

**数据库系统原理实践报告**

|  |  |
| --- | --- |
| 项目名称： | 网上销售系统 |
| 姓 名： | 顾时禹 |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS1808 |
| 学 号： | U201814734 |
| 指导教师： |  |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

20 年 月 日

**教师评分页**

|  |  |
| --- | --- |
| 子目标 | 子目标评分 |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
|  |  |

目 录

[1 课程任务概述 1](#_Toc73891777)

[1.1 软件功能学习部分 1](#_Toc73891778)

[1.2 SQL练习部分 1](#_Toc73891779)

[2 软件功能学习部分 6](#_Toc73891780)

[2.1 任务要求 6](#_Toc73891781)

[2.2 完成过程 6](#_Toc73891782)

[2.3 任务总结 8](#_Toc73891783)

[3 SQL联系部分 8](#_Toc73891784)

[3.1 任务要求 8](#_Toc73891785)

[3.2 完成过程 8](#_Toc73891786)

[3.3 任务总结 18](#_Toc73891787)

[4 综合实践任务 18](#_Toc73891788)

[4.1 系统设计目标 18](#_Toc73891789)

[4.2 需求分析 19](#_Toc73891790)

[4.3 总体设计 20](#_Toc73891791)

[4.4 数据库设计 21](#_Toc73891792)

[4.5 详细设计与实现 22](#_Toc73891793)

[4.6 系统测试 22](#_Toc73891794)

[4.7 系统设计与实现总结 22](#_Toc73891795)

[5 课程总结 22](#_Toc73891796)

[附录 23](#_Toc73891797)

# 1 课程任务概述

简要陈述介绍本实践课程的各项任务要求。

## 1.1 软件功能学习部分

完成下列1~2题，并在实践报告中叙述过程，可适当辅以插图

1）某个主流关系数据库管理系统软件（华为GaussDB、阿里云、SQL Server等）的备份方式：数据和日志文件的脱机备份、系统的备份功能。

2）练习在新增的数据库上增加用户并配置权限的操作，通过用创建的用户登录数据库并且执行未经授权的SQL语句验证自己的权限配置是否成功。

## 1.2 SQL练习部分

假设在某个区域内的所有地点都存储在地点表中。该地区中的所有人员信息存储在人员表中。根据收集到的人员行程建立了疫情期间某个地区的人员行程表。

该地区的部分人员进行了核酸检测，检测结果保存在诊断表中。根据诊断表中的检测结果，对新冠确诊和无症状感染者的密切接触者进行隔离，密切接触者的信息存储到密切接触表中，隔离信息存储到隔离表中。隔离地点共设置了4个，并且每个隔离地点都有容量限制，隔离点的信息存储在隔离地点表中。

### 1.2.1 建表

1. **根据以上背景，请在DBMS中创建这些关系，包括主码和外码的说明，并写出指定关系的建表SQL语句**

**人员表【人员编号，姓名，电话】**

person (id int, fullname char(20), telephone char(11))

主码为人员编号

**地点表【地点编号，地点名称】**

location (id int, location\_name char(20))

主码为地点编号

**行程表【行程编号，人员编号，所在地点编号，开始时间，结束时间】**

itinerary (id int, p\_id int, loc\_id int, s\_time datetime, e\_time datetime)

主码为行程编号，

**诊断表【诊断编号，人员编号，诊断日期，诊断结果】**

diagnose\_record (id int, p\_id int, diagnose\_date datetime, result int)

主码为诊断编号，诊断结果包括：1：新冠确诊，2：无症状感染者，3：正常

**密切接触者表【密切接触编号，被接触者编号，接触日期，接触地点，病例人员编号】**

close\_contact (id int, p\_id int, contact\_date datetime, loc\_id int, case\_p\_id int)

主码为密切接触编号

**隔离表【隔离编号，被隔离人员编号，开始隔离日期，结束隔离日期，隔离地点编号，隔离状态】**

isolation\_record (id int, p\_id int, s\_date datetime, e\_date datetime, isol\_loc\_id int, state int)

主码为隔离编号，隔离状态：1：正在隔离 2：隔离结束 3：转入医院

**隔离地点表【隔离地点编号，隔离地点名，房间容量】**

isolation\_location (id int, location\_name char(20), capacity int)

主码为隔离房间编号

1. **观察性实验**

验证在设置外码时是否一定要参考被参照关系的主码，并在实验报告中简述过程和结果。

1. **数据准备**

依据后续实验的要求，向上述表格中录入适当数量的实验数据，从而对相关的实验任务能够起到验证的作用。

### 1.2.2 数据更新

1. **分别用一条sql语句完成对人员表基本的增、删、改的操作；**
2. **批处理操作**

将行程表中所有到达地点2的记录插入到新表location\_record\_2中。

1. **数据导入导出**

通过查阅DBMS资料学习数据导入导出功能，并将任务2.1所建表格的数据导出到操作系统文件，然后再将这些文件的数据导入到相应空表。

1. **观察性实验**

建立一个关系，但是不设置主码，然后向该关系中插入重复元组，然后观察在图形化交互界面中对已有数据进行删除和修改时所发生的现象。

1. **触发器实验**

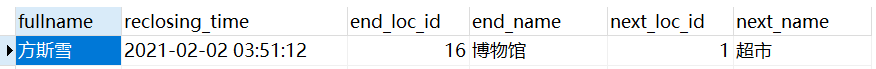
编写一个触发器，用于实现以下完整性控制规则：

当隔离表中的某位隔离人员在诊断表中的诊断结果为“1：新冠确诊”，将隔离状态从“1”改成“3”。

### 1.2.3 查询

请分别用一条SQL语句完成下列各个小题的需求：

1. 查询截至目前的人流量大于30的地点，并按照从高到低排序
2. 查询出每个隔离地中正在进行隔离的人数，并按数量由多到少排序
3. 查询行程表中连续行程中的一对行程地点（例如，方斯雪，在2021-02-02 03:02:12至2021-02-02 03:51:12 在博物馆中，2021-02-02 03:51:12 至 2021-02-02 04:23:12出现超市中，输出【姓名，重合时间，起始地点id，起始地点，结束地点id，结束地点】，查询结果如下图所示）



1. 查询贾涵山的行程情况
2. 查询地名中带有‘店’字的地点名称
3. 新发现一位确诊者，已知他在2021.2.2日20:05:40到21:25:40之间在“活动中心”，查询他接触到的人员的名字
4. 查询正在使用的隔离区名（使用DISTINCT关键字）
5. 查询有出行记录的人员（使用EXISTS关键字）
6. 查询没有达到地点“Today便利店”的人数（使用NOT EXISTS 关键字）
7. 查询去过所有地点的人员
8. 新建一个视图，里面有每个隔离点的id,名称，已用房间，容量
9. 从视图中查询还有空房间的隔离点的剩余房间数目
10. 查询靳宛儿传染的人有哪些
11. 查询每个地点的密切接触者的数量
12. 查询感染人数最多的用户的名字和感染人数
13. 查询2021-02-02 10:00:00到14:00:00，行程记录最频繁的五个人及行程记录数目

### 1.2.4 了解系统的查询性能分析功能（选做）

选择上述2.3任务中某些较为复杂的SQL语句，查看其执行之前系统给出的分析计划和实际的执行计划，记录观察的结果，并对其进行简单的分析。

### 1.2.5 DBMS函数及存储过程（选做）

1. 编写一个依据人员编号计算其达到所有地点的次数的自定义函数，并利用其查询至少到达过3个地点的人员。
2. 尝试编写DBMS的存储过程，建立每个隔离点的人数统计表，并通过存储过程更新该表。

**注意：任务书中的红色和蓝色文字不要出现在实践报告中。**

**关于正文排版要求：**

正文统一采用小四号宋体/Times New Roman和1.25倍行距，段前、段后均0磅间隔。

正文应论述清楚，文字简练通顺，插图简明，书写整洁。文中图、表按制图要求绘制。

**关于正文章节标题的排版要求：**

报告章标题称为一级标题，章内小节标题依次分为二级标题、三级标题等。一级标题的编号用数字1，2，…编制；二级标题的编号用1.1，1.2，…编制；三级标题的编号用1.1.1，1.2.1，… 编制；四级及以后各级标题可依此类推。建议标题不超过3级（如1.1.1），超出部分可根据需要使用(1)，①，A，a)，…等形式描述。

标题编号与标题文字之间均用空格隔开，如：“1 引言”、“2.1 需求分析”。报告正文的一级标题（章）须另起一页居中排版。

1级标题宋体小三号字加黑，单倍行距，段前、段后均12磅间隔。

2级标题宋体四号字加黑，1.25倍行距，段前6磅，段后0磅间隔。

3级标题宋体小四号字加黑，单倍行距，段前3磅、段后8磅间隔。

**关于正文中的图的排版要求：**

正文中所有插图要求图面整洁，布局合理，线条粗细均匀，圆弧连接光滑，尺寸标注规范。所有曲线、图表、线路图、流程图、程序框图、示意图等不准徒手画，必须按国家规定标准或工程要求采用计算机或手工绘制。

所有插图均应有图号和图名。图号按章编，如第2章的图为图2.1、图2.2、…，第3章的图为图3.1、图3.2、…等。图名是插图的名称，扼要概括图的内容，字数不宜太多。图号和图名写在图下方，并相对于图居中排版。少数图有图注，图注写在图下面且字号应比图号、图名的字小一号，图名和图注后面均不加标点符号。

所有插图均应在正文中予以引用。引用某插图时，一般写为“…见图x.y”或“图x.y是…”。正文中的插图均须安排在文中第一次引用到该图的正文下面，要求先见文，后见插图，且图一般不跨页绘制。

图中文字、图号和图名，统一采用小五号宋体。

**关于正文中的表的排版要求：**

表格由表号、表名、表头、表身等组成。表号按章编，如第2章的表为表2.1、表2.2、…，第3章的表为表3.1、表3.2、…等。表名是表格的名称，扼要概括表的内容，字数不宜太多。表号、表名放在表的正上方，相对于表体居中排版。表号及表名后不加标点。表头包括栏头、行头，与表身一起构成表格的主体。表中的竖称为栏，横格称为行。表身的内容，一般包括：数据、文字、公式和表图等。表内的数据对应位要对齐。少数表有表注，表注写在表下面且字号应比表号、表名的字小一号。

所有表格均应在正文中予以引用。引用某表格时，一般写为“…见表x.y”或“表x.y是…”。表格应尽量靠近正文的叙述，一般应先见文，后见表，表不跨节。表格允许转页。表格转页部分可以不写表号和表名，但要重复书写表头，并在表头右上角写“（续）”字标注。

表中文字、图号和图名，统一采用小五号宋体。

**关于正文中公式的排版要求：**

公式一般另行居中写，公式末不加标点。若公式前有文字，如例、解、证、假定等，文字顶格写，公式仍居中写。一行写不下时，公式允许转行。公式转行需处理得当，做到既意义正确，又使版面美观匀称。

公式要有编号，公式编号按章编，如第2章的公式为(2.1)、(2.2)、…，第3章的公式为(3.1)、(3.2)、…等。公式编号写在公式右侧行末顶边线，并加圆括号。

公式一般应在正文中予以引用，引用时以公式编号指示公式。正文中常有公式中表示量的符号说明，采用“式中”二字作为标志。一般可写成接排形式，如“式中，A指……；B指……”。

**关于实践报告的目录格式要求：**

目录是课程实验报告的纲要。正文的各级标题（一般最多取三级）、附录应编入目录，但目录本身不出现在其中。

目录要求层次清晰，含标题及对应的起始页号。目录的最后 “附录”无章节号。

课程实验报告正文、参考文献和附录页面，使用“1，2，3，… ”编连续页码。页码应标在页面的右下角。

目录中正文的各级标题名称、参考文献和附录及其对应的起始页号，务必与报告中正文的各级标题名称、附录及其对应的起始页号保持一致。

# 软件功能学习部分

## 2.1 任务要求

练习sql的备份方式。

练习在新增的数据库上增加用户并配置权限的操作。

## 2.2 完成过程

### 2.2.1 数据备份

在MySQL数据库中备份数据可以使用mysqldump命令，如图2.1，该命令生成输出一个包含创建并插入当前数据库中数据的sql语句的sql文件，如图2.2。



图2.1 数据备份

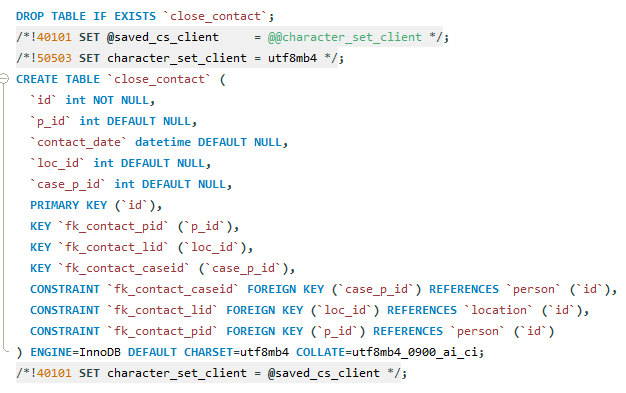


图2.2 备份得到的sql文件片段

在sql命令行中使用source命令即可恢复数据库。

### 2.2.2 增加用户并配置权限

通过create user命令可以创建新用户，如图2.3。

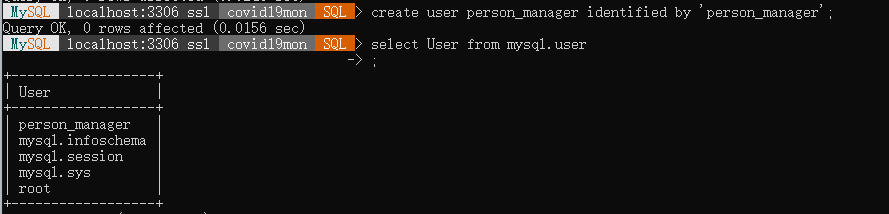


图2.3 创建用户person\_manager

通过grant命令可以授予用户权限，如图2.4。



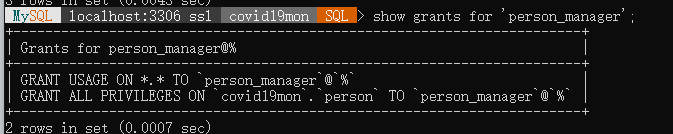


图2.4 授予pearson\_manager对person表的所有权限

登录到person\_manager用户，分别查询person表和location表，发现可以正常查询person表，但是查询location表被拒绝，如图2.5，可见用户权限的配置正确。

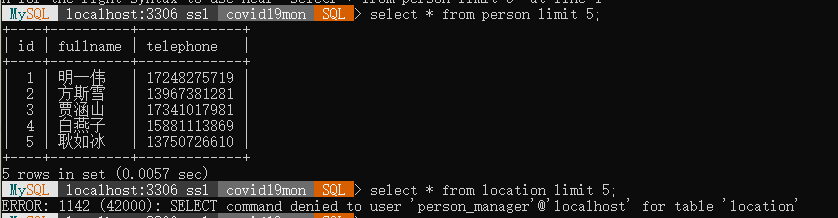


图2.5 用户权限检查

## 2.3 任务总结

通过这次实验我学会了如何为数据库做备份，以及如何管理数据库的用户和其权限。这些都可以用简单的命令完成，方便数据库管理员维护，管理数据库。

# 3 SQL联系部分

## 3.1 任务要求

假设在某个区域内的所有地点都存储在地点表中。该地区中的所有人员信息存储在人员表中。根据收集到的人员行程建立了疫情期间某个地区的人员行程表。

该地区的部分人员进行了核酸检测，检测结果保存在诊断表中。根据诊断表中的检测结果，对新冠确诊和无症状感染者的密切接触者进行隔离，密切接触者的信息存储到密切接触表中，隔离信息存储到隔离表中。隔离地点共设置了4个，并且每个隔离地点都有容量限制，隔离点的信息存储在隔离地点表中。

对这个数据库进行建表，数据更新，查询，了解系统的查询性能分析功能，DBMS函数及存储过程。

## 3.2 完成过程

实验基于MySQL。

### 3.2.1 创建数据库

create database covid19mon;

### 3.2.2 创建表和主外码约束

主要使用constraint语句来创建主键和外键约束，要注意创建约束时列名需要用括号括住，同时和关键词相同的列名需要用``括住。

create database if not exists covid19mon;

use covid19mon;

create table person (

id int,

fullname char(20) not null,

telephone char(11) not null,

constraint pk\_person primary key (id)

);

-- 表2 地点表(location)

create table location (

id int,

location\_name char(20) not null,

constraint pk\_location primary key (id)

);

-- 表3 行程表（itinerary）

create table itinerary (

id int,

p\_id int,

loc\_id int,

s\_time datetime,

e\_time datetime,

constraint pk\_itinerary primary key (id),

constraint fk\_itinerary\_pid foreign key (p\_id) references person(id),

constraint fk\_itinerary\_lid foreign key (loc\_id) references location(id)

);

-- 表4 诊断表（diagnose\_record）

create table diagnose\_record (

id int,

p\_id int,

diagnose\_date datetime,

result int,

constraint pk\_diagnose\_record primary key (id),

constraint fk\_diagnose\_pid foreign key (p\_id) references person(id)

);

-- 表5 密切接触者表（close\_contact）

create table close\_contact (

id int,

p\_id int,

contact\_date datetime,

loc\_id int,

case\_p\_id int,

constraint pk\_close\_contact primary key (id),

constraint fk\_contact\_pid foreign key (p\_id) references person(id),

constraint fk\_contact\_lid foreign key (loc\_id) references location(id),

constraint fk\_contact\_caseid foreign key (case\_p\_id) references person(id)

);

-- 表6 隔离地点表（isolation\_location）

create table isolation\_location (

id int,

location\_name char(20),

capacity int,

constraint pk\_isolation\_loc primary key (id)

);

-- 表7 隔离表（isolation\_record）

create table isolation\_record (

id int,

p\_id int,

s\_date datetime,

e\_date datetime,

isol\_loc\_id int,

state int,

constraint pk\_isolation primary key (id),

constraint fk\_isolation\_pid foreign key (p\_id) references person(id),

constraint fk\_isolation\_lid foreign key (isol\_loc\_id) references isolation\_location(id)

);

### 3.2.3 数据插入

Values关键字后用逗号隔开可以一次性插入多条数据。

use covid19mon;

INSERT INTO person( `id` , `fullname` , `telephone` ) VALUES (1, "张小敏", "13907110001"), (2, "李大锤", "18907110002"), (3, "孙二娘", "13307100003");

### 3.2.4 数据删除

use covid19mon;

DELETE FROM `person` WHERE `id` = 2;

### 3.2.5 数据更新

use covid19mon;

UPDATE `person` SET `telephone` = "13607176668" WHERE `id` = 1;

### 3.2.6 查询人流量大于30的地点

使用count函数和group by关键字可以按地点统计到场人数。

select location\_name, visitors

from (

select l.location\_name, count(\*) as visitors

from itinerary i, location l

where i.loc\_id=l.id

group by loc\_id

) temp

where visitors > 30

order by visitors desc, location\_name asc;

### 3.2.7 查询每个隔离点正在进行隔离的人数

可以使用desc和asc关键字控制排序的顺序

select location\_name, `number`

from (

select l.location\_name, count(\*) as `number`

from isolation\_location l, isolation\_record r

where l.id=r.isol\_loc\_id and r.state=1

group by isol\_loc\_id

) temp

order by `number` desc, location\_name asc;

### 3.2.8 查询接续行程

同时连接两个地点表查询，就可以得到接续行程。

select id, fullname, telephone, reclosing\_time, loc1, address1, loc2, address2

from (

select p.id, p.fullname, p.telephone, t2.reclosing\_time, t2.loc1, t2.address1, t2.loc2, t2.address2

from (

select t3.p\_id, t3.reclosing\_time, t3.loc1, t3.address1, t3.loc2, l1.location\_name as address2

from (

select t4.p\_id, t4.reclosing\_time, t4.loc1, l2.location\_name as address1, t4.loc2

from (

select i1.p\_id, i1.e\_time as reclosing\_time, i1.loc\_id as loc1, i2.loc\_id as loc2

from itinerary i1, itinerary i2

where i1.e\_time=i2.s\_time and i1.p\_id=i2.p\_id

) t4, location l2

where t4.loc1=l2.id

) t3, location l1

where t3.loc2=l1.id

) t2, person p

where p.id=t2.p\_id

) t1

where id > 30

order by id, reclosing\_time asc;

### 3.2.9 查询充珉瑶和贾涵山的行程情况

select fullname, telephone, location\_name, s\_time, e\_time

from (

select p.id, p.fullname, p.telephone, l.location\_name, i.s\_time, i.e\_time

from (

select \* from person where fullname="充珉瑶" or fullname="贾涵山"

) p

left join itinerary i

on i.p\_id=p.id

left join location l

on i.loc\_id=l.id

order by p.id desc, i.s\_time asc

) temp;

### 3.2.10 查询地名中带有‘店’字的地点名称

使用%作为通配符查询字符串

select \* from location where location\_name like "%店%";

### 3.2.11 查询确诊者的接触者

开始时间和结束时间任意在目标时间段内的即视为接触者

select p.fullname, p.telephone

from (

select p\_id

from itinerary i

left join location l

on i.loc\_id=l.id

where l.location\_name="活动中心"

and (i.s\_time between '2021-02-02 20:05:40' and '2021-02-02 21:25:40'

or i.e\_time between '2021-02-02 20:05:40' and '2021-02-02 21:25:40')

) temp, person p

where p.id=temp.p\_id

order by p.fullname;

### 3.2.12 查询仍在使用的隔离点

select location\_name

from isolation\_location l

where exists (

select i.isol\_loc\_id

from isolation\_record i

where i.state=1 and i.isol\_loc\_id=l.id

);

### 3.2.13 查询有出行记录的人员

select fullname, telephone

from person p

where exists (

select p\_id

from itinerary i

where i.p\_id=p.id

)

order by id

limit 30;

### 3.2.14 查询没有去过“Today便利店”的人数

select count(\*) as `number`

from person p

where not exists (

select p\_id

from itinerary i, location l

where i.loc\_id=l.id and l.location\_name="Today便利店" and i.p\_id=p.id

);

### 3.2.15 查询去过所有地点的人员

select fullname

from person p

where not exists (

select \*

from location l

where not exists (

select \*

from itinerary i

where i.p\_id=p.id and i.loc\_id=l.id

)

)

order by fullname;

### 3.2.16 创建隔离点的现状视图

create view isolation\_location\_status as (

select l.id, l.location\_name, l.capacity, ifnull(occupied, 0) as occupied

from isolation\_location l

left join (

select r.isol\_loc\_id, count(\*) as occupied

from isolation\_record r

where r.state=1

group by r.isol\_loc\_id

) temp

on temp.isol\_loc\_id=l.id

order by l.id desc

);

### 3.2.17 查询各隔离点的剩余房间数

select location\_name, (capacity - occupied) as available\_rooms

from isolation\_location\_status

order by id;

### 3.2.18 查询与无症状者靳宛儿有过接触的人

select p.fullname, p.telephone

from person p

where p.id in (

select i2.p\_id

from itinerary i1, itinerary i2

where i1.p\_id=29

and i1.loc\_id=i2.loc\_id

and (

(i2.s\_time < i1.s\_time) and (i2.e\_time > i1.s\_time)

or (i2.s\_time < i1.e\_time) and (i2.e\_time > i1.e\_time)

or (i2.s\_time > i1.s\_time) and (i2.e\_time < i1.e\_time)

)

)

order by p.fullname;

### 3.2.19 查询每个地点发生的密切接触者的人数

select p.fullname, p.telephone

select location\_name, count(\*) as close\_contact\_number

from location l, close\_contact c

where c.loc\_id=l.id

group by l.location\_name

order by close\_contact\_number desc, location\_name asc;

### 3.2.20 查询感染人数最多的人

select c.case\_p\_id, p.fullname, count(\*) as infected\_number

from person p, close\_contact c

where c.case\_p\_id=p.id

group by c.case\_p\_id

order by infected\_number desc

limit 1;

### 3.2.21 查询行程记录最频繁的3个人

Limit语句可以控制结果的条数

select p.fullname, count(\*) as record\_number

from itinerary i, person p

where i.p\_id=p.id and (

i.s\_time between "2021-02-02 10:00:00" and "2021-02-02 14:00:00"

or i.e\_time between "2021-02-02 10:00:00" and "2021-02-02 14:00:00")

group by p.fullname

order by record\_number desc, p.fullname asc

limit 3;

### 3.2.22 查询房间数第二多的隔离点

通过limit语句可以限制跳过和选取的条数，跳过第一条选取一条极为第二多的隔离点。

select location\_name, capacity

from isolation\_location

order by capacity desc

limit 2,1;

### 3.2.23 创建触发器

在隔离点的人员确诊新冠后，将人员送入医院，创建一个update触发器和一个insert触发器，分别监听当诊断表被修改或增加记录时，需要在隔离表中做响应的修改。

use covid19mon;

drop trigger if exists uTrig;

drop trigger if exists iTrig;

create trigger uTrig

after update

on diagnose\_record

for each row

update isolation\_record

set state=3

where (isolation\_record.p\_id = NEW.p\_id and NEW.result = 1);

create trigger iTrig

after insert

on diagnose\_record

for each row

update isolation\_record

set state=3

where (isolation\_record.p\_id = NEW.p\_id and NEW.result = 1);

### 3.2.24 创建并使用函数

/\*

(1)用create function语句创建符合以下要求的函数：

依据人员编号计算其到达所有地点的次数(即行程表中的记录数)。

函数名为：Count\_Records。函数的参数名可以自己命名:\*/

use covid19mon;

set global log\_bin\_trust\_function\_creators=1;

delimiter ;;

create function Count\_Records(uid int) returns int

begin

declare result int;

select count(\*) from itinerary i where i.p\_id=uid into result;

return result;

end;;

delimiter ;

/\*

(2) 利用创建的函数，仅用一条SQL语句查询在行程表中至少有3条行程记录的人员信息，查询结果依人员编号排序。\*/

select \*

from person p

where Count\_Records(p.id) >= 3

order by id;

## 3.3 任务总结

在本次实验中我以新冠疫情为背景，创建了相关的数据库和表，并对表进行了一系列增删改查的操作。前半部分表的创建和增加，删除，修改数据部分都比较简单，只要使用响应的create, insert, add, delete函数配合where语句就可以完成，只要不出现语法错误一般不会有问题。

相比起来查找是平时使用数据库中的难点也是重点，尤其是在多表嵌套或连接查询的时候，需要理清层次，为了减少错误的产生，我习惯每次只连接进一个表，然后将得到的结果表再做下一步连接。

在实践的过程中，我也对数据库如何管理表，存储表，数据库怎样的设计才是高效的有了更深刻的认识，对我以后实际进行数据库设计有着很大的帮助。

# 4 综合实践任务

## 4.1 系统设计目标

1）应用背景

在高度信息化的当代，人们最普遍的购物方式已经从线下转移到了线上。各种各样的网上购物平台层出不穷，从大型全品类的淘宝，京东，到专注某一领域的唯品会，当当网等，各种各样网购平台虽然具体内容有所不同，但其系统结构层面的设计需求往往大同小异。通过自己实现一个简单的网上商城系统，可以帮助我们更好的了解这类系统的内部设计，有助于我们对具体工业业务的熟悉。

2）总体目标

实现一个简单的网上商城系统，包含用户登录，创建订单，商品管理，商店管理等基础功能。使用前后端分离的模式开发，并实现数据库事务，触发器等高级功能。

## 4.2 需求分析

### 4.2.1 任务文档需求

网上销售系统要求提供包括商品信息管理、查询、订购、销售等功能的网上交易平台，对客户和商店管理员应提供不同的操作界面和使用权限。

具体功能包括：

1. 商品信息管理：商店可以对商品信息进行管理，包括商品的类别、名称、描述信息、售价、图片、折扣等。
2. 客户信息管理：客户可以自助注册并管理自己的个人信息。
3. 商品查询：客户可以通过多种方式查询并且挑选网上商店出售的商品，通过在网上填写并确认订单的方式来购买商品。
4. 订单查询：客户可以管理自己的订单信息，查询订单的处理情况。
5. 订单处理：商店可以对客户的订单信息、汇款单信息进行审核，以确定是否发货，并修改订单状态。
6. 系统需包含事务（包含commit，rollback），存储过程/触发器，视图，函数。
7. 需要在程序中体现SQL和编程语言的结合。

设计一个B/S模式的系统实现上述功能。

### 4.2.2 任务需求分析

网上销售系统主要的用户有买家和卖家，系统中的主要部分包括用户，商店，商品和订单四种实体。

买家可以在系统中完成商品选择，下单，付款等操作，卖家则可以在系统中对商店进行管理，进行添加商品，更改商品库存，收款等操作。通过对用户进行权限管理来控制商店管理的权限。一个商店可以有多个用户作为管理员，一个用户也可以成为多个商店的管理员。设置每个商店的店长，只有店长可以增删商店的管理员。用户可以查看自己的订单信息，做简单的订单统计，管理员可以查看管理商店的概况，包括产品库存，销售额等信息。

## 4.3 总体设计

系统总体分为五层，如图4.1，交互层为前端页面，控制层负责接受解析前端请求，并调用相应的业务层接口，业务层编写具体的业务逻辑，并调用数据层提供的接口来操作数据库中的数据，数据层将打通底层数据库和高级语言的交互通道，把sql语句封装成java方法。

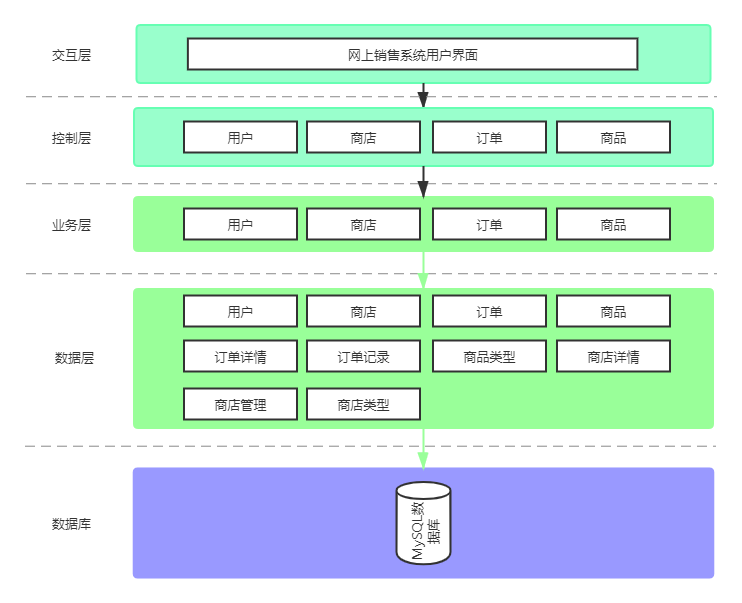


图4.1 系统层次结构图

用户，商店，订单和商品可以细分为几个功能模块，如图4.2。

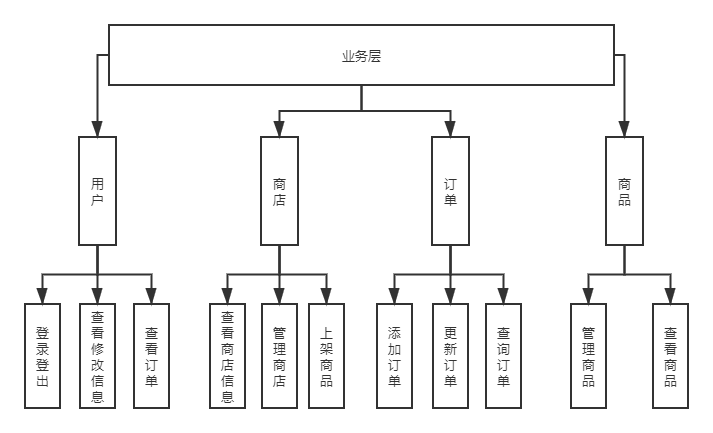


图4.2 系统功能模块图

## 4.4 数据库设计

数据库主要ER图如下

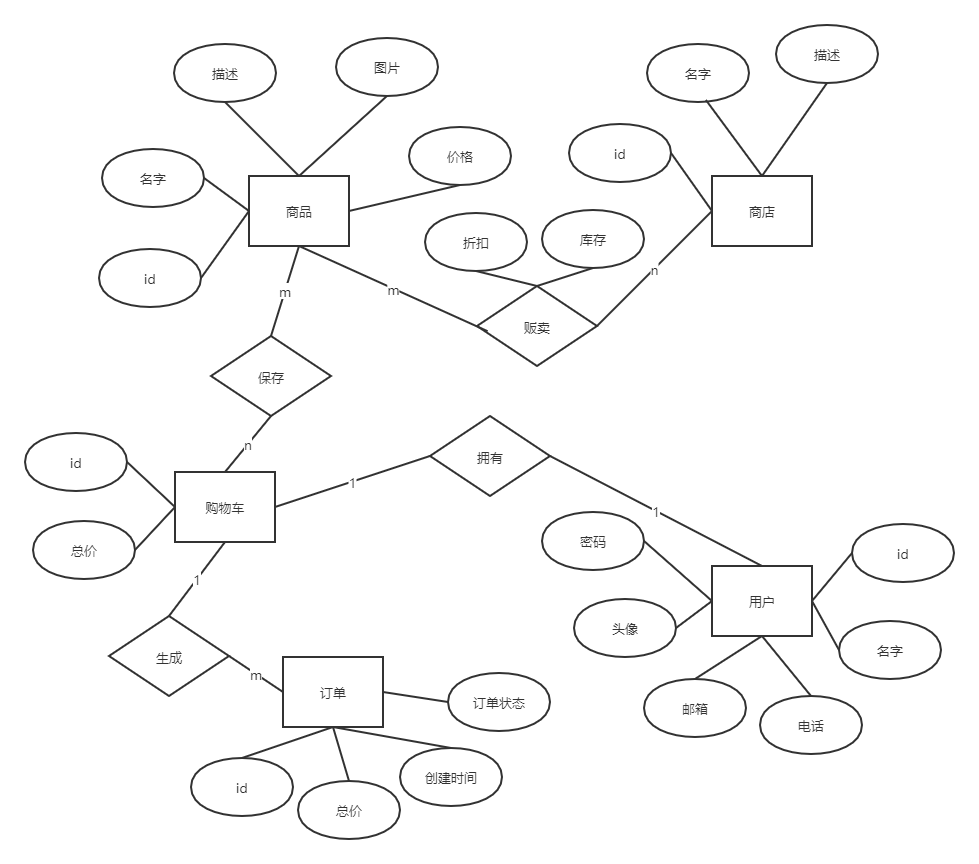


图4.3 数据库ER图

数据字典如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 创建时间 | datetime |  |  |  |
| 状态 | varchar(255) |  |  |  |

表4.1 order表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 总价 | double |  |  |  |
| 数量 | int |  |  |  |
| 订单id | int |  |  | 订单详情表项属于哪个订单 |
| 商品id | int |  |  | 订单详情表项描述的商品 |

表4.2 order\_item表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 商店id | int |  |  | 订单交易的商店 |
| 用户id | int |  |  | 订单交易的用户 |
| 订单id | int |  |  | 订单交易的订单 |
| 总价 | double |  |  |  |

表4.3 order\_record表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 描述 | varchar(255) |  |  |  |
| 名称 | varchar(255) |  |  |  |
| 图片 | varchar(255) |  |  |  |
| 价格 | double |  |  |  |
| 折扣 | double |  |  |  |
| 库存 | int |  |  |  |

表4.4 product表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 商品id | int |  |  |  |
| 类型id | int |  |  |  |

表4.5 product\_type表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 描述 | varchar(255) |  |  |  |
| 名称 | varchar(255) |  |  |  |
| 店主id | int |  |  | 商店管理员id |

表4.6 store表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 商店id | int |  |  |  |
| 商品id | int |  |  |  |

表4.7 store\_item表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 是否为店主 | bool |  |  |  |
| 商店id | int |  |  |  |
| 用户id | int |  |  |  |

表4.8 store\_manager表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 商店id | int |  |  |  |
| 类型id | int |  |  |  |

表4.9 store\_type表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 名称 | varchar(255) |  |  |  |

表4.10 type表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 主键 | 可空 | 备注 |
| id | int | 是 |  |  |
| 邮箱 | varchar(255) |  |  |  |
| 名称 | varchar(255) |  |  |  |
| 密码 | varchar(255) |  |  |  |
| 头像 | varchar(255) |  |  |  |
| 电话 | varchar(255) |  |  |  |

表4.11 user表

## 4.5 详细设计与实现

系统主要实现的复杂功能有：

用户创建订单操作，首先创建一个新的订单，初始化订单生成时间，订单的生成时间由触发器检测，当订单表中的一项被插入时，触发器自动将这项的创建时间填充为当前时间。之后从url中得到创建订单的用户id和商店id，以及商品id和数量列表，从数据库中查询出每个购买的商品，如果没有找到则抛出错误，找到商品后检查数量，如果商品数量不足则抛出错误，然后减少购买的数量，并在订单总价中累计购买商品的价值。最后更新订单的总价，并向订单明细表中插入相应的表项。

排序返回商品，从url中读取出两个参数，一个为排序的字段名称，为名称，价格，id中的一个，第二个参数为升序或降序，之后按照得到的参数调用不同的查询函数查询。

商品展示，创建一个商店商品详情视图，连表查询出每个商店存储商品的详细信息，包括商品名，描述，类型等。

系统前端部分使用React框架搭建，整体为一个单页面应用，不需要复杂的页面路由管理，只需通过切换页面的状态即可实现不同组件的展示，页面主要分为三个子页面，商城页面，用户页面和管理页面，分别提供商品购买，用户信息和订单展示，商店管理的功能。

## 4.6 系统测试

系统主界面如下

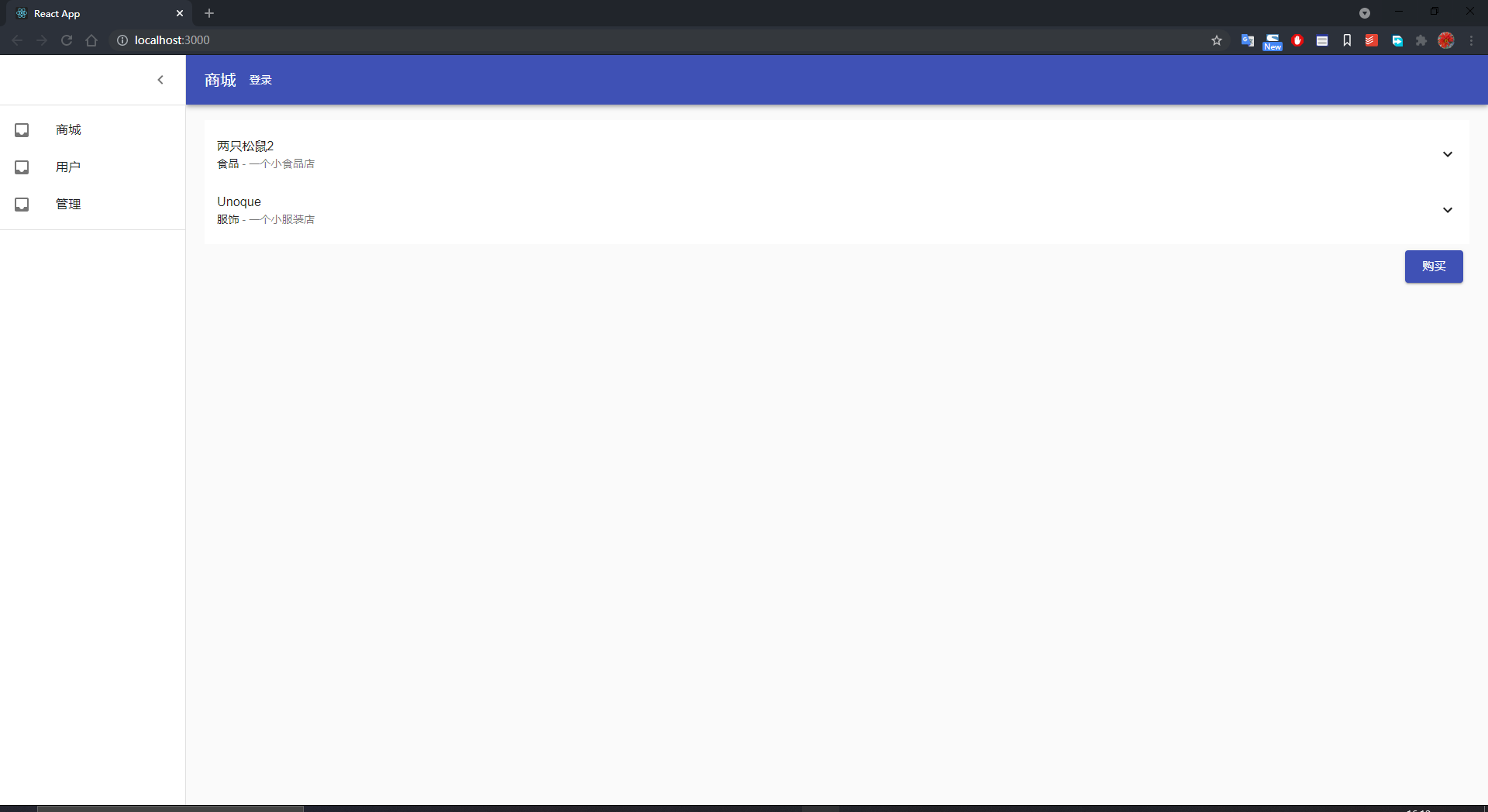


图5.1 主界面

用户输入电子邮箱和密码登录。

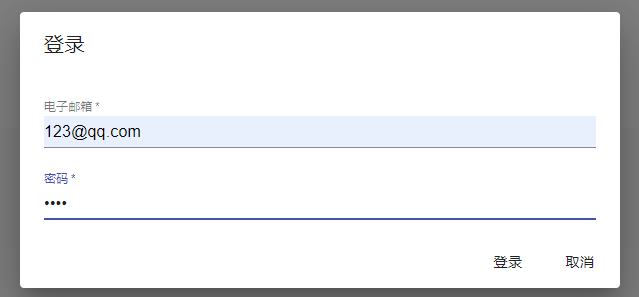


图5.2用户登录界面

商品展示界面展示了商品的简略信息，点击每个商品会弹出详情，可以修改右侧的购买数量调整数量，点击购买按钮购买商品。

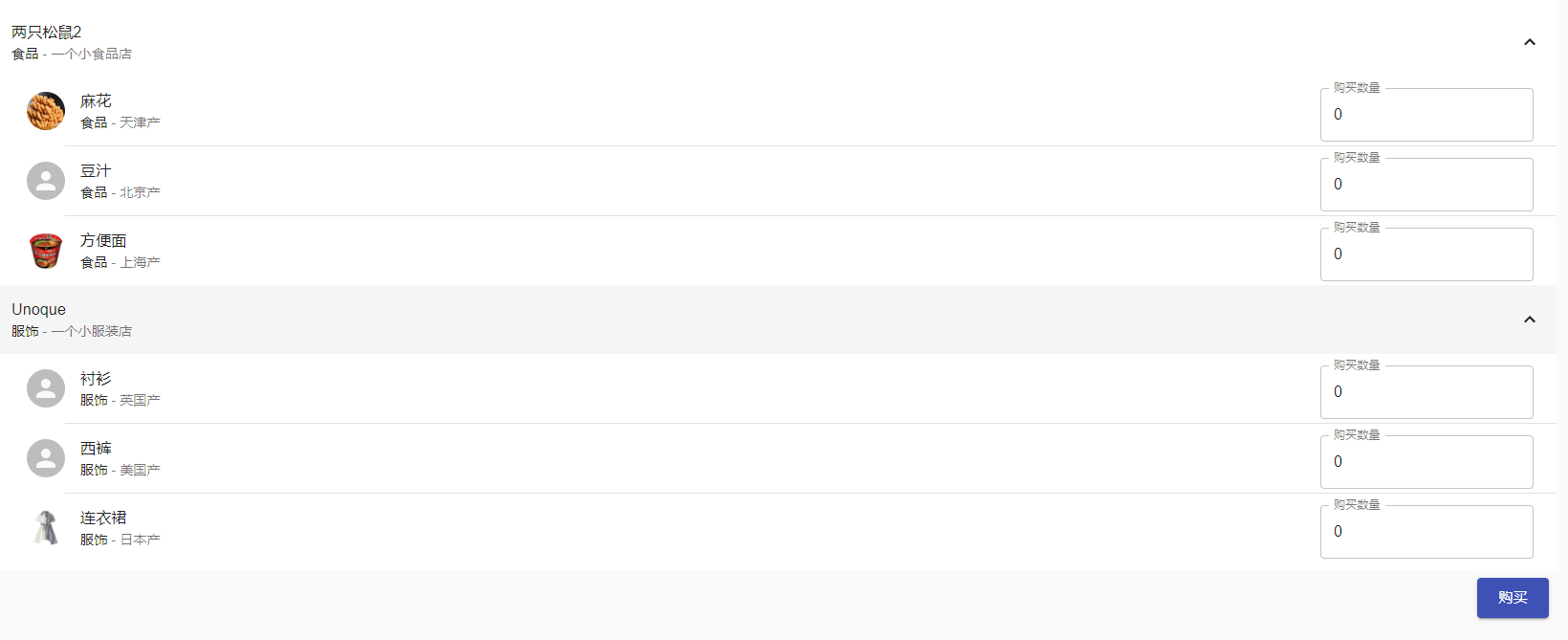


图5.3 商品展示界面

商品购买成功会显示订单总价。



图5.4 购买成功界面

用户界面展示用户信息，和用户的订单。

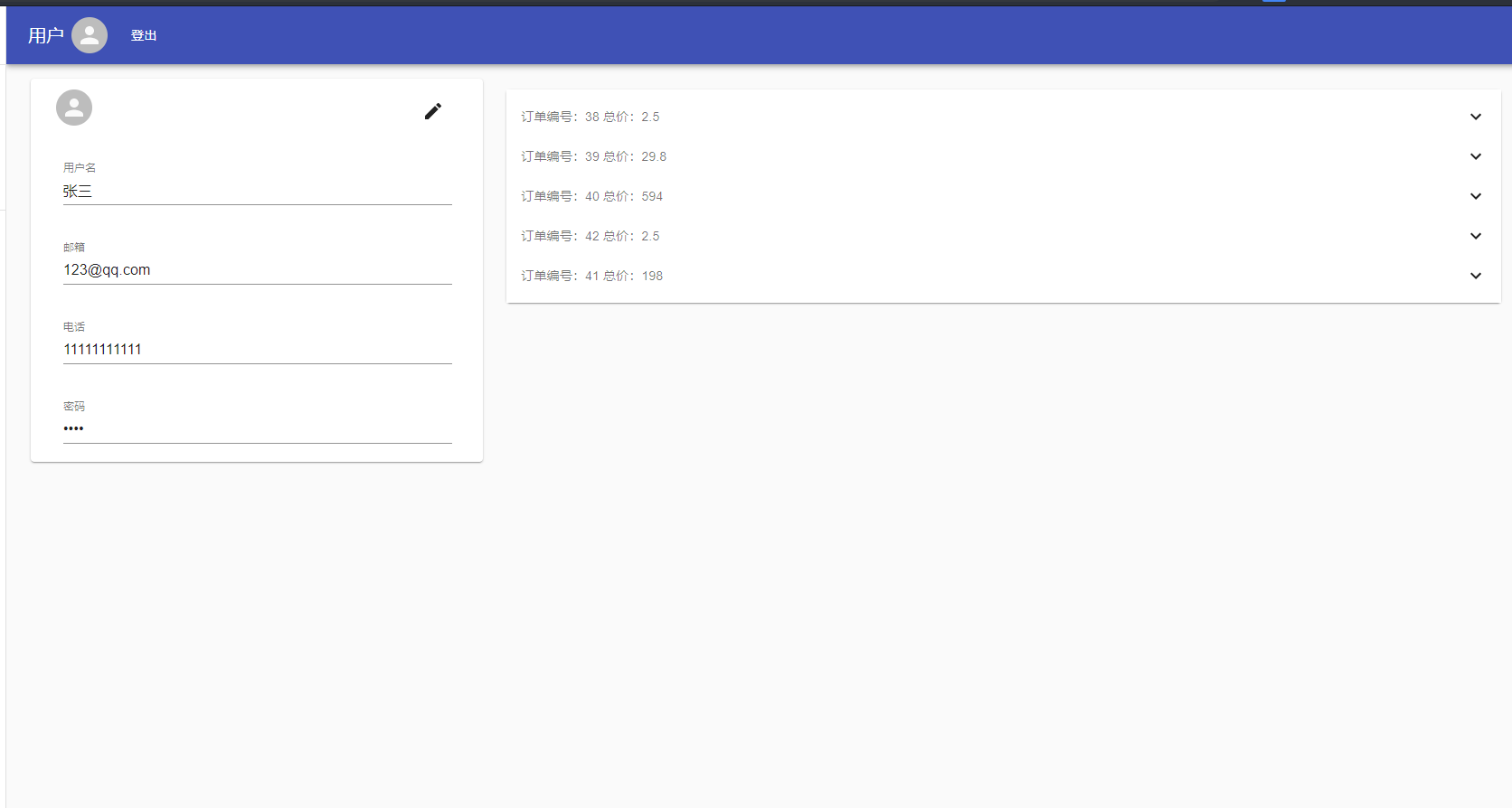


图5.5 用户信息界面

用户订单面板展开如图

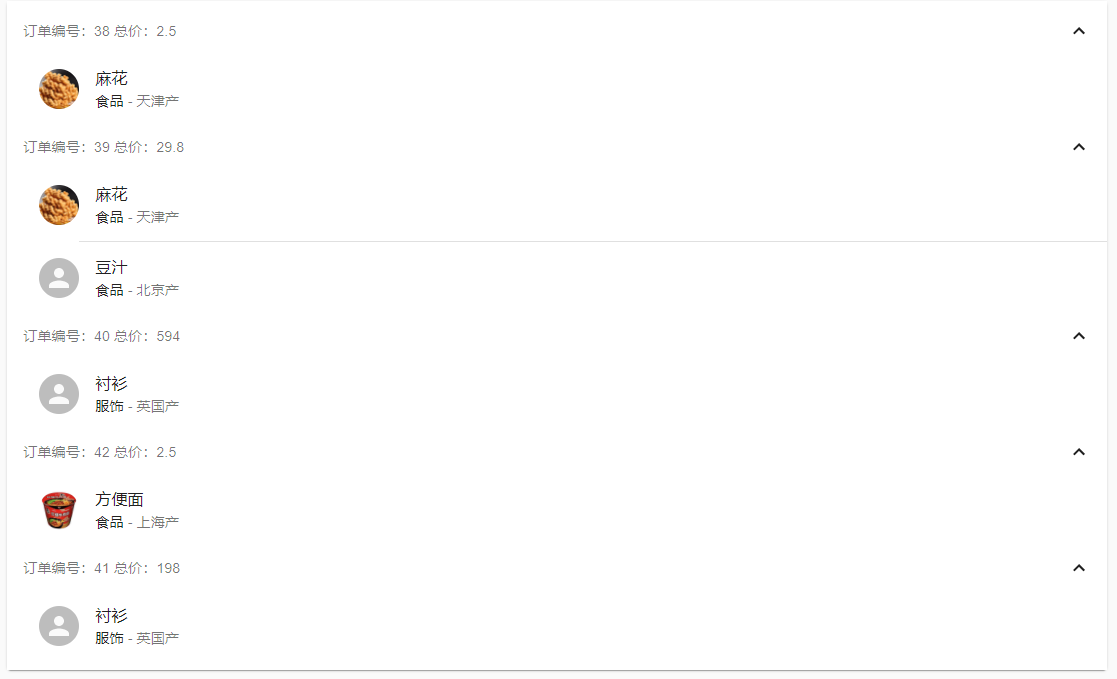


图5.6 用户订单详情

管理页面提供商店管理功能。

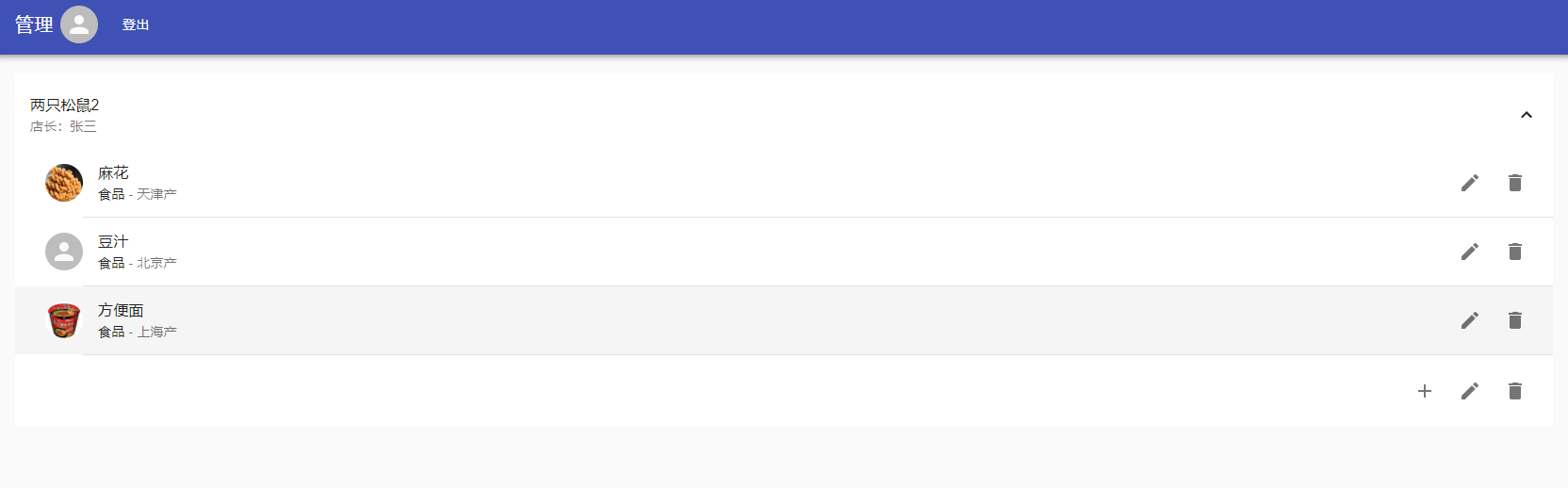


图5.7 商店管理页面

商品详情页面展示内容。



图5.8 商品详情页面

点击编辑按钮可以编辑商品内容。



图5.9 商品编辑界面

也可以删除商品。

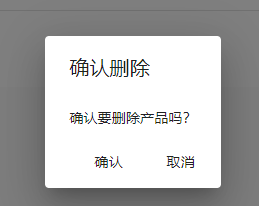


图5.10 商品删除界面

可以往商店中添加商品。



图5.11 商品添加界面

可以编辑商店的信息。

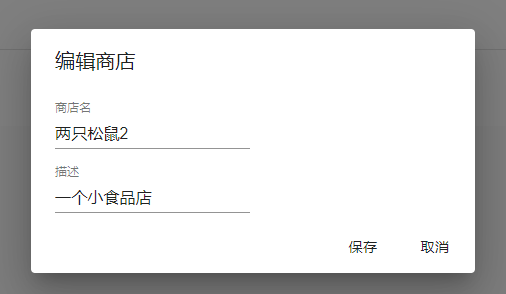


图5.12 商店编辑界面

## 4.7 系统设计与实现总结

系统数据库使用MySql数据库，除基本的数据存储功能外，使用到了触发器，视图，事务等高级功能。

系统后端部分使用SpringBoot+SpringData构建，分为模型层，DAO层，服务层和控制层四层，API结构采用RestFUL风格开发。

系统前端采用React框架编写，构建了一个单页面应用程序，使用异步请求从后端获得数据，并采用Material的UI设计风格。

系统总体实现了一个简单的网上商城管理系统，技术选型均使用工业界交较先进的技术，后续可以尝试引入redis，消息队列等技术，改进系统的性能。

# 5 课程总结

本次实验我首先学习了主流DBMS软件的使用方法，主要学习了MySql的使用方法。并熟悉了SQL语句的编写，在一个新冠疫情数据库中进行了简单的增删改查练习，最后我从零自行设计实现了一个商城管理系统，运用了理论课学习的数据库设计知识，自行设计了数据库的概念模型，关系模式，并学习了后端框架的相关知识，编写了一个简单的网上商城后端。也接触了简单的前端开发，熟悉了React的使用方法。

通过本次实验我对理论课中学到的知识有了更加深刻的认识，也实际进行了简单项目的开发，这次实践的经历对于我之后的学习工作有很大的意义。

# 附录