

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL

TÊN ĐỀ TÀI: QUẢN LÝ NHÀ HÀNG

GVHD: *TS. Nguyễn Thành Sơn*

Lớp HP: *DBMS330284_23_1_05*

Nhóm thực hiện: *Nhóm 13*

Học kỳ: *1*

Năm học: *2023 - 2024*

Thành phố Hồ Chí Minh, tháng 11, năm 2023

DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN

HỌC KỲ I NĂM HỌC 2023-2024

Nhóm 13

Đề tài: Quản lý nhà hàng

MSSV	Họ và tên
21133005	Trần Quốc Bảo
21133082	Lê Đức Thắng
21133097	Nguyễn Hồng Ân
20132213	Huỳnh Nguyễn Tuấn Kiệt

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG.....	1
1. Mô tả bài toán:	1
2. Nghiệp vụ bài toán:	2
3. Mô tả chức năng của bài toán:.....	3
4. Tính lương nhân viên:	4
5. Doanh thu nhà hàng:.....	5
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	6
1. Thiết kế cơ sở dữ liệu mức quan niệm:	6
2. Thiết kế cơ sở dữ liệu mức logic:	7
3. Các ràng buộc cần có:.....	8
4. Cài đặt các CSDL và các ràng buộc:	9
5. Các view:	12
6. Các trigger:	13
CHƯƠNG 3: THIẾT KẾ CHỨC NĂNG	16
1. Kết nối cơ sở dữ liệu.....	16
2. Xem thông tin danh mục	16
2.1. Danh mục nhân viên.....	16
2.2. Danh mục bảng phân ca	17
2.3. Danh mục ca làm:.....	18
2.4. Danh mục hóa đơn	19
2.5. Danh mục sản phẩm	19
2.6. Danh mục bàn đặt.....	20
2.7. Danh mục khách hàng:.....	21
3. Quản lý khách hàng	22
3.1. Hàm tìm kiếm khách hàng	22
3.2. Tạo khách hàng mới.....	23
3.3. Sửa thông tin khách hàng:.....	24
4. Quản lý hóa đơn	25
4.1. Xuất hóa đơn	25
4.2. Tìm kiếm hóa đơn	25
5. Quản lý sản phẩm	26
5.1. Thêm sản phẩm	26
5.2. Sửa sản phẩm	27

5.3. Xóa sản phẩm.....	28
5.4. Tìm kiếm sản phẩm.....	29
6. Thống kê doanh thu theo ngày, tháng, năm	30
7. Quản lý nhân viên.....	31
7.1. Thêm nhân viên.....	31
7.2. Xóa nhân viên	32
7.3. Sửa thông tin nhân viên.....	33
7.4. Tìm kiếm nhân viên	34
7.5. Tính lương nhân viên	35
8. Quản lý đặt bàn.....	36
8.1. Cập nhật trạng thái bàn	36
9. Quản lý phân ca	37
9.1. Thêm ca cho nhân viên	37
9.2. Xóa ca làm của nhân viên	38
CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN	39
CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG	40
Giao diện và chức năng chương trình:	40
1. Form Main:.....	40
2. Form đăng nhập:	40
3. Các form danh mục:	41
3.1. Danh mục nhân viên:	41
3.2. Danh mục bảng phân ca:.....	41
3.3. Danh mục ca làm:	41
3.4. Danh mục hóa đơn:	42
3.5. Danh mục sản phẩm:.....	42
3.6. Danh mục bàn đặt:	43
3.7. Danh mục khách hàng:	43

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

1. Mô tả bài toán:

Mục tiêu chính của hệ thống quản lý nhà hàng này là tối ưu hóa hoạt động hàng ngày của nhà hàng, giúp giảm thời gian làm việc, tiết kiệm chi phí và tối ưu hóa sức lao động, từ đó tạo ra hiệu suất kinh tế cao cho nhà hàng. Chương trình quản lý được xây dựng cho hai đối tượng người dùng là quản lý và nhân viên với các quyền khác nhau.

Người dùng là nhân viên sẽ được cấp quyền truy cập vào các chức năng như bán hàng, xem doanh thu và xem lịch làm việc. Trong khi đó, người dùng là quản lý của nhà hàng sẽ có quyền truy cập và quản lý các chức năng như quản lý doanh thu, quản lý nhân viên và quản lý lịch làm việc. Đối với hệ thống quản lý nhà hàng này cần lưu trữ những dữ liệu sau:

Nhà hàng có nhiều nhân viên, mỗi nhân viên được phân công vào một công việc cụ thể. Mỗi công việc được phân biệt bằng mã công việc, bao gồm các thông tin như mã công việc, tên công việc và mức lương. Nhân viên được phân biệt bằng mã nhân viên, mỗi nhân viên gồm các thông tin cá nhân như mã nhân viên, họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, mã công việc, thưởng, ngày tuyển dụng, hình thức công việc, số ca.

Khách hàng của nhà hàng được phân biệt với nhau bằng mã khách hàng. Mỗi khách hàng có các thông tin như mã khách hàng, tên, số điện thoại, điểm thưởng tích lũy. Khi tích lũy đủ điểm thưởng, khách hàng sẽ được quy đổi điểm thưởng sang thành voucher để giảm giá trên toàn bill, với mỗi lần ăn khách hàng sẽ được cộng 1 điểm thưởng vào thẻ thành viên của mình, với mỗi 10 điểm giảm 5%.

Các hóa đơn được phân biệt với nhau qua mã hóa đơn, mỗi hóa đơn gồm các thông tin như mã hóa đơn, ngày lập hóa đơn, mã khách hàng, mã bàn, trạng thái, trị giá hóa đơn, phương thức thanh toán và mã nhân viên thu ngân lập hóa đơn.

Các sản phẩm phân biệt với nhau bằng mã sản phẩm. Mỗi sản phẩm có các thông tin như mã sản phẩm, tên sản phẩm, đơn giá, tình trạng, loại sản phẩm.

Một khách hàng có thể có nhiều đánh giá, một đánh giá chỉ có thể được thực hiện bởi một khách hàng. Mỗi đánh giá gồm Mã đánh giá, chất lượng, bình luận, mã khách hàng và phân biệt với nhau bởi mã đánh giá.

Các bàn trong nhà hàng phân biệt với nhau bằng mã bàn, mỗi bàn có các thông tin như mã bàn, trạng thái (trống, đặt trước, có khách), tiền đặt cọc và mã khách hàng.

Các nhân viên sẽ được chia ca để làm việc. Một nhân viên có thể làm nhiều ca, một ca có thể có nhiều nhân viên. Các ca làm việc được phân biệt với nhau qua mã ca, mỗi ca có các thông tin như mã ca, ngày, giờ bắt đầu, giờ kết thúc. Một ngày sẽ có 3 ca: ca sáng, ca chiều và ca tối (mỗi ca 5 tiếng). Bảng thống kê tài chính sẽ là điều tất yếu để

duy trì các hoạt động khác của quán bao gồm mã giao dịch, loại giao dịch (doanh thu, chi phí), ngày giao dịch, số tiền, mô tả.

2. Nghiệp vụ bài toán:

Bài toán quản lý nhà hàng là một phần quan trọng của ngành dịch vụ nhà hàng và du lịch, yêu cầu sự chuyên nghiệp và khéo léo trong việc quản lý hoạt động hằng ngày, quản lý nhân viên, hóa đơn và sản phẩm, doanh thu và hoạt động chung của nhà hàng. Nhân viên là hình ảnh đại diện của nhà hàng, đóng vai trò quan trọng trong giao tiếp và hỗ trợ khách hàng từ lúc họ bước vào cho đến khi họ rời nhà hàng. Vì thế, chúng ta cần xây dựng một quy trình đào tạo toàn diện cho nhân viên với các kỹ năng như: tác nghiệp, văn hóa phục vụ, ứng xử...

- Quản lý nhân viên:

+ Yêu cầu khi tuyển dụng: Phải có trình độ, kèm theo các kỹ năng, kinh nghiệm, khả năng làm việc nhóm, sức khỏe tốt và có tinh thần phục vụ.

+ Hình thức (HT) công việc: Làm toàn thời gian hoặc bán thời gian, không được đi trễ, không được vắng không xin phép.

+ Lương: Mỗi nhân sự sẽ thỏa thuận với quản lý về lương của mình trong hợp đồng.

+ Làm việc: Nhân viên đã được tuyển dụng sẽ trải qua đợt đào tạo về nghiệp vụ. Trong thời gian đó, quản lý sẽ là người quan sát, hướng dẫn để nhân sự thực hiện tốt hơn.

- Khi khách đến nhà hàng, quy trình thực hiện thường bao gồm các bước sau:

Nhân viên lễ tân sẽ chào đón khách và cung cấp thông tin cần thiết về dịch vụ và tiện ích của nhà hàng. Kiểm tra thông tin đặt bàn của khách.

Sau khi kiểm tra thông tin đặt bàn, nhân viên lễ tân sẽ đưa khách tới bàn đã chuẩn bị sẵn, nếu khách chưa đặt bàn thì nhân viên lễ tân sẽ xem coi còn bàn nào trống để khách vào ngồi, nếu không còn bàn trống thì nhân viên lễ tân sẽ hỗ trợ khách đặt bàn vào một thời gian khác. Ngay sau khi khách ngồi xuống bàn, nhân viên phục vụ sẽ tiếp cận khách, chào hỏi và giới thiệu thực đơn. Ghi chú các yêu cầu thêm của khách như: món chay, món có vị cay, mặn hay một số thành phần gia vị khác. Nhân viên phục vụ sẽ chuyển thông tin đến bộ phận bếp để bếp chuẩn bị và mang thức ăn tới bàn. Khi khách đã hoàn thành bữa ăn, nhân viên phục vụ sẽ mang hóa đơn và thanh toán tới bàn. Cuối cùng nói lời cảm ơn và tiễn khách ra khỏi quán.

- Quản lý hóa đơn, sản phẩm, quản lý doanh thu và điều hành nhà hàng chia thành các giai đoạn:

+ Giai đoạn lập kế hoạch và chuẩn bị:

- **Xác định hệ thống quản lý hóa đơn:** Lựa chọn hệ thống quản lý hóa đơn phù hợp để ghi nhận, xử lý và lưu trữ thông tin về hóa đơn khách hàng.
- **Thiết lập quy trình sản phẩm:** Xác định các sản phẩm và dịch vụ cung cấp bởi nhà hàng, đặt danh sách sản phẩm, mô tả chi tiết và giá cả.
- **Thiết lập quy trình quản lý doanh thu:** Xác định các phương pháp thu tiền, quy trình thanh toán và hình thức thanh toán (tiền mặt, thẻ, chuyển khoản...).
- **Đào tạo nhân viên:** Đào tạo nhân viên về quy trình lập hóa đơn, quản lý sản phẩm và quản lý doanh thu.

+ **Giai đoạn vận hành:** Sau khi khởi động, đảm bảo rằng hóa đơn được lập đầy đủ, chính xác và gửi đến khách hàng theo quy trình đã thiết lập. Điều này bao gồm việc lập hóa đơn cho các đơn hàng, ghi chú các món ăn và dịch vụ được yêu cầu, tính toán giá cả và áp dụng các chiết khấu (nếu có). Kiểm soát số lượng và chất lượng sản phẩm trong kho, đảm bảo rằng các sản phẩm được cung cấp đúng theo yêu cầu của khách hàng và thực đơn. Theo dõi doanh thu hàng ngày, lập báo cáo doanh thu, phân tích xu hướng và biểu đồ tăng trưởng, đánh giá hiệu quả kinh doanh và điều chỉnh chiến lược bán hàng.

+ **Giai đoạn quản lý sau bán hàng:** Kiểm tra tính chính xác của hóa đơn, xử lý thanh toán từ khách hàng, và cung cấp hỗ trợ cho khách hàng trong việc giải quyết các vấn đề liên quan đến hóa đơn. Thu thập phản hồi từ khách hàng về chất lượng sản phẩm, dịch vụ và trải nghiệm, và đáp ứng các yêu cầu hoặc khiếu nại của khách hàng. Quản lý quy trình liên quan đến việc hoàn trả hoặc đổi trả sản phẩm, xử lý yêu cầu của khách hàng và tạo sự hài lòng sau bán hàng.

3. Mô tả chức năng của bài toán:

- Đối với nhân viên:

+ **Chức năng bán hàng:** cho phép nhân viên chọn số bàn, tạo đơn hàng cho bàn, thêm món, xóa món, xuất hóa đơn thanh toán.

+ **Quản lý khách hàng:** cho phép nhân viên tạo thẻ thành viên cho khách hàng, nhập số điện thoại ở bước thanh toán để tích điểm cho khách hàng.

+ **Xem doanh thu:** cho phép nhân viên có thể xem được doanh thu bán của bản thân trong ca làm việc của mình.

+ **Xem bảng ca làm việc:** cho phép nhân viên xem bảng phân ca làm việc trong tuần.

- Đối với người quản lý:

+ **Quản lý nhân viên:** cho phép người quản lý nhập, thêm, xóa, sửa thông tin của nhân viên. Người quản lý có thể xem các thông tin để liên lạc được với nhân viên như số điện thoại.

+ **Quản lý ca làm việc của nhân viên:** Các nhân viên sẽ đăng ký ca với người quản lý và người quản lý sẽ thêm ca làm việc vào bảng phân ca cho các nhân viên. Người quản lý có quyền thêm, xóa, sửa ca làm việc của các nhân viên.

+ **Quản lý doanh thu:** cho phép nhà hàng tính toán doanh thu theo ngày, tuần, tháng hoặc năm, dựa trên thông tin về đơn hàng và giao dịch tài chính.

+ **Quản lý tài chính:** cho người quản lý tạo ra các giao dịch tài chính, bao gồm mã giao dịch, loại giao dịch (doanh thu hoặc chi phí), ngày giao dịch, số tiền giao dịch và mô tả giao dịch.

4. Tính lương nhân viên:

Nhà hàng có 2 hình thức công việc là lương cố định và lương theo giờ.

- Lương cố định là những công việc như quản lý, đầu bếp.

- Lương theo giờ là những công việc như lễ tân, phục vụ.

Tính lương nhân viên trong nhà hàng có thể bao gồm các yếu tố sau:

1. **Lương cố định:** Đây là mức lương cố định mà nhân viên nhận hàng tháng. Dựa trên vị trí và mức độ kinh nghiệm của nhân viên. Các công việc lương cố định của nhà hàng yêu cầu thời gian làm việc là 1 tháng chỉ được nghỉ phép 2 ngày.

Quản lý: 15.000.000 đồng

Đầu bếp: 15.000.000 đồng

2. **Lương theo giờ:** Sẽ được trả theo số giờ làm việc thực tế. Mức lương theo giờ cũng khác nhau cho từng công việc và có thể được xác định bởi chính sách và quy định của nhà hàng. Dưới đây là mức lương theo giờ của các vị trí trong nhà hàng:

Phục vụ: 60.000 đồng/giờ

Lễ tân: 60.000 đồng/giờ

3. **Lương thưởng:** Khoản thưởng có thể được xác định dựa trên thành tích cá nhân, thành tích nhóm hoặc doanh thu của nhà hàng vượt mức hay ngày lễ lương thưởng sẽ nhận được 15% lương.

Ví dụ:

Đối với lương cố định thì mức thưởng = 10% * lương cố định.

Đối với lương theo giờ mức thưởng = 10% * số giờ * lương theo giờ.

Tổng lương:

- Lương cố định: Lương theo tháng + lương thưởng.

- Lương theo giờ: Lương theo giờ * số giờ làm trong tháng + lương thưởng.

Đối với quản lý, lương tháng nếu có thưởng = $15.000.000 + 15\% * 15.000.000 = 17.250.000đ$

Đối với 1 phục vụ tháng này làm 40 ca (1 ca 4 giờ) được thưởng thì lương là $(40 * 4 * 60.000) * 1.15 = 11.040.000đ$

Ngoài ra còn có các lợi ích khác như bảo hiểm y tế, bảo hiểm tai nạn, hỗ trợ ăn trưa, chế độ nghỉ phép và các chương trình khuyến mại khác cho nhân viên.

5. Doanh thu nhà hàng:

Đối với mỗi hóa đơn, giá trị hóa đơn = tổng tiền của các đơn giá từng món * số lượng mỗi món.

Ví dụ: khách hàng gọi 1 gà quay 200.000đ và 2 nước cam 30.000đ thì tổng hóa đơn = $1 * 200.000 + 2 * 30.000 = 260.000đ$

Đối với những bàn đặt trước, phí đặt cọc sẽ là 50.000đ.

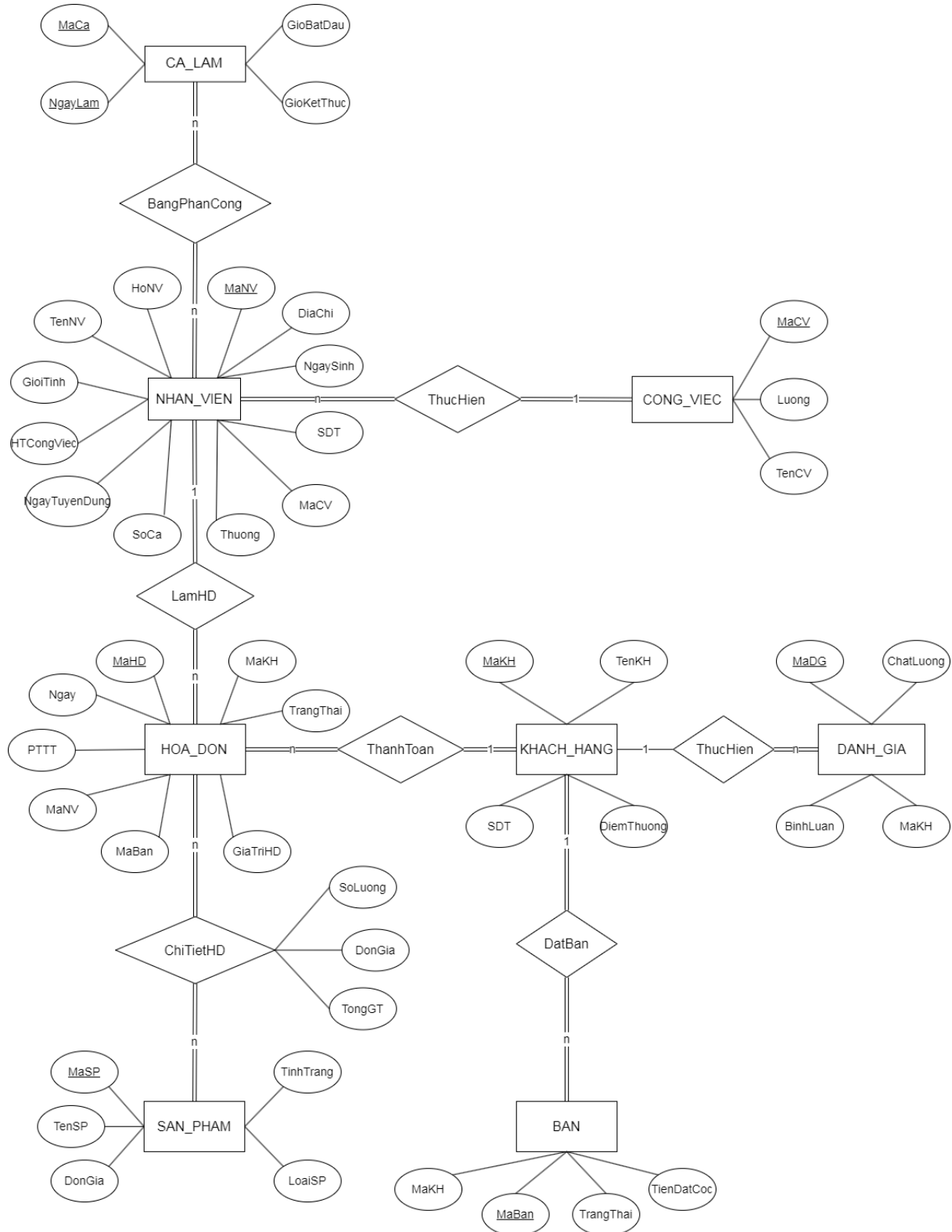
Doanh thu trong ngày được tính theo tổng giá trị các hóa đơn trong ngày hôm đó

Doanh thu tháng được tính theo tổng giá trị hóa đơn tháng đó. Nếu doanh thu tháng này vượt mức 50 triệu thì nhân viên được thưởng.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. Thiết kế cơ sở dữ liệu mức quan niệm:

Từ mô tả về dữ liệu cần có ở phần mô tả của bài toán ta hình thành được sơ đồ thực thể kết hợp (ERD).



2. Thiết kế cơ sở dữ liệu mức logic:

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

- CONG_VIEC (MaCV, Luong, TenCV).
- DANH_GIA (MaDG, ChatLuong, BinhLuan, MaKH).
- CA_LAM (MaCa, NgayLam, GioBatDau, GioKetThuc).
- NHAN_VIEN (MaNV, HoNV, TenNV, NgaySinh, SDT, MaCV, HTCongViec, NgayTuyenDung, SoCa, Thuong, GioiTinh, DiaChi).
- HOA_DON (MaHD, MaKH, Ngay, TrangThai, PTTT, MaNV, MaBan, GiaTriHD)
- KHACH_HANG (MaKH, TenKH, SDT, DiemThuong).
- SAN_PHAM (MaSP, TinhTrang, TenSP, DonGia, LoaiSP).
- BAN (MaBan, MaKH, TrangThai, TienDatCoc).
- BangPhanCong (MaCa, MaNV, NgayLam).
- ChiTietHD (MaSP, MaHD, SoLuong, DonGia, TongGT).

3. Các ràng buộc cần có:

STT	Bảng	Ràng buộc
1	CONG_VIEC	Khóa chính: MaNV Check Luong > 0
2	DANH_GIA	Khóa chính: MaDG Khóa ngoại: MaKH đến bảng KHACH_HANG Check ChatLuong > 0 and ChatLuong <= 5
3	CA_LAM	Khóa chính: MaCa và NgayLam NOT NULL GioBatDau NOT NULL GioKetThuc
4	NHAN_VIEN	Khóa chính: MaNV Khóa ngoại: MaCV đến bảng CONG_VIEC NOT NULL HoNV, TenNV, SDT, CHECK NgaySinh (>=18 tuổi), CHECK SDT (=10), CHECK NgayTuyenDung (>0)
5	HOA_DON	Khóa chính: MaHD Khóa ngoại: MaKH đến bảng KHACH_HANG, MaNV đến bảng NHAN_VIEN, MaBan đến bảng BAN NOT NULL GiaTriHD CHECK Ngay (>0)
6	KHACH_HANG	Khóa chính: MaKH NOT NULL TenKH NOT NULL SDT CHECK SDT (=10)
7	SAN_PHAM	Khóa chính: MaSP NOT NULL TenSP NOT NULL LoaiSP NOT NULL DonGia
8	BAN	Khóa chính: MaBan Khóa ngoại: MaKH đến bảng KHACH_HANG NOT NULL TrangThai
9	BangPhanCong	Khóa chính: MaNV, MaCa, NgayLam Khóa ngoại: (MaCa, NgayLam) đến bảng CA_LAM, MaNV đến bảng NHAN_VIEN
10	ChiTietHD	Khóa chính: MaHD, MaSP Khóa ngoại: MaHD đến bảng HOA_DON, MaSP đến bảng SAN_PHAM Check SoLuong (>0)

4. Cài đặt các CSDL và các ràng buộc:

- Bảng công việc:

```
CREATE TABLE CONG_VIEC(  
    MaCV nvarchar(20) CONSTRAINT PK_CONG_VIEC PRIMARY KEY,  
    TenCV nvarchar(50) NOT NULL,  
    Luong float check (Luong >0)  
)
```

- Bảng ca làm:

```
CREATE TABLE CA_LAM(  
    MaCa nvarchar(20),  
    NgayLam date check (DATEDIFF(day, NgayLam, GETDATE())>=0),  
    GioBatDau nchar(10) NOT NULL,  
    GioKetThuc nchar(10) NOT NULL,  
    CONSTRAINT PK_Ca_Lam PRIMARY KEY (MaCa, NgayLam)  
)
```

- Bảng nhân viên:

```
CREATE TABLE NHAN_VIEN(  
    MaNV nvarchar(10) CONSTRAINT PK_NHAN_VIEN PRIMARY KEY,  
    HoNV nvarchar(10) NOT NULL,  
    TenNV nvarchar(10) NOT NULL,  
    NgaySinh date check (DATEDIFF(year, NgaySinh, GETDATE())>=18),  
    SDT nchar(11) check (len(SDT)=10),  
    MaCV nvarchar(20) CONSTRAINT FK_NHANVIEN_CV FOREIGN KEY REFERENCES CONG_VIEC(MaCV),  
    SoCa int,  
    Thuong int,  
    NgayTuyenDung date check (DATEDIFF(day, NgayTuyenDung, GETDATE())>=0),  
    HTCongViec nvarchar(20)  
)
```

- Bảng khách hàng:

```
CREATE TABLE KHACH_HANG(  
    MaKH nvarchar(10) CONSTRAINT PK_KHACH_HANG PRIMARY KEY,  
    TenKH nvarchar(50) NOT NULL,  
    SDT nchar(11) NOT NULL check (len(SDT)=10),  
    DiemThuong int  
)
```

- Bảng bàn:

```
CREATE TABLE BAN(  
    MaBan nvarchar(10) CONSTRAINT PK_BAN PRIMARY KEY,  
    TrangThai nvarchar (20) NOT NULL,  
    TienDatCoc int check (TienDatCoc > 0),  
    MaKH nvarchar(10) CONSTRAINT FK_BAN_KH_DATCOC FOREIGN KEY REFERENCES KHACH_HANG(MaKH),  
)
```

- Bảng đánh giá:

```
CREATE TABLE DANH_GIA(  
    MaDG nvarchar(10) CONSTRAINT PK_DANH_GIA PRIMARY KEY,  
    ChatLuong int check(ChatLuong > 0 and ChatLuong <= 5),  
    MaKH nvarchar(10) CONSTRAINT FK_DANHGIA_KH FOREIGN KEY REFERENCES KHACH_HANG(MaKH),  
    BinhLuan nvarchar(50)  
)
```

- Bảng sản phẩm:

```
CREATE TABLE SAN_PHAM(  
    MaSP nvarchar(10) CONSTRAINT PK_SAN_PHAM PRIMARY KEY,  
    TenSP nvarchar(50) NOT NULL,  
    LoaiSP nvarchar(20) NOT NULL,  
    TinhTrang nvarchar(20),  
    DonGia float check(DonGia > 0) NOT NULL  
)
```

- Bảng hóa đơn:

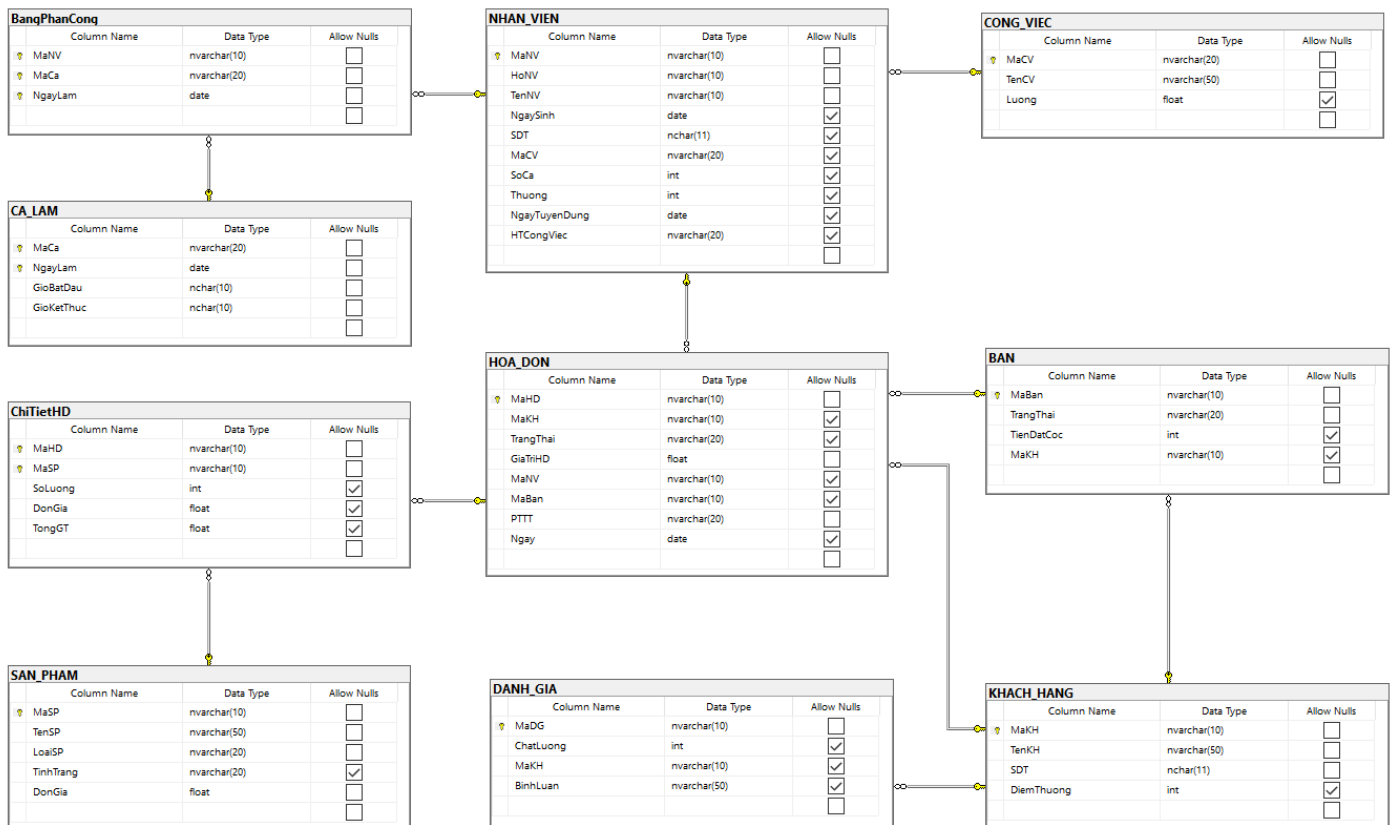
```
CREATE TABLE HOA_DON(  
    MaHD nvarchar(10) CONSTRAINT PK_HOA_DON PRIMARY KEY,  
    MaKH nvarchar(10) CONSTRAINT FK_HOADON_KH FOREIGN KEY REFERENCES KHACH_HANG(MaKH),  
    TrangThai nvarchar(20) ,  
    GiaTriHD float NOT NULL,  
    MaNV nvarchar(10) CONSTRAINT FK_HOADON_NV FOREIGN KEY REFERENCES NHAN_VIEN(MaNV),  
    MaBan nvarchar(10) CONSTRAINT FK_HOADON_BAN FOREIGN KEY REFERENCES BAN(MaBan),  
    PTTT nvarchar(20) NOT NULL ,  
    Ngay date check (DATEDIFF(year, Ngay, GETDATE())>=0),  
)
```

- Bảng chi tiết hóa đơn:

```
CREATE TABLE ChiTietHD(  
    MaHD nvarchar(10) CONSTRAINT FK_CTHD_HD FOREIGN KEY REFERENCES HOA_DON(MaHD),  
    MaSP nvarchar(10) CONSTRAINT FK_CTHD_SP FOREIGN KEY REFERENCES SAN_PHAM(MaSP),  
    SoLuong int check( SoLuong > 0) ,  
    DonGia float check(DonGia > 0),  
    TongGT float check(TongGT > 0),  
    CONSTRAINT PK_CTHD PRIMARY KEY (MaHD, MaSP)  
)
```

- Bảng phân công công việc:

```
CREATE TABLE BangPhanCong(
    MaNV nvarchar(10) CONSTRAINT FK_BPC_NV FOREIGN KEY REFERENCES NHAN_VIEN(MaNV),
    MaCa nvarchar(20) ,
    NgayLam date check (DATEDIFF(day, NgayLam, GETDATE())>=0),
    CONSTRAINT PK_BPC PRIMARY KEY (MaCa, MaNV, NgayLam),
    CONSTRAINT FK_BPC_CALAM FOREIGN KEY (MaCa, NgayLam) REFERENCES CA_LAM(MaCa, NgayLam)
)
```



5. Các view:

- Xem ca làm việc của nhân viên trong ngày:

```
CREATE VIEW V_CaLamViec AS
SELECT clv.*, nv.TenNV, nv.MaCV, cv.TenCV
FROM dbo.BangPhanCong pc
    JOIN dbo.NHAN_VIEN nv ON pc.MaNV = nv.MaNV
    JOIN dbo.CONG_VIEC cv ON cv.MaCV = nv.MaCV
    JOIN dbo.CA_LAM clv ON clv.MaCa = pc.MaCa
```

- Xem danh sách sản phẩm:

```
CREATE VIEW V_Pro_DanhSachSanPham AS
SELECT MaSP, TenSP, LoaiSP, TinhTrang, DonGia
FROM dbo.SAN_PHAM
```

- Xem số lượng sản phẩm đã bán trong ngày:

```
CREATE VIEW V_SoLuongSanPhamDaBanTrongNgay AS
SELECT sp.MaSP, SUM(cthd.SoLuong) AS SoLuongDaBanTaiQuay
FROM dbo.SAN_PHAM sp
    JOIN dbo.ChiTietHD cthd ON sp.MaSP = cthd.MaSP
    JOIN dbo.HOA_DON hd ON cthd.mahd = hd.MaHD
WHERE hd.Ngay = CONVERT(DATE, GETDATE())
GROUP BY sp.MaSP
```

- Xem thông tin nhân viên:

```
CREATE VIEW [dbo].[V_LoadNhanVien]
AS
SELECT nv.MaNV, nv.HoNV, nv.TenNV, nv.NgaySinh, nv.SDT, nv.MaCV, nv.SoCa,
    nv.Thuong, nv.NgayTuyenDung, nv.HTCongViec, cv.TenCV
FROM NHAN_VIEN nv INNER JOIN CONG_VIEC cv ON nv.MaCV = cv.MaCV
```

- Xem bảng phân công:

```
CREATE VIEW V_BangPhanCong
AS
SELECT *
FROM BangPhanCong
```

- Xem trạng thái bàn:

```
CREATE VIEW [dbo].[V_BAN]
AS
SELECT *
FROM BAN
```


6. Các trigger:

- Trigger đặt trạng thái hóa đơn:

```
CREATE TRIGGER trigger_HD
ON HOA_DON
AFTER INSERT
AS
BEGIN
    IF @@ROWCOUNT > 0
    BEGIN
        UPDATE HOA_DON
        SET TrangThai = 'Chưa thanh toán'
        FROM INSERTED
        WHERE HOA_DON.MaHD = INSERTED.MaHD
    END
END
```

- Trigger đặt trạng thái đặt chỗ của bàn:

```
CREATE TRIGGER trigger_BAN
ON BAN
AFTER INSERT
AS
BEGIN
    IF @@ROWCOUNT > 0
    BEGIN
        UPDATE BAN
        SET TrangThai = 'Đã đặt'
        FROM INSERTED
        WHERE BAN.MaKH = INSERTED.MaKH AND BAN.MaBan = INSERTED.MaBan
    END
END
```

- Trigger kiểm tra trùng lặp số điện thoại của khách hàng:

```
CREATE TRIGGER trigger_CheckSDTKH
ON KHACH_HANG
FOR INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT * FROM inserted WHERE TRIM(SDT) = '')
    BEGIN
        RAISERROR('Số điện thoại không được để trống', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF NOT EXISTS (SELECT * FROM inserted WHERE LEN(TRIM(SDT)) = (10))
    BEGIN
        RAISERROR('Số điện thoại phải có 10 số', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF EXISTS (SELECT 1 FROM inserted i INNER JOIN KHACH_HANG kh
    ON i.MaKH != kh.MaKH AND TRIM(i.SDT) = TRIM(kh.SDT))
    BEGIN
        RAISERROR('Số điện thoại đã tồn tại', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
END
```

- Trigger kiểm tra tên sản phẩm có bị trùng không:

```
CREATE TRIGGER trigger_CheckTenSP
ON SAN_PHAM
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @TenSP nvarchar(50)
    SET @TenSP = (SELECT TenSP FROM INSERTED)
    IF EXISTS (SELECT * FROM SAN_PHAM, INSERTED WHERE SAN_PHAM.TenSP = @TenSP AND SAN_PHAM.MaSP <> INSERTED.MaSP)
    BEGIN
        RAISERROR ('Tên sản phẩm đã tồn tại', 16, 1)
        ROLLBACK TRANSACTION
    END
END
```

- Trigger bắt lỗi khi thêm hoặc sửa thông tin nhân viên:

```
CREATE TRIGGER trigger_CheckNhanVien
ON NHAN_VIEN
FOR INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT * FROM inserted WHERE TRIM(TenNV) = '')
    BEGIN
        RAISERROR('Tên nhân viên không được để trống', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF EXISTS (SELECT * FROM inserted WHERE TRIM(HoNV) = '')
    BEGIN
        RAISERROR('Họ nhân viên không được để trống', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF EXISTS (SELECT * FROM inserted
    WHERE DATEDIFF(year, inserted.NgaySinh, GETDATE()) < (18))
    BEGIN
        RAISERROR('Nhân viên phải trên 18 tuổi', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF EXISTS (SELECT * FROM inserted WHERE TRIM(SDT) = '')
    BEGIN
        RAISERROR('Số điện thoại không được để trống', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF NOT EXISTS (SELECT * FROM inserted WHERE LEN(TRIM(SDT)) = (10))
    BEGIN
        RAISERROR('Số điện thoại phải có 10 số', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF EXISTS (SELECT 1 FROM inserted i INNER JOIN NHAN_VIEN nv
    ON i.MaNV != nv.MaNV AND TRIM(i.SDT) = TRIM(nv.SDT))
    BEGIN
        RAISERROR('Số điện thoại đã tồn tại', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
    IF NOT EXISTS (SELECT * FROM inserted
    WHERE DATEDIFF(day, inserted.NgayTuyenDung, GETDATE()) >= (0))
    BEGIN
        RAISERROR('Ngày tuyển dụng không thể trong tương lai', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
END
```

CHƯƠNG 3: THIẾT KẾ CHỨC NĂNG

1. Kết nối cơ sở dữ liệu

```
public class getConnection
{
    private const string sa = @"Data Source=BLAP;Initial Catalog=QLNH;Integrated Security=True";

    SqlConnection conn = new SqlConnection(sa);

    9 references
    public SqlConnection GetSqlConnection()
    {
        return conn;
    }

    9 references
    public void openConnection()
    {
        if(conn.State == ConnectionState.Closed)
        {
            conn.Open();
        }
    }

    12 references
    public void closeConnection()
    {
        if(conn.State == ConnectionState.Open)
        {
            conn.Close();
        }
    }
}
```

2. Xem thông tin danh mục

2.1. Danh mục nhân viên

- SQL:

```
CREATE VIEW [dbo].[V_LoadNhanVien]
AS
SELECT nv.MaNV, nv.HoNV, nv.TenNV, nv.NgaySinh, nv.SDT, nv.MaCV, nv.SoCa,
nv.Thuong, nv.NgayTuyenDung, nv.HTCongViec, cv.TenCV
FROM NHAN_VIEN nv INNER JOIN CONG_VIEC cv ON nv.MaCV = cv.MaCV
```

- C#:

```
public DataTable getAllEmployee()
{
    string cmd = "SELECT * FROM V_LoadNhanVien";
    db.openConnection();
    SqlDataAdapter adapter = new SqlDataAdapter(cmd, db.GetSqlConnection());
    db.closeConnection();
    DataTable dt = new DataTable();
    adapter.Fill(dt);
    if (dt.Rows.Count <= 0)
    {
        Console.WriteLine("Không đọc được thông tin");
    }
    return dt;
}
```

2.2. Danh mục bảng phân ca

- SQL:

```
CREATE PROCEDURE proc_PhanCong
    @MaNV nvarchar(10),
    @MaCa nvarchar(20),
    @NgayLam date
AS
BEGIN
    INSERT INTO BangPhanCong (MaNV, MaCa, NgayLam)
    VALUES (@MaNV, @MaCa, @NgayLam)
END

CREATE VIEW V_BangPhanCong
AS
SELECT *
FROM BangPhanCong
```

-C#:

```
public DataTable GetPhanCong()
{
    DataTable dtTable = new DataTable();

    string query = "SELECT * FROM V_BangPhanCong";
    SqlConnection con = getConnection.GetSqlConnection();
    try
    {
        con.Open();
        SqlDataAdapter sda = new SqlDataAdapter(query, con);
        sda.Fill(dtTable);
    }
    catch (Exception ex)
    {
        throw (ex);
    }
    finally
    {
        con.Close();
    }
    return dtTable;
}
```

2.3. Danh mục ca làm:

- SQL:

```
CREATE VIEW V_CaLamViec AS
SELECT clv.*, nv.TenNV, nv.MaCV, cv.TenCV
FROM dbo.BangPhanCong pc
    JOIN dbo.NHAN_VIEN nv ON pc.MaNV = nv.MaNV
    JOIN dbo.CONG_VIEC cv ON cv.MaCV = nv.MaCV
    JOIN dbo.CA_LAM clv ON clv.MaCa = pc.MaCa

CREATE PROCEDURE proc_GetCaLamViecByDate
    @NgayLam DATE
AS
BEGIN
    SELECT *
    FROM V_CaLamViec
    WHERE NgayLam = @NgayLam;
END;
```

- C#:

```
public DataTable GetAllShifts(string dateTime)
{
    DataTable dtTable = new DataTable();

    string query = $"proc_GetCaLamViecByDate @NgayLam = '{dateTime}'";
    SqlConnection con = getConnection.GetSqlConnection();
    try
    {
        con.Open();
        SqlDataAdapter sda = new SqlDataAdapter(query, con);
        sda.Fill(dtTable);
    }
    catch (Exception ex)
    {
        throw (ex);
    }
    finally
    {
        con.Close();
    }
    return dtTable;
}
```

2.4. Danh mục hóa đơn

- SQL:

```
CREATE VIEW [dbo].[V_XemTatCaHoaDon] AS
    SELECT MaHD, MaKH, TrangThai, GiaTriHD, MaNV, MaBan, PTTT, Ngay, Gio
    FROM dbo.HOA_DON
```

- C#:

```
public DataTable GetAllBill()
{
    dtBill = new DataTable();
    dtBill.Clear();
    string queryView = "SELECT * FROM V_XemTatCaHoaDon";

    con.Open();
    SqlDataAdapter sda = new SqlDataAdapter(queryView, con);
    con.Close();
    sda.Fill(dtBill);

    return dtBill;
}
```

2.5. Danh mục sản phẩm

- SQL:

```
CREATE VIEW V_Pro_DanhSachSanPham AS
    SELECT MaSP, TenSP, LoaiSP, TinhTrang, DonGia
    FROM dbo.SAN_PHAM
```

- C#:

```
public DataTable getAllProduct_Product()
{
    string queryView = "SELECT * FROM V_Pro_DanhSachSanPham";
    SqlConnection con = getConnection.GetSqlConnection();
    con.Open();
    SqlDataAdapter sda = new SqlDataAdapter(queryView, con);
    con.Close();
    DataTable dtproduct = new DataTable();
    sda.Fill(dtproduct);
    return dtproduct;
}
```

2.6. Danh mục bàn đặt

- SQL:

```
CREATE VIEW [dbo].[V_BAN]
AS
SELECT *
FROM BAN
```

- C#:

```
public DataTable GetAllTables()
{
    DataTable dtTable = new DataTable();

    string query = "SELECT * FROM V_BAN";
    SqlConnection con = getConnection.GetSqlConnection();
    con.Open();
    SqlDataAdapter sda = new SqlDataAdapter(query, con);
    con.Close();
    sda.Fill(dtTable);

    if (dtTable.Rows.Count > 0)
    {
        Console.WriteLine("cor");
    }
    else
    {
        Console.WriteLine("Sai ");
    }
    return dtTable;
}
```


2.7. Danh mục khách hàng:

- SQL:

```
CREATE VIEW V_KhachHang AS
    SELECT MaKH, TenKH, SDT, DiemThuong
    FROM dbo.KHACH_HANG
```

- C#:

```
public DataTable getAllCustomer()
{
    string queryView = "SELECT * FROM V_KhachHang";
    dtCustomer = new DataTable();
    con.Open();
    SqlDataAdapter sda = new SqlDataAdapter(queryView, con);

    dtCustomer.Clear();
    sda.Fill(dtCustomer);
    con.Close();
    return dtCustomer;
}
```

3. Quản lý khách hàng

3.1. Hàm tìm kiếm khách hàng

- SQL:

```
CREATE FUNCTION [dbo].[func_findCustomerByPhone] (@string NVARCHAR(11))
    RETURNS @CustomerLIST TABLE (MaKH NVARCHAR(10), TenKH NVARCHAR(50), SDT NCHAR(11), DiemThuong int)
    AS
    BEGIN
        INSERT INTO @CustomerLIST
        SELECT MaKH, TenKH, SDT, DiemThuong
        FROM dbo.KHACH_HANG
        WHERE SDT LIKE '%' + @string + '%'
        RETURN
    END

GO
USE QLNH
GO
CREATE FUNCTION [dbo].[func_findCustomerByID] (@string NVARCHAR(10))
    RETURNS @CustomerLIST TABLE (MaKH NVARCHAR(10), TenKH NVARCHAR(50), SDT NCHAR(11), DiemThuong int)
    AS
    BEGIN
        INSERT INTO @CustomerLIST
        SELECT MaKH, TenKH, SDT, DiemThuong
        FROM dbo.KHACH_HANG
        WHERE MaKH LIKE '%' + @string + '%'
        RETURN
    END
```

- C#:

```
public DataTable FindCustomerByPhone(string phone)
{
    dtCustomer = new DataTable();
    string func_name = "func_findCustomerByPhone";
    string queryView = $"SELECT * FROM {func_name} (@phone)";
    //SqlConnection con = getConnection.GetSqlConnection();
    con.Open();
    SqlCommand cmd = new SqlCommand(queryView, con);

    cmd.Parameters.AddWithValue("@phone", phone);

    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    dtCustomer.Clear();
    sda.Fill(dtCustomer);
    con.Close();

    return dtCustomer;
}

public DataTable FindCustomerByID(string ID)
{
    dtCustomer = new DataTable();
    string func_name = "func_findCustomerByID";
    string queryView = $"SELECT * FROM {func_name} (@ID)";
    //SqlConnection con = getConnection.GetSqlConnection();
    con.Open();
    SqlCommand cmd = new SqlCommand(queryView, con);

    cmd.Parameters.AddWithValue("@ID", ID);

    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    dtCustomer.Clear();
    sda.Fill(dtCustomer);
    con.Close();

    return dtCustomer;
}
```

3.2. Tạo khách hàng mới

- SQL:

```
CREATE PROCEDURE [dbo].[proc_ThemKH]
    (@MaKH nvarchar(10), @TenKH nvarchar(50) , @SDT nchar(11), @DiemThuong int)
AS
BEGIN
    BEGIN TRANSACTION
    BEGIN TRY
        -- Thêm mới sản phẩm
        INSERT INTO KHACH_HANG(MaKH, TenKH,SDT, DiemThuong)
        VALUES (@MaKH, @TenKH, @SDT, @DiemThuong)
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        ROLLBACK
        DECLARE @err NVARCHAR(MAX)
        SELECT @err = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@err, 16, 1)
    END CATCH
END
```

- C#:

```
public bool AddCustomer(CustomerDTO customerDTO, ref string err)
{
    string query = "proc_ThemKH @MaKH, @TenKH, @SDT , @DiemThuong";
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(query, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaKH", customerDTO.Id),
            new SqlParameter("@TenKH", customerDTO.Name),
            new SqlParameter("@SDT", customerDTO.Phone),
            new SqlParameter("@DiemThuong",customerDTO.Loyalreward),
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        err = ex.Message;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```

3.3. Sửa thông tin khách hàng:

- SQL:

```
CREATE PROCEDURE [dbo].[proc_SuaKH]
(@MaKH nvarchar(10), @TenKH nvarchar(50) , @SDT nchar(11), @DiemThuong int)
AS
BEGIN
    BEGIN TRY
        -- Thêm mới sản phẩm
        UPDATE dbo.KHACH_HANG SET MaKH = @MaKH, TenKH = @TenKH,
        SDT = @SDT , DiemThuong = @DiemThuong
        WHERE MaKH = @MaKH
    END TRY
    BEGIN CATCH
        DECLARE @err NVARCHAR(MAX)
        SELECT @err = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@err, 16, 1)
    END CATCH
END
```

- C#:

```
public bool UpdateCustomer(CustomerDTO customerDTO, ref string err) {
    string query = "proc_SuaKH @MaKH, @TenKH, @SDT, @DiemThuong";
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(query, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaKH", customerDTO.Id),
            new SqlParameter("@TenKH", customerDTO.Name),
            new SqlParameter("@SDT", customerDTO.Phone),
            new SqlParameter("@DiemThuong", customerDTO.Loyalreward),
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        err = ex.Message;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```

4. Quản lý hóa đơn

4.1. Xuất hóa đơn

- SQL:

```
CREATE PROCEDURE [dbo].[proc_get_single_BilljoinDetail]
    @MaHD int
AS
BEGIN
    SELECT * FROM V_Bill_DetailBill WHERE MaHD_HOA_DON = @MaHD
END
```

- C#:

```
public DataTable GetBillJoinDetail(int maHD)
{
    dtBill = new DataTable();
    dtBill.Clear();
    string queryView = "proc_get_single_BilljoinDetail";

    // Create a SqlCommand
    SqlCommand cmd = new SqlCommand(queryView, con);
    cmd.CommandType = CommandType.StoredProcedure; // Set command type as stored procedure

    // Add parameter to SqlCommand
    cmd.Parameters.Add(new SqlParameter("@MaHD", maHD));

    // Open the connection and execute the command
    con.Open();
    SqlDataAdapter sda = new SqlDataAdapter(cmd); // Use the SqlCommand instead of the string
    sda.Fill(dtBill);
    con.Close();

    return dtBill;
}
```

4.2. Tìm kiếm hóa đơn

- SQL:

- C#:

5. Quản lý sản phẩm

5.1. Thêm sản phẩm

- SQL:

```
CREATE PROCEDURE [dbo].[proc_ThemSanPham]
(@MaSP nvarchar(10), @TenSP nvarchar(50), @LoaiSP nvarchar(20),
@TinhTrang nvarchar(20), @DonGia float, @HinhAnh VARBINARY(MAX))
AS
BEGIN
    BEGIN TRANSACTION
    BEGIN TRY
        -- Thêm mới sản phẩm
        INSERT INTO SAN_PHAM(MaSP, TenSP, LoaiSP, TinhTrang, DonGia, AnhSP)
        VALUES (@MaSP, @TenSP, @LoaiSP, @TinhTrang, @DonGia, @HinhAnh)
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        ROLLBACK
        DECLARE @err NVARCHAR(MAX)
        SELECT @err = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@err, 16, 1)
    END CATCH
END
```

- C#:

```
public bool AddProduct(ProductDTO productDTO, ref string err)
{
    string queryInsert = "proc_ThemSanPham @MaSP, @TenSP, @LoaiSP, @TinhTrang, @DonGia, @HinhAnh";
    SqlConnection con = getConnection();
    string imageHex = null;
    try
    {
        MemoryStream ms = new MemoryStream();
        Image tmp = productDTO.Image;
        tmp.Save(ms, System.Drawing.Imaging.ImageFormat.Png);
        byte[] imageByteArray = ms.ToArray();
        //imageHex = BitConverter.ToString(imageByteArray).Replace("-", string.Empty);
        con.Open();
        SqlCommand cmd = new SqlCommand(queryInsert, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaSP", productDTO.Id),
            new SqlParameter("@TenSP", productDTO.Name),
            new SqlParameter("@LoaiSP", productDTO.Category),
            new SqlParameter("@TinhTrang", productDTO.State),
            new SqlParameter("@DonGia", productDTO.Price),
            new SqlParameter("@HinhAnh", imageByteArray)
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        err = ex.Message;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```

5.2. Sửa sản phẩm

- SQL:

```
CREATE PROCEDURE [dbo].[proc_suaSanPham]
@MaSP nvarchar(10), @TenSP nvarchar(50), @LoaiSP nvarchar(20),
@TinhTrang nchar(20), @DonGia float, @AnhSP varbinary(max)
AS
BEGIN
    BEGIN TRY
        -- Thêm mới sản phẩm
        UPDATE dbo.SAN_PHAM SET MaSP = @MaSP, TenSP = @TenSP,
        LoaiSP = @LoaiSP, TinhTrang = @TinhTrang, DonGia = @DonGia,
        AnhSP = @AnhSP
        WHERE MaSP = @MaSP
    END TRY
    BEGIN CATCH
        DECLARE @err NVARCHAR(MAX)
        SELECT @err = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@err, 16, 1)
    END CATCH
END
```

- C#:

```
public bool UpdateProduct(ProductDTO productDTO, ref string err)
{
    string query = "pro_suaSanPham @MaSP, @TenSP, @LoaiSP, @TinhTrang, @DonGia, @HinhAnh";
    try
    {
        MemoryStream ms = new MemoryStream();
        Image tmp = productDTO.Img;
        tmp.Save(ms, System.Drawing.Imaging.ImageFormat.Png);
        byte[] imageByteArray = ms.ToArray();
        con.Open();
        SqlCommand cmd = new SqlCommand(query, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaSP", productDTO.Id),
            new SqlParameter("@TenSP", productDTO.Name),
            new SqlParameter("@LoaiSP", productDTO.Category),
            new SqlParameter("@TinhTrang", productDTO.State),
            new SqlParameter("@DonGia", productDTO.Price),
            new SqlParameter("@HinhAnh", imageByteArray)
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        err = ex.Message;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```

5.3. Xóa sản phẩm

- SQL:

```
CREATE PROCEDURE [dbo].[proc_xoaSanPham]
@MaSP nvarchar(10)
AS
BEGIN
    BEGIN TRANSACTION
    BEGIN TRY
        --Xóa sản phẩm theo @MaSP trong bảng SanPham
        DELETE FROM dbo.SAN_PHAM WHERE SAN_PHAM.MaSP = @MaSP
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        ROLLBACK
        DECLARE @err NVARCHAR(MAX)
        SELECT @err = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@err, 16, 1)
    END CATCH
END
```

- C#:

```
public bool RemoveProduct(string id, ref string err)
{
    string queryDelete = "proc_xoaSanPham @MaSP";
    //SqlConnection conn = getConnection.GetSqlConnection();
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(queryDelete, con);
        cmd.Parameters.AddWithValue("@MaSP", id);
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch(Exception ex)
    {
        err = ex.Message;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```


5.4. Tìm kiếm sản phẩm

- SQL:

```
CREATE FUNCTION [dbo].[func_findProductByName] (@string NVARCHAR(50))
RETURNS @ProductList TABLE (MaSP NVARCHAR(10), TenSP NVARCHAR(50),
LoaiSP NVARCHAR(20), TinhTrang NVARCHAR(20), DonGia FLOAT)
AS
BEGIN
    INSERT INTO @ProductList
    SELECT MaSP, TenSP, LoaiSP, TinhTrang, DonGia
    FROM dbo.SAN_PHAM
    WHERE TenSP LIKE '%' + @string + '%'
    RETURN
END
```

- C#:

```
public DataTable FindProduct(string name)
{
    dtProduct = new DataTable();
    string func_name = "func_findProductByName";
    string queryView = $"SELECT * FROM {func_name} (@name)";
    //SqlConnection con = getConnection.GetSqlConnection();
    con.Open();
    SqlCommand cmd = new SqlCommand(queryView, con);

    cmd.Parameters.AddWithValue("@name", name);

    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    dtProduct.Clear();
    sda.Fill(dtProduct);

    con.Close();
    return dtProduct;
}
```

6. Thống kê doanh thu theo ngày, tháng, năm

- SQL:

```
CREATE FUNCTION [dbo].[func_tinhDoanhThuThang](@thang INT, @nam INT)
RETURNS float
BEGIN
    DECLARE @doanhThu float = 0;
    SELECT @doanhthu = COALESCE(SUM(GiaTriHD), 0)
    FROM HOA_DON
    WHERE MONTH(NgayDatHang) = @thang AND YEAR(NgayDatHang) = @nam;
    RETURN @doanhThu;
END;
```

- C#:

```
public void load_tabPage_Statics()
{
    db.openConnection();
    cmd = new SqlCommand("SELECT dbo.func_tinhDoanhThuThang(@thang, @nam)", db.getConnection());
    cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM")));
    cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy")));
    result = cmd.ExecuteScalar();
    doanhthutheothang = (double)result;
    doanhthuString = doanhthutheothang.ToString();
    labeldoanhthutheothang.Text = "Doanh thu theo tháng: " + doanhthuString + " VND";
    db.closeConnection();
}
```

7. Quản lý nhân viên

7.1. Thêm nhân viên

- SQL:

```
CREATE PROCEDURE [dbo].[proc_AddEmployee]
@manv nvarchar(10), @honv nvarchar(10), @tennv nvarchar(10), @ngaysinh date, @sdt nchar(11),
@macv nvarchar(20), @soca int, @thuong int, @ngaytd date, @htcongviiec nvarchar(20)
AS
    INSERT INTO NHAN_VIEN(MaNV, HoNV, TenNV, NgaySinh, SDT, MaCV, SoCa, Thuong,
    NgayTuyenDung, HTCongViec)
    VALUES(@manv, @honv, @tennv, @ngaysinh, @sdt, @macv, @soca, @thuong, @ngaytd, @htcongviiec)
```

- C#:

```
public bool Add_Employee(EmployeeDTO employee)
{
    try
    {
        SqlCommand cmd = new SqlCommand("EXEC proc_AddEmployee @manv, @honv, @tennv, @ngaysinh, " +
            "@sdt, @macv, @soca, @thuong, @ngaytd, @htcongviiec", db.GetSqlConnection());

        cmd.Parameters.Add("@manv", SqlDbType.NVarChar, 10).Value = employee.MaNV;
        cmd.Parameters.Add("@honv", SqlDbType.NVarChar, 10).Value = employee.HoNV;
        cmd.Parameters.Add("@tennv", SqlDbType.NVarChar, 10).Value = employee.TenNV;
        cmd.Parameters.Add("@ngaysinh", SqlDbType.Date).Value = employee.NgaySinh;
        cmd.Parameters.Add("@sdt", SqlDbType.NChar, 11).Value = employee.SDT;
        cmd.Parameters.Add("@macv", SqlDbType.NVarChar, 20).Value = employee.MaCV;
        cmd.Parameters.Add("@soca", SqlDbType.Int).Value = employee.Soca;
        cmd.Parameters.Add("@thuong", SqlDbType.Int).Value = employee.Thuong;
        cmd.Parameters.Add("@ngaytd", SqlDbType.Date).Value = employee.NgayTuyenDung;
        cmd.Parameters.Add("@htcongviiec", SqlDbType.NVarChar, 20).Value = employee.HTcongviiec;
        db.openConnection();
        Console.WriteLine("YES1");
        try
        {
            Console.WriteLine("YES2");
            cmd.ExecuteNonQuery();
            Console.WriteLine("YES3");
            db.closeConnection();
            Console.WriteLine("YES4");
            return true;
        }
        catch
        {
            Console.WriteLine("NO1");
            db.closeConnection();
            return false;
        }
    }
    catch
    {
        Console.WriteLine("NO2");
        return false;
    }
}
```

7.2. Xóa nhân viên

- SQL:

```
CREATE PROCEDURE [dbo].[proc_DeleteEmployee]
    @MaNV nvarchar(10)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRANSACTION;
    BEGIN TRY
        DELETE FROM BangPhanCong WHERE MaNV = @MaNV;
        DELETE FROM NHAN_VIEN WHERE MaNV = @MaNV;
    END TRY
    BEGIN CATCH
        DECLARE @error NVARCHAR(MAX)
        SELECT @error = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@error, 16, 1)
        ROLLBACK TRANSACTION;
        THROW;
    END CATCH
    COMMIT TRANSACTION;
END
```

- C#:

```
public bool Delete_Employee(EmployeeDTO employee)
{
    SqlCommand cmd = new SqlCommand("EXEC proc_DeleteEmployee @MaNV", db.GetSqlConnection());
    cmd.Parameters.Add("@MaNV", SqlDbType.NVarChar, 10).Value = employee.MaNV;

    db.openConnection();
    if(cmd.ExecuteNonQuery() == -1)
    {
        db.closeConnection();
        return true;
    }
    else
    {
        db.closeConnection();
        return false;
    }
}
```

7.3. Sửa thông tin nhân viên

- SQL:

```
CREATE PROCEDURE [dbo].[proc_EditEmployee]
    @manv nvarchar(10), @honv nvarchar(10), @tennv nvarchar(10), @ngaysinh date, @sdt nchar(11),
    @macv nvarchar(20), @soca int, @thuong int, @ngaytd date, @htcongviiec nvarchar(20)
AS
BEGIN
    BEGIN TRY
        UPDATE NHAN_VIEN
        SET HoNV = @honv, TenNV = @tennv, NgaySinh = @ngaysinh,
            SDT = @sdt, MaCV = @macv, SoCa = @soca, Thuong = @thuong,
            NgayTuyenDung = @ngaytd, HTCongViec = @htcongviiec
        WHERE MaNV = @manv
    END TRY
    BEGIN CATCH
        DECLARE @error NVARCHAR(MAX)
        SELECT @error = N'Lỗi' + ERROR_MESSAGE()
        RAISERROR(@error, 16, 1)
    END CATCH
END
```

- C#:

```
public bool Edit_Employee(EmployeeDTO employee)
{
    try
    {
        SqlCommand cmd = new SqlCommand("EXEC proc_EditEmployee @manv, @honv, @tennv, @ngaysinh, "
            + "@sdt, @macv, @soca, @thuong, @ngaytd, @htcongviiec", db.GetSqlConnection());

        cmd.Parameters.Add("@manv", SqlDbType.NVarChar, 10).Value = employee.MaNV;
        cmd.Parameters.Add("@honv", SqlDbType.NVarChar, 10).Value = employee.HoNV;
        cmd.Parameters.Add("@tennv", SqlDbType.NVarChar, 10).Value = employee.TenNV;
        cmd.Parameters.Add("@ngaysinh", SqlDbType.Date).Value = employee.NgaySinh;
        cmd.Parameters.Add("@sdt", SqlDbType.NChar, 11).Value = employee.SDT;
        cmd.Parameters.Add("@macv", SqlDbType.NVarChar, 20).Value = employee.MaCV;
        cmd.Parameters.Add("@soca", SqlDbType.Int).Value = employee.Soca;
        cmd.Parameters.Add("@thuong", SqlDbType.Int).Value = employee.Thuong;
        cmd.Parameters.Add("@ngaytd", SqlDbType.Date).Value = employee.NgayTuyenDung;
        cmd.Parameters.Add("@htcongviiec", SqlDbType.NVarChar, 20).Value = employee.HTCongViec;
        db.openConnection();
        try
        {
            cmd.ExecuteNonQuery();
            db.closeConnection();
            return true;
        }
        catch
        {
            db.closeConnection();
            return false;
        }
    }
    catch
    {
        return false;
    }
}
```

7.4. Tìm kiếm nhân viên

- SQL:

```
CREATE PROCEDURE [dbo].[proc_SearchByMaNV]
    @MaNV nvarchar(10)
AS
BEGIN
    SELECT *
    FROM V_LoadNhanVien
    WHERE MaNV = @MaNV
END

CREATE PROCEDURE [dbo].[proc_SearchBySDT]
    @SDT nchar(11)
AS
BEGIN
    SELECT *
    FROM V_LoadNhanVien
    WHERE SDT = @SDT
END

CREATE PROCEDURE [dbo].[proc_SearchByTenCV]
    @TenCV nvarchar(50)
AS
BEGIN
    SELECT *
    FROM V_LoadNhanVien
    WHERE TenCV = @TenCV
END

CREATE PROCEDURE [dbo].[proc_SearchByTenNV]
    @TenNV nvarchar(10)
AS
BEGIN
    SELECT *
    FROM V_LoadNhanVien
    WHERE TenNV = @TenNV
END
```

- C#:

```
public DataTable searchEmployee(EmployeeDTO employee, string type)
{
    string cmd;

    if(type == "Mã NV")
    {
        cmd = "EXEC proc_SearchByMaNV " + employee.MaNV;
    }
    else if(type == "SDT")
    {
        cmd = "EXEC proc_SearchBySDT " + employee.SDT;
    }
    else if(type == "Tên CV")
    {
        cmd = "EXEC proc_SearchByTenCV " + employee.MaCV;
    }
    else
    {
        cmd = "EXEC proc_SearchByTenNV " + employee.TenNV;
    }

    db.openConnection();
    SqlDataAdapter adapter = new SqlDataAdapter(cmd, db.GetSqlConnection());
    db.closeConnection();
    DataTable dt = new DataTable();
    adapter.Fill(dt);
    return dt;
}
```

7.5. Tính lương nhân viên

- SQL:

```
CREATE FUNCTION func_TinhLuongNV(@thang int, @nam int) RETURNS TABLE
AS RETURN (
    SELECT nv.MaNV, nv.HoNV, nv.TenNV, nv.SoCa,
        CASE nv.HTCongViec WHEN 'Full-time' THEN 15000000
        ELSE (60000 * nv.SoCa) *
        CASE WHEN dbo.func_TinhDoanhThuThang(@thang, @nam) > 50000000 THEN 1.15
        ELSE 1
        END AS Luong
    FROM NHAN_VIEN nv
    INNER JOIN CONG_VIEC cv ON nv.MaCV = cv.MaCV
)
```

- C#:

```
public DataTable CalWageEmployee()
{
    getConnection db = new getConnection();
    db.openConnection();

    SqlCommand cmd = new SqlCommand("SELECT * FROM func_TinhLuongNV(@thang, @nam)", db.GetSqlConnection());
    cmd.Parameters.AddWithValue("@thang", int.Parse(DateTime.Today.ToString("MM")));
    cmd.Parameters.AddWithValue("@nam", int.Parse(DateTime.Today.ToString("yyyy")));

    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    adapter.Fill(dt);

    db.closeConnection();
    return dt;
}
```

8. Quản lý đặt bàn

8.1. Cập nhật trạng thái bàn

- SQL:

```
CREATE PROCEDURE proc_UpdateTrangThaiBan
    @MaBan NVARCHAR(10),
    @TrangThai NVARCHAR(20)
AS
BEGIN
    UPDATE BAN
    SET TrangThai = @TrangThai
    WHERE MaBan = @MaBan;
END;
```

- C#:

```
public bool UpdateTrangThaiBan(string maBan, string trangThai)
{
    string queryInsert = "proc_UpdateTrangThaiBan @MaBan, @TrangThai";
    SqlConnection con = getConnection.GetSqlConnection();
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(queryInsert, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaBan", maBan),
            new SqlParameter("@TrangThai", trangThai),
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        throw(ex);
    }
    finally
    {
        con.Close();
    }
    return false;
}
```


9. Quản lý phân ca

9.1. Thêm ca cho nhân viên

- SQL:

```
CREATE PROCEDURE proc_ThemCaLam
    @MaCa nvarchar(20),
    @NgayLam date,
    @GioBatDau nchar(10),
    @GioKetThuc nchar(10)
AS
BEGIN
    INSERT INTO CA_LAM (MaCa, NgayLam, GioBatDau, GioKetThuc)
    VALUES (@MaCa, @NgayLam, @GioBatDau, @GioKetThuc)
END
```

- C#:

```
public bool createShift(ShiftDTO shiftDTO)
{
    string queryInsert = "proc_ThemCaLam @MaCa, @NgayLam, @BatDau, @KetThuc";
    SqlConnection con = getConnection.GetSqlConnection();
    try
    {
        MemoryStream ms = new MemoryStream();
        con.Open();
        SqlCommand cmd = new SqlCommand(queryInsert, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaCa", shiftDTO.Id),
            new SqlParameter("@NgayLam", shiftDTO.WorkingDate.ToString("yyyy-MM-dd")),
            new SqlParameter("@BatDau", shiftDTO.StartTime),
            new SqlParameter("@KetThuc", shiftDTO.EndTime)
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```

9.2. Xóa ca làm của nhân viên

- SQL:

```
CREATE PROCEDURE proc_DeletePhanCong
    @MaNV nvarchar(10),
    @MaCa nvarchar(20)
AS
BEGIN
    DELETE FROM BangPhanCong WHERE MaNV = @MaNV AND MaCa = @MaCa
END
```

- C#:

```
public bool deleteAssignment(AssignmentDTO assignmentDTO)
{
    string queryInsert = "proc_DeletePhanCong @MaCa, @MaNV";
    SqlConnection con = getConnection.GetSqlConnection();
    try
    {
        MemoryStream ms = new MemoryStream();
        con.Open();
        SqlCommand cmd = new SqlCommand(queryInsert, con);
        cmd.Parameters.AddRange(new[]
        {
            new SqlParameter("@MaCa", assignmentDTO.ShiftId),
            new SqlParameter("@MaNV", assignmentDTO.EmployeeId)
        });
        if (cmd.ExecuteNonQuery() > 0)
            return true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return false;
}
```

CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN

Ứng với hai nhóm người dung (Nhân viên và quản lý) là các role riêng:

+ Đối với Role Staff (dành cho nhân viên):

--

+ Đối với Role Manager, ta sử dụng server role sysadmin.

- Trigger tạo tài khoản

- Procedure xóa nhân viên

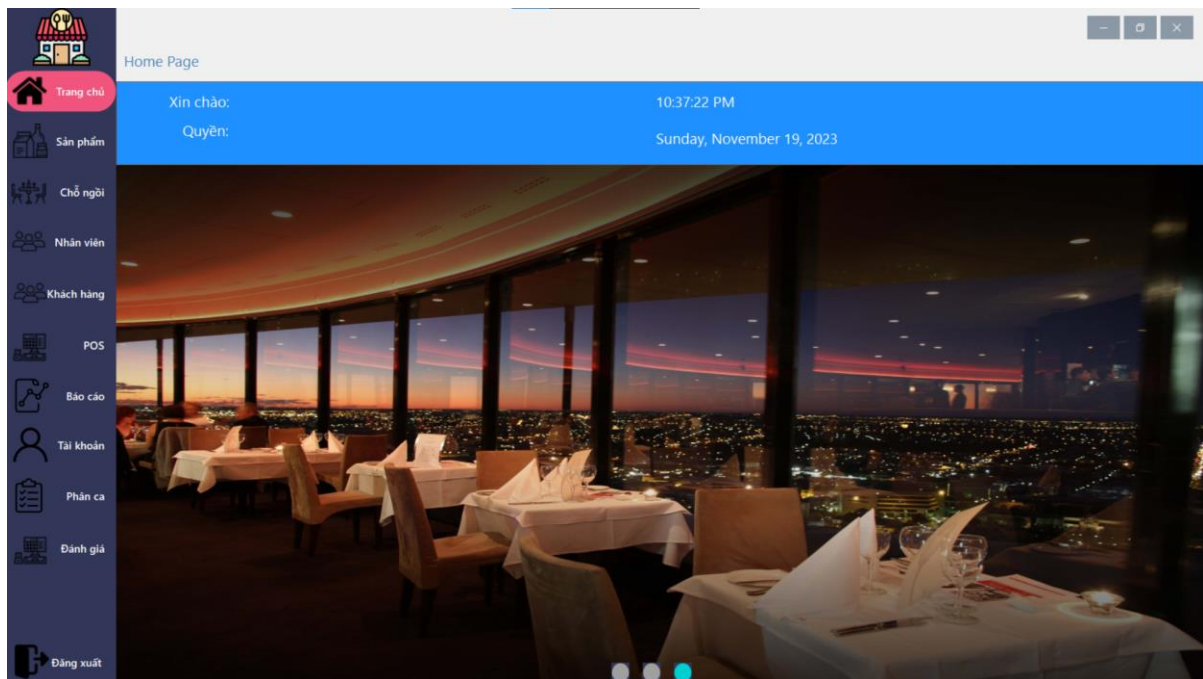
CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG

Môi trường cài đặt và công nghệ sử dụng

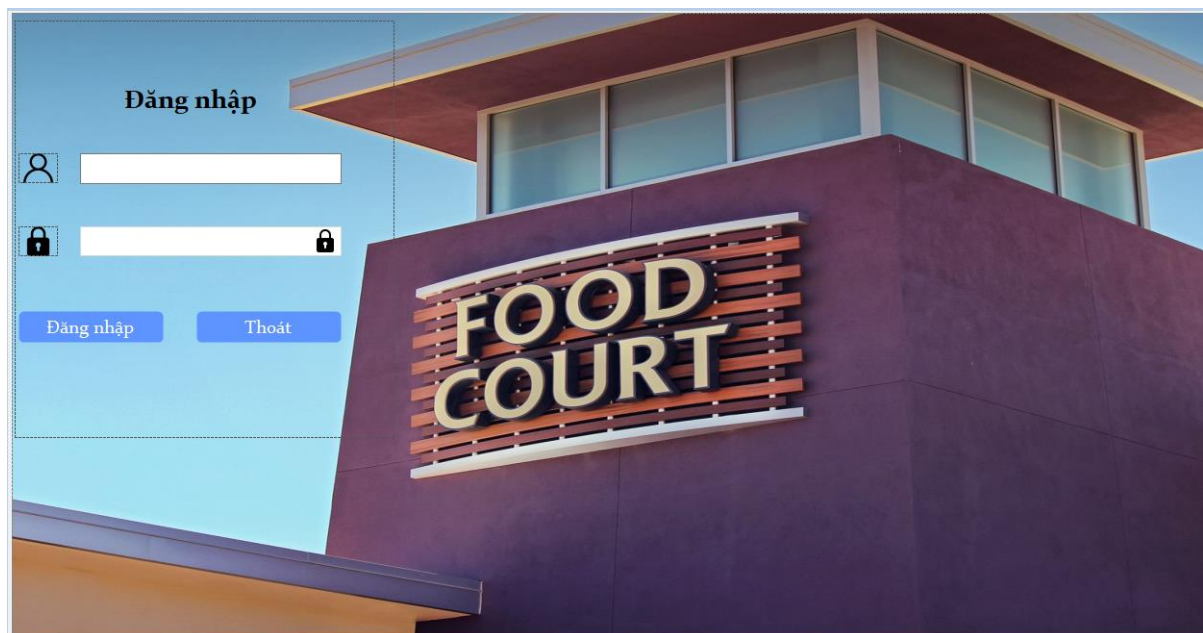
- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.
- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.

Giao diện và chức năng chương trình:

1. Form Main:



2. Form đăng nhập:



3. Các form danh mục:


3.1. Danh mục nhân viên:

Mã nhân viên	Họ nhân viên	Tên nhân viên	Ngày Sinh	SSN	Mã công việc	Số cơ	Thưởng	Ngày tuyển dụng	Hình thức	Tên công việc	
nv01	A	B	10-May-02	0000000000	cv1	1	1	10-May-23	Bán thời gian	Dầu bắp	
nv02	C	D	10-May-00	1111111111	cv2	2	2	10-May-23	Bán thời gian	Phục vụ	
nv03	D	B	01-Apr-01	2222222222	cv2	2	2	01-May-23	Bán thời gian	Phục vụ	







3.2. Danh mục bảng phân ca:

3.3. Danh mục ca làm:

3.4. Danh mục hóa đơn:





Danh Sách Hóa Đơn


Mã hóa đơn	Mã khách hàng	Trạng thái	Tổng hóa đơn	Mã phục vụ	Mã bán	PTTT	Ngày	Giờ		
1	cus01	Chưa thanh toán	260000	nv02	12	Tiền mặt	19-Nov-23	14:13:00		
2	cus02	Chưa thanh toán	580000	nv03	5	Tiền mặt	19-Nov-23	18:55:00		


Save


3.5. Danh mục sản phẩm:


 Trang chủ


 Sản phẩm


 Chỗ ngồi


 Nhân viên


 Khách hàng


 POS

 Bảo cáo


 Tài khoản

 Phân ca

 Đánh giá

 Đăng xuất

Danh sách sản phẩm



Tìm kiếm

Mã sản phẩm	Tên sản phẩm	Danh Mục	Tình trạng	Đơn giá
sp0110	ASCO	Món Test	Còn	8999
sp0000	ABC	Món Tráng Miếng	Hết	90000
sp01	Bánh Mì	Món Ăn Kiem	Còn	10000
sp010	Đầm bông súng bò	Món Ăn Chính	Còn	50000
sp011	Rượu JIM BEAM	Nước Uống	Còn	1000000
sp02	Pizza	Món Chính	Còn	100000
sp020	Beefsteak	Món Chính	Còn	120000
sp03	Wellington	Món Chính	Còn	100000
sp04	Cà s	Món Chính	Còn	70000
sp05	Gà nướng	Món Chính	Còn	300000
sp06	Bánh dứa	Món Tráng Miếng	Còn	30000
sp07	Nước Cam	Nước Uống	Còn	30000

3.6. Danh mục bàn đặt:

	Mã Bàn	Tạng thái bàn	Tiền đặt cọc	Khách hàng (đã đặt)		
	1	Đang dùng	NULL	cus01		
	10	Đang dùng	NULL	cus01		
	12	Còn trống	NULL	NULL		
	13	Đang dùng	NULL	cus02		
	14	Đã đặt	500	cus100		
	2	Đã đặt	NULL	cus100		
	3	Đã đặt	15,000	cus01		
	4	Còn trống	NULL	NULL		
	5	Đã đặt	50	cus02		
	6	Đang dùng	NULL	cus07		
	7	Đang dùng	NULL	NULL		
	8	Đã đặt	NULL	cus100		
	9	Đã đặt	50	cus02		

3.7. Danh mục khách hàng:

	Mã khách hàng	Tên khách hàng	Số điện thoại	Điểm thưởng	Chức năng
	cus01	A	222222222	0	
	cus000000	aaaaa	666666666	0	
	cus02	abc	0912380725	0	
	cus07	aod	0943280124	0	
	cus100	ABC	4444444444	0	
	cus1000	hh	555555555	0	