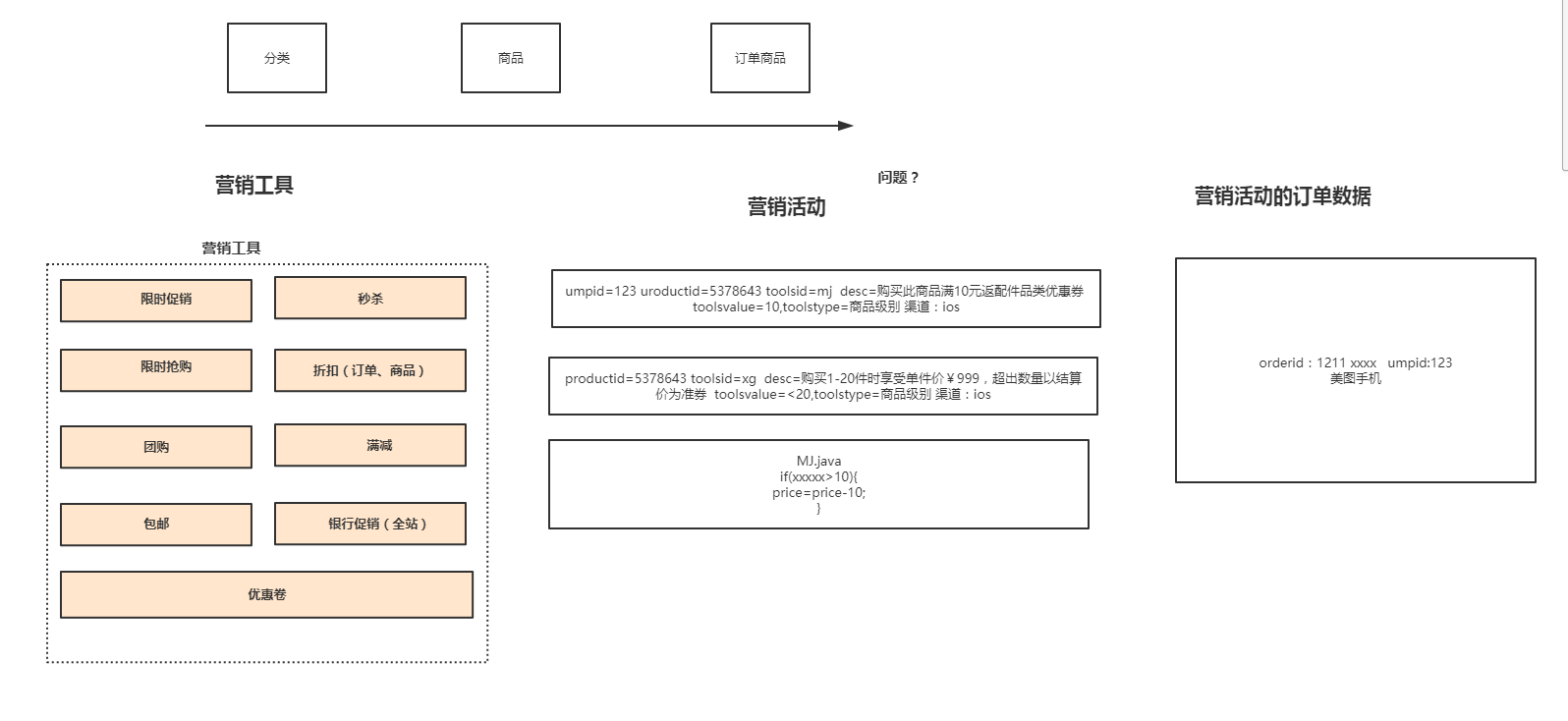
# 营销系统：



**技术如何去实现？**

一个商品会对应多个活动

购物车N个商品N个活动

100个商品 300个活动（300条记录 数据库查询）

100个悟空人 3w条记录

缓存？Redis mysql

Jvm map 数据同步的 在分布式下 不同的jvm 缓存数据不一样？

# 活动优先级问题：怎么解决？权重

jvm 也是内存 redis也是内存 那就不要redis了 将来发生 job 执行的

计划 时间

# 支付宝对接：

<https://openhome.alipay.com/platform/demoManage.htm#/alipay.trade.pay>

# 微信对接：

# 技术注意点：

接口的幂等性





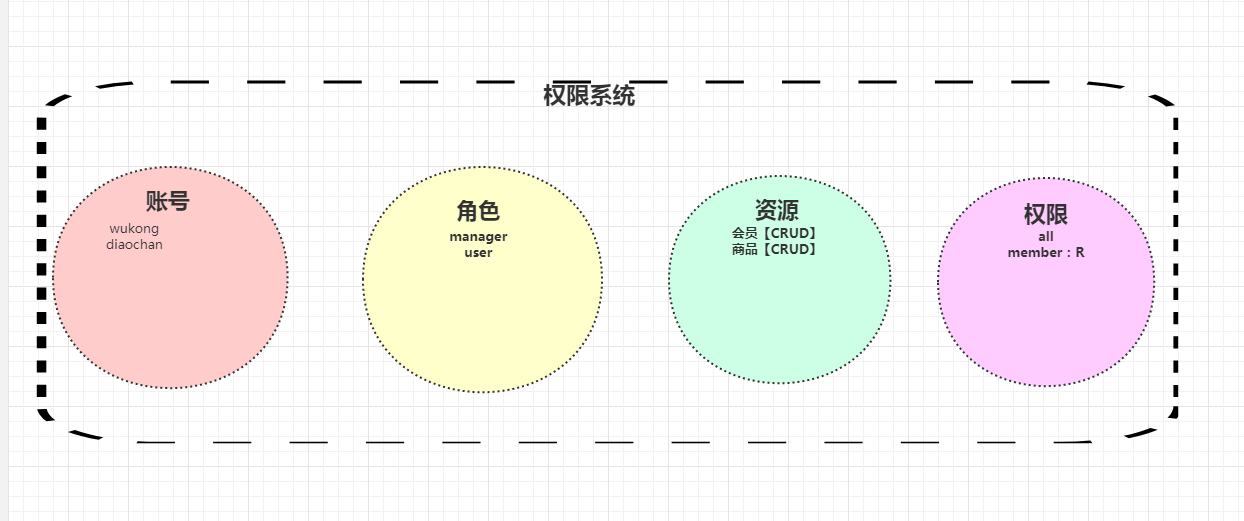
## 后台系统

后台管理：

电商没有关系、很多系统的权限这块 公共。

权限系统一般有：账号、角色、权限、资源。

权限可以分为三种：****页面权限，操作权限，数据权限。****



**页面权限：**

控制你可以看到哪个页面，看不到哪个页面。

很多系统都只做到了控制页面这一层级，它实现起来比较简单，一些系统会这样设计，但是比较古板，控制的权限不精细，难以在页面上对权限进行更下一层级的划分。

If else

**操作权限：**

则控制你可以在页面上操作哪些按妞。

延伸：当我们进入一个页面，我们的目的无非是在这个页面上进行增删改查，那在页面上对应的操作可能是：查询，删除，编辑，新增四个按钮。

可能你在某个页面上，只能查询数据，而不能修改数据。

****数据权限：****

数据权限则是控制你可以看到哪些数据，比如市场A部的人只能看到或者修改A部创建的数据，他看不到或者不能修改B部的数据。

延伸：数据的控制我们一般通过部门去实现，每条记录都有一个创建人，而每一个创建人都属于某个部门，因此部门分的越细，数据的控制层级也就越精细，这里是否有其他好的方式除了部门这个维度还有其他什么方式可以控制数据权限。

#### t\_user用户表

|  |  |  |  |
| --- | --- | --- | --- |
| t\_user用户表 | | | |
| 参数 | 名称 | 类型 | 备注 |
| id | 自增长 | int | 唯一 |
| username | 帐号 | String | 唯一 |
| password | 密码 | string | MD5加密 |
| createtime | 创建时间 | String |  |
| createAccount | 创建人 | String |  |
| updatetime | 最后修改时间 | String |  |
| updateAccount | 最后修改人 | String |  |
| status | 状态 | String | y:启用,n:禁用；默认y |
| rid | 角色ID | Int |  |
| nickname | 昵称 | String |  |
| email | 邮箱 | String |  |
|  |  |  |  |

#### t\_role角色表

|  |  |  |  |
| --- | --- | --- | --- |
| t\_role角色表 | | | |
| 参数 | 名称 | 类型 | 备注 |
| id | 自增长 | int | 唯一 |
| role\_name | 角色名称 | String |  |
| role\_desc | 角色描述 | string |  |
| role\_dbPrivilege | 数据库权限 | String |  |
| status | 角色状态，如果角色被禁用，则该角色下的所有的账号都不能使用，除非角色被解禁。 | String | y:启用，n:禁用；默认y |

#### t\_privilege权限表

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 名称 | 类型 | 备注 |
| id | 自增长 | int | 唯一 |
| rid | 角色ID | int |  |
| mid | 资源ID | int |  |

#### t\_menu资源表

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 名称 | 类型 | 备注 |
| id | 自增长 | int | 唯一 |
| pid | 父ID | Int |  |
| url | 资源 | String |  |
| name | 资源名称 | string |  |
| orderNum | 序号 | int |  |
| type | 功能类型 | String | module：模块  page：页面  button：功能 |

#### t\_systemsetting系统设置表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许为空 | 描述 |
| 基本设置 | | | |
| id | int |  | ID号 |
| systemCode | String |  | 系统代号 |
| name | String |  | 网站名称 |
| www | String |  | 门户地址根http路径 |
| manageHttp | String |  | 后台地址根http路径 |
| log | String |  | 网站门户的Log图标地址 |
| title | String |  | 网站标题 |
| description | String |  | 网站的描述 |
| keywords | String |  | 网站的关键字 |
| shortcuticon | String |  | 网站的图标 |
| address | String |  | 联系地址 |
| tel | String |  | 联系电话 |
| email | String |  | 邮箱 |
| qqHelpHtml | String |  | Qq沟通组建的HTML内容 |
| icp | String |  | 备案号 |
| isopen | String |  | 是否开放网站。y:开放,n不开放 |
| closeMsg | String |  | 网站关闭消息 |
| qq | String |  | Qq号码 |
| statisticsCode | String |  | 站长统计代码 |
| version | String |  | 系统版本相关信息 |
| 显示设置 | | | |
| openResponsive | String |  | y:启用响应式；n:禁用响应式。默认y |
| imageRootPath | String |  | 图片根路径，以后可以专门弄个图片服务器，图片和项目分离，提高站点访问速度。 |
| defaultProductImg | String |  | 商品的默认图片 |
|  |  |  |  |
| images | 图集 | String | 多张图片之间用分号分割。如果广告的useImagesRandom为n，则优先显示html；否则显示images图集的图片，每一次都会从图集中随机选取一张图片来显示。 |
| manageLeftTreeLeafIcon |  | String | 后台左侧菜单叶子节点的图标 |

#### t\_systemlog系统日志表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许为空 | 描述 |
| id | int |  | 自增 |
| title | String |  | 日志标题 |
| content | String |  | 日志内容 |
| type | Int |  | 日志类型。1：登陆日志，2：版本日志， |
| ~~createdate~~ | ~~date~~ |  | ~~记录时间，默认是当前时间~~ |
| account | String |  | 操作人 |
| loginIP | String |  | 登陆人IP地址 |
| logintime | String |  | 登陆时间 |
| loginArea | String |  | 登陆区域 |
| diffAreaLogin | String |  | 是否是异地登陆y:是;n:否 |