**TRƯỜNG ĐẠI HỌC KINH TẾ**

**KHOA THỐNG KÊ – TIN HỌC**

----\*\*\*----



**BÁO CÁO MÔN HỌC PHÂN TÍCH**

**& THIẾT KẾ HỆ THỐNG**

**ĐỀ TÀI:**

**QUẢN LÝ NGUYÊN LIỆU QUÁN CAFÉ LECHAT**

GVHD : Th.S Cao Thị Nhâm

Nhóm : 48K211.10

Sinh viên thực hiện : 1. Nguyễn Thị Thu Uyên

2. Lâm Hồng Phúc

3. Nguyễn Thị Thúy

4. Nguyễn Thảo Duyên

5. Nguyễn Thị Vân Ly

*Đà Nẵng, tháng 10 năm 2024*

**MỤC LỤC**

[I. Hiệu chỉnh các bảng (R1+R2) 1](#_Toc183813475)

[1. Chuyển sơ đồ thành các bảng (Cơ sở dữ liệu kế thừa từ nhóm PTTKHTTT) 1](#_Toc183813476)

[2. Hiệu chỉnh các bảng 2](#_Toc183813477)

[3. Sơ đồ lớp 2](#_Toc183813478)

[II. Thiết kế chi tiết các bảng (R1+R2) 2](#_Toc183813479)

[III. Xây dựng cơ sở dữ liệu (R3) 6](#_Toc183813480)

[IV. Xây dựng các module (R4) 6](#_Toc183813481)

[V. Xây dựng cơ chế bảo mật (R5) 26](#_Toc183813482)

[VI. Xây dựng cơ chế backup dữ liệu tự động (R6) 30](#_Toc183813483)

[VII. Giải pháp dữ liệu lớn (R7) 35](#_Toc183813484)

[VIII. Phương án để hạn chế tấn công SQL Injection (R8) 38](#_Toc183813485)

DANH MỤC BẢNG

[Bảng 1.1 NhaCungCap 4](#_Toc180313763)

[Bảng 1.2 NguyenLieu 4](#_Toc180313764)

[Bảng 1.3 PhieuNhapKho 4](#_Toc180313765)

[Bảng 1.4 PhieuNhapKhoChiTiet 5](#_Toc180313766)

[Bảng 1.5 PhieuXuatKho 5](#_Toc180313767)

[Bảng 1.6 PhieuXuatKhoChiTiet 6](#_Toc180313768)

[Bảng 1.7 PhieuKiemKho 6](#_Toc180313769)

[Bảng 1.8 PhieuKiemKhoChiTiet 7](#_Toc180313770)

1. **Hiệu chỉnh các bảng (R1+R2)**
2. **Chuyển sơ đồ thành các bảng (Cơ sở dữ liệu kế thừa từ nhóm PTTKHTTT)**



NguyenLieu (MaNguyenLieu, TenNguyenLieu, LoaiNguyenLieu,

SoPhieuNhap, SoPhieuXuat, SoPhieuKiem)

NguyenLieuChiTiet (MaNguyenLieu, GiaBan, HanSuDung, TenNguyenLieu,

LoaiNguyenLieu)

PhieuNhapKho (SoPhieuNhap, NgayNhap, TenNguyenLieu)

PhieuNhapKhoChiTiet (SoPhieuNhap, NgayNhap, TenNguyenLieu,

LoaiNguyenLieu, SoLuong, HanSuDung, TongGiaNhap)

PhieuXuatKho (SoPhieuXuat, SoLuongXuat,NgayXuat)

PhieuXuatKhoChiTiet (SoPhieuXuat, SoLuongXuat, TenNguyenLieu,

NgayXuat)

PhieuKiemKho (SoPhieuKiem, NgayKiem)

PhieuKiemKhoChiTiet (SoPhieuKiem, TenNguyenLieu, SoLuong,

HanSuDung, GiaBan)

1. **Hiệu chỉnh các bảng**

NhaCungCap (MaNCC, TenNCC, ĐiaChi, SDT)

PhieuNhapKho (MaNK, NgayNhap,MaNCC)

PhieuNhapKhoChiTiet (MaNKCT, MaNK,MaNL, SoLuongNhap, DonGia)

NguyenLieu (MaNL, TenNL,ĐVT,HSD,SoLuong)

PhieuXuatKho (MaXK, NgayXuat)

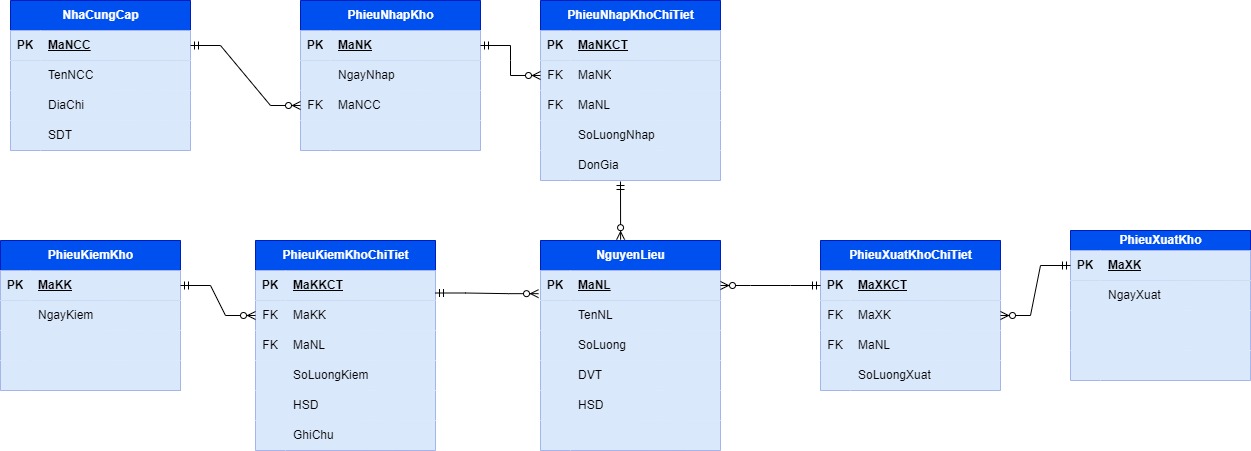
PhieuXuatKhoChiTiet (MaXKCT, MaNL, MaXK,SoLuongXuat)

PhieuKiemKho (MaKK,NgayKiem)

PhieuKiemKhoChiTiet (MaKKCT, MaKK, MaNL, SoLuongKiem, HSD,

GhiChu)

1. **Sơ đồ lớp**



Hình 1. Sơ đồ quan hệ

1. **Thiết kế chi tiết các bảng (R1+R2)**

Thay đổi: SDT Int 🡪 SDT Char(10)

MaNL Char(5) 🡪 MaNL Char(7)

Datetime 🡪 Date

* NhaCungCap

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaNCC | Char(7) | PK | Not null | Mã nhà cung cấp |
| 2 | TenNCC | Nvarchar(50) |  | Not null | Tên nhà cung cấp |
| 3 | DiaChi | Nvarchar(100) |  | Not null | Địa chỉ |
| 4 | SDT | Char(10) |  | Not null | Số điện thoại |

Bảng 1. 1 NhaCungCap

* NguyenLieu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaNL | Char(7) | PK | Not null | Mã nguyên liệu |
| 2 | TenNL | Nvarchar(50) |  | Not null | Tên nguyên liệu |
| 3 | ĐVT | Varchar(10) |  | Not null | Đơn vị tính |
| 4 | HSD | Date |  | Not null | Hạn sử dụng |
| 5 | SoLuong | Int |  | Not null | Số Lượng |

Bảng 1. 2 NguyenLieu

* PhieuNhapKho

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaNK | Char(7) | PK | Not null | Mã nhập kho |
| 2 | NgayNhap | Date |  | Not null | Ngày Nhập |
| 3 | MaNCC | Char(7) | FK | Not null | Khóa ngoại đến bảng NhaCungCap |

Bảng 1. 3 PhieuNhapKho

* PhieuNhapKhoChiTiet

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaNKCT | Char(7) | PK | Not null | Mã nhập kho chi tiết |
| 2 | MaNK | Char(7) | FK | Not null | Khóa ngoại đến bảng PhieuNhapKho |
| 3 | MaNL | Char(7) | FK | Not null | Khóa ngoại đến bảng NguyenLieu |
| 4 | SoLuongNhap | Int |  | Not null | Số lượng nhập |
| 5 | DonGia | Int |  | Not null | Đơn giá |

Bảng 1. 4 PhieuNhapKhoChiTiet

* PhieuXuatKho

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaXK | Char(7) | PK | Not null | Mã xuất kho |
| 2 | NgayXuat | Date |  | Not null | Ngày xuất |

Bảng 1. 5 PhieuXuatKho

* PhieuXuatKhoChiTiet

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaXKCT | Char(7) | PK | Not null | Mã xuất kho chi tiết |
| 2 | MaXK | Char(7) | FK | Not null | Khóa ngoại đến bảng PhieuXuatKho |
| 3 | MaNL | Char(7) | FK | Not null | Khóa ngoại đến bảng NguyenLieu |
| 4 | SoLuongXuat | Int(10) |  | Not null | Số lượng xuất |

Bảng 1. 6 PhieuXuatKhoChiTiet

* PhieuKiemKho

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaKK | Char(7) | PK | Not null | Mã nguyên liệu |
| 2 | NgayKiem | Date |  | Not null | Tên nguyên liệu |

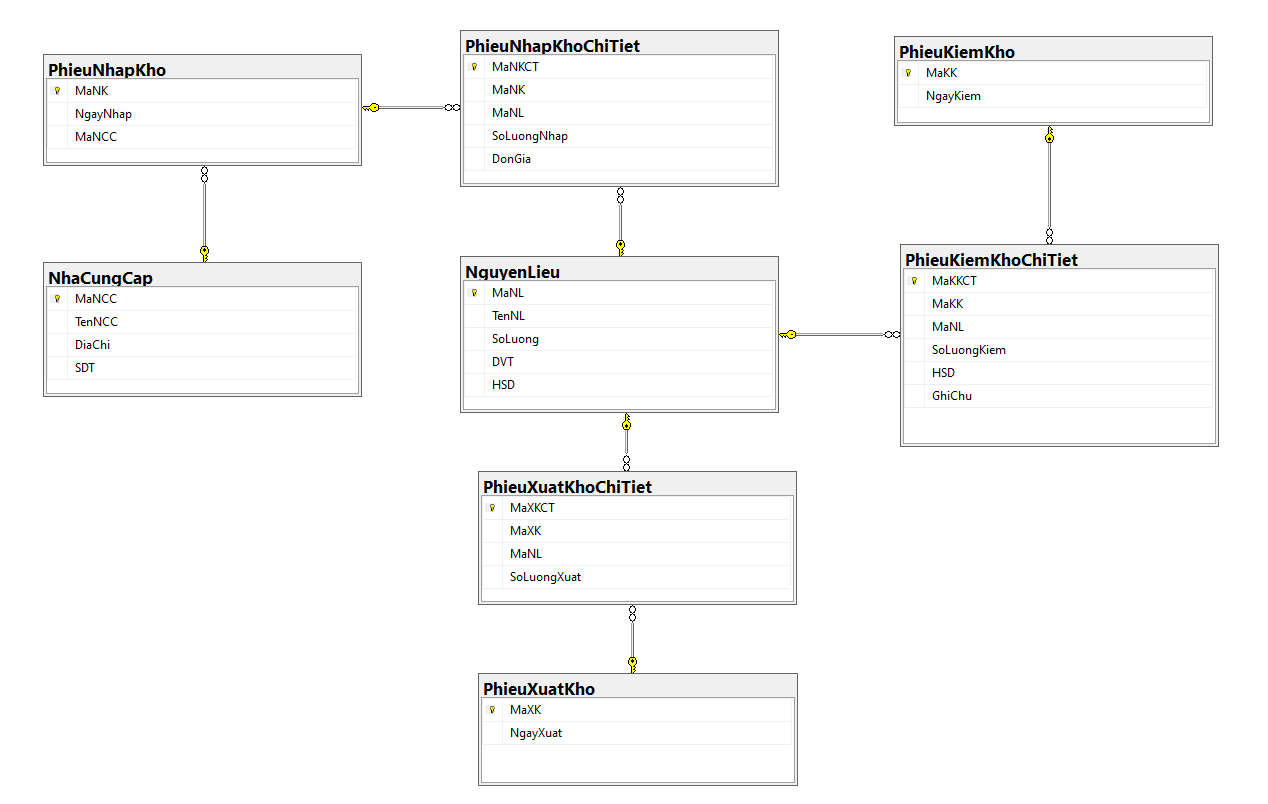
Bảng 1. 7 PhieuKiemKho

* PhieuKiemKhoChiTiet

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Cột | Kiểu dữ liệu | Khóa | Ràng buộc | Mô tả |
| 1 | MaKKCT | Char(7) | PK | Not null | Mã kiểm kho chi tiết |
| 2 | MaKK | Char(7) | FK | Not null | Khóa ngoại đến bảng PhieuKiemKho |
| 3 | MaNL | Char(7) | FK | Not null | Khóa ngoại đến bảng NguyenLieu |
| 4 | SoLuongKiem | Int |  | Not null | Số lượng kiểm |
| 5 | HSD | Date |  | Not null | Hạn sử dụng |
| 6 | GhiChu | Nvarchar(100) |  |  | Ghi chú |

Bảng 1. 8 PhieuKiemKhoChiTiet

1. **Xây dựng cơ sở dữ liệu (R3)**

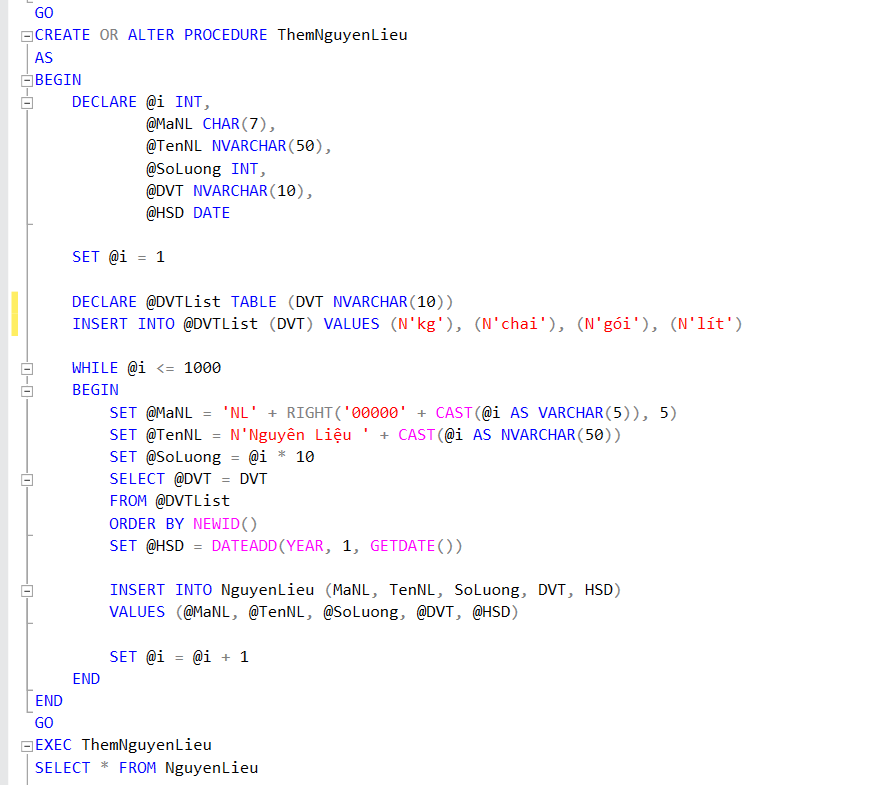


1. **Xây dựng các module (R4)**

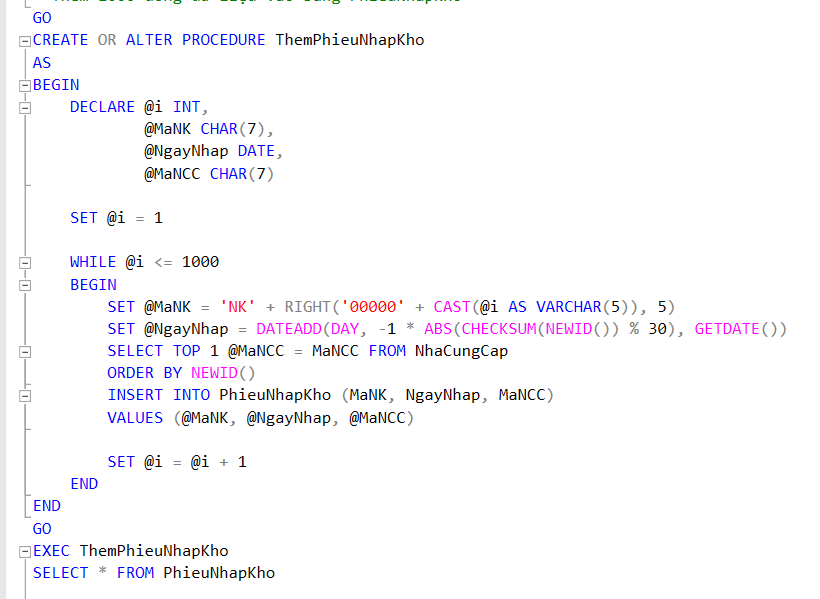
* Thêm 1000 dòng dữ liệu cho bảng NhaCungCap

****

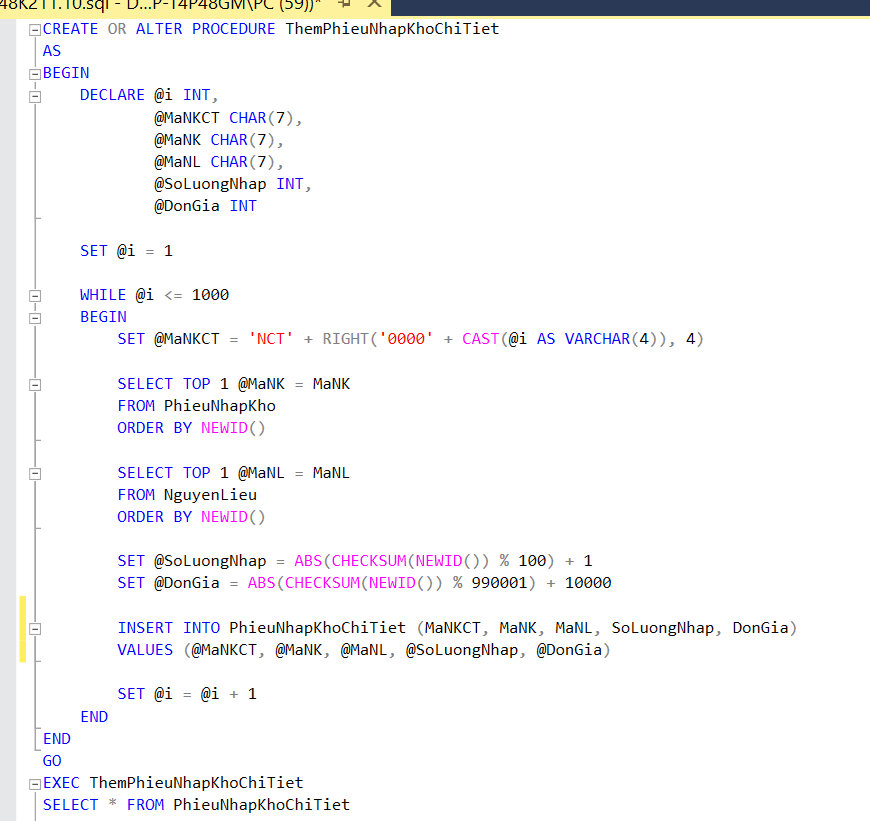
* Thêm 1000 dòng dữ liệu vào bảng NguyenLieu

f

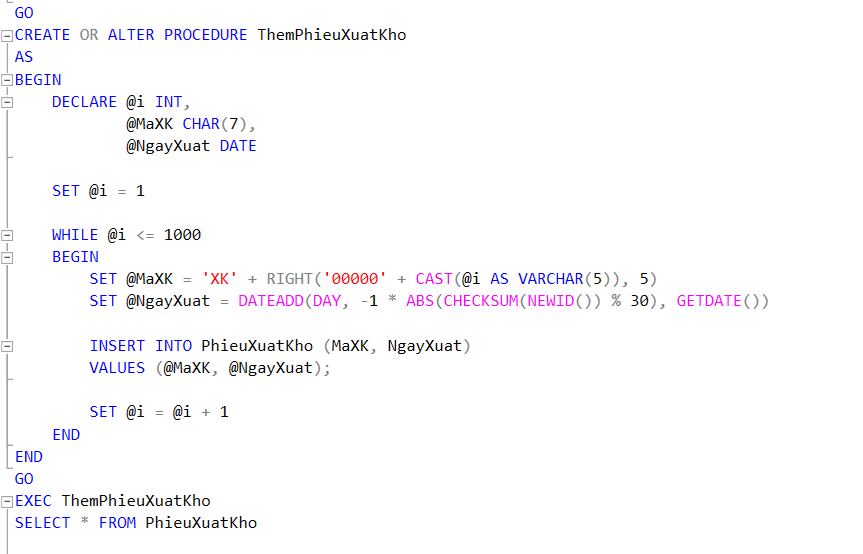
* Thêm 1000 dòng dữ liệu vào bảng PhieuNhapKho



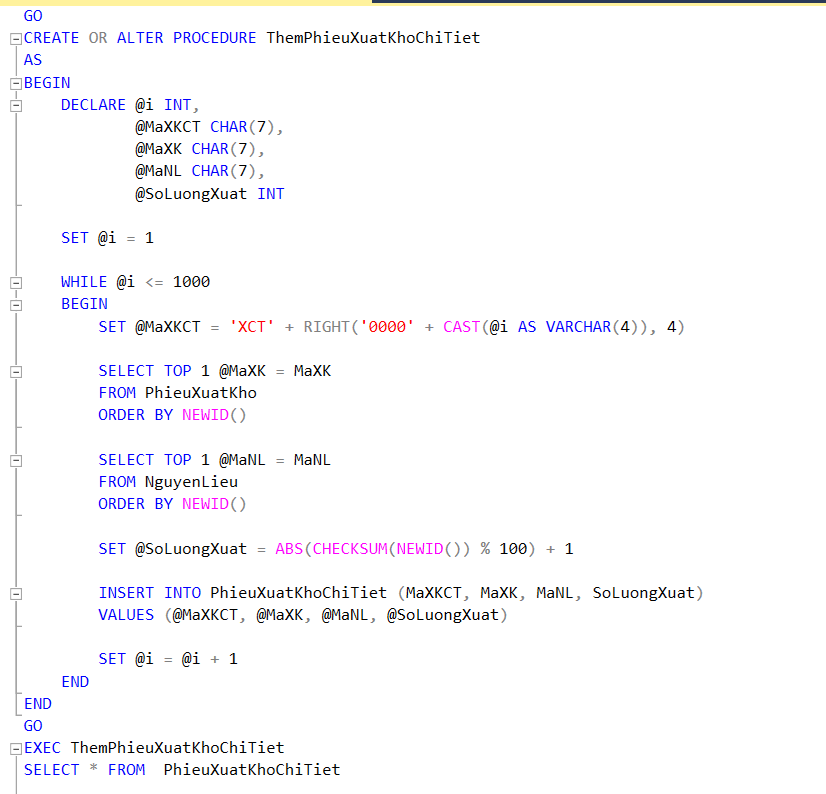
* Thêm 1000 dòng dữ liệu vào bảng PhieuNhapKhoChiTiet



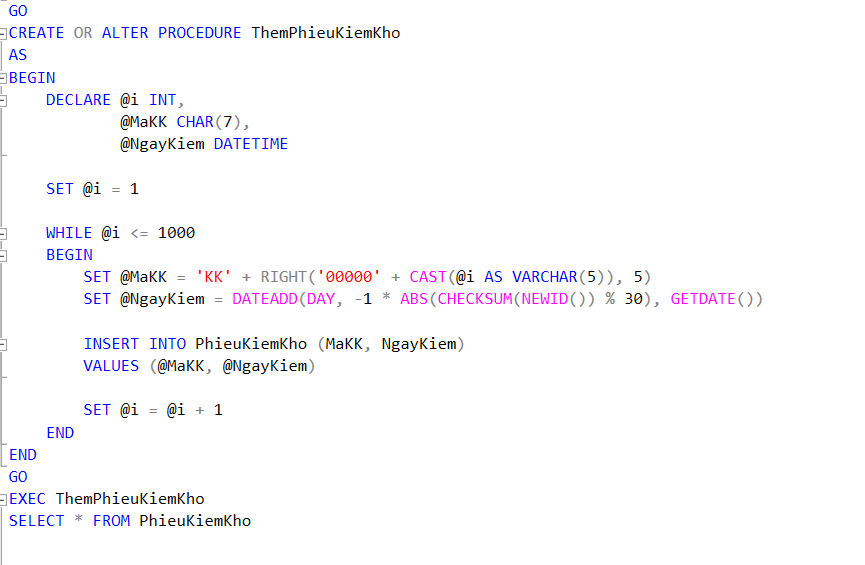
* Thêm 1000 dòng dữ liệu vào bảng PhieuXuatKho



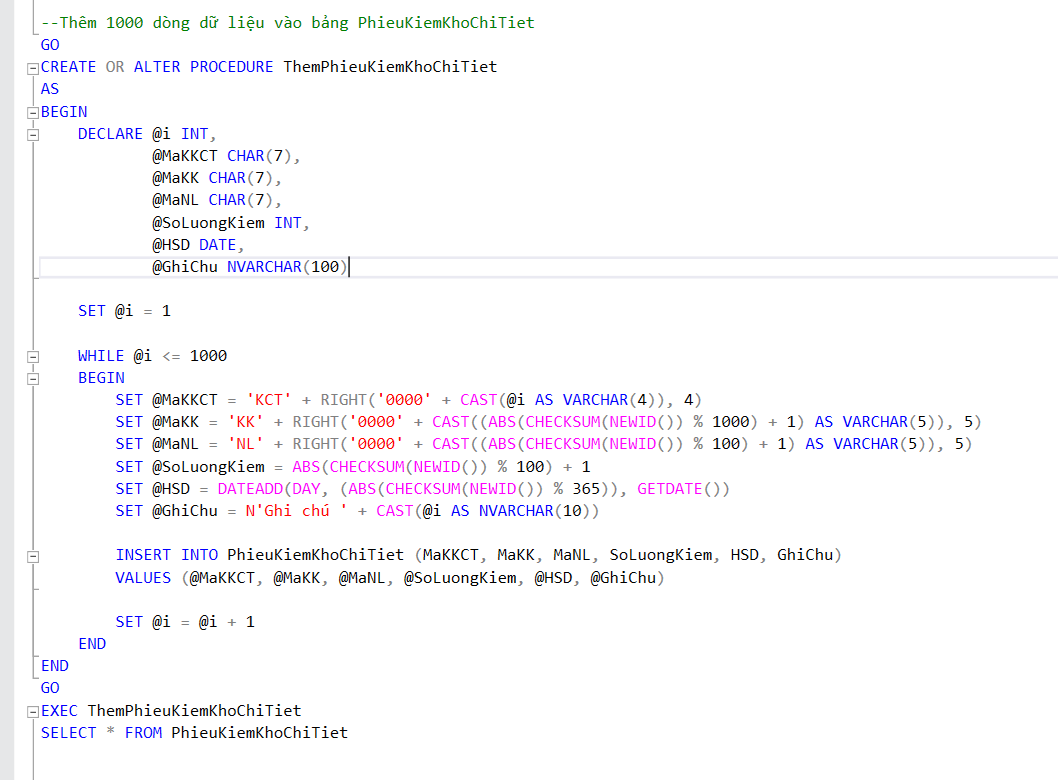
* Thêm 1000 dòng dữ liệu vào bảng PhieuXuatKhoChiTiet



* Thêm 1000 dòng dữ liệu vào bảng ThemPhieuKiemKho

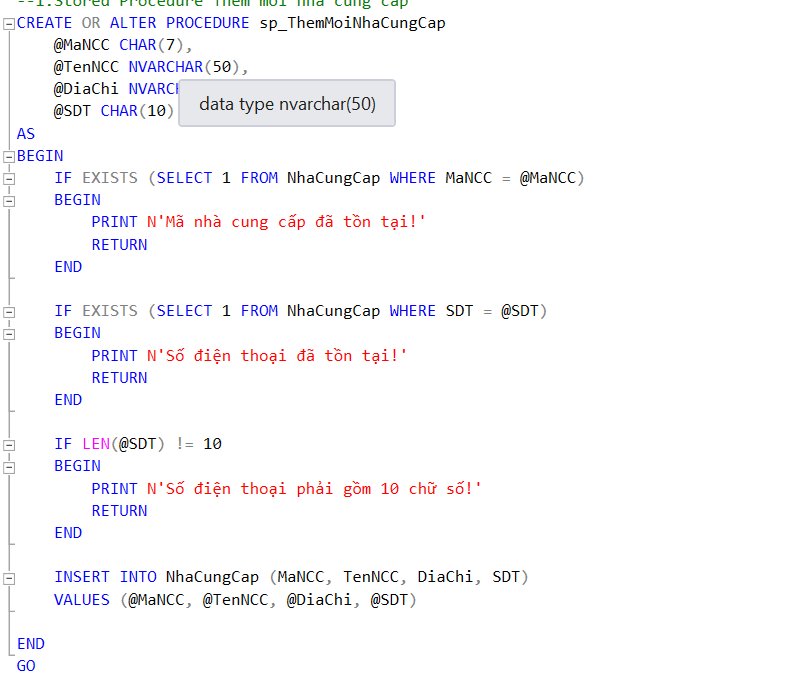


* Thêm 1000 dòng dữ liệu vào bảng PhieuKiemKhoChiTiet



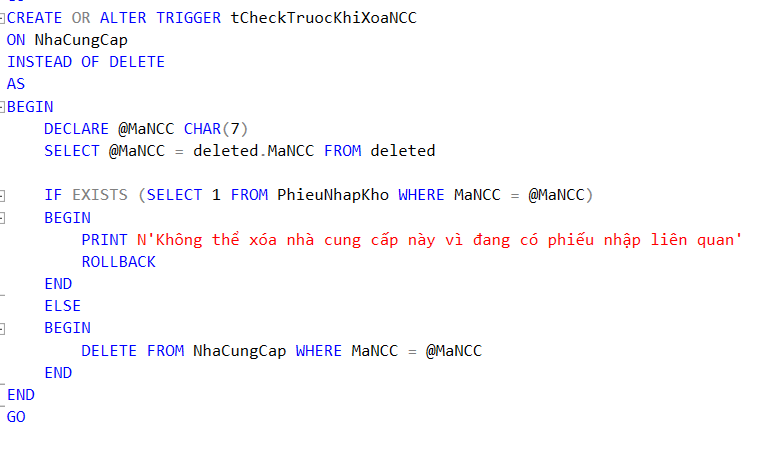
1. **Stored Procedure Thêm mới nhà cung cấp**

|  |  |
| --- | --- |
| **Mô tả:** | Stored Procedure thêm mới một nhà cung cấp vào bảng NhaCungCap trong cơ sở dữ liệu. |
| **Input:** | @MaNCC, @TenNCC, @DiaChi , @SDT |
| **Output:** | N/A |
| **Process**: | 1. Kiểm tra mã nhà cung cấp @MaNCC đã tồn tại hay chưa. Nếu tồn tại, in thông báo 'Mã nhà cung cấp đã tồn tại!’ và dừng xử lý. 2. Kiểm tra nếu số điện thoại @SDT đã tồn tại trong bảng. Nếu tồn tại, in thông báo 'Số điện thoại đã tồn tại!' và dừng xử lý. 3. Kiểm tra xem độ dài của số điện thoại có phải là 10 ký tự không. Nếu không, in thông báo 'Số điện thoại phải gồm 10 chữ số!' và dừng xử lý. 4. Nếu tất cả các kiểm tra đều thành công, thực hiện câu lệnh INSERT để thêm nhà cung cấp mới vào bảng NhaCungCap với các thông tin đã cung cấp. |



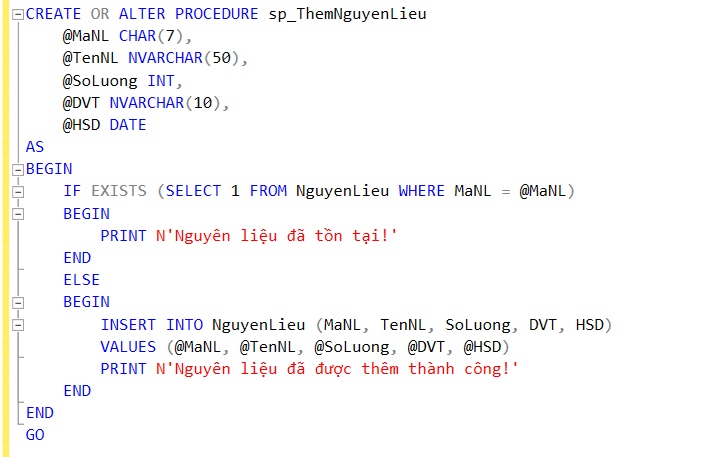
1. **Trigger kiểm tra trước khi xóa nhà cung cấp**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger kiểm tra trước khi xóa nhà cung cấp, đảm bảo việc xóa nhà cung cấp trong bảng NhaCungCap nếu nhà cung cấp đó đang có liên quan đến các phiếu nhập kho trong bảng PhieuNhapKho. |
| **Loại Trigger:** | INSTEAD OF |
| **Sự kiện:** | DELETE |
| **Bảng:** | NhaCungCap |
| **Process:** | 1. Lấy mã nhà cung cấp @MaNCC từ bảng deleted 2. Kiểm tra xem nhà cung cấp với mã @MaNCC có tồn tại trong bảng PhieuNhapKho không. Nếu có, thông báo 'Không thể xóa nhà cung cấp này vì đang có phiếu nhập liên quan' và dừng xử lý 3. Nếu không có liên quan, thực hiện câu lệnh DELETE để xóa bản ghi nhà cung cấp khỏi bảng NhaCungCap. |

****

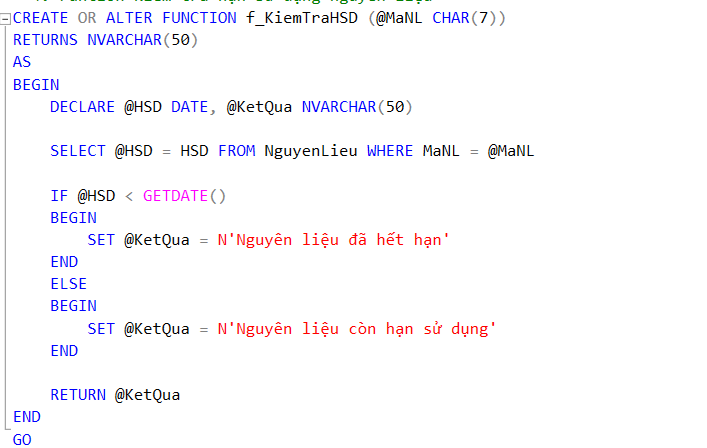
1. **Stored Procedure thêm mới nguyên liệu**:

|  |  |
| --- | --- |
| **Mô tả:** | Stored Procedure thêm mới một nguyên liệu vào bảng NguyenLieu, kiểm tra xem mã nguyên liệu đã tồn tại hay chưa trước khi thực hiện việc thêm mới. |
| **Input:** | @MaNL,@TenNL ,@SoLuong,@DVT,@HSD |
| **Output:** | N/A |
| **Process:** | 1. Kiểm tra xem nguyên liệu với mã @MaNL đã tồn tại trong hệ thống chưa. 2. Nếu nguyên liệu đã tồn tại, in thông báo rằng nguyên liệu này đã có sẵn. 3. Nếu nguyên liệu chưa tồn tại, thêm dữ liệu về nguyên liệu mới vào bảng NguyenLieu |



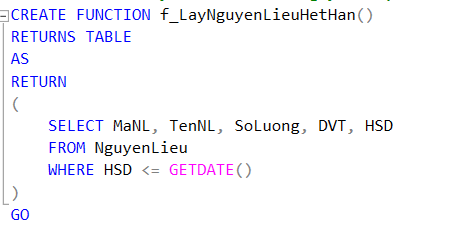
1. **Funtion Kiểm tra hạn sử dụng nguyên liệu**

|  |  |
| --- | --- |
| **Mô tả:** | Function Kiểm tra hạn sử dụng (HSD) của một nguyên liệu cụ thể trong bảng NguyenLieu. |
| **Input:** | @MaNL |
| **Output:** | In ra màn hình ‘Nguyên liệu đã hết hạn’ nếu nguyên liệu đã hết  hạn sử dụng. Ngược lại , in ra ‘Nguyên liệu còn hạn sử dụng’. |
| **Process:** | 1. Dựa trên mã nguyên liệu (MaNL), hàm sẽ kiểm tra ngày hết hạn (HSD) của nguyên liệu từ bảng NguyenLieu. 2. Nếu ngày hết hạn nhỏ hơn ngày hiện tại, hàm sẽ trả về thông báo "Nguyên liệu đã hết hạn" 3. Nếu ngày hết hạn lớn hơn hoặc bằng ngày hiện tại, hàm trả về thông báo "Nguyên liệu còn hạn sử dụng". |



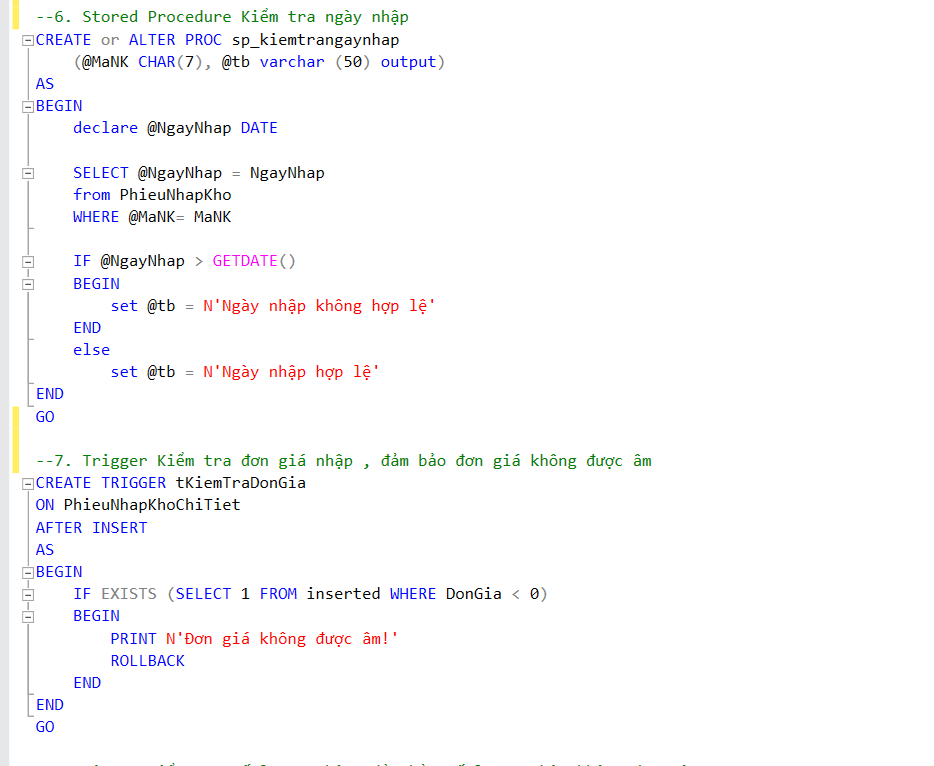
1. **Function lấy ra danh sách nguyên liệu hết HSD**

|  |  |
| --- | --- |
| **Mô tả:** | Function lấy danh sách tất cả các nguyên liệu trong bảng NguyenLieu đã hết hạn sử dụng. |
| **Input:** | N/A |
| **Output:** | Bảng (Danh sách nguyên liệu hết hạn) |
| **Process:** | 1. Hàm thực hiện truy vấn SQL trên bảng NguyenLieu để chọn tất cả các nguyên liệu có ngày hết hạn (HSD) nhỏ hơn hoặc bằng ngày hiện tại. |

****

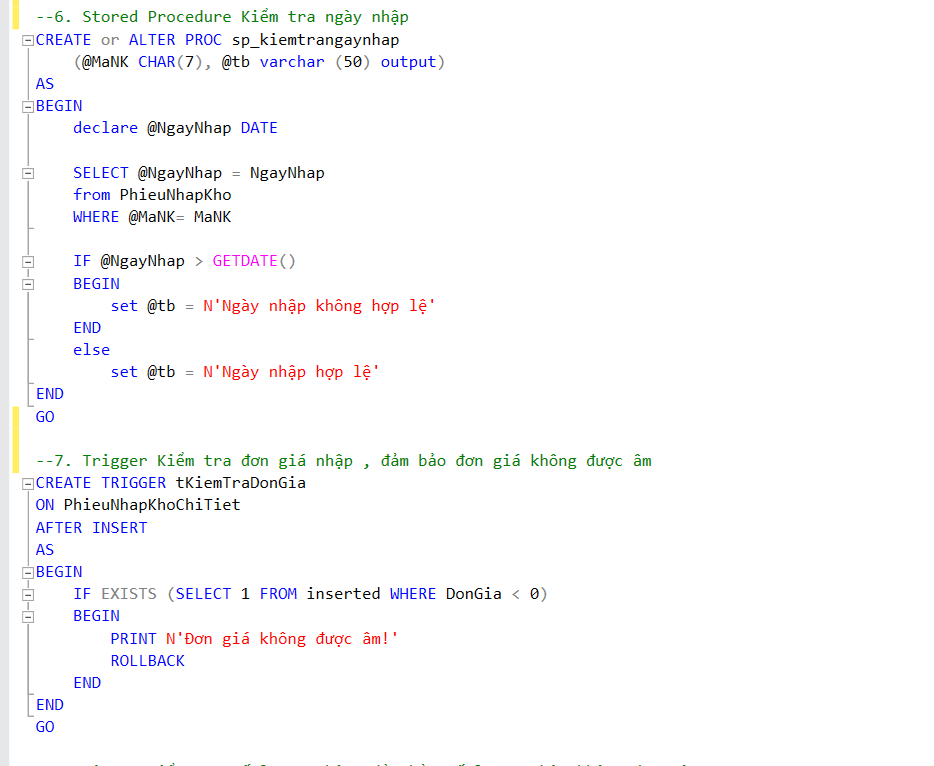
1. **Stored Procedure Kiểm tra ngày nhập**

|  |  |
| --- | --- |
| **Mô tả:** | Stored Procedure dựa trên mã nhập kho (MaNK) để kiểm tra tính hợp lệ của ngày nhập (NgayNhap) |
| **Input:** | @MaNK |
| **Output:** | @tb |
| **Process:** | 1. Truy vấn  để lấy @NgayNhap của phiếu nhập kho tương ứng với @MaNK từ bảng PhieuNhapKho. 2. Kiểm tra nếu @NgayNhap lớn hơn ngày hiện tại (getdate) thì thông báo ‘Ngày nhập không hợp lệ’ và dừng xử lý. Ngược lại, thông báo ‘Ngày nhập hợp lệ’ |



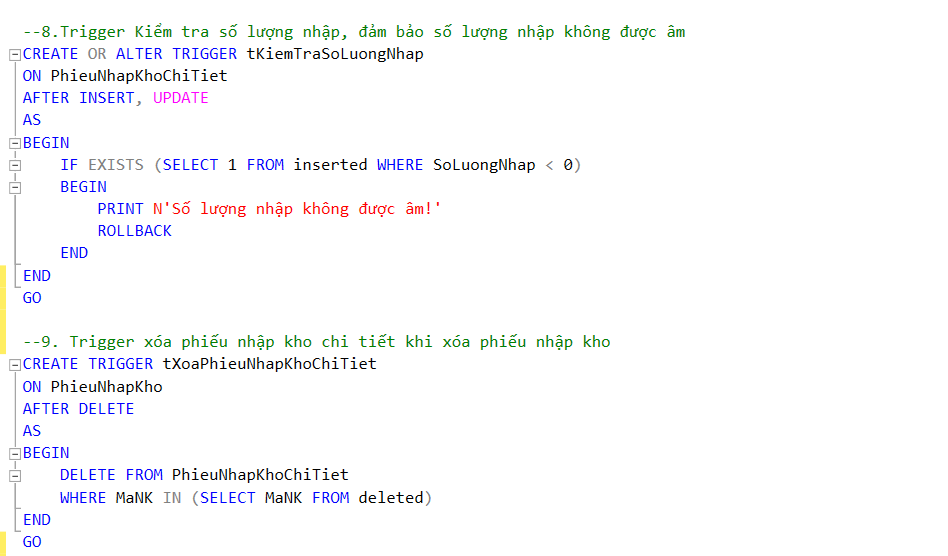
1. **Trigger Kiểm tra đơn giá nhập**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger kiểm tra đơn giá có hợp lệ không, , đảm bảo đơn giá không được âm |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | INSERT |
| **Bảng:** | PhieuNhapKhoChiTiet |
| **Process:** | 1. Kiểm tra bất kỳ DonGia nào trong bảng inserted có giá trị DonGia âm hay không nếu có thì thông báo ‘ Đơn giá không được âm’ và kết thúc 2. Nếu có có DonGia âm , sẽ thực hiện câu lệnh thêm bản ghi mới từ bảng inserted vào bảng PhieuNhapKhoChiTiet |



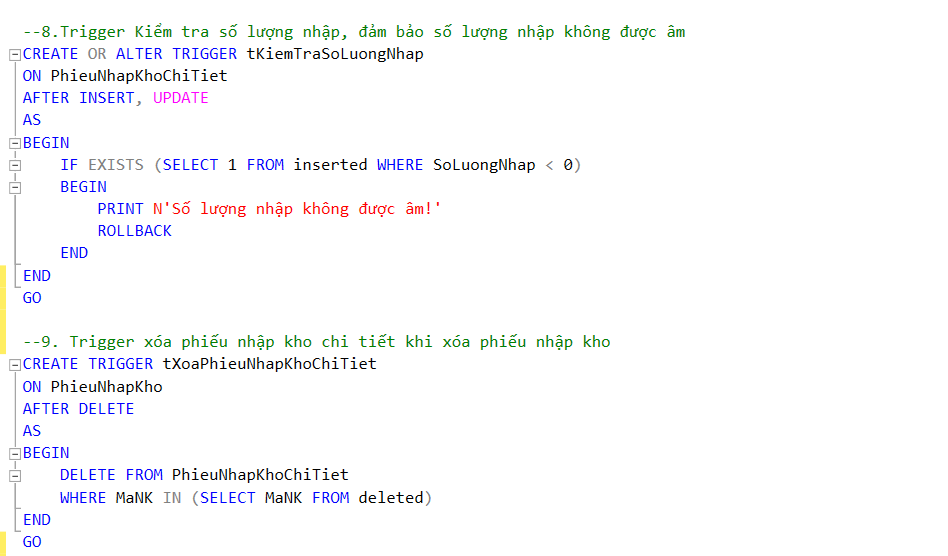
1. **Trigger Kiểm tra số lượng nhập**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger kiểm tra số lượng nhập kho có hợp lệ không, đảm bảo số lượng nhập không được âm |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | INSERT, UPDATE |
| **Bảng:** | PhieuNhapKhoChiTiet |
| **Process:** | 1. Kiểm tra bất kỳ SoLuongNhap nào trong bảng inserted có giá trị SoLuongNhap âm hay không . Nếu có thì thông báo ‘ Số lượng nhập không được âm’ và kết thúc 2. Nếu có SoLuongNhap âm , sẽ thực hiện câu lệnh thêm bản ghi mới từ bảng inserted vào bảng PhieuNhapKhoChiTiet |

****

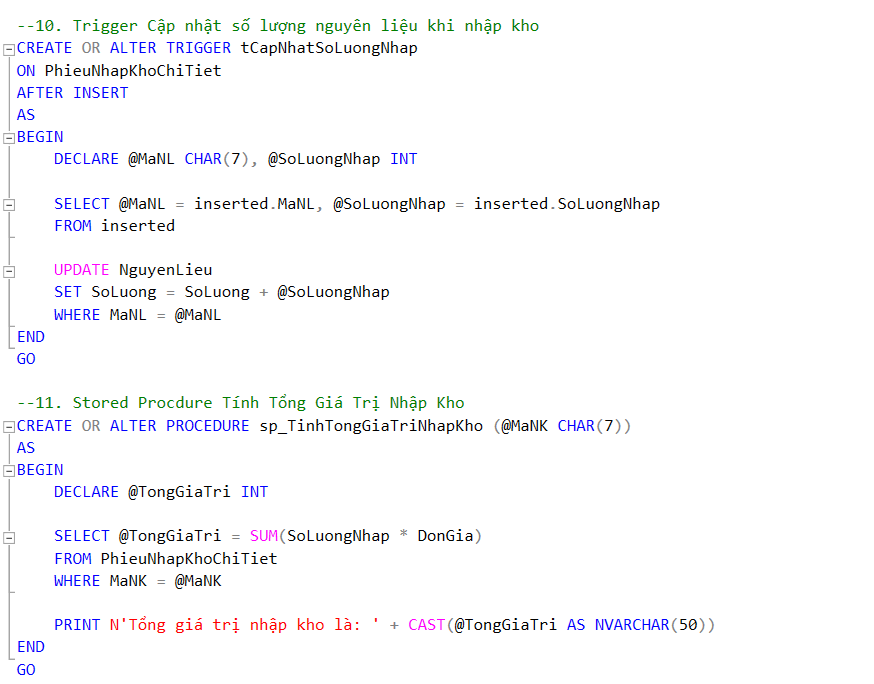
1. **Trigger xóa phiếu nhập kho chi tiết khi xóa phiếu nhập kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger giúp đảm bảo rằng khi một phiếu nhập kho bị xóa, tất cả các bản ghi chi tiết tương ứng trong bảng PhieuNhapKhoChiTiet cũng sẽ được xóa. |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | DELETE |
| **Bảng:** | PhieuNhapKho |
| **Process:** | 1. Xóa tất cả các bản ghi trong bảng PhieuNhapKhoChiTiet mà có MaNK (Mã phiếu nhập kho) tương ứng với những phiếu nhập kho vừa bị xóa. |



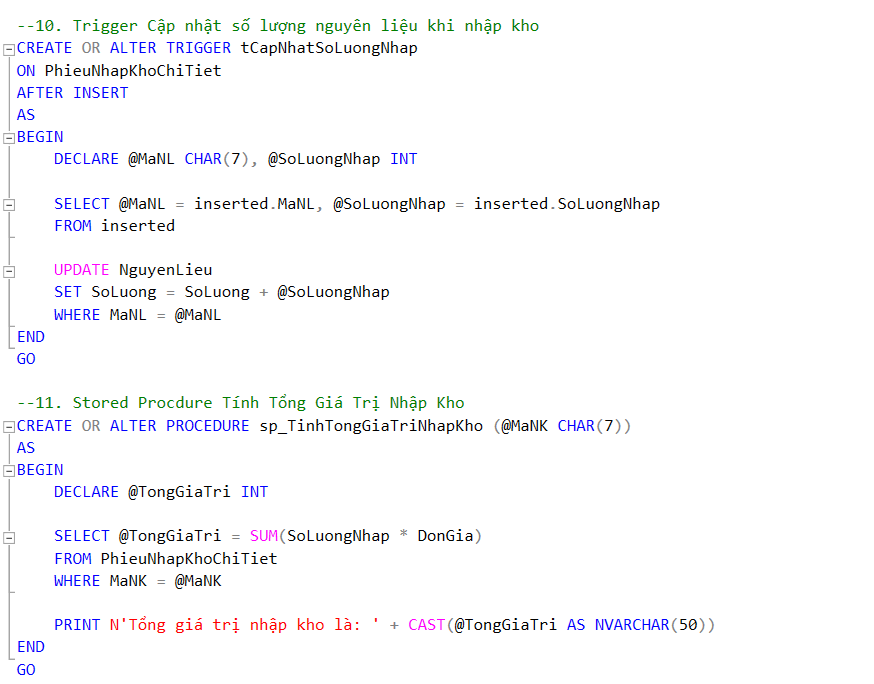
1. **Trigger cập nhật số lượng nguyên liệu khi nhập kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger cập nhật số lượng nguyên liệu khi nhập kho |
| **Loại trigger** | AFTER |
| **Sự kiện:** | INSERT |
| **Bảng:** | PhieuNhapKhoChiTiet |
| **Process:** | 1. Khi có bản ghi mới được thêm vào PhieuNhapKhoChiTiet, trigger sẽ lấy mã nguyên liệu (MaNL) và số lượng nhập (SoLuongNhap) từ bản ghi mới. 2. Sau đó, cập nhật số lượng của nguyên liệu tương ứng trong bảng NguyenLieu bằng cách cộng thêm số lượng nhập vào số lượng hiện có. |



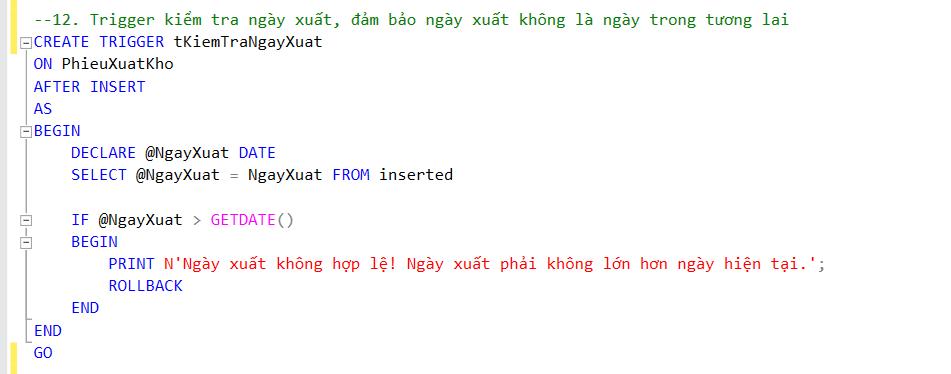
1. **Stored Procdure Tính tổng giá trị nhập kho**

|  |  |
| --- | --- |
| **Mô tả:** | Procedure Tính tổng giá trị nhập kho của hàng hóa đã nhập kho dựa trên mã phiếu nhập kho (MaNK). |
| **Input:** | @MaNK |
| **Output:** | @TongGiaTri |
| **Process:** | 1. Dựa trên mã phiếu nhập kho (MaNK), thủ tục sẽ tính tổng giá trị của phiếu nhập bằng cách nhân số lượng nhập với đơn giá cho mỗi nguyên liệu trong phiếu. |

****

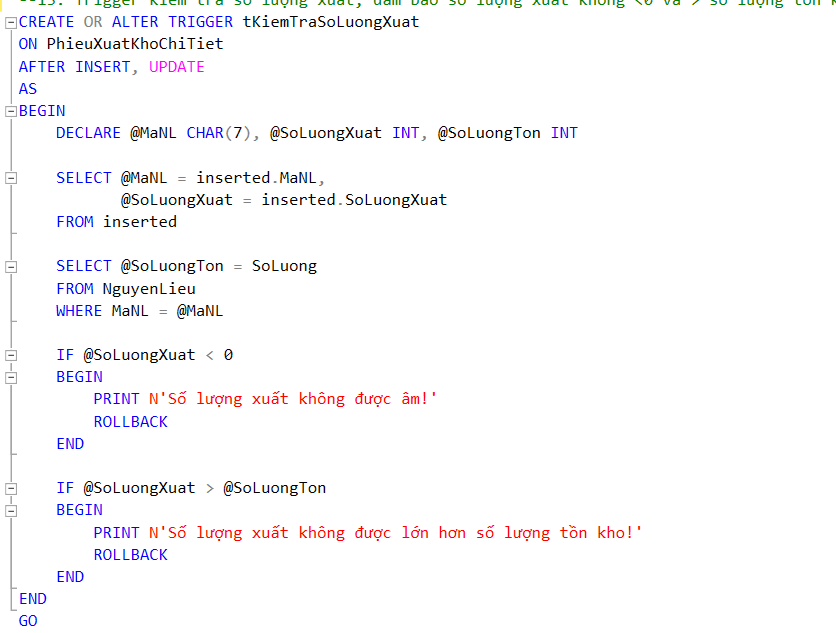
1. **Trigger kiểm tra ngày xuất**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger tự động kiểm tra tính hợp lệ của ngày xuất kho mỗi khi có bản ghi mới được chèn vào bảng PhieuXuatKho, đảm bảo ngày xuất không là ngày trong tương lai |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | INSERT |
| **Bảng:** | PhieuXuatKho |
| **Process:** | 1. Lấy dữ liệu @NgayXuat từ inserted 2. Nếu giá trị của @NgayXuat > ngày hiện tại thì in thông báo lỗi 3. Nếu ngày xuất hợp lệ (ngày không lớn hơn ngày hiện tại), trigger sẽ chèn bản ghi vào bảng PhieuXuatKho với các giá trị từ bảng inserted. |



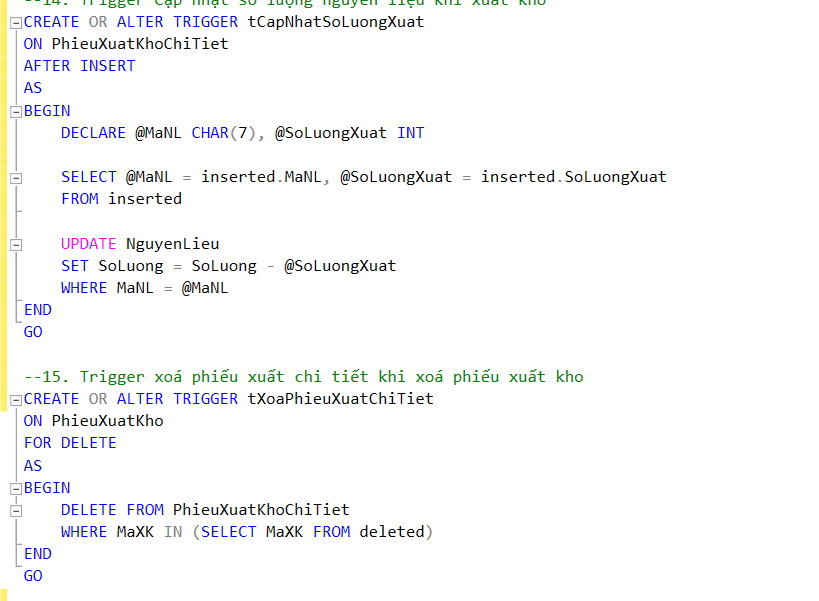
1. **Trigger kiểm tra số lượng xuất**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger sử dụng để kiểm tra tính hợp lệ của số lượng xuất trong bảng PhieuXuatKhoChiTiet trước khi chèn một bản ghi mới, đảm bảo số lượng xuất không nhỏ hơn 0 và lớn hơn số lượng tồn kho |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | INSERT, UPDATE |
| **Bảng:** | PhieuXuatKhoChiTiet |
| **Process:** | 1. Lấy giá trị số lượng xuất @SoLuongXuat  từ bản ghi vừa được chèn từ bảng inserted. 2. Kiểm tra tính hợp lệ của số lượng xuất: Nếu @SoLuongXuat <= 0: thông báo 'Số lượng xuất không được âm!' 3. Nếu @SoLuongXuat > @SoLuongTon: thông báo ‘Số lượng xuất không được lớn hơn số lượng tồn kho!' 4. Ngược lại, trigger sẽ thực hiện câu lệnh INSERT để thêm bản ghi mới vào bảng PhieuXuatKhoChiTiet với các thông tin MaXKCT, MaXK, MaNL, và SoLuongXuat từ bảng inserted. |



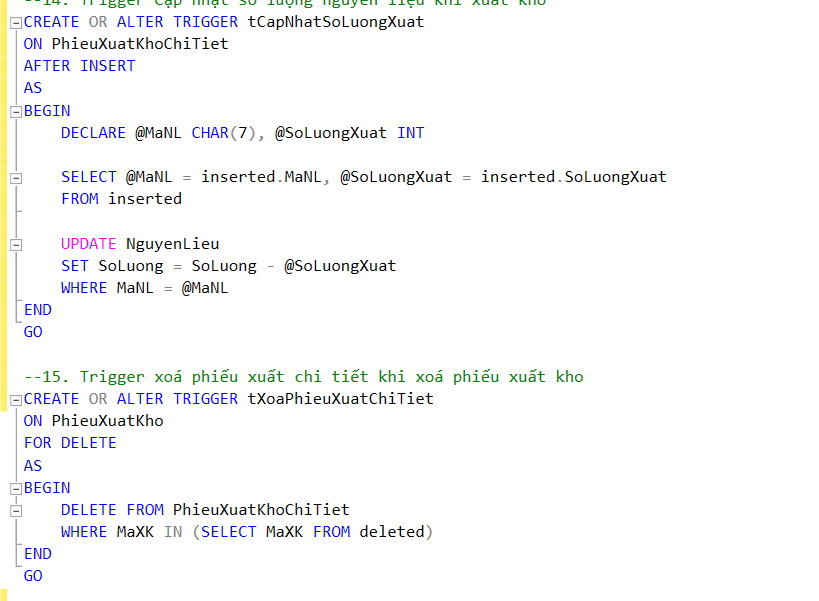
1. **Trigger Cập nhật số lượng nguyên liệu khi xuất kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger Cập nhật số lượng nguyên liệu khi xuất kho |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | INSERT |
| **Bảng:** | PhieuXuatKhoChiTiet |
| **Process:** | 1. Khi có bản ghi mới được thêm vào PhieuXuatKhoChiTiet, trigger sẽ lấy mã nguyên liệu (MaNL) và số lượng xuất (SoLuongXuat) từ bản ghi mới. 2. Số lượng nguyên liệu tương ứng trong bảng NguyenLieu sẽ được cập nhật bằng cách trừ đi số lượng xuất. |



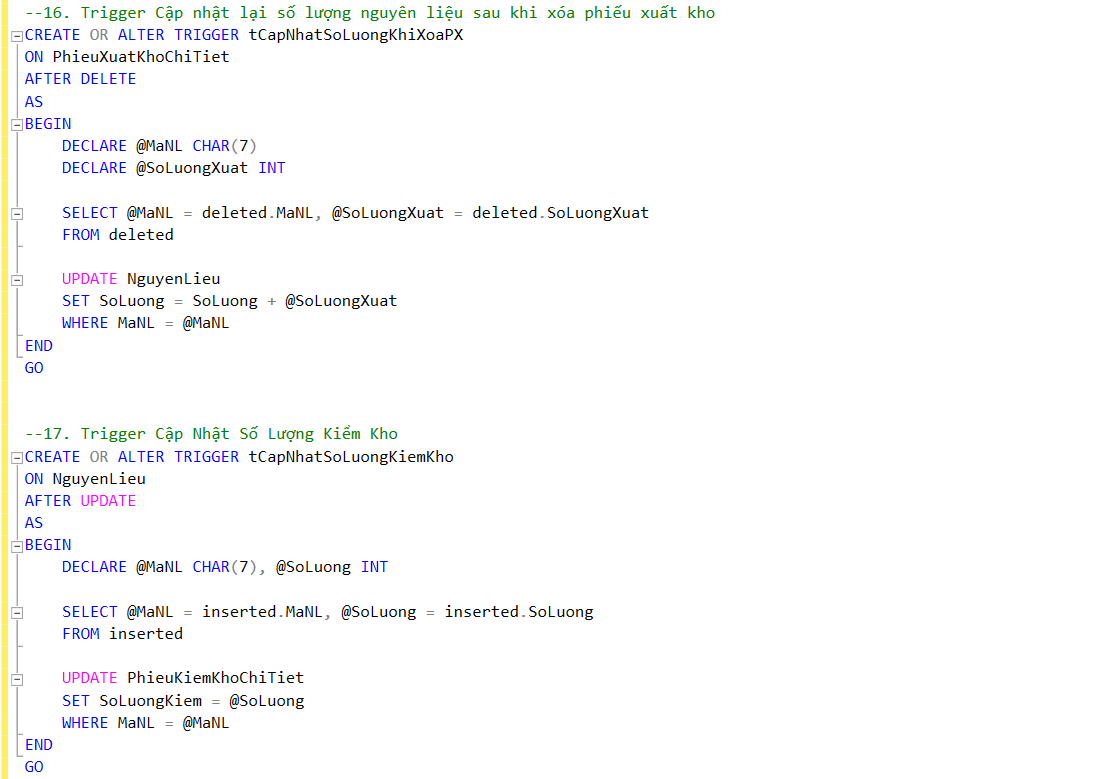
1. **Trigger xoá phiếu xuất chi tiết khi xoá phiếu xuất kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger giúp đảm bảo rằng khi một phiếu xuất kho bị xóa, tất cả các bản ghi chi tiết tương ứng trong bảng PhieuXuatKhoChiTiet cũng sẽ được xóa. |
| **Loại trigger:** | FOR |
| **Sự kiện:** | DELETE |
| **Bảng:** | PhieuXuatKho |
| **Process:** | 1. Xóa tất cả các bản ghi trong bảng PhieuXuatKhoChiTiet mà có MaXK (Mã phiếu xuất kho) tương ứng với những phiếu xuất kho vừa bị xóa. |

****

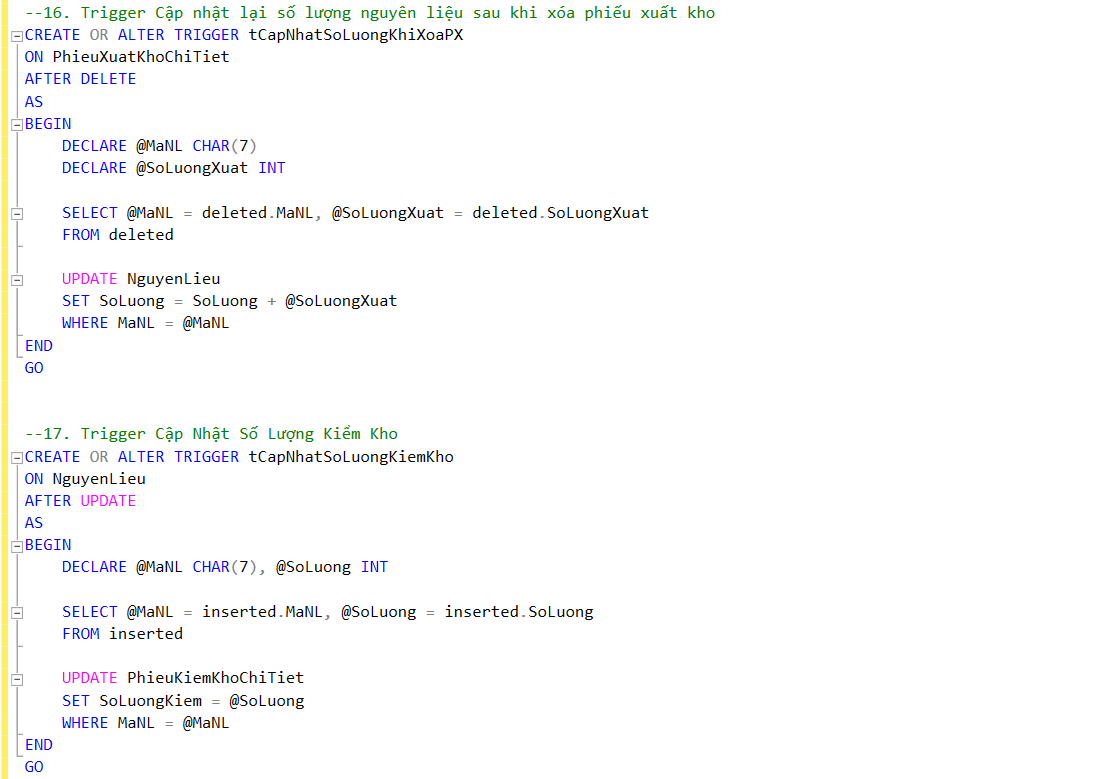
1. **Trigger Cập nhật lại số lượng nguyên liệu sau khi xóa phiếu xuất kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger Cập nhật lại số lượng nguyên liệu sau khi xóa phiếu xuất kho |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | DELETE |
| **Bảng:** | PhieuXuatKho |
| **Process:** | 1. Khi có bản ghi mới được thêm vào PhieuXuatKhoChiTiet, trigger sẽ lấy mã nguyên liệu (MaNL) và số lượng xuất (SoLuongXuat) từ bản ghi mới. 2. Số lượng nguyên liệu tương ứng trong bảng NguyenLieu sẽ được cập nhật bằng cách trừ đi số lượng xuất. |



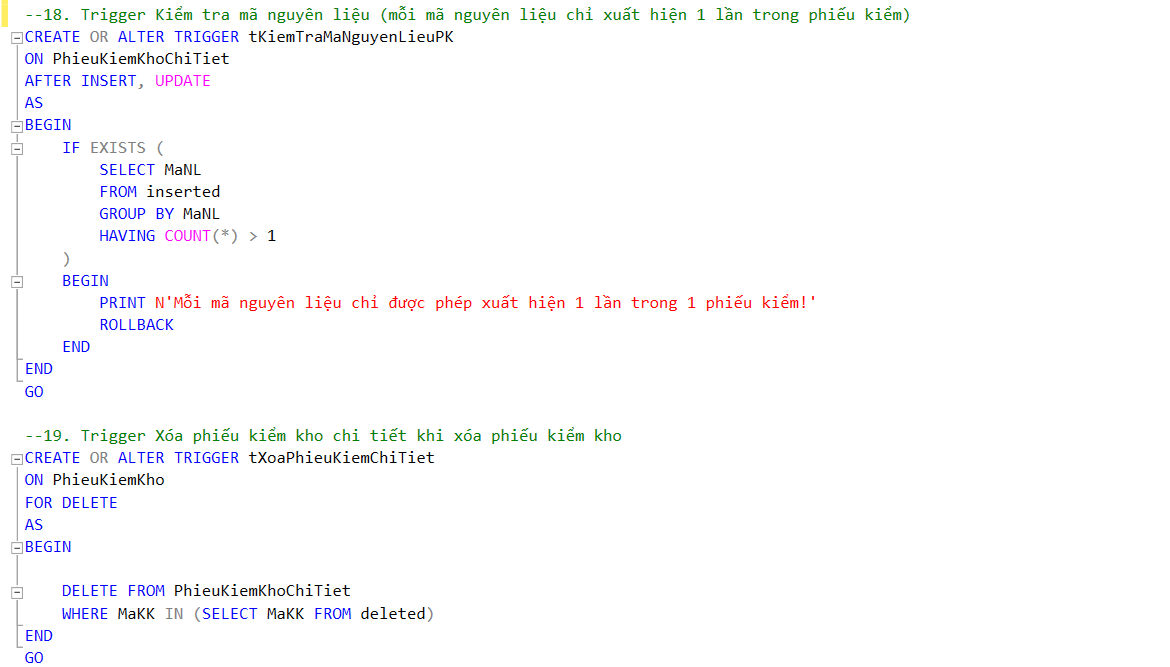
1. **Trigger Cập Nhật Số Lượng Kiểm Kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger Cập nhật số lượng kiểm kho |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | UPDATE |
| **Bảng:** | NguyenLieu |
| **Process:** | 1. Trigger này sẽ tự động cập nhật số lượng kiểm kho (SoLuongKiem) trong bảng PhieuKiemKhoChiTiet theo số lượng hiện tại của nguyên liệu trong bảng NguyenLieu |



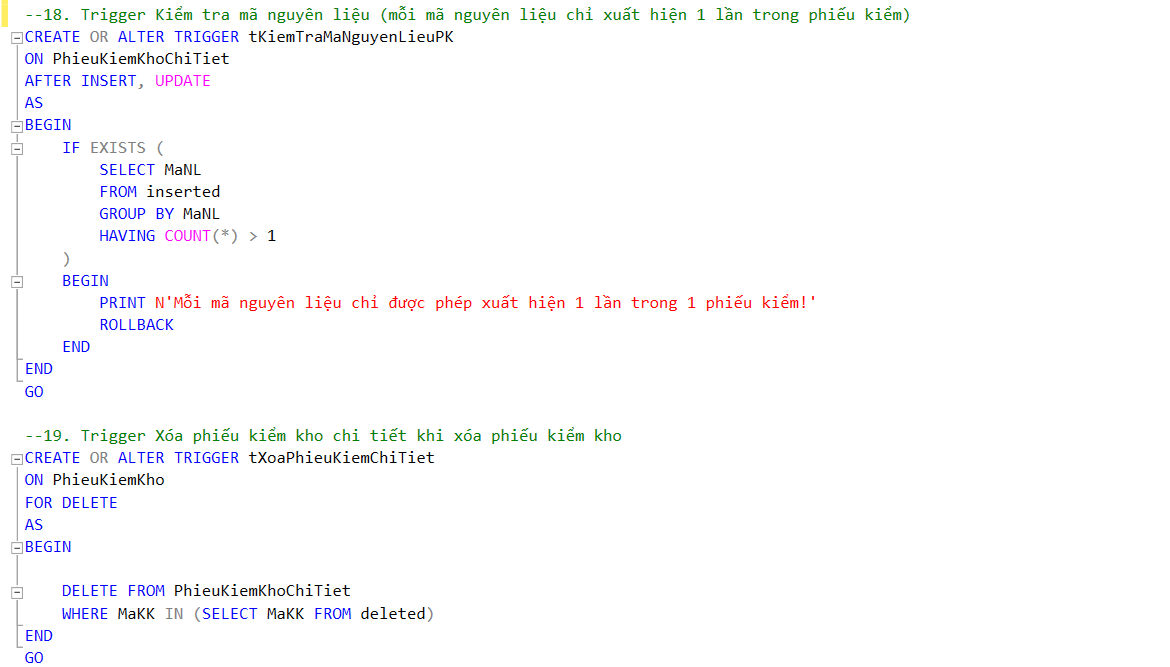
1. **Trigger Kiểm tra mã nguyên liệu (mỗi mã nguyên liệu chỉ xuất hiện 1 lần trong phiếu kiểm)**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger sử dụng để kiểm tra  Trigger kiểm tra xem có mã nguyên liệu nào trong bảng tạm thời inserted (chứa các bản ghi vừa được thêm hoặc cập nhật) xuất hiện nhiều hơn một lần hay không. |
| **Loại trigger:** | AFTER |
| **Sự kiện:** | INSERT, UPDATE |
| **Bảng:** | PhieuKiemtKhoChiTiet |
| **Process:** | 1. Trigger kiểm tra xem có sự trùng lặp mã nguyên liệu trong dữ liệu vừa được thêm vào hoặc cập nhật không. 2. Nếu có bất kỳ mã nguyên liệu nào xuất hiện nhiều hơn một lần (COUNT(\*) > 1), có nghĩa là mã nguyên liệu đã bị nhập trùng lặp, hiển thị thông báo lỗi "Mỗi mã nguyên liệu chỉ được phép xuất hiện 1 lần trong 1 phiếu kiểm và rollback |



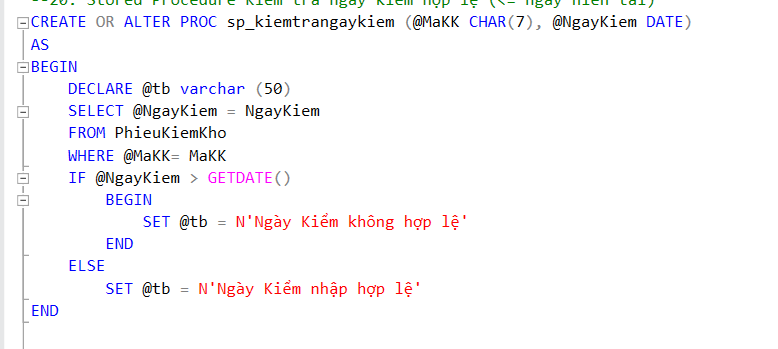
1. **Trigger Xóa phiếu kiểm kho chi tiết khi xóa phiếu kiểm kho**

|  |  |
| --- | --- |
| **Mô tả:** | Trigger giúp đảm bảo rằng khi một phiếu kiểm kho bị xóa, tất cả các bản ghi chi tiết tương ứng trong bảng PhieuKiemKhoChiTiet cũng sẽ được xóa. |
| **Loại trigger:** | FOR |
| **Sự kiện:** | DELETE |
| **Bảng:** | PhieuKiemKho |
| **Process:** | Xóa tất cả các bản ghi trong bảng PhieuKiemKhoChiTiet mà có MaKK (Mã phiếu kiểm kho) tương ứng với những phiếu kiểm kho vừa bị xóa. |



1. **Stored Procedure Kiểm tra ngày kiểm hợp lệ**

|  |  |
| --- | --- |
| **Mô tả:** | Stored ProcedureDựa trên mã kiểm kho  hàm sẽ kiểm tra ngày  kiểm có lớn hơn ngày hiện tại hay không (getdate()). |
| **Input:** | @NgayKiem |
| **Output:** | Giá trị dạng chuỗi |
| **Process:** | 1. Nhập liệu ngày kiểm kho 2. Kiểm tra ngày  kiểm có lớn hơn ngày hiện tại hay không (getdate()). Nếu có, thông báo ‘ Ngày kiểm không hợp lệ’ 3. Ngược lại thì thông báo ‘Ngày kiểm hợp lệ’ |



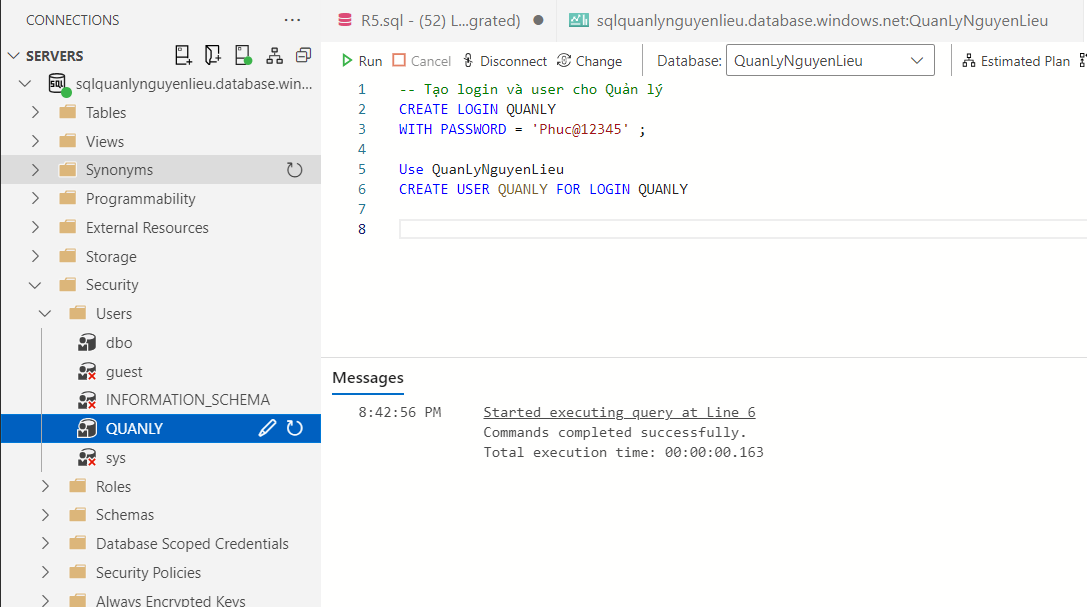
1. **Xây dựng cơ chế bảo mật (R5)**

|  |  |  |
| --- | --- | --- |
| **Nhóm** | **Quản lý** | **Nhân viên** |
| **Quyền** | Quản lý có toàn quyền quản lý truy cập vào database | • Được thêm, sửa,xem bảng: PhieuNhapKho, PhieuXuatKho, PhieuKiemKho, NguyenLieu.  • Không được xoá tất cả các bảng |

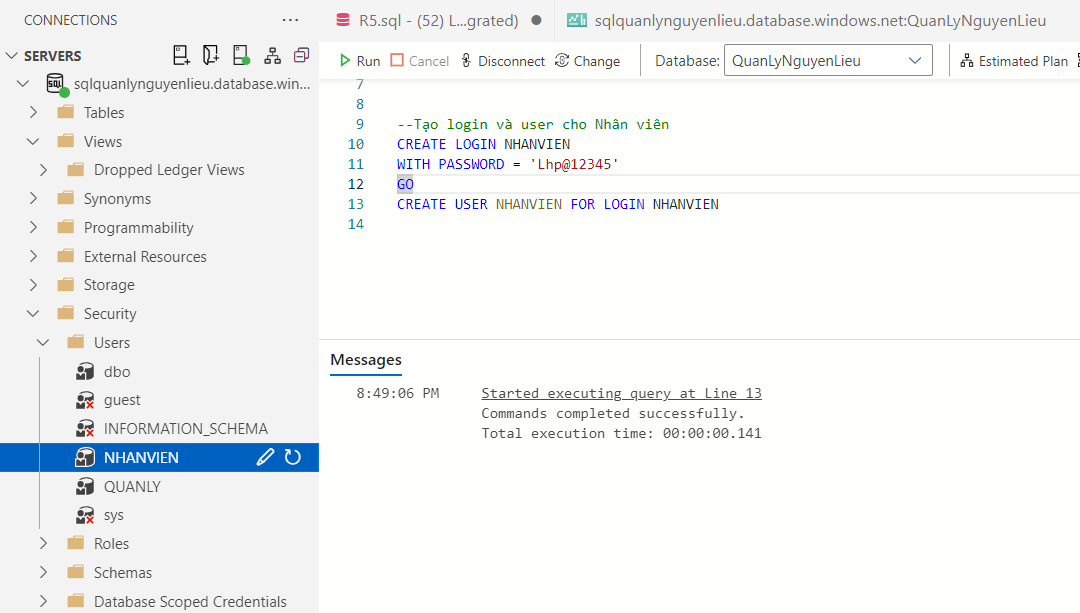
* **Thiết lập cơ chế authentication phù hợp**
* Ở cấp độ Server:
* Window Anthentication: Đăng nhập bằng ID và password  của Window
* SQL Server Authentication: Đăng nhập bằng User/Password người dùng tự tạo trong SQL Server
* Ở cấp độ database:
* Sử dụng SQL Server Authentication: tạo 2 user QUANLY và NHANVIEN

và 2 user chỉ được sử dụng trong database QuanLyNguyenLieu

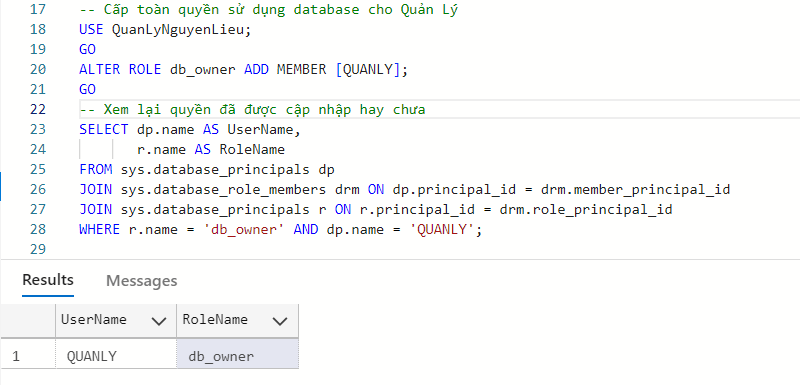
* Tạo login và user cho Quản lý, user chỉ được sử dụng cho database QuanLyNguyenLieu



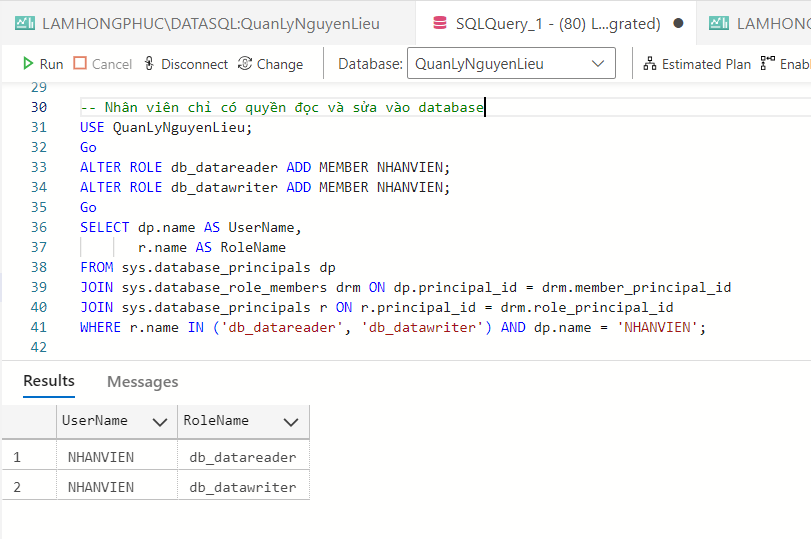
* Tạo login và user cho Nhân viên. user chỉ được sử dụng cho database QuanLyNguyenLieu



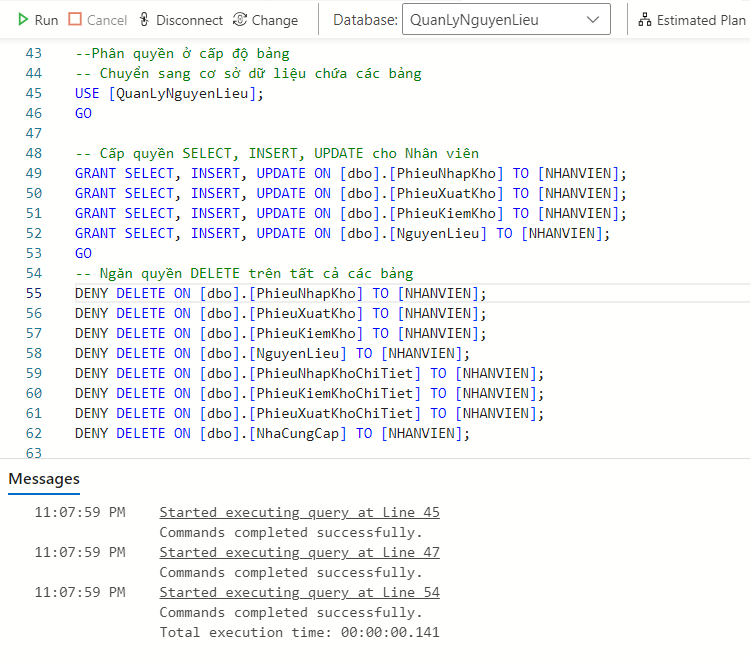
* **Tạo và phân quyền phù hợp cho người dùng (ở cấp độ server và cấp độ database)**
* Ở mức độ database
* Quản lý được phân quyền ở cấp độ db\_owner



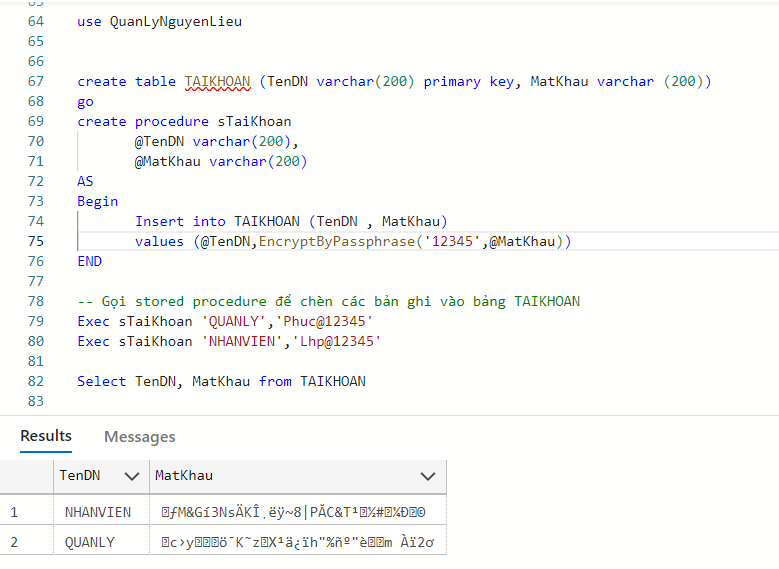
* Nhân viên được phân quyền ở cấp độ db\_datareader và db\_datawriter



* Ở mức độ bảng

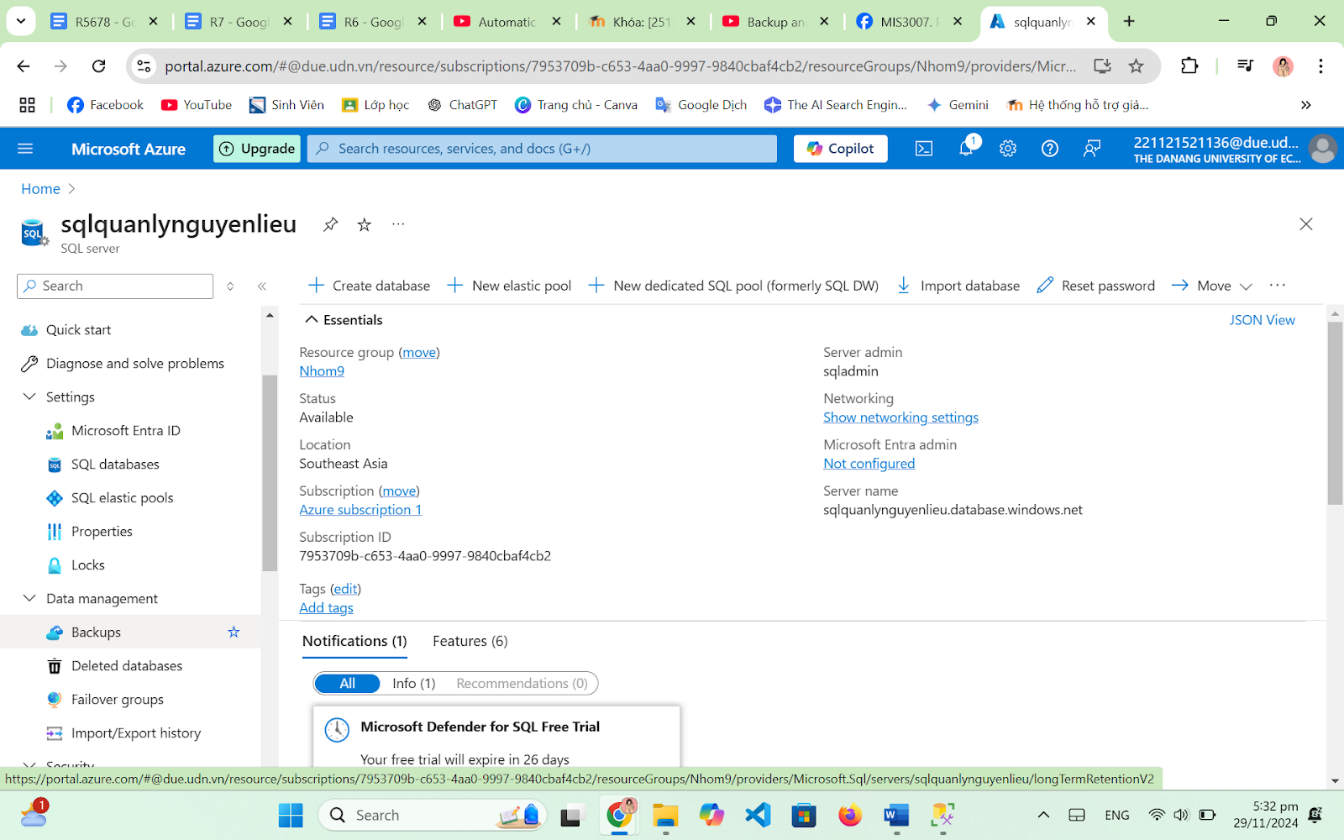


* **Cơ chế bảo mật: thực hiện mã hoá mật khẩu tự động**
* Cơ chế mã hóa bằng passphrase giúp bảo vệ mật khẩu trong cơ sở dữ liệu một cách hiệu quả, dễ triển khai và phù hợp với các ứng dụng yêu cầu mức độ bảo mật vừa phải.
* Cách triển khai:
* Tạo bảng TAIKHOAN, có 2 cột Tên Đăng Nhập (primary key) và Mật Khẩu
* Khi thực thi stored procedure sTaiKhoan, hàm EncryptByPassphrase nhận vào passphrase '12345' và mật khẩu @MatKhau.
* Hàm này sử dụng passphrase để tạo một khóa mã hóa đối xứng.
* Mật khẩu được mã hóa bằng khóa này, kết quả là dữ liệu nhị phân đã mã hóa được lưu vào cột MatKhau.

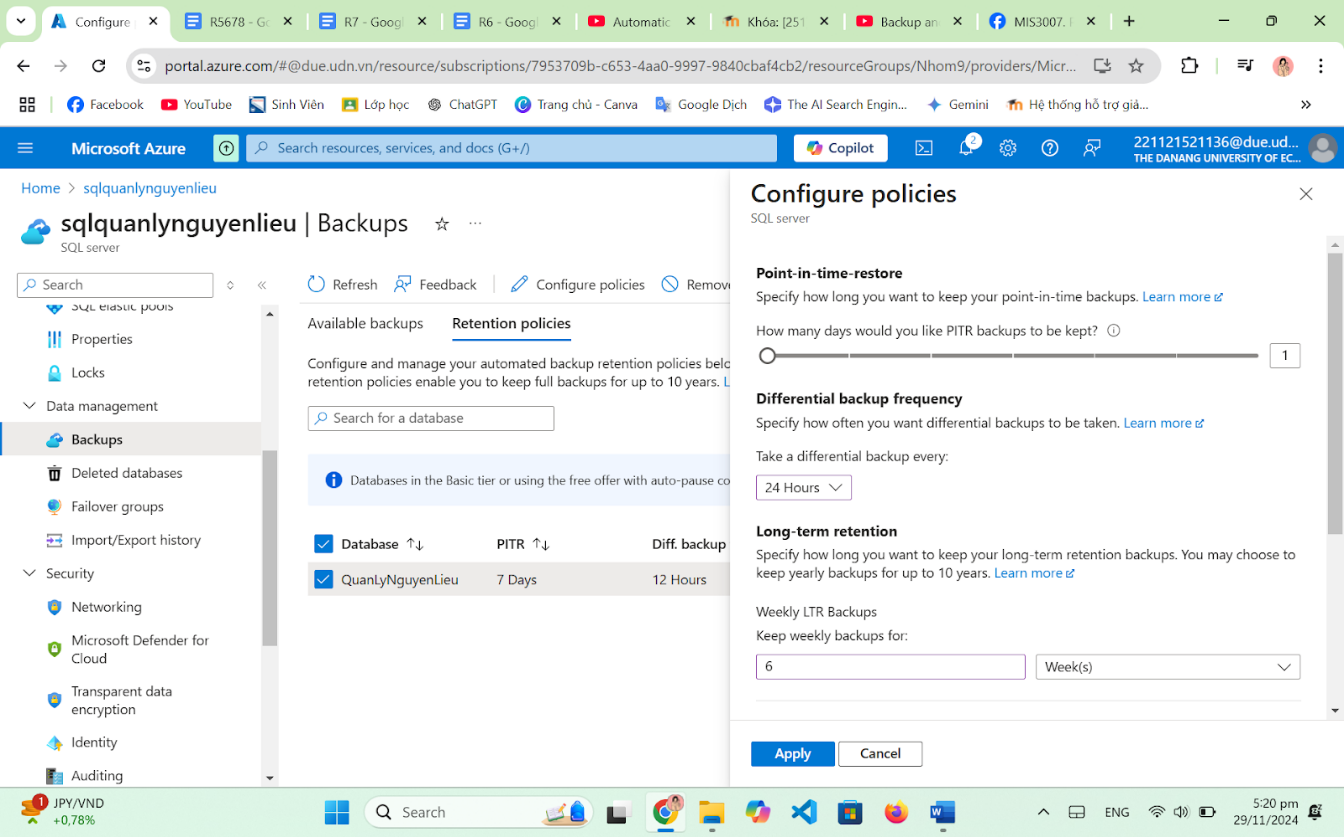


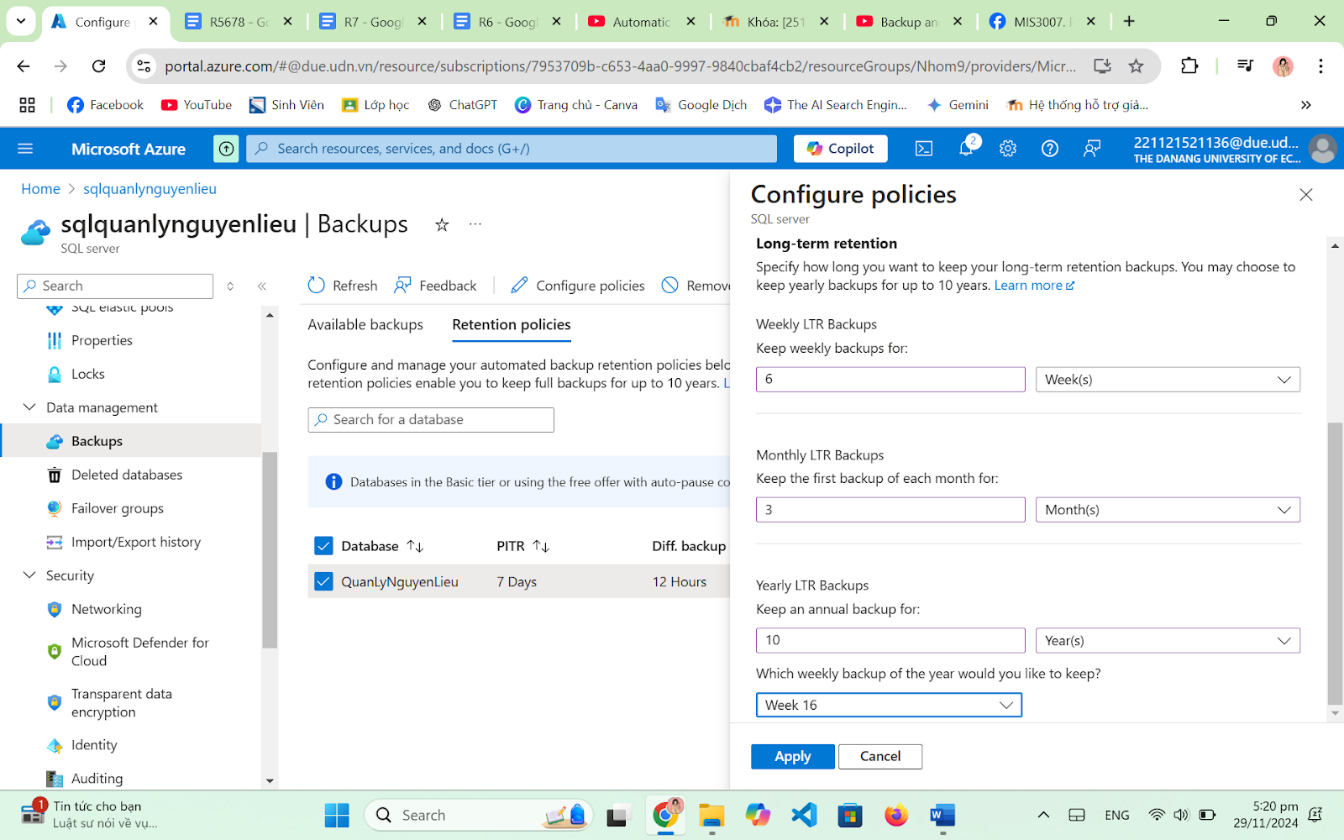
1. **Xây dựng cơ chế backup dữ liệu tự động (R6)**

* **Backup dữ liệu trên Microsoft Azure**
* Truy cập vào Server name đã tạo, trong Data management, chọn Backups

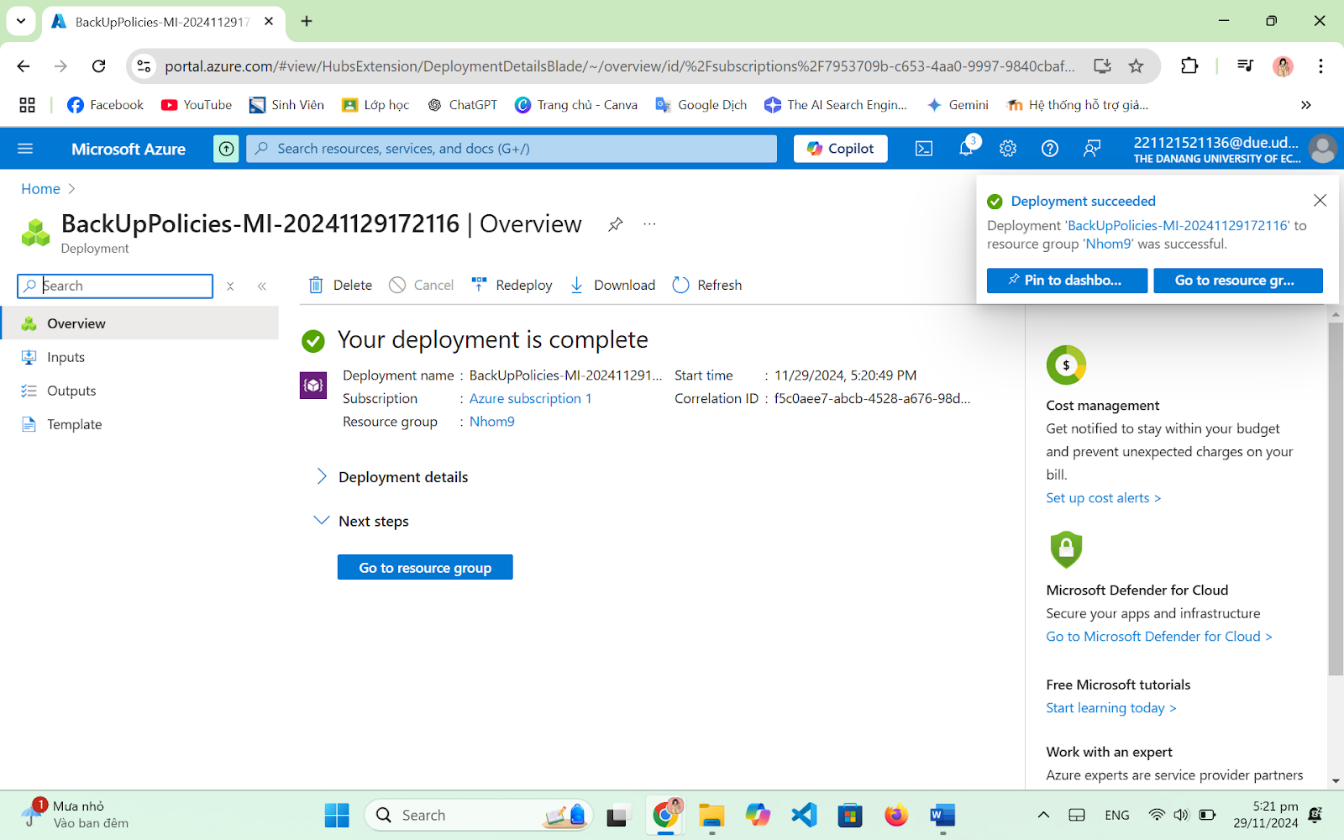
****

* Chọn Retention policies 🡪 chọn database QuanLyNguyenLieu 🡪 Chọn Configure policies

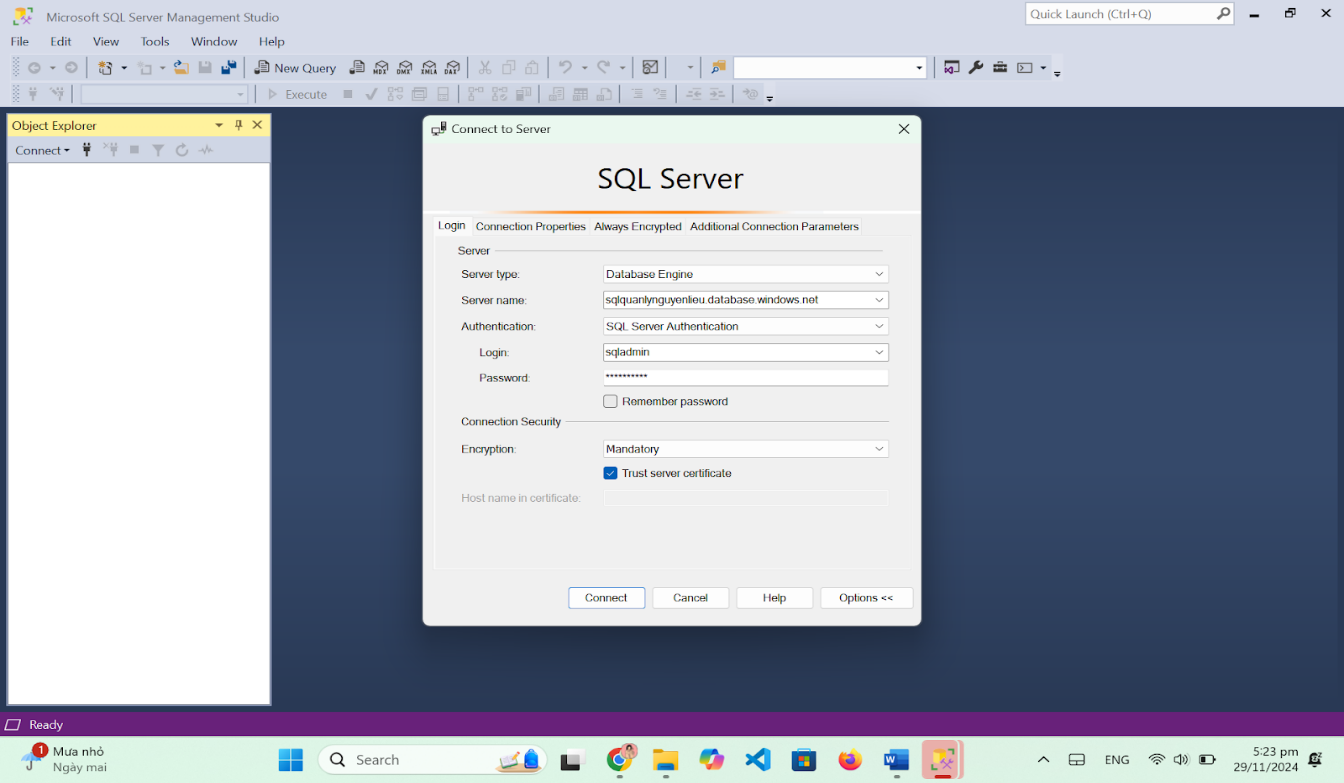




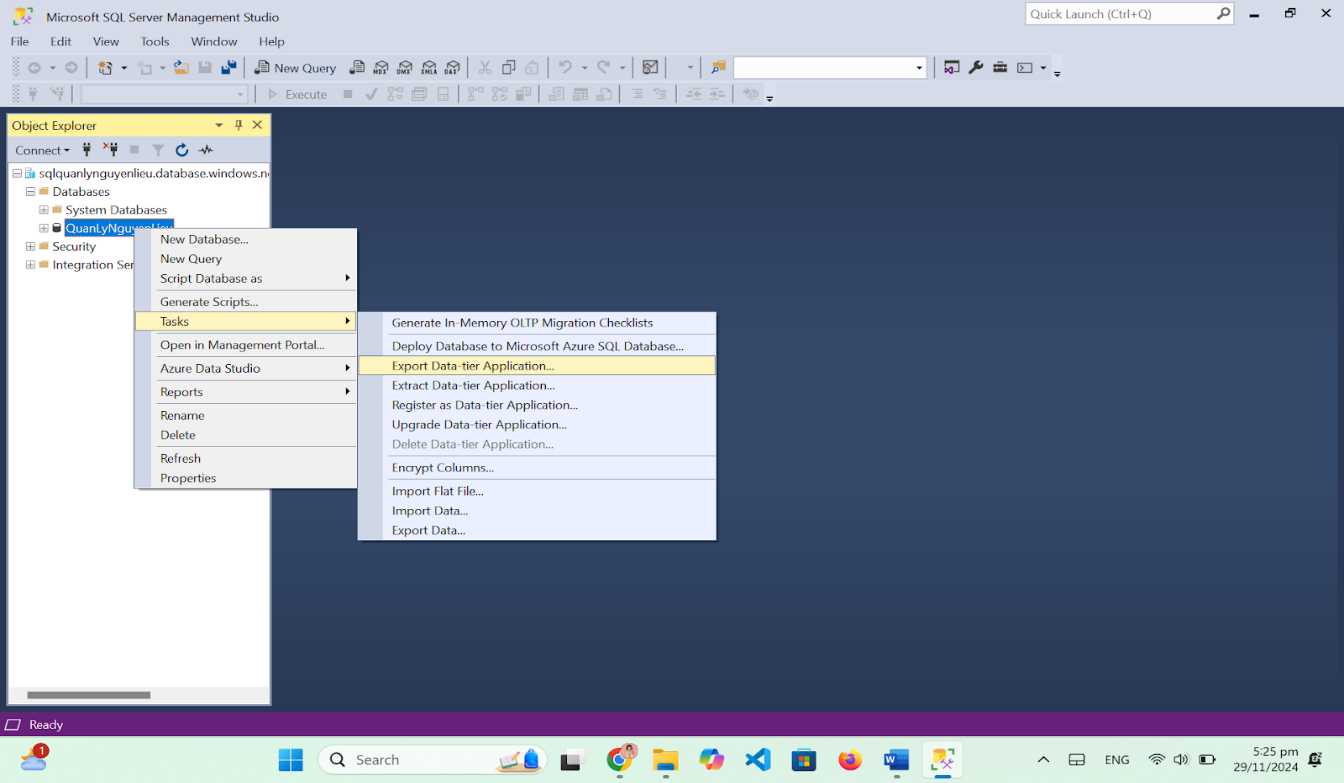
* Apply Backup thành công



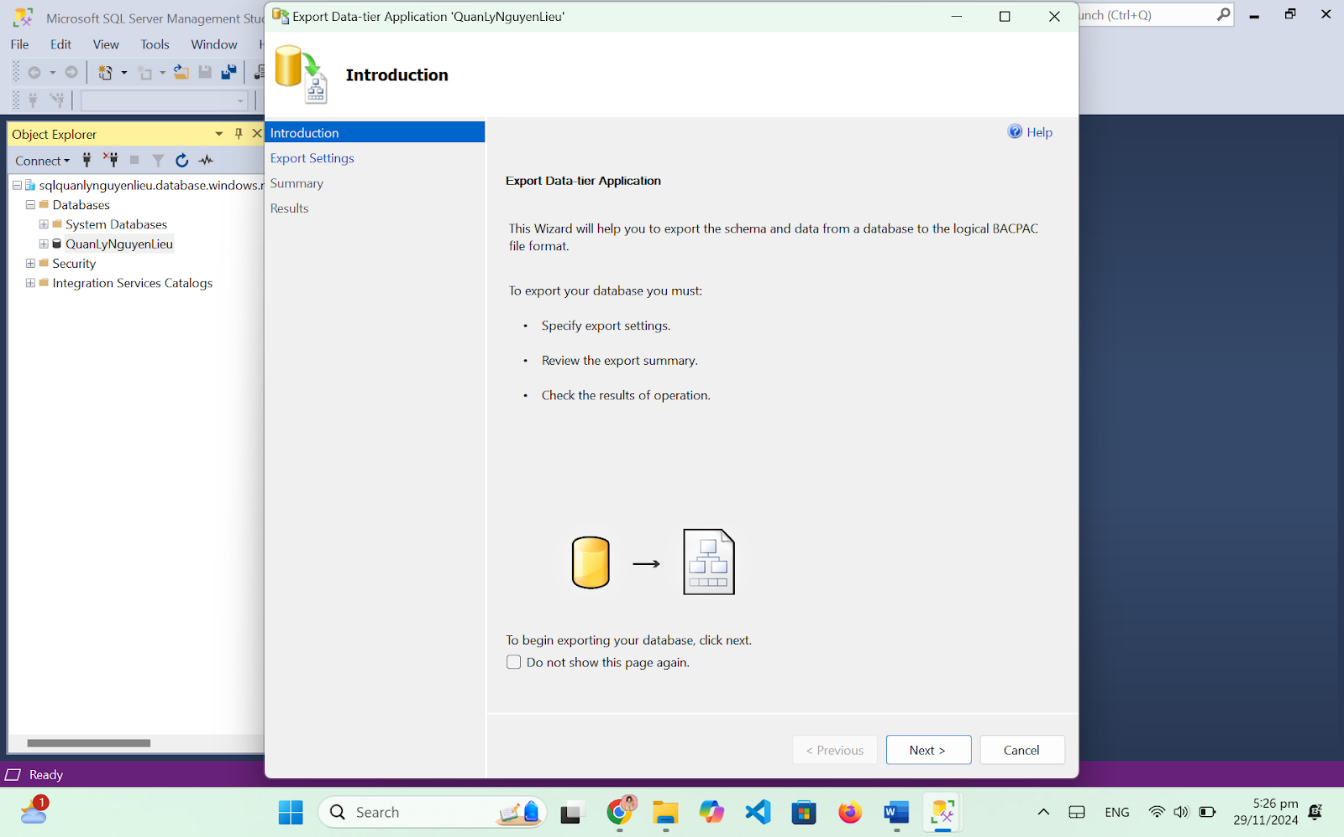
* Đăng nhập vào SQL Server

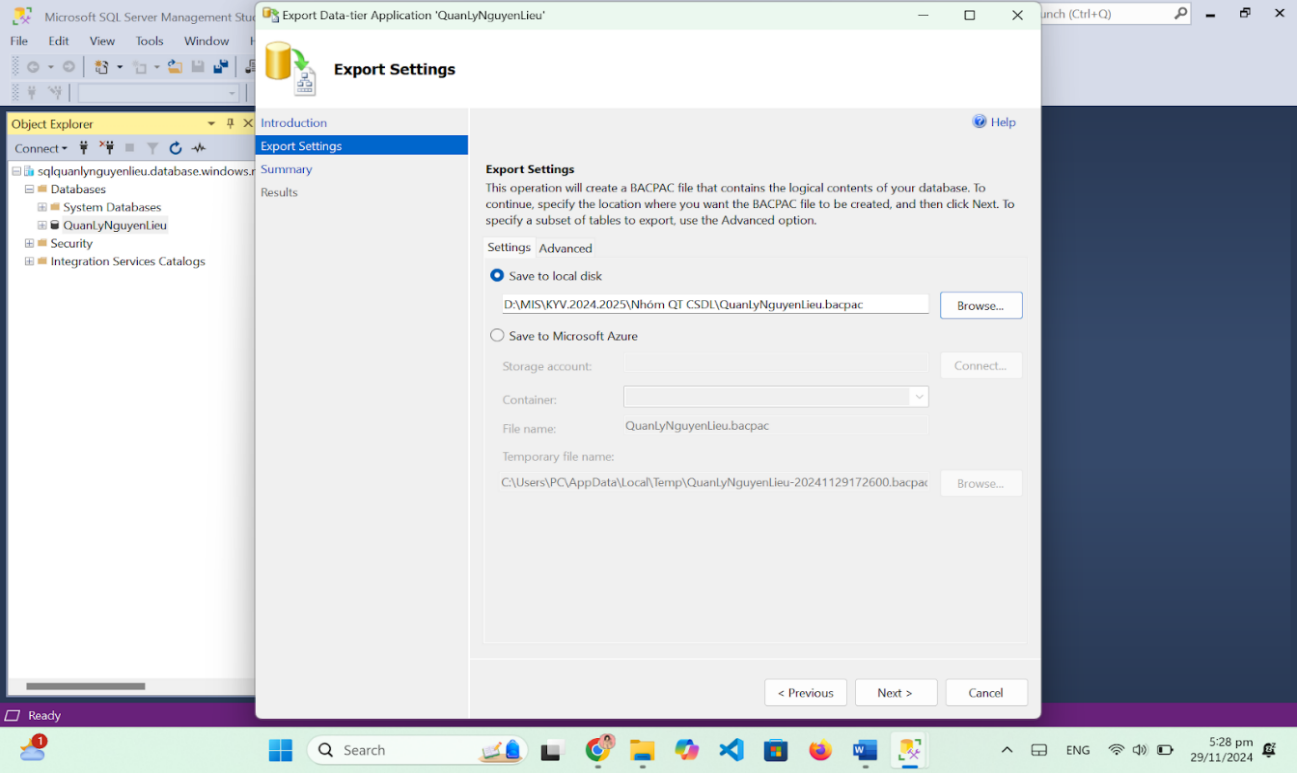


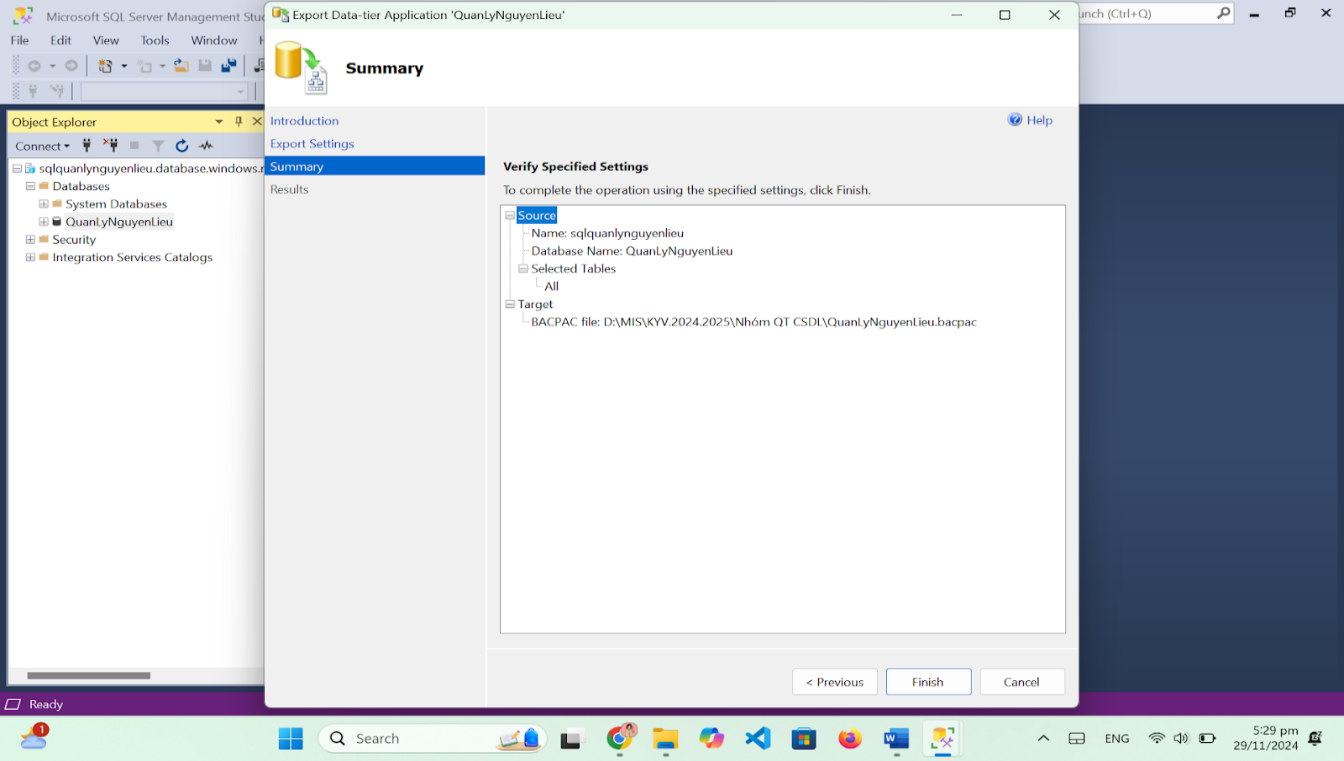
* Trong database QuanLyNguyenLieu 🡪 Task 🡪 Export Data – tier Application



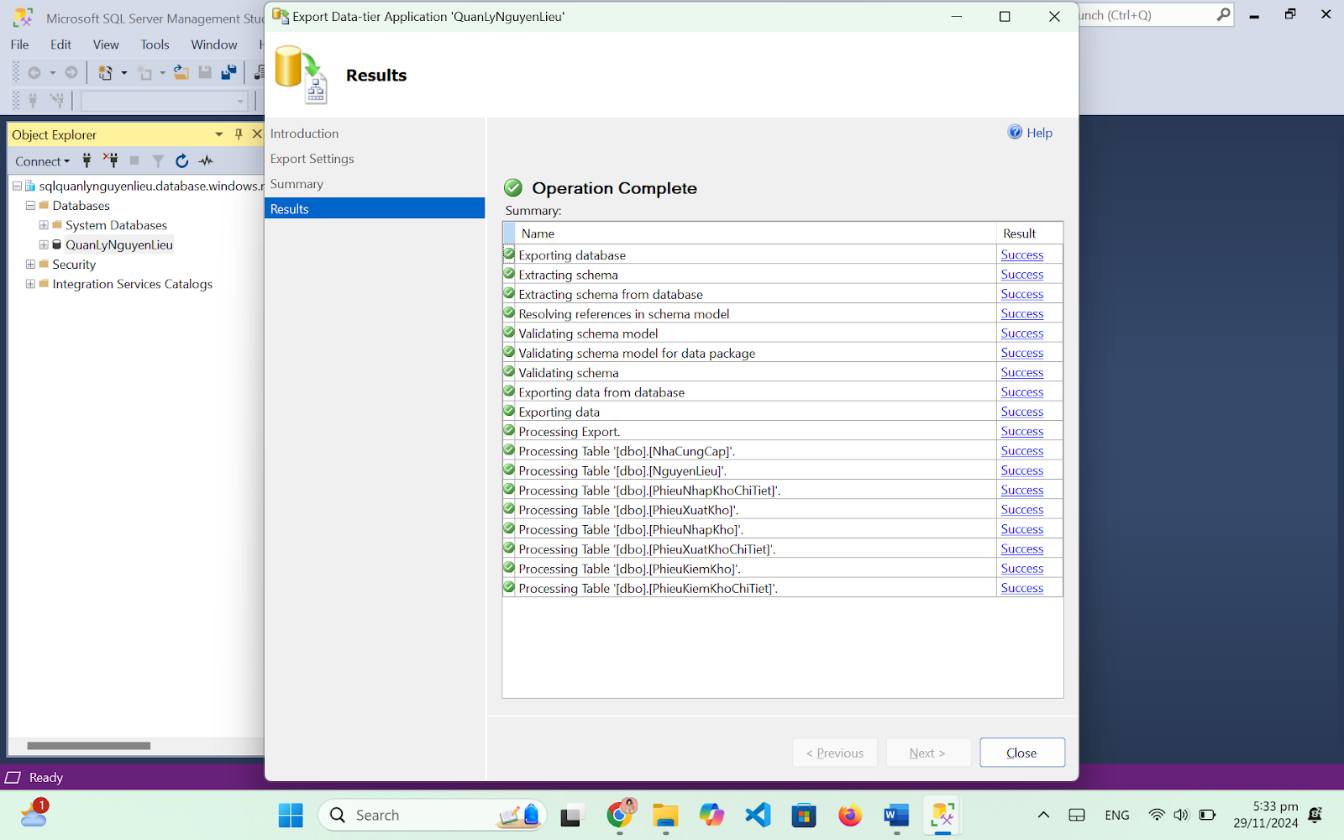
* Chọn vị trí lưu file



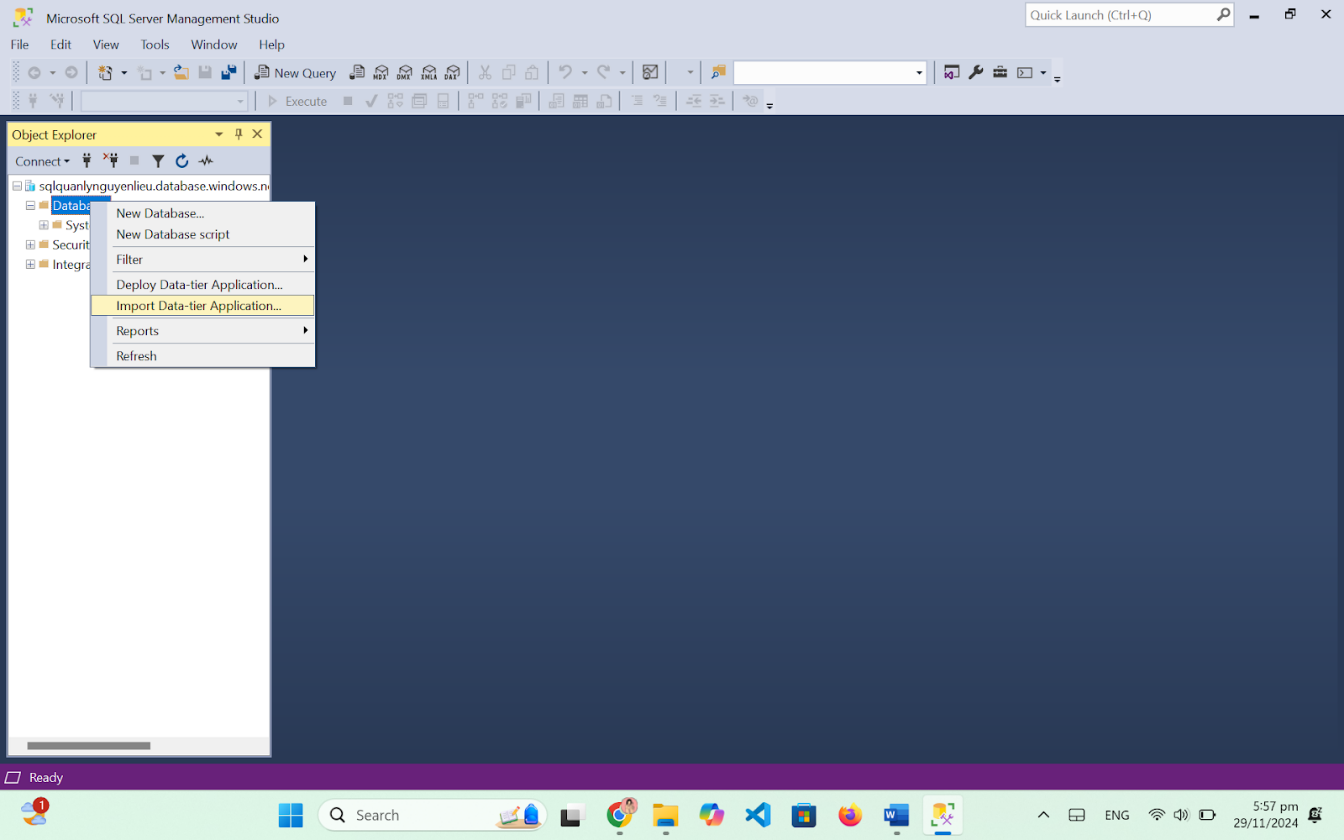




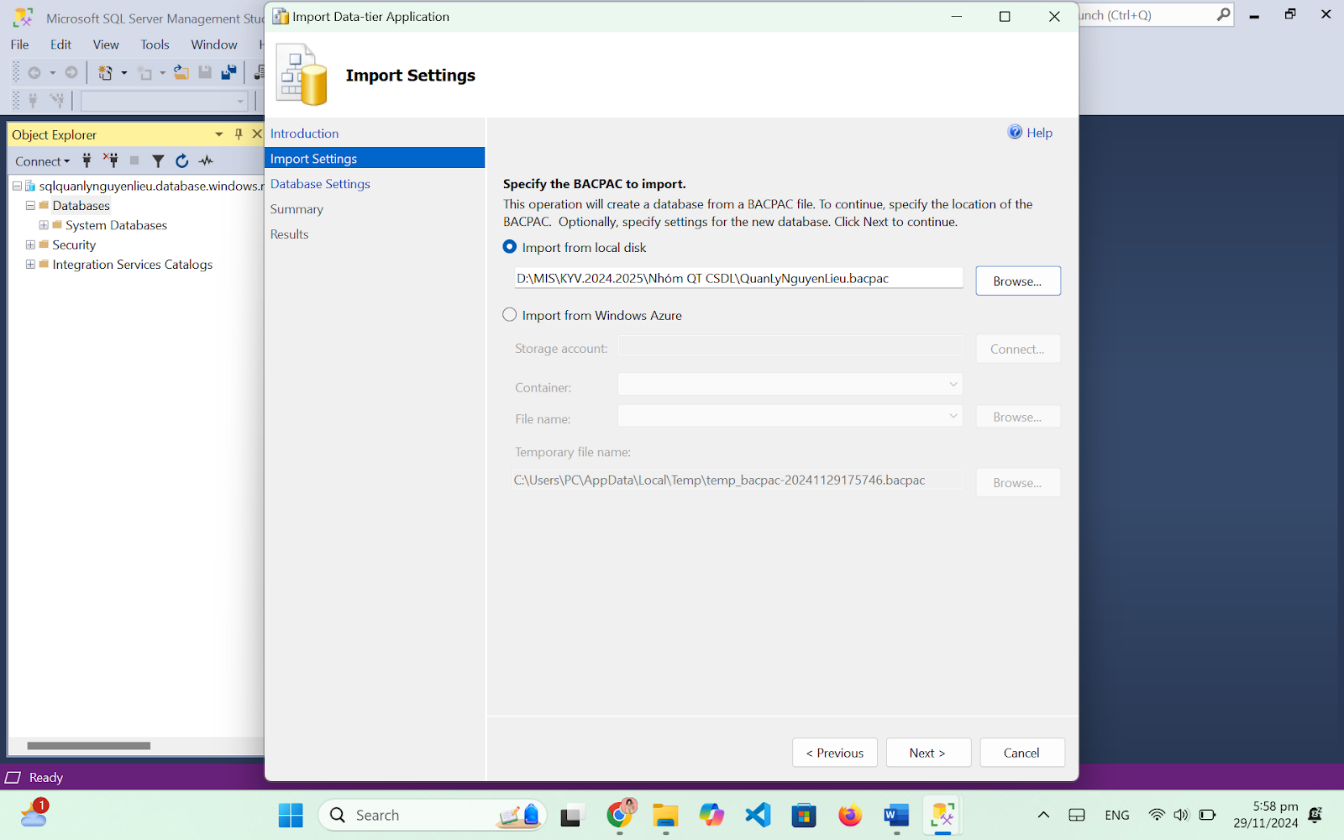
* Backup hoàn tất



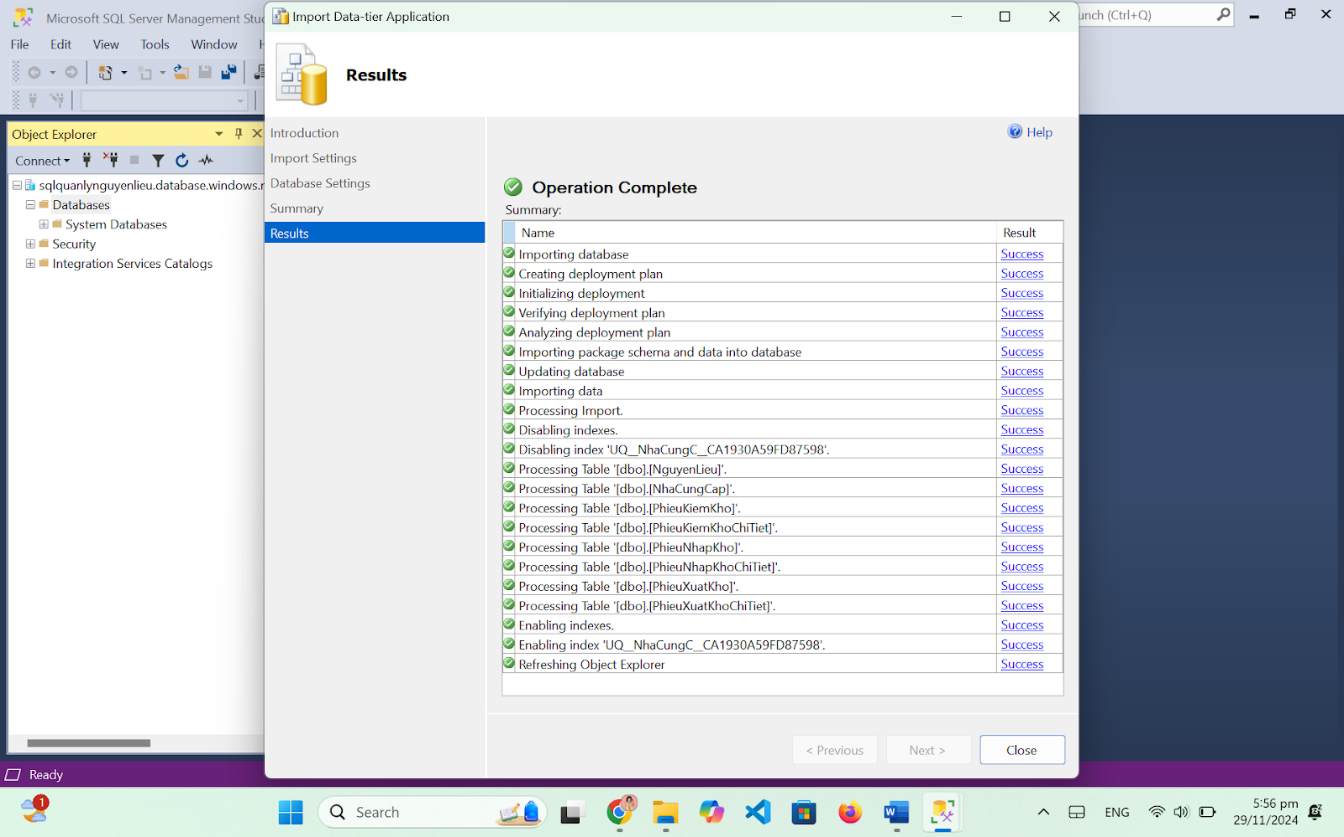
* **Restore dữ liệu**
* Trong Database, chọn Import Data – tier Application



* Chọn file cần Restore 🡪 Next



* Hoàn tất Restore



1. **Giải pháp dữ liệu lớn (R7)**

Khi khối lượng dữ liệu vượt quá khả năng lưu trữ của các thiết bị truyền thống,

Microsoft Azure trở thành giải pháp hiệu quả. Đây là một nền tảng điện toán đám mây của Microsoft, cung cấp khả năng truy cập và quản lý các dịch vụ, tài nguyên đám mây một cách dễ dàng và tiện lợi. Microsoft Azure là một lựa chọn đáng tin cậy, mang lại hiệu quả cao và phù hợp với doanh nghiệp quy mô nhỏ như Lechat Coffee.

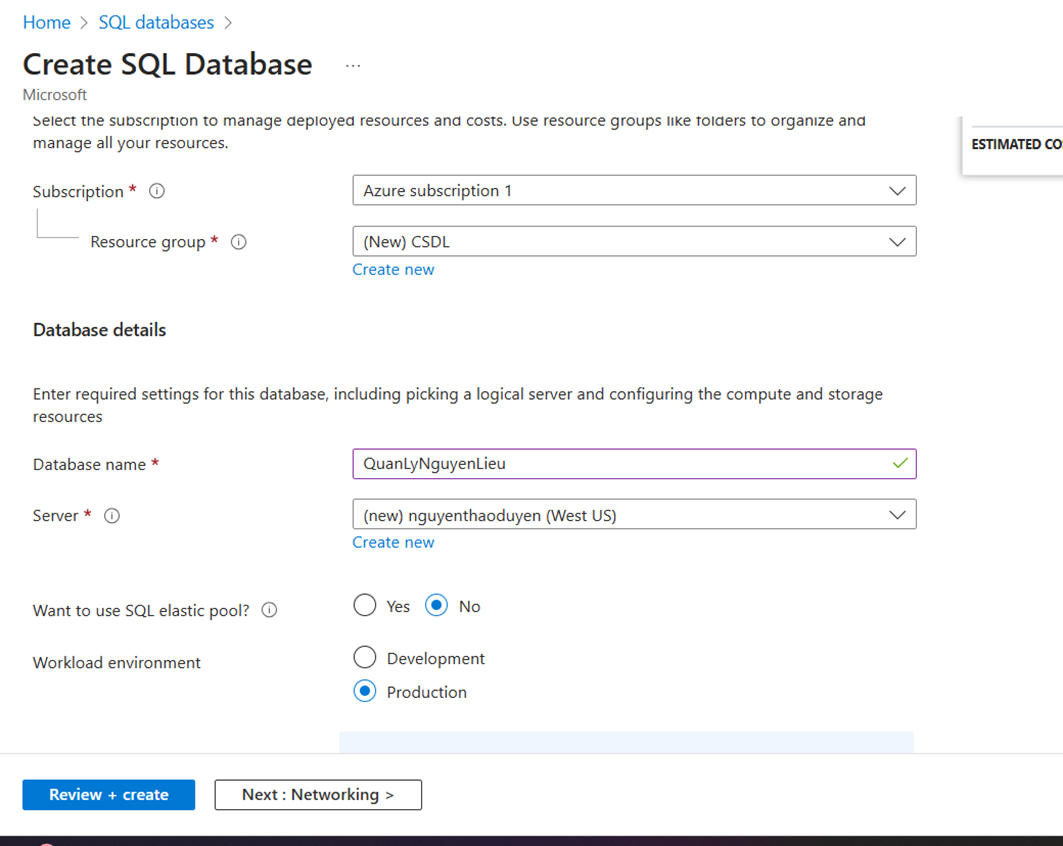
Những lý do cụ thể để chọn Microsoft Azure:

* Khả năng mở rộng linh hoạt: hỗ trợ tăng/giảm tài nguyên linh hoạt giúp đảm bảo hiệu suất khi dữ liệu và số lượng tăng
* Tiết kiệm chi phí: Với quy mô nhỏ như Lechat Coffee, Microsoft Azure cung cấp các gói dịch vụ phù hợp với ngân sách, đảm bảo tính kinh tế mà vẫn đáp ứng đầy đủ nhu cầu.
* Tính linh hoạt cao: Azure hỗ trợ sao lưu dữ liệu trên nhiều ngôn ngữ lập trình và hệ điều hành khác nhau, giúp dễ dàng tích hợp vào mọi hệ thống hiện có.
* Hỗ trợ SQL Server: Azure tích hợp tốt với SQL Server, cho phép thực hiện các thao tác sao lưu, khôi phục hoặc di chuyển dữ liệu từ môi trường cục bộ lên đám mây một cách đơn giản.
* Tùy chỉnh sao lưu tự động: Người dùng có thể linh hoạt thiết lập quy trình sao lưu tự động theo nhu cầu, giúp tối ưu hóa thời gian và tài nguyên.
* Bước 1: Tạo tài khoản trên Microsoft Azure

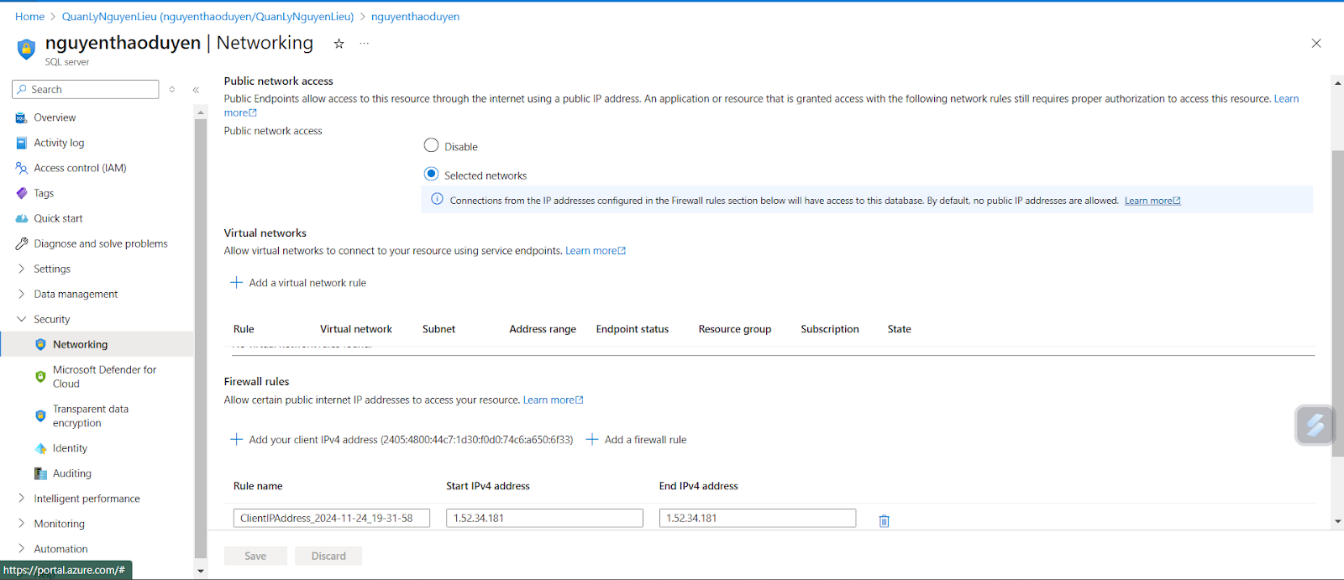
Chọn SQL Database => Create

* Bước 2: Thiết lập New sever

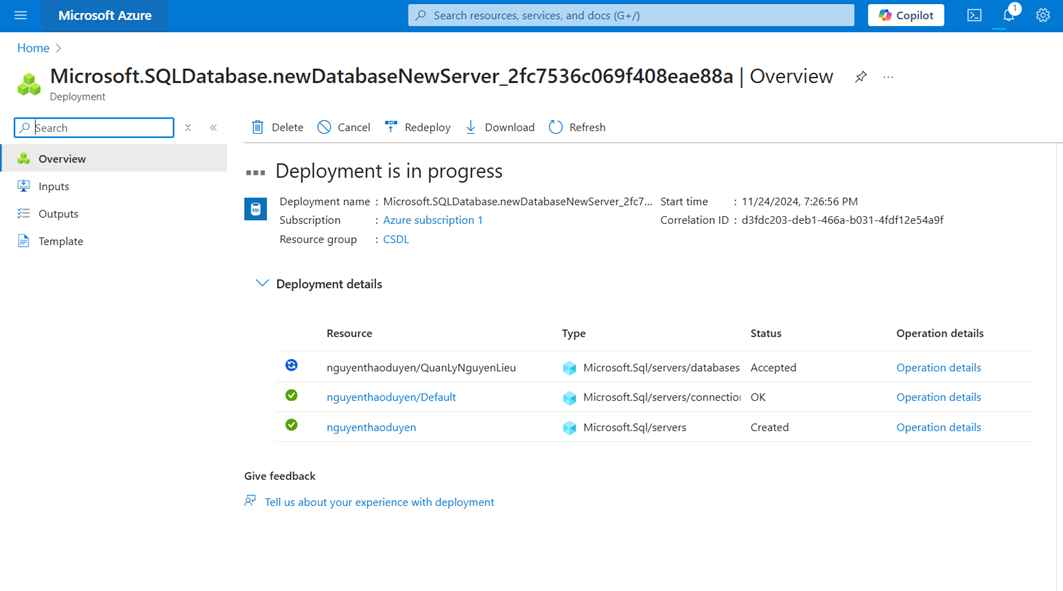
Điền thông tin cần thiết để Create SQL Database => Review+create

****

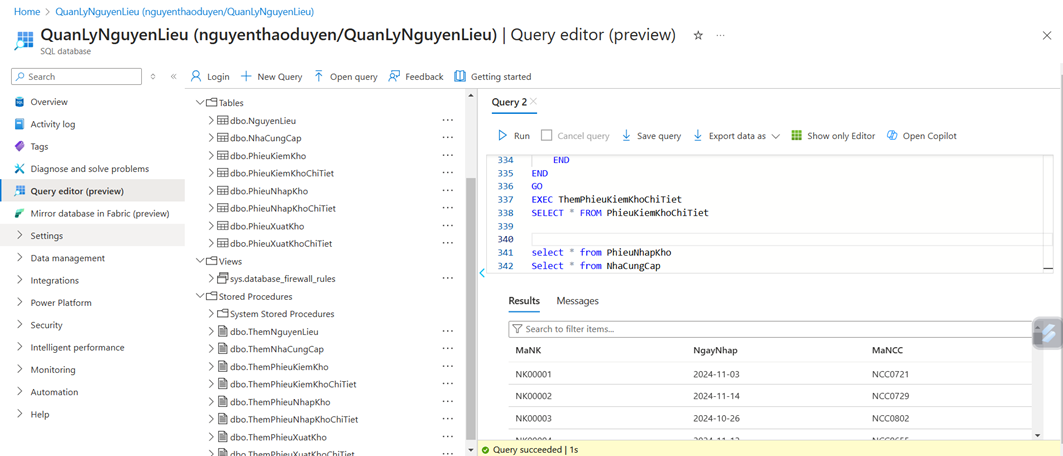
* Bước 3: Thiết lập Networking và Firewall rules cho Database => Nhấn save



* Bước 4: Tạo thành công cơ sở dữ liệu

****

* Bước 5: Sau khi Database được tạo, chúng ta có thể sử dụng Query editor trên Azure để thực hiện các thao tác cơ bản như : thêm dữ liệu, tìm kiếm dữ liệu,...

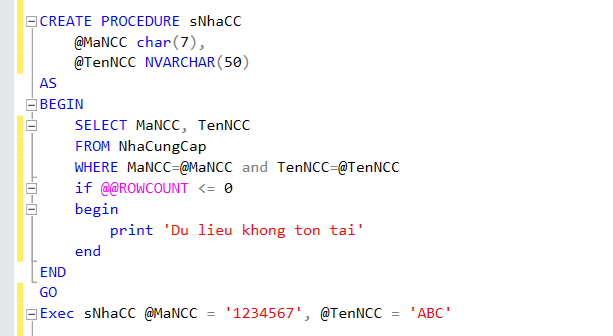
****

1. **Phương án để hạn chế tấn công SQL Injection (R8)**

SQL Injection là một kỹ thuật tấn công lợi dụng những sơ hở trong cách ứng dụng xử lý các câu truy vấn SQL. Kẻ tấn công có thể chèn mã SQL độc hại vào đầu vào của ứng dụng, từ đó thay đổi hoặc thao tác với câu truy vấn gốc, qua đó khai thác hoặc làm hỏng dữ liệu trong cơ sở dữ liệu. Việc này có thể dẫn đến việc chiếm quyền truy cập trái phép, xóa hoặc đánh cắp dữ liệu quan trọng, hoặc thậm chí chiếm quyền điều khiển toàn bộ hệ thống.

Để bảo vệ hệ thống khỏi các cuộc tấn công SQL Injection, nhóm đã quyết định sử dụng stored procedures làm phương án phòng ngừa. Việc sử dụng stored procedures giúp tách biệt mã SQL khỏi dữ liệu đầu vào, từ đó giảm thiểu nguy cơ tấn công. Các stored procedures có thể xử lý toàn bộ các tham số đầu vào, làm cho các lệnh SQL không bị thay đổi bởi dữ liệu nhập vào và tránh được các cuộc tấn công SQL Injection.

Khi hệ thống gặp lỗi, thay vì hiển thị thông tin chi tiết về lỗi, chúng tôi chỉ thông báo lỗi chung chung. Điều này giúp ngăn chặn kẻ tấn công từ việc khai thác thông tin lỗi để tìm hiểu cấu trúc cơ sở dữ liệu của hệ thống.



Bên cạnh đó, việc áp dụng các biện pháp bảo mật khác như mã hóa dữ liệu và phân quyền truy cập rõ ràng cũng đóng vai trò quan trọng trong việc bảo vệ cơ sở dữ liệu. Mã hóa dữ liệu giúp đảm bảo rằng ngay cả khi hacker truy cập vào cơ sở dữ liệu, họ cũng không thể đọc được các thông tin nhạy cảm. Đồng thời, việc phân quyền truy cập chi tiết cho từng đối tượng người dùng, ví dụ như phân quyền cho quản lý và nhân viên, giúp kiểm soát và giới hạn quyền truy cập, hạn chế việc truy cập trái phép vào các dữ liệu quan trọng.