**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🙟🕮🙝🙢

Logo

Description automatically generated

**BÁO CÁO CUỐI KỲ**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**Mã học phần: DBMS330284**

**HỆ THỐNG QUẢN LÝ**

**KINH DOANH BÁN LẺ NƯỚC UỐNG**

**Nhóm sinh viên thực hiện:**

1. **Vũ Bảo Long 22110368.**
2. **Trần Thị Kim Chung 22110288.**
3. **Phan Tất Duy 22110296.**
4. **Nguyễn Duy Đạt 22110307.**

**Giảng viên hướng dẫn: TS. Nguyễn Thành Sơn.**

**TP Hồ Chí Minh, ngày 20 tháng 11 năm 2024**

**DANH SÁCH NHÓM SINH VIÊN THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2024-2025**

**NHÓM 7**

*Đề tài*: Hệ thống quản lý kinh doanh bán lẻ nước uống.

|  |  |  |
| --- | --- | --- |
| **Mã số sinh viên** | **Họ và tên** | **Tỉ lệ hoàn thành** |
| 22110368 | Vũ Bảo Long (Nhóm trưởng) | 100% |
| 22110288 | Trần Thị Kim Chung | 100% |
| 22110296 | Phan Tất Duy | 100% |
| 22110307 | Nguyễn Duy Đạt | 100% |

Nhận xét của giảng viên

TP.HCM, ngày 20 tháng 11 năm 2024

TS. Nguyễn Thành Sơn

**LỜI CẢM ƠN**

Kính gửi thầy Nguyễn Thành Sơn,

Nhóm 7 chúng em xin cảm ơn thầy vì những kiến thức bổ ích mà chúng em được học, đồng thời thầy đã theo sát chỉ dẫn và đánh giá quá trình làm đồ án kết thúc môn học Hệ quản trị cơ sở dữ liệu của chúng em.

Nhờ vào đó chúng em có thể tạo nên được một thành phẩm hoàn chỉnh, tuy chưa hoàn hảo và còn nhiều sai sót mong thầy có thể bỏ qua.

Kiến thức chúng em học được từ môn Hệ quản trị cơ sở dữ liệu vô cùng quý giá đối với những môn học sau này và đồng thời cũng trong công việc của chúng em.

Em xin chân thành cảm ơn thầy,

Nhóm 7,

Vũ Bảo Long,

Trần Thị Kim Chung,

Nguyễn Duy Đạt,

Phan Tất Duy.

**MỤC LỤC**

**[Chương 1: TỔNG QUAN VỀ HỆ THỐNG 8](#_Toc9794)**

[1. Đặc tả đề tài: 8](#_Toc30489)

[2. Nghiệp vụ bài toán. 9](#_Toc23644)

**[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU. 11](#_Toc3691)**

[1. Thiết kế cơ sở dữ liệu mức quan niệm. 11](#_Toc22315)

[2. Thiết kế cơ sở dữ liệu mức logic. 12](#_Toc4821)

[3. Các ràng buộc cần thiết. 12](#_Toc17930)

[4. Cài đặt cơ sở dữ liệu và các ràng buộc. 14](#_Toc16895)

[5. Database Diagrams. 18](#_Toc29910)

[6. Các trigger. 19](#_Toc29148)

[6.1 Trigger mặc định trạng thái khi thêm hóa đơn là “Chưa thanh toán”. 19](#_Toc23832)

[6.2 Trigger thay đổi số lượng hàng khi bán được sản phẩm. 19](#_Toc19374)

[6.3 Trigger kiểm tra số lượng hàng hóa có đủ để bán hay không. 20](#_Toc10616)

[6.4 Trigger bắt lỗi trùng tên khi thêm và sửa sản phẩm. 21](#_Toc9254)

[6.5 Trigger bắt lỗi trùng tên khi thêm và sửa nhà cung cấp. 22](#_Toc1516)

[6.6 Trigger bắt lỗi khi trùng số điện thoại khi thêm nhân viên. 22](#_Toc345)

[6.7 Trigger bắt lỗi khi trùng số điện thoại khi thêm nhà cung cấp. 23](#_Toc10817)

[7. Một số transaction. 23](#_Toc9647)

[7.1 Procedure ThemHoaDon sử dụng transaction. 23](#_Toc14821)

[7.2 Procedure ThemDonNhapHang sử dụng transaction. 25](#_Toc28403)

[7.3 Trigger trg\_CheckDuplicateNhaCungCap sử dụng transaction. 27](#_Toc11046)

[7.4 Trigger KiemTraTrungSDT\_NV sử dụng transaction. 28](#_Toc14655)

[7.5 Trigger KiemTraTrungSDT\_NCC sử dụng transaction. 28](#_Toc20695)

[8. Index. 29](#_Toc16356)

**[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 30](#_Toc5868)**

[1. Kết nối cơ sở dữ liệu. 30](#_Toc5175)

[2. Xem thông tin các danh mục 31](#_Toc2143)

[2.1 Xem danh sách nhân viên. 31](#_Toc20034)

[2.2 Xem danh sách các hóa đơn. 32](#_Toc8759)

[2.3 Xem danh sách các sản phẩm. 33](#_Toc20335)

[2.4 Xem danh sách các loại sản phẩm. 36](#_Toc16050)

[2.5 Xem danh sách đơn nhập hàng. 37](#_Toc30102)

[2.6 Xem danh sách nhà cung cấp. 38](#_Toc18066)

[3. Quản lý nhân viên. 39](#_Toc10131)

[3.1 Tìm kiếm nhân viên. 39](#_Toc10717)

[3.2 Thêm nhân viên: 40](#_Toc3978)

[3.3 Sửa nhân viên. 42](#_Toc17298)

[4. Quản lý hóa đơn. 44](#_Toc14397)

[4.1 Thêm hóa đơn. 44](#_Toc10278)

[4.2 Hủy hóa đơn. 46](#_Toc11808)

[4.3 Xác nhận đã thanh toán hóa đơn. 48](#_Toc7362)

[4.4 Tìm kiếm hóa đơn. 50](#_Toc2582)

[5. Quản lý đơn nhập hàng. 52](#_Toc9628)

[6. Quản lý sản phẩm. 57](#_Toc31605)

[6.1 Thêm sản phẩm. 57](#_Toc3140)

[6.2 Xóa sản phẩm. 59](#_Toc8213)

[6.3 Sửa sản phẩm. 60](#_Toc29974)

[6.4 Tìm kiếm sản phẩm. 62](#_Toc23337)

[7. Quản lý loại sản phẩm. 63](#_Toc30299)

[7.1 Thêm loại sản phẩm. 63](#_Toc1655)

[7.2 Sửa loại sản phẩm. 64](#_Toc4923)

[8. Quản lý nhà cung cấp. 66](#_Toc20247)

[8.1 Thêm nhà cung cấp. 66](#_Toc26807)

[8.2 Sửa nhà cung cấp. 67](#_Toc6444)

[9. Thống kê doanh thu và chi phí nhập hàng. 68](#_Toc8453)

[9.1 Doanh thu theo ngày. 68](#_Toc19007)

[9.2 Doanh thu theo tháng. 69](#_Toc9966)

[9.3 Doanh thu theo năm. 71](#_Toc30656)

[9.4 Chi phí nhập hàng theo tháng. 72](#_Toc16606)

[9.5 Chi phí nhập hàng theo năm. 74](#_Toc8756)

[10. Báo cáo. 76](#_Toc14800)

[10.1 Tương quan doanh thu 2 tháng bất kỳ. 76](#_Toc11026)

[10.2 Top 3 sản phẩm bán chạy nhất. 79](#_Toc11834)

[10.3 Top 3 nhân viên bán hàng xuất sắc nhất. 82](#_Toc25191)

**[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 85](#_Toc1437)**

[1. Phân quyền cho các role. 85](#_Toc12981)

[2. Procedure tạo tài khoản phần quyền Sql Server và tài khoản phần mềm. 86](#_Toc9584)

[3. Procedure sp\_XoaUser để xóa nhân viên. 87](#_Toc25768)

**[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG. 88](#_Toc4214)**

[1. Form đăng nhập. 88](#_Toc407)

[2. Form Trang chủ. 91](#_Toc32200)

[3. Các loại danh mục. 92](#_Toc21226)

[3.1 Danh mục nhân viên. 92](#_Toc3346)

[3.2 Danh mục hóa đơn. 92](#_Toc11744)

[3.3 Danh mục sản phẩm. 93](#_Toc10709)

[3.4 Danh mục loại sản phẩm. 93](#_Toc30279)

[3.5 Danh mục đơn nhập hàng. 94](#_Toc32178)

[3.6 Danh mục nhà cung cấp. 94](#_Toc22557)

[4. Các chức năng quản lý. 95](#_Toc4613)

[4.1 Quản lý nhân viên. 95](#_Toc30532)

[4.2 Quản lý hóa đơn. 95](#_Toc855)

[4.3 Quản lý sản phẩm. 96](#_Toc3486)

[4.4 Quản lý loại sản phẩm. 97](#_Toc32118)

[4.5 Quản lý đơn nhập hàng. 97](#_Toc29525)

[4.6 Quản lý nhà cung cấp. 98](#_Toc11570)

[4.7 Quản lý người dùng. 98](#_Toc10780)

[5. Báo cáo 99](#_Toc32011)

[5.1 Báo cáo doanh thu theo ngày, tháng, năm;tiền nhập hàng theo tháng, năm. 99](#_Toc2980)

[5.2 Tương quan doanh thu 2 tháng bất kỳ. 99](#_Toc31904)

[5.3 Biểu đồ doanh thu. 100](#_Toc3264)

[5.4 Top 3 nhân viên xuất sắc. 100](#_Toc20750)

[5.5 Top 3 sản phẩm bán chạy nhất. 101](#_Toc4039)

**Chương 1: TỔNG QUAN VỀ HỆ THỐNG**

1. **Đặc tả đề tài:**

Một ông chủ cửa hàng kinh doanh nước uống bán lẻ cần có một hệ thống để quản lý việc buôn bán của cửa hàng. Chương trình quản lý được xây dựng cho hai đối tượng người dùng là người quản lý (chủ cửa hàng) và nhân viên. Đối với người dùng là nhân viên, họ có quyền nhập và xuất hóa đơn bán lẻ, thêm đơn nhập hàng, xem bảng sản phẩm, loại sản phẩm, hóa đơn…. Còn đối với người dùng là quản lý, họ sẽ có quyền truy cập vào các chức năng như quản lý kho, quản lý doanh thu, quản lý nhân viên. Đối với hệ thống quản lý cửa hàng này cần lưu trữ những dữ liệu sau:

Có một nhóm nhân viên: nhân viên có nhiệm vụ tạo hóa đơn, nhập đơn nhập hàng... Các nhân viên phân biệt với nhau bằng mã nhân viên, mỗi nhân viên gồm các thông tin như mã nhân viên, họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, hình ảnh nhân viên.

Sản phẩm được chia làm nhiều loại, các loại sản phẩm phân biệt với nhau bằng mã loại sản phẩm. Mỗi loại sản phẩm có thể có nhiều sản phẩm, một sản phẩm chỉ thuộc một loại sản phẩm. Các sản phẩm phân biệt với nhau bằng mã sản phẩm. Mỗi sản phẩm có các thông tin như mã loại sản phẩm, mã sản phẩm, tên sản phẩm, giá nhập, giá bán, số lượng, hình ảnh sản phẩm. Để quản lý sản phẩm cần có thông tin của nhà sản xuất. Các nhà cung cấp phân biệt với nhau bằng mã nhà cung cấp. Mỗi nhà cung cấp có các thông tin như mã nhà cung cấp, tên nhà cung cấp, địa chỉ, số điện thoại. Đơn nhập hàng được phân biệt bởi mã đơn nhập hàng, mỗi đơn nhập hàng có các thông tin như mã đơn nhập hàng, ngày nhập hàng, ngày hết hạn, trị giá đơn nhập hàng, mã nhà cung cấp. Chi tiết đơn nhập hàng được phân biệt qua Mã đơn nhập hàng và Mã sản phẩm, trong đó gồm các thành phần đơn giá, số lượng và tổng tiền. Trong chi tiết đơn nhập hàng, mỗi đơn nhập hàng có nhiều sản phẩm và mỗi sản phẩm cũng có thể được nhập nhiều lần và xuất hiện nhiều lần trong các đơn nhập hàng khác nhau, vì vậy chi tiết đơn nhập hàng là mối quan hệ kết nối giữa Đơn nhập hàng và Sản phẩm.

Hóa đơn bán lẻ được phân biệt với nhau qua mã hóa đơn, mỗi hóa đơn gồm các thông tin như mã hóa đơn, ngày bán hàng, mã nhân viên bán hàng, trạng thái thanh toán của hóa đơn. Chi tiết hóa đơn được phân biệt qua mã hóa đơn và mã sản phẩm, trong Chi tiết hóa đơn gồm mã hóa đơn, mã sản phẩm, đơn giá, tổng tiền, số lượng. Trong chi tiết hóa đơn, mỗi hóa đơn có nhiều sản phẩm, và mỗi sản phẩm có thể xuất hiện trong nhiều hóa đơn khác nhau, vì vậy chi tiết hóa đơn là mối quan hệ kết nối giữa Sản phẩm và Hóa đơn.

Quản lý kho cho phép quản lý xem, thêm, xóa và chỉnh sửa số lượng sản phẩm nhập vào cửa hàng.

Quản lý doanh thu sẽ cho phép quản lý xuất báo cáo doanh thu theo ngày, tháng dựa trên những thông tin được lưu trữ trong Hóa Đơn và Chi tiết hóa đơn như ngày bán hàng, tổng tiền, đơn giá,...

Quản lý nhân viên cho phép quản lý xem, thêm, xóa và sửa thông tin nhân viên.

1. **Nghiệp vụ bài toán.**

Phần mềm quản lý kinh doanh bán lẻ nước uống làm nghiệp vụ kinh doanh trong lĩnh vực buôn bán hàng hóa. Nó bao gồm việc quản lý nhân viên, quản lý sản phẩm và loại sản phẩm, quản lý kho và nhập hàng, quản lý doanh thu, quản lý người dùng.

Quản lý nhân viên cho phép việc nắm được thông tin nhân viên, thêm mới nhân viên, sửa thông tin nhân viên và xóa nhân viên nếu không còn làm việc. Thêm vào đó có chức năng tìm ra top 3 nhân viên bán hàng xuất sắc hàng tháng kèm theo doanh số bán hàng để có thể thấy hiệu suất làm việc của nhân viên.

Quản lý sản phẩm và loại sản phẩm cho phép việc nắm bắt sản phẩm có trong kho, thêm mới sản phẩm, sửa sản phẩm. Đồng thời cũng có chức năng thống kê việc bán sản phẩm giúp tìm ra sản phẩm bán chạy để củng cố chiến lược kinh doanh.

Quản lý kho và nhập hàng cho phép việc quản lý số lượng hàng trong kho, khi thêm đơn nhập hàng thì số lượng sản phẩm trong kho sẽ tăng lên. Qua đó có thể dễ dàng quản lý việc nhập hàng khi sản phẩm hết vào thông báo hết hàng khi có yêu cầu thêm sản phẩm vào hóa đơn.

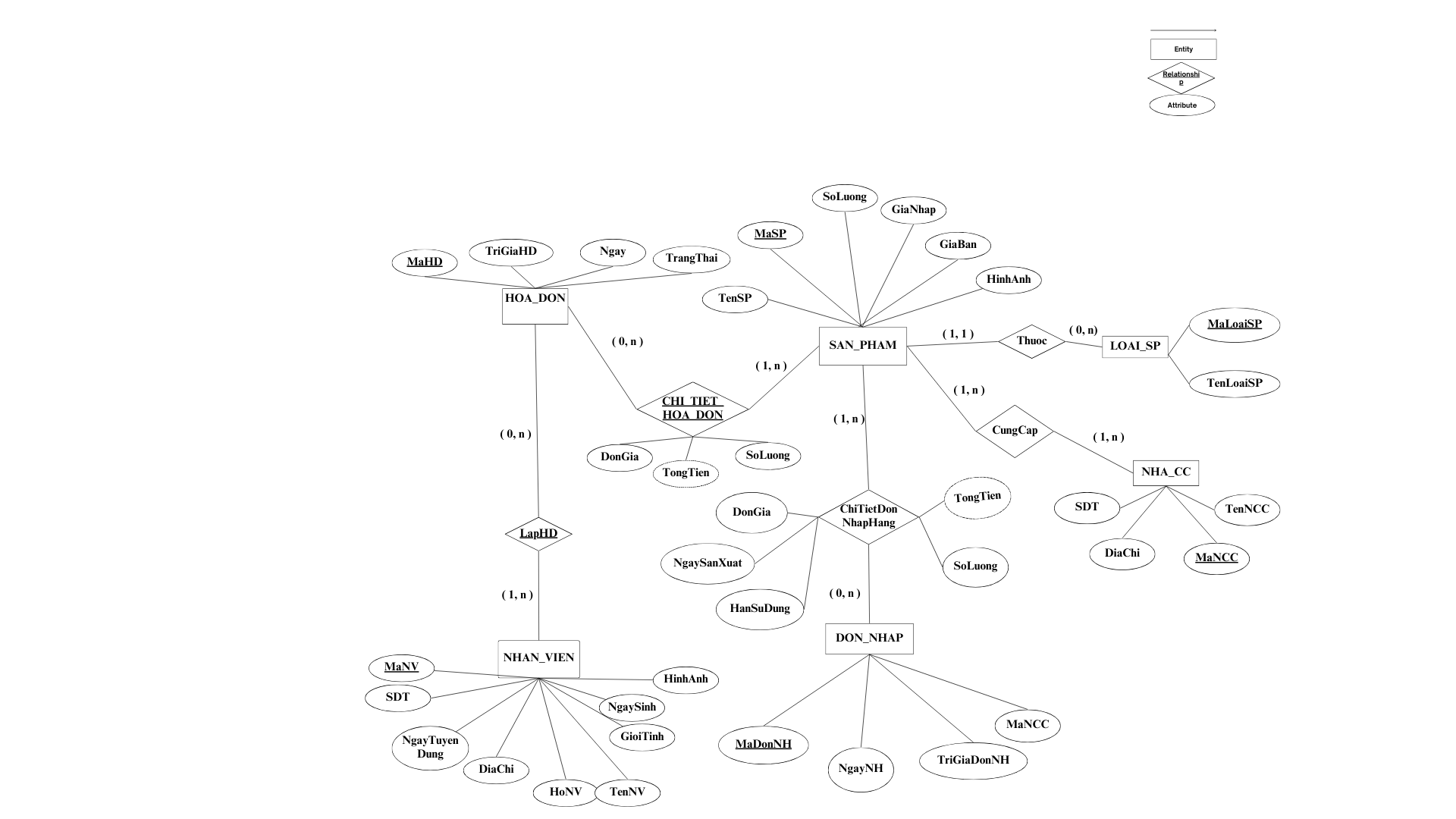
Quản lý doanh thu cho phép chủ cửa hàng xem xét doanh thu theo ngày hoặc theo tháng, theo năm; Xem tương quan doanh thu giữa 2 tháng bất kỳ và xem độ lệch doanh thu giữa 2 tháng ấy. Xem xét được top những sản phẩm bán chạy và những nhân viên năng suất nhất.

Quản lý người dùng cho phép thêm và xóa tài khoản, cấp quyền truy cập. Nếu là chủ cửa hàng hoặc admin sẽ được cấp toàn quyền và nhân viên sẽ bị giới hạn ở một số quyền nhất định.

**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU.**

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

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



1. **Thiết kế cơ sở dữ liệu mức logic.**

**NHAN\_VIEN**(MaNV, HoNV, TenNV, NgaySinh, GioiTinh, DiaChi, NgayTuyenDung, SDT, HinhAnh)

**HOA\_DON**(MaHD, Ngay, MaNV, TriGiaHD, TrangThai)

**LOAI\_SP**(MaLoaiSP, TenLoaiSP)

**SAN\_PHAM**(MaSP, TenSP, GiaNhap, GiaBan, SoLuong, MaLoaiSP, HinhAnh)

**CHI\_TIET\_HOA\_DON**(MaHD, MaSP, SoLuong, TongTien, DonGia)

**NHA\_CUNG\_CAP**(MaNCC, TenNCC, DiaChi, SDT)

**DON\_NHAP**(MaDonNH, NgayNH, TriGiaDonNH, MaNCC)

**CHI\_TIET\_DON\_NHAP\_HANG**( MaDonNH,MaSP, DonGia, SL, TongTien, NgaySanXuat, HanSuDung)

1. **Các ràng buộc cần thiết.**

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| 1 | **NHAN\_VIEN** | Khóa chính MaNV int,  Not Null HoNV varchar(50),  Not Null TenNV varchar(50),  Not Null SDT varchar(10),  Not Null NgaySinh date,  Not Null NgayTuyenDung date,  Not Null DiaChi varchar(100),  Not Null HinhAnh Image,  Check NgaySinh (>= 18 tuổi),  Check NgayTuyenDung (> 0),  Check SDT (=10) |
| 2 | **HOA\_DON** | Khóa chính MaHD int,  Not Null Ngay date,  Not Null MaNV,  Khóa ngoại MaNV int đến bảng NHAN\_VIEN,  Not Null TriGiaHD decimal(15,2) |
| 3 | **LOAI\_SAN\_PHAM** | Khóa chính MaLoaiSP int,  Not Null TenLoaiSP varchar(50) |
| 4 | **SAN\_PHAM** | Khóa chính MaSP int,  Khóa ngoại MaLoaiSP int đến bảng LOAI\_SANPHAM,  Not Null TenSP varchar(100),  Not Null GiaNhap decimal(10,2),  Not Null GiaBan decimal(10,2),  Not Null SoLuong int,  Not Null HinhAnh Image,  Check GiaNhap ( > 0),  Check GiaBan ( > 0),  Check SoLuong ( >= 0) |
| 5 | **CHI\_TIET\_HOA\_DON** | Khóa chính MaHD int, MaSP int,  Not Null SoLuong int,  Not Null TongTien decimal(15,2),  Not Null DonGia decimal(10,2),  Check SoLuong ( > 0),  Check TongTien ( > 0),  Check DonGia ( > 0 ) |
| 6 | **NHA\_CUNG\_CAP** | Khóa chính MaNCC int,  Not Null TenNCC varchar(100),  Not Null DiaChi varchar(100),  Not Null SDT varchar(10),  Check SDT ( = 10) |
| 7 | **DON\_NHAP** | Khóa chính MaDonNH int,  Khóa ngoại MaNCC int đến bảng NHA\_CUNG\_CAP,  Not Null NgayNH date,  Not Null TriGiaDonNH decimal(15,2) |
| 8 | **CHI\_TIET\_DON\_NHAP\_HANG** | Khóa chính MaDonNH int, MaSP int,  Not Null DonGia decimal(10,2),  Not Null SL int,  Not Null TongTien decimal(15,2),  Check (NgaySanXuat < HanSuDung),  Check (HanSuDung > '1111-01-01'),  Check DonGia ( > 0)  Check SL ( > 0),  Check TongTien ( > 0) |

1. **Cài đặt cơ sở dữ liệu và các ràng buộc.**

***Bảng Nhân viên***

CREATE TABLE NHAN\_VIEN (

MaNV INT PRIMARY KEY IDENTITY(1,1),

HoNV VARCHAR(50) NOT NULL,

TenNV VARCHAR(50) NOT NULL,

NgaySinh DATE NOT NULL,

SDT VARCHAR(10) NOT NULL,

NgayTuyenDung DATE NOT NULL,

DiaChi VARCHAR(100) NOT NULL,

HinhAnh IMAGE,

CHECK (DATEDIFF(YEAR, NgaySinh, GETDATE()) >= 18),

CHECK (NgayTuyenDung > '1111-01-01'),

CHECK (LEN(SDT) = 10)

);

***Bảng Hóa đơn***

CREATE TABLE HOA\_DON (

MaHD INT PRIMARY KEY IDENTITY(1,1),

Ngay DATE NOT NULL,

MaNV INT NOT NULL,

TrangThai NVARCHAR(20) NOT NULL,

TriGiaHD DECIMAL(15, 2) NOT NULL,

FOREIGN KEY (MaNV) REFERENCES NHAN\_VIEN(MaNV)

);

***Bảng Loại sản phẩm***

CREATE TABLE LOAI\_SAN\_PHAM (

MaLoaiSP INT PRIMARY KEY IDENTITY(1,1),

TenLoaiSP VARCHAR(50) NOT NULL

);

***Bảng Sản phẩm***

CREATE TABLE SAN\_PHAM (

MaSP INT PRIMARY KEY IDENTITY(1,1),

TenSP VARCHAR(100) NOT NULL,

GiaNhap DECIMAL(10, 2) NOT NULL,

GiaBan DECIMAL(10, 2) NOT NULL,

SoLuong INT NOT NULL,

MaLoaiSP INT NOT NULL,

HinhAnh IMAGE,

FOREIGN KEY (MaLoaiSP) REFERENCES LOAI\_SAN\_PHAM(MaLoaiSP),

CHECK (GiaNhap > 0),

CHECK (GiaBan > 0),

CHECK (SoLuong >= 0)

);

***Bảng Chi tiết hóa đơn***

CREATE TABLE CHI\_TIET\_HOA\_DON (

MaHD INT,

MaSP INT,

SoLuong INT NOT NULL,

TongTien DECIMAL(15, 2) NOT NULL,

DonGia DECIMAL(10, 2) NOT NULL,

PRIMARY KEY (MaHD, MaSP),

FOREIGN KEY (MaHD) REFERENCES HOA\_DON(MaHD),

FOREIGN KEY (MaSP) REFERENCES SAN\_PHAM(MaSP),

CHECK (SoLuong > 0),

CHECK (TongTien > 0),

CHECK (DonGia > 0)

);

***Bảng Nhà cung cấp***

CREATE TABLE NHA\_CUNG\_CAP (

MaNCC INT PRIMARY KEY IDENTITY(1,1),

TenNCC VARCHAR(100) NOT NULL,

DiaChi VARCHAR(100) NOT NULL,

SDT VARCHAR(10) NOT NULL,

CHECK (LEN(SDT) = 10)

);

***Bảng Đơn nhập***

CREATE TABLE DON\_NHAP (

MaDonNH INT PRIMARY KEY IDENTITY(1,1),

NgayNH DATE NOT NULL,

TriGiaDonNH DECIMAL(15, 2) NOT NULL,

MaNCC INT NOT NULL,

FOREIGN KEY (MaNCC) REFERENCES NHA\_CUNG\_CAP(MaNCC)

);

***Bảng Chi tiết đơn nhập hàng***

CREATE TABLE CHI\_TIET\_DON\_NHAP\_HANG (

MaDonNH INT,

MaSP INT,

DonGia DECIMAL(10, 2) NOT NULL,

SL INT NOT NULL,

TongTien DECIMAL(15, 2) NOT NULL,

NgaySanXuat DATE NOT NULL,

HanSuDung DATE NOT NULL,

PRIMARY KEY (MaDonNH, MaSP),

FOREIGN KEY (MaDonNH) REFERENCES DON\_NHAP(MaDonNH),

FOREIGN KEY (MaSP) REFERENCES SAN\_PHAM(MaSP),

CHECK (DonGia > 0),

CHECK (SL > 0),

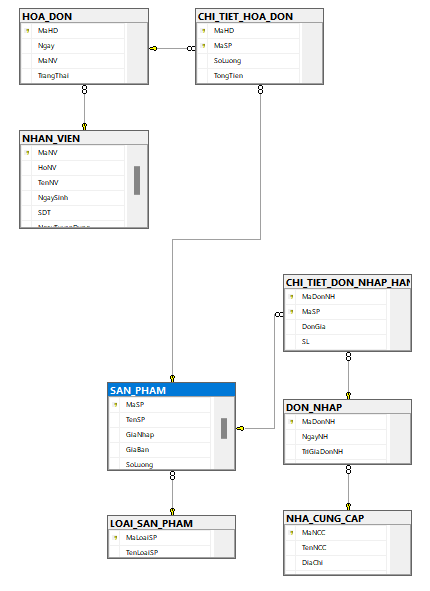
CHECK (TongTien > 0),

CHECK (NgaySanXuat < HanSuDung),

CHECK (HanSuDung > '1111-01-01')

);

1. **Database Diagrams.**



1. **Các trigger.**
   1. **Trigger mặc định trạng thái khi thêm hóa đơn là “Chưa thanh toán”.**

|  |
| --- |
| CREATE TRIGGER trg\_DefaultTrangThaiHoaDon  ON HOA\_DON  AFTER INSERT  AS  BEGIN  UPDATE HOA\_DON  SET TrangThai = 'Chưa thanh toán'  WHERE MaHD IN (SELECT MaHD FROM inserted)  AND TrangThai IS NULL;  END;  GO |

* 1. **Trigger thay đổi số lượng hàng khi bán được sản phẩm.**

|  |
| --- |
| CREATE TRIGGER trg\_UpdateSanPhamSoLuongOnPayment  ON HOA\_DON  AFTER UPDATE  AS  BEGIN  IF EXISTS (SELECT 1 FROM inserted WHERE TrangThai = 'Đã thanh toán')  BEGIN  DECLARE @MaHD INT;  SELECT @MaHD = MaHD FROM inserted;  UPDATE SAN\_PHAM  SET SAN\_PHAM.SoLuong = SAN\_PHAM.SoLuong - CTHD.SoLuong  FROM SAN\_PHAM  INNER JOIN CHI\_TIET\_HOA\_DON CTHD ON SAN\_PHAM.MaSP = CTHD.MaSP  WHERE CTHD.MaHD = @MaHD;  END  END;  GO |

* 1. **Trigger kiểm tra số lượng hàng hóa có đủ để bán hay không.**

|  |
| --- |
| CREATE TRIGGER trg\_CheckStockAfterInsertHoaDon  ON CHI\_TIET\_HOA\_DON  AFTER INSERT  AS  BEGIN  DECLARE @MaSP INT, @SoLuongMua INT, @SoLuongKho INT;  SELECT @MaSP = MaSP, @SoLuongMua = SoLuong FROM inserted;  SELECT @SoLuongKho = SoLuong FROM SAN\_PHAM WHERE MaSP = @MaSP;  IF @SoLuongKho < @SoLuongMua  BEGIN  RAISERROR('Không đủ số lượng sản phẩm trong kho cho mã sản phẩm %d.', 16, 1, @MaSP);  ROLLBACK TRANSACTION; -- Rollback the transaction to prevent invalid data  END  ELSE  BEGIN  UPDATE SAN\_PHAM  SET SoLuong = SoLuong - @SoLuongMua  WHERE MaSP = @MaSP;  PRINT 'Stock updated successfully for product ID: ' + CAST(@MaSP AS NVARCHAR);  END  END;  GO |

* 1. **Trigger bắt lỗi trùng tên khi thêm và sửa sản phẩm.**

|  |
| --- |
| CREATE TRIGGER trg\_CheckDuplicateProductNameOnInsert  ON SAN\_PHAM  INSTEAD OF INSERT  AS  BEGIN  DECLARE @TenSP VARCHAR(100);  SELECT @TenSP = inserted.TenSP FROM inserted;  IF EXISTS (SELECT 1 FROM SAN\_PHAM WHERE TenSP = @TenSP)  BEGIN  RAISERROR ('Tên sản phẩm đã tồn tại. Vui lòng chọn tên khác.', 16, 1);  END  ELSE  BEGIN  INSERT INTO SAN\_PHAM (TenSP, GiaNhap, GiaBan, SoLuong, MaLoaiSP)  SELECT TenSP, GiaNhap, GiaBan, SoLuong, MaLoaiSP  FROM inserted;  END  END;  GO CREATE TRIGGER trg\_CheckDuplicateProductNameOnUpdate  ON SAN\_PHAM  INSTEAD OF UPDATE  AS  BEGIN  DECLARE @TenSP VARCHAR(100), @MaSP INT;  SELECT @TenSP = inserted.TenSP, @MaSP = inserted.MaSP FROM inserted;  IF EXISTS (SELECT 1 FROM SAN\_PHAM WHERE TenSP = @TenSP AND MaSP <> @MaSP)  BEGIN  RAISERROR ('Tên sản phẩm đã tồn tại. Vui lòng chọn tên khác.', 16, 1);  END  ELSE  BEGIN  UPDATE SAN\_PHAM  SET TenSP = inserted.TenSP, GiaNhap = inserted.GiaNhap, GiaBan = inserted.GiaBan, SoLuong = inserted.SoLuong, MaLoaiSP = inserted.MaLoaiSP  FROM inserted  WHERE SAN\_PHAM.MaSP = inserted.MaSP;  END  END;  GO |

* 1. **Trigger bắt lỗi trùng tên khi thêm và sửa nhà cung cấp.**

|  |
| --- |
| CREATE TRIGGER trg\_CheckDuplicateNhaCungCap  ON NHA\_CUNG\_CAP  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT \*  FROM NHA\_CUNG\_CAP ncc  INNER JOIN inserted i ON ncc.TenNCC = i.TenNCC  GROUP BY ncc.TenNCC  HAVING COUNT(\*) > 1  )  BEGIN  RAISERROR ('Tên nhà cung cấp đã tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

* 1. **Trigger bắt lỗi khi trùng số điện thoại khi thêm nhân viên.**

|  |
| --- |
| CREATE TRIGGER KiemTraTrungSDT\_NV  ON NHAN\_VIEN  FOR INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM NHAN\_VIEN nv  INNER JOIN inserted i ON nv.SDT = i.SDT AND nv.MaNV <> i.MaNV)  BEGIN  RAISERROR ('Số điện thoại đã tồn tại trong hệ thống. Vui lòng kiểm tra lại.', 16, 1);  ROLLBACK TRANSACTION;  END  END;  GO |

**6.7 Trigger bắt lỗi khi trùng số điện thoại khi thêm nhà cung cấp.**

|  |
| --- |
| CREATE TRIGGER KiemTraTrungSDT\_NCC  ON NHA\_CUNG\_CAP  FOR INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM NHA\_CUNG\_CAP ncc  INNER JOIN inserted i ON ncc.SDT = i.SDT AND ncc.MaNCC <> i.MaNCC  )  BEGIN  RAISERROR ('Số điện thoại nhà cung cấp đã tồn tại trong hệ thống. Vui lòng kiểm tra lại.', 16, 1);  ROLLBACK TRANSACTION;  END  END;  GO |

1. **Một số transaction.**

**7.1 Procedure ThemHoaDon sử dụng transaction.**

|  |
| --- |
| CREATE PROCEDURE ThemHoaDon  @Ngay DATE,  @MaNV INT,  @ChiTietHoaDon NVARCHAR(MAX),  @MaHD INT OUTPUT  AS  BEGIN  BEGIN TRANSACTION;  INSERT INTO HOA\_DON (Ngay, MaNV, TriGiaHD)  VALUES (@Ngay, @MaNV, 0);  SET @MaHD = SCOPE\_IDENTITY();  DECLARE @MaSP INT, @SoLuong INT, @DonGia DECIMAL(10, 2), @TongTien DECIMAL(15, 2), @TriGiaHD DECIMAL(15, 2);  SET @TriGiaHD = 0; DECLARE @Pos INT = 1, @NextPos INT, @ChiTiet NVARCHAR(100), @Separator CHAR(1) = ';';  WHILE CHARINDEX(@Separator, @ChiTietHoaDon, @Pos) > 0  BEGIN  SET @NextPos = CHARINDEX(@Separator, @ChiTietHoaDon, @Pos);  SET @ChiTiet = SUBSTRING(@ChiTietHoaDon, @Pos, @NextPos - @Pos);  SET @Pos = @NextPos + 1;  SET @MaSP = CAST(SUBSTRING(@ChiTiet, 1, CHARINDEX(',', @ChiTiet) - 1) AS INT);  SET @SoLuong = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet) + 1, LEN(@ChiTiet)) AS INT);  SELECT @DonGia = GiaBan FROM SAN\_PHAM WHERE MaSP = @MaSP;  SET @TongTien = @DonGia \* @SoLuong;  INSERT INTO CHI\_TIET\_HOA\_DON (MaHD, MaSP, SoLuong, TongTien, DonGia)  VALUES (@MaHD, @MaSP, @SoLuong, @TongTien, @DonGia);  SET @TriGiaHD = @TriGiaHD + @TongTien;  END;  UPDATE HOA\_DON  SET TriGiaHD = @TriGiaHD  WHERE MaHD = @MaHD;  COMMIT TRANSACTION;  END;  GO |

**7.2 Procedure ThemDonNhapHang sử dụng transaction.**

|  |
| --- |
| CREATE PROCEDURE ThemDonNhapHang  @NgayNH DATE,  @MaNCC INT,  @ChiTietDonNhap NVARCHAR(MAX)  AS  BEGIN  BEGIN TRANSACTION;  INSERT INTO DON\_NHAP (NgayNH, MaNCC, TriGiaDonNH)  VALUES (@NgayNH, @MaNCC, 0);  DECLARE @MaDonNH INT;  SET @MaDonNH = SCOPE\_IDENTITY();  DECLARE @MaSP INT, @SoLuong INT, @DonGia DECIMAL(10, 2), @TongTien DECIMAL(15, 2), @TriGiaDonNH DECIMAL(15, 2);  DECLARE @NgaySanXuat DATE, @HanSuDung DATE;  SET @TriGiaDonNH = 0;  DECLARE @Pos INT = 1, @NextPos INT, @ChiTiet NVARCHAR(200), @Separator CHAR(1) = ';';  SET @ChiTietDonNhap = @ChiTietDonNhap + ';';  WHILE @Pos < LEN(@ChiTietDonNhap)  BEGIN  SET @NextPos = CHARINDEX(@Separator, @ChiTietDonNhap, @Pos);  SET @ChiTiet = SUBSTRING(@ChiTietDonNhap, @Pos, @NextPos - @Pos);  SET @Pos = @NextPos + 1;  SET @MaSP = CAST(SUBSTRING(@ChiTiet, 1, CHARINDEX(',', @ChiTiet) - 1) AS INT);  SET @SoLuong = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet) + 1,  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) - CHARINDEX(',', @ChiTiet) - 1) AS INT);  SET @DonGia = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1,  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) -  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) - 1) AS DECIMAL(10, 2));  SET @NgaySanXuat = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) + 1,  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) + 1) -  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) - 1) AS DATE);  SET @HanSuDung = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) + 1) + 1,  LEN(@ChiTiet)) AS DATE);  SET @TongTien = @DonGia \* @SoLuong;  INSERT INTO CHI\_TIET\_DON\_NHAP\_HANG (MaDonNH, MaSP, DonGia, SL, TongTien, NgaySanXuat, HanSuDung)  VALUES (@MaDonNH, @MaSP, @DonGia, @SoLuong, @TongTien, @NgaySanXuat, @HanSuDung);  SET @TriGiaDonNH = @TriGiaDonNH + @TongTien;  END; |

**7.3 Trigger trg\_CheckDuplicateNhaCungCap sử dụng transaction.**

|  |
| --- |
| CREATE TRIGGER trg\_CheckDuplicateNhaCungCap  ON NHA\_CUNG\_CAP  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT \*  FROM NHA\_CUNG\_CAP ncc  INNER JOIN inserted i ON ncc.TenNCC = i.TenNCC  GROUP BY ncc.TenNCC  HAVING COUNT(\*) > 1  )  BEGIN  RAISERROR ('Tên nhà cung cấp đã tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

**7.4 Trigger KiemTraTrungSDT\_NV sử dụng transaction.**

|  |
| --- |
| CREATE TRIGGER KiemTraTrungSDT\_NV  ON NHAN\_VIEN  FOR INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM NHAN\_VIEN nv  INNER JOIN inserted i ON nv.SDT = i.SDT AND nv.MaNV <> i.MaNV)  BEGIN  RAISERROR ('Số điện thoại đã tồn tại trong hệ thống. Vui lòng kiểm tra lại.', 16, 1);  ROLLBACK TRANSACTION;  END  END;  GO |

**7.5 Trigger KiemTraTrungSDT\_NCC sử dụng transaction.**

|  |
| --- |
| CREATE TRIGGER KiemTraTrungSDT\_NCC  ON NHA\_CUNG\_CAP  FOR INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM NHA\_CUNG\_CAP ncc  INNER JOIN inserted i ON ncc.SDT = i.SDT AND ncc.MaNCC <> i.MaNCC  )  BEGIN  RAISERROR ('Số điện thoại nhà cung cấp đã tồn tại trong hệ thống. Vui lòng kiểm tra lại.', 16, 1);  ROLLBACK TRANSACTION;  END  END;  GO |

1. **Index.**

Tạo index cho NHAN\_VIEN và SAN\_PHAM để tăng tốc độ tìm kiếm (nếu bộ dữ liệu lớn).

|  |
| --- |
| CREATE INDEX IDX\_NHAN\_VIEN\_HOTEN ON NHAN\_VIEN(HoNV, TenNV);  CREATE INDEX IDX\_SAN\_PHAM\_TenSP ON SAN\_PHAM(TenSP); |

**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

1. **Kết nối cơ sở dữ liệu.**

|  |
| --- |
| internal class MY\_DB  {  SqlConnection con = new SqlConnection(@"Data Source=localhost\SQLEXPRESS;Initial Catalog=QuanLyBanHang;User ID="+GlobalVariables.LoggedInUsername+";Password="+GlobalVariables.LoggedInPassword+";TrustServerCertificate=True");  public SqlConnection getConnection  {  get  {  return con;  }  }  public void openConnection()  {  if ((con.State == ConnectionState.Closed))  {  con.Open();  }  }  public void closeConnection()  {  if ((con.State == ConnectionState.Open))  {  con.Close();  }  }  } |

1. **Xem thông tin các danh mục**

**2.1 Xem danh sách nhân viên.**

|  |
| --- |
| CREATE VIEW v\_DanhMucNhanVien AS  SELECT  MaNV,  CONCAT(HoNV, ' ', TenNV) AS HoVaTen,  SDT,  NgaySinh,  NgayTuyenDung,  DiaChi,  HinhAnh  FROM  NHAN\_VIEN;  GO |

Thực thi trên C#:

|  |
| --- |
| private void LoadNhanVienData()  {  MY\_DB db = new MY\_DB();  try  {  db.openConnection();  SqlCommand command = new SqlCommand("SELECT \* FROM v\_DanhMucNhanVien", db.getConnection); // Use the existing connection from MY\_DB  SqlDataAdapter adapter = new SqlDataAdapter(command);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView1.DataSource = dataTable;  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.closeConnection();  }  } |

**2.2 Xem danh sách các hóa đơn.**

|  |
| --- |
| CREATE VIEW v\_DanhMucHoaDon AS  SELECT hd.MaHD, hd.Ngay, hd.MaNV, hd.TriGiaHD  FROM HOA\_DON hd  GO |

Thực thi trên C#:

|  |
| --- |
| private void LoadDanhMucHoaDon()  {  MY\_DB db = new MY\_DB();  try  {  db.openConnection();  string query = "SELECT \* FROM v\_DanhMucHoaDon";  SqlDataAdapter adapter = new SqlDataAdapter(query, db.getConnection);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView1.DataSource = dataTable;  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tải dữ liệu: " + ex.Message);  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.closeConnection();  }  } |

**2.3 Xem danh sách các sản phẩm.**

|  |
| --- |
| CREATE VIEW v\_DanhMucSanPham AS  SELECT  SP.MaSP,SP.TenSP,SP.GiaNhap,SP.GiaBan,SP.SoLuong,LSP.TenLoaiSP,SP.HinhAnh  FROM  SAN\_PHAM SP  JOIN  LOAI\_SAN\_PHAM LSP ON SP.MaLoaiSP = LSP.MaLoaiSP;  GO |

Thực thi trên C#:

|  |
| --- |
| private void LoadProductList()  {  MY\_DB db = new MY\_DB();  try  {  db.openConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_DanhMucSanPham", db.getConnection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  foreach (DataRow row in dataTable.Rows)  {  if (row["HinhAnh"] != DBNull.Value)  {  byte[] imageData = (byte[])row["HinhAnh"];  using (MemoryStream ms = new MemoryStream(imageData))  {  row["HinhAnh"] = Image.FromStream(ms);  }  }  }  dataGridView1.DataSource = dataTable;  dataGridView1.Columns["MaSP"].HeaderText = "Mã Sản Phẩm";  dataGridView1.Columns["TenSP"].HeaderText = "Tên Sản Phẩm";  dataGridView1.Columns["GiaNhap"].HeaderText = "Giá Nhập";  dataGridView1.Columns["GiaBan"].HeaderText = "Giá Bán";  dataGridView1.Columns["SoLuong"].HeaderText = "Số Lượng";  dataGridView1.Columns["TenLoaiSP"].HeaderText = "Tên Loại Sản Phẩm";  if (!dataGridView1.Columns.Contains("HinhAnh"))  {  DataGridViewImageColumn imageColumn = new DataGridViewImageColumn  {  Name = "HinhAnh",  HeaderText = "Hình Ảnh",  DataPropertyName = "HinhAnh",  ImageLayout = DataGridViewImageCellLayout.Zoom // Adjust image display style  };  dataGridView1.Columns.Add(imageColumn);  }  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tải dữ liệu: " + ex.Message);  }  finally  {  db.closeConnection(); // Ensure the connection is closed  }  } |

**2.4 Xem danh sách các loại sản phẩm.**

|  |
| --- |
| CREATE VIEW v\_DanhMucLoaiSanPham  AS  SELECT  MaLoaiSP,  TenLoaiSP  FROM LOAI\_SAN\_PHAM  GO |

Thực thi trên C#:

|  |
| --- |
| private void LoadDanhMucLoaiSanPham()  {  MY\_DB db = new MY\_DB();  try  {  db.openConnection(); // Open the connection using MY\_DB  string query = "SELECT \* FROM v\_DanhMucLoaiSanPham";  SqlDataAdapter adapter = new SqlDataAdapter(query, db.getConnection);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView1.DataSource = dataTable;  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tải dữ liệu: " + ex.Message);  }  finally  {  db.closeConnection(); // Ensure the connection is closed  }  } |

**2.5 Xem danh sách đơn nhập hàng.**

|  |
| --- |
| CREATE VIEW v\_DanhMucDonNhapHang AS  SELECT DN.MaDonNH, DN.NgayNH, DN.TriGiaDonNH, NCC.TenNCC  FROM DON\_NHAP DN  JOIN NHA\_CUNG\_CAP NCC ON DN.MaNCC = NCC.MaNCC; |

Thực thi trên C#:

|  |
| --- |
| private void LoadReceiptList()  {  MY\_DB db = new MY\_DB();  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_DanhMucDonNhapHang", db.getConnection);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView1.DataSource = dataTable;  dataGridView1.Columns["MaDonNH"].HeaderText = "Mã Đơn Nhập Hàng";  dataGridView1.Columns["NgayNH"].HeaderText = "Ngày Nhập Hàng";  dataGridView1.Columns["TriGiaDonNH"].HeaderText = "Trị Giá Đơn Nhập Hàng";  dataGridView1.Columns["TenNCC"].HeaderText = "Tên Nhà Cung Cấp";  } |

**2.6 Xem danh sách nhà cung cấp.**

|  |
| --- |
| CREATE VIEW v\_DanhMucNhaCungCap AS  SELECT  NCC.MaNCC,  NCC.TenNCC,  NCC.DiaChi,  NCC.SDT,  COUNT(DN.MaDonNH) AS SoDonNhapHang  FROM  NHA\_CUNG\_CAP NCC  LEFT JOIN  DON\_NHAP DN ON NCC.MaNCC = DN.MaNCC  GROUP BY  NCC.MaNCC,  NCC.TenNCC,  NCC.DiaChi,  NCC.SDT; |

Thực thi trên C#:

|  |
| --- |
| private void LoadNhaCungCap()  {  db.openConnection();  // Tạo SqlCommand để truy vấn view  SqlCommand cmd = new SqlCommand("SELECT \* FROM v\_DanhMucNhaCungCap", db.getConnection);  // Tạo SqlDataAdapter và DataTable để lấy kết quả  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  try  {  da.Fill(dt);  if (dt.Rows.Count > 0)  {  dataGridView1.DataSource = dt;  }  else  {  MessageBox.Show("Không có nhà cung cấp nào để hiển thị.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  catch (SqlException sqlEx)  {  MessageBox.Show("Lỗi SQL: " + sqlEx.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.closeConnection();  }  } |

1. **Quản lý nhân viên.**
   1. **Tìm kiếm nhân viên.**

|  |
| --- |
| CREATE FUNCTION fn\_TimKiemNhanVien  (  @TenNV VARCHAR(50)  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM NHAN\_VIEN  WHERE TenNV LIKE '%' + @TenNV + '%'  ); |

Thực thi trên C#:

|  |
| --- |
| private void buttonSearch\_Click(object sender, EventArgs e)  {  string tenNV = textBoxSearch.Text;  db.openConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM fn\_TimKiemNhanVien(@TenNV)", db.getConnection);  cmd.Parameters.AddWithValue("@TenNV", tenNV);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView\_NhanVien.DataSource = dataTable;  db.closeConnection();  } |

* 1. **Thêm nhân viên:**

|  |
| --- |
| CREATE PROCEDURE sp\_ThemNhanVien  @HoNV VARCHAR(50),  @TenNV VARCHAR(50),  @NgaySinh DATE,  @SDT VARCHAR(10),  @NgayTuyenDung DATE,  @DiaChi VARCHAR(100),  @HinhAnh IMAGE  AS  BEGIN  INSERT INTO NHAN\_VIEN (HoNV, TenNV, NgaySinh, SDT, NgayTuyenDung, DiaChi, HinhAnh)  VALUES (@HoNV, @TenNV, @NgaySinh, @SDT, @NgayTuyenDung, @DiaChi, @HinhAnh);  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonAdd\_Click(object sender, EventArgs e)  {  string hoNV = textBoxHoNV.Text;  string tenNV = textBoxTenNV.Text;  DateTime ngaySinh = dateTimePickerNgaySinh.Value;  string sdt = textBoxSDT.Text;  DateTime ngayTuyenDung = dateTimePickerNgayTuyenDung.Value;  string diaChi = textBoxDiaChi.Text;  byte[] hinhAnhData = null;  if (pictureBoxHinhAnh.Image != null)  {  using (MemoryStream ms = new MemoryStream())  {  pictureBoxHinhAnh.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg); // Adjust format if necessary  hinhAnhData = ms.ToArray();  }  }  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_ThemNhanVien", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@HoNV", SqlDbType.VarChar).Value = hoNV;  cmd.Parameters.Add("@TenNV", SqlDbType.VarChar).Value = tenNV;  cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = ngaySinh;  cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = sdt;  cmd.Parameters.Add("@NgayTuyenDung", SqlDbType.Date).Value = ngayTuyenDung;  cmd.Parameters.Add("@DiaChi", SqlDbType.VarChar).Value = diaChi;  if (hinhAnhData != null)  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = hinhAnhData;  }  else  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = DBNull.Value; // Handle null image case  }  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm nhân viên thành công!", "Thêm Nhân Viên", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Thêm nhân viên thất bại.", "Thêm Nhân Viên", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

**3.3 Sửa nhân viên.**

|  |
| --- |
| CREATE PROCEDURE sp\_SuaNhanVien  @MaNV INT,  @HoNV VARCHAR(50),  @TenNV VARCHAR(50),  @NgaySinh DATE,  @SDT VARCHAR(10),  @NgayTuyenDung DATE,  @DiaChi VARCHAR(100),  @HinhAnh IMAGE  AS  BEGIN  UPDATE NHAN\_VIEN  SET  HoNV = @HoNV,  TenNV = @TenNV,  NgaySinh = @NgaySinh,  SDT = @SDT,  NgayTuyenDung = @NgayTuyenDung,  DiaChi = @DiaChi,  HinhAnh = @HinhAnh  WHERE MaNV = @MaNV;  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonUpdate\_Click(object sender, EventArgs e)  {  int maNV = Convert.ToInt32(textBoxMaNV.Text);  string hoNV = textBoxHoNV.Text;  string tenNV = textBoxTenNV.Text;  DateTime ngaySinh = dateTimePickerNgaySinh.Value;  string sdt = textBoxSDT.Text;  DateTime ngayTuyenDung = dateTimePickerNgayTuyenDung.Value;  string diaChi = textBoxDiaChi.Text;  byte[] hinhAnhData = null;  if (pictureBoxHinhAnh.Image != null)  {  using (MemoryStream ms = new MemoryStream())  {  pictureBoxHinhAnh.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg); // Adjust format if necessary  hinhAnhData = ms.ToArray();  }  }  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_SuaNhanVien", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaNV", SqlDbType.Int).Value = maNV;  cmd.Parameters.Add("@HoNV", SqlDbType.VarChar).Value = hoNV;  cmd.Parameters.Add("@TenNV", SqlDbType.VarChar).Value = tenNV;  cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = ngaySinh;  cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = sdt;  cmd.Parameters.Add("@NgayTuyenDung", SqlDbType.Date).Value = ngayTuyenDung;  cmd.Parameters.Add("@DiaChi", SqlDbType.VarChar).Value = diaChi;  if (hinhAnhData != null)  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = hinhAnhData;  }  else  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = DBNull.Value; // Handle null image case  }  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Cập nhật nhân viên thành công!", "Cập Nhật Nhân Viên", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Cập nhật nhân viên thất bại.", "Cập Nhật Nhân Viên", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

1. **Quản lý hóa đơn.**
   1. **Thêm hóa đơn.**

|  |
| --- |
| CREATE PROCEDURE ThemHoaDon  @Ngay DATE,  @MaNV INT,  @ChiTietHoaDon NVARCHAR(MAX),  @MaHD INT OUTPUT  AS  BEGIN  BEGIN TRANSACTION;  INSERT INTO HOA\_DON (Ngay, MaNV, TriGiaHD)  VALUES (@Ngay, @MaNV, 0);  SET @MaHD = SCOPE\_IDENTITY();  DECLARE @MaSP INT, @SoLuong INT, @DonGia DECIMAL(10, 2), @TongTien DECIMAL(15, 2), @TriGiaHD DECIMAL(15, 2);  SET @TriGiaHD = 0; DECLARE @Pos INT = 1, @NextPos INT, @ChiTiet NVARCHAR(100), @Separator CHAR(1) = ';';  WHILE CHARINDEX(@Separator, @ChiTietHoaDon, @Pos) > 0  BEGIN  SET @NextPos = CHARINDEX(@Separator, @ChiTietHoaDon, @Pos);  SET @ChiTiet = SUBSTRING(@ChiTietHoaDon, @Pos, @NextPos - @Pos);  SET @Pos = @NextPos + 1;  SET @MaSP = CAST(SUBSTRING(@ChiTiet, 1, CHARINDEX(',', @ChiTiet) - 1) AS INT);  SET @SoLuong = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet) + 1, LEN(@ChiTiet)) AS INT);  SELECT @DonGia = GiaBan FROM SAN\_PHAM WHERE MaSP = @MaSP;  SET @TongTien = @DonGia \* @SoLuong;  INSERT INTO CHI\_TIET\_HOA\_DON (MaHD, MaSP, SoLuong, TongTien, DonGia)  VALUES (@MaHD, @MaSP, @SoLuong, @TongTien, @DonGia);  SET @TriGiaHD = @TriGiaHD + @TongTien;  END;  UPDATE HOA\_DON  SET TriGiaHD = @TriGiaHD  WHERE MaHD = @MaHD;  COMMIT TRANSACTION;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| private void btnThemHoaDon\_Click(object sender, EventArgs e)  {  DateTime ngay = dtpNgay.Value;  int maNV = int.Parse(txtMaNV.Text);  string chiTietHoaDon = txtChiTietHoaDon.Text;  using (SqlConnection connection = new SqlConnection(connectionString))  {  try  {  connection.Open();  SqlCommand command = new SqlCommand("ThemHoaDon", connection);  command.CommandType = System.Data.CommandType.StoredProcedure;  command.Parameters.AddWithValue("@Ngay", ngay);  command.Parameters.AddWithValue("@MaNV", maNV);  command.Parameters.AddWithValue("@ChiTietHoaDon", chiTietHoaDon);  command.ExecuteNonQuery();  MessageBox.Show("Hóa đơn đã được thêm thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  } |

* 1. **Hủy hóa đơn.**

|  |
| --- |
| CREATE PROCEDURE sp\_HuyHoaDon  @MaHD INT  AS  BEGIN  IF EXISTS (SELECT 1 FROM HOA\_DON WHERE MaHD = @MaHD AND TrangThai = 'Chưa thanh toán')  BEGIN  UPDATE HOA\_DON  SET TrangThai = 'Đã hủy'  WHERE MaHD = @MaHD;  PRINT 'Hóa đơn đã được hủy.';  END  ELSE  BEGIN  PRINT 'Hóa đơn không thể hủy, do đã thanh toán hoặc đã bị hủy.';  END  END;  GO |

Thực thi trên C#:

|  |
| --- |
| static void HuyHoaDon(int maHD)  {  using (SqlConnection conn = MY\_DB.getConnection())  {  try  {  conn.Open();  using (SqlCommand cmd = new SqlCommand("sp\_HuyHoaDon", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaHD", maHD);  cmd.ExecuteNonQuery();  Console.WriteLine("Hóa đơn đã được xử lý.");  }  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  Console.WriteLine("Error: " + ex.Message);  }  finally  {  MY\_DB.closeConnection(conn);  }  }  } |

* 1. **Xác nhận đã thanh toán hóa đơn.**

|  |
| --- |
| CREATE PROCEDURE sp\_XacNhanThanhToan  @MaHD INT  AS  BEGIN  IF EXISTS (SELECT 1 FROM HOA\_DON WHERE MaHD = @MaHD AND TrangThai = 'Chưa thanh toán')  BEGIN  UPDATE HOA\_DON  SET TrangThai = 'Đã thanh toán'  WHERE MaHD = @MaHD;  PRINT 'Hóa đơn đã được xác nhận thanh toán.';  END  ELSE  BEGIN  PRINT 'Hóa đơn không thể thanh toán, do đã thanh toán hoặc đã bị hủy.';  END  END;  GO |

Thực thi trên C#:

|  |
| --- |
| static void XacNhanThanhToan(int maHD)  {  using (SqlConnection conn = MY\_DB.getConnection())  {  try  {  conn.Open();  using (SqlCommand cmd = new SqlCommand("sp\_XacNhanThanhToan", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaHD", maHD);  cmd.ExecuteNonQuery();  Console.WriteLine("Thanh toán đã được xác nhận.");  }  }  catch (Exception ex)  {  Console.WriteLine("Error: " + ex.Message);  }  finally  {  MY\_DB.closeConnection(conn);  }  }  } |

* 1. **Tìm kiếm hóa đơn.**

|  |
| --- |
| CREATE PROCEDURE sp\_TimKiemHoaDon  @MaHD INT = NULL,  @Ngay DATE = NULL,  @MaNV INT = NULL,  @TrangThai NVARCHAR(50) = NULL  AS  BEGIN  SELECT MaHD, Ngay, MaNV, TrangThai, TriGiaHD  FROM HOA\_DON  WHERE (@MaHD IS NULL OR MaHD = @MaHD)  AND (@Ngay IS NULL OR Ngay = @Ngay)  AND (@MaNV IS NULL OR MaNV = @MaNV)  AND (@TrangThai IS NULL OR TrangThai = @TrangThai);  END;  GO |

Thực thi trên C#:

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e)  {  // Lấy dữ liệu từ các control  int? maHD = string.IsNullOrEmpty(txtMaHD.Text) ? (int?)null : int.Parse(txtMaHD.Text);  int? maNV = string.IsNullOrEmpty(txtMaNV.Text) ? (int?)null : int.Parse(txtMaNV.Text);  string trangThai = string.IsNullOrEmpty(txtTrangThai.Text) ? null : txtTrangThai.Text;  DateTime? ngay = dtpNgay.Checked ? dtpNgay.Value.Date : (DateTime?)null;  using (SqlConnection connection = new SqlConnection(connectionString))  {  try  {  connection.Open(); // Mở kết nối  SqlCommand command = new SqlCommand("sp\_TimKiemHoaDon", connection);  command.CommandType = CommandType.StoredProcedure;  // Thêm các tham số cho stored procedure  command.Parameters.AddWithValue("@MaHD", (object)maHD ?? DBNull.Value);  command.Parameters.AddWithValue("@Ngay", (object)ngay ?? DBNull.Value);  command.Parameters.AddWithValue("@MaNV", (object)maNV ?? DBNull.Value);  command.Parameters.AddWithValue("@TrangThai", (object)trangThai ?? DBNull.Value);  SqlDataAdapter adapter = new SqlDataAdapter(command);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dgvHoaDon.DataSource = dataTable;  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  } |

1. **Quản lý đơn nhập hàng.**

**Thêm đơn nhập hàng.**

|  |
| --- |
| CREATE PROCEDURE ThemDonNhapHang  @NgayNH DATE,  @MaNCC INT,  @ChiTietDonNhap NVARCHAR(MAX)  AS  BEGIN  BEGIN TRANSACTION;  INSERT INTO DON\_NHAP (NgayNH, MaNCC, TriGiaDonNH)  VALUES (@NgayNH, @MaNCC, 0);  DECLARE @MaDonNH INT;  SET @MaDonNH = SCOPE\_IDENTITY();  DECLARE @MaSP INT, @SoLuong INT, @DonGia DECIMAL(10, 2), @TongTien DECIMAL(15, 2), @TriGiaDonNH DECIMAL(15, 2);  DECLARE @NgaySanXuat DATE, @HanSuDung DATE;  SET @TriGiaDonNH = 0;  DECLARE @Pos INT = 1, @NextPos INT, @ChiTiet NVARCHAR(200), @Separator CHAR(1) = ';';  SET @ChiTietDonNhap = @ChiTietDonNhap + ';';  WHILE @Pos < LEN(@ChiTietDonNhap)  BEGIN  SET @NextPos = CHARINDEX(@Separator, @ChiTietDonNhap, @Pos);  SET @ChiTiet = SUBSTRING(@ChiTietDonNhap, @Pos, @NextPos - @Pos);  SET @Pos = @NextPos + 1;  SET @MaSP = CAST(SUBSTRING(@ChiTiet, 1, CHARINDEX(',', @ChiTiet) - 1) AS INT);  SET @SoLuong = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet) + 1,  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) - CHARINDEX(',', @ChiTiet) - 1) AS INT);  SET @DonGia = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1,  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) -  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) - 1) AS DECIMAL(10, 2));  SET @NgaySanXuat = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) + 1,  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) + 1) -  CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) - 1) AS DATE);  SET @HanSuDung = CAST(SUBSTRING(@ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet, CHARINDEX(',', @ChiTiet) + 1) + 1) + 1) + 1,  LEN(@ChiTiet)) AS DATE);  SET @TongTien = @DonGia \* @SoLuong;  INSERT INTO CHI\_TIET\_DON\_NHAP\_HANG (MaDonNH, MaSP, DonGia, SL, TongTien, NgaySanXuat, HanSuDung)  VALUES (@MaDonNH, @MaSP, @DonGia, @SoLuong, @TongTien, @NgaySanXuat, @HanSuDung);  SET @TriGiaDonNH = @TriGiaDonNH + @TongTien;  END;  UPDATE DON\_NHAP  SET TriGiaDonNH = @TriGiaDonNH  WHERE MaDonNH = @MaDonNH;  COMMIT TRANSACTION;  END;  GO |

Thực thi trên C#:

|  |
| --- |
| private void ThemDonNhap\_Click(object sender, EventArgs e)  {  DateTime ngayNhap = dateTimePicker1.Value;  int maNCC;  if (!int.TryParse(textBox1.Text, out maNCC))  {  MessageBox.Show("Mã nhà cung cấp phải là số nguyên.");  return;  }  string chiTietDonNhap = "";  foreach (DataGridViewRow row in dataGridView1.Rows)  {  if (row.Cells["MaSanPham"].Value != null &&  row.Cells["SoLuong"].Value != null &&  row.Cells["GiaNhap"].Value != null &&  row.Cells["NgaySanXuat"].Value != null &&  row.Cells["HanSuDung"].Value != null)  {  int maSP = Convert.ToInt32(row.Cells["MaSanPham"].Value);  int soLuong = Convert.ToInt32(row.Cells["SoLuong"].Value);  decimal giaNhap = Convert.ToDecimal(row.Cells["GiaNhap"].Value);  DateTime ngaySanXuat = Convert.ToDateTime(row.Cells["NgaySanXuat"].Value);  DateTime hanSuDung = Convert.ToDateTime(row.Cells["HanSuDung"].Value);  chiTietDonNhap += $"{maSP},{soLuong},{giaNhap},{ngaySanXuat:yyyy-MM-dd},{hanSuDung:yyyy-MM-dd};";  }  }  if (string.IsNullOrEmpty(chiTietDonNhap))  {  MessageBox.Show("Vui lòng nhập ít nhất một sản phẩm.");  return;  }  MY\_DB db = new MY\_DB();  SqlConnection conn = db.getConnection;  try  {  db.openConnection(); // Mở kết nối  using (SqlCommand cmd = new SqlCommand("ThemDonNhapHang", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@NgayNH", ngayNhap);  cmd.Parameters.AddWithValue("@MaNCC", maNCC);  cmd.Parameters.AddWithValue("@ChiTietDonNhap", chiTietDonNhap);  cmd.ExecuteNonQuery();  }  MessageBox.Show("Thêm đơn nhập hàng thành công.");  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi thêm đơn nhập hàng: " + ex.Message);  }  finally  {  db.closeConnection(); // Đóng kết nối  }  } |

1. **Quản lý sản phẩm.**
   1. **Thêm sản phẩm.**

|  |
| --- |
| CREATE PROCEDURE sp\_ThemSanPham  @MaLoaiSP INT,  @TenSP VARCHAR(100),  @GiaNhap DECIMAL(10, 2),  @GiaBan DECIMAL(10, 2),  @SoLuong INT,  @HinhAnh IMAGE  AS  BEGIN  INSERT INTO SAN\_PHAM (MaLoaiSP, TenSP, GiaNhap, GiaBan, SoLuong, HinhAnh)  VALUES (@MaLoaiSP, @TenSP, @GiaNhap, @GiaBan, @SoLuong, @HinhAnh);  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonAdd\_Click(object sender, EventArgs e)  {  int maLoaiSP = Convert.ToInt32(textBoxMaLoaiSP.Text);  string tenSP = textBoxTenSP.Text;  decimal giaNhap = Convert.ToDecimal(textBoxGiaNhap.Text);  decimal giaBan = Convert.ToDecimal(textBoxGiaBan.Text);  int soLuong = Convert.ToInt32(textBoxSoLuong.Text);  byte[] hinhAnhData = null;  if (pictureBoxHinhAnh.Image != null)  {  using (MemoryStream ms = new MemoryStream())  {  pictureBoxHinhAnh.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg); // Adjust format if necessary  hinhAnhData = ms.ToArray();  }  }  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_ThemSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaLoaiSP", SqlDbType.Int).Value = maLoaiSP;  cmd.Parameters.Add("@TenSP", SqlDbType.VarChar).Value = tenSP;  cmd.Parameters.Add("@GiaNhap", SqlDbType.Decimal).Value = giaNhap;  cmd.Parameters.Add("@GiaBan", SqlDbType.Decimal).Value = giaBan;  cmd.Parameters.Add("@SoLuong", SqlDbType.Int).Value = soLuong;  if (hinhAnhData != null)  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = hinhAnhData;  }  else  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = DBNull.Value; // Handle null image case  }  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm sản phẩm thành công!", "Thêm Sản Phẩm", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Thêm sản phẩm thất bại.", "Thêm Sản Phẩm", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

* 1. **Xóa sản phẩm.**

|  |
| --- |
| CREATE PROCEDURE sp\_XoaSanPham  @MaSP INT  AS  BEGIN  DELETE FROM SAN\_PHAM WHERE MaSP = @MaSP;  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonDelete\_Click(object sender, EventArgs e)  {  int maSP = Convert.ToInt32(textBoxMaSP.Text);  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_XoaSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaSP", SqlDbType.Int).Value = maSP;  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Xóa sản phẩm thành công!", "Xóa Sản Phẩm", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Xóa sản phẩm thất bại.", "Xóa Sản Phẩm", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

* 1. **Sửa sản phẩm.**

|  |
| --- |
| CREATE PROCEDURE sp\_SuaSanPham  @MaSP INT,  @MaLoaiSP INT,  @TenSP VARCHAR(100),  @GiaNhap DECIMAL(10, 2),  @GiaBan DECIMAL(10, 2),  @SoLuong INT,  @HinhAnh IMAGE  AS  BEGIN  UPDATE SAN\_PHAM  SET  MaLoaiSP = @MaLoaiSP,  TenSP = @TenSP,  GiaNhap = @GiaNhap,  GiaBan = @GiaBan,  SoLuong = @SoLuong,  HinhAnh = @HinhAnh  WHERE MaSP = @MaSP;  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonUpdate\_Click(object sender, EventArgs e)  {  int maSP = Convert.ToInt32(textBoxMaSP.Text); // Mã sản phẩm  int maLoaiSP = Convert.ToInt32(textBoxMaLoaiSP.Text);  string tenSP = textBoxTenSP.Text;  decimal giaNhap = Convert.ToDecimal(textBoxGiaNhap.Text);  decimal giaBan = Convert.ToDecimal(textBoxGiaBan.Text);  int soLuong = Convert.ToInt32(textBoxSoLuong.Text);  byte[] hinhAnhData = null;  if (pictureBoxHinhAnh.Image != null)  {  using (MemoryStream ms = new MemoryStream())  {  pictureBoxHinhAnh.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg); // Adjust format if necessary  hinhAnhData = ms.ToArray();  }  }  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_SuaSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaSP", SqlDbType.Int).Value = maSP;  cmd.Parameters.Add("@MaLoaiSP", SqlDbType.Int).Value = maLoaiSP;  cmd.Parameters.Add("@TenSP", SqlDbType.VarChar).Value = tenSP;  cmd.Parameters.Add("@GiaNhap", SqlDbType.Decimal).Value = giaNhap;  cmd.Parameters.Add("@GiaBan", SqlDbType.Decimal).Value = giaBan;  cmd.Parameters.Add("@SoLuong", SqlDbType.Int).Value = soLuong;  if (hinhAnhData != null)  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = hinhAnhData;  }  else  {  cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = DBNull.Value; // Handle null image case  }  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Cập nhật sản phẩm thành công!", "Cập Nhật Sản Phẩm", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Cập nhật sản phẩm thất bại.", "Cập Nhật Sản Phẩm", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

* 1. **Tìm kiếm sản phẩm.**

|  |
| --- |
| CREATE FUNCTION fn\_TimKiemSanPham (  @TenSP VARCHAR(100)  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM SAN\_PHAM  WHERE TenSP LIKE '%' + @TenSP + '%'  ); |

Thực thi trên C#:

|  |
| --- |
| private void buttonSearch\_Click(object sender, EventArgs e)  {  string tenSP = textBoxSearch.Text;  db.openConnection();  SqlCommand cmd = new SqlCommand("SELECT \* FROM fn\_TimKiemSanPham(@TenSP)", db.getConnection);  cmd.Parameters.Add("@TenSP", SqlDbType.VarChar).Value = tenSP;  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  dataGridView\_SanPham.DataSource = dataTable;  db.closeConnection();  } |

1. **Quản lý loại sản phẩm.**
   1. **Thêm loại sản phẩm.**

|  |
| --- |
| CREATE PROCEDURE sp\_ThemLoaiSanPham  @TenLoaiSP VARCHAR(50)  AS  BEGIN  INSERT INTO LOAI\_SAN\_PHAM (TenLoaiSP)  VALUES (@TenLoaiSP)  PRINT 'Thêm loại sản phẩm thành công!'  END |

Thực thi trên C#:

|  |
| --- |
| private void buttonAddLoaiSP\_Click(object sender, EventArgs e)  {  if (string.IsNullOrEmpty(textBoxTenLoaiSP.Text))  {  MessageBox.Show("Vui lòng nhập tên loại sản phẩm.", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  string tenLoaiSP = textBoxTenLoaiSP.Text;  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_ThemLoaiSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@TenLoaiSP", SqlDbType.NVarChar).Value = tenLoaiSP;  try  {  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  MessageBox.Show("Thêm loại sản phẩm thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  textBoxTenLoaiSP.Clear();  }  else  {  MessageBox.Show("Không thể thêm loại sản phẩm. Vui lòng thử lại.", "Thất bại", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.closeConnection();  }  } |

* 1. **Sửa loại sản phẩm.**

|  |
| --- |
| CREATE PROCEDURE sp\_SuaLoaiSanPham  @MaLoaiSP INT,  @TenLoaiSP VARCHAR(50)  AS  BEGIN  IF EXISTS (SELECT 1 FROM LOAI\_SAN\_PHAM WHERE MaLoaiSP = @MaLoaiSP)  BEGIN  UPDATE LOAI\_SAN\_PHAM  SET TenLoaiSP = @TenLoaiSP  WHERE MaLoaiSP = @MaLoaiSP  PRINT 'Sửa loại sản phẩm thành công!'  END  ELSE  BEGIN  PRINT 'Mã loại sản phẩm không tồn tại!'  END  END |

Thực thi trên C#:

|  |
| --- |
| private void buttonUpdateLoaiSP\_Click(object sender, EventArgs e)  {  if (string.IsNullOrEmpty(textBoxMaLoaiSP.Text) || string.IsNullOrEmpty(textBoxTenLoaiSP.Text))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin.", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  int maLoaiSP;  if (!int.TryParse(textBoxMaLoaiSP.Text, out maLoaiSP))  {  MessageBox.Show("Mã loại sản phẩm phải là số.", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  string tenLoaiSP = textBoxTenLoaiSP.Text;  db.openConnection();  // Tạo SqlCommand để gọi stored procedure sửa loại sản phẩm  SqlCommand cmd = new SqlCommand("sp\_SuaLoaiSanPham", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaLoaiSP", SqlDbType.Int).Value = maLoaiSP;  cmd.Parameters.Add("@TenLoaiSP", SqlDbType.NVarChar).Value = tenLoaiSP;  try  {  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  MessageBox.Show("Cập nhật loại sản phẩm thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Không tìm thấy loại sản phẩm với mã tương ứng.", "Cập nhật thất bại", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  db.closeConnection();  }  } |

1. **Quản lý nhà cung cấp.**
   1. **Thêm nhà cung cấp.**

|  |
| --- |
| -- function or procedure  -- Thêm, xóa, sửa nhà cung cấp.  CREATE PROCEDURE sp\_ThemNhaCungCap  @TenNCC VARCHAR(100),  @DiaChi VARCHAR(100),  @SDT VARCHAR(10)  AS  BEGIN  INSERT INTO NHA\_CUNG\_CAP (TenNCC, DiaChi, SDT)  VALUES (@TenNCC, @DiaChi, @SDT);  END;  CREATE PROCEDURE sp\_XoaNhaCungCap  @MaNCC INT  AS  BEGIN  DELETE FROM NHA\_CUNG\_CAP WHERE MaNCC = @MaNCC;  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonAdd\_Click(object sender, EventArgs e)  {  string tenNCC = textBoxTenNCC.Text;  string diaChi = textBoxDiaChi.Text;  string sdt = textBoxSDT.Text;  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_ThemNhaCungCap", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@TenNCC", SqlDbType.VarChar).Value = tenNCC;  cmd.Parameters.Add("@DiaChi", SqlDbType.VarChar).Value = diaChi;  cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = sdt;  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thêm nhà cung cấp thành công!", "Thêm Nhà Cung Cấp", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Thêm nhà cung cấp thất bại.", "Thêm Nhà Cung Cấp", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

* 1. **Sửa nhà cung cấp.**

|  |
| --- |
| CREATE PROCEDURE sp\_SuaNhaCungCap  @MaNCC INT,  @TenNCC VARCHAR(100),  @DiaChi VARCHAR(100),  @SDT VARCHAR(10)  AS  BEGIN  UPDATE NHA\_CUNG\_CAP  SET TenNCC = @TenNCC, DiaChi = @DiaChi, SDT = @SDT  WHERE MaNCC = @MaNCC;  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonUpdate\_Click(object sender, EventArgs e)  {  int maNCC = Convert.ToInt32(textBoxMaNCC.Text);  string tenNCC = textBoxTenNCC.Text;  string diaChi = textBoxDiaChi.Text;  string sdt = textBoxSDT.Text;  db.openConnection();  SqlCommand cmd = new SqlCommand("sp\_SuaNhaCungCap", db.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add("@MaNCC", SqlDbType.Int).Value = maNCC;  cmd.Parameters.Add("@TenNCC", SqlDbType.VarChar).Value = tenNCC;  cmd.Parameters.Add("@DiaChi", SqlDbType.VarChar).Value = diaChi;  cmd.Parameters.Add("@SDT", SqlDbType.VarChar).Value = sdt;  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Cập nhật nhà cung cấp thành công!", "Cập Nhật Nhà Cung Cấp", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Cập nhật nhà cung cấp thất bại.", "Cập Nhật Nhà Cung Cấp", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  db.closeConnection();  } |

1. **Thống kê doanh thu và chi phí nhập hàng.**
   1. **Doanh thu theo ngày.**

|  |
| --- |
| CREATE FUNCTION GetDoanhThuTheoNgay (@Ngay DATE)  RETURNS DECIMAL(15, 2)  AS  BEGIN  DECLARE @DoanhThu DECIMAL(15, 2);  SELECT @DoanhThu = SUM(TriGiaHD)  FROM HOA\_DON  WHERE CAST(Ngay AS DATE) = CAST(@Ngay AS DATE) and TrangThai = 'Đã thanh toán';  RETURN @DoanhThu;  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonDoanhThuTheoNgay\_Click(object sender, EventArgs e)  {  try  {  DateTime ngayCanTinh = dateTimePicker.Value;  db.openConnection();  using (SqlCommand cmd = new SqlCommand("SELECT dbo.GetDoanhThuTheoNgay(@Ngay)", db.getConnection))  {  cmd.Parameters.AddWithValue("@Ngay", ngayCanTinh.Date);  object result = cmd.ExecuteScalar();  double doanhThu = result != DBNull.Value ? Convert.ToDouble(result) : 0;  lblDoanhThu.Text = $"Doanh thu theo ngày {ngayCanTinh.ToString("dd/MM/yyyy")}: {doanhThu:N2} VND";  }  db.closeConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

* 1. **Doanh thu theo tháng.**

|  |
| --- |
| CREATE FUNCTION GetDoanhThuTheoThang (@ThangNam DATE)  RETURNS DECIMAL(15, 2)  AS  BEGIN  DECLARE @DoanhThu DECIMAL(15, 2);  SELECT @DoanhThu = SUM(TriGiaHD)  FROM HOA\_DON  WHERE MONTH(Ngay) = MONTH(@ThangNam)  AND YEAR(Ngay) = YEAR(@ThangNam)  AND TrangThai = 'Đã thanh toán';  RETURN ISNULL(@DoanhThu, 0); -- Trả về 0 nếu không có doanh thu  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonDoanhThuTheoThang\_Click(object sender, EventArgs e)  {  try  {  // Giả sử bạn có một DateTimePicker với tên là dateTimePickerThangNam  DateTime thangNam = dateTimePicker.Value; // Lấy giá trị từ DateTimePicker  int thang = thangNam.Month; // Lấy tháng  int nam = thangNam.Year; // Lấy năm  db.openConnection();  using (SqlCommand cmd = new SqlCommand("SELECT dbo.GetDoanhThuTheoThang(@ThangNam)", db.getConnection))  {  cmd.Parameters.AddWithValue("@ThangNam", thangNam);  object result = cmd.ExecuteScalar();  decimal doanhThu = result != DBNull.Value ? Convert.ToDecimal(result) : 0;  lblDoanhThu.Text = $"Doanh thu theo tháng {thang}/{nam}: {doanhThu:N2} VND";  }  db.closeConnection();  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

* 1. **Doanh thu theo năm.**

|  |
| --- |
| CREATE FUNCTION GetDoanhThuTheoNam (@Nam DATE)  RETURNS DECIMAL(15, 2)  AS  BEGIN  DECLARE @DoanhThu DECIMAL(15, 2);  SELECT @DoanhThu = SUM(TriGiaHD)  FROM HOA\_DON  WHERE YEAR(Ngay) = YEAR(@Nam) AND TrangThai = 'Đã thanh toán';  RETURN ISNULL(@DoanhThu, 0);  END; |

Thực thi trên C#:

|  |
| --- |
| private void buttonDoanhThuTheoNam\_Click(object sender, EventArgs e)  {  try  {  DateTime date = dateTimePicker.Value; // Lấy giá trị DateTime  db.openConnection();  using (SqlCommand cmd = new SqlCommand("SELECT dbo.GetDoanhThuTheoNam(@Nam)", db.getConnection))  {  cmd.Parameters.AddWithValue("@Nam", date);  object result = cmd.ExecuteScalar();  decimal doanhThu = result != DBNull.Value ? Convert.ToDecimal(result) : 0;  lblDoanhThu.Text = $"Doanh thu theo năm {date.Year}: {doanhThu:N2} VND";  }  db.closeConnection();  }  catch (SqlException sqlEx)  {  MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "SQL Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

* 1. **Chi phí nhập hàng theo tháng.**

|  |
| --- |
| CREATE FUNCTION dbo.GetTongTienNhapHangTheoThangNam  (  @Month INT,  @Year INT  )  RETURNS DECIMAL(18, 2)  AS  BEGIN  DECLARE @TongTien DECIMAL(18, 2);    -- Tính tổng tiền theo tháng và năm  SELECT @TongTien = SUM(TriGiaDonNH)  FROM DON\_NHAP  WHERE MONTH(NgayNH) = @Month AND YEAR(NgayNH) = @Year;  RETURN ISNULL(@TongTien, 0);  END;  GO |

Thực thi trên C#:

|  |
| --- |
| private decimal GetTongTienNhapHangTheoThangNam(int month, int year)  {  decimal tongTien = 0;  using (SqlConnection conn = GetConnection())  {  try  {  conn.Open();  using (SqlCommand cmd = new SqlCommand("SELECT dbo.GetTongTienNhapHangTheoThangNam(@Month, @Year)", conn))  {  cmd.Parameters.AddWithValue("@Month", month);  cmd.Parameters.AddWithValue("@Year", year);  object result = cmd.ExecuteScalar(); // Lấy kết quả trả về  if (result != DBNull.Value)  {  tongTien = Convert.ToDecimal(result);  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  }  return tongTien;  } |

* 1. **Chi phí nhập hàng theo năm.**

|  |
| --- |
| CREATE FUNCTION dbo.GetTongTienNhapHangTheoNam  (  @Year INT  )  RETURNS DECIMAL(18, 2)  AS  BEGIN  DECLARE @TongTien DECIMAL(18, 2);    -- Tính tổng tiền theo năm  SELECT @TongTien = SUM(TriGiaDonNH)  FROM DON\_NHAP  WHERE YEAR(NgayNH) = @Year;  RETURN ISNULL(@TongTien, 0);  END;  GO |

Thực thi trên C#:

|  |
| --- |
| private decimal GetTongTienNhapHangTheoNam(int year)  {  decimal tongTien = 0;  using (SqlConnection conn = GetConnection())  {  try  {  conn.Open();  using (SqlCommand cmd = new SqlCommand("SELECT dbo.GetTongTienNhapHangTheoNam(@Year)", conn))  {  cmd.Parameters.AddWithValue("@Year", year);  object result = cmd.ExecuteScalar(); // Lấy kết quả trả về  if (result != DBNull.Value)  {  tongTien = Convert.ToDecimal(result);  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  }  return tongTien;  } |

1. **Báo cáo.**
   1. **Tương quan doanh thu 2 tháng bất kỳ.**

|  |
| --- |
| CREATE FUNCTION GetDoanhThuTuongQuanHaiThang (@Nam INT, @Thang1 INT, @Thang2 INT)  RETURNS TABLE  AS  RETURN  (  WITH DoanhThuThang1 AS (  SELECT dbo.GetDoanhThuTheoThang(CONVERT(DATE, CONCAT(@Nam, '-', @Thang1, '-01'))) AS DoanhThu\_Thang1  ),  DoanhThuThang2 AS (  SELECT dbo.GetDoanhThuTheoThang(CONVERT(DATE, CONCAT(@Nam, '-', @Thang2, '-01'))) AS DoanhThu\_Thang2  )  SELECT  @Thang1 AS Thang1,  @Thang2 AS Thang2,  dt1.DoanhThu\_Thang1,  dt2.DoanhThu\_Thang2,  CASE  WHEN dt1.DoanhThu\_Thang1 = 0 THEN NULL  ELSE ((dt2.DoanhThu\_Thang2 - dt1.DoanhThu\_Thang1) \* 100.0 / dt1.DoanhThu\_Thang1)  END AS TyLeThayDoi  FROM  DoanhThuThang1 dt1  CROSS JOIN  DoanhThuThang2 dt2  ); |

Thực thi trên C#:

|  |
| --- |
| private void vebieudo()  {  try  {  int nam = dateTimePicker1.Value.Year;  int thang1 = int.Parse(txtThang1.Text);  int thang2 = int.Parse(txtThang2.Text);  db.openConnection();  using (SqlCommand cmd = new SqlCommand("SELECT \* FROM dbo.GetDoanhThuTuongQuanHaiThang(@Nam, @Thang1, @Thang2)", db.getConnection))  {  cmd.Parameters.AddWithValue("@Nam", nam);  cmd.Parameters.AddWithValue("@Thang1", thang1);  cmd.Parameters.AddWithValue("@Thang2", thang2);  SqlDataReader reader = cmd.ExecuteReader();  if (reader.Read())  {  decimal doanhThuThang1 = (decimal)reader["DoanhThu\_Thang1"];  decimal doanhThuThang2 = (decimal)reader["DoanhThu\_Thang2"];  decimal? tyLeThayDoi = reader["TyLeThayDoi"] as decimal?;  chartDoanhThu.Series.Clear();  chartDoanhThu.ChartAreas.Clear();  chartDoanhThu.ChartAreas.Add(new ChartArea("MainArea"));  var seriesDoanhThu = new Series("Doanh thu");  seriesDoanhThu.ChartType = SeriesChartType.Column;  seriesDoanhThu.Points.AddXY($"Tháng {thang1}", doanhThuThang1);  seriesDoanhThu.Points.AddXY($"Tháng {thang2}", doanhThuThang2);  var seriesTyLe = new Series("Tỷ lệ thay đổi (%)");  seriesTyLe.ChartType = SeriesChartType.Line;  seriesTyLe.YAxisType = AxisType.Secondary; seriesTyLe.Points.AddXY($"Tháng {thang1}", 0);  seriesTyLe.Points.AddXY($"Tháng {thang2}", tyLeThayDoi ?? 0);  chartDoanhThu.Series.Add(seriesDoanhThu);  chartDoanhThu.Series.Add(seriesTyLe);  chartDoanhThu.ChartAreas["MainArea"].AxisY2.Title = "Tỷ lệ thay đổi (%)";  chartDoanhThu.ChartAreas["MainArea"].AxisY2.Enabled = AxisEnabled.True;  }  reader.Close();  }  db.closeConnection();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

* 1. **Top 3 sản phẩm bán chạy nhất.**

|  |
| --- |
| CREATE PROCEDURE GetSalesReport  @Month INT  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  SELECT  sp.MaSP,  sp.TenSP,  MONTH(hd.Ngay) AS Month,  YEAR(hd.Ngay) AS Year,  SUM(cthd.SoLuong) AS TotalSold  FROM  HOA\_DON hd  JOIN  CHI\_TIET\_HOA\_DON cthd ON hd.MaHD = cthd.MaHD  JOIN  SAN\_PHAM sp ON cthd.MaSP = sp.MaSP  WHERE  hd.TrangThai = 'Đã thanh toán'  AND MONTH(hd.Ngay) = @Month  AND YEAR(hd.Ngay) = YEAR(GETDATE())  GROUP BY  sp.MaSP, sp.TenSP, MONTH(hd.Ngay), YEAR(hd.Ngay)  ORDER BY TotalSold DESC;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi, rollback transaction  ROLLBACK TRANSACTION;  -- Thông báo lỗi  THROW;  END CATCH  END;  CREATE PROCEDURE GetTopSellingProducts  @Month INT  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  -- Truy vấn 3 sản phẩm bán chạy nhất  SELECT TOP 3  sp.MaSP,  sp.TenSP,  SUM(cthd.SoLuong) AS TotalSold  FROM  HOA\_DON hd  JOIN  CHI\_TIET\_HOA\_DON cthd ON hd.MaHD = cthd.MaHD  JOIN  SAN\_PHAM sp ON cthd.MaSP = sp.MaSP  WHERE  hd.TrangThai = 'Đã thanh toán'  AND MONTH(hd.Ngay) = @Month  AND YEAR(hd.Ngay) = YEAR(GETDATE())  GROUP BY  sp.MaSP, sp.TenSP  ORDER BY TotalSold DESC;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  THROW;  END CATCH  END; |

Thực thi trên C#:

|  |
| --- |
| private void LoadSalesData(int month)  {  db.openConnection();  SqlCommand command1 = new SqlCommand("GetSalesReport", db.getConnection);  command1.CommandType = CommandType.StoredProcedure;  command1.Parameters.AddWithValue("@Month", month);  SqlDataReader reader1 = command1.ExecuteReader();  DataTable dataTable = new DataTable();  dataTable.Load(reader1);  dataGridView1.DataSource = dataTable;  reader1.Close();  SqlCommand command2 = new SqlCommand("GetTopSellingProducts", db.getConnection);  command2.CommandType = CommandType.StoredProcedure;  command2.Parameters.AddWithValue("@Month", month);  SqlDataReader reader2 = command2.ExecuteReader();  if (reader2.HasRows)  {  label2.Text = "3 Sản phẩm bán chạy nhất: \n";  while (reader2.Read())  {  label2.Text += $"{reader2["TenSP"]} - Số lượng bán: {reader2["TotalSold"]}\n";  }  }  else  {  label2.Text = "Không có kết quả sản phẩm bán chạy nhất.";  }  reader2.Close();  db.closeConnection();  } |

* 1. **Top 3 nhân viên bán hàng xuất sắc nhất.**

|  |
| --- |
| CREATE FUNCTION BaoCaoDoanhThuTop3NhanVienTheoThang(@Thang INT, @Nam INT)  RETURNS TABLE  AS  RETURN  SELECT TOP 3  nv.MaNV,  nv.HoNV + ' ' + nv.TenNV AS TenNhanVien,  ISNULL(SUM(hd.TriGiaHD), 0) AS TongDoanhThu  FROM NHAN\_VIEN nv  LEFT JOIN HOA\_DON hd ON nv.MaNV = hd.MaNV  AND hd.TrangThai = N'Đã thanh toán'  AND MONTH(hd.Ngay) = @Thang  AND YEAR(hd.Ngay) = @Nam  GROUP BY nv.MaNV, nv.HoNV, nv.TenNV  ORDER BY TongDoanhThu DESC; |

Thực thi trên C#:

|  |
| --- |
| private void btnGetReport\_Click(object sender, EventArgs e)  {  int month;  int year;  if (!int.TryParse(txtMonth.Text, out month) || !int.TryParse(txtYear.Text, out year))  {  MessageBox.Show("Vui lòng nhập tháng và năm hợp lệ.");  return;  }  string query = "SELECT \* FROM BaoCaoDoanhThuTop3NhanVienTheoThang(@Thang, @Nam)";  Try {  db.openConnection();  using (SqlCommand cmd = new SqlCommand(query, db.getConnection))  {  cmd.Parameters.AddWithValue("@Thang", month);  cmd.Parameters.AddWithValue("@Nam", year);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable table = new DataTable();  adapter.Fill(table);  dataGridView.DataSource = table;  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  finally  {  db.closeConnection();  }  } |

**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

1. **Phân quyền cho các role.**

+ Tạo role Nhân viên (Employee)

-- Tạo vai trò Employee

CREATE ROLE Employee;

|  |
| --- |
| -- Gán và phân quyền cho Nhân Viên (Employee)  CREATE ROLE Employee  GRANT SELECT, INSERT, UPDATE, DELETE ON Chi\_Tiet\_Hoa\_Don TO Employee;  GRANT SELECT, INSERT, UPDATE, DELETE ON Chi\_Tiet\_Don\_Nhap\_Hang TO Employee;  GRANT SELECT, INSERT, UPDATE, DELETE ON Don\_Nhap TO Employee;  GRANT SELECT, INSERT, UPDATE, DELETE ON Loai\_San\_Pham TO Employee;  GRANT SELECT, INSERT, UPDATE, DELETE ON Nha\_Cung\_cap TO Employee;  DENY SELECT, INSERT, UPDATE, DELETE ON NhanVien\_User TO Employee;  DENY SELECT, INSERT, UPDATE, DELETE ON Nhan\_Vien TO Employee;  GRANT EXECUTE ON SCHEMA::dbo TO Employee;  GO  DENY SELECT ON OBJECT::dbo.BaoCaoDoanhThuTop3NhanVienTheoThang TO Employee;  DENY SELECT ON OBJECT::dbo.GetDoanhThuTuongQuanHaiThang TO Employee;  DENY EXECUTE ON OBJECT::dbo.GetDoanhThuTheoNgay TO Employee;  DENY EXECUTE ON OBJECT::dbo.GetDoanhThuTheoThang TO Employee;  DENY EXECUTE ON OBJECT::dbo.GetDoanhThuTheoNam TO Employee;  DENY EXECUTE ON OBJECT::dbo.GetTongTienNhapHangTheoThangNam TO Employee;\  DENY EXECUTE ON OBJECT::dbo.sp\_ThemUser TO Employee;  DENY SELECT ON OBJECT::dbo.GetDoanhThuTuongQuanHaiThang TO Employee; |

+ Đối với role Manager ta dùng server sysadmin

1. **Procedure tạo tài khoản phần quyền Sql Server và tài khoản phần mềm.**

|  |
| --- |
| CREATE PROCEDURE sp\_ThemUser  @userName NVARCHAR(30),  @passWord NVARCHAR(10),  @maNV INT,  @role INT  AS  BEGIN  DECLARE @sqlString NVARCHAR(2000)  -- Tạo login mới trong SQL Server nếu chưa tồn tại  SET @sqlString = 'IF NOT EXISTS (SELECT 1 FROM sys.sql\_logins WHERE name = ''' + @userName + ''') ' +  'CREATE LOGIN [' + @userName + '] WITH PASSWORD = ''' + @passWord + ''', DEFAULT\_DATABASE = [QUANLYBANHANG], CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF';  EXEC (@sqlString);  -- Tạo user cho cơ sở dữ liệu hiện tại  SET @sqlString = 'IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = ''' + @userName + ''') ' +  'CREATE USER [' + @userName + '] FOR LOGIN [' + @userName + ']';  EXEC (@sqlString);  -- Thêm người dùng vào vai trò trong cơ sở dữ liệu hiện tại  IF @role = 1  BEGIN  SET @sqlString = 'ALTER SERVER ROLE sysadmin ADD MEMBER [' + @userName + ']'; -- Ví dụ thêm vào vai trò Manager  END  ELSE  BEGIN  SET @sqlString = 'ALTER ROLE [Employee] ADD MEMBER [' + @userName + ']'; -- Ví dụ thêm vào vai trò Employee  END  EXEC (@sqlString);  -- Thêm thông tin vào bảng NHANVIEN\_USER  INSERT INTO NHANVIEN\_USER (Username, Password, MaNV, Role)  VALUES (@userName, @passWord, @maNV, @role);  PRINT 'Thêm người dùng thành công';  END |

1. **Procedure sp\_XoaUser để xóa nhân viên.**

|  |
| --- |
| CREATE PROCEDURE sp\_XoaUser  @userName NVARCHAR(30)  AS  BEGIN  DECLARE @sqlString NVARCHAR(2000);  DECLARE @maNV INT;  SELECT @maNV = MaNV FROM NHANVIEN\_USER WHERE Username = @userName;  SET @sqlString = 'ALTER ROLE [Employee] DROP MEMBER [' + @userName + ']';  EXEC (@sqlString);  SET @sqlString = 'DROP USER [' + @userName + ']';  EXEC (@sqlString);  SET @sqlString = 'DROP LOGIN [' + @userName + ']';  EXEC (@sqlString);  DELETE FROM NHANVIEN\_USER WHERE Username = @userName;  DELETE FROM HoaDon WHERE MaNV = @maNV;  DELETE FROM NhanVien WHERE MaNV = @maNV;  PRINT 'Người dùng đã được xóa thành công';  END |

**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG.**

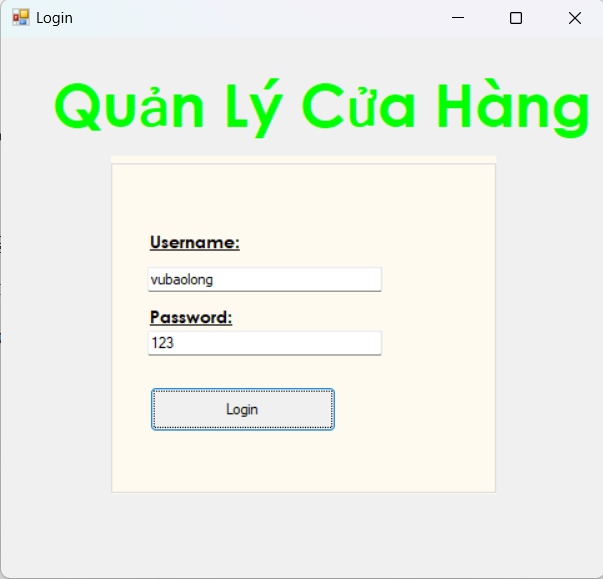
**Môi trường cài đặt và công nghệ sử dụng**

- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 20.

- 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 của chương trình:**

1. **Form đăng nhập.**

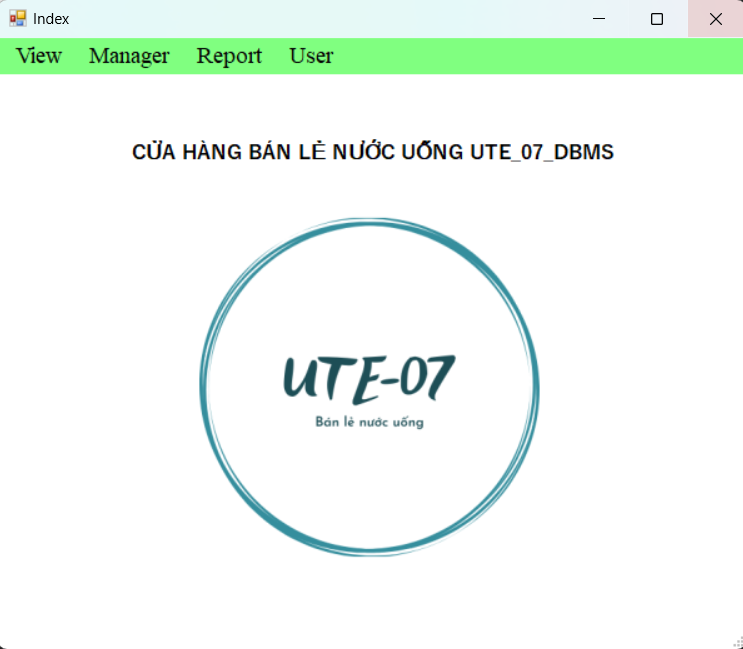
- Function Đăng nhập:

|  |
| --- |
| CREATE FUNCTION dbo.Fn\_LoginUser  (  @username NVARCHAR(30),  @password NVARCHAR(50)  )  RETURNS BIT  AS  BEGIN  DECLARE @isValid BIT;  IF EXISTS (  SELECT 1  FROM NHANVIEN\_USER  WHERE Username = @username  AND Password = @password  )  BEGIN  SET @isValid = 1;  END  ELSE  BEGIN  SET @isValid = 0;  END  RETURN @isValid;  END; |

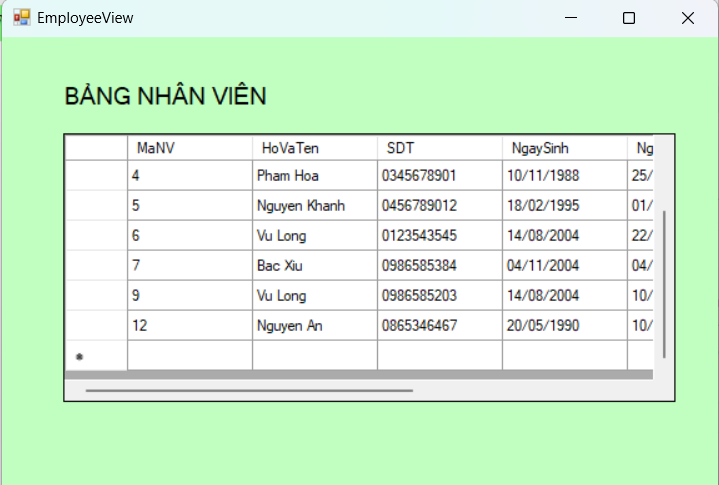
- Câu lệnh thực thi C#:

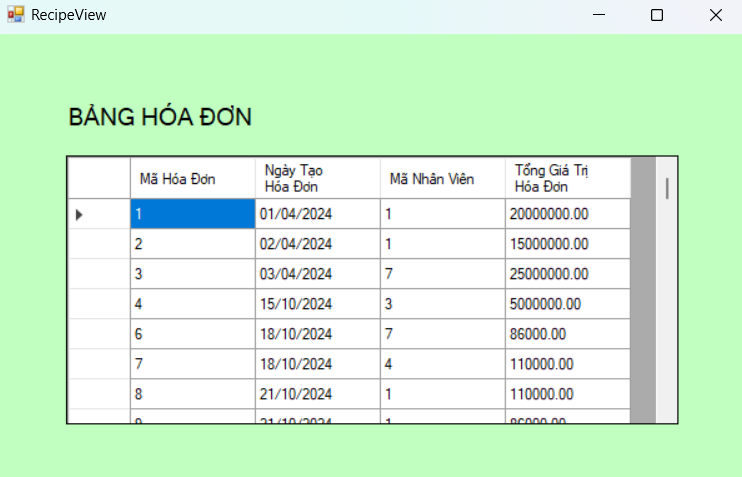
|  |
| --- |
| private void buttonlogin\_Click(object sender, EventArgs e)  {  string username = user.Text.ToString().Trim();  string password = pass.Text.ToString().Trim();  GlobalVariables.LoggedInUsername = username;  GlobalVariables.LoggedInPassword = password;  MY\_DB db = new MY\_DB();  try  {  db.getConnection.Open();  string sqlQuery = "SELECT dbo.Fn\_LoginUser(@username, @password)";  SqlCommand command = new SqlCommand(sqlQuery, db.getConnection);  command.Parameters.AddWithValue("@username", username);  command.Parameters.AddWithValue("@password", password);  object resultObj = command.ExecuteScalar();  int result = Convert.ToInt32(resultObj ?? 0);  if (result == 1)  {  Index index = new Index();  index.ShowDialog();  }  else  {  MessageBox.Show("Tên đăng nhập hoặc mật khẩu không chính xác", "Lỗi đăng nhập", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi hệ thống", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  // Đảm bảo đóng kết nối sau khi hoàn thành  if (db.getConnection.State == ConnectionState.Open)  {  db.getConnection.Close();  }  }  } |

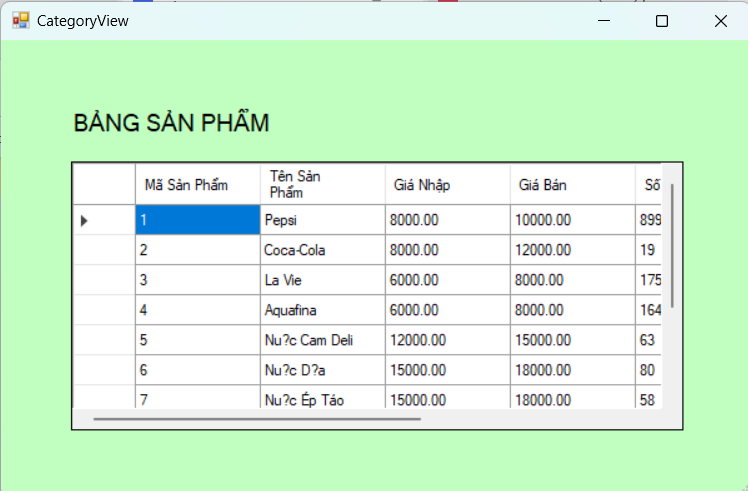
1. **Form Trang chủ.**



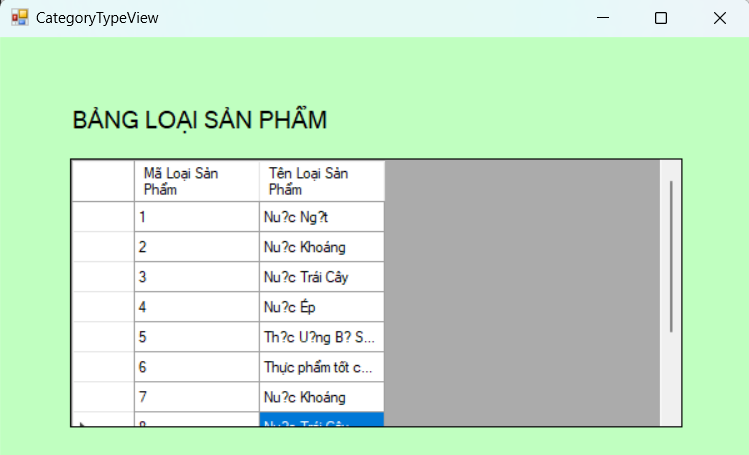
1. **Các loại danh mục.**
   1. **Danh mục nhân viên.**



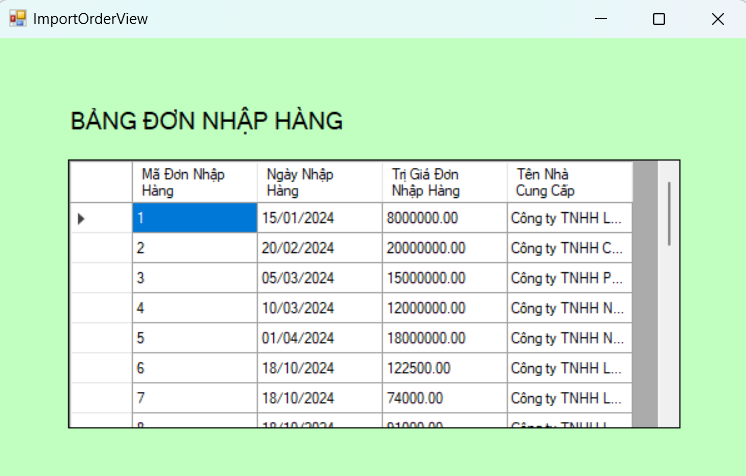
* 1. **Danh mục hóa đơn.**
  2. **Danh mục sản phẩm.**



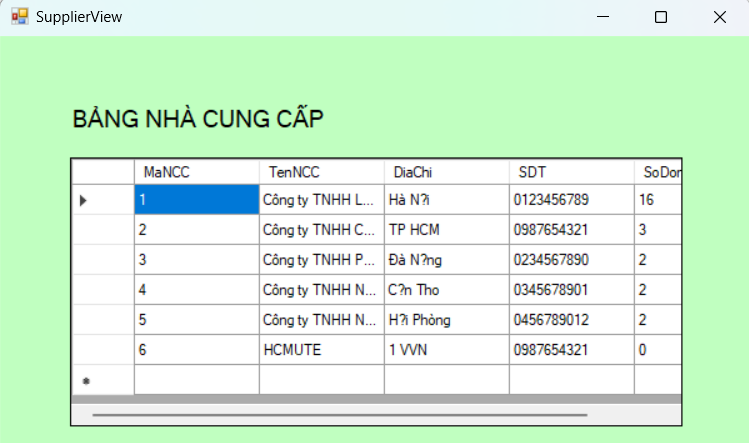
* 1. **Danh mục loại sản phẩm.**

****

* 1. **Danh mục đơn nhập hàng.**

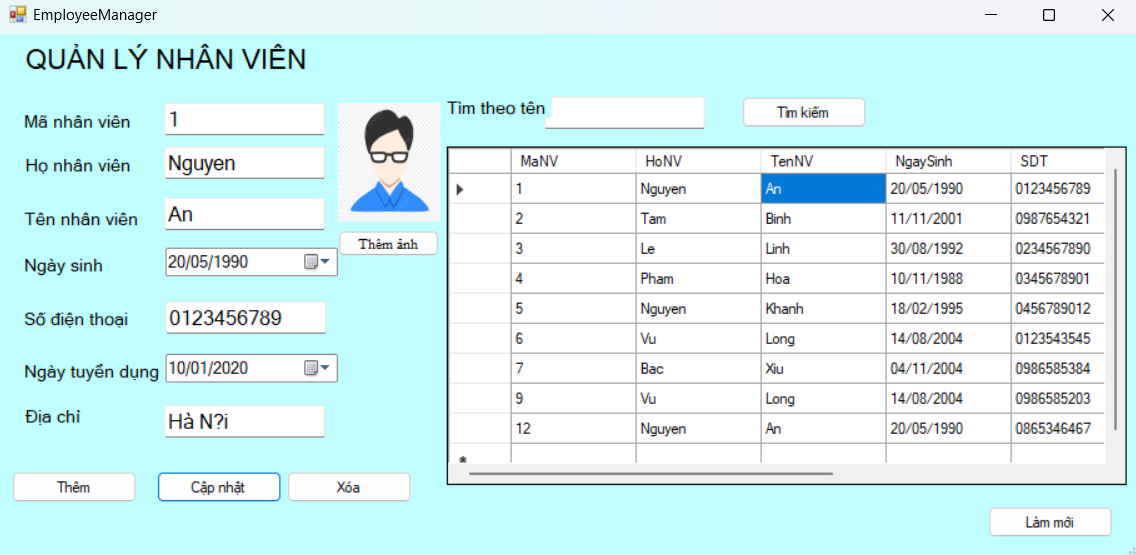


* 1. **Danh mục nhà cung cấp.**

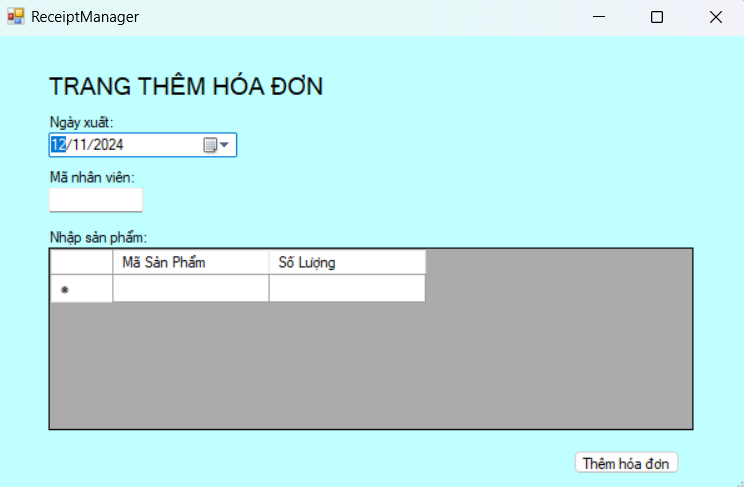


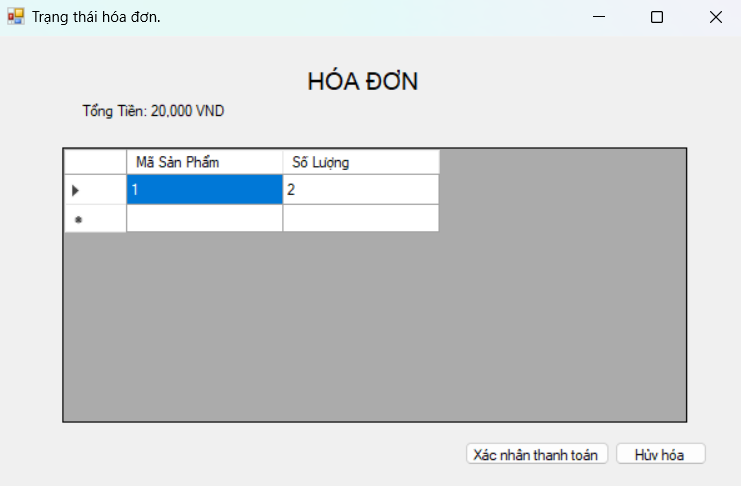
1. **Các chức năng quản lý.**

**4.1 Quản lý nhân viên.**

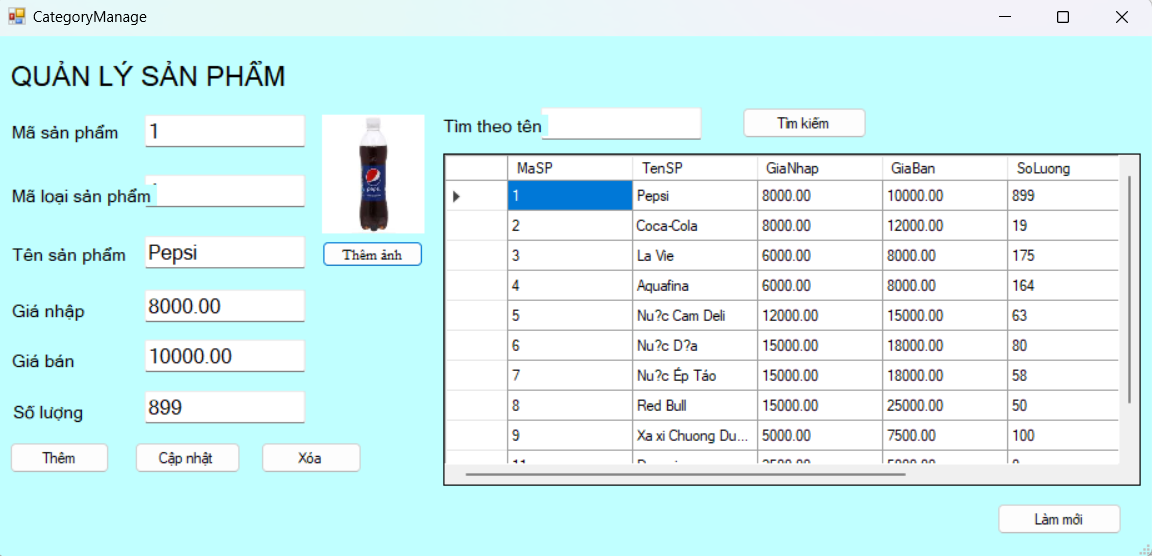


**4.2 Quản lý hóa đơn.**

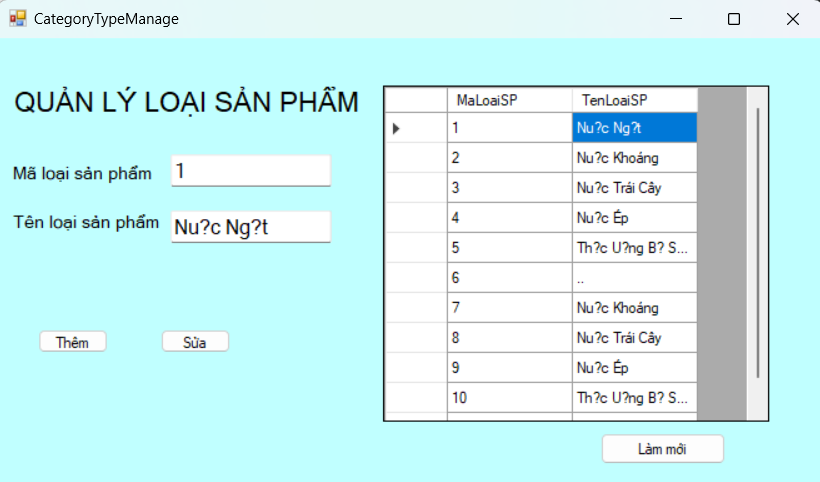




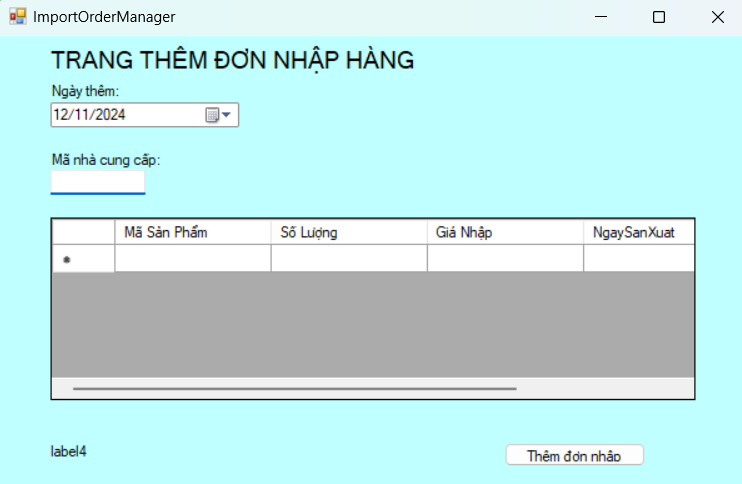
**4.3 Quản lý sản phẩm.**



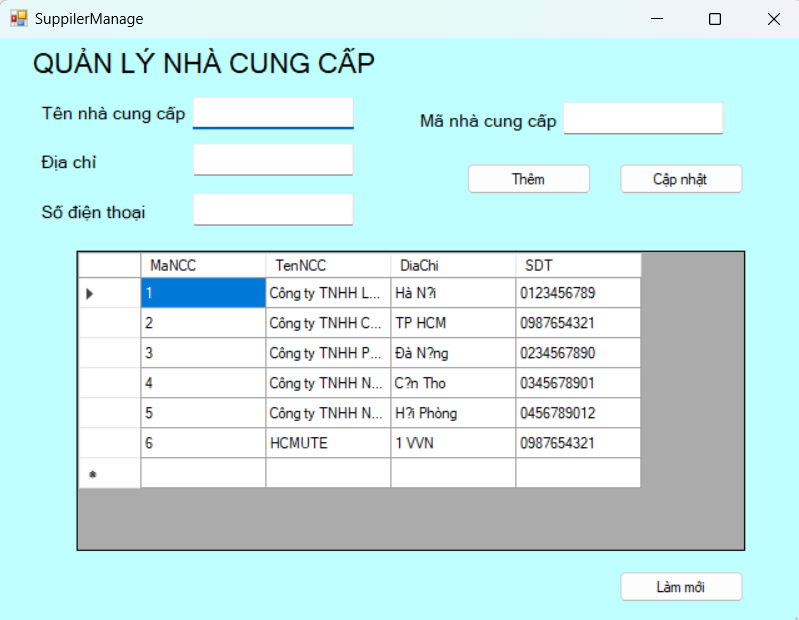
**4.4 Quản lý loại sản phẩm.**



**4.5 Quản lý đơn nhập hàng.**



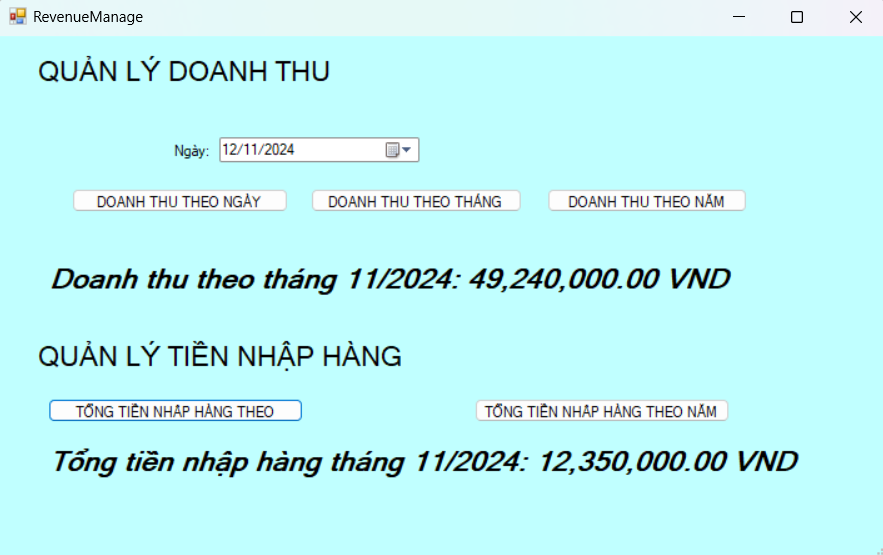
**4.6 Quản lý nhà cung cấp.**



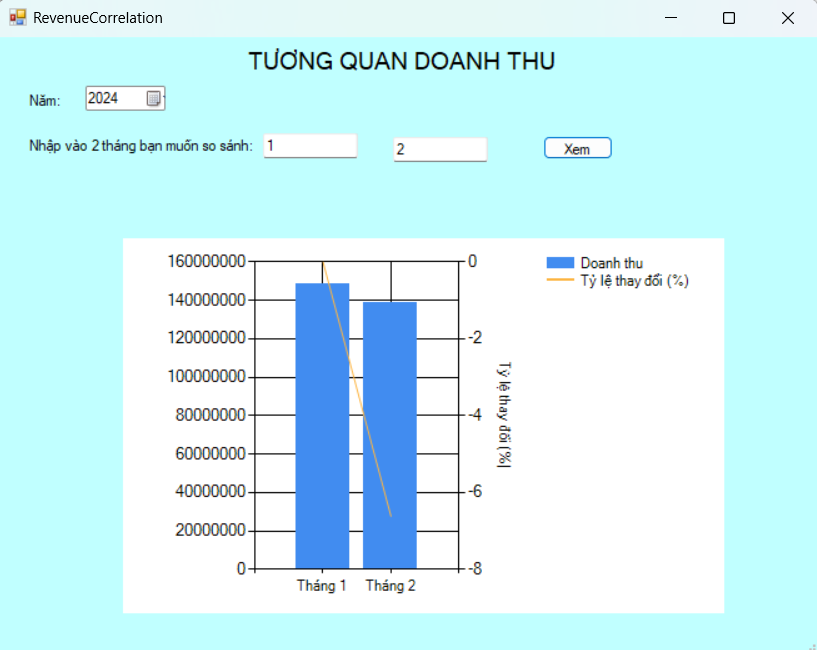
**4.7 Quản lý người dùng.**



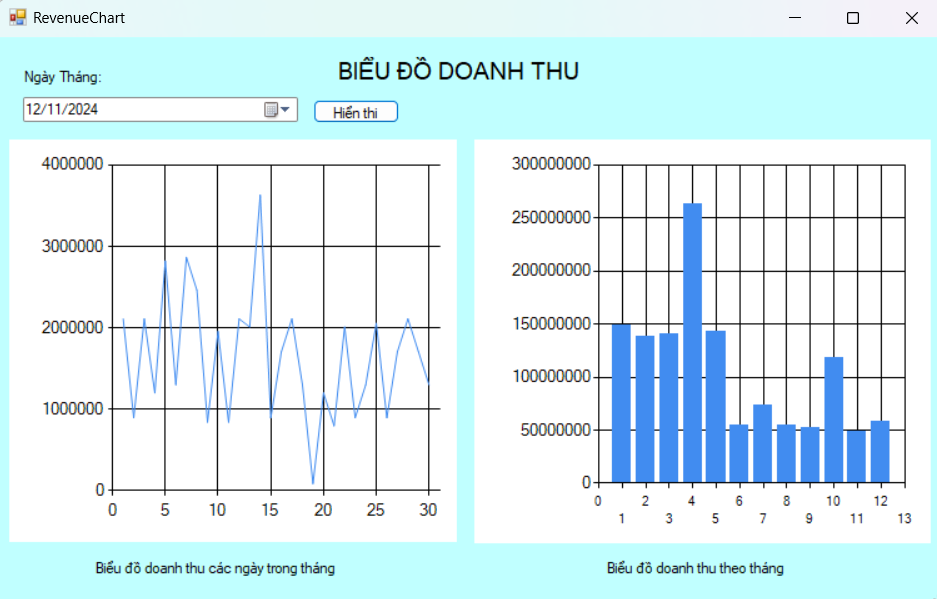
1. **Báo cáo**
   1. **Báo cáo doanh thu theo ngày, tháng, năm và tiền nhập hàng theo tháng, năm.**



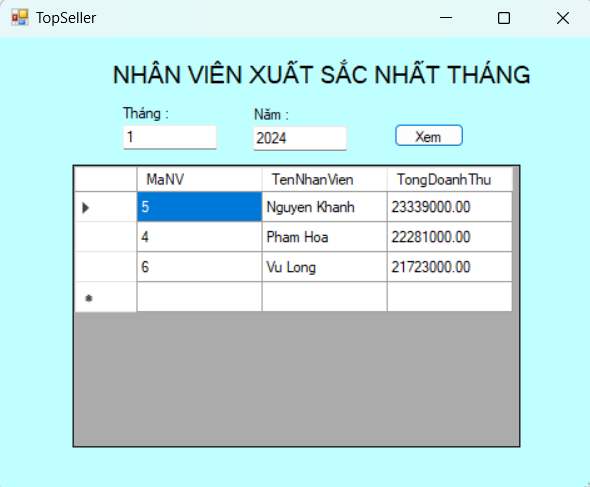
* 1. **Tương quan doanh thu 2 tháng bất kỳ.**



* 1. **Biểu đồ doanh thu.**



* 1. **Top 3 nhân viên xuất sắc.**



* 1. **Top 3 sản phẩm bán chạy nhất.**

