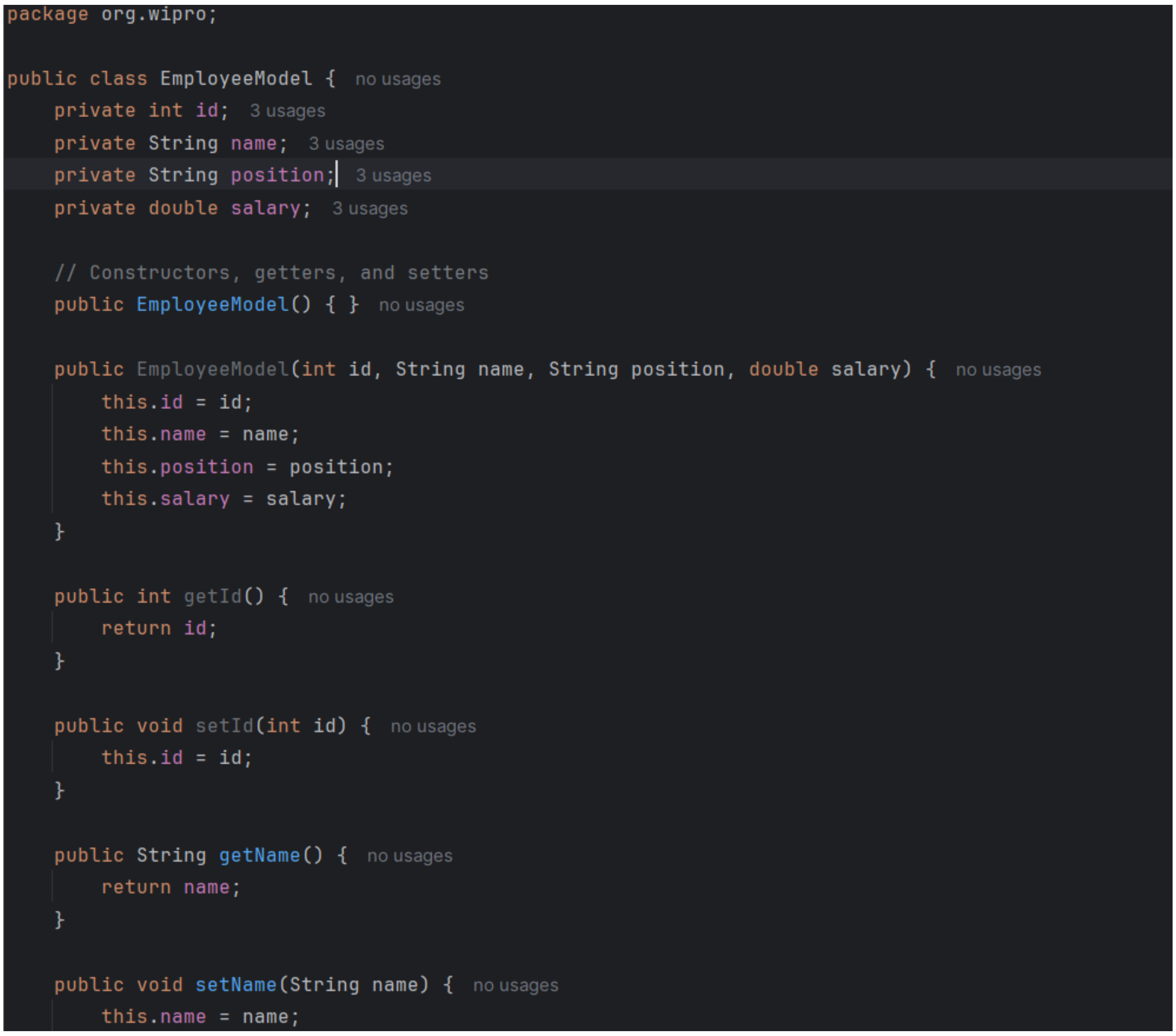
Assignment:12

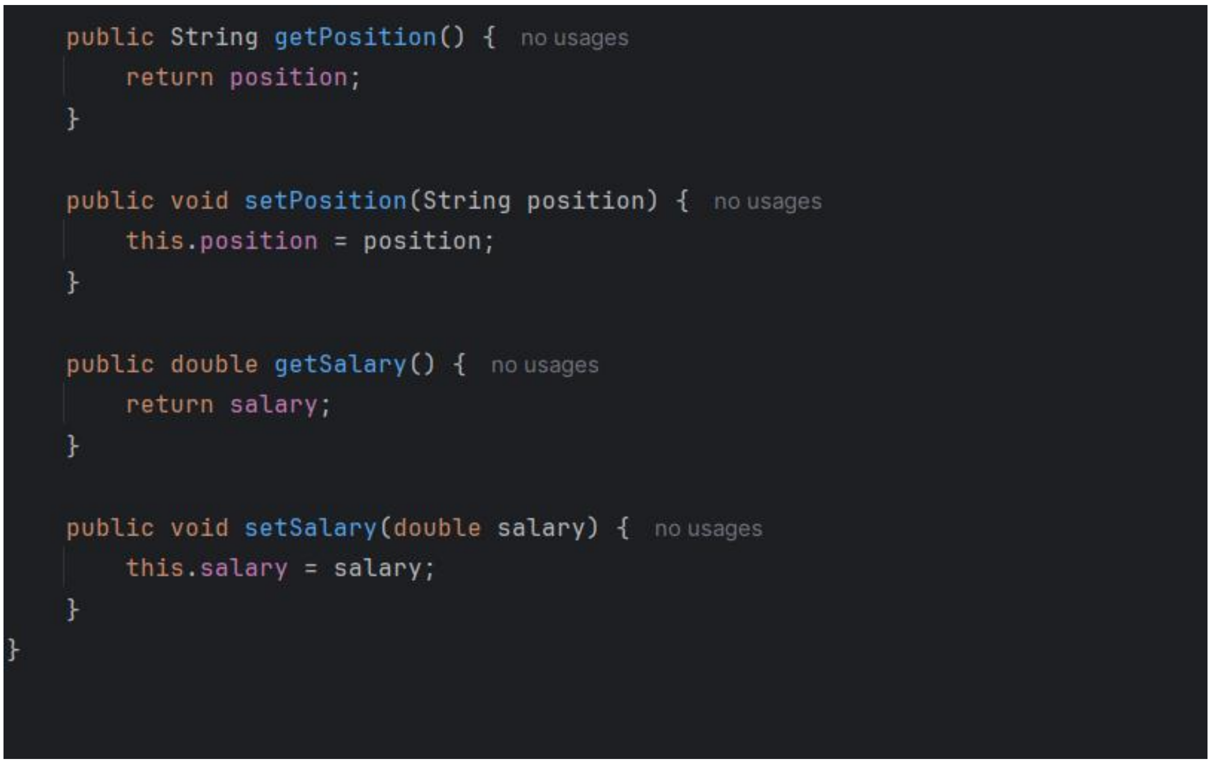
- **Create a Java application to perform basic CRUD operations on a database.**
- **Create a DAO layer to interact with Oracle database**

CODE:

EmployeeModel:

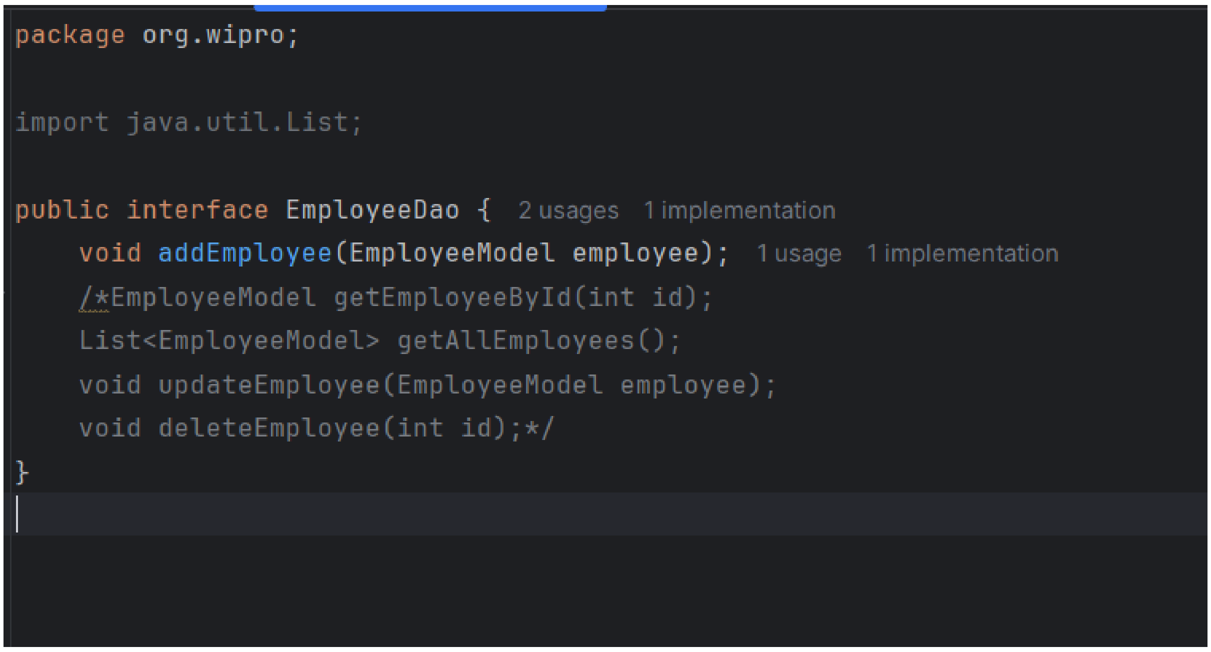


```
package org.wipro;

public class EmployeeModel {  no usages
    private int id;  3 usages
    private String name;  3 usages
    private String position;  3 usages
    private double salary;  3 usages

    // Constructors, getters, and setters
    public EmployeeModel() { }  no usages

    public EmployeeModel(int id, String name, String position, double salary) {  no usages
        this.id = id;
        this.name = name;
        this.position = position;
        this.salary = salary;
    }

    public int getId() {  no usages
        return id;
    }

    public void setId(int id) {  no usages
        this.id = id;
    }

    public String getName() {  no usages
        return name;
    }

    public void setName(String name) {  no usages
        this.name = name;
```



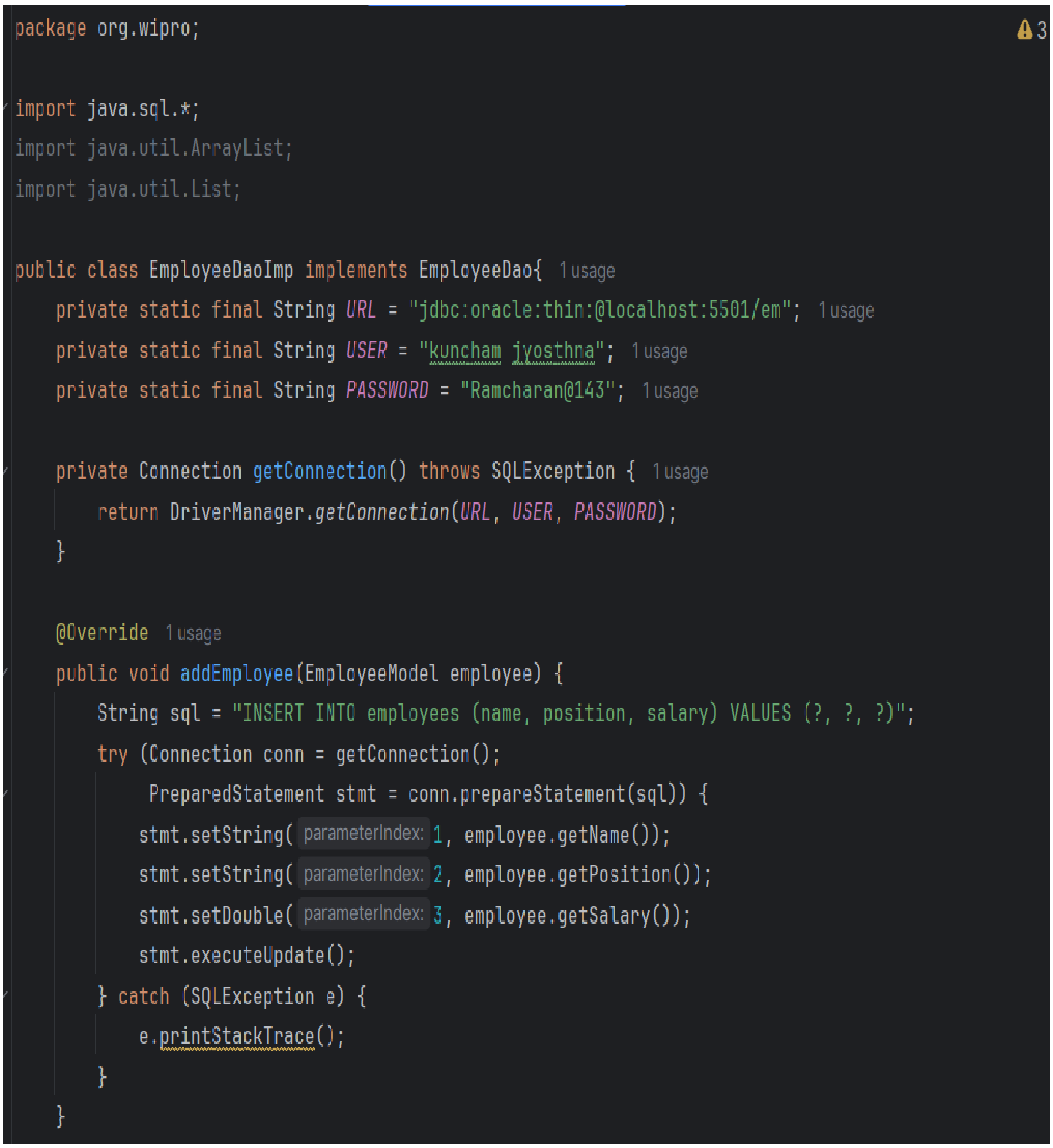
```
    public String getPosition() {  no usages
        return position;
    }

    public void setPosition(String position) {  no usages
        this.position = position;
    }

    public double getSalary() {  no usages
        return salary;
    }

    public void setSalary(double salary) {  no usages
        this.salary = salary;
    }
}
```

EmployeeDao:

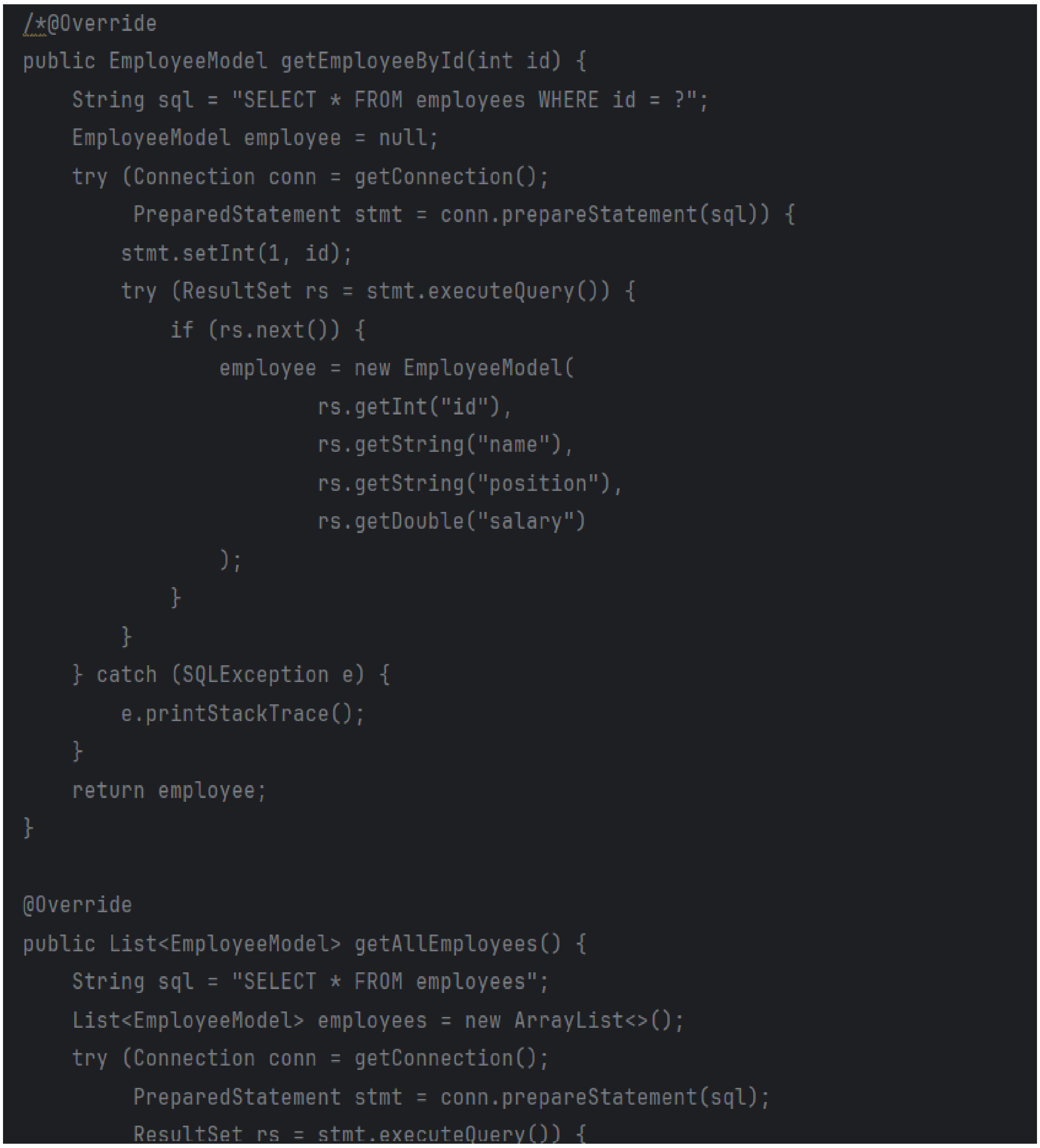


```
package org.wipro;

import java.util.List;

public interface EmployeeDao {  2 usages  1 implementation
    void addEmployee(EmployeeModel employee);  1 usage  1 implementation
    /*EmployeeModel getEmployeeById(int id);
    List<EmployeeModel> getAllEmployees();
    void updateEmployee(EmployeeModel employee);
    void deleteEmployee(int id);*/
}
```

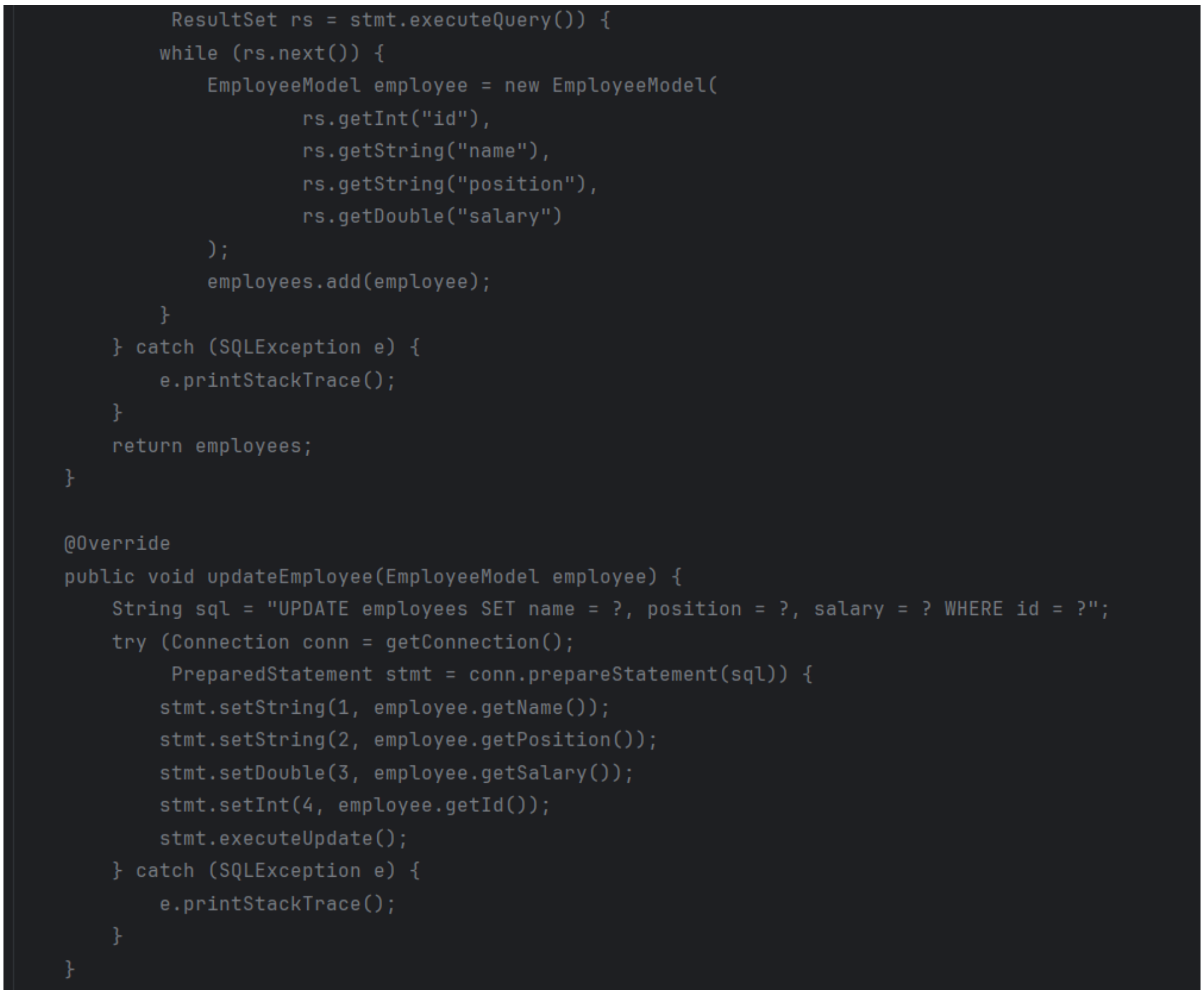
EmployeeeDaoImp:



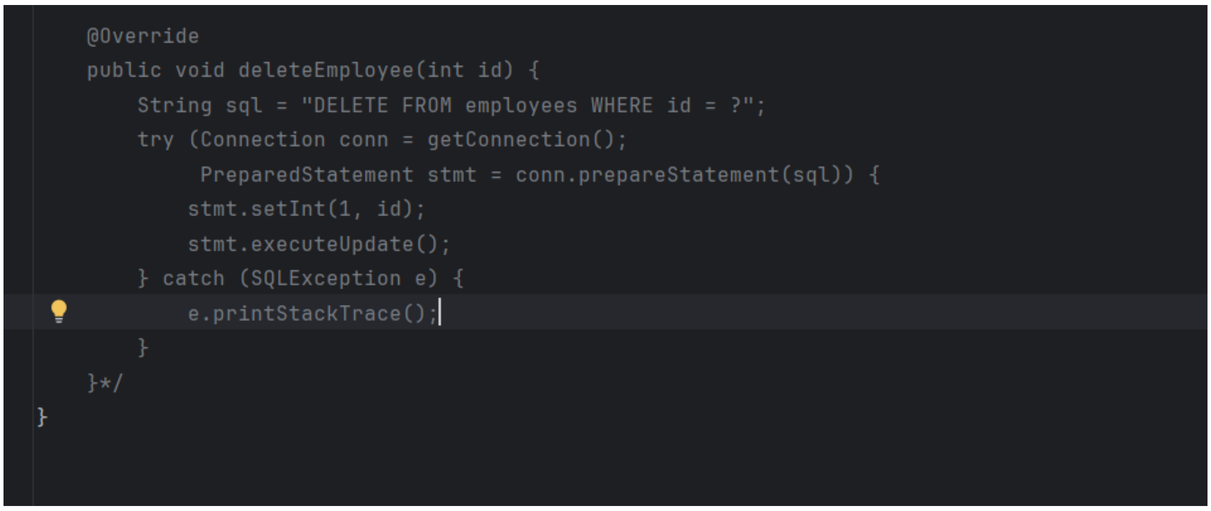
```
package org.wipro;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class EmployeeDaoImp implements EmployeeDao{  1 usage
    private static final String URL = "jdbc:oracle:thin:@localhost:5501/em";  1 usage
    private static final String USER = "kuncham jyosthna";  1 usage
    private static final String PASSWORD = "Ramcharan@143";  1 usage

    private Connection getConnection() throws SQLException {  1 usage
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }

    @Override  1 usage
    public void addEmployee(EmployeeModel employee) {
        String sql = "INSERT INTO employees (name, position, salary) VALUES (?, ?, ?)";
        try (Connection conn = getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString( parameterIndex: 1, employee.getName());
            stmt.setString( parameterIndex: 2, employee.getPosition());
            stmt.setDouble( parameterIndex: 3, employee.getSalary());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```



```
/*@Override
public EmployeeModel getEmployeeById(int id) {
    String sql = "SELECT * FROM employees WHERE id = ?";
    EmployeeModel employee = null;
    try (Connection conn = getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setInt(1, id);
        try (ResultSet rs = stmt.executeQuery()) {
            if (rs.next()) {
                employee = new EmployeeModel(
                        rs.getInt("id"),
                        rs.getString("name"),
                        rs.getString("position"),
                        rs.getDouble("salary")
                );
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return employee;
}

@Override
public List<EmployeeModel> getAllEmployees() {
    String sql = "SELECT * FROM employees";
    List<EmployeeModel> employees = new ArrayList<>();
    try (Connection conn = getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql);
         ResultSet rs = stmt.executeQuery()) {
```

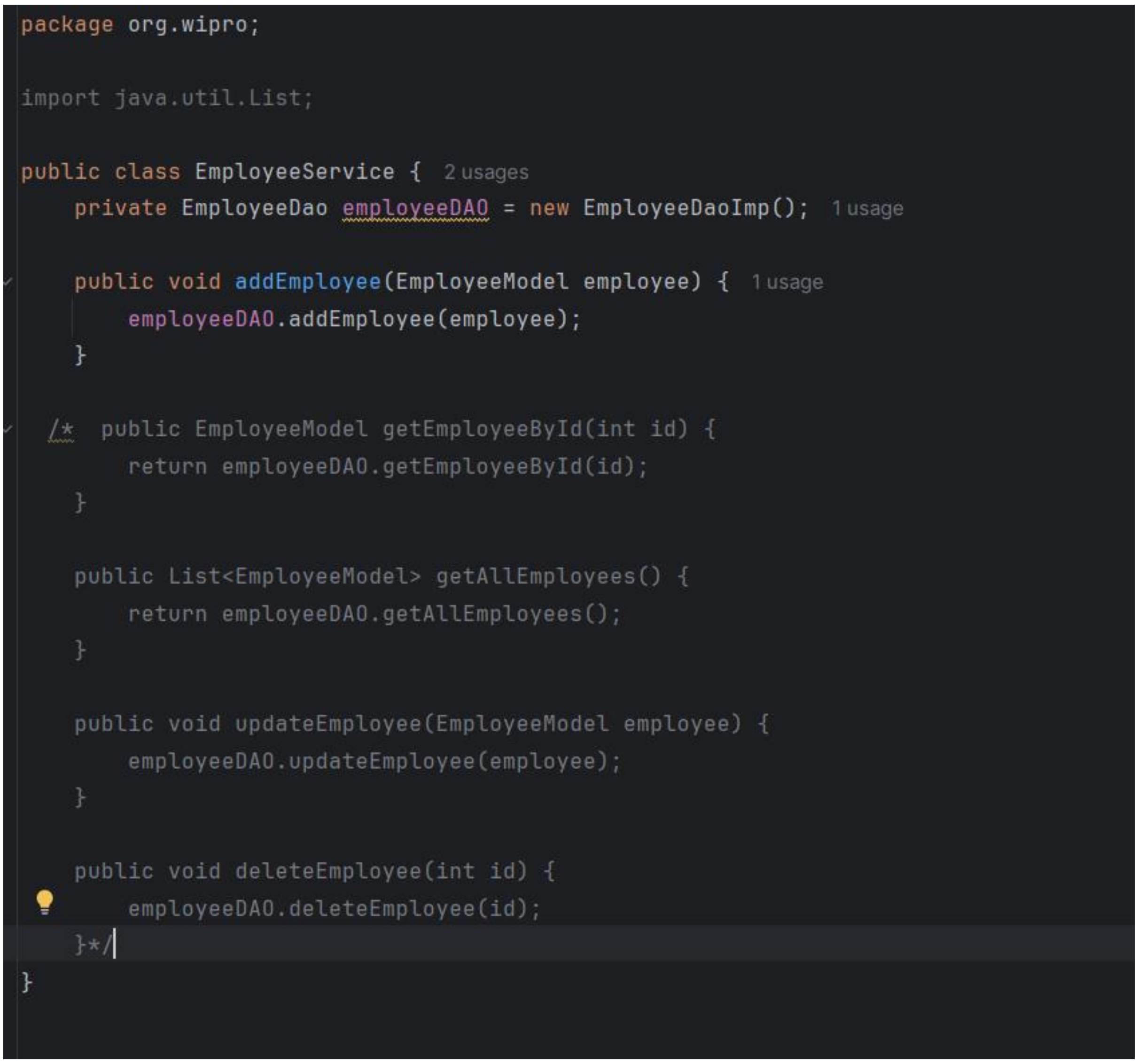


```
             ResultSet rs = stmt.executeQuery()) {
            while (rs.next()) {
                EmployeeModel employee = new EmployeeModel(
                        rs.getInt("id"),
                        rs.getString("name"),
                        rs.getString("position"),
                        rs.getDouble("salary")
                );
                employees.add(employee);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return employees;
    }

    @Override
    public void updateEmployee(EmployeeModel employee) {
        String sql = "UPDATE employees SET name = ?, position = ?, salary = ? WHERE id = ?";
        try (Connection conn = getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString(1, employee.getName());
            stmt.setString(2, employee.getPosition());
            stmt.setDouble(3, employee.getSalary());
            stmt.setInt(4, employee.getId());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```



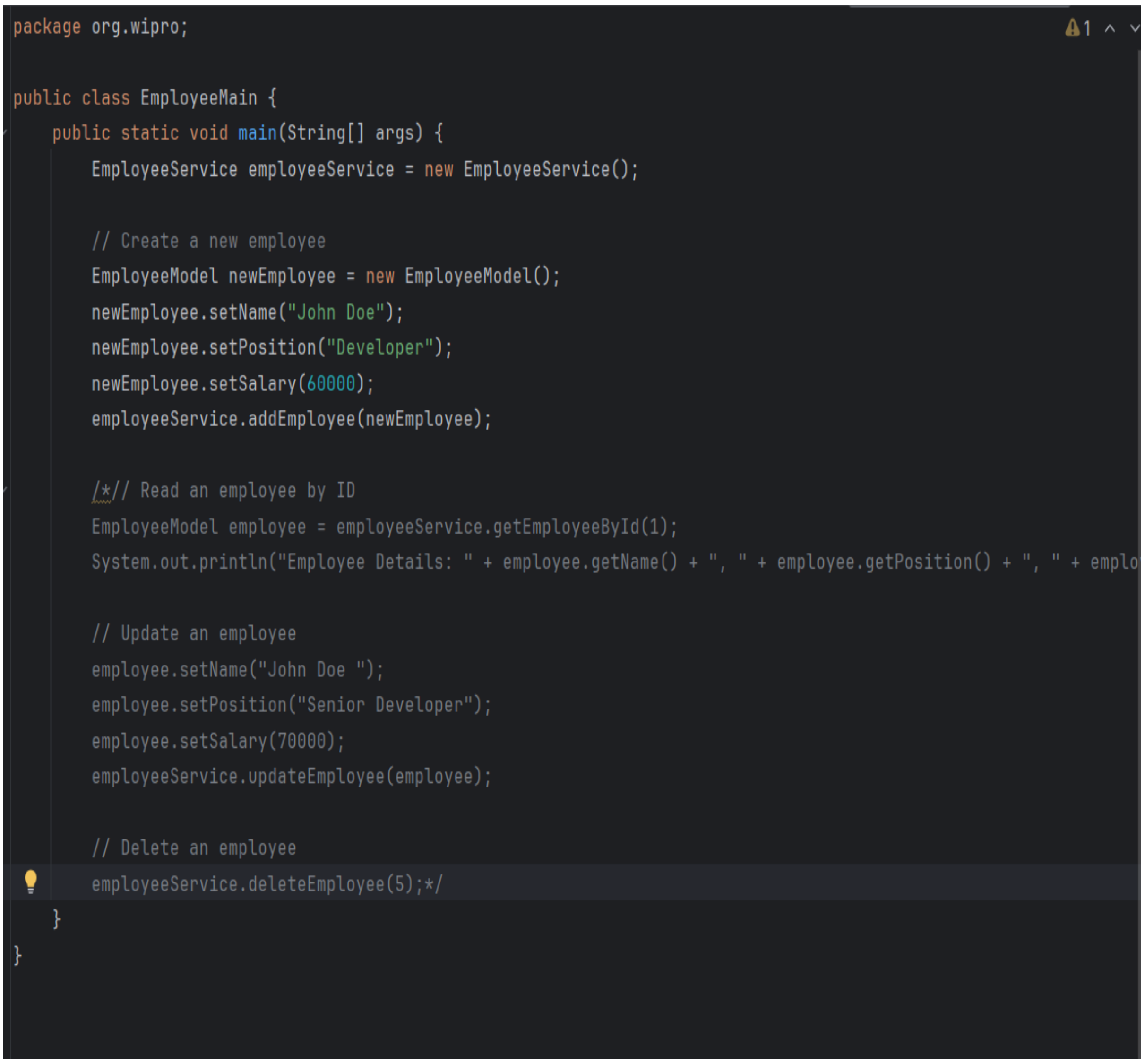
```
    @Override
    public void deleteEmployee(int id) {
        String sql = "DELETE FROM employees WHERE id = ?";
        try (Connection conn = getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(1, id);
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }*/
}
```

EmployeeService:



```java
package org.wipro;

import java.util.List;

public class EmployeeService {  2 usages
    private EmployeeDao employeeDAO = new EmployeeDaoImp();  1 usage

    public void addEmployee(EmployeeModel employee) {  1 usage
        employeeDAO.addEmployee(employee);
    }

  /*  public EmployeeModel getEmployeeById(int id) {
        return employeeDAO.getEmployeeById(id);
    }

    public List<EmployeeModel> getAllEmployees() {
        return employeeDAO.getAllEmployees();
    }

    public void updateEmployee(EmployeeModel employee) {
        employeeDAO.updateEmployee(employee);
    }

    public void deleteEmployee(int id) {
        employeeDAO.deleteEmployee(id);
    }*/
}
```

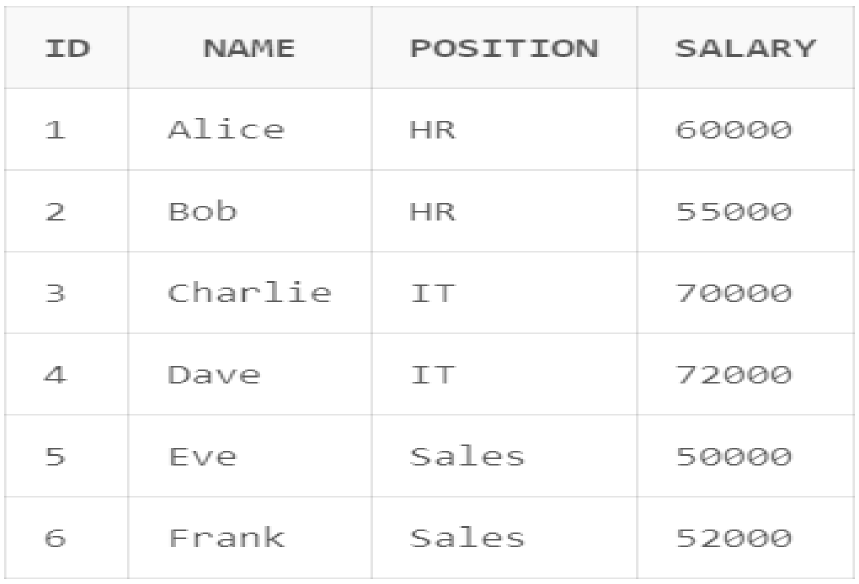
EmployeeMain:



```
package org.wipro;

public class EmployeeMain {
    public static void main(String[] args) {
        EmployeeService employeeService = new EmployeeService();

        // Create a new employee
        EmployeeModel newEmployee = new EmployeeModel();
        newEmployee.setName("John Doe");
        newEmployee.setPosition("Developer");
        newEmployee.setSalary(60000);
        employeeService.addEmployee(newEmployee);

        /*// Read an employee by ID
        EmployeeModel employee = employeeService.getEmployeeById(1);
        System.out.println("Employee Details: " + employee.getName() + ", " + employee.getPosition() + ", " + emplo

        // Update an employee
        employee.setName("John Doe ");
        employee.setPosition("Senior Developer");
        employee.setSalary(70000);
        employeeService.updateEmployee(employee);

        // Delete an employee
        employeeService.deleteEmployee(5);*/
    }
}
```

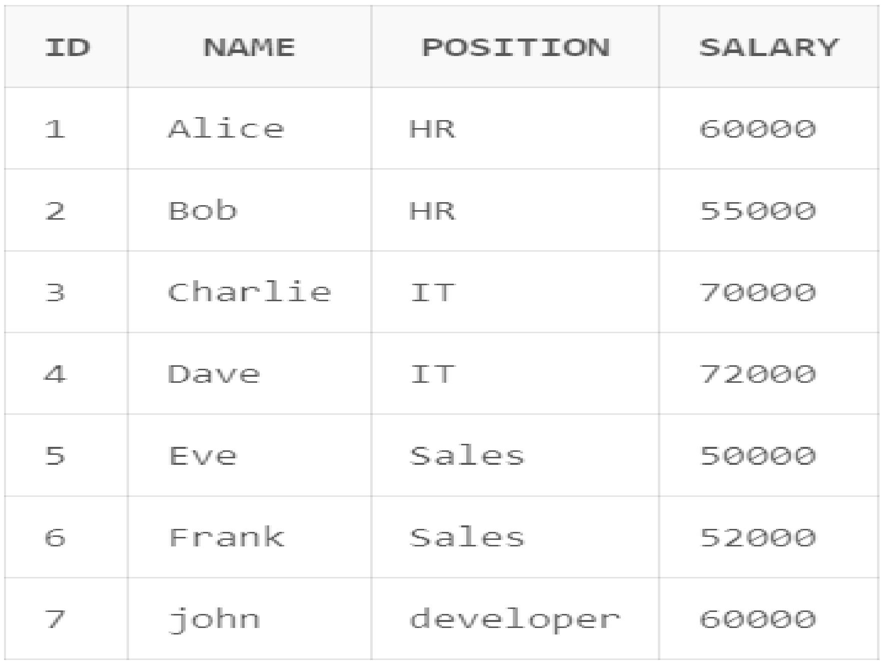
Outputs:

Insert elements:



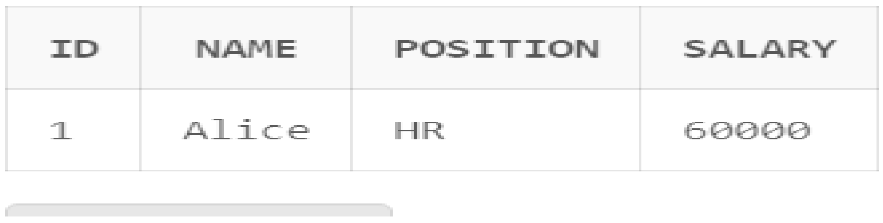
ID	NAME	POSITION	SALARY
1	Alice	HR	60000
2	Bob	HR	55000
3	Charlie	IT	70000
4	Dave	IT	72000
5	Eve	Sales	50000
6	Frank	Sales	52000

Adding new element:



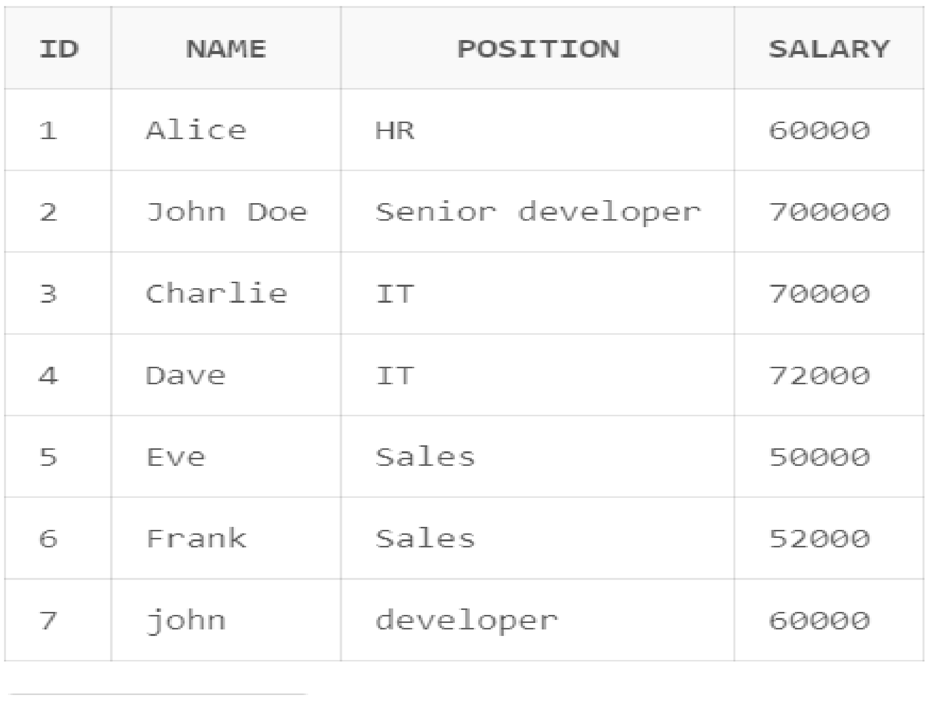
ID	NAME	POSITION	SALARY
1	Alice	HR	60000
2	Bob	HR	55000
3	Charlie	IT	70000
4	Dave	IT	72000
5	Eve	Sales	50000
6	Frank	Sales	52000
7	john	developer	60000

Getting element by ID:



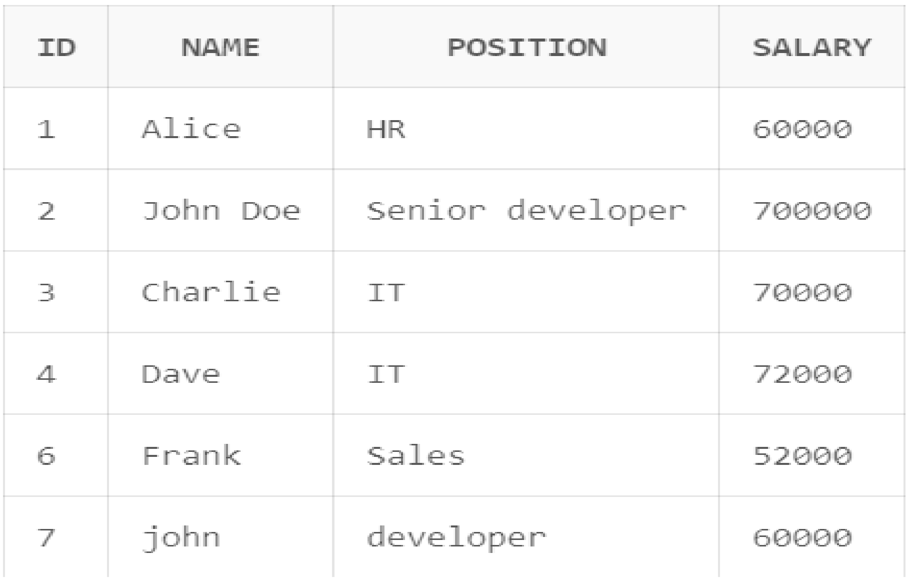
ID	NAME	POSITION	SALARY
1	Alice	HR	60000

Update:



ID	NAME	POSITION	SALARY
1	Alice	HR	60000
2	John Doe	Senior developer	700000
3	Charlie	IT	70000
4	Dave	IT	72000
5	Eve	Sales	50000
6	Frank	Sales	52000
7	john	developer	60000

Delete by ID:



ID	NAME	POSITION	SALARY
1	Alice	HR	60000
2	John Doe	Senior developer	700000
3	Charlie	IT	70000
4	Dave	IT	72000
6	Frank	Sales	52000
7	john	developer	60000