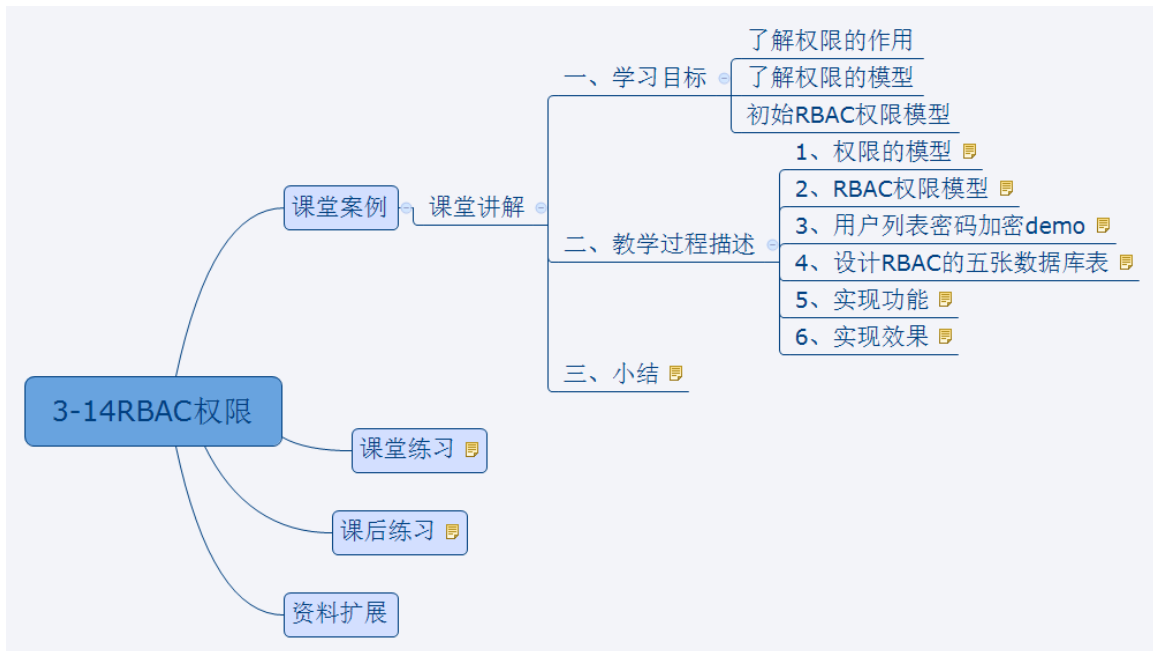


## 3-14RBAC权限

3-14RBAC权限 .....	1
1. 课堂案例 .....	2
课堂讲解 .....	2
一、学习目标 .....	2
了解权限的作用 .....	2
了解权限的模型 .....	2
初始RBAC权限模型.....	2
二、教学过程描述 .....	2
1、权限的模型 .....	2
2、RBAC权限模型.....	3
3、用户列表密码加密demo .....	4
4、设计RBAC的五张数据库表.....	5
5、实现功能 .....	10
6、实现效果 .....	12
三、小结 .....	12
2. 课堂练习 .....	12
3. 课后练习 .....	13
4. 资料扩展 .....	13



## 1. 课堂案例

### 课堂讲解

#### 一、学习目标

了解权限的作用

了解权限的模型

初始RBAC权限模型

#### 二、教学过程描述

##### 1、权限的模型

# 权限的模型

ACL	访问控制列表
RBAC	基于角色的权限控制
ABAC	基于属性的权限控制
PBAC	基于策略的权限控制

RBAC:Role Based Access Control, 翻译过来基本上就是基于角色的访问控制系统。

ACL: Access

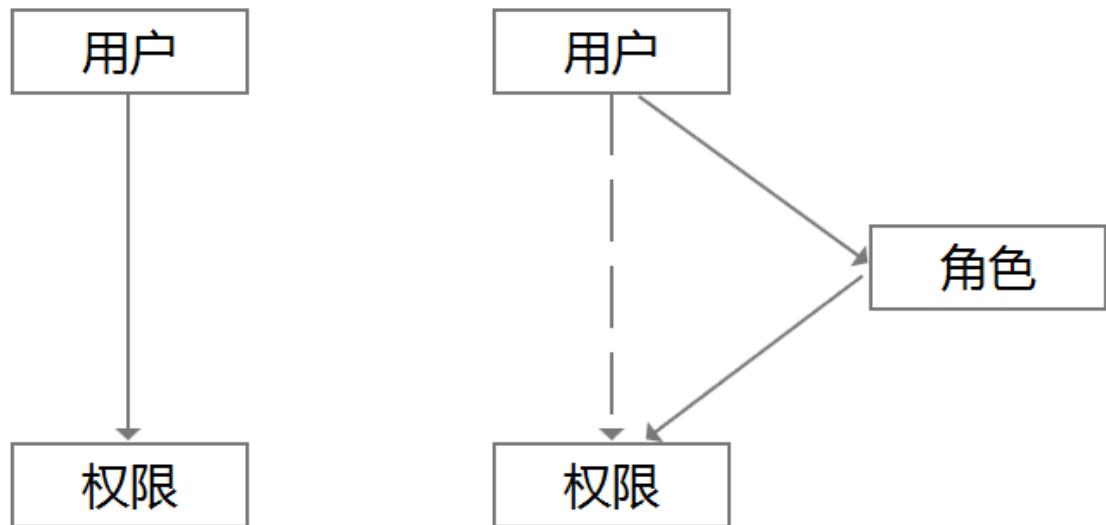
Control

List, 访问控制列表, 是前几年盛行的一种权限设计, 它的核心在于用户直接和权限挂钩。

而ABAC和PBAC模式是不适合互联网行业, 所以不多介绍。

## 2、RBAC权限模型

# ACL和RBAC对比



RBAC的核心是用户只和角色关联，而角色代表了对权限，这样设计的优势在于使得对用户而言，只需角色即可以，而某角色可以拥有各种各样的权限并可继承。ACL和RBAC相比缺点在于由于用户和权限直接挂钩，导致在授予时的复杂性，虽然可以利用组来简化这个复杂性，但仍然会导致系统不好理解，而且在取出判断用户是否有该权限时比较的困难，一定程度上影响了效率。还是要根据系统的需求来设计，如果系统的角色比较固定并且调整不是很大RBAC就是一个很好的选择，如果对于系统中的用户都必须单独来分配权限，并且每个用户的权限基本都不一样，那这种系统的设计通过ACL来实现更为适合。

## 3、用户列表密码加密demo

RBAC中关于用户表，因为用户表用于用户的登陆，所以呢密码一般需要进行加密处理，为了安全起见，我们加密一般都不用md5这么简单加密，除非用md5各种混合操作。  
那么下面我们提供了一个密码的加密和解密的算法，大家可以借鉴下，如下：

```
<?php
// 先md5加密后，再加盐crypt
// 加密过程
$a = '123456'; //注册时输入的
$pass = crypt(md5($a), 'abcdefg123456');

// 解密过程
$b = '123456'; //登陆时输入的
if(crypt(md5($b), $pass) == $pass) {
    echo '校验成功';
    echo '这里处理登陆后的业务逻辑'
}
```

#### 4、设计RBAC的五张数据库表

### 管理员用户表

rl\_manager @goshopping (localhost) - 表

文件	编辑	窗口	帮助				
新建	保存	另存为	添加栏位	插入栏位	删除栏位	主键	上移
栏位	索引	外键	触发器	选项	注释	SQL 预览	
名	类型	长度	小数点	允许空值 (			
id	int	20	0	<input type="checkbox"/>	1		
username	varchar	32	0	<input type="checkbox"/>			
pass	varchar	100	0	<input type="checkbox"/>			
sex	tinyint	1	0	<input checked="" type="checkbox"/>			
phone	varchar	16	0	<input checked="" type="checkbox"/>			
email	varchar	50	0	<input checked="" type="checkbox"/>			
status	tinyint	1	0	<input checked="" type="checkbox"/>			
remark	text	0	0	<input checked="" type="checkbox"/>			
addtime	int	11	0	<input checked="" type="checkbox"/>			
updatetime	int	11	0	<input checked="" type="checkbox"/>			

默认:

注释:

☒ 自动递增

☒ 无符号

☐ 填充零

## 管理员用户-角色关系表

rl\_manager\_role @goshopping (localhost) - 表

文件 编辑 窗口 帮助						
新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移						
栏位 索引 外键 触发器 选项 注释 SQL 预览						
名	类型	长度	小数点	允许空值 (		
id	int	11	0	<input type="checkbox"/>	1	
manager_id	int	11	0	<input type="checkbox"/>		
role_id	varchar	100	0	<input type="checkbox"/>		
ctime	int	11	0	<input checked="" type="checkbox"/>		

默认:

注释:  ...

☒ 自动递增

☒ 无符号

☐ 填充零

角色表

rl\_role @goshopping (localhost) - 表

文件 编辑 窗口 帮助						
新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上一步						
栏位 索引 外键 触发器 选项 注释 SQL 预览						
名	类型	长度	小数点	允许空值 (		
id	int	5	0	<input type="checkbox"/>	1	
name	varchar	50	0	<input type="checkbox"/>		
remark	varchar	250	0	<input checked="" type="checkbox"/>		
status	tinyint	1	0	<input type="checkbox"/>		
ctime	int	11	0	<input type="checkbox"/>		
mtime	int	11	0	<input checked="" type="checkbox"/>		

默认:

注释:

☒ 自动递增

☒ 无符号

☐ 填充零

## 角色-权限关系表



rl\_role\_auth @goshopping (localhost) - 表

文件 编辑 窗口 帮助

新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移

栏位 索引 外键 触发器 选项 注释 SQL 预览

名	类型	长度	小数点	允许空值 (	
▶ id	int	11	0	<input type="checkbox"/>	1
role_id	int	11	0	<input type="checkbox"/>	
auth_id	varchar	50	0	<input type="checkbox"/>	
ctime	int	11	0	<input checked="" type="checkbox"/>	

默认:

注释:

☒ 自动递增

☒ 无符号

☐ 填充零

## 权限表

rl\_auth @goshopping (localhost) - 表

文件 编辑 窗口 帮助

新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移

栏位 索引 外键 触发器 选项 注释 SQL 预览

名	类型	长度	小数点	允许空值 (	
auth_id	int	11	0	<input type="checkbox"/>	1
auth_name	varchar	20	0	<input type="checkbox"/>	
auth_pid	int	11	0	<input type="checkbox"/>	
auth_c	varchar	32	0	<input checked="" type="checkbox"/>	
auth_a	varchar	32	0	<input checked="" type="checkbox"/>	
auth_pic	varchar	100	0	<input checked="" type="checkbox"/>	

---

默认:

注释:

☒ 自动递增

☒ 无符号

☐ 填充零

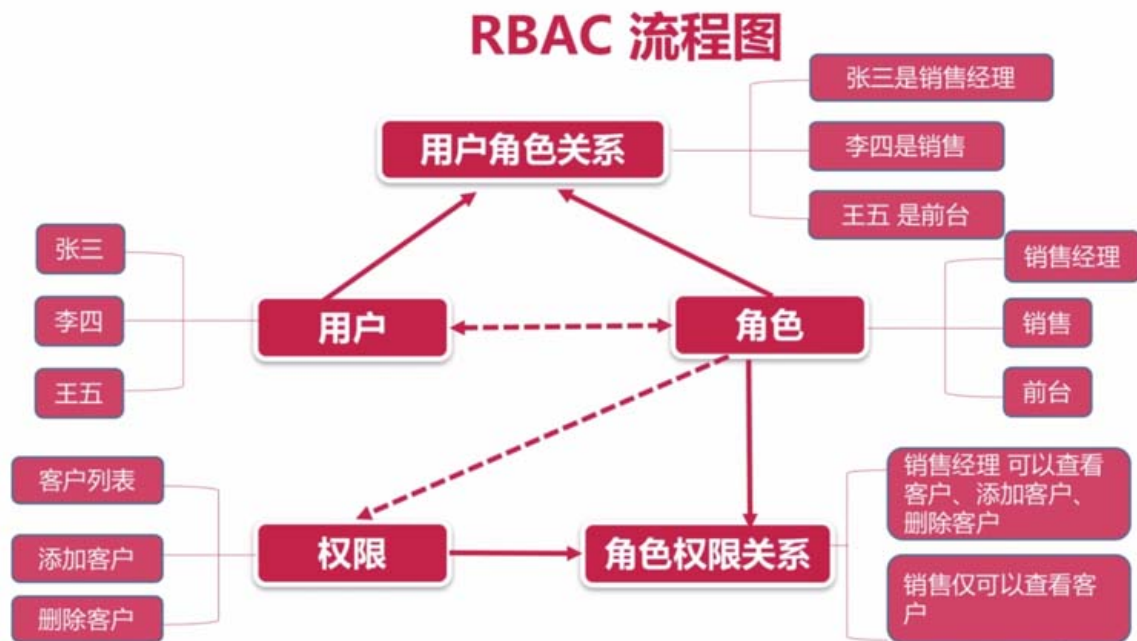
## 5、实现功能

RBAC权限实现的中心思想:核心思想:用户<---->角色<---->权限

意思就是:

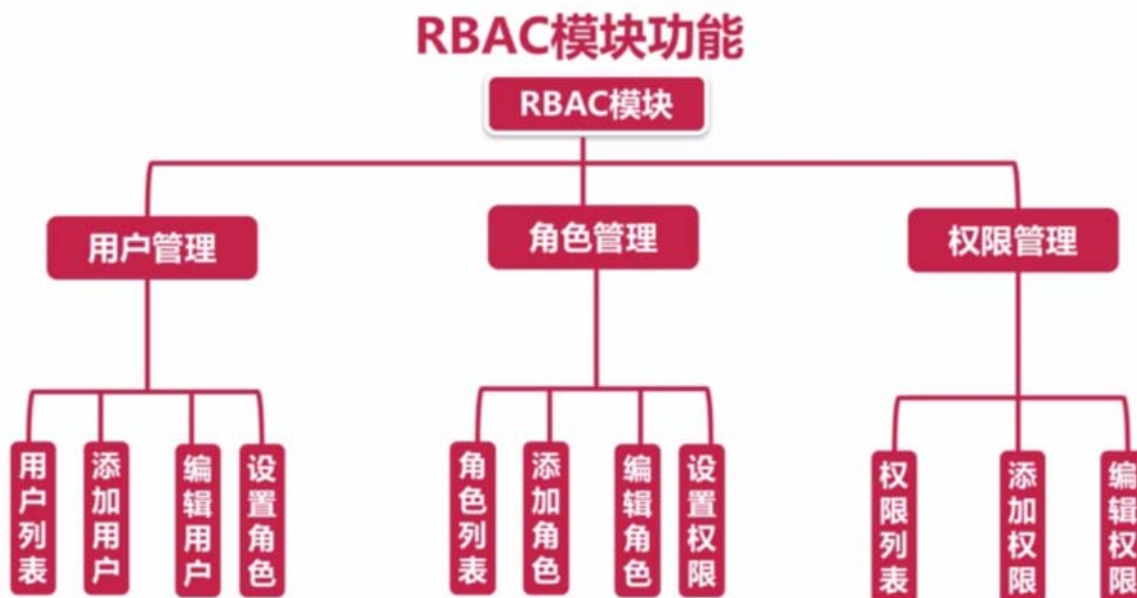
用户想操作某个页面或者功能, 那么必须有某个方面的权限, 那么就要拥有该权限的角色才能间接拥有那个操作权限;

某用户是某角色, 才间接有某权限, 然后才拥有操作相关的功能;



根据流程图的关系：

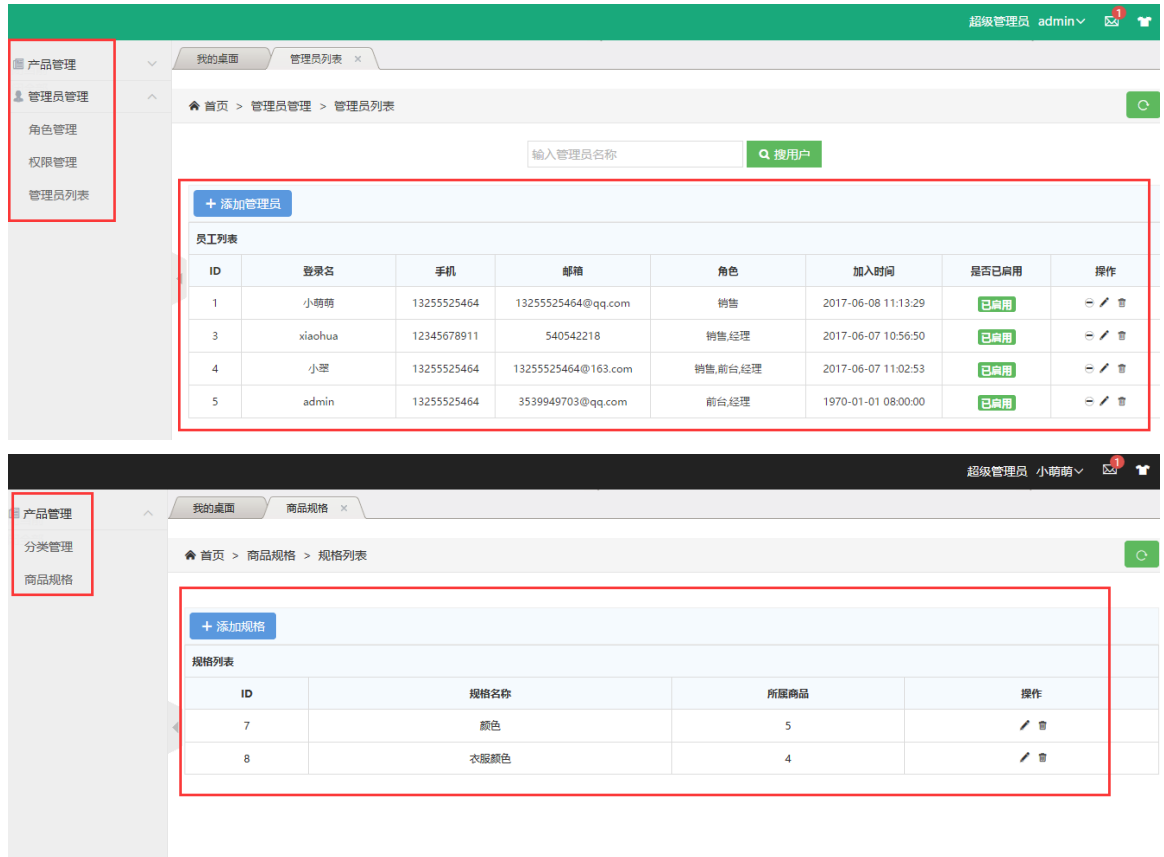
- 1、第一步可以先做权限的增删改查，
- 2、第二步可以做角色的增删改查，顺便把某角色拥有某权限勾选上，在保存数据的时候把权限和角色的id存入角色权限关系表，此时就把权限角色关联起来了。
- 3、第三步做用户的增删改查，这个时候顺便该用户拥有哪些角色勾选上，在保存的时候把用户和角色的id存入用户角色关系表，此时就把用户和角色关联起来了。



如上图所示建立三个模块就是三个控制器，交叉设置他们的功能，和数据表也是一一对应的。

## 6、实现效果

实现不同的管理员用户登陆展示不同的菜单，没有该访问权限的用法无权访问该页面(见备课文档里Common控制器)



## 三、小结

- 1、通过此结的学习学会制作RBAC基于角色的权限管理功能模块
- 2、学会制作一个父类控制器来控制访问权限，其他页面来继承

## 2. 课堂练习

角色管理的增删改查

权限管理的增删改查

管理列表的增删改查

### 3. 课后练习

三表建立关系

管理员用户<---->角色

角色<----->权限

三个页面实现互通

### 4. 资料扩展