
淘宝系统
详细设计说明书

初稿

温浩珉
2017 年 11 月

目录

1 系统概述.....	3
1.1 系统架构设计.....	3
1.2 系统功能概述.....	3
2 系统功能模块结构.....	4
2.1 前端功能.....	4
2.2 WEB 服务端.....	4
2.3 数据库端.....	5
3 系统界面设计.....	5
4 系统数据模型设计.....	8
4.1 表设计.....	8
4.1.1 表间关系.....	8
4.2.2 表&索引设计.....	8
4.2 视图设计.....	12
4.3 存储过程.....	13
4.4 触发器.....	14
4.5 并发事务.....	14
4.5.1 前端网页的并发访问.....	14
4.5.3 非加锁情况下的并发试错.....	16
5 系统安全体系设计.....	17
5.1 用户管理与控制.....	17
5.2 存储与恢复.....	17
6 系统运行环境设计与部署结构.....	17
6.1 系统硬件技术规划.....	17
6.2 系统主要设备软件技术规划.....	18
7 源代码列表及说明.....	19

1 系统概述

1.1 系统架构设计

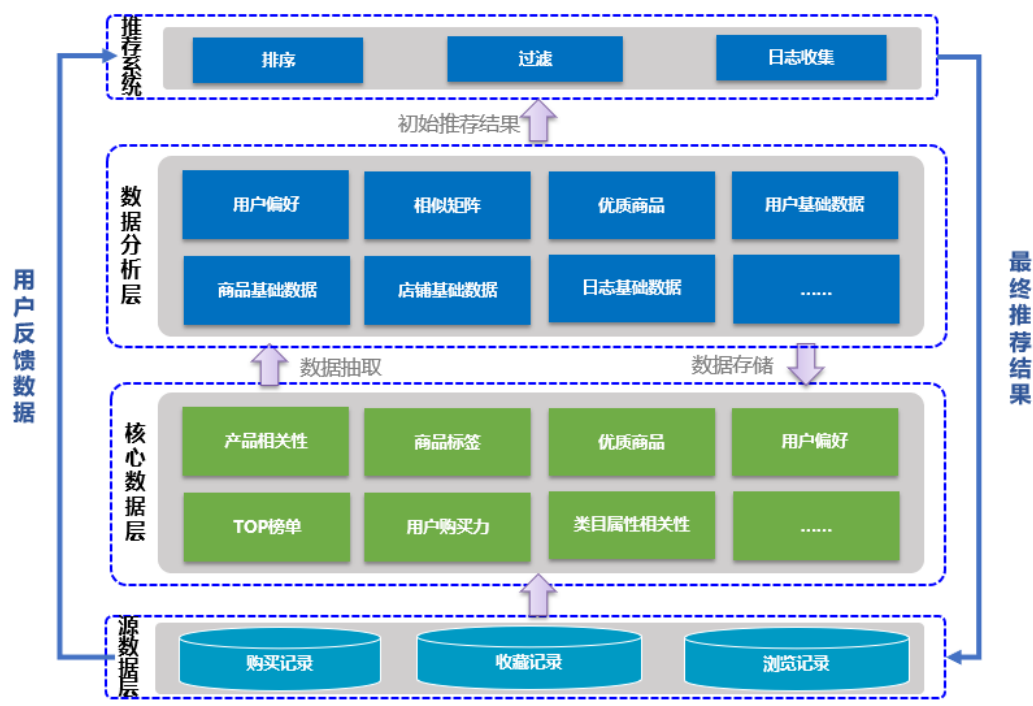


图 1.2 系统架构设计

整个数据库系统的架构如上图，在数据源层，用户产生购买、收藏、浏览等数据，送入核心数据层。核心数据层储存产品相关性，商品标签用户偏好等核心数据。抽取核心数据层的数据，在数据分析层进行进用户偏好、优质商品等分析，将分析结果存回核心数据层，并将初步的推荐结果送给推荐系统。推荐系统进行排序过滤等操作后将最终的推荐结果推给用户。之后用户为推荐系统提供反馈。

以上便是整个系统的架构设计，但是由于时间原因，只挑选了其中的一部分实现。具体见下。

1.2 系统功能概述

- 淘宝网业务系统的功能性需求主要包括以下内容：
- （1）用户在企业网站中选择需要的产品放入购物车，进行订购。
 - （2）用户输入购买细节，提交订单，系统自动将订单信息保存到数据库。
 - （3）用户能够和店家联系，进一步了解产品，协商价格，确认订单细节。
 - （4）店家在收到用户的联系要求后，及时与客户联系，为客户提供咨询服务，确认订单的细节，如需要修改原订单，将修改后的订单信息更新到数据库并向客户发出订单确认信

息。

- (5) 用户收到订单确认信息后，网上支付产品款项给店家。
- (6) 店家根据发货信息发货，修改订单状态。
- (7) 店家能够修改店铺商品信息

2 系统功能模块结构

2.1 前端功能

前端使用模板语言，主要实现界面显示的功能。主要界面设计和界面跳转逻辑如下图：

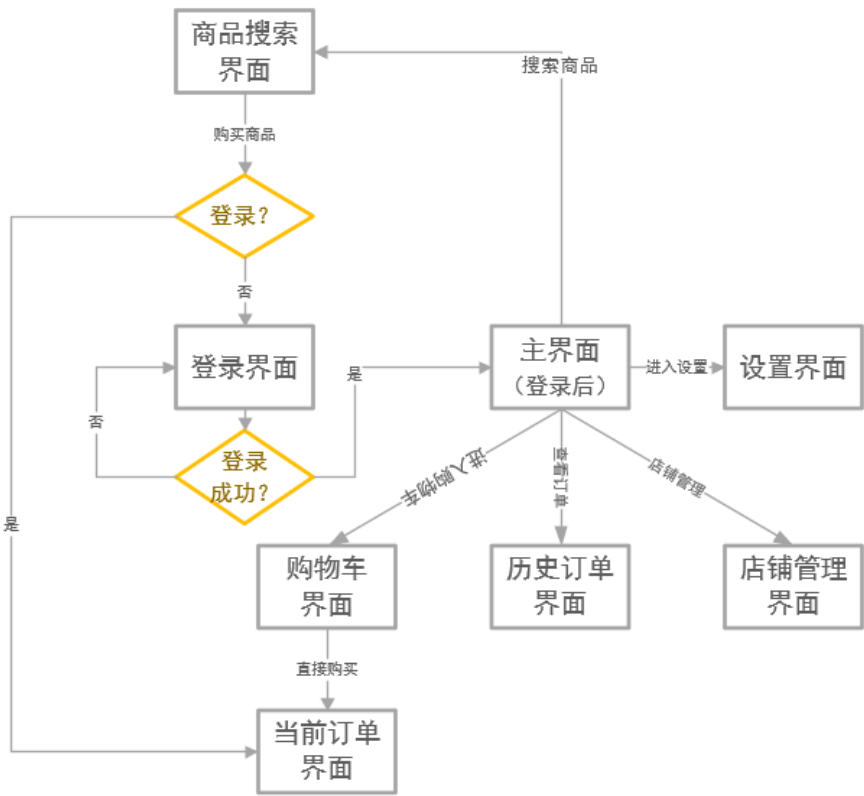


图 2.1 前端设计

2.2 WEB 服务端

web 服务端：监听并响应客户端的请求，为客户端提供相应的服务。服务端主要提供以下服务：

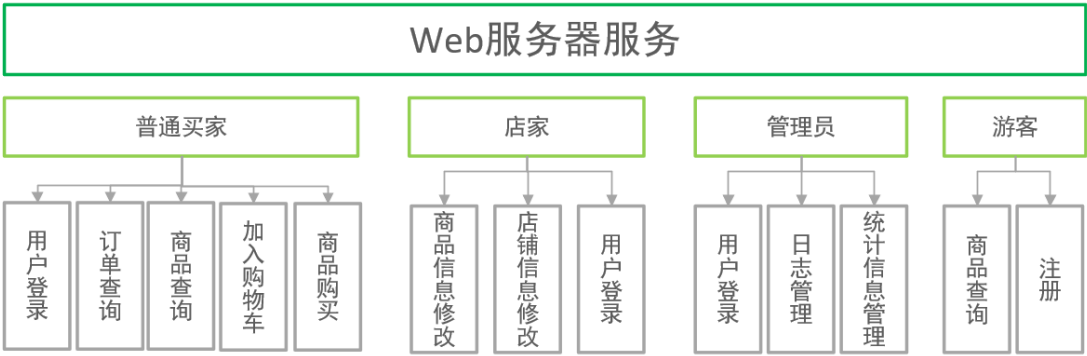


图 2.2 web 服务端服务

2.3 数据库端

数据库端用于储存系统核心数据。包括店铺信息、用户信息、商品信息、购物车信息、订单信息、店铺-商品关系、购物车-商品信息、订单-商品信息等。

3 系统界面设计

系统几个主要界面设计如下



图 3.1 游客商品搜索界面

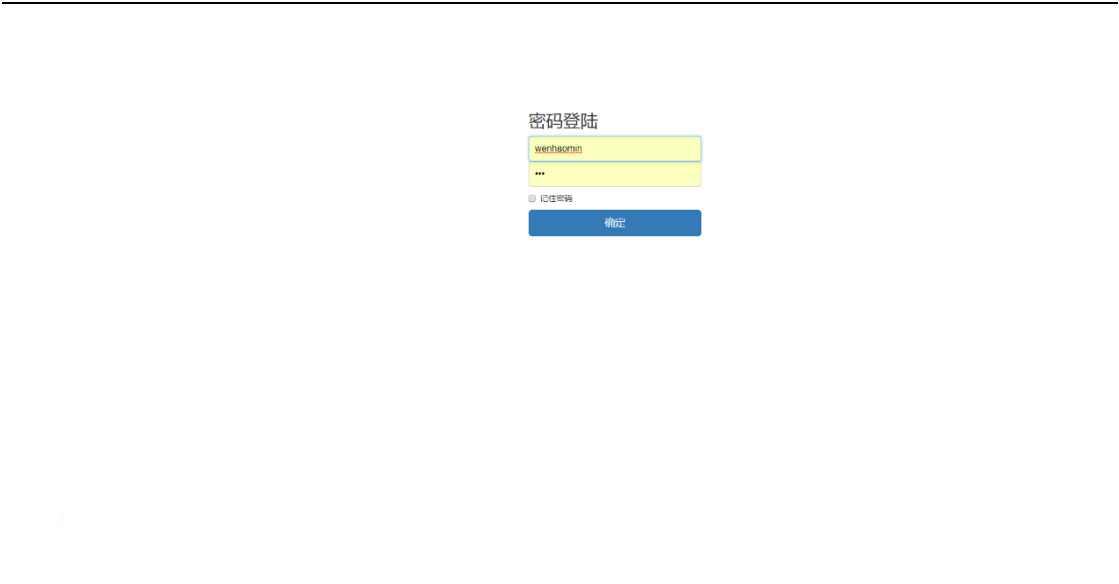


图 3.2 登录界面



图 3.3 登录后主界面

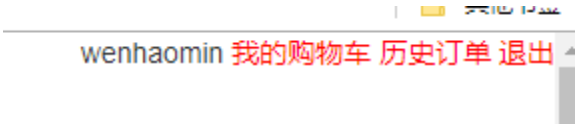


图 3.4 任务栏界面

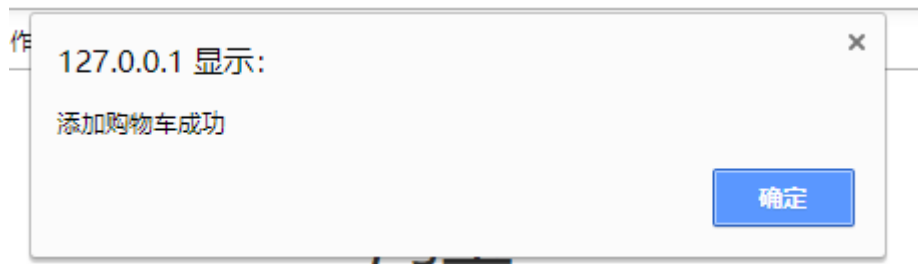


图 3.5 成功添加购物车提示框



图 3.7 历史订单界面

4 系统数据模型设计

4.1 表设计

4.1.1 表间关系

模型整体数据模型设计和表级之间的关系设计如下：

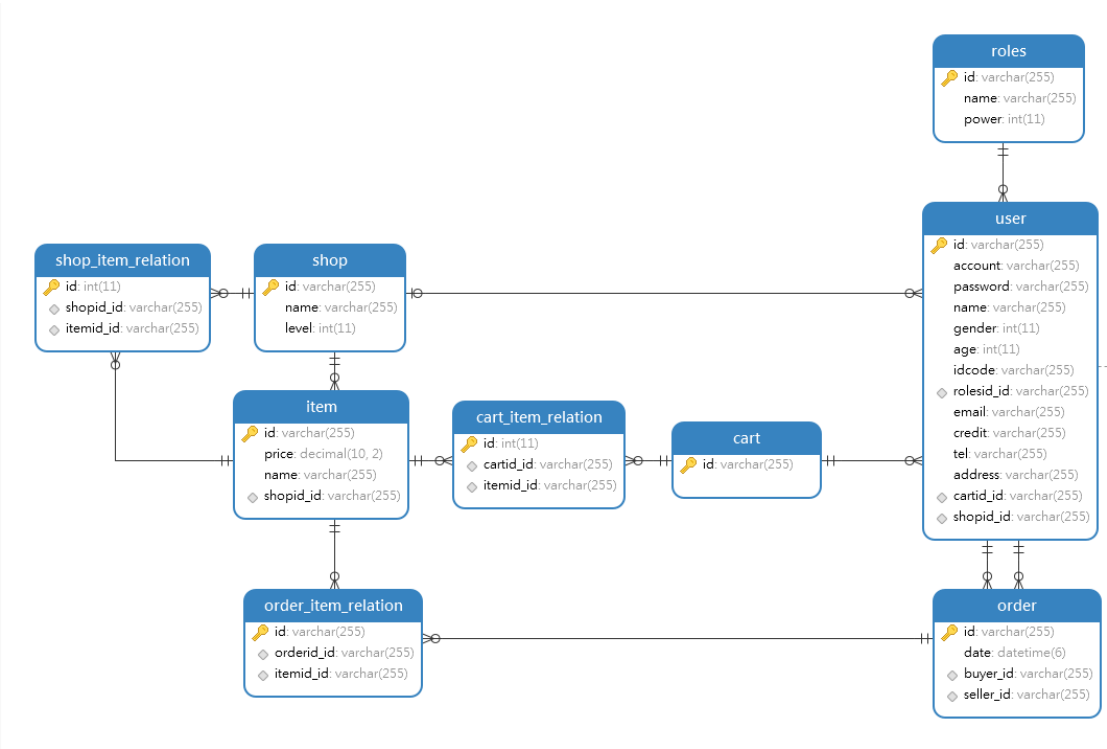


图 4.1 表关系

4.2.2 表&索引设计

表 4.2.2.1 order_item_relation

表名	order_item_relation				
数据库用户	root				
主键	(OrderID, ItemID)				
外键	OrderID, ItemID				
排序字段	OrderID				
索引字段					
字段名称	数据类型	允许为空	唯一	默认值	约束条件

OrderID	varchar(255)	N			外键
ItemID	varchar(255)	N			外键

表 4.2.2.2 cart_item_relation

表名	cart_item_relation				
数据库用户	root				
主键	(CartID,ItemID)				
外键	CartID,ItemID				
排序字段	CartID				
索引字段					
字段名称	数据类型	允许为空	唯一	默认值	约束条件
CartID	varchar(255)	N			外键
ItemID	varchar(255)	N			外键

表 4.2.2.3 item

表名	item				
数据库用户	root				
主键	ID				
外键	ShopID				
排序字段	ID				
索引字段	ID				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	varchar(255)	N			主键
Price	decimal(10,2)	N			
ShopID	varchar(255)	N			外键
Name	varchar(255)	N			

表 4.2.2.3 shop

表名	shop
----	------

数据库用户		root			
主键		ID			
外键					
排序字段		ID			
索引字段		ID			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	varchar(255)	N			主键
Level	int(11)	N			
Name	varchar(255)	N			

表 4.2.2.4 shop_item_relation

表名		shop_item_relation			
数据库用户		root			
主键		(ShopID,ItemID)			
外键		ShopID,ItemID			
排序字段		ShopID			
索引字段					
字段名称	数据类型	允许为空	唯一	默认值	约束条件
ShopID	varchar(255)	N			外键
ItemID	varchar(255)	N			外键

表 4.2.2.5 roles

表名		roles			
数据库用户		root			
主键		ID			
外键					
排序字段		ID			
索引字段		ID			
字段名称	数据类型	允许为空	唯一	默认值	约束条件

ID	varchar(255)	N			主键
Power	int(11)	N			
Name	varchar(255)	N			

表 4.2.2.6 user

表名		user			
数据库用户		root			
主键		ID			
外键		RolesID, CartID, ShopID			
排序字段		ID			
索引字段		ID			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	varchar(255)	N			主键
Account	varchar(255)	N			
Email	varchar(255)	N			
Address	varchar(255)	N			
Gender	tinyint(4)	N			
Name	varchar(255)	N			
RolesID	varchar(255)	N			外键
Credit	varchar(255)	N			
IDCode	varchar(255)	N			
Tel	varchar(255)	N			
CartID	varchar(255)	N			外键
Age	int(11)	N			
ShopID	varchar(255)	Y			外键
Password	varchar(255)	N			

表 4.2.2.7 cart

表名	cart
----	------

数据库用户		root			
主键		ID			
外键					
排序字段		ID			
索引字段		ID			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	varchar(255)				主键
ItemsID	varchar(255)				

表 4.2.2.8 order

表名		order			
数据库用户		root			
主键		ID			
外键		Buyer,Seller			
排序字段		ID			
索引字段		ID			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	varchar(255)	N			主键
Buyer	varchar(255)	N			外键
Seller	varchar(255)	N			外键
Date	datetime	N			

4.2 视图设计

(1) 普通用户视图

对于普通用户，其可见的表如下：

表 4.2.1 用户视图

关系模式	表名
user	用户表
cart	购物车表

item	商品表
order	订单表
cart_item_relation	购物车-商品关系表
order_item_relation	订单-商品关系表

(2) 商家视图

表 4.2.2 商家视图

关系模式	表名
user	用户表
cart	购物车表
item	商品表
order	订单表
cart_item_relation	购物车-商品关系表
order_item_relation	订单-商品关系表
shop	店铺表
shop_item_relation	店铺-商品关系表

(3) 系统管理员视图

表 4.2.3 系统管理员视图

关系模式	表名
user	用户表
cart	购物车表
item	商品表
order	订单表
cart_item_relation	购物车-商品关系表
order_item_relation	订单-商品关系表
shop	店铺表
shop_item_relation	店铺-商品关系表
role	角色表

4.3 存储过程

存储过程是经过编译、优化的查询语句序列，因而使用存储过程能加快查询效率。此外存储过程还可以实现复杂的完整性控制。因而，以上两个优点，在系统中，主要设计以下存

储过程：

(1) 商品查询存储过程

以商品的名字、店铺的名字为参数，查询结果返回满足条件的商品。

(2) 购买存储过程

以商品 id、用户 id 作为参数，在存储过程中生成订单信息，将 shop-item-relation 中的商品数量减一。

(3) 加入购物车储存过程

以商品 id、用户 id 为参数，将用户对应的购物车中的对应商品数量加一。

(4) 查看历史订单存储过程

以用户 id 为参数，查询该用户历史订单，并返回结果。

(5) 查看购物车储存过程

以用户 id 为参数，查看该用户当前购物车中所有未购买商品。

4.4 触发器

在系统中，主要使用后触发器的数据检查功能：

(1) 权限检查

当用户对 shop、shop_item_relation 表执行修改操作时，检查当前操作用户的 role id，判断是否拥有对 shop 等表的操作权限，如果没有则拒绝操作。（如非店铺用户不能够修改店铺和商品信息）

(2) 数据检查

向 user 表写记录时，检查其身份证字段是否合法，非法则拒绝执行操作。

4.5 并发事务

4.5.1 前端网页的并发访问

将写好的数据库系统搭建在服务器上，运行。测试并发访问，我使用了 GITHUB 上的工具 webbench，该工具可以对网页进行并发测试。具体的功能描述可见网页：

<https://github.com/EZLippi/WebBench>

测试设计如下：

实现一个网页，当用户访问该网页时，系统自动随机生成一条数据，并且插入数据库的表中，在前端网页显示数据库表中前 10 条记录。

使用 webbench 访问该网页，测试该数据库系统的并行能力。

测试过程如下：

← 不安全 | 162.219.123.220/add

应用 百度一下, 你就知道 bjtu 科研 学习 大创 语言 NOI Google 翻译 游览

编号	用户名	密码
1298	yz0k	IRH3O
1299	gj6l	HE4MA
1300	FPNr	s143q

用户: 密码:

编号:

编号: 用户: 密码:

```
Database changed
mysql> select * from taobaoapp_meat;
+-----+-----+-----+
| id    | name  | password |
+-----+-----+-----+
| 1298  | yz0k  | IRH3O    |
| 1299  | gj6I  | HE4MA    |
| 1300  | FPNr  | s143q    |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

测试前的数据库表内容

```
root@ubuntu:~# webbench http://162.219.123.220/add -t 30 -c 1000 > re.txt
```

输入的 webbench 命令

其中-t 代表运行时间（秒），-c 代表线程池中有多少的线程同时访问。

```
root@ubuntu:~# webbench http://162.219.123.220/add -t 30 -c 1000
Webbench - Simple Web Benchmark 1.5
Copyright (c) Radim Kolar 1997-2004, GPL Open Source Software.

Request:
GET /add HTTP/1.0
User-Agent: WebBench 1.5
Host: 162.219.123.220

Runing info: 1000 clients, running 30 sec.

Speed=8286 pages/min, 446352 bytes/sec.
Requests: 4138 succeed, 5 failed.
root@ubuntu:~#
```

分析：可以看到在 30 秒内，共有 4318 个访问请求发出，有 5 个失败。

```
9543 | c906 | jbkDO
9544 | csKB | Do8yB
9545 | Ry9A | jX40g
9546 | hA2g | 7INo4
9547 | mSh9 | tGiSV
9548 | 17Xg | KjfkY
9549 | qhE1 | 4vwgM
9550 | zk8Y | 1sg36
9551 | CX5k | nRCUz
9552 | P176 | 6cXUY
9553 | u4SP | kp40b
9554 | i7Rq | Y57R0
9555 | rHiR | 3k6dA
9556 | jdu3 | oHZ52
9557 | V7iC | YCBF1
9558 | BA9t | RNJVf
+-----+-----+
8261 rows in set (0.00 sec)
```

测试后数据库中表内容

分析：在 30 秒内多个客户端同时访问网站，每次访问都想数据库中插入一条记录。从以上的失败率来看，4318 个访问中，有 5 个失败，可见具有较为良好的并行处理能力。希望在之后得到进一步的提高。

4.5.3 非加锁情况下的并发试错

在数据库访问过程中，需要采用锁的机制，对一些正在处理的数据进行加锁。锁的种类有很多，如共享锁，排他锁等。锁的出现主要有两点原因：

一是为了保证数据库中数据的完整性和安全性；二是为了保证每个用户所得数据的正确性。如果不采用锁机制，将可能出现以下问题：丢失修改、脏读、不可重复读。

针对以上可能出现的问题，我设计了相关实验。详细代码可见附件《并行测试.py》

测试的主要思路如下，开两个线程，分别执行事务 A 和 B，A 中先改 id 为 2 的记录的 NAME 属性，B 中后 id 为 2 的 password 属性。

在不加锁的情况下，执行代码之后，B 的修改会覆盖 A，但是在实际情况下，我发现，mysql 的数据库引擎 INNODB 会自动将被操作的对象加上相应的锁。并且无法设置为不自动加锁。

5 系统安全体系设计

5.1 用户管理与控制

系统将用户分为四类：普通购物用户、店家、游客、管理员。在数据模型设计中，使用关系模式 `role` 描述一类用户。每一类的用户对应相应的 `role` 权限，这个权限体现在对数据库不同对象和在该对象上的操作集合的不同。在用户登录之后，获取对应用户的 `role id`。在 `web` 服务端，通过对该 `role id` 的判断，则可以判断该用户的身份，由此实现对不同的用户，呈现不同的视图。

5.2 存储于恢复

系统在运行过程中可能会出现故障，导致数据库中数据出现不一致的情况。因而在系统中，设计了定时备份的程序，每过一个设定的时间，则将数据库中的内容通过增量备份的方式备份到另外一个数据库中。

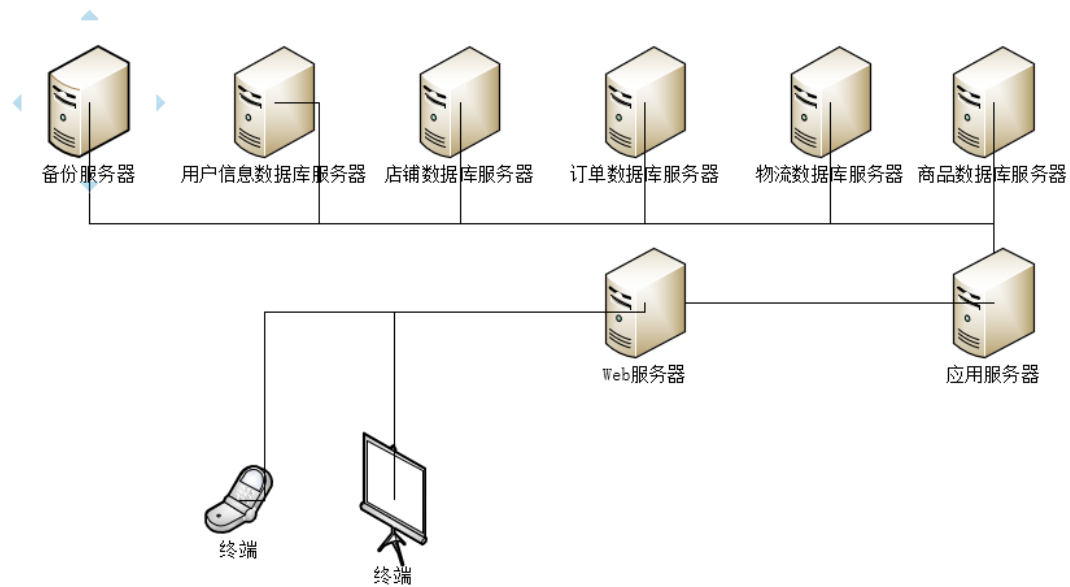
此外，在数据的备份中，我还尝试在一个表上，利用触发器进行备份。

恢复过程主要利用的是数据库管理系统的日志，由于数据恢复主要是数据库管理系统实现，故在此不做赘述。

6 系统运行环境设计与部署结构

6.1 系统硬件技术规划

(1) 系统运行环境逻辑规划



(2) 系统运行网络环境规划

整个系统架构在 4G 网络之上，终端通过网络访问 web 服务器和应用服务器上提供的各种服务。

同时，为保证核心服务器稳定运行和信息安全，系统应主要服务器应运行在专用网络，考虑到这些设备在空间上很近，此专网建设成本不高，具有可行性。

(3) 系统服务器规划

核心服务器应采用高性能的 PC 服务器，应采用单路 2 核或 4 核以上 CPU，内存 8G 以上，应配置 1T 以上存储空间，单机做 RAID5，双千兆网卡，建议采购主流品牌。

Web 服务器应采用高性能 PC 服务器，应采用单路 2 核以上 CPU，内存 4G 以上，应配置 300G 以上存储空间，单机做 RAID5，双千兆网卡，建议采购主流品牌。Web 服务器配备一台即可

系统应设置一台应用服务器，该服务器可以根据系统的负载情况，与 Web 服务或数据库共用一台服务器，或者采用一台独立的服务器。该服务器的性能应突出计算性能，应配置 4G 以上内存，300G 以上存储空间，双千兆网卡。

系统应至少设置一台备份服务器，该服务器应配备 1T 以上的存储空间，用于对各个服务器中的数据进行备份。

6.2 系统主要设备软件技术规划

(1) 核心数据库服务器和备份服务器软件环境规划

操作系统：

结合硬件平台采购，为保证系统的安全性和稳定性，建议采用 Linux、Solaris

或其它 Unix 操作系统。

数据库：

建议采用 Oracle 9i 或 10G

(2) Web 服务器软件环境规划

操作系统：

采用 Linux 或 Windows 2003 以 Server 版操作系统。

数据库：

在 Web 服务器可以不安装数据库。

Web 服务平台：

采用 Tomcat、IIS 等常见 Web 服务平台

(3) 应用服务器软环境规划

操作系统：

采用 Linux 或 Windows 2003 以 Server 版操作系统。

数据库：

可以根据需要安装 Oracle 或 SQL Server 数据库。

Web 服务平台：

采用 Tomcat、IIS 等常见 Web 服务平台

7 源代码列表及说明

文件名	说明
cart.html	购物车页面
index.html	主界面
login.html	登录界面
order.html	订单显示界面
models.py	生成数据库模型
views.py	各类视图控制
settings.py	数据库设置
urls.py	URL 设置
manange.py	系统运行主函数