**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU PHÂN TÁN**

**ĐỀ TÀI: THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÍ KHO THUỐC CỦA BỆNH VIỆN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1 | 1771020644 | CAO VĂN THIỆN | 05/09/2005 | CNTT 17-04 |
| 2 | 1771020175 | LÊ THÙY DUNG | 17/08/2005 | CNTT 17-04 |

### 

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU PHÂN TÁN**

**ĐỀ TÀI: THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÍ KHO THUỐC CỦA BỆNH VIỆN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020644 | CAO VĂN THIỆN | 05/09/2005 |  |  |
| 2 | 1771020175 | LÊ THÙY DUNG | 17/08/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Việc ứng dụng các hệ quản trị cơ sở dữ liệu vào quản lý thông tin đã trở thành một yêu cầu thiết yếu trong mọi lĩnh vực. Đặc biệt, trong ngành y tế, nơi việc quản lý dữ liệu chính xác và kịp thời đóng vai trò quan trọng trong công tác chăm sóc sức khỏe, một hệ thống quản lý kho thuốc hiệu quả không chỉ giúp tối ưu hóa quy trình làm việc mà còn góp phần đảm bảo an toàn cho người bệnh.

Quản lý kho thuốc tại bệnh viện là một bài toán quan trọng, đòi hỏi sự chính xác cao trong quá trình lưu trữ, kiểm soát và phân phối thuốc. Một hệ thống quản lý kho thuốc khoa học sẽ giúp bệnh viện giảm thiểu các rủi ro như thất thoát thuốc, nhầm lẫn trong cấp phát, hoặc sử dụng thuốc đã hết hạn. Đồng thời, hệ thống này còn hỗ trợ công tác kiểm kê, theo dõi tình trạng tồn kho, cũng như quản lý thông tin về nguồn gốc, lô sản xuất và hạn sử dụng của từng loại thuốc.

Việc xây dựng cơ sở dữ liệu quản lý kho thuốc tại bệnh viện được thực hiện với mục tiêu thiết kế một hệ thống dữ liệu khoa học, đáp ứng đầy đủ các yêu cầu nghiệp vụ như: theo dõi nhập - xuất thuốc, kiểm soát chất lượng và hạn sử dụng, truy xuất thông tin nhanh chóng, cũng như phân quyền sử dụng cho từng đối tượng phù hợp. Để đảm bảo hiệu quả quản lý, hệ thống được thiết kế dựa trên các kỹ thuật tiên tiến như view, procedure, trigger và cơ chế phân quyền người dùng, giúp nâng cao tính bảo mật, đảm bảo toàn vẹn dữ liệu và tạo điều kiện thuận lợi cho việc mở rộng khi cần thiết.

Bên cạnh đó, hệ thống không chỉ giúp tối ưu hóa quy trình hoạt động nội bộ mà còn góp phần nâng cao chất lượng dịch vụ y tế, đảm bảo việc cung ứng thuốc kịp thời và đúng yêu cầu. Điều này có ý nghĩa quan trọng trong việc nâng cao hiệu quả điều trị, giảm thiểu sai sót y khoa và đảm bảo sự an toàn cho bệnh nhân.

Xuất phát từ những nhu cầu thực tiễn trên, nghiên cứu và phát triển một hệ thống quản lý kho thuốc tại bệnh viện là một nhiệm vụ cần thiết và có ý nghĩa quan trọng. Với tinh thần đó, đề tài này sẽ tập trung vào việc phân tích, thiết kế và triển khai một hệ thống cơ sở dữ liệu phù hợp, góp phần nâng cao hiệu quả quản lý thuốc trong môi trường bệnh viện.

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc193225241)

[**CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI** 5](#_Toc193225243)

[**1.1.Bối cảnh thực tế** 5](#_Toc193225244)

[**1.2. Mục tiêu** 6](#_Toc193225245)

[*1.2.1. Mục tiêu tổng quát* 6](#_Toc193225246)

[1.2.2. *Mục tiêu cụ thể* 6](#_Toc193225247)

[**1.3. Lợi ích của hệ thống** 7](#_Toc193225248)

[**CHƯƠNG 2. MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ** 8](#_Toc193225249)

[**2.1. Xác định các thực thể, thuộc tính và ràng buộc** 8](#_Toc193225250)

[**2.2. Xây dựng các bảng** 10](#_Toc193225251)

[**CHƯƠNG 3. TẠO CƠ SỞ DỮ LIỆU** 13](#_Toc193225252)

[**3.1. Tạo Database** 13](#_Toc193225253)

[**3.2. Chèn dữ liệu** 13](#_Toc193225254)

[**3.3. In bảng dữ liệu** 21](#_Toc193225255)

[**CHƯƠNG 4 . XÂY DỰNG CÁC VIEW** 24](#_Toc193225256)

[**4.1. Danh sách thuốc và tồn kho** 24](#_Toc193225257)

[**4.2. Danh sách nhà cung cấp** 25](#_Toc193225258)

[**4.3. Tổng số lượng nhập kho theo thuốc** 26](#_Toc193225259)

[**4.4. Tổng số lượng xuất kho theo thuốc** 27](#_Toc193225260)

[**4.5. Lịch sử nhập kho** 28](#_Toc193225261)

[**4.6. Lịch sử nhập kho** 30](#_Toc193225262)

[**4.7. Tình trạng thuốc sắp hết hạn** 31](#_Toc193225263)

[**4.8. Tình trạng tồn kho dưới mức tối thiểu (10 đơn vị)** 32](#_Toc193225264)

[**4.9. Chi tiết tồn kho kèm thông tin thuốc** 33](#_Toc193225265)

[**4.10. Báo cáo nhập- xuất- tồn kho** 34](#_Toc193225266)

[**CHƯƠNG 5. XÂY DỰNG CÁC PROCEDURE** 36](#_Toc193225267)

[**5.1. Procedure thêm thuốc mới** 36](#_Toc193225268)

[**5.2. Procedure thêm nhà cung cấp mới** 36](#_Toc193225269)

[**5.3. Procedure nhập kho thuốc** 37](#_Toc193225270)

[**5.4. Procedure xuất kho thuốc** 38](#_Toc193225271)

[**5.5. Procedure kiểm tra tồn kho thuốc** 38](#_Toc193225272)

[**5.6. Procedure kiểm tra danh sách thuốc sắp hết hạn** 39](#_Toc193225273)

[**5.7. Procedure cập nhật giá thuốc** 40](#_Toc193225274)

[**5.8. Procedure xóa thuốc khỏi hệ thống** 41](#_Toc193225275)

[**5.9. Procedure lấy thông tin nhà cung cấp** 41](#_Toc193225276)

[**5.10. Proceudre lấy báo cáo tổng hợp nhập-xuất-tồn kho** 41](#_Toc193225277)

[**CHƯƠNG 6. XÂY DỰNG CÁC TRIGGER** 42](#_Toc193225278)

[**6.1. Trigger tự động cập nhật tồn kho khi nhập kho** 42](#_Toc193225279)

[**6.2. Trigger tự động cập nhật tồn kho khi xuất kho** 43](#_Toc193225280)

[**6.3. Trigger ngăn chặn xuất kho nếu số lượng không đủ** 44](#_Toc193225281)

[**6.4. Trigger xóa dữ liệu tồn kho khi xóa thuốc** 45](#_Toc193225282)

[**6.5. Trigger cấm nhập giá trị âm vào bảng nhập kho** 46](#_Toc193225283)

[**6.6. Trigger cấm nhập giá trị âm vào bảng xuất kho** 47](#_Toc193225284)

[**6.7. Trigger kiểm tra hạn sử dụng khi nhập kho** 48](#_Toc193225285)

[**6.8. Trigger kiểm tra cập nhật giá thuốc** 49](#_Toc193225286)

[**6.9. Trigger tạo nhật ký khi có nhập kho** 50](#_Toc193225287)

[**6.10. Trigger tạo nhật ký khi có xuất kho** 51](#_Toc193225288)

[**CHƯƠNG 7. PHÂN QUYỀN VÀ BẢO VỆ CÁC CƠ SỞ DỮ LIỆU** 52](#_Toc193225289)

[**7.1. Phân quyền** 52](#_Toc193225290)

[**7.2. Quản lý kho** 53](#_Toc193225291)

[**7.3. Nhân viên kho** 53](#_Toc193225292)

[**7.4. Khách hàng** 54](#_Toc193225293)

[**7.5. Tạo tài khoản đăng nhập** 54](#_Toc193225294)

[**7.6. Tạo người dùng** 54](#_Toc193225295)

[**7.7. Gán vai trò người dùng** 54](#_Toc193225296)

[**7.8. Kiểm tra danh sách phân quyền** 55](#_Toc193225297)

[**KẾT LUẬN** 56](#_Toc193225298)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 57](#_Toc193225299)

**CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI**

## **1.1.Bối cảnh thực tế**

Hiện nay, tại các bệnh viện, nhà thuốc hoặc kho dược phẩm, việc quản lý thuốc thường được thực hiện thủ công. Khi một đơn thuốc được kê hoặc cần kiểm kê hàng tồn kho, nhân viên thường phải ghi chép bằng tay hoặc sử dụng bảng tính để theo dõi số lượng thuốc nhập - xuất. Sau đó, thông tin này sẽ được chuyển đến các bộ phận liên quan như kho dược, kế toán hoặc bác sĩ để xác nhận và thực hiện các bước tiếp theo. Quy trình này không chỉ mất nhiều thời gian mà còn dễ dẫn đến sai sót, thất thoát thuốc hoặc nhập sai số liệu.

Một số vấn đề phổ biến trong việc quản lý kho thuốc hiện nay bao gồm:

* Sai sót trong nhập/xuất kho: Do nhập liệu thủ công nên dễ xảy ra lỗi về số lượng, hạn sử dụng hoặc tên thuốc.
* Quản lý hàng tồn kho chưa tối ưu: Không có hệ thống theo dõi chính xác dẫn đến tình trạng thừa hoặc thiếu thuốc cần thiết.
* Hạn chế trong theo dõi hạn sử dụng: Dược phẩm có hạn sử dụng ngắn, nếu không kiểm soát tốt có thể dẫn đến việc sử dụng thuốc hết hạn.
* Chậm trễ trong quy trình cấp phát: Việc tìm kiếm thông tin thuốc và xử lý đơn hàng mất nhiều thời gian, ảnh hưởng đến công tác điều trị.
* Khó khăn trong việc truy xuất nguồn gốc thuốc: Khi có sự cố về thuốc, việc truy xuất nguồn gốc để kiểm tra chất lượng và thu hồi gặp nhiều khó khăn.

Để khắc phục vấn đề trên, chúng ta sẽ xây dựng một hệ thống quản lý kho thuốc với ý tưởng như sau: khi có nhu cầu nhập/xuất thuốc, nhân viên kho sẽ sử dụng một thiết bị di động hoặc máy tính bảng chạy chương trình quản lý để ghi lại thông tin về thuốc (tên thuốc, số lượng, đơn vị, ngày nhập/xuất, hạn sử dụng, v.v.). Sau đó, dữ liệu này sẽ được gửi về database server thông qua kết nối không dây và được tự động cập nhật vào hệ thống

Với hệ thống này, quá trình quản lý kho thuốc sẽ trở nên nhanh chóng, chính xác hơn, giúp giảm thiểu sai sót trong nhập xuất, hạn chế tình trạng thiếu hụt hoặc dư thừa thuốc, đồng thời hỗ trợ việc kiểm kê hiệu quả. Một hệ thống cơ sở dữ liệu quản lý kho thuốc sẽ giúp tối ưu hóa việc vận hành, góp phần nâng cao chất lượng dịch vụ y tế.

## **1.2. Mục tiêu**

### *1.2.1. Mục tiêu tổng quát*

Mục tiêu chính của hệ thống quản lý kho thuốc là nâng cao hiệu quả quản lý dược phẩm, đảm bảo thuốc được bảo quản, phân phối chính xác, giảm sai sót và tối ưu hóa quy trình vận hành trong kho. Hệ thống giúp quản lý kho thuốc một cách nhanh chóng, chính xác và an toàn, đồng thời cải thiện chất lượng cung ứng thuốc trong bệnh viện, nhà thuốc và trung tâm y tế.

### 1.2.2. *Mục tiêu cụ thể*

**Quản lý thông tin thuốc hiệu quả**

* Lưu trữ thông tin chi tiết về từng loại thuốc: tên thuốc, số lô, hạn sử dụng, đơn vị tính, nhà cung cấp.
* Theo dõi lịch sử nhập - xuất thuốc, giúp kiểm soát hàng tồn kho và tránh tình trạng thiếu hụt hoặc dư thừa.
* Quản lý thuốc cận hạn sử dụng để giảm lãng phí và đảm bảo thuốc được phân phối hợp lý.
* Xây dựng chương trình kiểm soát thuốc theo dõi đặc biệt, ví dụ như thuốc kê đơn, thuốc kiểm soát đặc biệt hoặc vaccine.

**Tối ưu hóa quy trình nhập - xuất thuốc**

* Giảm thiểu nhầm lẫn trong việc nhập và xuất thuốc bằng cách số hóa quy trình nhập kho (quét mã vạch, tích hợp hệ thống điện tử).
* Đảm bảo quá trình phân phối thuốc đến bệnh nhân hoặc các khoa phòng diễn ra chính xác, đúng loại, đúng số lượng.
* Theo dõi và cảnh báo tự động khi lượng thuốc trong kho xuống dưới mức tối thiểu hoặc khi thuốc gần hết hạn.

**Tối ưu hóa sử dụng tài nguyên trong kho**

* Quản lý không gian lưu trữ thuốc hợp lý, phân bổ khu vực bảo quản theo nhiệt độ, độ ẩm phù hợp.
* Theo dõi số lượng thuốc trong từng khu vực kho, hạn chế tình trạng mất mát hoặc thuốc bị hư hỏng.
* Tích hợp hệ thống dự báo nhu cầu thuốc, giúp nhà thuốc hoặc bệnh viện nhập hàng một cách tối ưu, tránh tồn kho quá nhiều hoặc thiếu hụt thuốc quan trọng.

**Nâng cao hiệu quả vận hành và tốc độ xử lý**

* Giảm thời gian chờ đợi khi cấp phát thuốc, đặc biệt trong trường hợp cấp cứu.
* Hỗ trợ bác sĩ, dược sĩ truy xuất nhanh chóng thông tin về thuốc, đơn hàng và số lượng còn trong kho.
* Tích hợp hệ thống thanh toán và bảo hiểm y tế để giảm thiểu sai sót trong kê đơn và xuất thuốc.

**1.3. Lợi ích của hệ thống**

* Tăng cường kiểm soát chất lượng thuốc: Giúp theo dõi chất lượng, hạn sử dụng, xuất xứ của từng lô thuốc, hạn chế rủi ro do thuốc kém chất lượng.
* Nâng cao hiệu suất làm việc: Hệ thống giúp giảm thiểu các thao tác thủ công, giúp nhân viên dược phẩm tập trung vào các nhiệm vụ quan trọng hơn.
* Cải thiện khả năng dự báo và lập kế hoạch: Hỗ trợ quản lý kho trong việc dự đoán nhu cầu thuốc, giúp nhà thuốc và bệnh viện nhập hàng hợp lý hơn.
* Tối ưu hóa tài nguyên: Giúp tiết kiệm thời gian, chi phí và nhân lực trong quá trình quản lý và vận hành kho thuốc.

**CHƯƠNG 2. MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ**

## **2.1. Xác định các thực thể, thuộc tính và ràng buộc**

**Bảng Medicine (Thuốc)**

Chức năng: Lưu thông tin về các loại thuốc trong kho.

Thực thể: **tblMedicine** (**id**, name, price, expiry\_date)

Thuộc tính:

* id: Khóa chính (PRIMARY KEY), tự động tăng (IDENTITY(1,1))
* name: Kiểu NVARCHAR(255), không được để trống (NOT NULL)
* price: Kiểu DECIMAL(10,2), không được để trống (NOT NULL)
* expiry\_date: Kiểu DATE, không được để trống (NOT NULL)

Ràng buộc:

* PRIMARY KEY (id): Đảm bảo id là duy nhất cho mỗi bản ghi.
* IDENTITY(1,1): id sẽ tự động tăng từ 1, mỗi lần tăng 1 đơn vị.
* NOT NULL: Các cột name, price, expiry\_date không được chứa giá trị NULL.

**Bảng Supplier (Nhà cung cấp)**

Thực thể: **tblSupplier**(**id,**name,phone)

Thuộc tính:

* id: Khóa chính (PRIMARY KEY), tự động tăng (IDENTITY(1,1))
* name: Kiểu NVARCHAR(255), không được để trống (NOT NULL)
* phone: Kiểu NVARCHAR(20), có thể để trống (NULL mặc định)

Ràng buộc:

* PRIMARY KEY (id): Đảm bảo id là duy nhất cho mỗi bản ghi.
* IDENTITY(1,1): id sẽ tự động tăng từ 1, mỗi lần tăng 1 đơn vị.
* NOT NULL cho cột name: Bắt buộc phải nhập tên nhà cung cấp.

**Bảng StockIn (Nhập kho)**

Thực thể: **tblStockIn** (**id**, medicine\_id, supplier\_id, quantity, date\_in)

Thuộc tính:

* id: Khóa chính (PRIMARY KEY), tự động tăng (IDENTITY(1,1))
* medicine\_id: Kiểu INT, không được để trống (NOT NULL), liên kết với bảng Medicine
* supplier\_id: Kiểu INT, không được để trống (NOT NULL), liên kết với bảng Supplier
* quantity: Kiểu INT, không được để trống (NOT NULL), thể hiện số lượng thuốc nhập
* date\_in: Kiểu DATE, không được để trống (NOT NULL), ngày nhập kho

Ràng buộc:

* PRIMARY KEY (id): Đảm bảo id là duy nhất cho mỗi bản ghi.
* IDENTITY(1,1): id sẽ tự động tăng từ 1, mỗi lần tăng 1 đơn vị.
* FOREIGN KEY (medicine\_id) REFERENCES Medicine(id): Đảm bảo medicine\_id phải tồn tại trong bảng Medicine.
* FOREIGN KEY (supplier\_id) REFERENCES Supplier(id): Đảm bảo supplier\_id phải tồn tại trong bảng Supplier.
* NOT NULL: Các cột medicine\_id, supplier\_id, quantity, date\_in bắt buộc nhập dữ liệu.

**Bảng StockOut (Xuất kho)**

Thực thể: **tblStockOut** (**id**, medicine\_id, quantity, date\_out)

Thuộc tính:

* id: Khóa chính (PRIMARY KEY), tự động tăng (IDENTITY(1,1))
* medicine\_id: Kiểu INT, không được để trống (NOT NULL), liên kết với bảng Medicine
* quantity: Kiểu INT, không được để trống (NOT NULL), thể hiện số lượng thuốc xuất
* date\_out: Kiểu DATE, không được để trống (NOT NULL), ngày xuất kho

Ràng buộc:

* PRIMARY KEY (id): Đảm bảo id là duy nhất cho mỗi bản ghi.
* IDENTITY(1,1): id sẽ tự động tăng từ 1, mỗi lần tăng 1 đơn vị.
* FOREIGN KEY (medicine\_id) REFERENCES Medicine(id): Đảm bảo medicine\_id phải tồn tại trong bảng Medicine.
* NOT NULL: Các cột medicine\_id, quantity, date\_out bắt buộc nhập dữ liệu.

**Bảng Inventory (Tồn kho)**

Thực thể: **tblInventory** (**medicine\_id**, quantity)

Thuộc tính:

* medicine\_id: Kiểu INT, khóa chính, không được để trống (NOT NULL), liên kết với bảng Medicine.
* quantity: Kiểu INT, không được để trống (NOT NULL), thể hiện số lượng thuốc còn trong kho.

Ràng buộc:

* PRIMARY KEY (medicine\_id): Đảm bảo mỗi loại thuốc chỉ có một bản ghi tồn kho duy nhất.
* FOREIGN KEY (medicine\_id) REFERENCES Medicine(id): Đảm bảo medicine\_id phải tồn tại trong bảng Medicine.
* NOT NULL: Các cột medicine\_id, quantity bắt buộc nhập dữ liệu.
* CHECK (quantity >= 0): Đảm bảo số lượng tồn kho không âm.

## 

## **2.2. Xây dựng các bảng**

**tblMedicine: Lưu trữ thông tin về thuốc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | iId | INT | PK, IDENTITY(1,1) | Mã thuốc(tự động tăng) |
| 2 | sName | NVARCHAR(255) | NOT NULL | Tên thuốc |
| 3 | fPrice | DEMICAL  (10,2) | NOT NULL | Giá thuốc |
| 4 | dExpiryDate | DATE | NOT NULL | Hạn sử dụng |

**tblSupplier: Lưu trữ thông tin về nhà cung cấp**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | iId | INT | PK, IDENTITY(1,1) | Mã nhà cung cấp (tự động tăng) |
| 2 | sName | NVARCHAR(255) | NOT NULL | Tên nhà cung cấp |
| 3 | SPhone | NVARCHAR(20) | NULLABLE | Số điện thoại |

**tblStockIn: Lưu trữ thông tin nhập kho**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | iId | INT | PK, IDENTITY(1,1) | Mã nhập kho (tự động tăng) |
| 2 | iMedicineID | INT | NOT NULL, FK | Mã thuốc (Liên kết với Medicine) |
| 3 | iSupplierID | INT | NOT NULL, FK | Mã nhà cung cấp (Liên kết với Supplier) |
| 4 | iQuantity | INT | NOT NULL, CHECK (iQuantity>0) | Số lượng nhập kho |
| 5 | dDateIn | DATE | NOT NULL | Ngày nhập kho |

**tblStockOut: Lưu trữ thông tin xuất kho**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | iId | INT | PK, IDENTITY(1,1) | Mã xuất kho (tự động tăng) |
| 2 | iMedicineID | INT | NOT NULL | Mã thuốc (Liên kết với Medicine) |
| 3 | iQuantity | INT | NOT NULL | Số lượng xuất kho |
| 4 | DDateOut | DATE | NOT NULL | Ngày nhập kho |

**tblInventory: Lưu trữ thông tin tồn kho**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | iMedicineID | INT | PK, FK | Mã thuốc (Liên kết với Medicine) |
| 2 | iTotalQuantity | INT | NOT NULL, DEFAULT 0, CHECK (iTotalQuantity >= 0) | Số lượng xuất kho |

Mqh ràng buộc :

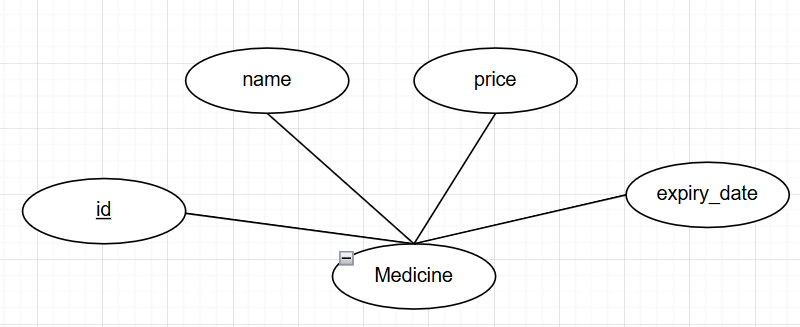
* Medicine – StockIn : 1-N
* Supplier – StockIn: 1-N
* Medine- StockOut: 1-N
* Medicine – Inventory: 1-1

**CHƯƠNG 3. TẠO CƠ SỞ DỮ LIỆU**

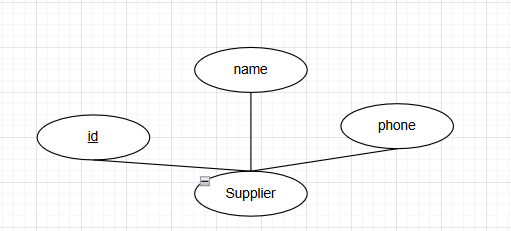
## **3.1. Tạo Database**

- Database diagram ( sơ đồ thực thể)

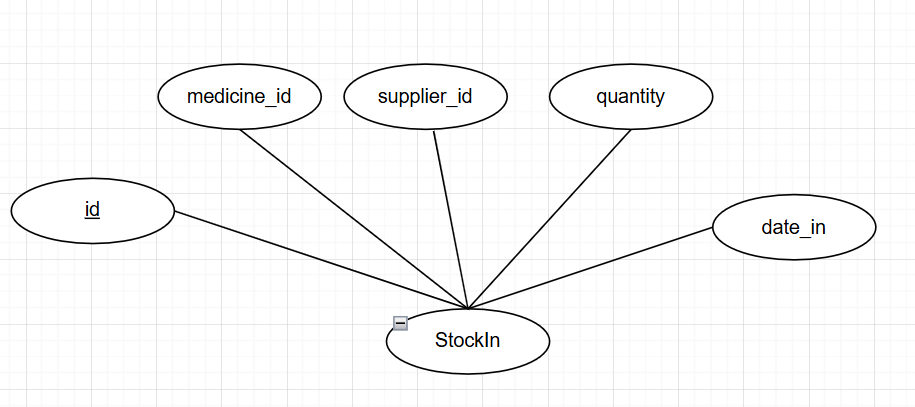
Sơ đồ thực thể Medicine (thuốc)



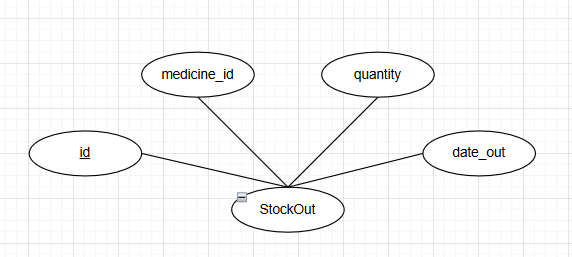
Sơ đồ thực thể Supplier (nhà phân phối)



Sơ đồ thực thể StockIn (nhập kho)



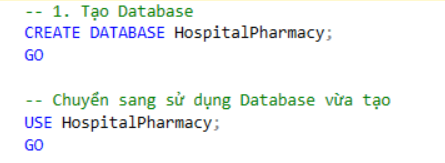
Sơ đồ thực thể StockOut (xuất kho)



Sơ đồ thực thể Inventory (tồn kho)



- Câu lệnh SQL :



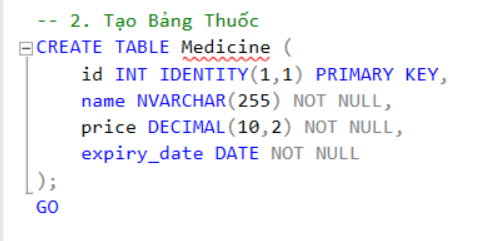
Khởi tạo cơ sở dữ liệu mới có tên là HospitalPharmacy – Quản lý kho thuốc của bệnh viện.

Khởi tạo database bằng câu lệnh : **CREATE DATABASE**.

Dùng USE để chuyển đổi ngữ cảnh sang sử dụng database HospitalPharmacy.

## **3.2. Chèn dữ liệu**

**Tạo bảng Medicine (thuốc) và chèn dữ liệu bảng**

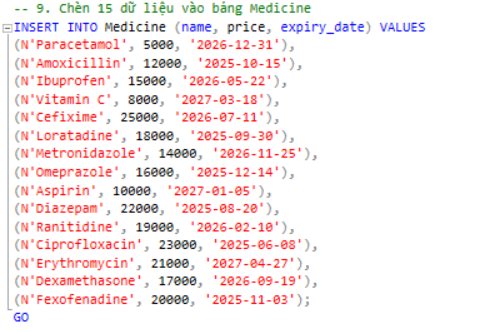


Sử dụng câu lệnh CREATE TABLE Medicine để tạo bảng thuốc với các cột:

* Cột ‘ id ’ là khóa chính của bảng để mô tả mã thuốc với thuộc tính IDENTITY(1,1) giúp id tự tăng bắt đầu từ 1 và tăng thêm 1 mỗi lần có bản ghi mới.
* Cột ‘ name ’ để lưu tên loại thuốc với kiểu dữ liệu hỗ trợ lưu trữ chuối có tối da 255 ký tự và dùng ‘N’ đảm bảo lưu được cả ký tự Unicode (tiếng việt), not null đảm bảo cột này luôn có giá trị - không được để trống.
* Cột ‘ price ’ để lưu giá của thuốc với kiểu dữ liệu demical có nghĩa là có tối đa 10 chữ số trong đó có 2 chữ số thập phân và not null đảm bảo giá thuốc luôn được nhập.
* Cột ‘ expiry\_date ’ lưu ngày hết hạn của thuốc với kiểu dữ liệu date chỉ lưu ngày ( không có giờ, phút, giây) và not nul yêu cầu ngày hết hạn phải được nhập.
* Sử dụng GO để đánh dấu kết thúc 1 batch lệnh.

Ý tưởng đoạn code :

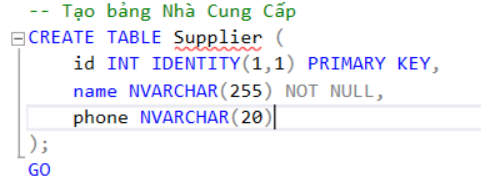
* Bảng này dùng để lưu thông tin về thuốc trong một hệ thống quản lý nhà thuốc, bệnh viện hoặc công ty dược phẩm.
* id giúp xác định duy nhất từng loại thuốc.
* name chứa tên thuốc, giúp tìm kiếm và phân loại.
* price giúp theo dõi giá bán hoặc nhập kho.
* expiry\_date giúp kiểm soát chất lượng, tránh bán thuốc hết hạn.



Ý nghĩa:

* Dùng Insert into để xác định cột dữ liệu gồm có tên thuốc, giá, hạn sử dụng.
* Có values để chèn nhiều bản ghi cùng 1 lúc, giúp tối ưu hiệu suất thay vì thực hiện nhiều lệnh Insert.

**Tạo bảng Supplier (nhà cung cấp) và chèn dữ liệu**

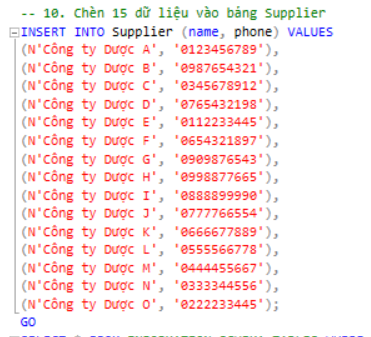
****

Sử dụng câu lệnh CREATE TABLE để tạo bảng Supplier - nhà cung cấp gồm các cột:

* Cột ‘ id ’ là khóa chính để mô tả mã của nhà cung cấp giúp xác định duy nhất từng nhà cung cấp. Vì là khóa chính nên giá trị id không được trùng lặp hoặc để trống.
* Cột ‘ name’ để lưu tên nhà cung cấp với kiểu dữ liệu hỗ trợ 255 ký tự Unicode và not null bắt buộc phải có giá trị ( không được để trống).
* Cột ‘ phone ’ để lưu số điện thoại các nhà cung cấp . Dùng NVARCHAR thay vì INT vì số điện thoại có thể bắt đầu bằng số 0, một số điện thoại có thể chứa dấu + cho mã quốc gia, không nhất thiết phải có số điện thoại nên dùng not null.

Ý tưởng:

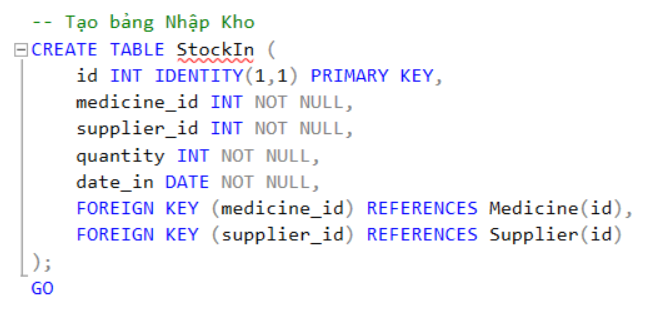
* Bảng Supplier dùng để lưu thông tin các công ty hoặc cá nhân cung cấp thuốc cho hiệu thuốc, bệnh viện, hoặc cơ sở y tế.
* id giúp quản lý từng nhà cung cấp một cách duy nhất.
* name giúp phân biệt nhà cung cấp.
* phone giúp liên hệ khi cần đặt hàng hoặc giải quyết vấn đề.

****

Ý nghĩa:

* Dùng Insert into để xác định cột dữ liệu gồm có tên nhà cung cấp, số điện thoại nhà cung cấp.
* Có values để chèn nhiều bản ghi cùng 1 lúc, giúp tối ưu hiệu suất thay vì thực hiện nhiều lệnh Insert.

**Tạo bảng StockIn (nhập kho) và chèn dữ liệu**

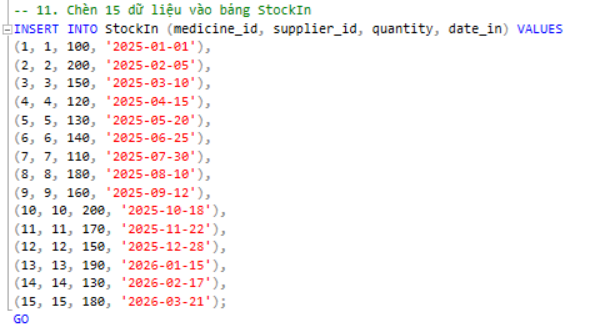
****

Sử dụng CREATE TABLE để tạo bảng StockIn – nhập kho với các cột:

* Cột ‘id’ là khóa chính để phân biệt từng lần nhập hàng, có IDENTITY giúp tự dộng tăng từ 1, mỗi lần nhập sẽ có id mới.
* Cột ‘ medicine\_id ’ để lưu ID của thuốc (lấy từ bảng Medicine) : đây là khóa ngoại liên kết với bảng Medicine, đảm bảo chỉ có thể nhập các loại thuốc đã tồn tại trong bảng Medicine với not null nếu không có thuốc thì không thể nhập hàng.
* Cột ‘ supplier-id ’ để lưu mã nhà cung cấp ( từ bảng Supplier ): đây là khóa ngoại liên kết với bảng Supplier , đảm bảo chỉ có thể nhập hàng từ nhà cung cấp có trong bảng Supplier và not null nếu có nhà cung ấp thì mới được nhập hàng.
* Cột ‘quantity ’ để lưu số lượng thuốc nhập vào với not null phải có số luọng mới được nhập hàng và cần ràng buộc check ( quantity > 0) để đảm bảo số lượng hợp lê.
* Cột ‘ date\_in’ lưu ngày nhập hàng và có thêm ràng buộc check để kông cho nhập ngày tương lai.
* Dùng foreign key (ràng buộc khóa ngoại) để đảm bỏ medicine\_id , supplier\_id chỉ tham chiếu đến thuốc và nhà cung cấp đã tồn tại.

Ý tưởng:

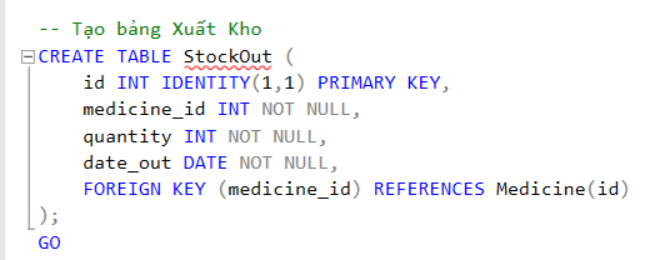
* id → Định danh duy nhất cho mỗi lần nhập hàng.
* medicine\_id → Xác định loại thuốc được nhập.
* supplier\_id → Xác định nhà cung cấp thuốc.
* quantity → Số lượng thuốc nhập vào.
* date\_in → Ngày nhập thuốc vào kho.

****

Ý nghĩa:

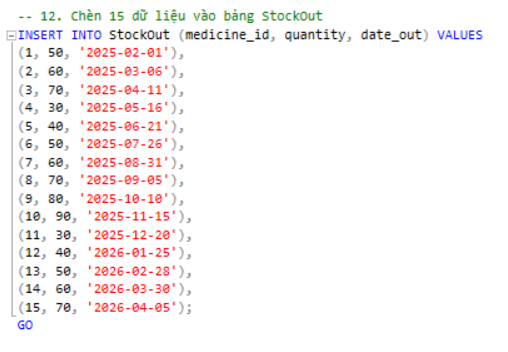
* Dùng Insert into để xác định cột dữ liệu gồm có mã thuốc, mã nhà cung cấp, số lượng, ngày nhập.
* Có values để chèn nhiều bản ghi cùng 1 lúc, giúp tối ưu hiệu suất thay vì thực hiện nhiều lệnh Insert.

**Tạo bảng StockOut (xuất kho) và chèn dữ liệu**

****

Sử dụng CREATE TABLE để tạo bảng StockOut – xuất kho với các cột:

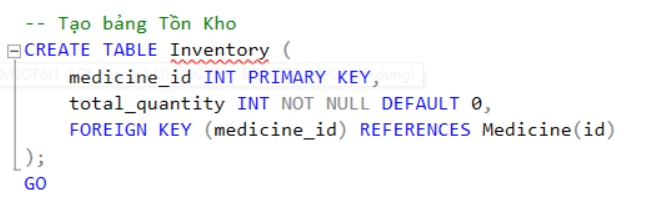
* Cột ‘ id ’ là khóa chính giúp phân biệt từng lần xuất kho.
* Cột ‘ medicine\_id ’ xác định loại thuốc được xuất : là khóa ngoại , đảm bảo thuốc xuất kho phải là thuốc có trong bảng Medicine.
* Cột ‘ quantity ’ để lưu số lượng thuốc đã xuất kho và cần ràng buộc check để tránh nhập sai dữ liệu.
* Cột ‘date\_ out’ để lưu ngày xuất kho và có ràng buộc check để không cho xuất kho vào ngày tương lai.

****

Ý tưởng:

* Bảng StockOut giúp theo dõi các lần xuất thuốc ra khỏi kho, phục vụ cho:
* Bán thuốc tại nhà thuốc.
* Chuyển thuốc đến chi nhánh khác hoặc bệnh viện.
* Hủy bỏ thuốc khi hết hạn hoặc không đạt chất lượng

**Tạo bảng Inventor ( bảng tồn kho) và chèn dữ liệu**

****

Sử dụng CREATE TABLE để tạo bảng Inventory – tồn kho với các cột:

* Cột medicine\_id là kháo chính để dâij diện cho từng loại thuốc.
* Cột total\_quantity lưu tổng số lượng thuốc tồn kho, mặc định là 0 nếu chưa nhập thuốc
* Dùng khóa ngoại để đảm bảo thuốc trong inventory phải tồn tại trong medicine.

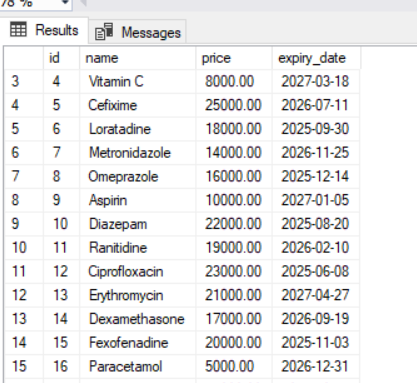
****

Do số liệu trong tồn kho bị lỗi nên cần xóa dữ liệu cũ để làm mới dữ liệu:

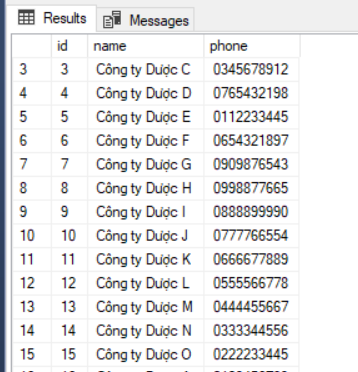
* Dùng câu lệnh DELETE để xóa dữ liệu cũ
* Sau đó chèn dữ liệu mới, khởi tạo lại tồn kho với câu lệnh Inset into. Dùng select lấy medicine\_id từ bảng stockin, dùng SUM để tính tổng số lượng nhập của mỗi loại thuốc.
* Dùng SELECT\*FROM để kiểm tra lại Inventory sau khi đã cập nhật.

## **3.3. In bảng dữ liệu**

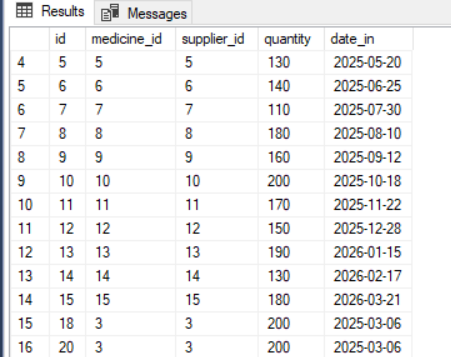
SELECT\*FROM tbl Medicine



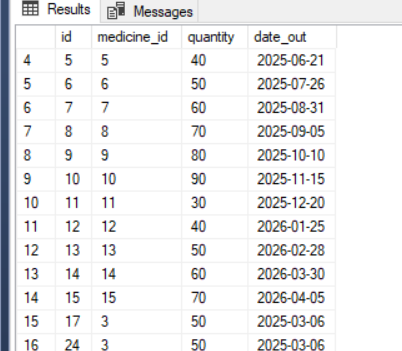
SELECT\*FROM tbl Supplier



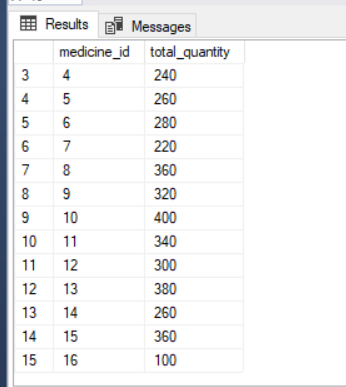
SELECT\*FROM tbl StockIN



SELECT\*FROM tbl StockOut

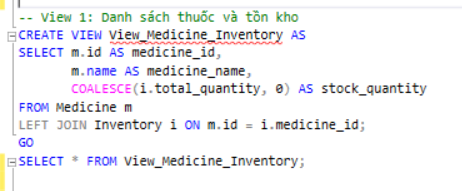


SELECT\*FROM tbl Inventory

****

# **CHƯƠNG 4 . XÂY DỰNG CÁC VIEW**

## **4.1. Danh sách thuốc và tồn kho**

****

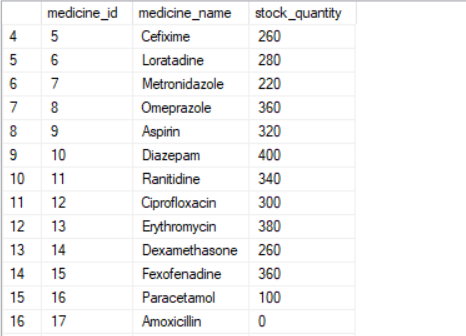
- Sử dụng câu lệnh CREATE VIEW để 1 tạo view mới. View này sẽ cung cấp thông tin về các loại thuốc và số lượng tồn kho của chúng.

- Dùng SELECT để chọn : lấy id thuốc từ bảng Medicine và gán tên cột là medicine\_id , lấy tên của thuốc từ bảng Medicine và gán tên cột là medicine\_name. Sau đó dùng hàm COALESCE để thay thế giá trị NULL ( khi không có thông tin tồn kho) bằng giá trị 0.

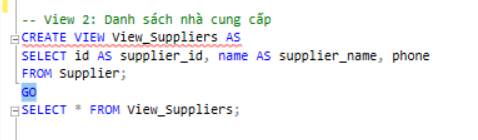
- Dùng LEFT JOIN để kết nối 2 bảng. Nó đảm bảo rằng tất cả các dòng từ bảng bên trái (Medicine trong trường hợp này) sẽ được giữ lại, ngay cả khi không có bản ghi tương ứng trong bảng bên phải (Inventory). Bảng Inventory (được biệt danh là i) chứa thông tin về số lượng tồn kho của các loại thuốc. Câu lệnh ON m.id = i.medicine\_id xác định rằng các bản ghi từ bảng Medicine và Inventory sẽ được kết nối qua cột id trong Medicine và cột medicine\_id trong Inventory.

- Ý tưởng: tạo ra một view tên View\_Medicine\_Inventory, hiển thị danh sách tất cả các loại thuốc từ bảng Medicine và số lượng tồn kho của chúng từ bảng Inventory. Nếu thuốc không có thông tin tồn kho, số lượng tồn kho sẽ được thay bằng 0. Điều này giúp dễ dàng theo dõi tồn kho của các loại thuốc mà không cần phải viết lại câu lệnh SQL phức tạp mỗi lần truy vấn.

- Kết quả chạy đoạn code này :

****

## **4.2. Danh sách nhà cung cấp**

****

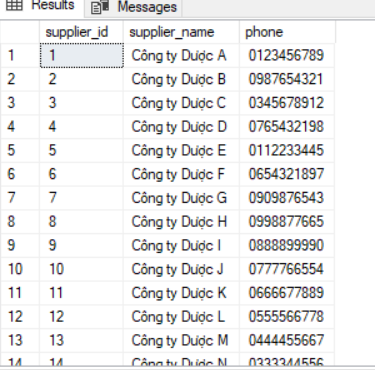
- Dùng CREATE VIEW để tạo view nới. View này cung cấp danh sách các nhà phân phối.

- Dùng SELECT để truy vấn các cột :

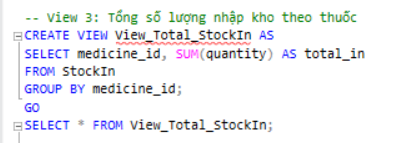
* id từ bảng Supplier và đổi tên thành supplier\_id.
* name từ bảng Supplier và đổi tên thành supplier\_name.
* phone từ bảng Supplier.

- Ý tưởng: Tạo một view để hiển thị danh sách các nhà cung cấp với thông tin bao gồm supplier\_id, supplier\_name, và phone từ bảng Supplier, giúp dễ dàng truy vấn thông tin nhà cung cấp mà không cần viết lại truy vấn phức tạp.

- Kết quả chạy:



## **4.3. Tổng số lượng nhập kho theo thuốc**



- Dùng CREATE VIEW tạo view mới để cung cấp tổng số luọng nhập kho theo thuốc.

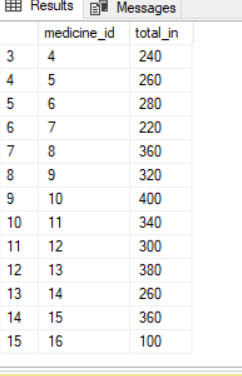
- Dùng SELECT truy vấn :

* medicine\_id: Lấy medicine\_id từ bảng StockIn, tương ứng với mã thuốc.
* SUM(quantity) AS total\_in: Tính tổng số lượng thuốc đã nhập vào kho (quantity) cho mỗi loại thuốc (mã thuốc). Sử dụng hàm SUM để cộng tất cả các giá trị quantity của cùng một medicine\_id.
* AS total\_in: Đổi tên cột tổng số lượng nhập kho thành total\_in.

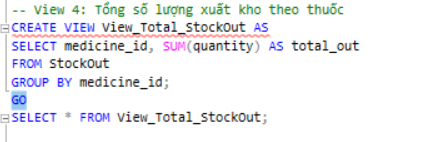
- GROUP BY medicine\_id: Nhóm các bản ghi theo medicine\_id, tức là kết quả sẽ tính tổng số lượng thuốc cho mỗi loại thuốc riêng biệt, dựa trên mã thuốc (medicine\_id).

- Ý tưởng: Tạo một view tên View\_Total\_StockIn để tính tổng số lượng nhập kho của mỗi loại thuốc từ bảng StockIn. View này nhóm các bản ghi theo medicine\_id và tính tổng số lượng cho mỗi thuốc, giúp dễ dàng theo dõi tổng lượng thuốc đã nhập vào kho.

- Kết quả chạy code:



## **4.4. Tổng số lượng xuất kho theo thuốc**

****

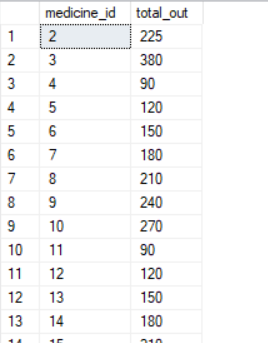
**-** Dùng CREATE VIEW để tạo 1 vỉew mới để cung cấp kết quả tổng số lượng xuất kho theo thuốc.

- Dùng SELECT để truy vấn:

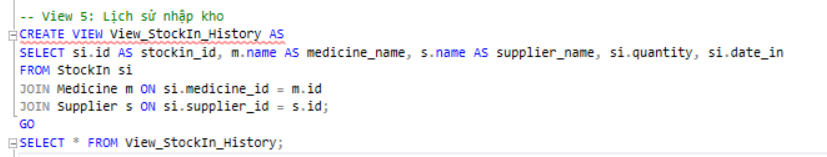
* medicine\_id: Lấy medicine\_id từ bảng StockOut, tương ứng với mã thuốc.
* SUM(quantity) AS total\_out: Tính tổng số lượng thuốc đã xuất kho (quantity) cho mỗi loại thuốc (mã thuốc). Hàm SUM sẽ cộng tất cả các giá trị quantity của cùng một medicine\_id.
* AS total\_out: Đổi tên cột tổng số lượng xuất kho thành total\_out.
* GROUP BY medicine\_id: Nhóm các bản ghi theo medicine\_id, tức là kết quả sẽ tính tổng số lượng thuốc đã xuất kho cho mỗi loại thuốc riêng biệt, dựa trên mã thuốc (medicine\_id).

-Ý tuỏng : Tạo một view tên View\_Total\_StockOut để tính tổng số lượng xuất kho của mỗi loại thuốc từ bảng StockOut. View này nhóm các bản ghi theo medicine\_id và tính tổng số lượng cho mỗi thuốc, giúp dễ dàng theo dõi tổng lượng thuốc đã xuất khỏi kho.

- Kết quả chạy:

****

## **4.5. Lịch sử nhập kho**

****

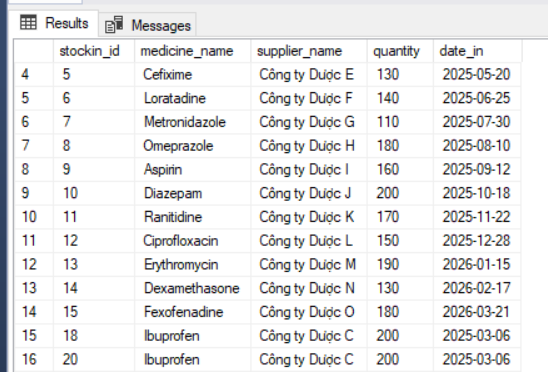
- Dùng CREATE VIEW để tạo view mới để cung cấp thông tin lịch sử nhập kho thuốc.

- Dùng SELECT để truy vấn:

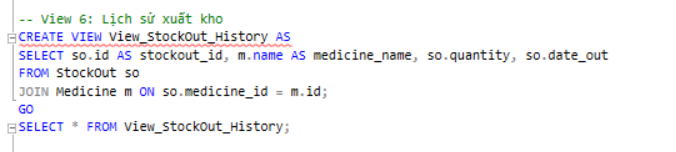
* si.id AS stockin\_id: Lấy id từ bảng StockIn và gán tên cột là stockin\_id (mã nhập kho).
* m.name AS medicine\_name: Lấy tên thuốc từ bảng Medicine và gán tên cột là medicine\_name.
* s.name AS supplier\_name: Lấy tên nhà cung cấp từ bảng Supplier và gán tên cột là supplier\_name.
* si.quantity: Lấy số lượng thuốc nhập vào kho từ bảng StockIn.
* si.date\_in: Lấy ngày nhập kho từ bảng StockIn.
* JOIN Medicine m ON si.medicine\_id = m.id: Kết nối bảng StockIn với bảng Medicine (được biệt danh là m) qua cột medicine\_id trong StockIn và cột id trong Medicine. Điều này giúp lấy tên thuốc cho mỗi bản ghi nhập kho.
* JOIN Supplier s ON si.supplier\_id = s.id: Kết nối bảng StockIn với bảng Supplier (được biệt danh là s) qua cột supplier\_id trong StockIn và cột id trong Supplier. Điều này giúp lấy tên nhà cung cấp cho mỗi bản ghi nhập kho.

-Ý tưởng ngắn gọn: Tạo một view tên View\_StockIn\_History để hiển thị lịch sử nhập kho thuốc, bao gồm thông tin về mã nhập kho, tên thuốc, tên nhà cung cấp, số lượng và ngày nhập kho. View này kết hợp thông tin từ ba bảng: StockIn, Medicine, và Supplier.

- Kết quả chạy dữ liệu:

****

## **4.6. Lịch sử nhập kho**

****

- CREATE VIEW View\_StockOut\_History: Tạo một view mới có tên View\_StockOut\_History để lưu trữ kết quả của truy vấn này. View này sẽ cung cấp thông tin về lịch sử xuất kho thuốc.

- SELECT so.id AS stockout\_id, m.name AS medicine\_name, so.quantity, so.date\_out:

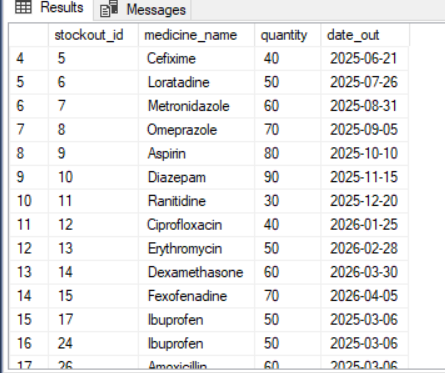
* so.id AS stockout\_id: Lấy id từ bảng StockOut và gán tên cột là stockout\_id (mã xuất kho).
* m.name AS medicine\_name: Lấy tên thuốc từ bảng Medicine và gán tên cột là medicine\_name.
* so.quantity: Lấy số lượng thuốc xuất khỏi kho từ bảng StockOut.
* so.date\_out: Lấy ngày xuất kho từ bảng StockOut.

- FROM StockOut so: Truy vấn dữ liệu từ bảng StockOut (được biệt danh là so), nơi lưu trữ thông tin xuất kho.

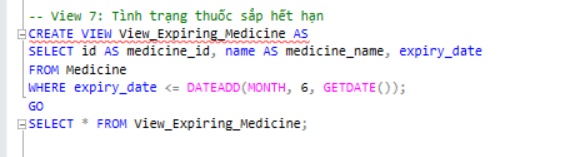
- JOIN Medicine m ON so.medicine\_id = m.id: Kết nối bảng StockOut với bảng Medicine (được biệt danh là m) qua cột medicine\_id trong StockOut và cột id trong Medicine. Điều này giúp lấy tên thuốc cho mỗi bản ghi xuất kho.

- Ý tưởng: Tạo một view tên View\_StockOut\_History để hiển thị lịch sử xuất kho thuốc, bao gồm thông tin về mã xuất kho, tên thuốc, số lượng và ngày xuất kho. View này kết hợp thông tin từ hai bảng: StockOut và Medicine.

- Kết quả chạy tương ứng:

****

## **4.7. Tình trạng thuốc sắp hết hạn**

****

- CREATE VIEW View\_Expiring\_Medicine: Tạo một view mới có tên View\_Expiring\_Medicine để lưu trữ kết quả của truy vấn này. View này sẽ cung cấp thông tin về các loại thuốc sắp hết hạn trong vòng 6 tháng tới.

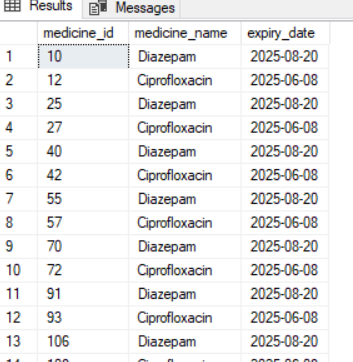
- Dùng SELECT để truy vấn lấy id, name, expiry\_date.

- HERE expiry\_date <= DATEADD(MONTH, 6, GETDATE()):

* Câu lệnh này lọc các thuốc có ngày hết hạn trong vòng 6 tháng tới.
* DATEADD(MONTH, 6, GETDATE()): Hàm này tính ngày hiện tại cộng thêm 6 tháng.
* expiry\_date <= DATEADD(MONTH, 6, GETDATE()): Lọc những thuốc có ngày hết hạn nhỏ hơn hoặc bằng ngày hiện tại cộng thêm 6 tháng.

- Ý tưởng: tạo một view tên View\_Expiring\_Medicine để hiển thị các loại thuốc sắp hết hạn trong vòng 6 tháng tới. View này lọc những thuốc có ngày hết hạn gần với ngày hiện tại và giúp quản lý thuốc dễ dàng hơn trước khi hết hạn.

- Kết quả chạy chương trình:

****

## **4.8. Tình trạng tồn kho dưới mức tối thiểu (10 đơn vị)**

****

- CREATE VIEW View\_Low\_Stock: Tạo một view mới có tên View\_Low\_Stock để lưu trữ kết quả của truy vấn này. View này sẽ hiển thị các loại thuốc có tồn kho dưới mức tối thiểu (dưới 10 đơn vị).

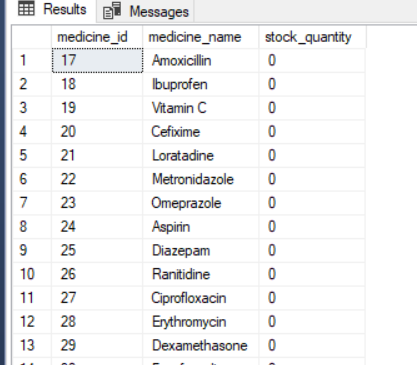
- Dùng SELECT truy vấn lấy id, name.

- Sử dụng hàm COALESCE để thay thế giá trị NULL (nếu có) trong cột total\_quantity của bảng Inventory bằng 0, sau đó đổi tên cột thành stock\_quantity.

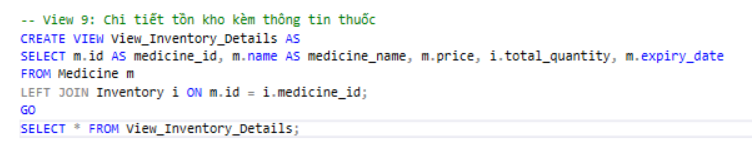
- WHERE COALESCE(i.total\_quantity, 0) < 10: Lọc các thuốc có tồn kho dưới 10 đơn vị, bao gồm những thuốc không có thông tin tồn kho (sử dụng COALESCE để thay thế NULL bằng 0)

- Ý tưởng: Tạo một view tên View\_Low\_Stock để hiển thị các loại thuốc có số lượng tồn kho dưới 10 đơn vị, giúp dễ dàng theo dõi và quản lý các thuốc cần được bổ sung vào kho.

- Kết quả :

****

## **4.9. Chi tiết tồn kho kèm thông tin thuốc**

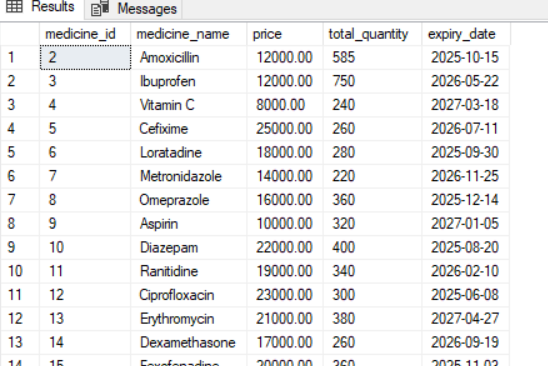
****

- CREATE VIEW View\_Inventory\_Details: Tạo một view mới có tên View\_Inventory\_Details để lưu trữ kết quả của truy vấn này. View này sẽ hiển thị thông tin chi tiết về thuốc trong kho, bao gồm tên thuốc, giá, số lượng tồn kho và ngày hết hạn.

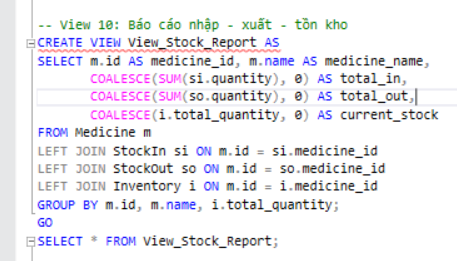
- Dùng SELECT truy vấn id, name, price, total quantity, expiry date.

- LEFT JOIN Inventory i ON m.id = i.medicine\_id: Kết nối bảng Medicine với bảng Inventory (được biệt danh là i) qua cột medicine\_id trong bảng Inventory và cột id trong bảng Medicine. Sử dụng LEFT JOIN để đảm bảo rằng tất cả các thuốc từ bảng Medicine đều xuất hiện, kể cả những thuốc không có thông tin tồn kho (trong trường hợp chưa từng nhập vào kho).

**-** Kết quả :

****

## **4.10. Báo cáo nhập- xuất- tồn kho**

****

- Tạo một view tên View\_Stock\_Report để hiển thị báo cáo tổng hợp về số lượng thuốc nhập, xuất và tồn kho.

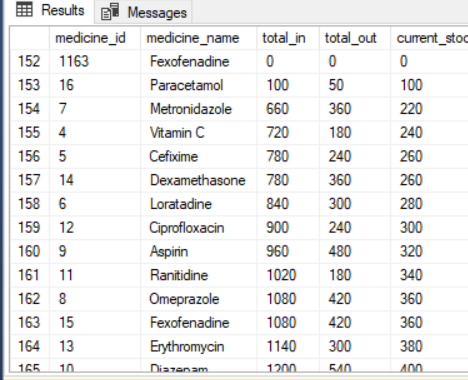
- Dùng SELECT truy vấn id, name.

- Dùng hàm COALESCE để nếu không có dữ liệu nhập, hiển thị 0 thay vì NULL. Dùng SUM để tính tổng số lượng thuốc đã nhập từ bảng StockIn.

- Dùng LEFT JOIN dùng để kết nối các bảng.

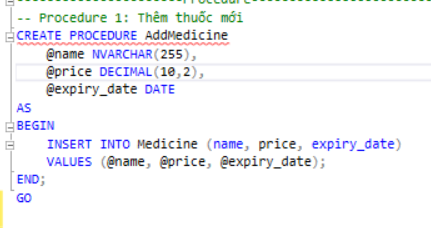
- Ý tưởng: Tạo một view tên View\_Stock\_Report để tổng hợp thông tin về số lượng thuốc nhập, xuất và tồn kho. View này giúp dễ dàng theo dõi tình trạng kho hàng của từng loại thuốc trong hệ thống.

- Kết quả :

****

# **CHƯƠNG 5. XÂY DỰNG CÁC PROCEDURE**

## **5.1. Procedure thêm thuốc mới**

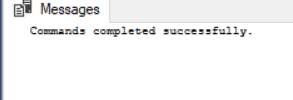
****

- Dùng CREATE để tạo 1 procedure có tên AddMedicine để thêm thuốc mới vào bảng Medicine.

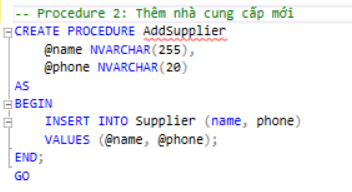
- Khai báo tham số đầu vào : @name, @price, @expiry\_date.

- Chèn dữ liệu mới vào bảng Medicine bằng Insert Into.

- Kết quả:

****

## **5.2. Procedure thêm nhà cung cấp mới**

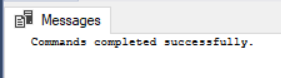
****

- Tạo một Stored Procedure có tên AddSupplier để thêm nhà cung cấp mới vào bảng Supplier.

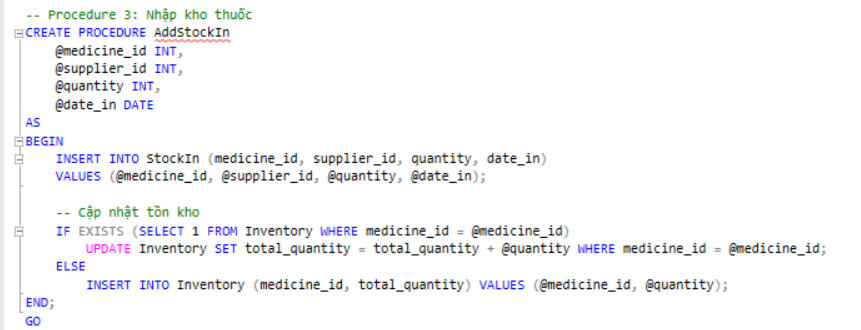
- Khai báo tham số đầu vào :@name, @phone

- Chèn dữ liệu đầu vào với tên và số điện thoại người cung cấp

- Kết quả:

****

## **5.3. Procedure nhập kho thuốc**



- Tạo một Stored Procedure để hêm thông tin nhập kho thuốc.

- Nhận tham số đầu vào về id thuốc, id nhà cung cấp, số lượng nhập thuốc, ngày nhập kho.

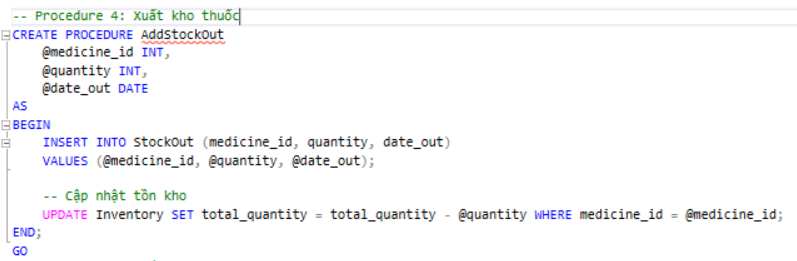
- Chèn dữ liệu mới vào bảng StockIn.

- Kiểm tra thuốc có trong Inventory chưa. Nếu có rồi thì cộng thêm số lượng nhập vào tổng số lượng tồn kho. Nếu chưa thì chèn thêm dữ liệu mới.

- Kết quả:

****

## **5.4. Procedure xuất kho thuốc**

****

- Tạo một Stored Procedure để hêm thông tin xuất kho thuốc.

- Nhận tham số đầu vào về id thuốc, số lượng xuất thuốc, ngày xuất kho.

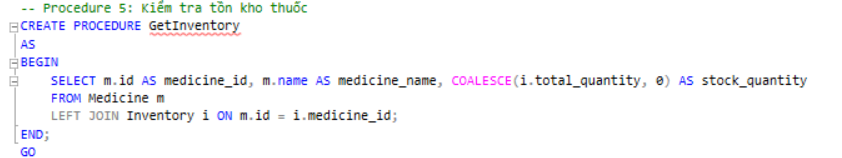
- Chèn dữ liệu mới vào bảng StockOut.

- Cập nhật số lượng tồn kho trong Inventory: giảm số lượng tồn kho theo số lượng xuất.

- Kết quả:

****

## **5.5. Procedure kiểm tra tồn kho thuốc**

****

- Tạo Stored Procedure GetInventory được sử dụng để lấy danh sách thuốc cùng với số lượng tồn kho hiện tại.

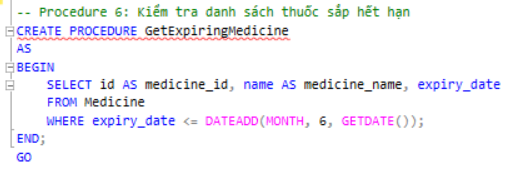
- Dùng SELECT truy vấn lấy dữ liệu:

* Medicine m: Bảng chứa thông tin về các loại thuốc.
* Inventory i: Bảng lưu trữ số lượng thuốc trong kho.
* LEFT JOIN Inventory i ON m.id = i.medicine\_id:
* Giữ tất cả các thuốc trong bảng Medicine, kể cả những thuốc chưa có trong kho (tức là chưa nhập hàng).
* COALESCE(i.total\_quantity, 0) AS stock\_quantity:
* Nếu thuốc chưa có trong Inventory, đặt stock\_quantity = 0 thay vì NULL.

- Kết quả:

****

## **5.6. Procedure kiểm tra danh sách thuốc sắp hết hạn**

****

- Stored Procedure GetExpiringMedicine được sử dụng để lấy danh sách các loại thuốc sắp hết hạn trong vòng 6 tháng kể từ ngày hiện tại.

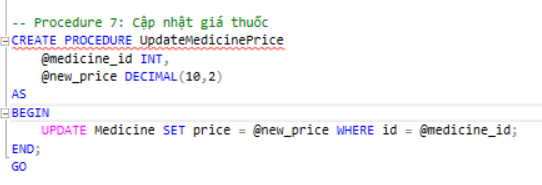
- Dùng SELECT truy vấn lấy dữ liệu:

* Lấy danh sách thuốc (id, name, expiry\_date) từ bảng Medicine.
* WHERE expiry\_date <= DATEADD(MONTH, 6, GETDATE()):
* Lọc các thuốc có ngày hết hạn (expiry\_date) ≤ ngày hiện tại + 6 tháng.
* GETDATE(): Lấy ngày hiện tại.
* DATEADD(MONTH, 6, GETDATE()): Tính ngày hiện tại + 6 tháng.

- Kết quả:

****

## **5.7. Procedure cập nhật giá thuốc**

****

- Stored Procedure UpdateMedicinePrice được sử dụng để cập nhật giá bán của một loại thuốc cụ thể trong bảng Medicine.

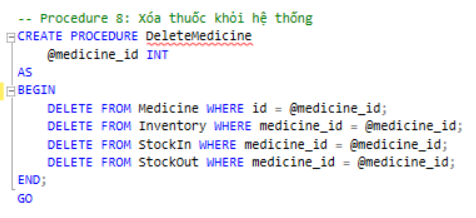
- Cập nhật giá thuốc :

* SET price = @new\_price: Gán giá thuốc mới (@new\_price) vào cột price.
* WHERE id = @medicine\_id: Xác định thuốc cần cập nhật theo medicine\_id.

- Kết quả:

****

## **5.8. Procedure xóa thuốc khỏi hệ thống**

****

****

## **5.9. Procedure lấy thông tin nhà cung cấp**

****

## **5.10. Proceudre lấy báo cáo tổng hợp nhập-xuất-tồn kho**

****

# **CHƯƠNG 6. XÂY DỰNG CÁC TRIGGER**

## **6.1. Trigger tự động cập nhật tồn kho khi nhập kho**

**A screenshot of a computer

AI-generated content may be incorrect.**

* CREATE TRIGGER trg\_AfterStockIn: Tạo trigger có tên trg\_AfterStockIn.
* ON StockIn: Trigger được gắn vào bảng StockIn.
* AFTER INSERT: Kích hoạt trigger sau khi dữ liệu được chèn vào StockIn.
* BEGIN ... END: Phần thân chứa logic xử lý.
* UPDATE Inventory: Cập nhật bảng Inventory.
* SET total\_quantity = total\_quantity + i.quantity: Tăng tổng số lượng thuốc tồn kho bằng số lượng thuốc vừa được nhập (i.quantity).
* JOIN inserted i:
* inserted là bảng tạm chứa các bản ghi vừa được thêm vào StockIn.
* Kết hợp bảng Inventory và inserted qua cột medicine\_id để cập nhật chính xác thuốc tương ứng.
* **Thuốc có medicine\_id = 1**: Thêm 500 đơn vị vào tồn kho.
* **Thuốc có medicine\_id = 3**: Thêm 200 đơn vị vào tồn kho.
* Sau khi chạy trigger:
* Số lượng tồn kho của thuốc có medicine\_id = 1 sẽ tăng thêm 500.
* Số lượng tồn kho của thuốc có medicine\_id = 3 sẽ tăng thêm 200.

## **6.2. Trigger tự động cập nhật tồn kho khi xuất kho**

A screenshot of a computer program

AI-generated content may be incorrect.

* CREATE TRIGGER trg\_AfterStockOut: Tạo trigger có tên trg\_AfterStockOut.
* ON StockOut: Trigger được gắn vào bảng StockOut.
* AFTER INSERT: Kích hoạt trigger sau khi dữ liệu được thêm vào StockOut.
* BEGIN ... END: Phần thân chứa logic xử lý.
* UPDATE Inventory: Cập nhật bảng Inventory.
* SET total\_quantity = total\_quantity - i.quantity: Giảm số lượng tồn kho bằng số lượng thuốc vừa xuất (i.quantity).
* JOIN inserted i:
* inserted là bảng tạm chứa các bản ghi vừa được thêm vào StockOut.
* Kết hợp bảng Inventory và inserted qua cột medicine\_id để cập nhật chính xác thuốc tương ứng.

## **6.3. Trigger ngăn chặn xuất kho nếu số lượng không đủ**

A screenshot of a computer program

AI-generated content may be incorrect.

* CREATE TRIGGER trg\_BeforeStockOut: Tạo trigger có tên trg\_BeforeStockOut.
* ON StockOut: Trigger được gắn vào bảng StockOut.
* INSTEAD OF INSERT: Trigger này chạy thay thế cho lệnh INSERT thông thường trên bảng StockOut.
* IF EXISTS: Kiểm tra xem có bản ghi nào trong bảng inserted mà:
* Số lượng cần xuất (i.quantity) lớn hơn số lượng tồn kho hiện có (inv.total\_quantity).
* RAISERROR ('Không đủ hàng trong kho để xuất!', 16, 1);: Hiển thị thông báo lỗi nếu điều kiện trên đúng.
* RETURN;: Dừng việc thêm dữ liệu vào bảng StockOut.
* ELSE: Nếu số lượng hợp lệ, dữ liệu từ bảng inserted sẽ được thêm vào bảng StockOut.

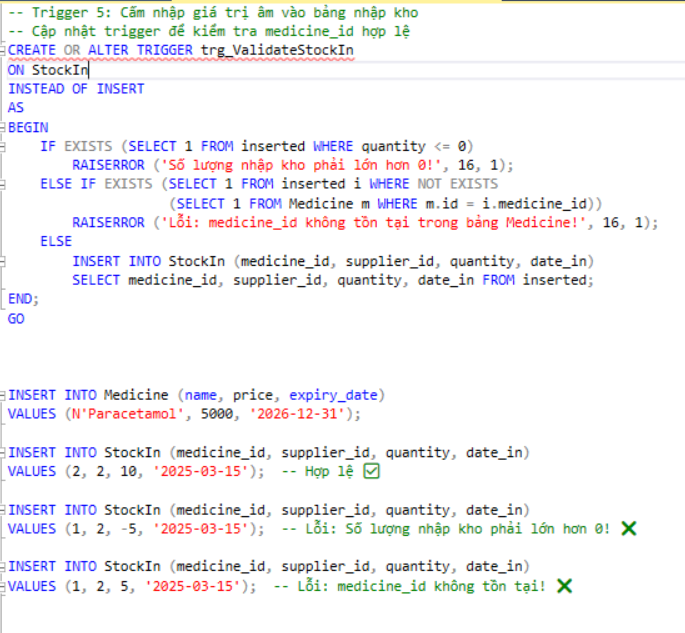
## **6.4. Trigger xóa dữ liệu tồn kho khi xóa thuốc**

A screenshot of a computer

AI-generated content may be incorrect.

* **AFTER DELETE**: Trigger này chạy **sau khi** một bản ghi trong bảng Medicine bị xóa thành công.
* **deleted**: Đây là bảng ảo chứa các bản ghi vừa bị xóa trong bảng Medicine.
* **DELETE FROM Inventory WHERE medicine\_id IN (SELECT id FROM deleted);**
* Xóa toàn bộ các bản ghi trong Inventory có medicine\_id trùng với các bản ghi vừa bị xóa khỏi Medicine.

## **6.5. Trigger cấm nhập giá trị âm vào bảng nhập kho**



* **INSTEAD OF INSERT**: Trigger sẽ **ngăn chặn thao tác INSERT trực tiếp** vào bảng StockIn nếu vi phạm các điều kiện kiểm tra.
* **inserted**: Bảng ảo chứa các bản ghi mới đang được thêm vào bảng StockIn.
* **IF EXISTS (SELECT 1 FROM inserted WHERE quantity <= 0)**
* Nếu có bản ghi nào trong inserted có quantity <= 0, trigger sẽ hiển thị lỗi:  
   "Số lượng nhập kho phải lớn hơn 0!"
* **ELSE IF EXISTS (SELECT 1 FROM inserted i WHERE NOT EXISTS (...) )**
* Nếu có medicine\_id trong inserted mà **không tồn tại** trong bảng Medicine, trigger sẽ hiển thị lỗi:  
  "Lỗi: medicine\_id không tồn tại trong bảng Medicine!"
* **INSERT INTO StockIn ... SELECT ... FROM inserted**
* Nếu cả hai điều kiện trên đều không vi phạm, trigger sẽ thêm các bản ghi hợp lệ vào bảng StockIn.

## **6.6. Trigger cấm nhập giá trị âm vào bảng xuất kho**

A screenshot of a computer program

AI-generated content may be incorrect.

* **AFTER INSERT**: Trigger này được kích hoạt **sau khi thêm dữ liệu** vào bảng StockOut.
* **inserted**: Bảng ảo chứa các bản ghi vừa được thêm vào StockOut.
* **IF EXISTS (SELECT 1 FROM inserted WHERE quantity <= 0)**
* Bản ghi đầu tiên thành công → dữ liệu được thêm vào bảng stockout.
* Bản ghi thứ hai thất bại → lỗi do quantity âm.
* Bản ghi thứ ba thất bại → lỗi do quantity bằng 0.

## **6.7. Trigger kiểm tra hạn sử dụng khi nhập kho**

A screenshot of a computer code

AI-generated content may be incorrect.

* **ALTER TRIGGER**: Dùng để sửa trigger hiện có (hoặc bạn cần CREATE TRIGGER nếu trigger chưa tồn tại).
* **ON StockIn AFTER INSERT**: Kích hoạt trigger sau khi thêm dữ liệu vào bảng StockIn.
* **INSERTED**: Bảng ảo lưu dữ liệu mới được thêm vào StockIn.
* **IF EXISTS**: Kiểm tra xem có thuốc nào trong danh sách thêm mới đã hết hạn không.
* **JOIN Medicine**: Kết hợp với bảng Medicine để kiểm tra hạn sử dụng của thuốc.
* **WHERE m.expiry\_date < GETDATE()**: Điều kiện để xác định thuốc đã hết hạn.
* **RAISERROR**: Thông báo lỗi nếu tìm thấy thuốc hết hạn.
* **ROLLBACK TRANSACTION**: Hủy giao dịch nếu điều kiện đúng.

## **6.8. Trigger kiểm tra cập nhật giá thuốc**

A screenshot of a computer program

AI-generated content may be incorrect.

* **CREATE TRIGGER**: Tạo trigger mới.
* **ON Medicine AFTER UPDATE**: Kích hoạt trigger sau khi cập nhật dữ liệu trong bảng Medicine.
* **INSERTED**: Bảng ảo chứa dữ liệu mới được cập nhật.
* **IF EXISTS**: Kiểm tra xem có thuốc nào bị cập nhật với giá nhỏ hơn 0 không.
* **RAISERROR**: Thông báo lỗi nếu phát hiện giá trị âm.
* **ROLLBACK TRANSACTION**: Hủy giao dịch nếu phát hiện lỗi.

## **6.9. Trigger tạo nhật ký khi có nhập kho**

A screenshot of a computer code

AI-generated content may be incorrect.

* **StockLog**: Bảng lưu trữ lịch sử giao dịch nhập kho.
* **log\_id**: Khóa chính, tự tăng.
* **action**: Hành động (ở đây là "Nhập kho").
* **medicine\_id**: ID của thuốc được nhập kho.
* **quantity**: Số lượng thuốc nhập kho.
* **log\_date**: Ngày giờ thực hiện giao dịch.

## **6.10. Trigger tạo nhật ký khi có xuất kho**

A screenshot of a computer program

AI-generated content may be incorrect.

* **CREATE TRIGGER**: Tạo trigger mới.
* **ON StockOut AFTER INSERT**: Kích hoạt sau khi thêm dữ liệu vào bảng StockOut.
* **IF EXISTS (SELECT 1 FROM inserted)**: Kiểm tra có dữ liệu vừa được thêm hay không (giúp tránh lỗi nếu lệnh INSERT trống).
* **INSERT INTO StockLog**: Chèn dữ liệu vào bảng StockLog.
* **SELECT ... FROM inserted**: Dữ liệu được thêm mới trong StockOut sẽ tự động ghi lại vào StockLog.

# **CHƯƠNG 7. PHÂN QUYỀN VÀ BẢO VỆ CÁC CƠ SỞ DỮ LIỆU**

## **7.1. Phân quyền**

**A list of goods

AI-generated content may be incorrect.**

**- Warehouse\_Manager**

* Mục đích: Quản lý kho hàng với quyền cao nhất trong các vai trò liên quan đến kho.
* Quyền hạn nên cấp:
  + SELECT, INSERT, UPDATE, DELETE trên các bảng chính (Medicine, StockIn, StockOut, StockLog).
  + EXECUTE trên các procedure liên quan đến quản lý kho.

**- Warehouse\_Staff**

* Mục đích: Nhân viên kho, thực hiện các thao tác nhập/xuất kho.
* Quyền hạn nên cấp:
  + SELECT trên bảng Medicine, StockIn, StockOut.
  + INSERT vào bảng StockIn, StockOut.
  + EXECUTE trên các procedure liên quan đến việc nhập/xuất kho**.**

**- Customer**

* Mục đích: Người dùng xem thông tin thuốc (không có quyền chỉnh sửa).
* Quyền hạn nên cấp:
  + SELECT trên bảng Medicine.

## **7.2. Quản lý kho**



* GRANT: Câu lệnh cấp quyền trong SQL.
* SELECT, INSERT, UPDATE, DELETE: Các quyền cơ bản để quản lý dữ liệu trong bảng:
* SELECT: Truy vấn dữ liệu
* INSERT: Thêm dữ liệu mới
* UPDATE: Cập nhật dữ liệu hiện có
* DELETE: Xóa dữ liệu
* ON Medicine: Chỉ định bảng Medicine là đối tượng cấp quyền.
* TO Warehouse\_Manager: Gán các quyền trên cho vai trò/người dùng Warehouse\_Manager.
* Kết quả:



## **7.3. Nhân viên kho**



* GRANT: Câu lệnh cấp quyền trong SQL.
* SELECT: Cho phép người dùng hoặc vai trò xem dữ liệu trong bảng Medicine.
* INSERT: Cho phép thêm dữ liệu mới vào bảng Medicine.
* ON Medicine: Chỉ định bảng Medicine là đối tượng cấp quyền.
* TO Warehouse\_Staff: Gán các quyền trên cho người dùng/nhóm có tên Warehouse\_Staff.
* Kết quả:



## **7.4. Khách hàng**

A close up of words

AI-generated content may be incorrect.

* GRANT: Câu lệnh cấp quyền trong SQL.
* SELECT: Cho phép người dùng/nhóm chỉ được xem dữ liệu trong bảng Medicine.
* ON Medicine: Chỉ định bảng Medicine là đối tượng cấp quyền.
* TO Customer: Gán quyền SELECT cho người dùng/nhóm có tên Customer.

## **7.5. Tạo tài khoản đăng nhập**

A close up of blue text

AI-generated content may be incorrect.

* Tạo 3 tài khoản đăng nhập (LOGIN) ở cấp máy chủ trong SQL Server.
* Mật khẩu 'password123'
* Kết quả:



## **7.6. Tạo người dùng**

A close up of words

AI-generated content may be incorrect.

* Tạo 3 USER tương ứng trong cơ sở dữ liệu (cấp cơ sở dữ liệu, không phải cấp máy chủ).
* USER liên kết với các tài khoản LOGIN đã tạo ở bước trên.
* LOGIN = Xác thực truy cập vào SQL Server.
* USER = Quản lý quyền trong từng cơ sở dữ liệu cụ thể.
* Kết quả:



## **7.7. Gán vai trò người dùng**

A close up of words

AI-generated content may be incorrect.

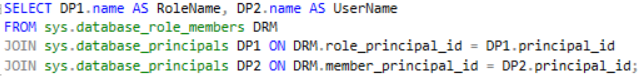
Gán các USER vào các vai trò (ROLE) có quyền cụ thể:

* admin\_user → Vai trò Warehouse\_Manager (toàn quyền trong kho thuốc).
* staff\_user → Vai trò Warehouse\_Staff (chỉ thêm và xem dữ liệu).
* customer\_user → Vai trò Customer (chỉ xem dữ liệu).

Kết quả:

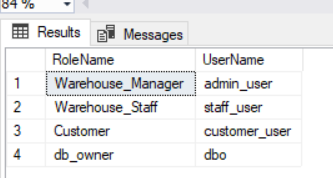


## **7.8. Kiểm tra danh sách phân quyền**



* Truy vấn danh sách các USER và vai trò mà họ thuộc về.
* Bảng hệ thống sys.database\_role\_members lưu thông tin thành viên trong vai trò.
* Bảng sys.database\_principals chứa thông tin chi tiết về vai trò và người dùng.

Kết quả:



**KẾT LUẬN**

Hệ thống quản lý kho thuốc trong bệnh viện mang lại nhiều ưu điểm đáng kể. Trước hết, hệ thống giúp tự động hóa quy trình nhập, xuất và kiểm kê thuốc, giảm thiểu sai sót trong quản lý số lượng và hạn sử dụng. Việc sử dụng các lệnh SQL như VIEW, PROCEDURE, và TRIGGER giúp cải thiện hiệu quả truy vấn dữ liệu, đồng thời đảm bảo tính toàn vẹn và nhất quán của thông tin. Phân quyền chi tiết qua các vai trò như Warehouse\_Manager, Warehouse\_Staff và Customer cũng giúp tăng cường tính bảo mật và hạn chế rủi ro truy cập trái phép.

Bên cạnh các ưu điểm, hệ thống vẫn tồn tại một số nhược điểm cần khắc phục. Quản lý mật khẩu chưa đủ mạnh có thể làm tăng nguy cơ bị xâm nhập. Việc cấp quyền trực tiếp trên các bảng dữ liệu thay vì thông qua VIEW có thể dẫn đến rò rỉ thông tin nhạy cảm. Ngoài ra, hệ thống cần thêm các cơ chế kiểm soát lỗi, chẳng hạn như ngăn chặn việc nhập trùng dữ liệu hoặc thêm thuốc đã hết hạn sử dụng.

Trong tương lai, hệ thống có thể được phát triển theo hướng tích hợp thêm giao diện người dùng trực quan để giúp nhân viên không chuyên về SQL dễ dàng thao tác. Việc ứng dụng công nghệ như AI hoặc Machine Learning có thể hỗ trợ dự báo nhu cầu thuốc, giúp tối ưu hóa kho lưu trữ. Đồng thời, việc bổ sung các báo cáo phân tích sẽ giúp ban quản lý bệnh viện ra quyết định nhanh chóng và hiệu quả hơn.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. **Giáo trình Cơ sở dữ liệu** – Tác giả: Nguyễn Xuân Huy, Nhà xuất bản Đại học Quốc gia TP.HCM.Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
2. **Hệ quản trị cơ sở dữ liệu SQL Server - Lý thuyết và thực hành** – Tác giả: Trần Quang Huy, Nhà xuất bản Bách Khoa Hà Nội.
3. **SQL: The Complete Reference** – Tác giả: James R. Groff, Paul N. Weinberg