补充前面对结果集处理：

前面学习过使用JavaBean对象来存储查询结果集，这是针对查询一张表的情况，在resultType中填写JavaBean的完整类名。
--这种方式还有一个要求就是：
查询结果的列名要和JavaBean对象的属性名一致。要保持这种一致性有两种方法：
1）javabean类的字段名与要查询的表中的字段名全部一致
2）在查询语句中将查询结果的列名起别名，该别名和JavaBean的字段名一致，达到传递数据结果到JavaBean对象的要求

--除了以上以外，如果JavaBean类中的字段名与表中的不一致，结果集列名又不起别名的情况下：
使用<resultMap>标签进行"结果映射",--指定结果集列名和java对象属性的对应关系
该标签是与sql语句标签平级的，在<select>等标签之前设置，有自己的唯一标识id，而且设置对应关系时列名如果是主键的值与其他普通列名的指定是由区别的。案例如下：

我的实体类如下User.java，一定要确保实体类中有无参数的构造方法，因为mybatis的动态代理是用实体类的无参构造来自动创建实体类对象的

```
package com.mybatis.domain;

public class User {

    private int id;
    private String account;
    private String password;
    private String name;

    public User() {
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getAccount() {
        return account;
    }

    public void setAccount(String account) {
        this.account = account;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
```

```
        this.password = password;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "User{" +
                "id=" + id +
                ", account='" + account + '\'' +
                ", password='" + password + '\'' +
                ", name='" + name + '\'' +
                '}';
    }
}
```

sql映射文件如下

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="test">
    <!--
        注意：parameterType属性是给sql语句中占位符传值的，定义占位符传什么类型对象的属性的值到该位置上
        翻译为：参数类型，下面insert语句中传递的是com.mybatis.domain.User类的属性字段值
        同时javabean在给占位符传值时，程序员要告诉mybatis应该将javabean的哪个属性字段值传到哪个占位符上
        所以占位符中需要有javabean的属性字段名
        mybatis中占位符不用问号表示，因为要放javabean的字段信息，所以用：#{这里写javabean的属性字段名} 表示
        #{loginName}底层原理是mybatis会将#{}中的属性字段名获取到，将其首字母大写，然后在前面添加get，调用javabean中的各字段
        的get方法来获取到值，从而组成sql语句执行
    -->
    <!--定义resultMap
         id自己定义，代表你定义的这个resultMap
         type：填对应Java类型的全限定名-->
    <resultMap id="user" type="com.mybatis.domain.User">
        <!--对主键结果列映射的设置
            使用id标签
            column：填查询结果的列名
            property：Java类型的属性名
            -->
        <id column="id" property="id"/>
        <!--非主键列
            使用result标签-->
        <result column="loginName" property="account"/>
        <result column="loginPwd" property="password"/>
```

```
        <result column="realName" property="name"/>

    </resultMap>

    <!--使用上面设置的映射关系，直接将某个resultMap的id放进resultType属性中-->
    <select id="getById"  resultType="user">
        select
            id,loginName,loginPwd,realName
        from
            t_user
        where
            id = #{id}
    </select>

    <select id="getAll" resultType="user">
        select
            *
        from
            t_user
    </select>

</mapper>
```

这种方式主要是将来不得不改表中的字段名，由于项目中字段名一般都是非常多的，修改java类麻烦，而sql语句起别名也麻烦，因为sql语句可能有很多，这样我们就可以使用resultMap属性设置映射关系。
--一般情况不用这种