

## 复合主键

- 基于实体类属性
- 基于主键类



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## 基于实体类属性

- 基于业务需求，会需要使用两个字段来作复合主键，例如在Person数据表中，也许会使用"name"与"phone"两个字段来定义复合主键。

```
CREATE TABLE Person(  
    name VARCHAR(100) NOT NULL,  
    phone VARCHAR(50) NOT NULL,  
    age INT,  
    PRIMARY KEY(name,phone)  
)
```

更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



建立数据库：

```
-- 创建 person 表  
  
-- 删除 Person 表  
DROP TABLE person ;  
  
-- 创建 Person 表  
CREATE TABLE person  
(  
    name VARCHAR(100) not null ,  
    phone VARCHAR(50) not null ,  
    age INT ,  
    PRIMARY KEY (name,phone)  
)  
  
-- 事务的提交  
commit ;
```

# Person.java

```
public class Person implements Serializable {  
    private Integer id;  
    private String name;  
    private String phone;  
    private Integer age;  
    // 要实现复合主键，必须复写equals和hashCode方法  
    public boolean equals(Object obj) {  
    }  
    public int hashCode() {  
    }  
}
```

续。。。



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



如果要实现复合主键，首先类应该先满足以下要求：

1、本类必须实现 `Serializable` 接口

2、复写 `equals` 和 `hashCode` 方法

- `equals`: 对象比较方法
- `hashCode`: 取得 Hash 编码

对于 `equals` 和 `hashCode` 方法的复写，可以使用第三方工具: **commons-lang-1.0.1.jar**

## 覆写equals

```
public boolean equals(Object obj) {  
    if (obj == this) {  
        return true;  
    }  
    if (!(obj instanceof Person)) {  
        return false;  
    }  
    Person p = (Person) obj;  
    return new  
        EqualsBuilder().append(this.id,p.getId()).append(this.name,  
        p.getName()).append(  
            this.phone, p.getPhone()).isEquals();  
}
```



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)

## 覆写 hashCode

```
public int hashCode() {  
    return new  
    HashCodeBuilder().append(id).append(this.name).app  
    end(this.phone).toHashCode();  
}
```



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## Person.hbm.xml

```
<hibernate-mapping>
<class name="lxh.demo10.hibernate.Person" table="Person">
    <composite-id>
        <key-property name="name" column="name"
                      type="java.lang.String" length="100"/>
        <key-property name="phone" column="phone"
                      type="java.lang.String" length="50"/>
    </composite-id>
    <property name="age" type="java.lang.Integer"
              column="age" length="10"/>
</class>
</hibernate-mapping>
```

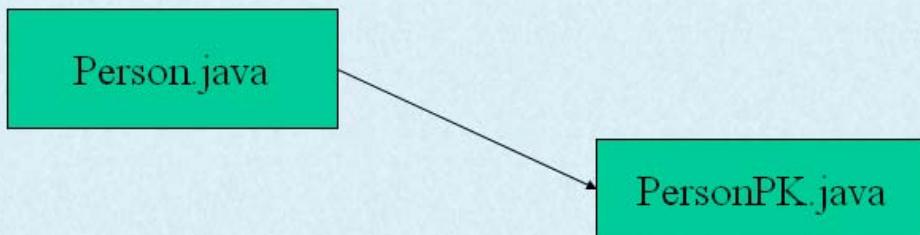


更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## 基于主键类

- 在这里，引入了PersonPK作为主键类的复合主键进行映射



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)



## 本章重点

- 掌握Hibernate的实体映射策略
- 了解Hibernate中复合主键的使用



更多学习资源尽在  
[www.MLDN.cn](http://www.MLDN.cn)

