

mybatis 【数据权限案例】

作者: [haxLook](#)

原文链接: <https://ld246.com/article/1617159380903>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

数据权案例

mybatis的sql构造，可能是我们程序猿每天都需经历的过程，前几天公司有了一个新的需求，做一个结果的数据权限。

举个栗子：

某张表中维护这大量的客户信息，但是客户信息不是每个用户都看到,系统中维护一个树形结果的组结，根据上下级的关系进行数据查看权限。

A

a1

a2

a1,a2只能看到自己维护的信息。

A可以看到自己以及a1,a2以及自己信息。

其实这个问题从最开始分析的话，其实是在先将所有的数据查询先查询出来在进行处理，但是一进行数据查询的时候，都是分页查询，这样的话一定会然查询出来数据和进行过滤出来的数据有区分可能分页是查询10条，结果在过滤下没有10条数据，而且就在进行其他处理达到效果，但是在效率是一定会有很大的影响的。

出发点只能走--sql构造数据权限

思路设计：我只需要查询出来的表的某个字段，或者关联其他表某个字段匹配到某个用户权限集合就了

```
select *from customer cus  
left join user use on use.id = cus.createuser  
where cus.createuser in (.....所需要的数据权限)
```

● 第一步

如何获取都某张表的所有集合，我们可以使用mysql的函数来进行构造出我们需要的数据，直接上脚本



```
CREATE DEFINER=`root`@`%` FUNCTION `getTableChildList`(rootId varchar(100)) RETURNS va  
char(2000) CHARSET utf8  
BEGIN  
DECLARE str varchar(2000);  
DECLARE cid varchar(2000);  
SET str = "";  
SET cid = rootId;  
SET str = concat(str, ',', cid);  
SELECT group_concat(id) INTO cid FROM team where FIND_IN_SET(parent_id, cid) > 0;  
RETURN str;  
END
```

解释：team是小组的表名 根据id找打它的parent_id的所对应的id，直至于下级全部查询完毕！

1.表名team 团队名称 --包含了一个
leaderid 组长id
parent_id 上级团队

2.表名 member 团队成员
memberid 成员id
salesteamid 团队主键

这个函数可以写的更加适配一些，可以供应多个表结构使用。上代码

```
CREATE DEFINER=`root`@`%` FUNCTION `getTableChildList`(tablename varchar(100),rootId
  varchar(100)) RETURNS varchar(2000) CHARSET utf8
BEGIN
  DECLARE str varchar(2000);
  DECLARE cid varchar(2000);
  SET str = "";
  SET cid = rootId;
  IF tablename = 'A' then
    WHILE cid is not null DO
      SET str = concat(str, ',', cid);
      SELECT group_concat(id) INTO cid FROM A where FIND_IN_SET(parent_id, cid) > 0;
    END WHILE;
  ELSEIF tablename = 'B' THEN
    WHILE cid is not null DO
      SET str = concat(str, ',', cid);
      SELECT group_concat(id) INTO cid FROM B where FIND_IN_SET(parent_id, cid) > 0;
    END WHILE;
  ELSEIF tablename = 'C' THEN
    WHILE cid is not null DO
      SET str = concat(str, ',', cid);
      SELECT group_concat(id) INTO cid FROM C where FIND_IN_SET(parentid, cid) > 0;
    END WHILE;
  END IF;
  RETURN str;
END
```

将函数脚本中，就可以实现我们需要的效果了，这里默认#{parameter.condition.loginUserId}是一定值的，如果是自己卸载mapper.xml文件中话，要加上if 判断操作。

```
SELECT
  *
FROM
  customer cus
  LEFT JOIN USER USE ON USE.id = cus.createuser
WHERE 1=1
<if test="parameter.condition != null and parameter.condition.loginUserId !=null and parame
er.condition.loginUserId !=" ">
  and
  cus.createuser IN (
    SELECT DISTINCT
      memberid
    FROM
      member
    WHERE
      salesteamid IN ( SELECT id FROM team WHERE FIND_IN_SET( id, getTableChildList (
        'team', ( SELECT id FROM team WHERE leaderid = #{parameter.condition.loginUserId} )
      )
  )
```

```
    ))  
  )  
</if>
```

注意这样的写法还会存一个问题，如果我查询的一个人是组长，而且刚好我的下级没有了组员，为了统的效率问题，可以加上一个 if 的语句进行判断

语法规则：IF (boolean condition,"true","false") ;

改进版效率提升，而且扩展性更好。

```
SELECT  
  *  
FROM  
  customer cus  
  LEFT JOIN USER USE ON USE.id = cus.createuser  
WHERE 1=1  
  <if test="parameter.condition != null and parameter.condition.loginUserId !=null and param  
ter.condition.loginUserId !=" ">  
  
AND IF  
  (  
  (  
  SELECT  
  count( id )  
  FROM  
  team  
  WHERE  
  FIND_IN_SET( id, getTableChildList ( 'team', ( SELECT id FROM team WHERE leaderid =  
#{parameter.condition.loginUserId} ) ) )  
  ) > 0,  
  cus.createuser IN (  
  SELECT DISTINCT  
  memberid  
  FROM  
  member  
  WHERE  
  salesteamid IN ( SELECT id FROM team WHERE FIND_IN_SET( id, getTableChildList ( 'team', ( SELECT id FROM team WHERE leaderid =  
#{parameter.condition.loginUserId} ) ) )  
  )  
  ))  
  )  
,  
  cus.createuser IN ( SELECT DISTINCT memberid FROM member WHERE memberid =  
#{parameter.condition.loginUserId} )  
  )  
</if>
```

扩展

- IFNULL ("t1","t2") 【mysql】
- ISNULL ("t1","t2") 【SQL Server / MS Access】
- NVL ("t1","t2") 【Oracle】

第一参数如果为null 使用第二个参数

- IF (boolean condition,"true","false") ;

第一个参数布尔判断和为true , 使用第二个参数, false使用第三个 参数; 和三目运算的规则一样。