

## mybatis03

笔记本： mybatis

创建时间： 2019/6/28 20:18

更新时间： 2019/6/28 23:02

作者： qmys2018@163.com

1.动态sql配置 执行的sql不是固定 参数是可变的

1)

`: select * from t_department where 1=1 and dname=? and remark=?`

```
public List<Department> find1(Map<String, Object> map);
```

mapper.xml

```
<select id="find1" resultType="department">
  select * from t_department where 1=1
  <!-- d r 就是map中key的值 一定要保持一致 才能获取值 m.put("d", "xxx"); m.put("r", "x");-->
  <if test="d!=null">
    and dname=#{d}
  </if>
  <if test="r!=null">
    and remark=#{r}
  </if>
</select>
```

2)

`select * from t_department WHERE dname=? and remark=?`

```
<select id="find2" resultType="department">
  select * from t_department
  <where>
    <!-- dname 就是map中key的值 -->
    <if test="d!=null">
      and dname=#{d}
    </if>
    <if test="r!=null">
      and remark=#{r}
    </if>
  </where>
</select>
```

3)

G - ==> `Preparing: select * from t_department WHERE dname in (?, ?, ?)`

```
<select id="find3" resultType="department">
  select * from t_department
  <where>
    <if test="m1!=null"><!--m.put("m1", 1);-->
      dname in
      <foreach collection="m1" item="t" open="(" separator="," close=")"> <!--t是自定义-->
        #{t}
      </foreach>
    </if>
  </where>
```

test:

```
/* select * from t_department where dname in ('aaa','xxx','222');
 * 思路： 将参数存入list中 将list存入map中 遍历的是map中的这个list
 */
Map<String, Object> m=new HashMap<String, Object>();
```

```
List<String> l=new ArrayList<String>();
l.add("aaa");
l.add("xxx");
l.add("222");
m.put("m1", l);
department.find3(m);
```

作业：

课程中的案例查询条件是=拼接 思考 条件也是动态的 例如可以是 like =

```
public List<Department> find3(List<Object> list);
```

## 2.调用存储过程

```
public String callPro(Map<String,Object> map);
```

xml:

```
<select id="callPro" statementType="CALLABLE">
  call p(
    #{a,mode=IN,jdbcType=VARCHAR},    a:map的key值
    #{b,mode=OUT,jdbcType=VARCHAR}    b:是获取map的key
  )
</select>
```

```
Map<String, Object> m=new HashMap<String, Object>();
m.put("a", "qqqq");
department.callPro(m);
System.out.println(m.get("b")); //返回结果
```

## 3.高级查询 多对一的关联映射 员工和部门的信息都要查询

### 1) 在model中创建一个User 并且正确的关联两个类的关系

在User中 private Department d;//每一个员工都有一个部门  
在Department中 private List<User> list;//一个部门中有多个员工  
生成get/set

### 2)创建接口 写方法

```
//所有员工的信息和部门信息
public List<User> find();
```

### 3) 创建xml

```
<resultMap type="user" id="u">
  <id column="T_ID" property="t_id" jdbcType="INTEGER"/>
  <result column="LOGINNAME" property="loginname" jdbcType="VARCHAR"/>
  <result column="USERNAME" property="username" jdbcType="VARCHAR"/>
  <result column="PASS" property="pass" jdbcType="VARCHAR"/>
  <result column="TEL" property="tel" jdbcType="INTEGER"/>
  <result column="COMM" property="comm" jdbcType="INTEGER"/>
  <result column="SAL" property="sal" jdbcType="INTEGER"/>
  <result column="STATE" property="state" jdbcType="INTEGER"/>
  <result column="D_ID" property="d_id" jdbcType="INTEGER"/>
  <association property="d" javaType="department">
    <id column="D_ID" property="d_id" jdbcType="INTEGER"/>
    <result column="DNAME" property="dname" jdbcType="VARCHAR"/>
    <result column="REMARK" property="remark" jdbcType="VARCHAR"/>
  </association>
</resultMap>
```

com. nm. model. User 的别名  
结果的自定义的标识  
com. nm. model. Department的别名

类的属性  
和表的字段  
的映射  
column: 字段名  
property 属性名

```

<select id="find" resultMap="u">
    select * from t_user u,t_department d where u.d_id=d.d_id
</select>

```

resultMap的标识u

#### 4) 测试

```

IUserMapper user=session.getMapper(IUserMapper.class);

List<User> list=user.find();
for(User u:list){
    System.out.println("員工名: "+u.getUsername()+"部門名: "+u.getD().getDname());
}

```

sql : select \* from t\_user u,t\_department d where u.d\_id=d.d\_id

#### 4.关联的嵌套查询 (多对一)

##### 1)

```
public List<User> find2();
```

##### 2) xml

```

<!-- 关联的嵌套查询 (多对一) -->
<resultMap type="user" id="u2">
    <id column="T_ID" property="t_id" jdbcType="INTEGER"/>
    <result column="LOGINNAME" property="loginname" jdbcType="VARCHAR"/>
    <result column="USERNAME" property="username" jdbcType="VARCHAR"/>
    <result column="PASS" property="pass" jdbcType="VARCHAR"/>
    <result column="TEL" property="tel" jdbcType="INTEGER"/>
    <result column="COMM" property="comm" jdbcType="INTEGER"/>
    <result column="SAL" property="sal" jdbcType="INTEGER"/>
    <result column="STATE" property="state" jdbcType="INTEGER"/>
    <result column="D_ID" property="d_id" jdbcType="INTEGER"/>
    <association column="D_ID" property="d" select="s"></association>
</resultMap>
<select id="find2" resultMap="u2">
    select * from t_user
</select>
<select id="s" resultType="department">
    select * from t_department where d_id=#{d_id}
</select>

```

外键

user中部门属性名

下面select的id值标识

#### 测试:

```

List<User> list=user.find2();
for(User u:list){
    System.out.println("員工名: "+u.getUsername()+"部門名: "+u.getD().getDname());
}

```

sql:

```
BUG - ==> Preparing: select * from t_user
BUG - ==> Parameters:
BUG - ==> Preparing: select * from t_department where d_id=?
BUG - ==> Parameters: 1(Integer)
BUG - <==== Total: 1
BUG - ==> Preparing: select * from t_department where d_id=?
BUG - ==> Parameters: 3(Integer)
BUG - <==== Total: 1
BUG - ==> Preparing: select * from t_department where d_id=?
BUG - ==> Parameters: 100(Integer)
BUG - <==== Total: 1
BUG - ==> Preparing: select * from t_department where d_id=?
BUG - ==> Parameters: 200(Integer)
BUG - <==== Total: 0
BUG - <== Total: 11
```