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

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



**BÁO CÁO ĐỒ ÁN MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**HỆ THỐNG QUẢN LÝ VẬT LIỆU XÂY DỰNG**

Giảng viên hướng dẫn: Phạm Trọng Huynh

Lớp: 09\_ĐH\_THMT

Nhóm 4

Sinh viên thực hiện:

1. Nguyễn Nhật Hào 0950080100
2. Ngô Lê Minh Thiện 0950080113
3. Đặng Ngọc Gia Bảo 0950080082

Tp. Hồ Chí Minh, tháng 8 năm 2023

PHIẾU ĐÁNH GIÁ CỦA GIẢNG VIÊN

Nhận xét:

Đánh giá:

GIẢNG VIÊN

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 7](#_Toc143466441)

[1.1. Lời giới thiệu 8](#_Toc143466442)

[1.2. Phân tích hệ thống quản lí vật liệu xây dựng 8](#_Toc143466443)

[1.2.1. Thực trạng của hệ thống quản lý vật liệu xây dựng 8](#_Toc143466444)

[1.2.2. Biểu đồ phần cấp chức năng 9](#_Toc143466445)

[1.2.3. Biểu đồ luồng dữ liệu 9](#_Toc143466447)

[1.2.4. Phân loại các thuộc tính vào một tập thực thể 10](#_Toc143466449)

[1.2.5. Xây dựng CSDL trên hệ QTCSDL 10](#_Toc143466450)

[1.3. Phân quyền 11](#_Toc143466451)

[**CHƯƠNG 2: ỨNG DỤNG LÝ THUYẾT VÀO HỆ THÔNG QUẢN LÝ VẬT LIỆU XÂY DỰNG** 14](#_Toc143466452)

[2.1. Xây dựng CSDL trên HQT CSDL 14](#_Toc143466453)

[2.1.1. Câu lệnh tạo CSDL, tạo bảng, tạo khóa 14](#_Toc143466454)

[2.1.2. Cấu trúc của mỗi bảng 16](#_Toc143466455)

[2.1.3. Dữ liệu của mỗi bảng 18](#_Toc143466462)

[2.1.4. Diagram liên kết các bảng 21](#_Toc143466469)

[2.2. Thao tác trên CSDL trên HQT CSDL 22](#_Toc143466471)

[2.2.1. Tạo và xóa tài khoản người dùng sql server bằng công cụ và câu lệnh……. 22](#_Toc143466472)

[2.2.2. Viết câu lệnh insert 29](#_Toc143466479)

[2.2.3. Viết câu lệnh update 30](#_Toc143466480)

[2.3. Store Procedure, Function, Trigger, View 31](#_Toc143466482)

[2.3.1. Store Procedure 31](#_Toc143466483)

[2.3.2. Function 39](#_Toc143466488)

[2.3.3. Trigger 48](#_Toc143466493)

[2.3.4. View 51](#_Toc143466495)

[2.4. Bảo mật và phâm quyền 58](#_Toc143466501)

[**CHƯƠNG 3. KẾT LUẬN** 59](#_Toc143466502)

[3.1 Những kết quả đạt được của đồ án 59](#_Toc143466503)

[4.1 Nhược điểm của đồ án 59](#_Toc143466504)

[**TÀI LIỆU THAM KHẢO** 60](#_Toc143466505)

**Danh Mục Hình Ảnh**

[Hình 1. 1 Bản đồ phân cấp chức năng 9](#_Toc143457345)

[Hình 1. 2 Biểu đồ luồng dữ liệu 10](#_Toc143457346)

[Hình 2. 1 Cấu trúc bảng chi tiết hóa đơn 16](#_Toc143457348)

[Hình 2. 2 Cấu trúc bảng đơn hàng 16](#_Toc143457349)

[Hình 2. 3 Cấu trúc bảng khách hàng 17](#_Toc143457350)

[Hình 2. 4 Cấu trúc bảng nhà cung cấp 17](#_Toc143457351)

[Hình 2. 5 Cấu trúc bảng sản phẩm 18](#_Toc143457352)

[Hình 2. 6 Cấu trúc bảng thông tin thanh toán 18](#_Toc143457353)

[Hình 2. 7 Dữ liệu bảng chi tiết hóa đơn 19](#_Toc143457354)

[Hình 2. 8 Dữ liệu bảng đơn hàng 19](#_Toc143457355)

[Hình 2. 9 Dữ liệu bảng khách hàng 20](#_Toc143457356)

[Hình 2. 10 Dữ liệu bảng nhà cung cấp 20](#_Toc143457357)

[Hình 2. 11 Dữ liệu bảng sản phẩm 21](#_Toc143457358)

[Hình 2. 12 Dữ liệu bảng thông tin thanh toán 21](#_Toc143457359)

[Hình 2. 13 Bảng Diagram 22](#_Toc143457360)

[Hình 2. 14 Tạo login bước 1 24](#_Toc143457361)

[Hình 2. 15 Tạo login bước 2 25](#_Toc143457362)

[Hình 2. 16 Tạo login bước 3 26](#_Toc143457363)

[Hình 2. 17 Tạo login bước 4 27](#_Toc143457364)

[Hình 2. 18 Đăng nhập tài khoản 27](#_Toc143457365)

[Hình 2. 20 Xóa login 28](#_Toc143457366)

[Hình 2. 22 Sau khi update số lượng 31](#_Toc143457367)

[Hình 2. 23 Store procedure thêm sản phẩm 35](#_Toc143457368)

[Hình 2. 24 Store procedure sửa dữ liệu sản phẩm 37](#_Toc143457369)

[Hình 2. 25 Store procedure xóa dữ liệu sản phẩm 38](#_Toc143457370)

[Hình 2. 26 Store procedure hiển thị dữ liệu sản phẩm 39](#_Toc143457371)

[Hình 2. 27 Function tổng tiền đơn hàng 44](#_Toc143457372)

[Hình 2. 28 Function tổng lượng sản phẩm 45](#_Toc143457373)

[Hình 2. 29 Function lấy thông tin chi tiết đơn hàng 46](#_Toc143457374)

[Hình 2. 30 Function tổng đơn hàng của khách hàng 48](#_Toc143457375)

[Hình 2. 31 Trigger thêm dữ liệu đơn hàng nhập tổng tiền 51](#_Toc143457376)

[Hình 2. 32 View hiển thị thông tin chi tiết hóa đơn 54](#_Toc143457377)

[Hình 2. 33 View danh sách sản phẩm và số lượng tồn 55](#_Toc143457378)

[Hình 2. 34 View danh sách đơn hàng của khách hàng 56](#_Toc143457379)

[Hình 2. 35 View danh sách đơn hàng của khách hàng 57](#_Toc143457380)

[Hình 2. 36 View danh sách nhà cung cấp và số lương sản phẩm 58](#_Toc143457381)

# **LỜI MỞ ĐẦU**

Ngày nay với sự phát triển không ngừng của ngành công nghệ thông tin trong tất cả các lĩnh vực khác nhau của xã hội như : kinh tế, chính trị, an ninh quốc phòng, thương mại điện tử và đã có tác động rất lớn đến sự phát triển chung của xã hội. Công nghệ thông tin đối với các ngành doanh nghiệp, công ty ngày càng đóng vai trò quan trọng không thể thiếu. Với việc áp dụng thành tựu tiên tiến hiện đại của ngành công nghệ thông tin đã cho ra đời các chương trình thay thế thủ công làm giảm bớt thời gian, sức lực, tăng độ chính xác và tính bảo mật. Công nghệ thông tin có thể thay thế hoàn toàn con người trong các hoạt động chân tay.

Cùng với sự phát triển nhanh chóng về phần cứng máy tính, các phần mềm ngày càng trở nên đa dạng, phong phú, hoàn thiện hơn và hỗ trợ hiệu quả cho con người. Các phần mềm hiện nay ngày càng mô phỏng được rất nhiều nghiệp vụ khó khăn, hỗ trợ cho người dùng thuận tiện sử dụng, thời gian xử lý nhanh chóng, và một số nghiệp vụ được tự động hoá cao.

Trong bối cảnh ngày càng phức tạp của ngành xây dựng, việc quản lý vật liệu trở thành một thách thức quan trọng và cần thiết để đảm bảo hiệu quả và thành công của các dự án xây dựng. Vật liệu xây dựng không chỉ đóng vai trò quan trọng trong việc xác định chất lượng và độ bền của công trình, mà còn ảnh hưởng đến khả năng duy trì lịch trình và ngân sách. Vì vậy, việc áp dụng hệ thống quản lý vật liệu xây dựng đúng mực là một yếu tố quyết định để đảm bảo sự thành công của các dự án xây dựng.

Báo cáo này được thực hiện để nắm bắt và trình bày về tầm quan trọng của hệ thống quản lý vật liệu xây dựng. Báo cáo bao gồm việc đề xuất một cái nhìn tổng quan về khái niệm của hệ thống này, từ các thành phần cơ bản cho đến những lợi ích mà nó mang lại trong việc quản lý và thực hiện các dự án xây dựng.

**Chương 1 : Phân Tích và thiết kế hệ thống**

* 1. **Lời giới thiệu**

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang biến đổi toàn diện mọi lĩnh vực, ngành xây dựng không phải là một ngoại lệ. Việc ứng dụng công nghệ thông tin và quản trị dự án hiện đang chuyển đổi ngành xây dựng từ việc thủ công, tốn kém đến sự hiện đại hóa, tối ưu hóa và tăng cường hiệu suất. Trong ngữ cảnh này, hệ thống quản lý vật liệu xây dựng (Construction Material Management System - CMMS) đã trở thành một phần không thể thiếu của quá trình quản lý dự án xây dựng.

Báo cáo này tập trung vào việc giới thiệu và trình bày về hệ thống quản lý vật liệu xây dựng, một công cụ quản lý hiện đại và tiện ích trong việc kiểm soát và quản lý các vật liệu xây dựng. Bằng cách sử dụng công nghệ và quy trình quản lý hiện đại, hệ thống này giúp tổ chức xây dựng tối ưu hóa việc sử dụng vật liệu, giảm thiểu lãng phí, tăng cường quản lý chất lượng, và đảm bảo tính nhất quán của dự án.

Trong phần tiếp theo của báo cáo, chúng tôi sẽ giới thiệu về mục tiêu và lợi ích của hệ thống quản lý vật liệu xây dựng, cùng với các chức năng chính mà nó cung cấp. Chúng tôi cũng sẽ trình bày về một số ví dụ về cách mà hệ thống này có thể được tích hợp vào quy trình quản lý dự án xây dựng, từ việc quản lý nhập/xuất vật liệu đến việc kiểm soát chất lượng và quản lý nhà cung cấp.

Bằng cách tiếp cận hiện đại và sáng tạo thông qua hệ thống quản lý vật liệu xây dựng, chúng ta có thể mở ra những cơ hội mới để tối ưu hóa nguồn lực và đạt được sự thành công bền vững trong các dự án xây dựng. Chúng tôi hy vọng rằng báo cáo này sẽ cung cấp cho bạn một cái nhìn rõ ràng về vai trò và lợi ích của hệ thống quản lý vật liệu xây dựng trong ngành xây dựng ngày nay.

* 1. **Phân tích hệ thống quản lí vật liệu xây dựng**
     1. **Thực trạng của hệ thống quản lý vật liệu xây dựng**

Thực trạng hệ thống quản lý vật liệu xây dựng đang đối mặt với các thách thức về hiệu suất vận hành, điều khiển tồn kho, tích hợp hệ thống, công nghệ và tự động hóa, bền vững và tái chế, phân phối và giao nhận, cùng với tuân thủ quy định pháp lý. Các xu hướng chung bao gồm áp dụng công nghệ, tích hợp thông tin, và tập trung vào bền vững trong quản lý vật liệu xây dựng. Để biết thêm thông tin chi tiết và cập nhật, nên tìm hiểu từ nguồn thông tin và chuyên gia hiện tại trong lĩnh vực xây dựng.

Ưu điểm:

- Tối ưu hóa hiệu suất vận hành và tiết kiệm tài nguyên.

- Tích hợp thông tin, giảm sai sót và tăng hiệu suất làm việc.

- Sử dụng công nghệ và tự động hóa để tăng độ chính xác và hiệu quả.

- Tập trung vào bền vững và tái chế vật liệu.

Nhược điểm:

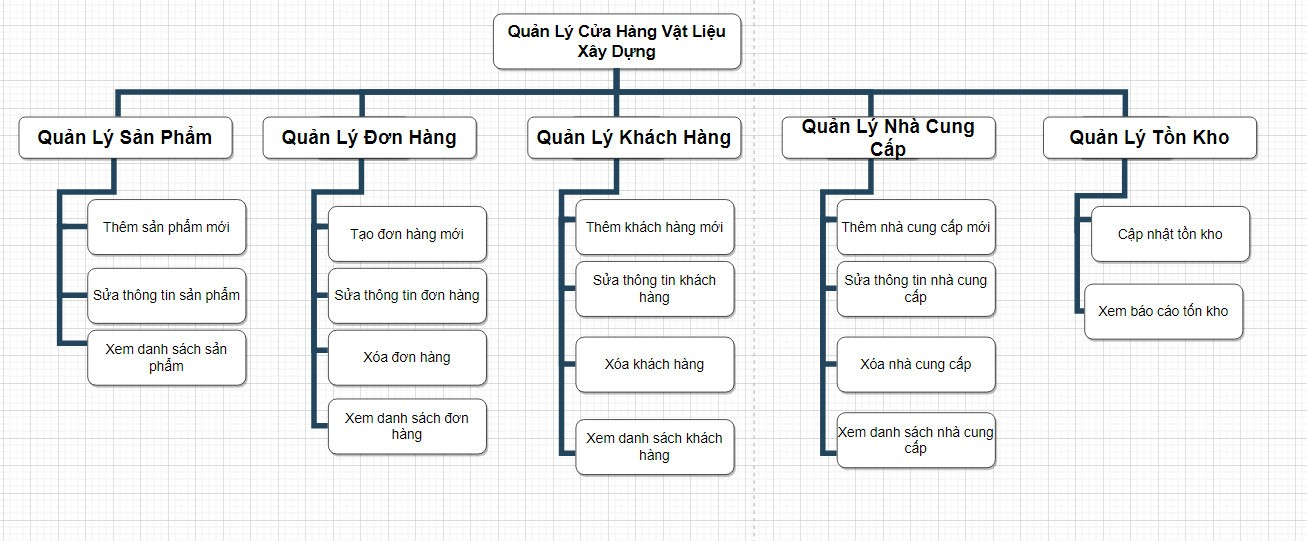
- Đầu tư ban đầu về tài chính, thời gian và nguồn nhân lực.

- Phức tạp trong triển khai và quản lý hàng ngày.

- Yêu cầu thay đổi văn hóa làm việc của tổ chức.

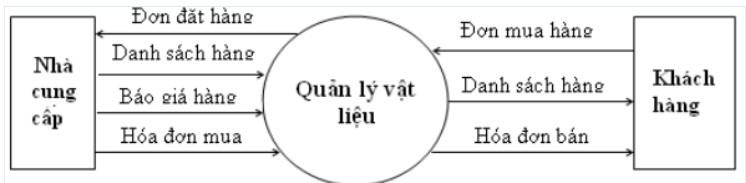
- Có thể phụ thuộc vào công nghệ và cần duy trì thường xuyên.

* + 1. **Biểu đồ phần cấp chức năng**



# *Hình 1. 1 Bản đồ phân cấp chức năng*

* + 1. **Biểu đồ luồng dữ liệu**



# *Hình 1. 2 Biểu đồ luồng dữ liệu*

* + 1. **Phân loại các thuộc tính vào một tập thực thể**
* Kiểu thực thể liên kết là nguồn thông tin cần thiết cho hệ thống ,các kiểu thực thể có thể xuất hiện ở các tác nhân ngoài , các luồng thông tin nội bộ hoặc các kho dữ liệu
* Kiểu thực thể Nhà Cung Cấp gồm: MaSanPham, TenSanPham, Gia, MoTa, SoLuong, MaNhaCungCap
* Kiểu thực thể Nhà Cung Cấp gồm: MaNhaCungCap, TenNhaCungCap, DiaChi, SoDienThoai
* Kiểu thực thể Chi Tiết Hóa Đơn gồm: MaChiTietHoaDon, MaDonHang, MaSanPham, SoLuong, GiaDonVi
* Kiểu thực thể Đơn Hàng gồm: MaDonHang, MaKhachHang, NgayDatHang, TongTien,
* Kiểu thực thể Khách Hàng gồm: MaKhachHang, TenDau, TenCuoi, DiaChi, SoDienThoat, Email
* Kiểu thực thể Thông Tin Thanh Toán gồm: MaThanhToan, MaDonHang, NgayThanhToan, SoTienThanhToan
  + 1. **Xây dựng CSDL trên hệ QTCSDL**

Để xây dựng CSDL hệ thống quản lý điểm sinh viên nhóm chúng em chọn hệ QTCSDL là SQL server

SQL Server là một hệ quản trị cơ sở dữ liệu (Hệ QT CSDL) có nhiều ưu điểm, dưới đây là một số lý do:

1. Hiệu suất và độ tin cậy cao: SQL Server là một hệ QT CSDL được phát triển bởi Microsoft, được thiết kế để cung cấp hiệu suất và độ tin cậy cao. Với SQL Server, có thể xử lý các tác vụ cơ sở dữ liệu lớn và phức tạp một cách nhanh chóng và dễ dàng.
2. Khả năng mở rộng: SQL Server cho phép mở rộng các cơ sở dữ liệu một cách dễ dàng. Có thể thêm các node mới vào hệ thống cơ sở dữ liệu của mình một cách linh hoạt và dễ dàng.
3. Hỗ trợ đa nền tảng: SQL Server là một hệ QT CSDL được hỗ trợ trên nhiều nền tảng, bao gồm Windows, Linux và Docker. Điều này cho phép triển khai các ứng dụng của mình trên nhiều nền tảng một cách dễ dàng.
4. Hỗ trợ quản lý dữ liệu: SQL Server cung cấp nhiều tính năng quản lý dữ liệu mạnh mẽ, bao gồm sao lưu và phục hồi dữ liệu, quản lý phiên, quản lý quyền truy cập và nhiều hơn nữa.
5. Tích hợp với các công cụ Microsoft: SQL Server được tích hợp tốt với các công cụ và dịch vụ khác của Microsoft như Visual Studio, Azure và Power BI, giúp tạo ra các ứng dụng đa nền tảng và dễ dàng tích hợp với các dịch vụ khác của Microsoft.
6. Tóm lại, SQL Server là một lựa chọn tốt cho Hệ QT CSDL vì nó cung cấp hiệu suất và độ tin cậy cao, khả năng mở rộng, hỗ trợ đa nền tảng, tính năng quản lý dữ liệu và tích hợp tốt với các công cụ Microsoft.
   1. **. Phân Quyền**
7. Phân hệ Nhân Viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức Năng** | **Mô tả Hoạt Động** |
| NV1 | Quản lý hàng tồn kho | Nhân viên có quyền xem thông tin hàng tồn kho, cập nhật số lượng tồn kho khi nhập hàng mới và tạo cảnh báo khi sắp hết hàng. |
| NV2 | Quản lý sản phẩm | Nhân viên có quyền thêm, sửa, xóa thông tin sản phẩm, bao gồm tên, mô tả, giá cả và thông tin kỹ thuật. |
| NV3 | Quản lý khách hàng | Nhân viên có quyền xem thông tin khách hàng, cập nhật thông tin cá nhân và lịch sử mua hàng của khách hàng. |
| NV4 | Giao dịch và thanh toán | Nhân viên có quyền ghi nhận giao dịch mua bán, tạo hóa đơn và cập nhật thông tin về phương thức thanh toán |

1. Phân hệ Khách Hàng

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức Năng** | **Mô Tả Hoạt Động** |
| KH1 | Đăng nhập và quản lý tài khoản | Khách hàng có quyền đăng nhập vào tài khoản cá nhân, cập nhật thông tin cá nhân và thay đổi mật khẩu. |
| KH2 | Tra cứu sản phẩm | Khách hàng có quyền tra cứu thông tin về các sản phẩm, bao gồm tên, mô tả, giá cả và thông tin kỹ thuật. |
| KH3 | Xem lịch sử mua hàng | Khách hàng có thể xem lịch sử các giao dịch mua hàng trước đây. |
| KH4 | Thêm sản phẩm vào giỏ hàng | Khách hàng có quyền thêm sản phẩm vào giỏ hàng và tiến hành thanh toán. |

1. Phân hệ Nhà cung cấp

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức Năng** | **Mô Tả Hoạt Động** |
| NCC1 | Xem thông tin sản phẩm | Nhà cung cấp có quyền xem thông tin chi tiết về các sản phẩm mà họ cung cấp, bao gồm tên, mô tả, giá cả và số lượng tồn kho. |
| NCC2 | Đặt hàng | Nhà cung cấp có quyền đặt hàng mới cho các sản phẩm mà họ cung cấp.  Có thể cập nhật số lượng đặt hàng và thông tin về giao hàng dự kiến |
| NCC3 | Theo dõi đơn hàng | Nhà cung cấp có thể theo dõi tình trạng các đơn hàng đã đặt từ họ.  Xem thông tin chi tiết về các đơn hàng, bao gồm sản phẩm, số lượng, ngày đặt hàng và tình trạng giao hàng. |
| NCC4 | Cập nhật thông tin cơ bản | Nhà cung cấp có quyền cập nhật thông tin cơ bản về họ, bao gồm tên, địa chỉ và thông tin liên hệ. |

# **CHƯƠNG 2: ỨNG DỤNG LÝ THUYẾT VÀO HỆ THÔNG QUẢN LÝ VẬT LIỆU XÂY DỰNG**

## 2.1. Xây dựng CSDL trên HQT CSDL

### 2.1.1. Câu lệnh tạo CSDL, tạo bảng, tạo khóa

CREATE DATABASE VLXD

GO

USE VLXD

GO

-- TAO BANG NHA CUNG CAP

CREATE TABLE NhaCungCap (

MaNhaCungCap INT PRIMARY KEY,

TenNhaCungCap NVARCHAR(255),

DiaChi NVARCHAR(255),

SoDienThoai NVARCHAR(20)

);

--TAO BANG KHACH HANG

CREATE TABLE KhachHang (

MaKhachHang INT PRIMARY KEY,

TenDau NVARCHAR(255),

TenCuoi NVARCHAR(255),

DiaChi NVARCHAR(255),

SoDienThoai NVARCHAR(20),

Email NVARCHAR(255)

);

--TAO BANG DON HANG

CREATE TABLE DonHang (

MaDonHang INT PRIMARY KEY,

MaKhachHang INT,

NgayDatHang DATE,

TongTien MONEY,

FOREIGN KEY (MaKhachHang) REFERENCES KhachHang(MaKhachHang)

);

--TAO BANG SAN PHAM

CREATE TABLE SanPham (

MaSanPham INT PRIMARY KEY,

TenSanPham NVARCHAR(255),

MoTa NVARCHAR(MAX),

Gia MONEY,

SoLuongTon INT,

MaNhaCungCap INT,

FOREIGN KEY (MaNhaCungCap) REFERENCES NhaCungCap(MaNhaCungCap)

);

--TAO BANG CHI TIET DON HANG

CREATE TABLE ChiTietDonHang (

MaChiTietDonHang INT PRIMARY KEY,

MaDonHang INT,

MaSanPham INT,

SoLuong INT,

GiaDonVi MONEY,

FOREIGN KEY (MaDonHang) REFERENCES DonHang(MaDonHang),

FOREIGN KEY (MaSanPham) REFERENCES SanPham(MaSanPham)

);

--TAO BANG THONG TIN THANH TOAN

CREATE TABLE ThongTinThanhToan (

MaThanhToan INT PRIMARY KEY,

MaDonHang INT,

NgayThanhToan DATE,

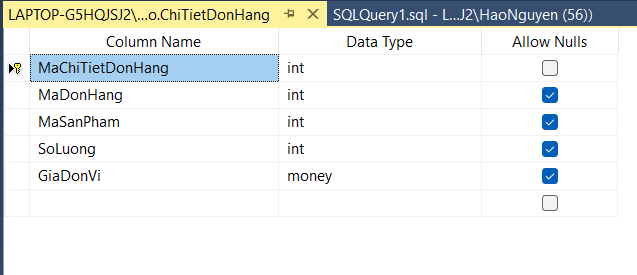
SoTienThanhToan MONEY,

FOREIGN KEY (MaDonHang) REFERENCES DonHang(MaDonHang)

);

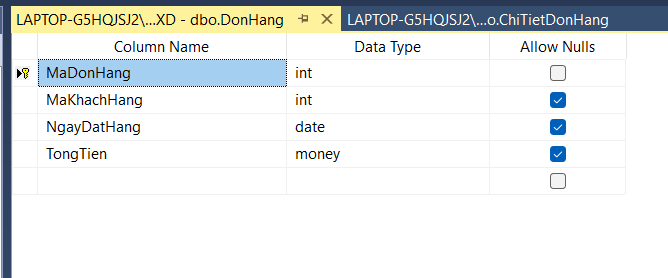
### 2.1.2. Cấu trúc của mỗi bảng

* Bảng ChiTietHoaDon



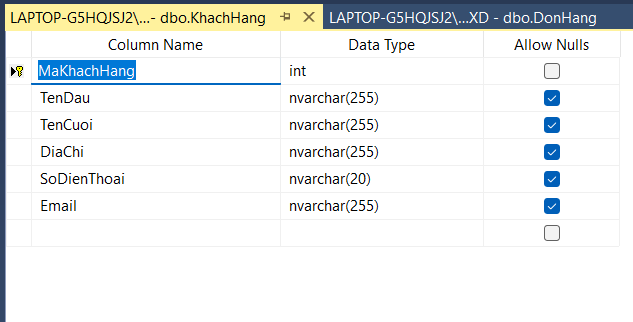
# *Hình 2. 1 Cấu trúc bảng chi tiết hóa đơn*

* Bảng DonHang



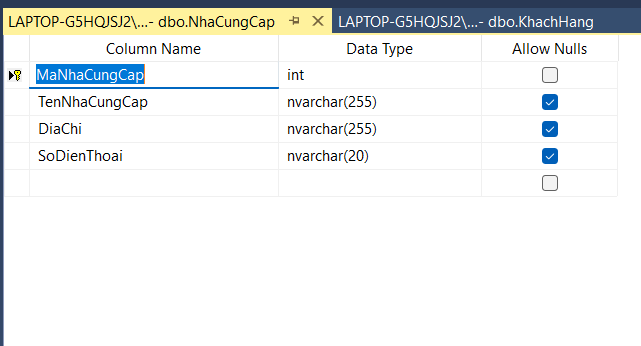
# *Hình 2. 2 Cấu trúc bảng đơn hàng*

* Bảng KhachHang



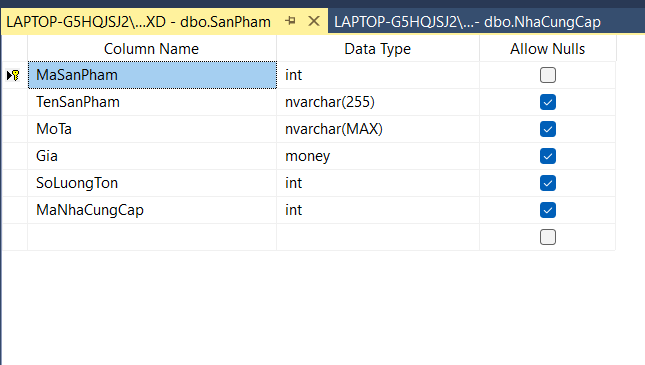
# *Hình 2. 3 Cấu trúc bảng khách hàng*

* Bảng NhaCungCap



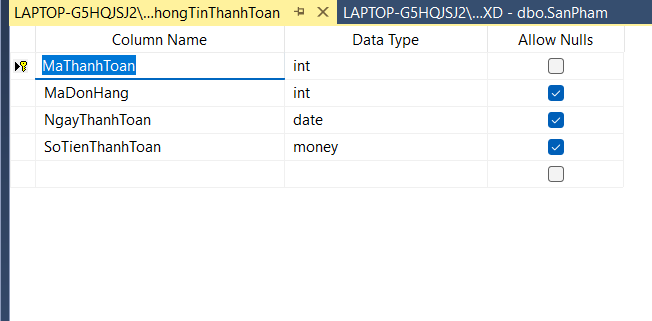
# *Hình 2. 4 Cấu trúc bảng nhà cung cấp*

* Bảng SanPham



# *Hình 2. 5 Cấu trúc bảng sản phẩm*

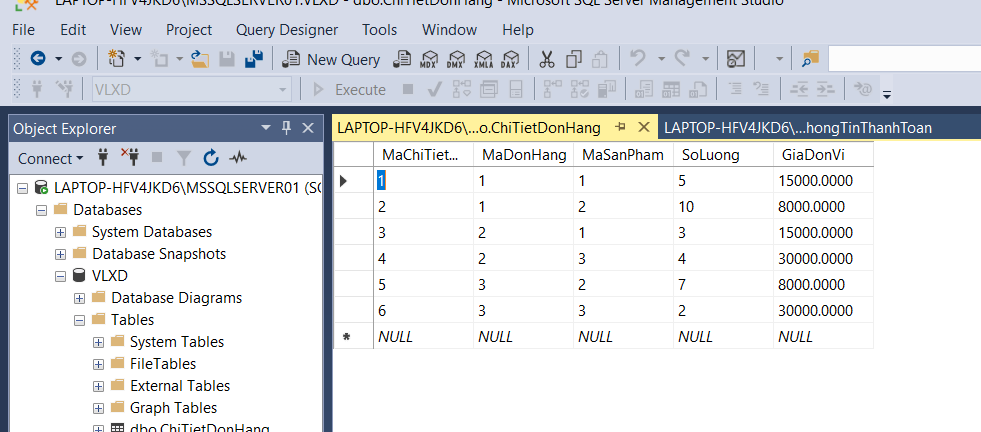
* Bảng ThongTinThanhToan



# *Hình 2. 6 Cấu trúc bảng thông tin thanh toán*

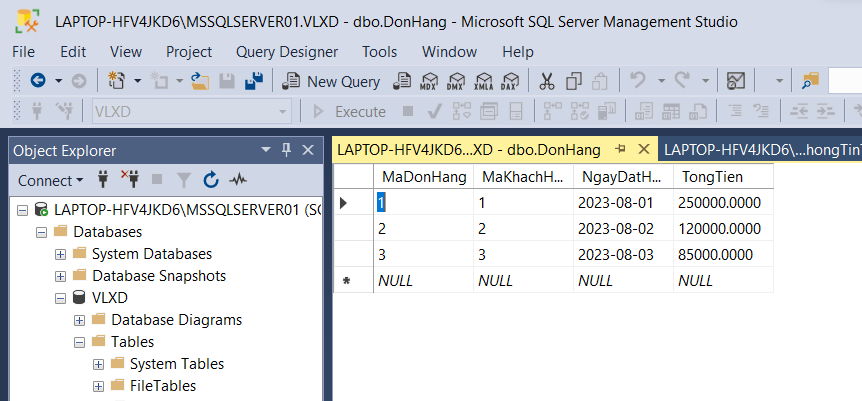
### 2.1.3. Dữ liệu của mỗi bảng

* Dữ liệu bảng ChiTietHoaDon



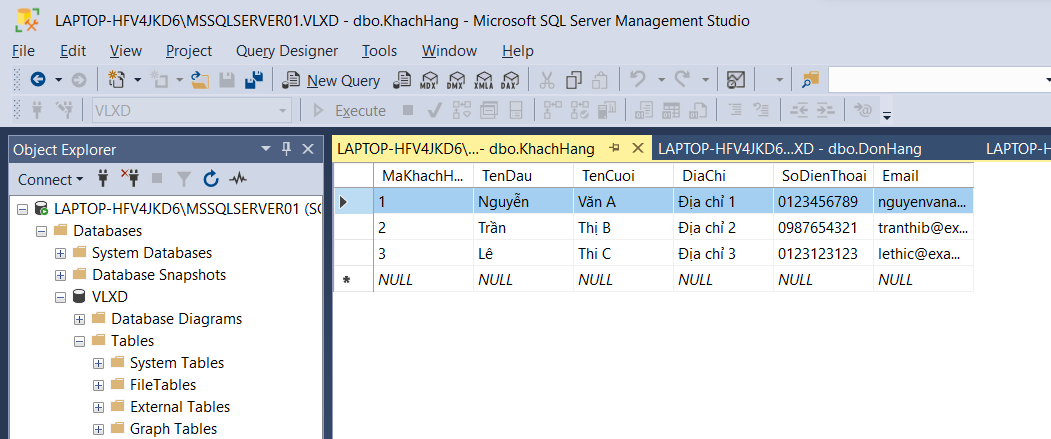
# *Hình 2. 7 Dữ liệu bảng chi tiết hóa đơn*

* Dữ liệu bảng DonHang



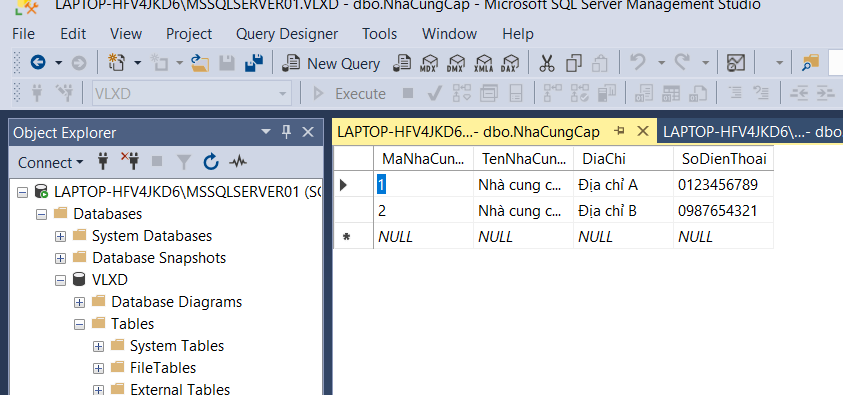
# *Hình 2. 8 Dữ liệu bảng đơn hàng*

* Dữ liệu bảng KhachHang



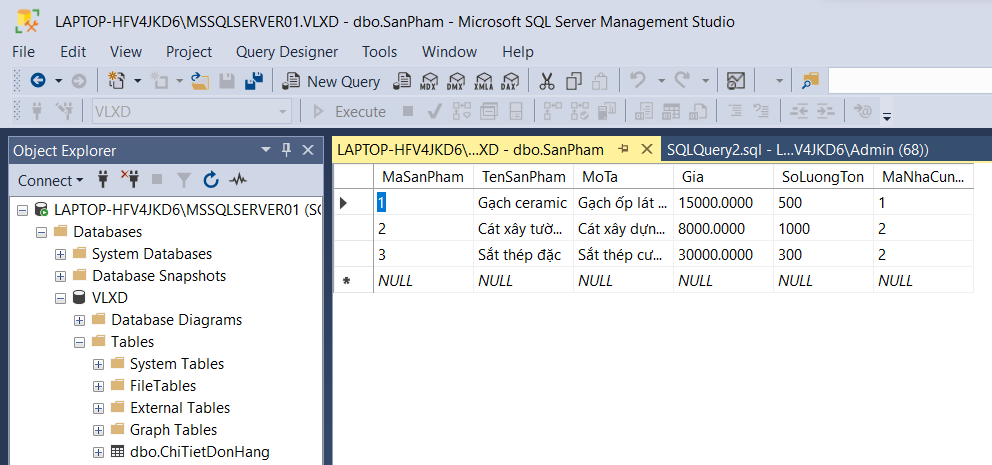
# *Hình 2. 9 Dữ liệu bảng khách hàng*

* Dữ liệu bảng NhaCungCap



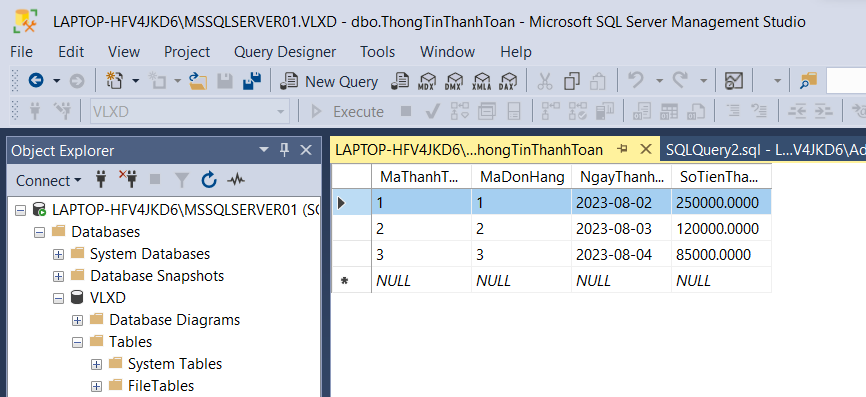
# *Hình 2. 10 Dữ liệu bảng nhà cung cấp*

* Dữ liệu bảng SanPham



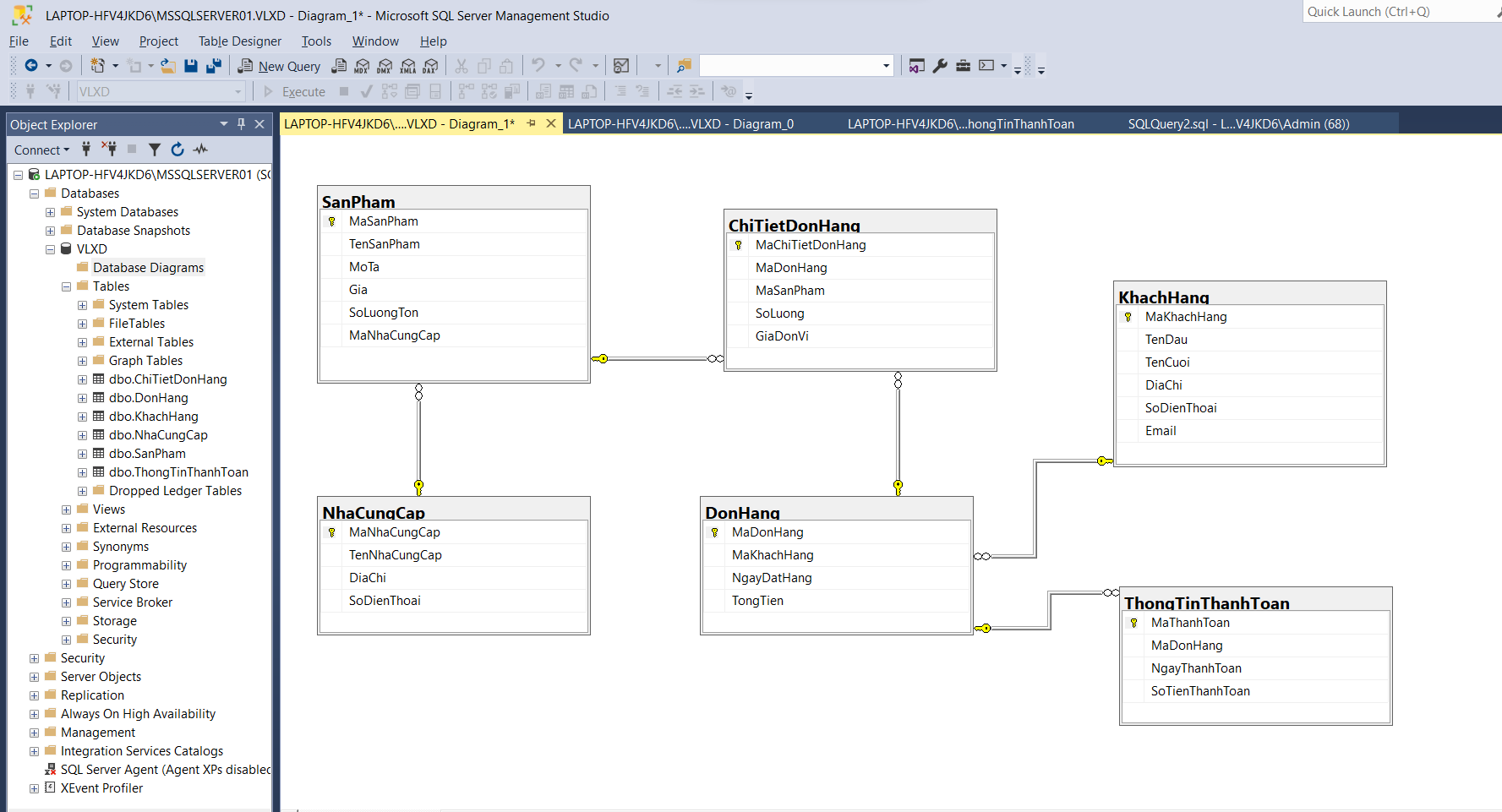
# *Hình 2. 11 Dữ liệu bảng sản phẩm*

* Dữ liệu bảng ThongTinThanhToan



# *Hình 2. 12 Dữ liệu bảng thông tin thanh toán*

### 2.1.4. Diagram liên kết các bảng



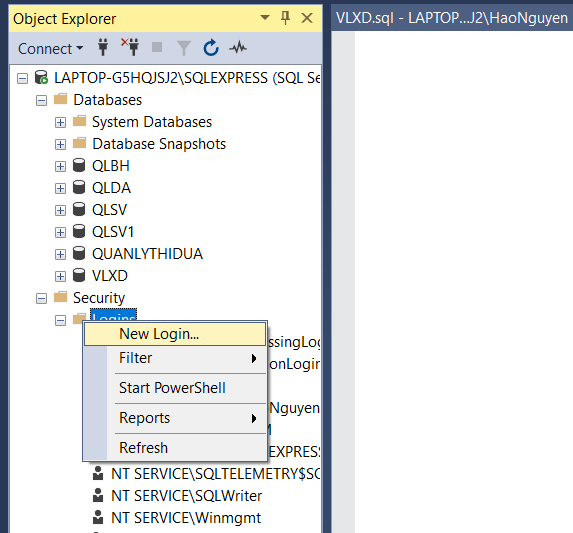
# *Hình 2. 13 Bảng Diagram*

## 2.2. Thao tác trên CSDL trên HQT CSDL

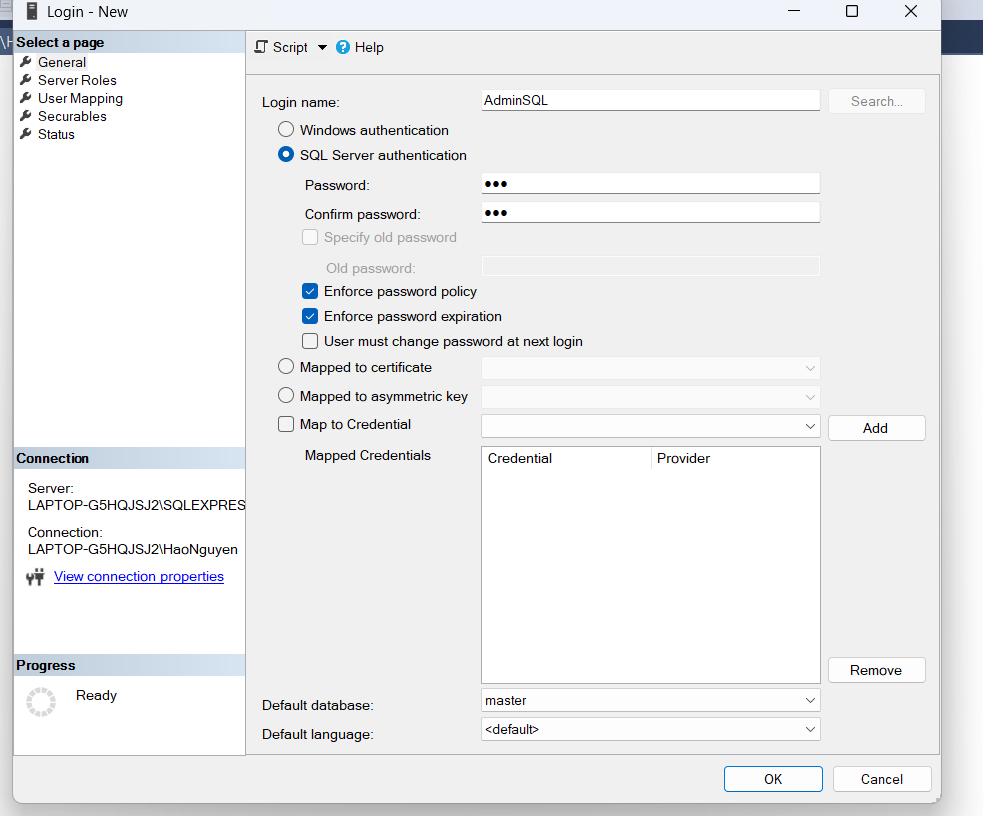
### 2.2.1 Tạo và xóa tài khoản người dùng đăng nhập quyền sql server bằng công cụ và câu lệnh.

Tạo và xóa tài khoản người dùng đăng nhập quyền sql server bằng công cụ:

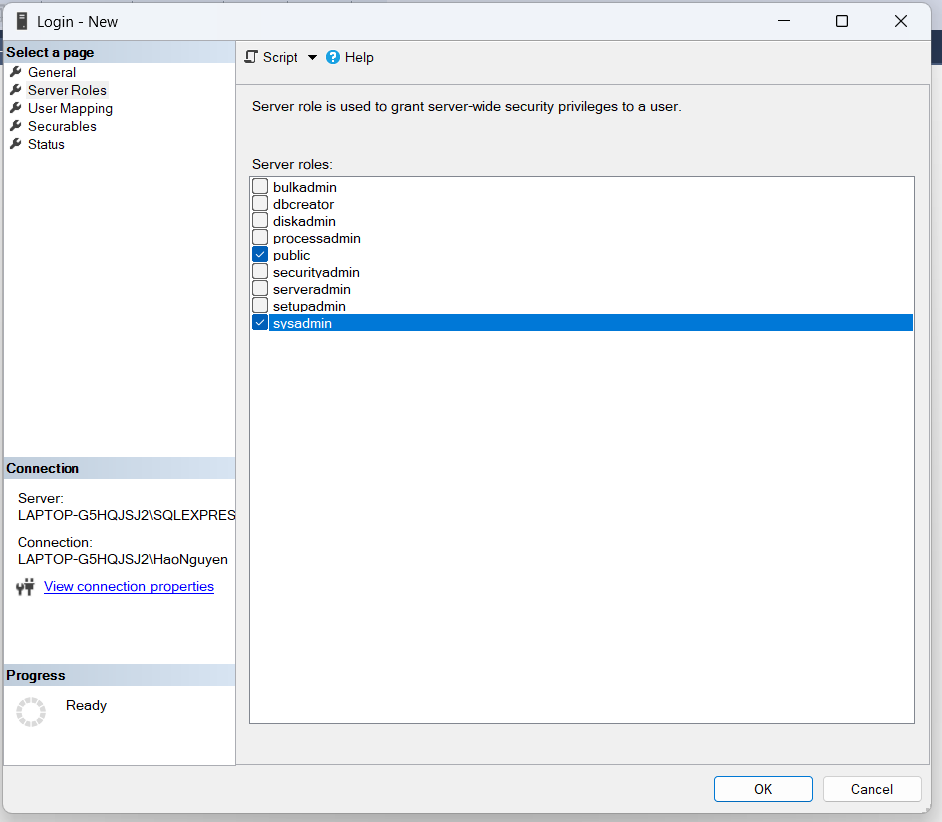
Các bước tạo tài khoản:



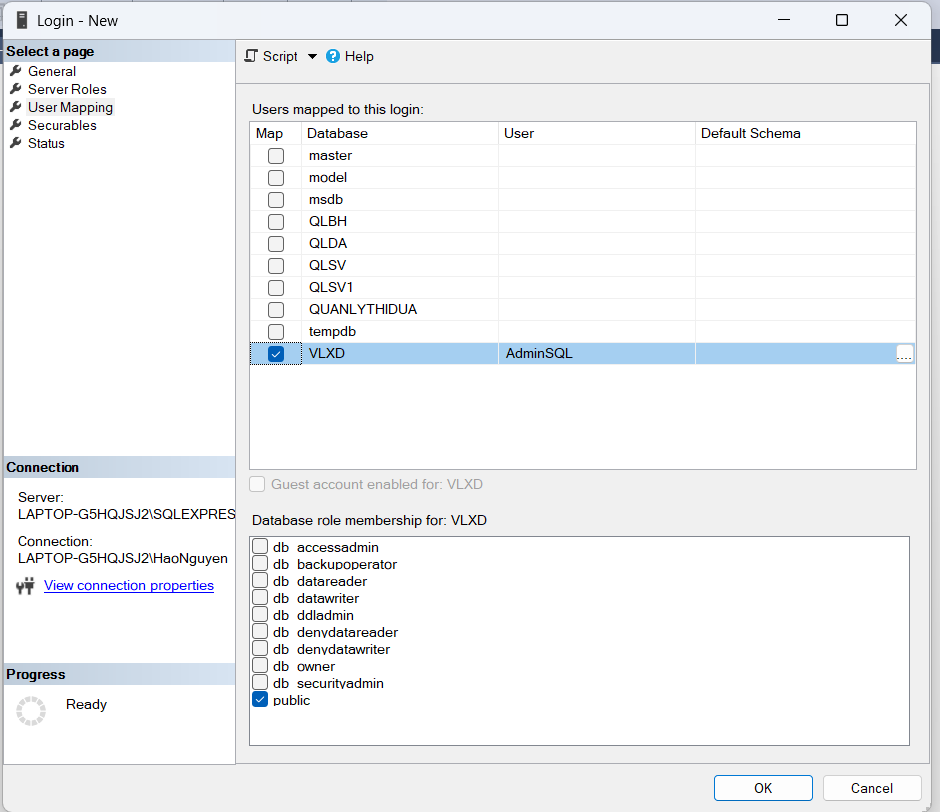
# *Hình 2. 14 Tạo login bước 1*



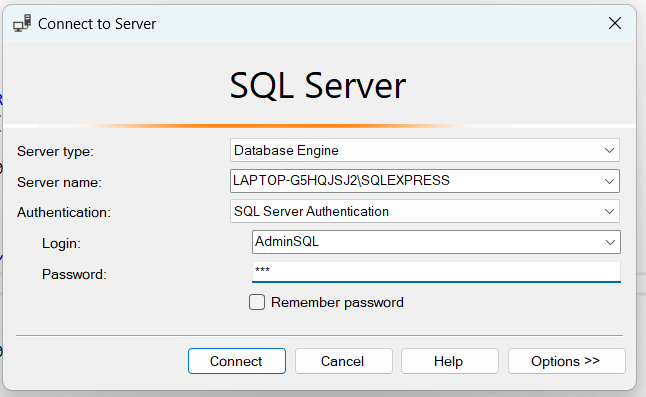
# *Hình 2. 15 Tạo login bước 2*



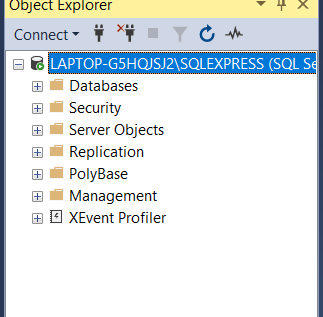
# *Hình 2. 16 Tạo login bước 3*



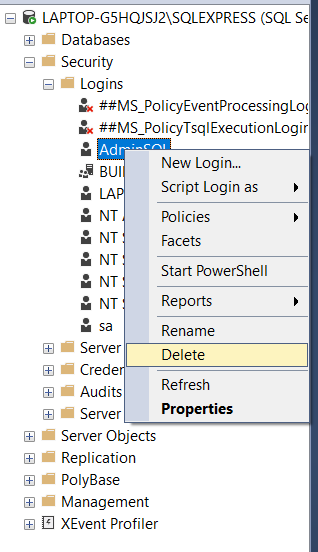
# *Hình 2. 17 Tạo login bước 4*



# *Hình 2. 18 Đăng nhập tài khoản*



*Hình 2. 19 Login thành công*



# *Hình 2. 20 Xóa login*

Tạo và xóa tài khoản người dùng đăng nhập quyền sql server bằng câu lệnh:

Tạo tài khoản:

CREATE LOGIN [AdminVLXD] WITH PASSWORD = '123'

USE [VLXD

CREATE USER [Admin] FOR LOGIN [AdminVLXD]

Xóa tài khoản:

DROP LOGIN AdminVLXD;

### 2.2.2. Viết câu lệnh insert

INSERT INTO NhaCungCap (MaNhaCungCap, TenNhaCungCap, DiaChi, SoDienThoai)

VALUES

(1, N'Nhà cung cấp A', N'Địa chỉ A', N'0123456789'),

(2, N'Nhà cung cấp B', N'Địa chỉ B', N'0987654321');

--Thêm dữ liệu cho bảng KhachHang

INSERT INTO KhachHang (MaKhachHang, TenDau, TenCuoi, DiaChi, SoDienThoai, Email)

VALUES

(1, N'Nguyễn', N'Văn A', N'Địa chỉ 1', N'0123456789', N'nguyenvana@example.com'),

(2, N'Trần', N'Thị B', N'Địa chỉ 2', N'0987654321', N'tranthib@example.com'),

(3, N'Lê', N'Thi C', N'Địa chỉ 3', N'0123123123', N'lethic@example.com');

--Thêm dữ liệu cho bảng DonHang

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongTien)

VALUES

(1, 1, '2023-08-01', 250000),

(2, 2, '2023-08-02', 120000),

(3, 3, '2023-08-03', 85000);

--Thêm dữ liệu cho bảng SanPham

INSERT INTO SanPham (MaSanPham, TenSanPham, MoTa, Gia, SoLuongTon, MaNhaCungCap)

VALUES

(1, N'Gạch ceramic', N'Gạch ốp lát sàn', 15000, 500, 1),

(2, N'Cát xây tường', N'Cát xây dựng tường', 8000, 1000, 2),

(3, N'Sắt thép đặc', N'Sắt thép cường độ cao', 30000, 300, 2);

--Thêm dữ liệu cho bảng ChiTietDonHang

INSERT INTO ChiTietDonHang (MaChiTietDonHang, MaDonHang, MaSanPham, SoLuong, GiaDonVi)

VALUES

(1, 1, 1, 5, 15000),

(2, 1, 2, 10, 8000),

(3, 2, 1, 3, 15000),

(4, 2, 3, 4, 30000),

(5, 3, 2, 7, 8000),

(6, 3, 3, 2, 30000);

--Thêm dữ liệu cho bảng ThongTinThanhToan

INSERT INTO ThongTinThanhToan (MaThanhToan, MaDonHang, NgayThanhToan, SoTienThanhToan)

VALUES

(1, 1, '2023-08-02', 250000),

(2, 2, '2023-08-03', 120000),

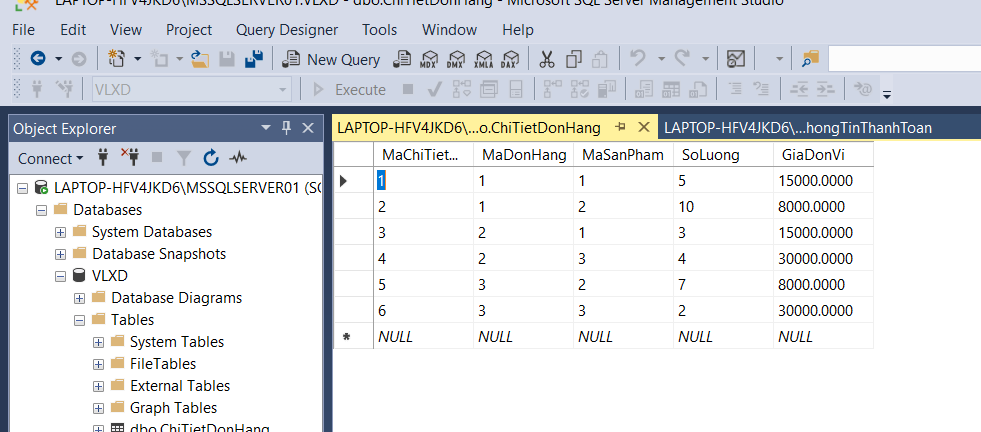
(3, 3, '2023-08-04', 85000);

### 2.2.3. Viết câu lệnh update

UPDATE ChiTietDonHang

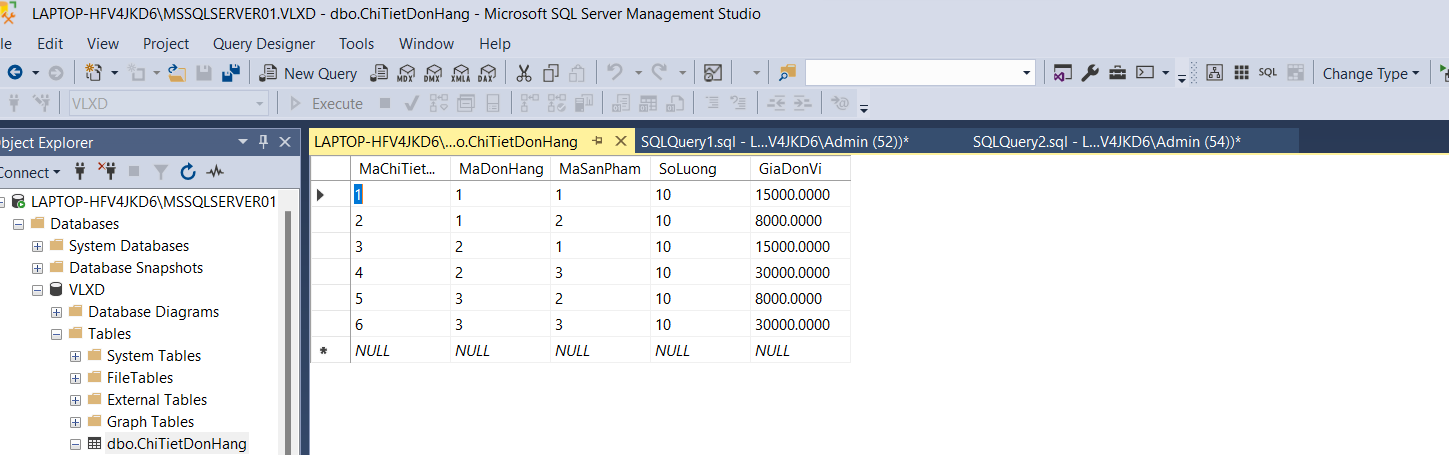
SET SoLuong = '10';

* Trước khi update SoLuong



*Hình 2. 21 Trước khi update số lương*

* Sau khi update SoLuong



# *Hình 2. 22 Sau khi update số lượng*

## 2.3. Store Procedure, Function, Trigger, View

### 2.3.1. Store Procedure

1. **Định nghĩa Store Procedure**

Stored Procedure là một khối mã SQL đã được lưu trữ trong cơ sở dữ liệu để thực thi một loạt các câu lệnh hoặc tác vụ. Stored Procedure giúp tối ưu hóa hiệu suất của cơ sở dữ liệu, giảm thiểu lỗi và đơn giản hóa mã SQL bằng cách chia nhỏ mã thành các phần nhỏ hơn và tái sử dụng chúng trong nhiều vị trí khác nhau. Stored Procedure cũng cung cấp tính năng bảo mật và kiểm soát truy cập vào cơ sở dữ liệu bằng cách cho phép chỉ những người dùng được phép truy cập vào các thủ tục đã được xác định trước.

1. **Cú pháp Store Procedure**

CREATE PROCEDURE procedure\_name

@parameter1 datatype [VARYING] [= default\_value],

@parameter2 datatype [VARYING] [= default\_value],

...

AS

-- Code

GO

Trong đó:

* procedure\_name là tên của Stored Procedure muốn tạo.
* @parameter là danh sách các tham số cho Stored Procedure, có thể có hoặc không. Mỗi tham số được định nghĩa bởi tên, kiểu dữ liệu và giá trị mặc định (nếu có).
* AS xác định bắt đầu của phần thân của Stored Procedure.
* Phần thân của Stored Procedure chứa các câu lệnh SQL để thực hiện tác vụ mong muốn.
* GO dùng để kết thúc định nghĩa Stored Procedure.

Sau khi tạo Stored Procedure, có thể gọi nó bằng cách sử dụng tên Stored Procedure và cung cấp giá trị tham số nếu cần. Ví dụ:

EXEC procedure\_name @parameter1 = value1, @parameter2 = value2, ...

Trong đó value1, value2, ... là giá trị tương ứng cho các tham số của Stored Procedure.

1. **Các loại Store Procedure**

Có hai loại Stored Procedure chính:

1. System Stored Procedure: Là những Stored Procedure được hệ thống quản lý và được lưu trữ trong cơ sở dữ liệu của hệ thống quản trị cơ sở dữ liệu. Những Stored Procedure này được sử dụng để thực hiện các tác vụ hệ thống như quản lý bảo mật, quản lý tập tin, sao lưu dữ liệu và phục hồi dữ liệu.
2. User-defined Stored Procedure: Là những Stored Procedure được định nghĩa bởi người dùng và được lưu trữ trong cơ sở dữ liệu. Những Stored Procedure này được sử dụng để thực hiện các tác vụ do người dùng định nghĩa, ví dụ như tìm kiếm dữ liệu, tính toán, thêm, sửa, xoá dữ liệu.
3. **Ưu điểm và khuyết điểm của từng loại Store Procedure**
4. System Stored Procedure

Ưu điểm:

* Được tối ưu hóa để thực hiện các tác vụ hệ thống nhanh chóng và hiệu quả.
* Có thể được sử dụng để quản lý bảo mật và kiểm soát truy cập vào cơ sở dữ liệu.
* Được tích hợp sẵn trong hệ thống quản trị cơ sở dữ liệu, do đó không cần phải viết lại hoặc triển khai lại từ đầu.

Khuyết điểm:

* Thường không được sử dụng cho các tác vụ do người dùng định nghĩa.
* Không thể thay đổi hoặc chỉnh sửa một System Stored Procedure do nó được quản lý bởi hệ thống quản trị cơ sở dữ liệu.

1. User-defined Stored Procedure

Ưu điểm:

* Cung cấp tính năng tùy biến mạnh mẽ cho người dùng để thực hiện các tác vụ theo yêu cầu.
* Có thể sử dụng để tối ưu hóa hiệu suất của cơ sở dữ liệu và giảm thiểu lỗi.
* Có thể được sử dụng cho các tác vụ phức tạp và lặp đi lặp lại.

Khuyết điểm:

* Cần phải viết và triển khai lại từ đầu.
* Đôi khi cần phải cập nhật lại các Stored Procedure khi có thay đổi trong cấu trúc cơ sở dữ liệu.
* Có thể gây ra sự cố bảo mật và tiềm ẩn các lỗ hổng bảo mật nếu không được thiết kế và triển khai đúng cách.

1. **Code Store procedure trên CSDL quản lý vật liệu xây dựng.**
2. ***C*ode Store procedure trên bảng SANPHAM.**

* **Thêm sản phẩm**

CREATE PROCEDURE sp\_InsertSanPham

@MaSanPham INT,

@TenSanPham NVARCHAR(255),

@MoTa NVARCHAR(MAX),

@Gia MONEY,

@SoLuongTon INT,

@MaNhaCungCap INT

AS

BEGIN

INSERT INTO SanPham (MaSanPham, TenSanPham, MoTa, Gia, SoLuongTon, MaNhaCungCap)

VALUES (@MaSanPham, @TenSanPham, @MoTa, @Gia, @SoLuongTon, @MaNhaCungCap);

END;

EXEC sp\_InsertSanPham

@MaSanPham = 4,

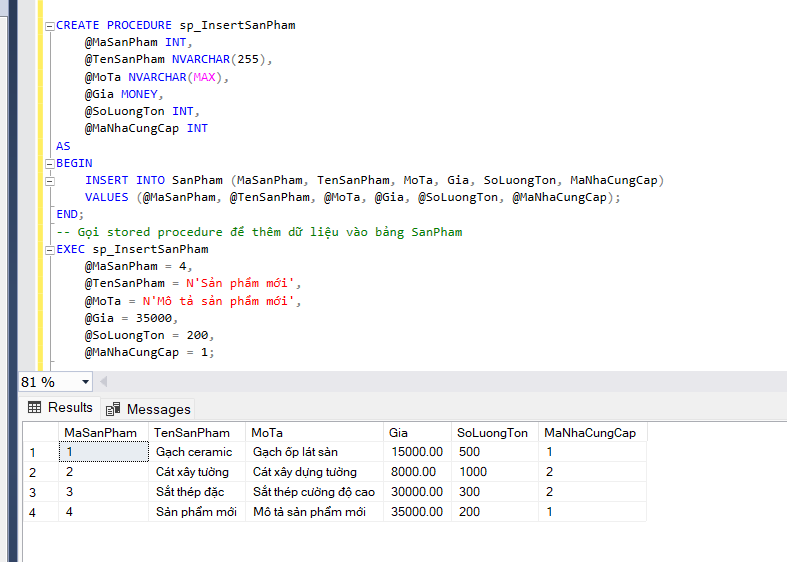
@TenSanPham = N'Sản phẩm mới',

@MoTa = N'Mô tả sản phẩm mới',

@Gia = 35000,

@SoLuongTon = 200,

@MaNhaCungCap = 1;



# *Hình 2. 23 Store procedure thêm sản phẩm*

* Sửa dữ liệu sản phẩm

CREATE PROCEDURE sp\_UpdateSanPham

@MaSanPham INT,

@TenSanPham NVARCHAR(255),

@MoTa NVARCHAR(MAX),

@Gia MONEY,

@SoLuongTon INT,

@MaNhaCungCap INT

AS

BEGIN

UPDATE SanPham

SET TenSanPham = @TenSanPham,

MoTa = @MoTa,

Gia = @Gia,

SoLuongTon = @SoLuongTon,

MaNhaCungCap = @MaNhaCungCap

WHERE MaSanPham = @MaSanPham;

END;

EXEC sp\_UpdateSanPham

@MaSanPham = 4,

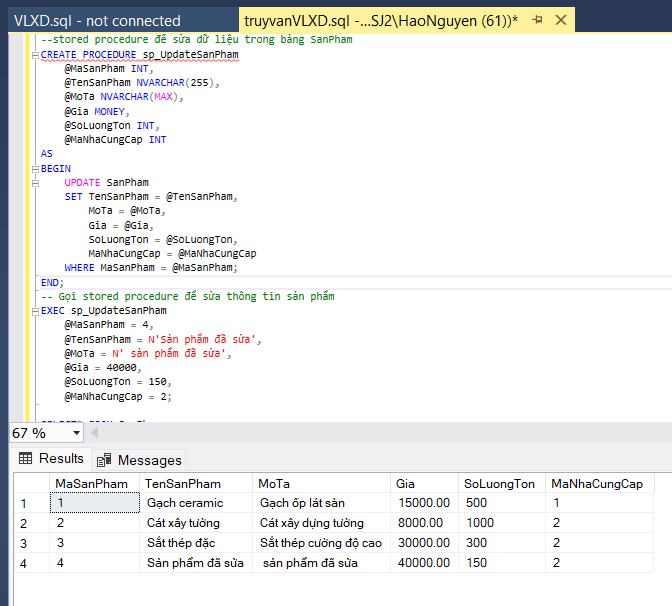
@TenSanPham = N'Sản phẩm đã sửa',

@MoTa = N' sản phẩm đã sửa',

@Gia = 40000,

@SoLuongTon = 150,

@MaNhaCungCap = 2;



# *Hình 2. 24 Store procedure sửa dữ liệu sản phẩm*

* Xóa dữ liệu sản phẩm

CREATE PROCEDURE sp\_DeleteSanPham

@MaSanPham INT

AS

BEGIN

DELETE FROM SanPham

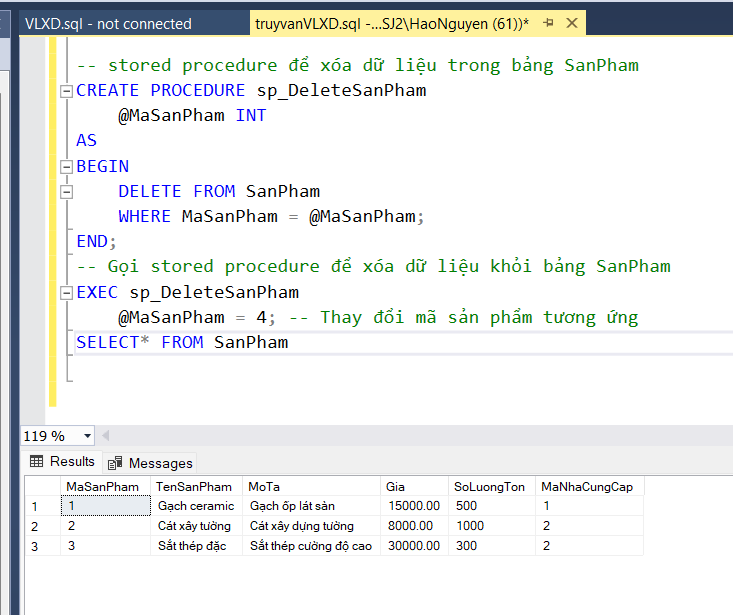
WHERE MaSanPham = @MaSanPham;

END;

EXEC sp\_DeleteSanPham

@MaSanPham = 4;

SELECT\* FROM SanPham



# *Hình 2. 25 Store procedure xóa dữ liệu sản phẩm*

* Hiện thị dữ liệu sản phẩm

CREATE PROCEDURE sp\_GetDanhSachSanPham

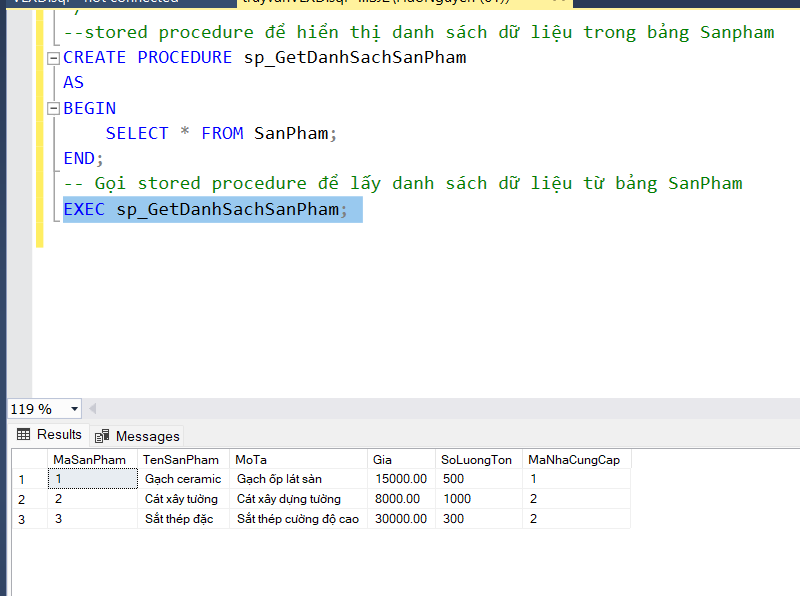
AS

BEGIN

SELECT \* FROM SanPham;

END;

EXEC sp\_GetDanhSachSanPham;



# *Hình 2. 26 Store procedure hiển thị dữ liệu sản phẩm*

* + 1. **Function**

1. Định nghĩa

Trong SQL Server, function (hàm) là một chương trình được định nghĩa trước có thể được sử dụng để thực hiện một tác vụ nào đó. Function cho phép tái sử dụng các đoạn mã thường xuyên được sử dụng trong các truy vấn SQL Server. Nó có thể nhận tham số đầu vào và trả về một giá trị đầu ra dựa trên tham số được cung cấp. Function có thể được sử dụng trong các truy vấn SELECT, WHERE, GROUP BY và ORDER BY.

1. Cú pháp Function

CREATE FUNCTION function\_name

(

@parameter1 data\_type,

@parameter2 data\_type

-- Các tham số đầu vào khác (nếu có)

)

RETURNS return\_data\_type

AS

BEGIN

-- Đoạn mã thực hiện tác vụ của function

-- Sử dụng các tham số đầu vào trong đoạn mã

-- Trả về giá trị đầu ra bằng lệnh RETURN

END

Trong đó:

* function\_name là tên của function.
* @parameter1, @parameter2, ... là các tham số đầu vào của function, mỗi tham số được định nghĩa với tên và kiểu dữ liệu.
* return\_data\_type là kiểu dữ liệu của giá trị trả về của function.
* Đoạn mã thực hiện tác vụ của function được đặt trong khối BEGIN...END.
* Giá trị đầu ra của function được trả về bằng lệnh RETURN.

1. **Các loại Function**

Trong SQL Server, có hai loại chính của function:

1. User-defined function (hàm do người dùng định nghĩa): Là những function được tạo ra bởi người dùng để phục vụ mục đích cụ thể. User-defined function có thể được sử dụng trong các truy vấn SQL Server giống như các hàm có sẵn.

Có 3 loại user-defined function:

* Scalar function: Trả về một giá trị đơn lẻ, chẳng hạn như một số, một chuỗi hoặc một giá trị logic.
* Inline table-valued function: Trả về một bảng kết quả với cấu trúc được xác định trước.
* Multi-statement table-valued function: Trả về một bảng kết quả với cấu trúc được định nghĩa bởi SELECT được sử dụng trong đoạn mã của function.

1. System function (hàm được cung cấp bởi hệ thống): Là những function được cung cấp bởi SQL Server để thực hiện các tác vụ nhất định, chẳng hạn như tính toán và chuyển đổi dữ liệu.

Có nhiều loại system function, bao gồm:

* Aggregate functions: Tính toán giá trị tổng hợp như SUM, AVG, MIN, MAX, COUNT.
* Date and time functions: Là các hàm liên quan đến ngày và thời gian, chẳng hạn như DATEPART, DATENAME, GETDATE.
* String functions: Là các hàm xử lý chuỗi, chẳng hạn như LEN, REPLACE, UPPER, LOWER.
* Mathematical functions: Là các hàm tính toán toán học, chẳng hạn như ABS, POWER, SQRT.
* Logical functions: Là các hàm logic, chẳng hạn như AND, OR, NOT.

Các loại function này đều có thể được sử dụng để giải quyết các vấn đề trong các truy vấn SQL Server.

1. **Ưu điểm và khuyết điểm của từng loại Function**
2. User-defined function

Ưu điểm:

* Cho phép người dùng tạo ra các hàm tùy chỉnh để giải quyết các vấn đề phức tạp trong các truy vấn SQL Server.
* Giúp rút ngắn đoạn mã của các truy vấn, giảm thiểu lỗi và dễ bảo trì hơn.
* Có thể sử dụng trong các truy vấn SELECT, WHERE, GROUP BY và ORDER BY.
* Khuyết điểm:
* Các user-defined function có thể làm giảm hiệu suất của các truy vấn nếu không được sử dụng đúng cách, đặc biệt là các Multi-statement table-valued function.
* Không thể sử dụng trong các truy vấn UPDATE, DELETE và INSERT.

1. System function

Ưu điểm:

* Được cung cấp sẵn bởi SQL Server, giúp người dùng thực hiện các tác vụ nhất định một cách nhanh chóng và dễ dàng.
* Có nhiều loại, đa dạng và phong phú, đáp ứng nhu cầu sử dụng của người dùng.
* Khuyết điểm:
* Không thể tùy biến hoặc mở rộng, do đó không thể giải quyết được tất cả các vấn đề.
* Có thể làm giảm hiệu suất của các truy vấn nếu sử dụng nhiều hàm đồng thời hoặc sử dụng không đúng cách.

1. **Code Function trên CSDL quản lý vật liệu xây dựng**

* Function để tính tổng giá trị đơn hàng dựa trên mã đơn hàng

CREATE FUNCTION dbo.CalculateTotalPrice

(

@MaDonHang INT

)

RETURNS MONEY

AS

BEGIN

DECLARE @Total MONEY;

SELECT @Total = SUM(ctdh.SoLuong \* sp.Gia)

FROM ChiTietDonHang ctdh

JOIN SanPham sp ON ctdh.MaSanPham = sp.MaSanPham

WHERE ctdh.MaDonHang = @MaDonHang;

RETURN @Total;

END;

--Sau khi đã tạo function như trên, có thể sử dụng nó để tính tổng giá trị đơn hàng

-- Sử dụng function để tính tổng giá trị đơn hàng

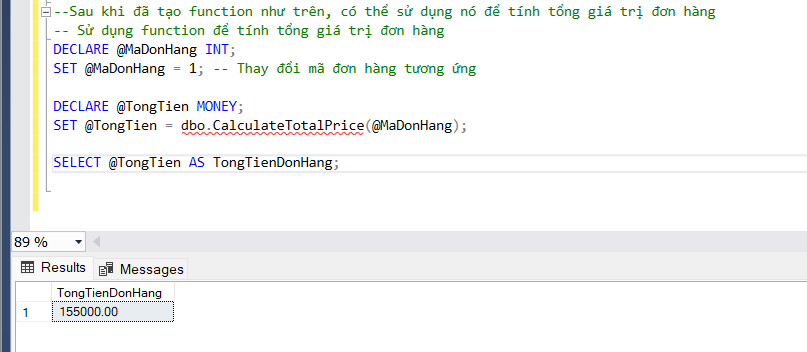
DECLARE @MaDonHang INT;

SET @MaDonHang = 1; -- Thay đổi mã đơn hàng tương ứng

DECLARE @TongTien MONEY;

SET @TongTien = dbo.CalculateTotalPrice(@MaDonHang);

SELECT @TongTien AS TongTienDonHang;



# *Hình 2. 27 Function tổng tiền đơn hàng*

* Tổng số lượng sản phẩm trong một đơn hàng dựa trên mã đơn hàng

CREATE FUNCTION dbo.CalculateTotalQuantity

(

@MaDonHang INT

)

RETURNS INT

AS

BEGIN

DECLARE @TotalQuantity INT;

SELECT @TotalQuantity = SUM(SoLuong)

FROM ChiTietDonHang

WHERE MaDonHang = @MaDonHang;

RETURN @TotalQuantity;

END;

Sau khi đã tạo function như trên, có thể sử dụng nó để tính tổng số lượng sản phẩm trong một đơn hàng.

-- Sử dụng function để tính tổng số lượng sản phẩm trong đơn hàng

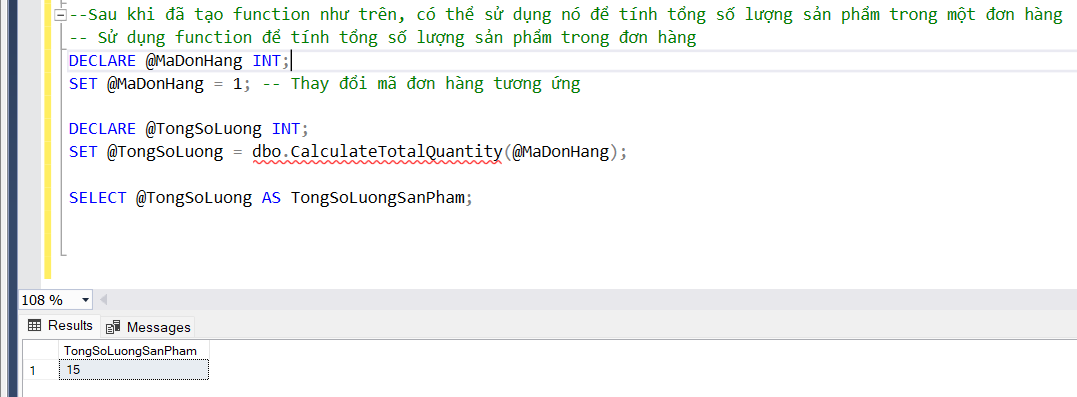
DECLARE @MaDonHang INT;

SET @MaDonHang = 1; -- Thay đổi mã đơn hàng tương ứng

DECLARE @TongSoLuong INT;

SET @TongSoLuong = dbo.CalculateTotalQuantity(@MaDonHang);

SELECT @TongSoLuong AS TongSoLuongSanPham;



# *Hình 2. 28 Function tổng lượng sản phẩm*

* Lấy thông tin chi tiết đơn hàng

CREATE FUNCTION dbo.LayChiTietDonHang (@MaDonHang INT)

RETURNS TABLE

AS

RETURN

(

SELECT ctdh.MaChiTietDonHang, sp.TenSanPham, ctdh.SoLuong, ctdh.GiaDonVi

FROM ChiTietDonHang ctdh

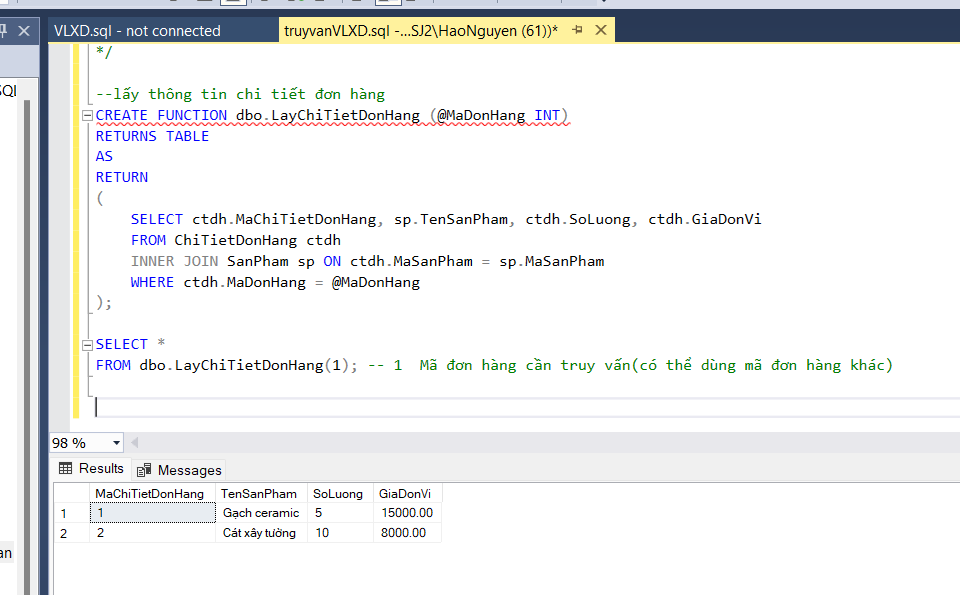
INNER JOIN SanPham sp ON ctdh.MaSanPham = sp.MaSanPham

WHERE ctdh.MaDonHang = @MaDonHang

);

SELECT \*

FROM dbo.LayChiTietDonHang(1); -- 1 Mã đơn hàng cần truy vấn(có thể dùng mã đơn hàng khác)



# *Hình 2. 29 Function lấy thông tin chi tiết đơn hàng*

Tổng số đơn hàng của một khách hàng

CREATE FUNCTION dbo.TinhTongDonHangCuaKhachHang (@MaKhachHang INT)

RETURNS INT

AS

BEGIN

DECLARE @TongDonHang INT;

SELECT @TongDonHang = COUNT(\*)

FROM DonHang

WHERE MaKhachHang = @MaKhachHang;

RETURN @TongDonHang;

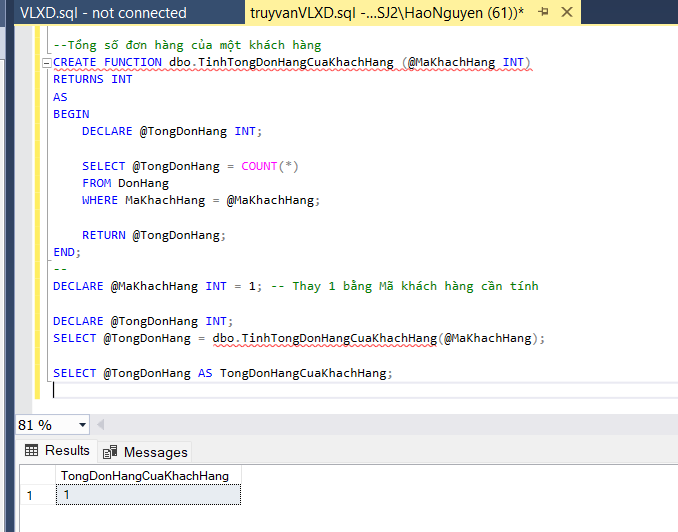
END;

DECLARE @MaKhachHang INT = 1; -- Thay 1 bằng Mã khách hàng cần tính

DECLARE @TongDonHang INT;

SELECT @TongDonHang = dbo.TinhTongDonHangCuaKhachHang(@MaKhachHang);

SELECT @TongDonHang AS TongDonHangCuaKhachHang;



# *Hình 2. 30 Function tổng đơn hàng của khách hàng*

### 2.3.3. Trigger

**a. Định nghĩa Trigger**

Trigger trong SQL Server là một đối tượng cơ sở dữ liệu được sử dụng để tự động thực hiện các hành động hoặc xử lý dữ liệu trong khi thực hiện các câu lệnh INSERT, UPDATE hoặc DELETE trên các bảng của cơ sở dữ liệu. Trigger có thể được sử dụng để kiểm soát hoặc giám sát dữ liệu, thực hiện các tác vụ phức tạp và đảm bảo tính toàn vẹn của dữ liệu trong cơ sở dữ liệu. Khi các sự kiện được kích hoạt, trigger sẽ tự động thực hiện các hành động được định nghĩa trước đó, như thêm, sửa hoặc xóa dữ liệu từ các bảng khác trong cơ sở dữ liệu.

**b. Cú pháp Trigger**

CREATE TRIGGER [Tên Trigger]

ON [Tên Bảng]

[FOR/AFTER/INSTEAD OF] [INSERT/UPDATE/DELETE]

AS

BEGIN

-- Các câu lệnh T-SQL để thực hiện trong Trigger

END;

Trong đó:

* [Tên Trigger] là tên của Trigger.
* [Tên Bảng] là tên của bảng mà Trigger được kích hoạt.
* [FOR/AFTER/INSTEAD OF] chỉ định thời điểm khi Trigger được kích hoạt: FOR sẽ được kích hoạt trước khi thực hiện câu lệnh INSERT/UPDATE/DELETE, AFTER sẽ được kích hoạt sau khi thực hiện câu lệnh INSERT/UPDATE/DELETE, INSTEAD OF sẽ thay thế câu lệnh INSERT/UPDATE/DELETE gốc bằng các câu lệnh trong Trigger.
* [INSERT/UPDATE/DELETE] chỉ định loại câu lệnh DML (Data Manipulation Language) kích hoạt Trigger.
* AS chỉ định phần thân của Trigger.
* BEGIN...END là khối lệnh T-SQL để thực hiện trong Trigger. Trong đó, có thể sử dụng các biến, các truy vấn SQL và các lệnh điều kiện để xử lý dữ liệu tùy ý.

**c. Các loại Trigger**

Trong SQL Server, có 2 loại Trigger chính, bao gồm:

1. Trigger After (After Trigger): Được kích hoạt sau khi thực hiện một câu lệnh DML (INSERT, UPDATE hoặc DELETE) trên bảng mà Trigger được đính kèm. Trigger này thường được sử dụng để kiểm tra và thay đổi dữ liệu sau khi nó đã được thêm mới, sửa đổi hoặc xóa bỏ. Ví dụ: Trigger After được sử dụng để cập nhật trường LastModifiedDate khi một bản ghi được thêm mới hoặc sửa đổi.

2. Trigger Instead Of (Instead Of Trigger): Thay thế câu lệnh DML gốc bằng các câu lệnh trong Trigger. Trigger này thường được sử dụng để xử lý các tác vụ phức tạp hoặc tùy chỉnh khi thực hiện câu lệnh DML. Ví dụ: Trigger Instead Of được sử dụng để thêm mới bản ghi vào nhiều bảng liên quan thay vì chỉ thêm vào bảng gốc.

Cả hai loại Trigger này đều có thể được sử dụng để kiểm tra tính toàn vẹn của dữ liệu, đồng bộ hóa dữ liệu giữa các bảng, thực hiện các tác vụ phức tạp và xử lý dữ liệu sau khi thực hiện câu lệnh DML.

**d. Ưu điểm và khuyết điểm của từng loại Trigger**

1. Trigger After:

Ưu điểm:

* Thích hợp cho các tác vụ kiểm tra, cập nhật, tính toán hoặc đồng bộ hóa dữ liệu sau khi câu lệnh DML đã hoàn thành.
* Cung cấp các giá trị mới hoặc cũ của các trường trong bảng mà Trigger được kích hoạt, giúp người lập trình xử lý dữ liệu một cách chính xác.
* Dễ dàng triển khai và quản lý.
* Khuyết điểm:
* Có thể làm chậm quá trình thực thi khi dữ liệu trong bảng rất lớn.
* Không thể thay đổi hoặc hủy bỏ một phần thực thi của câu lệnh DML sau khi đã kích hoạt Trigger.

2. Trigger Instead Of:

Ưu điểm:

* Cho phép người lập trình xử lý các tác vụ phức tạp hoặc tùy chỉnh khi thực hiện câu lệnh DML.
* Có thể thay đổi hoặc hủy bỏ một phần thực thi của câu lệnh DML sau khi đã kích hoạt Trigger.
* Dễ dàng triển khai và quản lý.
* Khuyết điểm:
* Phải đảm bảo rằng các câu lệnh trong Trigger thay thế hoàn toàn câu lệnh DML gốc, nếu không sẽ dẫn đến dữ liệu bị lỗi hoặc không đồng bộ.
* Không cung cấp giá trị mới hoặc cũ của các trường trong bảng mà Trigger được kích hoạt, giúp người lập trình xử lý dữ liệu một cách chính xác.
* Tùy thuộc vào yêu cầu và tình huống sử dụng, người lập trình có thể lựa chọn loại Trigger phù hợp để xử lý dữ liệu một cách chính xác và hiệu quả.

e.**Code Trigger trên CSDL quản lý vật liệu xây dựng**

* Thêm dữ liệu đơn hàng để nhập tổng tiền

CREATE TRIGGER UpdateTongTienAfterInsert

ON DonHang

AFTER INSERT

AS

BEGIN

DECLARE @MaDonHang INT;

SELECT @MaDonHang = inserted.MaDonHang FROM inserted;

UPDATE DonHang

SET TongTien = (SELECT SUM(GiaDonVi \* SoLuong) FROM ChiTietDonHang WHERE MaDonHang = @MaDonHang)

WHERE MaDonHang = @MaDonHang;

END;

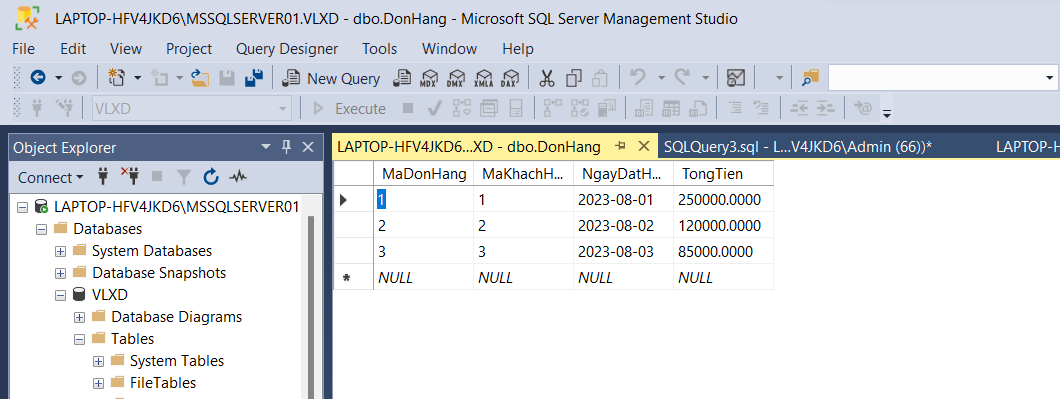
-- Thêm dữ liệu vào bảng DonHang với mã đơn hàng là 3

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang)

VALUES (3, 1, GETDATE());

-- Hiển thị thông tin trong bảng DonHang cho mã đơn hàng là 3

SELECT \* FROM DonHang WHERE MaDonHang = 3;



# *Hình 2. 31 Trigger thêm dữ liệu đơn hàng nhập tổng tiền*

* + 1. **View**

1. **Định nghĩa**

Trigger view trong SQL Server là một loại trigger được kích hoạt (activated) khi một bảng (table) được cập nhật, thêm mới (insert), hoặc xóa (delete), nhưng chỉ nếu dữ liệu thay đổi có ảnh hưởng đến một view nào đó trong cơ sở dữ liệu. Trigger view được sử dụng để thực thi các tác vụ tự động khi dữ liệu trong một view bị thay đổi.

Khi một trigger view được kích hoạt, SQL Server sẽ thực thi mã của trigger này, thực hiện các hành động cần thiết để đảm bảo rằng dữ liệu trong view phù hợp với các điều kiện được thiết lập trong trigger. Trigger view thường được sử dụng để đảm bảo tính toàn vẹn dữ liệu trong các view phức tạp hoặc khi dữ liệu trong các view phụ thuộc vào dữ liệu của các bảng khác.

1. **Cú pháp của view**

Cú pháp tạo view trong SQL Server như sau:

CREATE VIEW view\_name AS

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

Trong đó:

* view\_name là tên của view muốn tạo.
* column1, column2, ... là danh sách các cột muốn lấy từ bảng (table) để tạo view.
* table\_name là tên của bảng muốn lấy dữ liệu để tạo view.
* condition là điều kiện để lọc các bản ghi trong bảng (table) trước khi tạo view.

Sau khi tạo view thành công, có thể truy vấn view này bằng cách sử dụng câu lệnh SELECT giống như khi truy vấn bảng (table) bình thường. Với mỗi lần truy vấn view, hệ thống sẽ tự động thực hiện truy vấn lại bảng (table) và trả về kết quả theo cấu trúc đã được định nghĩa trong view.

1. **Các loại view**

Trong SQL Server, có 2 loại view chính là:

1. View cơ bản (Simple view): Đây là loại view đơn giản nhất, chỉ cho phép lấy dữ liệu từ một bảng (table) duy nhất. View cơ bản không cho phép sử dụng các hàm tính toán hoặc kết hợp dữ liệu từ nhiều bảng khác nhau.
2. View có thể cập nhật (Updatable view): Loại view này cho phép cập nhật dữ liệu trực tiếp vào view, và hệ thống sẽ tự động cập nhật dữ liệu vào các bảng (table) liên quan. View có thể cập nhật được tạo ra bằng cách sử dụng các câu lệnh INSERT, UPDATE, và DELETE trực tiếp trên view.
3. Ưu điểm và khuyết điểm của view

Cả hai loại view chính trong SQL Server (Simple view và Updatable view) đều có những ưu điểm và khuyết điểm riêng.

1. **View cơ bản (Simple view):**

Ưu điểm:

* Dễ dàng và nhanh chóng để tạo và sử dụng.
* Giúp che giấu thông tin bảng gốc và bảo vệ tính toàn vẹn dữ liệu của bảng.

Khuyết điểm:

* Chỉ cho phép truy vấn dữ liệu từ một bảng duy nhất.
* Không cho phép sử dụng các hàm tính toán hoặc kết hợp dữ liệu từ nhiều bảng khác nhau.

1. **View có thể cập nhật (Updatable view):**

Ưu điểm:

* Cho phép cập nhật dữ liệu trực tiếp vào view và tự động cập nhật dữ liệu vào các bảng liên quan.
* Giúp đơn giản hóa quy trình cập nhật dữ liệu và giảm thiểu các lỗi nhầm lẫn.

Khuyết điểm:

* Có thể gây ra hiệu suất chậm nếu view có liên quan đến nhiều bảng hoặc bao gồm các phép tính toán phức tạp.
* Không thể cập nhật được nếu view chứa các phép tính toán hoặc các câu lệnh GROUP BY, HAVING, UNION, hoặc DISTINCT.

1. Code view trên CSDL quản lý vật liệu xây dựng

* Tạo view để hiển thị thông tin chi tiết đơn hàng

CREATE VIEW dbo.ViewChiTietDonHang AS

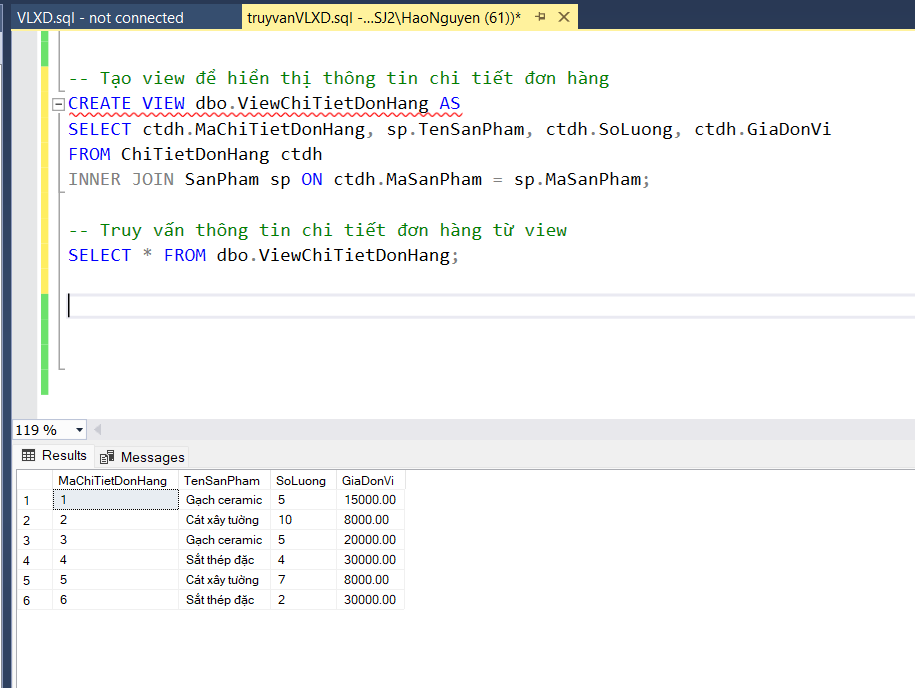
SELECT ctdh.MaChiTietDonHang, sp.TenSanPham, ctdh.SoLuong, ctdh.GiaDonVi

FROM ChiTietDonHang ctdh

INNER JOIN SanPham sp ON ctdh.MaSanPham = sp.MaSanPham;

-- Truy vấn thông tin chi tiết đơn hàng từ view

SELECT \* FROM dbo.ViewChiTietDonHang;



# *Hình 2. 32 View hiển thị thông tin chi tiết hóa đơn*

* View danh sách sản phẩm và số lượng tồn

CREATE VIEW dbo.ViewDanhSachSanPham

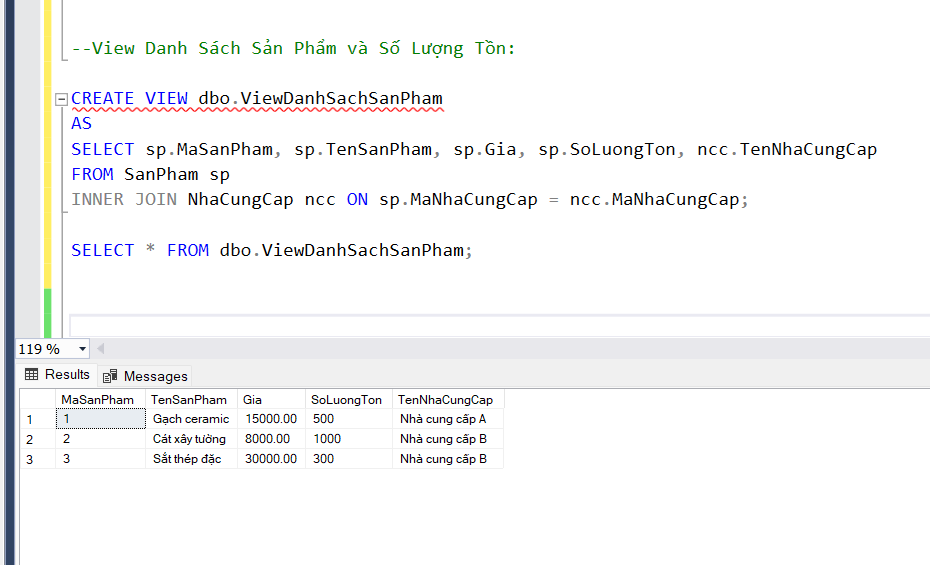
AS

SELECT sp.MaSanPham, sp.TenSanPham, sp.Gia, sp.SoLuongTon, ncc.TenNhaCungCap

FROM SanPham sp

INNER JOIN NhaCungCap ncc ON sp.MaNhaCungCap = ncc.MaNhaCungCap;

SELECT \* FROM dbo.ViewDanhSachSanPham;



# *Hình 2. 33 View danh sách sản phẩm và số lượng tồn*

* View danh đơn hàng của khách hàng

View Danh Sách Đơn Hàng của Khách Hàng

CREATE VIEW dbo.ViewDanhSachDonHangKhachHang

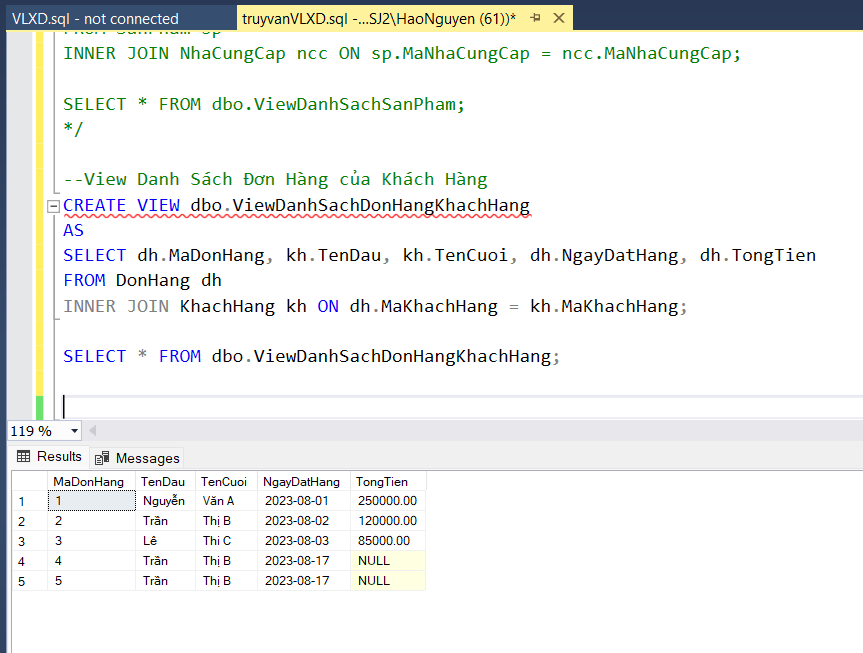
AS

SELECT dh.MaDonHang, kh.TenDau, kh.TenCuoi, dh.NgayDatHang, dh.TongTien

FROM DonHang dh

INNER JOIN KhachHang kh ON dh.MaKhachHang = kh.MaKhachHang;

SELECT \* FROM dbo.ViewDanhSachDonHangKhachHang;



# *Hình 2. 34 View danh sách đơn hàng của khách hàng*

* View danh sách đơn hàng và tổng tiền

CREATE VIEW dbo.ViewDanhSachDonHangVaTongTien

AS

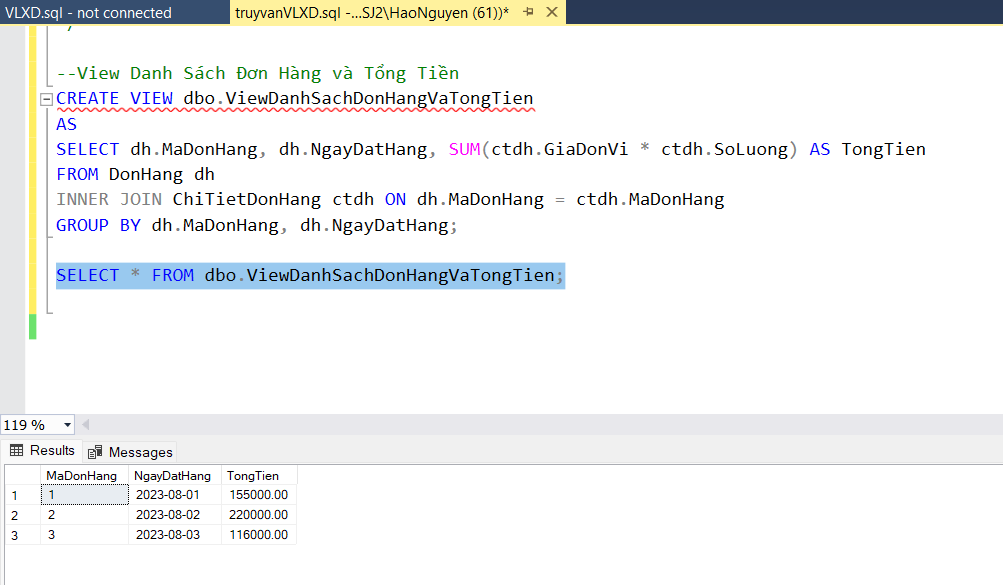
SELECT dh.MaDonHang, dh.NgayDatHang, SUM(ctdh.GiaDonVi \* ctdh.SoLuong) AS TongTien

FROM DonHang dh

INNER JOIN ChiTietDonHang ctdh ON dh.MaDonHang = ctdh.MaDonHang

GROUP BY dh.MaDonHang, dh.NgayDatHang;

SELECT \* FROM dbo.ViewDanhSachDonHangVaTongTien;



# *Hình 2. 35 View danh sách đơn hàng của khách hàng*

* View danh sách nhà cung cấp và số lương sản phẩm

CREATE VIEW dbo.ViewDanhSachNhaCungCapVaSoLuongSanPham

AS

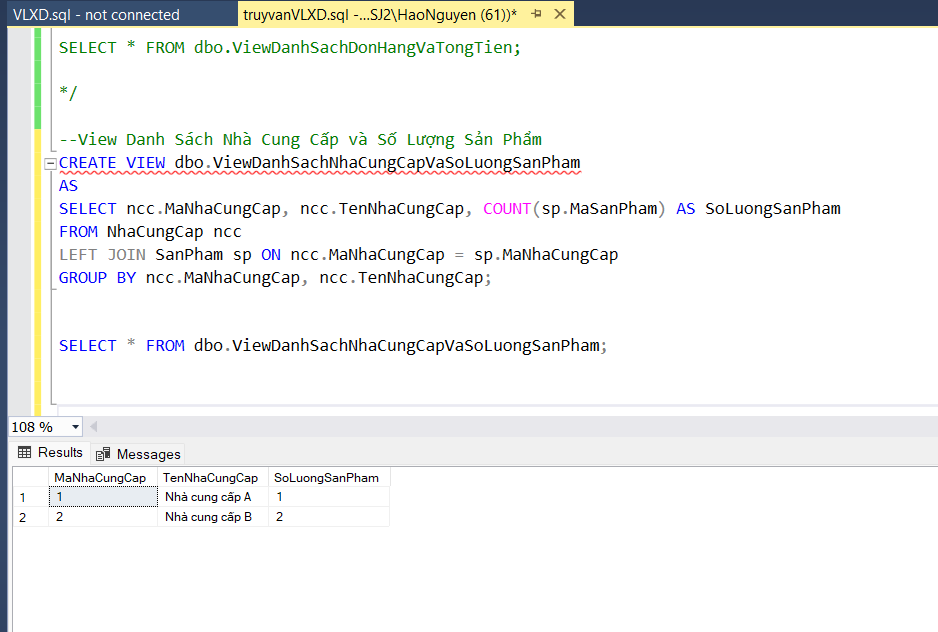
SELECT ncc.MaNhaCungCap, ncc.TenNhaCungCap, COUNT(sp.MaSanPham) AS SoLuongSanPham

FROM NhaCungCap ncc

LEFT JOIN SanPham sp ON ncc.MaNhaCungCap = sp.MaNhaCungCap

GROUP BY ncc.MaNhaCungCap, ncc.TenNhaCungCap;

SELECT \* FROM dbo.ViewDanhSachNhaCungCapVaSoLuongSanPham;

****

# *Hình 2. 36 View danh sách nhà cung cấp và số lương sản phẩm*

* 1. **Bảo mật và phâm quyền**

Bảo mật và phân quyền là hai khái niệm quan trọng trong SQL Server để đảm bảo an toàn và bảo vệ dữ liệu.

1. **Bảo mật SQL Server:**

SQL Server cung cấp các tính năng bảo mật như xác thực, ủy quyền và mã hóa để đảm bảo rằng người dùng chỉ có thể truy cập và thực thi các hoạt động phù hợp với quyền hạn của họ.

Xác thực giúp xác định danh tính của người dùng và đảm bảo rằng họ chỉ có thể truy cập và thực thi các hoạt động theo quyền hạn của mình.

Ủy quyền cho phép quản trị viên chỉ định các quyền hạn đối với người dùng, nhóm người dùng hoặc vai trò.

Mã hóa là một cách để bảo vệ dữ liệu khỏi các cuộc tấn công từ bên ngoài bằng cách mã hóa dữ liệu thành một dạng không đọc được nếu không có chìa khóa giải mã.

1. **Phân quyền SQL Server:**

Phân quyền là quá trình xác định các đối tượng cụ thể trong SQL Server (như bảng, thủ tục lưu trữ hoặc cơ sở dữ liệu) mà người dùng có thể truy cập và thực hiện các hoạt động cụ thể trên chúng.

Có nhiều cấp độ phân quyền, từ phân quyền cơ bản như SELECT, INSERT, UPDATE, DELETE cho đến phân quyền cao cấp như ALTER, DROP hoặc CONTROL.

Phân quyền giúp đảm bảo rằng người dùng chỉ có thể truy cập và thực hiện các hoạt động theo quyền hạn của họ và tránh được các cuộc tấn công từ bên ngoài.

Tóm lại, bảo mật và phân quyền là hai khái niệm quan trọng trong SQL Server để đảm bảo an toàn và bảo vệ dữ liệu. Việc thiết lập các quyền hạn và phân quyền phù hợp giúp người quản trị đảm bảo rằng các người dùng chỉ có thể truy cập và thực hiện các hoạt động theo quyền hạn của họ, đồng thời tránh được các cuộc tấn công từ bên ngoài.

# **CHƯƠNG 3. KẾT LUẬN**

## 3.1 Những kết quả đạt được của đồ án

* Học được nhiều kiến thức nâng cao về câu lệnh Sql
* Nhớ các câu lệnh cũng như những cú pháp cơ bản trong sql
* Thêm năng lực tự học hỏi
* Hoàn thiện cơ bản các chức năng quản lý sản phẩm

## 4.1 Nhược điểm của đồ án

* Chưa phát triển thành 1 app được
* Các câu lệnh còn đơn giản

# **TÀI LIỆU THAM KHẢO**

* <https://learn.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-2017>
* <https://learn.microsoft.com/vi-vn/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=sql-server-2017>
* <https://viblo.asia/p/gioi-thieu-stored-procedure-trong-sql-server-m68Z0VpM5kG>
* <https://learn.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql?view=sql-server-ver16>

**THÔNG TIN NHÓM**

Số thứ tự nhóm: **Nhóm 4**

|  |  |  |
| --- | --- | --- |
| **MSSV 0950080100** | **MSSV 0950080113** | **MSSV 0950080082** |
| -Phân tích nghiệp vụ và thiết kế CSDL  -Phân Quyền: Phân hệ Nhân viên  -Viết Báo Cáo  -Thao tác HQT CSDL  - Store Procedure  -Biểu đồ phân cấp chức năng | -Phân tích nghiệp vụ và thiết kế CSDL  -Phân quyền: Phân hệ khách hàng  **-**Viết báo cáo  -Thao tác HQT CSDL  - Function, Trigger  -Biểu đồ luồng dữ liệu | -Phân tích nghiệp vụ và thiết kế CSDL  -Phân quyền: phân hệ nhà cung cấp  -Viết Báo cáo  -Thao tácHQT CSDL  - View, bảo mật  -Phân loại các thuộc tính |

**ĐÁNH GIÁ**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MSSV 0950080100** | **MSSV 0950080113** | **MSSV 0950080082** |
| **Tham gia ( % )** | 34% | 33% | 33% |
| **Hoàn tất công việc ( % )** | 100% | 100% | 100% |
| **Yêu cầu đồ án** | | | |
| Phân tích nghiệp và thiết kế CSDL | | | 100% |
| Code dao diện chức năng | | | 75% |
| Phân quyền người dùng | | | 85% |
| Viết báo cáo | | | 100% |