# 手动配置

## 1.先建立student.java,黄色部分重要

**private** Integer id;

**private** String name;

**private** Department dept;

## 2.然后建立Department。Java

**private** Integer id;

**private** String name;

## 3.配置各自的xml文件 <!-- 对于private Department dept;就不能使用property column="dept\_id" 表示将来自动生成的表的外键名-->

<hibernate-mapping package=*"com.hlj.domain"*>

<class name=*"Student"*>

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

<generator class=*"increment"*/>

<!-- 指定主键生成策略 -->

</id>

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

<column name=*"name"* length=*"64"*/>

</property>

<many-to-one name=*"dept"*  column=*"dept\_id"*/>

</class>

</hibernate-mapping>

<hibernate-mapping package=*"com.hlj.domain"*>

<class name=*"Department"* lazy=*"false"* >

<!-- 配置主键属性 -->

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

<generator class=*"increment"*/>

<!-- 指定主键生成策略 -->

</id>

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

<column name=*"name"* length=*"64"* not-null=*"true"*/>

</property>

</class>

</hibernate-mapping>

# 2.开始自动生成表

<property name=*"hbm2ddl.auto"*>update</property>

<mapping resource=*"com/hlj/domain/Student.hbm.xml"* />

<mapping resource=*"com/hlj/domain/Department.hbm.xml"* />

# 3.开始执行//将我们的表通过映射文件来生成 //调用一个session 来加载hibernate //从而自动生成

**public** **class** TestMain **implements** updateEmployee {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

*CreateTable*();

}

**public** **static** **void** CreateTable(){

HibernateUtil.*openSession*();

}

}

# 4.将学生添加到系，先保存系，再保存学生，

StudentMany studentMany = **new** StudentMany();

studentMany.setName("zhao");

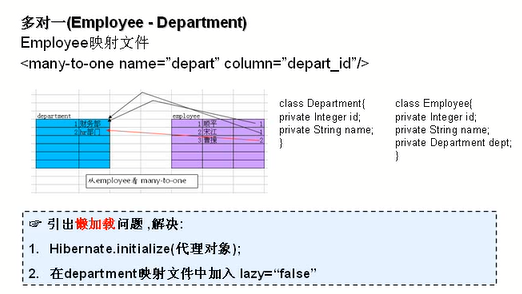
Department dept = **new** Department();

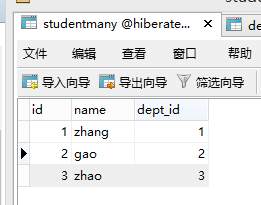
dept.setName("english");

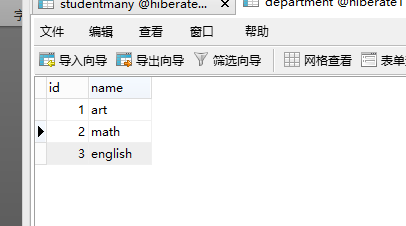
studentMany.setDept(dept);

session.save(dept);

session.save(studentMany);

5. 





# 5.懒加载//session关闭，还是可以运行，这个就是懒加载，可以保证运行，但是会报错，所以需要 我们解决

我们在查询一个对象时，返回的只是这个对象的普通属性，默认情况下是不会讲与他关联的 的表的 的属行，这个时候我们要 使用他的对象属性时会向数据库中再次查询，这个情况就是懒加载，解决方法是;

## //第一种 显示初始化代理对象 ，通过这个得到 department

//Hibernate.initialize(student1.getDept());

## //第二种 将在自己的xml文件中进行修改 将lazy 改为 false ，了

## // 第三种通过过滤器进行

student1=(StudentMany) s.load(StudentMany.**class**, 2);

//解决懒加载的方法

//第一种 显示初始化代理对象 ，通过这个得到 department

//Hibernate.initialize(student1.getDept());

System.*out*.println(student1.getName()+" 所在的系"

+student1.getDept().getName());//session还在才行