

普通工程需要手动导包，不然程序出不来。

一，SQL标签和include标签的组合使用

- (1)，要尽量避免使用“\$”符号进行取值。
- (2)，resultType中的数据类型返回的是数据库中一条记录的数据类型。
- (3)，使用map的话，当没有数据的时候，返回的数据是null，这样很容易报空指针异常；使用list的话，如果没有数据，返回的是一个空集合，并不会报空指针异常。
- (4)，在java中，只要是一样的东西的话，我们都可以将其抽取出来。

1，SQL标签中并不是存放SQL语句的，SQL标签是用来存储多条SQL语句中相同的部分。

在mybatis中，SQL和include中的标签需要配合使用，单独使用没有任何意义。

如果在实际操作中，查询的操作很多，而且操作的字段也非常多的话，可以考虑将这些公共的字段存储在SQL标签中，需要使用的时候使用include标签将其引用过来就可以了。

include标签是放在select标签下的。在局部配置文件中书写的代码如下：

```
<!-- sql标签可以存放多条SQL语句中的相同部分 -->
<sql id="common">
    id,username,password
</sql>
<!-- 其中保证SQL中的id和include中的refid中的值是一样的 -->
<select id="insertDemo" parameterType="map">
    insert into users(<include refid="common"></include>) values({id},{#
{username},{#password}});
</select>
```

2，具体的代码以及解释：

```
<!-- sql标签可以存放多条SQL语句中的相同部分 -->
<sql id="common">
    id,username,password
</sql>
<!-- 其中保证SQL中的id和include中的refid中的值是一样的 -->
<select id="insertDemo" parameterType="map">
```

```
insert into users(<include refid="common"></include>) values({id},{username},{password});
</select>
```

注意：

(1) 其中include标签是存储在select标签下面的。

(2) id的值和refid的值一样。

二，动态查询SQL (***)

1，创建表computer：

```
create table computer(id bigint primary key,computerName varchar (40),brand
varchar(40),runMem float,price float);
```

2，向表中插入数据：

```
insert into computer values(1,'deapad310s','ideapad',8,5500), (2,'联想 (Lenovo)
拯救者R720 15.6英寸游戏','联想',8,6500), (3,'Apple MacBook Air 13.3英寸笔记本电脑','苹果',4,7000), (4,'联想 (Lenovo) 小新潮7000 14英寸轻薄窄','联想',4,8000);
```

3，按照不同的条件查询数据：

(1) 全部查询：

```
select *from computer;
```

(2) 查询价格<=7000的电脑：

```
select *from computer where price <=7000;
```

(3) 查询价格<=7000并且牌子是“联想”的电脑：

```
select *from computer where price <=7000 and brand ='联想';
```

(4) 这样会存在很多的SQL语句，一个一个写是不实际的，如果使用SQL拼接的话基本也是不可实现的。这个时候就需要我们使用动态SQL进行动态查询。

(5) 在mysql中查询数据的时候，如果一个数据不是字符串类型的话，依然可以使用一对单引号将数据包裹起来，这样查询的结果依然是正确可靠的。

(6) 在整个mybatis中，“<”或者“>”不能直接写，有时候会报错。

“<”代表<(小于);

“>”代表>(大于);

(7) 局部配置文件中的配置信息：

```
<!-- 动态查询 "<"代表<(小于); ">"代表>(大于); -->
<select id="selectComputer" parameterType="map" resultType="map">
    select *from computer where 1 = 1
    <if test="computerName!=null">
```

```

        and computerName = #{computerName}
    </if>
    <if test="brand!=null">
        and brand = #{brand}
    </if>
    <if test="runMem!=null">
        <!-- 在test中，不需要使用#{ }就可以获取到参数的值 -->
        and runMem = #{runMem}
    </if>
    <if test="price!=null">
        and price <= #{price}
    </if>
</select>

```

(8) 测试的demo代码:

```

static SqlSession session = null;
static {
    SqlSessionFactoryBuilder sqlSessionFactoryBuilder = new
    SqlSessionFactoryBuilder();
    try {
        InputStream ins = Resources.getResourceAsStream("mybatis.xml");
        SqlSessionFactory sessionFactory =
sqlSessionFactoryBuilder.build(ins);
        session = sessionFactory.openSession();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

@Test
/*
 * 实现动态查询
 */
public void selectComputer() {
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("computerName", "联想 ( Lenovo ) 拯救者R720 15.6英寸游戏");
    map.put("brand", "联想");
    map.put("runMem", 8);
    map.put("price", 6500);
    List<Map<String, Object>> maps =
session.selectList("cn.dao.userDaoImpl.userDao.selectComputer", map);
    for (Map<String, Object> map2 : maps) {
        System.out.println(map2);
    }
}

```

(9) 查询的条件不确定的话，SQL语句就意味着也就不确定。动态SQL全靠where标签和if标签共同实现。另外使用where标签嵌套if的配置文件信息如下：

```
<!-- 动态查询 "<"代表<(小于); ">"代表>(大于); -->
<select id="selectComputer" parameterType="map" resultType="map">
    select *from computer
    <where>
        <!-- where标签用于嵌套if标签，第一个if的前面不需要and，
        如果第一个条件不要，where标签会自动去掉第二条记录前面的and -->
        <if test="computerName!=null">
            computerName = #{computerName}
        </if>
        <if test="brand!=null">
            and brand = #{brand}
        </if>
        <if test="runMem!=null">
            <!-- 在test中，不需要使用#{ }就可以获取到参数的值 -->
            and runMem = #{runMem}
        </if>
        <if test="price!=null">
            and price <=#{price}
        </if>
    </where>
</select>
```