**TRƯỜNG ĐẠI HỌC KINH TẾ**

**ĐẠI HỌC ĐÀ NẴNG**

**KHOA THỐNG KÊ – TIN HỌC**



**BÁO CÁO TIẾN ĐỘ**

**Môn:** Quản trị cơ sở dữ liệu

***GVHD****:* Cao Thị Nhâm

***SVTH****:* 48K21.2-Nhóm 10

Hà Thị Quỳnh Anh

Trương Thị Mỹ Hồng

Nguyễn Trúc Lam

Nguyễn Châu My

Lê Thị Cẩm Tú

*Đà Nẵng 10/2024*

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH i](#_Toc183784471)

[1 Giai đoạn R1-R2: Sưu tất cả các loại hoá đơn, chứng từ tại một cơ sở kinh doanh/sản xuất nhỏ và thiết kế cơ sở dữ liệu 1](#_Toc183784472)

[1.1 Thiết kế cơ sở dữ liệu mức khái niệm 1](#_Toc183784473)

[1.1.1 Hoá đơn nhập hàng tạp hoá cô Châu: 1](#_Toc183784474)

[1.1.2 Hoá đơn bán hàng tạp hoá cô Châu 4](#_Toc183784475)

[1.1.3 Tích hợp ER-1 và ER-2 7](#_Toc183784476)

[1.2 Thiết kế cơ sở dữ liệu mức logic 7](#_Toc183784477)

[1.3 Thiết kế chi tiết các bảng 8](#_Toc183784478)

[1.3.1 Bảng Nhà cung cấp 8](#_Toc183784479)

[1.3.2 Bảng Nhập hàng 9](#_Toc183784480)

[1.3.3 Bảng Hóa đơn nợ NCC 9](#_Toc183784481)

[1.3.4 Bảng Nhập Hàng\_ChiTiet 10](#_Toc183784482)

[1.3.5 Bảng Hàng 10](#_Toc183784483)

[1.3.6 Bảng Bán Hàng\_Chi tiết 10](#_Toc183784484)

[1.3.7 Bảng Bán hàng 10](#_Toc183784485)

[1.3.8 Bảng Khách hàng 11](#_Toc183784486)

[1.3.9 Bảng Hóa đơn KH nợ 11](#_Toc183784487)

[1.3.10 Bảng Users 11](#_Toc183784488)

[2 Giai đoạn R3: Xây dựng cơ sở dữ liệu cho bản thiết kế 12](#_Toc183784489)

[3 Giai đoạn R4: Tạo dữ liệu dump cho các bảng và các module trong cơ sở dữ liệu để phục vụ các thao tác xử lý dữ liệu 12](#_Toc183784490)

[3.1 Tạo dữ liệu dump 12](#_Toc183784491)

[3.2 10 module trong cơ sở dữ liệu để phục vụ các thao tác xử lý dữ liệu 12](#_Toc183784492)

[3.2.1 Câu 1 – Module 1: 12](#_Toc183784493)

[3.2.2 Câu 2 – Module 2: 12](#_Toc183784494)

[3.2.3 Câu 3 – Module 3: 13](#_Toc183784495)

[3.2.4 Câu 4 – Module 4: 14](#_Toc183784496)

[3.2.5 Câu 5 – Module 5: 15](#_Toc183784497)

[3.2.6 Câu 6 – Module 6: 15](#_Toc183784498)

[3.2.7 Câu 7 – Module 7: 16](#_Toc183784499)

[3.2.8 Câu 8 – Module 8: 16](#_Toc183784500)

[3.2.9 Câu 9 – Module 9: 17](#_Toc183784501)

[3.2.10 Câu 10 – Module 10: 18](#_Toc183784502)

[4 Giai đoạn R5: Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu 18](#_Toc183784503)

[4.1 Thiết lập cơ chế authentication phù hợp 18](#_Toc183784504)

[4.1.1 Lý do lựa chọn phân quyền 18](#_Toc183784505)

[4.1.2 Các bước triển khai 18](#_Toc183784506)

[4.2 Cơ chế bảo mật các dữ liệu nhạy cảm 21](#_Toc183784507)

[4.2.1 Mã hoá dữ liệu 21](#_Toc183784508)

[4.2.2 Giải mã dữ liệu 24](#_Toc183784509)

[4.2.3 Thêm dữ liệu 26](#_Toc183784510)

[5 Giai đoạn R6: Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu 29](#_Toc183784511)

[5.1 Lý do chọn loại sao lưu: Full backup 29](#_Toc183784512)

[5.2 Các bước sao lưu dữ liệu 29](#_Toc183784513)

[5.3 Thiết lập thời gian thực hiện công việc sao lưu tự động 33](#_Toc183784514)

[5.4 Các bước khôi phục dữ liệu 38](#_Toc183784515)

[6 Giai đoạn R7: Phương án giải pháp cho vấn đề dữ liệu lớn, không đủ lưu trữ trong một ổ đĩa 41](#_Toc183784516)

[6.1 Lý do lựa chọn giải pháp 41](#_Toc183784517)

[6.2 Thực hiện triển khai 42](#_Toc183784518)

[6.2.1 Đưa dữ liệu lên Microsoft Azure 42](#_Toc183784519)

[6.2.2 Phân quyền 46](#_Toc183784520)

[6.2.3 Backup dữ liệu trên sql server 48](#_Toc183784521)

[6.2.4 Backup tự động trên Azure Microsoft 50](#_Toc183784522)

[7 Giai đoạn R8: Phương án hạn chế tấn công SQL Injection 53](#_Toc183784523)

[7.1 Khái niệm 53](#_Toc183784524)

[7.2 Cách phòng chống 53](#_Toc183784525)

[7.3 Các cách tấn công 53](#_Toc183784526)

[7.3.1 Tấn công vào chức năng đăng nhập 53](#_Toc183784527)

[7.3.2 Tấn công vào chức năng Select (Tìm kiếm) 55](#_Toc183784528)

[7.3.3 Tấn công vào chức năng Update (Cập nhật/Chỉnh sửa) 56](#_Toc183784529)

[7.3.4 Tấn công vào chức năng Delete (Xoá) 62](#_Toc183784530)

[8 Ghi chú: Thay đổi về thiết kế 67](#_Toc183784531)

[8.1 Thay đổi R3 67](#_Toc183784532)

[8.1.1 Thay đổi các kiểu dữ liệu ở thiết kế chi tiết các bảng trong R2 67](#_Toc183784533)

[8.2 Thay đổi R5 69](#_Toc183784534)

[8.2.1 Thêm bảng Users vào thiết kế chi tiết các bảng ở R2 69](#_Toc183784535)

# DANH MỤC HÌNH ẢNH

[Hình 1 Hoá đơn nhập hàng tạp hoá cô Châu 1](#_Toc183098869)

[Hình 2 Sơ đồ ER-1 3](#_Toc183098870)

[Hình 3 Hoá đơn bán hàng tạp hoá cô Châu 5](#_Toc183098871)

[Hình 4 Sơ đồ ER-2 6](#_Toc183098872)

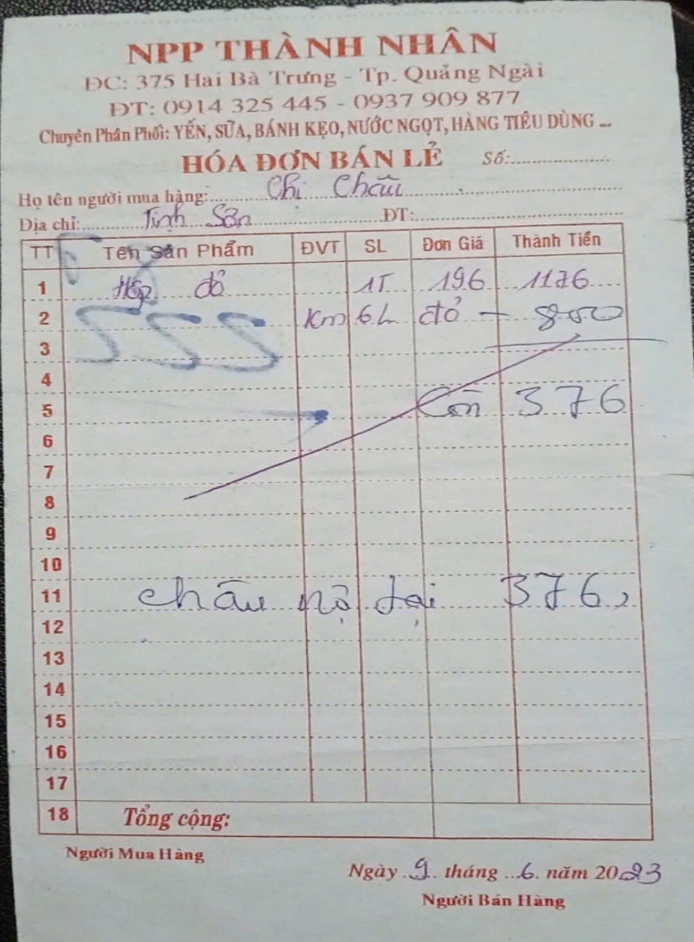
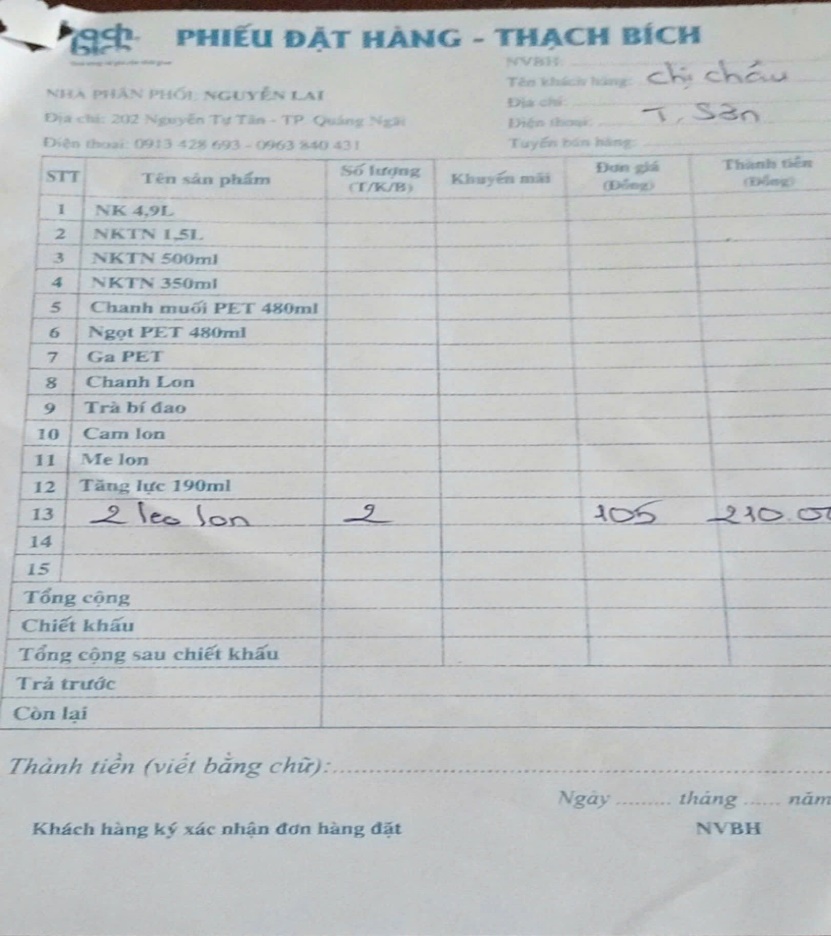
[Hình 5 Sơ đồ ER 7](#_Toc183098873)

[Hình 6 Sơ đồ quan hệ 8](#_Toc183098874)

# Giai đoạn R1-R2: Sưu tất cả các loại hoá đơn, chứng từ tại một cơ sở kinh doanh/sản xuất nhỏ và thiết kế cơ sở dữ liệu

## Thiết kế cơ sở dữ liệu mức khái niệm

### Hoá đơn nhập hàng tạp hoá cô Châu:

*Link hóa đơn:* <https://drive.google.com/drive/u/1/folders/1SInuTownU9BnNzqWW8nB54REbJItHnAk>

Hình 1 Hoá đơn nhập hàng tạp hoá cô Châu

**Bước 1: Chọn lọc thông tin**

|  |  |  |
| --- | --- | --- |
| **Từ trong HDSD** | **Từ rõ nghĩa** | **Viết tắt** |
| Tên nhà cung cấp | Tên nhà cung cấp | TenNCC |
| Địa chỉ | Địa chỉ | DiaChi |
| Số điện thoại | Số điện thoại | SĐT |
| Ngày giao hàng | Ngày giao hàng | NgayGiao |
| Tên hàng | Tên hàng | TenHang |
| Mã hoá đơn bán hàng | Mã hoá đơn nhập hàng | MaNhapHang |
| Số lượng | Số lượng | SoLuong |
| Đơn giá | Đơn giá nhập | DonGiaNhap |
| Đơn vị | Đơn vị | DonVi |
| Khuyến mãi | Khuyến mãi | KhuyenMai |
| Tổng tiền | Tổng tiền | TongTien |
| Thành tiền | Thành tiền | ThanhTien |
| Nhân viên giao hàng | Tên nhân viên giao hàng | TenNV |
| Mã nhân viên giao hàng | Mã nhân viên giao hàng | MaNV |
| Số tiền còn lại | Số tiền còn lại | SoTienCL |
| Số tiền đã thanh toán | Số tiền đã thanh toán | SoTienDaTT |

**Bước 2: Xác định thực thể, thuộc tính**

NHÀ CUNG CẤP (MaNCC, TenNCC, DiaChi, SĐT, MaNV, TenNV)

HÀNG (MaHang, TenHang, DonGiaNhap, DonVi)

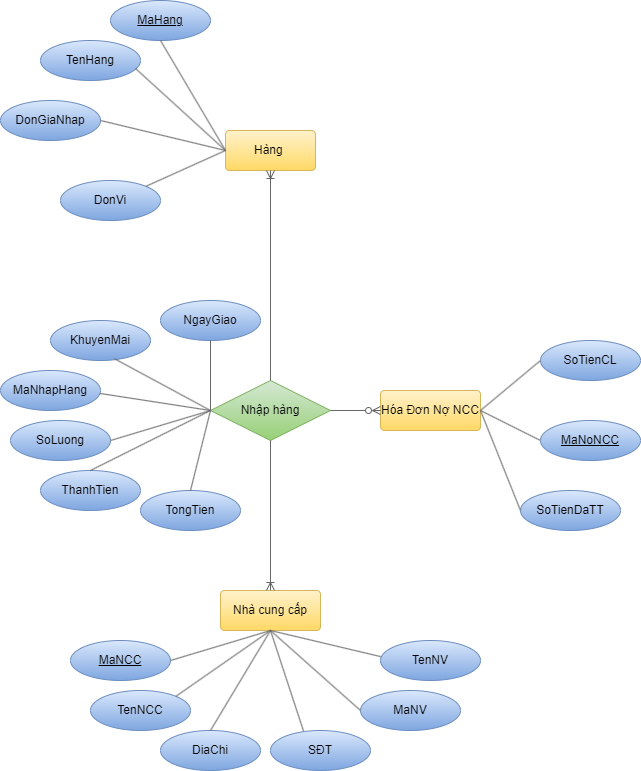
HÓA ĐƠN NỢ NCC (MaNoNCC, SoTienCL, SoTienDaTT)

|  |  |  |
| --- | --- | --- |
| **Từ trong HDSD** | **Từ rõ nghĩa** | **Viết tắt** |
| Tên nhà cung cấp | ~~Tên nhà cung cấp~~ | TenNCC |
| Địa chỉ | ~~Địa chỉ~~ | DiaChi |
| Số điện thoại | ~~Số điện thoại~~ | SĐT |
| Ngày giao hàng | Ngày giao hàng | NgayGiao |
| Tên hàng | ~~Tên hàng~~ | TenHang |
| Mã hoá đơn bán hàng | Mã hoá đơn nhập hàng | MaNhapHang |
| Số lượng | Số lượng | SoLuong |
| Đơn giá | ~~Đơn giá nhập~~ | DonGiaNhap |
| Đơn vị | ~~Đơn vị~~ | DonVi |
| Khuyến mãi | Khuyến mãi | KhuyenMai |
| Tổng tiền | Tổng tiền | TongTien |
| Thành tiền | Thành tiền | ThanhTien |
| Nhân viên giao hàng | ~~Tên nhân viên giao hàng~~ | TenNV |
| Mã nhân viên giao hàng | ~~Mã nhân viên giao hàng~~ | MaNV |
| Số tiền còn lại | ~~Số tiền còn lại~~ | SoTienCL |
| Số tiền đã thanh toán | ~~Số tiền đã thanh toán~~ | SoTienDaTT |

**Bước 3: Xác định quan hệ**

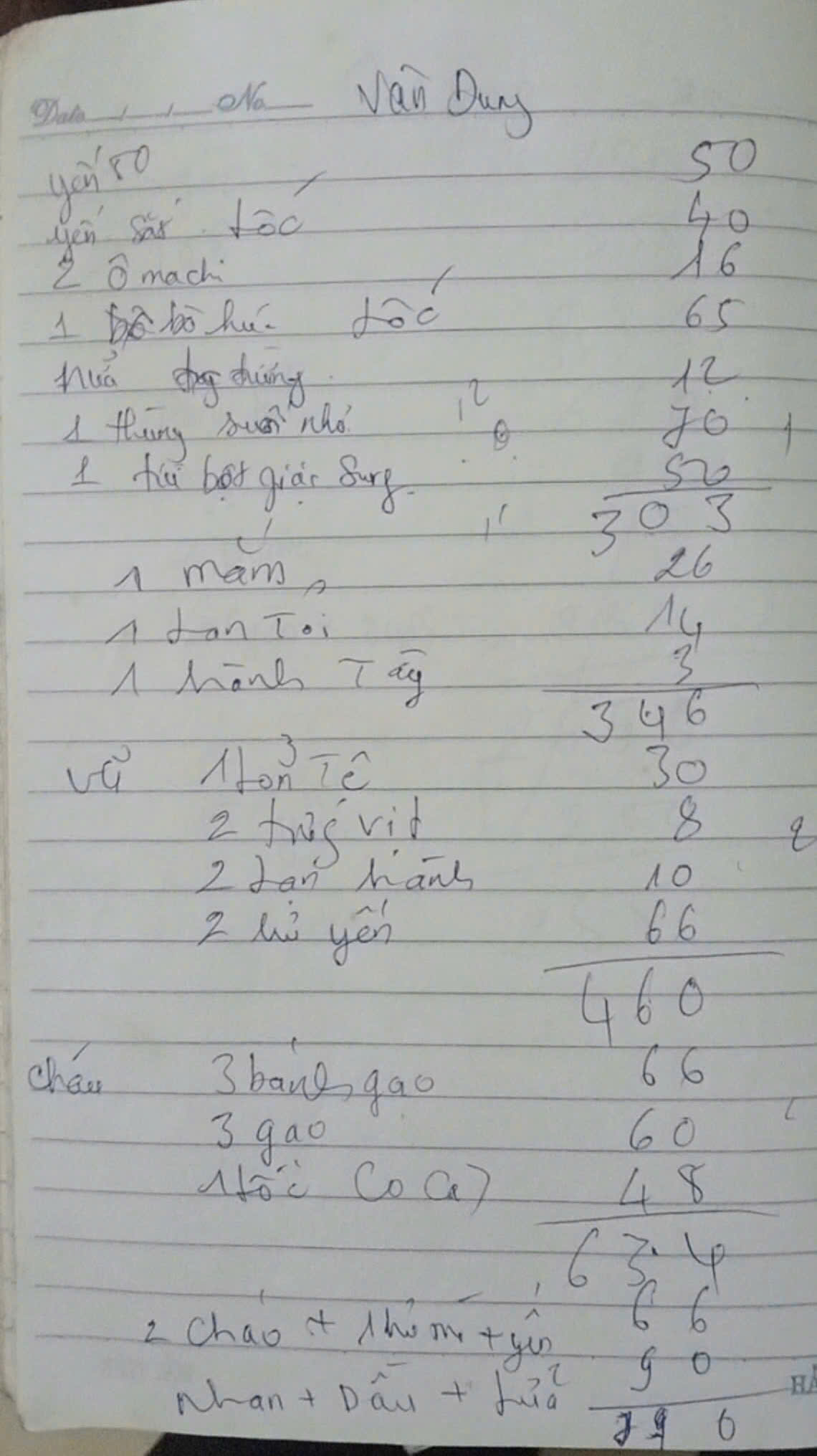
NHẬP HÀNG (MaNhapHang, SoLuong, KhuyenMai, ThanhTien, TongTien, NgayGiao)

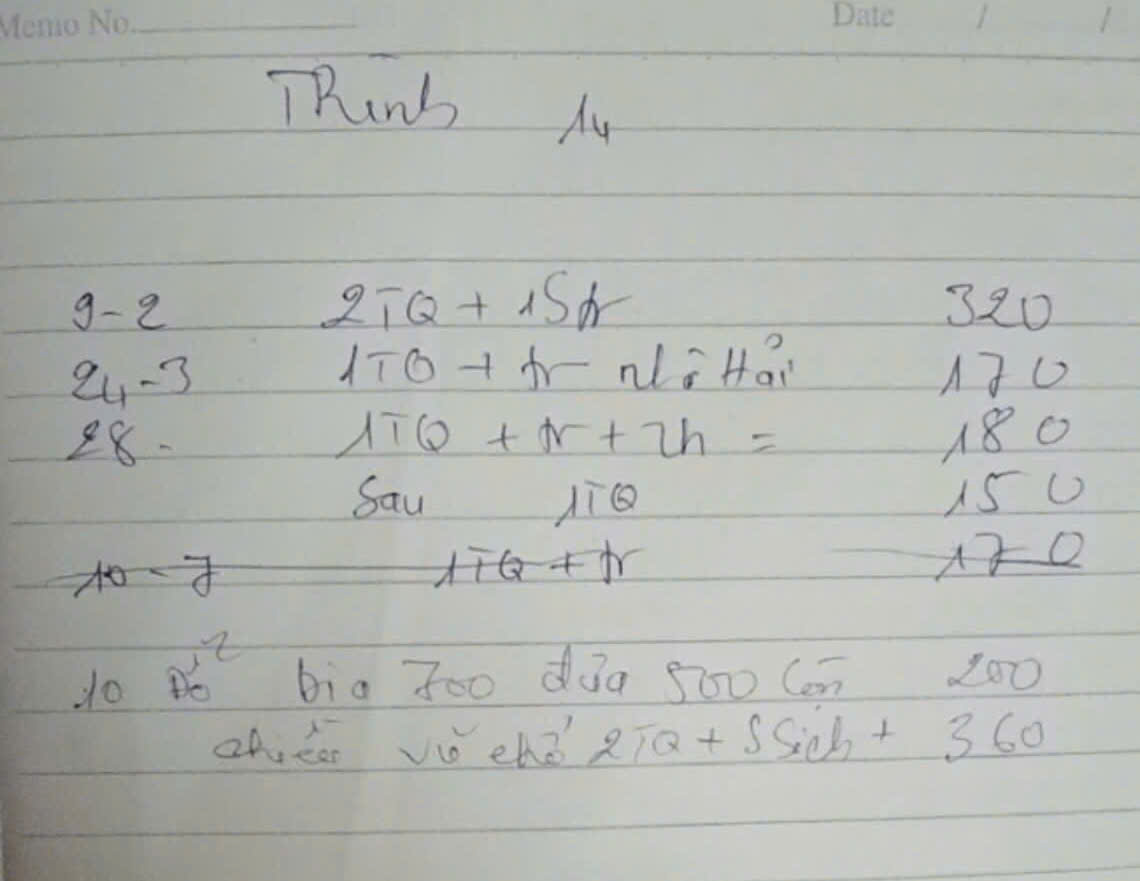
**Bước 4: Vẽ**



Hình 2 Sơ đồ ER-1

### Hoá đơn bán hàng tạp hoá cô Châu

*Link hóa đơn:* <https://drive.google.com/drive/u/1/folders/1q9BTczCw9Vfh2h3_0oUC0yT6rwPXrhYH>



Hình 3 Hoá đơn bán hàng tạp hoá cô Châu

**Bước 1: Chọn lọc thông tin**

|  |  |  |
| --- | --- | --- |
| **Từ trong HDSD** | **Từ rõ nghĩa** | **Viết tắt** |
| Tên khách hàng | Tên khách hàng | TenKH |
| Tên hàng | Tên hàng | TenHang |
| Mã hoá đơn bán hàng | Mã hoá đơn bán hàng | MaBanHang |
| Ngày bán hàng | Ngày bán hàng | NgayBan |
| Số lượng | Số lượng | SoLuong |
| Đơn giá | Đơn giá bán | DonGiaBan |
| Đơn vị | Đơn vị | DonVi |
| Khuyến mãi | Khuyến mãi | KhuyenMai |
| Tổng tiền | Tổng tiền | TongTien |
| Thành tiền | Thành tiền | ThanhTien |
| Số tiền còn lại | Số tiền còn lại | SoTienConLai |
| Số tiền đã thanh toán | Số tiền đã thanh toán | SoTienDaThanhToan |

**Bước 2: Xác định thực thể, thuộc tính**

KHÁCH HÀNG (MaKH, TenKH)

HÓA ĐƠN KH NỢ (MaNoKH, SoTienCL, SoTienDaTT)

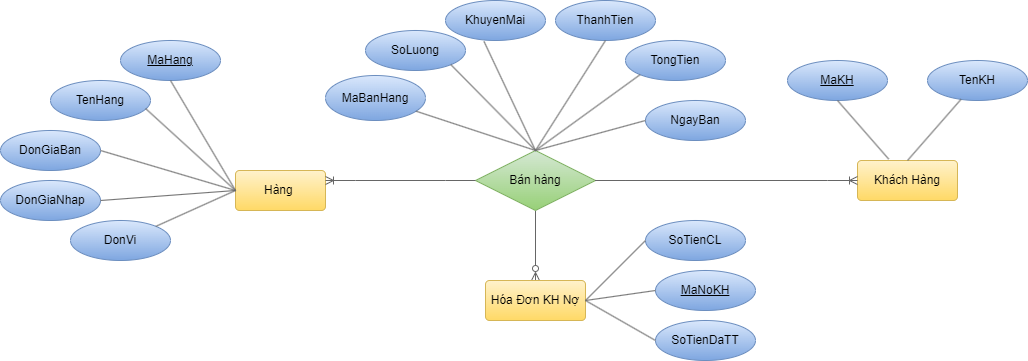
HÀNG (MaHang, TenHang, DonGiaBan, DonVi)

|  |  |  |
| --- | --- | --- |
| **Từ trong HDSD** | **Từ rõ nghĩa** | **Viết tắt** |
| Tên khách hàng | ~~Tên khách hàng~~ | TenKH |
| Tên hàng | ~~Tên hàng~~ | TenHang |
| Mã hoá đơn bán hàng | Mã hoá đơn bán hàng | MaBanHang |
| Ngày bán hàng | Ngày bán hàng | NgayBan |
| Số lượng | Số lượng | SoLuong |
| Đơn giá | ~~Đơn giá bán~~ | DonGiaBan |
| Đơn vị | ~~Đơn vị~~ | DonVi |
| Khuyến mãi | Khuyến mãi | KhuyenMai |
| Tổng tiền | Tổng tiền | TongTien |
| Thành tiền | Thành tiền | ThanhTien |
| Số tiền còn lại | ~~Số tiền còn lại~~ | SoTienConLai |
| Số tiền đã thanh toán | ~~Số tiền đã thanh toán~~ | SoTienDaThanhToan |

**Bước 3: Xác định quan hệ**

BÁN HÀNG (MaBanHang, SoLuong, KhuyenMai, TongTien, ThanhTien, NgayBan)

**Bước 4: Vẽ**



Hình 4 Sơ đồ ER-2

### Tích hợp ER-1 và ER-2

Hình 5 Sơ đồ ER

## Thiết kế cơ sở dữ liệu mức logic

**Bước 1: Chuyển thực thể, thu được**

NHÀ CUNG CẤP (MaNCC, TenNCC, DiaChi, SĐT, MaNV, TenNV)

HÀNG (MaHang, TenHang, DonGiaNhap, DonGiaBan, DonVi)

HÓA ĐƠN NỢ NCC (MaNoNCC, SoTienCL, SoTienDaTT)

KHÁCH HÀNG (MaKH, TenKH)

HÓA ĐƠN KH NỢ (MaNoKH, SoTienCL, SoTienDaTT)

**Bước 2 : Chuyển quan hệ, thu được**

NHẬP HÀNG (MaNhapHang, SoLuong, KhuyenMai, ThanhTien, TongTien, NgayGiao, MaNCC, MaHang, MaNoNCC)

BÁN HÀNG (MaBanHang, SoLuong, KhuyenMai, TongTien, ThanhTien, NgayBan, MaKH, MaHang, MaNoKH)

**Bước 3 : Chuẩn hoá**

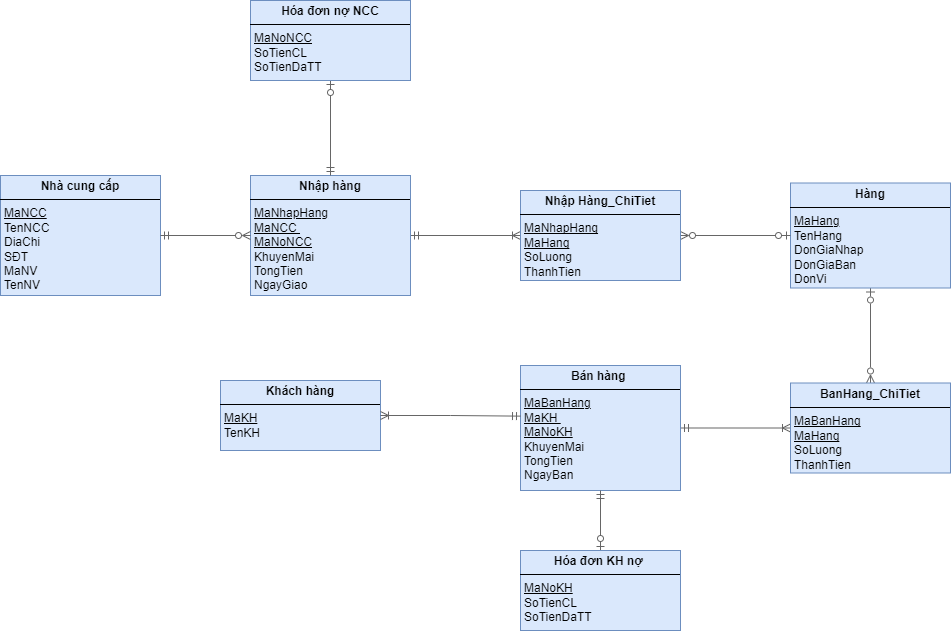
NHẬP HÀNG (MaNhapHang, MaNCC, MaNoNCC, KhuyenMai, TongTien, NgayGiao)

NHẬP HÀNG\_CHITIET (MaNhapHang, MaHang, SoLuong, ThanhTien)

BÁN HÀNG (MaBanHang, MaKH, MaNoKH, KhuyenMai, TongTien, NgayBan)

BÁN HÀNG\_CHITIET (MaBanHang, MaHang, SoLuong, ThanhTien)

**Bước 4: Vẽ**



Hình 6 Sơ đồ quan hệ

## Thiết kế chi tiết các bảng

### Bảng Nhà cung cấp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNCC | VARCHAR(20) | Khóa chính |  |
| 2 | TenNCC | VARCHAR(50) | NOT NULL |  |
| 3 | DiaChi | NVARCHAR(100) | NOT NULL |  |
| 4 | SĐT | CHAR(10) | NOT NULL,  UNIQUE |  |
| 5 | MaNV | VARCHAR (30) | NULL |  |
| 6 | TenNV | VARCHAR (50) | NULL |  |

### Bảng Nhập hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNhapHang | VARCHAR (20) | Khóa chính |  |
| 2 | MaNCC | VARCHAR (30) | Khóa ngoại |  |
| 3 | MaNoNCC | CHAR(10) | Khóa ngoại |  |
| 4 | KhuyenMai | NUMERIC | NULL |  |
| 5 | TongTien | NUMERIC | NOT NULL |  |
| 6 | NgayGiao | DATE | NOT NULL |  |

### Bảng Hóa đơn nợ NCC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNoNCC | CHAR(20) | Khóa chính |  |
| 2 | SoTienCL | NUMERIC | NOT NULL, >= 0 |  |
| 3 | SoTienDaTT | NUMERIC | NULL |  |

### Bảng Nhập Hàng\_ChiTiet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNhapHang | CHAR(20) | Khóa chính, khoá ngoại |  |
| 2 | MaHang | CHAR(20) | Khóa chính, khoá ngoại |  |
| 3 | SoLuong | INT | NOT NULL, >0 |  |
| 4 | ThanhTien | NUMERIC | NOT NULL |  |

### Bảng Hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaHang | CHAR(20) | Khóa chính |  |
| 2 | TenHang | VARCHAR(50) | NOT NULL |  |
| 3 | DonGiaNhap | NUMERIC | NOT NULL, >0 |  |
| 4 | DonGiaBan | NUMERIC | NOT NULL, >0 |  |
| 5 | DonVi | VARCHAR(20) | NULL |  |

### Bảng Bán Hàng\_Chi tiết

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaBanHang | CHAR(20) | Khóa chính, khoá ngoại |  |
| 2 | MaHang | CHAR(20) | Khóa chính, khoá ngoại |  |
| 3 | SoLuong | INT | NOT NULL, >0 |  |
| 4 | ThanhTien | NUMERIC | NOT NULL |  |

### Bảng Bán hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaBanHang | CHAR(20) | Khóa chính |  |
| 2 | MaNoKH | CHAR(20) | Khóa ngoại |  |
| 3 | MaKH | CHAR(20) | Khóa ngoại |  |
| 4 | KhuyenMai | NUMERIC | NULL |  |
| 5 | TongTien | NUMERIC | NOT NULL |  |
| 6 | NgayBan | DATE | NULL |  |

### Bảng Khách hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaKH | CHAR(20) | Khóa chính |  |
| 2 | TenKH | NVARCHAR(50) | NULL |  |
| 3 | SĐTKH | CHAR(10) | NOT NULL,  UNIQUE |  |

### Bảng Hóa đơn KH nợ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNoKH | CHAR(20) | Khóa chính |  |
| 2 | SoTienCL | NUMERIC | NOT NULL, >= 0 |  |
| 3 | SoTienDaTT | NUMERIC | NULL |  |

### Bảng Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | Admin | VARCHAR(50) |  |  |
| 2 | Passwords | VARCHAR(50) |  |  |

# Giai đoạn R3: Xây dựng cơ sở dữ liệu cho bản thiết kế

# Giai đoạn R4: Tạo dữ liệu dump cho các bảng và các module trong cơ sở dữ liệu để phục vụ các thao tác xử lý dữ liệu

## Tạo dữ liệu dump

## 10 module trong cơ sở dữ liệu để phục vụ các thao tác xử lý dữ liệu

### Câu 1 – Module 1:

Khi thêm một mặt hàng mới vào hệ thống, trước tiên cần kiểm tra rằng đơn giá nhập phải nhỏ hơn đơn giá bán và tên mặt hàng chưa tồn tại trong hệ thống. Nếu tên mặt hàng đã tồn tại hoặc điều kiện về đơn giá không thỏa mãn, thì thông báo “Không thỏa mãn” và dừng xử lý. Nếu các điều kiện này đều thỏa mãn, tiến hành thêm mới mặt hàng với mã hàng được tính bằng giá trị lớn nhất hiện có cộng thêm 1. Sau đó, thông báo “Cập nhật thành công”.

|  |  |
| --- | --- |
| Mục đích | Thêm một mặt hàng mới thành công |
| Input | @TenHang,@DonGiaNhap,@DonGiaBan,@DonVi |
| Output | Xác nhận kết quả thực hiện( 1: Cập nhật thành công/0: lỗi, không thêm được) |
| Process | B1. Kiểm tra nếu @DonGiaNhap >= @DonGiaBan hoặc TenHang không tồn tại, print 'Không thỏa mãn' —>kết thúc  B2. Lấy ra @MaHangNew, điều kiện: (max(MaHang) +1)  B3. Thêm một bản ghi mới vào bảng Hàng với các thông tin: Mã hàng, Tên hàng, Đơn giá nhập, Đơn giá bán, Đơn vị và print 'Cập nhật thành công' |

### Câu 2 – Module 2:

Kiểm tra định dạng số điện thoại của khách hàng. Nếu định dạng đúng thì thêm thông tin khách hàng mới, ngược lại thì kết thúc

|  |  |
| --- | --- |
| Mục đích | Kiểm tra định dạng số điện thoại và thêm được thông tin khách hàng mới |
| Input | Số điện thoại của khách hàng, tên khách hàng |
| Output | Xác nhận kết quả thực hiện( 1: thành công/ 0: lỗi, không insert) |
| Process | B1.1 Kiểm tra len(sdt)=10  B1.1.1 Với i<=len(sdt)  B1.1.1 Kiểm tra từng số trong sđt có nằm trong [0-9]   1. nếu có thì i=i+1—>1 2. Ngược lại—>0 và kết thúc   B1.2 Ngược lại—>0  B2. Nếu trả về 1 thì lấy makh=max(makh+1)  B3. Kiểm tra sự tồn tại của makh và sđt  B3.1 Nếu không tồn tại thì thêm mới 1 bản ghi vào bảng KhachHang—>1  B3.2 Ngược lại —>0 |

### Câu 3 – Module 3:

Tạo đơn hàng bán cho khách hàng với điều kiện số tiền nợ lại của khách hàng không bé hơn 2000000 theo số điện thoại. Nếu đủ điều kiện thì in ra tổng tiền đơn hàng và update tổng tiền vào bảng Bán hàng. Nếu vượt quá thì in ra thông báo 'Không đủ điều kiện bán hàng' rồi huỷ đơn hàng đó

|  |  |
| --- | --- |
| Mục đích | Tạo thành công đơn bán hàng cho khách hàng |
| Input | Số điện thoại của khách hàng |
| Output | Xác nhận kết quả thực hiện( 1: thành công/ 0: lỗi, không update) |
| Process | B1: Gọi thủ tục Taohoadon để lấy đầu ra @mahoadon, @ktr →@mahoadon, @ktr  B2a: kiểm tra nếu @ktr=0 →output:0, kết thúc  B2b: ngược lại  B2b.1a: kiểm tra nếu @mahoadon tồn tại thì  B2b.1a.1: Tạo con trỏ để duyệt qua các dòng trong bảng tạm (@ds)  B2b.1a.2: lấy ra các cột mahang và soluong từ biến bảng @ds, và con trỏ sẽ duyệt qua từng dòng dữ liệu này.  B2b.1a.3: Mở con trỏ để chuẩn bị duyệt qua các dòng dữ liệu trong @ds  B2b.1a.4: Lấy dòng dữ liệu tiếp theo từ con trỏ và gán giá trị của các cột mahang và soluong vào các biến @mahang và @soluong.  B2b.1a.5a: Nếu @@FETCH\_STATUS = 0, đã thành công và con trỏ vẫn còn dữ liệu để đọc.  B2b.1a.5a.1a: kiểm tra nếu @mahang không tồn tại trong @mahoadon của bảng BanHang\_chitiet thì:  B2b.1a.5a.1a.1: set @thanhtien=@soluong\* đơn giá bán của bảng Hàng với điều kiện Mahang=@mahang  B2b.1a.5a.1a.2: Thêm 1 bản ghi mới và bảng BanHang\_chitiet  B2b.1a.5a.1a.3: cập nhật tổng tiền của bảng BanHang TongTien=TongTien+@thanhtien với điều kiện Mabanhang=@mahoadon và Mahang=@mahang  B2b.1a.5a.1a.4: set @dem=@dem+1  B2b.1a.5a.1a.5: lấy bản ghi tiếp theo từ con trỏ để gán @mahang=mahang, @soluong=soluong  B2b.1a.5b: ngược lại dừng vòng lặp  B2b.1a.6: đóng con trỏ và giải phóng tài nguyên cấp phát cho con trỏ→output=1  B2.1b: ngược lại →output:0 |

### Câu 4 – Module 4:

Tính doanh thu/chi phí/ lợi nhuận của quán trong tháng 9 năm 2024

|  |  |
| --- | --- |
| Mục đích | Tính doanh thu/chi phí/ lợi nhuận của quán trong tháng 9 năm 2024 |
| Input | Thời gian: 9/2024 |
| Output | Doanh thu, chi phí, lợi nhuận |
| Process | B1.1. Nếu @thoigian có kí tự ‘/’ thì tiến hành lấy các ký tự trước ‘/’ làm giá trị tháng và 4 ký tự từ bên phải làm giá trị năm—>bước 2  B1.2. Nếu @thoigian có kí tự ‘-’ thì tiến hành lấy các ký tự trước ‘-’ làm giá trị tháng và 4 ký tự từ bên phải làm giá trị năm —>bước 2  B1.3. Nếu LEN(@thoigian) = 4 và @thoigian chỉ bao gồm các kí tự số thì lấy @thoigian=năm và tháng=null—>bước 2  B1.4 Còn lại, print ‘Định dạng không hợp lệ’---> kết thúc  B2. Lấy ra @doanhthu=sum(TongTien) trong bảng BanHang, đk: year(NgayBan) = @nam and (month(NgayBan) = @thang or @thang is null)  B3.1. Nếu @doanhthu=null —>@doanhthu=0  B3.2 Ngược lại @doanhthu=@doanhthu  B4. Lấy ra @chiphi=sum(TongTien) trong bảng NhapHang, đk: year(NgayBan) = @nam and (month(NgayBan) = @thang or @thang is null)  B4.1. Nếu @chiphi=null —>@chiphi=0  B4.2 Ngược lại @chiphi=@chiphi  B5. Tính @loinhuan = @doanhthu - @chiphi |

### Câu 5 – Module 5:

Xóa thông tin khách hàng nếu khách hàng không mua hàng   trong vòng 3 năm trở lại đây và không còn nợ thì tiến hành xóa thông tin khách hàng và các thông tin liên quan đến khách hàng đó

|  |  |
| --- | --- |
| Mục đích | Xóa tất cả thông tin liên quan đến  khách hàng không còn mua hàng ở cửa hàng trong vòng 3 năm trở lại đây |
| Input | Mã khách hàng |
| Output | Xác nhận kết quả thực hiện( 1: thành công/ 0: lỗi, không thực hiện xóa) |
| Process | 1. tìm thời gian mua hàng gần nhất của khách hàng, đk mã khách hàng = @MaKH  2. nếu thời gian mua hàng dưới 3 năm thì kết thúc --> output parameter = 0  3. ngược lại:  3.1 tính lại tổng tiền còn lại của khách hàng, đk: mã khách hàng = @MaKH  3.2 nếu tổng tiền còn lại > 0 thì kết thúc --> output parameter = 0, print ngayban  3.3 ngược lại:  3.3.1 xóa dữ liệu ở bảng Hoadon\_KHNo, đk: mã nợ khách hàng nằm trong bảng BanHang với mã khách hàng = @MaKH  3.3.1.1 Nếu không tồn tại thì kết thúc--> output parameter = 0, print 'Lỗi khi xóa Hóa đơn nợ'  3.3.1.2 xóa dữ liệu ở bảng KhachHang, đk: mã khách hàng = @MaKH-> output parameter = 0, print 'Lỗi khi xóa Khách hàng'  3.3.1.3 nếu xóa thành công ---> output parameter = 1 |

### Câu 6 – Module 6:

Cập nhật số tiền đã thanh toán và số tiền còn lại nếu khách hàng thanh toán hóa đơn nợ

|  |  |
| --- | --- |
| Mục đích | Cập nhật số tiền khách hàng thanh toán và số tiền nợ còn lại khi khách hàng thanh toán hoá đơn nợ |
| Input | Mã khách hàng, số tiền thanh toán |
| Output | Xác nhận kết quả thực hiện( 1: thành công/ 0: lỗi, không update) |
| Process | 1. Lấy ra SoTienCL của khách hàng trong bảng Hoadon\_KHNo, đk: MaNoKH=@ManoKH  2. nếu số tiền thanh toán > tổng tiền còn lại --> hiển thị thông báo "Số tiền thanh toán vượt quá số tiền còn lại"--> output parameter = 0 —>kết thúc  3. ngược lại:  3.1 update Hoadon\_KHNo, SoTienDaTT =SoTienDaTT + @SoTien,  SoTienCL = SoTienCL - @SoTien  Điều kiện: MaNoKH = @ManoKH  3.1.1 Kiểm tra trạng thái cập nhật  3.1.1.1 Nếu không có —> output parameter = 0  3.1.1.2 ngược lại output parameter = 1 |

### Câu 7 – Module 7:

Trước khi quyết định đặt hàng cần kiểm tra số lượng bán của từng sản phẩm muốn đặt theo MaHang trong 3 tháng vừa qua. Nếu số lượng <20 thì hiển thị thông báo “số lượng bán của sản phẩm này thấp hơn chỉ tiêu” và huỷ đặt mặt hàng đó. Ngược lại, tiến hành đưa ra thông tin những nhà cung cấp có mặt hàng đó

|  |  |
| --- | --- |
| Mục đích | Kiểm tra số lượng các mặt hàng đã bán trong 3 tháng vừa qua để quyết định có đặt hàng không |
| Input | MaHang |
| Output | Xác nhận kết quả thực hiện( 1: thành công/ 0: lỗi, không insert) |
| Process | B1. Kiểm tra sự tồn tại của mã hàng, đk: MaHang=@mahang  B1.1. Nếu tồn tại, tính tổng số lượng bán hàng, đk: MaHang=@mahang, DATEDIFF (month, NgayBan, GETDATE ()) <=3  B1.1.1. Nếu @TongSL<20—> in ra thông báo 'Số lượng bán của sản phẩm này thấp hơn chỉ tiêu'----->0  B1.1.2. Ngược lại tạo một biến bảng với các thông tin: MaNCC, TenNCC, NhaCungCap, DiaChi, SĐT trong bảng NhaCungCap, đk: MaHang=@mahang—>1  B1.2 Ngược lại thì kết thúc —>0 |

### Câu 8 – Module 8:

Kiểm tra số tiền thanh toán cho nhà cung cấp có vượt quá số tiền còn lại hay không. Nếu có thì hiển thị thông báo 'Số tiền vượt quá' và ngừng xử lý, ngược lại thì update số tiền đã thanh toán= số tiền đã thanh toán + số tiền vừa thanh toán và số tiền còn lại= Tổng tiền-số tiền đã thanh toán

|  |  |
| --- | --- |
| Mục đích | Thanh toán số tiền còn nợ cho nhà cung cấp |
| Input | MaNCC, số tiền thanh toán |
| Output | Update số tiền đã thanh toán, số tiền còn lại   * Thất bại: in ra 'Số tiền vượt quá số tiền còn lại' và số tiền còn lại * Thành công: in ra 'cập nhật thành công' |
| Process | B1. Tìm SoTienConLai, TongTien, đk : HoaDonNo\_NCC.MaNoNCC=@manoncc  B2a. số tiền thanh toán> SoTienConLai  —> print 'Số tiền vượt quá số tiền còn lại' và SoTienConLai  B2b Ngược lại, update HoaDonNo\_NCC , SoTienDaTT=SoTienDaTT+@sotien, SoTienCL=@tongtien-SoTienDaTT, đk: MaNoNCC=@manoncc và print 'cập nhật thành công' |

### Câu 9 – Module 9:

Cập nhật lại thông tin nhà cung cấp khi có sự thay đổi về số điện thoại hoặc nhân viên giao hàng.

|  |  |
| --- | --- |
| Mục đích | Cập nhật lại thông tin nhà cung cấp khi có sự thay đổi |
| Bảng | NhaCungCap |
| Loại | for |
| Sự kiện | update |
| Process | B1. Lấy ra MaNCC, SĐT, TenNV của bảng inserted—> @MaNCC, @sdt, @tennv  B2.  Kiểm tra sự tồn tại MaNCC trong bảng NhaCungCap, đk: MaNCC=@MaNCC  b2.1 Nếu ko tồn tại  thì in ra 'Nhà cung cấp không tồn tại' + rollback  b2.2 Ngược lại tiến hành kiểm tra sự tồn tại sđt và tennv trong bảng deleted với điều kiện: SĐT != @sdt hoặc TenNV!=@tennv   1. Nếu tồn tại thì in ra thông báo ‘Cập nhật thành công’ và update NhaCungCap, SĐT=@sdt, TenNV=@tennv, đk: MaNCC=@MaNCC 2. Ngược lại, print 'Không có sự thay đổi'+rollback |

### Câu 10 – Module 10:

Khi áp dụng khuyến mãi cho một đơn hàng, kiểm tra các điều kiện sau: Nếu tổng giá trị của đơn hàng dưới 500000 thì không áp dụng khuyến mãi và đưa ra thông báo 'Không đủ điều kiện áp dụng khuyến mãi', từ 500000-1000000 thì khuyến mãi 2%, còn trên 1000000 thì khuyến mãi 5% và tiến hành cập nhật đơn hàng và đưa ra thông báo 'Cập nhật thành công đơn hàng'

|  |  |
| --- | --- |
| Mục đích | Kiểm tra xem những đơn hàng nào áp dụng được khuyến mãi để tiến hành  áp dụng khuyến mãi cho hoá đơn bán hàng |
| Bảng | BanHang |
| Loại | for |
| Sự kiện | insert |
| Process | B1. Lấy ra MaBanHang, MaKH, TongTien của bảng inserted —> @MaBanHang, @MaKH, @TongTien  B2. Kiểm tra tổng tiền của đơn hàng  2.1 Nếu @TongTien<500000  print 'Không đủ điều kiện áp dụng khuyến mãi' và @KhuyenMai=0  2.2 Nếu @TongTien >= 500000 và @TongTien <= 1000000 thì @KhuyenMai=0.02  2.3 Nếu @TongTien >1000000 thì @KhuyenMai=0.05  B3. Tính @TongTien=@TongTien - (@TongTien \* @KhuyenMai)  B4. Update BanHang, KhuyenMai = @KhuyenMai\*100, TongTien = @TongTien, đk: MaBanHang = @MaBanHang và print  'Cập nhật thành công đơn hàng' |

# Giai đoạn R5: Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu

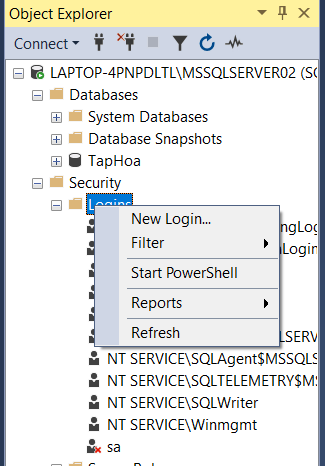
## Thiết lập cơ chế authentication phù hợp

### Lý do lựa chọn phân quyền

Nhóm chỉ phân quyền toàn quyền truy cập đối với chủ cửa hàng với vai trò là Admin. Vì đây là một cơ sở kinh doanh nhỏ lẽ không có thuê nhân viên. Và để khi nếu xảy ra lỗi hoặc sai sót, người duy nhất chịu trách nhiệm sẽ là chủ cửa hàng. Điều này giúp việc quản lý hệ thống trở nên minh bạch hơn.

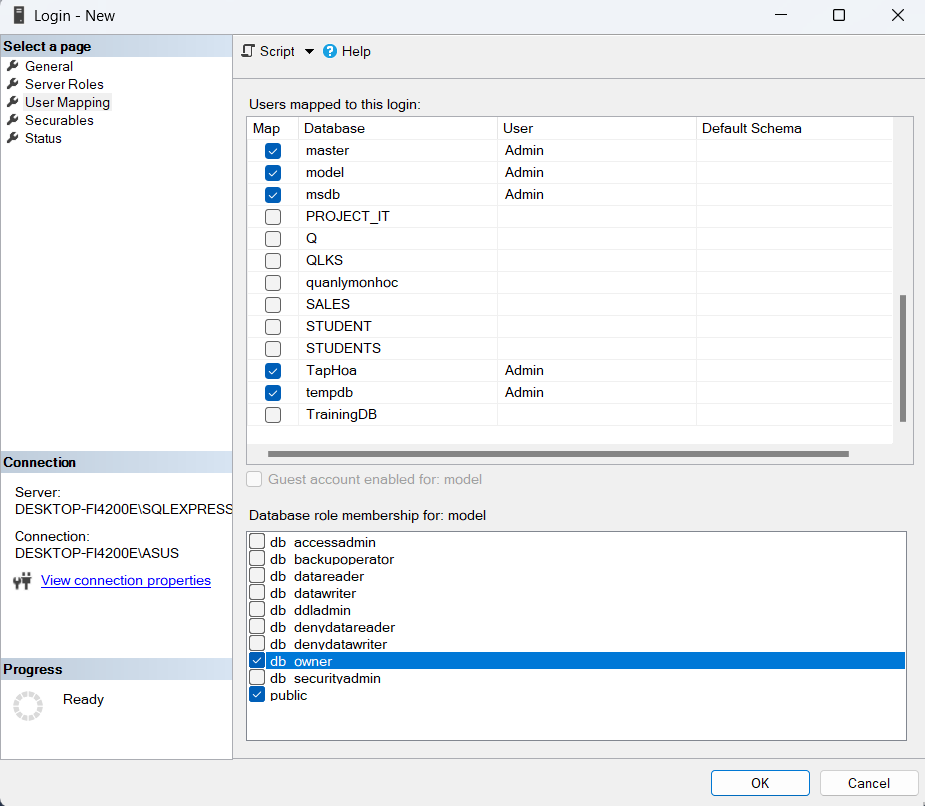
### Các bước triển khai

B1. Ở Security 🡪Login🡪New Login…

B2. Ở General 🡪 Name: Admin🡪Chọn SQL Server authentication🡪Nhập password và Confirm password: Admin123

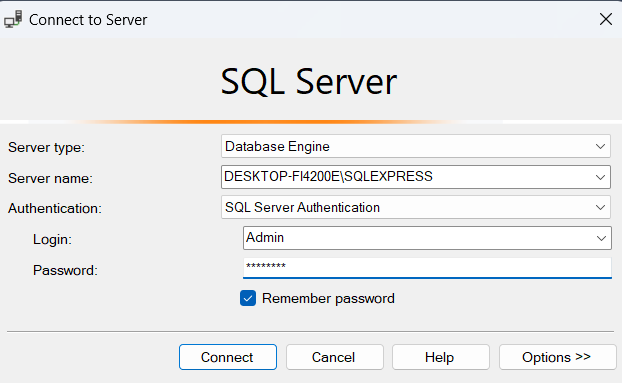


B3. Vào User Mapping🡪 Chọn như hình sau:

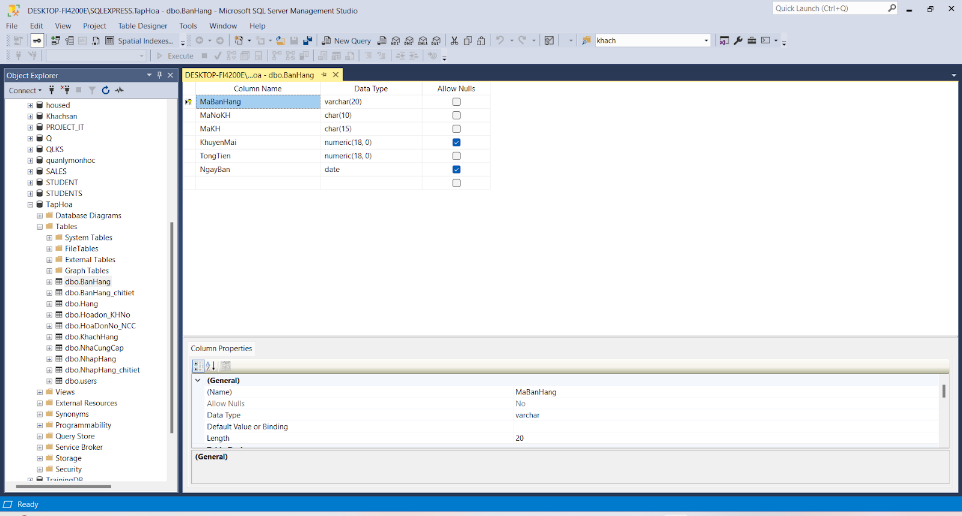


🡪Chọn Ok

B4. Vào lại SQL Server🡪Chọn Authentication: SQL Sever Authentication🡪Nhập login: Admin🡪 Password: Admin123🡪Chọn Connect



B5 Xác nhận lại đã phân quyền thành công



## Cơ chế bảo mật các dữ liệu nhạy cảm

### Mã hoá dữ liệu

##### Tạo các cột mã hoá

---Tạo cột SĐT\_encrypt varbinary(MAX) của bảng NhaCungCap

ALTER TABLE [dbo].[NhaCungCap]

ADD SĐT\_encrypt varbinary(MAX)

---Tạo cột SĐT\_encrypt varbinary(MAX) của bảng KhachHang

ALTER TABLE [dbo].[KhachHang]

ADD SĐT\_encrypt varbinary(MAX)

---Tạo cột Sotienconlai varbinary(MAX) của bảng Hoadon\_KHNo

ALTER TABLE [dbo].[Hoadon\_KHNo]

ADD Sotienconlai varbinary(MAX)

---Tạo cột Sotiendathanhtoan varbinary(MAX) của bảng Hoadon\_KHNo

ALTER TABLE [dbo].[Hoadon\_KHNo]

ADD Sotiendathanhtoan varbinary(MAX)

---Tạo cột Sotienconlai varbinary(MAX) của bảng HoaDonNo\_NCC

ALTER TABLE [dbo].[HoaDonNo\_NCC]

ADD Sotienconlai varbinary(MAX)

---Tạo cột Sotiendathanhtoan varbinary(MAX) của bảng HoaDonNo\_NCC

ALTER TABLE [dbo].[HoaDonNo\_NCC]

ADD Sotiendathanhtoan varbinary(MAX)

alter table users

add password varbinary(MAX)

##### Tạo khoá mã hoá

create or alter proc Taokhoa

as

begin

--Tạo khóa chính cơ sở dữ liệu

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'TapHoa48k21.2';

--Tạo chứng chỉ tự ký cho SQL Server

CREATE CERTIFICATE TapHoa\_test

WITH SUBJECT = 'Protect my data';

--Tạo Cấu hình khóa đối xứng để mã hóa

CREATE SYMMETRIC KEY TapHoaKey\_test

WITH ALGORITHM = AES\_256

ENCRYPTION BY CERTIFICATE TapHoa\_test

end

exec Taokhoa

##### Mã hoá dữ liệu trong bảng

create or alter proc MahoaBang

as

begin

-- mở khóa đối xứng và mã hóa bằng chứng chỉ.

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

UPDATE [dbo].[NhaCungCap]

SET SĐT\_encrypt = EncryptByKey (Key\_GUID('TapHoaKey\_test'),SĐT)

FROM NhaCungCap

--

UPDATE [dbo].[KhachHang]

SET SĐT\_encrypt = EncryptByKey (Key\_GUID('TapHoaKey\_test'), SĐTKH)

FROM [dbo].[KhachHang]

---

UPDATE [dbo].[Hoadon\_KHNo]

SET Sotienconlai = EncryptByKey (Key\_GUID('TapHoaKey\_test'),cast( SoTienCL as varchar))

FROM [dbo].[Hoadon\_KHNo]

--

UPDATE [dbo].[Hoadon\_KHNo]

SET Sotiendathanhtoan=EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast(SoTienDaTT as varchar))

FROM [dbo].[Hoadon\_KHNo]

--

UPDATE [dbo].[HoaDonNo\_NCC]

SET Sotienconlai = EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast( SoTienCL as varchar))

FROM [dbo].[HoaDonNo\_NCC]

--

UPDATE [dbo].[HoaDonNo\_NCC]

SET Sotiendathanhtoan=EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast(SoTienDaTT as varchar))

FROM [dbo].[HoaDonNo\_NCC]

--

update users

SET [password] = EncryptByKey (Key\_GUID('TapHoaKey\_test'),Passwords)

from users

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

declare @UQ\_SĐT varchar(50)

set @UQ\_SĐT=(SELECT CONSTRAINT\_NAME

FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

WHERE TABLE\_NAME = 'Nhacungcap' AND CONSTRAINT\_TYPE = 'UNIQUE')

declare @UQ\_SĐTKH varchar(50)

set @UQ\_SĐTKH=(SELECT CONSTRAINT\_NAME

FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

WHERE TABLE\_NAME = 'KhachHang' AND CONSTRAINT\_TYPE = 'UNIQUE')

-- Drop the [SĐT] column

EXEC('ALTER TABLE dbo.Nhacungcap DROP CONSTRAINT ' +@UQ\_SĐT)

EXEC('ALTER TABLE dbo.Nhacungcap DROP COLUMN SĐT')

set @UQ\_SĐTKH= (SELECT CONSTRAINT\_NAME

FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

WHERE TABLE\_NAME = 'KhachHang' AND CONSTRAINT\_TYPE = 'UNIQUE')

EXEC ('ALTER TABLE dbo.KhachHang DROP CONSTRAINT ' +@UQ\_SĐTKH)

EXEC ('ALTER TABLE dbo.KhachHang DROP COLUMN SĐTKH')

alter table [dbo].[Hoadon\_KHNo]

drop column [SoTienCL]

alter table [dbo].[Hoadon\_KHNo]

drop column [SoTienDaTT]

alter table [dbo].[HoaDonNo\_NCC]

drop column [SoTienCL]

alter table [dbo].[HoaDonNo\_NCC]

drop column [SoTienDaTT]

alter table users

drop column [Passwords]

end

exec MahoaBang

### Giải mã dữ liệu

##### Giải mã mật khẩu bảng Users

create or alter proc GiaimaBangUser

as

begin

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

SELECT \*,

CONVERT (varchar, DecryptByKey([password])) AS 'Decrypted\_password'

FROM users

--đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

end

exec GiaimaBangUser

##### Giải mã số điện thoại bảng NhaCungCap

create or alter proc GiaimaBangNCC @SĐT NVARCHAR (50) = NULL

as

begin

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

if @SĐT is null

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(SĐT\_encrypt)) AS 'Decrypted\_SĐT'

FROM NhaCungCap

end

else

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(SĐT\_encrypt)) AS 'Decrypted\_SĐT'

FROM NhaCungCap

where CONVERT (varchar, DecryptByKey(SĐT\_encrypt)) =@SĐT

end

--đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

end

exec GiaimaBangNCC

##### Giải mã số điện thoại bảng KhachHang

create or alter proc GiaimaBangKH @SĐTKH NVARCHAR (50) = NULL

as

begin

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

if @SĐTKH is null

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(SĐT\_encrypt)) AS 'Decrypted\_SĐTKH'

FROM KhachHang

end

else

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(SĐT\_encrypt)) AS 'Decrypted\_SĐTKH'

FROM KhachHang

where CONVERT (varchar, DecryptByKey(SĐT\_encrypt))=@SĐTKH

end

--đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

end

exec GiaimaBangKH

##### Giải mã số tiền còn lại và số tiền đã thanh toán bảng HoaDonNo\_NCC

create or alter proc GiaimaHoadonnoNCC @Mahoadon varchar (50) = null

as

begin

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

if @Mahoadon is null

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(Sotienconlai)) AS 'Sotiencldamahoa',

CONVERT (varchar, DecryptByKey(Sotiendathanhtoan)) AS 'Sotiendathanhtoandamahoa'

FROM HoaDonNo\_NCC

end

else

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(Sotienconlai)) AS 'Sotiencldamahoa',

CONVERT (varchar, DecryptByKey(Sotiendathanhtoan)) AS 'Sotiendathanhtoandamahoa'

FROM HoaDonNo\_NCC

where MaNoNCC=@Mahoadon

end

--đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

end

exec GiaimaHoadonnoNCC

##### Giải mã số tiền còn lại và số tiền đã thanh toán bảng HoaDon\_KHNo

create or alter proc GiaimaHoadonKHno @Mahoadon varchar(50)=null

as

begin

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

if @Mahoadon is null

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(Sotienconlai)) AS 'Sotiencldamahoa',

CONVERT (varchar, DecryptByKey(Sotiendathanhtoan)) AS 'Sotiendathanhtoandamahoa'

FROM Hoadon\_KHNo

end

else

begin

SELECT \*,

CONVERT (varchar, DecryptByKey(Sotienconlai)) AS 'Sotiencldamahoa',

CONVERT (varchar, DecryptByKey(Sotiendathanhtoan)) AS 'Sotiendathanhtoandamahoa'

FROM Hoadon\_KHNo

where MaNoKH=@Mahoadon

end

--đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

end

exec GiaimaHoadonKHno

### Thêm dữ liệu

##### Mã hoá số điện thoại khi thêm dữ liệu bảng NhaCungCap

create or alter proc Mahoathemdulieuncc (@tenncc nvarchar (50), @Diachi nvarchar (50), @SĐT varchar (11), @Tennv nvarchar (50), @ref bit output)

as

begin

declare @maNCC varchar (50), @manv varchar (50), @SĐT\_encrypt varbinary(max)

select @maNCC=ISNULL(MAX(MaNCC),0) + 1 FROM NhaCungCap;

select @manv=ISNULL(MAX(MaNV), 0) + 1 FROM NhaCungCap;

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

SET @SĐT\_encrypt = EncryptByKey (Key\_GUID('TapHoaKey\_test'), @SĐT)

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

insert into NhaCungCap (MaNCC, TenNCC, DiaChi, MaNV, TenNV, SĐT\_encrypt)

values (@maNCC, @tenncc, @Diachi, @manv, @Tennv, @SĐT\_encrypt)

if @@ROWCOUNT<=0

set @ref=0

else set @ref=1

end

declare @ret bit

exec Mahoathemdulieuncc 'Ncc1001','dc01','0344642480','Nguyen Van A', @ret output

##### Mã hoá số điện thoại khi thêm dữ liệu KhachHang

create or alter proc MahoathemdulieuKh (@tenkh nvarchar (50), @SĐT varchar (11), @ref bit output)

as

begin

declare @maKH varchar (50), @SĐT\_encrypt varbinary(max)

SELECT @maKH = cast(cast(max(MaKH) as numeric (18)) +1 as varchar) from KhachHang

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

SET @SĐT\_encrypt = EncryptByKey (Key\_GUID('TapHoaKey\_test'), @SĐT)

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

insert into KhachHang (MaKH, TenKH, SĐT\_encrypt)

values (@maKH, @tenkh, @SĐT\_encrypt)

if @@ROWCOUNT<=0

set @ref=0

else set @ref=1

end

select\*from KhachHang

declare @ret bit

exec MahoathemdulieuKh 'KH1001','0344642480', @ret output

##### Mã hoá số tiền đã thanh toán và số tiền còn lại khi thêm dữ liệu bảng HoaDonNo\_NCC

select\*from HoaDonNo\_NCC

create or alter proc MahoathemdulieuHDNoNCC (@sotiendathanhtoan numeric (10,2), @sotienconlai numeric (10,2), @ref bit output)

as

begin

declare @maNoNCC varchar (50), @Sotiencl varbinary(max), @Sotiendatt varbinary(max)

SELECT @maNoNCC = cast(cast(max(MaNoNCC) as numeric (18)) +1 as varchar)

FROM HoaDonNo\_NCC;

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

SET @Sotiencl = EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast (@sotienconlai as varchar))

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

--

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

SET @Sotiendatt = EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast (@sotiendathanhtoan as varchar))

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

insert into HoaDonNo\_NCC (MaNoNCC, Sotienconlai, Sotiendathanhtoan)

values (@maNoNCC, @Sotiencl, @Sotiendatt)

if @@ROWCOUNT<=0

set @ref=0

else set @ref=1

end

declare @ret bit

exec MahoathemdulieuHDNoNCC 10000000, 20000000, @ret out

##### Mã hoá số tiền đã thanh toán và số tiền còn lại khi thêm dữ liệu bảng HoaDon\_KHNo

create or alter proc MahoathemdulieuHDKHNo (@sotiendathanhtoan numeric (10,2), @sotienconlai numeric (10,2), @ref bit output)

as

begin

declare @maKHNo varchar (50), @Sotiencl varbinary(max), @Sotiendatt varbinary(max)

SELECT @maKHNo = cast(cast(max(MaNoKH) as numeric (18)) +1 as varchar)

FROM Hoadon\_KHNo;

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

SET @Sotiencl = EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast (@sotienconlai as varchar))

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

--

OPEN SYMMETRIC KEY TapHoaKey\_test

DECRYPTION BY CERTIFICATE TapHoa\_test;

--cập nhật dữ liệu đã mã hóa

SET @Sotiendatt = EncryptByKey (Key\_GUID('TapHoaKey\_test'), cast (@sotiendathanhtoan as varchar))

--Đóng khóa đối xứng

CLOSE SYMMETRIC KEY TapHoaKey\_test;

insert into Hoadon\_KHNo (MaNoKH, Sotienconlai, Sotiendathanhtoan)

values (@maKHNo, @Sotiencl, @Sotiendatt)

if @@ROWCOUNT<=0

set @ref=0

else set @ref=1

end

declare @ret bit

exec MahoathemdulieuHDNoNCC 10000000, 20000000, @ret out

# Giai đoạn R6: Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu

## Lý do chọn loại sao lưu: Full backup

Phù hợp với nhu cầu của cửa hàng tạp hoá.

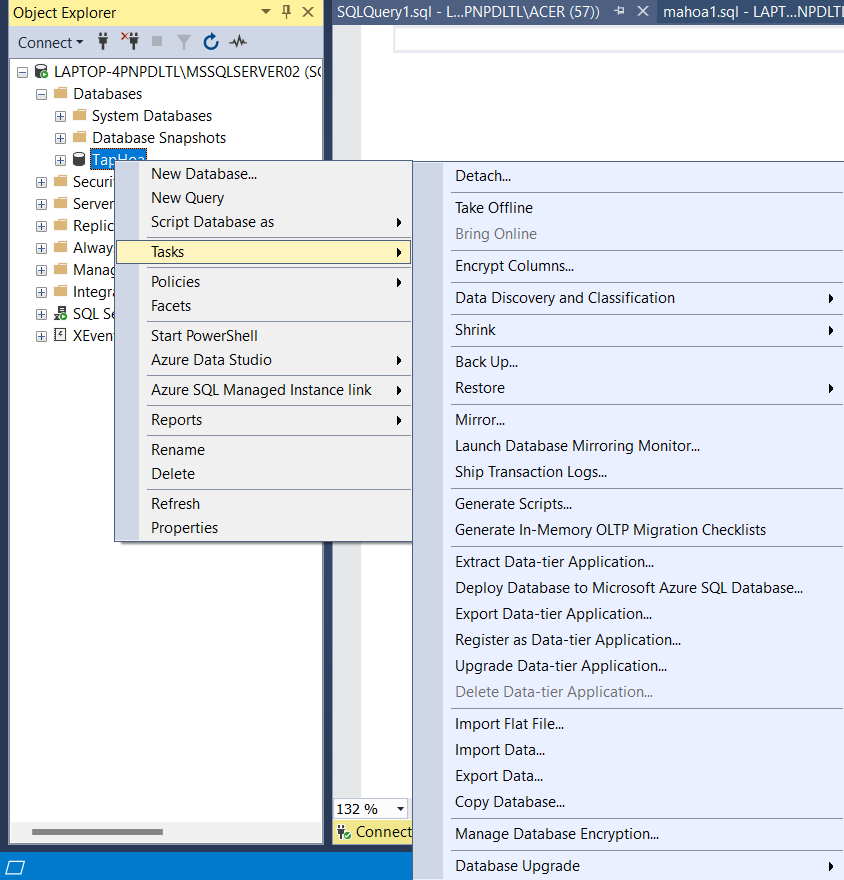
Dữ liệu của một cửa hàng tạp hóa thường không quá lớn so với các doanh nghiệp lớn, nên việc sao lưu toàn bộ không đòi hỏi nhiều tài nguyên.

Cửa hàng tạp hóa thường không có đội ngũ kỹ thuật chuyên nghiệp, nên cần một phương pháp dễ áp dụng. Full backup không đòi hỏi thiết lập phức tạp hoặc giám sát liên tục như các loại sao lưu khác.

## Các bước sao lưu dữ liệu

B1. Lựa chọn thao tác backup

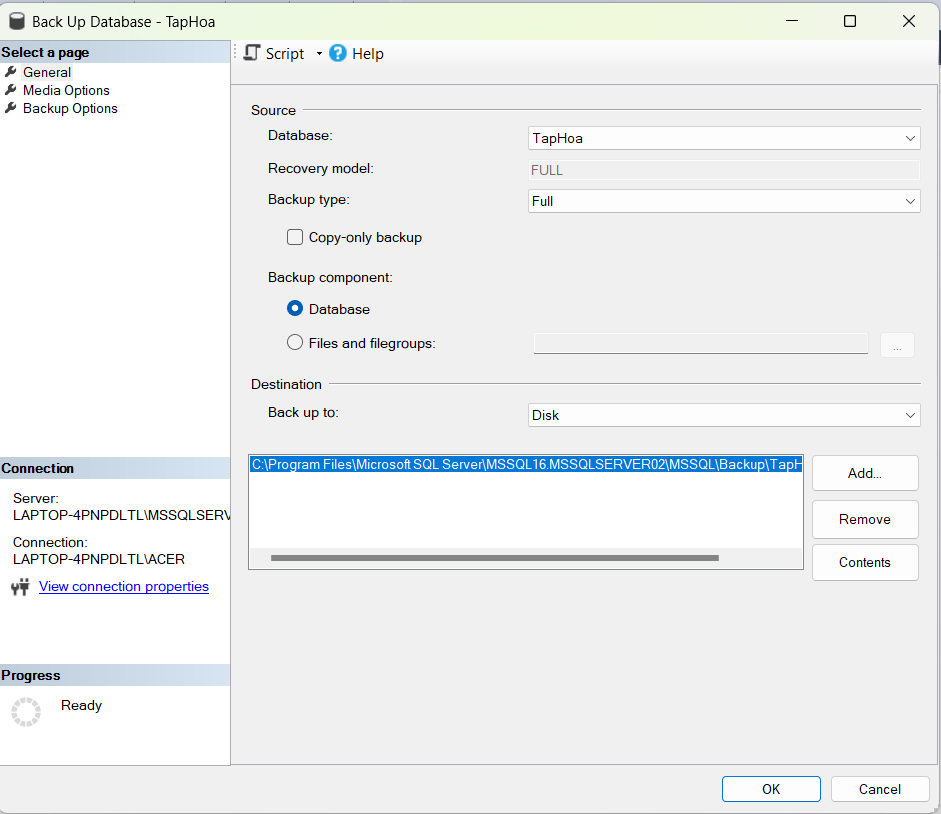
B1.1 Click chuột phải vào database cần backup (TapHoa)

B1.2 Chọn Task 🡪 Back Up

B2. Thiết lập các tùy chọn backup

Database: TapHoa

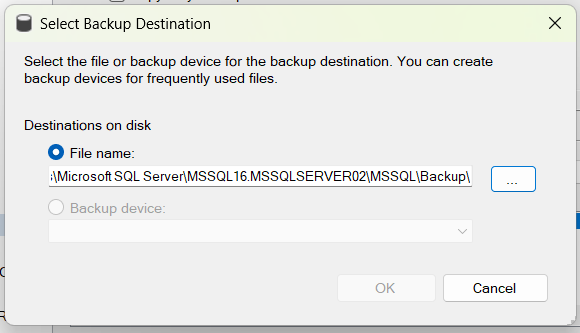
Backup type: Full

Backup component: Database

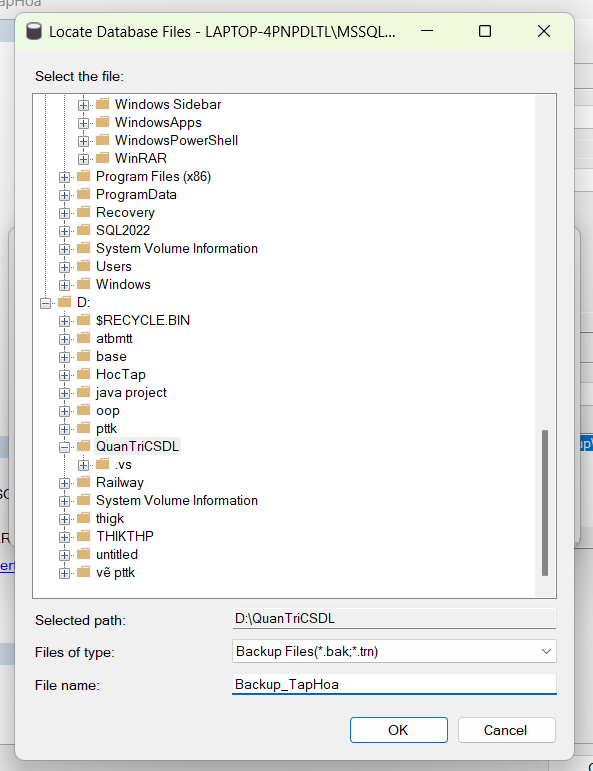
B3. Lựa chọn đường dẫn sao lưu

3.1 Click Remove 🡪 Click Add…

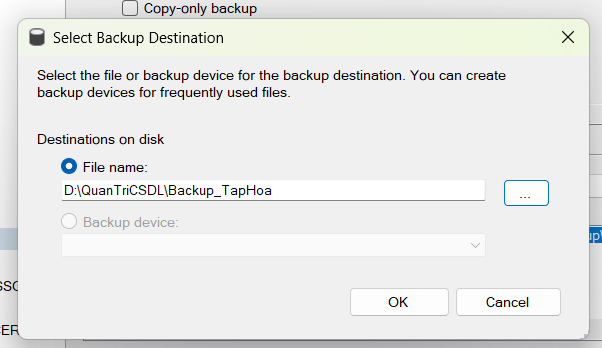
3.2 Click vào  ‘...’



3.4 Chọn D:\QuanTriCSDL🡪 Đặt File name : Backup\_TapHoa

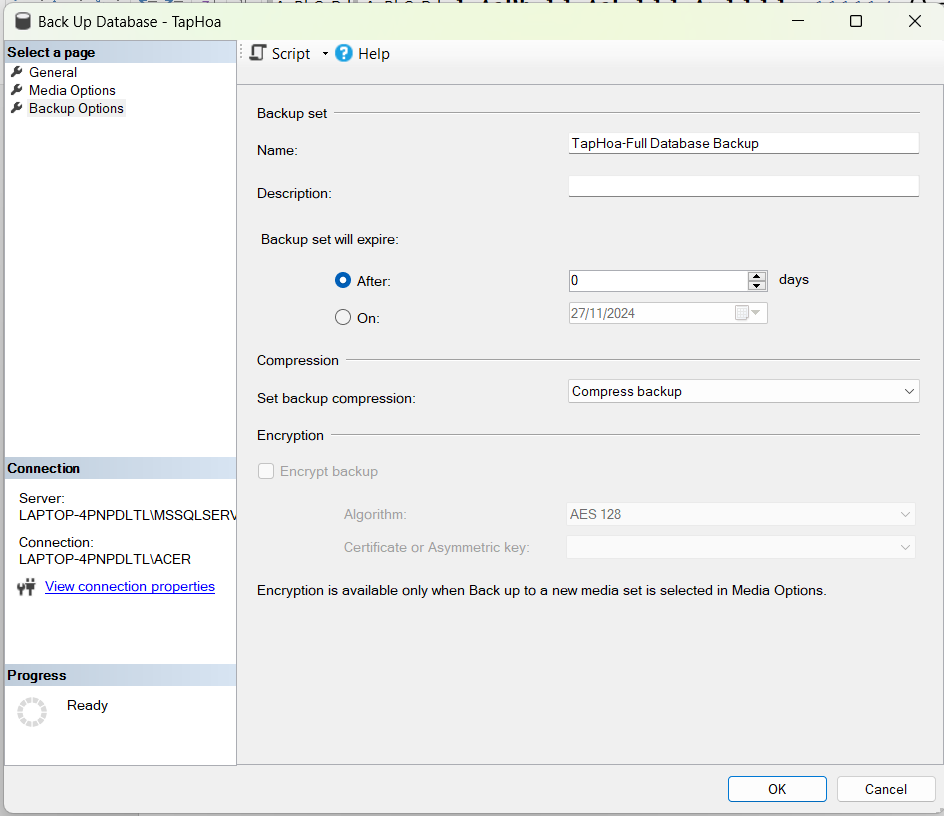


3.5 Click OK

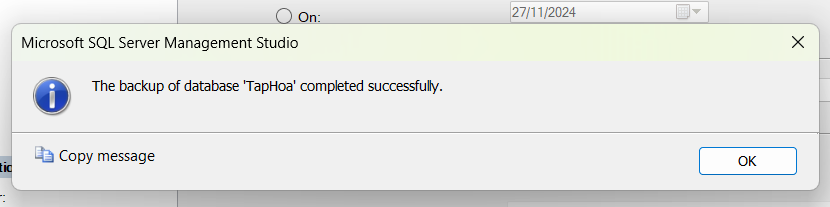


B4. Chọn loại sao lưu

4.1 Vào Backup Options🡪Chọn Compress backup (Chọn loại nén để tối ưu dung lượng)

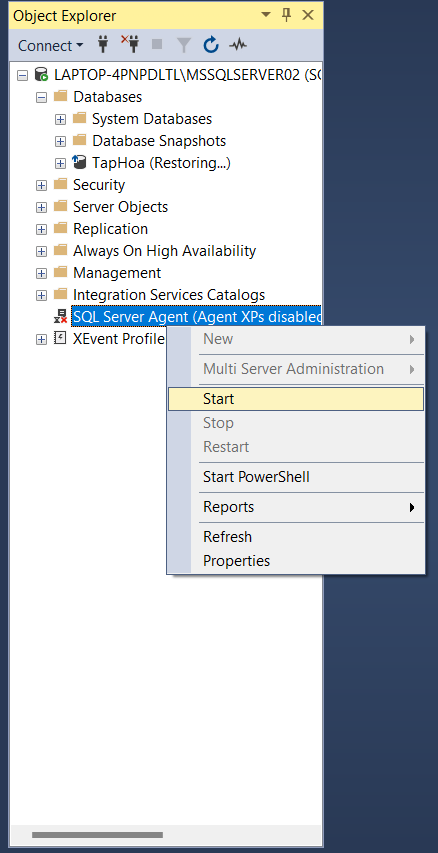


4.2 Click OK



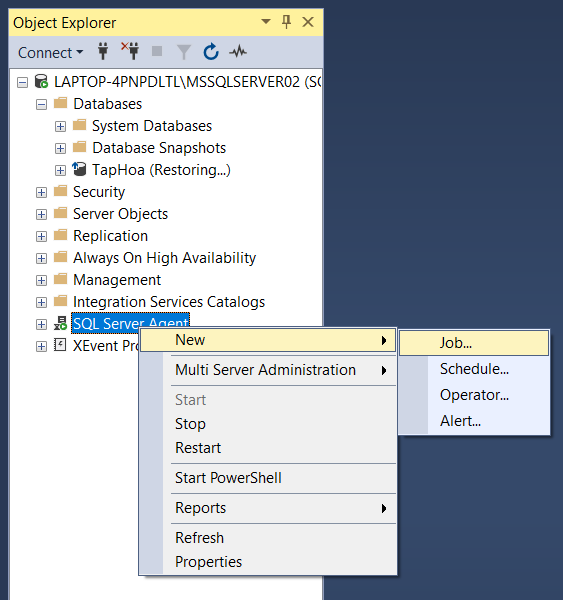
## Thiết lập thời gian thực hiện công việc sao lưu tự động

B1. Bật SQL Server Agent



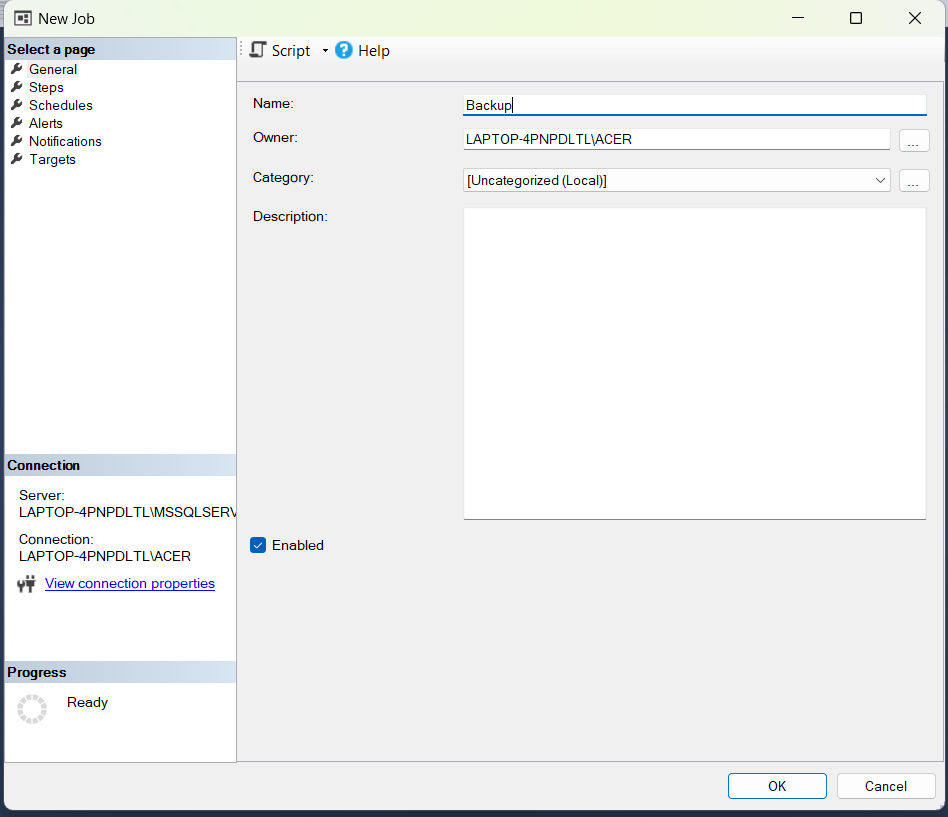
B2. Tạo job mới

2.1 Click vào SQL Server Agent🡪New🡪Job



B3. Tạo các step trong job

3.1 Ở General 🡪 Đặt tên: Name là Backup



3.2 Ở Steps

3.2.1 Click vào New



3.2.2 Đặt: Step name: Backup

Database: TapHoa

Command: BACKUP DATABASE TAPHOA

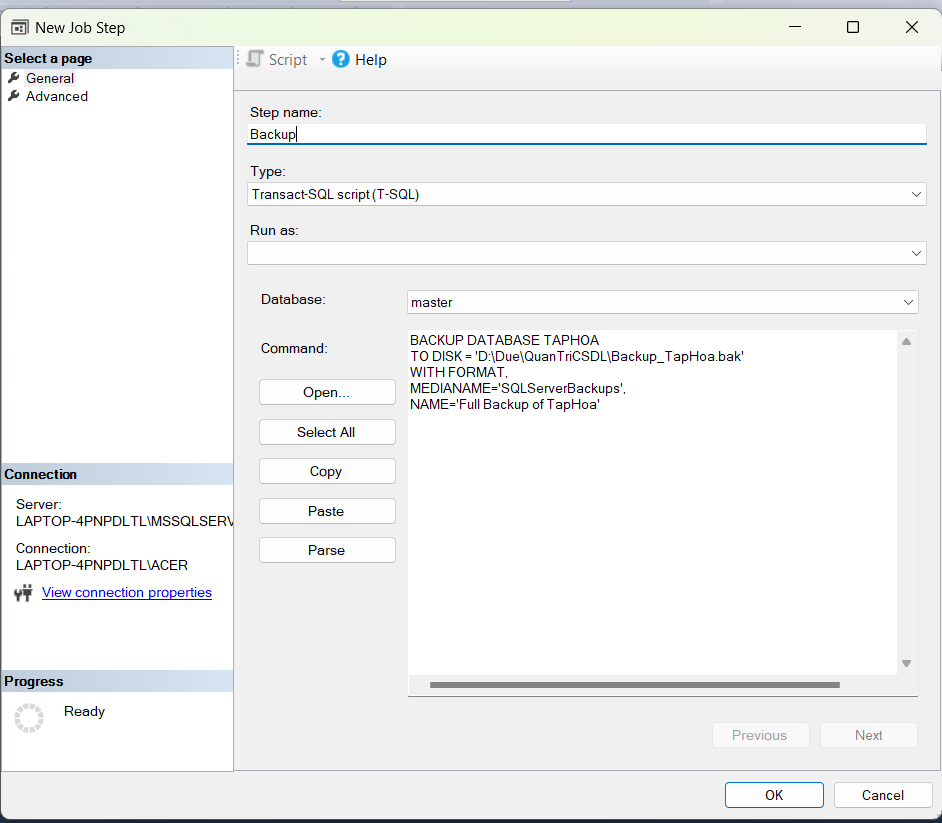
TO DISK = 'D:\Due\QuanTriCSDL\Backup\_TapHoa.bak'

WITH FORMAT,

MEDIANAME='SQLServerBackups',

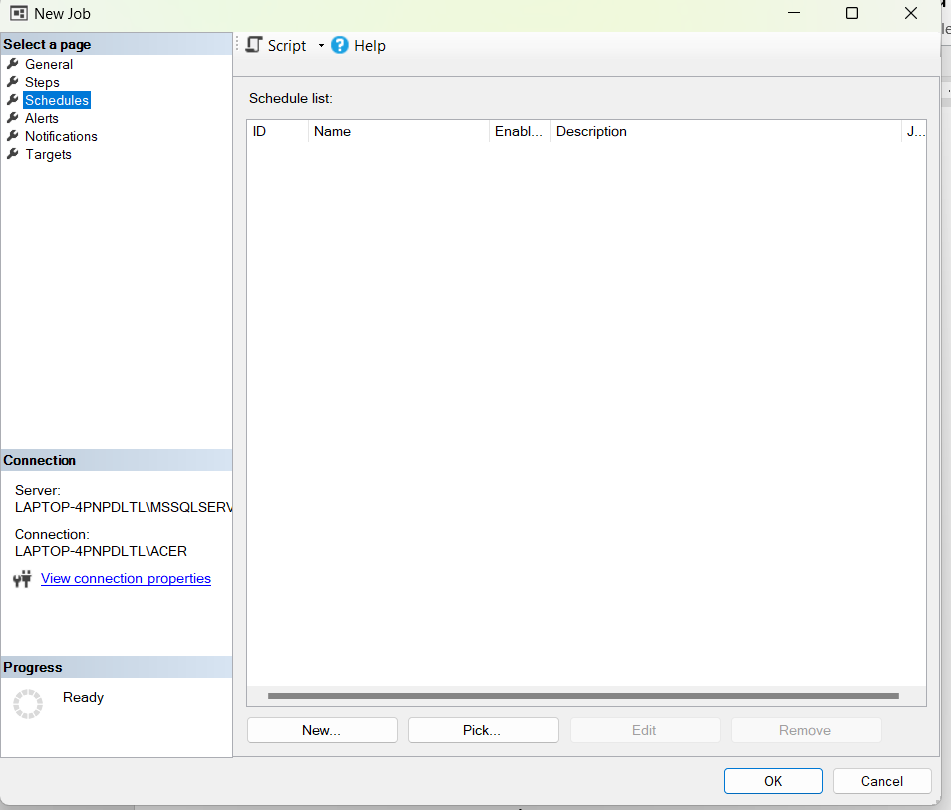
NAME='Full Backup of TapHoa'

🡪Click Ok

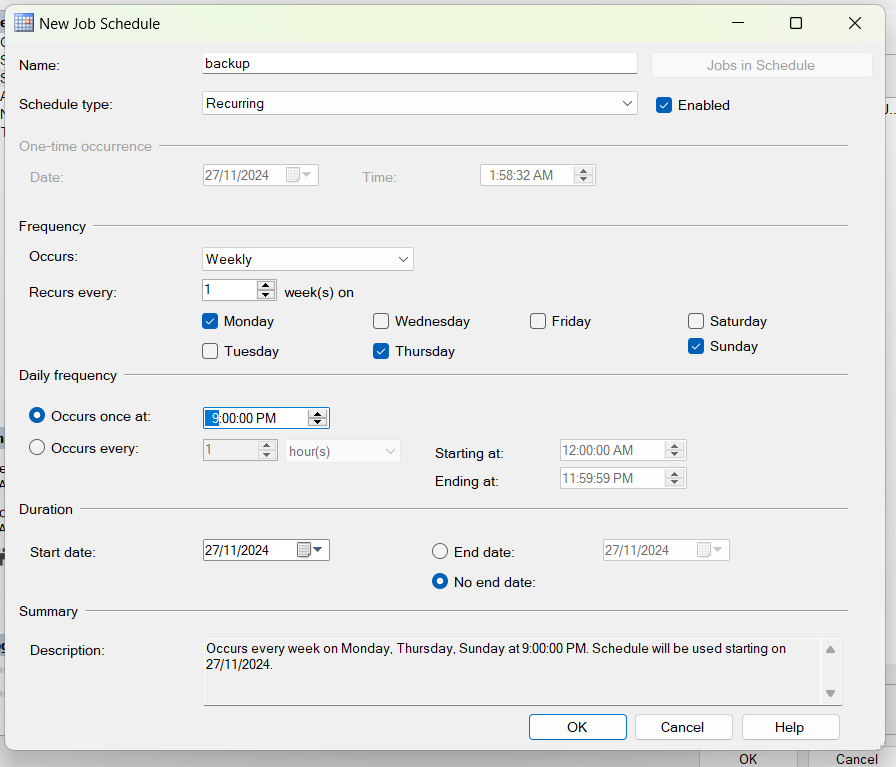


3.3 Ở Schedules

3.3.1 Click vào New



3.3.2  Name: Backup 🡪 Đặt lịch vào thứ 2, thứ 5, chủ nhật lúc 21 giờ (1 tuần/ 3 lần) như hình dưới



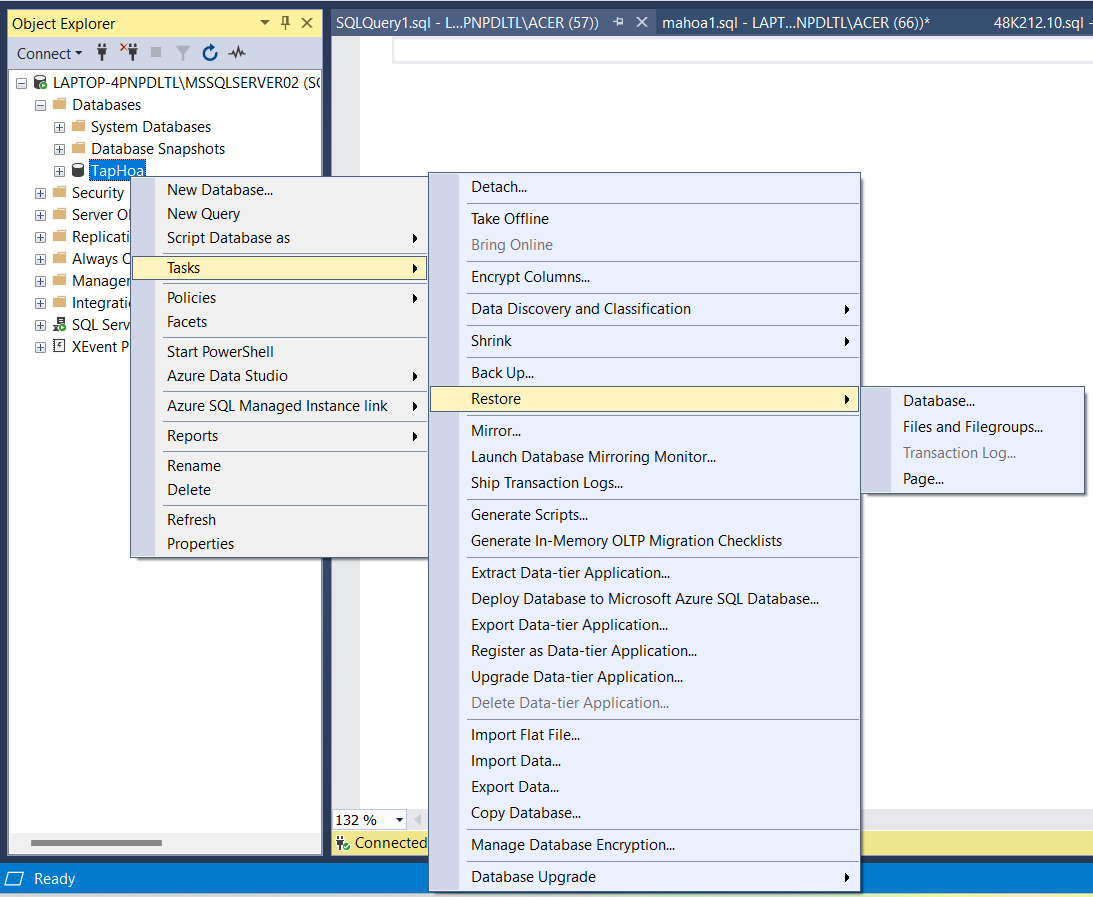
3.3.3 Click OK



## Các bước khôi phục dữ liệu

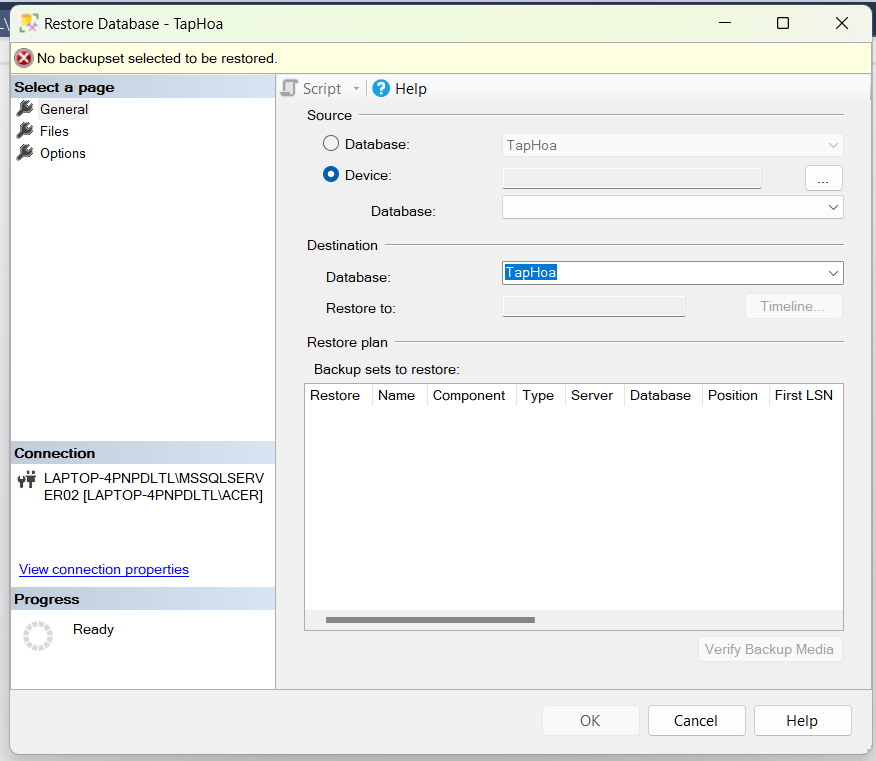
B1.  Lựa chọn loại phục hồi

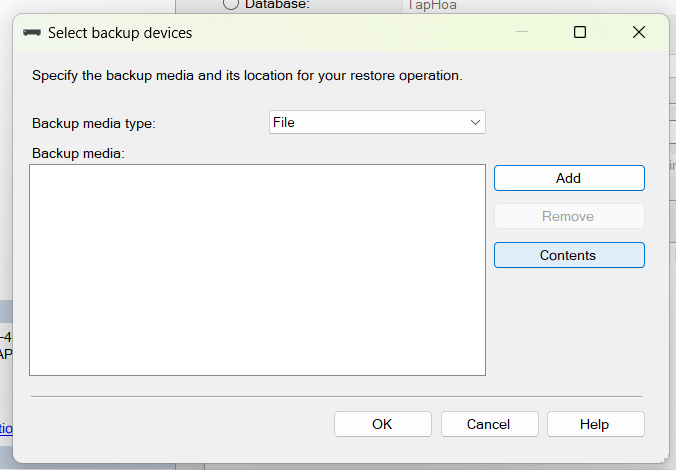
B1.1 Click chuột phải vào database cần restore (TapHoa)

B1.2 Chọn Task 🡪 Restore🡪 Database…

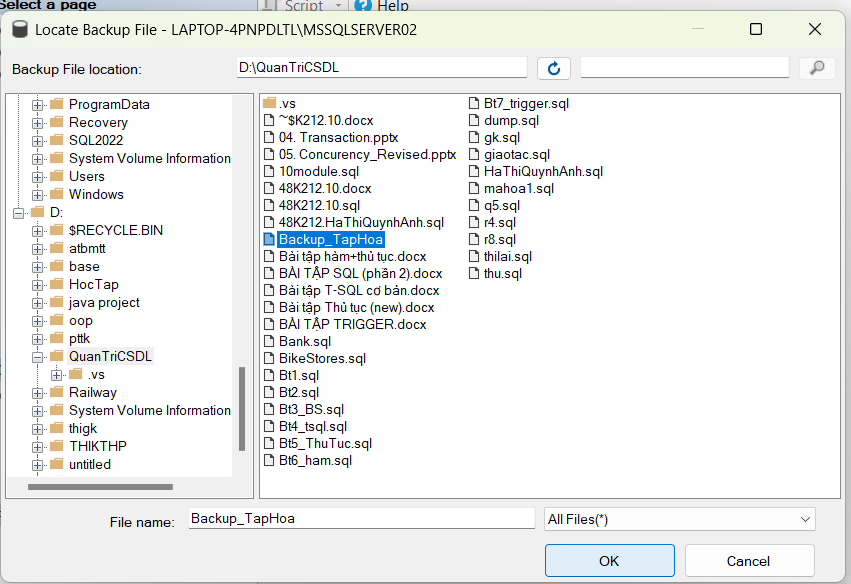
B2. Thiết lập tùy chọn phục hồi

2.1 Click vào Device 🡪 Click vào dấu ‘...’

2.2 Click Add



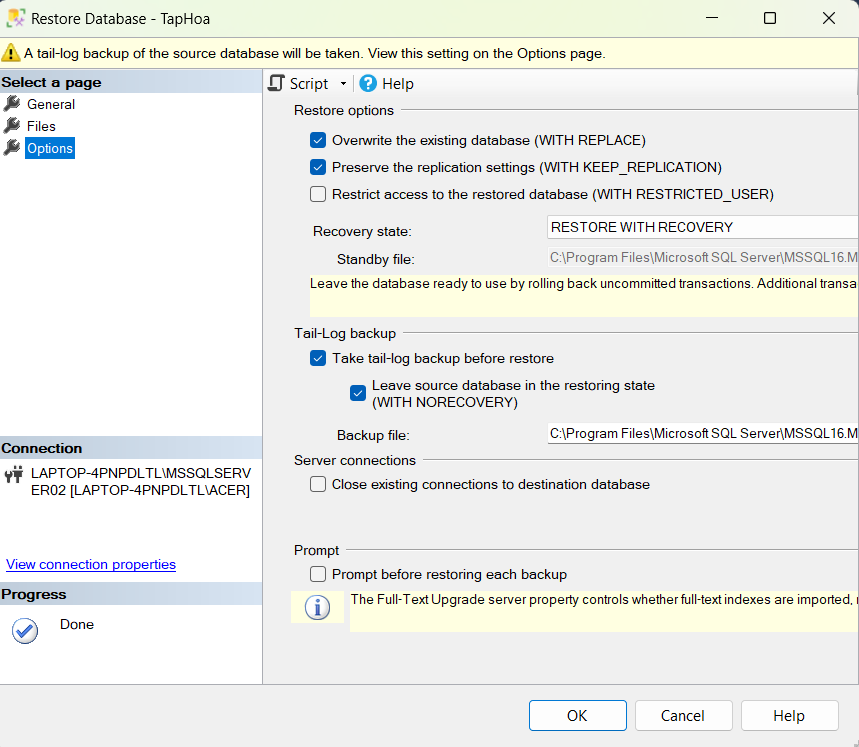
2.3 Chọn file Backup\_TapHoa.bak🡪 Click Ok



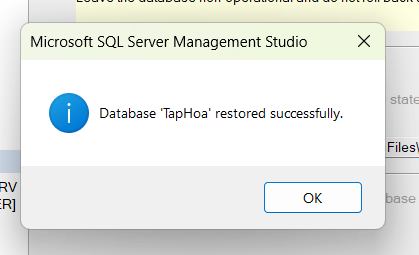
2.4 Click OK



2.5 Vào Options 🡪 Click Overwrite the existing database 🡪Click Preserve the replication settings🡪 Chọn RESTORE WITH RECOVERY🡪 Click OK



2.6 Click OK



# Giai đoạn R7: Phương án giải pháp cho vấn đề dữ liệu lớn, không đủ lưu trữ trong một ổ đĩa

## Lý do lựa chọn giải pháp

Sau một quá trình tìm hiểu và phân tích, nhóm chúng em xin được đề xuất phương pháp có tính khả thi cao để giải quyết tốt nhất cho vấn đề được đặt ra đó chính là sử dụng Azure SQL Database – là một dịch vụ cơ sở dữ liệu thông minh, mở rộng trên hệ thống đám mây, cung cấp khả năng tương thích rộng nhất.

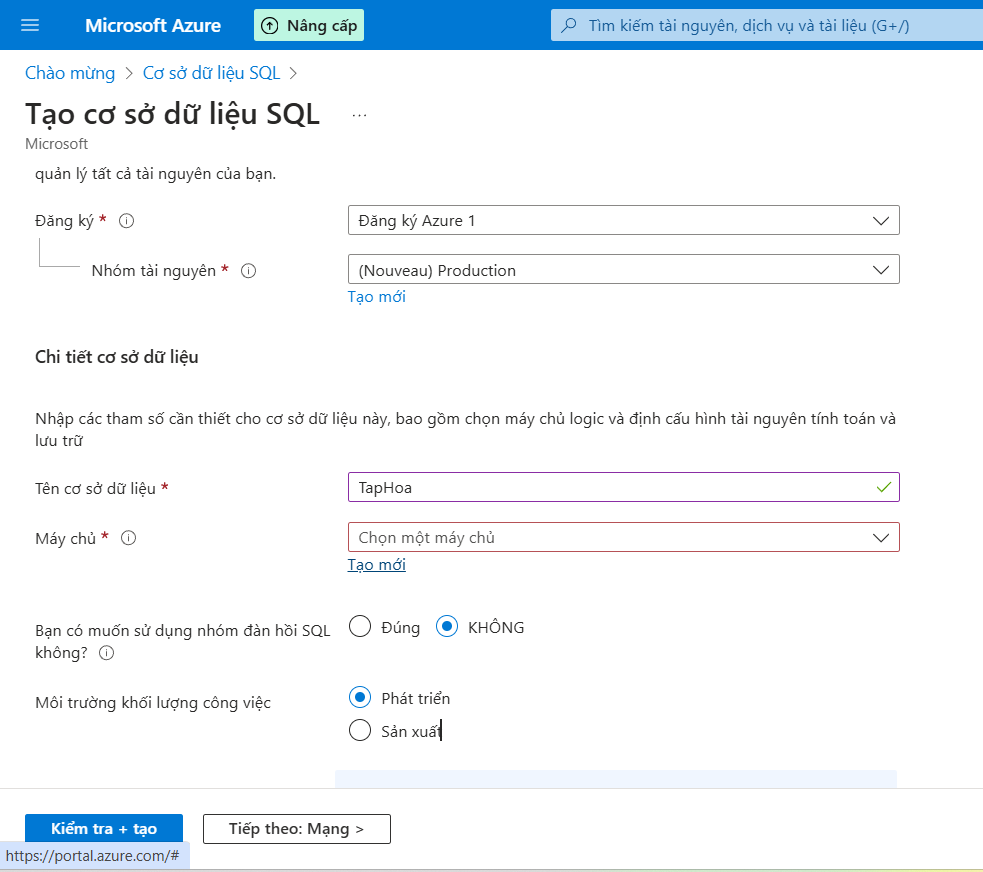
Các tính năng chính của Azure Sql database

* Chuyển đổi dễ dàng, chuyển dữ liệu trơn tru, mà không cần thay đổi code với giải pháp TCO hàng đầu.
* Tích hợp machine learning, mang lại hiệu suất cơ sở dữ liệu cao và ổn định cao nhất với công nghệ AI an toàn đáng tin cậy và đã được kiểm chứng, giúp tối ưu hiệu suất và bảo mật.
* Quy mô lớn, tính khả dụng cao cho điện toán và lưu trữ mà không giảm hiệu suất
* Bảo mật dữ liệu nâng cao bao gồm phát hiện và phân loại dữ liệu, đánh giá lỗ hổng và phát hiện mối đe dọa nâng cao tất cả trong một bảng điều khiển.

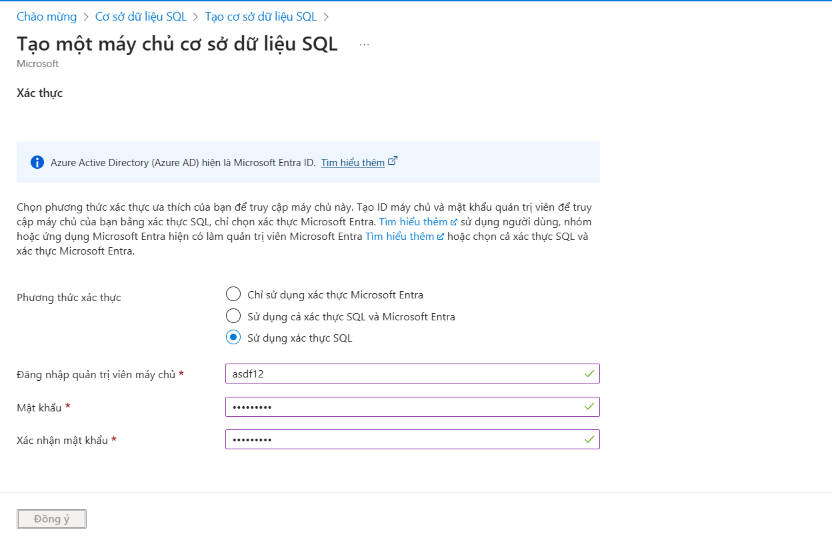
## Thực hiện triển khai

### Đưa dữ liệu lên Microsoft Azure

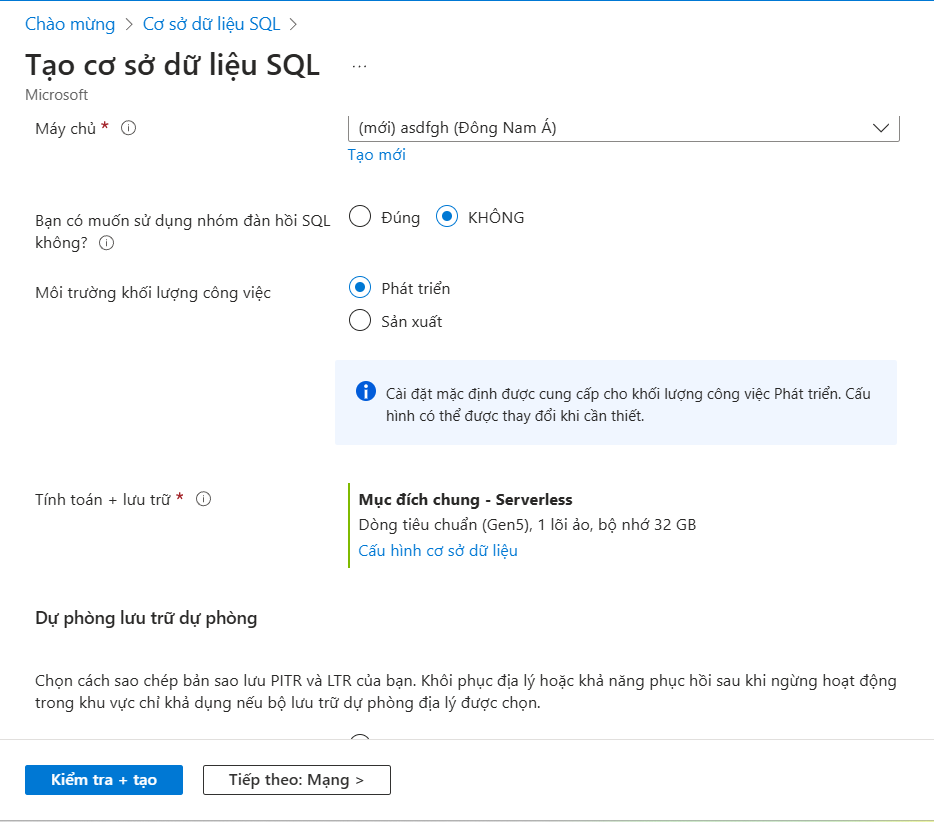
B1. Chọn nhóm tài nguyên: Production🡪 Đặt tên cơ sở dữ liệu: TapHoa🡪 Máy chủ: Chọn Tạo mới

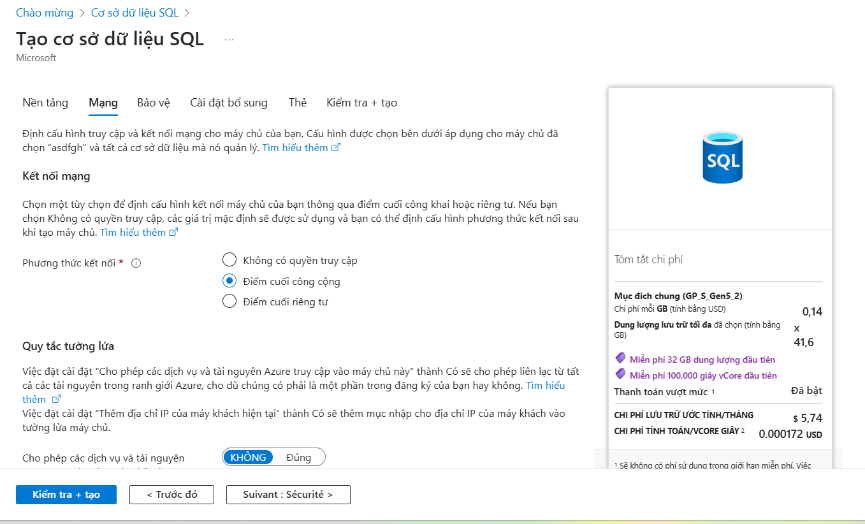


2.1 Nhập tên đăng nhập quản trị viên máy chủ: asdf12🡪 Nhập mật khẩu: Admin123🡪 Xác nhận lại mật khẩu: Admin123🡪Click Đồng ý

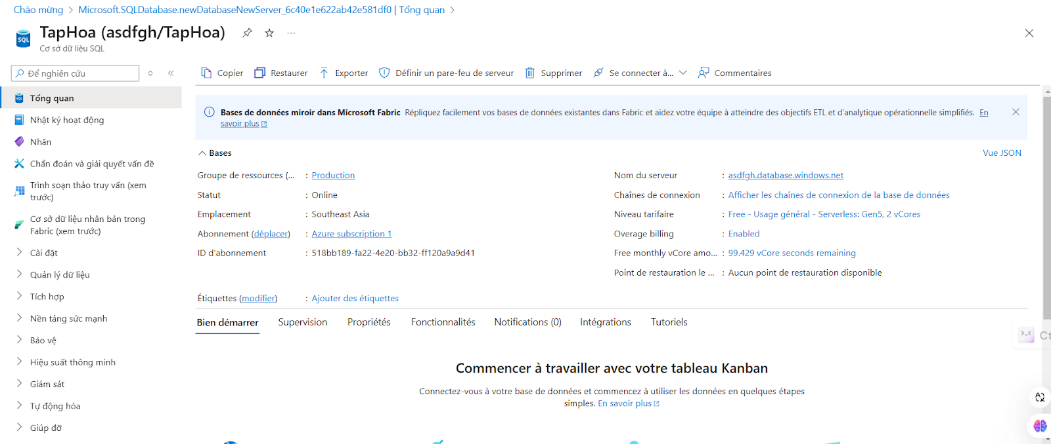


B2. Chọn nơi để máy chủ: Đông Nam Á và click vào những mực như 2 hình dưới🡪 Click Kiểm tra + tạo



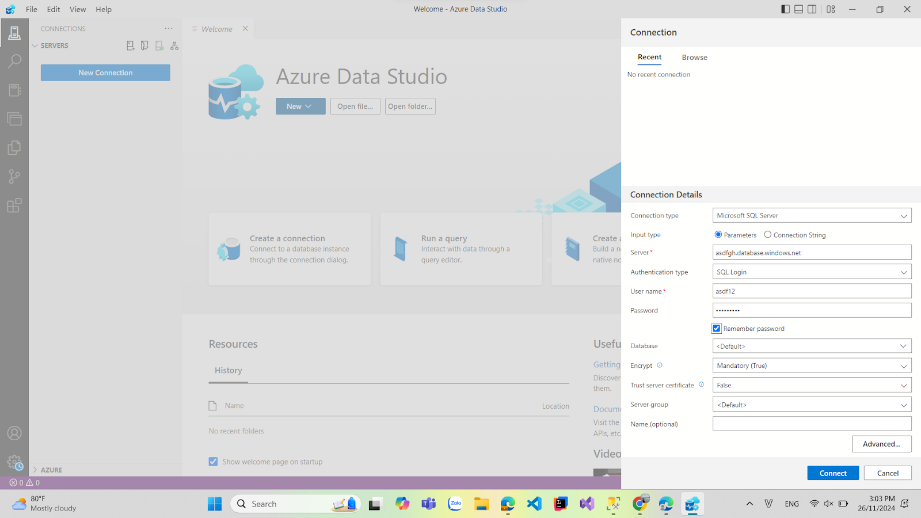


B3. Sao chép Nom du serveur (Tên server)

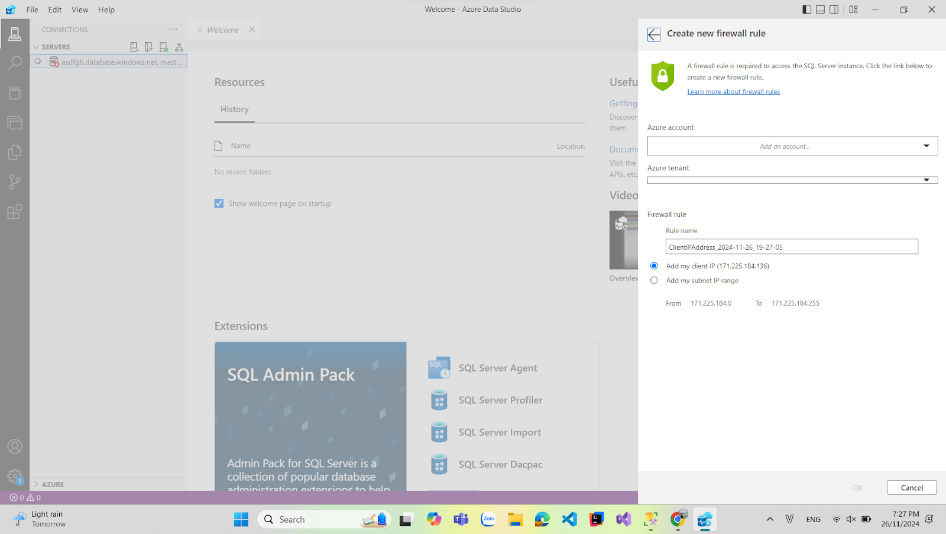


B4. Truy cập vào Azure Data Studio

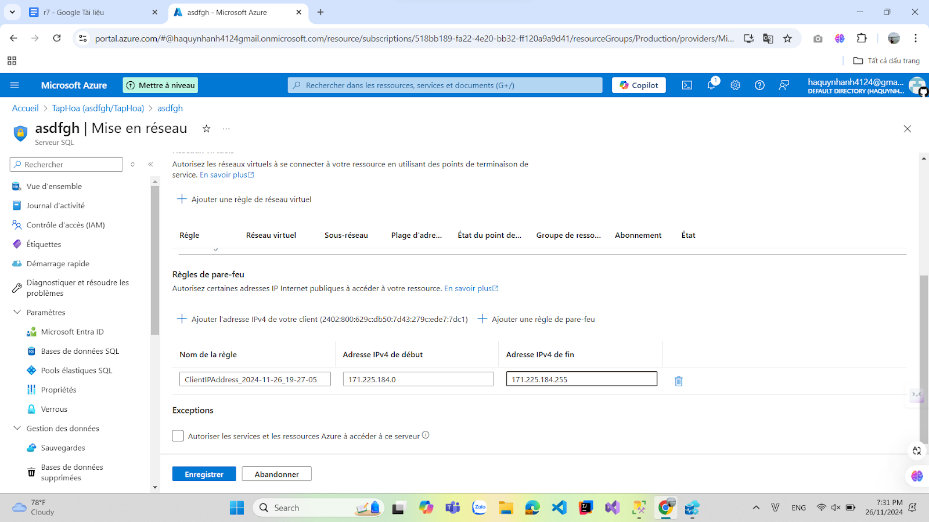
4.1 Click vào New connection🡪 Nhập các thông tin như hình dưới🡪 Click Connect



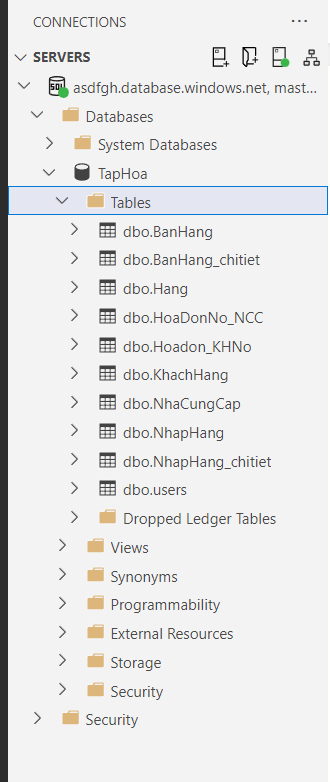
4.2 Click vào database asdfgh sẽ hiện thông tin như hình dưới đây



4.3 Vào lại Microsoft Azure để cấp quyền truy cập🡪 Thực hiện như hình dưới🡪 Click vào Enregister



4.4 Chạy lại dữ liệu trên Azure Data Studio🡪 Kết quả



### Phân quyền

B1. Phân quyền cho Admin

1.1 Chọn TapHoa🡪 Chọn Users🡪 Chọn New User

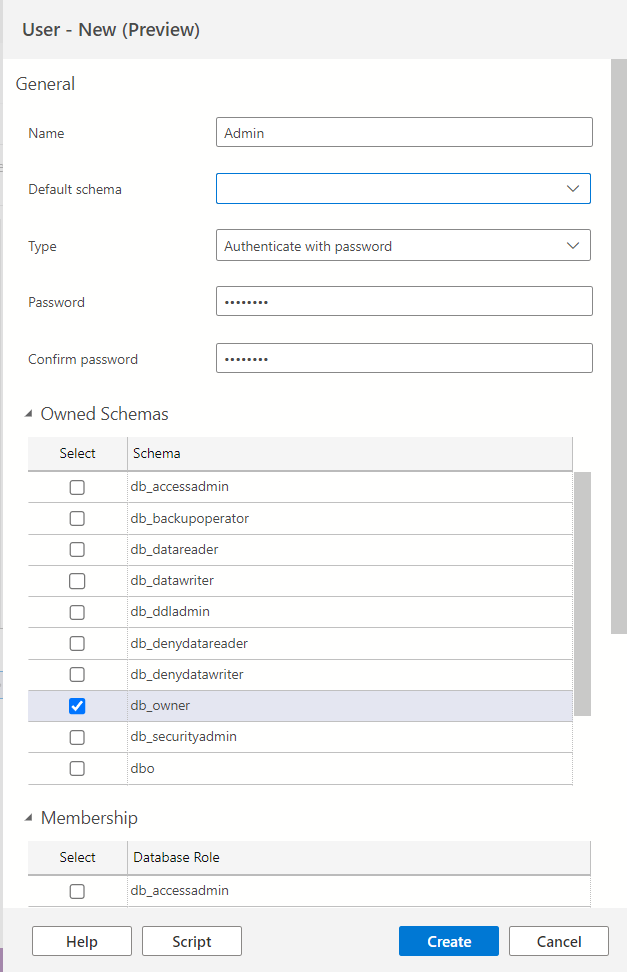


1.2 Cấp toàn bộ quyền cho Admin

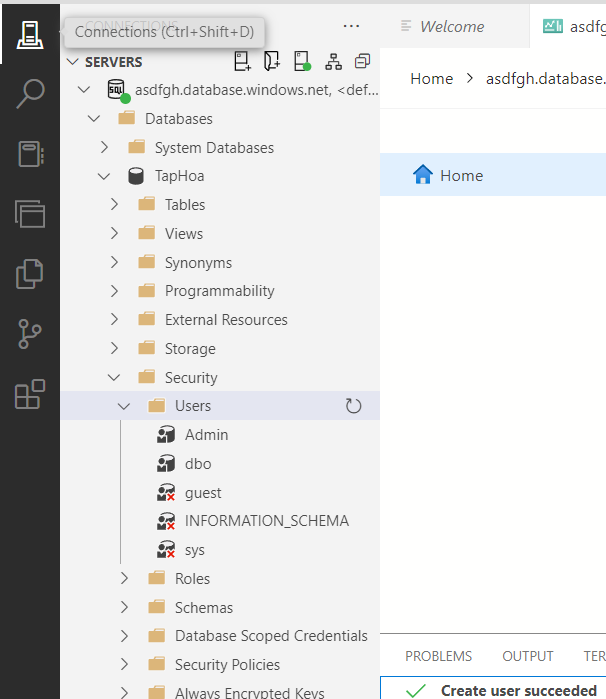
Tên tài khoản: Admin

Mật khẩu: Admin123

🡪Click Create

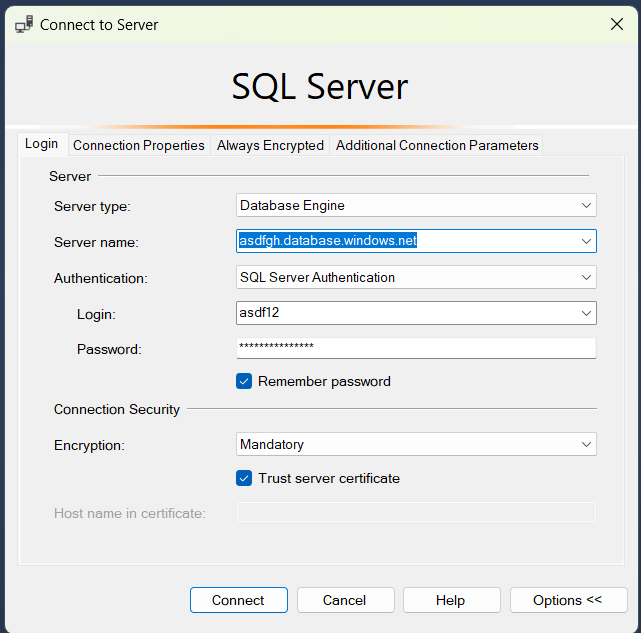


🡪Màn hình phân quyền thành công

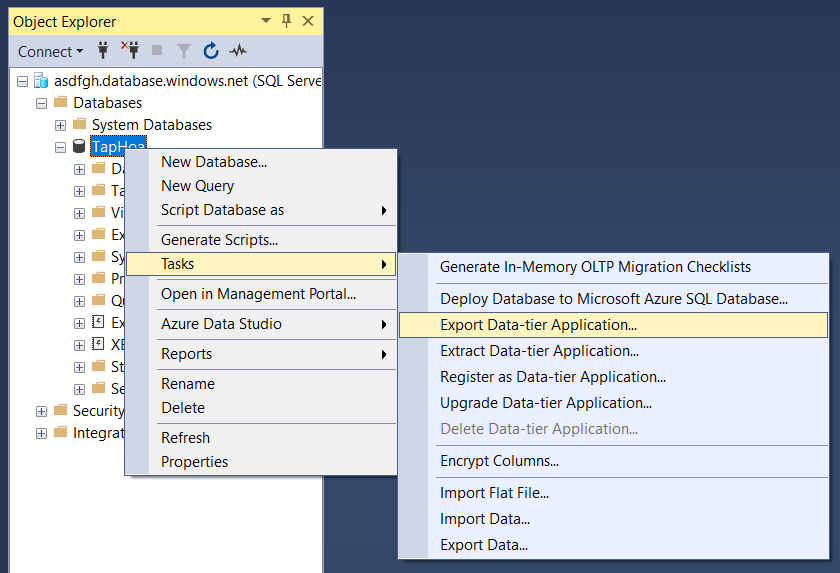


### Backup dữ liệu trên sql server

B1. Đăng nhập tài khoản Azure và sql server



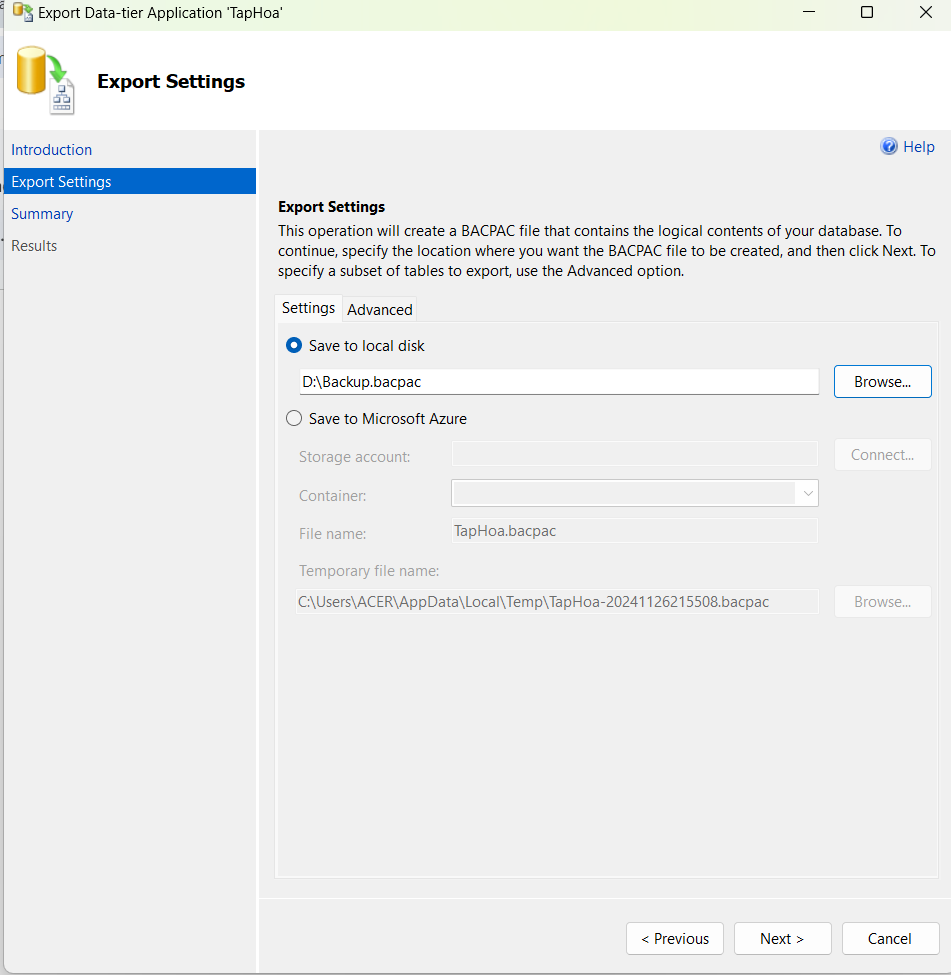
B2. Chọn database cần backup (TapHoa)🡪Tasks🡪 Export Data Aplication…



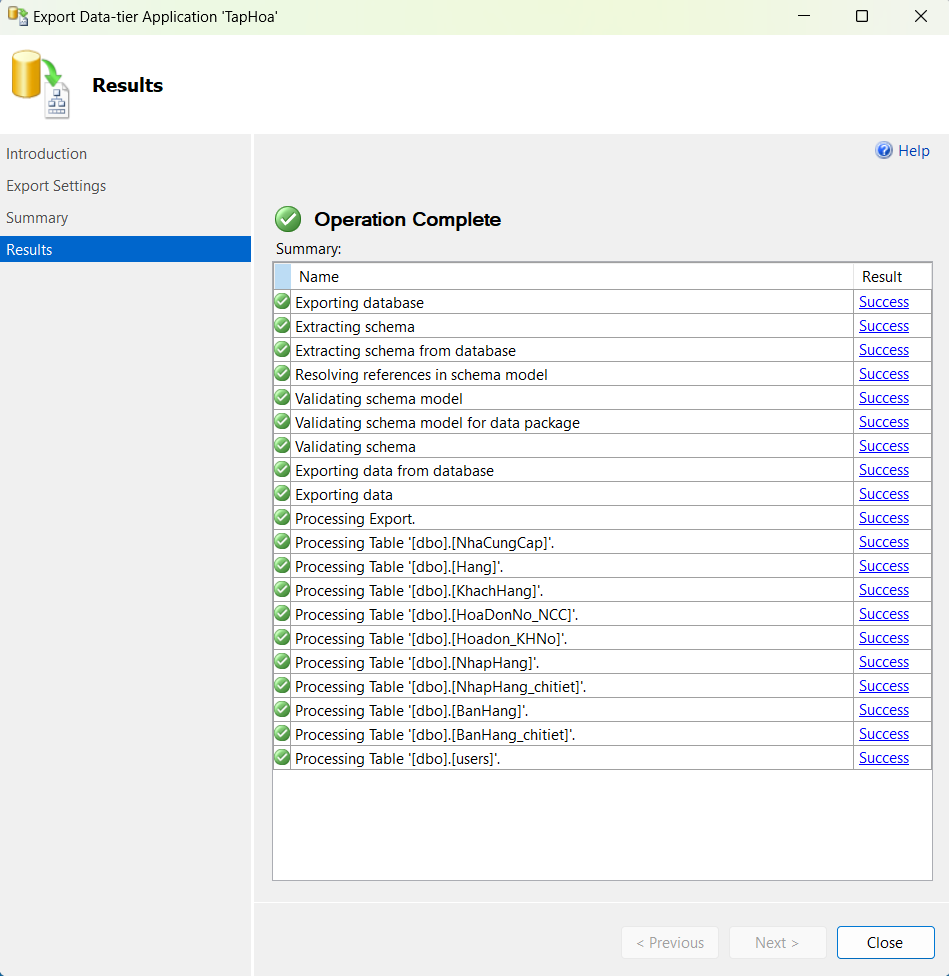
B3 Chọn Next



B4. Chọn Browse 🡪 Chọn tên file: D:\Backup.bacpac🡪 Chọn Next



B5. Chọn close



### Backup tự động trên Azure Microsoft

Vì Azure Microsoft cung cấp sao lưu tự động mặc định nên nhóm chỉ thực hiện việc chọn thời gian lưu trữ các bản sao lưu

B1. Chọn Sao lưu trong quản lý dữ liệu

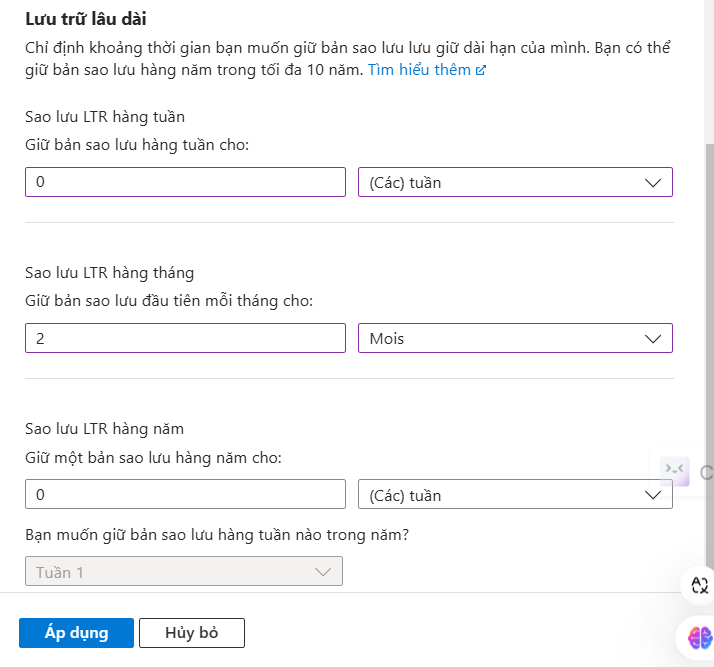


B2. Chọn thời gian lưu trữ các bản sao lưu

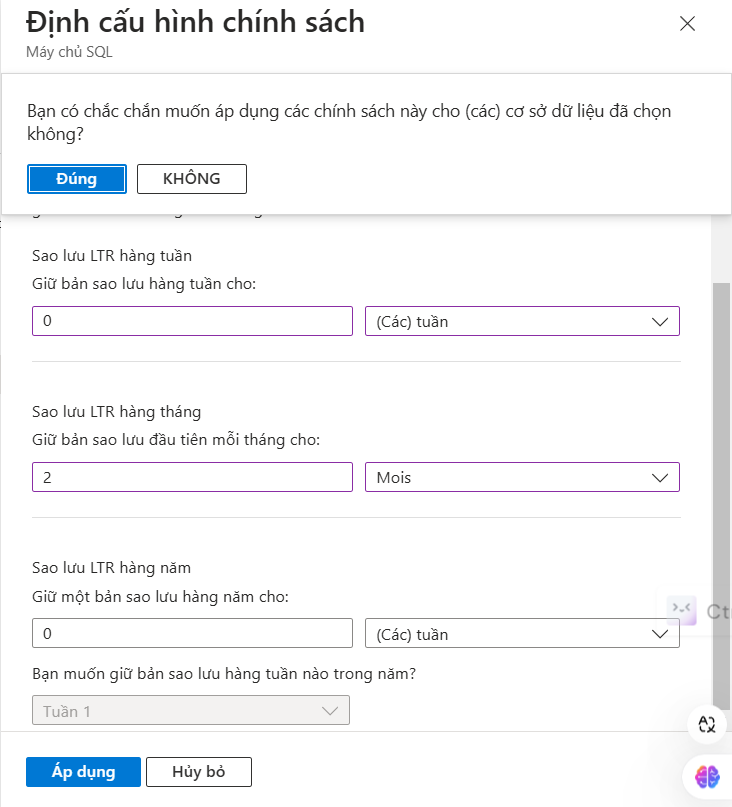
2.1 Mỗi bản sao lưu sẽ được giữ trong 3 ngày rồi xoá được thực hiện như hình dưới



2.2 Đối với các bản sao lưu đầu tiên trong 1 tháng sẽ được giữ trong vòng 2 tháng rồi mới xoá thực hiện như hình dưới🡪 Chọn Áp dụng



2.3 Nhấn Đúng để xác định lại những thao tác đã thực hiện



# Giai đoạn R8: Phương án hạn chế tấn công SQL Injection

## Khái niệm

SQL Injection là một kỹ thuật lợi dụng những lỗ hổng về câu truy vấn của các ứng dụng. Được thực hiện bằng cách chèn thêm một đoạn SQL để làm sai lệnh đi câu truy vấn ban đầu, từ đó có thể khai thác dữ liệu từ database. SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác như một người quản trị web, trên cơ sở dữ liệu của ứng dụng. Các chức năng đăng nhập, tìm kiếm, xóa là những mục tiêu tấn công chính của các kẻ tấn công.

## Cách phòng chống

Mã hoá dữ liệu

Sử dụng Stored Procedures kết hợp với kiểm tra và lọc đầu vào

* Stored Procedures (Tách biệt mã SQL và dữ liệu đầu vào):
* Stored Procedures thường xử lý dữ liệu đầu vào thông qua các tham số, thay vì gắn trực tiếp các giá trị vào câu lệnh SQL. Điều này ngăn không cho dữ liệu đầu vào ảnh hưởng đến cấu trúc câu lệnh SQL.
* Do không cho phép chèn trực tiếp các mã SQL, hacker khó có thể thay đổi được logic của câu lệnh SQL trong Stored Procedures
* Kiểm tra và lọc đầu vào
* Kiểm tra loại dữ liệu: Đảm bảo dữ liệu đầu vào đúng loại (như số, chuỗi, ngày tháng, v.v.) trước khi truyền vào hệ thống.
* Giới hạn độ dài dữ liệu: Hạn chế số ký tự tối đa mà dữ liệu đầu vào có thể chứa.
* Loại bỏ các ký tự nguy hiểm: Sử dụng các hàm chuẩn hóa hoặc mã hóa dữ liệu để loại bỏ các ký tự đặc biệt như ', ", ; , --, hoặc các chuỗi nguy hiểm khác.

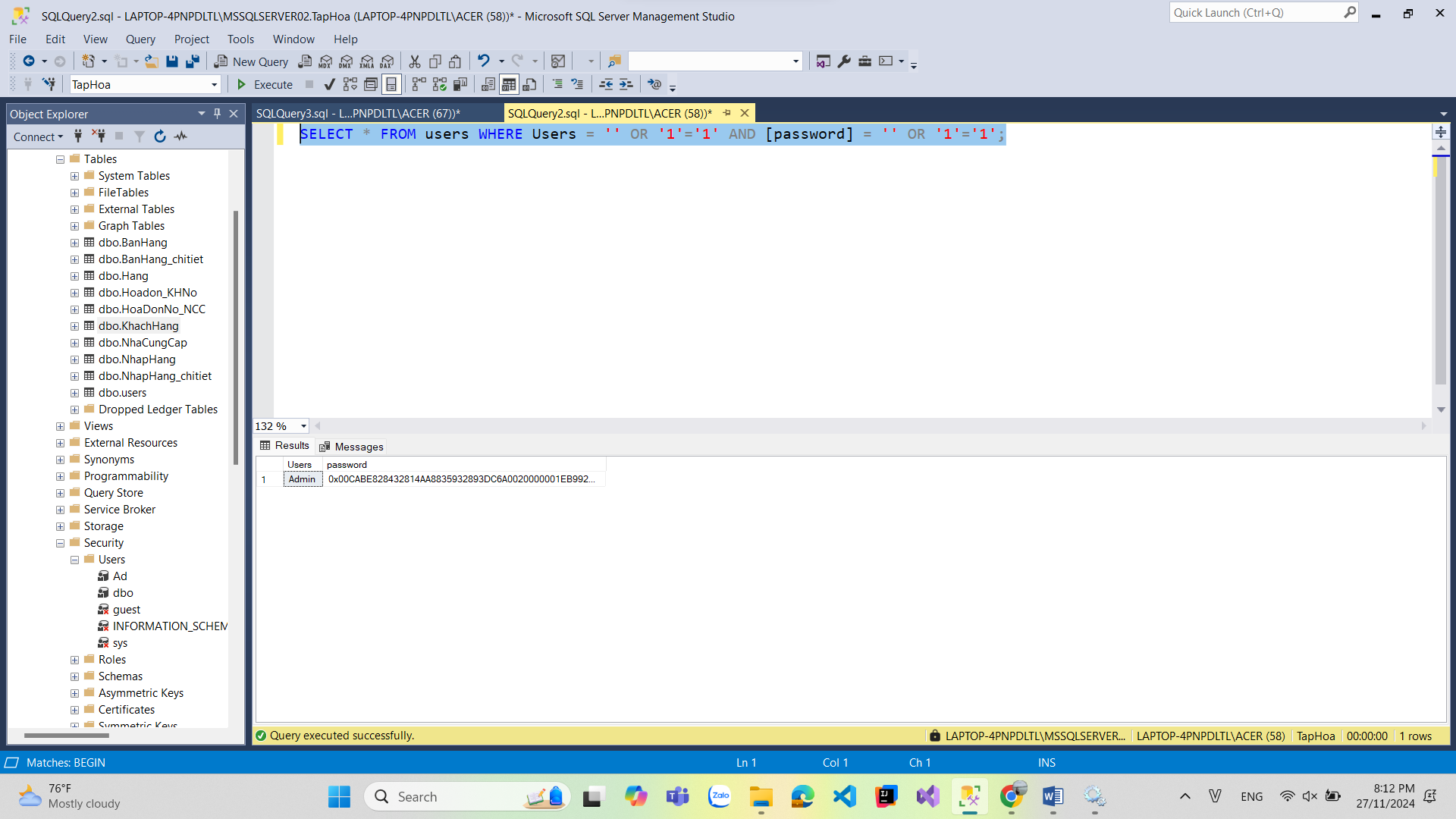
## Các cách tấn công

### Tấn công vào chức năng đăng nhập

Tấn công **Bypass Authentication**

Mục tiêu: Kẻ tấn công muốn đăng nhập vào hệ thống mà không cần cung cấp tài khoản và mật khẩu hợp lệ.

SELECT \* FROM users WHERE Users = '' OR '1'='1' AND [password] = '' OR '1'='1'



Giải pháp:

CREATE PROC DangNhap

@Users NVARCHAR (50),

@passwords NVARCHAR (50)

AS

BEGIN

-- Kiểm tra độ dài và ký tự không hợp lệ

IF LEN(@Users) = 0 OR LEN(@Passwords) = 0

BEGIN

PRINT N'Tên người dùng hoặc mật khẩu không được để trống.';

RETURN;

END

IF PATINDEX('%[^a-zA-Z0-9]%', @Users) > 0 OR PATINDEX('%[^a-zA-Z0-9]%', @Passwords) > 0

BEGIN

PRINT N'Ký tự không hợp lệ trong Tên người dùng hoặc Mật khẩu';

RETURN;

END

BEGIN

-- Thực hiện truy vấn an toàn, không có SQL Injection

SELECT \*

FROM users

WHERE Users = @Users

AND [password] = @passwords;

END

END

GO

---

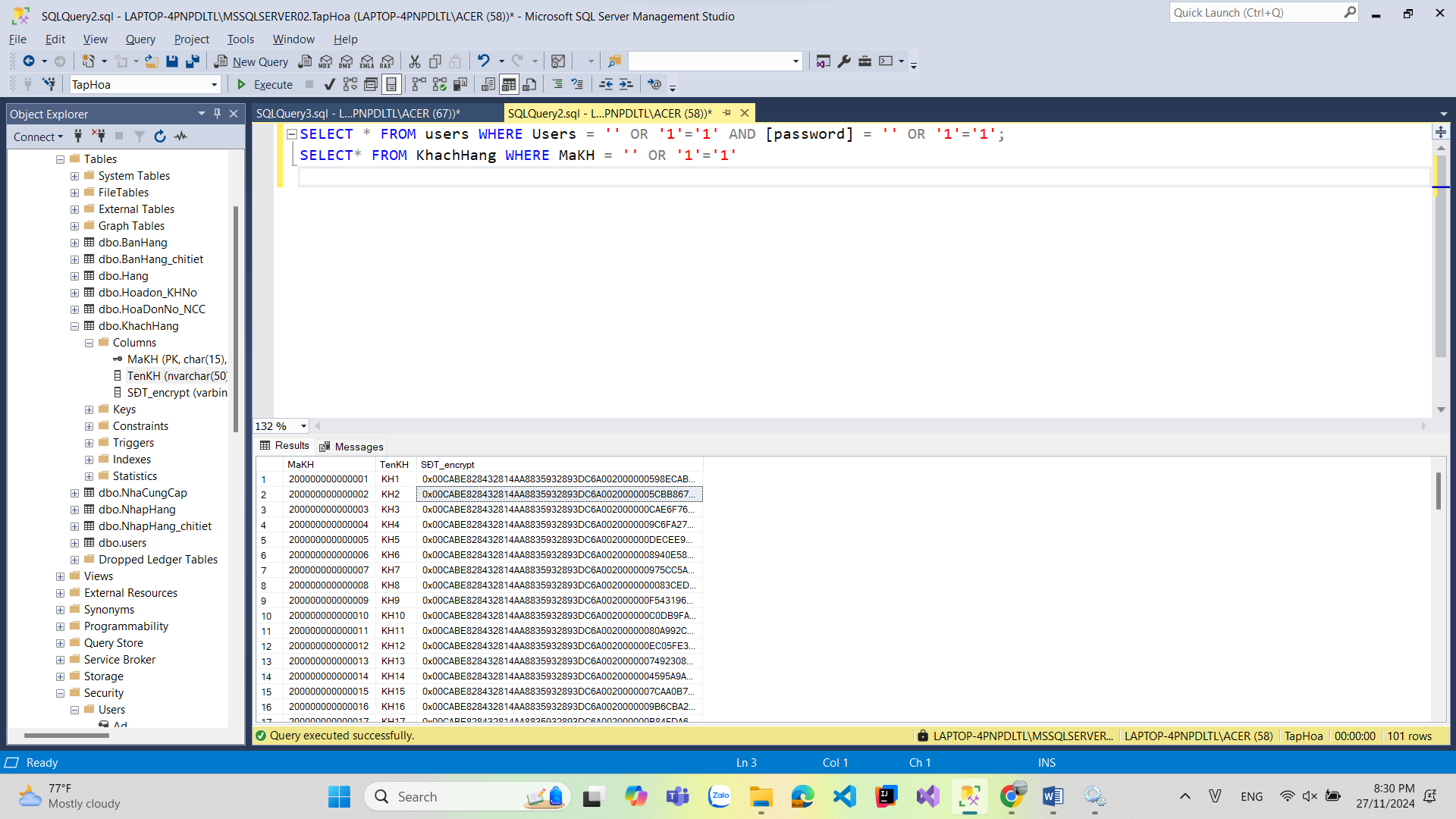
EXEC DangNhap @Users = ''' OR ''1''=''1''--', @Passwords= ''' OR ''1''=''1''--'

### Tấn công vào chức năng Select (Tìm kiếm)

##### Tìm kiếm thông tin của khách hàng trong bảng KhachHang

Mục tiêu: Kẻ tấn công muốn tìm ra tất cả thông tin của khách hàng mà không cần có quyền truy cập hợp lệ vào hệ thống.

SELECT \*FROM KhachHang WHERE MaKH = '' OR '1'='1'



Giải pháp:

CREATE PROCEDURE GetSdtByMaKH @MaKH CHAR (15)

AS

BEGIN

-- Kiểm tra độ dài và ký tự hợp lệ

IF LEN(@MaKH) <> 15 OR PATINDEX('%[^a-zA-Z0-9]%', @MaKH) > 0

BEGIN

PRINT N' Đầu vào không hợp lệ';

RETURN;

END

-- Truy vấn dữ liệu an toàn

SELECT \*

FROM KhachHang

WHERE MaKH = @MaKH

END

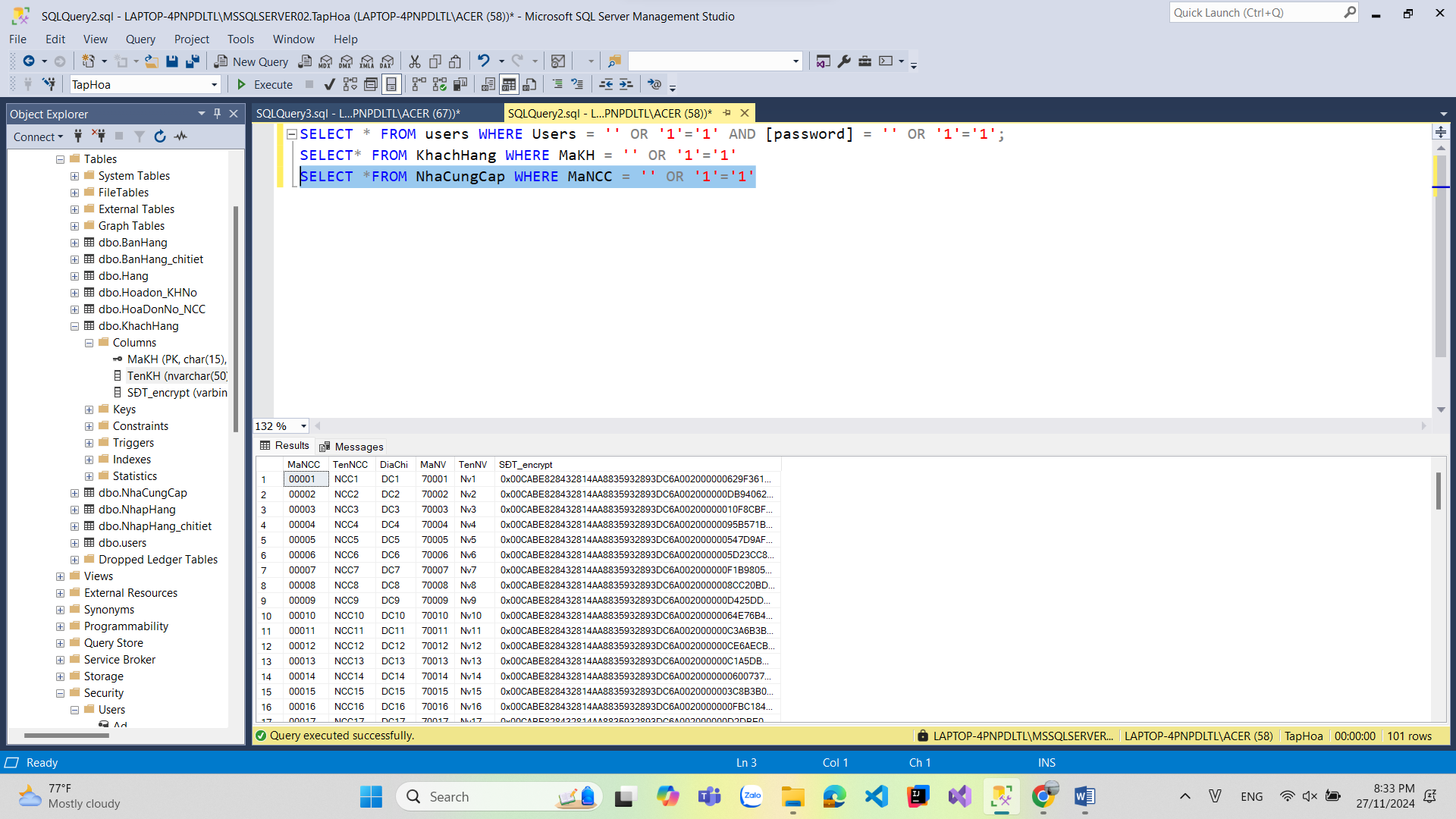
---

EXEC GetSdtByMaKH ' OR 1=1 --'

##### Tìm kiếm thông tin của nhà cung cấp trong bảng NhaCungCap

Mục tiêu: Kẻ tấn công muốn tìm ra tất cả thông tin của nhà cung cấp mà không cần có quyền truy cập hợp lệ vào hệ thống.

SELECT \*FROM NhaCungCap WHERE MaNCC = '' OR '1'='1'



Giải pháp:

CREATE PROCEDURE GetSdtByMaNCC @MaNCC CHAR (15)

AS

BEGIN

-- Kiểm tra độ dài và ký tự hợp lệ

IF LEN(@MaNCC) <> 15 OR PATINDEX('%[^a-zA-Z0-9]%', @MaNCC) > 0

BEGIN

PRINT N' Đầu vào không hợp lệ'

RETURN

END

-- Truy vấn dữ liệu an toàn

SELECT \*

FROM NhaCungCap

WHERE MaNCC = @MaNCC

END

---

EXEC GetSdtByMaNCC' OR 1=1 --'

### Tấn công vào chức năng Update (Cập nhật/Chỉnh sửa)

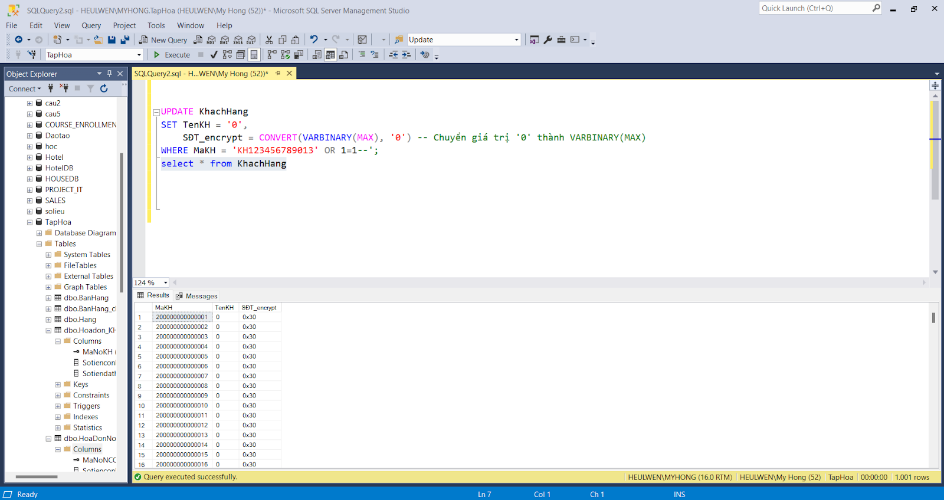
##### Cập nhật lại toàn bộ thông tin khách hàng trong bảng KhachHang

Mục tiêu: Kẻ tấn công muốn cập nhật số điện thoại khách hàng về 0 mà không cần có quyền cập nhật ở hệ thống này.

UPDATE KhachHang

SET TenKH = 'Nguyen Van A', SĐT\_encrypt = CONVERT(VARBINARY(MAX), '0')

WHERE MaKH = 'KH123456789013' OR 1=1--'



Giải pháp:

CREATE PROCEDURE UpdateKhachHang

@MaKH CHAR (15),

@TenKH NVARCHAR (50),

@SĐT\_encrypt CHAR (10)

AS

BEGIN

-- Kiểm tra số điện thoại có đúng định dạng: 10 chữ số

IF @SĐT\_encrypt NOT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

BEGIN

PRINT N'Số điện thoại khách hàng không hợp lệ.'

RETURN

END

-- Kiểm tra ký tự đặc biệt trong MaKH, TenKH và SĐT\_encrypt

IF PATINDEX ('%[^a-zA-Z0-9]%', @MaKH) > 0 OR

PATINDEX ('%[^a-zA-Z0-9 ]%', @TenKH) > 0 OR

PATINDEX ('%[^a-zA-Z0-9]%', @SĐT\_encrypt) > 0

BEGIN

PRINT N'Đầu vào chứa ký tự không hợp lệ.'

RETURN

END

-- Kiểm tra sự tồn tại của MaKH trong bảng KhachHang

IF NOT EXISTS (SELECT 1 FROM KhachHang WHERE MaKH = @MaKH)

BEGIN

PRINT N'Mã khách hàng không tồn tại.'

RETURN

END

UPDATE KhachHang

SET TenKH = @TenKH, SĐT\_encrypt = CONVERT(VARBINARY(MAX), @SĐT\_encrypt)

WHERE MaKH = @MaKH;

PRINT N'Cập nhật thông tin khách hàng thành công.'

END

GO

EXEC UpdateKhachHang

@MaKH = ''' OR ''1'' = ''1'''

@TenKH = N'Nguyễn Văn A ',

@SĐT\_encrypt = '0'

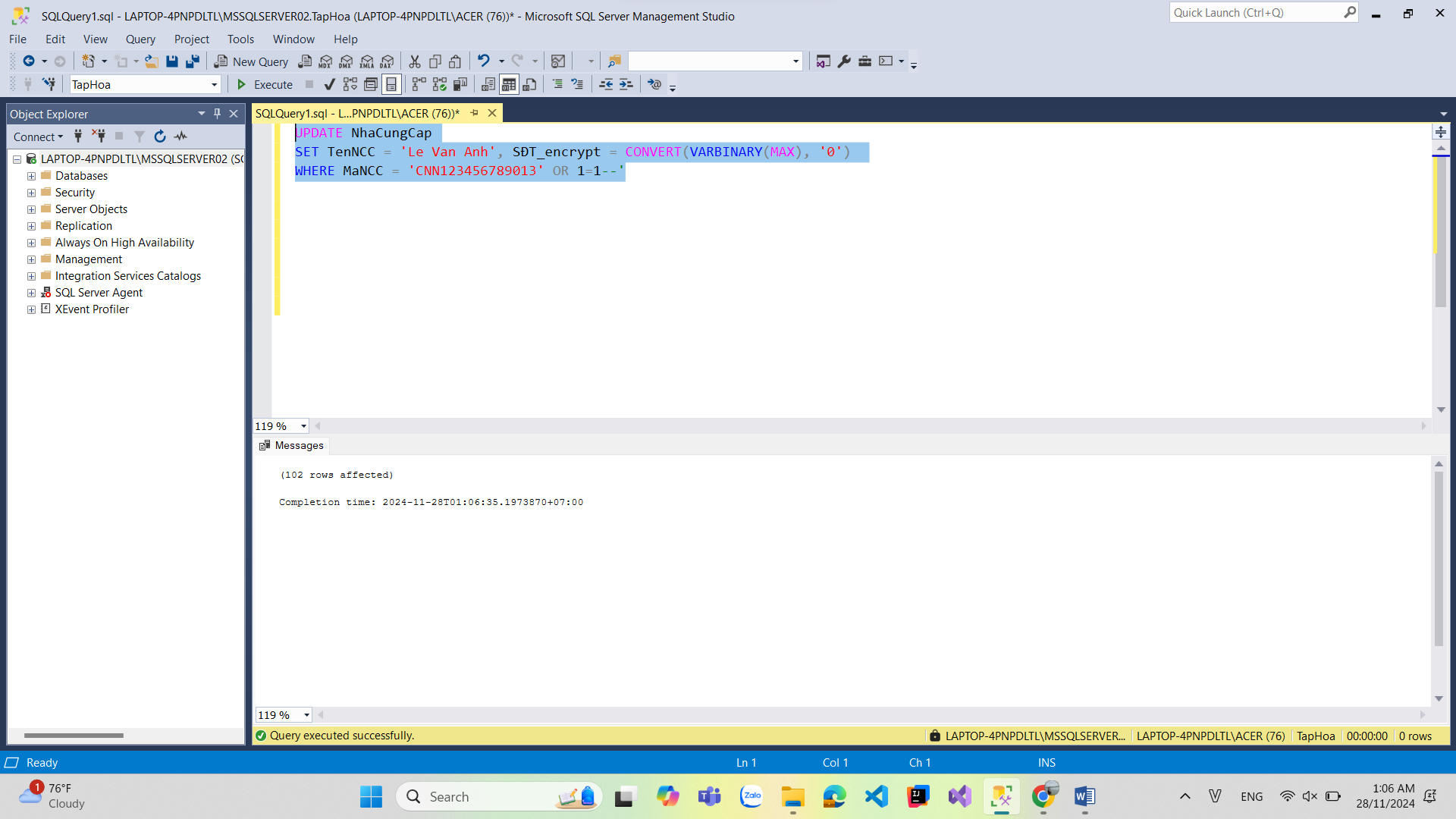
##### Cập nhật lại toàn bộ thông tin nhà cung cấp trong bảng NhaCungCap

Mục tiêu: Kẻ tấn công muốn cập nhật số điện thoại nhà cung cấp về 0 mà không cần có quyền cập nhật ở hệ thống này.

UPDATE NhaCungCap

SET TenNCC = 'Le Van Anh', SĐT\_encrypt = CONVERT(VARBINARY(MAX), '0')

WHERE MaNCC = 'CNN123456789013' OR 1=1--'



Giải pháp:

CREATE PROCEDURE UpdateNhaCungCap

@MaNCC CHAR (15),

@TenNCC NVARCHAR (50),

@DiaChi NVARCHAR (50),

@MaNV Char (15),

@TenNV NVARCHAR (50),

@SĐT\_encrypt CHAR (10)

AS

BEGIN

-- Kiểm tra số điện thoại có đúng định dạng: 10 chữ số

IF @SĐT\_encrypt NOT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

BEGIN

PRINT N' Số điện thoại nhà cung cấp không hợp lệ.'

RETURN

END

-- Kiểm tra ký tự đặc biệt trong MaNCC, TenKH và SĐT\_encrypt

IF PATINDEX('%[^a-zA-Z0-9]%', @MaNCC) > 0 OR

PATINDEX('%[^a-zA-Z0-9]%', @TenNCC) > 0 OR

PATINDEX('%[^a-zA-Z0-9]%', @DiaChi) > 0 OR

PATINDEX('%[^a-zA-Z0-9]%', @MaNV) > 0 OR

PATINDEX('%[^a-zA-Z0-9]%', @TenNV) > 0 OR

PATINDEX('%[^a-zA-Z0-9]%', @SĐT\_encrypt) > 0

BEGIN

PRINT N' Đầu vào chứa ký tự không hợp lệ.'

RETURN

END

-- Kiểm tra sự tồn tại của MaNCC trong bảng NhaCungCap

IF NOT EXISTS (SELECT 1 FROM NhaCungCap WHERE MaNCC = @MaNCC)

BEGIN

PRINT N' Mã nhà cung cấp không tồn tại.'

RETURN

END

UPDATE NhaCungCap

SET TenNCC = @TenNCC,

SĐT\_encrypt = CONVERT(VARBINARY(MAX), @SĐT\_encrypt)

WHERE MaNCC = @MaNCC;

PRINT N' Cập nhật thông tin nhà cung cấp thành công.';

END

---

EXEC UpdateNhaCungCap

@MaNCC = ''' OR ''1'' = ''1'''

@TenNCC = 'Le Van Anh',

@DiaChi = 'diachi',

@MaNV = 'NV123456789',

@TenNV = 'tenNV',

@SĐT\_encrypt = '0'

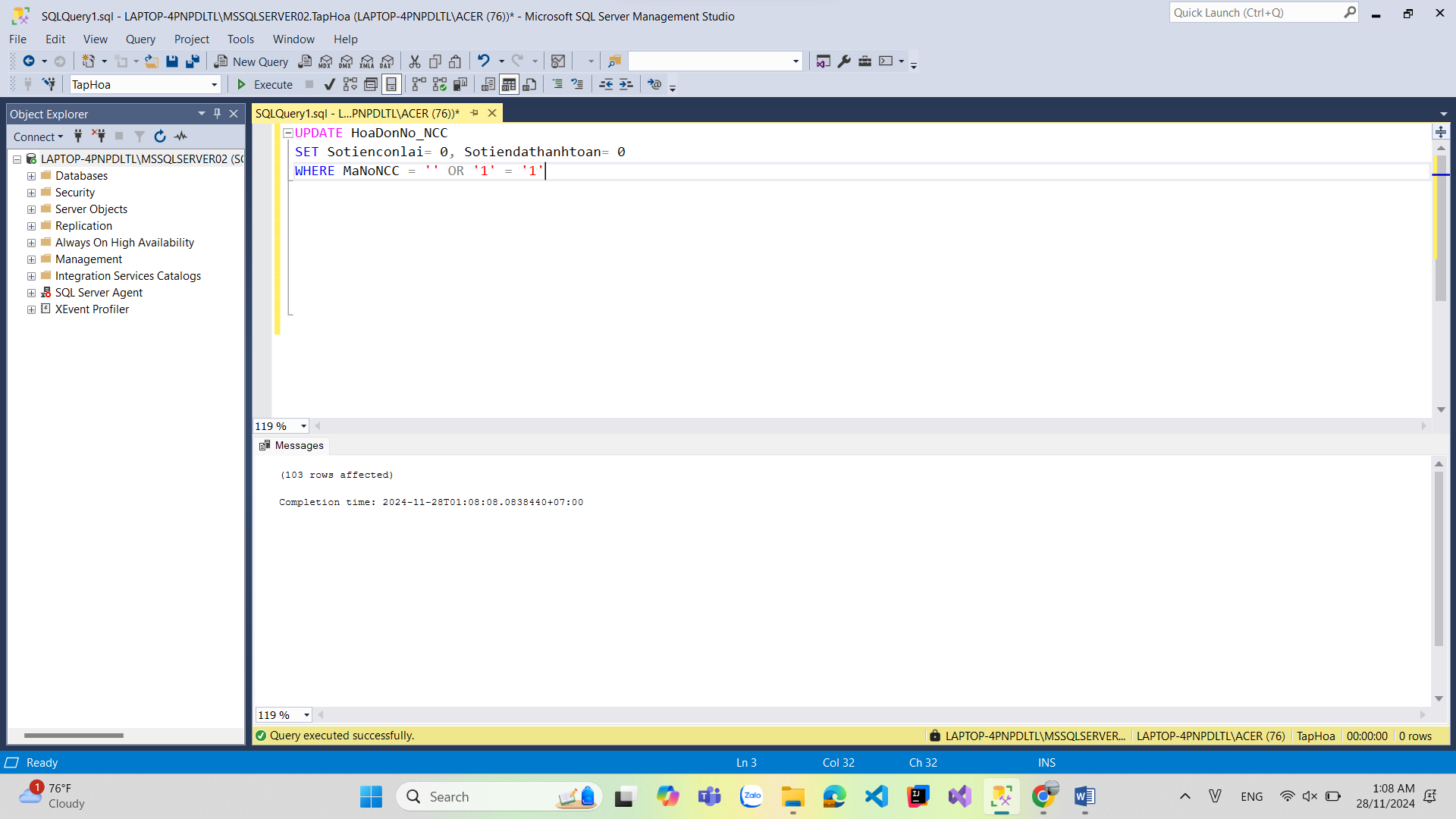
##### Cập nhật lại toàn bộ thông tin hoá đơn nợ nhà cung cấp trong bảng HoaDonNo\_NCC

Mục tiêu: Kẻ tấn công muốn cập nhật toàn bộ thông tin về số tiền còn lại và số tiền đã thanh toán của hoá đơn nợ nhà cung cấp về 0 mà không cần có quyền cập nhật ở hệ thống này.

UPDATE HoaDonNo\_NCC

SET Sotienconlai= 0, Sotiendathanhtoan= 0

WHERE MaNoNCC = '' OR '1' = '1'



Giải pháp

CREATE PROCEDURE Update\_HoaDonNo\_NCC

@MaNoNCC CHAR (15),

@Sotienconlai numeric (15,2),

@Sotiendathanhtoan numeric (15,2)

AS

BEGIN

IF LEN(@MaNoNCC) <> 15 OR PATINDEX('%[^a-zA-Z0-9]%', @MaNoNCC) > 0

BEGIN

PRINT N' Đầu vào không hợp lệ'

RETURN

END

IF NOT EXISTS (SELECT 1 FROM HoaDonNo\_NCC WHERE @MaNoNCC = MaNoNCC)

BEGIN

PRINT N'Mã không tồn tại.'

RETURN

END

UPDATE HoaDonNo\_NCC

SET Sotienconlai = CONVERT(VARBINARY(MAX), CAST (@Sotienconlai AS NVARCHAR(MAX))), Sotiendathanhtoan = CONVERT(VARBINARY(MAX), CAST (@Sotiendathanhtoan AS NVARCHAR(MAX)))

WHERE @MaNoNCC = MaNoNCC

PRINT N'Cập nhật thông tin khách hàng thành công.'

END

---

EXEC Update\_HoaDonNo\_NCC

@MaNoNCC = ''' OR ''1'' = ''1''',

@Sotienconlai = 0,

@Sotiendathanhtoan = 0

##### Cập nhật lại toàn bộ thông tin hoá đơn khách hàng nợ trong bảng HoaDon\_KHNo

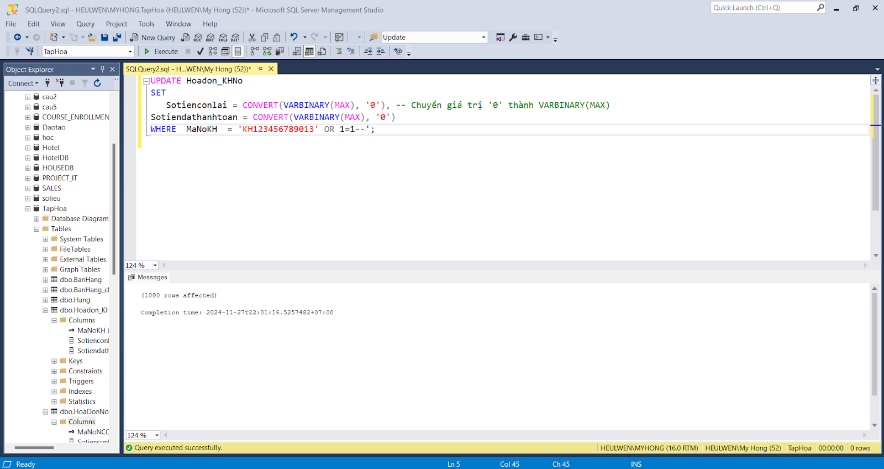
Mục tiêu: Kẻ tấn công muốn cập nhật số tiền còn lại và số tiền đã thanh toán của hoá đơn khách hàng nợ về 0 mà không cần có quyền cập nhật ở hệ thống này.

UPDATE Hoadon\_KHNo

SET Sotienconlai = CONVERT(VARBINARY(MAX), '0'),

Sotiendathanhtoan = CONVERT(VARBINARY(MAX), '0')

WHERE MaNoKH = 'KH123456789013' OR 1=1--';



Giải pháp:

CREATE PROCEDURE UpdateHoadon\_KHNo

@MaNoKH CHAR (10),

@Sotienconlai NUMERIC,

@Sotiendathanhtoan NUMERIC

AS

BEGIN

-- Kiểm tra điều kiện đầu vào cho số tiền (chỉ chấp nhận giá trị hợp lệ)

IF @Sotienconlai < 0 OR @Sotiendathanhtoan < 0

BEGIN

PRINT N'Số tiền không hợp lệ.';

RETURN;

END;

-- Kiểm tra ký tự đặc biệt trong MaNoKH

IF PATINDEX('%[^a-zA-Z0-9]%', @MaNoKH) > 0

BEGIN

PRINT N'Mã nợ khách hàng không hợp lệ.'

RETURN

END

-- Kiểm tra sự tồn tại của MaNoKH trong bảng Hoadon\_KHNo

IF NOT EXISTS (SELECT 1 FROM Hoadon\_KHNo WHERE MaNoKH = @MaNoKH)

BEGIN

PRINT N'Mã nợ khách hàng không tồn tại.'

RETURN

END

UPDATE Hoadon\_KHNo

SET Sotienconlai = CONVERT(VARBINARY(MAX), CAST (@Sotienconlai AS NVARCHAR(MAX))), Sotiendathanhtoan = CONVERT(VARBINARY(MAX), CAST (@Sotiendathanhtoan AS NVARCHAR(MAX)))

WHERE MaNoKH = @MaNoKH;

PRINT N'Cập nhật thông tin hóa đơn nợ khách hàng thành công.'

END

EXEC UpdateHoadon\_KHNo

@MaNoKH = ''' OR ''1'' = ''1'''

@Sotienconlai = 0,

@Sotiendathanhtoan = 0;

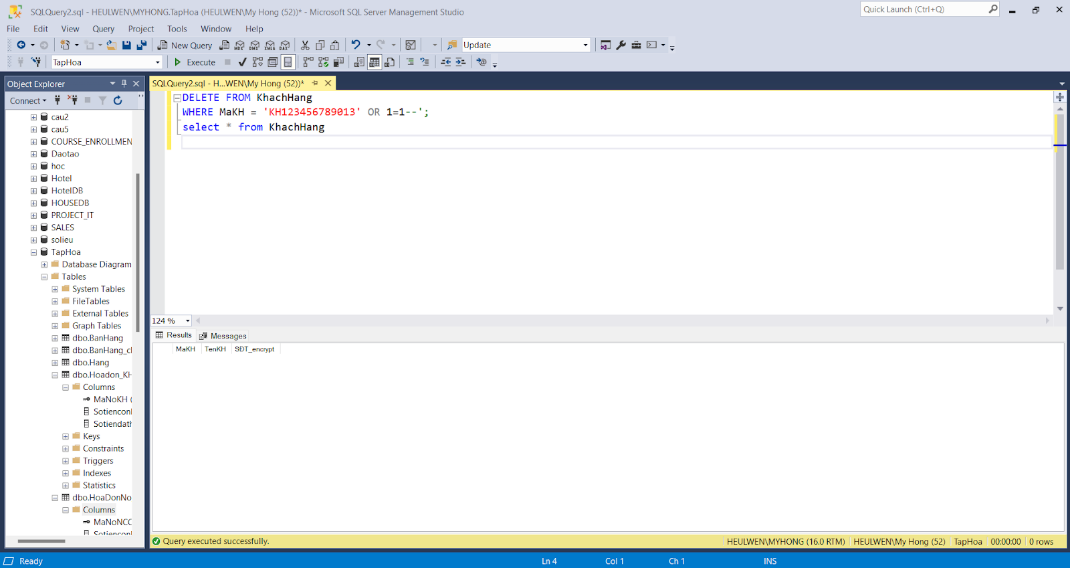
### Tấn công vào chức năng Delete (Xoá)

##### Xoá toàn bộ thông tin thông tin khách hàng trong bảng KhachHang

Mục tiêu: Kẻ tấn công muốn xoá toàn bộ thông tin khách hàng mà không cần có quyền xoá ở hệ thống này.

DELETE FROM KhachHang

WHERE MaKH = 'KH123456789013' OR 1=1--'



Giải pháp:

CREATE PROCEDURE DeleteKhachHang @MaKH CHAR (15)

AS

BEGIN

-- Kiểm tra ký tự đặc biệt trong MaKH

IF PATINDEX('%[^a-zA-Z0-9]%', @MaKH) > 0

BEGIN

PRINT N'Mã khách hàng chứa ký tự không hợp lệ!'

RETURN

END

-- Kiểm tra sự tồn tại của MaKH trong bảng KhachHang

IF NOT EXISTS (SELECT 1 FROM KhachHang WHERE MaKH = @MaKH)

BEGIN

PRINT N'Mã khách hàng không tồn tại.'

RETURN;

END

DELETE FROM KhachHang

WHERE MaKH = @MaKH

PRINT N'Xóa khách hàng thành công!'

END

---

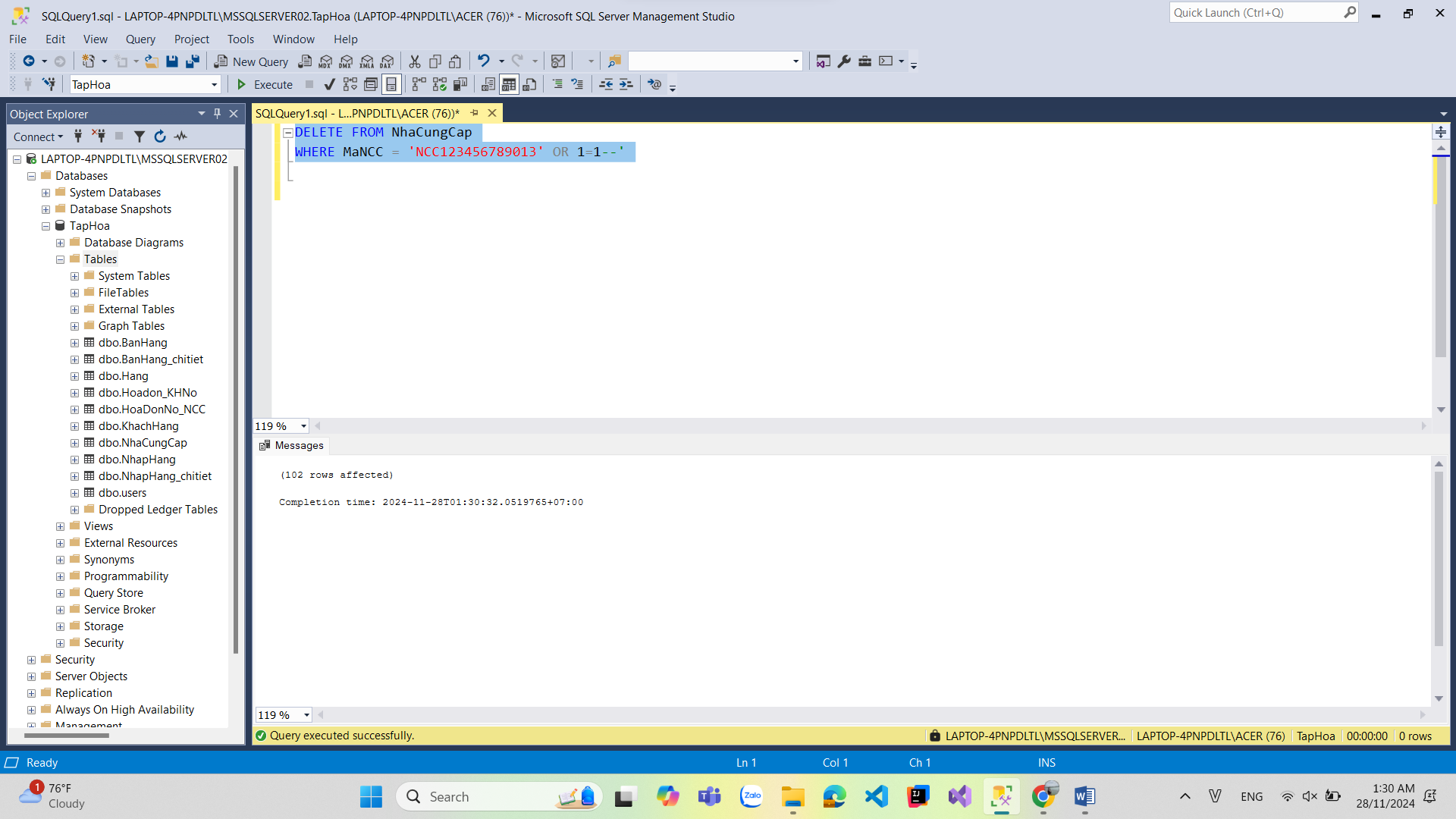
exec DeleteKhachHang 'OR 1=1'

##### Xoá toàn bộ thông tin nhà cung cấp trong bảng NhaCungCap

Mục tiêu: Kẻ tấn công muốn xoá toàn bộ thông tin nhà cung cấp mà không cần có quyền xoá ở hệ thống này.

DELETE FROM NhaCungCap

WHERE MaNCC = 'NCC123456789013' OR 1=1--'



Giải pháp

CREATE PROCEDURE DeleteNhaCungCap

@MaNCC CHAR (15)

AS

BEGIN

-- Kiểm tra ký tự đặc biệt trong MaNCC

IF PATINDEX('%[^a-zA-Z0-9]%', @MaNCC) > 0

BEGIN

PRINT N' Mã khách hàng chứa ký tự không hợp lệ.'

RETURN

END

-- Kiểm tra sự tồn tại của MaNCC trong bảng NhaCungCap

IF NOT EXISTS (SELECT 1 FROM NhaCungCap WHERE MaNCC = @MaNCC)

BEGIN

PRINT N' Mã nhà cung cấp không tồn tại.'

RETURN

END

DELETE FROM NhaCungCap

WHERE MaNCC = @MaNCC

PRINT N' Xóa nhà cung cấp thành công.'

END

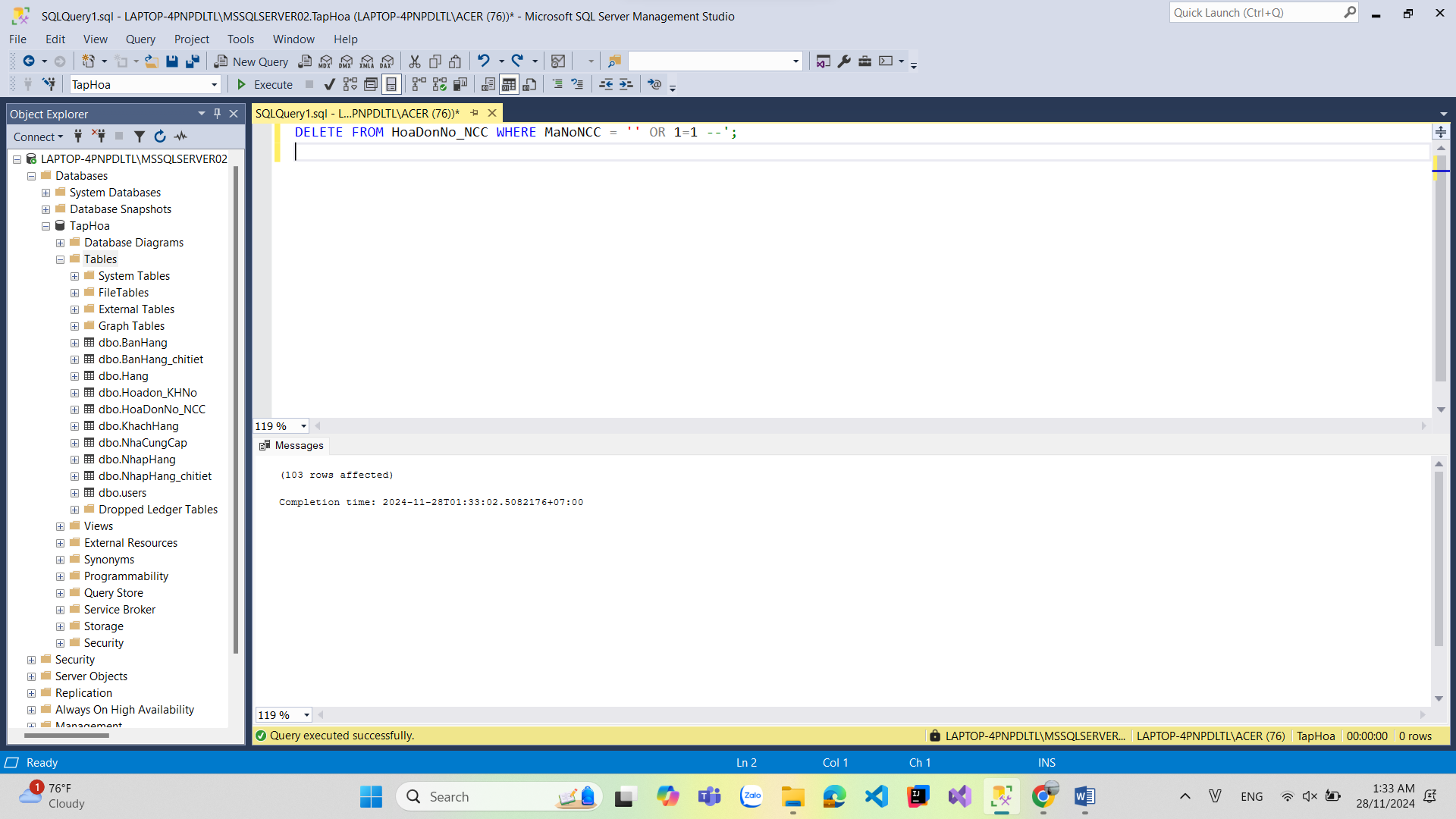
exec DeleteNhaCungCap 'OR 1=1'

##### Xoá toàn bộ thông tin hoá đơn nợ nhà cung cấp trong bảng HoaDonNo\_NCC

Mục tiêu: Kẻ tấn công muốn xoá toàn bộ thông tin hoá đơn mà không cần có quyền xoá ở hệ thống này.

DELETE FROM HoaDonNo\_NCC

WHERE MaNoNCC = '' OR 1=1 --';



Giải pháp

CREATE PROCEDURE Delete\_HoaDonNo\_NCC

@MaNoNCC CHAR (15)

AS

BEGIN

IF PATINDEX('%[^a-zA-Z0-9]%', @MaNoNCC) > 0

BEGIN

PRINT N'Mã nợ nhà cung cấp không hợp lệ.'

RETURN

END

DELETE FROM HoaDonNo\_NCC

WHERE MaNoNCC = @MaNoNCC

IF @@ROWCOUNT = 0

BEGIN

PRINT N'Mã nợ nhà cung cấp không tồn tại.'

END

ELSE

BEGIN

PRINT N'Xóa dữ liệu thành công.'

END

END

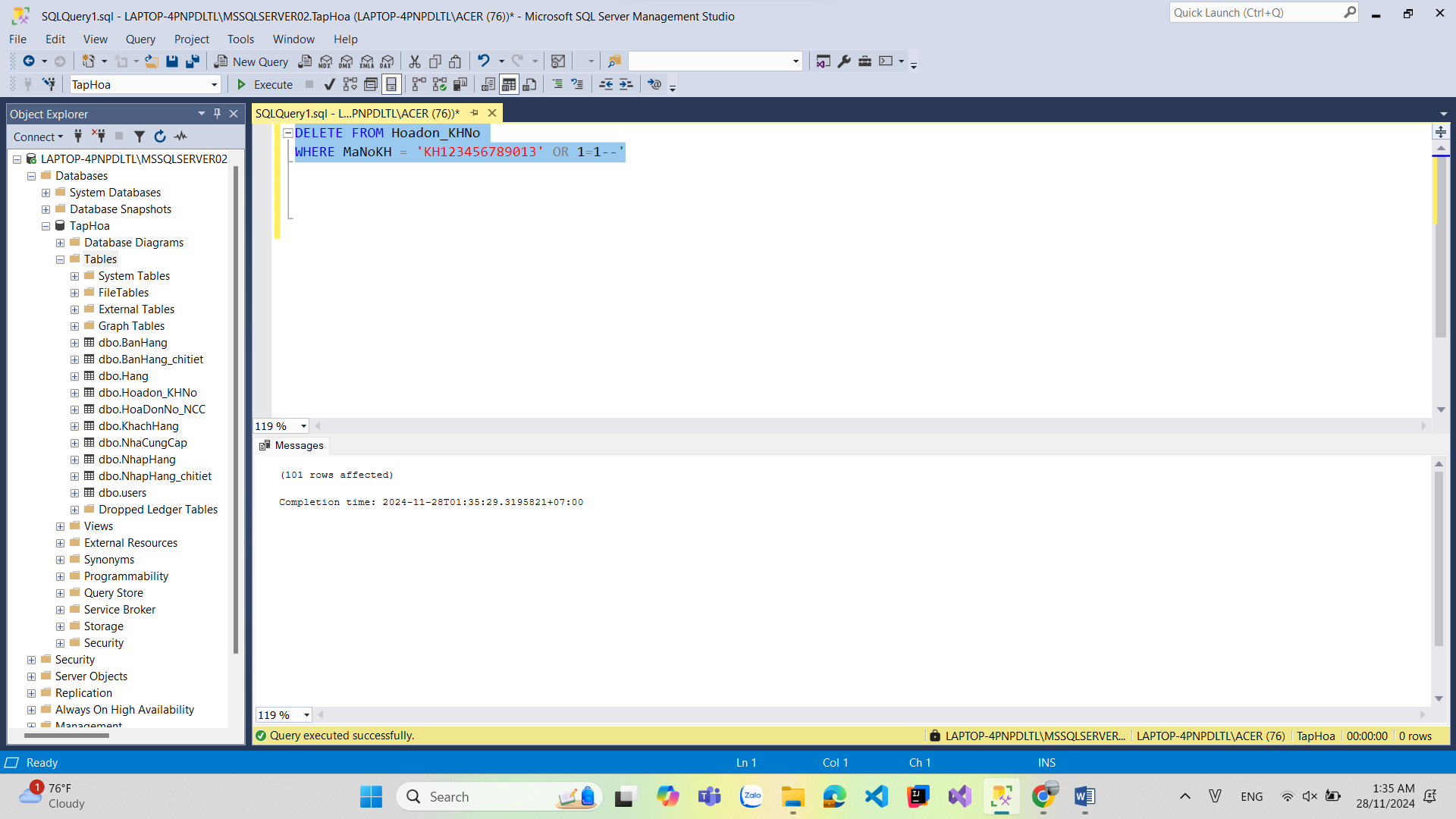
exec DeleteHoaDonNo\_NCC 'OR 1=1'

##### Xoá toàn bộ thông tin hoá đơn khách hàng nợ trong bảng HoaDon\_KHNo

Mục tiêu: Kẻ tấn công muốn xoá toàn bộ thông tin hoá đơn mà không cần có quyền xoá ở hệ thống này.

DELETE FROM Hoadon\_KHNo

WHERE MaNoKH = 'KH123456789013' OR 1=1--'



Giải pháp

CREATE PROCEDURE DeleteHoadon\_KHNo @MaNoKH CHAR (10)

AS

BEGIN

-- Kiểm tra ký tự đặc biệt trong MaNoKH

IF PATINDEX('%[^a-zA-Z0-9]%', @MaNoKH) > 0

BEGIN

PRINT N'Mã nợ khách hàng chứa ký tự không hợp lệ.'

RETURN

END

-- Kiểm tra sự tồn tại của MaNoKH trong bảng HoaDon\_KHNo

IF NOT EXISTS (SELECT 1 FROM HoaDon\_KHNo WHERE MaNoKH = @MaNoKH)

BEGIN

PRINT N'Mã nợ khách hàng không tồn tại.'

RETURN

END

DELETE FROM HoaDon\_KHNo

WHERE MaNoKH = @MaNoKH

PRINT N'Xóa hóa đơn nợ khách hàng thành công.'

END

EXEC DeleteHoadon\_KHNo @MaNoKH = 'OR 1=1'

# Ghi chú: Thay đổi về thiết kế

## Thay đổi R3

### Thay đổi các kiểu dữ liệu ở thiết kế chi tiết các bảng trong R2

##### Thay đổi kiểu dữ liệu trong bảng NhaCungCap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNCC | CHAR(20) | Khóa chính | Chuyển CHAR(20) thành VARCHAR(20) |
| 2 | TenNCC | CHAR(50) | NOT NULL | Chuyển CHAR(50) thành VARCHAR(50) |
| 3 | DiaChi | NVARCHAR(100) | NOT NULL |  |
| 4 | SĐT | CHAR(10) | NOT NULL,  UNIQUE |  |
| 5 | MaNV | CHAR (50) | NULL | Chuyển CHAR(50) thành VARCHAR (30) |
| 6 | TenNV | CHAR (50) | NULL | Chuyển CHAR(50) thành VARCHAR (50) |

##### Thay đổi kiểu dữ liệu trong bảng NhapHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNhapHang | CHAR(20) | Khóa chính | Chuyển CHAR(20) thành VARCHAR(20) |
| 2 | MaNCC | CHAR(20) | Khóa ngoại | Chuyển CHAR(20) thành VARCHAR(20) |
| 3 | MaNoNCC | CHAR(20) | Khóa ngoại | Chuyển CHAR (20) thành CHAR(10) |
| 4 | KhuyenMai | NUMERIC | NULL |  |
| 5 | TongTien | NUMERIC | NOT NULL |  |
| 6 | NgayGiao | DATE | NOT NULL |  |

##### Thêm một thuộc tính SĐTKH vào bảng KhachHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaKH | CHAR(20) | Khóa chính |  |
| 2 | TenKH | NVARCHAR(50) | NULL |  |
| 3 | SĐTKH | CHAR(10) | NOT NULL,  UNIQUE | Thêm 1 thuộc tính |

##### Thêm khóa ngoại, khóa chính NhapHang\_ChiTiet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaNhapHang | CHAR(20) | Khóa chính | Thêm khóa ngoại |
| 2 | MaHang | CHAR(20) | Khóa ngoại | Thêm khóa chính |
| 3 | SoLuong | INT | NOT NULL, >0 |  |
| 4 | ThanhTien | NUMERIC | NOT NULL |  |

##### Thêm khóa ngoại, khóa chính bảng BanHang\_ChiTiet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | MaBanHang | CHAR(20) | Khóa chính, khóa ngoại | Thêm khóa ngoại |
| 2 | MaHang | CHAR(20) | Khóa ngoại, khóa chính | Thêm khóa chính |
| 3 | SoLuong | INT | NOT NULL, >0 |  |
| 4 | ThanhTien | NUMERIC | NOT NULL |  |

## Thay đổi R5

### Thêm bảng Users vào thiết kế chi tiết các bảng ở R2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | Admin | VARCHAR(50) |  |  |
| 2 | Passwords | VARCHAR(50) |  |  |