**动态Sql概念**

MyBatis 的强大特性之一便是它的动态 SQL。如果你有使用 JDBC 或其他类似框架的经验，你就能体会到根据不同条件拼接 SQL 语句（**apend拼接sql**）有多么痛苦。拼接的时候要确保不能忘了必要的空格，还要注意省掉列名列表最后的逗号。**利用动态 SQL 这一特性可以彻底摆脱这种痛苦。**

**解决方案：在xml的sql代码中，根据条件参数，来动态拼接sql**

动态 SQL 元素（各种xml标记）和使用 JSTL 或其他类似基于 XML 的文本处理器**相似**。在 MyBatis 之前的版本中,有很多的元素需要来了解。MyBatis 3 大大提升了它们,现在用不到原先一半的元素就可以了。MyBatis 采用功能强大的基于 OGNL 的表达式来消除其他元素。

**动态Sql拼接的标记**

**test条件的设置**

**!= == >= > < <= && and || or eq neq ...**

**== （等价于eq-比较内容）**

**bind标记 对数据进行（ognl）运算得到新的参数**

<bind name="aa" value="user\_name.length()"/>

<bind name="bb" value="user\_age+50/>

<bind name="pattern" value="'%' + username + '%'" />

<bind name="age11" value='name1.toString().length()'/>

**if标记 单分支语句，类似于java的if**

<select id="findActiveBlogLike" resultType="Blog">

SELECT \* FROM BLOG WHERE state = ‘ACTIVE’

<if test="author != null and author.name != null">

AND author\_name like #{author.name}

</if>

</select>

**choose标记 (when, otherwise) 多分支语句，类似于java的if，else，if**

<select id="findActiveBlogLike"

resultType="Blog">

SELECT \* FROM BLOG WHERE state = ‘ACTIVE’

<choose>

<when test="title != null">

AND title like #{title}

</when>

<when test="author != null and author.name != null">

AND author\_name like #{author.name}

</when>

<otherwise>

AND featured = 1

</otherwise>

</choose>

</select>

**foreach标记 循环语句，类似于java的foreach**

<select id="selectPostIn" resultType="domain.blog.Post">

SELECT \*

FROM POST P

WHERE ID in

<foreach item="item" index="index" collection="list"

open="(" separator="," close=")">

#{item}

</foreach>

</select>

**mybatis中使用Ognl表达式**

**ognl常用表达式语法**

e1 or e2 或者

e1 and e2 并且

e1 == e2,e1 eq e2 都是比较内容-相等

e1 != e2,e1 neq e2 都是比较内容-不相等

e1 lt e2：小于

e1 lte e2：小于等于，其他gt（大于）,gte（大于等于）

e1 in e2 在一个集合

e1 not in e2 不在一个集合

e1 + e2,e1 \* e2,e1/e2,e1 - e2,e1%e2

!e,not e：非，求反

e.method(args)调用对象方法

e.property对象属性值

e1[ e2 ]按索引取值，List,数组和Map

@class@method(args)调用类的静态方法

@class@field调用类的静态字段值

**ognl使用位置**

a）动态SQL表达式中，if的test，bind的value等....

b）${param}参数中

**ognl使用例子1（模糊查询）**

**a）mysql自己连接-字符串**

<if test="name != null and name != ''">

name like concat('%', #{name}, '%')

</if>

**b）bind连接-字符串**

<bind name="nameLike" value="'%' + name + '%'"/>

<if test="name != null and name != ''">

name like #{nameLike}

</if>

**c）$连接-字符串**

<if test="name != null and name != ''">

name like '${'%' + name + '%'}'

</if>