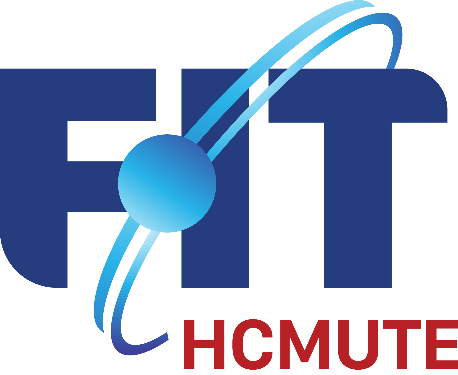
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

****

**ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL**

**TÊN ĐỀ TÀI: QUẢN LÝ QUÁN CÀ PHÊ**

**GVHD:** *TS. Nguyễn Thành Sơn*

**Lớp HP:** *DBMS330284\_07*

**Nhóm thực hiện:** *Nhóm 09*

**Học kỳ:** *1*

**Năm học:** *2024 – 2025*

**Thành phố Hồ Chí Minh, tháng 11, năm 2024**

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2024-2025**

**Nhóm 09**

***Đề tài: Quản lý quán cà phê***

|  |  |
| --- | --- |
| **MSSV** | **Họ và tên sinh viên** |
| 22110352 | Nguyễn Hoàng Anh Khoa |
| 22110444 | Võ Văn Trí |
| 22110337 | Nguyễn Lý Hùng |
| 22110318 | Trương Hồng Hạnh |

**NHẬN XÉT CỦA GV**

|  |
| --- |
|  |
|  |
|  |
|  |

Tp. Hồ Chí Minh, ngày …., tháng 11, năm 2024

Giảng viên chấm điểm

**MỤC LỤC**

[**CHƯƠNG 1: ĐẶC TẢ BÀI TOÁN** 1](#_Toc180759764)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 4](#_Toc180759765)

[**1. Thiết kế cơ sở dữ liệu mức quan niệm** 4](#_Toc180759766)

[**2. Thiết kế cơ sở dữ liệu mức logic** 4](#_Toc180759767)

[**3. Các ràng buộc cần có** 5](#_Toc180759768)

[**4.** **Cài đặt các CSDL và các ràng buộc.** 8](#_Toc180759769)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG** 13](#_Toc180759770)

[**1.** **Kết nối cơ sở dữ liệu.** 13](#_Toc180759771)

[**2.** **Các hàm thực thi thủ tục, hàm chung.** 13](#_Toc180759772)

[**2.1.** **Hàm load view.** 13](#_Toc180759773)

[**2.2.** **Hàm scalar.** 14](#_Toc180759774)

[**2.3.** **Hàm function inline và multi statement** 15](#_Toc180759775)

[**2.4.** **Hàm thực thi thủ tục có tham số output** 15](#_Toc180759776)

[**2.5.** **Hàm thực thi thủ tục có tham số hoặc không có tham số** 16](#_Toc180759777)

[**2.6.** **Hàm thực thi thủ tục có reader** 17](#_Toc180759778)

[**3.** **Xem thông tin danh mục.** 18](#_Toc180759779)

[**3.1.** **Danh mục nhân viên.** 18](#_Toc180759780)

[**3.2.** **Danh mục hóa đơn.** 18](#_Toc180759781)

[**3.3.** **Danh mục sản phẩm.** 18](#_Toc180759782)

[**3.4.** **Danh mục đơn hàng** 19](#_Toc180759783)

[**3.5.** **Danh mục phân ca** 19](#_Toc180759784)

[**3.6.** **Danh mục thống kê** 21](#_Toc180759785)

[**4.** **Quản lý nhân viên** 22](#_Toc180759786)

[**4.1.** **Trigger tạo id tự động khi thêm nhân viên** 22](#_Toc180759787)

[**4.2.** **Thêm nhân viên mới.** 23](#_Toc180759788)

[**4.3.** **Xóa nhân viên.** 24](#_Toc180759789)

[**4.4.** **Sửa nhân viên.** 26](#_Toc180759790)

[**4.5.** **Tìm kiếm nhân viên.** 28](#_Toc180759791)

[**5.** **Quản lý khách hàng** 29](#_Toc180759792)

[**5.1.** **Thêm khách hàng** 29](#_Toc180759793)

[**5.2.** **Xóa khách hàng** 30](#_Toc180759794)

[**5.3.** **Sửa thông tin khách hàng** 30](#_Toc180759795)

[**5.4.** **Tìm kiếm khách hàng theo số điện thoại** 31](#_Toc180759796)

[**5.5.** **Trigger tạo mã khách hàng** 31](#_Toc180759797)

[**6.** **Quản lý hóa đơn** 32](#_Toc180759798)

[**6.1.** **Thêm đơn hàng** 32](#_Toc180759799)

[**6.2.** **Xóa đơn chưa thanh toán** 33](#_Toc180759800)

[**6.3.** **Cập nhật giá trị đơn** 33](#_Toc180759801)

[**6.4.** **Xuất hóa đơn** 34](#_Toc180759802)

[**6.5.** **Trigger tự động tạo mã đơn hàng** 35](#_Toc180759803)

[**7.** **Quản lý sản phẩm** 36](#_Toc180759804)

[**7.1.** **Thêm sản phẩm** 36](#_Toc180759805)

[**7.2.** **Xóa sản phẩm** 36](#_Toc180759806)

[**7.3.** **Sửa sản phẩm** 37](#_Toc180759807)

[**7.4.** **Tìm kiếm sản phẩm** 37](#_Toc180759808)

[**7.5.** **Trigger tự động tạo mã sản phẩm** 38](#_Toc180759809)

[**7.6.** **Trigger kiểm tra số lượng tồn kho trước khi pha chế** 39](#_Toc180759810)

[**8.** **Quản lý kho nguyên liệu** 39](#_Toc180759811)

[**8.1.** **Thêm nguyên liệu** 39](#_Toc180759812)

[**8.2.** **Xóa nguyên liệu** 40](#_Toc180759813)

[**8.3.** **Sửa nguyên liệu** 40](#_Toc180759814)

[**8.4.** **Tìm kiếm nguyên liệu** 41](#_Toc180759815)

[**8.5.** **Trigger tự động tạo mã nguyên liệu** 41](#_Toc180759816)

[**9.** **Quản lý phân ca làm việc** 42](#_Toc180759817)

[**9.1.** **Thêm ca làm việc** 42](#_Toc180759818)

[**9.2.** **Tìm người thực hiện ca** 43](#_Toc180759819)

[**9.3.** **Thêm thông tin thực hiện ca làm việc** 44](#_Toc180759820)

[**10.** **Kiểm tra đăng nhập nhân viên** 44](#_Toc180759821)

[**11.** **Lấy lại mật khẩu nhân viên** 45](#_Toc180759822)

[**12. Trang chủ** 46](#_Toc180759823)

[**12.1. Tìm kiếm theo tên** 46](#_Toc180759824)

[**12.2. Thêm khách hàng mới** 46](#_Toc180759825)

[**12.3. Tạo đơn hàng mới** 47](#_Toc180759826)

[**12.4. Xóa khách hàng chưa nhập tên và đơn hàng chưa thanh toán** 47](#_Toc180759827)

[**12.5. Đặt hàng** 48](#_Toc180759828)

[**12.6. Thanh toán** 49](#_Toc180759829)

[**12.7. Tổng tiền chi tiết hóa đơn** 49](#_Toc180759830)

[**12.8 Trigger thêm chi tiết hóa đơn** 50](#_Toc180759831)

[**13. Đơn hàng** 51](#_Toc180759832)

[**13.1. Xác nhận thanh toán** 51](#_Toc180759833)

[**13.2. Load thông tin khách hàng** 51](#_Toc180759834)

[**13.3. Sửa khách hàng** 52](#_Toc180759835)

[**14. Thống kê** 53](#_Toc180759836)

[**14.1. Xem tổng doanh thu** 53](#_Toc180759837)

[**14.2. Xem tổng số lượng sản phẩm** 53](#_Toc180759838)

[**14.3. Xem tổng số lượng khách hàng** 54](#_Toc180759839)

[**15. Đăng ký** 54](#_Toc180759840)

[**16. Quản lý loại sản phẩm** 55](#_Toc180759841)

[**16.1 Thêm loại sản phẩm** 55](#_Toc180759842)

[**16.2. Xóa loại sản phẩm** 56](#_Toc180759843)

[**16.3. Sửa loại sản phẩm** 56](#_Toc180759844)

[**16.4. Tìm kiếm theo tên loại sản phẩm** 57](#_Toc180759845)

# **CHƯƠNG 1: ĐẶC TẢ BÀI TOÁN**

Thị trường kinh doanh quán cà phê ngày càng phát triển và cạnh tranh nên việc áp dụng công nghệ vào quản lý là vô cùng cần thiết, giúp quán vận hành hiệu quả, tiết kiệm nhiều thời gian và chi phí.

Một hệ thống quản lý quán cà phê cần lưu trữ những dữ liệu sau đây:

Quán cà phê có một quản lý và nhiều nhân viên, mỗi nhân viên làm một công việc, mỗi công việc có một hoặc nhiều nhân viên cùng làm. Các công việc được phân biệt với nhau bằng mã công việc. Công việc bao gồm các thông tin như mã công việc, tên công việc.

Một nhân viên làm một hoặc nhiều ca, một ca có thể có nhiều nhân viên cùng làm. Các ca làm việc được phân biệt với nhau bằng mã ca làm việc. Ca làm việc gồm có các thông tin như: mã ca làm việc, tên ca làm việc, ngày, giờ bắt đầu, giờ kết thúc.

Các nhân viên được phân biệt với nhau bằng mã nhân viên. Nhân viên gồm có các thông tin như: mã nhân viên, họ tên nhân viên, năm sinh, giới tính, địa chỉ, email, số điện thoại, tên đăng nhập, mật khẩu. Ngoài ra, nhân viên được phân thành hai loại là toàn thời gian, bán thời gian. Nhân viên toàn thời gian sẽ nhận lương cố định, còn nhân viên bán thời gian sẽ nhận lương theo giờ. Mỗi nhân viên sẽ chọn ca làm việc để thực hiện một công việc. Ngoài ra, hàng tháng nhân viên có thể được thưởng nếu trong tháng đó nhân viên bán được nhiều hơn năm đơn hàng và số tiền thưởng sẽ được tính bằng 1% tổng giá trị đơn hàng mà nhân viên đó bán được.

Các khách hàng phân biệt với nhau bằng mã khách hàng. Với mỗi lần mua hàng, khách hàng sẽ nhận được điểm tích lũy tương ứng với 1% giá trị tổng hóa đơn. Số điểm tích lũy có thể được sử dụng hoặc cộng dồn ở những lần thanh toán kế tiếp. Khách hàng gồm có các thông tin như: mã khách hàng, họ tên khách hàng, số điện thoại, số điểm tích lũy.

Các sản phẩm phân biệt với nhau bằng mã sản phẩm. Mỗi sản phẩm bao gồm các thông tin như: mã sản phẩm, tên sản phẩm, giá trị sản phẩm, ảnh sản phẩm, mã loại sản phẩm.

Có nhiều loại sản phẩm, các loại sản phẩm phân biệt với nhau bằng mã loại sản phẩm. Một loại sản phẩm có thể có nhiều sản phẩm, một sản phẩm chỉ thuộc một loại sản phẩm. Loại sản phẩm có các thông tin như mã loại sản phẩm, tên loại sản phẩm.

Một sản phẩm được làm ra từ nhiều nguyên liệu, một nguyên liệu có thể được dùng cho nhiều sản phẩm. Các nguyên liệu được phân biệt với nhau bằng mã nguyên liệu. Nguyên liệu có các thông tin như mã nguyên liệu, tên nguyên liệu, số lượng tồn kho.

Mỗi đơn hàng sẽ được một nhân viên thanh toán cho một khách hàng cụ thể. Đơn hàng có một hoặc nhiều sản phẩm và một sản phẩm có thể thuộc nhiều đơn hàng. Các đơn hàng phân biệt với nhau qua mã đơn hàng, mỗi đơn hàng gồm có các thông tin như: mã đơn hàng, giá trị đơn hàng, ngày mua và trạng thái.

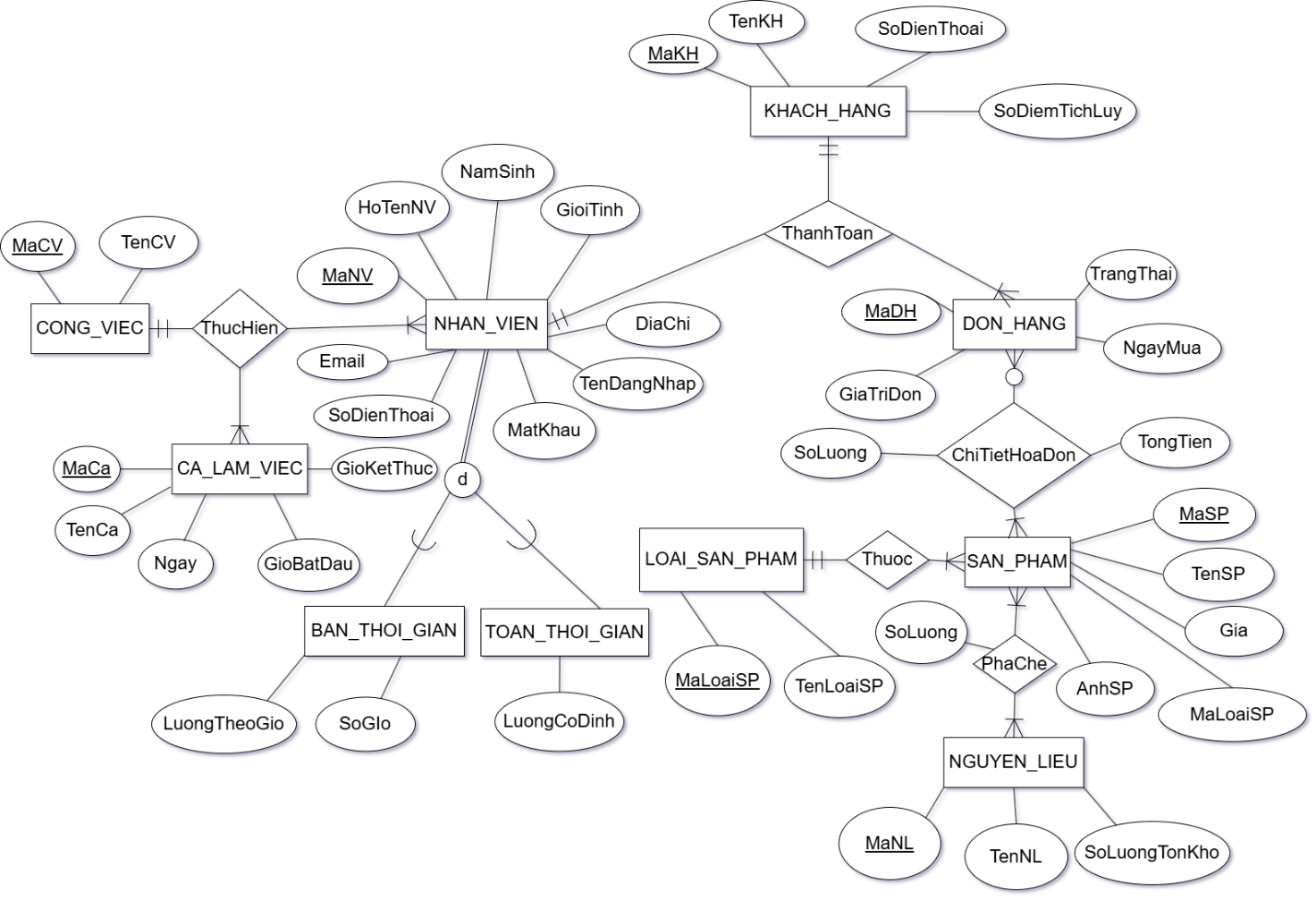
Để nâng cao hiệu quả kinh doanh, hệ thống quản lý quán cà phê được tích hợp nhiều chức năng bao gồm: chức năng bán hàng, quản lý nhân viên, quản lý khách hàng, quản lý sản phẩm, quản lý kho hàng, quản lý giao dịch, quản lý doanh thu và quản lý ca làm việc. Đối với quản lý sẽ được sử dụng toàn bộ chức năng của hệ thống, còn nhân viên chỉ được các chức năng như: chức năng bán hàng, quản lý khách hàng, quản lý sản phẩm, quản lý kho hàng, quản lý giao dịch, quản lý ca làm việc. Trong đó, chức năng bán hàng cho phép nhân viên tạo đơn hàng; thêm, xóa, chỉnh sửa món, xuất hóa đơn thanh toán. Chức năng quản lý nhân viên cho phép quản lý có thể thêm, xóa, chỉnh sửa thông tin nhân viên và tìm kiếm nhân viên. Chức năng quản lý khách hàng cho phép nhân viên thêm khách hàng, xóa khách hàng và sửa khách hàng. Chức năng quản lý sản phẩm cho phép quản lý có thể thêm, xóa, chỉnh sửa các món, điều chỉnh giá cả từng món, tìm kiếm món ăn trong menu, xem sách danh nguyên liệu và thêm nguyên liệu để pha chế. Chức năng quản lý kho hàng cho phép quản lý có thể thêm, xóa, chỉnh sửa nguyên liệu trong kho. Chức năng quản lý giao dịch cho phép quản lý có thể xem lại các hóa đơn của khách hàng và sửa thông tin của khách hàng. Chức năng quản lý doanh thu cho phép quản lý thống kê tổng doanh thu, số sản phẩm bán ra và tổng khách hàng của cửa hàng. Chức năng quản lý ca làm việc cho phép quản lý có thể thêm, xóa, chỉnh sửa ca làm việc của nhân viên.

Về giao diện, hệ thống quản lý quán cà phê gồm có: cửa sổ đăng nhập, màn hình chính. Cửa sổ đăng nhập bao gồm hai hộp văn bản để nhập tài khoản và mật khẩu, nút đăng nhập, nút đăng kí, nhãn liên kết quên mật khẩu. Khi nhấn vào nút đăng kí sẽ chuyển sang cửa sổ đăng kí, trong cửa số đăng kí có các hộp văn bản điền thông tin nhân viên và nút đăng kí. Khi nhấn vào nhãn liên kết quên mật khẩu sẽ chuyển sang cửa sổ quên mật khẩu, trong đó sẽ có hai hộp văn bản để nhập tên đăng nhập, số điện thoại và một nút lấy mật khẩu. Màn hình chính bao gồm các danh mục: trang chủ, kho hàng, sản phẩm, khách hàng, nhân viên, hóa đơn, thống kê, phân ca. Danh mục trang chủ hiển thị menu dạng lưới với hình ảnh, tên sản phẩm đi kèm giá, thanh tìm kiếm, hiển thị hóa đơn, nút tạo hóa đơn và nút xuất hóa đơn. Khi ấn vào nút, một cửa sổ nhỏ sẽ hiện ra để xác nhận hóa đơn. Danh mục kho hàng hiển thị danh sách nguyên liệu, có thanh tìm kiếm, nút thêm, xóa, sửa nguyên liệu. Danh mục sản phẩm hiển thị danh sách sản phẩm của quán, có thanh tìm kiếm, có nút thêm sản phẩm mới, nút chỉnh sửa, xóa sản phẩm, ngoài ra còn hiển thị danh sách các nguyên liệu dùng để chế biến và nút thêm nguyên liệu. Danh mục nhân viên hiển thị danh sách thông tin nhân viên, có các hộp văn bản nhập thông tin nhân viên, có thanh tìm kiếm, nút thêm, chỉnh sửa thông tin và xóa nhân viên. Danh mục khách hàng hiển thị danh sách khách hàng, có các hộp văn bản nhập thông tin khách hàng, thanh tìm kiếm, nút chỉnh sửa thông tin khách hàng, nút thêm khách hàng mới và nút xóa khách hàng. Danh mục hóa đơn hiển thị danh sách các hóa đơn mua hàng tại quán, có thanh tìm kiếm theo số điện thoại khách hàng. Danh mục thống kê hiển thị tổng doanh thu, tổng sản phẩm bán ra và tổng khách hàng đã mua ở quán. Danh mục phân ca hiển thị bảng phân công ca làm của nhân viên theo tuần, có các hộp văn bản nhập thông tin ca làm việc, có nút thêm, xóa và chỉnh sửa ca làm việc.

Hệ thống quản lý quán cà phê với các chức năng đa dạng và giao diện dễ sử dụng như trên là một ứng dụng thực tiễn, giúp quản lý hoạt động của quán thuận tiện và dễ dàng, tối ưu hóa quy trình trong bối cảnh công nghệ phát triển mạnh mẽ hiện nay.

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

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



## **2. Thiết kế cơ sở dữ liệu mức logic**

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

* CongViec(MaCV, TenCV)
* NhanVien(MaNV, HoTenNV, NamSinh, GioiTinh, DiaChi, Email, SDT, TenDangNhap, MatKhau)
* NhanVienToanThoiGian(MaNV, LuongCoDinh)
* NhanVienBanThoiGian(MaNV, LuongTheoGio, SoGio)
* CaLamViec(MaCa, TenCa, Ngay, GioBatDau, GioKetThuc)
* ThucHien(MaNV, MaCV, MaCa)
* DonHang(MaDH, NgayMua, GiaTriDon, TrangThai)
* KhachHang(MaKH, TenKH, SoDienThoai, SoDiemTichLuy)
* SanPham(MaSP, TenSP, Gia, AnhSP,MaLoaiSP)
* LoaiSanPham(MaLoaiSP, TenLoaiSP)
* NguyenLieu(MaNL, TenNL, SoLuongTonKho)
* PhaChe(MaSP, MaNL, SoLuong)
* ChiTietHoaDon(MaDH, MaSP, SoLuong, TongTien)
* ThanhToan (MaDH, MaNV, MaKH)

## **3. Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | CongViec | Kiểu dữ liệu:   * MaCV int * TenCV nvarchar(50)   Ràng buộc:  Khóa chính MaCV,  NOT NULL TenCV |
| 2 | NhanVien | Kiểu dữ liệu:   * MaNV nvarchar(50) * HoTenNV nvarchar(50) * SoDienThoai nvarchar(50) * NamSinh int * GioiTinh nvarchar(50) * DiaChi nvarchar(200) * Email nvarchar(50) * TenDangNhap nvarchar(50) * MatKhau nvarchar(50)   Ràng buộc:  Khóa chính MaNV,  NOT NULL HoTenNV,  NOT NULL SoDienThoai,  NOT NULL NamSinh,  NOT NULL GioiTinh,  NOT NULL DiaChi,  UNIQUE NOT NULL TenDangNhap,  UNIQUE SoDienThoai,  NOT NULL MatKhau,  CHECK NamSinh (>=16 tuổi),  CHECK SoDienThoai (10 ký tự) |
| 2 | NhanVienToanThoiGian | Kiểu dữ liệu:   * LuongCoDinh float   Ràng buộc:  Khóa chính MaNV,  Khóa ngoại MaNV đến bảng NhanVien,  CHECK LuongCoDinh (>=0) |
| 3 | NhanVienBanThoiGian | Kiểu dữ liệu:   * LuongTheoGio float * SoGio int   Ràng buộc:  Khóa chính MaNV,  Khóa ngoại MaNV đến bảng NhanVien,  CHECK SoGio (>=0),  CHECK LuongTheoGio (>=0) |
| 4 | CaLamViec | Kiểu dữ liệu:   * MaCa nvarchar(50) * TenCa nvarchar(50) * Ngay date * GioBatDau time(0) * GioKetThuc time(0)   Ràng buộc:  Khóa chính MaCa,  NOT NULL TenCa |
| 5 | ThucHien | Kiểu dữ liệu:   * MaNV nvarchar(50) * MaCV nvarchar(50) * MaCa nvarchar(50)   Ràng buộc:  Khóa chính (MaNV, MaCV, MaCa),  Khóa ngoại MaNV đến bảng NhanVienToanThoiGian, NhanVienBanThoiGian,  Khóa ngoại MaCV đến bảng bảng CongViec,  Khóa ngoại MaCa đến bảng CaLamViec |
| 6 | DonHang | Kiểu dữ liệu:   * MaDH nvarchar(50) * NgayMua date * GiaTriDon float   Ràng buộc:  Khóa chính MaDH,  Khóa ngoại MaKH đến bảng KhachHang,  Khóa ngoại MaNV đến bảng NhanVienToanThoiGian, NhanVienBanThoiGian,  CHECK NgayMua (>0),  CHECK GiaTriDH (>0) |
| 7 | KhachHang | Kiểu dữ liệu:   * MaKH nvarchar(50) * TenKH nvarchar(50) * SoDienThoai nvarchar(50) * SoDiemTichLuy float   Ràng buộc:  Khóa chính MaKH,  NOT NULL TenKH,  CHECK SoDienThoai (=10),  CHECK SoDiemTichLuy (>=0) |
| 8 | SanPham | Kiểu dữ liệu:   * MaSP nvarchar(50) * TenSP nvarchar(50) * Gia float * AnhSP nvarchar(200) * MaLoaiSP nvarchar(50)   Ràng buộc:  Khóa chính MaSP,  Khóa ngoại MaLoaiSP đến bảng LoaiSanPham,  NOT NULL TenSP,  NOT NULL AnhSP,  CHECK Gia (>0) |
| 9 | LoaiSanPham | Kiểu dữ liệu:   * MaLoaiSP nvarchar(50) * TenLoaiSP nvarchar(50)   Ràng buộc:  Khóa chính MaLoaiSP,  NOT NULL TenLoaiSP |
| 10 | NguyenLieu | Kiểu dữ liệu:   * MaNL nvarchar(50) * TenNL nvarchar(50) * SoLuongTonKho int   Ràng buộc:  Khóa chính MaNL,  NOT NULL TenNL,  CHECK SoLuongTonKho (>=0) |
| 11 | PhaChe | Kiểu dữ liệu:   * MaSP nvarchar(50) * MaNL nvarchar(50) * SoLuong int   Ràng buộc:  Khóa chính (MaSP, MaNL),  CHECK SoLuong (>=0)  Khóa ngoại MaSP đến bảng SanPham,  Khóa ngoại MaNL đến bảng NguyenLieu |
| 12 | ChiTietHoaDon | Kiểu dữ liệu:   * MaDH nvarchar(50) * MaSP nvarchar(50) * SoLuong int * TongTien float   Ràng buộc:  Khóa chính (MaDH, MaSP),  CHECK SoLuong (>0)  CHECK TongTien (>0)  Khóa ngoại MaDH đến bảng DonHang,  Khóa ngoại MaSP đến bảng SanPham |
| 13 | ThanhToan | Kiểu dữ liệu:   * MaDH nvarchar(50) * MaNV nvarchar(50) * MaKH nvarchar(50)   Ràng buộc:  Khóa chính (MaDH, MaNV, MaKH), |

## **4.** **Cài đặt các CSDL và các ràng buộc.**

***Bảng công việc***

|  |
| --- |
| CREATE TABLE CongViec(  MaCV nvarchar(50) CONSTRAINT PK\_CongViec PRIMARY KEY,  TenCV nvarchar(50) NOT NULL  ) |

***Bảng nhân viên***

|  |
| --- |
| CREATE TABLE NhanVien(  MaNV nvarchar(50) CONSTRAINT PK\_NhanVien PRIMARY KEY,  HoTenNV nvarchar(50) NOT NULL,  SoDienThoai nvarchar(50) NOT NULL UNIQUE CHECK(LEN(SoDienThoai)=10),  NamSinh int NOT NULL CHECK(YEAR(GETDATE())-NamSinh>=16),  GioiTinh nvarchar(50) NOT NULL,  DiaChi nvarchar(200) NOT NULL,  Email nvarchar(50) NOT NULL,  TenDangNhap nvarchar(50) UNIQUE NOT NULL,  MatKhau nvarchar(50) NOT NULL  ) |

***Bảng nhân viên toàn thời gian***

|  |
| --- |
| CREATE TABLE NhanVienToanThoiGian(  MaNV nvarchar(50) CONSTRAINT FK\_NhanVien\_ToanThoiGian FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),  LuongCoDinh float NOT NULL CHECK(LuongCoDinh>=0),  CONSTRAINT PK\_NhanVienToanThoiGian PRIMARY KEY(MaNV)  ) |

***Bảng nhân viên bán thời gian***

|  |
| --- |
| CREATE TABLE NhanVienBanThoiGian(  MaNV nvarchar(50) CONSTRAINT FK\_NhanVien\_BanThoiGian FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),  LuongTheoGio float NOT NULL CHECK(LuongTheoGio>=0),  SoGio int NOT NULL CHECK(SoGio>=0),  CONSTRAINT PK\_NhanVienBanThoiGian PRIMARY KEY(MaNV)  ) |

***Bảng ca làm việc***

|  |
| --- |
| CREATE TABLE CaLamViec(  MaCa nvarchar(50) CONSTRAINT PK\_CaLamViec PRIMARY KEY,  TenCa nvarchar(50) NOT NULL,  Ngay date ,  GioBatDau time(0),  GioKetThuc time(0)  ) |

***Bảng thực hiện***

|  |
| --- |
| CREATE TABLE ThucHien (  MaNV nvarchar(50) CONSTRAINT FK\_ThucHien\_NhanVien FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),  MaCV nvarchar(50) CONSTRAINT FK\_ThucHien\_CongViec FOREIGN KEY (MaCV) REFERENCES CongViec(MaCV),  MaCa nvarchar(50) CONSTRAINT FK\_ThucHien\_CaLamViec FOREIGN KEY (MaCa) REFERENCES CaLamViec(MaCa),  CONSTRAINT PK\_ThucHien PRIMARY KEY (MaNV,MaCV, MaCa)  ) |

***Bảng đơn hàng***

|  |
| --- |
| CREATE TABLE DonHang (  MaDH nvarchar(50) CONSTRAINT PK\_DonHang PRIMARY KEY,  NgayMua date CHECK (DATEDIFF(day, NgayMua, GETDATE()) >= 0),  GiaTriDon float CHECK (GiaTriDon > 0),  TrangThai nvarchar(50)  ) |

***Bảng khách hàng***

|  |
| --- |
| CREATE TABLE KhachHang (  MaKH nvarchar(50) CONSTRAINT PK\_KhangHang PRIMARY KEY,  TenKH nvarchar(50) NOT NULL,  SoDienThoai nvarchar(50) CHECK (LEN(SoDienThoai) = 10),  SoDiemTichLuy float CHECK (SoDiemTichLuy >= 0)  ) |

***Bảng sản phẩm***

|  |
| --- |
| CREATE TABLE SanPham(  MaSP nvarchar(50) CONSTRAINT PK\_MaSP PRIMARY KEY,  TenSP nvarchar(50) NOT NULL,  Gia float NOT NULL CHECK (Gia > 0),  AnhSP nvarchar(200) NOT NULL,  MaLoaiSP nvarchar(50) CONSTRAINT FK\_SanPham\_LoaiSP FOREIGN KEY REFERENCES LoaiSanPham(MaLoaiSP)  ) |

***Bảng loại sản phẩm***

|  |
| --- |
| CREATE TABLE LoaiSanPham(  MaLoaiSP nvarchar(50) CONSTRAINT PK\_LoaiSanPham PRIMARY KEY,  TenLoaiSP nvarchar(50) NOT NULL  ) |

***Bảng nguyên liệu***

|  |
| --- |
| CREATE TABLE NguyenLieu(  MaNL nvarchar(50) CONSTRAINT PK\_NguyenLieu PRIMARY KEY,  TenNL nvarchar(50) NOT NULL,  SoLuongTonKho int CHECK (SoLuongTonKho >= 0)  ) |

***Bảng pha chế***

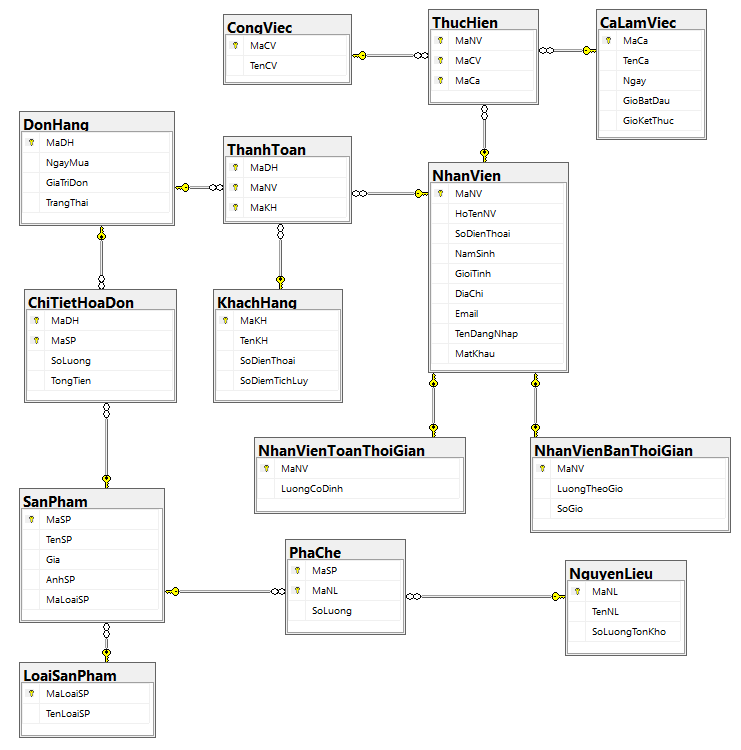
|  |
| --- |
| CREATE TABLE PhaChe (  MaSP nvarchar(50) CONSTRAINT FK\_PhaChe\_SanPHam FOREIGN KEY REFERENCES SanPham(MaSP),  MaNL nvarchar(50) CONSTRAINT FK\_PhaChe\_NguyenLieu FOREIGN KEY REFERENCES NguyenLieu(MaNL),  SoLuong int check (SoLuong >=0),  CONSTRAINT PK\_PhaChe PRIMARY KEY (MaSP, MaNL)  ) |

***Bảng chi tiết hóa đơn***

|  |
| --- |
| CREATE TABLE ChiTietHoaDon (  MaDH nvarchar(50) CONSTRAINT FK\_ChiTietHoaDon\_DonHang FOREIGN KEY REFERENCES DonHang(MaDH),  MaSP nvarchar(50) CONSTRAINT FK\_ChiTietHoaDon\_SanPham FOREIGN KEY REFERENCES SanPham(MaSP),  SoLuong int CHECK (SoLuong>0),  TongTien float CHECK (TongTien>0),  CONSTRAINT PK\_ChiTietHoaDon PRIMARY KEY (MaDH, MaSP)  ) |

***Bảng thanh toán***

|  |
| --- |
| CREATE TABLE ThanhToan (  MaDH nvarchar(50) CONSTRAINT FK\_ThanhToan\_DonHang FOREIGN KEY REFERENCES DonHang(MaDH),  MaNV nvarchar(50) CONSTRAINT FK\_ThanhToan\_NhanVien FOREIGN KEY REFERENCES NhanVien(MaNV),  MaKH nvarchar(50) CONSTRAINT FK\_ThanhToan\_KhachHang FOREIGN KEY REFERENCES KhachHang(MaKH),  CONSTRAINT PK\_ThanhToan PRIMARY KEY (MaDH,MaNV, MaKH)  ) |



**5. Các View**

**a) Xem danh sách sản phẩm đã chọn khi đặt sản phẩm**

|  |
| --- |
| CREATE VIEW v\_DanhSachSanPhamDaChon AS  SELECT sp.MaSP, sp.TenSP, sp.Gia, cthd.SoLuong, cthd.TongTien  FROM ChiTietHoaDon cthd  INNER JOIN SanPham sp ON cthd.MaSP = sp.MaSP  INNER JOIN DonHang dh ON cthd.MaDH = dh.MaDH  WHERE TrangThai = N'Chưa thanh toán' |

**b) Xem danh sách sản phẩm**

|  |
| --- |
| CREATE VIEW v\_DanhSachSanPham AS  SELECT sp.MaSP, sp.TenSP, sp.Gia, lsp.TenLoaiSP, sp.AnhSP  FROM SanPham sp  INNER JOIN LoaiSanPham lsp ON sp.MaLoaiSP = lsp.MaLoaiSP |

**c) Xem những nguyên liệu cần dùng để pha chế sản phẩm**

|  |
| --- |
| CREATE VIEW v\_NguyenLieuDungDePhaCheSanPham AS  SELECT nl.MaNL, nl.TenNL, pc.SoLuong, nl.SoLuongTonKho  FROM NguyenLieu nl  INNER JOIN PhaChe pc ON nl.MaNL = pc.MaNL  INNER JOIN SanPham sp ON pc.MaSP = sp.MaSP |

**d) Xem danh sách nhân viên**

|  |
| --- |
| CREATE VIEW v\_DanhSachNhanVien AS  SELECT nv.\*, nvttg.LuongCoDinh, nvbtg.SoGio, LuongTheoGio  FROM NhanVien nv  FULL OUTER JOIN NhanVienToanThoiGian nvttg ON nv.MaNV = nvttg.MaNV  FULL OUTER JOIN NhanVienBanThoiGian nvbtg ON nv.MaNV = nvbtg.MaNV |

**e) Xem danh sách hóa đơn đã thanh toán**

|  |
| --- |
| CREATE VIEW v\_DanhSachHoaDon AS  SELECT dh.MaDH, dh.NgayMua, kh.SoDienThoai, kh.TenKH, dh.GiaTriDon  FROM DonHang dh  INNER JOIN ChiTietHoaDon cthd ON dh.MaDH = cthd.MaDH  INNER JOIN ThanhToan tt ON dh.MaDH = tt.MaDH  INNER JOIN NhanVien nv ON tt.MaNV = nv.MaNV  INNER JOIN KhachHang kh ON tt.MaKH = kh.MaKH  WHERE dh.TrangThai = N'Đã thanh toán' |

**f) Xem số lượng và tổng tiền đã bán được ứng với từng loại sản phẩm**

|  |
| --- |
| CREATE VIEW v\_SoLuongSanPhamDaBanTheoTungSanPham AS  SELECT sp.MaSP, sp.TenSP, SUM(cthd.SoLuong) SoLuong, SUM(cthd.TongTien) AS TongTien  FROM ChiTietHoaDon cthd  INNER JOIN SanPham sp ON cthd.MaSP = sp.MaSP  GROUP BY sp.MaSP, sp.TenSP |

**6. Các Trigger**

**a) Trigger tự động tạo mã nguyên liệu**

|  |
| --- |
| CREATE TRIGGER trg\_TuDongTaoMaNL\_NguyenLieu  ON NguyenLieu  INSTEAD OF INSERT  AS  BEGIN  BEGIN TRY  DECLARE @maxMaNL NVARCHAR(50);  DECLARE @newMaNL NVARCHAR(50);  DECLARE @numPart INT;    SELECT @maxMaNL = MAX(maNL)  FROM NguyenLieu  WHERE maNL LIKE 'NL%';    IF @maxMaNL IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaNL, 3, LEN(@maxMaNL) - 2) AS INT) + 1;  END  ELSE  BEGIN  SET @numPart = 1;  END  -- Tạo giá trị mới cho maNL, với định dạng NLxx (2 số)  SET @newMaNL = 'NL' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);    INSERT INTO NguyenLieu(MaNL, TenNL, SoLuongTonKho)  SELECT @newMaNL, TenNL, SoLuongTonKho  FROM inserted;    END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

**b) Trigger tự động tạo mã loại sản phẩm**

|  |
| --- |
| CREATE TRIGGER trg\_TuDongTaoMaLoaiSanPham\_LoaiSanPham  ON LoaiSanPham  INSTEAD OF INSERT  AS  BEGIN  BEGIN TRY  DECLARE @maxMaLoaiSP NVARCHAR(50);  DECLARE @newMaLoaiSP NVARCHAR(50);  DECLARE @numPart INT;    SELECT @maxMaLoaiSP = MAX(MaLoaiSP)  FROM LoaiSanPham  WHERE MaLoaiSP LIKE 'LSP%';    IF @maxMaLoaiSP IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaLoaiSP, 4, LEN(@maxMaLoaiSP) - 2) AS INT)+1;  END  ELSE  BEGIN  SET @numPart = 1;  END  -- Tạo giá trị mới cho maLoaiSP, với định dạng LSPxx (2 số)  SET @newMaLoaiSP = 'LSP' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);    INSERT INTO LoaiSanPham(MaLoaiSP, TenLoaiSP)  SELECT @newMaLoaiSP, TenLoaiSP  FROM inserted;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

**c) Trigger tự động tạo mã sản phẩm**

|  |
| --- |
| CREATE TRIGGER trg\_TuDongTaoMaSP\_SanPham  ON SanPham  INSTEAD OF INSERT  AS  BEGIN  BEGIN TRY  DECLARE @maxMaSP NVARCHAR(50);  DECLARE @newMaSP NVARCHAR(50);  DECLARE @numPart INT;    SELECT @maxMaSP = MAX(MaSP)  FROM SanPham  WHERE MaSP LIKE 'SP%';    IF @maxMaSP IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaSP, 3, LEN(@maxMaSP) - 2) AS INT) + 1;  END  ELSE  BEGIN  SET @numPart = 1;  END  -- Tạo giá trị mới cho maSP, với định dạng SPxx (2 số)  SET @newMaSP = 'SP' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);    INSERT INTO SanPham (MaSP, TenSP, Gia, AnhSP, MaLoaiSP)  SELECT @newMaSP, TenSP, Gia, AnhSP, MaLoaiSP  FROM inserted;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

**d) Trigger tự động tạo mã nhân viên**

|  |
| --- |
| CREATE TRIGGER trg\_TuDongTaoMaNV\_NhanVien  ON NhanVien  INSTEAD OF INSERT  AS  BEGIN  BEGIN TRY  DECLARE @maxMaNV NVARCHAR(50);  DECLARE @newMaNV NVARCHAR(50);  DECLARE @numPart INT;  SELECT @maxMaNV = MAX(MaNV)  FROM NhanVien  WHERE MaNV LIKE 'NV%';  IF @maxMaNV IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaNV, 3, LEN(@maxMaNV) - 2) AS INT) + 1;  END  ELSE  BEGIN  SET @numPart = 1;  END  SET @newMaNV = 'NV' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);  INSERT INTO NhanVien (MaNV, HoTenNV, SoDienThoai, NamSinh, GioiTinh, DiaChi, Email, TenDangNhap, MatKhau)  SELECT @newMaNV, HoTenNV, SoDienThoai, NamSinh, GioiTinh, DiaChi, Email,TenDangNhap, MatKhau  FROM inserted;  END TRY  BEGIN CATCH    DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

**e) Trigger kiểm tra số lượng nguyên liệu có đủ để pha chế sản phẩm cho khách**

|  |
| --- |
| CREATE TRIGGER trg\_KiemTraSoLuongTruocKhiThem\_ChiTietHoaDon  ON ChiTietHoaDon  INSTEAD OF INSERT  AS  BEGIN  BEGIN TRY  DECLARE @maDH NVARCHAR(50);  DECLARE @maSP NVARCHAR(50);  DECLARE @soLuong INT;  DECLARE @soLuongTonKho INT;  DECLARE @tongNL INT;  DECLARE @TenNL NVARCHAR(50);    SELECT @maDH = MaDH, @maSP = MaSP  FROM inserted;  -- Kiểm tra sản phẩm đã tồn tại chưa  IF EXISTS (SELECT 1 FROM ChiTietHoaDon WHERE MaDH = @maDH AND MaSP = @maSP)  BEGIN  -- Nếu đã tồn tại, tăng số lượng lên 1  UPDATE ChiTietHoaDon  SET SoLuong = SoLuong + 1  WHERE MaDH = @maDH AND MaSP = @maSP;  -- Cập nhật tổng tiền  SELECT @soLuong = SoLuong  FROM ChiTietHoaDon  WHERE MaDH = @maDH AND MaSP = @maSP;  UPDATE ChiTietHoaDon  SET TongTien = (SELECT Gia \* @soLuong FROM SanPham WHERE MaSP = @maSP)  WHERE MaDH = @maDH AND MaSP = @maSP;  END  ELSE  BEGIN  -- Nếu chưa tồn tại thì thêm sản phẩm mới  INSERT INTO ChiTietHoaDon (MaDH, MaSP, SoLuong, TongTien)  VALUES (@maDH, @maSP, 1, (SELECT Gia \* 1 FROM SanPham WHERE MaSP = @maSP));  END  -- Kiểm tra số lượng nguyên liệu tồn kho sau khi insert (hoặc update)  -- Tạo con trỏ để duyệt qua các dòng trả về từ truy vấn  DECLARE cur CURSOR FOR  SELECT cthd.SoLuong \* pc.SoLuong AS TongNL, nl.SoLuongTonKho AS SoLuongTonKho, nl.TenNL AS TenNL  FROM ChiTietHoaDon cthd  LEFT OUTER JOIN PhaChe pc ON cthd.MaSP = pc.MaSP  LEFT OUTER JOIN NguyenLieu nl ON pc.MaNL = nl.maNL  WHERE MaDH = @maDH AND pc.SoLuong IS NOT NULL;  OPEN cur;  FETCH NEXT FROM cur INTO @tongNL, @soLuongTonKho, @TenNL;  WHILE @@FETCH\_STATUS = 0  BEGIN  -- Kiểm tra nếu tổng nguyên liệu cần lớn hơn số lượng tồn kho  IF @tongNL > @soLuongTonKho  BEGIN  CLOSE cur;  DEALLOCATE cur;  RAISERROR(N'%s không đủ để pha chế', 16, 1, @TenNL); --Khi RAISERROR thì SQL server sẽ tự động rollback trigger  END;  FETCH NEXT FROM cur INTO @tongNL, @soLuongTonKho, @TenNL;  END;  CLOSE cur;  DEALLOCATE cur;    END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

**f) Trigger đảm bảo nhân viên không vi phạm quy định về số ca làm và loại ca trong cùng ngày khi đăng ký**

|  |
| --- |
| CREATE TRIGGER Trg\_LuatPhanCa\_PhanCa  ON ThucHien  AFTER INSERT  AS  BEGIN  BEGIN TRY  DECLARE @MaNV nvarchar(50), @Ngay DATE, @MaCa nvarchar(50), @TenCa nvarchar(50);  -- Lấy thông tin nhân viên và ca làm việc mới thêm vào  SELECT @MaNV = MaNV, @MaCa = MaCa FROM inserted;    -- Lấy ngày làm việc từ bảng CaLamViec dựa trên MaCa  SELECT @Ngay = Ngay, @TenCa = TenCa FROM CaLamViec WHERE MaCa = @MaCa;  -- Nhân viên không được làm quá 2 ca trong cùng 1 ngày  IF (SELECT COUNT(\*)  FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE TH.MaNV = @MaNV AND CLV.Ngay = @Ngay) > 2  BEGIN  RAISERROR (N'Nhân viên này không được làm quá 2 ca trong 1 ngày. Vui lòng kiểm tra lại.', 16, 1);  END  -- Kiểm tra nhân viên không được làm trùng ca làm việc (tức ca đó nhân viên đã làm rồi mà lại đăng kí nữa)  IF (SELECT COUNT(\*)  FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE MaNV = @MaNV AND TenCa = @TenCa AND Ngay = @Ngay) > 1  BEGIN  RAISERROR (N'Ca làm việc bị trùng lặp cho nhân viên này trong ngày. Vui lòng kiểm tra lại.', 16, 1);  END  -- Làm fulltimeAM thì ko làm ca khác ngoại trừ ca Tối  IF (@TenCa = N'Sáng' OR @TenCa = N'FulltimePM')  BEGIN    IF (EXISTS (SELECT 1 FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE TH.MaNV = @MaNV  AND CLV.TenCa = N'FulltimeAM' -- Kiểm tra nếu nhân viên đã làm FulltimeAM trong cùng ngày  AND CLV.Ngay = @Ngay))  BEGIN  RAISERROR (N'Nhân viên đã làm FulltimeAM, không được làm ca Sáng, Chiều và FulltimePM .', 16, 1);  END  END  -- Làm fulltimePM thì ko làm ca khác ngoại trừ ca Sáng  ELSE IF ( @TenCa = N'Tối' OR @TenCa = N'FulltimeAM')  BEGIN  IF (EXISTS (SELECT 1 FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE TH.MaNV = @MaNV  AND CLV.TenCa = N'FulltimePM'  AND CLV.Ngay = @Ngay))  BEGIN  RAISERROR (N'Nhân viên đã làm FulltimePM, không được làm ca Chiều, Tối và FulltimeAM.', 16, 1);  END  END  --Làm FulltimeAM or FulltimePM thì không đk ca Chiều (do ca Chiều là giao thoa nên không để chung với 2 ĐK trên)  ELSE IF (@TenCa = N'Chiều' )  BEGIN  IF (EXISTS (SELECT 1 FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE TH.MaNV = @MaNV  AND (CLV.TenCa = N'FulltimeAM' OR CLV.TenCa = N'FulltimePM')  AND CLV.Ngay = @Ngay))  BEGIN  RAISERROR (N'Nhân viên đã làm FulltimeAM hoặc FulltimePM , không được làm ca Chiều .', 16, 1);  END  END  -------------------------------------  --Làm Sáng hoặc Chiều thì ko làm fulltimeAM nữa  IF (@TenCa = N'FulltimeAM' )  BEGIN  IF (EXISTS (SELECT 1 FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE TH.MaNV = @MaNV  AND (CLV.TenCa = N'Sáng' OR CLV.TenCa = N'Chiều')  AND CLV.Ngay = @Ngay))  BEGIN  RAISERROR (N'Nhân viên đã làm ca Sáng hoặc Chiều không thể đăng ki ca FulltimeAM .', 16, 1);  END  END    --Làm Tối hoặc Chiều thì ko làm fulltimePM nữa  IF (@TenCa = N'FulltimePM' )  BEGIN  IF (EXISTS (SELECT 1 FROM ThucHien TH  JOIN CaLamViec CLV ON TH.MaCa = CLV.MaCa  WHERE TH.MaNV = @MaNV  AND (CLV.TenCa = N'Chiều' OR CLV.TenCa = N'Tối')  AND CLV.Ngay = @Ngay))  BEGIN  RAISERROR (N'Nhân viên đã làm ca Chiều hoặc Tối không thể đăng ki ca FulltimePM .', 16, 1);  END  END    END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

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

1. **Kết nối cơ sở dữ liệu.**

|  |
| --- |
| public class DBConnection  {    public static SqlConnection conn = new SqlConnection(@"Data Source=(localdb)\mssqllocaldb;Initial Catalog=QuanLyQuanCaPhe;User Id=" + GLOBAL.tenDangNhap + ";Password=" + GLOBAL.matKhau+ ";");      public DBConnection()  {    }  public static void moKetNoi()  {  try  {  if (conn.State == ConnectionState.Closed)  {  conn.Open();  //MessageBox.Show("Kết nối thành công");  }  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  public static void dongKetNoi()  {  if (conn != null && conn.State == ConnectionState.Open)  {  conn.Close();  //MessageBox.Show("Đóng kết nối thành công");  }  else  {  MessageBox.Show("Chưa tạo kết nối");  }  }  } |

## **Các hàm thực thi thủ tục, hàm chung.**

### **Hàm load table và view.**

|  |
| --- |
| public static DataTable LoadTableVaView(string tenTable)  {  try  {  moKetNoi();  string truyVan = string.Format($"SELECT \* FROM {tenTable}");  SqlDataAdapter adapter = new SqlDataAdapter(truyVan, conn);  DataTable dt = new DataTable();  adapter.Fill(dt);  return dt;  }  catch (Exception exc)  {  MessageBox.Show(exc.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dongKetNoi();  }  } |

### **Hàm thực thi function scalar.**

|  |
| --- |
| public static object ThucThiFunction\_Scalar(string tenFunction, List<KeyValuePair<string, object>> parameters)  {  try  {  moKetNoi();  string truyVan = $@"SELECT dbo.{tenFunction}(";  for (int i = 0; i < parameters.Count; i++)  {  truyVan += parameters[i].Key;  if (i < parameters.Count - 1)  {  truyVan += ", ";  }  }  truyVan += ")";  SqlCommand cmd = new SqlCommand(truyVan, conn);  foreach (var param in parameters)  {  cmd.Parameters.AddWithValue(param.Key, param.Value);  }  return cmd.ExecuteScalar();  }  catch (Exception ex)  {  MessageBox.Show("Thất bại\n" + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dongKetNoi();  }  } |

### **Hàm thực thi function inline và multi statement**

|  |
| --- |
| public static DataTable ThucThiFunction\_InlineVaMultiStatement(string tenFunction, List<KeyValuePair<string, object>> parameters)  {  try  {  moKetNoi();  string truyVan = $@"SELECT \* FROM {tenFunction}(";  for (int i = 0; i < parameters.Count; i++)  {  truyVan += parameters[i].Key;  if (i < parameters.Count - 1)  {  truyVan += ", ";  }  else  {  truyVan += ")";  }  }  SqlCommand cmd = new SqlCommand(truyVan, conn);  foreach (var param in parameters)  {  cmd.Parameters.AddWithValue(param.Key, param.Value);  }  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  adapter.Fill(dt);  return dt;  }  catch (Exception ex)  {  MessageBox.Show("Thất bại\n" + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dongKetNoi();  }  } |

* 1. **Hàm thực thi thủ tục có tham số output**

|  |
| --- |
| public static Dictionary<string, object> ThucThiProc\_CoThamSoOutput(string tenProc, List<KeyValuePair<string, object>> parameters, List<SqlParameter> outputParams)  {  try  {  moKetNoi();  SqlCommand cmd = new SqlCommand(tenProc, conn);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm các tham số đầu vào  if (parameters != null)  {  foreach (var param in parameters)  {  cmd.Parameters.AddWithValue(param.Key, param.Value);  }  }  // Thêm các tham số OUTPUT  foreach (var outputParam in outputParams)  {  cmd.Parameters.Add(outputParam);  }  // Thực thi stored procedure  cmd.ExecuteNonQuery();  // Lấy giá trị của tất cả các tham số OUTPUT  Dictionary<string, object> result = new Dictionary<string, object>();  foreach (var outputParam in outputParams)  {  result.Add(outputParam.ParameterName, cmd.Parameters[outputParam.ParameterName].Value);  }  return result;  }  catch (Exception ex)  {  MessageBox.Show("Thất bại\n" + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dongKetNoi();  }  } |

* 1. **Hàm thực thi thủ tục có tham số hoặc không có tham số**

|  |
| --- |
| public static void ThucThiProc\_CoThamSoVaKhongCoThamSo(string tenProc, List<KeyValuePair<string, object>> parameters)  {  try  {  moKetNoi();  SqlCommand cmd = new SqlCommand(tenProc, conn);  cmd.CommandType = CommandType.StoredProcedure;  // Kiểm tra nếu có tham số  if (parameters != null && parameters.Count > 0)  {  foreach (var param in parameters)  {  if (param.Value.Equals(""))  {  cmd.Parameters.AddWithValue(param.Key, null);  }  else  {  cmd.Parameters.AddWithValue(param.Key, param.Value);  }  }  }  cmd.ExecuteNonQuery();  //if (cmd.ExecuteNonQuery() > 0)  //MessageBox.Show("Thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Thất bại\n" + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  dongKetNoi();  }  } |

* 1. **Hàm thực thi thủ tục có reader**

|  |
| --- |
| public static SqlDataReader ThucThiProc\_CoReader(string tenProc, List<KeyValuePair<string, object>> parameters, bool traVeReader = false)  {  SqlDataReader reader = null;  try  {  moKetNoi();  SqlCommand cmd = new SqlCommand(tenProc, conn);  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số vào cmd  foreach (var param in parameters)  {  cmd.Parameters.AddWithValue(param.Key, param.Value);  }  if (traVeReader)  {  // Trả về SqlDataReader nếu yêu cầu  reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);  }  else  {  // Thực thi không trả về dữ liệu  if (cmd.ExecuteNonQuery() > 0)  {  MessageBox.Show("Thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  }  catch (Exception ex)  {  MessageBox.Show("Thất bại\n" + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  return reader;  } |

1. **Xem danh mục**

**3.1. Xem danh sách sản phẩm đã chọn khi đặt sản phẩm**

Thực thi trên C#

|  |
| --- |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("v\_DanhSachSanPhamDaChon");  } |
| private void fBill\_Load(object sender, EventArgs e)  {  rpvBill.LocalReport.ReportEmbeddedResource = "QuanLyQuanCaPhe.Views.ReportBill.rdlc";  ReportDataSource rds = new ReportDataSource();  rds.Name = "DataSet1";  rds.Value = DBConnection.LoadTableVaView("v\_DanhSachSanPhamDaChon");    rpvBill.LocalReport.DataSources.Add(rds);  ReportParameter[] parameters = new ReportParameter[7];  parameters[0] = new ReportParameter("txtNgayMua", ngayMua);  parameters[1] = new ReportParameter("txtTenNV", tenNV);  parameters[2] = new ReportParameter("txtTenKH", tenKH);  parameters[3] = new ReportParameter("txtMaDH", maDH);  parameters[4] = new ReportParameter("txtTong", tong);  parameters[5] = new ReportParameter("txtGiam", giam);  parameters[6] = new ReportParameter("txtThanhTien", thanhTien);  rpvBill.LocalReport.SetParameters(parameters);  rpvBill.RefreshReport();  } |

**3.2. Xem danh sách sản phẩm**

Thực thi trên C#

|  |
| --- |
| private void LoadDgvDSSanPham()  {  dgvDSSanPham.DataSource = DBConnection.LoadTableVaView("v\_DanhSachSanPham");  } |
| private void LoadFlowPnlSanPham() //Mặc định, dùng để tạo listUCSanPham và thêm vào flowPnlSanPham  {  flowPnlSanPham.Controls.Clear();  DataTable dataTable = DBConnection.LoadTableVaView("SanPham");  ucSanPham sanPham;    foreach (DataRow row in dataTable.Rows)  {  sanPham = new ucSanPham();  sanPham.lblTenSP.Text = row["TenSP"].ToString(); // Lấy giá trị của cột TenSP  sanPham.lblGia.Text = Tien.DinhDangTien(row["Gia"].ToString());  sanPham.pbSanPham.Image = XyLyAnh.GetImage(row["AnhSP"].ToString());  sanPham.lblMaSP.Text = row["MaSP"].ToString();  sanPham.SuKienGoiFormChaTuFormCon += UCSanPhamGoiFTrangChu;  listUCSanPham.Add(sanPham);  }  ThemListUCVaoFlowPnlSanPham();  } |

**3.3. Xem những nguyên liệu cần dùng để pha chế sản phẩm**

Thực thi trên C#

|  |
| --- |
| private void fQLSanPham\_Load(object sender, EventArgs e)  {  LoadDgvDSSanPham();  dgvDSNguyenLieuCuaSanPham.Columns.Clear();  dgvDSNguyenLieuCuaSanPham.DataSource = DBConnection.LoadTableVaView("v\_NguyenLieuDungDePhaCheSanPham");    DataGridViewButtonColumn buttonColumn = new DataGridViewButtonColumn();  buttonColumn.HeaderText = "Action  buttonColumn.Text = "Xóa";  buttonColumn.UseColumnTextForButtonValue = true;  dgvDSNguyenLieuCuaSanPham.Columns.Add(buttonColumn);  LoadComboBox();  } |

**3.4. Xem danh sách nhân viên**

Thực thi trên C#

|  |
| --- |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("v\_DanhSachNhanVien");  } |

**3.5. Xem danh sách hóa đơn đã thanh toán**

Thực thi trên C#

|  |
| --- |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("v\_DanhSachHoaDon");  } |

**3.6. Xem số lượng và tổng tiền đã bán được ứng với từng loại sản phẩm**

Thực thi trên C#

|  |
| --- |
| private void LoadDoThiSoLuongDaBanTheoTungSanPham()  {  DataTable dt = DBConnection.LoadTableVaView("v\_SoLuongSanPhamDaBanTheoTungSanPham");  // Kiểm tra DataTable có null hoặc rỗng không  if (dt == null || dt.Rows.Count == 0)  {  MessageBox.Show("Không có dữ liệu để hiển thị đồ thị.");  return;  }  // Lấy dữ liệu từ DataTable và vẽ đồ thị  GraphPane myPane = zgcSoLuongDaBanTheoTungSanPham.GraphPane;  myPane.Title.Text = "";  myPane.XAxis.Title.Text = "Tên Sản phẩm";  myPane.YAxis.Title.Text = "Số Lượng";  // Thiết lập trục X về kiểu Text  myPane.XAxis.Type = AxisType.Text;  // Khởi tạo PointPairList cho các cột  PointPairList points = new PointPairList();  // Khởi tạo mảng string[] cho TextLabels  string[] textLabels = new string[dt.Rows.Count]; // Tạo mảng với số phần tử là số lượng dòng trong DataTable  int index = 0; // Biến index để quản lý chỉ số của trục X  foreach (DataRow row in dt.Rows)  {  // Kiểm tra nếu cột có dữ liệu hợp lệ  string tenSP = row["TenSP"].ToString();  double soLuong = 0;  if (row["SoLuong"] != DBNull.Value)  {  soLuong = Convert.ToDouble(row["SoLuong"]);  }  // Sử dụng chỉ số index làm giá trị trục X  points.Add(index, soLuong); // Chỉ số index dùng làm giá trị trục X  // Thêm tên sản phẩm vào mảng textLabels cho trục X  textLabels[index] = tenSP;  index++;  }  // Thiết lập TextLabels vào trục X, cần chỉ rõ các nhãn và chỉ số của trục X  myPane.XAxis.Scale.TextLabels = textLabels;  // Thiết lập độ rộng cột  myPane.BarSettings.Type = BarType.Cluster; // Đảm bảo sử dụng Cluster để các cột không chồng lên nhau  // Thêm biểu đồ cột vào đồ thị  BarItem myBar = myPane.AddBar("Số Lượng", points, System.Drawing.Color.Blue);  // Thiết lập xoay tên trục X nếu các tên sản phẩm quá dài  myPane.XAxis.Scale.FontSpec.Angle = 0; // Xoay tên sản phẩm 45 độ nếu cần  // Cập nhật đồ thị  zgcSoLuongDaBanTheoTungSanPham.AxisChange();  zgcSoLuongDaBanTheoTungSanPham.Invalidate();  } |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("v\_SoLuongSanPhamDaBanTheoTungSanPham");  } |

**3.7. Xem danh sách khách hàng**

Thực thi trên C#

|  |
| --- |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("KhachHang");  } |

**3.8. Xem danh sách nguyên liệu**

Thực thi trên C#

|  |
| --- |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("NguyenLieu");  } |

**3.9. Xem danh sách loại sản phẩm**

Thực thi trên C#

|  |
| --- |
| private void LoadDGVHienThi()  {  dgvHienThi.DataSource = DBConnection.LoadTableVaView("LoaiSanPham");  } |

**3.10. Xem danh sách loại sản phẩm và danh sách nguyên liệu**

Thực thi trên C#

|  |
| --- |
| private void LoadComboBox()  {  DataTable tableLoaiSanPham = DBConnection.LoadTableVaView("LoaiSanPham");  cboLoaiSP.DataSource = tableLoaiSanPham;  cboLoaiSP.DisplayMember = "TenLoaiSP";  cboLoaiSP.ValueMember = "MaLoaiSP";  DataTable tableNguyenLieu = DBConnection.LoadTableVaView("NguyenLieu");  cboTenNL.DataSource = tableNguyenLieu;  cboTenNL.DisplayMember = "TenNL";  cboTenNL.ValueMember = "MaNL";  } |

**4. Đăng ký**

**4.1. Thêm nhân viên**

|  |
| --- |
| CREATE PROCEDURE proc\_ThemNhanVien  @HoTenNV nvarchar(50),  @SoDienThoai nvarchar(50),  @NamSinh int,  @GioiTinh nvarchar(50),  @DiaChi nvarchar(200),  @Email nvarchar(50),  @TenDangNhap nvarchar(50),  @MatKhau nvarchar(50),  @LoaiNhanVien nvarchar(20)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO NhanVien (HoTenNV, SoDienThoai, NamSinh, GioiTinh, DiaChi, Email,TenDangNhap, MatKhau)  VALUES (@HoTenNV, @SoDienThoai, @NamSinh, @GioiTinh, @DiaChi, @Email, @TenDangNhap, @MatKhau);  DECLARE @mNV nvarchar(50);  SELECT @mNV = MaNV  FROM NhanVien  WHERE SoDienThoai = @SoDienThoai;  IF @LoaiNhanVien = N'Toàn thời gian'  BEGIN  INSERT INTO NhanVienToanThoiGian (MaNV, LuongCoDinh)  VALUES (@mNV, 0);  END  ELSE IF @LoaiNhanVien = N'Bán thời gian'  BEGIN  INSERT INTO NhanVienBanThoiGian (MaNV, LuongTheoGio, SoGio)  VALUES (@mNV, 0, 0);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

Thực thi trên C#

|  |
| --- |
| public static void Them(NhanVien lnv, string loaiNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@HoTenNV", lnv.HoTenNV);  DBConnection.AddParameters("@SoDienThoai", lnv.SoDienThoai);  DBConnection.AddParameters("@NamSinh", lnv.NamSinh);  DBConnection.AddParameters("@GioiTinh", lnv.GioiTinh);  DBConnection.AddParameters("@DiaChi", lnv.DiaChi);  DBConnection.AddParameters("@Email", lnv.Email);  DBConnection.AddParameters("@TenDangNhap", lnv.TenDangNhap);  DBConnection.AddParameters("@MatKhau", lnv.MatKhau);  DBConnection.AddParameters("@LoaiNhanVien", loaiNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_ThemNhanVien", DBConnection.parameters);  } |

**5. Đăng nhập**

**5.1. Tìm kiếm nhân viên dựa trên tên đăng nhập và mật khẩu**

|  |
| --- |
| CREATE PROCEDURE proc\_timNhanVienTheoTaiKhoan  @TenDangNhap nvarchar(50),  @MatKhau nvarchar(50),  @HoTenNV nvarchar(50) OUTPUT,  @SoDienThoai nvarchar(50) OUTPUT,  @NamSinh int OUTPUT,  @GioiTinh nvarchar(50) OUTPUT,  @DiaChi nvarchar(50) OUTPUT,  @Email nvarchar(50) OUTPUT,  @MaNV nvarchar(50) OUTPUT,  @LuongCoDinh float OUTPUT,  @LuongTheoGio float OUTPUT,  @SoGio int OUTPUT  AS  BEGIN  SELECT  @HoTenNV = NhanVien.HoTenNV,  @SoDienThoai = NhanVien.SoDienThoai,  @NamSinh = NhanVien.NamSinh,  @GioiTinh = NhanVien.GioiTinh,  @DiaChi = NhanVien.DiaChi,  @Email = NhanVien.Email,  @MaNV = NhanVien.MaNV,  @LuongCoDinh = NhanVienToanThoiGian.LuongCoDinh,  @LuongTheoGio = NhanVienBanThoiGian.LuongTheoGio,  @SoGio = NhanVienBanThoiGian.SoGio  FROM  NhanVien  LEFT OUTER JOIN NhanVienToanThoiGian ON NhanVien.MaNV = NhanVienToanThoiGian.MaNV  LEFT OUTER JOIN NhanVienBanThoiGian ON NhanVien.MaNV = NhanVienBanThoiGian.MaNV  WHERE  NhanVien.TenDangNhap = @TenDangNhap  AND NhanVien.MatKhau = @MatKhau;  END; |

Thực thi trên C#

|  |
| --- |
| public static NhanVien timKiemNhanVienTheoTaiKhoan(string tenDangNhap, string matKhau)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenDangNhap", tenDangNhap);  DBConnection.AddParameters("@MatKhau", matKhau);  List<SqlParameter> outputParameters = new List<SqlParameter>  {  new SqlParameter("@HoTenNV", SqlDbType.NVarChar, 50) { Direction = ParameterDirection.Output },  new SqlParameter("@SoDienThoai", SqlDbType.NVarChar, 50) { Direction = ParameterDirection.Output },  new SqlParameter("@NamSinh", SqlDbType.Int) { Direction = ParameterDirection.Output },  new SqlParameter("@GioiTinh", SqlDbType.NVarChar,50) { Direction = ParameterDirection.Output },  new SqlParameter("@DiaChi", SqlDbType.NVarChar,200) { Direction = ParameterDirection.Output },  new SqlParameter("@Email", SqlDbType.NVarChar,50) { Direction = ParameterDirection.Output },  new SqlParameter("@MaNV", SqlDbType.NVarChar,50) { Direction = ParameterDirection.Output },  new SqlParameter("@LuongCoDinh", SqlDbType.Float) { Direction = ParameterDirection.Output },  new SqlParameter("@LuongTheoGio", SqlDbType.Float) { Direction = ParameterDirection.Output },  new SqlParameter("@SoGio", SqlDbType.Float) { Direction = ParameterDirection.Output },  };  Dictionary<string, object> result = DBConnection.ThucThiProc\_CoThamSoOutput("proc\_timNhanVienTheoTaiKhoan", DBConnection.parameters, outputParameters);  if (result != null)  {  NhanVien nv = new NhanVien();  nv.MaNV = result["@MaNV"].ToString();  //MessageBox.Show(nv.MaNV);  nv.HoTenNV = result["@HoTenNV"].ToString();  nv.SoDienThoai = result["@SoDienThoai"].ToString();  nv.NamSinh = Convert.ToInt32(result["@NamSinh"].ToString());  nv.GioiTinh = result["@GioiTinh"].ToString();  nv.DiaChi = result["@DiaChi"].ToString();  nv.Email = result["@Email"].ToString();  nv.MaNV = result["@MaNV"].ToString();  nv.TenDangNhap = tenDangNhap;  nv.MatKhau = matKhau;  return nv;  }  else  {  return null;  }  } |

**6. Quên mật khẩu**

**6.1. Tìm kiếm mật khẩu dựa trên tên đăng nhập và số điện thoại**

|  |
| --- |
| CREATE FUNCTION func\_LayMatKhau\_NhanVien  (  @TenDangNhap NVARCHAR(50),  @SoDienThoai NVARCHAR(50)  )  RETURNS NVARCHAR(50)  AS  BEGIN  DECLARE @MatKhau NVARCHAR(50);    SELECT @MatKhau = MatKhau  FROM NhanVien  WHERE TenDangNhap = @TenDangNhap AND SoDienThoai = @SoDienThoai;    RETURN @MatKhau;  END |

Thực thi trên C#

|  |
| --- |
| public static string LayMatKhau(string tenDangNhap, string soDienThoai)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenDangNhap", tenDangNhap);  DBConnection.AddParameters("@SoDienThoai", soDienThoai);  return DBConnection.ThucThiFunction\_Scalar("func\_LayMatKhau\_NhanVien", DBConnection.parameters).ToString();  } |

**7. Đơn hàng trước khi thanh toán**

**7.1. Load mã đơn hàng chưa thanh toán**

|  |
| --- |
| CREATE FUNCTION func\_LayMaDHChuaThanhToan\_DonHang()  RETURNS NVARCHAR(50)  AS  BEGIN  DECLARE @MaDH NVARCHAR(50)  SELECT @MaDH = MaDH  FROM DonHang  WHERE TrangThai = N'Chưa thanh toán';  RETURN @MaDH;  END; |

Thực thi trên C#

|  |
| --- |
| public static string LayMaDHChuaThanhToan()  {  DBConnection.ClearParameters();  object maDH = DBConnection.ThucThiFunction\_Scalar("func\_LayMaDHChuaThanhToan\_DonHang", DBConnection.parameters);  if (maDH is DBNull || maDH == null)  return "Please click \"New Order\"";  return maDH.ToString();  } |

**7.2. Tính tổng tiền của một hóa đơn**

|  |
| --- |
| CREATE FUNCTION func\_TongTien\_ChiTietHoaDon  (  )  RETURNS FLOAT  AS  BEGIN  DECLARE @MaDH NVARCHAR(50)  DECLARE @Tong FLOAT = 0  SELECT @MaDH = MaDH  FROM DonHang  WHERE TrangThai = N'Chưa thanh toán'    SELECT @Tong = SUM(TongTien)  FROM ChiTietHoaDon  WHERE MaDH = @MaDH  RETURN @Tong  END; |

Thực thi trên C#

|  |
| --- |
| public static double TinhTongTien()  {  DBConnection.ClearParameters();  object tongTien = DBConnection.ThucThiFunction\_Scalar("func\_TongTien\_ChiTietHoaDon", DBConnection.parameters);  if (tongTien is DBNull || tongTien == null)  return 0;  return Convert.ToDouble(tongTien);  } |

**7.3. Tìm kiếm khách hàng dựa trên số điện thoại**

|  |
| --- |
| CREATE PROCEDURE proc\_LoadThongTin\_KhachHang  @SoDienThoai NVARCHAR(50),  @Ten NVARCHAR(50) OUTPUT,  @SoDiemTichLuy FLOAT OUTPUT  AS  BEGIN  SELECT @Ten = kh.TenKH, @SoDiemTichLuy = kh.SoDiemTichLuy  FROM KhachHang kh  WHERE SoDienThoai = @SoDienThoai  END |

Thực thi trên C#

|  |
| --- |
| public static KhachHang LoadThongTinKhachHang(string soDienThoai)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@SoDienThoai", soDienThoai);      // Danh sách tham số OUTPUT  List<SqlParameter> outputParameters = new List<SqlParameter>  {  new SqlParameter("@Ten", SqlDbType.NVarChar, 50) { Direction = ParameterDirection.Output },  new SqlParameter("@SoDiemTichLuy", SqlDbType.NVarChar, 50) { Direction = ParameterDirection.Output }  };  // Gọi stored procedure và nhận kết quả  Dictionary<string, object> result = DBConnection.ThucThiProc\_CoThamSoOutput("proc\_LoadThongTin\_KhachHang", DBConnection.parameters, outputParameters);  // Kiểm tra kết quả và hiển thị thông tin  if (result != null)  {  KhachHang kh = new KhachHang();  kh.TenKH = result["@Ten"].ToString();  kh.SoDiemTichLuy = Convert.ToDouble(result["@SoDiemTichLuy"]);  return kh;  }  else  {  return null;  }  } |

**7.4. Sửa đơn hàng, điểm tích lũy khách hàng và số lượng tồn kho nguyên liệu.**

|  |
| --- |
| CREATE PROCEDURE proc\_XacNhanThanhToan\_DonHangVaKhachHang  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DECLARE @GiaTriDon FLOAT  DECLARE @SoDiemTichLuy FLOAT  DECLARE @MaKH NVARCHAR(50)  DECLARE @maDHChuaThanhToan NVARCHAR(50)  SELECT @maDHChuaThanhToan = MaDH  FROM DonHang  WHERE TrangThai = N'Chưa thanh toán'  SELECT @GiaTriDon = dh.GiaTriDon, @MaKH = kh.MaKH  FROM DonHang dh  INNER JOIN ThanhToan tt ON dh.MaDH = tt.MaDH  INNER JOIN KhachHang kh ON tt.MaKH = kh.MaKH  WHERE dh.MaDH = @maDHChuaThanhToan  SELECT @SoDiemTichLuy = SoDiemTichLuy  FROM KhachHang  WHERE MaKH = @MaKH    IF @GiaTriDon > @SoDiemTichLuy  BEGIN  SET @GiaTriDon = @GiaTriDon - @SoDiemTichLuy  SET @SoDiemTichLuy = 0 + 0.01 \* @GiaTriDon -- Đã dùng hết điểm tích lũy  END  ELSE  BEGIN  SET @SoDiemTichLuy = (@SoDiemTichLuy - @GiaTriDon) + 0.01 \* @GiaTriDon  SET @GiaTriDon = 0 -- Đơn đã được thanh toán hết  END    UPDATE DonHang  SET GiaTriDon = @GiaTriDon  WHERE MaDH = @maDHChuaThanhToan;    UPDATE KhachHang  SET SoDiemTichLuy = @SoDiemTichLuy  WHERE MaKH = @MaKH    UPDATE DonHang  SET TrangThai = N'Đã thanh toán'  WHERE MaDH = @maDHChuaThanhToan    UPDATE nl  SET nl.SoLuongTonKho = nl.SoLuongTonKho - (cthd.SoLuong \* pc.SoLuong)  FROM ChiTietHoaDon cthd  LEFT OUTER JOIN PhaChe pc ON cthd.MaSP = pc.MaSP  LEFT OUTER JOIN NguyenLieu nl ON pc.MaNL = nl.maNL  WHERE cthd.MaDH = @maDHChuaThanhToan AND pc.SoLuong IS NOT NULL;    COMMIT TRANSACTION  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH    END |

Thực thi trên C#

|  |
| --- |
| public static void XacNhanThanhToan()  {  DBConnection.ClearParameters();  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_XacNhanThanhToan\_DonHangVaKhachHang", DBConnection.parameters);  } |

**7.5. Sửa khách hàng**

|  |
| --- |
| CREATE PROCEDURE proc\_Sua\_KhachHang  @TenKH NVARCHAR(50),  @SoDienThoaiCu NVARCHAR(50),  @SoDienThoaiMoi NVARCHAR(50)  AS  BEGIN  UPDATE KhachHang  SET TenKH = @TenKH, SoDienThoai = @SoDienThoaiMoi  WHERE SoDienThoai = @SoDienThoaiCu;  END |

Thực thi trên C#

|  |
| --- |
| public static void SuaTenVaSoDienThoai(string ten, string soDienThoaiCu, string soDienThoaiMoi)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenKH", ten);  DBConnection.AddParameters("@SoDienThoaiCu", soDienThoaiCu);  DBConnection.AddParameters("@SoDienThoaiMoi", soDienThoaiMoi);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Sua\_KhachHang", DBConnection.parameters);  MessageBox.Show("Sửa thành công");  } |

**8. Quản lý Hóa đơn**

**8.1. Tìm kiếm hóa đơn theo số điện thoại**

|  |
| --- |
| REATE FUNCTION Func\_TimKiemTheoSoDienThoai\_KhachHang(@SoDienThoai NVARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM KhachHang  WHERE SoDienThoai LIKE '%' + @SoDienThoai + '%'  ); |

Thực thi trên C#

|  |
| --- |
| public static DataTable TimKiemTheoSDT(string soDienThoai)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@SoDienThoai", soDienThoai);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_TimKiemTheoSoDienThoai\_DonHang", DBConnection.parameters);  } |

**9. Quản lý Khách hàng**

**9.1. Thêm khách hàng**

|  |
| --- |
| CREATE PROCEDURE Proc\_ThemKhachHang\_KhachHang  @MaKH NVARCHAR(50),  @TenKH NVARCHAR(50),  @SoDienThoai NVARCHAR(10),  @SoDiemTichLuy FLOAT  AS  BEGIN  INSERT INTO KhachHang (MaKH, TenKH, SoDienThoai, SoDiemTichLuy)  VALUES (@MaKH, @TenKH, @SoDienThoai, @SoDiemTichLuy);  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(KhachHang kh)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaKH", kh.MaKH);  DBConnection.AddParameters("@TenKH", kh.TenKH);  DBConnection.AddParameters("@SoDienThoai", kh.SoDienThoai);  DBConnection.AddParameters("@SoDiemTichLuy", kh.SoDiemTichLuy);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("Proc\_ThemKhachHang\_KhachHang", DBConnection.parameters);  } |

**9.2. Sửa khách hàng**

|  |
| --- |
| CREATE PROCEDURE Proc\_SuaThongTinKhachhang\_KhachHang  @MaKH NVARCHAR(50),  @TenKH NVARCHAR(50),  @SoDienThoai NVARCHAR(10),  @SoDiemTichLuy FLOAT  AS  BEGIN  UPDATE KhachHang  SET  TenKH = @TenKH,  SoDienThoai = @SoDienThoai,  SoDiemTichLuy = @SoDiemTichLuy  WHERE  MaKH = @MaKH;  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(KhachHang kh)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaKH", kh.MaKH);  DBConnection.AddParameters("@TenKH", kh.TenKH);  DBConnection.AddParameters("@SoDienThoai", kh.SoDienThoai);  DBConnection.AddParameters("@SoDiemTichLuy", kh.SoDiemTichLuy);    DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("Proc\_SuaThongTinKhachhang\_KhachHang", DBConnection.parameters);  } |

**9.3. Xóa khách hàng**

|  |
| --- |
| CREATE PROCEDURE Proc\_XoaKhachHang\_KhachHang  @MaKH NVARCHAR(50)  AS  BEGIN  DELETE FROM KhachHang  WHERE MaKH = @MaKH;  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(KhachHang kh)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaKH", kh.MaKH);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("Proc\_XoaKhachHang\_KhachHang", DBConnection.parameters);  } |

**9.4. Tìm kiếm khách hàng theo số điện thoại**

|  |
| --- |
| CREATE FUNCTION Func\_TimKiemTheoSoDienThoai\_KhachHang(@SoDienThoai NVARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM KhachHang  WHERE SoDienThoai LIKE '%' + @SoDienThoai + '%'  ); |

Thực thi trên C#

|  |
| --- |
| public static DataTable TimKiem(string sdt)  {  DBConnection.parameters = new List<KeyValuePair<string, object>>  {  new KeyValuePair<string, object>("@SoDienThoai", sdt)  };  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("Func\_TimKiemTheoSoDienThoai\_KhachHang", DBConnection.parameters);  } |

**10. Quản lý Kho nguyên liệu**

**10.1. Thêm nguyên liệu**

|  |
| --- |
| CREATE PROCEDURE proc\_Them\_NguyenLieu  @MaNL NVARCHAR(50),  @TenNL NVARCHAR(50),  @SoLuongTonKho INT  AS  BEGIN  INSERT INTO NguyenLieu(MaNL, TenNL, SoLuongTonKho)  VALUES (@MaNL, @TenNL, @SoLuongTonKho)  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(NguyenLieu nl)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNL", nl.MaNL);  DBConnection.AddParameters("@TenNL", nl.TenNL);  DBConnection.AddParameters("@SoLuongTonKho", nl.SoLuongTonKho);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Them\_NguyenLieu", DBConnection.parameters);  } |

**10.2. Sửa nguyên liệu**

|  |
| --- |
| CREATE PROCEDURE proc\_Sua\_NguyenLieu  @MaNL NVARCHAR(50),  @TenNL NVARCHAR(50),  @SoLuongTonKho INT  AS  BEGIN  UPDATE NguyenLieu  SET TenNL = @TenNL,  SoLuongTonKho = @SoLuongTonKho  WHERE MaNL = @MaNL  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(NguyenLieu nl)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNL",nl.MaNL);  DBConnection.AddParameters("@TenNL", nl.TenNL);  DBConnection.AddParameters("@SoLuongTonKho", nl.SoLuongTonKho);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Sua\_NguyenLieu", DBConnection.parameters);  } |

**10.3. Xóa nguyên liệu**

|  |
| --- |
| CREATE PROCEDURE proc\_Xoa\_NguyenLieu  @MaNL NVARCHAR(50)  AS  BEGIN  DELETE FROM NguyenLieu  WHERE MaNL = @MaNL  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string id)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNL", id);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Xoa\_NguyenLieu", DBConnection.parameters);  } |

**10.4. Tìm kiếm theo tên nguyên liệu**

|  |
| --- |
| CREATE FUNCTION func\_TimKiemTheoTen\_NguyenLieu  (  @TenNguyenLieu NVARCHAR(50)  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM NguyenLieu  WHERE TenNL LIKE N'%' + @TenNguyenLieu + '%'  ) |

Thực thi trên C#

|  |
| --- |
| public static DataTable TimKiemTheoTenNguyenLieu(string ten)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenLoaiSP", ten);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_TimKiemTheoTen\_NguyenLieu", DBConnection.parameters);  } |

**11. Quản lý Loại sản phẩm**

**11.1. Thêm loại sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Them\_LoaiSanPham  @MaLoaiSP NVARCHAR(50),  @TenLoaiSP NVARCHAR(50)  AS  BEGIN  INSERT INTO LoaiSanPham (MaLoaiSP, TenLoaiSP)  VALUES (@MaLoaiSP, @TenLoaiSP)  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(LoaiSanPham lsp)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaLoaiSP", lsp.MaLoaiSP);  DBConnection.AddParameters("@TenLoaiSP", lsp.TenLoaiSP);    DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Them\_LoaiSanPham", DBConnection.parameters);  } |

**11.2. Sửa loại sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Sua\_LoaiSanPham  @MaLoaiSP NVARCHAR(50),  @TenLoaiSP NVARCHAR(50)  AS  BEGIN  UPDATE LoaiSanPham  SET TenLoaiSP = @TenLoaiSP  WHERE MaLoaiSP = @MaLoaiSP  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(LoaiSanPham lsp)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaLoaiSP", lsp.MaLoaiSP);  DBConnection.AddParameters("@TenLoaiSP", lsp.TenLoaiSP);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Sua\_LoaiSanPham", DBConnection.parameters);  } |

**11.3. Xóa loại sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Xoa\_LoaiSanPham  @MaLoaiSP NVARCHAR(50)  AS  BEGIN  DELETE FROM LoaiSanPham  WHERE MaLoaiSP = @MaLoaiSP  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string id)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaLoaiSP", id);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Xoa\_LoaiSanPham", DBConnection.parameters);  } |

**11.4. Tìm kiếm loại sản phẩm theo tên**

|  |
| --- |
| CREATE FUNCTION func\_TimKiemTheoTen\_LoaiSanPham(@TenLoaiSP NVARCHAR(50))  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM LoaiSanPham  WHERE TenLoaiSP LIKE N'%' + @TenLoaiSP + '%'  ) |

Thực thi trên C#

|  |
| --- |
| public static DataTable TimKiem(string ten)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenLoaiSP", ten);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_TimKiemTheoTen\_LoaiSanPham", DBConnection.parameters);  } |

**12. Quản lý Sản phẩm**

**12.1. Load danh sách nguyên liệu theo từng sản phẩm**

|  |
| --- |
| CREATE FUNCTION func\_LoadDanhSachNguyenLieuTheoMaSP(@MaSP NVARCHAR(50))  RETURNS TABLE  AS  RETURN  (  SELECT nl.MaNL, nl.TenNL, pc.SoLuong, nl.SoLuongTonKho  FROM NguyenLieu nl  INNER JOIN PhaChe pc ON nl.MaNL = pc.MaNL  INNER JOIN SanPham sp ON pc.MaSP = sp.MaSP  WHERE sp.MaSP = @MaSP  ) |

Thực thi trên C#

|  |
| --- |
| public static DataTable LoadDSNguyenLieuTheoMaSanPham(string maSP)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaSP", maSP);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_LoadDanhSachNguyenLieuTheoMaSP", DBConnection.parameters);  } |

**12.2. Thêm sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Them\_SanPham  @MaSP NVARCHAR(50),  @TenSP NVARCHAR(50),  @Gia FLOAT,  @AnhSP NVARCHAR(50),  @MaLoaiSP NVARCHAR(50)  AS  BEGIN  INSERT INTO SanPham(MaSP, TenSP, Gia, AnhSP, MaLoaiSP)  VALUES (@MaSP, @TenSP, @Gia, @AnhSP, @MaLoaiSP)  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(SanPham sp)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaSP", sp.MaSP);  DBConnection.AddParameters("@TenSP", sp.TenSP);  DBConnection.AddParameters("@Gia", sp.Gia);  DBConnection.AddParameters("@AnhSP", sp.AnhSP);  DBConnection.AddParameters("MaLoaiSP", sp.MaLoaiSP);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Them\_SanPham", DBConnection.parameters);  } |

**12.3. Sửa sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Sua\_SanPham  @MaSP NVARCHAR(50),  @TenSP NVARCHAR(50),  @Gia FLOAT,  @AnhSP NVARCHAR(50),  @MaLoaiSP NVARCHAR(50)  AS  BEGIN  UPDATE SanPham  SET TenSP = @TenSP,  Gia = @Gia,  AnhSP = @AnhSP,  MaLoaiSP = @MaLoaiSP  WHERE MaSP = @MaSP  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(SanPham sp)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaSP", sp.MaSP);  DBConnection.AddParameters("@TenSP", sp.TenSP);  DBConnection.AddParameters("@Gia", sp.Gia);  DBConnection.AddParameters("@AnhSP", sp.AnhSP);  DBConnection.AddParameters("MaLoaiSP", sp.MaLoaiSP);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Sua\_SanPham", DBConnection.parameters);  } |

**12.4. Xóa sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Xoa\_SanPham  @MaSP NVARCHAR(50)  AS  BEGIN  DELETE FROM SanPham  WHERE MaSP = @MaSP  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string id)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaSP", id);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Xoa\_SanPham", DBConnection.parameters);  } |

**12.5. Tìm kiếm sản phẩm theo tên**

|  |
| --- |
| CREATE FUNCTION func\_TimKiemTheoTen\_SanPham(@TenSP NVARCHAR(50))  RETURNS @KetQua TABLE  (  MaSP NVARCHAR(50),  TenSP NVARCHAR(50),  Gia FLOAT,  AnhSP NVARCHAR(50)  )  AS  BEGIN  INSERT INTO @KetQua  SELECT MaSP, TenSP, Gia, AnhSP  FROM SanPham  WHERE TenSP LIKE N'%' + @TenSP + '%'  RETURN  END |

Thực thi trên C#

|  |
| --- |
| public static DataTable TimKiem(string ten)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenSP", ten);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_TimKiemTheoTen\_SanPham", DBConnection.parameters);  } |

**12.6. Thêm nguyên liệu để pha chế sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Them\_PhaChe  @MaSP NVARCHAR(50),  @MaNL NVARCHAR(50),  @SoLuong INT  AS  BEGIN  INSERT INTO PhaChe(MaSP, MaNL, SoLuong)  VALUES (@MaSP, @MaNL, @SoLuong)  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(PhaChe pc)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaSP", pc.MaSP);  DBConnection.AddParameters("@MaNL", pc.MaNL);  DBConnection.AddParameters("@SoLuong", pc.SoLuong);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Them\_PhaChe", DBConnection.parameters);  } |

**12.7. Xóa nguyên liệu dùng để pha chế sản phẩm**

|  |
| --- |
| CREATE PROCEDURE proc\_Xoa\_PhaChe  @MaSP NVARCHAR(50),  @MaNL NVARCHAR(50)  AS  BEGIN  DELETE FROM PhaChe  WHERE MaSP = @MaSP AND MaNL = @MaNL  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string maSP, string maNL)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaSP", maSP);  DBConnection.AddParameters("@MaNL", maNL);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Xoa\_PhaChe", DBConnection.parameters);  } |

**13. Quản lý Nhân viên**

**13.1. Thêm nhân viên**

|  |
| --- |
| CREATE PROCEDURE proc\_ThemNhanVien  @HoTenNV nvarchar(50),  @SoDienThoai nvarchar(50),  @NamSinh int,  @GioiTinh nvarchar(50),  @DiaChi nvarchar(200),  @Email nvarchar(50),  @TenDangNhap nvarchar(50),  @MatKhau nvarchar(50),  @LoaiNhanVien nvarchar(20)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO NhanVien (HoTenNV, SoDienThoai, NamSinh, GioiTinh, DiaChi, Email,TenDangNhap, MatKhau)  VALUES (@HoTenNV, @SoDienThoai, @NamSinh, @GioiTinh, @DiaChi, @Email, @TenDangNhap, @MatKhau);  DECLARE @mNV nvarchar(50);  SELECT @mNV = MaNV  FROM NhanVien  WHERE SoDienThoai = @SoDienThoai;  IF @LoaiNhanVien = N'Toàn thời gian'  BEGIN  INSERT INTO NhanVienToanThoiGian (MaNV, LuongCoDinh)  VALUES (@mNV, 0);  END  ELSE IF @LoaiNhanVien = N'Bán thời gian'  BEGIN  INSERT INTO NhanVienBanThoiGian (MaNV, LuongTheoGio, SoGio)  VALUES (@mNV, 0, 0);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

Thực thi trên C#

|  |
| --- |
| public static void Them(NhanVien lnv, string loaiNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@HoTenNV", lnv.HoTenNV);  DBConnection.AddParameters("@SoDienThoai", lnv.SoDienThoai);  DBConnection.AddParameters("@NamSinh", lnv.NamSinh);  DBConnection.AddParameters("@GioiTinh", lnv.GioiTinh);  DBConnection.AddParameters("@DiaChi", lnv.DiaChi);  DBConnection.AddParameters("@Email", lnv.Email);  DBConnection.AddParameters("@TenDangNhap", lnv.TenDangNhap);  DBConnection.AddParameters("@MatKhau", lnv.MatKhau);  DBConnection.AddParameters("@LoaiNhanVien", loaiNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_ThemNhanVien", DBConnection.parameters);  } |

**13.2. Sửa nhân viên toàn thời gian**

|  |
| --- |
| CREATE PROCEDURE proc\_SuaNhanVienToanThoiGian  @MaNV NVARCHAR(50),  @LuongCoDinh FLOAT  AS  BEGIN  UPDATE NhanVienToanThoiGian  SET LuongCoDinh = @LuongCoDinh  WHERE MaNV = @MaNV  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(NhanVienToanThoiGian lnvToanTG)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", lnvToanTG.MaNV);  DBConnection.AddParameters("@LuongCoDinh", lnvToanTG.LuongCoDinh);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_SuaNhanVienToanThoiGian", DBConnection.parameters);  } |

**13.3. Sửa nhân viên bán thời gian**

|  |
| --- |
| CREATE PROCEDURE proc\_SuaNhanVienBanThoiGian  @MaNV NVARCHAR(50),  @SoGio INT,  @LuongTheoGio FLOAT  AS  BEGIN  UPDATE NhanVienBanThoiGian  SET SoGio = @SoGio,  LuongTheoGio = @LuongTheoGio  WHERE MaNV = @MaNV  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(NhanVienBanThoiGian lnvBanTG)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", lnvBanTG.MaNV);  DBConnection.AddParameters("@SoGio", lnvBanTG.SoGio);  DBConnection.AddParameters("@LuongTheoGio", lnvBanTG.LuongTheoGio);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_SuaNhanVienBanThoiGian", DBConnection.parameters);  } |

**13.4. Sửa nhân viên**

|  |
| --- |
| CREATE PROCEDURE proc\_SuaNhanVien  @MaNV NVARCHAR(50),  @HoTenNV nvarchar(50),  @SoDienThoai nvarchar(50),  @NamSinh int,  @GioiTinh nvarchar(50),  @DiaChi nvarchar(200),  @Email nvarchar(50),  @TenDangNhap nvarchar(50),  @MatKhau nvarchar(50)  AS  BEGIN  UPDATE NhanVien  SET HoTenNV = @HoTenNV,  SoDienThoai = @SoDienThoai,  NamSinh = @NamSinh,  GioiTinh = @GioiTinh,  DiaChi = @DiaChi,  Email = @Email,  TenDangNhap = @TenDangNhap,  MatKhau = @MatKhau  WHERE MaNV = @MaNV  END |

Thực thi trên C#

|  |
| --- |
| public static void Sua(NhanVien lnv)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", lnv.MaNV);  DBConnection.AddParameters("@HoTenNV", lnv.HoTenNV);  DBConnection.AddParameters("@SoDienThoai", lnv.SoDienThoai);  DBConnection.AddParameters("@NamSinh", lnv.NamSinh);  DBConnection.AddParameters("@GioiTinh", lnv.GioiTinh);  DBConnection.AddParameters("@DiaChi", lnv.DiaChi);  DBConnection.AddParameters("@Email", lnv.Email);  DBConnection.AddParameters("@TenDangNhap", lnv.TenDangNhap);  DBConnection.AddParameters("@MatKhau", lnv.MatKhau);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_SuaNhanVien", DBConnection.parameters);  } |

**13.5. Xóa nhân viên toàn thời gian**

|  |
| --- |
| CREATE PROCEDURE proc\_XoaNhanVienToanThoiGian  @MaNV nvarchar(50)  AS  BEGIN  DELETE FROM NhanVienToanThoiGian  WHERE MaNV = @MaNV  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string maNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_XoaNhanVienToanThoiGian", DBConnection.parameters);  } |

**13.6. Xóa nhân viên toàn thời gian**

|  |
| --- |
| CREATE PROCEDURE proc\_XoaNhanVienBanThoiGian  @MaNV nvarchar(50)  AS  BEGIN  DELETE FROM NhanVienBanThoiGian  WHERE MaNV = @MaNV  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string maNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_XoaNhanVienBanThoiGian", DBConnection.parameters);  } |

**13.7. Xóa nhân viên**

|  |
| --- |
| CREATE PROCEDURE proc\_XoaNhanVien  @MaNV nvarchar(50)  AS  BEGIN  SET NOCOUNT ON;  DECLARE @username nvarchar(50);  SELECT @username = NhanVien.TenDangNhap FROM NhanVien WHERE NhanVien.MaNV = @MaNV;  DECLARE @sqlString nvarchar(50);  BEGIN TRANSACTION  BEGIN TRY  SET @sqlString = 'DROP USER '+ @username;  EXEC(@sqlString)  --  SET @sqlString = 'DROP LOGIN '+ @username;  EXEC(@sqlString)    DELETE FROM NhanVien WHERE NhanVien.MaNV = @MaNV;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string maNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_XoaNhanVien", DBConnection.parameters);  } |

**13.8. Xóa nhân viên khỏi bảng Thực hiện**

|  |
| --- |
| CREATE PROCEDURE proc\_XoaThucHienCongViec  @MaNV nvarchar(50)  AS  BEGIN  DELETE FROM ThucHien  WHERE MaNV = @MaNV  END |

Thực thi trên C#

|  |
| --- |
| public static void Xoa(string maNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_XoaThucHienCongViec", DBConnection.parameters);  } |

**13.9. Tìm kiếm nhân viên theo tên**

|  |
| --- |
| CREATE FUNCTION func\_timKiemTrenViewNhanVien (@HoTenNV NVARCHAR(50))  RETURNS @KetQua TABLE (  MaNV NVARCHAR(50),  HoTenNV NVARCHAR(50),  SoDienThoai NVARCHAR(50),  NamSinh INT,  GioiTinh NVARCHAR(50),  DiaChi NVARCHAR(200),  Email NVARCHAR(50),  TenDangNhap NVARCHAR(50),  MatKhau NVARCHAR(50),  LuongCoDinh FLOAT,  SoGio INT,  LuongTheoGio FLOAT  )  AS  BEGIN  INSERT INTO @KetQua  SELECT MaNV, HoTenNV, SoDienThoai, NamSinh, GioiTinh, DiaChi, Email, TenDangNhap, MatKhau, LuongCoDinh, SoGio, LuongTheoGio  FROM v\_DanhSachNhanVien  WHERE HoTenNV LIKE '%' + @HoTenNV + '%';  RETURN;  END; |

Thực thi trên C#

|  |
| --- |
| public static DataTable timKiemNhanVienTheoTen(string ten)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@HoTenNV", ten);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_timKiemTrenViewNhanVien", DBConnection.parameters);  } |

**13.10. Tính lương nhân viên**

|  |
| --- |
| CREATE FUNCTION func\_TinhLuong(@MaNV nvarchar(50), @Thang int, @Nam int)  RETURNS FLOAT  AS  BEGIN  DECLARE @Luong FLOAT;  DECLARE @LuongCoDinh FLOAT;  DECLARE @LuongTheoGio FLOAT;  DECLARE @SoGio INT;  DECLARE @TongGiaTriDon FLOAT;  DECLARE @Thuong FLOAT = 0;  -- Kiểm tra xem nhân viên là toàn thời gian hay bán thời gian  IF EXISTS (SELECT \* FROM NhanVienToanThoiGian WHERE MaNV = @MaNV)  BEGIN  -- Tính lương cho nhân viên toàn thời gian  SELECT @LuongCoDinh = LuongCoDinh FROM NhanVienToanThoiGian WHERE MaNV = @MaNV;  SET @Luong = @LuongCoDinh;  END  ELSE IF EXISTS (SELECT \* FROM NhanVienBanThoiGian WHERE MaNV = @MaNV)  BEGIN  -- Tính lương cho nhân viên bán thời gian  SELECT @LuongTheoGio = LuongTheoGio, @SoGio = SoGio  FROM NhanVienBanThoiGian  WHERE MaNV = @MaNV;  SET @Luong = @LuongTheoGio \* @SoGio;  END  -- Kiểm tra số ca làm việc của nhân viên  IF EXISTS (  SELECT 1  FROM ThanhToan  INNER JOIN DonHang ON ThanhToan.MaDH = DonHang.MaDH  WHERE ThanhToan.MaNV = @MaNV AND DonHang.GiaTriDon > 200000 AND MONTH(DonHang.NgayMua) = @Thang AND YEAR(DonHang.NgayMua) = @Nam  GROUP BY MaNV  HAVING COUNT(DonHang.MaDH) > 5 -- Ví dụ: thưởng nếu làm trong tháng đó có xuất được 5 đơn trị giá >200k  )  BEGIN  SELECT @TongGiaTriDon = SUM(GiaTriDon) – Tìền thưởng tính bằng 1% tổng giá trị đơn hàng mà nhân viên đó bán trong 1 tháng vừa rồi  FROM DonHang INNER JOIN ThanhToan ON DonHang.MaDH = ThanhToan.MaDH WHERE ThanhToan.MaNV = @MaNV  AND MONTH(DonHang.NgayMua) = @Thang  AND YEAR(DonHang.NgayMua) = @Nam;  -- Nếu không có đơn hàng nào thì tổng giá trị đơn sẽ là 0  SET @TongGiaTriDon = ISNULL(@TongGiaTriDon, 0);  SET @Thuong = @TongGiaTriDon \* 0.01;  END  -- Tính lương cuối cùng bao gồm cả thưởng  SET @Luong = @Luong + @Thuong;  RETURN @Luong;  END |

Thực thi trên C#

|  |
| --- |
| public static string tinhLuongNV(string maNV, int thang, int nam)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.AddParameters("@Thang", thang);  DBConnection.AddParameters("@Nam", nam);  return DBConnection.ThucThiFunction\_Scalar("func\_TinhLuong", DBConnection.parameters).ToString();  } |

**13.11. Gửi email thông báo lương tháng cho nhân viên.**

|  |
| --- |
| CREATE PROCEDURE proc\_guiThongBaoLuongThang  @MaNV NVARCHAR(50),  @Thang INT,  @Nam INT,  @LuongNV FLOAT  AS  BEGIN  DECLARE @HoTen nvarchar(50);  DECLARE @Email nvarchar(50);  DECLARE @Luong FLOAT;  DECLARE @LuongFormatted nvarchar(50);  DECLARE @NoiDung nvarchar(MAX);  DECLARE @TieuDe nvarchar(MAX);  -- Lấy thông tin nhân viên và lương  SELECT @HoTen = HoTenNV, @Email = Email FROM NhanVien WHERE MaNV = @MaNV;  SET @Luong = @LuongNV;  SET @LuongFormatted = FORMAT(@Luong, 'N0', 'vi-VN');  -- Nội dung email  SET @NoiDung =  N'Chào ' + @HoTen + N',<br/><br/>' +  N'Chúng tôi xin thông báo lương tháng ' + CAST(@Thang AS nvarchar) + N' năm ' + CAST(@Nam AS nvarchar) + N' của bạn là: ' +  @LuongFormatted + N'.<br/><br/>' +  N'Cảm ơn bạn đã làm việc chăm chỉ!<br/>' +  N'Thân ái,<br/>Quản lý';  -- Tiêu đề  SET @TieuDe = N'Thông báo lương tháng ' + CAST(@Thang AS nvarchar) + N'/' + CAST(@Nam AS nvarchar);  -- Gửi email  EXEC msdb.dbo.sp\_send\_dbmail  @profile\_name = 'MailProfile', -- Tên profile cấu hình trong Database Mail  @recipients = @Email,  @subject = @TieuDe,  @body = @NoiDung,  @body\_format = 'HTML';  END |

Thực thi trên C#

|  |
| --- |
| public static void guiMail(string maNV, int thang, int nam, double luong)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.AddParameters("@Thang", thang);  DBConnection.AddParameters("@Nam", nam);  DBConnection.AddParameters("@LuongNV", luong);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_guiThongBaoLuongThang",DBConnection.parameters);  } |

**14. Quản lý Phân ca**

**14.1. Tìm nhân viên thực hiện ca làm việc dựa trên tên ca và ngày cụ thể.**

|  |
| --- |
| CREATE PROCEDURE Proc\_TimNguoiThucHienCa  @shiftCa NVARCHAR(50),  @currentDate NVARCHAR(10)  AS  BEGIN  SELECT HoTenNV  FROM CaLamViec CLV  JOIN ThucHien TH ON CLV.MaCa = TH.MaCa  JOIN NhanVien NV ON TH.MaNV = NV.MaNV  WHERE TenCa = @shiftCa  AND CONVERT(NVARCHAR, Ngay, 23) = @currentDate;  END |

Thực thi trên C#

|  |
| --- |
| public List<string> TimNguoiThucHien(string shiftName, string currentDate)  {  List<string> employeeNames = new List<string>(); // Tạo danh sách để lưu tên nhân viên  DBConnection.ClearParameters();  DBConnection.AddParameters("@shiftCa", shiftName);  DBConnection.AddParameters("@currentDate", currentDate);  SqlDataReader layTen = null;  try  {  // Gọi hàm với tham số traVeReader = true để trả về SqlDataReader  layTen = DBConnection.ThucThiProc\_CoReader("Proc\_TimNguoiThucHienCa", DBConnection.parameters, true);  if (layTen != null)  {  while (layTen.Read()) // Lặp qua từng kết quả  {  employeeNames.Add(layTen["HoTenNV"].ToString()); // Thêm từng tên nhân viên vào danh sách  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  layTen?.Close(); // Đảm bảo đóng reader sau khi sử dụng  }  return employeeNames; // Trả về danh sách nhân viên  } |

**14.2. Tạo và trả về mã ca mới theo định dạng CAxx dựa trên mã ca lớn nhất hiện có trong bảng CaLamViec**

|  |
| --- |
| CREATE FUNCTION func\_LayMaCaMoiDeDangKy\_CaLamViec()  RETURNS NVARCHAR(50)  AS  BEGIN  DECLARE @maxMaCa NVARCHAR(50);  DECLARE @newMaCa NVARCHAR(50);  DECLARE @numPart INT;    SELECT @maxMaCa = MAX(MaCa)  FROM CaLamViec  WHERE MaCa LIKE 'CA%';    IF @maxMaCa IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaCa, 3, LEN(@maxMaCa) - 2) AS INT) + 1;  END  ELSE  BEGIN  SET @numPart = 1;  END  -- Tạo giá trị mới cho MaCa, với định dạng CAxx (2 số)  SET @newMaCa = 'CA' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);  RETURN @newMaCa;  END; |

Thực thi trên C#

|  |
| --- |
| public static String LayMaCaMoiDeDangKy()  {  DBConnection.ClearParameters();  object maCa = DBConnection.ThucThiFunction\_Scalar("func\_LayMaCaMoiDeDangKy\_CaLamViec", DBConnection.parameters);    if( maCa == null || maCa == DBNull.Value)  {  MessageBox.Show("Không lấy được mã ca để đăng ký","Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return string.Empty;  }  return maCa.ToString();  } |

**14.3. Thêm ca làm việc**

|  |
| --- |
| CREATE PROCEDURE Proc\_ThemCaLamViec\_CaLamViec  @MaCa NVARCHAR(50),  @TenCa NVARCHAR(50),  @Ngay NVARCHAR(10),  @GioBatDau TIME,  @GioKetThuc TIME  AS  BEGIN  INSERT INTO CaLamViec (MaCa, TenCa, Ngay, GioBatDau, GioKetThuc)  VALUES (@MaCa, @TenCa, @Ngay, @GioBatDau, @GioKetThuc);  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(CaLamViec clv)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaCa", clv.MaCa);  DBConnection.AddParameters("@TenCa", clv.TenCa);  DBConnection.AddParameters("@Ngay", clv.Ngay);  DBConnection.AddParameters("@GioBatDau", clv.GioBatDau);  DBConnection.AddParameters("@GioKetThuc", clv.GioKetThuc);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("Proc\_ThemCaLamViec\_CaLamViec", DBConnection.parameters);  } |

**14.4. Thêm thực hiện**

|  |
| --- |
| CREATE PROCEDURE Proc\_ThemKhoaChinhThucHien\_ThucHien  @MaCa NVARCHAR(50),  @MaNV NVARCHAR(50),  @MaCV NVARCHAR(50)  AS  BEGIN  INSERT INTO ThucHien (MaCa, MaNV, MaCV)  VALUES (@MaCa, @MaNV, @MaCV);  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(ThucHien th)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@MaNV", th.MaNV);  DBConnection.AddParameters("@MaCa", th.MaCa);  DBConnection.AddParameters("@MaCV", th.MaCV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("Proc\_ThemKhoaChinhThucHien\_ThucHien", DBConnection.parameters);  } |

**15. Thống kê**

**15.1. Tính tổng doanh thu tất cả đơn hàng**

|  |
| --- |
| CREATE FUNCTION func\_TongDoanhThu\_ChiTietHoaDon  (  )  RETURNS FLOAT  AS  BEGIN  DECLARE @TongDoanhThu FLOAT = 0  SELECT @TongDoanhThu = SUM(TongTien)  FROM ChiTietHoaDon  RETURN @TongDoanhThu  END; |

Thực thi trên C#

|  |
| --- |
| public static double TongDoanhThuTatCaDonHang()  {  DBConnection.ClearParameters();  object tongDoanhThu = DBConnection.ThucThiFunction\_Scalar("func\_TongDoanhThu\_ChiTietHoaDon", DBConnection.parameters);  if (tongDoanhThu is DBNull || tongDoanhThu == null)  return 0;  return Convert.ToDouble(tongDoanhThu);  } |

**15.2. Tính tổng số lượng sản phẩm đã bán**

|  |
| --- |
| CREATE FUNCTION func\_TongSoLuongSanPham\_ChiTietHoaDon  (  )  RETURNS INT  AS  BEGIN  DECLARE @TongSoLuong INT = 0  SELECT @TongSoLuong = SUM(SoLuong)  FROM ChiTietHoaDon  RETURN @TongSoLuong  END; |

Thực thi trên C#

|  |
| --- |
| public static double TongSoLuongSPDaBan()  {  DBConnection.ClearParameters();  object tongSLSPDaBan = DBConnection.ThucThiFunction\_Scalar("func\_TongSoLuongSanPham\_ChiTietHoaDon", DBConnection.parameters);  if (tongSLSPDaBan is DBNull || tongSLSPDaBan == null)  return 0;  return Convert.ToDouble(tongSLSPDaBan);  } |

**15.3. Tính tổng số lượng khách hàng**

|  |
| --- |
| CREATE FUNCTION func\_TongSoLuongKhachHang\_KhachHang  (  )  RETURNS INT  AS  BEGIN  DECLARE @SoLuongKH INT = 0  SELECT @SoLuongKH = COUNT(MaKH)  FROM KhachHang  RETURN @SoLuongKH  END; |

Thực thi trên C#

|  |
| --- |
| public static double TongSoLuongKH()  {  DBConnection.ClearParameters();  object tongSLKH = DBConnection.ThucThiFunction\_Scalar("func\_TongSoLuongKhachHang\_KhachHang", DBConnection.parameters);  if (tongSLKH is DBNull || tongSLKH == null)  return 0;  return Convert.ToDouble(tongSLKH);  } |

**16. Trang chủ**

**16.1. Sửa giá trị đơn hàng**

|  |
| --- |
| CREATE PROCEDURE proc\_CapNhatGiaTriDon\_DonHang  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DECLARE @MaDH NVARCHAR(50)  SELECT @MaDH = MaDH  FROM DonHang  WHERE TrangThai = N'Chưa thanh toán'    UPDATE DonHang  SET GiaTriDon = (SELECT SUM(TongTien) FROM ChiTietHoaDon WHERE MaDH = @MaDH)  WHERE TrangThai = N'Chưa thanh toán'  COMMIT TRANSACTION  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END |

Thực thi trên C#

|  |
| --- |
| public static void CapNhatGiaTriDonHangChuaThanhToan()  {  DBConnection.ClearParameters();  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_CapNhatGiaTriDon\_DonHang", DBConnection.parameters);  } |

**16.2. Tìm kiếm sản phẩm theo tên**

|  |
| --- |
| CREATE FUNCTION func\_TimKiemTheoTen\_NguyenLieu  (  @TenNguyenLieu NVARCHAR(50)  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM NguyenLieu  WHERE TenNL LIKE N'%' + @TenNguyenLieu + '%'  ) |

Thực thi trên C#

|  |
| --- |
| public static DataTable TimKiem(string ten)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@TenSP", ten);  return DBConnection.ThucThiFunction\_InlineVaMultiStatement("func\_TimKiemTheoTen\_SanPham", DBConnection.parameters);  } |

**16.3.** **Thêm khách hàng mới với mã khách hàng tự động tăng**

|  |
| --- |
| CREATE PROCEDURE proc\_Them\_KhachHang  @SoDienThoai NVARCHAR(50)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DECLARE @maxMaKH NVARCHAR(50);  DECLARE @newMaKH NVARCHAR(50);  DECLARE @numPart INT;    IF EXISTS (SELECT 1 FROM KhachHang WHERE SoDienThoai = @SoDienThoai)  BEGIN  COMMIT TRANSACTION -- Bắt buột commit hoặc rollback trước khi return  RETURN;  END    SELECT @maxMaKH = MAX(MaKH)  FROM KhachHang  WHERE MaKH LIKE 'KH%';    IF @maxMaKH IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaKH, 3, LEN(@maxMaKH) - 2) AS INT) + 1;  END  ELSE  BEGIN  SET @numPart = 1;  END  -- Tạo giá trị mới cho MaKH, với định dạng KHxx (2 số)  SET @newMaKH = 'KH' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);    INSERT INTO KhachHang(MaKH, TenKH, SoDienThoai, SoDiemTichLuy)  VALUES(@newMaKH, N'Chưa nhập tên', @SoDienThoai, 0)  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH    END |

Thực thi trên C#

|  |
| --- |
| public static void Them(string SoDienThoai)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@SoDienThoai", SoDienThoai);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Them\_KhachHang", DBConnection.parameters);  } |

**16.4. Thêm đơn hàng mới với mã đơn hàng tự động tăng**

|  |
| --- |
| CREATE PROCEDURE proc\_Them\_DonHang  @SoDienThoai NVARCHAR(50),  @MaNV NVARCHAR(50)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DECLARE @MaKH NVARCHAR(50);  DECLARE @maxMaDH NVARCHAR(50);  DECLARE @newMaDH NVARCHAR(50);  DECLARE @numPart INT;    SELECT @MaKH = MaKH  FROM KhachHang  WHERE SoDienThoai = @SoDienThoai    SELECT @maxMaDH = MAX(maDH)  FROM DonHang  WHERE maDH LIKE 'DH%';    IF @maxMaDH IS NOT NULL  BEGIN  SET @numPart = CAST(SUBSTRING(@maxMaDH, 3, LEN(@maxMaDH) - 2) AS INT) + 1;  END  ELSE  BEGIN  SET @numPart = 1;  END  -- Tạo giá trị mới cho maDH, với định dạng DHxx (2 số)  SET @newMaDH = 'DH' + RIGHT('00' + CAST(@numPart AS NVARCHAR), 2);    INSERT INTO DonHang(MaDH, NgayMua, GiaTriDon, TrangThai)  VALUES (@newMaDH, GETDATE(), 1, N'Chưa thanh toán')    INSERT INTO ThanhToan(MaDH, MaNV, MaKH)  VALUES (@newMaDH, @MaNV, @MaKH)  COMMIT TRANSACTION  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END |

Thực thi trên C#

|  |
| --- |
| public static void Them(string soDienThoai, string maNV)  {  DBConnection.ClearParameters();  DBConnection.AddParameters("@SoDienThoai", soDienThoai);  DBConnection.AddParameters("@MaNV", maNV);  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Them\_DonHang", DBConnection.parameters);    } |

**16.5. Xóa đơn hàng chưa thanh toán và khách hàng chưa nhập tên nếu khách hàng chỉ có một hoặc không có đơn hàng nào khác**

|  |
| --- |
| CREATE PROCEDURE proc\_Xoa\_KhachHangChuaNhapTen\_DonHangChuaThanhToan  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DECLARE @MaDH NVARCHAR(50),  @MaNV NVARCHAR(50),  @MaKH NVARCHAR(50),  @DonHangCount INT;    SELECT @MaDH = dh.MaDH, @MaKH = kh.MaKH, @MaNV = nv.MaNV  FROM DonHang dh  INNER JOIN ThanhToan tt ON dh.MaDH = tt.MaDH  INNER JOIN NhanVien nv ON tt.MaNV = nv.MaNV  INNER JOIN KhachHang kh ON tt.MaKH = kh.MaKH  WHERE TrangThai = N'Chưa thanh toán'  -- Đếm số đơn hàng còn lại của khách hàng trong bảng DonHang  SELECT @DonHangCount = COUNT(\*)  FROM DonHang dh  INNER JOIN ThanhToan tt ON dh.MaDH = tt.MaDH  INNER JOIN KhachHang kh ON tt.MaKH = kh.MaKH  WHERE kh.MaKH = @MaKH;  DELETE FROM ThanhToan  WHERE MaDH = @MaDH AND MaNV = @MaNV AND MaKH = @MaKH    DELETE FROM ChiTietHoaDon  WHERE MaDH = @MaDH    DELETE FROM DonHang  WHERE TrangThai = N'Chưa thanh toán'    --Nếu KH chỉ có 1 đơn (đơn chưa thanh toán) hoặc 0 có đơn nào thì mới được xóa  IF @DonHangCount <= 1  BEGIN  DELETE FROM KhachHang  WHERE MaKH = @MaKH;  END  COMMIT TRANSACTION  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END |

Thực thi trên C#

|  |
| --- |
| public static void XoaDonChuaThanhToan()  {  DBConnection.ClearParameters();  DBConnection.ThucThiProc\_CoThamSoVaKhongCoThamSo("proc\_Xoa\_KhachHangChuaNhapTen\_DonHangChuaThanhToan", DBConnection.parameters);  } |

**16.6. Tính tổng tiền chi tiết hóa đơn**

|  |
| --- |
| CREATE FUNCTION func\_TongTien\_ChiTietHoaDon  (  )  RETURNS FLOAT  AS  BEGIN  DECLARE @MaDH NVARCHAR(50)  DECLARE @Tong FLOAT = 0  SELECT @MaDH = MaDH  FROM DonHang  WHERE TrangThai = N'Chưa thanh toán'    SELECT @Tong = SUM(TongTien)  FROM ChiTietHoaDon  WHERE MaDH = @MaDH  RETURN @Tong  END; |

Thực thi trên C#

|  |
| --- |
| public static double TinhTongTien()  {  DBConnection.ClearParameters();  object tongTien = DBConnection.ThucThiFunction\_Scalar("func\_TongTien\_ChiTietHoaDon", DBConnection.parameters);  if (tongTien is DBNull || tongTien == null)  return 0;  return Convert.ToDouble(tongTien);  } |

**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

* Ứng với hai nhóm người dung (Nhân viên và quản lý) là các role riêng
* Đối với Role Staff (dành cho nhân viên):

|  |
| --- |
| CREATE ROLE Staff  -- Gán các quyền trên table cho role Staff  GRANT SELECT, INSERT, DELETE,REFERENCES ON KhachHang TO Staff  GRANT SELECT, INSERT, DELETE,REFERENCES ON DonHang TO Staff  GRANT SELECT, INSERT,REFERENCES ON ChiTietHoaDon TO Staff  GRANT SELECT, INSERT,REFERENCES ON PhaChe TO Staff  GRANT SELECT, REFERENCES ON LoaiSanPham TO Staff  GRANT SELECT, REFERENCES ON NguyenLieu TO Staff  GRANT SELECT, REFERENCES ON NhanVien TO Staff  GRANT SELECT, REFERENCES ON NhanVienBanThoiGian TO Staff  GRANT SELECT, REFERENCES ON NhanVienToanThoiGian TO Staff  GRANT SELECT, REFERENCES ON SanPham TO Staff  GRANT SELECT, REFERENCES ON ThucHien TO Staff  GRANT SELECT, REFERENCES ON CongViec TO Staff  GRANT SELECT, REFERENCES ON CaLamViec TO Staff  GRANT EXECUTE TO Staff -- Cấp quyền thực thi cho "Staff" trên các func và proc trong CSDL  GRANT SELECT TO Staff -- Cấp quyền truy vấn dữ liệu cho "Staff" trên các bảng trong CSDL  DENY EXECUTE ON proc\_ThemNhanVien to Staff;  DENY EXECUTE ON proc\_XoaNhanVien to Staff;  DENY EXECUTE ON proc\_XoaNhanVienBanThoiGian to Staff;  DENY EXECUTE ON proc\_XoaNhanVienToanThoiGian to Staff;  DENY EXECUTE ON proc\_SuaNhanVien to Staff;  DENY EXECUTE ON proc\_SuaNhanVienBanThoiGian to Staff;  DENY EXECUTE ON proc\_SuaNhanVienToanThoiGian to Staff;  DENY EXECUTE ON proc\_Xoa\_SanPham to Staff;  DENY EXECUTE ON proc\_Sua\_SanPham to Staff;  DENY EXECUTE ON proc\_Sua\_LoaiSanPham to Staff;  DENY EXECUTE ON proc\_Them\_LoaiSanPham to Staff;  DENY EXECUTE ON proc\_Sua\_NguyenLieu to Staff;  DENY EXECUTE ON proc\_Xoa\_NguyenLieu to Staff;  DENY EXECUTE ON proc\_XoaThucHienCongViec to Staff;  DENY EXECUTE ON proc\_guiThongBaoLuongThang to Staff;  DENY EXECUTE ON func\_TinhLuong to Staff;  ------------------------- Tạo Login và User cho nhân viên (tương ứng dữ liệu đã insert phía trên. Trong đó nv03 là admin)---------------  GO  CREATE LOGIN [nv01] WITH PASSWORD='nv01', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv01] FOR LOGIN [nv01];  GO  EXEC sp\_addrolemember 'Staff', 'nv01'  GO  GO  CREATE LOGIN [nv02] WITH PASSWORD='nv02', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv02] FOR LOGIN [nv02];  GO  EXEC sp\_addrolemember 'Staff', 'nv02'  GO  CREATE LOGIN [nv04] WITH PASSWORD='nv04', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv04] FOR LOGIN [nv04];  GO  EXEC sp\_addrolemember 'Staff', 'nv04'  GO  GO  CREATE LOGIN [nv05] WITH PASSWORD='nv05', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv05] FOR LOGIN [nv05];  GO  EXEC sp\_addrolemember 'Staff', 'nv05'  GO  GO  CREATE LOGIN [nv06] WITH PASSWORD='nv06', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv06] FOR LOGIN [nv06];  GO  EXEC sp\_addrolemember 'Staff', 'nv06'  GO  GO  CREATE LOGIN [nv07] WITH PASSWORD='nv07', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv07] FOR LOGIN [nv07];  GO  EXEC sp\_addrolemember 'Staff', 'nv07'  GO  GO  CREATE LOGIN [nv08] WITH PASSWORD='nv08', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv08] FOR LOGIN [nv08];  GO  EXEC sp\_addrolemember 'Staff', 'nv08'  GO |

+ Đối với Role Manager ta sử dụng server role sysadmin

|  |
| --- |
| CREATE LOGIN [nv03] WITH PASSWORD='nv03', DEFAULT\_DATABASE= QuanLyQuanCaPhe, CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;  GO  CREATE USER [nv03] FOR LOGIN [nv03];  GO  EXEC sp\_addsrvrolemember 'nv03', 'sysadmin'; |

* Trigger tạo tài khoản

|  |
| --- |
| CREATE TRIGGER trg\_taoTaiKhoan ON NhanVien  AFTER INSERT  AS  BEGIN  DECLARE @username nvarchar(50), @password nvarchar(50), @manv nvarchar(50)  DECLARE @sqlString nvarchar(2000), @macv nvarchar(50)  SELECT @username=i.TenDangNhap, @password=i.MatKhau, @manv=i.maNV  FROM inserted i;  SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+ @passWord +''', DEFAULT\_DATABASE=[QuanLyQuanCaPhe],  CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'  EXEC (@sqlString)  ----  SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName  EXEC (@sqlString)  ----  SET @sqlString = 'ALTER ROLE Staff ADD MEMBER ' + @userName;  EXEC (@sqlString)  END |

* Procedure xóa nhân viên

|  |
| --- |
| CREATE PROCEDURE proc\_XoaNhanVien  @MaNV nvarchar(50)  AS  BEGIN  SET NOCOUNT ON;  DECLARE @username nvarchar(50);  SELECT @username = NhanVien.TenDangNhap FROM NhanVien WHERE NhanVien.MaNV = @MaNV;  DECLARE @sqlString nvarchar(50);  BEGIN TRANSACTION  BEGIN TRY  SET @sqlString = 'DROP USER '+ @username;  EXEC(@sqlString)  --  SET @sqlString = 'DROP LOGIN '+ @username;  EXEC(@sqlString)    DELETE FROM NhanVien WHERE NhanVien.MaNV = @MaNV;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END |