# <http://blog.csdn.net/softimes/article/details/7008875>

引起问题的原因：

由Hibernate根据数据库表自动生成的"类名.hbm.xml"映射文件引起的。

首先我的表(Info)由两个字段组成，即：

int id;//主建

String name;

（自己做测试，所以就简单的建了个表）

由Hibernate生成的Info.hbm.xml中是这样写的：

-----------------------------------------------------

<id name="id" type="java.lang.Integer">  
            <column name="id" />  
            <generator class="assigned"/>  
</id>

-----------------------------------------------------

<id>这个是必须有的。它是用来定义实体的标识属性（对应数据库表的主键）

而我这里由于id本身就是主键，所以column的属性便是id

下面是很关键的一点<generator>，由于一时兴趣，于是找了很多资料，关于它的解释是：用于指定主键的生成策略。它的值有多，下面是转来的：

--------------------------------------------------------------------------------

“assigned”  
主键由外部程序负责生成，在   save()   之前指定一个。  
     
“hilo”  
通过hi/lo   算法实现的主键生成机制，需要额外的数据库表或字段提供高位值来源。  
     
“seqhilo”  
与hilo   类似，通过hi/lo   算法实现的主键生成机制，需要数据库中的   Sequence，适用于支持   Sequence   的数据库，如Oracle。  
     
“increment

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

<generator class=*"increment"*/>

</id>”  
主键按数值顺序递增。此方式的实现机制为在当前应用实例中维持一个变量，以保存着当前的最大值，之后每次需要生成主键的时候将此值加1作为主键。这种方式可能产生的问题是：不能在集群下使用。  
     
“identity”  
采用数据库提供的主键生成机制。如DB2、SQL   Server、MySQL   中的主键生成机制。  
     
“sequence”  
采用数据库提供的   sequence   机制生成主键。如   Oralce   中的Sequence。  
     
“native”  
由   Hibernate   根据使用的数据库自行判断采用   identity、hilo、sequence   其中一种作为主键生成方式。  
     
“uuid.hex”  
由   Hibernate   基于128   位   UUID   算法   生成16   进制数值（编码后以长度32   的字符串表示）作为主键。  
     
“uuid.string”  
与uuid.hex   类似，只是生成的主键未进行编码（长度16），不能应用在   PostgreSQL   数据库中。  
       
“foreign”  
使用另外一个相关联的对象的标识符作为主键。

--------------------------------------------------------------------------------

看了上面的介绍，再看看代码，原来是<Generator>属性设置有问题。

然后改为"identity"、"native"问题便解决。