

本章学习目标

- generator 主键策略
- 对象关系映射之一对多映射
- cascade 和 inverse 配置详解
- 对象关系映射之多对多映射
- 对象关系映射之一多一映射

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3     "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4     "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
5 <hibernate-mapping>
6 <!-- name: 实体类的类名; table: 表名 -->
7     <class name="hibernateday1.user" table="user">
8         <!-- id主键: name: 主键名, column: 主键名对应表中的列名 -->
9         <id name="id" column="id">
10             <generator class="native"></generator>
11         </id>
12         <!-- 其它属性: property : name: 实体类的变量名, column: 该变量名中对应的列名 -->
13         <property name="name" column="name"></property>
14         <property name="password" column="password"></property>
15     </class>
16 </hibernate-mapping>
```

generator配置：代表hibernate主键生成策略，在mysql中一个表必须要有主键，没有主键的开发是不被允许的。

一共分成三类策略：

1) 由数据库维护

identity: mysql使用identity, identity利用数据库的自增长的能力，例如mysql的auto_increment

sequence: 利用数据库的生成的能力，例如Oracle的sequence

native: 本地策略，由hibernate自动根据不同的数据库选择最优的策略

2) 由hibernate维护

uuid: 生成一个32位16进制的无序的字符串；（这个要求实体类中作为主键的属性是一个字符串类型的，如果属性的类型不是字符串的话，将会类型不匹配错误——>每一次生成的数据都是不一样的）

increment: 生成递进的数值类型（每次+1）auto_increment是数据库生成的，increment是数据库生成的。（实体类中对应的列是integer类型的数据-----它的做法是拿到数据库中对应列的最大值，将这个最大值将上1得到新插入数据的该列的属性值）

3) 由开发者维护

assigned: 将generator标签中的属性值写为assigned的话，在操作数据库的时候，需要开发中手动为主键设置值，按照自己的规则生成主键。

实际开发过程中，若没有特殊的要求，建议使用数据库维护或者hibernate维护策略来维护主键。