XÂY DỰNG CƠ CHẾ BẢO MẬT CHO CƠ SỞ DỮ LIỆU

# QUẢN LÝ NGƯỜI DÙNG

### **Xác thực người dùng:**

Bảo mật cơ sở dữ liệu SQL  Server giúp phân quyền người dùng trên cơ sở dữ liệu. SQL Server xác thực đăng nhập với 8 cơ chế, bao gồm:

* Windows Authentication
* SQL Server Authentication
* Microsoft Entra MFA
* Microsoft Entra Password
* Microsoft Entra Integrated
* Microsoft Entra Service Principal
* Microsoft Entra Managed Identity
* Microsoft Entra Default

Trong đó 2 cơ chế thông dụng nhất đó chính là Windows Authentication và SQL Server Authentication:

* Windows Authentication: sử dụng user login Windows trên chính máy tính cài đặt SQL Server. Windows User bao gồm cả [local user](https://www.engisv.info/?p=81) và [domain user](https://www.engisv.info/?p=3868) nếu máy tính cài đặt SQL server có [join domain](https://www.engisv.info/?p=88)
* SQL Server Authentication: Là tài khoản được tạo và quản lý bởi chính SQL Server.
  + (1): Tài khoản chứng thực Windows (local user)
  + (2): Tài khoản chứng thực của Windows (domain user)
  + (3): Tài khoản của SQL Server

Ở đây nhóm quyết định chọn cơ chế xác thực người dùng SQL Server Authentication vì nhận thấy một số ưu điểm như:

* + Cho phép SQL Server hỗ trợ các ứng dụng cũ hơn và các ứng dụng do bên thứ ba cung cấp yêu cầu Xác thực SQL Server
  + Cho phép SQL Server hỗ trợ các môi trường có hệ điều hành hỗn hợp, trong đó không phải tất cả người dùng đều được xác thực bởi miền Windows
  + Cho phép người dùng kết nối từ các miền không xác định hoặc không đáng tin cậy. Ví dụ: ứng dụng nơi khách hàng đã thiết lập kết nối với thông tin đăng nhập SQL Server được chỉ định để nhận trạng thái đơn hàng của họ
  + Cho phép SQL Server hỗ trợ các ứng dụng dựa trên web, nơi người dùng có thể tạo danh tính riêng của mình
  + Cho phép các nhà phát triển phần mềm phân phối ứng dụng của họ bằng cách sử dụng hệ thống phân cấp quyền phức tạp dựa trên thông tin đăng nhập SQL Server được cài đặt sẵn và đã biết

### **Login:**

### Để kết nối tới cơ sở dữ liệu, cần thiết lập một tài khoản. Tuy nhiên khi thực hiện tạo một server trên portal.azure đã lựa chọn cơ chế SQL Authentication và đã thực hiện tạo tài khoản login ‘admindkt’ ngay khi thiết lập. Vì vậy đã tồn tại tài khoản login là ‘admindkt’.

### **KỊCH BẢN PHÂN QUYỀN**

* 1. Phân quyền trên server:

|  |  |
| --- | --- |
| **Login** | **Quyền Server** |
| **Quyền** |
| admindkt | MS\_DatabaseConnector (Quản lý toàn bộ cơ sở dữ liệu)  MS\_DatabaseManager (Quản lý cơ sở dữ liệu)  MS\_ServerStateReader (Đọc trạng thái máy chủ)  MS\_DefinitionReader (Cấp quyền đọc định nghĩa của các đối tượng trong cơ sở dữ liệu)  MS\_LoginManager (Quản lý tài khoản đăng nhập)  MS\_SecurityDefinitionReader (cấp quyền đọc các định nghĩa bảo mật)  MS\_ServerStateManager (quản lý trạng thái máy chủ) |

**Bảng 1** Kịch bản phân quyền trên server

1. Phân quyền trên database:

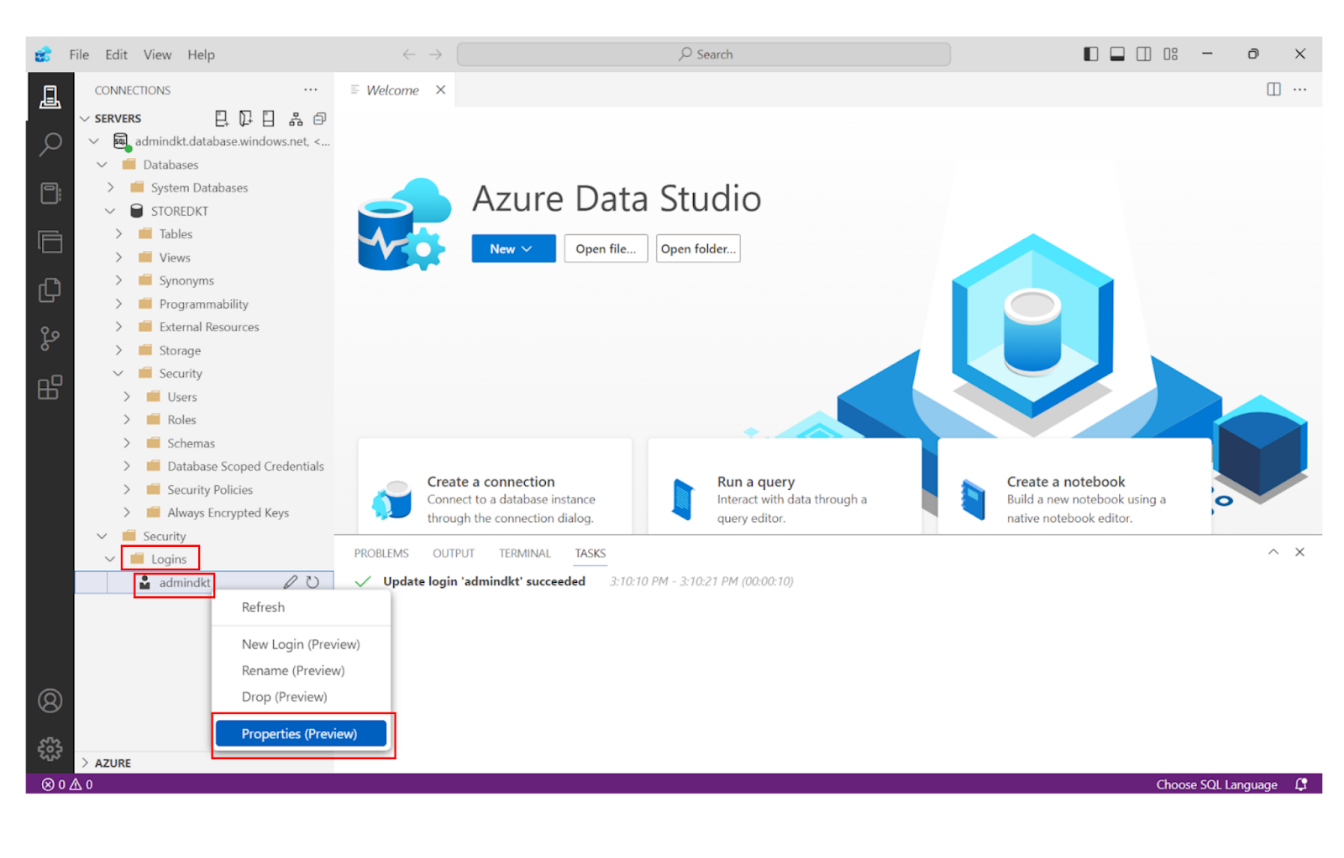
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Login** | **Users** | **Quyền Databases** | | **Quyền bảng** | |
| **Quyền** | **Database áp dụng** | **Quyền** | **Bảng áp dụng** |
| admindkt | dbo | db\_owner  (Có tất cả các quyền đối với CSDL) | STOREDKT | Tất cả | Tất cả các bảng |

**Bảng 2.** Kịch bản phân quyền trên database

### **Các bước thực hiện**

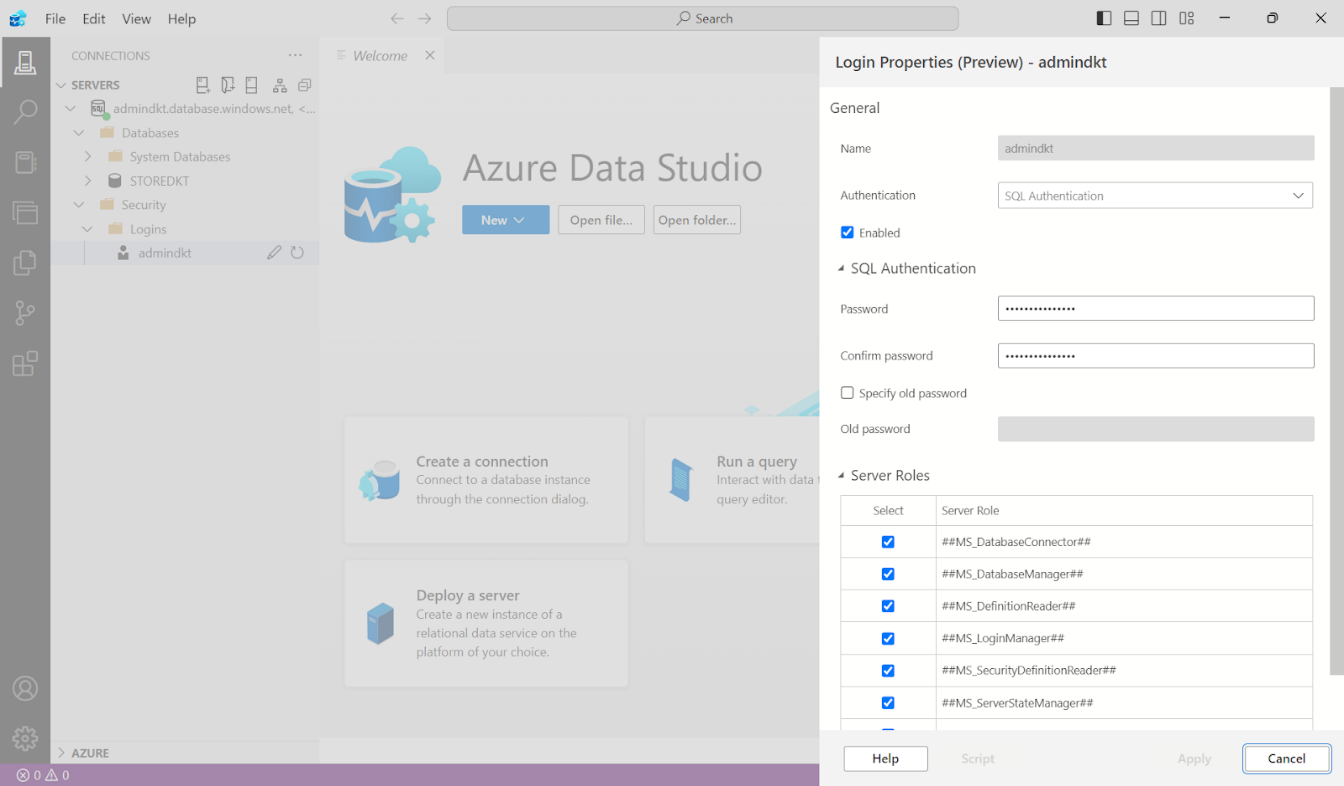
1. Phân quyền trên server:

Bước 1: Chọn Security → Chọn Logins → Chọn admindkt → Nhấn chuột phải → Chọn Properties

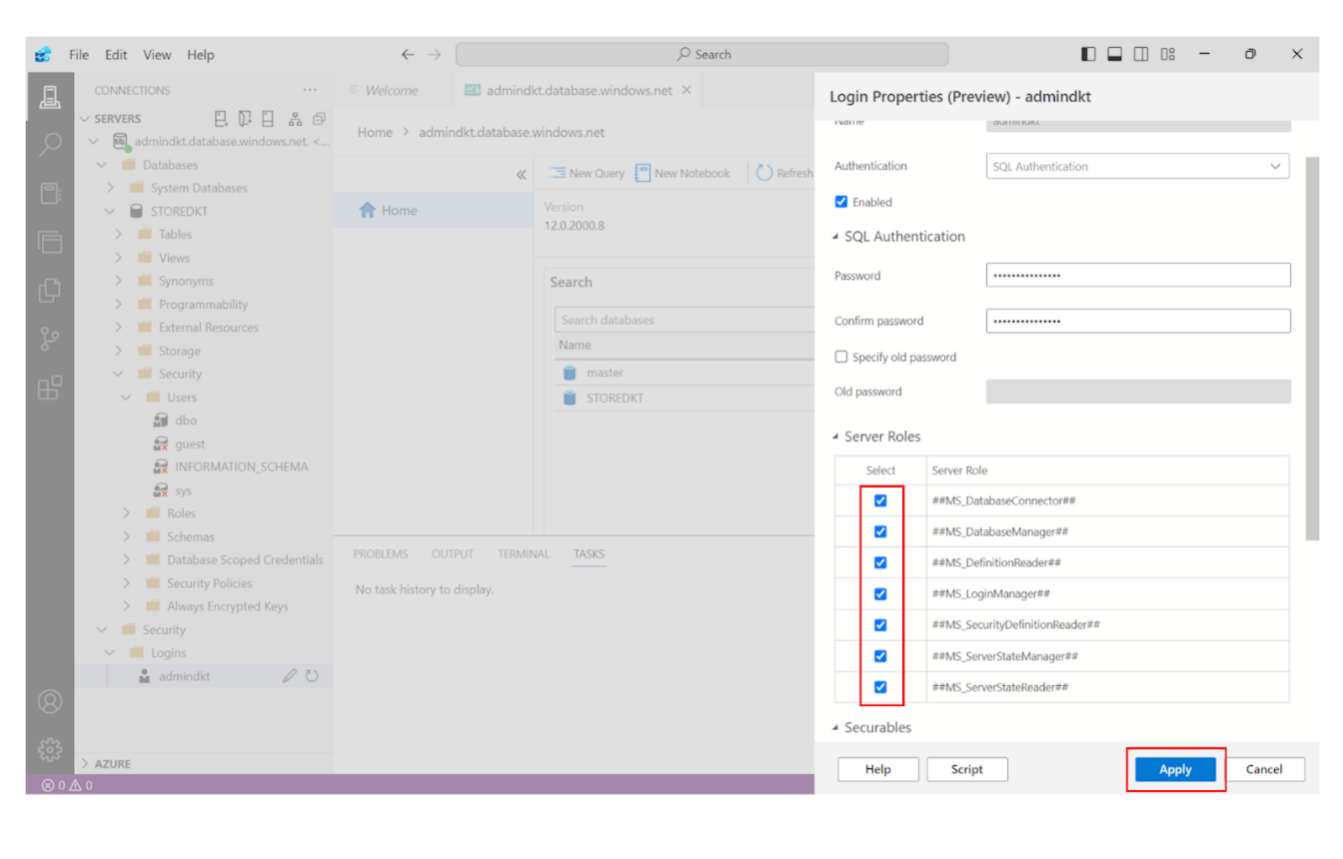


**Hình 1**. Demo phân quyền trên server

Bước 2: Thực hiện chọn tất cả các quyền có trong Server Roles → Chọn Apply

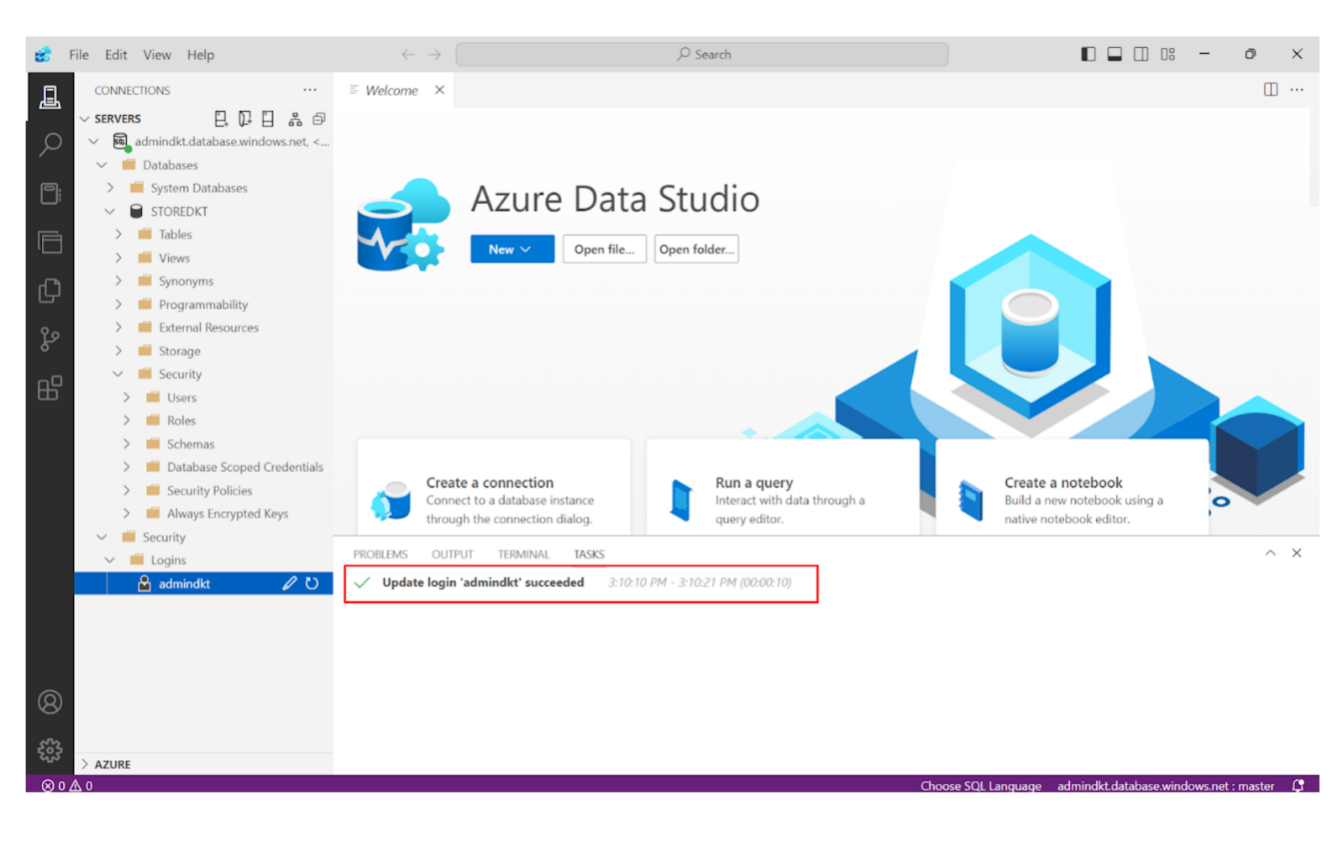


**Hình 2**. Demo phân quyền trên server



**Hình 3**. Demo phân quyền trên server

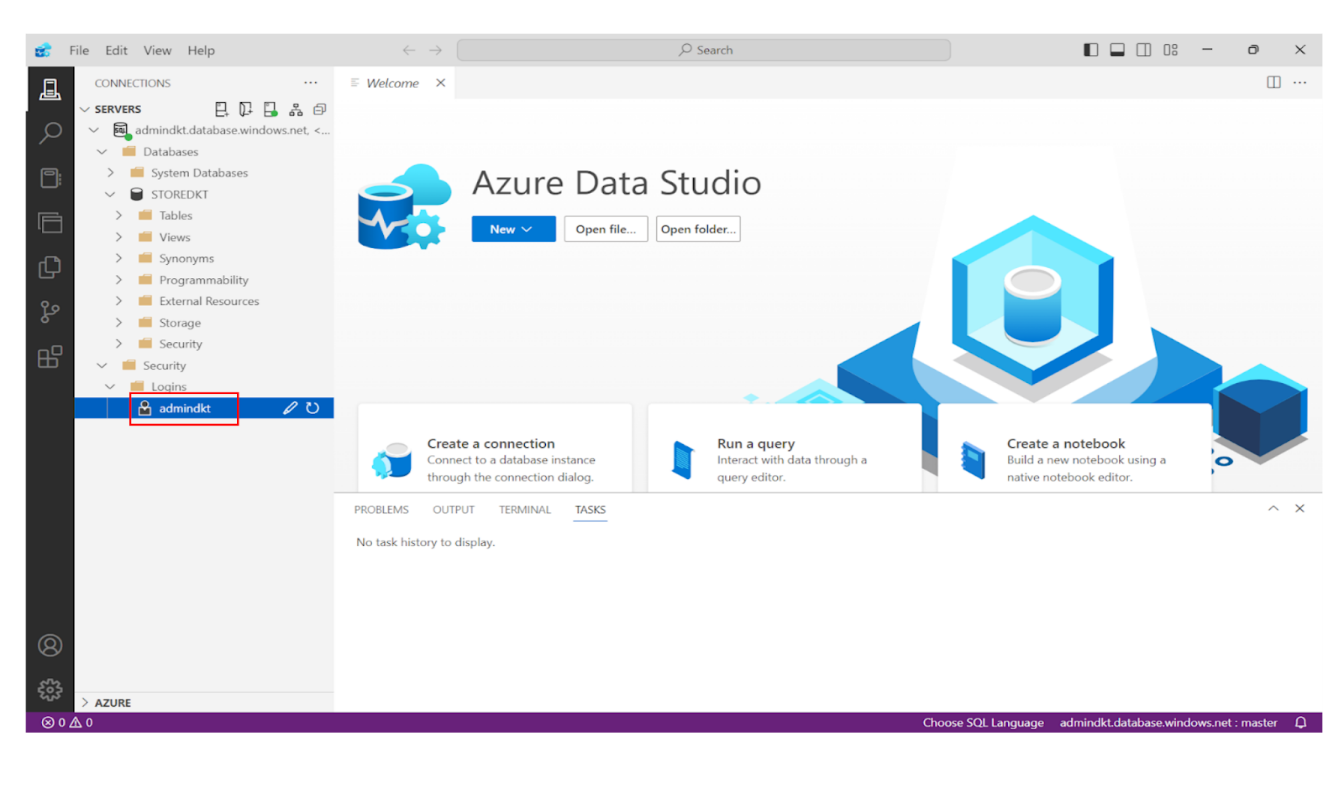
* + - * Màn hình thông báo update login “admindkt” thành công



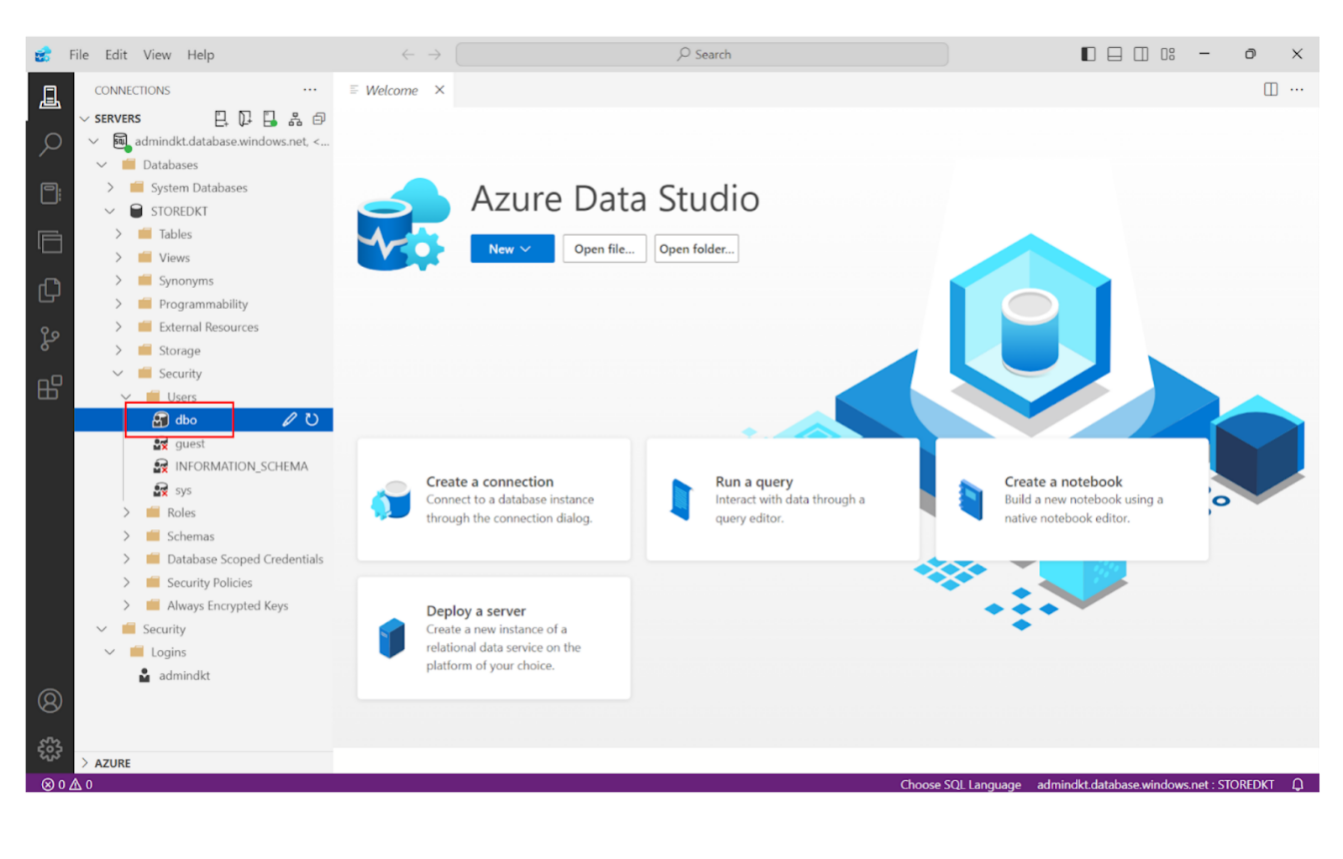
**Hình 4.** Demo phân quyền trên server

1. Phân quyền trên database:

dbo (database owner) là người dùng mặc định có quyền sở hữu toàn bộ các đối tượng trong cơ sở dữ liệu khi được tạo mới →"admindkt" đã được liên kết với users "dbo" → Thực hiện chỉnh sửa phân quyền admindkt trên users “dbo”

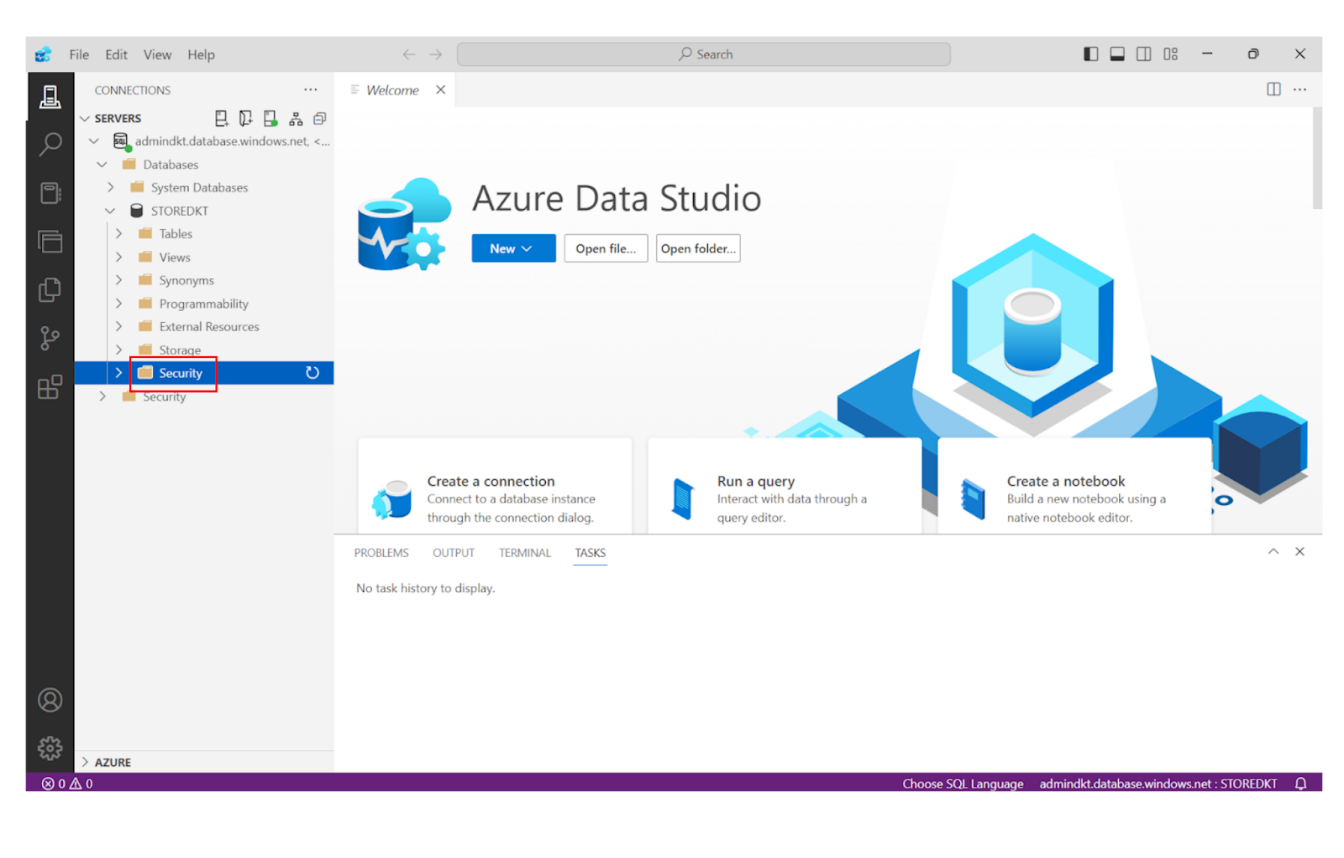


**Hình 5.** Demo phân quyền trên database



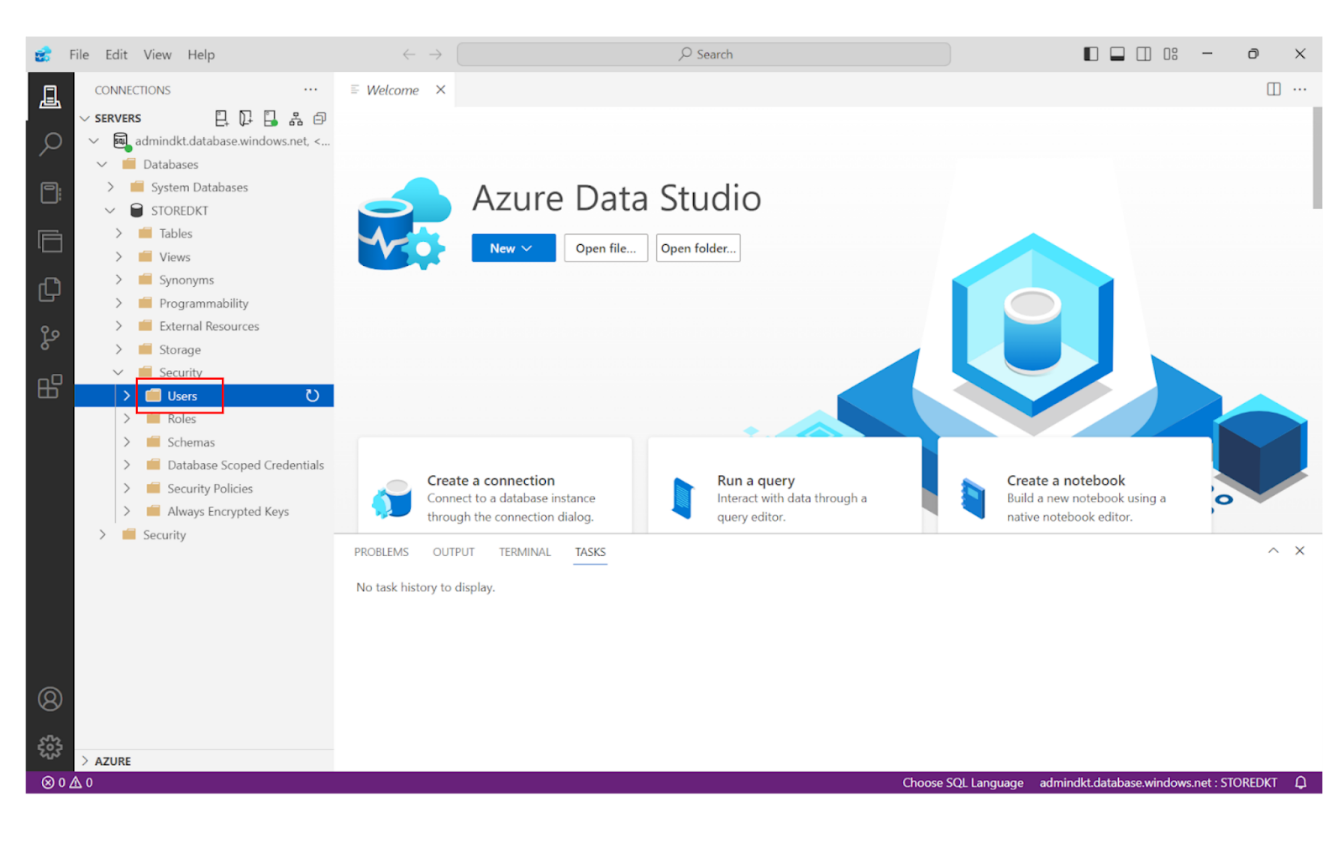
**Hình 6.** Demo phân quyền trên database

Bước 1: Trong database STOREDKT → Chọn Security



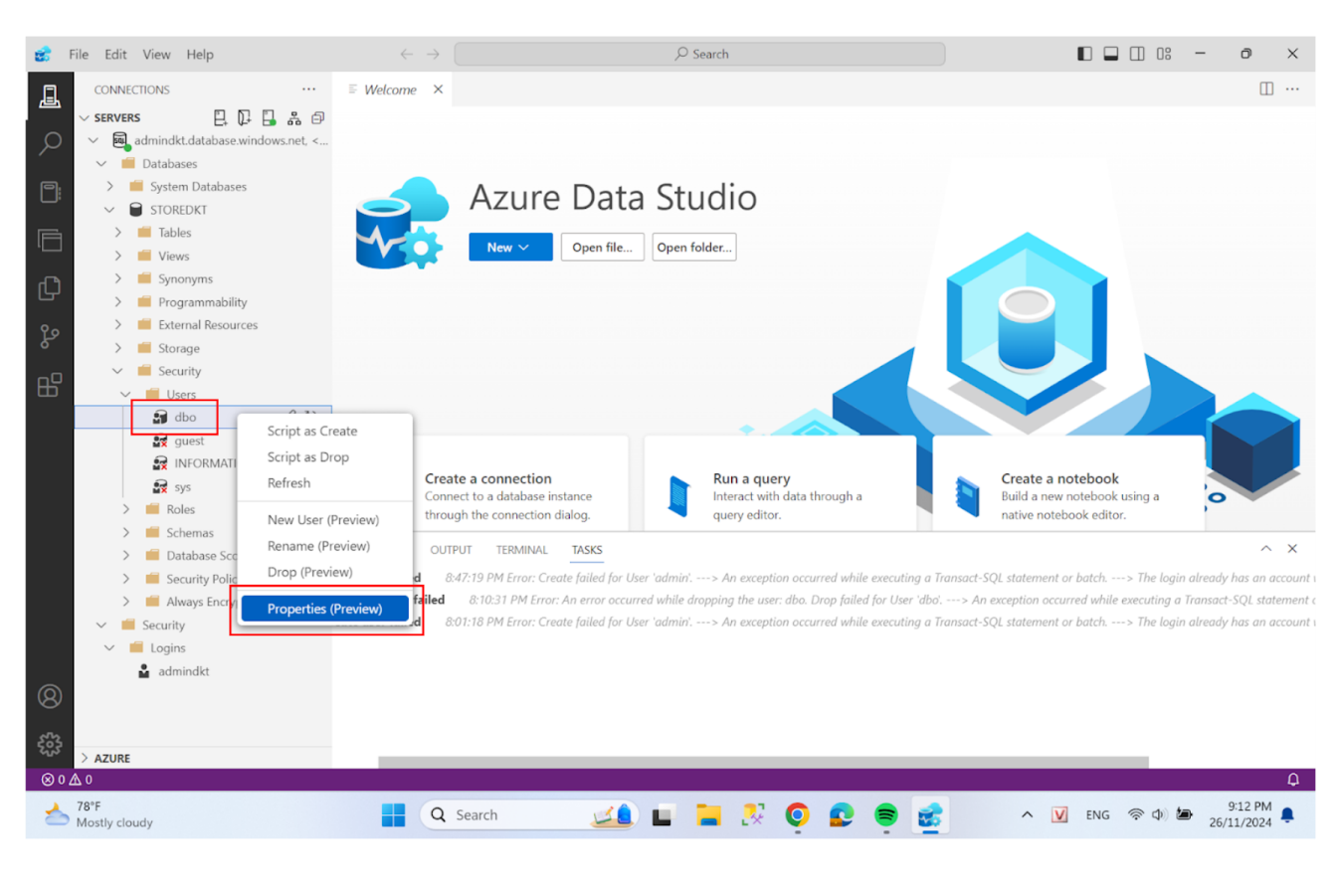
**Hình 7.** Demo phân quyền trên database

Bước 2: Sau khi chọn Security → Chọn Users



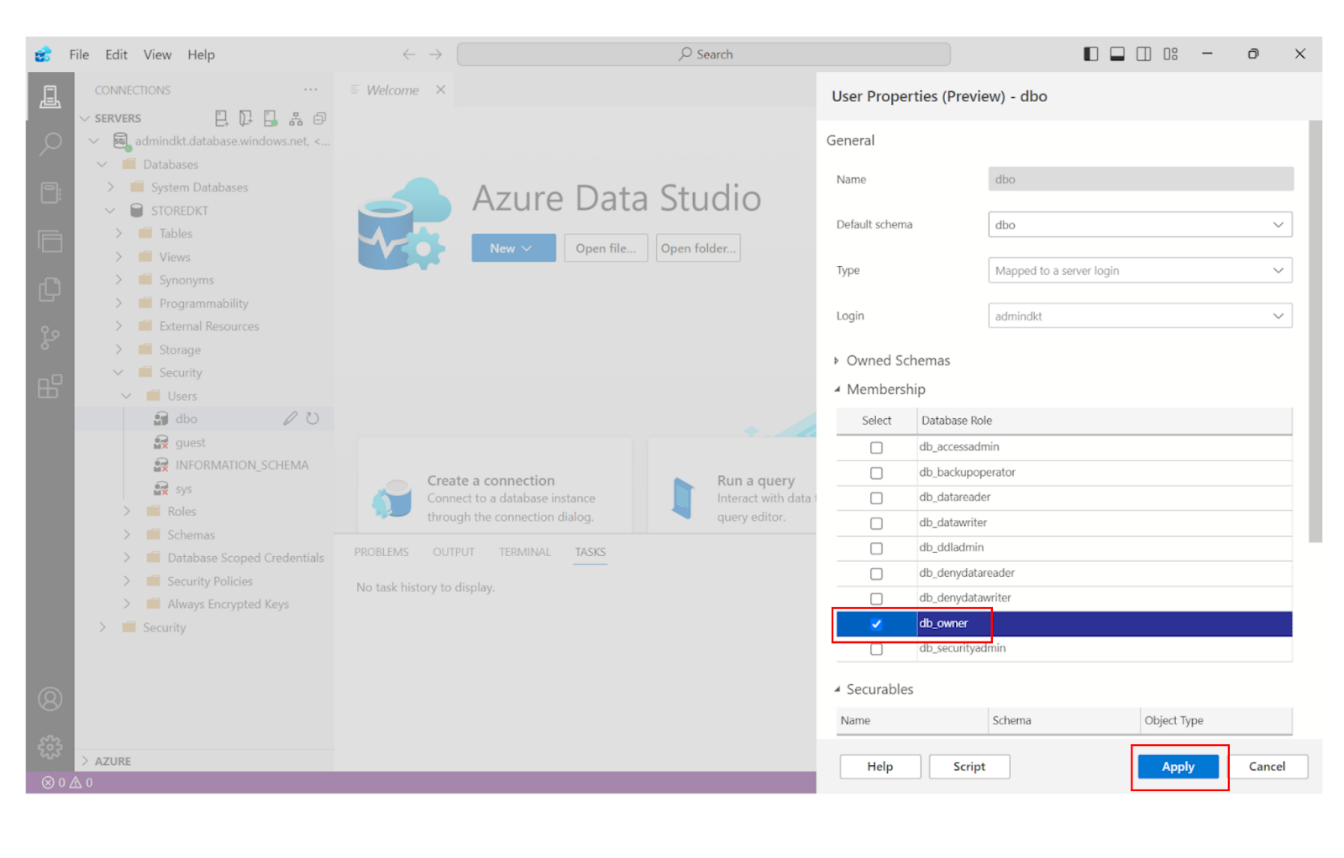
**Hình 8**. Demo phân quyền trên database

Bước 3: Chọn dbo → Kích chuột phải → Chọn Properties



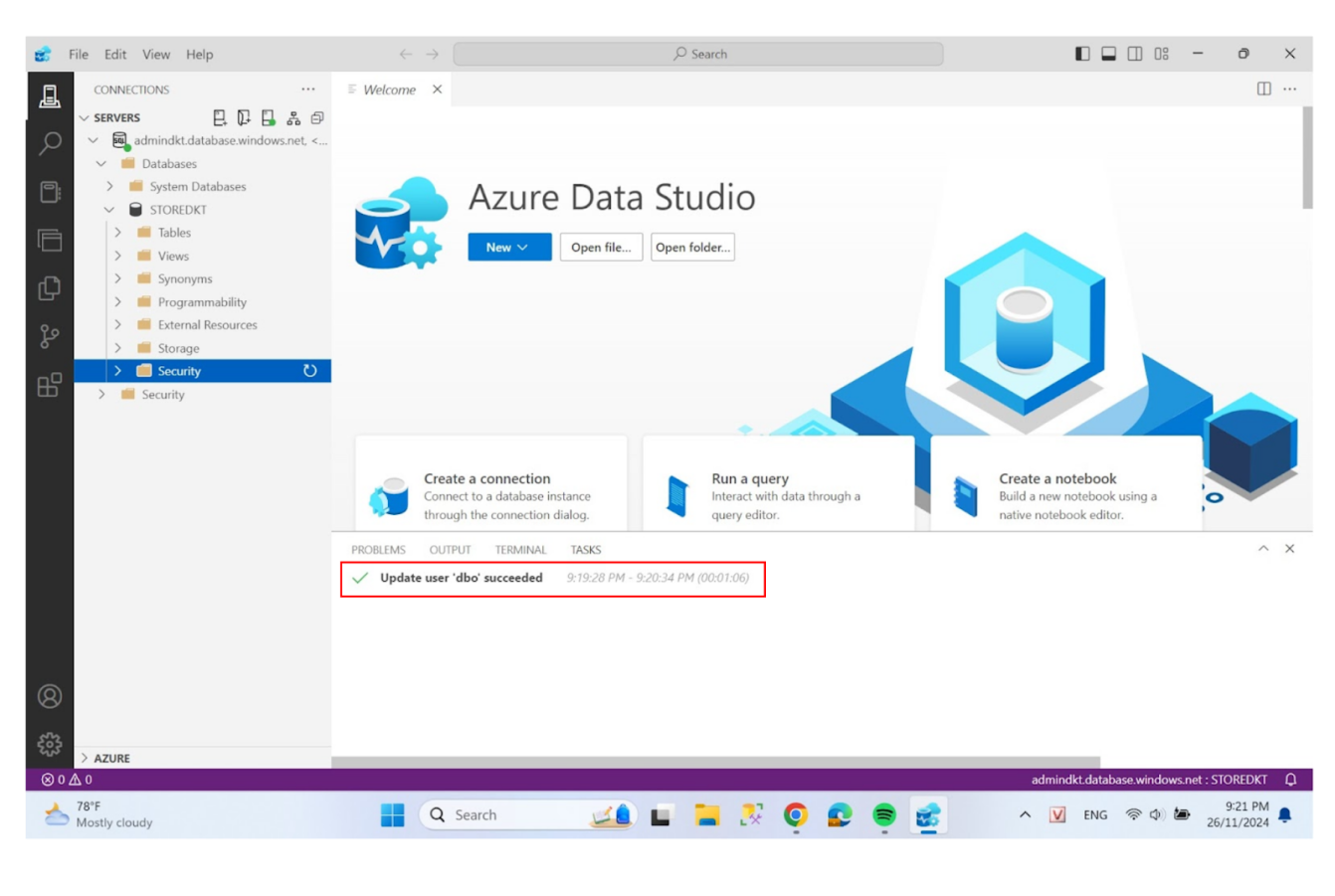
**Hình 9.** Demo phân quyền trên database

Bước 4: Tại Membership → Chọn db\_owner → Chọn Apply



**Hình 10.** Demo phân quyền trên database

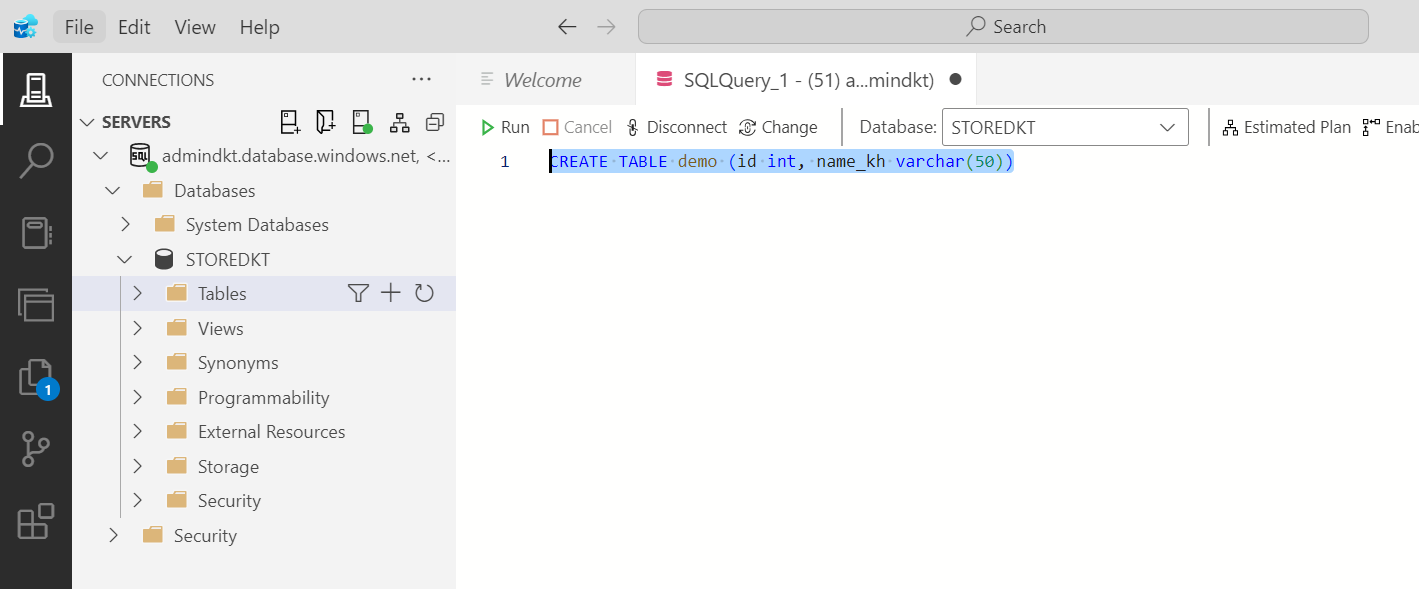
* + - * Màn hình cập nhật phân quyền users ‘dbo’ thành công



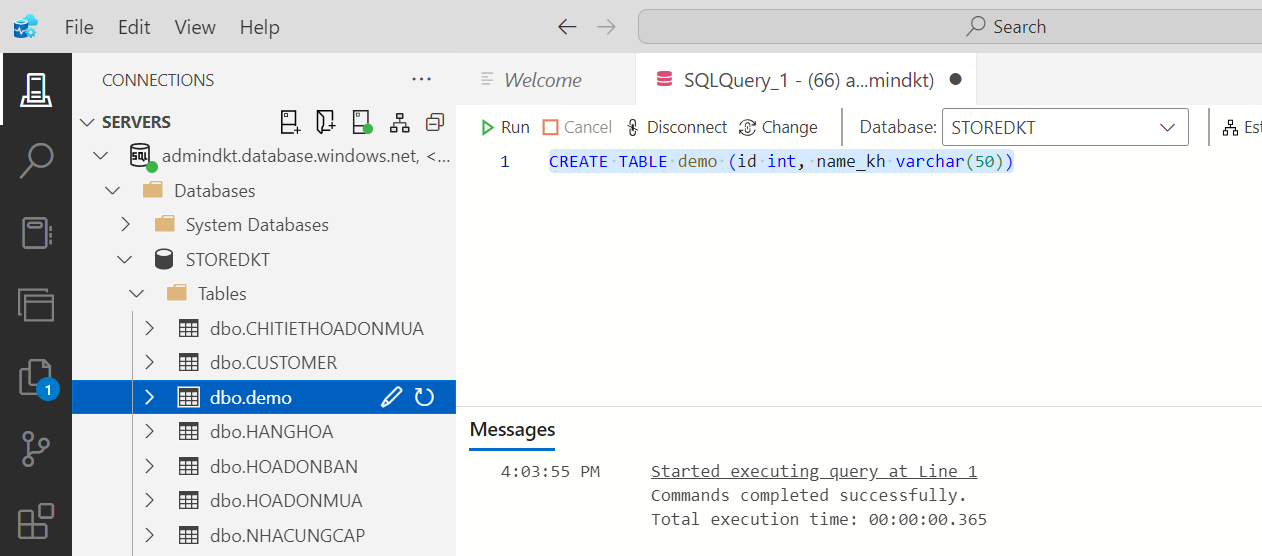
**Hình 11**. Demo phân quyền trên database

Bước 5: Kiểm tra phân quyền

1. Thực hiện tạo bảng mới -> Màn hình tạo bảng thành công

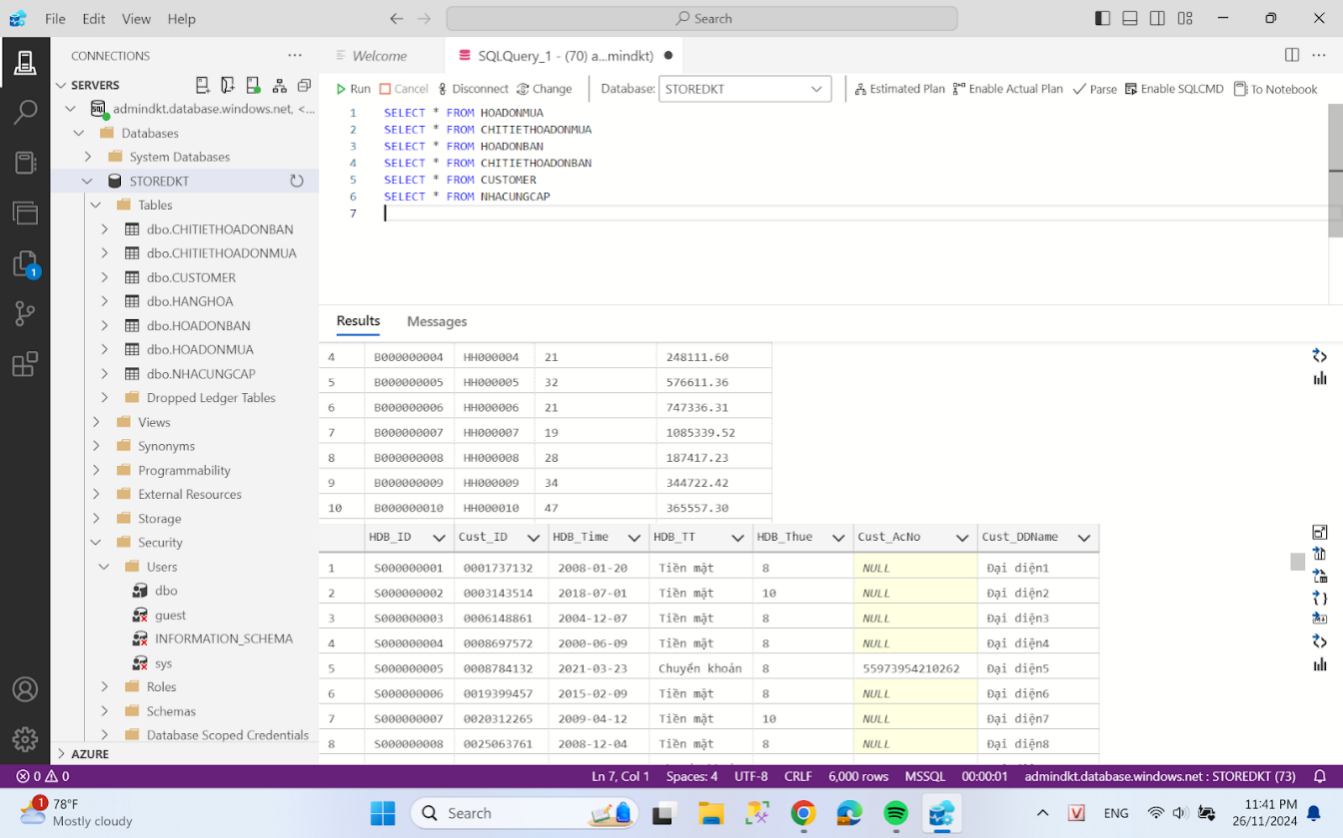
****

**Hình 12.** Demo phân quyền trên database

****

**Hình 13.** Demo phân quyền

* 1. Thực hiện lệnh Select -> Kết quả thực hiện lệnh Select thành công



**Hình 14.** Demo phân quyền

# BẢO MẬT DỮ LIỆU:

Trong cơ sở dữ liệu, bảo vệ thông tin nhạy cảm là yếu tố quan trọng để đảm bảo tính toàn vẹn và bảo mật. Nhóm chọn giải pháp mã hóa hai chiều sử dụng phương pháp EncryptByPassPhrase vì khả năng bảo mật cao và dễ dàng phục hồi dữ liệu khi cần thiết.

Khi áp dụng EncryptByPassPhrase, dữ liệu sẽ được mã hóa trước khi lưu trữ và có thể dễ dàng giải mã khi cần truy xuất. Cơ chế này sử dụng mật khẩu để tạo khóa mã hóa, giúp bảo vệ dữ liệu nhạy cảm khỏi bị truy cập trái phép. Điều đặc biệt của phương pháp này là khả năng giải mã khi người dùng có quyền hợp lệ, giúp duy trì tính bảo mật mà vẫn linh hoạt trong việc truy xuất thông tin.

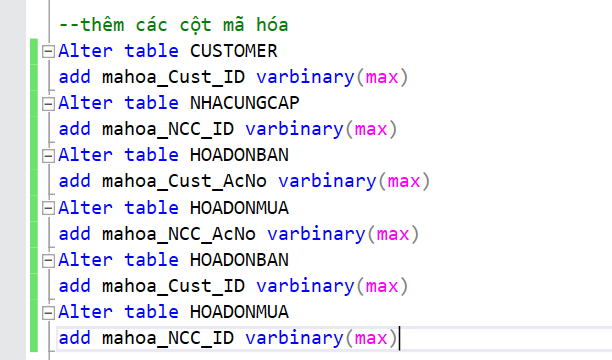
Phương pháp mã hóa hai chiều giúp tăng cường bảo mật dữ liệu khi truyền tải, đặc biệt là trong các hệ thống yêu cầu bảo mật cao. Dữ liệu sẽ được mã hóa khi lưu trữ trong cơ sở dữ liệu và chỉ có thể được giải mã khi sử dụng mật khẩu đúng, đảm bảo rằng thông tin không bị rò rỉ ngay cả khi cơ sở dữ liệu bị xâm nhập.

Với đặc điểm này, nhóm đã lựa chọn EncryptByPassPhrase để bảo vệ các dữ liệu quan trọng trong cơ sở dữ liệu của hệ thống, mang lại sự an toàn và bảo mật thông tin cho người dùng

Nhóm lựa chọn mã hóa cột *Cust\_ID* trong bảng CUSTOMER, *NCC\_ID* trong bảng NHACUNGCAP, *Cust\_AcNo* trong bảng HOADONBAN, *NCC\_AcNo* trong bảng HOADONMUA vì những thông tin này có tính chất nhạy cảm và quan trọng

Dưới đây là các bước thực hiện:

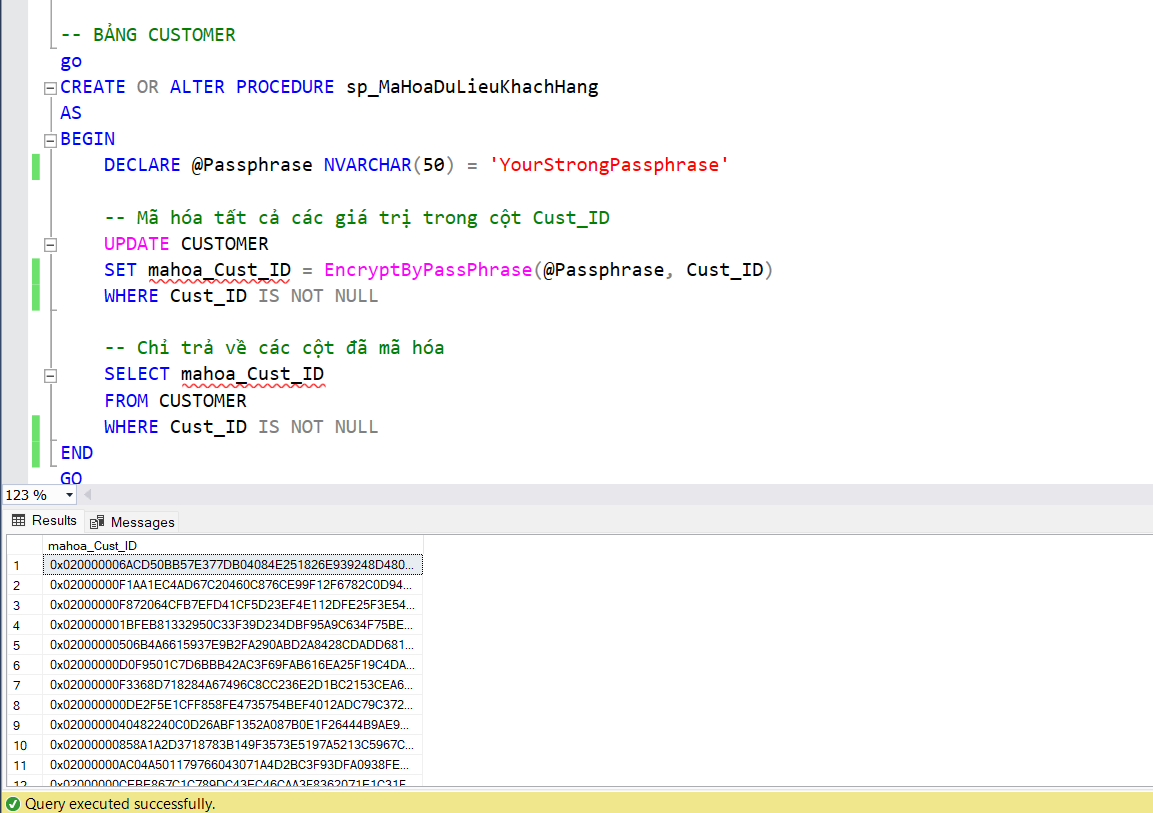
Bước 1: Thêm các cột mã hóa vào các bảng



**Hình 15.** Demo Mã hóa dữ liệu

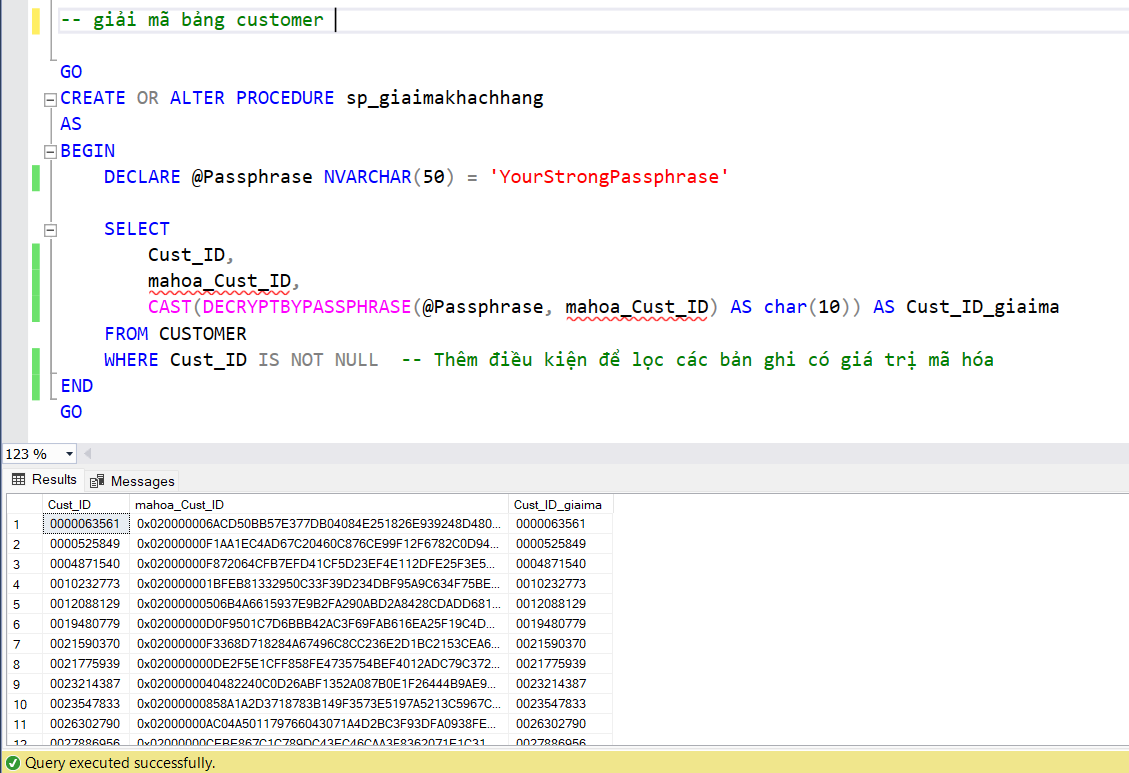
Bước 2: Thực hiện mã hóa và giải mã

* + 1. Mã hóa và giải mã Cust\_ID ở bảng Customer:
       - Mã hóa



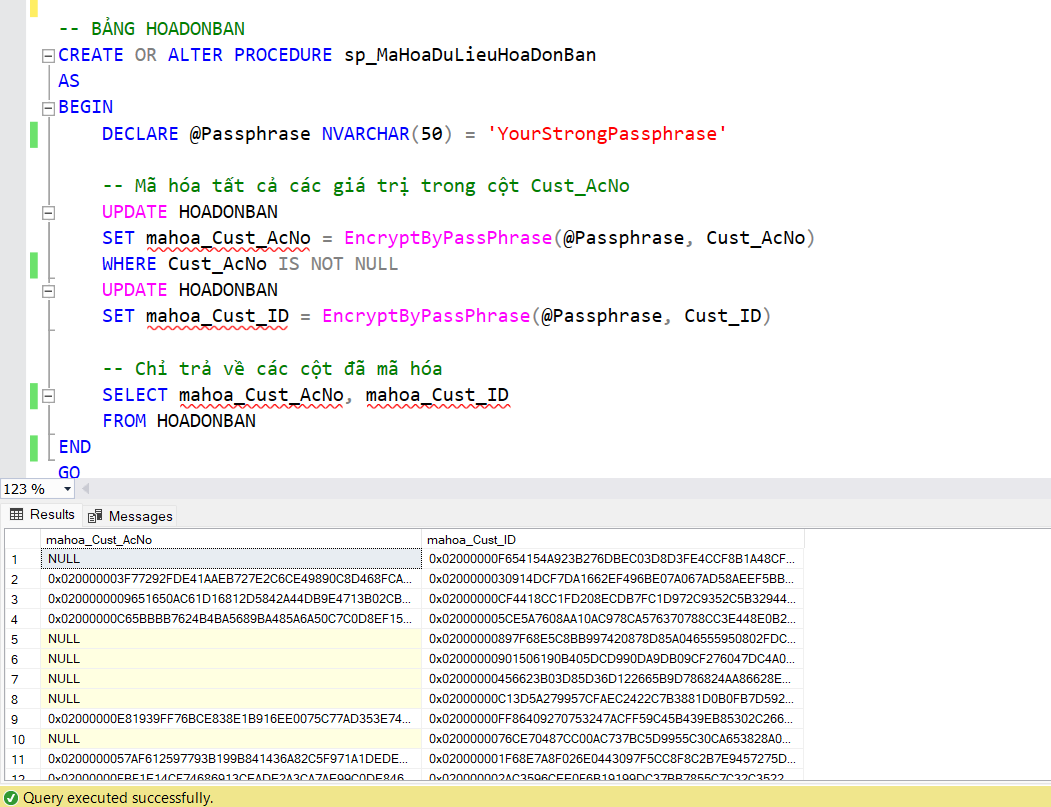
**Hình 16.** Demo Mã hóa dữ liệu bảng CUSTOMER

* + - * Giải mã



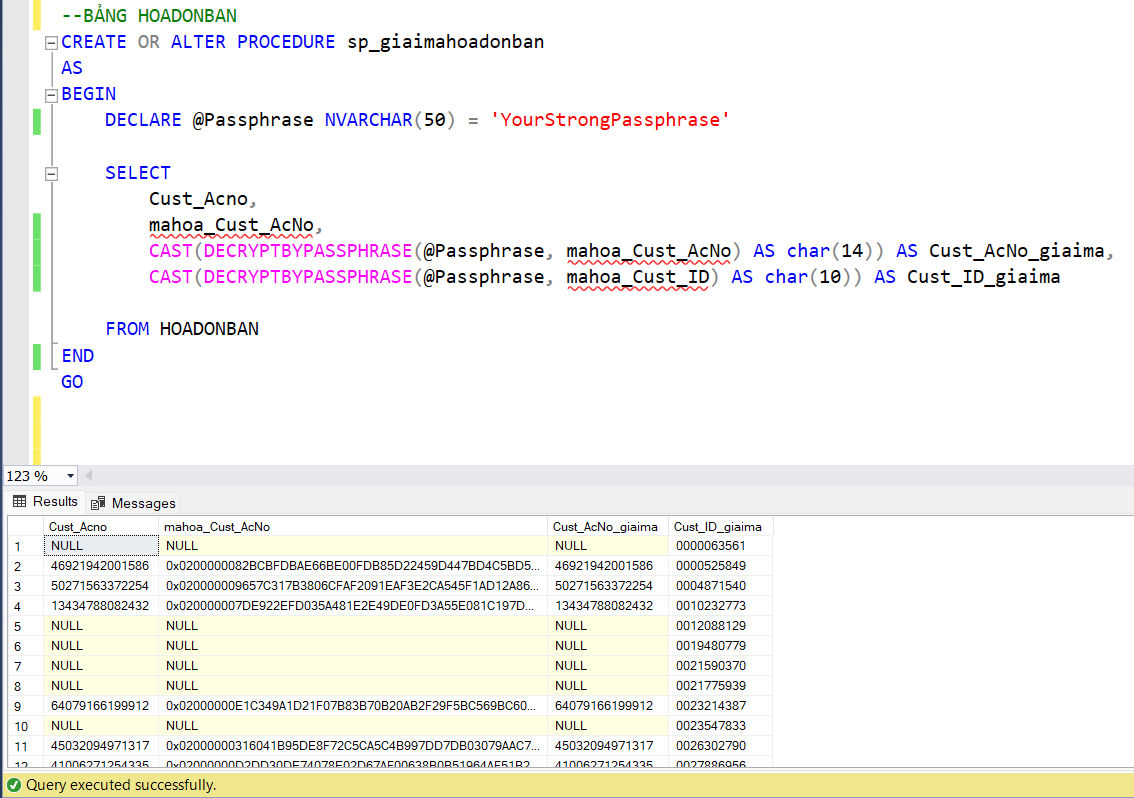
**Hình 17.** Demo giải mã dữ liệu bảng CUSTOMER

* + 1. Mã hóa và giải mã Cust\_ID, Cust\_Acno ở bảng HOADONBAN:
       - Mã hóa



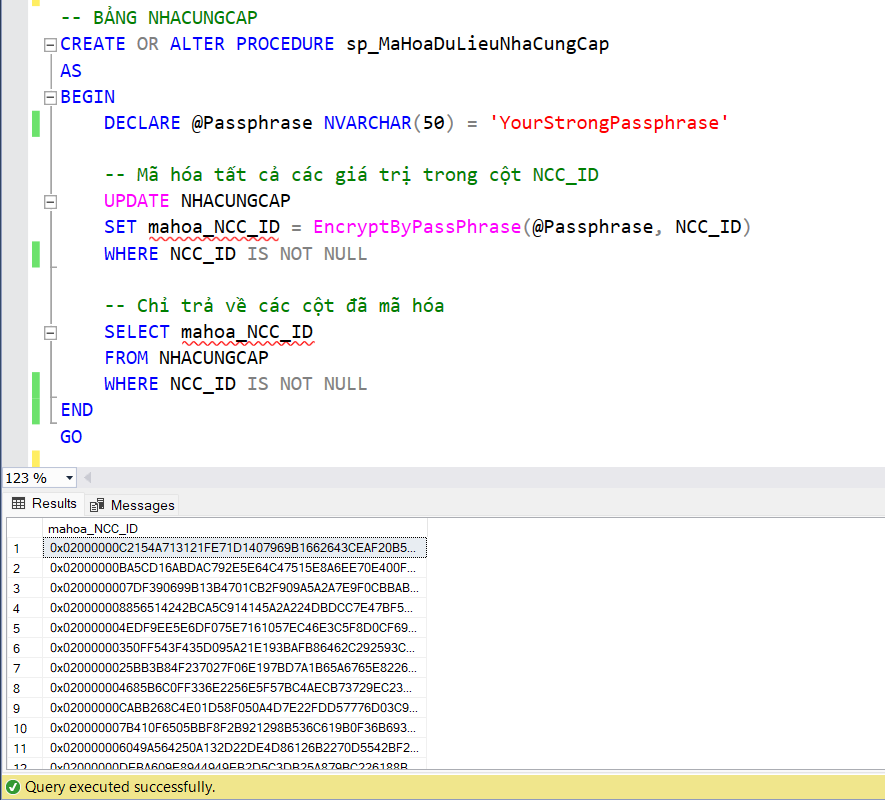
**Hình 18.** Demo Mã hóa dữ liệu bảng HOADONBAN

* + - * Giải mã



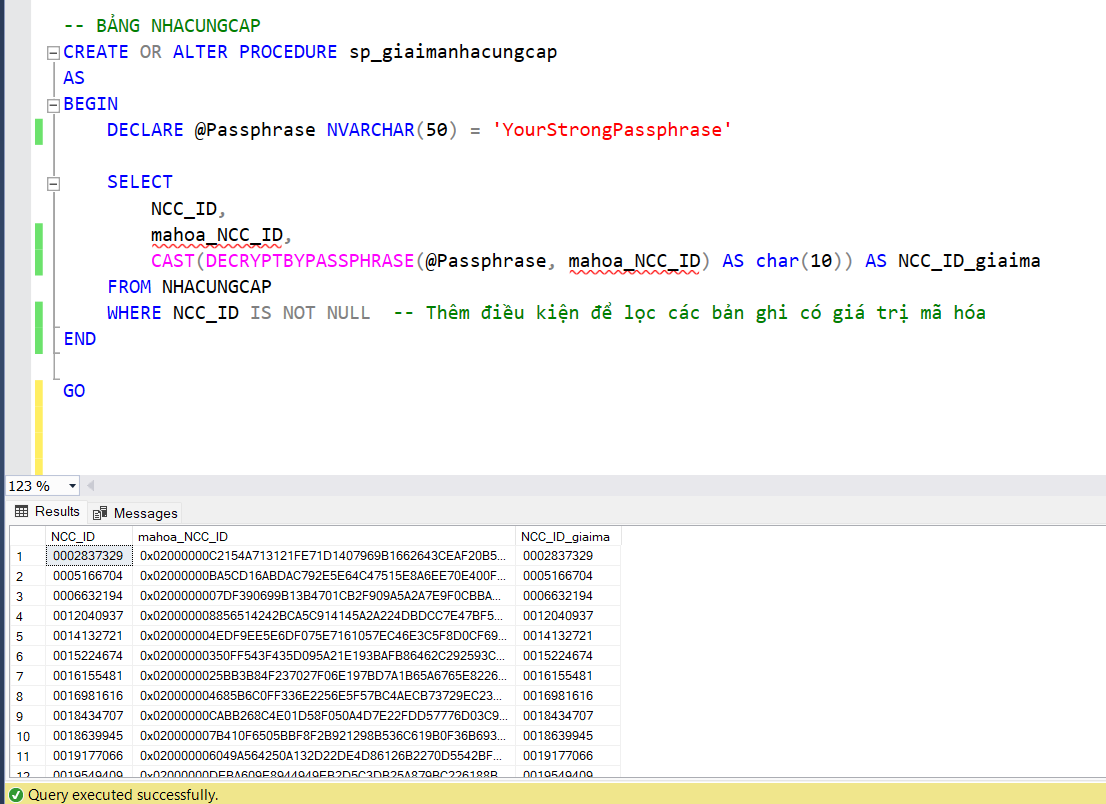
**Hình 19.** Demo giải mã dữ liệu bảng HOADONBAN

* + 1. Mã hóa và giải mã NCC\_ID ở bảng NHACUNGCAP:
       - Mã hóa



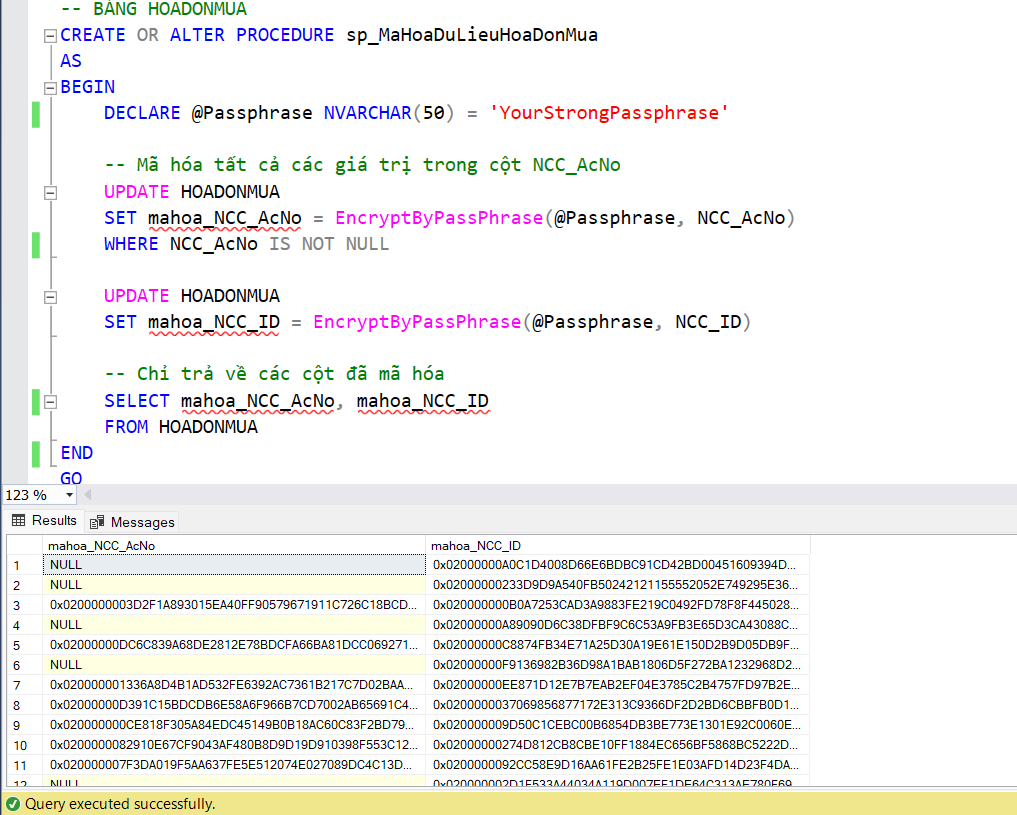
**Hình 20.** Demo Mã hóa dữ liệu bảng NHACUNGCAP

* + - * Giải mã



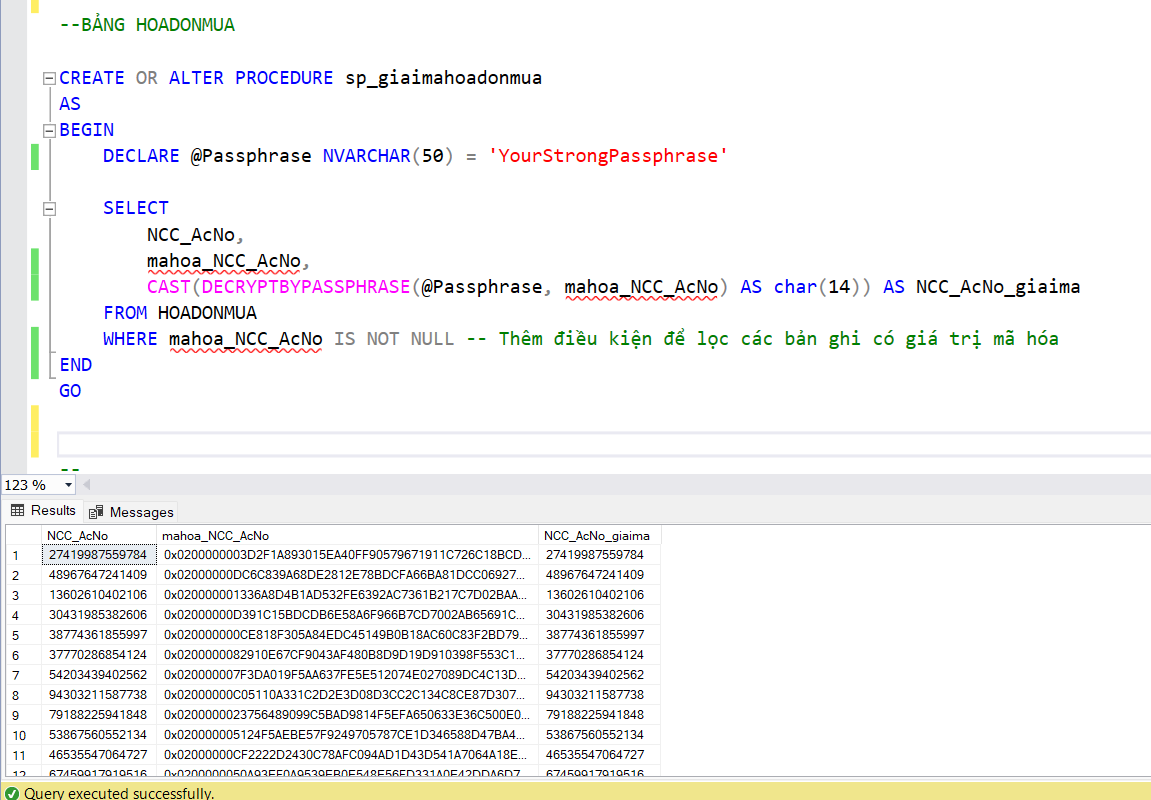
**Hình 21.** Demo giải mã dữ liệu bảng NHACUNGCAP

* + 1. Mã hóa và giải mã NCC\_ID, NCC\_AcNo ở bảng HOADONMUA:
       - Mã hóa



**Hình 22.** Demo Mã hóa dữ liệu bảng HOADONMUA

* + - * Giải mã



**Hình 23.** Demo giải mã dữ liệu bảng HOADONMUA