



河北师范大学软件学院  
Software College of Hebei Normal University

# HIBERNATE

---

## 第五讲 Hibernate一对多关联映射



BIG  
DATA

Java与大数据分析



Java与移动智能设备开发



- 一对一关联映射
  - 主键关联、唯一外键关联
- 组合关系映射



**1 实体一对多关联**

**2 数据库一对多关联**

**3 Hibernate 单向一对多关联**

**4 Hibernate 双向一对多关联**

# 两个实体型之间的联系



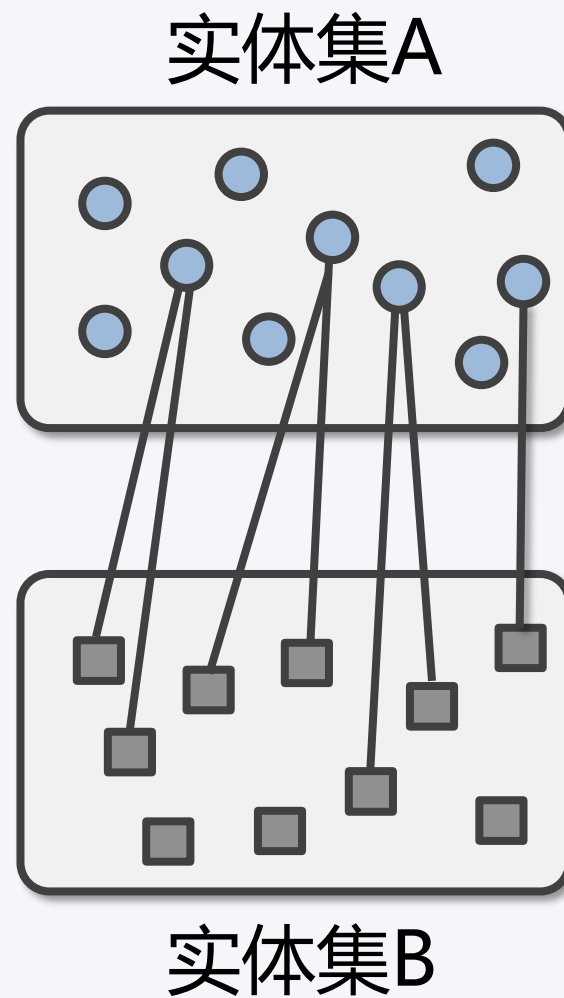
## ■ 一对多联系 (1 : n)

### ➤ 定义：

如果对于实体集A中的每一个实体，实体集B中有 $n$ 个实体 ( $n \geq 0$ ) 与之联系，反之，对于实体集B中的每一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B有一对多联系，记为1:n。

### ➤ 实例：

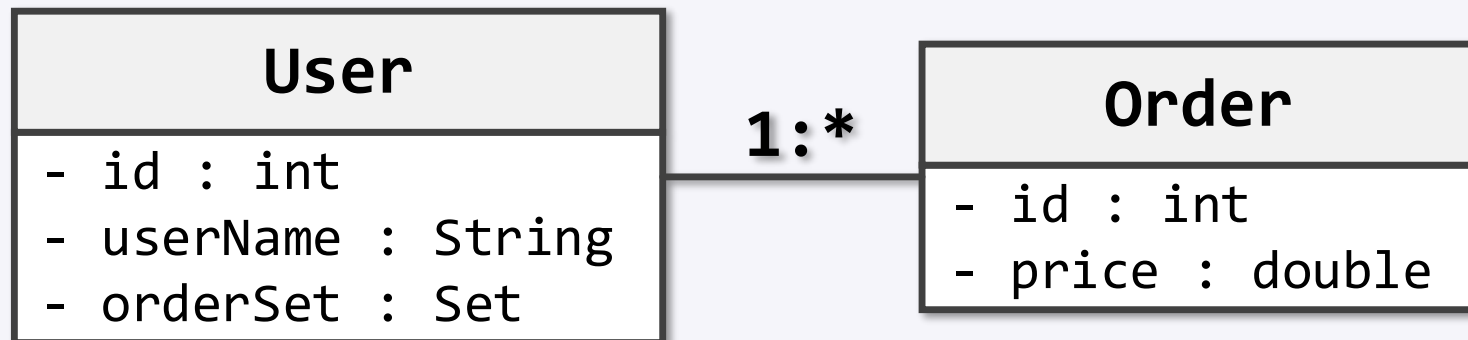
一个班级中有若干名学生，每个学生只在一个班级中学习。



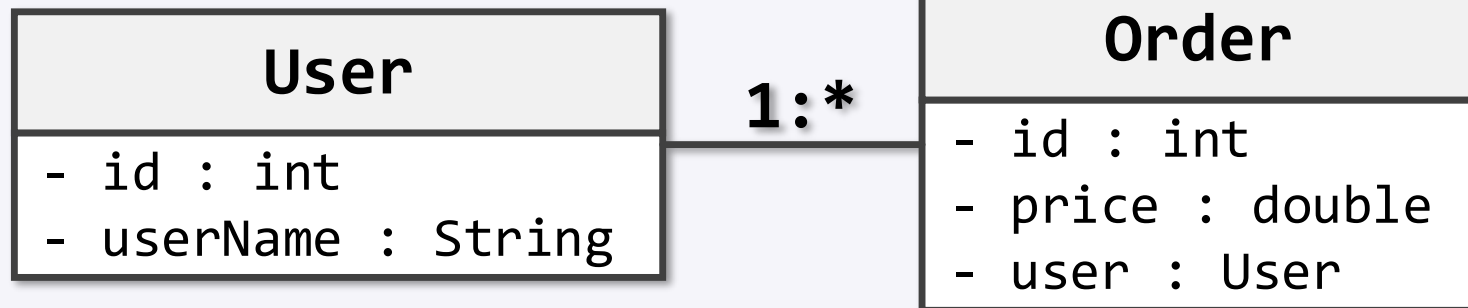


- 通常情况下，在一些购物网站系统中，用户和订单之间的关系就是一对多关联关系，并且对于用户来说需要知道自己有哪些订单，对于商家来说需要知道某个订单属于哪个用户，思考这种情况该如何实现？

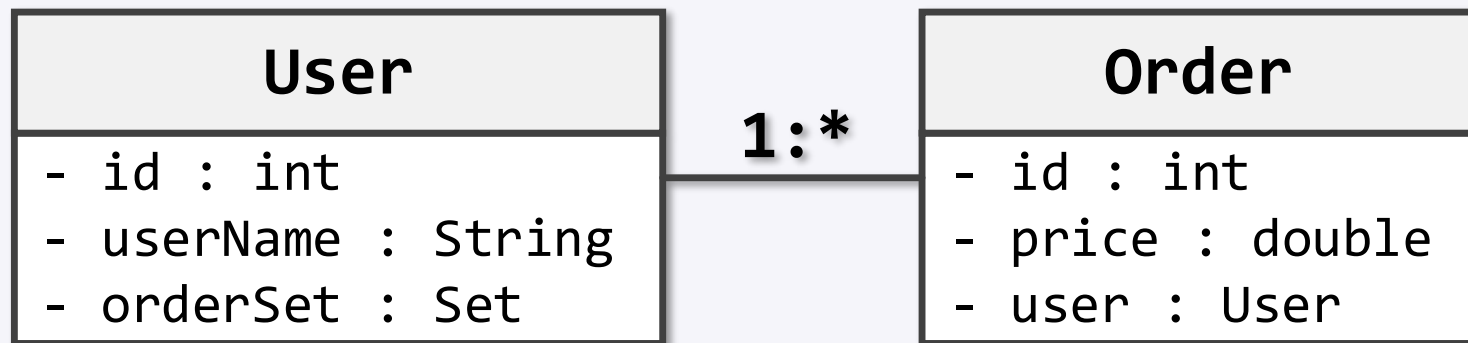
# 实体一对多关联



User 到 Order  
单向一对多关联



Order 到 User  
单向多对一关联



User 和 Order  
双向一对多关联



**1** 实体一对多关联

**2** 数据库一对多关联

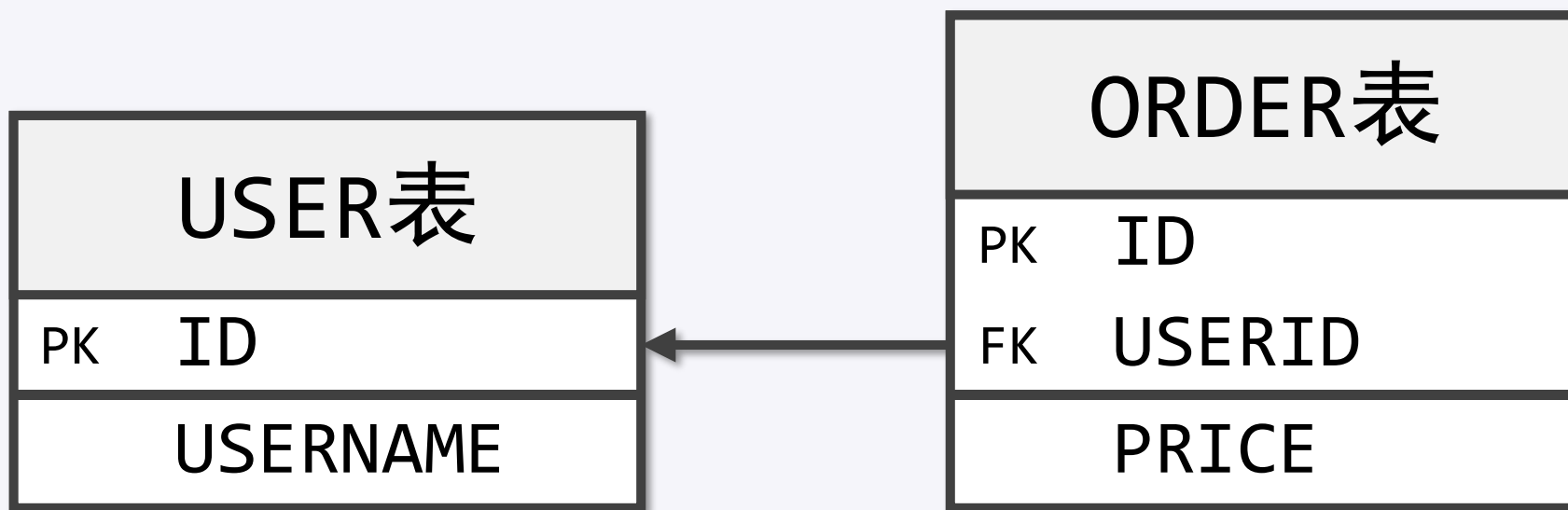
**3** Hibernate 单向一对多关联

**4** Hibernate 双向一对多关联

# 数据库一对多关联



- 在关系模型中，只存在外键参照关系，而且是 many 方参照 one 方。







**1** 实体一对多关联

**2** 数据库一对多关联

**3** **Hibernate 单向一对多关联**

**4** Hibernate 双向一对多关联

# Hibernate单向一对多关联



- 在 User 中定义一个 Order 的引用集合，而在 Order 中无需作任何定义。

```
public class User {  
    private Integer id;  
    private String userName;  
    private String password;  
    private Set orderSet = new HashSet<Order>();  
    .....  
}
```

```
public class Order {  
    private Integer id;  
    private Double price; // 价格  
    .....  
}
```



- 在 User.hbm.xml 文件中使用 <set> 元素配置。

```
<set name="orderSet" cascade="delete">  
    <key column="USERID"/>  
    <one-to-many class="Order" />  
</set>
```

- <set> 元素属性：

- name 属性：指定要映射的属性名；
- cascade 级联操作属性：save-update、delete、all、none。



## ■ <set>元素的子元素：

- <key>元素：column 属性设定所关联类对应表的外键；
- <one-to-many>元素：class 属性设定所关联的类。



- 在表示一对多关联关系时，还可以使用 List 和 Map 来表示引用集合。

```
public class User {  
    private Integer id;  
    private String userName;  
    private String password;  
    private List orderList = new ArrayList<Order>();  
    .....  
}
```



- 使用 List 映射时，需要在 Order 表中添加一个额外字段 (ORDERINDEX) 来表示插入顺序。

ORDER表	
PK	ID
FK	USERID
PRICE	
ORDERINDEX	



- 在User.hbm.xml文件中使用 <list>元素配置。

```
<list name="orderList" cascade="delete">  
    <key column="USERID"/>  
    <index column="ORDERINDEX"/>  
    <one-to-many class="Order" />  
</list>
```

- <index>子元素：

- column 属性：指定 ORDER 表中记录插入顺序的列名。



- 使用 Map 映射时，需要在 Order 表中添加一个额外字段 (ORDERKEY) 来记录 Map 的 key 值。

ORDER表	
PK	ID
FK	USERID
PRICE	
ORDERKEY	





- 在User.hbm.xml文件中使用 <map>元素配置。

```
<map name="orderMap" cascade="delete">  
    <key column="USERID"/>  
    <index column="ORDERKEY" type="string"/>  
    <one-to-many class="Order" />  
</map>
```

- <index>子元素：

- column 属性：指定记录 Map 中 key 值得字段名。
- type 属性：指定 Map 中 key 值的类型（**不可缺省**）。



- 1 实体一对多关联
- 2 数据库一对多关联
- 3 Hibernate 单向一对多关联
- 4 **Hibernate 双向一对多关联**



- 当类与类建立了关联，很方便的从一个对象导航到一个或一组与之关联的对象。
- 如有了User对象，就可以通过User对象得到这个用户所有的订单信息，`Set<Order> orders = user.getOrderSet()`。
- 对于Order对象，如果想要得到这个订单属于哪个用户，怎么办呢？

# Hibernate双向一对多关联



- 建立 User 和 Order 的双向一对多关联。
  - 在 User 到 Order 单向一对多关联关系基础上，在 Order 类中增加 User 类型的属性。

```
public class Order {  
    private Integer id;  
    private Double price; // 价格  
    private User user;  
    .....  
}
```



- Order.hbm.xml 中添加 user 属性的映射。

```
<many-to-one name="user" column="USERID" class="User"/>
```

- <many-to-one>元素的属性：
  - name 属性：指定需映射的属性名；
  - column 属性：指定ORDER中的外键列名；
  - class 属性：指定所关联的类型。

# 使用注解映射一对多关联



- 在 many 方 Order 类的 getUser() 方法上配置。

```
@ManyToOne
@JoinColumn(name="USERID")
public User getUser() {
    return user;
}
```

- @JoinColumn(name = "USERID") : 指定 ORDER 表中的外键列名。

# 使用注解映射一对多关联



- 在 one 方 User 类的 getOrderSet() 方法上配置。

```
@OneToMany(mappedBy="user", targetEntity=Order.class,  
            cascade=CascadeType.ALL)  
public Set getOrderSet() {  
    return orderSet;  
}
```

- mappedBy：表明是双向关联关系，并且与 user 建立对应。
- targetEntity：指定了所关联的类型。
- cascade：指定级联操作。



- 实体的一对多关联
  - 单向、双向
- 数据库的一对多关联
  - 外键参照关系，many方参照one方
- Hibernate的单向一对多关联
  - Set、List、Map
- Hibernate的双向一对多关联



# 练习





# THANK YOU

---