Hibernate关联映射

对于关联映射肯定需要知道user.hbm.xml作用：

其最大的作用是1.告诉hibernate加载对象

2.生成表

对象与关系之间的映像文件，配置文件根据不同映像文件来实现对象与关系转换

## 多对一：

多个用户一个组：

|  |  |
| --- | --- |
| Id | name |
| Group |  |

|  |  |  |
| --- | --- | --- |
| Id | Name | gid(group) |
| User |  | 不唯一 |

一张A表中有多个记录都与B表中某一条数据联系

单向联系：

Xml:

<hibernate-mapping package=*"com.hibernate.po"*>

<class name=*"User"* table=*"t\_users"* >

<id name=*"id"* type=*"int"*>

<generator class=*"native"*/>

</id>

<property name=*"name"* />

**<many-to-one name=*"group"* column=*"gid"* cascade=*"save-update"/*>** </class>

<class name=*"Group"* table=*"t\_group"* >

<id name=*"id"* type=*"int"*>

<generator class=*"native"*/>

</id>

<property name=*"name"* />

<set name=”users”>

<key colum=”gid”/>

<one-to-many name=”Group” />

</set>

</class>

</hibernate-mapping>

解释：其中cascade=”save-update”为级联，意思是当user或group中某一个保存了，则使用级联保存所有相关的事物

如： Group group = **new** Group();

group.setName("g1");

User user = **new** User();

user.setName("u1");

user.setGroup(group);

session.save(user);

// session.save(group);

如果没有使用级联就必须两个都保存，不然就会效率低下

1. 当没有设置外键为not null 是，会想插入两张表中的数据，后才会更新外键的数据，也就hibernate在一的一端维护，先插入多（user）数据
2. 当设置了外键为not null则报错

User user = (User)session.get(User.**class**, 1);

System.*out*.println(user.getGroup().getName());

通过查询语句出一张表，就可以得出联系表的信息，比以前的方便

说明：hibernate是通过主键加载（通过主键关联属性找到相应对象的主键）

默认为主键加载

单向：只是在用户表中加入一个 **private** Group group;或者

在用户表中加入 **private** Set<User> users = **new** HashSet<User>();

//为啥使用set而不用list，就是由于set是无序和不重付

通过用set，就可以知道效率比加group低

User user = **new** User();

user.setName("u1");

// user.setGroup(group);

User user1 = **new** User();

user.setName("u2");

// user.setGroup(group);

Group group = **new** Group();

group.setName("g1");

group.getUsers().add(user);

group.getUsers().add(user1);

session.save(user);

session.save(group);

其中必须要求update更改，双向效率比单向低

**在关系映射文件中需要添加相应的<set></set>,该标签中需指定name属性用来表示集合中存储的对象，并且该标签中需要有两个子:<one-to-many/>设定class属性指定B表映射的类；<key/>需要设定column属性指定对应表中的哪一列，范例代码如下**

**[html]** [view plaincopy](http://blog.csdn.net/sanjy523892105/article/details/7061602)

1. **<set** name="emps"**>**
2. **<key** column="depart\_id"**/>**
3. **<one-to-many** class="Employee"**/>**
4. **</set>**

一对一：

有三种方式：

1. 外键

|  |  |  |  |
| --- | --- | --- | --- |
| Id |  | Id | Hid |
| Husband | Wife |  |

1. 主键设置为外键

|  |  |  |  |
| --- | --- | --- | --- |
| Id |  |  | Id |
| Husband |  | Wife |

3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id |  | Hid | Wid |  | Id |
| Husband |  |  | Wife |

单向：一主键对外键：

另外一种将主键直接设置成外键的方法，在关系映射文件中需要将主键标签<generator>中的class属性设置为”foreign”，并添加<param>子标签，标签体为关联的对象名称。

<id name=*"id"* type=*"int"*>

<generator class=*"foreign"*>

<param name=*"property"*>group</param>

</generator>

</id>

<property name=*"name"* />

<one-to-one name=*"group"* constrained=*"true"*></one-to-one>

**constrained=*"true"表示主键为外键的联系***

2.在实体类中写： **private** Group group;

3.**一对一默认级联**

**双向：**

在<one-to-one name=*"user"*></one-to-one> 加个这个就ok

外向关联映射：（用外键关联较多）

双向：

<many-to-one name=*"group"* column=*"gid"* unique=*"true"/*>

<one-to-one name=*"user"/*>

多对多：

<set name=*"users"* table=*"t\_user\_group"* cascade=*"save-update"*>

<key column=*"uid"*></key>

<many-to-many class=*"Group"* column=*"gid"*></many-to-many>

</set>

<set name=*"groups"* table=*"t\_user\_group"* cascade=*"save-update"*>

<key column=*"gid"*></key>

<many-to-many class=*"User"* column=*"uid"*></many-to-many>

</set>

<hibernate-mapping package=*"com.hibernate.po"*>

<class name=*"User"* table=*"t\_users"* >

<id name=*"id"* type=*"int"*>

<generator class=*"native"*></generator>

</id>

<property name=*"name"* />

<set name=*"groups"*>

<key column=*"gid"*></key>

<one-to-many class=*"User"*/>

</set>

</class>

<class name=*"Group"* table=*"t\_group"* >

<id name=*"id"* type=*"int"*>

<generator class=*"native"*/>

</id>

<property name=*"name"* />

<many-to-one name=*"user"* column=*"uid"*></many-to-one>

</class>

</hibernate-mapping>

难点难点难点难点2：：：：成绩表（cjb））需要配置复合主键，

<hibernate-mapping>

<class name="org.model.Cjb" table="CJB" schema="dbo" catalog="XSCJ"> <!-- 复合主键配置 其中两个key-property,分别对应两个主键--> <composite-id name="id" class="org.model.CjbId">

<key-property name="xh" type="java.lang.String">

<column name="XH" length="6" />

</key-property>

<key-property name="kch" type="java.lang.String">

<column name="KCH" length="4" />

</key-property>

</composite-id>

<property name="cj" type="java.lang.Integer">

<column name="CJ" />

</property>

<property name="xf" type="java.lang.Integer">

<column name="XF" />

</property>

</class> </hibernate-mapping>

对应的POJO类由复合主键的POJO类和成绩表的POJO类组成，