动态SQL操作

# if标签

功能：可以对输入的参数进行判断。

利用**test**指定判断表达式，如test=“name != null AND name != ’’ ”。

（至于第一个用AND，是因为where可以自动检测并去除第一个多余的AND）



if标签可以**嵌套**使用。

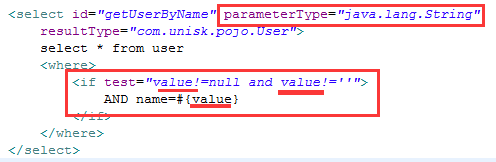
# where标签

可以默认去除后面的第一个AND，并且如果后面内容为空，自己默认把自己去除，即没有where语句了。

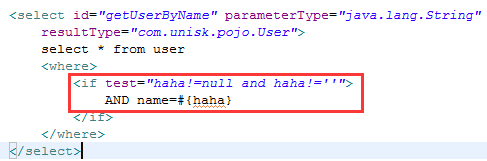
**if标签/where标签联合使用示例：**

**where 范围内添加条件，相当于WHERE #### ，AND表示连接。如果条件判断都不合适，则自动删除WHERE ，执行前面的语句。**

### 如果输入映射为简单类型，则可以利用value指示传入的值。也可以在接口方法中利用注解方式，对其起个名字：@param(“xxx”)

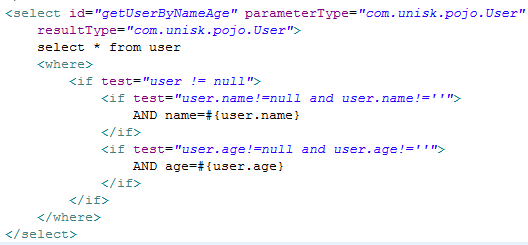






### 如果输入映射为POJO时，同样利用@Param(“xxx”)方式





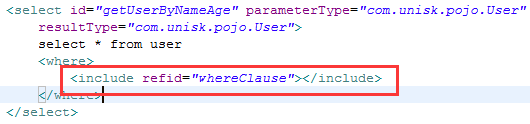
利用where、if标签可以实现很多需求，如果传入的User中只有name值，则就是getUserByName；如果只有age，则getUserByAge;如果都有就是getUserByNameAge;如果都没有，则是getAllUsers。

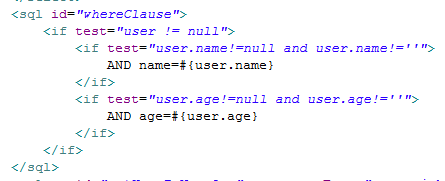
# sql片段：可以定义SQL语句中的任何部分。

作用：**就是提高重复利用率，降低映射文件的冗余。**

**最好不要把select和where等关键字声明在sql片段中，这样会降低给sql片段的重复利用率。**

**在SQL中引入sql片段，利用<include refid=“sqlID”></include> 引入。**

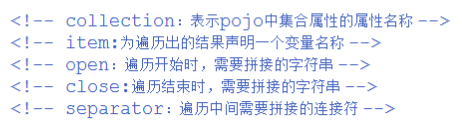




# foreach标签

可以循环传入参数。当参数是一个集合时，使用。

foreach中的属性意义：



**通常在in(xx,xx,xx)时使用foreach标签。**

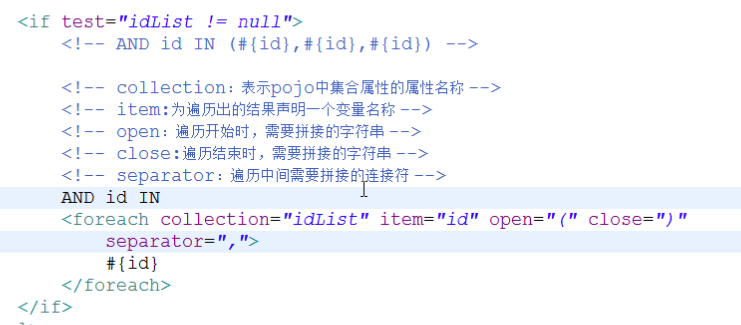
**掌握两种情况：**

## 直接传递List集合：parameterType=”java.util.List”

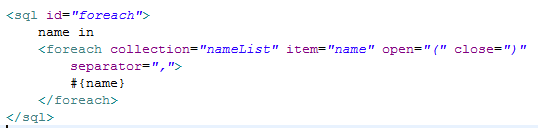
## 传递POJO中的List集合。

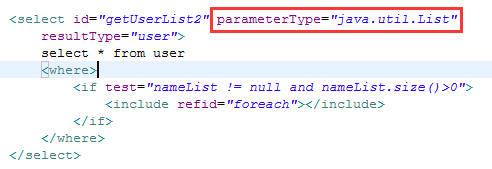
**示例：**





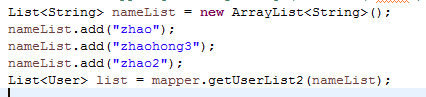
示例2：**直接传递List集合**。





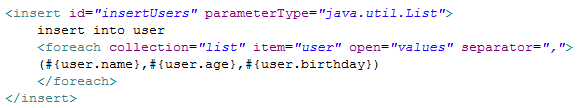


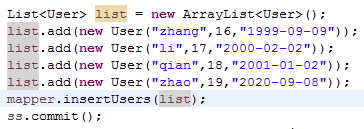
**注意**在这里，接口方法中采用了注解起名的方式，如果没有注解起名，必须在映射文件中必须使用list引用。如同简单类型必须使用value一样。



## 示例3：利用foreach完成同时插入多条记录

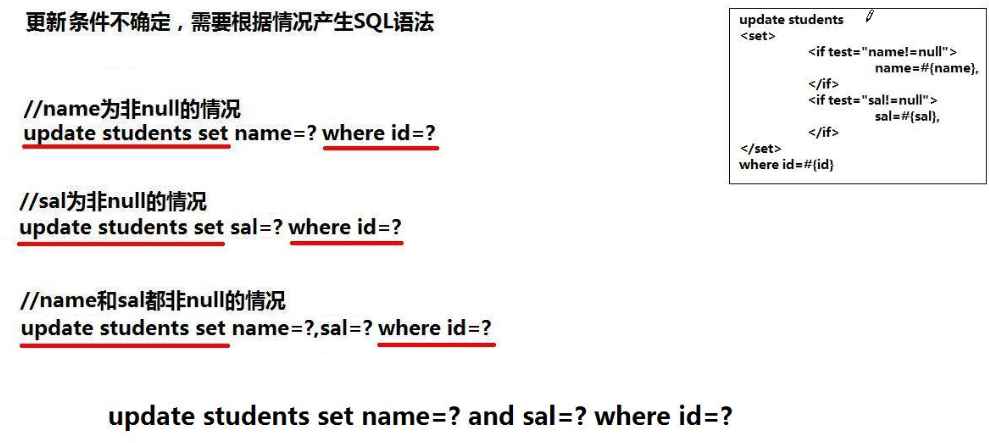






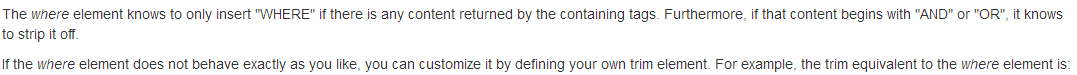
# 更新条件不确定时：set语句与if语句联合使用

**set语句会自动把最后一个逗号删去。**

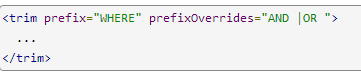


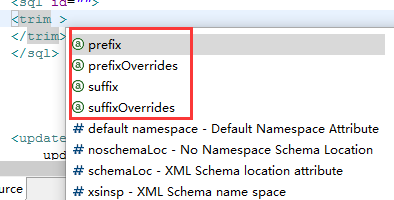
# trim标签：可以去除多余的字符

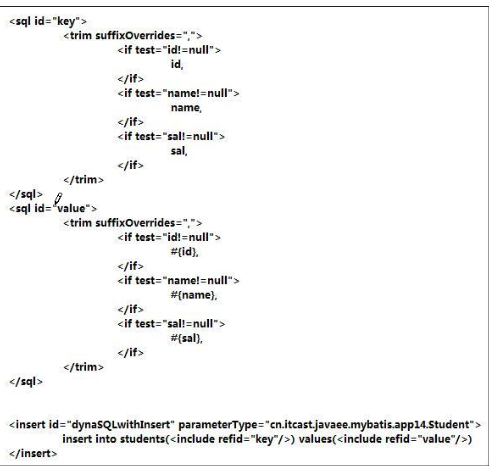
对于where、set等标签，可以自动去除多余的符号。但是对于sql片段不可以去除多余的，这时候，可以通过**trim标签**去除。



**where标签**之所有能够自动去除多的And或or，是因为内部使用了trim标签所具有的功能。

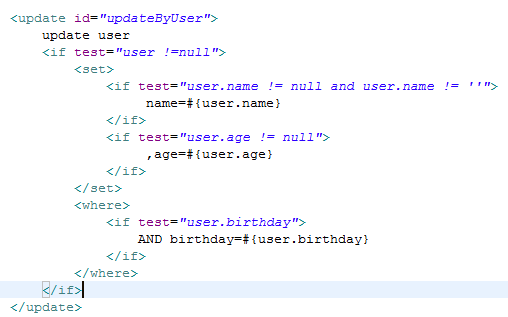






# update语句：使用set、if、where标签

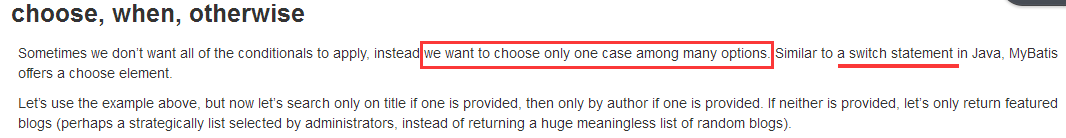


、



# choose、when、otherwise=》switch语句

从多个条件中选择一个。



choose就相当于switch

when就相当于case

otherwise 就相当于default

# 其他标签