




BÁO CÁO BÀI TẬP LỚN SỐ 2 MÔN HỆ CƠ SỞ DỮ LIỆU

HK231 - Lớp: **L04**

Nhóm: Cafe dữ liệu

Danh sách thành viên:

Stt	Họ tên	MSSV	Nội dung thực hiện	Ký tên xác nhận
1	Nguyễn Minh Điềm	2111056	Nhóm trigger 2, ứng dụng, báo cáo, các thủ tục hỗ trợ web	
2	Lê Nguyên Chương	2112945	Trigger 1, hàm 2, ứng dụng, thủ tục 2, thủ tục 1	
3	Hồ Huy Hoàng	2110181	Tạo dữ liệu mẫu, thủ tục Update bảng, ứng dụng, thủ tục tìm kiếm theo tên	
4	Võ Văn Khả	2110264	Tạo bảng, thủ tục Insert, Delete bảng, ứng dụng	
5	Nguyễn Lê Phúc	2112048	Tạo bảng, hàm 1, ứng dụng, thủ tục sắp xếp theo tên	

Chủ đề: Hệ thống quản lý quán cafe

1. Tạo bảng và dữ liệu mẫu

1.1 Các câu lệnh tạo bảng và ràng buộc

```
drop table if exists nhanvien;

CREATE TABLE NHANVIEN (

    MaNV INT auto_increment PRIMARY KEY,

    HoVaTen VARCHAR(50) NOT NULL,

    email      VARCHAR(100) NOT NULL UNIQUE,

    mk         VARCHAR(100) not null,

    DiaChi VARCHAR(50),

    NgaySinh DATE,

    SĐT VARCHAR(10),
```

```

        MaGiamSat INT,

        Luong INT,

        statusNV INT default 1,

        CONSTRAINT fk_NV_GIAMSAT FOREIGN KEY (MaGiamSat)
            REFERENCES NHANVIEN (MaNV) ON DELETE NO ACTION
    ) auto_increment 1
    auto_increment 1;

drop trigger if exists check_age_and_salary ;
DELIMITER //

CREATE TRIGGER check_age_and_salary
BEFORE INSERT ON NHANVIEN
FOR EACH ROW
BEGIN
    IF YEAR(NOW()) - YEAR(NEW.NgaySinh) <= 16 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tuổi phải lớn
hơn 16.';
    END IF;

    IF NEW.Luong < 20000 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Luong phải lớn
hơn hoặc bằng 20000.';
    END IF;
END//
DELIMITER ;

drop table if exists DonHang;
create table DonHang (
    MaDonHang INT auto_increment primary key,
    SoLuongMon INT default 0,

```

```

        TongGiaGoc INT default 0,
        TongGiaSauUuDai INT default 0
    )auto_increment 1
    auto_increment 1;

drop table if exists dondattruoc;
create table DonDatTruoc (
    MaDonHang INT primary key,
    MaBan INT,
    constraint Fk_dondattruoc foreign key (MaDonHang) references
    DonHang(MaDonHang) on delete cascade
);

drop table if exists donhangcanhan;
create table DonHangCaNhan (
    MaDonHang INT not null primary key,
    MaKhachHang INT not null,
    Sothe INT,
    constraint Fk_donhangcanhan foreign key (MaDonHang)
    references DonHang(MaDonHang) on delete cascade
);

drop table if exists hoadon;
CREATE TABLE HOADON (
    MaHD INT auto_increment primary KEY,
    tongGia INT default 0
)auto_increment 1 auto_increment 1;

```

```

drop table if exists xuat;

CREATE TABLE XUAT (
    MaNV INT NOT NULL,
    MaHD INT NOT NULL,
    MaDH INT NOT NULL,
    NgayXuat DATE ,
    GioXuat TIME,
    PRIMARY KEY (MaNV, MaHD, MaDH),
    CONSTRAINT fk_X_NHANVIEN FOREIGN KEY (MaNV)
        REFERENCES NHANVIEN (MaNV) ,
    CONSTRAINT fk_X_HOADON FOREIGN KEY (MaHD)
        REFERENCES HOADON (MaHD) ,
    CONSTRAINT fk_X_DONHANG FOREIGN KEY (MaDH)
        REFERENCES DonHang (MaDonHang)
);

-- trigger dùng để cập nhật tổng tiền trong hóa đơn
drop trigger if exists updateHoaDon;
delimiter //
create trigger updateHoaDon
before insert on xuat
for each row
begin
    declare price int;

    select tongGiaSauUuDai into price
    from donhang
    where madonhang = new.MaDH;

    update hoadon

```

```

        set tonggia = price
        where MaHD=new.MaHD;
end //
delimiter ;

-- trigger dùng để xóa hóa đơn
drop trigger if exists setdefaultHoaDon;
delimiter //
create trigger setdefaultHoaDon
before delete on xuất
for each row
begin
    update hoadon
    set tonggia = 0
    where MaHD=old.MaHD;
end //
delimiter ;

drop table if exists VOUCHER_TYPE;
CREATE TABLE VOUCHER_TYPE(
    id INT auto_increment KEY,
    discount_percent INT not null,
    note VARCHAR(200),
    max_discount INT,
    number INT default '0'
) auto_increment 1
auto_increment 1;

```

```

drop table if exists VOUCHER_CARD;

CREATE TABLE VOUCHER_CARD(
    id INT,
    STT INT,
    current_status INT default 1,
    PRIMARY KEY(id,STT),
    constraint Fk_voucher_card foreign key (id) references
VOUCHER_TYPE(id) ON DELETE CASCADE
);

drop table if exists voucher_effective_time;

CREATE TABLE VOUCHER_EFFECTIVE_TIME(
    id INT,
    start_time DATE,
    end_time DATE,
    PRIMARY KEY(id),
    constraint Fk_effective_time foreign key (id) references
VOUCHER_TYPE(id) ON DELETE CASCADE
);

drop table if exists payment_apply;

CREATE TABLE PAYMENT_APPLY(
    order_id INT unique not null,
    STT_voucher INT,
    id_voucher INT,
    PRIMARY KEY(STT_voucher,id_voucher),
    constraint Fk_payment_apply1 foreign key
(id_voucher,STT_voucher) references VOUCHER_CARD(id,STT) ON
DELETE CASCADE,

```

```
        constraint Fk_payment_apply2 foreign key (order_id) references
DonHang(MaDonHang)  ON DELETE CASCADE
);
```

```
drop table if exists customers;
```

```
CREATE TABLE customers (
    ID INT auto_increment PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(300) NOT NULL,
    CONSTRAINT uniqueCustomer UNIQUE(name, address)
) auto_increment=1
auto_increment=1;
```

```
drop table if exists customerPN;
```

```
CREATE TABLE customerPN (
    id INT NOT NULL auto_increment PRIMARY KEY,
    customerID INT NOT NULL,
    phoneNumber VARCHAR(50) NOT NULL,
    FOREIGN KEY (customerID) REFERENCES customers(ID) ON DELETE
CASCADE ON UPDATE CASCADE,
    CONSTRAINT phoneLength CHECK (length(phoneNumber) >= 10 AND
length(phoneNumber) <= 11)
) auto_increment 1
auto_increment 1;
```

```
drop table if exists bookingTables;
```

```
CREATE TABLE bookingTables (
    ID INT auto_increment PRIMARY KEY,
```

```

        seats INT NOT NULL,

        current_status INT NOT NULL DEFAULT 1

        CONSTRAINT enum_current_status    CHECK(current_status in
('0','1'))
    )auto_increment 1
auto_increment 1;
drop table if exists bookingInfo;
CREATE TABLE bookingInfo (
    ID INT auto_increment PRIMARY KEY,
    fk_tableID INT    DEFAULT 0,
    fk_customerID INT    DEFAULT 0,
    orderDate DATE NOT NULL,
    orderTime time not null,
    startDate date not null,
    startTime TIME NOT NULL,
    endTime TIME NOT NULL,
    CONSTRAINT startTime_endTime CHECK (startTime < endTime),
    constraint Fk_booking1 FOREIGN KEY (fk_tableID) REFERENCES
bookingTables(ID) ON DELETE SET null ON UPDATE CASCADE,
    constraint Fk_booking2 FOREIGN KEY (fk_customerID) REFERENCES
customers(ID) ON DELETE SET null ON UPDATE CASCADE
);

drop table if exists LoaiMon;
CREATE TABLE LoaiMon (
    Maloaimon INT    auto_increment    not null,
    Mota VARCHAR(255),
    Ten VARCHAR(50) not null,
    Giobatdau TIME,
    Gioketthuc TIME,

```



```

        Loaimon    VARCHAR(255) not null DEFAULT 'Do uong',
        CONSTRAINT    enum_Loai_mon    CHECK(Loaimon in ('Do an', 'Do
uong')),
        PRIMARY KEY (Maloaimon)
) auto_increment 1
auto_increment 1;
drop table if exists mon;
CREATE TABLE Mon (
        Maloaimon INT not null,

        Kichco    VARCHAR(255) not null DEFAULT 'M',
        Dongia    INT not null,
        current_status INT    default 1 ,
        CONSTRAINT enum_Kich_co1    CHECK(Kichco in ('M','L','XL')),
        CONSTRAINT fk_Ma_loai_mon    FOREIGN KEY (Maloaimon)
                                REFERENCES loaimon(Maloaimon) ON DELETE
CASCADE,
        PRIMARY KEY (Maloaimon, Kichco)
);

CREATE TABLE thuocvemon (
        Mamon INT not null,
        Madonhang INT not null,
        Kichco    CHAR(1) not null DEFAULT 'M',
        Giatheongay    INT not null,
        Soluong    INT not null,
        CONSTRAINT    enum_Kich_co2    CHECK(Kichco in
('M','L','XL')),
        CONSTRAINT    fk_Ma_don_hang    FOREIGN KEY (Madonhang)
REFERENCES donhang(madonhang) ON DELETE CASCADE,
        CONSTRAINT    fk_thuocvemon    FOREIGN KEY (Mamon, Kichco)

```

```

REFERENCES mon(Maloaimon, Kichco) ON DELETE
CASCADE,
    PRIMARY KEY (Mamon, Madonhang, Kichco)
);

CREATE TABLE dieukienapdung (
    Maloaimon INT not null,
    Mavoucher INT not null,
    Kichco      CHAR(1) not null DEFAULT 'M',
    minNum      INT not null,
    CONSTRAINT  enum_Kich_co3    CHECK(Kichco in
('M', 'L', 'XL')),
    CONSTRAINT  fk_dkapdung FOREIGN KEY (Maloaimon, Kichco)
REFERENCES mon(Maloaimon, Kichco) ON DELETE
CASCADE,
    CONSTRAINTfk_Voucher FOREIGN KEY (Mavoucher)
REFERENCES VOUCHER_TYPE(id),
    PRIMARY KEY (Maloaimon, Mavoucher, Kichco)
);

-- thêm các khóa ngoại còn thiếu
alter table dondattruoc add constraint Fk_dondattruoc1 foreign key
(maban) references bookingTables(ID) on delete set null;
alter table DonHangCaNhan add constraint Fk_donhangcanhan1 foreign
key (MaKhachHang) references customers(ID) on delete cascade;

```

1.2 Câu lệnh thêm dữ liệu

```
INSERT INTO ...
```

Kết quả dữ liệu của các bảng trong CSDL.

	ID	seats	current_status
▶	1	10	1
	2	10	1
	3	10	1
	4	10	1
	5	10	1
	6	10	1
	7	10	1
	8	10	1
	9	10	1
	10	10	1
	11	20	1
	12	20	1
	13	20	1
	14	20	1
	15	20	1
	16	30	1
	17	30	1
	18	30	1
	19	30	1
	20	30	1
✱	NULL	NULL	NULL

bookingtables 1 ×

1. Bảng bookingtables

	ID	fk_tableID	fk_customerID	orderDate	orderTime	startDate	startTime	endTime
▶	1	1	1	2023-01-01	12:00:00	2023-01-01	12:00:00	14:00:00
	2	2	2	2023-02-01	18:00:00	2023-02-01	18:00:00	20:00:00
	3	3	3	2023-03-01	14:00:00	2023-03-01	14:00:00	16:00:00
	4	4	4	2023-04-01	20:00:00	2023-04-01	20:00:00	22:00:00
	5	5	5	2023-05-01	13:00:00	2023-05-01	13:00:00	15:00:00
	6	6	6	2023-06-01	19:00:00	2023-06-01	19:00:00	21:00:00
	7	7	7	2023-07-01	15:00:00	2023-07-01	15:00:00	17:00:00
	8	8	8	2023-08-01	21:00:00	2023-08-01	21:00:00	23:00:00
	9	9	9	2023-09-01	11:00:00	2023-09-01	11:00:00	13:00:00
	10	10	10	2023-10-01	17:00:00	2023-10-01	17:00:00	19:00:00
	11	11	11	2023-11-01	16:00:00	2023-11-01	16:00:00	18:00:00
	12	12	12	2023-12-01	22:00:00	2023-12-01	22:00:00	24:00:00
	13	13	13	2024-01-01	12:00:00	2024-01-01	12:00:00	14:00:00
	14	14	14	2024-02-01	18:00:00	2024-02-01	18:00:00	20:00:00
	15	15	15	2024-03-01	14:00:00	2024-03-01	14:00:00	16:00:00
	16	16	16	2024-04-01	20:00:00	2024-04-01	20:00:00	22:00:00
	17	17	17	2024-05-01	13:00:00	2024-05-01	13:00:00	15:00:00
	18	18	18	2024-06-01	19:00:00	2024-06-01	19:00:00	21:00:00
	19	19	19	2024-07-01	15:00:00	2024-07-01	15:00:00	17:00:00
	20	20	20	2024-08-01	21:00:00	2024-08-01	21:00:00	23:00:00
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Bảng bookinginfo

	ID	name	address
▶	19	Amelia Brown	404 Grape Lane, CityS
	14	Ava Hernandez	456 Banana Road, CityN
	9	Christopher Martin	606 Cherry Lane, CityI
	7	Daniel Lee	404 Elm Drive, CityG
	12	Ella Garcia	909 Hemlock Street, CityL
	4	Emily Davis	101 Maple Lane, CityD
	6	Emma Brown	303 Birch Boulevard, CityF
	16	Isabella Johnson	101 Strawberry Drive, CityP
	13	James Rodriguez	123 Pineapple Avenue, CityM
	2	Jane Smith	456 Oak Avenue, CityB
	1	John Doe	123 Main Street, CityA
	15	Liam Martinez	789 Orange Lane, CityO
	20	Logan Lee	505 Mango Court, CityT
	17	Mia Davis	202 Raspberry Boulevard, Ci...
	3	Mike Johnson	789 Pine Road, CityC
	18	Noah Wilson	303 Blueberry Circle, CityR
	10	Olivia Anderson	707 Sycamore Court, CityJ
	5	Robert Wilson	202 Cedar Street, CityE
	8	Sophia Taylor	505 Walnut Circle, CityH
	11	William Moore	808 Redwood Place, CityK
•	NULL	NULL	NULL

customers 1 ×

3. Bảng customers

	id	customerID	phoneNumber
▶	1	1	1234567890
	2	2	2345678901
	3	3	3456789012
	4	4	4567890123
	5	5	5678901234
	6	6	6789012345
	7	7	7890123456
	8	8	8901234567
	9	9	9012345678
	10	10	1234567890
	11	11	2345678901
	12	12	3456789012
	13	13	4567890123
	14	14	5678901234
	15	15	6789012345
	16	16	7890123456
	17	17	8901234567
	18	18	9012345678
	19	19	1234567890
	20	20	2345678901
•	NULL	NULL	NULL

customerpn 1 ×

4. Bảng customerpn

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
▶	1	10	300000	240000
	2	10	228000	182400
	3	6	161000	161000
	4	6	92000	92000
	5	4	104000	104000
	6	6	124000	124000
	7	3	84000	84000
	8	5	83000	83000
	9	0	0	0
	10	0	0	0
	11	0	0	0
	12	0	0	0
	13	0	0	0
	14	0	0	0
	15	0	0	0
	16	0	0	0
	17	0	0	0
	18	0	0	0
	19	0	0	0
	20	0	0	0
✱	NULL	NULL	NULL	NULL

donhang 1 ✕

5. Bảng donhang

	MaDonHang	MaKhachHang	Sothe
▶	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
	6	6	6
	7	7	7
	8	8	8
	9	9	9
	10	10	10
	11	11	11
	12	12	12
	13	13	13
	14	14	14
	15	15	15
	16	16	16
	17	17	17
	18	18	18
	19	19	19
	20	20	20
•	NULL	NULL	NULL

donhangcanhan 1 ×

6. Bảng donhangcanhan

	MaDonHang	MaBan
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10
	11	11
	12	12
	13	13
	14	14
	15	15
	16	16
	17	17
	18	18
	19	19
	20	20
✱	NULL	NULL

dondattruoc 1 ✕

7. Bảng dondattruoc

	MaHD	tongGia
▶	1	240000
	2	182400
	3	161000
	4	92000
	5	104000
	6	124000
	7	84000
	8	83000
	9	0
	10	0
	11	0
	12	0
	13	0
	14	0
	15	0
	16	0
	17	0
	18	0
	19	0
	20	0
✱	NULL	NULL

hoadon 1 ✕

8. Bảng hoadon

	Maloaimon	Mota	Ten	Giobatdau	Gioketthuc	Loaimon
▶	1	NULL	Pudding	07:00:00	22:00:00	Do an
	2	NULL	Tart	07:00:00	22:00:00	Do an
	3	NULL	Tart trứng	07:00:00	22:00:00	Do an
	4	NULL	Tiramisu classic	07:00:00	22:00:00	Do an
	5	NULL	Tiramisu matcha	07:00:00	22:00:00	Do an
	6	NULL	Trà sữa TÇĐĐ	07:00:00	22:00:00	Do uong
	7	NULL	Trà sữa matcha	07:00:00	22:00:00	Do uong
	8	NULL	Hồng trà	07:00:00	22:00:00	Do uong
	9	NULL	Bạc xỉu	07:00:00	22:00:00	Do uong
	10	NULL	Cà phê phin đá	07:00:00	22:00:00	Do uong
	11	NULL	Bánh mì pate	07:00:00	22:00:00	Do an
	12	NULL	Bún riêu	07:00:00	22:00:00	Do an
	13	NULL	Gỏi cuốn	07:00:00	22:00:00	Do an
	14	NULL	Nước mía	07:00:00	22:00:00	Do uong
	15	NULL	Nước lọc	07:00:00	22:00:00	Do uong
	16	NULL	Mì xào	07:00:00	22:00:00	Do an
	17	NULL	Cơm rang	07:00:00	22:00:00	Do an
	18	NULL	Trà đào	07:00:00	22:00:00	Do uong
	19	NULL	Nước chanh	07:00:00	22:00:00	Do uong
	20	NULL	Bún chả cá	07:00:00	22:00:00	Do an
	21	NULL	Bún bò Huế	07:00:00	22:00:00	Do an
	22	NULL	Trà sen	07:00:00	22:00:00	Do uong
	23	NULL	Sinh tố bơ	07:00:00	22:00:00	Do uong
	24	NULL	Bánh flan	07:00:00	22:00:00	Do an
	25	NULL	Bánh tráng trộn	07:00:00	22:00:00	Do an
	NULL	NULL	NULL	NULL	NULL	NULL

loaimon 1 ×

9. Bảng loaimon

	Maloaimon	Kichco	Dongia	current_status
▶	1	M	30000	1
	2	M	30000	1
	3	M	30000	1
	4	M	30000	1
	5	M	25000	1
	6	M	30000	1
	7	M	28000	1
	8	M	15000	1
	9	M	12000	1
	10	M	35000	1
	11	M	32000	1
	12	M	18000	1
	13	M	10000	1
	14	M	27000	1
	15	M	23000	1
	16	M	20000	1
	17	M	22000	1
	18	M	28000	1
	19	M	19000	1
	20	M	16000	1
•	NULL	NULL	NULL	NULL

mon 1 ×

10. Bảng mon

	Mamon	Madonhang	Kichco	Giatheongay	Soluong
▶	1	1	M	30000	2
	2	1	M	30000	3
	3	1	M	30000	1
	4	1	M	30000	4
	5	2	M	25000	2
	6	2	M	30000	3
	7	2	M	28000	1
	8	2	M	15000	4
	9	3	M	12000	2
	10	3	M	35000	3
	11	3	M	32000	1
	12	4	M	18000	4
	13	4	M	10000	2
	14	5	M	27000	3
	15	5	M	23000	1
	16	6	M	20000	4
	17	6	M	22000	2
	18	7	M	28000	3
	19	8	M	19000	1
	20	8	M	16000	4
●	NULL	NULL	NULL	NULL	NULL

thuocvemon 1 ×

11. Bảng thuocvemon

	id	discount_percent	note	max_discount	number
▶	1	20	NULL	100000	0
	2	10	NULL	100000	0
	3	5	NULL	100000	0
	4	30	NULL	100000	0
	5	20	NULL	100000	0
	6	10	NULL	100000	0
	7	10	Discount for Category A	10000	0
	8	15	Special Discount	20000	0
	9	20	Holiday Promotion	30000	0
	10	25	New Year Sale	40000	0
	11	30	Customer Loyalty	50000	0
	12	12	Weekend Special	15000	0
	13	18	Limited Time Offer	25000	0
	14	22	Seasonal Discount	35000	0
	15	28	Mid-Year Clearance	45000	0
	16	32	Flash Sale	55000	0
	17	10	Category A	10000	0
	18	15	Special	20000	0
	19	20	Holiday	30000	0
	20	25	New Year	40000	0
	21	30	Loyalty	50000	0
	22	12	Weekend	15000	0
	23	18	Limited Time	25000	0
	24	22	Seasonal	35000	0
	25	28	Mid-Year	45000	0

voucher_type 1 ×

12. Bảng voucher_type

	id	STT	current_status
▶	1	1	0
	1	2	0
	2	1	0
	2	2	0
	3	1	0
	3	2	0
	4	1	0
	4	2	0
	5	1	0
	5	2	0
	6	1	0
	6	2	0
	7	1	0
	7	2	0
	8	1	0
	8	2	0
	9	1	0
	9	2	0
	10	1	0
	10	2	0
•	NULL	NULL	NULL

voucher_card 1 ×

13. Bảng voucher_card

	id	start_time	end_time
▶	1	2023-09-30	2023-12-30
	2	2023-01-01	2023-12-30
	3	2023-09-30	2023-12-30
	4	2023-01-01	2023-12-30
	5	2023-09-30	2023-12-30
	6	2023-01-01	2023-12-30
	7	2023-01-01	2023-02-01
	8	2023-03-01	2023-04-01
	9	2023-02-15	2023-03-15
	10	2023-04-15	2023-05-15
	11	2023-03-01	2023-04-01
	12	2023-05-01	2023-06-01
	13	2023-04-15	2023-05-15
	14	2023-06-15	2023-07-15
	15	2023-05-01	2023-06-01
	16	2023-07-01	2023-08-01
	17	2023-06-15	2023-07-15
	18	2023-08-15	2023-09-15
	19	2023-07-01	2023-08-01
	20	2023-09-01	2023-10-01
	21	2023-08-15	2023-09-15
	22	2023-10-15	2023-11-15
	23	2023-09-01	2023-10-01
	24	2023-11-01	2023-12-01
	25	2023-10-15	2023-11-15

14. Bảng voucher_effective_time

	Maloaimon	Mavoucher	Kichco	minNum
▶	1	1	M	1
	1	2	M	2
	2	2	M	1
	3	2	M	1
	3	3	M	1
	4	1	M	1
	4	3	M	3
	4	4	M	2
	5	1	M	3
	5	4	M	1
	5	10	M	2
	6	1	M	1
	6	4	M	2
	8	3	M	2
	10	1	M	3
	16	14	M	3
	17	2	M	3
	20	2	M	3
	20	3	M	2
•	NULL	NULL	NULL	NULL

ukienapdung 1 ×

15. Bảng dieukienapdung

	order_id	STT_voucher	id_voucher
▶	1	1	1
	2	2	1
	3	1	2
	4	2	2
	5	1	3
	6	2	3
	7	1	4
	8	2	4
	9	1	5
	10	2	5
	11	1	6
	12	2	6
	13	1	7
	14	2	7
	15	1	8
	16	2	8
	17	1	9
	18	2	9
	19	1	10
	20	2	10
✱	NULL	NULL	NULL

payment_apply 1 ×

16. Bảng payment_apply

	MaNV	HoVaTen	email	mk	DiaChi	NgaySinh	SDT	MaGiamSat	Luong	statusNV
▶	1	Nguyen Minh Diem	minhdiem@gmail.com	123456	Tp.A	1990-01-19	0909110000	NULL	50000	1
	2	Vo Van Kha	vankha@gmail.com	123456	Tp.A	2000-01-01	0909120000	1	20000	1
	3	Nguyen Le Phuc	lephuc@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1
	4	Le Nguyen Chuong	chuongle@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1
	5	Ho Huy Hoang	huyhoang@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1
	6	Nguyen Van A	vana@gmail.com	password123	Hanoi	1995-05-10	0912345678	NULL	45000	1
	7	Tran Thi B	thib@gmail.com	secret456	Ho Chi Minh City	1990-03-15	0987654321	2	30000	1
	8	Le Van C	levc@gmail.com	mypassword	Da Nang	1988-12-20	0901122334	1	40000	1
	9	Pham Thi D	phamd@gmail.com	securepass	Hue	1993-07-01	0978123456	3	35000	1
	10	Hoang Van E	hoange@gmail.com	topsecret	Can Tho	1998-11-25	0918765432	2	28000	1
	11	Tran Van F	tranf@gmail.com	myp@ss	Hai Phong	1992-09-03	0965432109	1	38000	1
	12	Nguyen Thi G	nguyeng@gmail.com	password123	Quang Ninh	1996-02-28	0932111222	3	32000	1
	13	Le Van H	leh@gmail.com	secretp@ss	Vinh	1985-06-07	0945678901	NULL	50000	1
	14	Pham Van I	phami@gmail.com	myp@ssword	Dak Lak	1987-04-12	0923456789	2	42000	1
	15	Ho Thi K	hok@gmail.com	mysecret	Nha Trang	1991-10-15	0909876543	1	37000	1
	16	Vo Van L	vol@gmail.com	secure123	Buon Ma Thuot	1997-08-20	0956789012	3	31000	1
	17	Le Van M	lem@gmail.com	p@ssword	Can Tho	1994-01-05	0978123456	2	34000	1
	18	Nguyen Van N	nguyenn@gmail.com	password123	Ha Long	1989-12-30	0912345678	NULL	46000	1
	19	Tran Van P	tranp@gmail.com	myp@ss	Hai Phong	1990-05-25	0934567890	1	39000	1
	20	Pham Thi Q	phamq@gmail.com	secretp@ss	Da Nang	1993-09-18	0967890123	3	33000	1
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

nhanvien 1 ✕

17. Bảng nhanvien

	MaNV	MaHD	MaDH	NgàyXuất	GioXuất
▶	1	1	1	2023-12-01	08:30:00
	1	20	20	2023-12-10	19:15:00
	2	2	2	2023-12-01	09:45:00
	3	3	3	2023-12-02	10:15:00
	4	4	4	2023-12-02	12:30:00
	5	5	5	2023-12-03	13:45:00
	6	6	6	2023-12-03	15:00:00
	7	7	7	2023-12-04	16:30:00
	8	8	8	2023-12-04	18:45:00
	9	9	9	2023-12-05	19:15:00
	10	10	10	2023-12-05	21:30:00
	11	11	11	2023-12-06	22:45:00
	12	12	12	2023-12-06	08:30:00
	13	13	13	2023-12-07	09:45:00
	14	14	14	2023-12-07	11:15:00
	15	15	15	2023-12-08	12:30:00
	16	16	16	2023-12-08	13:45:00
	17	17	17	2023-12-09	15:00:00
	18	18	18	2023-12-09	16:30:00
	19	19	19	2023-12-10	18:45:00
✱	NULL	NULL	NULL	NULL	NULL

xuat 1 ✕

18. Bảng xuất

2. Trigger – Thủ tục - Hàm

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

Bảng dữ liệu: **NHANVIEN**

a. Thủ tục INSERT:

Mô tả thủ tục: Thủ tục dùng để thêm nhân viên, có validate các thuộc tính.

Input: HoVaTenParam VARCHAR(50), EmailParam VARCHAR(100), MatKhauParam VARCHAR(100), DiaChiParam VARCHAR(50), NgaySinhParam DATE, SDTParam VARCHAR(10), MaGiamSatParam INT, LuongParam INT

Output: <kết quả trả về, nếu có>

Câu lệnh tạo thủ tục:

```
DELIMITER //
CREATE PROCEDURE ThemNhanVien(
    IN HoVaTenParam VARCHAR(50),
    IN EmailParam VARCHAR(100),
    IN MatKhauParam VARCHAR(100),
    IN DiaChiParam VARCHAR(50),
    IN NgaySinhParam DATE,
    IN SDTParam VARCHAR(10),
    IN MaGiamSatParam INT,
    IN LuongParam INT
)
BEGIN
    DECLARE age INT;

    SET age = YEAR(NOW()) - YEAR(NgaySinhParam);

    IF age <= 16 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tuổi phải lớn hơn 16.';
    END IF;

    IF LuongParam < 20000 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Lương phải lớn hơn hoặc bằng 20000.';
    END IF;

    INSERT INTO NHANVIEN
    SET HoVaTen = HoVaTenParam,
        email = EmailParam,
        mk = MatKhauParam,
        DiaChi = DiaChiParam,
        NgaySinh = NgaySinhParam,
```

```

        SDT = SDTParam,
        MaGiamSat = MaGiamSatParam,
        Luong = LuongParam;
END //
DELIMITER ;

```

Nhóm câu lệnh thực thi thủ tục mẫu:

```

call ThemNhanVien('Nguyen Van Khai','vankhai@gmail.com','123456','tpHCM','2013-09-12','0909120000','2','10000');

call ThemNhanVien('Nguyen Van Khai','vankhai@gmail.com','123456','tpHCM','2003-09-12','0909120000','2','10000');

call ThemNhanVien('Nguyen Van Khai','vankhai@gmail.com','123456','tpHCM','2003-09-12','0909120000','2','30000');

call ThemNhanVien('Nguyen Van Khai','vankhai@gmail.com','123456','tpHCM','2003-09-12','0909120000','2','30000');

```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện thủ tục

#	Time	Action	Message	Duration / Fetch
❌	1 22:43:29	call ThemNhanVien('Nguyen Van Khai','vankhai...	Error Code: 1644. Tuổi phải lớn hơn 16.	0.000 sec
❌	2 22:43:32	call ThemNhanVien('Nguyen Van Khai','vankhai...	Error Code: 1644. Lương phải lớn hơn hoặc bằng...	0.000 sec
❌	3 22:43:35	call ThemNhanVien('Nguyen Van Khai','vankhai...	Error Code: 1406. Data too long for column 'SDT...	0.000 sec
✅	4 22:43:38	call ThemNhanVien('Nguyen Van Khai','vankhai...	1 row(s) affected	0.000 sec

Sau khi thực hiện lần lượt các câu lệnh ở trên

MaNV	HoVaTen	email	mk	DiaChi	NgaySinh	SDT	MaGiamSat	Luong	statusNV
16	Vo Van L	vol@gmail.com	secure123	Bun Ma Thuot	1997-08-20	0956789012	3	31000	1
17	Le Van M	lem@gmail.com	p@ssword	Can Tho	1994-01-05	0978123456	2	34000	1
18	Nguyen Van N	nguyenn@gmail.com	password123	Ha Long	1989-12-30	0912345678	NULL	46000	1
19	Tran Van P	tranp@gmail.com	myp@ss	Hai Phong	1990-05-25	0934567890	1	39000	1
20	Pham Thi Q	phamq@gmail.com	secretp@ss	Da Nang	1993-09-18	0967890123	3	33000	1
22	Nguyen Van Khai	vankhai@gmail.com	123456	tpHCM	2003-09-12	0909120000	2	30000	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Bảng nhân viên sau khi thực hiện thủ tục thành công

b. Thủ tục DELETE:

Mô tả thủ tục: Thủ tục xóa nhân viên bằng cách chỉnh thuộc tính statusNV về 0.

Input: MaNVParam INT

Output: <kết quả trả về, nếu có>

Câu lệnh tạo thủ tục:

```
DELIMITER //
CREATE PROCEDURE XoaNhanVien(
    IN MaNVParam INT
)
BEGIN
    UPDATE NHANVIEN SET statusNV = 0 WHERE MaNV = MaNVParam;
END //
DELIMITER ;
```

Nhóm câu lệnh thực thi thủ tục mẫu:

```
call XoaNhanVien(21);
```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện thủ tục

	MaNV	HoVaTen	email	mk	DiaChi	NgaySinh	SDT	MaGiamSat	Luong	statusNV
	16	Vo Van L	vol@gmail.com	secure123	Buon Ma Thuot	1997-08-20	0956789012	3	31000	1
	17	Le Van M	lem@gmail.com	p@ssword	Can Tho	1994-01-05	0978123456	2	34000	1
	18	Nguyen Van N	nguyenn@gmail.com	password123	Ha Long	1989-12-30	0912345678	NULL	46000	1
	19	Tran Van P	tranp@gmail.com	myp@ss	Hai Phong	1990-05-25	0934567890	1	39000	1
	20	Pham Thi Q	phamq@gmail.com	secretp@ss	Da Nang	1993-09-18	0967890123	3	33000	1
	21	Nguyen Van Khai	vankhai@gmail.com	123456	tpHCM	2003-09-12	0909120000	2	30000	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

nhanvien 4 x

Bảng nhân viên sau khi thực hiện thủ tục thành công

c. Thủ tục UPDATE:

Mô tả thủ tục: Chia làm hai thủ tục:

- Thủ tục dùng bởi admin, giúp admin sửa lương và mã giám sát của nhân viên.
- Thủ tục dùng bởi nhân viên, giúp nhân viên tự sửa thông tin cá nhân của bản thân (ngoại trừ lương và mã giám sát).

Input: MaNVParam INT,LuongMoi INT,MaGiamSatMoi INT

Output: <kết quả trả về, nếu có>

Câu lệnh tạo thủ tục:

```
-- thủ tục giúp admin sửa lương và mã giám sát của nhân viên
DELIMITER //
CREATE PROCEDURE SuaLuongVaMaGiamSatNhanVien(
    IN MaNVParam INT,
    IN LuongMoi INT,
    IN MaGiamSatMoi INT
)
BEGIN
    IF MaGiamSatMoi = MaNVParam THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Mã giám sát
mới không được trùng với mã nhân viên.';
```

```

END IF;

IF LuongMoi < 20000 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Luong phải lớn
hơn hoặc bằng 20000.';
END IF;

IF MaGiamSatMoi IS NOT NULL AND NOT EXISTS (SELECT 1 FROM
NHANVIEN WHERE MaNV = MaGiamSatMoi AND statusNV = 1) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Mã giám sát
không tồn tại hoặc đã nghỉ việc.';
END IF;

IF (SELECT statusNV FROM NHANVIEN WHERE MaNV = MaNVParam) = 1
THEN
    UPDATE NHANVIEN
    SET Luong = LuongMoi,
        MaGiamSat = MaGiamSatMoi
    WHERE MaNV = MaNVParam;
ELSE
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Nhân viên
không tồn tại hoặc đã nghỉ việc.';
END IF;
END //
DELIMITER ;

-- thủ tục giúp nhân viên tự sửa thông tin cá nhân của bản thân
(ngoại trừ lương và mã giám sát)
DELIMITER //
CREATE PROCEDURE SuaThongTinNhanVien(
    IN MaNVParam INT,
    IN HoVaTenParam VARCHAR(50),
    IN EmailParam VARCHAR(100),
    IN MatKhauParam VARCHAR(100),
    IN DiaChiParam VARCHAR(50),
    IN NgaySinhParam DATE,
    IN SDTParam VARCHAR(10)
)
BEGIN
    DECLARE age INT;

    SET age = YEAR(NOW()) - YEAR(NgaySinhParam);

    IF age <= 16 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tuổi phải lớn
hơn 16.';
    END IF;

```



```

        IF (SELECT statusNV FROM NHANVIEN WHERE MaNV = MaNVParam) = 1
    THEN
        UPDATE NHANVIEN
        SET HoVaTen = HoVaTenParam,
            email = EmailParam,
            mk = MatKhauParam,
            DiaChi = DiaChiParam,
            NgaySinh = NgaySinhParam,
            SDT = SDTParam
        WHERE MaNV = MaNVParam;
    ELSE
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Nhân viên
không tồn tại hoặc đã nghỉ việc.';
    END IF;
END //
DELIMITER ;

```

Nhóm câu lệnh thực thi thủ tục mẫu:

```

-- procedure SuaLuongVaMaGiamSatNhanVien
call SuaLuongVaMaGiamSatNhanVien(3,20000,3);
call SuaLuongVaMaGiamSatNhanVien(3,10000,2);
call SuaLuongVaMaGiamSatNhanVien(3,20000,21);
call SuaLuongVaMaGiamSatNhanVien(21,20000,1);
call SuaLuongVaMaGiamSatNhanVien(3,30000,2);

-- procedure SuaThongTinNhanVien
call SuaThongTinNhanVien(21, 'Nguyen Van
Khai','vankhai@gmail.com','123456','tpHCM','2013-09-
12','0909120000');
call SuaThongTinNhanVien(21, 'Nguyen Van
Khai','vankhai@gmail.com','123456','tpHCM','2003-09-
12','0909120000');
call SuaThongTinNhanVien(13, 'Le Van H', 'vankha@gmail.com',
'123456', 'Vinh', '1985-06-07', '0945678901'); -- sửa trùng email
call SuaThongTinNhanVien(2, 'Vo Van
Kha','vankha@gmail.com','123456', 'Tp.HCM', '2003-01-01',
'0909123456'); -- sửa địa chỉ, năm sinh, sdt

```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện thủ tục

✓	5	03:05:38	call XoaNhanVien(21)	1 row(s) affected	0.000 sec
✗	6	03:05:39	call SuaLuongVaMaGiamSatNhanVien(3,20000,3)	Error Code: 1644. Mã giám sát mỗi không được trùng với mã nhân viên.	0.000 sec
✗	7	03:05:40	call SuaLuongVaMaGiamSatNhanVien(3,10000,2)	Error Code: 1644. Lương phải lớn hơn hoặc bằng 20000.	0.000 sec
✗	8	03:05:40	call SuaLuongVaMaGiamSatNhanVien(3,20000,21)	Error Code: 1644. Mã giám sát không tồn tại hoặc đã nghỉ việc.	0.000 sec
✗	9	03:05:41	call SuaLuongVaMaGiamSatNhanVien(21,20000,1)	Error Code: 1644. Nhân viên không tồn tại hoặc đã nghỉ việc.	0.000 sec
✓	10	03:05:41	call SuaLuongVaMaGiamSatNhanVien(3,30000,2)	1 row(s) affected	0.000 sec

Sau khi thực hiện lần lượt các câu lệnh cho thủ tục SuaLuongVaMaGiamSatNhanVien ở trên

	MaNV	HoVaTen	email	mk	DiaChi	NgaySinh	SDT	MaGiamSat	Luong	statusNV
▶	1	Nguyen Minh Diem	minhdiem@gmail.com	123456	Tp.A	1990-01-19	0909110000	HULL	50000	1
	2	Vo Van Kha	vankha@gmail.com	123456	Tp.A	2000-01-01	0909120000	1	20000	1
	3	Nguyen Le Phuc	lephuc@gmail.com	123456	Tp.A	2000-02-01	0909130000	2	30000	1
	4	Le Nguyen Chuong	chuongle@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1
	5	Ho Huy Hoang	huyhoang@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1

Bảng nhân viên sau khi thực hiện thủ tục SuaLuongVaMaGiamSatNhanVien thành công

✖	11	03:05:42	call SuaThongTinNhanVien(21, 'Nguyen Van Khai','vankha@gmail.com','123456','Tp.HCM','2013-09-12','090912... Error Code: 1644. Tuổi phải lớn hơn 16.	0.000 sec
✖	12	03:05:43	call SuaThongTinNhanVien(21, 'Nguyen Van Khai','vankha@gmail.com','123456','Tp.HCM','2003-09-12','090912... Error Code: 1644. Nhân viên không tồn tại hoặc đã nghỉ việc.	0.000 sec
✖	13	03:05:44	call SuaThongTinNhanVien(13, 'Le Van H','vankha@gmail.com','123456','Vinh','1985-06-07','0945678901') Error Code: 1062. Duplicate entry 'vankha@gmail.com' for key 'nhanvien_email'	0.000 sec
✔	14	03:05:45	call SuaThongTinNhanVien(2, 'Vo Van Kha','vankha@gmail.com','123456','Tp.HCM','2003-01-01','090912345... 1 row(s) affected	0.000 sec

Sau khi thực hiện lần lượt các câu lệnh cho thủ tục SuaThongTinNhanVien ở trên

	MaNV	HoVaTen	email	mk	DiaChi	NgaySinh	SDT	MaGiamSat	Luong	statusNV
▶	1	Nguyen Minh Diem	minhdiem@gmail.com	123456	Tp.A	1990-01-19	0909110000	HULL	50000	1
	2	Vo Van Kha	vankha@gmail.com	123456	Tp.HCM	2003-01-01	0909123456	1	20000	1
	3	Nguyen Le Phuc	lephuc@gmail.com	123456	Tp.A	2000-02-01	0909130000	2	30000	1
	4	Le Nguyen Chuong	chuongle@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1
	5	Ho Huy Hoang	huyhoang@gmail.com	123456	Tp.A	2000-02-01	0909130000	1	20000	1

Bảng nhân viên sau khi thực hiện thủ tục SuaThongTinNhanVien thành công

2.2 TRIGGER

a. Trigger 1 (hoặc Nhóm các Trigger 1)

Mô tả trigger: Trigger này dùng để cập nhật thuộc tính số lượng trong bảng voucher_type.

Sử dụng trong trường hợp một loại voucher có thêm thẻ voucher hoặc bị xóa bớt thẻ. Được kích hoạt khi có lệnh insert hay delete tới bảng voucher_card.

Đối với trigger add_voucher_num sẽ thực hiện việc tăng số lượng của loại voucher mới vừa có thêm thẻ mới trong bảng voucher_card. Ngược lại, sẽ giảm số lượng của loại voucher đi một đối với trigger delete_voucher_num.

Bởi vì trong nghiệp vụ chỉ có thêm hoặc xóa vào bảng voucher_card. Bảng này chỉ được tạo ra để quản lý chính xác số lượng thẻ voucher cho mỗi loại voucher nên chỉ có hai thao tác chính là thêm và xóa.

Câu lệnh tạo (các) trigger:

```

DROP TRIGGER IF EXISTS add_voucher_num;

DELIMITER //

create trigger add_voucher_num
after insert on VOUCHER_CARD
for each row
begin

```

```

        update VOUCHER_TYPE
        SET number=number+1
        WHERE VOUCHER_TYPE.id = new.id;
end //
DELIMITER ;

DROP TRIGGER IF EXISTS delete_voucher_num;
delimiter //
create trigger delete_voucher_num
before delete on VOUCHER_CARD
for each row
begin
    update VOUCHER_TYPE
    SET number=number-1
    where VOUCHER_TYPE.id = old.id;
end //
delimiter ;

```

Nhóm câu lệnh kiểm tra trigger: (minh họa đủ các trường hợp nêu ra trong trigger)

```

-- trigger thêm, xóa thẻ voucher
select * from voucher_type;
insert into voucher_card(id,STT) values ('11','1'),('11','2');
select * from voucher_type;
delete from voucher_card where id = '11';
select * from voucher_type;

```

Kết quả các màn hình hiển thị từ DBMS minh họa cho trigger.

	id	discount_percent	note	max_discount	number
	11	30	Customer Loyalty	50000	0
	12	15	Weekend Special	15000	0

Bảng voucher_type trước khi insert vào voucher_card

	id	discount_percent	note	max_discount	number
	11	30	Customer Loyalty	50000	2
	12	12	Weekend Special	15000	0

Sau khi insert hai dòng như đoạn code mô tả vào bảng voucher_card

	id	discount_percent	note	max_discount	number
	10	25	New Year Sale	40000	0
	11	30	Customer Loyalty	50000	0

Bảng voucher_type sau khi delete tất cả thẻ voucher có id là 1 trong bảng voucher_card

b. Trigger 2 (hoặc Nhóm các Trigger 2)

Mô tả trigger:

Có hai nhóm trigger chính:

Nhóm 1: cập nhật lại số lượng và tổng giá của đơn hàng

- Mô tả: trigger kích hoạt sau khi có thêm, xóa sửa trong bảng thuocvemon. Trigger sẽ gọi ra thủ tục tinhTongGiaUuDai để tính lại giá sau ưu đãi của đơn hàng (có thay đổi) và thủ tục TongGiaVaTongMon để cập nhật lại số lượng và tổng giá của đơn hàng (có thay đổi)

Nhóm 2: cập nhật lại tổng giá sau ưu đãi của đơn hàng

- Mô tả: trigger kích hoạt sau khi có thêm, xóa, sửa vào bảng payment_apply. Trigger sẽ gọi ra thủ tục tinhTongGiaUuDai để tính lại giá sau ưu đãi của đơn hàng (có thay đổi)

- Thủ tục TongGiaVaTongMon: Thủ tục này sẽ truy vấn vào bảng thuộc về món để tìm tất cả các món trong đơn hàng, tính tổng số lượng món và tổng giá. Sau đó update hai thuộc tính này trong bảng donhang ở dòng có mã đơn hàng tương ứng

+ input: orderId int

- Thủ tục tinhTongGiaUuDai: Thủ tục này dùng để kiểm tra một loại voucher có áp dụng được cho đơn hàng hay không và cập nhật lại tổng giá sau ưu đãi của đơn hàng

+ Cách hiện thực:

- Gán tổng giá gốc của đơn hàng vào biến tổng giá sau ưu đãi (TH voucher không áp dụng được)
- Truy vấn vào các bảng voucher_effective_time, dieukienapdung, thuocvemon để kiểm tra xem voucher này có áp dụng được cho đơn hàng này hay không

- Nếu voucher áp dụng được thì truy vấn bảng voucher_type để lấy phần trăm giảm giá và giảm tối đa của voucher
- Kiểm tra lượng tiền giảm từ voucher có vượt quá giảm tối đa hay không để gán giá trị vào biến tổng giá sau ưu đãi
- Cập nhật lại thuộc tính tương ứng của đơn hàng theo biến tổng giá sau ưu đãi

Câu lệnh tạo (các) trigger:

```
-- tạo trigger cập nhật tổng giá gốc và tổng số lượng trong đơn
hàng khi thêm hàng vào thuocvemon

drop trigger if exists afterIncreaseThuocvemon
delimiter //

create trigger afterIncreaseThuocvemon
after insert on thuocvemon
for each row
begin
    declare voucher_id2 int;
    select id_voucher into voucher_id2
    from payment_apply
    where order_id = new.madonhang;
    if ( voucher_id2 is not null) then call tinhTongGiaUuDai
(voucher_id2, new.madonhang);
    end if;
    call TongGiaVaTongMon(new.madonhang);
end//
delimiter ;

-- tạo trigger cập nhật tổng giá gốc và tổng số lượng trong đơn
hàng khi xóa hàng từ thuocvemon

drop trigger if exists afterDeleteThuocvemon;
delimiter //

create trigger afterDeleteThuocvemon
after delete on thuocvemon
for each row
```

```

begin
    declare voucher_id int;
    select id_voucher into voucher_id
    from payment_apply
    where order_id = old.madonhang;
    if ( voucher_id is not null) then call tinhTongGiaUuDai
(voucher_id, old.madonhang);
    end if;
    call TongGiaVaTongMon (old.madonhang);
end//

delimiter ;

-- tạo trigger cập nhật tổng giá gốc và tổng số lượng và tổng
trong đơn hàng khi sửa hàng trong thuocvemon

drop trigger if exists afterUpdateThuocvemon;

delimiter //

create trigger afterUpdateThuocvemon
after update on thuocvemon
for each row
begin
    declare voucher_id1 int;
    declare voucher_id2 int;

    select id_voucher into voucher_id1
    from payment_apply
    where order_id = old.madonhang;
    if ( voucher_id1 is not null) then call tinhTongGiaUuDai
(voucher_id1, old.madonhang);
    end if;

    select id_voucher into voucher_id2
    from payment_apply

```

```

        where order_id = new.madonhang;

        if ( voucher_id2 is not null) then call tinhTongGiaUuDai
(voucher_id2, new.madonhang);

        end if;

        call TongGiaVaTongMon (old.madonhang);
        call TongGiaVaTongMon (new.madonhang);
end//
delimiter ;

-- thủ tục cập nhật thuộc tính tổng giá gốc, số lượng món
drop procedure if exists TongGiaVaTongMon;
delimiter //
create procedure TongGiaVaTongMon(orderId int )
begin
    DECLARE totalCount INT DEFAULT 0;
    DECLARE totalPrice INT DEFAULT 0;
    select sum(soluong) into totalCount
    from thuocvemon
    where madonhang=orderId
    group by madonhang;

    select sum(soluong*giatheongay) into totalPrice
    from thuocvemon
    where madonhang=orderId
    group by madonhang;

    update donhang
    set soluongmon = totalCount, tonggiagoc = totalPrice
    where madonhang = orderId;

end //
delimiter ;

```

```

-- trigger kích hoạt sau khi thêm hàng vào payment apply
-- kết quả: thuộc tính tổng giá sau ưu đãi đã được cập nhật
drop trigger if exists  afterInsertPaymentApply;
delimiter //
create trigger afterInsertPaymentApply
after insert on payment_apply
for each row
begin
    call tinhTongGiaUuDai (new.id_voucher, new.order_id);
end //
delimiter ;

-- trigger kích hoạt sau khi thay đổi hàng trong payment apply
-- kết quả: thuộc tính tổng giá sau ưu đãi đã được cập nhật
drop trigger if exists  afterUdatePaymentApply;
delimiter //
create trigger afterUdatePaymentApply
after update on payment_apply
for each row
begin
    call tinhTongGiaUuDai (null, old.order_id);
    call tinhTongGiaUuDai (new.id_voucher, new.order_id);
end //
delimiter ;

-- trigger kích hoạt sau khi xóa hàng trong payment apply (một đơn
hàng không còn được áp dụng voucher nữa)
-- kết quả: thuộc tính tổng giá sau ưu đãi đã được cập nhật
drop trigger if exists  afterDeletePaymentApply;
delimiter //

```



```

create trigger afterDeletePaymentApply
after delete on payment_apply
for each row
begin
    update donhang
    set tongGiaSauUuDai = tongGiaGoc
    where MaDonHang = old.order_id;
end //
delimiter ;
-- thủ tục update thuộc tính tổng giá sau ưu đãi của đơn hàng sau
-- khi áp dụng voucher
-- input: mã voucher, mã đơn hàng
-- output: thuộc tính tổng giá sau ưu đãi đã được cập nhật
drop procedure if exists tinhTongGiaUuDai;
delimiter //
create procedure tinhTongGiaUuDai (voucher_id int , order_id int)
begin
    declare applyCondition int ;
    declare priceWithVoucher int;
    declare discount_total_percent int default 0;
    declare discount_max int default 0;

    select tonggiagoc into priceWithVoucher
    from donhang
    where donhang.Madonhang=order_id;

    drop temporary table if exists t1;
    create temporary table t1 as
    select *
    from dieukienapdung,voucher_effective_time

```

```
where mavoucher=voucher_id and current_date()>= start_time and  
current_date()<= end_time;
```

```
drop temporary table if exists t2;
```

```
create temporary table t2 as
```

```
select *
```

```
from thuocvemon
```

```
where madonhang=order_id;
```

```
drop temporary table if exists t3;
```

```
create temporary table t3 as
```

```
SELECT t1.Maloaimon,t2.Kichco,t2.soluong,t1.minNum
```

```
FROM t1 JOIN t2 ON t1.Kichco = t2.Kichco AND t1.Maloaimon =  
t2.Mamon;
```

```
select count(*) into applyCondition
```

```
from t3
```

```
where soluong >= minNum;
```

```
if (applyCondition > 0) then
```

```
select discount_percent into discount_total_percent
```

```
from voucher_type
```

```
where id=voucher_id;
```

```
select voucher_type.max_discount into discount_max
```

```
from voucher_type
```

```
where id=voucher_id;
```

```
end if;
```

```
if (priceWithVoucher*discount_total_percent/100 >  
discount_max) then
```

```

        set priceWithVoucher=priceWithVoucher-discount_max;
    else
        set priceWithVoucher=priceWithVoucher-
priceWithVoucher*discount_total_percent/100;
    end if;

    update donhang
    set tonggiasauUuDai = priceWithVoucher
    where madonhang=order_id;
end //
delimiter ;

```

Nhóm câu lệnh kiểm tra trigger: (Các nhóm câu lệnh kiểm tra từng trigger riêng biệt luôn được chạy trên dữ liệu tạo sẵn ban đầu)

```

-- trigger thêm xóa sửa thuocvemon
select * from donhang;

insert into thuocvemon values
('1','9','M','30000','1'),('2','9','M','30000','2'),('3','9','M','30000','3');

select * from donhang;

delete from thuocvemon where madonhang = '9' and mamon='2';

select * from donhang;

update thuocvemon
set madonhang = '8'
where madonhang='9' and mamon='3';

-- trigger thêm, xóa, sửa payment_apply
insert into voucher_card(id,STT) values ('1','5');

-- thêm món vào đơn hàng 8 và 9 để đảm bảo voucher có mã 1 đều có
thẻ áp dụng trên cả hai đơn hàng

insert into thuocvemon values
('1','9','M','30000','1'),('2','9','M','30000','2'),('3','9','M','30000','3');

```

```

insert into thuocvemon values ('1','8','M','30000','2');
-- đảm bảo đơn hàng 8 và 9 đang không áp dụng voucher nào
delete from payment_apply where order_id ='9';
delete from payment_apply where order_id ='8';
select * from donhang;
insert into payment_apply values ('9','5','1');
select * from donhang;
update payment_apply
set order_id = '8'
where id_voucher='1' and STT_voucher = '5';
select * from donhang;
delete from payment_apply where order_id = '8' and id_voucher='1'
and STT_voucher = '5';
select * from donhang;

```

Kết quả các màn hình hiển thị từ DBMS minh họa cho trigger thêm, xóa, sửa bảng thuocvemon.

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	7	3	84000	84000
	8	5	83000	83000
	9	0	0	0

Bảng donhang ban đầu

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	5	83000	83000
	9	6	180000	180000

Bảng donhang sau khi insert vào bảng thuocvemon

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	5	83000	83000
	9	4	120000	120000

Bảng đơn hàng sau khi delete trong bảng thuocvemon

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	8	173000	173000
	9	1	30000	30000

Bảng donhang sau khi update trong bảng thuocvemon

Kết quả các màn hình hiển thị từ DBMS minh họa cho trigger thêm, xóa, sửa bảng payment_apply.

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	7	143000	143000
	9	6	180000	180000

Bảng donhang trước khi test trigger

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	7	143000	143000
	9	6	180000	144000

Bảng donhang sau khi insert vào payment_apply

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	7	143000	114400
	9	6	180000	180000

Bảng donhang sau khi update trong payment_apply

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
	8	7	143000	143000
	9	6	180000	180000

Bảng donhang sau khi delete trong payment_apply

2.3 THỦ TỤC

a. Thủ tục 1

Mô tả thủ tục: Thủ tục này dùng để in ra thông tin chi tiết của một đơn hàng, bao gồm: Mã món, tên món, số lượng, tổng giá.

Input: order_id INT

Output: Một bảng dữ liệu thể hiện thông tin chi tiết của một đơn hàng cụ thể.

Câu lệnh tạo thủ tục:

```
drop procedure if exists showOrderInfo;
delimiter //
create procedure showOrderInfo( order_id INT)
begin
    select Mamon, Ten as
    TenMon, Kichco, Soluong, giatheongay, (Soluong*giatheongay) as
    tong_gia
    from thuocvemon, loaimon
    where Madonhang=order_id and mamon=Maloaimon;
end //
delimiter ;
```

Nhóm câu lệnh thực thi thủ tục mẫu:

```
call showOrderInfo(1);
call showOrderInfo(10);
```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện thủ tục

	Mamon	TenMon	Kichco	Soluong	giatheongay	tong_gia
►	1	Pudding	M	2	30000	60000
	2	Tart	M	3	30000	90000
	3	Tart trứng	M	1	30000	30000
	4	Tiramisu classic	M	4	30000	120000

Kết quả sau khi thực hiện thủ tục showOrderInfo(1) thành công

	Mamon	TenMon	Kichco	Soluong	giatheongay	tong_gia

Kết quả sau khi thực hiện thủ tục showOrderInfo(10) thành công

b. Thủ tục 2

Mô tả thủ tục: Thủ tục in ra thông tin của nhân viên và số hóa đơn mà nhân viên đó đã xuất với điều kiện nhân viên đó phải có số tiền lương không vượt quá số tiền đã nhập.

Input: LuongToiDa INT

Output: Bảng gồm mã nhân viên, họ và tên, Lương, Số lượng hóa đơn.

Câu lệnh tạo thủ tục:

```
DELIMITER //
CREATE PROCEDURE InThongTinNhanVienVaHoaDon(
    IN LuongToiDa INT
)
BEGIN
    SELECT
        NV.MaNV,
        NV.HoVaTen,
        NV.Luong,
        COUNT(X.MaDH) AS SoLuongHoaDon
    FROM
        NHANVIEN NV
    LEFT JOIN
        XUAT X ON NV.MaNV = X.MaNV
    WHERE
        NV.Luong <= LuongToiDa AND NV.statusNV = 1
    GROUP BY
        NV.MaNV
    ORDER BY
        NV.MaNV;
END //
DELIMITER ;
```

Nhóm câu lệnh thực thi thủ tục mẫu:

```
call InThongTinNhanVienVaHoaDon(20000);
call InThongTinNhanVienVaHoaDon(50000);
```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện thủ tục

	id	HoVaTen	Luong	SoLuongHoaDon
▶	2	Vo Van Kha	20000	1
	4	Le Nguyen Chuong	20000	1
	5	Ho Huy Hoang	20000	1

Kết quả sau khi thực hiện thủ tục InThongTinNhanVienVaHoaDon(20000) thành công

	id	HoVaTen	Luong	SoLuongHoaDon
▶	1	Nguyen Minh Diem	50000	2
	2	Vo Van Kha	20000	1
	3	Nguyen Le Phuc	30000	1
	4	Le Nguyen Chuong	20000	1
	5	Ho Huy Hoang	20000	1
	6	Nguyen Van A	45000	1
	7	Tran Thi B	30000	1
	8	Le Van C	40000	1
	9	Pham Thi D	35000	1
	10	Hoang Van E	28000	1
	11	Tran Van F	38000	1
	12	Nguyen Thi G	32000	1
	13	Le Van H	50000	1
	14	Pham Van I	42000	1
	15	Ho Thi K	37000	1
	16	Vo Van L	31000	1
	17	Le Van M	34000	1
	18	Nguyen Van N	46000	1
	19	Tran Van P	39000	1
	20	Pham Thi Q	33000	0

Kết quả sau khi thực hiện thủ tục InThongTinNhanVienvaHoaDon(50000) thành công

2.4 HÀM

a. Hàm 1

Mô tả hàm: Hàm trả về tổng doanh thu của một bàn trong một khoảng thời gian.

Input: MaBanParam INT, NgayBatDauParam DATE, GioBatDauParam TIME, NgayKetThucParam DATE, GioKetThucParam TIME

Output: TongTien INT

Câu lệnh hàm:

```
DELIMITER //
CREATE FUNCTION TongTienTheoBan(
    MaBanParam INT,
    NgayBatDauParam DATE,
    GioBatDauParam TIME,
    NgayKetThucParam DATE,
    GioKetThucParam TIME
)
)
```



```

RETURNS INT
READS SQL DATA
BEGIN
    DECLARE TongTien INT;

    CREATE TEMPORARY TABLE TempTable
    AS
    SELECT
        COALESCE(DH.TongGiaSauUuDai, 0) AS TongGiaSauUuDai
    FROM
        bookingTables BT
    JOIN
        DonDatTruoc DDT ON BT.ID = DDT.MaBan
    JOIN
        bookingInfo BI ON BT.ID = BI.fk_tableID
    JOIN
        DonHang DH ON DDT.MaDonHang = DH.MaDonHang
    WHERE
        BT.ID = MaBanParam
        AND (BI.startDate > NgayBatDauParam
            OR (BI.startDate = NgayBatDauParam
                AND BI.startTime >= GioBatDauParam))
        AND (BI.startDate < NgayKetThucParam
            OR (BI.startDate = NgayKetThucParam
                AND BI.endTime <= GioKetThucParam));

    SELECT SUM(TongGiaSauUuDai) INTO TongTien FROM TempTable;

    DROP TEMPORARY TABLE IF EXISTS TempTable;

    RETURN COALESCE(TongTien, 0);
END //
DELIMITER ;

```

Nhóm câu lệnh thực thi hàm:

```

SELECT TongTienTheoBan(1, '2023-01-01', '12:00:00', '2023-12-31',
'23:59:59');
SELECT TongTienTheoBan(1, '2023-01-01', '12:00:00', '2023-01-01',
'13:00:00');
SELECT TongTienTheoBan(3, '2023-01-01', '00:00:00', '2023-12-31',
'23:59:59');

```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện hàm

	TongTienTheoBan(1, '2023-01-01', '12:00:00', '2023-12-31', '23:59:59')
▶	240000

	TongTienTheoBan(1, '2023-01-01', '12:00:00', '2023-01-01', '13:00:00')
▶	0

	TongTienTheoBan(3, '2023-01-01', '00:00:00', '2023-12-31', '23:59:59')
▶	161000

Các kết quả sau khi lần lượt thực hiện các câu lệnh trên thành công

b. Hàm 2

Mô tả hàm: Hàm này tính toán tổng số doanh thu của một người khách đã đóng góp cho cửa hàng và trả về chuỗi thông báo quy định sẵn cho mỗi mức độ đóng góp tương ứng của khách.

Input: MaKhachHangParam INT

Output: VARCHAR(50) - chuỗi mức độ đóng góp là khách hàng VIP, thường hay mới

Câu lệnh hàm:

```
DELIMITER //
CREATE FUNCTION MucDoDongGopCustomer(
    MaKhachHangParam INT
)
RETURNS VARCHAR(50)
READS SQL DATA
BEGIN
    DECLARE TongTien INT;

    -- Check if customer ID exists
    DECLARE customerExists INT;
    SELECT COUNT(*) INTO customerExists
    FROM customers
    WHERE ID = MaKhachHangParam;

    IF customerExists = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Khách hàng không tồn tại.';
    END IF;
```

```

SELECT
    SUM(DH.TongGiaSauUuDai) INTO TongTien
FROM
    DonHang DH
JOIN
    DonHangCaNhan DHCN ON DH.MaDonHang = DHCN.MaDonHang
WHERE
    DHCN.MaKhachHang = MaKhachHangParam;

IF TongTien >= 3000000 THEN
    RETURN 'VIP_Customer';
ELSEIF TongTien >= 500000 THEN
    RETURN 'normal_customer';
ELSE
    RETURN 'new_customer';
END IF;
END //
DELIMITER ;

```

Nhóm câu lệnh thực thi hàm:

```

-- thêm lệnh sau vào file insert dữ liệu mẫu trước khi thực hiện
-- thủ tục applyVoucherForOrder cho áp dụng thanh toán voucher
INSERT INTO thuocvemom (Mamon, Madonhang, Kichco, Giatheongay,
Soluong) VALUES
    ('10', '1', 'M', '100000', '50'),
    ('16', '2', 'M', '100000', '20');

SELECT MucDoDongGopCustomer(22);
SELECT MucDoDongGopCustomer(1);
SELECT MucDoDongGopCustomer(2);
SELECT MucDoDongGopCustomer(3);

```

Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện hàm

22	03:38:40	SELECT MucDoDongGopCustomer(22) LIMIT 0, 1000	Error Code: 1644. Khách hàng không tồn tại.	0.000 sec
23	03:38:41	SELECT MucDoDongGopCustomer(1) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
24	03:39:31	SELECT MucDoDongGopCustomer(2) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
25	03:40:07	SELECT MucDoDongGopCustomer(3) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Sau khi thực hiện lần lượt các câu lệnh cho hàm MucDoDongGopCustomer ở trên

	MaDonHang	SoLuongMon	TongGiaGoc	TongGiaSauUuDai
▶	1	60	5300000	5200000
	2	30	2228000	2128000
	3	6	161000	161000

Đơn hàng 1, 2, 3 lần lượt của khách hàng 1, 2, 3

	MucDoDongGopCustomer(1)
▶	VIP_Customer

Kết quả sau khi thực hiện câu lệnh SELECT MucDoDongGopCustomer(1) thành công

	MucDoDongGopCustomer(2)
▶	normal_customer

Kết quả sau khi thực hiện câu lệnh SELECT MucDoDongGopCustomer(2) thành công

	MucDoDongGopCustomer(3)
▶	new_customer

Kết quả sau khi thực hiện câu lệnh SELECT MucDoDongGopCustomer(3) thành công

2.5 Các hàm và thủ tục khác hỗ trợ cho website hoạt động

2.5.1 Hàm 3

Mô tả hàm: Hàm giúp tạo đơn hàng cá nhân khi có khách vào, đồng thời xác nhận thẻ voucher mà khách sử dụng (nếu có).

Input: tenKH VARCHAR(100), diachi VARCHAR(200), sothe INT, id_voucher INT, STT_voucher INT

Output: donhangID INT

Câu lệnh hàm:

```
delimiter //
create function insertAndGetID (
    tenKH      varchar(100),
    diachi     varchar(200),
    sothe      int,
    id_voucher int ,
    STT_voucher int
)
returns int
reads sql data
begin
    declare kháchID int;
    declare donhangID int;
    declare existsCustomer int;
    declare check_voucher int;
    declare state int;
    declare start_date date;
    declare end_date date;
```

```

-- kiểm tra khách hàng này đã từng mua ở cửa hàng chưa
drop temporary table if exists s1;
create temporary table s1 as
select * from customers
where name = tenKH and address = diachi;

select count(*) into existsCustomer
from s1;
-- lấy ID của khách vừa đặt hàng
if ( existsCustomer=0) then
    insert into customers (name,address) values (tenKH,
diachi);
    select max(ID) into khachID
    from customers;
else
    select ID into khachID
    from s1;
end if;

-- tạo đơn hàng mới và lấy mã đơn hàng
insert into donhang() values ();
select max(MaDonHang) into donhangID
from donhang;
-- tạo đơn hàng cá nhân từ ID khách, số thẻ, ID đơn hàng
insert into donhangcanhan values (donhangID,khachID,sothe);
-- kiểm tra có áp dụng được voucher hay không
if (id_voucher <> '' and STT_voucher <> '') then
    select count(*) into check_voucher
    from voucher_card
    where voucher_card.id=id_voucher and
STT_voucher=voucher_card.STT;

    -- validate voucher có tồn tại trong hệ thống hay không
    if (check_voucher=1) then
        -- lấy tình trạng hiện tại của thẻ voucher
        select current_status into state
        from voucher_card
        where voucher_card.id=id_voucher and
STT_voucher=voucher_card.STT;
        -- lấy thời gian bắt đầu sử dụng của thẻ voucher
        select start_time into start_date
        from voucher_effective_time
        where id = id_voucher ;
        -- lấy thời gian kết thúc sử dụng của thẻ voucher
        select end_time into end_date
        from voucher_effective_time

```

```

        where id = id_voucher ;
        -- validate thẻ voucher có khả năng sử dụng không
        if (state = 0) then SIGNAL SQLSTATE '45001' SET
MESSAGE_TEXT = 'thẻ voucher đã qua sử dụng';
        elseif (current_date() > end_date) then SIGNAL
SQLSTATE '45001' SET MESSAGE_TEXT = 'thẻ voucher đã hết hạn';
        elseif (current_date() < start_date) then SIGNAL
SQLSTATE '45001' SET MESSAGE_TEXT = 'thẻ voucher chưa khả dụng';
        end if;
        else SIGNAL SQLSTATE '45001' SET MESSAGE_TEXT = 'thẻ
voucher không tồn tại';
        end if;
    end if;
    return donhangID;
end //
delimiter ;

```

2.5.2 Thủ tục 3

Mô tả thủ tục: Thủ tục này dùng để thêm thẻ voucher mới vào cửa hàng, có bao gồm validate. Nếu thẻ voucher đã có trong cửa hàng (khả dụng hoặc không khả dụng) thì báo lỗi. Nếu thẻ voucher chưa có trong cửa hàng thì tiến hành insert vào bảng voucher_card.

Input: voucher_id int , STT_voucher int

Output: Không có kết quả trả về.

Câu lệnh tạo thủ tục:

```

delimiter //
create procedure addVoucher_card (voucher_id int , STT_voucher
int)
begin
    declare checkExist int ;
    declare state int;
    set state =2;
    set checkExist =0;
    drop temporary table if exists t3;
    create temporary table t3 as
    select *
    from voucher_card
    where id = voucher_id and STT = STT_voucher;

    select count(*) into checkExist
    from t3;

    if (checkExist = 1) then
    select current_status into state
    from t3;

```

```

        if (state =0) then SIGNAL SQLSTATE '45001' SET
MESSAGE_TEXT = 'thẻ voucher đã được sử dụng';

        else SIGNAL SQLSTATE '45001' SET MESSAGE_TEXT = 'thẻ
voucher đang có sẵn';
        end if;
    else
        insert into voucher_card (id,STT) values (voucher_id,
STT_voucher);
        end if;
    end //
delimiter ;

```

2.5.2 Thủ tục 4

Mô tả thủ tục: Thủ tục này làm một thẻ voucher trong cửa hàng không khả dụng nữa. Nếu thẻ voucher đã có trong cửa hàng và đang khả dụng thì cập nhật current_status = 0. Ngược lại sẽ báo lỗi.

Input: voucher_id int, STT_voucher int

Output: Không có

Câu lệnh tạo thủ tục:

```

delimiter //
create procedure deleteVoucher_card (voucher_id int, STT_voucher
int)
begin
    declare state int ;
    select current_status into state
    from voucher_card
    where id = voucher_id and STT = STT_voucher;
    if (state =1) then
        update voucher_card
        set current_status =0
        where id = voucher_id and STT = STT_voucher;

        update voucher_type
        set number = number -1
        where id = voucher_id;
    elseif (state =0 ) then SIGNAL SQLSTATE '45001' SET
MESSAGE_TEXT = 'thẻ đang không khả dụng';
        else SIGNAL SQLSTATE '45001' SET MESSAGE_TEXT = 'thẻ
không tồn tại';
        end if;
    end //
delimiter ;

```

2.5.3 Thủ tục 5

Mô tả thủ tục: Thủ tục thêm hoặc cập nhật món mới vào cửa hàng. Nếu món đã tồn tại thì cập nhật lại đơn giá và current_status = 1. Ngược lại sẽ insert dòng mới vào bảng mon.

Input: dish_id int, size char(2), price int

Output: Không có

Câu lệnh tạo thủ tục:

```
delimiter //
create procedure themmon (dish_id int,    size char(2), price int)
begin
    if exists (select maloaimon, Kichco from mon where maloaimon =
dish_id and Kichco = size) THEN
        update mon set current_status = 1, dongia = price where
maloaimon = dish_id and Kichco = size;
    else INSERT INTO mon (maloaimon, kichco, dongia)
        VALUES (dish_id, size, price);
    end if;
end //
delimiter ;
```

2.5.4 Thủ tục 6

Mô tả thủ tục: Thủ tục làm cho món trong cửa hàng không khả dụng bằng cách cập nhật current_status = 0.

Input: dish_id int, size char(1)

Output: Không có

Câu lệnh tạo thủ tục:

```
delimiter //
create procedure xoamon (dish_id int,    size char(1))
begin
    update mon
        set current_status = 0
        where maloaimon=dish_id and kichco = size;
end //
delimiter ;
```

2.5.5 Thủ tục 7

Mô tả thủ tục: Thủ tục hiển thị danh sách các loại món chưa có món (chưa có món hoặc chỉ có các món có current_status = 0).

Input: Không có

Output: Bảng gồm tất cả các thuộc tính có trong loaimon và mỗi loại món trong đó đều rỗng.

Câu lệnh tạo thủ tục:

```
delimiter //
create procedure showLoaiMonRong()
begin
    drop temporary table if exists t4;
    create temporary table t4 as
    select *
        from loaimon
        where (select count(*) from mon where
loaimon.maloaimon=mon.maloaimon) > (select count(*) from mon
where loaimon.maloaimon=mon.maloaimon and mon.current_status = 0);

    select *
    from loaimon
    where loaimon.maloaimon not in (select t4.maloaimon from
t4);
end //
delimiter ;
```

2.5.6 Thủ tục 8

Mô tả: Tìm danh sách tất cả nhân viên có họ và tên chứa từ khóa đã được nhập.

Input: TuKhoa VARCHAR(50)

Output: Danh sách nhân viên thỏa mãn yêu cầu.

Câu lệnh tạo thủ tục:

```
DELIMITER //
CREATE PROCEDURE TimKiemNhanVien(
    IN TuKhoa VARCHAR(50)
)
BEGIN
    SELECT *
    FROM NHANVIEN
    WHERE HoVaTen LIKE CONCAT('%', TuKhoa, '%');
END //
DELIMITER ;
```

2.5.7 Thủ tục 9

Mô tả thủ tục: Thủ tục sắp xếp tất cả các nhân viên theo tên.

Input: Không có

Output: Bảng gồm tất cả các nhân viên đã được sắp xếp theo tên.

Câu lệnh tạo thủ tục:

```

DELIMITER //
CREATE PROCEDURE SapXepNhanVienTheoTen()
BEGIN
    SELECT
        *
    FROM
        NHANVIEN
    WHERE
        statusNV = 1
    ORDER BY
        SUBSTRING_INDEX(HoVaTen, ' ', -1) COLLATE
utf8mb4_unicode_ci;
END //
delimiter ;

```

2.5.8 Thủ tục 10

Mô tả thủ tục: Thủ tục áp dụng thẻ voucher cho một đơn hàng. Insert vào payment_apply để áp dụng voucher vào tính tổng giá sau ưu đãi cho đơn hàng, đồng thời vô hiệu hóa thẻ voucher vừa được áp dụng.

Input: order_id int, id_voucher int, STT_voucher int

Output: Không có.

Câu lệnh tạo thủ tục:

```


delimiter //
create procedure applyVoucherForOrder ( order_id int, id_voucher
int,STT_voucher int)
begin
    insert into payment_apply values (order_id, STT_voucher,
id_voucher);
    call deleteVoucher_card(id_voucher,STT_voucher);
end //
delimiter ;

```

3. ỨNG DỤNG KẾT NỐI VỚI CSDL

Màn hình 1:

- Thêm món: thêm món mới có cùng mã loại món của dòng tương ứng. Nếu món mới có cùng kích thước với món cũ thì chỉ cập nhật lại giá, ngược lại thì thêm dòng mới.
- Xóa món: xóa món ở dòng tương ứng.
- Sửa món: cập nhật lại thông tin của món.



Trang chủ **Quản lý**

Đăng xuất

Món Bàn Đơn hàng Voucher Nhân viên

Đồ uống

Mã loại món	Tên	Kích cỡ	Giá	Khác		
6	Trà sữa TÇĐĐ	M	30000	Thêm món	Sửa món	Xóa món
7	Trà sữa matcha	M	28000	Thêm món	Sửa món	Xóa món
8	Hồng trà	M	15000	Thêm món	Sửa món	Xóa món
9	Bạc xỉu	M	12000	Thêm món	Sửa món	Xóa món
10	Cà phê phin đá	M	35000	Thêm món	Sửa món	Xóa món
14	Nước mía	M	27000	Thêm món	Sửa món	Xóa món
15	Nước lọc	M	23000	Thêm món	Sửa món	Xóa món
18	Trà đào	M	28000	Thêm món	Sửa món	Xóa món

Màn hình 2:

- Điền thông tin vào các trường, ấn vào chi tiết để biết được doanh thu của bàn đã nhập trong khoảng thời gian đã nhập.

Bàn hợp số 15	20	Còn trống
Bàn hợp số 16	30	Còn trống
Bàn hợp số 17	30	Còn trống
Bàn hợp số 18	30	Còn trống
Bàn hợp số 19	30	Còn trống
Bàn hợp số 20	30	Còn trống

Tính tổng doanh thu theo bàn

Mã bàn

Ngày bắt đầu

dd/mm/yyyy

Giờ bắt đầu

--:--

Ngày kết thúc

dd/mm/yyyy


Giờ kết thúc

--:--

Chi tiết

Màn hình 3:

- Thể hiện danh sách thông tin chung của các đơn hàng. Ấn vào chi tiết để xem đơn hàng gồm các món nào.



Trang chủ

Quản lý

Đăng xuất

Món

Bàn

Đơn hàng


Voucher

Nhân viên

STT	Số lượng món	Tổng giá gốc	Tổng giá sau ưu đãi	Khác
1	10	300000	240000	Chi tiết
2	10	228000	182400	Chi tiết
3	6	161000	161000	Chi tiết
4	6	92000	92000	Chi tiết
5	4	104000	104000	Chi tiết
6	6	124000	124000	Chi tiết
7	3	84000	84000	Chi tiết
8	7	143000	143000	Chi tiết
9	6	180000	180000	Chi tiết

Màn hình 4:

- Màn hình hiển thị danh sách loại voucher có trong hệ thống.
- Ấn vào thêm thẻ và nhập số thứ tự cụ thể để tăng số lượng voucher.
- Ấn vào xóa thẻ để giảm số lượng của loại voucher.



Trang chủ

Quản lý

Đăng xuất

Món

Bàn

Đơn hàng

Voucher


Nhân viên

VOUCHER

STT	Phần trăm giảm giá	Ghi chú	Giảm giá tối đa	Số lượng	Thời gian bắt đầu	Thời gian kết thúc	Khác
1	20	Khong co note	100000	1	30/09/2023	30/12/2023	<div>Thêm thẻ</div> <div>Xóa thẻ</div>
2	10	Khong co note	100000	Da het the voucher	01/01/2023	30/12/2023	<div>Thêm thẻ</div> <div>Xóa thẻ</div>
3	5	Khong co note	100000	Da het the voucher	30/09/2023	30/12/2023	<div>Thêm thẻ</div> <div>Xóa thẻ</div>
4	30	Khong co note	100000	Da het the voucher	01/01/2023	30/12/2023	<div>Thêm thẻ</div> <div>Xóa thẻ</div>
5	20	Khong co note	100000	Da het the voucher	30/09/2023	30/12/2023	<div>Thêm thẻ</div> <div>Xóa thẻ</div>
				Da het the			<div></div> <div></div>

Màn hình 5:

- Ấn thêm nhân viên, sau đó nhập thông tin vào các trường để thêm nhân viên mới.
- Ấn sắp xếp theo tên để sắp xếp lại danh sách các nhân viên bên dưới theo tên.
- Ấn tìm theo tên để tìm tất cả nhân viên có họ và tên chứa từ khóa tìm kiếm.
- Nhập lương tối đa cần tìm kiếm và ấn tìm để tìm tất cả nhân viên và hóa đơn đã xuất có tiền lương không vượt quá số tiền đã nhập.
- Ấn reset để danh sách về trạng thái ban đầu.

 Trang chủ Quản lý Đăng xuất

Món Bàn Đơn hàng Voucher Nhân viên

Thêm nhân viên

Sắp xếp theo tên

Tìm theo tên

Tên cần tìm

Nhân viên

STT	Họ và tên	Email	Địa chỉ	SĐT	Mã giám sát	Lương	Khác
1	Nguyen Minh Diem	minhdiem@gmail.com	Tp.A	0909110000	Chưa có	50000	<div>Sửa</div> <div>Xóa</div>
2	Vo Van Kha	vankha@gmail.com	Tp.A	0909120000	1	20000	<div>Sửa</div> <div>Xóa</div>
3	Nguyen Le Phuc	lephuc@gmail.com	Tp.A	0909130000	1	20000	<div>Sửa</div> <div>Xóa</div>

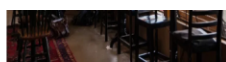
Nhân viên và hóa đơn

NHẬP GIÁ TRỊ LƯƠNG TỐI ĐA CẦN TÌM KIẾM: > 20000

TìmReset

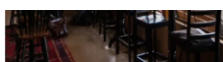
Màn hình 6:

- Sau khi điền thông tin vào các trường, ấn đặt món để tạo đơn hàng cá nhân cho khách, cũng như tạo các món đã chọn trong đơn ứng. Áp dụng voucher cho đơn hàng (nếu có).



CƠM RANG

☐ M



BÚN CHẢ CÁ

☐ M

THÔNG TIN KHÁCH HÀNG

Tên

Địa chỉ

Số thẻ


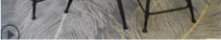
Mã voucher

Số thứ tự voucher

Đặt món

Màn hình 7:

- Nhập các thông tin và ấn đặt bàn để tạo đơn đặt trước, khách hàng, giờ đặt tương ứng.

	
Số lượng : 30	Số lượng : 30
Còn trống <input type="checkbox"/>	Còn trống <input type="checkbox"/>

THÔNG TIN KHÁCH HÀNG

Tên

Địa chỉ

Ngày đặt

Giờ bắt đầu

Giờ kết thúc

Màn hình 8:

- Trong màn hình chi tiết đơn hàng, nhân viên ấn xuất hóa đơn để xuất hóa đơn cho đơn hàng.

Chi tiết đơn hàng

Mã món:

Tên món:



Kích cỡ:

Số lượng:


Giá theo ngày:

Màn hình 9:


- Màn hình chi tiết đơn hàng ở trên có thể đi đến bằng cách ấn vào chi tiết trong trang hóa đơn của nhân viên dưới đây.

Trang chủMónBànHoá đơnĐăng xuất


Giỏ hàng



Tổng giá ban đầu: Tổng giá sau ưu đãi:
300000240000
Số lượng: 10[Chi tiết](#)



Tổng giá ban đầu: Tổng giá sau ưu đãi:
228000182400
Số lượng: 10[Chi tiết](#)



Tổng giá ban đầu: Tổng giá sau ưu đãi:
161000161000

4. Thiết kế ERD cập nhật mới nhất

<Hình ERD>

(Có thể in A3 đính kèm theo giống báo cáo BTL1)