TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP HỒ CHÍ MINH

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

****

**ĐỒ ÁN KẾT THÚC MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ỨNG DỤNG HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐỂ XÂY DỰNG HỆ THỐNG QUẢN LÝ DƯỢC PHẨM**

Giảng viên hướng dẫn: ThS. Phạm Trọng Huynh

Lớp: 09\_ĐH\_CNPM2

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| STT | Họ tên | MSSV |
| 1 | Nguyễn Lương Thùy Dương | 0950080044 |
| 2 | Vương Gia Huy | 0950080032 |
| 3 | Phạm Phú Đạt | 0950080042 |

Tp. Hồ Chí Minh, tháng 04 năm 2023

**PHIẾU ĐÁNH GIÁ CỦA GIẢNG VIÊN**

Nhận xét: ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Đánh giá: ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………….., ngày…...tháng 04 năm 2023

Giảng viên

**ThS. Phạm Trọng Huynh**

# MỤC LỤC

[MỤC LỤC 2](#_Toc132780376)

[Thông tin nhóm vá đánh giá 3](#_Toc132780377)

[LỜI CẢM ƠN 5](#_Toc132780378)

[LỜI NÓI ĐẦU 6](#_Toc132780379)

[CHƯƠNG 1. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 7](#_Toc132780380)

[1.1 LỜI GIỚI THIỆU 7](#_Toc132780381)

[1.2 PHÂN TÍCH HỆ THỐNG QUẢN LÝ DƯỢC PHẨM 7](#_Toc132780382)

[1.2.1 Thực trạng của hệ thống quản lý dược phẩm 7](#_Toc132780383)

[1.2.2 Biểu đồ phân cấp chức năng (BPC) 8](#_Toc132780384)

[1.2.3 Biểu đồ luồng dữ liệu: 9](#_Toc132780385)

[1.2.4 Phân loại các thuộc tính vào một tập thực thể 13](#_Toc132780386)

[1.2.5 Tình huống tranh chấp 14](#_Toc132780387)

[CHƯƠNG 2: ỨNG DỤNG LÝ THUYẾT (tùy vào đề tài mà tên của chương thay đổi phù hợp, hiện thực lý thuyết lên 1 HQT CSDL cụ thể) 19](#_Toc132780388)

[2.1. Xây dựng CSDL trên HQT CSDL 19](#_Toc132780389)

[2.2. Thao tác trên CSDL trên HQT CSDL 27](#_Toc132780390)

[2.3. Stored Procedure, Funtion, Trigger 34](#_Toc132780391)

[2.3.1. Stored Procedure 34](#_Toc132780392)

[2.3.2. Function 37](#_Toc132780393)

[2.3.3. Trigger 41](#_Toc132780394)

[2.4. Bảo mật, phân quyền 45](#_Toc132780395)

[2.4.1. Login Security 45](#_Toc132780396)

[2.4.2. Database user Security 50](#_Toc132780397)

[2.4.3. Permission Security 52](#_Toc132780398)

[Chương 3: KẾT LUẬN 58](#_Toc132780399)

[3.1. Những kết quả đạt được của đồ án 58](#_Toc132780400)

[3.2. Nhược điểm của đồ án 59](#_Toc132780401)

[Phụ lục 59](#_Toc132780402)

[TÀI LIỆU THAM KHẢO 60](#_Toc132780403)

# Thông tin nhóm vá đánh giá

|  |  |  |
| --- | --- | --- |
| **PHÂN TÍCH CÔNG VIỆC** | | |
| **Vương Gia Huy**  **0950080032** | **Nguyễn Lương Thùy Dương**  **0950080044** | **Phạm Phú Đạt**  **0950080042** |
| Thiết kế khái quát cơ sở dữ liệu | Biểu đồ luồng dữ liệu | Thao tác trên CSDL trên HQT CSDL |
| Store Procedure, Function, Trigger | Phân loại các thuộc tính vào một tập thực thể | Xây dựng CSDL trên HQT CSDL |
| Tình huống tranh chấp Unrepeatable Read, Phantom Read |  | Tình huống tranh chấp Dirty read vs Unrepeatable read |
| Bảo mật, phân quyền |  |  |
| Biểu đồ phân cấp chức năng |  |  |

**ĐÁNH GIÁ**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Vương Gia Huy**  **0950080032** | **Nguyễn Lương Thùy Dương**  **0950080044** | **Phạm Phú Đạt**  **0950080042** |
| **Tham gia (%)** | 45 | 20 | 35 |
| **Hoàn tất công việc (%)** | 100% | 100% | 100% |
| **Yêu cầu đồ án** | | | |
| Phân tích nghiệp vụ và thiết kế CSDL | | | 100% |
| Code giao diện chức năng  *(chỉ code các chức năng chính để demo lỗi tranh chấp)* | | |  |
| Xác định và giải quyết tranh chấp dữ liệu | | | 100% |
| Phân quyền người dùng | | |  |
| Viết báo cáo | | | 100% |

# LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cám ơn chân thành nhất đến thầy ThS. Phạm Trọng Huynh. Trong quá trình học tập và tìm hiểu môn Hệ quản trị cơ sở dữ liệu, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, định hướng tốt hơn. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này.

Do chưa có nhiều kinh nghiệm làm để tài cũng như những hạn chế về kiến thức, trong bài báo cáo chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự nhận xét, ý kiến đóng góp, phê bình từ phía thầy để bài báo cáo được hoàn thiện hơn.

Lời cuối cùng, chúng em xin kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc.

## 

# LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển nhanh chóng hiện nay, các hệ thống quản lý thông tin đang ngày càng trở nên quan trọng và phổ biến hơn bao giờ hết. Với sự tiến bộ trong lĩnh vực y tế cùng với tốc độ gia tăng dân số nhanh nên nhu cầu về quản lý dữ liệu trong lĩnh vực y tế cũng tăng theo. Cụ thể là quản lý trong lĩnh vực dược phẩm, một trong những lĩnh vực cần phải áp dụng công nghệ thông tin để quản lý một cách hiệu quả và đáp ứng nhu cầu của người dân. Đây là 1 chủ đề có tính ứng dụng cao, giúp ta hiểu rõ hơn về cách thức quản lý, những vấn đề cần lưu ý, những khó khăn và thách thức trong quản lý dược phẩm. Đó cũng chính là lí do mà nhóm chúng em thực hiện đề tài này.

Mục tiêu của chúng em là tập trung vào phần quản lý dữ liệu trên SQL Server, thực hiện các câu lệnh để quản lý dữ liệu trong SQL như tạo bảng, truy vấn,… để có thể đưa ra dữ liệu 1 cách chính xác và nhanh chóng.

# CHƯƠNG 1. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## LỜI GIỚI THIỆU

Trong đồ án này, nhóm chúng em sẽ tìm hiểu về việc xây dựng một hệ thống quản lý dược phẩm hiệu quả, về các chức năng cơ bản của hệ thống, các bảng dữ liệu cần thiết để lưu trữ thông tin và các liên kết giữa chúng. Ngoài ra, ta cũng sẽ tìm hiểu về các vấn đề liên quan đến quản lý dược phẩm như quản lý kho hàng, quản lý nhân viên, quản lý khách hàng và báo cáo doanh thu. Mục tiêu của đồ án này là giúp người đọc có cái nhìn tổng quan về cách thiết kế và triển khai một hệ thống quản lý dược phẩm hiệu quả để đáp ứng nhu cầu của người dùng.

## 1.2 PHÂN TÍCH HỆ THỐNG QUẢN LÝ DƯỢC PHẨM

### 1.2.1 Thực trạng của hệ thống quản lý dược phẩm

+ Yêu cầu xây dựng hệ thống quản lý dược phẩm

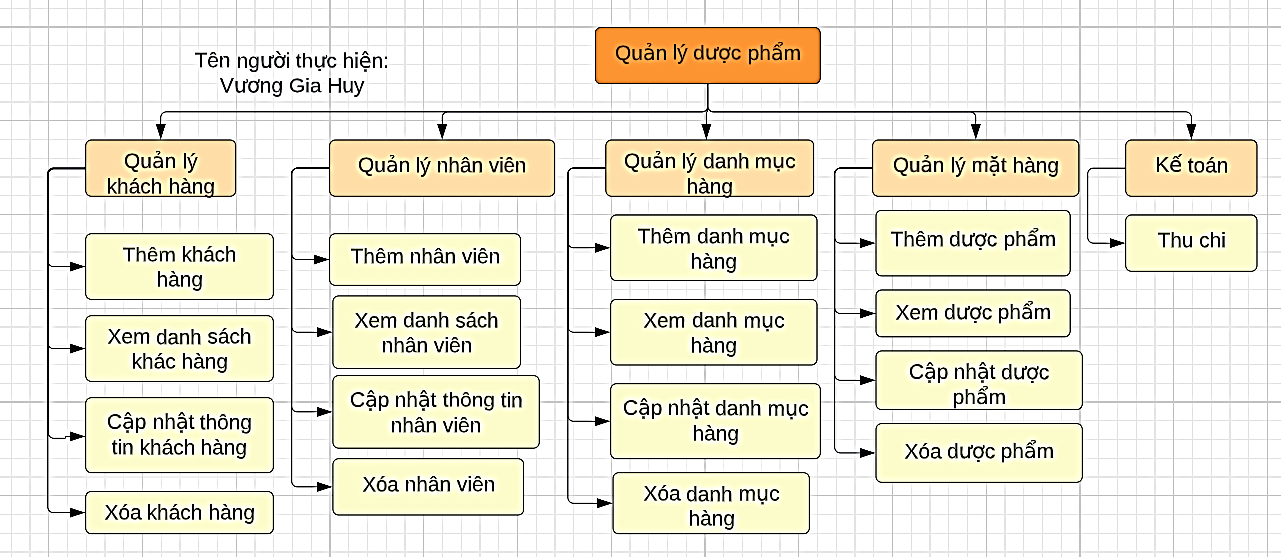
* Người quản lý nắm được tình hình mua bán, kho hàng của nhà thuốc, việc thống kê được thực hiện hàng tháng, hàng quý hoặc cũng có thể đột xuất theo yêu cầu.
* Dược sĩ bán thuốc sẽ nắm được các thông tin về dược phẩm như là giá cả theo từng loại dược phẩm, số lượng tồn kho, …
* Khách hàng sẽ biết được thông tin hóa đơn mà mình mua. Trong hóa đơn bao gồm ngày mua, thông tin về các loại thuốc, vv.

+ Việc quản lý dược phẩm sẽ được thực hiện như sau:

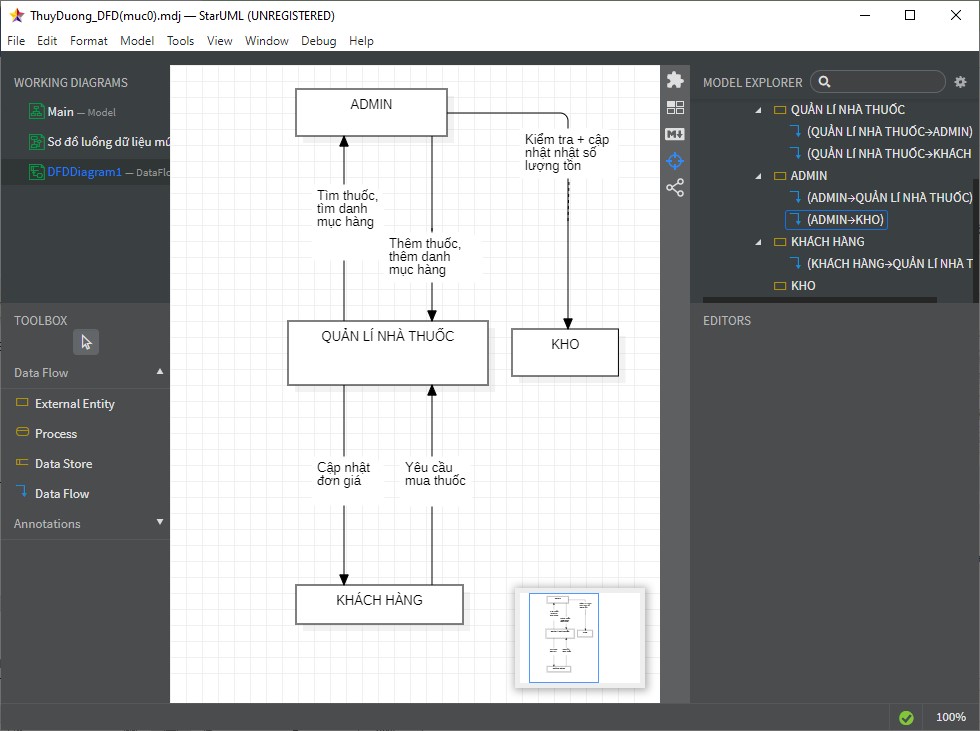
* Người quản lý trực tiếp liên hệ với nhà cung cấp dược để nhập hàng vào nhà thuốc, cũng như các quan hệ đối tác để xuất hàng ra bên ngoài. Người quản lý sẽ quản lý thông tin khách hàng, thống kê doanh thu của nhà thuốc.
* Khi hàng được nhập vào thì thủ kho sẽ tạo phiếu nhập hàng ghi đầy đủ thông tin về hàng đó. Khi hàng được bán trực tiếp cho đối tác hoặc được đưa lên quầy thì thủ kho cũng tạo phiếu xuất hàng. Thủ kho cũng có trách nhiệm kiểm kê hàng, khi có hàng hết hạn sử dụng thì tạo phiếu trả hàng cho nhà sản xuất.
* Dược sĩ bán thuốc làm việc theo ca, khi đến ca làm của mình, dược sĩ sẽ đăng nhập vào hệ thống. Nhập vào những thông tin sản phẩm bán ra (vd như bán cho khách thuốc sốt thì nhập vào đó) cùng với giá tiền, thời gian mua sau đó xuất hóa đơn ra. Đồng thời cũng giữ 1 bản sao cho phía nhà thuốc.

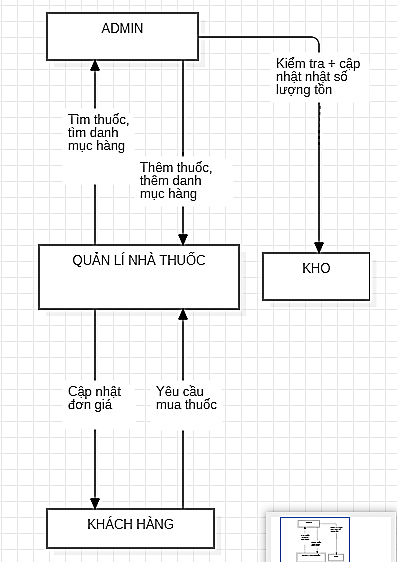
### 1.2.2 Biểu đồ phân cấp chức năng (BPC)

BPC là biểu đồ phân rã dần các chức năng lớn thành các chức năng chi tiết hơn. Cuối cùng ta thu được 1 cây chức năng giúp người đọc xác định rõ ràng, có cái nhìn khái quát, trực quan về hệ thống.

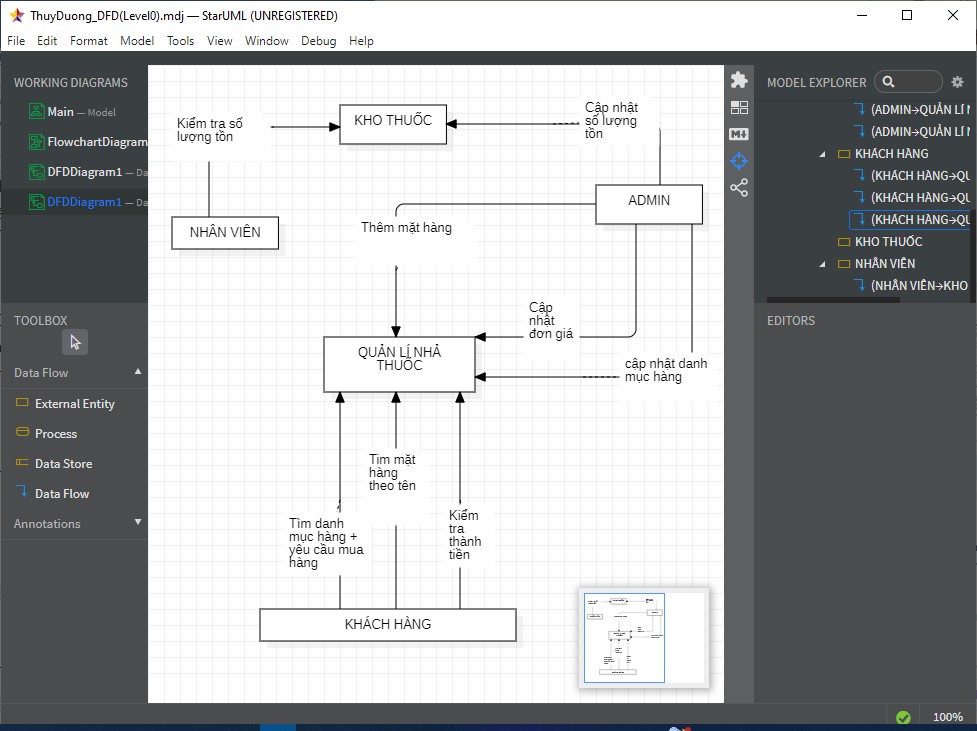


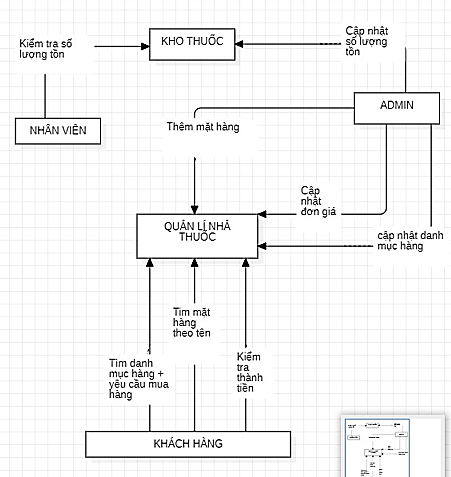
### 1.2.3 Biểu đồ luồng dữ liệu:





Hình 1. 1 Biểu đồ luồng dữ liệu cấp 0





Hình 1. 2 Biểu đồ luồng dữ liệu cấp 1

* KHÁCH HÀNG:
* Yêu cầu mua thuốc
* Tìm kiếm mặt hàng
* Tìm kiếm danh mục hàng
* Yêu cầu hóa đơn
* ADMIN
* Cập nhật danh mục hàng
* Kiểm tra số lượng tồn
* Cập nhật số lượng tồn
* Thêm mặt hàng
* Kiểm tra thành tiền
* HỆ THỐNG
* Cập nhật đơn giá cho khách hàng
* Gửi yêu cầu tìm kiếm mặt hàng lên admin
* Gửi yêu cầu tìm kiếm danh mục hàng lên admin
* Gửi yêu cầu hóa đơn lên admin

### 1.2.4 Phân loại các thuộc tính vào một tập thực thể

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên bảng** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** |
| KhachHang | MaKH  TenKH  DiaChi  SDT | Mã khách hàng  Tên khách hàng  Địa chỉ  Số điện thoại | NCHAR(10)  NVARCHAR(30)  NVARCHAR(50)  VARCHAR(15) |
| NhanVien | MaNV  TenNV  DiaChi  SDT  ChucVu | Mã nhân viên  Tên nhân viên  Địa chỉ  Số điện thoại  Chức vụ | NCHAR  NVARCHAR(30)  NVARCHAR(50)  VARCHAR(15)  NVARCHAR(30) |
| MatHang | MaMH  TenMH  DonViTinh  DonGia  SoLuongTon  MaDMH | Mã mặt hàng  Tên mặt hàng  Đơn vị tính  Đơn giá  Số lượng tồn  Mã danh mục hàng | NCHAR(10)  VARCHAR(30)  VARCHAR(10)  MONEY  INT  NCHAR(10) |
| KeToan | MaGG  MaNV  MaKH  MaMH  NgayGG  Tongtien | Mã nhân viên  Mã khách hàng  Mã mặt hàng  Tổng tiền | NCHAR(10)  NCHAR(10)  NCHAR(10)  NCHAR(10)  DATE  MONEY |
| DanhMucHang | MaDMH  TenDMH | Mã danh mục hàng  Tên danh mục hàng | NCHAR(10)  VARCHAR(50) |

### 1.2.5 Tình huống tranh chấp

Trong dự án này, nhóm chúng em sẽ thực hiện 3 loại tình huống tranh chấp.

* Unrepeatable read: Xảy ra khi một giao transaction đọc một hàng từ cơ sở dữ liệu và lưu giữ giá trị của nó. Trong khi giao transaction đang thực thi, một giao transacton khác cập nhật giá trị của hàng đó và commit. Sau đó, giao transacton đầu tiên đọc lại hàng đó và thấy giá trị đã thay đổi. Điều này có thể gây ra sự không nhất quán trong dữ liệu và ảnh hưởng đến tính đúng đắn của các kết quả truy vấn và tính toàn vẹn của cơ sở dữ liệu.
* Dirty read: Xảy ra khi một transaction đọc dữ liệu từ một record đang được transaction khác cập nhật, trước khi transaction đó commit hoặc rollback. Có nghĩa là transaction đọc dữ liệu "bẩn" hoặc "chưa đúng", vì dữ liệu đó có thể đã bị thay đổi bởi transaction khác và chưa được xác nhận. Ví dụ, nếu Transaction A cập nhật dữ liệu trong một bảng, nhưng chưa kết thúc transaction, Transaction B đọc dữ liệu từ bảng đó, thì Transaction B đang đọc "dirty data" (dữ liệu chưa xác nhận) do nó đang đọc dữ liệu từ Transaction A, dù dữ liệu đó có thể sẽ bị rollback nếu Transaction A thất bại.
* Phantom read: Là hiện tượng khi một transaction thực hiện hai lần truy vấn cùng một câu lệnh SELECT trên cùng một bảng, nhưng kết quả trả về khác nhau. Hiện tượng này thường xảy ra khi trong quá trình thực hiện transaction, một transaction khác đã thay đổi số lượng hoặc nội dung của các bản ghi trong bảng dữ liệu, dẫn đến kết quả truy vấn của transaction đầu tiên và transaction thứ hai không giống nhau.

Ví dụ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Chức năng 1 | Người dùng | Chức năng 2 | Người dùng | Lỗi tranh chấp | Mô tả |
| 1 | Tìm danh mục hàng bằng tên danh mục hàng | Khách hàng | Cập nhật tên danh mục hàng | Admin | Unrepeatable read | -A: Khách hàng tìm kiếm danh mục hàng với tên “Thuốc ho”  -B: Admin cập nhật tên danh mục hàng từ “Thuốc ho” sang “Thuốc giảm đau”  -Lần đọc 1-A: Tìm thấy danh mục hàng tương ứng với tên  -Lần ghi 1-B:  Cập nhật tên danh mục hàng sang “Thuốc giảm đau”  -Lần đọc 2-A:  Không tìm thấy sản phẩm tương ứng với tên “Thuốc ho”. |
| 2 | Kiểm tra xem số lượng tồn của mặt hàng | Nhân viên | Cập nhật số lượng tồn | Admin | Dirty read | -A: Nhân viên xem lại số lượng tồn của mặt hàng “Ibuprofen”  -B: Admin thực hiện cập nhật dữ liệu về số lượng tồn còn lại  -Lần đọc 1-A: Nhân viên thấy số lượng tồn của mặt hàng lúc admin chưa cập nhật là 800 viên  -Lần ghi 1-B: Admin cập nhật số số lượng tồn  -Lần đọc 2-A: Nhân viên kiểm tra lại số lượng tồn của mặt hàng ” Ibuprofen” đã được tăng lên 1200 viên |
| 3 | Tìm kiếm mặt hàng theo tên | Khách hàng | Thêm mặt hàng cùng tên | Admin | Phantom read | * A: Khách hàng tìm mặt hàng theo tên X. * B: Admin thêm mặt hàng X. * - Lần đọc 1-A: Hiển thị có 10 kết quả tìm kiếm. * - Lần đọc 2-A: Hiển thị danh sách 11 mặt hàng tương ứng. |
| 4 | Kiểm tra thành tiền | Khách hàng | Cập nhật đơn giá cho mặt hàng | Admin | Unrepeatable read | * A: Khách hàng kiểm tra thành tiền của mặt hàng X ứng với số lượng đã mua. (Thành tiền = Đơn giá \* Số lượng) * B: Admin cập nhật giá của mặt hàng X * - Lần đọc 1-A: Hiển thị đơn giá mặt hàng X(trước khi admin cập nhật giá) * - Lần đọc 2-A: Hiển thị thành tiền (ứng với giá đã cập nhật) |

Trong ví dụ đầu tiên, khi khách hàng tìm kiếm 1 loại thuốc ho thì người quản trị cập nhật thuốc ho thành thuốc giảm đau. Khi đó sẽ có 2 trường hợp là trước khi người quản trị cập nhật thì khách hàng vẫn tìm được thuốc ho, trường hợp còn lại khi người quản trị đã cập nhật thì lúc này dữ liệu đã thay đổi và không tìm được thuốc ho. Đây là tính không nhất quán của dữ liệu và ảnh hưởng đến tính chính xác của “unrepeatable read”.

Trong ví dụ số 2, chúng ta có thể thấy khi nhân viên kiểm tra số lượng còn lại của thuốc ‘Ibuprofen’. Lúc đó, người quản trị cập nhật lại số lượng của thuốc nhưng chưa được xác nhận. Thì lúc này người nhân viên đang đọc dữ liệu cũ hay còn gọi là “dirty read”.

Trong ví dụ thứ 3, khi khách hàng tìm danh sách mặt hàng theo tên X, người quản trị thêm sản phẩm vào danh sách mặt hàng X. Vậy trước khi người quản trị thêm thì khách hàng sẽ tìm được 10 kết quả, sau đó khách hàng sẽ tìm được 11 kết quả. Lần tìm kiếm thứ 2 sẽ có nhiều bản ghi hơn lần đầu tiên, đây là hiện tượng “phantom read”.

Ở ví dụ cuối cùng, khách hàng kiểm tra thành tiền của mặt hàng X tương ứng với số lượng đã mua, người quản trị lúc này cập nhật giá của mặt hàng X, vậy sẽ có 2 tình huống xảy ra đó là hệ thống sẽ hiển thị giá cũ và hệ thống sẽ hiển thị giá mới được cập nhật. Đây cũng tương tự như ví dụ đầu tiên về tính nhất quán và chính xác của dữ liệu của “unrepeatable read”.

# CHƯƠNG 2: ỨNG DỤNG LÝ THUYẾT (tùy vào đề tài mà tên của chương thay đổi phù hợp, hiện thực lý thuyết lên 1 HQT CSDL cụ thể)

## 2.1. Xây dựng CSDL trên HQT CSDL

- Câu lệnh tạo cơ sở dữ liệu, tạo bảng, tạo khóa.

CREATE DATABASE QuanLyDuocPham;

USE QuanLyDuocPham;

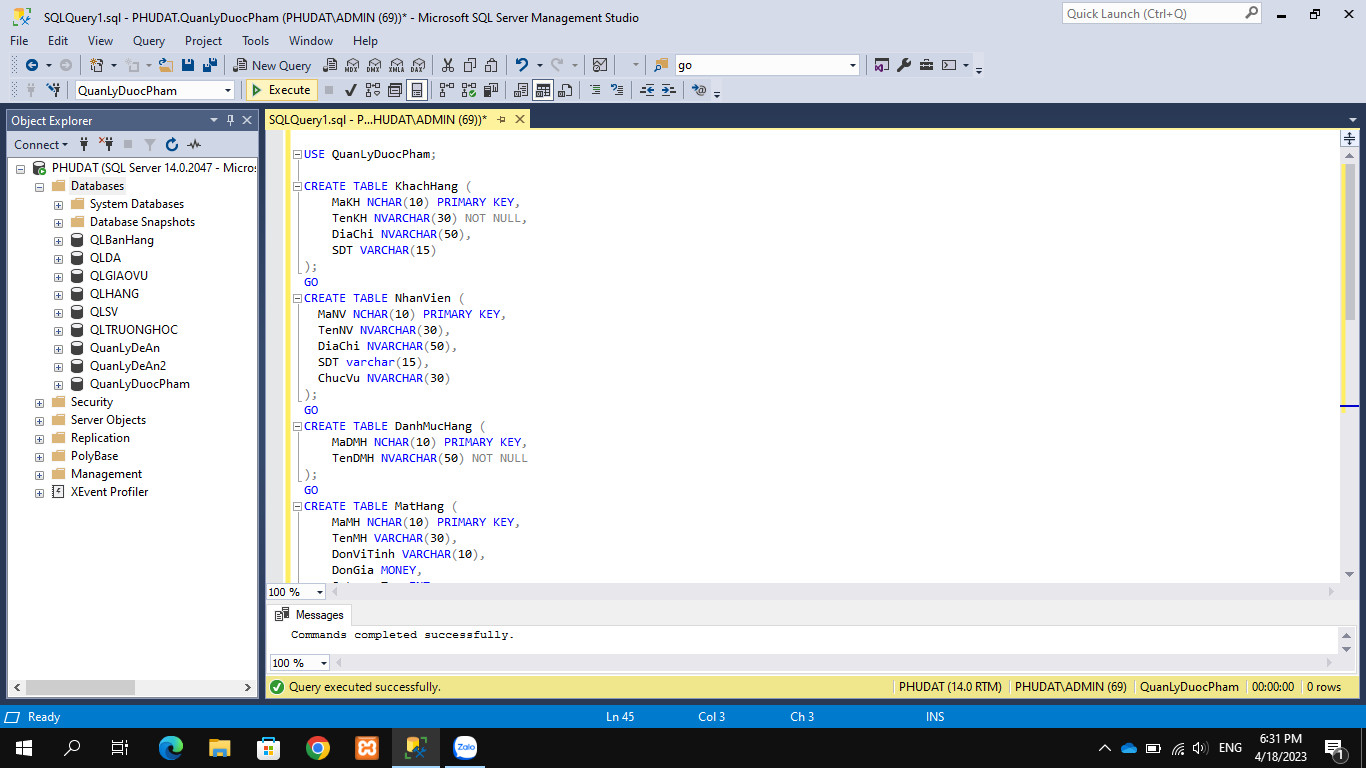
CREATE TABLE KhachHang ( MaKH NCHAR(10) PRIMARY KEY, TenKH NVARCHAR(30) NOT NULL, DiaChi NVARCHAR(50), SDT VARCHAR(15) );

CREATE TABLE NhanVien ( MaNV NCHAR(10) PRIMARY KEY, TenNV NVARCHAR(30), DiaChi NVARCHAR(50), SDT varchar(15), ChucVu NVARCHAR(30) );

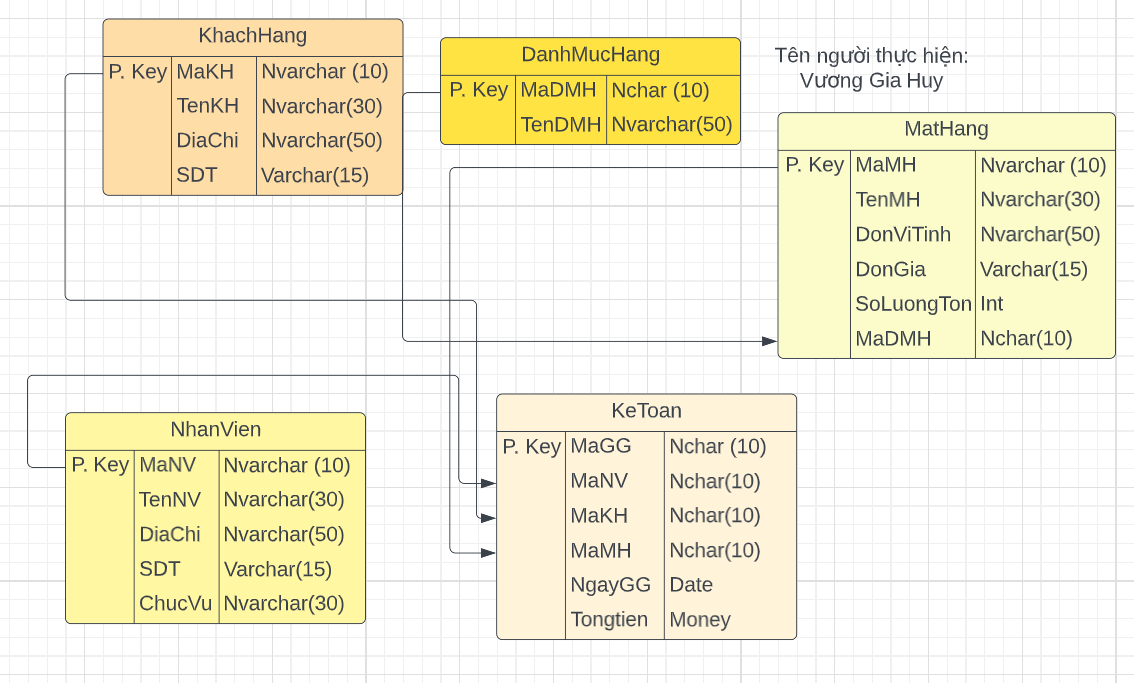
CREATE TABLE DanhMucHang ( MaDMH NCHAR(10) PRIMARY KEY, TenDMH VARCHAR(50) NOT NULL );

CREATE TABLE MatHang ( MaMH NCHAR(10) PRIMARY KEY, TenMH VARCHAR(30), DonViTinh VARCHAR(10), DonGia MONEY, SoLuongTon INT, MaDMH NCHAR(10), FOREIGN KEY (MaDMH) REFERENCES DanhMucHang(MaDMH) );

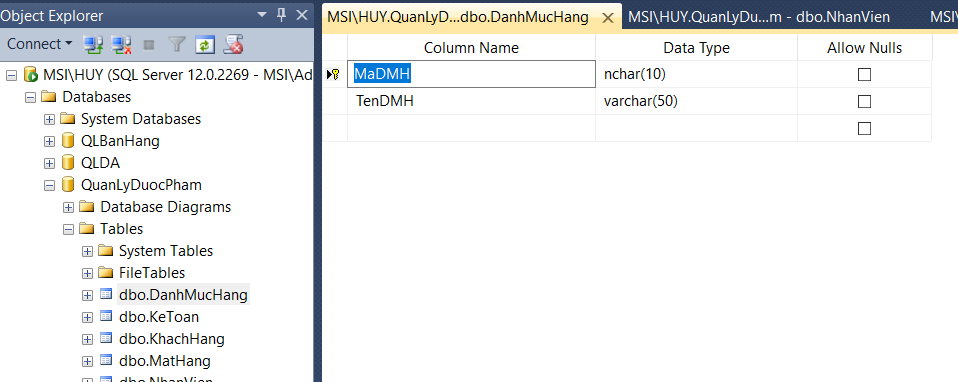
CREATE TABLE KeToan ( MaGG NCHAR(10) PRIMARY KEY, MaNV NCHAR(10), MaKH NCHAR(10), MaMH NCHAR(10), NgayGG DATE, Tongtien MONEY, FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV), FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH), FOREIGN KEY (MaMH) REFERENCES MatHang(MaMH);



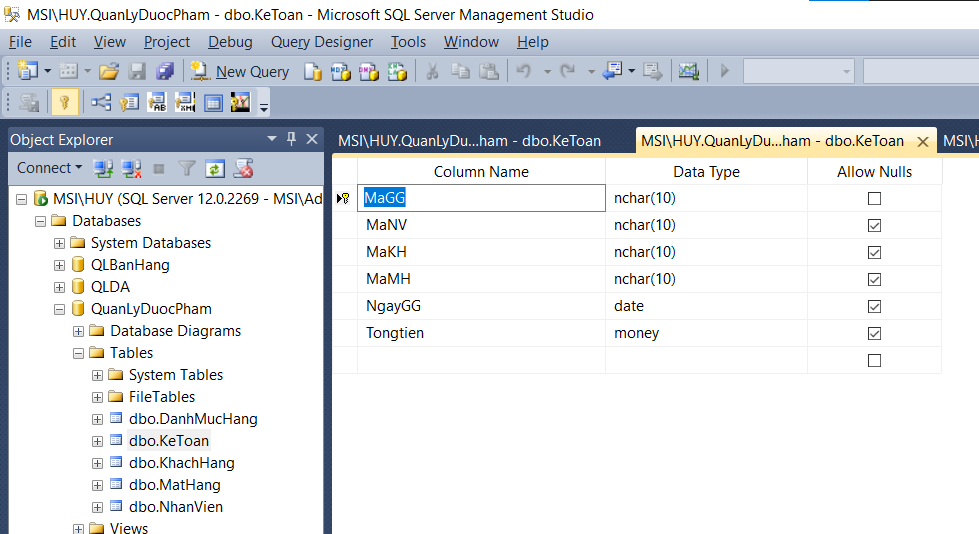
- Cấu trúc của mỗi bảng:



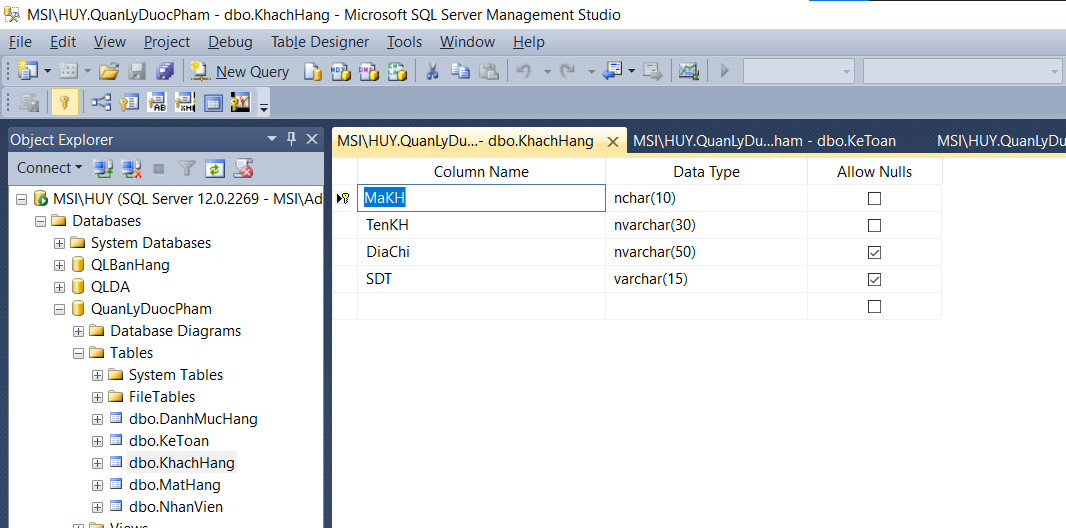
Bảng Danh mục hàng



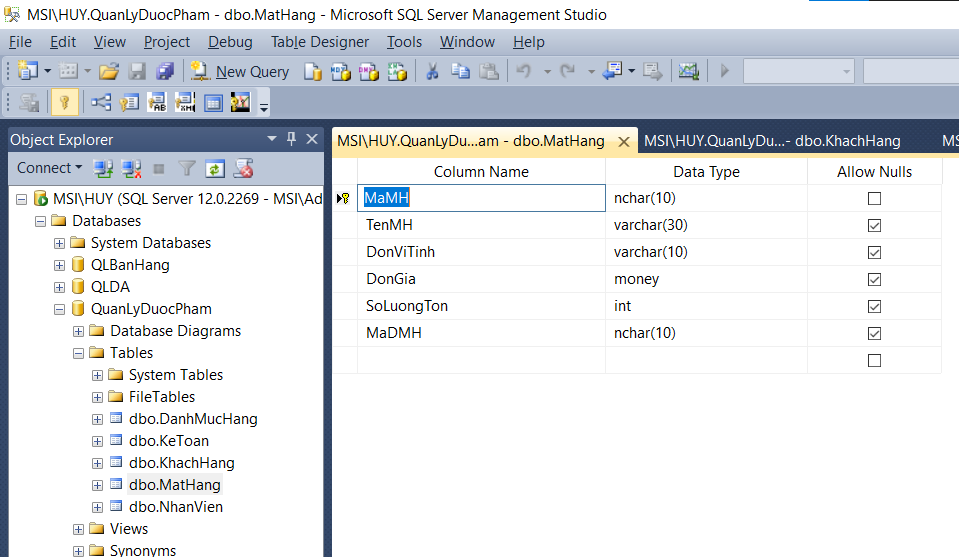
Bảng Kế toán



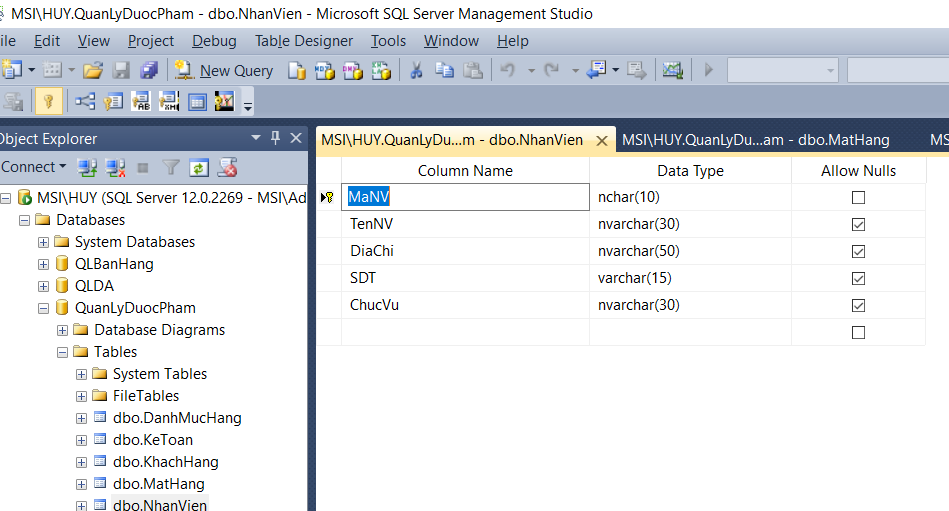
Bảng Khách hàng



Bảng Mặt hàng

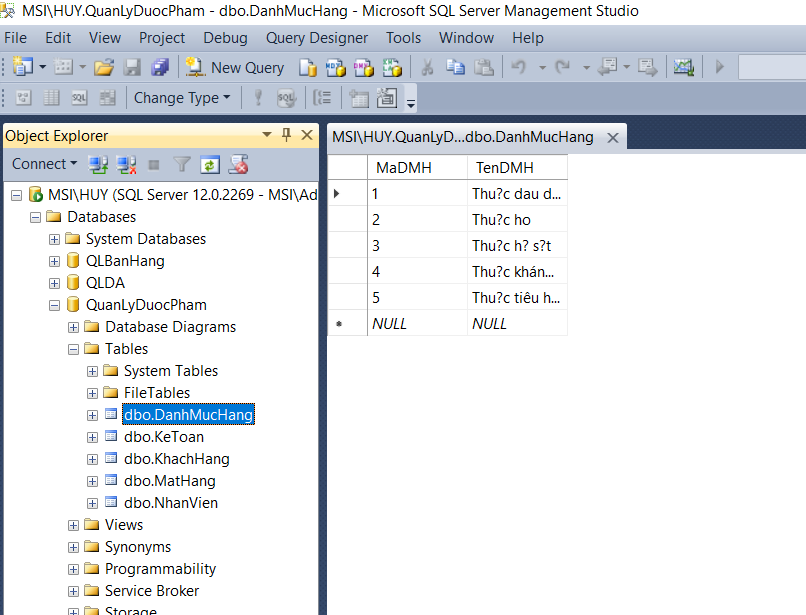


Bảng Nhân viên

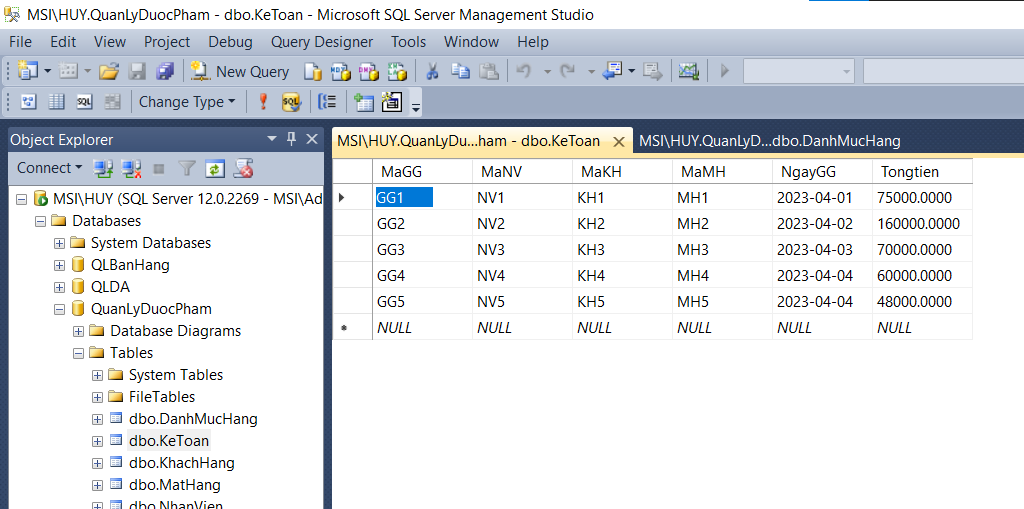


Dữ liệu của mỗi bảng:

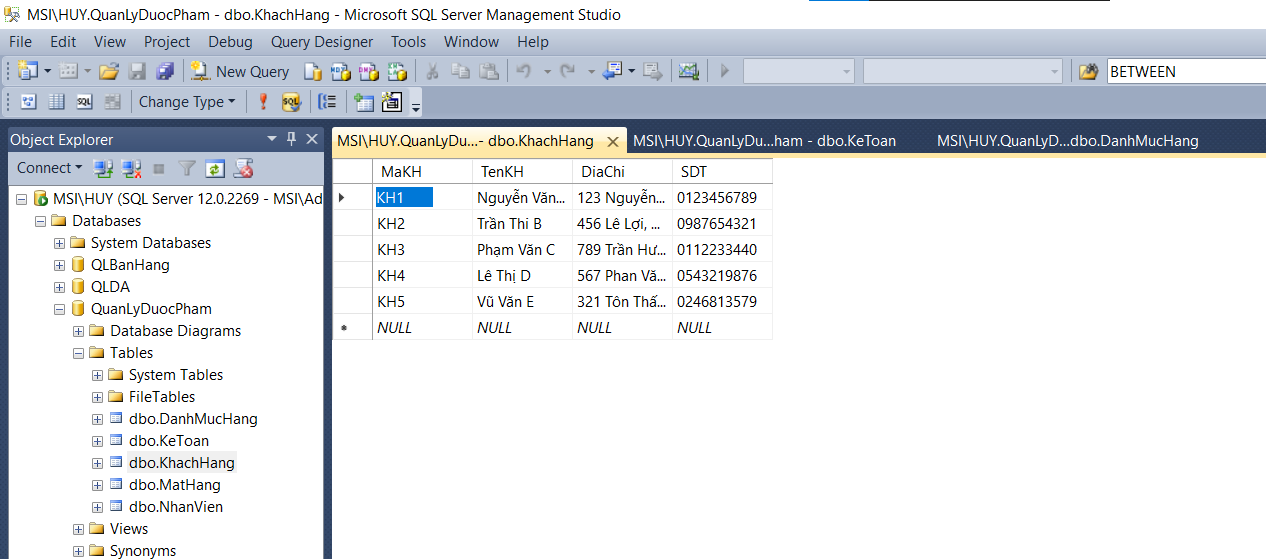
Bảng Danh mục hàng



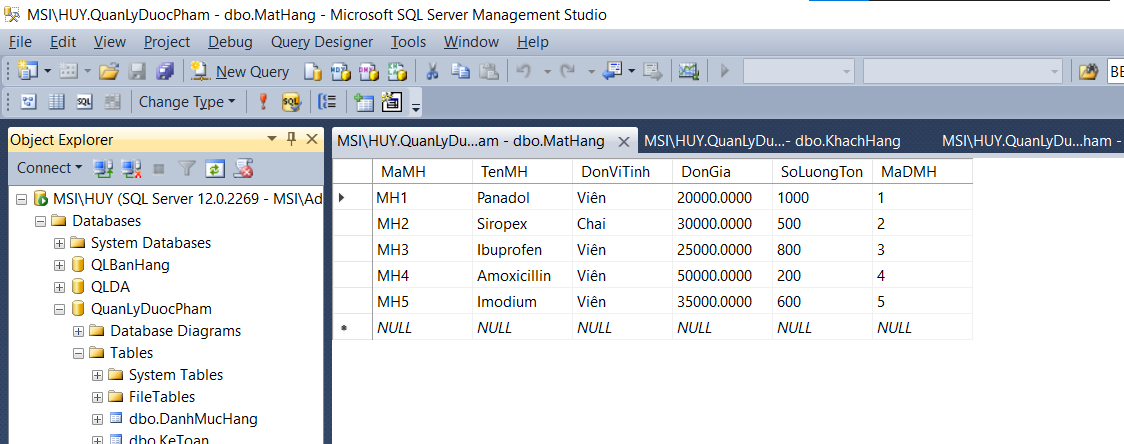
Bảng Kế toán



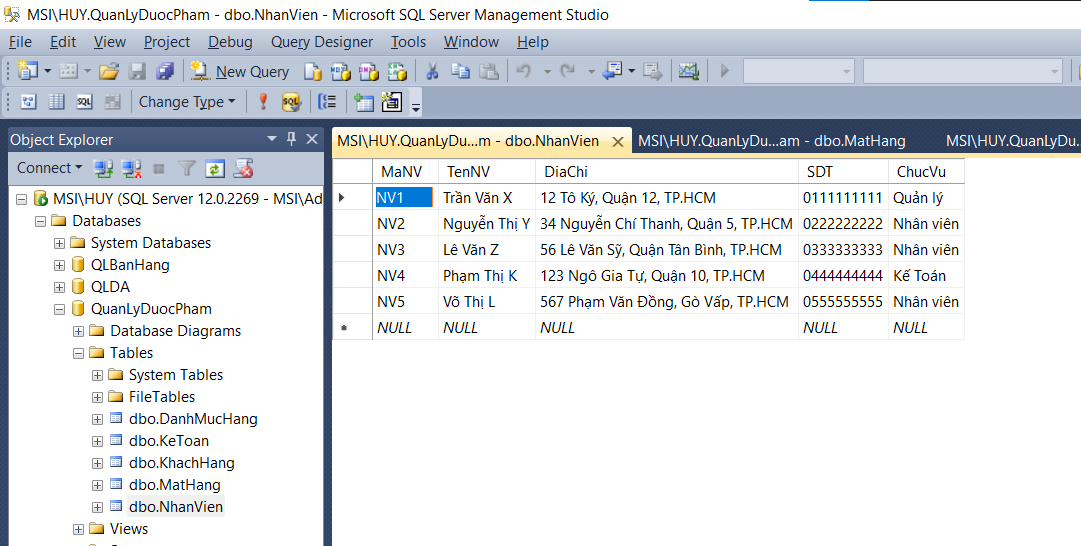
Bảng Khách hàng



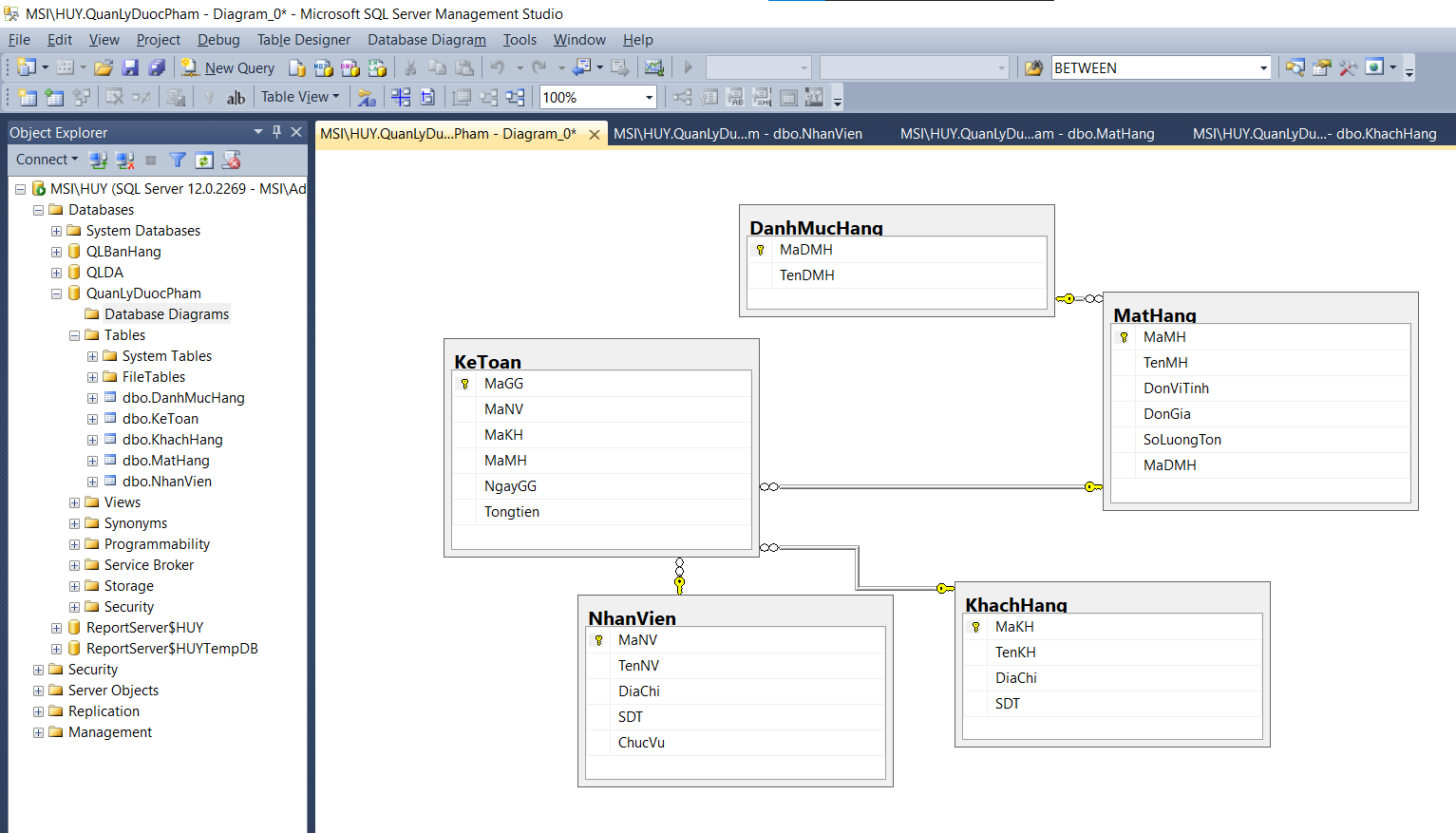
Bảng Mặt hàng



Bảng Nhân viên



- Màn hình Diagram liên kết các bảng.



## 2.2. Thao tác trên CSDL trên HQT CSDL

INSERT INTO KhachHang (MaKH, TenKH, DiaChi, SDT)

VALUES ('KH1', N'Nguyễn Văn A', N'123 Nguyễn Trãi, Quận 5, TP.HCM', '0123456789'),

('KH2', N'Trần Thi B', N'456 Lê Lợi, Quận 1, TP.HCM', '0987654321'),

('KH3', N'Phạm Văn C', N'789 Trần Hưng Đạo, Quận 10, TP.HCM', '0112233440'),

('KH4', N'Lê Thị D', N'567 Phan Văn Trị, Gò Vấp, TP.HCM', '0543219876'),

('KH5', N'Vũ Văn E', N'321 Tôn Thất Thuyết, Quận 4, TP.HCM', '0246813579');

INSERT INTO NhanVien (MaNV, TenNV, DiaChi, SDT, ChucVu)

VALUES ('NV1', N'Trần Văn X', N'12 Tô Ký, Quận 12, TP.HCM', '0111111111', N'Quản lý'),

('NV2', N'Nguyễn Thị Y', N'34 Nguyễn Chí Thanh, Quận 5, TP.HCM', '0222222222', N'Nhân viên'),

('NV3', N'Lê Văn Z', N'56 Lê Văn Sỹ, Quận Tân Bình, TP.HCM', '0333333333', N'Nhân viên'),

('NV4', N'Phạm Thị K',N'123 Ngô Gia Tự, Quận 10, TP.HCM', '0444444444', N'Kế Toán'),

('NV5', N'Võ Thị L', N'567 Phạm Văn Đồng, Gò Vấp, TP.HCM', '0555555555', N'Nhân viên');

INSERT INTO DanhMucHang (MaDMH, TenDMH)

VALUES ('1', N'Thuốc đau đầu'),

('2', N'Thuốc ho'),

('3', N'Thuốc hạ sốt'),

('4', N'Thuốc kháng sinh'),

('5', N'Thuốc tiêu hóa');

INSERT INTO MatHang (MaMH, TenMH, DonViTinh, DonGia, SoLuongTon, MaDMH)

VALUES ('MH1', 'Panadol', N'Viên', 20000, 1000, '1'),

('MH2', 'Siropex', N'Chai', 30000, 500, '2'),

('MH3', 'Ibuprofen',N'Viên', 25000, 800, '3'),

('MH4', 'Amoxicillin', N'Viên', 50000, 200, '4'),

('MH5', 'Imodium', N'Viên', 35000, 600, '5');

INSERT INTO KeToan (MaGG, MaNV, MaKH, MaMH, NgayGG, Tongtien)

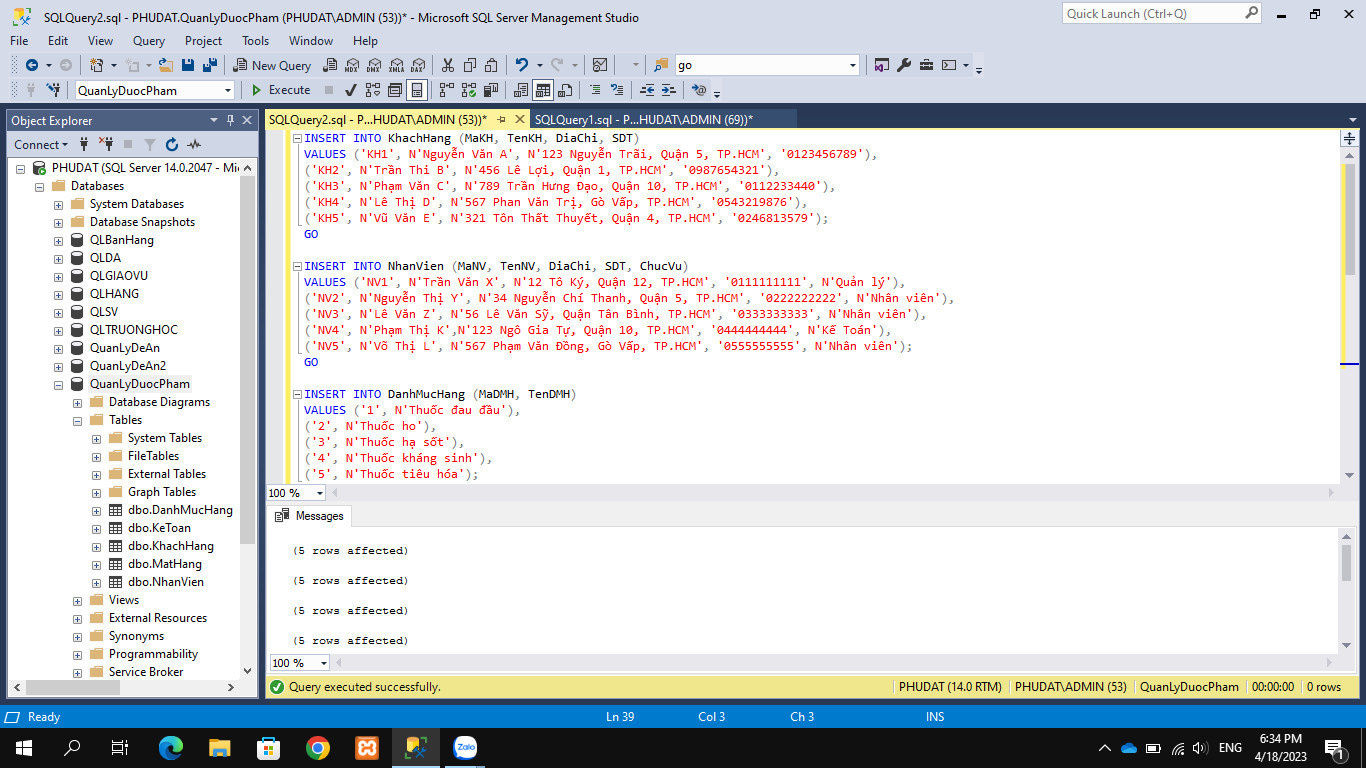
VALUES ('GG1', 'NV1', 'KH1', 'MH1', '2023-04-01', 75000),

('GG2', 'NV2', 'KH2', 'MH2', '2023-04-02', 160000),

('GG3', 'NV3', 'KH3', 'MH3', '2023-04-03', 70000),

('GG4', 'NV4', 'KH4', 'MH4', '2023-04-04', 60000),

('GG5', 'NV5', 'KH5', 'MH5', '2023-04-04', 48000);

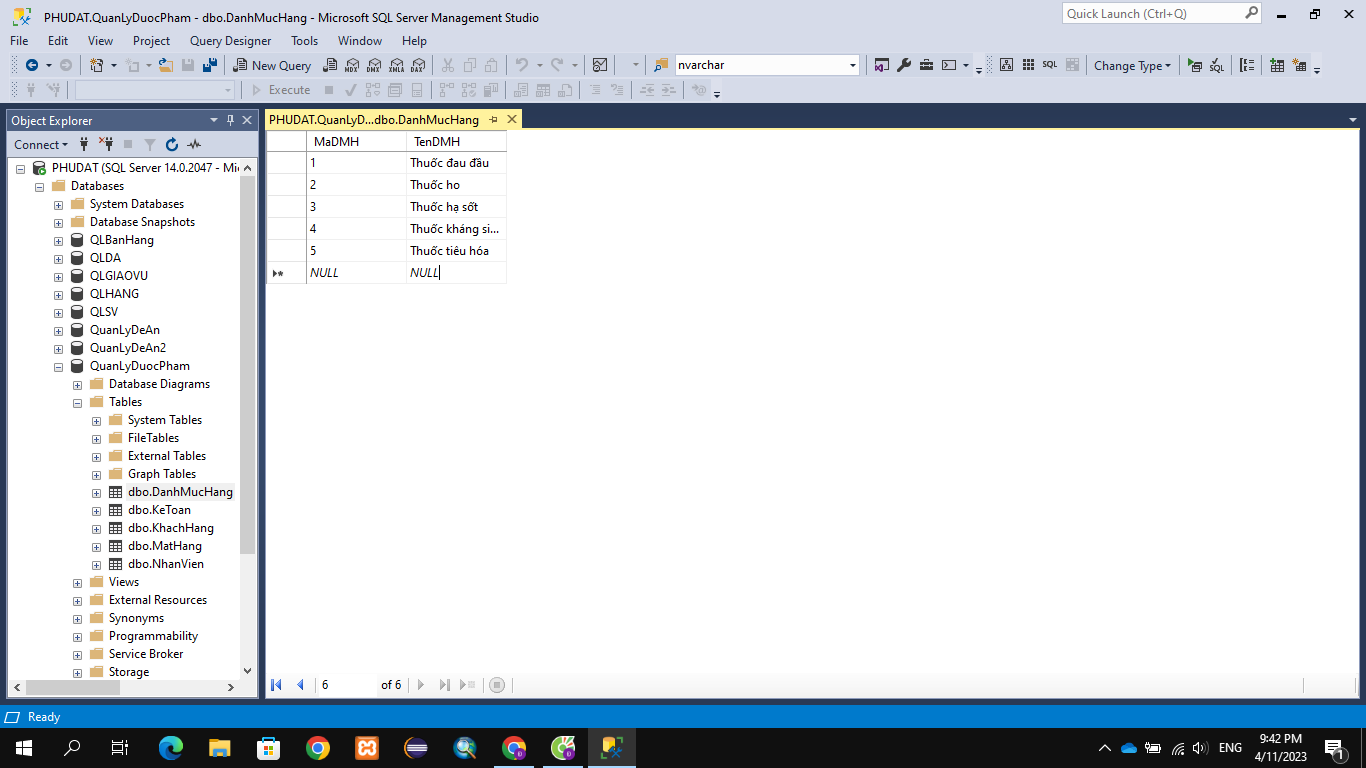


**CÂU LỆNH UPDATE**

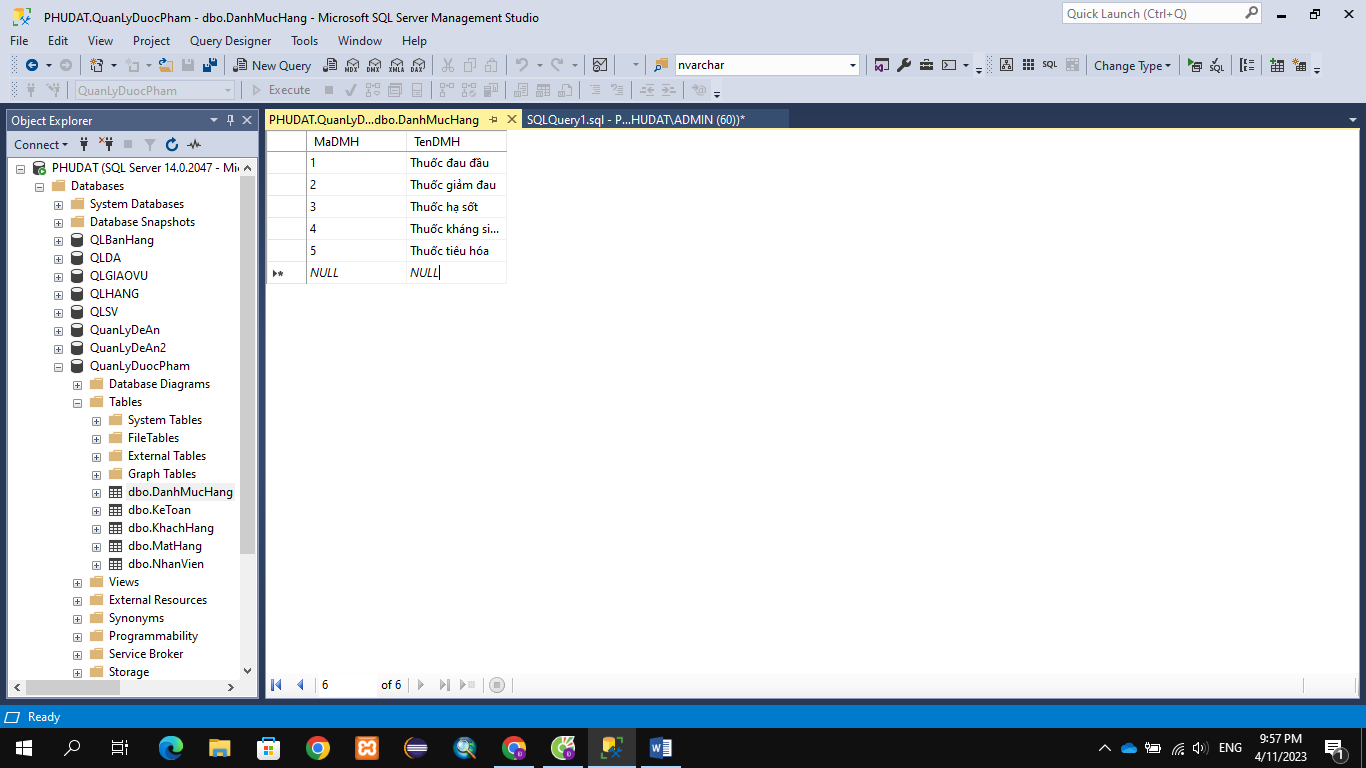
1. Cập nhật tên danh mục hàng có mã danh mục hàng là 2



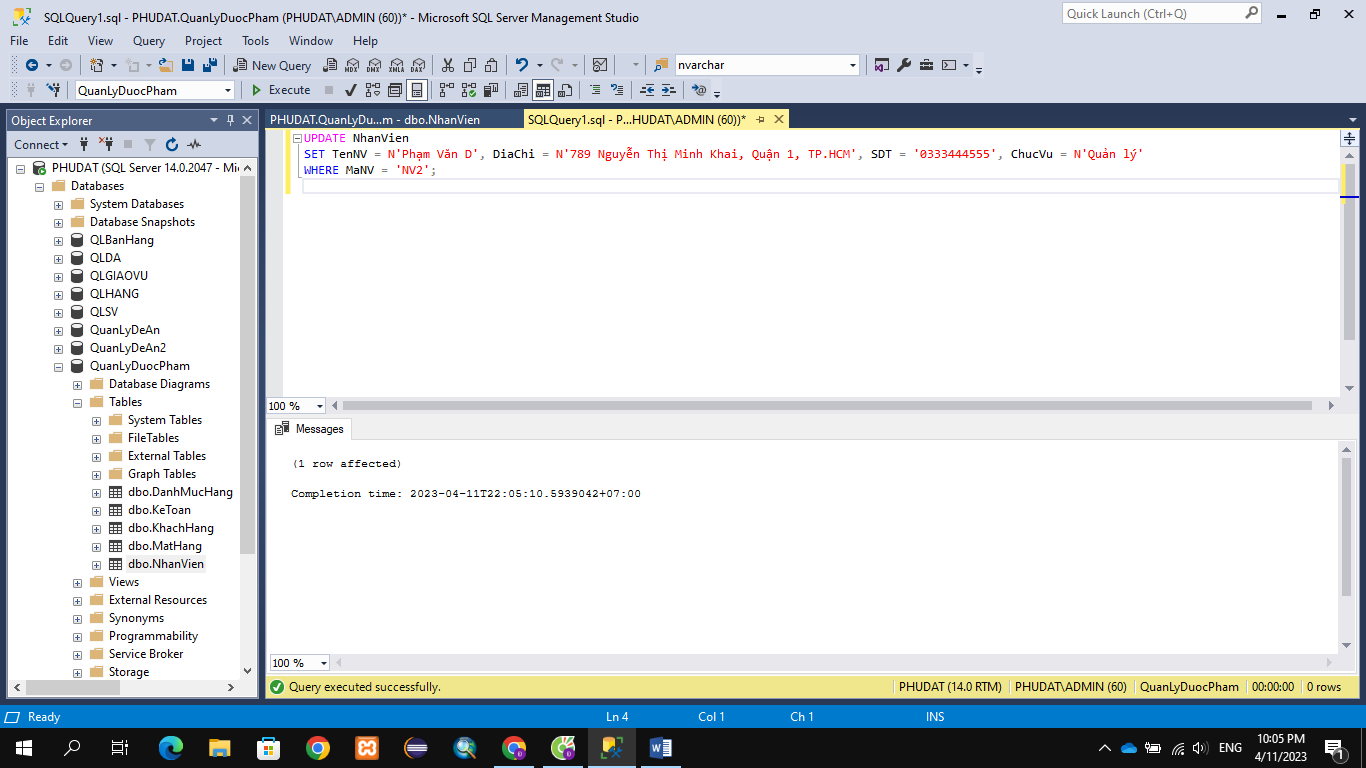
- Trước khi cập nhật



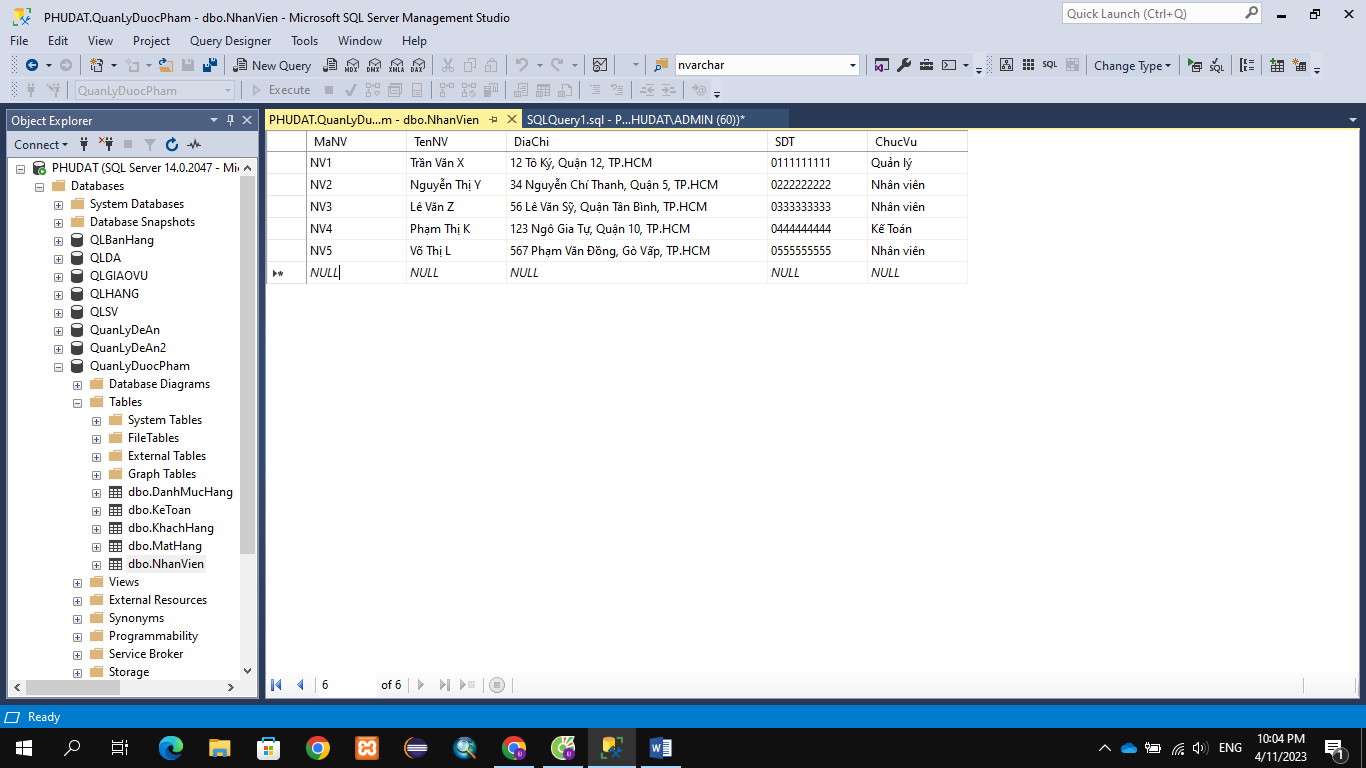
* Sau khi cập nhật



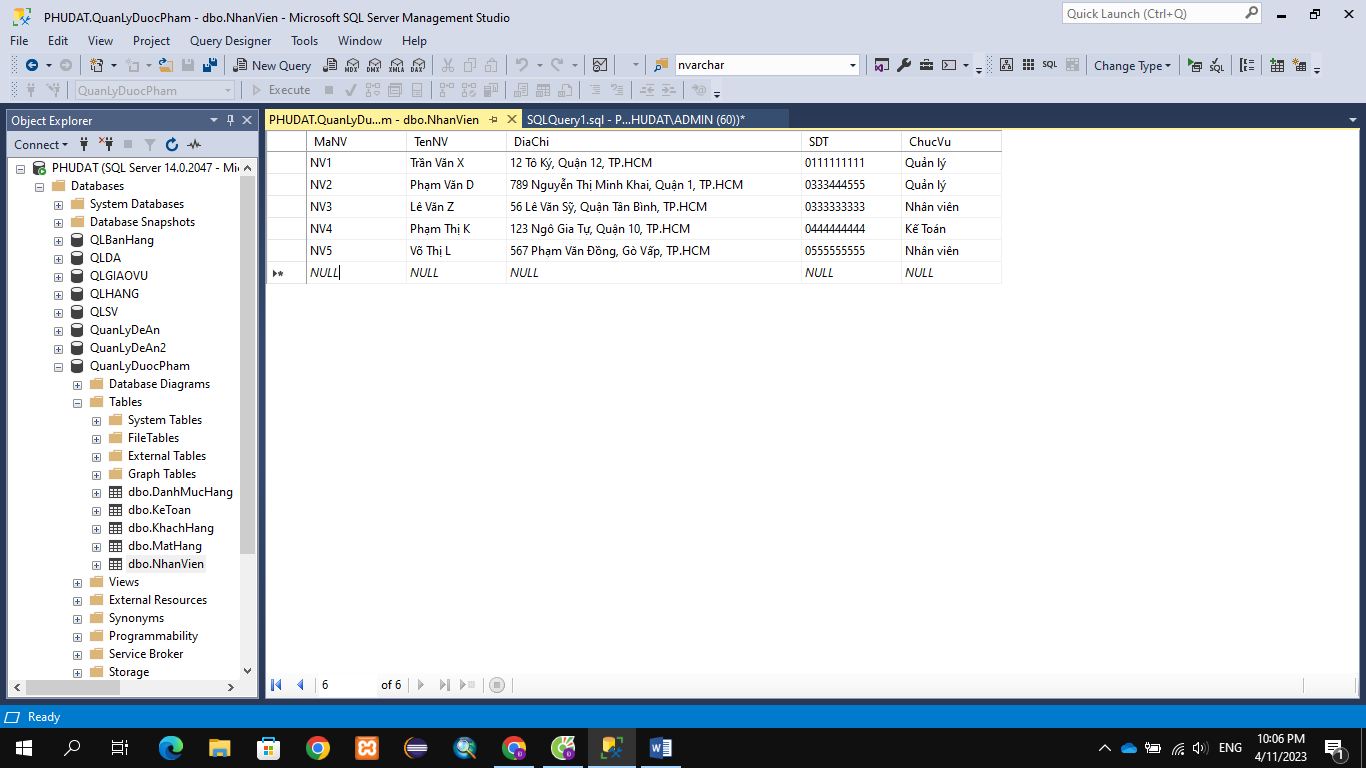
1. Cập nhật thông tin nhân viên có mã nhân viên là “NV2”



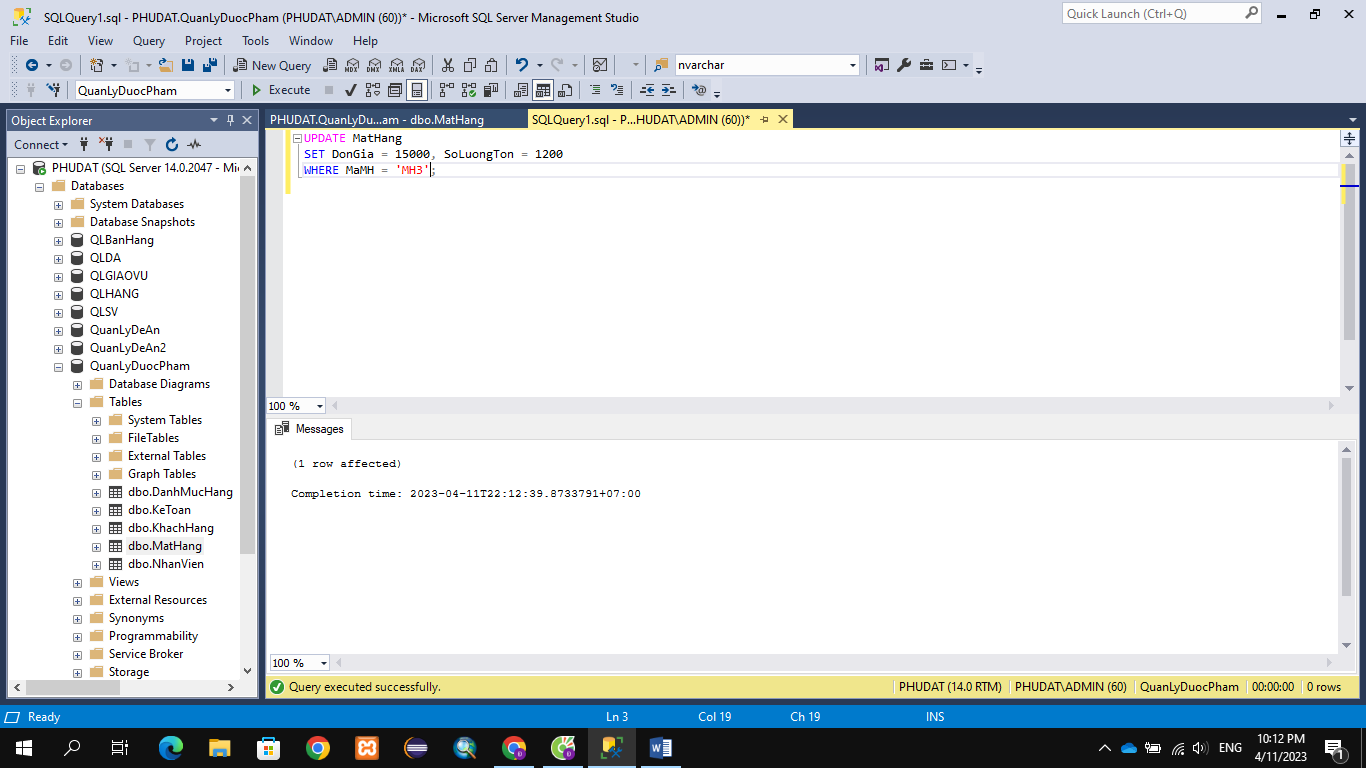
* Trước khi cập nhật



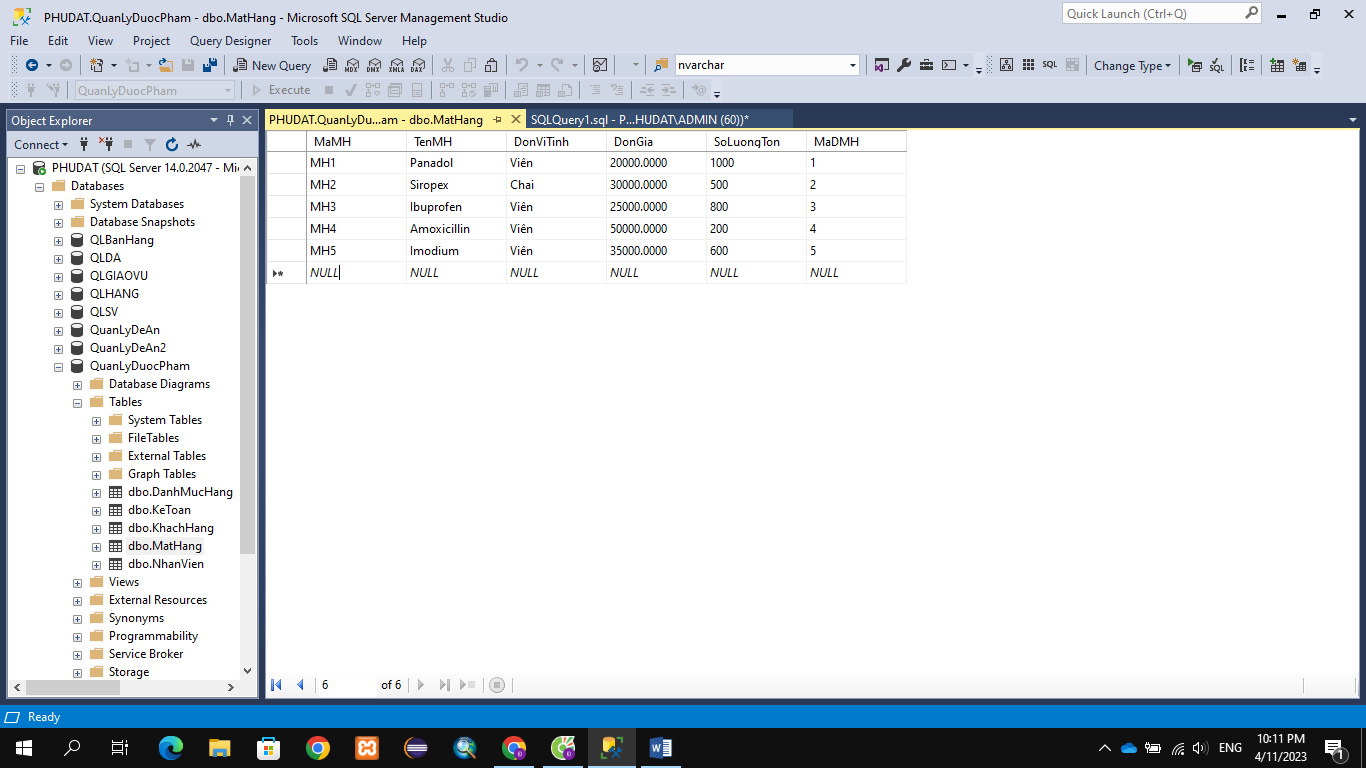
* Sau khi cập nhật

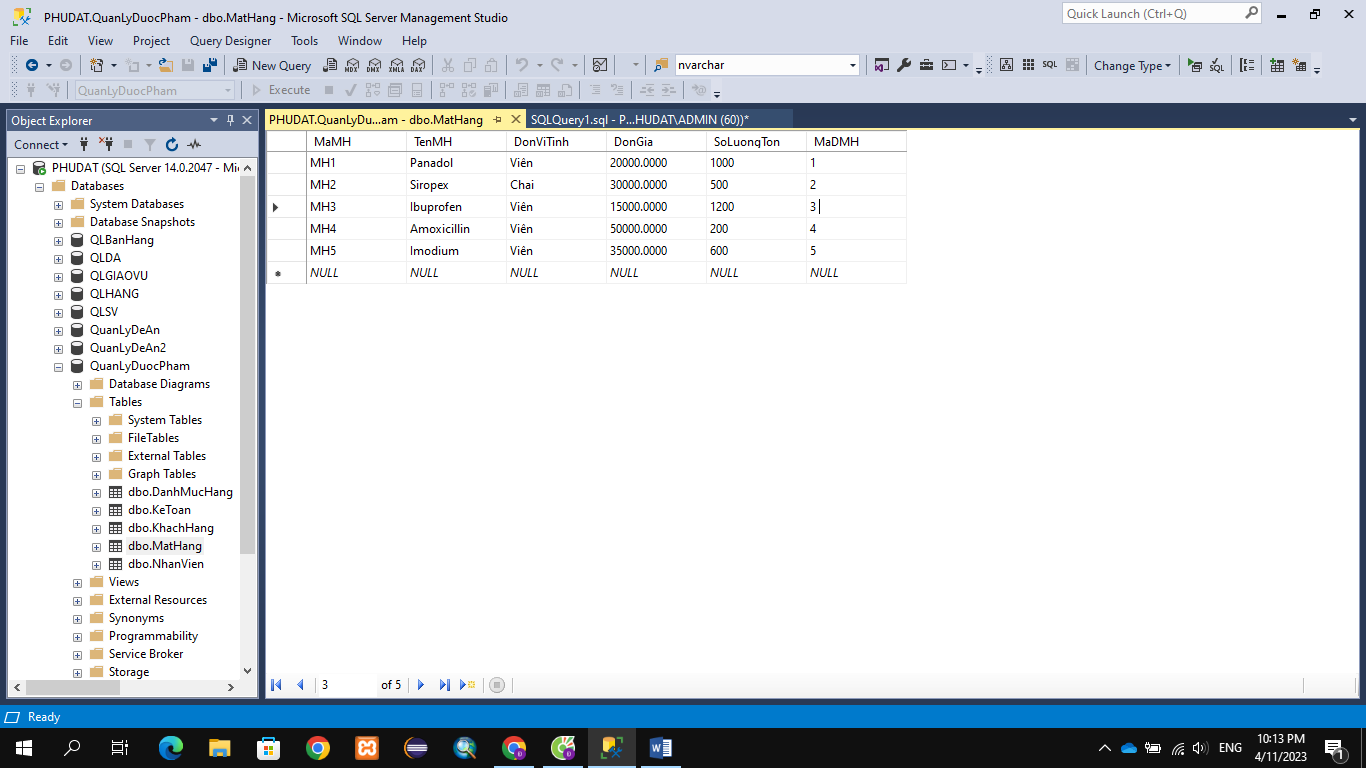


1. Cập nhật đơn giá và số lượng tồn của mặt hàng có mã mặt hàng là “MH3”



* Trước khi cập nhật



* Sau khi câp nhật
* 

## 2.3. Stored Procedure, Funtion, Trigger

### 2.3.1. Stored Procedure

* Định nghĩa:

Stored Procedure là 1 khối lệnh được đặt tên và lưu trữ trên database server, gồm 1 tập các câu lệnh SQL để thực hiện 1 công việc cụ thể. Giúp đơn giản hóa việc truy xuất dữ liệu, cải thiện hiệu suất của cơ sở dữ liệu bằng cách tối ưu câu lệnh truy vấn.

* Cú pháp:

Cú pháp của Stored Procedure bắt đầu bằng từ khóa CREATE PROCEDURE + tên ta đặt cho procedure và tham số (tùy chọn) + AS + 1 khối các câu lệnh nằm trong cặp BEGIN và END.

* Các loại Stored Procedure:

Có nhiều loại Stored Procedure, tùy vào mục đích sử dụng mà ta có thể phân loại như sau:

* Stored Procedure không trả về giá trị (Non-returning Stored Procedure): Chỉ thực hiện các lệnh truy vấn, cập nhật, thêm, xoá dữ liệu mà không trả về bất kỳ giá trị nào.
* Stored Procedure trả về giá trị (Returning Stored Procedure): Trả về giá trị, có thể là một bảng, một giá trị đơn, hoặc một tập hợp các giá trị.
* Stored Procedure có tham số đầu vào (Input Parameter Stored Procedure): Có chứa các tham số đầu vào, giúp truyền dữ liệu vào Stored Procedure để thực hiện các câu lệnh.
* Stored Procedure có tham số đầu ra (Output Parameter Stored Procedure): Chứa các tham số đầu ra, giúp trả về các giá trị sau khi Stored Procedure được thực thi.
* Stored Procedure hỗn hợp (Hybrid Stored Procedure): Stored Procedure này kết hợp cả hai loại trên, tức là có cả tham số đầu vào và đầu ra.
* Stored Procedure System (System Stored Procedure): Là Stored Procedure được hệ thống cài đặt sẵn, có chức năng thực hiện các tác vụ liên quan đến cơ sở dữ liệu, quản lý người dùng và phân quyền.
* Stored Procedure mở rộng (Extended Stored Procedure): Là Stored Procedure được tạo ra bởi các nhà sản xuất phần mềm khác, được kết nối với SQL Server và có thể được sử dụng bởi người dùng để thực hiện các tác vụ cụ thể.
* Ưu và khuyết điểm của từng loại:

**+ Stored Procedure không trả về giá trị (Non-returning Stored Procedure):**

Ưu điểm: Thực hiện các tác vụ cập nhật, thêm, xoá dữ liệu nhanh chóng và hiệu quả.

Khuyết điểm: Không thể truy xuất các giá trị trả về từ Stored Procedure.

**+ Stored Procedure trả về giá trị (Returning Stored Procedure):**

Ưu điểm: Có thể trả về các giá trị phức tạp và tập hợp giá trị.

Khuyết điểm: Có thể làm chậm hệ thống nếu kết quả trả về quá lớn.

**+ Stored Procedure có tham số đầu vào (Input Parameter Stored Procedure):**

Ưu điểm: Giúp truyền dữ liệu vào Stored Procedure để thực hiện các tác vụ.

Khuyết điểm: Nếu có quá nhiều tham số đầu vào, có thể làm cho việc gọi Stored Procedure trở nên phức tạp.

**+ Stored Procedure có tham số đầu ra (Output Parameter Stored Procedure):**

Ưu điểm: Có thể trả về các giá trị phức tạp và tập hợp giá trị.

Khuyết điểm: Có thể làm chậm hệ thống nếu kết quả trả về quá lớn.

**+ Stored Procedure hỗn hợp (Hybrid Stored Procedure):**

Ưu điểm: Kết hợp ưu điểm của cả Stored Procedure có tham số đầu vào và đầu ra.

Khuyết điểm: Việc định nghĩa các tham số phức tạp có thể làm cho Stored Procedure trở nên khó hiểu.

**+ Stored Procedure System (System Stored Procedure):**

Ưu điểm: Cung cấp các chức năng quản lý hệ thống cơ sở dữ liệu.

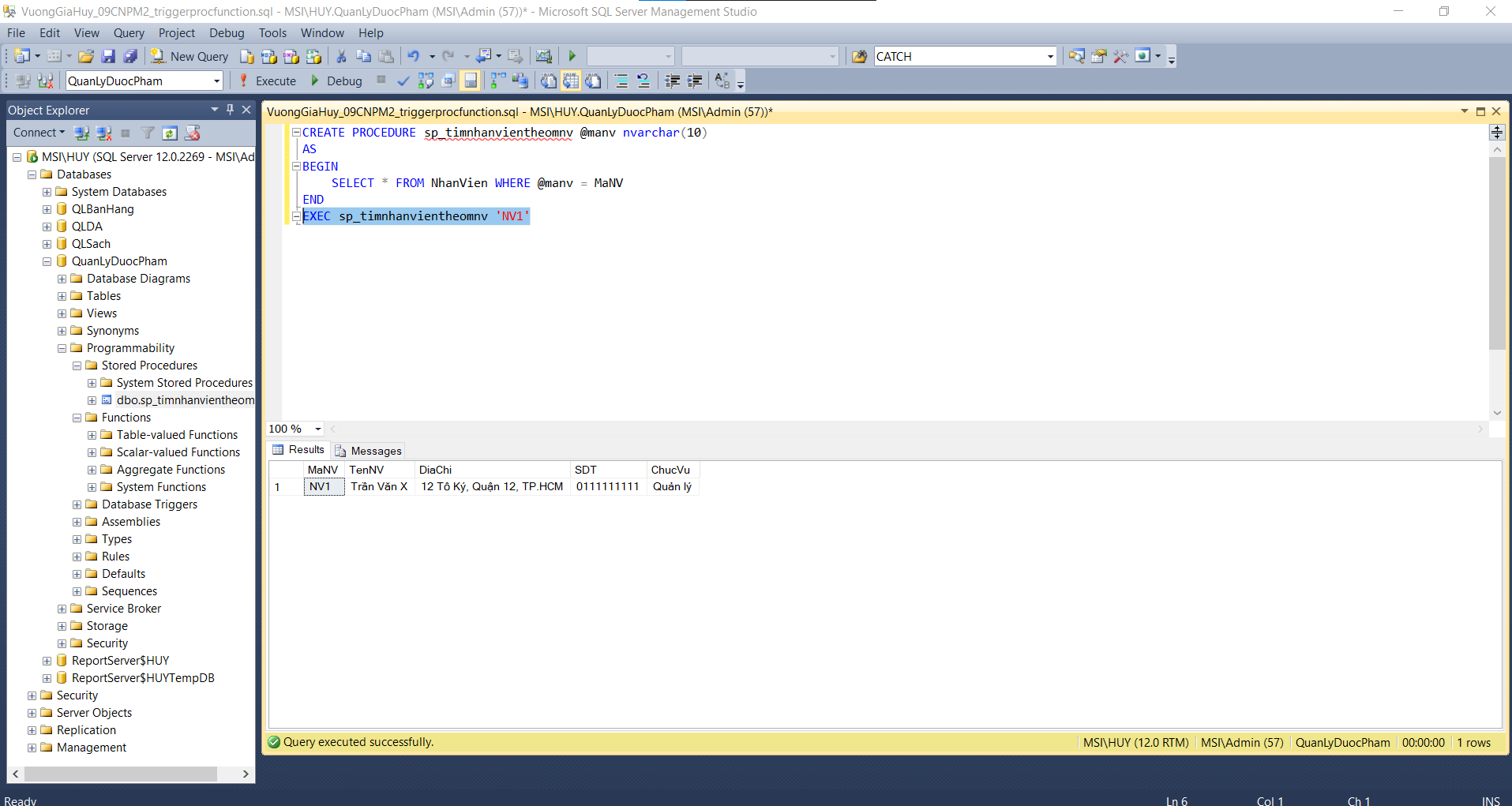
Khuyết điểm: Có thể ảnh hưởng đến hiệu suất của hệ thống nếu được sử dụng sai cách.

**+ Stored Procedure mở rộng (Extended Stored Procedure):**

Ưu điểm: Cung cấp các chức năng phức tạp không có sẵn trong SQL Server.

Khuyết điểm: Có thể tạo ra lỗ hổng bảo mật nếu được sử dụng không đúng cách.

Ví dụ: Tìm nhân viên theo mã nhân viên



### 2.3.2. Function

* **Định nghĩa**

Trong SQL, function là một đối tượng dùng để thực hiện các thao tác tính toán, xử lý dữ liệu và trả về kết quả. Một function có thể nhận tham số hoặc không có tham số nhưng luôn trả về một giá trị duy nhất hoặc một bảng kết quả.

* + **Cú pháp:**

Function có thể trả về 1 giá trị duy nhất hoặc 1 bảng dữ liệu. Cú pháp trả về 1 giá trị duy nhất.

CREATE FUNCTION tên function (@tham số kiểu dữ liệu,…)

RETURNS kiểu dữ liệu trả về

AS

BEGIN

PHẦN CODE TA CẦN FUNCTION THỰC HIỆN

END

Cú pháp thực thi function:

SELECT dbo.tên function(tham số)

Cú pháp trả về 1 bảng dữ liệu

CREATE FUNCTION tên function (@tham số kiểu dữ liệu,…)

RETURNS @tên bảng table (tên cột kiểu dữ liệu, tên cột kiểu dữ liệu…)

AS

BEGIN

PHẦN CODE TA CẦN FUNCTION THỰC HIỆN

END

Cú pháp thực thi function:

SELECT \* FROM dbo.tên function(tham số)

**Các loại function:**

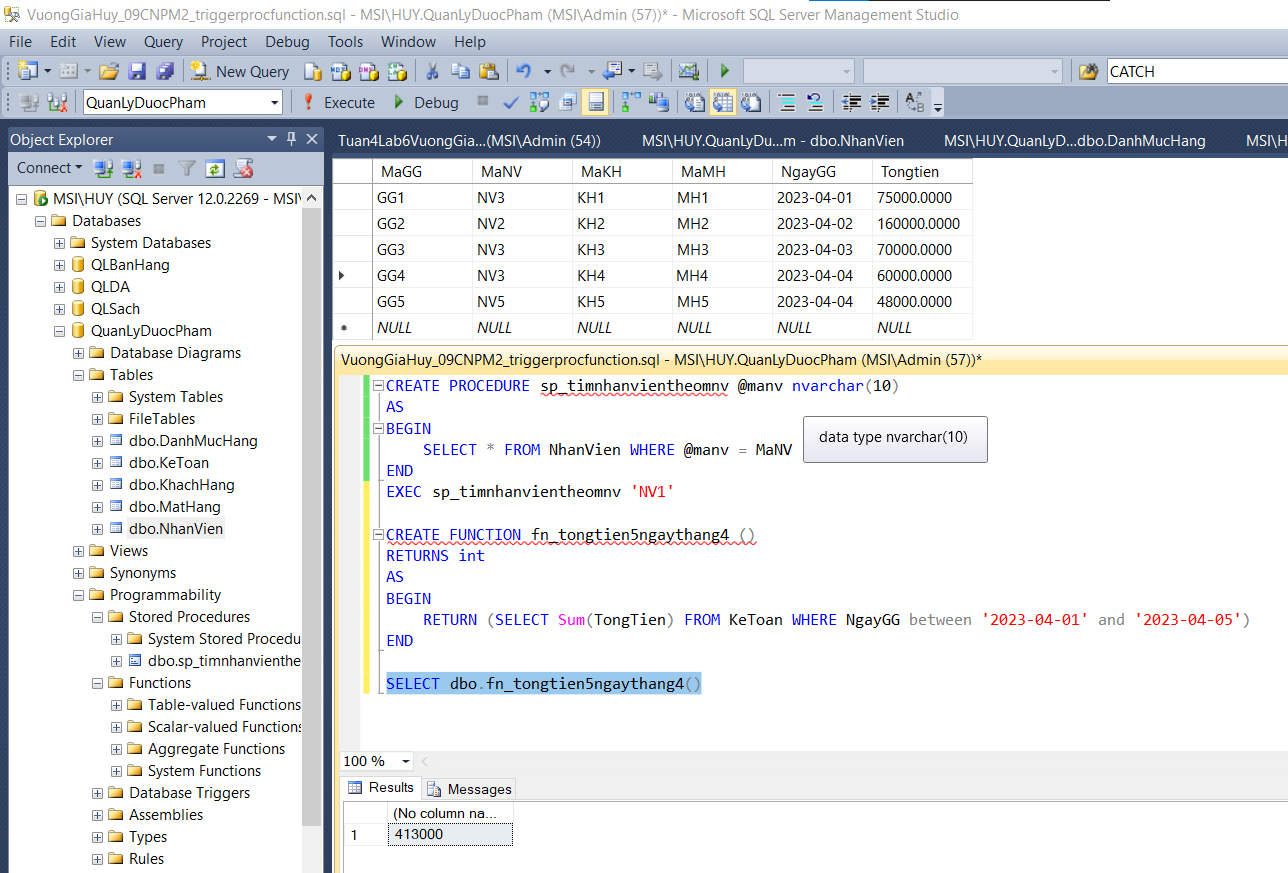
Trong SQL Server có nhiều loại hàm (function), bao gồm:

* Scalar-valued function: Hàm trả về một giá trị duy nhất dựa trên tham số đầu vào.
* Table-valued function: Hàm trả về một bảng dữ liệu dựa trên tham số đầu vào.
* Aggregate function: Hàm trả về một giá trị tổng hợp (sum, avg, count, max, min) của một tập dữ liệu đầu vào.
* Ranking function: Hàm trả về một giá trị liên quan đến vị trí (rank, dense\_rank, row\_number) của một bản ghi trong kết quả truy vấn.
* String function: Hàm thao tác trên chuỗi ký tự (concat, substring, charindex, replace, len, lower, upper, trim, ltrim, rtrim).
* Date and time function: Hàm thao tác trên dữ liệu thời gian và ngày tháng (getdate, datepart, datediff, dateadd, convert, format).
* Mathematical function: Hàm toán học (sqrt, power, round, floor, ceiling, abs).
* Logical function: Hàm trả về giá trị boolean (and, or, not).
* System function: Hàm được cung cấp sẵn bởi SQL Server (host\_name, db\_name, user\_name, @@rowcount).

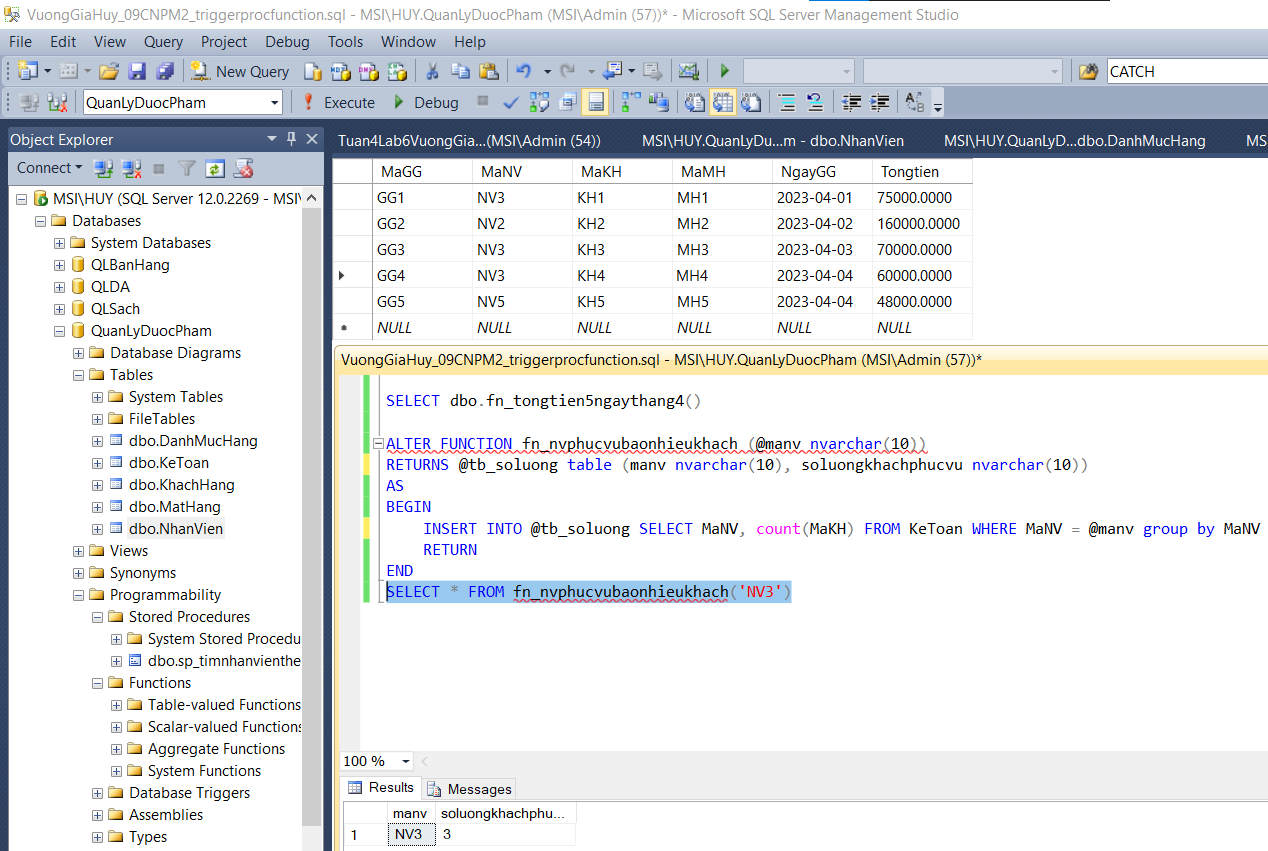
Ưu và khuyết điểm của từng loại:

* Scalar function: Ưu điểm là trả về giá trị duy nhất và linh hoạt trong việc sử dụng, tuy nhiên có thể làm chậm tốc độ thực thi của truy vấn nếu được sử dụng trong các truy vấn lớn.
* Table-valued function: Ưu điểm là trả về một bảng dữ liệu có thể được sử dụng như một bảng thông thường trong các truy vấn khác, tuy nhiên nó có thể làm chậm tốc độ thực thi của truy vấn nếu kết quả trả về quá lớn.
* Aggregate function: Ưu điểm là tính được các giá trị tổng hợp của tập dữ liệu đầu vào, tuy nhiên, nó không thể sử dụng được trong các truy vấn cần trả về các giá trị khác.
* Ranking function: Ưu điểm là trả về thông tin về vị trí của các bản ghi trong kết quả truy vấn, tuy nhiên, nó không thể sử dụng được trong các truy vấn cần trả về các giá trị khác.
* String function: Ưu điểm là thao tác được trực tiếp trên chuỗi ký tự, tuy nhiên, nó có thể làm chậm tốc độ thực thi của truy vấn nếu được sử dụng trong các truy vấn lớn.
* Date and time function: Ưu điểm là thao tác được trực tiếp trên dữ liệu thời gian và ngày tháng, tuy nhiên, nó có thể làm chậm tốc độ thực thi của truy vấn nếu được sử dụng trong các truy vấn lớn.
* Mathematical function: Ưu điểm là tính được các giá trị toán học của dữ liệu đầu vào, tuy nhiên, nó không thể sử dụng được trong các truy vấn cần trả về các giá trị khác.
* Logical function: Ưu điểm là trả về giá trị boolean dựa trên dữ liệu đầu vào, tuy nhiên, nó không thể sử dụng được trong các truy vấn cần trả về các giá trị khác.
* System function: Ưu điểm là cung cấp các thông tin hệ thống liên quan đến phiên làm việc, tuy nhiên, nó không thể sử dụng được trong các truy vấn cần trả về các giá trị khác.

Ví dụ: Hàm tính tổng số tiền đã thu được trong 5 ngày đầu của tháng 4



Ví dụ: Hàm tính số lượng nhân viên đó phục vụ khi ta nhập mã nhân viên.



### 2.3.3. Trigger

**- Định nghĩa:**

Trigger thường được sử dụng để giám sát và kiểm soát các thay đổi dữ liệu trong cơ sở dữ liệu, như khi chèn, cập nhật hoặc xóa bản ghi từ một bảng cụ thể. Trigger được kích hoạt tự động và thực hiện các hành động được xác định trước khi hoặc sau khi một sự kiện được thực hiện trên bảng hoặc cơ sở dữ liệu.

- Cú pháp với Trigger DML

CREATE TRIGGER tên trigger ON tên bảng ta cần thực hiện

FOR INSERT/UPDATE/DELETE

AS

BEGIN

PHẦN LỆNH TA CẦN THỰC HIỆN

END

- Cú pháp với Trigger AFTER

CREATE TRIGGER tên trigger ON tên bảng ta cần thực hiện

AFTER INSERT/UPDATE/DELETE

AS

BEGIN

PHẦN LỆNH TA CẦN THỰC HIỆN

END

- Cú pháp với Trigger INSTEAD OF

CREATE TRIGGER tên trigger ON tên bảng ta cần thực hiện

INSTEAD OF INSERT/UPDATE/DELETE

AS

BEGIN

PHẦN LỆNH TA CẦN THỰC HIỆN

END

**- Các loại Trigger:**

* Trigger DML (Data Manipulation Language): Là trigger được kích hoạt khi các hoạt động DML (Insert, Update, Delete) được thực hiện trên một bảng hoặc một view. Trigger DML có thể được kích hoạt trước hoặc sau khi thực hiện hoạt động DML.
* Trigger AFTER: Là trigger được kích hoạt sau khi hoạt động được thực hiện trên một bảng hoặc một view. Trigger AFTER có thể được sử dụng để kiểm tra hoặc thay đổi dữ liệu sau khi nó được thêm, cập nhật hoặc xóa.
* Trigger INSTEAD OF: Là trigger được kích hoạt thay cho hoạt động được yêu cầu trên một bảng hoặc một view. Trigger INSTEAD OF có thể được sử dụng để thay đổi hoặc kiểm tra dữ liệu trước khi thực hiện hoạt động.

**- Ưu và khuyết điểm của từng loại**

Trigger DML:

Ưu điểm: có thể kiểm soát được các hoạt động DML trên bảng hoặc view.

Nhược điểm: có thể ảnh hưởng đến hiệu suất nếu trigger được kích hoạt quá thường xuyên.

Trigger AFTER:

Ưu điểm: cho phép thay đổi dữ liệu sau khi hoạt động được thực hiện.

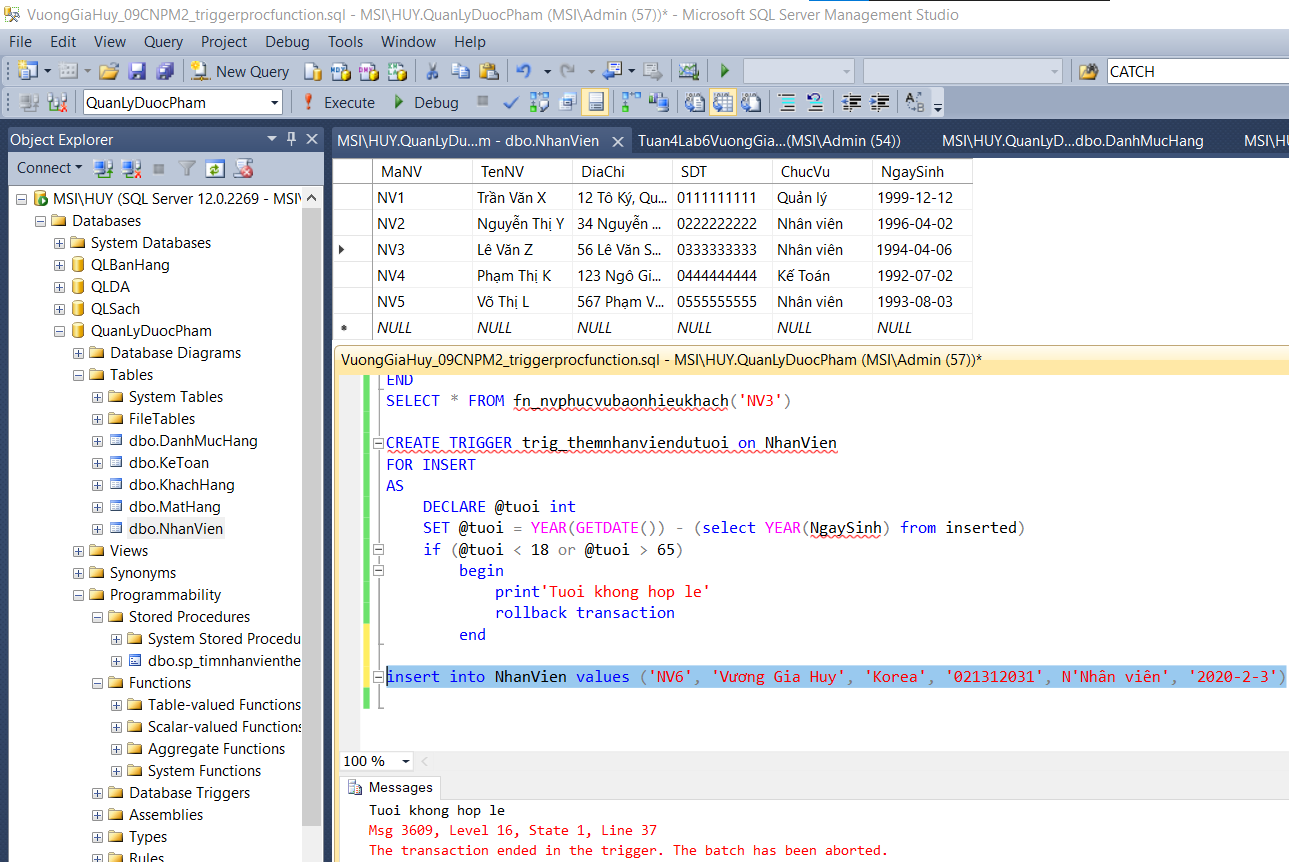
Nhược điểm: có thể làm chậm tốc độ thực thi nếu trigger được sử dụng để thực hiện các thao tác phức tạp.

Trigger INSTEAD OF:

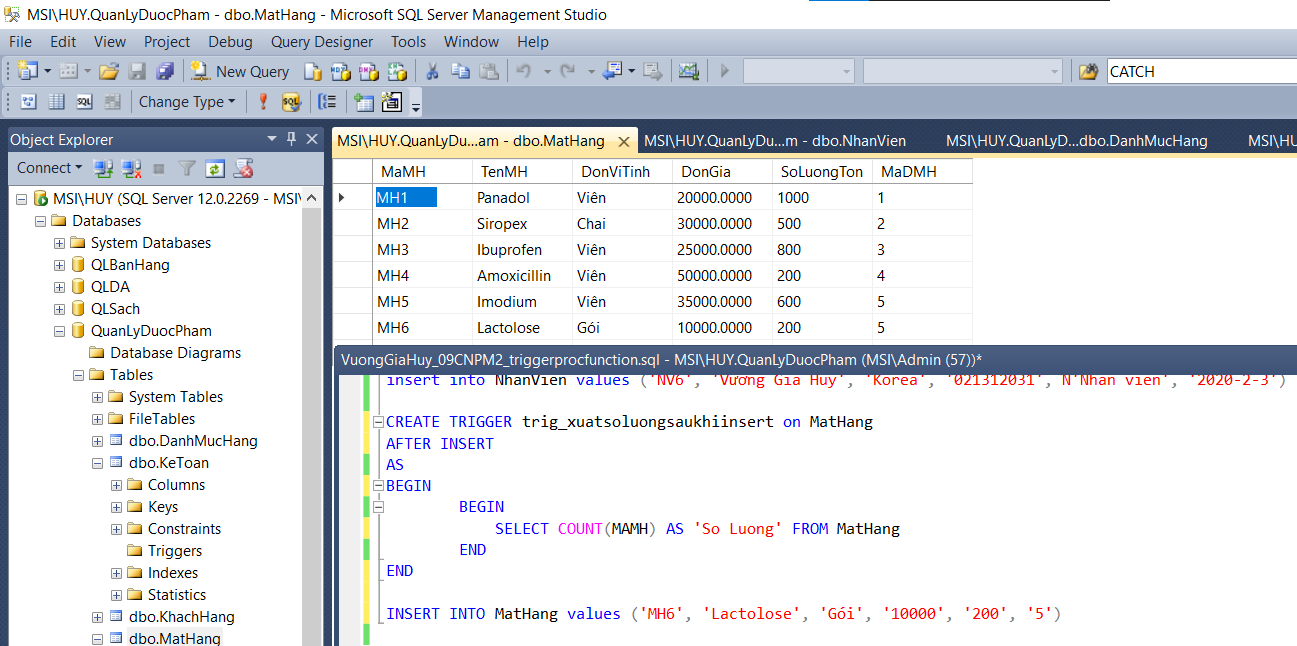
Ưu điểm: cho phép thay đổi hoặc kiểm tra dữ liệu trước khi thực hiện hoạt động.

Nhược điểm: có thể làm chậm tốc độ thực thi nếu trigger được sử dụng để thực hiện các thao tác phức tạp. Ngoài ra, trigger INSTEAD OF không hoạt động với các hoạt động DDL (Data Definition Language) như tạo bảng hoặc xóa bảng.

- Ví dụ: Trigger không cho phép thêm nhân viên dưới 18 hoặc trên 65 tuổi.



Trigger sau khi insert vào bảng Mặt hàng thì sẽ xuất tổng số lượng mặt hàng của bảng đó.



## 2.4. Bảo mật, phân quyền

Trong SQL, bảo mật giúp cho dữ liệu chỉ được truy cập bởi những người dùng mà ta cho phép, giúp dữ liệu được bảo mật an toàn. Còn với phân quyền sẽ cho phép ta quản lý, kiểm soát cơ sở dữ liệu đó được truy cập và thao tác bởi người dùng cụ thể.

Trong quản lý dược phẩm nói riêng thì việc này đóng một vai trò hết sức quan trọng trong việc bảo vệ, kiểm soát thông tin các loại dược phẩm. Chẳng hạn, người dược sĩ thì sẽ biết được đang có những mặt hàng nào, những mặt hàng sắp được cung cấp. Trong khi người kế toán thì quản lý thu chi, kinh doanh của cửa hàng.

Cụ thể ta có thể thực hiện như sau:

* Tạo tài khoản cho mỗi nhân viên theo chức vụ và cấp cho họ quyền truy cập vào các bảng dữ liệu chỉ liên quan đến công việc của họ.
* Ngăn chặn những người dùng vi phạm các chính sách của công ty.

Bảo mật trong SQL gồm 3 lớp:

* + Login security: người dùng có thể đăng nhập vào SQL Server.
  + Database access security: người dùng có thể truy cập vào 1 cơ sở dữ liệu cụ thể trên Server.
  + Permission security: người dùng có thể thực hiện những thao tác nào trên cơ sở dữ liệu.

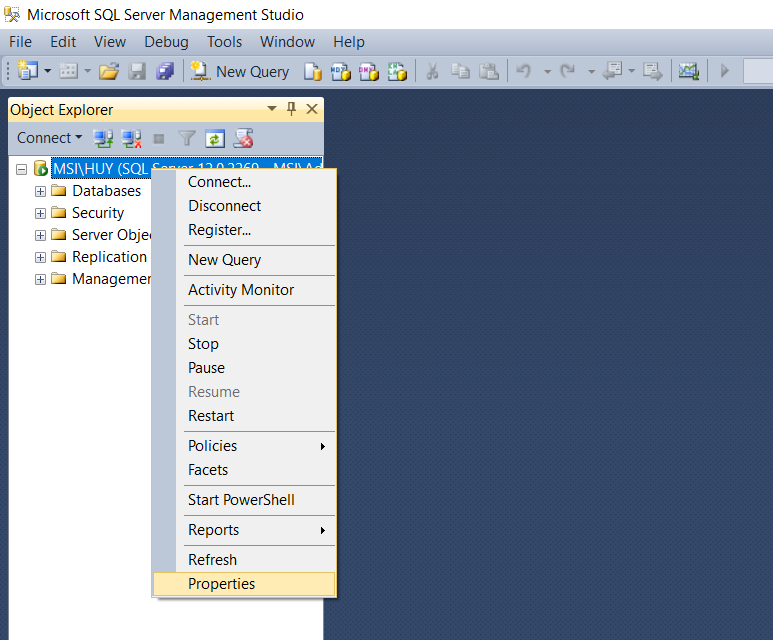
### 2.4.1. Login Security

Với Login security thì có 2 chế độ xác thực đó là Windows Authenication và SQL Authenication.

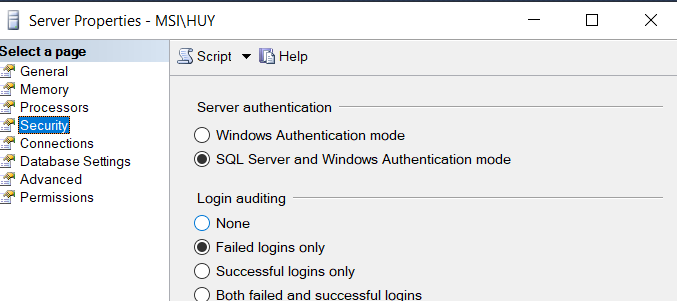
* Windows Authenication: người dùng chỉ cần có account trong Windows thì có thể xác thực được. SQL Server sẽ dựa vào Windows để xác thực cho người dùng.
* SQL Authenication: người quản trị sẽ tạo tài khoản và mật khẩu để đăng nhập vào SQL Server. Thông tin đăng nhập sẽ được lưu trong bảng syslogins của CSDL Master.

Để thay đổi giữa Windows Authenication và SQL Authenication ta có thể làm như sau:

Click chuột phải vào tài khoản ta đăng nhập chọn Properties

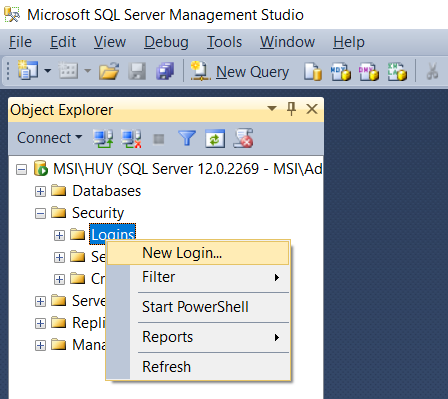


Sau đó chọn tab Security và chọn Windows hoặc SQL xong nhấn OK.



* Để tạo login trong SQL Server ta làm như sau:

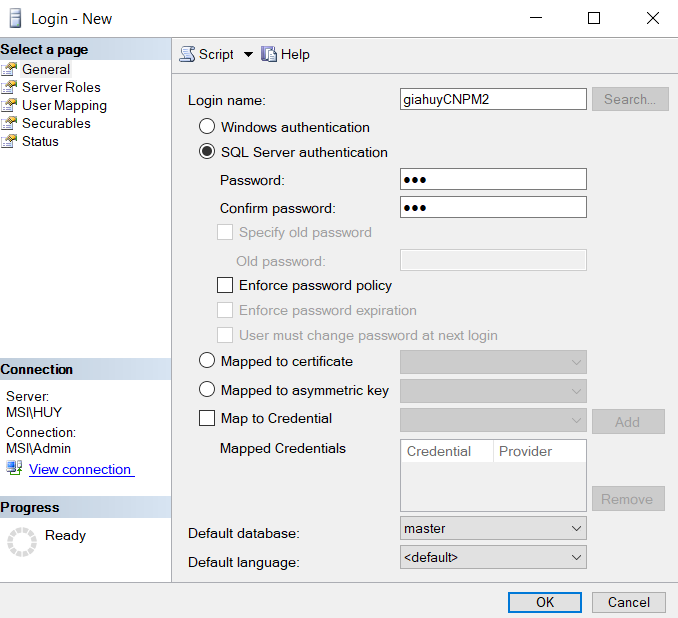
Trong Object Explorer ta mở thư mục Security trong Server ra, click chuột phải vào Logins chọn New Login.



Trong bảng New Login thì ta set tài khoản và mật khẩu mà ta muốn tạo, sau đó tick vào những mục ta cần cấu hình.

Cụ thể:

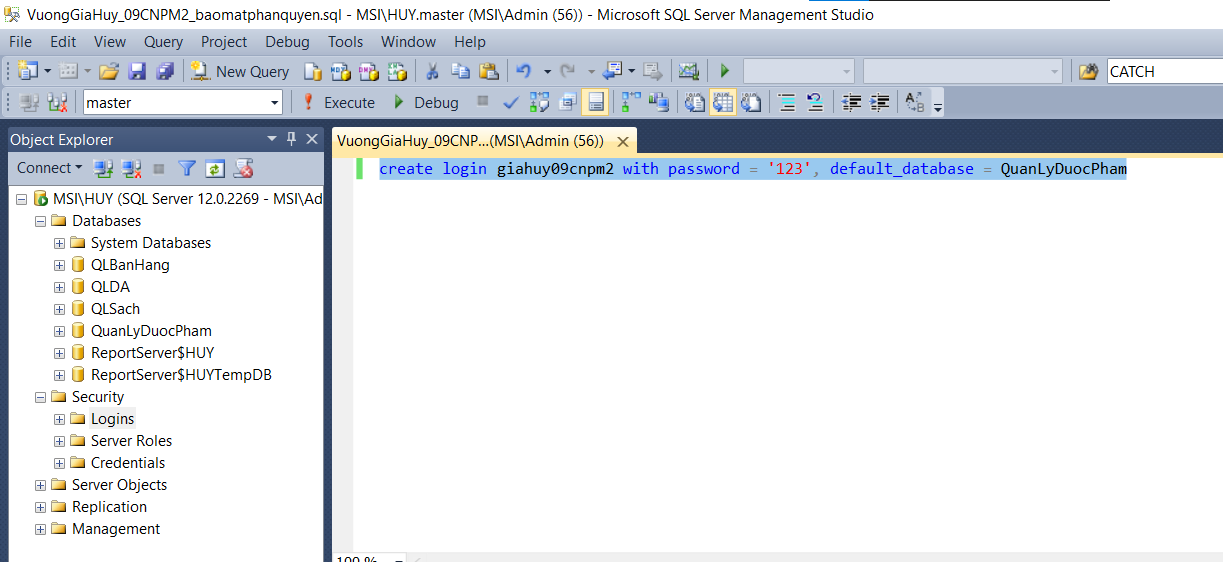
* Enforce Password Policy: Là áp dụng những điều kiện của mật khẩu như là độ dài tối thiểu, bao gồm chữ hoa, số…
* Enforce password expiration: người dùng sẽ được yêu cầu thay đổi mật khẩu vào một thời điểm nhất định (theo cấu hình) hoặc khi lần đăng nhập tiếp theo. Chức năng này giúp tăng cường bảo mật và giảm rủi ro của việc sử dụng mật khẩu cũ quá lâu.
* User must change password at next login: Người dùng phải đổi mật khẩu khi đăng nhập lần đầu.



* Để tạo Login bằng SQL bằng câu truy vấn SQL thì ta có thể sử dụng cú pháp:

CREATE LOGIN tên\_login WITH PASSWORD =’matkhau’ [MUST\_CHANGE]   
[,DEFAULT\_DATABASE = database\_name]   
[,DEFAULT\_LANGUAGE = language]   
[,CHECK\_EXPIRATION = {ON|OFF}   
[,CHECK\_POLICY = {ON|OFF}

Ví dụ: tạo login với tên login là giahuyCNPM2 với pass là 123



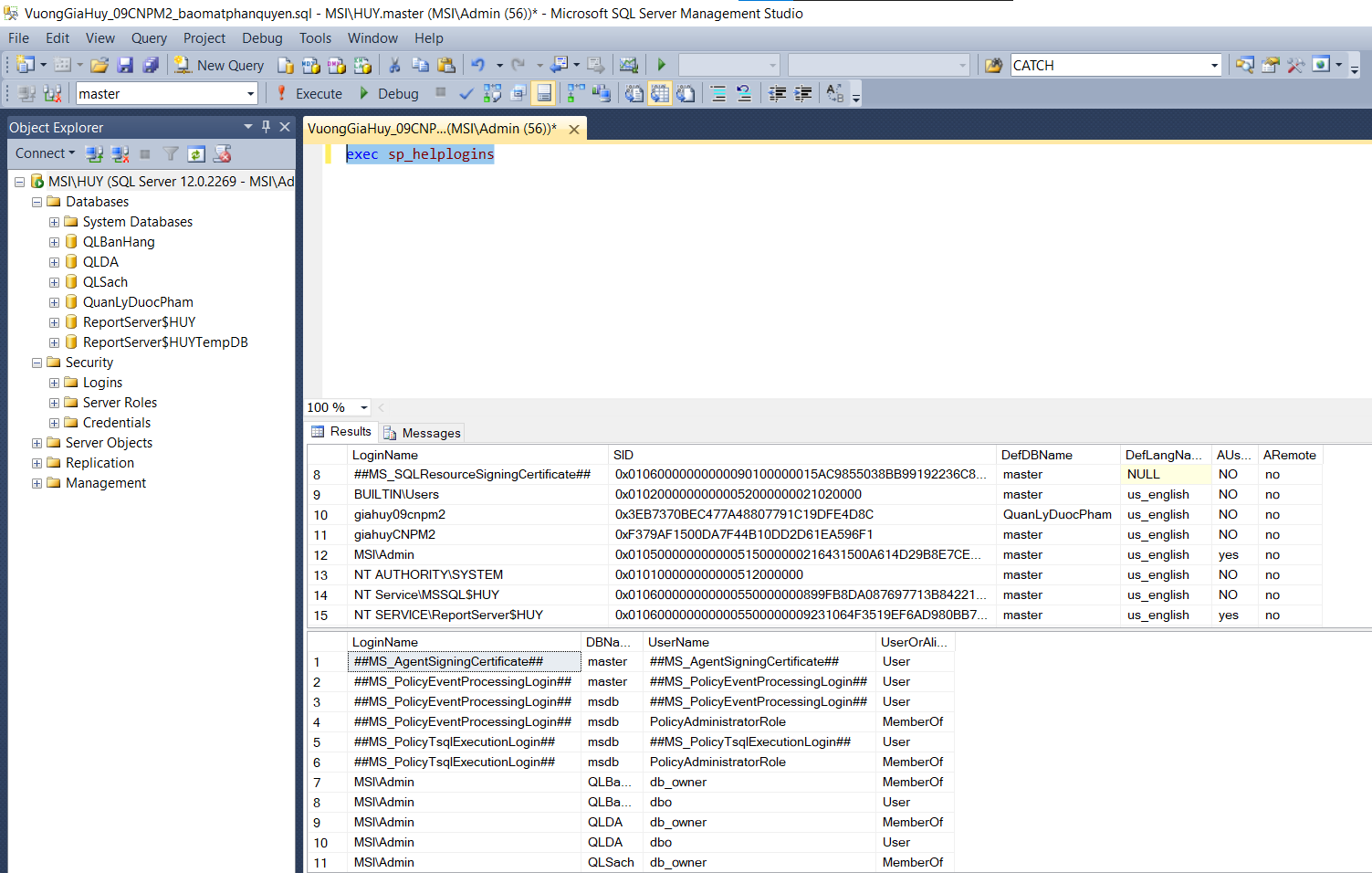
Để tạo Login bằng Windows bằng câu truy vấn SQL thì ta có thể sử dụng cú pháp:

CREATE LOGIN [tên domain\tên user] FROM WINDOWS

Ví dụ: CREATE LOGIN [MSI\HUY] FROM WINDOWS

Ta cũng có thể dùng ALTER và DROP để thay đổi password hoặc xóa login của người dùng. Để xem các login có sẵn thì ta có thể dùng cú pháp:

EXEC sp\_helplogins



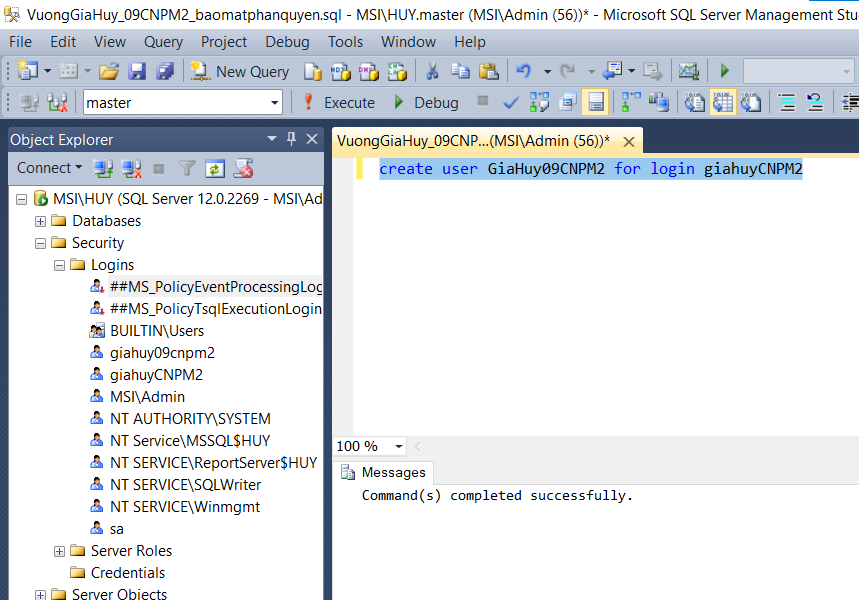
### 2.4.2. Database user Security

Mỗi database thì sẽ có danh sách người dùng cụ thể được xác thực có thể truy cập database đó. Người dùng chỉ có quyền xem các database chứ không có quyền thao tác trên đó. Để thao tác được thì người dùng cần phải được cấp quyền.

Ta tạo user bằng câu truy vấn SQL với cú pháp:

CREATE USER tênuser [{FOR|FROM} LOGIN tendangnhap]  
[WITH DEFAULT\_SCHEMA=tenschema]

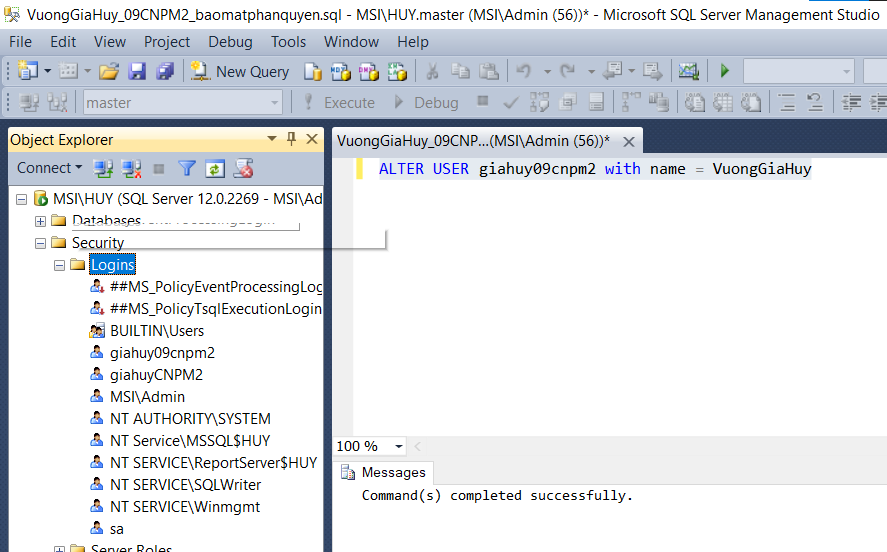
Ví dụ ta tạo user giahuy09cnpm2 trong login giahuycnpm2

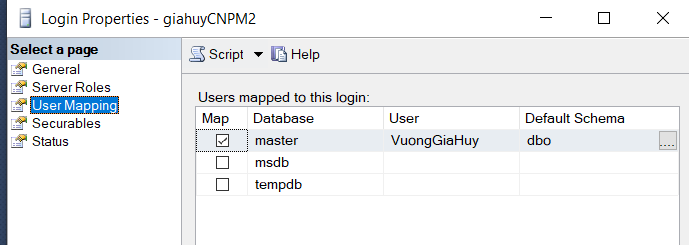


Ta cũng có thể dùng ALTER và DROP để chỉnh sửa và xóa user. Cú pháp:

ALTER USER tênuser WITH [NAME = tên mới] [,DEFAULT\_SCHEMA = tên schema]

DROP USER tênuser

Ví dụ: 



### 2.4.3. Permission Security

Trong SQL có 3 loại Permission:

* Object Permissions
* Statement Permissions
* Implied Permissions

Các quyền trong SQL tồn tại ở 3 trạng thái: GRANTED (được cấp), REVOKED (bị thu hồi), DENIED (từ chối).

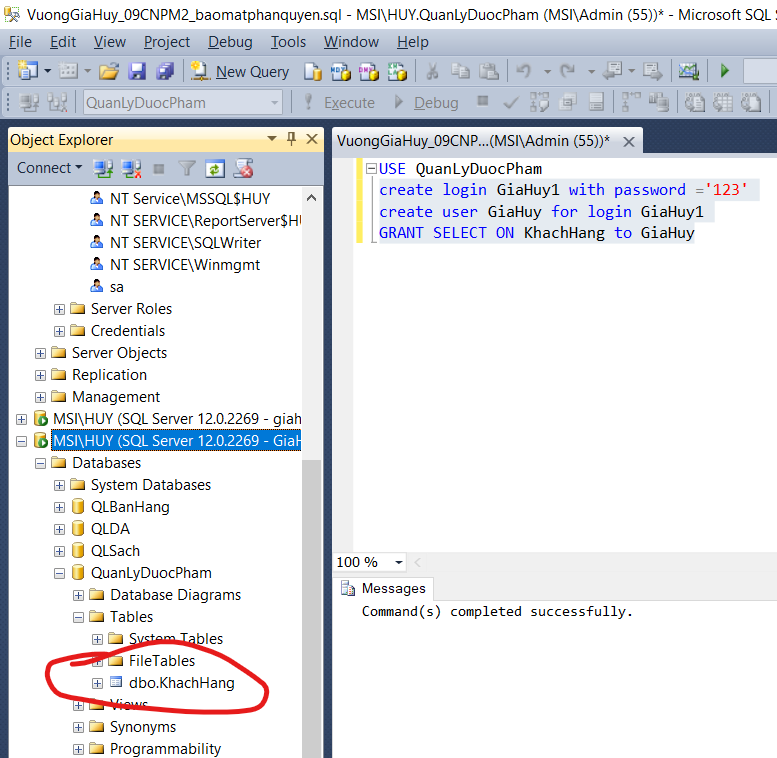
#### GRANT:

Dùng để cấp phát quyền cho người dùng cụ thể trên cơ sở dữ liệu trong 2 trường hợp.

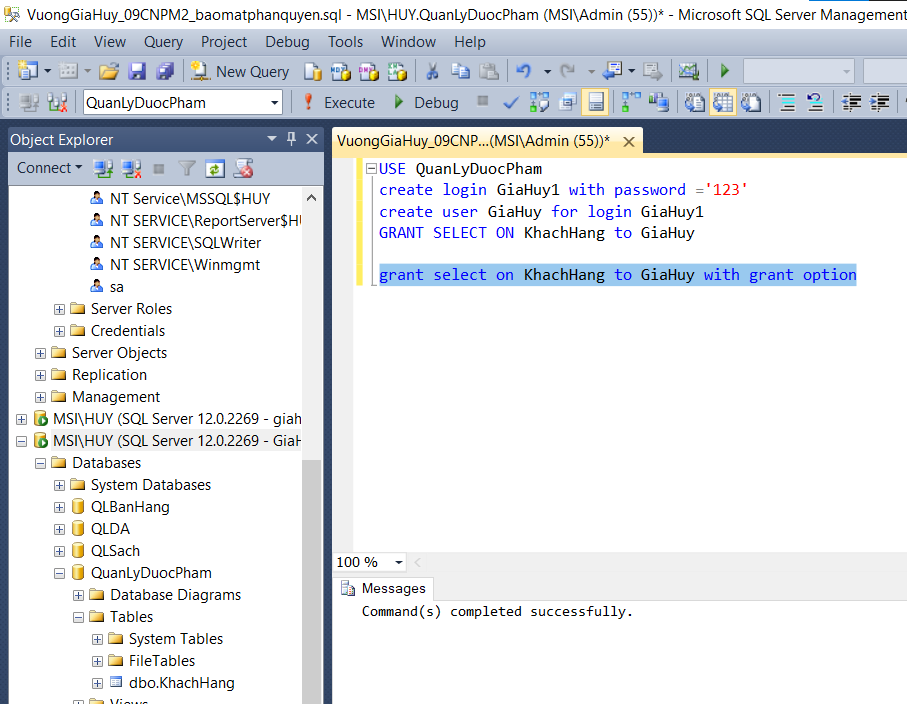
* Cấp quyền trên các đối tượng cơ sở dữ liệu như là các bảng.
* Cấp quyền thực thi câu lệnh trên cơ sở dữ liệu như là insert, update, delete…

Các quyền đó gồm các thao tác trên cơ sở dữ liệu như insert, update, delete, tạo hàm, khung nhìn, thủ tục…

Ví dụ cấp cho user GiaHuy trong login GiaHuy1 quyền xem bảng KhachHang. Khi đó user GiaHuy chỉ xem được mỗi bảng KhachHang.



Để có thể cấp cho người dùng GiaHuy quyền chuyển tiếp quyền cho người dùng khác ta có thể sử dụng cú pháp sau:



#### REVOKE

Được sử dùng để thu hồi quyền đã cấp cho người dùng. REVOKE được sử dụng trong 2 trường hợp:

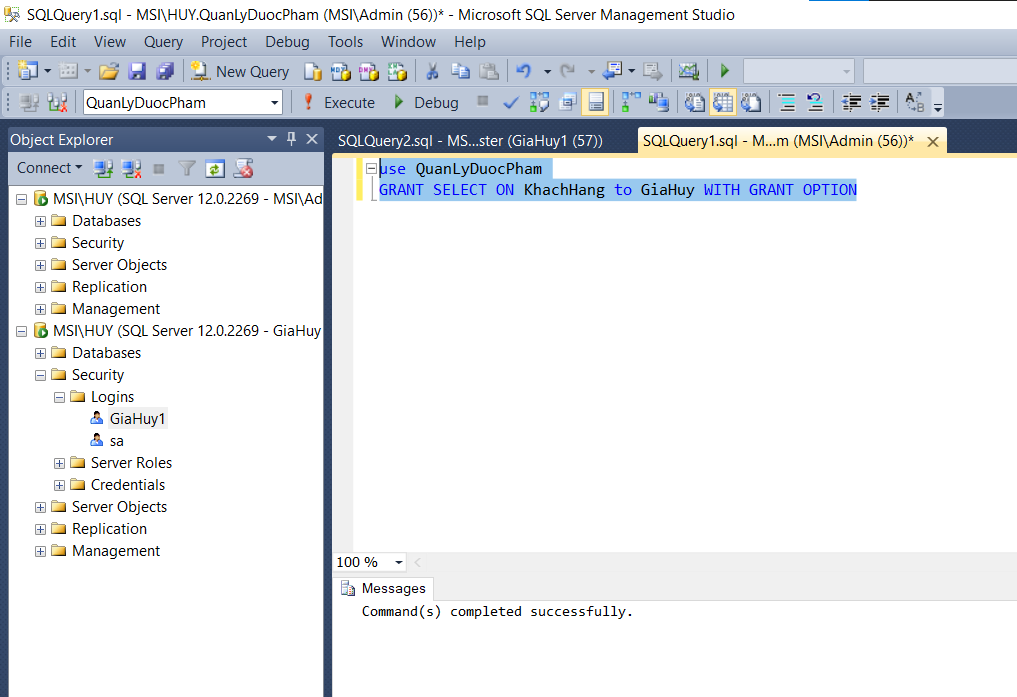
* Thu hồi quyền đã cấp trên các đối tượng cơ sở dữ liệu như là các bảng.
* Thu hồi quyền thực thi câu lệnh trên cơ sở dữ liệu như là insert, update, delete…

Tuy nhiên ta chỉ có thể thu hồi quyền mà ta đã cấp cho người dùng đó, những quyền được cấp bởi người dùng khác thì vẫn còn hiệu lực.

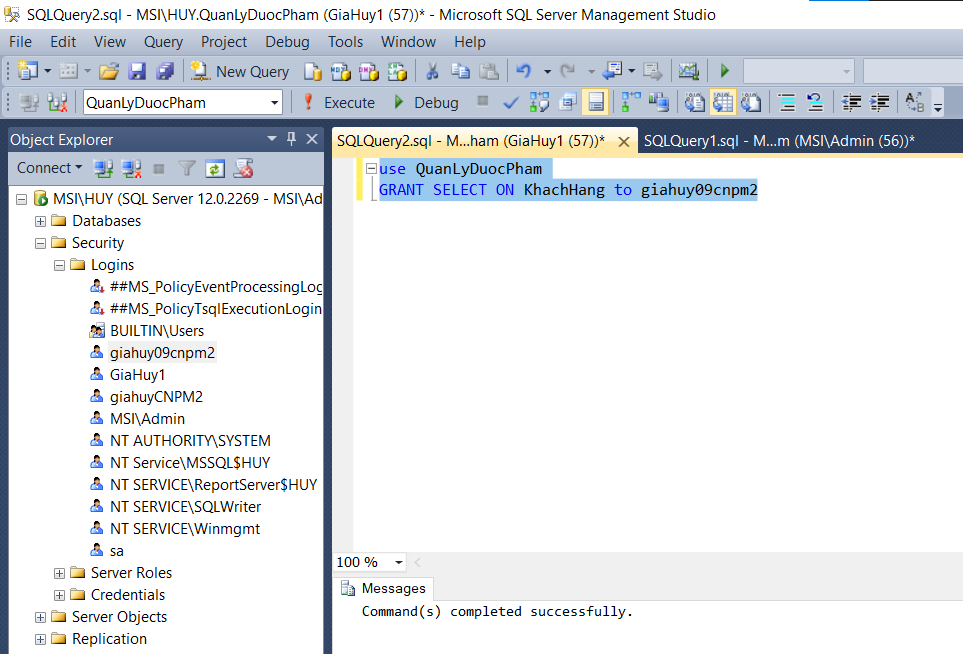
Khi ta cấp 1 quyền cho 1 người dùng mà người dùng đó cũng có thể chuyển tiếp quyền đó cho 1 người dùng khác. Để thu hồi quyền của cả 2 thì ta có thể dùng cú pháp:

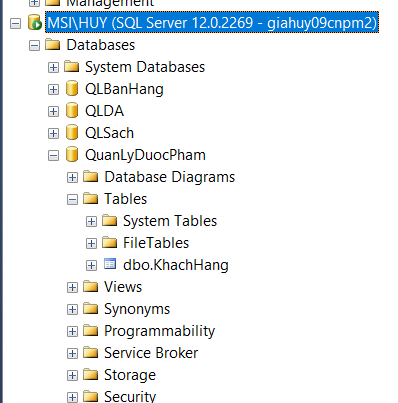
REVOKE lệnh ta cần thu hồi ON bảng FROM tên người dùng CASCADE

Ví dụ ở đây ta cấp cho GiaHuy quyền cấp SELECT cho người dùng khác.

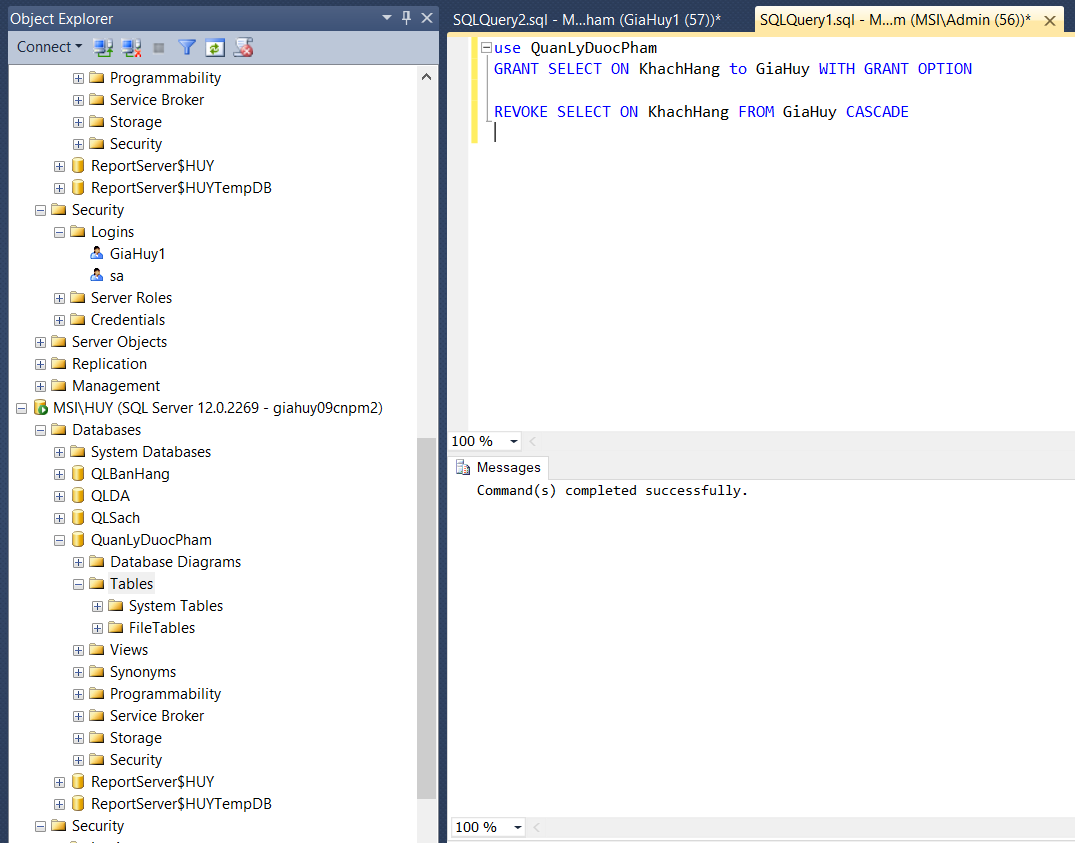


Thì khi đó GiaHuy có thể cấp cho người dùng khác là giahuy09cnpm2 quyền đó.





Sau khi ta dùng cú pháp trên thì cả GiaHuy và giahuy09cnpm2 đều mất quyền xem bảng Khách hàng.



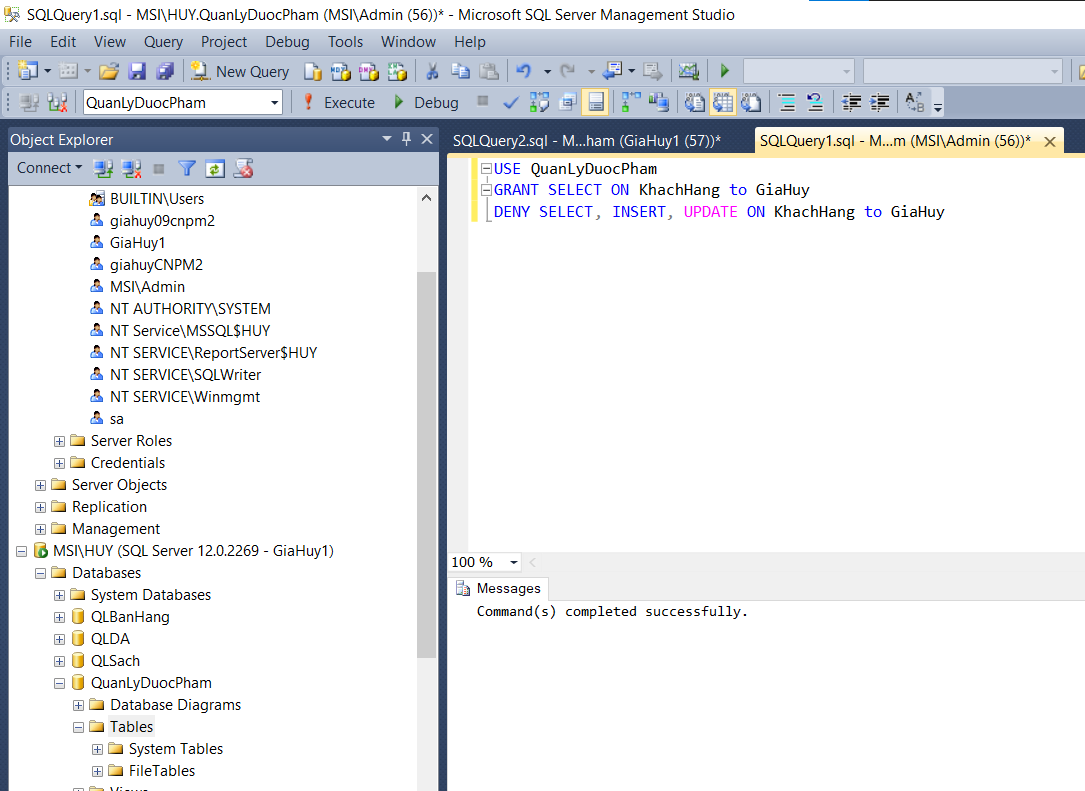
* DENY

DENY được dùng để từ chối 1 quyền nào đó và ngăn người dùng thừa kế quyền đó thông qua mối quan hệ thành viên trong group và role.

Cú pháp

DENY quyền ON bảng TO user/role

Ở đây ta DENY tất cả các quyền của GiaHuy trên bảng Khách hàng khi đó GiaHuy sẽ không thể SELECT, INSERT, UPDATE trên bảng Khách hàng.



# Chương 3: KẾT LUẬN

## 3.1. Những kết quả đạt được của đồ án

* Xây dựng thành công hệ thống quản lý dược phẩm trên nền tảng hệ quản trị cơ sở dữ liệu, giúp cho quá trình quản lý dược phẩm trở nên hiệu quả hơn.
* Hiểu rõ về các kiểu dữ liệu, thuộc tính và ràng buộc trong quản lý cơ sở dữ liệu.
* Hiểu thêm về mối quan hệ giữa các bảng và cách thiết kế cơ sở dữ liệu sao cho phù hợp với yêu cầu của hệ thống.
* Hiểu thêm về các công cụ và ngôn ngữ truy vấn cơ sở dữ liệu SQL để truy vấn và xử lý dữ liệu.
* Xây dựng các Stored Procedure, Function, Trigger để tối ưu hóa quy trình xử lý dữ liệu và tăng tính bảo mật.
* Nhóm có được kiến thức về bảo mật và phân quyền trong hệ quản cơ sở dữ liệu cụ thể là trong SQL Server giúp đảm bảo được tính an toàn về dữ liệu.

## Nhược điểm của đồ án

Vì đồ án thực hiện trên hệ quản trị cơ sở dữ liệu SQL là dữ liệu có cấu trúc nên bao gồm các ràng buộc về dữ liệu. Một mặt, nó đảm bảo dữ liệu luôn đầy đủ thông tin khi được thêm vào dữ liệu, cũng chính vì vậy nên khi thêm mới, cập nhật và xóa dữ liệu cũng không được tiện lợi lắm vì có các ràng buộc giữa các bảng.

Đồng thời đồ án cũng không dễ mở rộng do phải thay đổi cấu trúc của dữ liệu khi dữ liệu ngày một lớn. Bên cạnh đó, SQL cũng không có tốc độ nhanh với các ứng dụng có tần suất truy cập dữ liệu lớn.

SQL cũng không cung cấp các bảo mật tốt với các bảo mật từ phía bên trong. Nên việc bảo mật của đồ án cũng sẽ gặp khó khăn nếu không có các biện pháp bổ sung. SQL không có tính nhất quán về cú pháp và hỗ trợ giữa các hệ quản trị cơ sở dữ liệu khác nhau nên có thể dẫn đến khó khăn trong việc di chuyển dữ liệu giữa các hệ thống khác nhau.

Vì kiến thức còn hạn hẹp nên những thiếu sót là điều không thể tránh khỏi. Đồ án này chỉ bao gồm cơ sở dữ liệu, những truy vấn đơn giản có thể thực hiện trong SQL Server. Và đồ án không có giao diện như 1 trang web hoàn chỉnh. Phần tình huống tranh chấp cũng còn thiếu sót nhiều về lý thuyết và ứng dụng.

## Phụ lục

Giao diện các chức năng của chương trình (trình bày hình giao diện or form) nếu có.

# TÀI LIỆU THAM KHẢO

function

<https://freetuts.net/gioi-thieu-function-trong-sql-server-1617.html>

https://truyenhinhcapsongthu.net/top/function-nguoi-dung-tu-dinh-nghia-trong-sql-viblo/woR7FzS7Iqgeyel

procedure

https://freetuts.net/gioi-thieu-stored-procedure-trong-sql-server-1609.html

trigger

https://expressmagazine.net/development/1486/cach-viet-va-su-dung-stored-procedures

<https://fptcloud.com/trigger-trong-sql/>

<https://wiki.tino.org/trigger-trong-sql/>

bảo mật phân quyền

https://doan.edu.vn/do-an/bai-giang-microsoft-sql-server-2008-chuong-11-bao-mat-va-phan-quyen-tran-thi-kim-chi-44592/