/\* Emp.hbm.xml \*/

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!-- Generated 2018-11-27 10:44:48 by Hibernate Tools 5.2.6.Final --><!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN" "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping auto-import="true" default-access="property" default-cascade="none" default-lazy="true">

<class catalog="company1" dynamic-insert="false" dynamic-update="false" mutable="true" name="com.lanou.entity.Emp" optimistic-lock="version" polymorphism="implicit" select-before-update="false" table="emp">

<id name="empNo" type="java.lang.Integer">

<column name="empNo"/>

<generator class="identity"/>

</id>

<property generated="never" lazy="false" name="empName" optimistic-lock="true" type="string" unique="false">

<column length="20" name="empName" not-null="true"/>

</property>

<!-- 多对一 在实体中

1：在多的一方的实体类(Emp) 添加一的一方(Dept)的对象

2：hbm.xml(映射文件)

在多的一方的(emp.hbm.xml) 添加many-to-one

-->

<!-- 多对一

name :一的一方的属性名

column ：外键

class: 一的一方的全类名

-->

<many-to-one name="dept" column="deptId" class="com.lanou.entity.Dept"></many-to-one>

</class>

</hibernate-mapping>

/\* Dept.hbm.xml \*/

<hibernate-mapping auto-import="true" default-access="property" default-cascade="none" default-lazy="true">

<class catalog="company1" dynamic-insert="false" dynamic-update="false" mutable="true" name="com.lanou.entity.Dept" optimistic-lock="version" polymorphism="implicit" select-before-update="false" table="dept">

<id name="deptId" type="java.lang.Integer">

<column name="deptId"/>

<generator class="identity"/>

</id>

<property generated="never" lazy="false" name="deptName" optimistic-lock="true" type="string" unique="false">

<column length="20" name="deptName" not-null="true"/>

</property>

<property generated="never" lazy="false" name="location" optimistic-lock="true" type="string" unique="false">

<column length="100" name="location" />

</property>

<!-- 一对多

name 属性

key 外键

one-to-many class : 多的一方的全类名

-->

<!-- cascade 的取值：

all: 所有情况下均进行关联操作，即save-update和delete。

none: 所有情况下均不进行关联操作。这是默认值。

save-update: 在执行save/update/saveOrUpdate时进行关联操作。

delete: 在执行delete 时进行关联操作。

merge：当通过Session的merge()方法来保存或更新当前对象时，对其关联对象也执行merge()方法。

all-delete-orphan: 当一个节点在对象图中成为孤儿节点时，删除该节点

-->

<!-- inverse：

维护关联关系，默认为false 为主动方

true 不负责维护关联关系

-->

<set name="emps" cascade="all" inverse="true" lazy="true">

<key column="deptId"></key>

<one-to-many class="com.lanou.entity.Emp"/>

</set>

</class>

</hibernate-mapping>

/\* TestEmp.java \*/

package com.lanou.test;

import org.hibernate.Criteria;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.criterion.Restrictions;

import org.junit.Test;

import com.lanou.entity.Dept;

import com.lanou.entity.Emp;

import com.lanou.util.HB;

public class TestEmp {

Session session = null;

Transaction tx = null;

/\*\*

\* 添加一个部门的多个员工

\*/

@Test

public void test01() {

session = HB.getCurrentSession();

tx = session.beginTransaction();

//Dept d = new Dept();

//d.setDeptName("IT部");

Dept d = (Dept) session.get(Dept.class, 3);

//session.save(d);

Emp e = new Emp();

e.setEmpName("诸葛亮");

e.setDept(d);

Emp e1 = new Emp();

e1.setEmpName("庞统");

e1.setDept(d);

session.save(e);

session.save(e1);

tx.commit();

}

/\*\*

\* 把诸葛亮挪到IT部门

\*/

@Test

public void test02() {

session = HB.getCurrentSession();

tx = session.beginTransaction();

// 获取IT 部门

Dept d = (Dept) session.get(Dept.class, 2);

// 获取诸葛亮员工

Criteria c = session.createCriteria(Emp.class);

c.add(Restrictions.eq("empName", "诸葛亮"));

Emp e = (Emp) c.uniqueResult();

e.setDept(d);

// session.update(e);

tx.commit();

}

}

/\* TestDept.java \*/

package com.lanou.test;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.junit.Test;

import com.lanou.entity.Dept;

import com.lanou.entity.Emp;

import com.lanou.util.HB;

public class TestDept {

Session session = null;

Transaction tx = null;

/\*\*

\* 级联添加部门

\*

\* 级联：对一方的对象进行修改 添加 删除，都会影响多的一方对象

\*/

@Test

public void saveDept() {

Dept d = new Dept();

d.setDeptName("人事部");

Emp e = new Emp();

// 多对一 insert into 能够插入deptId

e.setEmpName("貂蝉");

e.setDept(d);

// 一对多 能够执行update 修改deptId

d.getEmps().add(e);

Emp e1 = new Emp();

// 多对一

e1.setEmpName("吕布");

e1.setDept(d);

// 一对多

d.getEmps().add(e1);

session = HB.getCurrentSession();

tx = session.beginTransaction();

session.save(d);

tx.commit();

}

@Test

public void testDelete() {

session = HB.getCurrentSession();

tx = session.beginTransaction();

Dept d = (Dept) session.get(Dept.class, 4);

session.delete(d);

tx.commit();

}

@Test

public void selectDept() {

// 延迟加载lazy 默认为true 不调用不加载，调用的时候，才加载

// lazy的好处：提高程序的效率

session = HB.getCurrentSession();

tx = session.beginTransaction();

Dept d = (Dept) session.get(Dept.class, 3);

System.out.println(d.getDeptId()+"---"+d.getDeptName()+"---"+"员工人数："+d.getEmps().size());

}

}

/\* Project.hbm.xml \*/

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<!-- Generated 2018-11-29 17:21:32 by Hibernate Tools 3.5.0.Final -->

<hibernate-mapping>

<class name="com.lanou.entity.Project" table="PROJECT">

<id name="projectId" type="java.lang.Integer">

<column name="PROJECTID" />

<generator class="identity" />

</id>

<property name="projectName" type="java.lang.String">

<column name="PROJECTNAME" />

</property>

<set name="emps" table="proemp" cascade="all" inverse="false" lazy="true">

<key>

<column name="proid" />

</key>

<many-to-many class="com.lanou.entity.Employee" column="empid" />

</set>

</class>

</hibernate-mapping>

/\* Employee.hbm.xml \*/

<hibernate-mapping>

<class name="com.lanou.entity.Employee" table="EMPLOYEE">

<id name="eId" type="java.lang.Integer" access="field">

<column name="EID" />

<generator class="identity" />

</id>

<property name="eName" type="java.lang.String" access="field">

<column name="ENAME" />

</property>

<set name="pros" table="proemp" inverse="true" lazy="true">

<key>

<column name="empid" />

</key>

<many-to-many class="com.lanou.entity.Project" column="proid" />

</set>

</class>

</hibernate-mapping>

/\* TestEmpProject.java \*/

package com.lanou.test;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.junit.Test;

import com.lanou.entity.Employee;

import com.lanou.entity.Project;

import com.lanou.util.HB;

public class TestEmpProject {

Session session = null;

Transaction tx = null;

@Test

public void test01() {

Project p = new Project();

p.setProjectName("西气东输");

Employee e1 = new Employee();

e1.seteName("者行孙");

Employee e2 = new Employee();

e2.seteName("行者孙");

p.getEmps().add(e1);

p.getEmps().add(e2);

e1.getPros().add(p);

e2.getPros().add(p);

session = HB.getCurrentSession();

tx = session.beginTransaction();

session.save(p);

session.save(e1);

session.save(e2);

tx.commit();

}

}