

数据库实验大作业报告

星云海·餐饮服务系统

姓名：魏天昊

学号：2019302733

班号：10011904

时间：2021.12.18

目录

概要.....	3
1. 实验内容与步骤.....	4
1.1 需求分析.....	4
1.1.1 用户需求分析.....	4
1.1.2 数据流图.....	4
1.1.3 数据字典.....	5
1.2 概念结构设计.....	6
1.2.1 实体型设计.....	6
1.2.2 分 ER 图(省去实体属性).....	7
1.2.3 总 ER 图(省去实体属性).....	8
1.3 逻辑结构设计.....	9
1.3.1 关系模式定义与范式分析.....	9
1.3.2 数据库中表的定义.....	10
1.4 关于视图、约束、存储过程、触发器与索引.....	11
1.4.1 视图.....	11
1.4.2 约束.....	11
1.4.3 存储过程.....	12
1.4.4 触发器.....	14
1.4.5 索引.....	16
2. 实验成果展示.....	16
2.1 主界面.....	16
2.1.1 管理员主界面.....	16
2.1.2 普通顾客主界面.....	17
2.2 功能界面示例.....	17
2.2.1 Provider.....	17
2.2.2 Dish.....	18
2.2.3 Purchase.....	18
2.2.4 Search.....	19
2.2.5 Login.....	19
3. 遇到的问题和方法.....	20
3.1 问题 1.....	20
3.2 问题 2.....	20
3.3 问题 3.....	20
4. 开发环境说明.....	21
4.1 开发系统.....	21
4.2 开发语言.....	21
4.3 开发平台.....	21
4.4 数据库版本.....	21

概要

在大学生活中，食堂是不可或缺的一部分。

经过我两年多的观察发现：

1. 同学们对于食堂的饭菜没有一个进行沟通与评价的平台，往往只能和身边朋友或表白墙来推荐或吐槽。
2. 食品安全存在着一些隐患，不能很快对食品进行溯源。

为此，我结合 MySQL 数据库，利用 Django 框架，精心设计了这套“星云海·餐饮服务系统”，提供日常生活中的食堂餐饮服务，力求克服以上两大问题。

服务包括：供应商查询与管理，窗口查询与管理，商品查询与管理，用户信息管理，历史购买记录查询与评价等等。

下面，我将详细介绍这一系统的设计与实现。

1. 实验内容与步骤

1.1 需求分析

1.1.1 用户需求分析

星云海餐饮服务系统主要的用户有两类：管理员和普通顾客。
下面分别对这两类用户的需求进行分析。

• **管理员：**

1. 增，查，删，改“供应商”相关信息。
2. 增，查，删，改“窗口”相关信息。
3. 增，查，删，改“商品”相关信息。
4. 增，查，删，改“顾客”相关信息。
5. 查，删，改“历史订单记录”相关信息。

• **普通顾客：**

1. 查看“供应商”相关信息。
2. 查看“窗口”相关信息。
3. 查看“商品”相关信息，对商品进行购买。
4. 查看个人信息，对个人信息进行修改。
5. 查看个人订单记录，对订单进行评价和删除。

1.1.2 数据流图

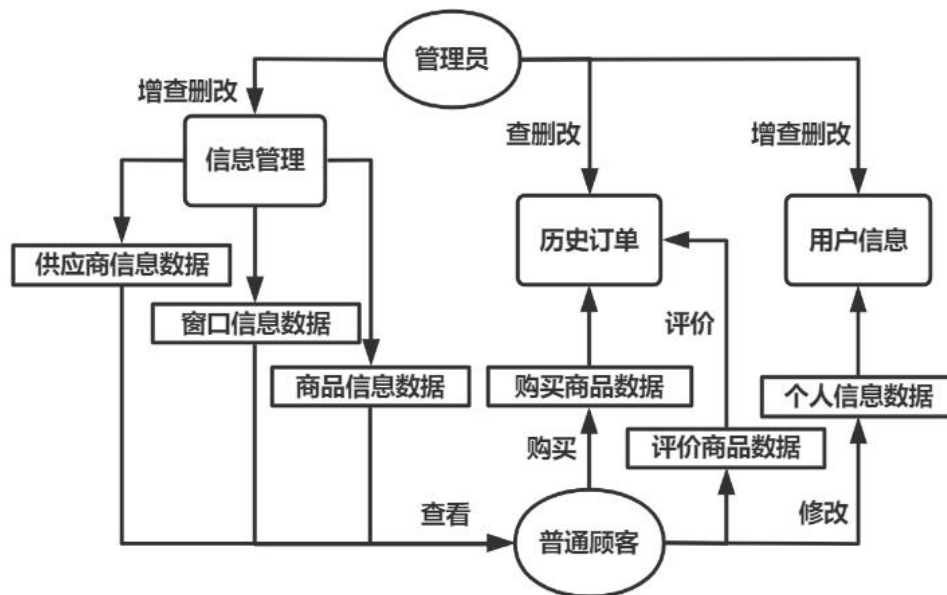


图 1：数据流图

1.1.3 数据字典

• 主要的数据结构定义：

1. 供应商(provider)
 - 含义：食堂每个窗口的食材供应商，与窗口存在一对多的关系
 - 输入输出：供应商信息
 - 数据描述：供应商号，名称，地址，电话
 - 数据量：取决于食堂供应商个数
 - 存取方式：联机处理，随机检索，查找与更新
2. 窗口(window)
 - 含义：食堂以窗口为单位供应商品，每个窗口供应多样商品，有唯一的供应商
 - 输入输出：窗口信息
 - 数据描述：窗口号，名称，位置，平均价格，评分，月销量，供应商号
 - 数据量：取决于食堂窗口个数
 - 存取方式：联机处理，随机检索，查找与更新
3. 商品(dish)
 - 含义：食堂给顾客供应的商品，每个窗口可供应多个不同商品
 - 输入输出：商品信息
 - 数据描述：商品号，图片，名称，价格，评分，月销量，窗口号
 - 数据量：取决于食堂窗口个数和每个窗口供应的商品个数
 - 存取方式：联机处理，随机检索，查找与更新
4. 用户(user)
 - 含义：使用“星云海餐饮服务系统”的用户
 - 输入输出：用户信息
 - 数据描述：用户号，用户名，学工号，电话号码，密码
 - 数据量：取决于使用该系统的用户个数
 - 存取方式：联机处理，随机检索，查找与更新
5. 历史订单记录(purchase)
 - 含义：所有用户的购买的历史订单（包含评分）
 - 输入输出：历史订单信息
 - 数据描述：订单号，商品号，用户号，时间，评分
 - 数据量：取决于用户购买的交易次数
 - 存取方式：联机处理，随机检索，查找与更新

• 主要的数据处理过程：

1. 供应商，窗口，商品处理流：
 - 说明：由管理员对供应商，窗口，商品信息进行增删改处理
 - 来源：管理员
 - 去向：管理员 & 普通顾客

- 数据结构：供应商，窗口，商品
- 2. 历史订单处理流
 - 说明：一般情况下，由普通顾客产生历史订单，并对订单进行评价
 - 来源：普通顾客
 - 去向：普通顾客 & 管理员
 - 数据结构：历史订单记录
- 3. 用户管理处理流
 - 说明：普通顾客进行注册，可以进行修改，管理员能够进行增删改管理
 - 来源：管理员 & 普通顾客
 - 去向：管理员 & 普通顾客
 - 数据结构：用户

1.2 概念结构设计

1.2.1 实体型设计

• 供应商

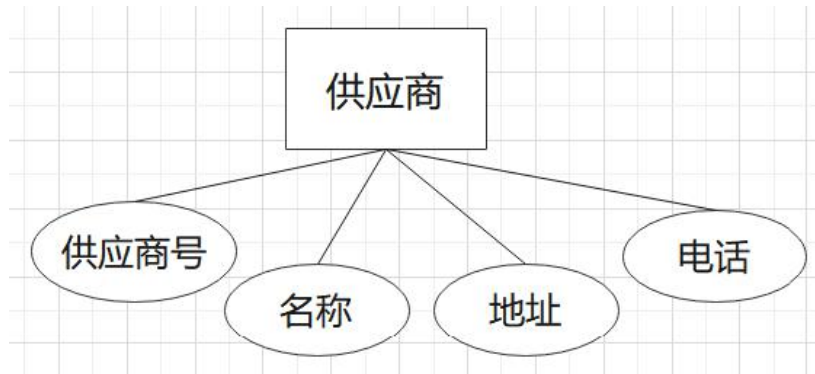


图 2：实体型：供应商

• 窗口

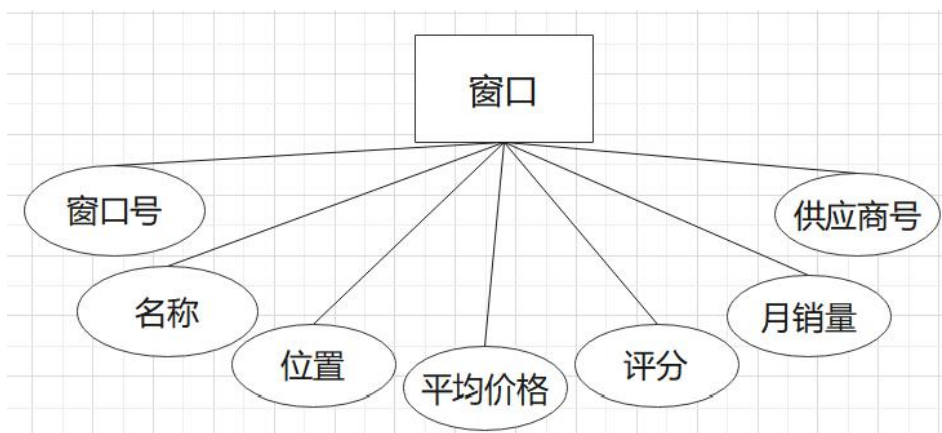


图 3：实体型：窗口

• 商品

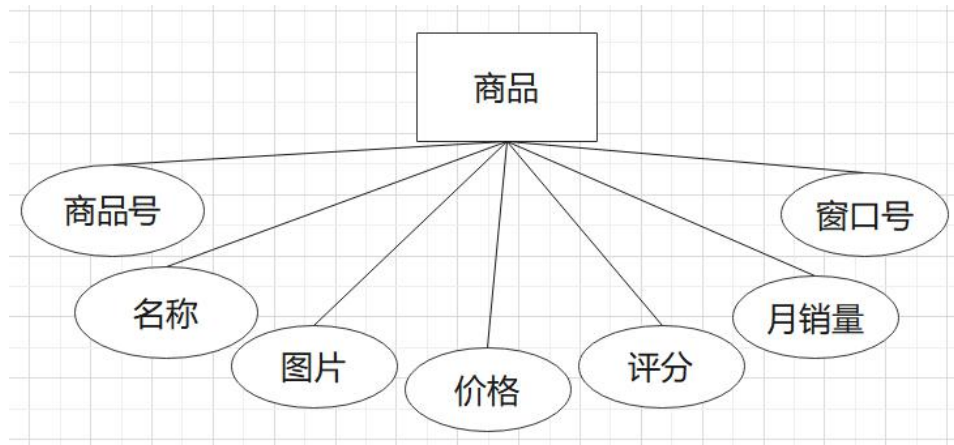


图 4：实体型：商品

• 用户

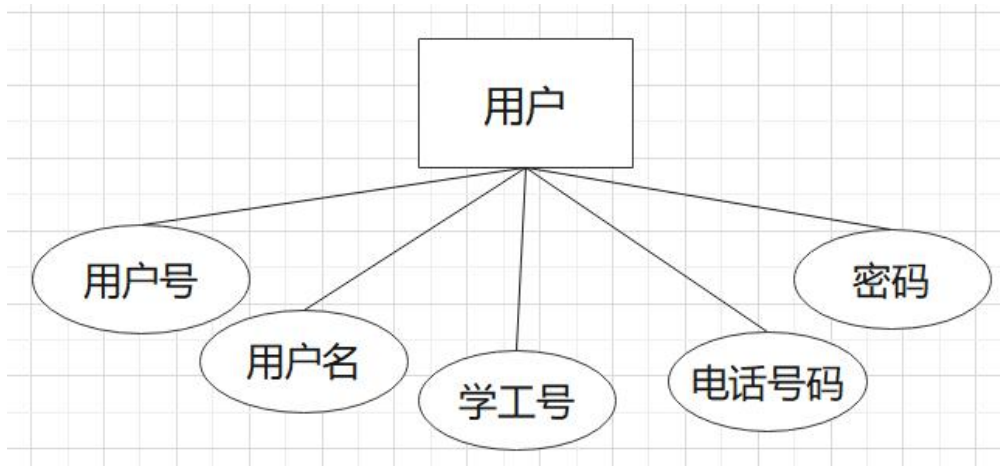


图 5：实体型：用户

1.2.2 分 ER 图(省去实体属性)

• 供应商与窗口

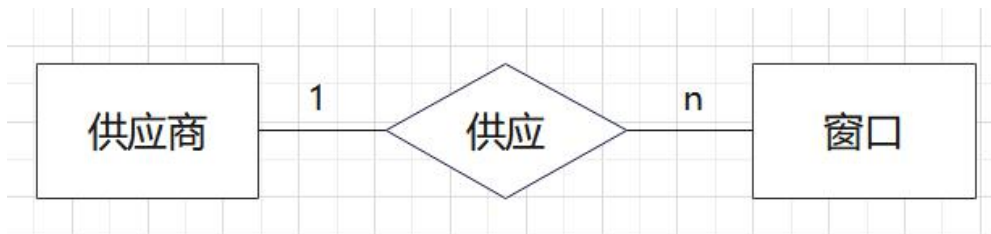


图 6：供应商与窗口：供应

• 窗口与商品

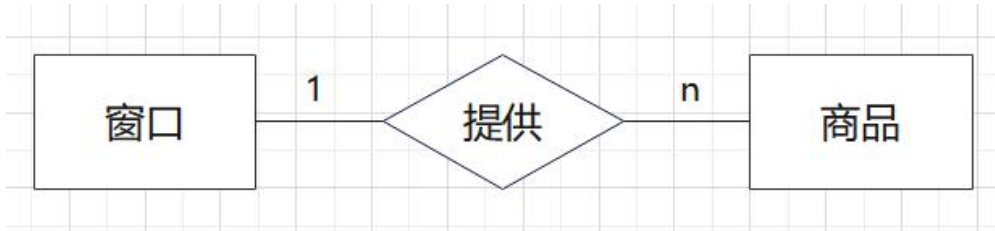


图 7: 窗口与商品: 提供

• 用户与商品

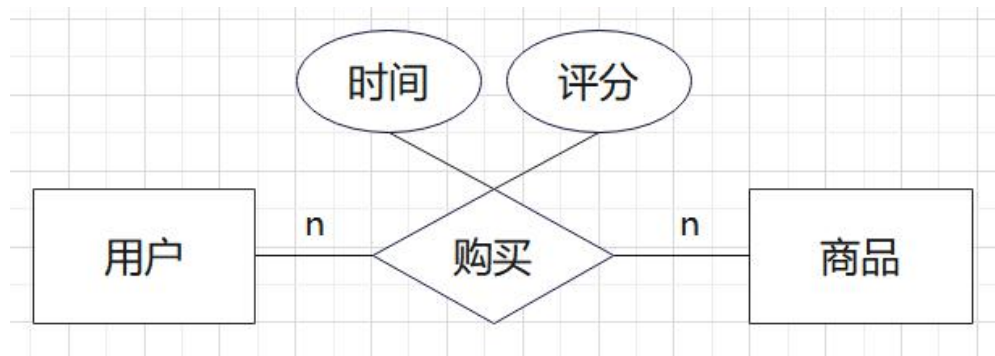


图 8: 用户与商品: 购买

1.2.3 总 ER 图(省去实体属性)

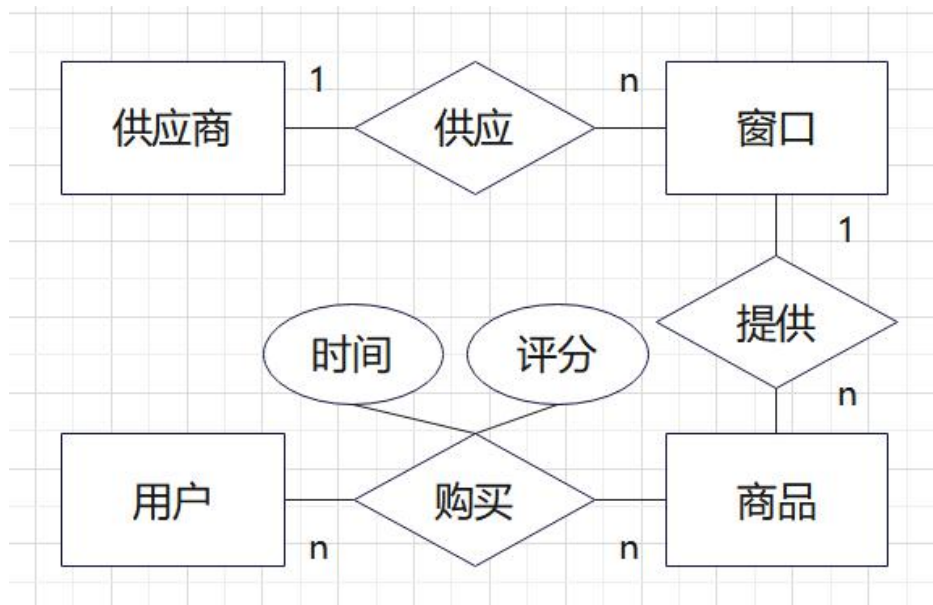


图 9: 总 ER 图

1.3 逻辑结构设计

1.3.1 关系模式定义与范式分析

- 数据库名: catering
- 供应商: provider
 - 供应商(供应商号, 名称, 地址, 电话)
 - 主键: 供应商号
 - 外键: 无
 - 所属范式: BCNF
 - 分析: 在 1NF 基础上, (供应商号) 为唯一候选码, 由于是单属性码, 所以不存在非主属性对码的部分函数依赖, 且没有非主属性对码的传递依赖, 所以属于 3NF。又每一个决定关系都包含码, 所以属于 BCNF。
- 窗口: window
 - 窗口(窗口号, 名称, 位置, 平均价格, 评分, 月销量, 供应商号)
 - 主键: 窗口号
 - 外键: 供应商号
 - 所属范式: BCNF
 - 分析: 在 1NF 基础上, (窗口号) 为唯一候选码, 由于是单属性码, 所以不存在非主属性对码的部分函数依赖, 且没有非主属性对码的传递依赖, 所以属于 3NF。又每一个决定关系都包含码, 所以属于 BCNF。
- 商品: dish
 - 商品(商品号, 名称, 价格, 评分, 月销量, 窗口号)
 - 主键: 商品号
 - 外键: 窗口号
 - 所属范式: BCNF
 - 分析: 在 1NF 基础上, (商品号) 为唯一候选码, 由于是单属性码, 所以不存在非主属性对码的部分函数依赖, 且没有非主属性对码的传递依赖, 所以属于 3NF。又每一个决定关系都包含码, 所以属于 BCNF。
- 用户: user
 - 用户(用户号, 用户名, 学工号, 电话号码, 密码)
 - 主键: 用户号
 - 外键: 无
 - 所属范式: BCNF
 - 分析: 在 1NF 基础上, (用户号) 为唯一候选码, 由于是单属性码, 所以不存在非主属性对码的部分函数依赖, 且没有非主属性对码的传递依赖, 所以属于 3NF。又每一个决定关系都包含码, 所以属于 BCNF。
- 历史订单记录: purchase

- 历史订单记录(订单号, 商品号, 用户号, 时间, 评分)
- 主键: 订单号
- 外键: 商品号, 用户号
- 所属范式: BCNF
- 分析: 在 1NF 基础上, (订单号) 为候选码, (商品号, 用户号, 时间) 也是候选码。不存在非主属性对码的部分函数依赖, 且没有非主属性对码的传递依赖, 所以属于 3NF。又每一个决定关系都包含码, 所以属于 BCNF。
- 图片表: image
 - 图片(图片号, 图片路径, 图片标签, 所属商品号)
 - 主键: 图片号
 - 外键: 所属商品号
 - 所属范式: BCNF
 - 分析: 在 1NF 基础上, (图片号) 为唯一候选码, 由于是单属性码, 所以不存在非主属性对码的部分函数依赖, 且没有非主属性对码的传递依赖, 所以属于 3NF。又每一个决定关系都包含码, 所以属于 BCNF。

1.3.2 数据库中表的定义

• provider

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
address	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
telephone	VARCHAR(12)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

图 10: 供应商表

• window

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
place	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
avg_price	DECIMAL(5,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
avg_star	DECIMAL(5,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
mon_sale	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
pro_id_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

图 11: 窗口表

• dish

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
price	DECIMAL(5,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
avg_star	DECIMAL(5,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
mon_sale	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
win_id_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

图 12: 商品表

(注: 图片信息由于其特殊性, 单独存放)

• user

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
last_time	DATETIME(6)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
number	VARCHAR(12)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
telephone	VARCHAR(12)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

图 13：用户表

• purchase

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
time	DATETIME(6)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
star	DECIMAL(5,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
dish_id_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
user_id_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

图 14：历史订单表

• image *

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
img_address	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
img_label	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dish_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

图 15：图片表

（注：存放 dish 商品表中的图片信息）

1.4 关于视图、约束、存储过程、触发器与索引

1.4.1 视图

为了保证数据的安全性以及系统运行的正常，需要对于不同用户定义相应的视图。

- 对于每个普通顾客，只能够看到与自己相关的历史订单记录
- 对于每个普通顾客，只能够看到自己的个人信息
- 对于管理员，能够看到所有用户的历史订单记录和用户信息
- 首页具备统计功能，能够筛选出销量最高的 10 种商品

1.4.2 约束

这里阐明一些在逻辑结构设计中定义的元组和属性约束外的一些约束：

- 供应商是窗口的外键，存在外键约束，在添加和修改窗口信息时，需要考虑该约束
- 窗口是商品的外键，存在外键约束，在添加和修改商品信息时，需

要考虑该约束

- 商品和用户是历史订单的外键，存在外键约束。
- 某商品的“评分”字段是对于历史订单记录中所有相关“评分”字段的取平均值
- 某商品的“月销量”字段是对于历史订单记录中所有相关数据的求数量
- 某窗口的“平均价格”字段是对于该窗口所有商品的价格求平均值
- 某窗口的“平均评分”字段是对于该窗口所有商品的评分求平均值
- 某窗口的“月销量”字段是对于该窗口所有商品的月销量求和

1.4.3 存储过程

为了满足上述约束，需要在更新数据的同时，对上述“评分”“月销量”“平均价格”字段进行自动计算与更新。为了能够简化代码，首先定义相关计算的存储过程，然后在触发器中调用存储过程进行更新。

1. 计算商品评分

```
1 DELIMITER $$
2 create procedure update_dish_star (in input_id int)
3 begin
4     declare sale int default 0;
5     declare total_star float default 0.0;
6     declare avg_star float default 0.0;
7     select count(id),sum(star) into sale,total_star from catering_purchase
8     where dish_id_id = input_id and star;
9     if sale = 0 then set avg_star = NULL;
10    else set avg_star = total_star/sale;
11    end if;
12    update catering_dish set avg_star = avg_star where id = input_id;
13 end$$
14 DELIMITER ;
```

2. 计算商品月销量

```
1 DELIMITER $$
2 • create procedure update_dish_sale (in input_id int)
3 begin
4     declare sale int default 0;
5     select count(id) into sale from catering_purchase
6     where dish_id_id = input_id;
7     update catering_dish set mon_sale = sale where id = input_id;
8 end$$
9 DELIMITER ;
```

3. 计算窗口平均价格

```
1 DELIMITER $$
2 create procedure update_window_price (in input_id int)
3 begin
4 declare num int default 0;
5 declare total_price float default 0.0;
6 declare avg_price float default 0.0;
7 select count(id),sum(price) into num,total_price from catering_dish
8 where win_id_id = input_id and price;
9 if num = 0 then set avg_price = NULL;
10 else set avg_price = total_price/num;
11 end if;
12 update catering_window set avg_price = avg_price where id = input_id;
13 end$$
14 DELIMITER ;
```

4. 计算窗口平均评分

```
1 DELIMITER $$
2 create procedure update_window_star (in input_id int)
3 begin
4 declare num int default 0;
5 declare total_star float default 0.0;
6 declare avg_star float default 0.0;
7 select count(catering_purchase.id),sum(catering_purchase.star)
8 into num,total_star from catering_dish,catering_purchase
9 where catering_dish.win_id_id = input_id
10 and catering_dish.id = catering_purchase.dish_id_id
11 and catering_purchase.star;
12 if num = 0 then set avg_star = NULL;
13 else set avg_star = total_star/num;
14 end if;
15 update catering_window set avg_star = avg_star where id = input_id;
16 end$$
17 DELIMITER ;
```

5. 计算窗口月销量

```
1 DELIMITER $$
2 create procedure update_window_sale (in input_id int)
3 begin
4 declare sale int default 0;
5 select sum(mon_sale) into sale from catering_dish
6 where win_id_id = input_id;
7 update catering_window set mon_sale = sale where id = input_id;
8 end$$
9 DELIMITER ;
```

1.4.4 触发器

在对历史订单记录进行插入，修改，删除时，调用存储函数对商品表的相关字段进行更新：

1. 插入触发器：

```
1 DELIMITER $$
2 • create trigger insert_on_purchase after insert on catering_purchase
3 for each row
4 begin
5 call update_dish_sale(new.dish_id_id);
6 call update_dish_star(new.dish_id_id);
7 end $$
8 DELIMITER ;
```

2. 更新触发器：

```
1 DELIMITER $$
2 • create trigger update_on_purchase after update on catering_purchase
3 for each row
4 begin
5 call update_dish_star(new.dish_id_id);
6 end $$
7 DELIMITER ;
```


3. 删除触发器:

```
1 DELIMITER $$
2 • create trigger delete_on_purchase after delete on catering_purchase
3   for each row
4   begin
5     call update_dish_sale(old.dish_id_id);
6     call update_dish_star(old.dish_id_id);
7   end $$
8 DELIMITER ;
```

在对商品表进行插入, 修改和删除时, 调用存储函数对窗口表的相关字段进行更新:

1. 插入触发器:

```
1 DELIMITER $$
2 • create trigger insert_on_dish after insert on catering_dish
3   for each row
4   begin
5     call update_window_price(new.win_id_id);
6     call update_window_sale(new.win_id_id);
7     call update_window_star(new.win_id_id);
8   end $$
9 DELIMITER ;
10
```

2. 更新触发器:

```
1 DELIMITER $$
2 • create trigger update_on_dish after update on catering_dish
3   for each row
4   begin
5     call update_window_price(new.win_id_id);
6     call update_window_sale(new.win_id_id);
7     call update_window_star(new.win_id_id);
8   end $$
9 DELIMITER ;
```

3. 删除触发器：

```
1 DELIMITER $$
2 • create trigger delete_on_dish after delete on catering_dish
3   for each row
4   begin
5     call update_window_price(old.win_id_id);
6     call update_window_sale(old.win_id_id);
7     call update_window_star(old.win_id_id);
8   end $$
9 DELIMITER ;
```

(注：这里构成了二级级联触发器，对历史订单表的更新将引起商品表的更新，对商品表的更新将引起窗口表的更新)

1.4.5 索引

- 在所有表的主键上建立聚簇索引。
- 在查询商品时，往往要查询它所在的窗口和窗口位置，例如：
“香酥鸡腿饭 江苏风味 云天苑二楼餐厅”
因此，可以在窗口名称和窗口位置上建立二级索引。

2. 实验成果展示

2.1 主界面

2.1.1 管理员主界面



图 16：管理员主界面

2.1.2 普通顾客主界面



图 17：普通顾客主界面

2.2 功能界面示例

2.2.1 Provider



图 18：供应商列表界面

2.2.2 Dish

商品图片	商品名称	窗口名称	窗口位置	商品价格	商品评分	商品月销量	商品操作
	香酥鸡腿饭	江苏风味	云天苑二楼	12.00	4.80	4	测试
	秘制鸡排饭	陕西风味	云天苑二楼	11.00	4.40	1	测试
	肉夹馍	陕西风味	云天苑二楼	8.80	4.90	3	测试
	凉皮	陕西风味	云天苑二楼	7.00	4.37	4	测试
	香菇炖鸡面	陕西风味	云天苑二楼	11.00	3.00	1	测试

图 19：商品列表界面

2.2.3 Purchase

商品名	用户名	购买时间	订单评分	订单操作
香酥鸡腿饭	王民航	Dec. 18, 2021, 12:09 p.m.	None	取消 删除
肉夹馍	申世东	Dec. 18, 2021, 12:24 p.m.	4.80	取消 删除
凉皮	江惠熙	Dec. 18, 2021, 12:29 p.m.	4.00	取消 删除
肉夹馍	江惠熙	Dec. 18, 2021, 12:29 p.m.	5.00	取消 删除
香酥鸡腿饭	abc	Dec. 18, 2021, 12:31 p.m.	None	取消 删除
秘制鸡排饭	abc	Dec. 18, 2021, 12:31 p.m.	4.40	取消 删除
凉皮	abc	Dec. 18, 2021, 12:31 p.m.	None	取消 删除
肉夹馍	abc	Dec. 18, 2021, 12:31 p.m.	None	取消 删除

图 20：历史订单列表界面

2.2.4 Search

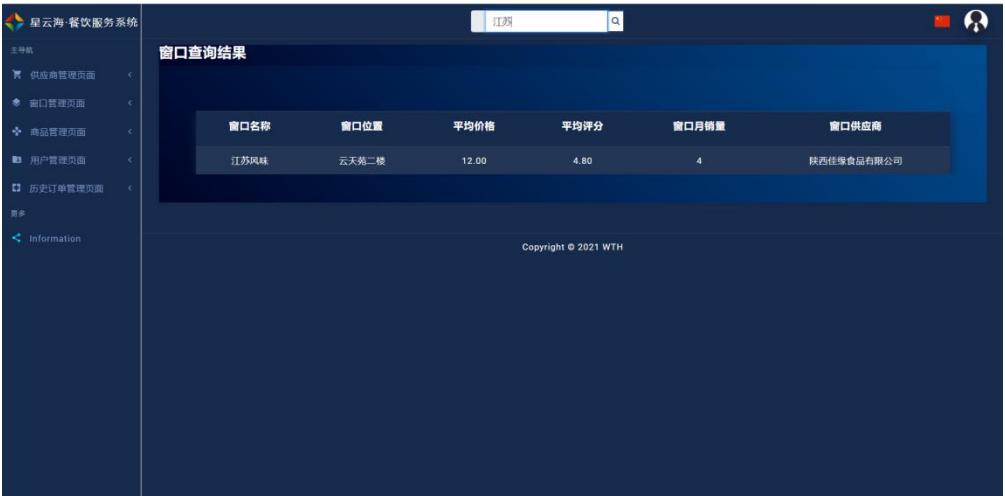


图 21：搜索结果界面

2.2.5 Login

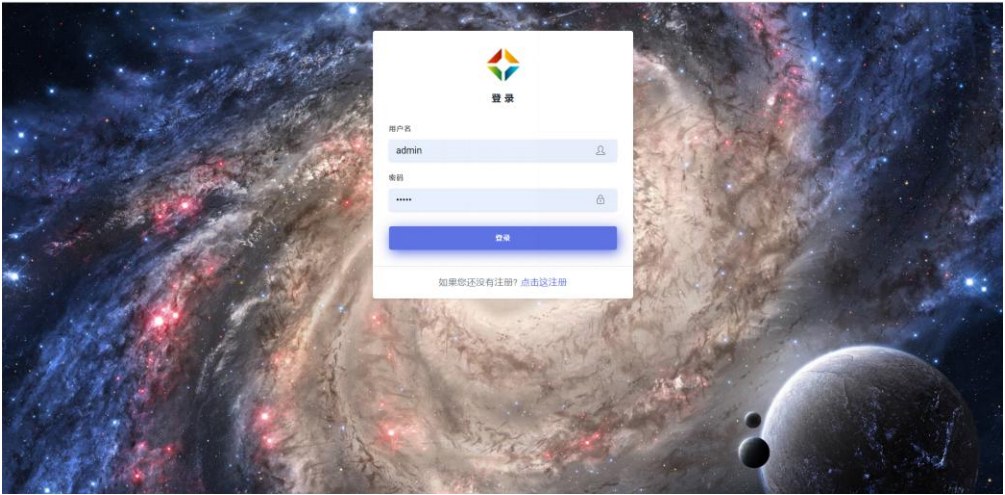


图 22：登录界面

3. 遇到的问题和解决方法

3.1 问题 1

电话号码不能使用 integer 类型存储，使用 char 更加合适。

3.2 问题 2

在 MySQL 中，任何与 NULL 进行算数逻辑操作的结果均为 NULL，不能使用 “=NULL” 去判断一个值是否为空，而应该直接 “与” 上该数据。

3.3 问题 3

尝试将项目部署在服务器上：将 Django 的 IP 和端口号设置为 0.0.0.0:8000。但本地浏览器无法访问网站。发现是服务器防火墙导致的，需要对防火墙进行配置。如图所示，在防火墙中添加如下规则：

我的轻量 
 39.99.81.248 | Ubuntu 18.04 | 呼和浩特 | ● 运行中

防火墙

应用类型	协议	端口范围	备注
全部TCP	TCP	1/65535	全部TCP

在启动后，出现访问错误：

```
DisallowedHost at / Invalid HTTP_HOST header: DisallowedHost at /
```

解决方法是修改 settings.py 文件，修改为：ALLOWED_HOSTS=['*']，其中*表示允许所有的主机访问。

4. 开发环境说明

4.1 开发系统

本程序开发的系统环境为 Windows 10。

4.2 开发语言

本程序基于 python3.8 版本。

4.3 开发平台

使用 PyCharm 搭建工程，调用 PyMysql 库与 Mysql 数据库进行通信交互，前后端基于 Django 框架进行搭建。

4.4 数据库版本

使用的数据库版本：Mysql v8.0.21。