Name:- Shubham Pawar

Course:- Java Full Stack Development

Ref. No. SCS/CG/2021/025

Program name :- Hibernate Many to One relationship

1)EmpDetails.class

```
package com.app.model;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

@Entity
public class EmpDetails {
@Id
private int eNo;
public EmpDetails(inteNo, String eName, long salary, Department department) {
super();
this.eNo = eNo;
this.eName = eName;
this.salary = salary;
this.department = department;
}

public int geteNo() {
return eNo;
}
public void seteNo(inteNo) {
this.eNo = eNo;
}
public String geteName() {
returneName;
}
public void seteName(String eName) {
this.eName = eName;
}
public long getSalary() {
return salary;
}
public void setSalary(long salary) {
this.salary = salary;
}
public Department getDepartment() {
return department;
}
public void setDepartment(Department department) {
this.department = department;
}
private String eName;
private long salary;
@ManyToOne(targetEntity = Department.class, cascade = CascadeType.ALL, fetch =
FetchType.EAGER)
@JoinColumn(name = "deptNo", referencedColumnName = "deptNo")
private Department department;
}
```

2)Department.class

```
package com.app.model;
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class Department {
@Id
private int deptNo;
private String deptName;

public Department(intdeptNo, String deptName, String deptHead) {
super();
this.deptNo = deptNo;
this.deptName = deptName;
this.deptHead = deptHead;
}
private String deptHead;
public int getDeptNo() {
return deptNo;
}
public void setDeptNo(intdeptNo) {
this.deptNo = deptNo;
}
public String getDeptName() {
returndeptName;
}
public void setDeptName(String deptName) {
this.deptName = deptName;
}
public String getDeptHead() {
returndeptHead;
}
public void setDeptHead(String deptHead) {
this.deptHead = deptHead;
}
}
```

3)Many to One Dao.class

```
packagecom.app.dao;

publicinterfaceManyToOneDao {
voidaddEmployeeWithDept();
voiddisplayEmpAndDept();
}
```

4)Many to One Dao Impl.class

```
packagecom.app.dao.impl;

importjava.util.ArrayList;
importjava.util.List;

importorg.hibernate.Session;
importorg.hibernate.Transaction;
importorg.hibernate.query.Query;

importcom.app.dao.ManyToOneDao;
importcom.app.model.Department;
importcom.app.model.EmpDetails;
importcom.app.model.User;
importcom.app.util.UtilityClass;

public class ManyToOneDaoImpl implements ManyToOneDao{

public void addEmployeeWithDept() {

Session session=UtilityClass.getSession();
Department dept=new Department(1, "HR", "Akash");
Department dept2=new Department(2, "devloper", "shubham");
EmpDetails emp1=new EmpDetails(2001, "sangharsh", 200020, dept);
EmpDetails emp2=new EmpDetails(2002, "Ajay", 30000, dept2);
Transaction tx=session.beginTransaction();
//session.save(emp1);
session.update(emp2);
tx.commit();
UtilityClass.closeSession();
List<EmpDetails>list=new ArrayList<EmpDetails>();
list.add(emp1);
list.add(emp2);
}
public void displayEmpAndDept() {
Session session=UtilityClass.getSession();
Query<EmpDetails>query=session.createQuery("from EmpDetails");
List<EmpDetails>list=query.list();
for(EmpDetailsemp : list ) {
System.out.println(emp.geteName() + "\t" + emp.geteNo() + "\t" + emp.getSalary());
}
UtilityClass.closeSession();

}
}
```

5)Many to One Dao Factory.class

```
packagecom.app.factory;

importcom.app.dao.ManyToOneDao;
importcom.app.dao.impl.ManyToOneDaoImpl;

public class ManyToOneFactory {
public static ManyToOneDaogetManyInstance() {
return new ManyToOneDaoImpl();
}
}
```

6)Utility.class

```
packagecom.app.util;

importorg.hibernate.Session;
importorg.hibernate.SessionFactory;
importorg.hibernate.cfg.Configuration;

public class UtilityClass {
private static SessionFactory factory;
static {
try {

factory = new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

} catch (Exception e) {
// TODO: handle exception
e.printStackTrace();
}
}
staticThreadLocal<Session>threadLocal = new ThreadLocal<Session>();
static Session session = null;

public static Session getSession() {
try {
```

```
if(threadLocal.get()==null) {
session=factory.openSession();
return session;
}
else {
returnthreadLocal.get();
}
}catch (Exception e) {
// TODO: handle exception
return null;
}

}
public static void closeSession() {
try {
session.close();

} catch (Exception e) {
// TODO: handle exception
e.printStackTrace();
}
}
}
```

7)hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>

<property name = "hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hbm2ddl.auto">create</property>

<property name = "hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
```

```
<!-- Assume test is the database name -->

<property name = "hibernate.connection.url">jdbc:mysql://localhost:3306/hb1</property>
<property name = "hibernate.connection.username">root</property>

<property name = "hibernate.connection.password"></property>
<property name="show_sql">True</property>


<!--<property name="show_sql">true</property>
<property name="format_sql">true</property> -->
<!-- List of XML mapping files -->
<mapping class="com.app.model.EmpDetails"/>
<mapping class="com.app.model.Department"/>
</session-factory>
</hibernate-configuration>
```

8)Test.class

```
packagecom.app.client;

importcom.app.dao.ManyToOneDao;

importcom.app.factory.ManyToOneFactory;

public class Test {

public static void main(String[] args) {

ManyToOneDaodao = ManyToOneFactory.getManyInstance();
dao.addEmployeeWithDept();
System.out.println("success");
}
}
```

Output:-

```
Hibernate: alter table EmpDetails drop foreign key
FK2u6irygfq6blakor56e4mnol3
Hibernate: drop table if exists Department
Hibernate: drop table if exists EmpDetails
Hibernate: create table Department (deptNo integer not null,
deptHeadvarchar(255), deptNamevarchar(255), primary key (deptNo))
engine=InnoDB
Hibernate: create table EmpDetails (eNo integer not null, eNamevarchar(255),
salary bigint not null, deptNo integer, primary key (eNo)) engine=InnoDB
Hibernate: alter table EmpDetails add constraint FK2u6irygfq6blakor56e4mnol3
foreign key (deptNo) references Department (deptNo)
Hibernate: select d1_0.deptNo, d1_0.deptHead, d1_0.deptName from Department
as d1_0 where d1_0.deptNo = ?
Hibernate: select d1_0.deptNo, d1_0.deptHead, d1_0.deptName from Department
as d1_0 where d1_0.deptNo = ?
Hibernate: insert into Department (deptHead, deptName, deptNo) values
(?, ?, ?)
Hibernate: insert into EmpDetails (deptNo, eName, salary, eNo) values
(?, ?, ?, ?)
Hibernate: insert into Department (deptHead, deptName, deptNo) values
(?, ?, ?)
Hibernate: insert into EmpDetails (deptNo, eName, salary, eNo) values
(?, ?, ?, ?)
success
```