# RBAC模型的概念

## 定义

RBAC（Role-Based Access Control）即：基于角色的权限控制。通过角色关联用户，角色关联权限的方式间接赋予用户权限。



## 优点

1. 只是直接给用户分配权限，少了一层关系，扩展性弱了许多，适合那些用户数量、角色类型少的平台。
2. 有了角色后，我们只需要为该角色制定好权限后，将相同权限的用户都指定为同一个角色即可，便于权限管理。
3. 对于批量的用户权限调整，只需调整用户关联的角色权限，无需对每一个用户都进行权限调整，既大幅提升权限调整的效率，又降低了漏调权限的概率。

# RBAC 模型的分类

RBAC 模型可以分为：RBAC0、RBAC1、RBAC2、RBAC3 四种。其中 RBAC0 是基础，也是最简单的，相当于底层逻辑，RBAC1、RBAC2、RBAC3 都是以 RBAC0 为基础的升级。

## RBAC0 模型-最简单的用户角色权限模型

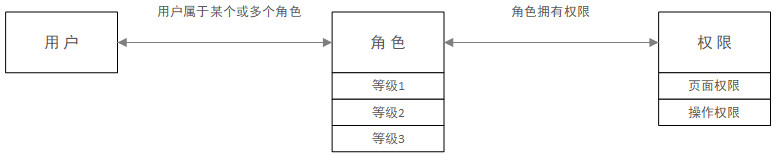
1. 用户和角色是多对一关系，一个用户只充当一种角色，一种角色可以有多个用户担当。

如果系统功能比较单一，使用人员较少，岗位权限相对清晰且确保不会出现兼岗的情况，此时可以考虑用多对一的权限体系。

1. 用户和角色是多对多关系，一个用户可同时充当多种角色，一种角色可以有多个用户担当。

大多数情况尽量使用多对多的权限体系，保证系统的可扩展性，一个用户可以拥有多个角色的权限。

## RBAC1 模型



相对于 RBAC0 模型，增加了子角色，引入继承概念，即子角色可以继承父角色的所有权限。

1. 应用场景

如某个业务部门，有经理、主管、专员。主管的权限不能大于经理，专员的权限不能大于主管，如果采用 RBAC0 模型做权限系统，极可能出现分配权限失误，最终出现主管拥有经理都没有的权限的情况。

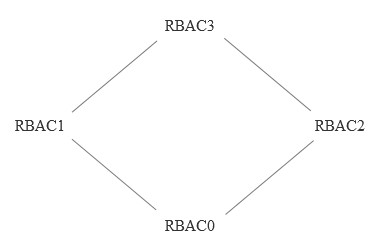
RBAC1 模型创建完经理角色并配置好权限后，主管角色的权限继承经理角色的权限，并且支持在经理权限上删减主管权限。

## RBAC2 模型

基于 RBAC0 模型，增加了对角色的一些限制：角色互斥、基数约束、先决条件角色等。

1. **角色互斥**：同一用户不能分配到一组互斥角色集合中的多个角色，互斥角色是指权限互相制约的两个角色。案例：财务系统中一个用户不能同时被指派给会计角色和审计员角色。
2. **基数约束**：一个角色被分配的用户数量受限，它指的是有多少用户能拥有这个角色。例如：一个角色专门为公司 CEO 创建的，那这个角色的数量是有限的。
3. **先决条件角色**：指要想获得较高的权限，要首先拥有低一级的权限。例如：先有副总经理权限，才能有总经理权限。
4. **运行时互斥**：例如，允许一个用户具有两个角色的成员资格，但在运行中不可同时激活这两个角色。

## RBAC3 模型



称为统一模型，它包含了 RBAC1 和 RBAC2，利用传递性，也把 RBAC0 包括在内，综合了 RBAC0、RBAC1 和 RBAC2 的所有特点。

# 权限的概念

权限是资源的集合，这里的资源指的是软件中所有的内容，包括模块、菜单、页面、字段、操作功能（增删改查）等等。可以将权限分为：页面权限、操作权限和数据权限。

## 页面权限

所有系统都是由一个个的页面组成，页面再组成模块，用户是否能看到这个页面的菜单、是否能进入这个页面就称为页面权限。



1. 商品中心分为基础信息管理，产品库管理，商品库管理。基础信息管理又分为图片库管理，品牌管理及后台分类管理。管理员可以看到所有的页面，而业务员只能看到基础信息管理。这就是页面权限。

## 操作权限

用户凡是在操作系统中的任何动作、交互都是操作权限，如增删改查等。

1. 操作权限是具体到接口的权限。例如某一个用户不允许查询，即使有查询按钮，在后端会检查该查询接口，如果没有会提示无查询权限。
2. 操作权限也可以通过前端控制，例如不允许用户编辑或者上传，直接在前端隐藏该按钮，也就是接口的入口。

## 数据权限

一般业务管理系统，都有数据私密性的要求：哪些人可以看到哪些数据，不可以看到哪些数据。

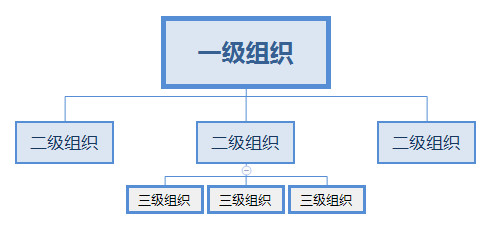
1. 例如部门A的人只能看到部门A的数据，部门B的人只能看到部门B的数据。如果是总经理，可以看到所有部门的数据。

### 实现方式

1. 可以利用 RBAC1 模型，通过角色分级来实现。

针对不同的角色，来限制数据的范围。例如数据对角色A开放，对角色B开放一部分,同时角色A是角色B的上级，这样子就能把角色B看到的数据控制在角色A中。

1. 在‘用户 - 角色 - 权限’的基础上，增加用户与组织的关联关系，用组织决定用户的数据权限。



数据可视权限规则制定：上级组织只能看到下级组织员工负责的数据，而不能看到其他平级组织及其下级组织的员工数据等。

# 用户组的使用

## 应用背景

1. 当平台用户基数增大，角色类型增多时，如果直接给用户配角色，管理员的工作量就会很大。引入一个概念 “用户组”，就是将相同属性的用户归类到一起。
2. 例如：加入用户组的概念后，可以将部门看做一个用户组，再给这个部门直接赋予角色（1 万员工部门可能就几十个），使部门拥有部门权限，这样这个部门的所有用户都有了部门权限，而不需要为每一个用户再单独指定角色，极大的减少了分配权限的工作量。
3. 同时，也可以为特定的用户指定角色，这样用户除了拥有所属用户组的所有权限外，还拥有自身特定的权限。

## 优点

1. 用户组的优点，除了减少工作量，还有更便于理解、增加多级管理关系等。
2. 如：我们在进行组织机构配置的时候，除了加入部门，还可以加入科室、岗位等层级，来为用户组内部成员的权限进行等级上的区分。

# 权限系统的实例设计

