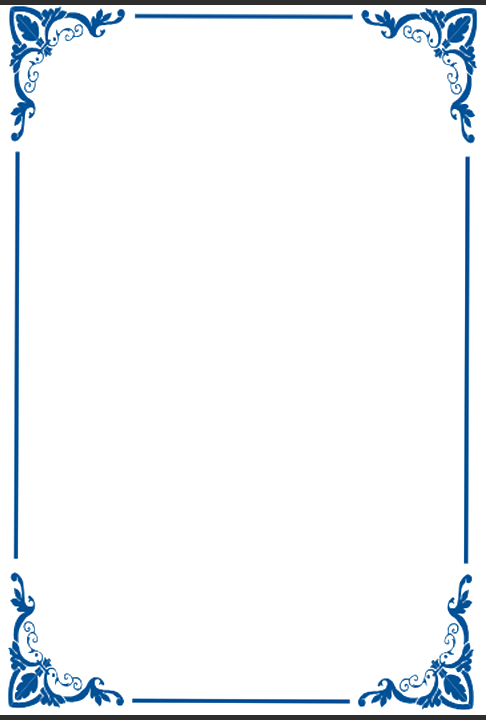
****

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

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO CUỐI KỲ**

**Tên đề tài:****QUẢN LÝ NHÀ HÀNG**

**TÊN HỌC PHẦN:** Hệ Quản Trị Cơ Sở Dữ Liệu

**MÃ SỐ LỚP HP:** 242DBMS330284\_03CLC

**GVHN:** TS. Nguyễn Thành Sơn

**Nhóm thực hiện:** 06

**TP.HCM, ngày 21 tháng 4 năm 2025**

**DANH SÁCH SINH VIÊN THAM GIA**

**Mã học phần:** 242DBMS330284\_03CLC

**Nhóm**: 06

***Tên đề tài:*** *QUẢN LÝ NHÀ HÀNG*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN THÀNH VIÊN** | **MÃ SỐ SINH VIÊN** | **TỶ LỆ THAM GIA** |
| 1 | Nguyễn Huy | 21110187 | 100% |
| 2 | Lương Nguyễn Thành Hưng | 23110111 | 100% |
| 3 | Bùi Hoàng Hưng | 21110832 | 100% |
| 4 | Đặng Gia Huy | 23110101 | 100% |

**Ghi chú:**

Tỷ lệ %: Mức độ phần trăm hoàn thành của từng sinh viên tham gia.

***Trưởng nhóm:*** Nguyễn Huy

**Nhận xét của giáo viên**

...……………………………………………………………………………………

……………………………………………………………………………………....

………………………………………………………………………………………

………………………………………………………………………………………

*Tp. Hồ Chí Minh - Tháng 4 năm 2025*

**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.

Mục Lục

[**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG** 6](#_Toc196172978)

[**1. Đặc tả đề tài** 6](#_Toc196172979)

[**1.1 . Mô tả bài toán** 6](#_Toc196172980)

[**1.2 . Nghiệp vụ của bài toán** 8](#_Toc196172981)

[**1.3 . Mô tả chức năng của bài toán** 9](#_Toc196172982)

[**1.4. Tính giá tiền khuyến mãi cho khách hàng** 10](#_Toc196172983)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 11](#_Toc196172984)

[**1.** **Thiết kế cơ sở dữ liệu mức quan niệm** 11](#_Toc196172985)

[**2.** **Thiết kế cơ sở dữ liệu mức logic** 13](#_Toc196172986)

[**3.** **Các ràng buộc cần có** 14](#_Toc196172987)

[**4.** **Cài đặt các CSDL và các ràng buộc** 16](#_Toc196172988)

[**5.** **Các view** 23](#_Toc196172989)

[**6.** **Các Trigger** 23](#_Toc196172990)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG** 27](#_Toc196172991)

[**1. Kết nối cơ sở dữ liệu** 27](#_Toc196172992)

[**2. Danh mục hóa đơn** 27](#_Toc196172993)

[**3. Chi Tiết Hóa Đơn** 30](#_Toc196172994)

[**4. Nhân viên** 33](#_Toc196172995)

[**5. Nhà cung cấp** 36](#_Toc196172996)

[**6. Khuyến Mãi** 38](#_Toc196172997)

[**7. Món Ăn** 41](#_Toc196172998)

[**8. Khách hàng** 43](#_Toc196172999)

[**9. Nguyên liệu** 45](#_Toc196173000)

[**10. Nhập kho** 48](#_Toc196173001)

[**11. Xuất Kho** 50](#_Toc196173002)

[**12. Lưu Trữ** 53](#_Toc196173003)

[**13. Chức Vụ** 56](#_Toc196173004)

[**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN** 57](#_Toc196173005)

[**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG** 60](#_Toc196173006)

[**5.1. Form đăng nhập** 60](#_Toc196173007)

[**5.2. Form đăng ký** 61](#_Toc196173008)

[**5.3. Form Trang Chủ** 63](#_Toc196173009)

[**5.4. Các Danh Mục Hệ Thống** 63](#_Toc196173010)

[**5.4.1. Danh Mục Thông Tin Tài Khoản** 63](#_Toc196173011)

[**5.4.2. Danh Mục Đổi Mật Khẩu** 64](#_Toc196173012)

[**5.5. Các Danh Mục Quản Lý** 65](#_Toc196173013)

[**5.5.1. Danh Mục Danh Sách Nhân Viên** 65](#_Toc196173014)

[**5.5.2. Danh Mục Danh Sách Hóa Đơn** 66](#_Toc196173015)

[**5.5.3. Danh Mục Nhập Kho** 67](#_Toc196173016)

[**5.5.4. Danh Mục Xuất Kho** 68](#_Toc196173017)

[**5.5.5. Danh Mục Lưu Trữ** 69](#_Toc196173018)

[**5.6. Các Danh Mục** 70](#_Toc196173019)

[**5.6.1. Danh Mục Nguyên Liệu** 70](#_Toc196173020)

[**5.6.2. Danh Mục Nhà Cung Cấp** 71](#_Toc196173021)

[**5.6.3. Danh Mục Khách Hàng** 72](#_Toc196173022)

[**5.6.4. Danh Mục Khuyến Mãi** 73](#_Toc196173023)

[**5.6.5. Danh Mục Món Ăn** 74](#_Toc196173024)

[**5.6.6. Danh Mục Chức Vụ** 75](#_Toc196173025)

[**5.6.7. Danh Mục Chi Tiết Hóa Đơn** 76](#_Toc196173026)

# **CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

## **1. Đặc tả đề tài**

### **1.1 . Mô tả bài toán**

Một quán ăn gia đình cần có một hệ thống để quản lý. Chương trình quản lý được xây dựng cho ba đối tượng người dùng là người quản lý, nhân viên quản lý kho và nhân viên bán hàng với các quyền khác nhau. Đối với người dùng là nhân viên bán hàng, họ sẽ có quyền truy cập vào các chức năng như quản lý danh sách hóa đơn, danh mục khách hàng, danh mục khuyến mãi, danh mục nhóm món ăn, danh mục chi tiết hóa đơn. Đối với người dùng là nhân viên quản lý kho, họ sẽ có quyền truy cập vào chức năng như quản lý nhập kho, quản lý xuất kho, quản lý lưu trữ, danh mục nguyên liệu, danh mục nhà cung cấp. Còn với người dùng quản lý quán ăn, họ sẽ có quyền truy cập vào các chức năng như quản lý hóa đơn, quản lý danh sách nhân viên, quản lý danh sách nhập kho, quản lý danh sách xuất kho, quản lý danh mục nguyên liệu, danh mục nhà cung cấp, danh mục khách hàng, danh mục khuyến mãi, danh mục món ăn, danh mục chức vụ, danh mục chi tiết hóa đơn. Đối với hệ thống quản lý quán ăn này cần lưu trữ những dữ liệu sau:

Quán ăn sẽ có nhiều nhân viên, mỗi nhân viên phân biệt với nhau bằng mã nhân viên. Một nhân viên sẽ bao gồm các thông tin như mã nhân viên, tên nhân viên, địa chỉ, số điện thoại, Email, mã chức vụ.

Mỗi nhân viên sẽ được cấp một tài khoản, các tài khoản sẽ có quyền truy cập dựa theo vị trí công việc của nhân viên đó. Vị trí công việc của các nhân viên phân biệt với nhau bằng mã vị trí công việc. Công việc của nhân viên bao gồm mã công việc và tên vị trí công việc. Mỗi tài khoản phân biệt với nhau bằng mã tài khoản. Một tài khoản chỉ có thể có một công việc, một công việc có thể có nhiều tài khoản làm. Mỗi tài khoản của nhân viên sẽ có các thông tin như mã tài khoản, tên tài khoản, mật khẩu, mã nhân viên.

Thực đơn được làm ra từ các nguyên liệu. Một thực đơn được làm ra từ nhiều nguyên liệu, một nguyên liệu có thể dùng trong nhiều thực đơn. Các nguyên liệu phân biệt nhau bằng mã nguyên liệu. Mỗi nguyên liệu có các thông tin như mã nguyên liệu, mã nhà cung cấp, tên nguyên liệu, đơn vị tính, giá.

Danh sách món ăn được phân biệt với nhau bằng mã món ăn. Danh sách món ăn có thể có những thông tin như mã món ăn, tên món ăn, giá, số lượng món ăn hiện có.

Để quản lý nguyên liệu cần có thông tin của nhà cung cấp nguyên liệu, cũng như lên đơn đặt hàng nguyên liệu đảm bảo luôn cung cấp đủ nguyên liệu để chế biến thực đơn. Một nhà cung cấp có thể sản xuất ra nhiều nguyên liệu, một nguyên liệu cũng có thể được cung cấp từ nhiều nhà cung cấp. 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, email nhà cung cấp.

Đơn nhập kho được phân biệt bởi mã đơn nhập kho, mỗi đơn nhập hàng có các thông tin như mã đơn nhập kho, mã nhân viên phụ trách nhập kho, mã nguyên liệu, ngày nhập hàng, số ngày hết hạn, số lượng, tổng tiền.

Trong kho hàng có thể có những nguyên nhân dẫn đến nguyên liệu phải được xuất kho, một đơn xuất kho có thể gồm một hoặc nhiều nguyên liệu, một nguyên liệu có thể có nhiều đơn xuất kho. Mỗi đơn xuất kho được phân biệt bằng mã xuất kho. Một đơn xuất kho gồm có mã xuất kho, mã nhân viên phụ trách xuất kho, số lượng, nguyên nhân xuất kho, ngày xuất kho, mã nguyên liệu, mã lưu trữ.

Hóa đơn của khách hàng được phân biệt với nhau bằng mã hóa đơn. Một hóa đơn có thể lập từ một nhân viên, một nhân viên có thể lập được nhiều hóa đơn. Một hóa đơn bao gồm những thông tin như mã hóa đơn, mã nhân viên phụ trách lập hóa đơn, mã khách hàng, ngày lập hóa đơn, tổng tiền. Một hóa đơn có thể bao gồm cả khuyến mãi trong thẻ khách hàng, nếu khách hàng sử dụng khuyến mãi thì có thể giảm giá hóa đơn.

Khách hàng có thể đăng ký thẻ khách hàng để sử dụng các khuyến mãi của cửa hàng. Một số điện thoại có thể dùng cho một thẻ khách hàng. Mỗi thẻ khách hàng phân biệt với nhau bằng mã khách hàng. Thẻ khách hàng bao gồm những thông tin như: mã khách hàng, tên khách hàng, số điện thoại, địa chỉ. Mỗi khuyến mãi phân biệt với nhau bằng mã khuyến mãi. Khuyến mãi của cửa hàng bao gồm các thông tin như mã khuyến mãi, mã khách hàng, tên khuyến mãi, ngày hết hạn, tình trạng sử dụng.

### **1.2 . Nghiệp vụ của bài toán**

Bài toán quản lý nhà hàng là một nghiệp vụ quản lý kinh doanh trong lĩnh vực dịch vụ ăn uống. Nó liên quan tới việc quản lý tài khoản nhân viên, quản lý hàng hóa, quản lý hóa đơn, quản lý doanh thu và điều hành nhà hàng. Nhân viên là bộ mặt của quán và giao tiếp với khách hàng từ lúc bước vào cho đến khi khách ra về. Vì vậy chúng ta cần xây dựng tài liệu đào tạo nghiệp vụ cho nhân viên về quy trình tác nghiệp, văn hóa phục vụ, cách ứng xử, xử lý tình huống, …

- Quy trình thực hiện khi khách đến quán: Đầu tiên, nhân viên sẽ chào khách khi khách bước vào quán. Sau đó nhân viên đứng quầy sẽ bấm tạo đơn hàng trên chương trình, yêu cầu khách hàng đọc số điện thoại, hệ thống sẽ kiểm tra xem khách hàng đã tồn tại hay chưa, nếu tồn tại rồi sẽ đến bước kế tiếp, nếu chưa thì tạo thẻ thành viên mới cho khách hàng và đến bước kế tiếp. Tiếp theo nhân viên chỉ vào menu hỏi khách muốn sử dụng món nào, ngồi khu vực bàn nào, và nếu cảm thấy khách phân vân thì có thể tư vấn cho khách về thực đơn. Sau khi khách chọn được món thì nhân viên tiến hành thêm các món vào đơn hàng vừa tạo. Sau khi thêm các món vào đơn hàng vừa tạo, nhân viên xuất hóa đơn để đưa cho khách thanh toán. Sau khi hoàn thành các món trong đơn hàng, nhân viên phục vụ đưa món ra cho khách hàng. Nhân viên cần phải kiểm tra đầy đủ các món trong đơn, vật dụng đi kèm với món rồi mới đem đến cho khách hàng.

Ở giai đoạn đầu, quán phải tìm được đối tác, nhà sản xuất cung cấp hàng hóa uy tín từ đầu vì có chính sách giá cả, giao hàng, chứng từ rõ ràng để tiện cho việc kiểm soát. Quán cần thống kê hàng hóa nhập vào và có kho hàng để lưu trữ, bảo quản và kiểm soát hàng hóa. Quán cần kiểm tra hàng hóa theo hóa đơn nhập hàng và nhập vào chương trình quản lý quán ăn.

Tiếp theo, ở giai đoạn vận hành, chúng ta cần xây dựng chi phí của từng sản phẩm và lên giá thành cho các sản phẩm đó (ví dụ: 1kg thịt lợn có thể chế biến 3 dĩa cơm tấm). Ngoài ra, quán cần phải kiểm soát được thu chi hàng ngày. Nhân viên bán hàng sẽ được phân quyền để xem được hóa đơn mỗi đơn hàng và bảng doanh thu theo ngày để tổng kết ngày. Nhân viên sẽ kiểm tra xem số tiền thu được có đúng với doanh thu của ngày hôm đó không. Người quản lý kho sẽ được phân quyền để chi tiền cho số tiền nhập hàng. Kho hàng của quán cần được kiểm soát chặt chẽ. Kho hàng cần được kiểm tra định kỳ hàng tuần để đối chiếu với số liệu thực tế ghi nhận ở chương trình quản lý quán. Người quản lý kho hàng cần phải nắm các bảng báo cáo phân tích như báo cáo doanh thu (theo ngày, tháng, năm), báo cáo chi phí (chi tiết các khoản chi), báo cáo thất thoát (bị đổ vỡ, hư hỏng, mất mát). Chủ của nhà hàng sẽ dựa trên các báo cáo để nắm rõ doanh thu, chi phí, … để có thể đưa ra điều chỉnh sao cho có lợi nhất.

### **1.3 . Mô tả chức năng của bài toán**

**Đối với nhân viên bán hàng:**

Chức năng bán hàng: cho phép nhân viên tạo đơn hàng, thêm món, xóa món, xuất hóa đơn thanh toán, thêm khuyên mãi.

Quản lý khách hàng: cho phép nhân viên tạo thẻ thành viên cho khách hàng, nhập số điện thoại ở bước thanh toán để tạo khuyến mãi cho khách hàng.

Xem doanh thu: cho phép nhân viên có thể xem được doanh thu bán trong ngày.

**Đối với nhân viên quản lý kho hàng:**

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

Quản lý danh mục nhà cung cấp: cho phép nhân viên quản lý kho xem, xóa, sửa, lưu được thông tin của các nhà cung cấp để có thể liên hệ đặt hàng

**Đối với người quản lý:**

Quản lý kho: cho phép người quản lý xem, thêm, xóa, sửa các nguyên liệu trong kho hàng, biết được số lượng của từng nguyên liệu còn trong kho hàng, ngày nhập hàng của nguyên liệu, hạn sử dụng.

Quản lý hóa đơn: cho phép người quản lý xem, thêm, xóa, sửa thông tin các hóa đơn để quản lý doanh thu của cửa hàng.

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

### **1.4. Tính giá tiền khuyến mãi cho khách hàng**

Có ba loại khuyến mãi khác nhau, trong đó:

1. Khuyến mãi 10% dành cho những dịp ngày phụ nữ Việt Nam, ngày quốc tế thiếu nhi,…
2. Khuyến mãi 15% dành cho những dịp lễ
3. Khuyến mãi 20% dành cho dịp kỷ niệm của quán

Tổng tiền hóa đơn được tính bằng công thức: Tiền hóa đơn – (Tiền hóa đơn \* Khuyến Mãi) = Tổng hóa đơn

Ví dụ: Tiền hóa đơn của khách hàng đó là 500.000 đồng. Khách hàng đó sử dụng mã khuyến mãi giảm 10% nhân dịp ngày phụ nữ Việt Nam. Vậy tổng tiền hóa đơn của khách hàng đó sẽ là 500.000 - (500.000 x 10%) = 450.000 đồng.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

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

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

A diagram of a flowchart

AI-generated content may be incorrect.

A diagram of a flowchart

AI-generated content may be incorrect.

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

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

* CHUC\_VU (MaChucVu, TenChucVu)
* NHAN\_VIEN (MaNhanVien, TenNhanVien, DiaChi, SoDienThoai, Email, MaChucVu)
* TAI\_KHOAN\_NHAN\_VIEN (MaTaiKhoan, MaNhanVien, TaiKhoan, MatKhau)
* KHACH\_HANG (MaKhachHang, TenKhachHang, DiaChi, SoDienThoai)
* KHUYEN\_MAI (MaKhuyenMai, TenKhuyenMai, MaKhachHang, DaDung, NgayHetHan)
* MON\_AN (MaMonAn, TenMonAn, Gia, SoLuongHienCo)
* HOA\_DON (MaHoaDon, MaNhanVien, MaKhachHang, NgayLap, TongTien)
* CHI\_TIET\_HOA\_DON (MaChiTietHoaDon, MaHoaDon, MaMonAn, SoLuong, ThanhTien)
* NGUYEN\_LIEU (MaNguyenLieu, TenNguyenLieu, DonViTinh, Gia, MaNhaCungCap)
* LUU\_TRU (MaLuuTru, MaNhanVien, MaNguyenLieu, SoLuong, NgayHetHan)
* NHAP\_KHO (MaNhapKho, MaNhanVien, MaNguyenLieu, NgayNhap, SoLuong, TongTien, SoNgayHetHan)
* XUAT\_KHO (MaXuatKho, MaNhanVien, MaNguyenLieu, SoLuong, NgayXuat, NguyenNhanXuatKho, MaLuuTru)
* NHA\_CUNG\_CAP (MaNhaCungCap, TenNhaCungCap, DiaChi, SoDienThoai, Email)

1. **Các ràng buộc cần có**

PK: PRIMARY KEY

FK: FOREIGN KEY

NN: NOT NULL

C: CHECK

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | CHUC\_VU | PK: MaChucVu  NN: TenChucVu |
| 2 | NHAN\_VIEN | PK: MaNhanVien  FK: MaChucVu đến CHUC\_VU  NN: TenNhanVien, DiaChi, SoDienThoai, Email, MatKhau |
| 3 | KHACH\_HANG | PK: MaKhachHang  NN: TenKhachHang, DiaChi, SoDienThoai |
| 4 | KHUYEN\_MAI | PK: MaKhuyenMai  FK: MaKhachHang đến KHACH\_HANG  NN: TenKhuyenMai, NgayHetHan  C: DaDung IN (0,1) |
| 5 | MON\_AN | PK: MaMonAn  NN: TenMonAn, Gia, SoLuongHienCo  C: Gia>=0, SoLuongHienCo>=0 |
| 6 | HOA\_DON | PK: MaHoaDon  FK: MaNhanVien đến NHAN\_VIEN MaKhachHang đên KHACH\_HANG  NN: NgayLap, TongTien  C: TongTien>=0 |
| 7 | CHI\_TIET\_HOA\_DON | PK: MaChiTietHoaDon  FK: MaHoaDon đến HOA\_DON  NN: MaMonAn, SoLuong, ThanhTien  C: SoLuong>=0, ThanhTien>=0 |
| 8 | NGUYEN\_LIEU | PK: MaNguyenLieu  FK: MaNhaCungCap đên NHA\_CUNG\_CAP  NN: TenNguyenLieu, DonViTinh, Gia  C: Gia>=0 |
| 9 | LUU\_TRU | PK: MaLuuTru  FK: MaNhanVien đến NHANVIEN MaNguyenLieu đến NGUYEN\_LIEU  NN: SoLuong, NgayHetHan |
| 10 | NHAP\_KHO | PK: MaNhapKho  FK: MaNhanVien đến NHANVIEN MaNguyenLieu đến NGUYEN\_LIEU  NN: NgayNhap, SoLuong, TongTien, SoNgayHetHan  C: SoLuong>=0, TongTien>=0, SoNgayHetHan>=0 |
| 11 | XUAT\_KHO | PK: MaXuatKho  FK: MaNhanVien đến NHAN\_VIEN MaNguyenLieu đến NGUYEN\_LIEU MaLuuTru đến LUU\_TRU  NN: NgayXuat, SoLuong, NguyenNhanXuatKho  C: SoLuong>=0 |
| 12 | NHA\_CUNG\_CAP | PK: MaNhaCungCap  NN: TenNhaCungCap, DiaChi, SoDienThoai, Email |
| 13 | TAI\_KHOAN\_NHAN\_VIEN | PK: MaTaiKhoan  FK: MaNhanVien đến NHAN\_VIEN  NN: TaiKhoan, MatKhau  C: UNIQUE(MaNhanVien) |

1. **Cài đặt các CSDL và các ràng buộc**

Bảng Chức Vụ

|  |
| --- |
| CREATE TABLE [ChucVus] (  [MaChucVu] int NOT NULL IDENTITY,  [TenChucVu] nvarchar(100) NOT NULL,  CONSTRAINT [PK\_ChucVus] PRIMARY KEY ([MaChucVu])  ); |

Bảng Nhân Viên

|  |
| --- |
| CREATE TABLE [NhanViens] (  [MaNhanVien] int NOT NULL IDENTITY,  [TenNhanVien] nvarchar(max) NOT NULL,  [DiaChi] nvarchar(max) NOT NULL,  [SoDienThoai] nvarchar(max) NOT NULL,  [Email] nvarchar(max) NOT NULL,  [MaChucVu] int NULL,  CONSTRAINT [PK\_NhanViens] PRIMARY KEY ([MaNhanVien]),  CONSTRAINT [FK\_NhanViens\_ChucVus\_MaChucVu] FOREIGN KEY ([MaChucVu]) REFERENCES [ChucVus] ([MaChucVu])  ); |

Bảng Tài Khoản Nhân Viên

CREATE TABLE [dbo].[TaiKhoanNhanViens] (

[MaTaiKhoan] INT IDENTITY (1, 1) NOT NULL,

[MaNhanVien] INT NOT NULL,

[TaiKhoan] NVARCHAR (MAX) NULL,

CONSTRAINT [PK\_TaiKhoanNhanViens] PRIMARY KEY CLUSTERED ([MaTaiKhoan] ASC),

CONSTRAINT [FK\_TaiKhoanNhanViens\_NhanViens\_MaNhanVien] FOREIGN KEY ([MaNhanVien]) REFERENCES [dbo].[NhanViens] ([MaNhanVien])

);

Bảng Khách Hàng

|  |
| --- |
| CREATE TABLE [KhachHangs] (  [MaKhachHang] int NOT NULL IDENTITY,  [TenKhachHang] nvarchar(100) NOT NULL,  [DiaChi] nvarchar(max) NULL,  [SoDienThoai] nvarchar(20) NOT NULL,  CONSTRAINT [PK\_KhachHangs] PRIMARY KEY ([MaKhachHang])  ); |

Bảng Khuyến Mãi

|  |
| --- |
| CREATE TABLE [KhuyenMais] (  [MaKhuyenMai] int NOT NULL IDENTITY,  [TenKhuyenMai] nvarchar(100) NOT NULL,  [MaKhachHang] int NOT NULL,  [DaDung] bit NOT NULL,  [NgayHetHan] datetime2 NOT NULL,  CONSTRAINT [PK\_KhuyenMais] PRIMARY KEY ([MaKhuyenMai]),  CONSTRAINT [FK\_KhuyenMais\_KhachHangs\_MaKhachHang] FOREIGN KEY ([MaKhachHang]) REFERENCES [KhachHangs] ([MaKhachHang])  ); |

Bảng Món Ăn

|  |
| --- |
| CREATE TABLE [MonAns] (  [MaMonAn] int NOT NULL IDENTITY,  [TenMonAn] nvarchar(100) NOT NULL,  [Gia] float NOT NULL,  [SoLuongHienCo] int NOT NULL,  CONSTRAINT [PK\_MonAns] PRIMARY KEY ([MaMonAn])  ); |

Bảng Hóa Đơn

|  |
| --- |
| CREATE TABLE [HoaDons] (  [MaHoaDon] int NOT NULL IDENTITY,  [MaNhanVien] int NOT NULL,  [MaKhachHang] int NOT NULL,  [NgayLap] datetime2 NOT NULL,  [TongTien] float NOT NULL,  CONSTRAINT [PK\_HoaDons] PRIMARY KEY ([MaHoaDon]),  CONSTRAINT [FK\_HoaDons\_KhachHangs\_MaKhachHang] FOREIGN KEY ([MaKhachHang]) REFERENCES [KhachHangs] ([MaKhachHang]),  CONSTRAINT [FK\_HoaDons\_NhanViens\_MaNhanVien] FOREIGN KEY ([MaNhanVien]) REFERENCES [NhanViens] ([MaNhanVien])  ); |

Bảng Chi Tiết Hóa Đơn

|  |
| --- |
| CREATE TABLE [ChiTietHoaDons] (  [MaChiTietHoaDon] int NOT NULL IDENTITY,  [MaHoaDon] int NOT NULL,  [MaMonAn] int NOT NULL,  [SoLuong] int NOT NULL,  [ThanhTien] float NOT NULL,  CONSTRAINT [PK\_ChiTietHoaDons] PRIMARY KEY ([MaChiTietHoaDon]),  CONSTRAINT [FK\_ChiTietHoaDons\_HoaDons\_MaHoaDon] FOREIGN KEY ([MaHoaDon]) REFERENCES [HoaDons] ([MaHoaDon]),  CONSTRAINT [FK\_ChiTietHoaDons\_MonAns\_MaMonAn] FOREIGN KEY ([MaMonAn]) REFERENCES [MonAns] ([MaMonAn])  ); |

Bảng Nguyên Liệu

|  |
| --- |
| CREATE TABLE [NguyenLieus] (  [MaNguyenLieu] int NOT NULL IDENTITY,  [TenNguyenLieu] nvarchar(100) NOT NULL,  [DonViTinh] nvarchar(20) NOT NULL,  [Gia] float NOT NULL,  [MaNhaCungCap] int NOT NULL,  CONSTRAINT [PK\_NguyenLieus] PRIMARY KEY ([MaNguyenLieu]),  CONSTRAINT [FK\_NguyenLieus\_NhaCungCaps\_MaNhaCungCap] FOREIGN KEY ([MaNhaCungCap]) REFERENCES [NhaCungCaps] ([MaNhaCungCap])  ); |

Bảng Lưu Trữ

|  |
| --- |
| CREATE TABLE [LuuTrus] (  [MaLuuTru] int NOT NULL IDENTITY,  [MaNhanVien] int NOT NULL,  [MaNguyenLieu] int NOT NULL,  [SoLuong] int NOT NULL,  [NgayHetHan] datetime2 NOT NULL,  CONSTRAINT [PK\_LuuTrus] PRIMARY KEY ([MaLuuTru]),  CONSTRAINT [FK\_LuuTrus\_NguyenLieus\_MaNguyenLieu] FOREIGN KEY ([MaNguyenLieu]) REFERENCES [NguyenLieus] ([MaNguyenLieu]),  CONSTRAINT [FK\_LuuTrus\_NhanViens\_MaNhanVien] FOREIGN KEY ([MaNhanVien]) REFERENCES [NhanViens] ([MaNhanVien])  ); |

Bảng Nhập Kho

|  |
| --- |
| CREATE TABLE [NhapKhos] (  [MaNhapKho] int NOT NULL IDENTITY,  [MaNhanVien] int NOT NULL,  [MaNguyenLieu] int NOT NULL,  [NgayNhap] datetime2 NOT NULL,  [SoLuong] int NOT NULL,  [TongTien] float NOT NULL,  [SoNgayHetHan] int NOT NULL,  CONSTRAINT [PK\_NhapKhos] PRIMARY KEY ([MaNhapKho]),  CONSTRAINT [FK\_NhapKhos\_NguyenLieus\_MaNguyenLieu] FOREIGN KEY ([MaNguyenLieu]) REFERENCES [NguyenLieus] ([MaNguyenLieu]),  CONSTRAINT [FK\_NhapKhos\_NhanViens\_MaNhanVien] FOREIGN KEY ([MaNhanVien]) REFERENCES [NhanViens] ([MaNhanVien])  ); |

Bảng Xuất Kho

|  |
| --- |
| CREATE TABLE [XuatKhos] (  [MaXuatKho] int NOT NULL IDENTITY,  [MaNhanVien] int NOT NULL,  [MaNguyenLieu] int NOT NULL,  [SoLuong] int NOT NULL,  [NgayXuat] datetime2 NOT NULL,  [NguyenNhanXuatKho] nvarchar(max) NOT NULL,  [MaLuuTru] int NOT NULL,  CONSTRAINT [PK\_XuatKhos] PRIMARY KEY ([MaXuatKho]),  CONSTRAINT [FK\_XuatKhos\_LuuTrus\_MaLuuTru] FOREIGN KEY ([MaLuuTru]) REFERENCES [LuuTrus] ([MaLuuTru]),  CONSTRAINT [FK\_XuatKhos\_NguyenLieus\_MaNguyenLieu] FOREIGN KEY ([MaNguyenLieu]) REFERENCES [NguyenLieus] ([MaNguyenLieu]),  CONSTRAINT [FK\_XuatKhos\_NhanViens\_MaNhanVien] FOREIGN KEY ([MaNhanVien]) REFERENCES [NhanViens] ([MaNhanVien])  ); |

Bảng Nhà Cung Cấp

|  |
| --- |
| CREATE TABLE [NhaCungCaps] (  [MaNhaCungCap] int NOT NULL IDENTITY,  [TenNhaCungCap] nvarchar(100) NOT NULL,  [DiaChi] nvarchar(100) NOT NULL,  [SoDienThoai] nvarchar(20) NOT NULL,  [Email] nvarchar(100) NOT NULL,  CONSTRAINT [PK\_NhaCungCaps] PRIMARY KEY ([MaNhaCungCap])  ); |

Bảng Tài Khoản Nhân Viên

-- Tài Khoản Nhân Viên

CREATE TABLE [dbo].[TaiKhoanNhanViens] (

[MaTaiKhoan] INT IDENTITY (1, 1) NOT NULL,

[MaNhanVien] INT NOT NULL,

[TaiKhoan] NVARCHAR (MAX) NULL,

[MatKhau] NVARCHAR (MAX) NULL,

CONSTRAINT [PK\_TaiKhoanNhanViens] PRIMARY KEY CLUSTERED ([MaTaiKhoan] ASC),

CONSTRAINT [FK\_TaiKhoanNhanViens\_NhanViens\_MaNhanVien] FOREIGN KEY ([MaNhanVien]) REFERENCES [dbo].[NhanViens] ([MaNhanVien])

);

A computer screen shot of a computer

AI-generated content may be incorrect.

1. **Các view**
2. **View xem thông tin khuyến mãi của khách hàng**

|  |
| --- |
| CREATE VIEW View\_KhuyenMai AS  SELECT  km.MaKhuyenMai,  km.TenKhuyenMai,  kh.MaKhachHang,  kh.TenKhachHang,  km.DaDung,  km.NgayHetHan,  -- Trạng thái khuyến mãi hiển thị dễ hiểu  CASE  WHEN km.DaDung = 1 THEN N'Đã dùng'  WHEN km.NgayHetHan < GETDATE() THEN N'Hết hạn'  ELSE N'Còn hạn'  END AS TrangThai  FROM KhuyenMais km  JOIN KhachHangs kh ON km.MaKhachHang = kh.MaKhachHang; |

1. **View xem thông tin nhân viên**

|  |
| --- |
| CREATE VIEW View\_NhanVien AS  SELECT  nv.MaNhanVien,  nv.TenNhanVien,  nv.DiaChi,  nv.SoDienThoai,  nv.Email,  cv.TenChucVu  FROM NhanViens nv  LEFT JOIN ChucVus cv ON nv.MaChucVu = cv.MaChucVu; |

1. **Các Trigger**
2. **Trigger Tăng số lượng món ăn nếu đã có trong hóa đơn đó**

|  |
| --- |
| CREATE TRIGGER trg\_MergeChiTietHoaDonIfExist  ON ChiTietHoaDons  INSTEAD OF INSERT  AS  BEGIN  SET NOCOUNT ON;  DECLARE @MaHoaDon INT,  @MaMonAn INT,  @SoLuong INT,  @Gia FLOAT,  @ThanhTien FLOAT;  SELECT  @MaHoaDon = MaHoaDon,  @MaMonAn = MaMonAn,  @SoLuong = SoLuong  FROM inserted;  SELECT @Gia = Gia FROM MonAns WHERE MaMonAn = @MaMonAn;  SET @ThanhTien = @SoLuong \* @Gia;  IF EXISTS (  SELECT 1 FROM ChiTietHoaDons  WHERE MaHoaDon = @MaHoaDon AND MaMonAn = @MaMonAn  )  BEGIN  UPDATE ChiTietHoaDons  SET SoLuong = SoLuong + @SoLuong,  ThanhTien = (SoLuong + @SoLuong) \* @Gia  WHERE MaHoaDon = @MaHoaDon AND MaMonAn = @MaMonAn;  END  ELSE  BEGIN  INSERT INTO ChiTietHoaDons (MaHoaDon, MaMonAn, SoLuong, ThanhTien)  VALUES (@MaHoaDon, @MaMonAn, @SoLuong, @ThanhTien);  END  END; |

1. **Trigger Kiểm Tra Số Điện Thoại và Email Trùng Lặp Của Nhân Viên**

|  |
| --- |
| CREATE TRIGGER trg\_KiemTraTrungLapNhanVien  ON NhanViens  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @ErrorMessage NVARCHAR(255);  -- Kiểm tra trùng lặp số điện thoại  IF EXISTS (  SELECT I.SoDienThoai  FROM inserted I  JOIN NhanViens N ON I.SoDienThoai = N.SoDienThoai AND I.MaNhanVien <> N.MaNhanVien  )  BEGIN  SET @ErrorMessage = N'Lỗi: Số điện thoại đã tồn tại!';  THROW 51000, @ErrorMessage, 1;  END  -- Kiểm tra trùng lặp email  IF EXISTS (  SELECT I.Email  FROM inserted I  JOIN NhanViens N ON I.Email = N.Email AND I.MaNhanVien <> N.MaNhanVien  )  BEGIN  SET @ErrorMessage = N'Lỗi: Email đã tồn tại!';  THROW 51000, @ErrorMessage, 1;  END  END; |

1. **Trigger Kiểm Tra Tên Món Ăn Trùng Lặp**

|  |
| --- |
| CREATE TRIGGER trg\_KiemTraTrungLapMonAn  ON MonAns  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @ErrorMessage NVARCHAR(255);  -- Kiểm tra trùng lặp tên món ăn  IF EXISTS (  SELECT I.TenMonAn  FROM inserted I  JOIN MonAns M ON I.TenMonAn = M.TenMonAn AND I.MaMonAn <> M.MaMonAn  )  BEGIN  SET @ErrorMessage = N'Lỗi: Tên món ăn đã tồn tại!';  THROW 51000, @ErrorMessage, 1;  END  END; |

1. **Trigger Kiểm Tra Số Điện Thoại Trùng Lặp Của Khách Hàng**

|  |
| --- |
| CREATE TRIGGER trg\_KiemTraTrungLapSDT\_KhachHang  ON KhachHangs  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  DECLARE @ErrorMessage NVARCHAR(255);  -- Kiểm tra trùng số điện thoại  IF EXISTS (  SELECT I.SoDienThoai  FROM inserted I  JOIN KhachHangs K ON I.SoDienThoai = K.SoDienThoai AND I.MaKhachHang <> K.MaKhachHang  )  BEGIN  SET @ErrorMessage = N'Lỗi: Số điện thoại đã tồn tại!';  THROW 51000, @ErrorMessage, 1;  END  END; |

# **CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

## **1. Kết nối cơ sở dữ liệu**

public class SqlConnectionData

{

private static string defaultConn = @"Data Source=HUNGBUI\PJDBMS;Initial Catalog=RestaurantManagement;Integrated Security=True";

// Connection string có thể thay đổi khi người dùng đăng nhập thành công

public static string connectionString = defaultConn;

public static SqlConnection Connect()

{

SqlConnection conn = new SqlConnection(connectionString);

return conn;

}

// Hàm để cập nhật connection string khi đăng nhập

public static void UpdateConnectionString(string username, string password)

{

SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder

{

DataSource = "HUNGBUI\\PJDBMS",

InitialCatalog = "RestaurantManagement",

UserID = username,

Password = password,

IntegratedSecurity = false

};

connectionString = builder.ConnectionString;

}

}

## **2. Danh mục hóa đơn**

- View hiển thị hóa đơn

CREATE VIEW View\_HoaDon AS

SELECT

hd.MaHoaDon,

nv.TenNhanVien,

kh.TenKhachHang,

hd.NgayLap,

hd.TongTien

FROM HoaDons hd

INNER JOIN NhanViens nv ON hd.MaNhanVien = nv.MaNhanVien

INNER JOIN KhachHangs kh ON hd.MaKhachHang = kh.MaKhachHang

Thực thi trên c#

public static DataTable Get\_ListHoaDonDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_HoaDon", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

**Thêm hóa đơn**

Procedure thêm hóa đơn

CREATE PROCEDURE sp\_ThemHoaDon

@MaNhanVien INT,

@MaKhachHang INT,

@NgayLap DATETIME,

@TongTien FLOAT

AS

BEGIN

INSERT INTO HoaDons (MaNhanVien, MaKhachHang, NgayLap, TongTien)

VALUES (@MaNhanVien, @MaKhachHang, @NgayLap, @TongTien)

END

Thực thi trên C#

CREATE PROCEDURE proc\_ThemHoaDon

@MaNhanVien INT,

@MaKhachHang INT,

@NgayLap DATETIME2(7)

AS

BEGIN

INSERT INTO HoaDons (MaNhanVien, MaKhachHang, NgayLap, TongTien)

VALUES (@MaNhanVien, @MaKhachHang, @NgayLap, 0);

END

**Cập nhật thông tin hóa đơn**

- Procedure cập nhật thông tin hóa đơn

CREATE PROCEDURE proc\_CapNhatThongTinHoaDon

@MaNhanVien INT,

@MaHoaDon INT,

@MaKhachHang INT,

@NgayLap DATETIME2(7)

AS

BEGIN

UPDATE HoaDons

SET

MaNhanVien = @MaNhanVien,

MaKhachHang = @MaKhachHang,

NgayLap = @NgayLap

WHERE MaHoaDon = @MaHoaDon;

END

- Thực thi trên C#

public static void Update\_InfoHoaDonDTO(HoaDon hoaDon)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaHoaDon", hoaDon.MaHoaDon);

cmd.Parameters.AddWithValue("@MaNhanVien", hoaDon.MaNhanVien);

cmd.Parameters.AddWithValue("@MaKhachHang", hoaDon.MaKhachHang);

cmd.Parameters.AddWithValue("@NgayLap", hoaDon.NgayLap);

cmd.ExecuteNonQuery();

conn.Close();

}

**Xóa hóa đơn**

- Procedure xóa hóa đơn

CREATE PROCEDURE proc\_XoaHoaDon

@MaHoaDon INT

AS

BEGIN

IF EXISTS (SELECT 1 FROM ChiTietHoaDons WHERE MaHoaDon = @MaHoaDon)

BEGIN

DELETE FROM ChiTietHoaDons WHERE MaHoaDon = @MaHoaDon;

END

DELETE FROM HoaDons WHERE MaHoaDon = @MaHoaDon;

END

Thực thi trên C#

public static void Delete\_HoaDonDTO(int maHD)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaHoaDon", maHD);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger kiểm tra số lượng món ăn trước khi thêm hóa đơn để ngănKha không cho tạo hóa đơn nếu món ăn đã hết hàng.**

CREATE TRIGGER trg\_Check\_SoLuongMonAn\_Before\_Insert\_HoaDon

ON HoaDons

INSTEAD OF INSERT

AS

BEGIN

DECLARE @MaHoaDon INT, @MaMonAn INT, @SoLuongDat INT;

SELECT @MaHoaDon = inserted.MaHoaDon, @MaMonAn = MA.MaMonAn, @SoLuongDat = MA.SoLuongHienCo

FROM inserted

JOIN MonAns MA ON MA.MaMonAn = inserted.MaHoaDon;

IF @SoLuongDat <= 0

BEGIN

PRINT 'Không thể tạo hóa đơn vì món ăn đã hết hàng!';

ROLLBACK TRANSACTION;

RETURN;

END;

INSERT INTO HoaDons (MaHoaDon, MaNhanVien, MaKhachHang, NgayLap, TongTien)

SELECT MaHoaDon, MaNhanVien, MaKhachHang, NgayLap, TongTien FROM inserted;

END;

GO

DISABLE TRIGGER [dbo].[trg\_Check\_SoLuongMonAn\_Before\_Insert\_HoaDon]

ON [dbo].[HoaDons];

## **3. Chi Tiết Hóa Đơn**

- Hiển thị chi tiết hóa đơn

CREATE PROCEDURE proc\_LayCTHD\_MaHoaDon

@MaHoaDon INT

AS

BEGIN

SELECT

cthd.MaChiTietHoaDon,

cthd.MaHoaDon,

ma.TenMonAn,

cthd.SoLuong,

ma.Gia,

cthd.ThanhTien

FROM ChiTietHoaDons cthd

INNER JOIN MonAns ma ON cthd.MaMonAn = ma.MaMonAn

WHERE cthd.MaHoaDon = @MaHoaDon

END

Thực thi trên C#

public static DataTable Get\_ListChiTietHoaDonDTO(int maHD)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_LayCTHD\_MaHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaHoaDon", maHD);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Cập nhật chi tiết hóa đơn

CREATE PROCEDURE proc\_CapNhatThongTinChiTietHoaDon

@MaChiTietHoaDon INT,

@SoLuong INT

AS

BEGIN

UPDATE ChiTietHoaDons

SET

SoLuong = @SoLuong

WHERE MaChiTietHoaDon = @MaChiTietHoaDon;

END

Thực thi trên C#

public static void Update\_InfoChiTietHoaDonDTO(ChiTietHoaDon chiTietHoaDon)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinChiTietHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaChiTietHoaDon", chiTietHoaDon.MaChiTietHoaDon);

cmd.Parameters.AddWithValue("@SoLuong", chiTietHoaDon.SoLuong);

cmd.ExecuteNonQuery();

conn.Close();

}

- Thêm chi tiết hóa đơn

public static void Add\_ChiTietHoaDonDTO(ChiTietHoaDon chiTietHoaDon)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemChiTietHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaHoaDon", chiTietHoaDon.MaHoaDon);

cmd.Parameters.AddWithValue("@MaMonAn", chiTietHoaDon.MaMonAn);

cmd.Parameters.AddWithValue("@SoLuong", chiTietHoaDon.SoLuong);

cmd.ExecuteNonQuery();

conn.Close();

}

Thực thi trên C#

REATE PROCEDURE proc\_ThemChiTietHoaDon

@MaHoaDon INT,

@MaMonAn INT,

@SoLuong INT

AS

BEGIN

INSERT INTO ChiTietHoaDons (MaHoaDon, MaMonAn, SoLuong, ThanhTien)

VALUES (@MaHoaDon, @MaMonAn, @SoLuong, 0);

END

- Xóa chi tiết hóa đơn

CREATE PROCEDURE proc\_XoaChiTietHoaDon

@MaChiTietHoaDon INT

AS

BEGIN

DELETE FROM ChiTietHoaDons WHERE MaChiTietHoaDon = @MaChiTietHoaDon;

END

Thực thi trên C#

public static void Delete\_ChiTietHoaDonDTO(int maCTHD)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaChiTietHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaChiTietHoaDon", maCTHD);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger Tăng số lượng món ăn nếu đã có trong hóa đơn đó**

CREATE TRIGGER trg\_MergeChiTietHoaDonIfExist

ON ChiTietHoaDons

INSTEAD OF INSERT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @MaHoaDon INT,

@MaMonAn INT,

@SoLuong INT,

@Gia FLOAT,

@ThanhTien FLOAT;

SELECT

@MaHoaDon = MaHoaDon,

@MaMonAn = MaMonAn,

@SoLuong = SoLuong

FROM inserted;

SELECT @Gia = Gia FROM MonAns WHERE MaMonAn = @MaMonAn;

SET @ThanhTien = @SoLuong \* @Gia;

IF EXISTS (

SELECT 1 FROM ChiTietHoaDons

WHERE MaHoaDon = @MaHoaDon AND MaMonAn = @MaMonAn

)

BEGIN

UPDATE ChiTietHoaDons

SET SoLuong = SoLuong + @SoLuong,

ThanhTien = (SoLuong + @SoLuong) \* @Gia

WHERE MaHoaDon = @MaHoaDon AND MaMonAn = @MaMonAn;

END

ELSE

BEGIN

INSERT INTO ChiTietHoaDons (MaHoaDon, MaMonAn, SoLuong, ThanhTien)

VALUES (@MaHoaDon, @MaMonAn, @SoLuong, @ThanhTien);

END

END;

**Trigger tự động cập nhật thành tiền trong chi tiết hóa đơn sau mỗi lần thêm món vào hóa đơn hoặc sửa số lượng món trong hóa đơn**

CREATE TRIGGER trg\_UpdateThanhTien

ON ChiTietHoaDons

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

UPDATE cthd

SET cthd.ThanhTien = cthd.SoLuong \* ma.Gia

FROM ChiTietHoaDons cthd

INNER JOIN inserted i ON cthd.MaChiTietHoaDon = i.MaChiTietHoaDon

INNER JOIN MonAns ma ON i.MaMonAn = ma.MaMonAn;

END;

**Trigger cập nhật tổng tiền của hóa đơn dựa trên chi tiết hóa đơn**

-- trigger cập nhật tổng tiền của hóa đơn dựa trên cthd

CREATE TRIGGER trg\_UpdateTongTien

ON ChiTietHoaDons

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @MaHoaDonTable TABLE (MaHoaDon INT);

-- Trường hợp INSERT và UPDATE

INSERT INTO @MaHoaDonTable(MaHoaDon)

SELECT DISTINCT MaHoaDon FROM inserted;

-- Trường hợp DELETE

INSERT INTO @MaHoaDonTable(MaHoaDon)

SELECT DISTINCT MaHoaDon FROM deleted;

-- Cập nhật lại tổng tiền trong bảng HoaDons

UPDATE hd

SET TongTien = ISNULL((

SELECT SUM(ThanhTien)

FROM ChiTietHoaDons cthd

WHERE cthd.MaHoaDon = hd.MaHoaDon

), 0)

FROM HoaDons hd

WHERE hd.MaHoaDon IN (SELECT MaHoaDon FROM @MaHoaDonTable);

END;

## **4. Nhân viên**

- Hiển thị thông tin nhân viên

CREATE VIEW View\_NhanVien AS

SELECT

nv.MaNhanVien,

nv.TenNhanVien,

nv.DiaChi,

nv.SoDienThoai,

nv.Email,

cv.TenChucVu

FROM NhanViens nv

LEFT JOIN ChucVus cv ON nv.MaChucVu = cv.MaChucVu;

Thực thi trên C#

public static DataTable Get\_ListNhanVienDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_NhanVien", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Cập nhật thông tin nhân viên

Procedure cập nhật thông tin nhân viên

CREATE PROCEDURE proc\_CapNhatThongTinNhanVien

@MaNhanVien INT,

@TenNhanVien NVARCHAR(100),

@MaChucVu NVARCHAR(50),

@SoDienThoai NVARCHAR(15),

@DiaChi NVARCHAR(200),

@Email NVARCHAR(100)

AS

BEGIN

UPDATE NhanViens

SET

TenNhanVien = @TenNhanVien,

MaChucVu = @MaChucVu,

SoDienThoai = @SoDienThoai,

DiaChi = @DiaChi,

Email = @Email

WHERE MaNhanVien = @MaNhanVien;

END

Thực thi trên C#

public static void Update\_InfoNhanVienDTO(NhanVien nhanVien, int cv)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinNhanVien", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhanVien", nhanVien.MaNhanVien);

cmd.Parameters.AddWithValue("@TenNhanVien", nhanVien.TenNhanVien);

cmd.Parameters.AddWithValue("@DiaChi", nhanVien.DiaChi);

cmd.Parameters.AddWithValue("@SoDienThoai", nhanVien.SoDienThoai);

cmd.Parameters.AddWithValue("@Email", nhanVien.Email);

cmd.Parameters.AddWithValue("@MaChucVu", cv);

cmd.ExecuteNonQuery();

conn.Close();

}

- Thêm nhân viên

Procedure thêm nhân viên

CREATE PROCEDURE proc\_ThemNhanVien

@TenNhanVien NVARCHAR(100),

@MaChucVu NVARCHAR(50),

@SoDienThoai NVARCHAR(15),

@DiaChi NVARCHAR(200),

@Email NVARCHAR(100)

AS

BEGIN

INSERT INTO NhanViens (TenNhanVien, MaChucVu, SoDienThoai, DiaChi, Email)

VALUES (@TenNhanVien, @MaChucVu, @SoDienThoai, @DiaChi, @Email);

DECLARE @MaNhanVienMoi INT = SCOPE\_IDENTITY();

INSERT INTO TaiKhoanNhanViens (MaNhanVien, TaiKhoan, MatKhau)

VALUES (@MaNhanVienMoi, Null, Null);

END

Thực thi trên C#

public static void Add\_NhanVienDTO(NhanVien nhanVien)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemNhanVien", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TenNhanVien", nhanVien.TenNhanVien);

cmd.Parameters.AddWithValue("@DiaChi", nhanVien.DiaChi);

cmd.Parameters.AddWithValue("@SoDienThoai", nhanVien.SoDienThoai);

cmd.Parameters.AddWithValue("@Email", nhanVien.Email);

cmd.Parameters.AddWithValue("@MaChucVu", nhanVien.MaChucVu);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa nhân viên

CREATE PROCEDURE proc\_XoaNhanVien

@MaNhanVien INT

AS

BEGIN

IF EXISTS (SELECT 1 FROM TaiKhoanNhanViens WHERE MaNhanVien = @MaNhanVien)

BEGIN

DELETE FROM TaiKhoanNhanViens WHERE MaNhanVien = @MaNhanVien;

END

DELETE FROM NhanViens WHERE MaNhanVien = @MaNhanVien;

END

Thực thi trên C#

public static void Delete\_NhanVienDTO(int maNV)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaNhanVien", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhanVien", maNV);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger kiểm tra trùng lặp email và số điện thoại của nhân viên**

CREATE TRIGGER trg\_KiemTraTrungLapNhanVien

ON NhanViens

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @ErrorMessage NVARCHAR(255);

-- Kiểm tra trùng lặp số điện thoại

IF EXISTS (

SELECT I.SoDienThoai

FROM inserted I

JOIN NhanViens N ON I.SoDienThoai = N.SoDienThoai AND I.MaNhanVien <> N.MaNhanVien

)

BEGIN

SET @ErrorMessage = N'Lỗi: Số điện thoại đã tồn tại!';

THROW 51000, @ErrorMessage, 1;

END

-- Kiểm tra trùng lặp email

IF EXISTS (

SELECT I.Email

FROM inserted I

JOIN NhanViens N ON I.Email = N.Email AND I.MaNhanVien <> N.MaNhanVien

)

BEGIN

SET @ErrorMessage = N'Lỗi: Email đã tồn tại!';

THROW 51000, @ErrorMessage, 1;

END

END;

## **5. Nhà cung cấp**

- Hiển thị nhà cung cấp

Thực thi trên C#

public static DataTable Get\_ListNhaCungCapDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM NhaCungCaps", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Thêm nhà cung cấp

CREATE PROCEDURE proc\_ThemNhaCungCap

@TenNhaCungCap NVARCHAR(100),

@SoDienThoai NVARCHAR(15),

@DiaChi NVARCHAR(200),

@Email NVARCHAR(100)

AS

BEGIN

INSERT INTO NhaCungCaps (TenNhaCungCap, SoDienThoai, DiaChi, Email)

VALUES (@TenNhaCungCap, @SoDienThoai, @DiaChi, @Email);

END

Thực thi trên C#

public static void Add\_NhaCungCapDTO(NhaCungCap nhaCungCap)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemNhaCungCap", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TenNhaCungCap", nhaCungCap.TenNCC);

cmd.Parameters.AddWithValue("@DiaChi", nhaCungCap.DiaChi);

cmd.Parameters.AddWithValue("@SoDienThoai", nhaCungCap.SoDienThoai);

cmd.Parameters.AddWithValue("@Email", nhaCungCap.Email);

cmd.ExecuteNonQuery();

conn.Close();

}

- Cập nhật nhà cung cấp

Procedure cập nhật nhà cung cấp

CREATE PROCEDURE proc\_CapNhatThongTinNhaCungCap

@MaNhaCungCap INT,

@TenNhaCungCap NVARCHAR(100),

@SoDienThoai NVARCHAR(15),

@DiaChi NVARCHAR(200),

@Email NVARCHAR(100)

AS

BEGIN

UPDATE NhaCungCaps

SET

TenNhaCungCap = @TenNhaCungCap,

SoDienThoai = @SoDienThoai,

DiaChi = @DiaChi,

Email = @Email

WHERE MaNhaCungCap = @MaNhaCungCap;

END

Thực thi trên C#

public static void Update\_InfoNhaCungCapDTO(NhaCungCap nhaCungCap)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinNhaCungCap", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhaCungCap", nhaCungCap.MaNCC);

cmd.Parameters.AddWithValue("@TenNhaCungCap", nhaCungCap.TenNCC);

cmd.Parameters.AddWithValue("@DiaChi", nhaCungCap.DiaChi);

cmd.Parameters.AddWithValue("@SoDienThoai", nhaCungCap.SoDienThoai);

cmd.Parameters.AddWithValue("@Email", nhaCungCap.Email);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa nhà cung cấp

CREATE PROCEDURE proc\_XoaNhaCungCap

@MaNhaCungCap INT

AS

BEGIN

IF EXISTS (SELECT 1 FROM NguyenLieus WHERE MaNhaCungCap = @MaNhaCungCap)

BEGIN

DELETE FROM NguyenLieus WHERE MaNhaCungCap = @MaNhaCungCap;

END

DELETE FROM NhaCungCaps WHERE MaNhaCungCap = @MaNhaCungCap;

END

Thực thi trên C#

public static void Delete\_NhaCungCapDTO(int maNCC)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaNhaCungCap", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhaCungCap", maNCC);

cmd.ExecuteNonQuery();

conn.Close();

}

## **6. Khuyến Mãi**

- Hiển thị thông tin khuyến mãi

View hiển thị thông tin khuyến mãi

CREATE VIEW View\_KhuyenMai AS

SELECT

km.MaKhuyenMai,

km.TenKhuyenMai,

kh.MaKhachHang,

kh.TenKhachHang,

km.DaDung,

km.NgayHetHan,

-- Trạng thái khuyến mãi hiển thị dễ hiểu

CASE

WHEN km.DaDung = 1 THEN N'Đã dùng'

WHEN km.NgayHetHan < GETDATE() THEN N'Hết hạn'

ELSE N'Còn hạn'

END AS TrangThai

FROM KhuyenMais km

JOIN KhachHangs kh ON km.MaKhachHang = kh.MaKhachHang;

Thực thi trên C#

public static DataTable Get\_ListKhuyenMaiDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_KhuyenMai", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Cập nhật thông tin khuyến mãi

Procedure cập nhật thông tin khuyến mãi

CREATE PROCEDURE sp\_SuaKhuyenMai

@MaKhuyenMai INT,

@TenKhuyenMai NVARCHAR(100),

@MaKhachHang INT,

@DaDung BIT,

@NgayHetHan DATE

AS

BEGIN

UPDATE KhuyenMais

SET

TenKhuyenMai = @TenKhuyenMai,

MaKhachHang = @MaKhachHang,

DaDung = @DaDung,

NgayHetHan = @NgayHetHan

WHERE MaKhuyenMai = @MaKhuyenMai

END

Thực thi trên C#

public static void Update\_InfoKhuyenMaiDTO(KhuyenMai khuyenMai)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("sp\_SuaKhuyenMai", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaKhuyenMai", khuyenMai.MaKhuyenMai);

cmd.Parameters.AddWithValue("@TenKhuyenMai", khuyenMai.TenKhuyenMai);

cmd.Parameters.AddWithValue("@DaDung", khuyenMai.DaDung);

cmd.Parameters.AddWithValue("@NgayHetHan", khuyenMai.NgayHetHan);

cmd.Parameters.AddWithValue("@MaKhachHang", khuyenMai.MaKhachHang);

cmd.ExecuteNonQuery();

conn.Close();

}

- Thêm khuyến mãi

Procedure thêm khuyến mãi

CREATE PROCEDURE sp\_ThemKhuyenMai

@TenKhuyenMai NVARCHAR(100),

@MaKhachHang INT,

@DaDung BIT,

@NgayHetHan DATE

AS

BEGIN

INSERT INTO KhuyenMais (TenKhuyenMai, MaKhachHang, DaDung, NgayHetHan)

VALUES (@TenKhuyenMai, @MaKhachHang, @DaDung, @NgayHetHan)

END

Thực thi trên C#

public static void Add\_KhuyenMaiDTO(KhuyenMai khuyenMai)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("sp\_ThemKhuyenMai", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TenKhuyenMai", khuyenMai.TenKhuyenMai);

cmd.Parameters.AddWithValue("@DaDung", khuyenMai.DaDung);

cmd.Parameters.AddWithValue("@NgayHetHan", khuyenMai.NgayHetHan);

cmd.Parameters.AddWithValue("@MaKhachHang", khuyenMai.MaKhachHang);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa khuyến mãi

Procedure xóa khuyến mãi

CREATE PROCEDURE sp\_XoaKhuyenMai

@MaKhuyenMai INT

AS

BEGIN

DELETE FROM KhuyenMais

WHERE MaKhuyenMai = @MaKhuyenMai

END

Thực thi trên C#

public static void Delete\_KhuyenMaiDTO(int maKM)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaKhuyenMai", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaKhuyenMai", maKM);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger tự chặn khuyến mãi nếu khuyến mãi tạo ra đã hết hạn hoặc đã dùng**

CREATE TRIGGER trg\_Block\_Invalid\_KhuyenMai

ON KhuyenMais

INSTEAD OF INSERT

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted

WHERE DaDung = 1 OR NgayHetHan < CAST(GETDATE() AS DATE)

)

BEGIN

THROW 50001, N'Lỗi: Không thể thêm khuyến mãi đã dùng hoặc đã hết hạn!', 1;

RETURN;

END

## **7. Món Ăn**

- Hiển thị danh mục món ăn

Thực thi trên C#

public static DataTable Get\_ListMonAnDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM MonAns", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Thêm món ăn

Procedure thêm món ăn

CREATE PROCEDURE proc\_ThemMonAn

@TenMonAn NVARCHAR(100),

@Gia FLOAT,

@SoLuong INT

AS

BEGIN

INSERT INTO MonAns (TenMonAn, Gia, SoLuongHienCo)

VALUES (@TenMonAn, @Gia, @SoLuong);

END

Thực thi trên C#

public static void Add\_MonAnDTO(MonAn monAn)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemMonAn", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TenMonAn", monAn.TenMonAn);

cmd.Parameters.AddWithValue("@Gia", monAn.Gia);

cmd.Parameters.AddWithValue("@SoLuong", monAn.SoLuongHienCo);

cmd.ExecuteNonQuery();

conn.Close();

}

- Cập nhật thông tin món ăn

Procedure cập nhật thông tin món ăn

CREATE PROCEDURE proc\_CapNhatThongTinMonAn

@MaMonAn INT,

@TenMonAn NVARCHAR(100),

@Gia FLOAT,

@SoLuong INT

AS

BEGIN

UPDATE MonAns

SET

TenMonAn = @TenMonAn,

Gia = @Gia,

SoLuongHienCo = @SoLuong

WHERE MaMonAn = @MaMonAn;

END

Thực thi trên C#

public static void Update\_InfoMonAnDTO(MonAn monAn)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinMonAn", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaMonAn", monAn.MaMonAn);

cmd.Parameters.AddWithValue("@TenMonAn", monAn.TenMonAn);

cmd.Parameters.AddWithValue("@Gia", monAn.Gia);

cmd.Parameters.AddWithValue("@SoLuong", monAn.SoLuongHienCo);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa món ăn

Procedure xóa món ăn

CREATE PROCEDURE proc\_XoaMonAn

@MaMonAn INT

AS

BEGIN

DELETE FROM MonAns WHERE MaMonAn = @MaMonAn;

END

Thực thi trên C#

public static void Delete\_MonAnDTO(int maMA)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaMonAn", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaMonAn", maMA);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger kiểm tra trùng lặp món**

CREATE TRIGGER trg\_KiemTraTrungLapMonAn

ON MonAns

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @ErrorMessage NVARCHAR(255);

-- Kiểm tra trùng lặp tên món ăn

IF EXISTS (

SELECT I.TenMonAn

FROM inserted I

JOIN MonAns M ON I.TenMonAn = M.TenMonAn AND I.MaMonAn <> M.MaMonAn

)

BEGIN

SET @ErrorMessage = N'Lỗi: Tên món ăn đã tồn tại!';

THROW 51000, @ErrorMessage, 1;

END

END;

## **8. Khách hàng**

- Hiển thị danh mục khách hàng

Thực thi trên C#

public static DataTable Get\_ListKhachHangDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM KhachHangs", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Thêm thông tin khách hàng

Procedure thêm thông tin khách hàng

CREATE PROCEDURE proc\_ThemKhachHang

@TenKhachHang NVARCHAR(100),

@DiaChi NVARCHAR(100),

@SoDienThoai NVARCHAR(100)

AS

BEGIN

INSERT INTO KhachHangs (TenKhachHang, DiaChi, SoDienThoai)

VALUES (@TenKhachHang, @DiaChi, @SoDienThoai);

END

Thực thi trên C#

public static void Add\_KhachHangDTO(KhachHang khachHang)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemKhachHang", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TenKhachHang", khachHang.TenKhachHang);

cmd.Parameters.AddWithValue("@DiaChi", khachHang.DiaChi);

cmd.Parameters.AddWithValue("@SoDienThoai", khachHang.SoDienThoai);

cmd.ExecuteNonQuery();

conn.Close();

}

- Cập nhật thông tin khách hàng

Procedure cập nhật thông tin khách hàng

CREATE PROCEDURE proc\_CapNhatThongTinKhachHang

@MaKhachHang INT,

@TenKhachHang NVARCHAR(100),

@DiaChi NVARCHAR(100),

@SoDienThoai NVARCHAR(100)

AS

BEGIN

UPDATE KhachHangs

SET

TenKhachHang = @TenKhachHang,

DiaChi = @DiaChi,

SoDienThoai = @SoDienThoai

WHERE MaKhachHang = @MaKhachHang;

END

Thực thi trên C#

public static void Update\_InfoKhachHangDTO(KhachHang khachHang)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinKhachHang", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaKhachHang", khachHang.MaKhachHang);

cmd.Parameters.AddWithValue("@TenKhachHang", khachHang.TenKhachHang);

cmd.Parameters.AddWithValue("@DiaChi", khachHang.DiaChi);

cmd.Parameters.AddWithValue("@SoDienThoai", khachHang.SoDienThoai);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa thông tin khách hàng

Procedure xóa thông tin khách hàng

CREATE PROCEDURE proc\_XoaKhachHang

@MaKhachHang INT

AS

BEGIN

IF EXISTS (SELECT 1 FROM KhuyenMais WHERE MaKhachHang = @MaKhachHang)

BEGIN

DELETE FROM KhuyenMais WHERE MaKhachHang = @MaKhachHang;

END

IF EXISTS (SELECT 1 FROM HoaDons WHERE MaKhachHang = @MaKhachHang)

BEGIN

DELETE FROM HoaDons WHERE MaKhachHang = @MaKhachHang;

END

DELETE FROM KhachHangs WHERE MaKhachHang = @MaKhachHang;

END

Thực thi trên C#

public static void Delete\_KhachHangDTO(int maKH)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaKhachHang", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaKhachHang", maKH);

cmd.ExecuteNonQuery();

conn.Close();

}

**Kiểm tra trùng lặp số điện thoại của khách hàng, nếu trùng thì sẽ không cho phép nhập**

CREATE TRIGGER trg\_KiemTraTrungLapSDT\_KhachHang

ON KhachHangs

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @ErrorMessage NVARCHAR(255);

-- Kiểm tra trùng số điện thoại

IF EXISTS (

SELECT I.SoDienThoai

FROM inserted I

JOIN KhachHangs K ON I.SoDienThoai = K.SoDienThoai AND I.MaKhachHang <> K.MaKhachHang

)

BEGIN

SET @ErrorMessage = N'Lỗi: Số điện thoại đã tồn tại!';

THROW 51000, @ErrorMessage, 1;

END

END;

## **9. Nguyên liệu**

- Hiển thị nguyên liệu

View hiển thị danh mục nguyên liệu

CREATE VIEW View\_NguyenLieu AS

SELECT

nl.MaNguyenLieu,

nl.TenNguyenLieu,

nl.DonViTinh,

nl.Gia,

ncc.TenNhaCungCap

FROM NguyenLieus nl

LEFT JOIN NhaCungCaps ncc ON nl.MaNhaCungCap = ncc.MaNhaCungCap;

Thực thi trên C#

public static DataTable Get\_ListNguyenLieuDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_NguyenLieu", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Cập nhật nguyên liệu

Procedure cập nhật nguyên liệu

CREATE PROCEDURE proc\_CapNhatThongTinNguyenLieu

@MaNguyenLieu INT,

@TenNguyenLieu NVARCHAR(100),

@DonViTinh NVARCHAR(50),

@Gia FLOAT,

@MaNhaCungCap INT

AS

BEGIN

UPDATE NguyenLieus

SET

TenNguyenLieu = @TenNguyenLieu,

DonViTinh = @DonViTinh,

Gia = @Gia,

MaNhaCungCap = @MaNhaCungCap

WHERE MaNguyenLieu = @MaNguyenLieu;

END

Thực thi trên C#

public static void Update\_InfoNguyenLieuDTO(NguyenLieu nguyenLieu)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinNguyenLieu", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNguyenLieu", nguyenLieu.MaNguyenLieu);

cmd.Parameters.AddWithValue("@TenNguyenLieu", nguyenLieu.TenNguyenLieu);

cmd.Parameters.AddWithValue("@DonViTinh", nguyenLieu.DonViTinh);

cmd.Parameters.AddWithValue("@Gia", nguyenLieu.Gia);

cmd.Parameters.AddWithValue("@MaNhaCungCap", nguyenLieu.MaNhaCungCap);

cmd.ExecuteNonQuery();

conn.Close();

}

- Thêm nguyên liệu

Procedure thêm nguyên liệu

CREATE PROCEDURE proc\_ThemNguyenLieu

@TenNguyenLieu NVARCHAR(100),

@DonViTinh NVARCHAR(50),

@Gia FLOAT,

@MaNhaCungCap INT

AS

BEGIN

INSERT INTO NguyenLieus(TenNguyenLieu, DonViTinh, Gia, MaNhaCungCap)

VALUES (@TenNguyenLieu, @DonViTinh, @Gia, @MaNhaCungCap);

END

Thực thi trên C#

public static void Add\_NguyenLieuDTO(NguyenLieu nguyenLieu)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemNguyenLieu", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TenNguyenLieu", nguyenLieu.TenNguyenLieu);

cmd.Parameters.AddWithValue("@DonViTinh", nguyenLieu.DonViTinh);

cmd.Parameters.AddWithValue("@Gia", nguyenLieu.Gia);

cmd.Parameters.AddWithValue("@MaNhaCungCap", nguyenLieu.MaNhaCungCap);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa nguyên liệu

Procedure xóa nguyên liệu

CREATE PROCEDURE proc\_XoaNguyenLieu

@MaNguyenLieu INT

AS

BEGIN

DELETE FROM NguyenLieus WHERE MaNguyenLieu = @MaNguyenLieu;

END

Thực thi trên C#

public static void Delete\_NguyenLieuDTO(int maNL)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaNguyenLieu", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNguyenLieu", maNL);

cmd.ExecuteNonQuery();

conn.Close();

}

## **10. Nhập kho**

- Hiển thị danh mục nhập kho

View hiển thị nhập kho

CREATE VIEW View\_NhapKho AS

SELECT

nk.MaNhapKho,

nv.TenNhanVien,

nl.TenNguyenLieu,

nl.Gia,

nk.NgayNhap,

nk.SoLuong,

nk.TongTien,

nk.SoNgayHetHan

FROM NhapKhos nk

INNER JOIN NhanViens nv ON nk.MaNhanVien = nv.MaNhanVien

INNER JOIN NguyenLieus nl ON nl.MaNguyenLieu = nk.MaNguyenLieu

Thực thi trên C#

public static DataTable Get\_ListNhapKhoDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_NhapKho", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Thêm nhập kho

Procedure thêm nhập kho

CREATE PROCEDURE proc\_ThemNhapKho

@MaNhanVien INT,

@MaNguyenLieu INT,

@SoLuong INT,

@NgayNhap DATETIME2(7),

@SoNgayHetHan INT

AS

BEGIN

INSERT INTO NhapKhos(MaNhanVien, MaNguyenLieu, NgayNhap, SoLuong, TongTien, SoNgayHetHan)

VALUES (@MaNhanVien, @MaNguyenLieu, @NgayNhap, @SoLuong, 0, @SoNgayHetHan);

END

Thực thi trên C#

public static void Add\_NhapKhoDTO(NhapKho NhapKho)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemNhapKho", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhanVien", NhapKho.MaNhanVien);

cmd.Parameters.AddWithValue("@MaNguyenLieu", NhapKho.MaNguyenLieu);

cmd.Parameters.AddWithValue("@SoLuong", NhapKho.SoLuong);

cmd.Parameters.AddWithValue("@NgayNhap", NhapKho.NgayNhap);

cmd.Parameters.AddWithValue("@SoNgayHetHan", NhapKho.SoNgayHetHan);

cmd.ExecuteNonQuery();

conn.Close();

}

- Cập nhật thông tin nhập kho

Procedure cập nhật thông tin nhập kho

CREATE PROCEDURE proc\_CapNhatThongTinNhapKho

@MaNhapKho INT,

@SoLuong INT

AS

BEGIN

UPDATE NhapKhos

SET

SoLuong = @SoLuong

WHERE MaNhapKho = @MaNhapKho;

END

Thực thi trên C#

public static void Update\_InfoNhapKhoDTO(NhapKho NhapKho)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinNhapKho", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhapKho", NhapKho.MaNhapKho);

cmd.Parameters.AddWithValue("@SoLuong", NhapKho.SoLuong);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa nhập kho

Procedure xóa nhập kho

CREATE PROCEDURE proc\_XoaNhapKho

@MaNhapKho INT

AS

BEGIN

DELETE FROM NhapKhos WHERE MaNhapKho = @MaNhapKho;

END

Thực thi trên C#

public static void Delete\_NhapKhoDTO(int maNhapKho)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaNhapKho", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhapKho", maNhapKho);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger tự động tính tổng tiền khi nhập kho**

CREATE TRIGGER trg\_UpdateTongTien\_NhapKho

ON NhapKhos

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

UPDATE nk

SET nk.TongTien = nk.SoLuong \* nl.Gia

FROM NhapKhos nk

INNER JOIN inserted i ON nk.MaNhapKho = i.MaNhapKho

INNER JOIN NguyenLieus nl ON i.MaNguyenLieu = nl.MaNguyenLieu;

END;

## **11. Xuất Kho**

- Hiển thị thông tin xuất kho

View hiển thị thông tin xuất kho

CREATE VIEW View\_XuatKho AS

SELECT

nk.MaXuatKho,

nv.TenNhanVien,

nl.TenNguyenLieu,

nk.NgayXuat,

nk.SoLuong,

nk.NguyenNhanXuatKho,

nk.MaLuuTru

FROM XuatKhos nk

INNER JOIN NhanViens nv ON nk.MaNhanVien = nv.MaNhanVien

INNER JOIN NguyenLieus nl ON nl.MaNguyenLieu = nk.MaNguyenLieu

Thực thi trên C#

public static DataTable Get\_ListXuatKhoDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_XuatKho", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Cập nhật thông tin xuất kho

Procedure cập nhật thông tin xuất kho

CREATE PROCEDURE proc\_CapNhatThongTinXuatKho

@MaXuatKho INT,

@SoLuong INT,

@NguyenNhan NVARCHAR(MAX)

AS

BEGIN

UPDATE XuatKhos

SET

SoLuong = @SoLuong,

NguyenNhanXuatKho = @NguyenNhan

WHERE MaXuatKho = @MaXuatKho;

END

Thực thi trên C#

public static void Update\_InfoXuatKhoDTO(XuatKho xuatKho)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinXuatKho", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaXuatKho", xuatKho.MaXuatKho);

cmd.Parameters.AddWithValue("@SoLuong", xuatKho.SoLuong);

cmd.Parameters.AddWithValue("@NguyenNhan", xuatKho.NguyenNhanXuatKho);

cmd.ExecuteNonQuery();

conn.Close();

}

- Thêm thông tin xuất kho

Procedure thêm thông tin xuất kho

CREATE PROCEDURE proc\_ThemXuatKho

@MaNhanVien INT,

@MaNguyenLieu INT,

@SoLuong INT,

@NgayXuat DATETIME2(7),

@NguyenNhan NVARCHAR(MAX),

@MaLuuTru INT

AS

BEGIN

INSERT INTO XuatKhos(MaNhanVien, MaNguyenLieu, SoLuong, NgayXuat, NguyenNhanXuatKho, MaLuuTru)

VALUES (@MaNhanVien, @MaNguyenLieu, @SoLuong, @NgayXuat, @NguyenNhan, @MaLuuTru);

END

Thực thi trên C#

public static void Add\_XuatKhoDTO(XuatKho xuatKho)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_ThemXuatKho", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhanVien", xuatKho.MaNhanVien);

cmd.Parameters.AddWithValue("@MaNguyenLieu", xuatKho.MaNguyenLieu);

cmd.Parameters.AddWithValue("@SoLuong", xuatKho.SoLuong);

cmd.Parameters.AddWithValue("@NgayXuat", xuatKho.NgayXuat);

cmd.Parameters.AddWithValue("@NguyenNhan", xuatKho.NguyenNhanXuatKho);

cmd.Parameters.AddWithValue("@MaLuuTru", xuatKho.MaLuuTru);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa thông tin xuất kho

Procedure xóa thông tin xuất kho

CREATE PROCEDURE proc\_XoaXuatKho

@MaXuatKho INT

AS

BEGIN

DELETE FROM XuatKhos WHERE MaXuatKho = @MaXuatKho;

END

Thực thi trên C#

public static void Delete\_XuatKhoDTO(int maXuatKho)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaXuatKho", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaXuatKho", maXuatKho);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger kiểm tra và cập nhật tồn kho khi xuất kho**

CREATE TRIGGER trg\_XuatKho\_Insert

ON XuatKhos

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

UPDATE lt

SET lt.SoLuong = lt.SoLuong - i.SoLuong

FROM LuuTrus lt

INNER JOIN inserted i ON lt.MaLuuTru = i.MaLuuTru;

IF EXISTS (SELECT 1 FROM LuuTrus WHERE SoLuong < 0)

BEGIN

RAISERROR(N'Số lượng trong kho không đủ để xuất!', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**Trigger kiểm tra và cập nhật tồn kho khi cập nhật thông tin xuất kho**

CREATE TRIGGER trg\_XuatKho\_Update

ON XuatKhos

AFTER UPDATE

AS

BEGIN

SET NOCOUNT ON;

-- Trừ số lượng cũ

UPDATE lt

SET lt.SoLuong = lt.SoLuong + d.SoLuongXuat

FROM LuuTrus lt

INNER JOIN deleted d ON lt.MaLuuTru = d.MaLuuTru;

-- Trừ số lượng mới

UPDATE lt

SET lt.SoLuong = lt.SoLuong - i.SoLuongXuat

FROM LuuTrus lt

INNER JOIN inserted i ON lt.MaLuuTru = i.MaLuuTru;

-- Kiểm tra không âm

IF EXISTS (SELECT 1 FROM LuuTrus WHERE SoLuong < 0)

BEGIN

RAISERROR(N'Cập nhật không hợp lệ: Số lượng trong kho âm!', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**Trigger khôi phục lại số lượng trong kho khi xoá phiếu xuất kho**

CREATE TRIGGER trg\_XuatKho\_Delete

ON XuatKhos

AFTER DELETE

AS

BEGIN

SET NOCOUNT ON;

UPDATE lt

SET lt.SoLuong = lt.SoLuong + d.SoLuongXuat

FROM LuuTrus lt

INNER JOIN deleted d ON lt.MaLuuTru = d.MaLuuTru;

END;

## **12. Lưu Trữ**

- Hiển thị danh sách lưu trữ

View hiển thị danh sách lưu trữ

--view luu tru

CREATE VIEW View\_LuuTru AS

SELECT

lt.MaLuuTru,

lt.MaNguyenLieu,

nv.TenNhanVien,

nl.TenNguyenLieu,

lt.NgayHetHan,

lt.SoLuong

FROM LuuTrus lt

INNER JOIN NhanViens nv ON lt.MaNhanVien = nv.MaNhanVien

INNER JOIN NguyenLieus nl ON nl.MaNguyenLieu = lt.MaNguyenLieu

Thực thi trên C#

public static DataTable Get\_ListLuuTruDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_LuuTru", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Cập nhật thông tin lưu trữ

Procedure cập nhật thông tin lưu trữ

CREATE PROCEDURE proc\_CapNhatThongTinLuuTru

@MaLuuTru INT,

@SoLuong INT,

@NgayHetHan DATETIME2(7)

AS

BEGIN

UPDATE LuuTrus

SET

SoLuong = @SoLuong,

NgayHetHan = @NgayHetHan

WHERE MaLuuTru = @MaLuuTru;

END

Thực thi trên C#

public static void Update\_InfoLuuTruDTO(LuuTru LuuTru)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinLuuTru", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaLuuTru", LuuTru.MaLuuTru);

cmd.Parameters.AddWithValue("@SoLuong", LuuTru.SoLuong);

cmd.Parameters.AddWithValue("@NgayHetHan", LuuTru.NgayHetHan);

cmd.ExecuteNonQuery();

conn.Close();

}

- Thêm thông tin lưu trữ

Procedure thêm thông tin lưu trữ

CREATE PROCEDURE proc\_ThemLuuTru

@MaNhanVien INT,

@MaNguyenLieu INT,

@SoLuong INT,

@NgayHetHan DATETIME2(7)

AS

BEGIN

INSERT INTO LuuTrus(MaNhanVien, MaNguyenLieu, SoLuong, NgayHetHan)

VALUES (@MaNhanVien, @MaNguyenLieu, @SoLuong, @NgayHetHan);

END

Thực thi trên C#

public static void Update\_InfoLuuTruDTO(LuuTru LuuTru)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_CapNhatThongTinLuuTru", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaLuuTru", LuuTru.MaLuuTru);

cmd.Parameters.AddWithValue("@SoLuong", LuuTru.SoLuong);

cmd.Parameters.AddWithValue("@NgayHetHan", LuuTru.NgayHetHan);

cmd.ExecuteNonQuery();

conn.Close();

}

- Xóa thông tin lưu trữ

Procedure xóa thông tin lưu trữ

CREATE PROCEDURE proc\_XoaLuuTru

@MaLuuTru INT

AS

BEGIN

DELETE FROM LuuTrus WHERE MaLuuTru = @MaLuuTru;

END

Thực thi trên C#

public static void Delete\_LuuTruDTO(int maLuuTru)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_XoaLuuTru", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaLuuTru", maLuuTru);

cmd.ExecuteNonQuery();

conn.Close();

}

**Trigger tự động đặt số lượng nguyên liệu về 0 khi hết hạn**

CREATE TRIGGER trg\_LuuTrus\_CheckExpiration

ON [dbo].[LuuTrus]

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

-- Cập nhật SoLuong = 0 cho các bản ghi đã hết hạn

UPDATE [dbo].[LuuTrus]

SET SoLuong = 0

WHERE NgayHetHan < GETDATE()

AND SoLuong > 0;

END;

GO

## **13. Chức Vụ**

- Hiển thị danh sách thông tin chức vụ

Thực thi trên C#

public static DataTable Get\_ListChucVuDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM ChucVus", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

# **CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

Ứng với ba nhóm người dùng (nhân viên bán hàng, nhân viên quản lý kho và quản lý) là các role riêng:

+ Đối với role nhân viên bán hàng:

-- Tạo login

CREATE LOGIN NhanVienPhucVuLogin WITH PASSWORD = '1';

-- Tạo user trong cơ sở dữ liệu cho các login

CREATE USER NhanVienPhucVuUser FOR LOGIN NhanVienPhucVuLogin;

-- Tạo role trong cơ sở dữ liệu để quản lý quyền

CREATE ROLE PhucVuRole;

-- Gán user vào các role tương ứng

EXEC sp\_addrolemember 'PhucVuRole', 'NhanVienPhucVuUser';

-- Phân quyền cho PhucVuRole

---- Phân quyền cho PhucVuRole

GRANT SELECT ON [dbo].[MonAns] TO PhucVuRole;

GRANT SELECT, INSERT ON [dbo].[HoaDons] TO PhucVuRole;

GRANT SELECT, INSERT ON [dbo].[ChiTietHoaDons] TO PhucVuRole;

GO

---- Phân quyền Procedure cho PhucVuRole

-- MonAns

GRANT EXECUTE ON dbo.proc\_ThemMonAn TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinMonAn TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_XoaMonAn TO PhucVuRole;

-- HoaDons

GRANT EXECUTE ON dbo.proc\_ThemHoaDon TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinHoaDon TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_XoaHoaDon TO PhucVuRole;

-- ChiTietHoaDons

GRANT EXECUTE ON dbo.proc\_ThemChiTietHoaDon TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinChiTietHoaDon TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_XoaChiTietHoaDon TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_LayCTHD\_MaHoaDon TO PhucVuRole;

GRANT EXECUTE ON dbo.proc\_LayThongTin TO PhucVuRole;

GO

---- Phân quyền View cho PhucVuRole

GRANT SELECT ON dbo.View\_HoaDon TO PhucVuRole;

GRANT SELECT ON dbo.View\_ChiTietHoaDon TO PhucVuRole;

+ Đối với role nhân viên quản lý kho:

-- Tạo login

CREATE LOGIN NhanVienQuanLyKhoLogin WITH PASSWORD = '2';

-- Tạo user trong cơ sở dữ liệu cho các login

CREATE USER NhanVienQuanLyKhoUser FOR LOGIN NhanVienQuanLyKhoLogin;

-- Tạo role trong cơ sở dữ liệu để quản lý quyền

CREATE ROLE QuanLyKhoRole;

-- Gán user vào các role tương ứng

EXEC sp\_addrolemember 'QuanLyKhoRole', 'NhanVienQuanLyKhoUser';

-- Phân quyền cho QuanLyKhoRole

---- Phân quyền cho QuanLyKhoRole

GRANT SELECT, INSERT, UPDATE ON [dbo].[NguyenLieus] TO QuanLyKhoRole;

GRANT SELECT, INSERT, UPDATE ON [dbo].[NhapKhos] TO QuanLyKhoRole;

GRANT SELECT, INSERT, UPDATE ON [dbo].[XuatKhos] TO QuanLyKhoRole;

GRANT SELECT, INSERT, UPDATE ON [dbo].[LuuTrus] TO QuanLyKhoRole;

GO

---- Phân quyền Procedure cho QuanLyKhoRole

-- NguyenLieus

GRANT EXECUTE ON dbo.proc\_ThemNguyenLieu TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinNguyenLieu TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_XoaNguyenLieu TO QuanLyKhoRole;

-- NhapKhos

GRANT EXECUTE ON dbo.proc\_ThemNhapKho TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinNhapKho TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_XoaNhapKho TO QuanLyKhoRole;

-- XuatKhos

GRANT EXECUTE ON dbo.proc\_ThemXuatKho TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinXuatKho TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_XoaXuatKho TO QuanLyKhoRole;

-- LuuTrus

GRANT EXECUTE ON dbo.proc\_ThemLuuTru TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_CapNhatThongTinLuuTru TO QuanLyKhoRole;

GRANT EXECUTE ON dbo.proc\_XoaLuuTru TO QuanLyKhoRole;

GO

---- Phân quyền View cho QuanLyKhoRole

GRANT SELECT ON dbo.View\_NguyenLieu TO QuanLyKhoRole;

GRANT SELECT ON dbo.View\_XuatKho TO QuanLyKhoRole;

GRANT SELECT ON dbo.View\_NhapKho TO QuanLyKhoRole;

GRANT SELECT ON dbo.View\_LuuTru TO QuanLyKhoRole;

+ Đối với role quản lý:

-- Tạo login

CREATE LOGIN AdminLogin WITH PASSWORD = '3';

-- Tạo user trong cơ sở dữ liệu cho các login

CREATE USER AdminUser FOR LOGIN AdminLogin;

-- Tạo role trong cơ sở dữ liệu để quản lý quyền

CREATE ROLE AdminRole;

-- Gán user vào các role tương ứng

EXEC sp\_addrolemember 'AdminRole', 'AdminUser';

-- Phân quyền cho AdminRole (toàn quyền trên tất cả bảng)

---- Phân quyền cho AdminRole (toàn quyền trên tất cả bảng)

GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::[dbo] TO AdminRole;

GO

---- Phân quyền Procedure cho AdminRole(toàn quyền trên tất cả procedure)

DECLARE @sql NVARCHAR(MAX) = '';

SELECT @sql = @sql + 'GRANT EXECUTE ON dbo.' + name + ' TO AdminRole;' + CHAR(13)

FROM sys.objects

WHERE type = 'P';

EXEC sp\_executesql @sql;

GO

---- Phân quyền View cho AdminRole(toàn quyền trên tất cả view)

DECLARE @sql NVARCHAR(MAX) = '';

SELECT @sql += 'GRANT SELECT ON [' + s.name + '].[' + v.name + '] TO AdminRole;' + CHAR(13)

FROM sys.views v

JOIN sys.schemas s ON v.schema\_id = s.schema\_id;

EXEC sp\_executesql @sql;

Thực thi trên C#

public static int GetMaChucVuDTO(TaiKhoanNhanVien taiKhoan)

{

int maChucVu = -1;

using (SqlConnection conn = Connect())

{

string query = "SELECT nv.MaChucVu FROM TaiKhoanNhanViens tknv " +

"JOIN NhanViens nv ON tknv.MaNhanVien = nv.MaNhanVien " +

"WHERE tknv.TaiKhoan = @taikhoan";

SqlCommand cmd = new SqlCommand(query, conn);

cmd.Parameters.AddWithValue("@taikhoan", taiKhoan.TaiKhoan);

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

maChucVu = reader.GetInt32(0);

}

}

return maChucVu;

}

Tạo case connection trong C#

public void UpdateConnectionStringBasedOnRole(int maChucVu)

{

switch (maChucVu)

{

case 2:

SqlConnectionData.UpdateConnectionString("NhanVienPhucVuLogin", "PhucVu@2025");

break;

case 3:

SqlConnectionData.UpdateConnectionString("NhanVienQuanLyKhoLogin", "QuanLyKho@2025");

break;

case 1:

SqlConnectionData.UpdateConnectionString("AdminLogin", "Admin@2025");

break;

default:

throw new Exception("Chức vụ không hợp lệ");

}

}

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

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

- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.

**Giao diện và chức năng của chương trình:**

## **5.1. Form đăng nhập**

A screenshot of a login screen

AI-generated content may be incorrect.

- Đăng nhập

Code C#:

public static string Check\_DangNhapDTO(TaiKhoanNhanVien taiKhoan)

{

string user = null;

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_DangNhap", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TaiKhoan", taiKhoan.TaiKhoan);

cmd.Parameters.AddWithValue("@MatKhau", taiKhoan.MatKhau);

cmd.Connection = conn;

SqlDataReader rdr = cmd.ExecuteReader();

if (rdr.HasRows)

{

if (rdr.Read())

{

user = rdr.GetInt32(0).ToString();

return user;

}

rdr.Close();

conn.Close();

}

else

{

return "taikhoan\_hoac\_matkhau\_wrong";

}

return user;

}

Code SQL:

-- DangNhap

CREATE PROCEDURE proc\_DangNhap

@TaiKhoan NVARCHAR(MAX),

@MatKhau NVARCHAR(MAX)

AS

BEGIN

SELECT \* FROM TaiKhoanNhanViens WHERE TaiKhoan = @TaiKhoan AND MatKhau = @MatKhau

END

## **5.2. Form đăng ký**

A screenshot of a computer

AI-generated content may be incorrect.

- Tạo tài khoản:

Code C#

public static void Check\_DangKyDTO(TaiKhoanNhanVien taiKhoan)

{

SqlConnection conn = Connect();

conn.Open();

try

{

SqlCommand cmd = new SqlCommand("proc\_DangKy", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNhanVien", taiKhoan.MaNhanVien);

cmd.Parameters.AddWithValue("@TaiKhoan", taiKhoan.TaiKhoan);

cmd.Parameters.AddWithValue("@MatKhau", taiKhoan.MatKhau);

cmd.ExecuteNonQuery();

}

catch (SqlException ex)

{

if (ex.Message.Contains("Nhân viên này đã có tài khoản!"))

throw new Exception("had\_user");

else if (ex.Message.Contains("Tên tài khoản đã được sử dụng!"))

throw new Exception("already\_in\_use");

else

throw;

}

finally

{

conn.Close();

}

}

Code SQL:

CREATE PROCEDURE proc\_DangKy

@MaNhanVien INT,

@TaiKhoan NVARCHAR(MAX),

@MatKhau NVARCHAR(MAX)

AS

BEGIN

SET XACT\_ABORT ON;

IF EXISTS (SELECT 1 FROM TaiKhoanNhanViens WHERE TaiKhoan = @TaiKhoan)

BEGIN

RAISERROR('Tên tài khoản đã được sử dụng!', 16, 1);

RETURN;

END

IF EXISTS (SELECT 1 FROM TaiKhoanNhanViens WHERE MaNhanVien = @MaNhanVien AND TaiKhoan IS NOT NULL)

BEGIN

RAISERROR('Nhân viên này đã có tài khoản!', 16, 1);

RETURN;

END

UPDATE TaiKhoanNhanViens

SET TaiKhoan = @TaiKhoan,

MatKhau = @MatKhau

WHERE MaNhanVien = @MaNhanVien;

END

GO

## **5.3. Form Trang Chủ**

A screenshot of a menu

AI-generated content may be incorrect.

## **5.4. Các Danh Mục Hệ Thống**

### **5.4.1. Danh Mục Thông Tin Tài Khoản**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

private void LoadThongTinNhanVien()

{

DataTable dt = taiKhoan.Get\_Info();

foreach (DataRow row in dt.Rows)

{

if ((int)row["MaNhanVien"] == Properties.Settings.Default.MaNhanVien)

{

textBox\_MaNhanVien.Text = row["MaNhanVien"].ToString();

textBox\_TenNhanVien.Text = row["TenNhanVien"].ToString();

textBox\_ChucVu.Text = row["TenChucVu"].ToString();

textBox\_DiaChi.Text = row["DiaChi"].ToString();

textBox\_SoDienThoai.Text = row["SoDienThoai"].ToString();

textBox\_Email.Text = row["Email"].ToString();

break;

}

}

}

### **5.4.2. Danh Mục Đổi Mật Khẩu**

A screenshot of a computer

AI-generated content may be incorrect.

Code C# đổi mật khẩu:

public static string DoiMatKhauDTO(string currentPass, string username, string newPass, string cfPass)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand checkCmd = new SqlCommand("proc\_LayMatKhau", conn);

checkCmd.CommandType = CommandType.StoredProcedure;

checkCmd.Parameters.AddWithValue("@TaiKhoan", username);

object result = checkCmd.ExecuteScalar();

string pass = result.ToString();

if (currentPass != pass)

{

return "wrong\_current\_pass";

}

SqlCommand cmd = new SqlCommand("proc\_DoiMatKhau", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TaiKhoan", username);

cmd.Parameters.AddWithValue("@MatKhauMoi", newPass);

cmd.ExecuteNonQuery();

conn.Close();

return "success\_change\_password";

}

Code SQL đổi mật khẩu:

CREATE PROCEDURE proc\_DoiMatKhau

@TaiKhoan NVARCHAR(MAX),

@MatKhauMoi NVARCHAR(MAX)

AS

BEGIN

UPDATE TaiKhoanNhanViens

SET MatKhau = @MatKhauMoi

WHERE TaiKhoan = @TaiKhoan

END

## **5.5. Các Danh Mục Quản Lý**

### **5.5.1. Danh Mục Danh Sách Nhân Viên**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public class DSNhanVienDAL : SqlConnectionData

{

public static DataTable Get\_ListNhanVienDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_NhanVien", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

}

- Code tạo view trên SQL:

CREATE VIEW View\_NhanVien AS

SELECT

nv.MaNhanVien,

nv.TenNhanVien,

nv.DiaChi,

nv.SoDienThoai,

nv.Email,

cv.TenChucVu

FROM NhanViens nv

LEFT JOIN ChucVus cv ON nv.MaChucVu = cv.MaChucVu;

### **5.5.2. Danh Mục Danh Sách Hóa Đơn**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListHoaDonDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_HoaDon", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo view danh mục:

CREATE VIEW View\_HoaDon AS

SELECT

hd.MaHoaDon,

nv.TenNhanVien,

kh.TenKhachHang,

hd.NgayLap,

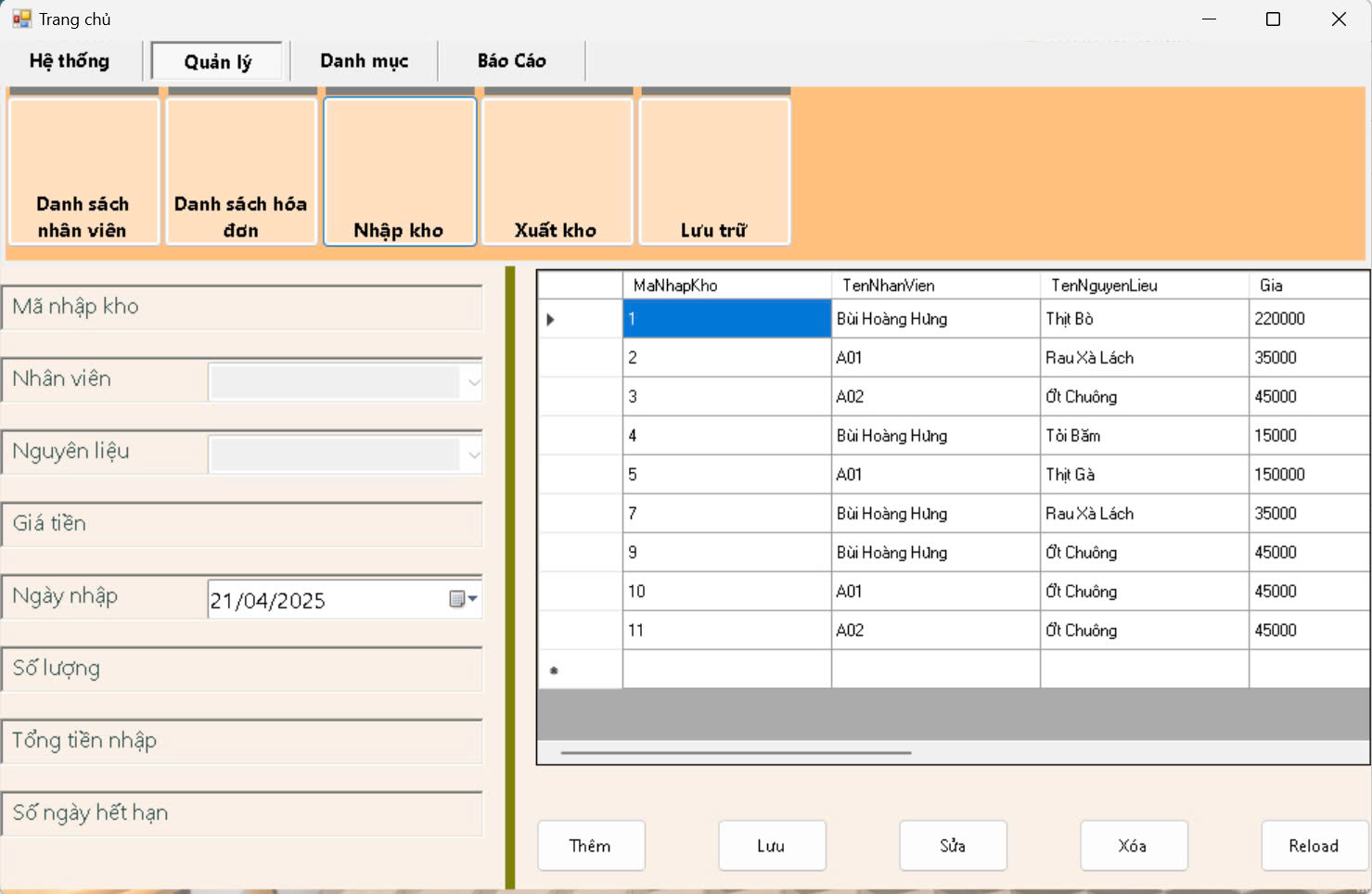
hd.TongTien

FROM HoaDons hd

INNER JOIN NhanViens nv ON hd.MaNhanVien = nv.MaNhanVien

INNER JOIN KhachHangs kh ON hd.MaKhachHang = kh.MaKhachHang

### **5.5.3. Danh Mục Nhập Kho**



- Code C# gọi danh mục:

public static DataTable Get\_ListNhapKhoDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_NhapKho", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo view danh mục:

CREATE VIEW View\_NhapKho AS

SELECT

nk.MaNhapKho,

nv.TenNhanVien,

nl.TenNguyenLieu,

nl.Gia,

nk.NgayNhap,

nk.SoLuong,

nk.TongTien,

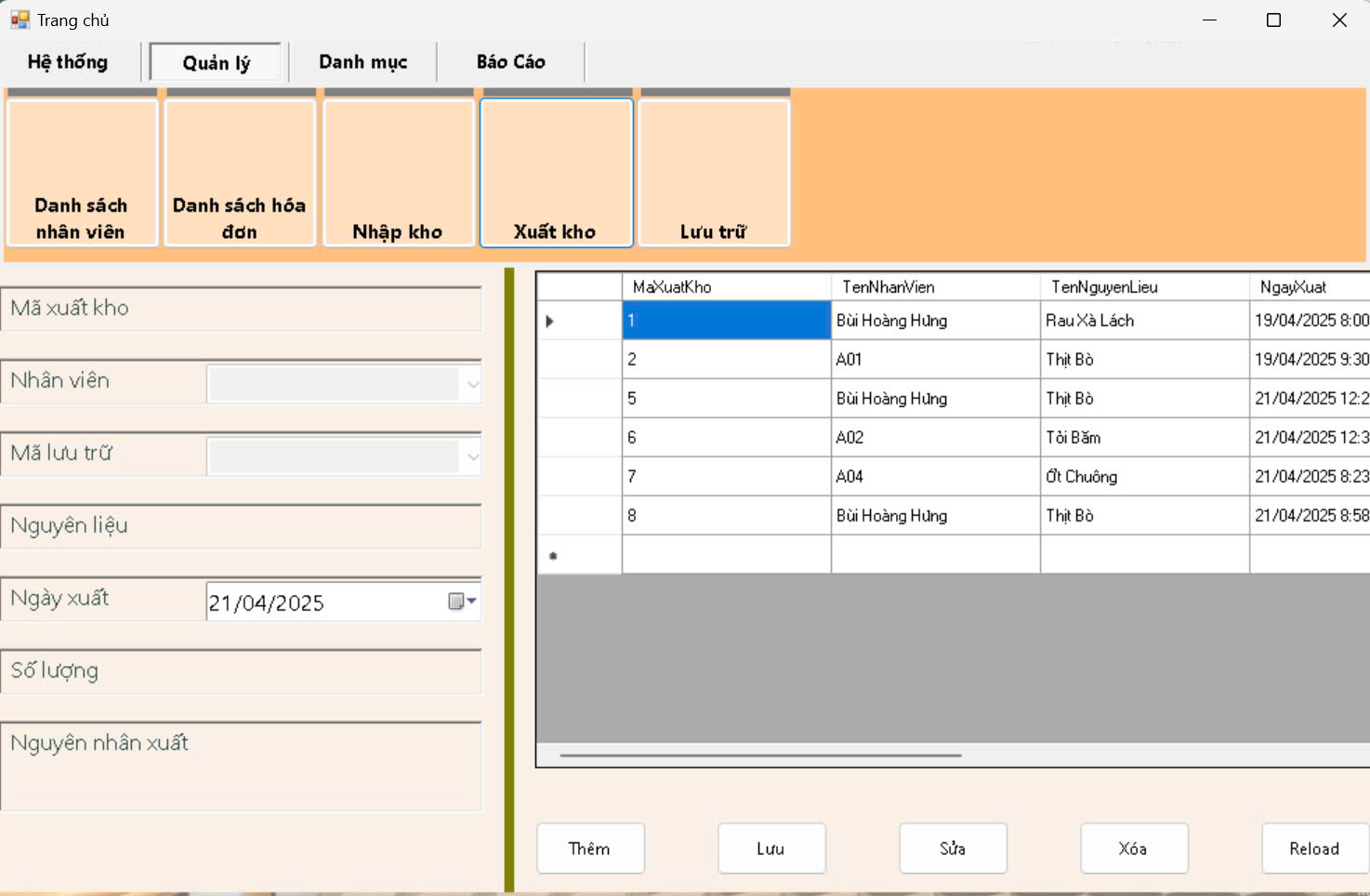
nk.SoNgayHetHan

FROM NhapKhos nk

INNER JOIN NhanViens nv ON nk.MaNhanVien = nv.MaNhanVien

INNER JOIN NguyenLieus nl ON nl.MaNguyenLieu = nk.MaNguyenLieu

### **5.5.4. Danh Mục Xuất Kho**



- Code C# gọi danh mục:

public static DataTable Get\_ListXuatKhoDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_XuatKho", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo view danh mục

CREATE VIEW View\_XuatKho AS

SELECT

nk.MaXuatKho,

nv.TenNhanVien,

nl.TenNguyenLieu,

nk.NgayXuat,

nk.SoLuong,

nk.NguyenNhanXuatKho,

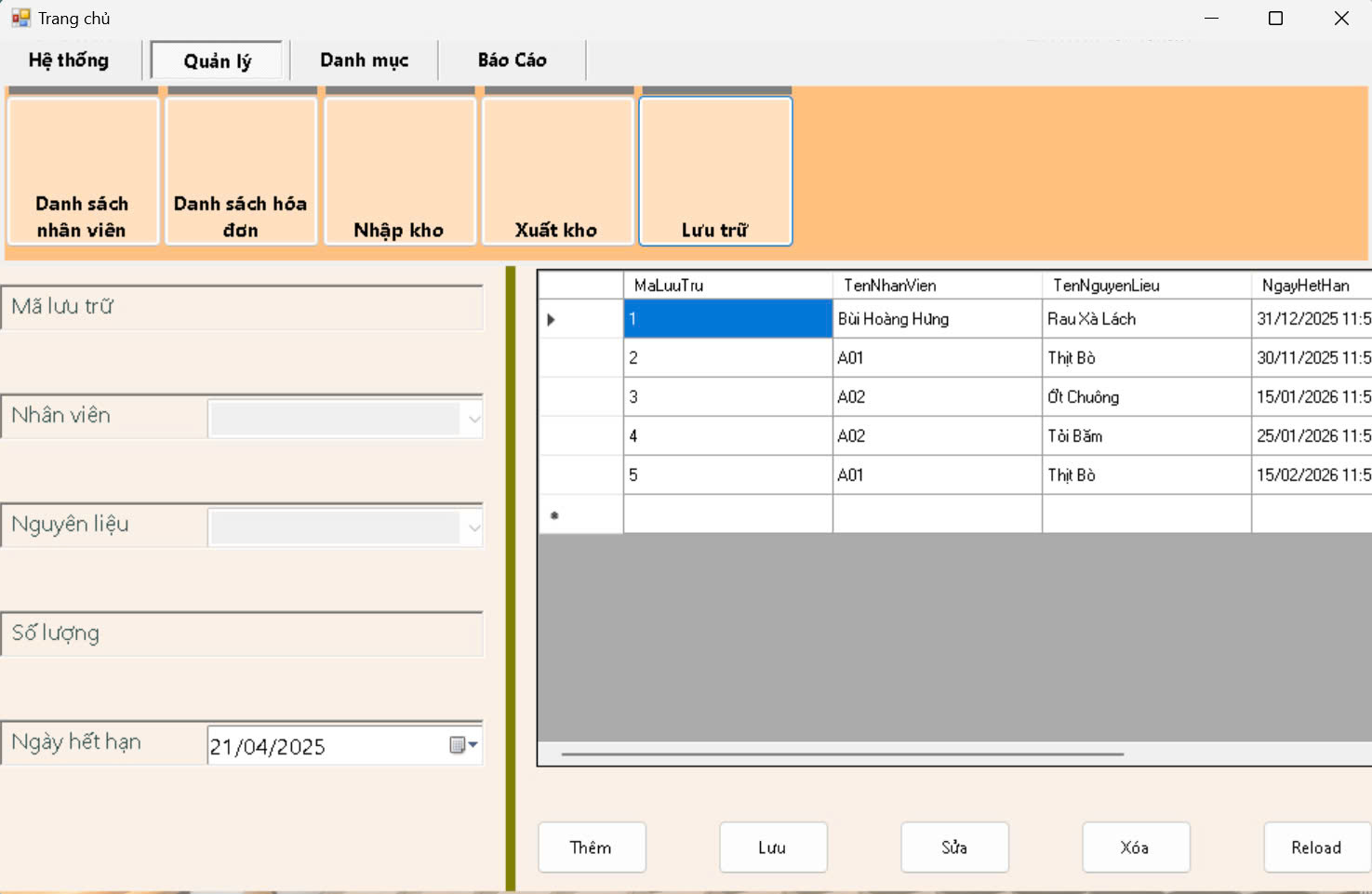
nk.MaLuuTru

FROM XuatKhos nk

INNER JOIN NhanViens nv ON nk.MaNhanVien = nv.MaNhanVien

INNER JOIN NguyenLieus nl ON nl.MaNguyenLieu = nk.MaNguyenLieu

### **5.5.5. Danh Mục Lưu Trữ**



- Code C# gọi danh mục lưu trữ

public static DataTable Get\_ListLuuTruDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_LuuTru", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo view danh mục

CREATE VIEW View\_LuuTru AS

SELECT

lt.MaLuuTru,

lt.MaNguyenLieu,

nv.TenNhanVien,

nl.TenNguyenLieu,

lt.NgayHetHan,

lt.SoLuong

FROM LuuTrus lt

INNER JOIN NhanViens nv ON lt.MaNhanVien = nv.MaNhanVien

INNER JOIN NguyenLieus nl ON nl.MaNguyenLieu = lt.MaNguyenLieu

## **5.6. Các Danh Mục**

### **5.6.1. Danh Mục Nguyên Liệu**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListNguyenLieuDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_NguyenLieu", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo view danh mục

CREATE VIEW View\_NguyenLieu AS

SELECT

nl.MaNguyenLieu,

nl.TenNguyenLieu,

nl.DonViTinh,

nl.Gia,

ncc.TenNhaCungCap

FROM NguyenLieus nl

LEFT JOIN NhaCungCaps ncc ON nl.MaNhaCungCap = ncc.MaNhaCungCap;

### **5.6.2. Danh Mục Nhà Cung Cấp**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListNhaCungCapDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM NhaCungCaps", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

### **5.6.3. Danh Mục Khách Hàng**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListKhachHangDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM KhachHangs", conn);

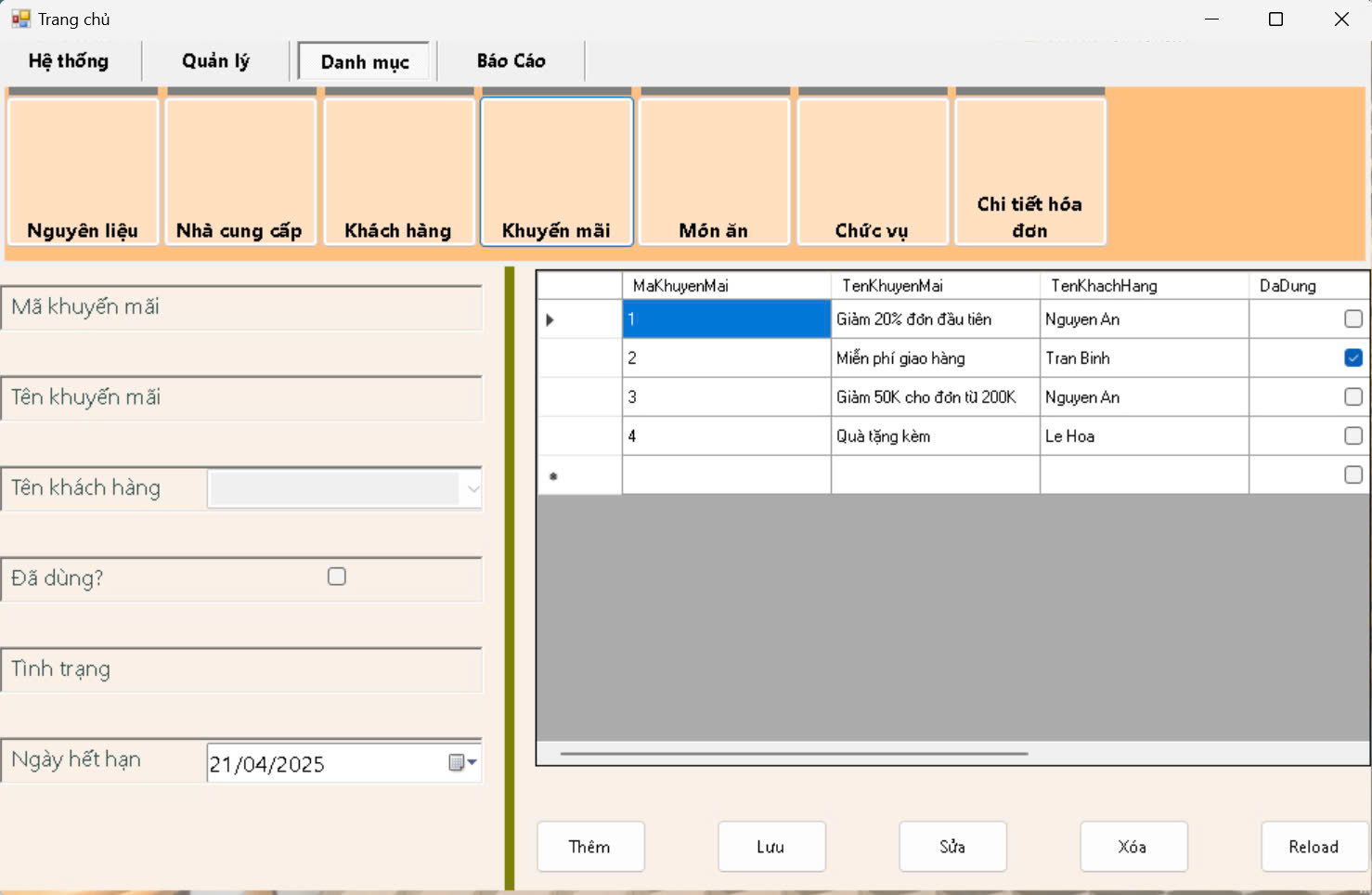
DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

### **5.6.4. Danh Mục Khuyến Mãi**



- Code C# gọi danh mục:

public static DataTable Get\_ListKhuyenMaiDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM View\_KhuyenMai", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo view danh mục

CREATE VIEW View\_KhuyenMai AS

SELECT

km.MaKhuyenMai,

km.TenKhuyenMai,

kh.MaKhachHang,

kh.TenKhachHang,

km.DaDung,

km.NgayHetHan,

-- Trạng thái khuyến mãi hiển thị dễ hiểu

CASE

WHEN km.DaDung = 1 THEN N'Đã dùng'

WHEN km.NgayHetHan < GETDATE() THEN N'Hết hạn'

ELSE N'Còn hạn'

END AS TrangThai

FROM KhuyenMais km

JOIN KhachHangs kh ON km.MaKhachHang = kh.MaKhachHang;

### **5.6.5. Danh Mục Món Ăn**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListMonAnDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM MonAns", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

### **5.6.6. Danh Mục Chức Vụ**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListChucVuDTO()

{

SqlConnection conn = Connect();

conn.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM ChucVus", conn);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

### **5.6.7. Danh Mục Chi Tiết Hóa Đơn**

A screenshot of a computer

AI-generated content may be incorrect.

- Code C# gọi danh mục:

public static DataTable Get\_ListChiTietHoaDonDTO(int maHD)

{

SqlConnection conn = Connect();

conn.Open();

SqlCommand cmd = new SqlCommand("proc\_LayCTHD\_MaHoaDon", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaHoaDon", maHD);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

- Code SQL tạo procedure:

CREATE PROCEDURE proc\_LayCTHD\_MaHoaDon

@MaHoaDon INT

AS

BEGIN

SELECT

cthd.MaChiTietHoaDon,

cthd.MaHoaDon,

ma.TenMonAn,

cthd.SoLuong,

ma.Gia,

cthd.ThanhTien

FROM ChiTietHoaDons cthd

INNER JOIN MonAns ma ON cthd.MaMonAn = ma.MaMonAn

WHERE cthd.MaHoaDon = @MaHoaDon

END