

硕士研究生学位论文

|  |  |
| --- | --- |
| 题目： | 网上商城促销系统 |
|  | 的设计与实现 |

|  |  |
| --- | --- |
| 姓 名： | 龙利剑 |
| 学 号： | 1201220801 |
| 院 系： | 软件与微电子学院 |
| 专 业： | 软件工程 |
| 研究方向： | 嵌入式系统设计 |
| 导师姓名： | 吴中海 教授 |
|  | 张齐勋 讲师 |

二〇一六年一月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则，引起有碍作者著作权之问题，将可能承担法律责任。

摘要

本项目基于某公司电子商务平台，旨在开发出一套高效、稳定、智能的网上商城促销系统，方便支持运营团队发布营销活动（满减、满赠、打折等）。

业务流程为：运营人员通过商品管理系统来发布商品，通过销售管理系统来发布促销活动；顾客在网上商城购买相关商品享受促销折扣。从促销商品的发布到用户购买相关商品享受促销折扣，整个业务过程使用WEB技术来实现（包括促销规则，促销商品成本控制等）。通过分析用户注册数据和行为数据（浏览记录，购买记录），使用机器学习、推荐算法等技术，尽量给顾客推荐合适的促销活动。

本论文从实际需求背景出发，研究了促销系统的功能与目标。介绍了促销系统开发的价值和意义。讲述了采用PHP + MySQL + NGINX这套WEB框架的缘由，并分析相关技术的优缺点。项目使用NGINX作为WEB服务器，处理用户请求；使用PHP脚本语言实现后台促销业务逻辑；使用MySQL作为数据库存储业务数据；使用Hadoop分布式计算平台作为计算工具，统计分析热度数据和运行推荐算法。项目总体上以实现促销系统基本功能为主，实现智能促销推荐为辅的思路进行设计与实现。促销系统基本功能主要包括促销活动成本控制、促销活动发布、促销活动优惠记录。促销系统推荐模块引入商品热度数据，这不仅能够缓解推荐算法冷启动问题还能够自适应用户兴趣爱好动态推荐。相比传统的促销系统，结合热度数据而改进的推荐算法可以提高了促销系统的质量。

关键词：促销系统，电子商务，协同过滤，WEB技术

Design and Implementation of Online Shopping Mall Promotion System

Long LiJian(Software Engineering)

Directed by Wu ZhongHai

**ABSTRACT**

The project is based on a company's e-commerce platform, it is designed to develop a highly efficient, stable, intelligent online store promotional system to facilitate marketing activities to support the operations team released (full cut, full of gifts, discounts, etc.).

Business processes: operations staff to publish a commodity by commodity management system to publish promotional activities through sales management system; customer in the online store to buy related products enjoy promotional discounts. Promotional merchandise purchased from release to user-related goods to enjoy promotional discounts, the entire business process using WEB technology to achieve (including the promotion rules, promotional merchandise cost control, etc.). By analyzing user registration data and behavioral data (browsing history, purchase records), using machine learning, recommendation algorithm technology, as much as possible to customers recommend the appropriate promotional activities.

In this paper, starting from the actual needs of the background to study the function and target marketing system. It describes the value and significance of the promotion system development. About the use PHP + MySQL + NGINX WEB framework of this reason, and to analyze the advantages and disadvantages of the related art. WEB server project using NGINX as handling user requests; using PHP scripting language background Promotions business logic; using MySQL as the database store business data; use Hadoop distributed computing platform as the computing tools, data and run statistical analysis heat recommendation algorithm. The overall project on the promotion system in order to achieve the basic functions based, supplemented by intelligent promotional recommended design and realization of ideas. Promotions basic system functions include promotional activities cost control, promotional activities publishing, promotional Events Offers records. Promotional System recommended introduction of commodity temperature data module, which not only can relieve cold start issue recommendation algorithm also can recommend dynamic adaptive user interests. Compared to traditional marketing system that combines heat data and recommend improved algorithm greatly improves the quality of the promotion system.

KEY WORDS: Promotion system, Machine learning, Collaborative filtering, Web technology

# 

目录

[第一章 绪论 1](#_Toc432705451)

[1.1 研究背景和意义 1](#_Toc432705452)

[1.2 国内外研究现状 2](#_Toc432705453)

[1.3 论文的研究内容 2](#_Toc432705454)

[1.4 论文的组织结构 3](#_Toc432705455)

[第二章 促销相关技术及业务简述 5](#_Toc432705456)

[2.1 NGINX 简介 5](#_Toc432705458)

[2.2 PHP 简介 5](#_Toc432705459)

[2.2.1 基本概念 5](#_Toc432705460)

[2.2.2 应用领域 6](#_Toc432705461)

[2.2.3 基本工作原理 6](#_Toc432705462)

[2.2.4 选择PHP的优势 8](#_Toc432705463)

[2.3 MySQL 简介 8](#_Toc432705464)

[2.4 促销推荐简介 9](#_Toc432705465)

[2.4.1 促销商品热度数据 9](#_Toc432705468)

[2.4.2 促销系统推荐系统 9](#_Toc432705469)

[2.5 促销系统运行流程 10](#_Toc432705470)

[2.6 Hadoop 简介 12](#_Toc432705471)

[2.7 本章小结 12](#_Toc432705472)

[第三章 系统需求分析和概要设计 15](#_Toc432705473)

[3.1 系统需求分析 15](#_Toc432705474)

[3.1.1 功能性需求 15](#_Toc432705475)

[3.1.2 非功能性需求 17](#_Toc432705476)

[3.2 系统概要设计 17](#_Toc432705477)

[3.2.1 商品管理系统 18](#_Toc432705478)

[3.2.2 商品销售管理系统 19](#_Toc432705479)

[3.2.3 订单系统 20](#_Toc432705480)

[3.3 促销系统 21](#_Toc432705481)

[3.3.1 促销活动规则 21](#_Toc432705483)

[3.3.2 促销活动创建 22](#_Toc432705484)

[3.3.3 促销活动限购的设计 22](#_Toc432705485)

[3.3.4 促销活动均摊逻辑设计 23](#_Toc432705486)

[3.3.5 用户标签关联表设计 24](#_Toc432705487)

[3.3.6 反作弊设计 25](#_Toc432705488)

[3.3.7 促销活动热度数据 26](#_Toc432705489)

[3.4 本章小结 26](#_Toc432705490)

[第四章 系统详细设计与实现 27](#_Toc432705491)

[4.1 商品发布模块 27](#_Toc432705492)

[4.1.1 商品列表及发布 27](#_Toc432705493)

[4.1.2 商品详情 28](#_Toc432705494)

[4.1.3 商品编辑 28](#_Toc432705495)

[4.2 订单系统模块 29](#_Toc432705496)

[4.2.1 促销规则验证 29](#_Toc432705497)

[4.2.2 促销优惠券均摊 30](#_Toc432705498)

[4.3 促销系统模块 33](#_Toc432705499)

[4.3.1 促销活动发布模块 33](#_Toc432705504)

[4.3.2 促销系统商品限购 37](#_Toc432705505)

[4.3.3 促销系统热度数据 37](#_Toc432705506)

[4.3.4 促销系统智能推荐 39](#_Toc432705507)

[4.4 本章小结 43](#_Toc432705508)

[第五章 数据库设计 45](#_Toc432705509)

[5.1 数据模型 45](#_Toc432705510)

[5.2 数据模型详细描述 46](#_Toc432705511)

[5.3 本章小结 50](#_Toc432705512)

[第六章 系统测试及效果 51](#_Toc432705513)

[6.1 系统测试 51](#_Toc432705520)

[6.2 部分效果截图 56](#_Toc432705521)

[6.3 本章小结 57](#_Toc432705522)

[第七章 总结与展望 59](#_Toc432705523)

[7.1 总结 59](#_Toc432705524)

[7.2 展望 59](#_Toc432705525)

[参考文献 61](#_Toc432705526)

[致谢 63](#_Toc432705527)

[北京大学学位论文原创性声明和使用授权说明 65](#_Toc432705528)

# 绪论

* 1. 研究背景和意义

生活在互联网信息技术发展浪潮的今天，利用互联网信息技术，整合线上线下资源，高效的为互联网用户提供优质的服务，是大势所趋。电子商务就是一种目前被普遍接受认同的方式。近几年国内的电子商务层出不穷，经过几年的发展，人们正在享受信息化时代的浪潮。

北京某公司是中国著名的农产品交易平台，为中国上千家餐饮连锁店提供优质的蔬菜水果售卖服务。该公司目前主要采用跟农户对接，把采集过来的农产品对接到菜市场，餐饮店等。运营互联网技术打造这样的一个B2C平台。为农民增收减负，解决农民销售农产品困难，同时为餐饮行业提供优质安全的产品，免去餐饮工作人员每天凌晨去菜市场，采购食材之苦。

本项目就是在这样的背景下产品的一个基于电子商务平台而开发的促销系统，旨在提高公司销售额，提高市场份额，同时给用户以最优的价格提供给用户最优质的服务。促销系统作为每一个电子商务的重要组成部分，是公司运营发布活动的重要操作窗口，目前市面上各家的促销系统支持的活动形式各异。同时因为竞争关系，每家电商在实现自己的促销系统方式上也不尽相同。但是总的目的都是一样的，在竞争关系中占据至高点，同时让用户享受到更优惠的价格。促销活动可以定义为：给什么用户提供什么方式的优惠活动。不管什么促销总体上都是这样要支持：1.能够制定活动范围，2.发布什么样子的活动。

本项目在分析了产品的需求之上，设计了自己的促销系统技术框架。不仅实现了运营团队发布促销活动的任务，同时引入了当前流行的大数据技术并结合现有系统的DM(数据挖掘)标签管理系统使得运营人员发布的促销活动能够在用户端有效的查看到用户关注的促销活动。这大大减少了用户因为信息过载而未能发现它感兴趣的促销活动的可能性。从实际的效果来看，开发促销系统，根据公司BI部门的统计。在促销活动期间，销售额度提高200%相对于没有促销活动。同时添加促销推荐模块整体的销售额度，并且在这基础上再提高15%。相比于发传单之类的促销活动，使用信息技术平台不仅仅降低了人员成本，还能在促销活动中使得用户的反馈实现更精细化的运营。具有更大的价值和收益。

* 1. 国内外研究现状

电子商务系统就使用技术上相对已经比较成熟，更多的是在于企业运营的模式上的区分。该模式近年在国内外都有比较成熟的企业应用，比如国内的阿里巴巴和国外的亚马逊等。就促销系统而言，更多的是体现在促销的策略上，例如天猫的双十一，每个电商平台的促销各有不同并且促销活动多样。不同的电商平台对促销的重视程度也是不一样的，实现的方式也不同。比如亚马逊就主要依靠它的那套推荐系统，相对国内的京东而言促销活动种类比较少，或者使用优惠券代替。而京东的促销活动就比较多样且复杂。

大家熟知的天猫双十一的交易额，从某些数据反馈的结果可以反映促销对于一个电商是多么的重要，相比以外非信息化时代的促销，使用现代信息技术的促销具有更加高效、更低的成本、更快的传播速度。这些都是传统方式（类似发传单等）无法比拟的。在未来网络商城促销必然是大势所趋。

* 1. 论文的研究内容

促销系统因商品运营情况有时要减少滞销商品，有时因友商的一些促销活动，要能及时的提供一些促销方案。同时降低人力运营成本等总结出促销系统的总体目标为：

1. 能够低成本运营；
2. 能够灵活的发布促销活动；
3. 发布的活动能够方便用户知道；
4. 能够灵活的扩展出新的促销活动，应对多变的促销方案；
5. 能够对成本有很好的把控，要对促销的商品进行限购；
6. 为了防止亏损运营，一个商品不能参加多个促销活动。

通过对公司需求的分析，以及参考亚马逊、京东等电商的促销活动策略制定了一系列的促销活动规则。每个促销活动都有相应的一系列的活动规则结合。来规避亏损运营，同时使用用户购买记录，我的关注纪录，用户注册信息来给用户推荐相关的促销活动。

考虑现有框架和技术情况使用NGINX + PHP + MySQL的技术框架来设计发布促销活动。采用这样的框架不仅仅是因为开源免费。更多的是因为这些技术相对成熟可靠。对用户体验有一定的保障。同时他们相对灵活，高效配置也比较方便。

促销推荐系统的设计采用离线的方式算好。因为一般商户不会经常购买可能一天也就几次。正常一天购买一次。对时间要求不是很高。同时推荐算法要大量的计算。在用户购买的商品的时候，正确的做法是用户—登录就已经把用户可能相关的感兴趣的促销商品已经算好。这样用户—登录我们就能把他可能感兴趣的促销活动以最快的速度呈现出来。

同时页面的呈现部分，当用户的购物车还差多少可以参与什么促销活动，我们可以提示再买什么，再买多少免除运费。或者赠送某个商品活优惠券信息等。

* 1. 论文的组织结构

本文主要讨论了时下，电子商务的应用背景，以及从工程的角度讲述开发电子商务促销系统的对企业营销方案的支持程度。并针对需求情况，描述了整个项目的设计框架和部分详细设计细节。最后对整个项目进行总结和展望。

本论文各章具体细节如下：

第一章 简要的阐述了项目开发背景及其意义，并根据需求提出项目目标要求。

第二章 论述了相关技术的在国内外的发展状况，及业务环境。

第三章 介绍了系统需求设计、概要设计。

第四章 介绍了系统各模块的详细设计。

第五章 介绍了系统数据库设计。

第六章 描述相关功能的测试情况，及该促销系统的相关指标。

第七章 总结全文、展望未来，为后续研究开发奠定基础。

# 促销相关技术及业务简述

本项目采用NGINX + PHP + MySQL 的技术架构[1]来实现促销应用的实现。使用NGINX作为服务器处理用户请求，使用PHP来生产动态网页数据，使用MySQL来保存业务数据。使用Python来处理用户注册信息，购买记录等日志信息。使用Hadoop分布式计算平台分析用数据。项目中各个关键组件选择的开发环境如下：

表2.1 开发环境相关环境配置情况

|  |  |
| --- | --- |
| 名称 | 版本信息 |
| OS | Ubuntu 64 |
| NGINX | NGINX /1.4.6 (Ubuntu)。 |
| PHP | PHP 5.5.9-1ubuntu4.9 |
| MySQL | 5.5.41 |
| Python | 2.7.6 |
| Hadoop | 2.6.0 |



## NGINX 简介

NGINX聚焦于高性能，高并发和低内存消耗问题。同时包含多种WEB服务器便捷功能特性：负载均衡，缓存，访问控制，带宽控制，以及高效整合各种应用的能力，这些特性使NGINX很适合于现代网站架构。相对于APACHE服务器而言NGINX相对比较轻量级。但是他本身的处理高并发能力，是我们电子商务网站看重的一个重要特性。

## PHP 简介

* + 1. 基本概念

PHP原意为Personal Home Page (个人主页) 的缩写，现在已经正式更名为“PHP: Hypertext Preprocessor”（超文本预处理器）。 PHP于1994年由Rasmus Lerdorf[2]创建，后几经发展在WEB开发中占有非常重要的地位。它主要包含的特点是：

1. PHP语法混合了 C, JAVA, Perl以及PHP自创新的语法,集合了多种语言的优良语法特性；
2. 简单易学，开发成本低；
3. 支持所有主流的操作系统，包括Linux、Unix的各种变种，Microsoft Windows、Mac OS X等；
4. 模块化灵活，对于某些高性能的模块可以使用C、C++进行程序扩展。
   * 1. 应用领域

PHP能做的事情非常多，其主要用于服务端脚本程序，因此可以使用PHP来替代其他任何CGI程序能够完成的工作，比如生成动态网页，发送接收COOKIES, 收集表单数据等，同时它的功能远不止此，同时相比其他语言它的优良特性引导着越来越多的WEB开发人员选择PHP。其主要应用领域包括三个方向：服务端脚本、命令行脚本和编写桌面应用程序[3]。

1. PHP能够根据网站的访问者客户端的语言设置为访问者提供本地化的服务，自动的地以访问者的母语或习惯语言提供页面；
2. PHP能够很容易地创建FLASH、PDF等多媒体文件；
3. PHP能都有效地支持加密，并且支持多种数据库服务器。因此，[PHP](http://www.phpdo.net/)可用于企业内部管理系统的开发、B/S架构的电子商务系统的开发、公共网站的卡发等。

PHP主要擅长的服务端脚本：

1. 内容管理系统（CMS）：用于管理新闻、资料数据等。包括前台和后台。典型产品如DedeCms等。适合于个人网站及一般商业网站的应用；
2. 论坛系统（Forum）：是一个支持用户间传递和共享信息的交流平台。典型产品如Discuz等。Discuz在稳定、负载、安全等方面都是领先的；
3. 电子商务系统（e-Business）:典型产品如shopEX等。
   * 1. 基本工作原理

和很多其他WEB架构一样，都是被动的处理用户请求。一个完整的会话过程总是由客户端先发起的，然后客户的请求通过网络传输到服务器上，比如我们选择NGINX。然后服务器根据用户的请求分析用户要访问的数据由NGINX的WORK进程去处理用户的请求。服务器进行解析用户的请求资源，如果碰到请求的是PHP资源则调用PHP解释器执行，并把解析的内容通过服务器返回给客户；如果碰到要请求数据库资源，那么就会请求MySQL服务器获取保存的资源，然后组织业务信息，最后按照给定的逻辑通过网络再发送给用户的浏览器。浏览器接收到返回数据后呈现给用户观看。大致的过程如下：

1. 客户发送资源请求，整个过程遵从HTTP协议；
2. 然后底层协议栈通过路由算法找到合适的路由，最终发送给服务器；
3. 服务器一层层解析发送过来的协议数据，最终按照应用层的HTTP协议获取用户请求信息；
4. 如果请求的是静态页面nginx服务器会直接通过协议把数据发送给之前请求的客户；
5. 如果有动态页面则调用php解释器解释执行，这之间包括读取缓存或者数据库；
6. 在解释器执行完成之后会把数据交付给服务器由服务器通过协议把数据返回给客户呈现，正如图2.1 拓扑结构所示。

Master

Load configuration

Launch workers

Non-stop upgrade

Worker

ht\_core

ht\_upstream

ht\_proxy

ht\_fastcgi

Worker

ht\_core

ht\_upstream

ht\_proxy

ht\_fastcgi

Worker

ht\_core

ht\_upstream

ht\_proxy

ht\_fastcgi

HTTP

Web server

HTTP/HTTPS

FastCGI

Application server

memcache

Memcached

Advanced I/O

sendfile, AIO

mmap etc.

Cache loader

Cache manager

Backend

Multiplexing

via kevent/

epoll/select

proxy

cache

图2.1 PHP应用工作原理

* + 1. 选择PHP的优势

1. 代码开源：能够获取所有的PHP实现细节，同时可以学习借鉴其他社区优秀的设计思想，能够深度按需定制；
2. 免费性：任何商业公司，成本是一个不得不考虑的问题。和其他技术相比PHP不仅高效稳定而且是免费的。选择它作为技术实现没有没有成本费用；
3. 快捷性：相对其他语言PHP学习开发成本低，程序开发时间短，运行也比较快。编程简单高效。语言本身提供很多内置高效的函数库来处理一些业务逻辑；
4. 跨平台：可以运行在LINUX、UNIX等各种变种的操作系统以及微软的Windows和苹果的MAC OS X等；
5. 图像处理：可以使用PHP动态创建图像，PHP图像本身默认使用GD2来做图像处理，当然也可以通过配置使用image magick进行图像处理；
6. 支持面向对象：在我们项目选择的5.0以后的版本中，PHP 能够很好的支持一些面向对象特性编程。这样就能对于我们开发大型复杂的系统提供支持。同时应对复杂多变的促销需求，使用面向对象技术可以使用敏捷开发；
7. 有很多成熟的框架，比如ThinkPHP, CodeIgnode, Laravel等基于MVC的框架。能够很好应对开发需求。

## MySQL 简介

MySQL[4]由于其体积小、速度快、总体拥有成本低，尤其代码开源，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。其主要特点：

1. 使用C和C++编写，并使用了多种编译器进行测试，保证源代码的可移植性；
2. 支持众多主流操作系统，服务器架构部署便捷；
3. 为多种编程语言提供了API。这些编程语言包括C、C++、Python、Java、Perl、PHP等主流开发语言；
4. 支持多线程，充分利用CPU资源；
5. SQL查询优化，以及缓存技术能够有效的提高查询速度；
6. 提供优秀的索引技术支持；
7. 既能够作为一个单独的应用程序应用在客户端服务器网络环境中，也能够作为一个库而嵌入到其他的软件中提供多语言支持，常见的编码如中文的GBK、BIG5等；
8. 提供TCP/IP、ODBC和JDBC等多种数据库连接途径；
9. 提供用于检查、管理、优化查询数据库操作的管理工具，有比较丰富的工具集；
10. 可以处理拥有上千万条记录的大型数据库。

## 促销推荐简介

* 2. 1. 促销商品热度数据

本项目使用的热度数据，主要用于反应促销商品的一个喜爱程度。促销商品的热度主要从商品的购买数量，商品的关注数量，商品的销售时间属性，热度数据的生成主要是利用Hadoop技术[5]从不同的维度去统计。相关Hadoop的介绍参考2.6小结。

* + 1. 促销系统推荐系统

推荐系统和检索系统都是为了解决用户信息过载的产物。相比检索系统推荐系统主要解决的是挖掘长尾用户，对于检索系统而言一般应知道用户要什么，所以说用户有明确的关键字。而作为电子商务促销系统平台，促销活动的发布是有企业运营人员发布的，而用户不知道有什么样子的促销活动；但是用的注册信息，用户的购买行为明确的指示我们他们的喜好。本项目主要采用基于物品的协同过滤算法。

协同过滤算法[6，7，8]正如其名字，是大家一起协同把那些不相关的过滤掉。协同过滤算法主要有基于物品的协同过滤算法和基于用户的协同过滤算法这两种。正如其名基于用户的协同过滤算法是看看你买了什么，我拿你这个人的购买列表参考下。而基于物品的协同过滤算法是买了这个商品的人一般还会买别的相关商品。比如买了数据挖掘书籍的很有可能再去买机器学习。这两种算法各有优缺点。总的来说都是体现物以类聚的思想。因原理类似以基于物品的协同过滤算法进行深入研究[9，10，11]。

基于物品的协同过滤算法存在这样一种假设：如果用户对于一些商品有相似的喜好（可以是评分，评论，点赞等任何形式），则他们可能有相同的兴趣爱好。既然他们可能有相同的兴趣爱好，通过统计技术找出与目标用户兴趣爱好最接近的用户，然后根据这些最接近的用户的评分来预测实际用户的评分。

基于用户的协同过滤算法，思想与基于物品的协同过滤算法类似，都是找出与目标用户最近邻的用户，然后用最近邻用户的评分去预测实际用户。以用户A及用户B的购物车类比定义。相似度大致可以定义为：

相似度 ＝ |A ∩ B| / | A ∪ B|

或者通过余弦相似度［12］去计算：

余弦相似度 ＝ |A ∩ B| /

大致的意思就是AB两个用户他们购物车相同的购买商品的个数除以他们所有的商品个数。既然是要推荐，我们计算出相似度以后，还是远远不够的，一个用户对一个商品感兴趣的程度，是通过与他兴趣相似的用户购买的记录中推荐出来的。可以用如下的公式表示一个用户对一个商品的感兴趣程度：

如何评价一个好的推荐系统，业界一般使用如下一些指标：

* 1. 预测准确度；
  2. 覆盖率：对于商品长尾理论[13，14]的发掘能力，让尽量少的商品被淹没；
  3. 多样性：现实的环境事物一般都有多样性，环境的多样，使用的多样。多样性指标主要指能够覆盖的范围大小；
  4. 新颖性：用户可能不希望推荐它一些他一定会去买的商品，因为这样的推荐可能等于没有推荐。能否推荐一些他没有买过的商品。比如说：“啤酒炸鸡”当一中东西形成潮流，某个用户的相似用户可能会响应潮流，相似用户的购物车的变化会被推荐到和这些用户有相似行为的用户等。这样的推荐系统才能给用户带来新颖和惊喜。

## 促销系统运行流程

在该电商公司，整个业务线分为前端业务线和后端业务线。在这里要说明的是，这里的前端和后端的划分跟我们WEB技术的前后端不是一个概念。在这里业务的划分是根据用户购买的关系划分的。前端业务主要负责商品的销售，而后端主要负责商品的库存和物流仓库部分。促销系统是为了营销活动而衍生出的一个依赖于商品管理系统，商品销售管理系统，订单系统等。从一个促销活动的发布到用户购买商品享受一个促销优惠的活动大致的过程为：首先运营人员根据相关的数据，分析策划要发布促销活动，然后就在促销活动的录入接口填写相关的活动规则，活动一发布默认不会上线也就不会设置为有效。而是等运营人员把促销活动设置为上线状态后，在商城普通的用户才能看到相关的促销活动，相关的促销活动可能有很多。用户看到相关活动时，根据用户的购买记录利用算法计算出相关促销商品的热度，然后调用促销推荐系统计算出每个用户可能感兴趣的商品把他保存到数据库中。这样当用户登录的时候就能快速的查看到自己可能感兴趣的促销活动。如果用点击了该商品加入到购物车，购物车在结算的时候就会根据用户的信息已经购买的商品信息找出符合条件的促销活动信息，根据这些信息去创建订单，然后把订单中对每个商品的优惠信息都保存到订单系统中这样方便以后的售后服务比如说退货应该给用户退换多少金额等，至此一个促销系统的运行过程才基本完成。正如上文所诉，促销系统整个业务流程可以使用图2.2所示。

用户浏览商品信息

显示相关促销商品

运营

发布活动

规则检测

是否合法

禁止发布促销活动

创建活动

存储活动

上线活动

促销热度分析

计算价格

均摊促销

创建订单

是否有优惠券

购物流程结束

发送优惠券给用户

进入物流系统

否

是

是

否

图2.2 促销系统运行示意图

## Hadoop 简介

Hadoop[15]是一个由Apache基金会所开发的分布式系统基础架构。它是一款可靠、可扩展、分布式计算的开源软件。用户可以在不了解分布式底层细节的情况下，按照规定开发分布式应用程序。充分利用分布式云平台进行高速运算和存储。正如任何东西都有两面性Hadoop也是一样。

1. Hadoop集群的扩展性是其一大特点，Hadoop可以扩展至数千个节点，对数据持续增长，数据量特别巨大的需求很合适；
2. 使用Hadoop的成本很低，由于Hadoop是开源项目，因此所有的代码可以无偿使用并且可以定制化。另外一个就是它的可扩展性，对硬件的要求不高，能够充分发挥硬件资源的价值。
3. Hadoop 开源社区活跃，相关系统开源项目丰富。

同时Hadoop的缺点也是很明显的：

* 1. 任务内串行，不够快速高效；
  2. 不够可靠，响应时间完全没有保证；
  3. 中间结果不可见，不可分享；
  4. 单输入单输出；
  5. 图计算不友好；
  6. 大量的IO操作，效率低下；
  7. 迭代计算不友好。

但是在我们的促销推荐系统中，因为用户的购物活动不是那么的频繁一天一般来说就一次。也就是当天购买当天所需的商品即可。所以说在我们促销推荐系统中对Hadoop的时间要求不是很高。同时我们关系的只是通过推荐算法计算出来的，相关协同过滤算法的结果而已。即哪些促销活动用户可能更感兴趣。同时公司本身也有Hadoop系统集群，可以0部署安装使用。我们要做的就是简单的根据算法在空闲时间比如说凌晨在服务器压力比较小的时候编写定时脚本计算用户可能喜欢的活动然后保存结果到数据库，然后用户在登录的时候就可以直接查询出相关的结果。

## 本章小结

本章就促销系统所使用的各项技术做了系统的介绍。首先介绍了WEB服务器的选择，分析论证了NGINX和APACHE各自的优缺点。并且结合业务场景分析了选择NGINX的原因。同里分析了PHP和MySQL数据库。最后介绍了为了环境用户对促销活动信息过载或者说不对称而引入的促销推荐系统，旨在让用户最快的获取自己感兴趣的促销。这部分主要论证了为什么选择推荐系统的基于用户和物品的混合协调过滤算法，并在这个基础上描述到了使用的相关分布式计算。

# 系统需求分析和概要设计

* 1. 系统需求分析

经过与产品经理的多次沟通，整理出来关于本系统关于促销系统以下一些详细的需求。包括促销对象范围的确定，促销活动范围的确定和具体的促销活动相关内容。本促销系统的需求大致如下：

* + 1. 功能性需求
       1. 促销活动对象

促销活动对象，是指在促销活动发布时指定标签［16］的那批属于这些标签的用户，并且促销活动只对这些指定活动对象可见。促销活动对象可以使用三种标签来表示一类给定的用户。它们分别是固定标签，行为标签和自定义标签，所有这些标签都可以登陆DM标签系统查看。比如固定标签可以是这个用户的一些固有属性，一般为用户的规模等。而行为标签主要是用户的购买行为，比如首次购买时间在2015年6月1日之前的用户，或者用户的组册时间在2015年7月1日之前等。所有指定的标签之间默认使用并集的关系。如果选择了一个固有标签用户规模大小为10平米以下，同时选择了首次购买时间在2015年8月1日之前，则表示所有满足用户规模是10平米以下或者首次购买时间在2015年8月1日之前的用户都可以参与该促销活动。如果选择逻辑交，则表示要求同时满足两个条件的用户才可见该促销活动，如果选择逻辑差则表示所有满足用户规模在10平米以下但是首次购买时间都在2015年8月1日以后的用户才可见。最后要去活动对象的逻辑运行是左结合的，一次从左至右计算所有的标签逻辑运算结果。

* + - 1. 促销活动时间

活动时间包含活动开始时间和活动结束时间以及最晚配送时间。活动开始时间和活动结束时间界定促销活动的有效活动时间，因部分商品配送的要求规格的区别，有些商品的配送时间可能不同。有些商品可能因促销活动带来配送时间的变化，因此指定一个最晚配送时间，促销活动的条件要求用户的购买时间在促销活动界定的时间范围内，同时有些商品可能要过几天才能配送到位，因此只有用户选择的最晚配送时间在活动指定的最晚配送时间之前的才能参与此促销活动。

* + - 1. 促销活动商品

促销活动商品包含两种粒度的范围指定，一种活动范围仅包含单个商品，另外一种包含一类商品。运营人员可以填写这两种类型的范围来指定哪些商品参与促销活动。预定两者都不填代表整个网络商城的所有商品都是促销商品，其他情况取二者指定商品的并集。

* + - 1. 促销活动类型

不同的促销活动有不同的优惠策略，要求促销系统能够快速的添加不同活动的类型。促销活动包括单品促销，套装促销，满减，满赠，打折。其中满赠可以赠送优惠券或者赠送某些商品，所有的买几赠几都可以用满赠来表示。

* + - 1. 促销活动限购

促销活动有些会有限时抢购，很多都是商业策略。所以为控制成本有些商品参与促销活动要求要有限购策略。限购有两个维度，一个是单个客户的维度，一个是所有该促销活动的可见用户。

* + - 1. 促销活动其他限制条件

促销策略要求使用微信支付的都优惠5元。同时为了支持不同渠道发布促销活动支持，添加订单来源。用户的订单来源有H5页面，PC端，移动端。用户的支付类型包含微信支付，账期支付，银行卡，支付宝。

* + - 1. 促销活动推荐

促销推荐包含两个部分一个促销热度统计分析，这样运营人员就可以有针对性的提出可靠的促销活动供运营人员参考分析。同时促销热度数据分析可以供推荐系统分析结果。

促销推荐使用基于物品的协同过滤算法，因为在中国有不同菜系之分有些菜系经常做的食材也是不同的，具有对同样的商品有近似兴趣的，很可能是相同的菜系，同时基于物品的协同过滤算法做推荐解释的时候也比较方便，不会侵害用户隐私。把促销推荐的结果保存起来每日一更新。网上商城部分用户只要登陆购买时候查询即可。

促销活动优惠信息可追溯，用户一次购买行为的所有信息都会保存到订单之中。用户可能在一次购物活动中有部分或者全部商品满足促销条件而参与优惠。如果用户对某些商品不满意要求退货，因为优惠活动有些是赠送优惠券，有些是满减直接减免了金额，对于减免了金额的该如何给用户退款。需要把所有的减免信息保存到订单中。

* + 1. 非功能性需求
       1. 性能需求

因促销模块要求大量的计算，包括促销推荐，促销热度统计。同时用户购买到订单生成要走大量的核心流程，要求系统每秒并发访问500以上的访问量。对于单次购买系统的响应速度不能低于2s；

* + - 1. 稳定性及故障处理

系统要有一定的容错性，当有错误数据输入时要及时的过滤，万一没有办法过滤要有备选方案，让程序的业务逻辑能够正常的执行不会报错。如果系统遇到重大错误要求有日志系统跟踪问题所在同时要求有报警机制短信提醒开发人员或者邮件提醒开发人员。以让业务能够快速正常执行。

* + - 1. 可维护性及可扩展性

系统开发要求遵循统一的规范，包括变量，函数，类名等命名规范。开发的功能要求与文档中的描述同步，系统在设计接口的时候要求尽量的考虑到扩展性这一部分。应该尽量使用一些设计模式，同时对关键的代码要求注释便于日后查看修改和便于他人阅读。

* + - 1. 可移植性及部署

因业务扩展需求，系统开发过程中要考虑系统的可移植性，应该尽量使用成熟的技术和可跨平台的语言进行开发。防止将来移植时发生错误，同时要求系统方便移植和调试。以保证系统能够快速正常的执行。

* + - 1. 相关接口文档清晰

拥有清晰的文档不仅可以减少沟通成本，而且也方便日后维护和扩展要求促销系统提供相关接口描述文档和示例以及该接口的注意事项。

* 1. 系统概要设计

本小结首先介绍了跟促销系统相关的子系统的构造结构和相互关系。并通过分析一个促销系统的发布，到用户购买商品享受促销活动，再到最后的退货等售后服务，总之促销系统相关的模块众多。

从业务逻辑上讲，促销系统首先要保证促销的商品要有库存，库存这块关联到WMS（仓库管理系统）。然后是必需要有商品信息,这设计到PMS(商品管理系统)，同样的物品在不同的地区售价和规格也不相同，由物品变为商品要在SMS（商品售卖系统里面）设置。然后用户在购物的时候才能看到有促销活动，用户在下单购物的时候要创建订单这涉及到订单系统，然后物流系统根据订单去配送商品。所以一般来说跟促销系统相关的电商子系统包括商品管理系统，仓库管理系统，销售管理系统，和订单系统。如3.1图所示：

促销系统相关模块

商品管理系统

商品销售系统

仓库管理系统

订单系统

商城系统

我的关注

图3.1 促销系统相关模块结构

* + 1. 商品管理系统

商品管理系统，主要的职责就是把采购过来的产品进行加工处理变成预卖商品。在库存中的形式就是形成一个SKU=Stock Keeping Unit（库存量单位）。

主要的功能就是显示相关货物信息，添加货物信息，修改货物信息，删除活动信息，添加货物分类映射。其中添加一个货物信息一般主要指货物的规格属性包括：名称、所属分类信息、规格、图片、保质期、重量、库存等货物基本属性。在这里添加的货物跟商品的最大的区别就是没有价格信息。同时一样东西在不同的地区，人们对他有不同的叫法，为了能够让不同地域的用户在浏览商城的时候能够清晰的知道商品的名称做了一个这样的分类映射，方便地域性显示。商品管理系统里是不售卖的商品，促销系统跟商品管理系统有关系是因为有满赠类的活动，满赠的赠送物品可以是售卖的也可以是不售卖的，但是赠送给用户肯定是必须存在的物品。整个商品管理系统结如图3.2所示。

商品管理系统

规格管理

分类管理

货物管理

图3.2 商品管理系统结构图

* + 1. 商品销售管理系统

商品销售管理系统主要集中跟销售运营的所有活动。可以简单的理解为在物品上添加价格。以及发布相关运营活动的这样一个系统。所以整个销售管理系统大致包括的功能就是商品管理，商品广告管理，优惠券管理等。其中商品管理包括商品列表显示，商品发布，商品价格的制定；广告位管理发布广告商品信息，广告商品信息列表，下架广告商品等；优惠券包括发布优惠券活动的规则，发布优惠券活动，系统中优惠券设计两种方式一种是用户促销的优惠活动，另外一种是系统促发的优惠券活动，商品销售管理系统是整个电商平台的一个重要组成部分。该系统集成了所有的跟商品运营活动有关的功能，促销系统最终也要集成到销售管理系统上给运营人员发布促销活动。整个商品销售管理系统的拓扑图如图3.3。

商品管理系统

商品定价策略

发布优惠券

商品改价

发布商品

广告管理

首页推荐

专题模块

库存管理

券码管理

上

下架商品

标签管理系统

广告位管理

修改商品属性

图3.3 商品销售管理系统结构图

* + 1. 订单系统

任何订单流程都是电子商务的核心。处于商业安全的原因，这里只大概的讲述下跟促销系统相关的一些订单创建的流程。用户在浏览电商网站购买商品的时候，相关的商品列表，商品详情只会显示那些商品销售管理子系统中发布的商品。所有的商品都会去读取商品信息表。商品在商品销售系统中发布商品的时候会创建一条商品。一般来说会保护该商品的一些基本属性和价格属性。然后再结合促销系统，就可以确定哪类商品或者哪个商品有促销活动。促销活动有相关促销活动关键字和促销广告语。会在之后的促销相关设计或者表结构设计中有体现。

然后用户再浏览商品的时候就能看到有促销信息。如果用户把商品添加进入购物车，那么商品会调用促销活动相关接口检查购物车商品是否满足相关的促销活动规则。比如说会查看购物车购买商品的价格，用户选择的支付方式，用户所在的地域信息，用户购物的时间，用户的历史购买记录等去刷选出符合用户的促销活动信息。如果满足，用户就会把相关的促销活动ID号一同提交给订单系统。然后在订单系统里面，我们要计算是哪些商品满足促销活动，哪些商品不满足促销活动。比如说满减活动，那么我们要把相关的优惠信息均摊到每个商品最小单位。然后持久化到订单系统里面。这样我就能根据订单的情况收取相关费用和根据订单系统记录的数据提供售后服务。比如说退货，设计到促销活动的应该如何给用户退钱。这也是整个促销系统的实现设计到比较核心的一个流程。具体的流程如下图3.4所示。



图3.4 促销活动使用流程示意图

促销活动整体上可以分成两大部分，即运营发布促销活动，用户使用促销活动（包括促销推荐）。整个流程为用户登录商城看到促销推荐系统推荐好的促销活动，用户可能买也可能不买。用户看到自己想要购买的商品加入购物车，促销活动有很多类型：比如说单品促销，套装促销，满赠，满减等。用户把一些参与促销活动的商品加入到购物车，系统通过分析购物车中的商品，提示用户加买什么商品可以享受什么优惠。或者还差多少钱能够参与什么类型的活动。等用户提交了订单，用户可能对我们提供的服务、商品不满意要提出退货处理。但是用户购买的商品都是参加了相关的优惠活动的。因此所有的订单系统，在创建订单的时候要明确的记录下哪些商品参加了什么类型活动。另外由于物流的原因有些商品可能不能一并交付给客户。需要按照一定的规则把订单拆分成相关子订单。然后优惠的信息统计到订单和子订单上。并记录所购买的商品哪个是参与了促销活动的，而哪个没有。这样的设计就可以保证用户在提出退货的时候能够清晰的知道应该怎样给用户返现。

* 1. 促销系统

本章前文分别从总体和局部说明了促销系统在整个电商的应用部署环境，然后再分别介绍跟促销系统相关的子系统的功能和架构特点。在接下来的这个小结中将详细讨论如何设计促销系统，并完美的整合到该电商公司的所有相关系统中去。

促销系统的本质就是要能够发布促销活动，提高运营效果。但是公司发布促销活动，往往是做出了让利运营的，有些可能亏损运营。所以如何控制多个促销活动之间不发生冲突。是促销系统设计的一个重点。同时促销活动种类繁多，如何设计一个良好的框架，能够对未来扩展促销活动类型做好铺垫。

* 1. 1. 促销活动规则

促销活动规则主要的作用就是规定当有多个促销活动同时满足的时候以一种怎样的规则就选择多个同时满足的促销活动中的一个活动。在本项目当前的需求是要支持五种促销活动，分别为单品促销，套装促销，满赠，满减，打折。单品促销活动就是只针对某个特定的商品做促销活动。同时运营也可能需要发布对某一批商品进行促销活动。所以除了单品促销和套装促销活动是制定一个或几个特定的商品集合。其他的促销活动都是可以有两个维度来发布活动的。即在发布活动的时候除了可以指定某个特定的商品参加促销活动，也可以指定某些分类参加活动。没中类型都有两种可能所以总的来说发布促销活动有4种可能的范围如表3.1。

表3.1 促销活动规则范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 是 | 是 | 否 | 否 |
| SKU | 是 | 否 | 是 | 否 |
| 结果 | 指定分类和SKU参与活动 | 指定分类参与活动 | 指定SKU商品参与活动 | 全场商品参与活动 |

怎样保证让同一个商品不会出现在两个促销活动中。一种可行的办法就是设置优先级。但是如果把优先级交由运营人员是不可靠的。因为不同的人，甚至同一个运营人员在不同的时间段内可能会把优先级设为相同而退化为没有优先级。如果程序给固定死优先级，这样必然带来添加新促销，或者改动促销优先级带来的开发成本。同时软件敏捷性不高。比较折中的办法是配置优先级，在给定的优先级下，如果一个商品满足多个促销活动有限按照优先级高的处理。

但是规则这样的优先级还不够，假如给蔬菜做满减活动满200减10元，同时对某个蔬菜的子分类比如说叶菜做促销满赠，比如说买五赠一。这样就导致了有些促销活动不能叠加。有时候为了增大促销力度，可以有叠加活动的促销。

* + 1. 促销活动创建

活动的创建主要就是在给定的规则下要让活动的创建有足够的灵活性。对新的促销活动能够快速的支持而不要对以前的代码进行修改实现开闭原则。活动的创建流程可以理解为首先验证运营表单数据的正确性，然后依据促销规则信息进行检查，如果都合法就创建促销活动。同时为了灵活的创建促销活动使用工厂模式创建活动。促销活动公共部分保存到促销活动属性上，每一个促销活动可以绑定一条或者多条活动规则。

* + 1. 促销活动限购的设计

促销限购指一个促销活动对每个用户在享受活动的一种限制。考虑两个维度：一个为对用户的限购，一个为对整个系统的限购。其中对用户的限购需求是用户每天限购的数量。比如发布一个全场满减的活动，满200减10块。但是因为库存，经济性因素的考虑需要对促销商品进行限购。如前10减促销，之后的正常购买。用户促销购物车分析的时候一个分析的维度去检查促销限购快照表，然后去计算当前购买的商品数量和当前还剩余的限购数量，如果当前的限购数量大于0，代表有商品可以参加本次活动。如果购买的商品数量小于剩余限购数量更新限购数量，否则为0。

促销活动的限购有一个并发性［17，18，19］的问题，比如限制总限购200件商品已经出售了190件，在这个场景中还剩下10件可以参与促销活动，如果有多个用户同时购物并且同时都满足促销活动，那么该给哪个用户促销优惠。此类问题有个经典的解决方案乐观锁［20］和悲观锁［21，22］。在本项目中使用了乐观锁的机制解决了该问题。用户在购物时为查看是否是限购信息会拿到一个最新的版本号，如果得到的是最新的版本号并且最新的版本号有足够的限购数量则满足规则。其他情况则要更新版本号，更新后的版本号限购的数量会变为0表示不能购买了。

* + 1. 促销活动均摊逻辑设计

活动发布，用户享用了促销带来的优惠，但是因各种原因用户可能会退货及其他相关服务。为了让服务能够可跟踪。必须要设计相关的字段信息来保存用户购买了哪些商品参与了哪些促销活动。主要记录的信息包括：用户信息，用户购买了什么商品，商品的价格和数量，商品使用的优惠券信息，商品实际享用的活动信息。已经因为防止用户恶意使用促销活动而做的防止作弊的逻辑。反作弊的详细细节参考3.3.6小结。

用户购买商品哪些会参与促销活动，首先按照促销活动的优先级顺序把所有的有效的促销活动并且时间在用户购买时间范围内的活动都取出来并且按照优先级排序。然后一次分析商品购买记录，分析购物车已经满足了哪些促销活动，并且提示还差什么满足条件满足更多的活动。计算完成后如果用户购买了相关推荐的商品，那么就会满足更多的促销活动，如果用没有购买商品或者指定的商品满足活动。那么订单活动的均摊逻辑就会首先找出哪些商品是参与了促销活动的。然后把相关的订单数据记录到订单详情，子订单，母订单上。同时在计算满足促销活动的时候要计算出限购要利用到限购快照表。会在数据库那一章节中详细介绍。整个促销订单均摊逻辑的计算过程可以总结为：购物车促销活动分析，检查促销活动，检查限购促销表。计算满足活动列表和不满足活动列表，其中不满足促销活动按照优先级和差额排序。整个过程设计到的相关数据如图3.5。

促销系统相关模块

订单

子单

订单详情

促销活动

促销规则

限购总数快照表

限购总数快照表

图3.5 促销均摊计算相关表数据

* + 1. 用户标签关联表设计

促销系统有时候要能够针对特定的用户群在发放相关活动，现有系统有用户标签系统。用户标签系统包含三种类型的标签固有的标签，行为标签，自定义标签。发布促销活动相关表不可能保存所有的用户id，尤其是用户规模量比较大的时候把所有的用订单保存到用户促销表的一个字段上有严重的问题的。但是要指定一批特定的用户，在发布促销活动的时候使用用户标签来指定用户范围，而不是直接的用户id。而在用户范围促销活动的时候，提供一个接口输入用户信息查询出与该用户相关的标签信息，这样就必须设计到一张用户标签与用户id的一张映射表。逻辑是在发布活动的时候，根据促销活动相关的标签，找出所有符合条件的用户，然后建立映射表。用户在登录的时候，根据用户id去查这张映射表，如果没有查询到该用户的相关信息，那么该用户肯定不在促销活动范围之类。如果查询到了该用户的id，再使用用户标签信息和促销活动标签信息进行对比。如果对比成功，则表示有促销活动，如果没有则表示没有活动。指定用户的逻辑过程如图3.6所示。



图3.6 促销活动指定用户范围逻辑

* + 1. 反作弊设计

促销的本质就是为了提高销量和企业品牌效应。促销商品一般售卖价格很接近成本价格。因此有相关商品促销活动要求限购。限购的需求就是只要用户有促销活动就要求用户能够参与，不管是全体还是部分。但是如果不加以限制就有可能出现商品大量低价卖出。反作弊[23]系统就是在活动规则的基础之上设计的一个简单逻辑判断，由于时间有限，目前的反作弊系统主要的功能就是在超过相关促销边界值的时候，进行相关的设定限制。但是恶意用户可能会想尽各种办法，但是这样毕竟比较少。所以本版本的促销反作弊的主要设计为：如果促销活动中有限购策略，那么如果用户购买了一部分商品之后他再次购买活动以获取两次减免金额。系统在历史快照里查询分析历史记录如果历史里面已经买了同样的一个商品N次。我们的反作弊系统就是做一个限制。现在用户一天能够使用多少次，只要超过反作弊系统设定的阀值就不会出现促销活动。同时另外一个维度的限制就是如果用户购物的过程中发现只要买了什么商品就可以有促销活动。但是这个商品限购10件。如果用户利用系统先购买9件，然后再购买100件。我们的反作弊系统只会记录一个给定的限购值10%。也就是说如果你先购买9件，如果在购买100件，我们认为这种反常的购买行为可能包含恶意攻击。提示用户明天再继续购买。

* + 1. 促销活动热度数据

促销热度统计的是用户在商城活动中的一些数据反馈。统计的维度包括用户注册时间，用户类型，用户商城购物记录，用户商城浏览记录，用户我的关注信息列表，搜索记录，商品评价。

* 1. 本章小结

本章通过对需求的深入分析，并且结合现有系统的特点。分别介绍了和促销系统相关的商品管理系统和销售管理系统这些订单系统的特点。在了解这些系统的特点和新促销系统的需求分析，结合系统论述和再结合相关系统的情况下，分析了促销系统的开发和大致实现。下一章中将对这些相关模块做详细的说明。

# 系统详细设计与实现

本章主要是在之前章节的概要设计的基础上，详细设计促销系统各功能的实现细节。先后分别介绍了跟促销系统相关的商品发布模块和订单模块部分的详细设计实现过程，然后分析促销系统本身的详细设计与实现，最后介绍促销推荐模块的设计与实现。

* 1. 商品发布模块

本节介绍商品信息相关的设计与实现所有促销系统发布的促销商品均来自本小节发布的商品，商品发布是一个纯运营后台操作的发布商品平台。所有的商品信息都是实用WEB技术保存到数据库中。这块设计的核心就是规定好保存商品字段信息，然后收集表单数据保存的过程。本小节先后介绍了商品列表，商品发布，商品详情，商品编辑的详细设计信息。

* + 1. 商品列表及发布

商品发布主要用来发布一个新的可以售卖的商品。本项目中发布商品的流程相对简单，首先收集表单数据，然后把新的商品信息发送给服务器，最后在后台数据库中保存设定好的商品信息，供之后查询编辑。

商品的信息主要包含：所属分类, 提供商家信息，计量单位，名称，图片信息，位置信息，价格，市场价格，状态，规格，货号信息，结算单价，结算单位，售卖地区，库存量，创建时间，更新时间。其中商品的价格和商品的市场价格供定价策略使用。商品的包装规格信息都保存在规格信息中，商品的价格就是商品的结算单价乘以商品规格中数量，重量信息等。比如商品结算单价为斤，那么如果包装规格中说，5斤／包。那么该商品的价格就应该为结算单价乘以五。商品发布功能如表4.1所示。

表4.1 商品发布

|  |  |
| --- | --- |
| 功能描述 | 发布一个商品信息，能够在商品列表页面显示 |
| 输入内容 | 商品属性信息表单数据 |
| 输出内容 | 提示活动发布成功或失败 |

为能方便查看有已经发布的相关商品提供显示功能如表4.2 显示已成功发布商品信息

表4.2 商品显示

|  |  |
| --- | --- |
| 功能描述 | 查询数据库，显示所有已发布商品 |
| 输入内容 | 无 |
| 输出内容 | 已发布商品信息 |

发布的商品一般会有很多，如何快速的找到相关系统必须提供一个查询功能如表4.3 商品查询

表4.3 商品查询

|  |  |
| --- | --- |
| 功能描述 | 通过给定关键字查询符合条件的商品 |
| 输入内容 | 搜索关键字 |
| 输出内容 | 符合搜索关键字的商品信息 |

* + 1. 商品详情

商品的属性信息包含价格地区等很多的信息在一个列表中很难都显示出来。所以在商品列表页显示的商品信息是经常需要查看的高频字段信息。业务上有时候也需要查看商品的低频属性信息这些信息必然要提供一个查看入口。就是表4.4所示。

表4.4 商品详情

|  |  |
| --- | --- |
| 功能描述 | 根据提供的商品ID，查询出该商品的详细信息。不存在返回空 |
| 输入内容 | 商品ID |
| 输出内容 | 商品所有字段信息 |

* + 1. 商品编辑

商品的信息因包含很多字段，为提供快速修改信息的功能必须提供相应的编辑功能：

表4.5 商品编辑

|  |  |
| --- | --- |
| 功能描述 | 根据提供的商品ID，查询出该商品的详细信息并初始化表单数据。不存在返回空 |
| 输入内容 | 商品表单详细信息 |
| 输出内容 | 提示更新成功或失败信息 |

* 1. 订单系统模块

订单系统模块从业务的角度主要归属于使用促销活动部分。用户购物如果享受了优惠就应该把优惠的信息计入到订单详情中去，这样的目的是方便售卖的商品日后的服务，比如说发生退货现象金额计算问题。同时对统计促销活动相关的统计也是很有帮助。本小节首先从用户的角度讨论了在一次购买中促销规则的验证设计实现，然后介绍了参与优惠活动记录把促销价格均摊的设计细节。

* + 1. 促销规则验证

用户确认购买商品生成订单之前，系统首先会对用户购买的商品进行合法性检查。检查的范围包括购买商品的状态信息，商品的价格，库存等。所有购买的商品只有合法的商品才可以被售卖。同时在这个检查的过程中还包含一个重要的检查项目就是购物车是否满足促销规则。如果满足相关的促销规则会在确认订单页显示促销活动相关信息，因系统订单流程遗留一个问题就是最晚配送时间要在点击生存订单的时候才去让用户选择他们满意的配送时间，而用户选择的最晚配送时间可能和公司运营发布的促销活动的最晚配送时间冲突。所以在把促销系统嵌入到网上商城订单流程中的就有两次促销检查。这样做优点就是系统更稳健（两次检查内容不同），把错误数据都拦截在订单生成之前。

在这个业务场景下，促销系统的验证也包含两次检查。第一次把除了支付方式和最晚配送时间之外都满足条件的促销活动都显示给用户，可能有多个活动，但是一次购物只能选择一个促销活动。

表4.6 促销检查

|  |  |
| --- | --- |
| 功能描述 | 根据用户信息，购买时间，购买商品信息检查出所有符合条件的已发布促销活动（除支付方式和最晚配送时间），然后让用户选择多个活动中的一个 |
| 输入内容 | 购物车商品信息，用户信息，购买时间，支付方式，最晚配送时间 |
| 输出内容 | 所有已发布符合促销规则的有效活动 |

业务效果：用户提交订单，在提交订单的时候用户能够看到符合条件的相关促销活动信息，活动可以是满减，满赠，打折，单品促销，套装促销等多种类型。用户选择完促销活动，选择支付的类型，选择商品的配送时间，系统计算用户满足的促销活动，进行促销活动的验证。如果验证促销活动满足条件，则通过促销规则检测可以参与此活动。反之，则视为该促销活动对该用户本次购买不可见。促销验证的活动规则流程图4.1。



图4.1 促销系统促销活动验证流程

* + 1. 促销优惠券均摊

用户在一次购买行为中，可能有部分或者全部的商品满足促销活动。如果只是购买所有商品的中的部分满足促销活动，，那么记录到底是哪一部分商品使用的促销活动是很有必要的。订单的系统的逻辑分为母订单，子订单和订单详情。母订单记录了一次购物活动的全部宏观信息，同时不同的商品配送物流成本不同，根据购买商品的信息把订单按照采集类型分为子订单。子订单只是把母订单拆分出来，便于配送。订单详情则记录了购买商品的详细信息。

其中母订单记录一次购物中此次购买行为参与了什么促销活动。这个促销活动信息要根据购买商品划分出现在其中的某些子订单中。一个母订单对应多个子订单，如果母订单中有促销活动，而某个子订单中没有促销活动信息，那么促销活动信息已定在其他子订单中。然后在订单详情中把优惠的价格平均绑定到每个具体的商品上面。在均摊逻辑的计算上使用整除，因浮点数可能有问题，采用整除策略，即对于参与活动的N件商品，前面N-1件商品的价格取整，最后一个取总优惠金额减去前面商品之和。

表4.7 促销活动优惠均摊

|  |  |
| --- | --- |
| 功能描述 | 订单创建完成，能够把参与促销活动那部分商品的优惠金额记录到母订单，子订单和订单详情上 |
| 输入内容 | 促销活动信息，购物车商品 |
| 输出内容 | 提示均摊成功或者失败 |
| 备注 | 均摊成功能够查看订单中什么商品减免的多少金额或其他 |

订单均摊逻辑的详细执行过程为：根据用户的信息和购物车购买商品信息，从促销系统的处理过程是，计算总购物车商品，把促销活动的信息绑定到订单之上，把促销活动的减免金额与用户购买的金额做比较，看是否为安全值，如果安全就把减免金额绑定到订单上，然后按照商品的采集类型去拆分订单，不同的子订单有不同的配送系统。同时要方便统计使用，按照购物车商品的详情，已经商品的属性，必然会把参与一个活动的商品拆分到不同的子订单上。如本小节前文描述，为了能够处理优惠的精度不丢失太多。使用的逻辑就是按照采集把商品分成多个类别。同时根据促销活动指定的类型有两种维度分类，还是特定商品。然后不同的子订单，按照参与活动的贡献比例，把相关促销信息进行绑定。如果母订单中有促销信息，但是在一个子订单中没有商品参与活动，那么这个子单上面就不记录促销相关信息。如果这个参与的子订单只要有商品就应该记录促销信息。如果满足促销活动的商品分在两个不同的子订单中，那么就逐个处理统计订单的减免金额，及相关促销活动。最后的那个减免金额取总额减去其他金额。详细的订单均摊流程如图4.2所示。



图4.2 订单均摊业务逻辑流程示意图

* 1. 促销系统模块

促销系统的发布和使用是促销系统中两个重要的终端。本小结主要讨论促销系统中跟促销有关的详细实现细节。

1. 3. 1. 促销活动发布模块

促销活动发布涉及到一个复杂的冲突检查问题。一个商品可能同时满足多个促销活动。每种活动有不同的优先级，哪些活动能够共存，哪些活动不能给共存。本项目开发阶段产品要求做了限制，同一种商品最多只能参加一种促销活动。如果是可以共存的促销活动则只算一次活动。促销活动冲突检查包含两种粒度的检查和两个时间维度的检查。即指定商品、指定分类在当前发布的活动中和历史有效活动是否冲突。如果有冲突则提示活动不允许创建。

表4.8 促销活动冲突检查

|  |  |
| --- | --- |
| 功能描述 | 检查当前发布促销活动是否和现有的有效促销活动冲突，冲突的定义为一种商品出现在两种以上的促销活动中（除特别情况外） |
| 输入内容 | 促销活动范围信息 |
| 输出内容 | 真或假 |
| 备注 | 活动发布前必走流程 |

要求能够快速的支持一种新的促销规则活动的要求。使用工厂方法模式来创建每一种活动。同时用户在使用活动的时候只要知道活动ID。因为不同的活动有不同的规则，所以促销活动表通过促销活动类型和促销规则ID就可以联合一个具体的促销规则。但是不同的活动促销规则没有对系统造成任何别的影响。在支持新促销活动的时候只要创建一个相关活动的规则类就可以了。同时因为不同的促销活动有不同的促销使用规则，本系统使用策略模式来计算每种促销活动的减免金额。促销活动创建的类图模型见图4.3：



图4.3 类图模型

发布的活动如果跟之前已经发布过的活动有冲突就提示运营人员你这样发布促销活动，可能会导致用户购买活动享受重叠优惠。如果运营人员在发布促销活动的时候选择了该活动可以和其他运营活动共存，就不需要进行冲突性检查。如果运营选择不共存，就需要对发布的活动进行判断。首先判断当前活动指定的商品是否出现在以前的活动中，如果当前发布的商品出现在以前发布的活动或者发布的分类活动中就提示不允许创建。或者当前的发布的分类商品包含了以前的商品。每种有两种可能共四种检查的可能性。这个活动的发布流程如图4.4。



图4.4 促销活动发布流程

* 1. 单品促销

只针对某些特定商品做促销，促销的方式有直接给定优惠价格或者打折或者赠送优惠券。

输入数据：输入指定的商品和促销价格，限制购买数量

使用场景：运营发布指定商品优惠促销，设定该商品的售卖价格和购买数量。如果用户能够看到该出校活动就可以使用该促销活动，并且相印的会显示促销减免的优惠价格。即原始价格减去促销价。

* 1. 套装促销

该促销模式主要支持捆绑销售策略。

输入数据：套装促销商品及件数

功能简介：运营指定一些商品集，这些商品集合包括商品的种类和数量，当用户购买的商品满足了商品的种类或者数量的时候就会减免设置的优惠价格。

* 1. 满减促销

输入数据：指定商品或指定分类参与促销活动，活动每人购买限制数量，活动总体限制购买数量，活动满减规则，支持阶梯减免方案。比如满100减10元，满20减20元等。

使用场景：用户购买商品，如果在购买的商品中不管总类还是件数都满足满减活动（因活动优先级关系抛除掉那些已经计算过了的商品），用户在阶梯满减活动中会有限使用价格最优的促销活动。即有限使用价格对用户最有利的活动。

* 1. 满赠促销

输入数据：指定商品或者指定属于某个分类的商品，活动每人每天限购数量，活动整体限购数量，活动满赠规则方式。主要包括两种类型是满多少钱赠送物品还是满多少钱赠送优惠券等。可以总结为满赠物品还是满赠优惠券还是免运费等。

使用场景：如果用户购买的商品满足促销活动的相关规则包括限购，在活动期间，商品购买的价格满足活动要求等等。只要用户的购买商品满足了促销活动需求就能参与满赠活动。参与满赠活动赠送的商品价格为0。如果赠送给用户的是优惠券信息，那么赠送的优惠券不能再当天购买的活动中使用。

* 1. 打折促销

输入场景：指定促销活动范围包括制定分类商品或者指定的商品信息，每人每天限购多少数量。同样支持阶梯打折，买的越多打折的力度就越大。

使用场景：用户购买的商品如果满足打折活动，在结算统计用户购买的商品价格的时候会统计打折价格。

所有的促销活动都设计成每个促销活动自己计算记得的打折方法，在折扣计算方面使用了策略模式，当分析用户购物车分配合适的促销活动给用户时，分配促销任务模块会按照如下优先级使用，活动的优先级规则表4.9：

表4.9 活动优先级定义

|  |  |  |
| --- | --- | --- |
| 活动名 | 优先级 | 备注 |
| 单品促销 | 1 | 最高 |
| 套装促销 | 2 | 第二高 |
| 满减 | 3 | 满减、满赠、打折相同 |
| 满赠 | 3 | 满减、满赠、打折相同 |
| 打折 | 3 | 满减、满赠、打折相同 |

* + 1. 促销系统商品限购

限购包含全体限购和每人每天限购两种方式。解决限购问题的并发数据同步问题使用了类似乐观锁的机制。如果通过查询订单来分析限购数量将非常麻烦且容易出错。通过建立两个中间表不仅简单高效同时稳定不易错还效率高。详细的数据库中间表设计见第五章数据库设计部分。

对于全体限购部分保存促销活动ID,SKU号，限购数量，版本号，查询Token。查询Token为促销活动ID加SKU号的MD5哈希值，并对该哈希值建立索引［24］方便查询。同时因为Token长度固定，及实际分析可以只给Token建立部分索引，不仅可以有比较好的索引区分度，同时也可以减少索引的大小。

对于个人每天限购根全体限购没有本质的区别，仅仅是增加了一个日期信息和一个用户信息来计算Token,因此每天的口令是不同的。

限购的流程就是获取促销活动信息、商品信息、日起信息、用户信息计算出查询口令，然后通过查询中间表来更新中间表和操作实际的订单表。每次更新数据只有最新的版本号可以更新。

* + 1. 促销系统热度数据

促销活动总类繁多，加上优惠券，为了最大效率的提高促销系统下单转化率，如果在用户下单之前或者下单之后提示用户感兴趣的促销活动可能将提高用户下单量。如何推荐所有活动中的部分活动，本项目中使用热度数据加推荐算法来智能推荐。促销活动热度数据统计不仅可以反映出用户对促销活动的兴趣分布；同时可以为运营提供最优的促销策略。促销活动热度统计主要使用Hadoop分布式计算平台。



图4.5 Hadoop作业流程图

统计的维度主要有用户关注列表商品，用户浏览历史记录商品，用户购物车购买商品，通过统计不同用户的日常购买行为分析出用户对促销活动的一个相关关系热度排序图。

热度数据首先收集所有用户的购买行为数据，如果用户在看到促销活动后点击了促销活动信息则表示用户对该促销活动感兴趣，如果用户在点击了某个促销活动的条件后添加相应的商品到购物车则表示用户对该促销商品满意，其他情况标记为普通。所有用户关注的商品标记为一个感兴趣。

热度数据的统计过程：首先收集用户购买日志数据。通过收集的日志数据利用Hadoop先通过map输出每个商品购买的时间间隔，形如（(item1,start,end）, (item2,start,end), …,(itemN, start, end))的数据格式，以itemX为key, (start, end)键值对。然后利用reduce统计所有的商品平均消耗时间和每个用户平均消耗商品时间，保存到中间表。然后把一个用户每天购买的商品归一化处理，归一化的原则为一天内多次购买只计算一次，防止热度数据过于局部化，之后结合商品时间消耗，商品我的关注列表加权求和，最后对加权求和的热度进行排序，只选取全局热度的前五个热度商品（可用于解决推荐算法冷启动）和个人相关的前十热度商品（个人数据相对稀疏）。

热度数据每天更新一次，因用户购物的频率一般也为每天一次。系统更新热度数据的时机选择在每天的凌晨。总体的计算策略为：每天在凌晨定时把所有的日志数据抓取出来，计算商品消耗时间，用户关注商品，把日志数据随机分为训练数据和测试数据，分割比例为4比1。然后不断的迭代计算权值最优解，把训练好的算法，利用历史数据做评测，但凡预测的商品用户购买了加一分。选取训练成绩最好的算法。然后再全量的统计分析所有的因素预测出当天最优的热度数据。



图4.6 热度数据生产过程

产生的热度数据不仅可供运营人员发布新的促销活动，也可以供推荐算法智能推荐，详见下小节。

* + 1. 促销系统智能推荐

传统的协同过滤算法不能自适应反应用户的具体兴趣变化。比如用户在购买了某件商品之后更大的概率是在该商品快要用完的时候才有兴趣去购买。本项目采用的基协同过滤算法，引入了动态的热度数据做算法数据输入矩阵。所有商品随着时间的推移商品的权重足部提升，当购买行为距离上次购买相差一个商品平均消耗时间时权值最大。

基于物品的协同过滤算法，根据其假设对相同商品有近似喜爱的情感的那些用户是目标用户的相似兴趣爱好用户，通过相似兴趣爱好用户对某商品的喜爱程度，可以预测目标用户对某商品的喜爱程度。基于这个思想结合促销系统具体的算法流程可以分为5步：

1. 数据表述

数据的理论模型是－个m\*n的用户－商品反馈矩阵M。如图4.7所示，m表示用户数量，n表示商品数量，矩阵中Mmn表示第m个用户对第n个商品的喜爱分数。

图4.7 用户－商品关系矩阵

这个喜爱分数是经过分析用户的注册数据和行为数据加权求出的。其喜爱分数设计如表4.10。

表 4.10 商品便好分值设计

|  |  |
| --- | --- |
| 描述 | 分值 |
| 用户买过 | 满意 |
| 用户收藏 | 感兴趣 |
| 其他 | 普通 |

1. 相似用户搜索

给出了数据商品偏好程度，这一步就是要获取与目标商品最相似的商品集合。计算的方法常用的有余弦相似度或者其他的比如皮尔逊相关系数等 。

商品的评分在输入矩阵中为输入矩阵的一个列向量，如果使用余弦相似度来计算两个商品的相似度可以使用两个向量的之间的余弦值来代替。

其中ix,iy表示两个具体的商品，sim(ix,iy)表示两个商品的相似度也就是商品向量的余弦值。理解向量的余弦值就很容易知道Uxy表示所有评价过商品ix和iy，Ux表示评价过ix，其它商品表示同理。Mix和Miy分别表示Ui对商品ix和iy的评分。

1. 得出目标用户推荐结果

之前通过计算商品之间相似度，可以得出目标用户有相似兴趣的用户的所有评分，然后要预测目标用户对某商品的喜爱，可以使用近邻用户对商品的评价来评估目标用户对商品的评价。

把所有近邻用户对该商品iy的评价作为目标用户的预测，翻译成数学语意就是所有的相似兴趣爱好用户对该商品的喜爱程度求和，然后目标用户对这个商品也有一个主管的便好，相乘就是整体的偏好。同时因为向量相乘可能为负，使用近邻用户对目标商品的偏好之和调和下实际目标用户对该商品的喜好。公式变量说明：simy,i表示商品y和商品i之间的对于近邻用户集合喜爱的相似度。Pay为目标用户对该商品的喜爱程度。rai 表示目标用户对商品的偏好程度。

然后对于计算结果按照从大到小的顺序，推荐出最合适的商品列表。

1. 获取推荐解释

根据之前介绍，近邻用户对某商品的偏好，是推荐获得该推荐的原因。而你之所以和某部分用户是近邻用户是因为你们对相同的一部分商品有类似的行为偏好。那么就可以把这相似的行为偏好语意化。比如因为你对xxx感兴趣（而这些兴趣是浏览，购买，收藏）。所以可以解释为：猜你喜欢XXX商品？因为你曾经收藏。

1. 结合促销系统

获得了推荐的商品，并且推荐的商品上都具有推荐解释信息。所有这些带有推荐解释的商品都是通过Hadoop平台提前计算好，然后保存到了数据库中。在促销显示的时候操作过程是直接查询MySQL数据库表。依次检查，然后把概率最高的相关促销信息排在最前面并显示给顾客。

整体这5个步骤如图4.8所示。



图4.8 促销系统推荐过程

正如前文所述，在推荐系统中需要大量计算的就两个地方。一个是利用余弦相似度来计算商品的相似度；另外一个则是目标用户对某商品的预测评分阶段。下文将就这两个算法在Hadoop平台的应用具体展开论述：

1. 商品间相似度的计算

输入：商品集合I,用户集合U,用户商品评分矩阵Rui。

输出：商品间的相似度矩阵sim(i，i)。

1. 初始化i=1;
2. 如果 i >= |I|,结束；
3. i++, 若 i >= |I| 结束；
4. 从评分矩阵R中找到表示过商品ix，iy,然后分别找出对应的评分；
5. 利用余弦相似度计算商品间相似度sim(x,y)；
6. i++, 跳转到c。
7. 预测评分

输入: 目标用户Ua、目标商品 Iy、选取的最近邻数量、评分矩阵 RU×I、项目间相似度矩阵 sim(I×I)。

输出: 用户 Ua对项目 Iy的预测评分。

1. 从项目相似度矩阵中选择K个与项目iy相似度最高的项目组成最近邻集合 Ny;
2. 根据公式(2)计算用户 Ua对项目Iy的预测评分 Pay;



图4.9 算法在Hadoop平台中执行流程图

根据之前算法的描述步骤，在Hadoop平台中拆分成三个串行的MapReduce来执行。

第一步：获取用户－商品评价矩阵。原始数据直接输出到map阶段的程序，map输出中key为userId,值value为（productId, rating）这样的二元序列。Reduce阶段接收用户对每个商品的评价合并得到用户－商品评价矩阵。

第二步：计算所有商品之间的相似度。Map阶段接收用户－商品评价矩阵，提取每个用户下的商品评分，以(productId(x),productId(y))这样的二元组作为key,对应的评分二元组（rating(x), rating(y)）作为value, 输入到reduce阶段。然后由reduce阶段计算所有商品之间的相似度。

第三部：根据所有商品之间的相似度，取出每个商品与之最相似的K个商品标记为最相似。然后根据算法2计算出用户对某个商品的评分信息，并绑定推荐原因排序输出。

不管是热度统计分析，还是推荐算法都要依赖于数据。用户数据信息的采集主要包括平常使用行为的记录，包括用户购买商品，收藏商品，浏览商品，搜索商品等。其具体部署结构如图4.10。在该结构中，用户使用的时间越长，访问系统越多，记录用户的信息更全面。相应在反应顾客对商品的偏好程度也会更准确。



图4.10 整体框架示意图

* 1. 本章小结

本小节首先阐述了商品信息模块的相关设计与实现，然后分析了促销活动优惠信息记录的规则设计以及如何完美的潜入到订单系统做到售后优惠金额可追踪，之后研究了促销模块活动规则的创建设计，最后设计了热度数据和推荐系统来推广用户可能感兴趣的促销活动，热度数据算法能够每天动态的反应总体和个人的兴趣变化，推过这些数据结推荐算法能自适应的根据环境的变化而动态的智能推荐，使促销活动的效果最大化。

# 数据库设计

* 1. 数据模型

跟促销系统相关的表关系结构主要如图5.1所示。



图5.1 促销系统表结构关系图

* 1. 数据模型详细描述

整个促销系统设计的数据库表众多，主要的一些表结构下面一些表格所示：促销活动表，主要包含促销活动id,促销活动的广告语，促销活动的开始和结束时间，促销活动指定的用户范围，促销活动的规则属性等，主要的表字段信息如表5.2所示。

表5.2 促销活动表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度(字节) | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 促销活动ID |
| 2 | ads\_words | text | 0 | 是 | 广告语 |
| 3 | promotion\_type | tinyint | 1 | 否 | 促销类型 |
| 4 | rule\_ids | varchar | 0 | 否 | 活动规则id |
| 5 | source | int | 4 | 否 | 来源 |
| 6 | location\_id | int | 4 | 否 | 促销地区 |
| 7 | user\_ids | text | 0 | 是 | 可参与用户 |
| 8 | start\_time | Int | 4 | 否 | 开始时间 |
| 9 | end\_time | int | 4 | 否 | 结束时间 |
| 10 | last\_deliver\_time | int | 4 | 是 | 最晚配送时间 |
| 11 | pay\_type | tinyint | 1 | 否 | 支付类型 |
| 12 | status | tinyint | 1 | 否 | 活动状态 |

单品促销规则表主要保存的对某个单品进行促销活动的一些指定规则信息比如说对什么商品进行促销，促销的价格，促销商品对每个用户的限购信息，整体的限购信息等。

表5.3 单品促销活动规则表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 单品促销规则id |
| 2 | sku\_numbers | int | 4 | 否 | 指定促销商品 |
| 3 | limit | int | 4 | 是 | 每个用户限购数量 |
| 4 | total\_limit | int | 4 | 是 | 整体限购数量 |

满减促销活动规则表主要包含活动规则范围,同满赠活动规则一样。范围包含两个维度一个为指定SKU(商品最小单元),另外一种为指定一个分类。

同时包含指定的规则信息，规则信息支持阶梯活动，比如满100是怎样，满200怎样。同时和单品促销一样也用limit表示个人限购，total\_limit来用总体限购。

表5.4 满减促销活动规则表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 11 | 否 | 满减活动规则id |
| 2 | sku\_numbers | text | 0 | 是 | 商品范围，如果为空代表不限制 |
| 3 | category\_ids | text | 0 | 是 | 分类范围，空不限制 |
| 4 | rules | text | 0 | 否 | 活动规则 |
| 5 | limit | text | 0 | 是 | 个人限购 |
| 6 | total\_limit | text | 0 | 是 | 总体限购 |

满赠活动活动表与满减活动表基本类型，除了多了一个规则类型来区分满赠活动赠送的是商品还是优惠券。

表5.5 满赠促销活动规则表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 满赠活动ID |
| 2 | sku\_numbers | text | 0 | 是 | 活动商品范围 |
| 3 | category\_ids | text | 0 | 是 | 活动分类范围 |
| 4 | rules\_type | tinyint | 1 | 否 | 赠送类型 |
| 5 | rules | text | 0 | 否 | 活动规则 |
| 6 | limit | text | 0 | 是 | 个人限购 |
| 7 | total\_limit | text | 0 | 是 | 总体限购 |

订单表是任何一个电商的核心，设计的表字段和信息也是非常多的。因本项目只是设计到促销相关的信息，所以只列出和促销有相关的字段信息如表5.6所示。

表5.6 商品订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 订单ID |
| 2 | order\_number | char | 0 | 否 | 订单号 |
| 3 | username | char | 0 | 否 | 用户名 |
| 4 | user\_id | int | 4 | 否 | 用户ID |
| 5 | total\_price | int | 4 | 否 | 应收金额 |
| 6 | deal\_price | int | 4 | 否 | 实收金额 |
| 7 | city\_id | int | 4 | 否 | 城市ID |
| 8 | deliver\_time | int | 4 | 否 | 配送时间 |
| 9 | deliver\_date | int | 4 | 否 | 配送日期 |
| 10 | line\_id | int | 4 | 否 | 线路ID |
| 11 | minus\_amount | int | 4 | 否 | 减免金额 |
| 12 | pay\_type | int | 4 | 否 | 支付类型 |
| 13 | promotion\_ids | varchar | 0 | 是 | 订单促销id |

因业务需求问题，有些商品不能一起送达，当用户下单后，系统会根据相关规则把一个订单拆分可以配送的子弹。配送按照子单进行。一个母单对应多个多个子单。和促销系统相关的字段信息如表5.7所示。

表5.7 商品子订单列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 子单id |
| 2 | order\_number | char | 0 | 否 | 订单编号 |
| 3 | total\_price | int | 4 | 否 | 应收金额 |
| 4 | deal\_price | int | 4 | 否 | 实收金额 |
| 5 | location\_id | int | 4 | 否 | 位置ID |
| 6 | minus | Int | 4 | 否 | 减免 |
| 7 | order\_id | int | 4 | 否 | 订单ID |
| 8 | order\_resource | tinyint | 1 | 否 | 订单资源 |
| 9 | promotion\_ids | varchar | 0 | 是 | 促销ID |

除了记录订单和子订单的信息我们还是没有办处理相关售后服务每天限购和整体限购的逻辑，详情见表5.8。

表5.8 商品订单详情表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 订单详情id |
| 2 | order\_id | int | 4 | 否 | 订单ID |
| 3 | quantity | int | 4 | 否 | 质量 |
| 4 | Price | int | 4 | 否 | 价格 |
| 5 | sku\_number | int |  | 否 | SKU数量 |
| 6 | category\_id | int | 4 | 否 | 策略ID |
| 7 | single\_price | int | 4 | 否 | 单价 |
| 8 | suborder\_id | int | 4 | 否 | 子订单ID |
| 9 | promotion\_ids | varchar | 0 | 是 | 促销ID |
| 10 | promotion\_quantity | int | 4 | 是 | 促销质量 |

限购逻辑实现依赖两张快照表，通过使用版本控制方法来实现限购数据的同步。总限购表限购的是所有的人，所以总限购使用促销活动和商品信息两个维度唯一确定总限购数量。个人用户每人每天限购，包括促销活动id, 用户id, 日期信息，sku商品信息。个人限购表和总限购表都使用这些维度的信息创建md5来保存到数据库中，同时对这个token建立索引来快速查询限购数量。两张表结构分别如下：

表5.9 促销商品总限购表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 总限购id |
| 2 | promotion\_id | int | 4 | 否 | 促销活动id |
| 3 | sku\_number | varchar | 0 | 否 | 商品编号 |
| 4 | quantity | int | 4 | 否 | 限购数量 |
| 5 | version | int | 4 | 否 | 版本号 |
| 6 | token | varchar | 0 | 否 | 查询口令 |
| 7 | created\_time | int | 4 | 否 | 创建时间 |
| 8 | updated\_time | int | 4 | 否 | 更新时间 |

表5.10 促销商品每人每天限购表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 长度（字节） | 可否为空 | 备注 |
| 1 | id | int | 4 | 否 | 个人限购id |
| 2 | promotion\_id | int | 4 | 否 | 促销活动id |
| 3 | user\_id | int | 4 | 否 | 用户id |
| 4 | date\_info | int | 4 | 否 | 日期信息 |
| 5 | sku\_number | varchar | 0 | 否 | 货物编号 |
| 6 | token | varchar | 0 | 否 | 查询口令 |
| 7 | version | Int | 4 | 否 | 版本编号 |

* 1. 本章小结

本章主要针对促销系统相关的数据库表设计。从数据库层面上展示了整个促销系统各个数据关系之间的相互作用，并且对相关重要的表结构做了详细的介绍。在下一章的系统实现中将详细的论述如何使用到这些表数据信息。

# 系统测试及效果

6. 1. 系统测试

本小结的系统测试用例主要包括促销活动的发布，已经基于促销系统发布的活动，用户在创建订单的过程包括订单逻辑的测试。对于测试用例有如下的说明，所有的测试对促销活动发布的信息在前面，用户实际购买的行为在后面，如（25元/20元）指的是前面25元为促销系统发布的价格，后面的20元为用户在购买的时候实际的价格。

* 1. 单品促销活动测试

单品促销是系统给指定商品的促销活动，主要的测试包括活动时间，购买数量，用户类型，活动地点等，详情见表6.1。

表6.1 单品促销测试用例示意图

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 促销商品 | 促销价格 | 个人限购 | 总限购 | 活动地点 | 支付类型 | 配送时间 | 预期结果 |
| 1 | 100001 | 25元/20元 |  |  |  |  |  | 减免0元 |
| 2 | 100001 | 12元/20元 |  |  |  |  |  | 减免8元 |
| 3 | 100001 | 12元/20元 | 3/1 |  |  |  |  | 减免8元 |
| 4 | 100001 | 12元/20元 | 3/5 |  |  |  |  | 减免0元 |
| 5 | 100001 | 12元/20元 |  | 3/1 |  |  |  | 减免8元 |
| 6 | 100001 | 12元/20元 |  | 3/10 |  |  |  | 减免0元 |
| 7 | 100000 | 12元／20元 |  |  | 北京／上海 |  |  | 减免0元 |
| 8 | 100000 | 12元／20元 |  |  |  | 微信／货到付款 |  | 减免0元 |
| 9 | 100000 | 12元／20元 |  |  |  |  | 2015.9.8  /2015.9.9 | 减免0元 |

* 1. 套装促销活动测试

表6.2 套装促销测试用例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 促销商品 | 促销价格 | 个人限购 | 总限购 | 活动地点 | 支付类型 | 配送时间 | 预期结果 |
| 1 | 100001，100002/10001，10003 | 150/180 |  |  |  |  |  | 减免0元 |
| 2 | 100001，100002/100001，100002 | 150/180 | 3/5 |  |  |  |  | 减免0元 |
| 3 | 100001，100002/100001，100002 | 150/180 |  | 3/5 |  |  |  | 减免0元 |
| 4 | 100001，100002/100001，100002 | 150/180 |  |  | 北京／上海 |  |  | 减免0元 |
| 5 | 100001，100002/100001，100002 | 150/180 |  |  |  | 微信／账期 |  | 减免0元 |
| 6 | 100001，100002/100001，100002 | 150/180 |  |  |  |  | 2015.9.1  ／2015.9.2 | 减免0元 |

* 1. 满减活动测试

表6.3 满减促销测试用例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 促销商品 | 促销价格 | 个人限购 | 总限购 | 活动地点 | 支付类型 | 配送时间 | 预期结果 |
| 1 | 蔬菜类/菠菜 | 100-10，200-20/150 |  |  |  |  |  | 减免10元 |
| 2 | 100001／100001 | 100-10，200-20/150 |  |  |  |  |  | 减免10元 |
| 3 | 100001／100001 | 100-10，200-20/250 |  |  |  |  |  | 减免20元 |
| 4 | 100001/100002 | 100-10，200-20/150 |  |  |  |  |  | 减免0元 |
| 5 | 100001 | 100-10，200-20/150 | 3/5 |  |  |  |  | 减免0元 |
| 6 | 100001 | 100-10，200-20/150 |  | 100/101 |  |  |  | 减免0元 |
| 7 | 100001 | 100-10，200-20/150 |  |  | 北京／上海 |  |  | 减免0元 |
| 8 | 100001 | 100-10，200-20/150 |  |  |  | 微信／账期 |  | 减免0元 |
| 9 | 100001 | 100-10，200-20/150 |  |  |  |  | 2015.9.1  ／2015.9.2 | 减免0元 |

* 1. 满赠活动测试

表6.4 满赠活动测试用例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 促销商品 | 促销价格 | 个人限购 | 总限购 | 活动地点 | 支付类型 | 配送时间 | 预期结果 |
| 1 | 蔬菜类/菠菜 | 100-100001，200-100002/150 |  |  |  |  |  | 赠送  100001商品 |
| 2 | 100001／100001 | 100-10，200-20/150 |  |  |  |  |  | 赠送  100001商品 |
| 3 | 100001 | 100-10，200-20/250 | 3/5 |  |  |  |  | 无优惠 |
| 4 | 100001 | 100-10，200-20/150 |  | 3/5 |  |  |  | 无优惠 |
| 5 | 100001 | 100-10，200-20/150 |  |  | 北京／上海 |  |  | 无优惠 |
| 8 | 100001 | 100-10，200-20/150 |  |  |  | 微信／账期 |  | 无优惠 |
| 9 | 100001 | 100-10，200-20/150 |  |  |  |  | 2015.9.1  ／2015.9.2 | 无优惠 |

* 1. 打折活动测试

表6.5 打折促销测试用例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 促销范围/购买商品 | 促销价格/购买价格 | 个人限购 | 总限购 | 活动地点 | 支付类型 | 配送时间 | 预期结果 |
| 1 | 蔬菜类/菠菜 | 100-9,200-8/120 |  |  |  |  |  | 优惠12元 |
| 2 | 100001／100001 | 100-9,200-8/120 |  |  |  |  |  | 优惠12元 |
| 3 | 100001／100001 | 100-9,200-8/120 | 3/5 |  |  |  |  | 无优惠 |
| 4 | 100001 | 100-9,200-8/120 |  | 3/5 |  |  |  | 无优惠 |
| 5 | 100001 | 100-9,200-8/120 |  |  | 北京／上海 |  |  | 无优惠 |
| 8 | 100001 | 100-9,200-8/120 |  |  |  | 微信／账期 |  | 无优惠 |
| 9 | 100001 | 100-9,200-8/120 |  |  |  |  | 2015.9.1  ／2015.9.2 | 无优惠 |

* 1. 均摊逻辑测试

表6.7 促销均摊逻辑测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 购物车 | 促销活动 | 预期结果 | | | |
| 现采子单 | 预采子单 | 总订单 | 订单详情 |
| 1 | 苹果（105元，现），菠菜（50元，现） | 水果100减10蔬菜50减5 | 优惠15元 | 优惠0元 | 优惠15元 | 苹果优惠10元，菠菜优惠5元 |
| 2 | 苹果（105元，现），菜籽油（200元，预） | 水果100减10，油满150减10元 | 优惠10元 | 优惠10元 | 优惠20元 | 苹果优惠10元，菜籽油优惠10元 |

* 1. 部分效果截图

因版权原因，只能截取部分成果图片，类似推荐效果截图设计到商业机密信息不再这里描述。图6.1为商品详情页面包括商品的详细规格信息和关注信息；图6.2 为指定促销商品对象范围；图6.3为指定促销活动方式对象逻辑运算效果示意图；图6.4为促销推荐信息显示页。



图6.1 商品详情页



图6.2 促销发布部分效果截图



图6.3 促销标签逻辑运算



图6.4 促销推荐效果

* 1. 本章小结

本章先后对系统的测试以及跟测试用例相关的结果，保证了系统的稳定性。通过测试结果，总结出需求很好的完成了促销系统的功能性需求和非功能性需求。

# 总结与展望

* 1. 总结

本论文分析了目前快速发展的互联网现状，对本项目的使用到的相关技术做了大致的介绍。并根据工程实际背景和项目需求，介绍了相关系统的设计。并根据某电商公司的需求设计和开发了这套促销系统。

本文通过概述系统设计的背景，分析各模块的相互依赖关系。然后重点分析促销系统在功能性需求和非功能性需求方面未来可能应对的挑战。使用了设计模式等相关敏捷开发的思想，不仅能够快速的拓展信息促销活动，发布新的促销规则。同时引入了促销推荐系统，能够尽可能的让用户发现可能感兴趣的促销活动。最后通过测试，很好了完成了系统需求。通过公司BI部门的统计，该促销活动上线以来，销售额不断攀升。提高了公司的业绩。

因时间和精力有限，本系统可能还有些不足：

1. 后台发布页面不是很美观。因为不是设计出身，很多操作界面比较生硬。
2. 系统在监测冲突那块进行了大量的查询操作，虽然这块时面对运营人员，但是还是可以优化相关数据设计，或者使用缓存技术来改进运营发布活动的体验。因时间关系以及重点放在普通用户上，以及开发周期短，这块后期可以改进下
3. 对于促销推荐系统，在现有的协同过滤算法上，通过统计分析商品的消耗时间，细化现有的时间维度。同时因为地域维度，可能会得到更好的推荐效果。提高促销系统的使用效率。

因此在接下的时间里面主要的工作是：因为页面美工人员美化操作窗口。通过DM标签接入用户的地域信息，通过统计用户购买没中商品的一个时间间隔，来统计一般商品的消耗时间。因为不同的用户消耗的周期不一样。所以这个统计维度主要是对同一个用户的平均值统计。然后再引入这两个变量到推荐系统里面优化推荐效果。同时添加完善目前的日志系统，提高系统的稳定性和问题的快速定位及服务恢复。

* 1. 展望

电子商务因借助现代互联网技术不仅成本低，效率高，发布消息快。必将代表未来贸易的发展方向。同时，我们正处在一个信息爆炸的社会，换言之就每个个体我们都处在信息过载的时代，传统的促销方式已经没有能力去解决这一个问题。利用互联网技术，利用时下的大数据技术。我们就能分析出用户需要什么，用户对什么感兴趣。然后智能的推荐相关商品给他。同时因现在互联网一个站点往往功能众多。有些需要的功能隐藏很深，不需要的功能或者商品又放在一个明显的位置，浪费空间。未来甚至可以利用大数据技术深度挖掘用户行为爱好，为每个人生成一个深度定制化的电商网站。提供的服务类似与每位顾客的私人助理。同时相对而言，大数据的思维在信息技术的今天实现也相对较为容易，为生活在这样的技术时代而兴奋。相信随着规模的逐步扩大这样的设想，为期不久。

# 参考文献

［1］林丹．基于LAMP的电子商务平台设计与实现，2007年12月.

［2］https://en.wikipedia.org/wiki/Rasmus\_Lerdorf.

［3］<http://php.net/manual/zh/intro-whatcando.php>.

［4］MySQL官方网站，[www.mysql.com](http://www.mysql.com).

［5］王峰，雷葆华.Hadoop分布式文件系统的模型分析.

［6］刘文龙，张桂芸.基于加权信息熵相似度的协同过滤算法研究.

［7］杨志文，刘波.基于Hadoop平台协同过滤推荐算法.

［8］范波，程久军.用户间相似度协同过滤推荐算法.计算机科学，2012(1).

［9］Resnick P，Iakovou N，Sushak M，etal.GroupLens：An open Architecture for Collaborative Filtering of Netnews[C]//Process of the 1994 Computer Supported Cooperative Work Conference，1994:175-190.

［10］Ungar L H，Foster D P.Clustering methods for collaborative filtering[C]//Process of the Workshop on Recommendation Systems. Menlo Park：AAAI Press.

［11］刑春晓，高凤荣，战思南，周立柱.适应用户兴趣变化的协同过滤推荐算法.

［12］张振亚，王进，程红梅，王煦法.基于余弦相似度的文本空间索引方法研究.2005,32(9).

［13］刘艳苏，桂秀梅.二八定律与长尾理论在现代图书馆的共生应用.2009,29(8).

［14］马锋，陈富明.长尾理论与微内容开发：Web2.0时代媒体网站发发展的依据和路径.2007,06.

［15］Hadoop官网文档.http://Hadoop.apache.org/.

［16］田莹颖.基于社会化标签系统的个性化信息推荐探讨.2010,(01).

［17］唐中元，王煦法，孟宪海.大型信息处理系统中软件并发性策略的设计.计算机工程.2001,09.

［18］任午[令](http://search.cnki.com.cn/Search.aspx?q=author:%E4%BB%BB%E5%8D%88%E4%BB%A4).Btrieve中数据的完整性和并发性控制.1995.

［19］沈斌，李兴国，钟金宏，沈丽娜.基于多队列和多线程的短信实时并发控制算法.2008,34(9).

［20］毛目容，李继东，詹永照.实时分布式协同工作环境的研究与开发.2002,(4).

［21］周建强，谢立，孙钟秀，朱根江.K锁算法——一种数据库上的可适应并发控制策略.计算机学报.1991,12.

［22］汤伟，程家兴，顾婷婷，胡必云.基于Hibernate锁机制的事物管理方案研究与实现.2007,28(18).

［23］刘子微.社交网站广告反作弊系统的实现和优化.2015.

［24］张士军，陆海伦.索引在MySql查询优化中的应用.2007.

# 致谢

转眼间在北大已经学习生活了快三年了。在这三年里对我的一生意义深远。首先我要感谢我的导师吴中海教授和张齐勋老师对我的教导。在研究生学习阶段不仅教育我们高深的专业知识，同时教育我们做人之道。

本论文是在吴中海、张齐勋老师的悉心指导下完成的。张齐勋老师作为一名优秀的、经验丰富的教师，具有丰富的专业知识和项目经验，在整个论文实验和论文写作过程中，对我进行了耐心的指导和帮助，提出严格要求，引导我不断开阔思路，为我答疑解惑，鼓励我大胆创新，使我在这一段宝贵的时光中，既增长了知识、开阔了视野、锻炼了心态，又培养了良好的实验习惯和科研精神。在此，我向我的指导老师表示最诚挚的谢意！

# 北京大学学位论文原创性声明和使用授权说明

**原创性声明**

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名： 日期： 年 月 日

**学位论文使用授权说明**

（必须装订在提交学校图书馆的印刷本）

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

* 按照学校要求提交学位论文的印刷本和电子版本；
* 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
* 学校可以采用影印、缩印、数字化或其它复制手段保存论文；
* 因某种特殊原因需要延迟发布学位论文电子版，授权学校□一年/□两年/□三年以后，在校园网上全文发布。

（保密论文在解密后遵守此规定）

论文作者签名： 导师签名：

日期： 年 月 日