**只为成功找方法，不为失败找借口！**

[**MyBatis学习总结(四)——解决字段名与实体类属性名不相同的冲突**](http://www.cnblogs.com/xdp-gacl/p/4264425.html)

　　在平时的开发中，我们表中的字段名和表对应实体类的属性名称不一定都是完全相同的，下面来演示一下这种情况下如何解决字段名与实体类属性名不相同的冲突。

**一、准备演示需要使用的表和数据**

[复制代码](javascript:void(0);)

CREATE TABLE orders(

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_no VARCHAR(**20**),

order\_price FLOAT

);

INSERT INTO orders(order\_no, order\_price) VALUES('aaaa', **23**);

INSERT INTO orders(order\_no, order\_price) VALUES('bbbb', **33**);

INSERT INTO orders(order\_no, order\_price) VALUES('cccc', **22**);

[复制代码](javascript:void(0);)

**二、定义实体类**

[复制代码](javascript:void(0);)

1 package me.gacl.domain;

2

3 /\*\*

4 \* @author gacl

5 \* 定义orders表对应的实体类

6 \*/

7 public class Order {

8 /\*\*

9 \*

10 CREATE TABLE orders(

11 order\_id INT PRIMARY KEY AUTO\_INCREMENT,

12 order\_no VARCHAR(20),

13 order\_price FLOAT

14 );

15 \*/

16

17 //Order实体类中属性名和orders表中的字段名是不一样的

18 private int id; //id===>order\_id

19 private String orderNo; //orderNo===>order\_no

20 private float price; //price===>order\_price

21

22 public int getId() {

23 return id;

24 }

25

26 public void setId(int id) {

27 this.id = id;

28 }

29

30 public String getOrderNo() {

31 return orderNo;

32 }

33

34 public void setOrderNo(String orderNo) {

35 this.orderNo = orderNo;

36 }

37

38 public float getPrice() {

39 return price;

40 }

41

42 public void setPrice(float price) {

43 this.price = price;

44 }

45

46 @Override

47 public String toString() {

48 return "Order [id=" + id + ", orderNo=" + orderNo + ", price=" + price+ "]";

49 }

50 }

[复制代码](javascript:void(0);)

**三、编写测试代码**

**3.1、编写SQL的xml映射文件**

　　1、创建一个orderMapper.xml文件，orderMapper.xml的内容如下：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8" ?>

2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

3 <!-- 为这个mapper指定一个唯一的namespace，namespace的值习惯上设置成包名+sql映射文件名，这样就能够保证namespace的值是唯一的

4 例如namespace="me.gacl.mapping.orderMapper"就是me.gacl.mapping(包名)+orderMapper(orderMapper.xml文件去除后缀)

5 -->

6 <mapper namespace="me.gacl.mapping.orderMapper">

7

8 <!--

9 根据id查询得到一个order对象，使用这个查询是查询不到我们想要的结果的，

10 这主要是因为实体类的属性名和数据库的字段名对应不上的原因，因此无法查询出对应的记录

11 -->

12 <select id="getOrderById" parameterType="int"

13 resultType="me.gacl.domain.Order">

14 select \* from orders where order\_id=#{id}

15 </select>

16

17 <!--

18 根据id查询得到一个order对象，使用这个查询是可以正常查询到我们想要的结果的，

19 这是因为我们将查询的字段名都起一个和实体类属性名相同的别名，这样实体类的属性名和查询结果中的字段名就可以一一对应上

20 -->

21 <select id="selectOrder" parameterType="int"

22 resultType="me.gacl.domain.Order">

23 select order\_id id, order\_no orderNo,order\_price price from orders where order\_id=#{id}

24 </select>

25

26 <!--

27 根据id查询得到一个order对象，使用这个查询是可以正常查询到我们想要的结果的，

28 这是因为我们通过<resultMap>映射实体类属性名和表的字段名一一对应关系 -->

29 <select id="selectOrderResultMap" parameterType="int" resultMap="orderResultMap">

30 select \* from orders where order\_id=#{id}

31 </select>

32 <!--通过<resultMap>映射实体类属性名和表的字段名对应关系 -->

33 <resultMap type="me.gacl.domain.Order" id="orderResultMap">

34 <!-- 用id属性来映射主键字段 -->

35 <id property="id" column="order\_id"/>

36 <!-- 用result属性来映射非主键字段 -->

37 <result property="orderNo" column="order\_no"/>

38 <result property="price" column="order\_price"/>

39 </resultMap>

40

41 </mapper>

[复制代码](javascript:void(0);)

　　2、在conf.xml文件中注册orderMapper.xml映射文件

<mappers>

<!-- 注册orderMapper.xml文件，

orderMapper.xml位于me.gacl.mapping这个包下，所以resource写成me/gacl/mapping/orderMapper.xml-->

<mapper resource="me/gacl/mapping/orderMapper.xml"/>

</mappers>

**3.2、编写单元测试代码**

[复制代码](javascript:void(0);)

1 package me.gacl.test;

2

3 import me.gacl.domain.Order;

4 import me.gacl.util.MyBatisUtil;

5 import org.apache.ibatis.session.SqlSession;

6 import org.junit.Test;

7

8 public class Test2 {

9

10 @Test

11 public void testGetOrderById(){

12 SqlSession sqlSession = MyBatisUtil.getSqlSession();

13 /\*\*

14 \* 映射sql的标识字符串，

15 \* me.gacl.mapping.orderMapper是orderMapper.xml文件中mapper标签的namespace属性的值，

16 \* getOrderById是select标签的id属性值，通过select标签的id属性值就可以找到要执行的SQL

17 \*/

18 String statement = "me.gacl.mapping.orderMapper.getOrderById";//映射sql的标识字符串

19 //执行查询操作，将查询结果自动封装成Order对象返回

20 Order order = sqlSession.selectOne(statement,1);//查询orders表中id为1的记录

21 //使用SqlSession执行完SQL之后需要关闭SqlSession

22 sqlSession.close();

23 System.out.println(order);//打印结果：null，也就是没有查询出相应的记录

24 }

25

26 @Test

27 public void testGetOrderById2(){

28 SqlSession sqlSession = MyBatisUtil.getSqlSession();

29 /\*\*

30 \* 映射sql的标识字符串，

31 \* me.gacl.mapping.orderMapper是orderMapper.xml文件中mapper标签的namespace属性的值，

32 \* selectOrder是select标签的id属性值，通过select标签的id属性值就可以找到要执行的SQL

33 \*/

34 String statement = "me.gacl.mapping.orderMapper.selectOrder";//映射sql的标识字符串

35 //执行查询操作，将查询结果自动封装成Order对象返回

36 Order order = sqlSession.selectOne(statement,1);//查询orders表中id为1的记录

37 //使用SqlSession执行完SQL之后需要关闭SqlSession

38 sqlSession.close();

39 System.out.println(order);//打印结果：Order [id=1, orderNo=aaaa, price=23.0]

40 }

41

42 @Test

43 public void testGetOrderById3(){

44 SqlSession sqlSession = MyBatisUtil.getSqlSession();

45 /\*\*

46 \* 映射sql的标识字符串，

47 \* me.gacl.mapping.orderMapper是orderMapper.xml文件中mapper标签的namespace属性的值，

48 \* selectOrderResultMap是select标签的id属性值，通过select标签的id属性值就可以找到要执行的SQL

49 \*/

50 String statement = "me.gacl.mapping.orderMapper.selectOrderResultMap";//映射sql的标识字符串

51 //执行查询操作，将查询结果自动封装成Order对象返回

52 Order order = sqlSession.selectOne(statement,1);//查询orders表中id为1的记录

53 //使用SqlSession执行完SQL之后需要关闭SqlSession

54 sqlSession.close();

55 System.out.println(order);//打印结果：Order [id=1, orderNo=aaaa, price=23.0]

56 }

57 }

[复制代码](javascript:void(0);)

　　执行单元测试的结果：

　　1、testGetOrderById方法执行查询后返回一个null。

　　2、testGetOrderById2方法和testGetOrderById3方法执行查询后可以正常得到想要的结果。

**四、总结**

　　上面的测试代码演示当实体类中的属性名和表中的字段名不一致时，使用MyBatis进行查询操作时无法查询出相应的结果的问题以及针对问题采用的两种办法：

**解决办法一: 通过在查询的sql语句中定义字段名的别名，让字段名的别名和实体类的属性名一致，这样就可以表的字段名和实体类的属性名一一对应上了，这种方式是通过在sql语句中定义别名来解决字段名和属性名的映射关系的。**

**解决办法二: 通过<resultMap>来映射字段名和实体类属性名的一一对应关系。这种方式是使用MyBatis提供的解决方式来解决字段名和属性名的映射关系的。**