目录

[第1章 前言 1](#_Toc21320)

[1.1 背景和意义 1](#_Toc16581)

[1.2 设计目标 1](#_Toc25377)

[第2章 数据库设计 2](#_Toc19652)

[2.1 需求分析 2](#_Toc10788)

[2.1.1 功能概述 2](#_Toc24030)

[2.1.2 用例图 3](#_Toc20238)

[2.1.3 用例描述 3](#_Toc18067)

[2.2 概念结构设计 6](#_Toc28214)

[2.2.1 实体列表 6](#_Toc15979)

[2.2.2 系统 E-R 模型 6](#_Toc25441)

[2.3 逻辑结构设计 8](#_Toc13038)

[2.4 关系模式规范化检查及处理 9](#_Toc19310)

[第 3 章 数据库定义与操作 12](#_Toc25899)

[3.1 数据库及数据表定义 12](#_Toc17453)

[3.2 数据查询操作 14](#_Toc24746)

[3.2.1 关系代数实现 14](#_Toc25627)

[3.3 数据增删改操作 15](#_Toc32446)

[3.4 索引及视图应用 16](#_Toc28895)

[3.4.1 索引的应用 16](#_Toc400)

[3.4.2 视图的应用 18](#_Toc21169)

[第 4 章 应用系统实现 20](#_Toc27740)

[4.1 登录页面 20](#_Toc29729)

[4.1.1 页面设计 20](#_Toc10932)

[4.1.2 功能实现 21](#_Toc30922)

[4.2 注册页面 21](#_Toc26311)

[4.2.1 页面设计 21](#_Toc19198)

[4.2.2 功能实现 22](#_Toc12034)

[4.3 产品管理页面 22](#_Toc16236)

[4.3.1 页面设计 22](#_Toc19838)

[4.4 订单管理页面 23](#_Toc18247)

[4.4.1 页面设计 23](#_Toc2074)

[4.5 购物页面 24](#_Toc11402)

[4.5.1 页面设计 24](#_Toc30260)

[4.5 库存统计页面 25](#_Toc28982)

[4.5.1 页面设计 25](#_Toc16734)

[第 5 章 总结 27](#_Toc29133)

# 

# 第1章 前言

### 1.1 背景和意义

随着互联网的普及，网络购物的优点更加突出，日益成为一种重要的购物形式。随着越来越多的人在网上购物，网购的人数数量也在逐年增加对于网站的服务器是一大挑战。有购物卡顿，付款缓慢等问题存在，因此希望设计一个系统，能够实现产品-订单-顾客三者能够比较好沟通协调的一个系统，优化购物过程，给予用户良好的购物体验。

### 1.2 设计目标

本系统设计初衷是希望能够解决用户购物卡顿，付款缓慢等问题。欲实现用户能便捷快速的选择自己喜爱的产品，预约方便空闲的时间去购物。购物平台后台也能够及时处理订单信息。为顾客选择收件地址，商品。后台管理员也可以统筹管理所有人的信息，查看操作记录等。

# 第2章 数据库设计

### 2.1 需求分析

需求分析的任务是通过详细调查现实世界要处理的对象，充分了解原系统的工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。新系统必须充分考虑今后可能的扩充和改变。

基于上述的要求，对本数据库的需求分析做一下几点的调查分析，功能概述部分，获得用户需要从数据库获取的信息，要求可以通过该系统存储或者查询某些数据库内的信息。而后绘制用例图，描写用例描述来获取不同元素、角色、用例之间的关系，获取需求，指导测试，在整个过程中指导其他的工作流。

## 2.1.1 功能概述

经分析，顾客购物订单管理系统应具有以下功能模块：

1. 顾客方面，功能要求如下:
2. 可以查看并修改个人信息。
3. 可以查看所有产品信息。
4. 可以查看并购买产品，包括选择数量。
5. 可以对订单信息进行管理，包括增删修。
6. 可以对收件信息进行维护，包括增删改查。

（2）管理员方面，功能要求如下：

① 可以查看并修改个人信息。

② 可以对产品信息进行维护，包括增删改查。

③ 可以对订单信息进行维护，包括增删改查。

主要功能模块图如下所示：

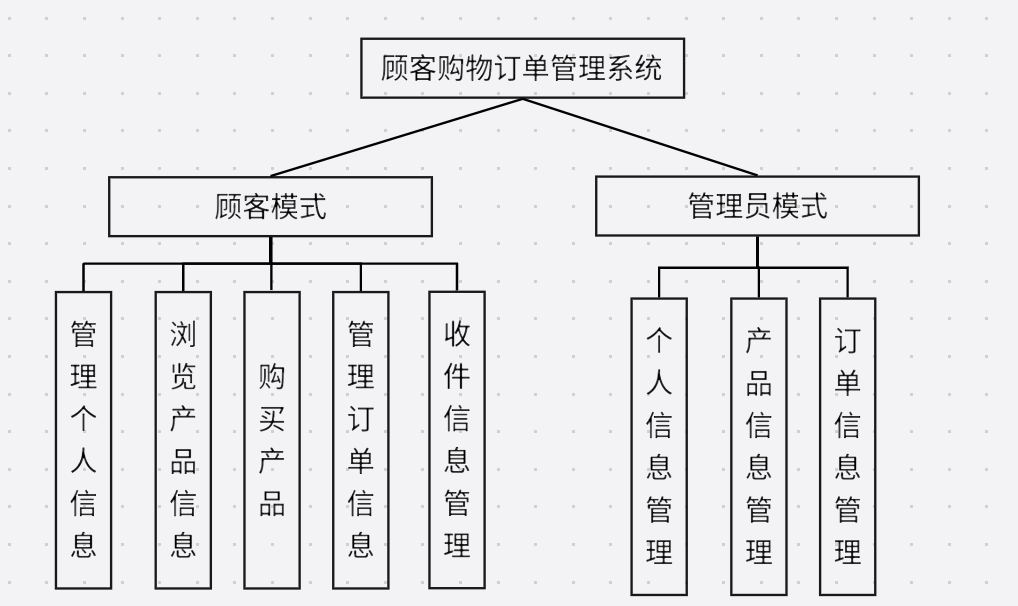


图 2.1 系统总功能模块图

## 2.1.2 用例图

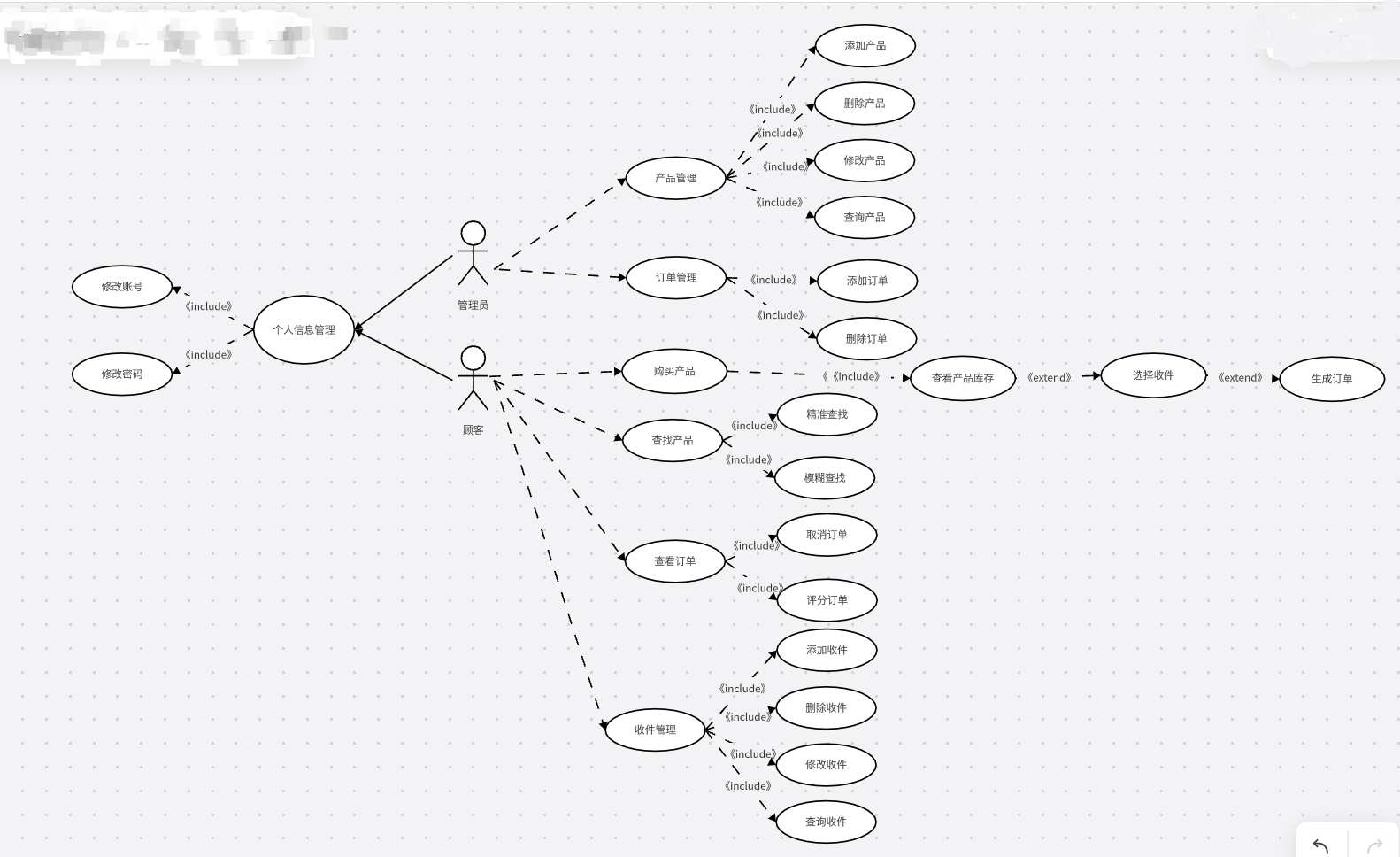


图 2. 2 用例图

## 2.1.3 用例描述

表 1 产品管理用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Al | 名称 | 产品管理 |
| 参与者 | 管理员，目标是能够管理产品 | | |
| 优先级 | 高 | | |
| 触发条件 | 管理员需要管理产品信息 | | |
| 前置条件 | 用户已登录，且身份为管理员 | | |
| 后置条件 | 无 | | |
| 正常流程 | 1. 用户输入产品信息，提交   2.系统判断数据库内是否存在名字相同的产品  3.系统授权增加  4.系统提示增加成功  5.系统向管理员表明增加产品成功  6.管理员输入产品名称，搜索  7.系统判断数据库内是否存在该名字的产品  8.管理员修改产品信息  9.系统判断管理员输入信息是否合法  10.系统授权修改  11.系统提示修改成功  12.管理员删除产品信息  13.系统授权删除  14.系统提示删除成功 | | |
| 扩展流程 | 2a:存在名字相同的产品  2a1：系统提示名字重复，并给出解决方案  3a：系统提示不存在该订单  3a1：系统提示不存在该订单 | | |
| 特殊流程 | 给出的提示信息应当足够清晰，提示管理员操作失败原因，并给出解决方案。 | | |

表 2 订单管理用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| ID | A2 | 名称 | 订单管理 |
| 参与者 | 管理员，目标是能够管理订单 | | |
| 优先级 | 高 | | |
| 触发条件 | 管理员需要管理订单信息 | | |
| 前置条件 | 用户已登录，且身份为管理员 | | |
| 后置条件 | 无 | | |
| 正常流程 | 1.用户输入订单名称，搜索  2.系统判断数据库内是否存在该编号的订单  3.系统输出订单信息  4.用户取消订单  5.系统授权删除  6.系统提示删除成功 | | |
| 扩展流程 | 2a:系统提示不存在该编号的订单  2a1：系统提示不存在该订单 | | |
| 特殊流程 | 给出的提示信息应当足够清晰，提示管理员操作失败原因，并给出解决方案。 | | |

表 3 购买产品用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| ID | A3 | 名称 | 购买产品 |
| 参与者 | 顾客，目标是能够购买产品 | | |
| 优先级 | 中 | | |
| 触发条件 | 顾客需要购买产品 | | |
| 前置条件 | 用户已登录，且身份为顾客 | | |
| 后置条件 | 无 | | |
| 正常流程 | 1.顾客输入产品信息，提交  2.系统判断数据库内是否存在名字相同的产品  3.系统提示产品查询成功  4.顾客输入产品数量，提交  5.系统判断数据库内的该产品是否库存的是否足够  6.系统提示产品库存充足  7.系统向管理员表明增加订单成功 | | |
| 扩展流程 | 2a:系统提示不存在该编号的订单  2a1：系统提示不存在该订单  3a：系统提示 不存在该医生  3a1：系统提示不存在该医生 | | |
| 特殊流程 | 给出的提示信息应当足够清晰，提示管理员操作失败原因，并给出解决方案。 | | |

表 4 订单评分用例流程图

|  |  |  |  |
| --- | --- | --- | --- |
| ID | A4 | 名称 | 评分订单 |
| 参与者 | 顾客，目标是能够评分订单 | | |
| 优先级 | 中 | | |
| 触发条件 | 顾客需要评分订单 | | |
| 前置条件 | 用户已登录，且身份为顾客 | | |
| 后置条件 | 无 | | |
| 正常流程 | 1.顾客输入订单名称，搜索  2.系统判断数据库内是否存在该编号的订单  3.系统提示输入订单评分  4.顾客输入订单评分  5.系统提示修改成功 | | |
| 扩展流程 | 2a:系统提示不存在该编号的订单  2a1：系统提示不存在该订单 | | |
| 特殊流程 | 给出的提示信息应当足够清晰，提示管理员操作失败原因，并给出解决方案。 | | |

表 5 收件管理用例流程图

|  |  |  |  |
| --- | --- | --- | --- |
| ID | A5 | 名称 | 收件管理 |
| 参与者 | 顾客，目标是能够收件管理 | | |
| 优先级 | 中 | | |
| 触发条件 | 顾客需要收件管理 | | |
| 前置条件 | 用户已登录，且身份为顾客 | | |
| 后置条件 | 无 | | |
| 正常流程 | 1.用户输入收件信息，提交  2.系统判断数据库内是否存在名字相同的收件  3.系统提示收件增加成功  4.用户修改收件  5.系统判断数据库内是否存在名字相同的收件  4.用户输入修改后的收件信息  6.系统提示收件修改成功  7.用户删除收件  8.系统判断数据库内是否存在名字相同的收件  9.系统提示收件删除成功  10.用户查询收件  11.系统判断数据库内是否存在名字相同的收件  12.系统输出收件信息 | | |
| 扩展流程 | 2a:系统提示不存在该编号的订单  2a1：系统提示不存在该订单 | | |
| 特殊流程 | 给出的提示信息应当足够清晰，提示管理员操作失败原因，并给出解决方案。 | | |

### 2.2 概念结构设计

## 2.2.1 实体列表

|  |  |
| --- | --- |
| 实体 | 描述 |
| 产品 | 所有的产品信息，由产品编号唯一标识 |
| 订单 | 所有的订单信息，由订单编号唯一标识 |
| 顾客 | 所有的顾客信息，由顾客编号唯一标识 |
| 收件 | 所有的收件信息，由收件编号唯一标识 |

## 2.2.2 系统 E-R 模型

顾客购物订单管理系统的 E-R 模型，如图 2.3~图 2.6 所示。其中，重要关系局部 E-R 图如图 2.3~图 2.5，整合的全局 E-R 图如图 2.6。

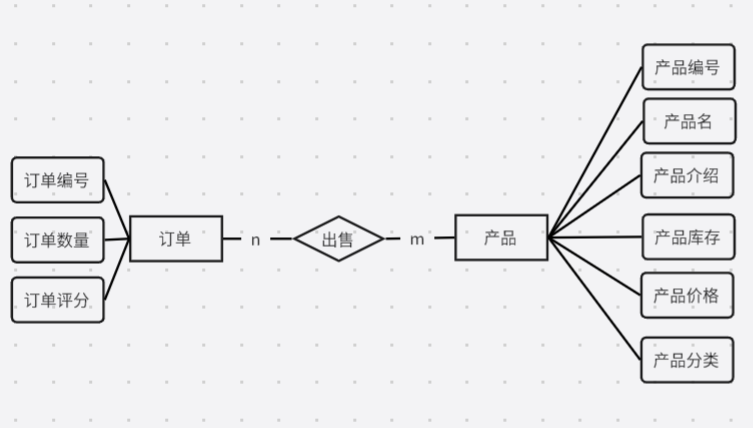


图 2. 3 订单-产品 E-R图

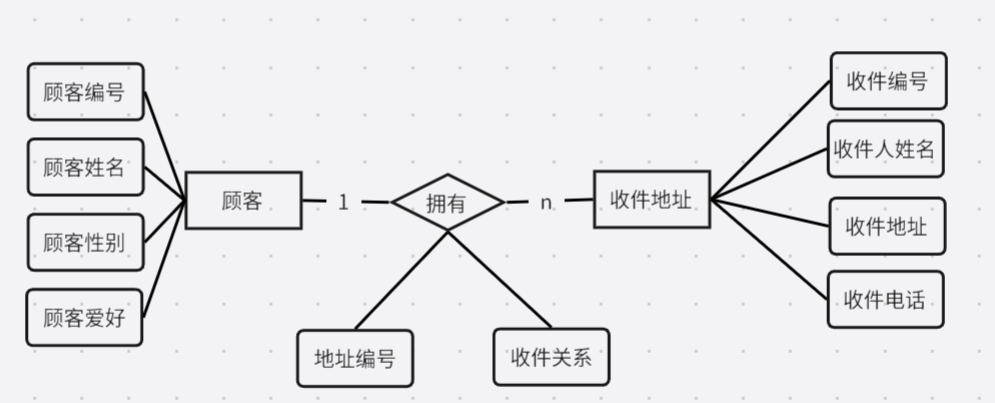


图 2. 4 顾客-收件 E-R图

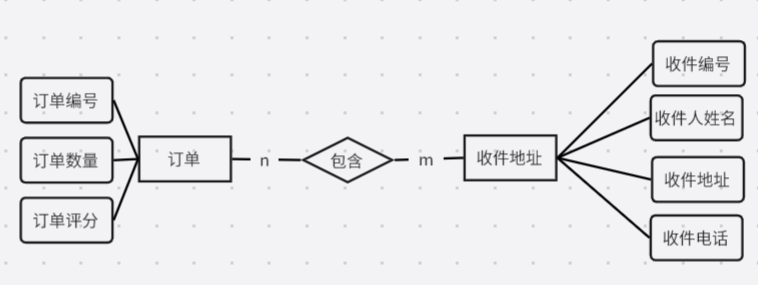


图 2. 5 订单-收件 E-R图

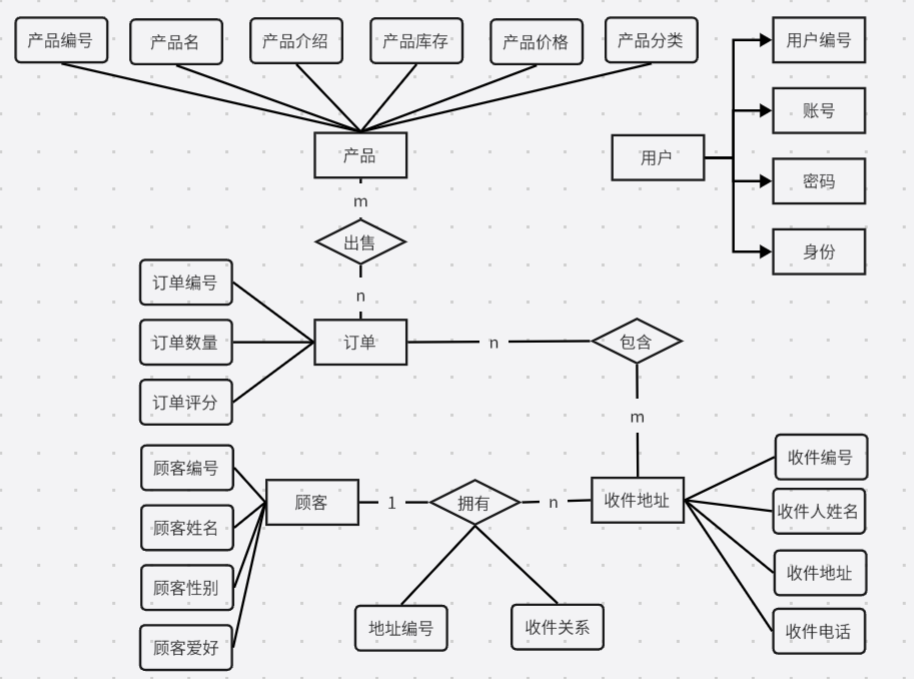


图 2. 6 顾客购物订单管理系统全局 E-R图

### 2.3 逻辑结构设计

本系统的关系模式如下（加粗下划线表示主码，斜体下划线表示外码）：

⚫ 产品信息表（**产品编号**，**产品名**，产品介绍，产品库存，产品价格，产品分类）

⚫ 订单信息表（**订单编号**，订单数量，订单评分,*产品编号*,*收件编号*）

⚫ 收件信息表（**收件编号**，收件人姓名，收件地址，收件电话）

⚫ 地址管理表（**地址编号**，收件编号，顾客编号，收件关系）

⚫ 顾客信息表（**顾客编号**，顾客姓名，顾客性别，顾客爱好）

⚫ 用户表（**编号**，账号，密码，身份）

将系统的 E-R 图转换为数据库模式如下：

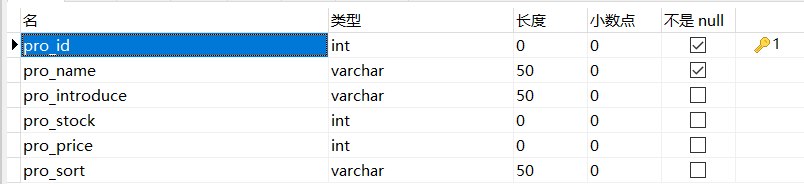


图 2. 7 产品信息表

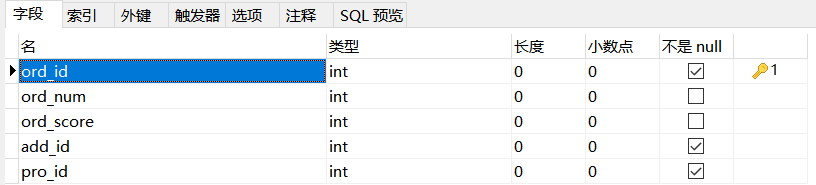


图 2. 8 订单信息表



图 2. 9 收件信息表



图 2. 10 地址管理表



图 2. 11 顾客信息表



图 2. 12 用户表

### 2.4 关系模式规范化检查及处理

范式是符合某一种级别的关系模式的集合。关系数据库中的关系必须满足一定的要

求，满足不同程度要求的为不同范式。

第一范式（1NF）：如果一个关系模式 R 的所有属性都是不可分的基本数据项，则

称该关系模式满足第一范式，记作 R∈1NF。

第二范式（2NF）：若关系模式 R∈1NF，并且每一个非主属性都完全函数依赖于任

何一个候选码，则 R∈2NF。

第三范式（3NF）：设关系模式 R<U，F>∈1NF，如果 R 中不存在候选码 X、属性组

Y 以及非主属性 Z（Z ⊄Y），使得 X→Y（反之不可推出），Y→Z 成立，则 R∈3NF。

BC 范式（BCNF）：关系模式 R<U，F>∈1NF，若 X→Y 且 Y ⊄X 时 X 必含有码，则

R<U，F>∈BCNF。

根据上述的要求，将本数据库的关系模式进行规范化检查，如下（**注：表内的自增编号不在规范化中考虑**）：

**产品信息表（产品编号，产品名，产品介绍，产品库存，产品价格，产品分类）**

该表的码为：产品编号

函数依赖：产品编号→产品名，（产品编号，产品名）→产品介绍，（产品编号，产品名）→产品库存，（产品编号，产品名）→产品价格，（产品编号，产品名）→产品分类

由该函数依赖可以分析得知，存在对候选码的部分依赖，满足 1NF

**订单信息表（订单编号，产品编号，收件编号，订单数量，订单评分）**

该表的码为：订单编号

函数依赖：订单编号→订单数量,订单编号→产品编号,订单编号→收件编号，订单编号→订单评分

由该函数依赖可以分析得知，不存在对候选码的部分依赖，满足 2NF；不存在传递依赖，满足 3NF，每一个决定属性因素都包含码，满足 BCNF。

**收件信息表（收件编号，收件人姓名，收件地址，收件电话）**

该表的码为：收件编号

函数依赖：收件编号→收件人姓名，收件编号→收件地址，收件编号→收收件电话

由该函数依赖可以分析得知，不存在对候选码的部分依赖，满足 2NF；不存在传递依赖，满足 3NF，每一个决定属性因素都包含码，满足 BCNF。

**地址管理表（地址编号，收件编号，顾客编号，收件关系）**

该表的码为：地址编号

函数依赖：地址编号→收件编号，地址编号→顾客编号，地址编号→收件关系

由该函数依赖可以分析得知，不存在对候选码的部分依赖，满足 2NF；不存在传递依赖，满足 3NF，每一个决定属性因素都包含码，满足 BCNF。

**顾客信息表（顾客编号，顾客姓名，顾客性别，顾客爱好）**

该表的码为：顾客编号

函数依赖：顾客编号→顾客姓名，顾客编号→顾客性别，顾客编号→顾客爱好

由该函数依赖可以分析得知，不存在对候选码的部分依赖，满足 2NF；不存在传递依赖，满足 3NF，每一个决定属性因素都包含码，满足 BCNF。

**用户表（编号，账号，密码，身份）**

该表的码为：编号

函数依赖：编号→账号，编号→密码，编号→身份

由该函数依赖可以分析得知，不存在对候选码的部分依赖，满足 2NF；不存在传递依赖，满足 3NF，每一个决定属性因素都包含码，满足 BCNF。

# 第 3 章 数据库定义与操作

### 3.1 数据库及数据表定义

数据库的创建：

⚫ create database Online\_Shopping;

数据表的创建及定义：

⚫ 产品信息表：

create table product

(

pro\_id int auto\_increment

primary key,

pro\_name varchar(50) not null,

pro\_introduce varchar(50) null,

pro\_stock int default 0 null,

pro\_price int default 0 null,

pro\_sort varchar(50) null

);

⚫ 订单信息表：

create table orders

(

ord\_id int auto\_increment

primary key,

ord\_num int default 0 null,

ord\_score int default 10 null comment '0-10分',

add\_id int not null,

pro\_id int not null,

constraint order\_address\_add\_id\_fk

foreign key (add\_id) references address (add\_id),

constraint order\_product\_pro\_id\_fk

foreign key (pro\_id) references product (pro\_id)

);

⚫ 收件信息表：

create table address

(

add\_id int not null

primary key,

add\_name varchar(50) null,

add\_phone varchar(50) null,

add\_info varchar(50) null

);

⚫ 地址管理表：

create table address\_info

(

inf\_id int not null

primary key,

use\_id int not null,

add\_id int not null,

inf\_relationship varchar(50) null

);

⚫ 顾客信息表：

create table users

(

use\_id int not null

primary key,

use\_sex int default 0 not null comment '0 - 男1 - 女,

use\_hobby varchar(50) null,

use\_name varchar(50) null,

constraint users\_login\_log\_id\_fk

foreign key (use\_id) references login (log\_id)

on delete cascade

);

⚫ 用户表：

create table login

(

log\_id int not null

primary key,

log\_name varchar(50) null,

log\_password varchar(50) null,

log\_role int default 1 null comment '0 — 管理员1 — 顾客

);

### 3.2 数据查询操作

## 3.2.1 关系代数实现

由于表名字和属性过长，故在写关系代数时将部分表名字和属性简写，规则如下：

表 3.2. 1 名称简写表

|  |  |  |  |
| --- | --- | --- | --- |
| 原名 | 简写 | 原名 | 简写 |
| product | p | order | o |
| address | a | login | l |
| address\_info | ai | users | u |

**注：所有关系代数的实现按照顺序从上往下对应 SQL 实现的顺序，画图工具限制，部分的符号没法绘制一模一样的。**

1. 查询登录账号对应的用户名：



1. 查询订单的产品信息（产品名，产品数量）



1. 查询编号为2的订单的收件信息（收件人名字，收件地址）



1. 查询账号名为‘123’的订单信息（订单号，产品名，收件姓名，发件人名）：



### 3.2.2 SQL 实现

（1）查询登录账号对应的用户名：

select users.use\_name

from users

where use\_id = (

select log\_id

from login

);

（2）查看该订单的产品名

select p.pro\_name,o.ord\_num

from product p ,orders o

where p.pro\_id = o.pro\_id;

（3）查询该订单的地址信息

select add\_name,add\_info,add\_phone

from address

where add\_id = (select add\_id

from orders

where orders.ord\_id = 2);

（4）查询该订单的发件人，收件信息，与产品信息

select o.ord\_id,p.pro\_name,a.add\_name,l.log\_name

from product p

right join orders o on p.pro\_id = o.pro\_id

left join address a on o.add\_id = a.add\_id

left join address\_info ai on a.add\_id = ai.add\_id

left join login l on ai.use\_id = l.log\_id

where l.log\_name = '123';

### 3.3 数据增删改操作

⚫ 数据增加：

1. 新增产品

insert into product values (null,null,0,0,null);

1. 增加订单

insert into `order` values (?,1,10,?,?);

⚫ 数据删除：

(1)删除收件信息

delete from address

where add\_id = ?;

(2)删除产品信息

delete from product

where pro\_id = ?;

(3)删除订单，当我们删除了该地址

delete from orders

where add\_id = (select add\_id

from address

where add\_name = null);

⚫ 数据修改：

(1)根据地址ID更新收件人信息

update address

set add\_name = ?

where add\_id = ?;

(2)当我们购买了商品，减少库存

update product

set pro\_stock = pro\_stock - (select ord\_num

from orders

where ord\_id = ?)

where pro\_id = ?;

1. 增加库存

update product

set pro\_stock = ?

where pro\_id = ?;

### 3.4 索引及视图应用

## 3.4.1 索引的应用

索引提供指向存储在表的指定列中的数据值的指针，然后根据您指定的排序顺序对这些指针排序。数据库使用索引以找到特定值，然后顺指针找到包含该值的行。这样可以使对应于表的 SQL 语句执行得更快，可快速访问数据库表中的特定信息。

当表中有大量记录时，若要对表进行查询，在表中建立索引，然后在索引中找到符合查询条件的索引值，最后通过保存在索引中的 ROWID（相当于页码）快速找到表中对应的记录，可以大大的提高检索效率。

基于上述的诸多优点，本系统的许多地方都可以创建索引，由于并没有实际投入运作，本次只选取其中一个表插入大量数据测试索引的用途，具体过程如下：

1. 在 login 表中插入 8 千条左右的数据：

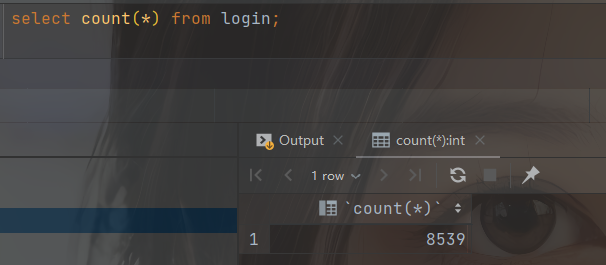


图 3. 1 数据总数图

1. 在未加索引的情况下直接查询某一个数据：

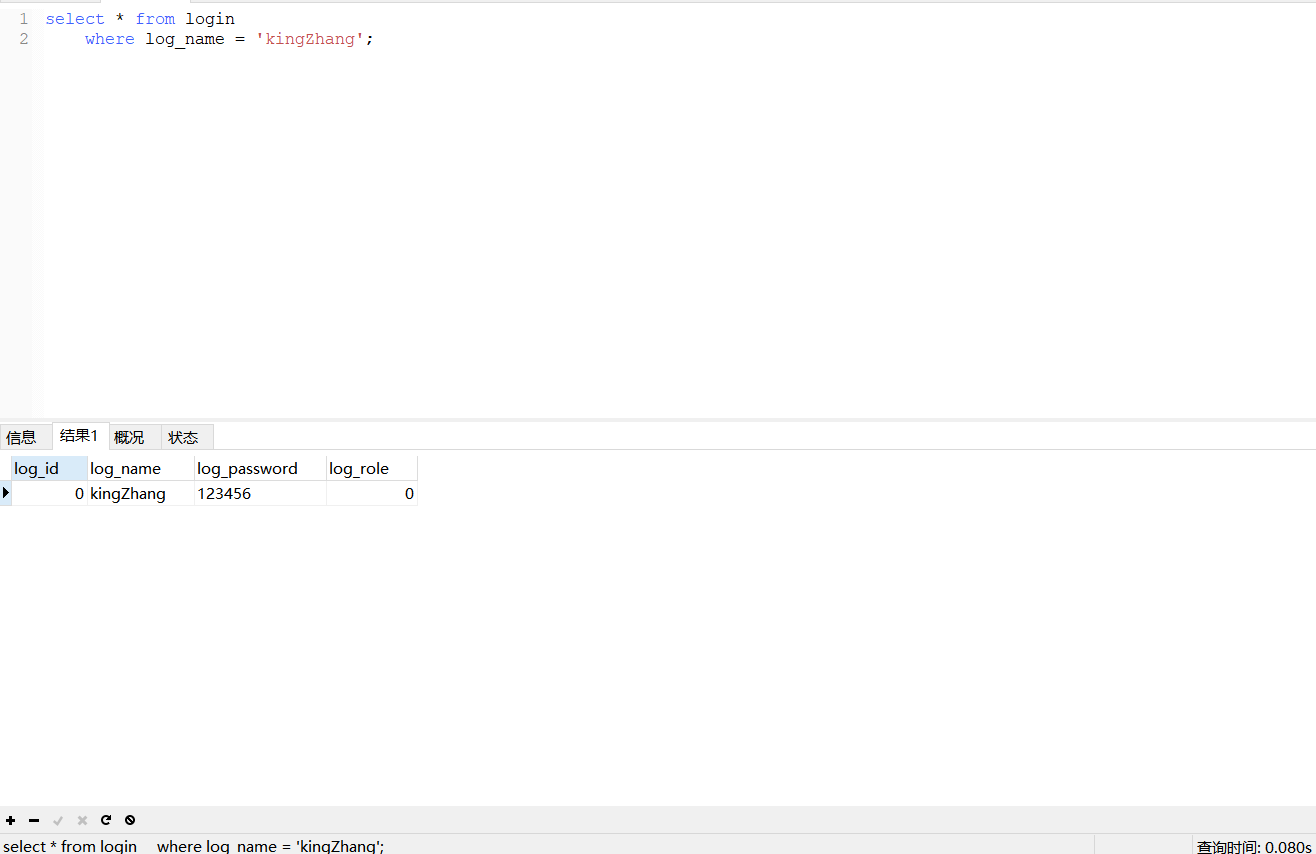


图 3. 2 未加索引的查询结果

③ 从上述的结果可以得知，查询一条数据需要花费 0.08 秒，为了提高查询效率，创建一个索引，语句如下：

create index login\_log\_name\_index

on login (log\_name);

④ 创建索引之后，可以看到在该表中便有了该索引，如下：



图 3. 3 索引图

⑤ 在创建好索引之后再次查询相同的语句，得到的结果如下：

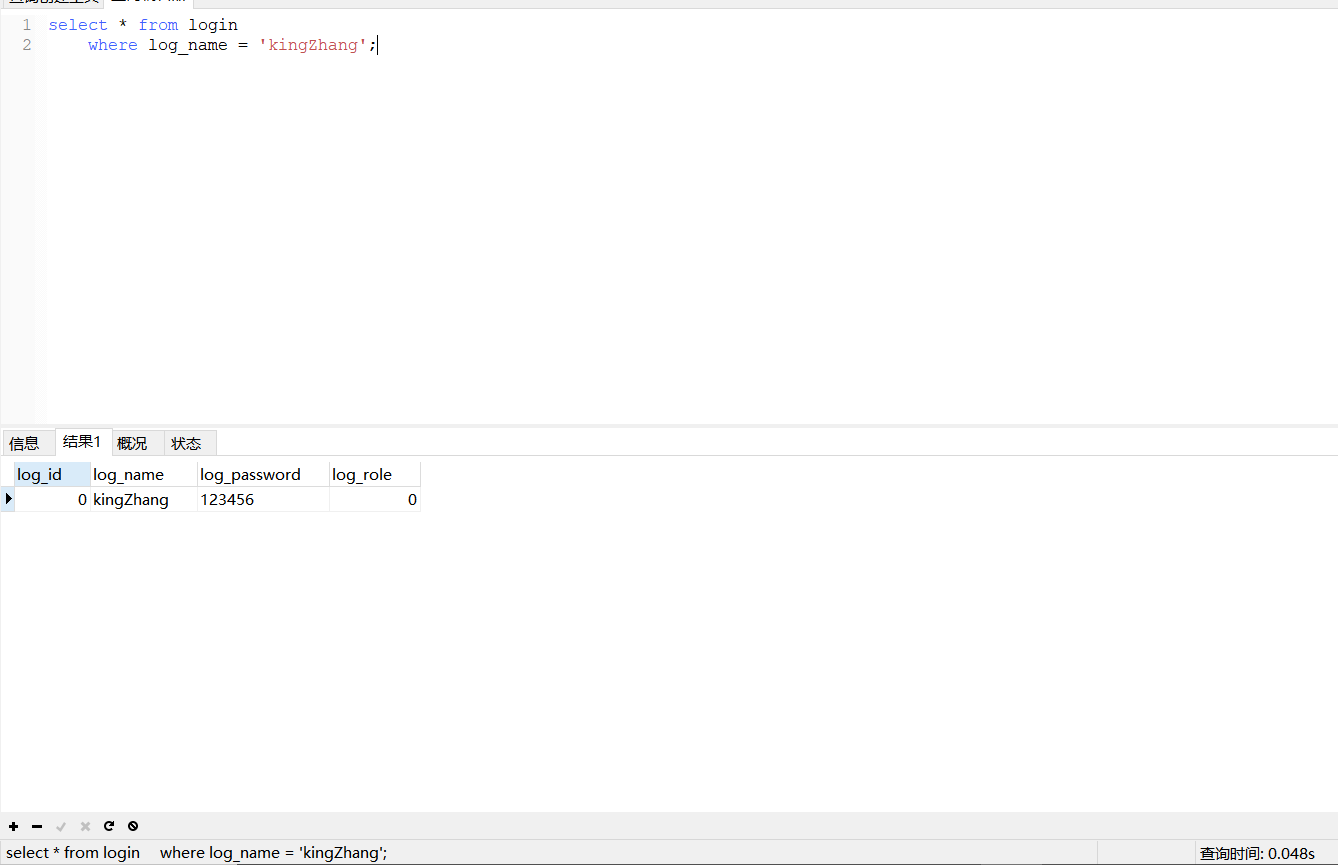


图 3. 4 添加索引后查询结果

⑥ 结论：多次查询相同的语句发现结果都是 0.048 秒，与没有使用索引的 0.08s 相比 较，提高了查询的效率，本次只使用了最基础简单的查询，当遇到实际复杂的业务时，使用索引会极大地提高效率。

## 3.4.2 视图的应用

视图是指计算机数据库中的视图，是一个虚拟表，其内容由查询定义。同真实的表一样，视图包含一系列带有名称的列和行数据。但是，视图并不在数据库中以存储的数据值集形式存在。行和列数据来自由定义视图的查询所引用的表，并且在引用视图时动态生成。

由于用户和产品在数据库中存储的信息都是他们的 ID，查询一些具体信息都需要多重的连接，比较的麻烦，所以我创建了视图，将病人和医生匹配的信息都汇总在一张视图表中，创建语句如下：

由于订单和顾客在数据库中存储的信息都是他们的 ID，查询一些具体信息都需要多重的连接，比较的麻烦，所以我创建了视图，将订单和顾客匹配的信息都汇总在一张视图表中，创建语句如下：

create view orderProductAddress as

select o.ord\_id,p.pro\_name,o.ord\_num,l.log\_name,a.add\_name,o.ord\_score

from login l,address\_info ai,address a,orders o,product p

where l.log\_id = ai.use\_id and ai.add\_id = a.add\_id

and a.add\_id = o.add\_id and o.pro\_id = p.pro\_id;

and a.add\_id = o.add\_id and o.pro\_id = p.pro\_id;

在查询相关需求的时候就会比较容易，例如:需要查询姓名为‘123’的用户是否购买过了名字为‘电饭煲’的产品，以及对他的评分，就可以简单的实现，如下：

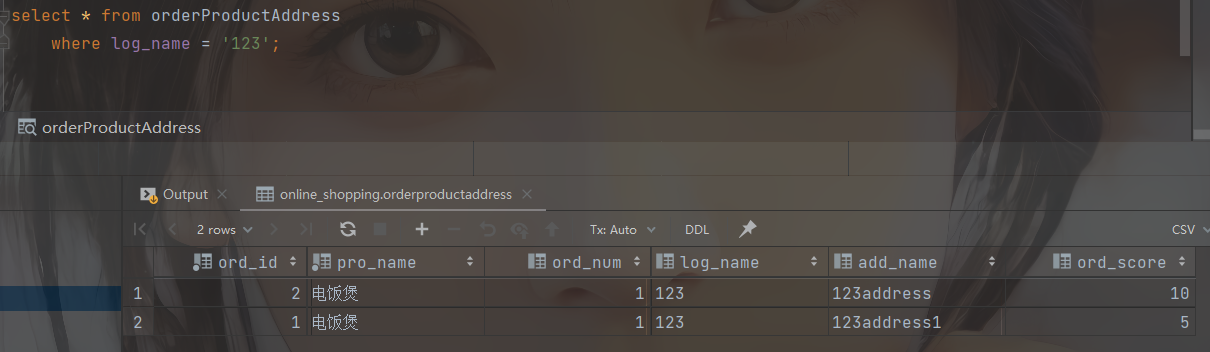


图 3. 5 通过视图查询实例

# 第 4 章 应用系统实现

采用Java的SpringBoot+ mybatisplus框架开发后端，使用RESTFUL风格定义后端接口，采用vue2 +Element+ vueRouter系列框架开发前端，使用了封装echarts的v-charts技术开发图表，数据库采用 mySQL 来完成 数据的操作。本系统通过 MVCM 设计模型实现。如下图所示：

IMG_256

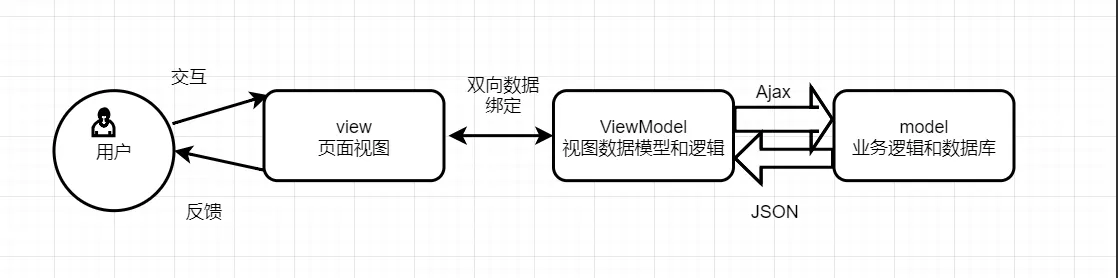


图 3. 6 MVCM 设计模式

### 4.1 登录页面

首先在创建vue项目，导入vueRouter，V-echarts，element-ui，axios等包，之后创建vueRouter的js文件管理页面与Hash地址的联系。创建login页面

## 4.1.1 页面设计

Element-ui的el-form-item组件创建表单，使用data创建登录要用的用户对象，通过axios使用后端的接口。



图 4. 1 登录页面

## 4.1.2 功能实现

用户在使用时，输入账号与密码，点击登录。若输入正确，则进入对应的窗口，反之，当账户错的时候提示“账号错误”，当账户对的时候密码错误，提示“密码错误”。点击注册可以进入注册页面。

首先是通过v-model获取输入框的内容，再通过具体的函数 login 来判断该账号和密码是否对应数据库 login 表中的一行记录，若符合，再将其身份传递回来，判断其身份，通过管理员的身份登录。当身份为用户时，提示“用户权限不够”。

后端get方法是通过 loginService 查询是否存在账号和密码都与输入内容一致的记录，和用户的权限是否满足。若有则返回 true，若没有则返回 false。

### 4.2 注册页面

创建register页面。并编写基本格式。

## 4.2.1 页面设计

Element-ui的el-form-item组件创建表单，使用data创建登录要用的用户对象，通过axios使用后端的接口。



图 4. 2 注册页面

## 4.2.2 功能实现

该功能与登录类似，唯一不同的是需要判断账号是否存在数据库，和选择管理员身份后管理员码输入是否正确，其余类似，不再赘述。

### 4.3 产品管理页面

创建userinfo页面。并编写基本格式。在data创建产品的属性对象

## 4.3.1 页面设计

主页面设计的思路是上下两部分组成，上部分其他功能键以及搜索输入框。下面部分是产品的展示表格以及修改与删除操作。



图 4. 3 产品管理页面

### 4.3.2 功能实现

当登录的进去的时候页面通过getInfo显示产品信息。点击删除，前端通过scope.row 获取该行的id，之后通过 deleteById 删除产品。点击编辑，前端通过scope.row 获取该行的id，并弹出修改页面，之后通过 update 修改产品信息。比如：图4.4 点击添加，前端弹出添加页面。添加产品的内容。之后通过 saveInfo 上传产品。比如：图4.5



图 4. 4 产品信息修改页面



图 4. 5 添加产品页面

### 4.4 订单管理页面

在data创建订单的属性对象

## 4.4.1 页面设计

主页面设计的思路是上下两部分组成，上部分其他功能键以及搜索输入框。下面部分是产品的展示表格以及评价与删除操作。



图 4. 6 订单管理页面

### 4.4.2 功能实现

当登录的进去的时候页面通过getOrderInfo显示产品信息。点击删除，前端通过scope.row 获取该行的id，之后通过 deleteOrderById 删除产品。点击评价，前端通过scope.row 获取该行的id，并弹出评价页面，之后通过 updateOrder评价订单。比如：图4.7

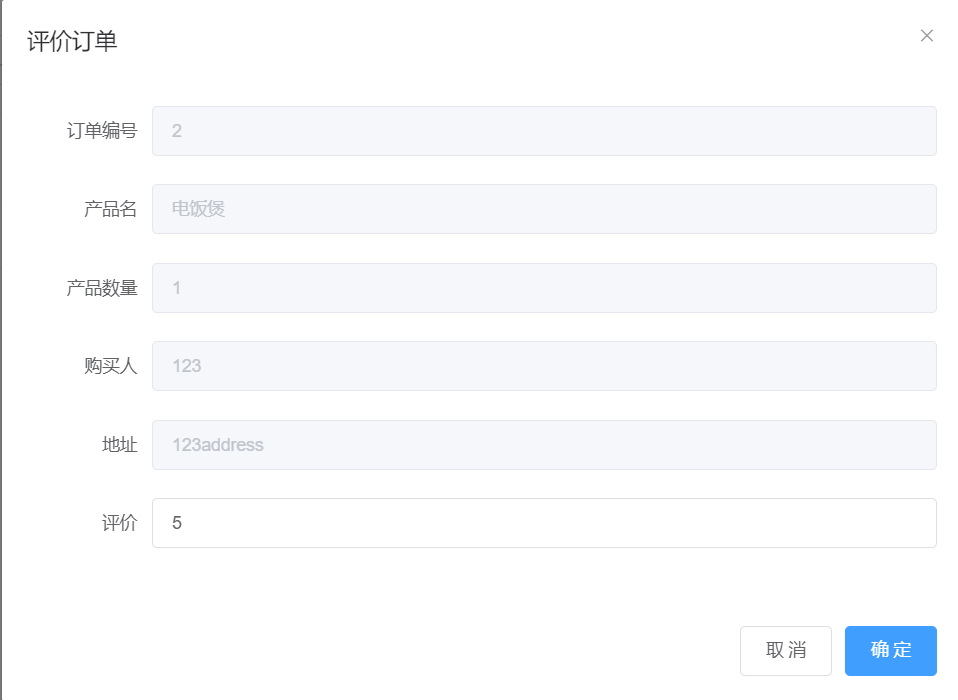


图 4. 7 订单评价页面

### 4.5 购物页面

以data的产品的属性对象为基础

## 4.5.1 页面设计

主页面设计的思路是上下两部分组成，上部分其他功能键以及搜索输入框。下面部分是产品的展示表格以及购买操作。



图 4. 8 购物页面

### 4.4.2 功能实现

当登录的进去的时候页面通过getInfo显示产品信息。点击购买，前端通过scope.row 获取该行的id，并弹出购买页面，之后填写产品数量。比如：图4.9

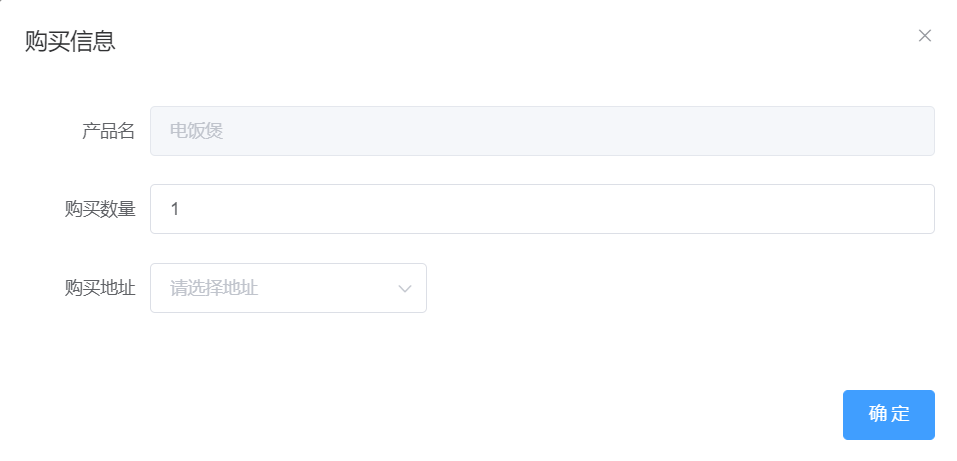


图 4. 9 购买页面

### 4.5 库存统计页面

使用了v-charts技术，开发库存统计表

## 4.5.1 页面设计

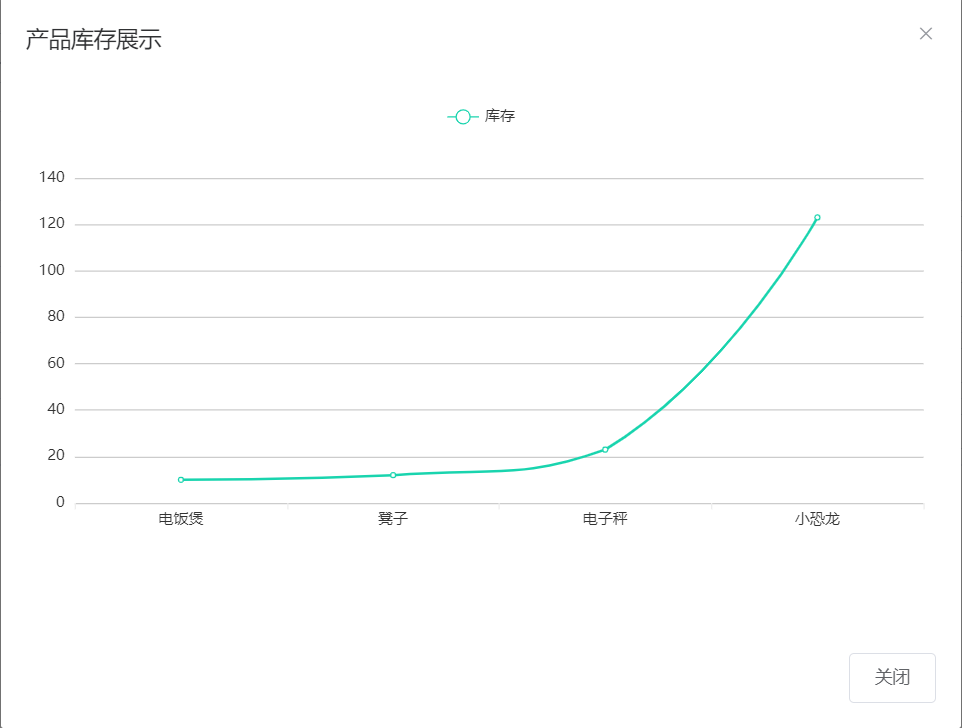


图 4. 10 库存统计页面

### 4.5.2 功能实现

当登录的进去的时候页面通过getInfo显示产品信息。点击展示，前端利用v-charts技术，并通过拼接的手段组合该数据需要的数据

# 第 5 章 总结

本次的课程设计我认为意义重大，在报告的从头到下我重新温故知新了数据库的大多数的知识，对于遗忘了的知识的重新学习了一遍。对于已经了解的知识有了新的了解。这是我第一次尝试数据库到java后端最后到前端的整体联系。在这过程中，也让我熟悉了前后端的直接交互方式，对于前后交互的流程有了更清晰的认识。

在设计表的过程中我还算顺利，在咨询林老师和王老师时，都对于我的ER图提出了宝贵的意见。在设计功能模块和用例的时候，我感觉脑子非常混乱，感觉这些步骤没有什么用，而且特别繁琐。但是当我们写到第四章的应用系统实现。这些步骤的条理性才显现出来，对于java程序的设计就更加合理和简洁。而不是跟java高级大作业一样。想到什么写什么。使功能很冗余。多个方法功能类似，功能重复。功能想一个写一个。

在写关系代数实现的时候，我是头大的，首先是符号难以编辑，例外是一堆关系代数理不清逻辑，其本质是关系代数的理解不够深入。当我再次观看《数据库系统概论》里面的知识点。对于关系代数理论有了更深的理解。比如：对于条件查询可以利用选择和投影这两个运算的结合，选择是为了选择需要的属性组，投影是为了选择需要的数据。对于多表查询，可以加入连接运算符连接多表。

在写前后端的交互时，对于后端的接口考虑如何编写让前端识别的返回类。那就是公共返回类设计。之后还要在后端解决前后端的跨域问题。之后就是考虑前端的设计。前端使用axios方式接收数据。之后在data里面编辑对应数据库的数据对象用来接收后端的接口的数据。之后展示在前端。

总的来说，此次的课设我非常的满足，我的收获很大,在编写项目的过程，知识储备很重要，如果没有知识储备，做项目肯定会遇到很多难题，这些难题不是设计的难题而是技术的难题，这是很痛苦且难以坚持下去的。所以我之后的目标多学习技术。