目录

[总述： 1](#_Toc528941134)

[一、 if判断&where 1](#_Toc528941135)

[二、 trim 2](#_Toc528941136)

[三、 choose（switch-case） 2](#_Toc528941137)

[四、 set 3](#_Toc528941138)

[五、 foreach 4](#_Toc528941139)

[六、 批量保存 4](#_Toc528941140)

[MySQL下批量保存 4](#_Toc528941141)

[Oracle数据库批量保存： 5](#_Toc528941142)

[七、 两个内置参数： 7](#_Toc528941143)

[八、 抽取可重用的sql片段 8](#_Toc528941144)

## 总述：

* **if:判断**
* **choose (when, otherwise):分支选择；带了break的swtich-case**
* **trim 字符串截取(where(封装查询条件), set(封装修改条件))**
* **foreach 遍历集合**

## if判断&where

**查询员工，要求，携带了哪个字段查询条件就带上这个字段的值**

**public List<Employee> getEmpsByConditionIf(Employee employee);**

<select id=*"getEmpsByConditionIf"* resultType=*"com.atguigu.mybatis.bean.Employee"*>

select \* from tbl\_employee

where会自动去掉第一个多余的and

<where>

**test：判断表达式（OGNL）从参数中取值进行判断**

**遇见特殊符号应该去写转义字符：&&：&amp;&amp;**

<if test=*"lastName!=null &amp;&amp; lastName!=&quot;&quot;"*>

and last\_name like #{lastName}

</if>

**可以调用方法**

<if test=*"email!=null and email.trim()!=&quot;*

*&quot;"*>

and email=#{email}

</if>

**ognl会进行字符串与数字的转换判断 "0"==0**

<if test=*"gender==0 or gender==1"*>

and gender=#{gender}

</if>

</where>

</select>

## trim

**public List<Employee> getEmpsByConditionChoose(Employee employee);**

<select id=*"getEmpsByConditionTrim"* resultType=*"com*

*.atguigu.mybatis.bean.Employee"*>

select \* from tbl\_employee

**后面多出的and或者or where标签不能解决**

**prefix="":前缀：trim标签体中是整个字符串拼串 后的结果。**

**prefix给拼串后的整个字符串加一个前缀**

**prefixOverrides="":**

**前缀覆盖： 去掉整个字符串前面多余的字符**

**suffix="":后缀**

**suffix给拼串后的整个字符串加一个后缀**

**suffixOverrides=""**

**后缀覆盖：去掉整个字符串后面多余的字符**

<!-- 自定义字符串的截取规则 -->

<trim prefix=*"where"* suffixOverrides=*"and"*>

<if test=*"id!=null"*>

id=#{id} and

</if>

</trim>

</select>

## choose（switch-case）

**public List<Employee> getEmpsByConditionChoose**

**(Employee employee);**

<select id=*"getEmpsByConditionChoose"* resultType=*"com.atguigu.mybatis.bean.Employee"*>

select \* from tbl\_employee

<where>

**如果带了id就用id查，如果带了lastName就用lastName查;只会进入其中一个**

<choose>

<when test=*"id!=null"*>

id=#{id}

</when>

<when test=*"lastName!=null"*>

last\_name like #{lastName}

</when>

<otherwise>

gender = 0

</otherwise>

</choose>

</where>

</select>

## set

**public void updateEmp(Employee employee);**

<update id=*"updateEmp"*>

**Set标签的使用**

update tbl\_employee

<set>

<if test=*"lastName!=null"*>

last\_name=#{lastName},

</if>

<if test=*"email!=null"*>

email=#{email},

</if>

</set>

where id=#{id}

<!--

Trim：更新拼串

update tbl\_employee

<trim prefix="set" suffixOverrides=",">

<if test="lastName!=null">

last\_name=#{lastName},

</if>

<if test="email!=null">

email=#{email},

</if>

</trim>

where id=#{id} -->

</update>

## foreach

**public List<Employee> getEmpsByConditionForeach**

**(List<Integer> ids);**

<select id=*"getEmpsByConditionForeach"* resultType=*"com.atguigu.mybatis.bean.Employee"*>

select \* from tbl\_employee

**collection：指定要遍历的集合：**

**list类型的参数会特殊处理封装在map中，map的key就叫list**

**item：将当前遍历出的元素赋值给指定的变量**

**separator:每个元素之间的分隔符**

**open：遍历出所有结果拼接一个开始的字符**

**close:遍历出所有结果拼接一个结束的字符**

**index:索引。遍历list的时候是index就是索引，item就是当前值**

**遍历map的时候index表示的就是map的key，item就是map的值**

**#{变量名}就能取出变量的值也就是当前遍历出的元素**

<foreach collection=*"ids"* item=*"item\_id"* separator=*","*

open=*"where id in("* close=*")"*>

#{item\_id}

</foreach>

</select>

## 批量保存

### MySQL下批量保存

**public void addEmps(@Param("emps")List<Employee> emps);**  **MySQL下批量保存：可以foreach遍历 mysql支持values(),(),()语法**

<insert id=*"addEmps"*>

insert into tbl\_employee(

<include refid=*"insertColumn"*></include>

)

values

<foreach collection=*"emps"* item=*"emp"* separator=*","*>

(#{emp.lastName},#{emp.email},#{emp.gender},#{emp.dept.id})

</foreach>

</insert>

**这种方式需要数据库连接属性allowMultiQueries=true；**

*jdbc:mysql://localhost:3306/mybatis?allowMultiQueries=true*

**这种分号分隔多个sql可以用于其他的批量操作（删除，修改）**

<!-- <insert id="addEmps">

<foreach collection="emps" item="emp" separator=";">

insert into tbl\_employee(last\_name,email,gender,d\_id)

values(#{emp.lastName},#{emp.email},#{emp.gender},#{emp.dept.id})

</foreach>

</insert> -->

### Oracle数据库批量保存：

Oracle不支持values(),(),()

Oracle支持的批量方式

1、多个insert放在begin - end里面

begin

insert into employees(employee\_id,last\_name,email)

values(employees\_seq.nextval,'test\_001','test\_001@atguigu.com');

insert into employees(employee\_id,last\_name,email)

values(employees\_seq.nextval,'test\_002','test\_002@atguigu.com');

end;

2、利用中间表：

insert into employees(employee\_id,last\_name,email)

select employees\_seq.nextval,lastName,email from(

select 'test\_a\_01' lastName,'test\_a\_e01' email from dual

union

select 'test\_a\_02' lastName,'test\_a\_e02' email from dual

union

select 'test\_a\_03' lastName,'test\_a\_e03' email from dual

)

-->

<insert id=*"addEmps"* databaseId=*"oracle"*>

**oracle第一种批量方式**

<!-- <foreach collection="emps" item="emp" open="begin" close="end;">

insert into employees(employee\_id,last\_name,email)

values(employees\_seq.nextval,#{emp.lastName},#{emp.email});

</foreach> -->

**oracle第二种批量方式**

insert into employees(

<!-- 引用外部定义的sql -->

<include refid=*"insertColumn"*>

<property name=*"testColomn"* value=*"abc"*/>

</include>

)

<foreach collection=*"emps"* item=*"emp"* separator=*"union"*

open=*"select employees\_seq.nextval,lastName,email from("*

close=*")"*>

select #{emp.lastName} lastName,#{emp.email} email from dual

</foreach>

</insert>

## 两个内置参数：

不**只是方法传递过来的参数可以被用来判断，取值。。。**

**mybatis默认还有两个内置参数：**

**\_parameter:代表整个参数**

单个参数：\_parameter就是这个参数

多个参数：参数会被封装为一个map；\_parameter就是代表这个map

**\_databaseId:如果配置了databaseIdProvider标签。**

\_databaseId就是代表当前数据库的别名oracle

-->

<!--public List<Employee> getEmpsTestInnerParameter(Employee employee); -->

<select id=*"getEmpsTestInnerParameter"* resultType=*"com.atguigu.mybatis.bean.Employee"*>

<!-- bind：可以将OGNL表达式的值绑定到一个变量中，方便后来引用这个变量的值 -->

<bind name=*"\_lastName"* value=*"'%'+lastName+'%'"*/>

<if test=*"\_databaseId=='mysql'"*>

select \* from tbl\_employee

<if test=*"\_parameter!=null"*>

where last\_name like #{lastName}

</if>

</if>

<if test=*"\_databaseId=='oracle'"*>

select \* from employees

<if test=*"\_parameter!=null"*>

where last\_name like #{\_parameter.lastName}

</if>

</if>

</select>

## 抽取可重用的sql片段

**抽取可重用的sql片段。方便后面引用**

**1、sql抽取：经常将要查询的列名，或者插入用的列名抽取出来方便引用**

**2、include来引用已经抽取的sql：**

**3、include还可以自定义一些property，sql标签内部就能使用自定义的属性**

**include-property：取值的正确方式${prop},**

**#{不能使用这种方式}**

<sql id=*"insertColumn"*>

<if test=*"\_databaseId=='oracle'"*>

employee\_id,last\_name,email

</if>

<if test=*"\_databaseId=='mysql'"*>

last\_name,email,gender,d\_id

</if>

</sql>