

## 彬彬外卖平台程序设计文档

1800094804 张峻优

## 1. 程序说明

本程序名称为“彬彬外卖平台”，是基于 MySQL 8.0 设计的数据库和 PHP+Apache 客户端方式的外卖管理系统程序。用户可以以客户，商户，骑手，管理员的身份注册并登录使用本系统。根据用户类别，用户可在本程序上使用的功能如下：

1. **顾客：**顾客可以登录并修改自己的部分基本信息，查看现在营业中的商家列表并从商店选择菜品，加入到购物车。加入购物车后可以修改数量，确认信息后下单。下单后如果商家还没有接单，用户方可以取消订单。用户也可以查询自己的历史订单。
2. **商家：**商家可以登录并修改商家部分基本信息，查询自己商家的菜单，增删菜品，查询菜品信息，销售量等。商家还可以查询用户给自己商家下的订单，并选择接单。
3. **骑手：**骑手可以登录并修改骑手部分基本信息，如果核酸符合要求，可以接受商家已接的订单。之后可以更新订单状态为已取餐，配送中和订单完成。骑手也可以查询自己的历史订单。
4. **管理员：**管理员拥有数据库所有权限，可在程序中对客户，商家，商家菜品，骑手进行增减（增加功能由于存在注册功能没放在管理员页面），对系统中的区域进行疫情信息调整。管理员也可以在页面进行各种查询。

## 2. 数据库设计说明

本数据库在 MySQL 内命名为 waimaidb，由若干表组成。下面将对只看名字不太明确的 field 做出附加说明。

## a) 表设计

（一）customer 表

```
mysql> describe customer;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
dob	date	NO		NULL	
sex	enum('MALE', 'FEMALE')	NO		NULL	
phone	char(11)	NO		NULL	
email	varchar(256)	YES		NULL	
username	varchar(15)	NO	UNI	NULL	
password	varchar(20)	NO		NULL	
id_num	varchar(18)	NO	UNI	NULL	
delivery_addr	varchar(500)	NO		NULL	

10 rows in set (0.05 sec)

customer 表由上述 field 组成。它存储顾客的基本信息。email 为可选信息，可以为 null。

## (二) merchant 表

```
mysql> describe merchant;
```

Field	Type	Null	Key	Default	Extra
merchant_id	int	NO	PRI	NULL	auto_increment
merchant_name	varchar(45)	NO		NULL	
merchant_addr	varchar(500)	NO		NULL	
phone	char(11)	NO		NULL	
username	varchar(15)	NO		NULL	
password	varchar(20)	NO		NULL	
delivery_coverage	varchar(10)	NO		NULL	
is_open	enum('YES','NO')	NO		YES	
menu_id	int	YES	UNI	NULL	
area_id	int	NO	MUL	0	

10 rows in set (0.00 sec)

merchant 表结构如上，它存储商家基本信息。Area 表示商家所在的区域。商家编号 menu\_id 设置成可以为 null 是因为客户端创建 order 时会先空着 menu\_id 进行插入，之后会运行查询保证 merchant\_id 与菜单编号 menu\_id 会被设置为相同的数。area\_id 为外码，参考 area 表的 area\_id 键，有着 ON UPDATE RESTRICT 和 ON DELETE RESTRICT 来保证引用完整性。

## (三) rider 表

```
mysql> describe rider;
```

Field	Type	Null	Key	Default	Extra
rider_id	int	NO	PRI	NULL	auto_increment
rider_name	varchar(50)	NO		NULL	
phone	char(11)	NO		NULL	
joinTime	datetime	NO		NULL	
homeAddr	varchar(500)	NO		NULL	
temp	decimal(3,1)	NO		36.0	
pcr	enum('POSITIVE','NEGATIVE')	NO		NEGATIVE	
username	varchar(15)	NO		NULL	
password	varchar(20)	NO		NULL	
pcr_time	datetime	YES		NULL	
service_status	enum('ACTIVE','SUSPENDED')	YES		ACTIVE	

11 rows in set (0.00 sec)

rider 表结构如上，它存储骑手的基本信息。service\_status 表示骑手是否能正常工作并接单，用于触发器判辩是否处于异常状态。Pcr\_time 会随着骑手在个人页面更新核酸结果时间更新。刚注册时，pcr 结果会默认为阴性。

## (四) login 表

```
mysql> describe login;
```

Field	Type	Null	Key	Default	Extra
username	varchar(15)	NO	PRI	NULL	
password	varchar(20)	NO		NULL	
type	enum('CUSTOMER','MERCHANT','RIDER','ADMIN')	NO		NULL	

3 rows in set (0.00 sec)

login 表结构如上，存储用户登录信息（用户名，密码，用户类别）。

## (五) orders 表

```
mysql> describe orders;
```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	auto_increment
order_time	datetime	NO		NULL	
customer_id	int	NO	MUL	NULL	
phone	varchar(11)	NO		NULL	
total_price	int	NO		NULL	
merchant_id	int	NO	MUL	NULL	
rider_id	int	YES	MUL	NULL	
delivery_addr	varchar(500)	NO		NULL	
order_status	enum('ORDERED','YIJIEDAN','QISHOUDAODIAN','PEISONGZHONG','DONE','CANCELED')	NO		ORDERED	
updated_at	datetime	YES		NULL	

10 rows in set (0.00 sec)

orders 表结构如上，存储每个订单信息。Order\_status 的每个状态表示为：‘ORDERED’（顾客已下单，商家还没有接单，此时顾客可以取消订单），‘YIJIEDAN’（商家已接单），‘QISHOUDAODIAN’（骑手已接单并到店铺了），‘PEISONGZHONG’（骑手已取到餐，正在配送中），‘DONE’（订单已完成），‘CANCELED’（用户已取消订单）。customer\_id, merchant\_id, rider\_id 均为外码，参考 customer, merchant 和 rider 表的相应 id 键，都有着 ON UPDATE RESTRICT 和 ON DELETE RESTRICT 来保证引用完整性。updated\_at 为订单送达时，订单送达的时间，用来以后的查询。

## (六) order\_menu 表

```
mysql> describe order_menu;
```

Field	Type	Null	Key	Default	Extra
ordermenu_id	int	NO	PRI	NULL	auto_increment
order_id	int	YES	MUL	NULL	
food_name	varchar(45)	NO		NULL	
food_price	int	NO		NULL	
food_id	int	NO		NULL	
quantity	int	NO		NULL	

6 rows in set (0.00 sec)

order\_menu 表结构如上，存储每个订单内的菜品和数量信息。order\_menu 都会有相应的 order\_id 匹配。order\_id 为外码，参考 orders 表的 order\_id 键，有着 ON UPDATE RESTRICT 和 ON DELETE RESTRICT 来保证引用完整性。

## (七) cart 表

```
mysql> describe cart;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
food_id	int	NO	PRI	NULL	
food_name	varchar(45)	NO		NULL	
food_price	int	NO		NULL	
merchant_id	int	NO	MUL	NULL	
merchant_name	varchar(45)	NO		NULL	
quantity	int	NO		1	

7 rows in set (0.00 sec)

cart 表结构如上，存储每个用户放在购物车的菜品信息。customer\_id, food\_id, merchant\_id 均为外码，参考 customer, food, merchant 表的相应 id 键，都有着 ON UPDATE RESTRICT 和 ON DELETE RESTRICT 来保证引用完整性。默认菜品数量为 1。

## (八) food 表

```
mysql> describe food;
```

Field	Type	Null	Key	Default	Extra
food_id	int	NO	PRI	NULL	auto_increment
merchant_id	int	NO	MUL	NULL	
food_name	varchar(45)	NO		NULL	
food_price	int	NO		NULL	
ordered_count	int	NO		0	

5 rows in set (0.00 sec)

food 表结构如上，存储每个菜品信息。Ordered\_count 表示这道菜被用户点的次数，默认为 1。merchant\_id 为外码，参考 merchant 表的 merchant\_id 键，有着 ON UPDATE RESTRICT 和 ON DELETE RESTRICT 来保证引用完整性。

## (九) alert\_table 表

```
mysql> describe alert_table;
```

Field	Type	Null	Key	Default	Extra
alert_id	int	NO	PRI	NULL	auto_increment
rider_id	int	NO		NULL	
alert_type	varchar(200)	NO		NULL	
alert_time	datetime	NO		NULL	
resolved	tinyint	NO		0	

5 rows in set (0.00 sec)

alert\_table 表结构如上，存储对骑手通过触发器需要通知内容的信息。Resolved 表示这通知的问题是否已被解决。

## (十) area 表

```
mysql> describe area;
```

Field	Type	Null	Key	Default	Extra
area_id	int	NO	PRI	NULL	
covid_stat	enum('YES', 'NO')	NO		NO	

2 rows in set (0.00 sec)

area 表结构如上，存储系统内商家所在区域的疫情信息。

## b) 触发器设计

```
CREATE DEFINER='root'@'localhost' TRIGGER `alert_trigger` AFTER UPDATE ON `rider` FOR EACH ROW BEGIN
  IF NEW.pcr = 'POSITIVE' OR NEW.temp > 37.3 THEN
    INSERT INTO alert_table (rider_id, alert_type, alert_time)
    VALUES (NEW.rider_id, 'You are under high danger of COVID-19 Infection, Please Complete Stian 3jian!', NOW());
  END IF;
  IF NEW.pcr = 'NEGATIVE' AND NEW.temp < 37.3 THEN
    UPDATE alert_table SET resolved = 1 WHERE rider_id = NEW.rider_id AND resolved = 0;
  END IF;
END
```

以上触发器检测配送员感染新冠的风险，若配送员核算结果为阳性或者体温为 37.3 度以上，则触发器会在 alert\_table 表上插入针对有风险的骑手编号的提醒，这提醒会在配送员的主页上显示。

## c) 用户权限设计

```

> Execute
create user 'dba'@'localhost' identified by 'dba';
> Execute
create user 'rider'@'localhost' identified by 'rider';
> Execute
create user 'merchant'@'localhost' identified by 'merchant';
> Execute
create user 'customer'@'localhost' identified by 'customer';

```

数据库内有四种用户，分别是 dba（管理员），rider（骑手），merchant（商家），customer（客户）。密码与用户名相同。下面介绍每个用户权限中的重点信息。

```

> Execute
GRANT SELECT, INSERT, UPDATE, DELETE ON `waimaodb`.`cart` TO `customer`@`localhost`;
> Execute
GRANT SELECT, INSERT, UPDATE (`delivery_addr`, `email`, `password`, `phone`) ON `waimaodb`.`customer` TO `customer`@`localhost`;
> Execute
GRANT SELECT, UPDATE (`ordered_count`) ON `waimaodb`.`food` TO `customer`@`localhost`;
> Execute
GRANT SELECT (`username`), INSERT, UPDATE (`password`) ON `waimaodb`.`login` TO `customer`@`localhost`;
> Execute
GRANT SELECT ON `waimaodb`.`merchant` TO `customer`@`localhost`;
> Execute
GRANT SELECT, INSERT, UPDATE ON `waimaodb`.`order_menu` TO `customer`@`localhost`;
> Execute
GRANT SELECT, INSERT, UPDATE, UPDATE (`order_status`, `ordermenu_id`) ON `waimaodb`.`orders` TO `customer`@`localhost`;
> Execute
GRANT SELECT ON `waimaodb`.`rider` TO `customer`@`localhost`;

```

**客户：**不可修改自己的顾客编号，登录名，姓名，性别，身份证号。

```

> Execute
GRANT SELECT (`username`), INSERT, UPDATE (`password`) ON `waimaodb`.`login` TO `merchant`@`localhost`;
> Execute
GRANT SELECT, INSERT, UPDATE, DELETE ON `waimaodb`.`food` TO `merchant`@`localhost`;
> Execute
GRANT SELECT (`username`), INSERT, UPDATE (`password`) ON `waimaodb`.`login` TO `merchant`@`localhost`;
> Execute
GRANT SELECT, INSERT, INSERT (`area`),
UPDATE (`area`, `area_id`, `delivery_coverage`, `is_open`, `menu_id`, `merchant_addr`, `merchant_name`, `password`, `phone`)
ON `waimaodb`.`merchant` TO `merchant`@`localhost`;
> Execute
GRANT SELECT ON `waimaodb`.`order_menu` TO `merchant`@`localhost`;
> Execute
GRANT SELECT, UPDATE ON `waimaodb`.`orders` TO `merchant`@`localhost`;

```

**商家：**不可修改登录名，商家编号。客户端上无法修改自己的菜单编号（menu\_id），但是权限上允许了，这是为了保证每次插入新商户之后更新菜单编号和商家编号相同。

```

> Execute
GRANT SELECT, UPDATE ON `waimaodb`.`alert_table` TO `rider`@`localhost`;
> Execute
GRANT SELECT, INSERT, UPDATE (`password`) ON `waimaodb`.`login` TO `rider`@`localhost`;
> Execute
GRANT SELECT ON `waimaodb`.`order_menu` TO `rider`@`localhost`;
> Execute
GRANT SELECT, UPDATE, UPDATE (`updated_at`) ON `waimaodb`.`orders` TO `rider`@`localhost`;
> Execute
GRANT SELECT, INSERT, UPDATE (`phone`, `homeAddr`, `temp`, `pcr`, `password`, `pcr_time`, `service_status`) ON `waimaodb`.`rider` TO `rider`@`localhost`;

```

**骑手：**不可修改自己的登录名，骑手编号，骑手名和入职时间。只能更新订单状态。

```
GRANT ALL PRIVILEGES ON `waimaodb`.* TO `dba`@`localhost`;
```

**管理员：**本程序只有一个用户名和密码为 admin1 的拥有管理者身份的用户。拥有数据库所有权限。

### 3. 程序设计说明

Apache 服务器下的网页文件结构如下：

```

C:\BITNAMI\WAMPSTACK-8.1.12-0\APACHE2\HTDOCS
|
| bitnami.css
| favicon.ico
| index.php
| info_update.php
| readme.md
|
+---create_account_waimai
|   add_customer.php
|   add_merchant.php
|   add_rider.php
|   create_customer_page.php
|   create_merchant_page.php
|   create_rider_page.php
|
+---customer_page
|   add_cart.php
|   browse_cart.php
|   browse_menu.php
|   browse_orders.php
|   browse_stores.php
|   cancelOrder_page.php
|   customer_page.php
|   delete_cart.php
|   order_details.php
|   place_order.php
|   update_quantity.php
|
+---dba_page
|   add_menu.php
|   browse_customer.php
|   browse_rider.php
|   browse_stores.php
|   change_area_covidstat.php
|   dba_page.php
|   delete_customer.php
|   delete_menu.php
|   delete_rider.php
|   delete_store.php
|   edit_menu.php
|   update_area.php
|
\---sql
|   sql1.php
|   sql2.php
|   sql3.php
|   sql4.php
|   sql5.php
|   sql6.php
|
+---login
|   login.php
|
+---merchant_page
|   add_menu.php
|   browse_orders.php
|   delete_menu.php
|   edit_menu.php
|   jiedan_page.php
|   merchant_page.php
|   order_details.php
|
+---rider_page
|   browse_kejie_orders.php
|   browse_rider_orders.php
|   jiedan_page.php
|   ordercomplete_page.php
|   order_details.php
|   peisongstart_page.php
|   rider_page.php

```

下面将说明每个文件夹内的文件的功能说明。

root:

- **index.php**: 主页，用于用户登录，注册（顾客，商家，骑手）。
- **info\_update.php**: 用于用户信息更新，包括相应 SQL 查询。

create\_account\_waimai:

- **add\_<客户类别>.php**: 用于增加用户到相应的表（customer/merchant/rider），包括相应 SQL 查询。
- **create\_<客户类别>\_page**: 用户注册页面。此页面填写的信息会转到 add\_<客户类别>.php，并完成注册。

customer\_page:

- **add\_cart.php**: 用于增加菜品到用户的购物车中，包括相应 SQL 查询。
- **browse\_cart.php**: 用于浏览用户的购物车。用户可以查询自己购物车里的菜品和信息，如果购物车里的菜品都来自一个商家，用户可以确认总价格后选择下单。
- **browse\_menu.php**: 用于浏览用户选择的商家的菜品。用户可以看到菜品编号，菜名，单价等信息，并选择把菜品加入到购物车。
- **browse\_orders.php**: 用于浏览用户下过的订单。用户可以看到订单编号，商家编号，下单时间，订单总价和订单状态。用户可以查询订单详情，如果订单处于‘ORDERED’状态（用户已下单，商家未接单），用户可以选择取消订单。
- **browse\_stores.php**: 用于浏览目前营业中，可以下单的商家。用户可以看到商家名，商家编号，商家地址和电话，并选择浏览选择的商家的菜单。
- **cancelOrder\_page.php**: 用于顾客取消订单，包括相应 SQL 查询来改变订单状态。
- **customer\_page.php**: 顾客主页面，页面中可以修改顾客的部分基本信息。可以选择浏览商店/菜单，购物车管理/下单，查询订单信息。

- **delete\_cart.php:** 用于顾客删除购物车里的菜品，包括相应 SQL 查询。
- **order\_details.php:** 用于顾客查询订单详情，顾客可以看到订单内的菜品，菜品编号，单价和数量。
- **place\_order.php:** 用于顾客下单。下单后可看到本次订单编号。包括相应 SQL 查询。
- **update\_quantity.php:** 用于顾客在购物车修改菜品数量。包括相应 SQL 查询来更新数量。

#### dba\_page:

- **add\_menu.php:** 与 merchant\_page 的 add\_menu.php 相同，用于增加商家菜品，包括相应 SQL 查询。
- **browse\_<customer/rider/stores>.php:** 用于管理员查询所有数据库内顾客/骑手/商家信息，并删除顾客信息。
- **change\_area\_covidstat.php:** 用于管理员更改系统内区域的疫情状态。本状态会影响位于此区域的所有商家的营业状态- 如果区域存在疫情，此区域的商家都会停止营业直至管理员修改状态为疫情消失。
- **dba\_page.php:** 管理员主页。页面内管理员可以更改区域疫情信息/浏览&删除用户，执行 6 种查询。
- **delete\_<customer/rider/store>.php:** 用于管理员删除选择的顾客/骑手/商家。包括相应的 SQL 查询。
- **delete\_menu.php:** 与 merchant\_page 的 delete\_menu 相同，用于删除选择的菜品，包括相应 SQL 查询。
- **edit\_menu.php:** 与 merchant\_page 的 edit\_menu.php 相同，用于浏览已有菜品并增删菜品的页面。
- **update\_area.php:** 用于更新选择区域的疫情信息，包括相应的 SQL 查询。

#### dba\_page/sql:

- **sql1.php:** 找出 2021 年 1 月 1 日之后点餐次数前十的顾客，按照点餐次数进行降序排序；若次数相同，则按照顾客姓名拼音次序排序。
- **sql2.php:** 查找顾客消费次数最多的商家的信息，包括商家的名称、顾客消费次数，顾客消费次数相同的按照订单总额排序
- **sql3.php:** 找出在 30 天内顾客消费次数少于全部商家平均接待次数的商家姓名，编号
- **sql4.php:** 找出在 14 天内体温高于 37.3 度的配送员所接触过的商家和顾客的姓名、地址。
- **sql5.php:** 找出在 7 天内的所有订单中，消费总额最少的菜品名称，若存在多种菜品的总额相同且最少，则都列出
- **sql6.php:** 找出在 30 天内订单异常次数(系统中存在顾客有效订单,但是超过 2 小时没有到达订单完成状态)超过 2 次的顾客的姓名和配送员姓名

**\*注意: 所有查询不会包括已取消的订单 (order\_status 为 'CANCELED' 的订单)**



**login:**

- **login.php:** 用于通过用户输入的用户名和密码，确认用户类别并往相应个人页面跳转。包括相应的 SQL 查询。

**merchant\_page:**

- **add\_menu.php:** 用于增加商家菜品，包括相应 SQL 查询。
- **browse\_orders.php:** 用于浏览用户给商家下过的订单。用户可以看到订单编号，商家编号，下单时间，订单总价和订单状态。商家可以查询订单详情，如果订单处于‘ORDERED’状态（用户已下单，商家未接单），商家可以选择接单。
- **delete\_menu.php:** 用于删除选择的菜品，包括相应 SQL 查询。
- **jiedan\_page.php:** 用于更改订单状态为商家已接单（‘YIJIEDAN’），包括相应 SQL 查询。
- **merchant\_page.php:** 商家主页。页面内商家可以修改基本信息，选择编辑菜单&查看菜品购买次数，查看订单。
- **order\_details.php:** 用于商家查询选择的订单详情，商家可以看到订单编号，点单用户编号，订单内的菜品，菜品编号，单价和数量。

**rider\_page:**

- **brwose\_kejie\_orders.php:** 用于骑手查询可接的订单（订单状态‘YIJIEDAN’）。骑手可以看到订单编号，商家编号，下单时间，总价格，订单状态。骑手可以选择查看订单详情，接单。
- **browse\_rider\_orders.php:** 用于骑手查询自己接过的所有订单。骑手可以查询每个订单的订单编号，商家编号，下单时间，订单总价和订单状态。进行重点订单，骑手可以更新状态为已取到餐，配送中（‘PEISONGZHONG’），配送已完（‘DONE’）。
- **jiedan\_page.php:** 用于骑手接单，包括相应 SQL 查询来更新订单状态。
- **ordercomplete\_page.php:** 用于骑手更新订单状态为已完成，包括相应 SQL 查询来更新订单状态。
- **order\_details.php:** 用于骑手查询选择的订单详情，商家可以看到订单编号，点单用户编号，订单内的菜品，菜品编号，单价和数量。
- **peisongstart\_page.php:** 用于骑手更新订单状态为开始配送，包括相应 SQL 查询来更新订单状态。
- **rider\_page.php:** 骑手主页。页面内骑手可以修改基本信息，选择查看可接订单/查看我的订单。本页面可以上传核酸结果，如果核算结果上传时间已超过 48 小时，骑手不可接单。如果核算结果为阳性，同样骑手不可接单。核酸上传时间会随着核酸结果更新一起更新。