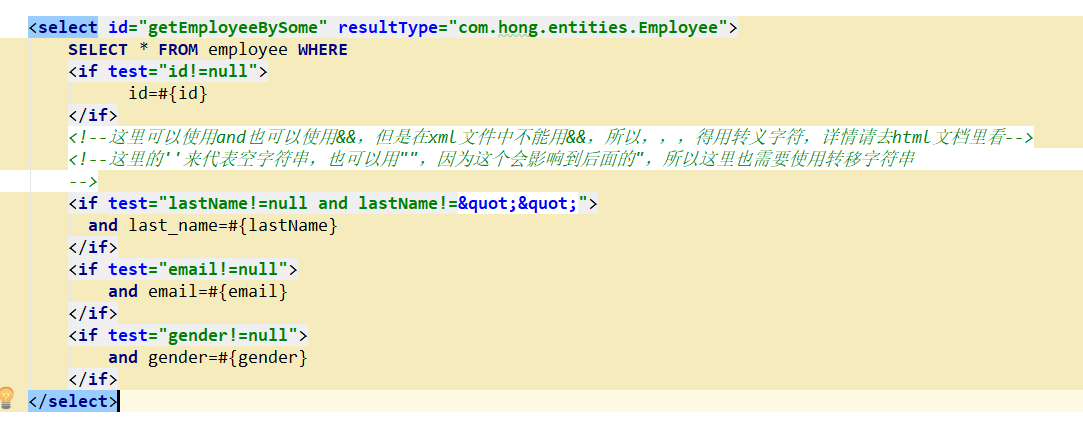
Mybaits的动态sql（各种标签的使用以及OGNL），类似于jstl和el

OGNL文档

<https://commons.apache.org/proper/commons-ognl/language-guide.html>

1. if标签、where标签、trim标签

需求：想要进行查询的时候将传入的属性不为空的作为查询条件



首先注意OGNL里面的字符串的转义的问题

<http://www.w3school.com.cn/tags/html_ref_entities.html>

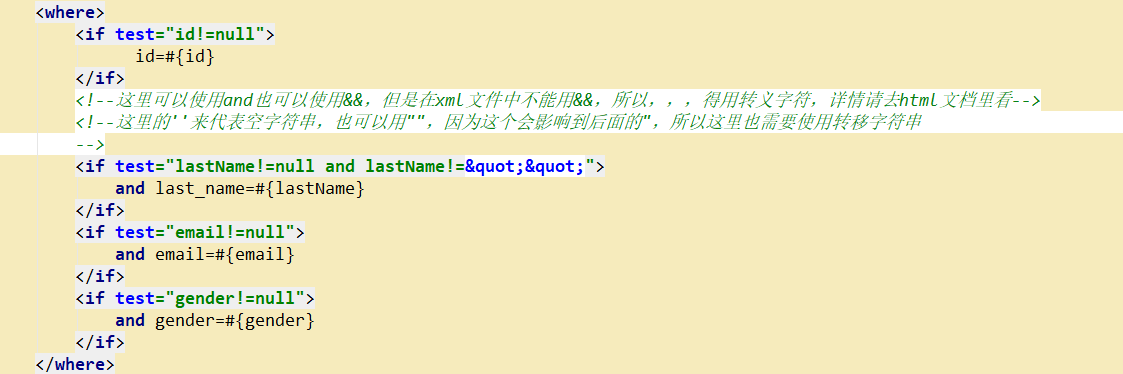
然后注意一下，看一下，那个and拼接，如果有没有id，拼接的sql上就会多一个and，所以说，需要解决这个问题

解决方案1.

在头上加上一个1=1，下面都用and来连接上

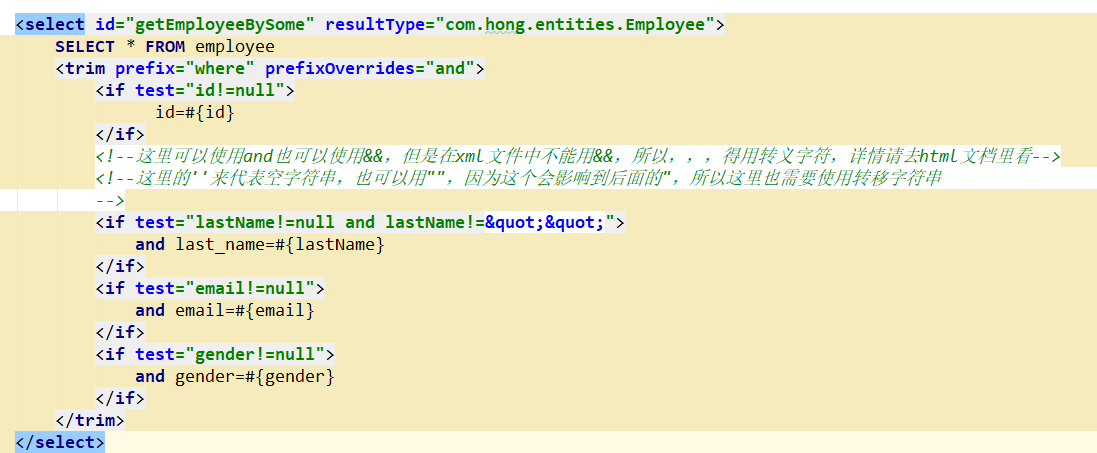
解决方案2

使用where标签，这个标签会帮你去掉头上的and/or，但是如果你习惯写在每个的后面的话，这种解决方式又不能用了



解决方案3（虽说比较好用，但好像用的比较少）

使用trim标签，这里面有prefix属性（用来指定下面这一串前面的前缀，比如“where”，prefixOverrides这个指定前缀覆盖什么，，比如前面有个and他就给你去掉，类似的，还有suffix和suffixOverrides）

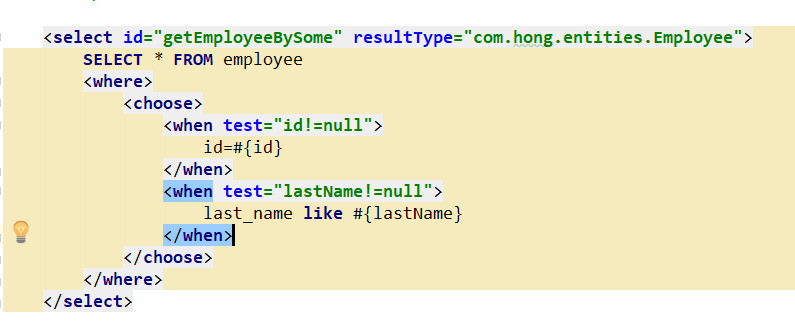


1. set标签

需求：哪个值不为空，就更新哪个值，这时，如果不使用set标签会出现跟上面一样的问题，多出一个，逗号，而使用set标签的话就会自动帮你去掉那个逗号，当然，这里你也可以像前面那样，使用trim标签，但是还是推荐使用set标签

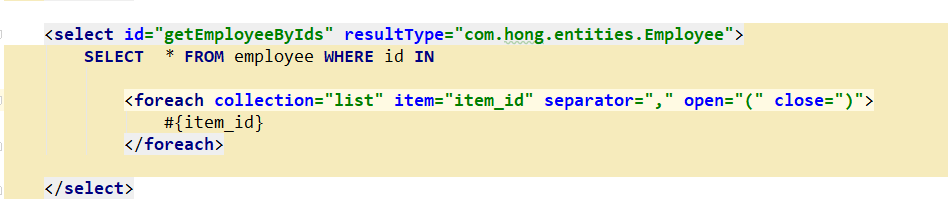
1. choose标签、when标签、ohterwise标签（类似于switch和case和break和default的组合的效果）

需求：如果传入了id，就是用id进行查询，若没有传入id，则使用lastName进行查询，如果都没有就把所有都查出来（注意：此处下图并没有使用list作为返回值，所以没有做这个的实现）

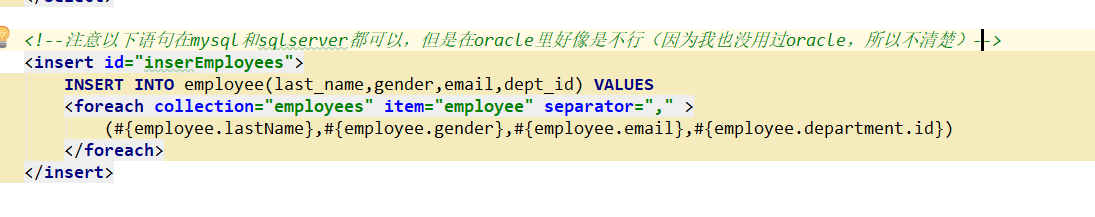


1. foreach标签

需求：给定一窝id，查出这一窝id的对象（即select \* where id in{ 1,2,3,4 }）



就这样，collection指定一个集合（如果想用ids这个集合名字，请在接口上加上@param（ids）），item是指，遍历出来的东西放在哪个里面，separator指定分隔符，open是指定开头是什么，close是指定结束是什么



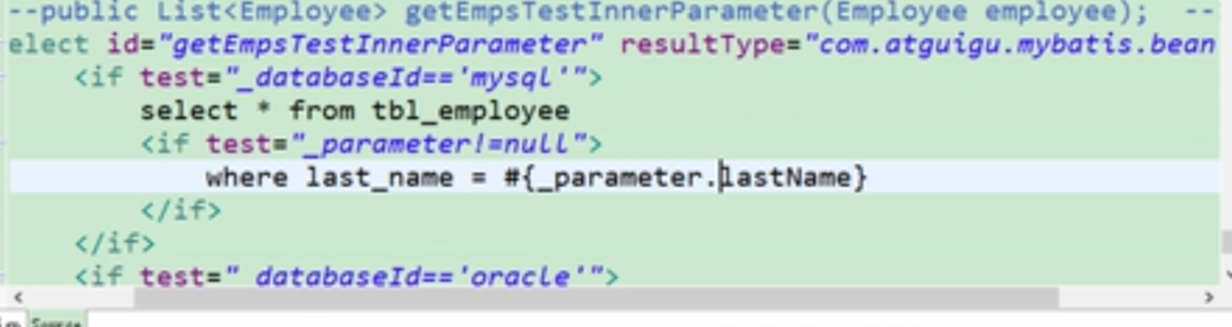
这个是批量插入，嗯，，批量插入，没毛病。

1. mybatis内置参数
   1. . \_databaseId

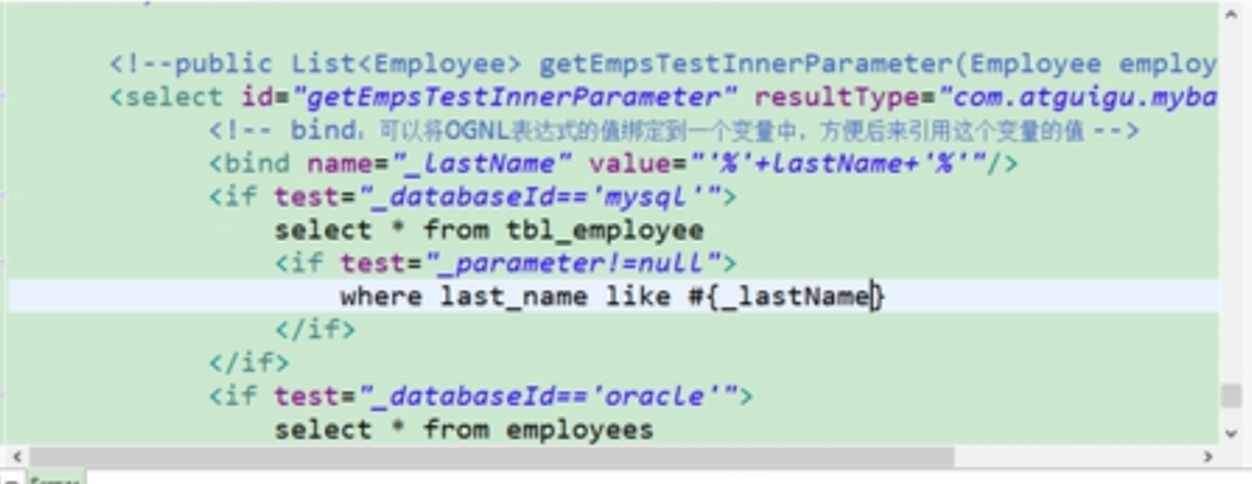
这个内置参数可以select’、insert等标签上使用，可以用if标签用该属性写判断是否是mysql数据

* 1. \_parameter

这个内置参数是指整个对象（例如employee）



1. 在select下面还有一个<bind>标签，这个标签，可以指定绑定的参数，然后在下面的语句中使用



1. 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

${prop}



1. 两级缓存

一级缓存（本地缓存），sqlSession级别的缓存，一级缓存是一直开启的，就是sqlSession级别的一个Map

与数据库同一次会话期间查询到的数据会放到本地缓存中，

以后如果需要获取相同的数量，直接从缓存中拿，没必要去查询数据库

一级缓存失效情况（没有使用到当前一级缓存的情况，效果就是还乣再向数据库发出查询）

1. sqlSession不同（废话）
2. sqlSession’相同，查询条件不同（当前一级缓存中还没有这个数据）
3. sqlSession相同，两次查询之间执行了增删改操作（这次增删改可能对当前数据有影响）
4. sqlSession相同，手动清楚了一级缓存（缓存清空）

二级缓存

1. 一个会话，查询一条数据，这个数据就会被放在当前会话的一级缓存中
2. 如果会话关闭，一级保存中的数据会被保存到二级缓存中，新的会话查询信息，就可以参照二级缓存
3. sqlSession=EmployeeMapper==》Employee
   * + 1. DepartmentMapper==》Department

不同namespace查出的数据会放在自己对应的缓存中（map中）

效果：数据会从二级缓存中获取

查出数据会被默认放置在一级缓存中

只有会话提交或者关闭之后，一级缓存中的数据才会被转义到二级缓存中

使用

1. 开启全局二级缓存配置<setting name=”cacheEnable” value=”true”>
2. 去mapper.xml中配置使用二级缓存

<cache></cache>

1. 我们的POJO需要实现序列化接口

和缓存有关的设置/属性

1. cacheEnabled=true/false，关闭缓存（二级缓存关闭）（一级缓存一直可以被使用）
2. 每个select标签都有useCache=”true”

False，不使用缓存（一级缓存依然使用，二级缓存不使用）

1. 每个增删改标签的，flushCache=”true”，（一级二级都会清除）

增删改执行完成后就会清除缓存

测试，flushCache=“true”，一级缓存就清空了，二级也会被清除

查询标签，flushCache=“false”，

如果flushCache=true；每次查询之后都会清空缓存

1. sqlSession.clearCache（）；只是清除单签session的一级缓存，
2. localCacheScope，本地缓存作用域，（一级缓存SESSION）；

STATEMENT，可以禁用一级缓存

