## TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN

**BỘ MÔN HỆ THỐNG THÔNG TIN**

🙞🙞🙞🙞🙞

A blue logo with a ball and a black background

Description automatically generated

## ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL

**TÊN ĐỀ TÀI:**

**QUẢN LÝ BÁN HÀNG CỦA CỬA HÀNG TỰ CHỌN**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Lớp HP:** *DBMS*

**Nhóm thực hiện:** *Nhóm 12*

### Học kỳ: *1*

**Năm học:** *2024 - 2025*

*Thành phố Hồ Chí Minh, tháng , năm 2024*

## DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN

**HỌC KÌ 1 NĂM HỌC 2024-2025**

### Nhóm 12

***Đề tài: Quản lý bán hàng của cửa hàng tự chọn***

|  |  |
| --- | --- |
| 22110460 | Trương Quốc Vương |
| 22110423 | Huỳnh Quốc Thắng |
| 22110351 | Hồng Anh Khoa |
| 22110434 | Nguyễn Đức Tín |

## NHẬN XÉT CỦA GV

Tp. Hồ Chí Minh, ngày …., tháng 11, năm 2022

Giảng viên chấm điểm

## MỤC LỤC

## LỜI CẢM ƠN

Kính thưa thầy Nguyễn Thành Sơn,

Nhóm em xin gửi lời cảm ơn chân thành đến thầy vì đã hướng dẫn và giúp đỡ chúng em trong quá trình hoàn thiện đồ án môn Hệ quản trị cơ sở dữ liệu.

Nhờ vào sự chỉ dẫn và giúp đỡ của thầy, nhóm em đã có thể hoàn thành đồ án một cách hiệu quả và đạt được kết quả như mong đợi. Chúng em rất biết ơn thầy đã dành thời gian và tâm huyết để truyền đạt kiến thức và kinh nghiệm cho nhóm.

Thầy đã giúp chúng em có được những kiến thức và kỹ năng cần thiết để làm việc với cơ sở dữ liệu một cách hiệu quả, và nhóm em sẽ sử dụng những kiến thức này để phát triển bản thân và đóng góp vào công việc sau này.

Một lần nữa, nhóm em xin chân thành cảm ơn thầy Nguyễn Thành Sơn vì sự giúp đỡ và hướng dẫn của thầy trong quá trình học tập của nhóm.

Trân trọng

## CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

### Đặc tả đề tài

* 1. **. Mô tả bài toán**

Một cửa hàng tự chọn cần có một hệ thống để quản lý bán hàng. Chương trình quản lý được xây dựng cho hai đối tượng người dùng là quản lý và nhân viên của cửa hàng tự chọn với các quyền khác nhau. Đối với hệ thống bán hàng, cửa hàng tự chọn cần lưu trữ những dữ liệu sau:

Cửa hàng gồm nhiều nhân viên, mỗi nhân viên có các thông tin như mã nhân viên, họ tên, giới tính, ngày sinh, số điện thoại, địa chỉ, ảnh đại diện, ngày tuyển dụng, mật khẩu, chức vụ.

Một công việc có thể có một hay nhiều nhân viên cùng làm, mỗi nhân viên chỉ làm một công việc, mỗi công việc có mã công việc và tên công việc. Các nhân viên sẽ được chia ca để làm việc.

Mỗi ca làm việc gồm các thông tin như mã ca làm việc, thời gian bắt đầu, thời gian kết thúc. Mỗi nhân viên có thể làm nhiều ca, một ca có thể có nhiều nhân viên.

Cửa hàng bán nhiều sản phẩm, một sản phẩm chỉ thuộc một nhóm sản phẩm, mỗi nhóm sản phẩm có một hoặc nhiều sản phẩm. Mỗi sản phẩm gồm các thông tin như mã sản phẩm, tên sản phẩm, hình ảnh, nhà sản xuất, giá bán, số lượng tồn kho và số lượng sản phẩm đã bán. Các sản phẩm phân biệt với nhau qua mã sản phẩm.

Mỗi nhóm sản phẩm có thể bao gồm nhiều sản phẩm, một sản phẩm chỉ thuộc 1 nhóm sản phẩm. Nhóm sản phẩm chứa thông tin về: mã nhóm sản phẩm và tên nhóm sản phẩm.

Mỗi hóa đơn chỉ được xuất bởi một nhân viên và một nhân viên có thể xuất nhiều hóa đơn hoặc không xuất hóa đơn nào. Mỗi hoá đơn có nhiều sản phẩm hoặc ít nhất một sản phẩm, một sản phẩm có thể nằm trong nhiều hóa đơn hoặc không nằm trong hóa đơn nào. Hóa đơn bao gồm: mã hóa đơn, tên khách hàng, ngày xuất hóa đơn, tổng tiền, số tiền khách đã đưa, số tiền trả lại, các thông tin về hóa đơn được in ra như mã hóa đơn, mã sản phẩm, tên sản phẩm, số lượng, đơn giá, thành tiền. Một bản sao của hóa đơn bán hàng (phiếu xuất) sẽ được lưu lại để tiện cho quá trình quản lý, còn một bản giao cho khách hàng.

Mỗi đơn nhập chứa nhiều sản phẩm và có ít nhất một sản phẩm; một sản phẩm có thể nằm trong nhiều hoặc không nằm trong đơn nhập nào. Đơn nhập bao gồm: mã đơn nhập, ngày nhập, Các thông tin về đơn nhập gồm có mã sản phẩm, mã đơn nhập, số lượng nhập, giá nhập sản phẩm. Đơn nhập cũng có thể được lưu lại để tiện quản lý.

* 1. **Mô tả chức năng bài toán.**

**Quản lý bán hàn**g: cho phép chọn sản phẩm, tạo đơn hàng cho khách, thêm sản phẩm, xóa sản phẩm, xuất hóa đơn thanh toán.

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

**Quản lý kho:** cho phép biết được số lượng của từng nguyên liệu còn trong kho hàng, ngày nhập hàng của nguyên liệu, hạn sử dụng.

**Quản lý nhân viên:** cho phép 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.

**Quản lý ca làm việc của nhân viên:** Người quản lý có quyền thêm, xóa, sửa ca làm việc của các nhân viên.

**Quản lý doanh thu:** cho phép tính toán doanh thu theo ngày, tuần, tháng hoặc năm, dựa trên thông tin trên các hóa đơn đã lập.

**Chức năng Quản lý Hóa đơn:** Tạo hóa đơn, lưu trữ hóa đơn, tra cứu hóa đơn, in hóa đơn, báo cáo hóa đơn, báo cáo số lượng hóa đơn đã tạo trong một khoảng thời gian cụ thể, phân tích doanh thu theo hóa đơn, giúp quản lý cửa hàng nắm rõ tình hình kinh doanh.

* 1. **Mô tả màn hình hệ thống.**

Cuối cùng là mô tả về các màn hình hệ thống. Hệ thống bao gồm 5 màn hình:

Đầu tiên là *màn hình đăng nhập*. Trên màn hình sẽ có tên cửa hàng, bên dưới nó là một khung để đăng nhập gồm:

Tài khoản: Ô trống để người dùng nhập mã nhân viên.

Mật khẩu: Ô trống để người dùng nhập mật khẩu.

Nút "Đăng nhập": Nút để xác thực thông tin đăng nhập và truy cập vào ứng dụng.

Nút đóng: Một dấu "x" nhỏ ở góc trên bên phải cửa sổ đăng nhập, cho phép người dùng đóng cửa sổ này.

Sau khi đăng nhập thành công sẽ hiển thị **màn hình chính** bao gồm:

*Phần trên:*

Thanh tìm kiếm: Cho phép người dùng tìm kiếm sản phẩm bằng từ khóa.

Tên người dùng.

Phần giữa gồm:

*Bên trái:*

Danh sách sản phẩm: Hiển thị các sản phẩm được chia theo danh mục. Mỗi sản phẩm bao gồm:

Hình ảnh sản phẩm: Giúp người dùng hình dung về sản phẩm.

Tên sản phẩm: Cho biết tên và thông tin cơ bản về sản phẩm.

Giá sản phẩm: Hiển thị giá tiền của mỗi sản phẩm.

Nút "Thêm": Cho phép người dùng thêm sản phẩm vào giỏ hàng.

Nhãn "Đã hết hàng": Cho biết sản phẩm đã hết hàng.

Nhãn "Mới": Đánh dấu sản phẩm mới.

*Bên phải:*

Thông tin giỏ hàng: Hiển thị danh sách các sản phẩm đã thêm vào giỏ hàng, bao gồm: tên sản phẩm, số lượng, giá, nút tăng giảm số lượng, nút xóa sản phẩm.

*Phần dưới:*

Thanh thêm tên khách hàng: nhập tên khách hàng.

Thông tin thanh toán:

Thành tiền: Hiển thị tổng số tiền cần thanh toán.

Nút "Hủy": Hủy bỏ đơn hàng.

Nút "Thanh toán": Hoàn tất đơn hàng và tiến hành thanh toán.

*Các nút khác:*

* Nút sắp xếp sản phẩm: Có thể sắp xếp sản phẩm theo các tiêu chí như giá cả, tên sản phẩm, ...
* Nút lọc sản phẩm: Cho phép người dùng lọc sản phẩm theo danh mục, giá cả, ...

Ngoài ra còn có **màn hình quản lý kho** gồm:

*Thanh tiêu đề:*

Tên cửa hàng.

Tên màn hình: "Danh sách sản phẩm".

Các nút chức năng: chọn, thêm, xóa, sửa thông tin sản phẩm (chỉ người quản lý mới có quyền thực hiện)

Tên người dùng đang đăng nhập.

*Thanh công cụ:*

Tìm kiếm: Cho phép người dùng tìm kiếm sản phẩm theo tên, mã hoặc các tiêu chí khác.

Lọc: Cho phép lọc danh sách sản phẩm theo các tiêu chí như loại sản phẩm, nhà sản xuất, ngày nhập...

*Bảng danh sách sản phẩm:*

Các cột:

Mã sản phẩm: Mã định danh duy nhất của từng loại sản phẩm.

Tên sản phẩm: Tên gọi của loại sản phẩm.

Nhóm sản phẩm: Phân loại sản phẩm (ví dụ: thực phẩm, đồ gia dụng, gia vị, mỹ phẩm,…)

Nhà sản xuất: Tên của nhà sản xuất

Giá nhập: Giá nhập của một đơn vị sản phẩm.

Giá bán: Giá bán của một đơn vị sản phẩm.

Tồn kho: Số lượng sản phẩm hiện có trong kho.

Ngày nhập sản phẩm: Ngày nhập sản phẩm vào kho.

Số lượng đã bán: Số lượng sản phẩm đã bán.

Các dòng: Mỗi dòng đại diện cho một loại sản phẩm cụ thể, hiển thị thông tin chi tiết của sản phẩm đó.

Thanh cuốn: Cho phép người dùng di chuyển theo chiều dọc của danh sách sản phẩm khi danh sách quá dài.

Bên cạnh đó sẽ có một **màn hình được dùng để quản lý các hóa đơn** của cửa hàng, bao gồm các đặc điểm sau:

Các thành phần chính trên màn hình:

Thanh tìm kiếm: Cho phép người dùng tìm kiếm các hóa đơn dựa trên nhiều tiêu chí như mã hóa đơn, tên hàng.

Mỗi hàng trong bảng đại diện cho một hóa đơn, bao gồm các thông tin chi tiết như:

Mã hóa đơn: Mã số duy nhất của mỗi hóa đơn.

Ngày xuất: Thời điểm tạo hóa đơn.

Tên khách hàng: Tên của khách mua hàng.

Tổng tiền hàng: Tổng số tiền hàng hóa.

Các nút chức năng:

Xuất hóa đơn.

Các tùy chọn lọc: Cho phép lọc dữ liệu theo thời gian.

Hai nút mũi tên: Cho phép người dùng di chuyển giữa các trang của danh sách hóa đơn khi danh sách quá dài.

Cuối cùng là **màn hình thông tin cá nhân của người đang đăng nhập**, bao gồm những đặc điểm sau:

*Phần trên:*

Bên trái có ảnh đại diện, tên người dùng.

Bên phải có các thông tin cá nhân gồm: mã nhân viên, họ tên nhân viên, giới tính, ngày sinh, số điện thoại, địa chỉ, tên công việc, ngày tuyển dụng.

Phần dưới:

Bảng ca làm việc gồm các ngày làm việc (từ thứ hai đến chủ nhật) ở một thanh ngang trên cùng, bên dưới là thời gian làm việc vào mỗi ngày, nếu không có ca sẽ để trống ngày đó.

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆTHỐNG

##### 2.1. Thiết kế cơ sở dữ liệu mức quan niệm

**Tập các thực thể:**

A diagram of different languages

Description automatically generated

A diagram of a diagram

Description automatically generated

A diagram of a company

Description automatically generated with medium confidence**A diagram of a group of circles with text

Description automatically generated with medium confidence**

##### A black and white rectangle with black text Description automatically generated

##### A diagram of a network Description automatically generated

##### Mối quan hệ của tập thực thể:

##### A diagram of a flowchart Description automatically generated

##### 2.2. Thiết kế cơ sở dữ liệu mức logic

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

* NHANVIEN(MaNV, HoTen, GioiTinh, NgaySinh, SĐT, ĐChi, AnhDaiDien, MKhau, NgTuyenDung, ChucVu)
* LAMVIEC(MaNV, MaCViec)
* CONGVIEC(MaCongViec, TenCongViec)
* THAMGIA(MaCa, MaNV)
* CALAMVIEC(MaCaLamViec, ThGianBDau, ThGianKThuc)
* SANPHAM(MaSPham, TenSPham, HinhAnh, NhaSanXuat, GiaBan, TonKho, DaBan)
* PHANLOAI(MaSanPham, MaNhom)
* NHOMSANPHAM(MaNhom, TenNhom)
* HOADON(MaHoaDon, TenKhachHang, NgayXuat, TongTien, TienKhachDua, TienThoi)
* XUATHOADON(MaHoaDon, MaNV)
* THONGTINHOADON(MaHoaDon, MaSPham, SoLuong, DonGia, ThanhTien)
* DONNHAP(MaDonNhap, NgNhap)
* CHITIETDONNHAP(MaDonNhap, MaSanPham, SoLuongNhap, GiaNhap)

**2.3. Thiết kế cơ sở dữ liệu mức vật lý**

**2.4. Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | NHANVIEN | Khóa chính MaNV,  NOT NULL HoTen,  NOT NULL DiaChi,  NOT NULL GioiTinh (check IN (‘Nam’ , ‘Nu’)),  Khóa ngoại MaNQL tham chiếu đến MaNV,  NOT NULL MKhau,  NOT NULL SDT  NOT NULL NgaySinh,  NOT NULL AnhDaiDien |
| 2 | LAMVIEC | Khóa chính (MaCViec, MaNV)  Khóa ngoại MaCViec tham chiếu đến CONGVIEC(MaCongViec)  Khóa ngoại MaNV tham chiếu đến NHANVIEN(MaNV) |
| 3 | CONGVIEC | Khoá chính MaCongViec |
| 4 | THAMGIA | Khóa chính (MaCa, MaNV)  Khóa ngoại MaCa tham chiếu đến CALAMVIEC(MaCaLamViec)  Khóa ngoại MaNV tham chiếu đến NHANVIEN(MaNV) |
| 5 | CALAMVIEC | Khoá chính MaCaLamViec,  ThGianBDau NOT NULL  ThGianKThuc (check > ThGianBDau) |
| 6 | SANPHAM | Khoá chính MaSPham,  Khoá ngoại MaNhom  tham chiếu đến NHOMSANPHAM(MaNhom),  NOT NULL TenSPham,  NOT NULL NhaSanXuat,  NOT NULL HinhAnh  GiaNhap (check > 0) NOT NULL  GiaBan (check > 0) NOT NULL  TonKho (check > 0) NOT NULL  DaBan (check > 0) NOT NULL |
| 7 | NHOMSANPHAM | Khóa chính MaNhom,  NOT NULL TenNhom |
| 8 | PHANLOAI | Khóa chính MaSPham  NOT NULL MaNhom  Khóa ngoại MaSPham tham chiếu đến SANPHAM(MaSPham)  Khoá ngoại MaNhom tham chiếu đến NHOMSANPHAM(MaNhom) |
| 9 | HOADON | Khoá chính MaHoaDon,  NgayXuat (check > 0), NOT NULL    TongTien (check > 0), NOT NULL    TienKhachDua (check>0), NOT NULL  TienThoi (check >= 0), NOT NULL |
| 10 | XUATHOADON | Khoá chính (MaHoaDon)  NOT NULL MaNV  Khoá ngoại MaNV tham chiếu đến NHANVIEN(MaNV)  Khóa ngoại MaHoaDon tham chiếu đến HOADON(MaHoaDon) |
| 11 | THONGTINHOADON | Khoá chính (MaHoaDon, MaSPham)  Khoá ngoại MaHoaDon tham chiếu đến HOADON(MaHoaDon)  Khóa ngoại MaSPham tham chiếu đến SANPHAM(MaSPham)  NOT NULL TenSPham,  DonGia (check > 0), NOT NULL  ThanhTien (check > 0), NOT NULL  SoLuong (check > 0), NOT NULL |
| 12 | DONNHAP | Khoá chính (MaDonNhap)  NgNhap NOT NULL |
| 13 | CHITIETDONNHAP | Khóa chính (MaDonNhap, MaSPham)  Khóa ngoại (MaDonNhap) tham chiếu đến DONNHAP(MaDonNhap),  Khóa ngoại (MaSPham) tham chiếu đến SANPHAM(MaSPham)  NOT NULL SoLuongNhap, GiaNhap  SoLuongNhap (check > 0) NOT NULL  GiaNhap (check > 0) NOT NULL |

##### 2.5. Code tạo cơ sở dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Bảng** | **Code** |
| 1 | NHANVIEN | CREATE TABLE NHANVIEN(  MaNV nvarchar(10) PRIMARY KEY,  HoTen nvarchar(20) NOT null,  GioiTinh nvarchar(5) NOT null check(GioiTinh IN ('Nam' , 'Nu')),  NgaySinh date NOT null,  SDT nvarchar(20) NOT null,  DChi nvarchar(20) NOT null,  AnhDaiDien nvarchar(100) NOT null,  MKhau nvarchar(50) NOT null,  NgTuyenDung date NOT null,  ChucVu nvarchar(20) NOT null  ) |
| 2 | CONGVIEC | CREATE TABLE CONGVIEC(  MaCViec nvarchar(10) PRIMARY KEY,  TenCViec nvarchar(20) NOT NULL  ) |
| 3 | LAMVIEC | CREATE TABLE LAMVIEC(  MaNV nvarchar(10) PRIMARY KEY,  MaCViec nvarchar(10) NOT null,  FOREIGN KEY (MaNV) references NhanVien(MaNV),  FOREIGN KEY (MaCViec) references CongViec(MaCviec)  ) |
| 4 | CALAMVIEC | CREATE TABLE CALAMVIEC(  MaCaLamViec nvarchar(10) PRIMARY KEY,  ThGianBDau Time not null,  ThGianKThuc Time not null  ) |
| 5 | THAMGIA | CREATE TABLE THAMGIA(  MaNV nvarchar(10),  MaCa nvarchar(10),  PRIMARY KEY (MaNV, MaCa),  FOREIGN KEY (MaNV) references NhanVien(MaNV),  FOREIGN KEY (MaCa) references CaLamViec(MaCaLamViec)  ) |
| 6 | SANPHAM | CREATE TABLE SANPHAM(  MaSPham nvarchar(10) PRIMARY KEY,  TenSPham nvarchar(50) not null,  HinhAnh nvarchar(50) NOT null,  NhaSanXuat nvarchar(50) not null,  GiaBan int not null CHECK(GiaBan > 0),  TonKho int not null CHECK(TonKho > 0),  DaBan int not null CHECK(DaBan > 0)  ) |
| 7 | NHOMSANPHAM | CREATE TABLE NHOMSANPHAM(  MaNhom NVARCHAR(10) PRIMARY KEY,  TenNhom NVARCHAR(20) NOT NULL,  ) |
| 8 | PHANLOAI | CREATE TABLE PHANLOAI(  MaSPham NVARCHAR(10) PRIMARY KEY,  MaNhom NVARCHAR(10) NOT NULL,  FOREIGN KEY (MaSPham) REFERENCES SANPHAM(MaSPham),  FOREIGN KEY (MaNhom) REFERENCES NHOMSANPHAM(MaNhom)  ) |
| 9 | HOADON | CREATE TABLE HOADON(  MaHoaDon NVARCHAR(10) PRIMARY KEY,  TenKhachHang NVARCHAR(30) NULL,  NgayXuat DATE NOT NULL,  TongTien INT NOT NULL CHECK(TongTien > 0),  TienKhachDua INT NOT NULL CHECK(TienKhachDua > 0),  TienThoi INT NOT NULL CHECK(TienThoi >= 0)  ) |
| 10 | XUATHOADON | CREATE TABLE XUATHOADON(  MaHoaDon NVARCHAR(10) PRIMARY KEY,  MaNV NVARCHAR(10) NOT NULL,  FOREIGN KEY (MaNV) REFERENCES NHANVIEN(MaNV),  FOREIGN KEY (MaHoaDon) REFERENCES HOADON(MaHoaDon)    ) |
| 11 | THONGTINHOADON | CREATE TABLE THONGTINHOADON(  MaHoaDon NVARCHAR(10),  MaSPham NVARCHAR(10),  SoLuong int not null CHECK(SoLuong > 0),  DonGia int not null CHECK(DonGia > 0),  ThanhTien int not null CHECK(ThanhTien > 0)  PRIMARY KEY (MaHoaDon, MaSPham),  FOREIGN KEY (MaHoaDon) REFERENCES HOADON(MaHoaDon),  FOREIGN KEY (MaSPham) REFERENCES SANPHAM(MaSPham)  ) |
| 12 | DONNHAP | CREATE TABLE DONNHAP(  MaDonNhap NVARCHAR(10) PRIMARY KEY,  NgNhap DATE NOT NULL,  ) |
| 13 | CHITIETDONNHAP | CREATE TABLE CHITIETDONNHAP(  MaDonNhap NVARCHAR(10),  MaSPham NVARCHAR(10),  SoLuongNhap int not null CHECK(SoLuongNhap > 0),  GiaNhap int not null CHECK(GiaNhap > 0)  PRIMARY KEY (MaDonNhap, MaSPham),  FOREIGN KEY (MaDonNhap) REFERENCES DONNHAP(MaDonNhap),  FOREIGN KEY (MaSPham) REFERENCES SANPHAM(MaSPham)  ) |

***A computer screen shot of a computer flow chart

Description automatically generated*2.6. Lược đồ vật lý**

**2.6. Các trigger**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Trigger** | **Code** |
| 1 | Trigger đảm bảo rằng sản phẩm vẫn còn đủ số lượng trước khi tạo đơn bán. | CREATE TRIGGER trg\_CheckProductIsEnough  ON THONGTINHOADON  INSTEAD OF INSERT  AS  BEGIN      DECLARE @MaSPham NVARCHAR(50),                         @SoLuong int,      SELECT @MaSPham = MaSPham FROM Inserted                     @SoLuong = SoLuong FROM Inserted;      IF (Select TonKho From SanPham where MaSPham = @MaSpham ) >= @SoLuong      BEGIN          -- Nếu còn đủ hàng, chèn bản ghi vào bảng THONGTINHOADON          INSERT INTO THONGTINHOADON          SELECT \* FROM Inserted;      END      ELSE      BEGIN          -- Nếu sản phẩm không đủ, báo lỗi          RAISERROR('Số lượng sản phẩm không đủ.', 16, 1);      END  END;  go |
| 2 | Khi một hóa đơn được tạo, trigger cập nhật số lượng đã bán và tồn kho trong bảng “sản phẩm” | CREATE TRIGGER trg\_UpdateProductsAfterInsertBill  ON THONGTINHOADON  AFTER INSERT  AS  BEGIN      -- Cập nhật số lượng đã bán và tồn kho trong bảng SANPHAM      UPDATE SANPHAM      SET          SoLuongDaBan = SoLuongDaBan + i.SoLuong,          TonKho = TonKho - i.SoLuong      FROM          SANPHAM sp          INNER JOIN inserted i ON sp.MaSPham = i.MaSPham  END |
| 3 | Khi một đơn nhập được tạo, cập nhật tồn kho trong bảng “sản phẩm” | CREATE TRIGGER TR\_UpdateTonKho  ON DONNHAP  AFTER INSERT  AS  BEGIN      -- Cập nhật số lượng tồn kho cho sản phẩm      UPDATE SANPHAM      SET TonKho = TonKho + i.SoLuongNhap      FROM SANPHAM sp      INNER JOIN inserted i      ON sp.MaSPham = i.MaSPham  END |
| 4 | Trigger gửi thông báo khi tồn kho dưới mức cho phép | CREATE TRIGGER trg\_SanPhamSapHet  ON THONGTINHOADON  AFTER OF INSERT  AS  Declare @MaSanPham = MaSanPham From Inserted  BEGIN      IF (Select TonKho From SanPham where MaSPham = @MaSpham )  <= 20      BEGIN                  Print ‘San pham sap het!!!‘      END      END; |
| 5 | Trigger tự động cập nhật tổng tiền trên hóa đơn | CREATE TRIGGER trg\_AutoUpdateTongTien  ON THONGTINHOADON  AFTER INSERT, UPDATE, DELETE  AS  BEGIN      UPDATE HOADON      SET TongTien = (          SELECT SUM(ThanhTien)          FROM THONGTINHOADON tthd          WHERE tthd.MaHoaDon = HOADON.MaHoaDon      )      FROM HOADON hd      JOIN INSERTED i ON hd.MaHoaDon = i.MaHoaDon;  END; |

**2.7 Các view**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên VIEW** | **Code** |
| 1 | View hiển thị tất cả sản phẩm trong cửa hàng cùng thông tin như tên sản phẩm, giá, số lượng tồn kho,... | CREATE VIEW v\_ProductList  AS  SELECT  MaSPham, TenSPham, HinhAnh, NhaSanXuat, GiaBan, MaNhom, TonKho, DaBan  FROM SANPHAM |
| 2 | View hiển thị thông tin chi tiết về từng hóa đơn, bao gồm các sản phẩm đã mua và số lượng. | CREATE VIEW v\_infoHOADON  AS  SELECT  hd.MaHoaDon, hd.NgayXuat, hd.TongTien, hd.TenKhachHang, tt.MaSPham, tthd.SoLuong, tthd.ThanhTien  FROM HOADON hd  Inner join THONGTINHOADON tthd  ON  hd.MaHoaDon = tthd.MaHoaDon |
| 3 | View hiển thị tình trạng tồn kho của từng sản phẩm, giúp quản lý kho dễ hơn. | Create View v\_ProductStatus  AS  Select  sp.MaSPham, pl.MaNhom, TenSPham, Ton Kho, DaBan  From SANPHAM sp inner join PHANLOAI pl  On sp.MaSPham = pl.MaSPham |
| 4 | View hiển thị thông tin về nhân viên đã thực hiện giao dịch , bao gồm số hóa đơn mà họ đã xử lí | Create View v\_NhanVienHoaDon  AS  Select \* From XUATHOADON xhd inner join HOADON hd on xhd.MaHoaDon = hd.MaHoaDon |
| 5 | View hiển thị thông tin về các ca làm việc của từng nhân viên, thời gian bắt đầu và kết thúc ca làm việc. | Create View v\_CaLamViecNV  AS  Select \*From THAMGIA tg inner join CALAMVIEC clv on tg.MaCa = clv.MaCaLamViec |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

# 3.1. Kết nối cơ sở dữ liệu

|  |
| --- |
| public class DBConn  {  private static string connStr = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\QLCuaHang.mdf;Integrated Security=True";  private static SqlConnection connection;  public SqlConnection getConnection  {  get { return connection; }  }  public static void ConnOpen()  {  try  {  connection = new SqlConnection(connStr);  if (connection.State == ConnectionState.Closed)  {  connection.Open();  }  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  public static void ConnClose()  {  if (connection != null && connection.State == ConnectionState.Open)  {  connection.Close();  }  }  } |

**3.2. Xem thông tin danh mục:**

**Hàm chạy các danh mục trên C#:**

|  |
| --- |
| public static DataTable LoadTable(DBConn db, string sqlquery)  {  SqlCommand cmd = new SqlCommand(sqlquery, db.getConnection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  return dataTable;  } |

**3.2.1 Danh mục nhân viên:**

***Thủ tục SQL:***

CREATE View v\_ThongTinNhanVien as

SELECT nv.MaNV, HoTen, GioiTinh, NgaySinh, SDT, Dchi, NgTuyenDung, ChucVu, cv.TenCViec

FROM NHANVIEN nv, CONGVIEC cv, LAMVIEC lv

WHERE nv.MaNV = lv.MaNV

AND lv.MaCViec = cv.MaCViec

***Thực thi trên C#:***

|  |
| --- |
| public static DataTable getEmployees(DBConn db)  {  string sqlquery = “SELECT\* FROM v\_ThongTinNhanVien”;  return LoadTable(db, sqlquery);  } |

**3.2.2.** **Danh mục sản phẩm:**

***Thủ tục SQL:***

CREATE VIEW [dbo].[v\_DanhSachSanPham] AS

SELECT sp.MaSPham, TenSPham, HinhAnh, NhaSanXuat, GiaBan, nsp.TenNhom

FROM SANPHAM sp, NHOMSANPHAM nsp, PHANLOAI pl

WHERE sp.MaSPham = pl.MaSPham AND nsp.MaNhom = pl.MaNhom

***Thực thi trên C#:***

|  |
| --- |
| public static DataTable loadProducts(DBConn db)  {  string sqlquery = "SELECT \* FROM v\_DanhSachSanPham";  return LoadTable(db, sqlquery);  } |

**3.2.3. Danh mục bảng phân ca:**

***Thủ tục SQL:***

|  |
| --- |
| CREATE VIEW [dbo]. V\_BangPhanCa AS  SELECT nv.MaNV, nv.HoTen, nv.ChucVu, clv.MaCaLamViec, clv.ThGianBDau, clv.ThGianKThuc  FROM NHANVIEN nv, CALAMVIEC clv, THAMGIA tg  WHERE nv.MaNV = tg.MaNV AND clv.MaCaLamViec = tg.MaCa |

***Thực thi trên C#:***

|  |
| --- |
| public static DataTable shiftDivision(DBConn db)  {  string sqlquery = “SELECT\* FROM v\_BangPhanCa”;  return LoadTable(db, sqlquery);  } |

**3.3. Chức năng quản lý sản phẩm**

**3.3.1. Thêm sản phẩm**

***\*Hàm SQL:***

CREATE PROCEDURE sp\_ThemSanPham

@masp NVARCHAR(10),

@tensp NVARCHAR(10),

@hinhanh NVARCHAR(50),

@nsx NVARCHAR(50),

@giaban INT,

@tonkho INT,

@daban INT,

@Result INT OUTPUT

AS

BEGIN

BEGIN TRY

IF (EXISTS (SELECT 1

FROM SANPHAM sp

WHERE sp.MaSPham = @masp))

BEGIN

SET @Result = 0;

END

ELSE

BEGIN

INSERT INTO SANPHAM (MaSPham, TenSPham, HinhAnh, NhaSanXuat, GiaBan, TonKho, DaBan)

VALUES (@masp, @tensp, @hinhanh, @nsx, @giaban, @tonkho, @daban);

SET @Result = 1;

END

END TRY

BEGIN CATCH

SET @Result = 0;

END CATCH

RETURN @Result

END

**\*Code C#:**

|  |
| --- |
| public static void InsertProducts(DBConn db, SanPham sp)  {  DBConn.ConnOpen();  try  {  SqlCommand cmd = new SqlCommand("sp\_ThemSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@masp", SqlDbType.NVarChar).Value = sp.MaSP;  cmd.Parameters.Add("@tensp", SqlDbType.NVarChar).Value = sp.TenSP;  cmd.Parameters.Add("@hinhanh", SqlDbType.NVarChar).Value = sp.Hinhanh.ToArray();  cmd.Parameters.Add("@nsx", SqlDbType.NVarChar).Value = sp.NhaSX;  cmd.Parameters.Add("@giaban", SqlDbType.Int).Value = sp.Giaban;  cmd.Parameters.Add("@tonkho", SqlDbType.Int).Value = sp.Tonkho;  cmd.Parameters.Add("@daban", SqlDbType.Int).Value = sp.Daban;  if (cmd.ExecuteNonQuery() > 0)  {  Console.WriteLine("Đã thêm thành công");  }  }  catch (Exception ex)  {  Console.WriteLine("Error: ", ex.Message);  }  DBConn.ConnClose();  } |

**3.3.2. Xóa sản phẩm**

***\*Hàm SQL:***

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_XoaSanPhamTheoMa]  @masp NVARCHAR(10),  @Result INT OUTPUT  AS  BEGIN  BEGIN TRY  IF (EXISTS (SELECT 1  FROM SANPHAM sp  WHERE sp.MaSPham = @masp))  BEGIN  DELETE FROM SANPHAM WHERE SANPHAM.MaSPham = @masp  END  ELSE  BEGIN  SET @Result = 0;  END  END TRY  BEGIN CATCH  SET @Result = 0;  END CATCH  END |

***\*Code C#***

|  |
| --- |
| public static void DeleteProducts(DBConn db, SanPham sp)  {  DBConn.ConnOpen();  try  {  SqlCommand cmd = new SqlCommand("sp\_XoaSanPhamTheoMa", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@masp", SqlDbType.NVarChar).Value = sp.MaSP;  SqlParameter result = new SqlParameter("@Result", SqlDbType.Int)  {  Direction = ParameterDirection.Output  };  cmd.Parameters.Add(result);  if (cmd.ExecuteNonQuery() > 0)  {  Console.WriteLine("Đã xóa thành công");  }  }  catch (Exception ex)  {  Console.WriteLine("Error: ", ex.Message);  }  DBConn.ConnClose();  } |

**3.3.3. Cập nhật sản phẩm**

***\*Hàm SQL:***

CREATE PROCEDURE [dbo].[sp\_SuaSanPham]

@masp NVARCHAR(10),

@tensp NVARCHAR(10),

@hinhanh NVARCHAR(50),

@nsx NVARCHAR(50),

@giaban INT,

@tonkho INT,

@daban INT,

@Result INT OUTPUT

AS

BEGIN

BEGIN TRY

IF (EXISTS (SELECT 1

FROM SANPHAM sp

WHERE sp.MaSPham = @masp))

BEGIN

UPDATE SANPHAM

SET TenSPham = @tensp,

HinhAnh = @hinhanh,

NhaSanXuat = @nsx,

GiaBan = @giaban,

TonKho = @tonkho,

DaBan = @daban

WHERE MaSPham = @masp;

SET @Result = 1;

END

ELSE

BEGIN

SET @Result = 0;

END

END TRY

BEGIN CATCH

SET @Result = 0;

END CATCH

RETURN @Result

END

***\*Code C#***

|  |
| --- |
| public static void UpdateProducts(DBConn db, SanPham sp)  {  try  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("sp\_SuaSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@masp", SqlDbType.NVarChar).Value = sp.MaSP;  cmd.Parameters.Add("@tensp", SqlDbType.NVarChar).Value = sp.TenSP;  cmd.Parameters.Add("@hinhanh", SqlDbType.NVarChar).Value = sp.Hinhanh.ToArray();  cmd.Parameters.Add("@nsx", SqlDbType.NVarChar).Value = sp.NhaSX;  cmd.Parameters.Add("@giaban", SqlDbType.Int).Value = sp.Giaban;  cmd.Parameters.Add("@tonkho", SqlDbType.Int).Value = sp.Tonkho;  cmd.Parameters.Add("@daban", SqlDbType.Int).Value = sp.Daban;  SqlParameter result = new SqlParameter("@Result", SqlDbType.Int)  {  Direction = ParameterDirection.Output  };  cmd.Parameters.Add(result);  if (cmd.ExecuteNonQuery() > 0)  {  Console.WriteLine("Đã sửa thành công");  }  }  catch (Exception ex)  {  Console.WriteLine("Error: ", ex.Message);  }  DBConn.ConnClose();  } |

**3.3.4. Xem thông tin sản phẩm**

***\*Hàm SQL:***

CREATE FUNCTION [dbo].[ fn\_LayDanhSachSanPham] ()

RETURNS TABLE

AS

RETURN

(

SELECT MaSPham, TenSPham, HinhAnh, NhaSanXuat, GiaBan, TonKho, DaBan

FROM SANPHAM

);

GO

***\*Code C#***

|  |
| --- |
| public List<SanPham> ProductList(DBConn db)  {  List<SanPham> danhsachsp = new List<SanPham>();  try  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("SELECT \* FROM fn\_LayDanhSachSanPham", db.getConnection);  cmd.CommandType = CommandType.Text;  SqlDataReader reader = cmd.ExecuteReader();  SanPham sp;  if (reader.Read())  {  sp = new SanPham();  sp.MaSP = reader["MaSPham"].ToString();  ;  sp.TenSP = reader["TenSPham"].ToString();  sp.Hinhanh = reader["HinhAnh"].ToString();  sp.NhaSX = reader["NhaSanXuat"].ToString();  sp.Giaban = Convert.ToInt32(reader["GiaBan"]);  sp.Tonkho = Convert.ToInt32(reader["TonKho"]);  sp.Daban = Convert.ToInt32(reader["DaBan"]);  danhsachsp.Add(sp);  }  reader.Close();  }  catch (Exception ex)  {  Console.WriteLine("Error: ", ex.Message);  }  DBConn.ConnClose();  return danhsachsp;  } |

**3.3.5. Lấy thông tin sản phẩm bằng mã sản phẩm**

***\*Hàm SQL:***

|  |
| --- |
| CREATE FUNCTION fn\_LayThongTinSanPham (@masp NVARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT MaSPham, TenSPham, HinhAnh, NhaSanXuat, GiaBan, TonKho, DaBan  FROM SANPHAM  WHERE SANPHAM.MaSPham = @masp  );  GO |

***\*Code C#***

|  |
| --- |
| public SanPham GetProductById(DBConn db, string masp)  {  SanPham sp = new SanPham();  try  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("SELECT \* FROM fn\_LayThongTinSanPham(@masp)", db.getConnection);  cmd.CommandType = CommandType.Text;  SqlDataReader reader = cmd.ExecuteReader();  if (reader.Read())  {  sp = new SanPham  {  MaSP = reader["MaSPham"].ToString(),  TenSP = reader["TenSPham"].ToString(),  Hinhanh = reader["HinhAnh"].ToString(),  NhaSX = reader["NhaSanXuat"].ToString(),  Giaban = Convert.ToInt32(reader["GiaBan"]),  Tonkho = Convert.ToInt32(reader["TonKho"]),  Daban = Convert.ToInt32(reader["DaBan"])  };  }  reader.Close();  }  catch (Exception ex)  {  Console.WriteLine("Error: ", ex.Message);  }  DBConn.ConnClose();  return sp;  } |

**3.3.6. Tìm kiếm sản phẩm (bằng các thông tin trừ hình ảnh)**

***\*Hàm SQL:***

CREATE FUNCTION fn\_TimKiemSanPham (@GiaTriTimKiem NVARCHAR(150))

RETURNS TABLE

AS

RETURN

(

SELECT DISTINCT

sp.MaSPham, sp.TenSPham, sp.HinhAnh, sp.NhaSanXuat, sp.GiaBan, sp.TonKho, sp.DaBan

FROM SANPHAM sp

INNER JOIN PHANLOAI pl ON sp.MaSPham = pl.MaSPham

INNER JOIN NHOMSANPHAM nsp ON pl.MaNhom = nsp.MaNhom

WHERE

sp.MaSPham LIKE '%' + @GiaTriTimKiem + '%'

OR sp.TenSPham LIKE '%' + @GiaTriTimKiem + '%'

OR sp.NhaSanXuat LIKE '%' + @GiaTriTimKiem + '%'

OR CAST(sp.GiaBan AS NVARCHAR(150)) LIKE '%' + @GiaTriTimKiem + '%'

OR CAST(sp.TonKho AS NVARCHAR(150)) LIKE '%' + @GiaTriTimKiem + '%'

OR CAST(sp.DaBan AS NVARCHAR(150)) LIKE '%' + @GiaTriTimKiem + '%'

);

GO

***\*Code C#***

|  |
| --- |
| public List<SanPham> FindProduct(DBConn db, string chuoitimkiem)  {  List<SanPham> danhsachsp = new List<SanPham>();  try  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("SELECT \* FROM fn\_TimKiemSanPham(@GiaTriTimKiem)", db.getConnection);  cmd.CommandType = CommandType.Text;  cmd.Parameters.AddWithValue("@GiaTriTimKiem", chuoitimkiem);  SqlDataReader reader = cmd.ExecuteReader();  if (reader.Read())  {  SanPham sp = new SanPham()  {  MaSP = reader["MaSPham"].ToString(),  TenSP = reader["TenSPham"].ToString(),  Hinhanh = reader["HinhAnh"].ToString(),  NhaSX = reader["NhaSanXuat"].ToString(),  Giaban = Convert.ToInt32(reader["GiaBan"]),  Tonkho = Convert.ToInt32(reader["TonKho"]),  Daban = Convert.ToInt32(reader["DaBan"])  };  danhsachsp.Add(sp);  }  reader.Close();  }  catch (Exception ex)  {  Console.WriteLine("Error: ", ex.Message);  }  DBConn.ConnClose();  return danhsachsp;  } |

**3.5. Chức năng quản lý đơn nhập**

**3.5.1. Thêm đơn nhập (6.1)**

**3.5.2. Cập nhật đơn nhập**

**3.5.5. Xóa đơn nhập**

**3.5.7. Lấy ra toàn bộ đơn nhập (6.6)**

**3.5.8. Tìm kiếm đơn nhập (6.10)**

**3.6. Chức năng quản lý hóa đơn và thông tin hóa đơn**

**3.6.1. Thêm hóa đơn (6.2)**

**3.6.2. Cập nhật hóa đơn (6.3)**

**3.6.3. Xóa hóa đơn**

**3.6.4. Lấy ra toàn bộ hóa đơn (6.5)**

**3.6.5. Tìm kiếm hóa đơn (6.9)**

**3.6.7. Nhập số tiền khách đưa**

**3.6.8. Chỉnh sửa số lượng sản phẩm trên hóa đơn**

**3.7. Chức năng quản lý ca làm việc**

**3.7.1. Cập nhật mã ca**

**3.7.2. Cập nhật thời gian bắt đầu ca làm việc**

**3.7.3. Cập nhật thời gian kết thúc ca làm việc**

**3.8. Chức năng quản lý thông tin nhân viên**

**3.8.1. Thêm nhân viên**

***\*Hàm SQL:***

|  |
| --- |
| CREATE PROC [dbo].[ADDNhanVien] (@MaNV nvarchar(10),@HoTen nvarchar(20), @GioiTinh nvarchar(5), @NgaySinh date, @SDT nvarchar(20), @DiaChi nvarchar(50), @AnhDaiDien nvarchar(100),@MKhau nvarchar(50), @NgayTuyenDung date, @ChucVu nvarchar(20))  AS  BEGIN  INSERT INTO NHANVIEN VALUES (@MaNV, @HoTen, @GioiTinh , @NgaySinh ,@SDT , @DiaChi , @AnhDaiDien ,@MKhau ,@NgayTuyenDung , @ChucVu )  END |

***\*Code C#***

|  |
| --- |
| public static void ThemNhanVien(NhanVien nv)  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("ADDNhanVien", DBConn.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.NVarChar, 10)).Value = nv.MaNV;  cmd.Parameters.Add(new SqlParameter("@HoTen", SqlDbType.NVarChar, 20)).Value = nv.HoTen;  cmd.Parameters.Add(new SqlParameter("@GioiTinh", SqlDbType.NVarChar, 5)).Value = nv.GioiTinh;  cmd.Parameters.Add(new SqlParameter("@NgaySinh", SqlDbType.Date)).Value = nv.NgaySinh;  cmd.Parameters.Add(new SqlParameter("@SDT", SqlDbType.NVarChar, 20)).Value = nv.Sdt;  cmd.Parameters.Add(new SqlParameter("@DiaChi", SqlDbType.NVarChar, 50)).Value = nv.DChi;  cmd.Parameters.Add(new SqlParameter("@AnhDaiDien", SqlDbType.NVarChar, 100)).Value = nv.AnhDaiDien;  cmd.Parameters.Add(new SqlParameter("@MKhau", SqlDbType.NVarChar, 50)).Value = nv.MKhau;  cmd.Parameters.Add(new SqlParameter("@NgayTuyenDung", SqlDbType.Date)).Value = nv.NgayTuyenDung;  cmd.Parameters.Add(new SqlParameter("@ChucVu", SqlDbType.NVarChar, 20)).Value = nv.ChucVu;  try  {  int rowsAffected = cmd.ExecuteNonQuery();  Console.WriteLine($"{rowsAffected} row(s) inserted. Thêm nhân viên thành công.");  }  catch (Exception ex)  {  Console.WriteLine("Error: " + ex.Message);  }  DBConn.ConnClose();  } |

**3.8.2. Xóa nhân viên**

***\*Hàm SQL:***

|  |
| --- |
| CREATE PROCEDURE [dbo].[Xoa\_NV] (@MaNV nvarchar(10))  AS  BEGIN  DELETE FROM NHANVIEN WHERE MaNV = @MaNV;  END |

***\*Code C#:***

|  |
| --- |
| public static void XoaNhanVien (NhanVien nv)  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("Xoa\_NV", DBConn.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.NVarChar, 10)).Value = nv.MaNV;  try  {  int rowsAffected = cmd.ExecuteNonQuery();  Console.WriteLine($"{rowsAffected} row(s) inserted. Xóa nhân viên thành công.");  }  catch (Exception ex)  {  Console.WriteLine("Error: " + ex.Message);  }  DBCon |

**3.8.3. Sửa thông tin nhân viên**

***\*Hàm SQL:***

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaNhanVien]  @MaNV nvarchar(10),  @HoTen nvarchar(20),  @GioiTinh nvarchar(5),  @NgaySinh date,  @SDT nvarchar(20),  @DiaChi nvarchar(50),  @AnhDaiDien nvarchar(100),  @MKhau nvarchar(50),  @NgayTuyenDung date,  @ChucVu nvarchar(20)  AS  BEGIN  UPDATE NHANVIEN  SET  HoTen = @HoTen,  GioiTinh = @GioiTinh,  NgaySinh = @NgaySinh,  SDT = @SDT,  DChi = @DiaChi,  AnhDaiDien = @AnhDaiDien,  MKhau = @MKhau,  NgTuyenDung = @NgayTuyenDung,  ChucVu = @ChucVu  WHERE  MaNV = @MaNV;  END; |

***\*Code C#***

|  |
| --- |
| public static void SuaNhanVien (NhanVien nv)  {  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("SuaNhanVien", DBConn.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.NVarChar, 10)).Value = nv.MaNV;  cmd.Parameters.Add(new SqlParameter("@HoTen", SqlDbType.NVarChar, 20)).Value = nv.HoTen;  cmd.Parameters.Add(new SqlParameter("@GioiTinh", SqlDbType.NVarChar, 5)).Value = nv.GioiTinh;  cmd.Parameters.Add(new SqlParameter("@NgaySinh", SqlDbType.Date)).Value = nv.NgaySinh;  cmd.Parameters.Add(new SqlParameter("@SDT", SqlDbType.NVarChar, 20)).Value = nv.Sdt;  cmd.Parameters.Add(new SqlParameter("@DiaChi", SqlDbType.NVarChar, 50)).Value = nv.DChi;  cmd.Parameters.Add(new SqlParameter("@AnhDaiDien", SqlDbType.NVarChar, 100)).Value = nv.AnhDaiDien;  cmd.Parameters.Add(new SqlParameter("@MKhau", SqlDbType.NVarChar, 50)).Value = nv.MKhau;  cmd.Parameters.Add(new SqlParameter("@NgayTuyenDung", SqlDbType.Date)).Value = nv.NgayTuyenDung;  cmd.Parameters.Add(new SqlParameter("@ChucVu", SqlDbType.NVarChar, 20)).Value = nv.ChucVu;  try  {  int rowsAffected = cmd.ExecuteNonQuery();  Console.WriteLine($"{rowsAffected} row(s) inserted. Sửa thông tin nhân viên thành công.");  }  catch (Exception ex)  {  Console.WriteLine("Error: " + ex.Message);  }  DBConn.ConnClose();  } |

**3.8.4. Chức năng tìm nhân viên theo tên**

***\*Hàm SQL:***

|  |
| --- |
| create proc [dbo].[SearchTenNV] @str nvarchar(20)  as  begin  select \*  from NHANVIEN  where HoTen like '%'+ @str + '%';  end |

***\*Code C#***

|  |
| --- |
| public static DataTable TimTenNhanVien(NhanVien nv)  {  DataTable dt = new DataTable();  DBConn.ConnOpen();  SqlCommand cmd = new SqlCommand("SearchTenNV", DBConn.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add(new SqlParameter("@str", SqlDbType.NVarChar, 10)).Value = nv.HoTen;  try  {  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(dt);  }  catch (Exception ex)  {  MessageBox.Show("Error: " + ex.Message);  }  return dt;  } |