

TRƯỜNG ĐẠI HỌC KINH TẾ
ĐẠI HỌC ĐÀ NẴNG
KHOA THỐNG KÊ – TIN HỌC

~*~*~



BÁO CÁO BÀI TẬP NHÓM

Hệ Thống Quản Lý Quán HighLands Coffee

Cơ sở 1, 93-95 Nguyễn Văn Thoại, Q. Sơn Trà, Tp. Đà Nẵng

Môn: Quản trị cơ sở dữ liệu

GVHD: Cao Thị Nhâm

SVTH: NHÓM 7 – 48K14.2

Đinh Sỹ Quốc Doanh

Cao Văn Khánh Duy

Đồng Thị Ngọc Dương

Lê Phan Hồng Khanh

Đà Nẵng, 09/2024

Ghi chú thay đổi:

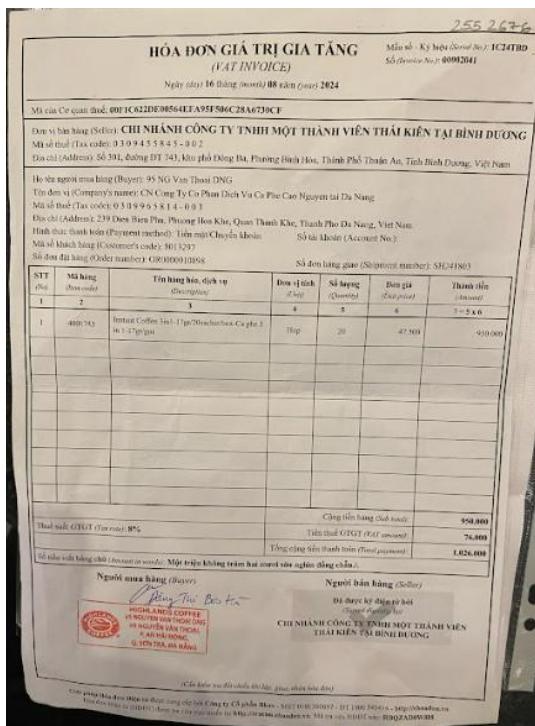
MỤC LỤC

I.	Hoá đơn nhóm chọn	1
II.	Thiết kế cơ sở dữ liệu:	1
II.1.	Thiết kế cơ sở dữ liệu ở mức khái niệm:	1
II.1.1.	ER-Bán:	1
II.1.2.	ER-Nhập:	2
II.1.3.	Tích hợp các ER:	2
II.2.	Thiết kế cơ sở dữ liệu ở mức logic:	3
II.2.1.	Chuyển thực thể, thu được:.....	3
II.2.2.	Chuyển quan hệ, thu được:	3
II.2.3.	Sơ đồ quan hệ	3
II.3.	Thiết kế cơ sở dữ liệu ở mức vật lý:	4
III.	Xây dựng cơ sở dữ liệu:	6
III.1.	Module tạo dữ liệu dump cho các bảng:	6
III.1.1.	Bảng nhân viên (NhanVien):.....	6
III.1.2.	Bảng Nhà cung cấp (NhaCungCap):.....	8
III.1.3.	Bảng Sản Phẩm (SanPham):	9
III.1.4.	Bảng Nguyên Liệu (NguyenLieu):.....	10
III.1.5.	Bảng Phiếu nhập (PhieuNhap):	11
III.1.6.	Bảng Chi tiết phiếu nhập (ChiTietPhieuNhap):	13
III.1.7.	Bảng Hóa đơn (HoaDon):	14
III.1.8.	Bảng Chi tiết hóa đơn (ChiTietHoaDon):	15
III.2.	Module phục vụ thao tác xử lý dữ liệu	16
III.2.1.	Module 1 - Kiểm tra sự tồn tại của nguyên liệu:.....	16
III.2.2.	Module 2 - Kiểm tra mức tồn kho và trả về cảnh báo:.....	17

III.2.3. Module 3 - Cập nhật giá nhập mới của nguyên liệu:	18
III.2.4. Module 4 - Tự động cập nhật các tổng tiền phiếu nhập khi thêm mới thông tin chi tiết phiếu nhập:	19
III.2.5. Module 5 – Tự động cập nhật số lượng nguyên liệu sau khi thêm chi tiết phiếu nhập:	20
III.2.6. Module 6 - Tự động cập nhật tổng thành tiền cho hoá đơn khi thêm mới thông tin chi tiết hoá đơn:	20
III.2.7. Module 7 - Tính doanh thu dựa vào bảng hoá đơn trong mốc thời gian tuỳ chọn:21	
III.2.8. Module 8 - Tính chi phí nhập hàng dựa vào bảng phiếu nhập trong mốc thời gian tuỳ chọn:	22
III.2.9. Module 9 - Tự động lưu vết Hoá đơn lỗi	23
III.2.10. Module 10 - Tự động cập nhật giảm số lượng nguyên liệu tồn kho khi tạo mới chi tiết hoá đơn:.....	24
IV. Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu:	25
IV.1. Phân quyền cho các nhóm đối tượng:.....	25
IV.2. Phân quyền cho quản lý:	25
IV.3. Phân quyền cho nhân viên:	30
IV.4. Minh chứng:.....	36
V. Backup dữ liệu tự động	37
V.1. Tạo cơ chế Backup dữ liệu tự động (AutoBackup)	38
V.2. Tạo cơ chế xóa dữ liệu đã Backup	45
V.3. Set thời gian thực hiện Plan	47
V.4. Lưu Plan và Execute	49
V.5. Dừng hoạt động backup tự động.....	51
VI. Phương án giải quyết TỐT NHẤT cho lượng dữ liệu rất lớn, không đủ lưu trữ trong một ổ đĩa:	52

VI.1. Lý do lựa chọn Partitioning	52
VI.2. Cách hosting Server trong mạng LAN/ WIFI và kết nối Server:	53
VI.3. Cách phân tán dữ liệu cho Server con và liên kết ngược lại:.....	61
VII. Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection:.....	85
VII.1. SQL Injection là gì?	85
VII.2. Các giải pháp hạn chế tấn công SQL Injection:.....	85
VII.2.1. Sử dụng Stored Procedures (Thủ tục lưu trữ):.....	85
VII.2.1.1. Stored Procedure cho bảng NhanVien:	85
VII.2.1.2. Stored Procedure cho bảng SanPham:	86
VII.2.1.3. Stored Procedure cho bảng HoaDon:	87
VII.2.1.4. Stored Procedure cho bảng PhieuNhap:.....	88
VII.2.1.5. Stored Procedure cho bảng ChiTietHoaDon:.....	89
VII.2.1.6. Stored Procedure cho bảng ChiTietPhieuNhap:	90
VII.2.1.7. Stored Procedure cho bảng NhaCungCap:.....	92
VII.2.1.8. Stored Procedure cho bảng NguyenLieu:	93
VII.2.2. Dùng Views để hạn chế truy cập trực tiếp:.....	94
VII.2.2.1. View cho Bảng SanPham:.....	94
VII.2.2.2. View cho Bảng NguyenLieu:.....	94
VII.2.2.3. View cho Bảng NhaCungCap:	95

I. Hóa đơn nhóm chọn

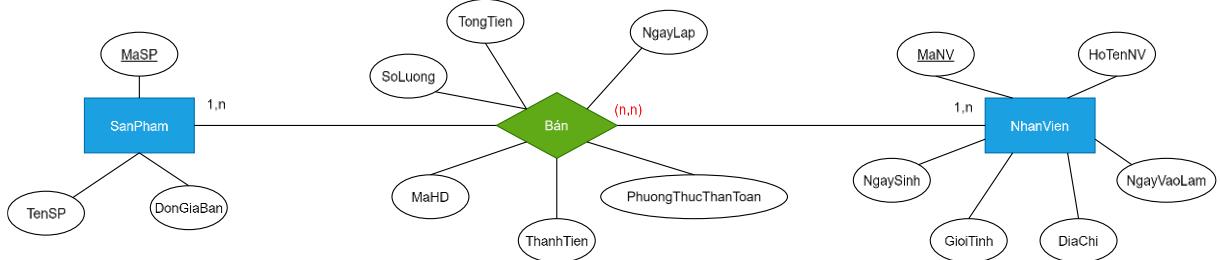


II. Thiết kế cơ sở dữ liệu:

II.1. Thiết kế cơ sở dữ liệu ở mức khái niệm:

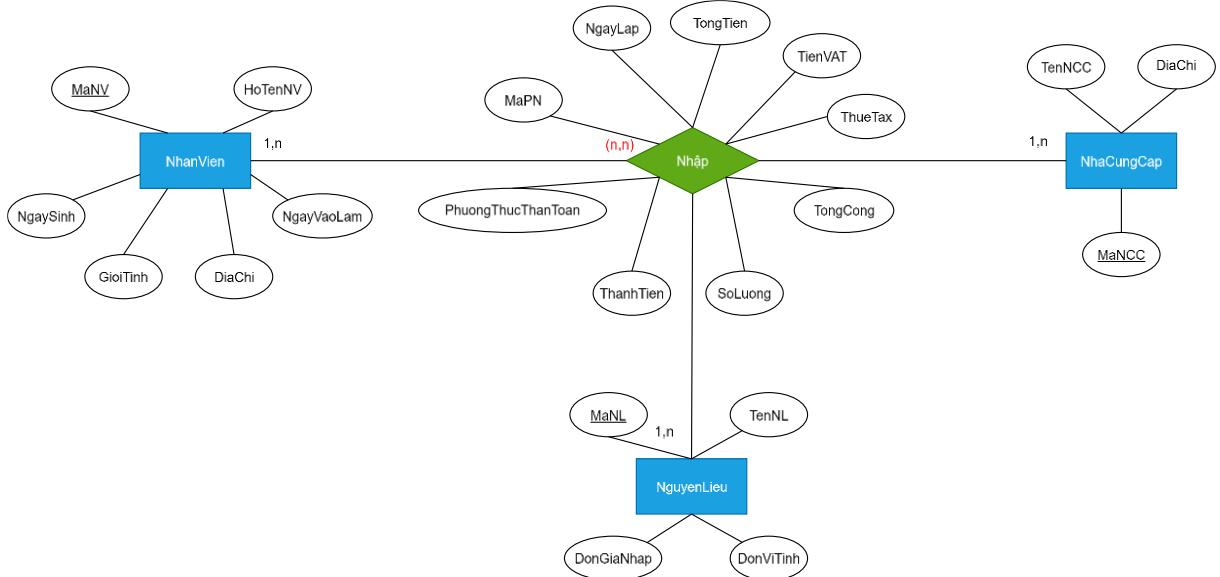
II.1.1. ER-Bán:

- Thực thể NhanVien (MaNV, HoTenNV, NgaySinh, GioiTinh, DiaChi, NgayVaoLam).
- Thực thể SanPham (MaSP, TenSP, DonGiaBan).

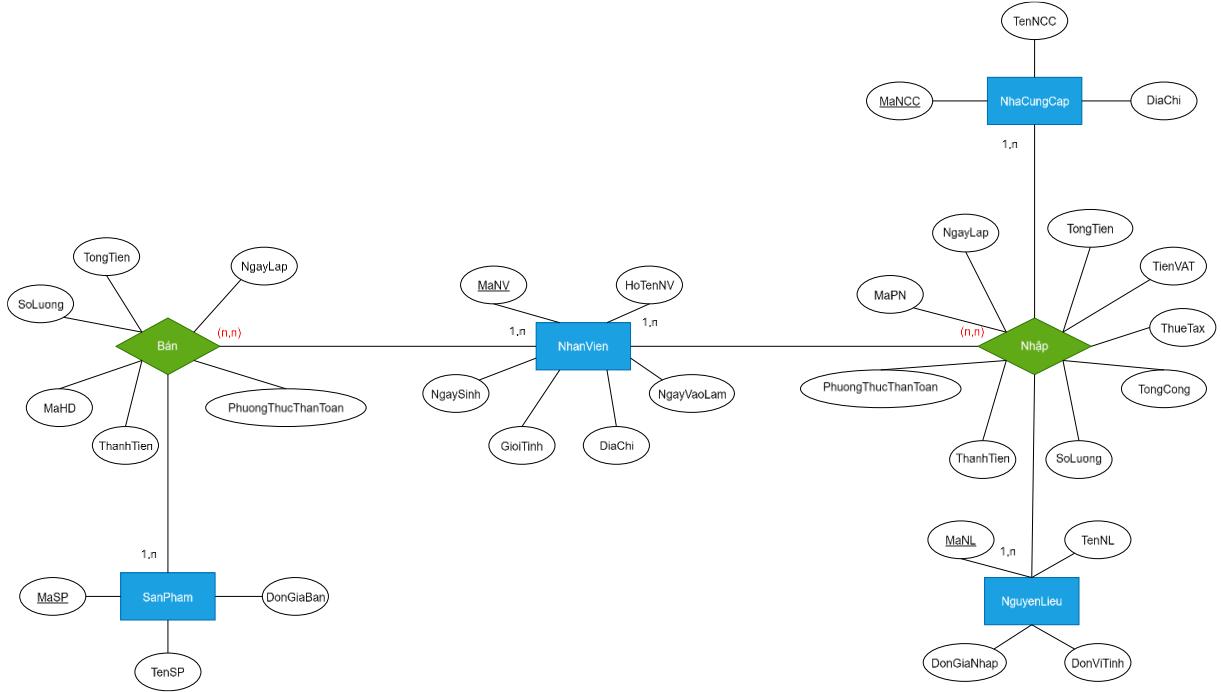


II.1.2. ER-Nhập:

- Thực thể NhanVien (MaNV, HoTenNV, NgaySinh, GioiTinh, DiaChi, NgayVaoLam).
- Thực thể NguyenLieu (MaNL, TenNL, DonGiaNhap, DonViTinh).
- Thực thể NhaCungCap (MaNCC, TenNCC, DiaChi).



II.1.3. Tích hợp các ER:



II.2. Thiết kế cơ sở dữ liệu ở mức logic:

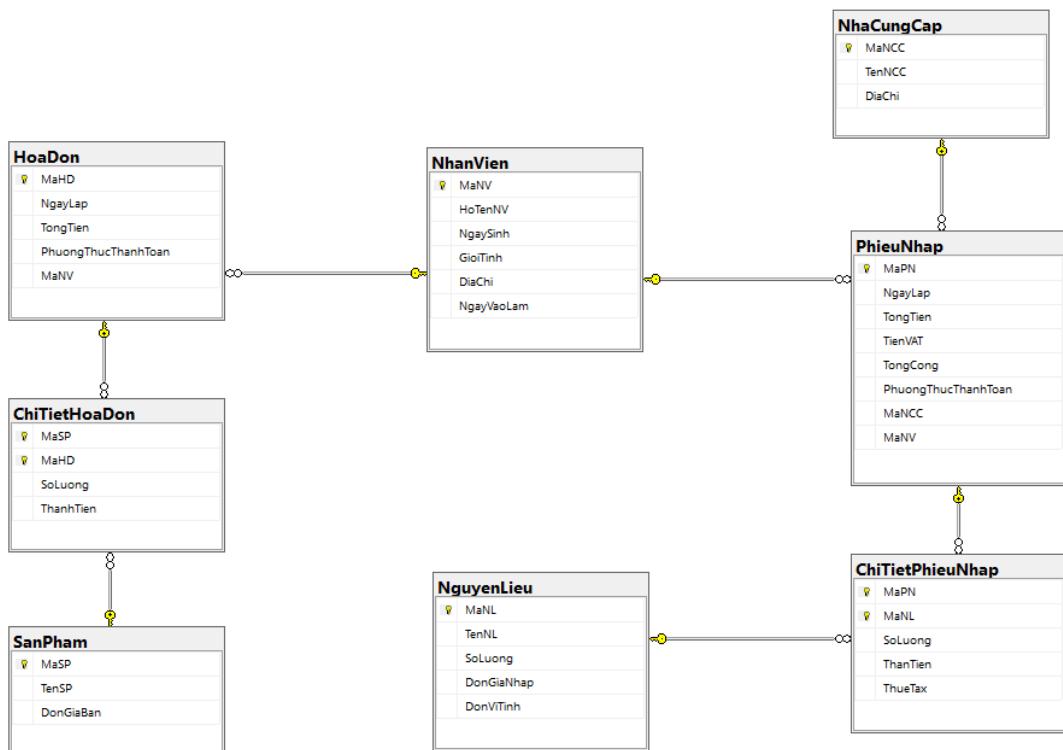
II.2.1. Chuyển thực thể, thu được:

- NhanVien (MaNV, HoTenNV, NgaySinh, GioiTinh, DiaChi, NgayVaoLam).
- SanPham (MaSP, TenSP, DonGiaBan)
- NguyenLieu (MaNL, TenNL, DonGiaNhap, DonViTinh).
- NhaCungCap (MaNCC, TenNCC, DiaChi)

II.2.2. Chuyển quan hệ, thu được:

- Quan hệ Bán:
 - + HoaDon (MaHD, NgayLap, TongTien, PhuongThucThanhToan, MaNV)
 - + ChiTietHoaDon (MaSP, MaHD, SoLuong, ThanhTien)
- Quan hệ Nhập:
 - + PhieuNhap (MaPN, NgayLap, TongTien, TienVAT, TongCong, PhuongThucThanhToan, MaNCC, MaNV)
 - + ChiTietPhieuNhap (MaPN, MaNL, SoLuong, ThanhTien, ThueTax)

II.2.3. Sơ đồ quan hệ



II.3. Thiết kế cơ sở dữ liệu ở mức vật lý:

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaNV	Mã nhân viên	Char(10)	Khoá chính (PK)
HoTenNV	Họ tên nhân viên	Nvarchar(50)	Not null
NgaySinh	Ngày sinh	Datetime	Not null
GioiTinh	Giới tính nhân viên	Bit	Not null
DiaChi	Dịa chỉ nhân viên	Nvarchar(100)	Not null
NgayVaoLam	Ngày vào làm	Datetime	Not null

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaSP	Mã đồ ăn, thức uống	Char(10)	Khoá chính (PK)
TenSP	Tên đồ ăn, thức uống	Nvarchar(50)	Not null
DonGiaBan	Đơn giá	Int	Not null

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaNL	Mã nguyên liệu	Char(10)	Khoá chính (PK)
TenNL	Tên nguyên liệu	Nvarchar(50)	Not null
SoLuong	Số lượng	Int	Not null
DonGiaNhap	Đơn giá	Int	Not null

DonViTinh	Đơn vị tính	Char(10)	Not null
-----------	-------------	----------	----------

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaNCC	Mã nhà cung cấp	Char(10)	Khoá chính (PK)
TenNCC	Tên nhà cung cấp	Nvarchar(100)	Not null
DiaChi	Địa chỉ	Nvarchar(100)	Not null

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaHD	Mã hóa đơn	Char(10)	Khoá chính (PK)
NgayLap	Ngày lập hóa đơn	Datetime	Not null
TongTien	Tổng tiền	Int	Not null
PhuongThucThanhToan	Phương thức thanh toán	Bit	Not null
MaNV	Mã nhân viên	Char(10)	Khoá ngoại từ bảng NhanVien (FK)

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaSP	Mã đồ ăn, thức uống	Char(10)	Khoá chính (PK), Khoá ngoại từ bảng SanPham (FK)
MaHD	Mã hóa đơn	Char(10)	Khoá chính (PK), Khoá ngoại từ bảng HoaDon (FK)
SoLuong	Số lượng	Int	Not null
ThanhTien	Thành tiền	Int	Not null

Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
MaPN	Mã phiếu nhập	Char(10)	Khoá chính (PK)
NgayLap	Ngày lập	Datetime	Not null
TongTien	Cộng tiền hàng	Int	Not null
TienVAT	Tiền thuế GTGT	Int	Not null
TongCong	Tổng cộng tiền thanh toán	Int	Not null

PhuongThucThanhToan	Phương thức thanh toán	Bit	Not null
MaNCC	Mã nhà cung cấp	Char(10)	Khoá ngoại từ bảng NhaCungCap (FK)
MaNV	Mã nhân viên	Char(10)	Khoá ngoại từ bảng NhanVien (FK)

Thuộc tính	Mô tả	Kiểu dữ liệu	Rằng buộc
MaPN	Mã phiếu nhập	Char(10)	Khoá chính (PK), Khoá ngoại từ bảng PhieuNhap (FK)
MaNL	Mã nguyên liệu	Char(10)	Khoá chính (PK), Khoá ngoại từ bảng NguyenLieu (FK)
SoLuong	Số lượng	Int	Not null
ThanTien	Thành tiền	Int	Not null
ThueTax	Thuế suất GTGT	Decimal(4,2)	Not null

III. Xây dựng cơ sở dữ liệu:

III.1. Module tạo dữ liệu dump cho các bảng:

III.1.1. Bảng nhân viên (NhanVien):

- Mã giả:

```
-- Sinh mã tự động
create or alter function fBornNhanVienId ()
returns char(10)
as
begin
    declare @newStaffId varchar(10)

    select
        @newStaffId = MAX(MaNV)
    from NhanVien
    if @newStaffId is null
        set @newStaffId = 'NV00000001'
    else
        begin
            set @newStaffId = 'NV' + RIGHT('00000000'+
CAST((CAST(RIGHT(@newStaffId,8) as int) + 1) as varchar(8)),8)
        end

    return @newStaffId
end

-- Module tạo dữ liệu dump
create or alter proc spInsertValueNhanVien
as
```

```

begin
    declare @i int = 1,
            @RandomDate date,
            @dateJoin date
    while @i <= 1000
    begin
        -- Tạo ngày sinh hợp lý từ 18-60 tuổi
        -- NewId: tạo ra mã id ngẫu nhiên
        -- CheckSum: tính tổng giá trị của chuỗi ngẫu nhiên
        set @RandomDate = DATEADD(day,
                                    ABS(CHECKSUM(NEWID()) % 365),
                                    -- Tính ngày ngẫu nhiên trong năm,
                                    DATEADD(year,
                                            - (18 + ABS(CHECKSUM(NEWID()) % (60-18+1))),
                                            -- Tính năm ngẫu nhiên trong khoảng 18-60 tuổi
                                            GETDATE()))
        )

        -- Sinh ngày làm tự động với ngày vào làm nhỏ hơn ngày hiện tại và lớn
        -- hoặc bằng ngày nhân viên đủ 18 tuổi
        set @dateJoin = DATEADD(day,
                                ABS(CHECKSUM(NEWID()) % 365),
                                DATEADD(year,
                                        - ABS(CHECKSUM(NEWID()) % (YEAR(getdate())-
                                YEAR(@RandomDate)+1)),
                                GETDATE()))
        )

        if YEAR(@dateJoin) - YEAR(@RandomDate) < 18
            set @dateJoin = GETDATE()

        insert into NhanVien (MaNV, HoTenNV, NgaySinh, GioiTinh, DiaChi,
        NgayVaoLam)
        values (
            dbo.fBornNhanVienId(),
            N'Nhân viên ' + CAST(@i as varchar(5)),
            @RandomDate,
            case when @i%2 = 0 then 0 else 1 end,
            N'Địa chỉ minh họa ' + CAST(@i as varchar(5)),
            @dateJoin
        )
        if @@rowcount <= 0
        begin
            print N'insert thất bại'
            return
        end
        set @i = @i + 1
    end
end

exec spInsertValueNhanVien

```

- Kết quả:

	MaNV	HoTenNV	NgaySinh	GioiTinh	DiaChi	NgayVaoLam
1	NV00000001	Nhân viên 1	2000-06-02 00:00:00.000	1	Địa chỉ minh họa 1	2024-10-17 00:00:00.000
2	NV00000002	Nhân viên 2	1991-04-15 00:00:00.000	0	Địa chỉ minh họa 2	2019-06-10 00:00:00.000
3	NV00000003	Nhân viên 3	2004-05-08 00:00:00.000	1	Địa chỉ minh họa 3	2024-10-17 00:00:00.000
4	NV00000004	Nhân viên 4	1976-05-03 00:00:00.000	0	Địa chỉ minh họa 4	1999-11-11 00:00:00.000
5	NV00000005	Nhân viên 5	2004-04-05 00:00:00.000	1	Địa chỉ minh họa 5	2024-10-17 00:00:00.000
6	NV00000006	Nhân viên 6	2005-07-10 00:00:00.000	0	Địa chỉ minh họa 6	2024-10-17 00:00:00.000
7	NV00000007	Nhân viên 7	1971-07-26 00:00:00.000	1	Địa chỉ minh họa 7	2000-11-10 00:00:00.000
8	NV00000008	Nhân viên 8	1990-06-24 00:00:00.000	0	Địa chỉ minh họa 8	2024-10-17 00:00:00.000
9	NV00000009	Nhân viên 9	1992-02-06 00:00:00.000	1	Địa chỉ minh họa 9	2016-06-22 00:00:00.000
10	NV00000010	Nhân viên 10	2002-11-08 00:00:00.000	0	Địa chỉ minh họa 10	2024-10-17 00:00:00.000

III.1.2. Bảng Nhà cung cấp (NhaCungCap):

- Mã giả:

```
-- Sinh mã tự động
create or alter function fBornNhaCungCapId ()
returns char(10)
as
begin
    declare @newSupplierId varchar(10)
    select
        @newSupplierId = MAX(MaNCC)
    from NhaCungCap

    if @newSupplierId is null
        set @newSupplierId = 'NCC0000001'
    else
        set @newSupplierId = 'NCC' + RIGHT('0000000' +
CAST((CAST(RIGHT(@newSupplierId,7) as int) + 1) as varchar(7)),7)

    return @newSupplierId
end

-- Module tạo dữ liệu dump
create or alter proc spInsertValueNhaCungCap
as
begin
    declare @i int = 1

    while @i <= 1000
    begin
        insert into NhaCungCap (MaNCC, TenNCC,DiaChi)
        values (
            dbo.fBornNhaCungCapId(),
            N'Nhà cung cấp ' + CAST(@i as varchar(5)),
            N'Dịa chỉ nhà cung cấp ' + CAST(@i as varchar(5))
        )
        if @@rowcount <= 0
        begin
            print N'insert thất bại'
            return
        end

        set @i = @i + 1
    end
end

exec spInsertValueNhaCungCap
```

- Kết quả:

	MaNCC	TenNCC	DiaChi
1	NCC0000001	Nhà cung cấp 1	Địa chỉ nhà cung cấp 1
2	NCC0000002	Nhà cung cấp 2	Địa chỉ nhà cung cấp 2
3	NCC0000003	Nhà cung cấp 3	Địa chỉ nhà cung cấp 3
4	NCC0000004	Nhà cung cấp 4	Địa chỉ nhà cung cấp 4
5	NCC0000005	Nhà cung cấp 5	Địa chỉ nhà cung cấp 5
6	NCC0000006	Nhà cung cấp 6	Địa chỉ nhà cung cấp 6
7	NCC0000007	Nhà cung cấp 7	Địa chỉ nhà cung cấp 7
8	NCC0000008	Nhà cung cấp 8	Địa chỉ nhà cung cấp 8
9	NCC0000009	Nhà cung cấp 9	Địa chỉ nhà cung cấp 9
10	NCC0000010	Nhà cung cấp 10	Địa chỉ nhà cung cấp 10

III.1.3. Bảng Sản Phẩm (SanPham):

- Mã giả:

```
-- Sinh mã tự động
create or alter function fBornSanPhamId()
returns char(10)
as
begin
    declare @newProductId varchar(10)

    -- MaSP: "SP00100001" (SP[xxx (dòng SP)] [xxxxx (loại SP)] (có 10 kí tự)
    select
        @newProductId = MAX(MaSP)
    from SanPham

    if @newProductId is null
        set @newProductId = 'SP00000001'
    else
        set @newProductId = 'SP' + RIGHT('00000000' +
    CAST((CAST(RIGHT(@newProductId,8) as int) + 1) as varchar(8)),8)

    return @newProductId
end

-- Module tạo dữ liệu dump
create or alter proc spInsertValueSanPham
as
begin
    declare @i int = 1

    while @i <= 1000
    begin
        insert into SanPham (MaSP,TenSP,DonGiaBan)
        values (
            dbo.fBornSanPhamId(),
            N'San pham ' + CAST(@i as varchar(5)),
            CAST((RAND()*100000) as int)
        )

        if @@rowcount <= 0
        begin
            print N'insert thất bại'
            return
        end
    end
end
```

```

        end

        set @i = @i + 1
    end
end

exec spInsertValueSanPham

```

- Kết quả:

	MãSP	TênSP	ĐơnGiáBan
1	SP00000001	San pham 1	27061
2	SP00000002	San pham 2	19818
3	SP00000003	San pham 3	3662
4	SP00000004	San pham 4	63933
5	SP00000005	San pham 5	55927
6	SP00000006	San pham 6	42251
7	SP00000007	San pham 7	61536
8	SP00000008	San pham 8	15392
9	SP00000009	San pham 9	94108
10	SP00000010	San pham 10	11257

III.1.4. Bảng Nguyên Liệu (NguyenLieu):

- Mã giả:

```

-- Sinh mã tự động
create or alter function fBornValueNguyenLieuId()
returns char(10)
as
begin
    declare @newIngredientId varchar(10)

    -- MaNL: "NL00100001" (NL[xxx (dòng NL)] [xxxxx (loại NL)] (có 10 kí tự)
    select
        @newIngredientId = MAX(MaNL)
    from NguyenLieu

    if @newIngredientId is null
        set @newIngredientId = 'NL00000001'
    else
        set @newIngredientId = 'NL' + RIGHT('00000000'+
CAST((CAST(RIGHT(@newIngredientId,8) as int) + 1) as varchar(8)),8)

    return @newIngredientId
end

-- Module tạo dữ liệu dump
create or alter proc spInsertValueNguyenLieu
as
begin
    declare @i int = 1

    while @i <= 1000
    begin
        insert into NguyenLieu (MaNL, TenNL, SoLuong, DonGiaNhap, DonViTinh)
        values (
            dbo.fBornValueNguyenLieuId(),
            N'Nguyen lieu ' + CAST(@i as varchar(5)),

```

```

        @i%10*5,
        CAST((RAND()*10000) as int),
        case
            when @i%3 = 0 then 'Ky'
            when @i%3 = 1 then 'Thung'
            when @i%3 = 2 then 'Hop'
        end
    )

    if @@rowcount <= 0
    begin
        print N'insert thất bại'
        return
    end

    set @i = @i + 1
end
end

exec spInsertValueNguyenLieu

```

- Kết quả:

	MaNL	TenNL	SoLuong	DonGiaNhap	DonViTinh
1	NL00000001	Nguyen lieu 1	5	8719	Thung
2	NL00000002	Nguyen lieu 2	10	9437	Hop
3	NL00000003	Nguyen lieu 3	15	519	Ky
4	NL00000004	Nguyen lieu 4	20	7932	Thung
5	NL00000005	Nguyen lieu 5	25	9653	Hop
6	NL00000006	Nguyen lieu 6	30	2879	Ky
7	NL00000007	Nguyen lieu 7	35	3736	Thung
8	NL00000008	Nguyen lieu 8	40	5898	Hop
9	NL00000009	Nguyen lieu 9	45	3757	Ky
10	NL00000010	Nguyen lieu 10	0	1895	Thung

III.1.5. Bảng Phiếu nhập (PhieuNhap):

- Mã giả:

```

-- Sinh mã tự động
create or alter function fBornPhieuNhapId ()
returns char(10)
as
begin
    declare @newEntryId varchar(10)

    select
        @newEntryId = MAX(MaPN)
    from PhieuNhap

    if @newEntryId is null
        set @newEntryId = 'PN00000001'
    else
        set @newEntryId = 'PN' + RIGHT('00000000' +
CAST((CAST(RIGHT(@newEntryId,8) as int) + 1) as varchar(8)),8)

    return @newEntryId
end

```

```

-- Module tạo dữ liệu dump
create or alter proc spInsertValuePhieuNhap
as
begin
    declare      @i int = 1

    while @i <= 1000
    begin
        declare @dateCreate date,
                @total int
        -- Sinh ngày lập tự động trong vòng 10 năm đổ lại
        -- Ngày lập <= ngày hiện tại
        set @dateCreate = DATEADD(day,
            - ABS(CHECKSUM(NEWID()) % 365),
            DATEADD(year,
            - ABS(CHECKSUM(NEWID()) % 15),
            --giới hạn 10 năm tính từ năm hiện tại
            GETDATE()))
    )

    set @total = CAST((RAND()*100000) as int)

    -- chèn dữ liệu vào bảng phieunhap với mã phiếu nhập tự
    -- Tổng cộng = Tổng tiền + Tien VAT
    insert into PhieuNhap(MaPN, NgayLap, TongTien, TienVAT, TongCong,
    PhuongThucThanhToan, MaNCC, MaNV)
    values (
        dbo.fBornPhieuNhapId(),
        @dateCreate,
        @total,
        @total*0.08,
        @total*1.08,
        @i%2,
        'NCC'+ RIGHT('0000000'+CAST( (FLOOR(1 + (RAND() * 1000)))
    as varchar(7)),7),
        'NV'+ RIGHT('00000000'+CAST( (FLOOR(1 + (RAND() * 1000)))
    as varchar(8)),8)
    )
    if @@rowcount <= 0
    begin
        print N'insert thất bại'
        return
    end

    set @i = @i + 1
end
end
exec spInsertValuePhieuNhap

```

- Kết quả:

	MaPN	NgayLap	TongTien	TienVAT	TongCong	PhuongThucThanhToan	MaNCC	MaNV
1	PN00000001	2019-07-08 00:00:00.000	6015	481	6496	1	NCC0000351	NV00000565
2	PN00000002	2010-12-10 00:00:00.000	15981	1278	17259	0	NCC0000974	NV00000995
3	PN00000003	2022-02-03 00:00:00.000	34687	2774	37461	1	NCC0000590	NV00000535
4	PN00000004	2014-06-17 00:00:00.000	90476	7238	97714	0	NCC0000943	NV00000101
5	PN00000005	2010-12-26 00:00:00.000	18084	1446	19530	1	NCC0000155	NV00000630
6	PN00000006	2017-06-10 00:00:00.000	4768	381	5149	0	NCC0000509	NV00000367
7	PN00000007	2013-10-29 00:00:00.000	35864	2869	38733	1	NCC0000019	NV0000022
8	PN00000008	2023-12-11 00:00:00.000	9748	779	10527	0	NCC0000547	NV00000114
9	PN00000009	2013-03-26 00:00:00.000	29050	2324	31374	1	NCC0000365	NV00000393
10	PN00000010	2023-06-25 00:00:00.000	11728	938	12666	0	NCC0000921	NV00000946

III.1.6. Bảng Chi tiết phiếu nhập (ChiTietPhieuNhap):

- Mã giả:

```

create or alter proc spInsertValueChiTietPhieuNhap
as
begin
    declare @i int = 1

    while @i <= 1000
    begin
        declare @MaNL varchar(10),
                @DonGiaNL int,
                @SoLuong int,
                @ThanhTien numeric(12,0)

        set @MaNL = 'NL'+ RIGHT('00000000'+CAST( (FLOOR(1 + (RAND() * 1000))) as
varchar(8)),8)
        set @SoLuong = FLOOR(1 + (RAND() * 10))

        select @DonGiaNL = DonGiaNhap
        from NguyenLieu

        insert into ChiTietPhieuNhap (MaPN, MaNL, SoLuong, ThanTien, ThueTax)
        values (
            'PN'+ RIGHT('00000000'+CAST( (FLOOR(1 + (RAND() * 1000))) as
varchar(8)),8),
            @MaNL,
            @SoLuong,
            @SoLuong*@DonGiaNL,
            0.08
        )

        set @i = @i + 1
    end
end

exec spInsertValueChiTietPhieuNhap

```

- Kết quả:

	MaPN	MaNL	SoLuong	ThanTien	ThueTax
1	PN00000002	NL00000136	7	38605	0.08
2	PN00000002	NL00000322	10	55150	0.08
3	PN00000003	NL00000434	1	5515	0.08
4	PN00000004	NL00000498	1	5515	0.08
5	PN00000004	NL00000646	2	11030	0.08
6	PN00000006	NL00000338	4	22060	0.08
7	PN00000008	NL00000054	10	55150	0.08
8	PN00000010	NL00000415	8	44120	0.08
9	PN00000010	NL00000911	5	27575	0.08
10	PN00000012	NL00000216	3	16545	0.08

III.1.7. Bảng Hóa đơn (HoaDon):

- Mã giả:

```
-- Sinh mã tự động
create or alter function fBornHoaDonId ()
returns char(10)
as
begin
    declare @newBillId varchar(10)

    select
        @newBillId = MAX(MaHD)
    from HoaDon

    if @newBillId is null
        set @newBillId = 'HD00000001'
    else
        set @newBillId = 'HD' + RIGHT('00000000' + CAST((CAST(RIGHT(@newBillId,8)
as int) + 1) as varchar(8)),8)

    return @newBillId
end

-- Module tạo dữ liệu dump
create or alter proc spInsertValueHoaDon
as
begin
    declare @i int = 1

    while @i <= 1000
    begin
        declare @dateCreate date,
                @total int
        -- Sinh ngày lập tự động trong vòng 10 năm đổ lại
        -- Ngày lập <= ngày hiện tại
        set @dateCreate = DATEADD(day,
            - ABS(CHECKSUM(NEWID()) % 365),
            DATEADD(year,
            - ABS(CHECKSUM(NEWID()) % 15),
            --giới hạn 10 năm tính từ năm hiện tại
            GETDATE()))
    )
    set @total = CAST((RAND()*100000) as int)
```

```

-- chèn dữ liệu vào bảng hoaDon với mã phiếu nhập tự động
insert into HoaDon (MaHD, NgayLap, TongTien, PhuongThucThanhToan, MaNV)
values (
            dbo.fBornHoaDonId(),
            @dateCreate,
            @total,
            @i%2,
            'NV'+ RIGHT('00000000'+CAST((FLOOR(1 + (RAND() * 1000))) as
varchar(8)),8)
        )
if @@rowcount <= 0
begin
    print N'insert thất bại'
    return
end

set @i = @i + 1
end
end

exec spInsertValueHoaDon

```

- Kết quả:

	MaHD	NgayLap	TongTien	PhuongThucThanhToan	MaNV
1	HD00000001	2014-01-22 00:00:00.000	81981	1	NV00000403
2	HD00000002	2010-07-23 00:00:00.000	89498	0	NV00000766
3	HD00000003	2023-12-10 00:00:00.000	6241	1	NV00000989
4	HD00000004	2014-03-30 00:00:00.000	70695	0	NV00000989
5	HD00000005	2023-11-27 00:00:00.000	34497	1	NV00000421
6	HD00000006	2023-06-16 00:00:00.000	68729	0	NV00000252
7	HD00000007	2023-03-19 00:00:00.000	74238	1	NV00000833
8	HD00000008	2010-06-15 00:00:00.000	96769	0	NV00000620
9	HD00000009	2012-03-06 00:00:00.000	77166	1	NV00000034
10	HD00000010	2016-08-16 00:00:00.000	74149	0	NV00000176

III.1.8. Bảng Chi tiết hóa đơn (ChiTietHoaDon):

- Mã giả:

```

create or alter proc spInsertValueChiTietHoaDon
as
begin
    declare @i int = 1

    while @i <= 1000
    begin
        declare @MaHD varchar(10),
                @DonGiaBan int,
                @SoLuong int,
                @ThanhTien numeric(12,0)

        set @MaHD = 'HD'+ RIGHT('00000000'+CAST( (FLOOR(1 + (RAND() * 1000))) as
varchar(8)),8)
        set @SoLuong = FLOOR(1 + (RAND() * 10))

        select @DonGiaBan = DonGiaBan
        from SanPham

```

```

        insert into ChiTietHoaDon (MaHD, MaSP, SoLuong, ThanhTien)
        values (
            @MaHD,
            'SP'+ RIGHT('00000000' +CAST( (FLOOR(1 + (RAND() * 1000)))
as varchar(8)),8),
            @SoLuong,
            @SoLuong*@DonGiaBan
        )

        set @i = @i + 1
    end
end

exec spInsertValueChiTietHoaDon

```

- Kết quả:

	MaSP	MaHD	SoLuong	ThanhTien
1	SP00000001	HD00000241	9	693387
2	SP00000001	HD00000388	7	539301
3	SP00000001	HD00000984	7	539301
4	SP00000002	HD00000020	8	616344
5	SP00000002	HD00000273	8	616344
6	SP00000002	HD00000579	9	693387
7	SP00000004	HD00000874	8	616344
8	SP00000005	HD00000211	3	231129
9	SP00000006	HD00000532	8	616344
10	SP00000008	HD00000872	8	616344

III.2. Module phục vụ thao tác xử lý dữ liệu

III.2.1. Module 1 - Kiểm tra sự tồn tại của nguyên liệu:

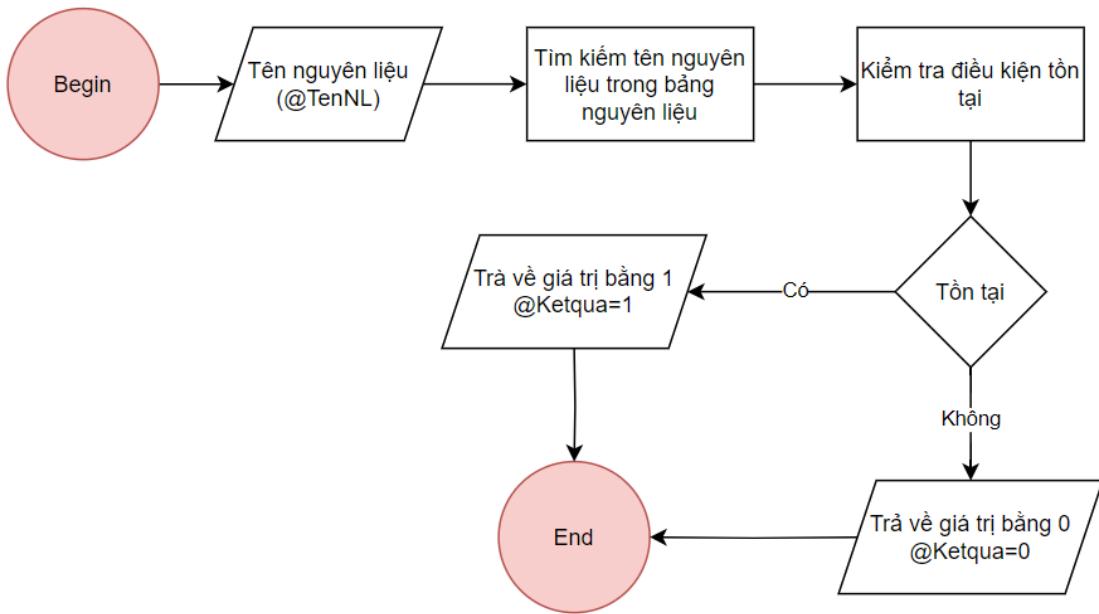
- Mục đích: Các nguyên liệu trong quán thường được đặt tên khác nhau để dễ dàng cho việc phân loại, lựa chọn và tránh nhầm lẫn khi pha chế.
- Đặc tả Module:

Input: Tên nguyên liệu

Output:

- + 1: Tên nguyên liệu đã tồn tại trong hệ thống
- + 0: Tên nguyên liệu chưa tồn tại trong hệ thống

Process:



III.2.2. Module 2 - Kiểm tra mức tồn kho và trả về cảnh báo:

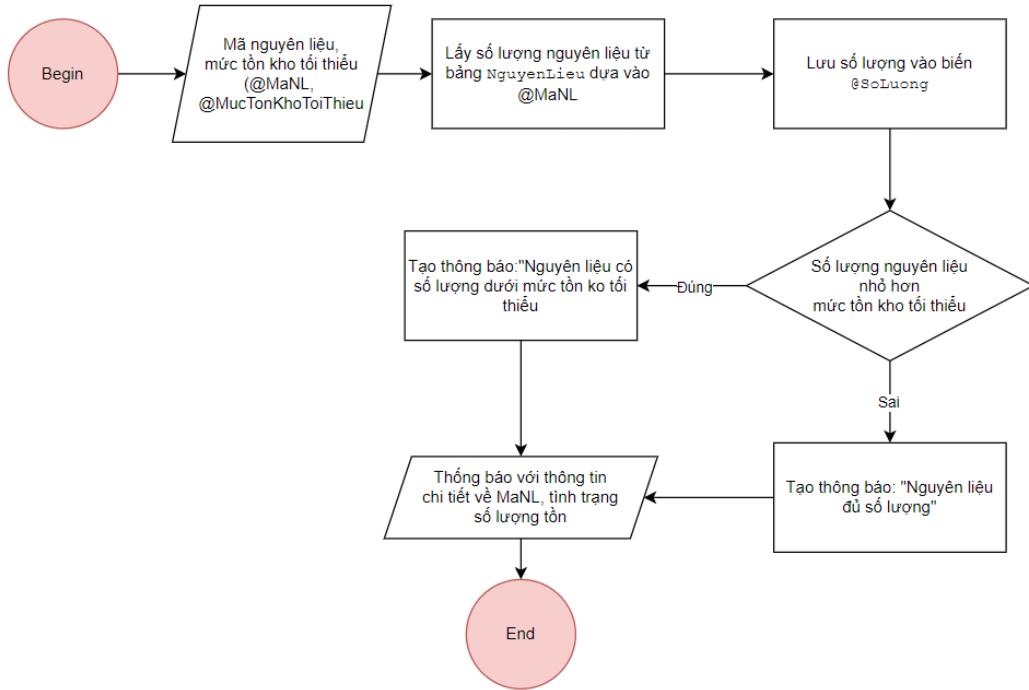
- Mục đích: Số lượng nguyên liệu của quán phải luôn trong tình trạng sẵn sàng đáp ứng các buổi bán hàng nên cần kiểm soát số lượng nguyên liệu tồn kho.
- Đặc tả Module:

Input: Mã nguyên liệu, mức tồn kho tối thiểu

Output: Thông báo

- + "Cảnh báo: Nguyên liệu <mã nguyên liệu> có số lượng dưới mức tồn kho tối thiểu!" hoặc "Nguyên liệu < mã nguyên liệu > có số lượng đủ."

Process:



III.2.3. Module 3 - Cập nhật giá nhập mới của nguyên liệu:

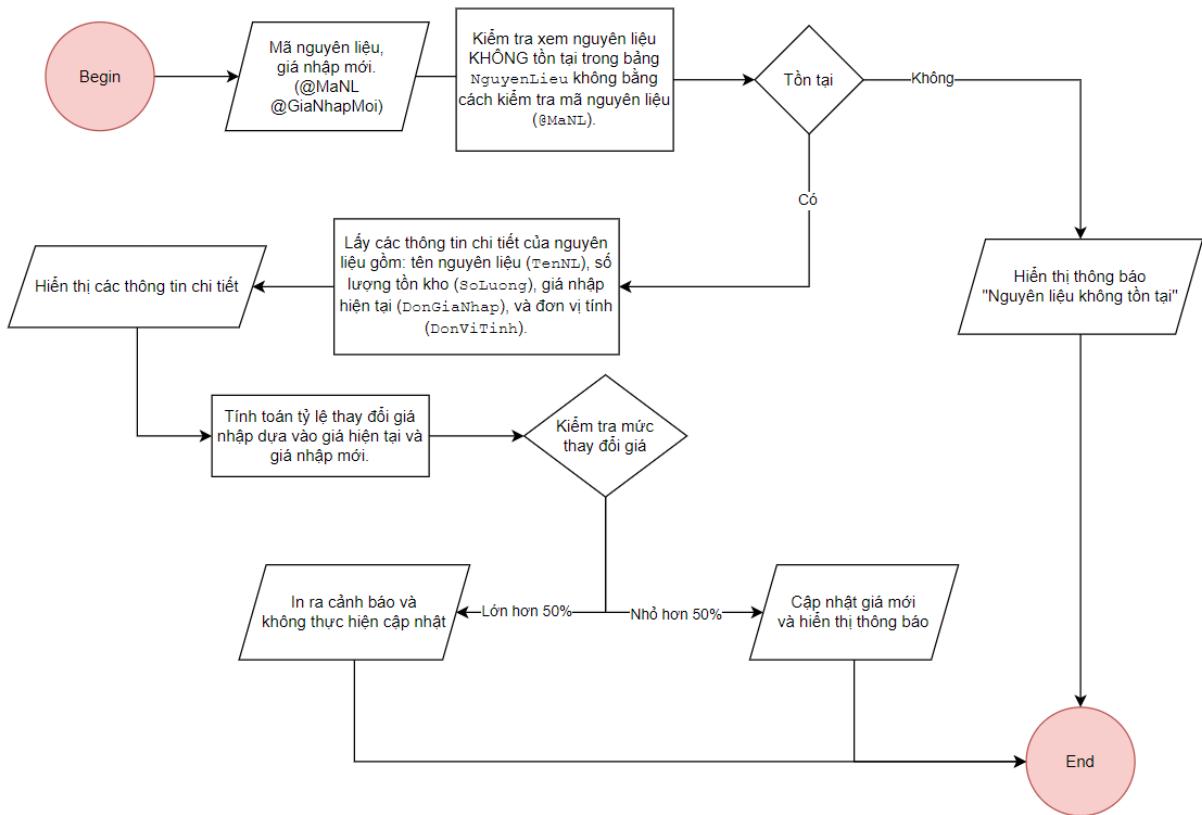
- Mục đích: Khi thực hiện việc nhập hàng sẽ có được những nhà cung cấp mới bán nguyên liệu với giá cả rẻ hơn giá nguyên liệu trong kho. Việc thực hiện cập nhật này nhằm đưa ra mức giá để tìm nhà cung cấp hợp lý tại thời điểm đó. Nhưng nếu sự thay đổi lớn hơn 50% thì thuộc trường hợp chỉ là giá thuộc sự kiện, thời điểm tức thời nào đó mà nhà cung cấp giảm hoặc trường hợp giá cao hơn nhiều so với giá hiện tại.
- Đặc tả module:

Input: Mã nguyên liệu, giá nhập mới

Output:

- + Thông tin chi tiết về nguyên liệu, bao gồm:
 - Mã nguyên liệu
 - Tên nguyên liệu
 - Số lượng tồn kho
 - Đơn giá hiện tại
 - Đơn vị tính
- + Thông báo “Cập nhật giá nhập thành công”, giá nhập mới hoặc cảnh báo “Giá nhập vượt quá 50%. Cập nhật không thành công”

Process:



III.2.4. Module 4 - Tự động cập nhật các tổng tiền phiếu nhập khi thêm mới thông tin chi tiết phiếu nhập:

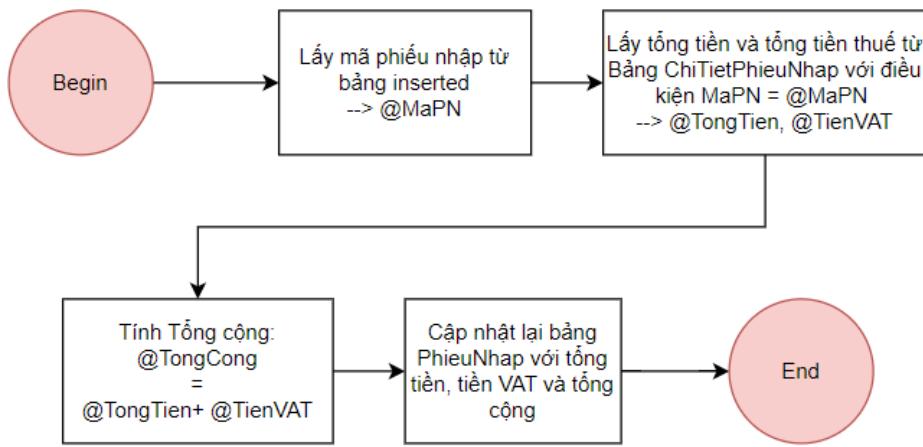
- Mục đích: việc cập nhật các thuộc tính như tổng tiền hàng, tiền thuế, và tổng cộng tiền thanh toán trong bảng “Phiếu nhập” giúp tiết kiệm thời gian cũng như hạn chế sai sót khi chúng ta thêm mới thông tin trong bảng “Chi tiết phiếu nhập”.
- Đặc tả module:

Bảng: ChiTietPhieu Nhap

Loại: After

Sự kiện: Insert

Process:



III.2.5. Module 5 – Tự động cập nhật số lượng nguyên liệu sau khi thêm chi tiết phiếu nhập:

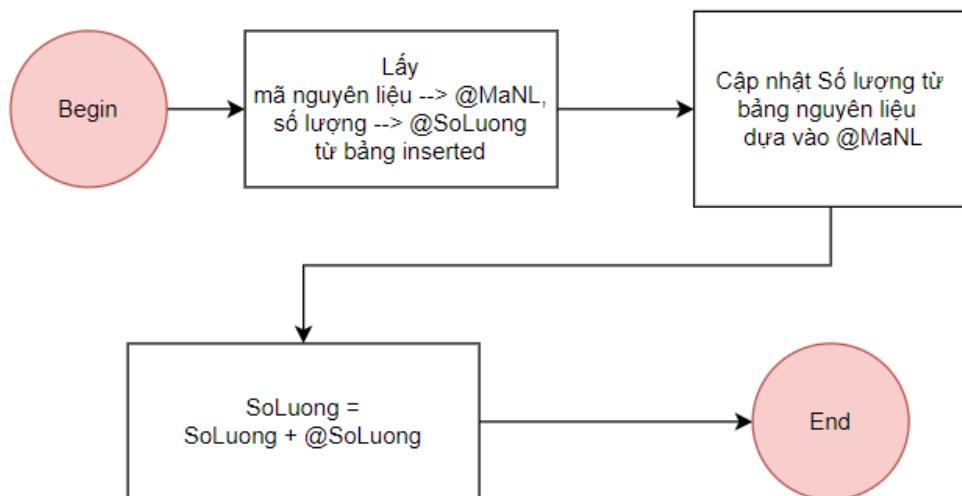
- Mục đích: Sau khi thêm mới thông tin “Chi tiết phiếu nhập” việc tự động cập nhật số lượng nguyên liệu tồn kho rất quan trọng tránh bị sai sót về mặt logic dữ liệu.
- Đặc tả module:

Bảng: ChiTietPhieuNhap

Loại: After

Sự kiện: Insert

Process:



III.2.6. Module 6 - Tự động cập nhật tổng thành tiền cho hóa đơn khi thêm mới thông tin chi tiết hóa đơn:

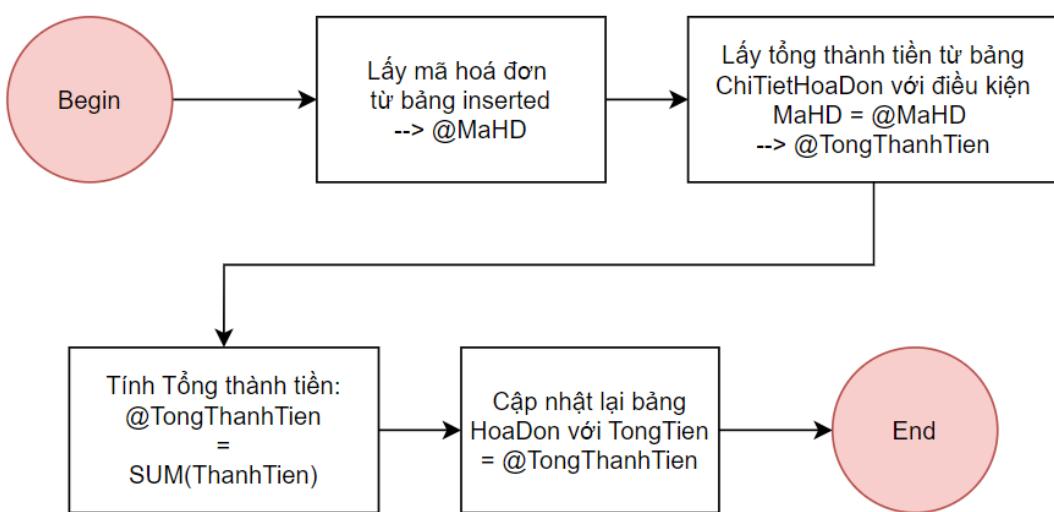
- Mục đích: việc cập nhật thuộc tính tổng thành tiền khách hàng cần thanh toán trong bảng “Hóa đơn” giúp tiết kiệm thời gian cũng như hạn chế sai sót khi chúng ta thêm mới thông tin trong bảng “Chi tiết hóa đơn”.
- Đặc tả module:

Bảng: ChiTietHoaDon

Loại: After

Sự kiện: Insert

Process:



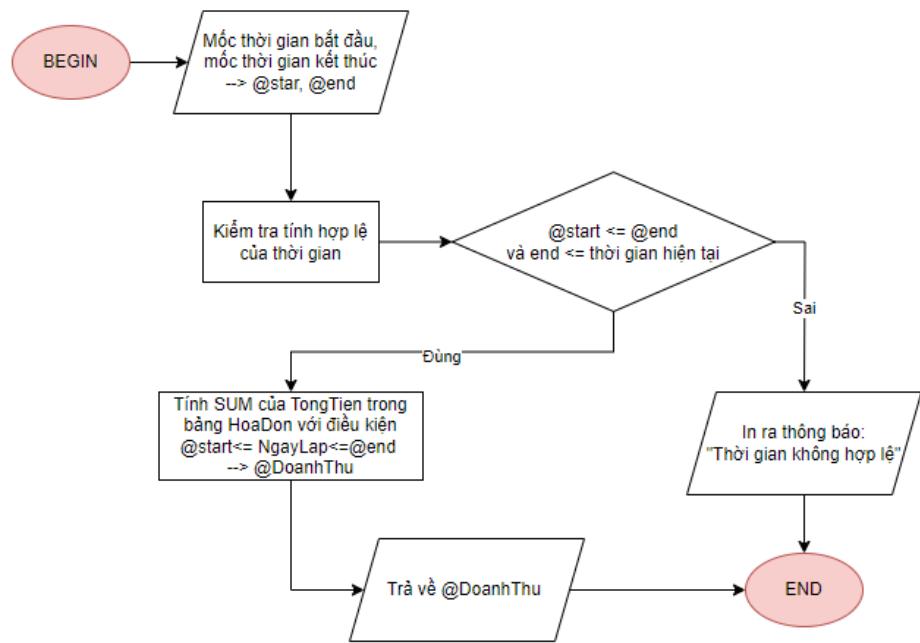
III.2.7. Module 7 - Tính doanh thu dựa vào bảng hóa đơn trong mốc thời gian tuỳ chọn:

- Mục đích: Giúp các nhà quản lý kinh doanh quan xem được doanh thu theo mốc thời gian mà họ mong muốn. (vd: 01/10/2020 đến 31/12/2020)
- Đặc tả module:

Input: Mốc thời gian bắt đầu, mốc thời gian kết thúc

Output: Doanh thu hoặc thông báo “Thời gian không hợp lệ”

Process:



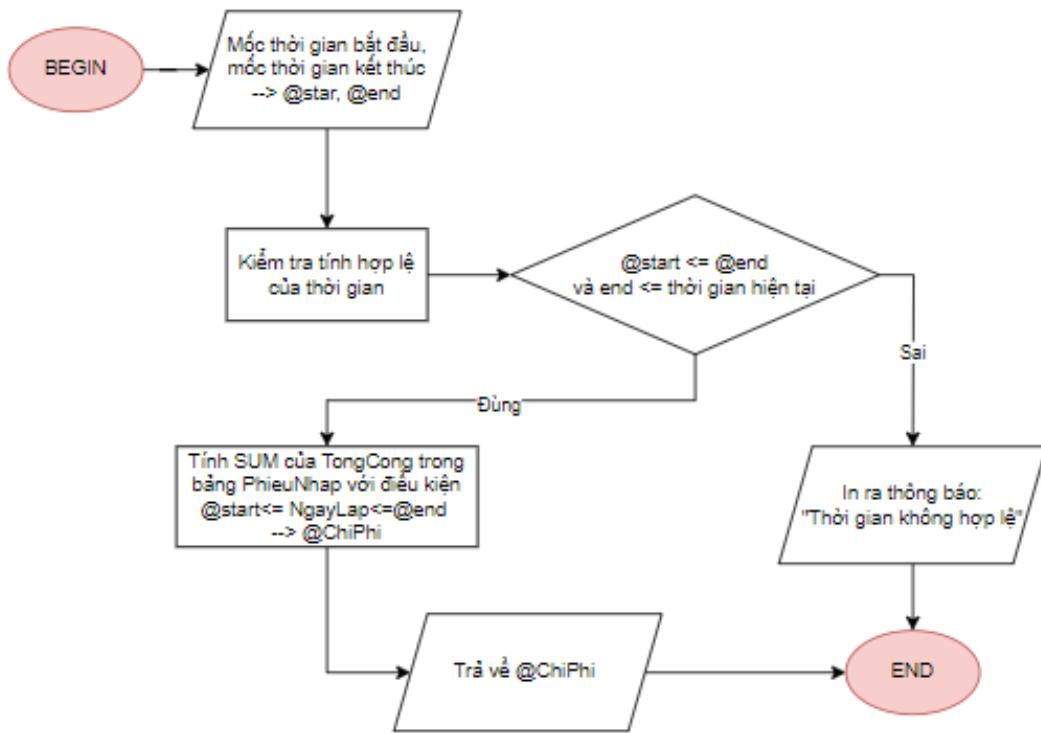
III.2.8. Module 8 - Tính chi phí nhập hàng dựa vào bảng phiếu nhập trong mốc thời gian tự chọn:

- Mục đích: Giúp các nhà quản lý kinh doanh quan xem được chi phí nhập hàng theo mốc thời gian mà họ mong muốn. (vd: 01/10/2020 đến 31/12/2020)
- Đặc tả module:

Input: Mốc thời gian bắt đầu, mốc thời gian kết thúc

Output: Chi phí hoặc thông báo “Thời gian không hợp lệ”

Process:



III.2.9. Module 9 - Tự động lưu vết Hoá đơn lỗi

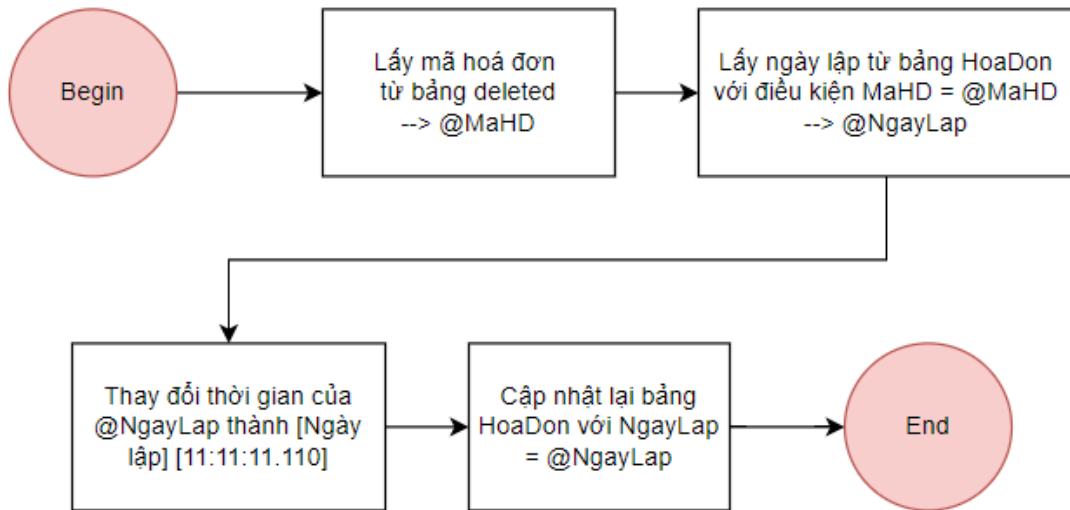
- Mục đích: Trong việc lập hoá đơn đôi khi dẫn đến một số trường hợp gian lận trong việc đổi món thanh toán hay làm sai quy định của quán. Module này giúp nhà quản lý có thể truy xuất được những nhân viên lập hoá đơn lỗi hay xoá thông tin hoá đơn. (Những hoá đơn bị xoá sẽ cập nhật lại thời gian [Ngày lập] [11:11:11.110])
- Đặc tả module:

Bảng: HoaDon

Loại: Instead of

Sự kiện: Delete

Process:



III.2.10. Module 10 - Tự động cập nhật giảm số lượng nguyên liệu tồn kho khi tạo mới chi tiết hóa đơn:

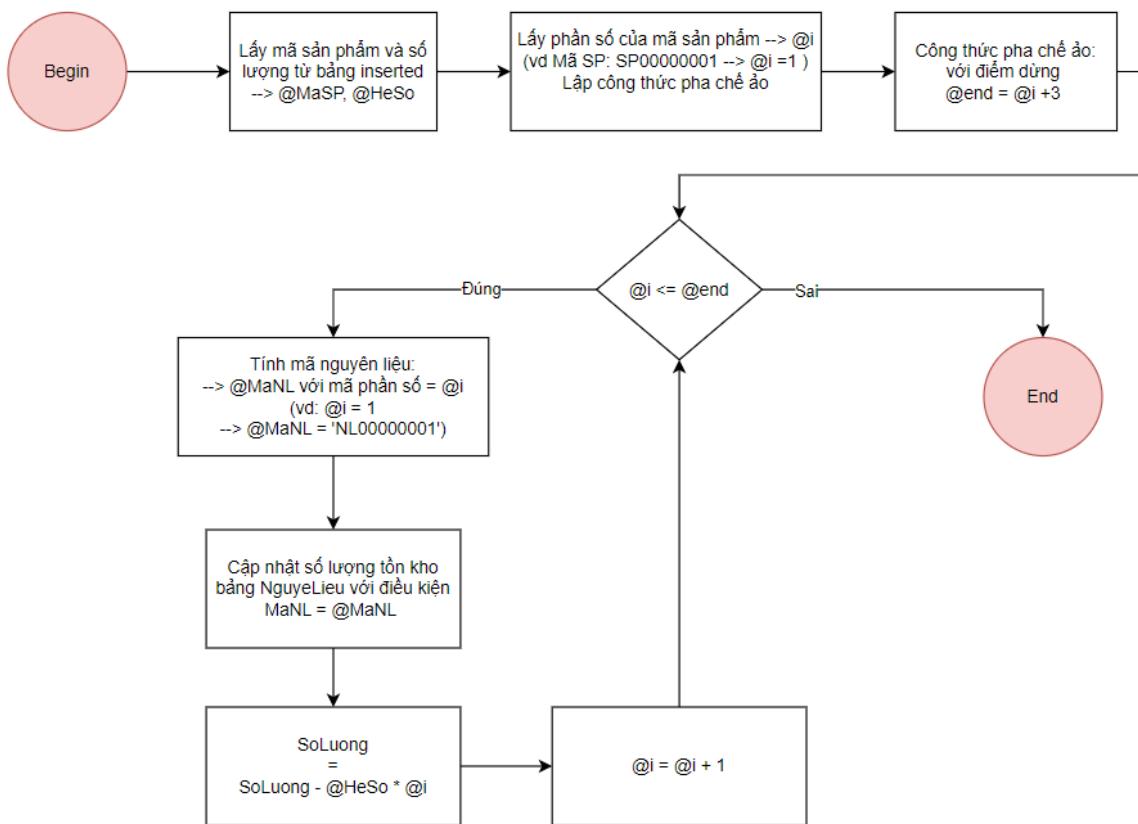
- Mục đích: Mỗi quán sẽ có công thức để tính toán nguyên liệu làm ra một sản phẩm đồ ăn thức uống. Khi thanh toán đồng nghĩa với việc sản phẩm đó đã được pha chế và nguyên liệu cũng giảm đi. Từ đó ta cập nhật lại số lượng tồn kho của nguyên liệu tránh thiếu hụt nguyên liệu pha chế trong bán hàng.
- Đặc tả module:

Bảng: ChiTietHoaDon

Loại: After

Sự kiện: Insert

Process:



IV.Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu:

IV.1. Phân quyền cho các nhóm đối tượng:

Nhân viên (tại chi nhánh):

- Thêm, sửa thông tin HoaDon
- Xem và truy xuất kiểm tra các hoá đơn trước khi chỉnh sửa hoặc thêm mới
- Xem thông tin của NhanVien

Quản lý (tại máy chủ):

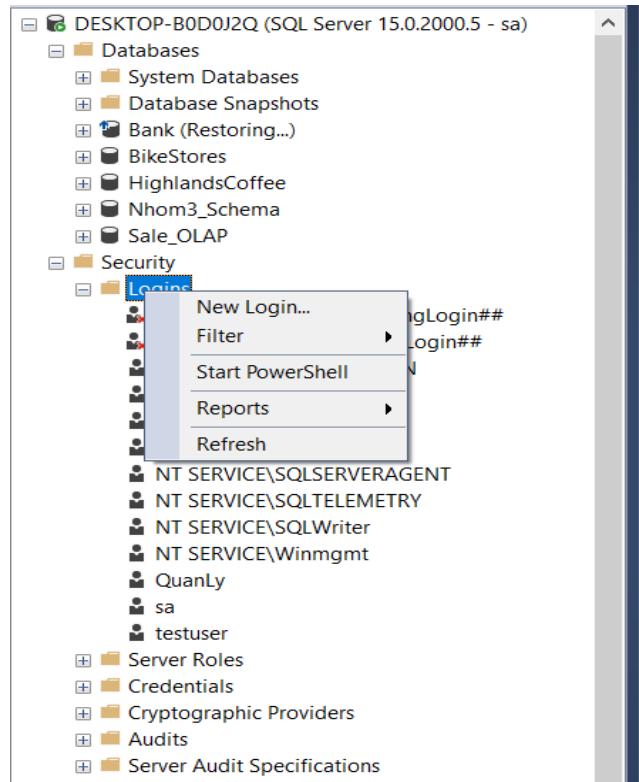
- Thêm, sửa, xóa thông tin của tất cả các bảng.
- Xem báo cáo, thống kê chung của tất cả các chi nhánh.
- Truy cập vào dữ liệu có trong hệ thống, cũng như có tất cả các quyền của các đối tượng trên

IV.2. Phân quyền cho quản lý:

Bước 1: Tạo Login cho người dùng (Quản Lý)

1. Mở SQL Server Management Studio (SSMS) và kết nối với server.

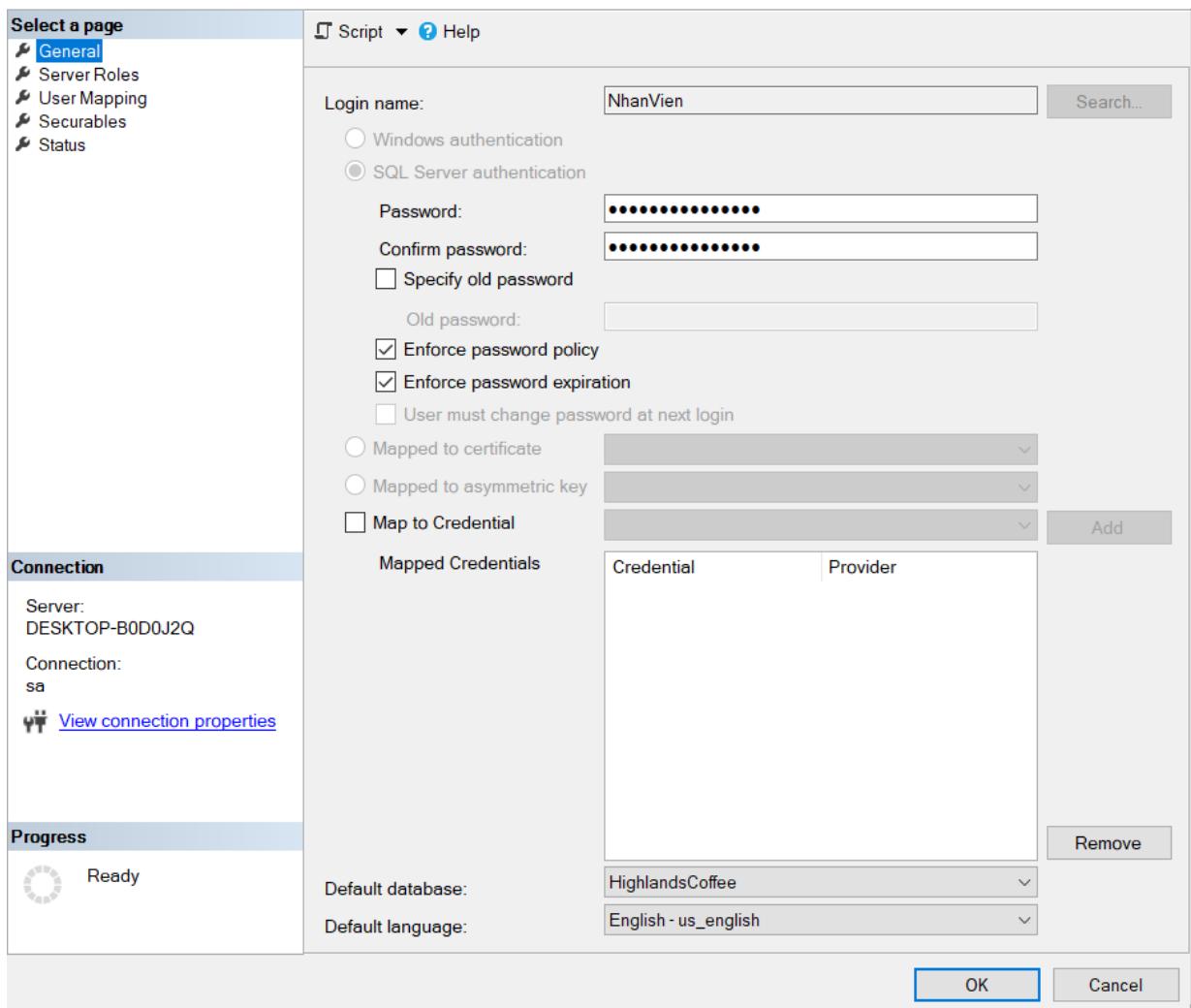
2. Trong khung bên trái, mở rộng mục **Security** và click chuột phải vào **Logins**, sau đó chọn **New Login** để tạo một tài khoản Quản Lý.



3. Ở cửa sổ **Login - New**, nhập tên đăng nhập của người dùng trong ô **Login name**. Chọn **SQL Server Authentication** (xác thực bằng tài khoản SQL)

Login name: QuanLy

Mk: **highland123!**

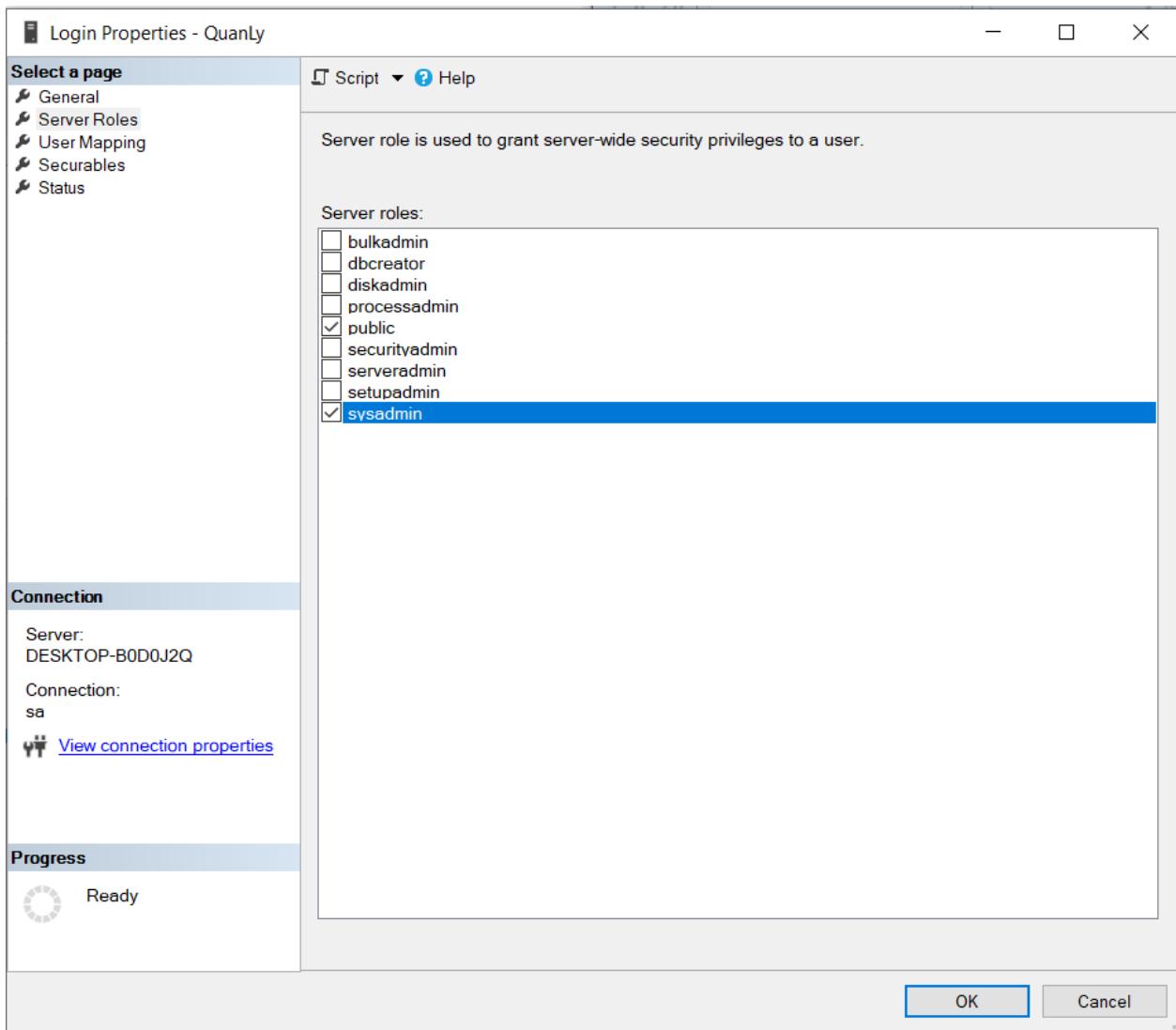


Bước 2: Gán Server Roles (Vai trò trên máy chủ) cho quản lý

- Trong cửa sổ **Login - New** (hoặc chọn login đã tạo và click chuột phải chọn **Properties** để mở lại), vào tab **Server Roles**.
- Đánh dấu các vai trò cần thiết cho người quản lý. Cụ thể:

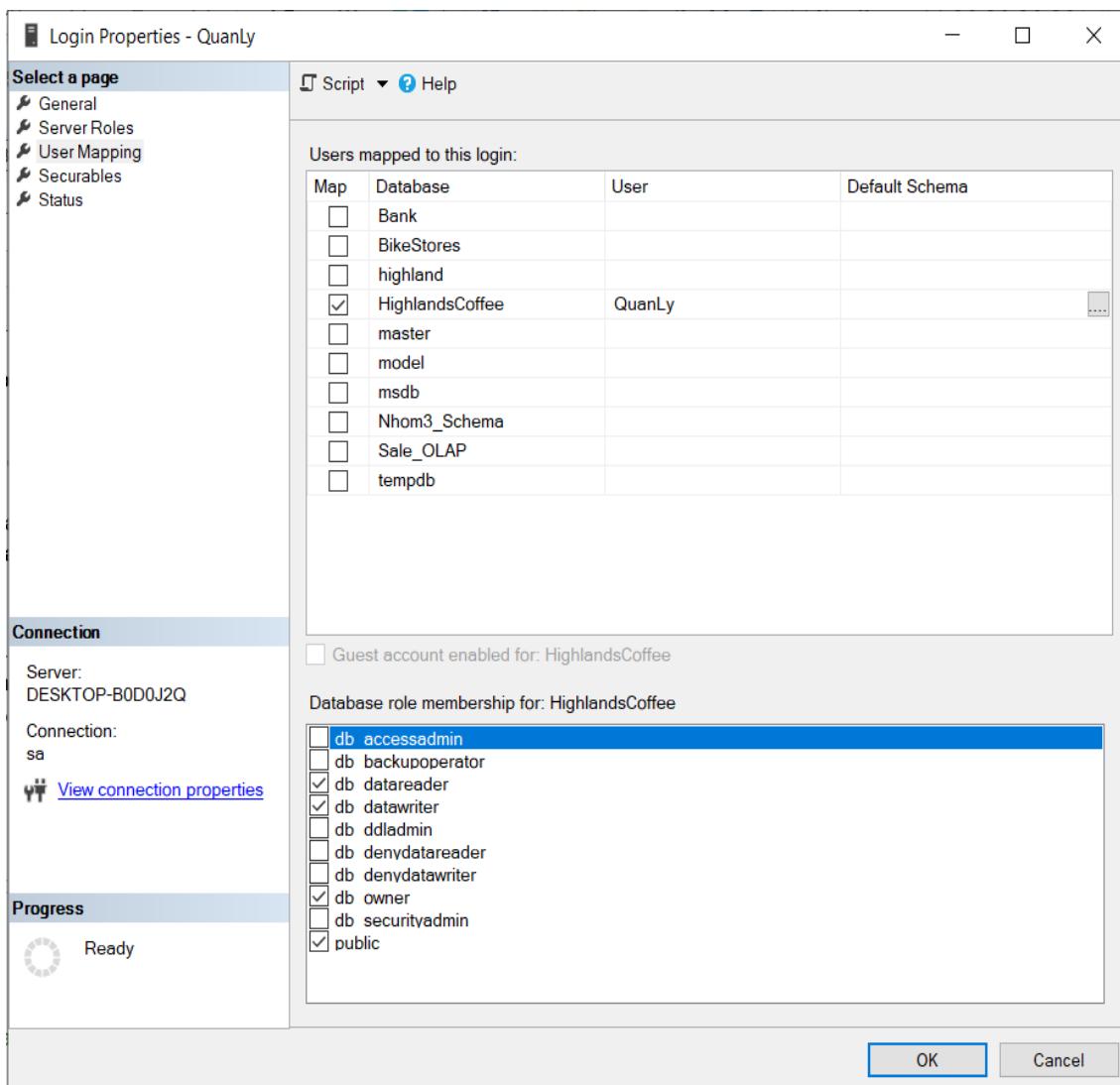
Quản lý: gán các quyền sysadmin

Sysadmin: Vai trò này là cao nhất trên SQL Server và cấp toàn quyền trên tất cả các cơ sở dữ liệu và server. Tài khoản có vai trò này có quyền truy cập vào mọi dữ liệu, có thể thêm, sửa, xóa dữ liệu, và quản lý cấu hình của server.

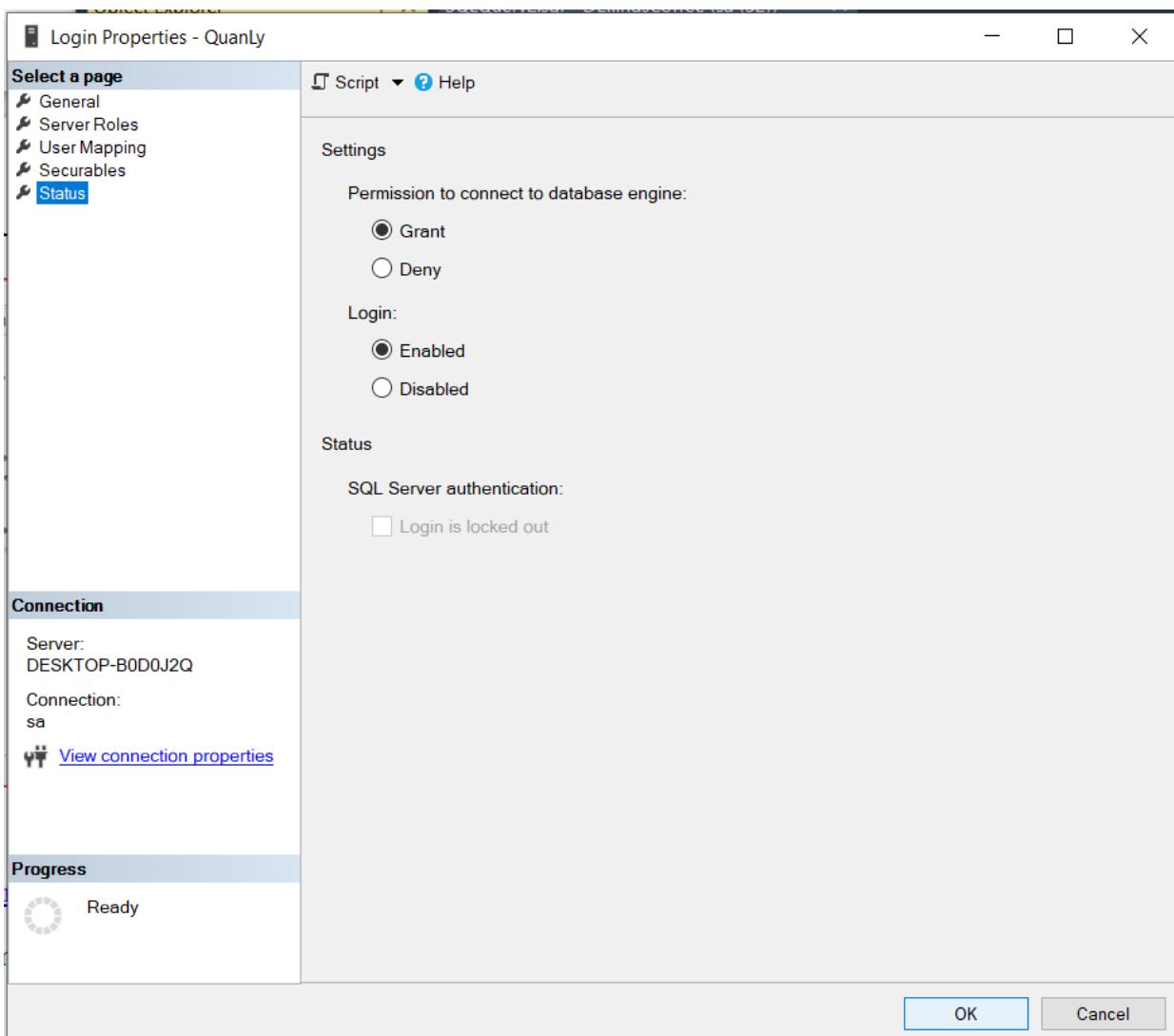


Bước 3: Gán User Mapping (Bản đồ vai trò quản lý trong cơ sở dữ liệu)

1. Vẫn trong cửa sổ **Login - New** hoặc **Properties** của quản lý, vào tab **User Mapping**.
2. Tại đây, chọn cơ sở dữ liệu mà quản lý sẽ được truy cập. Khi chọn một database (cơ sở dữ liệu), phần **Database role membership for:** sẽ hiển thị các vai trò trong cơ sở dữ liệu đó.
3. Gán vai trò cho quản lý trong database:
 - o **Quản lý:** Đánh dấu các vai trò như `db_owner`, `db_datareader`, `db_datawriter`, để Quản lý có quyền thêm, sửa, xóa và truy vấn dữ liệu trên tất cả các bảng trong database.



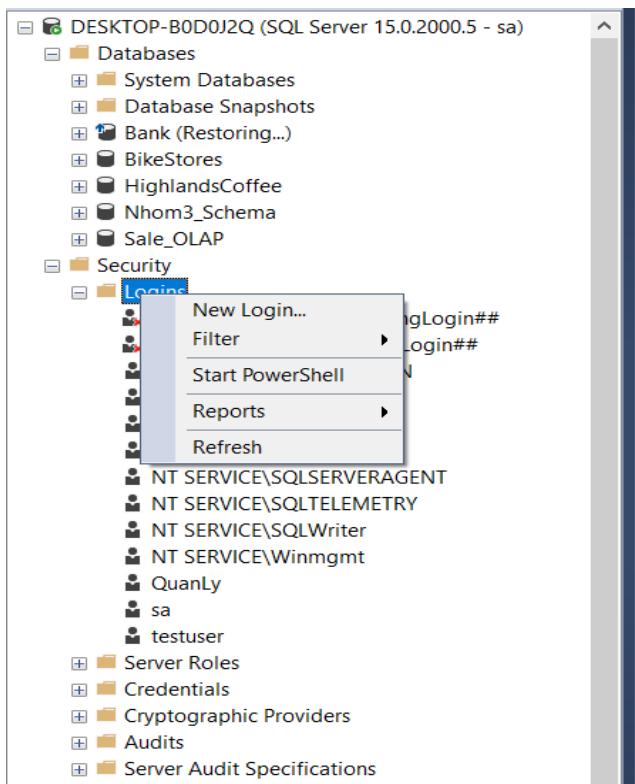
Bước 4: Thiết lập trạng thái



IV.3. Phân quyền cho nhân viên:

Bước 1: Tạo Login cho Nhân Viên

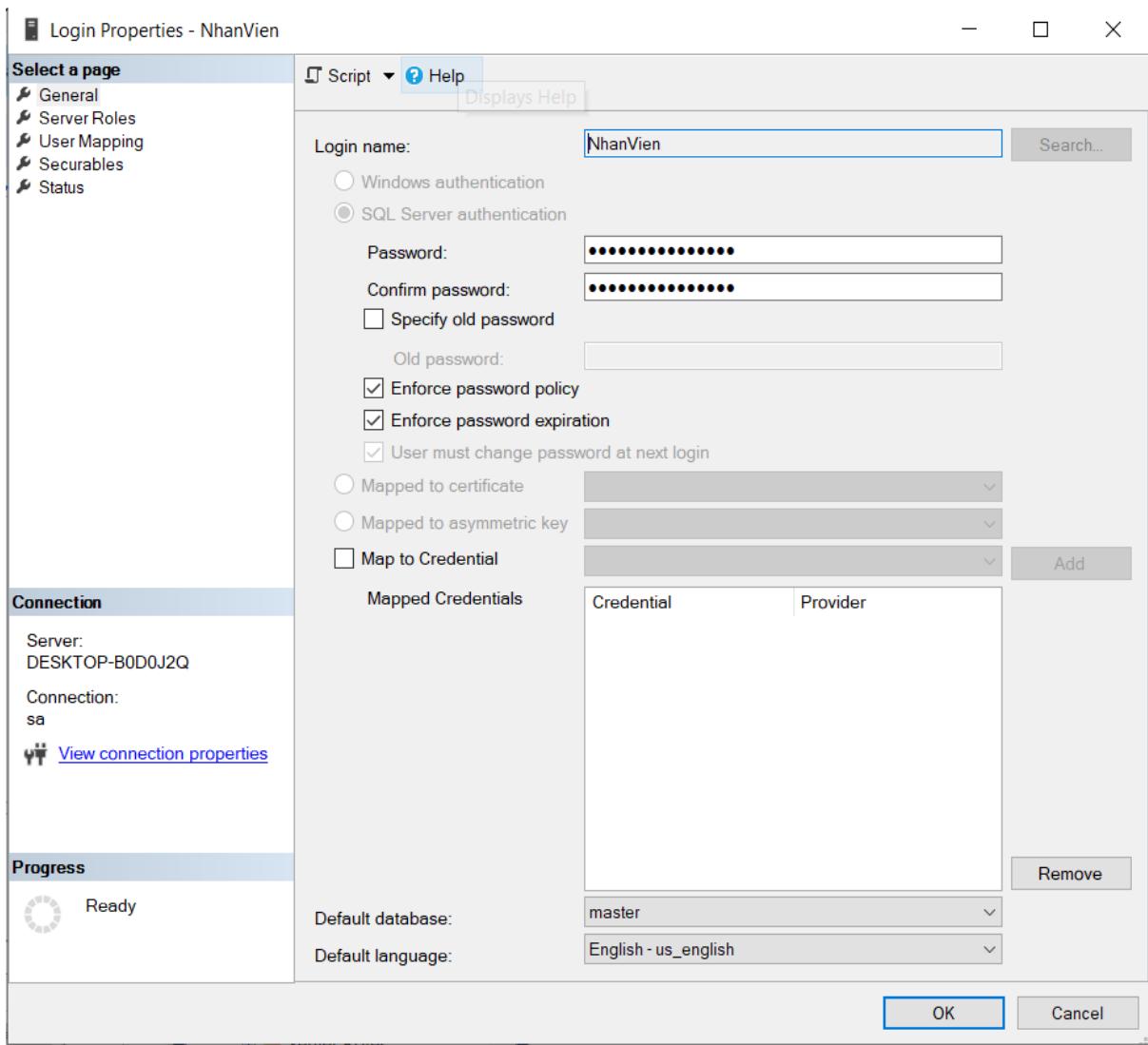
1. Mở SQL Server Management Studio (SSMS) và kết nối với server.
2. Trong khung bên trái, mở rộng mục **Security** và click chuột phải vào **Logins**, sau đó chọn **New Login** để tạo một tài khoản Nhân Viên.



3. Ở cửa sổ **Login - New**, nhập tên đăng nhập của người dùng trong ô **Login name**.
Bạn chọn **SQL Server Authentication** (xác thực bằng tài khoản SQL)

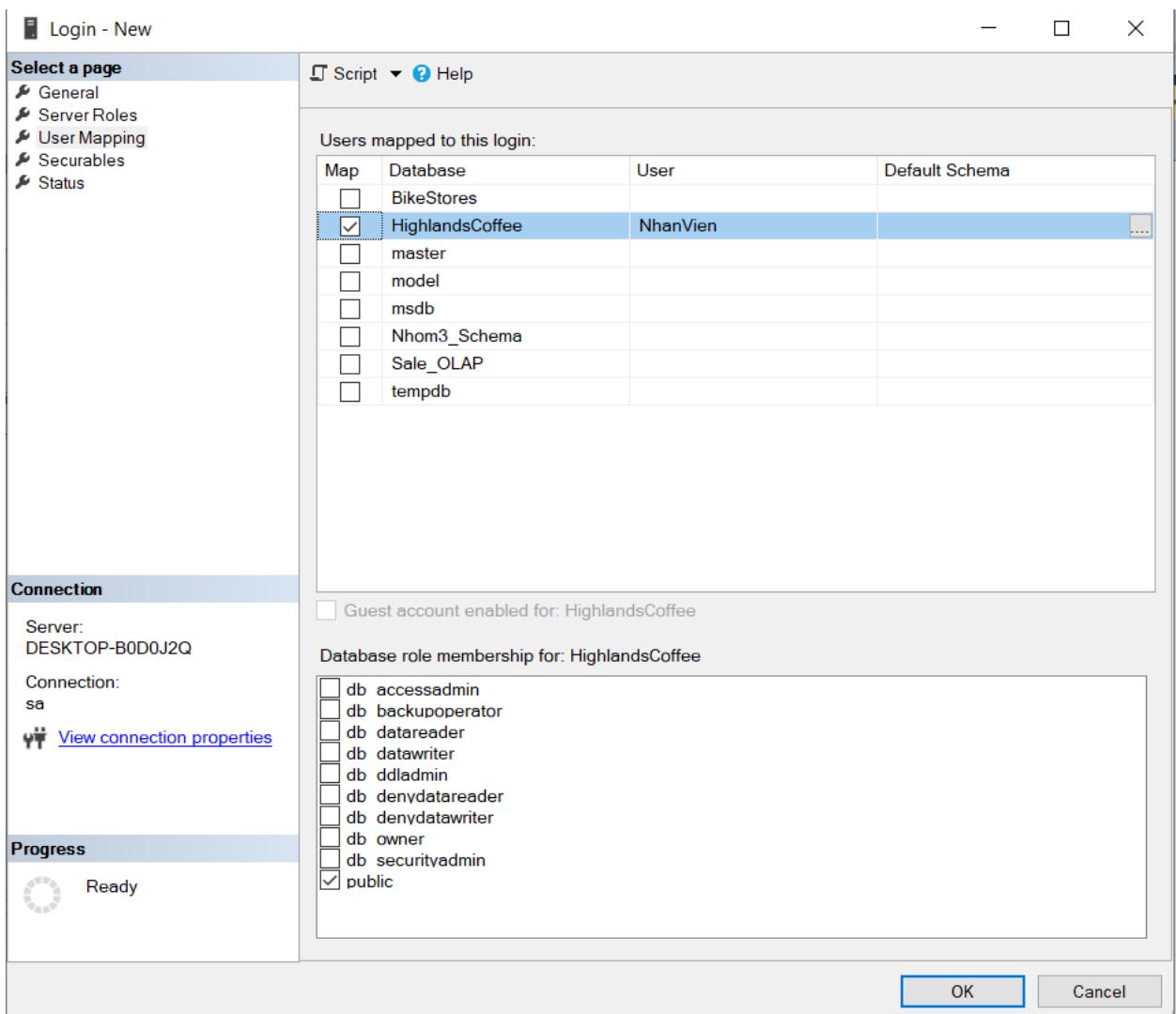
Login name: NhanVien

Mk: **nhanvien123@**



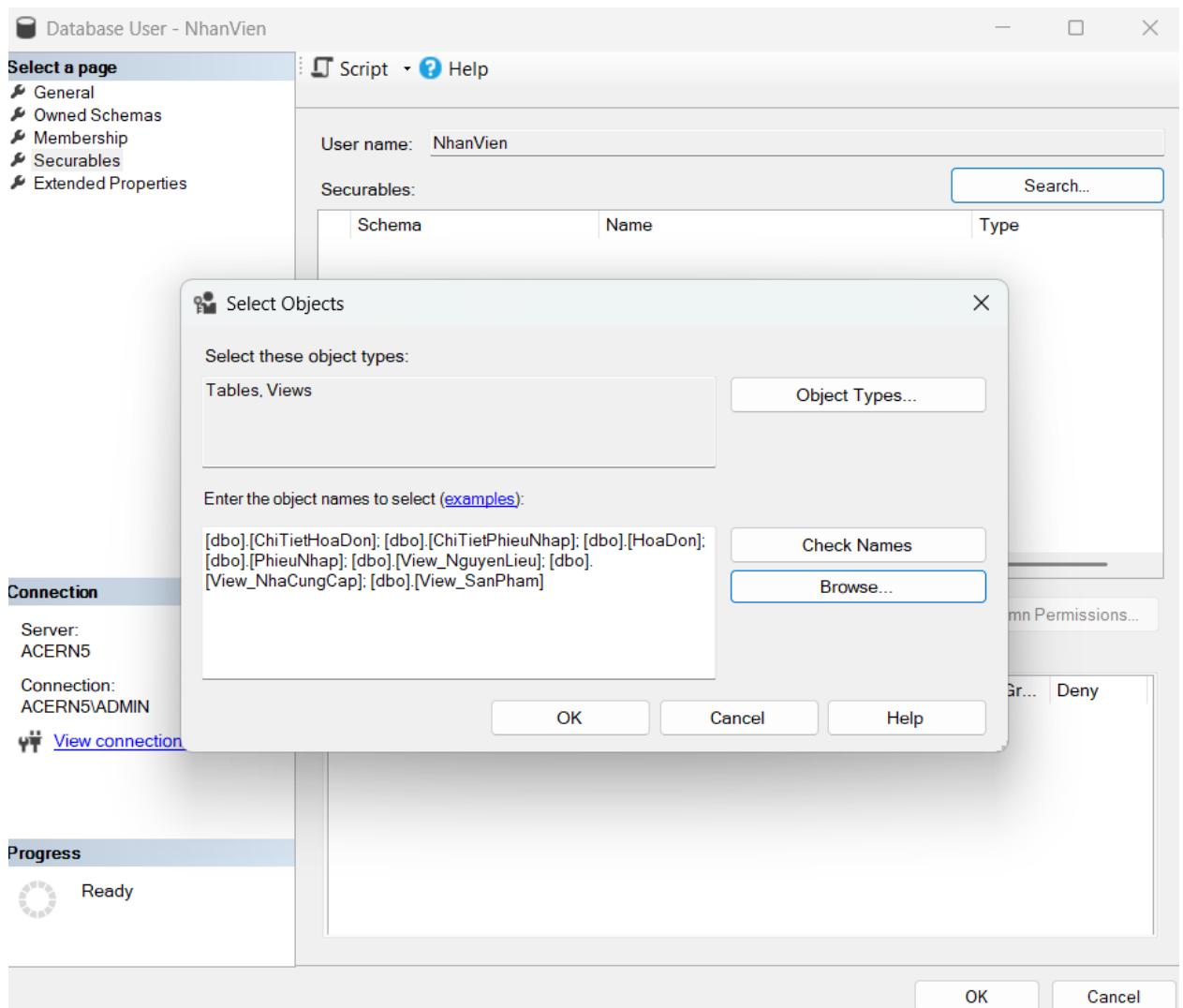
Bước 2: Gán User Mapping (Bản đồ vai trò người dùng trong cơ sở dữ liệu)

1. Vẫn trong cửa sổ **Login - New** hoặc **Properties** của nhân viên, vào tab **User Mapping**.
2. Tại đây, chọn cơ sở dữ liệu mà nhân viên này sẽ được truy cập. Khi bạn chọn một database (cơ sở dữ liệu), phần **Database role membership for:** sẽ hiển thị các vai trò trong cơ sở dữ liệu đó.
3. Gán vai trò cho nhân viên trong database:



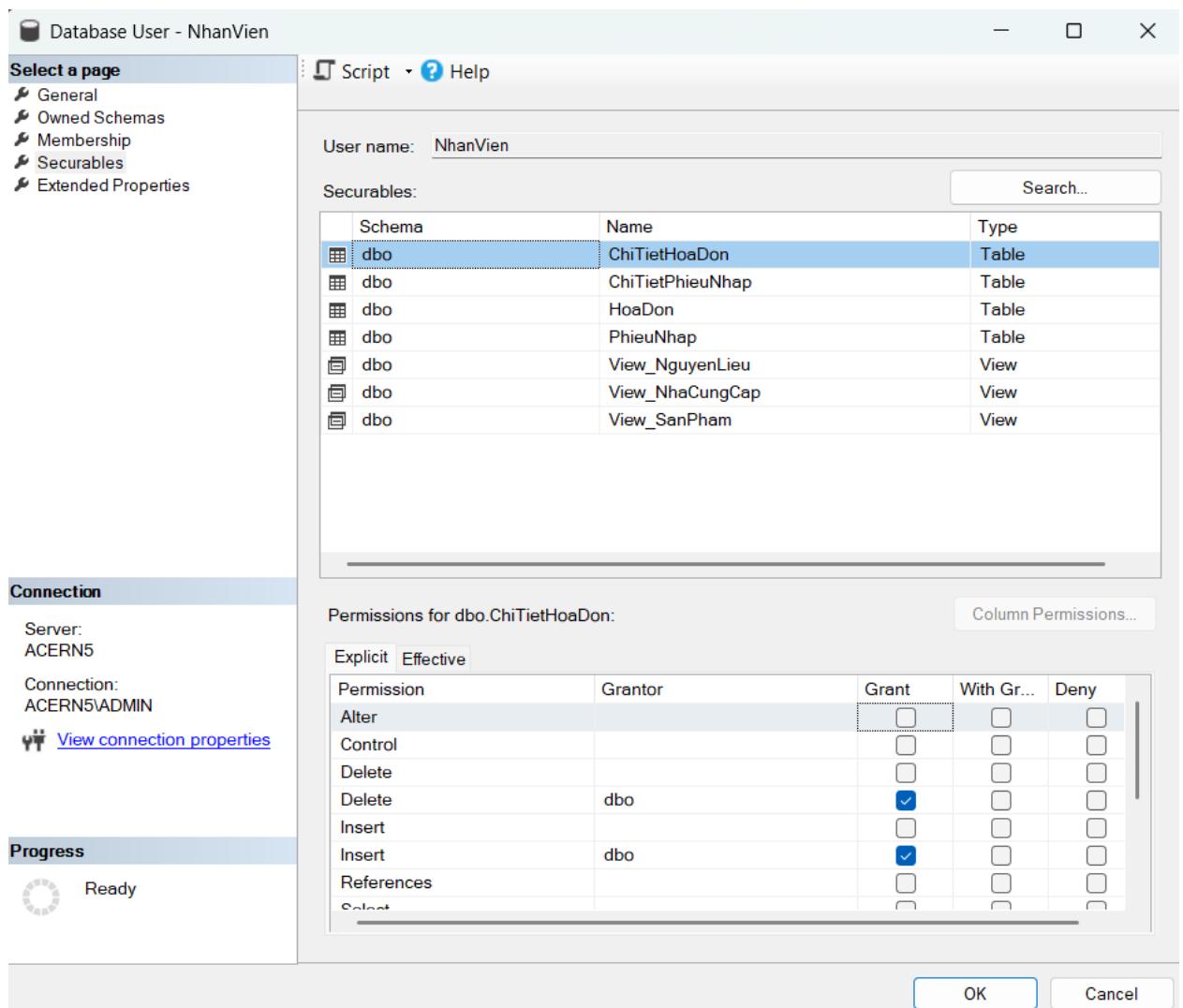
Bước 3: Thiết lập các quyền tùy chỉnh cho nhân viên trên từng bảng

1. Mở rộng database mà bạn muốn thiết lập quyền truy cập.
2. Mở rộng mục **Security** -> **Users** trong database đó.
3. Click chuột phải vào người dùng (User) mà bạn đã gán, chọn **Properties**.
4. Trong cửa sổ **Database User - Properties**, chọn tab **Securables**.
5. Nhấn **Search** và chọn **Specific objects → Object Types → Tables và Views → Ok**
 → **Browse** để chọn bảng cụ thể (HoaDon, ChiTietHoaDon, PhieuNhap, ChiTietPhieuNhap, View_NguyenLieu, View_NhaCungCap, View_SanPham)
 → rồi nhấn **OK**.

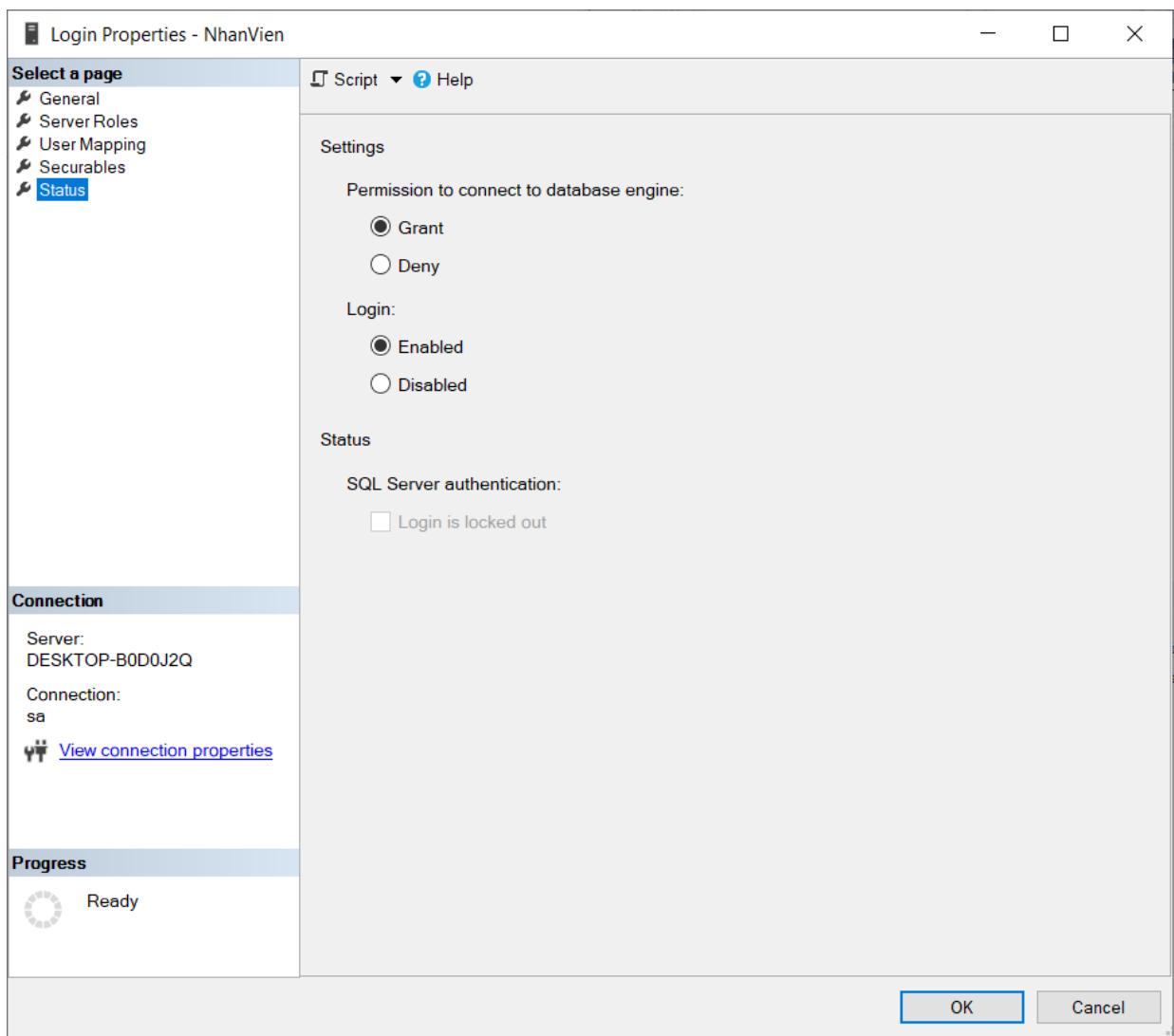


6. Sau khi đã chọn bảng, trong danh sách **Permissions** bên dưới, bạn có thể chọn các quyền cụ thể cho bảng đó như SELECT, INSERT, UPDATE, DELETE tùy theo yêu cầu của từng đối tượng:

Cho phép **SELECT, INSERT, DELETE** trên các bảng **HoaDon, ChiTietHoaDon, PhieuNhap, ChiTietPhieuNhap** và **SELECT** trên các bảng **View_NguyenLieu, View_NhaCungCap, View_SanPham**.



Bước 4: Thiết lập trạng thái



IV.4. Minh chứng:

Do nhân viên chỉ có quyền truy cập vào bảng *HoaDon*, *ChiTietHoaDon*, *PhieuNhap*, *ChiTietPhieuNhap*, *View_NguyenLieu*, *View_NhaCungCap*, *View_SanPham* nên không thể truy xuất thông tin từ các bảng khác.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the connection to 'ACERNS (SQL Server 16.0.1135.2 - ACERN)' and the database 'HighlandsCoffee'. The main window displays a query window titled 'SQLQuery3.sql - AC...fee (NhanVien (58))' with the following SQL code:

```
1 | Select * From View_SanPham
```

The results pane shows a table with two columns: 'TenSP' and 'DoGiaBan'. The data consists of 16 rows, each containing a product name and its price:

TenSP	DoGiaBan
San pham 1	771460
San pham 2	96942
San pham 3	37365
San pham 4	31971
San pham 5	25884
San pham 6	28856
San pham 7	96097
San pham 8	68962
San pham 9	23063
San pham 10	21255
San pham 11	42665
San pham 12	14188
San pham 13	44327
San pham 14	40341
San pham 15	66371
San pham 16	25691

A yellow status bar at the bottom indicates: 'Query executed successfully.' and 'ACERNS (16.0 RTM) NhanVien (58) HighlandsCoffee 00:00:00 1,000 rows'.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the connection to 'ACERNS (SQL Server 16.0.1135.2 - ACERN)' and the database 'HighlandsCoffee'. The main window displays a query window titled 'SQLQuery3.sql - AC...fee (NhanVien (58))' with the following SQL code:

```
1 | Select * From NhanVien
```

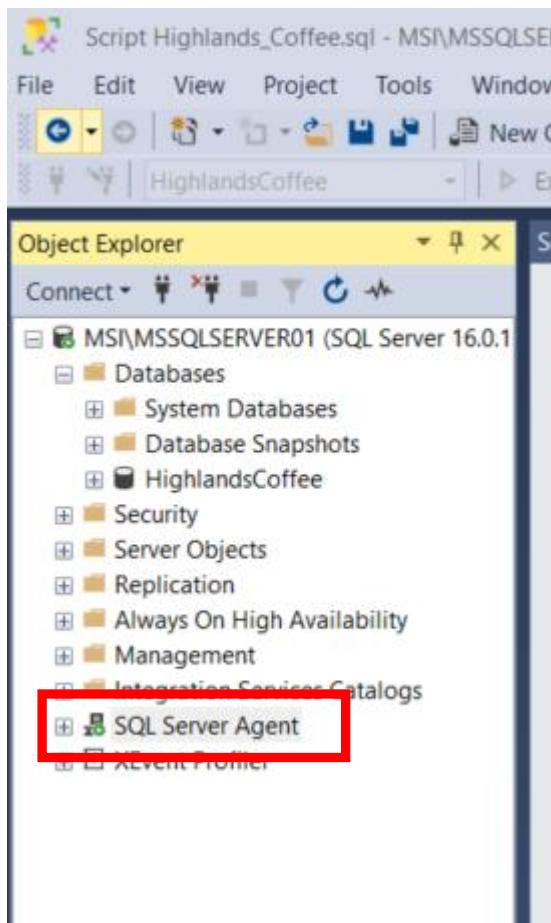
The results pane shows a single row of messages indicating a permission denied error:

Messages
Msg 229, Level 14, State 5, Line 1 The SELECT permission was denied on the object 'NhanVien', database 'HighlandsCoffee', schema 'dbo'.

A yellow status bar at the bottom indicates: 'Query completed with errors.' and 'ACERNS (16.0 RTM) NhanVien (58) HighlandsCoffee 00:00:00 0 rows'.

V. Backup dữ liệu tự động

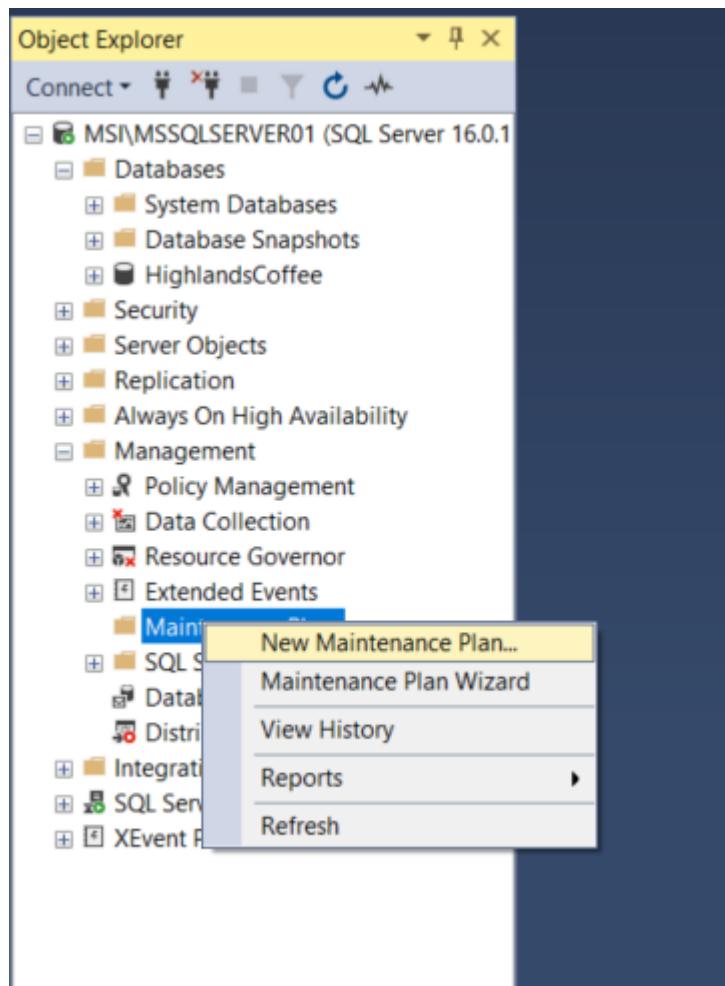
Lưu ý: Đảm bảo **SQL Server Agent** luôn trong trạng thái hoạt động



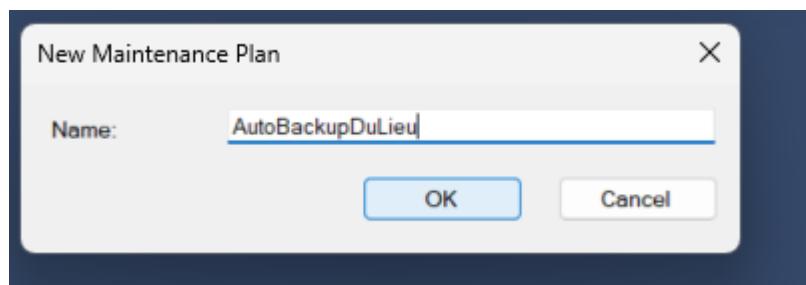
V.1. Tạo cơ chế Backup dữ liệu tự động (AutoBackup)

Bước 1: Tạo “New Maintenance Plan”

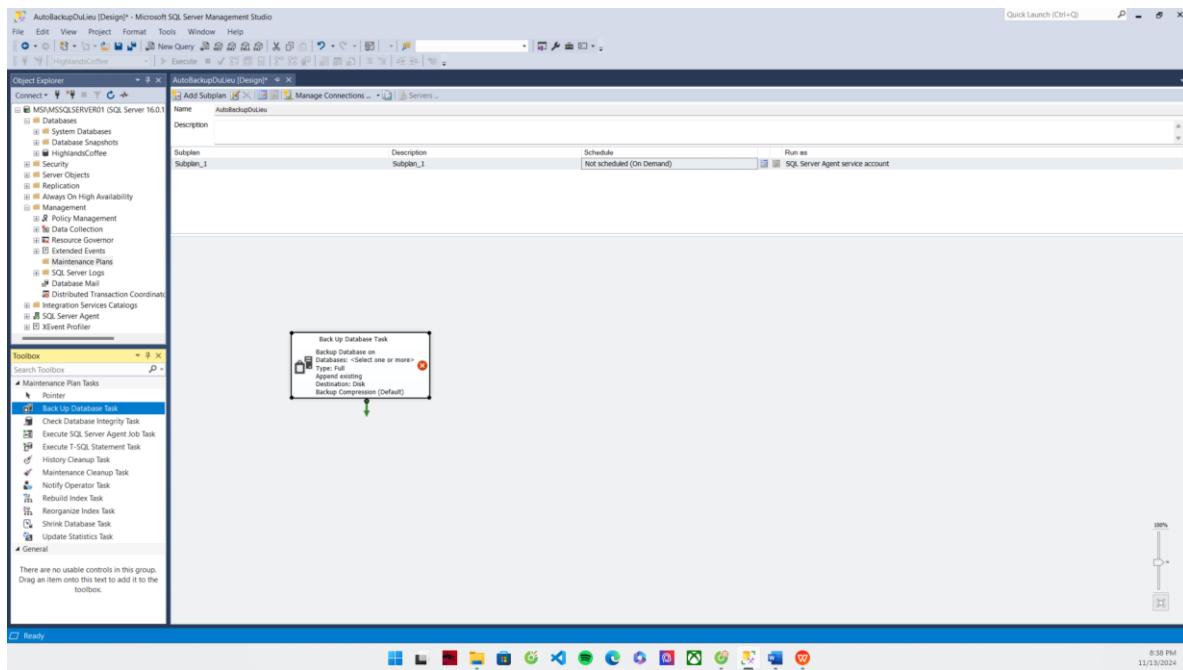
- Mở rộng mục **Management** → Click chuột phải vào **Maintenance Plan** → Chọn **New Maintenance Plan**



- Thực hiện lưu tên mới cho Plan



- Ở thanh Toolbox bên trái → Kéo Task “**Back Up Database Task**” và thả vào ô bên trái.



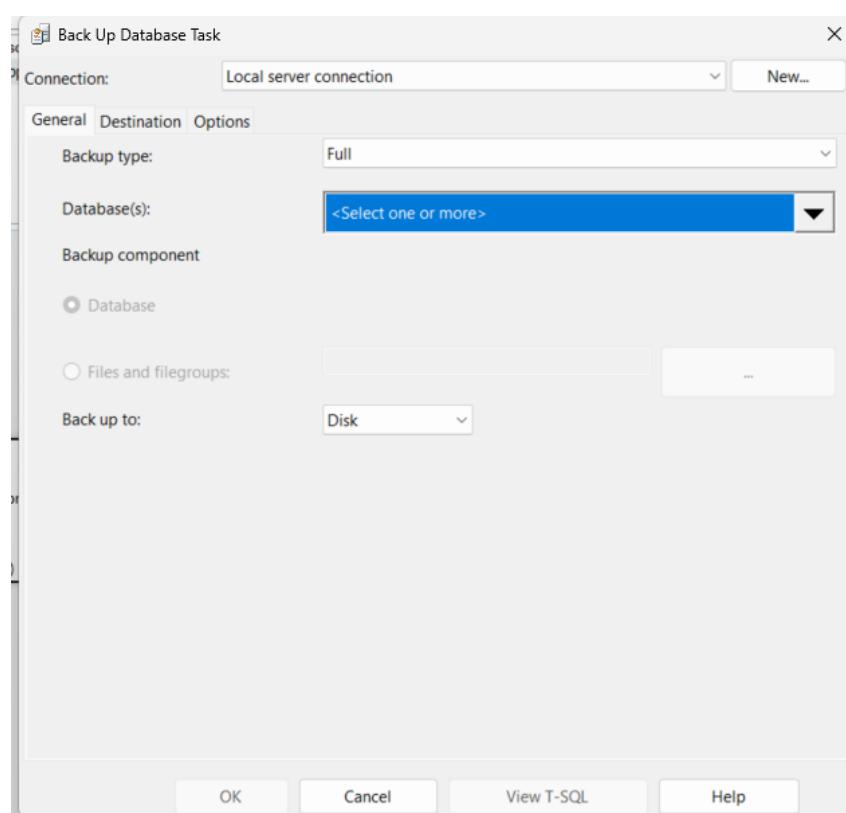
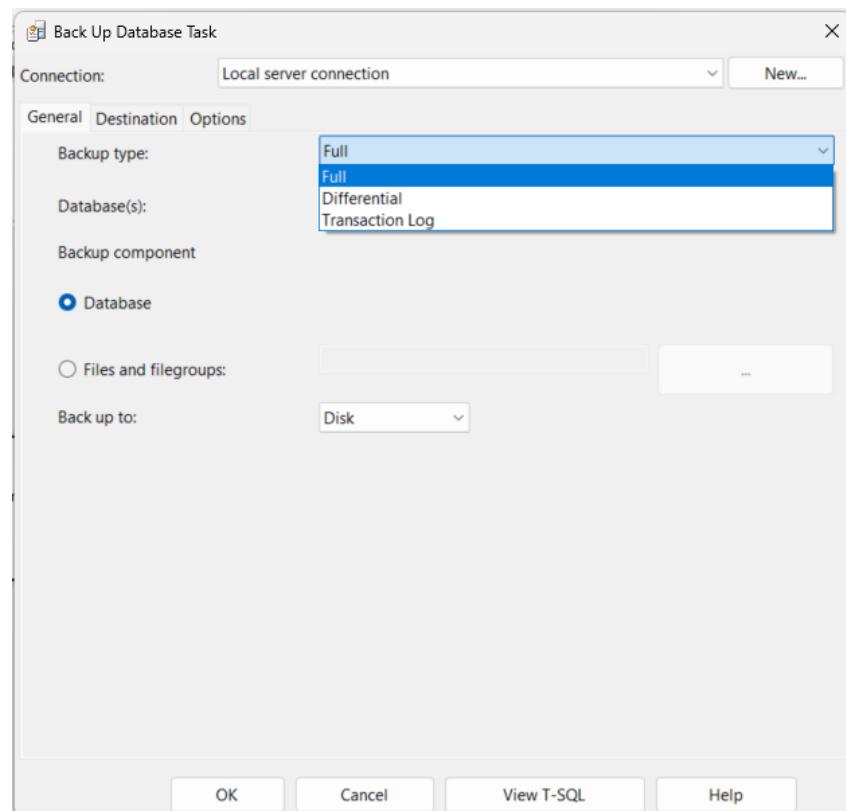
Bước 2: Tiến hành tùy chỉnh các tùy chọn trong 3 mục: GENERAL, DESTINATION, OPTIONS

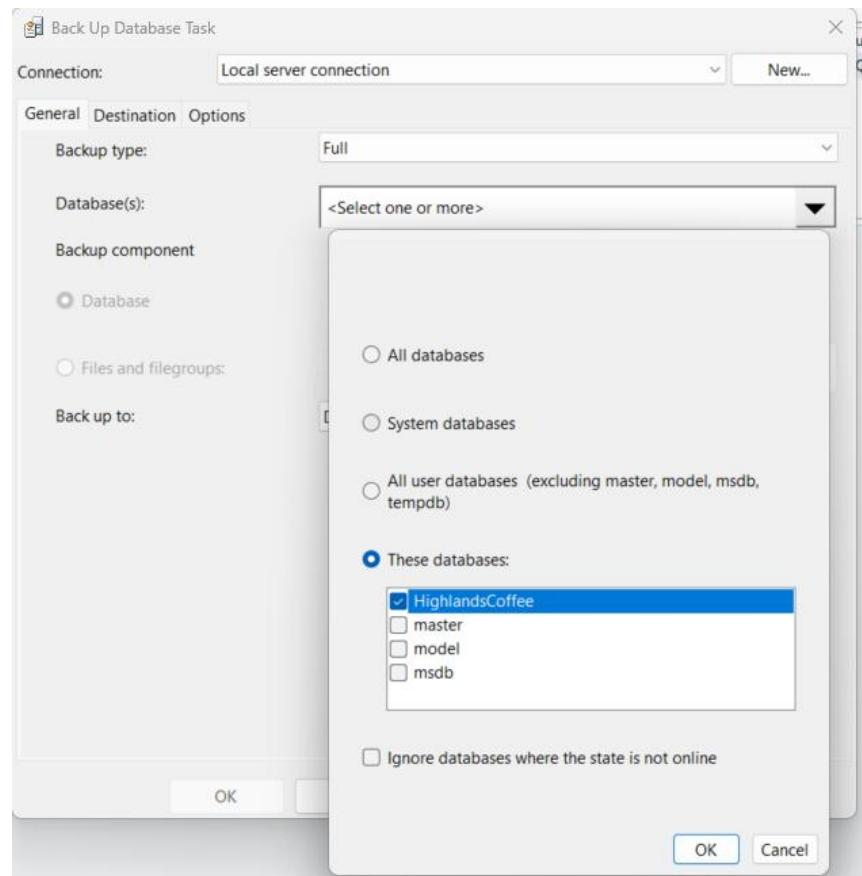
- Click đúp vào Task để hiện thị các tùy chỉnh
 - General:

Backup type:

- + *Full (sao lưu đầy đủ)*: thực hiện backup toàn bộ các đối tượng, bảng hệ thống và dữ liệu trong cơ sở dữ liệu.

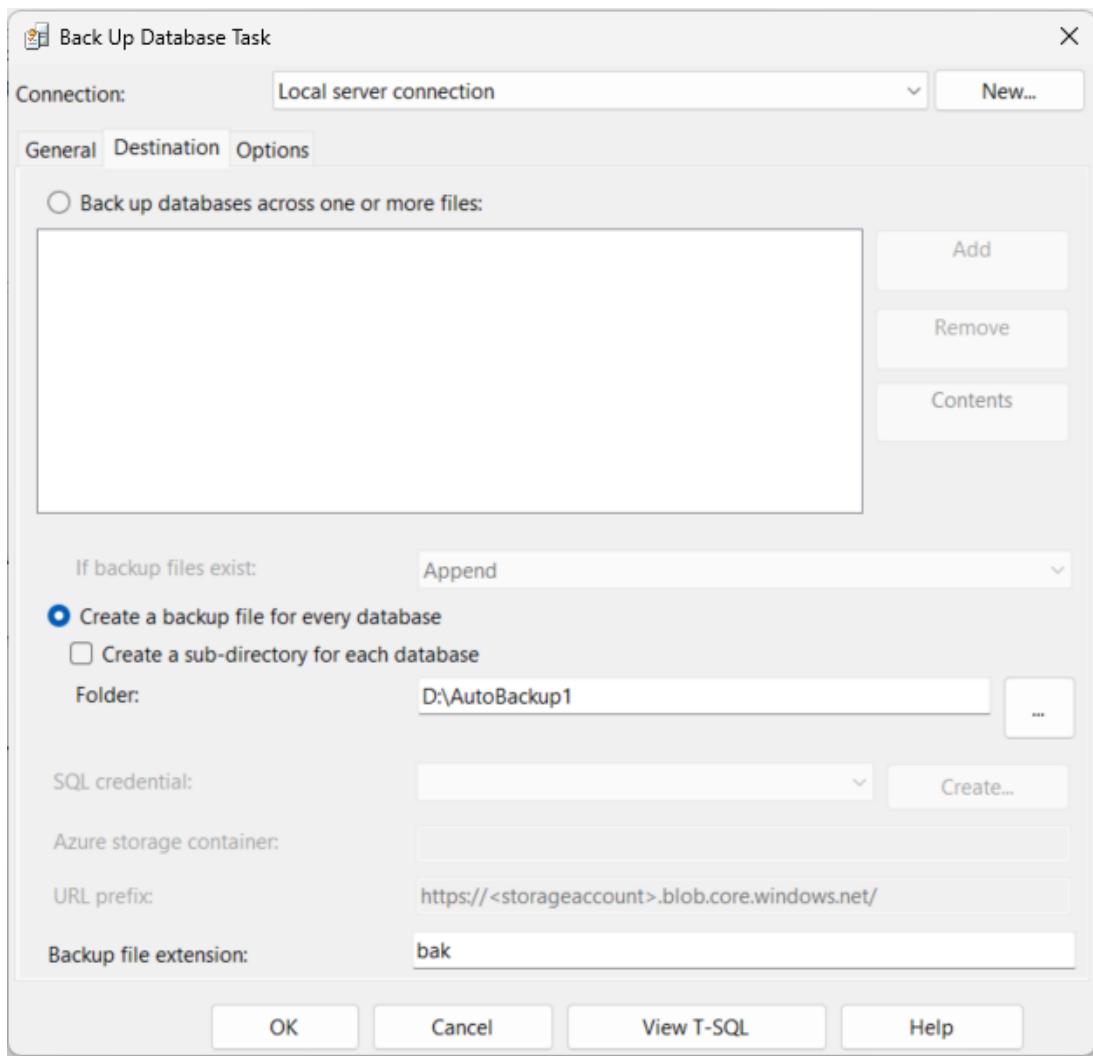
Database: tùy chọn 1 hoặc nhiều database khác nhau



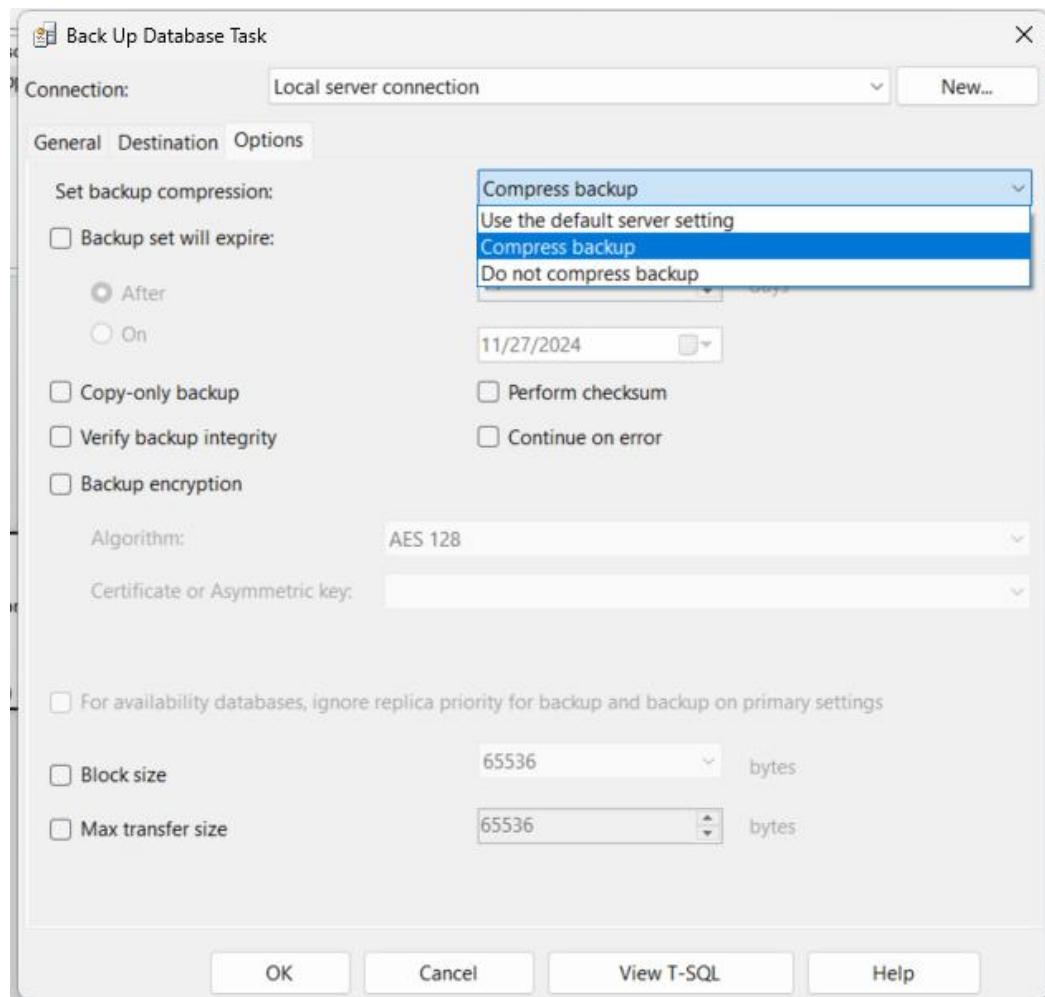


- Database sử dụng để Backup ở đây là **HighlandsCoffee**

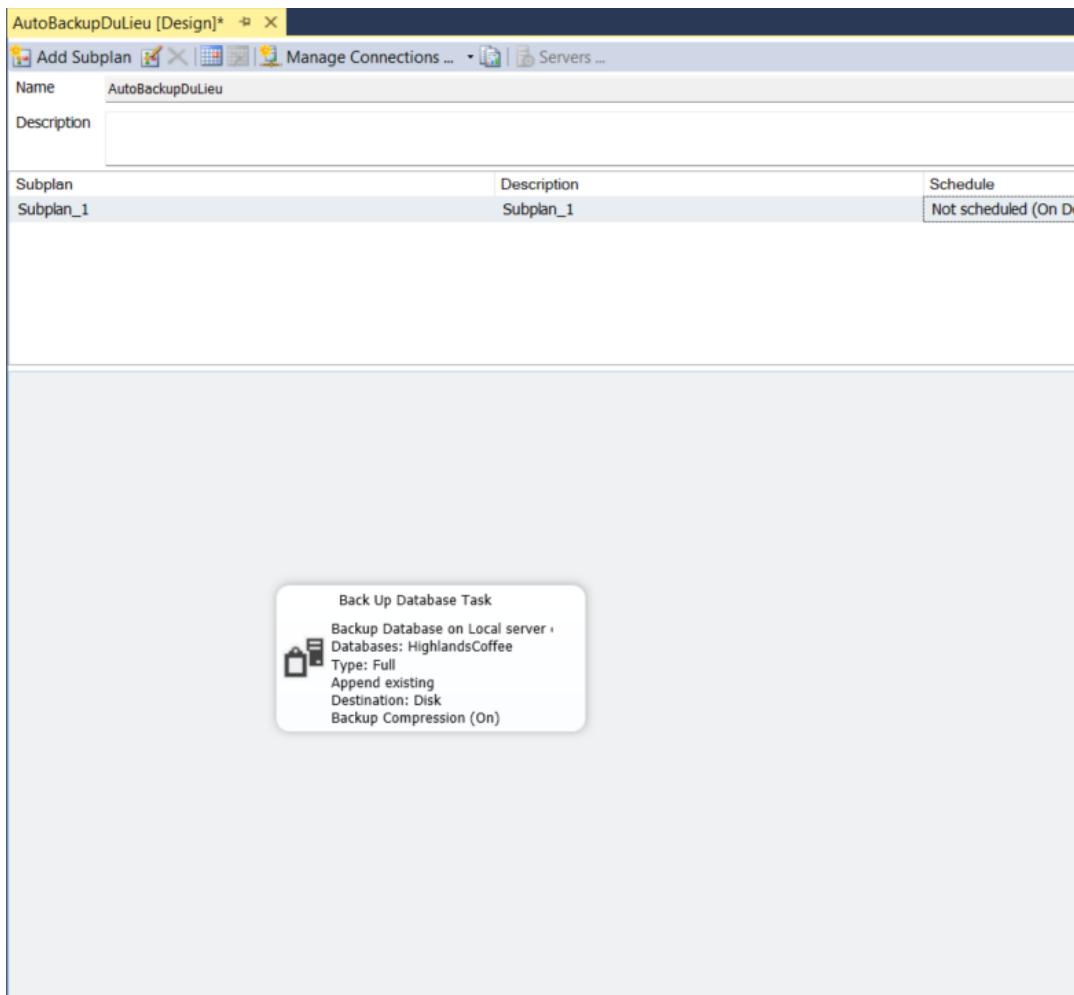
- Destination:
 - + Tạo 1 folder mới để thực hiện việc Backup dữ liệu
 - + Folder được tạo ở đây là **D:\AutoBackup1**
 - + Tùy chỉnh định dạng file : **file.bak**



- Options:
 - + Set Backup compression: **Compress backup**
 - + Giúp nén dữ liệu Database Tối ưu dung lượng



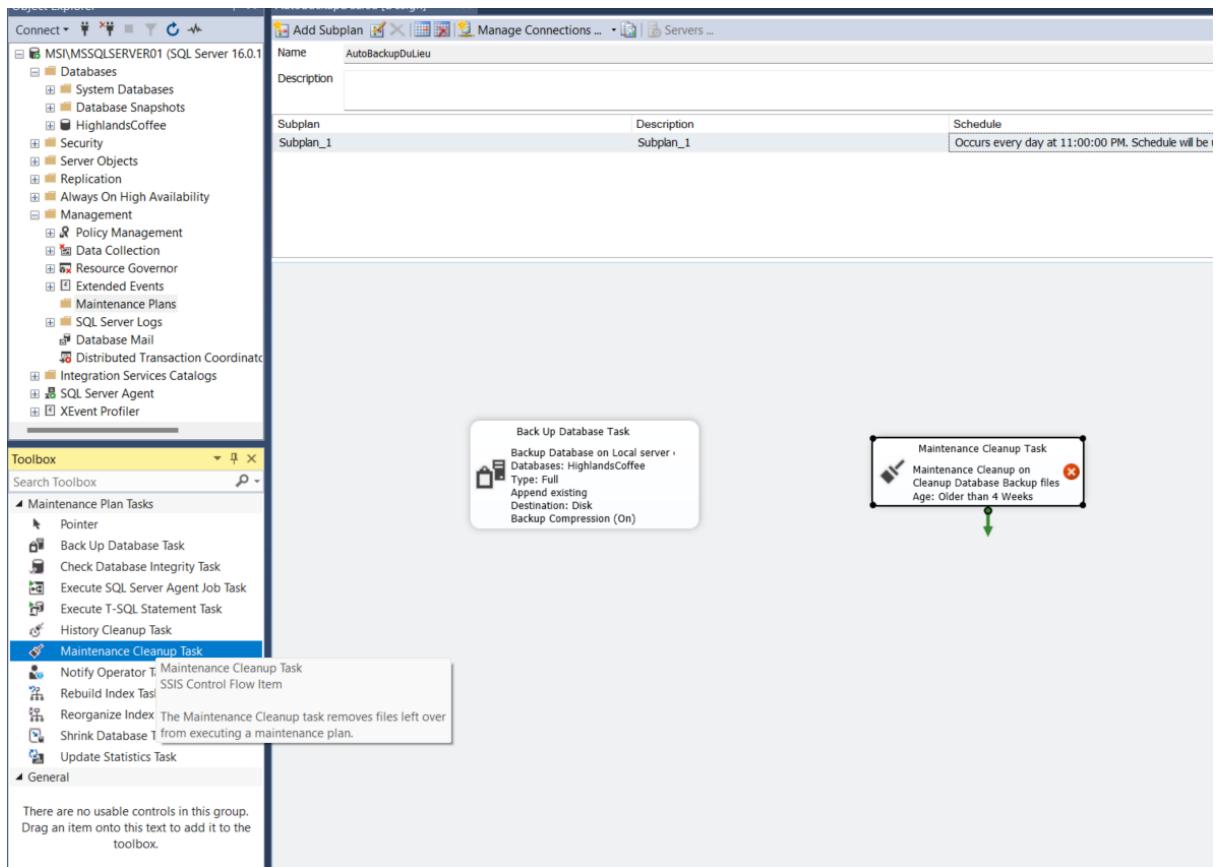
- Done



V.2. Tạo cơ chế xóa dữ liệu đã Backup

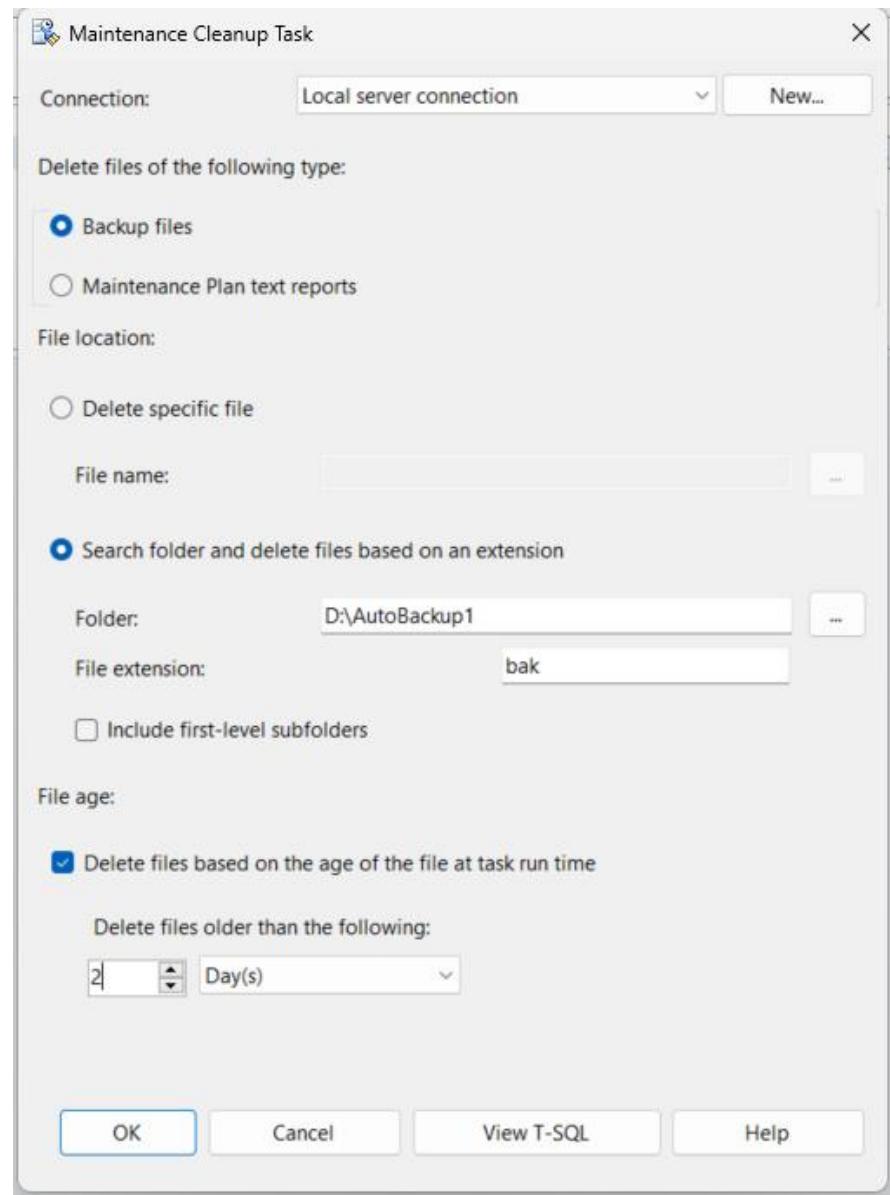
Bước 1: Tạo “MAINTENANCE CLEANUP TASK”

Ở thanh Toolbox bên trái → Kéo Task “Maintenance Cleanup Task” và thả vào ô bên trái.



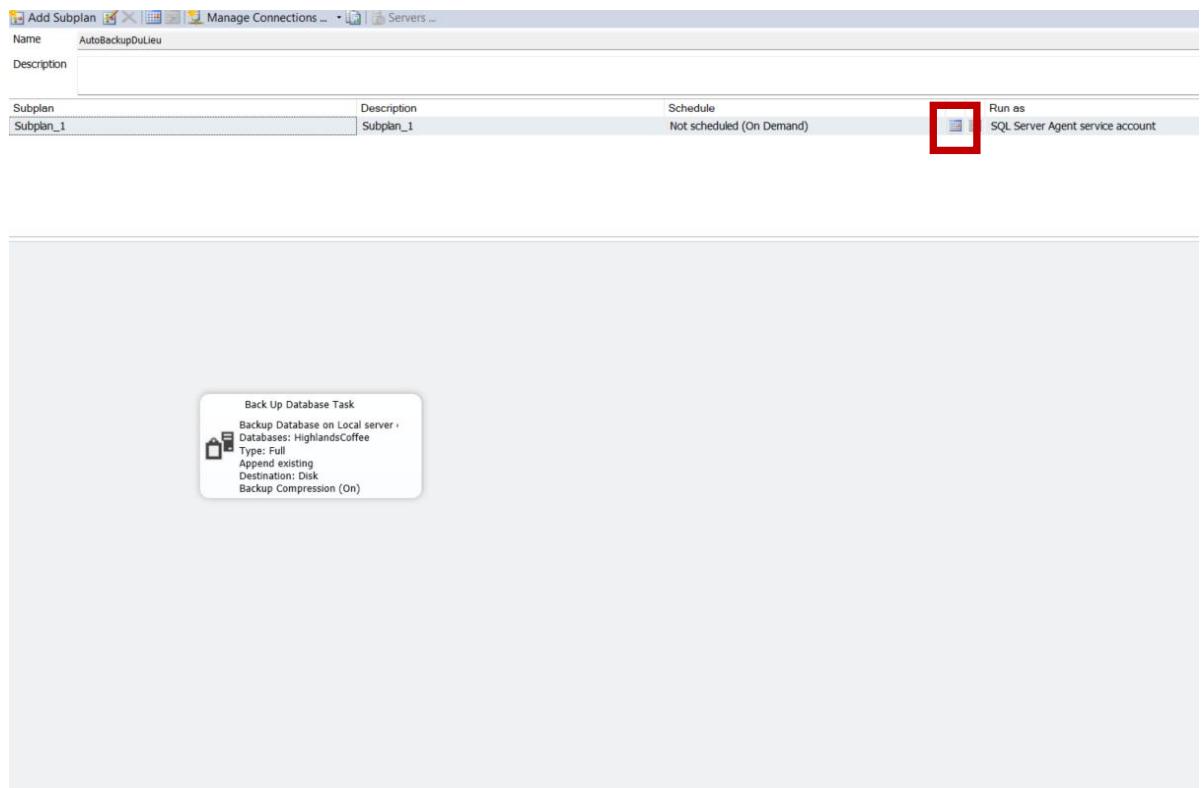
Bước 2: Tùy chỉnh các yếu tố

- Click đúp vào Task để hiển thị các tùy chỉnh
- Như hình: **Sẽ thực hiện xóa đi những file đã có thời gian tồn tại trong folder hơn 2 ngày**



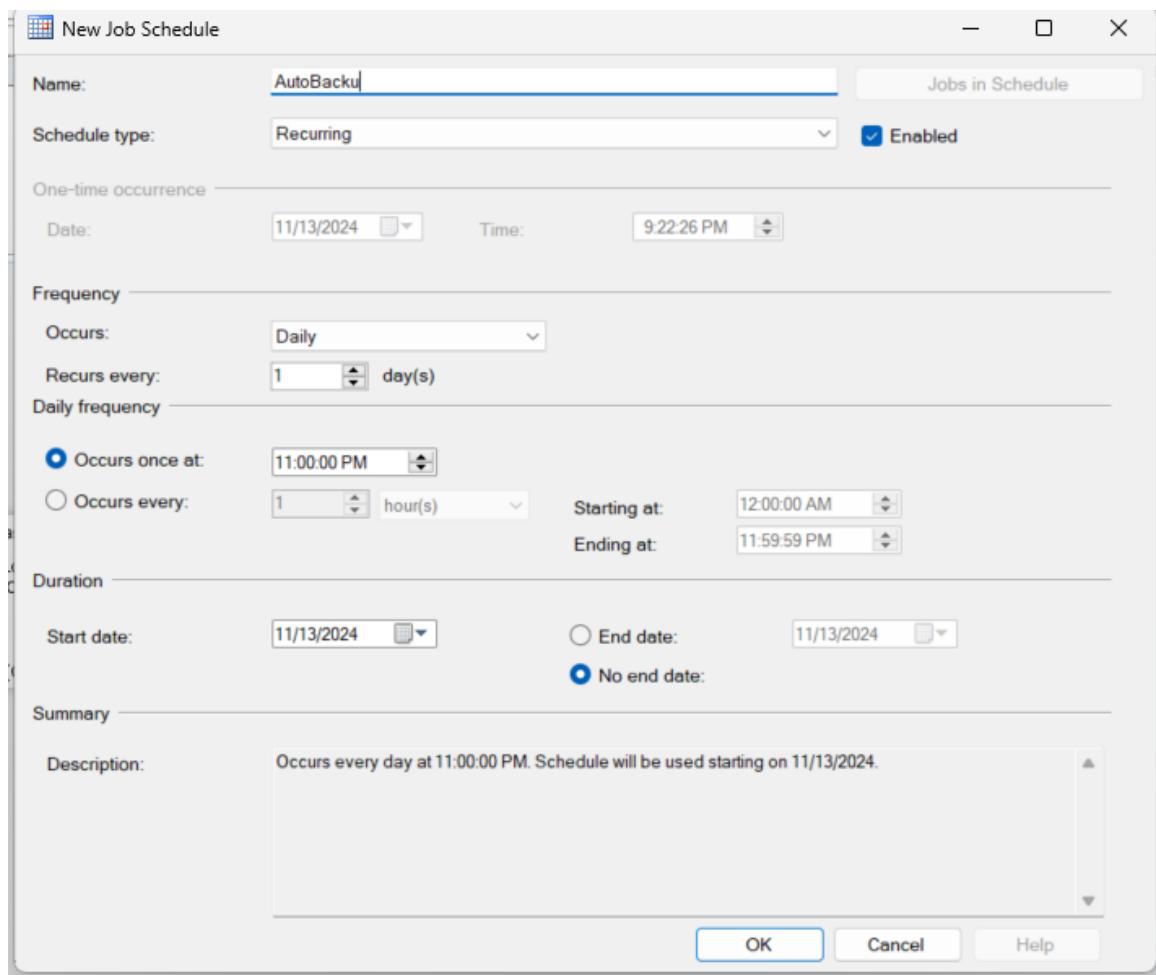
V.3. Set thời gian thực hiện Plan

Bước 1: Chọn vào icon như hình



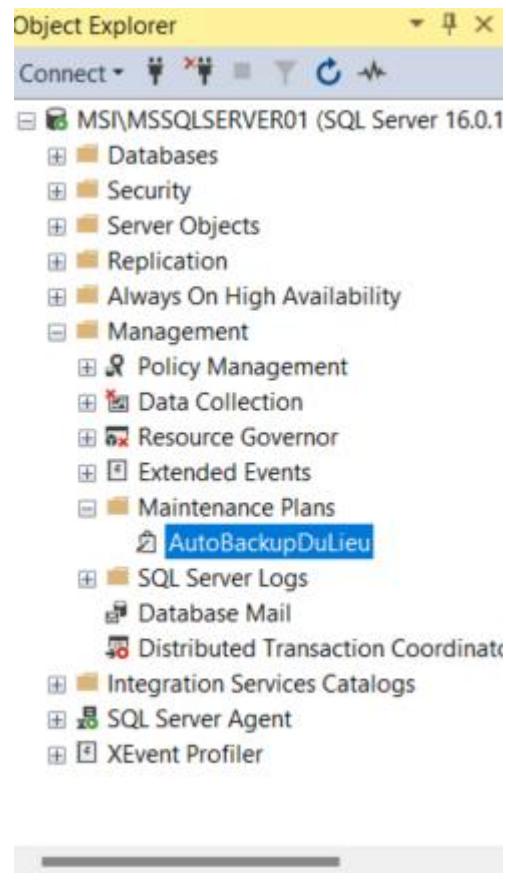
Bước 2: Thực hiện tùy chỉnh các yếu tố trong bảng

- Như hình:
 - + Start date: 11/13/2024.
 - + Sẽ thực hiện Plan mỗi ngày vào 11:00:00pm



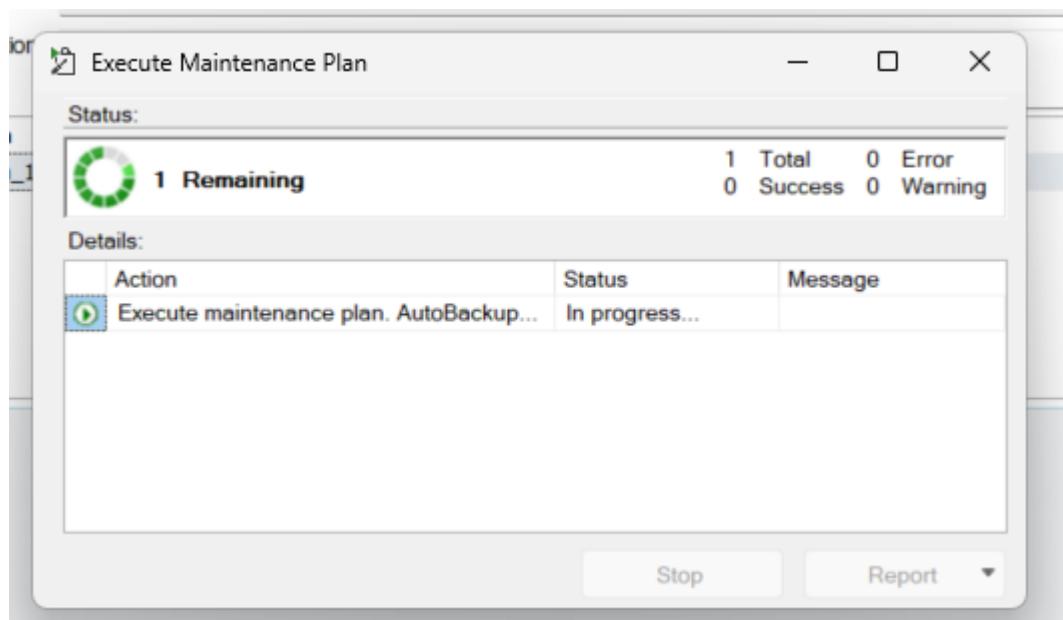
V.4. Lưu Plan và Execute

Bước 1: Lưu Plan

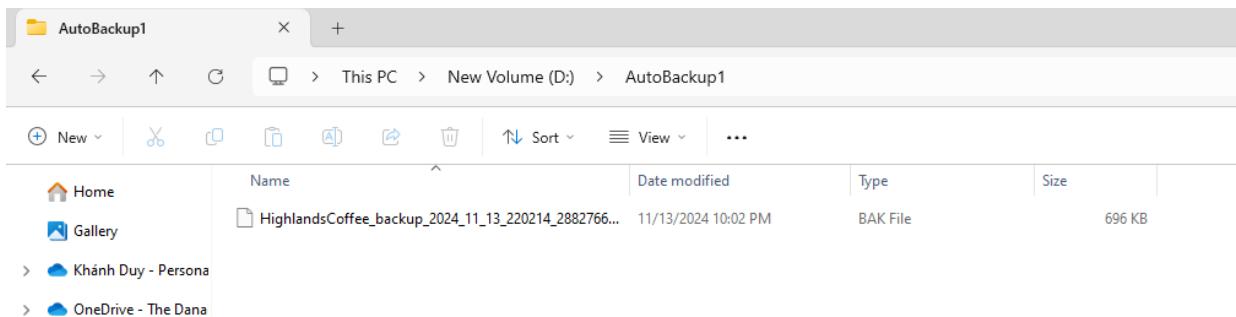


Bước 2: Thực hiện Execute:

- Click chuột phải vào Plan “AutoBackupDuLieu” → Chọn Execute



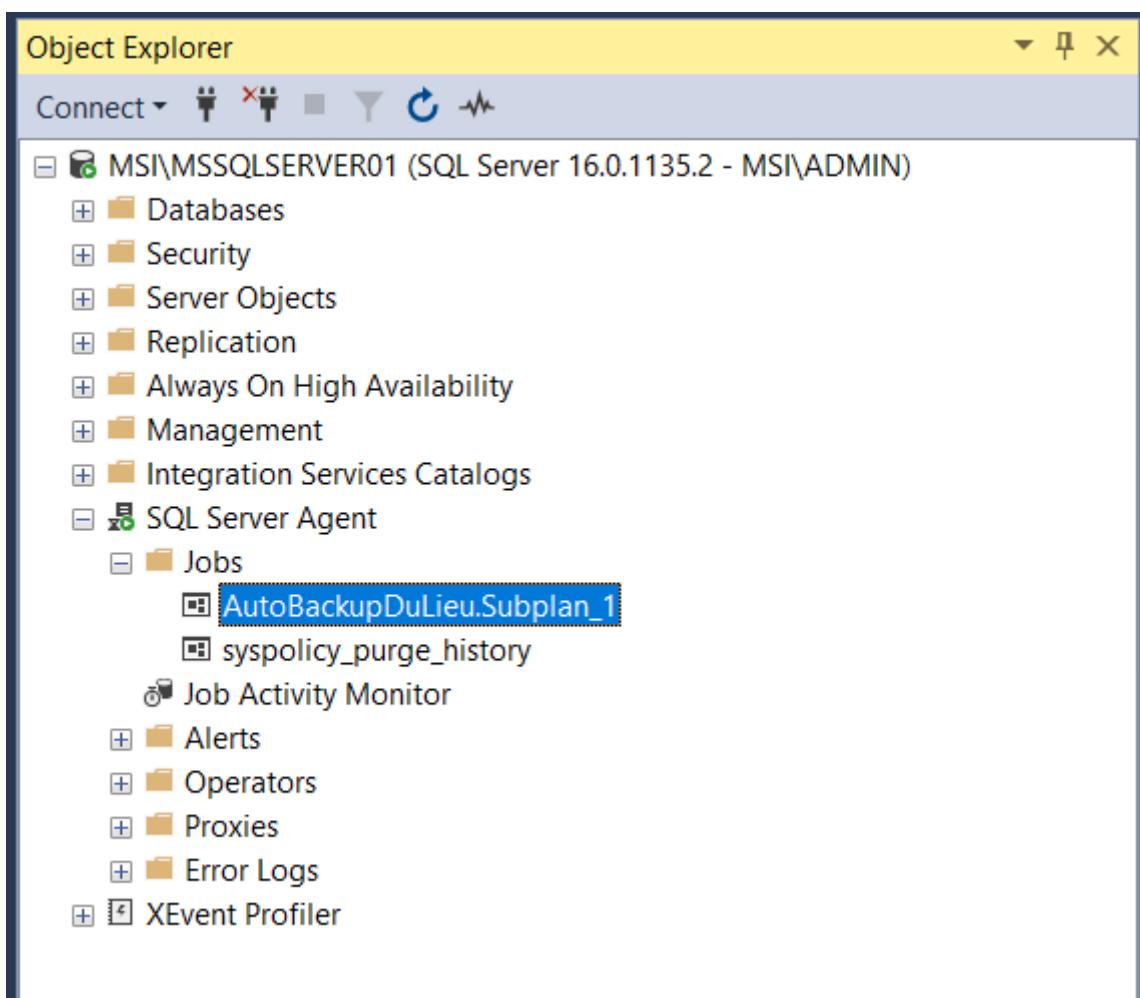
- Execute thành công → file Backup sẽ hiển thị trong folder D:\AutoBackup1



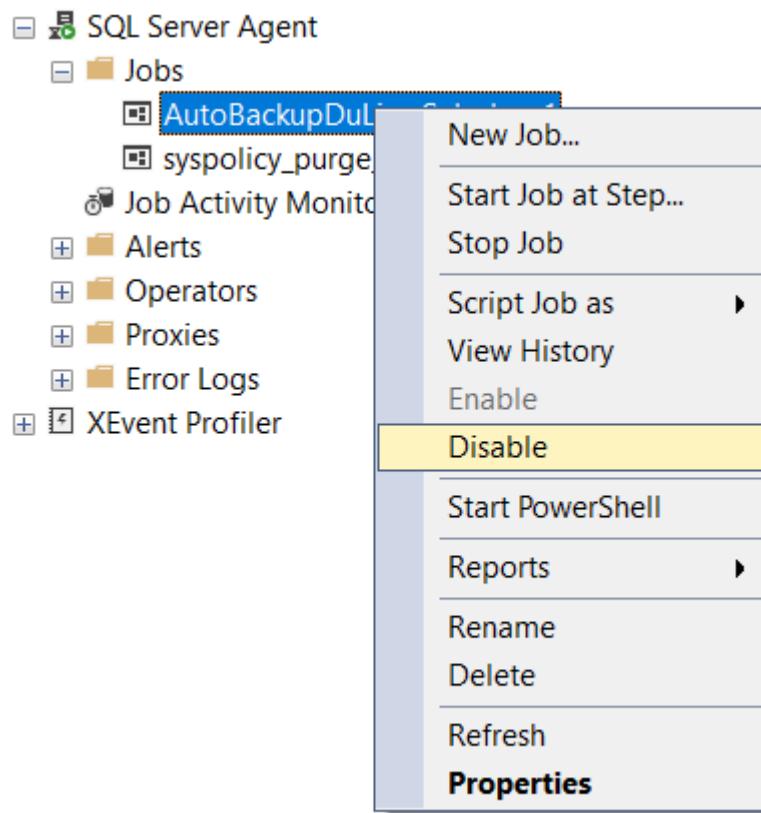
V.5. Dừng hoạt động backup tự động

Bước 1: Trong Object Explorer

- Chọn SQL Server Agent → Chọn Jobs



Bước 2: Click chuột phải vào Job vừa tạo (AutoBackupDuLieu) → Chọn Disable



VI. Phương án giải quyết TỐT NHẤT cho lượng dữ liệu rất lớn, không đủ lưu trữ trong một ổ đĩa:

Để giải quyết vấn đề lưu trữ dữ liệu lớn trong SQL Server với hệ thống quản lý quán Highlands Coffee, phương án tối ưu nhất là áp dụng kỹ thuật "Partitioning" (phân chia bảng). Phương pháp này sẽ giúp quản lý dữ liệu trên nhiều ổ đĩa, cải thiện hiệu suất truy vấn, giảm thiểu chi phí bảo trì và tăng tính sẵn sàng của hệ thống.

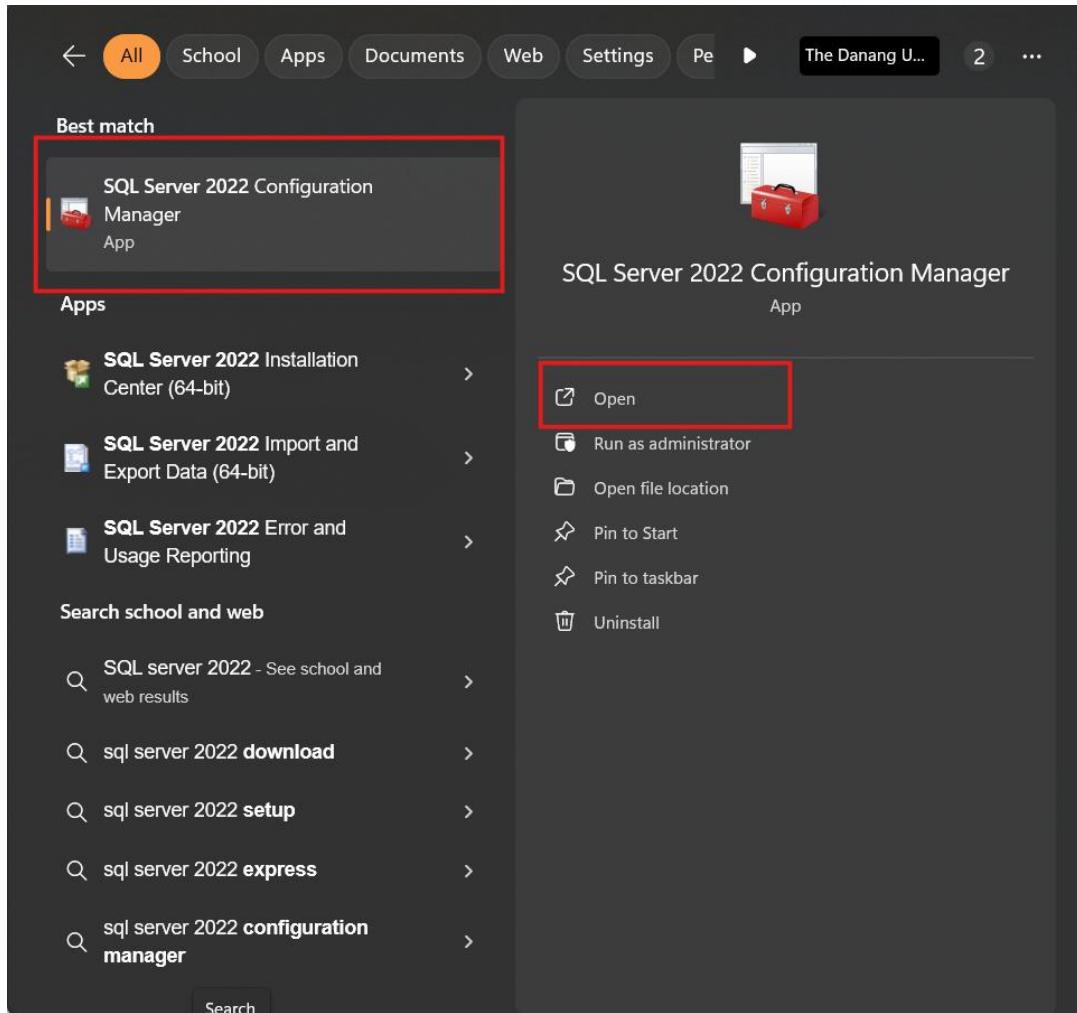
VI.1. Lý do lựa chọn Partitioning

1. Hiệu suất cao: Partitioning cho phép chia dữ liệu lớn thành các phần nhỏ hơn, giúp tối ưu hóa truy vấn khi chỉ cần truy cập vào từng phân vùng (partition) nhất định thay vì toàn bộ bảng.
2. Dễ dàng quản lý dữ liệu trên nhiều ổ đĩa: Mỗi partition có thể được lưu trữ trên ổ đĩa khác nhau, giúp giảm tải và tận dụng tối đa không gian lưu trữ.
3. Quản lý tốt theo thời gian: Các bảng lớn như bảng hóa đơn (HoaDon, ChiTietHoaDon) và phiếu nhập (PhieuNhap, ChiTietHoaDon) có thể được

phân chia theo thời gian (ví dụ: theo tháng hoặc năm), giúp dễ dàng quản lý, truy xuất và lưu trữ dữ liệu lịch sử.

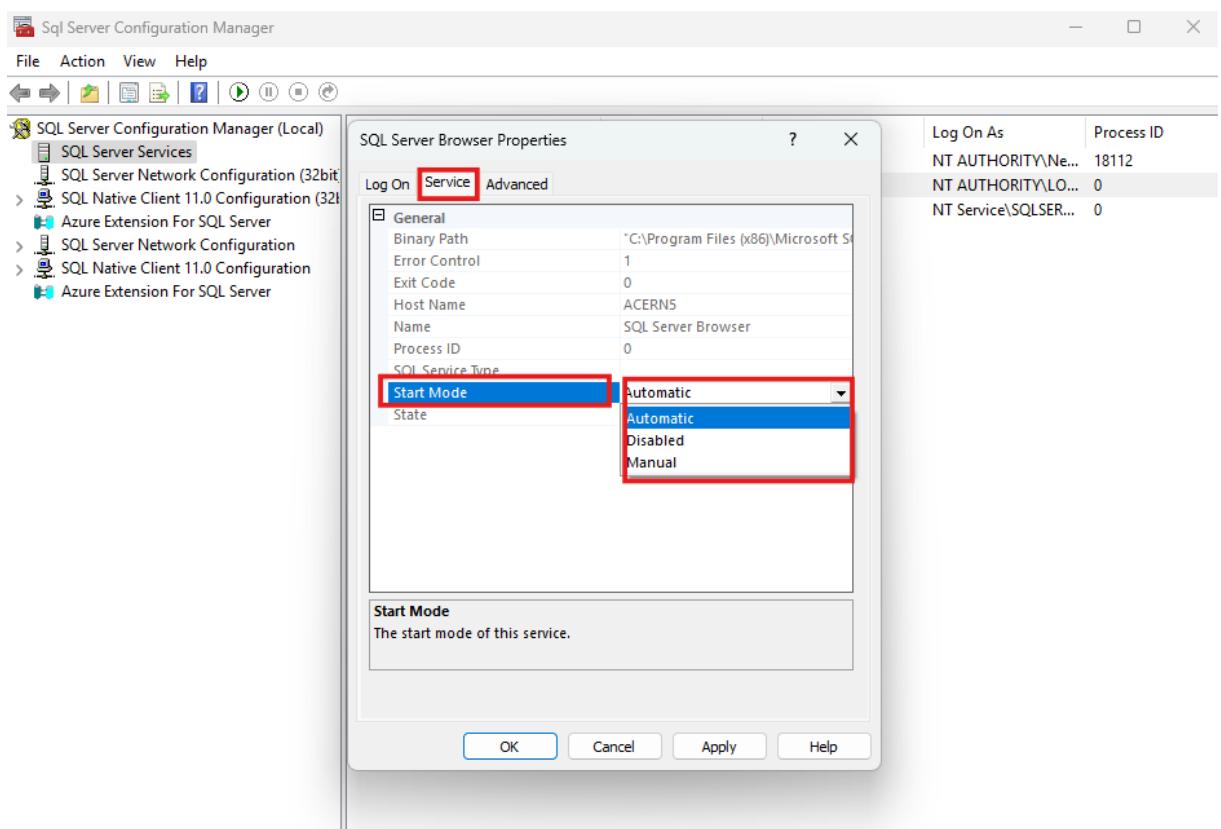
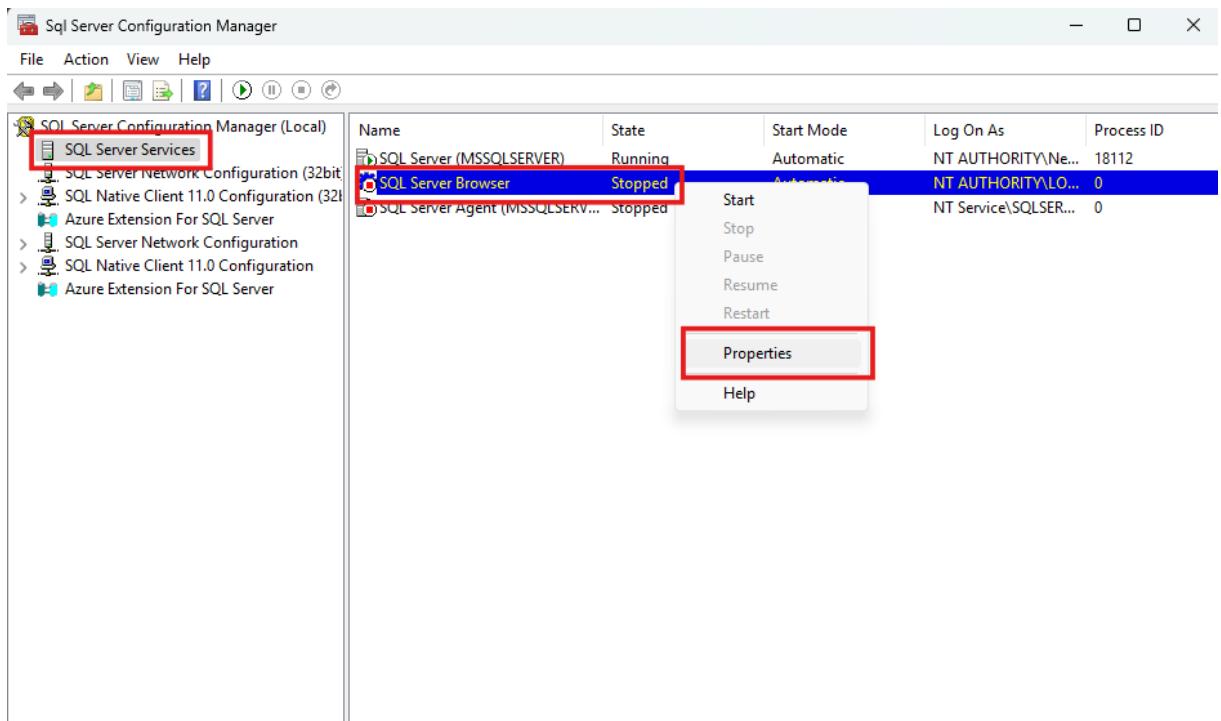
VI.2. Cách hosting Server trong mạng LAN/ WIFI và kết nối Server:

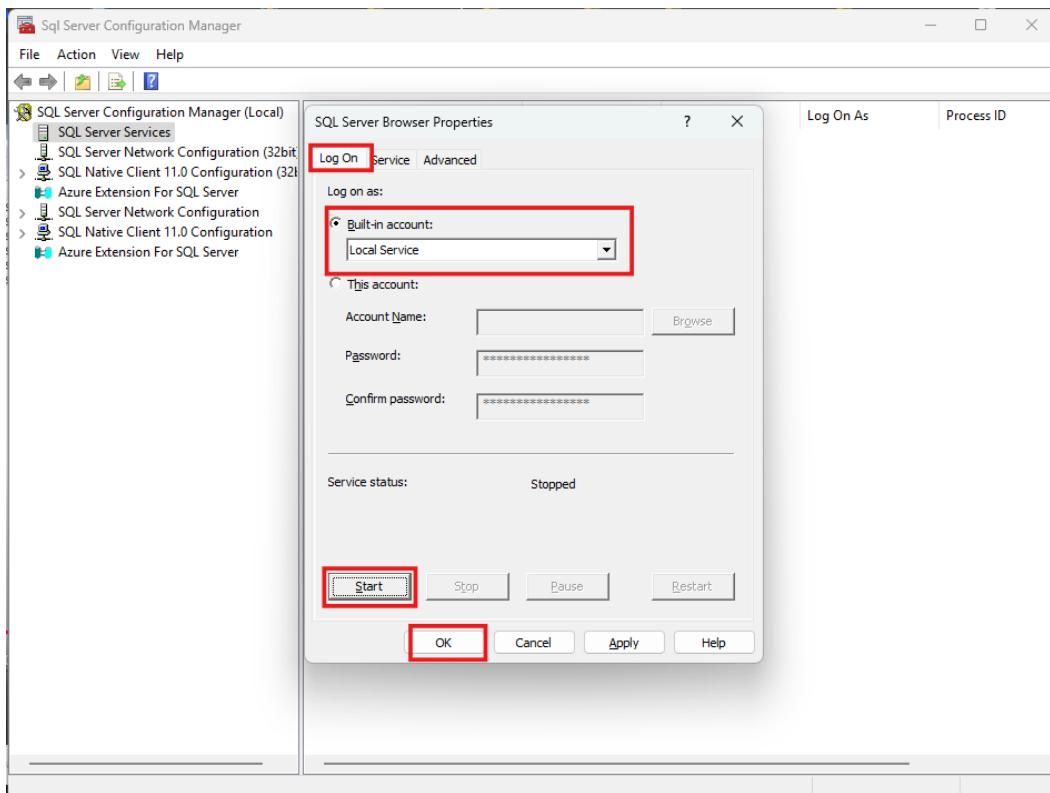
Bước 1: Mở ứng dụng SQL Server Configuration Manager



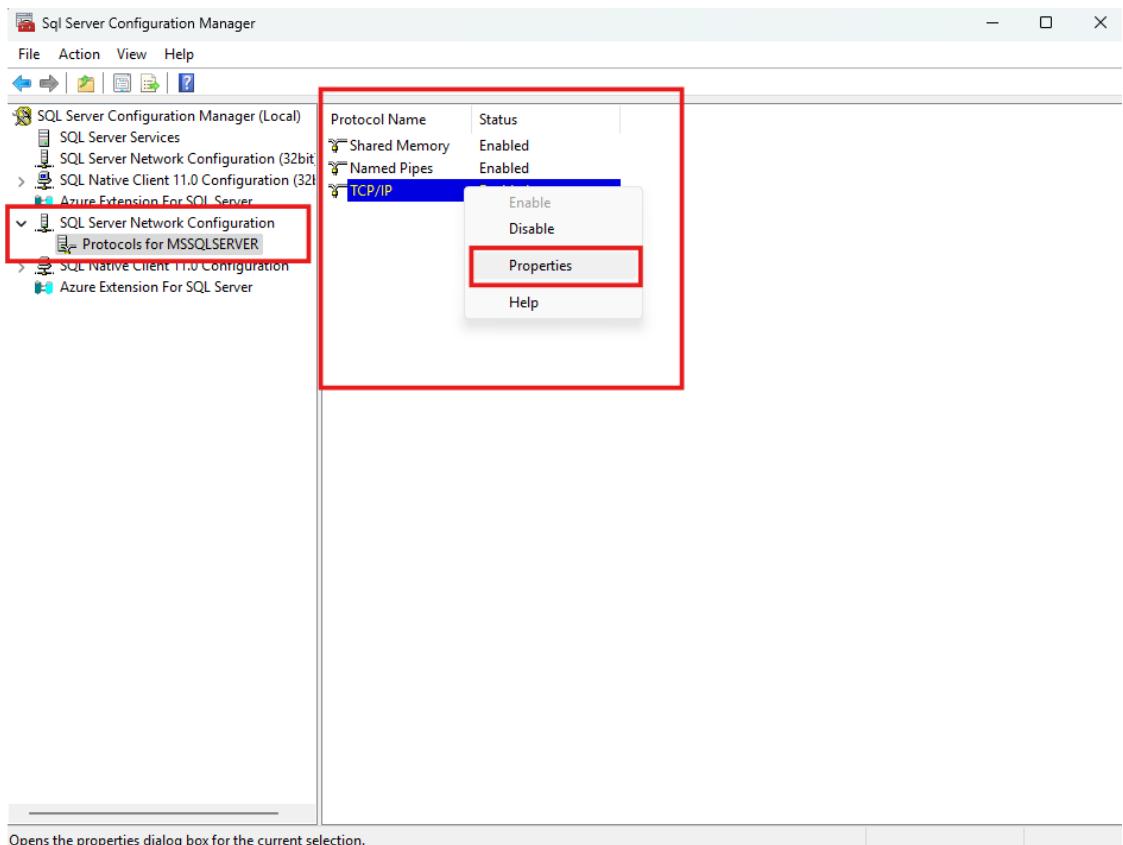
Bước 2: Ở SQL Server Services đảm bảo SQL Server Browser luôn trong trạng thái (State) là Running :

- Chuột phải SQL Server Browser → Properties → Service → Start Mode → Automatic → Log On → Start → OK

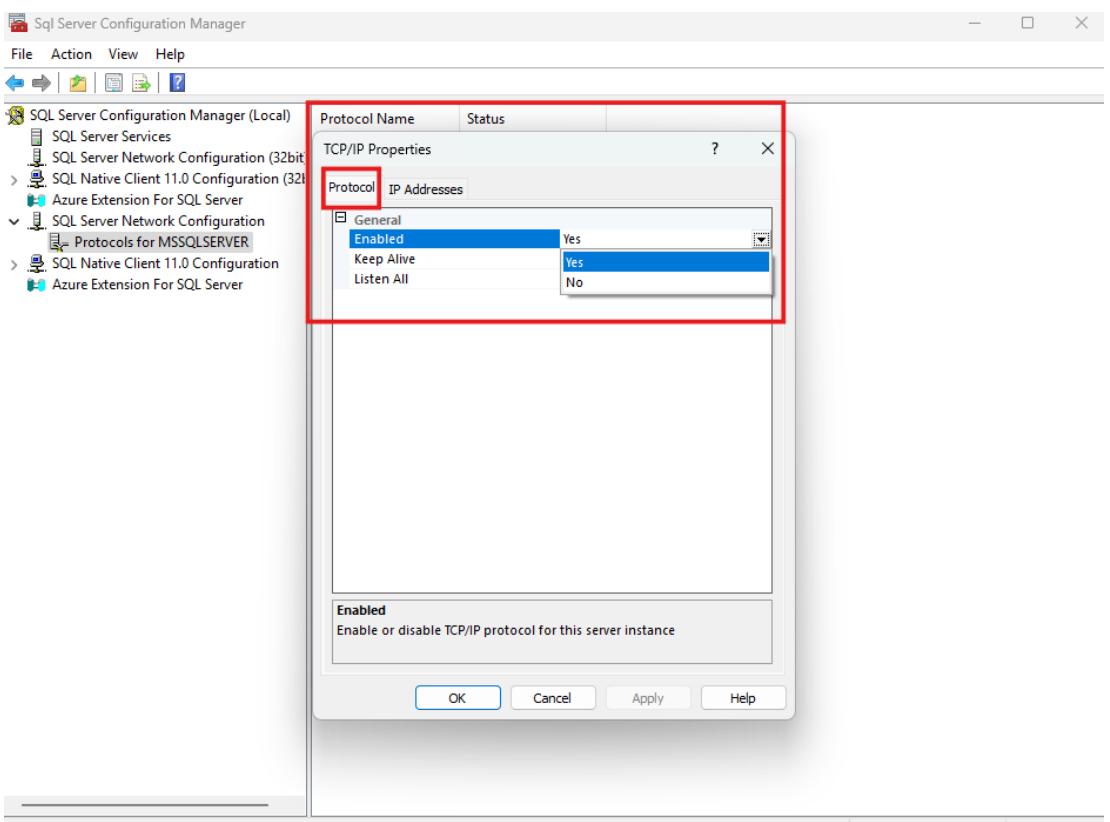


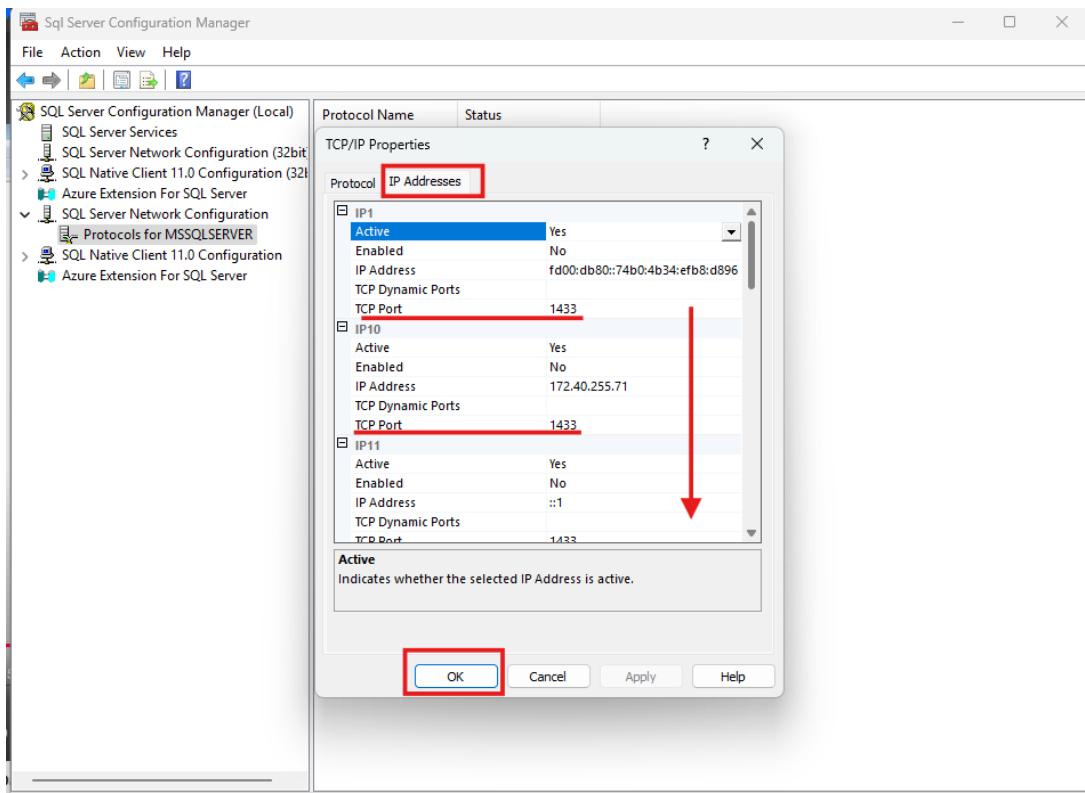


- Chuyển đến **SQL Server Network Configuration** → **Protocols for MSSQLSERVER** → Đảm bảo cả 3 protocol đều trong trạng thái **Enabled**: **TCP/IP** → **Properties** → **Enabled: YES** → **IP Addresses** → Đảm bảo các **TCP Port** là **1433** → **OK**

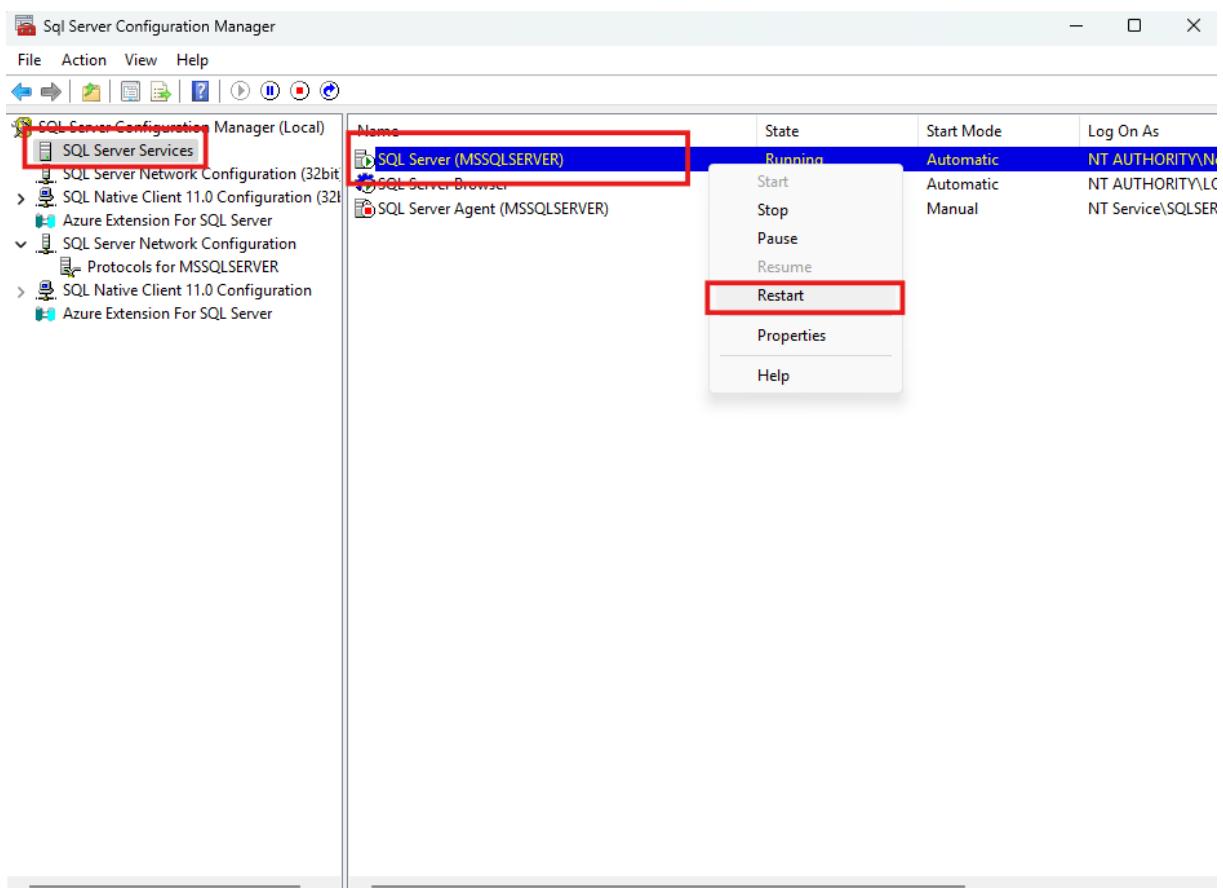


Opens the properties dialog box for the current selection.



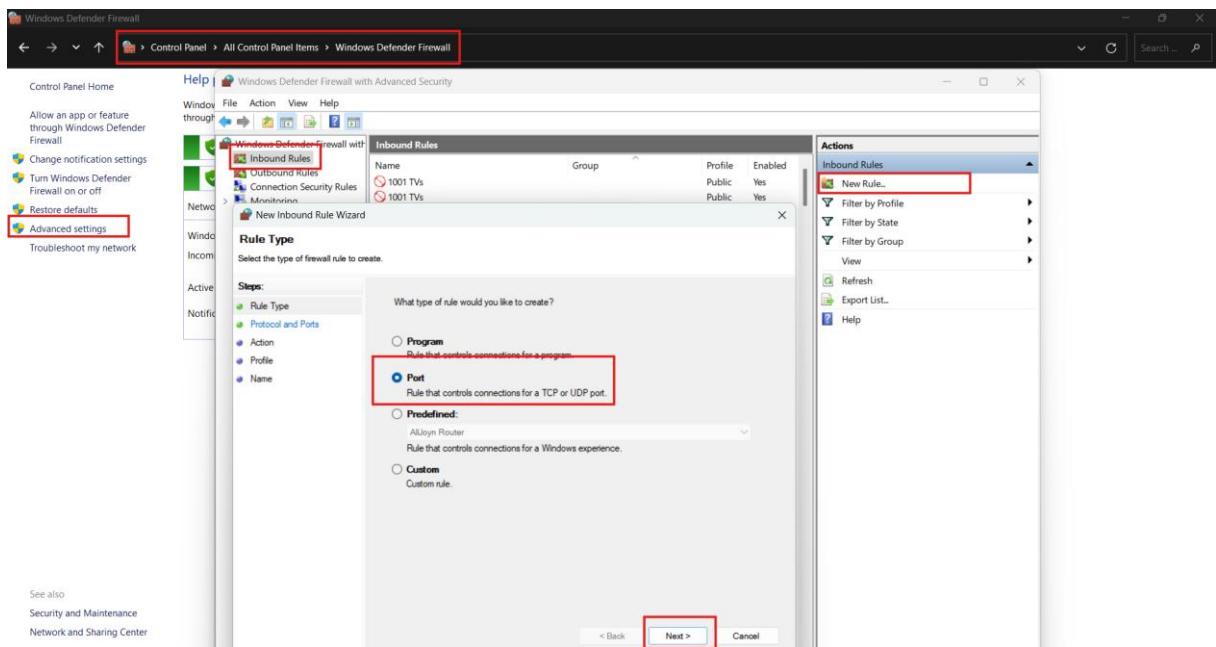


- Quay lại SQL Server Services → SQL Server (MSSQLSERVER) → Restart

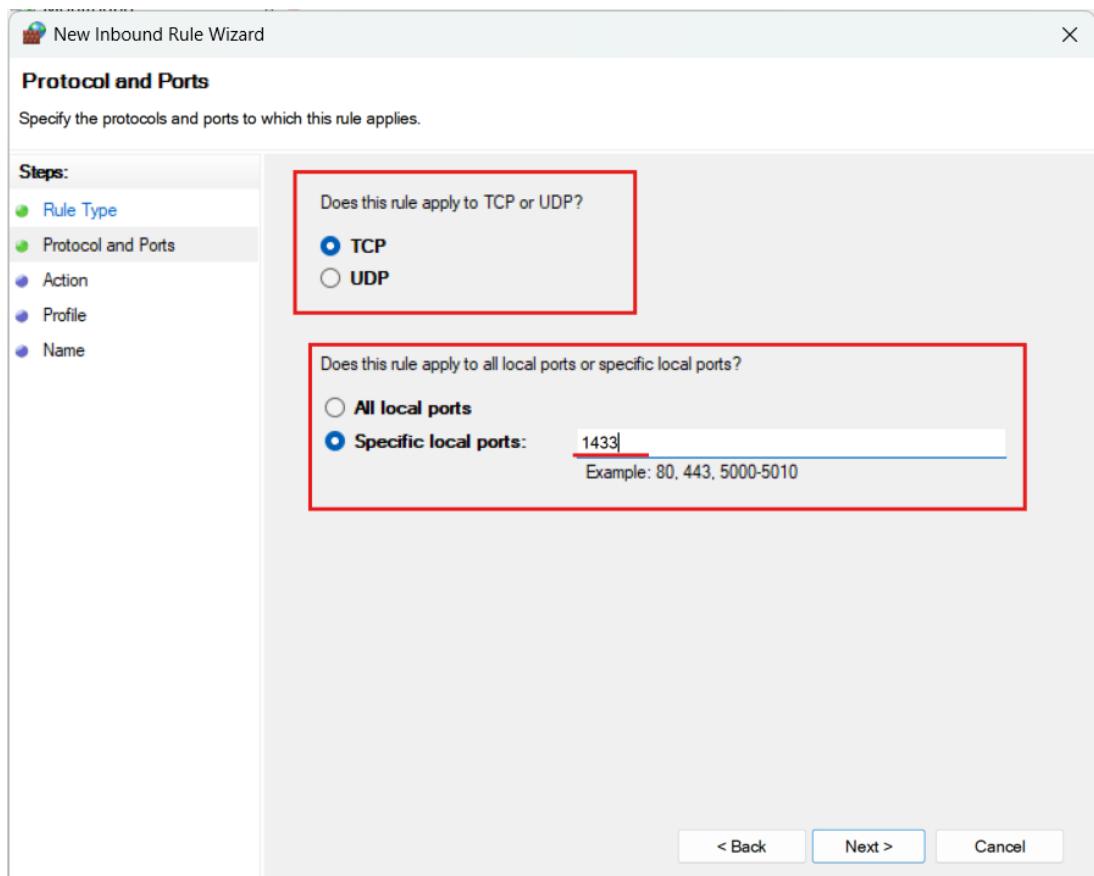


* Lưu ý: Với 1 số trường hợp bị tường lửa chặn thì cài đặt như sau:

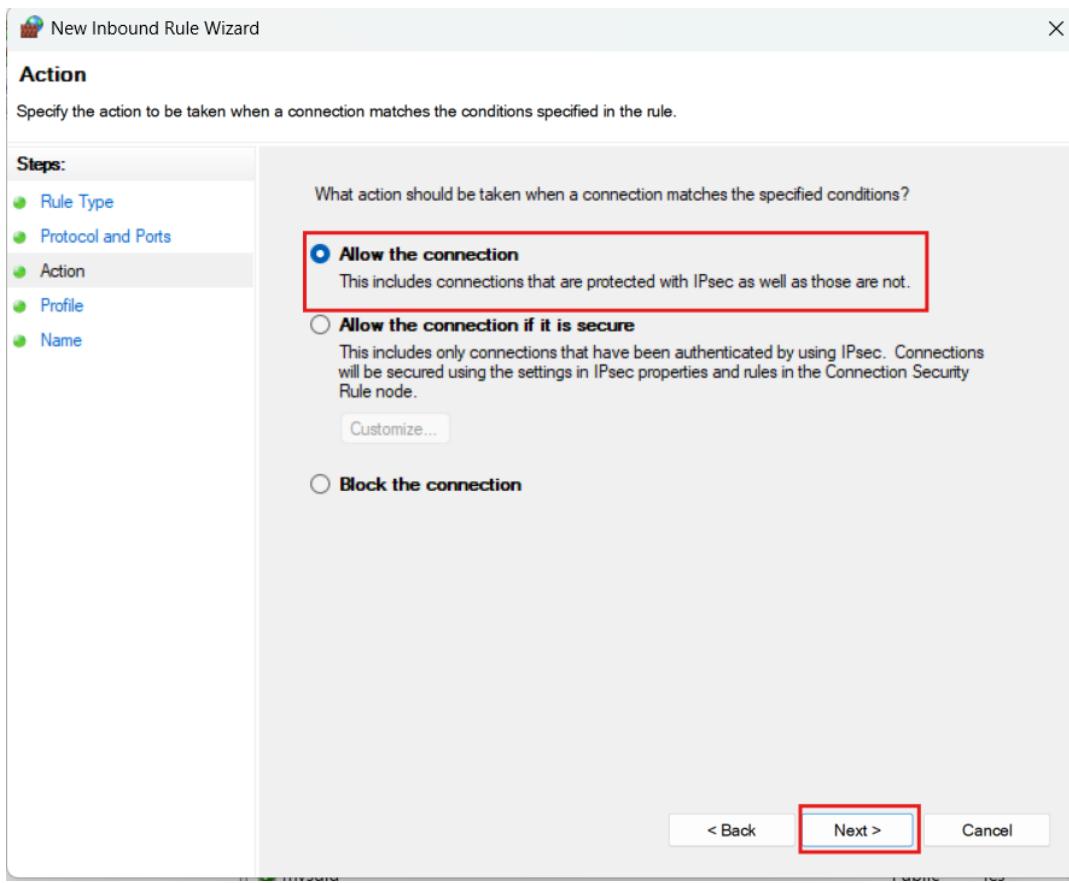
- Control Panel → Windows Firewall → Advanced Settings
- Tiếp đến chọn Inbound Rules → New Rule... → Port → Next



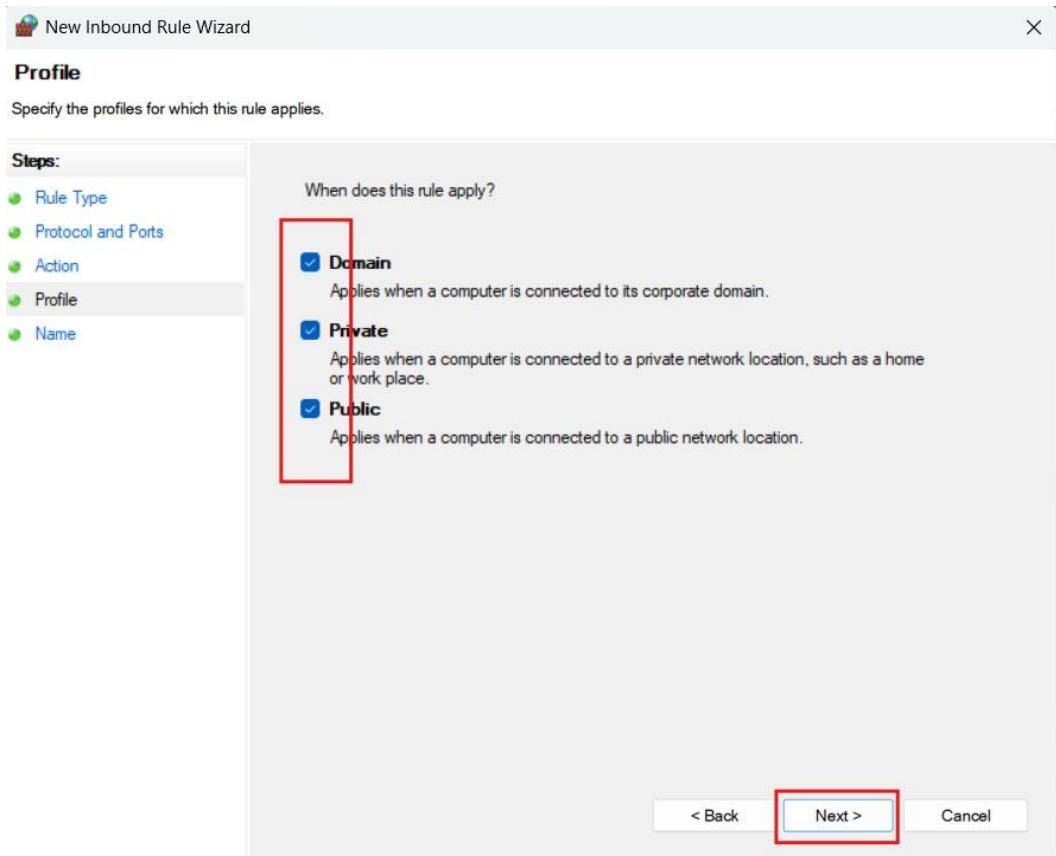
- Chọn TCP → Specific local port: 1433 → Next



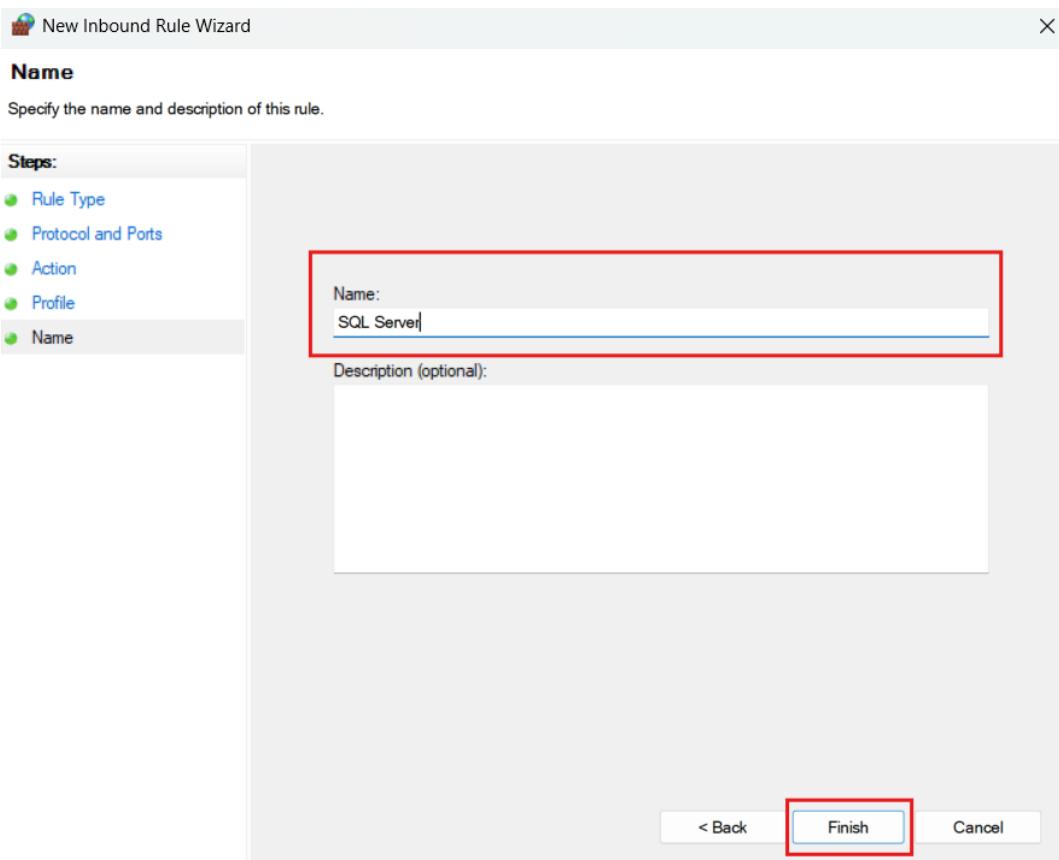
- Chọn Allow the connection → Next



- Tích chọn hết → Next



- Đặt tên → Finish

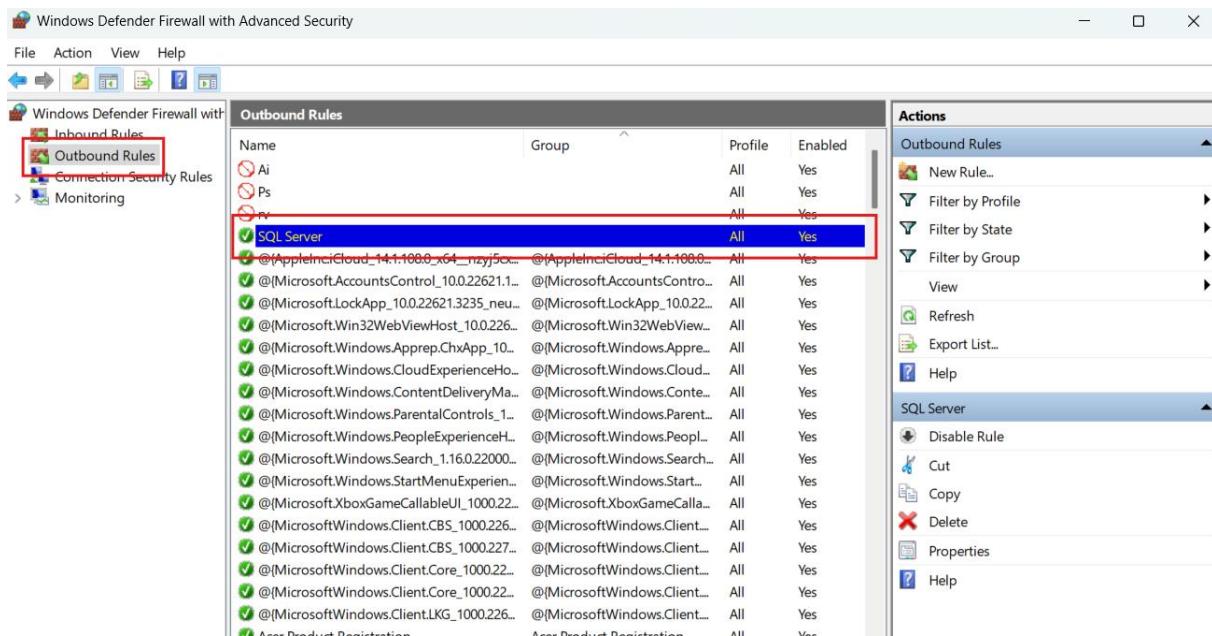


Kết quả có được:

The screenshot shows the Windows Defender Firewall with Advanced Security interface. The left navigation pane includes 'File', 'Action', 'View', 'Help', and categories like 'Inbound Rules', 'Outbound Rules', 'Connection Security Rules', and 'Monitoring'. The main area displays a table of 'Inbound Rules' with columns: Name, Group, Profile, and Enabled. A row for 'SQL Server' is selected and highlighted with a blue background. The right side features an 'Actions' pane with options like 'New Rule...', 'Filter by Profile', 'Filter by State', 'Filter by Group', 'View', 'Refresh', 'Export List...', 'Help', 'Disable Rule', 'Cut', 'Copy', 'Delete', 'Properties', and 'Help'. The 'SQL Server' item is also listed under the Actions pane.

Name	Group	Profile	Enabled
SPSS Basic Script Editor		Private	No
SPSS Basic Script Editor		Private	No
SQL Server		All	Yes
SSJJ_BattleClient_Unity.exe		Private	Yes
SSJJ_BattleClient_Unity.exe		Public	Yes
SSJJ_BattleClient_Unity.exe		Public	Yes
SSJJ_BattleClient_Unity.exe		Private	Yes
Statistics23.com		Private	No
Statistics23.com		Private	No
Statistics23.exe		Private	No
Statistics23.exe		Private	No
Steam		All	Yes
Steam Web Helper		All	Yes
Steam Web Helper		All	Yes
teams.exe		Public	Yes
teams.exe		Public	Yes
teams.exe		Private	Yes
teams.exe		Private	Yes
Teamviewer Remote Control Application		Private	Yes
Teamviewer Remote Control Application		Private	Yes
Teamviewer Remote Control Service		Private	Yes
Teamviewer Remote Control Service		Private	Yes
teraboxhost.exe		Private	Yes
teraboxhost.exe		Public	Yes

- LÀM TƯƠNG TỰ VỚI [Outbound Rules]



Bước 3: Kết nối với Server Database thông qua **Server name**: “<Ipv4 Address>,<port>”

- Lấy địa chỉ IP của máy hosting: Mở CMD → Nhập lệnh ipconfig → Địa chỉ IP là IPv4. (port là 1433 đã setting ở bên trên). Vậy ta có Server name:

192.168.100.174,1433

```
Microsoft Windows [Version 10.0.26100.2314]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

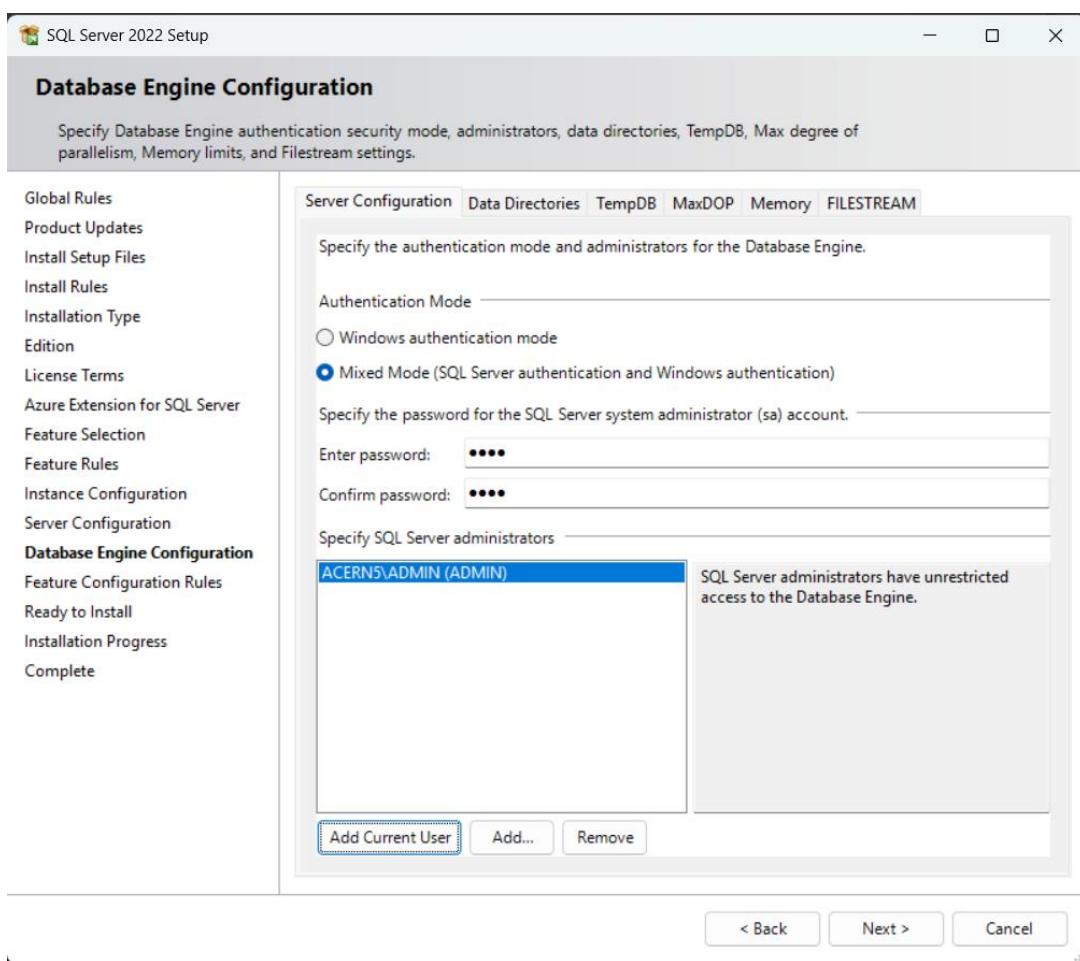
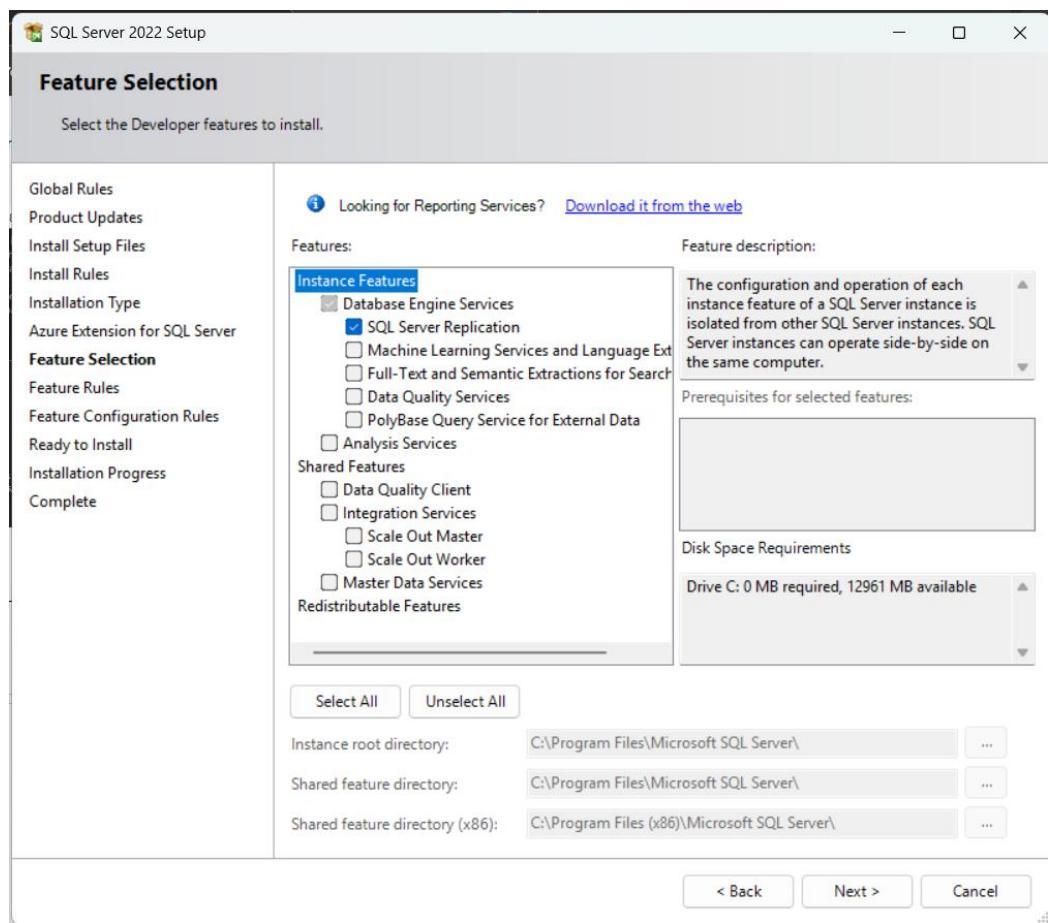
  Connection-specific DNS Suffix . : lan
  IPv6 Address . . . . . : 2405:4802:6084:880f:37ab:3406:89c3:7d87
  IPv6 Address . . . . . : fd00:db80:909b:78f3:c003:5cc0
  Temporary IPv6 Address . . . . . : 2405:4802:6084:880f:6020:bd33:228c:a96b
  IPv6 Address . . . . . : fd00:db80::5
  Temporary IPv6 Address . . . . . : fd00:db80::6020:bd33:228c:a96b
  Link-local IPv6 Address . . . . . : fe80::52:d8a3:c6a:c965%15
  IPv4 Address . . . . . : 192.168.100.174
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::3260:aff:fe60:67ab%15
                           192.168.100.1

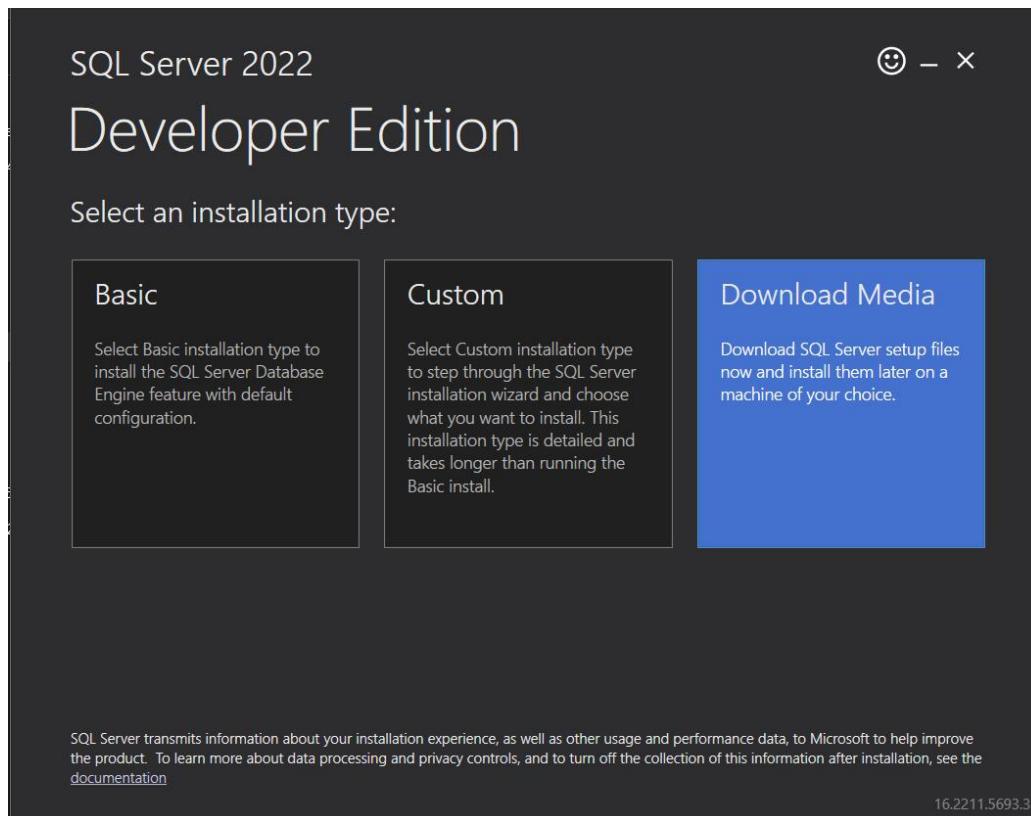
Wireless LAN adapter Wi-Fi:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . : lan
```

VI.3. Cách phân tán dữ liệu cho Server con và liên kết ngược lại:

LƯU Ý: Đảm bảo đã cài feature Replication, Tạo sa account, Download Media, SQL Server Agent đang chạy

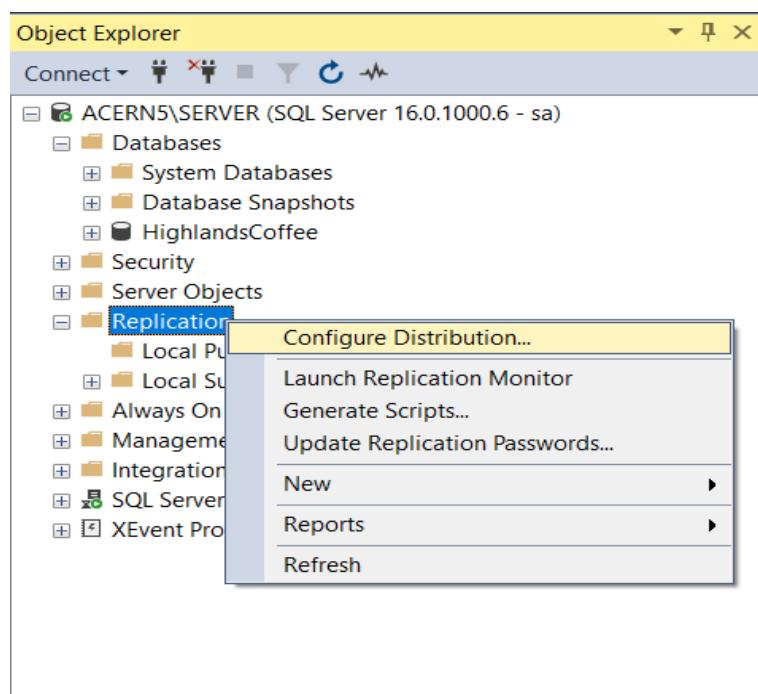




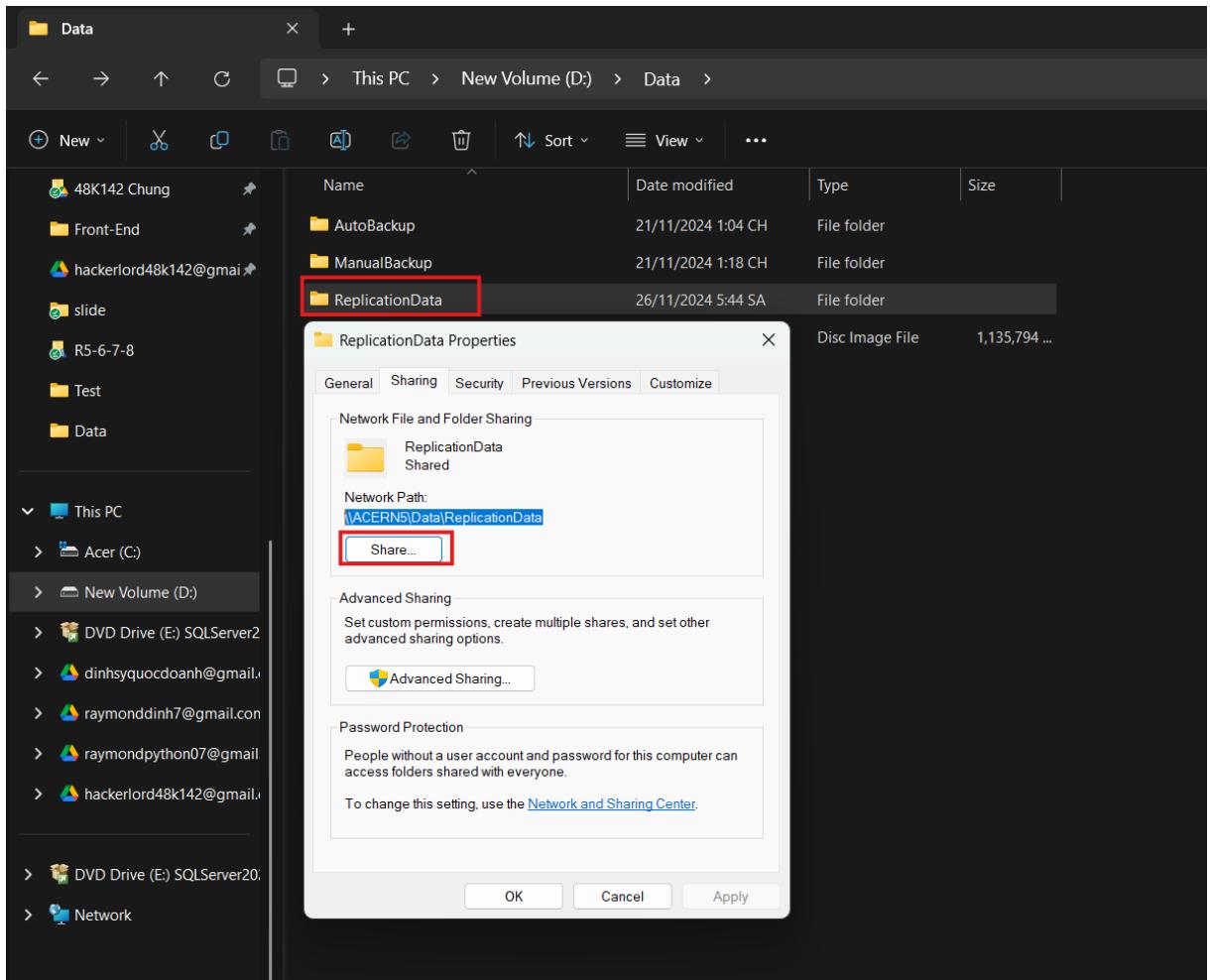
16.2211.5693.3

- + Integration Services Catalogs
- + SQL Server Agent
- + XEvent Profiler

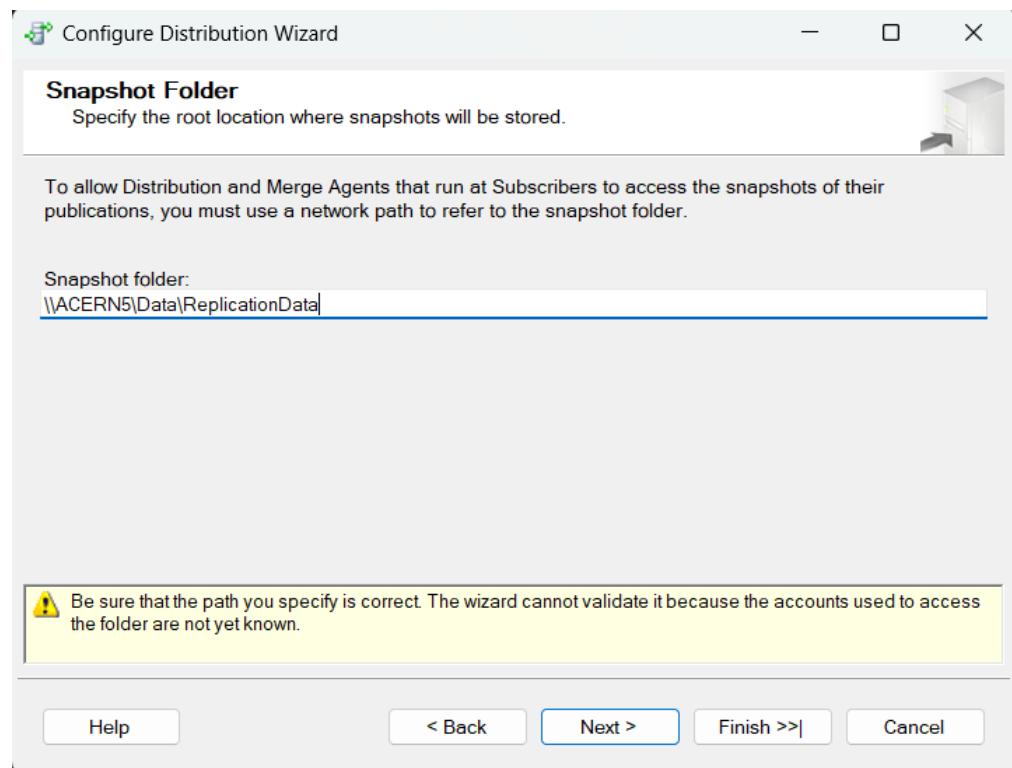
Bước 1: Ở server gốc → Replication → Configure Distribution - Next đến Snapshot Folder



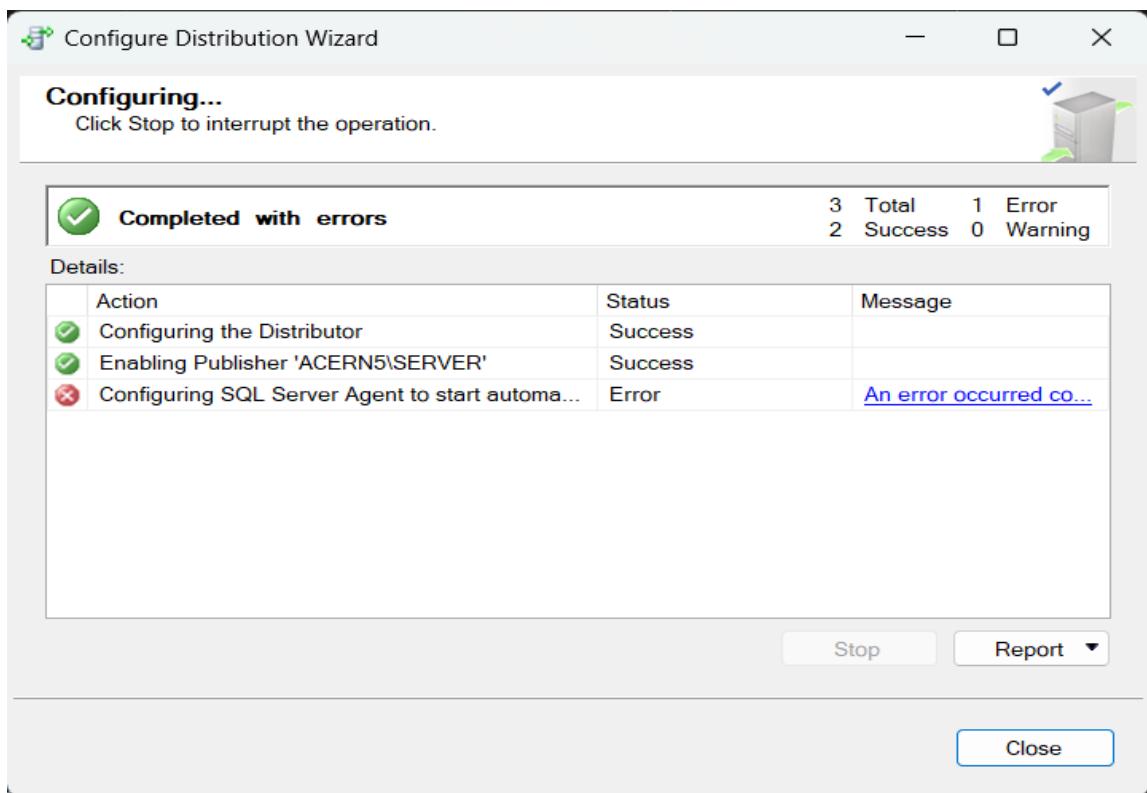
- Tạo Folder ReplicationData và share Everyone với quyền Read/Write → Có đường dẫn **\ACERN5\Data\ReplicationData**



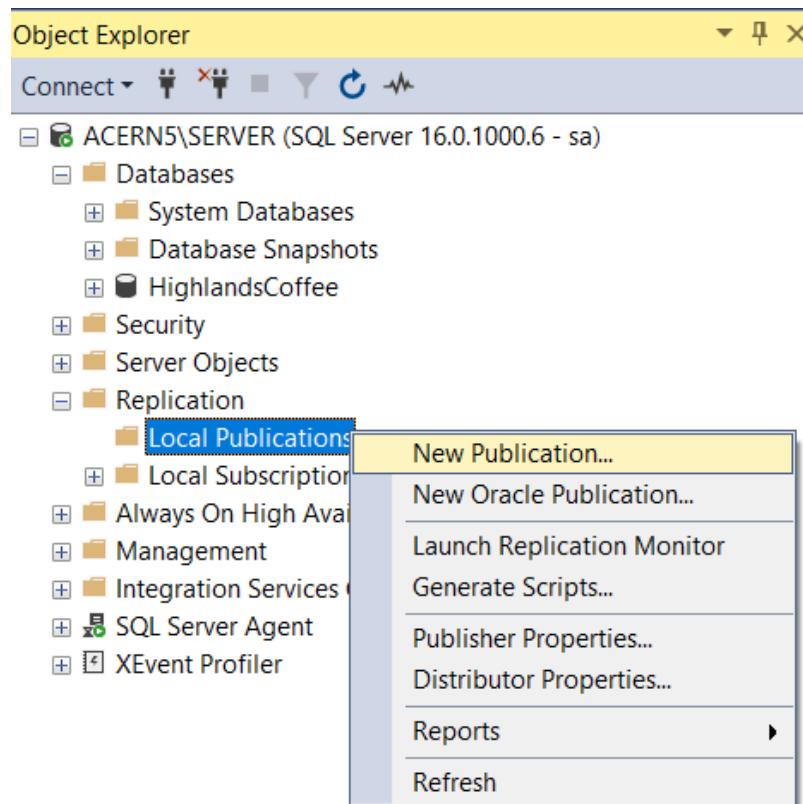
- Thay đường dẫn Snapshot folder → Next đến hết



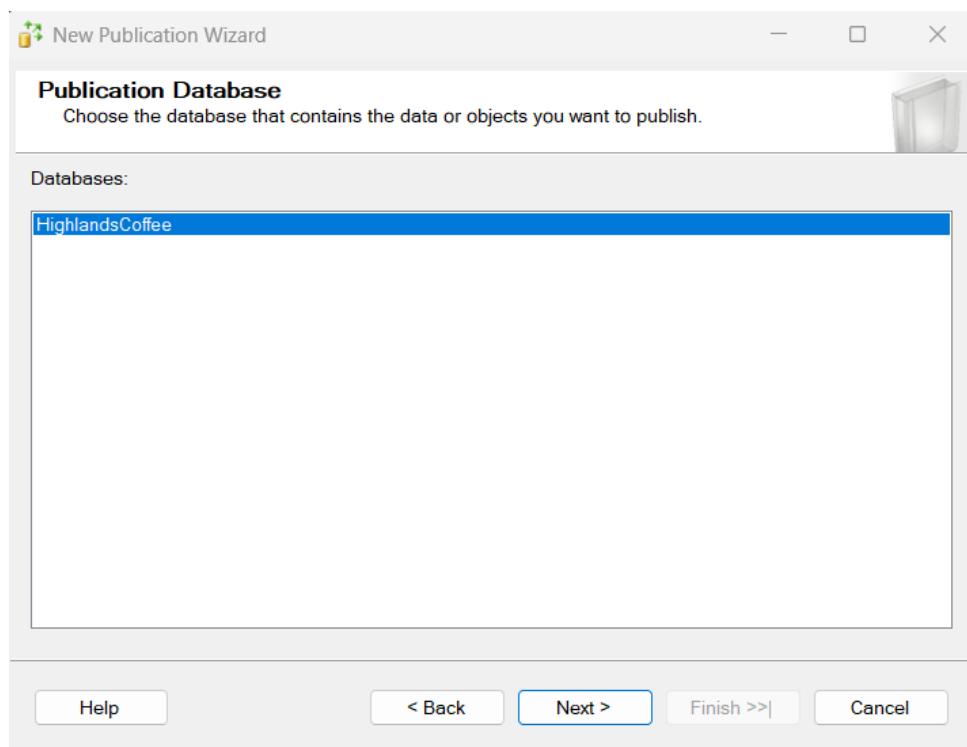
- Kết quả

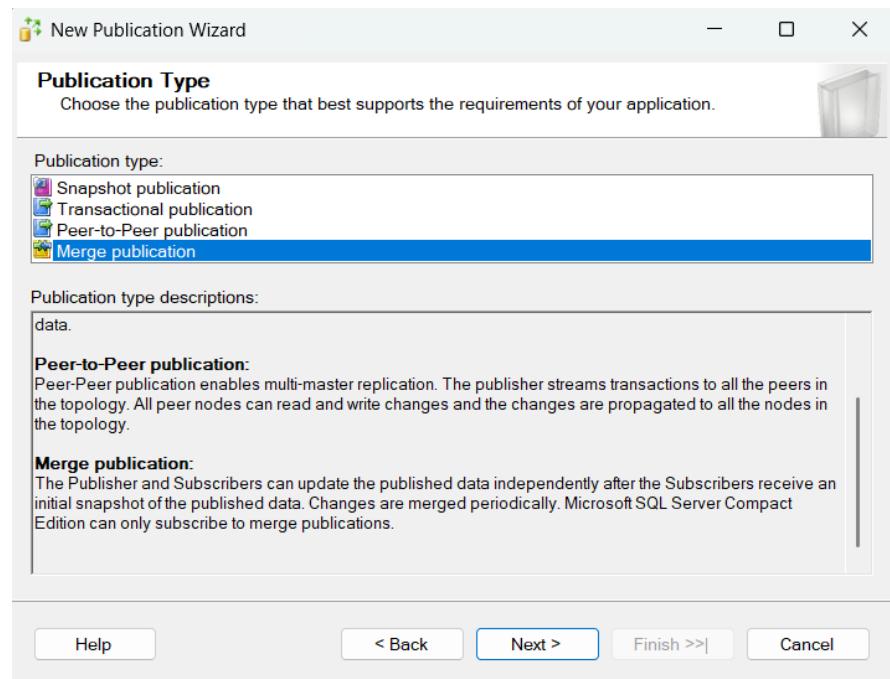


- Local Publications → New Publication... → Next đến khi gặp Publication Database

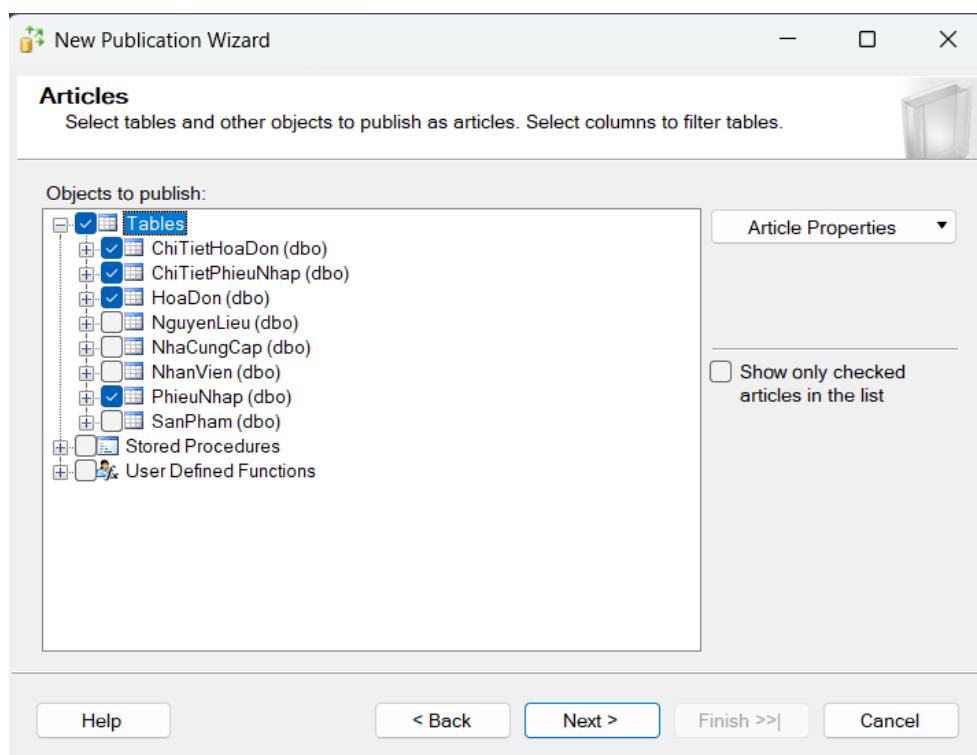


- Chọn database “HighlandsCoffee” → Next → Chọn Merge publication → Next đến Articles

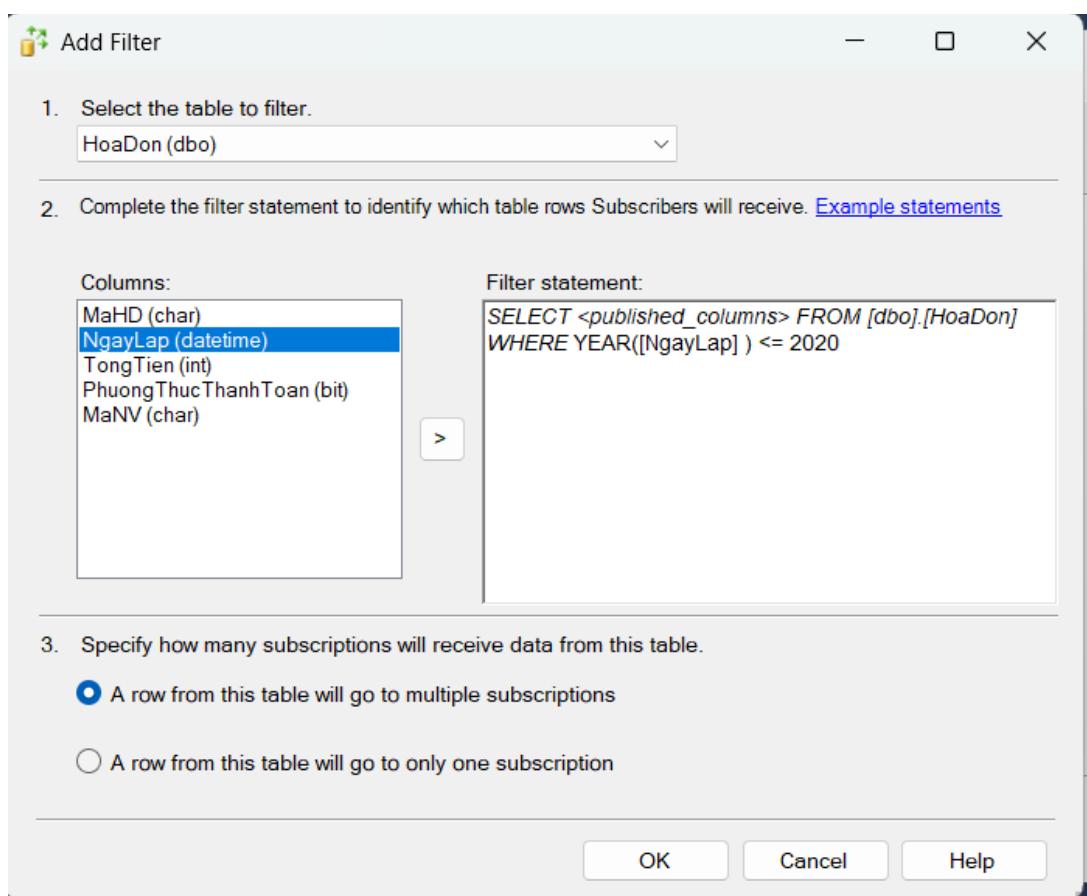
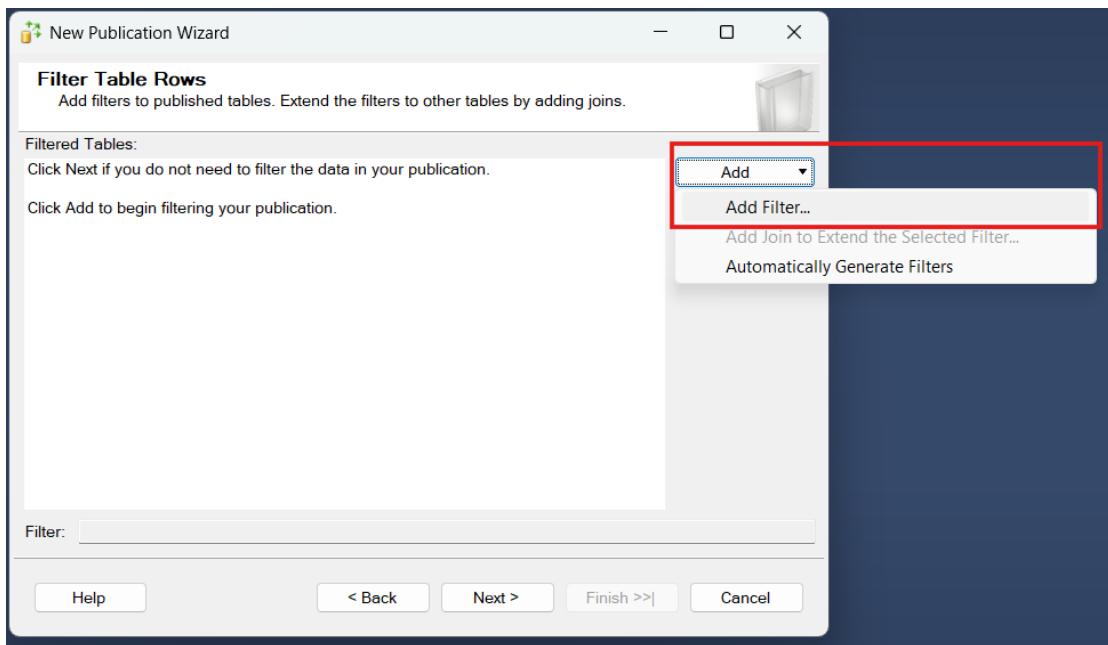


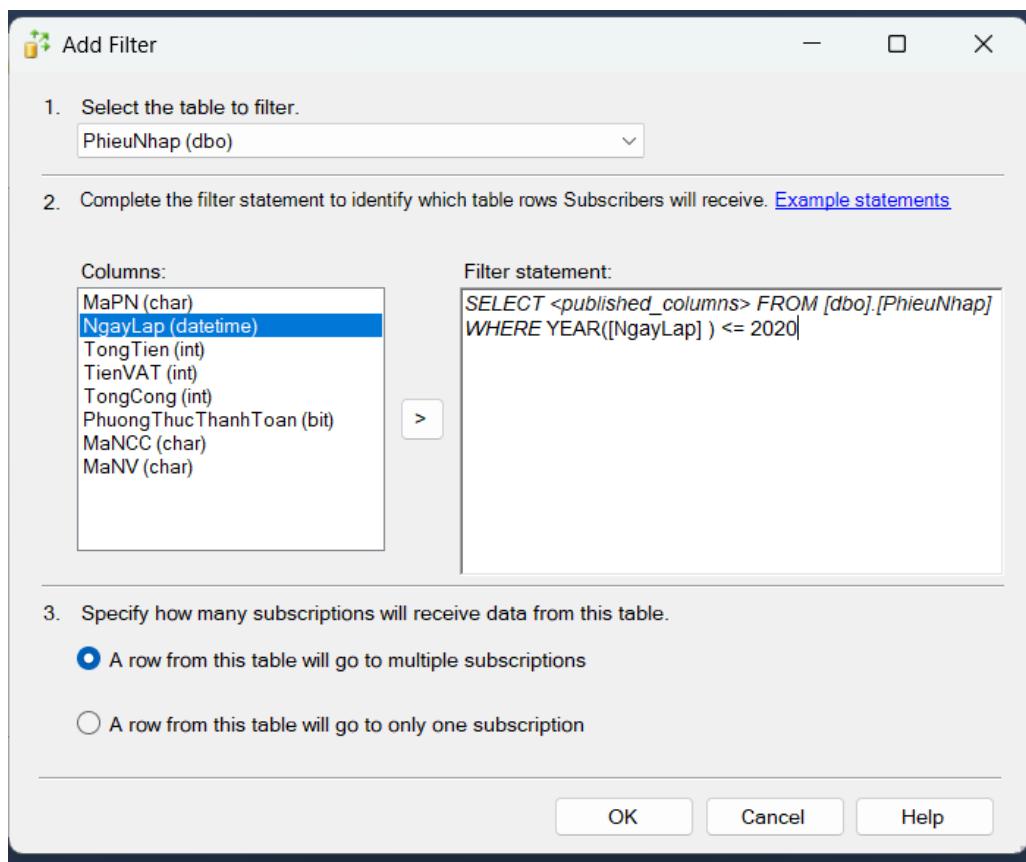


- Chọn các bảng cần phân tán dữ liệu: HoaDon, ChiTietHoaDon, PhieuNhap, ChiTietPhieuNhap → Next đến Filter Table Rows

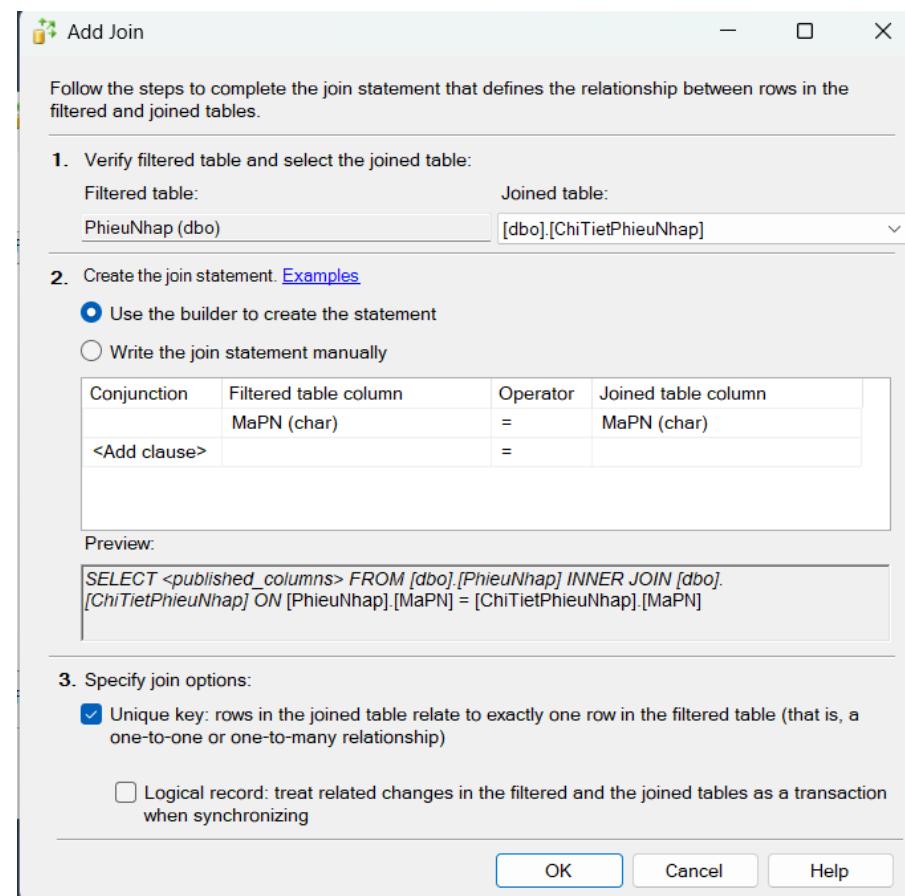
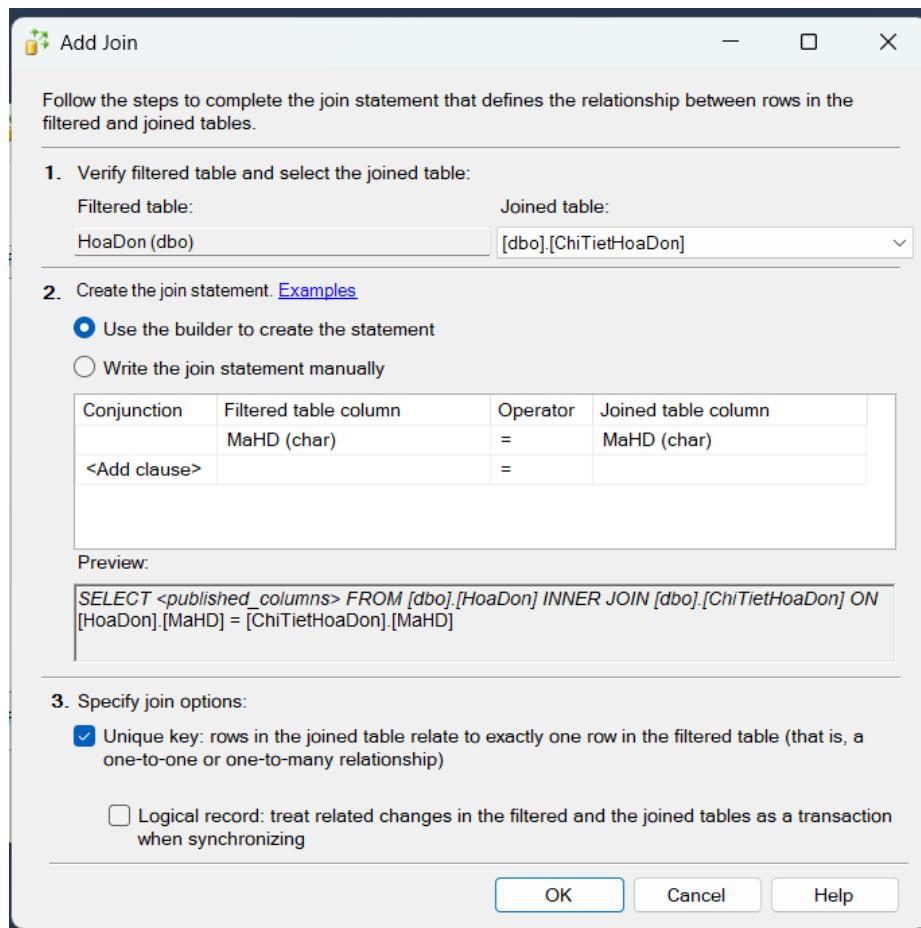


- Chọn Add → Add filter: Ở đây chúng ta sẽ tiến hành tạo điều kiện để phân tán dữ liệu bảng theo NĂM. Ví dụ: Các dữ liệu thuộc trước 2020 của 2 bảng HoaDon và PhieuNhap.

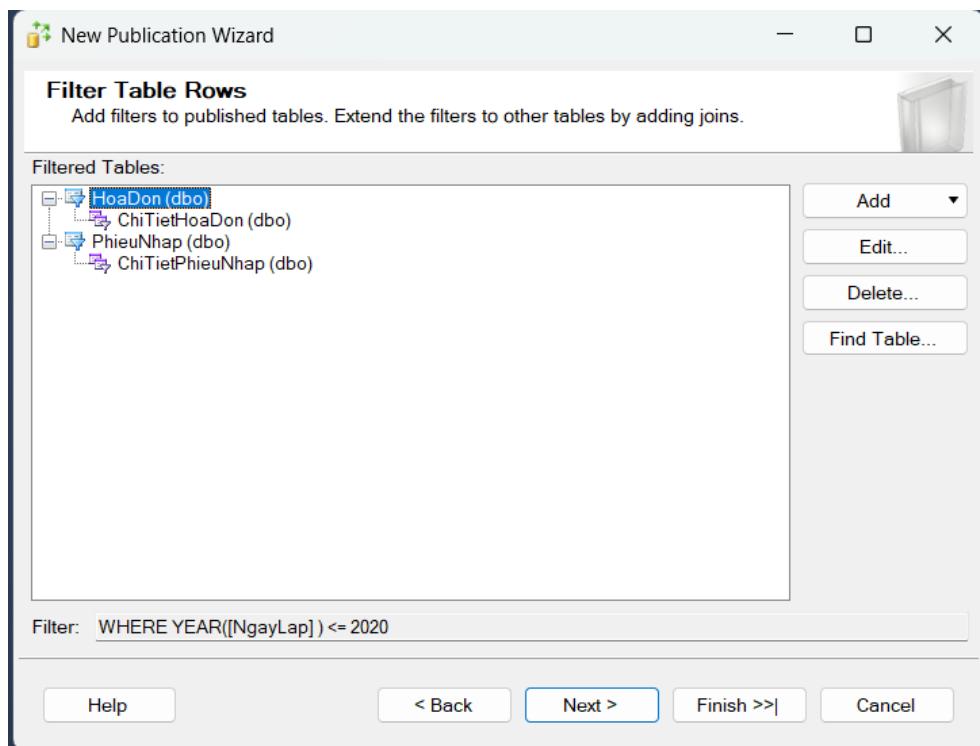




- Chọn bảng HoaDon/PhieuNhap → Add → Add join to Extend the Selected Filter thông qua điều kiện MaHD/MaPN để liên kết các dữ liệu liên quan sẽ được phân tán sang Server 2



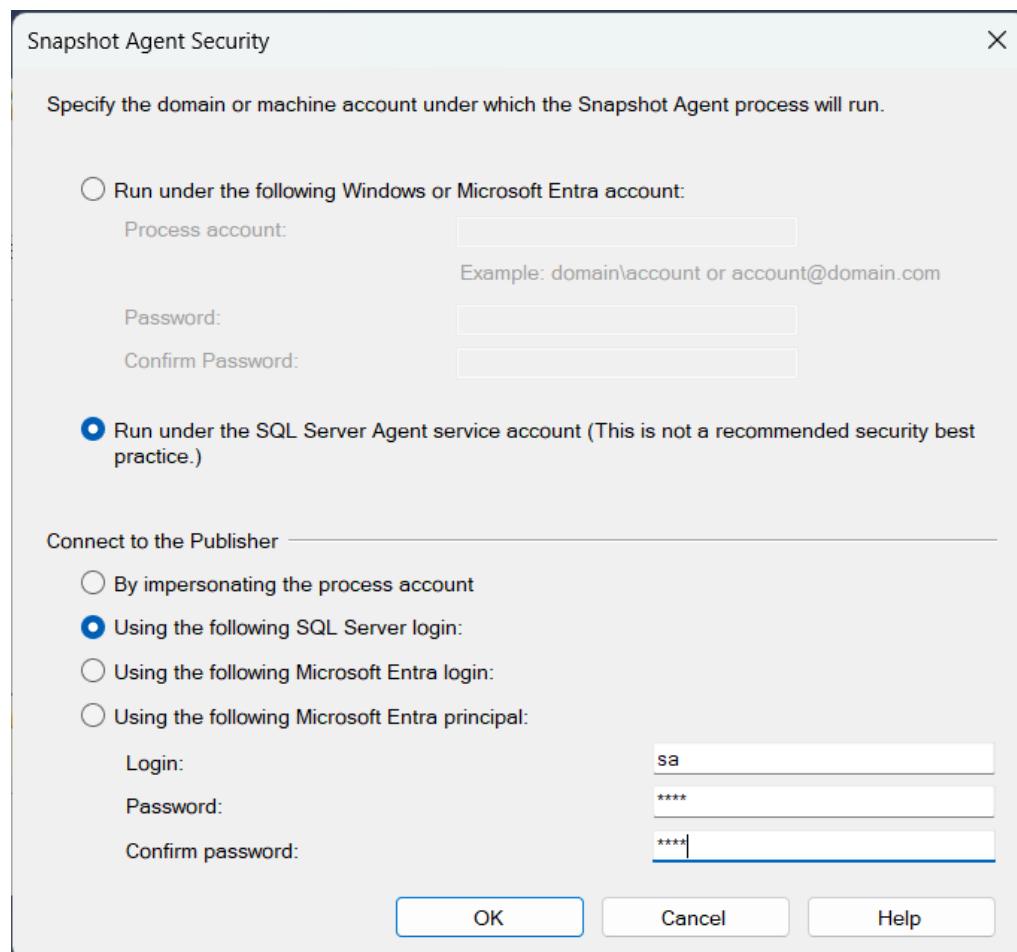
- Kết quả có được: Next tới Agent Security



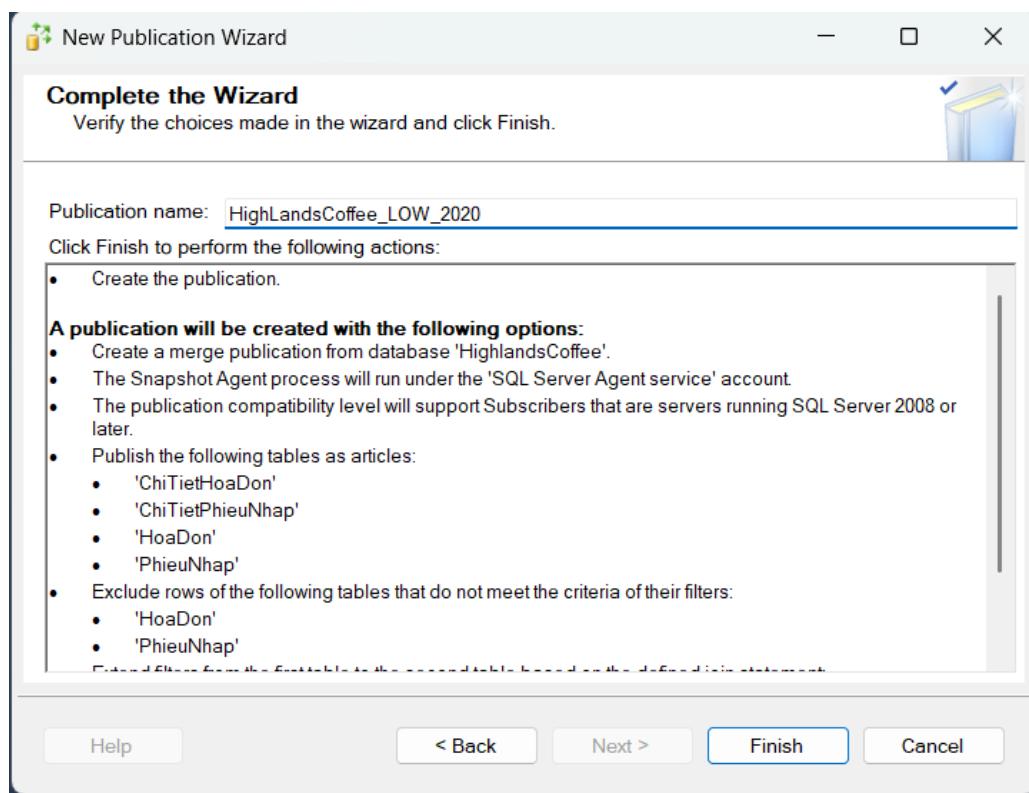
- Chọn Sercurity Setting, cài đặt như trong hình → OK → Next đến Complete the Wizard

Login: sa

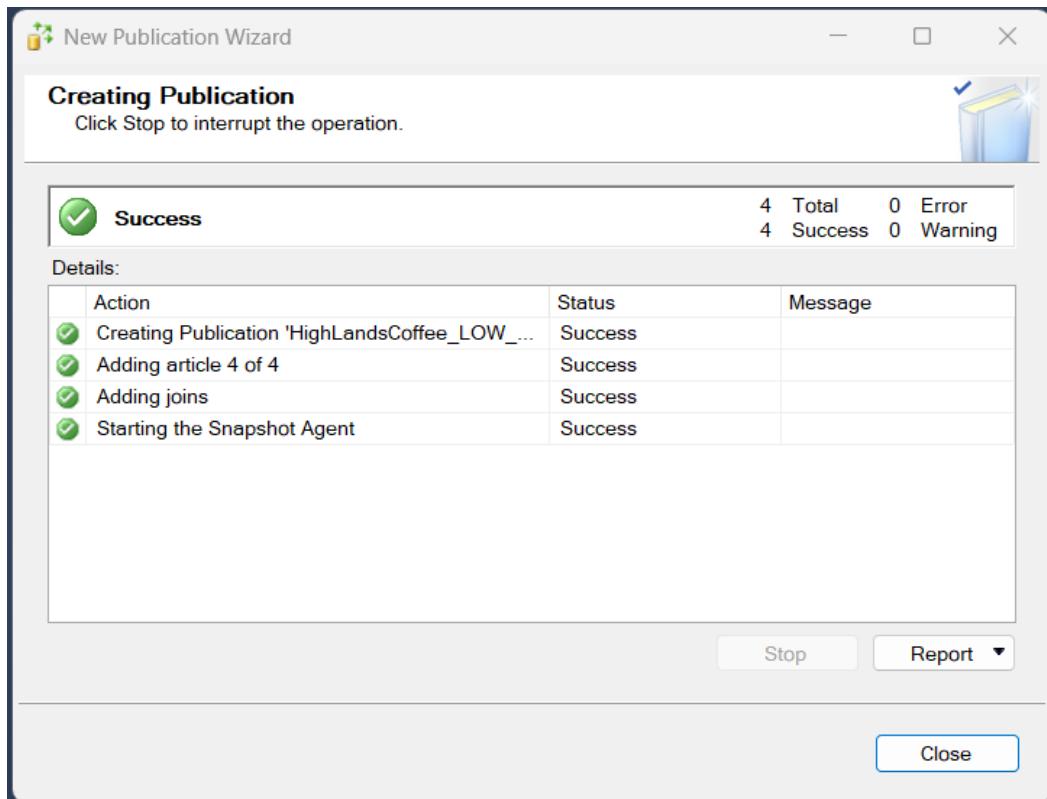
Password: <được cài khi tạo server>



- Đặt tên cho Publication name → Finish

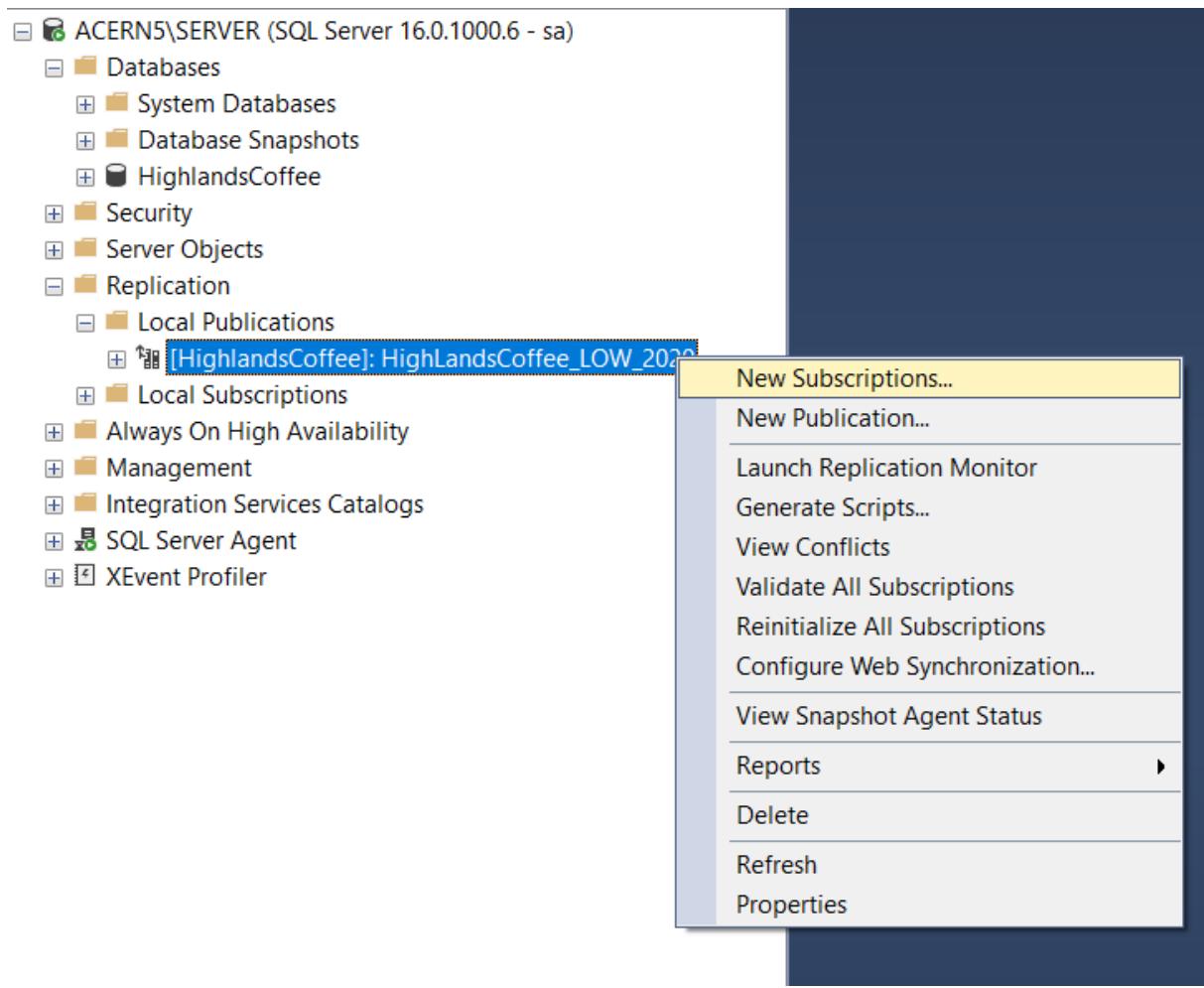


- Kết quả

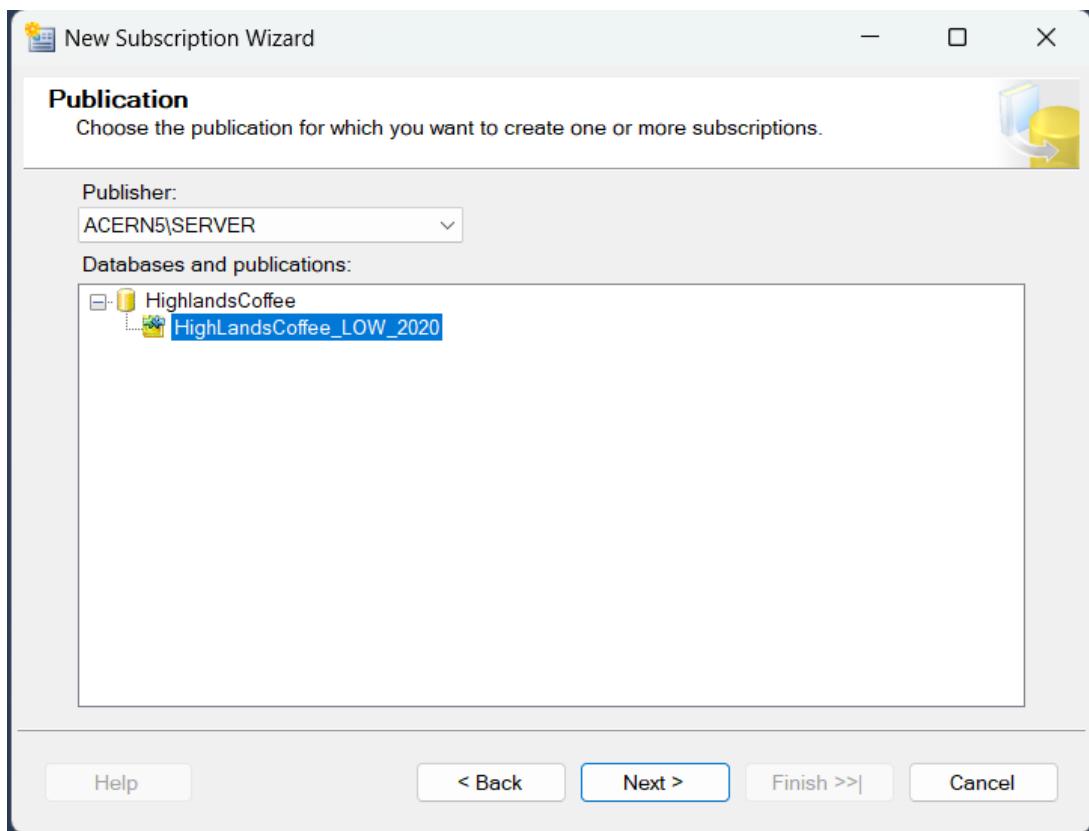


Bước 2: Chuyển dữ liệu sang Server 2

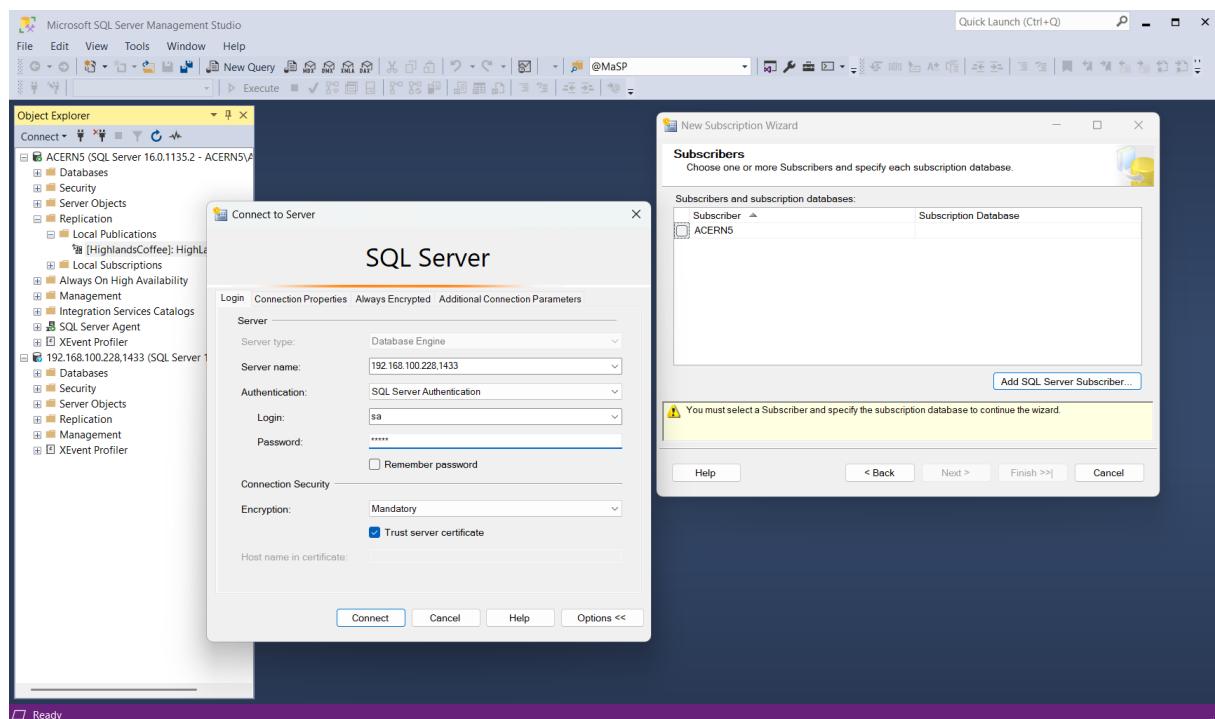
- Tạo New Subscriptions... → Next



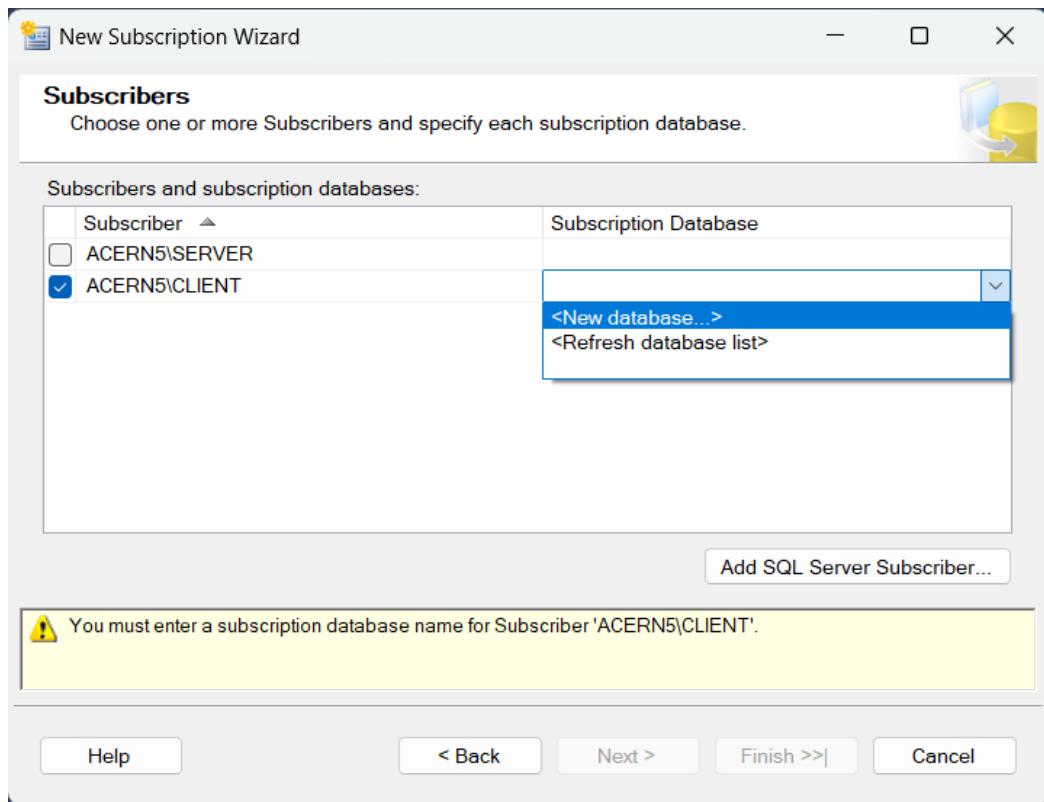
- Chọn đúng Publication → Next tới Subscribers



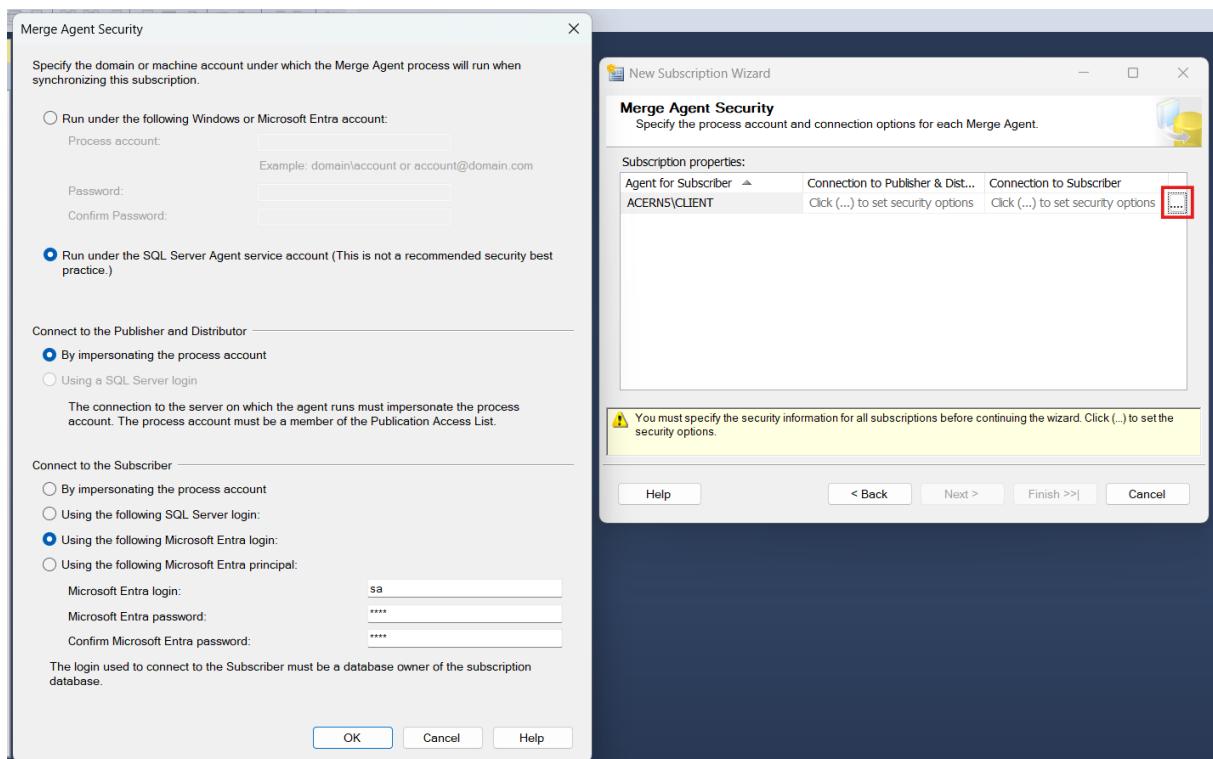
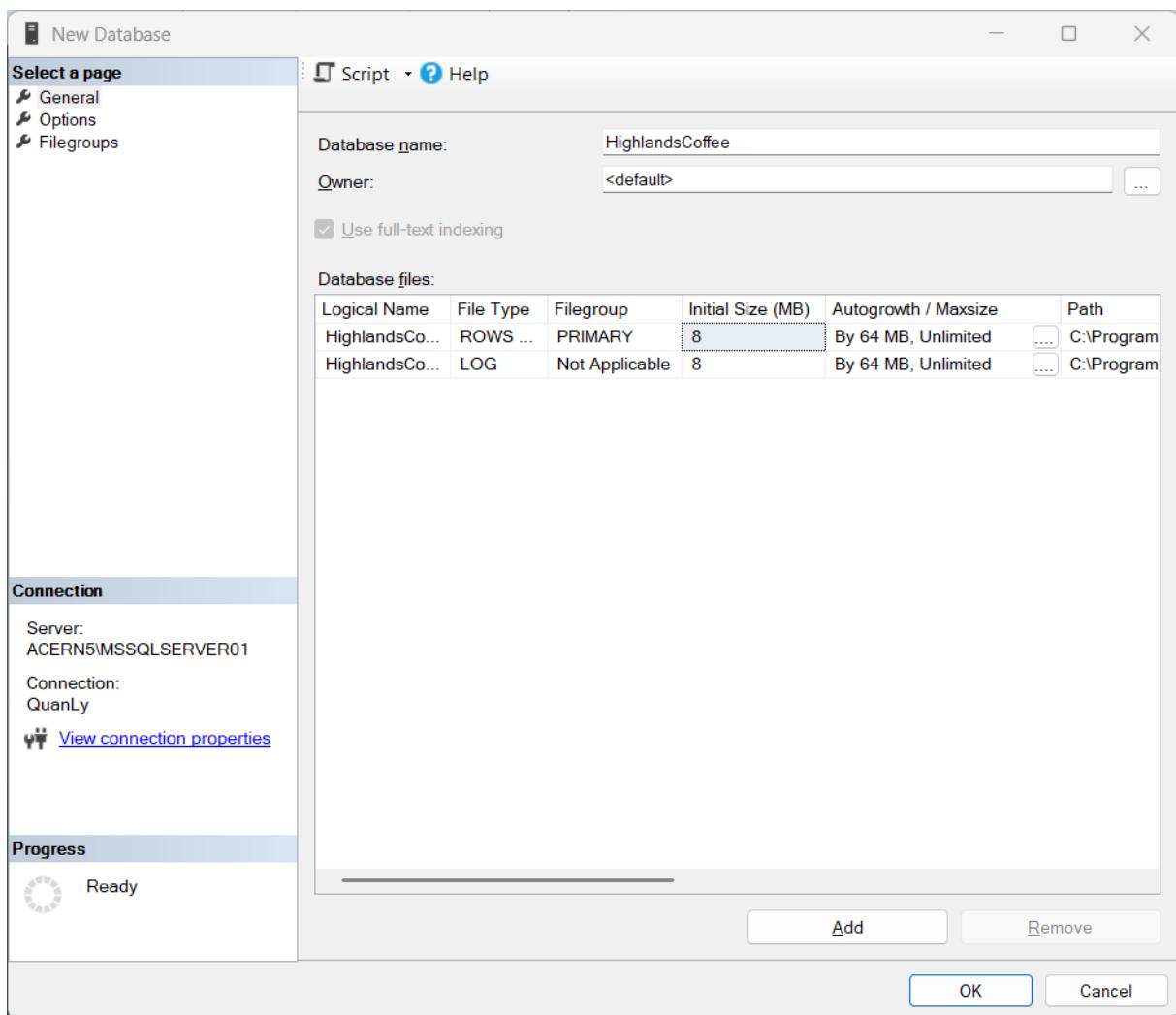
- Muốn liên kết đến Server nào thì chọn Add SQL Server Subscribers... → Thực hiện connect với Server 2 qua server name: <IPv4 Address>,1433



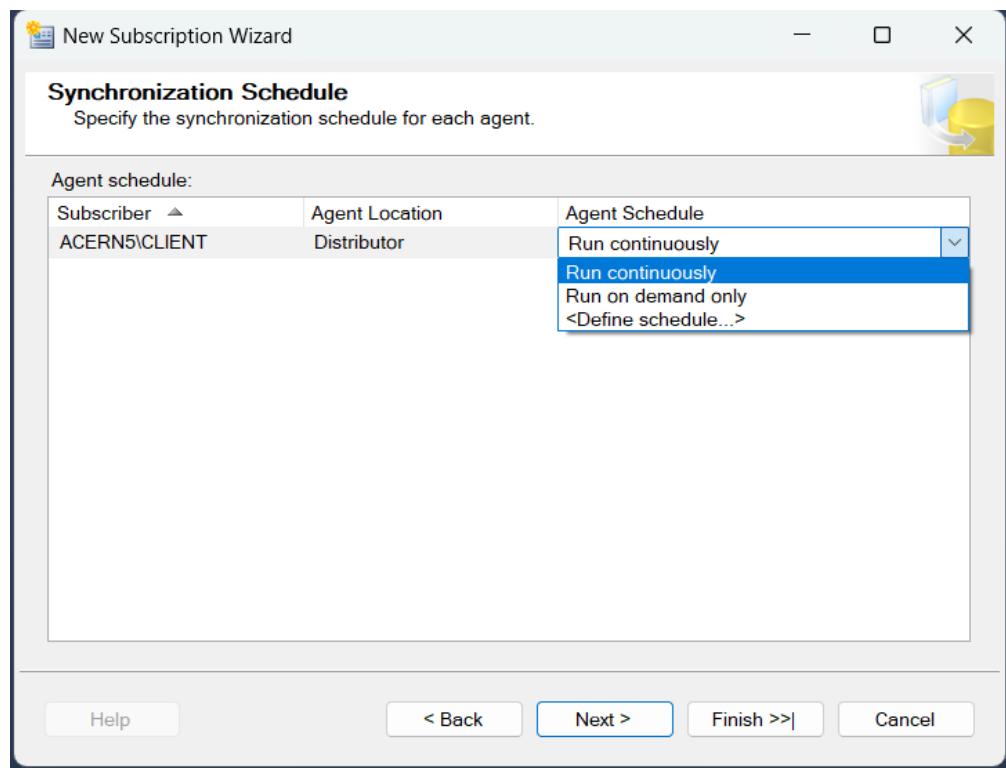
- Chọn new database



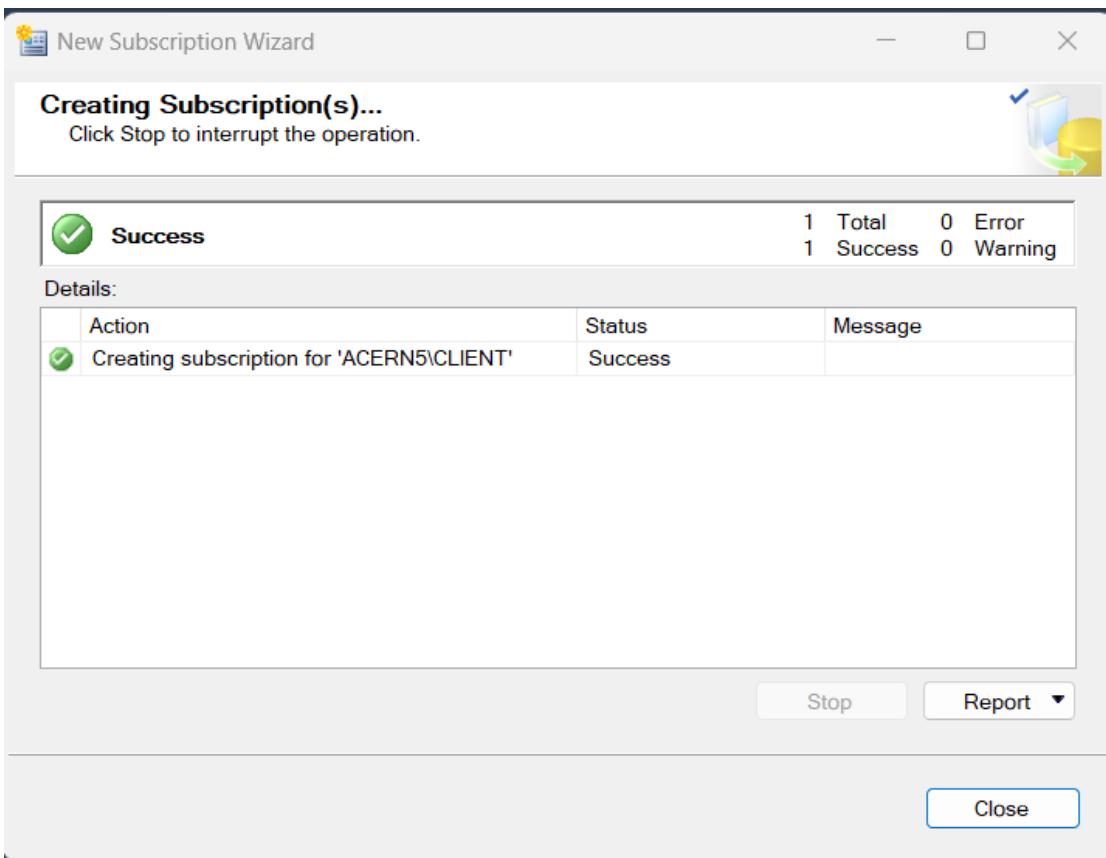
- Đặt tên Database ở server 2 → OK → Next → Cài đặt Merge Agent Security → Next

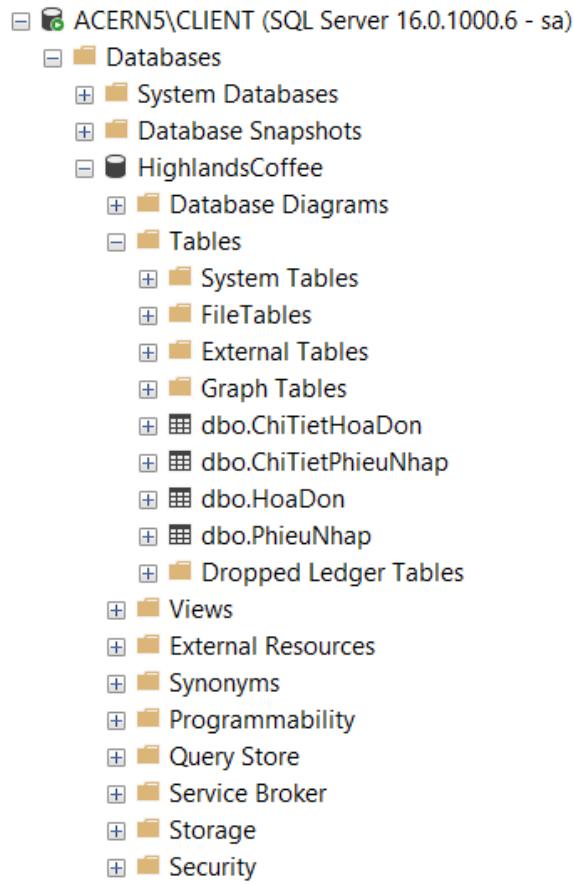


- Đưa Agent Schedule thành Run continuously → Next đέn hét



- Kết quả:



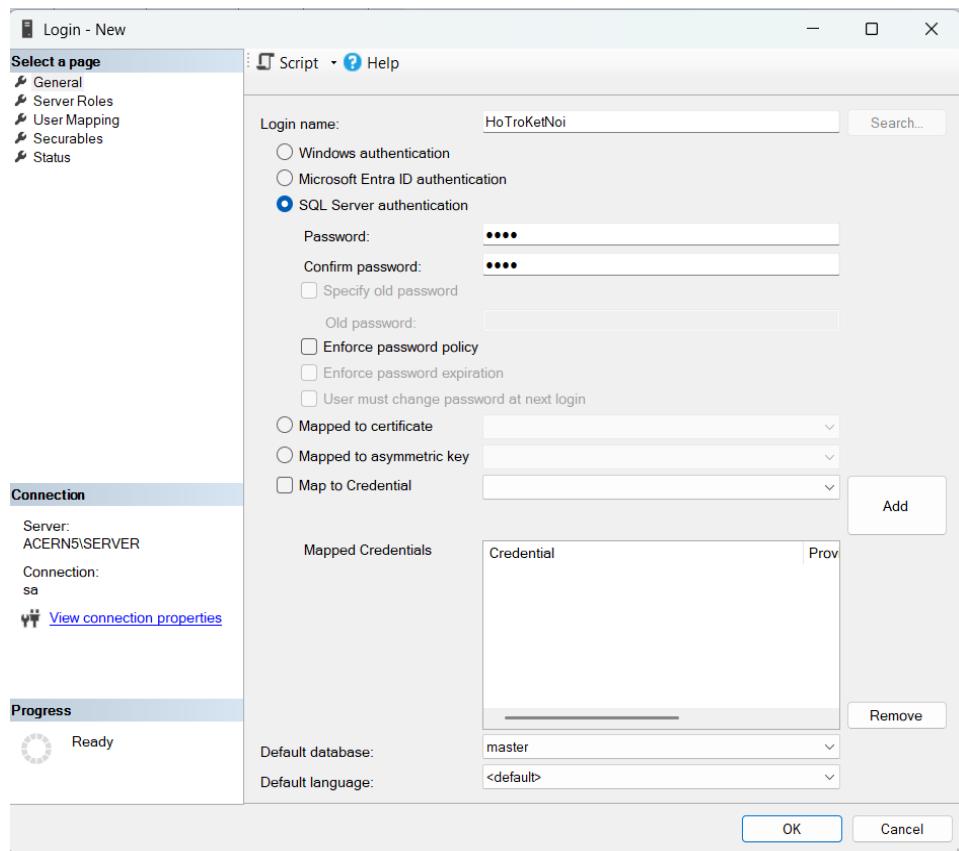


Bước 3: Đóng bộ dữ liệu từ máy phân tán về máy Server

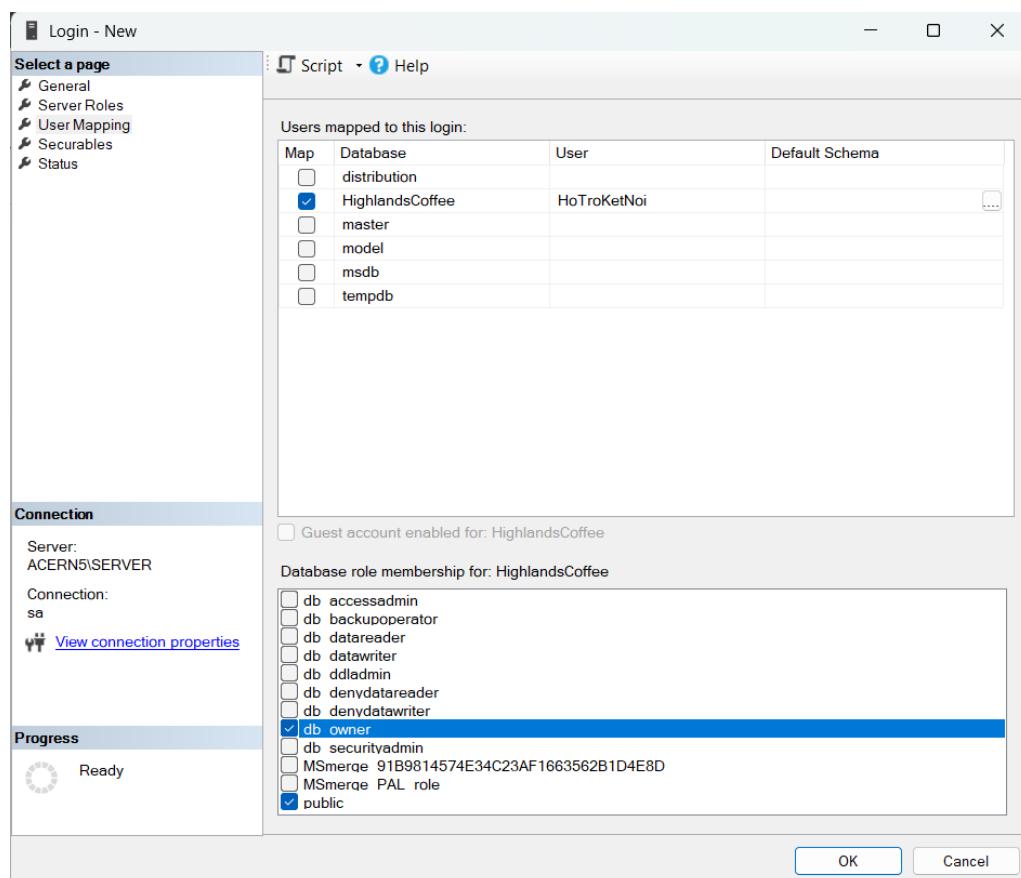
- Tạo Login ở từng sever

Tài khoản: HoTroKetNoi

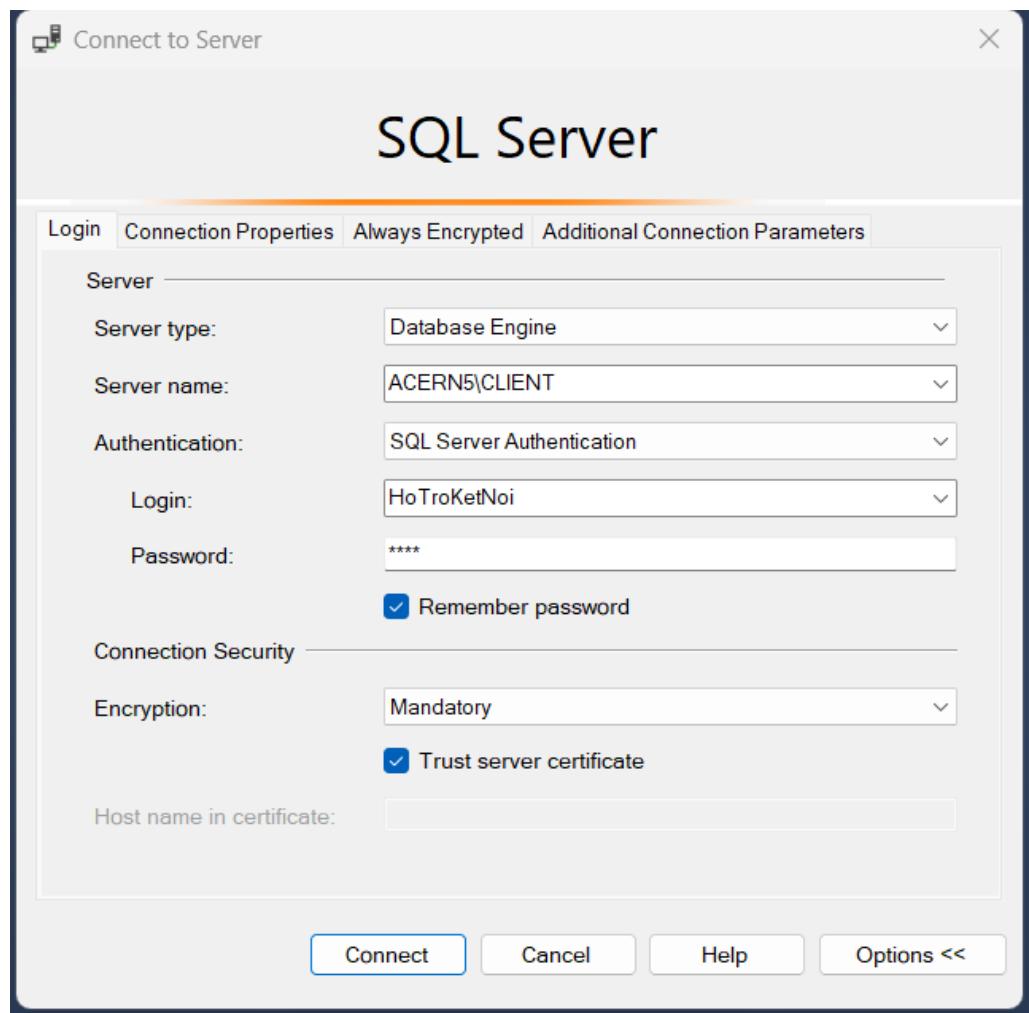
Mật khẩu: 1111



- Tạo user mapping → chọn database → db owner → OK

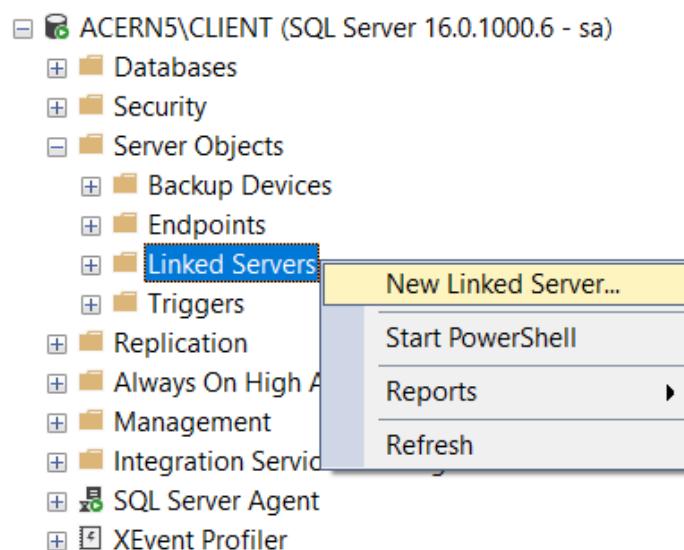


- Tương tự làm cho CLIENT và kết nối lại với login HoTroKetNoi

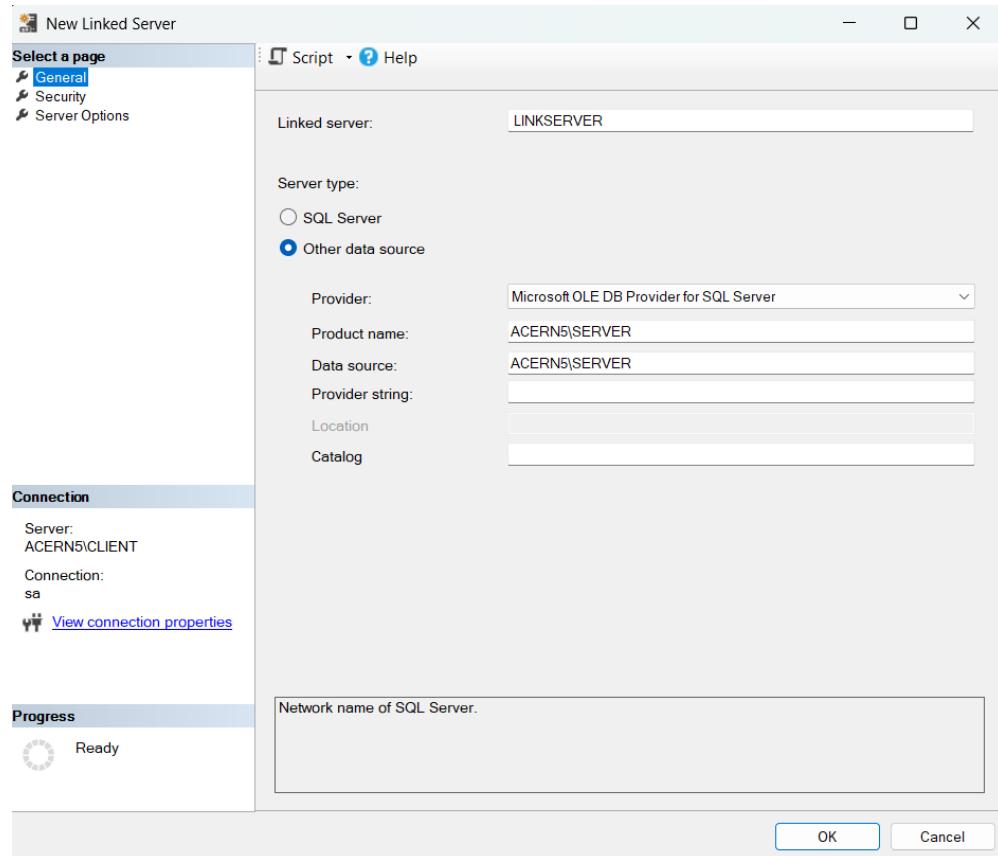


Bước 4: Tạo Linked Server

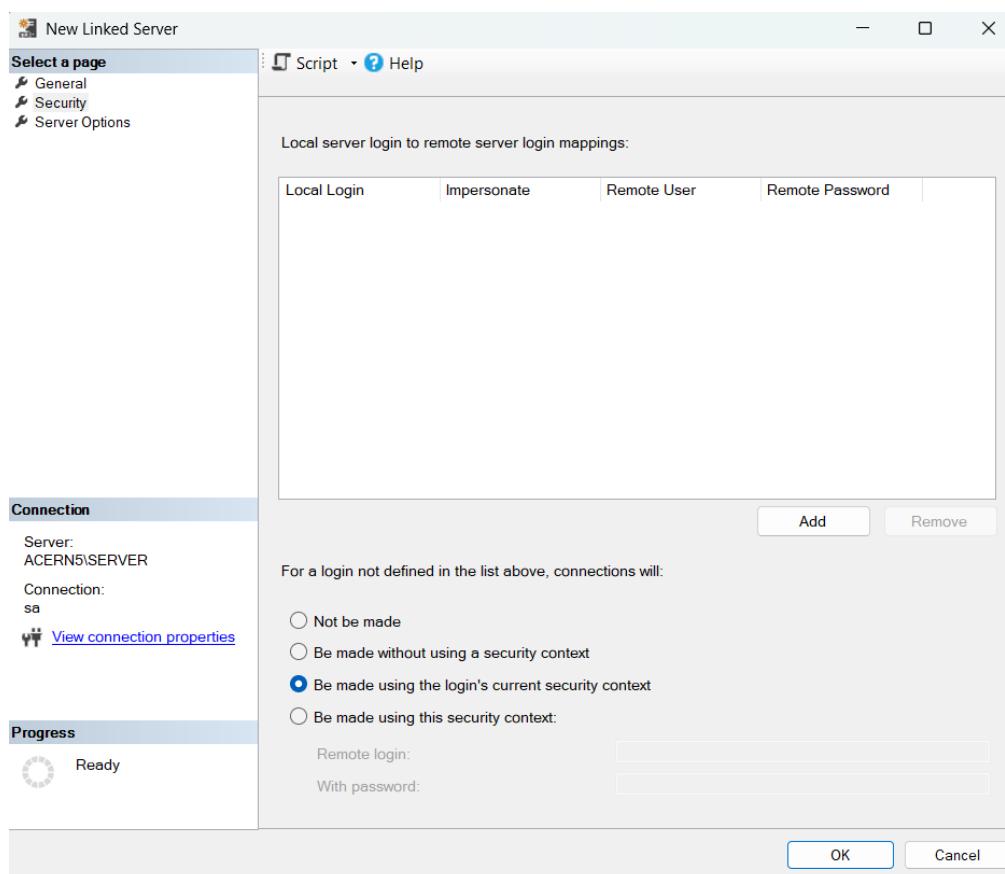
- Tạo New Linked Server bằng account sa ở phía SERVER/CLIENT như hình bên dưới



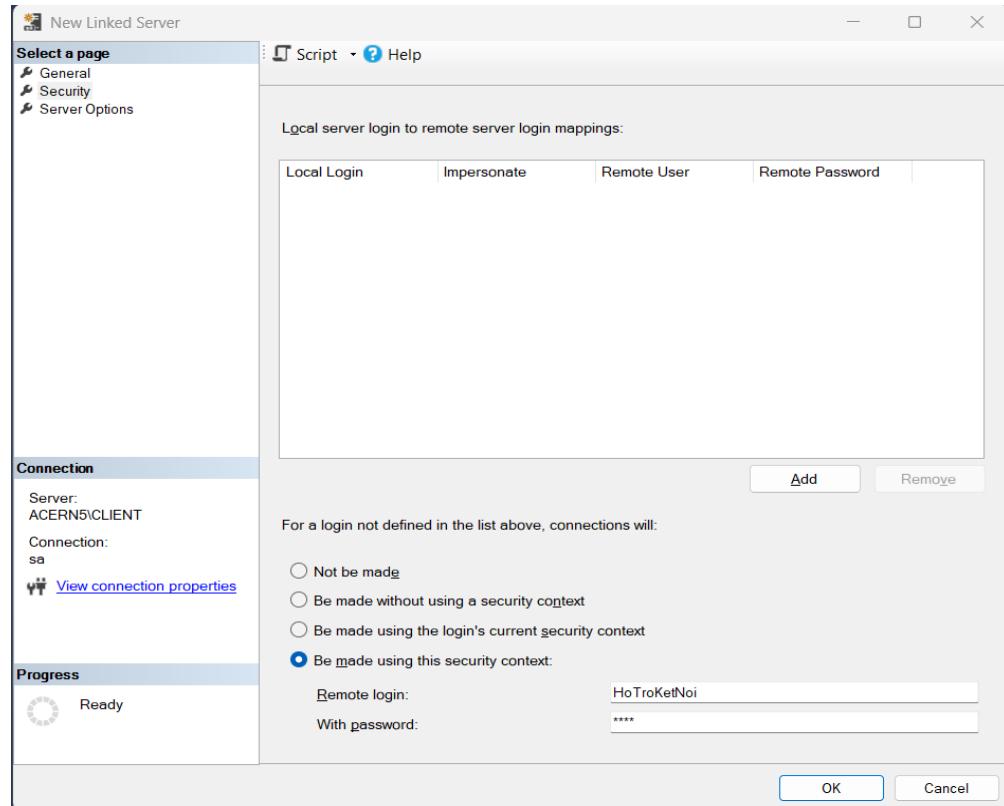
- Điện tên và SERVER gốc trên cả SERVER/CLIENT



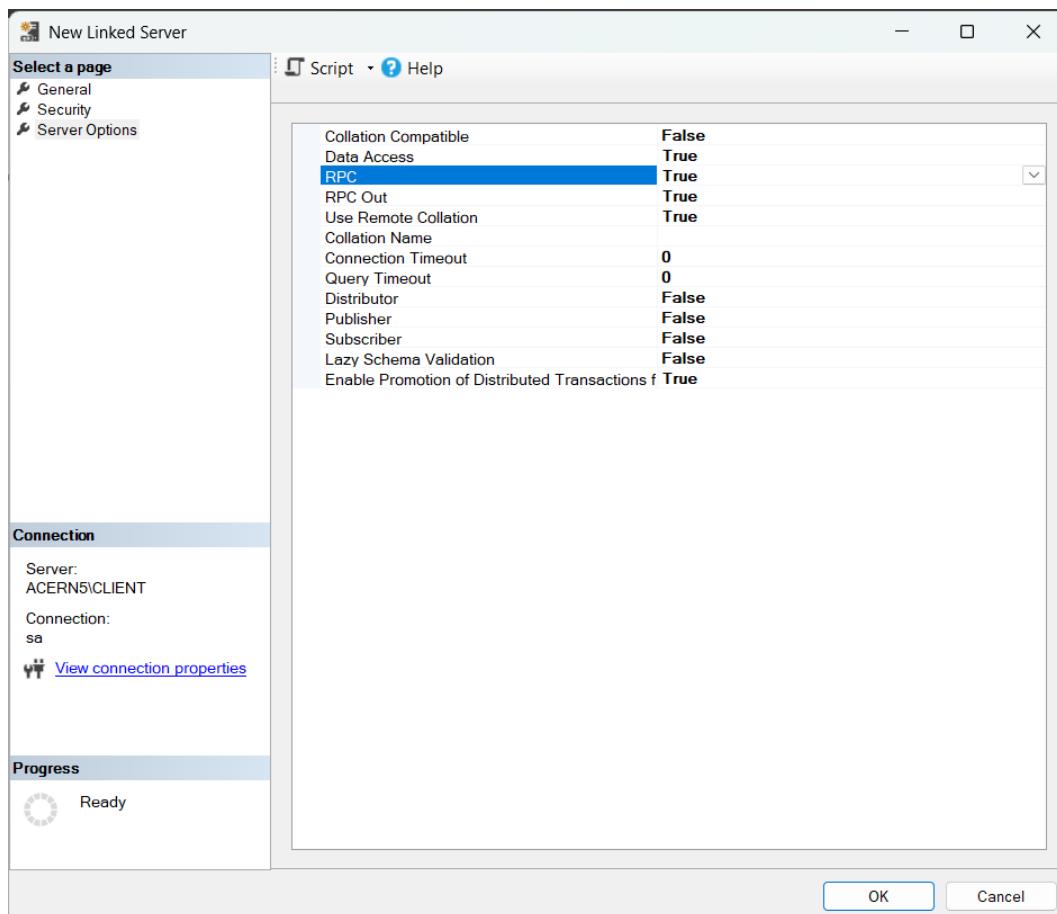
- Cài đặt security cho SERVER



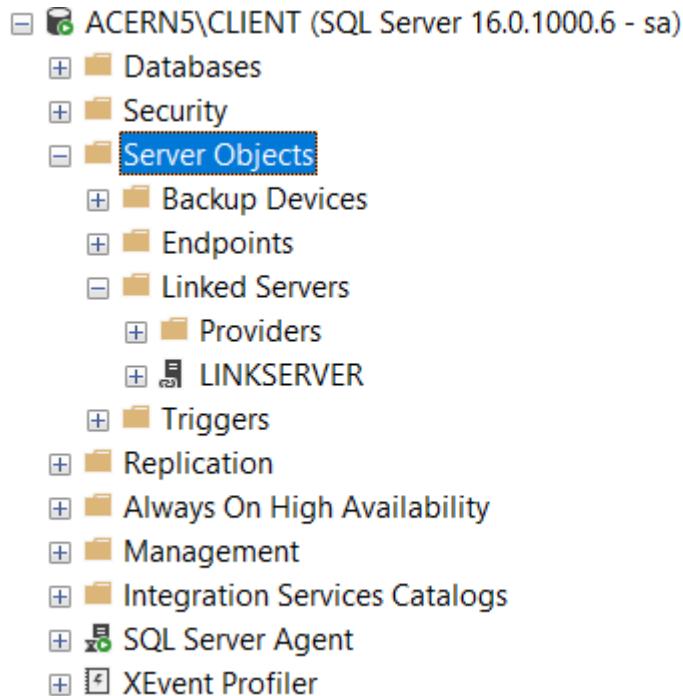
- Cài đặt security CLIENT



- Cài đặt RPC/RPC Out thành True trên cả SERVER/CLIENT



- Kết quả:



Bước 5: Kiểm tra Linked Server đã tạo

- Khi ở CLIENT thì vẫn truy vấn thông tin đầy đủ thông qua LINKSERVER

```
1 | select YEAR(NgayLap) from LINKSERVER.HighlandsCoffee.dbo.HoaDon
10 | 2013
11 | 2013
12 | 2021
13 | 2017
14 | 2020
15 | 2021
16 | 2019
17 | 2015
18 | 2012
19 | 2020
20 | 2019
21 | 2022
22 | 2023
23 | 2024
24 | 2021
25 | 2013
26 | 2024
27 | 2014
28 | 2011

```

Query executed successfully.

VII. Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection:

VII.1. SQL Injection là gì?

SQL Injection là một lỗ hổng bảo mật trong các ứng dụng web, xảy ra khi dữ liệu đầu vào của người dùng không được kiểm tra kỹ càng và được đưa trực tiếp vào câu lệnh SQL. Kẻ tấn công có thể thêm mã SQL để thay đổi hoặc phá hoại các truy vấn, cho phép truy cập trái phép, sửa đổi dữ liệu.

VII.2. Các giải pháp hạn chế tấn công SQL Injection:

VII.2.1. Sử dụng Stored Procedures (Thủ tục lưu trữ):

Thay vì thực thi trực tiếp các câu lệnh SQL, hãy sử dụng stored procedures để truy xuất hoặc chỉnh sửa dữ liệu. Điều này giúp kiểm soát và giảm thiểu nguy cơ SQL Injection.

VII.2.1.1. Stored Procedure cho bảng NhanVien:

```
-- Thêm nhân viên mới
CREATE OR ALTER PROC AddNhanVien
    @HoTenNV NVARCHAR(50),
    @NgaySinh DATETIME,
    @GioiTinh BIT,
    @DiaChi NVARCHAR(100),
    @NgayVaoLam DATETIME
AS
BEGIN
    INSERT INTO NhanVien (MaNV, HoTenNV, NgaySinh, GioiTinh, DiaChi, NgayVaoLam)
        VALUES (dbo.fBornNhanVienId(), @HoTenNV, @NgaySinh, @GioiTinh, @DiaChi,
@NgayVaoLam);
END;
GO
EXEC AddNhanVien
    @HoTenNV = N'Trần Văn A',
    @NgaySinh = '1990-01-01',
    @GioiTinh = 1,
    @DiaChi = N'Hà Nội',
    @NgayVaoLam = '2023-01-01';

-- Lấy thông tin nhân viên
CREATE OR ALTER PROC GetNhanVien
    @MaNV CHAR(10)
AS
BEGIN
    SELECT MaNV, HoTenNV, NgaySinh, GioiTinh, DiaChi, NgayVaoLam
    FROM NhanVien
    WHERE MaNV = @MaNV;
END;
GO
EXEC GetNhanVien
    @MaNV = 'NV00000001';

-- Cập nhật thông tin nhân viên
```

```

CREATE OR ALTER PROC UpdateNhanVien
    @MaNV CHAR(10),
    @HoTenNV NVARCHAR(50),
    @NgaySinh DATETIME,
    @GioiTinh BIT,
    @DiaChi NVARCHAR(100),
    @NgayVaoLam DATETIME
AS
BEGIN
    UPDATE NhanVien
    SET HoTenNV = @HoTenNV,
        NgaySinh = @NgaySinh,
        GioiTinh = @GioiTinh,
        DiaChi = @DiaChi,
        NgayVaoLam = @NgayVaoLam
    WHERE MaNV = @MaNV;
END;
GO
EXEC UpdateNhanVien
    @MaNV = 'NV00000001',
    @HoTenNV = N'Trần Văn B',
    @NgaySinh = '1990-01-02',
    @GioiTinh = 1,
    @DiaChi = N'Dà Nẵng',
    @NgayVaoLam = '2023-01-02';

```

VII.2.1.2. Stored Procedure cho bảng SanPham:

```

-- Thêm sản phẩm mới
CREATE OR ALTER PROC AddSanPham
    @TenSP NVARCHAR(50),
    @DonGiaBan INT
AS
BEGIN
    INSERT INTO SanPham (MaSP, TenSP, DonGiaBan)
    VALUES (dbo.fBornSanPhamId(), @TenSP, @DonGiaBan);
END;
GO
EXEC AddSanPham
    @TenSP = N'Cà phê sữa đá',
    @DonGiaBan = 35000;

-- Lấy thông tin sản phẩm
CREATE OR ALTER PROC GetSanPham
    @MaSP CHAR(10)
AS
BEGIN
    SELECT MaSP, TenSP, DonGiaBan
    FROM SanPham
    WHERE MaSP = @MaSP;
END;
GO
EXEC GetSanPham
    @MaSP = 'SP00000001';

```

```
-- Cập nhật thông tin sản phẩm
CREATE OR ALTER PROC UpdateSanPham
    @MaSP CHAR(10),
    @TenSP NVARCHAR(50),
    @DonGiaBan INT
AS
BEGIN
    UPDATE SanPham
    SET TenSP = @TenSP,
    DonGiaBan = @DonGiaBan
    WHERE MaSP = @MaSP;
END;
GO
EXEC UpdateSanPham
    @MaSP = 'SP00000001',
    @TenSP = N'Cà phê sữa nóng',
    @DonGiaBan = 37000;
```

VII.2.1.3. Stored Procedure cho bảng HoaDon:

```
-- Thêm hóa đơn mới
CREATE OR ALTER PROC AddHoaDon
    @NgayLap DATETIME,
    @TongTien INT,
    @PhuongThucThanhToan BIT,
    @MaNV CHAR(10)
AS
BEGIN
    INSERT INTO HoaDon (MaHD, NgayLap, TongTien, PhuongThucThanhToan, MaNV)
        VALUES (dbo.fBornHoaDonId(), @NgayLap, @TongTien, @PhuongThucThanhToan,
@MaNV);
END;
GO
EXEC AddHoaDon
    @NgayLap = '2023-11-01',
    @TongTien = 500000,
    @PhuongThucThanhToan = 1,
    @MaNV = 'NV00000001';

-- Lấy thông tin hóa đơn
CREATE OR ALTER PROC GetHoaDon
    @MaHD CHAR(10)
AS
BEGIN
    SELECT MaHD, NgayLap, TongTien, PhuongThucThanhToan, MaNV
    FROM HoaDon
    WHERE MaHD = @MaHD;
END;
GO
EXEC GetHoaDon
    @MaHD = 'HD00000001';

-- Cập nhật thông tin hóa đơn
CREATE OR ALTER PROC UpdateHoaDon
    @MaHD CHAR(10),
```

```

@NgayLap DATETIME,
@TongTien INT,
@PhuongThucThanhToan BIT,
@MaNV CHAR(10)
AS
BEGIN
    UPDATE HoaDon
    SET NgayLap = @NgayLap,
    TongTien = @TongTien,
    PhuongThucThanhToan = @PhuongThucThanhToan,
    MaNV = @MaNV
    WHERE MaHD = @MaHD;
END;
GO
EXEC UpdateHoaDon
    @MaHD = 'HD00000001',
    @NgayLap = '2023-11-02',
    @TongTien = 550000,
    @PhuongThucThanhToan = 0,
    @MaNV = 'NV00000002';

```

VII.2.1.4. Stored Procedure cho bảng PhieuNhap:

```

-- Thêm phiếu nhập mới
CREATE OR ALTER PROC AddPhieuNhap
    @NgayLap DATETIME,
    @TongTien INT,
    @TienVAT INT,
    @TongCong INT,
    @PhuongThucThanhToan BIT,
    @MaNCC CHAR(10),
    @MaNV CHAR(10)
AS
BEGIN
    INSERT INTO PhieuNhap (MaPN, NgayLap, TongTien, TienVAT, TongCong,
    PhuongThucThanhToan, MaNCC, MaNV)
    VALUES (dbo.fBornPhieuNhapId(), @NgayLap, @TongTien, @TienVAT, @TongCong,
    @PhuongThucThanhToan, @MaNCC, @MaNV);
END;
GO
EXEC AddPhieuNhap
    @NgayLap = '2023-11-01',
    @TongTien = 600000,
    @TienVAT = 50000,
    @TongCong = 650000,
    @PhuongThucThanhToan = 1,
    @MaNCC = 'NCC0000001',
    @MaNV = 'NV00000001';

-- Lấy thông tin phiếu nhập
CREATE OR ALTER PROC GetPhieuNhap
    @MaPN CHAR(10)
AS
BEGIN

```

```

        SELECT MaPN, NgayLap, TongTien, TienVAT, TongCong, PhuongThucThanhToan, MaNCC,
MaNV
        FROM PhieuNhap
        WHERE MaPN = @MaPN;
END;
GO
EXEC GetPhieuNhap
    @MaPN = 'PN00000001';

-- Cập nhật thông tin phiếu nhập
CREATE OR ALTER PROC UpdatePhieuNhap
    @MaPN CHAR(10),
    @NgayLap DATETIME,
    @TongTien INT,
    @TienVAT INT,
    @TongCong INT,
    @PhuongThucThanhToan BIT,
    @MaNCC CHAR(10),
    @MaNV CHAR(10)
AS
BEGIN
    UPDATE PhieuNhap
    SET NgayLap = @NgayLap,
    TongTien = @TongTien,
    TienVAT = @TienVAT,
    TongCong = @TongCong,
    PhuongThucThanhToan = @PhuongThucThanhToan,
    MaNCC = @MaNCC,
    MaNV = @MaNV
    WHERE MaPN = @MaPN;
END;
GO
EXEC UpdatePhieuNhap
    @MaPN = 'PN00000001',
    @NgayLap = '2023-11-02',
    @TongTien = 700000,
    @TienVAT = 60000,
    @TongCong = 760000,
    @PhuongThucThanhToan = 0,
    @MaNCC = 'NCC0000002',
    @MaNV = 'NV00000002';

```

VII.2.1.5. Stored Procedure cho bảng ChiTietHoaDon:

```

-- Thêm chi tiết hóa đơn
CREATE OR ALTER PROC AddChiTietHoaDon
    @MaHD CHAR(10),
    @MaSP CHAR(10),
    @SoLuong INT,
    @ThanhTien INT
AS
BEGIN
    -- Đảm bảo hóa đơn và sản phẩm tồn tại trước khi thêm
    IF EXISTS (SELECT 1 FROM HoaDon WHERE MaHD = @MaHD)

```

```

        AND EXISTS (SELECT 1 FROM SanPham WHERE MaSP = @MaSP)
    BEGIN
        INSERT INTO ChiTietHoaDon (MaHD, MaSP, SoLuong, ThanhTien)
        VALUES (@MaHD, @MaSP, @SoLuong, @ThanhTien);
    END
    ELSE
    BEGIN
        RAISERROR (N'Hóa đơn hoặc sản phẩm không tồn tại', 16, 1);
    END
END;
GO

-- Lấy chi tiết hóa đơn
CREATE OR ALTER PROC GetChiTietHoaDon
    @MaHD CHAR(10)
AS
BEGIN
    SELECT MaHD, MaSP, SoLuong, ThanhTien
    FROM ChiTietHoaDon
    WHERE MaHD = @MaHD;
END;
GO

-- Cập nhật chi tiết hóa đơn
CREATE OR ALTER PROC UpdateChiTietHoaDon
    @MaHD CHAR(10),
    @MaSP CHAR(10),
    @SoLuong INT,
    @ThanhTien INT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM ChiTietHoaDon WHERE MaHD = @MaHD AND MaSP = @MaSP)
    BEGIN
        UPDATE ChiTietHoaDon
        SET SoLuong = @SoLuong,
            ThanhTien = @ThanhTien
        WHERE MaHD = @MaHD AND MaSP = @MaSP;
    END
    ELSE
    BEGIN
        RAISERROR (N'Chi tiết hóa đơn không tồn tại', 16, 1);
    END
END;
GO

```

VII.2.1.6. Stored Procedure cho bảng ChiTietPhieuNhap:

```

-- Thêm chi tiết phiếu nhập
CREATE OR ALTER PROC AddChiTietPhieuNhap
    @MaPN CHAR(10),
    @MaNL CHAR(10),
    @SoLuong INT,
    @ThanhTien INT,
    @ThueTax DECIMAL(4, 2)
AS

```

```

BEGIN
    -- Đảm bảo phiếu nhập và nguyên liệu tồn tại trước khi thêm
    IF EXISTS (SELECT 1 FROM PhieuNhap WHERE MaPN = @MaPN)
    AND EXISTS (SELECT 1 FROM NguyenLieu WHERE MaNL = @MaNL)
    BEGIN
        INSERT INTO ChiTietPhieuNhap (MaPN, MaNL, SoLuong, ThanTien, ThueTax)
        VALUES (@MaPN, @MaNL, @SoLuong, @ThanhTien, @ThueTax);
    END
    ELSE
    BEGIN
        RAISERROR (N'Phiếu nhập hoặc nguyên liệu không tồn tại', 16, 1);
    END
END;
GO
EXEC AddChiTietPhieuNhap
    @MaPN = 'PN00000001',
    @MaNL = 'NL00000001',
    @SoLuong = 100,
    @ThanhTien = 500000,
    @ThueTax = 0.05;

-- Lấy chi tiết phiếu nhập
CREATE OR ALTER PROC GetChiTietPhieuNhap
    @MaPN CHAR(10)
AS
BEGIN
    SELECT MaPN, MaNL, SoLuong, ThanTien, ThueTax
    FROM ChiTietPhieuNhap
    WHERE MaPN = @MaPN;
END;
GO
EXEC GetChiTietPhieuNhap
    @MaPN = 'PN00000001';

-- Cập nhật chi tiết phiếu nhập
CREATE OR ALTER PROC UpdateChiTietPhieuNhap
    @MaPN CHAR(10),
    @MaNL CHAR(10),
    @SoLuong INT,
    @ThanhTien INT,
    @ThueTax DECIMAL(4, 2)
AS
BEGIN
    IF EXISTS (SELECT 1 FROM ChiTietPhieuNhap WHERE MaPN = @MaPN AND MaNL = @MaNL)
    BEGIN
        UPDATE ChiTietPhieuNhap
        SET SoLuong = @SoLuong,
            ThanTien = @ThanhTien,
            ThueTax = @ThueTax
        WHERE MaPN = @MaPN AND MaNL = @MaNL;
    END
    ELSE
    BEGIN
        RAISERROR (N'Chi tiết phiếu nhập không tồn tại', 16, 1);
    END

```

```

END;
GO
EXEC UpdateChiTietPhieuNhap
    @MaPN = 'PN00000001',
    @MaNL = 'NL00000001',
    @SoLuong = 120,
    @ThanhTien = 600000,
    @ThueTax = 0.07;

```

VII.2.1.7. Stored Procedure cho bảng NhaCungCap:

```

-- Thêm nhà cung cấp mới
CREATE OR ALTER PROC AddNhaCungCap
    @TenNCC NVARCHAR(100),
    @DiaChi NVARCHAR(100)
AS
BEGIN
    INSERT INTO NhaCungCap (MaNCC, TenNCC, DiaChi)
        VALUES (dbo.fBornNhaCungCapId(), @TenNCC, @DiaChi);
END;
GO
EXEC AddNhaCungCap
    @TenNCC = N'Công ty Sửa ABC',
    @DiaChi = N'TP. HCM';

-- Lấy thông tin nhà cung cấp
CREATE OR ALTER PROC GetNhaCungCap
    @MaNCC CHAR(10)
AS
BEGIN
    SELECT MaNCC, TenNCC, DiaChi
    FROM NhaCungCap
    WHERE MaNCC = @MaNCC;
END;
GO
EXEC GetNhaCungCap
    @MaNCC = 'NCC0000001';

-- Cập nhật thông tin nhà cung cấp
CREATE OR ALTER PROC UpdateNhaCungCap
    @MaNCC CHAR(10),
    @TenNCC NVARCHAR(100),
    @DiaChi NVARCHAR(100)
AS
BEGIN
    IF EXISTS (SELECT 1 FROM NhaCungCap WHERE MaNCC = @MaNCC)
    BEGIN
        UPDATE NhaCungCap
        SET TenNCC = @TenNCC,
            DiaChi = @DiaChi
        WHERE MaNCC = @MaNCC;
    END
    ELSE

```

```

BEGIN
RAISERROR (N'Nhà cung cấp không tồn tại', 16, 1);
END
GO
EXEC UpdateNhaCungCap
    @MaNCC = 'NCC00000001',
    @TenNCC = N'Công ty Sửa XYZ',
    @DiaChi = N'Hà Nội';

```

VII.2.1.8. Stored Procedure cho bảng NguyenLieu:

```

-- Thêm nguyên liệu mới
CREATE OR ALTER PROC AddNguyenLieu
    @TenNL NVARCHAR(50),
    @SoLuong INT,
    @DonGiaNhap INT,
    @DonViTinh CHAR(10)
AS
BEGIN
    INSERT INTO NguyenLieu (MaNL, TenNL, SoLuong, DonGiaNhap, DonViTinh)
        VALUES (dbo.fBornValueNguyenLieuId() @TenNL, @SoLuong, @DonGiaNhap,
@DonViTinh);
END;
GO
EXEC AddNguyenLieu
    @TenNL = N'Sửa đặc',
    @SoLuong = 100,
    @DonGiaNhap = 10000,
    @DonViTinh = N'Hộp';

-- Lấy thông tin nguyên liệu
CREATE OR ALTER PROC GetNguyenLieu
    @MaNL CHAR(10)
AS
BEGIN
    SELECT MaNL, TenNL, SoLuong, DonGiaNhap, DonViTinh
    FROM NguyenLieu
    WHERE MaNL = @MaNL;
END;
GO
EXEC GetNguyenLieu
    @MaNL = 'NL00000001';

-- Cập nhật thông tin nguyên liệu
CREATE OR ALTER PROC UpdateNguyenLieu
    @MaNL CHAR(10),
    @TenNL NVARCHAR(50),
    @SoLuong INT,
    @DonGiaNhap INT,
    @DonViTinh CHAR(10)
AS
BEGIN
    IF EXISTS (SELECT 1 FROM NguyenLieu WHERE MaNL = @MaNL)

```

```

BEGIN
UPDATE NguyenLieu
SET TenNL = @TenNL,
    SoLuong = @SoLuong,
    DonGiaNhap = @DonGiaNhap,
    DonViTinh = @DonViTinh
WHERE MaNL = @MaNL;
END
ELSE
BEGIN
RAISERROR (N'Nguyên liệu không tồn tại', 16, 1);
END
END;
GO
EXEC UpdateNguyenLieu
    @MaNL = 'NL00000001',
    @TenNL = N'Sữa đặc ngọt',
    @SoLuong = 150,
    @DonGiaNhap = 12000,
    @DonViTinh = N'Hộp';

```

VII.2.2. Dùng Views để hạn chế truy cập trực tiếp:

Sử dụng VIEW để hiển thị dữ liệu giới hạn thay vì truy cập trực tiếp các bảng, giúp người dùng chỉ xem được các trường cần thiết và không thể truy cập toàn bộ dữ liệu.

Bước 1: Tạo View cho tất cả bảng:

VII.2.2.1. View cho Bảng SanPham:

```

CREATE OR ALTER VIEW View_SanPham AS
SELECT TenSP, DonGiaBan
FROM SanPham;
GO
SELECT * FROM View_SanPham

```

TenSP	DonGiaBan
1 San pham 1	34120
2 San pham 2	69025
3 San pham 3	45876
4 San pham 4	15313
5 San pham 5	47577
6 San pham 6	51494
7 San pham 7	11346
8 San pham 8	35489
9 San pham 9	53489
10 San pham 10	9901
11 San pham 11	29423
12 San pham 12	58945
13 San pham 13	34645
14 San pham 14	99467
15 San pham 15	48573
16 San pham 16	49863
San pham 17	86662

VII.2.2.2. View cho Bảng NguyenLieu:

```

CREATE OR ALTER VIEW View_NguyenLieu AS
SELECT TenNL, SoLuong, DonGiaNhap, DonViTinh

```

```

FROM NguyenLieu;
GO
SELECT * FROM View_NguyenLieu

```

	TenNL	SoLuong	DonGiaNhap	DonViTinh
1	Nguyen lieu 1	5	9954	Thung
2	Nguyen lieu 2	10	2293	Hop
3	Nguyen lieu 3	15	2464	Ky
4	Nguyen lieu 4	20	5193	Thung
5	Nguyen lieu 5	25	3329	Hop
6	Nguyen lieu 6	30	3007	Ky
7	Nguyen lieu 7	35	2784	Thung
8	Nguyen lieu 8	40	4939	Hop
9	Nguyen lieu 9	45	1375	Ky
10	Nguyen lieu 10	0	1644	Thung
11	Nguyen lieu 11	5	4949	Hop
12	Nguyen lieu 12	10	1503	Ky
13	Nguyen lieu 13	15	9351	Thung
14	Nguyen lieu 14	20	1920	Hop
15	Nguyen lieu 15	25	7195	Ky
16	Nguyen lieu 16	30	1011	Thung
17	Nguyen lieu 17	25	200	Hop

VII.2.2.3. View cho Bảng NhaCungCap:

```

CREATE OR ALTER VIEW View_NhaCungCap AS
SELECT TenNCC, DiaChi
FROM NhaCungCap;
GO
SELECT * FROM View_NhaCungCap

```

	TenNCC	DiaChi
1	Nhà cung cấp 1	Địa chỉ nhà cung cấp 1
2	Nhà cung cấp 2	Địa chỉ nhà cung cấp 2
3	Nhà cung cấp 3	Địa chỉ nhà cung cấp 3
4	Nhà cung cấp 4	Địa chỉ nhà cung cấp 4
5	Nhà cung cấp 5	Địa chỉ nhà cung cấp 5
6	Nhà cung cấp 6	Địa chỉ nhà cung cấp 6
7	Nhà cung cấp 7	Địa chỉ nhà cung cấp 7
8	Nhà cung cấp 8	Địa chỉ nhà cung cấp 8
9	Nhà cung cấp 9	Địa chỉ nhà cung cấp 9
10	Nhà cung cấp 10	Địa chỉ nhà cung cấp 10
11	Nhà cung cấp 11	Địa chỉ nhà cung cấp 11
12	Nhà cung cấp 12	Địa chỉ nhà cung cấp 12
13	Nhà cung cấp 13	Địa chỉ nhà cung cấp 13
14	Nhà cung cấp 14	Địa chỉ nhà cung cấp 14
15	Nhà cung cấp 15	Địa chỉ nhà cung cấp 15
16	Nhà cung cấp 16	Địa chỉ nhà cung cấp 16

Bước 2: Cấp quyền truy cập View:

- Nếu muốn cấp quyền **SELECT** cho người dùng NhanVien để họ có thể xem dữ liệu nhưng không chỉnh sửa.

```

GRANT SELECT ON View_SanPham TO "NhanVien";
GRANT SELECT ON View_NguyenLieu TO "NhanVien";
GRANT SELECT ON View_NhaCungCap TO "NhanVien";
GO

```

- Nếu cần thu hồi quyền truy cập, có thể dùng lệnh REVOKE như sau:

```
REVOKE SELECT ON View_SanPham FROM "NhanVien";
```

```
REVOKE SELECT ON View_NguyenLieu FROM "NhanVien";
REVOKE SELECT ON View_NhaCungCap FROM "NhanVien";
GO
```