

MyBatis高级

王新阳

wxyyuppie@bjfu.edu.cn



主要内容

- **MyBatis级联查询**



级联查询——一对一级联

通过<resultMap>元素的子元素<association>处理这种一对一级联关系。在<association>元素中，通常使用以下属性。

property: 指定映射到实体类的对象属性。

column: 指定表中对应的字段（即查询返回的列名）。

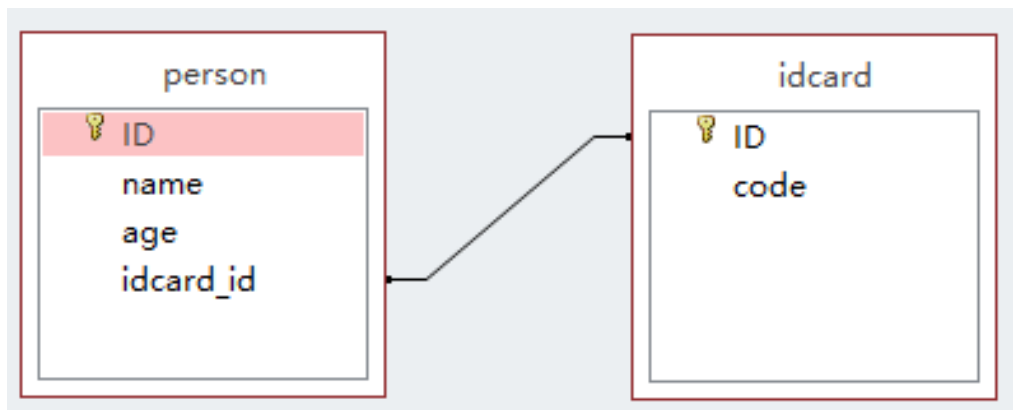
javaType: 指定映射到实体对象属性的类型。

select: 指定引入嵌套查询的子SQL语句，该属性用于关联映射中的嵌套查询。



级联查询——一对一级联

ER图



类图



见工程ch7
com.po.Person

两张数据表，一张是个人信息表person，
一张是身份证表idcard，具有一对一关系

```
public class Person {
    private Integer id;
    private String name;
    private Integer age;
    private Idcard card; // 个人身份证关联
    // getters and setters ...
}
```

```
public class Idcard {
    private Integer id;
    private String code;
    // getters and setters
}
```



级联查询——一对一级联

查询出所有个人信息以及所持有银行卡信息

SQL语句怎么写

SELECT

person.id,
person.`name`,
person.age,
idcard.`code`

FROM

person

LEFT JOIN idcard ON **person.idcard_id = idcard.id**



查询结果

id	name	age	code
1	陈恒	88	123456789123456789



级联查询——一对一级联

resultMap实现1——嵌套查询，执行两个SQL

```
<mapper namespace="com.dao.IdCardDao">
  <select id="selectCodeById" parameterType="Integer" resultType="com.po.Idcard">
    select * from idcard where id=#{id}
  </select>
</mapper>
```

IdCardMapper.xml

```
<!-- 一对一 根据id查询个人信息：第一种方法（嵌套查询） -->
<resultMap type="com.po.Person" id="cardAndPerson1">
  <id property="id" column="id"/>
  <result property="name" column="name"/>
  <result property="age" column="age"/>
  <!-- 一对一关联查询 -->
  <association property="card" column="idcard id" javaType="com.po.Idcard"
    select="com.dao.IdCardDao.selectCodeById"/>
</resultMap>
<select id="selectPersonById1" parameterType="Integer" resultMap="cardAndPerson1">
  select * from person where id=#{id}
</select>
```

PersonMapper.xml

```
public class Person {
  private Integer id;
  private String name;
  private Integer age;
  private Idcard card; //个人身份证关联
  //getters and setters ...
}
```

相当于指定连接
查询的外键



级联查询——一对一级联

resultMap实现2——嵌套结果，执行一个SQL

```
<!-- 一对一 根据id查询个人信息：第二种方法（嵌套结果） -->
<resultMap type="com.po.Person" id="cardAndPerson2">
  <id property="id" column="id"/>
  <result property="name" column="name"/>
  <result property="age" column="age"/>
  <!-- 一对一关联查询 -->
  <association property="card" javaType="com.po.Idcard">
    <id property="id" column="idcard_id"/>
    <result property="code" column="code"/>
  </association>
</resultMap>
```

PersonMapper.xml

将POJO类Idcard中的id字段与
person表中的外键idcard_id关联

```
<select id="selectPersonById2" parameterType="Integer" resultMap="cardAndPerson2">
  select p.*,ic.code
  from person p, idcard ic
  where p.idcard_id = ic.id and p.id=#{id}
</select>
```



级联查询——一对一级联

resultMap实现3——根据id查询，使用POJO 存储结果

PersonMapper.xml

```
<!-- 一对一 根据id查询个人信息：第三种方法（使用POJO存储结果） -->
<select id="selectPersonById3" parameterType="Integer"
        resultType="com.pojo.SelectPersonById">
    select p.*,ic.code
    from person p, idcard ic
    where p.idcard_id = ic.id and p.id=#{id}
</select>
```

```
public class SelectPersonById {
    private Integer id;
    private String name;
    private Integer age;
    private String code;
    //getters and setters...
}
```

POJO类



级联查询——一对一级联

测试入口

com.controller.OneToOneController

```
public void test() {  
    Person p1 = personDao.selectPersonById1(1);  
    System.out.println(p1);  
    System.out.println("=====");  
    Person p2 = personDao.selectPersonById2(1);  
    System.out.println(p2);  
    System.out.println("=====");  
    SelectPersonById p3 = personDao.selectPersonById3(1);  
    System.out.println(p3);  
}
```



级联查询——一对多级联

通过 `<resultMap>` 元素的子元素 `<collection>` 处理这种一对多级联关系。在 `<collection>` 元素中，通常使用以下属性。

- property**: 指定映射到实体类的对象属性。
- column**: 指定表中对应的字段（即查询返回的列名）。
- ofType**: 指定映射到实体对象属性的列表或其他容器的元素类型。
- select**: 指定引入嵌套查询的子SQL语句，该属性用于关联映射中的嵌套查询。

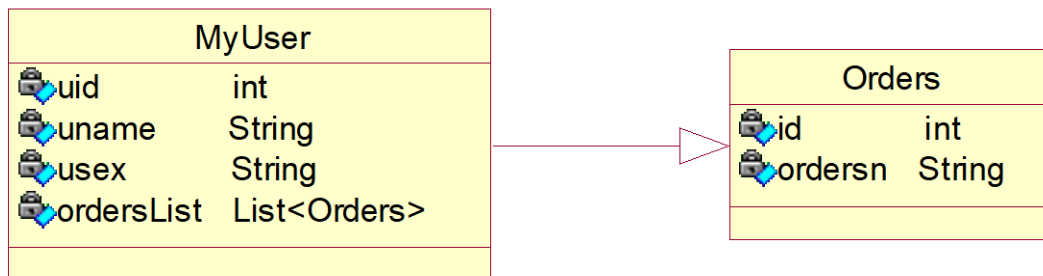
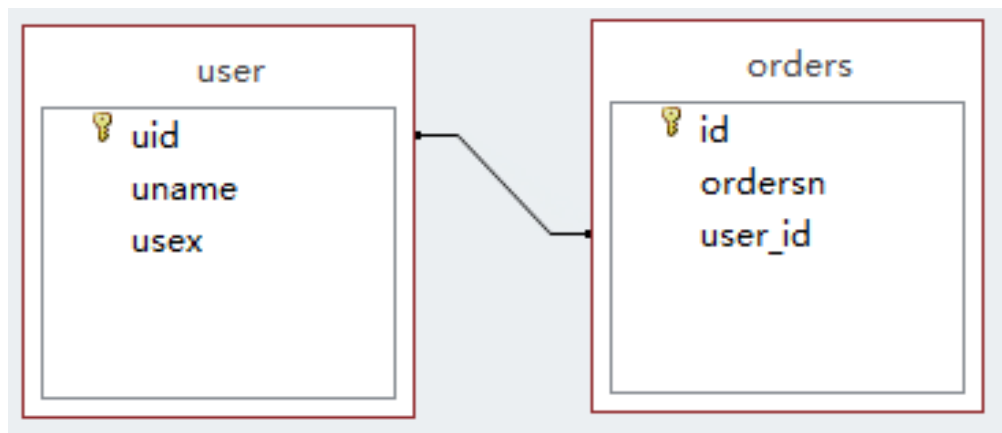


级联查询——一对多级联

见工程ch7

两张数据表，一张是用户表user，一张是订单表orders，具有一对多关系

2021- <div></div>		订单号: 2140- <div></div>		您订单中的商品在不同库房或属不同商家, 故拆分为以下订单分开配送, 给您带来的不便敬请谅解。					
收货人: <div></div>		订单金额: ¥2- <div></div>		支付方式: 在线支付		订单状态: 已拆分		查看拆分详情>	
2021- <div></div>		订单号: 21341- <div></div>		文轩旗舰店 		400-610-1360转115692			
	人工智能数学基础		x1	申请售后		¥1- <div></div>	已完成  +8	选购京东服务 查看发票 追评	
 找搭配						在线支付	订单详情	 立即购买	
2021- <div></div>		订单号: 2134122- <div></div>		互动出版图书专营店 					
	现货包邮 TensorFlow 2.0神经网络实践 [8071294]		x1	申请售后		<div></div>	已完成  +5	选购京东服务 查看发票 追评	
 找搭配						在线支付	订单详情	 立即购买	
	电子书VIP畅读月卡 [7955088]		x1	申请售后					
 找搭配									
2021- <div></div>		订单号: 2134- <div></div>		京东 					
	深度学习(异步图书出品)		x1	申请售后		1- <div></div>	已完成  +10	选购京东服务 查看发票 追评	
 找搭配						在线支付	订单详情	 立即购买	



```
public class MyUser {  
    private Integer uid; //主键  
    private String uname;  
    private String usex;  
    //一对多关联查询, 用户关联的订单  
    private List<Orders> ordersList;  
    //getters and setters  
}
```

```
public class Orders {  
    private Integer id;  
    private String ordersn;  
    //getters and setters...  
}
```



级联查询——一对多级联

查询出用户id=1的个人信息及所有订单信息

SQL语句怎么写



SELECT

`user`.uid,
`user`.uname,
`user`.usex,
orders.id,
orders.ordersn

FROM

USER

JOIN orders ON **`user`.uid = orders.user_id**

WHERE

`user`.uid = 1

查询结果

uid	uname	usex	id	ordersn
1	张三	女	1	999999
1	张三	女	2	88888



级联查询——一对多级联

resultMap实现1——嵌套查询，执行两个SQL

```
<!-- 根据用户uid查询订单信息 -->
<select id="selectOrdersById" parameterType="Integer" resultType="com.po.Orders">
    select * from orders where user_id=#{id}
</select>
```

OrdersMapper.xml

```
<!-- 一对多 根据uid查询用户及其关联的订单信息，第一种方法（嵌套查询） -->
<resultMap type="com.po.MyUser" id="userAndOrders1">
    <id property="uid" column="uid"/>
    <result property="uname" column="uname"/>
    <result property="usex" column="usex"/>
    <!-- 一对多关联查询，ofType表示集合中的元素类型，将uid传递给selectOrdersById -->
    <collection property="ordersList" ofType="com.po.Orders" column="uid"
        select="com.dao.OrdersDao.selectOrdersById"/>
</resultMap>
<select id="selectUserOrdersById1" parameterType="Integer" resultMap="userAndOrders1">
    select * from user where uid = #{id}
</select>
```

UserMapper.xml

不是
javatype

将uid传递给
selectOrdersById

```
public class MyUser {
    private Integer uid; //主键
    private String uname;
    private String usex;
    //一对多关联查询，用户关联的订单
    private List<Orders> ordersList;
    //getters and setters
}
```



级联查询——一对多级联

resultMap实现2——嵌套结果，执行一个SQL

UserMapper.xml

```
<!-- 一对多 根据uid查询用户及其关联的订单信息：第二种方法（嵌套结果） -->
<resultMap type="com.po.MyUser" id="userAndOrders2">
  <id property="uid" column="uid" />
  <result property="uname" column="uname" />
  <result property="usex" column="usex" />
  <!-- 一对多关联查询，ofType表示集合中的元素类型 -->
  <collection property="ordersList" ofType="com.po.Orders">
    <id property="id" column="id" />
    <result property="ordersn" column="ordersn" />
  </collection>
</resultMap>
<select id="selectUserOrdersById2" parameterType="Integer"
  resultMap="userAndOrders2">
  select u.*,o.id,o.ordersn from user u, orders o where u.uid
  = o.user_id and u.uid=#{id}
</select>
```

不是
association



级联查询——一对多级联

resultMap实现3——根据id查询，使用POJO 存储结果

UserMapper.xml

```
<!-- 一对多 根据uid查询用户及其关联的订单信息：第三种方法（使用POJO存储结果） -->
<select id="selectUserOrdersById3" parameterType="Integer"
        resultMap="com.pojo.SelectUserOrdersById">
    select u.*,o.id,o.ordersn from user u, orders o where u.uid
    = o.user_id and u.uid=#{id}
</select>
```

```
public class SelectUserOrdersById {
    private Integer uid;
    private String uname;
    private String usex;
    private Integer id;
    private String ordersn;
    //getters and setters...
}
```

POJO类



级联查询——一对多级联

测试入口

com.controller.OneToMoreController

```
public void test() {  
    //查询一个用户及订单信息  
    MyUser auser1 = userDao.selectUserOrdersById1(1);  
    System.out.println(auser1);  
    System.out.println("=====");  
    MyUser auser2 = userDao.selectUserOrdersById2(1);  
    System.out.println(auser2);  
    System.out.println("=====");  
    List<SelectUserOrdersById> auser3 = userDao.selectUserOrdersById3(1);  
    System.out.println(auser3);  
    System.out.println("=====");  
}
```




感谢聆听

Thanks For Your Listening!