

案例：一个订单信息只能对应一个用户

订单表OrdersMapper.xml映射的写法：**resultMap**方式实现

1、查询所有订单信息对应的所有用户信息，返回一个map结果

2、采用内连接关联查询，返回map集合

```
<select id="findOrdersWithUserResultMap" resultMap="ordersResultMap">
```

```
    SELECT
```

```
    orders.*,
```

```
    user.* FROM orders,
```

```
    USER
```

```
    WHERE orders.user_id = user.id
```

```
</select>
```

3、定义关系映射，把订单映射订单javaBean属性

```
<!-- type:指定映射javaBean类型    id:resultMap唯一标识    -->
```

```
<resultMap type="orders" id="ordersResultMap">
```

```
<!-- 定义查询订单列名和订单javaBean属性一一对应映射关系 -->
```

```
<!-- id:定义主键映射    column:查询数据库列名    property: javaBean属性-->
```

```
<id column="id" property="id"/>
```

```
<!-- result:定义普通属性映射    column:查询数据库显示列名    property: javaBean属性名称-->
```

```
<result column="user_id" property="userId"/>
```

```
<result column="number" property="number"/>
```

```
<result column="createtime" property="createTime"/>
```

```
<result column="note" property="note"/>
```

```
<!-- 一个订单属于一个用户：一对一关系 -->
```

```
<!-- association:表示一对一关系映射    property:表示查询数据映射到orders中那个属性
javaType:映射需要映射对象类型-->
```

```
<association property="user" javaType="user">
```

<!-- 主键映射 -->

```
<id column="user_id" property="id"/>
```

<!-- 普通属性映射 -->

```
<result column="username" property="username"/>
```

```
<result column="birthday" property="birthday"/>
```

```
<result column="sex" property="sex"/>
```

```
<result column="address" property="address"/>
```

```
</association>
```

```
</resultMap>
```

resultType方式实现：这种方式需要有一个类同时拥有两张表的所有属性

创建类继承订单表并将用户表所有数据写入

```
/**
 * 继承订单类，将用户数据写入
 */
public class OrdersCustom extends Orders{
```

```
    private String username;
    private Date birthday;
    private String sex;
    private String address;
```

用户的属性

```
    public String getUsername() {
        return username;
    }
}
```

<!--

需求:查询订单，关联查询出此订单的用户

关系：一对一关系

关系映射：ResultType

-->

```
<select id="findOrdersWithUserResultType" resultType="ordersCustom">
```

```
SELECT
```

```
orders.id,
```

```
orders.user_id userId,
```

```
orders.number,
```

```
orders.createtime createTime,
```

```
orders.note,
```

```
user.* FROM orders,
```

```
USER
```

```
WHERE orders.user_id = user.id
```

```
</select>
```