# mybatis进阶详解

返回值

1）返回值： resultMap

能代替以前的resultType

能进行一对一，一对多，对多的关系映射

懒加载

<**resultMap id="resultMapStu" type="com.offcn.pojo.Student"**>  
 <**id property="sid" column="sid"**></**id**>  
 <**result property="sname" column="sname"** ></**result**>  
 <**result property="address" column="address"** ></**result**>  
</**resultMap**>

2）动态sql： 标签

<where> +<if>+<sql>

<**sql id="sql001"**>  
 SELECT te.*\** FROM teacher te  
</**sql**>  
  
<**sql id="sql002"**>  
 <**if test="tname != null"**>  
 AND te.`tname` = #{tname}  
 </**if**>  
</**sql**>  
  
<**sql id="sql003"**>  
 <**if test="address != null"**>  
 AND te.`address` = #{address}  
 </**if**>  
</**sql**>  
  
<**select id="getTeacherByNameAndAdd" parameterType="com.offcn.pojo.Teacher" resultType="com.offcn.pojo.Teacher"**>  
 <**include refid="sql001"**></**include**>  
 <**where**>  
 <**include refid="sql002"**></**include**>  
 <**include refid="sql003"**></**include**>  
 </**where**>  
</**select**>

<when>+<otherwise>+<choose>

<**select id="getTeacherByNameAndAddAndAge" parameterType="com.offcn.pojo.Teacher"  
 resultType="com.offcn.pojo.Teacher"**>  
 SELECT te.*\** FROM teacher te  
 <**where**>  
 <**choose**>  
 <**when test="tname != null"**>  
 te.`tname` = #{tname}  
 </**when**>  
 <**when test="address != null"**>  
 AND te.`address` = #{address}  
 </**when**>  
 *<!-- <when test="age != null">  
 AND te.`age`=#{age}  
 </when>-->* <**otherwise**>  
 AND te.`age`=#{age}  
 </**otherwise**>  
 </**choose**>  
 </**where**>  
</**select**>

模糊查询：

<**select id="selectTeacherLike" parameterType="com.offcn.pojo.Teacher" resultType="com.offcn.pojo.Teacher"**>  
 select *\** from teacher te  
 <**where**>  
 <**if test="tname != null"**>  
 and te.tname like '%${tname}%'  
 </**if**>  
 <**if test="address!= null"**>  
 and te.address like '%${address}%'  
 </**if**>  
 <**if test="age != null"**>  
 and te.age like '%${age}%'  
 </**if**>  
 </**where**>  
</**select**>

3）入参类型： int String map person @Parm List

*<!-- public List<Teacher> selectTeacherByIds(List<Integer> list);  
 入参和返回值要区分开  
 返回值是集合，我们就 resultType 就写泛型类型  
 入参：是什么就是什么类型 parameterType  
foreach遍历 数据（item）  
collection 值是list  
item代表是数据  
open 开头的字符串拼接  
close 结尾的字符串拼接  
separator 中间用什么分割  
 -->* <**select id="selectTeacherByIds" parameterType="java.util.List" resultType="com.offcn.pojo.Teacher"**>  
 SELECT *\** FROM teacher  
 WHERE tid  
 <**foreach collection="list" item="id" open="in( " close=" )" separator=","**>  
 #{id}  
 </**foreach**>  
 </**select**>  
  
 *<!--  
 public int insertTeachers(List<Teacher> tlist);  
 INSERT INTO teacher VALUES (2006,'laowang2','gebi',16)  
 这里不要用 open="( " close=" )"  
 item代表了teacher对象，可以自定义名字  
 -->* <**insert id="insertTeachers" parameterType="java.util.List"** >  
 INSERT INTO teacher VALUES

<**foreach collection="list" item="tea" separator=","**>  
 #{tea.tid},#{tea.tname},#{tea.address},#{tea.age}  
 </**foreach**>  
 </**insert**>

1. 一对一

一对多

前奏知识点：继承

<**resultMap id="map001" type="com.offcn.pojo.Teacher"**>  
 <**id property="tid" column="ttid"**></**id**>  
 <**result property="tname" column="ttname"**></**result**>  
</**resultMap**>  
  
<**resultMap id="map002" type="com.offcn.pojo.Teacher" extends="map001"**>  
 <**result property="address" column="taddress"**></**result**>  
</**resultMap**>

**一对一：三种方式**

## **方式一：**

类中代码：

card类

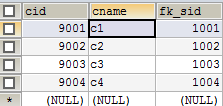
**public class** Card {  
 **private int cid** ;  
 **private** String **cname** ;  
 **private int fk\_sid** ;  
 **private** Student **stu** ; *//一个身份证对应一个学生*

student类

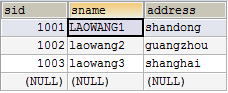
**public class** Student {  
 **private int sid** ;  
 **private** String **sname** ;  
 **private** String **address**;

表关系如下

card：



student：



配置文件： CardMapper.xml

前两种的配置：

*<!-- 一对一：第一种方式-->*<**resultMap id="map001" type="com.offcn.pojo.Card"**>  
 <**id column="cid" property="cid"**></**id**>  
 <**result property="cname" column="cname"** ></**result**>  
 *<!--<result property="fk\_sid" column="fk\_sid" ></result>-->  
 <!--***association 级联：用于 一对一  
 property 级联的属性对象名 Student stu  
 column 这个级联对象（stu）所对应的的表中外键** *-->* <**association property="stu" column="fk\_sid" javaType="com.offcn.pojo.Student"** >  
 <**id property="sid" column="sid"** ></**id**>  
 <**result property="sname" column="sname"**></**result**>  
 <**result property="address" column="address"**></**result**>  
 </**association**>  
</**resultMap**>  
  
第二种方式<**resultMap id="map002" type="com.offcn.pojo.Card"**>  
 <**id column="cid" property="cid"**></**id**>  
 <**result property="cname" column="cname"** ></**result**>  
 <**association property="stu" column="fk\_sid" javaType="com.offcn.pojo.Student"  
 resultMap="stuMap001yidongguolaide"** >  
 </**association**>  
</**resultMap**>  
  
<**resultMap id="stuMap001yidongguolaide" type="com.offcn.pojo.Student"**>  
 <**id property="sid" column="sid"** ></**id**>  
 <**result property="sname" column="sname"**></**result**>  
 <**result property="address" column="address"**></**result**>  
</**resultMap**>

**第一种和第二种使用共同的sql**

<**select id="getCardById" parameterType="int" resultMap="map002"**>  
  
SELECT s.*\**,c.*\**FROM student s , card c  
WHERE s.`sid` = c.`fk\_sid`  
AND c.`cid` = #{id}  
  
</**select**>

**一对一的第三种方式：**

CardMapper.xml配置：

*<!--一对一： 第三种方式-->*<**resultMap id="map003" type="com.offcn.pojo.Card"**>  
 <**id column="cid" property="cid"**></**id**>  
 <**result property="cname" column="cname"** ></**result**>  
 *<!--  
 select表示，当执行完id="getCardByIdOneToOne"的时候，立马去调用  
 com.offcn.mapper.StudentMapper.selectStudentById 这个方法立即执行  
 -->* <**association property="stu" column="fk\_sid" javaType="com.offcn.pojo.Student"  
 select="com.offcn.mapper.StudentMapper.selectStudentById"**

**<!--上面这个路径是studentmap的id-->**

>  
 </**association**>  
</**resultMap**>

*<!--第三种方式sql如下  
 public Card getCardByIdOneToOne(int id);  
 -->*<**select id="getCardByIdOneToOne" parameterType="int" resultMap="map003"**>  
 select *\** from card where cid= #{id}  
</**select**>

一对多的三种方式

前两种方式：

类：

student类:

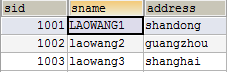
**public class** Student {  
 **private int sid** ;  
 **private** String **sname** ;  
 **private** String **address**;  
 **private** List<Book> **bookList**;

book类:

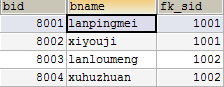
**public class** Book {  
 **private int bid** ;  
 **private** String **bname** ;  
 **private int fk\_sid** ;

数据库结构

student表



book表



StdeuntMapper.xml中的配置如下：

前两种方式： 一对多的第一种方式

<**resultMap id="stuMap002" type="com.offcn.pojo.Student"**>  
 <**id property="sid" column="sid"** ></**id**>  
 <**result property="sname" column="sname"**></**result**>  
 <**result property="address" column="address"**></**result**>  
 *<!--*  **collection 代表了一对多的级联  
 下列的column对应当前对象书库表的字段  
 javatype换成了oftype** *-->*

**<!--booklist是Student实体类中的虚拟对象 oftype是这个对象的返回值类型-->**

<**collection property="bookList" column="sid" ofType="com.offcn.pojo.Book"** >  
 <**id property="bid" column="bid"**></**id**>  
 <**result property="bname" column="bname"**></**result**>  
 *<!--<result property="fk\_sid" column="fk\_sid"></result>-->* </**collection**>  
 </**resultMap**>

一对多的第二种方式<**resultMap id="stuMap003" type="com.offcn.pojo.Student"**>  
 <**id property="sid" column="sid"** ></**id**>  
 <**result property="sname" column="sname"**></**result**>  
 <**result property="address" column="address"**></**result**>  
 <**collection property="bookList" column="sid" ofType="com.offcn.pojo.Book" resultMap="bookResultMap"**>  
  
 </**collection**>  
 </**resultMap**>  
  
<**resultMap id="bookResultMap" type="com.offcn.pojo.Book"**>  
 <**id property="bid" column="bid"**></**id**>  
 <**result property="bname" column="bname"**></**result**>  
</**resultMap**>

*<!--  
 一对多前两种使用共同的sql  
 public Student getStubyId(int id);  
 -->* <**select id="getStubyId" parameterType="int" resultMap="stuMap003"**>  
SELECT stu.*\**,bk.*\**FROM student stu , book bk  
WHERE stu.`sid` = bk.`fk\_sid`  
AND stu.`sid` = #{id}  
 </**select**>

第三种方式如下：  
*<!--  
一对多 第三种方式  
-->*<**resultMap id="stuMap004" type="com.offcn.pojo.Student"**>  
 <**id property="sid" column="sid"** ></**id**>  
 <**result property="sname" column="sname"**></**result**>  
 <**result property="address" column="address"**></**result**>  
 <**collection property="bookList" column="sid" ofType="com.offcn.pojo.Book"  
 select="com.offcn.mapper.BookMapper.selectBookByStudentId"** >  
 </**collection**>  
</**resultMap**>

**第三种方式所使用的sql如下：**  
 *<!-- public Student getStubyIdOneToMany(int id);-->*<**select id="getStubyIdOneToMany" parameterType="int" resultMap="stuMap004"**>  
 select *\** from student where sid = #{id}  
</**select**>