) {
e.printStackTrace();
}
return employees;
}
// Update an employee's information
public  static void updateEmployee(int id, String name, String position, double salary) {
   String sql = "UPDATE employees SET name = ?, position = ?, salary = ? WHERE id = ?"; // Corrected SQL
   try (Connection conn = databaseconnec.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {
       stmt.setString(1, name);
       stmt.setString(2, position);
       stmt.setDouble(3, salary);
       stmt.setInt(4, id);
       int rowsAffected = stmt.executeUpdate();
       System.out.println("Employee updated successfully. Rows affected: " + rowsAffected);
   } catch (SQLException e) {
       e.printStackTrace();
   }
}
// Delete an employee
public static void deleteEmployee(int id) {
String sql = "DELETE FROM employees WHERE id = ?";
try (Connection conn = databaseconnec.getConnection();
```

```
PreparedStatement stmt = conn.prepareStatement(sql)) {
stmt.setInt(1, id);
int rowsAffected = stmt.executeUpdate();
System.out.println("Employee deleted successfully. Rows affected: " + rowsAffected);
} catch (SQLException e) {
e.printStackTrace();
}
}
}
```

Databaseconnec.java

```
package jdbcexamplea;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;



public class databaseconnec {
private static final String URL ="jdbc:mysql://localhost:3306/employee_db"; // Database URL
private static final String USER = "root"; // Your MySQL username
private static final String PASSWORD = ""; // Your MySQL password
public static Connection getConnection() throws SQLException {
try {
// Load the JDBC driver
Class.forName("com.mysql.cj.jdbc.Driver");
// Return the database connection
return DriverManager.getConnection(URL, USER, PASSWORD);
} catch (ClassNotFoundException | SQLException e) {
System.out.println("Connection failed:" + e.getMessage());
```

```
throw new SQLException("Failed to establish connection");
}
}
}
```

Employee.java

```
package jdbcexamplea;

public class Employee {
private int id;
private String name;
private String position;
private double salary;
public Employee(int id, String name, String position, double salary) {
this.id = id;
this.name = name;
this.position = position;
this.salary = salary;
}
// Getters and setters
public int getId() { return id; }
public void setId(int id) { this.id = id; }
public String getName() { return name; }
public void setName(String name) { this.name = name; }
public String getPosition() { return position; }
public void setPosition(String position) { this.position = position; }
public double getSalary() { return salary; }
public void setSalary(double salary) { this.salary = salary; }
@Override
public String toString() {
```

```
    return "Employee{id=" + id + ", name='" + name + "', position='" + position + "', salary=" + salary +
'}';

}

}
```

Output:

	id	name	position	salary
Edit Copy Delete	1	John Doe	Senior Software Engineer	90000.00
Edit Copy Delete	3	Steve Brown	Team Lead	85000.00
Edit Copy Delete	4	Alice Cooper	Developer	70000.00
Edit Copy Delete	5	Bob Marley	Manager	80000.00
Edit Copy Delete	6	Alice Cooper	Developer	70000.00
Edit Copy Delete	7	Bob Marley	Manager	80000.00