

```

<select id="dynamicChooseTest" parameterType="Blog" resultType="Blog">
    select * from t_blog where 1 = 1
    <choose>
        <when test="title != null">
            and title = #{title}
        </when>
        <when test="content != null">
            and content = #{content}
        </when>
        <otherwise>
            and owner = "owner1"
        </otherwise>
    </choose>
</select>

```

http://www.yiibai.com/mybatis/mybatis_choose.html

choose (when, otherwise)标签

有时候我们并不想应用所有的条件，而只是想从多个选项中选择一个。而使用if标签时，只要test中的表达式为 true，就会执行 if 标签中的条件。MyBatis 提供了 choose 元素。if标签是与(and)的关系，而 choose 是或(or)的关系。

choose标签是按顺序判断其内部when标签中的test条件出否成立，如果有一个成立，则 choose 结束。当 choose 中所有 when 的条件都不满足时，则执行 otherwise 中的sql。类似于 Java 的 switch 语句，choose 为 switch，when 为 case，otherwise 则为 default。

例如下面例子，同样把所有可以限制的条件都写上，方便使用。choose会从上到下选择一个when标签的test为true的sql执行。安全考虑，我们使用where将choose包起来，放置关键字多于错误。

```

<!-- choose(判断参数) - 按顺序将实体类 User 第一个不为空的属性作为: where条件 -->
<select id="getUserList_choose" resultMap="resultMap_user" parameterType="com.yiibai.pojo.User">

```

```

    SELECT *
    FROM User u
    <where>
        <choose>
            <when test="username !=null ">
                u.username LIKE CONCAT(CONCAT(' ', #
{username, jdbcType=VARCHAR}),'%')
            </when >
            <when test="sex != null and sex != '' ">
                AND u.sex = #{sex, jdbcType=INTEGER}
            </when >
            <when test="birthday != null ">
                AND u.birthday = #{birthday, jdbcType=DATE}
            </when >
            <otherwise>
            </otherwise>
        </choose>

```

```
        </where>
    </select>
```

choose (when,otherwise) ,相当于java 语言中的 switch ,与 jstl 中的 choose 很类似。

```
<select id="dynamicChooseTest" parameterType="Blog" resultType="Blog">
    select * from t_blog where 1 = 1
    <choose>
        <when test="title != null">
            and title = #{title}
        </when>
        <when test="content != null">
            and content = #{content}
        </when>
        <otherwise>
            and owner = "owner1"
        </otherwise>
    </choose>
</select>
```

when元素表示当 when 中的条件满足的时候就输出其中的内容，跟 JAVA 中的 switch 效果差不多的是按照条件的顺序，当 when 中有条件满足的时候，就会跳出 choose，即所有的 when 和 otherwise 条件中，只有一个会输出，当所有的条件都不满足的时候就输出 otherwise 中的内容。所以上述语句的意思非常简单，当 title!=null 的时候就输出 and title = #{title}，不再往下判断条件，当title为空且 content!=null 的时候就输出 and content = #{content}，当所有条件都不满足的时候就输出 otherwise 中的内容。