**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL**

**TÊN ĐỀ TÀI:**

**QUẢN LÝ CỬA HÀNG LINH KIỆN MÁY TÍNH**

**GVHD:** TS.Nguyễn Thành Sơn

**Lớp HP:** 241DBMS330284\_05

**Nhóm thực hiện:** Nhóm 6

**Học kỳ:** 1

**Năm học:** 2024-2025

**Tp. Hồ Chí Minh, tháng năm 2024**

**DANH SÁCH SINH VIÊN THỰC HIỆN**

**NHÓM 6**

|  |  |
| --- | --- |
| **MSSV** | **Họ và tên** |
| 22110455 | Nguyễn Mạnh Tú (Nhóm trưởng) |
| 22110393 | Hà Đức Phát |
| 22110333 | Nguyễn Sang Huy |
| 22110438 | Nguyễn Thị Thuỳ Trang |

**NHẬN XÉT CỦA GV**

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

Giảng viên chấm điểm

**MỤC LỤC**

[**I-ĐẶC TẢ BÀI TOÁN** 7](#_Toc181879412)

[**1. Đặc tả** 7](#_Toc181879413)

[**2. Chức năng** 8](#_Toc181879414)

[**3. Giao diện** 8](#_Toc181879415)

[**II-THIẾT KẾ CƠ SỞ DỮ LIỆU** 10](#_Toc181879416)

[**1.Thiết kế cơ sở dữ liệu mức quan niệm** 10](#_Toc181879417)

[**2. Thiết kế cơ sở dữ liệu mức logic** 11](#_Toc181879418)

[**3. Các ràng buộc cần có** 12](#_Toc181879419)

[**4.Cài đặt các CSDL và các ràng buộc** 15](#_Toc181879420)

[**5.Các trigger** 19](#_Toc181879421)

[**6. Các View** 22](#_Toc181879422)

[**III. CÀI ĐẶT CÁC CHỨC NĂNG** 25](#_Toc181879423)

[**1. Kết nối CSDL (SQL Server)** 25](#_Toc181879424)

[**2. Cài đặt chức năng** 26](#_Toc181879425)

[**2.1. Chức năng tìm kiếm linh kiện theo tên** 26](#_Toc181879426)

[**2.2. Chức năng tìm kiếm theo bộ có sẵn** 26](#_Toc181879427)

[**2.3. Chức năng lưu thông tin khách hàng** 27](#_Toc181879428)

[**2.4. Chức năng xuất hóa đơn cho khách hàng** 28](#_Toc181879429)

[**2.5. Chức năng lưu các linh kiện của 1 hóa đơn** 29](#_Toc181879430)

[**2.6. Chức năng hiển thị danh sách hóa đơn** 30](#_Toc181879431)

[**2.7. Chức năng tìm hóa đơn theo tên khách hàng** 30](#_Toc181879432)

[**2.8. Chức năng tìm hóa đơn theo giá trị hóa đơn** 31](#_Toc181879433)

[**2.9. Chức năng tìm hóa đơn theo ngày xuất hóa đơn** 32](#_Toc181879434)

[**2.10. Chức năng hiển thị chi tiết hóa đơn** 32](#_Toc181879435)

[**2.11. Chức năng hiển thị Ca làm việc để chấm công** 33](#_Toc181879436)

[**2.12. Chức năng Chấm công** 34](#_Toc181879437)

[**2.13. Chức năng load danh sách ca làm đã thực hiện của nhân viên** 35](#_Toc181879438)

[**2.14. Chức năng xem doanh thu theo ca** 37](#_Toc181879439)

[**2.15. Chức năng xem doanh thu theo từng ca làm việc trong khoảng thời gian được chọn** 38](#_Toc181879440)

[**2.16. Chức năng tìm kiếm nhân viên** 40](#_Toc181879441)

[**2.17 Chức năng quản lý nhân viên** 40](#_Toc181879442)

[**2.17.1 Chức năng thêm tài khoản nhân viên** 40](#_Toc181879443)

[**2.17.2 Chức năng xoá tài khoản nhân viên** 42](#_Toc181879444)

[**2.17.3 Chức năng sửa tài khoản nhân viên** 43](#_Toc181879445)

[**2.18. Tính doanh thu của cửa hàng trong khoảng thời gian** 45](#_Toc181879446)

[**2.19. Tính doanh thu và lợi nhuận của cửa hàng theo tháng** 46](#_Toc181879447)

[**2.20 Chức năng quản lý sản phẩm** 47](#_Toc181879448)

[**2.20.1 Chức năng thêm sản phẩm** 47](#_Toc181879449)

[**2.20.2 Chức năng sửa thông tin sản phẩm** 49](#_Toc181879450)

[**2.21 Chức năng tìm kiếm linh kiện theo tên linh kiện** 51](#_Toc181879451)

[**2.22 Chức năng tìm kiếm linh kiện theo giá** 52](#_Toc181879452)

[**2.23 Chức năng tìm kiếm linh kiện theo nhóm linh kiện** 53](#_Toc181879453)

[**2.24 Chức năng thêm đơn nhập hàng** 54](#_Toc181879454)

[**2.25 Chức năng xoá đơn nhập hàng** 56](#_Toc181879455)

[**2.26 Chức năng thêm linh kiện vào đơn nhập hàng** 57](#_Toc181879456)

[**2.27 Chức năng xoá linh kiện khỏi đơn nhập hàng** 58](#_Toc181879457)

[**2.28 Chức năng sửa linh kiện trong đơn nhập hàng** 60](#_Toc181879458)

[**2.29 Chức năng thêm nhóm linh kiện vào kho** 61](#_Toc181879459)

[**2.30 Chức năng sửa nhóm linh kiện trong kho** 62](#_Toc181879460)

[**2.31 Chức năng thêm nhà cung cấp** 63](#_Toc181879461)

[**2.32 Chức năng sửa nhà cung cấp** 64](#_Toc181879462)

[**2.33 Chức năng tìm kiếm nhà cung cấp theo tên nhà cung cấp** 66](#_Toc181879463)

[**2.34 Chức năng thêm ca làm việc của nhân viên** 67](#_Toc181879464)

[**2.35 Chức năng thêm nhân viên vào ca làm việc** 68](#_Toc181879465)

[**2.36 Chức năng xoá nhân viên khỏi ca làm việc** 69](#_Toc181879466)

[**2.37 Hàm thực hiện truy vấn** 70](#_Toc181879467)

[**IV- TẠO USER VÀ PHÂN QUYỀN** 72](#_Toc181879468)

**LỜI CẢM ƠN**

Nhóm em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Thành Sơn vì thầy đã tận tình hướng dẫn, hỗ trợ và truyền đạt kiến thức cho chúng em trong suốt quá trình học tập và thực hiện đồ án này. Nhờ có sự chỉ dẫn tận tình của thầy, nhóm em đã có cái nhìn hoàn thiện hơn về hệ quản trị cơ sở dữ liệu. Qua đó, nhóm em có thể hoàn thành đồ án một cách tốt nhất.

Tuy nhiên, thiếu sót là không thể tránh khỏi nên nhóm em rất mong có thể nhận được sự góp ý và đánh giá của thầy để có thể hoàn thiện hơn nữa.

Nhóm em xin chân thành cảm ơn thầy!

# **I-ĐẶC TẢ BÀI TOÁN**

## **1. Đặc tả**

Trong thời đại công nghệ ngày càng phát triển như hiện nay, nhu cầu về các mặt hàng linh kiện máy tính ngày càng tăng. Tuy nhiên cùng với cơ hội phát triển các cửa hàng kinh doanh sản phẩm linh kiện máy tính phải đối mặt với sự cạnh tranh khá cao, có nhiều thách thức trong quản lý hàng hoá và chăm sóc khách hàng. Một trong các yếu tố giúp cho cửa hàng phát triển là khả năng quản lý một cách hiệu quả. Vì thế, việc xây dựng và phát triển một hệ thống quản lý cửa hàng linh kiện máy tính là vô cùng cần thiết. Hệ thống quản lý này cần lưu trữ các dữ liệu sau:

Cửa hàng bao gồm nhiều nhân viên cùng làm việc, thông tin của mỗi nhân viên gồm mã nhân viên, họ tên, ngày sinh, giới tính, số điện thoại, địa chỉ, ngày bắt đầu làm việc, lương. Mỗi nhân viên phân biệt với nhau bằng một mã nhân viên duy nhất.

Nhân viên tại cửa hàng sẽ làm việc theo ca, mỗi nhân viên có thể làm nhiều ca khác nhau. Giữa ca làm việc và nhân viên sẽ có trạng thái nhân viên điểm danh ca làm. Mỗi ca làm việc có thể có nhiều nhân viên làm việc cùng nhau. Mỗi ca làm việc sẽ có mã ca, tên ca khác nhau, ngày, tháng làm việc của ca đó, thời gian bắt đầu, thời gian kết thúc ca làm việc và các nhân viên làm việc trong ca đó.

Mỗi khách hàng khi đến mua hàng tại cửa hàng sẽ được lưu lại thông tin bao gồm họ tên, số điện thoại, email, địa chỉ và được phân biệt bằng một mã khách hàng duy nhất. Thông tin này dùng để lưu lại lịch sử mua hàng và bảo hành cho khách hàng. Khi khách hàng mua sản phẩm, cửa hàng sẽ xuất hóa đơn cho lần mua hàng đó. Thông tin của hóa đơn bao gồm một mã số hóa đơn duy nhất, ngày xuất hóa đơn, tổng giá trị hóa đơn, mã khách hàng, mã nhân viên, mã và tên các sản phẩm, đơn giá và số lượng mỗi sản phẩm.

Các linh kiện của cửa hàng được chia thành các nhóm linh kiện khác nhau (ví dụ: RAM, SSD, main,...). Mỗi nhóm linh kiện gồm mã nhóm linh kiện, tên nhóm linh kiện và số lượng loại sản phẩm trong nhóm đó. Mỗi nhóm linh kiện có nhiều linh kiện khác nhau. Mỗi linh kiện gồm một mã linh kiện duy nhất, tên linh kiện, một mô tả ngắn gọn về thông số của linh kiện đó, hình ảnh của linh kiện, giá tiền của linh kiện, trạng thái sản phẩm (còn hàng, hết hàng, ngừng kinh doanh), số lượng còn lại trong kho; ngoài ra còn có hình ảnh về linh kiện đó để tăng tính trực quan.

Các linh kiện tại cửa hàng được cung cấp bởi nhiều nhà cung cấp khác nhau. Mỗi nhà cung cấp bao gồm một mã nhà cung cấp duy nhất, tên nhà cung cấp, số điện thoại liên lạc và địa chỉ của nhà cung cấp. Khi cửa hàng nhập hàng từ nhà cung cấp sẽ xuất một đơn nhập hàng. Đơn nhập hàng gồm các thông tin: mã nhập đơn, mã nhà cung cấp, ngày nhập đơn, giá trị đơn nhập. Đơn nhập chứa danh sách các linh kiện trong lần nhập hàng đó và số lượng mỗi loại, đơn giá, tổng tiền mỗi loại.

**2. Chức năng**

Về chức năng, chủ cửa hàng và nhân viên sẽ có những chức năng đặc trưng phù hợp với từng vai trò trong cửa hàng.

Đối với nhân viên

Nhân viên có thể **tra cứu linh kiện,** nhân viên có thể tìm các linh kiện theo các bộ lọc khác nhau(tên, tên nhóm, giá, trạng thái)

Nhân viên có thể **tạo đơn hàng,** nhân viên thêm linh kiện vào đơn hàng và xóa, sửa sản phẩm của đơn hàng, điền thông tin khách hàng và xuất hóa đơn thanh toán và nhân viên cũng có thể **tra cứu đơn hàng** tìm lại và xem chi tiết các đơn hàng đã bán

**Chấm công,** nhân viên điểm danh ca làm của mình.

Ngoài ra, nhân viên có thể xem ca làm việc và doanh thu trong ca làm việc của mình.

Đối với chủ cửa hàng

Chủ cửa hàng sẽ **quản lý nhân viên** của mình bao gồm xem, nhập thêm, xóa, sửa thông tin của nhân viên và có thể tìm kiếm nhân viên.

Chủ cửa hàng cũng sẽ **quản lý ca làm việc** của cửa hàng bao gồm việc thêm các ca làm việc và thêm hoặc xóa nhân viên trong ca làm.

Về **quản lý sản phẩm**, chủ cửa hàng có thể xem chi tiết, tìm kiếm, thêm, sửa các linh kiện, nhóm linh kiện hoặc tra cứu các mặt hàng trong kho.

Chủ cửa hàng còn có thể **thống kê doanh thu** của cửa hàng theo ngày, tháng, năm, tạo ra hoặc xóa, sửa các hóa đơn nhập cho cửa hàng.

Chủ cửa hàng có thể thêm, sửa các **nhà cung cấp** cho cửa hàng.

**3. Giao diện**

Giao diện phần mềm quản lý cửa hàng linh kiện máy tính được phân chia theo hai đối tượng người dùng: nhân viên và chủ cửa hàng.

Đối với nhân viên, giao diện sẽ hiển thị tên và mã nhân viên của nhân viên đó, ca làm việc ngày hôm đó và bao gồm các mục cơ bản như linh kiện, xử lý đơn hàng, lịch sử đơn hàng, chấm công, doanh thu. Cửa sổ “**Linh kiện**” sẽ hiển thị các sản phẩm mà cửa hàng đang kinh doanh, có ô “**nhập tên linh kiện**” để tìm tên linh kiện theo tên, có các bộ lọc “**Nhóm**”, “**Giá**”, “**Trạng thái**”, nhân viên bấm nút “**Tìm kiếm**” để tìm theo các bộ lọc này. Khi bấm vào linh kiện, thông tin chi tiết của linh kiện hiển thị gồm ảnh sản phẩm, mã linh kiện, tên linh kiện, nhóm linh kiện, số lượng sản phẩm và đơn giá. Muốn thêm sản phẩm vào đơn hàng, nhân viên bấm “**Thêm vào đơn hàng**”.

Ở cửa sổ “**Xử lý đơn hàng**” nhân viên sẽ xem được danh sách các linh kiện đã thêm vào đơn hàng, bấm vào linh kiện có thể xem được ảnh sản phẩm, tên linh kiện, đơn giá, số lượng (có thể thay đổi được) và tổng tiền, bấm “**cập nhật**” để cập nhật lại số lượng của 1 linh kiện, có thể xóa 1 linh kiện trong hóa đơn bằng cách bấm “**Xóa**”; xuất hóa đơn cho khách hàng trước tiên phải nhập đầy đủ các thông tin họ tên khách hàng, số điện thoại, email, địa chỉ và bấm “**xuất hóa đơn**”, nếu thành công sẽ hiện thông báo “**Xuất hóa đơn thành công**”.

Trong cửa sổ “**Lịch sử đơn hàng**” sẽ bao gồm “**danh sách hóa đơn**” được thực hiện bởi nhân viên trong ca làm đó, các ô tìm kiếm hóa đơn theo tên khách hàng, giá trị hóa đơn và ngày xuất hóa đơn. Khi nhân viên bấm vào 1 hóa đơn trong danh sách sẽ hiển thị chi tiết từng món trong hóa đơn đó trong danh sách “**chi tiết hóa đơn**”

Cửa sổ “**Chấm công**” cho phép nhân viên vào ca khi bắt đầu ca làm và ra ca khi kết thúc ca làm. Ngoài ra cửa sổ “**Doanh thu ca**” sẽ hiện tổng quan về doanh thu và số lượng sản phẩm bán ra trong ca làm của nhân viên.

Cửa sổ **“Chấm công”** cho phép nhân viên khi bắt đầu ca làm sẽ thực hiện chấm công ca làm việc được phân. Giao diện sẽ hiển thị ra thông tin của một ca làm gần nhất của nhân viên đã được chủ của hàng phân công. Nhân viên phải thực hiện chấm công để bắt đầu ca làm.

Cửa sổ **“Doanh thu ca”** sẽ hiển thị tất cả ca làm việc của nhân viên đã thực hiện. Nhân viên chịn vào một ca bất kỳ sẽ xem được chi tiết doanh thu của ca đó. Những thông tin được hiển thị về chi tiết doanh thu của ca bao gồm số linh kiện đã bán được, số hóa đơn được thực hiện, tổng doanh thu của ca. Ngoài ra, nhân viên còn có thể xem doanh thu của mình trong một khoảng thời gian khi thực hiện việc tìm theo thời gian.

Đối với chủ cửa hàng, giao diện gồm các mục như quản lý sản phẩm, quản lý nhập hàng, kho linh kiện, nhà cung cấp, quản lý nhân viên, bảng phân ca, doanh thu.

**Quản lý sản phẩm** gồm danh sách các linh kiện hiện tại của cửa hàng , bấm vào linh kiện bất kỳ để xem chi tiết và chỉnh sửa linh kiện đó, bấm nút “**Thêm”** sẽ hiện ra các mục để điền thông tin sản phẩm và bấm nút **“Thêm”** để thêm một linh kiên mới, có thể tìm kiếm linh kiếm qua bộ tìm kiếm ( tên, giá(<=), tên nhóm).

**Quản lý nhập hàng** hiện thông tin các đơn nhập hàng, bấm “**Thêm đơn nhập”** sẽ hiện ra các mục điền thông tin của đơn nhập(Mã nhà cung cấp, ngày nhập) và bấm “**Thêm đơn nhập”**  để thêm đơn nhập linh kiện, có thể bấm vào các đơn nhập để hiện các sản phẩm trong đơn nhập, bấm vào **“Xóa đơn nhập”** để xóa đơn nhập đó, bấm vào “**Thêm linh kiện”** điền thông tin(tên, số lượng, giá nhập) rồi bấm **“Thêm”** để thêm linh kiện cho đơn nhậpđể thêm linh kiện. Có thể bấm vào linh kiện bất kỳ trong danh sách đơn kiện để xem chi tiết hoặc chỉnh sửa thông tin nhập của linh kiện đó.

**Kho linh kiện** hiện danh sách các nhóm linh kiện, có thể bấm chuột vào nhóm linh kiện và chỉnh sửa lại thông tin rồi bấm nút **“Sửa”** hoặc điền thông tin của nhóm linh kiện mới và bấm nút thêm **“Thêm”**. Bấm đúp vào nhóm linh kiện bất kỳ để xem các sản phẩm trong nhóm linh kiện đó. Có thể tìm kiếm tên nhóm linh kiện bằng mục tìm kiếm.

**Nhà cung cấp** hiện ra danh sách nhà cung cấp của cửa hàng, tìm kiếm nhà cung cấp bằng tên ở mục tìm kiếm, bấm vào nhà cung cấp bất kỳ và sửa lại thông tin của nhà cung cấp rồi bấm “**Sửa”** hoặc điền thông tin của nhà cung cấp mới và bấm “**Thêm”**.

**Quản lý nhân viên** hiện ra danh sách nhân viên của cửa hàng, có thể tìm kiếm nhân viên bằng mã nhân viên ở mục tìm kiếm, bấm vào nhân viên bất kỳ rồi sửa lại thông tin nhân viên và bấm **“Sửa”** hoặc bấm **“Xóa”** để xóa nhân viên đó hoặc có thể điền thông tin nhân viên mới và bấm **“Thêm”.**

**Bảng phân ca** hiện ra các ca làm theo ngày , chọn ngày để xem ca làm việc của ngày đó , bấm vào ngày đang trống và bấm “**Tạo ca làm việc”** để thêm ca làm việc, bấm vào ca làm việc bất kỳ để xem nhân viên của ca làm việc, bấm **“Thêm nhân viên”** để thêm nhân viên vào ca làm việc đó.

**Doanh thu** chọn tháng, năm bên mục ngày tháng để tính doanh thu và lợi nhuận của tháng đó, hoặc có thể tình doanh thu giữa 2 ngày , tháng, năm.

# **II-THIẾT KẾ CƠ SỞ DỮ LIỆU**

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

Từ đặc tả bài toán ta có ERD như sau:

A diagram of a company

Description automatically generated

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

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

- NhanVien(MaNV, HoTen, NgaySinh, GioiTinh, SDT, DiaChi, NgayBDLamViec, Luong)

- CaLamViec(MaCa, TenCa, Ngay, ThoiGianBD, ThoiGianKT)

- PhanCa(MaNhanVien, MaCa, trangthai)

- LinhKien(MaLK, TenLK, GiaTien, MoTa, TrangThai, SoLuong, AnhLK, MaNhom, MaNCC)

- NhomLinhKien( MaNhom, TenNhom, SoLuongLK)

- KhachHang(MaKH, HoTen, SDT, Email, DiaChi)

- HoaDon(MaHD, NgayXuat, TongGiaTri, MaKH, MaNV)

- ChiTietHD(MaLK, MaHD, SoLuong, DonGia, TongTien)

- NhaCungCap(MaNCC, TenNCC, DiaChi, SDT)

- DonNhapHang(MaDonNhap, NgayNhap, MaNCC, GiaTriDonNhap)

- NhapHang(MaLK, MaDonNhap, SoLuong, DonGia, TongTien)

## **3. Các ràng buộc cần có**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên bảng** | **Thuộc tính và kiểu dữ liệu** | **Ràng buộc** |
| 1 | NhanVien | MaNV : nvarchar (30) | Khoá chính |
| HoTen : nvarchar (50) | NOT NULL |
| SDT : char (11) | NOT NULL  CHECK SDT(10 số) |
| NgayBDLamViec : date | NOT NULL  CHECK NgayBDLamViec(>0) |
| NgaySinh : date | CHECK NgaySinh(>=18) |
| GioiTinh : nvarchar (3) |  |
| DiaChi : nvarchar (100) |  |
| Luong : float | Check Luong >= 0 |
| 2 | CaLamViec | MaCa : nvarchar (30) | Khoá chính |
| TenCa : nvarchar (10) |  |
| Ngay : date | NOT NULL |
| ThoiGianBD : time (0) | NOT NULL |
| ThoiGianKT : time (0) | NOT NULL |
|  | CHECK (ThoiGianBD < ThoiGianKT) |
| 3 | PhanCa | MaNhanVien : nvarchar (30)  MaCa : nvarchar (30)  TrangThai : nvarchar (30) | Khóa chính (MaNhanVien, MaCa)  Khóa ngoại MaNhanVien đến NhanVien  Khóa ngoại MaCa đến CaLamViec |
| 4 | LinhKien | MaLK : nvarchar (30) | MaLK |
| TenLK: nvarchar (50)  GiaTien : float  MoTa : nvarchar (500)  TrangThai : nvarchar (20) |  |
| SoLuong : int | CHECK SoLuong(>=0) |
| MaNhom : nvarchar (30)  MaNCC : nvarchar (30)  AnhLK : varbinary (max) |  |
|  | Khóa ngoại MaNhom đến NhomLinhKien  Khóa ngoại MaNCC đến NhaCungCap |
| 5 | NhomLinhKien | MaNhom : nvarchar (30) | Khoá chính |
| TenNhom : nvarchar (50) | NOT NULL |
| SoLuongLK : int | CHECK SoLuongLK (>=0) |
| 6 | KhachHang | MaKH : nvarchar (30) | Khoá chính |
| HoTen : nvarchar (50) | NOT NULL |
| SDT : char (11) | NOT NULL  CHECK SDT( 10 số) |
| Email : nvarchar (50) |  |
| DiaChi : nvarchar (50) | NOT NULL |
| 7 | HoaDon | MaHD : nvarchar (30) | Khoá chính |
| NgayXuat : date | NOT NULL  CHECK NgayXuat(>=0) |
| TongGiaTri : float | NOT NULL |
| MaKH : nvarchar (30)  MaNV : nvarchar (30) |  |
|  | Khóa ngoại MaKH đến KhachHang  Khóa ngoại MaNV đến NhanVien |
| 8 | ChiTietHD | MaLK : nvarchar (30) |  |
| MaHD : nvarchar (30) |  |
| SoLuong : int | CHECK SoLuong(>0) |
| DonGia : float | NOT NULL |
| TongTien : float | NOT NULL |
|  | Khóa chính (MaLK,MaHD)  Khóa ngoại MaLK đến LinhKien  Khóa ngoại MaHD đến HoaDon |
| 9 | NhaCungCap | MaNCC : nvarchar (30) | Khoá chính |
| SDT : char (11) | NOT NULL  CHECK SDT( 10 số) |
| TenNhaCungCap : nvarchar (50) | NOT NULL |
| DiaChi : nvarchar (50) | NOT NULL |
| 10 | DonNhapHang | MaDonNhap : nvarchar (30) | Khoá chính |
| NgayNhap : date | NOT NULL  CHECK NgayNhap >= 0 |
| MaNCC : nvarchar (30) | NOT NULL |
| GiaTriDonNhap : float | NOT NULL  CHECK GiaTriDonNhap > 0 |
|  | Khóa ngoại MaNCC đến NhaCungCap |
| 11 | NhapHang | MaLK: nvarchar (30) |  |
| MaDonNhap : nvarchar (30) |  |
| SoLuong : int | CHECK SoLuong > 0 |
| DonGia : float | CHECK DonGia > 0 |
| TongTien : float | CHECK TongTien > 0 |
|  | Khóa chính (MaLK,MaDonNhap)  Khóa ngoại MaLK đến LinhKien  Khóa ngoại MaDonNhap đến DonNhapHang |

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

***Bảng nhân viên***

|  |
| --- |
| CREATE TABLE NhanVien (  MaNV NVARCHAR(30) CONSTRAINT PK\_MaNV PRIMARY KEY,  HoTen NVARCHAR(50) NOT NULL,  SDT CHAR(11) CHECK (LEN(SDT) = 10) NOT NULL,  NgayBDLamViec DATE CHECK (DATEDIFF(day, NgayBDLamViec, GETDATE()) >= 0) NOT NULL,  NgaySinh DATE CHECK (DATEDIFF(year, NgaySinh, GETDATE()) >= 18),  GioiTinh NVARCHAR(3),  DiaChi NVARCHAR(100),  Luong FLOAT CHECK (Luong >= 0) NOT NULL  ); |

***Bảng ca làm việc***

|  |
| --- |
| CREATE TABLE CaLamViec (  MaCa NVARCHAR(30) CONSTRAINT PK\_MaCa PRIMARY KEY,  TenCa NVARCHAR(10) NOT NULL,  Ngay DATE NOT NULL,  ThoiGianBD TIME(0) NOT NULL,  ThoiGianKT TIME(0) NOT NULL,  CONSTRAINT CHK\_ThoiGian CHECK (ThoiGianBD < ThoiGianKT)  ); |

***Bảng phân ca***

|  |
| --- |
| CREATE TABLE PhanCa (  MaNhanVien NVARCHAR(30) CONSTRAINT FK\_PhanCa\_MaNhanVien FOREIGN KEY REFERENCES NhanVien(MaNV),  MaCa NVARCHAR(30) CONSTRAINT FK\_PhanCa\_MaCa FOREIGN KEY REFERENCES CaLamViec(MaCa),  CONSTRAINT PK\_PhanCa PRIMARY KEY (MaNhanVien, MaCa),  trangthai nvarchar(30)  ); |
|  |

***Bảng nhóm linh kiện***

|  |
| --- |
| CREATE TABLE NhomLinhKien (  MaNhom NVARCHAR(30) CONSTRAINT PK\_MaNhom PRIMARY KEY,  TenNhom NVARCHAR(50) NOT NULL,  SoLuongLK INT CHECK (SoLuongLK >= 0) NOT NULL  ); |

***Bảng nhà cung cấp***

|  |
| --- |
| CREATE TABLE NhaCungCap (  MaNCC NVARCHAR(30) CONSTRAINT PK\_MaNCC PRIMARY KEY,  SDT CHAR(11) CHECK (LEN(SDT) = 10) NOT NULL,  TenNhaCungCap NVARCHAR(50) NOT NULL,  DiaChi NVARCHAR(50) NOT NULL  ); |

***Bảng linh kiện***

|  |
| --- |
| CREATE TABLE LinhKien (  MaLK NVARCHAR(30) CONSTRAINT PK\_MaLK PRIMARY KEY,  TenLK NVARCHAR(50) NOT NULL,  GiaTien FLOAT CHECK (GiaTien > 0) NOT NULL,  MoTa NVARCHAR(500) NOT NULL,  TrangThai NVARCHAR(20) NOT NULL,  SoLuong INT CHECK (SoLuong >= 0) NOT NULL,  MaNhom NVARCHAR(30) CONSTRAINT FK\_LinhKien\_MaNhom FOREIGN KEY REFERENCES NhomLinhKien(MaNhom),  MaNCC NVARCHAR(30) CONSTRAINT FK\_LinhKien\_MaNCC FOREIGN KEY REFERENCES NhaCungCap(MaNCC),  AnhLK VARBINARY(MAX)  ); |

***Bảng khách hàng***

|  |
| --- |
| CREATE TABLE KhachHang (  MaKH NVARCHAR(30) CONSTRAINT PK\_MaKH PRIMARY KEY,  HoTen NVARCHAR(50) NOT NULL,  SDT CHAR(11) CHECK (LEN(SDT) = 10) NOT NULL,  Email NVARCHAR(50),  DiaChi NVARCHAR(50) NOT NULL  ); |

***Bảng hoá đơn***

|  |
| --- |
| CREATE TABLE HoaDon (  MaHD NVARCHAR(30) CONSTRAINT PK\_MaHD PRIMARY KEY,  NgayXuat DATE CHECK (DATEDIFF(day, NgayXuat, GETDATE()) >= 0) NOT NULL,  TongGiaTri FLOAT CHECK (TongGiaTri > 0) NOT NULL,  MaKH NVARCHAR(30) CONSTRAINT FK\_HoaDon\_MaKH FOREIGN KEY REFERENCES KhachHang(MaKH),  MaNV NVARCHAR(30) CONSTRAINT FK\_HoaDon\_MaNV FOREIGN KEY REFERENCES NhanVien(MaNV)  ); |

***Bảng chi tiết hoá đơn***

|  |
| --- |
| CREATE TABLE ChiTietHD (  MaLK NVARCHAR(30) CONSTRAINT FK\_ChiTietHD\_MaLK FOREIGN KEY REFERENCES LinhKien(MaLK),  MaHD NVARCHAR(30) CONSTRAINT FK\_ChiTietHD\_MaHD FOREIGN KEY REFERENCES HoaDon(MaHD),  SoLuong INT CHECK (SoLuong > 0) NOT NULL,  DonGia FLOAT CHECK (DonGia > 0) NOT NULL,  TongTien FLOAT CHECK (TongTien > 0) NOT NULL,  CONSTRAINT PK\_ChiTietHD PRIMARY KEY (MaLK, MaHD)  ); |

***Bảng đơn nhập hàng***

|  |
| --- |
| CREATE TABLE DonNhapHang (  MaDonNhap NVARCHAR(30) CONSTRAINT PK\_MaDonNhap PRIMARY KEY,  NgayNhap DATE CHECK (DATEDIFF(day, NgayNhap, GETDATE()) >= 0) NOT NULL,  MaNCC NVARCHAR(30) CONSTRAINT FK\_DonNhapHang\_MaNCC FOREIGN KEY REFERENCES NhaCungCap(MaNCC) NOT NULL,  GiaTriDonNhap FLOAT CHECK (GiaTriDonNhap > 0) NOT NULL  ); |

***Bảng nhập hàng***

|  |
| --- |
| CREATE TABLE NhapHang (  MaLK NVARCHAR(30) CONSTRAINT FK\_NhapHang\_MaLK FOREIGN KEY REFERENCES LinhKien(MaLK),  MaDonNhap NVARCHAR(30) CONSTRAINT FK\_NhapHang\_MaDonNhap FOREIGN KEY REFERENCES DonNhapHang(MaDonNhap),  SoLuong INT CHECK (SoLuong > 0),  DonGia FLOAT CHECK (DonGia > 0),  TongTien FLOAT CHECK (TongTien > 0),  CONSTRAINT PK\_NhapHang PRIMARY KEY (MaLK, MaDonNhap)  ); |

**A screenshot of a computer

Description automatically generated**

## **5.Các trigger**

**a.Trigger tự động tạo mã khách hàng**

Khi insert vào bảng khách hàng mã khách hàng sẽ tự động tạo

|  |
| --- |
| CREATE TRIGGER tg\_genCustomerID  ON KhachHang  INSTEAD OF INSERT  AS  BEGIN  INSERT INTO KhachHang (MaKH, HoTen, SDT, Email, DiaChi)  SELECT 'KH' + i.SDT, i.HoTen, i.SDT, i.Email, i.DiaChi  FROM inserted i;  END;  GO |

**b.Trigger tự động tạo mã ca làm việc**

Khi tạo ca làm việc, mã ca làm việc sẽ được tạo tự động

|  |
| --- |
| CREATE TRIGGER trg\_AutoGenerateMaCa  ON CaLamViec  INSTEAD OF INSERT  AS  BEGIN  INSERT INTO CaLamViec (MaCa, TenCa, Ngay, ThoiGianBD, ThoiGianKT)  SELECT    LEFT(CONCAT('CA', FORMAT(i.Ngay, 'yyMMdd'), SUBSTRING(i.TenCa, 1, 1)), 10) AS MaCa,  i.TenCa,  i.Ngay,  i.ThoiGianBD,  i.ThoiGianKT  FROM inserted i;  END;  GO |

**c. Trigger tự động tạo mã nhà cung cấp**

Khi thêm nhà cung cấp , mã nhà cung được tự động tạo khi nhà cung cấp được thêm vào

|  |
| --- |
| CREATE TRIGGER trg\_genMaNCC  ON NhaCungCap  INSTEAD OF INSERT  AS  BEGIN  DECLARE @MaNCC NVARCHAR(30);  DECLARE @TenNhaCungCap NVARCHAR(50);  DECLARE @SDT NCHAR(10);  SELECT @TenNhaCungCap = TenNhaCungCap, @SDT = SDT  FROM inserted;  SET @MaNCC = LEFT(@TenNhaCungCap, 3) + RIGHT(@SDT, 4);  INSERT INTO NhaCungCap (MaNCC, TenNhaCungCap, SDT, DiaChi)  SELECT @MaNCC, TenNhaCungCap, SDT, DiaChi  FROM inserted;  END;  GO |

**d. Trigger tự động tạo mã đơn nhập khi thêm đơn nhập hàng**

Khi thêm vào bảng nhập hàng, mã nhập hàng sẽ được tạo tự động

|  |
| --- |
| CREATE TRIGGER trg\_genMaDonNhap  ON DonNhapHang  INSTEAD OF INSERT  AS  BEGIN  INSERT INTO DonNhapHang (MaDonNhap, NgayNhap, MaNCC, GiaTriDonNhap)  SELECT RTRIM(i.MaNCC) + '\_' + FORMAT(i.NgayNhap,'yyMMdd'), i.NgayNhap, i.MaNCC, i.GiaTriDonNhap  FROM inserted i;  END;  GO |

**e. Trigger tự động tạo mã hóa đơn**

Khi tạo hóa đơn, mã hóa đơn sẽ được tự động tạo

|  |
| --- |
| CREATE TRIGGER trg\_genMaHD  ON HoaDon  INSTEAD OF INSERT  AS  BEGIN  INSERT INTO HoaDon(MaHD, NgayXuat, TongGiaTri, MaKH, MaNV)  SELECT RTRIM(i.MaKH) + '\_' + RTRIM(i.MaNV) + '\_' + FORMAT(i.NgayXuat,'yyMMdd'), i.NgayXuat, i.TongGiaTri, i.MaKH, i.MaNV  FROM inserted i;  END;  GO |

**f. Trigger Tự Động Cập Nhật Tổng Giá Trị Đơn Nhập**

Khi thêm, xóa, sửa trong bảng NhapHang giá trị đơn nhập sẽ tự cập nhật

|  |
| --- |
| --Trigger Tự Động Cập Nhật Tổng Giá Trị Đơn Nhập khi xóa  CREATE TRIGGER trg\_DeleteNhapHang  ON NhapHang  AFTER DELETE  AS  BEGIN  UPDATE DonNhapHang  SET GiaTriDonNhap = (SELECT ISNULL(SUM(TongTien), 1) FROM NhapHang WHERE MaDonNhap = d.MaDonNhap)  FROM DonNhapHang d  INNER JOIN deleted del ON d.MaDonNhap = del.MaDonNhap;  END;  GO  --Trigger Tự Động Cập Nhật Tổng Giá Trị Đơn Nhập khi thêm, sửa  CREATE TRIGGER trg\_UpdateTongGiaTriDonNhap  ON NhapHang  AFTER INSERT, UPDATE  AS  BEGIN  UPDATE DonNhapHang  SET GiaTriDonNhap = (SELECT SUM(TongTien) FROM NhapHang WHERE MaDonNhap = u.MaDonNhap)  FROM DonNhapHang d  INNER JOIN inserted u ON d.MaDonNhap = u.MaDonNhap;  END  GO |

**g. Trigger cập nhật số lượng linh kiện khi thêm, sửa, xóa dữ liệu vào bảng NhapHang**

Khi thêm, sửa, xóa trong bảng nhập hàng thì số lượng kinh kiện cũng tự động cập nhật theo

|  |
| --- |
| CREATE TRIGGER trg\_UpdateSoLuongLinhKien\_NhapHang  ON NhapHang  AFTER INSERT, UPDATE, DELETE  AS  BEGIN  IF EXISTS (SELECT \* FROM inserted)  BEGIN  UPDATE LinhKien  SET SoLuong += ISNULL(i.SoLuong, 0)  FROM LinhKien lk  INNER JOIN inserted i ON lk.MaLK = i.MaLK;  END  IF EXISTS (SELECT \* FROM deleted)  BEGIN  UPDATE LinhKien  SET SoLuong -= ISNULL(d.SoLuong, 0)  FROM LinhKien lk  INNER JOIN deleted d ON lk.MaLK = d.MaLK;  END  END; |

**h. Trigger cập nhật số lượng linh kiện khi xuất hóa đơn**

Khi xóa hóa đơn số lượng linh kiện cũng được cập nhật lại

|  |
| --- |
| CREATE TRIGGER trg\_UpdateSoLuongLinhKien\_XuatHoaDon  ON ChiTietHD  AFTER INSERT, UPDATE  AS  BEGIN  IF EXISTS (SELECT \* FROM inserted)  BEGIN  UPDATE LinhKien  SET SoLuong -= ISNULL(i.SoLuong, 0)  FROM LinhKien lk  INNER JOIN inserted i ON lk.MaLK = i.MaLK;  END  END;  GO |

## **6. Các View**

**a. View nhà cung cấp**

|  |
| --- |
| CREATE VIEW ViewNhaCungCap AS  SELECT  MaNCC AS 'Mã Nhà Cung Cấp',  TenNhaCungCap AS 'Tên Nhà Cung Cấp',  SDT AS 'Số Điện Thoại',  DiaChi AS 'Địa Chỉ'  FROM  NhaCungCap;  GO |

**b. View linh kiện**

|  |
| --- |
| CREATE VIEW ViewSanPham AS  SELECT  L.MaLK,  L.TenLK,  L.GiaTien,  L.MoTa,  L.TrangThai,  L.SoLuong,  N.TenNhaCungCap,  NH.TenNhom,  L.AnhLK  FROM  LinhKien L  JOIN  NhaCungCap N ON L.MaNCC = N.MaNCC  JOIN  NhomLinhKien NH ON L.MaNhom = NH.MaNhom;  GO |

**c. View đơn nhập hàng**

|  |
| --- |
| CREATE VIEW ViewDonNhapHang AS  SELECT  dn.MaDonNhap,  dn.MaNCC,  dn.NgayNhap,  dn.GiaTriDonNhap,  ncc.TenNhaCungCap,  ncc.SDT  FROM  DonNhapHang dn  INNER JOIN  NhaCungCap ncc ON dn.MaNCC = ncc.MaNCC  GO |

**d. View nhập hàng**

|  |
| --- |
| CREATE VIEW ViewNhapHang AS  SELECT  nh.MaDonNhap,  nh.MaLK,  lk.TenLK,  nh.SoLuong,  nh.DonGia,  nh.TongTien  FROM  NhapHang nh  INNER JOIN  LinhKien lk ON nh.MaLK = lk.MaLK;  GO |

**e. View nhóm linh kiện**

|  |
| --- |
| CREATE VIEW ViewNhomLK AS  SELECT  nh.MaNhom,  nh.TenNhom,  nh.SoLuongLK  FROM  NhomLinhKien nh  GO |

**f. View bảng ca làm việc**

|  |
| --- |
| CREATE VIEW ViewCaLamViec AS  SELECT  MaCa,  TenCa,  Ngay,  ThoiGianBD,  ThoiGianKT  FROM  CaLamViec;  GO |

**g. View bảng phân ca**

|  |
| --- |
| CREATE VIEW ViewPhanCa AS  SELECT  p.MaNhanVien,  n.HoTen AS TenNhanVien,  n.SDT,  p.MaCa,  c.TenCa,  c.Ngay,  c.ThoiGianBD,  c.ThoiGianKT  FROM  PhanCa p  JOIN  NhanVien n ON p.MaNhanVien = n.MaNV  JOIN  CaLamViec c ON p.MaCa = c.MaCa;  GO |

**h. View linh kiện nhân viên**

|  |
| --- |
| CREATE VIEW LinhKienView AS  SELECT  LinhKien.MaLK,  LinhKien.TenLK,  LinhKien.SoLuong,  NhomLinhKien.TenNhom,  LinhKien.TrangThai,  LinhKien.GiaTien,  LinhKien.AnhLK,  LinhKien.MoTa  FROM  LinhKien join NhomLinhKien on LinhKien.MaNhom = NhomLinhKien.MaNhom |

# **III. CÀI ĐẶT CÁC CHỨC NĂNG**

## **1. Kết nối CSDL (SQL Server)**

|  |
| --- |
| internal class ConnectDB  {  private SqlConnection sqlCon;  public ConnectDB(string connectionString)  {  sqlCon = new SqlConnection(connectionString);  }  public SqlConnection GetConnection()  {  return sqlCon;  }  public void Open()  {  try  {  if (sqlCon.State == ConnectionState.Closed)  {  sqlCon.Open();  }  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  }  public void Close()  {  if (sqlCon != null && sqlCon.State == ConnectionState.Open)  {  sqlCon.Close();  }  else  {  MessageBox.Show("NO CONNECTED");  }  } |

## **2. Cài đặt chức năng**

### **2.1. Chức năng tìm kiếm linh kiện theo tên**

**- Input: tên linh kiện**

**- Output: danh sách các linh kiện có tên giống input**

**- Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_timTenLK(@TenLK NVARCHAR(50))  RETURNS TABLE  AS  RETURN(  SELECT \*  FROM LinhKienView  WHERE TenLK LIKE '%' + @TenLK + '%'  ); |

**- Code C#:**

|  |
| --- |
| public static DataTable getProductByName(string tenLK)  {  string query = string.Format("Select \* From fn\_timTenLK(N'{0}')", tenLK);  return ExecuteQuery(query);  } |

### **2.2. Chức năng tìm kiếm theo bộ có sẵn**

**- Input: Tên nhóm, giá, trạng thái**

**- Ouput: danh sách các linh kiện theo các input**

**- Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_GetProductByFilter  (  @TenNhom NVARCHAR(50) = NULL,  @Gia FLOAT = NULL,  @TrangThai NVARCHAR(50) = NULL  )  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM LinhKienView  WHERE ( @TenNhom IS NULL OR TenNhom = @TenNhom )  AND ( @Gia IS NULL OR GiaTien <= @Gia )  AND ( @TrangThai IS NULL OR TrangThai = @TrangThai )  );  GO |

**- Code C#:**

|  |
| --- |
| public static DataTable getProductByFilter(string tenNhom, float gia, string trangThai)  {  DataTable dataTable = new DataTable();  string connectionString = DataConnector.connectionString;  using (SqlConnection connection = new SqlConnection(connectionString))  {  try  {  // Tạo truy vấn gọi hàm như bảng  string query = "SELECT \* FROM fn\_GetProductByFilter(@TenNhom, @Gia, @TrangThai)";  SqlCommand command = new SqlCommand(query, connection);  // Thêm tham số cho hàm  command.Parameters.AddWithValue("@TenNhom", string.IsNullOrEmpty(tenNhom) ? (object)DBNull.Value : tenNhom);  command.Parameters.AddWithValue("@Gia", gia == 0 ? (object)DBNull.Value : gia);  command.Parameters.AddWithValue("@TrangThai", string.IsNullOrEmpty(trangThai) ? (object)DBNull.Value : trangThai);  SqlDataAdapter dataAdapter = new SqlDataAdapter(command);  dataAdapter.Fill(dataTable);  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  }  return dataTable;  } |

### **2.3. Chức năng lưu thông tin khách hàng**

**- Input: họ tên, số điện thoại, email, địa chỉ khách hàng**

**- Ouput: khách hàng được thêm vào bảng KhachHang**

**- Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_InsertKhachHang  @HoTen NVARCHAR(50),  @Phone CHAR(11),  @Email NVARCHAR(50),  @DiaChi NVARCHAR(100)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO KhachHang (HoTen, SDT, Email, DiaChi)  VALUES (@HoTen, @Phone, @Email, @DiaChi);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Thêm khách hàng lỗi!' + ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END; |

**- Code C#:**

|  |
| --- |
| public static void saveInfoKH(string hoTen, string SDT, string email, string diaChi)  {  string query = String.Format("EXEC sp\_InsertKhachHang N'{0}', '{1}', '{2}', N'{3}'", hoTen, SDT, email, diaChi);  ExecutingNonResult(query);  } |

### **2.4. Chức năng xuất hóa đơn cho khách hàng**

**- Input: Ngày xuất, tổng giá trị, mã khách hàng, mã nhân viên**

**- Ouput: Thêm hóa đơn vào bảng HoaDon**

**- Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_insertHoaDon  @NgayXuat DATE,  @TongGiaTri FLOAT,  @MaKH NVARCHAR(30),  @MaNV NVARCHAR(30)  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  INSERT INTO HoaDon(NgayXuat, TongGiaTri, MaKH, MaNV)  VALUES(@NgayXuat, @TongGiaTri, @MaKH, @MaNV);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Xuất hóa đơn lỗi!' + ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END; |

**-Code C#:**

|  |
| --- |
| public static void xuatHoaDon(DateTime dateXuat, float tongGiaTri, string maKH, string maNV)  {  string query = String.Format("EXEC sp\_insertHoaDon '{0}', '{1}', '{2}', '{3}'", DateTime.Now.Date, tongGiaTri, maKH, maNV);  ExecutingNonResult(query);  } |

### **2.5. Chức năng lưu các linh kiện của 1 hóa đơn**

**- Input: mã linh kiện, mã hóa đơn , số lương, đơn giá , tổng tiền**

**- Output: lưu các linh kiện của hóa đơn vào ChitietHD**

**- Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_insertChiTietHD  @MaLK NVARCHAR(30),  @MaHD NVARCHAR(30),  @SoLuong INT,  @DonGia FLOAT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO ChiTietHD(MaLK, MaHD, SoLuong, DonGia, TongTien)  VALUES(@MaLK, @MaHD, @SoLuong, @DonGia, @TongTien);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi khi thêm chi tiết đơn hàng!' + ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END; |

**-Code C#:**

|  |
| --- |
| public static void themChiTietHD(string maLK, string maHD, int soLuong, float donGia, float tongTien)  {  string query = string.Format("EXEC sp\_insertChiTietHD '{0}', '{1}', '{2}', '{3}', '{4}'", maLK, maHD, soLuong, donGia, tongTien);  ExecutingNonResult(query);  } |

### **2.6. Chức năng hiển thị danh sách hóa đơn**

**- Input: Mã nhân viên**

**- Ouput: Danh sách hóa đơn của nhân viên đó**

**- Code SQL:**

|  |
| --- |
| Create Function fn\_GetHoaDonByMaNV (@MaNV nvarchar(30))  Returns Table  As  Return  (  Select  HoaDon.MaKH,  HoaDon.MaHD,  HoaDon.NgayXuat,  HoaDon.TongGiaTri,  KhachHang.HoTen  From HoaDon  Join KhachHang on HoaDon.MaKH = KhachHang.MaKH  Where HoaDon.MaNV = @MaNV  ); |

**- Code C#:**

|  |
| --- |
| public static DataTable getDanhSachHD(string maNV)  {  string query = String.Format("Select \* From fn\_GetHoaDonByMaNV('{0}')", maNV);  return ExecuteQuery(query);  } |

### **2.7. Chức năng tìm hóa đơn theo tên khách hàng**

**- Input: Mã nhân viên, tên khách hàng**

**- Ouput: danh sách các hóa đơn có mã nhân viên và tên khách hàng giống input**

**- Code C#:**

|  |
| --- |
| public static DataTable getDanhSachHDByName(string maNV, string tenKH)  {  string query = String.Format("Select \* From fn\_GetHoaDonByMaNV('{0}') Where HoTen Like N'%{1}%'", maNV, tenKH);  return ExecuteQuery(query);  } |

### **2.8. Chức năng tìm hóa đơn theo giá trị hóa đơn**

**- Input: mã nhân viên, giá trị Min, giá trị Max**

**- Ouput: danh sách các hóa đơn có giá trị trong khoảng min và max**

**- Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_timTheoGiaTri(@GiaTriMin FLOAT, @GiaTriMax FLOAT, @MaNV NVARCHAR(30))  RETURNS TABLE  AS  RETURN(  SELECT  HoaDon.MaKH,  HoaDon.MaHD,  HoaDon.NgayXuat,  HoaDon.TongGiaTri,  KhachHang.HoTen  FROM HoaDon Join KhachHang on HoaDon.MaKH = KhachHang.MaKH  WHERE HoaDon.MaNV = @MaNV AND TongGiaTri >= @GiaTriMin AND TongGiaTri <= @GiaTriMax  ); |

**-Code C#:**

|  |
| --- |
| public static DataTable getDanhSachHDByValue(float giaTriMin, float giaTriMax, string maNV)  {  string query = String.Format("Select \* From fn\_timTheoGiaTri('{0}', '{1}', '{2}')", giaTriMin, giaTriMax, maNV);  return ExecuteQuery(query);  } |

### **2.9. Chức năng tìm hóa đơn theo ngày xuất hóa đơn**

**- Input: Ngày xuất hóa đơn, mã nhân viên**

**- Ouput: hóa đơn có ngày xuất như input**

**- Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_timTheoNgayXuat(@NgayXuat DATE, @MaNV NVARCHAR(30))  RETURNS TABLE  AS  RETURN(  SELECT \*  FROM HoaDon  WHERE MaNV = @MaNV AND NgayXuat = @NgayXuat  ); |

**-Code C#:**

|  |
| --- |
| public static DataTable getDanhSachHDByDate(DateTime date, string maNV)  {  string query = String.Format("Select \* From fn\_timTheoNgayXuat('{0}', '{1}')", date, maNV);  return ExecuteQuery(query);  } |

### **2.10. Chức năng hiển thị chi tiết hóa đơn**

**- Input: MaHD**

**- Ouput: Các linh kiện có trong hóa đơn**

**- Code SQL:**

|  |
| --- |
| CREATE Function fn\_loadChiTietHD(@MaHD NVARCHAR(30))  Returns Table  As  Return(  Select ChiTietHD.MaHD,  ChiTietHD.MaLK,  LinhKien.TenLK,  ChiTietHD.SoLuong,  ChiTietHD.DonGia,  ChiTietHD.TongTien  From ChiTietHD join LinhKien on ChiTietHD.MaLK = LinhKien.MaLK  Where ChiTietHD.MaHD = @MaHD  ); |

**-Code C#:**

|  |
| --- |
| public static DataTable getChiTietHD(string maHD)  {  string query = String.Format("Select \* From fn\_loadChiTietHD('{0}')", maHD);  return ExecuteQuery(query);  } |

### **2.11. Chức năng hiển thị Ca làm việc để chấm công**

**- Input: Mã nhân viên**

**- Output: Ca làm việc theo lịch phân công là gần nhất**

**- Code SQL:**

|  |
| --- |
| Create PROCEDURE sp\_GetCaLamViec  @MaNV NVARCHAR(30)  AS  BEGIN  SELECT TOP 1 -- Chỉ lấy 1 phân ca gần nhất  CL.MaCa,  CL.TenCa,  CL.Ngay,  CL.ThoiGianBD,  CL.ThoiGianKT  FROM  PhanCa PC  JOIN CaLamViec CL ON PC.MaCa = CL.MaCa  WHERE  PC.MaNhanVien = @MaNV  AND PC.TrangThai = 0 -- Chỉ lấy các phân ca chưa chấm công    ORDER BY  DATEDIFF(MINUTE, GETDATE(), CAST(CONCAT(CL.Ngay, ' ', CL.ThoiGianBD) AS DATETIME)) ASC; -- Sắp xếp theo độ gần gũi với thời gian hiện tại  END;  GO |

**-Code C#:**

|  |
| --- |
| public static DataTable GetCaLamViec(string maNV)  {  DataTable data = new DataTable();  try  {  using (SqlConnection conn = new SqlConnection(DataConnector.connectionString))  {  SqlCommand cmd = new SqlCommand("sp\_GetCaLamViec", conn);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", maNV);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(data);  }  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  return data;  } |

### **2.12. Chức năng Chấm công**

**- Input: Mã nhân viên, Mã ca**

**- Output: Nhân viên đã điểm danh cho ca làm**

**- Code SQL:**

|  |
| --- |
| Create PROCEDURE sp\_ChamCong  @MaNV NVARCHAR(30),  @MaCa NVARCHAR(30)  AS  BEGIN  BEGIN TRY  UPDATE PhanCa  SET TrangThai = 1 -- 1 có thể đại diện cho "Đã Chấm Công"  WHERE MaNhanVien = @MaNV AND MaCa = @MaCa;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Thực hiện chấm công lỗi!' + ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**- Code C#:**

|  |
| --- |
| public static void chamCong(string maNV, string MaCa)  {  using (SqlConnection conn = new SqlConnection(DataConnector.connectionString))  {  SqlCommand cmd = new SqlCommand("sp\_ChamCong", conn);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", maNV);  cmd.Parameters.AddWithValue("@MaCa", MaCa);  try  {  conn.Open();  cmd.ExecuteNonQuery(); // Thực thi stored procedure  MessageBox.Show("Chấm công thành công!");  }  catch(SqlException ex)  {  MessageBox.Show(ex.Message);  }  catch (Exception ex)  {  MessageBox.Show("Có lỗi xảy ra: " + ex.Message);  }  }  } |

### **2.13. Chức năng load danh sách ca làm đã thực hiện của nhân viên**

**- Input: Mã nhân viên**

**- Output: Danh sách ca làm việc đã được thực hiện theo mã nhân viên**

**- Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_GetCaLamViecByNhanVien  @MaNV NVARCHAR(30)  AS  BEGIN  SELECT  CL.MaCa,  CL.TenCa,  CL.Ngay,  CL.ThoiGianBD,  CL.ThoiGianKT  FROM  PhanCa PC  JOIN CaLamViec CL ON PC.MaCa = CL.MaCa  WHERE  PC.MaNhanVien = @MaNV and  pc.trangthai=1;  END;  GO |

**- Code C#:**

|  |
| --- |
| public static DataTable loadDanhSachCaLam(string maNV)  {  DataTable data = new DataTable();  try  {  using (SqlConnection conn = new SqlConnection(DataConnector.connectionString))  {  SqlCommand cmd = new SqlCommand("sp\_GetCaLamViecByNhanVien", conn);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", maNV);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);    adapter.Fill(data);    }  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  return data;  } |

### **2.14. Chức năng xem doanh thu theo ca**

**- Input: Mã ca, Mã nhân viên**

**- Output: Doanh thu của ca tìm theo mã ca và mã nhân viên**

**- Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_GetDoanhThuByMaCaVaNhanVien (  @MaCa NVARCHAR(30),  @MaNhanVien NVARCHAR(30)  )  RETURNS TABLE  AS  RETURN  (  SELECT  PC.MaCa,  CL.Ngay,  ISNULL(COUNT(DISTINCT HD.MaHD), 0) AS SoLuongDonHang,  ISNULL(SUM(CTHD.SoLuong), 0) AS SoLuongLinhKien,  ISNULL(SUM(HD.TongGiaTri), 0) AS TongDoanhThu  FROM PhanCa PC  JOIN CaLamViec CL ON PC.MaCa = CL.MaCa  LEFT JOIN HoaDon HD ON HD.MaNV = PC.MaNhanVien AND HD.NgayXuat = CL.Ngay  LEFT JOIN ChiTietHD CTHD ON CTHD.MaHD = HD.MaHD  WHERE PC.MaCa = @MaCa AND PC.MaNhanVien = @MaNhanVien  GROUP BY PC.MaCa, CL.Ngay  );  GO |

**- Code C#:**

|  |
| --- |
| public static DataTable getDoanhThuTheoCa(string maCa, string maNV)  {  DataTable data = new DataTable();  string query = @"  SELECT  MaCa, Ngay, SoLuongLinhKien, SoLuongDonHang, TongDoanhThu  FROM dbo.fn\_GetDoanhThuByMaCaVaNhanVien(@MaCa, @MaNhanVien)";  using (SqlConnection conn = new SqlConnection(DataConnector.connectionString))  {  SqlCommand cmd = new SqlCommand(query, conn);  cmd.Parameters.AddWithValue("@MaCa", maCa);  cmd.Parameters.AddWithValue("@MaNhanVien", maNV);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);    adapter.Fill(data);    }  return data;  } |

### **2.15. Chức năng xem doanh thu theo từng ca làm việc trong khoảng thời gian được chọn**

**- Input: mã nhân viên, thời gian bắt đầu, thời gian tìm kiếm**

**- Output: Danh sách doanh thu của các ngày trong khoảng thời gian tìm kiếm**

**- Code SQL:**

|  |
| --- |
| Create FUNCTION fn\_GetDoanhThuTheoNgay (  @MaNhanVien NVARCHAR(30),  @NgayBatDau DATE,  @NgayKetThuc DATE  )  RETURNS TABLE  AS  RETURN  (  SELECT  PC.MaCa,  CL.Ngay,  COUNT(DISTINCT HD.MaHD) AS SoLuongDonHang,  ISNULL(SUM(CTHD.SoLuong), 0) AS SoLuongLinhKien,  ISNULL(SUM(HD.TongGiaTri), 0) AS TongDoanhThu  FROM PhanCa PC  JOIN CaLamViec CL ON PC.MaCa = CL.MaCa  LEFT JOIN HoaDon HD ON HD.MaNV = PC.MaNhanVien AND HD.NgayXuat = CL.Ngay  LEFT JOIN ChiTietHD CTHD ON CTHD.MaHD = HD.MaHD  WHERE  PC.MaNhanVien = @MaNhanVien  AND CL.Ngay BETWEEN @NgayBatDau AND @NgayKetThuc  GROUP BY PC.MaCa, CL.Ngay  );  GO |

**- Code C#:**

|  |
| --- |
| public static DataTable getDoanhThuTheoCa(string maNV, DateTime ngayBatDau, DateTime ngayKetThuc)  {  DataTable dataTable = new DataTable();  using (SqlConnection connection = new SqlConnection(DataConnector.connectionString))  {  string query = "SELECT \* FROM fn\_GetDoanhThuTheoNgay(@MaNhanVien, @NgayBatDau, @NgayKetThuc)";  using (SqlCommand command = new SqlCommand(query, connection))  {  command.Parameters.AddWithValue("@MaNhanVien", maNV);  command.Parameters.AddWithValue("@NgayBatDau", ngayBatDau);  command.Parameters.AddWithValue("@NgayKetThuc", ngayKetThuc);  SqlDataAdapter adapter = new SqlDataAdapter(command);  adapter.Fill(dataTable);  }  }  return dataTable;  } |

### **2.16. Chức năng tìm kiếm nhân viên**

**- Input: Mã nhân viên**

**- Output: Dòng chứa nhân viên có mã nhân viên cần tìm**

**- Code SQL**:

|  |
| --- |
| CREATE FUNCTION fn\_timMaNV (@MaNV NVARCHAR(30))  RETURNS TABLE  AS  RETURN(  SELECT \* FROM NhanVien WHERE MaNV=@MaNV  ); |

**- Code C#:**

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e)  {  try  {  sqlCon.Open();  string sql = "SELECT \*FROM dbo.fn\_timMaNV(@MaNV)";  SqlDataAdapter adapter = new SqlDataAdapter(sql, sqlCon.GetConnection());  adapter.SelectCommand.Parameters.AddWithValue("@MaNV", txtNhap.Text);  DataTable dt = new DataTable();  adapter.Fill(dt);  dgvDanhSachNV.DataSource = dt;  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  finally  {  sqlCon.Close();  }  } |

### **2.17 Chức năng quản lý nhân viên**

#### **2.17.1 Chức năng thêm tài khoản nhân viên**

**- Input: mã nhân viên, họ tên, số điện thoại, ngày bắt đầu làm việc, ngày sinh, giới tính, địa chỉ, lương của nhân viên cần thêm vào**

**- Output: tài khoản nhân viên được thêm thành công vào hệ thống**

**- Code SQL:**

|  |
| --- |
| CREATE PROC nv\_InsertNhanVien  @MaNV NVARCHAR(30),  @HoTen NVARCHAR(50),  @SDT CHAR(11),  @NgayBDLamViec DATE,  @NgaySinh DATE,  @GioiTinh NVARCHAR(3),  @DiaChi NVARCHAR(100),  @Luong FLOAT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO NhanVien(MaNV, HoTen, SDT, NgayBDLamViec, NgaySinh, GioiTinh, DiaChi, Luong)  VALUES (@MaNV, @HoTen, @SDT, @NgayBDLamViec, @NgaySinh, @GioiTinh, @DiaChi, @Luong);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Thêm nhân viên lỗi!' + ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END; |

**- Code C#:**

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  sqlCon.Open();  string proc = String.Format("Execute nv\_InsertNhanVien @MaNV, @HoTen, @SDT, @NgayBDLamViec, @NgaySinh, @GioiTinh, @DiaChi, @Luong");  SqlCommand cmd = new SqlCommand(proc, sqlCon.GetConnection());  cmd.Parameters.AddWithValue("@MaNV", txtMaNV.Text);  cmd.Parameters.AddWithValue("@HoTen", txtHoTen.Text);  cmd.Parameters.AddWithValue("@SDT", txtSDT.Text);  cmd.Parameters.AddWithValue("@NgayBDLamViec", dtpNgayBDLamViec.Value);  cmd.Parameters.AddWithValue("@NgaySinh", dtpNgaySinh.Value);  cmd.Parameters.AddWithValue("@GioiTinh", cbGioiTinh.SelectedItem.ToString());  cmd.Parameters.AddWithValue("@DiaChi", txtDiaChi.Text);  cmd.Parameters.AddWithValue("@Luong", 0);  cmd.ExecuteNonQuery();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  finally  {  sqlCon.Close();  }  FQuanlynhanvien\_Load(sender, e);  } |

#### **2.17.2 Chức năng xoá tài khoản nhân viên**

**-Input: mã nhân viên của nhân viên cần xoá**

**-Output: tài khoản nhân viên bị xoá ra khỏi hệ thống**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE [dbo].[nv\_deleteNhanVien]  @MaNV nvarchar(30)  AS  BEGIN  DECLARE @sqlString nvarchar(2000);  BEGIN TRANSACTION  BEGIN TRY  DELETE FROM PhanCa  WHERE MaNhanVien = @MaNV  DELETE FROM NhanVien  WHERE MaNV = @MaNV  SET @sqlString = 'ALTER ROLE NhanVien DROP MEMBER [' + @MaNV + ']'  EXEC sp\_executesql @sqlString;  SET @sqlString = 'DROP USER [' + @MaNV + ']'  EXEC sp\_executesql @sqlString;  SET @sqlString = 'DROP LOGIN [' + @MaNV + ']'  EXEC sp\_executesql @sqlString;  COMMIT TRANSACTION;  PRINT N'Đã xoá nhân viên ' + @MaNV+ N' thành công'  END TRY  BEGIN CATCH  DECLARE @err nvarchar(MAX)  SELECT @err = N'Xoá nhân viên ' + @MaNV + N' thất bại' + ERROR\_MESSAGE()  RAISERROR (@err,16,1)  ROLLBACK TRANSACTION  END CATCH  END; |

-Code C#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {  sqlCon.Open();  string proc = String.Format("Execute nv\_deleteNhanVien @MaNV");  SqlCommand cmd = new SqlCommand(proc, sqlCon.GetConnection());  cmd.Parameters.AddWithValue("@MaNV", txtNhap.Text);  cmd.ExecuteNonQuery();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  finally  {  sqlCon.Close();  }  FQuanlynhanvien\_Load(sender, e);  } |

#### **2.17.3 Chức năng sửa tài khoản nhân viên**

**-Input: thông tin cần sửa của nhân viên**

**-Output: thông tin của nhân viên được sửa thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROC nv\_updateNhanVien  @MaNV NVARCHAR(30),  @HoTen NVARCHAR(50),  @SDT CHAR(11),  @NgayBDLamViec DATE,  @NgaySinh DATE,  @GioiTinh NVARCHAR(3),  @DiaChi NVARCHAR(100),  @Luong FLOAT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  UPDATE [dbo].[NhanVien]  SET MaNV =@MaNV,  HoTen=@HoTen,  SDT=@SDT,  NgayBDLamViec=@NgayBDLamViec,  NgaySinh=@NgaySinh,  GioiTinh=@GioiTinh,  DiaChi=@DiaChi,  Luong=@Luong  WHERE MaNV=@MaNV  COMMIT TRANSACTION  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Sửa nhân viên lỗi!' + ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  try  {  sqlCon.Open();  string proc = String.Format("Execute nv\_updateNhanVien @MaNV, @HoTen, @SDT, @NgayBDLamViec, @NgaySinh, @GioiTinh, @DiaChi, @Luong");  SqlCommand cmd = new SqlCommand(proc, sqlCon.GetConnection());  cmd.Parameters.AddWithValue("@MaNV", txtMaNV.Text);  cmd.Parameters.AddWithValue("@HoTen", txtHoTen.Text);  cmd.Parameters.AddWithValue("@SDT", txtSDT.Text);  cmd.Parameters.AddWithValue("@NgayBDLamViec", dtpNgayBDLamViec.Value);  cmd.Parameters.AddWithValue("@NgaySinh", dtpNgaySinh.Value);  cmd.Parameters.AddWithValue("@GioiTinh", cbGioiTinh.SelectedItem.ToString());  cmd.Parameters.AddWithValue("@DiaChi", txtDiaChi.Text);  cmd.Parameters.AddWithValue("@Luong", 0);  cmd.ExecuteNonQuery();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  finally  {  sqlCon.Close();  }  FQuanlynhanvien\_Load(sender, e);  } |

### **2.18. Tính doanh thu của cửa hàng trong khoảng thời gian**

**-Input: ngày bắt đầu, ngày kết thúc**

**-Output: doanh thu của cửa hàng từ ngày bắt đầu đến ngày kết thúc do chủ cửa hàng nhập vào**

**-Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_tongDoanhThu (@Start Date, @End Date) RETURNS MONEY  AS  BEGIN  DECLARE @Total MONEY  SELECT @Total = sum(TongGiaTri)  FROM HoaDon  WHERE NgayXuat BETWEEN @Start AND @End  RETURN ISNULL(@Total,0);  END; |

**-Code C#:**

|  |
| --- |
| private void btnDoanhThu\_Click(object sender, EventArgs e)  {  try  {  sqlCon.Open();  string func = "SELECT dbo.fn\_tongDoanhThu(@Start, @End)";  SqlCommand cmd = new SqlCommand(func, sqlCon.GetConnection());  cmd.Parameters.AddWithValue("@Start", dtpNgayBatDau.Value);  cmd.Parameters.AddWithValue("@End", dtpNgayKetThuc.Value);  txtDoanhThu.Text = cmd.ExecuteScalar().ToString();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  finally  {  sqlCon.Close();  }  } |

### **2.19. Tính doanh thu và lợi nhuận của cửa hàng theo tháng**

**-Input: tháng, năm cần tính doanh thu**

**-Output: doanh thu và lợi nhuận theo tháng trong năm cần tính**

**-Code SQL tính doanh thu theo tháng:**

|  |
| --- |
| CREATE FUNCTION fn\_doanhThuTheoThang (@Month INT, @Year INT) RETURNS MONEY  AS  BEGIN  DECLARE @Start Date, @End Date, @Total MONEY  SET @Start = DATEFROMPARTS(@Year, @Month, 1);  SET @End = EOMONTH(@Start);  SELECT @Total = sum(TongGiaTri)  FROM HoaDon  WHERE NgayXuat >= @Start AND NgayXuat <= @End;  RETURN ISNULL(@Total,0);  END; |

**-Code SQL tính tổng tiền nhập hàng theo tháng:**

|  |
| --- |
| CREATE FUNCTION fn\_tongNhapHangTrongThang(@Month INT, @Year INT) RETURNS MONEY  AS  BEGIN  DECLARE @Start Date, @End Date, @Total MONEY  SET @Start = DATEFROMPARTS(@Year, @Month, 1);  SET @End = EOMONTH(@Start);  SELECT @Total = sum(GiaTriDonNhap)  FROM DonNhapHang  WHERE NgayNhap >= @Start AND NgayNhap <= @End;  RETURN ISNULL(@Total,0);  END; |

**-Code SQL tính lợi nhuận theo tháng:**

|  |
| --- |
| CREATE FUNCTION fn\_loiNhuanTheoThang(@Month INT, @Year INT) RETURNS MONEY  AS  BEGIN  DECLARE @Total MONEY  SET @Total = (SELECT dbo.fn\_doanhThuTheoThang(@Month, @Year)) - (SELECT dbo.fn\_tongNhapHangTrongThang(@Month, @Year))  RETURN ISNULL(@Total,0);  END; |

**-Code C#:**

|  |
| --- |
| private void btnTinh\_Click(object sender, EventArgs e)  {  try  {  sqlCon.Open();  string func1 = "SELECT dbo.fn\_doanhThuTheoThang(@Month, @Year)";  SqlCommand cmd = new SqlCommand(func1, sqlCon.GetConnection());  cmd.Parameters.AddWithValue("@Month", Convert.ToInt32(cbThang.SelectedItem));  cmd.Parameters.AddWithValue("@Year", Convert.ToInt32(txtNam.Text));  txtDoanhThuThang.Text = cmd.ExecuteScalar().ToString();  string func2 = "SELECT dbo.fn\_loiNhuanTheoThang(@Month, @Year)";  SqlCommand cmd2 = new SqlCommand(func2, sqlCon.GetConnection());  cmd2.Parameters.AddWithValue("@Month", Convert.ToInt32(cbThang.SelectedItem));  cmd2.Parameters.AddWithValue("@Year", Convert.ToInt32(txtNam.Text));  txtLoiNhuan.Text = cmd2.ExecuteScalar().ToString();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  finally  {  sqlCon.Close();  }  } |

### **2.20 Chức năng quản lý sản phẩm**

#### **2.20.1 Chức năng thêm sản phẩm**

**-Input: mã linh kiện, tên linh kiện, giá tiền, mô tả, trạng thái, số lượng, hình ảnh, mã nhóm linh kiện, mã nhà cung cấp linh kiện**

**-Output: thông tin của linh kiện được thêm thành công vào hệ thống**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_AddLinhKien  @MaLK nchar(10),  @TenLK nvarchar(50),  @GiaTien float,  @MoTa nvarchar(500),  @TrangThai nvarchar(20),  @SoLuong int,  @HinhAnh varbinary(max),  @MaNhom nchar(10),  @MaNCC nchar(10)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  IF @MaLK IS NULL OR @MaLK = ''  BEGIN  RAISERROR(N'Mã linh kiện không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @TenLK IS NULL OR @TenLK = ''  BEGIN  RAISERROR(N'Tên linh kiện không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @GiaTien IS NULL OR @GiaTien <= 0  BEGIN  RAISERROR(N'Giá tiền không hợp lệ.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @MoTa IS NULL OR @MoTa = ''  BEGIN  RAISERROR(N'Mô tả không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @TrangThai IS NULL OR @TrangThai = ''  BEGIN  RAISERROR(N'Trạng thái không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @MaNhom IS NULL OR @MaNhom = ''  BEGIN  RAISERROR(N'Mã nhóm không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @MaNCC IS NULL OR @MaNCC = ''  BEGIN  RAISERROR(N'Mã nhà cung cấp không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF @HinhAnh = ''  BEGIN  RAISERROR(N'Hình ảnh không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  IF EXISTS (SELECT 1 FROM LinhKien WHERE MaLK = @MaLK)  BEGIN  RAISERROR(N'Thêm linh kiện lỗi! Linh kiện với mã %s đã tồn tại.', 16, 1, @MaLK)  ROLLBACK TRANSACTION;  RETURN;  END  INSERT INTO LinhKien (MaLK, TenLK, GiaTien, MoTa, TrangThai, SoLuong, AnhLK, MaNhom, MaNCC)  VALUES (@MaLK, @TenLK, @GiaTien, @MoTa, @TrangThai, @SoLuong, @HinhAnh, @MaNhom, @MaNCC);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Thêm linh kiện lỗi! ';  RAISERROR(@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnThemLK\_Click(object sender, EventArgs e)  {  try  {  byte[] hinhAnhBytes = null;  if (ptbHinhAnh.Image != null)  {  hinhAnhBytes = ImageToByteArray(ptbHinhAnh.Image);  }  cnt.Open();  using (SqlCommand command = new SqlCommand("sp\_AddLinhKien", cnt.GetConnection()))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaLK", txtMaSP.Text);  command.Parameters.AddWithValue("@TenLK", txtTenSP.Text);  command.Parameters.AddWithValue("@GiaTien", txtGiaTien.Text);  command.Parameters.AddWithValue("@MoTa", txtMoTa.Text);  command.Parameters.AddWithValue("@SoLuong", 0);  command.Parameters.AddWithValue("@TrangThai", "Đang Bán");  command.Parameters.AddWithValue("@HinhAnh", (object)hinhAnhBytes ?? Encoding.Unicode.GetBytes(DBNull.Value.ToString()));  if (cbMaNH.SelectedItem != null)  {  var selectedItem = cbMaNH.SelectedItem as dynamic;  if (selectedItem != null && selectedItem.Value != null)  {  string selectedId = selectedItem.Value;  command.Parameters.AddWithValue("@MaNhom", selectedId);  }  else  {  command.Parameters.AddWithValue("@MaNhom", DBNull.Value);  }  }  else  {  command.Parameters.AddWithValue("@MaNhom", DBNull.Value);  }  if (cbMaNCC.SelectedItem != null)  {  var selectedItem = cbMaNCC.SelectedItem as dynamic;  if (selectedItem != null && selectedItem.Value != null)  {  string selectedId = selectedItem.Value;  command.Parameters.AddWithValue("@MaNCC", selectedId);  }  else  {  command.Parameters.AddWithValue("@MaNCC", DBNull.Value);  }  }  else  {  command.Parameters.AddWithValue("@MaNCC", DBNull.Value);  }  int result = command.ExecuteNonQuery();  MessageBox.Show("Thêm sản phẩm thành công!");  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  finally  {  cnt.Close();  }  FQuanlysanpham\_Load(sender, e);  } |

#### **2.20.2 Chức năng sửa thông tin sản phẩm**

**-Input: thông tin cần sửa của linh kiện**

**-Output: thông tin của linh kiện được sửa thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_UpdateLinhKien  @MaLK nchar(10),  @TenLK nvarchar(50),  @GiaTien float,  @MoTa nvarchar(500),  @TrangThai nvarchar(20),  @SoLuong int,  @MaNhom nchar(10),  @MaNCC nchar(10),  @HinhAnh varbinary(max) = NULL  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  IF NOT EXISTS (SELECT 1 FROM LinhKien WHERE MaLK = @MaLK)  BEGIN  RAISERROR(N'Cập nhật lỗi! Linh kiện với mã %s không tồn tại.', 16, 1, @MaLK);  ROLLBACK TRANSACTION;  RETURN;  END  UPDATE LinhKien  SET TenLK = @TenLK,  GiaTien = @GiaTien,  MoTa = @MoTa,  TrangThai = @TrangThai,  SoLuong = @SoLuong,  MaNhom = @MaNhom,  MaNCC = @MaNCC,  AnhLK = @HinhAnh  WHERE MaLK = @MaLK;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Cập nhật lỗi! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  byte[] hinhAnhBytes = ImageToByteArray(ptbHinhAnh2.Image);  try  {  cnt.Open();  using (SqlCommand command = new SqlCommand("sp\_UpdateLinhKien", cnt.GetConnection()))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaLK", txtMaSP2.Text);  command.Parameters.AddWithValue("@TenLK", txtTenSP2.Text);  command.Parameters.AddWithValue("@GiaTien", txtGiaTien2.Text);  command.Parameters.AddWithValue("@MoTa", txtMoTa2.Text);  command.Parameters.AddWithValue("@SoLuong", soluong);  command.Parameters.AddWithValue("@TrangThai", cbTrangThai.Text);  command.Parameters.AddWithValue("@HinhAnh", (object)hinhAnhBytes ?? DBNull.Value);  if (cbMaNH2.SelectedItem != null)  {  var selectedItem = (dynamic)cbMaNH2.SelectedItem;  string selectedId = selectedItem.Value;  command.Parameters.AddWithValue("@MaNhom", selectedId);  }  if (cbMaNCC2.SelectedItem != null)  {  var selectedItem = (dynamic)cbMaNCC2.SelectedItem;  string selectedId = selectedItem.Value;  command.Parameters.AddWithValue("@MaNCC", selectedId);  }  int result = command.ExecuteNonQuery();  }  MessageBox.Show("Sửa sản phẩm thành công!");  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  FQuanlysanpham\_Load(sender, e);  } |

### **2.21 Chức năng tìm kiếm linh kiện theo tên linh kiện**

**-Input: tên linh kiện**

**-Output: dòng chứa linh kiện có tên linh kiện cần tìm**

**-Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_timTenLK(@TenLK NVARCHAR(50))  RETURNS TABLE  AS  RETURN(  SELECT \*  FROM ViewSanPham  WHERE TenLK LIKE '%' + @TenLK + '%'  ); |

**-Code C#:**

|  |
| --- |
| private void txtTimKiem\_TextChanged(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand command = new SqlCommand("SELECT \* FROM fn\_timTenLK(@TenLK)", cnt.GetConnection()))  {  command.Parameters.AddWithValue("@TenLK", txtTimKiem.Text);  using (SqlDataAdapter adapter = new SqlDataAdapter(command))  {  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); // Đổ dữ liệu vào DataTable  gvSP.DataSource = dataTable;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tìm kiếm: " + ex.Message);  }  finally  {  cnt.Close();  }  } |

### **2.22 Chức năng tìm kiếm linh kiện theo giá**

**-Input: giá linh kiện**

**-Output: dòng chứa linh kiện có giá cần tìm**

**-Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_timTheoGiaLK(@GiaTien FLOAT)  RETURNS TABLE  AS  RETURN(  SELECT \*  FROM LinhKien  WHERE GiaTien <= @GiaTien  ); |

**-Code C#:**

|  |
| --- |
| private void txtTimGia\_TextChanged(object sender, EventArgs e)  {    try  {  cnt.Open();  using (SqlCommand command = new SqlCommand("SELECT \* FROM fn\_timTheoGiaLK(@GiaTien)", cnt.GetConnection()))  {  string giaTienStr = txtTimGia.Text;  float giaTien;  if (giaTienStr == "")  {  command.Parameters.AddWithValue("@GiaTien", float.MaxValue);  }  else  if (float.TryParse(giaTienStr, out giaTien))  {  command.Parameters.AddWithValue("@GiaTien", giaTien);  }  using (SqlDataAdapter adapter = new SqlDataAdapter(command))  {  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  gvSP.DataSource = dataTable;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tìm kiếm nhà cung cấp: " + ex.Message);  }  finally  {  cnt.Close();  }  } |

### **2.23 Chức năng tìm kiếm linh kiện theo nhóm linh kiện**

**-Input: nhóm linh kiện**

**-Output: dòng chứa linh kiện theo nhóm linh kiện cần tìm**

**-Code SQL:**

|  |
| --- |
| CREATE FUNCTION fn\_timNhomLK(@TenNhom NVARCHAR(30))  RETURNS TABLE  AS  RETURN(  SELECT \*  FROM ViewSanPham  WHERE TenNhom LIKE '%' + @TenNhom + '%'  ); |

**-Code C#:**

|  |
| --- |
| private void txtTimNhom\_TextChanged(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand command = new SqlCommand("SELECT \* FROM fn\_timNhomLK(@TenNhom)", cnt.GetConnection()))  {  command.Parameters.AddWithValue("@TenNhom", txtTimNhom.Text);  using (SqlDataAdapter adapter = new SqlDataAdapter(command))  {  DataTable dataTable = new DataTable();  adapter.Fill(dataTable); // Đổ dữ liệu vào DataTable  gvSP.DataSource = dataTable;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tìm kiếm nhà cung cấp: " + ex.Message);  }  finally  {  cnt.Close();  }  } |

### **2.24 Chức năng thêm đơn nhập hàng**

**-Input: ngày nhập hàng, mã nhà cung cấp, giá trị đơn nhập hàng**

**-Output: đơn nhập hàng thêm vào thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_InsertDonNhapHang  @NgayNhap DATE,  @MaNCC NVARCHAR(30),  @GiaTriDonNhap FLOAT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO DonNhapHang (NgayNhap, MaNCC, GiaTriDonNhap)  VALUES (@NgayNhap, @MaNCC, @GiaTriDonNhap);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi thêm đơn nhập hàng! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  ROLLBACK TRANSACTION;  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnThemDonNhap\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_InsertDonNhapHang", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@NgayNhap",dtNgayNhap.Value.Date);  if (cbMaNCC2.SelectedItem != null)  {  var selectedItem = cbMaNCC2.SelectedItem as dynamic;  if (selectedItem != null && selectedItem.Value != null)  {  string selectedId = selectedItem.Value;  cmd.Parameters.AddWithValue("@MaNCC", selectedId);  }  else  {  cmd.Parameters.AddWithValue("@MaNCC", DBNull.Value);  }  }  else  {  cmd.Parameters.AddWithValue("@MaNCC", DBNull.Value);  }  cmd.Parameters.AddWithValue("@GiaTriDonNhap", 1);  cmd.ExecuteNonQuery();  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi" + ex.Message);  }  finally  {  cnt.Close();  }  FQuanlynhaphang\_Load(sender, e);  } |

### **2.25 Chức năng xoá đơn nhập hàng**

**-Input: mã đơn nhập hàng**

**-Output: xoá đơn nhập hàng thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_DeleteDonNhapHang  @MaDonNhap NCHAR(30)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  IF NOT EXISTS (SELECT 1 FROM DonNhapHang WHERE MaDonNhap = @MaDonNhap)  BEGIN  RAISERROR(N'Lỗi xóa! Đơn nhập hàng với mã %s không tồn tại.', 16, 1, @MaDonNhap);  ROLLBACK TRANSACTION;  RETURN;  END  DELETE FROM NhapHang  WHERE MaDonNhap = @MaDonNhap;  DELETE FROM DonNhapHang  WHERE MaDonNhap = @MaDonNhap;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi xóa đơn nhập hàng! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnXoaDonNhap\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_DeleteDonNhapHang", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;    cmd.Parameters.AddWithValue("@MaDonNhap", MaDonNhap);    cmd.ExecuteNonQuery();  MessageBox.Show("Xóa đơn nhập hàng và các sản phẩm liên quan thành công!");  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  finally  {  cnt.Close();  }  btnBack\_Click(sender, e);  } |

### **2.26 Chức năng thêm linh kiện vào đơn nhập hàng**

**-Input: mã linh kiện, mã đơn nhập, số lượng, đơn giá, tổng tiền**

**-Output: linh kiện được thêm vào đơn nhập hàng thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_InsertNhapHang  @MaLK NVARCHAR(30),  @MaDonNhap NVARCHAR(30),  @SoLuong INT,  @DonGia FLOAT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  INSERT INTO NhapHang (MaLK, MaDonNhap, SoLuong, DonGia, TongTien)  VALUES (@MaLK, @MaDonNhap, @SoLuong, @DonGia, @TongTien);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi thêm nhập hàng! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {    try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_InsertNhapHang", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaDonNhap", MaDonNhap);  if (cbTenSP.SelectedItem != null)  {  var selectedItem = cbTenSP.SelectedItem as dynamic;  if (selectedItem != null && selectedItem.Value != null)  {  string selectedId = selectedItem.Value;  cmd.Parameters.AddWithValue("@MaLK", selectedId);  }  else  {  cmd.Parameters.AddWithValue("@MaLK", DBNull.Value);  }  }  else  {  cmd.Parameters.AddWithValue("@MaLK", DBNull.Value);  }  int soluong = 0;  float dongia = 0f;  if (string.IsNullOrEmpty(txtSoLuong.Text) || !int.TryParse(txtSoLuong.Text, out soluong))  {  MessageBox.Show("Số lượng không hợp lệ.");  return;  }  if (string.IsNullOrEmpty(txtGiaNhap.Text) || !float.TryParse(txtGiaNhap.Text, out dongia))  {  MessageBox.Show("Giá nhập không hợp lệ.");  return;  }  cmd.Parameters.AddWithValue("@SoLuong", soluong);  cmd.Parameters.AddWithValue("@DonGia", dongia);  cmd.Parameters.AddWithValue("@TongTien", soluong \* dongia);  cmd.ExecuteNonQuery();  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi" + ex.Message);  }  finally  {  cnt.Close();  }  btnBack2\_Click(sender, e);  } |

### **2.27 Chức năng xoá linh kiện khỏi đơn nhập hàng**

**-Input: mã linh kiện, mã đơn nhập**

**-Output: xoá linh kiện khỏi đơn nhập hàng thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_DeleteNhapHang  @MaLK NVARCHAR(30),  @MaDonNhap NVARCHAR(30)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DELETE FROM NhapHang  WHERE  MaLK = @MaLK AND  MaDonNhap = @MaDonNhap;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi xóa nhập hàng! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnXoaLK\_Click(object sender, EventArgs e)  {  try  {  // Mở kết nối  cnt.Open();  // Tạo đối tượng SqlCommand để gọi thủ tục  using (SqlCommand cmd = new SqlCommand("sp\_DeleteNhapHang", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaDonNhap", MaDonNhap);  cmd.Parameters.AddWithValue("@MaLK", MaLk);  cmd.ExecuteNonQuery();  }  }  catch (SqlException sqlEx)  {    MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi xóa đơn nhập hàng: " + ex.Message);  }  finally  {  cnt.Close();  }  btnBack5\_Click(sender, e);  } |

### **2.28 Chức năng sửa linh kiện trong đơn nhập hàng**

**-Input: thông tin linh kiện cần sửa trong đơn nhập hàng**

**-Output: sửa linh kiện trong đơn nhập hàng thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_UpdateNhapHang  @MaLK NVARCHAR(30),  @MaDonNhap NVARCHAR(30),  @SoLuong INT,  @DonGia FLOAT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  UPDATE NhapHang  SET  SoLuong = @SoLuong,  DonGia = @DonGia,  TongTien = @TongTien  WHERE  MaLK = @MaLK AND  MaDonNhap = @MaDonNhap;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi cập nhật nhập hàng! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_UpdateNhapHang", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaDonNhap", MaDonNhap);    cmd.Parameters.AddWithValue("@MaLK", MaLk);  int soluong = 0;  float dongia = 0f;  if (string.IsNullOrEmpty(txtSoLuong.Text) || !int.TryParse(txtSoLuong.Text, out soluong))  {  MessageBox.Show("Số lượng không hợp lệ.");  return;  }  if (string.IsNullOrEmpty(txtGiaNhap.Text) || !float.TryParse(txtGiaNhap.Text, out dongia))  {  MessageBox.Show("Giá nhập không hợp lệ.");  return;  }  cmd.Parameters.AddWithValue("@SoLuong", soluong);  cmd.Parameters.AddWithValue("@DonGia", dongia);  cmd.Parameters.AddWithValue("@TongTien", soluong \* dongia);  cmd.ExecuteNonQuery();    }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi sửa đơn nhập hàng: " + ex.Message);  }  finally  {  cnt.Close();  }  btnBack5\_Click(sender, e);  } |

### **2.29 Chức năng thêm nhóm linh kiện vào kho**

**-Input: mã nhóm linh kiện, tên nhóm linh kiện**

**-Output: nhóm linh kiện thêm vào kho thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_InsertNhomLinhKien  @MaNhom NVARCHAR(10),  @TenNhom NVARCHAR(50)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY    IF @MaNhom IS NULL OR LTRIM(RTRIM(@MaNhom)) = ''  BEGIN  RAISERROR(N'Lỗi thêm nhóm linh kiện! Mã nhóm không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  -- Insert the new component group  INSERT INTO NhomLinhKien (MaNhom, TenNhom, SoLuongLK)  VALUES (@MaNhom, @TenNhom, 0);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Rollback the transaction and raise a custom error message  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi thêm nhóm linh kiện! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_InsertNhomLinhKien", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;    cmd.Parameters.AddWithValue("@MaNhom", txtMaNH.Text);  cmd.Parameters.AddWithValue("@TenNhom", txtTenNH.Text);  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm Thành Công");  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi thêm Nhóm Linh Kiện: " + ex.Message);  }  finally  {  cnt.Close();  }  FQuanlykho\_Load(sender, e);  } |

### **2.30 Chức năng sửa nhóm linh kiện trong kho**

**-Input: thông tin linh kiện trong kho cần sửa**

**-Output: nhóm linh kiện trong kho được sửa thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_UpdateNhomLinhKien  @MaNhom NVARCHAR(10),  @TenNhom NVARCHAR(50)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  IF @MaNhom IS NULL OR LTRIM(RTRIM(@MaNhom)) = ''  BEGIN  RAISERROR(N'Lỗi cập nhật! Mã nhóm không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  UPDATE NhomLinhKien  SET TenNhom = @TenNhom  WHERE MaNhom = @MaNhom;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi cập nhật nhóm linh kiện! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_UpdateNhomLinhKien", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNhom", txtMaNH.Text);  cmd.Parameters.AddWithValue("@TenNhom", txtTenNH.Text);  cmd.ExecuteNonQuery();  MessageBox.Show("Update Thành Công");  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi sửa Nhóm Linh Kiện: " + ex.Message);  }  finally  {  cnt.Close();  }  FQuanlykho\_Load(sender, e);  } |

### **2.31 Chức năng thêm nhà cung cấp**

**-Input: tên nhà cung cấp, địa chỉ, số điện thoại**

**-Output: nhà cung cấp được thêm thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE AddNhaCungCap  @TenNhaCungCap NVARCHAR(100),  @DiaChi NVARCHAR(200),  @SoDienThoai NVARCHAR(10)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  IF @TenNhaCungCap IS NULL OR LTRIM(RTRIM(@TenNhaCungCap)) = ''  BEGIN  RAISERROR(N'Lỗi thêm nhà cung cấp! Tên nhà cung cấp không được để trống.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  INSERT INTO NhaCungCap (TenNhaCungCap, DiaChi, SDT)  VALUES (@TenNhaCungCap, @DiaChi, @SoDienThoai);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi thêm nhà cung cấp! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("AddNhaCungCap", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TenNhaCungCap", txtTenNCC.Text);  cmd.Parameters.AddWithValue("@DiaChi", txtDiaChi.Text);  cmd.Parameters.AddWithValue("@SoDienThoai", txtSDT.Text);  cmd.ExecuteNonQuery();  }  MessageBox.Show("Thêm nhà cung cấp thành công!");  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi thêm nhà cung cấp: " + ex.Message);  }  finally  {  cnt.Close();  }  FNhaCungCap\_Load(sender,e);  } |

### **2.32 Chức năng sửa nhà cung cấp**

**-Input: thông tin cần sửa của nhà cung cấp**

**-Output: thông tin nhà cung cấp được sửa thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE UpdateNhaCungCap  @MaNhaCungCap NVARCHAR(20),  @TenNhaCungCap NVARCHAR(100),  @DiaChi NVARCHAR(200),  @SoDienThoai NVARCHAR(15)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  UPDATE NhaCungCap  SET TenNhaCungCap = @TenNhaCungCap,  DiaChi = @DiaChi,  SDT = @SoDienThoai  WHERE MaNCC = @MaNhaCungCap;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi cập nhật nhà cung cấp! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("UpdateNhaCungCap", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNhaCungCap", txtMaNCC.Text);  cmd.Parameters.AddWithValue("@TenNhaCungCap", txtTenNCC.Text);  cmd.Parameters.AddWithValue("@DiaChi", txtDiaChi.Text);  cmd.Parameters.AddWithValue("@SoDienThoai", txtSDT.Text);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  MessageBox.Show("Cập nhật nhà cung cấp thành công!");  }  else  {  MessageBox.Show("Không tìm thấy nhà cung cấp với mã đã cho.");  }  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi cập nhật nhà cung cấp: " + ex.Message);  }  finally  {  cnt.Close();  }  FNhaCungCap\_Load(sender, e);  } |

### **2.33 Chức năng tìm kiếm nhà cung cấp theo tên nhà cung cấp**

**-Input: tên nhà cung cấp**

**-Output: dòng chứa nhà cung cấp theo tên cần tìm**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE SearchNhaCungCap  @TenNhaCungCap NVARCHAR(100)  AS  BEGIN  SELECT \* FROM ViewNhaCungCap  WHERE [Tên Nhà Cung Cấp] LIKE '%' + @TenNhaCungCap + '%';  END |

**-Code C#:**

|  |
| --- |
| private void txtTimKiem\_KeyUp(object sender, KeyEventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("SearchNhaCungCap", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TenNhaCungCap", txtTimKiem.Text);  using (SqlDataAdapter adapter = new SqlDataAdapter(cmd))  {  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  gvNCC.DataSource = dataTable;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tìm kiếm nhà cung cấp: " + ex.Message);  }  finally  {  cnt.Close();  }  } |

### **2.34 Chức năng thêm ca làm việc của nhân viên**

**-Input: tên ca, ngày, thời gian bắt đầu, thời gian kết thúc**

**-Output: ca làm việc được thêm thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_AddCaLamViec  @Ngay DATE  AS  BEGIN  BEGIN TRANSACTION;  BEGIN TRY  IF EXISTS (SELECT 1 FROM CaLamViec WHERE Ngay = @Ngay)  BEGIN  RAISERROR(N'Đã tạo ca làm', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  INSERT INTO CaLamViec (TenCa, Ngay, ThoiGianBD, ThoiGianKT)  VALUES  (N'Sáng', @Ngay, '07:00', '12:00'),  (N'Chiều', @Ngay, '12:00', '17:00'),  (N'Tối', @Ngay, '17:00', '22:00');  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = 'Lỗi phân ca';  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnTaoCa\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_AddCaLamViec", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@Ngay", dtNgay.Value.Date);  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm Thành Công");  }  }  catch (Exception ex)  {  MessageBox.Show("Error loading data: " + ex.Message);  }  finally  {  cnt.Close();  }  FPhanCa\_Load(sender, e);  } |

### **2.35 Chức năng thêm nhân viên vào ca làm việc**

**-Input: mã nhân viên, mã ca làm việc**

**-Output: nhân viên được thêm vào ca làm việc thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_InsertPhanCa  @MaNhanVien NCHAR(10),  @MaCa NCHAR(10)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  IF NOT EXISTS (SELECT 1 FROM PhanCa WHERE MaNhanVien = @MaNhanVien AND MaCa = @MaCa)  BEGIN  INSERT INTO PhanCa (MaNhanVien, MaCa, trangthai)  VALUES (@MaNhanVien, @MaCa, 0);  END  ELSE  BEGIN  RAISERROR(N'Phân ca đã tồn tại cho nhân viên %s với mã ca %s.', 16, 1, @MaNhanVien, @MaCa);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi trong quá trình phân ca! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

**-Code C#:**

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_InsertPhanCa", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  if (cbNhanVien.SelectedItem != null)  {  var selectedItem = cbNhanVien.SelectedItem as dynamic;  if (selectedItem != null && selectedItem.Value != null)  {  string selectedId = selectedItem.Value;  cmd.Parameters.AddWithValue("@MaNhanVien", selectedId);  }  else  {  cmd.Parameters.AddWithValue("@MaNhanVien", DBNull.Value);  }  }  else  {  cmd.Parameters.AddWithValue("@MaNhanVien", DBNull.Value);  }  cmd.Parameters.AddWithValue("@MaCa", Maca);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  MessageBox.Show("Thêm phân ca thành công!");  }  else  {  MessageBox.Show("Phân ca đã tồn tại!");  }  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  finally  {  cnt.Close();  }  btnBack3\_Click(sender, e);  } |

### **2.36 Chức năng xoá nhân viên khỏi ca làm việc**

**-Input: mã nhân viên, mã ca**

**-Output: xoá nhân viên khỏi ca làm việc thành công**

**-Code SQL:**

|  |
| --- |
| CREATE PROCEDURE sp\_DeletePhanCa  @MaNhanVien NCHAR(10),  @MaCa NCHAR(10)  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  DELETE FROM PhanCa  WHERE MaNhanVien = @MaNhanVien AND MaCa = @MaCa;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Rollback transaction and raise error message  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = N'Lỗi trong quá trình xóa phân ca! ' + ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

-Code C#:

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {  cnt.Open();  using (SqlCommand cmd = new SqlCommand("sp\_DeletePhanCa", cnt.GetConnection()))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNhanVien", txtMaNV.Text);  cmd.Parameters.AddWithValue("@MaCa", Maca);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  MessageBox.Show("Xóa phân ca thành công!");  }  else  {  MessageBox.Show("Không tìm thấy phân ca để xóa!");  }  }  }  catch (SqlException sqlEx)  {  MessageBox.Show(sqlEx.Message);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  finally  {  cnt.Close();  }  btnBack4\_Click(sender, e);  } |

### **2.37 Hàm thực hiện truy vấn**

**-Input: câu truy vấn gọi hàm, thủ tục**

**-Output: kết quả truy vấn SQL**

**-Code C#:**

|  |
| --- |
| public static DataTable ExecuteQuery(string query)  {  DataTable dataTable = new DataTable();  string connectionString = DataConnector.connectionString;  SqlConnection connection = new SqlConnection(connectionString);  try  {  connection.Open();  SqlDataAdapter dataAdapter = new SqlDataAdapter(query, connection);  dataAdapter.Fill(dataTable);  }  catch(SqlException sqlex)  {  MessageBox.Show(sqlex.ToString());  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  finally { connection.Close(); }  return dataTable;  }  public static void ExecutingNonResult(string query)  {  string connectionString = DataConnector.connectionString;  SqlConnection connection = new SqlConnection(connectionString);  try  {  connection.Open();  SqlCommand command = new SqlCommand(query, connection);  command.ExecuteNonQuery();  }  catch(SqlException sqlex)  {  MessageBox.Show(sqlex.ToString());  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString());  }  finally  {  connection.Close();  }  } |

# **IV- TẠO USER VÀ PHÂN QUYỀN**

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

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

|  |
| --- |
| create ROLE NhanVien  GRANT SELECT, REFERENCES ON CaLamViec TO NhanVien;  GRANT SELECT, REFERENCES ON ChiTietHD TO NhanVien;  GRANT SELECT, REFERENCES ON DonNhapHang TO NhanVien;  GRANT SELECT, REFERENCES ON HoaDon TO NhanVien;  GRANT SELECT, REFERENCES ON KhachHang TO NhanVien;  GRANT SELECT, REFERENCES ON LinhKien TO NhanVien;  GRANT SELECT, REFERENCES ON NhaCungCap TO NhanVien;  GRANT SELECT, REFERENCES ON NhanVien TO NhanVien;  GRANT SELECT, REFERENCES ON NhapHang TO NhanVien;  GRANT SELECT, REFERENCES ON NhomLinhKien TO NhanVien;  GRANT SELECT, REFERENCES ON PhanCa TO NhanVien;  GRANT EXECUTE TO NhanVien  GRANT SELECT TO NhanVien  DENY EXECUTE ON [dbo].[AddNhaCungCap] to NhanVien;  DENY EXECUTE ON [dbo].[nv\_deleteNhanVien] to NhanVien;  DENY EXECUTE ON [dbo].[nv\_InsertNhanVien] to NhanVien;  DENY EXECUTE ON [dbo].[nv\_updateNhanVien] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_AddCaLamViec] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_AddLinhKien] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_DeletePhanCa] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_DeleteDonNhapHang] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_DeleteNhapHang] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_InsertNhapHang] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_InsertNhomLinhKien] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_InsertPhanCa] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_UpdateLinhKien] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_UpdateNhapHang] to NhanVien;  DENY EXECUTE ON [dbo].[sp\_UpdateNhomLinhKien] to NhanVien;  DENY EXECUTE ON [dbo].[UpdateNhaCungCap] to NhanVien; |

Trigger tạo tài khoản:

|  |
| --- |
| Create TRIGGER [dbo].[Trigger\_CreateSQLAccount] ON [dbo].[NhanVien]  AFTER INSERT  AS  DECLARE @userName nvarchar(30), @passWord nvarchar(10)  SELECT @userName=nl.MaNV, @passWord=nl.SDT  FROM inserted nl  BEGIN  DECLARE @sqlString nvarchar(2000), @macv nvarchar(10)  ----  SET @sqlString= 'CREATE LOGIN [' + @userName +'] WITH PASSWORD='''+  @passWord  +''', DEFAULT\_DATABASE=[LinhKienDienTu2], CHECK\_EXPIRATION=OFF,  CHECK\_POLICY=OFF'  EXEC (@sqlString)  ----  SET @sqlString= 'CREATE USER ' + @userName +' FOR LOGIN '+ @userName  EXEC (@sqlString)  ----  SET @sqlString = 'ALTER ROLE NhanVien ADD MEMBER ' + @userName;  EXEC (@sqlString)  END |