



# HIBERNATE

---

## 第四讲 Hibernate一对一关联映射



Java与大数据分析



Java与移动智能设备开发



- 继承映射的三种方式：
  - 每个具体类对应一个表
  - 父类对应一个表
  - 每个类对应一个表



## 1 一对一关联映射

## 2 组合关系映射



## ■ 联系 ( Relationship )

- 现实世界中事物内部以及事物之间的联系在信息世界中反映为实体内部的联系和实体之间的联系。
- 实体间联系可分为：
  - 一对一联系 ( 1:1 ) ；
  - 一对多联系 ( 1:n ) ；
  - 多对多联系 ( m:n ) 。

# 两个实体型之间的联系



## ■ 一对一联系 ( 1:1 )

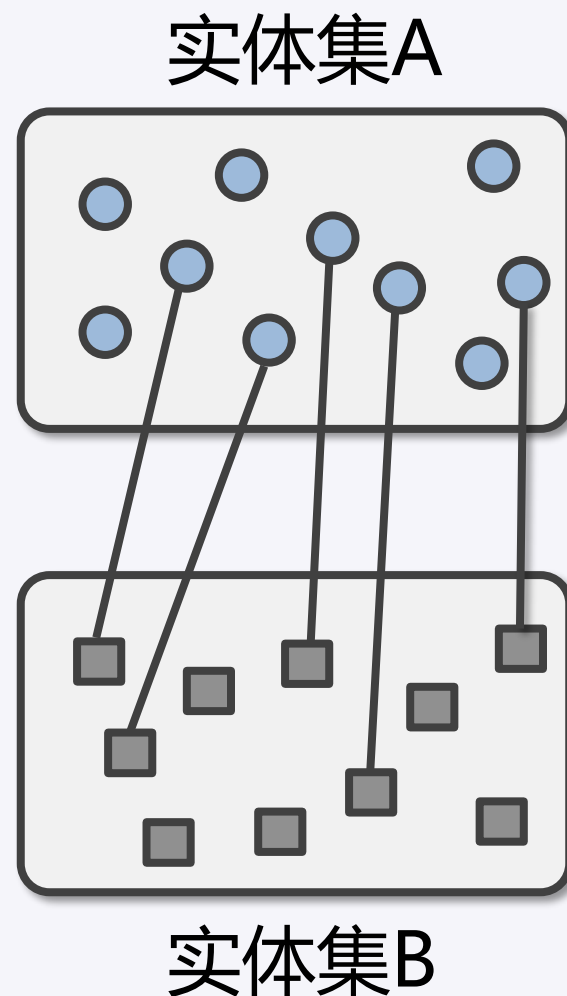
### ➤ 定义：

如果对于实体集A中的每一个实体，实体集B中至多有一个（也可以没有）实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系，记为1:1。

### ➤ 实例：

一个班级只有一个正班长。

一个班长只在一个班中任职。

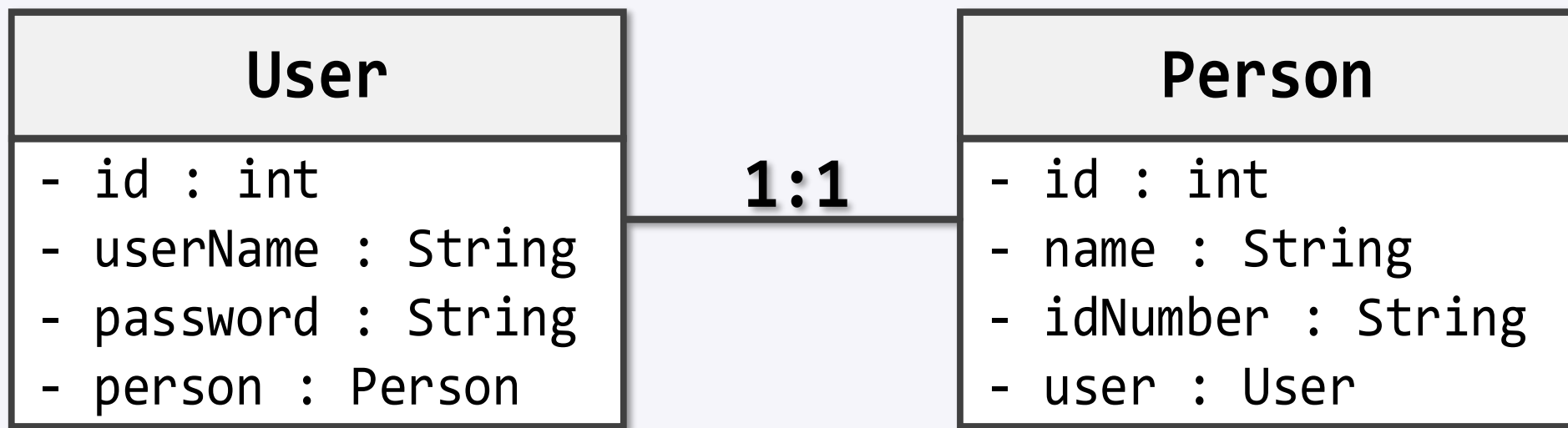




- 某网贷系统的需求中，要求每个账号都进行实名认证，也就是说一个账号只能跟一个人绑定，一个人也只能注册一个账号。



## ■ 域模型。





- 假如数据库中已经存在 USER、PERSON 表，并且 User、Person 已经设计成实体类，则 User 类与 Person 类可以映射成为一对一关联关系。





```
public class User {  
    private int id;  
    private String userName;  
    private String password;  
    private Person person;  
    .....  
}
```

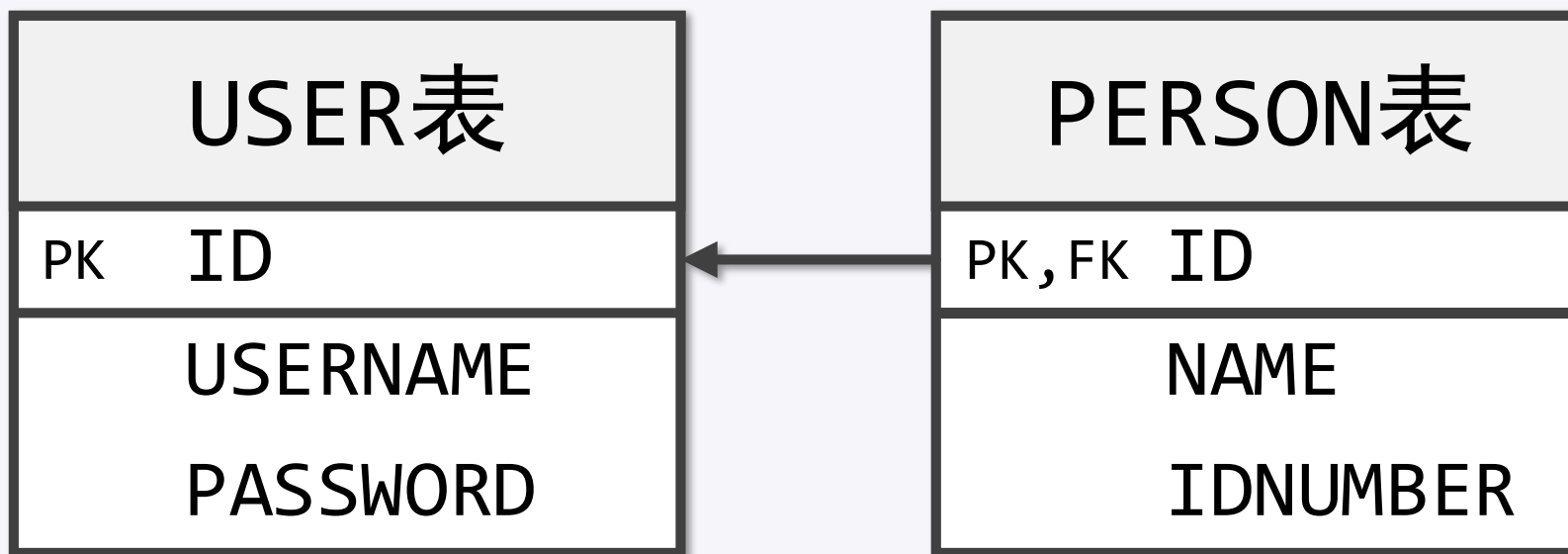
```
public class Person {  
    private Integer id;  
    private String name;  
    private String idNumber;  
    private User user;  
    .....  
}
```



- Hibernate提供两种映射一对一关联关系的方式：
  - 主键关联映射；
  - 唯一外键关联映射。
    - 外键必须设定 unique 约束。



## ■ 主键关联 - 关系数据模型。





- 主键关联映射 – 主键表（USER）。
  - 通过<one-to-one> 元素配置。
    - cascade属性（级联属性）为 all。
  - User.hbm.xml

```
<one-to-one name="person" class="Person"
            cascade="all"/>
```

# 主键关联映射



## ■ 主键关联映射 – 外键表 ( PERSON )

### ➤ 通过 <one-to-one> 元素配置。

- constrained 属性为 true , 表明 PERSON 表 ID 为外键 , 参照主表 ( USER ) 。
- 外键表实体类配置文件中 OID 使用 foreign 生成策略。

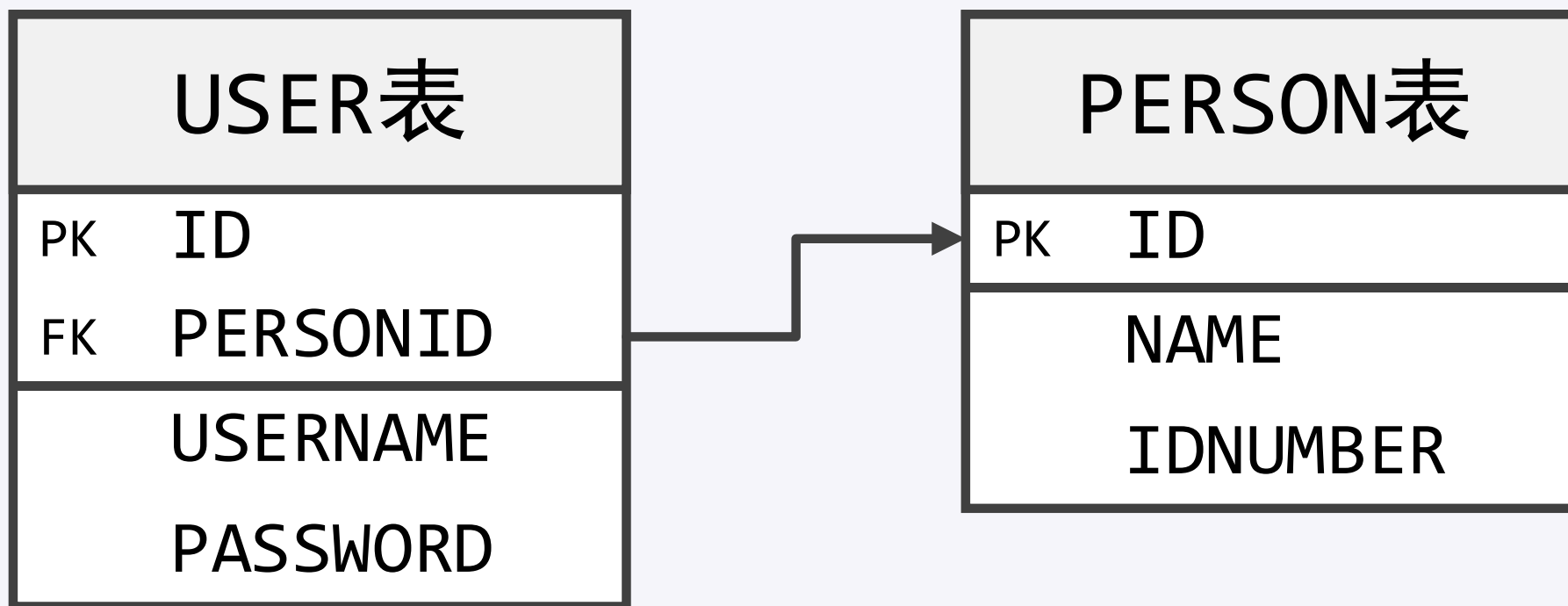
### ➤ Person.hbm.xml

```
<id name="id" >  
    <generator class="foreign" >  
        <param name="property" >user</param>  
    </generator>  
</id>  
<one-to-one name="user" constrained="true" />
```

# 唯一外键关联



## ■ 唯一外键关联 - 关系数据模型。





## ■ 唯一外键关联映射 - 外键表（USER）。

➤ 通过<many-to-one> 元素配置。

- column 属性指明外键列；
- unique 属性设置为 true，表明唯一约束；
- cascade 属性设置为 all，表明删除 USER 时同时删除 PERSON。

```
<many-to-one name="person" column="PERSONID"  
    cascade="all" unique="true"/>
```



## ■ 唯一外键关联映射 - 主键表 ( PERSON )。

➤ 通过 <one-to-one> 元素配置。

- property-ref 属性，表明建立了从User到Person的一对一关联关系。

```
<one-to-one name="user" property-ref="person"/>
```



# 使用注解映射主键关联的一对一



- 在 User 一端的 person 属性上配置。
  - @OneToOne(cascade=CascadeType.ALL) : 指定一对一关联关系，并设置级联属性。
  - @PrimaryKeyJoinColumn(name="ID") : 指定 PERSON 表主键列名。

# 使用注解映射主键关联的一对一



- 在 Person 一端的 id 属性上配置主键生成策略为 foreign。
  - `@GeneratedValue(generator="foreign")`
  - `@GenericGenerator(name="foreign", strategy="foreign", parameters={@Parameter(name="property",value="user")})`
- 在 Person 一端的 user 属性上配置一对一关联关系。
  - `@OneToOne(mappedBy="person")`

# 使用注解映射唯一外键关联的一对一



- 在User一端的person属性上配置。
  - @OneToOne(cascade=CascadeType.ALL)
  - @JoinColumn(name="PERSONID")：指明USER表中的外键列名。
- 在Person一端的user属性上配置。
  - @OneToOne(mappedBy="person")



**1** 一对一关联映射

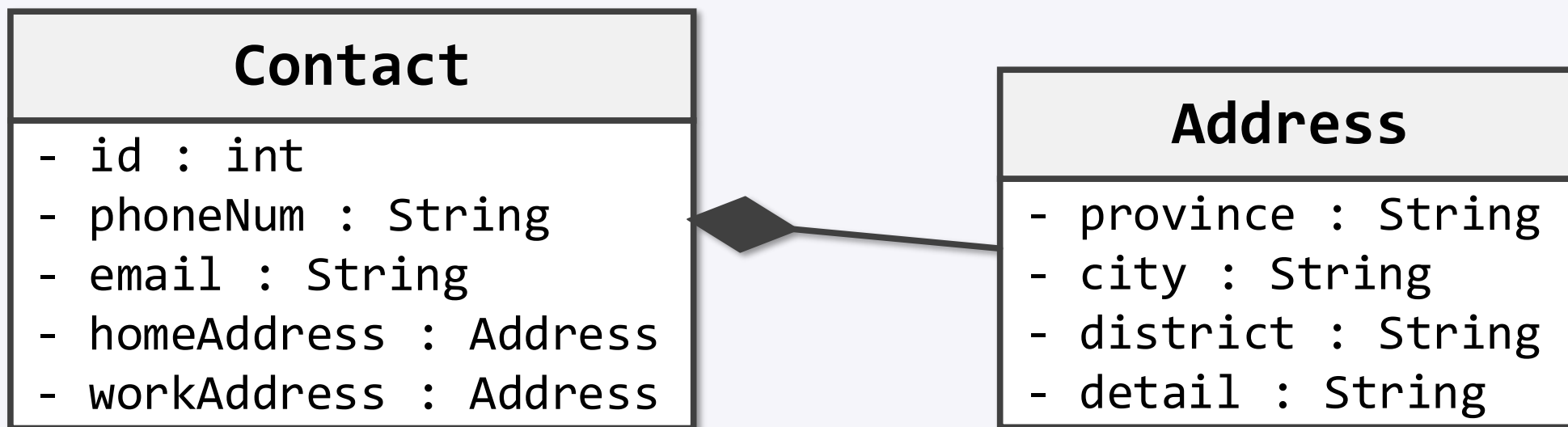
**2** 组合关系映射



- 某单位职工档案管理系统的需求中，希望能够保存职工的各种联系方式，包括：家庭地址、工作地址、籍贯地址、手机号、邮箱等，每个地址又需包含省、市、县、详细街道信息，应该如何实现？



- 联系方式和地址之间是组合关系。





- 通过对象的细化，实现更加清晰的系统逻辑，增强代码的可重用性。

```
public class Contact {  
    private int id;  
    private String phoneNum;  
    private String email;  
    private Address homeAddress;  
    private Address workAddress;  
    .....  
}
```



```
public class Address {  
  
    private String province; // 省  
    private String city;     // 市  
    private String district; // 区  
    private String detail;   // 详细地址  
    .....  
}
```





- 数据库设计的重要原则：在不导致数据冗余的情况下，要尽量减少数据库表的数目以及库表之间的外键参照关系。

CONTACT表	
PK	ID
	PHONENUM
	EMAIL
	HOMEPROVINCE
	HOMECITY
	HOMEDISTRICT
	.....



## ■ <component> 元素。

```
<component name="homeAddress" class="Address" >
    <property name="province" column="HOMEPROVINCE"/>
    <property name="city" column="HOMECITY"/>
    <property name="district" column="HOMEDISTRICT"/>
    <property name="detail" column="HOMEDETAIL"/>
</component>
```



■ 在 部分类 ( Address ) 端配置。

➤ @Embeddable : 指明为嵌入式类，不需要单独进行映射。

```
@Embeddable
public class Address {
    .....
}
```



- 在整体类（Contact）端的 homeAddress 配置。

```
@Embedded
@AttributeOverrides(value={
    @AttributeOverride(
        name = "province",
        column = @Column(name="WORKPROVINCE")),
    @AttributeOverride(
        name = "city",
        column = @Column(name="WORKCITY")),
    . . . . .
})
```



- 一对一关联映射
- 组合关系映射

# 练习





# THANK YOU

---