TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO MÔN HỌC QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**HỆ THỐNG: QUẢN LÝ PHÒNG TRỌ**

Lớp: 48K21.1

Nhóm: 11

GVBM: Cao Thị Nhâm

Thành viên nhóm:

Vũ Thị Thu Hằng

Đỗ Thị Huyền Na

Phan Minh Quân

Nguyễn Bích Thy

Lê Thị Kim Yến

Đà Nẵng, 11/2024

MỤC LỤC

[CHƯƠNG 1. THIẾT KẾ CƠ SỞ DỮ LIỆU (R1, R2) 1](#_Toc183769212)

[1.1. Chuyển sơ đồ lớp thành bảng 1](#_Toc183769213)

[1.2. Chuẩn hoá các bảng 1](#_Toc183769214)

[1.3. Sơ đồ quan hệ 2](#_Toc183769215)

[1.4. Thiết kế chi tiết các bảng 2](#_Toc183769216)

[1.4.1. Bảng CHUTRO 2](#_Toc183769217)

[1.4.2. Bảng TAIKHOAN\_NT 3](#_Toc183769218)

[1.4.3. Bảng NGUOITHUETRO 3](#_Toc183769219)

[1.4.4. Bảng PHONG 3](#_Toc183769220)

[1.4.5. Bảng DICHVU 4](#_Toc183769221)

[1.4.6. Bảng HOPDONG 4](#_Toc183769222)

[1.4.7. Bảng HOADON 5](#_Toc183769223)

[1.4.8. Bảng HOADONCHITIET 5](#_Toc183769224)

[CHƯƠNG 2. XÂY DỰNG CƠ SỞ DỮ LIỆU (R3) 5](#_Toc183769225)

[CHƯƠNG 3. XÂY DỰNG MODULE (R4) 8](#_Toc183769226)

[3.1. Module tạo dữ liệu 8](#_Toc183769227)

[3.2. Module xử lí dữ liệu 19](#_Toc183769228)

[3.2.1. Tính thành tiền bảng HOADONCHITIET 19](#_Toc183769229)

[3.2.2. Tính tổng cộng bảng HOADON 20](#_Toc183769230)

[3.2.3. Tính thành tiền bảng HOADONCHITIET khi thêm mới, cập nhật dữ liệu trong bảng HOADONCHITIET 21](#_Toc183769231)

[3.2.4. Tính thành tiền bảng HOADONCHITIET khi thêm mới, cập nhật dữ liệu bảng DICHVU 23](#_Toc183769232)

[3.2.5. Tính tổng cộng bảng HOADON khi thêm mới, cập nhật dữ liệu trong bảng HOADONCHITIET 24](#_Toc183769233)

[3.2.6. Tính tổng cộng bảng HOADON khi cập nhật dữ liệu trong bảng PHONG 26](#_Toc183769234)

[3.2.7. Khi xóa dữ liệu trong bảng NGUOITHUETRO, cập nhật Ten\_NT là ‘không thuê’ thay vì xóa 27](#_Toc183769235)

[3.2.8. Kiểm tra thông tin người thuê trọ đã tồn tại trong hệ thống hay chưa 28](#_Toc183769236)

[3.2.9. Kiểm tra người thuê trọ đang sử dụng loại phòng nào 30](#_Toc183769237)

[3.2.10. Kiểm tra hợp đồng cho thuê của người thuê trọ 31](#_Toc183769238)

[3.2.11. Trả về mã hóa đơn mới. 32](#_Toc183769239)

[3.2.12. Trả về mã hợp đồng mới. 34](#_Toc183769240)

[3.2.13. Thêm mới một hợp đồng 35](#_Toc183769241)

[3.2.14. Thêm mới một hóa đơn 39](#_Toc183769242)

[3.2.15. Thêm mới một hóa đơn chi tiết 43](#_Toc183769243)

[CHƯƠNG 4. XÂY DỰNG CƠ CHẾ BẢO MẬT (R5) 47](#_Toc183769244)

[4.1. Cơ chế bảo mật 47](#_Toc183769245)

[4.2. Bảng phân quyền 47](#_Toc183769246)

[4.3. Thực hiện triển khai trên SQL Server 47](#_Toc183769247)

[4.3.1. Đối với chủ trọ 47](#_Toc183769248)

[4.3.2. Đối với người thuê trọ 50](#_Toc183769249)

[CHƯƠNG 5. XÂY DỰNG CƠ CHẾ BACKUP DỮ LIỆU TỰ ĐỘNG (R6) 54](#_Toc183769250)

[5.1. Backup dữ liệu 54](#_Toc183769251)

[5.2. Restore dữ liệu 59](#_Toc183769252)

[CHƯƠNG 6. LƯU TRỮ DỮ LIỆU (R7) 62](#_Toc183769253)

[6.1. Lý do lựa chọn 62](#_Toc183769254)

[6.2. Thực hiện triển khai trên SQL Server 62](#_Toc183769255)

[CHƯƠNG 7. PHƯƠNG ÁN HẠN CHẾ TẤN CÔNG SQL INJECTION 67](#_Toc183769256)

[7.1. Đề ra phương án để hạn chế tấn công SQL Injection 67](#_Toc183769257)

[7.2. Thực hiện phương án mã hóa mật khẩu và thiết lập thủ tục đăng nhập ngăn chặn SQL Injection 68](#_Toc183769258)

[7.2.1. Đối với bảng CHUTRO ta sẽ mã hóa cột MatKhau\_CT 68](#_Toc183769259)

[7.2.2. Đối với bảng TAIKHOAN\_NT ta sẽ mã hóa cột MatKhau\_NT 70](#_Toc183769260)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Tạo User ChuTro1 thành công 48](#_Toc183769261)

[Hình 2: Tạo Role ChuTro thành công 48](#_Toc183769262)

[Hình 3: Gán User ChuTro1 với Role ChuTro thành công 49](#_Toc183769263)

[Hình 4: Đăng nhập user ChuTro1 49](#_Toc183769264)

[Hình 5: Đăng nhập thành công user ChuTro1 50](#_Toc183769265)

[Hình 6: Tạo User NguoiThue1 thành công 51](#_Toc183769266)

[Hình 7: Tạo Role NguoiThue thành công 51](#_Toc183769267)

[Hình 8: Gán User ChuTro1 với Role ChuTro thành công 52](#_Toc183769268)

[Hình 9: Đăng nhập User NguoiThue1 52](#_Toc183769269)

[Hình 10: Đăng nhập thành công User NguoiThue1 53](#_Toc183769270)

[Hình 11: Chọn SQL Server 54](#_Toc183769271)

[Hình 12: Trình duyệt SQL Server dbnhom11 54](#_Toc183769272)

[Hình 13: Chọn Backups 55](#_Toc183769273)

[Hình 14: Thiết lập thời gian 55](#_Toc183769274)

[Hình 15: Triển khai Backup thành công 56](#_Toc183769275)

[Hình 16: Backup thành công 56](#_Toc183769276)

[Hình 17: Liên kết SQL Server với database vừa tạo trên Azure 57](#_Toc183769277)

[Hình 18: Chọn Export Data tier Application 57](#_Toc183769278)

[Hình 19: Chọn thư mục để lưu trữ file Backup 58](#_Toc183769279)

[Hình 20: Tạo file Backup thành công 58](#_Toc183769280)

[Hình 21: Xem nơi lưu trữ Backup 59](#_Toc183769281)

[Hình 22: Chọn Import Data -tier Application 59](#_Toc183769282)

[Hình 23: Chọn file Backup 60](#_Toc183769283)

[Hình 24: Cấu hình các thông số 60](#_Toc183769284)

[Hình 25: Restore thành công 61](#_Toc183769285)

[Hình 26: Xem Database vừa Restore 61](#_Toc183769286)

[Hình 27: Tạo SQL database trên Microsoft Azure 63](#_Toc183769287)

[Hình 28: Tạo máy chủ SQL mới 63](#_Toc183769288)

[Hình 29: Thiết lập các cấu hình cho SQL Database 64](#_Toc183769289)

[Hình 30: Thiết lập các cấu hình cho SQL Database 64](#_Toc183769290)

[Hình 31: Tạo thàng công SQL Database 65](#_Toc183769291)

[Hình 32: Màn hình Database 65](#_Toc183769292)

[Hình 33: Liên kết SQL Server với Database vừa tạo trên Microsoft Azure 66](#_Toc183769293)

[Hình 34: Liên kết SQL Server thành công 66](#_Toc183769294)

[Hình 35: Mã hóa cột MatKhau\_CT 68](#_Toc183769295)

[Hình 36: Trigger mã hóa cột MatKhau\_CT khi thêm mới dữ liệu vào bảng CHUTRO 69](#_Toc183769296)

[Hình 37: Thủ tục kiểm tra đăng nhập cho tài khoản chủ trọ 69](#_Toc183769297)

[Hình 38: Mã hóa cột MatKhau\_NT 70](#_Toc183769298)

[Hình 39: Trigger mã hóa cột MatKhau\_NT khi thêm mới dữ liệu vào bảng TAIKHOAN\_NT 70](#_Toc183769299)

[Hình 40: Thủ tục kiểm tra đăng nhập cho tài khoản người thuê trọ 71](#_Toc183769300)

# THIẾT KẾ CƠ SỞ DỮ LIỆU (R1, R2)

## Chuyển sơ đồ lớp thành bảng

* CHUTRO(**CCCD\_CT**, Ten\_CT, SDT\_CT, DiaChi\_CT, TenDN\_CT, MatKhau\_CT)
* TAIKHOAN\_NT(**TenDN\_NT**, MatKhau\_NT)
* NGUOITHUETRO(**CCCD\_NT**, Ten\_NT, SDT\_NT, DiaChi\_NT, Email, TenDN\_NT)
* PHONG(**SoPhong**, LoaiPhong, GiaPhong, DienTichPhong)
* DICHVU(**MaDV**, TenDV, DonVi, DonGia)
* HOPDONG(**MaHopDong**, CCCD\_NT, SoPhong, NgayThueTro, NgayHetHan)
* HOADON(**MaHoaDon**, CCCD\_NT, SoPhong, NgayThanhToan, TongCong)
* HOADONCHITIET**(MaHoaDon, MaDV**, SoDung, ThanhTien)

## Chuẩn hoá các bảng

Chuẩn hóa dữ liệu:

* CHUTRO(**CCCD\_CT**, Ten\_CT, SDT\_CT, DiaChi\_CT, TenDN\_CT, MatKhau\_CT)
* TAIKHOAN\_NT(**TenDN\_NT**, MatKhau\_NT)
* NGUOITHUETRO(**CCCD\_NT**, Ten\_NT, SDT\_NT, DiaChi\_NT, Email, TenDN\_NT)
* PHONG(**SoPhong**, LoaiPhong, GiaPhong, DienTichPhong)
* DICHVU(**MaDV**, TenDV, DonVi, DonGia)
* HOPDONG(**MaHopDong**, CCCD\_NT, SoPhong, NgayThueTro, NgayHetHan)
* HOADON(**MaHoaDon**, CCCD\_NT, SoPhong, NgayThanhToan, TongCong)
* HOADONCHITIET**(MaHoaDon, MaDV**, SoDung, ThanhTien)

## Sơ đồ quan hệ

## Thiết kế chi tiết các bảng

### Bảng CHUTRO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | CCCD\_CT | char(12) | not null | PK | Khóa chính của bảng |
| 2 | Ten\_CT | nvarchar (50) | not null |  | Họ tên của chủ trọ |
| 3 | SDT\_CT | varchar (15) | not null |  | Số điện thoại của chủ trọ |
| 4 | DiaChi\_CT | nvarchar (150) | not null |  | Địa chỉ của chủ trọ |
| 5 | TenDN\_CT | varchar(50) | not null |  | Tên đăng nhập của chủ trọ |
| 6 | MatKhau\_CT | varchar (8) | not null |  | Mật khẩu của chủ trọ |

### Bảng TAIKHOAN\_NT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | TenDN\_NT | varchar(50) | not null | PK | Khóa chính của bảng |
| 2 | MatKhau\_NT | varchar (8) | not null |  | Mật khẩu |

### Bảng NGUOITHUETRO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | CCCD\_NT | char(12) | not null | PK | Khóa chính của bảng |
| 2 | Ten\_NT | nvarchar (50) | not null |  | Họ tên của người thuê trọ |
| 3 | SDT\_NT | varchar (15) | not null |  | Số điện thoại của người thuê trọ |
| 4 | DiaChi\_NT | nvarchar (150) | not null |  | Địa chỉ của người thuê trọ |
| 5 | Email | varchar(100) |  |  | Email của người thuê trọ |
| 6 | TenDN\_NT | varchar(50) | not null | FK | Khoá ngoại liên kết với bảng TAIKHOAN\_NT |

### Bảng PHONG

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | SoPhong | char(3) | not null | PK | Khóa chính của bảng |
| 2 | LoaiPhong | char(1) | not null |  | Mô tả loại phòng |
| 3 | GiaPhong | decimal(10, 2) | not null |  | Giá phòng |
| 4 | DienTichPhong | decimal(5, 2) | not null |  | Diện tích của phòng |

### Bảng DICHVU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | MaDV | char(4) | not null | PK | Khóa chính của bảng |
| 2 | TenDV | nvarchar (50) | not null |  | Tên dịch vụ |
| 3 | DonVi | nvarchar (20) |  |  | Đơn vị tính của dịch vụ |
| 4 | DonGia | decimal(10, 2) | not null |  | Đơn giá của dịch vụ |

### Bảng HOPDONG

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | MaHopDong | char(5) | not null | PK | Khóa chính của bảng |
| 2 | CCCD\_NT | char(12) | not null | FK | Khoá ngoại liên kết với bảng NGUOITHUETRO |
| 3 | SoPhong | char(3) | notnull | FK | Khoá ngoại liên kết với bảng PHONG |
| 4 | NgayThueTro | date | not null |  | Ngày vào thuê trọ |
| 5 | NgayHetHan | date | not null |  | Ngày hết hạn hợp đồng |

### Bảng HOADON

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | MaHoaDon | char(6) | not null | PK | Khóa chính của bảng |
| 2 | CCCD\_NT | char(12) | not null | FK | Khoá ngoại liên kết với bảng NGUOITHUETRO |
| 3 | SoPhong | char(3) | notnull | FK | Khoá ngoại liên kết với bảng PHONG |
| 4 | NgayThanhToan | date | not null |  | Ngày thanh toán tiền |
| 5 | TongCong | decimal(10, 2) |  |  | Tổng tiền thanh toán |

### Bảng HOADONCHITIET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| 1 | MaHoaDon | char(6) | not null | PK | Khóa chính của bảng |
| 2 | MaDV | char(4) | not null | PK | Khóa chính của bảng |
| 3 | SoDung | int | not null |  | Số dùng dịch vụ |
| 4 | ThanhTien | decimal(10, 2) |  |  | Thành tiền |

# XÂY DỰNG CƠ SỞ DỮ LIỆU (R3)

create database ThueTro

use ThueTro

create table CHUTRO

(

CCCD\_CT char(12) not null primary key,

Ten\_CT nvarchar(50) not null,

SDT\_CT varchar(15) not null,

DiaChi\_CT nvarchar(150) not null,

TenDN\_CT varchar(50) not null,

MatKhau\_CT varchar(8) not null

)

create table TAIKHOAN\_NT

(

TenDN\_NT varchar(50) not null primary key,

MatKhau\_NT varchar(8) not null

)

create table NGUOITHUETRO

(

CCCD\_NT char(12) not null primary key,

Ten\_NT nvarchar(50) not null,

SDT\_NT varchar(15) not null,

DiaChi\_NT nvarchar(150) not null,

Email varchar(100),

TenDN\_NT varchar(50) not null,

foreign key (TenDN\_NT) references TAIKHOAN\_NT(TenDN\_NT),

)

create table PHONG

(

SoPhong char(3) not null primary key,

LoaiPhong char(1) not null,

GiaPhong decimal(10,2) not null,

DienTichPhong decimal(5,2) not null

)

create table DICHVU

(

MaDV char(4) not null primary key,

TenDV nvarchar(50) not null,

DonVi nvarchar(20),

DonGia decimal(10,2) not null

)

create table HOPDONG

(

MaHopDong char(5) not null primary key,

CCCD\_NT char(12) not null,

SoPhong char(3) not null,

NgayThueTro date not null,

NgayHetHan date not null,

foreign key (CCCD\_NT) references NGUOITHUETRO(CCCD\_NT),

foreign key (SoPhong) references PHONG(SoPhong)

)

create table HOADON

(

MaHoaDon char(6) not null primary key,

CCCD\_NT char(12) not null,

SoPhong char(3) not null,

NgayThanhToan date not null,

TongCong decimal(10,2),

foreign key (CCCD\_NT) references NGUOITHUETRO(CCCD\_NT),

foreign key (SoPhong) references PHONG(SoPhong)

)

create table HOADONCHITIET

(

MaHoaDon char(6) not null,

MaDV char(4) not null,

SoDung int not null,

ThanhTien decimal(10,2),

primary key(MaHoaDon, MaDV),

foreign key (MaHoaDon) references HOADON(MaHoaDon),

foreign key (MaDV) references DICHVU(MaDV)

)

# XÂY DỰNG MODULE (R4)

## Module tạo dữ liệu

--CHỦ TRỌ

go

create or alter proc chenChuTro

as

begin

declare @dem int = 1,

@CCCD\_CT char(12),

@Ten\_CT nvarchar(50),

@SDT\_CT varchar(15),

@DiaChi\_CT nvarchar(150),

@TenDN\_CT varchar(50),

@MatKhau\_CT varchar(8)

while @dem <= 1000

begin

set @CCCD\_CT = RIGHT('00111000000' + CAST(@dem as varchar), 12)

set @Ten\_CT = case (@dem % 10) + 1

when 1 then N'Nguyễn Văn Anh'

when 2 then N'Trần Lê Bảo'

when 3 then N'Lê Thị Cương'

when 4 then N'Phạm Thị Duyên'

when 5 then N'Đặng Anh Huy'

when 6 then N'Phan Phương Quỳnh'

when 7 then N'Hồ Phạm An'

when 8 then N'Đinh Hoàng'

when 9 then N'Nguyễn Thị Tuyết Trinh'

when 10 then N'Bùi Anh Tuấn'

end

set @SDT\_CT = case (@dem % 5) + 1

when 1 then '098' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 2 then '070' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 3 then '035' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 4 then '093' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 5 then '079' + RIGHT('000000' + CAST(@dem as varchar), 7)

end

set @DiaChi\_CT = case (@dem % 10) + 1

when 1 then N'Số 1, Đường A, Đà Nẵng'

when 2 then N'Số 2, Đường B, Đà Nẵng'

when 3 then N'Số 3, Đường C, Đà Nẵng'

when 4 then N'Số 4, Đường D, Đà Nẵng'

when 5 then N'Số 5, Đường E, Đà Nẵng'

when 6 then N'Số 6, Đường F, Đà Nẵng'

when 7 then N'Số 7, Đường G, Đà Nẵng'

when 8 then N'Số 8, Đường H, Đà Nẵng'

when 9 then N'Số 9, Đường I, Đà Nẵng'

when 10 then N'Số 10, Đường J, Đà Nẵng'

end

set @TenDN\_CT = 'chutro' + CAST(@dem as varchar)

set @MatKhau\_CT = 'pass' + CAST(@dem as varchar)

insert into CHUTRO (CCCD\_CT, Ten\_CT, SDT\_CT, DiaChi\_CT, TenDN\_CT, MatKhau\_CT )

values (@CCCD\_CT, @Ten\_CT, @SDT\_CT, @DiaChi\_CT,@TenDN\_CT, @MatKhau\_CT)

set @dem = @dem + 1

end

end

exec chenChuTro

select \* from CHUTRO

--TÀI KHOẢN NGƯỜI THUÊ TRỌ

go

create or alter proc chenTaiKhoan\_NT

as

begin

declare @dem int = 1,

@TenDN\_NT varchar(50),

@MatKhau\_NT varchar(8)

while @dem <= 1000

begin

set @TenDN\_NT = 'nguoithue' + CAST(@dem as varchar)

set @MatKhau\_NT = 'pass' + CAST(@dem as varchar)

insert into TAIKHOAN\_NT (TenDN\_NT, MatKhau\_NT)

values (@TenDN\_NT, @MatKhau\_NT)

set @dem = @dem + 1

end

end

exec chenTaiKhoan\_NT

select \* from TAIKHOAN\_NT

--NGƯỜI THUÊ TRỌ

go

create or alter proc chenNguoiThueTro

as

begin

declare @dem int = 1,

@CCCD\_NT char(12),

@Ten\_NT nvarchar(50),

@SDT\_NT varchar(15),

@DiaChi\_NT nvarchar(150),

@Email varchar(100),

@TenDN\_NT varchar(50)

while @dem <= 1000

begin

set @CCCD\_NT = RIGHT('000000000000' + CAST(@dem as varchar), 12)

set @Ten\_NT = case (@dem % 10) + 1

when 1 then N'Nguyễn Văn An'

when 2 then N'Trần Thị Bình'

when 3 then N'Lê Văn Cường'

when 4 then N'Phạm Thị Dung'

when 5 then N'Đặng Văn Huy'

when 6 then N'Phan Như Quỳnh'

when 7 then N'Hồ Thị Giang'

when 8 then N'Đinh Văn Hoàng'

when 9 then N'Nguyễn Thị Tuyết Trinh'

when 10 then N'Bùi Văn Tuấn'

end

set @SDT\_NT = case (@dem % 5) + 1

when 1 then '098' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 2 then '070' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 3 then '035' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 4 then '093' + RIGHT('000000' + CAST(@dem as varchar), 7)

when 5 then '079' + RIGHT('000000' + CAST(@dem as varchar), 7)

end

set @DiaChi\_NT = case (@dem % 10) + 1

when 1 then N'Số 1, Đường A, TP.HCM'

when 2 then N'Số 2, Đường B, TP.HCM'

when 3 then N'Số 3, Đường C, TP.HCM'

when 4 then N'Số 4, Đường D, Hà Nội'

when 5 then N'Số 5, Đường E, Đà Nẵng'

when 6 then N'Số 6, Đường F, TP.HCM'

when 7 then N'Số 7, Đường G, TP.HCM'

when 8 then N'Số 8, Đường H, TP.HCM'

when 9 then N'Số 9, Đường I, TP.HCM'

when 10 then N'Số 10, Đường J, Huế'

end

set @Email = 'nguoithue' + CAST(@dem as varchar) + '@gmail.com'

set @TenDN\_NT = 'nguoithue' + CAST(@dem as varchar)

insert into NGUOITHUETRO (CCCD\_NT, Ten\_NT, SDT\_NT, DiaChi\_NT, Email, TenDN\_NT)

values (@CCCD\_NT, @Ten\_NT, @SDT\_NT, @DiaChi\_NT, @Email, @TenDN\_NT)

set @dem = @dem + 1

end

end

exec chenNguoiThueTro

select \* from NGUOITHUETRO

--PHÒNG

go

create or alter proc chenPhong

as

begin

declare @dem int = 1,

@SoPhong char(3),

@LoaiPhong nvarchar(20),

@GiaPhong decimal(10,2),

@DienTichPhong decimal(5,2)

while @dem <= 1000

begin

set @SoPhong = RIGHT('000' + CAST(@dem as varchar), 3)

set @LoaiPhong = case (@dem % 2) + 1

when 1 then '1' --Phòng đơn

when 2 then '2' --Phòng đôi

end

set @GiaPhong = case

when @LoaiPhong = '1' then 2000000

when @LoaiPhong = '2' then 2500000

end

set @DienTichPhong = case

when @LoaiPhong = '1' then 14.5

when @LoaiPhong = '2' then 20

end

insert into PHONG (SoPhong, LoaiPhong, GiaPhong, DienTichPhong)

values (@SoPhong, @LoaiPhong, @GiaPhong, @DienTichPhong)

set @dem = @dem + 1

end

end

exec chenPhong

select \* from PHONG

--DỊCH VỤ

go

create or alter proc chenDichVu

as

begin

declare @dem int = 1,

@MaDV char(4),

@TenDV nvarchar(50),

@DonVi nvarchar(20),

@DonGia decimal(10,2)

while @dem <= 1000

begin

set @MaDV = RIGHT('0000' + CAST(@dem as varchar), 4)

set @TenDV = case (@dem % 4) + 1

when 1 then N'Điện'

when 2 then N'Nước'

when 3 then N'Wifi'

when 4 then N'Rác'

end

set @DonVi = case

when @TenDV = N'Điện' then N'kW '

when @TenDV = N'Nước' then N'm3'

when @TenDV = N'Wifi' then N'tháng'

when @TenDV = N'Rác' then N'tháng'

end

set @DonGia = case

when @TenDV = N'Điện' then 4000

when @TenDV = N'Nước' then 7000

when @TenDV = N'Rác' then 30000

when @TenDV = N'Wifi' then 15000

end

insert into DICHVU (MaDV, TenDV, DonVi, DonGia)

values (@MaDV, @TenDV, @DonVi, @DonGia)

set @dem = @dem + 1

end

end

exec chenDichVu

select \* from DICHVU

--HỢP ĐỒNG

go

create or alter proc chenHopDong

as

begin

declare @dem int = 1,

@MaHopDong char(5),

@CCCD\_NT char(12),

@SoPhong char(3),

@NgayThueTro date,

@NgayHetHan date

while @dem <= 1000

begin

set @MaHopDong = RIGHT('00000' + CAST(@dem as varchar), 5)

set @CCCD\_NT = RIGHT('000000000000' + CAST(@dem as varchar), 12)

set @SoPhong = RIGHT('000' + CAST(@dem as varchar), 3)

set @NgayThueTro = DATEADD(DAY, ROUND(RAND() \* 365, 0), '2022-01-01')

set @NgayHetHan = DATEADD(DAY, ROUND(RAND() \* 365, 0), '2024-01-01')

insert into HOPDONG (MaHopDong, CCCD\_NT, SoPhong, NgayThueTro, NgayHetHan)

VALUES (@MaHopDong, @CCCD\_NT, @SoPhong, @NgayThueTro, @NgayHetHan)

set @dem = @dem + 1

end

end

exec chenHopDong

select \* from HOPDONG

-- HÓA ĐƠN

go

create or alter proc chenHoaDon

as

begin

declare @dem int = 1,

@MaHoaDon char(6),

@CCCD\_NT char(12),

@SoPhong char(3),

@NgayThanhToan date

while @dem <= 1000

begin

set @MaHoaDon = RIGHT('000000' + CAST(@dem as varchar), 6)

set @CCCD\_NT = RIGHT('000000000000' + CAST(@dem as varchar), 12)

set @SoPhong = RIGHT('000' + CAST(@dem as varchar), 3)

set @NgayThanhToan = DATEADD(DAY, ROUND(RAND() \* 365,0), '2023-01-01')

insert into HOADON (MaHoaDon, CCCD\_NT, SoPhong, NgayThanhToan)

values (@MaHoaDon, @CCCD\_NT, @SoPhong, @NgayThanhToan)

set @dem = @dem + 1

end

end

exec chenHoaDon

select \* from HOADON

--HÓA ĐƠN CHI TIẾT

go

create or alter proc chenHoaDonChiTiet

as

begin

declare @dem int = 1,

@MaHoaDon char(6),

@MaDV char(4),

@SoDung int

while @dem <= 1000

begin

set @MaHoaDon = RIGHT('000000' + CAST(@dem as varchar), 6)

set @MaDV = RIGHT('0000' + CAST(@dem as varchar), 4)

set @SoDung = ROUND(RAND() \* 50, 0)

insert into HOADONCHITIET (MaHoaDon, MaDV, SoDung)

values (@MaHoaDon, @MaDV, @SoDung)

set @dem = @dem + 1

end

end

exec chenHoaDonChiTiet

select \* from HOADONCHITIET

## Module xử lí dữ liệu

### Tính thành tiền bảng HOADONCHITIET

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thủ tục này tính thành tiền (ThanhTien) của bảng HOADONCHITIET sau khi nhập dữ liệu bảng HOADONCHITIET và bảng DICHVU** | |
| **INPUT** | **N/A** |
| **OUTPUT** | **N/A** |
| **PROCESS** | 1. Cập nhật bảng HOADONCHITIET 2. Tính ThanhTien=SoDung \*DonGia với SoDung của bảng HOADONCHITIET và DonGia của bảng DICHVU 3. In ra thông báo 'Đã tính thành tiền' |
| **CODE** | go  create or alter proc spTinhThanhTien  as  begin  update HOADONCHITIET  set HOADONCHITIET.ThanhTien = HOADONCHITIET.SoDung \* DICHVU.DonGia  from HOADONCHITIET join DICHVU on HOADONCHITIET.MaDV = DICHVU.MaDV  print N'Đã tính thành tiền'  end  exec spTinhThanhTien |

### Tính tổng cộng bảng HOADON

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thủ tục này tính tổng cộng (TongCong) của bảng HOADON sau khi nhập dữ liệu bảng HOADONCHITIET và bảng PHONG** | |
| **INPUT** | **N/A** |
| **OUTPUT** | **N/A** |
| **PROCESS** | 1. Cập nhật bảng HOADON 2. Tính TongCong=SUM(ThanhTien) +GiaPhong với ThanhTien của bảng HOADONCHITIET và GiaPhong của bảng PHONG 3. In ra thông báo 'Đã tính tổng cộng' |
| **CODE** | go  create or alter proc spTinhTongCong  as  begin  update HOADON  set TongCong = (select SUM(HOADONCHITIET.ThanhTien)  from HOADONCHITIET  where HOADONCHITIET.MaHoaDon = HOADON.MaHoaDon) +  (select PHONG.GiaPhong  from PHONG  where PHONG.SoPhong = HOADON.SoPhong)  from HOADON  print N'Đã tính tổng cộng'  end  exec spTinhTongCong |

### Tính thành tiền bảng HOADONCHITIET khi thêm mới, cập nhật dữ liệu trong bảng HOADONCHITIET

|  |  |
| --- | --- |
| **LOẠI: Trigger** | |
| **MÔ TẢ: Trigger này tự động tính thành tiền khi thêm mới, cập nhật dữ liệu trong bảng HOADONCHITIET** | |
| **LOẠI** | after |
| **SỰ KIỆN** | insert, update |
| **KÍ SINH** | HOADONCHITIET |
| **PROCESS** | 1. Cập nhật bảng HOADONCHITIET 2. Tính ThanhTien=SoDung \*DonGia với SoDung của bảng HOADONCHITIET và DonGia của bảng DICHVU 3. Điều kiện MaHoaDon bảng HOADONCHITIET = MaHoaDon bảng inserted |
| **CODE** | go  create or alter trigger tTinhThanhTien  on HOADONCHITIET  after insert, update  as  begin  update HOADONCHITIET  set HOADONCHITIET.ThanhTien = HOADONCHITIET.SoDung \* DICHVU.DonGia  from HOADONCHITIET join DICHVU on HOADONCHITIET.MaDV = DICHVU.MaDV  where HOADONCHITIET.MaHoaDon in (select MaHoaDon from inserted)  end  insert into HOADONCHITIET (MaHoaDon, MaDV, SoDung)  values('000001','0002',10) |

### Tính thành tiền bảng HOADONCHITIET khi thêm mới, cập nhật dữ liệu bảng DICHVU

|  |  |
| --- | --- |
| **LOẠI: Trigger** | |
| **MÔ TẢ: Trigger này tự động tính thành tiền khi thêm mới, cập nhật dữ liệu trong bảng DICHVU** | |
| **LOẠI** | after |
| **SỰ KIỆN** | insert, update |
| **KÍ SINH** | DICHVU |
| **PROCESS** | 1. Cập nhật bảng HOADONCHITIET 2. Tính ThanhTien=SoDung\*DonGia với SoDung thuộc bảng inserted (chứa các bản ghi mới hoặc được cập nhật) với DonGiatừ bảng DICHVU 3. Điều kiện MaDV bảng DICHVU = MaDV bảng inserted |
| **CODE** | go  create or alter trigger tTinhThanhTienKhiCapNhatDonGia  on DICHVU  after insert, update  as  begin  update HOADONCHITIET  set HOADONCHITIET.ThanhTien = HOADONCHITIET.SoDung \* DICHVU.DonGia  from HOADONCHITIET join DICHVU on HOADONCHITIET.MaDV = DICHVU.MaDV  where DICHVU.MaDV in (select MaDV from inserted)  end  update DICHVU  set DonGia=13000  where MaDV='0002' |

### Tính tổng cộng bảng HOADON khi thêm mới, cập nhật dữ liệu trong bảng HOADONCHITIET

|  |  |
| --- | --- |
| **LOẠI: Trigger** | |
| **MÔ TẢ: Trigger đảm bảo rằng mỗi khi có sự thay đổi trong bảng HOADONCHITIET, tổng cộng (TongCong) trong bảng HOADON liên quan sẽ được được tính toán lại** | |
| **LOẠI** | after |
| **SỰ KIỆN** | insert, update |
| **KÍ SINH** | HOADONCHITIET |
| **PROCESS** | 1. Cập nhật bảng HOADON 2. Tính TongCong=SUM(ThanhTien)+GiaPhong với ThanhTien của bảng HOADONCHITIET và GiaPhong của bảng PHONG 3. Điều kiện MaHoaDon bảng HOADON = MaHoaDon bảng inserted |
| **CODE** | go  create or alter trigger tTinhTongCong  on HOADONCHITIET  after insert, update  as  begin  update HOADON  set TongCong = (select SUM(HOADONCHITIET.ThanhTien)  from HOADONCHITIET  where HOADONCHITIET.MaHoaDon = HOADON.MaHoaDon) +  (select PHONG.GiaPhong  from PHONG  where PHONG.SoPhong = HOADON.SoPhong)  from HOADON  where MaHoaDon in (select MaHoaDon from inserted)  end  insert into HOADONCHITIET (MaHoaDon, MaDV, SoDung)  values('000001','0003',20) |

### Tính tổng cộng bảng HOADON khi cập nhật dữ liệu trong bảng PHONG

|  |  |
| --- | --- |
| **LOẠI: Trigger** | |
| **MÔ TẢ: Trigger đảm bảo rằng mỗi khi giá phòng (GiaPhong) được cập nhật, tổng cộng (TongCong) trong bảng HOADON liên quan cũng sẽ tự động được tính toán lại** | |
| **LOẠI** | after |
| **SỰ KIỆN** | update |
| **KÍ SINH** | PHONG |
| **PROCESS** | 1. Cập nhật bảng HOADON 2. Tính TongCong=SUM(ThanhTien)+GiaPhong với ThanhTien của bảng HOADONCHITIET và GiaPhong của bảng PHONG 3. Điều kiện MaHoaDon bảng HOADON = MaHoaDon bảng inserted |
| **CODE** | go  create or alter trigger tTinhTongCongKhiCapNhatGiaPhong  on PHONG  after update  as  begin  update HOADON  set TongCong = (select SUM(HOADONCHITIET.ThanhTien)  from HOADONCHITIET  where HOADONCHITIET.MaHoaDon = HOADON.MaHoaDon) +  (select PHONG.GiaPhong  from PHONG  where PHONG.SoPhong = HOADON.SoPhong)  from HOADON  where MaHoaDon in (select MaHoaDon from inserted)  end  update PHONG  set GiaPhong=1600000  where SoPhong='001' |

### Khi xóa dữ liệu trong bảng NGUOITHUETRO, cập nhật Ten\_NT là ‘không thuê’ thay vì xóa

|  |  |
| --- | --- |
| **LOẠI: Trigger** | |
| **MÔ TẢ: Trigger này được sử dụng để ngăn việc xóa dữ liệu trong bảng NGUOITHUETRO. Thay vì xóa bản ghi, nó cập nhật giá trị của cột Ten\_NT thành 'không thuê'** | |
| **LOẠI** | instead of |
| **SỰ KIỆN** | delete |
| **KÍ SINH** | NGUOITHUETRO |
| **PROCESS** | 1. Cập nhật bảng NGUOITHUETRO 2. Thay đổi giá trị của cột Ten\_NT thành “không thuê”, điều kiện CCCD\_NT bị xóa |
| **CODE** | go  create or alter trigger tXoaNguoiThueTro  on NGUOITHUETRO  instead of delete  as  begin  update NGUOITHUETRO  set Ten\_NT=N'Không thuê'  where CCCD\_NT in (select CCCD\_NT from deleted)  end  select \* from NGUOITHUETRO  delete NGUOITHUETRO where CCCD\_NT='000000000002' |

### Kiểm tra thông tin người thuê trọ đã tồn tại trong hệ thống hay chưa

|  |  |
| --- | --- |
| **LOẠI: Hàm** | |
| **MÔ TẢ: Hàm này được sử dụng kiểm tra thông tin người thuê trọ đã tồn tại trong hệ thống hay chưa nếu biết họ tên và số điện thoại. Tồn tại trả về 1, không tồn tại trả về 0** | |
| **INPUT** | Ten\_NT, SDT\_NT |
| **OUTPUT** | Xác nhận tồn tại (Tồn tại trả về 1, không tồn tại trả về 0) |
| **PROCESS** | 1. Nếu Ten\_NT và SDT\_NT có trong bảng NGUOITHUETRO thì trả về 1 2. Còn lại thì trả về 0 |
| **CODE** | go  create or alter function fKiemTraThongTinKhachHang(@Ten\_NT nvarchar(50), @SDT\_NT varchar(15))  returns char(1)  as  begin  declare @ret char(1)  if @Ten\_NT in (select Ten\_NT from NGUOITHUETRO) and @SDT\_NT in (select SDT\_NT from NGUOITHUETRO)  begin  set @ret=1  end  else  begin  set @ret=0  end  return @ret  end  select dbo.fKiemTraThongTinKhachHang(N'Phạm Thị Dung','0930000003') |

### Kiểm tra người thuê trọ đang sử dụng loại phòng nào

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thủ tục này được sử dụng để kiểm tra thông tin loại phòng mà khách hàng (người thuê trọ) đang thuê, dựa trên số CCCD\_NT của người thuê** | |
| **INPUT** | CCCD\_NT |
| **OUTPUT** | Nhận xét thông tin loại phòng |
| **PROCESS** | 1. Tìm LoaiPhong trong bảng PHONG , điều kiện CCCD\_NT=@CCCD\_NT 🡪@LoaiPhong 2. Nếu @LoaiPhong=1 thì là phòng đơn   Nếu @LoaiPhong=2 thì là phòng đôi |
| **CODE** | go  create or alter proc spKiemTraKhachHang(@CCCD\_NT char(12), @ret nvarchar(20) output)  as  begin  declare @LoaiPhong char(1)  select @LoaiPhong=LoaiPhong  from PHONG join HOPDONG on PHONG.SoPhong=HOPDONG.SoPhong  join NGUOITHUETRO on NGUOITHUETRO.CCCD\_NT=HOPDONG.CCCD\_NT  where NGUOITHUETRO.CCCD\_NT=@CCCD\_NT  set @ret = case when @LoaiPhong=1 then N'Phòng đơn'  when @LoaiPhong=2 then N'Phòng đôi'  end  end  go  declare @ret nvarchar(20)  exec spkiemTraKhachHang '000000000001', @ret output  print @ret |

### Kiểm tra hợp đồng cho thuê của người thuê trọ

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thủ tục này được sử dụng để kiểm tra hợp đồng cho thuê của người thuê trọ đã hết hạn chưa khi biết CCCD\_NT, nếu có thì cập nhật tên người thuê trọ là 'Hết hạn hợp đồng'** | |
| **INPUT** | CCCD\_NT |
| **OUTPUT** | **N/A** |
| **PROCESS** | 1. Tìm NgayHetHan trong bảng HOPDONG, điều kiện CCCD\_NT=@CCCD\_NT 🡪@NgayHetHan 2. Nếu @NgayHetHan < ngày hiện tại thì cập nhật tên người thuê trọ trong bảng NGUOITHUETRO là 'Hết hạn hợp đồng' và đưa ra thông báo ‘Hợp đồng đã hết hạn’   Nếu không thì đưa ra thông báo ‘Hợp đồng vẫn còn hạn’ |
| **CODE** | go  create or alter proc spKiemTraHopDong (@CCCD\_NT char(12))  as  begin  declare @NgayHetHan date  select @NgayHetHan=NgayHetHan  from HOPDONG  where CCCD\_NT=@CCCD\_NT  if @NgayHetHan<getdate()  begin  update NGUOITHUETRO  set Ten\_NT=N'Hết hạn hợp đồng'  where CCCD\_NT=@CCCD\_NT  print N'Hợp đồng đã hết hạn'  end  else  begin  print N'Hợp đồng vẫn còn hạn'  end  end  exec spKiemTraHopDong '000000000001' |

### Trả về mã hóa đơn mới.

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Trả về mã hóa đơn mới bằng cách lấy mã hóa đơn lớn nhất hiện có, sau đó tăng giá trị đó lên 1 và trả về kết quả dưới dạng chuỗi có độ dài cố định.** | |
| **INPUT** | **N/A** |
| **OUTPUT** | Mã hóa đơn mới |
| **PROCESS** | 1. Lấy mã hóa đơn lớn nhất hiện có trong bảng HOADON. 2. Tạo mã hóa đơn mới    1. Chuyển mã hóa đơn lớn nhất từ chuỗi (varchar) sang số nguyên (int) để có thể cộng thêm 1.    2. Tăng mã giao dịch lớn nhất lên 1 để tạo mã mới.    3. Sau đó chuyển kết quả cộng lại về dạng chuỗi varchar    4. Ghép thêm chuỗi '000000' phía trước giá trị số vừa chuyển đổi để đảm bảo mã có độ dài đủ 6 ký tự.    5. Dùng RIGHT() lấy 6 ký tự cuối cùng của chuỗi sau khi đã ghép, đảm bảo mã giao dịch luôn có độ dài 6 ký tự. Nếu mã nhỏ hơn 6 ký tự, phần đầu sẽ được bù bởi các số 0. |
| **CODE** | go  create or alter proc spMaHoaDonMoi(@MaHoaDonMoi char(6) output)  as  begin  declare @MaHoaDonLonNhat varchar(10)  select @MaHoaDonLonNhat = max(MaHoaDon)  from HOADON  set @MaHoaDonMoi=RIGHT('000000' + convert(varchar, convert(int, @MaHoaDonLonNhat)+1),6)  end  go  declare @MaHoaDonMoi varchar(10)  exec spMaHoaDonMoi @MaHoaDonMoi output  print @MaHoaDonMoi |

### Trả về mã hợp đồng mới.

|  |  |
| --- | --- |
| **LOẠI: Hàm** | |
| **MÔ TẢ: Trả về mã hợp đồng tiếp theo bằng cách lấy mã hợp đồng lớn nhất hiện có, sau đó tăng giá trị đó lên 1 và trả về kết quả dưới dạng chuỗi có độ dài cố định.** | |
| **INPUT** | **N/A** |
| **OUTPUT** | Mã hợp đồng mới |
| **PROCESS** | 1. Lấy mã hóa đơn lớn nhất hiện có trong bảng HOPDONG. 2. Tạo mã hợp đồng mới    1. Chuyển mã hợp đồng lớn nhất từ chuỗi (varchar) sang số nguyên (int) để có thể cộng thêm 1.    2. Tăng mã giao dịch lớn nhất lên 1 để tạo mã mới.    3. Sau đó chuyển kết quả cộng lại về dạng chuỗi varchar    4. Ghép thêm chuỗi '00000' phía trước giá trị số vừa chuyển đổi để đảm bảo mã có độ dài đủ 5 ký tự.    5. Dùng RIGHT() lấy 5 ký tự cuối cùng của chuỗi sau khi đã ghép, đảm bảo mã giao dịch luôn có độ dài 5 ký tự. Nếu mã nhỏ hơn 5 ký tự, phần đầu sẽ được bù bởi các số 0. |
| **CODE** | go  create or alter function fMaHopDongMoi()  returns char(5)  as  begin  declare @mahopdonglonnhat varchar(10), @mahopdongmoi varchar(10)  select @mahopdonglonnhat = max(MaHopDong)  from HOPDONG  set @mahopdongmoi=RIGHT('00000' + convert(varchar, convert(int, @mahopdonglonnhat)+1),5)  return @mahopdongmoi  end  select dbo.fMaHopDongMoi() |

### Thêm mới một hợp đồng

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thêm mới một hợp đồng nếu biết: CCCD\_NT, SoPhong, NgayThueTro, NgayHetHan. Bao gồm những công việc sau:**   1. **Kiểm tra CCCD\_NT đã tồn tại trong bảng NGUOITHUETRO chưa? Nếu chưa, ngừng xử lý** 2. **Kiểm tra SoPhong đã tồn tại trong bảng PHONG chưa? Nếu chưa, ngừng xử lý** 3. **Kiểm tra ngày thuê trọ có hợp lệ không. Nếu không, ngừng xử lý (hợp lệ không lớn hơn ngày và thời gian hiện tại)** 4. **Kiểm tra ngày hết hạn có hợp lệ không. Nếu không, ngừng xử lý (hợp tính từ ngày thuê đến ngày hết hạn ít nhất 3 tháng)** 5. **Tính mã hợp đồng mới** 6. **Thêm mới bản ghi vào bảng HOPDONG với dữ liệu đã có** | |
| **INPUT** | CCCD\_NT, SoPhong, NgayThueTro, NgayHetHan |
| **OUTPUT** | Xác nhận (Lỗi trả về 0, thành công trả về 1) |
| **PROCESS** | 1. Nếu CCCD\_NT không có trong bảng NGUOITHUETRO thì trả về 0 và đưa ra thông báo ‘CCCD\_NT chưa tồn tại trong bảng NGUOITHUETRO’ và ngừng xử lý 2. Nếu SoPhong không có trong bảng PHONG thì trả về 0 và đưa ra thông báo ‘Số phòng chưa tồn tại trong bảng PHONG’ và ngừng xử lý 3. Nếu ngày thuê trọ lớn ngày hiện tại thì trả về 0 và đưa ra thông báo ‘Ngày thuê trọ không hợp lệ’ và ngừng xử lý 4. Nếu khoảng cách giữa ngày hết hạn và ngày thuê trọ bé hơn 3 tháng thì trả về 0 và đưa ra thông báo ‘Ngày hết hạn không hợp lệ’ và ngừng xử lý 5. Gọi hàm tính mã hợp đồng mới đã làm (fMaHopDongMoi()) 6. Thêm mới bản ghi vào bảng HOPDONG với dữ liệu(@MaHopDongMoi, @CCCD\_NT, @SoPhong, @NgayThueTro, @NgayHetHan) 7. Nếu @@Rowcount <=0 thì trả về 0 và đưa ra thông báo ‘Thêm mới không thành công’ và ngừng xử lý 8. Còn lại thì trả về 1 và đưa ra thông báo ‘Thêm mới thành công’ |
| **CODE** | go  create or alter proc spThemMoiHopDong (@CCCD\_NT char(12), @SoPhong char(3), @NgayThueTro date, @NgayHetHan date, @ret varchar(50) output)  as  begin  declare @MaHopDongMoi varchar(10)  --cau 1  if @CCCD\_NT not in (select CCCD\_NT from NGUOITHUETRO)  begin  set @ret = 0  print N'CCCD\_NT chưa tồn tại trong bảng NGUOITHUETRO'  return  end  --cau 2  if @SoPhong not in (select SoPhong from Phong)  begin  set @ret = 0  print N'Số phòng chưa tồn tại trong bảng PHONG'  return  end  --cau 3  if @NgayThueTro > getdate()  begin  set @ret = 0  print N'Ngày thuê trọ không hợp lệ'  return  end  --cau 4  if DATEDIFF(Month, @NgayThueTro, @NgayHetHan) < 3  begin  set @ret = 0  print N'Ngày hết hạn không hợp lệ'  return  end  --cau 5  set @MaHopDongMoi = dbo.fMaHopDongMoi()  --cau 6  insert into HOPDONG values (@MaHopDongMoi, @CCCD\_NT, @SoPhong, @NgayThueTro, @NgayHetHan)  if @@ROWCOUNT <= 0  begin  set @ret = 0  print N'Thêm mới không thành công'  return  end  else  begin  set @ret = 1  print N'Thêm mới thành công'  end  end  go  declare @CCCD\_NT char(12)='000000000001',  @SoPhong char(3)='001',  @NgayThueTro date='2022-10-10',  @NgayHetHan date='2024-06-30',  @ret varchar(50)  exec spThemMoiHopDong @CCCD\_NT, @SoPhong, @NgayThueTro, @NgayHetHan, @ret output  print @ret |

### Thêm mới một hóa đơn

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thêm mới một hóa đơn nếu biết: CCCD\_NT, SoPhong, NgayThanhToan. Bao gồm những công việc sau:**   1. **Kiểm tra CCCD\_NT đã tồn tại trong bảng NGUOITHUETRO chưa? Nếu chưa, ngừng xử lý** 2. **Kiểm tra SoPhong đã tồn tại trong bảng PHONG chưa? Nếu chưa, ngừng xử lý** 3. **Kiểm tra NgayThanhToan có hợp lệ không. Nếu không, ngừng xử lý (hợp lệ không lớn hơn ngày và thời gian hiện tại)** 4. **Tính mã hóa đơn mới** 5. **Thêm mới bản ghi vào bảng HOADON với dữ liệu đã có** | |
| **INPUT** | **CCCD\_NT, SoPhong, NgayThanhToan** |
| **OUTPUT** | Xác nhận (Lỗi trả về 0, thành công trả về 1) |
| **PROCESS** | 1. Nếu CCCD\_NT không có trong bảng NGUOITHUETRO thì trả về 0 và đưa ra thông báo ‘CCCD\_NT chưa tồn tại trong bảng NGUOITHUETRO’ và ngừng xử lý 2. Nếu SoPhong không có trong bảng PHONG thì trả về 0 và đưa ra thông báo ‘Số phòng chưa tồn tại trong bảng PHONG’ và ngừng xử lý 3. Nếu ngày thanh toán lớn ngày hiện tại thì trả về 0 và đưa ra thông báo ‘Ngày thanh toán không hợp lệ’ và ngừng xử lý 4. Gọi thủ tục tính mã hóa đơn mới đã làm (spMaHoaDonMoi ()) 5. Thêm mới bản ghi vào bảng HOADON dữ liệu (@MaHoaDonMoi, @CCCD\_NT, @SoPhong, @NgayThanhToan) 6. Nếu @@Rowcount <=0 thì trả về 0 và đưa ra thông báo ‘Thêm mới không thành công’ và ngừng xử lý 7. Còn lại thì trả về 1 và đưa ra thông báo ‘Thêm mới thành công’ |
| **CODE** | go  create or alter proc spThemMoiHoaDon (@CCCD\_NT char(12), @SoPhong char(3), @NgayThanhToan date, @ret nvarchar(50) output)  as  begin  declare @MaHoaDonMoi varchar(10)  --cau 1  if @CCCD\_NT not in (select CCCD\_NT from NGUOITHUETRO)  begin  set @ret = 0  print N'CCCD\_NT chưa tồn tại trong bảng NGUOITHUETRO'  return  end  --cau 2  if @SoPhong not in (select SoPhong from Phong)  begin  set @ret = 0  print N'Số phòng chưa tồn tại trong bảng PHONG'  return  end  -- cau 3  if @NgayThanhToan > getdate()  begin  set @ret = 0  print N'Ngày thanh toán không hợp lệ'  return  end  --cau 4  exec spMaHoaDonMoi @MaHoaDonMoi output  --cau 5  insert into HOADON (MaHoaDon, CCCD\_NT, SoPhong, NgayThanhToan)  values (@MaHoaDonMoi, @CCCD\_NT, @SoPhong, @NgayThanhToan)  if @@ROWCOUNT <= 0  begin  set @ret = 0  print N'Thêm mới không thành công'  return  end  else  begin  set @ret = 1  print N'Thêm mới thành công'  end  end    go  declare @CCCD\_NT char(12)='000000000001',  @SoPhong char(3)='001',  @NgayThanhToan date='2023-06-23',  @ret nvarchar(50)  exec spThemMoiHoaDon @CCCD\_NT, @SoPhong, @NgayThanhToan, @ret output  print @ret |

### Thêm mới một hóa đơn chi tiết

|  |  |
| --- | --- |
| **LOẠI: Thủ tục** | |
| **MÔ TẢ: Thêm mới một hóa đơn chi tiết nếu biết: MaHoaDon, MaDV, SoDung. Bao gồm những công việc sau:**   1. **Kiểm tra MaHoaDon đã tồn tại trong bảng HOADON chưa? Nếu chưa, ngừng xử lý** 2. **Kiểm tra MaDV đã tồn tại trong bảng DICHVU chưa? Nếu chưa, ngừng xử lý** 3. **Kiểm tra SoDung có hợp lệ không. Nếu không, ngừng xử lý (không thể <0)** 4. **Thêm mới bản ghi vào bảng HOADONCHITIET với dữ liệu đã có** | |
| **INPUT** | **MaHoaDon, MaDV, SoDung** |
| **OUTPUT** | Xác nhận (Lỗi trả về 0, thành công trả về 1) |
| **PROCESS** | 1. Nếu MaHoaDon không có trong bảng HOADON thì trả về 0 và đưa ra thông báo ‘Mã hóa đơn chưa tồn tại trong bảng HOADON’ và ngừng xử lý 2. Nếu MaDV không có trong bảng DICHVU thì trả về 0 và đưa ra thông báo ‘Mã dịch vụ chưa tồn tại trong bảng DICHVU’ và ngừng xử lý 3. Nếu số dùng < 0 thì trả về 0 và đưa ra thông báo ‘Số dùng không hợp lệ’ và ngừng xử lý 4. Thêm mới bản ghi vào bảng HOADONCHITIET dữ liệu (@MaHoaDon, @MaDV, @SoDung) 5. Nếu @@Rowcount <=0 thì trả về 0 và đưa ra thông báo ‘Thêm mới không thành công’ và ngừng xử lý 6. Còn lại thì trả về 1 và đưa ra thông báo ‘Thêm mới thành công’ |
| **CODE** | go  create or alter proc spThemMoiHoaDonChiTiet (@MaHoaDon char(6), @MaDV char(4), @SoDung int, @ret nvarchar(50) output)  as  begin  --cau 1  if @MaHoaDon not in (select MaHoaDon from HOADON)  begin  set @ret = 0  print N'Mã hóa đơn chưa tồn tại trong bảng HOADON'  return  end  --cau 2  if @MaDV not in (select MaDV from DICHVU)  begin  set @ret = 0  print N'Mã dịch vụ chưa tồn tại trong bảng DICHVU'  return  end  -- cau 3  if @SoDung < 0  begin  set @ret = 0  print N'Số dùng không hợp lệ'  return  end  --cau 4  insert into HOADONCHITIET (MaHoaDon, MaDV, SoDung)  values(@MaHoaDon, @MaDV, @SoDung)  if @@ROWCOUNT <= 0  begin  set @ret = 0  print N'Thêm mới không thành công'  return  end  else  begin  set @ret = 1  print N'Thêm mới thành công'  end  end  go  declare @MaHoaDon char(6)='001001',  @MaDV char(4)='0001',  @SoDung int=15,  @ret varchar(50)  exec spThemMoiHoaDonChiTiet @MaHoaDon, @MaDV, @SoDung, @ret output  print @ret |

# XÂY DỰNG CƠ CHẾ BẢO MẬT (R5)

## Cơ chế bảo mật

Tạo Login và phân quyền cho hai đối tượng: chủ trọ và người thuê trọ

## Bảng phân quyền

|  |  |  |
| --- | --- | --- |
| **Bảng** | **Chủ trọ** | **Người thuê trọ** |
| CHUTRO | delete, insert, select, update | N/A |
| DICHVU | delete, insert, select, update | select |
| HOADONCHITIET | delete, insert, select, update | select |
| HOADON | delete, insert, select, update | select |
| HOPDONG | delete, insert, select, update | select |
| NGUOITHUETRO | delete, insert, select, update | select |
| PHONG | delete, insert, select, update | select |
| TAIKHOAN\_NT | delete, insert, select, update | N/A |

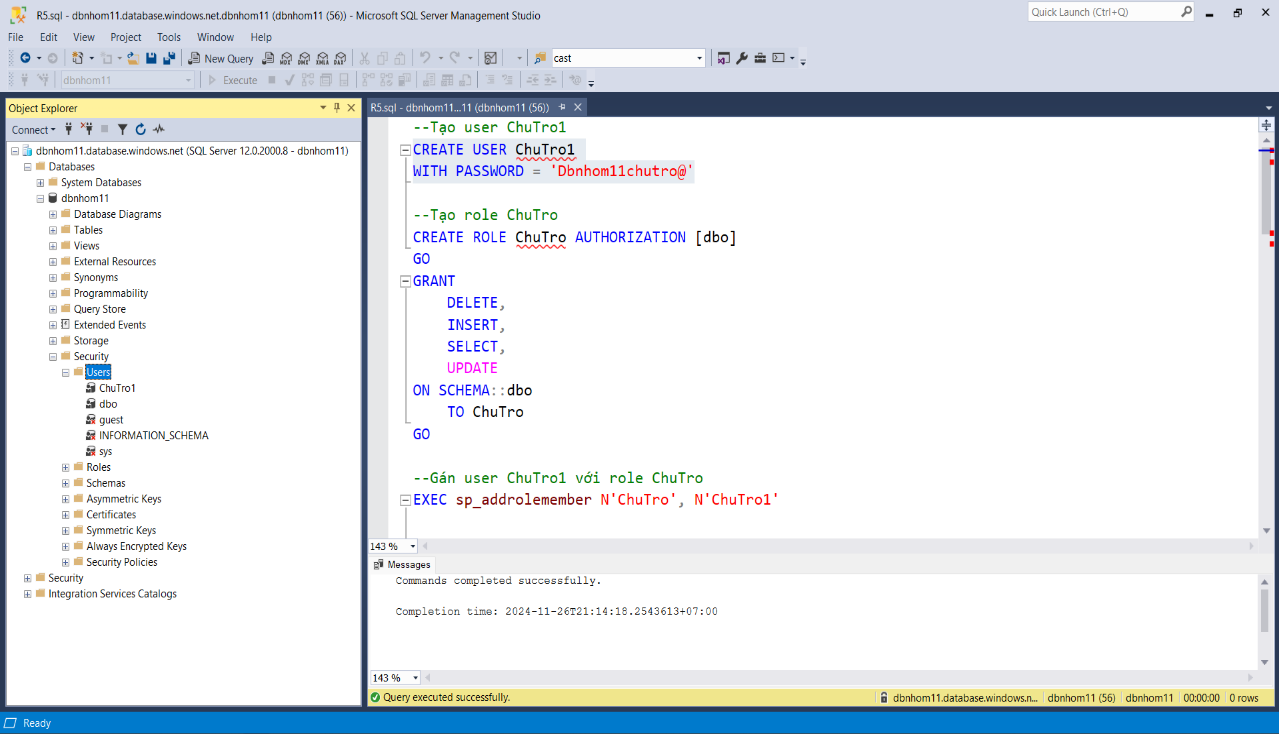
## Thực hiện triển khai trên SQL Server

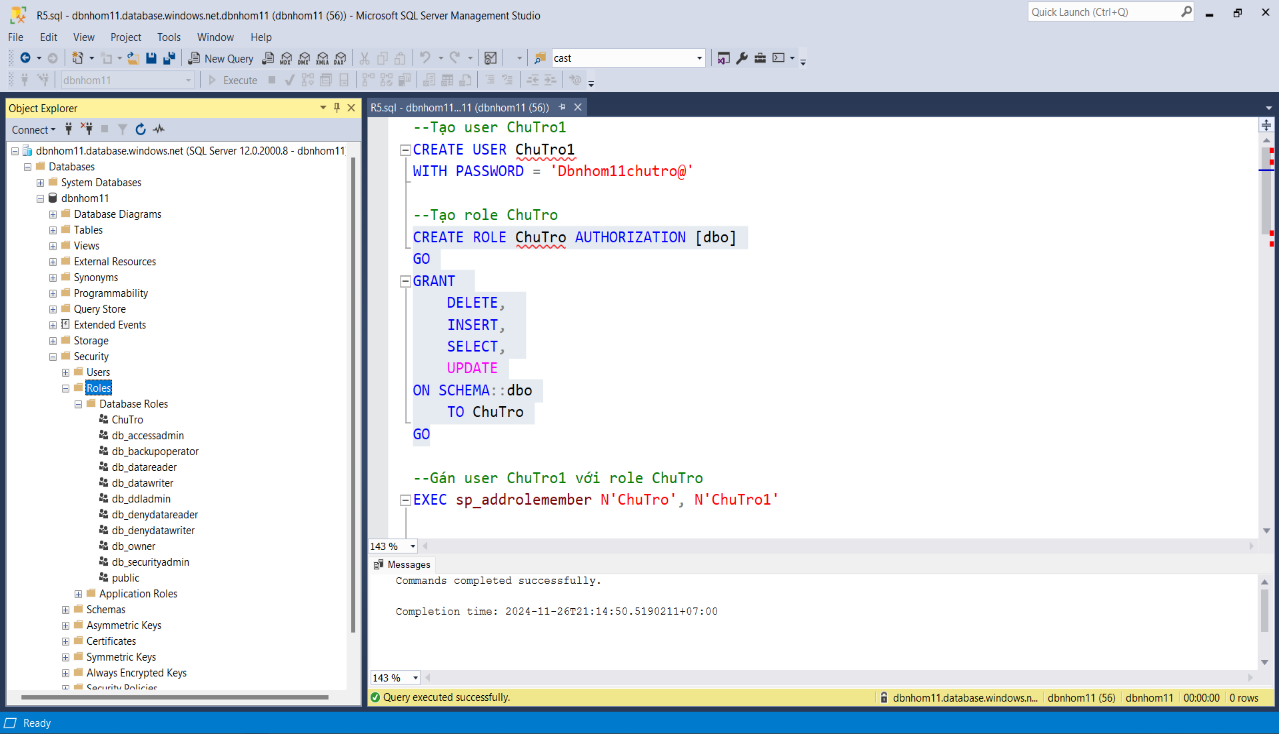
### Đối với chủ trọ

* **Bước 1:** Tạo User

Username: ChuTro1

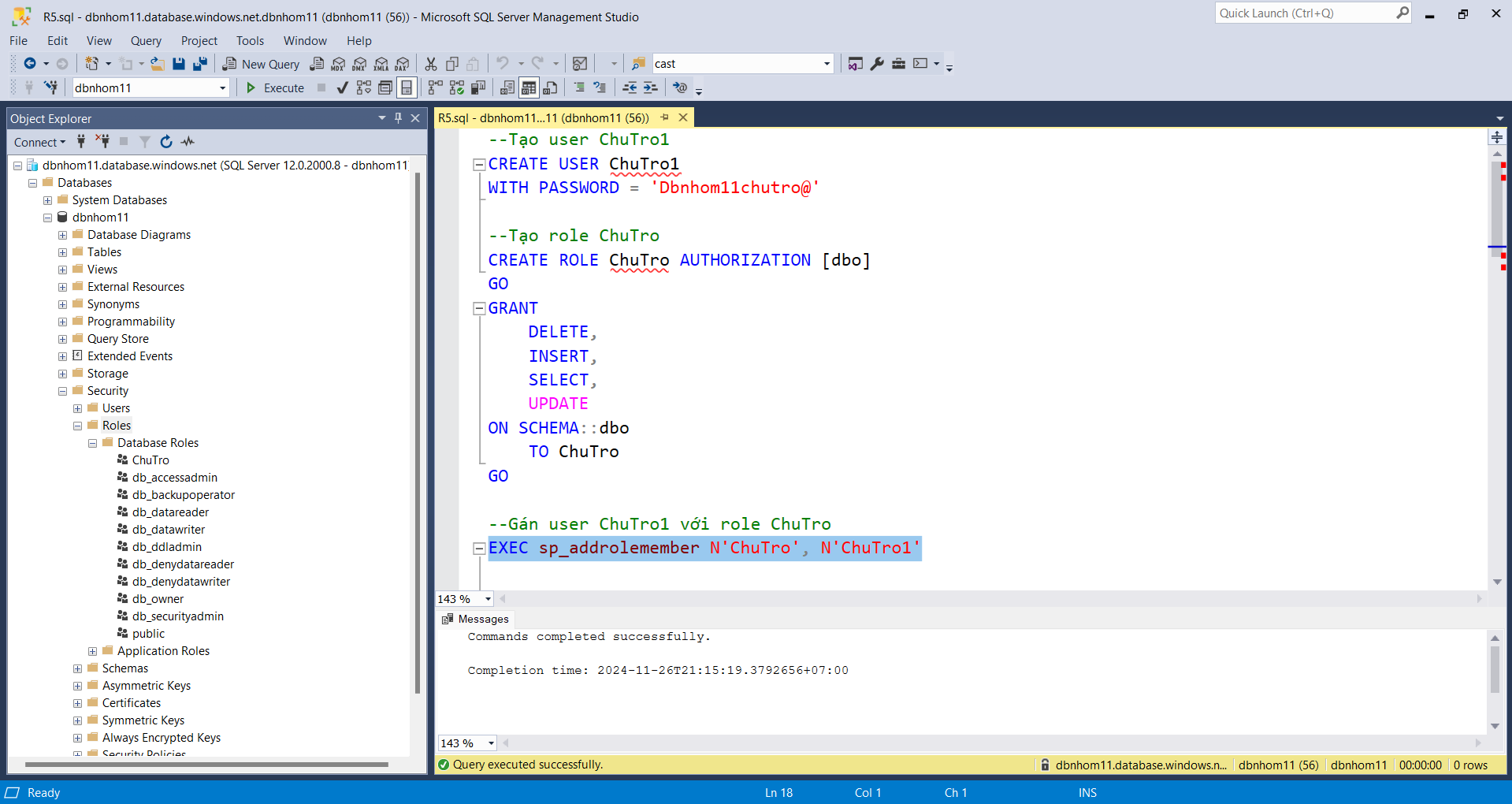
Password: Dbnhom11chutro@

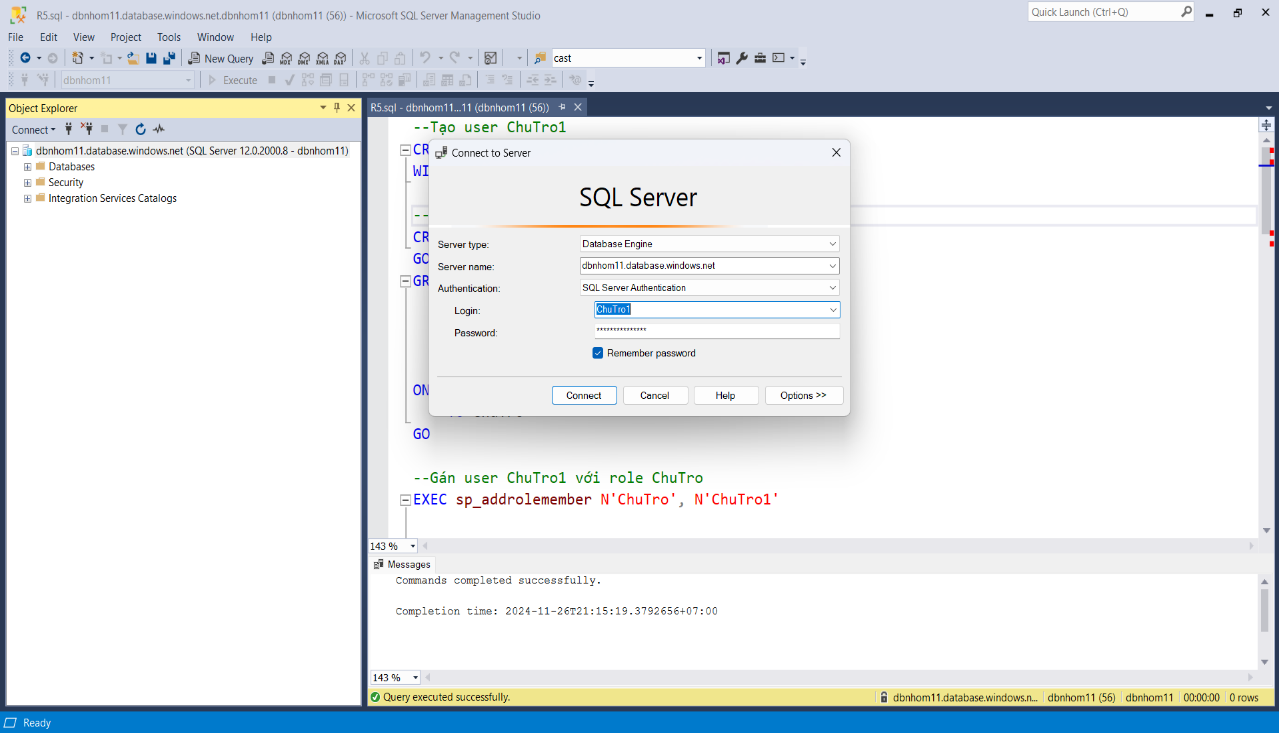
Hình : Tạo User ChuTro1 thành công

* **Bước 2:** Tạo Role ChuTro với quyền được phân trong bảng phân quyền

Hình : Tạo Role ChuTro thành công

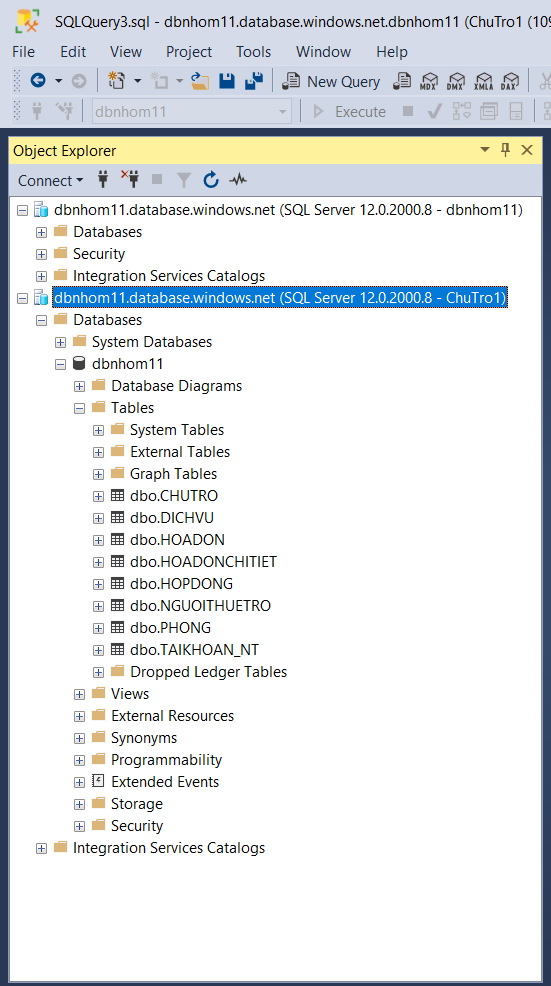
* **Bước 3:** Gán User ChuTro1 với Role ChuTro được tạo

Hình : Gán User ChuTro1 với Role ChuTro thành công

* **Bước 4:** Đăng nhập User ChuTro1

Hình : Đăng nhập user ChuTro1

* **Bước 5:** Đăng nhập thành công User ChuTro1

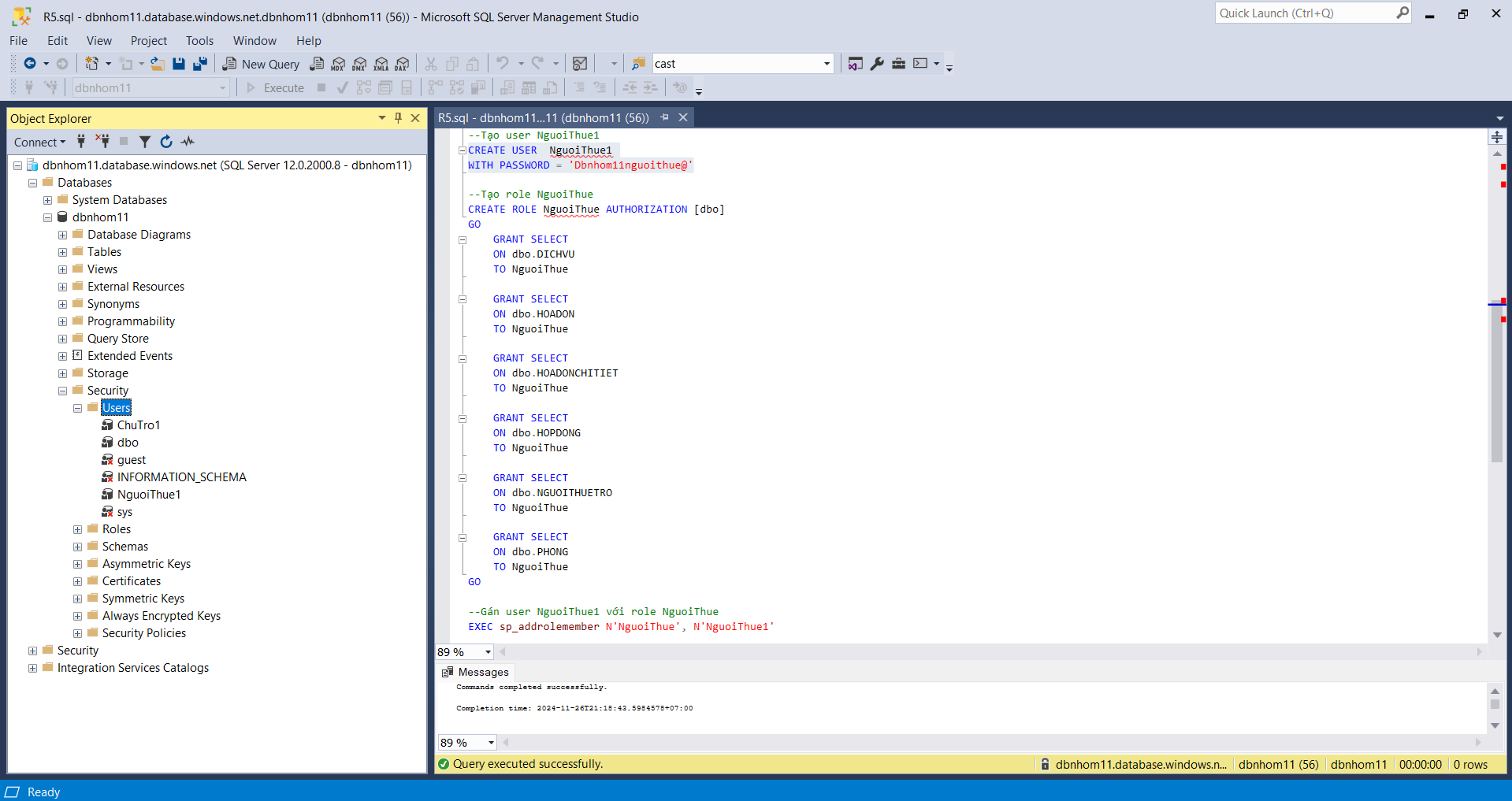
Hình : Đăng nhập thành công user ChuTro1

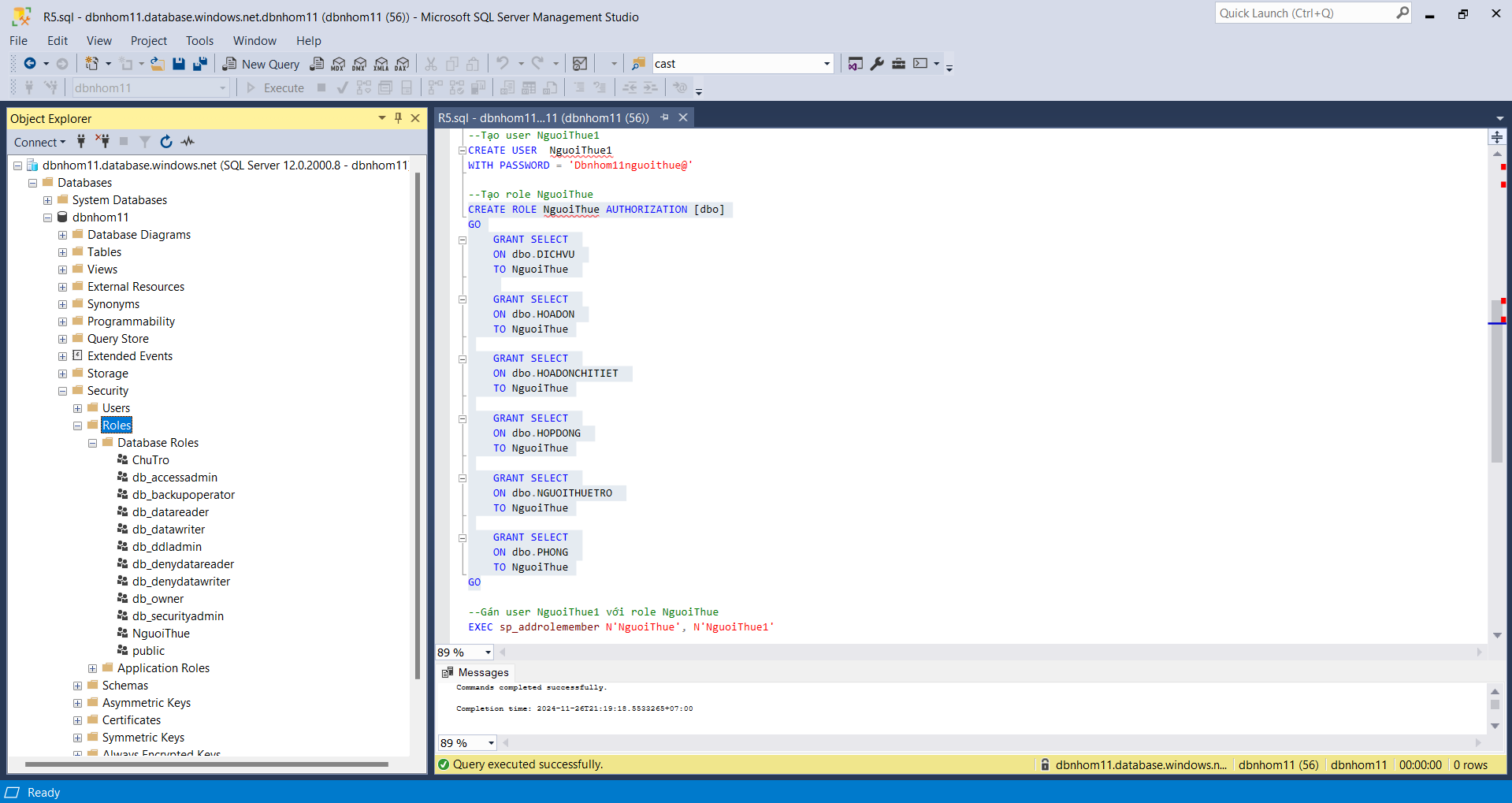
### Đối với người thuê trọ

* **Bước 1:** Tạo User

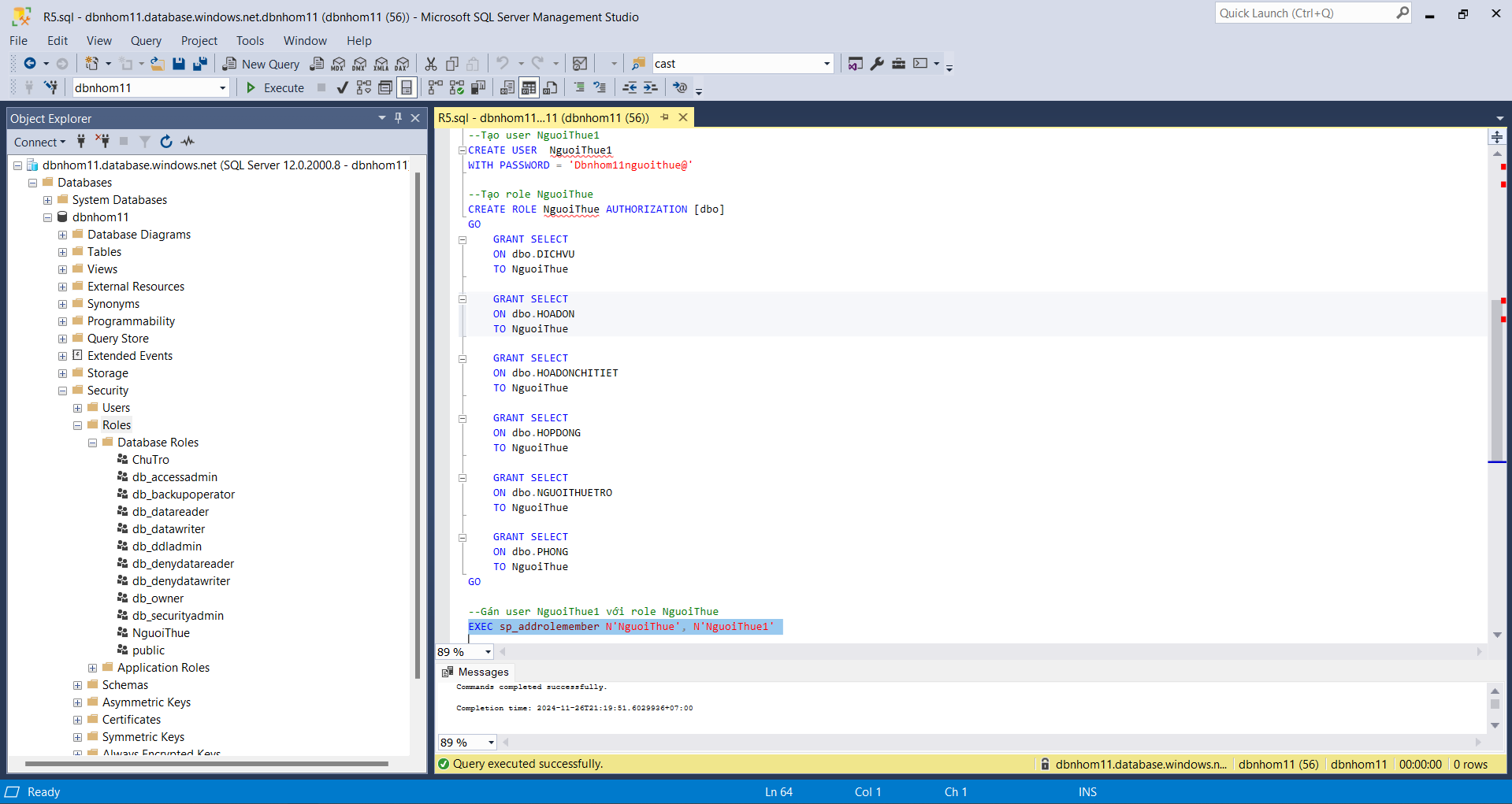
Username: NguoiThue1

Password: Dbnhom11nguoithue@

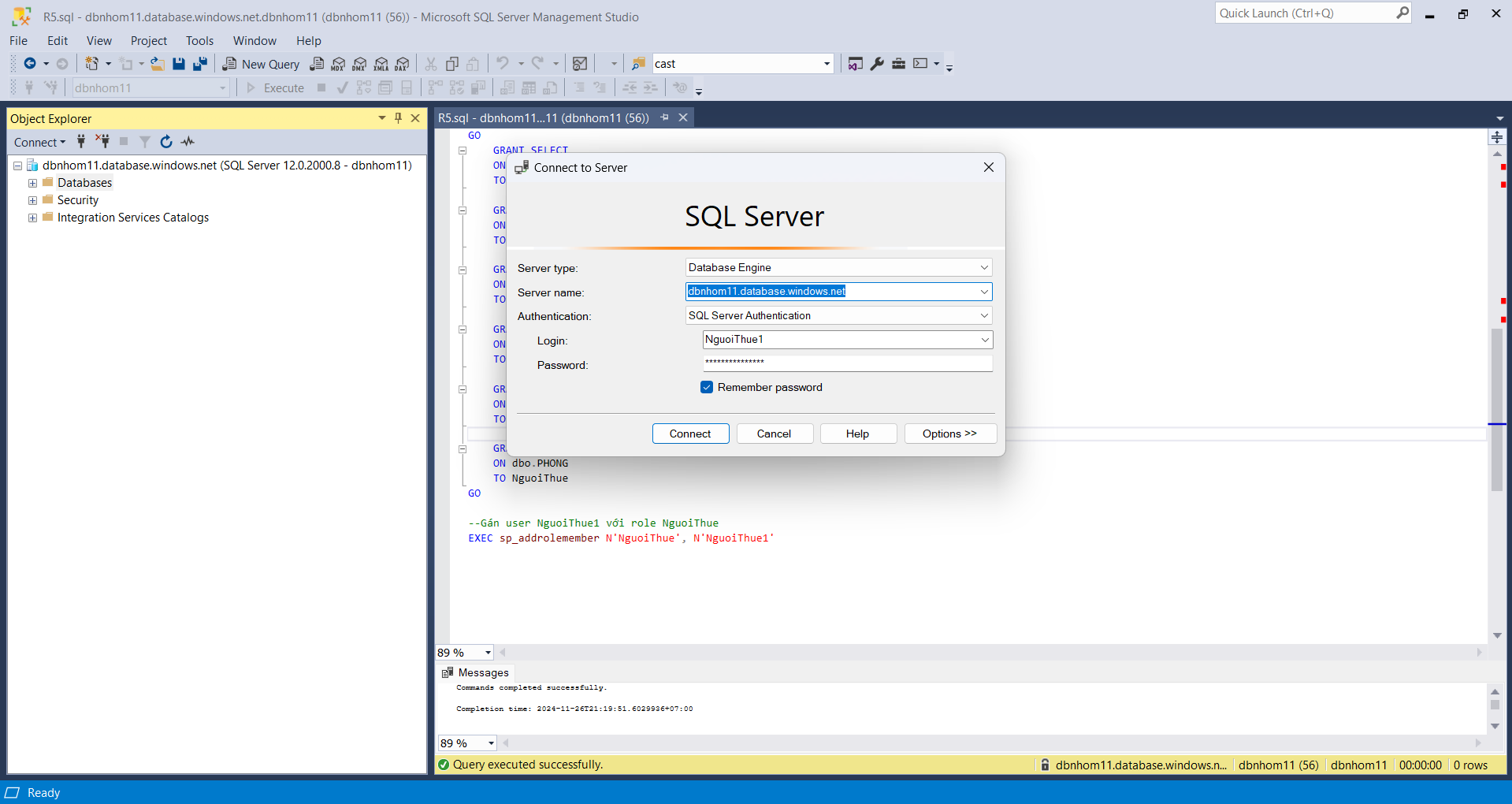
Hình : Tạo User NguoiThue1 thành công

* **Bước 2:** Tạo Role NguoiThue với quyền được phân trong bảng phân quyền

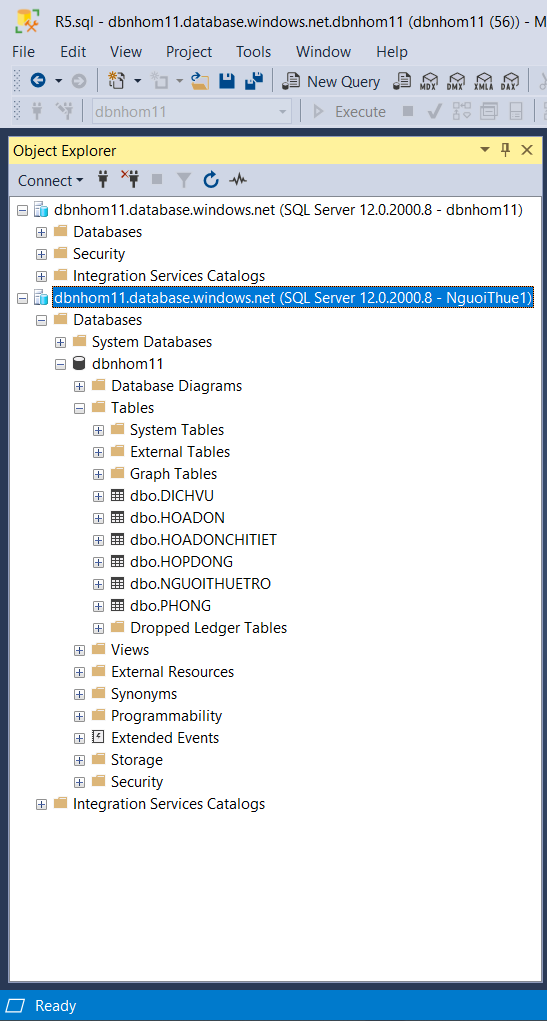
Hình : Tạo Role NguoiThue thành công

* **Bước 3:** Gán User NguoiThue1 với Role NguoiThue được tạo

Hình : Gán User ChuTro1 với Role ChuTro thành công

* **Bước 4:** Đăng nhập User NguoiThue1

Hình : Đăng nhập User NguoiThue1

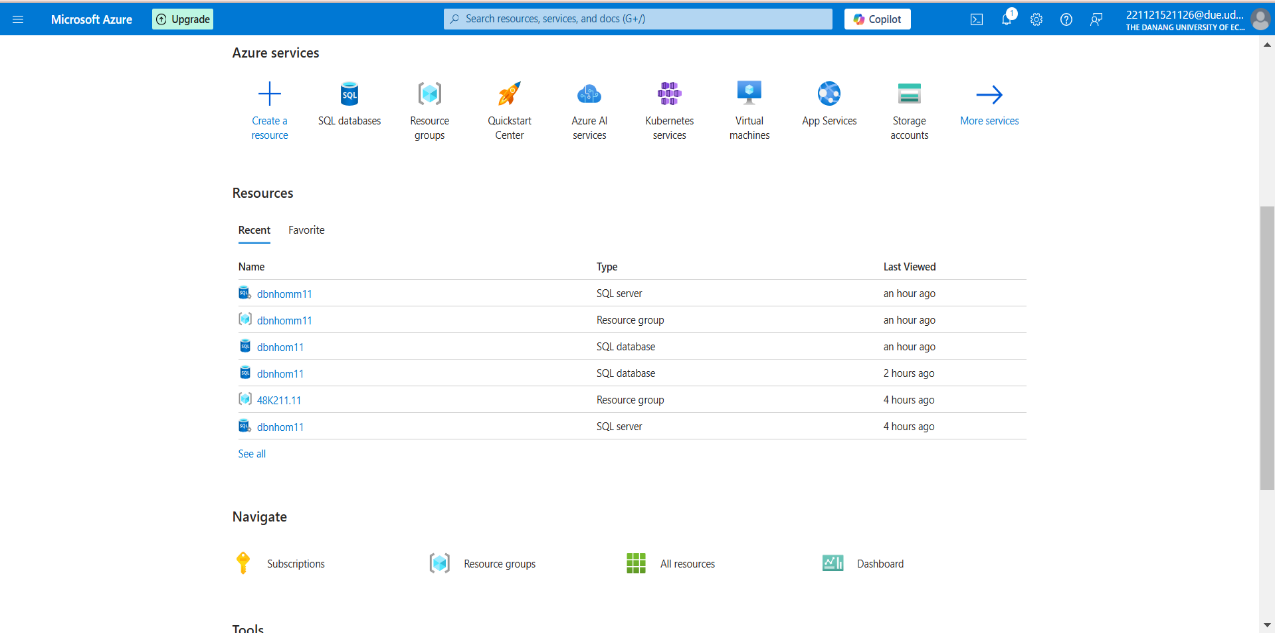
* **Bước 5:** Đăng nhập thành công User NguoiThue1

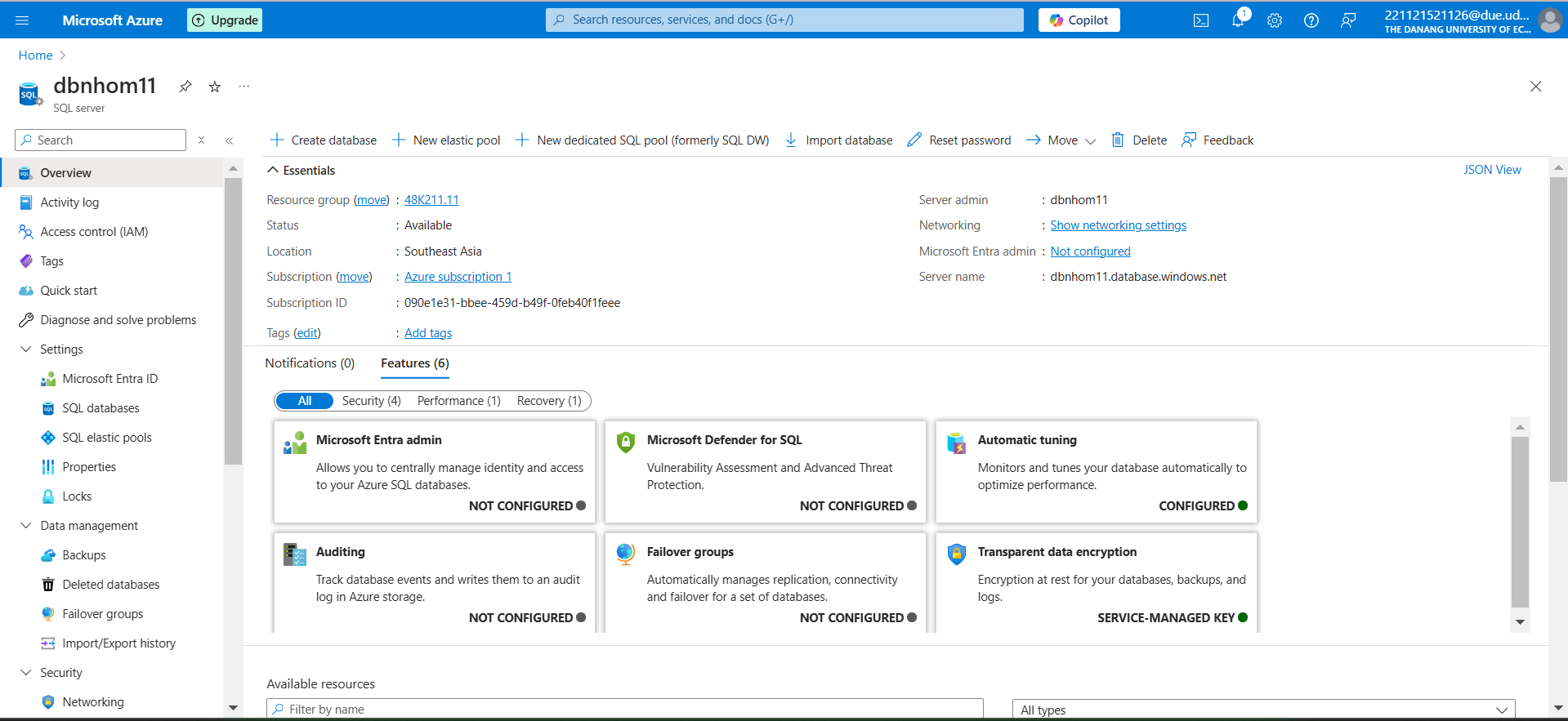
Hình : Đăng nhập thành công User NguoiThue1

# XÂY DỰNG CƠ CHẾ BACKUP DỮ LIỆU TỰ ĐỘNG (R6)

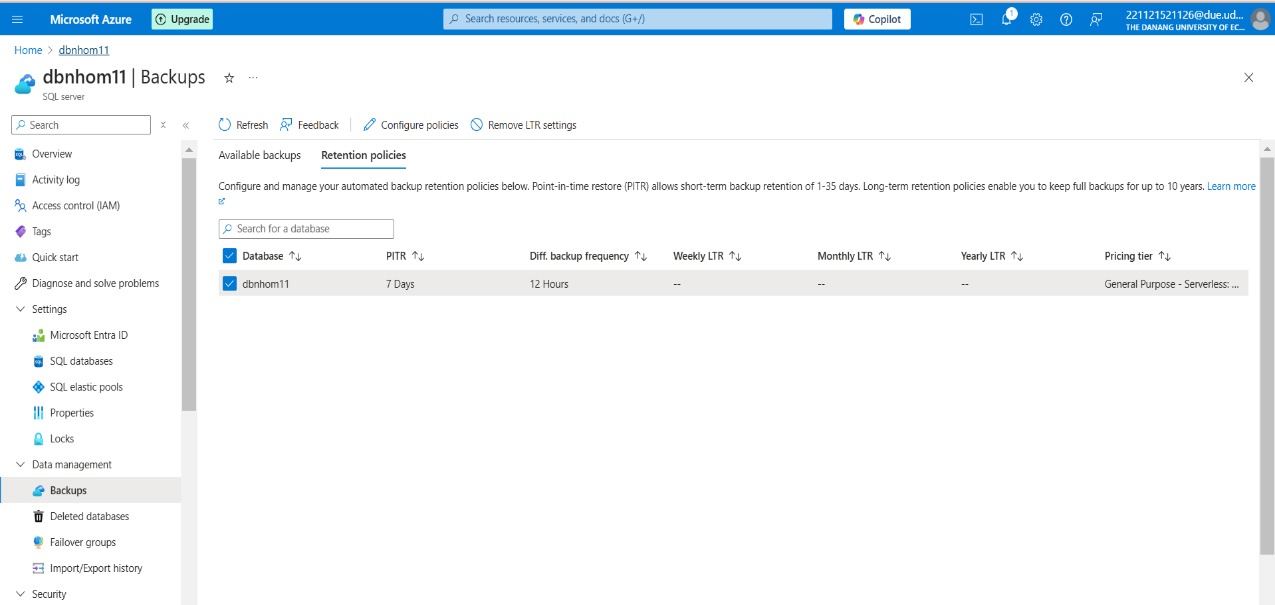
## Backup dữ liệu

* **Bước 1:** Đăng nhập tài khoản vào Microsoft Azure, chọn SQL Server muốn Backup

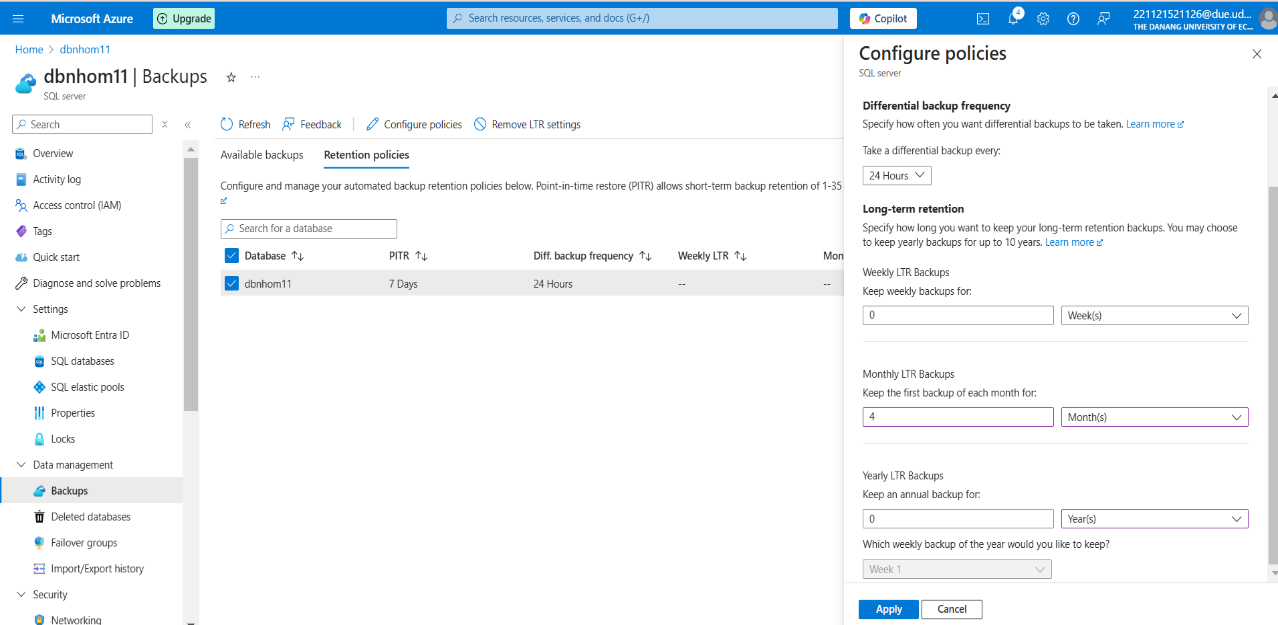
Hình : Chọn SQL Server

* **Bước 2:** Mở ra trình duyệt của SQL Server dbnhom11

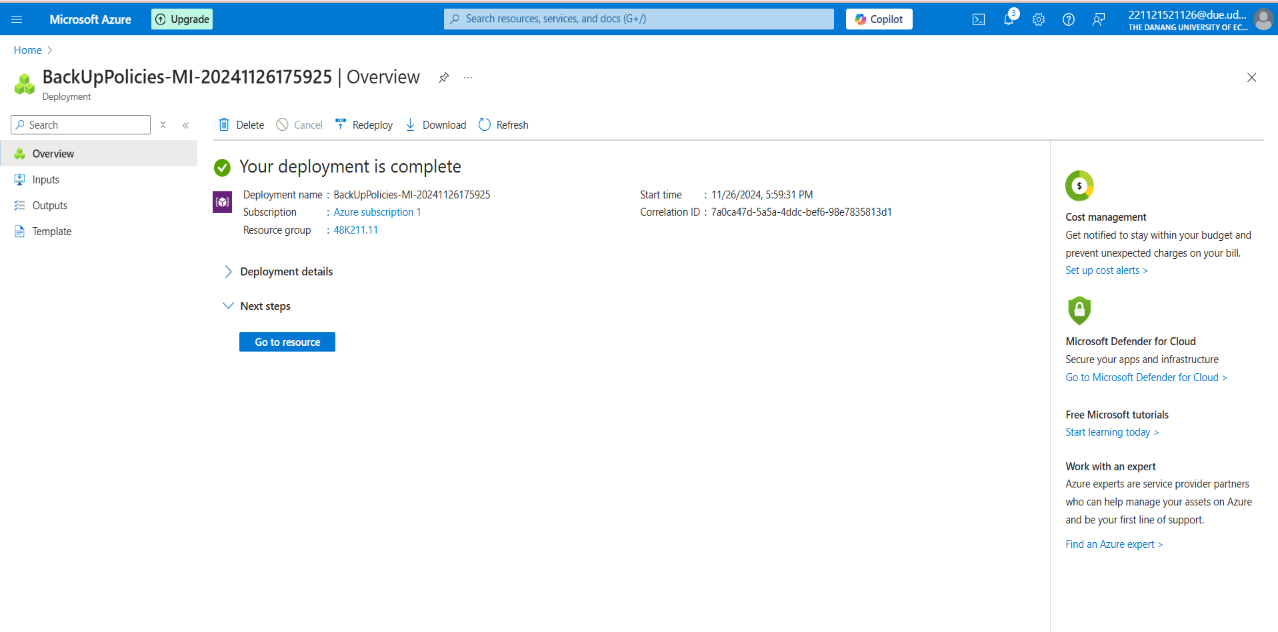
Hình : Trình duyệt SQL Server dbnhom11

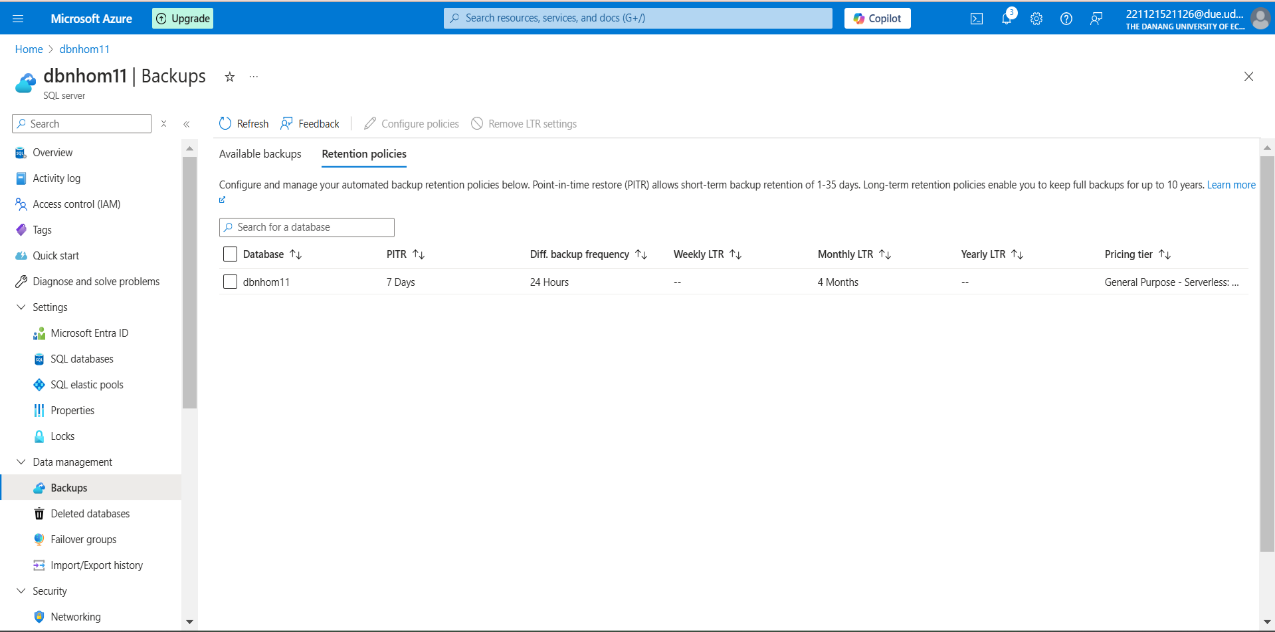
* **Bước 3:** Chọn vào mục Backups, sau đó chọn Retention policies , chọn Database dbnhom11

Hình : Chọn Backups

* **Bước 4:** Thiết lập thời gian giữ Backup

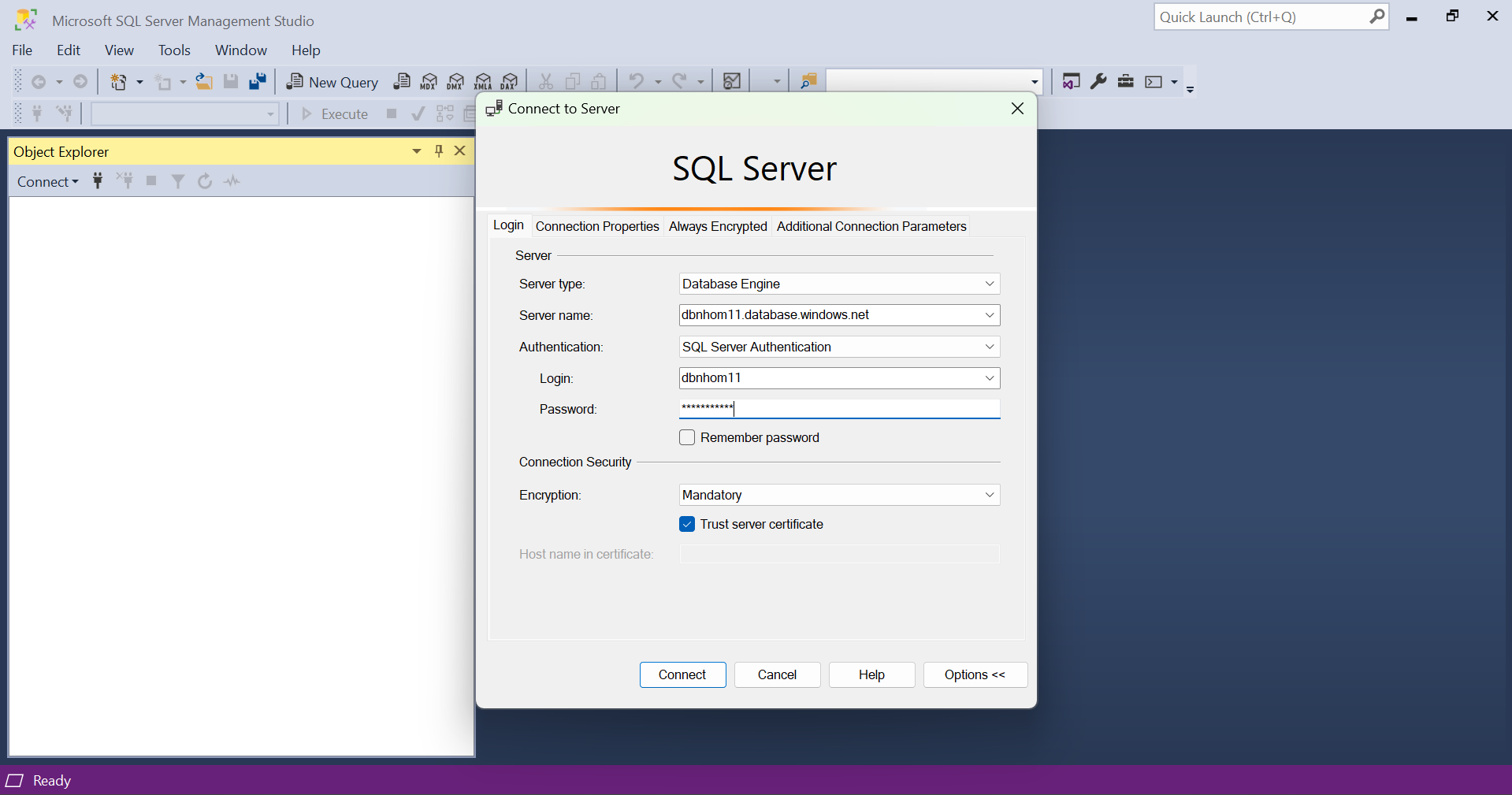
Hình : Thiết lập thời gian

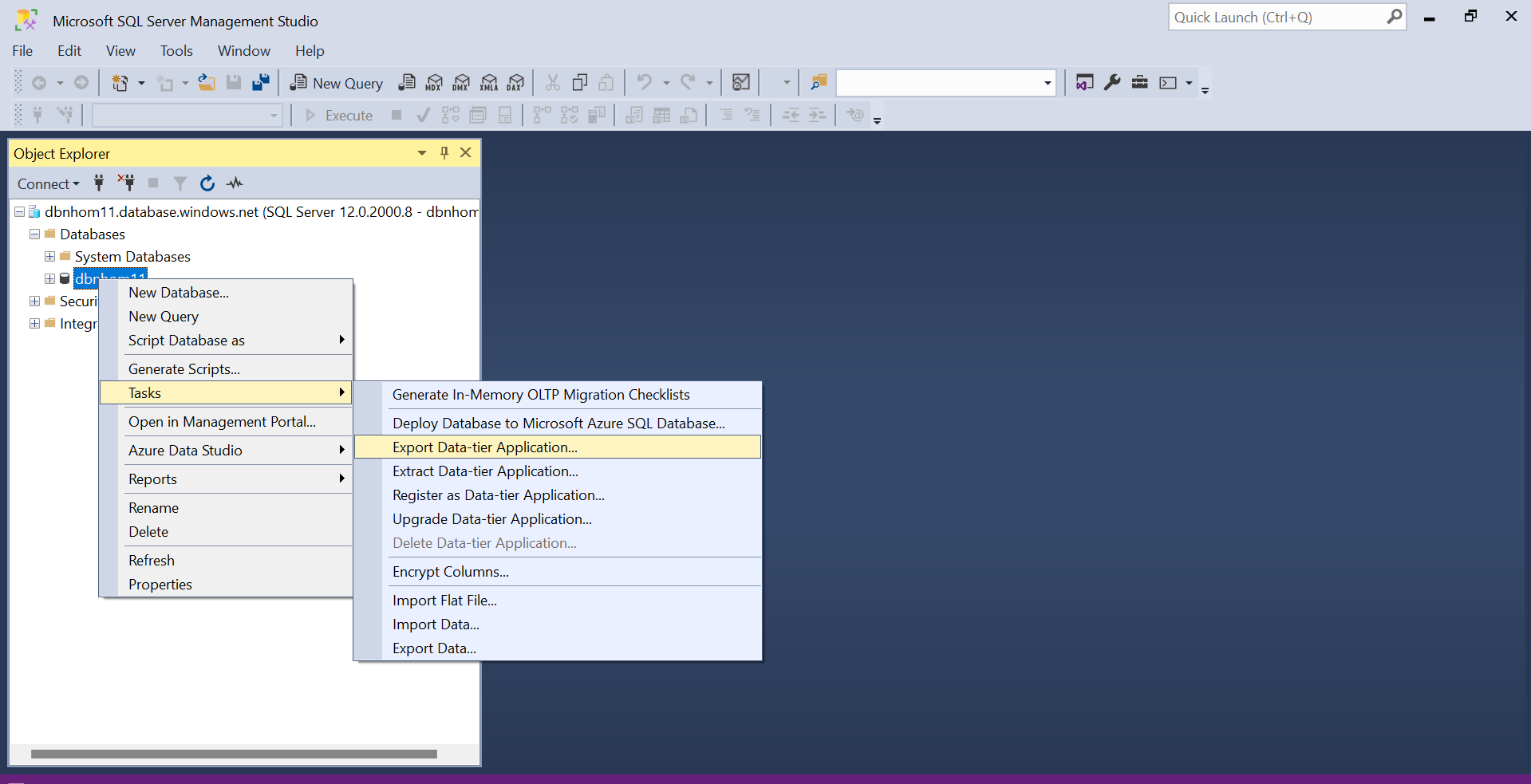
* **Bước 5:** Chọn Apply và Backup thành công

Hình : Triển khai Backup thành công

Hình : Backup thành công

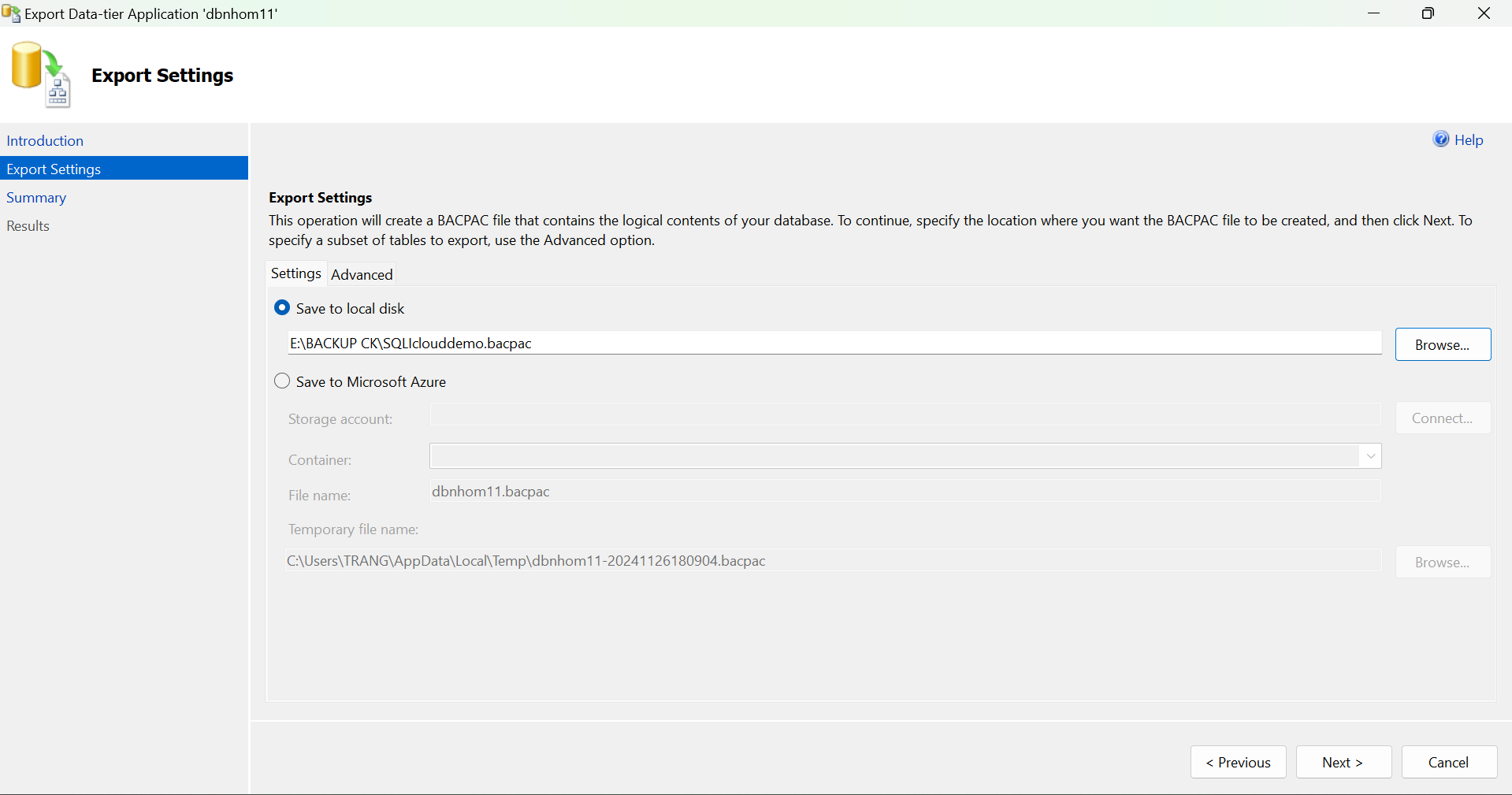
* **Bước 6:** Sao chép link Server name để liên kết với SQL server

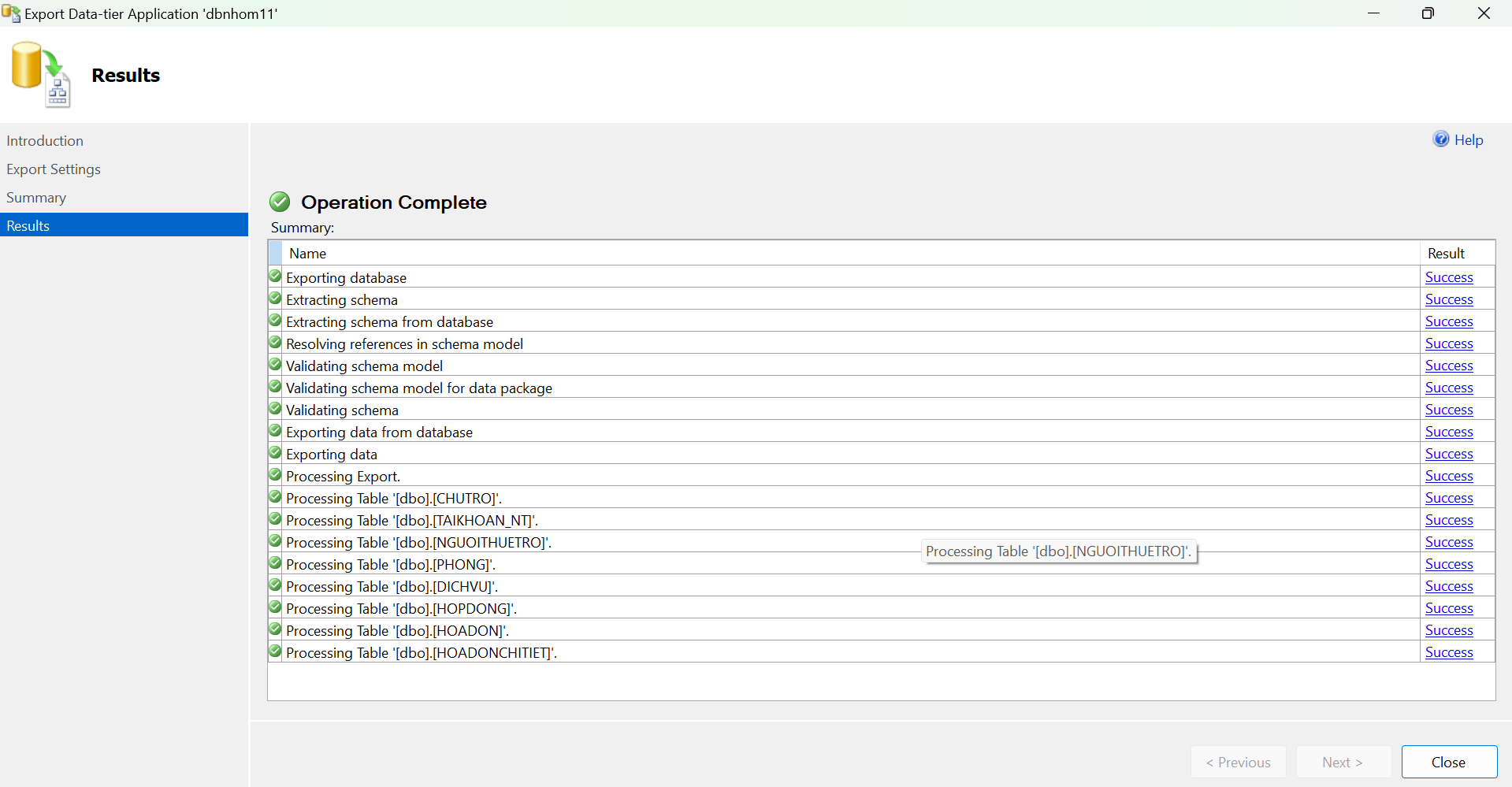
Hình : Liên kết SQL Server với database vừa tạo trên Azure

* **Bước 7:** Tại Databases dbnhom11, chọn Tasks , sau đó chọn Export Data tier Application

Hình : Chọn Export Data tier Application

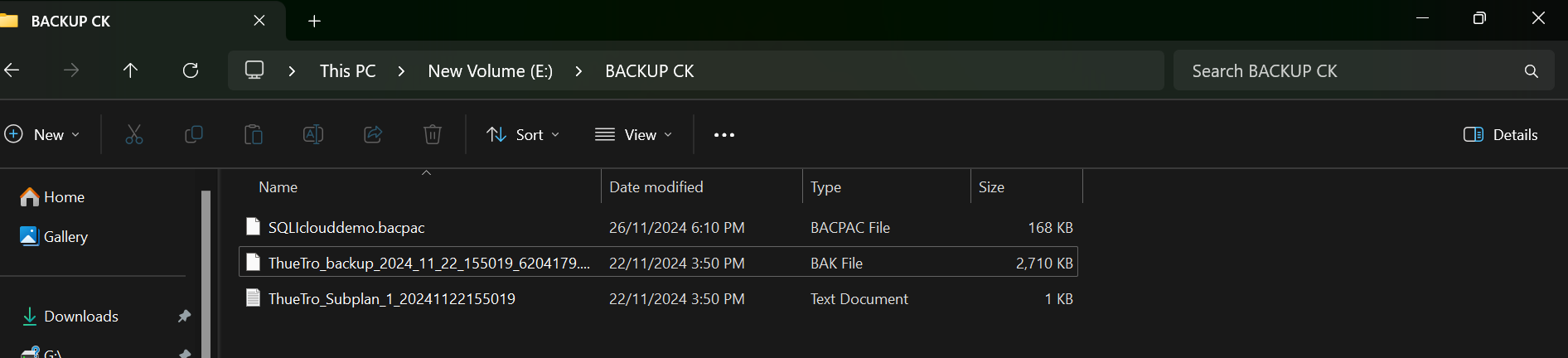
* **Bước 8:** Tại Export Settings chọn thư mục để lưu trữ file Backup

Hình : Chọn thư mục để lưu trữ file Backup

* **Bước 9:** Chọn Finish và tạo file Backup thành công

Hình : Tạo file Backup thành công

* **Bước 10:** Kiểm tra nơi lưu trữ file Backup

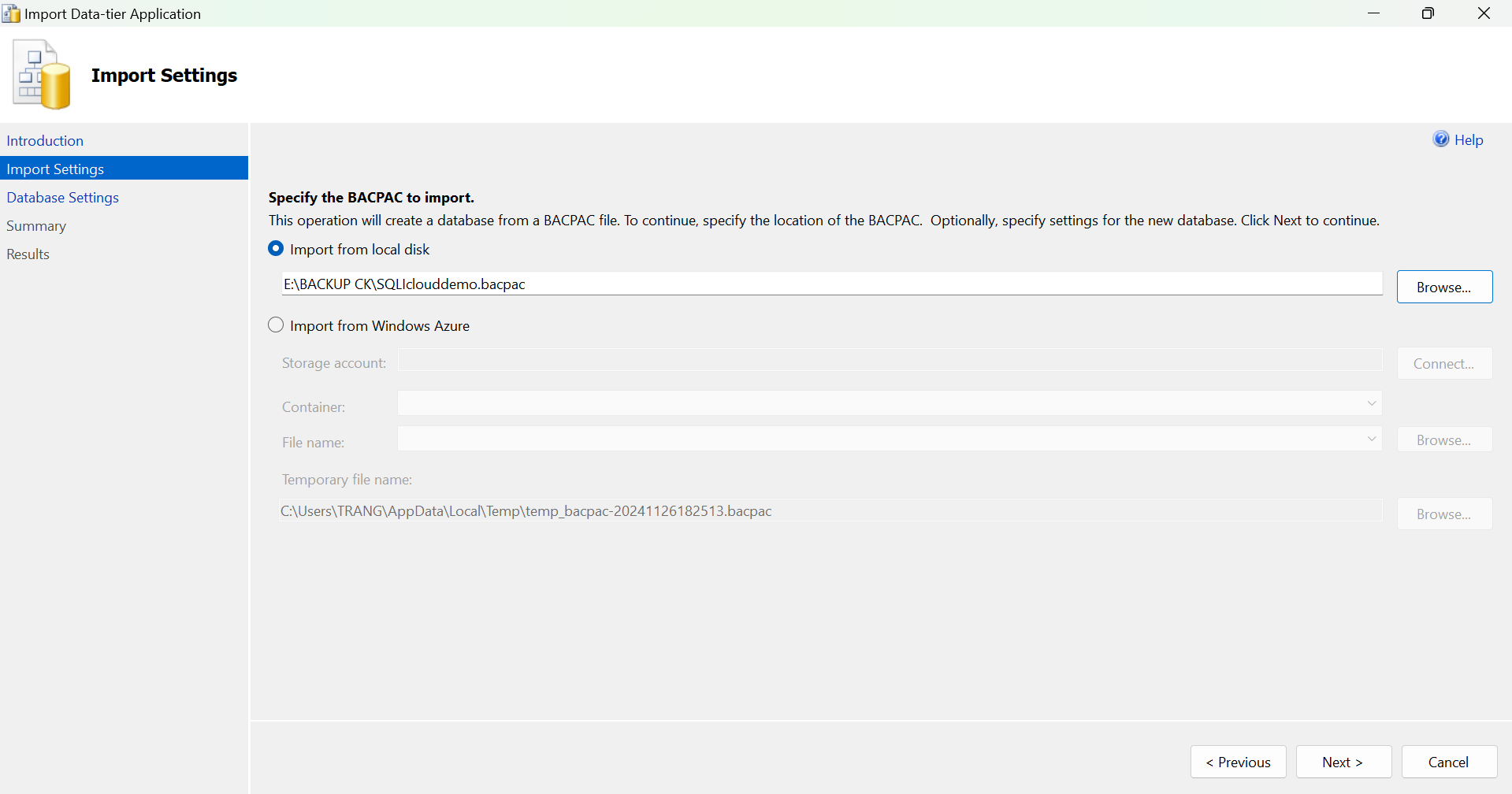
Hình 21: Xem nơi lưu trữ Backup

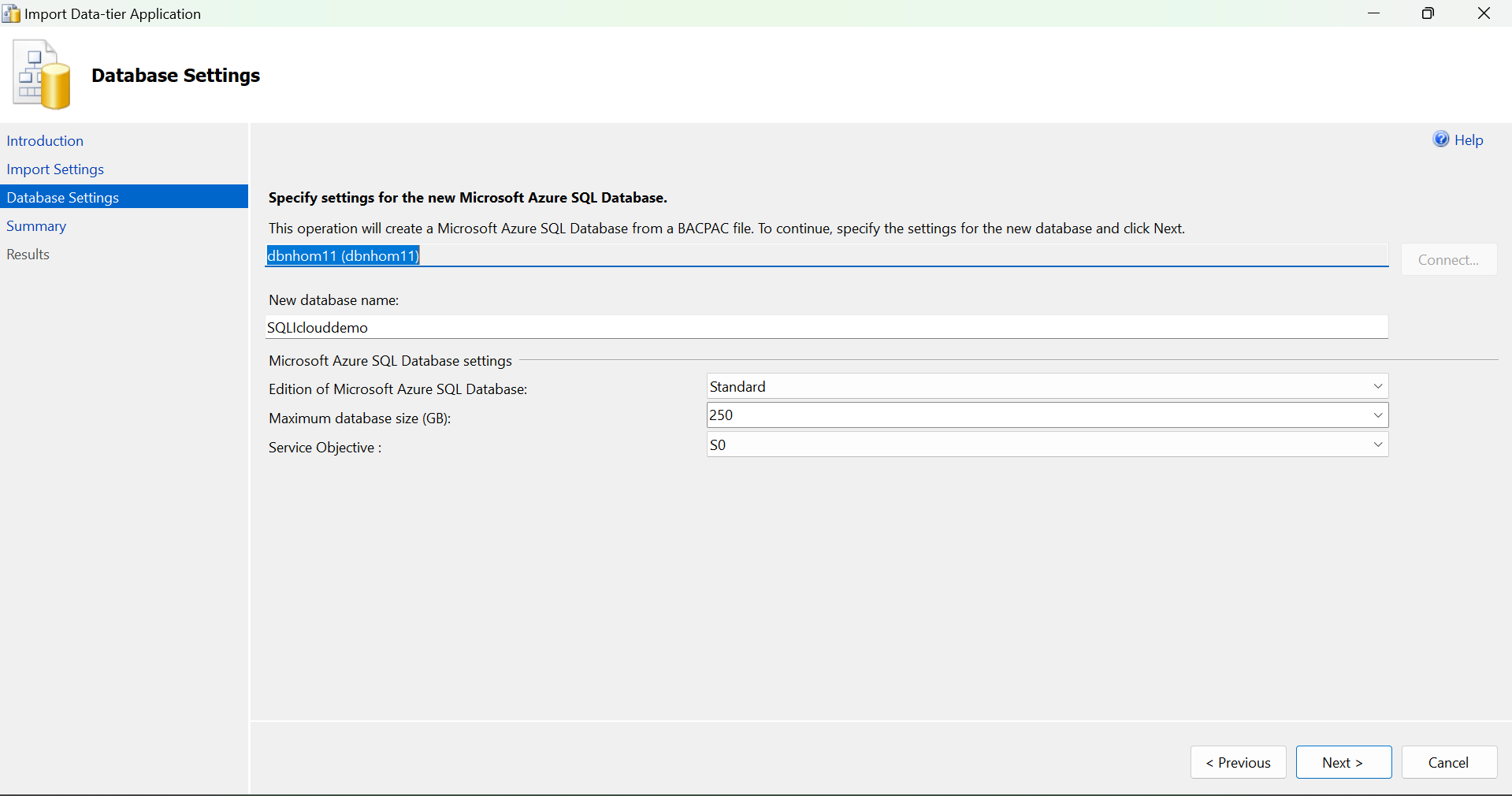
## Restore dữ liệu

* **Bước 1:** Tại Databases dbnhom11, chọn Import Data-tier Application

Hình : Chọn Import Data-tier Application

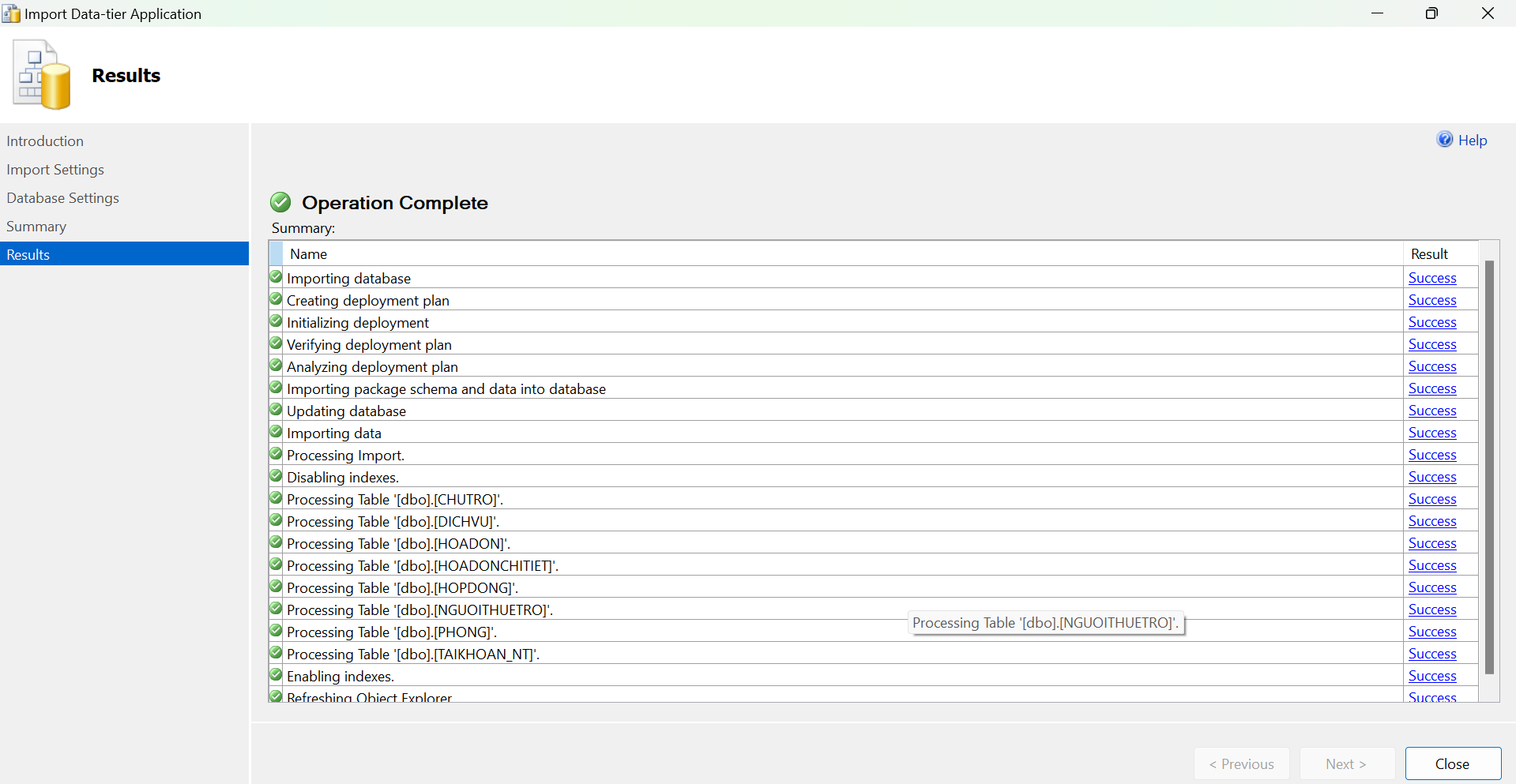
* **Bước 2:** Tại Import Settings , chọn file Backup đã lưu trữ là ‘E:\BACKUP CK\SQLIclouddemo’

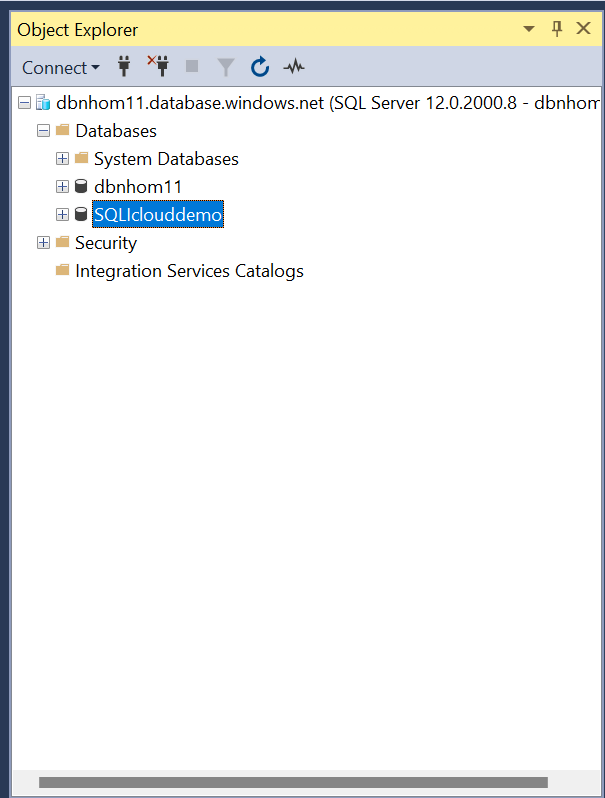
****Hình : Chọn file Backup

* **Bước 3:** Chọn Next để cấu hình các thông số khi nhập dữ liệu từ file BACPAC

Hình 24: Cấu hình các thông số

* **Bước 4:** Chọn Finish và Restore file Backup thành công

Hình ****: Restore thành công

* **Bước 5:** Kiểm tra Database vừa Restore

Hình : Xem Database vừa Restore

# LƯU TRỮ DỮ LIỆU (R7)

## Lý do lựa chọn

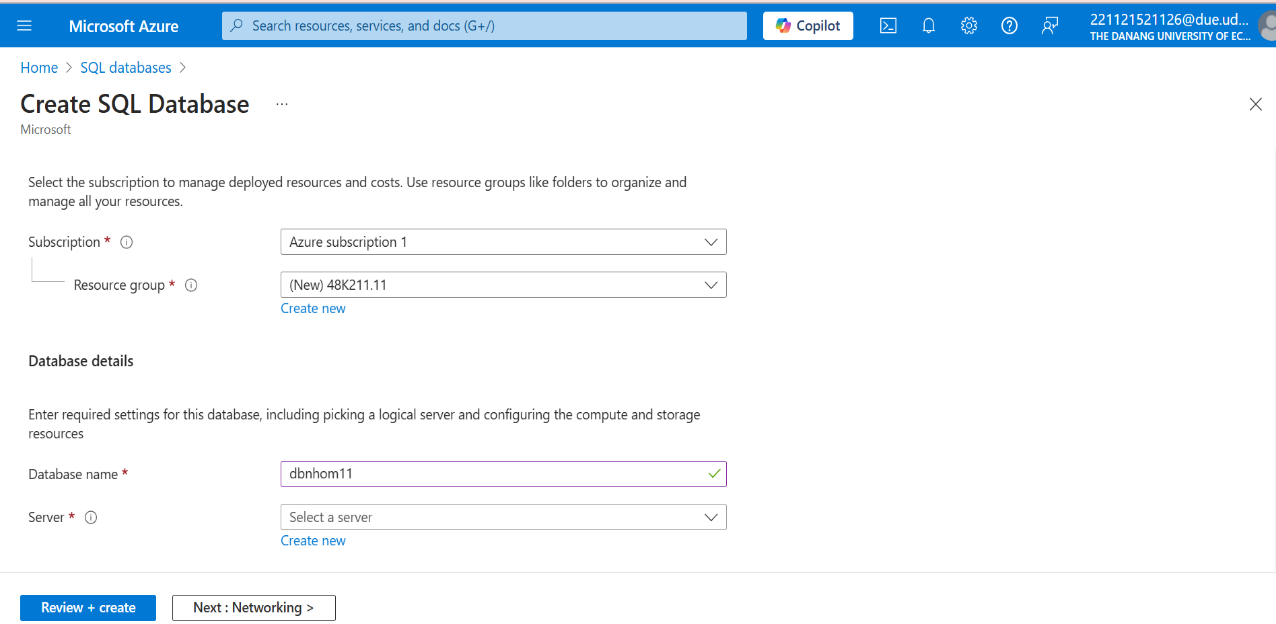
Nhóm đề xuất sử dụng **Microsoft Azure** để giải quyết vấn đề lượng dữ liệu lớn vượt quá khả năng lưu trữ của một ổ đĩa. Microsoft Azure là nền tảng điện toán đám mây do Microsoft cung cấp, cho phép người dùng truy cập và quản lý các dịch vụ, tài nguyên đám mây một cách linh hoạt và hiệu quả.

**Lý do lựa chọn Microsoft Azure:**

* **Chi phí hợp lý:** Phù hợp với các doanh nghiệp nhỏ, cung cấp giải pháp đám mây linh hoạt mà không đòi hỏi đầu tư lớn ban đầu.
* **Hỗ trợ đa nền tảng:** Azure hỗ trợ sao lưu và quản lý dữ liệu bằng hầu hết các ngôn ngữ lập trình và trên mọi hệ điều hành, giúp tích hợp dễ dàng vào hệ thống hiện tại.
* **An toàn và đáng tin cậy:**
* Dữ liệu được tự động lưu dưới dạng ba bản sao tại các khu vực khác nhau trong cùng trung tâm dữ liệu, đảm bảo khả năng truy cập ngay cả khi có sự cố cục bộ.
* Ngoài ra, Azure còn lưu thêm ba bản sao dự phòng tại một trung tâm dữ liệu từ xa, tăng cường mức độ an toàn và khả năng phục hồi dữ liệu.
* **Tùy chỉnh linh hoạt:** Microsoft Azure cho phép người dùng tùy chỉnh quy trình sao lưu tự động, phù hợp với nhu cầu sử dụng thực tế.

## Thực hiện triển khai trên SQL Server

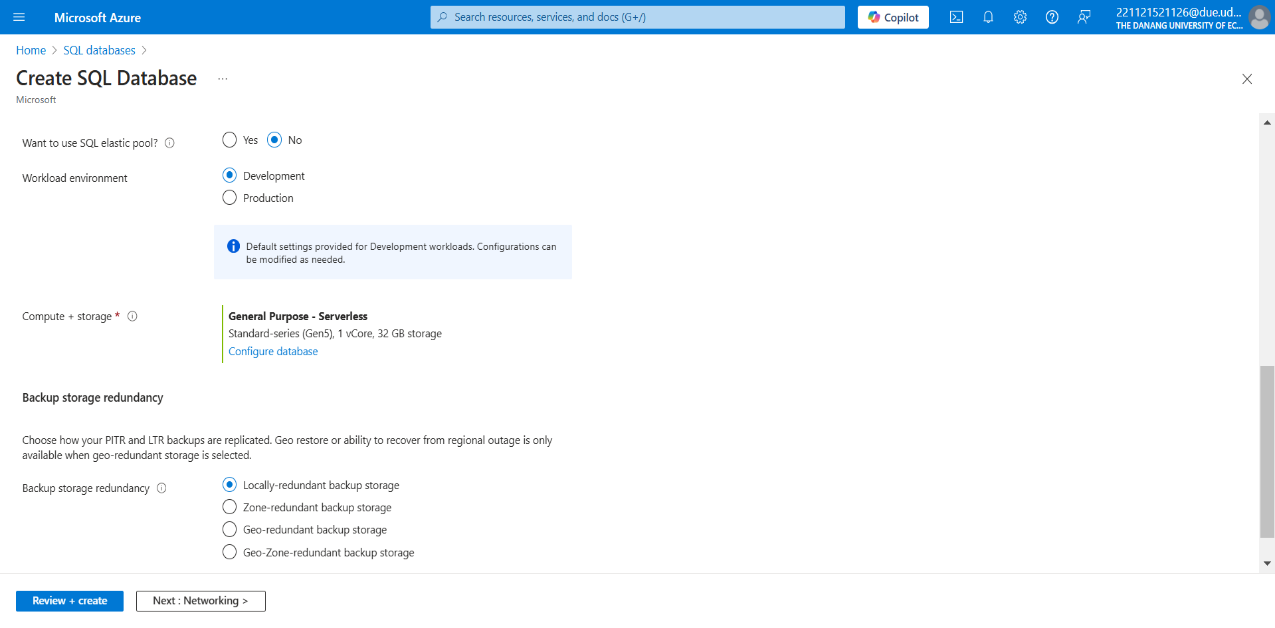
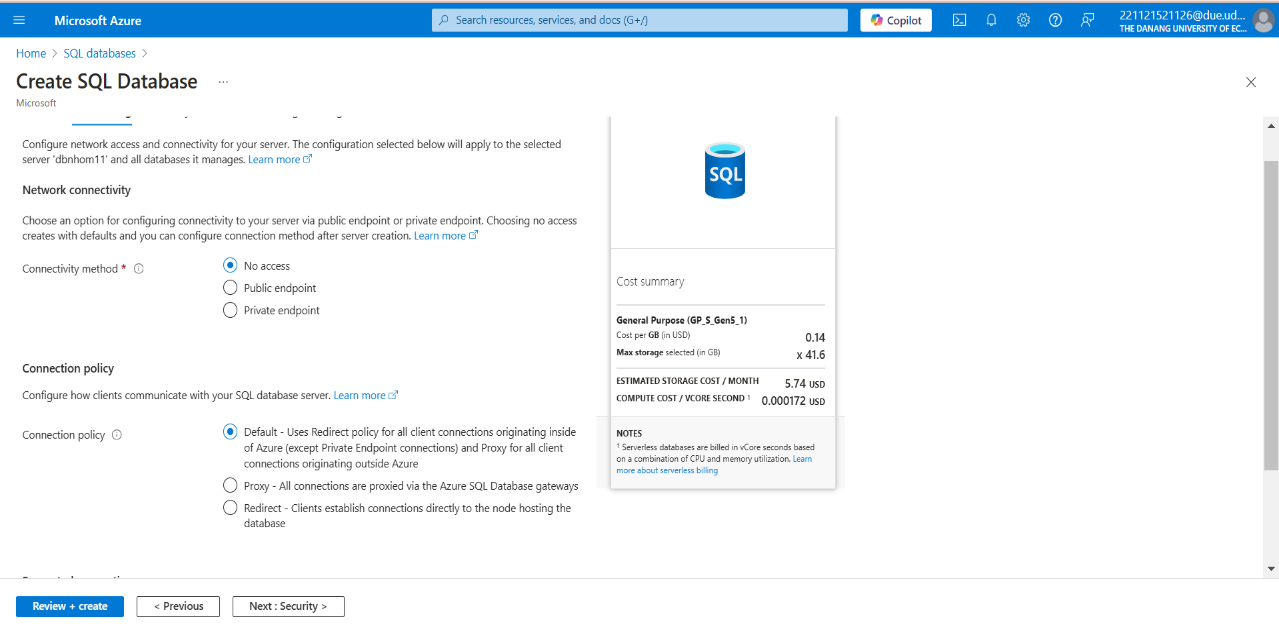
* **Bước 1:** Tạo tài khoản trên Microsoft Azure, sau đó chọn SQL Database → Create
* **Bước 2:**. Ở Resource group, chọn “Create new” để tạo nhóm (48K21.1), sau đó nhập tên cho cơ sở dữ liệu muốn tạo (dbnhom11)

Hình : Tạo SQL database trên Microsoft Azure

* **Bước 3:** Chọn **“Create new”** để tạo máy chủ SQL mới

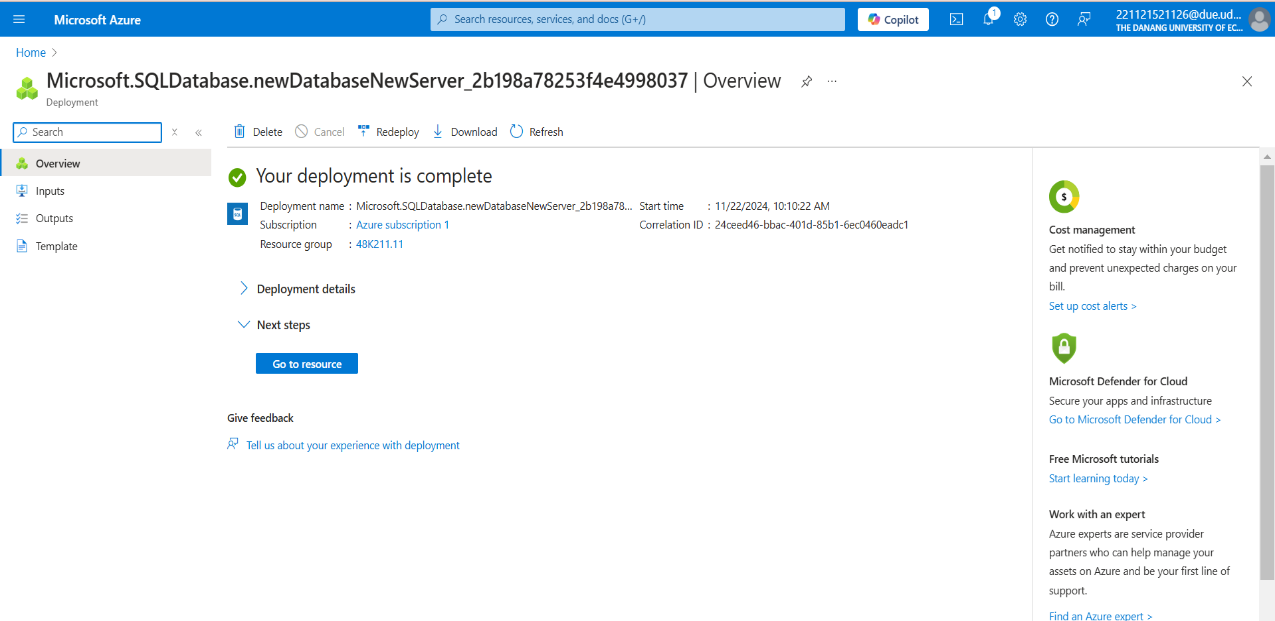
Hình 28: Tạo máy chủ SQL mới

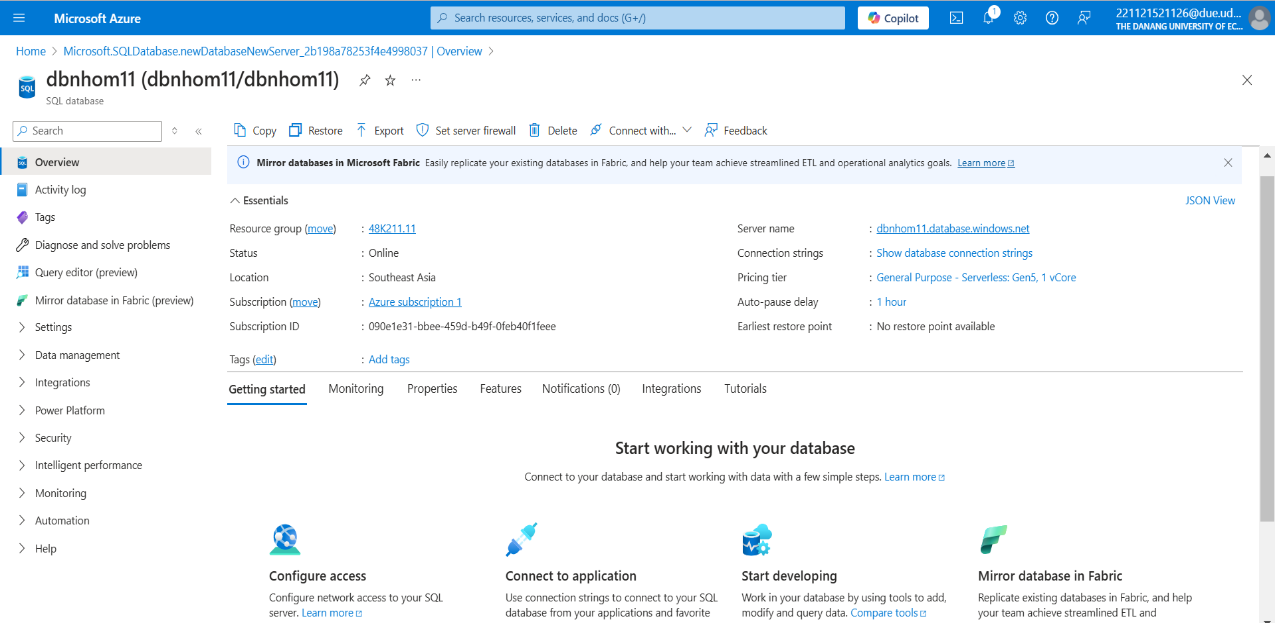
* **Bước 4:** Thiết lập các cấu hình cho SQL Database

Hình 29: Thiết lập các cấu hình cho SQL Database

Hình : Thiết lập các cấu hình cho SQL Database

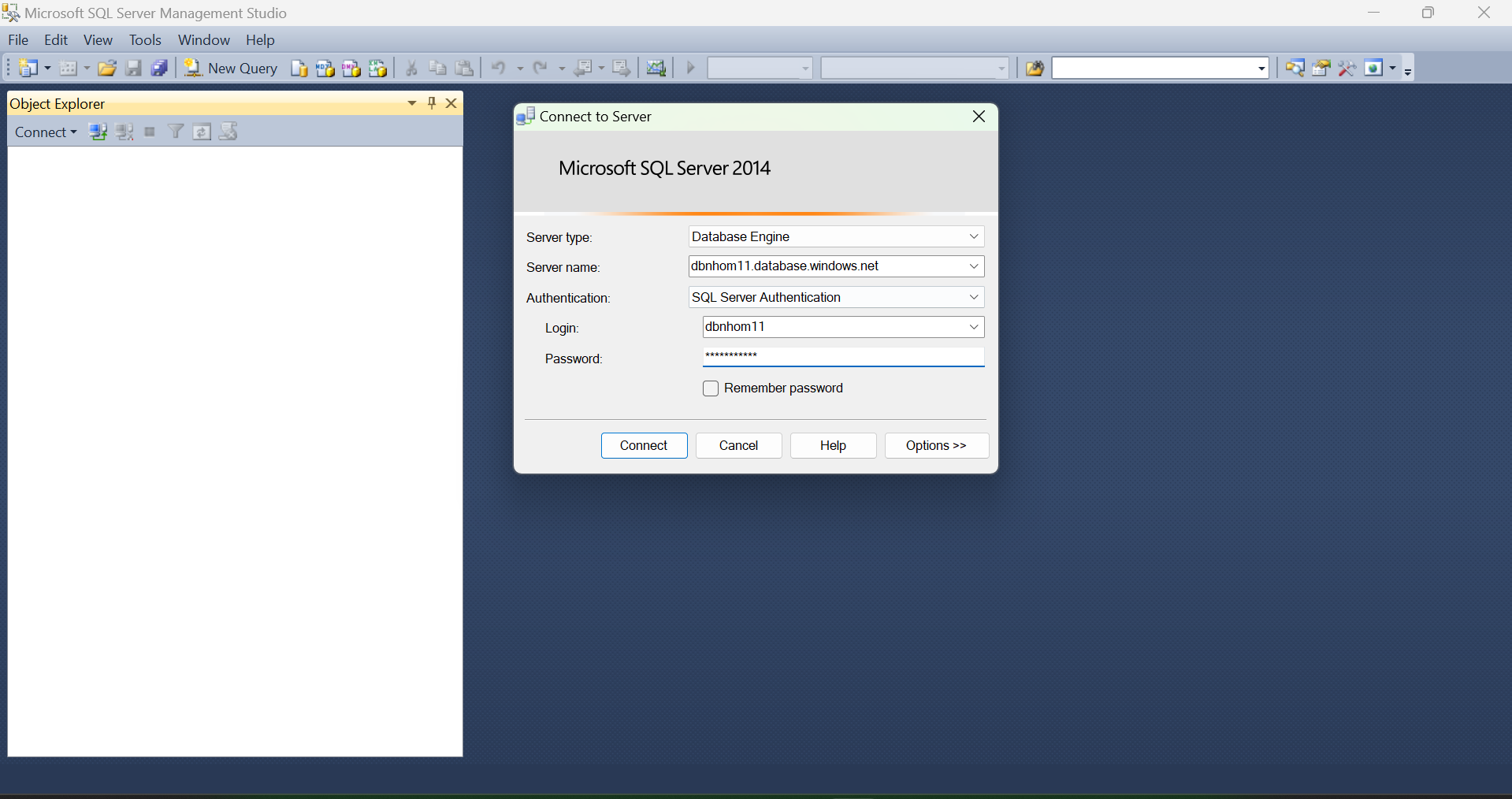
* **Bước 5:** Chọn “Review + create” để xem lại cấu hình trước khi tạo SQL Database, sau đó chọn “Create” để tạo SQL Database

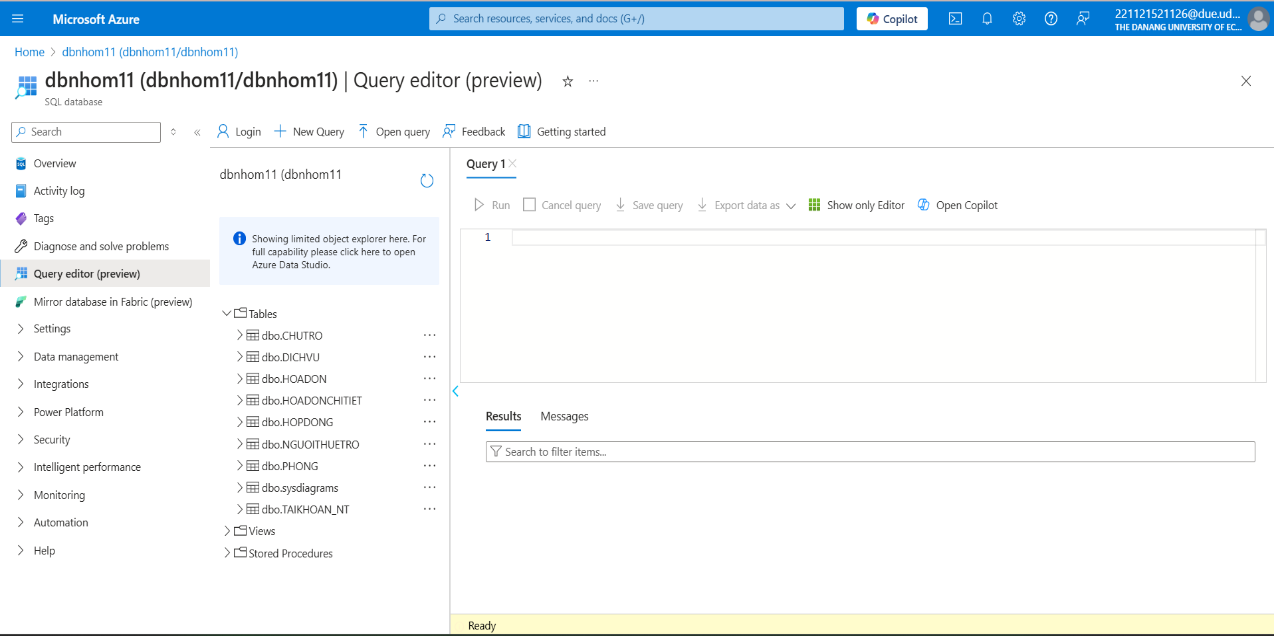
Hình 31: Tạo thàng công SQL Database

* **Bước 6:** Chọn vào Go to resource

Hình : Màn hình Database

* **Bước 7:** Sao chép link Server name để liên kết với SQL server

Hình : Liên kết SQL Server với Database vừa tạo trên Microsoft Azure

* **Bước 8:** Kiểm tra trên Microsoft Azure

Hình : Liên kết SQL Server thành công

# PHƯƠNG ÁN HẠN CHẾ TẤN CÔNG SQL INJECTION

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

* **Phương án:** Mã hóa mật khẩu và thiết lập thủ tục đăng nhập ngăn chặn SQL Injection
* **Cơ chế bảo mật:** Hệ thống sử dụng phương pháp mã hóa một chiều bằng thuật toán băm SHA-256 để bảo vệ mật khẩu.
* **Lý do lựa chọn:**

**Lưu trữ an toàn:**

* Thay vì lưu trữ trực tiếp mật khẩu, hệ thống sẽ lưu giá trị băm (hash) của mật khẩu được tạo ra bằng thuật toán SHA-256.
* Khi người dùng đăng nhập, hệ thống sẽ băm mật khẩu nhập vào bằng SHA-256 và so sánh với giá trị hash đã lưu trữ. Nếu giá trị khớp, quá trình đăng nhập được chấp nhận.
* Điều này đảm bảo rằng ngay cả khi cơ sở dữ liệu bị xâm nhập, kẻ tấn công không thể khôi phục mật khẩu gốc từ giá trị hash.

**Tăng cường độ an toàn:**

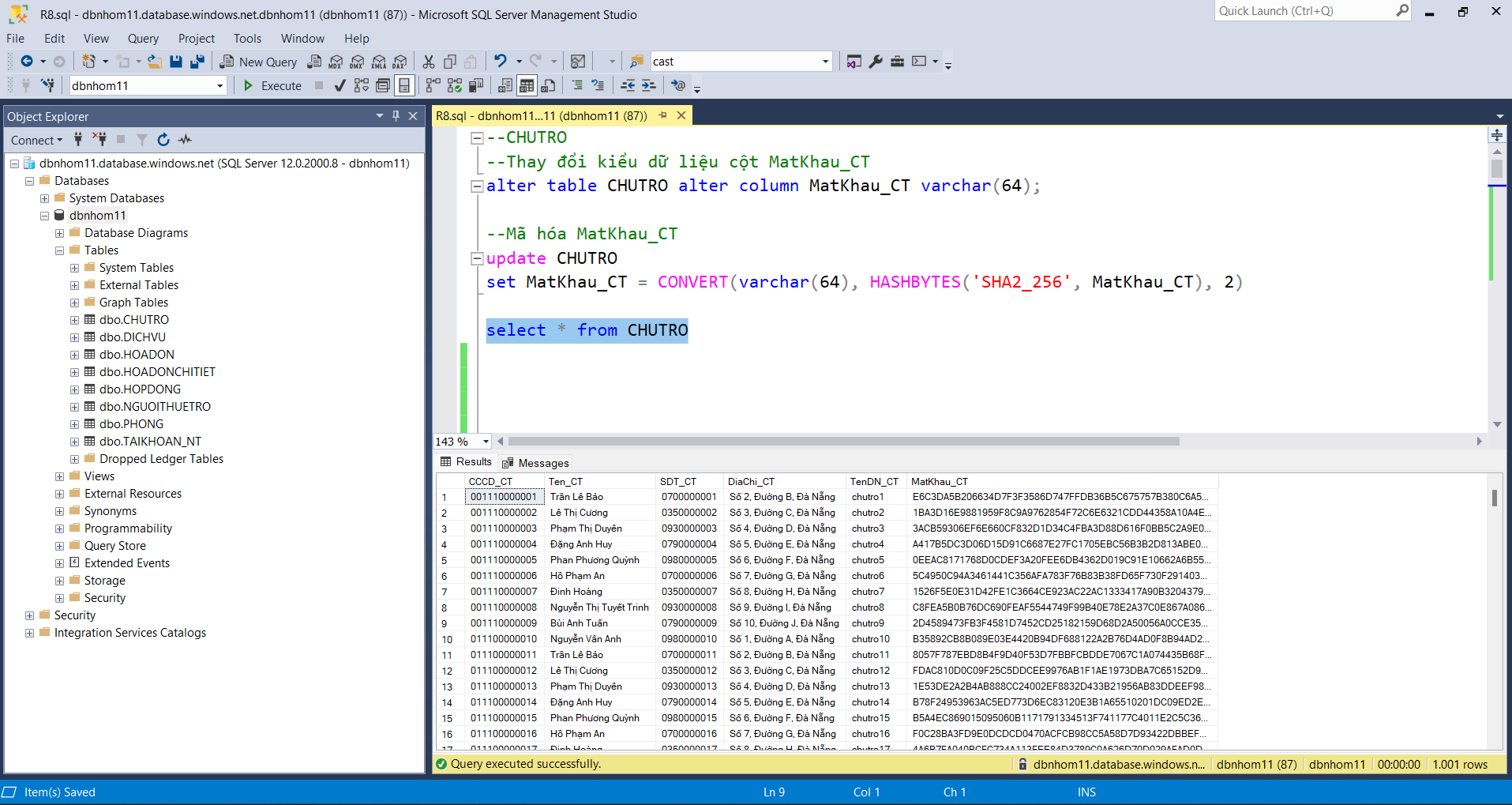
* SHA-256 là một trong những thuật toán mã hóa băm tiêu chuẩn hiện đại với độ dài hash lớn (256 bit), đảm bảo tính an toàn cao và khó bị tấn công bằng cách dò tìm.
* Thuật toán này được sử dụng rộng rãi trong các ứng dụng bảo mật yêu cầu độ an toàn cao như ngân hàng, hệ thống giao dịch trực tuyến, và cơ sở dữ liệu người dùng.

**Chống SQL Injection:**

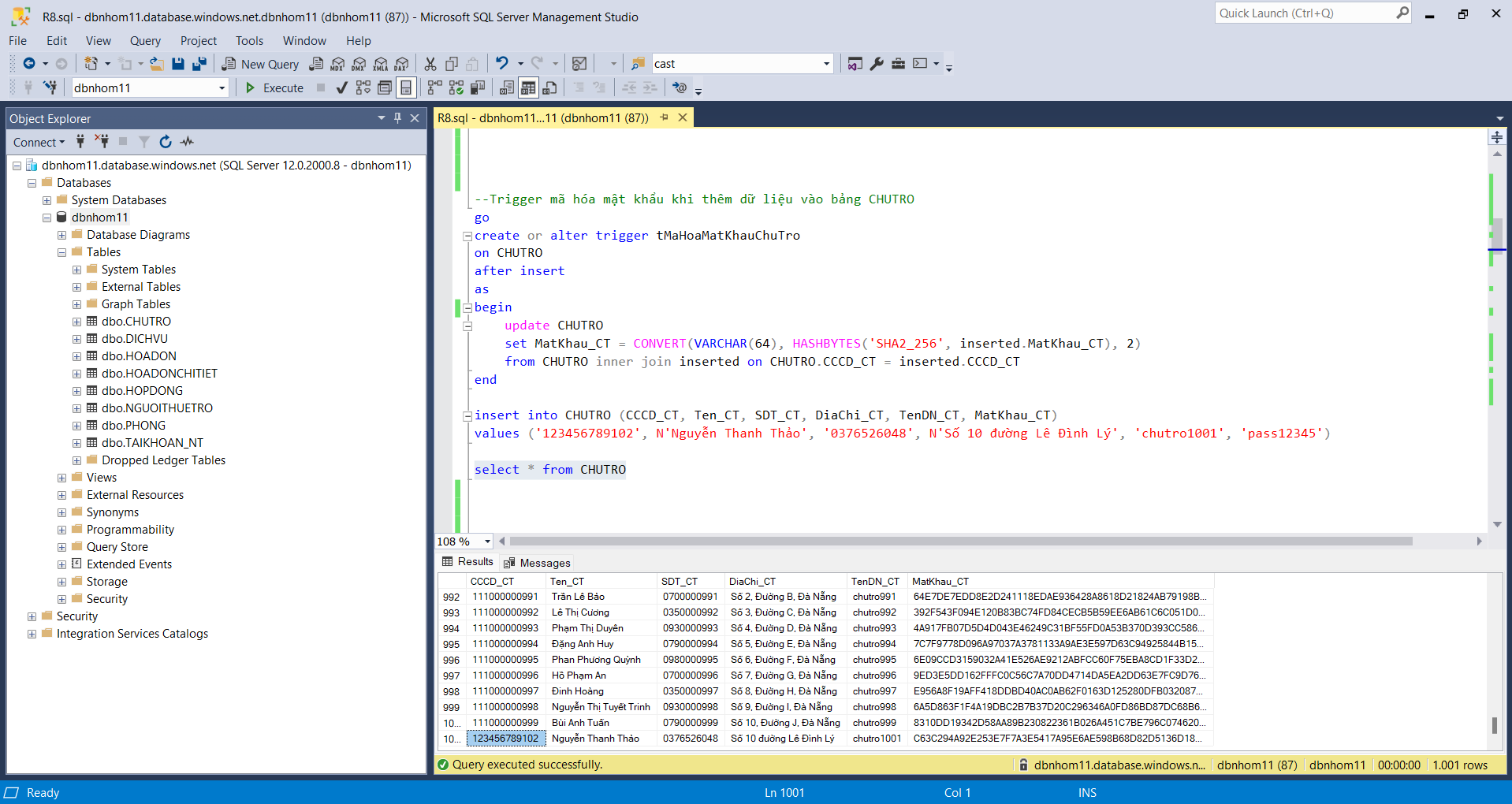
* Thủ tục đăng nhập được thiết kế sử dụng tham số hóa truy vấn thay vì nối chuỗi trực tiếp, giúp ngăn chặn các cuộc tấn công SQL Injection.
* Dữ liệu nhập vào từ người dùng được xử lý an toàn và không thể thay đổi cấu trúc câu lệnh SQL, bảo vệ hệ thống khỏi các lỗ hổng bảo mật phổ biến.

## Thực hiện phương án mã hóa mật khẩu và thiết lập thủ tục đăng nhập ngăn chặn SQL Injection

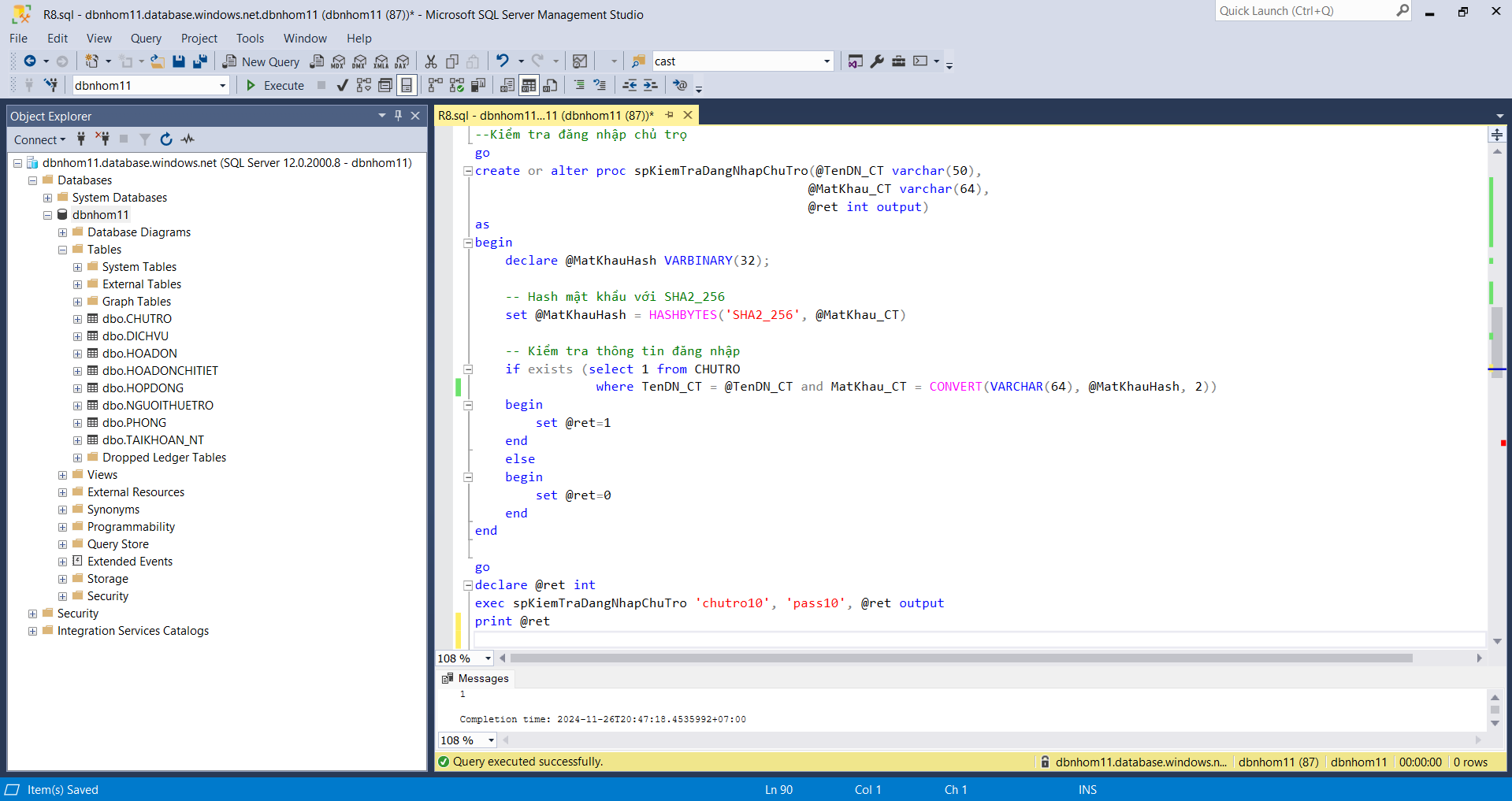
### Đối với bảng CHUTRO ta sẽ mã hóa cột MatKhau\_CT

* **Bước 1:** Thay đổi kiểu dữ liệu cột MatKhau\_CT thành varchar(64) để có đủ kích thước để chứa giá trị mã hóa. Sau đó mã hóa cột MatKhau\_CT

Hình : Mã hóa cột MatKhau\_CT

* **Bước 2:** Thiết lập trigger mã hóa cột MatKhau\_CT khi thêm mới dữ liệu vào bảng CHUTRO

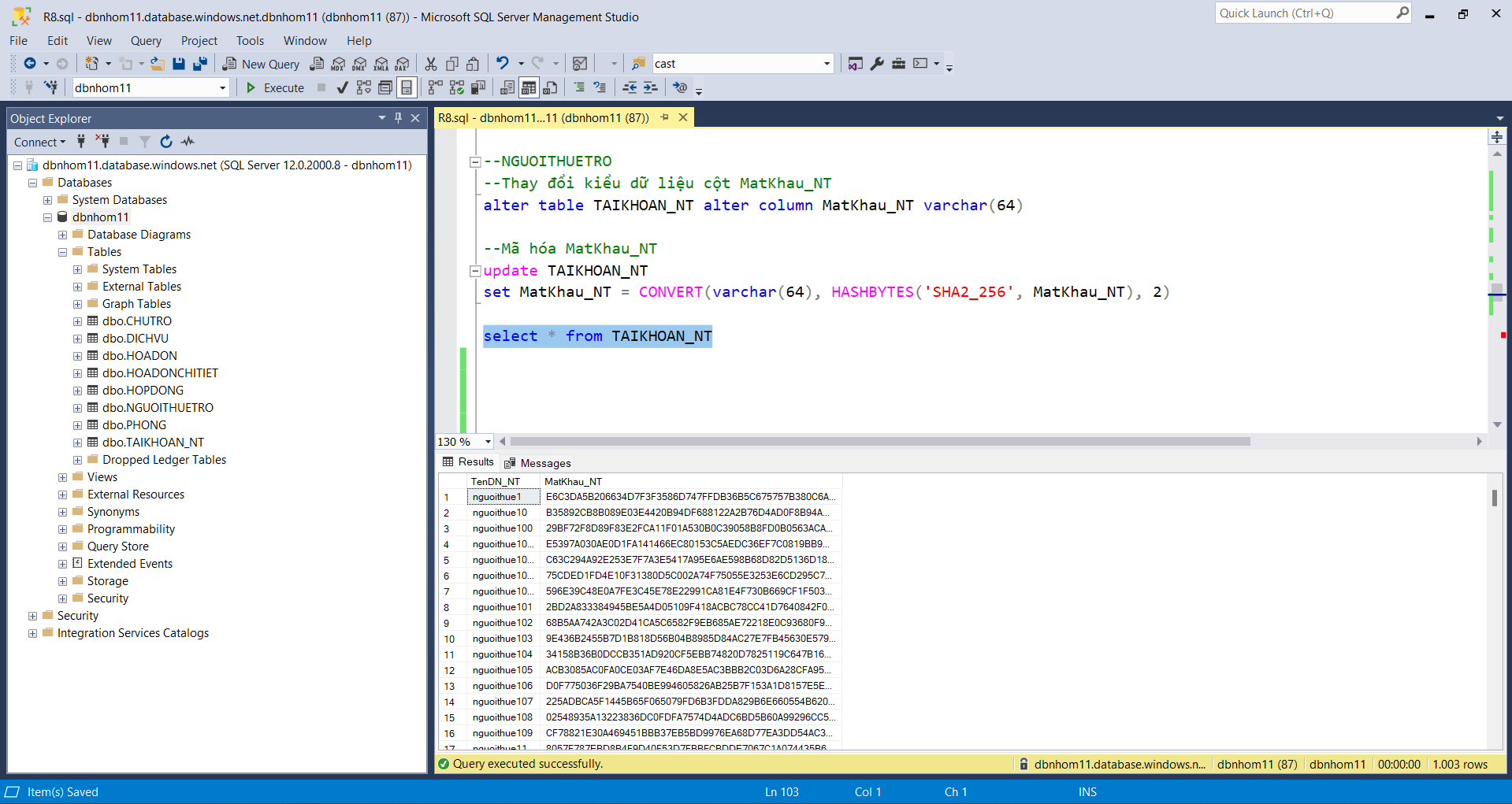
Hình : Trigger mã hóa cột MatKhau\_CT khi thêm mới dữ liệu vào bảng CHUTRO

* **Bước 3:** Thiết lập thủ tục kiểm tra đăng nhập cho tài khoản chủ trọ. Trả về 1 giá trị int (0: đăng nhập thất bại, 1: đăng nhập thành công)

Hình : Thủ tục kiểm tra đăng nhập cho tài khoản chủ trọ

### Đối với bảng TAIKHOAN\_NT ta sẽ mã hóa cột MatKhau\_NT

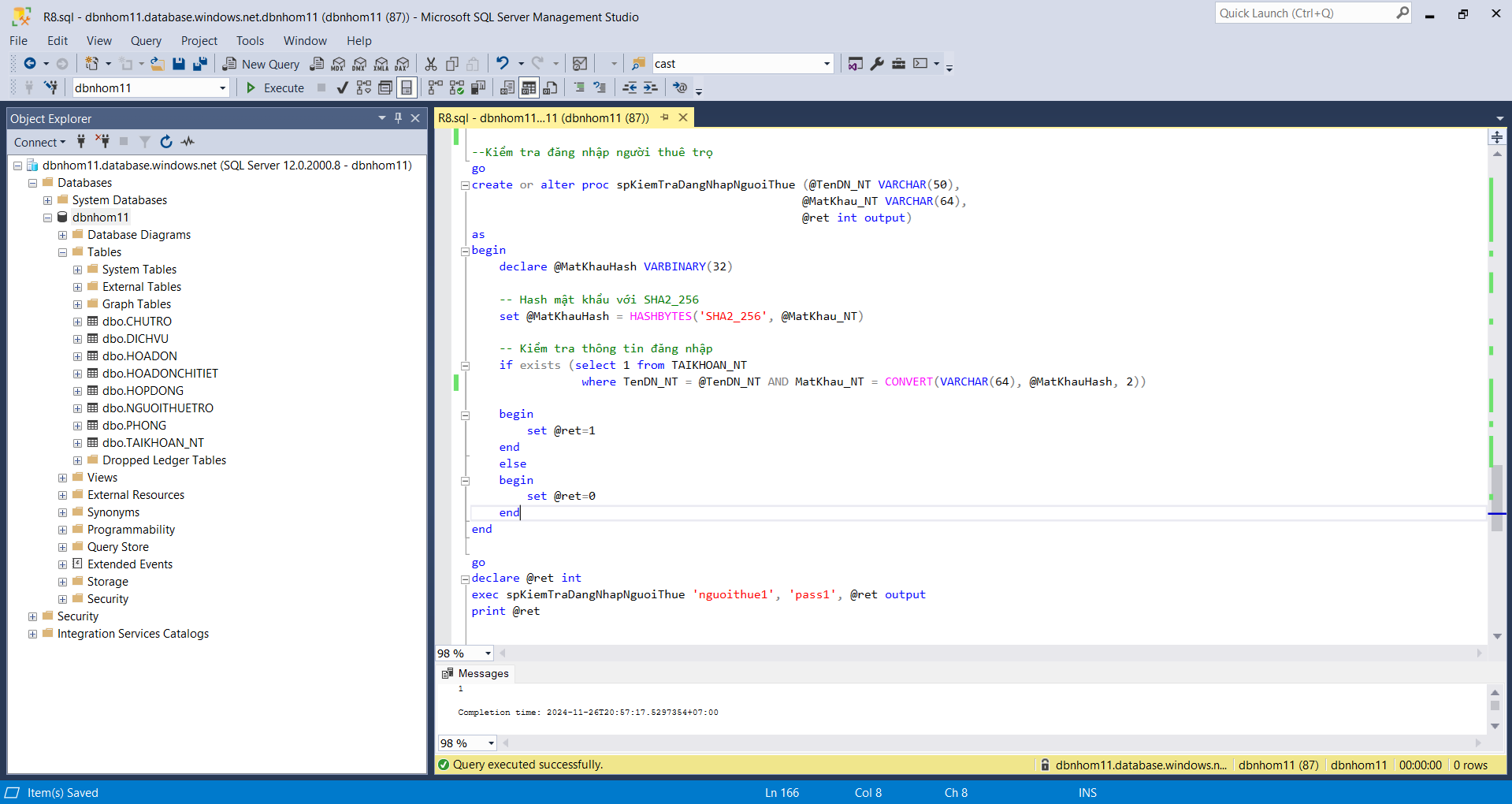
* **Bước 1:** Thay đổi kiểu dữ liệu cột MatKhau\_NT thành varchar(64) để có đủ kích thước để chứa giá trị mã hóa. Sau đó mã hóa cột MatKhau\_NT

Hình : Mã hóa cột MatKhau\_NT

* **Bước 2:** Thiết lập trigger mã hóa cột MatKhau\_NT khi thêm mới dữ liệu vào bảng TAIKHOAN\_NT

Hình : Trigger mã hóa cột MatKhau\_NT khi thêm mới dữ liệu vào bảng TAIKHOAN\_NT

* **Bước 3:** Thiết lập thủ tục kiểm tra đăng nhập cho tài khoản người thuê trọ. Trả về 1 giá trị int (0: đăng nhập thất bại, 1: đăng nhập thành công)

Hình : Thủ tục kiểm tra đăng nhập cho tài khoản người thuê trọ

**SỬA ĐỔI THIẾT KẾ**

* Bỏ ràng buộc not null của cột ThanhTien bảng HOADON và cột TongCong bảng HOADONCHITIET để không cần nhập dữ liệu mà tính từ những cột khác trong các bảng
* Đổi kiểu dữ liệu cột LoaiPhong của bảng PHONG thành char(1)