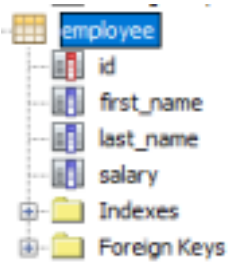
Implement Hibernate Example.
a) Create an Employee class and map it to a table in the database by annotations
b) Perform the following operations on the Employee table
a. Adding a new employee
b. Deleting an existing employee
c. Updating the salary of a given employee by 10%.

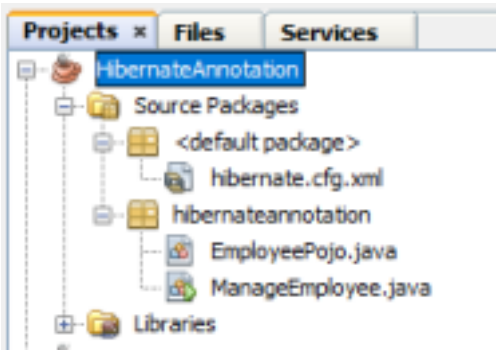
Step 1:




```
create table EMPLOYEE(
id INT NOT NULL auto_increment,
first_name VARCHAR(20) default NULL,
last_name VARCHAR(20) default NULL,
salary INT default NULL,
PRIMARY KEY(id)
);
```

Insert 1 record

Step 2:



EmployeePojo.java

```
package hibernateannotation;

import javax.persistence.*;

 @Entity

 @Table(name="employee")
```

```
public class EmployeePojo {
 @Id @GeneratedValue
 @Column(name = "id")
 private int id;

 @Column(name = "first_name")
 private String firstName;

 @Column(name = "last_name")
 private String lastName;

 @Column(name = "salary")
 private int salary;

 public int getId() {
 return id;
 }
 public void setId(int id) {
 this.id = id;
 }

 public String getFirstName() {
 return firstName;
 }
 public void setFirstName(String firstName) {
this.firstName = firstName;
 }

 public String getLastName() {
```

```
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}

public int getSalary() {
return salary;
}

public void setSalary(int salary) {
this.salary = salary;
}

}
```

Step 3:

Create hibernate.cfg.xml file (select proper table)

```
<hibernate-configuration>
 <session-factory>
 <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
<property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
 <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/retailer?zeroDateTimeBehavio
r=convertToNull</property>
 <property name="hibernate.connection.username">root</property>
 </session-factory>
</hibernate-configuration>
```

Note: this code will be automatically available in cfg file.

Step 4:

ManageEmployee.java

```
package hibernateannotation;

import java.util.Iterator;
import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.AnnotationConfiguration;

public class ManageEmployee {
 private static SessionFactory factory;
 public static void main(String[] args) {

 try {
 factory = new AnnotationConfiguration().
 configure().
 addPackage("com"). //add package if used.
 addAnnotatedClass(EmployeePojo.class).
 buildSessionFactory();
 } catch (Throwable ex) {
 System.err.println("Failed to create sessionFactory object." + ex);
throw new ExceptionInInitializerError(ex);
 }

 ManageEmployee ME = new ManageEmployee();

 /* Add few employee records in database */
 Integer empID1 = ME.addEmployee(111, "Radha", "Mehta", 1000);  Integer
empID2 = ME.addEmployee(112, "Meera", "Jani", 5000);  Integer empID3 =
```

```
ME.addEmployee(114, "Suhani", "Patel", 10000);


 /* List down all the employees */
 ME.listEmployees();
 /* Delete an employee from the database */
 ME.deleteEmployee(empID2);


 /* List down new list of the employees */
 ME.listEmployees();
 }


 /* Method to CREATE an employee in the database */
 public Integer addEmployee(int id, String fname, String lname, int salary){
Session session = factory.openSession();
 Transaction tx = null;
 Integer employeeID = null;


 try {
 tx = session.beginTransaction();
 EmployeePojo employee = new EmployeePojo();
 employee.setId(id);
 employee.setFirstName(fname);
 employee.setLastName(lname);
 employee.setSalary(salary);
 employeeID = (Integer) session.save(employee);
 tx.commit();
 } catch (HibernateException e) {
 if (tx!=null) {
 tx.rollback();
 }
```

```
e.printStackTrace();
} finally {
session.close();
}
return employeeID;
}

/* Method to READ all the employees */
public void listEmployees( ){
Session session = factory.openSession();
Transaction tx = null;

try {
tx = session.beginTransaction();
List employees = session.createQuery("FROM EmployeePojo").list();   for
(Iterator iterator = employees.iterator(); iterator.hasNext();){  EmployeePojo
employee = (EmployeePojo) iterator.next();   System.out.print("First Name:
" + employee.getFirstName());   System.out.print(" Last Name: " +
employee.getLastName());   System.out.println(" Salary: " +
employee.getSalary());   }
tx.commit();
} catch (HibernateException e) {
if (tx!=null) {
tx.rollback();
}
e.printStackTrace();
} finally {
session.close();
}
```

```
 }
 /* Method to DELETE an employee from the records */
public void deleteEmployee(Integer EmployeeID){
 Session session = factory.openSession();
 Transaction tx = null;


 try {
 tx = session.beginTransaction();
 EmployeePojo employee = (EmployeePojo)session.get(EmployeePojo.class,
EmployeeID);
 session.delete(employee);
 tx.commit();
 } catch (HibernateException e) {
 if (tx!=null) tx.rollback();
 e.printStackTrace();
 } finally {
 session.close();
 }
 }


 }
```