Hibernate的关联映射之1-1关联映射

# 单向1-1映射关联

## 基于主键的单向1-1映射关联

对于基于主键的单向1-1关联，基于主键关联的持久化类不能拥有自己的主键生成器策略，它的主键由关联实体来负责生成。

是根据他自己的person属性来获得的。即他通过自身的一个getPerson()方法来获得Person对象。然后通过Person对象中的getID()方法获得id，然后赋值给自身id。这样就可以不需要自己来生成id。采用基于主键的1-1关联时，应使用<one-to-one.../>元素来映射关联实体，配置<one-to-one.../>元素时需要指定一个name属性。实例(Person<--IDCard)

|  |
| --- |
| -- 实体类  public class Person {  private Integer id;  private String name;  private IDCard idCard;  //无参构造器，set,get,tostring等...  public class IDCard {  private Integer id;  private String useLife;  private Person person;  //无参构造器，set,get,tostring等... |
| 映射文件  <?xml version="1.0"?>  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">  <hibernate-mapping package="com.immoc.entity">  <class name="Person" table="person">  <id name="id" column="personID">  <generator class="native" />  </id>    <property name="name" column="personName" />  </class>  </hibernate-mapping>  <?xml version="1.0"?>  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">  <hibernate-mapping package="com.immoc.entity">  <class name="IDCard" table="idCard">  <id name="id" column="idCardID">  <!-- 基于主键关联时,主键生成策略是foreign，表明根据关联类的主键来生成该实体的主键 -->  <generator class="foreign">  <!-- 指定引用关联实体的属性名 -->  <param name="property">person</param>  </generator>  </id>    <property name="useLife" column="useLife" />  <one-to-one name="person" constrained="true" />  </class>  </hibernate-mapping>  <mapping resource=*"Person.hbm.xml"* />  <mapping resource=*"IDCard.hbm.xml"* />  在上面的映射文件中，采用这种关联映射策略是，idCard表作为从表，此时idCard表的主键将没有自己的主键生成策略。他是根据person表中的主键来生成的。同时从表(idCard)里记录的主键将会与主表（person）里的记录保存一致。  IDCard中的id既是主键也是外键。那么idCard表如何通过person表来生成主键的呢？  foreign生成器中有一个元素：property。这个元素代表了该表的外键是从哪一个属性中获得的。通过上面的配置可以发现idCard表的外键是从person属性中获得的。它的外键就是通过person属性中的getId获得id，然后将该id直接赋给id。  使用constrained="true"表明该类对应表和被关联的对象所对应的数据库表之间通过一个外键引用对主键进行约束。通过上面的配置后，就可以对两个实体进行操作了： |
| **public** **class** OneToOneTest {  **private** SessionFactory sessionFactory;  **private** Session session;  **private** Transaction transaction;  @Before  **public** **void** init() {  //创建配置对象  Configuration config = **new** Configuration().configure();  //创建会话工厂对象  sessionFactory = config.buildSessionFactory();  //创建会话对象  session = sessionFactory.openSession();  //开启事务  transaction = session.beginTransaction();  }  @After  **public** **void** destory() {  //提交事务  transaction.commit();  //关闭会话  session.close();  //关闭会话工厂  sessionFactory.close();  }  @Test  **public** **void** testOneToOneadd1() {  IDCard idCard = **new** IDCard();  idCard.setUseLife("10年");  Person person = **new** Person();  person.setName("chentmt");  idCard.setPerson(person);  //session.save(person);  session.save(idCard);  }  } |

## 基于外键的单向1-1映射关联

基于外键的关联映射与一般的N-1关联映射没有多大区别，只需要将<many-to-one.../>元素中增加unique="true"属性即可。如下：

|  |
| --- |
| <hibernate-mapping package="com.hibernate.domain">  <class name="IDCard" table="idCard">  <id name="id" column="idCardID">  <generator class="native" />  </id>    <property name="useLife" column="useLife" />  <many-to-one name="person" column="personID" unique="true"/>  </class>  </hibernate-mapping>  unique="true"代表了idCard表的personid外键列上增加了唯一约束--这样就完成基于外键的单向1-1映射了。其他的配置文件和持久化类都不需要做修改 |

## 有连接表的单向1-1映射关联

和上面差不多，只需要在有连接表的N-1关联映射中的<many-to-one.../>元素增加一个unique="true"即可。如下：

|  |
| --- |
| <hibernate-mapping package="com.hibernate.domain">  <class name="IDCard" table="idCard">  <id name="id" column="idCardID">  <generator class="native" />  </id>    <property name="useLife" column="useLife" />  <!-- 使用join元素强制使用连接表 -->  <join table="person\_idCard">  <key column="idCardID"/>  <many-to-one name="person" column="personID" unique="true"/>  </join>  </class>  </hibernate-mapping> |

# 双向1-1关联映射

对于双向的1-1关联需要让两个持久化列都增加引用关联实体的属性，并为该属性提供setter和getter方法。持久化类：如上。

## 基于主键的双向1-1关联映射

基于主键的双向1-1关联映射同样需要在一端的主键生成器策略使用foreign策略，表明将根据对方的主键来生成自己的主键，本实体不能拥有自己的主键生成策略。另一端需要使用<one-to-one.../>元素用来映射关联实体，否则就变成了单向的。映射文件如下：

|  |
| --- |
| <hibernate-mapping package="com.hibernate.domain">  <class name="Person" table="person">  <id name="id" column="personID">  <generator class="native" />  </id>    <property name="name" column="personName" />  <!-- 映射关联实体 -->  <one-to-one name="idCard" />  </class>  </hibernate-mapping>  <hibernate-mapping package="com.hibernate.domain">  <class name="IDCard" table="idCard">  <id name="id" column="idCardID">  <!-- 基于主键关联时,主键生成策略是foreign，表明根据关联类的主键来生成该实体的主键 -->  <generator class="foreign">  <!-- 指定引用关联实体的属性名 -->  <param name="property">person</param>  </generator>  </id>    <property name="useLife" column="useLife" />  <one-to-one name="person" constrained="false" />  </class>  </hibernate-mapping> |

## 基于外键的双向1-1关联映射

对于基于外键的双向1-1关联映射。外键可以存放任意一边。需要存放外键的一端，需要增加<many-to-one../>元素。同时也需要添加unique="true"属性。

对于双向单位1-1关联映射，两个实体原本是处于平等状态的。但是当我们选择一个表来增加外键后，该表就变成了从表，另一个表变成主表。

另一端需要使用<one-to-one../>元素，该元素需要使用name属性指定关联属性名，同时也需要使用property-ref属性来指定引用关联类的属性。property-ref的值是从表中的引用属性。映射文件如下：

|  |
| --- |
| <hibernate-mapping package="com.hibernate.domain">  <class name="Person" table="person">  <id name="id" column="personID">  <generator class="native" />  </id>    <property name="name" column="personName" />  <!-- 映射关联实体 -->  <one-to-one name="idCard" property-ref="person"/>  </class>  </hibernate-mapping>  <hibernate-mapping package="com.hibernate.domain">  <class name="IDCard" table="idCard">  <id name="id" column="idCardID">  <generator class="native" />    </id>    <property name="useLife" column="useLife" />  <many-to-one name="person" column="personID" unique="true" />  </class>  </hibernate-mapping> |

## 有连接表的双向1-1关联映射

采用这个方式是非常少的，因为这中情况映射相当复杂，数据模型繁琐，一般不推荐采用这种策略。

双向1-1关联两端都需要使用<join.../>元素指定连接表，<join../>元素的table属性用于指定连接表的表名，所有两端的table属性值应该是一致的。同时两个也需要增加key元素映射连接表的外键列，还需要增加<many-to-one../.>元素映射关联属性，两个<many-to-one.../>元素都需要增加unique="true"属性。注意这里两端的key元素和<many-to-one../>中column的属性值应该是相反的。

同时为了让hibernate在连接表的两个数据列上增加唯一约束，映射文件应该为两个<key.../>元素指定unique="true"。

当使用连接表来建立1-1关联关系时，两个实体应该是绝对的平等，不存在任何的主从约束关系，hibernate映射他们的连接表时，将会选择某一个外键作为连接表的主键--因此两个持久化类的映射文件依然不是完全相同的。映射文件必须在一端的<join.../>元素中指定inverse="true"，而另一端则指定option="true"。下面是两个映射文件

|  |
| --- |
| <hibernate-mapping package="com.hibernate.domain">  <class name="Person" table="person">  <id name="id" column="personID">  <generator class="native" />  </id>    <property name="name" column="personName" />    <join table="person\_idCard" inverse="true">  <key column="personID" unique="true" />  <many-to-one name="idCard" column="idCardID" unique="true" />  </join>  </class>  </hibernate-mapping>  <hibernate-mapping package="com.hibernate.domain">  <class name="IDCard" table="idCard">  <id name="id" column="idCardID">  <generator class="native" />  </id>    <property name="useLife" column="useLife" />  <join table="person\_idCard" optional="true">  <key column="idCardID" unique="true" />  <many-to-one name="person" column="personID" unique="true"></many-to-one>  </join>  </class>  </hibernate-mapping> |