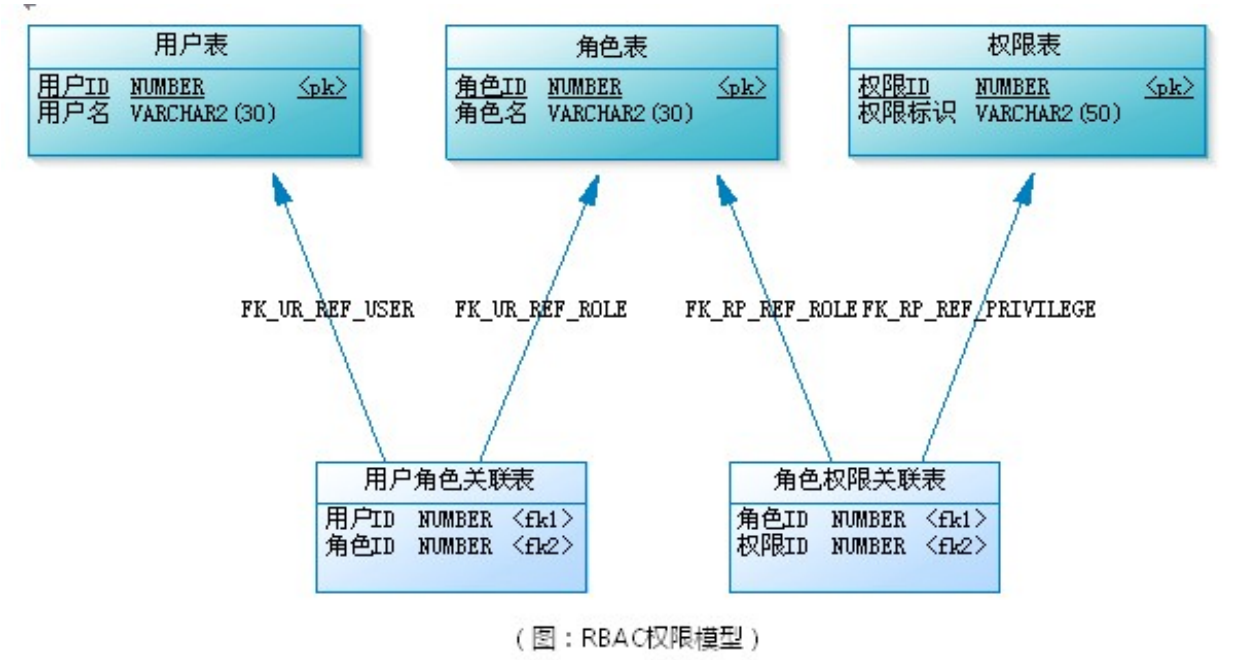
权限控制模块

基于 **RBAC（Role-based Access Control）权限访问控制**。也就是说一个用户可以有多个角色，一个角色可以有多个权限，通过将角色和权限分离开来提高设计的可扩展性，通常一个用户有多个角色，一个角色也会属于多个用户（多对多），一个角色有多个权限，一个权限也会属于多个角色（多对多）。

一、最简单版本
假设：我们拿到一个用户对象，
可以通过：用户 id –>角色 id–>角色名称（什么角色）–>权限 id –> 权限标识 –>获取权限。最终获取权限获取角色信息。

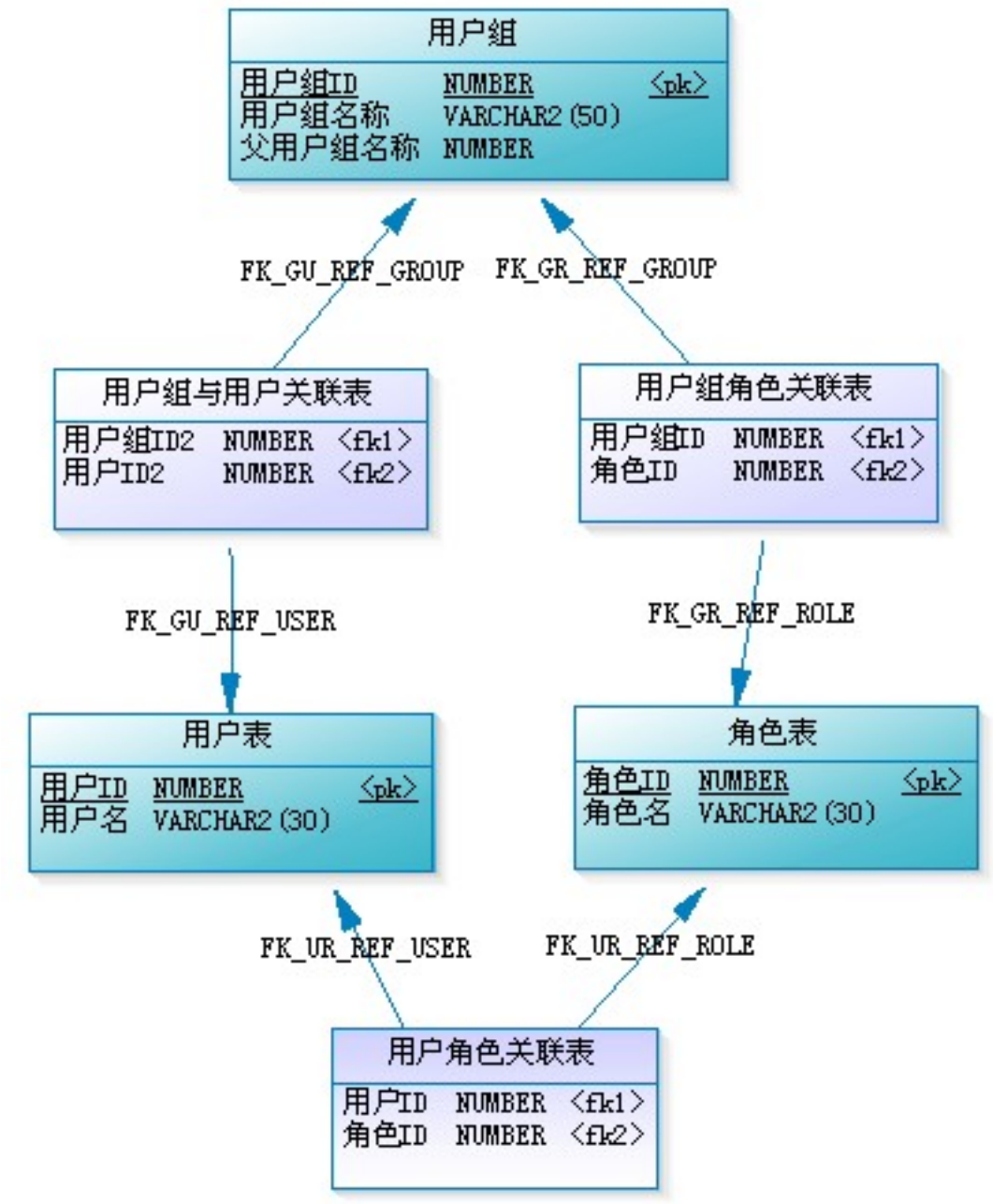


（图：RBAC权限模型）

角色是什么？可以理解为一定数量的权限的集合，权限的载体。例如：一个论坛系统，“超级管理员”、“版主”都是角色。**版主可管理版内的帖子、可管理版内的用户等，这些是权限。**要给某个用户授予这些权限，不需要直接将权限授予用户，可将“版主”这个角色赋予该用户。

二、用户组模式
如果用户数量比较庞大，可以加入用户组模式。**需要给用户分组，每个用户组内有多个用户，可以给用户授权外，也可以给用户组授权。**最终用户拥有的所有权限 = 用户个人拥有的权限+该用户所在用户组拥有的权限。（这个设计类似 svn 中的用户权限，比如，将

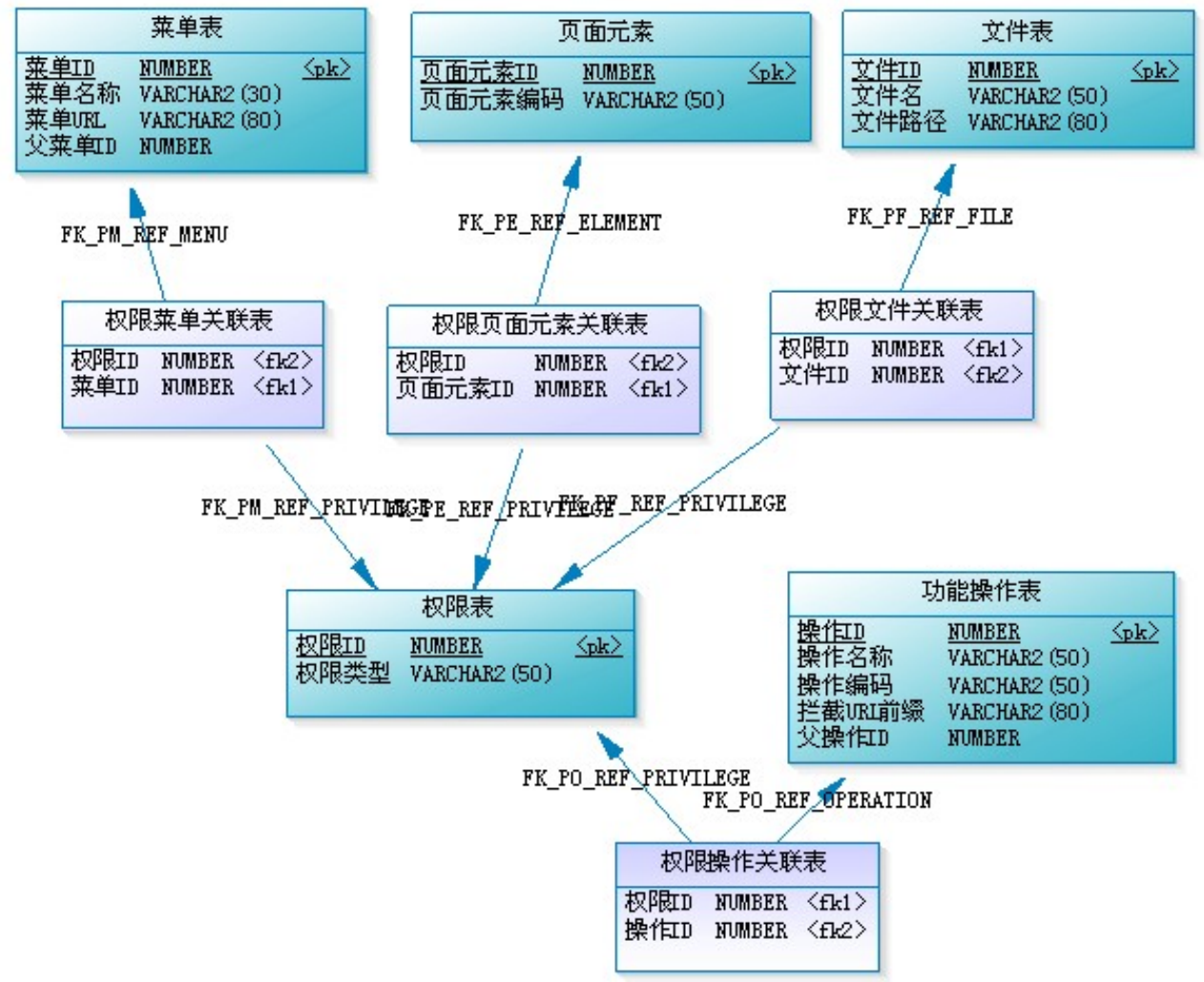
一个 svn 用户加入到 group 中，然后设置 group 的权限，以后加入更多的用户，就不用再一一设置用户的权限了。）



（图：引入用户组）

三、权限分类

大部分是针对功能模块，比如对信息记录的增删改（信息状态修改，文件的删除修改等），菜单的访问，输入框，按钮的可见性，是否可以新增下级管理员等。有些权限设计，会把功能操作作为一类，而把文件、菜单、页面元素等作为另一类，这样构成“用户-角色-权限-资源”的授权模型。而在做数据表建模时，可把功能操作和资源统一管理，也就是都直接与权限表进行关联，这样可能更具便捷性和易扩展性。



（图：权限分类）

四、完整版

请留意权限表中有一列“权限类型”，我们根据它的取值来区分是哪一类权限，如“MENU”表示菜单的访问权限、“OPERATION”表示功能模块的操作权限、“FILE”表示文件的修改权限、“ELEMENT”表示页面元素的可见性控制等。

优点：（1）不需要区分哪些是权限操作，哪些是资源。
（2）方便扩展，当系统要对新的东西进行权限控制时，我只需要建立一个新的关联表“权限 XX 关联表”，并确定这类权限的权限类型字符串。

这里要注意的是，权限表与权限菜单关联表、权限菜单关联表与菜单表都是一对一的关系。（文件、页面权限点、功能操作等同理）。也就是每添加一个菜单，就得同时往这三个表中各插入一条记录。这样，可以不需要权限菜单关联表，让权限表与菜单表直接关联，此时，须在权限表中新增一列用来保存菜单的 ID，权限表通过“权限类型”和这个 ID 来区分是种类型下的哪条记录。

（图：RBAC权限模型扩展）