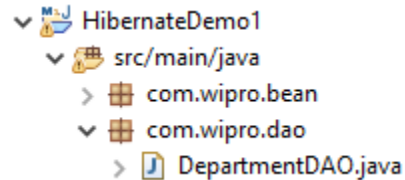# Hibernate CURD Operations

Hibernate Application for CRUD Operations

1. Let us take the same Hibernate Demo1 example
2. For the CRUD operations, let us create a separate DAO class
   a. Create a new package called com.wipro.dao
   b. Under the dao package create a class called DepartmentDAO

   - ∨ HibernateDemo1
     - ∨ src/main/java
       - > com.wipro.bean
       - ∨ com.wipro.dao
         - > DepartmentDAO.java

   c. In this class add the below code

```java
package com.wipro.dao;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

import com.wipro.bean.Department;

public class DepartmentDAO {
	/**
	 * SessionFactory object is declared at instance level
	 * so it can be called from any method under the DAO class
	 */
	SessionFactory sessionFactory;
	/**
	 * SessionFactory object is created in the constructor using the configuration file
	 */
	public DepartmentDAO() {
		Configuration cfg = new Configuration().configure();
		sessionFactory = cfg.buildSessionFactory();
	}
	/**
	 * Accepts the department Object and inserts the same to the DB
	 * @param department
	 * @return
	 */
	public String insertDepartment(Department department) {
		Session session = sessionFactory.openSession();
		Transaction transaction = session.beginTransaction();
```

```java
                session.save(department);
                transaction.commit();
                session.close();
                return "Successful";
        }

        /**
         * Accepts the department Object and updates the changes to the DEpt table
         * based on the deptno
         * @param department
         * @return
         */

        public String updateDepartment(Department department) {
                Session session = sessionFactory.openSession();
                Transaction transaction = session.beginTransaction();
                session.update(department);
                transaction.commit();
                session.close();
                return "Successful";
        }

        /**
         * Accepts the deptno to be deleted
         * Creates a Department Object initialize the deptno
         * this objct is passed to delete method of Session object for DB deletion
         * @param deptno
         * @return
         */
        public String deleteDepartment(int deptno) {
                Session session = sessionFactory.openSession();
                Transaction transaction = session.beginTransaction();
                Department deptobj = new Department();
                deptobj.setDeptno(deptno);
                session.delete(deptobj);
                transaction.commit();
                session.close();
                return "Successful";
        }

        /**
         * get method returns the Department object with values based on the given
deptno
         * If the deptno doesn't exist in the table it returns NULL
```

```
             * @param deptno
             * @return
             */
            public Department getDepartment(int deptno) {
                    Session session = sessionFactory.openSession();
                    Department deptobj = new Department();
                    deptobj=session.get(Department.class, deptno);
                    session.close();
                    return deptobj;
            }

            /**
             * load method returns the Department object with values based on the given
deptno
             * If the deptno doesn't exist in the table it throws ObjectNotFoundException
             * @param deptno
             * @return
             */

            public Department loadDepartment(int deptno) {
                    Session session = sessionFactory.openSession();
                    Department deptobj = new Department();
                    deptobj=session.load(Department.class, deptno);
                    session.close();
                    return deptobj;
            }

}
```

d. Test each of the operation from the DepartmentTester class under service package
   i. Similar to the below given sample code we can test all the methods of dao class

Sample code:

```
package com.wipro.services;

import com.wipro.bean.Department;
import com.wipro.dao.DepartmentDAO;

public class DepartmentTesterInsert {
    public static void main(String[] args) {
        DepartmentDAO dao = new DepartmentDAO();
        Department department = new Department(80, "HEALTH",
"BDC");
        String result = dao.insertDepartment(department);
        System.out.println(result);
    }
}
```