# 第三章：复杂的关联关系映射

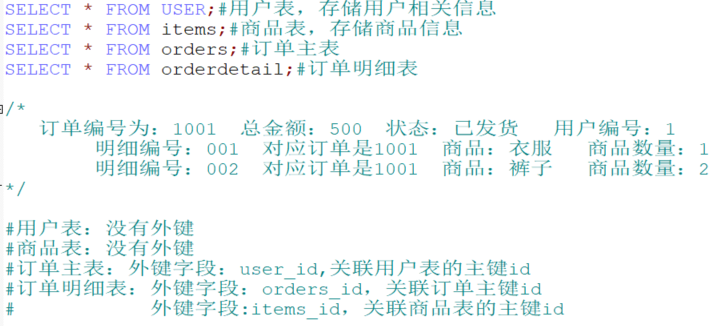
## 环境准备

### 数据库

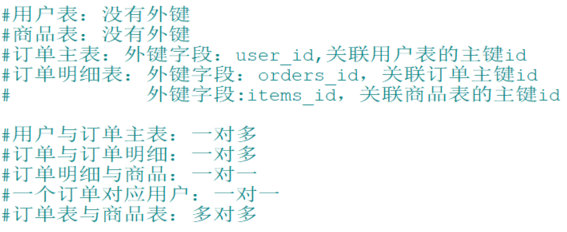


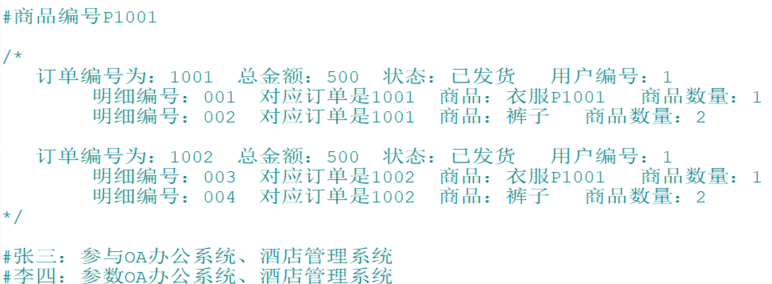
### 数据模型分析

#### 表分析



#### 关系分析





## 一对一

### 需求

**查询订单信息，关联查询用户信息**

### resultType

#### SQL语句

SELECT o.\*,u.username FROM orders o

INNER JOIN `user` u ON o.`user\_id` = u.`id`

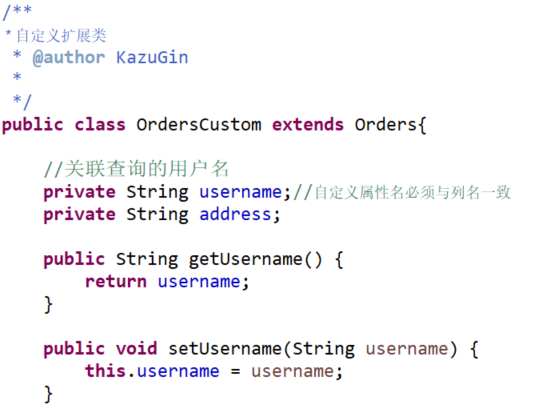
#### 创建实体类



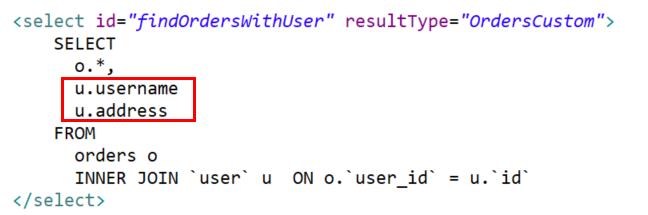
涉及连表查询，需要在实体类中添加相应的属性；

原因：在Orders类中没有username属性，所以需要在orders类中添加username

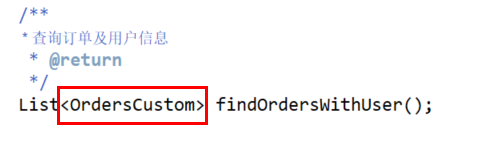
建议：自定义实体类（自定义扩展\*Custom）



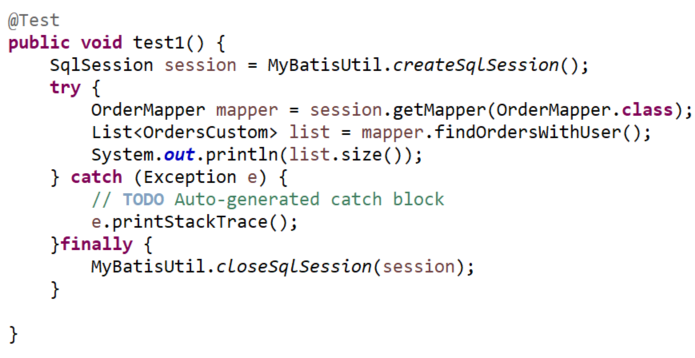
#### 映射文件



#### mapper接口



#### 测试类



### resultMap

#### SQL语句

SELECT

o.\*,

u.id user\_id,

u.username,

u.address

FROM

orders o

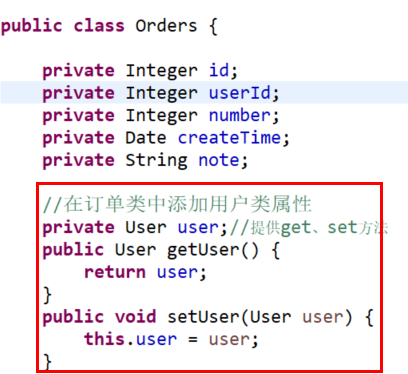
INNER JOIN `user` u ON o.`user\_id` = u.`id`

#### resultMap映射思路

将关联的用户信息配置到resultmap中即可

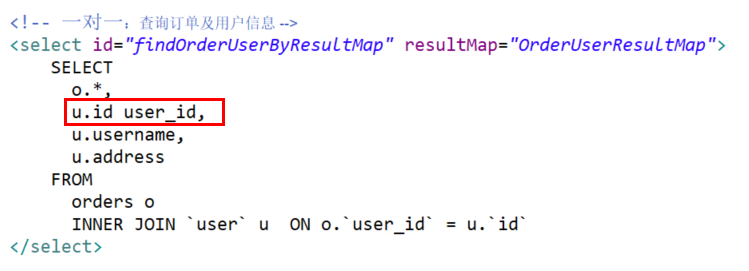
使用resultMap将查询结果中的订单信息映射到Orders对象中，在orders类中添加User属性，将关联查询出来的用户信息映射到orders对象中的user属性

#### 修改实体类

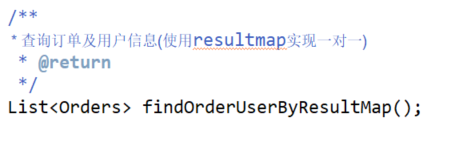


#### 映射文件

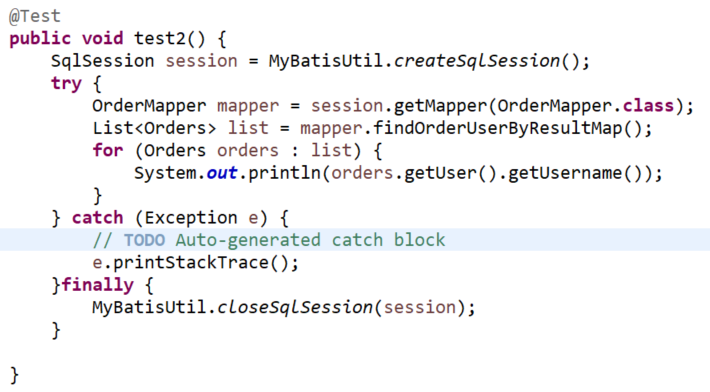




#### mapper接口



#### 测试类



### 一对一总结

实现一对一查询：

　　a.resultType：使用resultType实现较为简单，如果pojo中没有包括查询出来的列名，需要增加列名对应的属性，即可完成映射。

　　b.如果没有查询结果的特殊要求建议使用resultType。

　　c.resultMap：需要单独定义resultMap，实现有点麻烦，如果对查询结果有特殊的要求，使用resultMap可以完成将关联查询映射pojo的属性中。

d.resultMap可以实现延迟加载，resultType无法实现延迟加载

## 一对多

### 需求

**查询订单(关联用户)及订单明细**

### SQL语句

SELECT

o.\*,

u.username ,

u.`address`,

u.`id` user\_id,

d.`id` orderdetail\_id,

d.`items\_id`,

d.`items\_num`,

d.`orders\_id`

FROM

orders o

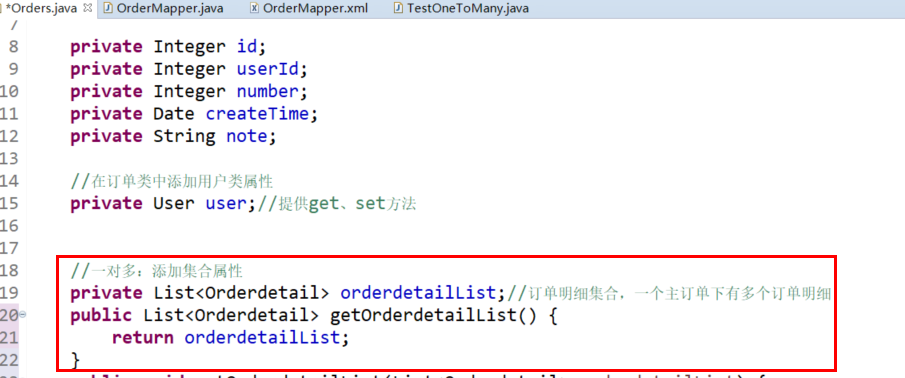
INNER JOIN `user` u ON o.`user\_id` = u.`id`

INNER JOIN orderdetail d ON d.`orders\_id` = o.`id`

### 修改实体类

一对多，需要在实体类中添加集合属性

在Orders类中添加订单明细集合

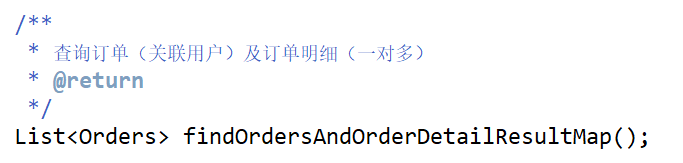


### 映射文件

定义resultMap



### mapper接口



### 测试类



## 多对多

### 需求

**查询用户以及用户购买的商品信息**

### SQL语句

SELECT

u.\*,

o.`id` order\_id,

o.`user\_id` order\_user\_id,

o.`number` order\_number,

d.`id` orderdetail\_id,

d.`items\_id`,

d.`items\_num`,

d.`orders\_id`,

i.`id` items\_item\_id,

i.`name` items\_name,

i.`price` items\_price

FROM

orders o

INNER JOIN `user` u ON o.`user\_id` = u.`id`

INNER JOIN orderdetail d ON d.`orders\_id` = o.`id`

INNER JOIN items i ON i.`id` = d.`items\_id`

### 映射思路

找准表与表之间的关系，关系主要是一对一、一对多、多对多

分析数据库表之间的关系：

用户与订单：一对多关系

订单与订单明细：一对多关系

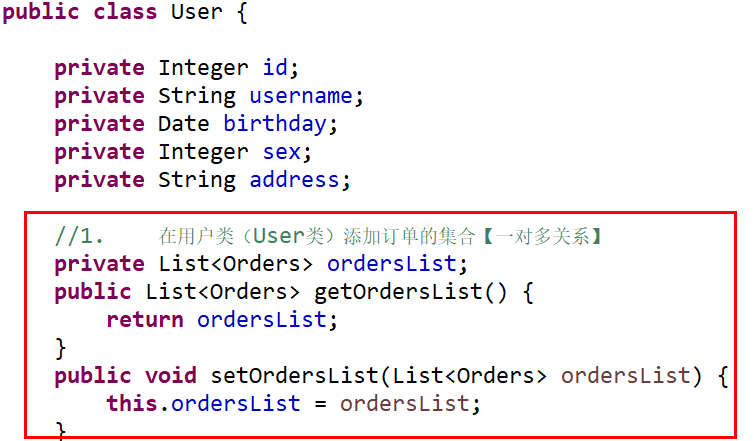
订单明细与商品：一对一关系

思路：

1. 在用户类（User类）添加订单的集合【一对多关系】
2. 在订单类中添加订单明细的集合【一对多关系】
3. 在订单明细类中添加商品对象（添加一个商品类的变量）

### 修改实体类

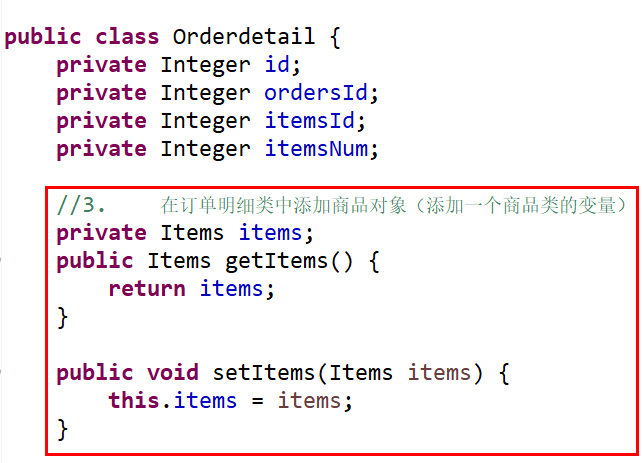
#### 用户User类



#### 订单Orders类

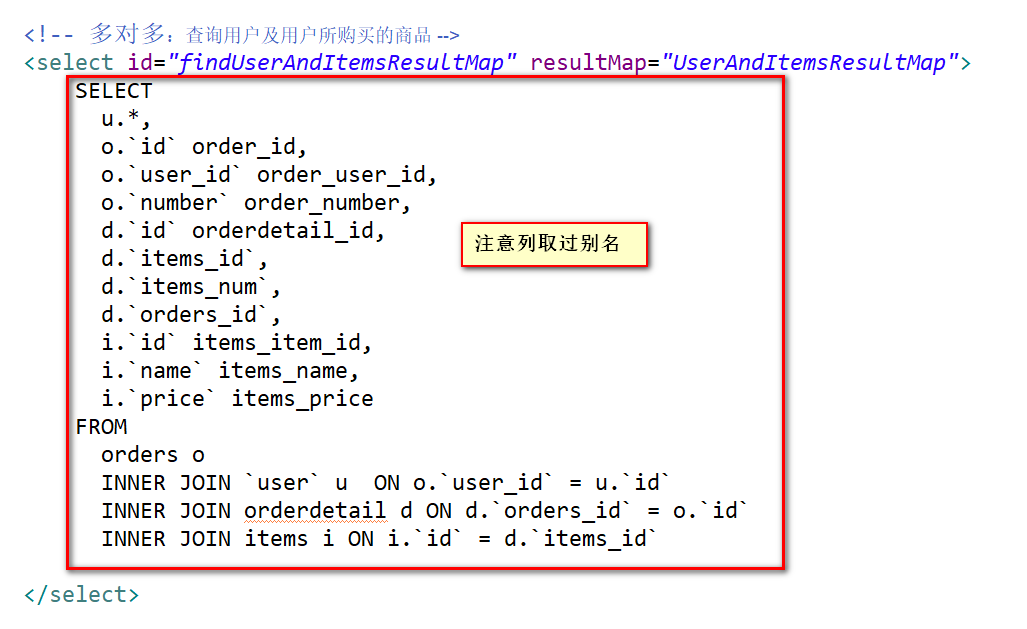


#### 订单明细Orderdetail类



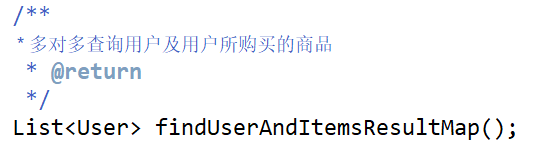
#### 映射文件



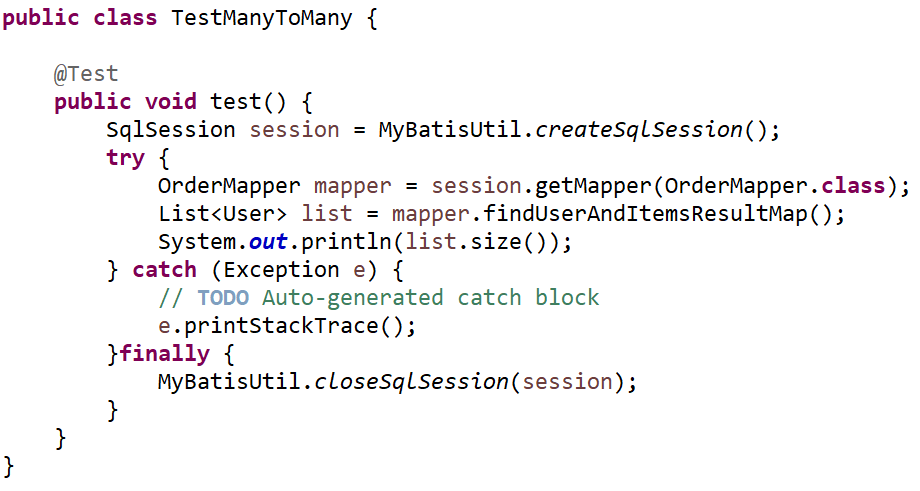


注意：resultMap配置的数据库column属性，要与SQL语句的别名一致

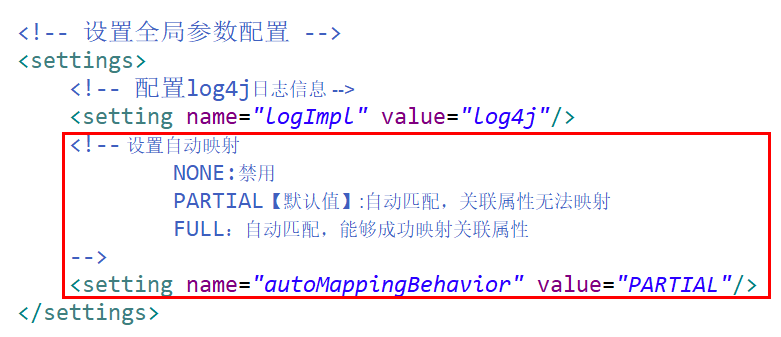
#### mapper接口



#### 测试类



## resultMap自动映射



## resultMap总结

resultType：

作用：

将查询结果按照sql列名pojo属性名一致性映射到pojo中。

场合：

常见一些明细记录的展示，比如用户购买商品明细，将关联查询信息全部展示在页面时，此时可直接使用resultType将每一条记录映射到pojo中，在前端页面遍历list（list中是pojo）即可。

resultMap：

使用association和collection完成一对一和一对多高级映射（对结果有特殊的映射要求）。

association：

作用：

将关联查询信息映射到一个pojo对象中。

场合：

为了方便查询关联信息可以使用association将关联订单信息映射为用户对象的pojo属性中，比如：查询订单及关联用户信息。

使用resultType无法将查询结果映射到pojo对象的pojo属性中，根据对结果集查询遍历的需要选择使用resultType还是resultMap。

collection：

作用：

将关联查询信息映射到一个list集合中。

场合：

为了方便查询遍历关联信息可以使用collection将关联信息映射到list集合中，比如：查询用户权限范围模块及模块下的菜单，可使用collection将模块映射到模块list中，将菜单列表映射到模块对象的菜单list属性中，这样的作的目的也是方便对查询结果集进行遍历查询。

如果使用resultType无法将查询结果映射到list集合中。