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

**KHOA THỐNG KÊ – TIN HỌC**



**BẤO CÁO BÀI TẬP NHÓM**

***Bộ môn : Quản trị hệ thống cơ sở dữ liệu***

Nhóm 48K211.06

**Sinh viên thực hiện**: Đoàn Ngọc Ban Nhiên

Vương Thị Mỹ Lệ

Bùi Khắc Thanh Tú

Phạm Hoàng Linh Chi

Nguyễn Đức Thịnh

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

**Đà Nẵng, 2024**

**MỤC LỤC**

[1. Chuẩn hoá theo dạng chuẩn 2NF trở lên (R2) 1](#_Toc183801134)

[1.1. Chuyển thực thể, thu được: 1](#_Toc183801135)

[1.2. Chuyển quan hệ, thu được: 1](#_Toc183801136)

[1.3. Chuẩn hoá quan hệ: 1](#_Toc183801137)

[1.3.1. Chuẩn hoá quan hệ donHangBan, thu được: 1](#_Toc183801138)

[1.3.2. Chuẩn hoá quan hệ donHangNhap, thu được: 1](#_Toc183801139)

[1.4. Vẽ sơ đồ quan hệ: 2](#_Toc183801140)

[2. Thiết kế chi tiết: 2](#_Toc183801141)

[2.1. Chuyển thực thể, thu được: 2](#_Toc183801142)

[3. Tạo bảng cơ sở dữ liệu (R3) 12](#_Toc183801143)

[3.1.1. Tạo cơ sở dữ liệu 12](#_Toc183801144)

[3.1.2. Tạo bảng taiKhoan 12](#_Toc183801145)

[3.1.3. Tạo bảng nguoiDung 12](#_Toc183801146)

[3.1.4. Tạo bảng khachHang 12](#_Toc183801147)

[3.1.5. Tạo bảng nhaCungCap 12](#_Toc183801148)

[3.1.6. Tạo bảng sanPham 13](#_Toc183801149)

[3.1.7. Tạo bảng donHangBan 13](#_Toc183801150)

[3.1.8. Tạo bảng donHangBanCT 13](#_Toc183801151)

[3.1.9. Tạo bảng donHangNhap 14](#_Toc183801152)

[3.1.10. Tạo bảng donHangNhapCT 14](#_Toc183801153)

[3.1.11. Tạo bảng noPhaiTraKH 14](#_Toc183801154)

[3.1.12. Tạo bảng noPhaiTraNCC 15](#_Toc183801155)

[4. Thiết kế module 15](#_Toc183801156)

[4.1. Module dữ liệu 15](#_Toc183801157)

[4.1.1. Tạo Dữ Liệu cho Bảng taiKhoan 15](#_Toc183801158)

[4.1.2. Tạo Dữ Liệu cho Bảng nguoiDung 16](#_Toc183801159)

[4.1.3. Tạo Dữ Liệu cho Bảng sanPham 16](#_Toc183801160)

[4.1.4. Tạo Dữ Liệu cho Bảng khachHang 17](#_Toc183801161)

[4.1.5. Tạo Dữ Liệu cho Bảng nhaCungCap 18](#_Toc183801162)

[4.1.6. Tạo Dữ Liệu cho Bảng donHangBan 18](#_Toc183801163)

[4.1.7. Tạo Dữ Liệu cho Bảng donHangBanCT 19](#_Toc183801164)

[4.1.8. Tạo Dữ Liệu cho Bảng donHangNhap 21](#_Toc183801165)

[4.1.9. Tạo Dữ Liệu cho Bảng donHangNhapCT 22](#_Toc183801166)

[4.1.10. Tạo Dữ Liệu cho Bảng noPhaiTraKH 24](#_Toc183801167)

[4.1.11. Tạo Dữ Liệu cho Bảng noPhaiTraNCC 24](#_Toc183801168)

[4.2. Module nghiệp vụ 25](#_Toc183801169)

[4.2.1. Thủ tục 25](#_Toc183801170)

[4.2.2. Trigger 35](#_Toc183801171)

[5. Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu 42](#_Toc183801172)

[5.1. Tạo và phân quyền phù hợp cho người dùng (ở cấp độ server và cấp độ database) 42](#_Toc183801173)

[5.2. Mã hoá dữ liệu cần bảo mật 47](#_Toc183801174)

[6. Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu 48](#_Toc183801175)

[7. Giả sử lượng dữ liệu rất lớn, không đủ lưu trữ trong một ổ đĩa. Hãy đưa ra phương án giải quyết TỐT NHẤT cho vấn đề đã đặt ra và thực hiện trên CSDL đã xây dựng. (Nêu lý do lựa chọn giải pháp và thực hiện triển khai giải pháp cho hệ thống) 63](#_Toc183801176)

[8. Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection 67](#_Toc183801177)

**DANH MỤC HÌNH ẢNH**

[Hình 1‑1. Sơ đồ quan hệ 2](#_Toc183800899)

[Hình 5‑1. Đăng nhập vào hệ thống SQL server bằng Azure server 43](#_Toc183800900)

[Hình 5‑2. Tạo tài khoản login 43](#_Toc183800901)

[Hình 5‑3. Tạo user trong cơ sở dữ liệu 44](#_Toc183800902)

[Hình 5‑4. Phân quyền cho user vùa tạo 44](#_Toc183800903)

[Hình 5‑5. Đăng nhập bằng user đã được phân quyền 45](#_Toc183800904)

[Hình 5‑6. Kiểm tra phân quyền 46](#_Toc183800905)

[Hình 5‑7. Mã hoá dữ liệu số điện thoại khách hàng 47](#_Toc183800906)

[Hình 5‑8. Mã hoá dữ liệu số điện thoại của nhà cung cấp 48](#_Toc183800907)

[Hình 6‑1. Đăng nhập vào Microsoft Azure và chọn server nhom48k21106 qlbh 48](#_Toc183800908)

[Hình 6‑2. Chọn Data **management →** chọn **Backups** 49](#_Toc183800909)

[Hình 6‑3. Nhấn vào **Retention policies** và nhấn tích vào **Nhom48K21106** 49](#_Toc183800910)

[Hình 6‑4. Chọn **Configure policies**  để điều chỉnh thời gian backups 50](#_Toc183800911)

[Hình 6‑5. Hoàn thành Back Up trên Microsoft Azure 50](#_Toc183800912)

[Hình 6‑6. Truy cập vào ứng dụng **Microsoft SQL Server** 51](#_Toc183800913)

[Hình 6‑7. Chọn **Export Data-tier Application** 51](#_Toc183800914)

[Hình 6‑8. Chọn vị trí cần lưu 52](#_Toc183800915)

[Hình 6‑9. Chọn Finish 53](#_Toc183800916)

[Hình 6‑10. Hoàn thành Back up 54](#_Toc183800917)

[Hình 6‑11. Kiểm tra thư mục đã có file bacpac hay chưa 55](#_Toc183800918)

[Hình 6‑12. Restore using Bacpac file in local 55](#_Toc183800919)

[Hình 6‑13 Nhấn Next 56](#_Toc183800920)

[Hình 6‑14. Chọn file backpack đã lưu ở bước 10 56](#_Toc183800921)

[Hình 6‑15. Chọn Next 57](#_Toc183800922)

[Hình 6‑16. Chọn Next tiếp 58](#_Toc183800923)

[Hình 6‑17. Chọn Finish 59](#_Toc183800924)

[Hình 6‑18. Khôi phục (restore) cơ sở dữ liệu trên local hoàn thành 60](#_Toc183800925)

[Hình 6‑19. Danh sách các bảng trên máy chủ CSDL Azure 61](#_Toc183800926)

[Hình 6‑20. Danh sách các bảng trên SQL Local 62](#_Toc183800927)

[Hình 7‑1. Tạo server và các database trong Microsoft Azure 64](#_Toc183800928)

[Hình 7‑2. Connect trong SQL server với server name: nhom48k21106qlbh.database.windows.net. 64](#_Toc183800929)

[Hình 7‑3. Database vừa tạo trên Microsoft Azure đã hiển thị trong SQL Server 65](#_Toc183800930)

[Hình 7‑4. Tạo các bảng và các module trong Database “Nhom48K21106” 65](#_Toc183800931)

[Hình 7‑5. Hình ảnh danh sách các bảng hiện ra ở cột Object Explorer trong SSMS 66](#_Toc183800932)

[Hình 7‑6. Kết quả sau khi chạy các database trên SSMS sẽ hiển thị trên MS Azure 66](#_Toc183800933)

[Hình 8‑1. Triển khai phương án mã hoá dữ liệu đầu vào 67](#_Toc183800934)

**DANH MỤC BẢNG BIỂU**

[Bảng 2‑1. Bảng tài khoản 2](#_Toc183801620)

[Bảng 2‑2. Bảng người dùng 3](#_Toc183801621)

[Bảng 2‑3. Bảng khách hàng 4](#_Toc183801622)

[Bảng 2‑4. Bảng tài khoản 4](#_Toc183801623)

[Bảng 2‑5. Bảng sản phảm 5](#_Toc183801624)

[Bảng 2‑6. Bảng đơn hàng bán 6](#_Toc183801625)

[Bảng 2‑7. Bảng đơn hàng bán chi tiết 7](#_Toc183801626)

[Bảng 2‑8. Bảng nợ phải trả khách hàng 8](#_Toc183801627)

[Bảng 2‑9. Bảng đơn hàng nhập 9](#_Toc183801628)

[Bảng 2‑10. Bảng đơn hàng nhập chi tiết 10](#_Toc183801629)

[Bảng 2‑11. Bảng nợ phải trả nhà cung cấp 11](#_Toc183801630)

[Bảng 4‑1. Thủ tục kiểm tra sản phẩm trước khi insert 26](#_Toc183801631)

[Bảng 4‑2. Thủ tục kiểm tra đơn hàng bán trước khi insert 27](#_Toc183801632)

[Bảng 4‑3. Thủ tục kiểm tra đơn hàng bán chi tiết trước khi insert 29](#_Toc183801633)

[Bảng 4‑4. Thủ tục kiểm tra nợ khách hàng trước khi insert 30](#_Toc183801634)

[Bảng 4‑5. Thủ tục kiểm tra đơn hàng nhập trước khi insert 31](#_Toc183801635)

[Bảng 4‑6. Thủ tục kiểm tra đơn hàng nhập chi tiết trước khi insert 32](#_Toc183801636)

[Bảng 4‑7. Thủ tục kiểm tra nợ nhà cung cấp trước khi insert 34](#_Toc183801637)

[Bảng 4‑8. Thủ tục kiểm tra khách hàng trước khi insert 34](#_Toc183801638)

[Bảng 4‑9. Trigger kiểm tra giá bán 35](#_Toc183801639)

[Bảng 4‑10. Tính tổng tiền đơn hàng bán 37](#_Toc183801640)

[Bảng 4‑11. Trigger cập nhật số lượng tồn kho khi bán 37](#_Toc183801641)

[Bảng 4‑12. Kiểm tra ngày trả tiền của khách hàng 39](#_Toc183801642)

[Bảng 4‑13. Tính tổng tiền đơn hàng nhập 40](#_Toc183801643)

[Bảng 4‑14. Trigger cập nhật số lượng tồn kho khi nhập 41](#_Toc183801644)

[Bảng 4‑15. Trigger kiểm tra ngày trả tiền cho nhà cung cấp 42](#_Toc183801645)

# Chuẩn hoá theo dạng chuẩn 2NF trở lên (R2)

## Chuyển thực thể, thu được:

* taiKhoan (**tenDangNhap**, matKhau)
* nguoiDung (**SDT**, tenDangNhap, hoTen, email, diaChi)
* khachHang (**maKH**, hoTen, SDT)
* nhaCungCap(**maNCC**, tenNCC, SDT)
* sanPham (**maSP**, tenSP, soLuongSP, giaBan, giaNhap, hinhAnhSP, HSD)

## Chuyển quan hệ, thu được:

* donHangBan (**maDHB**, maKH, maSP,giaBan, soLuong, thanhTien, ngayTao, tongTien, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)
* donHangNhap (**maDHN**, maNCC, maSP, soLuong, giaNhap, thanhTien, ngayTao, tongTien, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

## Chuẩn hoá quan hệ:

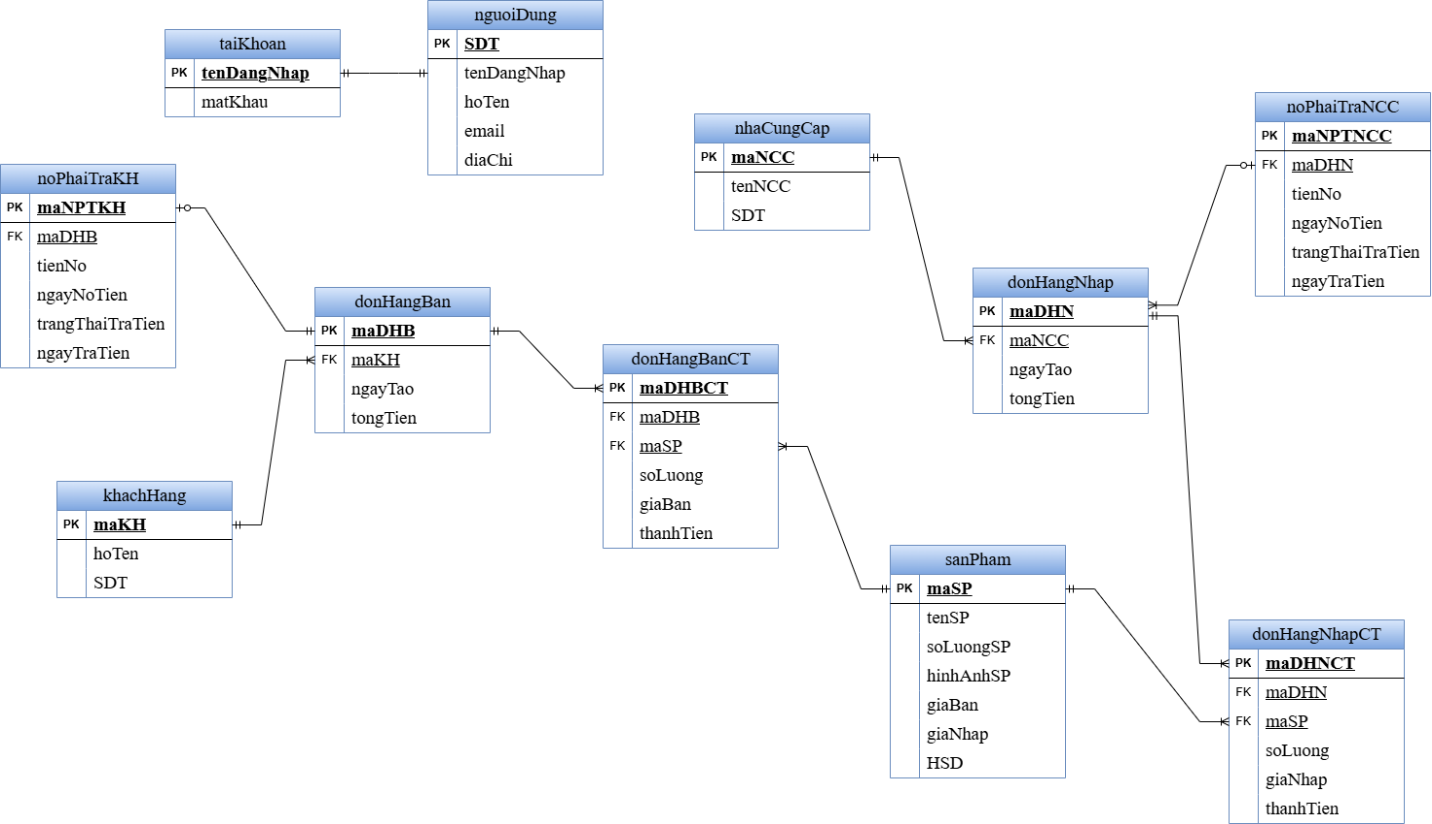
### Chuẩn hoá quan hệ donHangBan, thu được:

* donHangBan (**maDHB**, maKH, ngayTao, tongTien)
* donHangBanCT (**maDHBCT**, maDHB, maSP, soLuong, giaBan, thanhTien)
* noPhaiTraKH (**maNPTKH**, maKH, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

### Chuẩn hoá quan hệ donHangNhap, thu được:

* donHangNhap (**maDHN**, maNCC, ngayTao, tongTien)
* donHangNhapCT (**maDHNCT**, maDHN, maSP, soLuong, giaNhap, thanhTien)
* noPhaiTraNCC (**maNPTNCC**, maNCC, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

## Vẽ sơ đồ quan hệ:



Hình 1‑1. Sơ đồ quan hệ

# Thiết kế chi tiết:

## Chuyển thực thể, thu được:

* taiKhoan (**tenDangNhap**, matKhau)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | tenDangNhap | varchar (50) | Not null | PK | Khóa chính của bảng |
| 2 | matKhau | varchar (50) | Not null |  | Mật khẩu |

Bảng 2‑1. Bảng tài khoản

* nguoiDung (**SDT**, tenDangNhap, hoTen, email, diaChi)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | SDT | char (10) | Not null | PK | Khóa chính của bảng |
| 2 | tenDangNhap | varchar (50) | Not null |  | Tên đăng nhập |
| 3 | hoTen | varchar (50) | Not null |  | Họ tên người dùng |
| 4 | email | varchar(100) | Not null |  | Email |
| 5 | diaChi | varchar(100) | Not null |  | Địa chỉ |

Bảng 2‑2. Bảng người dùng

* khachHang (**maKH**, hoTen, SDT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maKH | int | Not null | PK | Khóa chính của bảng |
| 2 | hoTen | varchar (50) | Not null |  | Họ tên khách hàng |
| 3 | SDT | char (10) | Not null |  | Số điện thoại của khách hàng |

Bảng 2‑3. Bảng khách hàng

* nhaCungCap(**maNCC**, tenNCC, SDT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maNCC | int | Not null | PK | Khóa chính của bảng |
| 2 | tenNCC | varchar (50) | Not null |  | Tên nhà cung cấp |
| 3 | SDT | char (10) | Not null |  | Số điện thoại nhà cung cấp |

Bảng 2‑4. Bảng tài khoản

* sanPham (**maSP**, tenSP, soLuongSP, hinhAnhSP, giaNhap, giaBan, HSD)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maSP | int | Not null | PK | Khóa chính của bảng |
| 2 | tenSP | varchar (50) | Not null |  | Tên sản phẩm |
| 3 | soLuongSP | int | Not null |  | Số lượng sản phẩm |
| 4 | hinhAnhSP | image | Not null |  | Hình ảnh sản phẩm |
| 5 | giaNhap | Numeric (15, 0) | Not null |  | giaNhap |
| 6 | giaBan | Numeric (15, 0) | Not null |  | giaBan |
| 7 | HSD | date | Not null |  | Hạn sử dụng |

Bảng 2‑5. Bảng sản phảm

* donHangBan(**maDHB**, maKH, ngayTao, tongTien)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maDHB | int | not null | PK | Khóa chính của bảng |
| 2 | maKH | int | not null | FK | Khóa ngoại |
| 3 | ngayTao | date | not null |  | Ngày khách hàng mua hàng |
| 4 | tongTien | Numeric (15, 0) | not null |  | Tổng tiền hàng khách hàng phải trả |

Bảng 2‑6. Bảng đơn hàng bán

* donHangBanCT(**maDHBCT**, maDHB, maSP, soLuong, giaBan, thanhTien)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maDHBCT | int | not null | PK | Khóa chính của bảng |
| 2 | maDHB | int | not null | FK | Khóa ngoại |
| 3 | maSP | int | not null | FK | Khóa ngoại |
| 4 | soLuong | int | not null |  | Số lượng sản phẩm |
| 5 | giaBan | Numeric (15, 0) | not null |  | Giá bán |
| 6 | thanhTien | Numeric (15, 0) | not null |  | Tổng tiền mỗi mặt hàng |

Bảng 2‑7. Bảng đơn hàng bán chi tiết

* noPhaiTraKhachHang (**maNPTKH**, maDHB, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maNPTKH | int | not null | PK | Khóa chính của bảng |
| 2 | maDHB | int | not null | FK | Khóa ngoại |
| 3 | tienNo | Numeric (15, 0) | not null |  | Tiền nợ của khách hàng |
| 4 | ngayNoTien | date | not null |  | Ngày trả tiền của khách hàng |
| 5 | trangThaiTraTien | bit | not null |  | Trạng thái trả tiền |
| 6 | ngayTraTien | date | not null |  | Ngày trả tiền của khách hàng |

Bảng 2‑8. Bảng nợ phải trả khách hàng

* donHangNhap (**maDHN**, maNCC, ngayTao, tongTien)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maDHN | int | not null | PK | Khóa chính của bảng |
| 2 | maNCC | int | not null | FK | Khóa ngoại của bảng |
| 3 | ngayTao | date | notnull |  | Ngày tạo đơn hàng nhập |
| 4 | tongTien | Numeric (15, 0) | notnull |  | Tổng tiền |

Bảng 2‑9. Bảng đơn hàng nhập

* donHangNhapCT (**maDHNCT**, maDHN, maSP, soLuong, giaNhap, thanhTien)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maDHNCT | int | not null | PK | Khóa chính của bảng |
| 2 | maDHN | int | not null | FK | Khóa ngoại của bảng |
| 3 | maSP | int | not null | FK | Khóa ngoại của bảng |
| 4 | giaNhap | Numeric (15, 0) | not null |  | Giá nhập |
| 5 | soLuong | int | notnull |  | Số lượng sản phẩm |
| 6 | thanhTien | Numeric (15, 0) | notnull |  | Tiền đơn hàng nhập |

Bảng 2‑10. Bảng đơn hàng nhập chi tiết

* noPhaiTraNCC (**maNPTNCC**, maDNH, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Ghi chú** |
| 1 | maNPTNCC | int | not null | PK | Khóa chính của bảng |
| 2 | maDNH | int | not null | FK | Khóa ngoại của bảng |
| 3 | tienNo | Numeric (15, 0) | not null |  | Tiền nợ của nhà cung cấp |
| 4 | ngayNoTien | date | not null |  | Ngày nợ tiền của nhà cung cấp |
| 5 | trangThaiTraTien | bit | not null |  | Trạng thái trả tiền của nhà cung cấp |
| 6 | ngayTraTien | date | not null |  | Ngày trả tiền của nhà cung cấp |

Bảng 2‑11. Bảng nợ phải trả nhà cung cấp

# Tạo bảng cơ sở dữ liệu (R3)

### Tạo cơ sở dữ liệu

CREATE DATABASE Nhom48K21106;

### Tạo bảng taiKhoan

CREATE TABLE taiKhoan (

tenDangNhap VARCHAR(50) PRIMARY KEY,

matKhau VARCHAR(50)

);

### Tạo bảng nguoiDung

CREATE TABLE nguoiDung (

SDT CHAR(10) NOT NULL PRIMARY KEY,

tenDangNhap VARCHAR(50) NOT NULL,

hoTen VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

diaChi VARCHAR(100) NOT NULL,

FOREIGN KEY (tenDangNhap) REFERENCES taiKhoan(tenDangNhap)

);

### Tạo bảng khachHang

CREATE TABLE khachHang (

maKH INT NOT NULL PRIMARY KEY IDENTITY(1,1),

hoTen VARCHAR(50) NOT NULL,

SDT CHAR(10) NOT NULL

);

### Tạo bảng nhaCungCap

CREATE TABLE nhaCungCap (

maNCC INT NOT NULL PRIMARY KEY IDENTITY(1,1),

tenNCC VARCHAR(50) NOT NULL,

SDT CHAR(10) NOT NULL

);

### Tạo bảng sanPham

CREATE TABLE sanPham (

maSP INT NOT NULL PRIMARY KEY IDENTITY(1,1),

tenSP VARCHAR(50) NOT NULL,

soLuongSP INT NOT NULL,

hinhAnhSP IMAGE,

giaBan NUMERIC(15,0),

giaNhap NUMERIC(15,0),

HSD DATE NOT NULL

);

### Tạo bảng donHangBan

CREATE TABLE donHangBan (

maDHB INT NOT NULL PRIMARY KEY IDENTITY(1,1),

maKH INT NOT NULL,

ngayTao DATE NOT NULL,

tongTien NUMERIC(15,0) NOT NULL,

FOREIGN KEY (maKH) REFERENCES khachHang(maKH)

);

### Tạo bảng donHangBanCT

CREATE TABLE donHangBanCT (

maDHBCT INT NOT NULL PRIMARY KEY IDENTITY(1,1),

maDHB INT NOT NULL,

maSP INT NOT NULL,

giaBan NUMERIC(15,0),

soLuong INT NOT NULL,

thanhTien NUMERIC(15,0),

FOREIGN KEY (maDHB) REFERENCES donHangBan(maDHB),

FOREIGN KEY (maSP) REFERENCES sanPham(maSP)

);

### Tạo bảng donHangNhap

CREATE TABLE donHangNhap (

maDHN INT NOT NULL PRIMARY KEY IDENTITY(1,1),

maNCC INT NOT NULL,

ngayTao DATE NOT NULL,

tongTien NUMERIC(15,0),

FOREIGN KEY (maNCC) REFERENCES nhaCungCap(maNCC)

);

### Tạo bảng donHangNhapCT

CREATE TABLE donHangNhapCT (

maDHNCT INT NOT NULL PRIMARY KEY IDENTITY(1,1),

maDHN INT NOT NULL,

maSP INT NOT NULL,

giaNhap NUMERIC(15,0),

soLuong INT NOT NULL,

thanhTien NUMERIC(15,0),

FOREIGN KEY (maDHN) REFERENCES donHangNhap(maDHN),

FOREIGN KEY (maSP) REFERENCES sanPham(maSP)

);

### Tạo bảng noPhaiTraKH

CREATE TABLE noPhaiTraKH (

maNPTKH INT NOT NULL PRIMARY KEY IDENTITY(1,1),

maDHB INT NOT NULL,

tienNo NUMERIC(15,0) NOT NULL,

ngayNoTien DATE NOT NULL,

trangThaiTraTien BIT NOT NULL,

ngayTraTien DATE,

FOREIGN KEY (maDHB) REFERENCES donHangBan(maDHB)

);

### Tạo bảng noPhaiTraNCC

CREATE TABLE noPhaiTraNCC (

maNPTNCC INT NOT NULL PRIMARY KEY IDENTITY(1,1),

maDHN INT NOT NULL,

tienNo NUMERIC(15,0) NOT NULL,

ngayNoTien DATE NOT NULL,

trangThaiTraTien BIT NOT NULL,

ngayTraTien DATE,

FOREIGN KEY (maDHN) REFERENCES donHangNhap(maDHN)

);

# Thiết kế module

## Module dữ liệu

### Tạo Dữ Liệu cho Bảng taiKhoan

GO

DECLARE @i INT = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO taiKhoan (tenDangNhap, matKhau)

VALUES (

'user' + CAST(@i AS VARCHAR(50)), -- Tên đăng nhập từ user1 đến user100

'password' + CAST(@i AS VARCHAR(50)) -- Mật khẩu từ password1 đến password100

);

SET @i = @i + 1;

END;

### Tạo Dữ Liệu cho Bảng nguoiDung

GO

DECLARE @i INT = 1;

SET @i = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO nguoiDung (SDT, tenDangNhap, hoTen, email, diaChi)

VALUES (

RIGHT('090' + CAST(@i AS VARCHAR(10)), 10), -- Số điện thoại từ 0900000001 đến 0900000100

'user' + CAST(@i AS VARCHAR(50)), -- Liên kết tên đăng nhập với bảng taiKhoan

'Nguoi Dung ' + CAST(@i AS VARCHAR(50)), -- Họ tên

'nguoidung' + CAST(@i AS VARCHAR(50)) + '@email.com', -- Email

'Dia Chi ' + CAST(@i AS VARCHAR(100)) -- Địa chỉ

);

SET @i = @i + 1;

END;

### Tạo Dữ Liệu cho Bảng sanPham

GO

DECLARE @i INT = 1;

SET @i = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO sanPham (tenSP, soLuongSP, hinhAnhSP, giaNhap, giaBan, HSD)

VALUES (

'San Pham ' + CAST(@i AS VARCHAR(50)), -- Tên sản phẩm

ROUND(RAND() \* 100, 0), -- Số lượng ngẫu nhiên từ 0 đến 100

Null, -- Hình ảnh sản phẩm (dữ liệu rỗng)

ROUND(RAND() \* 1000000, 0), -- Giá nhập sản phẩm

ROUND(RAND() \* 1000000, 0), -- Giá bán sản phẩm

DATEADD(DAY, @i, GETDATE()) -- Hạn sử dụng

);

SET @i = @i + 1;

END;

SELECT \* FROM sanPham

### Tạo Dữ Liệu cho Bảng khachHang

GO

DECLARE @i INT = 1;

SET @i = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO khachHang (hoTen, SDT)

VALUES (

'Khach Hang ' + CAST(@i AS VARCHAR(50)), -- Họ tên khách hàng

'012345' + RIGHT('0000' + CAST(@i AS VARCHAR(4)), 4) -- Số điện thoại

);

SET @i = @i + 1;

END;

SELECT \* FROM khachHang

### Tạo Dữ Liệu cho Bảng nhaCungCap

GO

DECLARE @i INT = 1;

SET @i = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO nhaCungCap (tenNCC, SDT)

VALUES (

'Nha Cung Cap ' + CAST(@i AS VARCHAR(50)), -- Tên nhà cung cấp

'098765' + RIGHT('0000' + CAST(@i AS VARCHAR(4)), 4) -- Số điện thoại nhà cung cấp

);

SET @i = @i + 1;

END;

SELECT \* FROM nhaCungCap

### Tạo Dữ Liệu cho Bảng donHangBan

GO

DECLARE @i INT = 1;

DECLARE @tongTien NUMERIC(15, 0);

WHILE @i <= 1000

BEGIN

-- Tạo tổng tiền bằng tổng các thanhTien từ bảng donHangBanCT

SET @tongTien = (

SELECT SUM(thanhTien)

FROM donHangBanCT

WHERE maDHB = @i

);

-- Nếu chưa có dòng nào trong bảng chi tiết, mặc định tổng tiền = 0

IF @tongTien IS NULL

SET @tongTien = 0;

INSERT INTO donHangBan (maKH, ngayTao, tongTien)

VALUES (

ROUND(RAND() \* 1000, 0) + 1, -- Mã khách hàng ngẫu nhiên

DATEADD(DAY, @i, GETDATE()), -- Ngày tạo ngẫu nhiên tăng dần

@tongTien -- Tổng tiền từ bảng chi tiết

);

SET @i = @i + 1;

END;

SELECT \* FROM donHangBan;

### Tạo Dữ Liệu cho Bảng donHangBanCT

GO

DECLARE @i INT = 1;

DECLARE @maSP INT, @giaBan NUMERIC(15,0), @soLuong INT, @thanhTien NUMERIC(15,0);

SET @i = 1;

WHILE @i <= 1000

BEGIN

-- Lấy mã sản phẩm ngẫu nhiên từ bảng sanPham

SET @maSP = ROUND(RAND() \* 1000, 0) + 1;

-- Lấy giá bán từ bảng sanPham

SET @giaBan = (

SELECT giaBan

FROM sanPham

WHERE maSP = @maSP

);

-- Tạo số lượng ngẫu nhiên từ 1 đến 10

SET @soLuong = ROUND(RAND() \* 10, 0) + 1;

-- Tính thành tiền

SET @thanhTien = @giaBan \* @soLuong;

-- Thêm dữ liệu vào bảng donHangBanCT

INSERT INTO donHangBanCT (maDHB, maSP, giaBan, soLuong, thanhTien)

VALUES (

ROUND(RAND() \* 1000, 0) + 1, -- Mã đơn hàng bán ngẫu nhiên từ 1-1000

@maSP, -- Mã sản phẩm ngẫu nhiên

@giaBan, -- Giá bán lấy từ bảng sanPham

@soLuong, -- Số lượng

@thanhTien -- Thành tiền (giaBan \* soLuong)

);

SET @i = @i + 1;

END;

SELECT \* FROM donHangBanCT;

### Tạo Dữ Liệu cho Bảng donHangNhap

GO

DECLARE @i INT = 1;

DECLARE @tongTien NUMERIC(15, 0);

WHILE @i <= 1000

BEGIN

-- Tạo tổng tiền bằng tổng các thanhTien từ bảng donHangNhapCT

SET @tongTien = (

SELECT SUM(thanhTien)

FROM donHangNhapCT

WHERE maDHN = @i

);

-- Nếu chưa có dòng nào trong bảng chi tiết, mặc định tổng tiền = 0

IF @tongTien IS NULL

SET @tongTien = 0;

INSERT INTO donHangNhap (maNCC, ngayTao, tongTien)

VALUES (

ROUND(RAND() \* 100, 0) + 1, -- Mã nhà cung cấp ngẫu nhiên

DATEADD(DAY, @i, GETDATE()), -- Ngày tạo ngẫu nhiên tăng dần

@tongTien -- Tổng tiền từ bảng chi tiết

);

SET @i = @i + 1;

END;

SELECT \* FROM donHangNhap;

### Tạo Dữ Liệu cho Bảng donHangNhapCT

GO

DECLARE @i INT = 1;

DECLARE @maSP INT, @giaNhap NUMERIC(15,0), @soLuong INT, @thanhTien NUMERIC(15,0);

SET @i = 1;

WHILE @i <= 1000

BEGIN

-- Lấy mã sản phẩm ngẫu nhiên từ bảng sanPham

SET @maSP = ROUND(RAND() \* 1000, 0) + 1;

-- Lấy giá nhập từ bảng sanPham

SET @giaNhap = (

SELECT giaNhap

FROM sanPham

WHERE maSP = @maSP

);

-- Tạo số lượng ngẫu nhiên từ 1 đến 10

SET @soLuong = ROUND(RAND() \* 10, 0) + 1;

-- Tính thành tiền

SET @thanhTien = @giaNhap \* @soLuong;

-- Thêm dữ liệu vào bảng donHangNhapCT

INSERT INTO donHangNhapCT (maDHN, maSP, giaNhap, soLuong, thanhTien)

VALUES (

ROUND(RAND() \* 1000, 0) + 1, -- Mã đơn hàng nhập ngẫu nhiên từ 1-1000

@maSP, -- Mã sản phẩm ngẫu nhiên

@giaNhap, -- Giá nhập lấy từ bảng sanPham

@soLuong, -- Số lượng

@thanhTien -- Thành tiền (giaNhap \* soLuong)

);

SET @i = @i + 1;

END;

SELECT \* FROM donHangNhapCT;

### Tạo Dữ Liệu cho Bảng noPhaiTraKH

GO

DECLARE @i INT = 1;

SET @i = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO noPhaiTraKH (maDHB, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

VALUES (

ROUND(RAND() \* 1000, 0) + 1, -- Mã đơn hàng bán ngẫu nhiên

ROUND(RAND() \* 1000000, 0), -- Tiền nợ ngẫu nhiên

DATEADD(DAY, -(@i % 365), GETDATE()), -- Ngày nợ tiền

0, -- Trạng thái chưa trả tiền

NULL -- Ngày trả tiền

);

SET @i = @i + 1;

END;

SELECT \* FROM noPhaiTraKH

### Tạo Dữ Liệu cho Bảng noPhaiTraNCC

GO

DECLARE @i INT = 1;

SET @i = 1;

WHILE @i <= 1000

BEGIN

INSERT INTO noPhaiTraNCC (maDHN, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)

VALUES (

ROUND(RAND() \* 1000, 0) + 1, -- Mã đơn hàng nhập ngẫu nhiên

ROUND(RAND() \* 1000000, 0), -- Tiền nợ ngẫu nhiên

DATEADD(DAY, -(@i % 365), GETDATE()), -- Ngày nợ tiền

0, -- Trạng thái chưa trả tiền

NULL -- Ngày trả tiền

);

SET @i = @i + 1;

END;

SELECT \* FROM noPhaiTraNCC

## Module nghiệp vụ

Nhóm đã thay đổi hoàn toàn các module nghiệp vụ

### Thủ tục

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA SẢN PHẨM TRƯỚC KHI INSERT** | |
| **INPUT** | **Tên sản phẩm, Số lượng, Giá nhập, Giá bán, HSD** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | Tiến hành kiểm tra:   * Sản phẩm đã tồn tại chưa, rồi thì báo lỗi * Kiểm tra hạn sử dụng có lớn hơn ngày hiện tại không, không thì báo lỗi   Kiểm tra xong thì insert sản phẩm vào bảng sản phẩm |
|  | CREATE OR ALTER PROC kiemTraVaChenSanPham  (@tenSP NVARCHAR(50), -- Tên sản phẩm  @soLuongSP INT, -- Số lượng sản phẩm  @giaNhap NUMERIC(15, 0), -- Giá nhập  @giaBan NUMERIC(15, 0), -- Giá bán  @HSD DATE, -- Hạn sử dụng  @ret BIT OUT) -- Kết quả trả về (1: Thành công, 0: Lỗi)  AS  BEGIN  -- Kiểm tra sản phẩm đã tồn tại chưa (dựa trên tên sản phẩm)  IF EXISTS (SELECT 1 FROM sanPham WHERE tenSP = @tenSP)  BEGIN  PRINT 'San pham da ton tai!';  SET @ret = 0;  RETURN;  END  -- Kiểm tra hạn sử dụng có lớn hơn ngày hiện tại không  IF @HSD <= GETDATE()  BEGIN  PRINT 'Han su dung khong hop le (phai lon hon ngay hien tai)';  SET @ret = 0;  RETURN;  END  declare @hinhAnhSP VARBINARY(MAX)  set @hinhAnhSP = null;  -- Chèn sản phẩm mới vào bảng sanPham  INSERT INTO sanPham (tenSP, soLuongSP, giaNhap, giaBan, hinhAnhSP, HSD)  VALUES (@tenSP, @soLuongSP, @giaNhap, @giaBan, @hinhAnhSP, @HSD);  if @@ROWCOUNT > 0  begin  PRINT 'Da insert san pham thanh cong';  SET @ret = 1;  end  else  begin  PRINT 'Insert that bai';  SET @ret = 0;  rollback  end  END; |

Bảng 4‑1. Thủ tục kiểm tra sản phẩm trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA ĐƠN HÀNG BÁN TRƯỚC KHI INSERT** | |
| **INPUT** | **Mã khách hàng** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | * Kiểm tra xem đã tồn tại khách hàng chưa, nếu chưa thì báo lỗi. Kiểm tra xong thì insert đơn hàng bán * Để ngày tạo là ngày hiện tại * Để tổng tiền mặc định là 0   Kiểm tra xong thì insert sản phẩm vào bảng đơn hàng bán |
|  | create or alter proc kiemTraDonHangBan (@SDT CHAR(10), @ret bit out)  as  begin  declare @ngayTao date = getdate()  declare @tongTien numeric (15, 0) = 0  declare @maKH int  --Kiểm tra đã tồn tại khach hàng chưa, nếu chưa thì thông báo lỗi  if not exists (select 1 from khachHang where SDT=@SDT)  begin  print N'Không tồn tại khach hang'  set @ret = 0  return  end  select @maKH = maKH  from khachHang where SDT = @SDT  insert into donHangBan(maKH, ngayTao, tongTien)  values (@maKH, @ngayTao, @tongTien)  if @@ROWCOUNT > 0  begin  PRINT 'Da insert thanh cong';  SET @ret = 1;  end  else  begin  PRINT 'Insert that bai';  SET @ret = 0;  rollback  end  end |

Bảng 4‑2. Thủ tục kiểm tra đơn hàng bán trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA ĐƠN HÀNG BÁN CHI TIẾT TRƯỚC KHI INSERT** | |
| **INPUT** | **Mã đơn hàng bán, Mã sản phẩm, Số lượng** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | Tiến hành kiểm tra:   * Kiểm tra đơn hàng bán tồn tại chưa, chưa thì báo lỗi * Kiểm tra sản phẩm tồn tại chưa, chưa thì báo lỗi * Kiểm tra số lượng có lớn hơn số lượng tồn kho hay không, nếu lớn hơn thì báo lỗi * Cập nhật thành tiền = số lượng \* giá bán   Kiểm tra xong thì insert vào bảng đơn hàng bán chi tiết |
|  | create or alter proc themDHBCT(@maDHB int, @tenSP nvarchar(100), @soLuong int, @ret bit out)  as  begin  declare @maSP int  select @maSP = maSP from sanPham where tenSP = @tenSP  -- 1. Kiem tra maDHB ton tai chua  if not exists (select 1 from donHangBan where maDHB = @maDHB)  begin  print N'Chua ton tai maDHB'  set @ret = 0  return  end  -- 2. Kiem tra SP ton tai chua  if not exists (select 1 from sanPham where maSP = @maSP)  begin  print N'Khong ton tai san pham'  set @ret = 0  return  end  -- 3. Kiem tra soLuong  declare @soLuongTonKho int  select @soLuongTonKho = soLuongSP  from sanPham  where maSP = @maSP  if @soLuong > @soLuongTonKho  begin  print N'So luong ton kho khong du'  set @ret = 0  return  end    declare @giaBan numeric(15,0)  select @giaBan = giaBan  from sanPham  where maSP = @maSP  declare @thanhTien int  set @thanhTien = @soLuong\*@giaBan  insert into donHangBanCT(maDHB, maSP, giaBan, soLuong, thanhTien)  values (@maDHB, @maSP, @giaBan, @soLuong, @thanhTien)  if @@ROWCOUNT>0  begin  print N'Thêm dữ liệu thành công'  set @ret = 1  end  else  begin  print N'Thất bại'  set @ret = 0  rollback  end  end |

Bảng 4‑3. Thủ tục kiểm tra đơn hàng bán chi tiết trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA NỢ KHÁCH HÀNG TRƯỚC KHI INSERT** | |
| **INPUT** | **Mã đơn hàng bán (khách hàng nợ thông qua đơn hàng bán), Tiền nợ, Ngày nợ tiền** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | Tiến hành kiểm tra:   * Kiểm tra mã đơn hàng bán tồn tại chưa (vì khách hàng nợ thông qua đơn hàng bán), nếu chưa thì báo lỗi * Kiểm tra tiền nợ có lớn hơn tiền nợ của đơn hàng bán đó, nếu lớn hơn thì báo lỗi * Kiểm tra ngày nợ tiền có lớn hơn ngày hiện tại không, nếu có thì báo lỗi * Để trạng thái trả tiền mặc định là 0 (chưa trả tiền)   Kiểm tra xong thì insert sản phẩm vào bảng nợ khách hàng |
|  | create or alter proc spNPTKH (@maDHB int , @tienNo numeric (15,0) , @ngayNoTien date,@ret bit output)  as  begin  -- kiểm tra maDHb tồn tại chưa, chưa thì báo lỗi  if not exists (select 1 from donHangBan where @maDHB = maDHB)  begin  print N'Mã DHB không tồn tại'  set @ret = 0  return  end  --kiểm tra tiền nợ có lớn hơn tiền nợ của maDHB đó không, lớn hơn thì báo lỗi  declare @tongtien numeric (15,0)  select @tongtien = tongTien  from donHangBan  where @maDHB = maDHB  if @tienNo > @tongtien  begin  print N'Lỗi tiền nợ'  set @ret = 0  return  end  --ngày nợ tiền <= getdate  if @ngayNoTien > CAST(GETDATE() as date)  begin  print N'Lỗi ngày nợ tiền'  set @ret =0  return  end  --set trạng thái trả tiền mặc định là 0  declare @trangThaiTraTien bit = 0  --thêm nợ phải trả  insert into noPhaiTraKH(maDHB, tienNo,ngayNoTien,trangThaiTraTien)  values (@maDHB, @tienNo, @ngayNoTien, @trangThaiTraTien)  if @@ROWCOUNT > 0  begin  print N'Thêm thành công'  set @ret =1  end  else  begin  print N'Thêm thất bại'  set @ret =0  end  end |

Bảng 4‑4. Thủ tục kiểm tra nợ khách hàng trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA ĐƠN HÀNG NHẬP TRƯỚC KHI INSERT** | |
| **INPUT** | **Mã nhà cung cấp** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | * Kiểm tra xem đã tồn tại nhà cung cấp chưa, nếu chưa thì báo lỗi. Kiểm tra xong thì insert đơn hàng bán * Để ngày tạo là ngày hiện tại * Để tổng tiền mặc định là 0   Kiểm tra xong thì insert sản phẩm vào bảng đơn hàng nhập |
|  | create or alter proc kiemTraDonHangNhap (@maNCC int, @ret bit out)  as  begin  declare @ngayTao date = getdate()  declare @tongTien numeric(15, 0) = 0  --Kiểm tra ngày tạo đơn hàng có hợp lệ hay không  --Kiểm tra đã tồn tại hàng chưa, nếu chưa thì thông báo lỗi  if not exists (select 1 from donHangNhap where maNCC=@maNCC)  begin  print N'Không tồn tại NCC'  set @ret = 0  return  end  insert into donHangNhap(maNCC, ngayTao, tongTien)  values (@maNCC, @ngayTao, @tongTien)  if @@ROWCOUNT > 0  begin  PRINT 'Da insert thanh cong';  SET @ret = 1;  end  else  begin  PRINT 'Insert that bai';  SET @ret = 0;  rollback  end  end |

Bảng 4‑5. Thủ tục kiểm tra đơn hàng nhập trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA ĐƠN HÀNG NHẬP CHI TIẾT TRƯỚC KHI INSERT** | |
| **INPUT** | **Mã đơn hàng nhập, Mã sản phẩm, Giá nhập, Số lượng** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | Tiến hành kiểm tra:   * Kiểm tra đơn hàng nhập tồn tại chưa, chưa thì báo lỗi * Kiểm tra sản phẩm tồn tại chưa, yêu cầu thêm sản phẩm * Kiểm tra giá nhập có < 0 hay không, nếu bé hơn thì báo lỗi * Kiểm tra số lượng có < 0 hay không, nếu bé hơn thì báo lỗi * Cập nhật thành tiền = số lượng \* giá nhập   Kiểm tra xong thì insert vào bảng đơn hàng nhập chi tiết |
|  | Create or alter proc spDHNct (@maDHN int, @maSP int , @giaNhap numeric (15,0) , @soLuong int , @ret bit output)  as  begin  -- 1. kiểm tra maDHN tồn tại chưa, chưa thì báo lỗi  if not exists ( select 1 from [dbo].[donHangNhap] where @maDHN = maDHN )  begin  print N'maDHN chưa tồn tại'  set @ret = 0  return  end  -- 2. kiểm tra maSP tồn tại chưa, chưa thì báo lỗi  if not exists ( select 1 from [dbo].[sanPham] where @maSP = maSP )  begin  print N'maSP chưa tồn tại'  set @ret = 0  return  end  -- 3. kiểm tra giá nhập có < 0 ko, bé hơn thì báo lỗi  if @giaNhap < 0  begin  print N'Lỗi : giaNhap '  set @ret = 0  return  end  -- 4.kiểm tra số lượng có <0 ko, bé hơn thì báo lỗi  if @soLuong < 0  begin  print N'Lỗi : giaNhap '  set @ret = 0  return  end  -- 5. set thành tiền = số lượng \* giaNhap  declare @thanhTien numeric(15,0)  set @thanhTien = @giaNhap \* @soLuong  --Thêm đơn hàng chi tiết mới  insert into [dbo].[donHangNhapCT] (maDHN,maSP,giaNhap,soLuong,thanhTien)  values (@maDHN,@maSP,@giaNhap,@soLuong,@thanhTien)  if @@ROWCOUNT > 0  begin  print N'Thêm thành công '  set @ret = 1  end  else  begin  print N'Thêm thất bại '  set @ret = 0  end  end |

Bảng 4‑6. Thủ tục kiểm tra đơn hàng nhập chi tiết trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA NỢ NHÀ CUNG CẤP TRƯỚC KHI INSERT** | |
| **INPUT** | **Mã đơn hàng nhập (nợ nhà cung cấp thông qua đơn hàng nhập), Tiền nợ, Ngày nợ tiền** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | Tiến hành kiểm tra:   * Kiểm tra mã đơn hàng nhập tồn tại chưa (vì nợ nhà cung câp thông qua đơn hàng nhập), nếu chưa thì báo lỗi * Kiểm tra tiền nợ có lớn hơn tiền nợ của đơn hàng nhập đó, nếu lớn hơn thì báo lỗi * Kiểm tra ngày nợ tiền có lớn hơn ngày hiện tại không, nếu có thì báo lỗi * Để trạng thái trả tiền mặc định là 0 (chưa trả tiền)   Kiểm tra xong thì insert sản phẩm vào bảng nợ nhà cung cấp |
|  | CREATE OR ALTER PROC kiemTraVaCapNhatNoPhaiTraNCC  (@maDHN INT, -- Mã đơn hàng nhập  @tienNo NUMERIC(15, 0), -- Số tiền nợ  @ngayNoTien DATE, -- Ngày nợ tiền  @ret BIT OUT) -- Trả về kết quả (1: Thành công, 0: Lỗi)  AS  BEGIN  -- Kiểm tra mã đơn hàng nhập có tồn tại không  IF NOT EXISTS (SELECT 1 FROM donHangNhap WHERE maDHN = @maDHN)  BEGIN  PRINT 'Loi: Ma don hang nhap khong ton tai!';  SET @ret = 0;  RETURN;  END  -- Kiểm tra tiền nợ có lớn hơn tổng tiền của đơn hàng nhập hay không  DECLARE @tongTien NUMERIC(15, 0);  SELECT @tongTien = tongTien FROM donHangNhap WHERE maDHN = @maDHN;  IF @tienNo > @tongTien  BEGIN  PRINT 'Loi: Tien no vuot qua tong tien cua don hang nhap!';  SET @ret = 0;  RETURN;  END  -- Kiểm tra ngày nợ tiền có hợp lệ hay không  IF @ngayNoTien > GETDATE()  BEGIN  PRINT 'Loi: Ngay no tien phai nho hon hoac bang ngay hien tai!';  SET @ret = 0;  RETURN;  END  -- Chèn thông tin nợ vào bảng noPhaiTraNCC với trạng thái trả tiền mặc định là 0  INSERT INTO noPhaiTraNCC (maDHN, tienNo, ngayNoTien, trangThaiTraTien, ngayTraTien)  VALUES (@maDHN, @tienNo, @ngayNoTien, 0, NULL);  PRINT 'Thong tin no phai tra da duoc cap nhat thanh cong!';  SET @ret = 1;  END; |

Bảng 4‑7. Thủ tục kiểm tra nợ nhà cung cấp trước khi insert

|  |  |
| --- | --- |
| **TÊN: THỦ TỤC KIỂM TRA KHÁCH HÀNG TRƯỚC KHI INSERT** | |
| **INPUT** | **Họ tên , SDT** |
| **OUTPUT** | Thông báo insert thành công |
| **XỬ LÝ** | Tiến hành kiểm tra:   * Kiểm tra SDT đã tồn tại trong bảng KhachHang chưa   Kiểm tra xong thì insert thông tin vào bảng khách hàng |
|  | create or alter proc kTraKH(@SDT CHAR(10), @hoTen nvarchar(50), @ret bit out)  as  begin  if exists (select 1 from khachHang where SDT = @SDT)  begin  print N'Đã tồn tại khách hàng'  set @ret = 0  end  insert into khachHang (SDT, hoTen)  values (@SDT, @hoTen)  if @@ROWCOUNT > 0  begin  print N'Thêm khách hàng thành công'  set @ret = 1  end  else  begin  print N'Thêm khách hàng thất bại'  set @ret = 0  end  end |

Bảng 4‑8. Thủ tục kiểm tra khách hàng trước khi insert

### Trigger

|  |  |
| --- | --- |
| **TÊN: Kiểm tra giá bán**  **LOẠI: Trigger** | |
| **MÔ TẢ:**  Khi update hoặc insert Giá bán vào bảng thì tiến hành kiểm tra xem giá bán nhập vào có bé hơn giá nhập không, nếu bé hơn thì báo lỗi | |
| **SỰ KIỆN** | **Insert, Update** |
| **KÝ SINH** | sanPham |
| **XỬ LÝ** | 1. Lấy giá bán từ bảng inserted 2. Kiểm tra giá bán: nếu giá bán > giá nhập thì báo lỗi, rollback transaction |
|  | Go  CREATE OR ALTER TRIGGER trg\_KiemTraGiaBan  ON sanPham  AFTER INSERT, UPDATE  AS  BEGIN  -- Kiểm tra điều kiện  IF EXISTS (  SELECT 1  FROM inserted i  WHERE i.giaBan > i.giaNhap  )  BEGIN  -- Rollback thao tác nếu điều kiện vi phạm  ROLLBACK TRANSACTION;  -- Thông báo lỗi  PRINT ' Khong hop le: Gia ban khong duoc lon hon gia nhap. Vui long chinh gia nhap truoc.';  END  END; |

Bảng 4‑9. Trigger kiểm tra giá bán

|  |  |
| --- | --- |
| **TÊN: Tính tổng tiền đơn hàng bán**  **LOẠI: Trigger** | |
| **MÔ TẢ:** Khi insert hoặc delete trong bảng đơn hàng bán chi tiết thì tiến hàng cập nhật lại tổng tiền trong bảng đơn hàng bán | |
| **SỰ KIỆN** | **Insert, Delete** |
| **KÝ SINH** | **donHangBanCT** |
| **XỬ LÝ** | 1. Lấy Thành tiền, Mã DHB từ bảng inserted 2. Lấy Thành tiền, Mã DHB từ bảng deleted 3. Nếu là thêm đơn hàng bán chi tiết thì cập nhập trong bảng donHangBan:Tổng tiền = Tổng tiền + Thành tiền với mã DHB đã thêm 4. Nếu là xóa đơn hàng bán chi tiết thì cập nhật trong bản donHangBan : Tổng tiền = Tổng tiền - Thành tiền với mã DHB đã xóa |
|  | go  create or alter trigger tinhTongTien  on donHangBanCT  after insert, delete  as  begin  declare @thanhTienIS numeric(15, 0), @thanhTienDL numeric(15, 0)  declare @maDHBIS int, @maDHBDL int  select @thanhTienIS = thanhTien, @maDHBIS = maDHB  from inserted  select @thanhTienDL = thanhTien, @maDHBDL = maDHB  from deleted  update donHangBan  set tongTien = tongTien + @thanhTienIS  where maDHB = @maDHBIS  update donHangBan  set tongTien = tongTien - @thanhTienDL  where maDHB = @maDHBDL  end |

Bảng 4‑10. Tính tổng tiền đơn hàng bán

|  |  |
| --- | --- |
| **TÊN: Cập nhật số lượng hàng tồn kho khi bán**  **LOẠI: Trigger** | |
| **MÔ TẢ:** Khi insert Đơn hàng bán chi tiết thì tiền hành cập nhật số lượng hàng tồn kho trong bảng sản phẩm | |
| **SỰ KIỆN** | **Insert** |
| **KÝ SINH** | **donHangBanCT** |
| **XỬ LÝ** | 1. Lấy số lượng đã bán , mã SP trong bảng inserted 2. Khi thêm mới đơn hàng bán chi tiết thì cập nhập trong bảng sản phẩm : Số lượng sản phẩm = số lượng sản phẩm – số lượng đã bán với mã SP đã thêm |
|  | go  create or alter trigger updateTonKho  on donHangBanCT  after insert  as  begin  declare @soLuongDaBan int, @maSP int  select @soLuongDaBan = soLuong, @maSP = maSP  from inserted  update sanPham  set soLuongSP = soLuongSP - @soLuongDaBan  where maSP = @maSP  end |

Bảng 4‑11. Trigger cập nhật số lượng tồn kho khi bán

|  |  |
| --- | --- |
| **TÊN: Kiểm tra ngày trả tiền của khách hàng**  **LOẠI: Trigger** | |
| **MÔ TẢ:** Khi update ngày trả tiền thì tiền hành kiểm tra ngày trả tiên có nhỏ hơn ngày nợ tiền không , nếu nhỏ hơn thì thông báo lỗi  Ngược lại thì tiền hành cập nhật trạng thái trả tiền | |
| **SỰ KIỆN** | **Update** |
| **KÝ SINH** | **noPhaiTraKH** |
| **XỬ LÝ** | 1. Lấy số ngày nợ tiền , ngày trả tiền , trạng thái trả tiền , mã DHB từ bảng inserted 2. Kiểm tra khi cập nhật ngày trả tiền trong bảng noPhatTraKH :  * Nếu ngày trả tiền > ngày nợ tiền thì cập nhật trạng thái trả tiền = 1 với mã DHB đã cập nhật trong bảng NoPhaiTraKH * Nếu ngày trả tiền < ngày nợ tiền thì thông báo “Lỗi : ngày nợ tiền không thể lớn hơn ngày trả tiền” và hủy thao tác |
|  | go  create trigger tnoKH  on noPhaiTraKH  after update  as  begin  declare @ngayNoTien date,  @ngayTraTien date,  @trangThaiTraTien bit,  @maDHB int  select @ngayNoTien = ngayNoTien,  @ngayTraTien = ngayTraTien,  @trangThaiTraTien = trangThaiTraTien,  @maDHB = maDHB  from inserted  if @ngayNoTien > @ngayTraTien  begin  print N'Lỗi : ngày nợ tiền không thể lớn hơn ngày trả tiền'  rollback  end  else  begin  update noPhaiTraKH  set trangThaiTraTien = 1  where maDHB = @maDHB  end  end |

Bảng 4‑12. Kiểm tra ngày trả tiền của khách hàng

|  |  |
| --- | --- |
| **TÊN: Tính tổng tiền đơn hàng nhập**  **LOẠI: Trigger** | |
| **MÔ TẢ:** Khi insert hoặc delete trong bảng đơn hàng nhập chi tiết thì tiến hàng cập nhật lại tổng tiền trong bảng đơn hàng nhập | |
| **SỰ KIỆN** | **Insert, Delete** |
| **KÝ SINH** | **donHangNhapCT** |
| **XỬ LÝ** | 1. Lấy Thành tiền, Mã DHN từ bảng inserted 2. Lấy Thành tiền, Mã DHN từ bảng deleted 3. Nếu là thêm đơn hàng nhập chi tiết thì cập nhập trong bảng donHangNhap:Tổng tiền = Tổng tiền + Thành tiền với mã DHN đã thêm 4. Nếu là xóa đơn hàng nhập chi tiết thì cập nhật trong bản donHangNhap : Tổng tiền = Tổng tiền - Thành tiền với mã DHN đã xóa |
|  | go  create or alter trigger tinhTongTienNhap  on donHangNhapCT  after insert, delete  as  begin  declare @thanhTienIS numeric(15, 0), @thanhTienDL numeric(15, 0)  declare @maDHNIS int, @maDHNDL int  select @thanhTienIS = thanhTien, @maDHNIS = maDHN  from inserted  select @thanhTienDL = thanhTien, @maDHNDL = maDHN  from deleted  update donHangNhap  set tongTien = tongTien + @thanhTienIS  where maDHN = @maDHNIS  update donHangNhap  set tongTien = tongTien - @thanhTienDL  where maDHN = @maDHNDL  end |

Bảng 4‑13. Tính tổng tiền đơn hàng nhập

|  |  |
| --- | --- |
| **TÊN: Cập nhật số lượng hàng tồn kho khi nhập**  **LOẠI: Trigger** | |
| **MÔ TẢ:** Khi insert Đơn hàng nhập chi tiết thì tiền hành cập nhật số lượng hàng tồn kho trong bảng sản phẩm | |
| **SỰ KIỆN** | **Insert** |
| **KÝ SINH** | **donHangNhapCT** |
| **XỬ LÝ** | 1. Lấy số lượng đã nhập , mã SP trong bảng inserted 2. Khi thêm mới đơn hàng nhập chi tiết thì cập nhập trong bảng sản phẩm : Số lượng sản phẩm = số lượng sản phẩm + số lượng đã nhập với mã SP đã thêm |
|  | go  create or alter trigger tDHNct  on donHangNhapCT  after insert  as  begin  declare @maSP int , @soLuongDaNhap int  select @soLuongDaNhap = soLuong,@maSP = maSP  from inserted  update sanPham  set soLuongSP = soLuongSP + @soLuongDaNhap  where maSP = @maSP  end |

Bảng 4‑14. Trigger cập nhật số lượng tồn kho khi nhập

|  |  |
| --- | --- |
| **TÊN: Kiểm tra ngày trả tiền nhà cung cấp**  **LOẠI: Trigger** | |
| **MÔ TẢ:** Khi update ngày trả tiền thì tiền hành kiểm tra ngày trả tiền có nhỏ hơn ngày nợ tiền không , nếu nhỏ hơn thì thông báo lỗi  Ngược lại thì tiền hành cập nhật trạng thái trả tiền | |
| **SỰ KIỆN** | **Update** |
| **KÝ SINH** | **noPhaiTraNCC** |
| **XỬ LÝ** | 1. Lấy số ngày nợ tiền , ngày trả tiền , trạng thái trả tiền , mã DHN từ bảng inserted 2. Kiểm tra khi cập nhật ngày trả tiền trong bảng noPhatTraNCC :  * Nếu ngày trả tiền > ngày nợ tiền thì cập nhật trạng thái trả tiền = 1 với mã DHN đã cập nhật trong bảng NoPhaiTraNCC * Nếu ngày trả tiền < ngày nợ tiền thì thông báo “Lỗi : ngày nợ tiền không thể lớn hơn ngày trả tiền” và hủy thao tác |
|  | go  create trigger tnoNCC  on noPhaiTraNCC  after update  as  begin  declare @ngayNoTien date,  @ngayTraTien date,  @trangThaiTraTien bit,  @maDHN int  select @ngayNoTien = ngayNoTien,  @ngayTraTien = ngayTraTien,  @trangThaiTraTien = trangThaiTraTien,  @maDHN = maDHN  from inserted  if @ngayNoTien > @ngayTraTien  begin  print N'Lỗi : ngày nợ tiền không thể lớn hơn ngày trả tiền'  rollback  end  else  begin  update noPhaiTraNCC  set trangThaiTraTien = 1  where maDHN = @maDHN  end  end |

Bảng 4‑15. Trigger kiểm tra ngày trả tiền cho nhà cung cấp

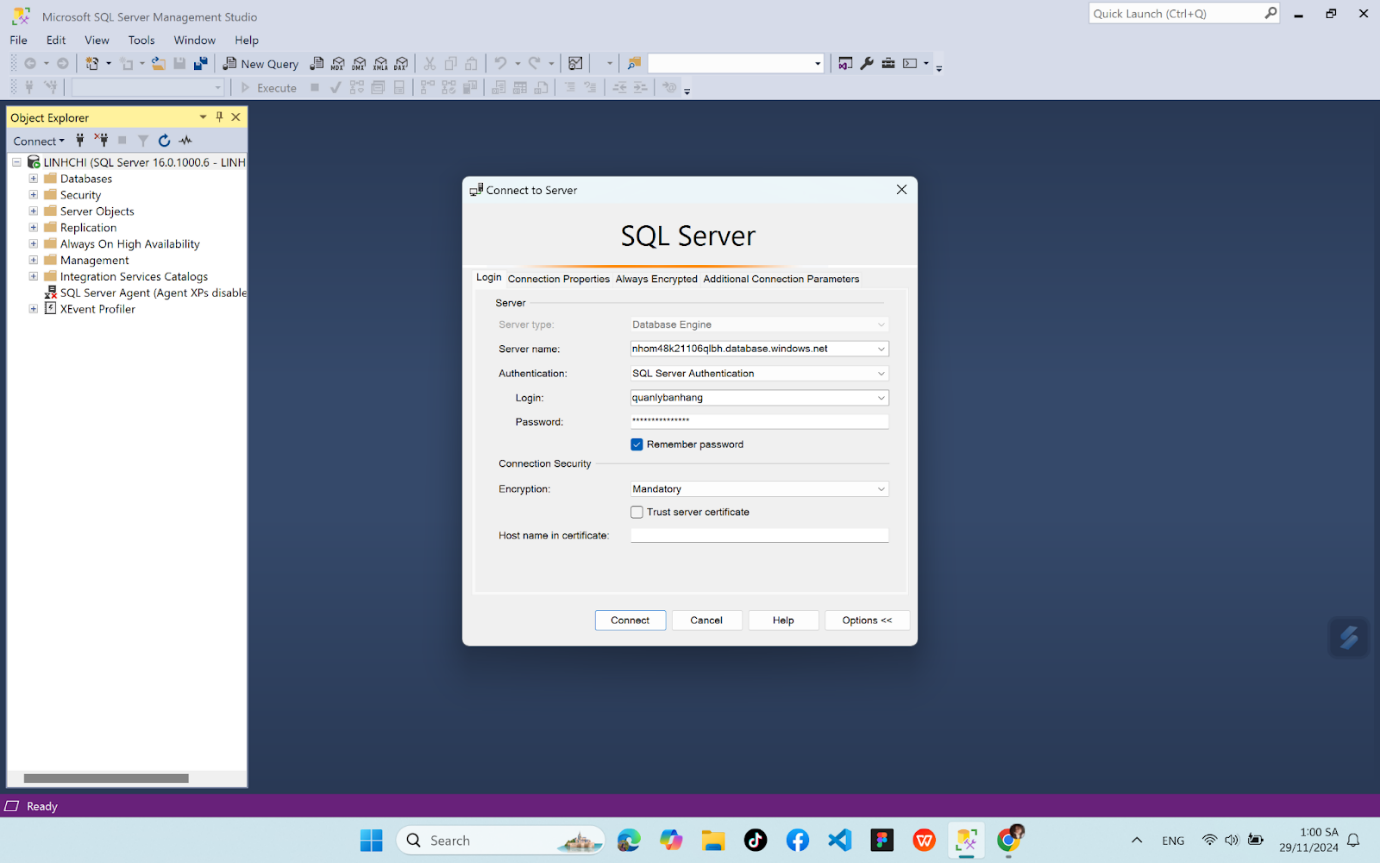
# Xây dựng cơ chế bảo mật phù hợp cho cơ sở dữ liệu

## Tạo và phân quyền phù hợp cho người dùng (ở cấp độ server và cấp độ database)

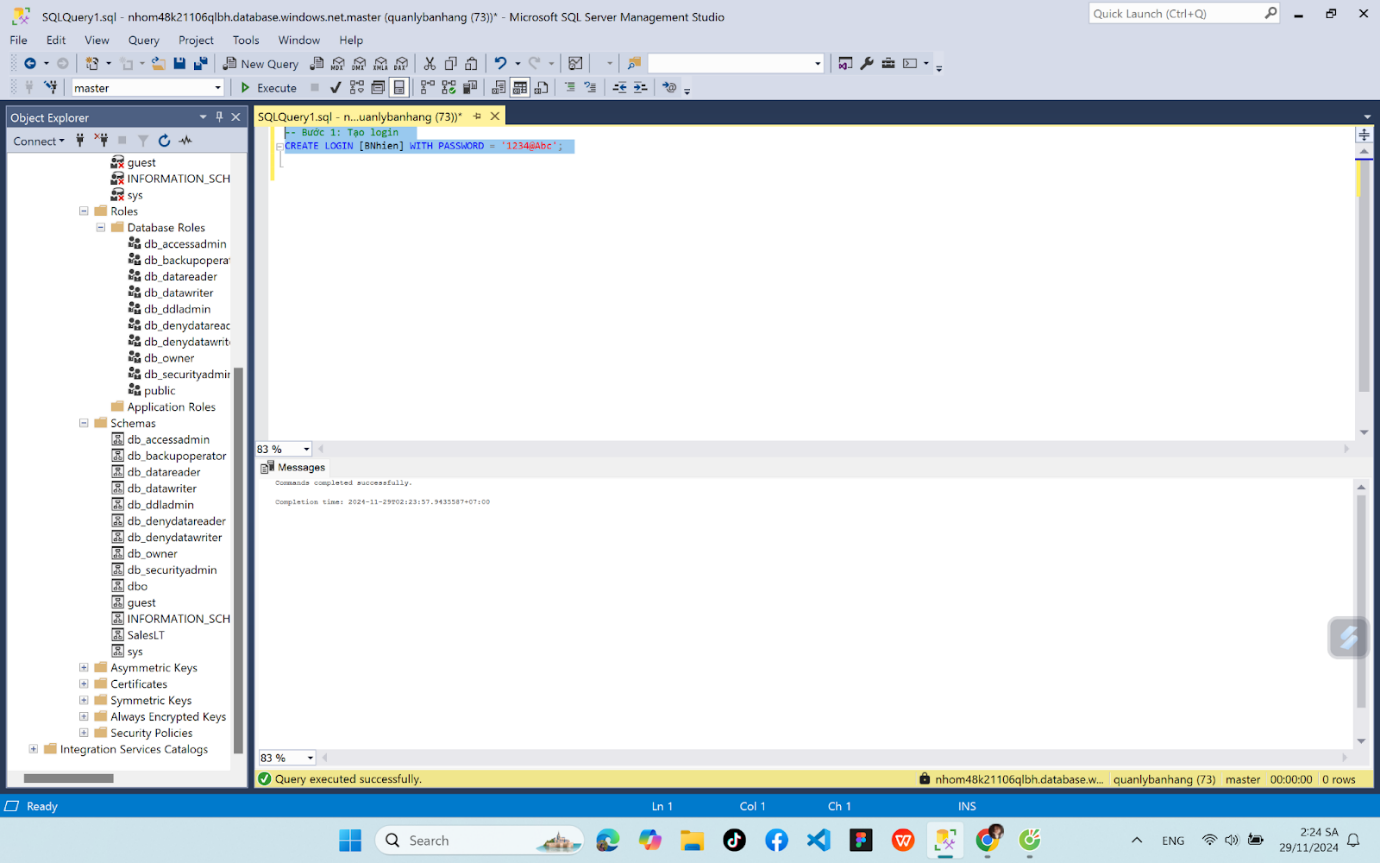
Bước 1:  Thiết lập thông tin cơ bản về login bao gồm: tên đăng nhập (login name), phương thức xác thực (windows, SQL server, mapped to certificate, mapped to asymetric key,…). Trong dự án này, Admin sẽ chọn SQL Server authentication.

Server name: nhom48k21106qlbh.database.windows.net

Mật khẩu: Abc@1231234

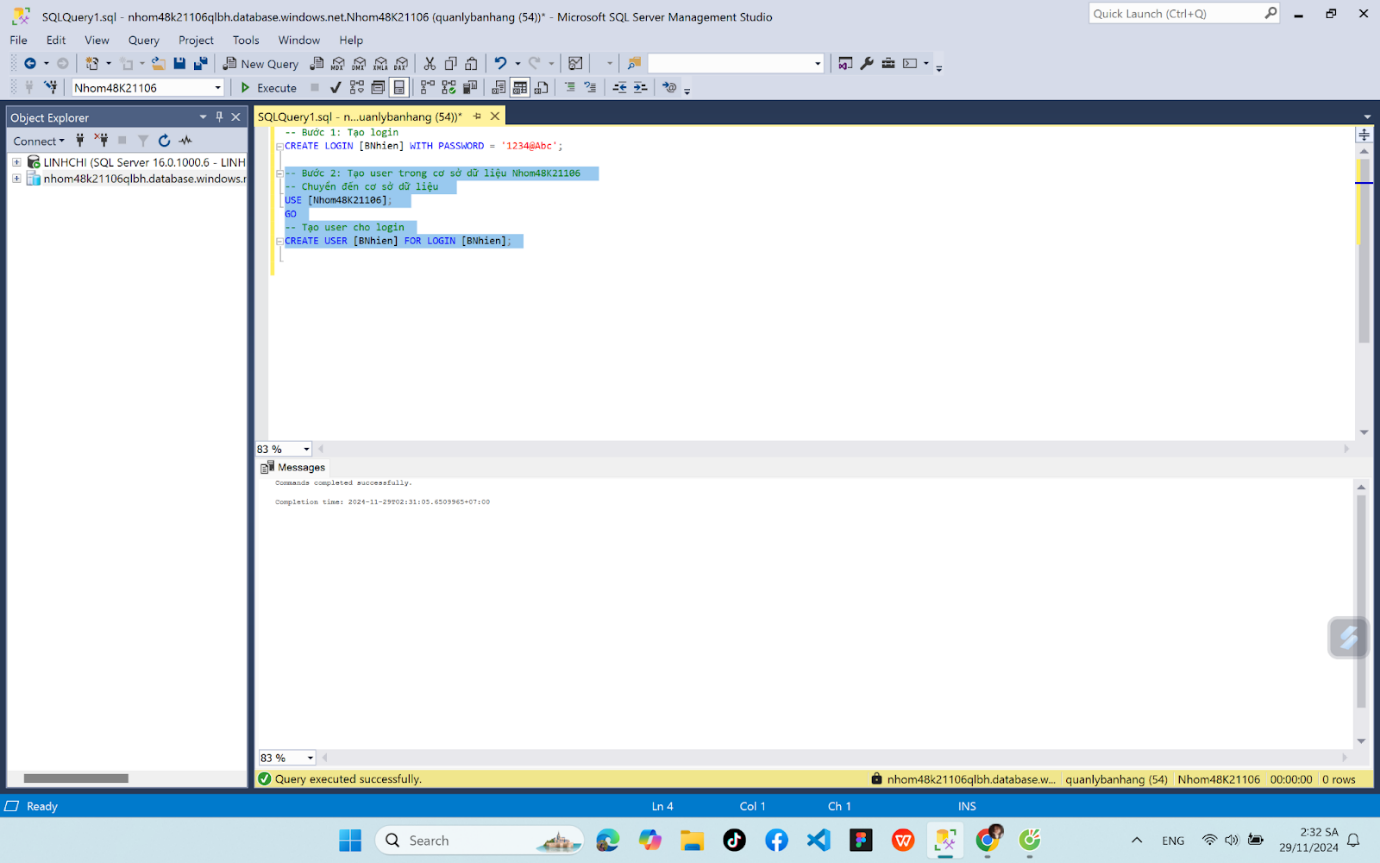
Hình 5‑1. Đăng nhập vào hệ thống SQL server bằng Azure server

Bước 2: Tạo tài khoản login là một tài khoản đăng nhập vào SQL Server. Tài khoản này có thể được sử dụng để đăng nhập vào bất kỳ cơ sở dữ liệu nào trên server.



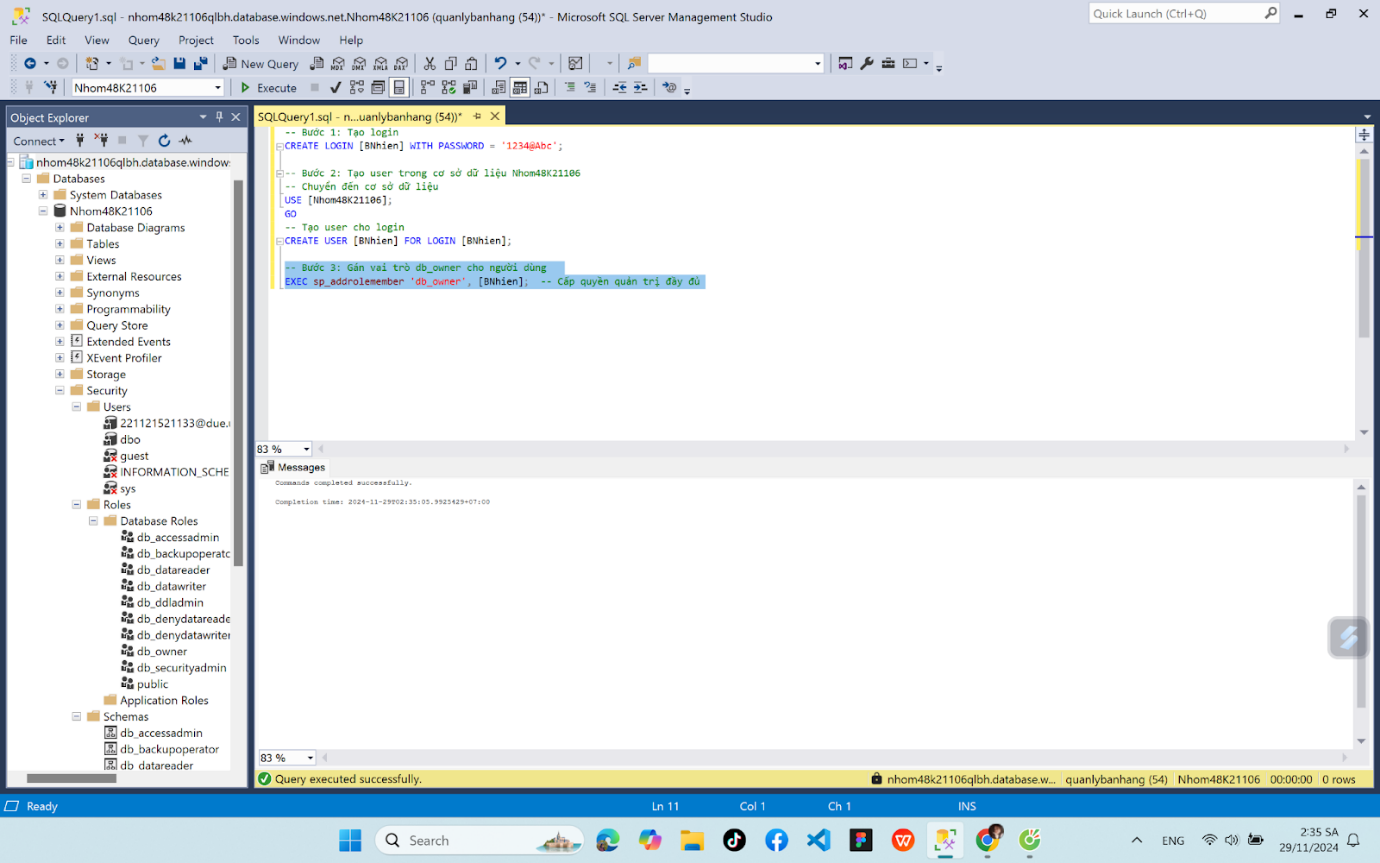
Hình 5‑2. Tạo tài khoản login

Bước 3: Sau khi tạo login, tạo user trong cơ sở dữ liệu cụ thể



Hình 5‑3. Tạo user trong cơ sở dữ liệu

Bước 4: Phân quyền phù hợp cho người dùng với vai trò db\_owner



Hình 5‑4. Phân quyền cho user vùa tạo

Bước 5: Kiểm tra quyền

Đăng nhập với:

Admin: SQL Server authentication.

User name: BNhien

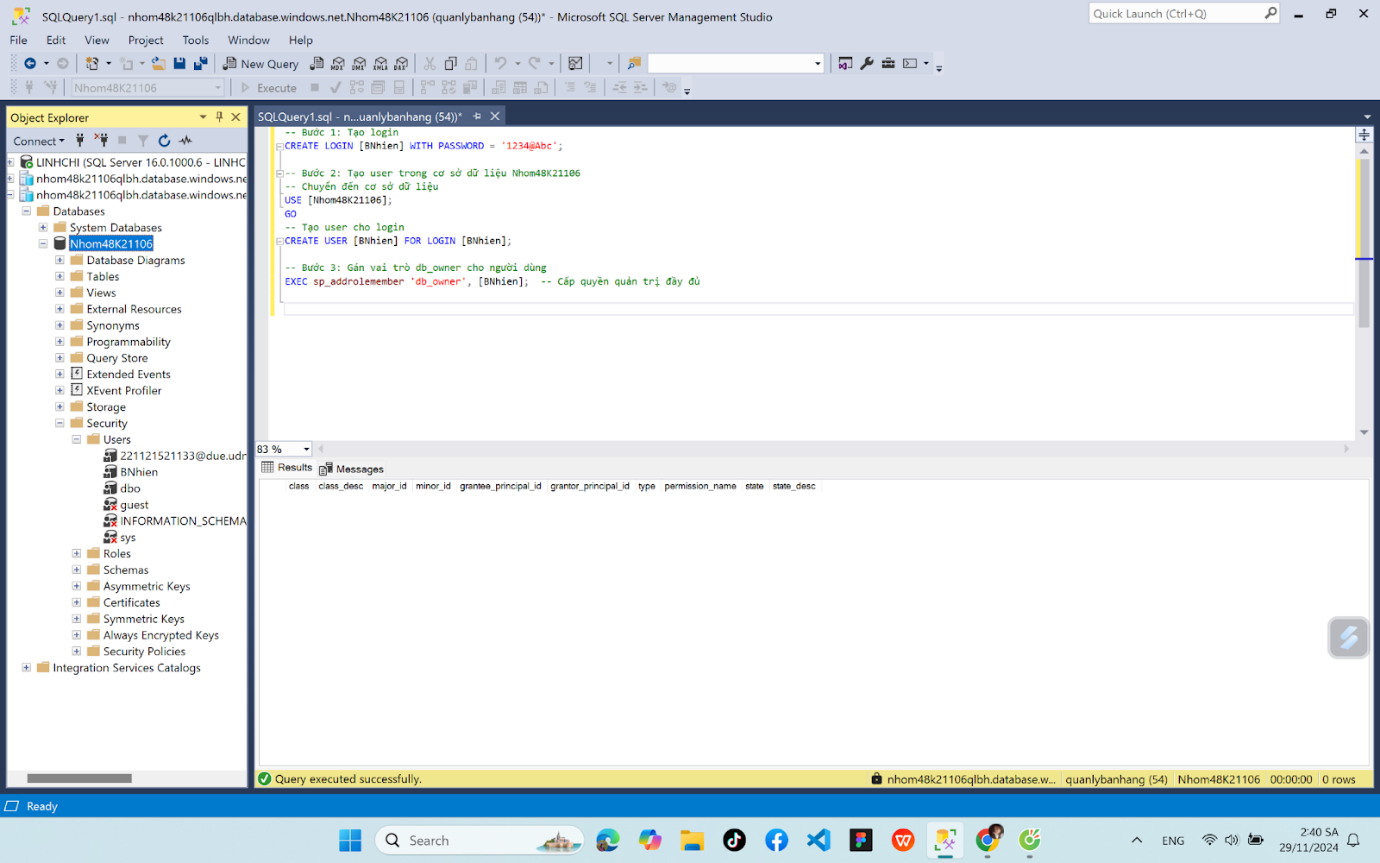
Password: 1234@Abc

Connect to database: Nhom48K21106



Hình 5‑5. Đăng nhập bằng user đã được phân quyền

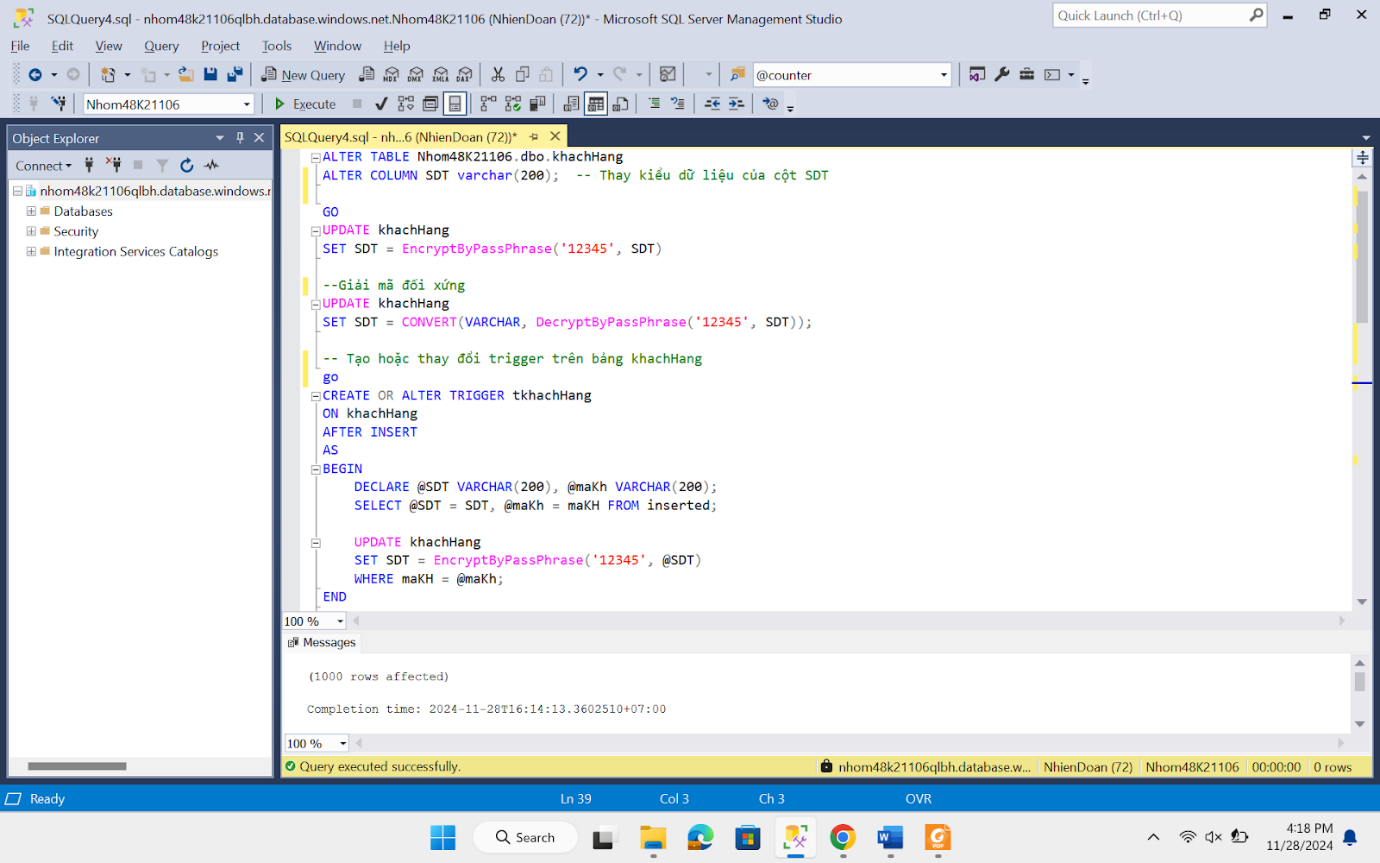
* Kiểm tra phân quyền



Hình 5‑6. Kiểm tra phân quyền

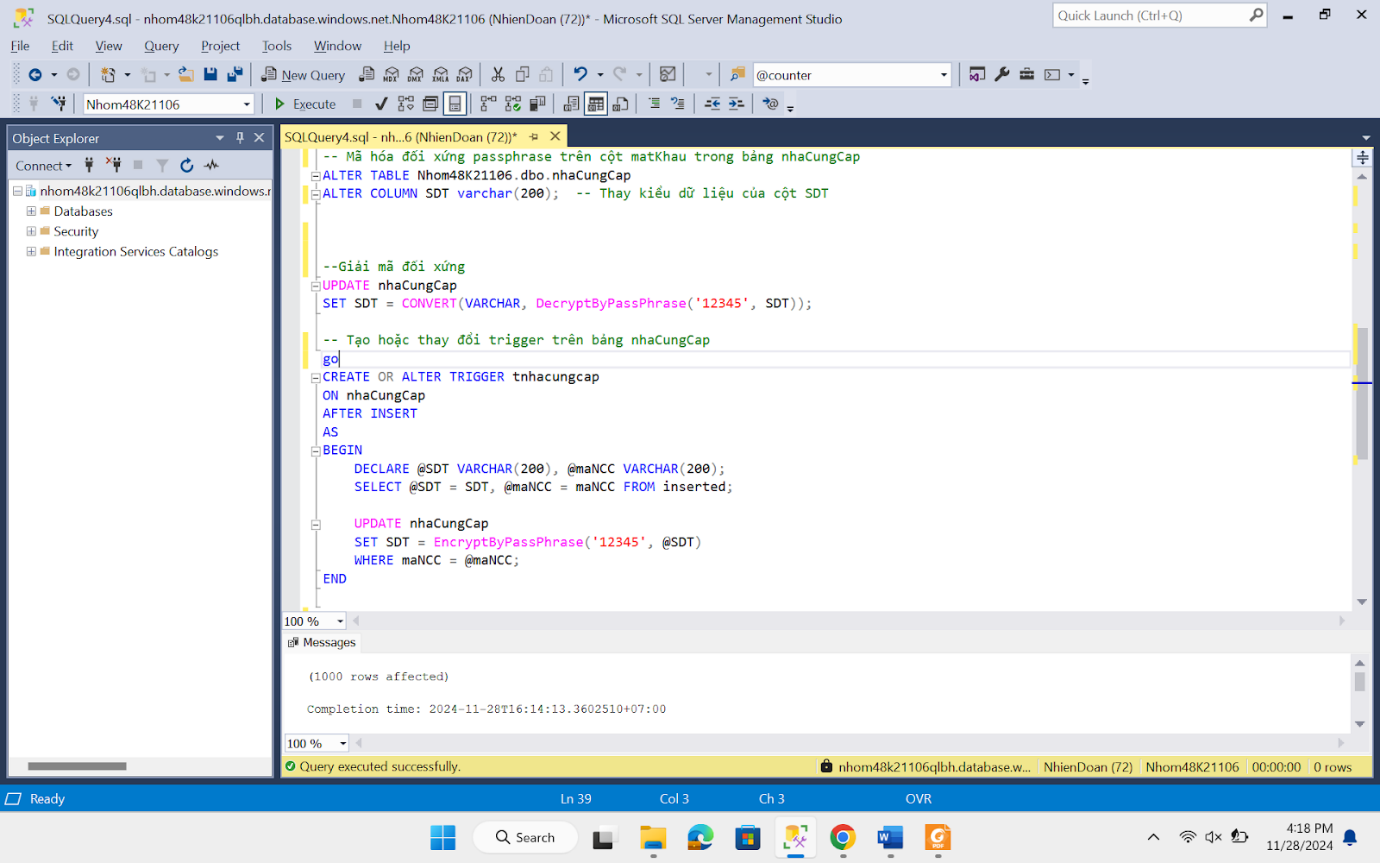
## Mã hoá dữ liệu cần bảo mật

* SDT khách hàng :



Hình 5‑7. Mã hoá dữ liệu số điện thoại khách hàng

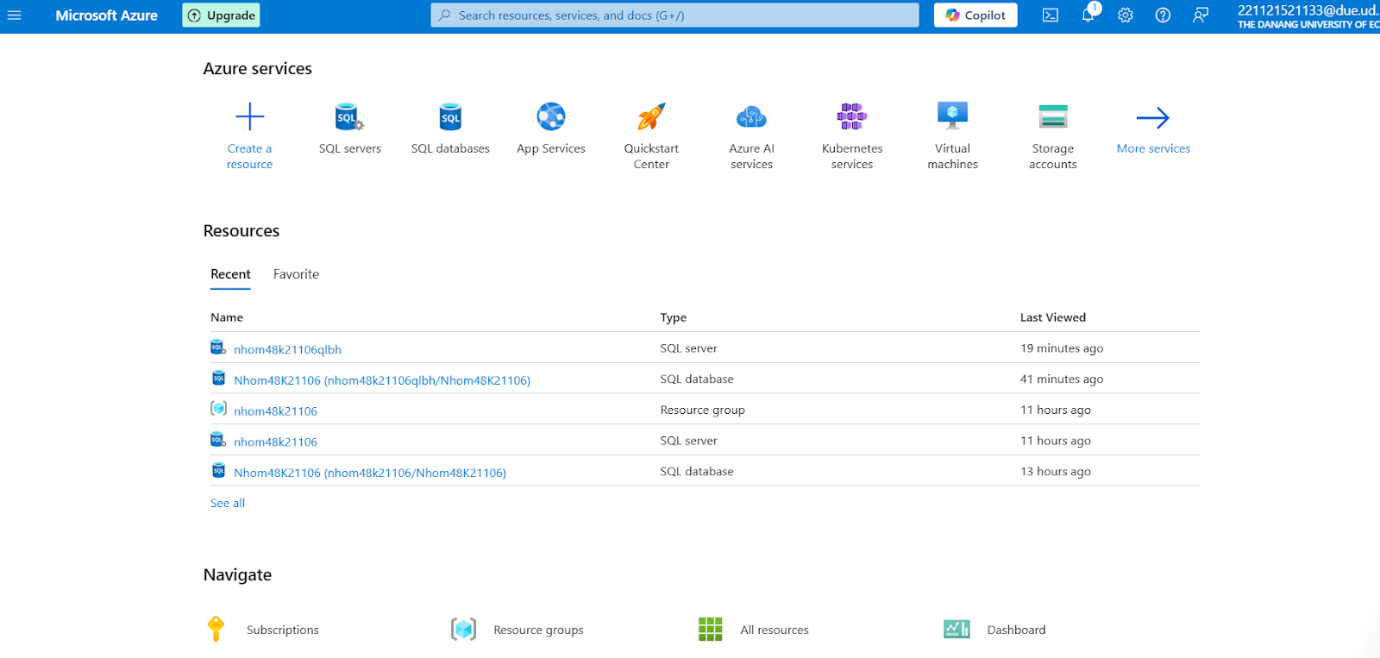
* SDT nhà cung cấp :



Hình 5‑8. Mã hoá dữ liệu số điện thoại của nhà cung cấp

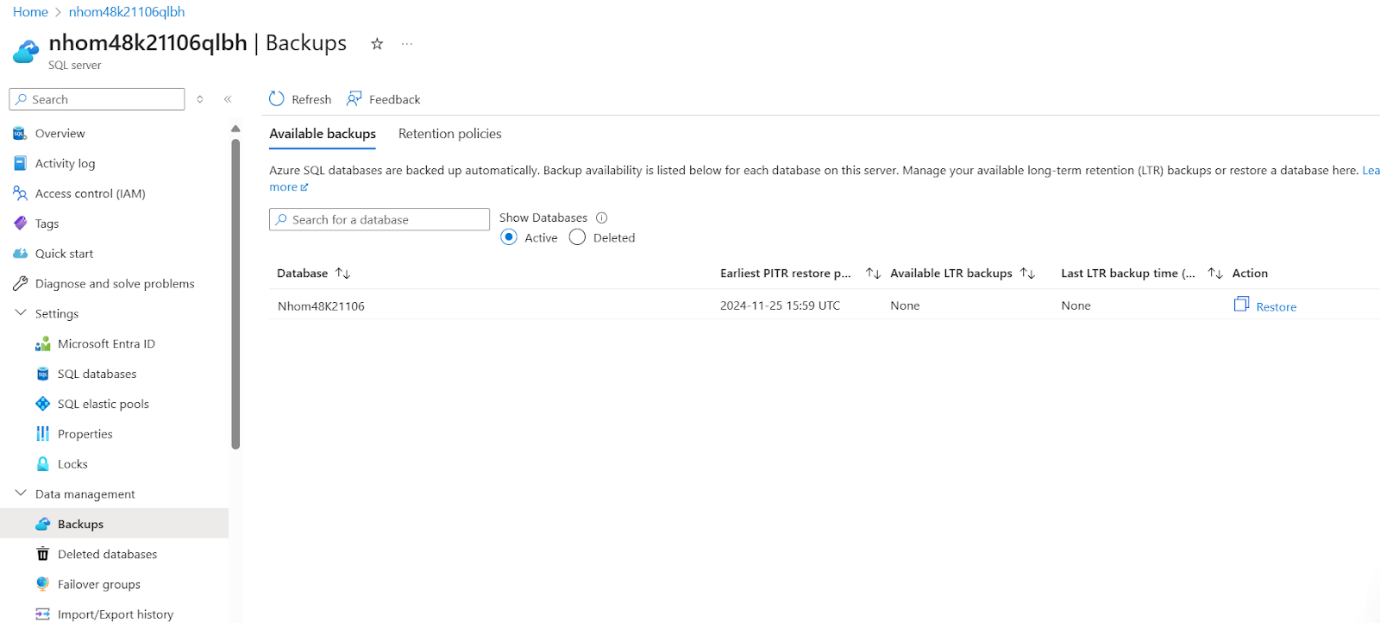
# Xây dựng cơ chế backup dữ liệu tự động cho cơ sở dữ liệu

Bước 1: Đăng nhập vào Microsoft Azure → chọn server **nhom48k2101qlbh**



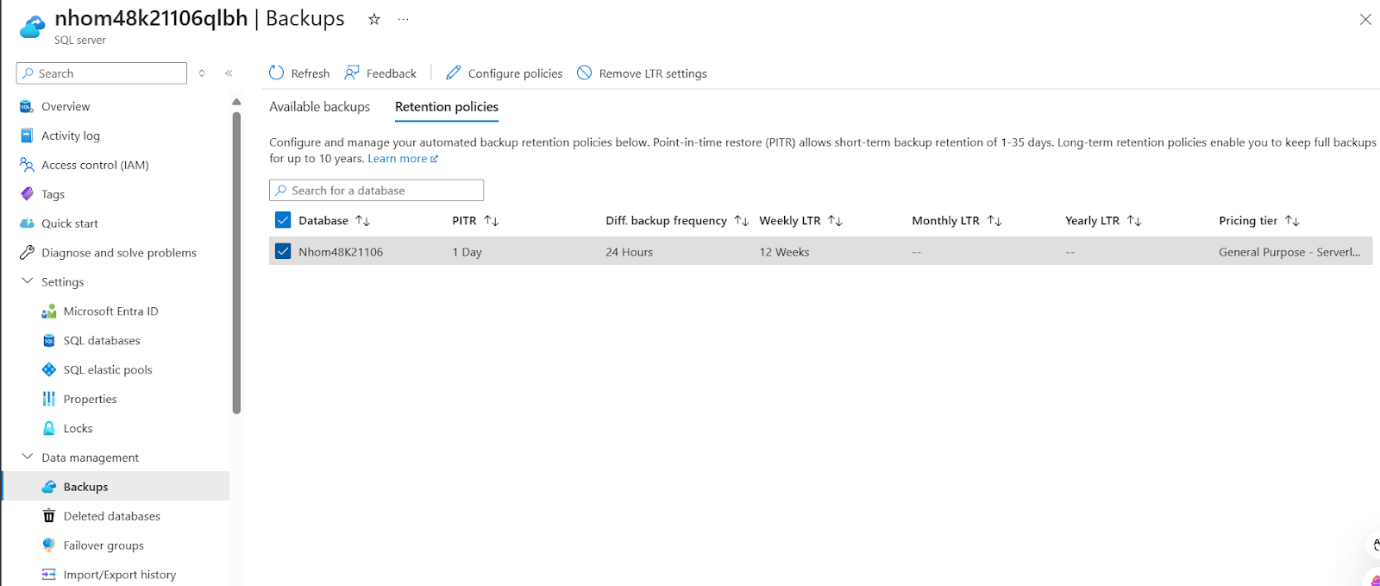
Hình 6‑1. Đăng nhập vào Microsoft Azure và chọn server nhom48k21106 qlbh

Bước 2: Trên thanh công cụ kích vào **Data** **management →** chọn **Backups**

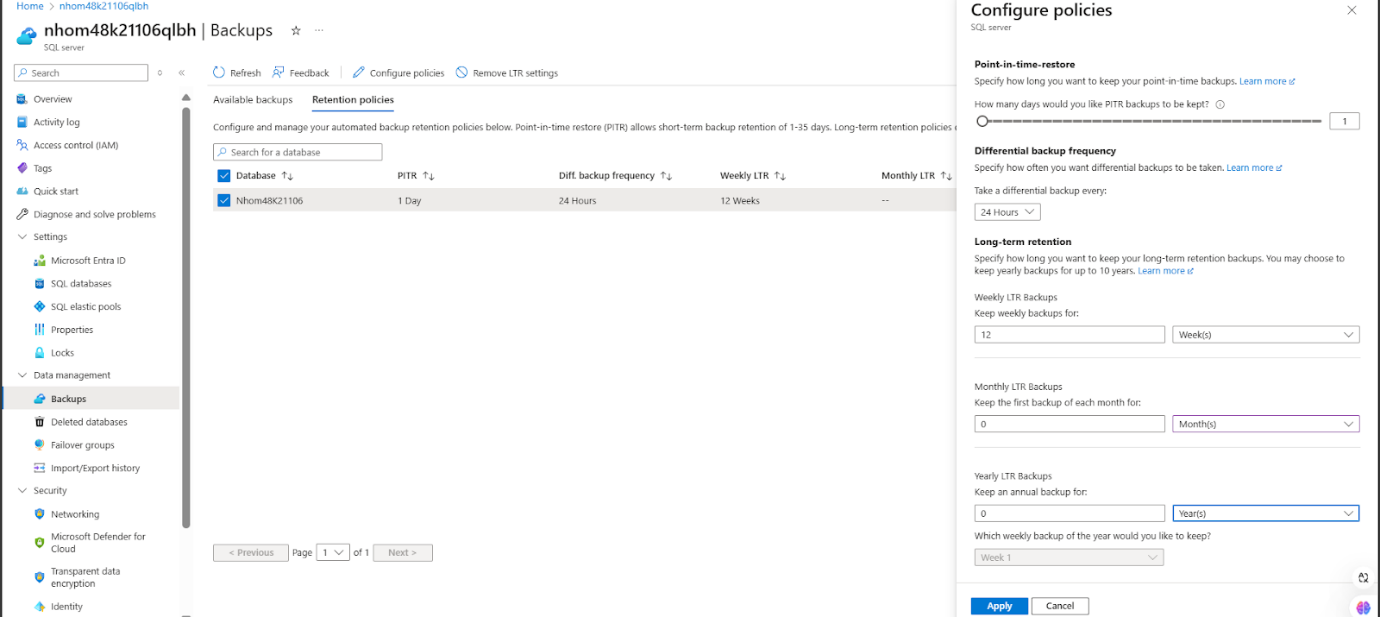


Hình 6‑2. Chọn Data **management →** chọn **Backups**

Bước 3: Nhấn vào **Retention policies** và nhấn tích vào **Nhom48K21106**

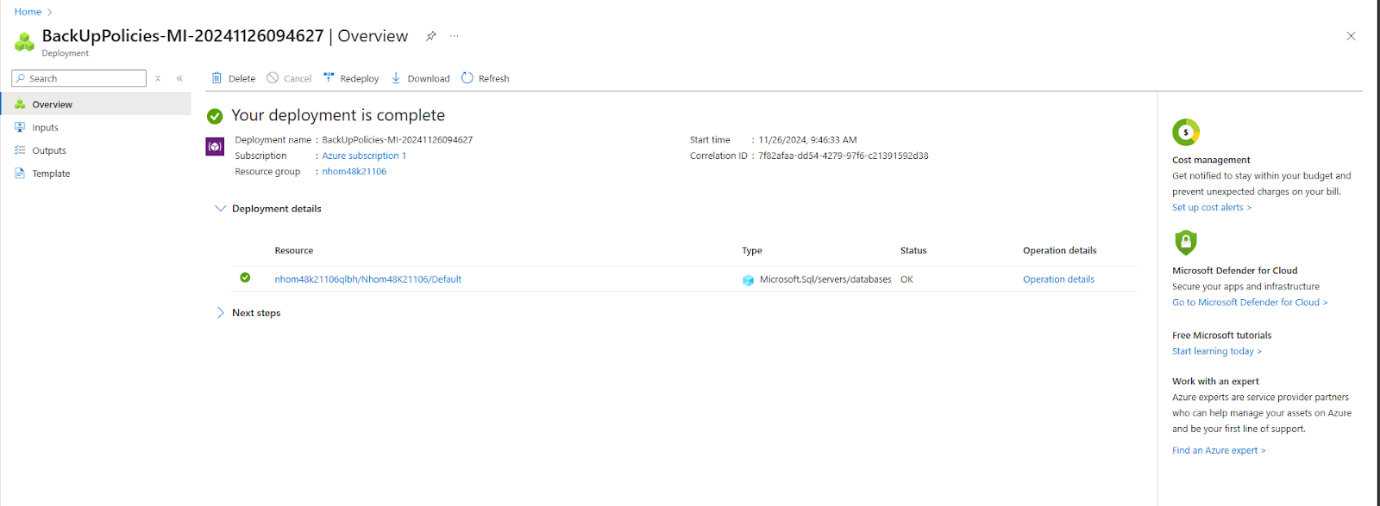


Hình 6‑3. Nhấn vào **Retention policies** và nhấn tích vào **Nhom48K21106**

Bước 4: Sau đó chọn biểu tượng ******Configure policies**  để điều chỉnh thời gian backups

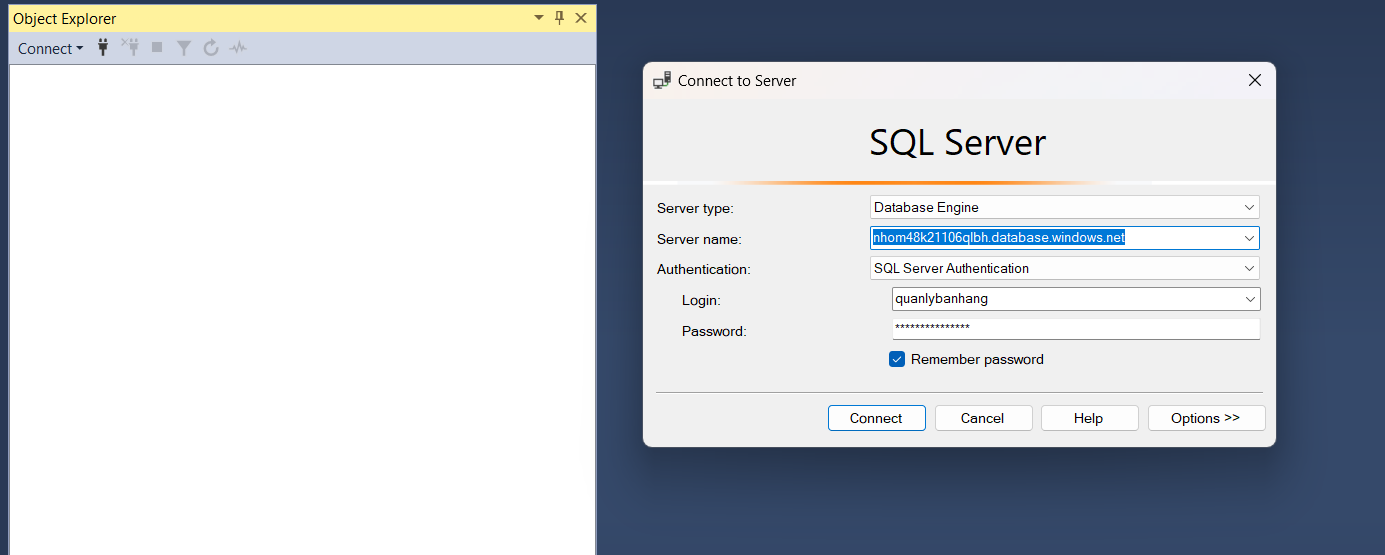
Hình 6‑4. Chọn **Configure policies**  để điều chỉnh thời gian backups

Bước 5: Hoàn thành



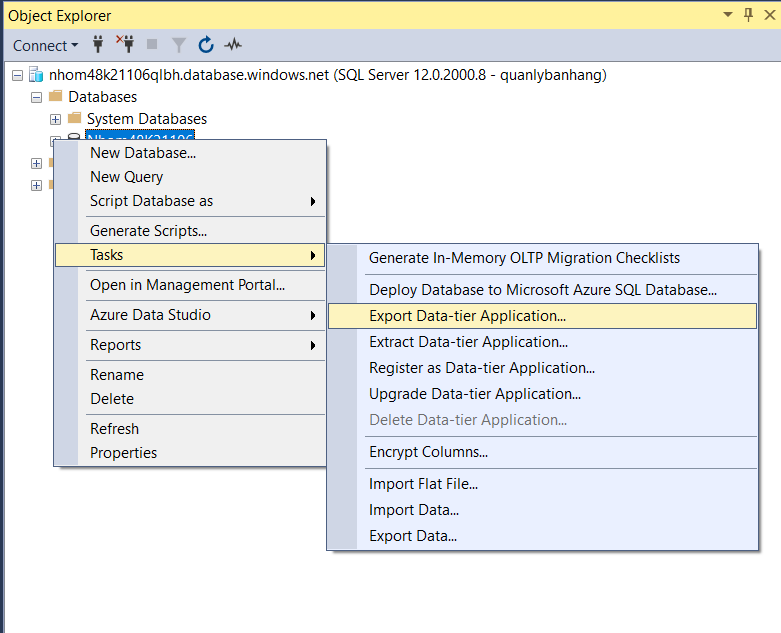
Hình 6‑5. Hoàn thành Back Up trên Microsoft Azure

Bước 6: Truy cập vào ứng dụng **Microsoft SQL Server** → nhấn **connect**

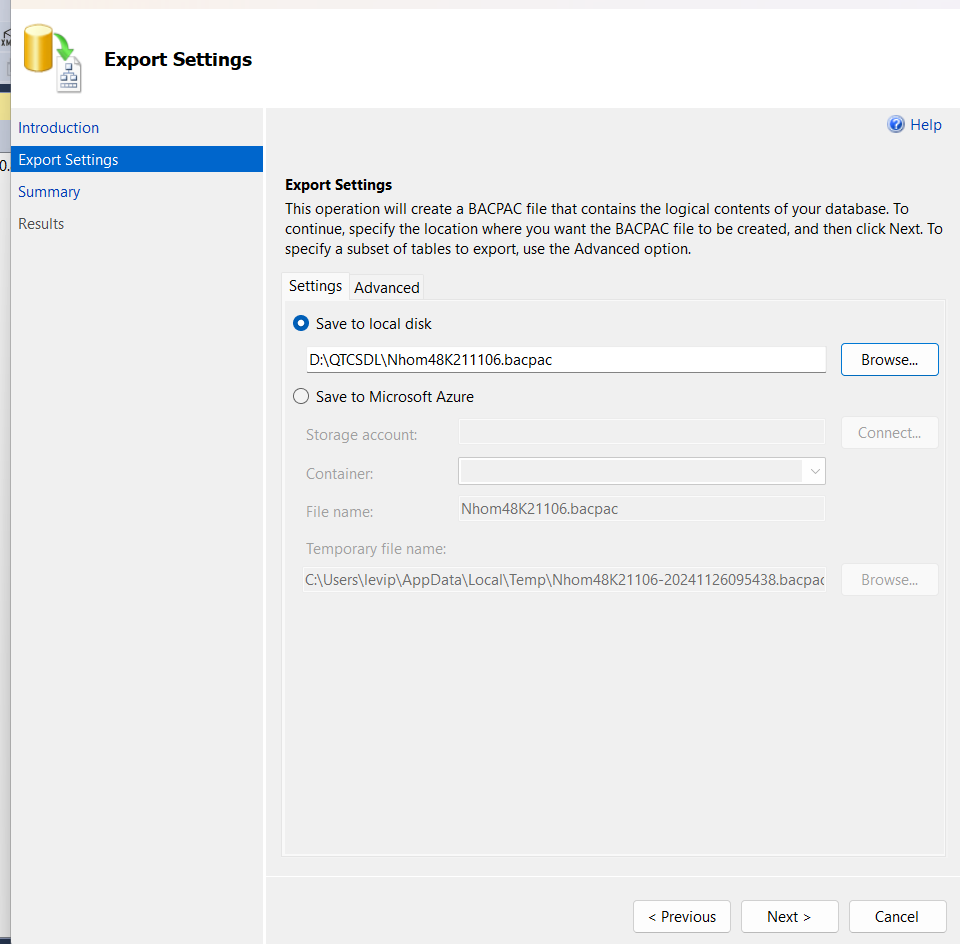


Hình 6‑6. Truy cập vào ứng dụng **Microsoft SQL Server**

Bước 7: Chọn  **Databases** → chọn **Nhom48K21106** cần BackUps → Chọn **Tasks** → Chọn **Export Data-tier Application**

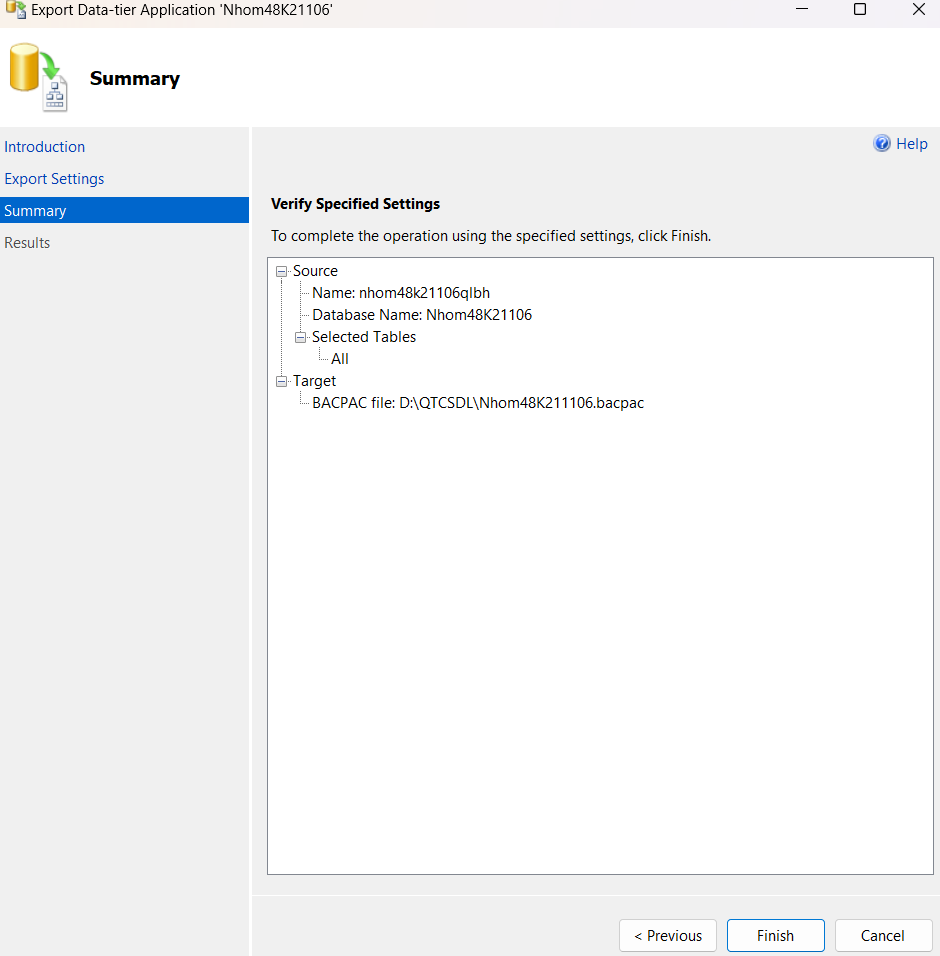


Hình 6‑7. Chọn **Export Data-tier Application**

Bước 8: Chọn vị trí cần lưu → chọn **Next**

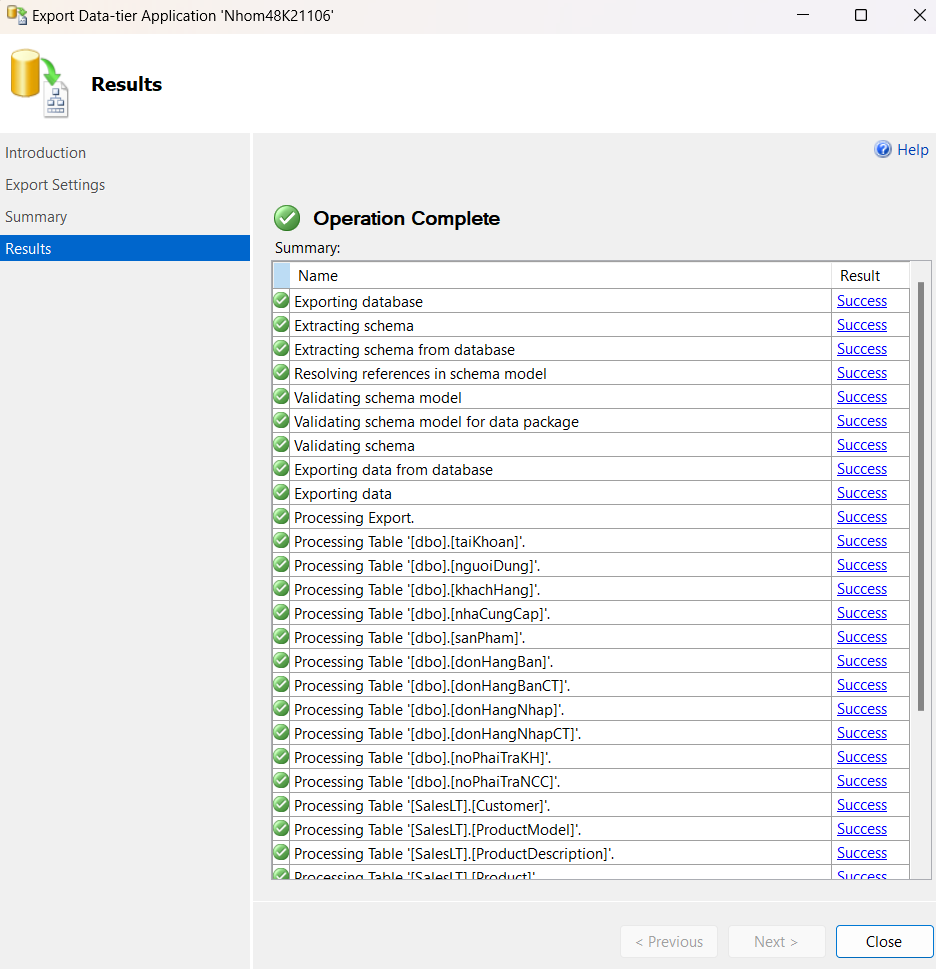
Hình 6‑8. Chọn vị trí cần lưu

Bước 9: Chọn **Finish**



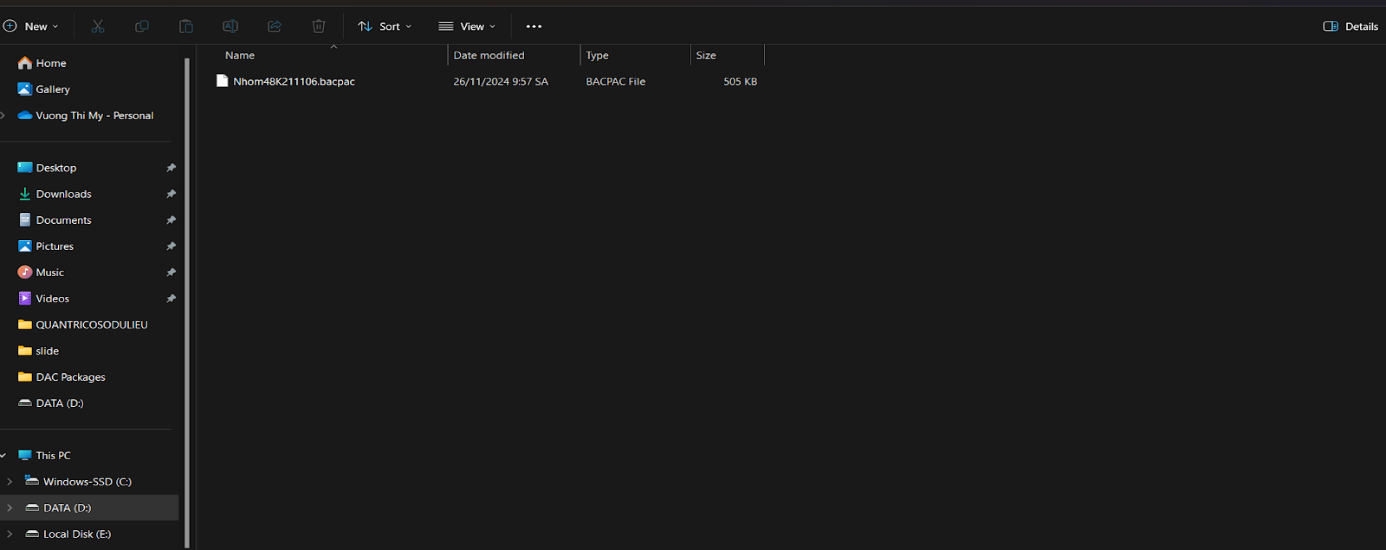
Hình 6‑9. Chọn Finish

Bước 10: Hoàn thành Backups



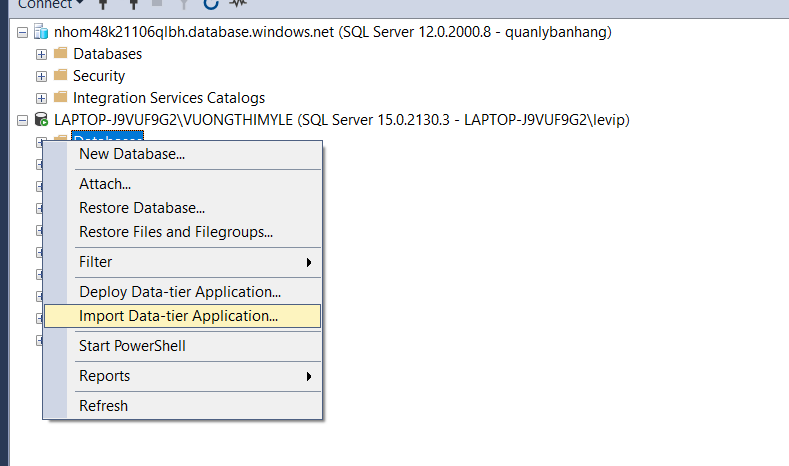
Hình 6‑10. Hoàn thành Back up

Bước 11: Kiểm tra thư mục đã có file bacpac hay chưa



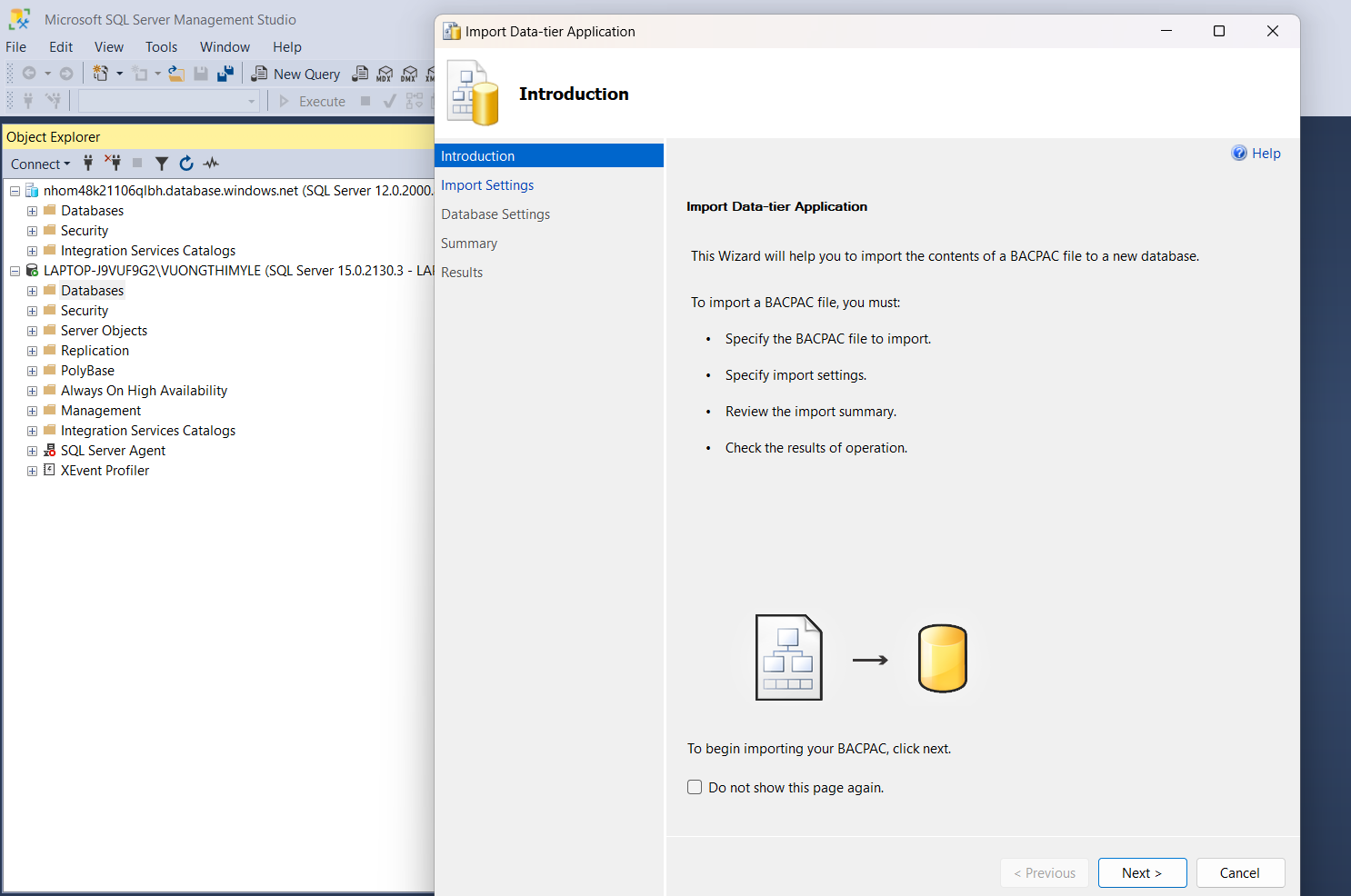
Hình 6‑11. Kiểm tra thư mục đã có file bacpac hay chưa

Bước 12: Restore using Bacpac file in local. Chọn Database → chọn Import Data-tier Application..



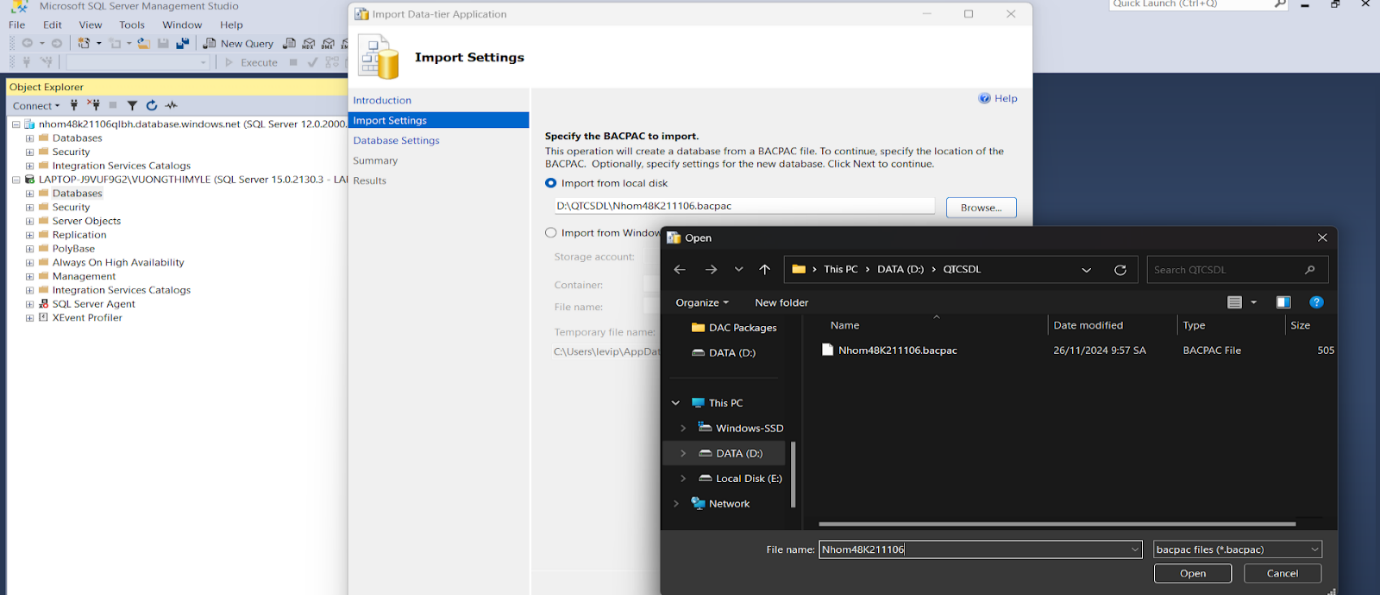
Hình 6‑12. Restore using Bacpac file in local

Bước 13: Nhấn Next



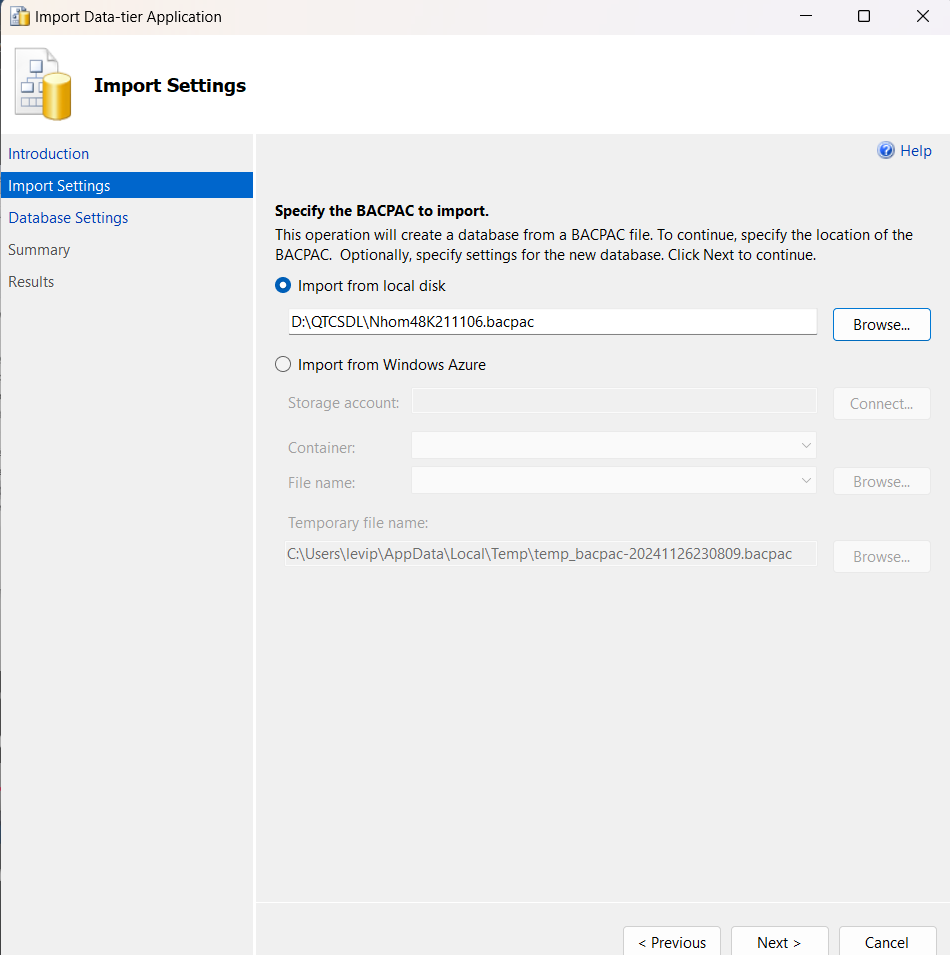
Hình 6‑13 Nhấn Next

Bước 14: Chọn file backpack đã lưu ở bước 10



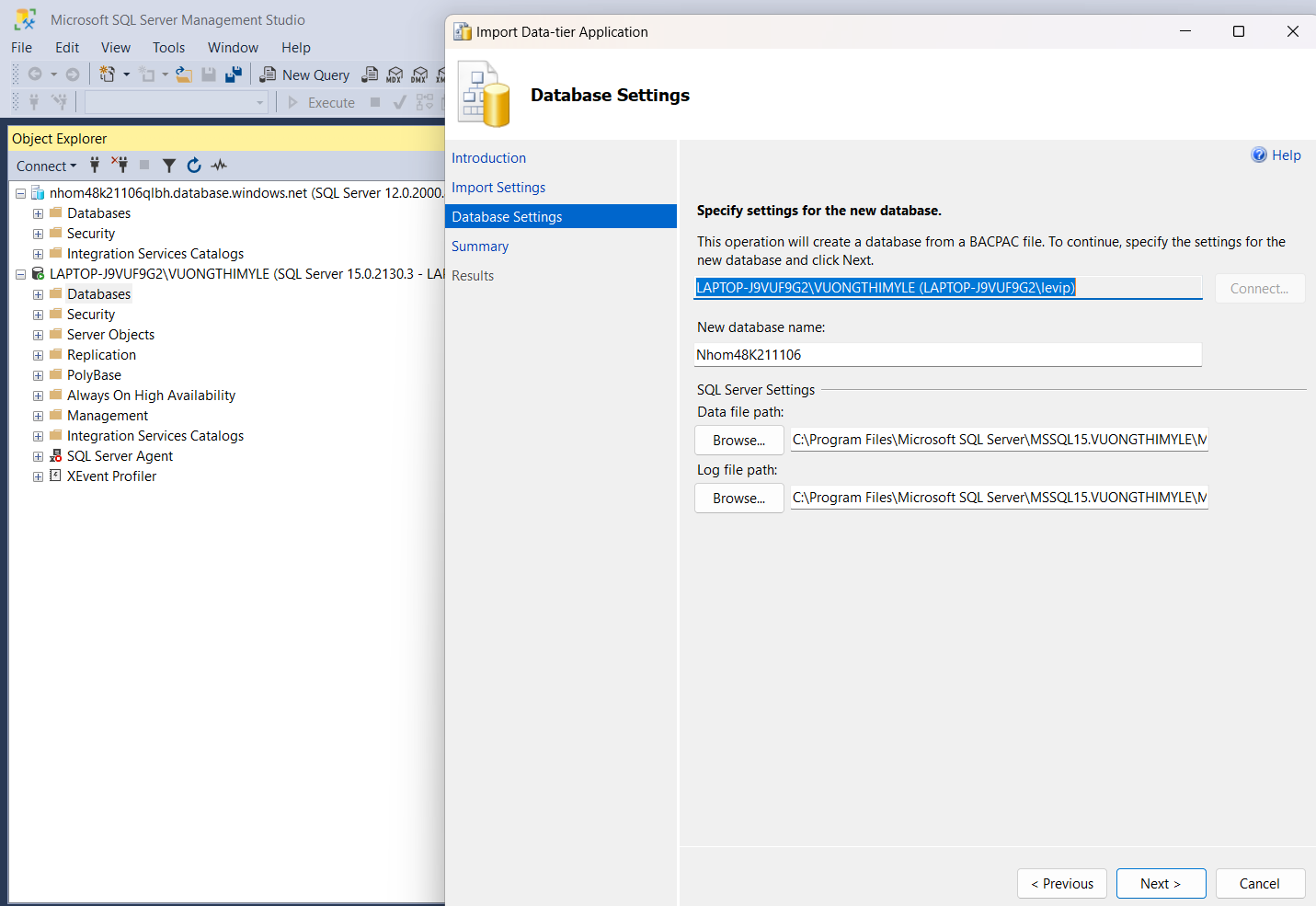
Hình 6‑14. Chọn file backpack đã lưu ở bước 10

Bước 15: Chọn Next

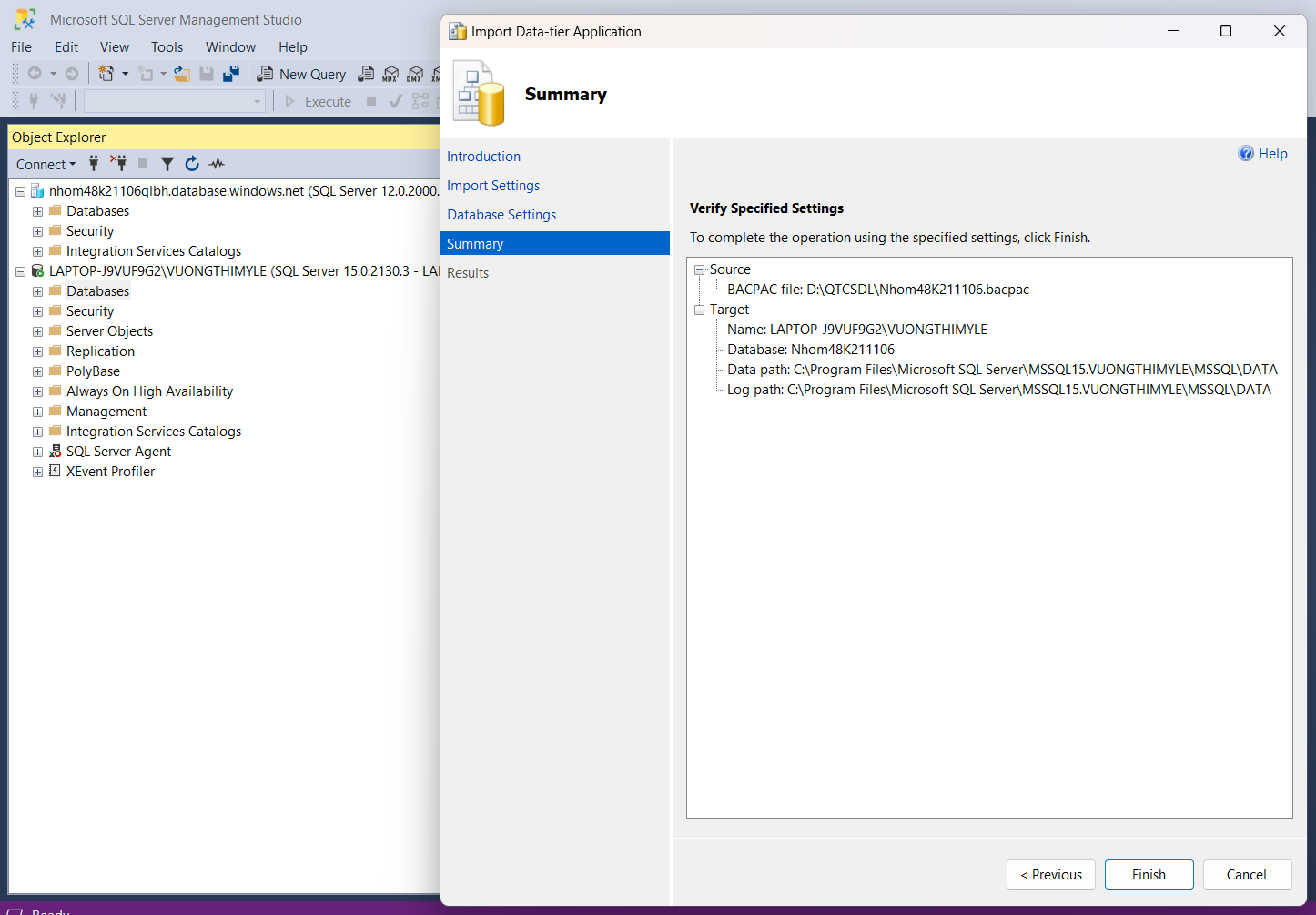


Hình 6‑15. Chọn Next

Bước 16: Chọn Next tiếp

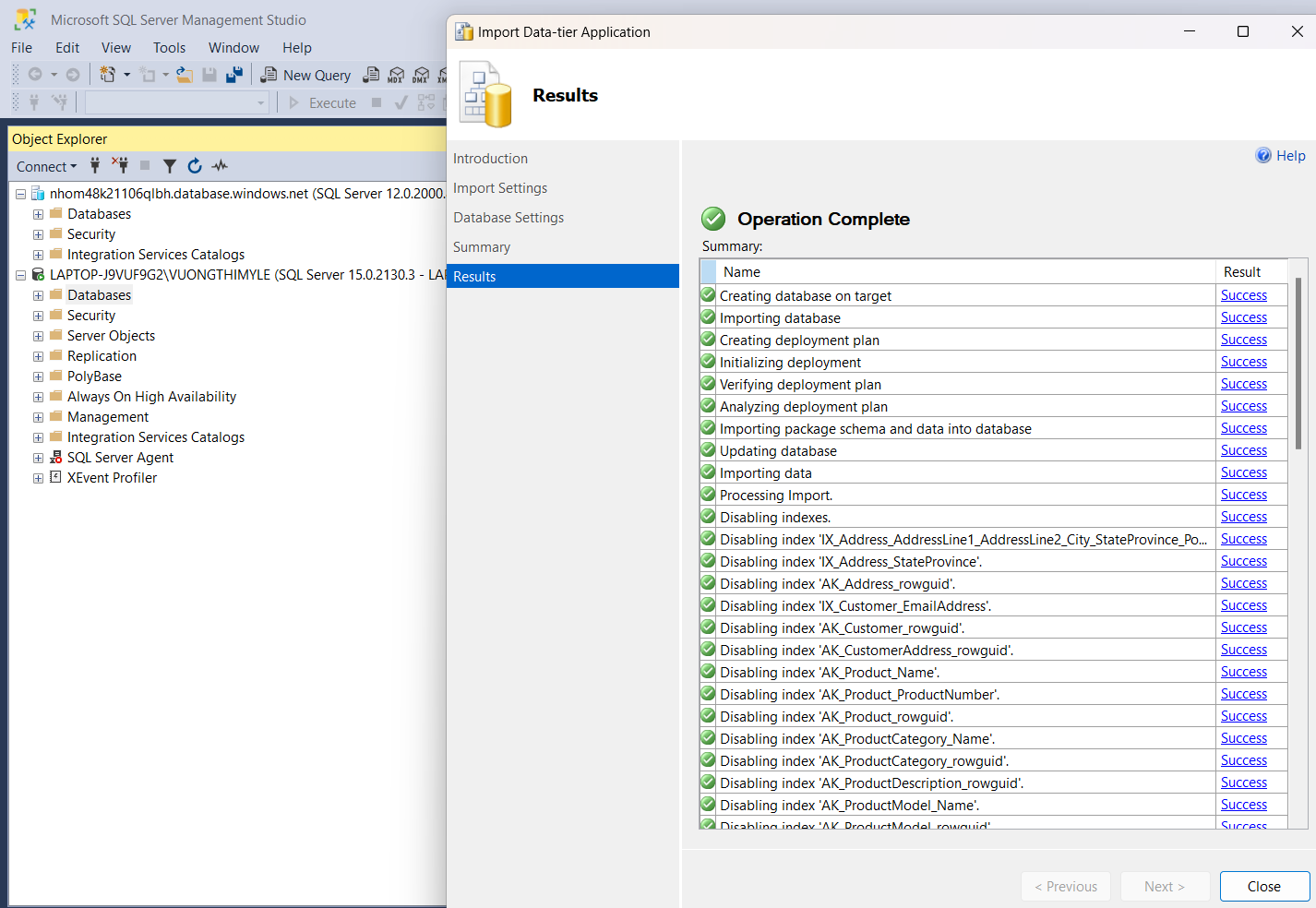


Hình 6‑16. Chọn Next tiếp

Bước 17: Chọn Finish

Hình 6‑17. Chọn Finish

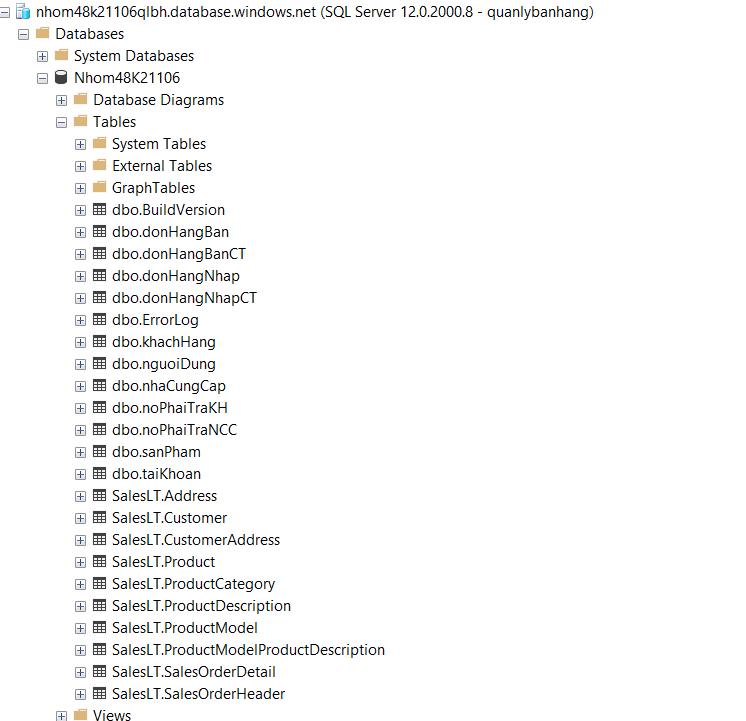
Bước 18: Khôi phục (restore) cơ sở dữ liệu trên local hoàn thành



Hình 6‑18. Khôi phục (restore) cơ sở dữ liệu trên local hoàn thành

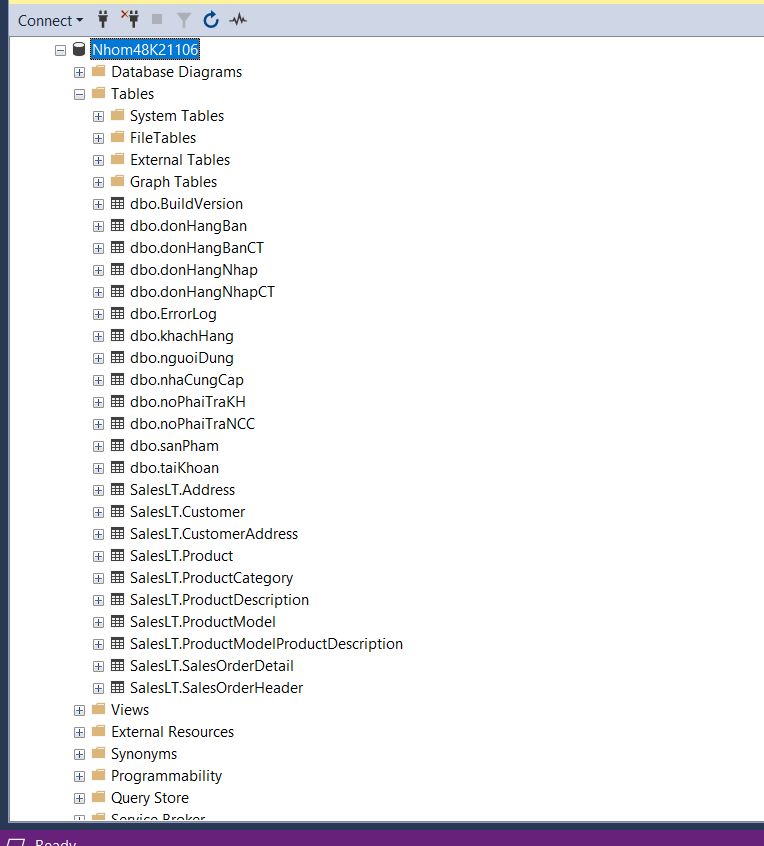
Bước 19: So sánh danh sách bảng:

Máy chủ cơ sở dữ liệu Azures



Hình 6‑19. Danh sách các bảng trên máy chủ CSDL Azure

Bảng trên local



Hình 6‑20. Danh sách các bảng trên SQL Local

# Giả sử lượng dữ liệu rất lớn, không đủ lưu trữ trong một ổ đĩa. Hãy đưa ra phương án giải quyết TỐT NHẤT cho vấn đề đã đặt ra và thực hiện trên CSDL đã xây dựng. (Nêu lý do lựa chọn giải pháp và thực hiện triển khai giải pháp cho hệ thống)

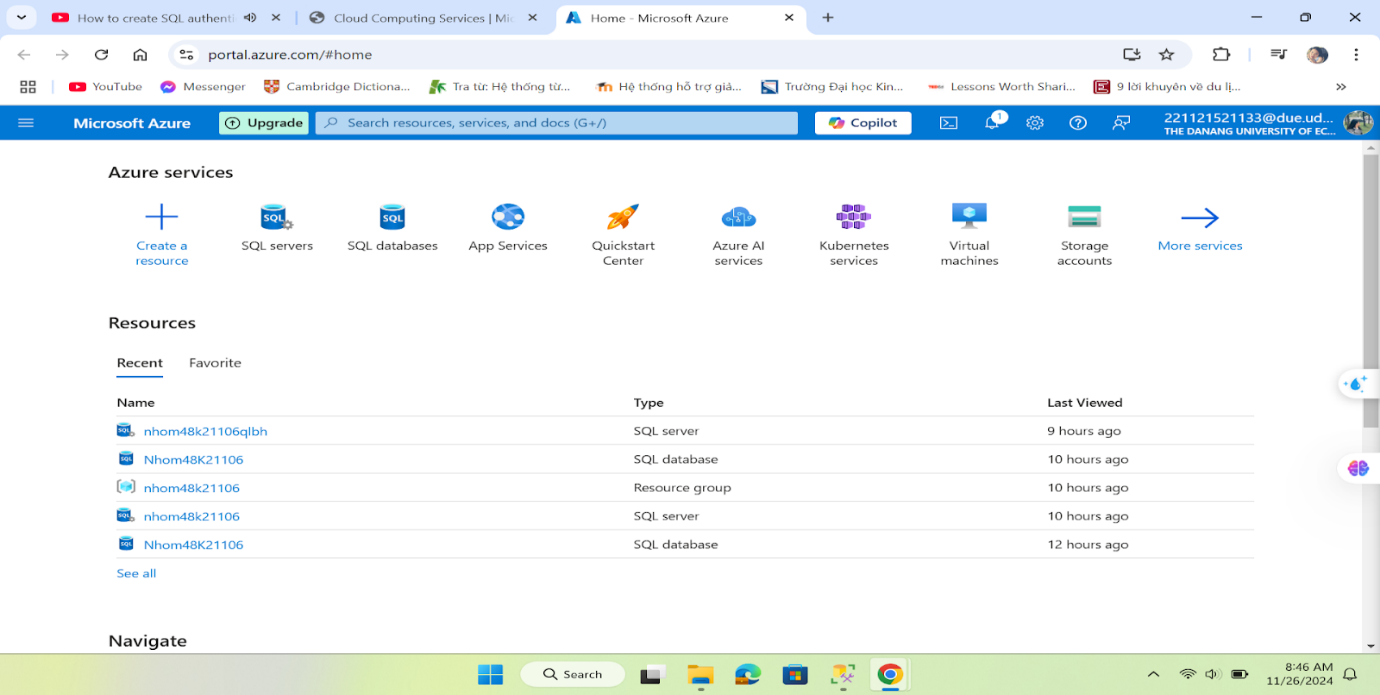
Để giải quyết vấn đề dữ liệu quá lớn, không đủ để lưu trữ trong một ổ đĩa, nhóm chọn phương pháp **Lưu trữ đám mây**

**Lý do:** Lưu trữ đám mây là giải pháp tối ưu khi dữ liệu quá lớn, vượt khả năng lưu trữ của một ổ đĩa nhờ khả năng mở rộng linh hoạt, truy cập mọi lúc mọi nơi và chi phí thấp hơn so với đầu tư cơ sở hạ tầng vật lý. Dữ liệu trên đám mây được sao lưu an toàn tại nhiều địa điểm, tích hợp sẵn các cơ chế bảo mật và hỗ trợ phân tích dữ liệu lớn (Big Data) để khai thác hiệu quả. Ngoài ra, đám mây giúp tối ưu hóa hiệu suất, giảm lãng phí tài nguyên, đồng thời phù hợp với nhu cầu lưu trữ hiện đại, thân thiện với môi trường.

**Dịch vụ lưu trữ đám mây**: Microsoft Azure

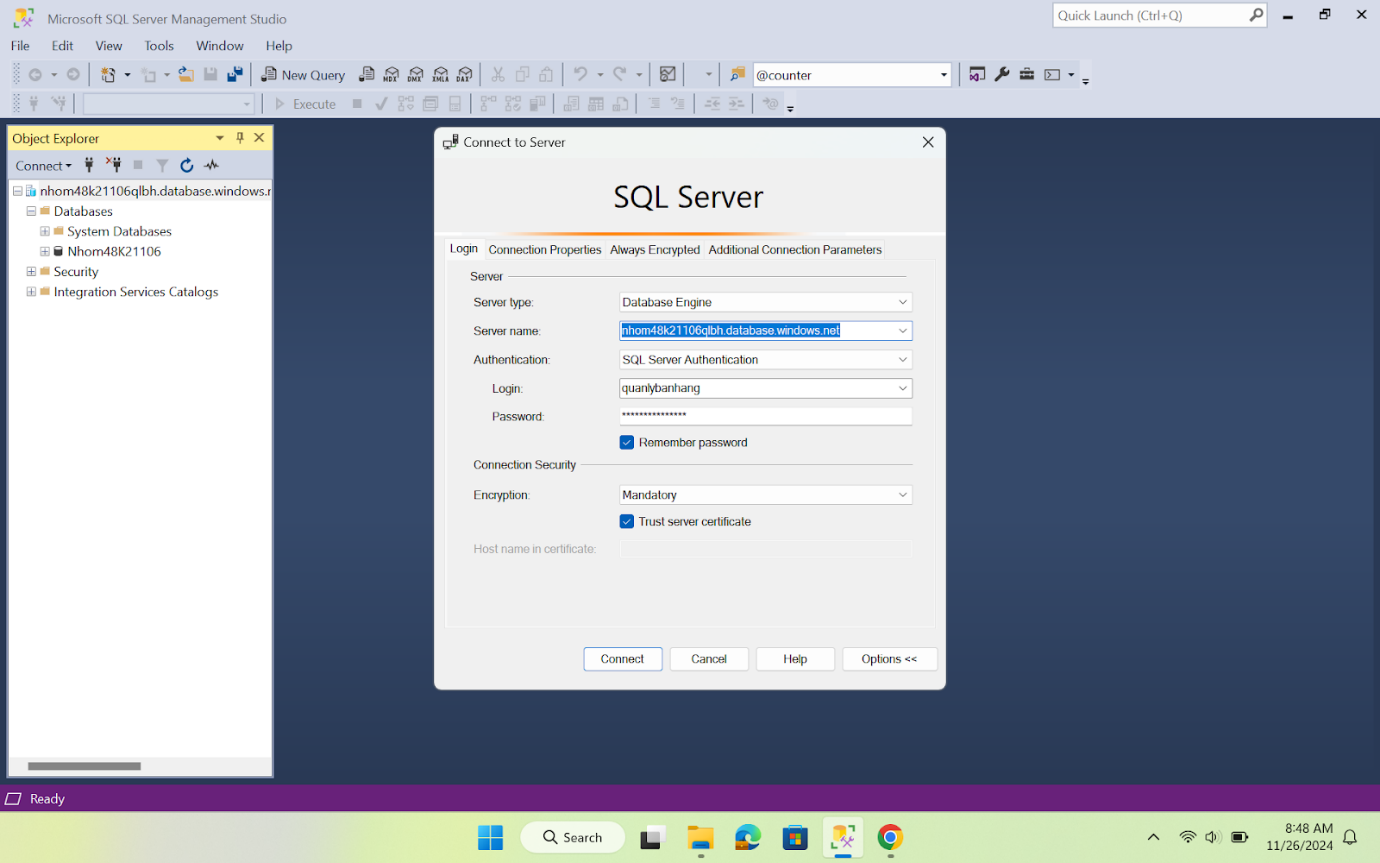
**Cách thực hiện:**

**Bước 1:** Tạo server và các database trong Microsoft Azure



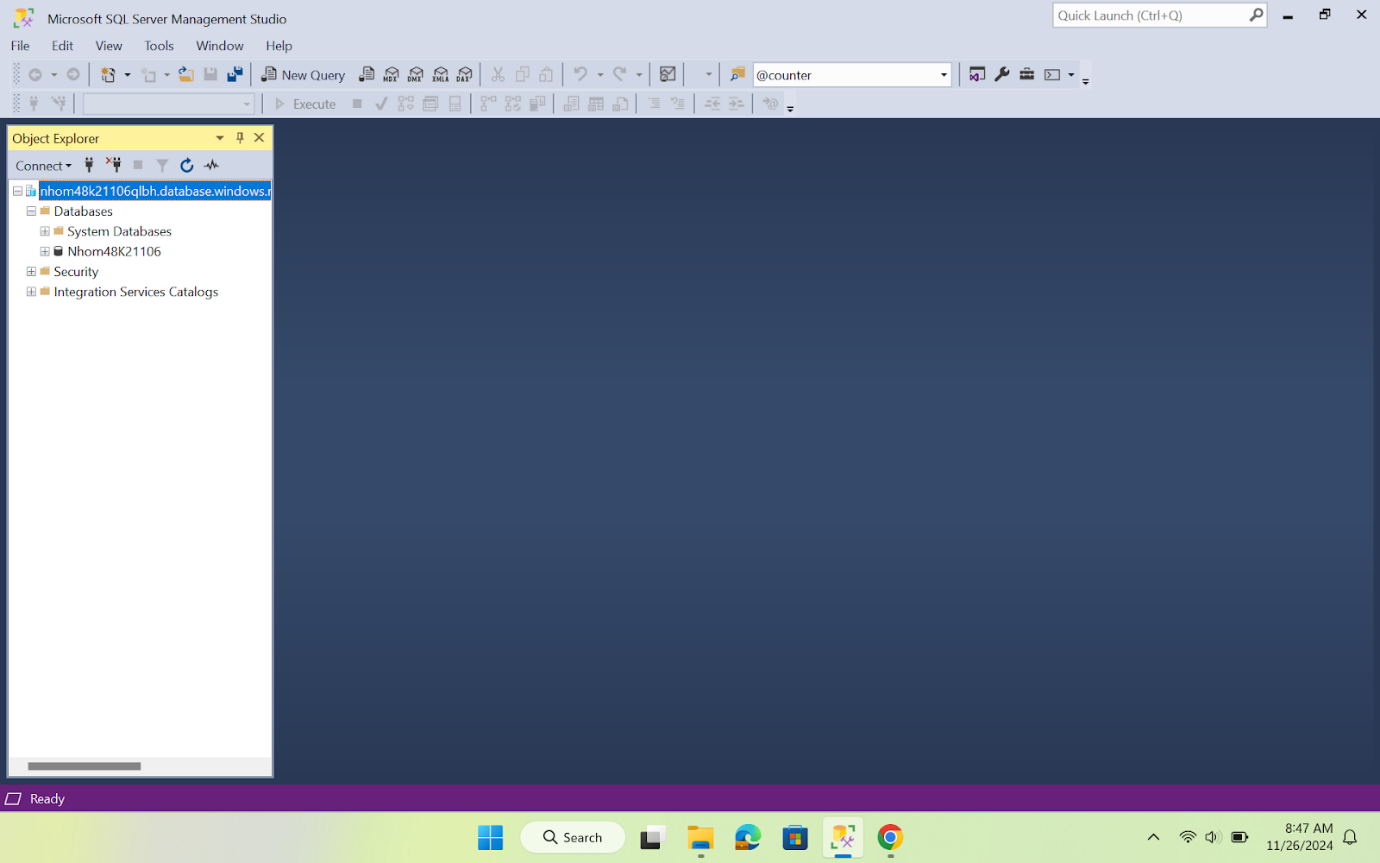
Hình 7‑1. Tạo server và các database trong Microsoft Azure

**Bước 2:** Lấy dữ liệu từ trên Microsoft Azure  bằng cách connect trong SQL server với server name: nhom48k21106qlbh.database.windows.net.



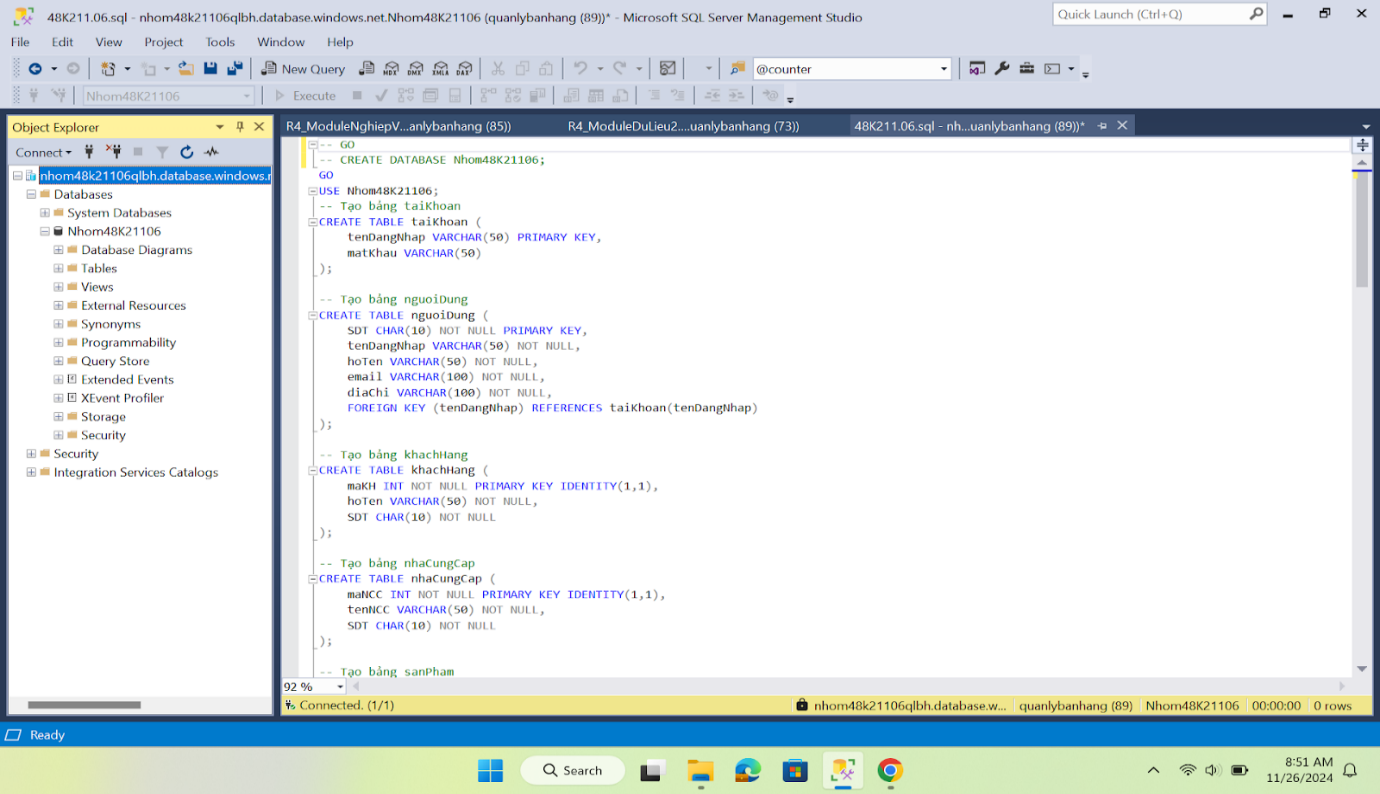
Hình 7‑2. Connect trong SQL server với server name: nhom48k21106qlbh.database.windows.net.

**Bước 3**: Database vừa tạo trên Microsoft Azure đã hiển thị trong SQL Server



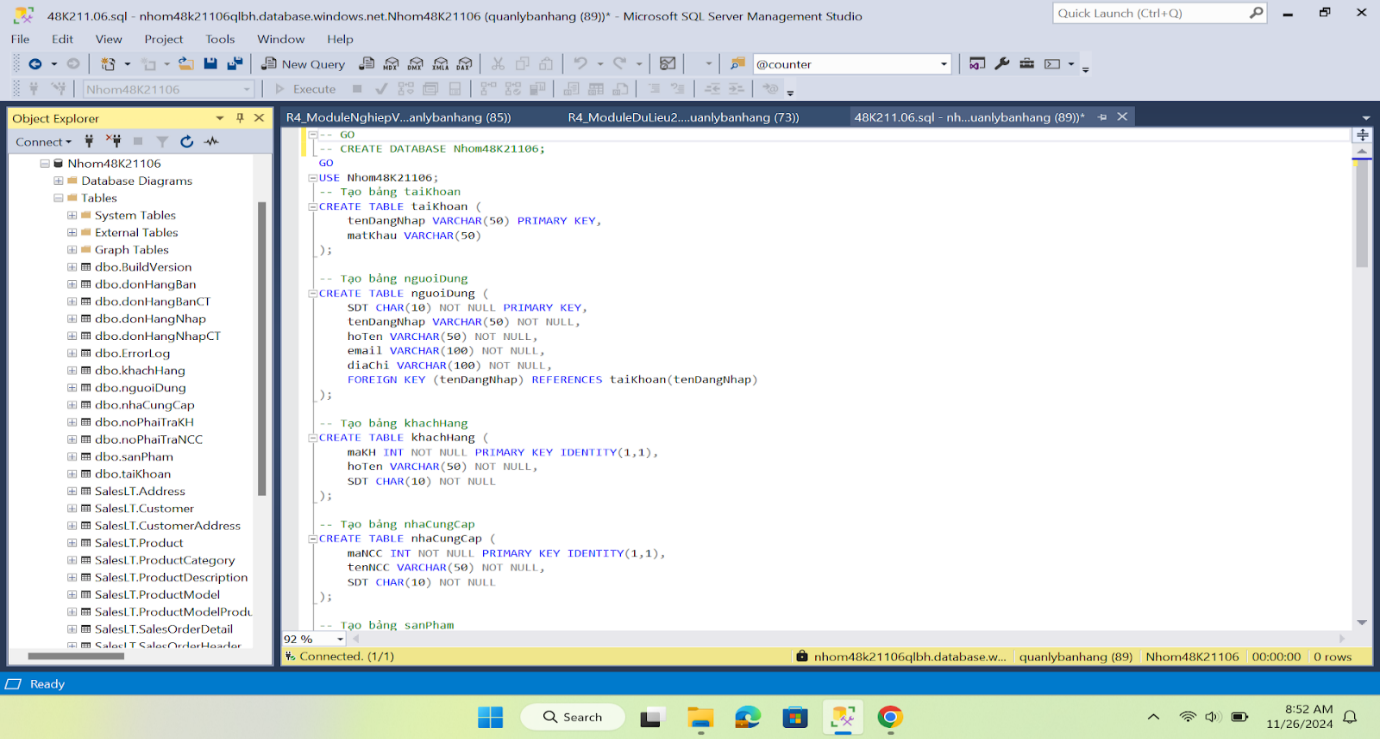
Hình 7‑3. Database vừa tạo trên Microsoft Azure đã hiển thị trong SQL Server

**Bước 4:** Tạo các bảng và các module trong Database “Nhom48K21106”



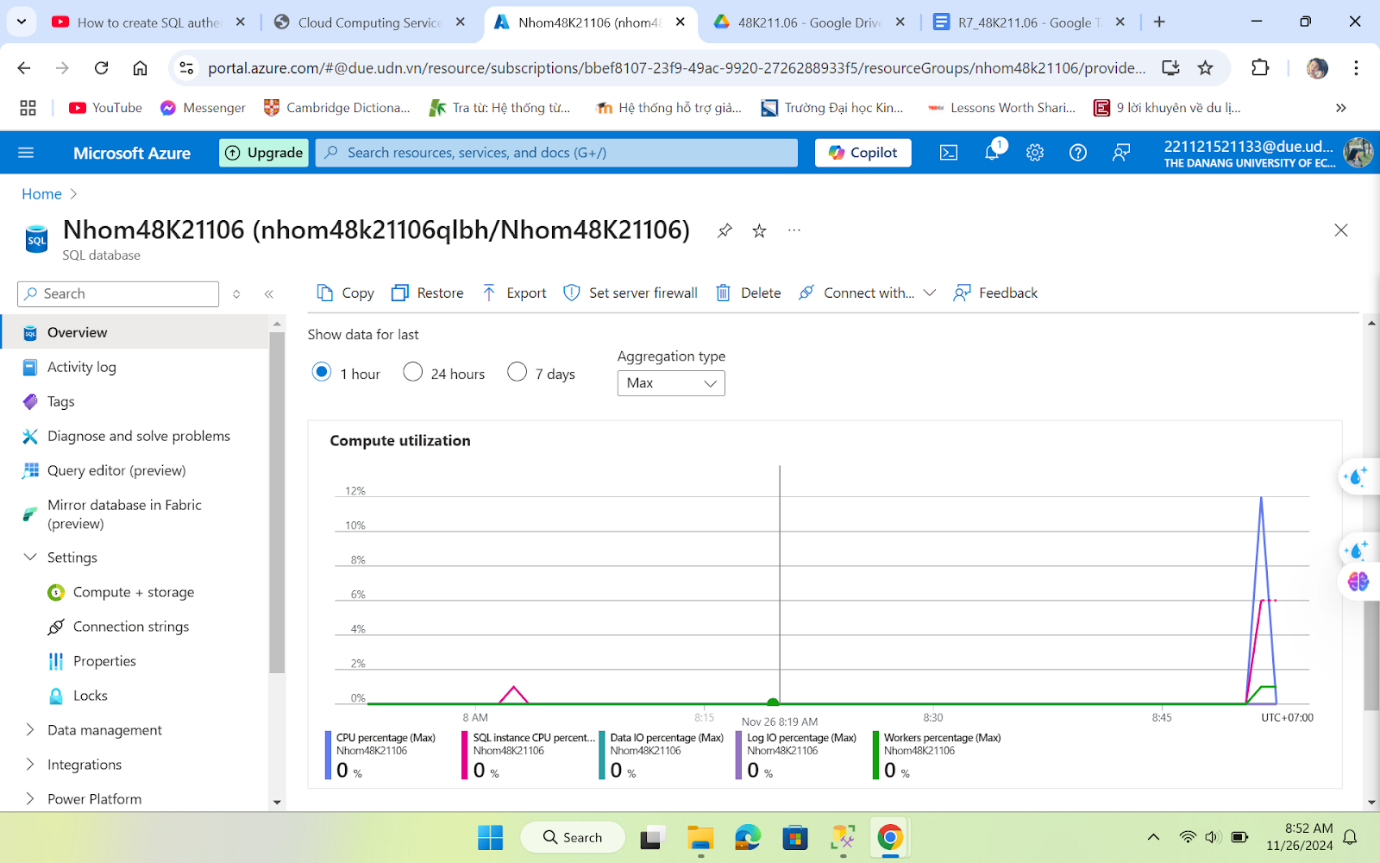
Hình 7‑4. Tạo các bảng và các module trong Database “Nhom48K21106”

**Bước 5**: Các bảng lần lượt hiển thị trong SSMS.



Hình 7‑5. Hình ảnh danh sách các bảng hiện ra ở cột Object Explorer trong SSMS

**Bước 6:** Kết quả sau khi chạy các database trên SSMS sẽ hiển thị trên MS Azure.



Hình 7‑6. Kết quả sau khi chạy các database trên SSMS sẽ hiển thị trên MS Azure

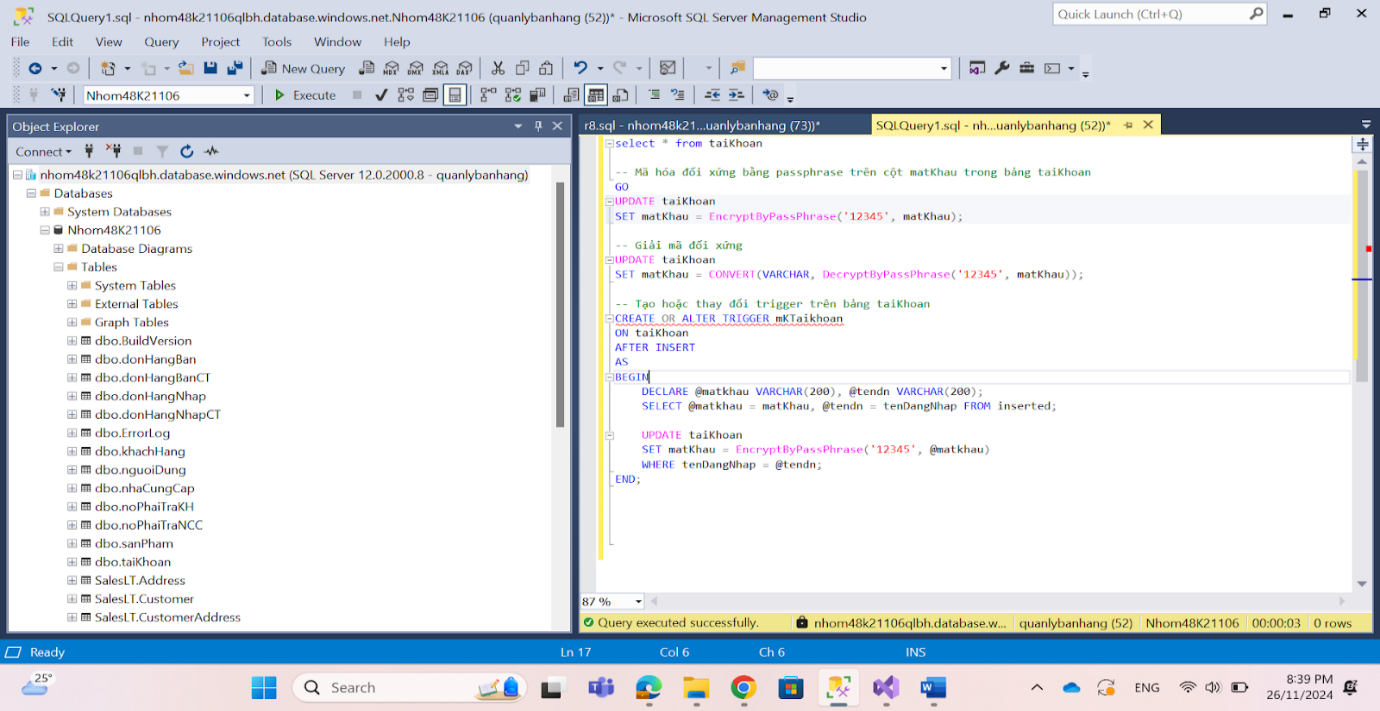
# Đề ra và thực hiện phương án để hạn chế tấn công SQL Injection

**SQL Injection** là một kỹ thuật tấn công nguy hiểm mà kẻ xấu lợi dụng lỗ hổng trong việc kiểm tra và xử lý dữ liệu đầu vào của ứng dụng web, cùng với những thông báo lỗi từ hệ quản trị cơ sở dữ liệu, để chèn vào các câu lệnh SQL bất hợp pháp. Bằng cách này, kẻ tấn công có thể thay đổi logic của các câu truy vấn SQL, từ đó thực hiện các hành động như đọc dữ liệu nhạy cảm, xóa thông tin quan trọng, chèn thêm dữ liệu giả, hoặc thậm chí thay đổi cấu trúc cơ sở dữ liệu. Điều này không chỉ làm lộ dữ liệu của hệ thống mà còn làm giảm độ tin cậy và khả năng bảo mật của ứng dụng.

Một trong những phương pháp phổ biến mà kẻ tấn công sử dụng là dựa vào các điều kiện luôn đúng, chẳng hạn như "1=1", hoặc các biểu thức so sánh " = " luôn đúng, để vượt qua cơ chế xác thực hoặc kiểm tra quyền truy cập. Khi cơ sở dữ liệu không được bảo vệ đúng cách, các truy vấn bị chèn lệnh độc hại có thể gây ra hậu quả nghiêm trọng, bao gồm việc chiếm quyền kiểm soát toàn bộ hệ thống hoặc làm tổn thất dữ liệu quan trọng của tổ chức.

Để giảm nguy cơ bị tấn công SQL Injection, nhóm đề xuất mã hóa dữ liệu đầu vào trước khi thực thi câu lệnh SQL, giúp ngăn chặn mã độc.

**Triển khai phương án mã hoá dữ liệu đầu vào**



Hình 8‑1. Triển khai phương án mã hoá dữ liệu đầu vào