**CREATING THE DAO**

1. DAO (Data Access Object) is a class , which is used to ***access data*** from database.

2. All database related operations required by an application are defined as methods in a DAO class. For example our DAO will contain methods for **adding employee record to the database** , **searching an employee in the database** , **updating employee record** etc

3. Like POJO , it is also created on per table basis.

4. In our case , since we have only 1 table , so we would have 1 DAO class called **EmployeeDAO** in the package **empmgmt.dao**

5. Following will be the total methods in our **EmployeeDAO** class:

a. **addEmployee( )**

b. **findEmployeeById( )**

c. **getAllEmployees( )**

d. **updateEmployee( )**

e. **deleteEmployee( )**

**CREATING THE addEmployee( ) METHOD**

1. The **addEmployee( )** method of our **EmployeeDAO** class will add a new record in the **Employees** table of the database.

2. Following are it's important points:

**a. It will accept an Employee POJO object as argument containing all the fields of data**

**b. It will get a Connection object from DBConnection class using the method getConnection( )**

**c. It will then create a PreparedStatement object with insert query , fill all the values in it and add the record to the Employee table**

**d. If insertion is successful it will return true otherwise return false.**

**e. It will not handle any SQLException and will simply pass it on to it's caller**

3. Based upon the above facts the prototype of the method **addEmployee( )** is:

***public static boolean addEmployee(Employee obj) throws SQLException***

4. Following is it's code:

***public class EmpDAO {***

***public static boolean addEmployee(Employee e)throws SQLException***

***{***

***Connection conn=DBConnection.getConnection();***

***PreparedStatement ps=conn.prepareStatement("Insert into employees values(?,?,?)");***

***ps.setInt(1, e.getEmpNo());***

***ps.setString(2,e.getEmpName());***

***ps.setDouble(3, e.getEmpSal());***

***int result=ps.executeUpdate();***

***return result==1;***

***}***

**CREATING THE findEmployeeById( ) METHOD**

1. The **findEmployeeById( )** method of our **EmployeeDAO** class will search the **Employee** table for a given **EmpNo** and return the record(**EmpNo** , **Ename** and **Sal**) of that Employee

2. Following are it's important points:

**a. It will accept an Employee Number as argument.**

**b. It will get a Connection object from DBConnection class using the method getConnection( )**

**c. It will then create a PreparedStatement object with Select query with where clause, replace the placeholder with Employee Number and execute the query**

**d. If the query returns a record then it will create an Employee pojo object , fill all the column values in it and return it otherwise it will return null.**

**e. It will not handle any SQLException and will simply pass it on to it's caller**

3. Based upon the above facts the prototype of the method **findEmployeeById( )** is:

***public static Employee findEmployeeById(int empNo) throws SQLException***

4. Following is it's code:

***public static Employee findEmployeeById(int eno)throws SQLException{***

***Connection conn=DBConnection.getConnection();***

***PreparedStatement ps=conn.prepareStatement("select \* from employees where empno=?");***

***ps.setInt(1, empno);***

***ResultSet rs=ps.executeQuery();***

***Employee e=null;***

***if(rs.next())***

***{***

***e=new Employee();***

***e.setEmpNo(rs.getInt(1));***

***e.setEmpName(rs.getString(2));***

***e.setEmpSal(rs.getDouble(3));***

***}***

***return e;***

***}***

**CREATING THE getAllEmployees( ) METHOD**

1. The **getAllEmployees( )** method of our **EmployeeDAO** class will pull all the records from the **Employee** table and return them

2. Following are it's important points:

**a. It will accept no argument.**

**b. It will get a Connection object from DBConnection class using the method getConnection( )**

**c. It will then create a PreparedStatement object with Select query to select all the rows, and execute the query**

**d. It will then run a loop , fetching one row at a time , creating Employee pojo object and fill all the column values in it . It will then add this Employee object to a list.**

**e. Finally it will return that list.**

**f. It will not handle any SQLException and will simply pass it on to it's caller**

3. Based upon the above facts the prototype of the method **getAllEmployees( )** is:

***public static ArrayList<Employee> getAllEmployees() throws SQLException***

4. Following is it's code:

***public static ArrayList<Employee> getAllEmployees()throws SQLException***

***{***

***Connection conn=DBConnection.getConnection();***

***Statement st=conn.createStatement();***

***ResultSet rs=st.executeQuery("Select \* from employees");***

***ArrayList<Employee> empList=new ArrayList<>();***

***while(rs.next())***

***{***

***Employee e=new Employee();***

***e.setEmpNo(rs.getInt(1));***

***e.setEmpName(rs.getString(2));***

***e.setEmpSal(rs.getDouble(3));***

***empList.add(e);***

***}***

***return empList;***

***}***

Based upon the above understanding , students are advised to implement the remaining 2 methods **updateEmployee( )** and **deleteEmploye( )** in the **EmployeeDAO** class themselves.