标签：mybatis

内容

[与resultType实现](https://github.com/brianway/springmvc-mybatis-learning/blob/master/mybatis/mybatis%E5%AD%A6%E4%B9%A0%E7%AC%94%E8%AE%B0(10)-%E4%B8%80%E5%AF%B9%E4%B8%80%E6%9F%A5%E8%AF%A2.md#resulttype%E5%AE%9E%E7%8E%B0)

[结果映射实现](https://github.com/brianway/springmvc-mybatis-learning/blob/master/mybatis/mybatis%E5%AD%A6%E4%B9%A0%E7%AC%94%E8%AE%B0(10)-%E4%B8%80%E5%AF%B9%E4%B8%80%E6%9F%A5%E8%AF%A2.md#resultmap%E5%AE%9E%E7%8E%B0)

[与resultType和结果映射实现一对一查询小结](https://github.com/brianway/springmvc-mybatis-learning/blob/master/mybatis/mybatis%E5%AD%A6%E4%B9%A0%E7%AC%94%E8%AE%B0(10)-%E4%B8%80%E5%AF%B9%E4%B8%80%E6%9F%A5%E8%AF%A2.md#resulttype%E5%92%8Cresultmap%E5%AE%9E%E7%8E%B0%E4%B8%80%E5%AF%B9%E4%B8%80%E6%9F%A5%E8%AF%A2%E5%B0%8F%E7%BB%93)

本文使用两种方式（与resultType和结果映射）实现一对一查询，查询订单信息，关联查询创建订单的用户信息

与resultType实现

SQL语句

确定查询的主表：订单表

确定查询的关联表：用户表

关联查询使用内连接？还是外连接？

由于订单表中有一个外键（USER\_ID），通过外键关联查询用户表只能查询出一条记录，可以使用内连接。

SELECT
订单。\*，
USER。用户名，
USER。性，
用户。地址
FROM
命令，
用户
在哪里 订单。user\_id = user。ID

创建POJO

将上边的SQL查询的结果映射到POJO中，POJO中必须包括所有查询列名。

原始的Orders.java不能映射全部字段，需要新创建的POJO。

创建一个POJO继承包括查询字段较多的口服类。

对应数据表的几个POJO类（相关文件，的OrderDetail，订单）就是把该类的属性名设为和数据表列字段名相同，并为这些属性添加吸气和设定器，在这里就不贴代码了，只贴出对应于关联查询的自定义POJO类OrdersCustom的代码

包 com.iot.mybatis.po ;
/ \*\*
\*
\* < p >标题：OrdersCustom </ p>
\* < p >描述：订单的扩展类</ p>
\* / //通过此类映射订单和用户查询的结果，让此类继承包括字段较多的pojo类公共类OrdersCustom 扩展订单 {
//添加用户属性
/ \* USER.username，
USER.sex，
USER.address \* /
私有 String用户名;
私人 弦性;
私有 字符串地址;
public String getUsername（）{
return username;
}
public void setUsername（String username）{
this 。username = username;
}
public String getSex（）{
return sex;
}
public void setSex（String sex）{
this 。性=性;
}
public String getAddress（）{
return address;
}
public void setAddress（String address）{
this 。地址=地址;
}
}

mapper.xml

<！ -查询订单关联查询用户信息- >
< select id = “ findOrdersUser ” resultType = “ com.iot.mybatis.po.OrdersCustom ” >
选择
命令。\*，
user.username，
user.sex，
user.address
从
命令，
用户
WHERE orders.user\_id = user.id
</ select >

mapper.java

//查询订单关联查询用户信息
public List < OrdersCustom > findOrdersUser（）抛出异常 ;
}

结果映射实现

使用结果映射将查询结果中的订单信息映射到订单对象中，在订单类中添加用户属性，将关联查询出来的用户信息映射到订单对象中的用户属性中。

定义的resultMap

<！ -订单查询关联用户的resultMap
将整个查询的结果映射到com.iot.mybatis.po.Orders中
- >
< resultMap type = “ com.iot.mybatis.po.Orders ” id = “ OrdersUserResultMap ” >
<！ -配置映射的订单信息- > <！ - id：指定查询列中的唯一标识，订单信息的中的唯一标识，如果有多个列组成唯一标识，配置多个id 列：订单信息的唯一标识列 属性：订单信息的唯一标识列所映射到Orders中哪个属性- >
< id column = “ id ” property = “ id ” />
< result column = “ user\_id ” property = “ userId ” />
< result column = “ number ” property = “ number ” />
< result column = “ createtime ” property = “ createtime ” />
< result column = “ note ” property = “ note ” />
<！ -配置映射的关联的用户信息- >
<！ - association：用于映射关联查询
单个
对象的信息属性：要将关联查询的用户信息映射到Orders中哪个属性- >
< association 属性 = “ user ” javaType = “ com.iot.mybatis.po.User ” >
<！ - id：关联查询用户的唯一标识 栏：指定唯一标识用户信息的列 javaType：映射到用户的哪个属性- - >
< id column = “ user\_id ” property = “ id ” />
< result column = “ username ” property = “ username ” />
< result column = “ sex ” property = “ sex ” />
< result column = “ address ” property = “ address ” />
</ association >
</ resultMap >

声明定义

<！ -查询订单关联查询用户信息- >
< select id = “ findOrdersUserResultMap ” resultMap = “ OrdersUserResultMap ” >
选择
命令。\*，
user.username，
user.sex，
user.address
从
命令，
用户
WHERE orders.user\_id = user.id
</ select >

mapper.java

//查询订单关联查询用户使用resultMap
public List < Orders > findOrdersUserResultMap（）抛出异常 ;

测试代码

@Test
public void testFindOrdersUserResultMap（）throws Exception {
SqlSession sqlSession = sqlSessionFactory 。的openSession（）;
//创建代理对象
OrdersMapperCustom ordersMapperCustom = sqlSession
.getMapper（OrdersMapperCustom 。类）;
//调用maper的方法
List < Orders > list = ordersMapperCustom。findOrdersUserResultMap（）;
系统。出。的println（列表）;
sqlSession 。关（）;
}

与resultType和结果映射实现一对一查询小结

实现一对一查询：

与resultType：使用与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，即可完成映射如果没有查询结果的特殊要求建议使用与resultType。

结果映射：需要单独定义的结果映射，实现有点麻烦，如果对查询结果有特殊的要求，使用结果映射可以完成将关联查询映射POJO的属性中。

结果映射可以实现延迟加载，与resultType无法实现延迟加载