**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**HỌC PHẦN: THỰC TẬP CƠ SỞ DỮ LIỆU**

**Đề tài: Quản lý bán thuốc**

|  |  |
| --- | --- |
| Giáo viên hướng dẫn:  Họ và tên: | Chu Thị Hường  Hoàng Văn Việt  Phan Thanh Tùng  Lỗ Trung Hiếu  Nguyễn Văn Dũng |

**MỤC LỤC**

**Chương 1. Đặc tả hệ thống.**

* 1. Mục đích yêu cầu.
  2. Mô tả hệ thống.

1.2.1. Quy trình nghiệp vụ.

1.2.2. Quy tắc nghiệp vụ.

* 1. Đặc tả các chức năng cần xây dựng.

**Chương 2. Phân tích dữ liệu hệ thống.**

2.1. Mô tả hệ thống dưới dạng mối quan hệ dữ liệu.

* 1. Mô tả cụ thể từng mối quan hệ.
  2. Mô hình liên kết thực thể.
  3. Cách chuyển đổi mô hình liên kết thực thể sang mô hình quan hệ.
  4. Phân tích nhu cầu sử dụng thông tin.
  5. Đặc tả bảng dữ liệu.
  6. Cách đặt các loại mã

.

**Chương 1: ĐẶC TẢ HỆ THỐNG**

* 1. **Mục đích, yêu cầu.**

**1.1.1. Mục đích:**

Xây dựng hệ thống quản lý bán thuốc tại một cửa hàng thuốc nhằm quản lý hoạt động, nâng cao chất lượng dịch vụ và đảm bảo tính chặt chẽ trong việc nhập thuốc từ các nhà cung cấp, với các lô thuốc được cung cấp bởi các hãng sản xuất khác nhau, kiểm kê số lượng thuốc còn hoặc hết trong kho, cập nhật thuốc vào kho và bán thuốc đúng theo nhu cầu người mua hoặc bán theo đơn thuốc của người mua đã có trước đó.

**1.1.2. Yêu cầu:**

* Nhân viên phải cung cấp đầy đủ và nắm chắc thông tin về thuốc: công dụng, thành phần thuốc, cách sử dụng, tác dụng phụ của thuốc,...
* Các phiếu nhập, xuất phải đưa ra được các thông tin như: mã phiếu, mã thuốc, tên thuốc, số lượng, đơn giá,... một cách rõ ràng.
* Phải xác định đúng đắn tình trạng lô thuốc, thuốc trong kho thuốc còn hoặc hết và hạn sử dụng của thuốc.
* Kịp thời bổ sung, cập nhật lại các thuốc trong kho.
* Đảm bảo được tính linh động, mạch lạc và quy cách trong việc hoạt động của nhà thuốc, giải quyết tốt việc chồng chéo quy trình nghiệp vụ.
* Có thể linh động cho khách hàng trong việc đổi các loại thuốc có cùng công dụng của các nhà sản xuất thuốc để phù hợp với túi tiền của người mua.
  1. **Mô tả hệ thống.**

**1.2.1. Quy trình nghiệp vụ**

* Nhà sản xuất sản xuất các loại thuốc nhằm phục vụ cho y tế. Thuốc được xuất theo lô; mỗi lô sẽ có nhiều loại thuốc khác nhau phân phối đến các nhà cung cấp.
* Nhà cung cấp sẽ cung cấp thuốc đến các cơ sở bán thuốc thông qua phiếu nhập.
* Khi cơ sở bán thuốc có nhu cầu mua thuốc, một bộ phận nhân viên sẽ gửi phiếu nhập đến nhà cung cấp để mua thuốc, nhà cung cấp sẽ xem xét phiếu được gửi đến để cung cấp thuốc theo đúng yêu cầu, kèm theo hóa đơn (bao gồm: đơn giá từng loại thuốc và thành tiền), hóa đơn này được lưu lại phục vụ cho việc kiểm kê sau này.
* Khách hàng đến hỏi mua thuốc được nhân viên tư vấn rồi nhận thuốc. Nhân viên giao thuốc và xuất hóa đơn (phiếu xuất) cho khách hàng.
* Nhân viên có thể là người quản lí, giám sát nhân viên làm việc hay quản lí cơ sở, kiểm tra kho hàng hoặc nhân viên có thể là người bán hàng, tư vấn bán thuốc cho khách hàng và xuất hóa đơn cho khách hàng (*nhân viên quản lý nhân viên*).

**1.2.2. Quy tắc nghiệp vụ**

Mô hình tổ chức: Nhà thuốc gồm có nhà cung cấp thuốc, hãng sản xuất thuốc, nhân viên (quản lý và bán hàng).

* + **Hãng sản xuất:**
* Sản xuất ra các lô thuốc.

***-* Nhà cung cấp thuốc:**

* Cung cấp thuốc cho nhà thuốc theo lô, phù hợp với yêu cầu của nhà thuốc.

**- Nhân viên:**

* *Quản lý:*
* Tổng hợp thuốc theo danh sách từng loại thuốc.
* **Kiểm tra**: Theo định kỳ, số thuốc còn lưu trữ, lô thuốc cũ hay mới, hạn sử dụng và chất lượng thuốc trong kho cần được người quản lý kiểm kê.
* **Nhập thuốc**: Liên hệ với người đại diện của nhà cung cấp để thực hiện việc nhập thuốc. Cần có phiếu nhập, được cập nhật thường xuyên trên hệ thống danh sách với đầy đủ các thông tin: Ngày nhập, lô nhập, số lượng thuốc nhập, đơn giá nhập.
* *Bán thuốc:*
* Tư vấn cho khách hàng đầy đủ các thông tin về loại thuốc: Liều lượng sử dụng, tác dụng, giá tiền; cũng như giải đáp các câu hỏi của khách hàng về từng loại thuốc.
* Xuất hóa đơn bán thuốc: Nhân viên bán thuốc cho khách hàng cần cung cấp hóa đơn với đầy đủ thông tin: ngày xuất, số lượng thuốc mua, đơn giá, chi tiết cách sử dụng từng loại thuốc, thông tin liên hệ nhà thuốc.
* Giao thuốc cho khách hàng.

**- Khách hàng:**

* Sử dụng thuốc theo hướng dẫn của nhân viên nhà thuốc để đảm bảo thuốc có tác dụng tốt nhất.
* Liên hệ ngay với nhà thuốc nếu việc sử dụng thuốc có vấn đề do ảnh hưởng của tác dụng phụ.
* Có nhu cầu thông tin về chất lượng dịch vụ.
  1. **Đặc tả chức năng.**
* Chức năng: **Nhập thuốc**

Nhân viên lập phiếu nhập và gửi phiếu nhập đến nhà cung cấp để yêu cầu nhập thuốc. Nhà cung cấp tiếp nhận rồi xem xét tình trạng đơn thuốc xem có đủ loại và số lượng. Sau đó, chuyển thuốc đến nhà thuốc.

* Chức năng: **Bán thuốc**

Khách hàng đến mua thuốc, thông báo bệnh tình rồi được nhân viên tư vấn, lấy thuốc, xuất hóa đơn, thu tiền, hoàn trả tiền thừa và giao thuốc; đồng thời, hướng dẫn cách sử dụng thuốc cho khách.

* Chức năng: **Kiểm thuốc**

Theo định kì, nhân viên kiểm kê lại số lượng thuốc đã bán và số lượng thuốc còn đang lưu trữ trong kho để có kế hoạch nhập thuốc và điều chỉnh lại số lượng thuốc.

* Chức năng : **Cập nhật thuốc mới và số lượng thuốc tồn**

Sau khi kiểm thuốc trong kho, nếu loại thuốc nào đó không cần phải nhập thêm hoặc nếu thuốc đó còn tồn, thì thực hiện bán số thuốc cũ trước rồi mới bán đến thuốc mới nhập.

**Chương 2: Phân tích dữ liệu hệ thống**

**2.1. Mô tả hệ thống dưới dạng mối quan hệ dữ liệu.**

* + - Mỗi Hãng sản xuất có Mã hãng sản xuất duy nhất, có Tên hãng, Quốc gia. Hãng sẽ sản xuất ra các Lô thuốc có Mã lô duy nhất, có Ngày sản xuất và Hạn sử dụng.
    - Mỗi Lô thuốc bao gồm nhiều Thuốc (có nhiều loại). Mỗi Thuốc có Mã thuốc duy nhất, có Tên thuốc, Công dụng, Thành phần và Số lượng tồn.

*Số lượng tồn: số lượng thuốc còn lại trong kho đến thời điểm kiểm kê theo định kì.*

* + - Mỗi Loại thuốc có Mã loại duy nhất, có Tên loại. Mỗi Loại thuốc bao gồm nhiều Thuốc.
    - Mỗi Nhà cung cấp có Mã nhà cung cấp duy nhất, có Tên, CMND/TCCCD và Thông tin đại diện, cung cấp Thuốc cho Nhà thuốc thông qua Phiếu nhập.
    - Nhân viên có Mã nhân viên duy nhất, có tên Nhân viên, Số điện thoại và CMND/TCCCD. Nhân viên quản lí Nhân viên.
    - Nhân viên in Phiếu nhập gửi Nhà cung cấp. Mỗi Phiếu nhập có Mã phiếu nhập duy nhất. Một Nhân viên có thể viết nhiều Phiếu nhập, có Ngày nhập và một Nhà cung cấp có được nhiều Phiếu nhập. Phiếu nhập nhập thuốc với Số lượng và Đơn giá khác nhau.
    - Khách hàng có Mã khách hàng, Tên khách hàng, Số điện thoại và CMND/TCCCD.
    - Khi Khách hàng mua thuốc, Nhân viên in Phiếu xuất cho Khách hàng. Mỗi Phiếu xuất có Mã phiếu xuất duy nhất. Một Nhân viên có thể in nhiều Phiếu xuất, với Ngày xuất. Một Khách hàng có thể có nhiều Phiếu xuất. Phiếu xuất có số lượng thuốc và đơn giá khác nhau.

**2.2. Mô tả cụ thể từng mối quan hệ.**

* Thuốc bao gồm: Mã thuốc (khóa chính), Tên thuốc, Dạng thuốc, Công dụng, Thành phần, Số lượng tồn.

|  |
| --- |
| Thuốc |
| Mã thuốc  Tên thuốc  Dạng thuốc  Công dụng  Thành phần  Số lượng tồn |

* Loại thuốc bao gồm: Mã loại thuốc (khóa chính), Tên loại thuốc, Ghi chú.

|  |
| --- |
| Loại thuốc |
| Mã loại thuốc  Tên loại thuốc |

* Lô thuốc bao gồm: Mã lô thuốc (khóa chính), Ngày sản xuất, Hạn sử dụng.

|  |
| --- |
| Lô thuốc |
| Mã lô thuốc  Ngày sản xuất  Hạn sử dụng |

* Hãng sản xuất bao gồm: Mã hãng sản xuất (khóa chính), Tên hãng sản xuất, Quốc gia.

|  |
| --- |
| Hãng sản xuất |
| Mã hãng sản xuất  Tên hãng sản xuất  Quốc gia |

* Nhà cung cấp bao gồm: Mã nhà cung cấp (khóa chính), Tên nhà cung cấp, CMND/TCCCD, Thông tin người đại diện.

|  |
| --- |
| Nhà cung cấp |
| Mã nhà cung cấp  Tên nhà cung cấp  Số điện thoại liên hệ  Thông tin đại diện |

* Phiếu nhập bao gồm: Mã phiếu nhập (khóa chính), Ngày nhập.

|  |
| --- |
| Phiếu nhập |
| Mã phiếu nhập  Ngày nhập |

* Nhân viên bao gồm: Mã nhân viên (khóa chính), Tên nhân viên, Số điện thoại nhân viên, CMND/TCCCD.

|  |
| --- |
| Nhân viên |
| Mã nhân viên  Tên nhân viên  Số điện thoại NV  CMND/TCCCD |

* Phiếu xuất bao gồm: Mã phiếu xuất (khóa chính), Ngày xuất.

|  |
| --- |
| Phiếu xuất |
| Mã phiếu xuất  Ngày xuất |

* Khách hàng bao gồm: Mã khách hàng (khóa chính), Tên khách hàng, Số điện thoại khách hàng, CMND/TCCCD.

|  |
| --- |
| Khách hàng |
| Mã khách hàng  Tên khách hàng  Số điện thoại KH  CMND/TCCCD |

* Một loại thuốc thuốc gồm nhiều thuốc.

|  |
| --- |
| Thuốc |
| Mã thuốc  Tên thuốc  Dạng thuốc  Công dụng  Thành phần  Số lượng tồn |

|  |
| --- |
| Loại thuốc |
| Mã loại thuốc  Gồm  Tên loại thuốc |

1 n

* Thuốc được nhập bởi phiếu nhập, với đơn giá và số lượng được định ra ở thời điểm nhập.

Được nhập

|  |
| --- |
| Thuốc |
| Mã thuốc  Tên thuốc  Dạng thuốc  Công dụng  Thành phần  Số lượng tồn |

|  |
| --- |
| Phiếu nhập |
| Mã phiếu nhập  Ngày nhập |

n n

* Nhà cung cấp thuốc được nhận nhiều phiếu nhập.

|  |
| --- |
| Nhà cung cấp |
| Mã nhà cung cấp  có  Tên nhà cung cấp  CMND/TCCCD  Thông tin đại diện |

|  |
| --- |
| Phiếu nhập |
| Mã phiếu nhập  Ngày nhập |

1 n

* Hãng sản xuất sản xuất nhiều lô thuốc.

|  |
| --- |
| Lô thuốc |
| Mã lô thuốc  Ngày sản xuất  Hạn sử dụng |

|  |
| --- |
| Hãng sản xuất  Sản xuất |
| Mã hãng sản xuất  Tên hãng sản xuất  Quốc gia |

1 n

* Nhân viên viết nhiều phiếu nhập vào các ngày nhập thuốc.

|  |
| --- |
| Nhân viên |
| Mã nhân viên  Viết  Tên nhân viên  Số điện thoại NV  CMND/TCCCD |

|  |
| --- |
| Phiếu nhập |
| Mã phiếu nhập  Ngày nhập |

1 n

* Nhân viên viết nhiều phiếu xuất vào các ngày xuất thuốc.

|  |
| --- |
| Nhân viên |
| Mã nhân viên  Viết  Tên nhân viên  Số điện thoại NV  CMND/TCCCD |

|  |
| --- |
| Phiếu xuất |
| Mã phiếu xuất  Ngày xuất |

1 n

* Khách hàng có thể có nhiều phiếu xuất.

|  |
| --- |
| Khách hàng |
| Mã khách hàng  Có  Tên khách hàng  Số điện thoại KH  CMND/TCCCD |

|  |
| --- |
| Phiếu xuất |
| Mã phiếu xuất  Ngày xuất |

1 n

* Thuốc được xuất bởi phiếu xuất với số lượng và đơn giá ở thời điểm xuất.

Được xuất

|  |
| --- |
| Thuốc |
| Mã thuốc  Tên thuốc  Dạng thuốc  Công dụng  Thành phần  Số lượng tồn |

|  |
| --- |
| Phiếu xuất |
| Mã phiếu xuất  Ngày xuất |

n n

* Một lô thuốc gồm nhiều thuốc.

|  |
| --- |
| Thuốc |
| Mã thuốc  Tên thuốc  Gồm  Dạng thuốc  Công dụng  Thành phần  Số lượng tồn |

|  |
| --- |
| Lô thuốc |
| Mã lô thuốc  Ngày sản xuất  Hạn sử dụng |

n 1

**2.3. Mô hình liên kết thực thể**

Sản xuất

|  |
| --- |
| Hãng sản xuất |
| Mã hãng sản xuất  Tên hãng sản xuất  Quốc gia |

1

|  |
| --- |
| Loại thuốc |
| Mã loại thuốic  Tên loại thuốc |

n

1

|  |
| --- |
| Lô thuốc |
| Mã lô thuốc  Ngày sản xuất  Hạn sử dụng |

1

Gồm

|  |
| --- |
| Khách hàng |
| Mã khách hàng  Tên khách hàng  Số điện thoại KH  CMND/TCCCD |

n

Gồm

|  |
| --- |
| Thuốc  n |
| Mã thuốc  Tên thuốc  Dạng thuốc  Công dụng  n  Thành phần  Số lượng tồn |

1

|  |
| --- |
| Nhà cung cấp |
| Mã nhà cung cấp  Tên nhà cung cấp  CMND/TCCCD  Thông tin đại diện |

n

Được xuất

Có

1

n

Được nhập

|  |
| --- |
| Phiếu xuất |
| Mã phiếu xuất  Ngày xuất |

n

Có

|  |
| --- |
| Nhân viên  1 |
| Mã nhân viên  Tên nhân viên  Số điện thoại NV  CMND/TCCCD |

|  |
| --- |
| Phiếu nhập |
| Mã phiếu nhập  Ngày nhập |

n

1

n

n

n

Viết

Viết

* 1. **Cách chuyển đổi mô hình liên kết thực thể sang mô hình quan hệ.**

HANGSANXUAT (**MaHangSX**, TenHangSX, QuocGia)

NHACUNGCAP (**MaNhaCungCap**, TenNhaCungCap, CMND/TCCCD, ThongTinNguoiDaiDien)

LOAITHUOC (**MaLoaiThuoc**, TenLoaiThuoc)

LOTHUOC (**MaLoThuoc**, NgaySanXuat, HanSuDung)

THUOC (**MaThuoc**, TenThuoc, DangThuoc, CongDung, ThanhPhan, SoLuongTon)

PHIEUNHAP (**MaPhieuNhap**, NgayNhap)

PHIEUXUAT (**MaPhieuXuat**, NgayXuat)

NHANVIEN (**MaNV**, TenNV, SĐTNV, CMND/TCCCD)

KHACHHANG (**MaKH**, TenKH, SĐTKH, CMND/TCCCD)

**- Mối quan hệ 1-1:** (Chuyển khóa chính của quan hệ này sang làm khóa ngoại của quan hệ kia hoặc ngược lại - Nhập 2 kiểu thực thể và mối liên kết thành 1 quan hệ với khóa chính phù hợp.)

**- Mối quan hệ 1-n:** (Chuyển khóa chính của quan hệ cha sang làm khóa ngoại của quan hệ con.)

+ Hãng sản xuất (con) - Lô thuốc (cha):

LOTHUOC (**MaLoThuoc**, MaHangSX, NgaySanXuat, HanSuDung)

+ Lô thuốc (con) - Thuốc (cha):

THUOC (**MaThuoc**, MaLoThuoc, TenThuoc, DangThuoc, CongDung, ThanhPhan, SoLuongTon) (1)

+ Loại thuốc (con) - Thuốc (cha):

THUOC (**MaThuoc**, MaLoaiThuoc, TenThuoc, DangThuoc, CongDung, ThanhPhan, SoLuongTon) (2)

+ Nhà cung cấp (con) - Phiếu nhập (cha):

PHIEUNHAP (**MaPhieuNhap**, MaNhaCungCap, NgayNhap) (3)

+ Nhân viên (con) - Phiếu nhập (cha):

PHIEUNHAP (**MaPhieuNhap**, MaNV, NgayNhap) (4)

+ Nhân viên (con) - Phiếu xuất (cha):

PHIEUXUAT (**MaPhieuXuat**, MaNV, NgayXuat) (5)

+ Khách hàng (con) - Phiếu xuất (cha):

PHIEUXUAT (**MaPhieuXuat**, MaKH, NgayXuat) (6)

**- Mối quan hệ n-n:** (Chuyển mối liên kết thành quan hệ có thuộc tính là thuộc tính của mối liên kết, thêm vào các thuộc tính khóa chính của các quan hệ có liên quan, khóa chính quan hệ mới là thuộc tính mới thêm vào.)

+ Thuốc - Phiếu nhập:

CHITIETPHIEUNHAP (**MaThuoc**, **MaPhieuNhap**, SoLuongNhap, DonGiaNhap)

+ Thuốc - Phiếu xuất:

CHITIETPHIEUXUAT (**MaThuoc**, **MaPhieuXuat**, SoLuongXuat, DonGiaXuat)

**\* Mô hình quan hệ:**

**CHÚ** **THÍCH**:

- **MaLoThuoc**: Khóa chính

**-** *MaHangSX*: Khóa ngoại

|  |
| --- |
| **LOTHUOC** |
| **MaLoThuoc**  *MaHangSX*  NgaySanXuat  HanSuDung |

|  |
| --- |
| **HANGSANXUAT** |
| **MaHangSX**  TenHangSX  QuocGia |

|  |
| --- |
| **LOAITHUOC** |
| **MaLoaiThuoc**  TenLoaiThuoc |

|  |
| --- |
| **THUOC** |
| **MaThuoc**  *MaLoThuoc*  *MaLoaiThuoc*  TenThuoc  DangThuoc  CongDung  ThanhPhan  SoLuongTon |

|  |
| --- |
| **CHITIETPHIEUXUAT** |
| **MaThuoc**  **MaPhieuXuat**  SoLuongXuat  DonGiaXuat |

|  |
| --- |
| **CHITIETPHIEUNHAP** |
| **MaThuoc**  **MaPhieuNhap**  SoLuongNhap  DonGiaNhap |

|  |
| --- |
| **PHIEUXUAT** |
| **MaPhieuXuat**  *MaNV*  *MaKH*  NgayXuat |

|  |
| --- |
| **PHIEUNHAP** |
| **MaPhieuNhap**  *MaNhaCungCap*  *MaNV*  NgayNhap |

|  |
| --- |
| **NHANVIEN** |
| **MaNV**  TenNV  SDTNV  CMND/TCCCD |

|  |
| --- |
| **KHACHHANG** |
| **MaKH**  TenKH  SDTKH  CMND/TCCCD |

|  |
| --- |
| **NHACUNGCAP** |
| **MaNhaCungCap**  TenNhaCungCap  CMND/TCCCD  ThongTinNguoiDaiDien |

* 1. **Phân tích nhu cầu sử dụng thông tin.**
     1. **Khẳng định mối quan hệ thu được.**

HANGSANXUAT (**MaHangSX**, TenHangSX, QuocGia)

NHACUNGCAP (**MaNhaCungCap**, TenNhaCungCap, SDTNCC, ThongTinNguoiDaiDien)

LOAITHUOC (**MaLoaiThuoc**, TenLoaiThuoc)

LOTHUOC (**MaLoThuoc**, MaHangSX, NgaySanXuat, HanSuDung)

PHIEUNHAP (**MaPhieuNhap**, MaNhaCungCap, MaNV, NgayNhap) theo (3) (4)

PHIEUXUAT(**MaPhieuXuat**, MaNV, MaKH, NgayXuat) theo (5) (6)

NHANVIEN (**MaNV**, TenNV, SDTNV, CMND/TCCCD)

KHACHHANG (**MaKH**, TenKH, SDTKH, CMND/TCCCD)

CHITIETPHIEUNHAP (**MaThuoc**, **MaPhieuNhap**, SoLuongNhap, DonGiaNhap)

CHITIETPHIEUXUAT (**MaThuoc**, **MaPhieuXuat**, SoLuongXuat, DonGiaXuat)

THUOC (**MaThuoc**, MaLoThuoc, MaLoaiThuoc, TenThuoc, DangThuoc, CongDung, ThanhPhan, SoLuongTon) theo (1) (2)

* + 1. **Kiểm tra các dạng chuẩn.**

1. HANGSANXUAT (**MaHangSX**, TenHangSX, QuocGia)

Các phụ thuộc hàm: MaHangSX→TenHangSX,

MaHangSX →QuocGia.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaHangSXduy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaHangSX → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ HANGSANXUAT đã đạt dạng chuẩn 3NF.

1. NHACUNGCAP (**MaNhaCungCap**, TenNhaCungCap, SDTNCC, ThongTinNguoiDaiDien)

Các phụ thuộc hàm: MaNhaCungCap→TenNhaCungCap,

MaNhaCungCap →SDTNCC,

MaNhaCungCap→ ThongTinNguoiDaiDien.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaNhaCungCapduy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaNhaCungCap → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ NHACUNGCAP đã đạt dạng chuẩn 3NF.

1. LOAITHUOC (**MaLoaiThuoc**, TenLoaiThuoc)

Các phụ thuộc hàm: MaLoaiThuoc→TenLoaiThuoc.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaLoaiThuoc duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaLoaiThuoc → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ LOAITHUOC đã đạt dạng chuẩn 3NF.

1. LOTHUOC (**MaLoThuoc**, MaHangSX, NgaySanXuat, HanSuDung)

Các phụ thuộc hàm: MaLoThuoc → MaHangSX,

MaLoThuoc → NgaySanXuat,

MaLoThuoc → HanSuDung.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaLoThuoc duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaLoThuoc → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ LOTHUOC đã đạt dạng chuẩn 3NF.

1. PHIEUNHAP(**MaPhieuNhap**, MaNhaCungCap, MaNV, NgayNhap)

Các phụ thuộc hàm: MaPhieuNhap→MaNhaCungCap,

MaPhieuNhap→ MaNV,

MaPhieuNhap→NgayNhap.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaPhieuNhap duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaPhieuNhap → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ PHIEUNHAP đã đạt dạng chuẩn 3NF.

1. PHIEUXUAT (**MaPhieuXuat**, MaNV, MaKH, NgayXuat)

Các phụ thuộc hàm: MaPhieuXuat→MaNV,

MaPhieuXuat→ MaKH,

MaPhieuXuat→NgayXuat.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaPhieuXuat duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaPhieuXuat → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ PHIEUXUAT đã đạt dạng chuẩn 3NF.

1. NHANVIEN (**MaNV**, TenNV, SDTNV, SDTNV)

Các phụ thuộc hàm: MaNV→TenNV,

MaNV→SDTNV,

MaNV→NgaySinh.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaNV duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaNV → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ NHANVIEN đã đạt dạng chuẩn 3NF.

1. KHACHHANG (**MaKH**, TenKH, SDTKH, SDTKH)

Các phụ thuộc hàm: MaKH→TenKH,

MaKH→SDTKH,

MaKH→ DiaChi.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Quan hệ chỉ có một khoá chính MaKH duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaKH → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ KHACHHANG đã đạt dạng chuẩn 3NF.

1. CHITIETPHIEUNHAP (**MaThuoc**, **MaPhieuNhap**, SoLuongNhap, DonGiaNhap)

Do DonGiaNhap của mỗi thuốc sẽ phụ thuộc vào từng đợt (MaPhieuNhap) nên ta có phụ thuộc hàm MaThuoc,MaPhieuNhap→DonGiaNhap.

Các phụ thuộc hàm: MaThuoc,MaPhieuNhap→SoLuongNhap,

MaThuoc,MaPhieuNhap→DonGiaNhap.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Không tồn tại các thuộc tính không là khoá chỉ phụ thuộc vào một phần của khoá chính → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ CHITIETPHIEUNHAP đã đạt dạng chuẩn 3NF.

1. CHITIETPHIEUXUAT (**MaThuoc**, **MaPhieuXuat**, SoLuongXuat, DonGiaXuat)

Các phụ thuộc hàm: MaThuoc,MaPhieuXuat→SoLuongXuat,

MaThuoc,MaPhieuXuat→DonGiaXuat.

* Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.
* Không tồn tại các thuộc tính không là khoá chỉ phụ thuộc vào một phần của khoá chính → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ CHITIETPHIEUXUAT đã đạt dạng chuẩn 3NF.

1. THUOC (**MaThuoc**,MaLoThuoc,MaLoaiThuoc,TenThuoc,DangThuoc, CongDung, ThanhPhan, SoLuongTon)

Các phụ thuộc hàm: MaThuoc →MaLoThuoc,

MaThuoc →MaDongThuoc,

MaThuoc →TenThuoc,

MaThuoc →DangThuoc,

MaThuoc →CongDung,

MaThuoc →ThanhPhan,

MaThuoc →SoLuongTon.

* Do lô thuốc mới chỉ nhập về khi đã bán hết lô cũ lên Mã Lô của mỗi thuốc chỉ nhận một giá trị duy nhất tại một thời điểm →thuộc tính MaLoThuoc là thuộc tính đơn trị.

Quan hệ không chứa thuộc tính đa trị → Đạt chuẩn 1NF.

* Quan hệ chỉ có một khoá chính MaThuoc duy nhấtnên các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá chính MaThuoc → Đạt chuẩn 2NF.
* Do không tồn tại phụ thuộc hàm vào thuộc tính không phải là khoá → Đạt chuẩn 3NF.
* Quan hệ THUOC đã đạt dạng chuẩn 3NF.
  1. **Đặc tả dữ liệu.**

1. THUOC:

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaThuoc | Char[30] | Primary-key | Mã thuốc |
| MaLoThuoc | Char[15] | Foreign-key | Mã lô thuốc |
| MaLoaiThuoc | Char[15] | Foreign-key | Mã loại thuốc |
| TenThuoc | NvarChar[30] | Not null | Tên thuốc |
| DangThuoc | NvarChar[30] | Not null | Dạng thuốc (tuýp, viên nang , dung dịch) |
| CongDung | NvarChar[50] | Not null | Công dụng của thuốc |
| ThanhPhan | NvarChar[50] | Not null | Thành phần thuốc |
| SoLuongTon | Int | Not null | Số lượng tồn của thuốc sau khi được kiểm kê hàng tháng |

1. LOAITHUOC

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaLoaiThuoc | Char[15] | Primary-key | Mã loại thuốc |
| TenLoaiThuoc | NvarChar[30] | Not null | Tên loại thuốc |

1. LOTHUOC

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaLoThuoc | Char[15] | Primary-key | Mã lô thuốc |
| MaHangSanXuat | Char[30] | Not null | Tên lô thuốc |
| NgaySanXuat | Date | Not null | Ngày sản xuất |
| HanSuDung | Date | Not null | Hạn sử dụng của thuốc |

1. HANGSANXUAT

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaHangSX | Char[15] | Primary-key | Mã hãng sản xuất |
| TenHangSX | NvarChar[30] | Not null | Tên hãng sản xuất |
| QuocGia | NvarChar[30] | Not null | Quốc gia |

1. PHIEUNHAP

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaPhieunhap | Char[15] | Primary-key | Mã phiếu nhập |
| NgayNhap | Date | Not null | Ngày nhập |
| MaNhaCungCap | Char[15] | Foreign-key | Mã nhà cung cấp |
| MaNhanVien | Char[15] | Foreign-key | Mã nhân viên |

1. NHACUNGCAP

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaNhaCungCap | Char[15] | Primary-key | Mã nhà cung cấp |
| TenNhaCungCap | Nvarchar[30] | Not null | Tên nhà cung cấp |
| SDTNNC | Nvarchar[50] | Not null | Số điện thoại nhà cung cấp |
| ThongTinNguoiDaiDien | Nvarchar[50] | Not null | Thông tin người đại diện của nhà cung cấp |

1. NHANVIEN

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaNhanVien | Char[15] | Primary-key | Mã nhân viên |
| TenNhanVien | Nvarchar[40] | Not null | Tên nhân viên |
| SoDienThoai | String | Not null | Số điện thoại nhân viên |
| CMND/TCCCD | String | Not null | Số chứng minh thư hoặc thẻ căn cước công dân của nhân viên |

1. PHIEUXUAT

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaPhieuXuat | Char[15] | Primary-key | Mã phiếu xuất |
| NgayXuat | Date | Not null | Ngày xuất |
| MaKhachHang | Char[15] | Foreign-key | Mã khách hàng mua thuốc |
| MaNhanVien | Char[15] | Foreign-key | Mã nhân viên bán thuốc |

1. KHACHHANG

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaKhachHang | Char[15] | Primary-key | Mã khách hàng mua thuốc |
| TenKhachHang | Nvarchar[40] | Not null | Tên khách hàng mua thuốc |
| SoDienThoai | String | Not null | Số điện thoại của KH mua |
| CMND/TCCCD | Nvarchar[50] | Not null | Số chứng minh nhân dân hoặc thẻ căn cước công dân của khách hàng |

1. CHITIETPHIEUNHAP

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaPhieuNhap | Char[15] | Primary-key | Mã phiếu nhập |
| MaThuoc | Char[15] | Primary-key | Mã thuốc nhập |
| SoLuong | Int | Not null | Số lượng nhập |
| DonGia | Int | Not null | Đơn giá nhập |

1. CHITIETPHIEUXUAT

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả | Chú giải |
| MaPhieuXuat | Char[15] | Primary-key | Mã phiếu xuất  ( hóa đơn bán) |
| MaThuoc | Char[15] | Primary-key | Mã thuốc bán |
| SoLuong | Int | Not null | Số lượng bán |
| DonGia | Int | Not null | Đơn giá bán |

**2.7. Cách đặt các loại mã**

**\* HÃNG SẢN XUẤT:** Mỗi hãng sản xuất sẽ có một mã quy định chung

**VD**: Công ty dược phẩm Hoa Linh – có mã HSX: HL

Công ty TNHH Tâm Bình – có mã HSX: TB

**\* LÔ THUỐC:** gồm 2 phần viết cách nhau bởi dấu chấm.

+ Phần 1 (dãy số): mã hãng sản xuất.

+ Phần 2 (dãy số): dãy số ghi tháng năm sản xuất ra lô thuốc

**VD**: HL0320, TB0719

**\* LOẠI THUỐC:** gồm kí tự chữ cái in hoa được nhà thuốc quy định. **VD**: *thuốc Đông Y - DY*, ...

**\* THUỐC:**

+ Phần 1 (dãy số): mã lô thuốc, mã loại thuốc, mã dòng thuốc

+ Phần 2 (dãy kí tự): mã riêng của thuốc – mã định danh thuốc

VD: HL0320DYCHONGHO**132TB - 132TB là thuốc ho Prospan**

**\* NHÂN VIÊN:** mã nhân viên là mã định danh của nhân viên, được tạo thành bởi mã số CMND/Thẻ căn cước của nhân viên

**\* NHÀ CUNG CẤP:** mã định danh nhà cung cấp tự sinh ngẫu nhiên, nhưng chỉ có các lý tự số. **VD**: *001*, *002*, ...

**\* KHÁCH HÀNG: VD**: *001099026417****0359999574***

+ Phần 1 (dãy số): 5 số đầu trong SĐT khách hàng.

+ Phần 2 (dãy số): mã thẻ căn cước/CMND của khách hàng.

+ Phần 3 (dãy số): các số còn lại trong SĐT khách hàng.

**\* PHIẾU NHẬP:** Ngày nhập+tháng nhập+ mã nhân viên nhập.

**VD**: *20122019031099005432*

**\* PHIẾU XUẤT:** Ngày bán+tháng bán+năm bán+SĐT khách hàng

**VD**: *031020190359999574*

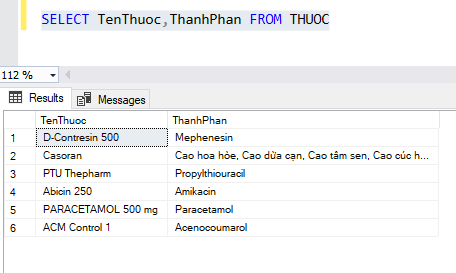
1. **Lý thuyết câu lệnh SELECT**
   1. Khái niệm:

* Lệnh SELECT là [lệnh SQL](https://quantrimang.com/13-cau-lenh-sql-quan-trong-programmer-nao-cung-can-biet-136595)  được sử dụng để lấy dữ liệu từ bảng rồi trả về dưới dạng bảng kết quả
  1. Cấu trúc:
     1. Cấu trúc truy vấn đơn giản

1. Truy vấn theo cột tùy ý

SELECT cột 1, cột 2, cột 3.... FROM Tên bảng WHERE <điều kiện>

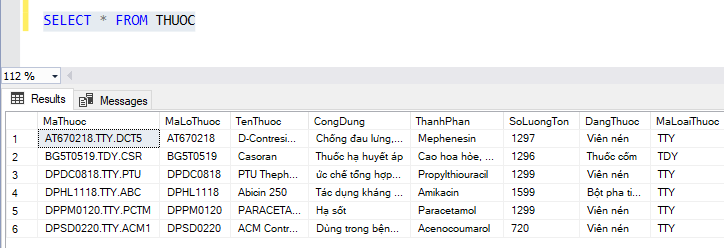
VD:



1. Truy vấn lấy toàn bộ trường dữ liệu

SELECT \* FROM Tên bảng

VD:



* + 1. Cấu trúc truy vấn tổng hợp nâng cao

**SELECT** [ALL | DISTINCT]

[TOP (giá trị đầu) [ PERCENT ] [ WITH TIES ]

Trường 1, trường 2, trường n,...

**FROM** Tên bảng

**WHERE** <điều kiện>

**GROUP BY** <Tên trường>

**HAVING** <điều kiện>

**ORDER BY** <Tên trường > [ **ASC | DESC** ]

**ALL** - trả lại tất cả các trường phù hợp.

**DISTINCT** - Loại bỏ tất cả các giá trị trùng lặp khỏi bộ kết quả.

**TOP** **(giá trị)** – Lấy số gía trị kết quả đầu. Ví dụ TOP(10) sẽ trả về 10 hàng đầu tiên trong bộ kết quả.

**PERCENT** - Nếu chỉ định cụ thể thì các hàng đầu tiên dựa theo phần trăm trên bộ kết quả (chỉ định bằng **giá trị**).

Ví dụ TOP(10) PERCENT sẽ trả về top 10% kết quả đầu tiên trong bộ kết quả.

**WITH TIES** - Nếu chỉ định cụ thể thì các hàng cố định ở cuối trong kết quả có giới hạn sẽ được trả về. Điều này có thể khiến cho số hàng trả về nhiều hơn biến TOP cho phép.

**Trường** – Trường hay giá trị tính toán mà bạn muốn lấy về. Dùng \* nếu muốn lấy tất cả các trường.

**Bảng** - Bảng mà bạn muốn lấy kết quả về. Phải có ít nhất 1 bảng được liệt kê trong lệnh FROM.

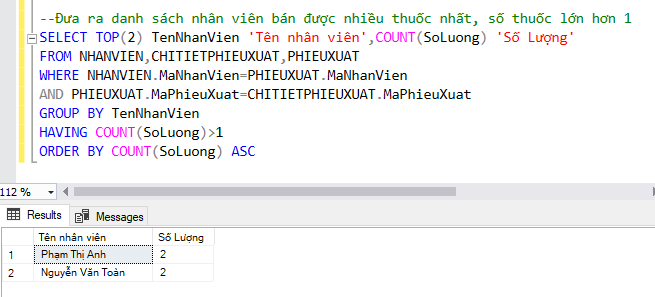
**WHERE “điều kiện”** - Điều kiện mà kết quả trả về phải đáp ứng được.

**GROUP BY “tên trường”** - Thu thập dữ liệu từ nhiều bản ghi và nhóm kết quả theo một hoặc nhiều trường.

**HAVING “điều kiện”** - Dùng kết hợp với**GROUP BY** để giới hạn nhóm các hàng trả về khi điều kiện được đáp ứng là TRUE.

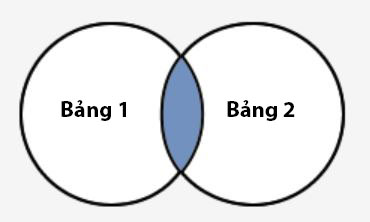
**ORDER BY “tên trường”** - Được dùng để sắp xếp bộ kết quả. **ASC** sẽ sắp xếp theo thứ tự tăng dần và **DESC** sẽ sắp xếp theo thứ tự giảm dần.

VD:



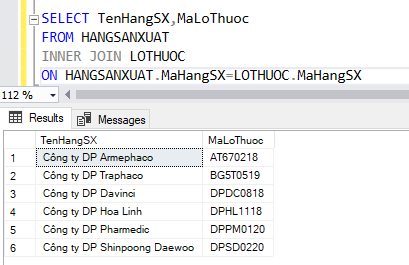
* + 1. Truy vấn có kết nối bảng
* Có 4 kiểu nối bảng

1. INNER JOIN: Trả về những bản ghi chung của các bảng dữ liệu

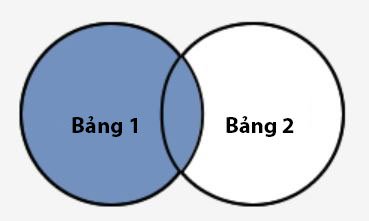


SELECT Tên trường  
FROM bảng 1  
INNER JOIN bảng 2  
ON bảng 1.trường = bảng 2.trường; <trường đều phải xuất hiện trong 2 bảng>

VD:



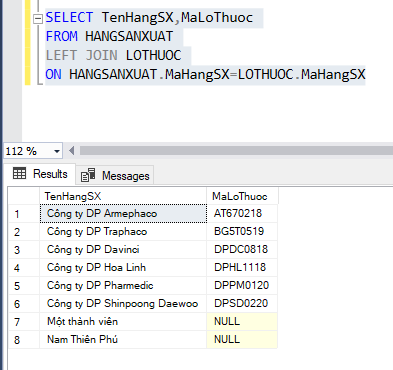
1. LEFT JOIN: trả về các trường nằm ở bảng bên trái trong điều kiện ON và chỉ lấy ở các bảng còn lại những hàng đáp ứng được điều kiện đặt ra



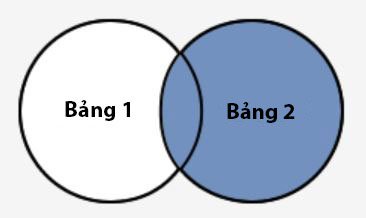
SELECT Tên trường  
FROM bảng 1  
LEFT JOIN bảng 2

ON bảng 1.trường = bảng 2.trường<trường đều phải xuất hiện trong 2 bảng>

VD:

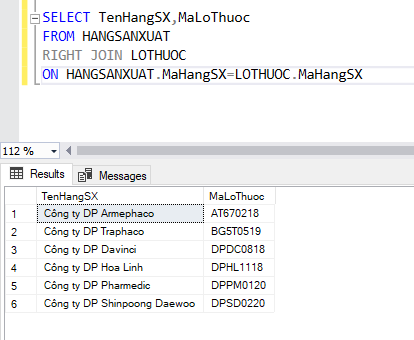


1. RIGHT JOIN: trả về các trường nằm ở bảng bên phải trong điều kiện ON và chỉ lấy ở các bảng còn lại những hàng đáp ứng được điều kiện đặt ra

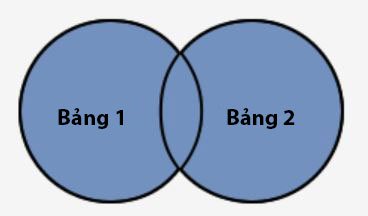


SELECT Tên trường  
FROM bảng 1  
RIGHT JOIN bảng 2  
ON bảng 1.trường = bảng 2.trường<trường đều phải xuất hiện trong 2 bảng>

VD:

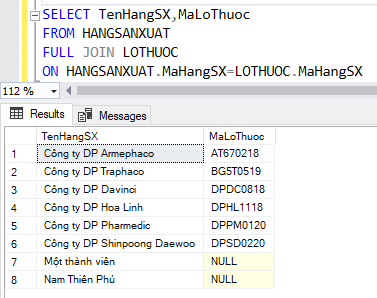


1. FULL JOIN: trả về tất cả các trường nằm ở bảng trái và phải và giá trị NULL đặt ở những nơi không đáp ứng được điều kiện.



SELECT Tên trường  
FROM bảng 1  
FULL JOIN bảng 2  
ON bảng 1.trường = bảng 2.trường<trường đều phải xuất hiện trong 2 bảng>

VD:



**1. Lý thuyết câu lệnh SELECT:**

**1.1. Khái niệm:**

Lệnh **SELECT** là [lệnh SQL](https://quantrimang.com/13-cau-lenh-sql-quan-trong-programmer-nao-cung-can-biet-136595) được sử dụng để lấy dữ liệu từ bảng rồi trả về dưới dạng bảng kết quả

**1.2. Cấu trúc:**

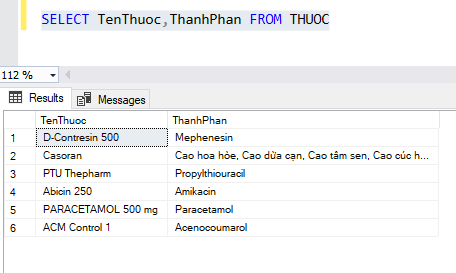
1.2.1. Cấu trúc truy vấn đơn giản:

a) Truy vấn theo cột tùy ý

SELECT cột\_1, cột\_2, cột\_3....

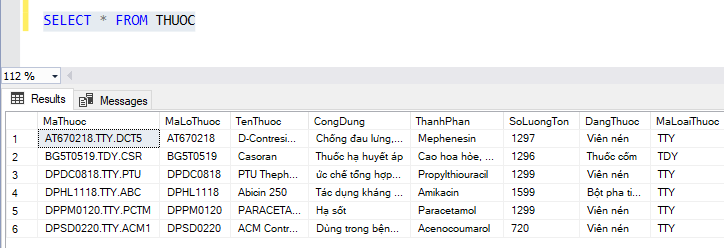
FROM tên\_bảng

WHERE <điều kiện>

**VD**: 

b) Truy vấn lấy toàn bộ trường dữ liệu:

SELECT \* FROM Tên bảng

**VD**: 

1.2.2. Cấu trúc truy vấn tổng hợp nâng cao:

**SELECT** [ALL | DISTINCT]

[TOP (giá trị đầu) [ PERCENT ] [ WITH TIES ]

Trường 1, trường 2, trường n,...

**FROM** Tên bảng

**WHERE** <điều kiện>

**GROUP BY** <Tên trường>

**HAVING** <điều kiện>

**ORDER BY** <Tên trường > [ **ASC | DESC** ]

**ALL** - trả lại tất cả các trường phù hợp.

**DISTINCT** - Loại bỏ tất cả các giá trị trùng lặp khỏi bộ kết quả.

**TOP** **(giá trị)** – Lấy số gía trị kết quả đầu. Ví dụ TOP(10) sẽ trả về 10 hàng đầu tiên trong bộ kết quả.

**PERCENT** - Nếu chỉ định cụ thể thì các hàng đầu tiên dựa theo phần trăm trên bộ kết quả (chỉ định bằng **giá trị**).

Ví dụ TOP(10) PERCENT sẽ trả về top 10% kết quả đầu tiên trong bộ kết quả.

**WITH TIES** - Nếu chỉ định cụ thể thì các hàng cố định ở cuối trong kết quả có giới hạn sẽ được trả về. Điều này có thể khiến cho số hàng trả về nhiều hơn biến TOP cho phép.

**Trường** – Trường hay giá trị tính toán mà bạn muốn lấy về. Dùng \* nếu muốn lấy tất cả các trường.

**Bảng** - Bảng mà bạn muốn lấy kết quả về. Phải có ít nhất 1 bảng được liệt kê trong lệnh FROM.

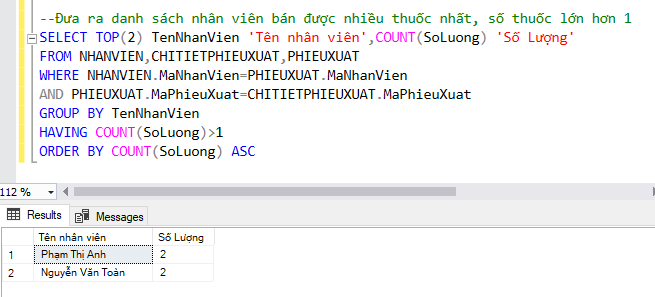
**WHERE “điều kiện”** - Điều kiện mà kết quả trả về phải đáp ứng được.

**GROUP BY “tên trường”** - Thu thập dữ liệu từ nhiều bản ghi và nhóm kết quả theo một hoặc nhiều trường.

**HAVING “điều kiện”** - Dùng kết hợp với**GROUP BY** để giới hạn nhóm các hàng trả về khi điều kiện được đáp ứng là TRUE.

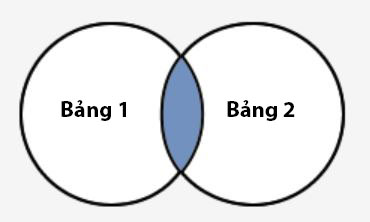
**ORDER BY “tên trường”** - Được dùng để sắp xếp bộ kết quả. **ASC** sẽ sắp xếp theo thứ tự tăng dần và **DESC** sẽ sắp xếp theo thứ tự giảm dần.

**VD**:



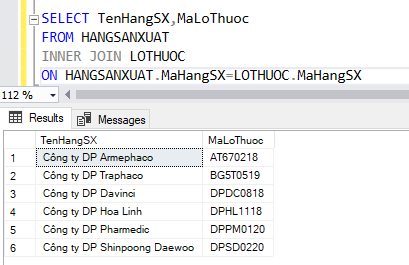
1.2.3. Truy vấn có kết nối bảng: Có **4 kiểu nối bảng**

a) **INNER JOIN**: Trả về những bản ghi chung của các bảng dữ liệu.

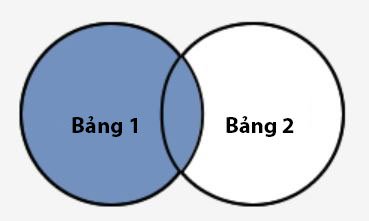


SELECT Tên trường  
FROM bảng 1  
INNER JOIN bảng 2  
ON bảng 1.trường = bảng 2.trường; <trường đều phải xuất hiện trong 2 bảng>

**VD**:



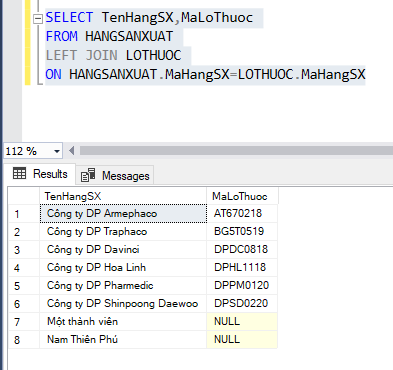
b) **LEFT JOIN**: trả về các trường nằm ở bảng bên trái trong điều kiện ON và chỉ lấy ở các bảng còn lại những hàng đáp ứng được điều kiện đặt ra.



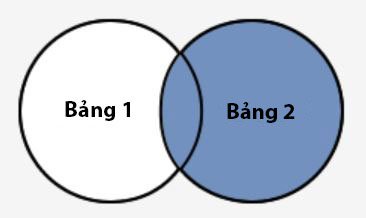
SELECT Tên trường  
FROM bảng 1  
LEFT JOIN bảng 2

ON bảng 1.trường = bảng 2.trường<trường đều phải xuất hiện trong 2 bảng>

**VD**:

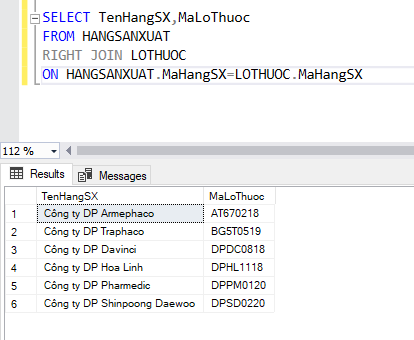


c) **RIGHT JOIN**: trả về các trường nằm ở bảng bên phải trong điều kiện ON và chỉ lấy ở các bảng còn lại những hàng đáp ứng được điều kiện đặt ra.

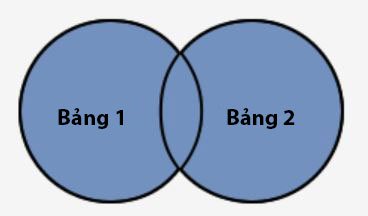


SELECT Tên trường  
FROM bảng 1  
RIGHT JOIN bảng 2  
ON bảng 1.trường = bảng 2.trường<trường đều phải xuất hiện trong 2 bảng>

**VD**:

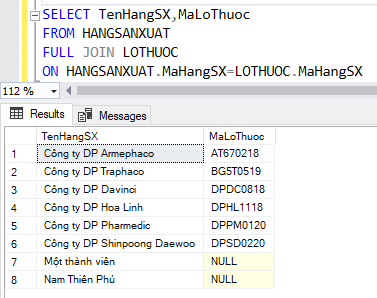


d) **FULL JOIN**: trả về tất cả các trường nằm ở bảng trái và phải và giá trị NULL đặt ở những nơi không đáp ứng được điều kiện.



SELECT Tên trường  
FROM bảng 1  
FULL JOIN bảng 2  
ON bảng 1.trường = bảng 2.trường<trường đều phải xuất hiện trong 2 bảng>

**VD**:



**NỘI DUNG BÁO CÁO TUẦN 7-8**

**Nhóm: 12 - Quản lý bán thuốc**

Thành viên:

- Hoàng Văn Việt.

- Phan Thanh Tùng.

- Lỗ Trung Hiếu.

- Nguyễn Văn Dũng.

**(I) PHẦN LÝ THUYẾT:**

**1. Lập trình Cấu trúc trong SQL Server:**

**1.1. Toán tử trong SQL.**

**a) Toán tử số học:**

SQL hỗ trợ 5 toán tử số học cơ bản là “ + (cộng) , - (Trừ) , \* (Nhân) , / (Chia) , % (Chia lấy dư) ”.

**b) Toán tử so sánh:**

* “ = ” trả về **true** khi 2 toán hạng **bằng** nhau.
* “ != ” trả về **true** khi 2 toán hạng **khác** nhau.
* “ <> ” trả về **true** khi 2 toán hạng **không bằng** nhau.
* “ > ” trả về **true** khi toán hạng **trái lớn hơn** toán hạng phải.
* “ < ” trả về **true** khi toán hạng **phải lớn hơn** toán hạng trái.
* “ >= ” trả về **true** khi toán hạng **trái lớn hơn hoặc bằng** toán hạng phải.
* “ <= ” trả về **true** khi toán hạng **phải lớn hơn** toán hạng trái.
* “ !< ” trả về **true** khi toán hạng **trái không nhỏ** **hơn** toán hạng phải.
* “ !> ” trả về **true** khi toán hạng **phải không nhỏ hơn** toán hạng trái.

**c) Toán tử logic:**

* “ALL” so sánh một giá trị với **tất cả** các giá trị trong tập hợp giá trị khác.
* “AND” cho phép nhiều **điều kiện cùng tồn tại** trong mệnh đề **WHERE** của lệnh SQL.
* “ANY” sử dụng để so sánh một giá trị với **bất kỳ** giá trị thích hợp nào trong danh sách tùy theo điều kiện.
* “BETWEEN” sử dụng để **tìm kiếm** giá trị **trong một tập hợp giá trị**, với giá trị nhỏ nhất và giá trị lớn nhất được cho trước.
* “EXISTS” sử dụng để **tìm kiếm sự có mặt** của một **hàng** trong bảng thỏa mãn điều kiện cụ thể.
* “IN” sử dụng để **so sánh giá trị với danh sách chuỗi** **giá trị** đã được xác định trước.
* “LIKE” sử dụng để **so sánh giá trị với các giá trị tương tự**, sử dụng các toán tử **Wildcard**.
* “NOT” Toán tử NOT **đảo ngược ý nghĩa** của toán tử logic mà nó được sử dụng.

**VD**: *NOT EXISTS*, *NOT BETWEEN*, *NOT IN*,...Đây là một toán tử phủ định.

* “OR” sử dụng để **nối nhiều điều kiện** trong mệnh đề **WHERE** của lệnh SQL.
* “IS NULL” sử dụng để **so sánh một giá trị** với giá trị **NULL**.
* “UNIQUE” Toán tử UNIQUE **tìm kiếm tính đơn nhất** cho mọi **hàng** trong bảng (không có bản sao).

**1.2. Biến trong T-SQL.**

Có **2** loại biến được sử dụng:

* Biến cục bộ
* Biến toàn cục

**a) Biến cục bộ:**

Giá trị của biến chỉ **tồn tại trong đoạn chương trình nhất định**.

**Cú pháp**: declare @<Tên biến> <Kiểu dữ liệu>

**VD**: declare @sum int

**Gán giá trị cho biến:**

**C1**: Gán khi khởi tạo

declare @<Tên biến> <Kiểu dữ liệu> = <Giá trị>

**VD**: declare @count int = 0

**C2**: Gán sau khởi tạo

set @<Tên biến> <Giá trị>

**VD**: declare @count int

Set @count = 0

**b) Biến toàn cục:**

Biến có **giá trị như nhau** tại **toàn bộ chương trình**.

Các biến này thường được **máy tự định nghĩa**.

**Cú pháp**: select @@<Tên biến >

Một số biến toán cục như: @@ERROR, @@VERSION, @@SERVERNAME ,…

**1.3. Cấu trúc lập trình.**

**a) Vòng lặp WHILE:**

Sử dụng để **lặp lại** 1 công việc trong khi **điều kiện cho trước** là **TRUE**.

**Cú pháp**:

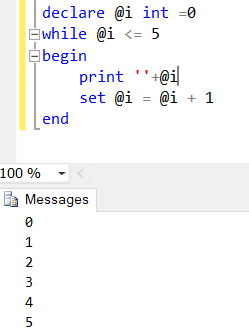
while <điều kiện>

begin

<tập lệnh lặp>

end

**VD**: Dùng vòng while để in ra 0 đến 5



**b) Lệnh If…Else**

Dùng thực thi các **lệnh có điều kiện**, nếu điều kiện **đúng** thì **thực thi lệnh đó**; nếu **không** thì **thực thi** 1 **lệnh khác**.

**Cú pháp**:

if <điều kiện>

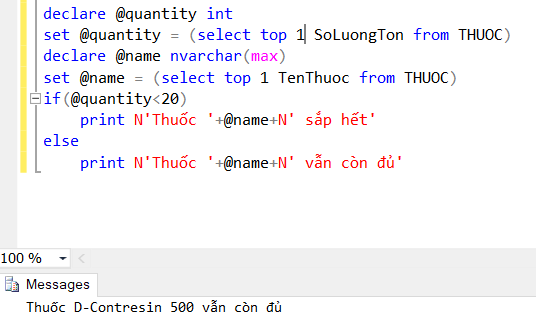
<tập lệnh>

else

<tập lệnh>

-> Vì SQL **không hỗ trợ ELSE IF** nên cần **dùng nhiều câu lệnh IF…ELSE** lồng nhau để thay thế.

**VD**: Dùng lệnh if-else để in ra tình trạng của 1 đầu thuốc



**c) Lệnh GOTO**

GOTO là một **lệnh nhảy đơn giản**, cho phép chương trình **nhảy vô điều kiện** từ GOTO tới **một vị trí trong chương trình** có **lệnh được gán nhãn** (*label*) trong cùng một hàm.

**Cú pháp**: GOTO label\_name;

Phần **nhãn** bao gồm **tên nhãn**(*label\_name*) và **câu lệnh** để thực hiện tiếp **sau**.

Lable\_name:{ đoạn lệnh }

**Chú ý**:

* label\_name phải là duy nhất trong phạm vi hàm.
* Phải có ít nhất một câu lệnh để thực thi sau khi khai báo nhãn.

**d) Hàm TRY\_CAST**

Hàm TRY\_CAST trong SQL Server **chuyển đổi một biểu thức** từ **một kiểu dữ liệu này** sang **kiểu dữ liệu khác**. Nếu chuyển đổi **không thành công**, TRY\_CAST sẽ trả về **NULL**, **ngược lại** nó sẽ trả về **giá trị chuyển đổi tương ứng**.

**Cú pháp**: TRY\_CAST (biểu thức AS kiểu dữ liệu [(độ dài dữ liệu)])

**e) Hàm CASE**

Câu lệnh CASE dùng để **thiết lập điều kiện rẽ nhánh** trong SQL Server, tương tự chức năng của câu lệnh IF-THEN-ELSE.

CASE có **2 định dạng**:

* Hàm CASE đơn giản, hay còn gọi là **Simple CASE**, là **so sánh** một **biểu thức** với một **bộ các biểu thức đơn giản** để xác định kết quả.
* Hàm CASE tìm kiếm, hay còn gọi là **Searched CASE**, là **đánh giá** một **bộ các biểu thức Boolean** để xác định kết quả.

**Cú pháp**:

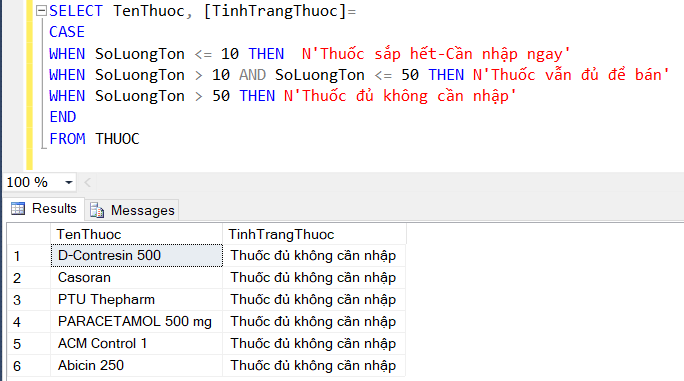
* **Simple CASE**

CASE bieuthuc\_dauvao  
  WHEN bieuthuc\_1 THEN ketqua\_1  
 WHEN bieuthuc\_2 THEN ketqua\_2  
 ...  
  WHEN bieuthuc\_n THEN ketqua\_n  
 ELSE ketqua\_khac

* **Searched CASE**

CASE  
  WHEN dieukien\_1 THEN ketqua\_1  
 WHEN dieukien\_2 THEN ketqua\_2  
  ...  
  WHEN dieukien\_n THEN ketqua\_n  
 ELSE ketqua\_khac

**VD**: Dùng hàm Searched CASE để in tình trạng các đầu thuốc.



**2. Store Procedure - Thủ tục:**

**2.1. Khái niệm:**

Stored procedure là tập hợp một hoặc nhiều câu lệnh T-SQL thành một nhóm đơn vị xử lý logic (chương trình con) và được lưu trữ trên Database Server. Khi một câu lệnh gọi chạy stored procedure lần đầu tiên thì SQL Server sẽ chạy nó và lưu trữ vào bộ nhớ đệm, gọi là **plan cache**, những lần tiếp theo SQL Server sẽ sử dụng lại stored procedure đã được biên dịch nên sẽ cho tốc độ xử lý nhanh.

**2.2. Các thao tác:**

|  |  |  |
| --- | --- | --- |
|  | **Không có tham số truyền vào** | **Có tham số truyền vào** |
| **Tạo** | CREATE PROC *Tên thủ tục*  AS  BEGIN  *Câu lệnh T-SQL*  END  **VD**:  CREATE PROC *Xem\_Thuoc*  AS  BEGIN  *SELECT \* FROM THUOC*  END | CREATE PROC *Tên thủ tục*  (@var1 *kiểu dữ liệu*, @varn *kiểu dữ liệu*)  AS  BEGIN  *Câu lệnh T-SQL*  END  **VD**:  CREATE PROC *Xem\_Thuoc (@maloaithuoc char(10) )*  AS  BEGIN  *SELECT \* FROM THUOC WHERE MaLoaiThuoc = @maloaithuoc*  END |
| **Sửa** | ALTER PROC *Tên thủ tục*  AS  BEGIN  *Câu lệnh T-SQL*  END  **VD**:  ALTER PROC *Xem\_Thuoc*  AS  BEGIN  *SELECT \* FROM THUOC WHERE MaLoaiThuoc = ‘TTY’*  END | ALTER PROC *Tên thủ tục*  (@var1 kiểu dữ liệu, @varn kiểu dữ liệu)  AS  BEGIN  *Câu lệnh T-SQL*  END  **VD**:  ALTER PROC *Xem\_Thuoc (@maloaithuoc char(10) )*  AS  BEGIN  *SELECT \* FROM THUOC WHERE MaLoaiThuoc = @maloaithuoc AND SoLuongTon > 400*  END |
| **Chạy** | EXECUTE *Tên thủ tục*  **VD**: EXECUTE *Xem\_Thuoc* | EXECUTE *Tên thủ tục* value1, value2,..., valuen  **VD**: EXECUTE  *Xem\_Thuoc ‘TTY’* |
| **Xóa** | DROP PROC *Tên thủ tục*  **VD**: DROP PROC *Xem\_Thuoc* | DROP PROC *Tên thủ tục*  **VD**: DROP PROC *Xem\_Thuoc* |

**Lưu ý**: có thể dùng PROCEDURE thay thế cho PROC hoặc ngược lại

**3. Function - Hàm:**

**3.1. Khái niệm:**

- Là đối tượng cơ sở dữ liệu tương tự như **Thủ tục** (*Stored Procedure*), với điểm khác biệt: **Hàm** trả về một giá trị thông qua **tên hàm** còn **Thủ tục** thì **không**.

- **Hàm** có thể được sử dụng như là một **thành phần** của một **biểu thức** (**VD**: *trong danh sách chọn của lệnh SELECT*).

- Người dùng có thể **định nghĩa** các **hàm** nhằm phục vụ cho **mục đích riêng** của mình.

- Hàm được chia làm **2** loại hàm **chính**:

+ **Hàm có sẵn** (Built-in Function).

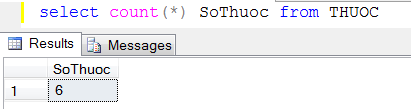
+ **Hàm do người dùng tự định nghĩa** (User-defined Function): được chia thành **2** loại hàm **nhỏ**:

+) **Scalar Functions**: hàm trả về một **giá trị với kiểu dữ liệu** được **khai báo** trong RETURNS lúc tạo hàm.

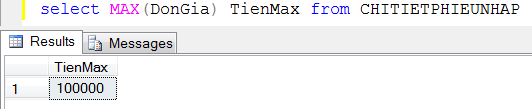
+) **Table-Valued Functions**: hàm trả về dữ liệu kiểu bảng chứa giá trị từ câu lệnh SELECT.

**3.2. Các hàm có sẵn (Function built-in):**

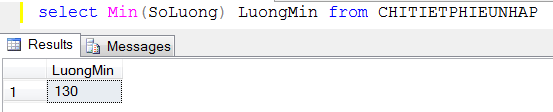
- hàm COUNT(\*): hàm được sử dụng để đưa ra **số lượng bản ghi** trong một **bảng** của CSDL.

**VD**: 

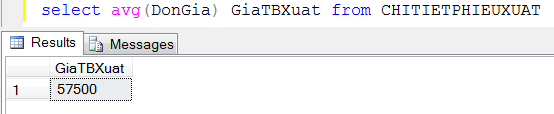
- hàm MAX(col\_name): hàm được sử dụng để đưa ra **giá trị lớn nhất** cho một **cột** cụ thể trong bảng.

**VD**: 

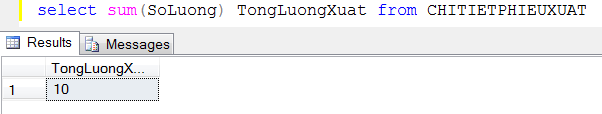
- hàm MIN(col\_name): hàm được sử dụng để đưa ra **giá trị nhỏ nhất** cho một **cột** cụ thể trong bảng.

**VD**: 

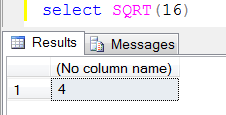
- hàm AVG(col\_name): hàm được sử dụng để đưa ra **giá trị trung bình** cho một **cột** cụ thể trong bảng.

**VD**: 

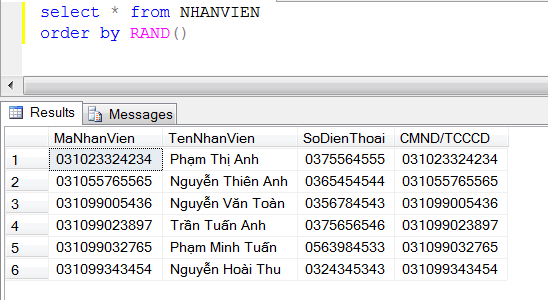
- hàm SUM(col\_name): hàm được sử dụng để đưa ra **giá trị tổng** cho một **cột** cụ thể trong bảng.

**VD**: 

- hàm SQRT(num): hàm được sử dụng để đưa ra **giá trị căn bậc hai** của một **số** đã cho.

**VD**: 

- hàm RAND(): hàm được sử dụng để đưa ra một **giá trị số ngẫu nhiên**.

**VD**: 

- hàm *xử lý* **DATE**: danh sách đầy đủ các hàm trong SQL để thao tác với **Date**.

+ GETDATE(): trả về 'date' và 'time' hiện tại.

+ DATEADD(datepart,number,date): cộng 'number' vào 'date'.

+ DATEDIFF(datepart,date1,date2): trừ 2 'date' cho nhau.

- hàm *xử lý* **NUMBER**: danh sách đầy đủ các hàm trong SQL để thao tác với **Number**.

+ ABS(num): trả về **giá trị tuyệt đối** của biểu thức số.

+ CEILING(num): trả về **giá trị nguyên nhỏ nhất** mà không nhỏ hơn biểu thức số đã truyền.

+ FLOOR(num): trả về **giá trị nguyên lớn nhất** mà không lớn hơn biểu thức số đã truyền.

+ POWER(num,y): trả về **giá trị hàm mũ** của 2 số (num^y).

+ ROUND(num,length): trả về biểu thức số đã được **làm tròn** về một số **nguyên**.

+ SIN(num): trả về **sin** của biểu thức số đã cho.

+ SQRT(float): trả về **căn bậc hai** của biểu thức số.

- hàm *xử lý* **STRING**: danh sách đầy đủ các hàm trong SQL để thao tác với **String**.

+ ASCII('string') với string = {0;255}: trả về **giá trị** của ký tự bên trái nhất của chuỗi. (= 0 nếu chuỗi trống, = NULL nếu chuỗi NULL)

+ LENGTH('string'): trả về **độ dài byte** của chuỗi.

+ CHAR\_LENGTH('string'): trả về **độ dài ký tự** của chuỗi.

+ LOWER('string'): trả về chuỗi **ở dạng chữ thường**.

**3.3. Tạo, thay đổi, xóa và chạy Hàm:**

|  |  |  |
| --- | --- | --- |
|  | **Scalar Functions** | **Table-valued Functions** |
| **Tạo** | CREATE FUNCTION <Tên function>  ([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], …,[...]])  RETURNS <kiểu dữ liệu>  [AS]  BEGIN  [Thân của hàm]  RETURN <Biểu thức giá trị đơn>  END  **VD:**  CREATE FUNCTION TONGTHUOC(@SLTHUOC INT) RETURNS INT  AS  BEGIN  DECLARE @DEM INT  SELECT @DEM = SUM(SoLuong)  FROM CHITIETPHIEUXUAT  RETURN (@DEM)  END | CREATE FUNCTION <Tên function>  ([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], …,[...]])  RETURNS @<tên biến trả về> TABLE (<tên cột i> <kiểu dữ liệu> [tùy chọn thuộc tính])  [AS]  BEGIN  <Câu lệnh SQL>  RETURN  END  **VD:**  CREATE FUNCTION XEMTHUOC(@SLTHUOC INT) RETURNS TABLE  AS RETURN  (SELECT A.MaLoThuoc, TenThuoc  FROM LOTHUOC A, THUOC B  WHERE A.MaLoThuoc = B.MaLoThuoc AND SoLuongTon > @SLTHUOC) |
| **Thay đổi** | ALTER FUNCTION <Tên function>  ([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], …,[...]])  RETURNS <kiểu dữ liệu> | TABLE  [AS]  BEGIN  [Thân của hàm]  RETURN <Biểu thức giá trị đơn> | Câu lệnh SQL  END  **VD:**  ALTER FUNCTION XEMTHUOC(@SLTHUOC INT) RETURNS TABLE  AS RETURN  (SELECT A.MaLoThuoc, TenThuoc  FROM LOTHUOC A, THUOC B  WHERE A.MaLoThuoc = B.MaLoThuoc AND SoLuongTon <= @SLTHUOC) | |
| **Xóa** | DROP FUNCTION [schema\_name.] <Tên function>  **VD:**  DROP FUNCTION XEMTHUOC | |
| **Chạy** | Có thể đặt **hàm** vào trong *phương thức* SELECT hoặc *phương thức* FROM.  **VD:**  SELECT \* FROM XEMTHUOC('500') | |

**4. Trigger:**

**4.1. Khái niệm:**

[Trigger](https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql)có thể hiểu nhanh là một công đoạn hẹn giờ thực thi SQL, được chạy từ Server không cần chờ phía client gọi hoặc là xảy ra ở phía server SAU KHI có các events Insert, Delete hay Update.

Tuy nhiên khác với **Store Procedure**, Trigger hoàn toàn không có tham số. Và không thể gọi thực hiện trực tiếp Trigger bằng lệnh EXECUTE(như Strore Procedure hay bất kì lệnh nào khác). Trigger chỉ có thể chạy một cách tự động. Chúng được lưu trữ trong CSDL của Server và thường hay được dùng để kiểm tra ràng buộc toàn vẹn của dữ liệu.

**4.2. Tạo và quản lý Trigger:**

**a) Sử dụng trigger:**

Câu lệnh tạo trigger: CREATE TRIGGER

Cấu trúc của trigger: CREATE TRIGGER ten\_trigger

ON doi\_tuong

FOR hanh\_dong

AS cau\_lenh

Tùy thuộc vào đối tượng và hành động, trigger có thể chia làm **hai loại**:

**Loại 1**: DDL Trigger

Sử dụng để định nghĩa cấu trúc dữ liệu với các cấu trúc table, view. Các hành động trong **DDL Trigger** như Create, Alter, Drop. Đối tượng thao tác thường là database hoặc all server.

**DDL Trigger** có thể xem như là một lớp “phòng thủ” khi dữ liệu các bảng bị thay đổi hoặc xóa.

**VD**: Tạo Trigger DDL cảnh báo về việc thay đổi cấu trúc dữ liệu bảng trong CSDL

Create Trigger ddl\_trigger

on database

for

create\_table, alter\_table, drop\_table

as

print N'Bạn không thể thay đổi cấu trúc dữ liệu'

rollback;

**Loại 2**: DML Trigger

Sử dụng để thao tác dữ liệu với các bản ghi của table, view. Các hành động trong **DML Trigger** như Insert (chèn dữ liệu), Update (cập nhật dữ liệu) và Delete (xóa dữ liệu). Trước các hành động, trong **DML Trigger** cần phải có thời điểm kích hoạt Trigger gồm **AFTER** và **INSTEAD OF**.

+) **After**: Kích hoạt **sau** khi hành động

**VD**:  Nếu bạn chèn bản ghi / hàng trong một bảng, sau đó **Trigger** liên kết với các trường hợp chèn vào bảng này, sẽ chỉ thực thi sau khi liên tiếp vượt qua tất cả những **Trigger**, như hạn chế chính là khóa chính, và một số quy tắc.

Nếu chèn bản ghi lỗi, SQL Server sẽ không thực thi After Trigger.

+) **Instead Of**: Kích hoạt **thay vì** hành động

**VD**:  Nếu bạn chèn bản ghi / hàng trong một bảng, sau đó **Trigger** liên kết với các sự kiện chèn vào bảng này, sẽ thực hiện vượt qua tất cả những điều kiện, chẳng hạn như khóa chính và một số quy tắc.

Nếu insert lỗi, SQL Server sẽ thực thi Trigger, tạo ra thông báo.

**DML Trigger** có thể được sử dụng như một cách để kiểm tra tính toàn vẹn của cơ sở dữ liệu, nó cũng có thể sử dụng hiệu quả trong công việc kiểm soát những thay đổi của dữ liệu trong bảng.

**VD**: Tạo Trigger cảnh báo khi số lượng thuốc tồn kho không đủ để xuất ra bán

create trigger insert\_chitietphieuxuat

on CHITIETPHIEUXUAT

after insert

as

begin

declare @a int;

set @a = (select SoLuongTon from THUOC where THUOC.MaThuoc in ( select MaThuoc from inserted)) ;

declare @b int;

set @b = (select SoLuong from inserted);

if (@a < @b)

begin

print N'Số lượng tồn không đủ';

rollback;

end

end

**4.3. Một số thao tác với Trigger:**

**a) Xóa trigger:**

Câu lệnh **DROP**:

Với **DML Trigger**: DROP TRIGGER ten\_trigger

Với **DDL Trigger**: DROP TRIGGER ten\_trigger

ON {DATABASE | ALL SERVER}

**b) Vô hiệu hóa trigger hoặc làm trigger có hiệu lực trở lại:**

Câu lệnh **DISABLE**:

DISABLE | ENABLE TRIGGER ten\_trigger

ON {object\_name | DATABASE | ALL SERVER}

**4.4. Ưu nhược điểm của trigger**

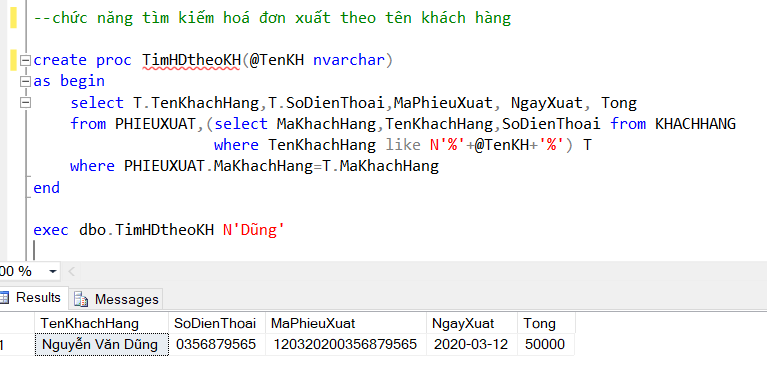
**a) Ưu điểm:**

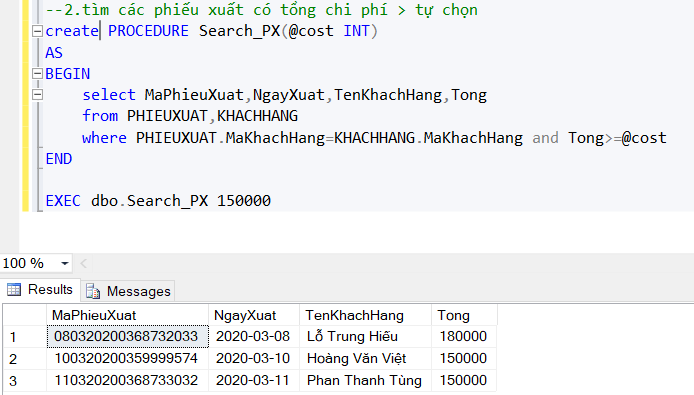
* Sử dụng Trigger để **kiểm tra tính toàn vẹn** của cơ sở dữ liệu.
* Trigger có thể **bắt lỗi logic** ở mức cơ sở dữ liệu.
* Có thể dùng trigger là một cách khác để **thay thế việc thực hiện những công việc hẹn giờ theo lịch**.
* Trigger rất **hiệu quả** khi sử dụng để **kiểm soát những thay đổi** của **dữ liệu** trong bảng.

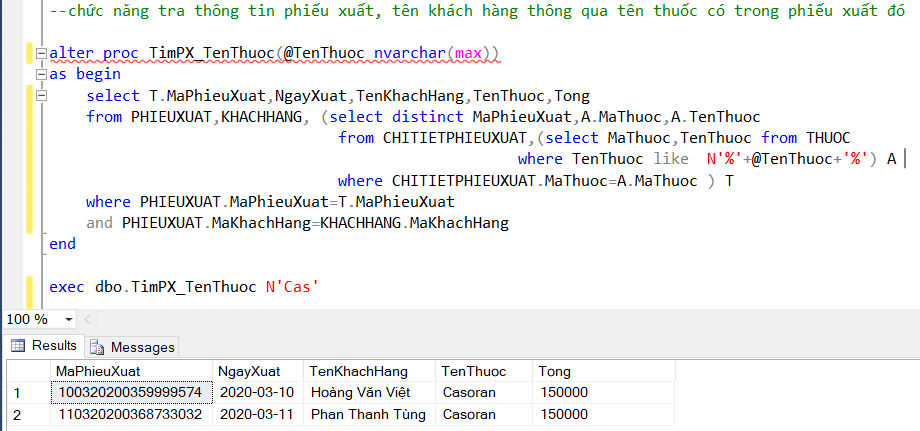
**b) Nhược điểm:**

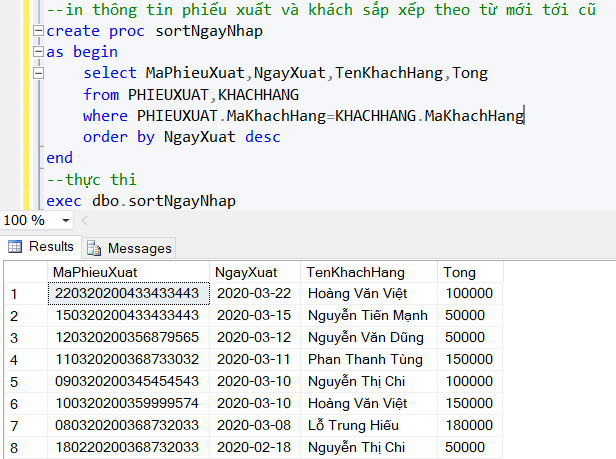
* Trigger chỉ là **một phần mở rộng** của việc kiểm tra tính hợp lệ của dữ liệu, không thể thay thế hoàn toàn được công việc này.
* Trigger **hoạt động ngầm** trong CSDL, không hiển thị ở tầng giao diện. Do đó, khó chỉ ra được điều gì xảy ra ở tầng csdl.
* Trigger thực hiện các update lên bảng dữ liệu vì thế nó làm **tăng lượng công việc** lên CSDL và làm cho hệ thống **chạy chậm**.

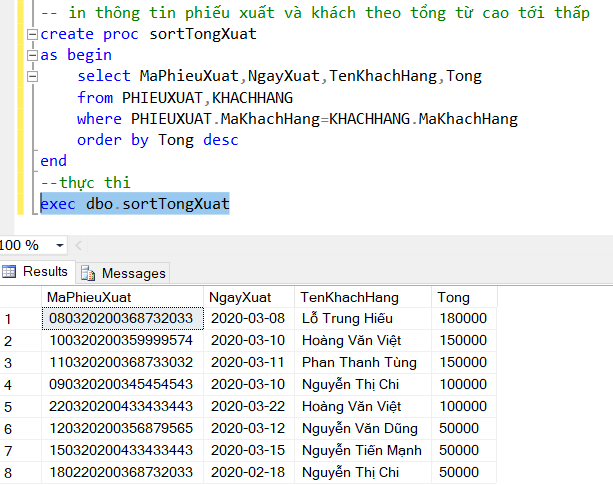
**(II) PHẦN THỰC HÀNH:** Gồm 47 câu lệnh truy vấn

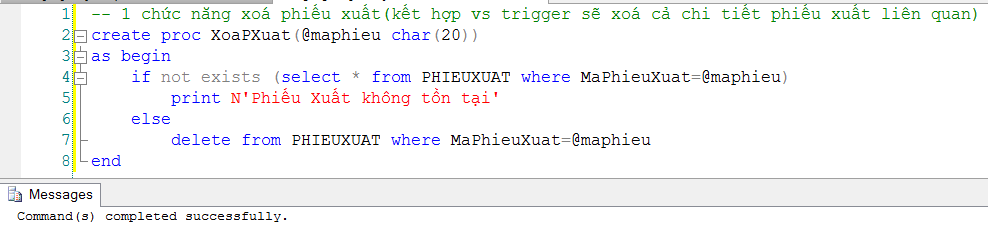


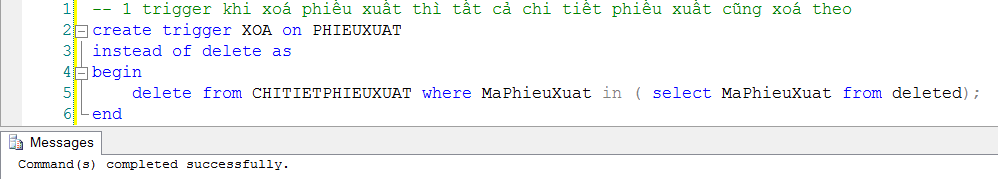


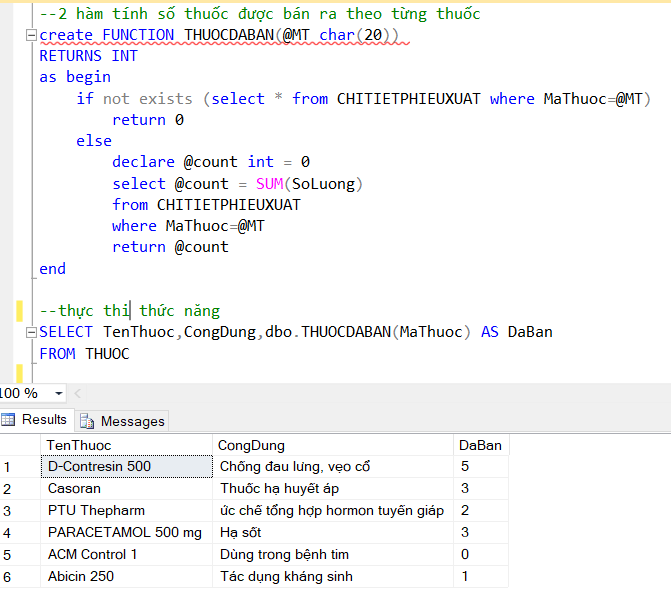


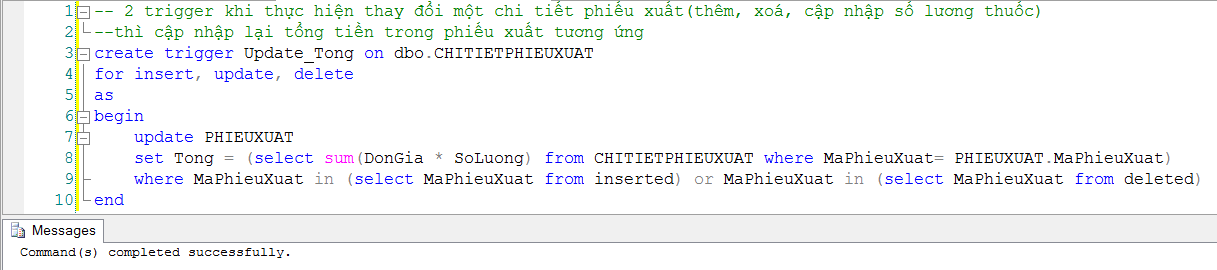


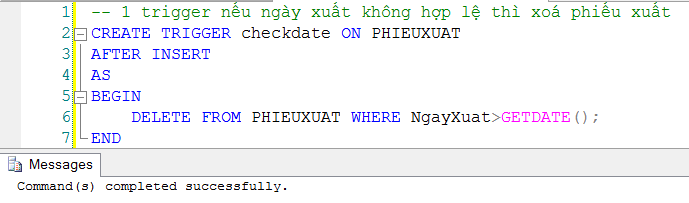


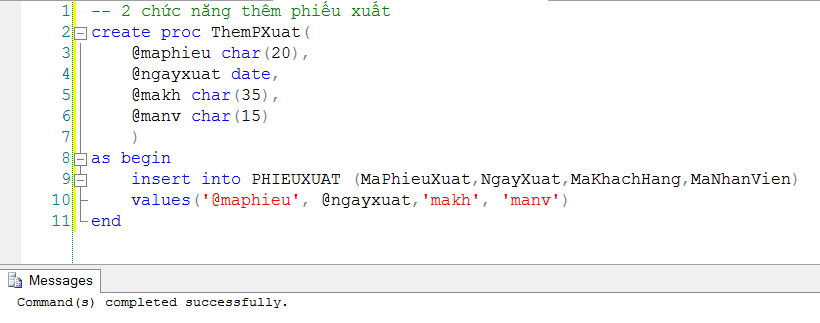


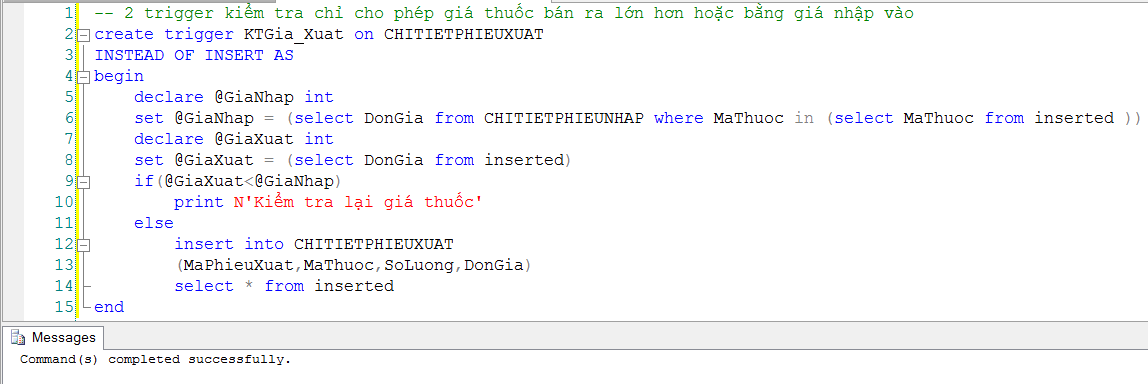


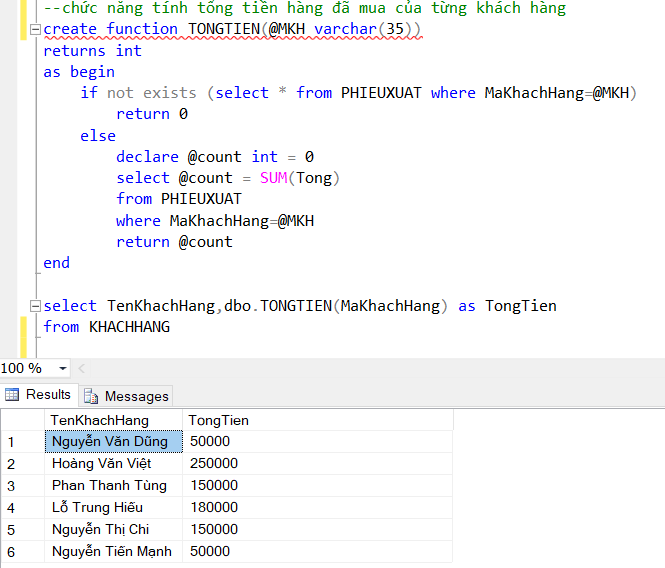


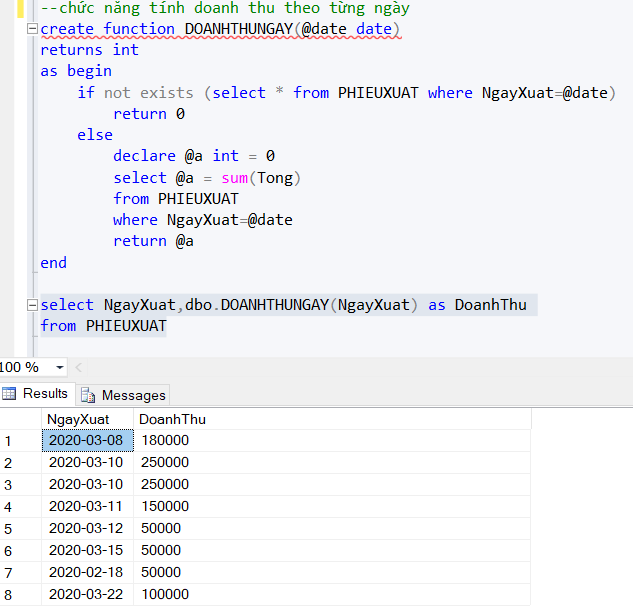
****

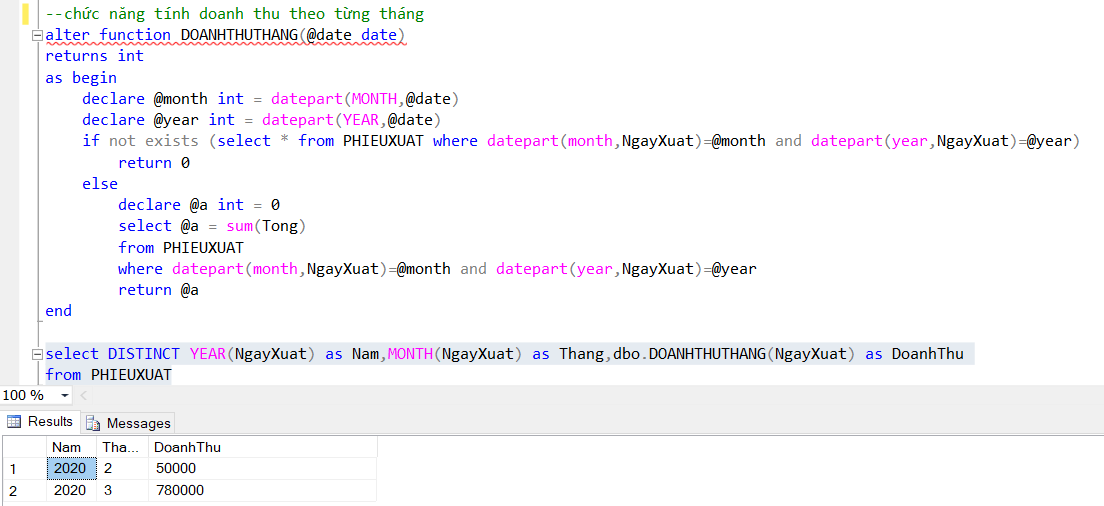


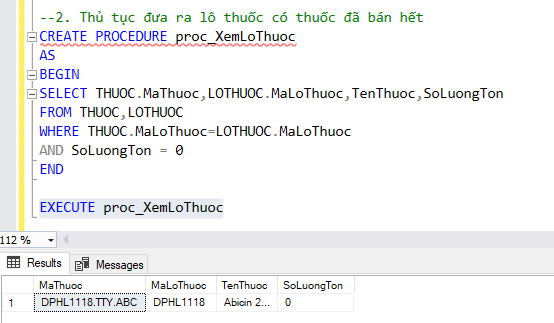
****

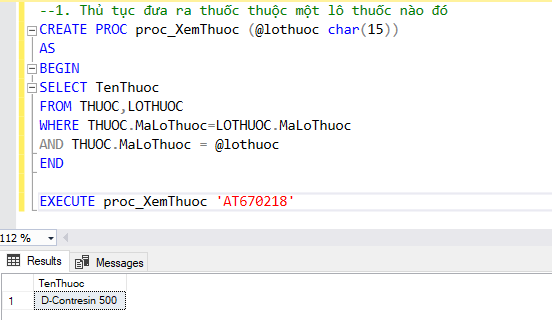


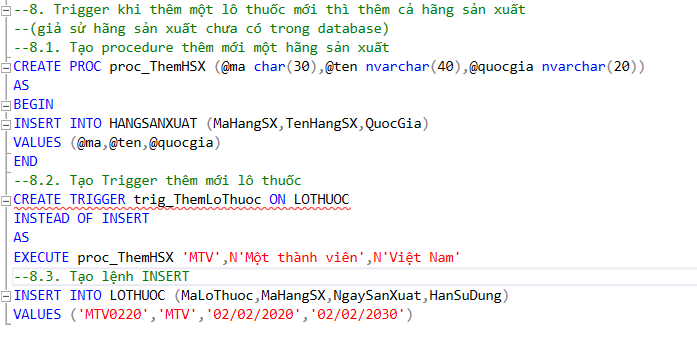


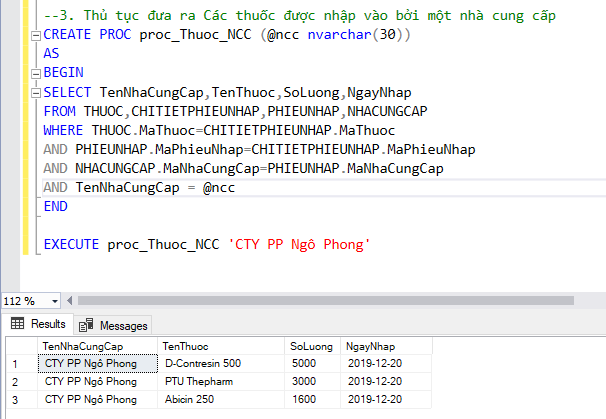


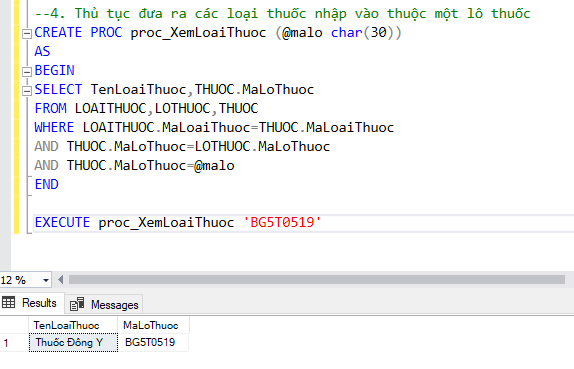


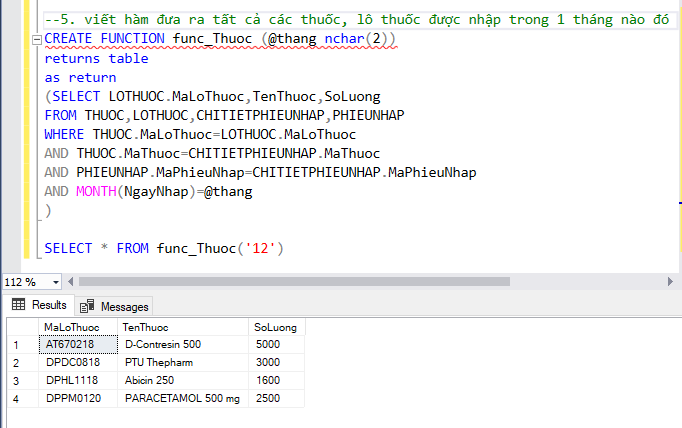


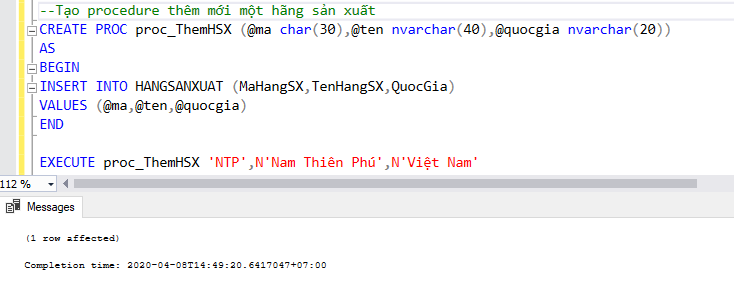


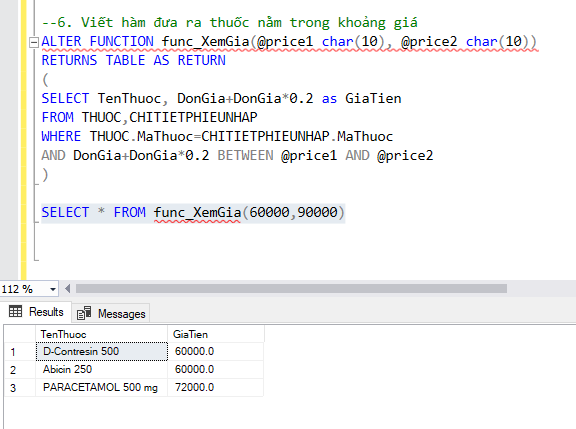


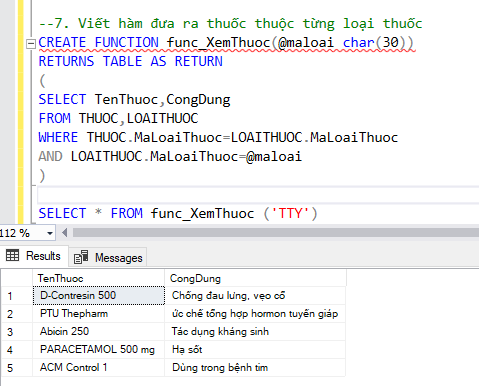


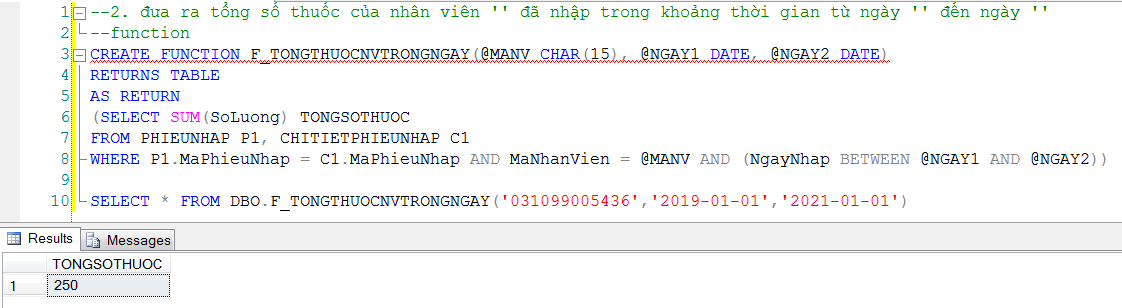


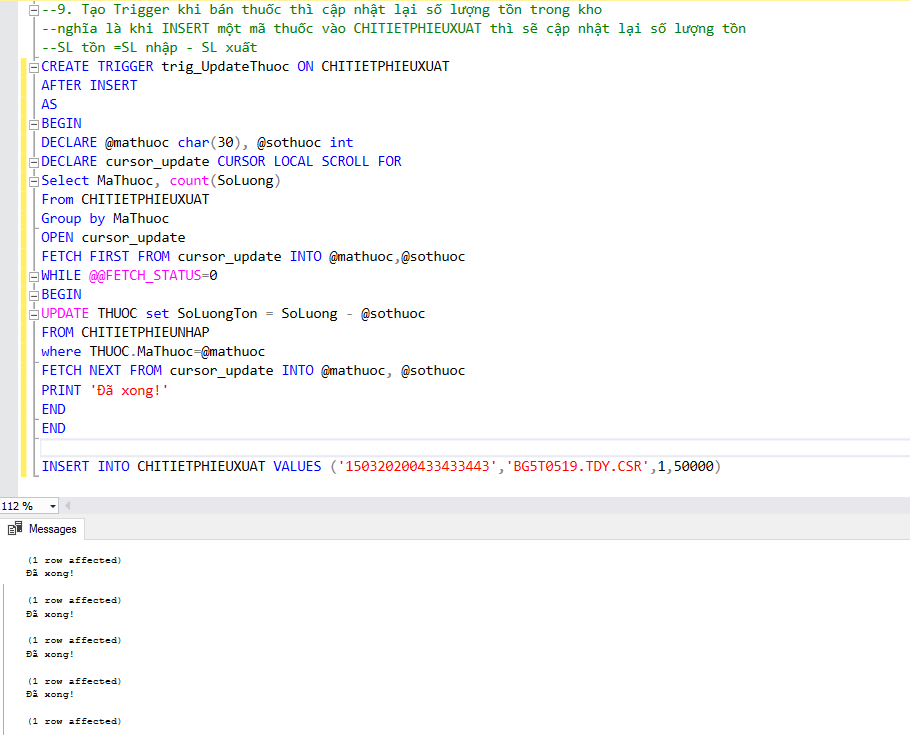


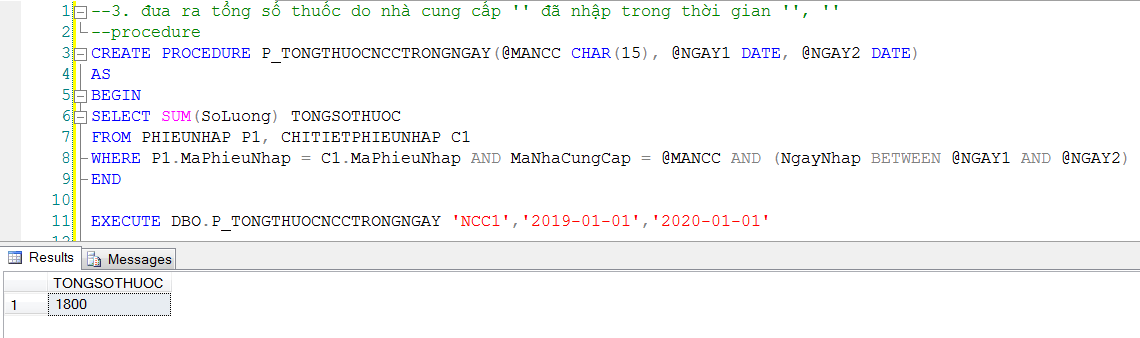


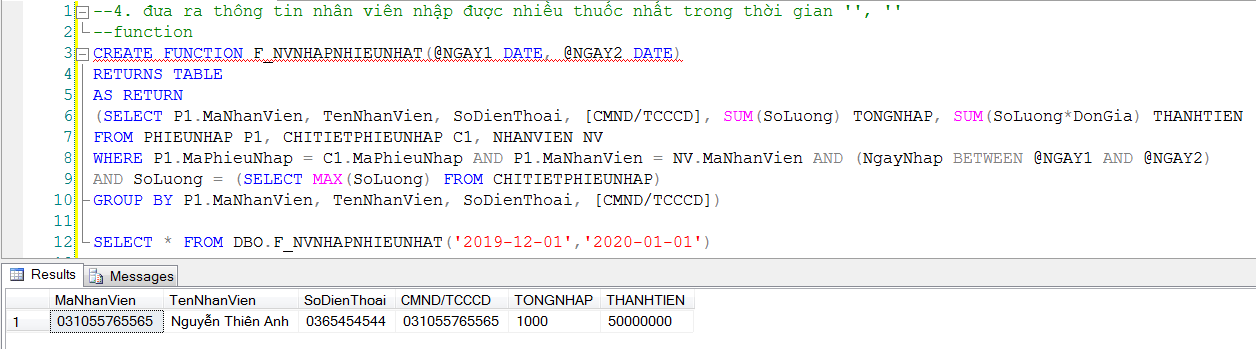


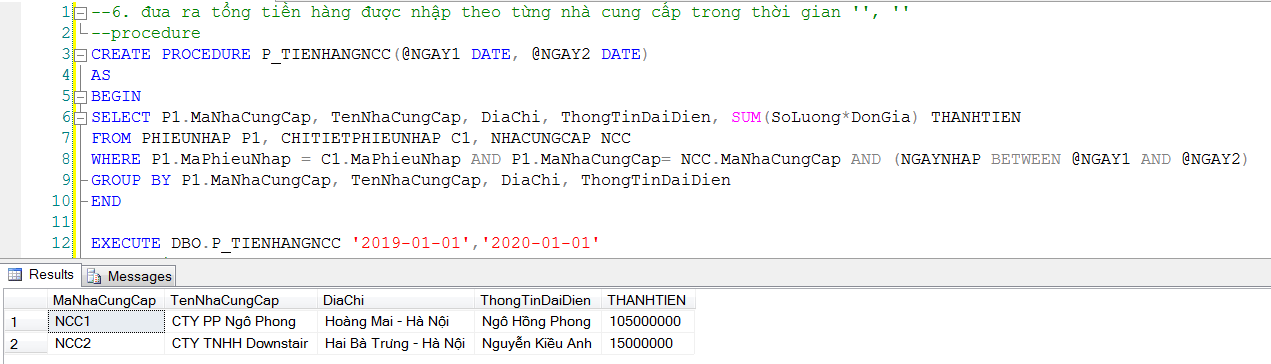


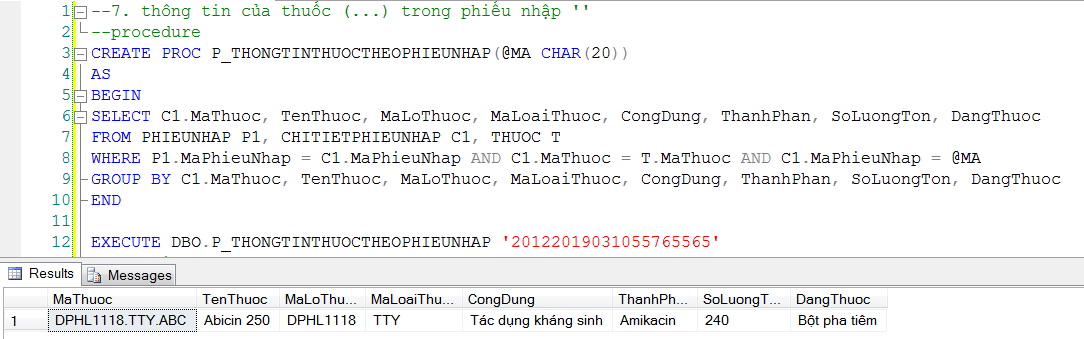
****

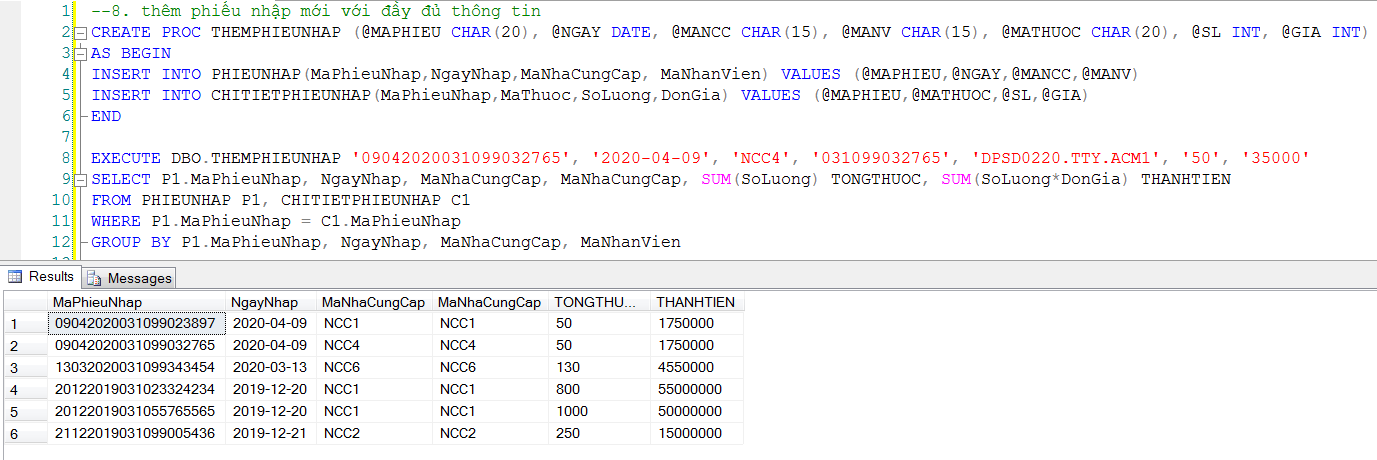


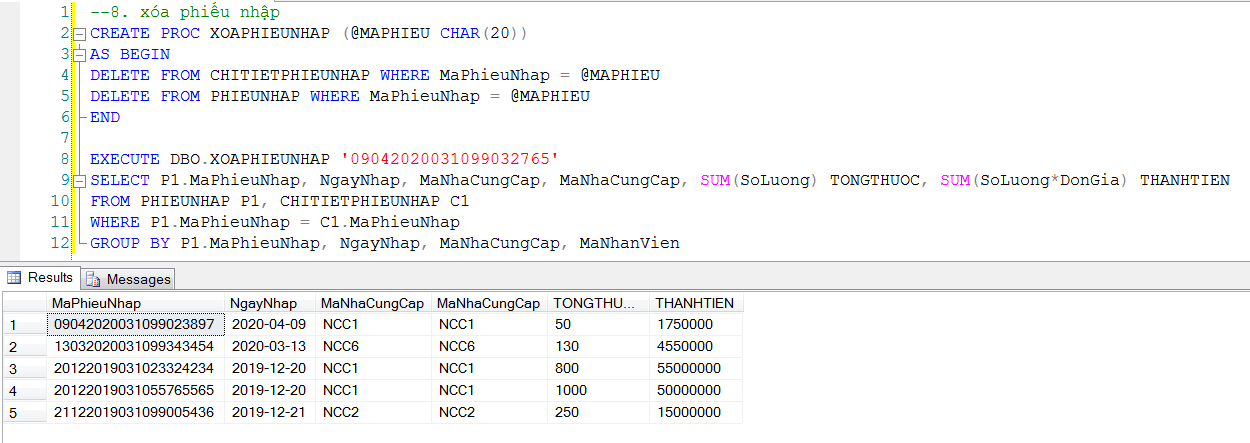
****

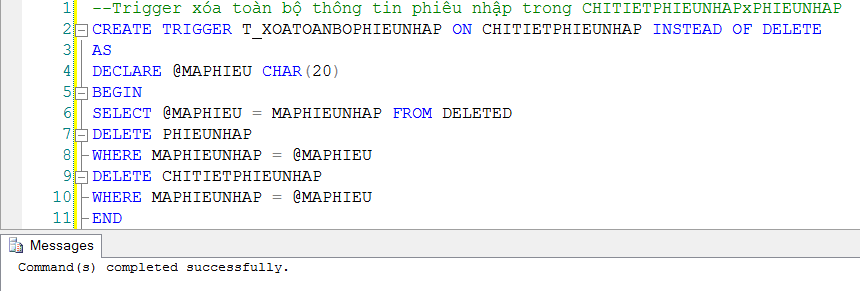
****

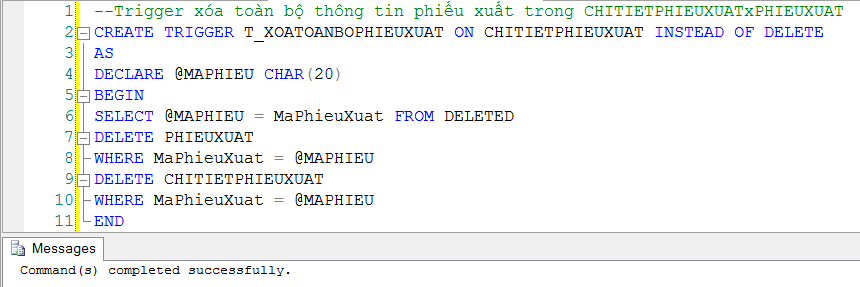
****

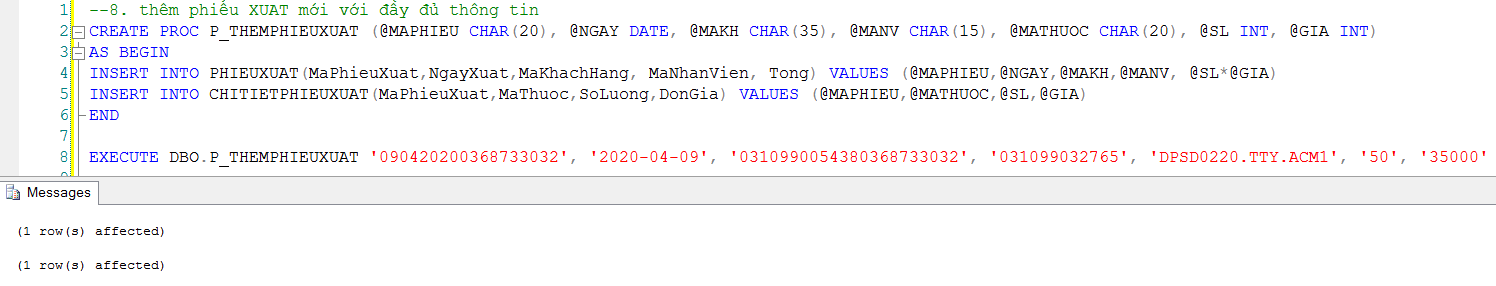
****

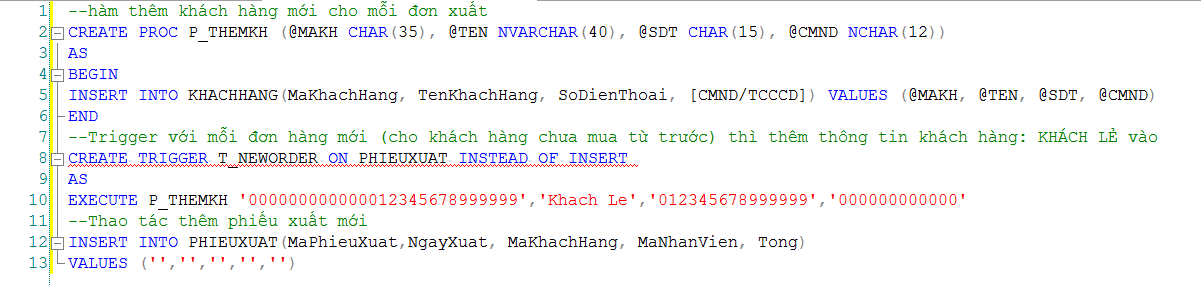
****

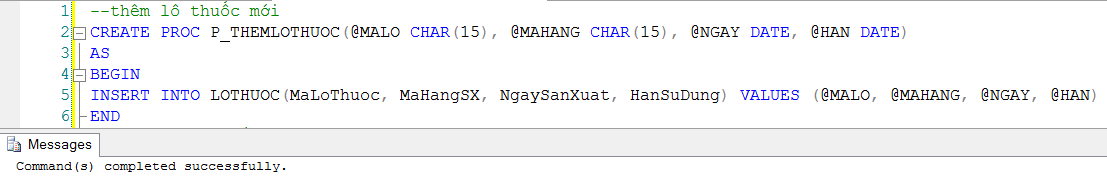
****

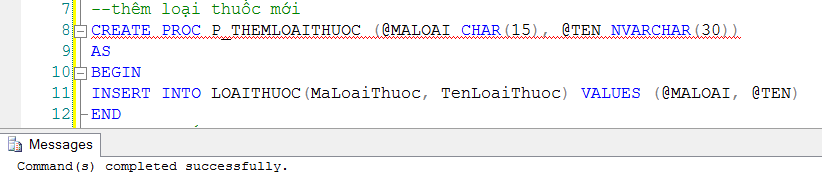
****

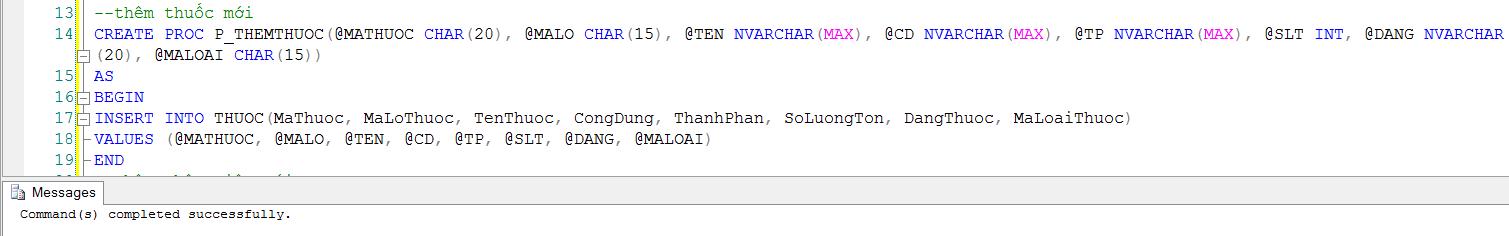
****

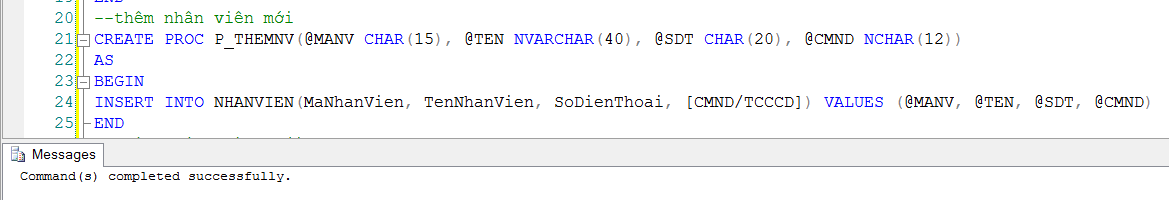
****

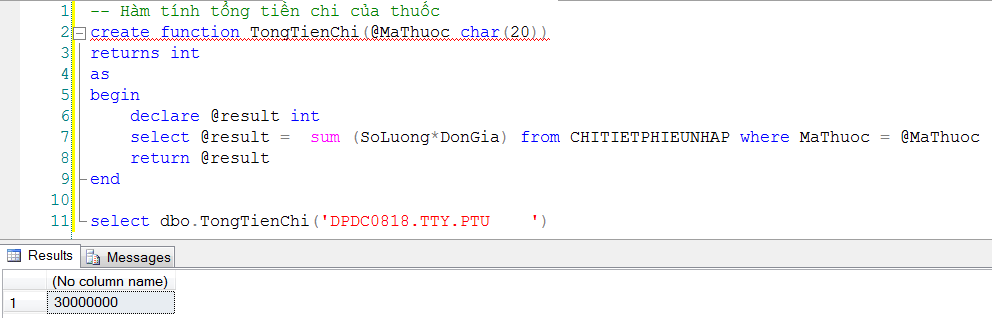
****

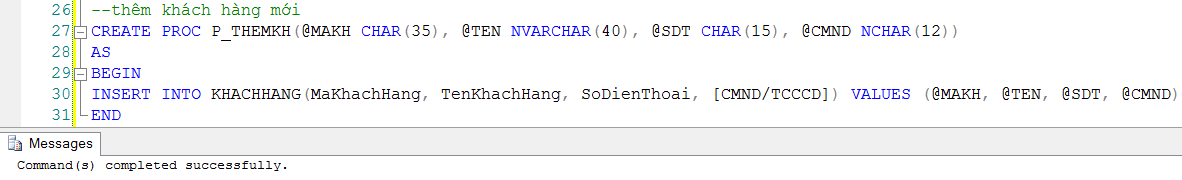
****

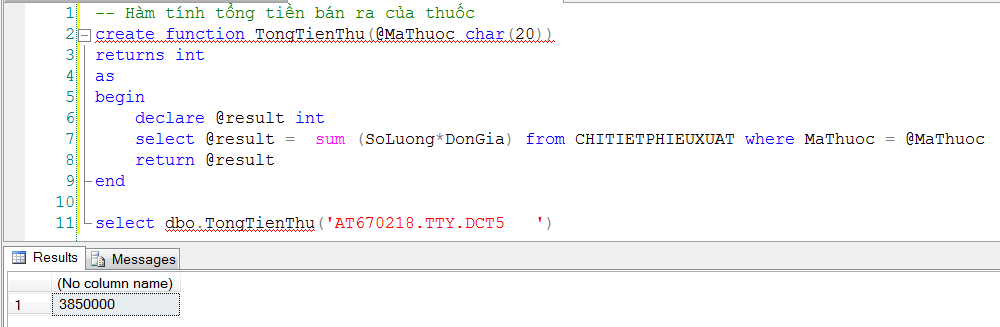
****

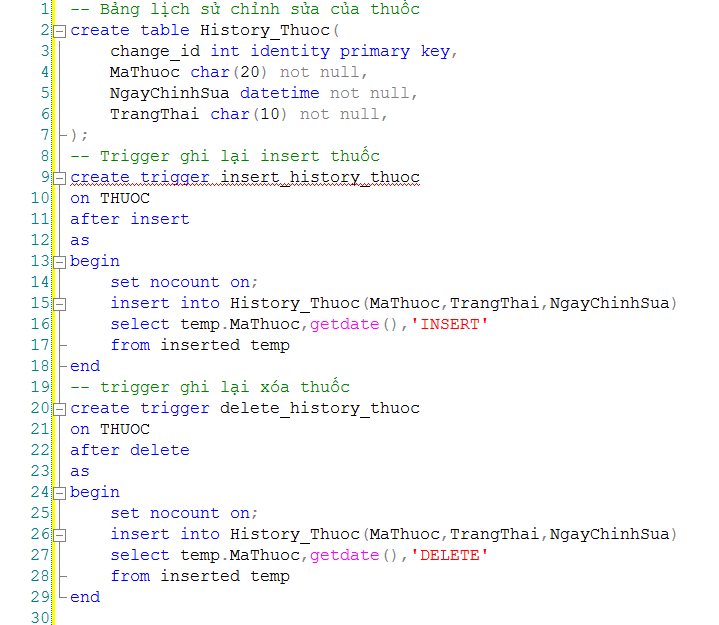
****

****



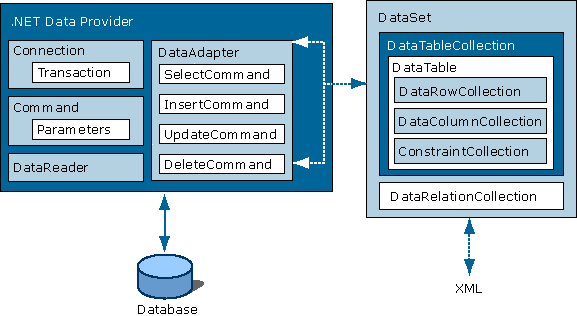
****





**Kiến trúc ADO.NET**

ADO.NET được chia ra làm hai phần chính rõ rệt, được thể hiện qua hình

Kiến trúc ADO.NET

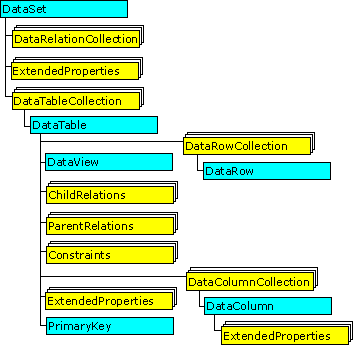
**DataSet** là thành phần chính cho đặc trưng kết nối không liên tục của kiến trúc ADO.NET. DataSet được thiết kế để có thể thích ứng với bất kỳ nguồn dữ liệu nào. DataSet chứa một hay nhiều đối tượng DataTable mà nó được tạo từ tập các dòng và cột dữ liệu, cùng với khoá chính, khóa ngoại, ràng buộc và các thông tin liên quan đến đối tượng **DataTable** này. Bản thân DataSet được dạng như một tập tin XML.

Thành phần chính thứ hai của ADO.NET chính là NET Provider Data, nó chứa các đối tượng phục vụ cho việc thao tác trên cơ sở dữ liệu được hiệu quả và nhanh chóng, nó bao gồm một tập các đối tượng **Connection**, **Command**, **DataReader**và **DataAdapter**. Đối tượng Connection cung cấp một kết nối đến cơ sở dữ liệu, Command cung cấp một thao tác đến cơ sở dữ liệu, DataReader cho phép chỉ đọc dữ liệu và **DataAdapter** là cấu nối trung gian giữa DataSet và nguồn dữ liệu.

**Mô hình đối tượng ADO.NET**

Có thể nói mô hình đối tượng của ADO.NET khá uyển chuyển, các đối tượng của nó được tạo ra dựa trên quan điểm đơn giản và dễ dùng. Đối tượng quan trọng nhất trong mô hình ADO.NET chính là Dataset. Dataset có thể được xem như là thể hiện của cả một cơ sở dữ liệu con, lưu trữ trên vùng nhớ cache của máy người dùng mà không kết nối đến cơ sở dữ liệu.

**Mô hình đối tượng của Dataset**

Mô hình đối tượng Dataset

**DataSet** bao gồm một tập các đối tượng **DataRelation** cũng như tập các đối tượng **DataTable**. Các đối tượng này đóng vai trò như các thuộc tính của **DataSet**.

**Đối tượng DataTable và DataColumn**

Ta có thể viết mã C# để tạo ra đối tượng DataTable hay nhận về từ kết quả của câu truy vấn đến cơ sở dữ liệu. DataTable có một số thuộc tính dùng chung ( public ) như thuộc tính Columns, từ thuộc tính này ta có thể truy cập đến đối tượng DataColumnsCollection thông qua chỉ mục hay tên của cột để nhận về các đối tượng DataColumn thích hợp, mỗi DataColumn tương ứng với một cột trong một bảng dữ liệu.

DataTable dt = new DataTable("tenBang"); DataColumn dc = dt.Columns["tenCot"];

**Đối tượng DataRelation**

Ngoài tập các đối tượng DataTable được truy cập thông qua thuộc tính Tables, DataSet còn có một thuộc tính Relations. Thuộc tính này dùng để truy cập đến đối tượng DataRelationCollection thông qua chỉ mục hay tên của quan hệ và sẽ trả về đối tượng DataRelation tương ứng.

DataSet ds = new DataSet("tenDataSet"); DataRelation dre = ds.Relations["tenQuanHe"];

**Các bản ghi ( Rows )**

Tương tự như thuộc tính **Columns** của đối tượng DataTable, để truy cập đến các dòng ta cũng có thuộc tính Rows. ADO. NET không đưa ra khái niệm RecordSet, thay vào đó để duyệt qua các dòng ( **Row** ), ta có thể truy cập các dòng thông qua thuộc tính Rows bằng vòng lặp foreach.

**Đối tượng SqlConnection và SqlCommand**

Đối tượng **SqlConnection** đại diện cho một kết nối đến cơ sở dữ liệu, đối tượng này có thể được dùng chung cho các đối tượng SqlCommand khác nhau. Đối tượng SqlCommand cho phép thực hiện một câu lệnh truy vấn trực tiếp : như **SELECT**, **UPDATE** hay **DELETE** hay gọi một thủ tục (Store Procedure) từ cơ sở dữ liệu.

**Đối tượng DataAdapter**

ADO.NET dùng DataAdapter như là chiếc cầu nối trung gian giữa DataSet và DataSource ( nguồn dữ liệu ), nó lấy dữ liệu từ cơ sở dữ liệu sau đó dùng phương **Fill()** để đẩy dữ liệu cho đối tượng DataSet. Nhờ đối tượng DataAdapter này mà DataSet tồn tại tách biệt, độc lập với cơ sở dữ liệu và một DataSet có thể là thể hiện của một hay nhiều cơ sở dữ liệu. Ví dụ :

//Tạo đối tượng SqlDataAdapter

SqlDataAdapter sda = new SqlDataAdapter();

// cung cấp cho sda một SqlCommand và SqlConnection ...

// lấy dữ liệu ...

//tạo đối tượng DataSet mới

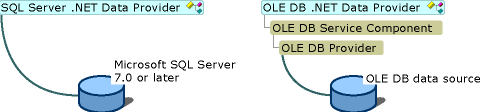
DataSet ds = new DataSet("tenDataSet");

//Đẩy dữ liệu trog sda vào ds bằng hàm Fill();

sda.Fill(ds);

**Trình cung cấp dữ liệu (.NET Data Providers)**

**.NET Framework** hỗ trợ hai trình cung cấp dữ liệu là **SQL Server .NET Data Provider** ( dành cho phiên bản SQL Server 7.0 của Microsoft trở lên ) và **OLE DB .NET Data Provider** ( dành cho các hệ quản trị cơ sở dữ liệu khác ) để truy cập vào cơ sở dữ liệu.

So sánh SQL Server .NET Data Provider và the OLE DB .NET Data Provider

**SQL Server .NET Data**Provider có các đặc điểm :

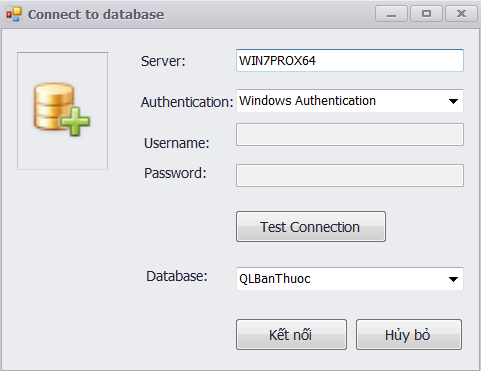
* Dùng nghi thức riêng để truy cập cơ sở dữ liệu
* Truy xuất dữ liệu sẽ nhanh hơn và hiệu quả hơn do không phải thông qua lớp OLE DB Provider hay ODBC
* Chỉ được dùng với hệ quản trị cơ sở dữ liệu SQL Server 7.0 trở lên.
* Được Mircrosoft hỗ trợ khá hoàn chỉnh.

**OLE DB .NET Data Provider** có các đặc điểm :

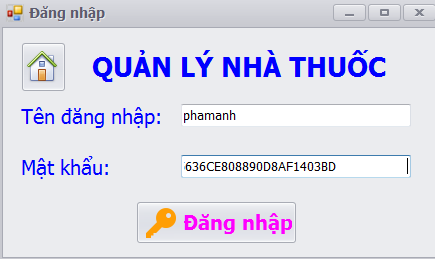
* Phải thông qua 2 lớp vì thế sẽ chậm hơn
* Thực hiện được các dịch vụ “Connection Pool”
* Có thể truy cập vào mọi Datasource có hỗ trợ OLE DB Provider thích hợp

**\* Giao diện phần mềm.**

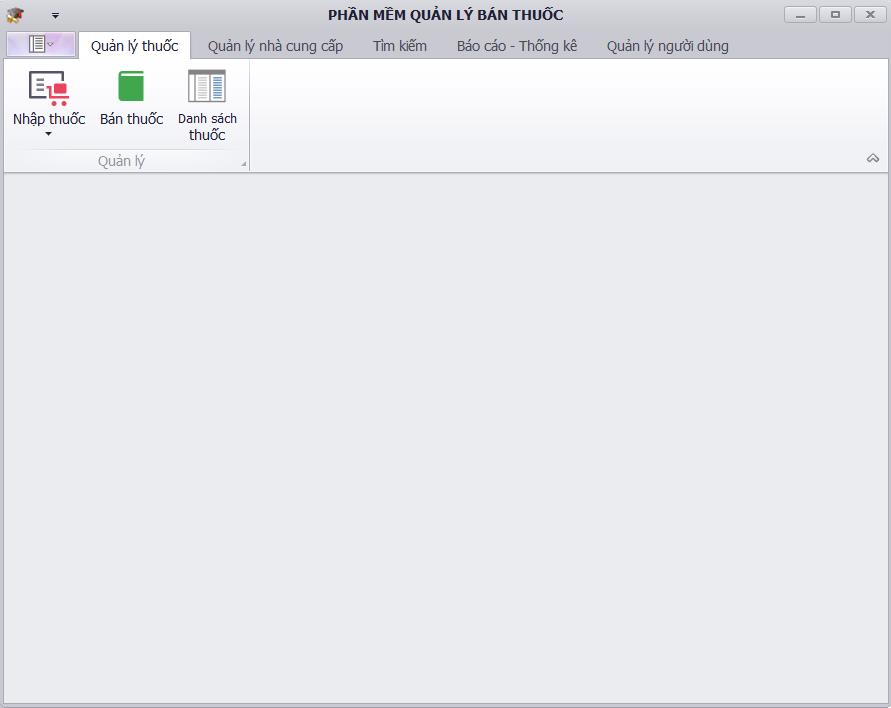
1. **Giao diện kết nối cơ sở dữ liệu:**

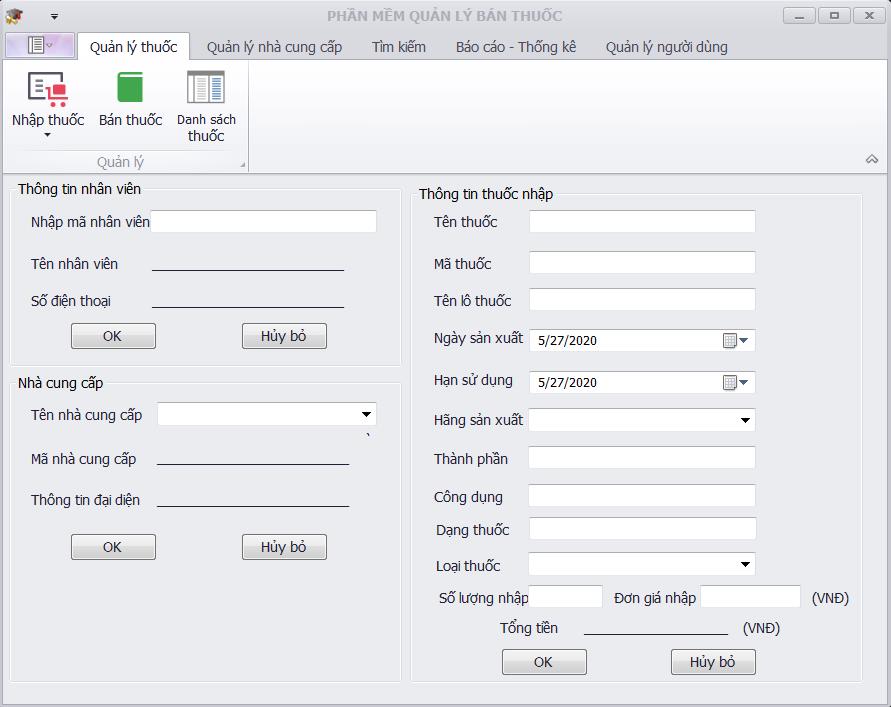
****

1. **Giao diện đăng nhập:**

****

**3. Giao diện trang chủ:**

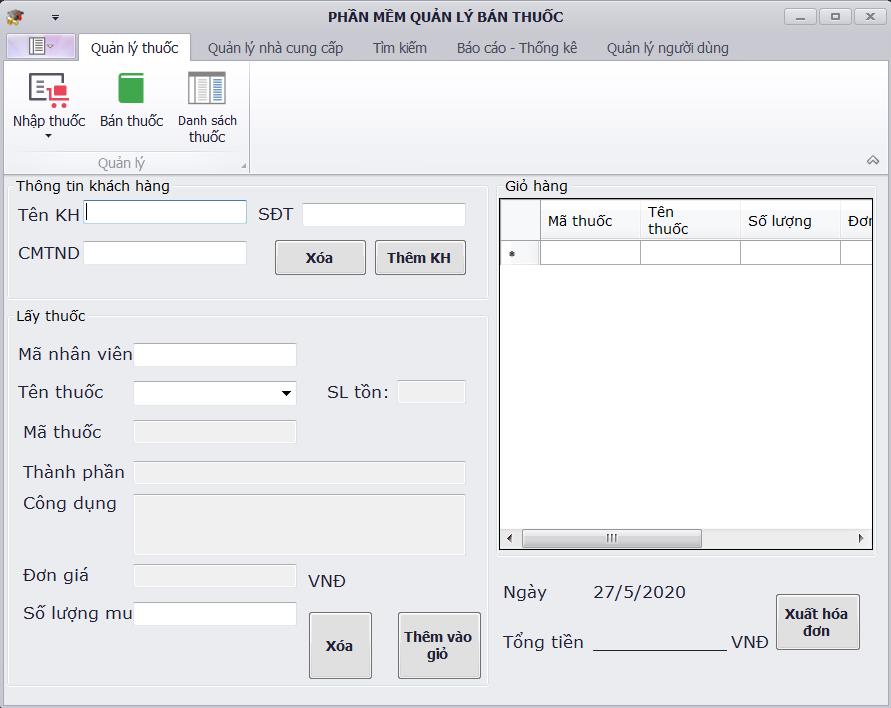
****

1. **Giao diện nhập thuốc:**

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmNhapThuoc** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbMaNV | textbox | Cho phép nhập mã nhân viên, rồi tự điền nốt tên và SĐT của nhân viên |
| 2 | cmbNCC | combobox | Hiển thị và cho phép lựa chọn tênnhà cung cấp. |
| 3 | txbTenThuoc | textbox | Cho phép nhập tên thuốc |
| 4 | txbMaThuoc | textbox | Cho phép nhập mã thuốc |
| 5 | txbLoThuoc | textbox | Cho phép nhập tên lô thuốc |
| 6 | dateNSX | datetimePicker | Cho phép lựa chọn ngày sản xuất của thuốc |
| 7 | dateHSD | datetimePicker | Cho phép lựa chọn hạn sử dụng của thuốc |
| 8 | cmbHSX | combobox | Hiển thị và cho phép lựa chọn hang sản xuất |
| 9 | txbThanhPhan | textbox | Cho phép nhập thông tin thành phần thuốc |
| 10 | txbCongDung | textbox | Cho phép nhập thông tin công dụng thuốc |
| 11 | txbDangThuoc | textbox | Cho phép nhập dạng thuốc |
| 12 | cmbLoaiThuoc | combobox | Hiển thị và cho phép lựa chọn loại thuốc |
| 13 | txbSoLuongNhap | textbox | Cho phép nhập số lượng thuốc |
| 14 | txbGiaNhap | textbox | Cho phép nhập đơn giá thuốc |
| 15 | btnXacNhanNV | button | Xác nhận nhân viên |
| 16 | btnHuyBoNV | button | Hủy bỏ rồi lựa chọn nhân viên khác |
| 17 | btnXacNhanNCC | button | Xác nhận nhà cung cấp |
| 18 | btnHuyBoNCC | button | Hủy bỏ rồi lựa chọn nhà cung cấp khác |
| 19 | btnXacNhanDon | button | Xuất đơn nhập |
| 20 | btnHuyBoDon | button | Hủy bỏ đơn vừa nhập |

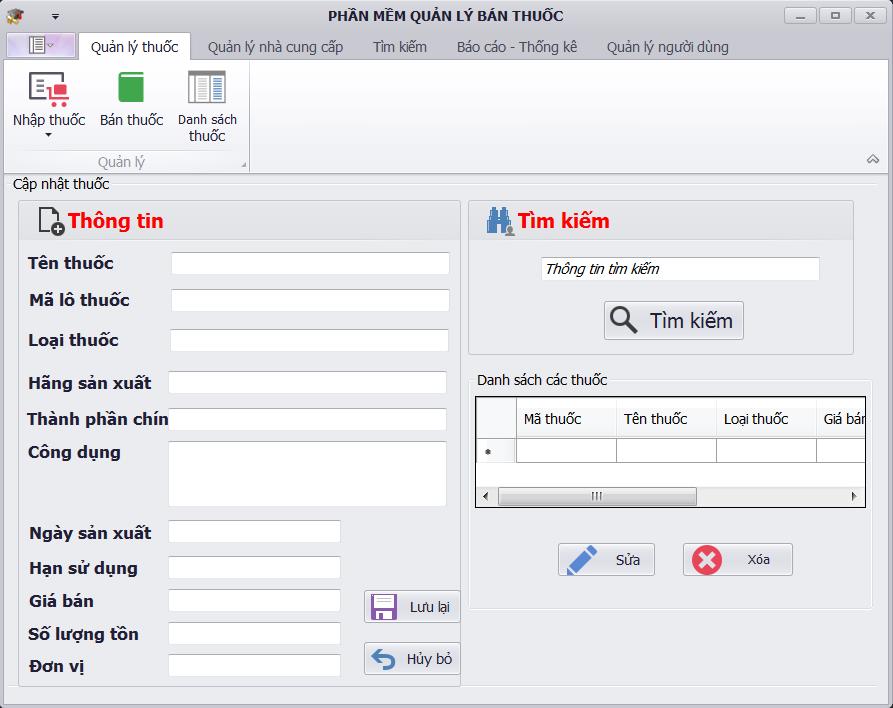
1. **Giao diện bán thuốc:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmBanThuoc** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbTenKH | textbox | Cho phép nhập tên khách hang |
| 2 | txbSDT | textbox | Cho phép nhập số điện thoại khách hang |
| 3 | txbCMTND | textbox | Cho phép nhập số chứng minh thư nhân dân/ mã thẻ căn cước của khách hàng |
| 4 | txbMaNV | textbox | Cho phép nhập mã nhân viên |
| 5 | cmbTenThuoc | combobox | Hiển thị và cho phép lựa chọn tên thuốc |
| 6 | txbSoLuong | textbox | Cho phép nhập số lượng thuốc |
| 7 | btnXoaKH | button | Xóa thông tin khách hang |
| 8 | btnThemKH | button | Thêm mới thông tin khách hang |
| 9 | btnXoaThuoc | butotn | Xóa thuốc vừa nhập khỏi giỏ hang |
| 10 | btnThemThuoc | button | Thêm thuốc vừa chọn vào giỏ hang |
| 11 | dgvGioHang | dataGridView | Hiện thị thông tin thuốc trong giỏ hàng |
| 12 | btnXuatDoan | button | Xuất hóa đơn bán thuốc |

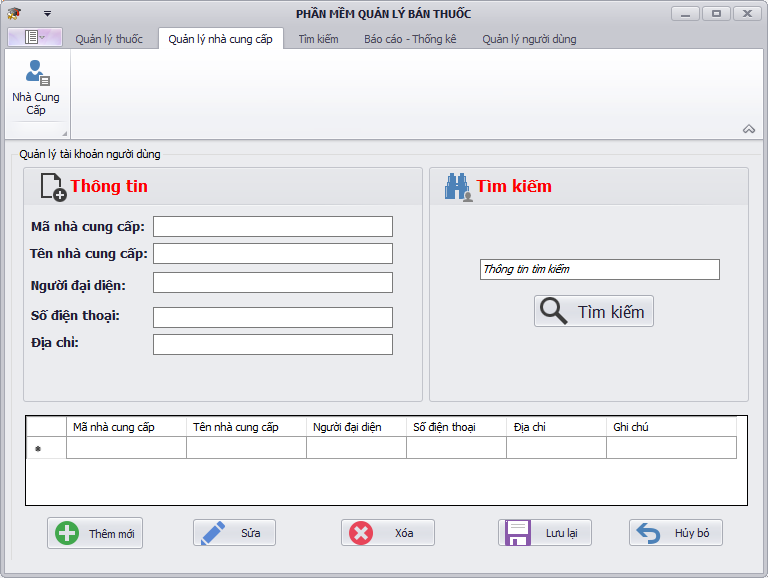
1. **Giao diện danh sách thuốc:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmDanhSachThuoc** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbTenThuoc | textbox | Cho phép nhập tên thuốc |
| 2 | txbLoThuoc | textbox | Cho phép nhập mã lô thuốc |
| 3 | cmbLoaiThuoc | combobox | Cho phép chọn loại thuốc |
| 4 | txbHangSanXuat | textbox | Cho phép nhập hãng sản xuất |
| 5 | txbThanhPhan | textbox | Cho phép nhập thông tin thành phần chính của thuốc |
| 6 | txbCongDung | textbox | Cho phép nhập thông tin công dụng thuốc |
| 7 | txbNSX | textbox | Cho phép nhập ngày sản xuất |
| 8 | txbHSD | textbox | Cho phép nhập hạn sử dụng |
| 9 | txbGiaBan | textbox | Cho phép nhập giá bán |
| 10 | txbSoLuongTon | textbox | Cho phép nhập số lượng tồn |
| 11 | txbDonVi | textbox | Cho phép nhập đơn vị thuốc (dạng thuốc) |
| 12 | txbTimKiem | textbox | Nhập thông tin tìm kiếm |
| 13 | dgvKetQua | dataGridView | Hiển thị dữ liệu tìm kiếm |
| 14 | btnTimKiem | button | Tìm kiếm thông tin thuốc |
| 15 | btnLuu | button | Lưu lại thông tin thuốc vừa nhập |
| 16 | btnHuy | button | Hủy bỏ thao tác nhập thuốc |
| 17 | btnSua | button | Chỉnh sửa các thông tin thuốc vừa nhập |
| 18 | btnXoa | button | Xóa thông tin thuốc trong cơ sở dữ liệu |

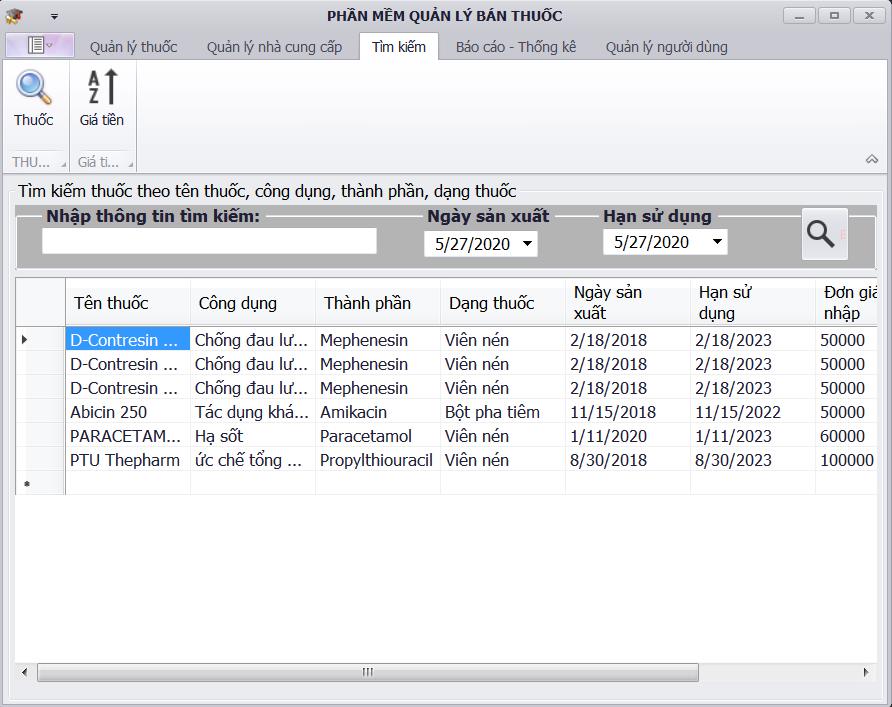
1. **Giao diện quản lý nhà cung cấp:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmNhaCungCap** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbMaNCC | textbox | Cho phép nhập mã nhà cung cấp |
| 2 | txbTenNCC | textbox | Cho phép nhập tên nhà cung cấp |
| 3 | txbNguoiDaiDien | textbox | Cho phép nhập tên người đại diện |
| 4 | txbSDT | textbox | Cho phép nhập số điện thoại liên lạc với nhà cung cấp |
| 5 | txbDiaChi | textbox | Cho phép nhập địa chỉ nhà cung cấp |
| 6 | txbTimKiem | textbox | Nhập thông tin tìm kiếm |
| 7 | btnThem | button | Thêm mới nhà cung cấp |
| 8 | btnSua | button | Chỉnh sửa thông tin nhà cung cấp |
| 9 | btnXoa | button | Xóa thông tin nhà cung cấp |
| 10 | btnLuu | button | Lưu dữ liệu vừa nhập |
| 11 | btnHuy | button | Hủy bỏ thao tác vừa nhập |
| 12 | btnTimKiem | button | Tìm kiếm thông tin nhà cung cấp |
| 13 | dgvNCC | dataGridView | Hiển thị danh sách nhà cung cấp |

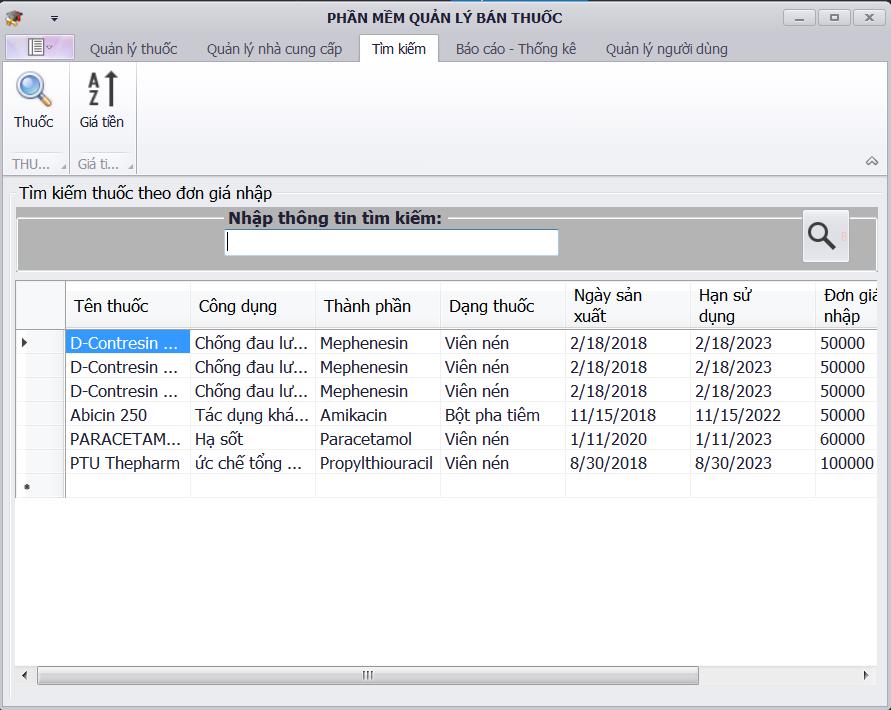
1. **Giao diện tìm kiếm theo tên thuốc:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmTenThuoc** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbTimKiem | textbox | Nhập tên thuốc cần tìm kiếm |
| 2 | dateNSX | datetimePicker | Cho phép lựa chọn ngày sản xuất |
| 3 | dateHSD | datetimePicker | Cho phép lựa chọn hạn sử dụng |
| 4 | dgvTimKiem | dateGridView | Đưa ra thông tin thuốc cần tìm kiếm |
| 5 | btnTimKiem | button | Tìm kiếm theo tên thuốc |

1. **Giao diện tìm kiếm theo đơn giá nhập thuốc:**

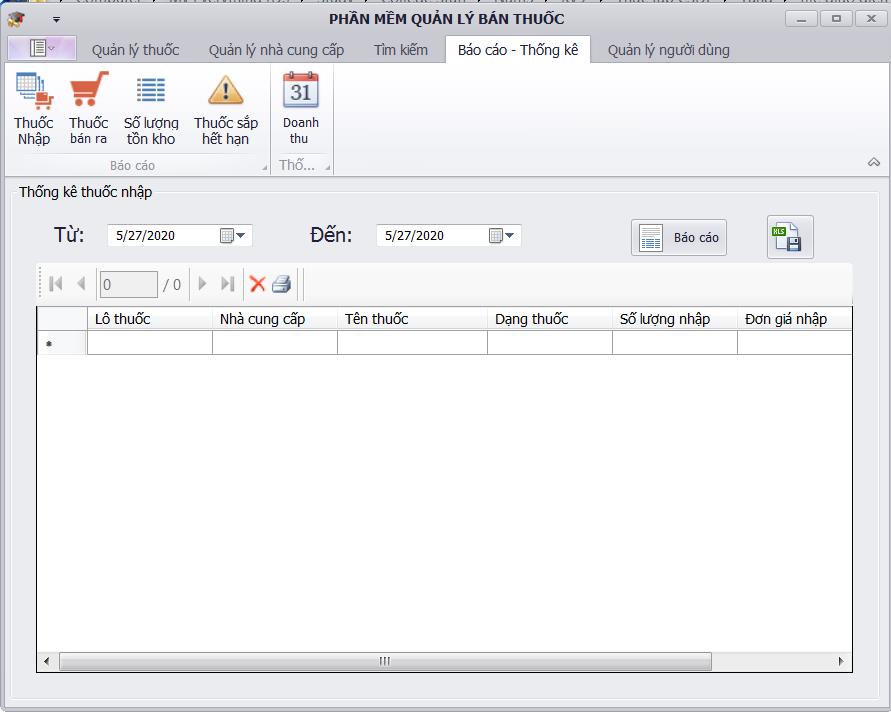
****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmThapCao** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbTimKiem | textbox | Nhập tên thuốc cần tìm kiếm |
| 2 | dgvTimKiem | dateGridView | Đưa ra thông tin thuốc cần tìm kiếm |
| 3 | btnTimKiem | button | Tìm kiếm theo giá thuốc |

1. **Giao diện thống kê - báo cáo:**

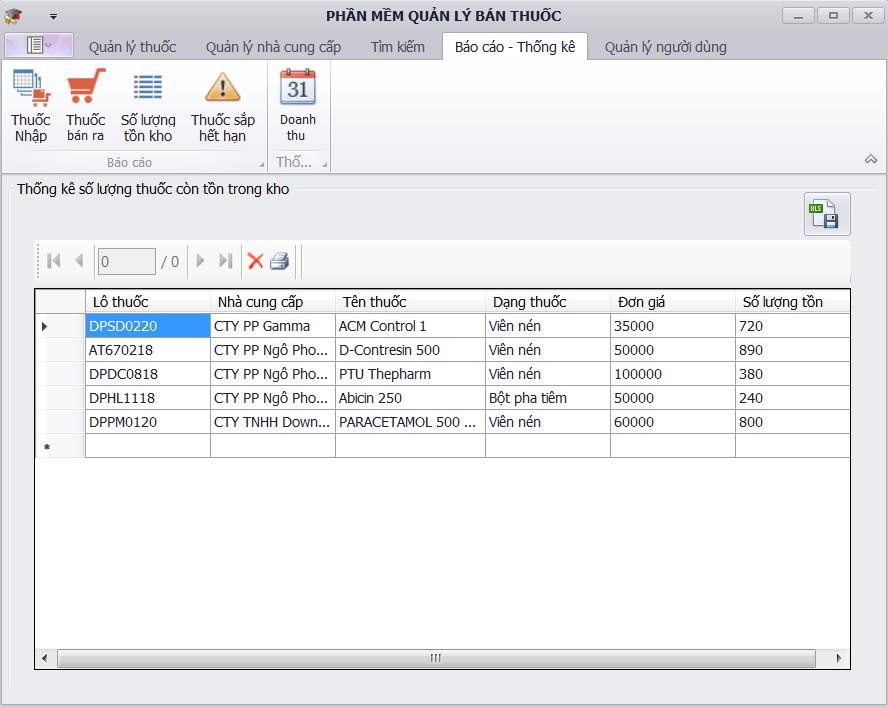
**9.1.Giao diện thống kê thuốc đã nhập:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmThuocNhap** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | dateBatDau | datetimePicker | Chọn ngày bắt đầu |
| 2 | dateKetThuc | datetimePicker | Chọn ngày kết thúc |
| 3 | btnThongKe | button | Đưa ra kết quả thống kê |
| 4 | btnExcel | button | Xuất file excel |
| 5 | dgvKetQua | dataGridView | Hiển thị kết quả thống kê |

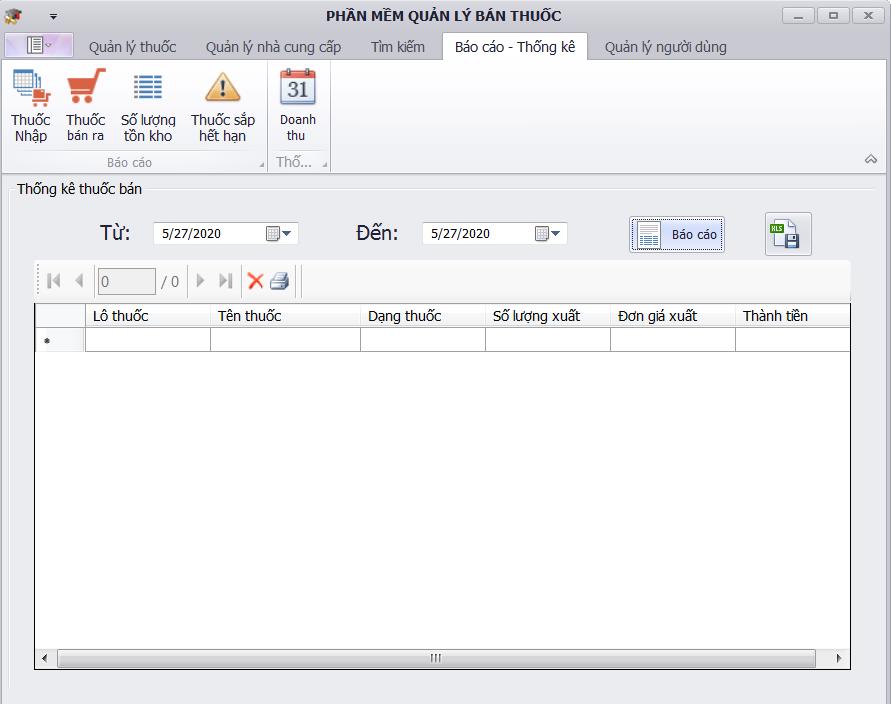
**9.2. Giao diện thống kê số lượng thuốc tồn:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmSoLuongTon** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | btnExcel | button | Xuất file excel |
| 2 | dgvKetQua | dataGridView | Hiển thị danh sách thuốc |

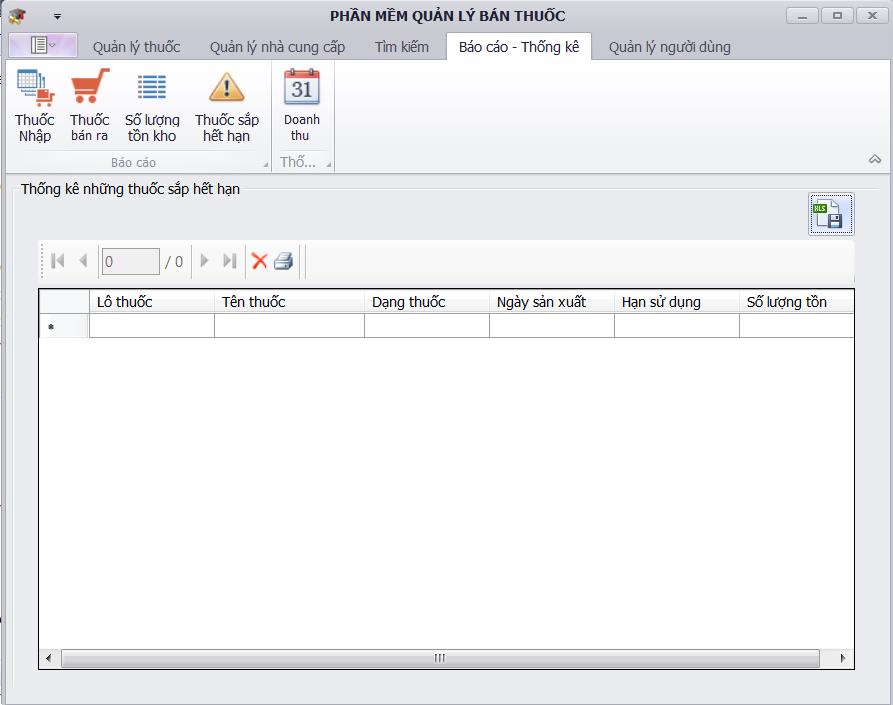
**9.3. Giao diện thống kê thuốc đã bán ra:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmThuocBanRa** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | dateBatDau | datetimePicker | Chọn ngày bắt đầu |
| 2 | dateKetThuc | datetimePicker | Chọn ngày kết thúc |
| 3 | btnThongKe | button | Đưa ra kết quả thống kê |
| 4 | btnExcel | button | Xuất file excel |
| 5 | dgvKetQua | dataGridView | Hiển thị kết quả thống kê |

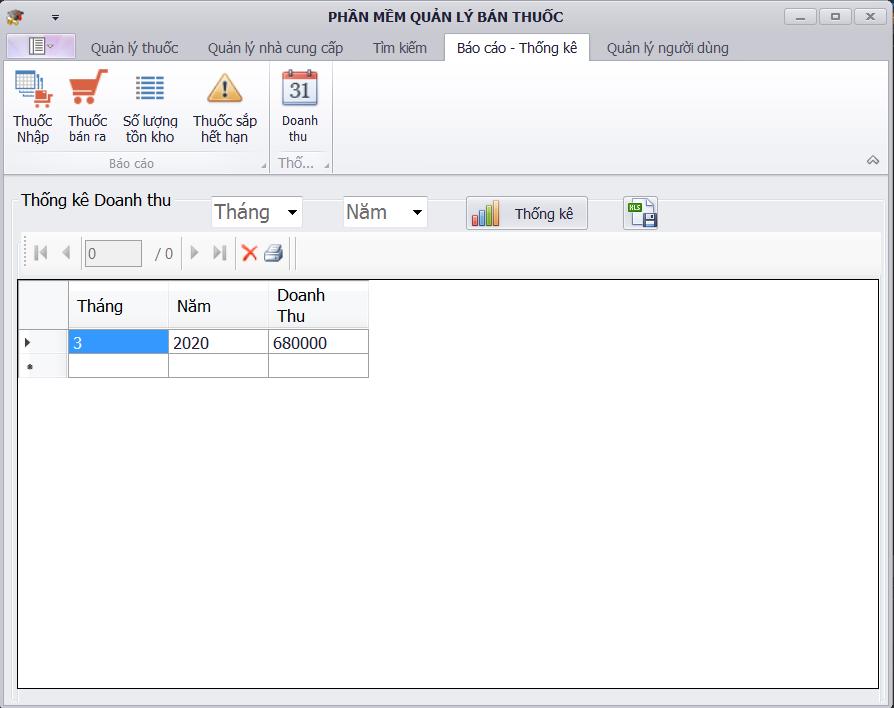
**9.4. Giao diện thống kê thuốc hết hạn:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmThuocHetHan** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | btnExcel | button | Xuất file excel |
| 2 | dGVKetQua | dataGridView | Hiển thị danh sách thuốc |

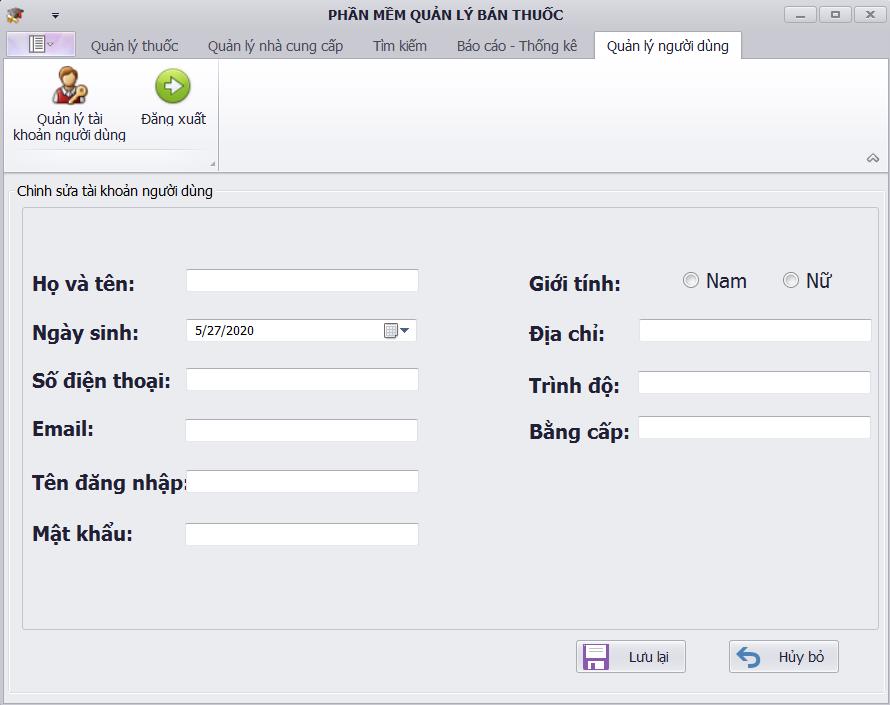
**9.5. Giao diện thống kê doanh thu:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmDoanhThu** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | cmbThang | combobox | Cho phép lựa chọn tháng |
| 2 | cmbNam | combobox | Cho phép lựa chọn năm |
| 3 | btnThongKe | button | Đưa ra kết quả thống kê |
| 4 | btnExcel | button | Xuất file excel |
| 5 | dgvKetQua | dataGridView | Hiển thị kết quả thống kê doanh thu |

1. **Giao diện quản lý tài khoản người dùng:**

****

**- Đặc tả giao diện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **frmQuanLyNguoiDung** | | | |
| **STT** | **Tên thành phần** | **Định dạng** | **Mô tả** |
| 1 | txbHoTen | textbox | Cho phép nhập họ tên người dùng |
| 2 | checkGioiTinh | radiobutton | Cho phép lựa chọn giới tính (Nam/Nữ) |
| 3 | dateNS | datetimePicker | Cho phép lựa chọn ngày sinh người dùng |
| 4 | txbDiaChi | textbox | Cho phép nhập địa chỉ người dùng |
| 5 | txbSDT | textbox | Cho phép nhập số điện thoại người dùng |
| 6 | txbTrinhDo | textbox | Cho phép nhập trình độ của người dùng |
| 7 | txbEmail | textbox | Cho phép nhập email người dùng |
| 8 | txbBangCap | texbox | Cho phép nhập thông tin bằng cấp |
| 9 | txbTenDangNhap | texbox | Cho phép nhập tên đăng nhập |
| 10 | txbMatKhau | texbox | Cho phép nhập mật khẩu đăng nhập |
| 11 | btnLuu | button | Lưu thông tin người dùng vừa nhập |
| 12 | btnHuy | button | Hủy bỏ thao tác nhập thông tin người dùng |