# 基于主键的，和基于外键的

## 1.新建一个person.java

**private** Integer id;

**private** String name;

**private** IdCard idCard;

## 2.Idcard.java

**private** Integer id;

**private** Date validateDte;

**private** Person person;

## 3.外键不在person 在Idcard,下面是person配置文件中的

<!-- 这里配置person和idcard属性是一对一的关系 -->

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

## 4. IdCard.hbm.xml制定主键 <!-- 因为我们这里讲的是基于主键的one-to-one, 所以我们使用外键策略 -->

## <!-- 这里值，是指定更哪个属性one-to-one -->，下面这个既是主键也是外键

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

<generator class="foreign">

<param name="property">person</param>

</generator>

## 4.上面还没有配置玩 <!-- 这里配置IdCard和person属性是一对一的关系constrained没有配置将不会生成 外键 person -->

<one-to-one name="person" constrained="true"/>-->

# 基于外键的

# 1.只要将IdCard.hbm.ml中修改即可

<class name=*"IdCard"* table=*"idCard"*>

<!-- 基于外键的one-to-one -->

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

<generator class=*"assigned"*/>

</id>

<property name=*"validateDte"* type=*"java.util.Date"*>

<column name=*"validateDte"*/>

</property>

<!-- 这里配置IdCard和person属性是一对一的关系constrained没有配置将不会生成 外键 person -->

<!--

<many-to-one name=*"person"* unique=*"true"*/>

</class>

</hibernate-mapping>

# 2.测试

Session s=**null**;

Transaction tx=**null**;

**try** {

//����ʹ�û�ģ��������.

s=HibernateUtil.*getCurrentSession*();

tx=s.beginTransaction();

Person p1=**new** Person();

p1.setId(12345);

p1.setName("fasdf");

IdCard idCard=**new** IdCard();

idCard.setId(1945);

idCard.setValidateDte(**new** Date());

idCard.setPerson(p1);

s.save(p1);

s.save(idCard);

tx.commit();

} **catch** (Exception e) {

e.printStackTrace();

**if**(tx!=**null**){

tx.rollback();

}

}**finally**{

**if**(s!=**null** && s.isOpen()){

s.close();

}

}