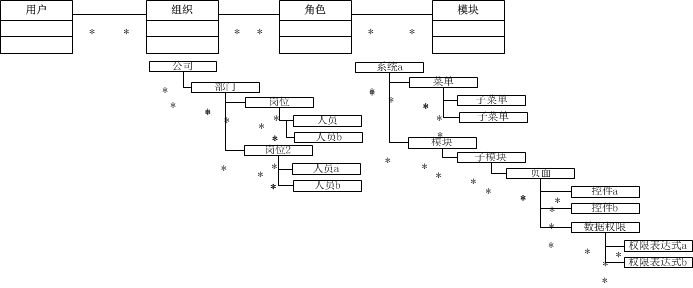
# [基于组织角色的权限设计](https://www.cnblogs.com/zq8024/p/5003050.html)

**一、 目标：**

1. 可以对功能进行权限控制，控制粒度可以达到页面级别和控件级别
2. 可以对数据进行权限控制
3. 可以按角色、按人、按部门、按岗位等授权，
4. 可以分级授权，方便管理

**二、 设计思路**



1. 模块

模块是一个抽象的概念，大到可以表示一个系统，小到可以表示一个控件的功能，包括了菜单、模块、页面、功能、数据权限及数据权限表达式等。

1)   菜单：是一个指向某个功能的指针，指向一个页面或者功能，所以可以将其视为一个功能。

2)   模块：页面功能的分类，模块下面拥在子模块和页面。

3)   页面：呈现给最终用户的界面，用户通过页面来和系统交互，根据系统的不同，BS架构时是web页面， CS架构时是windows界面。

4)   控件功能：每个界面上的控件，例如保存、删除等按钮所代表的功能，某个控件的显示功能等

5)   数据权限： 标识一个需要进行权限控制的数据集合，下面包含多个权限表达式，每一个表达式就是一个SQL语句的where条件的一部分。例如查看全部的表达式是 1= 1， 查看本部门的数据的表达式是 DepartmentId = ‘$User.DepartId$’、查看我的数据的表达式是UserId = ‘$User.UserId$’。

6)   权限表达式：用来对数据进行过滤，本质上是一个SQL语句的where条件的一部分。每个表达式都拥有优先级，当同时拥有多个表达式时，取优先级高的表达式。

2. 角色

角色是一组功能的集合，一个角色包含了多个功能权限，方便管理员进行授权，例如系统管理员拥有后台管理的功能，后台管理的功能包括相关菜单的权限、相关界面及控件的操作权限。

3. 组织

组织分为公司、部门、岗位和人员4级，其中公司和部门可以自循环，公司下面可以有子公司， 部门下面可以有子部门。

授权时，将角色分配给组织上的节点，可以将角色分配给公司、部门、岗位和人员，下级节点自动继承上级组织的权限， 给某个部门分配角色后， 该部门下面的子部门和相关人员就都具有了该角色所拥有的权限。当某个人归属到组织上后，就自动拥有组织的权限。在分配权限时，采用权限递增的原则进行分配。

4. 分级授权

分级授权时，只能分配那些分配给组织的权限，这个和组织所拥有的权限是不同的， 例如某个组织节点拥有10项权限， 但是可以分配的权限可能只是这10个中的一部分。

5. 相关问题

1)   用户和组织中的人员是什么关系？

人员是用户与组织的关系的表示， 如果一个人只有一个岗位，那么在组织中就会有一个人员。如果用户在组织中有多个岗位，即一人多岗的情况，那么在组织中就会有多个人员。

2)   授权时，为什么不直接把模块分配给组织，而要先把模块分配给角色，再把角色分配给组织？

因为模块中的功能粒度过细，用户的管理员分不清楚每一个模块具体有什么功能，但用户知道角色代表的意义。例如用户的管理员知道会计角色，但是不知道会计具体使用哪些功能。

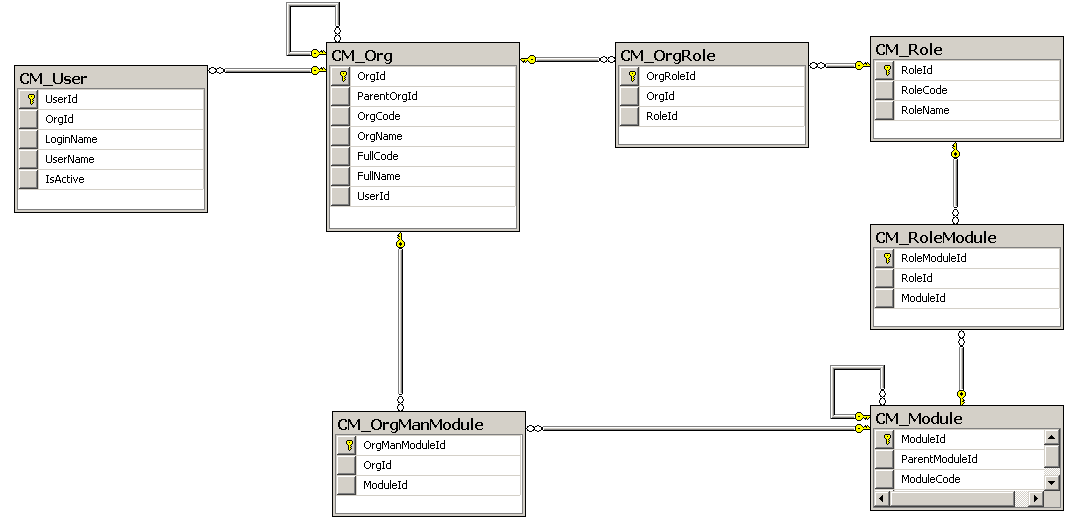
这样角色和组织中的岗位会有一定的重叠，但是两者又有一些区别。组织中的岗位是用户在实际生活中的角色， 而系统中的角色是为了方便权限管理而设置的，即权限角色。实际生活中的角色可能拥有多个权限角色。例如总经理可能拥有全部的权限角色，而在组织中总经理是一个岗位。

**三、库表设计**

　　相关表以CM\_开头（ Common的简写）

|  |  |  |
| --- | --- | --- |
| 序号 | 表名 | 说明 |
| 1 | CM\_User | 用户 |
| 2 | CM\_Org | 组织 |
| 3 | CM\_OrgRole | 组织角色 |
| 4 | CM\_Role | 角色 |
| 5 | CM\_RoleModule | 角色功能 |
| 6 | CM\_Module | 功能 |
| 7 | CM\_OrgManModule | 组织管理的功能（分级授权用） |

库表关系图:



**四、后续思考**

　　1、系统应该基于角色的权限控制。

　　2、角色有层级关系，其作用主要用于角色管理，而不是权限继承。

　　3、组织代表了角色，当组织存在的时候，可以让程序自动为每个组织中的节点建立一个角色，以避免组织与角色的重叠。

　　4、人员与组织是多对多关系，应该要建立一个组织人员表，而不是将人员也归到组织表中去。

　　5、为了方便授权，应该有以下几个功能：按功能授权，按角色授权（包括按人授权）

　　6、有组织的按组织授权，没有组织的按角色授权