**ĐẠI HỌC ĐÀ NẴNG - TRƯỜNG ĐẠI HỌC KINH TẾ**

**HỆ THỐNG THÔNG TIN QUẢN LÝ**

****

**HỌC PHẦN: QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**R1 - R8**

**HỆ THỐNG QUẢN LÝ NHÂN SỰ**

**Nhóm 48K212.01**

**Giảng viên hướng dẫn: Cao Thị Nhâm**

**Thành viên nhóm: Trần Thị Thùy Trang (Nhóm trưởng)**

**Lê Thị Phương Thảo**

**Lê Thị Kim Cương**

**Trịnh Thị Khánh Huyền**

**Nguyễn Thị Cẩm Nhi**

**Đà Nẵng, ngày 29 tháng 11 năm 2024**

**MỤC LỤC**

[**1 SƠ ĐỒ QUAN HỆ: 1**](#_heading=h.gjdgxs)

[**2 BẢNG MÔ TẢ CHI TIẾT: 2**](#_heading=h.30j0zll)

[**2.1 Bảng Nhân viên 2**](#_heading=h.1fob9te)

[**2.2 Bảng ChamCong 3**](#_heading=h.3znysh7)

[**2.3 Bảng Luong 3**](#_heading=h.2et92p0)

[**2.4 Bảng DanhGiaNangLuc 4**](#_heading=h.tyjcwt)

[**2.5 Bảng CongViec 4**](#_heading=h.3dy6vkm)

[**2.6 Bảng CongViec\_BaoCao 5**](#_heading=h.1t3h5sf)

[**2.7 Bảng BangCap: 6**](#_heading=h.4d34og8)

[**2.8 Bảng KyNang: 6**](#_heading=h.2s8eyo1)

[**2.9 Bảng ChungChi: 7**](#_heading=h.17dp8vu)

[**2.10 Bảng LoiPhat: 7**](#_heading=h.3rdcrjn)

[**2.11 Bảng NghiPhep: 7**](#_heading=h.26in1rg)

[**2.12 Bảng ChucVu: 8**](#_heading=h.lnxbz9)

[**2.13 Bảng PhongBan: 8**](#_heading=h.35nkun2)

[**3 MODULES 9**](#_heading=h.1ksv4uv)

[**4 CƠ CHẾ BẢO MẬT: 21**](#_heading=h.44sinio)

[**4.1 Quản lý người dùng (phân quyền): 22**](#_heading=h.2jxsxqh)

[**4.2 Always Encrypted (Mã hóa dữ liệu): 33**](#_heading=h.z337ya)

[**5 PHÂN TÁN DỮ LIỆU: 39**](#_heading=h.3j2qqm3)

[**5.1 Lý do lựa chọn giải pháp phân tán dữ liệu: 39**](#_heading=h.1y810tw)

[**5.2 Kịch bản phân tán: 39**](#_heading=h.4i7ojhp)

[**5.3 DEMO: 40**](#_heading=h.2xcytpi)

[**6 BACKUP: 60**](#_heading=h.1ci93xb)

[**6.1 Lịch trình backup 60**](#_heading=h.3whwml4)

[**6.2 DEMO BACKUP 61**](#_heading=h.2bn6wsx)

[**6.3 RESTORE 80**](#_heading=h.qsh70q)

[**7 STORED PROCEDURE` NGĂN CHẶN TẤN CÔNG SQL INJECTION: 87**](#_heading=h.3as4poj)

[**7.1 Đăng nhập 87**](#_heading=h.1pxezwc)

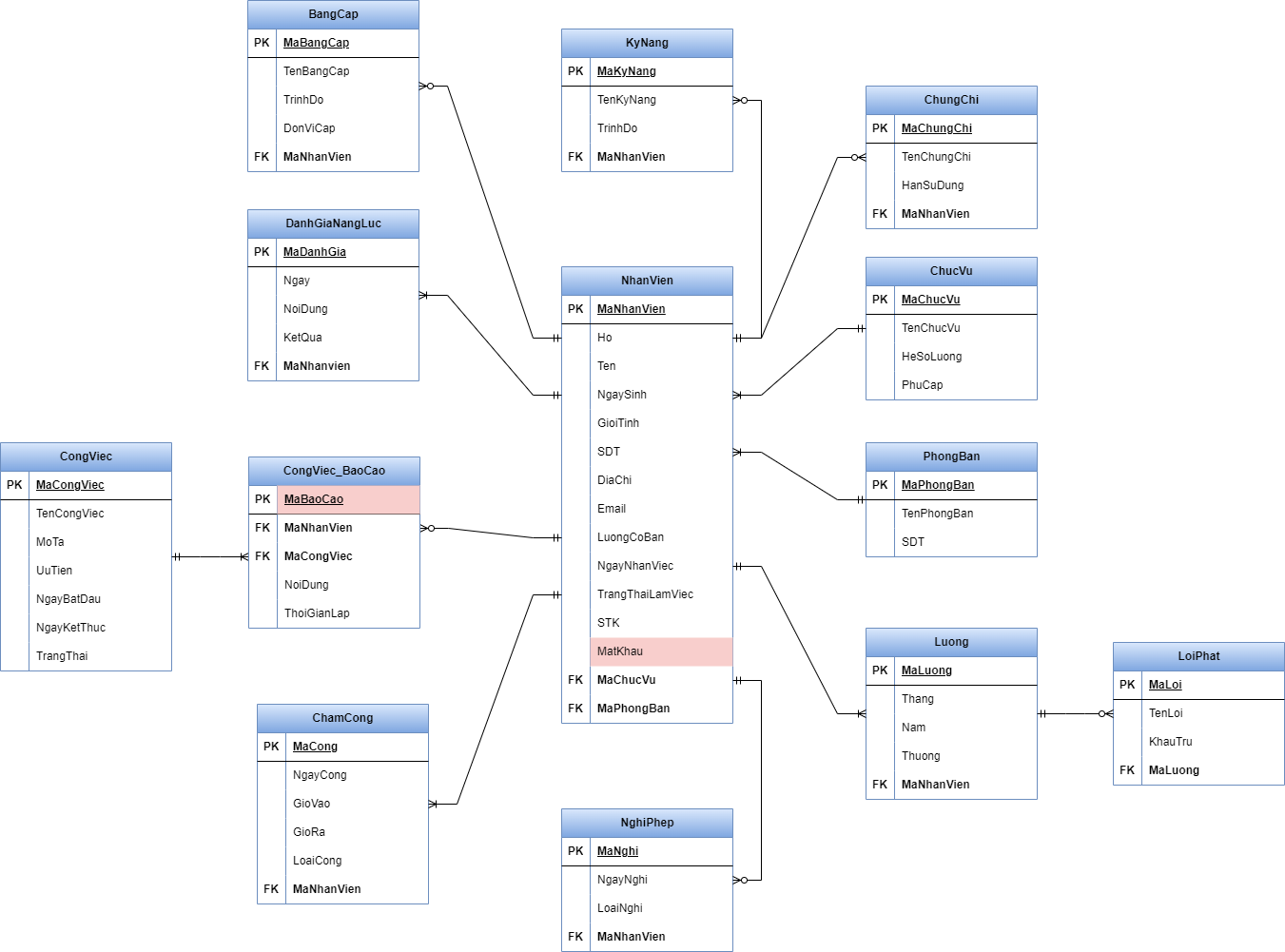
[**7.2 Tìm kiếm thông tin nhân viên 89**](#_heading=h.49x2ik5)

[**7.3 Cập nhật thông tin nhân viên 90**](#_heading=h.2p2csry)

[**7.4 Xóa thông tin Nhân viên 92**](#_heading=h.147n2zr)

# SƠ ĐỒ QUAN HỆ:

*(các thuộc tính bôi đỏ là những thay đổi so với thiết kế ban đầu)*



# BẢNG MÔ TẢ CHI TIẾT:

*(các thuộc tính bôi đỏ là những thay đổi so với thiết kế ban đầu)*

## Bảng Nhân viên

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaNhanVien | CHAR(5) | Primary Key, not null | Bắt đầu bằng “NV…” |
| 2 | Ho | NVARCHAR(50) | not null |  |
| 3 | Ten | NVARCHAR(30) | not null |  |
| 4 | NgaySinh | DATE | not null |  |
| 5 | GioiTinh | BIT | not null | 0: Nam, 1: Nữ |
| 6 | SDT | CHAR(10) | not null, unique |  |
| 7 | DiaChi | NVARCHAR(50) | not null |  |
| 8 | Email | VARCHAR(20) | not null, unique |  |
| 9 | LuongCoBan | NUMERIC(10,3) | not null |  |
| 10 | NgayNhanViec | DATE | not null |  |
| 11 | TrangThaiLamViec | BIT | not null | 0: trạng thái đã nghỉ việc  1: trạng thái đang làm việc |
| 12 | STK | VARCHAR(50) | not null, unique |  |
| 13 | MatKhau | VARCHAR(50) | Not null | Bổ sung |
| 13 | MaChucVu | CHAR(5) | Foreign Key, not null | Bắt đầu bằng “CV…” |
| 14 | MaPhongBan | CHAR(5) | Foreign Key, not null | Bắt đầu bằng “PB…” |

## Bảng ChamCong

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaCong | CHAR(11) | Primary key, not null | Bắt đầu bằng “MC…” |
| 2 | NgayCong | DATE | not null |  |
| 3 | GioVao | TIME | null | Đã thay đổi với ban đầu là not null |
| 4 | GioRa | TIME | null | Đã thay đổi với ban đầu là not null |
| 5 | LoaiCong | BIT | not null | 0: ngày thường  1:Ngày công tác |
| 6 | MaNhanVien | CHAR(5) | Foreign Key, not null | Bắt đầu bằng “NV…” |

## Bảng Luong

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaLuong | CHAR(7) | Primary Key, not null |  |
| 2 | Thang | INT | Not null |  |
| 3 | Nam | INT | Not null |  |
| 7 | Thuong | NUMERIC(10,3) | Null |  |
| 10 | MaNhanVien | CHAR(5) | Foreign Key, Not null |  |

## Bảng DanhGiaNangLuc

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaDanhGia | CHAR(5) | Primary Key, not null | Cấu trúc:  ‘DG…’ |
| 2 | Ngay | DATE | not null |  |
| 3 | NoiDung | NVARCHAR(200) | not null |  |
| 4 | KetQua | INT | not null | 0: TỐT  1: KHÁ  2: CHƯA ĐẠT |
| 5 | MaNhanVien | CHAR(5) | Foreign Key, Not null | Khoá ngoại tham chiếu đến bảng nhân viên |

## Bảng CongViec

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaCongViec | CHAR(5) | Primary Key, not null | Cấu trúc:  ‘CV…’ |
| 2 | TenCongViec | NVARCHAR(50) | not null |  |
| 3 | MoTa | NVARCHAR(100) | null |  |
| 4 | UuTien | INT | not null | 0: Nghiêm trọng  1: Cao  2: Trung bình  3: Thấp |
| 5 | NgayBatDau | DATE | not null |  |
| 6 | NgayKetThuc | DATE | not null |  |
| 7 | Trangthai | BIT | not null | 0: chưa hoàn thành  1:đã hoàn thành |

## Bảng CongViec\_BaoCao

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaBaoCao | CHAR(8) | Primary key, not null | ***Bổ sung*** |
| 2 | MaNhanVien | CHAR(5) | Foreign key, not null | Khoá ngoại tham chiếu đến bảng NhanVien |
| 3 | MaCongViec | CHAR(5) | Foreign key, not null | Khoá ngoại tham chiếu đến bảng CongViec |
| 4 | NoiDung | NVARCHAR(200) | not null |  |
| 5 | ThoiGianLap | Datetime | not null |  |

## Bảng BangCap:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaBangCap | CHAR(5) | Primary Key, not null | Cấu trúc:  ‘BC…’ |
| 2 | TenBangCap | NVARCHAR(100) | not null |  |
| 3 | TrinhDo | NVARCHAR(30) | not null |  |
| 4 | DonViCap | NVARCHAR(100) | not null |  |
| 5 | MaNhanVien | CHAR(5) | Foreign Key, not null |  |

## Bảng KyNang:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaKyNang | CHAR(5) | Primary Key, not null | Cấu trúc:  ‘KN…’ |
| 2 | TenKyNang | NVARCHAR(50) | not null |  |
| 3 | TrinhDo | NVARCHAR(20) | not null |  |
| 4 | MaNhanVien | CHAR(5) | Foreign Key, not null |  |

## Bảng ChungChi:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaChungChi | CHAR(5) | Primary Key, not null | Cấu trúc:  ‘CC…’ |
| 2 | TenChungChi | NVARCHAR(50) | not null |  |
| 3 | HanSuDung | Date | not null |  |
| 4 | MaNhanVien | CHAR(5) | Foreign Key, not null |  |

## Bảng LoiPhat:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaLoi | CHAR(5) | Primary Key, not null | Cấu trúc:  ‘ML…’ |
| 2 | TenLoi | NVARCHAR(20) | Not null |  |
| 3 | TienPhat | NUMERIC(10,3) | Not null |  |
| 4 | MaLuong | CHAR(5) | Foreign Key, Not null |  |

## Bảng NghiPhep:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaNghi | CHAR(5) | Primary Key, not null | Bắt đầu bằng “MN…” |
| 2 | NgayNghi | DATE | not null |  |
| 3 | LoaiNghi | BIT | not null | 0: Nghỉ không phép  1: Nghỉ có phép |
| 4 | MaNhanVien | CHAR(5) | Foreign Key, Not null | Bắt đầu bằng “NV…” |

## Bảng ChucVu:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaChucVu | CHAR(5) | Primary Key, not null | Bắt đầu bằng “CVu…” |
| 2 | TenChucVu | NVARCHAR(30) | not null |  |
| 3 | HeSoLuong | NUMERIC(4,2) | not null |  |
| 4 | PhuCap | NUMERIC(10,3) | Not null |  |

## Bảng PhongBan:

| STT. | Tên trường | Kiểu dữ liệu | Ràng buộc dữ liệu | Ghi chú |
| --- | --- | --- | --- | --- |
| 1 | MaPhongBan | CHAR(5) | Primary Key, not null | Bắt đầu bằng “PB…” |
| 2 | TenPhongBan | NVARCHAR(30) | not null |  |
| 3 | SDT | CHAR(10) | not null, UNIQUE |  |

# MODULES

1. Khi thực hiện xóa thông tin nhân viên khỏi bảng NhanVien. Hãy đổi trạng thái làm việc của Nhân viên thành 0 (đã nghỉ việc) thay vì xóa.

| Mục đích | * Module phục vụ cho chức năng XoaThongTinNhanVien(), đảm bảo duy trì lịch sử làm việc và tránh mất mát thông tin. |
| --- | --- |
| Input | * Không |
| Output | * Không |
| Loại | * Instead of |
| Sự kiện | * Delete |
| Bảng | * NhanVien |
| Process | * 1. Lấy ra MaNhanVien từ bảng deleted => @MaNhanVien * 2. Update NhanVien:       set TrangThaiLamViec = ‘0’      điều kiện: MaNhanVien = @MaNhanVien |

1. Khi thêm mới dữ liệu bảng CongViec\_BaoCao, hãy kiểm tra tính hợp lệ của cột ThoiGianLap .Nếu ThoiGianLap không nằm trong NgayBatDau và NgayKetThuc của CongViec thì Thông báo lỗi ThoiGianLap không hợp lệ và hủy bỏ toàn bộ thao tác.

| Mục đích | * Module phục vụ cho chức năng LapBaoCaoCV(),kiểm tra sự hợp lệ của dữ liệu đầu vào (cột ThoiGianLap) trước khi thêm mới dữ liệu vào bảng CongViec\_BaoCao và  Đảm bảo rằng thời gian lập báo cáo (ThoiGianLap) nằm trong khoảng thời gian hợp lệ của công việc tương ứng. |
| --- | --- |
| Bảng | * CongViec\_BaoCao |
| Loại | * After |
| Sự Kiện | * Insert |
| Process | * 1.  Lấy ra ThoiGianLap,MaCongViec từ bảng inserted --> @ThoiGianLap, @MaCongViec * 2.  Lấy ra NgayBatDau, NgayKetThuc từ bảng CongViec → @NgayBatDau, @NgayKetThuc * 3.  Kiểm tra sự hợp lệ ThoiGianLap phải nằm trong NgayBatDau và NgayKetThuc của CongViec                  3a.Nếu @ThoiGianLap không nằm trong NgayBatDau và   NgayKetThuc của CongViec: print 'Thời gian lập không hợp lệ'+ Rollback      3b.Ngược lại: Insert dữ liệu vào bảng CongViec\_BaoCao |

1. In ra danh sách nhân viên có ngày công trùng với ngày nghỉ phép, với các cột sau: Mã nhân viên, tên nhân viên, ngày công, Mã công, ngày nghỉ phép, loại nghỉ.

| Mục đích | * Module phục vụ cho chức năng READ (đọc dữ liệu) của dữ liệu công - ngày nghỉ phép, nhằm phát hiện các xung đột hoặc lỗi trong việc quản lý chấm công và nghỉ phép. |
| --- | --- |
| Input | * Không |
| Output | * In ra danh sách bao gồm(Mã nhân viên,Tên nhân viên,Mã công,Ngày công,Mã nghỉ phép,Ngày nghỉ, Loại nghỉ) |
| Process | 1. Lấy ra Mã nhân viên, Tên nhân viên, Ngày công, Mã công, Mã nghỉ, Ngày nghỉ. Loại nghỉ từ bảng NhanVien, trong đó Bảng NhanVien được nối với bảng ChamCong với khóa ngoại là MaNhanVien, và bảng NghiPhep với khóa ngoại là MaNhanVien.              Điều kiện: NgayCong ở bảng ChamCong = NgayNghi ở  bảng NghiPhep |

1. Trả về danh sách nhân viên còn làm việc của một Phòng Ban nếu biết mã phòng ban.

| Mục đích | * Module phục vụ cho chức năng READ (đọc dữ liệu) danh sách nhân viên của một phòng ban. |
| --- | --- |
| Input | * Mã phòng ban |
| Output | * Danh sách nhân viên của một phòng ban |
| Process | * 1. Lấy ra mã phòng ban => @MaPhongBan * 2. Trả về danh sách nhân viên của một phòng ban với điều kiện MaPhongBan=@MaPhongBan, Trạng thái làm việc = 1 |

1. Trả về danh sách nhân viên còn làm việc có cùng chức vụ nếu biết mã chức vụ.

| * Mục đích | Module này phục vụ cho chức năng tìm danh sách nhân viên có cùng chức vụ và đảm bảo truy xuất đầy đủ, chính xác thông tin nhân viên thuộc cùng một chức vụ cụ thể trong hệ thống. |
| --- | --- |
| * Input | @MaChucVu |
| * Output | Danh sách Nhân Viên có cùng chức vụ |
| * Process | 1. Nhận tham số đầu vào là @MaChucVu  2. Lấy ra danh sách nhân viên có cùng chức vụ bao gồm: MaNhanVien, Ho, Ten, NgaySinh, GioiTinh, SDT, DiaChi, Email, LuongCoBan, NgayNhanViec, TrangThaiLamViec, STK, MaChucVu, MaPhongBan từ bảng NhanVien với điều kiện MaNhanVien = @MaNhanVien và Trạng Thái làm việc =1 |
| * Script | CREATE OR ALTER PROC spDsNhanVien\_CungChucvu  (  @MaChucVu char(6)  )  AS  BEGIN  -- Lấy danh sách nhân viên có cùng chức vụ  SELECT MaChucVu,MaNhanVien, Ho, Ten, NgaySinh, GioiTinh, SDT, DiaChi, Email, LuongCoBan, NgayNhanViec, TrangThaiLamViec, STK, MaPhongBan  FROM NhanVien  WHERE MaChucVu = @MaChucVu  END  GO  exec spDsNhanVien\_CungChucvu 'CV0010' |

1. Thêm một nhân viên mới vào bảng NhanVien nếu biết các thông tin về Họ, Tên, Ngày sinh, Giới tính, SĐT, Địa chỉ, Email, Lương cơ bản, Ngày nhận viện, Trạng thái làm việc, Số tài khoản, Mã Chức Vụ, Mã Phòng Ban. Công việc cần làm bao gồm:
   1. Kiểm tra Họ, Tên, Số điện thoại đã có trong bảng Nhân viên chưa. Nếu rồi thì thông báo “Đã tồn tại nhân viên” và ngừng xử lý.
   2. Kiểm tra Mã chức vụ có tồn tại trong bảng Chức vụ không. Nếu không hãy đưa ra thông báo “Mã chức vụ không tồn tại” và ngừng xử lý.
   3. Kiểm tra Mã phòng ban có tồn tại trong bảng Phòng ban không. Nếu không hãy đưa ra thông báo “Mã phòng ban không tồn tại” và ngừng xử lý.
   4. Kiểm tra Email nhập phải đúng định dạng. Nếu không hãy đưa ra thông báo “Email sai định dạng” và ngừng xử lý.
   5. Tính Mã nhân viên mới. Mã nhân viên tiếp theo được tính như sau: MAX(mã nhân viên đang có) + 1. Hãy đảm bảo số lượng ký tự luôn đúng với quy định về mã nhân viên.
   6. Thêm mới bản ghi vào bảng NhanVien

| Mục đích | * Module phục vụ cho chức năng ThemNhanVien()  và đảm bảo rằng dữ liệu nhân viên mới được thêm vào một cách chính xác và tránh sự trùng lặp. |
| --- | --- |
| Input | * Họ, Tên, Ngày sinh, Giới tính, SĐT, Địa chỉ, Email, Lương cơ bản, Ngày nhận viện, Trạng thái làm việc, Số tài khoản, Mã Chức Vụ, Mã Phòng Ban. |
| Output | * Xác nhận kết quả thực hiện (1:insert; 0:lỗi/không insert) |
| Process | * 1. Kiểm tra @Ho; @Ten; @Cust\_phone đã tồn tại trong bảng Nhân viên chưa. * Nếu đã tồn tại => Hiển thị thông báo “Đã tồn tại nhân viên” và ngừng xử lý => output 0. * 3. Kiểm tra @MaChucVu được thêm vào có tồn tại trong bảng Chức vụ không. * Nếu chưa tồn tại => Hiển thị thông báo “Mã chức vụ không tồn tại” và ngừng xử lý   => output 0.   * 4. Kiểm tra @MaPhongBan được thêm vào có tồn tại trong bảng Phòng ban không. * Nếu chưa tồn tại => Hiển thị thông báo “Mã phòng ban không tồn tại” và ngừng xử lý   => output 0.   * 5. Kiểm tra @Email đã nhập đúng định dạng chưa:   Tên người dùng + ‘@gmail.com’  (Tên người dùng có thể bao gồm các ký tự: chữ cái (a-z, A-Z), chữ số (0-9), dấu chấm (.), dấu gạch dưới (\_), dấu gạch ngang (-), nhưng không được bắt đầu hoặc kết thúc bằng dấu chấm, dấu gạch dưới, hoặc dấu gạch ngang)   * Nếu không đúng định dạng hãy đưa ra thông báo “Email sai định dạng” và ngừng xử lý => output 0 * 5. Tính mã nhân viên mới: * 5.1 Xác định mã nhân viên lớn nhất hiện tại: Max (cust\_id) * 5.2 Tạo ra mã nhân viên mới: cộng mã nhân viên lớn nhất hiện tại với 1 * format số => 6 ký tự * 6. Thêm mới bản ghi vào bảng NhanVien => Insert thành công => output 1. |

1. Thêm mới dữ liệu trong Bảng ChamCong nếu biết các thông tin sau mã công, ngày công, giờ vào, giờ ra, loại công, mã nhân viên. Hãy thực hiện các công việc sau:
   1. Tính mã chấm công mới với công thức MAX(mã chấm công đang có) + 1. Hãy đảm bảo số lượng ký tự luôn đúng với quy định về mã chấm công.
   2. Kiểm tra ngày công nhập có lớn hơn ngày hiện tại không. Nếu có thì thông báo ‘Ngày công không hợp lệ’ và ngừng xử lý
   3. Kiểm tra Mã nhân viên được nhập có tồn tại trong bảng NhanVien không. Nếu không hãy thông báo “Không tồn tại mã nhân viên” và hủy thao tác đã thực hiện.
   4. Kiểm tra TrangThaiLamViec của Nhân viên. Nếu TrangThaiLamViec = 0 thì đưa ra thông báo ‘Nhân viên đã nghỉ việc’ và hủy thao tác đã thực hiện.
   5. Kiểm tra nếu GioRa nhỏ hơn GioVao thì đưa ra thông báo “Thời gian không hợp lệ” và hủy toàn bộ thao tác.
   6. Thêm mới bảng ghi vào Bảng công với dữ liệu đã có

| Mục đích | * Phục vụ cho phương thức Thêm mới chấm công |
| --- | --- |
| Input | * NgayCong ,GioVao, GioRa, LoaiCong, MaNhanVien |
| Output | * xác nhận kết quả thực hiện (0:lỗi/chưa insert; 1: đã insert) |
| Process | 1. Khai báo biến @TrangThaiLamViec, @Tienphat, @maxCongID, @newCongID 2. Lấy mã công lớn nhất hiện có MAX(mã công) 3. Tính mã công mới = ‘MC’ + max(mã công hiện có) + 1, đảm bảo mã công mới đủ format 11 ký tự 4. Kiểm tra @NgayCong >getdate() hay không. Nếu có thì thông báo ‘Ngày công không hợp lệ’ và ngừng xử lý => output 0 5. Kiểm tra @MaNhanVien tồn tại trong bảng NhanVien không.   Nếu không hãy thông báo ‘Mã nhân viên không tồn tại’ và ngừng xử lý => output 0  6. Kiểm tra Trạng thái làm việc của nhân viên với điều kiện MaNhanVien = @MaNhanVien   * Nếu Trạng thái = 0 thì thông báo ‘Nhân viên đã nghỉ việc’ và ngừng xử lý => output 0   7. Kiểm tra Giờ ra < giờ vào không. Nếu có thì thông báo ‘Thời gian không hợp lệ’ và ngừng xử lý => output 0  8. Thêm mới vào bảng Chấm công với dữ liệu đầu vào   * Nếu thêm mới thành công => output 1 * Nếu thêm mới không thành công => output 0 |

1. Thêm mới một bản ghi vào bảng CongViec nếu biết :Tên Công việc, Mô tả, Ưu tiên, TrangthaiCV, Ngày bắt đầu, Ngày kết thúc. Bao gồm những công việc sau: (phục vụ phương thức )

a. Kiểm tra Ngày bắt đầu và ngày kết thúc có hợp lệ không? Nếu không, ngừng xử lý (Hợp lệ nếu Ngày bắt đầu không được lớn hơn Ngày kết thúc )

b. Tính mã công việc mới. mã công việc mới bằng MAX(mã công việc cũ) + 1

c. Thêm mới bản ghi vào bảng CongViec với dữ liệu đã có.

| Mục đích | * Module phục vụ cho chức năng ThemCongViec() và đảm bảo rằng dữ liệu công việc mới được thêm vào một cách chính xác và tránh sự trùng lặp. |
| --- | --- |
| Input | * Mã công việc, Tên Công việc, Mô tả, Ưu tiên, Ngày bắt đầu, Ngày kết thúc. |
| Output | * Xác nhận kết quả thực hiện:   1: Insert thành công/đã thêm mới  0: Insert thất bại/ chưa thêm được |
| Process | 1. Kiểm tra ngày bắt đầu và ngày kết thúc  1.1: Hợp lệ ( NgayKetThuc >= NgayBatDau) => Tiếp tục B2  1.2: Không hợp lệ => ngừng xử lý => output 0  2. Tính mã công việc mới:  2.1: Lấy ra max(mã công việc cũ)  2.2: Lấy phần số từ MaCongViec, loại bỏ ký tự 'C' và tăng thêm 1  2.3: Tạo mã công việc mới với định dạng C00001, C00002, ... = max(mã công việc cũ) +1, đảm bảo format 6 ký tự.  3.  Insert bản ghi vào bảng CongViec với các giá trị: Tên Công việc, Mô tả, Ưu tiên, TrangthaiCV (mặc định là 0), Ngày bắt đầu, Ngày kết thúc   * Nếu insert thành công => output 1 * Nếu insert không thành công => output 0 |

1. Tính tổng số ngày công đạt chuẩn của nhân viên trong bảng ChamCong với tham số đầu vào là tháng/năm. Ngày công đạt chuẩn là ngày công có đủ Giờ vào và giờ ra và làm việc >= 3 giờ.

| Mục đích | * Module phục vụ cho chức năng TinhSoNgayCong(), và đảm bảo tính toán tổng số công trong 1 tháng của 1 nhân viên bất kì |
| --- | --- |
| Input | * Mã nhân viên, Tháng, Năm |
| Output | * Tổng ngày công |
| Process | 1. Lấy ra Tổng ngày công → @TongNgayCong  2.1. @TongNgayCong = count(\*) from ChamCong join NhanVien  2.2.. Điều kiện:Month(Ngày công) = @Thang  và year(Ngày công) = @Nam  và Giờ vào not null  và Giờ ra not null  và Giờ ra - Giờ vào >= 3  và Mã nhân viên = @MaNV  2. Trả về @TongNgayCong |

1. Tính tổng số ngày công tăng ca của nhân viên nếu biết Mã nhân viên, Tháng, Năm (Ngày Tăng ca là ngày công đạt chuẩn và có Giờ ra lớn hơn 19h00)

| Mục đích | * Module phục vụ cho chức năng TinhSoNgayCongTangCa(), giúp theo dõi hiệu suất làm việc và quản lý việc tính lương hoặc thưởng cho giờ làm ngoài giờ quy định. |
| --- | --- |
| Input | * Mã nhân viên, Tháng, Năm |
| Output | * Tổng số ngày tăng ca |
| Process | * 1. Thực hiện đếm Tổng số ngày tăng ca của nhân viên: COUNT với điều kiện   MaNhanVien = @MaNhanVien và  Thang = @Thang và  Nam = @Nam và  Giờ ra - Giờ vào >= 3 và  GioRa > 19h   * 2. Trả về Tổng số ngày tăng ca của nhân viên |

1. Hãy thực hiện in ra danh sách Nhân viên bao gồm: số ngày công đạt chuẩn, tổng số ngày công tăng ca, tổng số ngày nghỉ nếu biết Tháng, Năm (Ngày công đạt chuẩn là Ngày công có tổng số Giờ công >= 3 giờ, Ngày công tăng ca là ngày công đạt chuẩn và có giờ ra lớn hơn 19h)

| Mục đích | Phục vụ cho việc in tổng số ngày công thường và ngày công chuẩn khi biết mã nhân viên |
| --- | --- |
| Input | Tháng, Năm |
| Output | Trả về danh sách Nhân viên bao gồm: Tổng số ngày công đạt chuẩn, Tổng số ngày công tăng ca và Tổng số ngày nghỉ của |
| Process | 1.Thực hiện đếm Tổng số ngày công đạt chuẩn của nhân viên, với điều kiện  Thang = @Thang và  Nam = @Nam và  Giờ ra - Giờ vào >= 3  2. Thực hiện đếm Tổng số ngày Công tăng ca của nhân viên, với điều kiện:  Thang = @Thang và  Nam = @Nam và  Giờ ra - Giờ vào >= 3 và  Giờ ra >= 19h  3. Thực hiện đếm Tổng số ngày nghỉ của nhân viên, với điều kiện:  Thang = @Thang và  Nam = @Nam và  LoaiNghi phải nằm trong (0,1) and LoaiNghi is null  4. Trả về Tổng ngày công đạt chuẩn, Tổng ngày công tăng ca và Tổng số Ngày nghỉ của nhân viên |

1. Tính lương tăng ca của nhân viên với công thức sau:

Lương tăng ca = 350.000 \* SoNgayTangCa

| Mục đích | Module phục vụ cho chức năng TinhLuongTangCa() của nhân viên và  đảm bảo dữ liệu về số ngày tăng ca và lương được cập nhật đầy đủ, chính xác theo từng tháng. |
| --- | --- |
| Input | Tháng, Năm, Mã nhân viên |
| Output | Lương tăng ca |
| Process | 1.. Tính @LuongTangCa = TongSoNgayTangCa(=select dbo.fnTinhTongNgayTangCa)\*350000  1.1. Điều kiện: MaNhanVien=@MaNV        và month(Ngày công)=@Thang        và  year(Ngày công)=@Nam  3. Trả về @LuongTangCa |

1. Thực hiện in ra các công việc chưa hoàn thành của nhân viên nếu biết mã nhân viên.
   1. Kiểm tra xem mã nhân viên có tồn tại trong bảng NhanVien không. Nếu mã nhân viên không tồn tại, in ra thông báo “Không tồn tại” và dừng thực hiện thủ tục.
   2. Nếu mã hợp lệ, Đếm số công việc chưa hoàn thành từ bảng CongViec. Nếu không có công việc nào, in thông báo “ Không có công việc nào ” và ngược lại nếu có công việc chưa hoàn thành in ra danh sách công việc chưa hoàn thành

| Mục đích | * Module phục vụ cho chức năng ThongBaoCongViec() và đảm bảo rằng nhân viên có thể nhận thông báo về tình trạng công việc của mình một cách kịp thời và chính xác. |
| --- | --- |
| Input | * Mã Nhân viên |
| Output | * Th1: Print “Không có công việc nào” * Th2: Print danh sách công việc chưa hoàn thành |
| Process | 1. Kiểm tra mã nhân viên:    1.1:  Không tồn tại => Print “Mã nhân viên không tồn tại” và dừng thủ tục    1.2: Tồn tại => Tiếp tục b2  2. Đếm số công việc chưa hoàn thành    2.1: Count(Công việc) =0 với điều kiện Trạng thái CV = 0 => Print “Nhân viên đã hoàn thành tất cả công việc”    2.2: Ngược lại => In ra  danh sách công việc chưa hoàn thành |

1. Tính tổng số ngày nghỉ của nhân viên khi biết mã nhân viên, tháng, năm.

| Mục đích | * Module phục vụ cho chức năng TinhSoNgayNghi() và đảm bảo tính toán chính xác tổng số ngày nghỉ của mỗi nhân viên dựa trên mã nhân viên từ bảng NghiPhep |
| --- | --- |
| Input | * Mã nhân viên, Tháng, Năm |
| Output | * TongSoNgayNghi |
| Process | 1. Lấy @TongSoNgayNghi = count(NgayNghi) của bảng NghiPhep với điều kiện MaNhanVien= @MaNhanVien và gán giá trị mặc định là 0 nếu không có ngày nghỉ nào  2. Trả về @TongSoNgayNghi |

1. Tính tiền khấu trừ của nhân viên trong 1 tháng bất kỳ nếu biết Mã nhân viên, Tháng, Năm

| Mục đích | * Module này phục vụ cho chức năng TinhKhauTru() và đảm bảo tính toán đầy đủ, chính xác tổng tiền khấu trừ (tổng tiền phạt) của từng nhân viên trong một tháng của năm cụ thể |
| --- | --- |
| Input | * Mã nhân viên, Tháng, Năm |
| Output | * Tiền Khấu trừ của nhân viên |
| Process | 1.Lấy @KhauTru= SUM(TienPhat) từ bảng lương và lỗi phạt với điều kiện (Luong.Thang = @Thang AND Luong.Nam = @Nam, MaNhanVien=@MaNhanVien)  2. Nếu @KhauTru is null trả về kết quả 0  3. Ngược lại, trả về @KhauTru |

1. Thực hiện tính lương thực nhận cho nhân viên với công thức sau:

Lương thực nhận = (LuongCoBan \* HeSoLuong) \* (TongNgayCong / (SoNgayTrongThang - 1)) + LươngTangCa+ PhuCap - KhauTru

| Mục đích | Phục vụ cho việc tính lương thực nhận của nhân viên trong tháng bất kì |
| --- | --- |
| Input | Tháng, năm, Mã nhân viên |
| Output | Lương thực nhận |
| Process | 1. Lấy hệ số lương, phụ cấp từ bảng ChucVu dựa vào MaCV  2. Lấy lương cơ bản từ bảng NhanVien dựa vào MaNV  3. Tính lương thực nhận = (LuongCoBan \* HeSoLuong) \* (TongNgayCong / (SoNgayTrongThang - 1)) + LươngTangCa+ PhuCap + Thuong - KhauTru . với điều kiện tháng = @thang, năm=@nam, mã nhân viên=@maNV   * trong đó: Tổng ngày công được gọi từ hàm tính tổng ngày công * Số ngày công trong tháng là số ngày có trong tháng * Khấu trừ được gọi từ hàm tính khấu trừ * Lương tăng ca được gọi từ hàm tính lương tăng ca |

1. Trả về thông tin Lương của nhân viên nếu biết mã nhân viên và tháng, năm. Thông tin lương bao gồm Lương cơ bản, Hệ số lương, phụ cấp, Thưởng, Lương tăng ca, Lương thực nhận khi biết Mã Nhân viên.

| Mục đích | * Module phục vụ cho chức năng Truy vấn thông tin lương dựa trên Mã Nhân viên, đảm bảo tính chính xác về dữ liệu lương. |
| --- | --- |
| Input | * Mã nhân viên * Thời gian |
| Output | * Lương cơ bản * Hệ số lương * Phụ cấp * Thưởng * Lương tăng ca * Lương thực nhận |
| Process | 1. Khai báo các biến đầu vào: Mã nhân viên, Tháng, Năm—> @MaNhanVien, @Thang, @Nam  2. Lấy Lương cơ bản, Hệ số lương, Phụ cấp, Thưởng, Lương tăng ca, Lương thực nhận, Khấu trừ, Tổng ngày công—>@LuongCoBan, @Hesoluong, @PhuCap, @Thuong, @LuongTangca, @LuongThucNhan, @KhauTru, @TongNgayCong  3. Tính @LuongTangCa = 350 000\*count(MaCong)  3.1. Điều kiện: MaNhanVien=@MaNV        và month(Ngày công)=@Thang        và  year(Ngày công)=@Nam  4. Tính @KhauTru = tổng tiền phạt  4.1. Điều kiện: MaNhanVien=@MaNV        và tháng từ bảng lương=@Thang        và  năm từ bảng lương=@Nam  5. Tính @TongNgayCong= count(Mã công)  5.1. Điều kiện: MaNhanVien=@MaNV        và month(Ngày công)=@Thang        và  year(Ngày công)=@Nam  6. Tính  (LuongCoBan \* HeSoLuong) \* (TongNgayCong / (SoNgayTrongThang - 1)) + LươngTangCa+ PhuCap + Thuong-KhauTru  6.1. Điều kiện: Tháng từ bảng Lương=@Thang  Năm từ bảng Lương = @Nam  7. Đưa ra Lương cơ bản, Tổng Ngày công, Hệ số lương, Phụ cấp, Thưởng, Lương tăng ca, Lương thực nhận |

1. Hãy trả về thông tin chấm công bao gồm: tổng ngày công thường, tổng ngày công tăng ca, số ngày nghỉ của nhân viên trong một tháng .

| Mục đích | Module này phục vụ cho chức năng đọc (Read) thông tin chấm công bao gồm (tổng ngày công thường, tổng ngày công tăng ca, số ngày nghỉ trong một tháng) |
| --- | --- |
| Input | Không có |
| Output | Thông tin chấm công bao gồm (tổng ngày công thường, tổng ngày công tăng ca, số ngày nghỉ trong một tháng) |
| Process | 1.Lấy ra mã nhân viên, tháng chấm công, năm chấm công, Tổng ngày công thường (Điều kiện: GioRa <= '19:00'), Tổng ngày công tăng ca( Điều kiện: GioRa > '19:00'),  Tổng số ngày nghỉ, tháng nghỉ, năm nghỉ từ bảng nhân viên, chấm công, nghỉ phép  2.In ra Thông tin chấm công |

# CƠ CHẾ BẢO MẬT:

Đối với hệ thống quản lý nhân sự cho doanh nghiệp vừa và nhỏ, để đảm bảo tính bảo mật của hệ thống và không làm ảnh hưởng đến nghiệp vụ công việc của công ty. Nhóm chọn 2 phương án sau: Quản lý người dùng, thực hiện phân quyền cho từng người dùng cơ sở dữ liệu; mã hóa những dữ liệu quan trọng có tính cá nhân cao để tránh bị rò rĩ thông tin ra ngoài.

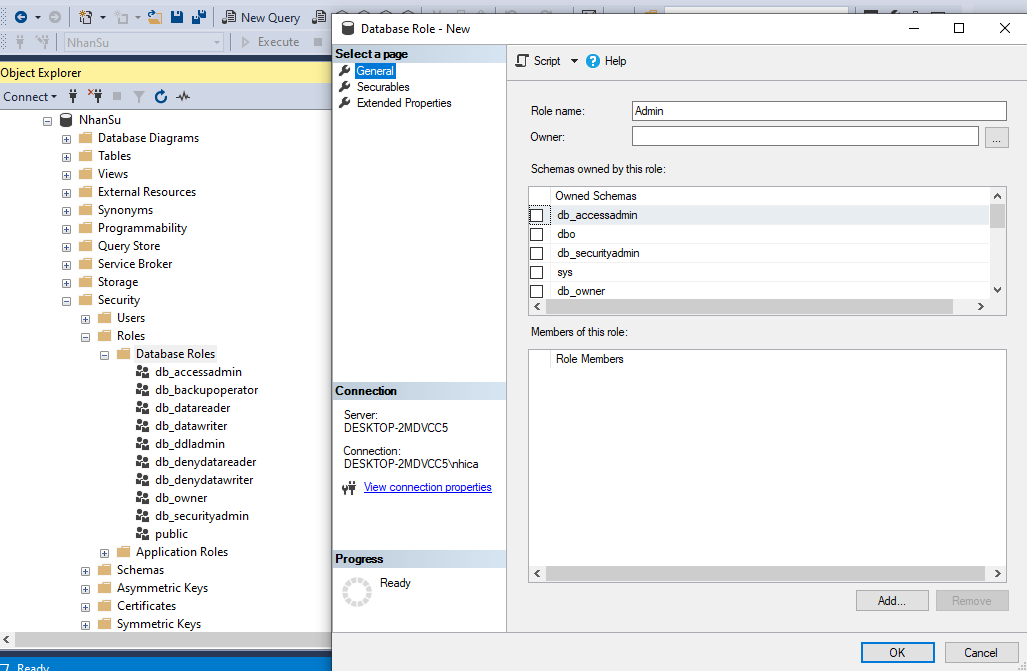
## Quản lý người dùng (phân quyền):

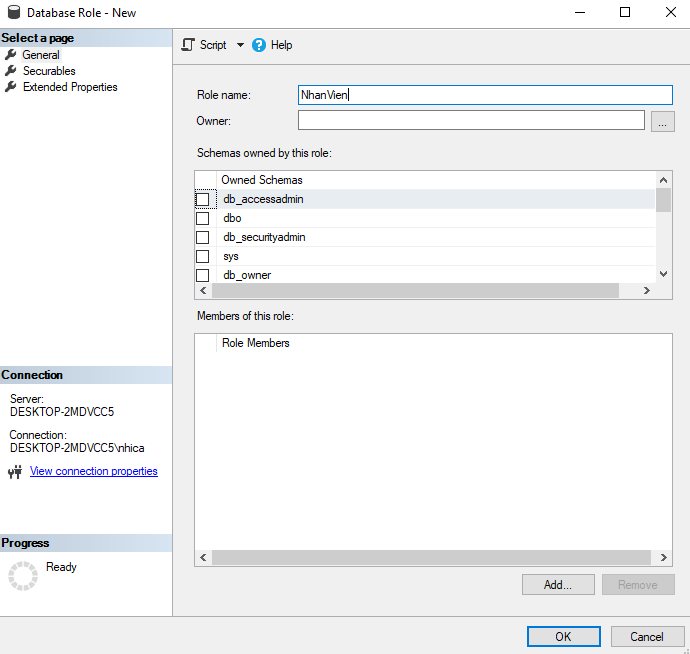
| Role | Account | Quyền DB | Quyền bảng |
| --- | --- | --- | --- |
| Admin | GiamDoc | Db\_owner | Tất cả |
| KeToan | KeToan1 | db\_datawriter,  db\_reader | select, insert, update trên các bảng:  NhanVien, LoiPhat, NghiPhep,Luong, ChamCong, ChucVu |
| NhanVien | NhanVien1 | db\_reader | select trên các bảng:  NhanVien,  DanhGiaNangLuc, CongViec,  CongViec\_BaoCao,  ChamCong, NghiPhep,  LoiPhat,  KyNang |

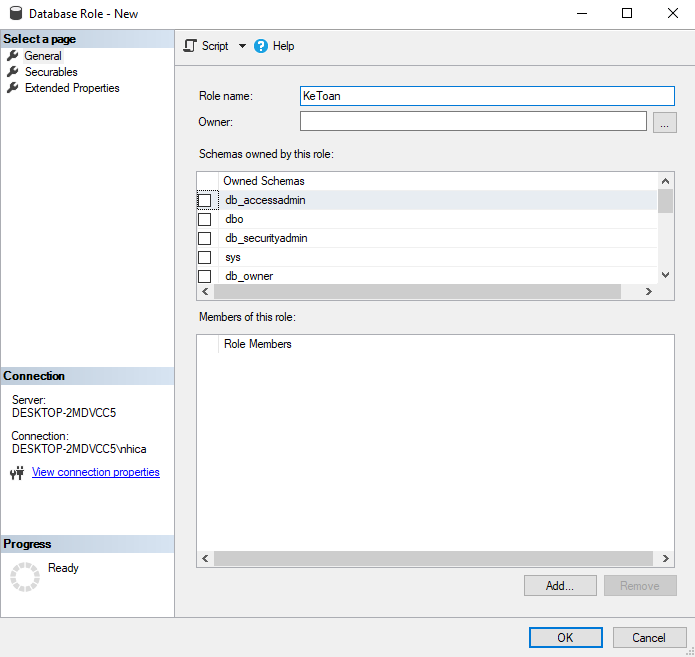
1. **Tạo Role:**

* Tạo Role Admin, KeToan, NhanVien:

NhanSu > Security >Roles>  Nhấn chuột phải vào Database Roles> New Database Role

* Admin

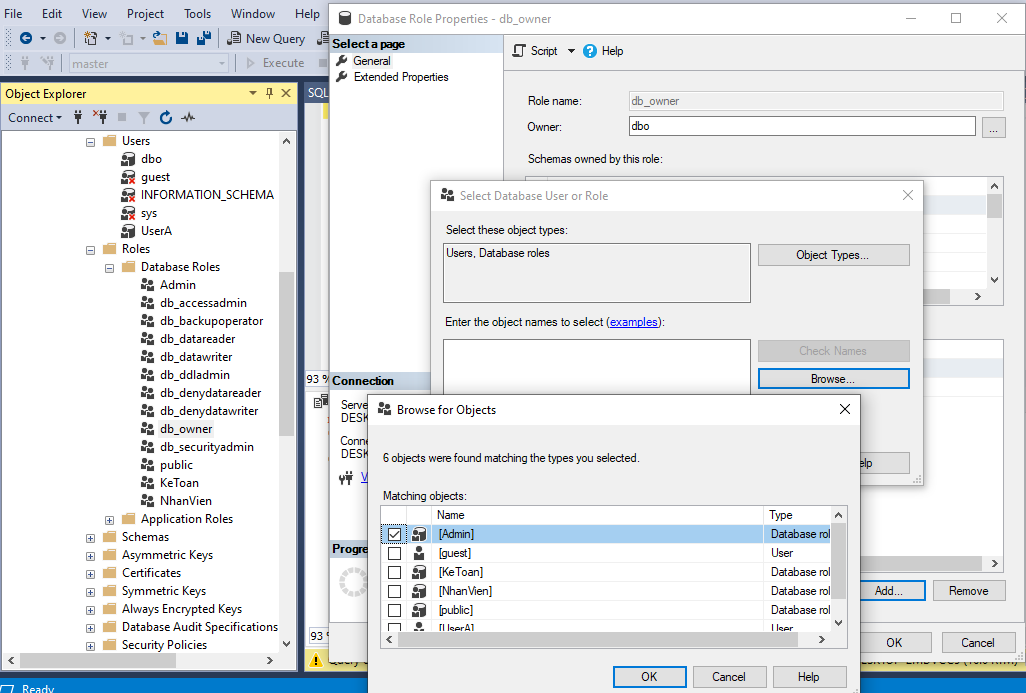
* KeToan

* NhanVien

1. **Quyền Database**

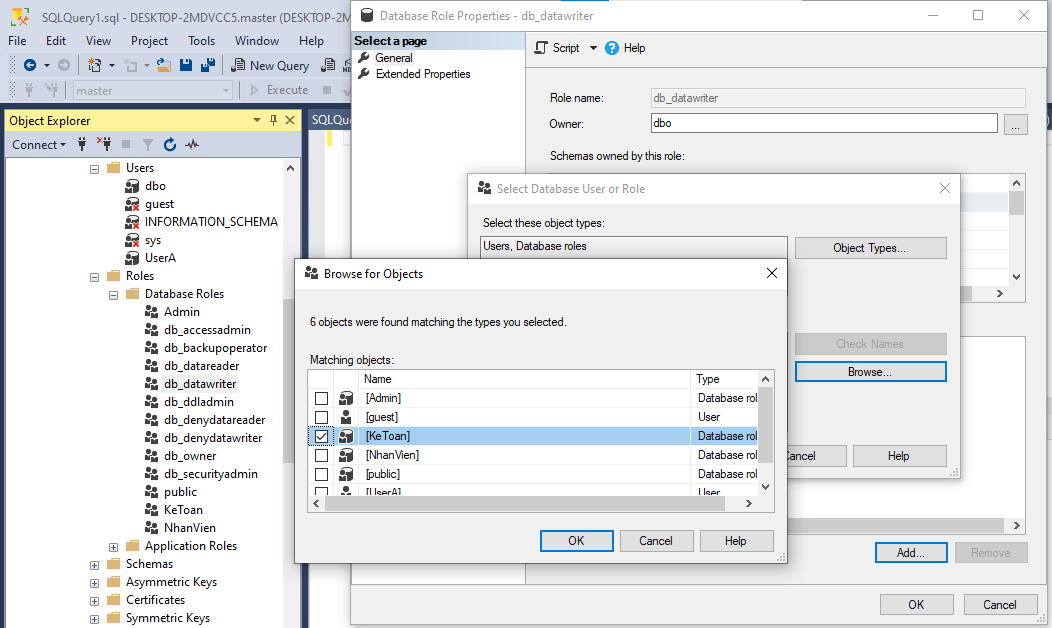
* **Thêm quyền db\_owner cho Role Admin:**

NhanSu > Security > Nhấn chuột phải vào Database Roles> db\_owner> Nhấn chuột phải> chọn Properties> chọn Add> Browser> tích chọn Admin> OK



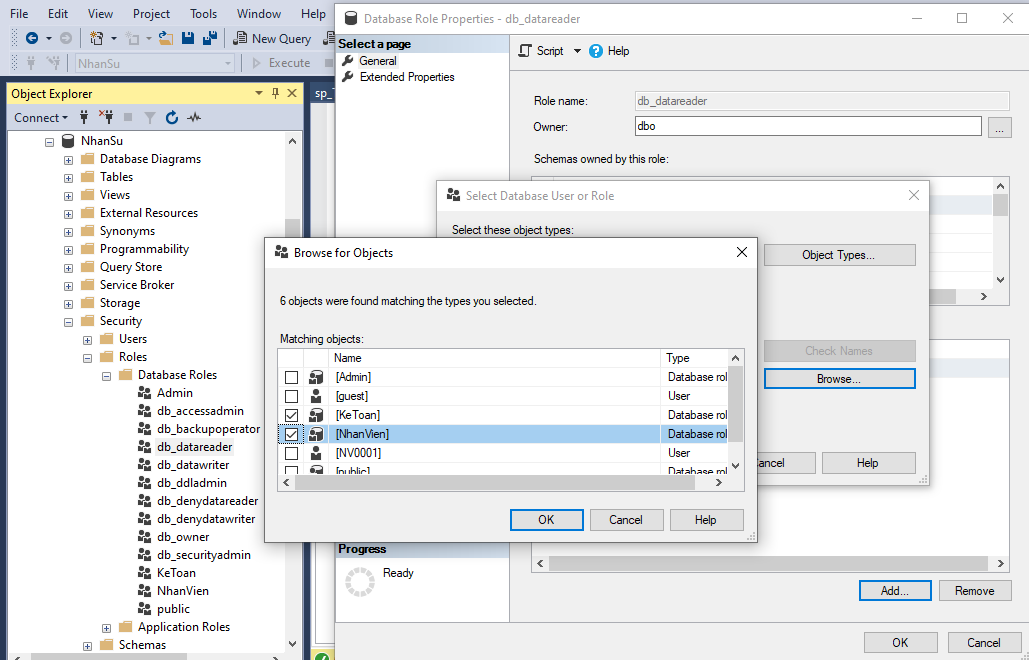
* **Thêm quyền db\_datawriter cho Role KeToan:**

NhanSu > Security > Nhấn chuột phải vào Database Roles> db\_datawriter > Nhấn chuột phải> chọn Properties> chọn Add> Browser> tích chọn KeToan> OK



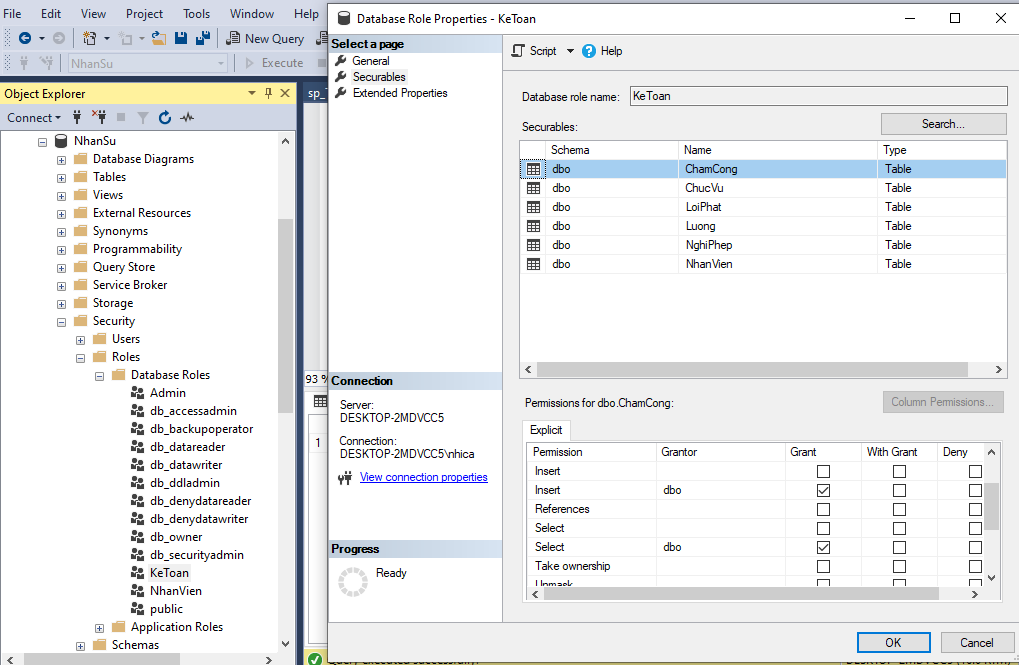
* Thêm quyền db\_datareader cho Role NhanVien  và KeToan:

NhanSu > Security > Nhấn chuột phải vào Database Roles> db\_datawriter > Nhấn chuột phải> chọn Properties> chọn Add> Browser> tích chọn NhanVien và KeToan> OK

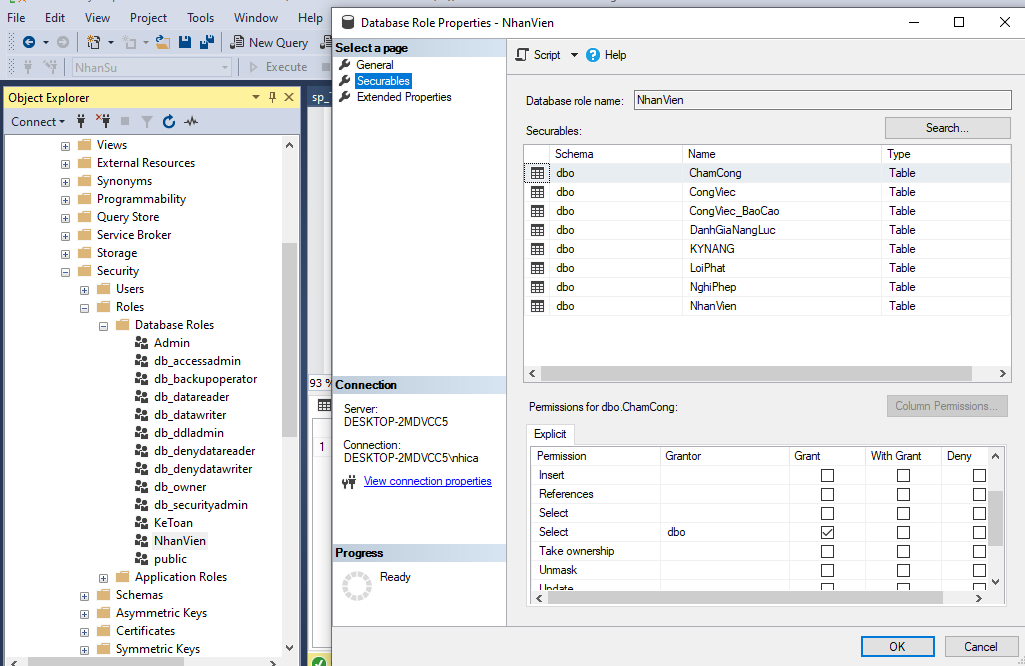


1. **Quyền bảng**

* Cho phép KeToan có quyền select, insert, update trên các bảng: NhanVien, ChucVu, Luong, NghiPhep, LoiPhat, ChamCong

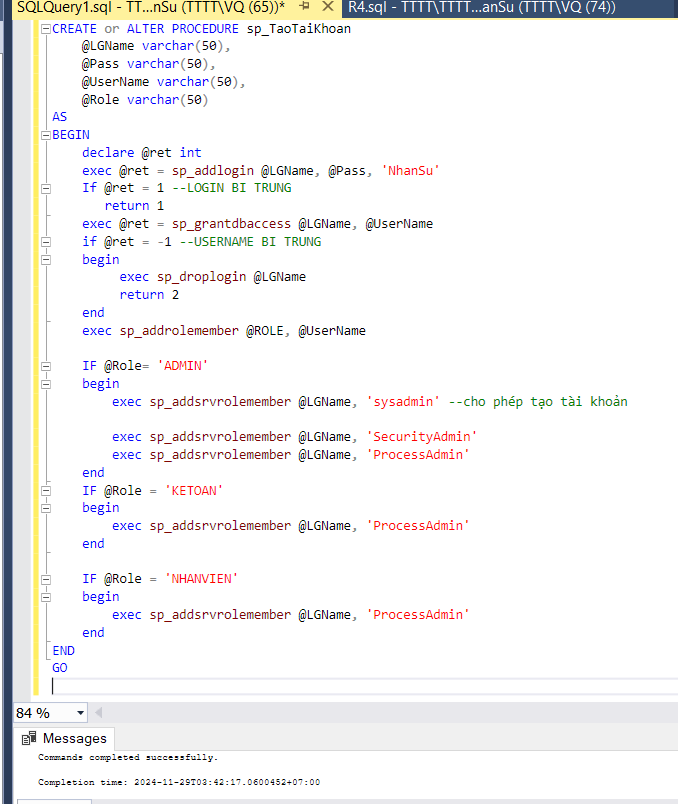


* Cho phép NhanVien có quyền select trên các bảng : NhanVien, DanhGiaNangLuc, CongViec, CongViecBaoCao, ChamCong, NghiPhep, LoiPhat, KyNang



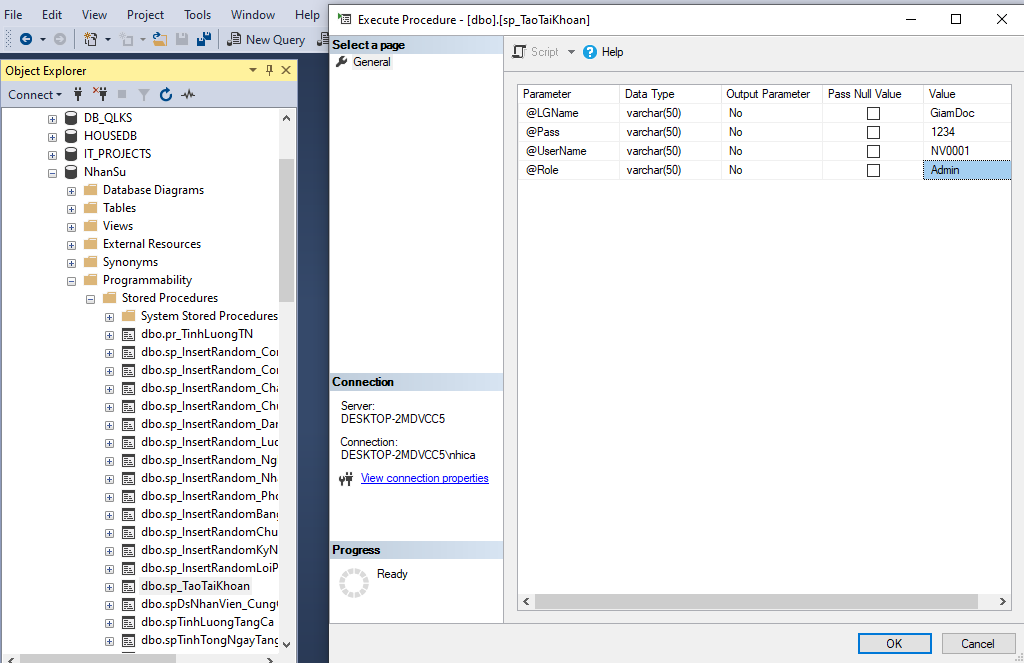
1. Account

* Tạo thủ tục tạo tài khoản

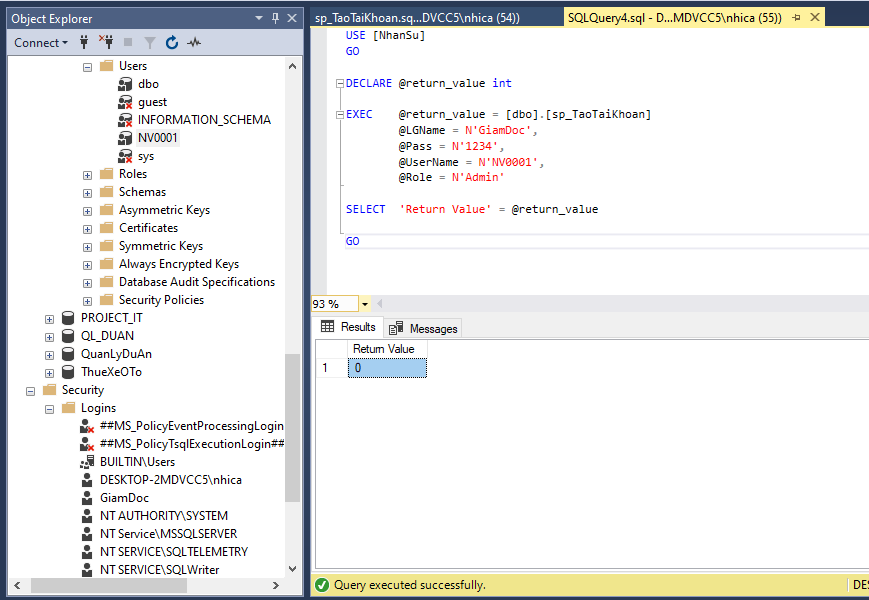


* Tạo tài khoản cho người dùng với role =Admin

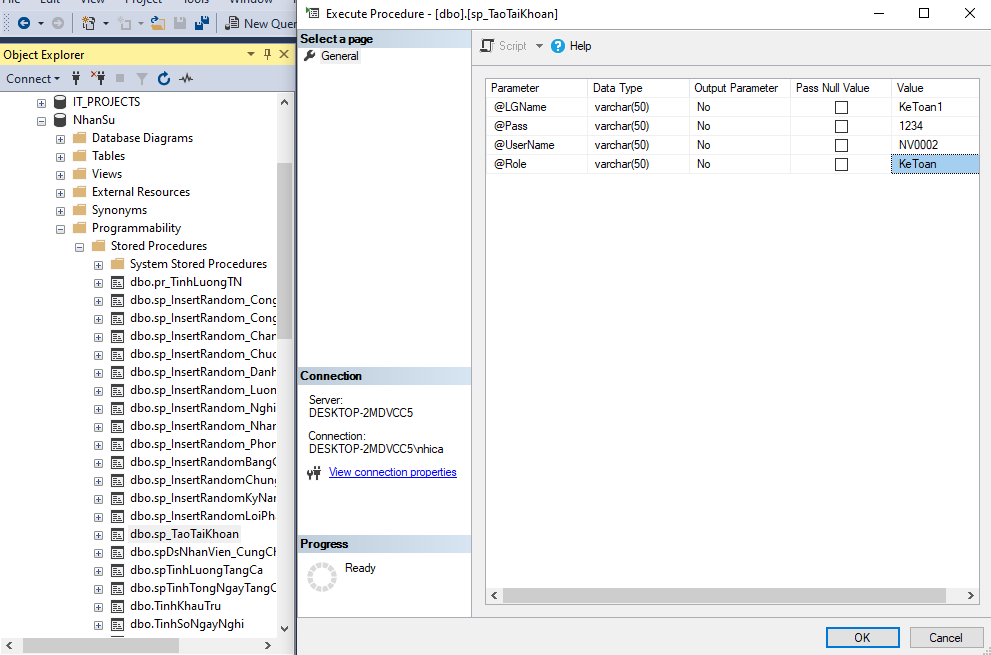
NhanSu> Programmability>Stored Procedures> dbo.sp\_TaoTaiKhoan> Nhấn chuột phải Execute Stored Procedures > Điền thông tin> Nhấn OK



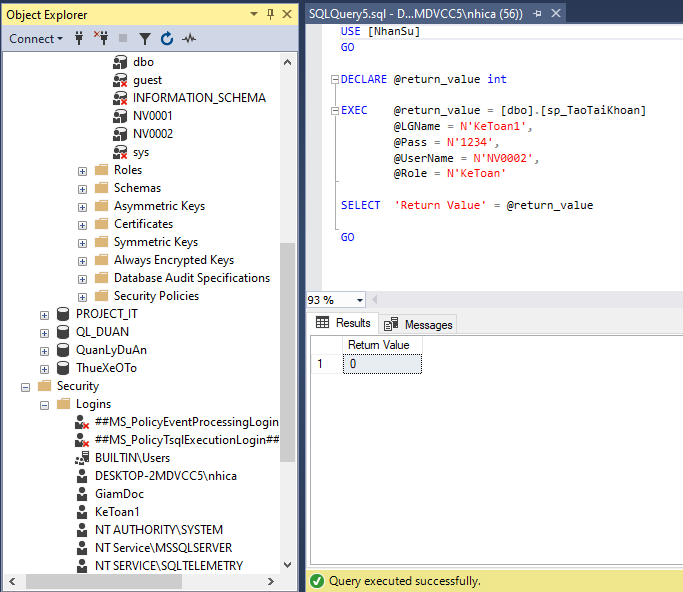
* Thủ tục được thực thi và hiển thị kết quả như bên dưới:



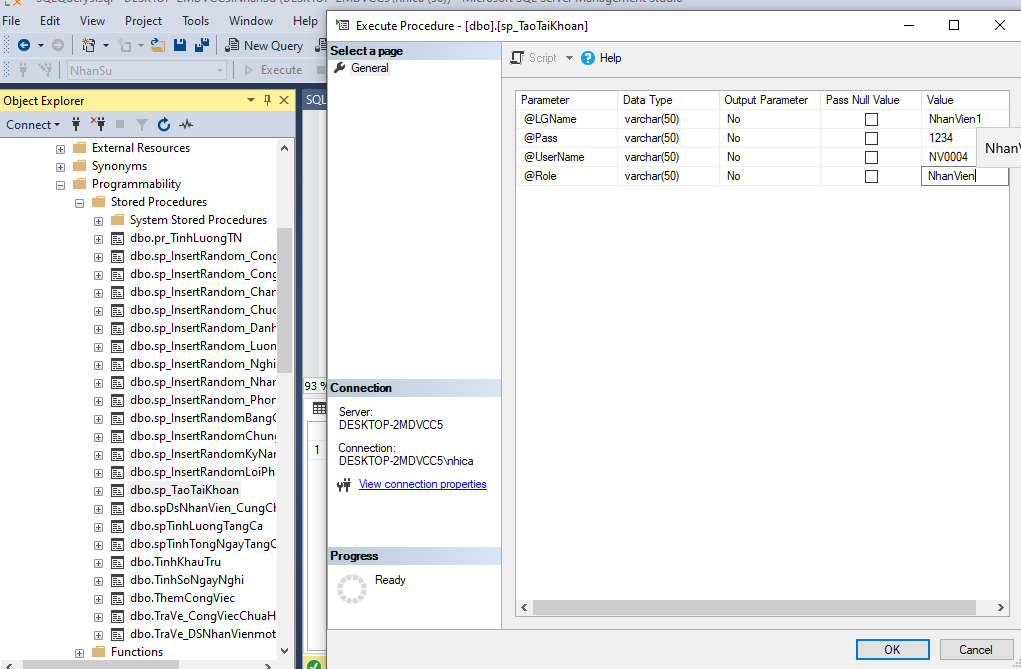
* Tạo tài khoản cho người dùng với Role = KeToan



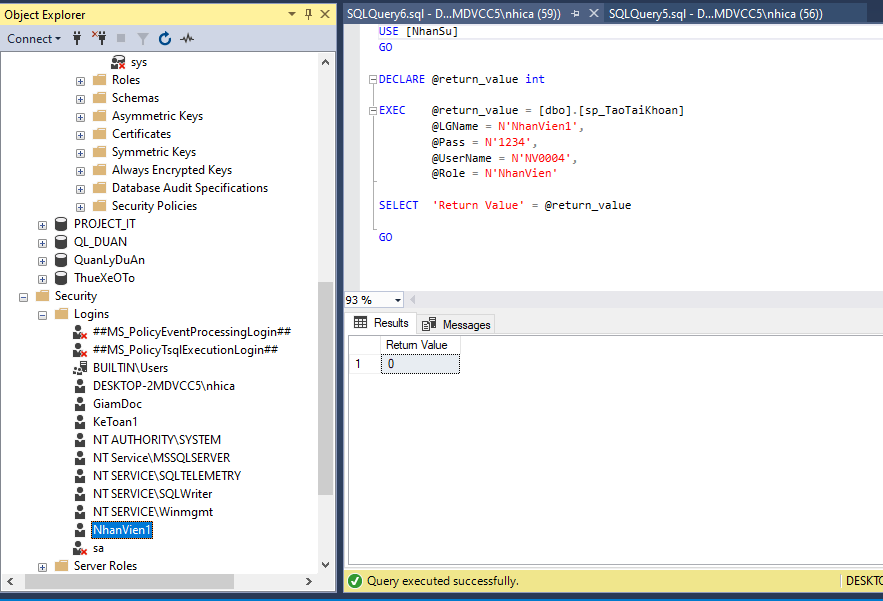
* Thủ tục được thực thi và hiển thị kết quả như bên dưới:



* Tạo người dùng với Role = NhanVien



Thủ tục được thực thi và hiển thị kết quả như bên dưới:

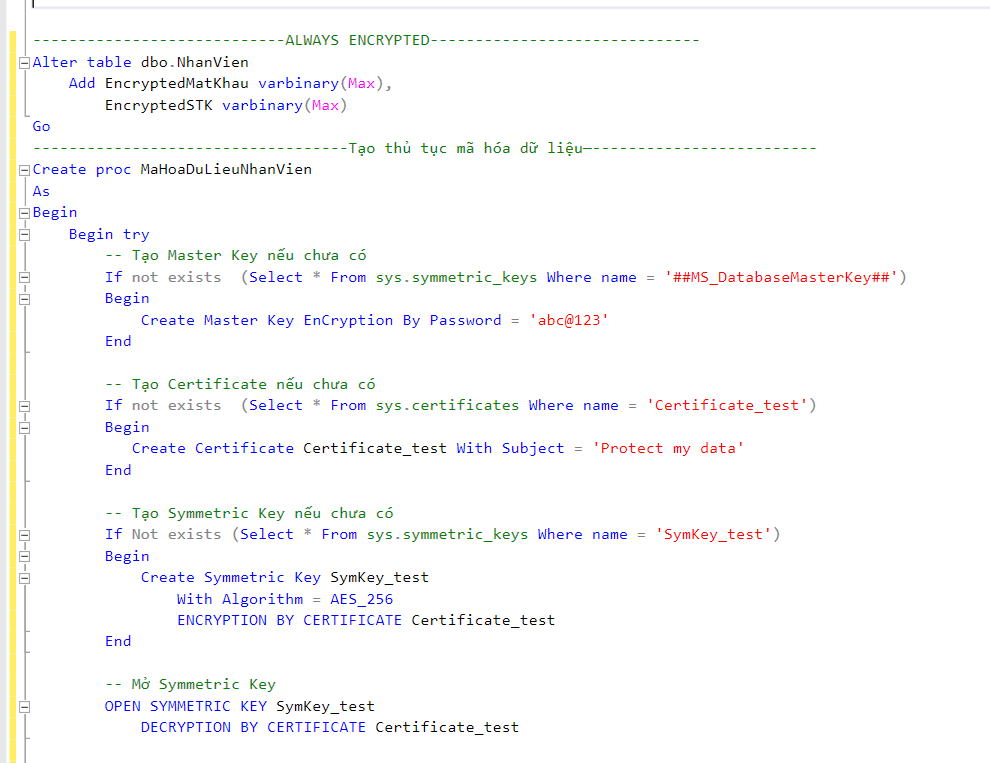


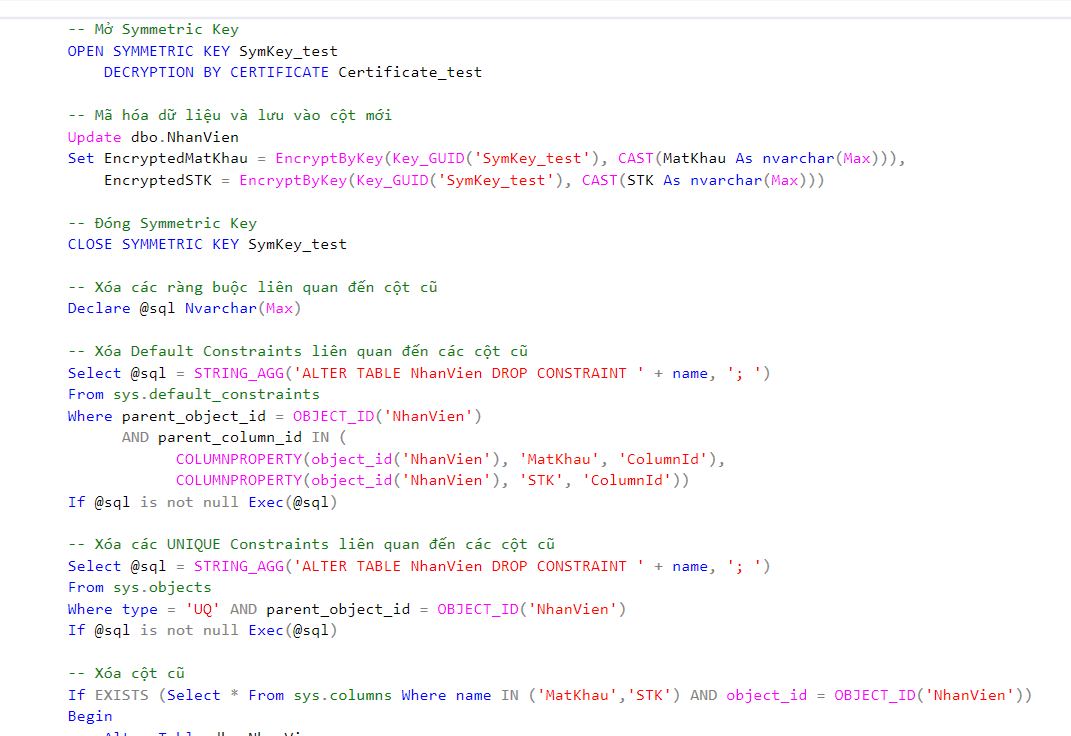
Sau khi hoàn thành tất cả các bước trên, nhóm đã thực hiện cài đặt quản lý người dùng thành công với các tài khoản Admin, kế toán và nhân viên theo kịch bản.

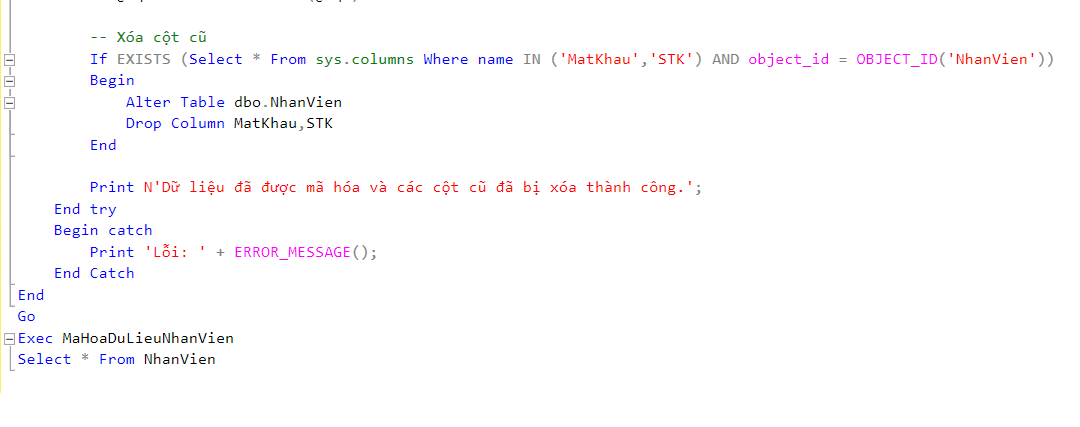
## Always Encrypted (Mã hóa dữ liệu):

Nhóm quyết định lựa chọn mã hóa các cột *MatKhau* và *STK* trong bảng *NhanVien*, vì đây là những cột chứa thông tin nhạy cảm, dễ trở thành mục tiêu của các cuộc tấn công, đặc biệt là tấn công SQL Injection. Việc bảo vệ các cột này không chỉ giúp tăng cường bảo mật cho dữ liệu mà còn đảm bảo tính riêng tư và an toàn cho người dùng, đồng thời giảm thiểu rủi ro bị lộ thông tin quan trọng nếu hệ thống bị xâm nhập.

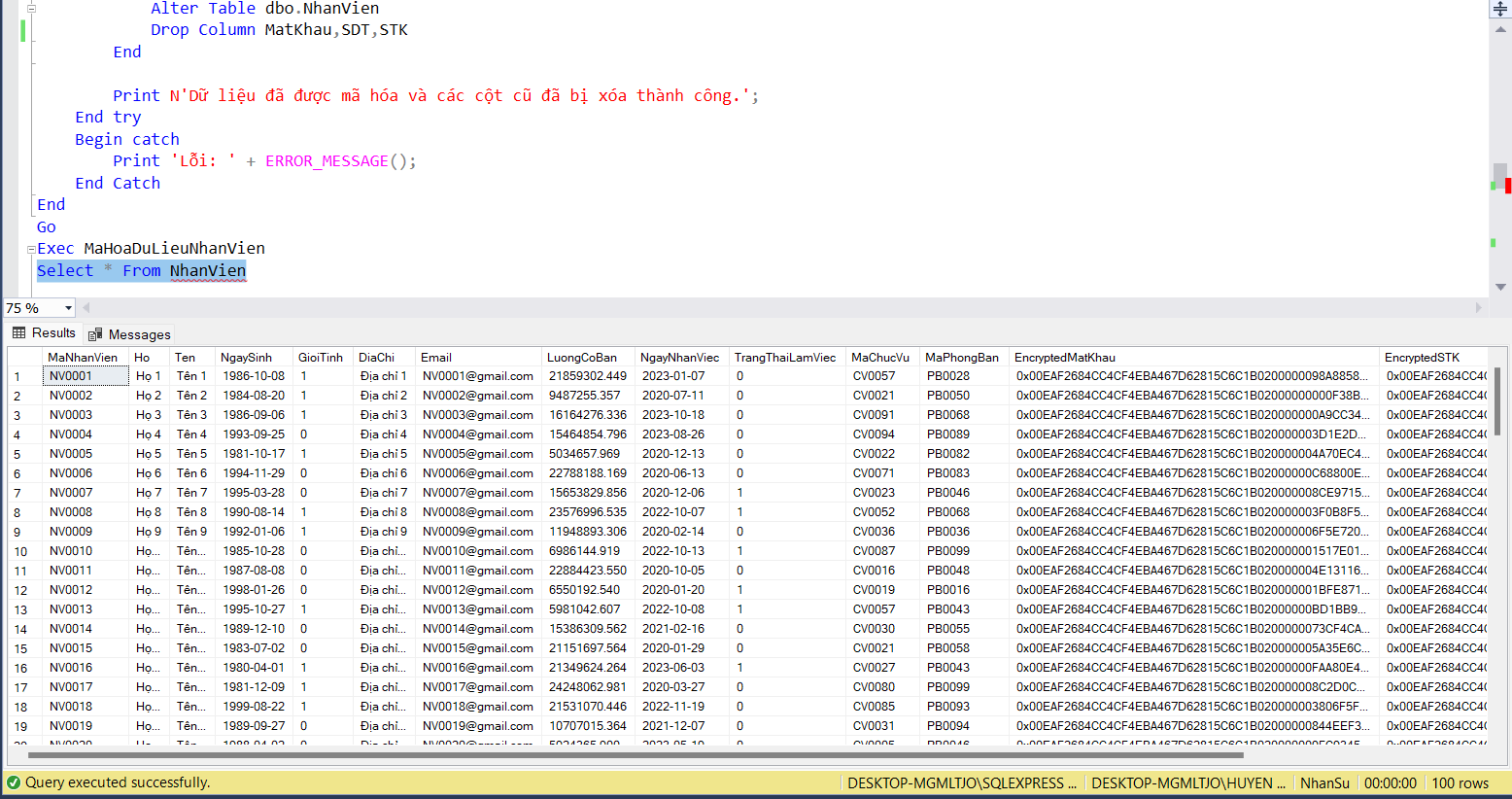
* 1. Mã hóa dữ liệu
* Dưới đây là đoạn code thực hiện mã hóa các cột mật khẩu, số điện thoại, địa chỉ, email và số tài khoản trong bảng Nhân viên



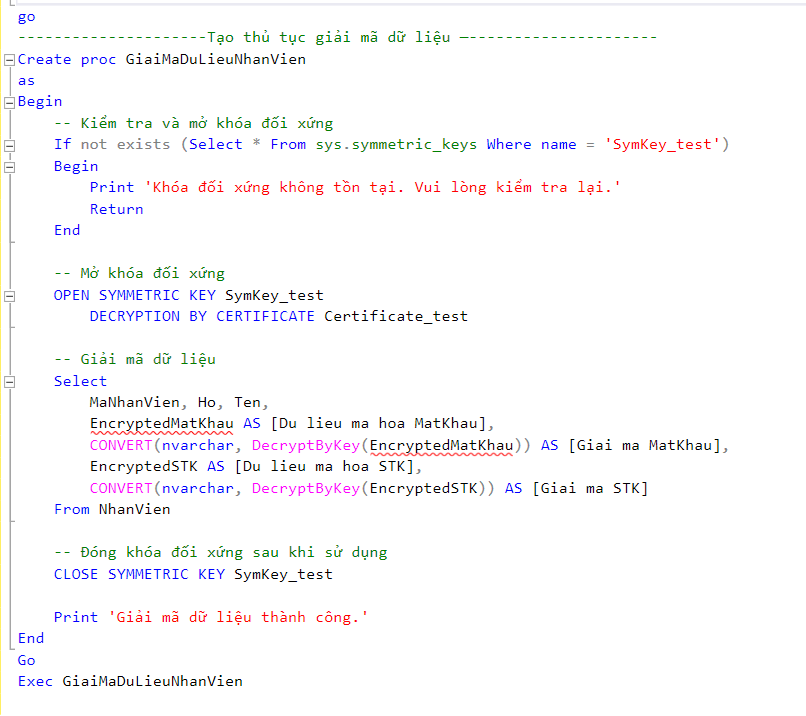




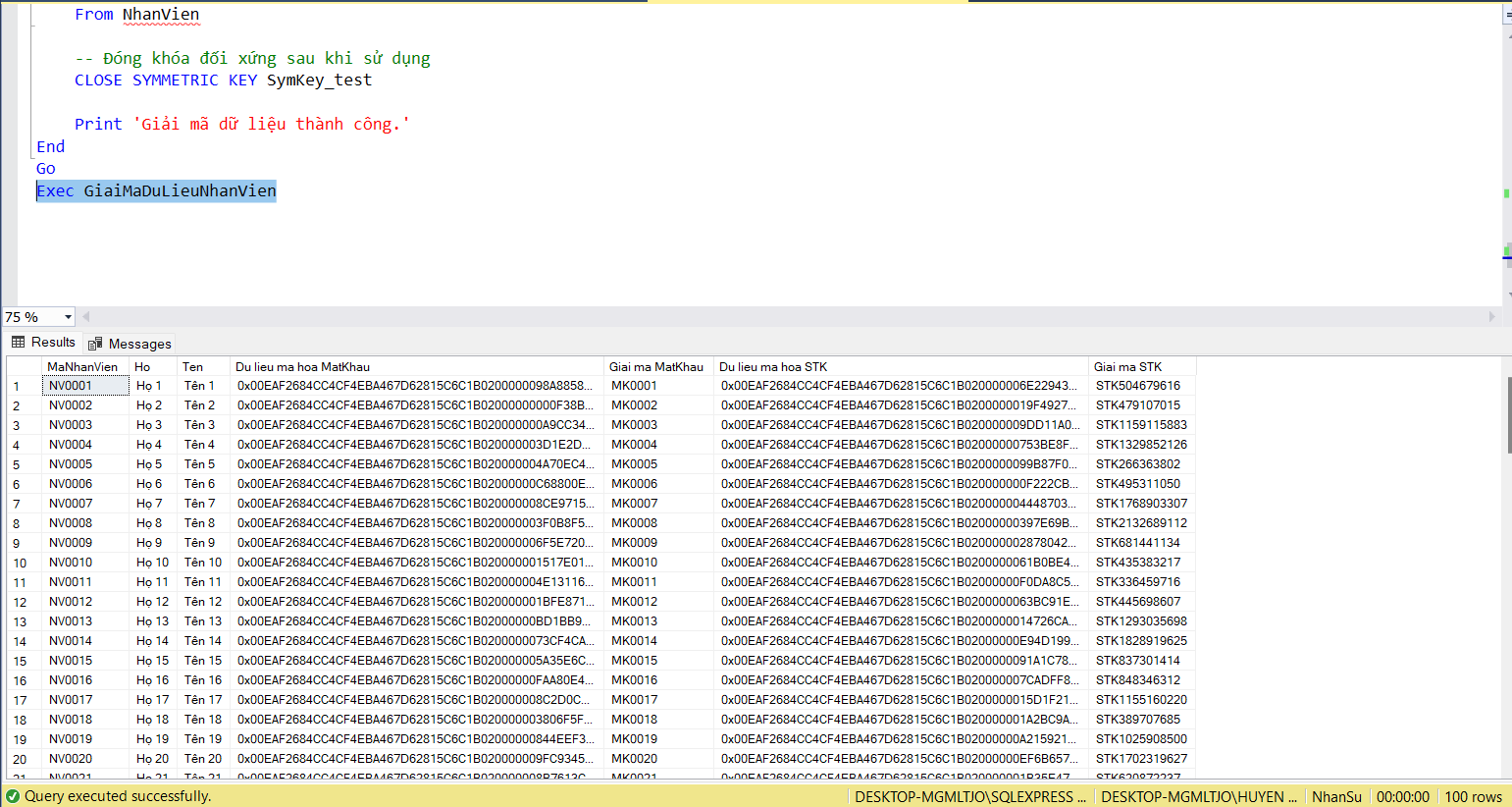
* Kết quả chạy được đoạn code trên như sau:



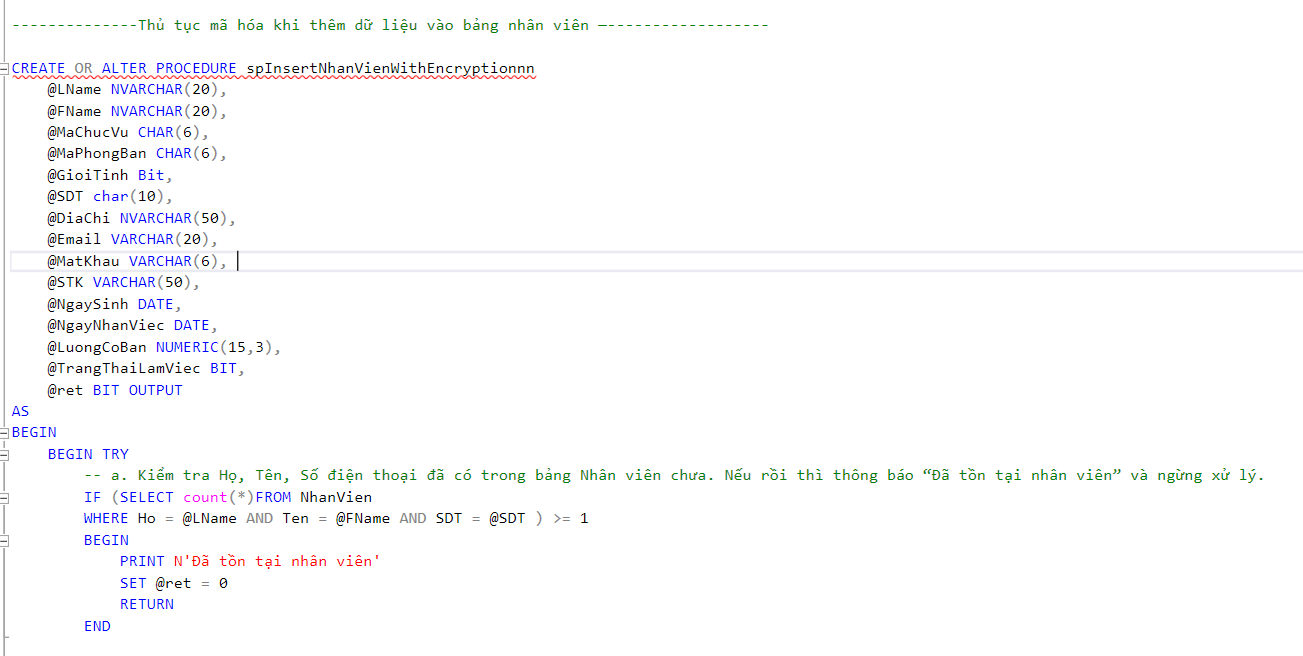
* 1. Giải mã dữ liệu:
* Dưới đây là đoạn code thực hiện giải mã các cột mật khẩu, số tài khoản trong bảng Nhân viên

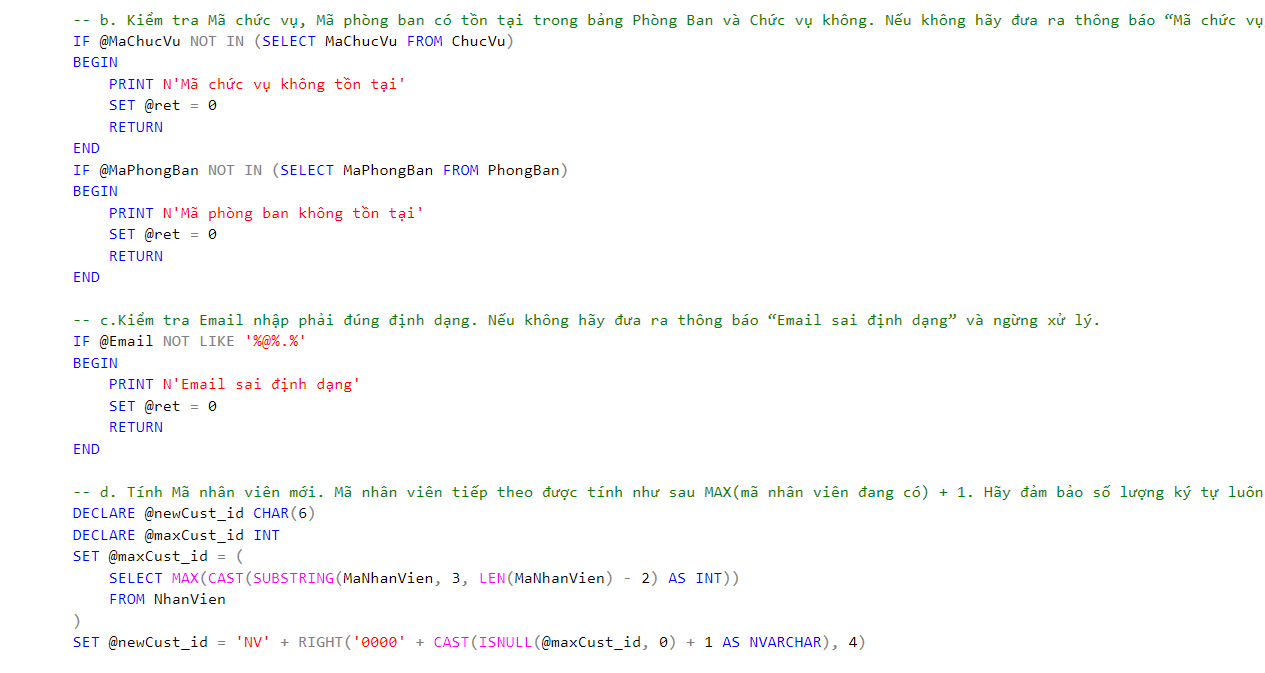


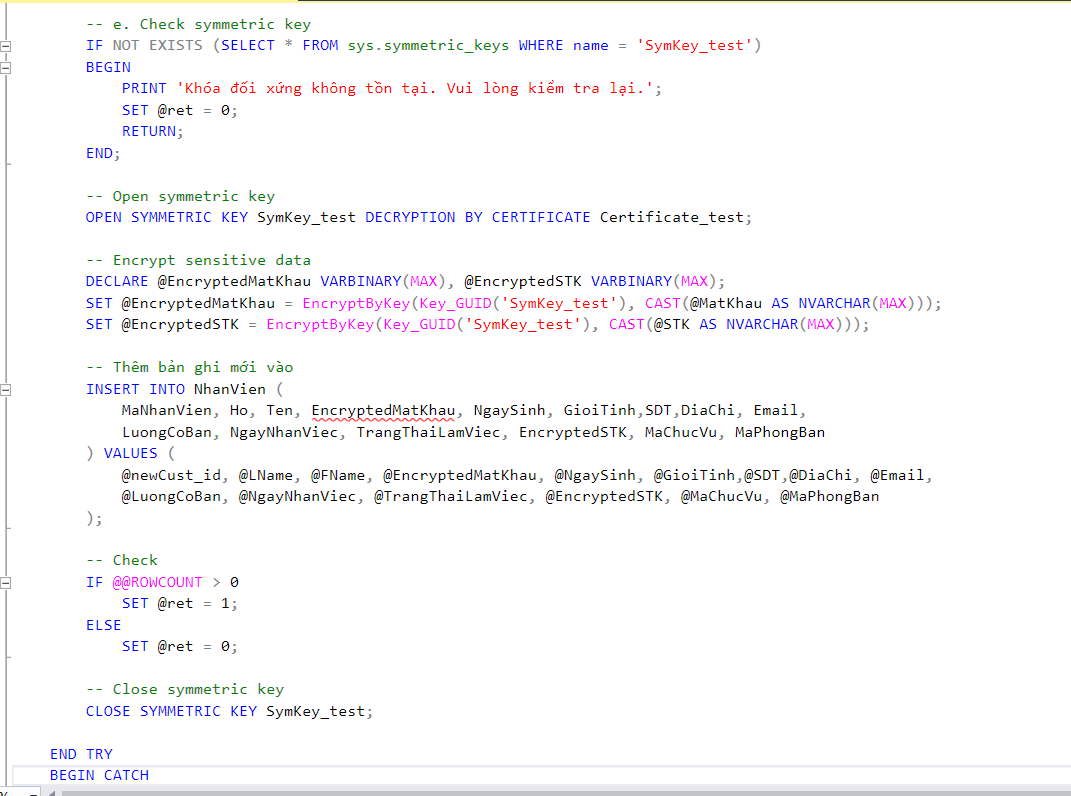
* Kết quả chạy được đoạn code trên như sau:



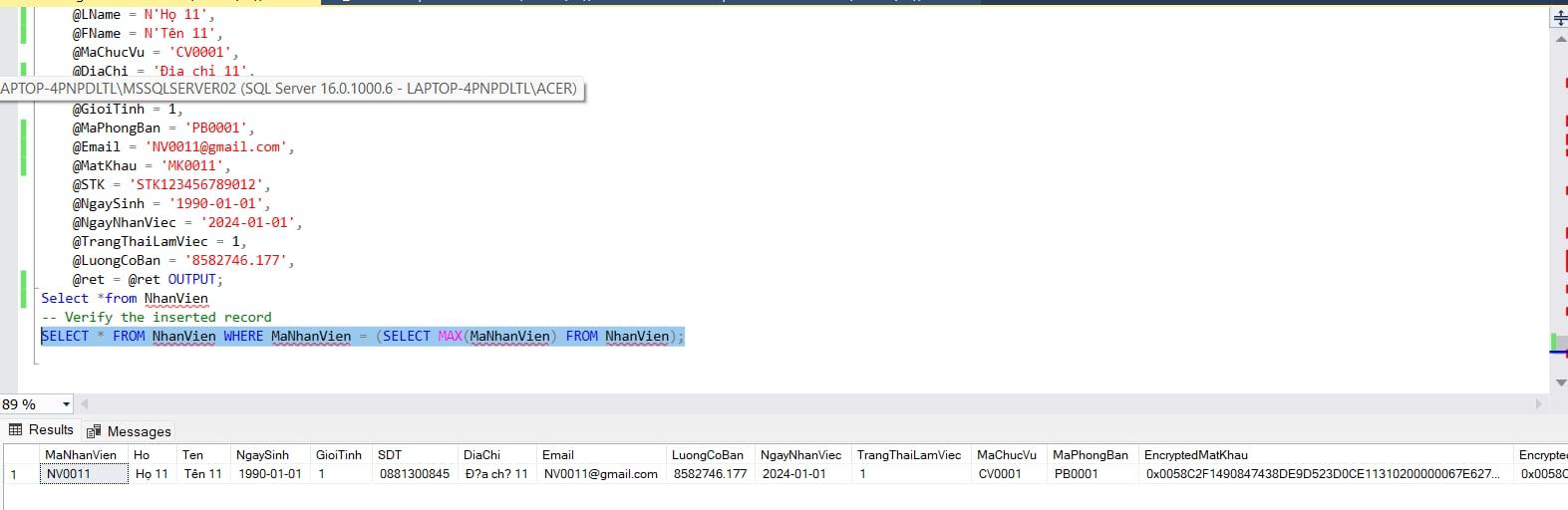
* 1. Thủ tục INSERT vào bảng nhân viên và mã hóa dữ liệu:
* Dưới đây là thủ tục mã hóa dữ liệu đầu vào là mật khẩu và số tài khoản thì người dùng thực hiện nhập mới dữ liệu vào bảng Nhân viên:

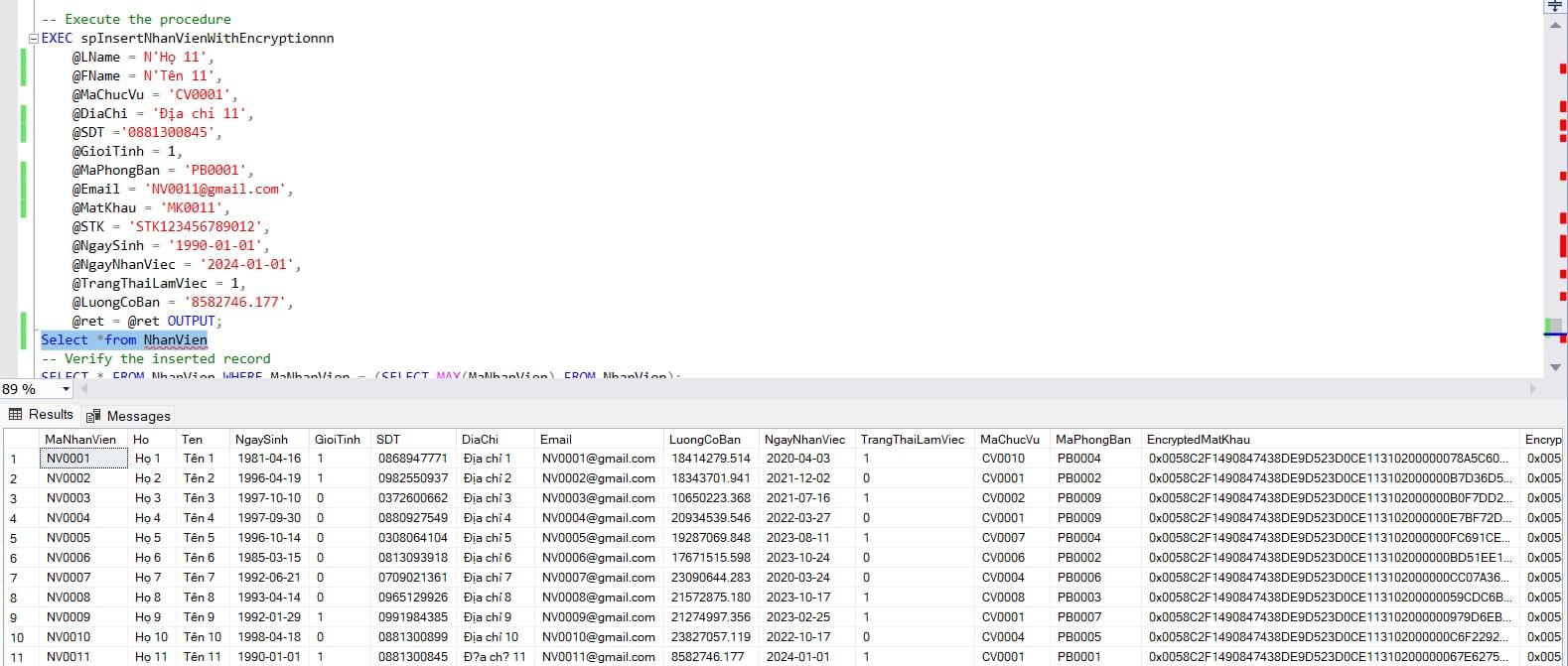




* Kết quả chạy như sau:





# PHÂN TÁN DỮ LIỆU:

## Lý do lựa chọn giải pháp phân tán dữ liệu:

* Không tốn chi phí cao.
* Tăng tính an toàn dữ liệu khi phân tán ra nhiều máy khác nhau.
* Giảm dung lượng lưu trữ trên ổ đĩa máy chủ.
* Vẫn đảm bảo được tính đồng nhất của dữ liệu.
* Dễ dàng mở rộng hệ thống, có thể thêm ổ đĩa hoặc máy chủ mới vào hệ thống.
* Cho phép khôi phục dữ liệu về một thời điểm trước đó một cách nhanh chóng mà không phải trải qua quá trình phục hồi từ bản sao lưu đầy đủ. Điều này đặc biệt hữu ích khi xảy ra lỗi hệ thống hoặc dữ liệu bị thay đổi ngoài ý muốn.

## Kịch bản phân tán:

Kịch bản tạo bảng phân tán chiều dọc với publications là LƯƠNG và CÔNG VIỆC.

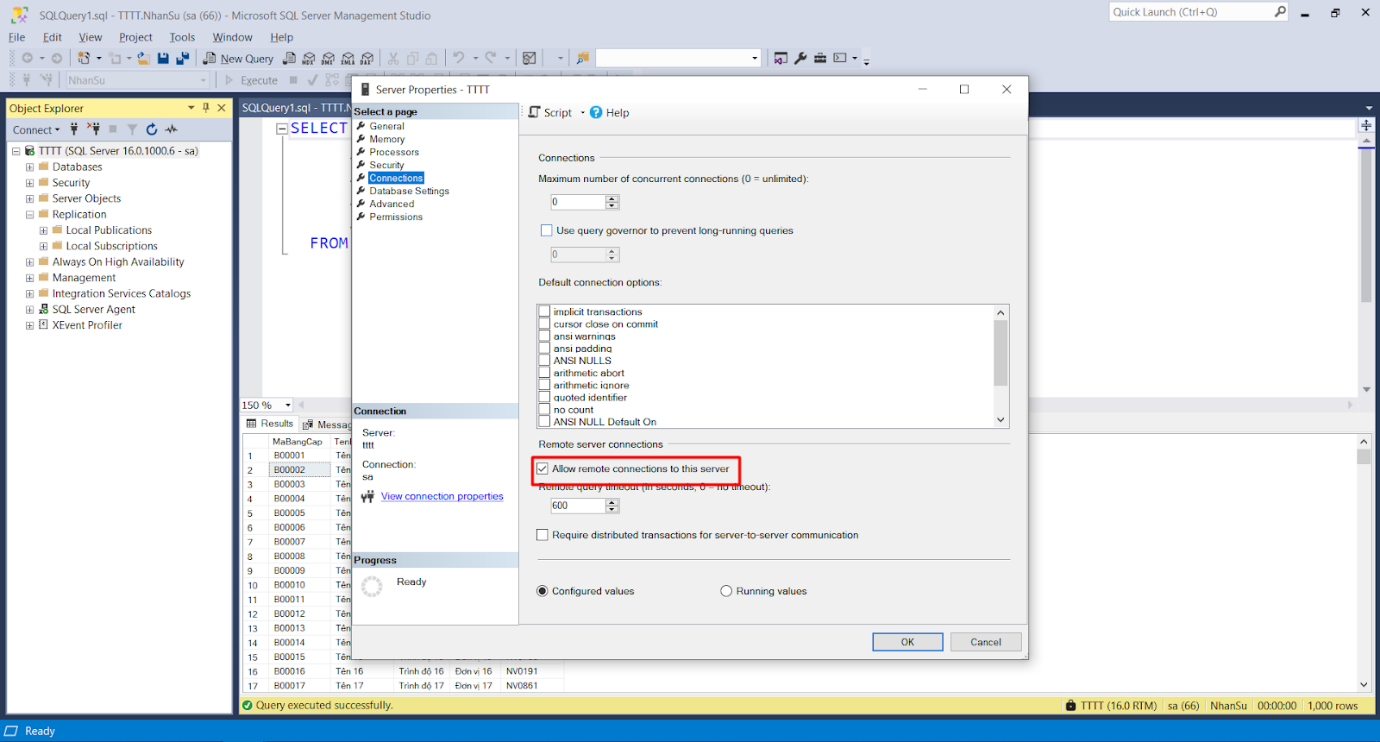
| Publications | Subcriber | Bảng |
| --- | --- | --- |
| dis\_Luong | 172.26.62.172 | ChamCong, ChucVu,  NghiPhep,  NhanVien,  Luong |
| dis\_CongViec | 172.26.62.184 | BangCap,  ChucVu, PhongBan,  NhanVien,  CongViec  CongViec\_Baocao  KyNang,  ChungChi |

## DEMO:

PHẦN 1: THIẾT LẬP MÔI TRƯỜNG VÀ THỰC HIỆN KẾT NỐI SERVER CHÍNH VÀ 2 SERVER CON:

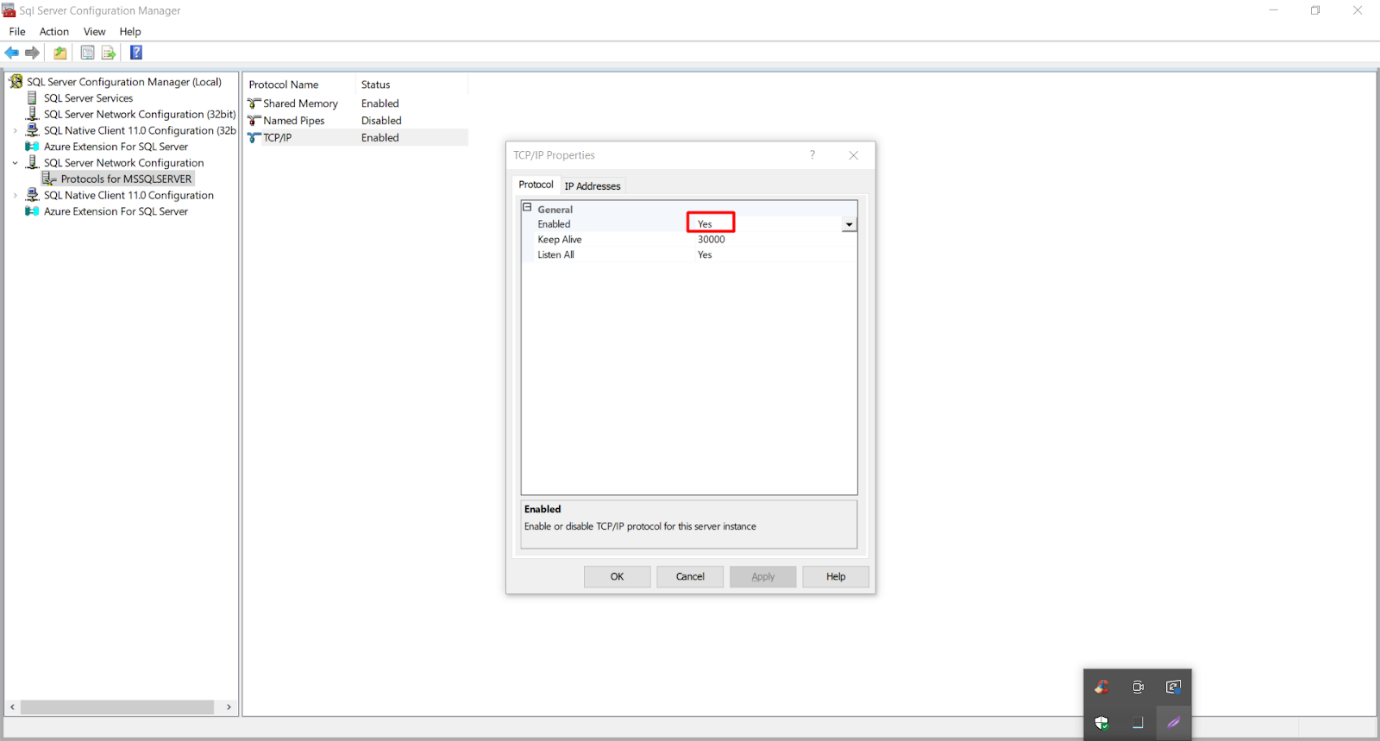
* SERVER máy chủ: TTTT\TTTT
* SERVER 1 có địa chỉ IP là 172.26.62.172
* SERVER 2 có địa chỉ IP là 172.26.62.184  
  Bước 0: Đăng nhập vào tài khoản sa của SERVER chính với tài khoản system admin

Bước 1: Vào properties, cho phép truy cập từ xa

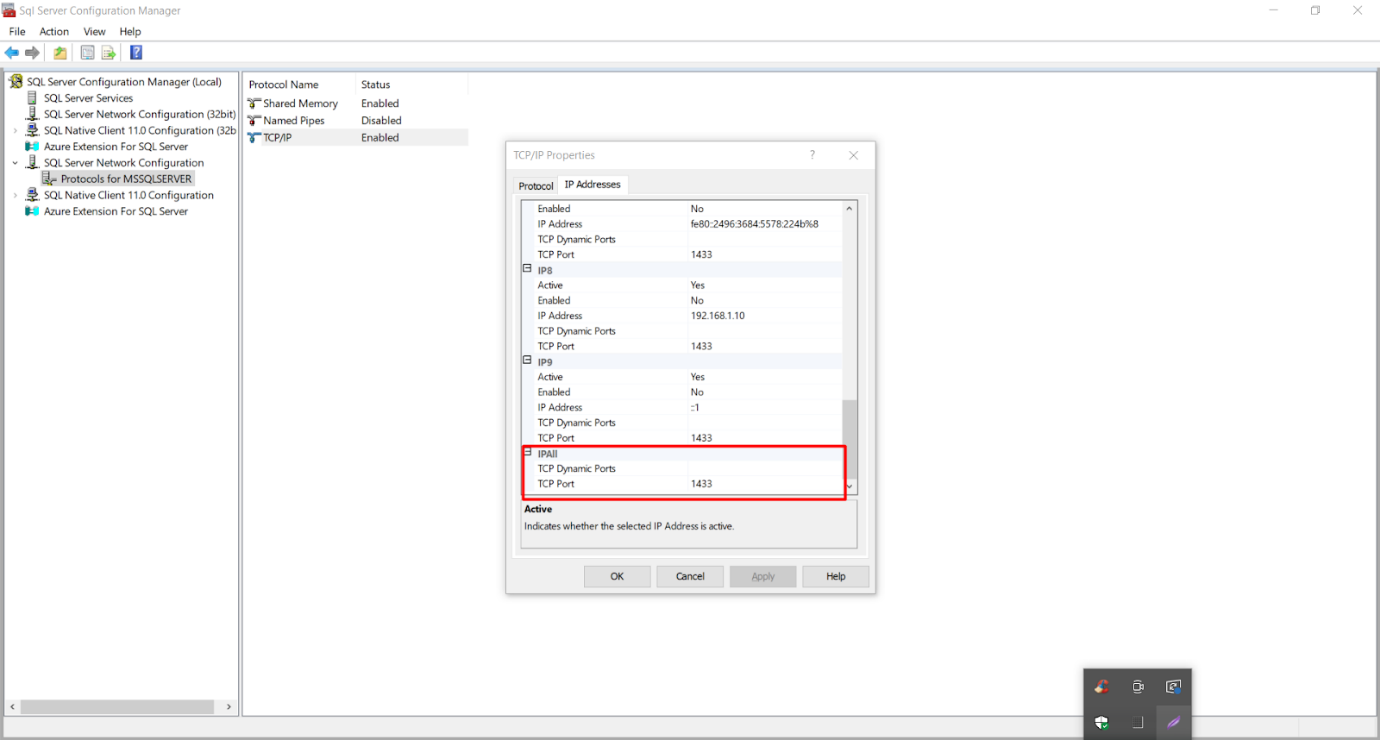


Bước 2: Vào SQL configuration manager, cài port 1433 và run SQL Browser

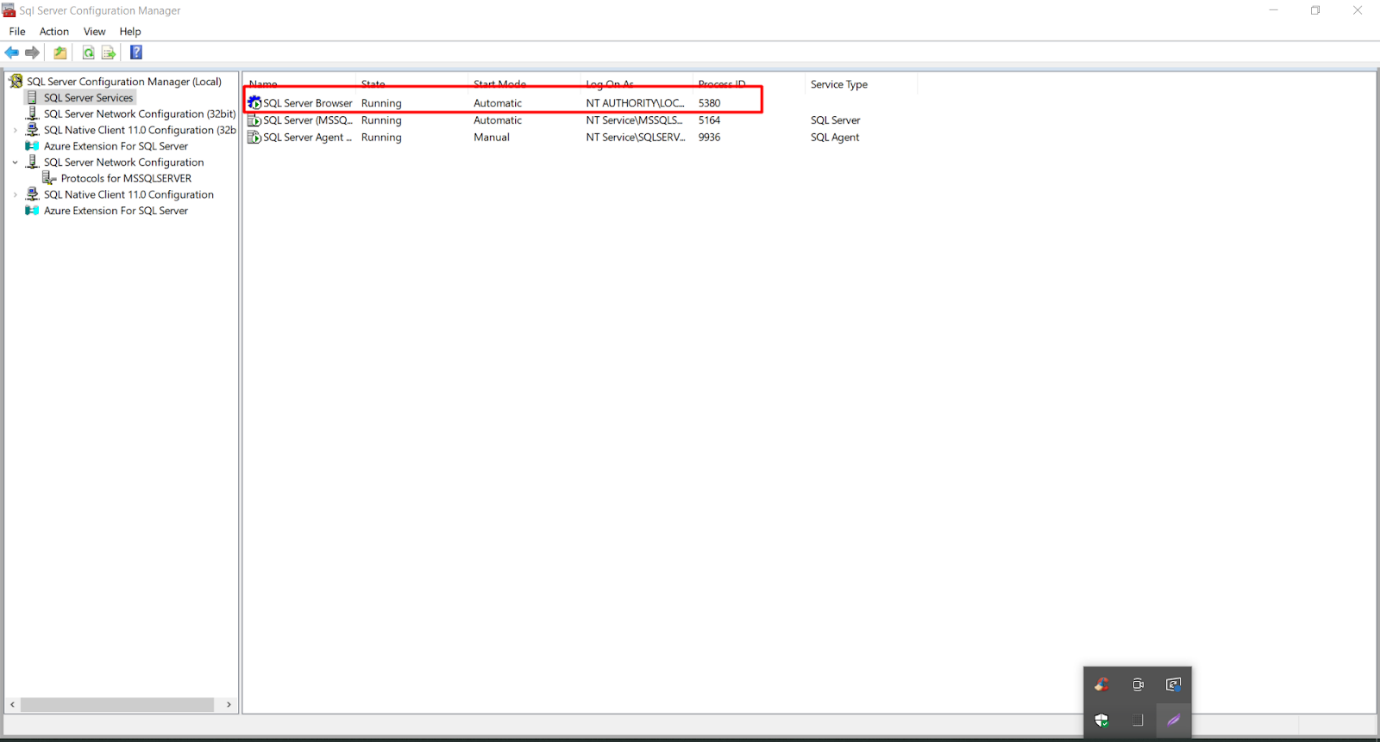
* Mở kết nối TCP/IP



* Cài đặt cổng 1433



* Chạy SQL Server Browser

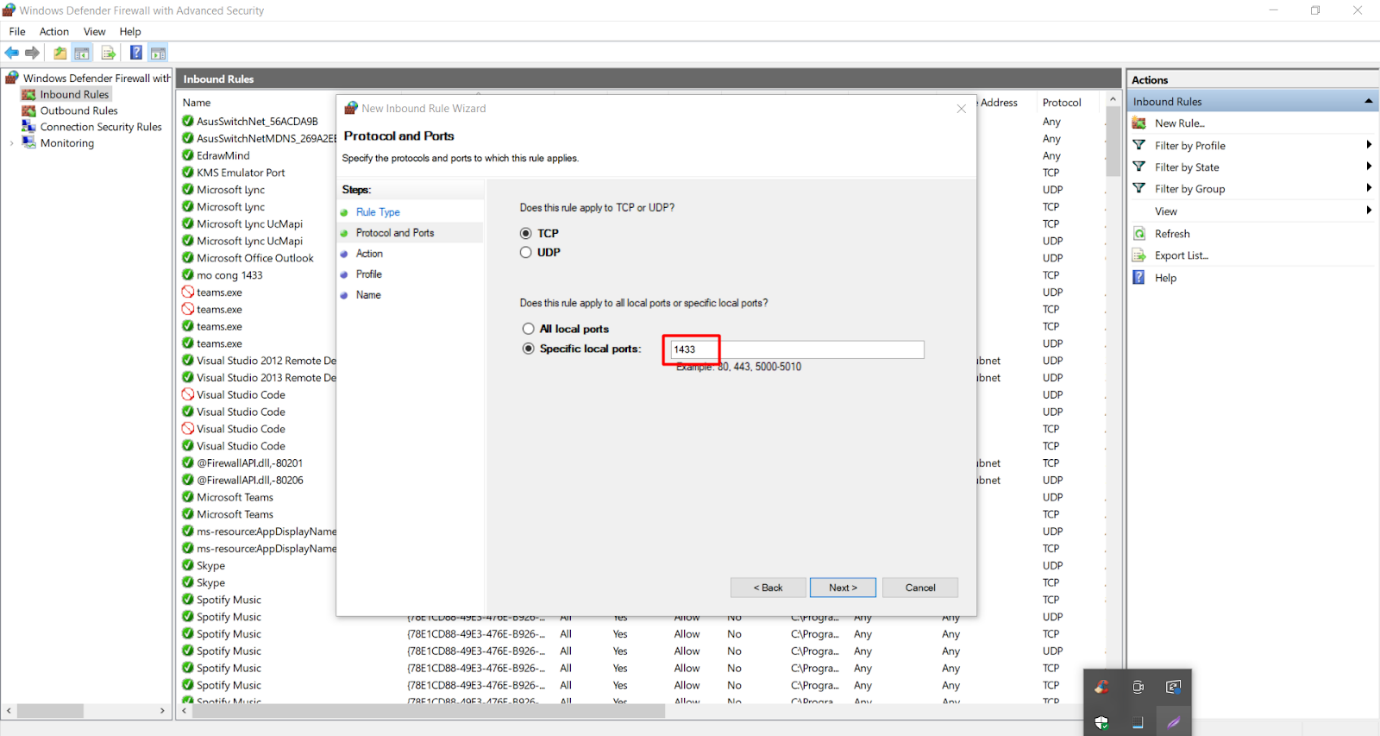


Bước 3: Vào ‘Defender firewall with Advanced Security’, mở port 1433 ở inbound và outbound

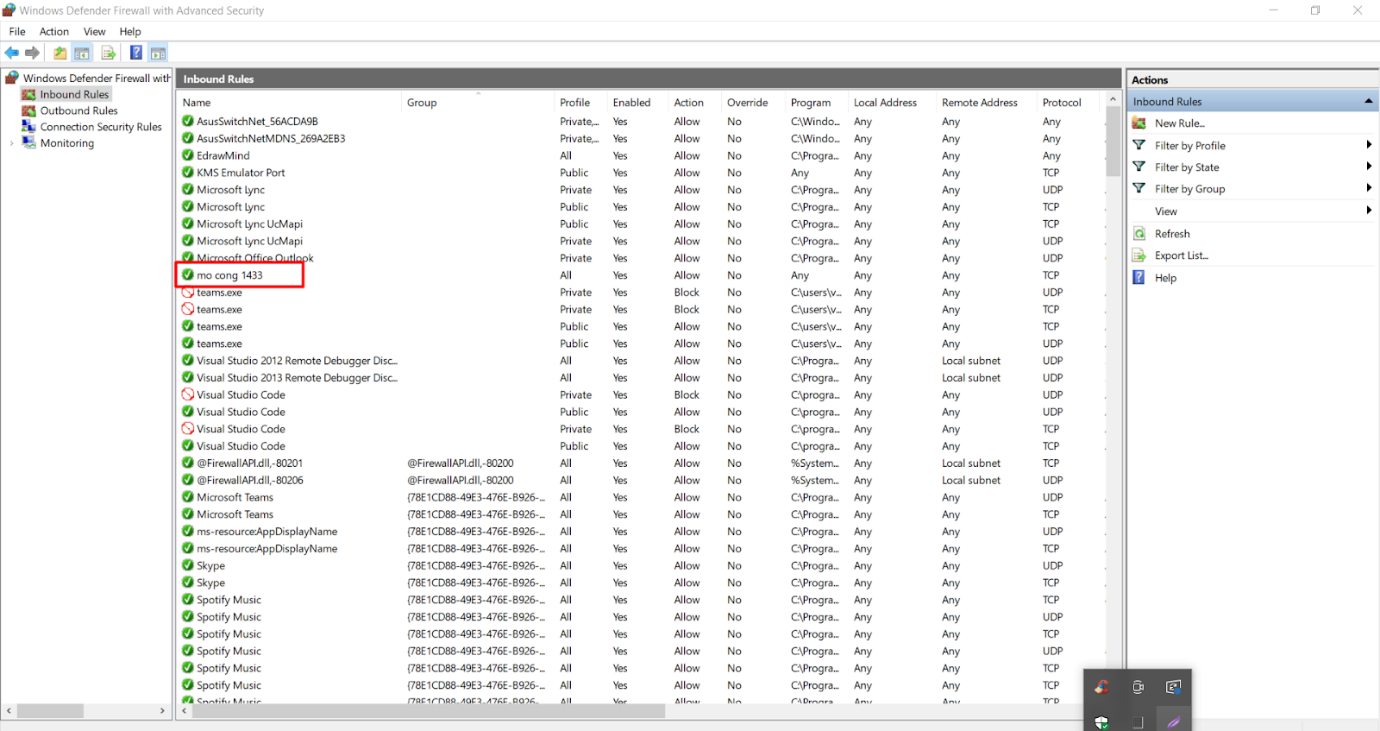
* Chọn Port



* Mở cổng 1433

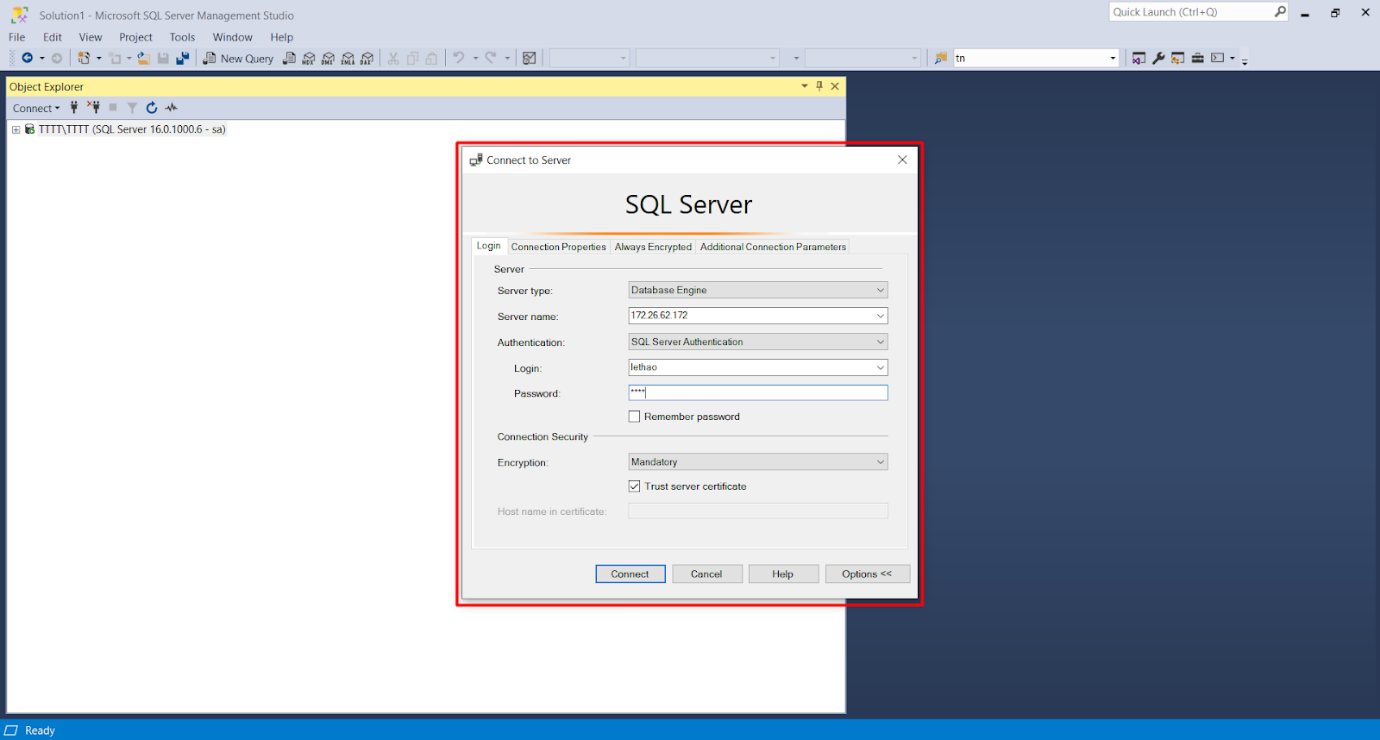


* Next cho đến cuối để đặt tên ‘mo cong 1433’ và hoàn thành mở cổng 1433 trên firewall

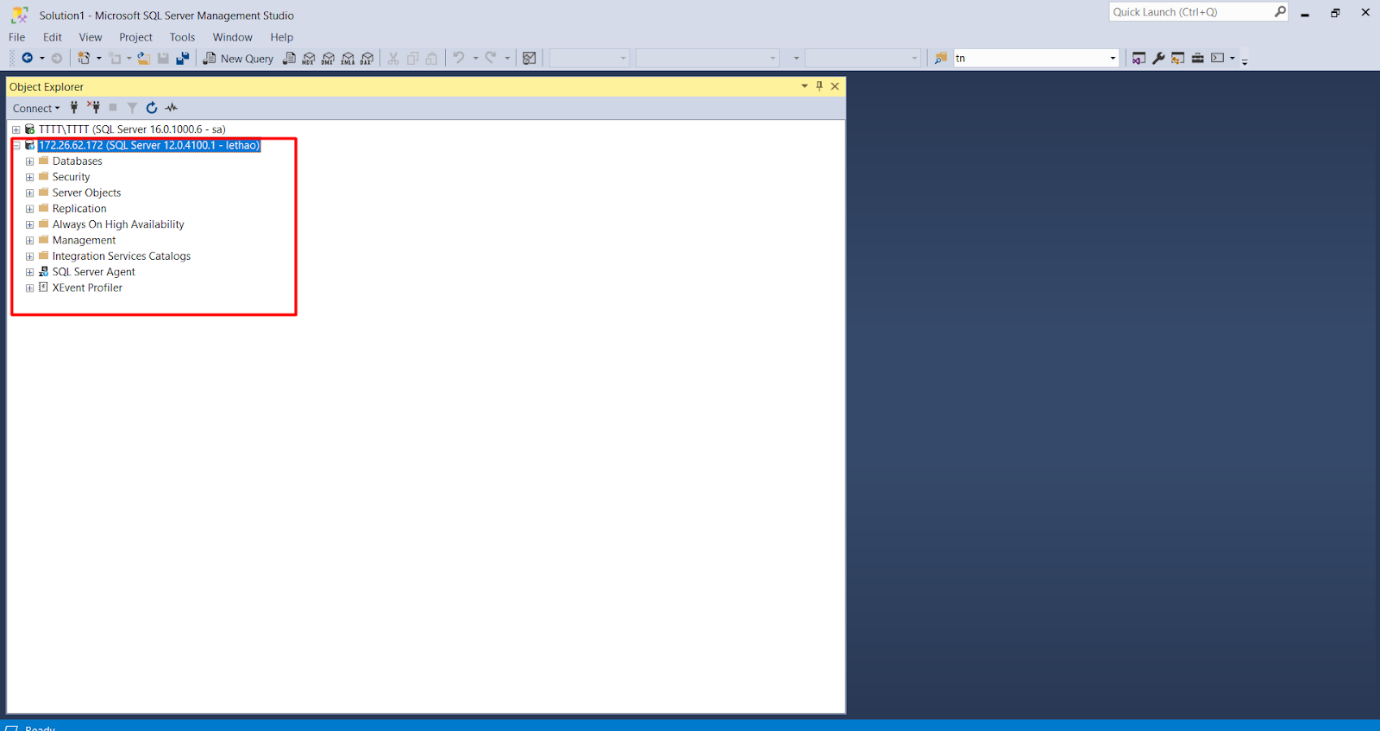


*Thực hiện các bước trên tương tự trên 2 máy con còn lại*

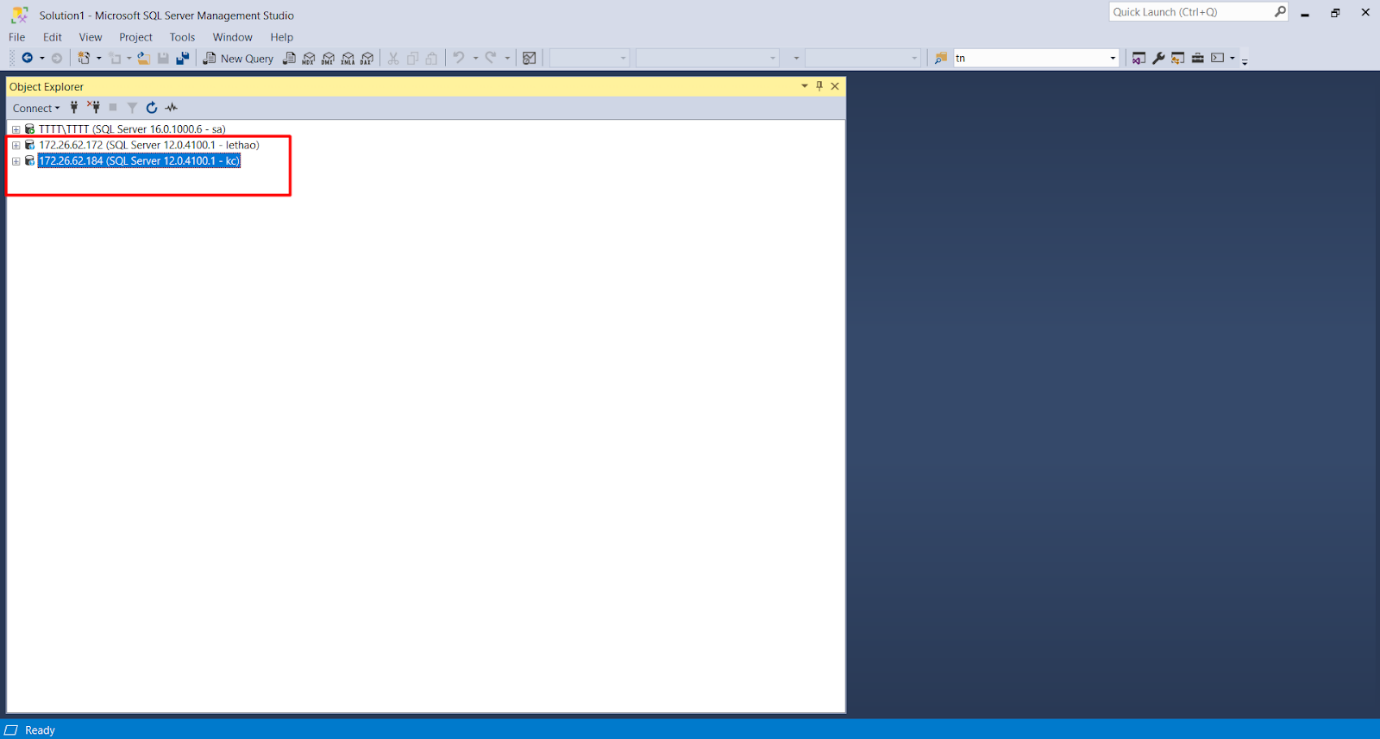
Bước 4: Kết nối vào SERVER 1 là 172.26.62.172(đăng nhập với quyền systemsadmin)



* Kết nối SERVER 1 thành công



* *Thực hiện tương tự với SERVER 2 172.26.62.184*

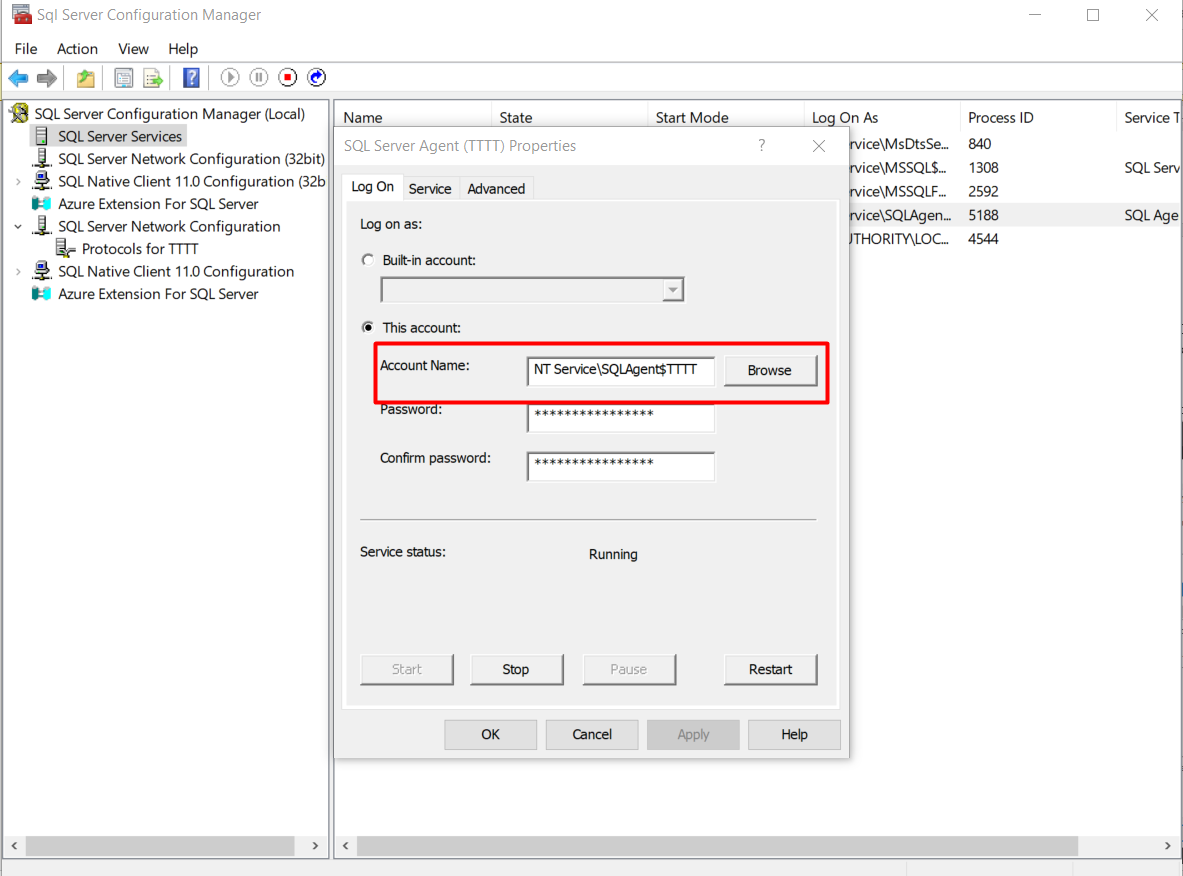


* Đã kết nối thành công cả 2 server

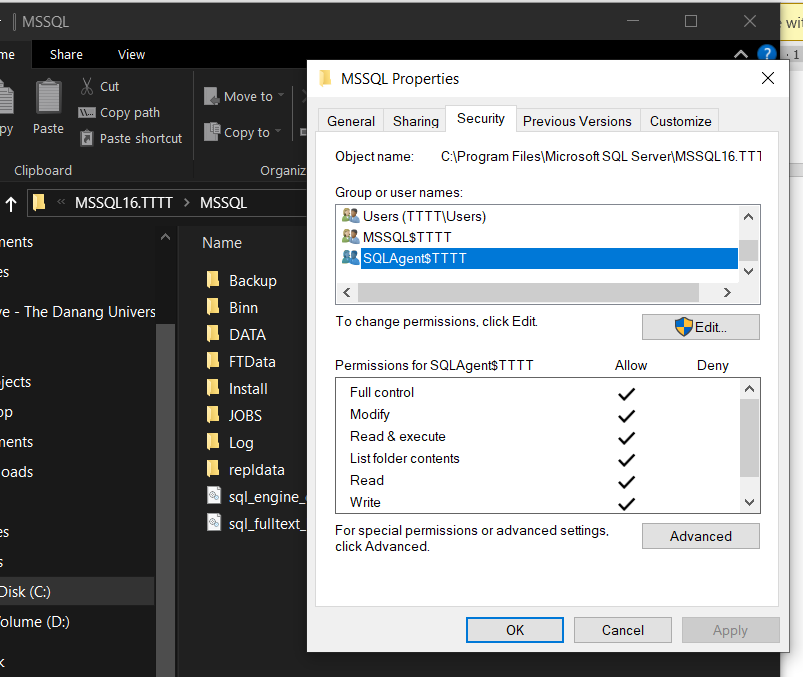
**PHẦN 2: THỰC HIỆN PHÂN TÁN DỌC CSDL**

Bước 0: Cho phép tài khoản agent quyền full control cho file repldata.

* + Lấy tài khoản của sql agent trong SQL Configuration



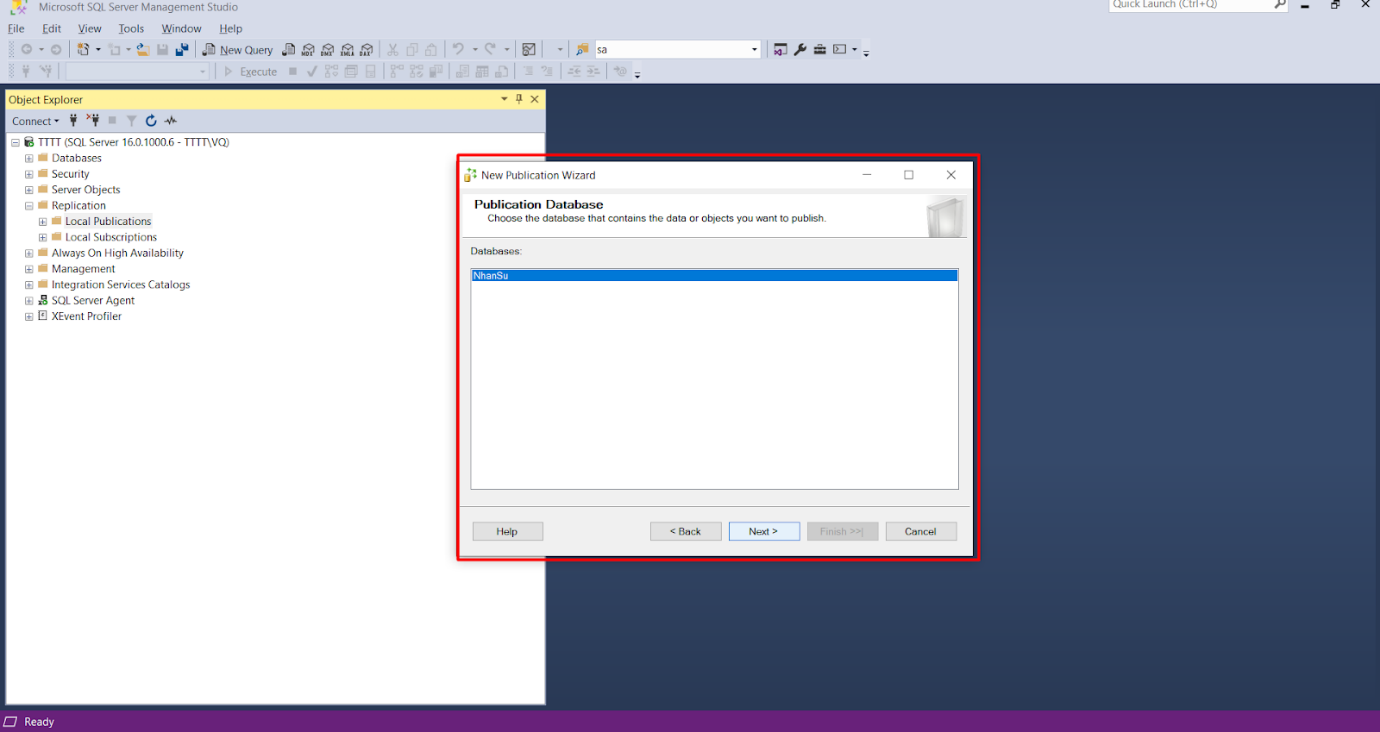
* + Cho phép tài khoản Agent quyền full control cho file repldata



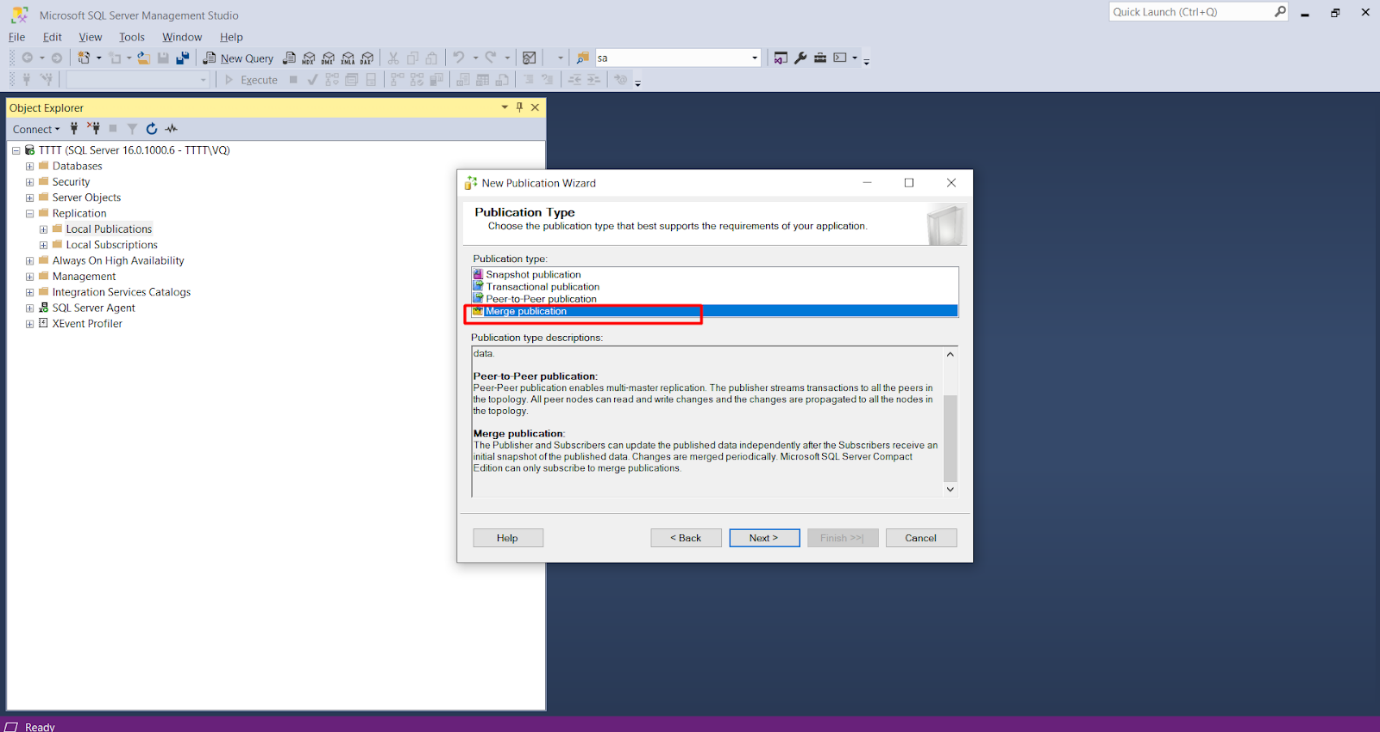
***Thực hiện cho phép tài khoản Agent quyền full control cho file repldata cho 2 server con.***

Bước 1: Tạo Publications

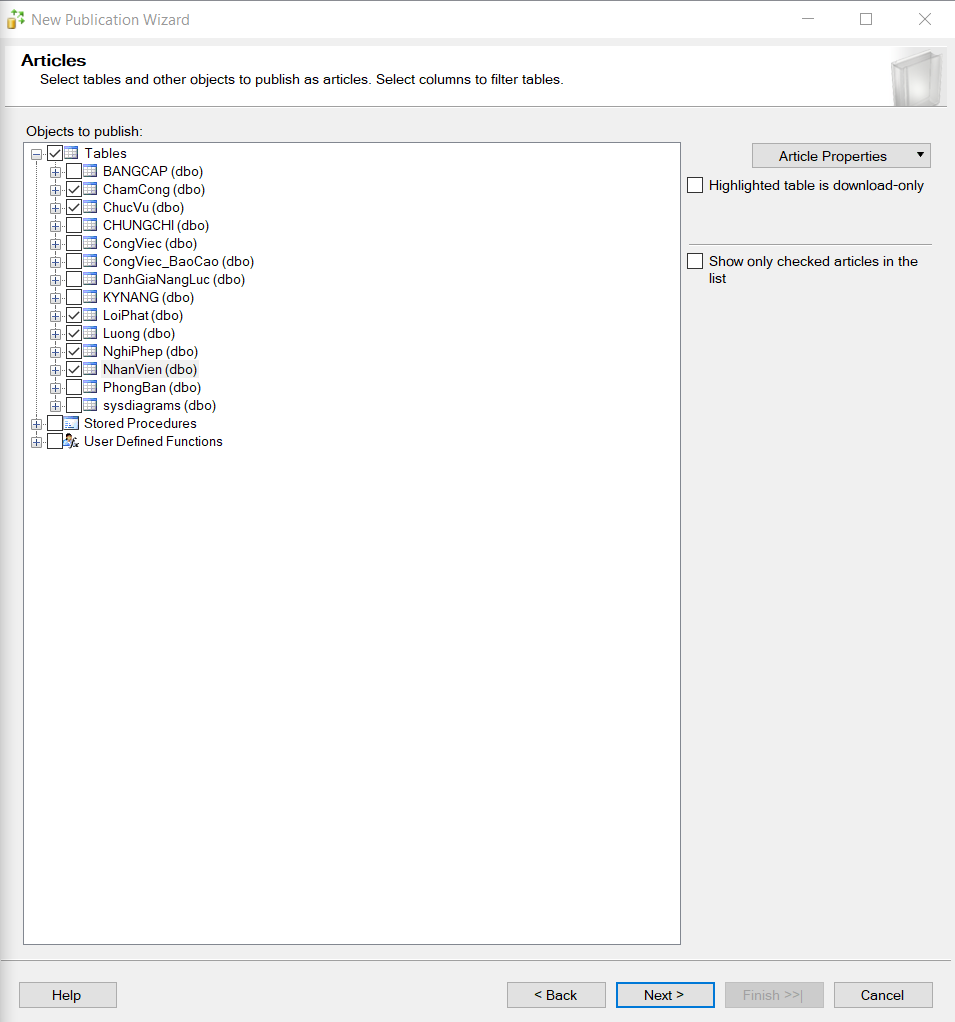
* + Chọn database thực hiện phân tán



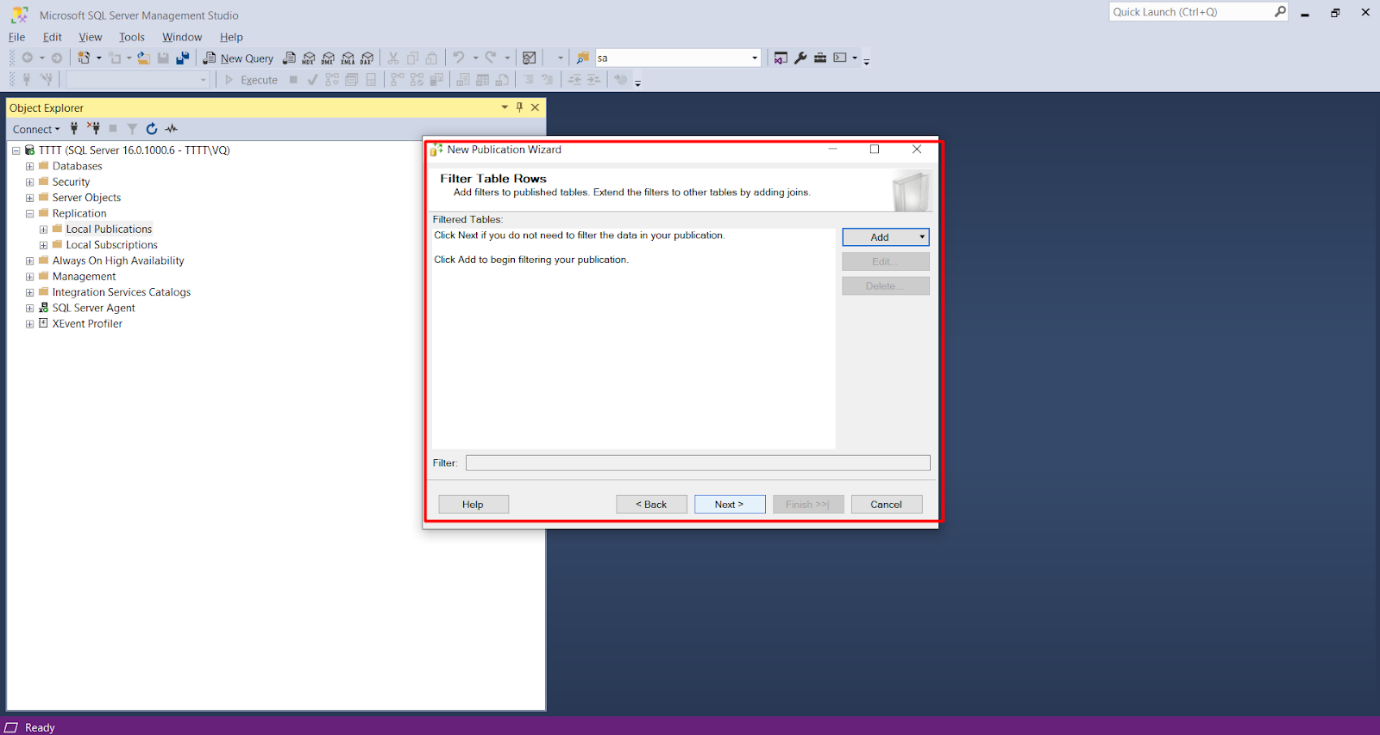
* Chọn Merge publication *(có thể thực hiện thay đổi trên dữ liệu, và đồng bộ những thay đổi này ở cả hai phía)*



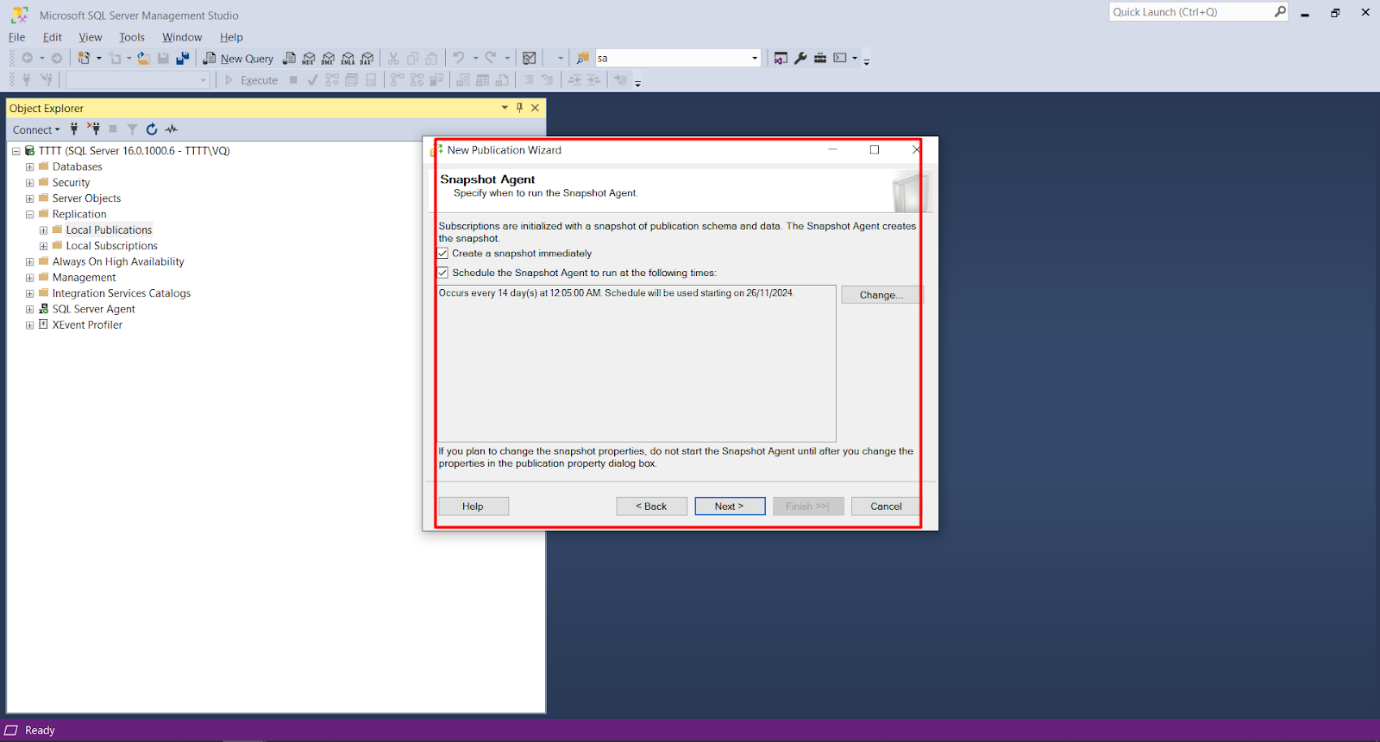
* Chọn những bảng để thực hiện phân tán cho publications Lương theo kịch bản



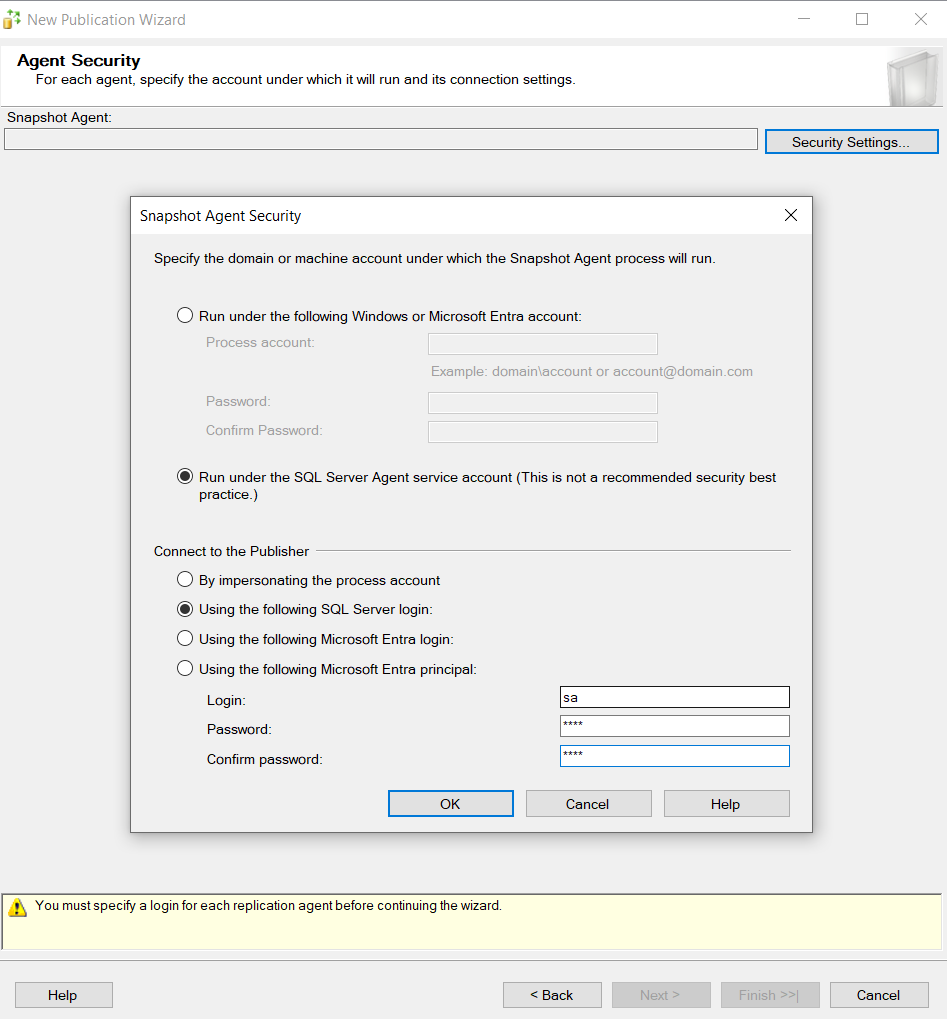
* Bỏ qua bước tạo Filter vì đang phân tán dọc



* Tạo snapshot



* Connect với Publisher bằng tài khoản sa

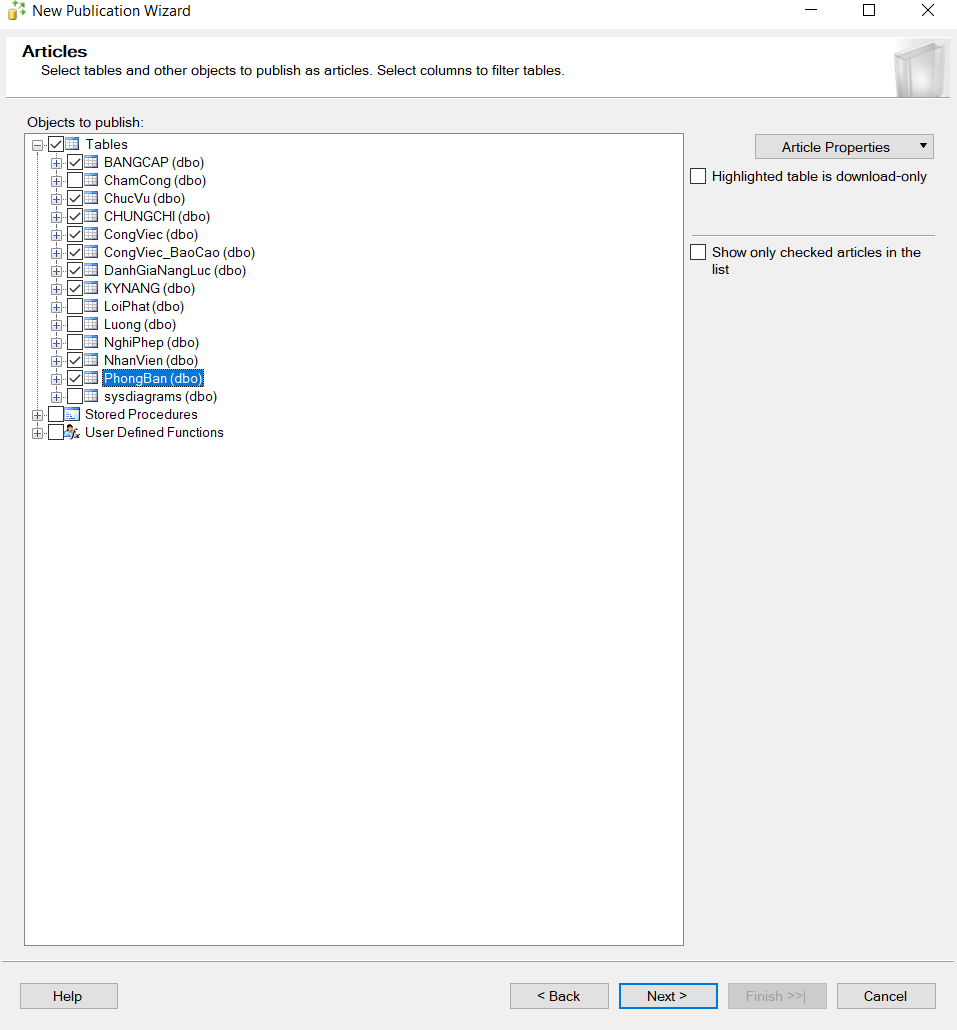


* Tạo Publication thành công

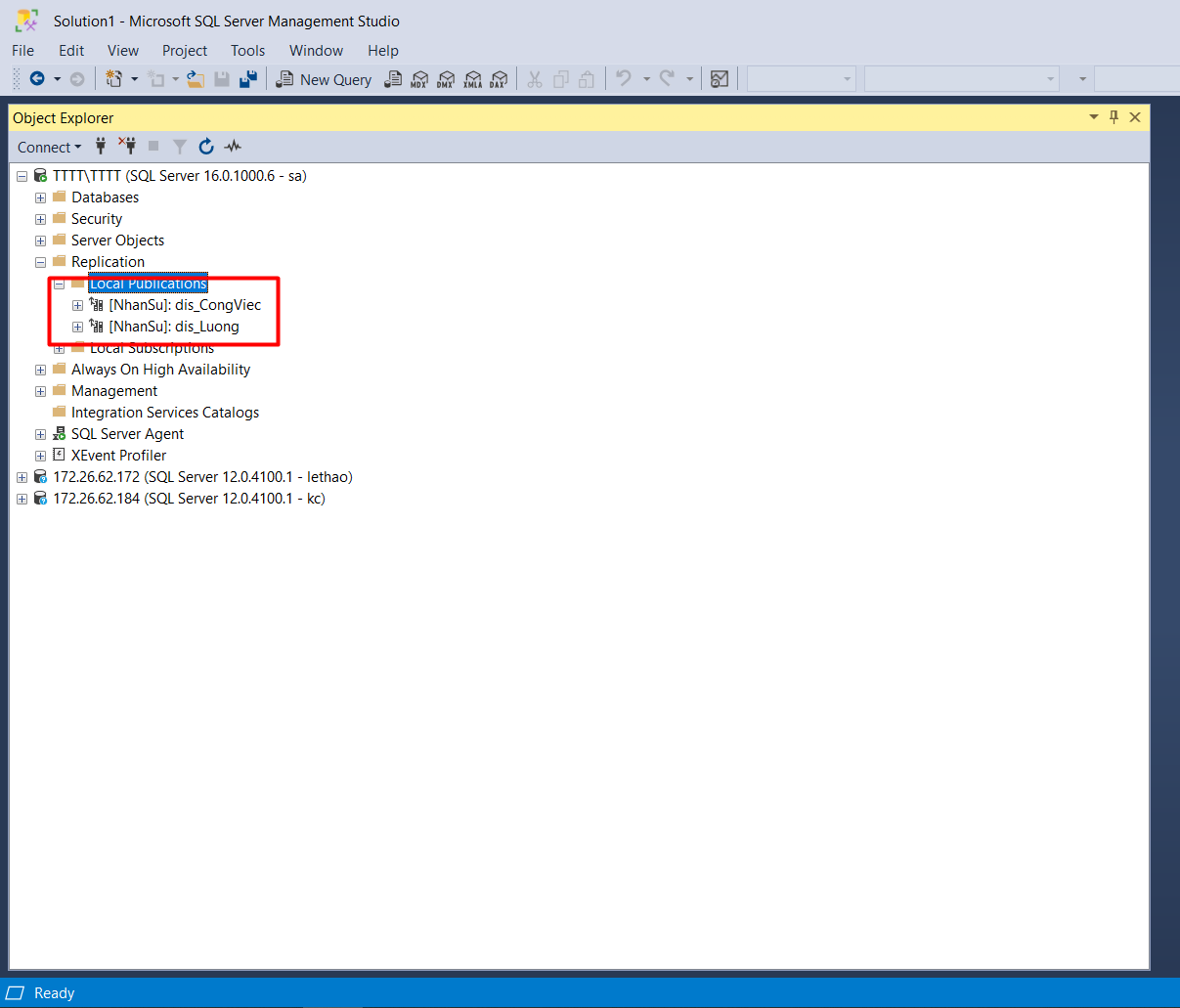


*Thực hiện các bước như trên với phân tán Công việc*

* Chọn các bảng phân tán công việc theo kịch bản

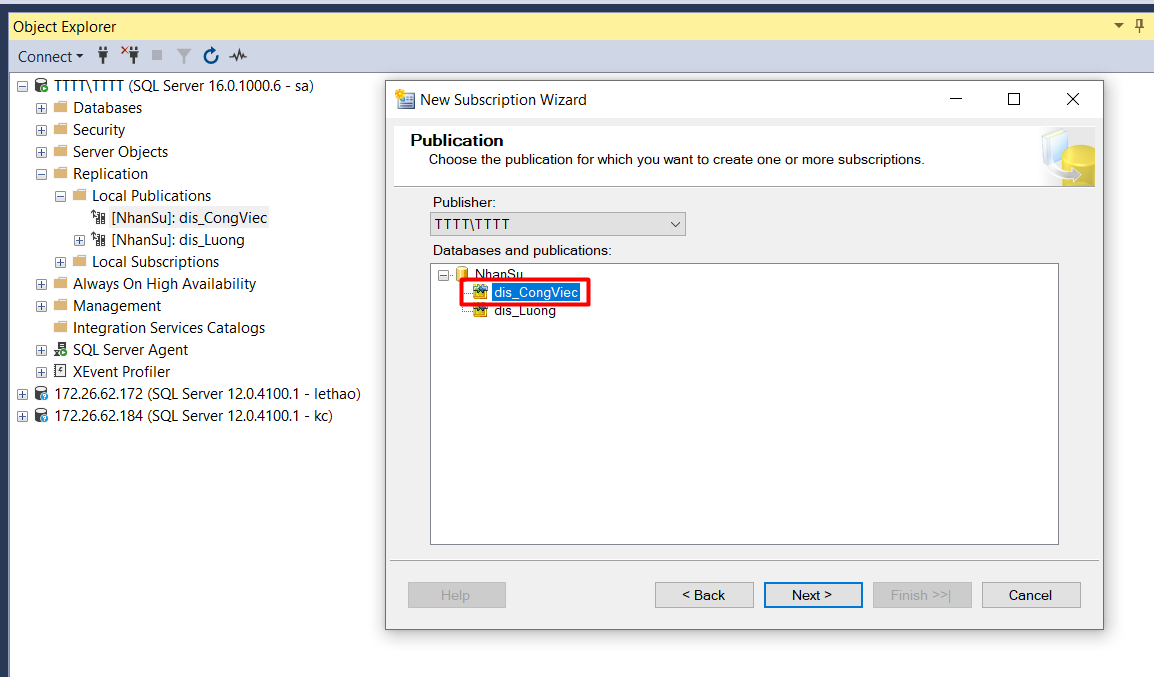


* Tạo thành công 2 publications

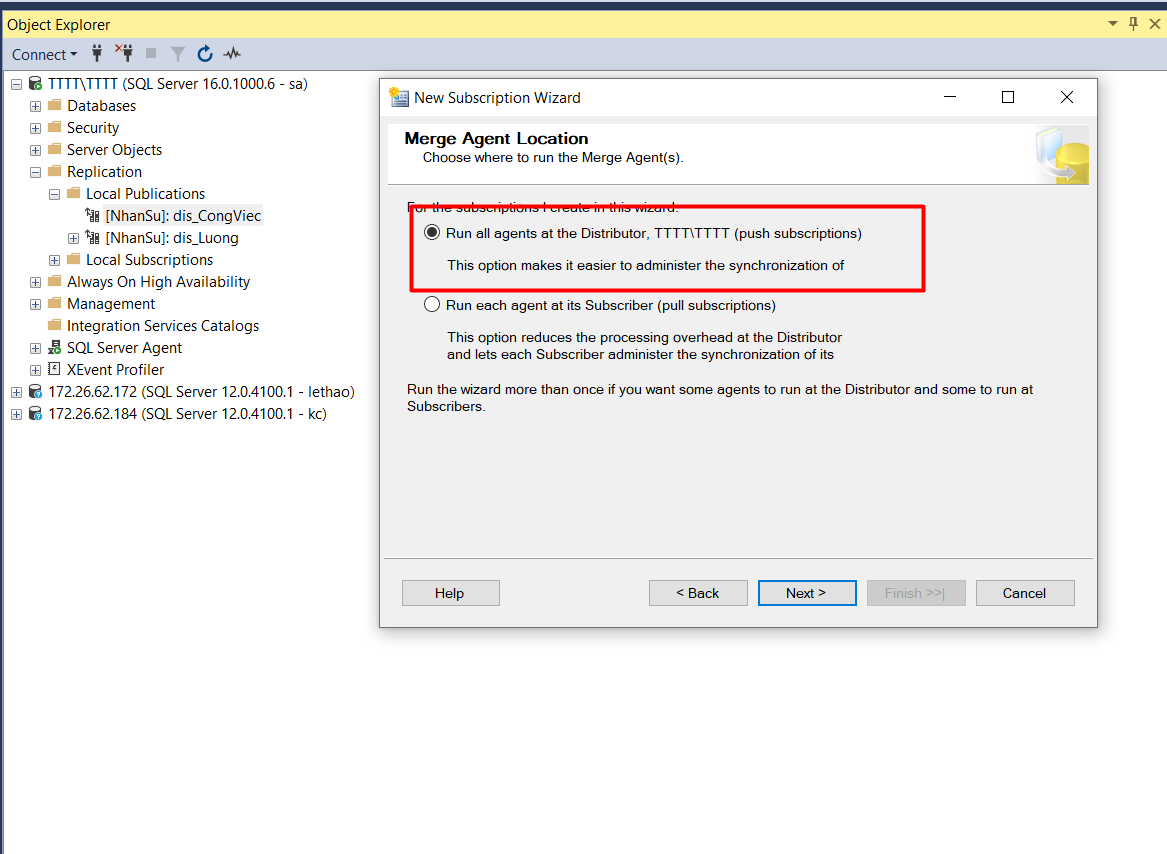


Bước 2: Tạo Subscriptions

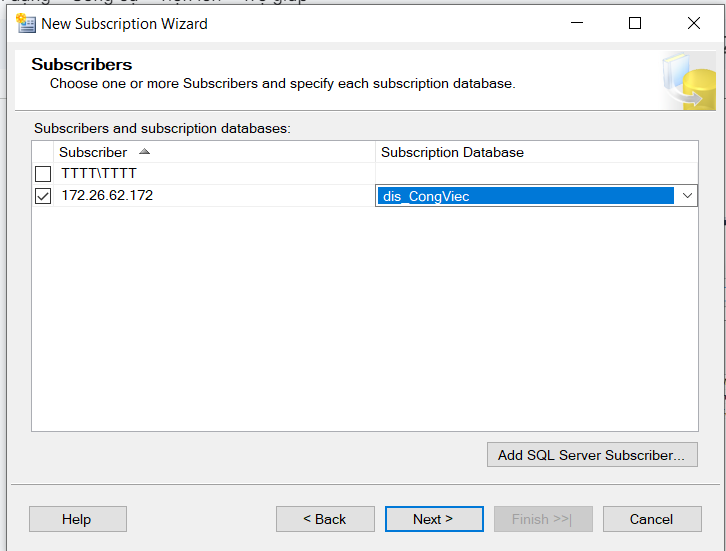
* Tạo subcription cho publication Công việc



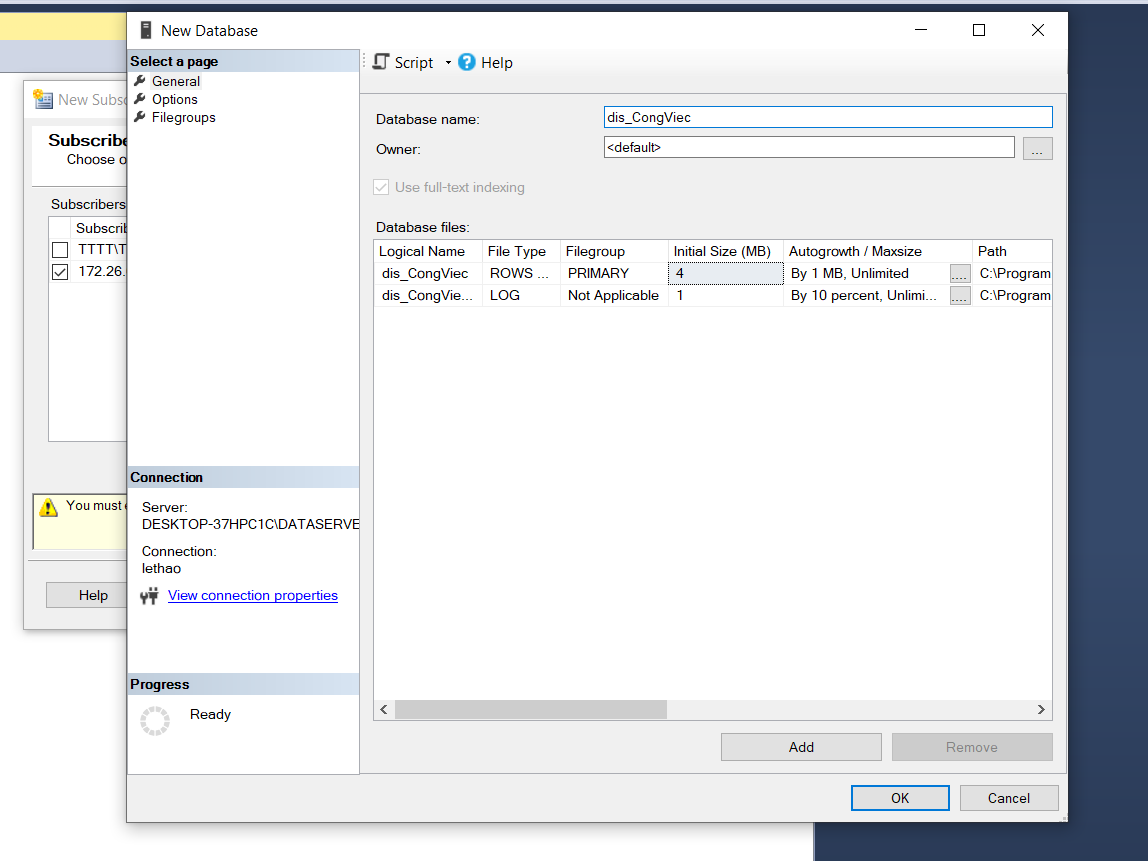
* Chọn đẩy dữ liệu từ Publisher đến Subscriber



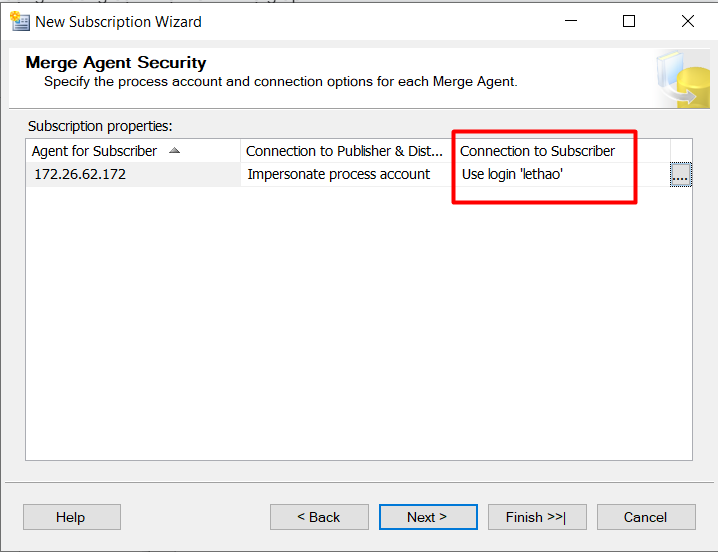
* Kết nối với server sẽ trở thành subcriber



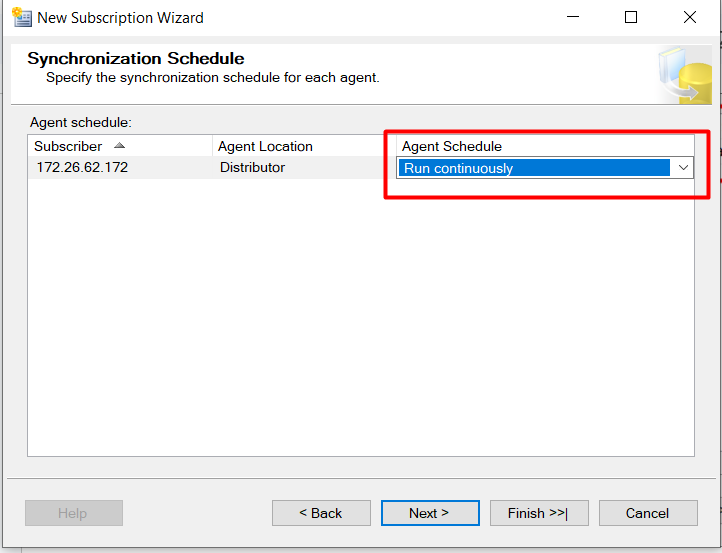
* Tạo database mới để lưu Publications



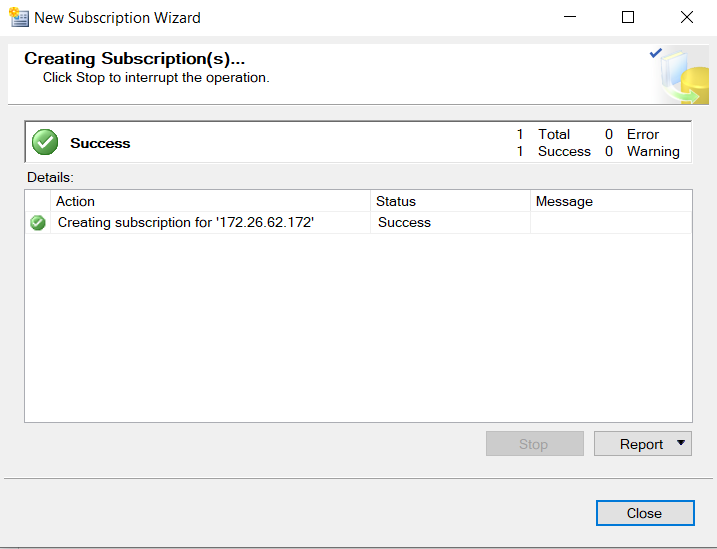
* Kết nối với subcripber bằng tài khoản có quyền system admin của 172.26.62.172



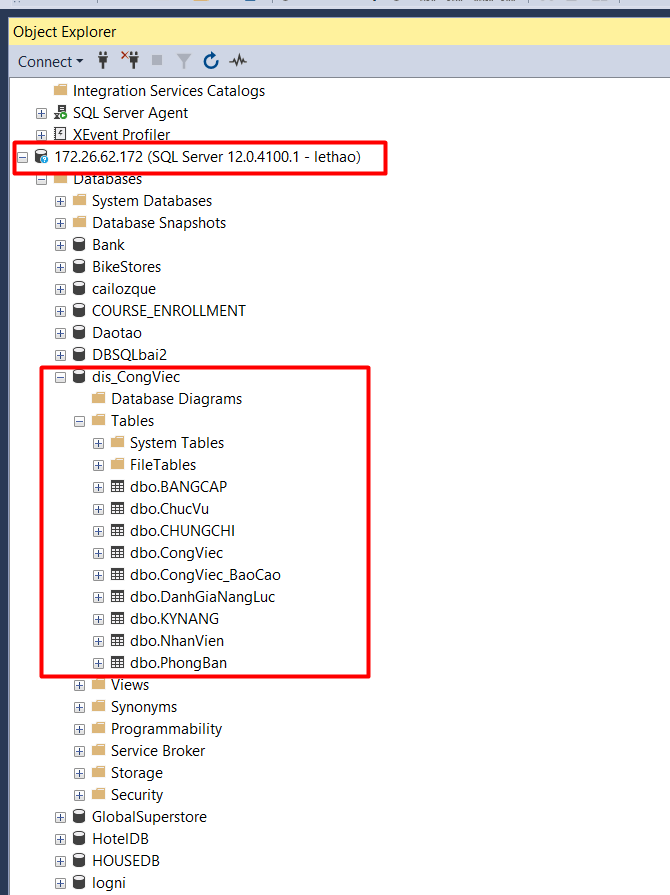
* Để Run continuously, để jobs được thực thi liên tục mà không bị gián đoạn.



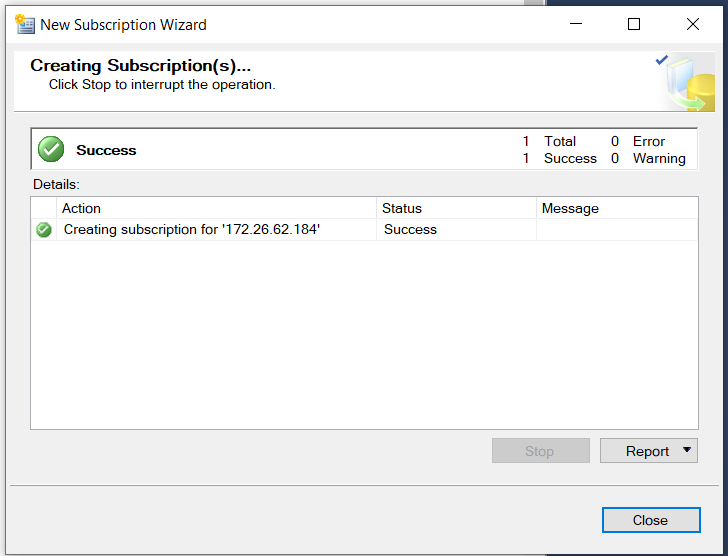
* Tạo subcription thành công



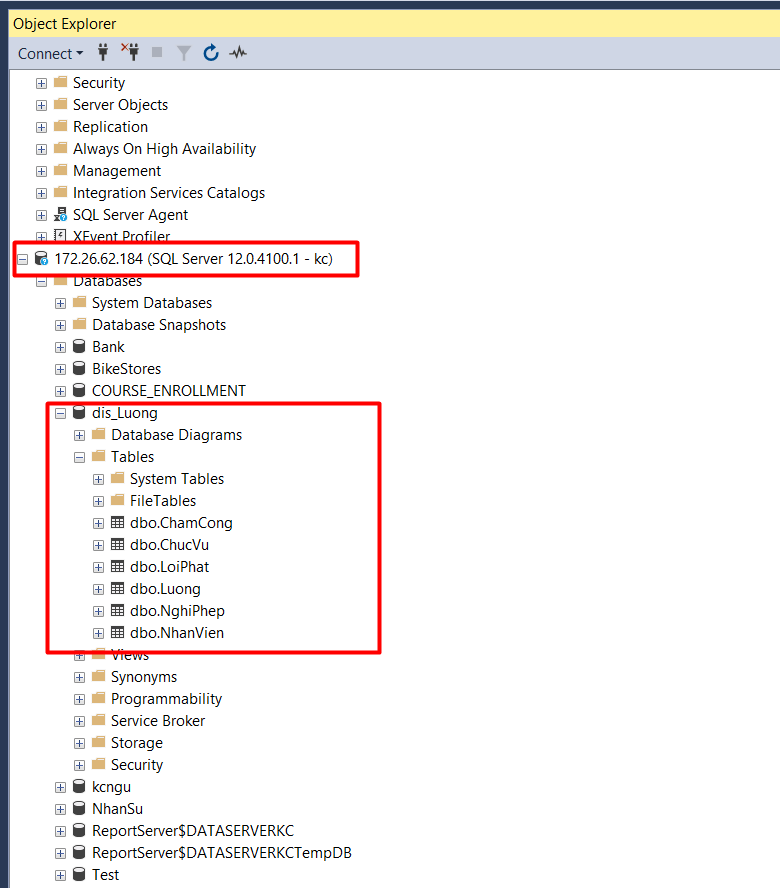
* Kiểm tra đã phân tán dữ liệu thành công sang server 172.26.62.



* ***Thực hiện tương tự như trên với publication dis\_Luong cho server 172.26.62.184***



* Phân tán publication dis\_Luong thành công



# BACKUP:

## Lịch trình backup

Nhóm sẽ tiến hành thực hiện Backup tự động với 2 hình thức: Full Backup và Differential Backup. Lý do vì:

* Hệ thống quản lý nhân sự có dữ liệu chấm công và các thông tin cập nhật thường xuyên, đòi hỏi quy trình sao lưu đảm bảo an toàn cho dữ liệu này.
* Việc sử dụng Full Backup hàng ngày sẽ làm tăng dung lượng lưu trữ, gây lãng phí tài nguyên lưu trữ và tăng thời gian sao lưu.
* Differential Backup được sử dụng để giảm tải, chỉ sao lưu các thay đổi kể từ lần Full Backup gần nhất, giúp tiết kiệm dung lượng ổ đĩa mà vẫn duy trì được tính toàn vẹn dữ liệu.

→ Sử dụng Full Backup kết hợp Differential Backup giúp cân bằng giữa an toàn dữ liệu và hiệu quả sử dụng tài nguyên lưu trữ.

Lịch trình Backup:

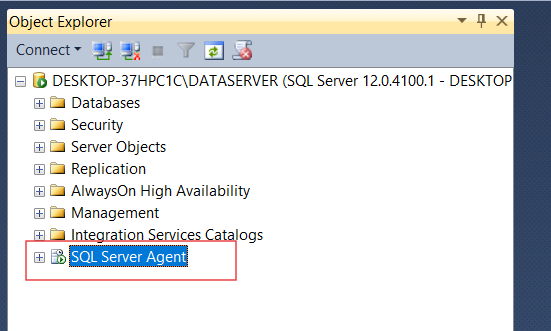
| Full-Backup | Differential Backup |
| --- | --- |
| Backup 1 tuần/1 lần vào thứ 7 (12:00:00 PM)  → Thứ 7 là thời điểm kết thúc 1 tuần làm việc, nên cần Full-backup tất cả những thay đổi dữ liệu so với tuần trước. | Backup hàng ngày từ thứ 2 đến thứ 6 (12:00:00 PM)  → Từ thứ 2 đến thứ 6, dữ liệu sẽ có sự thay đổi ở một số bảng. Lúc này, differential-backup những thay đổi sẽ đỡ tốn dung lượng hơn là full-backup. |
| 1. Tất cả thời điểm backup dữ liệu ở các ngày trong tuần đều vào khung giờ 12:00:00 PM (buổi trưa). Vì:   * Đây là thời điểm nghỉ trưa của nhân viên trong công ty, nên sẽ không thao tác trên cơ sở dữ liệu, giúp giảm thiểu nguy cơ xung đột hoặc lỗi ghi dữ liệu trong quá tình sao lưu. * Đây cũng là thời điểm các máy tính trong công ty được bật, nên không lo ngại về việc hệ thống hoặc thiết bị sao lưu không được hoạt động.   → Thời điểm này tránh được giờ cao điểm sử dụng dữ liệu, đảm bảo hiệu suất tối ưu cho quá trình backup mà không làm ảnh hưởng đến hoạt động làm việc của nhân viên, nên nhóm quyết định chọn Backup dữ liệu vào lúc 12 giờ trưa  2. Các file backup có tuổi thọ 4 tuần kể từ khi lưu vào ổ đĩa sẽ được tự động xóa.  → Điều này giúp giải phóng dung lượng lưu trữ trên hệ thống, đảm bảo không gian đủ cho các bản backup mới. | |

## DEMO BACKUP

Bước 1: Tạo 2 thư mục “Differential-backup” và “Full-backup” vào ổ đĩa C

Bước 2: Khởi động SQL Server Agent

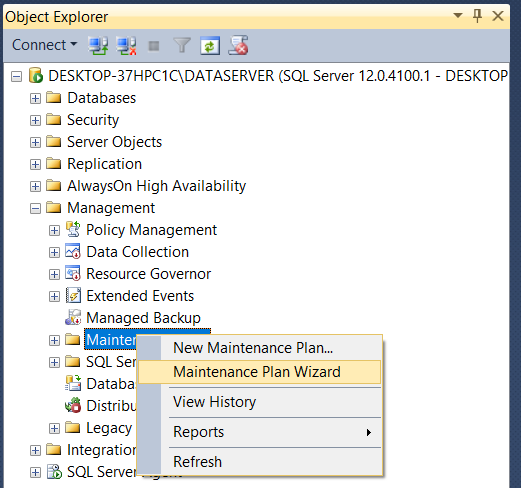
* Click chuột trái vào SQL Server Agent
* Chọn Start



Bước 3: Thiết lập lịch trình full-backup tự động:

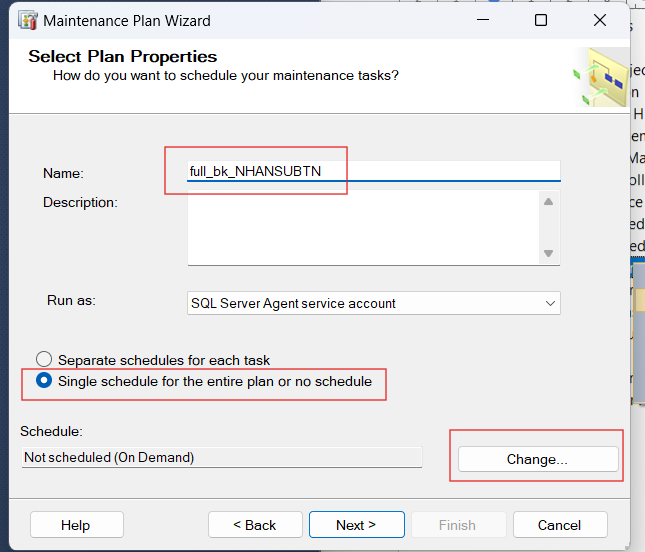
3.1. Mở cửa sổ Maintenance Plan Wizard bằng cách:

* Truy cập vào thư mục Management
* Trong thư mục Management, click chuột trái vào thư mục Maintenance Plans→ chọn Maintenance Plan Wizard



3.2: Bắt đầu thiết lập Full-Backup:

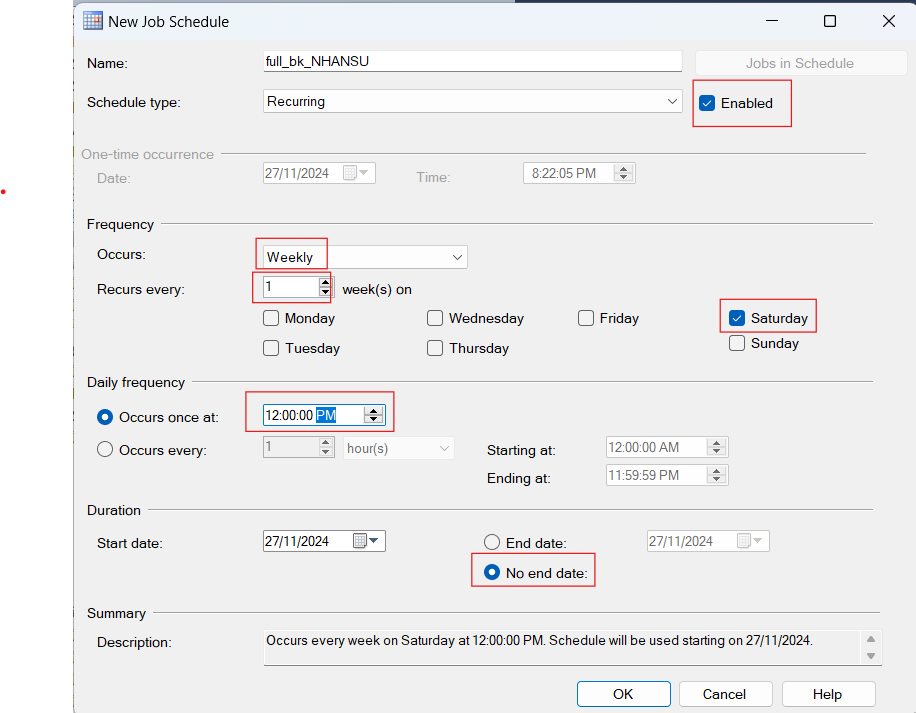
* Đặt tên cho job: full\_bk\_NHANSU và thiết lập các mục tương ứng



* Click Change ở mục Schedule để tiến hành thiết lập lịch backup

3.3. Thiết lập lịch trình full\_backup tự động:

* Occurs: Weekly → full backup sẽ xảy ra theo tuần
* Recurs every: 1 week → full backup sẽ tiến hành mỗi tuần 1 lần vào thứ 7 (Sartuday)
* Occurs once at: 12:00:00PM→ full backup sẽ diễn ra vào 12 giờ trưa thứ 7
* No end date: lịch trình full-bakup không có ngày dừng lại



Sau khi đã thiết lập lịch trình full - backup tự động, click OK→Next

3.4. Thiết lập các task có trong lịch trình full-backup:

Ở cửa sổ này, tiến hành chọn 2 task là:

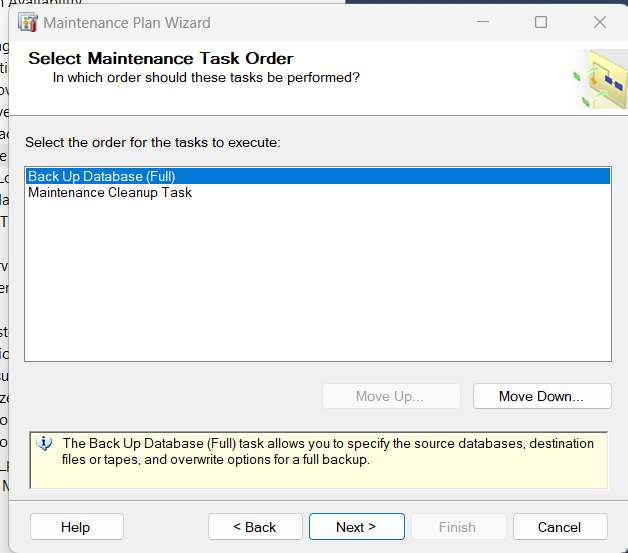
* Back Up Database (Full)--> thiết lập task full-backup
* Maintenance Cleanup Task→Thiết lập task xóa các file full backup khi đến thời gian quy định

Sau đó, nhấn Next



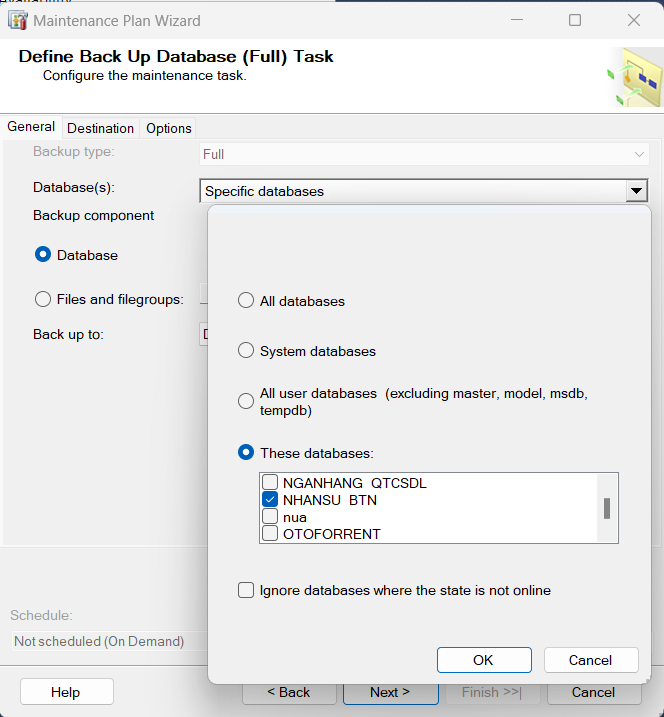
3.5. Thiết lập thứ tự thực hiện của các task:

Chọn Backup Database (Full) là task được thực hiện đầu tiên→Next

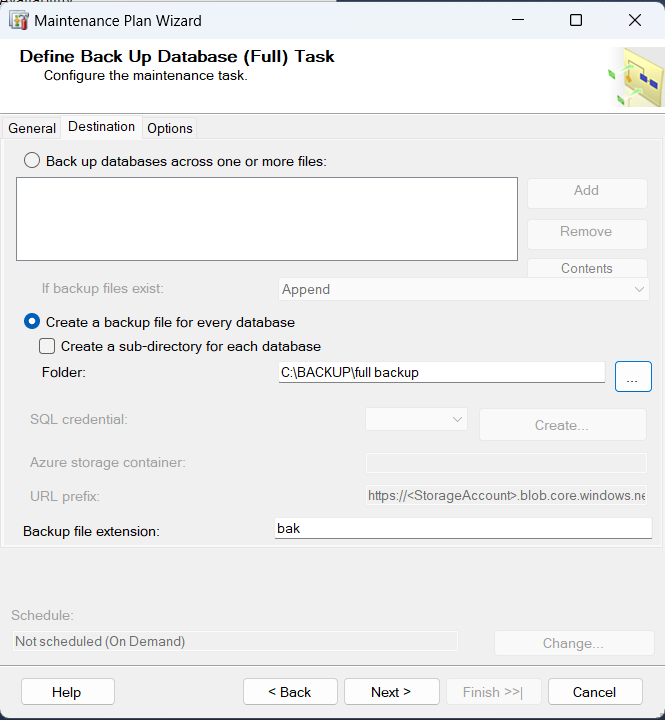


3.6. Tiến hành Define Backup Database (Full) Task

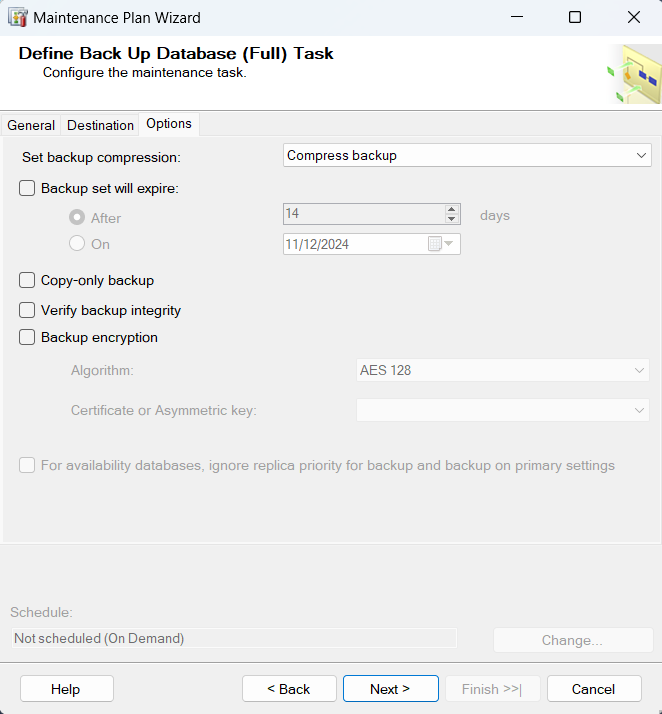
* Ở mục General, chọn Database cần Backup: NHANSU\_BTN



* Ở mục Destination, chọn đường dẫn đến folder sẽ chứa file backup: C:\BACKUP\full\_backup

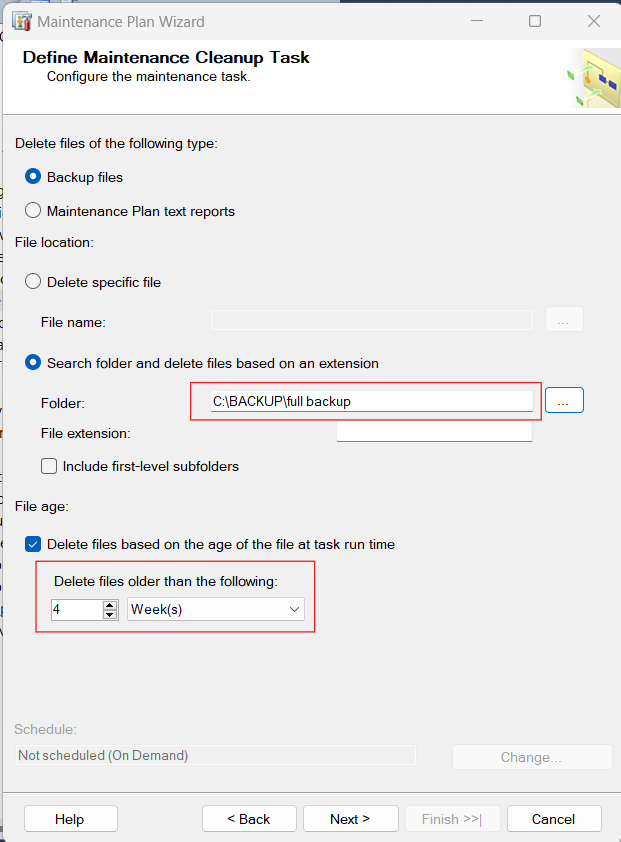


* Ở mục Options, chọn Compress backup để giảm dung lượng khi lưu vào ổ đĩa



3.7. Tiến hành Define Maintenance Cleanup Task:

* Chọn đường dẫn đến folder sẽ thực hiện Task Xóa các file full-backup: C:\BACKUP\full\_backup
* Thiết lập thời gian xóa các file full-backup: Thiết lập xóa các file full-backup đã tồn tại 4 tuần



Sau khi đã thiết lập hết các bước trên, Click Finish Wizad để hoàn thành việc tạo lịch trình full-backup tự động

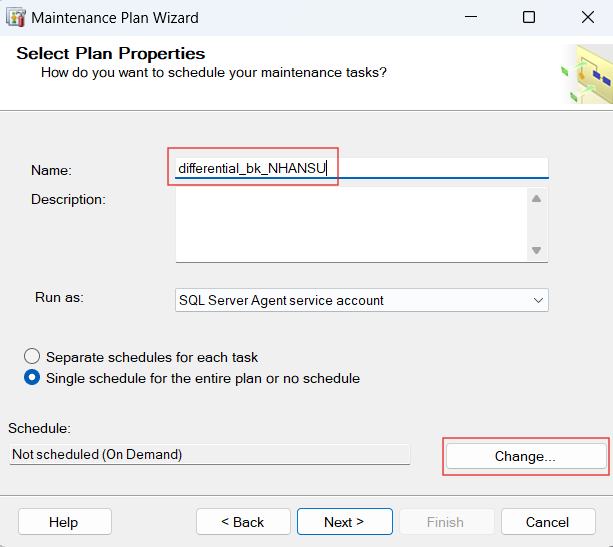
Bước 4: Thiết lập lịch trình Differential-Backup tự động

4.1. Mở cửa sổ Maintenance Plan Wizard (như đã trình bày ở phần thiết lập Full-Backup):

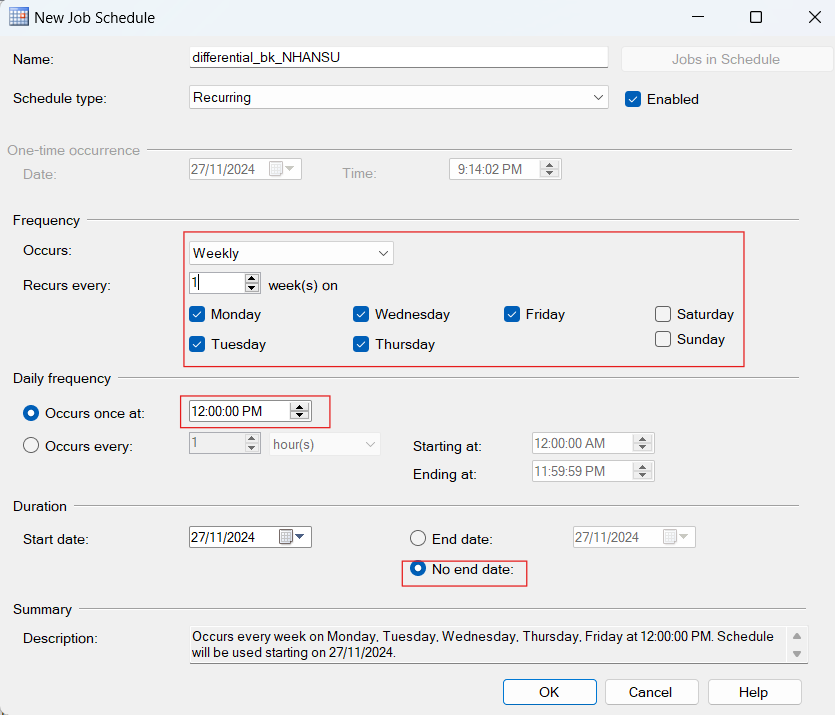
* Truy cập vào thư mục Management
* Trong thư mục Management, click chuột trái vào thư mục Maintenance Plans→ chọn Maintenance Plan Wizard

4.2: Bắt đầu thiết lập Differential-Backup:

* Đặt tên cho job: differential\_bk\_NHANSU và thiết lập các mục tương ứng
* Click Change ở mục Schedule để tiến hành thiết lập lịch backup



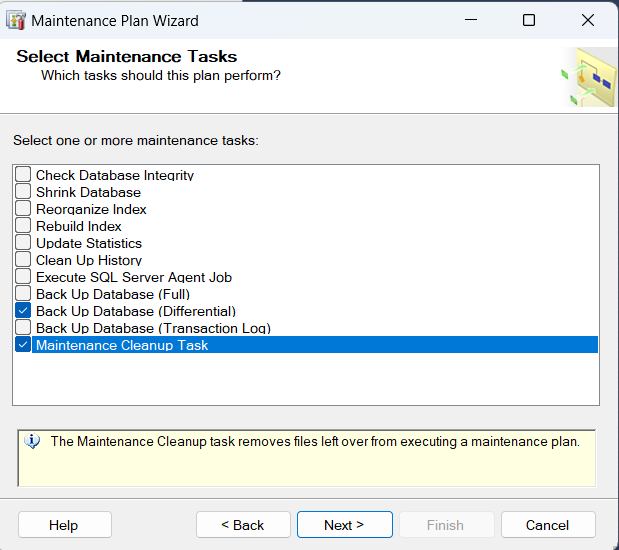
4.3. Thiết lập lịch trình differential-backup tự động:



* Thiết lập differential-backup theo mỗi tuần: vào thứ 2, thứ 3, thứ 4, thứ 5, thứ 6
* Thiết lập giờ backup: vào lúc 12:00:00PM (12 giờ trưa)
* Thiết lập ngày kết thúc kế hoạch differential-backup: không có

Sau khi đã thiết lập lịch trình, click OK để qua bước tiếp theo

4.4. Thiết lập các task có trong lịch trình differential-backup:



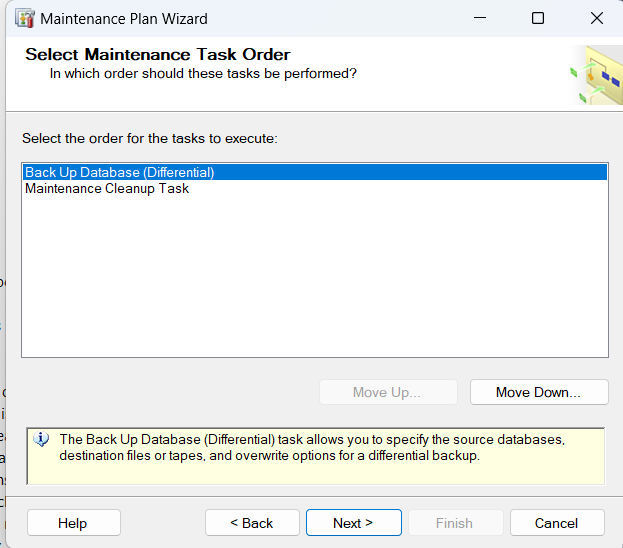
Ở cửa sổ này, tiến hành chọn 2 task là:

* Back Up Database (Differentiall)--> thiết lập task differentiall-backup
* Maintenance Cleanup Task→Thiết lập task xóa các file differential backup khi đến thời gian quy định

Sau đó, nhấn Next

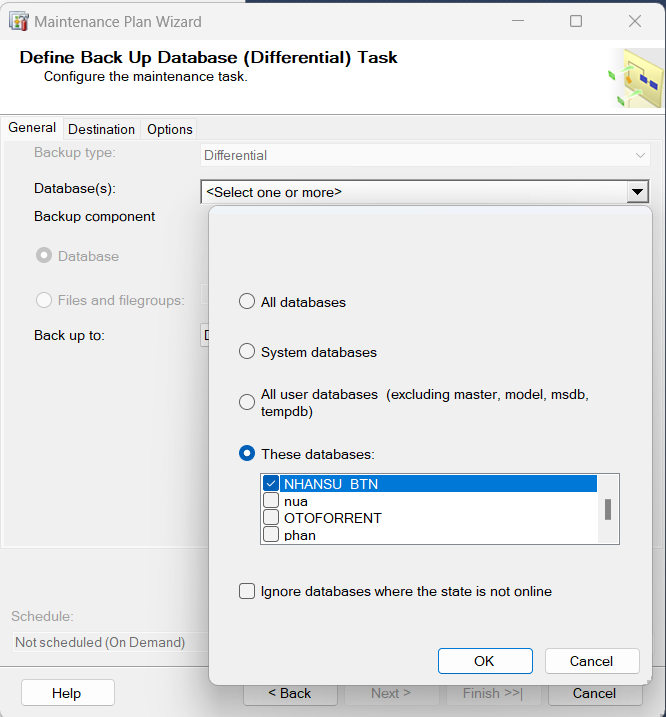
4.5. Chọn thứ tự thực hiện của các task:

Chọn Backup Database (Diferential) là task được thực hiện đầu tiên→Next

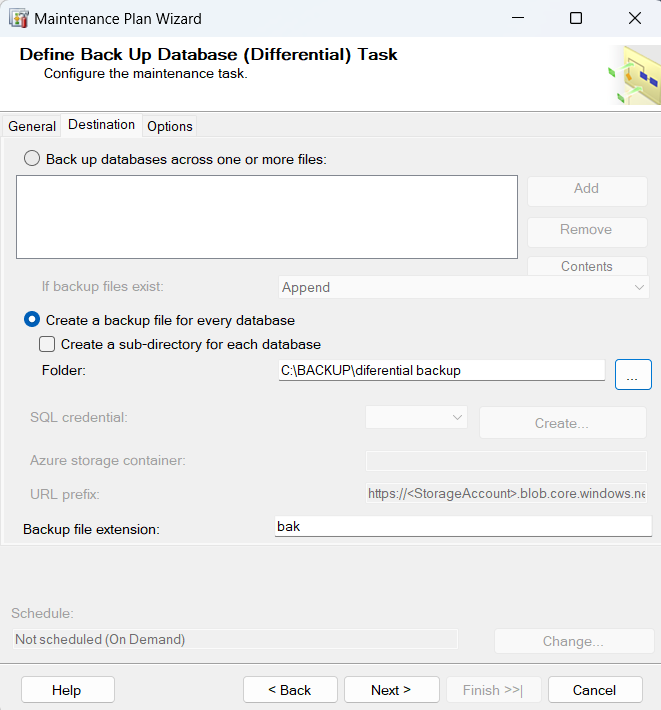


4.6. Tiến hành Define Backup Database (Differential) Task

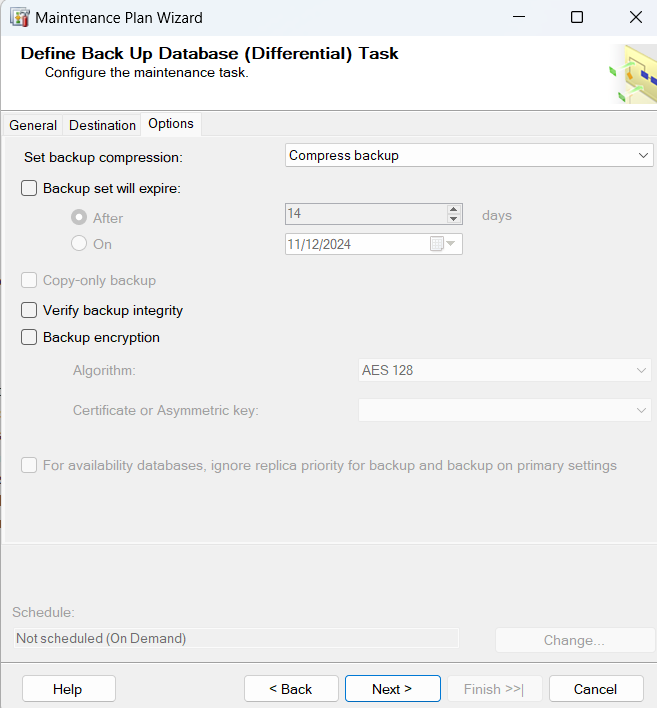
* Ở mục General, chọn Database cần Backup: NHANSU\_BTN



* Ơ mục Destination, chọn đường dẫn đến folder sẽ chứa file backup: C:\BACKUP\diferential backup

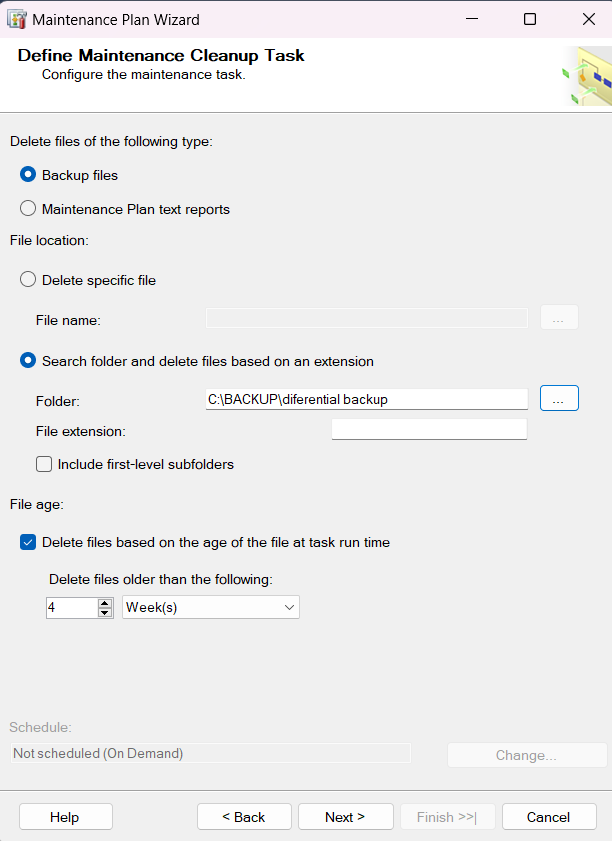


* Ở mục Options, chọn Compress backup để giảm dung lượng khi lưu vào ổ đĩa



4.7. Tiến hành Define Maintenance Cleanup Task:

* Chọn đường dẫn đến folder sẽ thực hiện Task Xóa các file differential-backup: C:\BACKUP\diferential backup
* Thiết lập thời gian xóa các file differential-backup: Thiết lập xóa các file differential-backup đã tồn tại 4 tuần



Sau khi đã thiết lập hết các bước trên, Click Finish Wizad để hoàn thành việc tạo lịch trình full-backup tự động

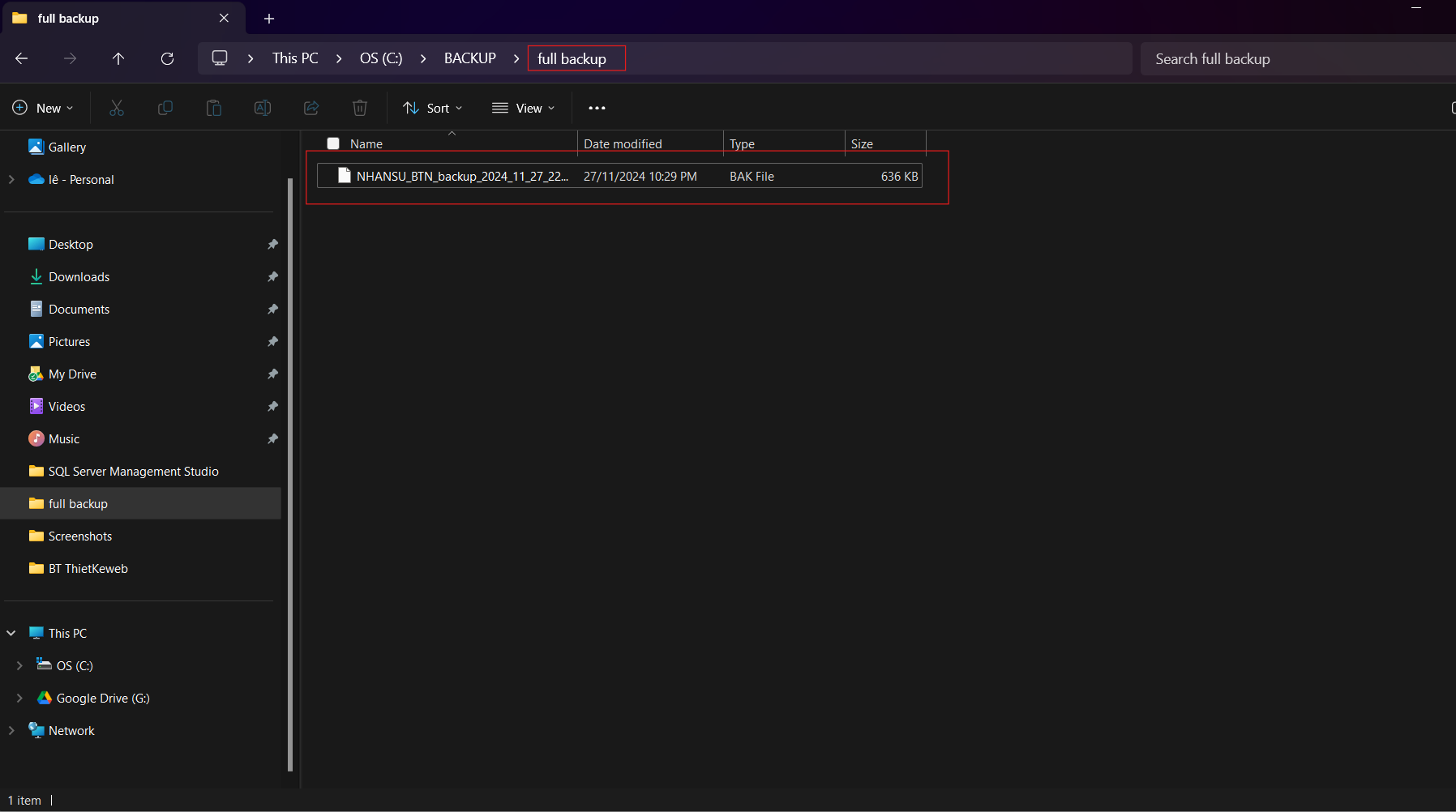
**Bước 5: Kiểm tra 2 lịch trình tự động backup vừa tạo có hoạt động không?**

Ở thư mục Jobs trong SQL Server Agent, tiến hành Start 2 job:

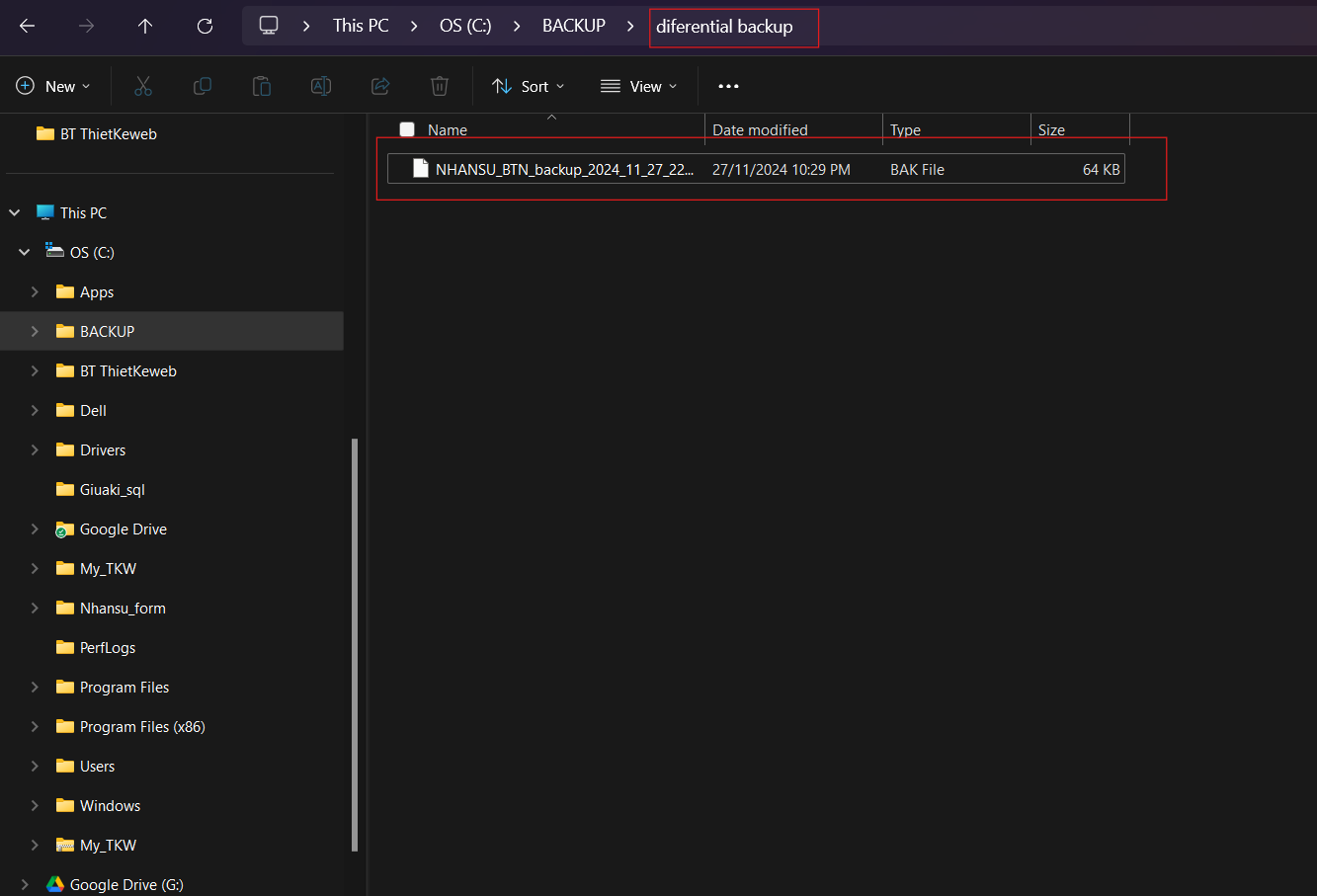
* full\_bk\_NHANSU.Subplan1
* diferential\_bk\_NHANSU.Subplan\_1

→ Kiểm tra 2 folder ful backup và differential backup đã tạo ở bước 1, nếu mỗi folder tồn tại 1 file tương ứng thì lịch trình tự động backup được tạo thành công.

* Folder full backup

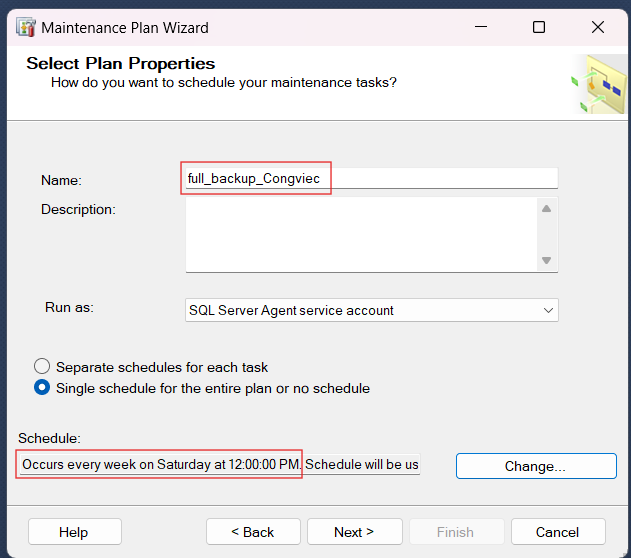


* Folder differential backup

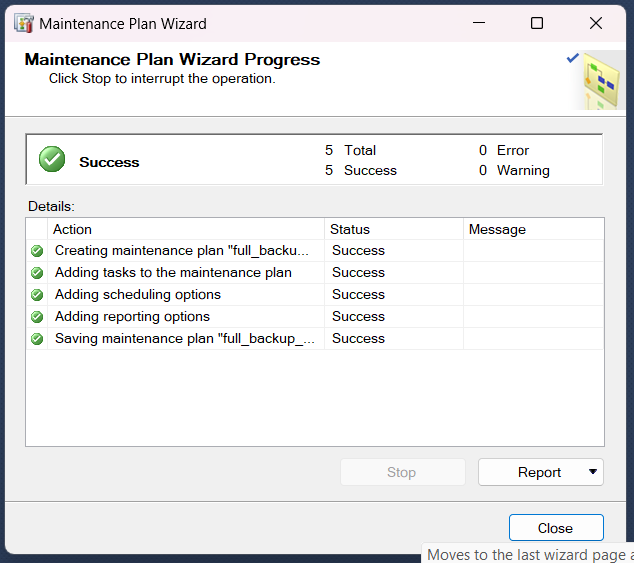


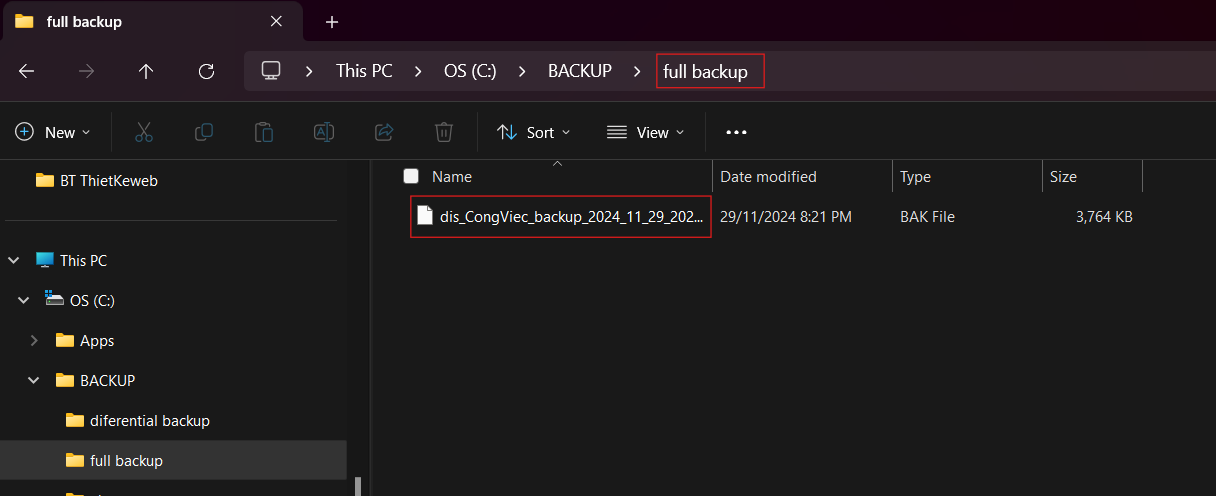
***Thực hiện tương tự các bước trên cho phần backup dữ liệu của 2 server là subcriber phân tán***

* **Bước 1: Tạo Full-Backup cho phân tán Công việc**

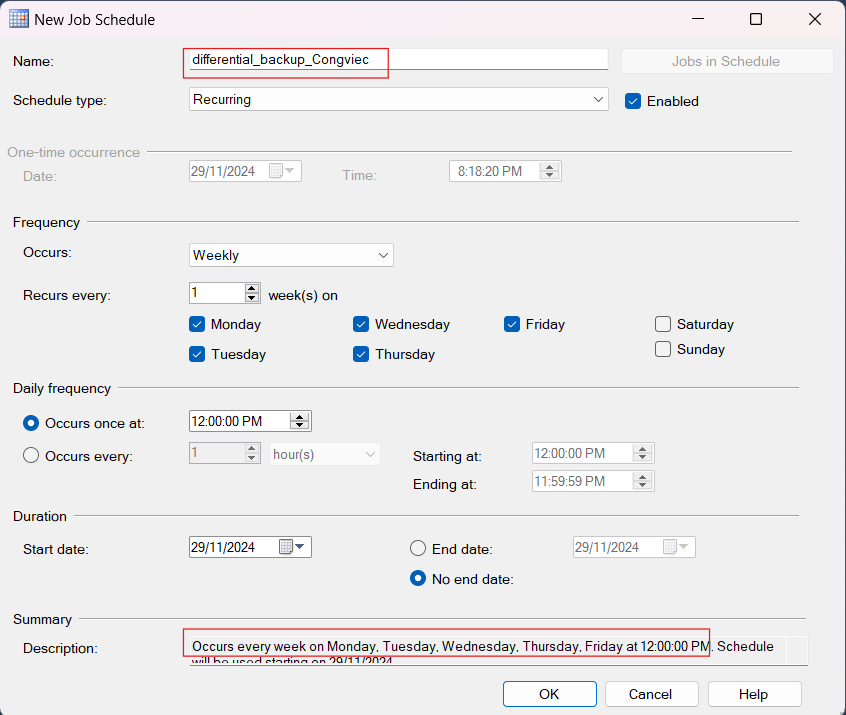


* **Tiến hành thực hiện các bước tiếp theo. Tạo lịch trình Full-Backup tự động thành công khi xuất hiện cửa sổ thông báo Success dưới đây:**





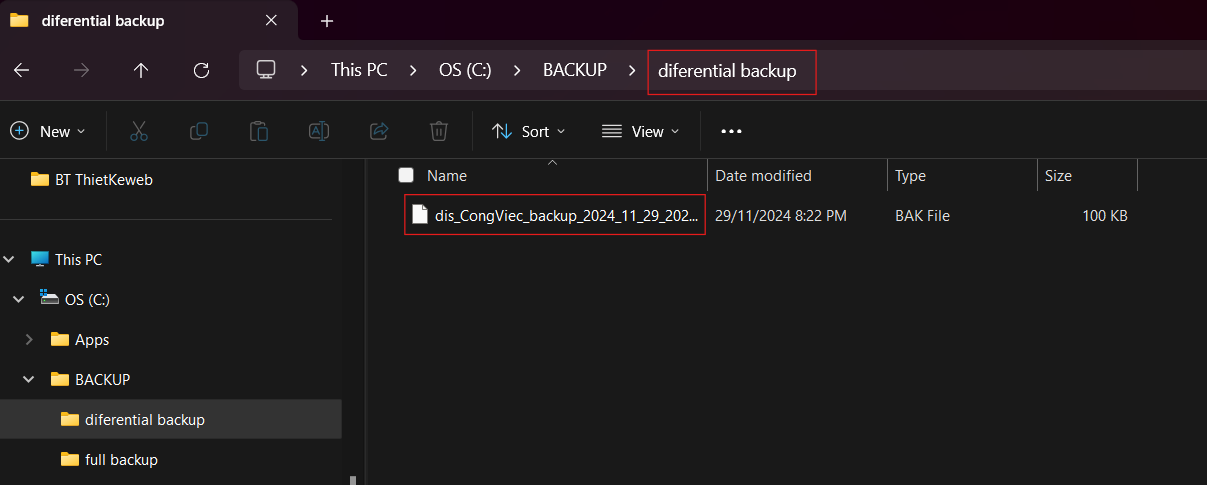
* Bước 2: Tạo Differential-Backup cho phân tán Công việc



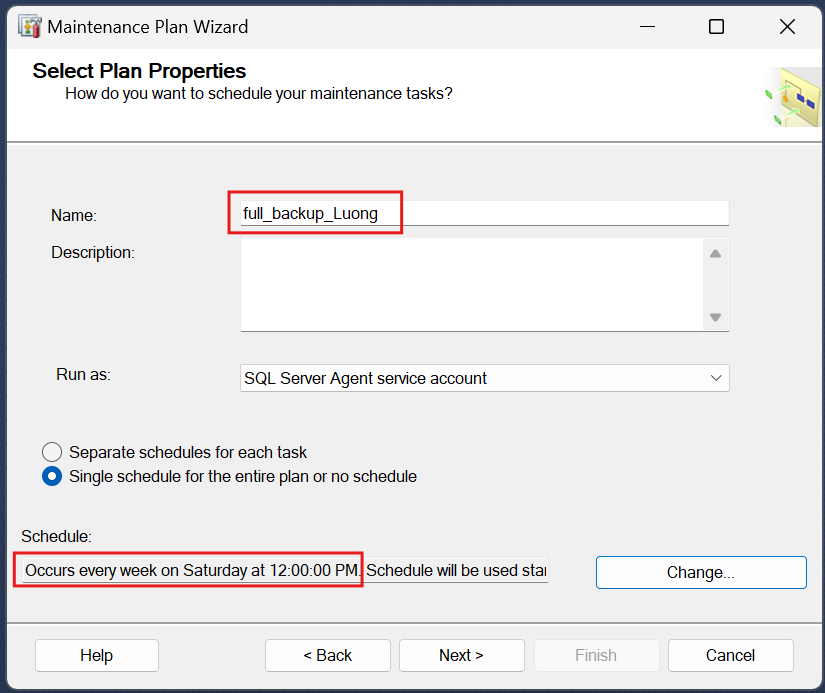
* Tiến hành thực hiện các bước tiếp theo. Tạo lịch trình Differential-Backup tự động thành công khi xuất hiện cửa sổ thông báo Success dưới đây:



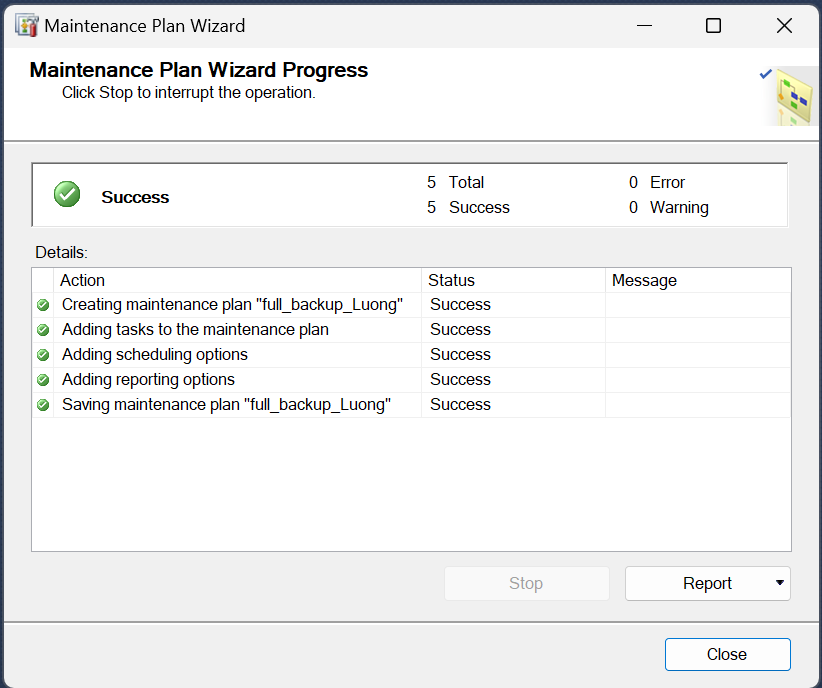
* Tiến hành chạy thử Job differential\_backup\_Congviec. Vào Folder differential backup đã tạo để kiểm tra. Nếu file differential - backup đã được lưu, thì Job differential\_backup\_Congviec vừa tạo đã thực thi được.



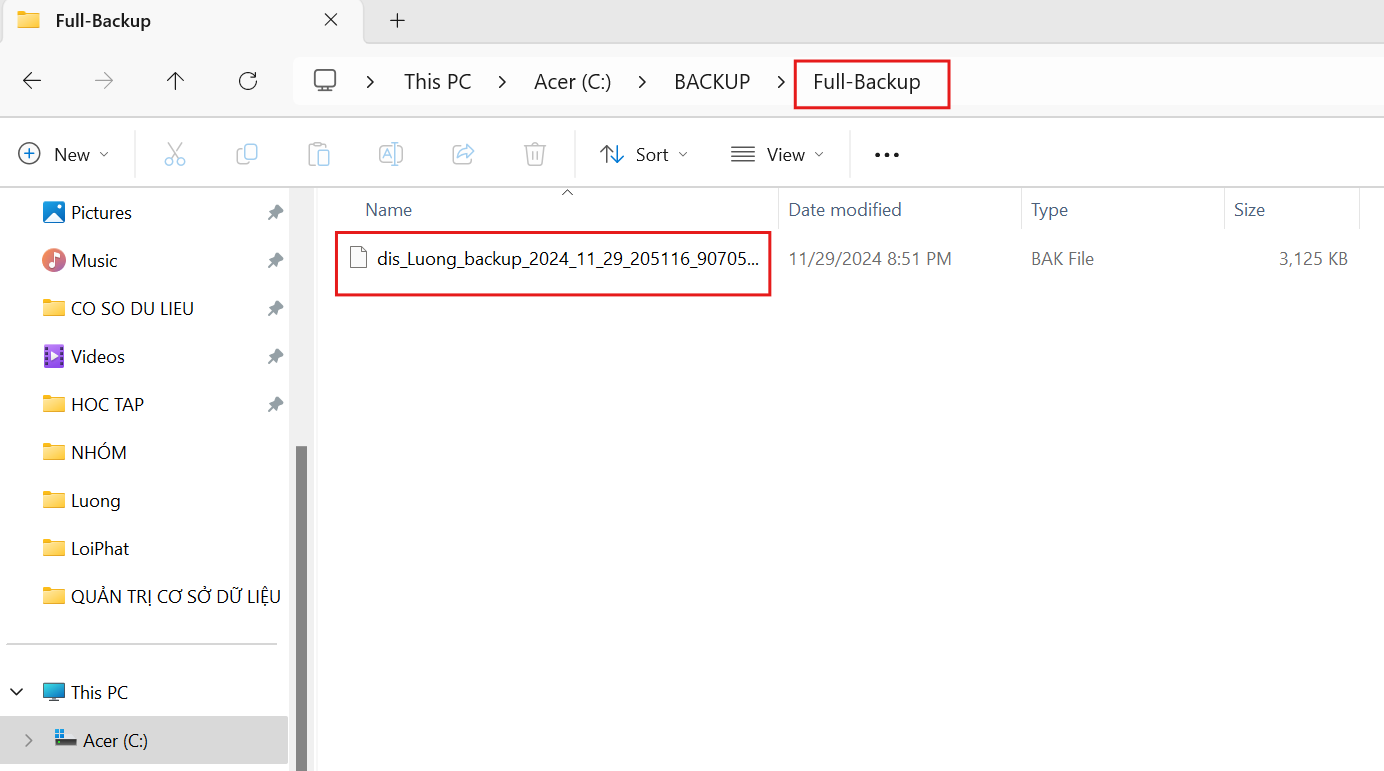
* Bước 3: Tạo full backup cho phân tán Lương



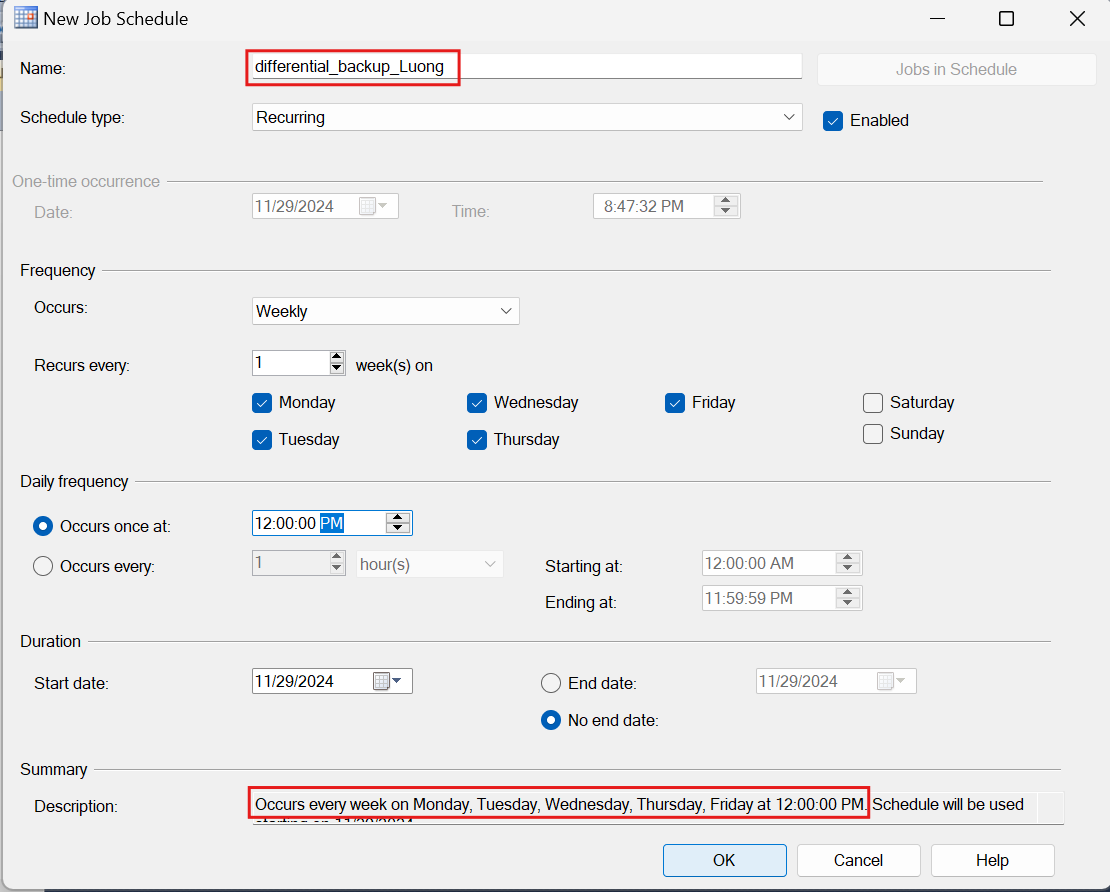
* Tiến hành thực hiện các bước tiếp theo. Tạo lịch trình Full-Backup tự động thành công khi xuất hiện cửa sổ thông báo Success dưới đây:



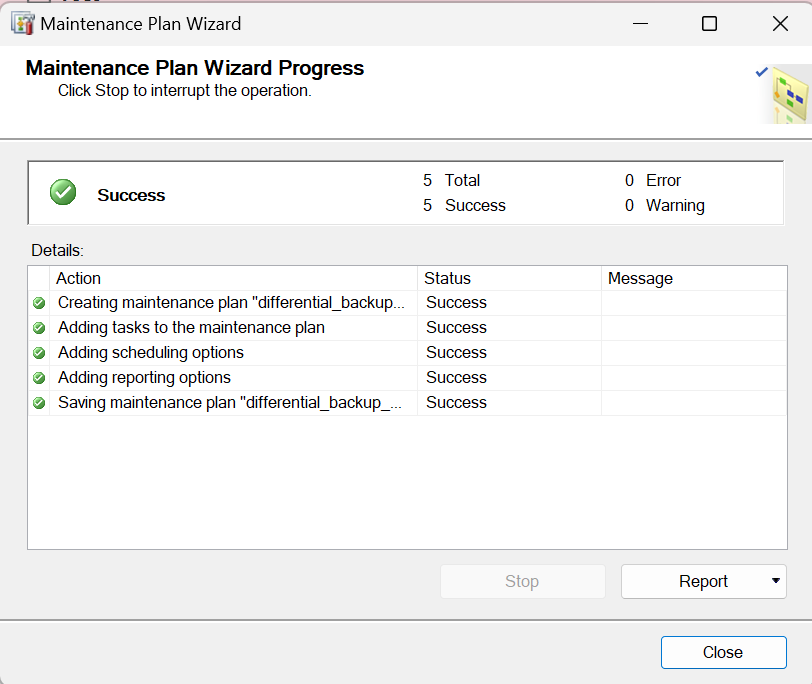
* Tiến hành chạy thử Job full\_backup\_Luong. Vào Folder full backup đã tạo để kiểm tra. Nếu file full - backup đã được lưu, thì Job full\_backup\_Luong vừa tạo đã thực thi được.



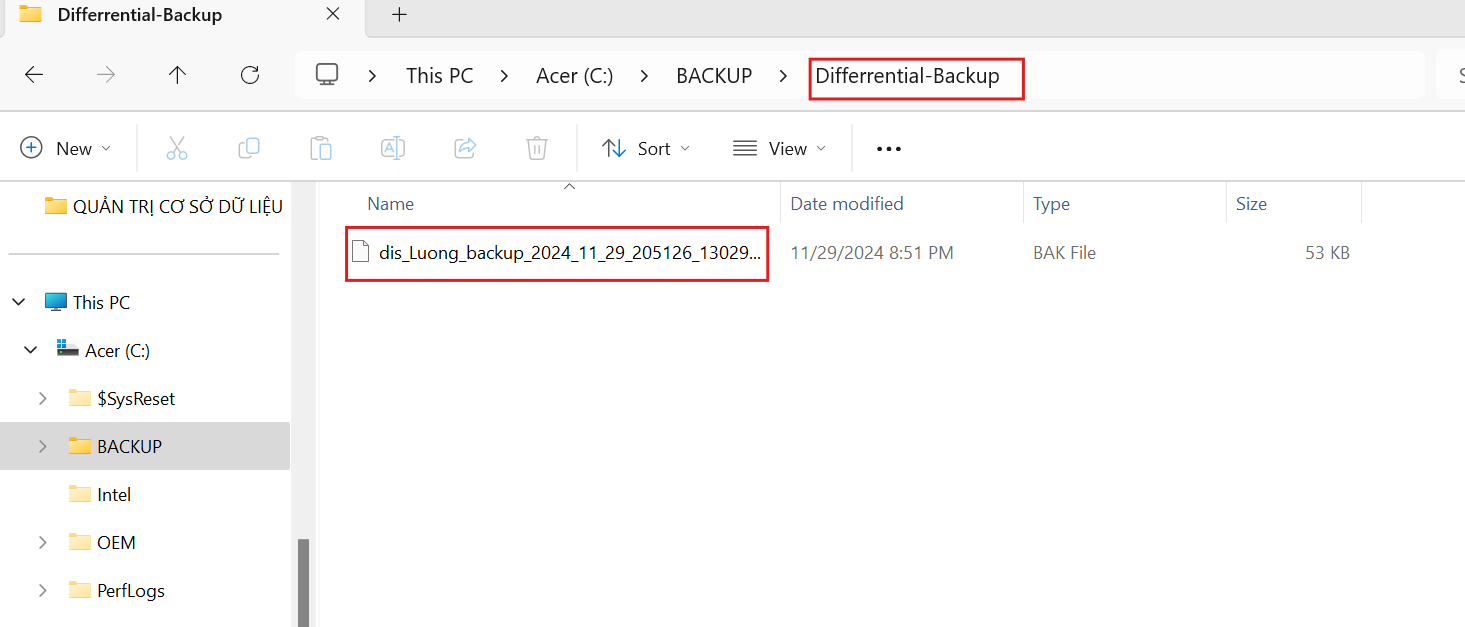
* Bước 4: Tạo Differential-Backup cho phân tán Lương



* Tiến hành thực hiện các bước tiếp theo. Tạo lịch trình Differential-Backup tự động thành công khi xuất hiện cửa sổ thông báo Success dưới đây:



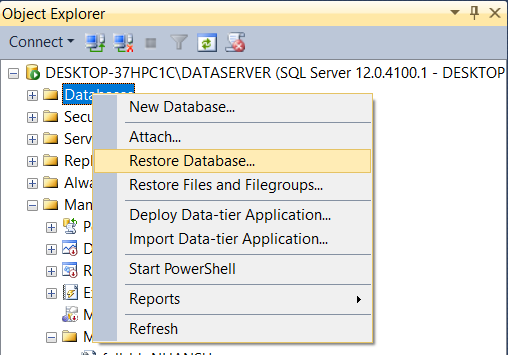
* Tiến hành chạy thử Job differential\_backup\_Luong. Vào Folder differential backup đã tạo để kiểm tra. Nếu file differential - backup đã được lưu, thì Job differential\_backup\_Luong vừa tạo đã thực thi được.



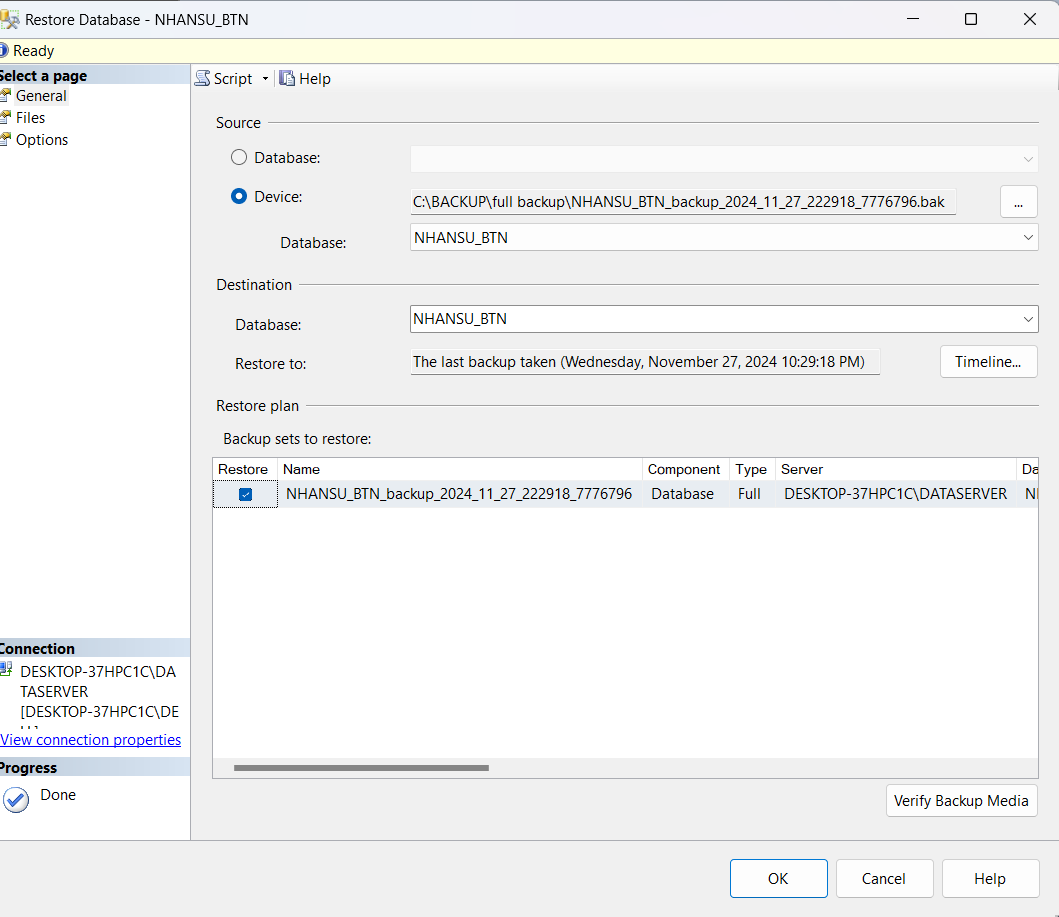
## RESTORE

Bước 1: Restore file Full-backup

1.1. Click chuột trái vào Database, chọn Restore Database



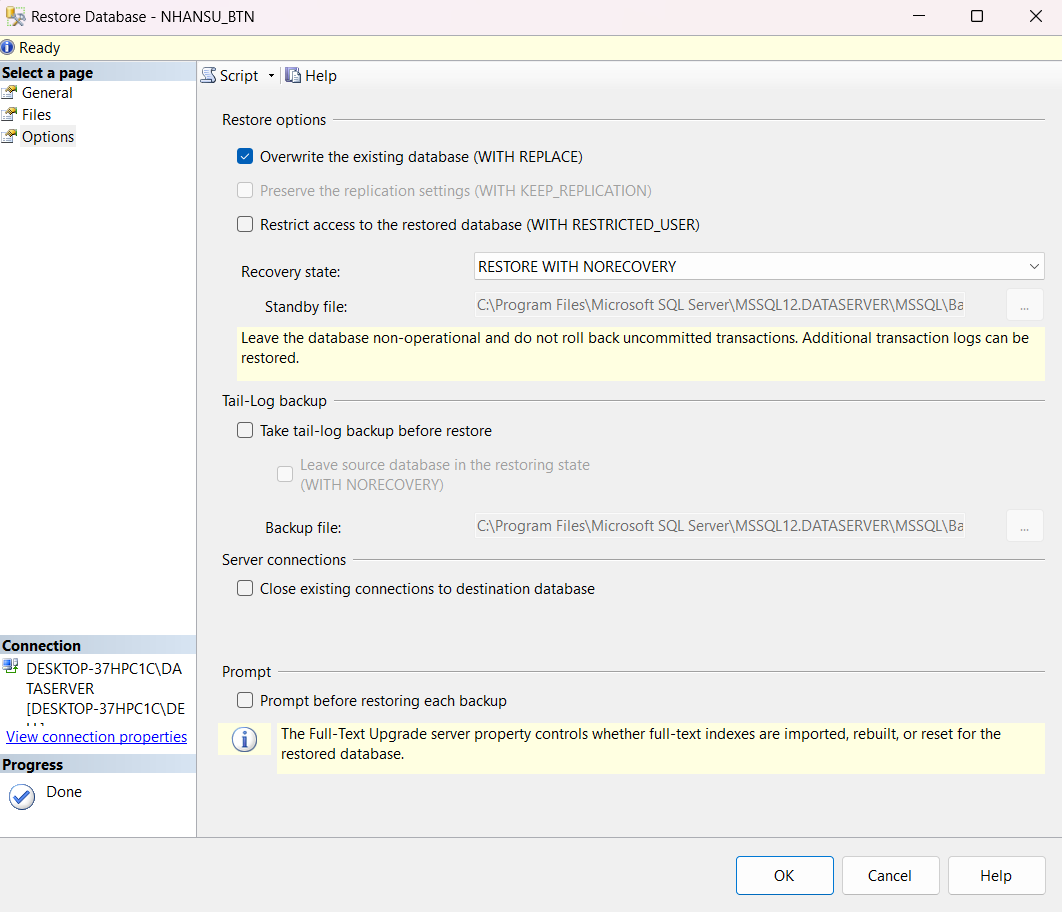
1.2. Tại General: Click vào Device, sau đó chọn đường dẫn đến file Full-backup đã lưu



1.3. Tại Option:

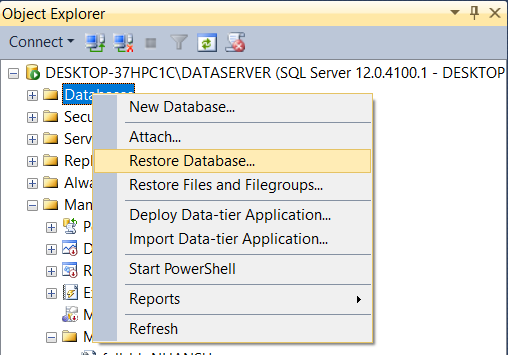
* Chọn Overwrite the existing database
* Ở mục Recovery state, chọn “RESTORE WITH NORECOVERY”

→ Sau đó, click OK

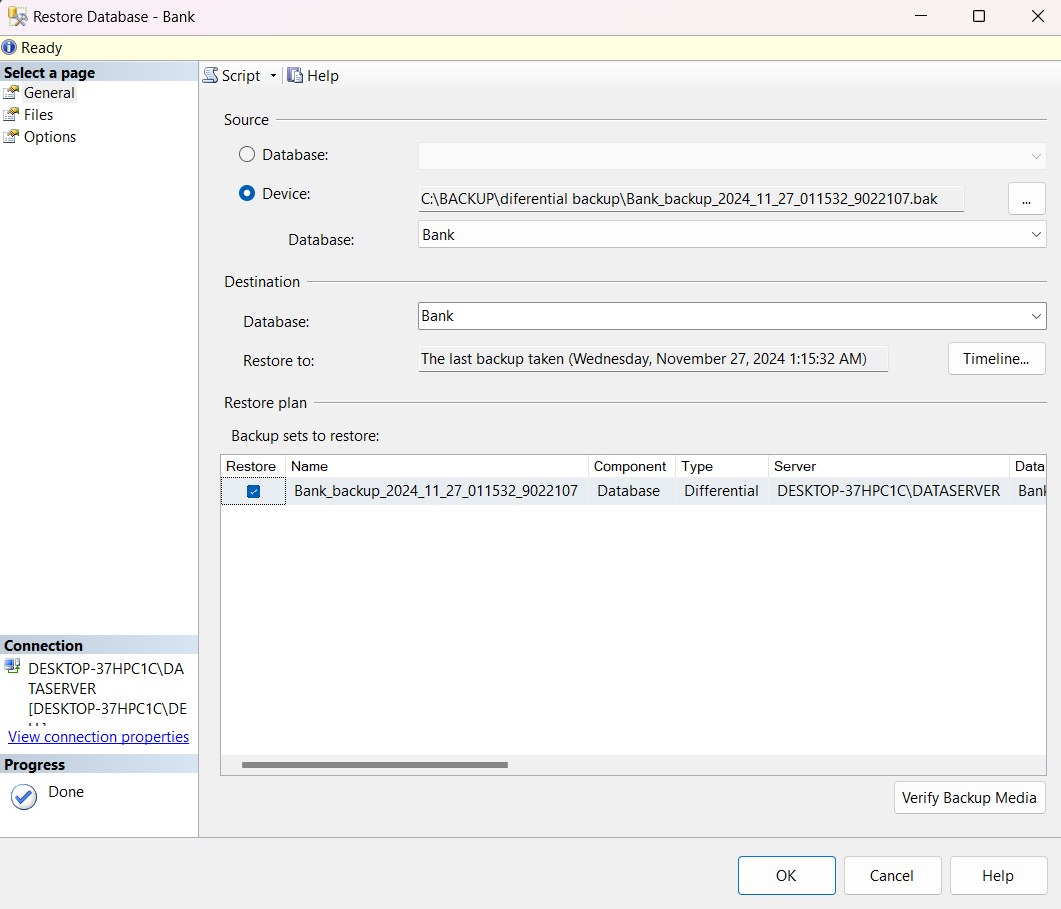


Bước 2:  Restore file Differential-Backup

2.1. Click chuột trái vào Database, chọn Restore Database



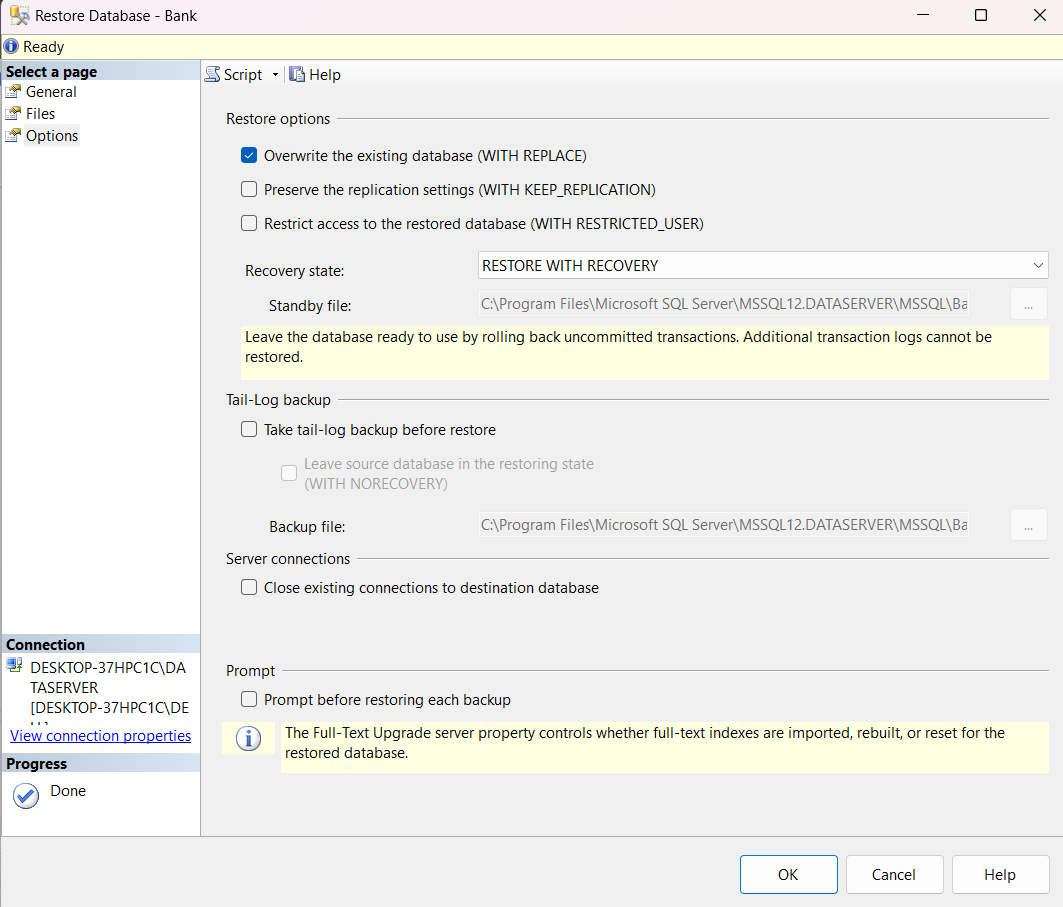
2.2. Tại General: Click vào Device, sau đó chọn đường dẫn đến file Differential-backup đã lưu



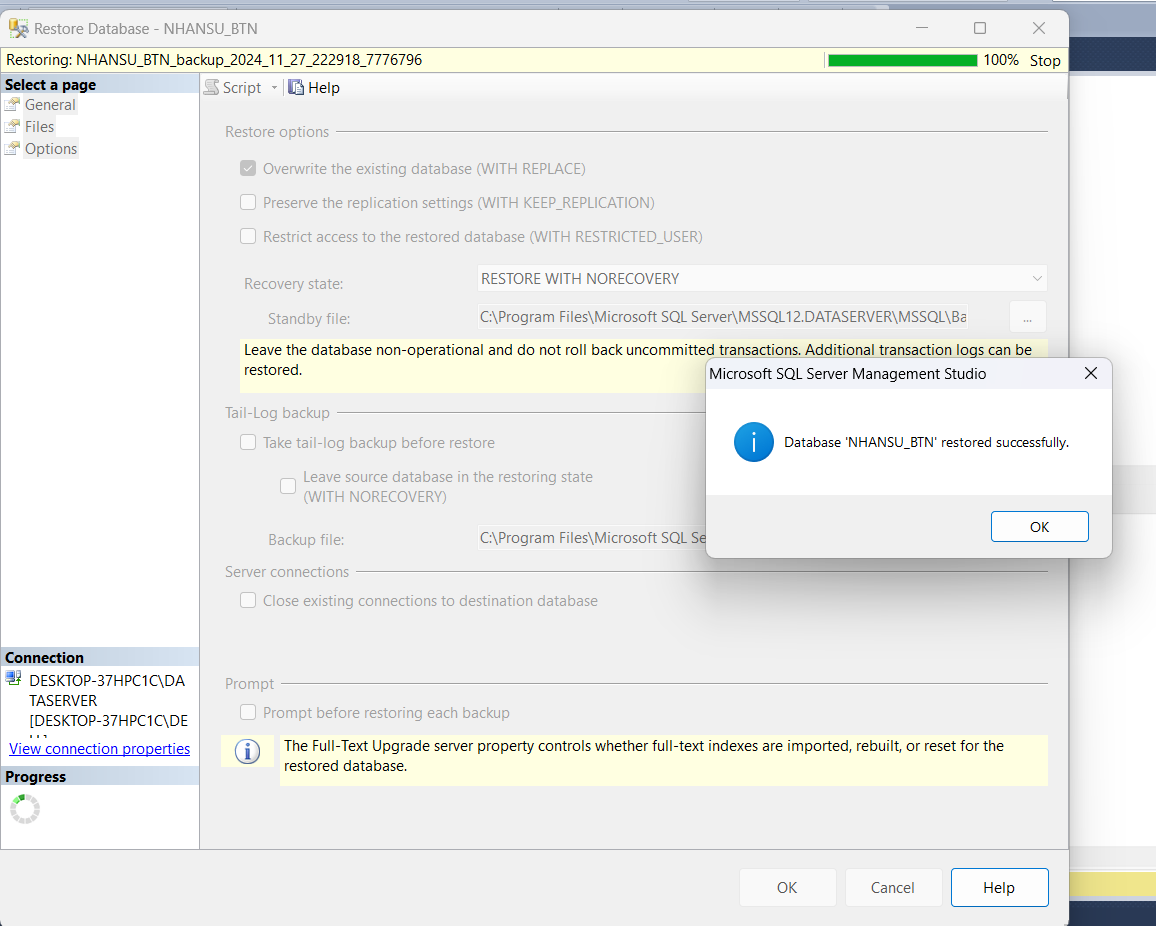
1.3. Tại Option:

* Chọn Overwrite the existing database
* Ở mục Recovery state, chọn “RESTORE WITH RECOVERY”

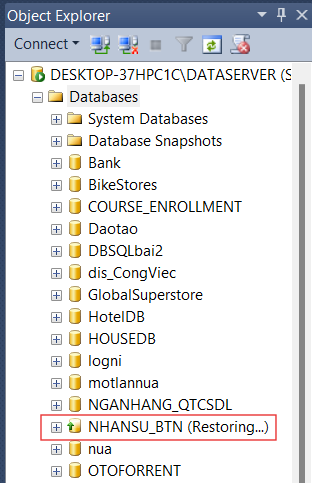
→ Sau đó, click OK



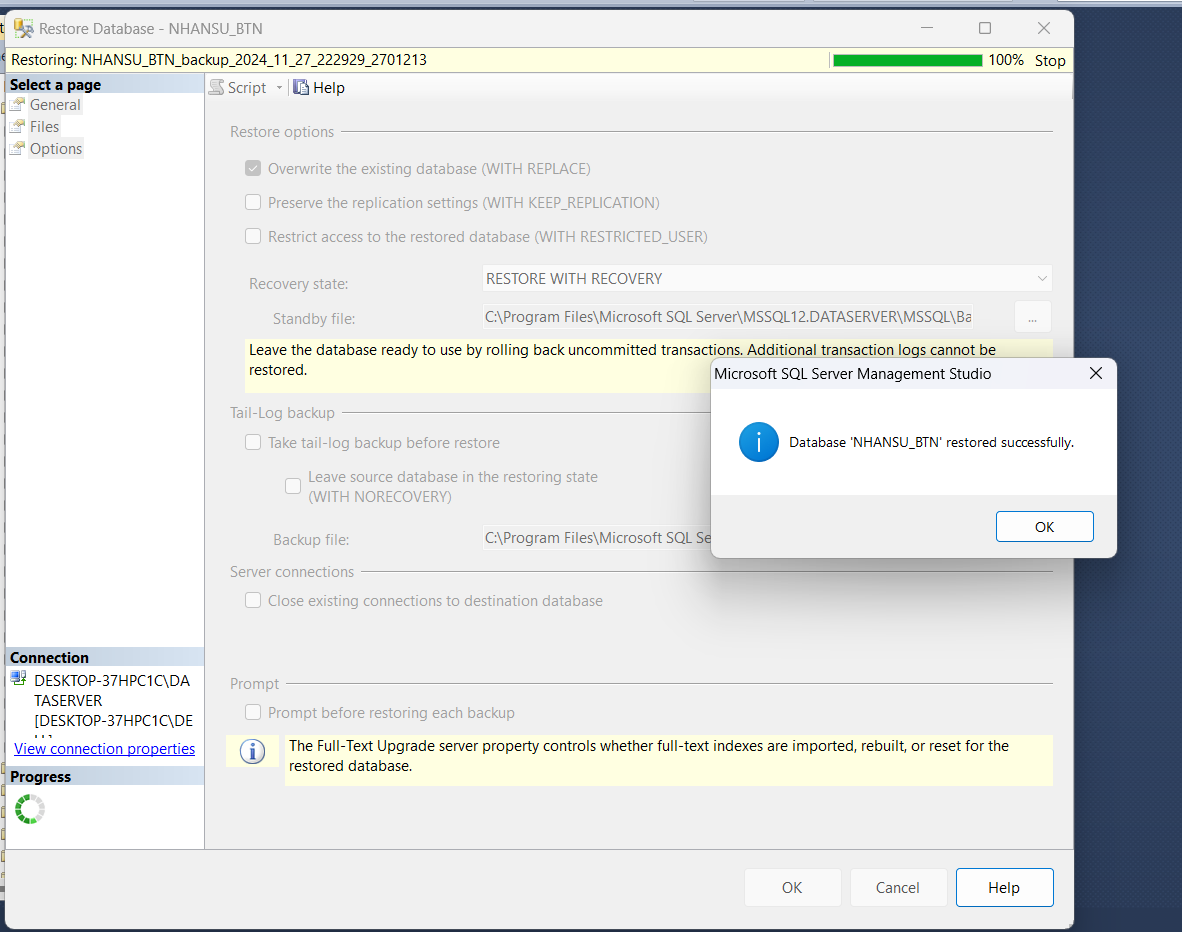
* **Sau khi hoàn thành tất cả bước trên thì đã restore thành công:**
* Ảnh khi đã restore file full BK



* Database NHANSU\_BTN khi đã restore file full backup



* Ảnh khi đã restore file Differential backup



Database NHANSU\_BTN khi đã restore file Differential Backup



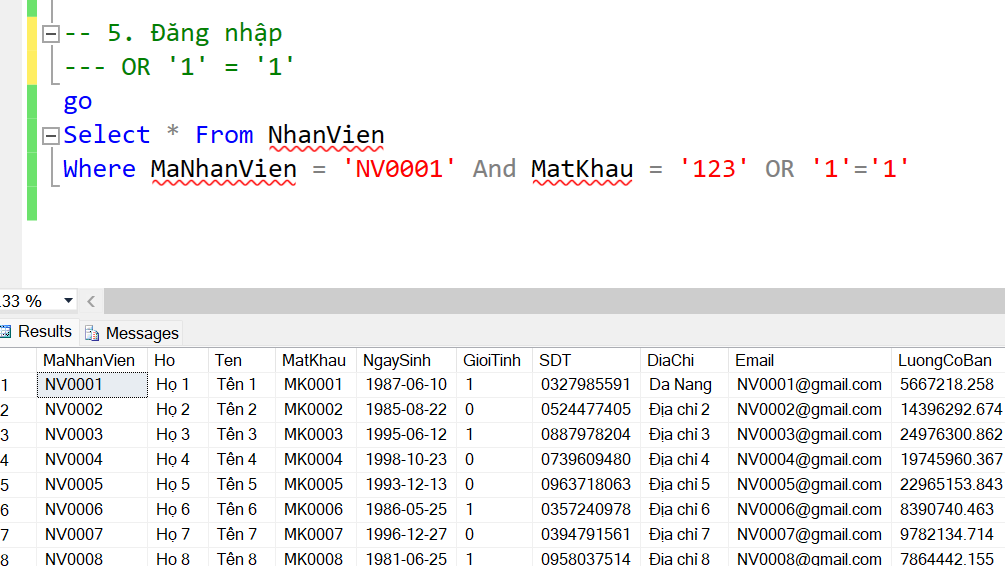
# STORED PROCEDURE` NGĂN CHẶN TẤN CÔNG SQL INJECTION:

SQL Injection là một trong những hình thức tấn công phổ biến và nguy hiểm nhất đối với các ứng dụng sử dụng cơ sở dữ liệu. Kẻ tấn công lợi dụng lỗ hổng trong việc xử lý dữ liệu đầu vào của ứng dụng để “tiêm” các câu lệnh SQL độc hại, từ đó thực hiện các thao tác trái phép như truy xuất, sửa đổi hoặc xóa dữ liệu, thậm chí chiếm quyền kiểm soát toàn bộ cơ sở dữ liệu. Các chức năng như đăng nhập, tìm kiếm, cập nhật, xóa hoặc tạo mới thường là mục tiêu chính của SQL Injection do chúng thường sử dụng các câu lệnh SQL để xử lý dữ liệu người dùng nhập vào.

Một giải pháp hiệu quả để ngăn chặn SQL Injection là sử dụng Stored Procedure. Stored Procedure giúp tách biệt logic ứng dụng và cơ sở dữ liệu, đồng thời tham số hóa dữ liệu, đảm bảo rằng dữ liệu đầu vào được xử lý an toàn trước khi thực thi. Ngoài việc bảo vệ hệ thống khỏi các cuộc tấn công, Stored Procedure còn giúp dễ dàng quản lý, bảo trì và tối ưu hóa hiệu năng của cơ sở dữ liệu.

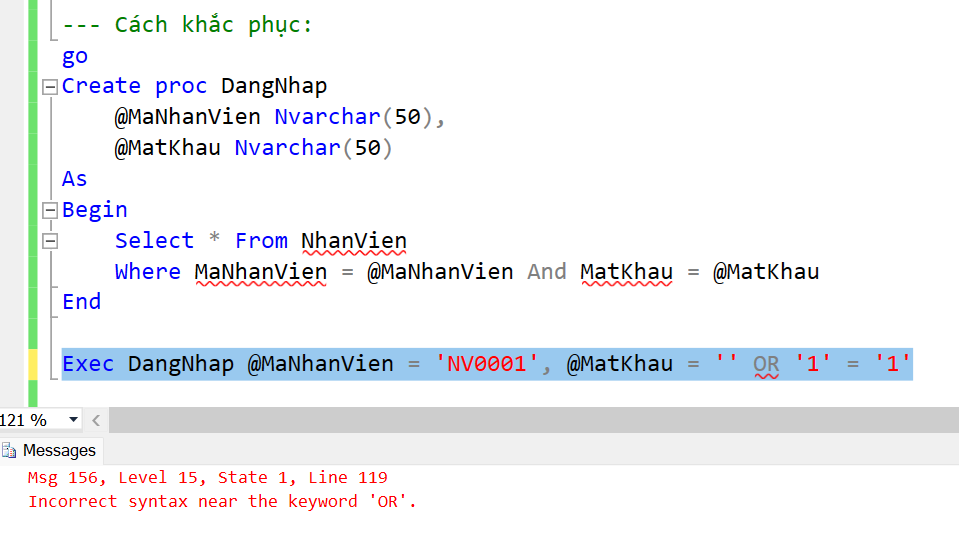
## Đăng nhập

* **Cách tấn công 1:**
* Khi người dùng nhập dữ liệu độc hại vào trường MaNhanVien hoặc MatKhau, hệ thống không kiểm tra mà chèn trực tiếp vào câu truy vấn. Ví dụ:



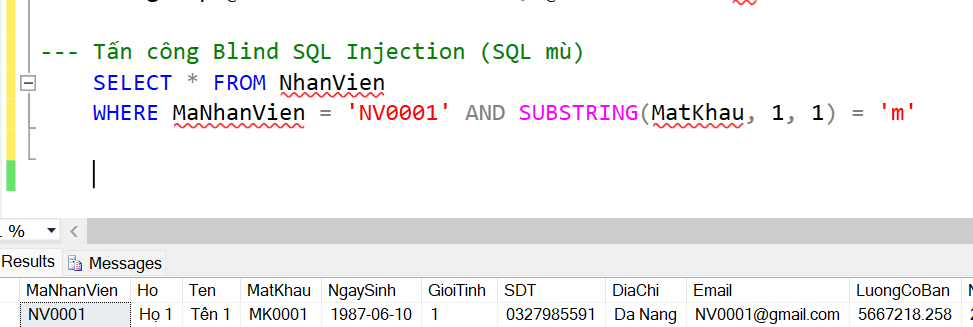
Phần OR '1'='1' luôn trả về True, dẫn đến việc điều kiện Where không còn ý nghĩa và trả về toàn bộ dữ liệu bảng NhanVien.

* **Cách khắc phục:**



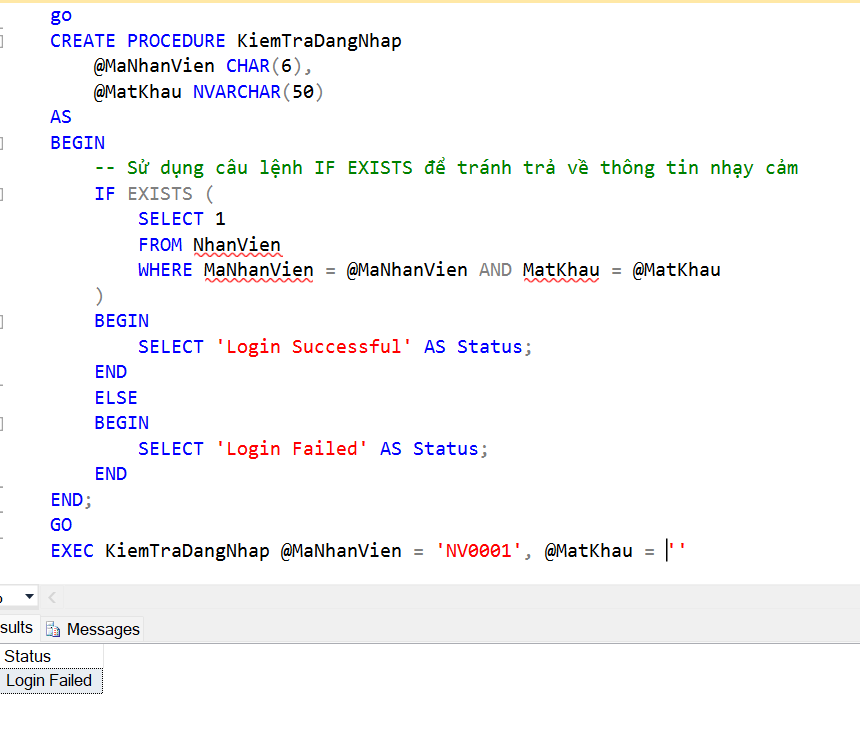
Khi sử dụng Stored procedure cho DangNhap thì không thể tiêm lệnh **OR '1'='1'** vào thủ tục được vì khi sử dụng tham số (@MaNhanVien và @MatKhau) trong Stored Procedure, SQL Server sẽ tự động xử lý và mã hóa giá trị đầu vào, tránh việc ghép chuỗi trực tiếp, do đó không thể tiêm lệnh **OR '1'='1'**

* Cách tấn công 2:
* Tấn công Blind SQL Injection (SQL mù). Blind SQL Injection (SQL mù) là một kiểu tấn công mà kẻ tấn công không nhận được phản hồi trực tiếp từ cơ sở dữ liệu (như dữ liệu trả về), nhưng dựa vào các phản hồi đúng/sai để suy đoán dữ liệu nhạy cảm.



Với cách tấn công này, kẻ tấn công sẽ thử từng ký tự của mật khẩu bằng cách thay đổi câu truy vấn và dựa trên phản hồi để "đoán" được mật khẩu. Nếu hệ thống phản hồi là "đúng" (có kết quả), nghĩa là ký tự đầu tiên là ‘m’. Kẻ tấn công sẽ lặp lại với từng ký tự trong bảng chữ cái (a-z, A-Z, 0-9, ký tự đặc biệt, ...) và thay đổi vị trí trong mật khẩu, bằng cách này, chúng có thể lấy được toàn bộ mật khẩu.

* **Cách khắc phục:**



Tham số hóa ngăn người dùng chèn mã độc. Kết quả trả về chỉ là trạng thái đăng nhập, không tiết lộ thông tin chi tiết.

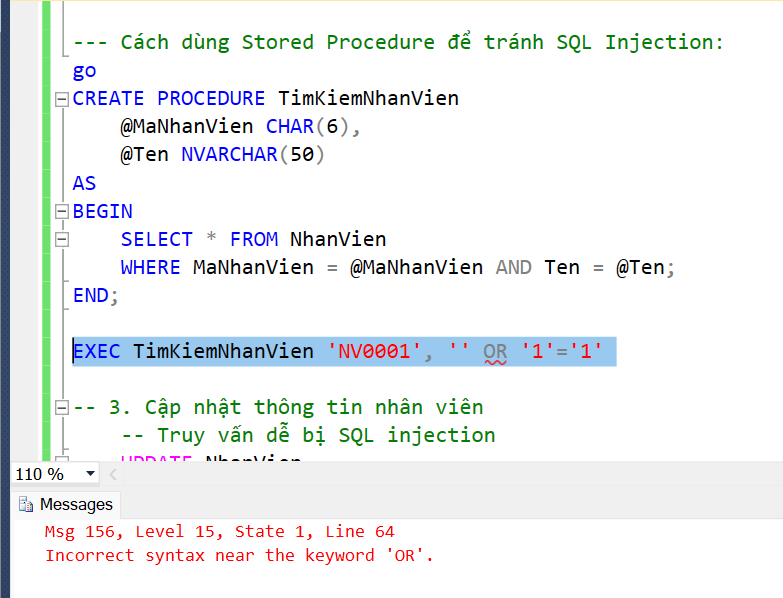
## Tìm kiếm thông tin nhân viên

* **Truy vấn dễ bị SQL injection:**

**

Khi kẻ tấn công chèn Ten là ***‘ OR ‘1’ = ‘1,*** vì phần OR '1'='1' luôn trả về True, do đó tất cả bản ghi của bảng NhanVien sẽ **hiển thị, dẫn đến** việc lộ toàn bộ thông tin như mật khẩu, số tài khoản và các thông tin quan trọng khác của toàn bộ nhân viên. Điều này có thể bị lợi dụng để thực hiện các hành vi bất hợp pháp như giả mạo, tống tiền, hoặc xâm phạm quyền riêng tư của nhân viên, gây tổn hại đến uy tín và an ninh của tổ chức.

* **Cách khắc phục:**

**

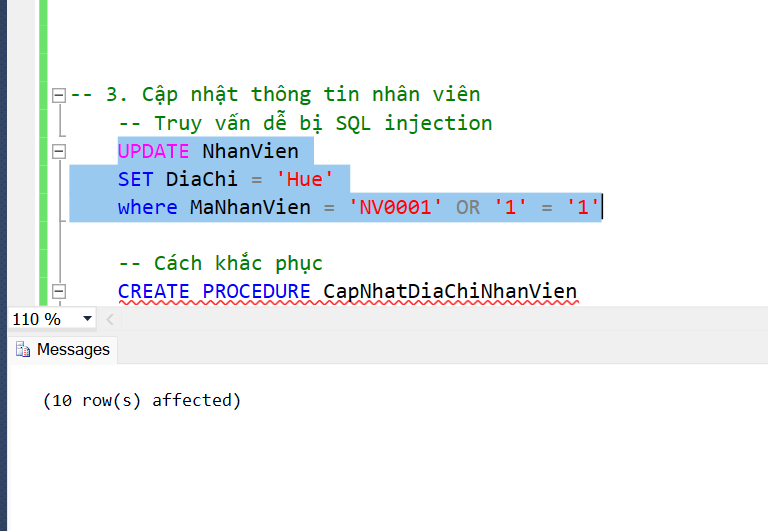
Với cách làm này, Stored Procedures thường nhận dữ liệu đầu vào dưới dạng tham số, giúp phân tách rõ ràng giữa câu lệnh SQL và dữ liệu. Khi dữ liệu đầu vào được xử lý qua tham số, hệ thống cơ sở dữ liệu tự động mã hóa dữ liệu, ngăn không cho các ký tự đặc biệt (như dấu ' hoặc ;) gây ra ảnh hưởng đến câu truy vấn SQL

Khi người dùng cố chèn các giá trị MaNhanVien và Ten không hợp lệ, hàm EXEC sẽ trả về thông báo Lỗi

## Cập nhật thông tin nhân viên

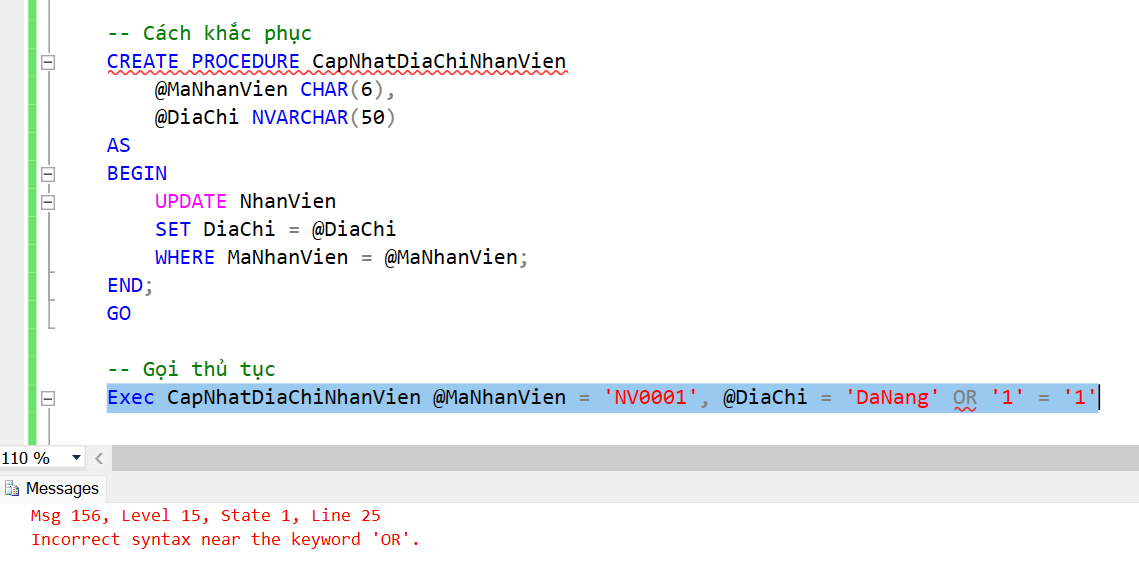
* **Truy vấn dễ bị SQL injection:**

Khi giá trị của MaNhanVien hoặc DiaChi được chèn từ đầu vào của người dùng mà không có kiểm tra, kẻ tấn công có thể cung cấp đầu vào độc hại như sau:



Khi kẻ tấn công chèn MaNhanVien là ***NV0001’ OR ‘1’ = ‘1,*** vì phần OR '1'='1' luôn trả về True, do đó tất cả bản ghi của bảng NhanVien sẽ bị **update toàn bộ,** dẫn đến  việc thay đổi không mong muốn dữ liệu trên toàn bộ bảng, làm sai lệch dữ liệu và có thể gây hỏng hóc hệ thống nếu bảng có vai trò quan trọng. Kết quả là câu lệnh trên sẽ xóa toàn bộ bảng Nhân viên

* **Cách khắc phục**

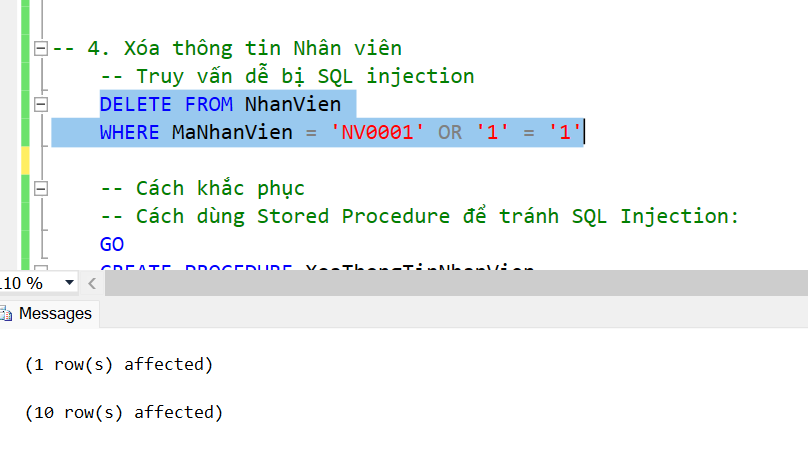
****

Với cách làm này, Stored Procedures thường nhận dữ liệu đầu vào dưới dạng tham số, giúp phân tách rõ ràng giữa câu lệnh SQL và dữ liệu. Khi dữ liệu đầu vào được xử lý qua tham số, hệ thống cơ sở dữ liệu tự động mã hóa dữ liệu, ngăn không cho các ký tự đặc biệt (như dấu ' hoặc ;) gây ra ảnh hưởng đến câu truy vấn SQL.

Khi người dùng cố chèn các giá trị MaNhanVien và Diachi không hợp lệ, hàm EXEC sẽ trả về thông báo Lỗi.

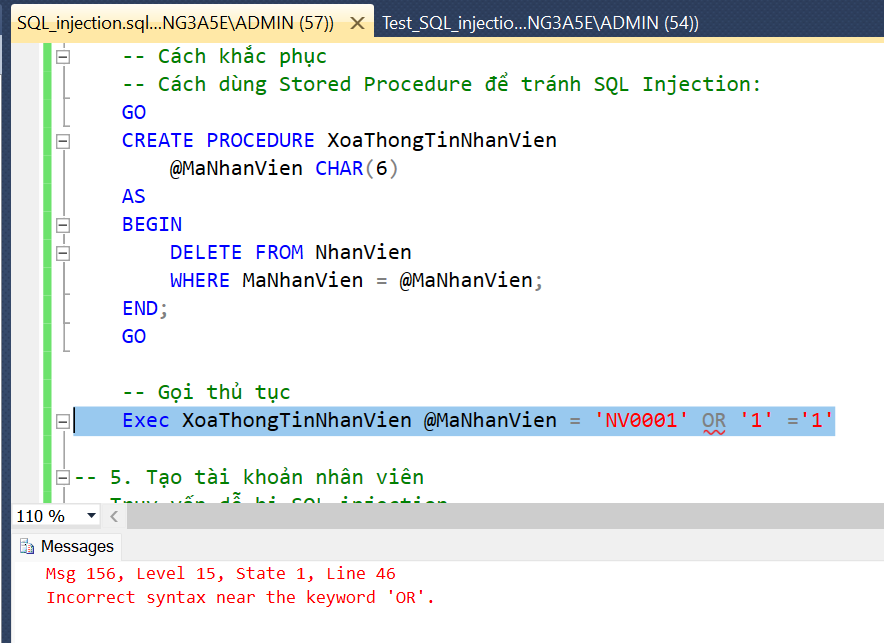
## Xóa thông tin Nhân viên

* **Truy vấn dễ bị SQL injection:**

****

Khi kẻ tấn công chèn MaNhanVien là ***NV0001’ OR ‘1’ = ‘1,*** vì phần OR '1'='1' luôn trả về True, do đó tất cả bản ghi của bảng NhanVien sẽ bị **Xóa** , dẫn đến  việc mất toàn bộ dữ liệu quan trọng của hệ thống, bao gồm thông tin nhân viên, mật khẩu, địa chỉ, và các thông tin cần thiết khác. Điều này có thể làm gián đoạn hoạt động của tổ chức, gây thiệt hại lớn về tài chính và uy tín, đồng thời mất dữ liệu quan trọng mà không thể phục hồi nếu không có sao lưu thích hợp.

* **Cách khắc phục:**



Với cách làm này, Stored Procedures thường nhận dữ liệu đầu vào dưới dạng tham số, giúp phân tách rõ ràng giữa câu lệnh SQL và dữ liệu. Khi dữ liệu đầu vào được xử lý qua tham số, hệ thống cơ sở dữ liệu tự động mã hóa dữ liệu, ngăn không cho các ký tự đặc biệt (như dấu ' hoặc ;) gây ra ảnh hưởng đến câu truy vấn SQL.

Khi người dùng cố chèn giá trị MaNhanVien không hợp lệ, hàm EXEC sẽ trả về thông báo Lỗi.