

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN 2 MÔN HỆ CƠ SỞ DỮ LIỆU
HK241 - Lớp: L04 (Thứ 4, tiết 2 - 7, Phòng: H6 - 702)

Hệ thống quản lý ứng dụng đặt món ăn

Giảng viên hướng dẫn: Trần Thị Quế Nguyệt

Nhóm: 3

Thành viên: Nguyễn Khánh Lộc 2211927
Lê Thị Hoàng Anh 2210070
Nguyễn Minh Thành 2433043
Nguyễn Quốc Kiệt 2211762

TP Hồ Chí Minh, Tháng 10/2024



Mục lục

1 Tạo bảng và dữ liệu mẫu	2
1.1 Các câu lệnh tạo bảng và ràng buộc	2
1.2 Thêm dữ liệu	9
2 Hiệu thực ứng dụng	16
2.1 Thủ tục INSERT / UPDATE / DELETE dữ liệu vào 1 bảng dữ liệu	16
2.1.1 Thủ tục INSERT	16
2.1.2 Thủ tục UPDATE	19
2.1.3 Thủ tục DELETE	21
2.2 TRIGGER	24
2.2.1 Trigger 1	24
2.2.2 Nhóm các trigger 2	26
2.2.3 Nhóm các trigger 3	31
2.3 THỦ TỤC	34
2.3.1 Thủ tục 1	34
2.3.2 Thủ tục 2	36
2.4 HÀM	38
2.4.1 Hàm 1	38
2.4.2 Hàm 2	41
3 ỨNG DỤNG KẾT NỐI VỚI CSDL	44
3.1 Tổng quan công nghệ	44
3.2 Các màn hình demo	44
3.2.1 Màn hình 1: Đặt món ăn	44
3.2.2 Màn hình 2: Xem đơn hàng	45
3.2.3 Màn hình 3: Quản lý món ăn	47
3.2.4 Màn hình 4: Quản lý người dùng	49
3.2.5 Màn hình 5: Thống kê doanh thu	50
Bảng kê nhiệm vụ	53
Đánh giá của giáo viên	54
Phụ lục: Thiết kế ERD cập nhật mới nhất	55



1 Tạo bảng và dữ liệu mẫu

1.1 Các câu lệnh tạo bảng và ràng buộc

- **Bảng account:**

Câu lệnh tạo bảng cho bảng account:

```
1 CREATE TABLE `account` (
2     `account_id` int(10) NOT NULL AUTO_INCREMENT,
3     `user_name` varchar(255) NOT NULL,
4     `password` varchar(32) NOT NULL,
5     PRIMARY KEY (`account_id`)
6 );
```

Câu lệnh tạo ràng buộc cho bảng account:

```
1 -- ! Rang buoc ve do dai cua password phai lon hon hoac bang 8, gom ca chu va so
2 ALTER TABLE `account`
3     ADD CONSTRAINT account_check_length_password CHECK (LENGTH(password) >= 8);
4 -- ! Rang buoc ve password phai la co it nhat 1 chu in hoa, 1 chu in thuong va 1 so
5 ALTER TABLE `account`
6     ADD CONSTRAINT account_check_valid_password CHECK (password REGEXP
7         '^[a-zA-Z0-9]+$',');
```

- **Bảng restaurant:**

Câu lệnh tạo bảng cho bảng restaurant:

```
1 CREATE TABLE `restaurant` (
2     `restaurant_id` int(10) NOT NULL,
3     `restaurant_name` varchar(255) NOT NULL,
4     `restaurant_address` varchar(255) NOT NULL,
5     `restaurant_description` varchar(255),
6     CONSTRAINT fk_restaurant_account FOREIGN KEY (`restaurant_id`) REFERENCES
7         `account`(`account_id`) ON DELETE CASCADE,
8     PRIMARY KEY (`restaurant_id`)
9 );
```

- **Bảng restaurant_image:**

Câu lệnh tạo bảng cho bảng restaurant_image:

```
1 CREATE TABLE `restaurant_image` (
2     `restaurant_image_url` varchar(255) NOT NULL,
3     `restaurant_id` int(10) NOT NULL,
4     CONSTRAINT fk_restaurantimage_restaurant FOREIGN KEY (`restaurant_id`)
5         REFERENCES `restaurant`(`restaurant_id`) ON DELETE CASCADE,
6     PRIMARY KEY (`restaurant_id`, `restaurant_image_url`)
7 );
```



- **Bảng customer:**

Câu lệnh tạo bảng cho bảng customer:

```
1 CREATE TABLE `customer` (
2     `customer_id` int(10) NOT NULL,
3     `customer_first_name` varchar(255) NOT NULL,
4     `customer_last_name` varchar(255) NOT NULL,
5     `phone_number` varchar(10) NOT NULL,
6     `email` varchar(255) NOT NULL,
7     `points` int(10),
8     CONSTRAINT fk_customer_account FOREIGN KEY (`customer_id`) REFERENCES
9         `account`(`account_id`) ON DELETE CASCADE,
10        PRIMARY KEY (`customer_id`)
11 );
```

Câu lệnh tạo ràng buộc cho bảng customer:

```
1 -- ! Rang buoc ve diem tich luy cua khach hang phai lon hon hoac bang 0
2 ALTER TABLE `customer`
3     ADD CONSTRAINT customer_check_point CHECK (points >= 0);
4 -- ! Rang buoc ve so dien thoai cua khach hang phai dung dinh dang
5 ALTER TABLE `customer`
6     ADD CONSTRAINT customer_check_valid_phonenumber CHECK (LENGTH(phone_number) =
7         10);
8 -- ! Rang buoc ve email cua khach hang phai dung dinh dang
9 ALTER TABLE `customer`
10    ADD CONSTRAINT customer_check_valid_email CHECK (email LIKE '%_@%._%');
```

- **Bảng food:**

Câu lệnh tạo bảng cho bảng food:

```
1 CREATE TABLE `food` (
2     `food_id` int(10) NOT NULL AUTO_INCREMENT,
3     `food_name` varchar(255) NOT NULL,
4     `food_price` int(10) NOT NULL,
5     `food_description` varchar(255) NOT NULL,
6     PRIMARY KEY (`food_id`)
7 );
```

Câu lệnh tạo ràng buộc cho bảng food:

```
1 -- ! Rang buoc ve gia cua mon an phai lon hon 0
2 ALTER TABLE `food`
3     ADD CONSTRAINT food_check_foodprice CHECK (food_price > 0);
4 );
```

- **Bảng food_image:**

Câu lệnh tạo bảng cho bảng food_image:



```
1 CREATE TABLE `food_image` (
2   `food_image_url` varchar(255) NOT NULL,
3   `food_id` int (10) NOT NULL,
4   CONSTRAINT fk_foodimage_food FOREIGN KEY (`food_id`) REFERENCES
5     `food`(`food_id`) ON DELETE CASCADE,
6   PRIMARY KEY (`food_id`, `food_image_url`)
7 );
```

- **Bảng manage_food:**

Câu lệnh tạo bảng cho bảng manage_food:

```
1 CREATE TABLE `manage_food` (
2   `restaurant_id` int(10) NOT NULL,
3   `food_id` int(10) NOT NULL,
4   `price_sell` int(10),
5   CONSTRAINT fk_managefood_restaurant FOREIGN KEY (`restaurant_id`) REFERENCES
6     `restaurant`(`restaurant_id`) ON DELETE CASCADE,
7   CONSTRAINT fk_managefood_food FOREIGN KEY (`food_id`) REFERENCES
8     `food`(`food_id`) ON DELETE CASCADE,
9   PRIMARY KEY (`restaurant_id`,`food_id`)
10 );
```

Câu lệnh tạo ràng buộc cho bảng manage_food:

```
1 -- ! Rang buoc ve gia ban cua mon an phai lon hon 0
2 ALTER TABLE `manage_food`
3   ADD CONSTRAINT manage_food_check_pricesell CHECK (price_sell > 0);
```

- **Bảng user_review:**

Câu lệnh tạo bảng cho bảng user_review:

```
1 CREATE TABLE `user_review` (
2   `user_review_id` int(10) NOT NULL AUTO_INCREMENT,
3   `customer_id` int(10) NOT NULL,
4   `food_id` int(10) NOT NULL,
5   `rating` char(1) NOT NULL,
6   `review_description` varchar(255),
7   CONSTRAINT fk_userreview_customer FOREIGN KEY (`customer_id`) REFERENCES
8     `customer`(`customer_id`) ON DELETE CASCADE,
9   CONSTRAINT fk_userreview_food FOREIGN KEY (`food_id`) REFERENCES
10    `food`(`food_id`) ON DELETE CASCADE,
11   PRIMARY KEY (`user_review_id`)
12 );
```

Câu lệnh tạo ràng buộc cho bảng user_review:

```
1 -- ! Rang buoc rating cua danh gia mon an phai la 1, 2, 3, 4 hoac 5 sao
2 ALTER TABLE `user_review`
```



```
3     ADD CONSTRAINT user_review_check_rating CHECK (rating = 1 OR rating = 2 OR
        rating = 3 OR rating = 4 OR rating = 5);
```

- **Bảng discount:**

Câu lệnh tạo bảng cho bảng discount:

```
1 CREATE TABLE `discount` (
2     `discount_id` int(10) NOT NULL AUTO_INCREMENT,
3     `discount_name` varchar(255),
4     `restaurant_id` int(10) NOT NULL,
5     `status_use` int(1) NOT NULL, -- ! 0: not used, 1: used
6     CONSTRAINT fk_discount_restaurant FOREIGN KEY (`restaurant_id`) REFERENCES
        `restaurant`(`restaurant_id`) ON DELETE CASCADE,
7     PRIMARY KEY (`discount_id`)
8 );
```

Câu lệnh tạo ràng buộc cho bảng discount:

```
1 -- ! Rang buoc ve trang thai cua ma giam gia phai la 0 hoac 1
2 ALTER TABLE `discount`
3     ADD CONSTRAINT discount_check_statususe CHECK (status_use = 0 OR status_use =
        1);
```

- **Bảng discount_on_percent:**

Câu lệnh tạo bảng cho bảng discount_on_percent:

```
1 CREATE TABLE `discount_on_percent` (
2     `discount_id` int(10) NOT NULL,
3     `percent` int(2) NOT NULL,
4     CONSTRAINT fk_discountpercent_discount FOREIGN KEY (`discount_id`) REFERENCES
        `discount`(`discount_id`) ON DELETE CASCADE,
5     PRIMARY KEY (`discount_id`)
6 );
```

Câu lệnh tạo ràng buộc cho bảng discount_on_percent:

```
1 -- ! Rang buoc muc giam gia toi da cua ma giam gia theo phan tram la 40%
2 ALTER TABLE `discount_on_percent`
3     ADD CONSTRAINT discount_check_percent CHECK (percent >= 0 AND percent <= 40);
```

- **Bảng discount_on_number:**

Câu lệnh tạo bảng cho bảng discount_on_number:

```
1 CREATE TABLE `discount_on_number` (
2     `discount_id` int(10) NOT NULL,
3     `discount_number` int(10) NOT NULL,
4     CONSTRAINT fk_discountnumber_discount FOREIGN KEY (`discount_id`) REFERENCES
        `discount`(`discount_id`) ON DELETE CASCADE,
```



```
5     PRIMARY KEY ('discount_id')
6 );
```

Câu lệnh tạo ràng buộc cho bảng discount_on_number:

```
1 -- ! Rang buoc gia tri giam gia toi thieu cua ma giam gia theo so tien lon hon 0
2 ALTER TABLE 'discount_on_number'
3     ADD CONSTRAINT discount_check_discountnumber CHECK (discount_number > 0);
```

- **Bảng customer_discounts:**

Câu lệnh tạo bảng cho bảng customer_discounts:

```
1 CREATE TABLE 'customer_discounts' (
2     'customer_id' int(10) NOT NULL,
3     'discount_id' int(10) NOT NULL,
4     'type_discount' varchar(255) NOT NULL, -- ! percent or number
5     'expired_date' date,
6     CONSTRAINT fk_customerdiscount_customer FOREIGN KEY ('customer_id') REFERENCES
7         'customer'('customer_id') ON DELETE CASCADE,
8     CONSTRAINT fk_customerdiscount_discount FOREIGN KEY ('discount_id') REFERENCES
9         'discount'('discount_id') ON DELETE CASCADE,
10    PRIMARY KEY ('customer_id','discount_id')
11 );
```

Câu lệnh tạo ràng buộc cho bảng customer_discounts:

```
1 -- ! Rang buoc ve loai ma giam gia phai la percent hoac number
2 ALTER TABLE 'customer_discounts'
3     ADD CONSTRAINT customer_discount_check_typediscount CHECK (type_discount =
4         'percent' OR type_discount = 'number');
5 -- ! Rang buoc ve ngay het han cua ma giam gia phai lon hon ngay 01/12/2024
6 ALTER TABLE 'customer_discounts'
7     ADD CONSTRAINT customer_discount_check_expireddate CHECK (expired_date >
8         '2024-12-01');
```

- **Bảng discount_creator:**

Câu lệnh tạo bảng cho bảng discount_creator:

```
1 CREATE TABLE 'discount_creator' (
2     'restaurant_id' int(10) NOT NULL,
3     'discount_id' int(10) NOT NULL,
4     CONSTRAINT fk_discountcreator_restaurant FOREIGN KEY ('restaurant_id')
5         REFERENCES 'restaurant'('restaurant_id') ON DELETE CASCADE,
6     CONSTRAINT fk_discountcreator_discount FOREIGN KEY ('discount_id') REFERENCES
7         'discount'('discount_id') ON DELETE CASCADE,
8     PRIMARY KEY ('restaurant_id','discount_id')
9 );
```



- **Bảng payment_order:**

Câu lệnh tạo bảng cho bảng payment_order:

```
1 CREATE TABLE `payment_order` (
2     `bill_id` int(10) NOT NULL AUTO_INCREMENT,
3     `payment_status` int(1) NOT NULL, -- ! 0: unpaid, 1: paid
4     PRIMARY KEY (`bill_id`)
5 );
```

Câu lệnh tạo ràng buộc cho bảng payment_order:

```
1 -- ! Rang buoc ve trang thai cua hoa don phai la 0 hoac 1
2 ALTER TABLE `payment_order`
3     ADD CONSTRAINT payment_order_check_paymentstatus CHECK (payment_status = 0 OR
4         payment_status = 1);
```

- **Bảng order:**

Câu lệnh tạo bảng cho bảng order:

```
1 CREATE TABLE `order` (
2     `order_id` int(10) NOT NULL AUTO_INCREMENT,
3     `customer_id` int(10) NOT NULL,
4     `order_status` int(1) NOT NULL, -- ! 0: waiting, 1: processing, 2: delivering,
5         3: delivered
6     `bill_id` int(10) NOT NULL,
7     `discount_id` int(10),
8     `address_delivery` varchar(255) NOT NULL DEFAULT 'N/A',
9     `final_price` int(10) NOT NULL DEFAULT 0,
10    `order_date` date NOT NULL,
11    CONSTRAINT fk_order_customer FOREIGN KEY (`customer_id`) REFERENCES
12        `customer`(`customer_id`) ON DELETE CASCADE,
13    CONSTRAINT fk_order_bill FOREIGN KEY (`bill_id`) REFERENCES
14        `payment_order`(`bill_id`) ON DELETE CASCADE,
15    CONSTRAINT fk_createbill_discount FOREIGN KEY (`discount_id`) REFERENCES
16        `discount`(`discount_id`) ON DELETE CASCADE,
17    PRIMARY KEY (`order_id`)
18 );
```

Câu lệnh tạo ràng buộc cho bảng order:

```
1 -- ! Rang buoc ve trang thai cua don hang phai la 0, 1, 2 hoac 3
2 ALTER TABLE `order`
3     ADD CONSTRAINT order_check_orderstatus CHECK (order_status = 0 OR order_status
4         = 1 OR order_status = 2 OR order_status = 3);
5 -- ! Rang buoc ve gia tri cuoi cung cua don hang phai lon hon 0
6 ALTER TABLE `order`
7     ADD CONSTRAINT order_check_finalprice CHECK (final_price > 0);
```

- **Bảng shipping_employee:**



Câu lệnh tạo bảng cho bảng shipping_employee:

```
1 CREATE TABLE `shipping_employee` (
2     `employee_id` int(10) NOT NULL AUTO_INCREMENT,
3     `employee_name` varchar(255) NOT NULL,
4     `employee_phone` int(10) NOT NULL,
5     `order_id` int(10) NOT NULL,
6     CONSTRAINT fk_shippingemployee_order FOREIGN KEY (`order_id`) REFERENCES
7         `order`(`order_id`) ON DELETE CASCADE,
8     PRIMARY KEY (`employee_id`)
9 );
```

Câu lệnh tạo ràng buộc cho bảng shipping_employee:

```
1 -- ! Rang buoc ve so dien thoai cua nhan vien giao hang phai dung dinh dang
2 ALTER TABLE `shipping_employee`
3     ADD CONSTRAINT shipper_check_valid_phonenumber CHECK (LENGTH(employee_phone) =
4         10);
```

- **Bảng receiver:**

Câu lệnh tạo bảng cho bảng receiver:

```
1 CREATE TABLE `receiver` (
2     `receiver_name` varchar(255) NOT NULL,
3     `receiver_phone` int(10) NOT NULL,
4     `receiver_address` varchar(255) NOT NULL,
5     `order_id` int(10) NOT NULL,
6     CONSTRAINT fk_receiver_order FOREIGN KEY (`order_id`) REFERENCES
7         `order`(`order_id`) ON DELETE CASCADE,
8     PRIMARY KEY (`receiver_name`, `order_id`)
9 );
```

Câu lệnh tạo ràng buộc cho bảng receiver:

```
1 -- ! Rang buoc ve so dien thoai cua nguoi nhan phai dung dinh dang
2 ALTER TABLE `receiver`
3     ADD CONSTRAINT receiver_check_valid_phonenumber CHECK (LENGTH(receiver_phone) =
4         10);
```

- **Bảng creat_order:**

Câu lệnh tạo bảng cho bảng creat_order:

```
1 CREATE TABLE `creat_order` (
2     `order_id` int(10) NOT NULL,
3     `food_id` int(10) NOT NULL,
4     `quantity` int(10) NOT NULL,
5     `temp_price` int(10) NOT NULL,
6     `creat_date` date NOT NULL,
7     CONSTRAINT fk_createbill_order FOREIGN KEY (`order_id`) REFERENCES
```



```
8     'order'('order_id') ON DELETE CASCADE,
9     CONSTRAINT fk_createbill_food FOREIGN KEY ('food_id') REFERENCES
10    'food'('food_id') ON DELETE CASCADE,
11    PRIMARY KEY ('order_id', 'food_id')
12 );
```

Câu lệnh tạo ràng buộc cho bảng creat_order:

```
1 -- ! Rang buoc ve gia tri thanh toan phai lon hon 0
2 ALTER TABLE 'creat_order'
3     ADD CONSTRAINT creat_order_check_totalprice CHECK (temp_price > 0);
4 -- ! Rang buoc ve ngay tao hoa don phai lon hon ngay 01/12/2024
5 ALTER TABLE 'creat_order'
6     ADD CONSTRAINT creat_order_check_createdate CHECK (creat_date > '2024-12-01');
```

1.2 Thêm dữ liệu

Kết quả dữ liệu của các bảng trong CSDL:

- **Bảng account:**

	account_id	user_name	password
<input type="checkbox"/>	1	nguyenanh	MatKhau1
<input type="checkbox"/>	2	tranminh	MatKhau2
<input type="checkbox"/>	3	lethuy	MatKhau3
<input type="checkbox"/>	4	phamquang	MatKhau4
<input type="checkbox"/>	5	hoanglan	MatKhau5
<input type="checkbox"/>	6	minhhoang	MatKhau6
<input type="checkbox"/>	7	thanhvan	MatKhau7
<input type="checkbox"/>	8	quanghuy	MatKhau8
<input type="checkbox"/>	9	huyentrang	MatKhau9
<input type="checkbox"/>	10	tuananh	MatKhau10
<input type="checkbox"/>	11	vananh	MatKhau11
<input type="checkbox"/>	12	nhutdo	MatKhau12

Hình 1: Dữ liệu của bảng account sau khi thêm dữ liệu

- **Bảng restaurant:**



	restaurant_id	restaurant_name	restaurant_address	restaurant_description
<input type="checkbox"/>	1	Nhà Hàng Hải Sản Tươi Sống	123 Đường Biển, Quận 1, TP.HCM	Chuyên phục vụ hải sản tươi sống và món ăn đặc sản...
<input type="checkbox"/>	2	Pizza Italia	456 Đường Pizzaiolo, Quận 3, TP.HCM	Pizza kiểu Ý với nguyên liệu nhập khẩu, hương vị đ...

Hình 2: Dữ liệu của bảng restaurant sau khi thêm dữ liệu

- **Bảng restaurant_image:**

	restaurant_image_url	restaurant_id
<input type="checkbox"/>	https://www.shutterstock.com/image-illustration/fr...	1
<input type="checkbox"/>	https://www.shutterstock.com/image-vector/italian-...	2

Hình 3: Dữ liệu của bảng restaurant_image sau khi thêm dữ liệu

- **Bảng customer:**

	customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	6	Bob	Brown	4567890123	bob.brown@example.com	0
<input type="checkbox"/>	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5

Hình 4: Dữ liệu của bảng customer sau khi thêm dữ liệu

- **Bảng food:**



	<input type="button" value="← T →"/>	<input type="button" value="▼"/>	food_id	food_name	food_price	food_description
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>			
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	Pizza Hải Sản	150000 Pizza với tôm, mực và cá hồi tươi ngon.
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	Mì Ý Sốt Bolognese	120000 Mì Ý truyền thống với sốt thịt bò và gia vị đặc tr...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	Salad Trái Cây	80000 Salad tươi ngon với nhiều loại trái cây mùa hè.
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	Sushi Cá Hồi	200000 Sushi tươi với cá hồi và cơm dẻo, cuộn với rong bi...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	Bánh Mì Thịt Nướng	50000 Bánh mì giòn với thịt nướng và rau sống tươi mát.
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	Pizza Margherita	120000 Pizza truyền thống với sốt cà chua, phô mai mozzar...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7	Pizza Pepperoni	150000 Pizza với lớp xúc xích pepperoni thơm ngon, phô ma...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	Pizza Hải Sản đặc biệt	180000 Pizza đặc biệt với tôm, mực và cá hồi tươi ngon, p...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9	Pizza Thập Cẩm	200000 Pizza với đầy đủ các loại nhân như thịt, xúc xích,...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	Pizza Trứng Muối	170000 Pizza sáng tạo với trứng muối, phô mai và rau củ t...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	Pizza Chay	110000 Pizza dành cho người ăn chay với rau củ tươi ngon ...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	Pizza BBQ Gà	160000 Pizza với gà nướng và sốt BBQ thơm lừng, kèm theo ...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	Pizza Quattro Stagioni	190000 Pizza bốn mùa với các loại nhân khác nhau: nấm, gi...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	Pizza Phô Mai Đặc Biệt	140000 Pizza với nhiều loại phô mai khác nhau, dành cho t...
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15	Pizza Trái Cây	130000 Pizza ngọt với trái cây tươi như dứa, kiwi và dâu ...

Hình 5: Dữ liệu của bảng food sau khi thêm dữ liệu

• Bảng food_image:

	<input type="button" value="← T →"/>	<input type="button" value="▼"/>	food_image_url	food_id
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15

Hình 6: Dữ liệu của bảng food_image sau khi thêm dữ liệu

• Bảng manage_food:



		restaurant_id	food_id	price_sell
<input type="checkbox"/>	Edit Copy Delete	1	1	160000
<input type="checkbox"/>	Edit Copy Delete	1	2	140000
<input type="checkbox"/>	Edit Copy Delete	1	4	240000
<input type="checkbox"/>	Edit Copy Delete	1	5	60000
<input type="checkbox"/>	Edit Copy Delete	1	6	140000
<input type="checkbox"/>	Edit Copy Delete	1	7	180000
<input type="checkbox"/>	Edit Copy Delete	1	8	190000
<input type="checkbox"/>	Edit Copy Delete	1	9	130000
<input type="checkbox"/>	Edit Copy Delete	1	10	180000
<input type="checkbox"/>	Edit Copy Delete	1	11	120000
<input type="checkbox"/>	Edit Copy Delete	1	13	210000
<input type="checkbox"/>	Edit Copy Delete	2	1	170000
<input type="checkbox"/>	Edit Copy Delete	2	2	110000
<input type="checkbox"/>	Edit Copy Delete	2	3	90000
<input type="checkbox"/>	Edit Copy Delete	2	4	230000
<input type="checkbox"/>	Edit Copy Delete	2	5	70000
<input type="checkbox"/>	Edit Copy Delete	2	6	120000
<input type="checkbox"/>	Edit Copy Delete	2	8	200000
<input type="checkbox"/>	Edit Copy Delete	2	9	210000
<input type="checkbox"/>	Edit Copy Delete	2	12	170000

Hình 7: Dữ liệu của bảng manage_food sau khi thêm dữ liệu

- **Bảng user_review:**

		user_review_id	customer_id	food_id	rating	review_description
<input type="checkbox"/>	Edit Copy Delete	1	3	1	5	Pizza hải sản rất ngon, đầy đủ các loại hải sản tư...
<input type="checkbox"/>	Edit Copy Delete	2	6	2	4	Mì Ý Sốt Bolognese rất hấp dẫn, ăn cực kỳ thích.
<input type="checkbox"/>	Edit Copy Delete	3	3	3	3	Salad trái cây tươi mát, nhưng một chút ngọt nhat.
<input type="checkbox"/>	Edit Copy Delete	4	6	4	5	Sushi cá hồi tuyệt vời, tươi ngon và cách trình bày...
<input type="checkbox"/>	Edit Copy Delete	5	5	5	4	Bánh mì thịt nướng rất giòn và đậm vị.
<input type="checkbox"/>	Edit Copy Delete	6	6	6	5	Pizza Margherita ngon như ở Italia, cực kỳ hài lòng...
<input type="checkbox"/>	Edit Copy Delete	7	5	7	4	Pizza Pepperoni rất thơm ngon, sốt cà chua và phô ...
<input type="checkbox"/>	Edit Copy Delete	8	3	8	5	Pizza hải sản đặc biệt quá tuyệt, nhiều hải sản và...
<input type="checkbox"/>	Edit Copy Delete	9	8	9	4	Pizza Thập Cẩm đủ các loại nhân, rất đa dạng và ng...
<input type="checkbox"/>	Edit Copy Delete	10	5	10	3	Pizza trứng muối khá ngon, nhưng một chút quá mặn.
<input type="checkbox"/>	Edit Copy Delete	11	7	11	4	Pizza chay rất tươi ngon, sốt cà chua cũng rất hấp...
<input type="checkbox"/>	Edit Copy Delete	12	6	12	5	Pizza BBQ gà rất thơm ngon, vị cay nồng của ớt và ...
<input type="checkbox"/>	Edit Copy Delete	13	9	13	4	Pizza Quattro Stagioni rất đa dạng nhân, mỗi loại ...
<input type="checkbox"/>	Edit Copy Delete	14	4	14	5	Pizza phô mai đặc biệt quá tuyệt, các loại phô mai...
<input type="checkbox"/>	Edit Copy Delete	15	11	15	4	Pizza trái cây khá ngon, vị ngọt của trái cây rất ...

Hình 8: Dữ liệu của bảng user_review sau khi thêm dữ liệu

- **Bảng discount:**



		discount_id	discount_name	restaurant_id	status_use
<input type="checkbox"/>	Edit Copy Delete	1	XMAS2024	1	0
<input type="checkbox"/>	Edit Copy Delete	2	SUMMER2024	2	1
<input type="checkbox"/>	Edit Copy Delete	3	STUDENTDISCOUNT	2	0
<input type="checkbox"/>	Edit Copy Delete	4	NEWCUSTOMER	2	1

Hình 9: Dữ liệu của bảng discount sau khi thêm dữ liệu

- **Bảng discount_on_percent:**

		discount_id	percent
<input type="checkbox"/>	Edit Copy Delete	1	20
<input type="checkbox"/>	Edit Copy Delete	2	15
<input type="checkbox"/>	Edit Copy Delete	3	10
<input type="checkbox"/>	Edit Copy Delete	4	25

Hình 10: Dữ liệu của bảng discount_on_percent sau khi thêm dữ liệu

- **Bảng discount_on_number:**

		discount_id	discount_number
<input type="checkbox"/>	Edit Copy Delete	1	50000
<input type="checkbox"/>	Edit Copy Delete	2	30000
<input type="checkbox"/>	Edit Copy Delete	3	20000
<input type="checkbox"/>	Edit Copy Delete	4	40000

Hình 11: Dữ liệu của bảng discount_on_number sau khi thêm dữ liệu

- **Bảng customer_discounts:**

		customer_id	discount_id	type_discount	expired_date
<input type="checkbox"/>	Edit Copy Delete	3	1	percent	2024-12-31
<input type="checkbox"/>	Edit Copy Delete	4	2	number	2025-01-15
<input type="checkbox"/>	Edit Copy Delete	5	3	percent	2025-01-15
<input type="checkbox"/>	Edit Copy Delete	7	4	number	2025-01-15
<input type="checkbox"/>	Edit Copy Delete	8	1	percent	2024-12-31

Hình 12: Dữ liệu của bảng customer_discounts sau khi thêm dữ liệu

- **Bảng discount_creator:**



		restaurant_id	discount_id
<input type="checkbox"/>	Edit Copy Delete	1	1
<input type="checkbox"/>	Edit Copy Delete	1	4
<input type="checkbox"/>	Edit Copy Delete	2	2
<input type="checkbox"/>	Edit Copy Delete	2	3

Hình 13: Dữ liệu của bảng discount_creator sau khi thêm dữ liệu

- **Bảng payment_order:**

		bill_id	payment_status
<input type="checkbox"/>	Edit Copy Delete	1	0
<input type="checkbox"/>	Edit Copy Delete	2	1
<input type="checkbox"/>	Edit Copy Delete	3	0
<input type="checkbox"/>	Edit Copy Delete	4	1
<input type="checkbox"/>	Edit Copy Delete	5	1
<input type="checkbox"/>	Edit Copy Delete	6	0
<input type="checkbox"/>	Edit Copy Delete	7	1
<input type="checkbox"/>	Edit Copy Delete	8	0
<input type="checkbox"/>	Edit Copy Delete	9	1
<input type="checkbox"/>	Edit Copy Delete	10	0

Hình 14: Dữ liệu của bảng payment_order sau khi thêm dữ liệu

- **Bảng order:**

		order_id	customer_id	order_status	bill_id	discount_id	address_delivery	final_price	order_date
<input type="checkbox"/>	Edit Copy Delete	1	3	0	1	NULL	123 Main St, Anytown USA	100000	2023-05-01
<input type="checkbox"/>	Edit Copy Delete	2	4	1	2	NULL	456 Oak Rd, Othertown USA	150000	2023-06-15
<input type="checkbox"/>	Edit Copy Delete	3	5	0	3	NULL	789 Elm St, Someplace USA	80000	2023-07-20
<input type="checkbox"/>	Edit Copy Delete	4	6	1	4	NULL	321 Maple Ave, Nowhere USA	200000	2023-08-05
<input type="checkbox"/>	Edit Copy Delete	5	7	2	5	NULL	654 Pine Rd, Everytown USA	120000	2023-09-10
<input type="checkbox"/>	Edit Copy Delete	6	8	0	6	NULL	987 Birch Blvd, Anotherton USA	90000	2023-10-01
<input type="checkbox"/>	Edit Copy Delete	7	9	1	7	NULL	159 Oak St, Somewhereville USA	180000	2023-11-15
<input type="checkbox"/>	Edit Copy Delete	8	10	0	8	NULL	753 Elm Ave, Nowhere USA	70000	2023-12-01
<input type="checkbox"/>	Edit Copy Delete	9	11	2	9	NULL	258 Maple Dr, Everytown USA	150000	2024-01-05
<input type="checkbox"/>	Edit Copy Delete	10	12	0	10	NULL	456 Pine St, Anotherton USA	110000	2024-02-10

Hình 15: Dữ liệu của bảng order sau khi thêm dữ liệu

- **Bảng shipping_employee:**



	employee_id	employee_name	employee_phone	order_id
<input type="checkbox"/>	1	Nguyễn Văn Hiền	2147483647	1
<input type="checkbox"/>	2	Trần Thị Hoa	2147483647	2
<input type="checkbox"/>	3	Lê Hoàng Công	2147483647	3
<input type="checkbox"/>	4	Phạm Minh Danh	2147483647	4
<input type="checkbox"/>	5	Võ Thị Ngân	1597539514	5

Hình 16: Dữ liệu của bảng shipping_employee sau khi thêm dữ liệu

- **Bảng receiver:**

	receiver_name	receiver_phone	receiver_address	order_id
<input type="checkbox"/>	Lê Hoàng Công	2147483647	789 Nguyễn Trãi, Quận 1, TP.HCM	3
<input type="checkbox"/>	Ngô Thanh Thảo	2147483647	987 Phan Đình Phùng, Quận Phú Nhuân, TP.HCM	6
<input type="checkbox"/>	Nguyễn Văn Minh	2147483647	123 Nguyễn Văn Cừ, Quận 5, TP.HCM	1
<input type="checkbox"/>	Phạm Minh Tiến	2147483647	321 Trần Hưng Đạo, Quận 5, TP.HCM	4
<input type="checkbox"/>	Trần Thị Lan	1234567890	456 Lê Văn Sỹ, Quận 3, TP.HCM	2
<input type="checkbox"/>	Võ Thị Chi	1597539514	654 Lê Đại Hành, Quận 11, TP.HCM	5

Hình 17: Dữ liệu của bảng receiver sau khi thêm dữ liệu

- **Bảng creat_order:**

	order_id	food_id	quantity	temp_price	creat_date
<input type="checkbox"/>	1	1	2	300000	2025-01-01
<input type="checkbox"/>	1	2	1	120000	2025-01-01
<input type="checkbox"/>	1	3	1	80000	2025-01-01
<input type="checkbox"/>	2	4	1	200000	2025-01-15
<input type="checkbox"/>	2	5	1	50000	2025-01-15
<input type="checkbox"/>	3	6	1	120000	2025-02-20
<input type="checkbox"/>	3	7	1	150000	2025-02-20
<input type="checkbox"/>	4	8	1	180000	2025-03-05
<input type="checkbox"/>	4	9	1	200000	2025-03-05
<input type="checkbox"/>	5	10	1	170000	2025-04-10
<input type="checkbox"/>	5	11	1	110000	2025-04-10
<input type="checkbox"/>	6	12	1	160000	2025-05-01
<input type="checkbox"/>	6	13	1	190000	2025-05-01
<input type="checkbox"/>	7	14	1	140000	2025-06-15
<input type="checkbox"/>	7	15	1	130000	2025-06-15

Hình 18: Dữ liệu của bảng creat_order sau khi thêm dữ liệu



2 Hiện thực ứng dụng

2.1 Thủ tục INSERT / UPDATE / DELETE dữ liệu vào 1 bảng dữ liệu

2.1.1 Thủ tục INSERT

a. Mô tả thủ tục:

- **Chức năng:** Thủ tục InsertCustomer được thiết kế để thêm một khách hàng mới vào bảng customer. Thủ tục thực hiện các kiểm tra về tính hợp lệ của dữ liệu đầu vào trước khi chèn thông tin vào cơ sở dữ liệu.
- **Các trường hợp sử dụng:** Thủ tục InsertCustomer được thiết kế để thêm một khách hàng mới vào bảng customer. Thủ tục thực hiện các kiểm tra về tính hợp lệ của dữ liệu đầu vào trước khi chèn thông tin vào cơ sở dữ liệu.
- **Các thao tác chính:**
 - Kiểm tra xem account_id có tồn tại trong bảng account.
 - Đảm bảo rằng account_id chưa tồn tại trong bảng customer (tránh trùng lặp).
 - Kiểm tra định dạng số điện thoại (phone_number) phải hợp lệ (10 chữ số và bắt đầu bằng số 0).
 - Kiểm tra định dạng email (email) phải hợp lệ.
 - Thêm bản ghi vào bảng customer với điểm tích lũy (points) mặc định là 0 nếu không được cung cấp.

- **Các điều kiện kiểm tra:**

- account_id phải tồn tại trong bảng account.
- account_id không được trùng lặp trong bảng customer.
- phone_number phải có 10 chữ số và tuân theo định dạng 0[0-9]9.
- email phải tuân theo định dạng thông thường (có @ và .).
- Giá trị điểm tích lũy (points) không hợp lệ sẽ được mặc định là 0.

b. Input

- **Các thông số nhập vào:**

- p_account_id: ID của tài khoản (INT).
- p_first_name: Họ của khách hàng (VARCHAR(255)).
- p_last_name: Tên của khách hàng (VARCHAR(255)).
- p_phone_number: Số điện thoại của khách hàng (VARCHAR(10)).
- p_email: Email của khách hàng (VARCHAR(255)).
- p_points: Điểm tích lũy (INT, tùy chọn).

c. Output:



- **Kết quả trả về:** Không trả về giá trị.

- Nếu dữ liệu hợp lệ, thủ tục thêm khách hàng mới vào bảng customer.
- Nếu xảy ra lỗi:
 - * Lỗi 'Account ID does not exist in account table.' nếu account_id không tồn tại.
 - * Lỗi 'Customer with this Account ID already exists.' nếu account_id đã tồn tại trong customer.
 - * Lỗi 'Invalid phone number format.' nếu số điện thoại không hợp lệ.
 - * Lỗi 'Invalid email format.' nếu email không hợp lệ.

d. Câu lệnh tạo thủ tục:

- Câu lệnh tạo thủ tục:

```
1  DELIMITER //
2
3  CREATE PROCEDURE InsertCustomer(
4      IN p_account_id INT,
5      IN p_first_name VARCHAR(255),
6      IN p_last_name VARCHAR(255),
7      IN p_phone_number VARCHAR(10),
8      IN p_email VARCHAR(255),
9      IN p_points INT
10 )
11 BEGIN
12     IF NOT EXISTS (SELECT 1 FROM account WHERE account_id = p_account_id) THEN
13         SIGNAL SQLSTATE '45000'
14         SET MESSAGE_TEXT = 'Account ID does not exist in account table.';
15     END IF;
16
17     IF EXISTS (SELECT 1 FROM customer WHERE customer_id = p_account_id) THEN
18         SIGNAL SQLSTATE '45000'
19         SET MESSAGE_TEXT = 'Customer with this Account ID already exists.';
20     END IF;
21
22     IF CHAR_LENGTH(p_phone_number) != 10 OR p_phone_number NOT REGEXP
23         '^0[0-9]{9}$' THEN
24         SIGNAL SQLSTATE '45000'
25         SET MESSAGE_TEXT = 'Invalid phone number format.';
26     END IF;
27
28     IF p_email NOT LIKE '%_@%._%' THEN
29         SIGNAL SQLSTATE '45000'
30         SET MESSAGE_TEXT = 'Invalid email format.';
31     END IF;
32
33     INSERT INTO customer (customer_id, customer_first_name, customer_last_name,
34                         phone_number, email, points)
35     VALUES (p_account_id, p_first_name, p_last_name, p_phone_number, p_email,
36             COALESCE(p_points, 0));
37
38 END //
```



- Câu lệnh thực thi thủ tục - Trường hợp thêm thành công:

```
1 CALL InsertCustomer( 13, 'Anh', 'Le', '0337808545', 'anh.le@gmail.com', 0);
```

The screenshot shows the results of a MySQL query execution. The message at the top says "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0031 seconds.)". Below it is the SQL command: "CALL InsertCustomer(13, 'Anh', 'Le', '0337808545', 'anh.le@gmail.com', 0);". At the bottom, there are links for "[Edit inline]" and "[Create PHP code]".

Hình 19: Thực thi chèn dữ liệu thành công

The screenshot shows the "customer" table in MySQL Workbench. The table has columns: customer_id, customer_first_name, customer_last_name, phone_number, email, and points. There are 13 rows of data. The last row inserted, with customer_id 13, has values: Anh, Le, 0337808545, anh.le@gmail.com, and 0.

	customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	6	Bob	Brown	4567890123	bob.brown@example.com	0
<input type="checkbox"/>	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5
<input type="checkbox"/>	13	Anh	Le	0337808545	anh.le@gmail.com	0

Hình 20: Dữ liệu trong bảng customer sau khi thêm một bản ghi mới

- Câu lệnh thực thi thủ tục - Trường hợp lỗi:

```
1 CALL InsertCustomer( 12, 'Anh', 'Le', '0337808545', 'anh.le@gmail.com', 0);
```

The screenshot shows an error message from MySQL Workbench. The title is "Error". The SQL query is "CALL InsertCustomer(12, 'Anh', 'Le', '0337808545', 'anh.le@gmail.com', 0);". The MySQL error message is "#1644 - Customer with this Account ID already exists.".

Hình 21: Thực thi chèn dữ liệu lỗi: Đã có khách hàng có account_id = 12 trong bảng



2.1.2 Thủ tục UPDATE

a. Mô tả thủ tục:

- **Chức năng:** Thủ tục UpdateCustomer được thiết kế để cập nhật thông tin của một khách hàng trong bảng customer.
- **Các trường hợp sử dụng:** Sử dụng khi cần thay đổi thông tin như họ, tên, số điện thoại, email hoặc điểm tích lũy của khách hàng.
- **Các thao tác chính:**
 - Kiểm tra xem account_id có tồn tại trong bảng account và bảng customer.
 - Kiểm tra định dạng số điện thoại (phone_number) phải hợp lệ (10 chữ số và bắt đầu bằng số 0).
 - Kiểm tra định dạng email (email) phải hợp lệ.
 - Cập nhật thông tin khách hàng trong bảng customer.
- **Các điều kiện kiểm tra:**
 - account_id phải tồn tại trong bảng account.
 - account_id phải tồn tại trong bảng customer.
 - phone_number phải có 10 chữ số và tuân theo định dạng 0[0-9]9.
 - email phải tuân theo định dạng thông thường (có @ và .).
 - Giá trị điểm tích lũy (points) không hợp lệ sẽ được mặc định là 0.

b. Input:

• Các thông số nhập vào:

- p_account_id: ID của tài khoản (INT).
- p_first_name: Họ của khách hàng (VARCHAR(255)).
- p_last_name: Tên của khách hàng (VARCHAR(255)).
- p_phone_number: Số điện thoại của khách hàng (VARCHAR(10)).
- p_email: Email của khách hàng (VARCHAR(255)).
- p_points: Điểm tích lũy (INT, tùy chọn).

c. Output:

• Kết quả trả về: Không trả về giá trị.

- Nếu dữ liệu hợp lệ, thủ tục cập nhật thông tin khách hàng trong bảng customer.
- Nếu xảy ra lỗi:
 - * Lỗi 'Account ID does not exist in account table.' nếu account_id không tồn tại trong bảng account.
 - * Lỗi 'Customer with this Account ID does not exist.' nếu account_id không tồn tại trong bảng customer.



- * Lỗi 'Invalid phone number format.' nếu số điện thoại không hợp lệ.
- * Lỗi 'Invalid email format.' nếu email không hợp lệ.

d. Câu lệnh tạo thủ tục:

- Câu lệnh tạo thủ tục:

```
1  DELIMITER //
2
3  CREATE PROCEDURE UpdateCustomer(
4      IN p_account_id INT,
5      IN p_first_name VARCHAR(255),
6      IN p_last_name VARCHAR(255),
7      IN p_phone_number VARCHAR(10),
8      IN p_email VARCHAR(255),
9      IN p_points INT
10 )
11 BEGIN
12     IF NOT EXISTS (SELECT 1 FROM account WHERE account_id = p_account_id) THEN
13         SIGNAL SQLSTATE '45000'
14         SET MESSAGE_TEXT = 'Account ID does not exist in account table.';
15     END IF;
16
17     IF NOT EXISTS (SELECT 1 FROM customer WHERE customer_id = p_account_id) THEN
18         SIGNAL SQLSTATE '45000'
19         SET MESSAGE_TEXT = 'Customer with this Account ID does not exist.';
20     END IF;
21
22     IF CHAR_LENGTH(p_phone_number) != 10 OR p_phone_number NOT REGEXP
23         '^0[0-9]{9}$' THEN
24         SIGNAL SQLSTATE '45000'
25         SET MESSAGE_TEXT = 'Invalid phone number format.';
26     END IF;
27
28     IF p_email NOT LIKE '%_@%._%' THEN
29         SIGNAL SQLSTATE '45000'
30         SET MESSAGE_TEXT = 'Invalid email format.';
31     END IF;
32
33     UPDATE customer
34     SET customer_first_name = p_first_name,
35         customer_last_name = p_last_name,
36         phone_number = p_phone_number,
37         email = p_email,
38         points = COALESCE(p_points, 0)
39     WHERE customer_id = p_account_id;
40 END //
```

- Câu lệnh thực thi thủ tục - Trường hợp update thành công:

```
1  --- update email
2  CALL UpdateCustomer( 13, 'Anh', 'Le', '0337808545', 'hoanganh.le@gmail.com', 0);
```



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)  
CALL UpdateCustomer( 13, 'Anh', 'Le', '0337808545', 'hoanganh.le@gmail.com', 0);  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

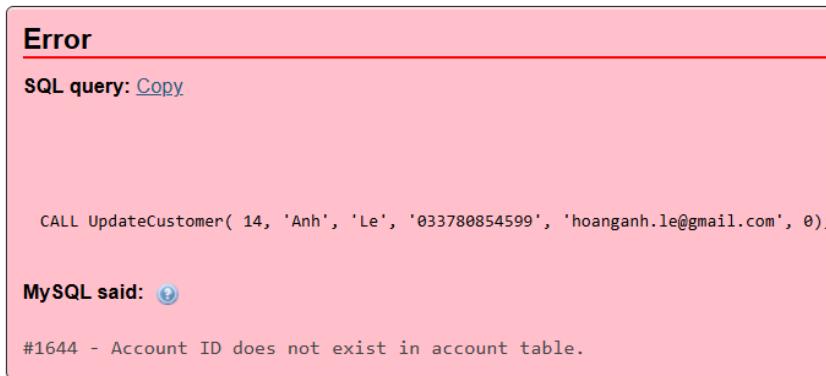
Hình 22: Thực thi update dữ liệu thành công

	customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	6	Bob	Brown	4567890123	bob.brown@example.com	0
<input type="checkbox"/>	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5
<input type="checkbox"/>	13	Anh	Le	0337808545	hoanganh.le@gmail.com	0

Hình 23: Dữ liệu trong bảng customer sau khi update phần email

- Câu lệnh thực thi thủ tục - Trường hợp lỗi:

```
1 CALL UpdateCustomer( 14, 'Anh', 'Le', '0337808545', 'hoanganh.le@gmail.com', 0);
```



Hình 24: Thực thi update dữ liệu lỗi: Không tồn tại khách hàng có account_id = 14 trong bảng

2.1.3 Thủ tục DELETE

a. Mô tả thủ tục:

- Chức năng:** Thủ tục DeleteCustomer được thiết kế để xóa một khách hàng khỏi bảng customer.



- **Các trường hợp sử dụng:** Sử dụng khi cần loại bỏ thông tin khách hàng không còn sử dụng trong hệ thống.

- **Các thao tác chính:**

- Kiểm tra xem account_id có tồn tại trong bảng account và bảng customer.
 - Xóa khách hàng khỏi bảng customer.

- **Các điều kiện kiểm tra:**

- account_id phải tồn tại trong bảng account.
 - account_id phải tồn tại trong bảng customer.

b. Input:

- **Các thông số nhập vào:**

- p_account_id: ID của tài khoản (INT).

c. Output:

- **Kết quả trả về:** Không trả về giá trị.

- Nếu thành công, khách hàng được xóa khỏi bảng customer.
 - Nếu xảy ra lỗi:
 - * Lỗi 'Account ID does not exist in account table.' nếu account_id không tồn tại trong bảng account.
 - * Lỗi 'Customer with this Account ID does not exist.' nếu account_id không tồn tại trong bảng customer.

d. Câu lệnh tạo thủ tục:

- Câu lệnh tạo thủ tục:

```
1  DELIMITER //
2
3  CREATE PROCEDURE DeleteCustomer(
4      IN p_account_id INT
5  )
6  BEGIN
7      IF NOT EXISTS (SELECT 1 FROM account WHERE account_id = p_account_id) THEN
8          SIGNAL SQLSTATE '45000'
9          SET MESSAGE_TEXT = 'Account ID does not exist in account table.';
10     END IF;
11
12     IF NOT EXISTS (SELECT 1 FROM customer WHERE customer_id = p_account_id) THEN
13         SIGNAL SQLSTATE '45000'
14         SET MESSAGE_TEXT = 'Customer with this Account ID does not exist.';
15     END IF;
16
17     DELETE FROM customer
18     WHERE customer_id = p_account_id;
19 END //
```



- Câu lệnh thực thi thủ tục - Trường hợp xóa thành công:

```
1 CALL DeleteCustomer(13);
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0023 seconds.)

```
CALL DeleteCustomer(13);
```

[Edit inline] [Edit] [Create PHP code]

Hình 25: Thực thi delete dữ liệu thành công

	customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/> Edit Copy Delete	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/> Edit Copy Delete	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/> Edit Copy Delete	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/> Edit Copy Delete	6	Bob	Brown	4567890123	bob.brown@example.com	0
<input type="checkbox"/> Edit Copy Delete	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/> Edit Copy Delete	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/> Edit Copy Delete	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/> Edit Copy Delete	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/> Edit Copy Delete	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/> Edit Copy Delete	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5

Hình 26: Dữ liệu trong bảng customer sau khi delete

- Câu lệnh thực thi thủ tục - Trường hợp xóa lỗi:

```
1 CALL DeleteCustomer(14);
```

Error

SQL query: [Copy](#)

```
CALL DeleteCustomer(14);
```

MySQL said: [Error](#)

#1644 - Account ID does not exist in account table.

Hình 27: Thực thi delete dữ liệu lỗi: Không tồn tại khách hàng có account_id = 14 trong bảng



2.2 TRIGGER

2.2.1 Trigger 1

Trigger cập nhật điểm thưởng cho khách hàng đánh giá 5 sao

- **Chức năng:** Trigger này tự động tăng điểm thưởng cho khách hàng khi họ đánh giá 5 sao cho món ăn trong đơn hàng đã giao.
- **Các thao tác chính:**
 - Kiểm tra xem khách hàng và món ăn có tồn tại.
 - Nếu điểm đánh giá là 5, cộng thêm điểm thưởng cho khách hàng.
- **Câu lệnh tạo trigger:**

```
1 DELIMITER $$  
2  
3 CREATE TRIGGER update_customer_points  
4 AFTER INSERT ON user_review  
5 FOR EACH ROW  
6 BEGIN  
7     IF EXISTS (SELECT 1 FROM customer WHERE customer_id = NEW.customer_id) THEN  
8         IF EXISTS (SELECT 1 FROM food WHERE food_id = NEW.food_id) THEN  
9             IF NEW.rating = 5 THEN  
10                 UPDATE customer  
11                     SET points = points + 10  
12                     WHERE customer_id = NEW.customer_id;  
13             END IF;  
14         END IF;  
15     END IF;  
16 END;  
17 $$  
18 DELIMITER ;
```

- **Câu lệnh kiểm tra trigger:** Trường hợp đánh giá 5* và được cộng điểm tích lũy:

```
1 INSERT INTO user_review (customer_id, food_id, rating, review_description) VALUES  
2 (6, 1, 5, 'Amazing Pizza!');
```



	customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	6	Bob	Brown	4567890123	bob.brown@example.com	0
<input type="checkbox"/>	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5

Hình 28: Dữ liệu bảng customer trước khi khách hàng Bob Brown đánh giá món ăn 5*

✓ 1 row inserted.
Inserted row id: 16 (Query took 0.0003 seconds.)

```
INSERT INTO user_review (customer_id, food_id, rating, review_description) VALUES (6, 1, 5, 'Amazing Pizza!');
```

[Edit inline] [Edit] [Create PHP code]

Hình 29: Câu lệnh kiểm tra: Khách hàng đánh giá món ăn 5*

	customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	6	Bob	Brown	4567890123	bob.brown@example.com	10
<input type="checkbox"/>	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5

Hình 30: Dữ liệu bảng customer sau khi khách hàng Bob Brown đánh giá món ăn 5*: points được cộng 10 điểm

- **Câu lệnh test trigger:** Trường hợp đánh giá không phải 5* và không được cộng điểm tích lũy:

```
1   INSERT INTO user_review (customer_id, food_id, rating, review_description)
      VALUES (4, 2, 3, 'So so');
```



		customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	Edit Copy Delete	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	Edit Copy Delete	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	Edit Copy Delete	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	Edit Copy Delete	6	Bob	Brown	4567890123	bob.brown@example.com	10
<input type="checkbox"/>	Edit Copy Delete	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	Edit Copy Delete	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	Edit Copy Delete	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	Edit Copy Delete	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	Edit Copy Delete	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	Edit Copy Delete	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5

Hình 31: Dữ liệu bảng customer trước khi khách hàng Jane Smith đánh giá món ăn 3*

```
✓ 1 row inserted.  
Inserted row id: 17 (Query took 0.0005 seconds.)  
  
INSERT INTO user_review (customer_id, food_id, rating, review_description) VALUES (4, 2, 3, 'So so');  
  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

Hình 32: Câu lệnh kiểm tra: Khách hàng đánh giá món ăn 3*

		customer_id	customer_first_name	customer_last_name	phone_number	email	points
<input type="checkbox"/>	Edit Copy Delete	3	John	Doe	1234567890	john.doe@example.com	100
<input type="checkbox"/>	Edit Copy Delete	4	Jane	Smith	2345678901	jane.smith@example.com	200
<input type="checkbox"/>	Edit Copy Delete	5	Alice	Johnson	3456789012	alice.johnson@example.com	50
<input type="checkbox"/>	Edit Copy Delete	6	Bob	Brown	4567890123	bob.brown@example.com	10
<input type="checkbox"/>	Edit Copy Delete	7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150
<input type="checkbox"/>	Edit Copy Delete	8	Diana	Moore	6789012345	diana.moore@example.com	75
<input type="checkbox"/>	Edit Copy Delete	9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300
<input type="checkbox"/>	Edit Copy Delete	10	Fiona	Anderson	8901234567	fiona.anderson@example.com	20
<input type="checkbox"/>	Edit Copy Delete	11	George	Thomas	9012345678	george.thomas@example.com	10
<input type="checkbox"/>	Edit Copy Delete	12	Hannah	Jackson	1023456789	hannah.jackson@example.com	5

Hình 33: Dữ liệu bảng customer sau khi khách hàng Jane Smith đánh giá món ăn 5*: points không thay đổi

2.2.2 Nhóm các trigger 2

Tính giá cuối cùng (final_price) trong bảng 'order' sau khi insert/update/delete trong bảng 'creat_order'

- Chức năng:** Các trigger này được thiết kế để cập nhật giá cuối cùng (final_price) của đơn hàng mỗi khi có thay đổi trong bảng 'creat_order'.
 - Các thao tác chính:**
 - Tính tổng giá tạm tính của tất cả các món ăn trong đơn hàng.



- Áp dụng các mã giảm giá (nếu có) để tính giá cuối cùng.
- Cập nhật giá cuối cùng vào bảng ‘order’.

- Câu lệnh tạo trigger:

- Trigger After Insert

```
1  DELIMITER $$  
2  
3  CREATE TRIGGER after_insert_creat_order  
4    AFTER INSERT ON creat_order  
5    FOR EACH ROW  
6    BEGIN  
7      DECLARE temp_final INT DEFAULT 0;  
8      DECLARE discount_type VARCHAR(255);  
9      DECLARE discount_value INT;  
10     DECLARE discount_percent INT;  
11  
12     IF NOT EXISTS (SELECT 1 FROM `order` WHERE `order_id` = NEW.order_id) THEN  
13         SIGNAL SQLSTATE '45000'  
14         SET MESSAGE_TEXT = 'Error: order_id does not exist in `order` table';  
15     END IF;  
16  
17     SELECT SUM(temp_price) INTO temp_final  
18     FROM creat_order  
19     WHERE order_id = NEW.order_id;  
20  
21     IF EXISTS (SELECT 1 FROM `order` WHERE `order_id` = NEW.order_id AND  
22         `discount_id` IS NOT NULL) THEN  
23         SELECT d.discount_id, dc.type_discount, dp.percent, dn.discount_number  
24         INTO @discount_id, @type_discount, @percent, @discount_number  
25         FROM `discount` d  
26         LEFT JOIN `discount_on_percent` dp ON d.discount_id = dp.discount_id  
27         LEFT JOIN `discount_on_number` dn ON d.discount_id = dn.discount_id  
28         LEFT JOIN `customer_discounts` dc ON d.discount_id = dc.discount_id  
29         WHERE d.discount_id = (SELECT discount_id FROM `order` WHERE `order_id` =  
30             NEW.order_id);  
31  
32         IF @type_discount = 'percent' THEN  
33             SET temp_final = temp_final - (temp_final * @percent / 100);  
34         ELSEIF @type_discount = 'number' THEN  
35             SET temp_final = temp_final - @discount_number;  
36         END IF;  
37     END IF;  
38  
39     IF temp_final < 0 THEN  
40         SET temp_final = 1;  
41     END IF;  
42  
43     UPDATE `order`  
44     SET final_price = temp_final  
        WHERE order_id = NEW.order_id;  
45 END;
```



```
45  $$  
46  DELIMITER ;  
  
- Trigger After Update  


---



```
1 DELIMITER $$
2
3 CREATE TRIGGER after_update_creat_order
4 AFTER UPDATE ON creat_order
5 FOR EACH ROW
6 BEGIN
7 DECLARE temp_final INT DEFAULT 0;
8 DECLARE discount_type VARCHAR(255);
9 DECLARE discount_value INT;
10 DECLARE discount_percent INT;
11
12 SELECT SUM(temp_price) INTO temp_final
13 FROM creat_order
14 WHERE order_id = NEW.order_id;
15
16 IF EXISTS (SELECT 1 FROM 'order' WHERE 'order_id' = NEW.order_id AND
17 'discount_id' IS NOT NULL) THEN
18 SELECT d.discount_id, dc.type_discount, dp.percent, dn.discount_number
19 INTO @discount_id, @type_discount, @percent, @discount_number
20 FROM 'discount' d
21 LEFT JOIN 'discount_on_percent' dp ON d.discount_id = dp.discount_id
22 LEFT JOIN 'discount_on_number' dn ON d.discount_id = dn.discount_id
23 LEFT JOIN 'customer_discounts' dc ON d.discount_id = dc.discount_id
24 WHERE d.discount_id = (SELECT discount_id FROM 'order' WHERE 'order_id' =
25 NEW.order_id);
26
27 IF @type_discount = 'percent' THEN
28 SET temp_final = temp_final - (temp_final * @percent / 100);
29 ELSEIF @type_discount = 'number' THEN
30 SET temp_final = temp_final - @discount_number;
31 END IF;
32 END IF;
33
34 IF temp_final < 0 THEN
35 SET temp_final = 1;
36 END IF;
37
38 UPDATE 'order'
39 SET final_price = temp_final
40 WHERE order_id = NEW.order_id;
41 END;
42 $$
43 DELIMITER ;

```


```

- Trigger After Delete

```
1  DELIMITER $$
```



```
2
3  CREATE TRIGGER after_delete_creat_order
4  AFTER DELETE ON creat_order
5  FOR EACH ROW
6  BEGIN
7      DECLARE temp_final INT DEFAULT 0;
8      DECLARE discount_type VARCHAR(255);
9      DECLARE discount_value INT;
10     DECLARE discount_percent INT;
11
12     SELECT SUM(temp_price) INTO temp_final
13     FROM creat_order
14     WHERE order_id = OLD.order_id;
15
16     IF EXISTS (SELECT 1 FROM `order` WHERE `order_id` = OLD.order_id AND
17         `discount_id` IS NOT NULL) THEN
18         SELECT d.discount_id, dc.type_discount, dp.percent, dn.discount_number
19         INTO @discount_id, @type_discount, @percent, @discount_number
20         FROM `discount` d
21         LEFT JOIN `discount_on_percent` dp ON d.discount_id = dp.discount_id
22         LEFT JOIN `discount_on_number` dn ON d.discount_id = dn.discount_id
23         LEFT JOIN `customer_discounts` dc ON d.discount_id = dc.discount_id
24         WHERE d.discount_id = (SELECT discount_id FROM `order` WHERE `order_id` =
25             OLD.order_id);
26
27         IF @type_discount = 'percent' THEN
28             SET temp_final = temp_final - (temp_final * @percent / 100);
29         ELSEIF @type_discount = 'number' THEN
30             SET temp_final = temp_final - @discount_number;
31         END IF;
32     END IF;
33
34     IF temp_final < 0 THEN
35         SET temp_final = 1;
36     END IF;
37
38     UPDATE `order`
39     SET final_price = temp_final
40     WHERE order_id = OLD.order_id;
41 END;
42 $$;
43 DELIMITER ;
```

- Câu lệnh kiểm tra trigger: Thêm một bản ghi vào bảng create_order:

```
1  INSERT INTO creat_order (order_id, food_id, quantity, temp_price, creat_date)
2  VALUES (4, 1, 1, 150000, CURDATE());
```



	order_id	customer_id	order_status	bill_id	discount_id	address_delivery	final_price	order_date
<input type="checkbox"/>	1	3	0	1	NULL	123 Main St, Anytown USA	100000	2023-05-01
<input type="checkbox"/>	2	4	1	2	NULL	456 Oak Rd, Othertown USA	150000	2023-06-15
<input type="checkbox"/>	3	5	0	3	NULL	789 Elm St, Someplace USA	80000	2023-07-20
<input type="checkbox"/>	4	6	1	4	NULL	321 Maple Ave, Nowhere USA	200000	2023-08-05
<input type="checkbox"/>	5	7	2	5	NULL	654 Pine Rd, Everytown USA	120000	2023-09-10
<input type="checkbox"/>	6	8	0	6	NULL	987 Birch Blvd, Anotherton USA	90000	2023-10-01
<input type="checkbox"/>	7	9	1	7	NULL	159 Oak St, Somewhereville USA	180000	2023-11-15
<input type="checkbox"/>	8	10	0	8	NULL	753 Elm Ave, Nowhere USA	70000	2023-12-01
<input type="checkbox"/>	9	11	2	9	NULL	258 Maple Dr, Everytown USA	150000	2024-01-05
<input type="checkbox"/>	10	12	0	10	NULL	456 Pine St, Anotherton USA	110000	2024-02-10

Hình 34: Dữ liệu bảng order chưa trường final_price trước khi thêm một bản ghi mới vào bảng create_order

✓ 1 row inserted. (Query took 0.0004 seconds.)

```
INSERT INTO creat_order (order_id, food_id, quantity, temp_price, creat_date) VALUES (4, 1, 1, 150000, CURDATE());
```

[Edit inline] [Edit] [Create PHP code]

	order_id	food_id	quantity	temp_price	creat_date
<input type="checkbox"/>	1	1	2	300000	2025-01-01
<input type="checkbox"/>	1	2	1	120000	2025-01-01
<input type="checkbox"/>	1	3	1	80000	2025-01-01
<input type="checkbox"/>	2	4	1	200000	2025-01-15
<input type="checkbox"/>	2	5	1	50000	2025-01-15
<input type="checkbox"/>	3	6	1	120000	2025-02-20
<input type="checkbox"/>	3	7	1	150000	2025-02-20
<input type="checkbox"/>	4	1	1	150000	2024-12-08
<input type="checkbox"/>	4	8	1	180000	2025-03-05
<input type="checkbox"/>	4	9	1	200000	2025-03-05
<input type="checkbox"/>	5	10	1	170000	2025-04-10
<input type="checkbox"/>	5	11	1	110000	2025-04-10
<input type="checkbox"/>	6	12	1	160000	2025-05-01
<input type="checkbox"/>	6	13	1	190000	2025-05-01
<input type="checkbox"/>	7	14	1	140000	2025-06-15
<input type="checkbox"/>	7	15	1	130000	2025-06-15

Hình 35: Chèn bản ghi mới ngày 08-12-2024 vào bảng creat_order thành công, trigger kích hoạt cho đơn hàng có order_id = 4



	order_id	customer_id	order_status	bill_id	discount_id	address_delivery	final_price	order_date
<input type="checkbox"/>	1	3	0	1	NULL	123 Main St, Anytown USA	100000	2023-05-01
<input type="checkbox"/>	2	4	1	2	NULL	456 Oak Rd, Othertown USA	150000	2023-06-15
<input type="checkbox"/>	3	5	0	3	NULL	789 Elm St, Someplace USA	80000	2023-07-20
<input type="checkbox"/>	4	6	1	4	NULL	321 Maple Ave, Nowhere USA	530000	2023-08-05
<input type="checkbox"/>	5	7	2	5	NULL	654 Pine Rd, Everytown USA	120000	2023-09-10
<input type="checkbox"/>	6	8	0	6	NULL	987 Birch Blvd, Anotherton USA	90000	2023-10-01
<input type="checkbox"/>	7	9	1	7	NULL	159 Oak St, Somewhereville USA	180000	2023-11-15
<input type="checkbox"/>	8	10	0	8	NULL	753 Elm Ave, Nowhere USA	70000	2023-12-01
<input type="checkbox"/>	9	11	2	9	NULL	258 Maple Dr, Everytown USA	150000	2024-01-05
<input type="checkbox"/>	10	12	0	10	NULL	456 Pine St, Anotherton USA	110000	2024-02-10

Hình 36: Dữ liệu bảng order chưa trùm final_price sau khi thêm một bản ghi mới vào bảng create_order

2.2.3 Nhóm các trigger 3

Tính giá tạm tính (temp_price) trong creat_order khi insert/update trong bảng 'creat_order'

- Chức năng:** Các trigger này được thiết kế để tính toán giá tạm tính (temporary_price) của từng món ăn trong bảng 'creat_order' mỗi khi thêm, cập nhật một dòng dữ liệu.
- Các thao tác chính:**
 - Kiểm tra xem 'order_id' và 'food_id' có tồn tại trong các bảng liên quan.
 - Tính toán giá tạm tính dựa trên số lượng ('quantity') và giá của món ăn ('food_price').
- Câu lệnh tạo trigger: - Trigger Before Insert:**

```
1  DELIMITER $$
2
3  CREATE TRIGGER before_insert_creat_order
4    BEFORE INSERT ON creat_order
5    FOR EACH ROW
6    BEGIN
7      DECLARE item_price INT;
8
9      IF NOT EXISTS (SELECT 1 FROM `order` WHERE `order_id` = NEW.order_id) THEN
10        SIGNAL SQLSTATE '45000'
11        SET MESSAGE_TEXT = 'Error: order_id does not exist in `order` table';
12      END IF;
13
14      IF NOT EXISTS (SELECT 1 FROM `food` WHERE `food_id` = NEW.food_id) THEN
15        SIGNAL SQLSTATE '45000'
16        SET MESSAGE_TEXT = 'Error: food_id does not exist in `food` table';
17      END IF;
18
19      SELECT food_price INTO item_price
20      FROM food
21      WHERE food_id = NEW.food_id;
22
23      SET NEW.temp_price = NEW.quantity * item_price;
```



```
24 END;  
25 $$  
26 DELIMITER ;
```

- Trigger Before Update:

```
1 DELIMITER $$  
2  
3 CREATE TRIGGER before_update_creat_order  
4 BEFORE UPDATE ON creat_order  
5 FOR EACH ROW  
6 BEGIN  
7     DECLARE item_price INT;  
8  
9     IF NOT EXISTS (SELECT 1 FROM `order` WHERE `order_id` = NEW.order_id) THEN  
10        SIGNAL SQLSTATE '45000'  
11        SET MESSAGE_TEXT = 'Error: order_id does not exist in `order` table';  
12    END IF;  
13  
14    IF NOT EXISTS (SELECT 1 FROM `food` WHERE `food_id` = NEW.food_id) THEN  
15        SIGNAL SQLSTATE '45000'  
16        SET MESSAGE_TEXT = 'Error: food_id does not exist in `food` table';  
17    END IF;  
18  
19    SELECT food_price INTO item_price  
20    FROM food  
21    WHERE food_id = NEW.food_id;  
22  
23    SET NEW.temp_price = NEW.quantity * item_price;  
24 END;  
25 $$  
26 DELIMITER ;
```

- Câu lệnh kiểm tra trigger: Chèn một bản ghi mới vào bảng creat_order

```
1 INSERT INTO creat_order (order_id, food_id, quantity, creat_date)  
2 VALUES (7, 10, 3, CURDATE());
```



		order_id	food_id	quantity	temp_price	creat_date
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete			
		1	1	2	300000	2025-01-01
		1	2	1	120000	2025-01-01
		1	3	1	80000	2025-01-01
		2	4	1	200000	2025-01-15
		2	5	1	50000	2025-01-15
		3	6	1	120000	2025-02-20
		3	7	1	150000	2025-02-20
		4	1	1	150000	2024-12-08
		4	8	1	180000	2025-03-05
		4	9	1	200000	2025-03-05
		5	10	1	170000	2025-04-10
		5	11	1	110000	2025-04-10
		6	12	1	160000	2025-05-01
		6	13	1	190000	2025-05-01
		7	14	1	140000	2025-06-15
		7	15	1	130000	2025-06-15

Hình 37: Dữ liệu bảng creat_order trước khi thêm một bản ghi mới



✓ 1 row inserted. (Query took 0.0011 seconds.)

```
INSERT INTO creat_order (order_id, food_id, quantity, creat_date) VALUES (7, 10, 3, CURDATE());
```

[Edit inline] [Edit] [Create PHP code]

	← T →	▼	order_id	food_id	quantity	temp_price	creat_date
<input type="checkbox"/>		Edit		Copy		Delete	1 1 2 300000 2025-01-01
<input type="checkbox"/>		Edit		Copy		Delete	1 2 1 120000 2025-01-01
<input type="checkbox"/>		Edit		Copy		Delete	1 3 1 80000 2025-01-01
<input type="checkbox"/>		Edit		Copy		Delete	2 4 1 200000 2025-01-15
<input type="checkbox"/>		Edit		Copy		Delete	2 5 1 50000 2025-01-15
<input type="checkbox"/>		Edit		Copy		Delete	3 6 1 120000 2025-02-20
<input type="checkbox"/>		Edit		Copy		Delete	3 7 1 150000 2025-02-20
<input type="checkbox"/>		Edit		Copy		Delete	4 1 1 150000 2024-12-08
<input type="checkbox"/>		Edit		Copy		Delete	4 8 1 180000 2025-03-05
<input type="checkbox"/>		Edit		Copy		Delete	4 9 1 200000 2025-03-05
<input type="checkbox"/>		Edit		Copy		Delete	5 10 1 170000 2025-04-10
<input type="checkbox"/>		Edit		Copy		Delete	5 11 1 110000 2025-04-10
<input type="checkbox"/>		Edit		Copy		Delete	6 12 1 160000 2025-05-01
<input type="checkbox"/>		Edit		Copy		Delete	6 13 1 190000 2025-05-01
<input type="checkbox"/>		Edit		Copy		Delete	7 10 3 510000 2024-12-08
<input type="checkbox"/>		Edit		Copy		Delete	7 14 1 140000 2025-06-15
<input type="checkbox"/>		Edit		Copy		Delete	7 15 1 130000 2025-06-15

Hình 38: Chèn bản ghi mới ngày 08-12-2024 vào bảng creat_order thành công.

2.3 THỦ TỤC

2.3.1 Thủ tục 1

Thủ tục hiển thị thông tin khách hàng đã đánh giá một số sao cụ thể:

a. Mô tả thủ tục:

- **Chức năng:** Thủ tục GetCustomersByRating hiển thị thông tin khách hàng đã đánh giá một số sao cụ thể cho món ăn, bao gồm thông tin cá nhân và chi tiết đánh giá.
- **Các trường hợp sử dụng:**
 - Quản trị viên muốn kiểm tra danh sách khách hàng đánh giá cho món ăn.
 - Phân tích mức độ hài lòng của khách hàng dựa trên số sao họ đánh giá.
- **Các thao tác chính:**
 - Kiểm tra giá trị số sao nhập vào có hợp lệ không (trong khoảng từ 1 đến 5).
 - Truy vấn và hiển thị thông tin khách hàng cùng chi tiết đánh giá.
- **Các điều kiện kiểm tra:** Giá trị số sao (p_rating) phải trong khoảng từ 1 đến 5.



b. **Input:** p_rating: Số sao cần truy vấn (INT).

c. **Output:** Danh sách khách hàng đã đánh giá số sao cụ thể, bao gồm: customer_id, full_name, phone_number, email, food_id, rating, review_description.

d. Câu lệnh tạo thủ tục:

- Câu lệnh tạo thủ tục:

```
1  DELIMITER //
2
3  CREATE PROCEDURE GetCustomersByRating(
4      IN p_rating INT
5  )
6  BEGIN
7      IF p_rating NOT BETWEEN 1 AND 5 THEN
8          SIGNAL SQLSTATE '45000'
9          SET MESSAGE_TEXT = 'Sao nhp vo phi trong khong t 1 n 5';
10     END IF;
11
12    SELECT
13        c.customer_id,
14        CONCAT(c.customer_first_name, ' ', c.customer_last_name) AS full_name,
15        c.phone_number,
16        c.email,
17        ur.food_id,
18        ur.rating,
19        ur.review_description
20    FROM
21        customer c
22    INNER JOIN
23        user_review ur ON c.customer_id = ur.customer_id
24    WHERE
25        ur.rating = p_rating;
26 END //
27 DELIMITER ;
```

- Câu lệnh kiểm tra thủ tục: Trường hợp gọi thành công

```
1  CALL GetCustomersByRating(5);
```



✓ Showing rows 0 - 6 (7 total, Query took 0.0007 seconds.)

```
CALL GetCustomersByRating(5);
```

[Edit inline] [Edit] [Create PHP code]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

customer_id	full_name	phone_number	email	food_id	rating	review_description
3	John Doe	1234567890	john.doe@example.com	1	5	Pizza hải sản rất ngon, đầy đủ các loại hải sản tu...
6	Bob Brown	4567890123	bob.brown@example.com	4	5	Sushi cá hồi tuyệt vời, tươi ngon và cách trình bày...
6	Bob Brown	4567890123	bob.brown@example.com	6	5	Pizza Margherita ngon như ở Italia, cực kỳ hải lòn...
3	John Doe	1234567890	john.doe@example.com	8	5	Pizza hải sản đặc biệt quá tuyệt, nhiều hải sản và...
6	Bob Brown	4567890123	bob.brown@example.com	12	5	Pizza BBQ gà rất thơm ngon, vị cay nồng của ớt và ...
4	Jane Smith	2345678901	jane.smith@example.com	14	5	Pizza phô mai đặc biệt quá tuyệt, các loại phô mai...
6	Bob Brown	4567890123	bob.brown@example.com	1	5	Amazing Pizza!

Show all Number of rows: 25 Filter rows: Search this table

Hình 39: Gọi thủ tục với giá trị p_rating = 5, kết quả trả về danh sách khách hàng đã đánh giá 5 sao.

- Câu lệnh kiểm tra thủ tục: Trường hợp lỗi

```
1 CALL GetCustomersByRating(10);
```



Hình 40: Gọi thủ tục với giá trị p_rating = 10, lỗi p_rating nằm ngoài khoảng 1 đến 5.

2.3.2 Thủ tục 2

Thủ tục hiển thị danh sách các món ăn có giá thấp hơn một mức giá cụ thể được nhập vào.

a. Mô tả thủ tục:

- Chức năng:** Thủ tục GetFoodsBelowPrice hiển thị danh sách các món ăn có giá thấp hơn một mức giá cụ thể được nhập vào.



- Các trường hợp sử dụng:

- Tìm kiếm các món ăn phù hợp với ngân sách của khách hàng.
- Phân tích danh mục món ăn theo phân khúc giá.

- Các thao tác chính:

- Kiểm tra giá trị mức giá nhập vào có hợp lệ không (lớn hơn 0).
- Truy vấn và hiển thị danh sách món ăn thỏa mãn điều kiện.

- Các điều kiện kiểm tra: Giá trị mức giá (p_price) phải lớn hơn 0.

- b. Input: p_price: Mức giá cần truy vấn (INT).

- c. Output: Danh sách các món ăn có giá thấp hơn mức giá nhập vào, bao gồm: food_id, food_name, food_price, food_description.

- d. Câu lệnh tạo thủ tục:

- Câu lệnh tạo thủ tục:

```
1  DELIMITER //
2
3  CREATE PROCEDURE GetFoodsBelowPrice(
4      IN p_price INT
5  )
6  BEGIN
7      IF p_price <= 0 THEN
8          SIGNAL SQLSTATE '45000'
9          SET MESSAGE_TEXT = 'Giá tiền kim phi là 0.';
10     END IF;
11
12     SELECT
13         food_id,
14         food_name,
15         food_price,
16         food_description
17     FROM
18         food
19     WHERE
20         food_price < p_price;
21 END //
22 DELIMITER ;
```

- Câu lệnh kiểm tra thủ tục: Trường hợp thành công

```
1  CALL GetFoodsBelowPrice(150000);
```



Showing rows 0 - 6 (7 total, Query took 0.0004 seconds.)

```
CALL GetFoodsBelowPrice(150000);
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

food_id	food_name	food_price	food_description
2	Mì Ý Sốt Bolognese	120000	Mì Ý truyền thống với sốt thịt bò và gia vị đặc tr...
3	Salad Trái Cây	80000	Salad tươi ngon với nhiều loại trái cây mùa hè.
5	Bánh Mì Thịt Nướng	50000	Bánh mì giòn với thịt nướng và rau sống tươi mát.
6	Pizza Margherita	120000	Pizza truyền thống với sốt cà chua, phô mai mozzar...
11	Pizza Chay	110000	Pizza dành cho người ăn chay với rau củ tươi ngon ...
14	Pizza Phô Mai Đặc Biệt	140000	Pizza với nhiều loại phô mai khác nhau, dành cho t...
15	Pizza Trái Cây	130000	Pizza ngọt với trái cây tươi như dứa, kiwi và dâu ...

Hình 41: Gọi thủ tục với giá trị p_price = 150000, kết quả trả về danh sách các món ăn có giá thấp hơn .

- Câu lệnh kiểm tra thủ tục: Trường hợp lỗi

```
1 CALL GetFoodsBelowPrice(0);
```

Error

SQL query: [Copy](#)

```
CALL GetFoodsBelowPrice(0);
```

MySQL said: [?](#)

#1644 - Giá tìm kiếm phải lớn hơn 0.

Hình 42: Gọi thủ tục với giá trị p_price = 0, lỗi p_price <= 0

2.4 HÀM

2.4.1 Hàm 1

Hàm đếm số lượng khách hàng có điểm số trên một ngưỡng cụ thể:

a. Mô tả hàm:

- Chức năng:** Hàm CountCustomersWithPointsAbove đếm số lượng khách hàng có điểm số lớn hơn giá trị tối thiểu được nhập vào.



- Các trường hợp sử dụng:

- Quản trị viên muốn biết số lượng khách hàng có điểm số cao hơn một mức nhất định.
- Phân tích số lượng khách hàng dựa trên điểm số của họ.

- Các thao tác chính:

- Truy vấn và đếm số lượng khách hàng có điểm số lớn hơn `min_points`.

b. **Input:** `min_points`: Điểm số tối thiểu cần kiểm tra (INT).

c. **Output:** Số lượng khách hàng có điểm số lớn hơn `min_points` (INT).

d. Câu lệnh tạo hàm:

- Câu lệnh tạo hàm:

```
1  DELIMITER //
2
3  CREATE FUNCTION CountCustomersWithPointsAbove(min_points INT)
4  RETURNS INT
5  DETERMINISTIC
6  BEGIN
7      DECLARE customer_count INT;
8      IF min_points < 0 THEN
9          SIGNAL SQLSTATE '45000'
10         SET MESSAGE_TEXT = 'The min_points must be a non-negative integer';
11     END IF;
12
13     SELECT COUNT(*) INTO customer_count
14     FROM customer
15     WHERE points > min_points;
16
17     RETURN customer_count;
18 END //
19 DELIMITER ;
```

- Câu lệnh kiểm tra hàm:

```
1  SELECT CountCustomersWithPointsAbove(100);
```

- Hình ảnh kết quả gọi hàm:



The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** pizza
- Query Tab:** Contains the SQL query: `SELECT CountCustomersWithPointsAbove(100);`
- Status Bar:** Shows "Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)"
- Result Area:** Displays the result of the query: `CountCustomersWithPointsAbove(100)` with a value of `5`.
- Toolbar:** Includes buttons for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, and a refresh icon.
- Message Bar:** A warning message: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available." with a help icon.

Hình 43: Kết quả trả về khi gọi hàm CountCustomersWithPointsAbove(100)

- Câu lệnh kiểm tra hàm: Trường hợp lỗi

```
1   SELECT CountCustomersWithPointsAbove(-10);
```

- Hình ảnh lỗi:

The screenshot shows the MySQL Workbench interface with the following details:

- Query Tab:** Contains the SQL query: `SELECT CountCustomersWithPointsAbove(-10);`
- Error Message:** An error message is displayed in a pink box:
 - SQL query:** `Copy`
 - MySQL said:** `#1644 - The min_points must be a non-negative integer`

Hình 44: Kết quả lỗi khi gọi hàm CountCustomersWithPointsAbove(-10)



2.4.2 Hàm 2

Hàm tính tổng doanh thu của món ăn trong khoảng thời gian cụ thể:

a. Mô tả hàm:

- **Chức năng:** Hàm SaleInTime tính tổng doanh thu của món ăn theo food_id trong khoảng thời gian từ start_day đến end_day.
- **Các trường hợp sử dụng:**

- Quản trị viên muốn biết doanh thu của một món ăn trong một khoảng thời gian nhất định.
- Phân tích doanh thu theo thời gian cho từng món ăn.

• **Các thao tác chính:**

- Kiểm tra tính hợp lệ của food_id và các ngày nhập vào.
- Truy vấn và tính tổng doanh thu cho món ăn trong khoảng thời gian đã chỉ định.

b. Input:

- id: ID của món ăn (INT).
- start_day: Ngày bắt đầu (DATE).
- end_day: Ngày kết thúc (DATE).

c. Output: Tổng doanh thu của món ăn trong khoảng thời gian (INT).

d. Câu lệnh tạo hàm:

- Câu lệnh tạo hàm:

```
1  DELIMITER //
2
3  CREATE FUNCTION SaleInTime(id INT, start_day DATE, end_day DATE)
4  RETURNS INT
5  DETERMINISTIC
6  BEGIN
7      DECLARE total_sale INT DEFAULT 0;
8      DECLARE done INT DEFAULT FALSE;
9      DECLARE current_sale INT;
10
11     DECLARE sale_cursor CURSOR FOR
12         SELECT temp_price FROM creat_order WHERE food_id = id AND creat_date
13             BETWEEN start_day AND end_day;
14
15     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
16
17     IF id < 0 THEN
18         SIGNAL SQLSTATE '45000'
19         SET MESSAGE_TEXT = 'The food_id must be a non-negative integer';
20     END IF;
21
22     IF start_day IS NULL OR end_day IS NULL THEN
23         SIGNAL SQLSTATE '45000'
```

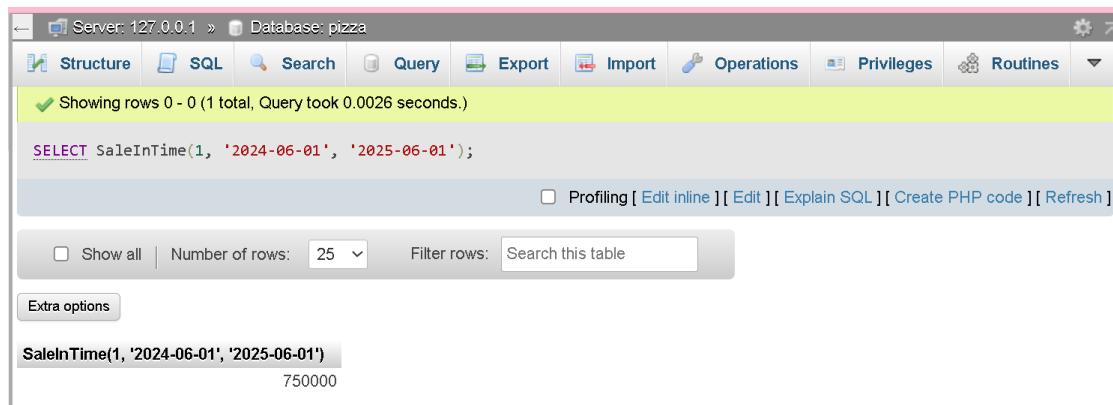


```
23     SET MESSAGE_TEXT = 'The start_day and end_day must be valid dates';
24 END IF;
25
26 IF start_day > end_day THEN
27     SIGNAL SQLSTATE '45000'
28     SET MESSAGE_TEXT = 'The start_day must be before or equal to the end_day';
29 END IF;
30
31 OPEN sale_cursor;
32
33 read_loop: LOOP
34     FETCH sale_cursor INTO current_sale;
35     IF done THEN
36         LEAVE read_loop;
37     END IF;
38
39     SET total_sale = total_sale + current_sale;
40 END LOOP;
41
42 CLOSE sale_cursor;
43
44 RETURN total_sale;
45 END //
46 DELIMITER ;
```

- Câu lệnh kiểm tra hàm:

```
1   SELECT SaleInTime(1, '2024-06-01', '2025-06-01');
```

- Hình ảnh kết quả gọi hàm:



Hình 45: Kết quả trả về khi gọi hàm `SELECT SaleInTime(1, '2024-06-01', '2025-06-01');`

- Câu lệnh kiểm tra hàm: Trường hợp lỗi

```
1   SELECT SaleInTime(-1, '2024-06-01', '2025-06-01');
```



- Hình ảnh lỗi:

Error

SQL query: [Copy](#)

```
SELECT SaleInTime(-1, '2024-06-01', '2025-06-01');
```

MySQL said: #1644 - The food_id must be a non-negative integer

Hình 46: Kết quả trả về lỗi khi gọi hàm SELECT SaleInTime(-1, '2024-06-01', '2025-06-01');



3 ỨNG DỤNG KẾT NỐI VỚI CSDL

3.1 Tổng quan công nghệ

Công nghệ sử dụng: MySQL, PHP.

Kiến trúc ứng dụng: Monolithic.

3.2 Các màn hình demo

Trình bày các màn hình của ứng dụng. Mỗi màn hình cần có:

3.2.1 Màn hình 1: Đặt món ăn

Mô tả các chức năng có thể thực hiện trên màn hình:

- Chọn món ăn từ danh sách. - Nhập giá món tối đa.

Mô tả đoạn code gọi API và render dữ liệu từ response của API:

```
fetch('/api/menu')
    .then(response => response.json())
    .then(data => {
        // Render danh sách món ăn
    });
}
```

Mô tả API

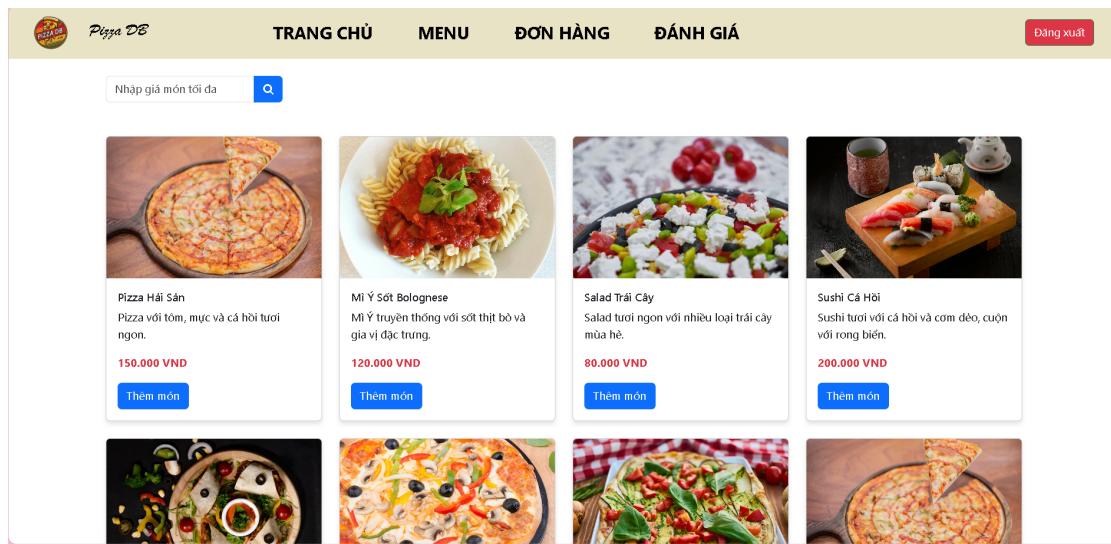
- API lấy danh sách món ăn:

```
1 $sql = "
2     SELECT
3         f.food_id,
4         f.food_name,
5         f.food_price,
6         f.food_description,
7         fi.food_image_url
8     FROM
9         food f
10    LEFT JOIN
11        food_image fi
12    ON
13        f.food_id = fi.food_id
14    WHERE
15        1 = 1";
16
17    if ($max_price !== null) {
18        $sql .= " AND f.food_price <= $max_price";
19    }
20
21    $sql .= "
22        GROUP BY
23            f.food_id
24        LIMIT $products_per_page
25        OFFSET $offset";
```



26

```
$result = mysqli_query($connect, $sql);
```



Hình 47: Ảnh chụp màn hình 1: Giao diện đặt món ăn

3.2.2 Màn hình 2: Xem đơn hàng

Mô tả các chức năng có thể thực hiện trên màn hình:

- Hiển thị chi tiết đơn hàng. - Cập nhật số lượng món ăn. - Hủy đơn hàng.

Dish Name	Price	Quantity	Temporary Total
Pizza Hải Sản	150,000 VND	1	150,000 VND
Mi Ý Sốt Bolognese	120,000 VND	3	360,000 VND
Salad Trái Cây	80,000 VND	1	80,000 VND
Sushi Cá Hồi	200,000 VND	1	200,000 VND

Hình 48: Ảnh chụp màn hình 2: Giao diện xem đơn hàng



Mô tả đoạn code gọi API và render dữ liệu từ response của API:

```
1 $sql = "SELECT
2     c.food_id,
3     f.food_name,
4     f.food_price,
5     c.quantity,
6     c.temp_price
7   FROM
8     creat_order c
9   JOIN
10    food f
11  ON
12    c.food_id = f.food_id
13  JOIN
14    `order` o
15  ON
16    c.order_id = o.order_id
17 WHERE
18    c.order_id = ? AND o.order_status = 0";
19$stmt = mysqli_prepare($connect, $sql);
20if ($stmt) {
21    mysqli_stmt_bind_param($stmt, "i", $order_id);
22    mysqli_stmt_execute($stmt);
23    $result = mysqli_stmt_get_result($stmt);
24
25    // Check if the result is valid
26    if ($result) {
27        // Process the result if needed
28    } else {
29        echo "Error: " . mysqli_error($connect);
30    }
31    mysqli_stmt_close($stmt);
32} else {
33    echo "Error: " . mysqli_error($connect);
34}
35
36    // Calculate the total temporary price
37    $total_query = "SELECT SUM(temp_price) AS total_price FROM creat_order WHERE order_id
38    = ?";
39    $total_stmt = mysqli_prepare($connect, $total_query);
40    if ($total_stmt) {
41        mysqli_stmt_bind_param($total_stmt, "i", $order_id);
42        mysqli_stmt_execute($total_stmt);
43        $total_result = mysqli_stmt_get_result($total_stmt);
44
45        // Check if the result is valid
46        if ($total_result) {
47            $total_row = mysqli_fetch_assoc($total_result);
48            if ($total_row) {
49                $total_price = $total_row['total_price'];
50            } else {
50                $total_price = 0; // Default value if no rows are returned
```



```
51         }
52     } else {
53         echo "Error: " . mysqli_error($connect);
54     }
55     mysqli_stmt_close($total_stmt);
56 } else {
57     echo "Error: " . mysqli_error($connect);
58 }
59
60 // X lý m gim gi
61 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['coupon_code'])) {
62     $coupon_code = $_POST['coupon_code'];
63
64     $query = "
65         SELECT cd.type_discount, cd.expired_date, cd.discount_id, d.status_use,
66             dop.percent, don.discount_number
67         FROM customer_discounts cd
68         LEFT JOIN discount_on_percent dop ON cd.discount_id = dop.discount_id
69         LEFT JOIN discount_on_number don ON cd.discount_id = don.discount_id
70         JOIN discount d ON cd.discount_id = d.discount_id
71         WHERE cd.customer_id = ?
72             AND d.discount_name = ?
73             AND cd.expired_date > CURDATE()
74             AND d.status_use = 0
75     ";
76
77     $stmt = mysqli_prepare($connect, $query);
78     mysqli_stmt_bind_param($stmt, "is", $customer_id, $coupon_code);
79     mysqli_stmt_execute($stmt);
80     $discount_result = mysqli_stmt_get_result($stmt);
81
82     if ($discount_row = mysqli_fetch_assoc($discount_result)) {
83         if ($discount_row['type_discount'] === 'percent') {
84             $discount = ($total_price * $discount_row['percent']) / 100;
85         } elseif ($discount_row['type_discount'] === 'number') {
86             $discount = $discount_row['discount_number'];
87         }
88         $discount = min($discount, $total_price);
89         $total_to_pay = $total_price + $delivery_fee - $discount;
90         $valid_discount = true;
91     } else {
92         $error_message = "Invalid or expired coupon code.";
93         $total_to_pay = $total_price + $delivery_fee;
94     }
95 } else {
96     $total_to_pay = $total_price + $delivery_fee;
97 }
```

3.2.3 Màn hình 3: Quản lý món ăn

Mô tả các chức năng có thể thực hiện trên màn hình:

- Thêm món ăn mới. - Sửa thông tin món ăn. - Xóa món ăn.



The screenshot shows a user interface for a food delivery service. At the top, there is a navigation bar with links for 'Pizza DB', 'NGƯỜI DÙNG', 'MÓN ĂN', 'ĐIỂM THƯỞNG', 'BÌNH LUẬN', 'DOANH SỐ', and a 'Đăng xuất' button. Below the navigation bar is a search bar with placeholder text 'Nhập tên món ăn...' and a 'Tìm kiếm' button. A green button labeled 'Thêm món mới' is located above the search bar. The main content area displays a grid of food items with their names, descriptions, prices, and two small buttons ('Xóa' and 'Sửa') for each item.

Món Ăn	Mô Tả	Giá	��作
Pizza Hải Sản	Pizza với tôm, mực và cá hồi tươi ngon.	150.000 VND	Xóa Sửa
Mì Ý Sốt Bolognese	Mì Ý truyền thống với sốt thịt bò và giá vị đặc trưng.	120.000 VND	Xóa Sửa
Salad Trái Cây	Salad tươi ngon với nhiều loại trái cây mùa hè.	80.000 VND	Xóa Sửa
Sushi Cá Hồi	Sushi tươi với cá hồi và cơm dẻo, cuộn với rong biển.	200.000 VND	Xóa Sửa
(Image 1)	(Image 2)	(Image 3)	(Image 4)

Hình 49: Ảnh chụp màn hình 3: Giao diện quản lý món ăn

Mô tả đoạn code Truy vấn món ăn dựa trên tìm kiếm

```
1 $search_query = isset($_GET['query']) ? $_GET['query'] : null;
2 $sql = "
3     SELECT
4         f.food_id,
5         f.food_name,
6         f.food_price,
7         f.food_description,
8         fi.food_image_url
9     FROM
10        food f
11    LEFT JOIN
12        food_image fi
13    ON
14        f.food_id = fi.food_id
15    WHERE
16        f.food_name LIKE ?
17    GROUP BY
18        f.food_id
19    LIMIT $products_per_page
20    OFFSET $offset";
21    $stmt = $connect->prepare($sql);
22    $search_term = '%' . $search_query . '%';
23    $stmt->bind_param("s", $search_term);
24    $stmt->execute();
25    $result = $stmt->get_result();
```



3.2.4 Màn hình 4: Quản lý người dùng

Mô tả các chức năng có thể thực hiện trên màn hình:

- Xem danh sách người dùng.
- Thêm người dùng mới.
- Cập nhật thông tin người dùng.

ID	Họ	Tên	Số Điện Thoại	Email	Điểm	Hành Động
3	John	Doe	1234567890	john.doe@example.com	1000	<button>Sửa</button> <button>Xóa</button>
4	Jane	Smith	2345678901	jane.smith@example.com	200	<button>Sửa</button> <button>Xóa</button>
5	Alice	Johnson	3456789012	alice.johnson@example.com	50	<button>Sửa</button> <button>Xóa</button>
6	Bob	Brown	4567890123	bob.brown@example.com	0	<button>Sửa</button> <button>Xóa</button>
7	Charlie	Wilson	5678901234	charlie.wilson@example.com	150	<button>Sửa</button> <button>Xóa</button>
8	Diana	Moore	6789012345	diana.moore@example.com	75	<button>Sửa</button> <button>Xóa</button>
9	Ethan	Taylor	7890123456	ethan.taylor@example.com	300	<button>Sửa</button> <button>Xóa</button>
10	Fiona	Anderson	8901234567	fiona.anderson@example.com	335	<button>Sửa</button> <button>Xóa</button>

Hình 50: Ảnh chụp màn hình 4: Giao diện quản lý người dùng

Mô tả đoạn code các thao tác sửa, xóa, tìm kiếm khách hàng

```
1 $connect = mysqli_connect('localhost', 'root', '', 'pizza');
2     if (!$connect) {
3         die("Connection failed: " . mysqli_connect_error());
4     }
5     mysqli_set_charset($connect, "utf8");
6
7     // X lý sa khch hng
8     if (isset($_POST['edit_customer'])) {
9         $customer_id = intval($_POST['customer_id']);
10        $first_name = mysqli_real_escape_string($connect, $_POST['first_name']);
11        $last_name = mysqli_real_escape_string($connect, $_POST['last_name']);
12        $phone_number = mysqli_real_escape_string($connect, $_POST['phone_number']);
13        $email = mysqli_real_escape_string($connect, $_POST['email']);
14        $points = intval($_POST['points']);
15
16        // Use prepared statements to prevent SQL injection
17        $update_query = "UPDATE customer SET customer_first_name='$first_name',
18                                         customer_last_name='$last_name', phone_number='$phone_number',
19                                         email='$email', points=$points WHERE customer_id=$customer_id";
20        if (mysqli_query($connect, $update_query)) {
21            echo "<script>alert('Sa khch hng thnh cng !'); window.location.href =
22                  'admin_manager_user.php';</script>";
23        } else {
```



```
22         echo "<script>alert('Li    sa    khch   hng : " . $customer_id . "');</script>";
23     }
24 }
25
26 // X lý xa khch hng
27 if (isset($_POST['delete_customer'])) {
28     $customer_id = intval($_POST['customer_id']);
29     $delete_query = "DELETE FROM customer WHERE customer_id=$customer_id";
30
31     if (mysqli_query($connect, $delete_query)) {
32         echo "<script>alert('Xa khch hng thnh cng !');
33             window.location.reload();</script>";
34     } else {
35         echo "<script>alert('Li    xa    khch   hng : " . mysqli_error($connect) .
36             "');</script>";
37     }
38 }
39
40 // X lý tm kim
41 $search_query = '';
42 if (isset($_POST['search'])) {
43     $search_term = mysqli_real_escape_string($connect, $_POST['search_term']);
44     $search_query = "WHERE customer_first_name LIKE '%$search_term%' OR
45                     customer_last_name LIKE '%$search_term%' OR phone_number LIKE
46                     '%$search_term%' OR email LIKE '%$search_term%' OR points LIKE
47                     '%$search_term%'";
48 }
```

3.2.5 Màn hình 5: Thông kê doanh thu

Mô tả các chức năng có thể thực hiện trên màn hình:

- Hiển thị biểu đồ doanh thu theo ngày. - Lọc doanh thu theo thời gian.



Pizza DB

NGƯỜI DÙNG MÓN ĂN ĐIỂM THƯỞNG BÌNH LUẬN DOANH SỐ Đăng xuất

Quản lý doanh số

Lựa chọn món ăn:

Lựa chọn...

Ngày bắt đầu:

mm/dd/yyyy

Ngày kết thúc:

mm/dd/yyyy

Lấy doanh số

Kết quả bán hàng:

Tổng doanh số bán hàng của sản phẩm: 450.000 VNĐ

Hình 51: Ảnh chụp màn hình 5: Giao diện thống kê doanh thu

Mô tả đoạn code Tính doanh số

```
1 $connect = mysqli_connect('localhost', 'root', '', 'pizza');
2 if (!$connect) {
3     die("Connection failed: " . mysqli_connect_error());
4 }
5 mysqli_set_charset($connect, "utf8");
6
7 // Fetch list of dishes for the dropdown
8 $dishes = [];
9 $dish_query = "SELECT food_id, food_name FROM food";
10 $dish_result = mysqli_query($connect, $dish_query);
11 while ($row = mysqli_fetch_assoc($dish_result)) {
12     $dishes[] = $row;
13 }
14
15 // Initialize variables
16 $sales_result = null;
17 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
18     $food_id = $_POST['food_id'];
19     $start_date = $_POST['start_date'];
20     $end_date = $_POST['end_date'];
21
22     // Run the MySQL function SaleInTime
23     $stmt = mysqli_prepare($connect, "SELECT SaleInTime(?, ?, ?) AS sales");
24     if ($stmt) {
25         mysqli_stmt_bind_param($stmt, "iss", $food_id, $start_date, $end_date);
26         mysqli_stmt_execute($stmt);
27         $result = mysqli_stmt_get_result($stmt);
28         if ($result) {
29             $sales_result = mysqli_fetch_assoc($result)['sales'];
30         } else {
```



```
31         echo "Error: " . mysqli_error($connect);
32     }
33     mysqli_stmt_close($stmt);
34 } else {
35     echo "Error: " . mysqli_error($connect);
36 }
37 }
```



BẢNG KÊ NHIỆM VỤ

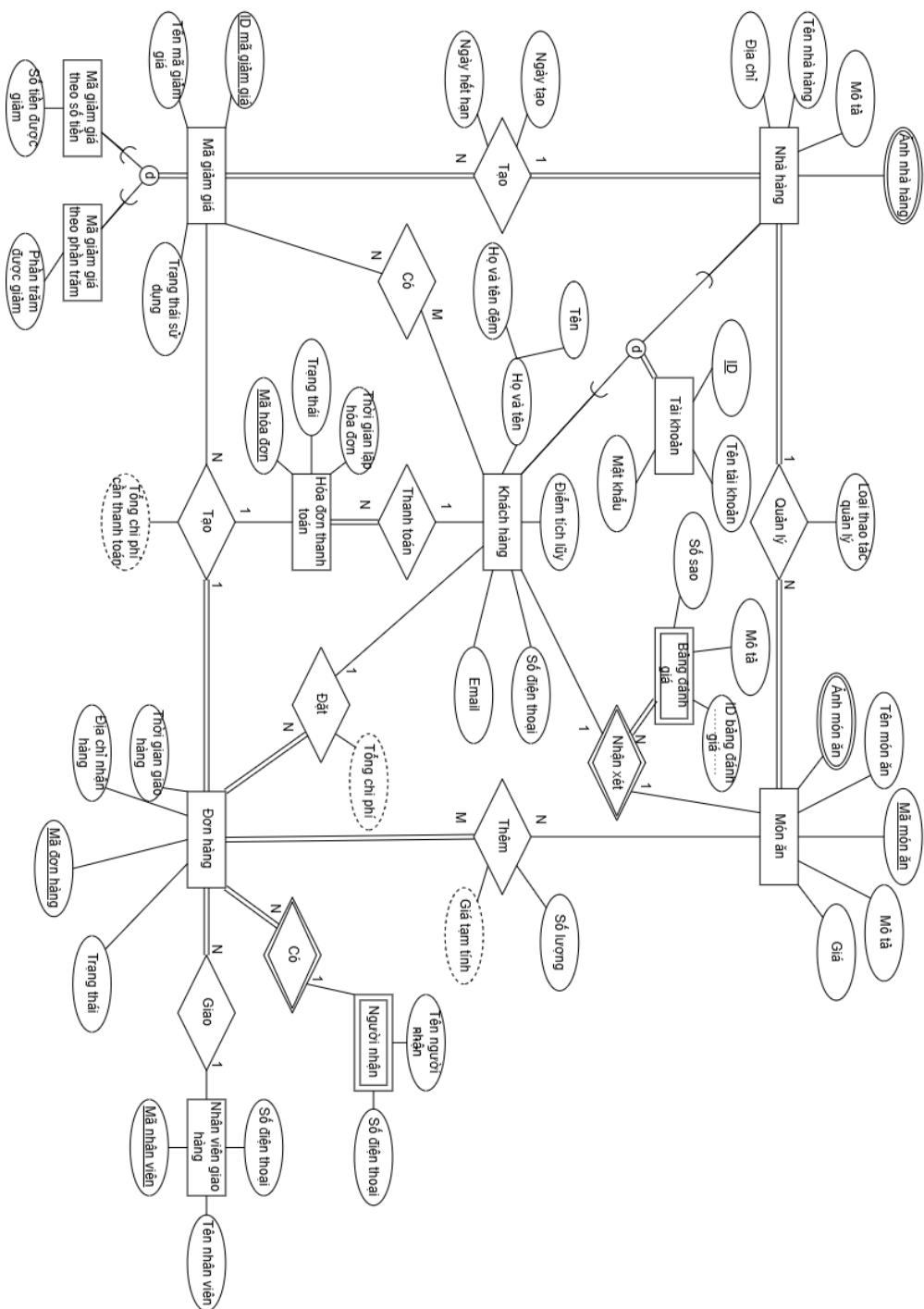
STT	Họ và tên	MSSV	Phân công	Ký tên
1	Nguyễn Khánh Lộc	2211927	<ul style="list-style-type: none">- Viết các câu lệnh tạo bảng- Viết hàm tính doanh số.- Ứng dụng: Trang chủ (chung), trang đăng nhập (chung), trang hiển thị các khách hàng có điểm lớn hơn một số được nhập (admin), trang tính doanh số, trang hiển thị cho thủ tục hiển thị thông tin các khách hàng đã đánh giá x sao cho một món ăn (admin) - Fullstack.	
2	Lê Thị Hoàng Anh	2210070	<ul style="list-style-type: none">- Thiết kế giao diện- Viết trigger 1, 2, 3.- Ứng dụng: Trang đơn hàng (user), trang hiển thị các đơn hàng đã giao nhưng chưa được đánh giá (user), trang đánh giá món ăn dành cho khách hàng có đơn hàng đã được giao (user) - Fullstack- Viết báo cáo	
3	Nguyễn Minh Thành	2433043	<ul style="list-style-type: none">- Viết các thủ tục insert/update/delete dữ liệu vào bảng 'customer'- Viết thủ tục 1, 2.- Ứng dụng: Trang menu (user), chức năng insert/update/delete dữ liệu món ăn (admin), trang hiển thị các món ăn với có giá dưới một số được nhập (admin) - Fullstack	
4	Nguyễn Quốc Kiệt	2211762	<ul style="list-style-type: none">- Tạo dữ liệu mẫu- Viết hàm đếm số lượng khách hàng có số điểm lớn hơn một số được nhập- Ứng dụng: Trang quản lý người dùng (admin), chức năng insert/update/delete dữ liệu khách hàng (admin) - Fullstack- Viết báo cáo	



ĐÁNH GIÁ CỦA GIÁO VIÊN

STT	Họ và tên	MSSV	% Điểm	Nhận xét	Điểm trừ	Điểm cộng	Tổng cộng
1	Nguyễn Khánh Lộc	2211927	25%				
2	Lê Thị Hoàng Anh	2210070	25%				
3	Nguyễn Minh Thành	2433043	25%				
4	Nguyễn Quốc Kiệt	2211762	25%				

[PHỤ LỤC]



Hình 52: Thiết kế ERD



Bảng 1. Bảng ghi chú các thay đổi so với BTL1

Phần thay đổi	Nội dung BTL1	Nội dung BTL2
Quan hệ 'đơn hàng'	(Mã đơn hàng, Mã khách hàng, Địa chỉ giao hàng, Thời gian tạo)	(Mã đơn hàng, Mã khách hàng, Địa chỉ giao hàng, Thời gian tạo, ID mã giảm giá, Mã hóa đơn, Giá thanh toán, Trạng thái)
Quan hệ 'tạo đơn'	(Mã đơn hàng, Mã món ăn, Giá thanh toán, Thời gian tạo, ID mã giảm giá, Mã hóa đơn)	(Mã đơn hàng, Mã món ăn, Giá thanh toán, Thời gian tạo, Số lượng)

Bảng 2. Bảng đổi chiếu tên các quan hệ trong ánh xạ lược đồ cơ sở dữ liệu và trong các bảng trong database:

STT	Tên quan hệ trong ánh xạ lược đồ CSDL	Tên bảng trong database
1	Tài khoản	account
2	Tạo đơn	creat_order
3	Khách hàng	customer
4	Mã giảm giá hiện có	customer_discounts
5	Mã giảm giá	discount
6	Tạo mã giảm giá	discount_creator
7	Mã giảm giá theo số tiền	discount_on_number
8	Mã giảm giá theo phần trăm	discount_on_percent
9	Món ăn	food
10	Ảnh món ăn	food_image
11	Quản lý món ăn	manage_food
12	Đơn hàng	order
13	Hóa đơn thanh toán	payment_order
13	Người nhận	receiver
14	Nhà hàng	restaurant
15	Ảnh nhà hàng	restaurant_image
16	Nhân viên giao hàng	shipping_employee
16	Dánh giá món ăn	user_review