

**TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT**  
**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO GIỮA KỲ**  
**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**MÃ HỌC PHẦN: 242MI6601**

**THIẾT KẾ VÀ TRIỂN KHAI HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**  
**CHO DOANH NGHIỆP LOGISTIC**

**Giảng viên hướng dẫn: TS Nguyễn Thôn Dã**

**Danh sách thành viên nhóm:**

1. K224060785, Trần Trung Hiếu
2. K224060792, Nguyễn Thành Lợi
3. K214060421, Phạm Tấn Tùng
4. K214061736, Nguyễn Duy Duẩn
5. K224060788, Ngô Văn Kiên

Thành phố Hồ Chí Minh, 2025

## Bảng tự đánh giá thành viên nhóm

---

<i>TT</i>	MSSV	Họ và tên	Điểm tự đánh giá (thang điểm 10)
1	K224060785	Trần Trung Hiếu	10/10
2	K224060792	Nguyễn Thành Lợi	10/10
3	K214060421	Phạm Tuấn Tùng	10/10
4	K214061736	Nguyễn Duy Duẩn	10/10
5	K224060788	Ngô Văn Kiên	10/10

# Lời cảm ơn của nhóm

---

Trước hết, chúng em xin cảm ơn Trường Đại học Kinh tế - Luật - Đại học Quốc gia Thành phố Hồ Chí Minh và Khoa Hệ thống thông tin đã đưa môn Hệ Quản Trị Cơ Sở Dữ Liệu vào chương trình giảng dạy.

Đặc biệt, để hoàn thành báo cáo này, chúng em xin bày tỏ lòng biết ơn sâu sắc tới TS. Nguyễn Thôn Dã – giảng viên hiện đang giảng dạy môn Hệ Quản Trị Cơ Sở Dữ Liệu cho lớp chúng em. Những bài giảng cung cấp những kiến thức bổ ích, thú vị cùng với sự tâm huyết của các thầy trong quá trình giảng dạy là nguồn cảm hứng to lớn giúp chúng em hoàn thành báo cáo này.

Kiến thức là vô hạn nhưng kiến thức của bản thân mỗi người luôn có những hạn chế nhất định. Trong quá trình thực hiện báo cáo chúng em sẽ khó tránh khỏi những sai sót. Nhóm chúng em rất mong nhận được sự góp ý cô để nhóm có thể hoàn thiện hơn nữa. Điều này có ý nghĩa rất lớn đối với nhóm của chúng em.

Chúc thầy thành công hơn nữa trong sự nghiệp giảng dạy.

Chúng em chân thành cảm ơn !

## Lời cam kết

---

Chúng em cam đoan kết quả nghiên cứu này là của riêng chúng em, chúng em khẳng định không sao chép kết quả nghiên cứu của những cá nhân hoặc nhóm nghiên cứu nào khác.

Ho Chi Minh City, tháng 3 năm 2025

Tập thể thành viên nhóm 9

# MỤC LỤC

<b>Bảng tự đánh giá thành viên nhóm.....</b>	<b>0</b>
<b>Lời cảm ơn của nhóm.....</b>	<b>1</b>
<b>Lời cam kết.....</b>	<b>2</b>
1.1. Mô tả:.....	1
1.2. Điểm nổi bật :.....	2
2.1. Thiết kế dtb ( Mô tả bảng, ERD):.....	2
2.2. Sơ đồ ERD.....	11
2.3. Thực hiện.....	11
+ Indexes and Statistics.....	17
+ SQL Server Security:.....	20
+ Phân quyền dựa trên vai trò (Role-Based Security).....	21
+ Theo dõi truy cập (Audit).....	37
+ Encryption.....	38
+ Backups and Restores.....	46

## DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ ERD.....	9
Hình 2: Tạo database Giữa kỳ.....	11
Hình 3: Thao tác mở File SQL.....	12
Hình 4: Giao diện thực hiện.....	12
Hình 6: Khởi chạy database.....	13
Hình 7: Tạo Partition cho bảng Payment.....	13
Hình 8: Giao diện thực hiện tạo Partition.....	14
Hình 9: Sử dụng cột Payment_Date là Partitioning Column.....	14
Hình 10: Nén bảng Payment, sử dụng Row Compression cho tất cả các partition.....	15
Hình11: Tạo role CustomerServiceRole.....	20
Hình 12: Gán quyền SELECT, INSERT, UPDATE cho role CustomerServiceRole ở bảng CONSULTATION.....	21
Hình 13: Gán quyền SELECT cho role CustomerServiceRole ở bảng CONTRACT.....	22
Hình 14: Gán quyền SELECT, INSERT, UPDATE cho role CustomerServiceRole ở bảng CUSTOMER.....	23
Hình 15: Gán quyền SELECT, INSERT, UPDATE cho role CustomerServiceRole ở bảng SERVICE_REQUEST.....	24
Hình 16: Khởi tạo các role còn lại bằng TSQL.....	25
Hình 17: CustomerServiceRole bị giới hạn một số cột ở bảng CUSTOMER.....	26
Hình 18: Khởi tạo view về thông tin khách hàng.....	27
Hình 19: Kết quả sau khi tạo View vw_CustomerPublicInfo.....	27
Hình 20: Lựa chọn view cần gán quyền.....	28
Hình 21: Gán quyền SELECT cho CustomerServiceRole ở vw_CustomerPublicInfo.....	29
Hình 22: Tạo Login CS_NguyenVan.....	30
Hình 23: Cấu hình User Mapping CS_NguyenVan.....	31
Hình 24: Gán quyền cho Login.....	32
Hình 25: Tạo User CS_NguyenVan.....	33

Hình 26: Chọn Role Membership cho User CS_NguyenVan.....	34
Hình 27: Tạo các Login và User khác còn thiếu.....	35
Hình 28: Tạo bảng log và gán quyền truy cập cho ManagementRole.....	36
Hình 29: SQL Server Audit ghi nhận các hoạt động truy cập dữ liệu.....	36
Hình 30: Thao tác back up.....	38
Hình 31: Thực hiện 2 lần backup “Full” và “Transaction Log” rồi nhấn Ok.....	39
Hình 32: chọn Encrypt Column.....	40
Hình 33: Chọn Encryption type và Encryption Key.....	41
Hình 34: Truy vấn vào dữ liệu.....	42
Hình 35: Thao tác mã hóa DB.....	42
Hình 36: Chọn Options, set True cho Encryption Enabled.....	43
Hình 37: Kiểm tra xem DB đã được mã hóa chưa.....	44
Hình 38: Tạo job backup weekly cho account manager.....	45
Hình 39: Tạo step backup weekly.....	46
Hình 40: Tạo schedule backup weekly.....	47
Hình 41: Kết quả sau khi bắt đầu khởi chạy job.....	48

## **DANH MỤC BẢNG BIỂU**

Bảng phân quyền cơ sở dữ liệu.....	19
Bảng Thông tin User.....	29
Bảng giải pháp mã hóa.....	36



## 1. Mô tả tình huống thực hành

### 1.1. Mô tả:

Nghiên cứu điển hình này tập trung vào việc xây dựng một hệ thống cơ sở dữ liệu hiện đại cho một công ty chuyên xử lý thủ tục hải quan, theo mô hình kinh doanh B2B. Công ty cung cấp các dịch vụ chuyên sâu liên quan đến quản lý hồ sơ, khai báo và kiểm tra hàng hóa xuất nhập khẩu, từ đó đảm bảo quy trình thông quan được thực hiện nhanh chóng, chính xác và hiệu quả. Mục tiêu của dự án là phát triển một hệ thống thông minh, có khả năng quản lý và xử lý khối lượng dữ liệu lớn, góp phần nâng cao hiệu quả vận hành và tạo ra lợi thế cạnh tranh trên thị trường.

Hệ thống cơ sở dữ liệu được thiết kế để quản lý một loạt thông tin quan trọng, bao gồm dữ liệu về khách hàng, nhân viên, hàng hóa, hợp đồng, tờ khai hải quan và các quy trình liên quan. Từ giai đoạn tiếp nhận tư vấn khách hàng ban đầu cho đến quá trình thanh toán và dịch vụ hậu thông quan, toàn bộ các nghiệp vụ được tích hợp một cách đồng bộ và tối ưu, giúp giảm thiểu sai sót và tăng cường hiệu quả vận hành. Qua đó, công ty không chỉ đáp ứng được nhu cầu ngày càng cao về quản lý dữ liệu mà còn tạo dựng được hệ thống thông tin vững chắc phục vụ cho chiến lược kinh doanh dài hạn.

*Lợi ích đem lại:*

#### *a. Đối với doanh nghiệp:*

Đối với doanh nghiệp, dự án mang lại nhiều lợi ích thiết thực. Hệ thống giúp duy trì vị thế cạnh tranh, quản lý hiệu quả khối lượng dữ liệu khổng lồ liên quan đến thủ tục hải quan, từ đó tối ưu hóa chi phí đầu tư cho phần cứng và công nghệ. Ngoài ra, khả năng phối hợp nhịp nhàng giữa các bộ phận được cải thiện rõ rệt, góp phần duy trì chất lượng dịch vụ và thúc đẩy phát triển chiến lược kinh doanh hiệu quả. Việc xử lý thủ tục hải quan trở nên nhanh chóng, chính xác và tiết kiệm chi phí, đồng thời hỗ trợ khách hàng tối ưu hóa thời gian và nguồn lực trong quy trình xuất nhập khẩu.

#### *b. Đối với khách hàng:*

Đối với khách hàng, hệ thống đảm bảo an toàn thông tin với mức độ bảo mật cao, giảm thiểu các rắc rối phát sinh từ dữ liệu không chính xác. Họ được trải nghiệm dịch vụ nhanh chóng, hiệu quả và tiết kiệm về thời gian cũng như chi phí, qua đó gia tăng niềm tin và sự hài lòng. Một điểm nổi bật của nghiên cứu là việc xác định rõ ràng mô hình kinh doanh B2B, giúp định hướng chính xác các nghiệp vụ và hiểu rõ nhu cầu của đối tượng khách hàng, từ đó xây dựng một hệ thống cơ sở dữ liệu toàn diện, linh hoạt và phù hợp với xu hướng phát triển của ngành.

## **1.2. Điểm nổi bật :**

Một điểm mạnh nổi bật của dự án là việc xác định một cách rõ ràng mô hình kinh doanh B2B, từ đó giúp doanh nghiệp tập trung vào việc phát triển và hoàn thiện các nghiệp vụ thiết yếu phù hợp với nhu cầu đặc thù của khách hàng doanh nghiệp. Cách định hướng này không chỉ giúp hiểu rõ đặc điểm, kỳ vọng và thách thức của từng nhóm khách hàng mục tiêu mà còn tạo điều kiện thuận lợi cho việc thiết kế một hệ thống quản lý dữ liệu thông minh và hiệu quả. Hệ thống này được xây dựng để xử lý và quản lý toàn diện thông tin liên quan đến hồ sơ, tờ khai hải quan, hợp đồng, hàng hóa và các quy trình hậu cần khác, đảm bảo quá trình làm việc trở nên mạch lạc, chính xác và tối ưu về chi phí. Qua đó, doanh nghiệp không chỉ cải thiện khả năng phối hợp nội bộ, nâng cao chất lượng dịch vụ mà còn khẳng định vị thế cạnh tranh trên thị trường, đồng thời phát triển các chiến lược kinh doanh bền vững trong bối cảnh cạnh tranh ngày càng khốc liệt.

## **2. Trình bày chi tiết nội dung thực hiện**

- Các bước thực hiện, chụp hình giao diện thao tác, source code hỗ trợ

### **2.1. Thiết kế đtb ( Mô tả bảng, ERD):**

#### **2.1.1. CUSTOMER : Khách hàng**

Là một cá nhân có nhu cầu xuất nhập khẩu hàng hóa, được đại diện bằng tài khoản với các thông tin đăng ký. Thực thể CUSTOMER bao gồm các thuộc tính sau:

- Customer\_ID (PK): Mã số khách hàng
- Customer Name: Tên khách hàng
- Customer Email: Địa chỉ email của khách hàng
- Customer Phone: Số điện thoại của khách hàng
- Bank Account Number: Số tài khoản ngân hàng khách hàng sử dụng để thanh toán
- Tax Code: Mã số thuế
- Company Name: Tên công ty (nếu là doanh nghiệp)
- Address: Địa chỉ khách hàng

#### 2.1.2. EMPLOYEE : Nhân viên

Là các nhân viên của Forwarder, gồm nhiều loại nhân viên công việc và nhiệm vụ khác nhau. Thực thể EMPLOYEE bao gồm các thuộc tính sau:

- Employee\_ID (PK): Mã số nhân viên
- Department\_ID (FK): Mã số phòng ban
- Employee Name: Tên nhân viên
- Employee Phone: số điện thoại nhân viên
- Employee Email: Email nhân viên
- Hired Date: Ngày vào làm của nhân viên
- Salary: Tiền lương

- Role: chức vị của employee (manager,staff)

### 2.1.3. DEPARTMENT : Phòng ban

Là các phòng ban được phân theo chức năng của Forwarder . Thực thể

DEPARTMENT bao gồm các thuộc tính sau:

- Department\_ID (PK): Mã số phòng ban
- Department\_Name: Tên phòng ban
- Manager\_ID: Mã người quản lý

### 2.1.4. GOODS DETAIL : Danh sách hàng hóa

Là thông tin về hàng hóa của khách hàng cần được thông quan để vận chuyển đi.

Thực thể GOODS DETAIL bao gồm các thuộc tính sau:

- Detail\_ID (PK): Mã đơn
- Declaration\_ID (FK): Mã tờ khai báo hải quan
- Goods \_ID (FK): Mã hàng hóa (xác định loại hàng hóa)
- Quantity : Số lượng
- Net-Weight : Trọng lượng hàng hoá
- Gross-Weight: Trọng lượng hàng hoá khi đã đóng gói
- Total value : Giá trị tổng

### 2.1.5. GOODS: Hàng hóa

- Goods \_ID (PK): Mã hàng hóa (xác định loại hàng hóa)
- Goods Name :Tên hàng hóa
- HS code : Mã HS code

- Unit price: Đơn giá
- Goods\_Type (Perishable\_Goods, Hazardous\_Goods, General\_Goods)

Trong mô hình dữ liệu thông tin hàng hoá được đề cập ở đây bao gồm 3 loại sau, chỉ được chọn 1 trong 3 loại:

- PERISHABLE GOODS: Loại hàng hóa này bao gồm các sản phẩm có thời hạn sử dụng ngắn và yêu cầu bảo quản đặc biệt để tránh hư hỏng.

- Exp Date: Hạn sử dụng của hàng hoá

- Temperature: Nhiệt độ bảo quản cần thiết (nếu có)

- Humidity: Độ ẩm bảo quản cần thiết (nếu có)

- HAZARDOUS GOODS: Đây là loại hàng hóa có tính chất nguy hiểm, có thể gây hại đến sức khỏe con người hoặc môi trường nếu không được xử lý và vận chuyển đúng cách.

- Hazard Level: Mức độ nguy hiểm (ví dụ: Low, Medium, High)

- Safety Instructions: Hướng dẫn an toàn

- GENERAL GOODS: Đây là loại hàng hóa không dễ hỏng và cũng không có tính chất nguy hiểm được gọi là hàng hoá thông thường

#### 2.1.6. CONSULTATION : Thông tin về tư vấn khách hàng

Là thông tin về việc tiếp nhận và tư vấn khách gọi tới. Thực thể

CONSULTATION bao gồm các thuộc tính sau:

- Consultant\_ID (PK): Mã cuộc tư vấn

- Customer\_ID (FK): Mã khách hàng
- Employee\_ID (FK): mã nhân viên tư vấn
- Consultation Date: Thời gian tư vấn
- Goods Type : Loại hàng hóa
- Estimate HS Code: Ước tính phí mã HS
- Estimate tax rate: Ước tính thuế
- Estimate cost: Ước tính chi phí tổng
- Required permit: Yêu cầu về giấy phép
- Status: Tình trạng cuộc tư vấn
- Notes: Ghi chú

#### 2.1.7. SERVICE REQUEST : Yêu cầu dịch vụ

Là thông tin về việc khách hàng đồng ý sử dụng dịch vụ thì sẽ khởi tạo một yêu cầu dịch vụ . Thực thể SERVICE REQUEST bao gồm các thuộc tính sau:

- Request\_ID (PK): Mã yêu cầu, khóa chính
- Customer\_ID (FK): Mã khách hàng, khóa ngoại tham chiếu đến Customer
- Employee\_ID (FK): Mã số nhân viên
- Service\_Type : Loại dịch vụ yêu cầu
- Service\_Charges: Phí dịch vụ
- Request\_Date: Ngày yêu cầu
- Status: Trạng thái yêu cầu

#### 2.1.8. GOODS INSPECTION : Kiểm tra hàng hóa

Là thông tin về việc kiểm tra lô hàng hóa. Thực thể GOODS INSPECTION bao gồm các thuộc tính sau:

- Inspection\_ID (PK): Mã kiểm tra, khóa chính
- Declaration\_ID (FK): Mã tờ khai báo hải quan, khóa ngoại tham chiếu đến Customs Declaration
- Employees\_ID (FK): Mã nhân viên chịu trách nhiệm kiểm tra
- Inspection Type : Loại kiểm tra
- Schedule Date: Ngày kiểm tra
- Inspection Location: Địa điểm kiểm tra
- Status: Tình Trạng
- Inspector Name: Tên nhân viên kiểm tra
- Results: Kết quả kiểm tra
- Notes: Ghi chú

#### 2.1.9. POST CLEARANCE SERVICE : Dịch vụ sau giao dịch

Là thông tin về chăm sóc khách hàng sau khi hoàn thành hợp đồng. Thực thể POST CLEARANCE SERVICE bao gồm các thuộc tính sau:

- Post Clearance Service ID (PK): Mã yêu cầu
- Declaration\_ID (FK): Mã tờ khai báo hải quan
- Document Storage Ref : Tham chiếu lưu trữ tài liệu
- Consultant Notes : Ghi chú tư vấn

- Review Date : Ngày đánh giá
- Report: Báo cáo tổng
- Improvement Suggestion : Góp ý cải thiện
- Status: Tình trạng

#### 2.1.10. CUSTOMER DECLARATION : Khai báo hải quan

Là thông tin về việc khai báo hải quan của mỗi hợp đồng. Thực thể CUSTOMER DECLARATION bao gồm các thuộc tính sau:

- Declaration\_ID (PK): Mã tờ khai báo hải quan
- Contract\_ID (FK): Mã hợp đồng, tham chiếu đến CONTRACT
- Employee\_ID (FK): Mã nhân viên chịu trách nhiệm
- Declaration Type: Loại khai báo
- VNACCS\_ID: Mã VNACCS
- Inspection Channel (Green, Yellow, Red) :Kênh kiểm tra
- Status: Tình trạng
- Submission Date: Ngày nộp

#### 2.1.11. CONTRACT : Hợp đồng

Là thông tin về hợp đồng được ký kết giữa Forwarder và khách hàng. Thực thể CONTRACT bao gồm các thuộc tính sau:

- Contract\_ID (PK): Mã hợp đồng
- Request\_ID (FK): Mã yêu cầu



- Contract Date: Hạn hợp đồng
- Total Value: Tổng giá trị hợp đồng
- Payment Term: Phương thức thanh toán
- Status: Tình trạng

#### 2.1.12. DOCUMENT SET : Bộ hồ sơ

Là thông tin về hồ sơ cần để thực hiện khai báo hải quan . Thực thể DOCUMENT SET bao gồm các thuộc tính sau:

- Document\_Set\_ID (PK): Mã bộ hồ sơ
- Declaration\_ID (FK): Mã tờ khai báo hải quan
- Contract\_ID (FK): Mã hợp đồng
- Verification Status: Trạng thái xác minh
- Notes: Ghi chú
- Status: Tình trạng

#### 2.1.13. DOCUMENT : Bộ chứng từ

Là thông tin về các giấy tờ cụ thể trong DOCUMENT SET. Thực thể DOCUMENT bao gồm các thuộc tính sau:

- Document\_ID (PK): Mã bộ chứng từ
- Document\_Set\_ID (FK): Mã bộ hồ sơ
- Commercial Invoice: Hóa đơn thương mại
- Packing List : Phiếu đóng gói

- Bill\_Of\_Lading: Chứng từ vận chuyển
- C/O: Chứng từ xuất xứ
- Import Permit: Giấy phép nhập khẩu
- Other Certification: Các chứng từ khác

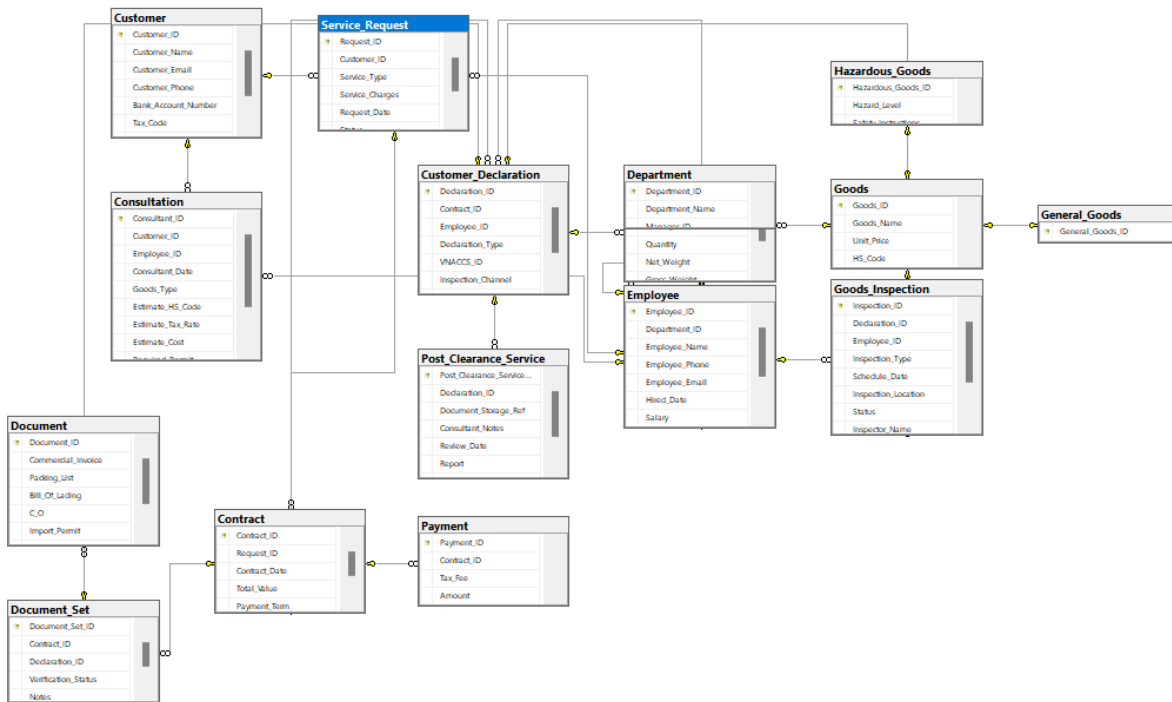
#### 2.1.14. PAYMENT : Thanh toán

Là thông tin về việc thanh toán các khoản thuế-phí sau khi hoàn thành hợp đồng.

Thực thể PAYMENT bao gồm các thuộc tính sau:

- Payment\_ID (PK): Mã thanh toán, khóa chính
- Contract\_ID (FK): Mã hợp đồng, khóa ngoại tham chiếu đến Contract
- Tax\_Fee : Thuế/phí
- Amount: Số tiền thanh toán
- Payment Date: Ngày thanh toán

## 2.2. Sơ đồ ERD



Hình 1: Sơ đồ ERD

## 2.3. Thực hiện

-- Bước 1: Tạo cơ sở dữ liệu GiuaKy với các filegroup và file log

CREATE DATABASE GiuaKy

ON PRIMARY

-- Bước 1.1: Tạo file chính trong Primary Filegroup trên ổ C

(NAME = N'GiuaKy\_Primary',

FILENAME = N'C:\SQLData\GiuaKy\_Primary.mdf',

SIZE = 10240KB, -- Dung lượng ban đầu 10MB

FILEGROWTH = 1024KB), -- Tăng trưởng 1MB mỗi lần

FILEGROUP CustomerDataFG

-- Bước 1.2: Tạo file trong CustomerDataFG trên ổ D cho dữ liệu khách hàng

(NAME = N'GiuaKy\_CustomerData',

FILENAME = N'D:\SQLData\GiuaKy\_CustomerData.ndf',

SIZE = 10240KB,

FILEGROWTH = 1024KB),

FILEGROUP GoodsDataFG

*-- Bước 1.3: Tạo file thứ nhất trong GoodsDataFG trên ổ C cho dữ liệu hàng hóa*

(NAME = N'GiuaKy\_GoodsData',

FILENAME = N'C:\SQLData\GiuaKy\_GoodsData.ndf',

SIZE = 10240KB,

FILEGROWTH = 1024KB),

*-- Bước 1.4: Tạo file thứ hai trong GoodsDataFG trên ổ C cho dữ liệu hàng hóa*

(NAME = N'GiuaKy\_GoodsData2',

FILENAME = N'C:\SQLData\GiuaKy\_GoodsData2.ndf',

SIZE = 10240KB,

FILEGROWTH = 1024KB),

FILEGROUP DeclarationFG

*-- Bước 1.5: Tạo file trong DeclarationFG trên ổ D cho dữ liệu khai báo*

(NAME = N'GiuaKy\_Declaration',

FILENAME = N'D:\SQLData\GiuaKy\_Declaration.ndf',

SIZE = 10240KB,

FILEGROWTH = 1024KB),

FILEGROUP DocumentFG

*-- Bước 1.6: Tạo file trong DocumentFG trên ổ E cho dữ liệu tài liệu*

(NAME = N'GiuaKy\_Document',

FILENAME = N'E:\SQLData\GiuaKy\_Document.ndf',

SIZE = 10240KB,

FILEGROWTH = 1024KB),

FILEGROUP PaymentFG

*-- Bước 1.7: Tạo file trong PaymentFG trên ổ C cho dữ liệu thanh toán*

(NAME = N'GiuaKy\_Payment',

FILENAME = N'C:\SQLData\GiuaKy\_Payment.ndf',

SIZE = 10240KB,

FILEGROWTH = 1024KB)

## LOG ON

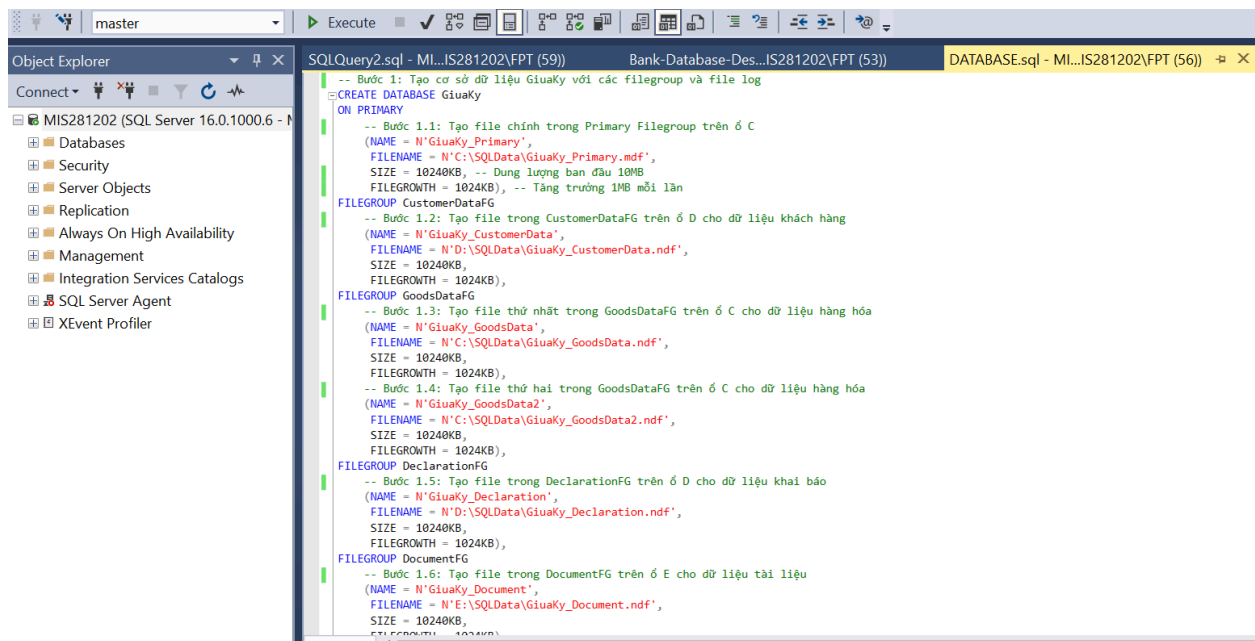
-- Bước 1.8: Tạo file log trên ổ E cho nhật ký giao dịch

(NAME = N'GiuaKy\_Log',

FILENAME = N'E:\SQLLog\GiuaKy\_Log.ldf',

SIZE = 5120KB, -- 5MB

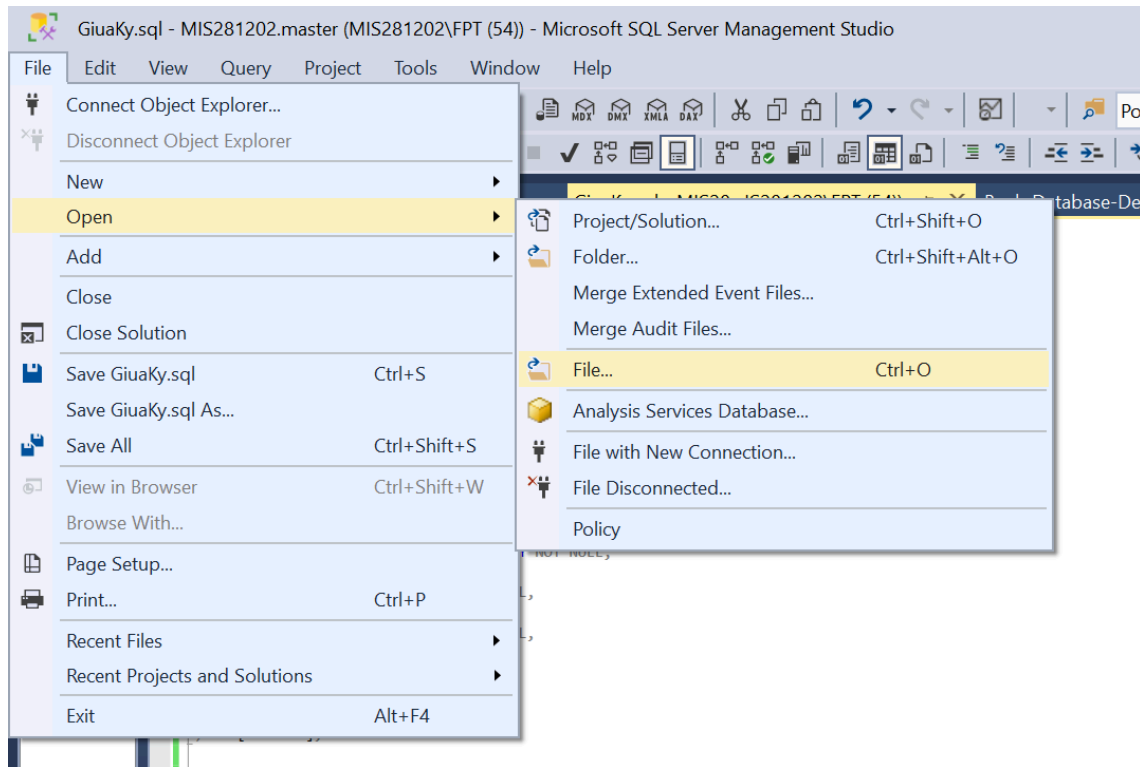
FILEGROWTH = 1024KB);



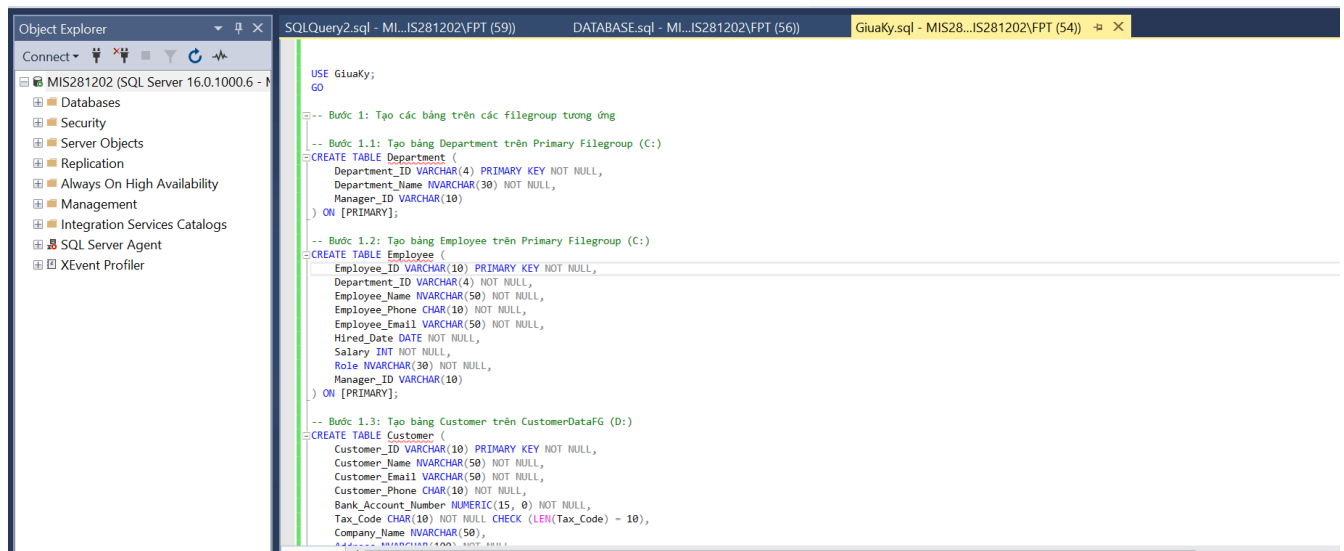
Hình 2: Tạo database Giữa kỳ

-- Bước 2: Nhập dữ liệu tạo bảng, thêm khóa ngoại và chèn dữ liệu vào bảng từ file script

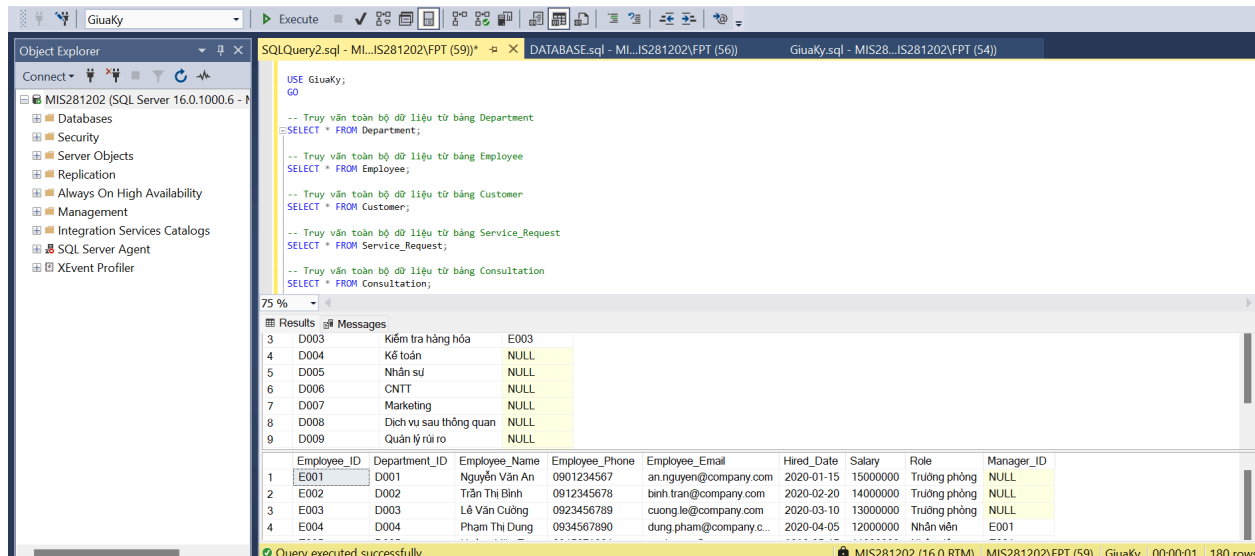
[GiuaKy.sql](#)



Hình 3: Thao tác mở File SQL



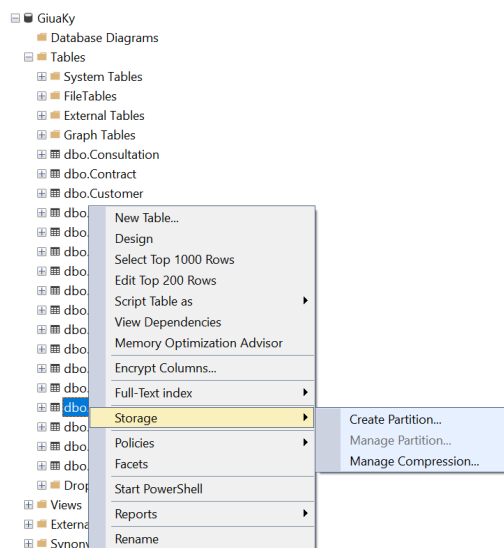
Hình 4: Giao diện thực hiện



Hình 6: Khởi chạy database

## + Table Optimizations:

Tiến hành tạo Partition cho bảng Payment:



Hình 7: Tạo Partition cho bảng Payment

Create Partition Wizard - Payment

### Select a Partitioning Column

Select the column on which you want to partition your table.

Available partitioning columns:

	Column name	Data type	Length	Precision	Scale
<input type="radio"/>	Amount	int	4	10	0
<input type="radio"/>	Contract_ID	varchar	10	0	0
<input checked="" type="radio"/>	Payment_Date	date	3	10	0
<input type="radio"/>	Payment_ID	varchar	15	0	0
<input type="radio"/>	Tax_Fee	numeric	9	10	0

☐ Collocate this table to the selected partitioned table:

☐ Storage-align all non-unique indexes and unique indexes with indexed partitioning column

The above grid contains the partitioning columns for the selected table. Select the column you want to use as the partitioning column in this table.

Hình 8: Giao diện thực hiện tạo Partition

Sử dụng cột Payment\_Date là Partitioning Column.

Create Partition Wizard - Payment

### Map Partitions

Map your partitions to filegroups and specify range values.

Range

☒ Left boundary

☐ Right boundary

Select filegroups and specify boundary values:

	Filegroup	<= Boundary	Rowcount	Required space	Available space
▶	PRIMARY	2/1/2024	3	0.000 MB	5.375 MB
	PRIMARY	3/1/2024	6	0.000 MB	5.375 MB
	PRIMARY		3	0.000 MB	5.375 MB
*					

Hình 9: Sử dụng cột Payment\_Date là Partitioning Column



Chia bảng Payment với 3 partition, cả 3 đều nằm trong filegroup, vì có sử dụng index.

Tiếp tục nén bảng Payment, sử dụng Row Compression cho tất cả các partition ( vì dữ liệu thuộc dạng Date)

Partition no.	Compression type	Boundary	Row count	Current space	Requested compressed space
1	Row	2/1/2024 12:...	3	0.047 MB	0.047 MB
2	Row	3/1/2024 12:...	6	0.016 MB	0.016 MB
3	Row		3	0.016 MB	0.016 MB

Hình 10: Nén bảng Payment, sử dụng Row Compression cho tất cả các partition

#### + Indexes and Statistics

---Tối ưu các truy vấn lọc theo khoảng thời gian thanh toán (*Payment\_Date*) và liên kết hợp đồng (*Contract\_ID*)

```
CREATE NONCLUSTERED INDEX IX_Payment_Date_Contract
ON Payment (Payment_Date, Contract_ID)
INCLUDE (Tax_Fee, Amount);
GO
```

-----Tối ưu các truy vấn phân tích hoặc lọc theo loại khai báo (*Declaration\_Type*) và trạng thái (*Status*) của khai báo hải quan

```
CREATE NONCLUSTERED INDEX IX_CustomerDeclaration_DeclType_Status
```

ON Customer\_Declaration (Declaration\_Type, Status)

INCLUDE (Submission\_Date, Contract\_ID);

GO

*-----Tối ưu các truy vấn liên quan đến khai báo hải quan (Declaration\_ID) và tính toán số lượng, giá trị hàng hóa*

CREATE NONCLUSTERED INDEX IX\_GoodsDetail\_Declaration\_Quantity

ON Goods\_Detail (Declaration\_ID, Quantity)

INCLUDE (Goods\_ID, Total\_Value);

GO

*-----Tối ưu truy vấn theo Goods\_ID*

CREATE NONCLUSTERED INDEX IX\_GoodsDetail\_GoodsID

ON Goods\_Detail (Goods\_ID)

INCLUDE (Quantity, Total\_Value);

GO

*-----hỗ trợ lựa chọn kế hoạch thực thi tối ưu khi truy vấn theo thuế*

CREATE STATISTICS ST\_Payment\_TaxFee

ON Payment (Tax\_Fee) WITH SAMPLE 100 PERCENT;

GO

*-----Cung cấp thông tin về phân bố ngày nộp khai báo hải quan*

CREATE STATISTICS ST\_CustomerDeclaration\_SubmissionDate

ON Customer\_Declaration (Submission\_Date) WITH FULLSCAN;

GO

*-----Giúp tối ưu các truy vấn tính tổng số lượng hàng hóa theo từng khai báo*

CREATE STATISTICS ST\_GoodsDetail\_Quantity

ON Goods\_Detail (Quantity) WITH SAMPLE 50 PERCENT;

GO

*-----lọc hoặc tính toán theo Amount*

CREATE STATISTICS ST\_Payment\_Amount

ON Payment (Amount) WITH FULLSCAN;

GO

*-----truy vấn lọc theo cột Status*

CREATE NONCLUSTERED INDEX IX\_ServiceRequest\_Status

ON Service\_Request (Status)

INCLUDE (Request\_Date);

GO

*----Tạo thống kê trên cột Request\_Date của bảng Service\_Request*

CREATE STATISTICS ST\_ServiceRequest\_RequestDate

ON Service\_Request (Request\_Date) WITH SAMPLE 100 PERCENT;

GO

*-----Truy vấn lấy số lần sử dụng của các chỉ mục trên bảng Payment từ hệ thống*

SELECT

    i.name,

    s.user\_seeks, s.user\_scans, s.user\_lookups, s.user\_updates

FROM sys.indexes i

INNER JOIN sys.dm\_db\_index\_usage\_stats s

    ON i.object\_id = s.object\_id AND i.index\_id = s.index\_id

WHERE i.object\_id = OBJECT\_ID('Payment');

*-----Phân tích hiệu suất truy vấn, đánh giá mức độ sử dụng IO, thời gian xử lý*

SET STATISTICS IO ON;

SET STATISTICS TIME ON;

SELECT Payment\_ID, Payment\_Date, Tax\_Fee, Amount

FROM Payment

WHERE Payment\_Date BETWEEN '2024-03-01' AND '2024-03-10';

```
SET STATISTICS IO OFF;
```

```
SET STATISTICS TIME OFF;
```

-----*Hiển thị thông tin về thống kê ST\_Payment\_TaxFee trên bảng Payment*

```
DBCC SHOW_STATISTICS ('Payment', 'ST_Payment_TaxFee');
```

----*Liệt kê danh sách tất cả các chỉ mục (INDEX) có trên bảng Payment*

```
SELECT name, index_id, type_desc
```

```
FROM sys.indexes
```

```
WHERE object_id = OBJECT_ID('Payment');
```

-----*Kiểm tra mức độ phân mảnh*

```
SELECT
```

```
    dbschemas.name AS SchemaName,
```

```
    dbtables.name AS TableName,
```

```
    dbindexes.name AS IndexName,
```

```
    indexstats.index_id,
```

```
    indexstats.avg_fragmentation_in_percent
```

```
FROM sys.dm_db_index_physical_stats (DB_ID(), NULL, NULL, NULL, 'DETAILED') AS  
indexstats
```

```
    INNER JOIN sys.tables dbtables on indexstats.object_id = dbtables.object_id
```

```
    INNER JOIN sys.schemas dbschemas on dbtables.schema_id = dbschemas.schema_id
```

```
    INNER JOIN sys.indexes dbindexes on dbindexes.object_id = indexstats.object_id
```

```
        AND dbindexes.index_id = indexstats.index_id
```

```
ORDER BY avg_fragmentation_in_percent DESC;
```

### **+ SQL Server Security:**

Đối với một hệ thống quản lý thủ tục hải quan, việc bảo mật dữ liệu là yếu tố then chốt để đảm bảo tính toàn vẹn và bảo mật thông tin. Sau đây là các giải pháp bảo mật được triển khai dựa trên công nghệ SQL Server. Vì vậy chúng tôi lựa chọn hình thức xây dựng hệ quản trị cơ sở dữ liệu đa lớp:

- Kiểm soát truy cập chặt chẽ: Mỗi nhân viên chỉ thấy đúng dữ liệu họ cần để thực hiện công việc, giảm thiểu rủi ro rò rỉ thông tin dựa vào tính phân quyền.

- Theo dõi hoạt động: Hệ thống audit giúp phát hiện sớm các hành vi đáng ngờ khi truy cập tờ khai hải quan, hỗ trợ việc điều tra khi có sự cố.

#### **+ Phân quyền dựa trên vai trò (Role-Based Security)**

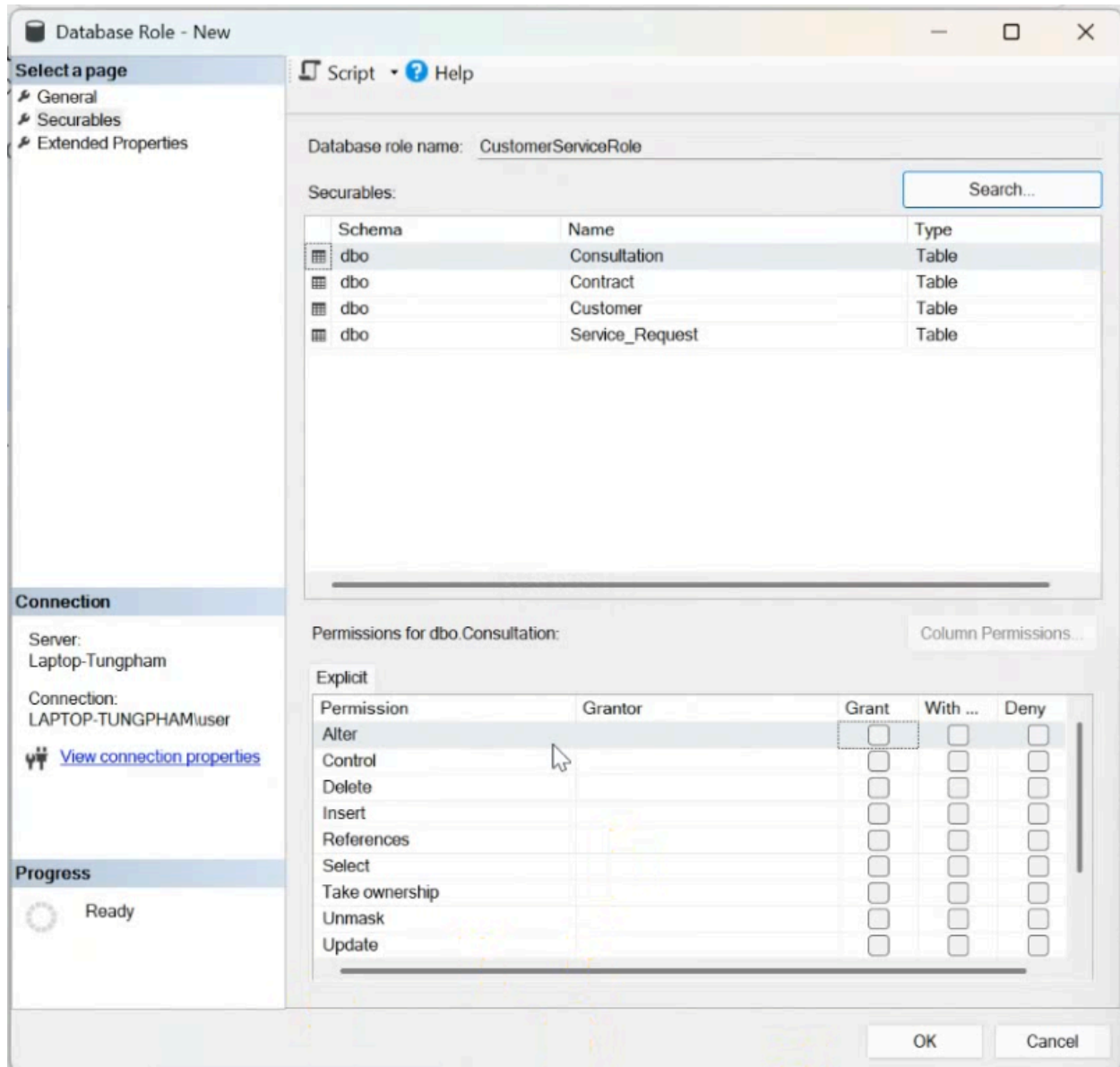
Hệ thống được thiết kế với cơ chế phân quyền chi tiết, phân chia người dùng theo các vai trò nghiệp vụ cụ thể:

Bảng phân quyền cơ sở dữ liệu

<b>Vai trò</b>	<b>Quyền truy cập</b>	<b>Hạn chế</b>
CustomerServiceRole	- Quản lý thông tin khách hàng - Quản lý yêu cầu dịch vụ	- Không được truy cập dữ liệu tài chính - Không được truy cập chi tiết tờ khai
DeclarationProcessorRole	- Quản lý tờ khai - Quản lý thông tin hàng hóa	- Không được truy cập thông tin thanh toán
InspectionRole	- Xem thông tin tờ khai - Xem thông tin hàng hóa - Cập nhật kết quả kiểm tra	- Không có quyền chỉnh sửa thông tin khách hàng hoặc hàng hóa - Không truy cập thanh toán
FinanceRole	- Quản lý thanh toán - Quản lý hợp đồng	- Không được thay đổi thông tin tờ khai
ManagementRole	- Có toàn quyền truy cập và quản lý mọi dữ liệu	- Không có hạn chế

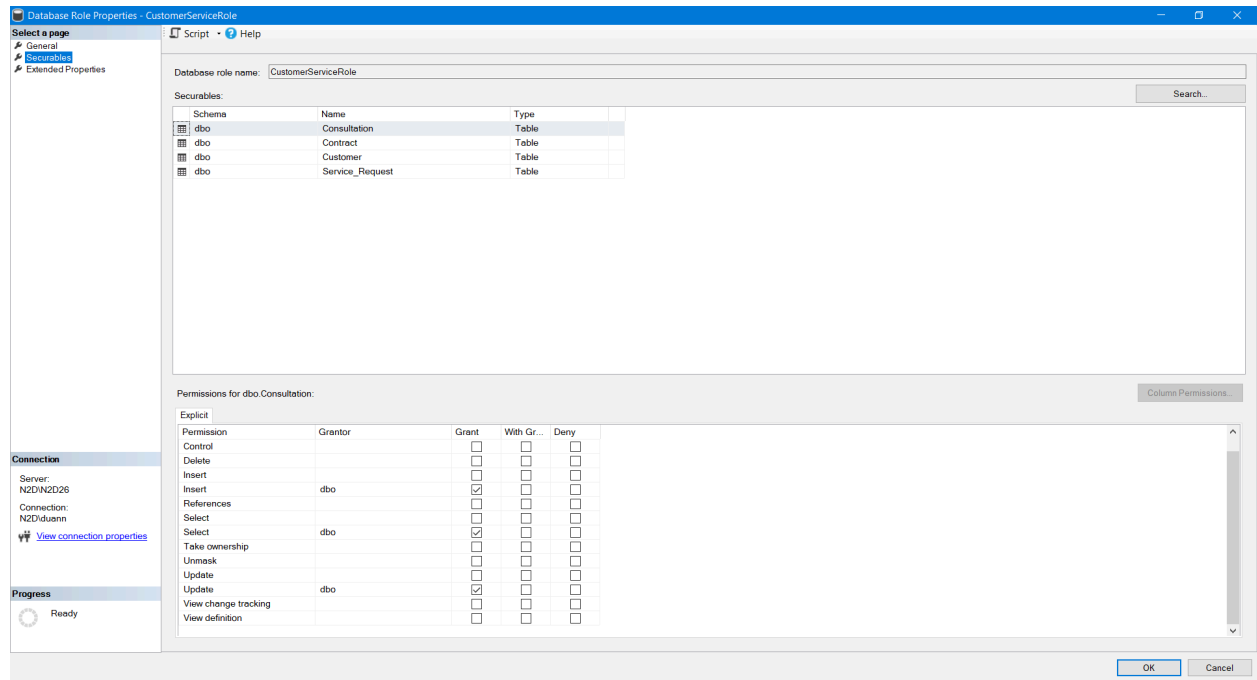
Tạo các role theo phân quyền

CustomerServiceRole

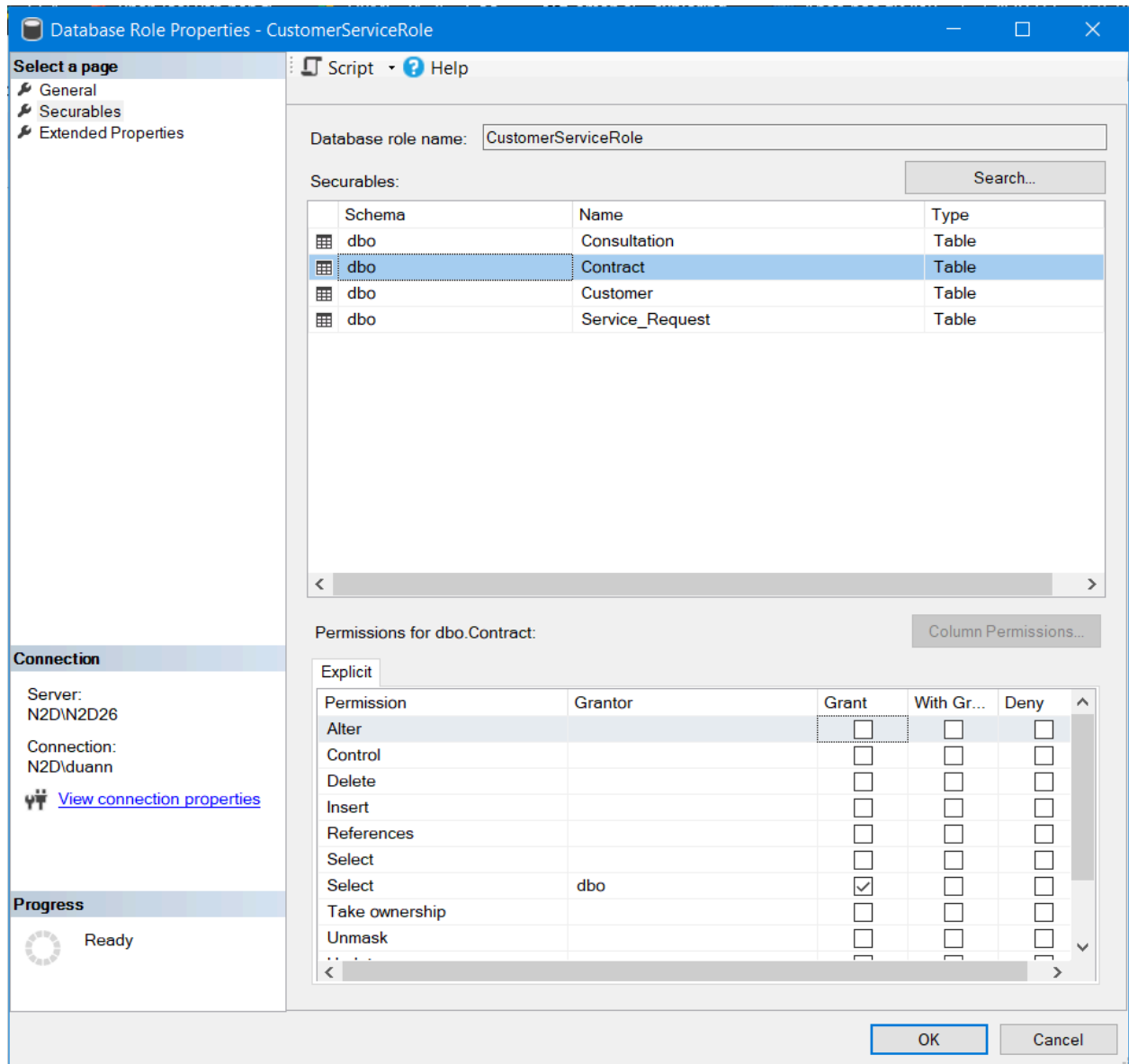


Hình 11: Tạo role CustomerServiceRole

Nhân viên chăm sóc khách hàng có các quyền truy cập vào dữ liệu khách hàng và yêu cầu dịch vụ vì thế các bảng được truy cập bao gồm: CUSTOMER, CONTRACT, CONSULTATION, SERVICE\_REQUEST

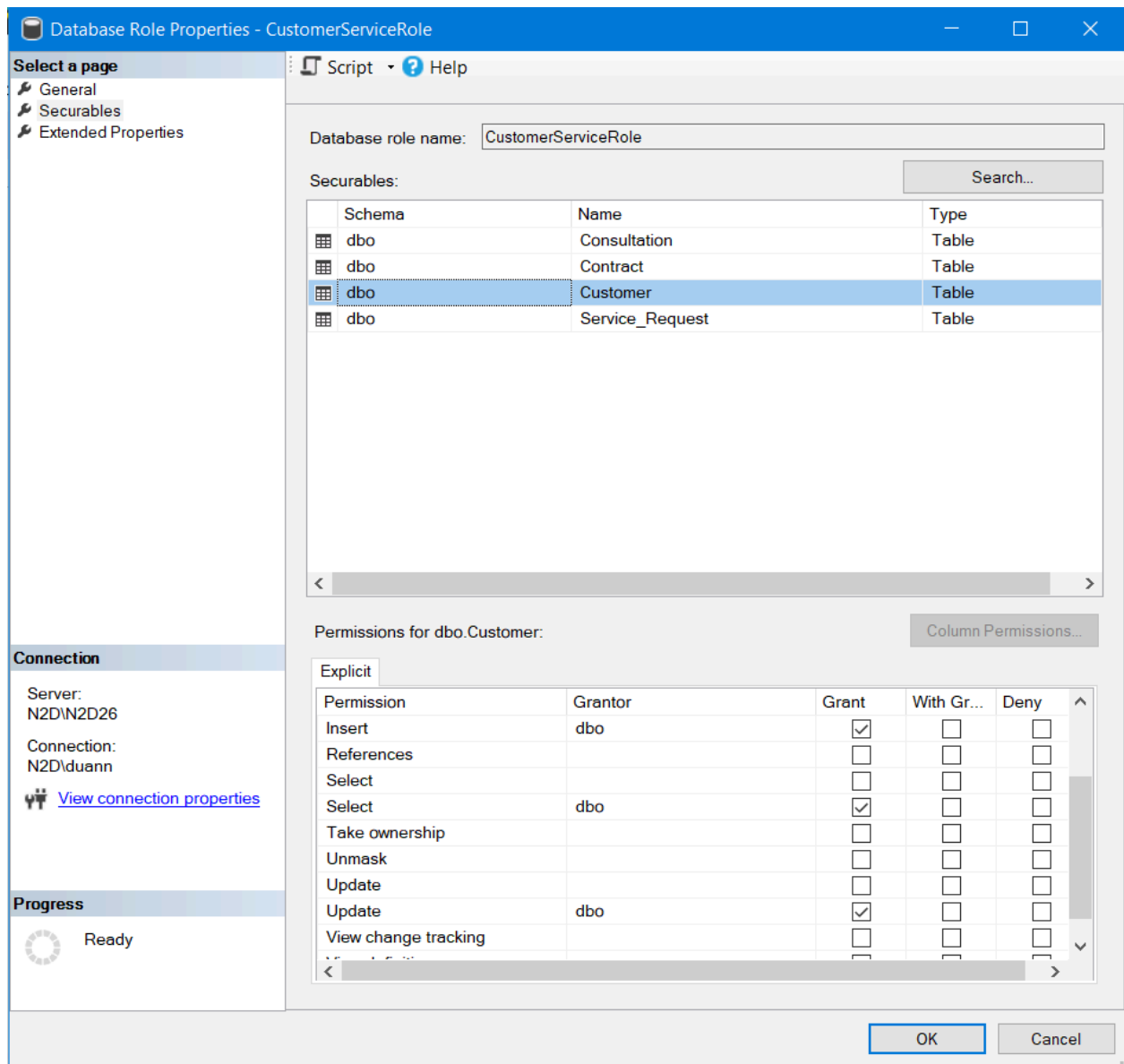


Hình 12: Gán quyền SELECT, INSERT, UPDATE cho role CustomerServiceRole ở bảng CONSULTATION

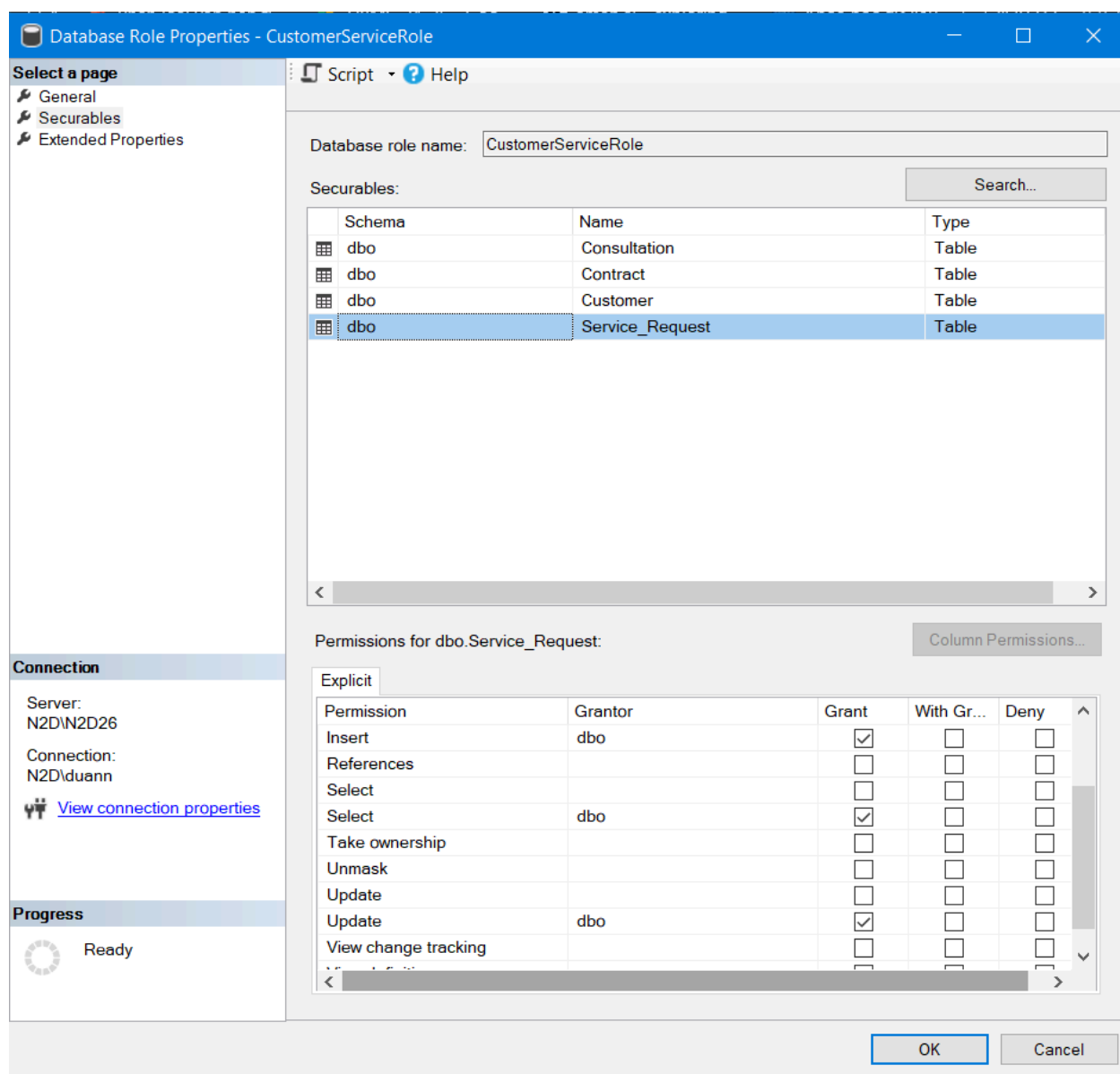


Hình 13: Gán quyền SELECT cho role CustomerServiceRole ở bảng CONTRACT





Hình 14: Gán quyền SELECT, INSERT, UPDATE cho role CustomerServiceRole ở bảng CUSTOMER



Hình 15: Gán quyền SELECT, INSERT, UPDATE cho role CustomerServiceRole ở bảng SERVICE\_REQUEST

Thực hiện tương tự, bước khởi tạo các role khác bằng thao tác dòng lệnh:

```
-- Tạo các Database Roles dựa trên chức năng nghiệp vụ
CREATE ROLE DeclarationProcessorRole;
CREATE ROLE InspectionRole;
CREATE ROLE FinanceRole;
CREATE ROLE ManagementRole;

-- Cấp quyền cho từng vai trò

-- DeclarationProcessorRole: quản lý tờ khai và hàng hóa
GRANT SELECT, INSERT, UPDATE ON Customer_Declaration TO DeclarationProcessorRole;
GRANT SELECT, INSERT, UPDATE ON Goods TO DeclarationProcessorRole;
GRANT SELECT, INSERT, UPDATE ON Goods_Detail TO DeclarationProcessorRole;
GRANT SELECT, INSERT, UPDATE ON Perishable_Goods TO DeclarationProcessorRole;
GRANT SELECT, INSERT, UPDATE ON Hazardous_Goods TO DeclarationProcessorRole;
GRANT SELECT, INSERT, UPDATE ON General_Goods TO DeclarationProcessorRole;

-- InspectionRole: quản lý kiểm tra hàng hóa
GRANT SELECT, INSERT, UPDATE ON Goods_Inspection TO InspectionRole;
GRANT SELECT ON Customer_Declaration TO InspectionRole;
GRANT SELECT ON Goods_Detail TO InspectionRole;
GRANT SELECT ON Goods TO InspectionRole;

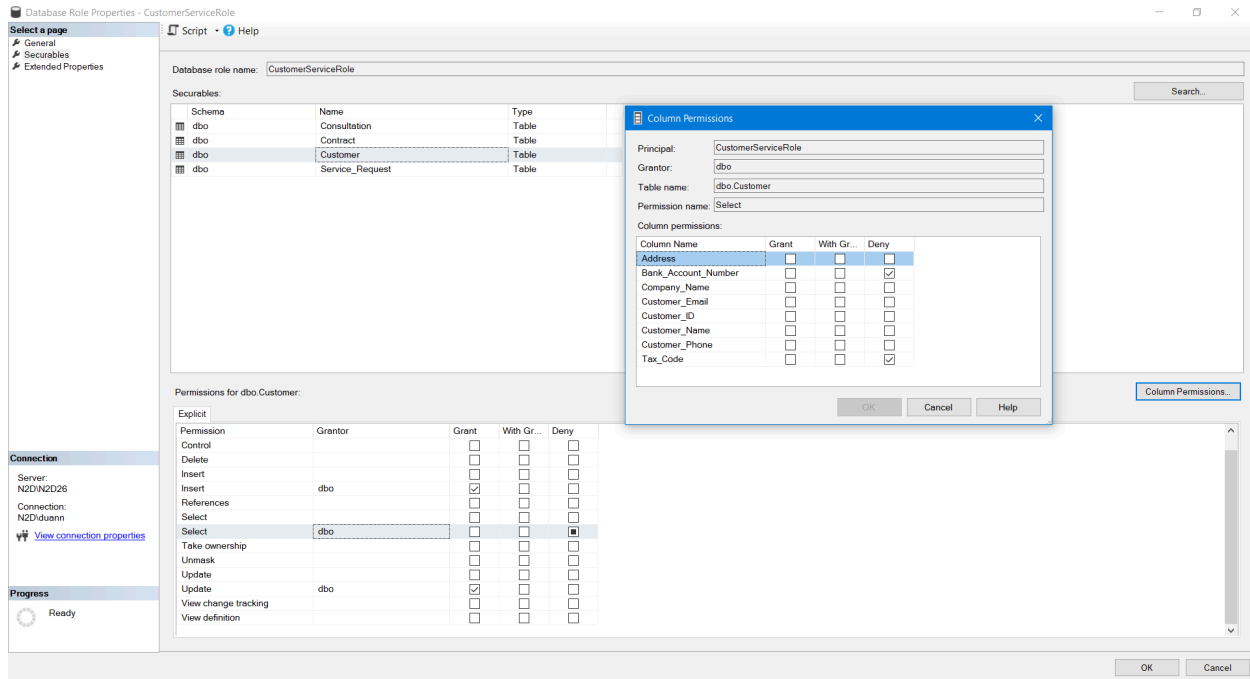
-- FinanceRole: quản lý thanh toán và hợp đồng
GRANT SELECT, INSERT, UPDATE ON Payment TO FinanceRole;
GRANT SELECT, INSERT, UPDATE ON Contract TO FinanceRole;
GRANT SELECT ON Service_Request TO FinanceRole;

-- ManagementRole: toàn quyền truy cập
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::GiuaKy TO ManagementRole;
```

Hình 16: Khởi tạo các role còn lại bằng TSQL

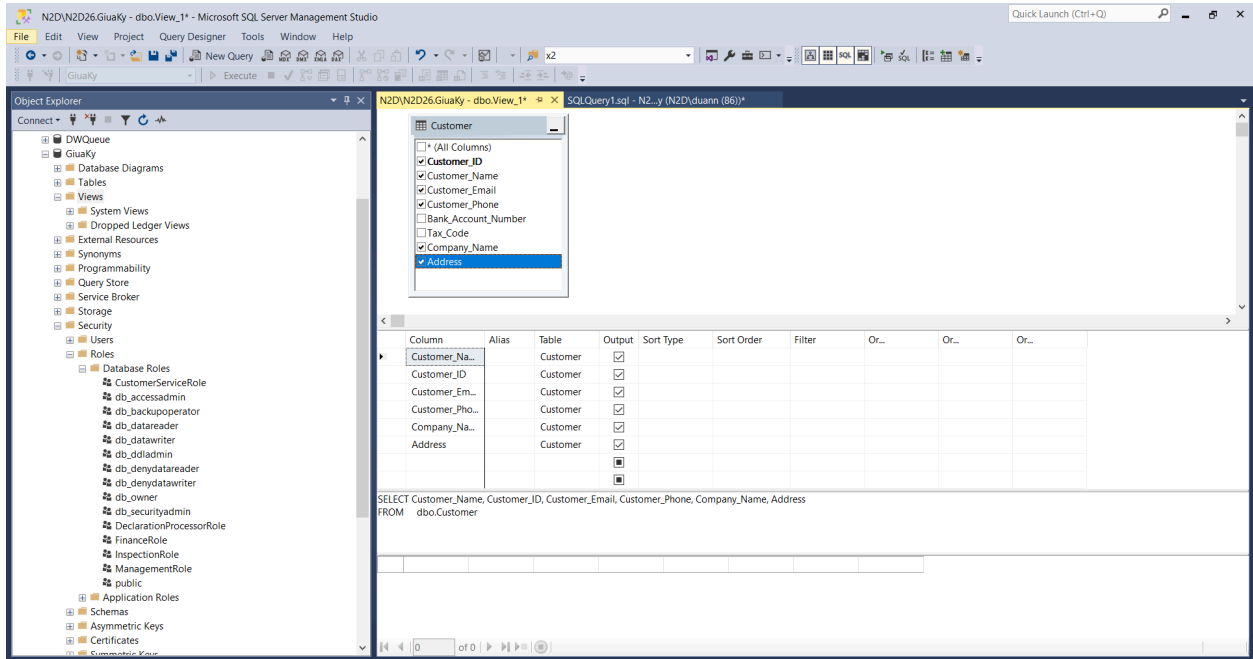
## +Bảo mật cấp cột (Column-Level Security)

Để bảo vệ thông tin nhạy cảm, hệ thống áp dụng bảo mật cấp cột, hạn chế quyền truy cập vào các thông tin tài chính của khách hàng, cụ thể là CustomerServiceRole không được phép xem các thông tin của khách hàng liên quan đến mã số thuế và số tài khoản ngân hàng:



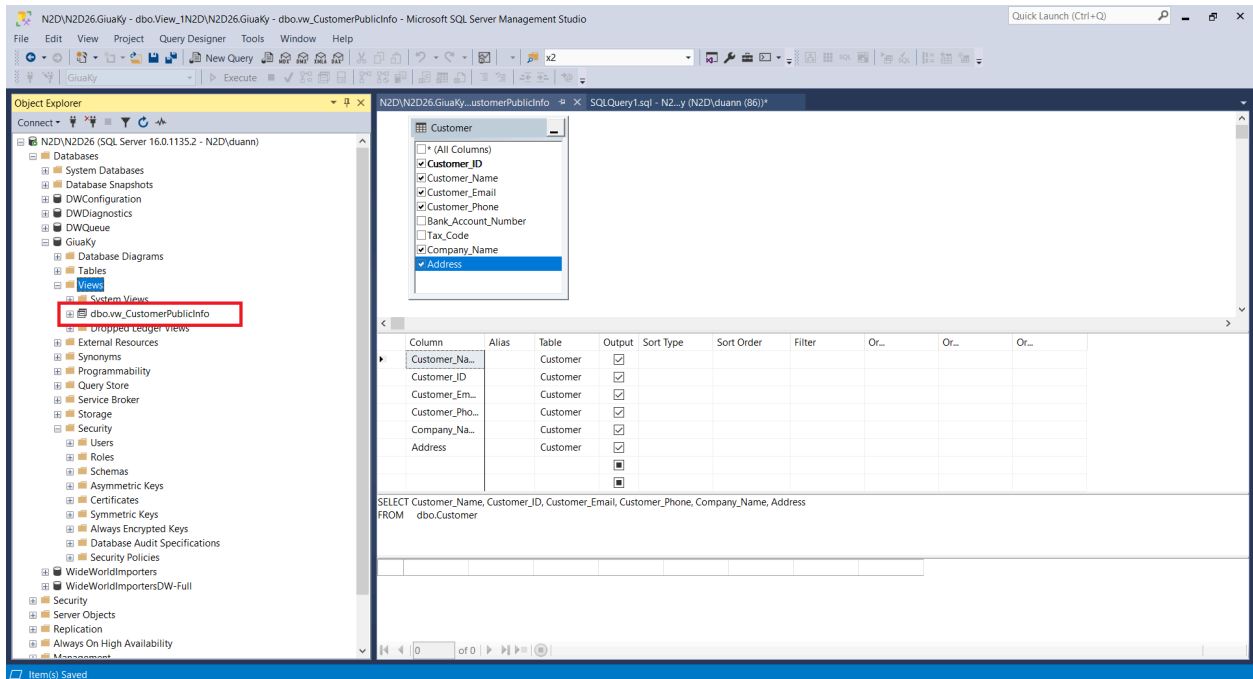
Hình 17: CustomerServiceRole bị giới hạn một số cột ở bảng CUSTOMER

Để thực hiện điều trên và vẫn đảm bảo sự truy cập của CustomerServiceRole vào bảng CUSTOMER, cần tạo View chứa các thông tin cơ bản của CUSTOMER:



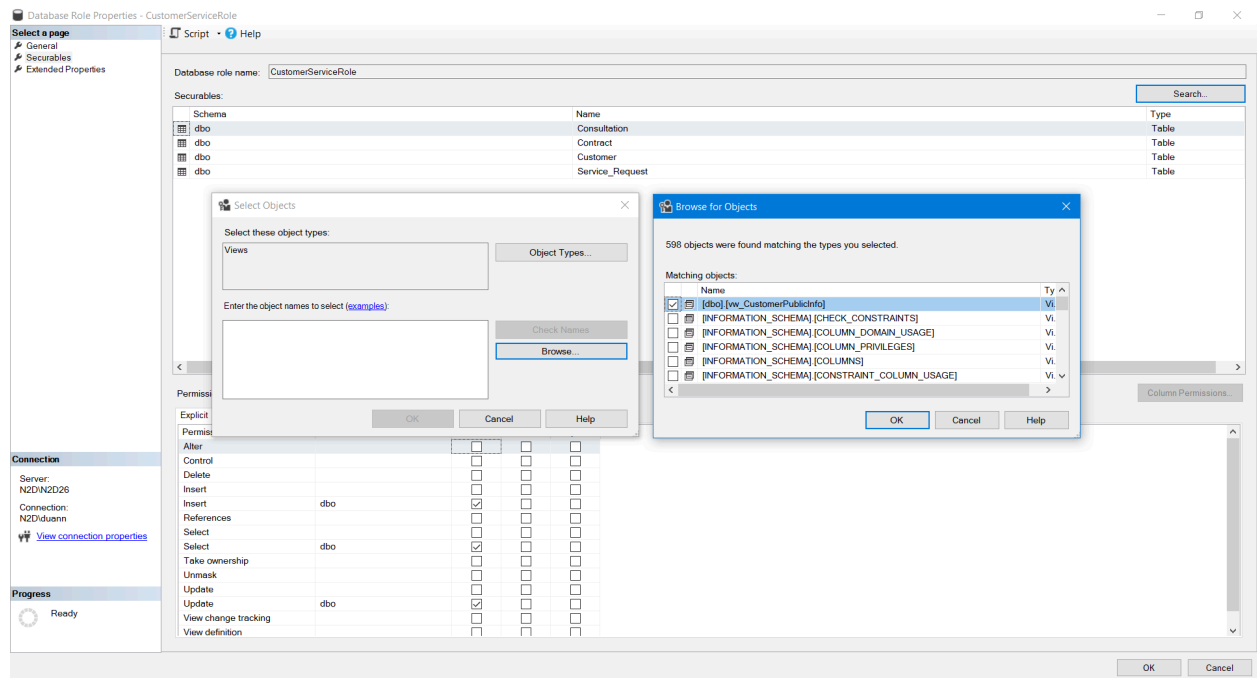
Hình 18: Khởi tạo view về thông tin khách hàng

View được lưu dưới tên là vw\_CustomerPublicInfo, kết quả thu được như hình bên dưới:

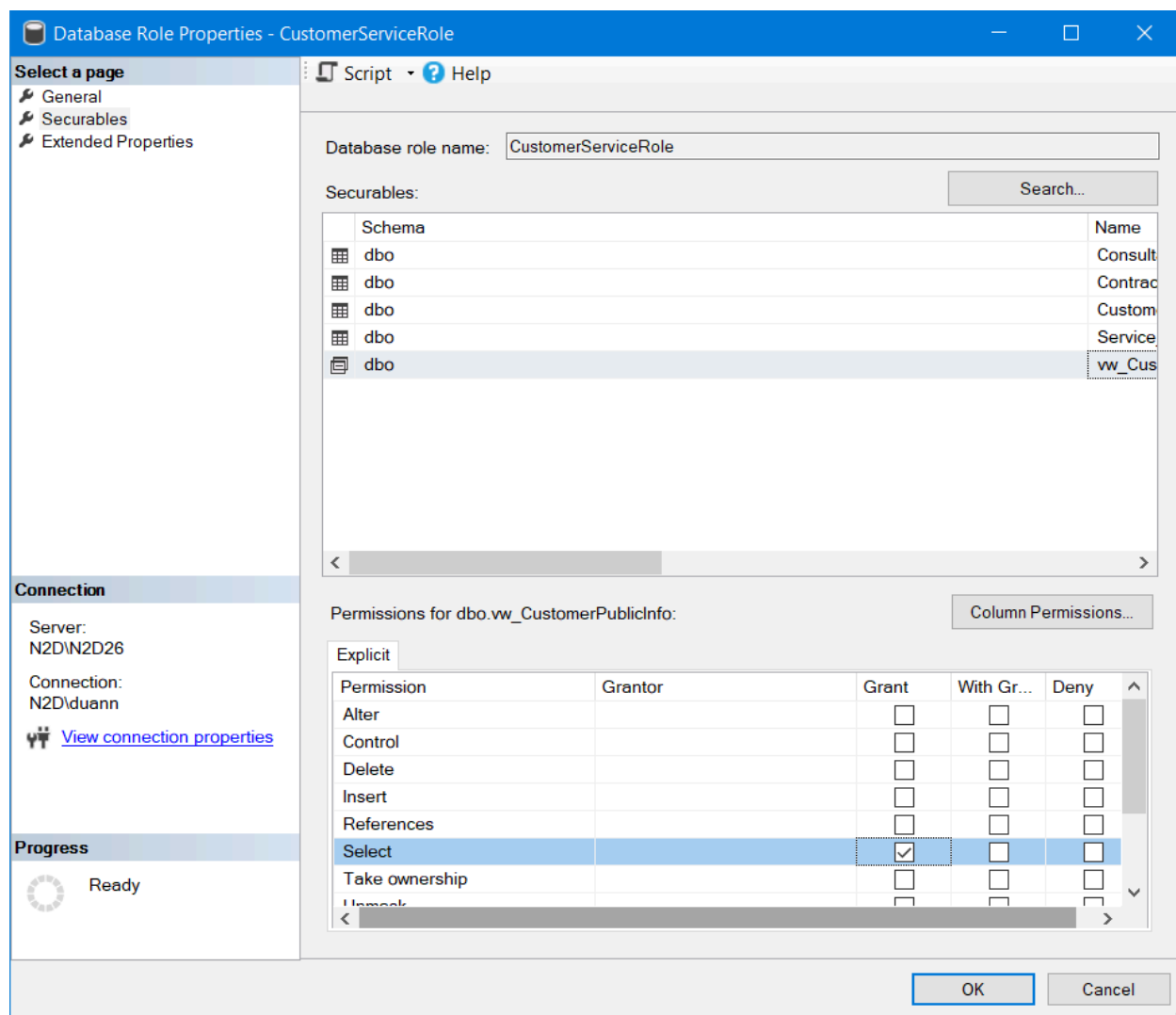


Hình 19: Kết quả sau khi tạo View vw\_CustomerPublicInfo

Tiếp tục gán quyền truy cập view cho CustomerServiceRole



Hình 20: Lựa chọn view cần gán quyền



Hình 21: Gán quyền SELECT cho CustomerServiceRole ở vw\_CustomerPublicInfo

Gán role cho từng User

Bảng Thông tin User

User	Role	Quyền trên schema chính
CS_NguyenVan	CustomerServiceRole	Đầy đủ trên CustomerService
DP_TranThi	DeclarationProcessorRole	Đầy đủ trên Declaration
INSP_LeMinh	InspectionRole	Đầy đủ trên Inspection
FIN_PhamHong	FinanceRole	Đầy đủ trên Finance
MGT_DoanhThanh	ManagementRole	db_owner (toàn bộ database)

Khởi tạo Login:

**Login - New**

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Login name: CS\_NguyenVan Search...

☐ Windows authentication

☐ Microsoft Entra ID authentication

☒ SQL Server authentication

Password: .....

Confirm password: .....

☐ Specify old password

Old password: .....

☒ Enforce password policy

☐ Enforce password expiration

☐ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Add

Mapped Credentials

Credential	Prov
------------	------

Remove

Default database: master

Default language: <default>

OK Cancel

Connection

Server: N2D\N2D26

Connection: N2D\duann

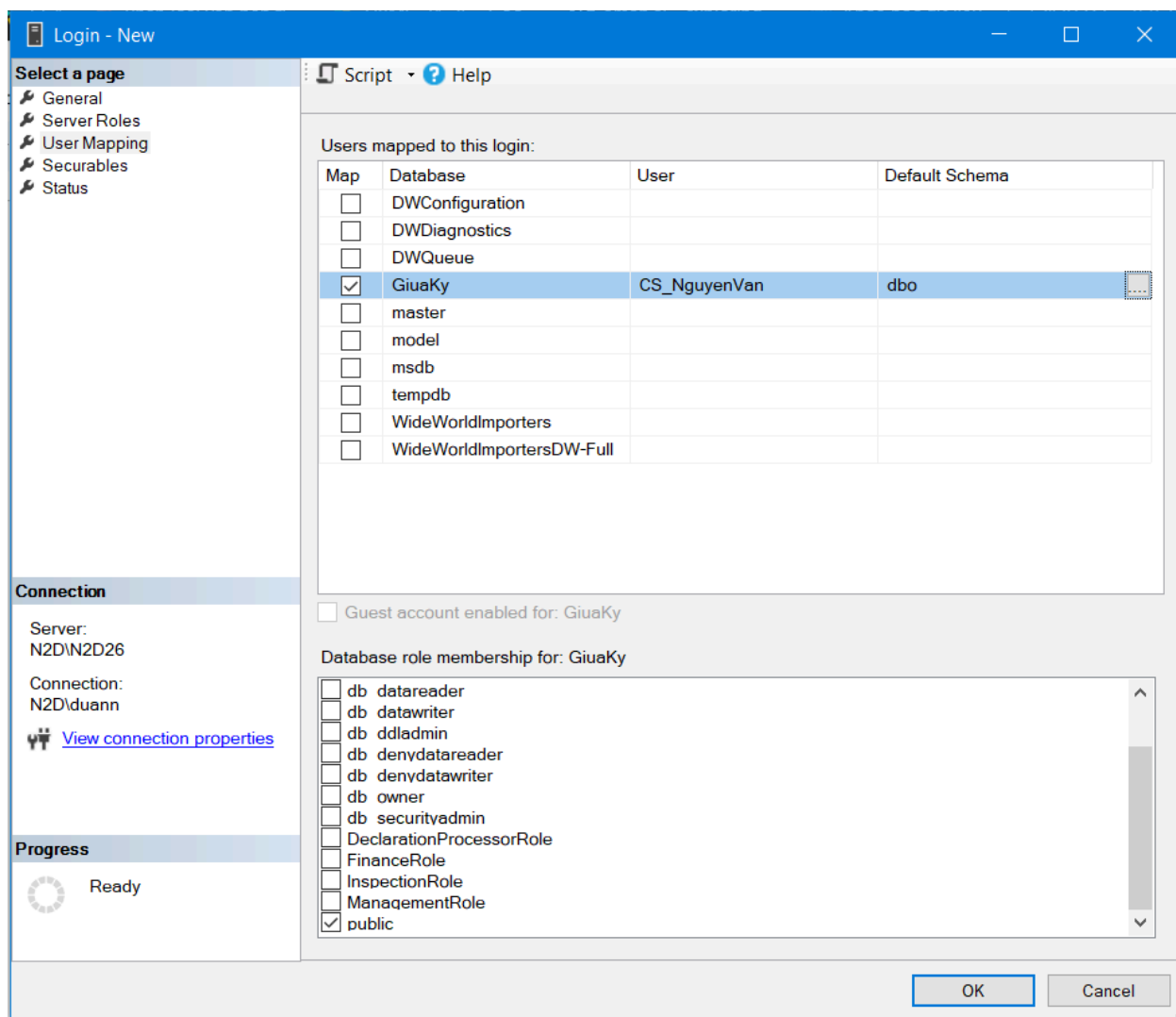
[View connection properties](#)

Progress

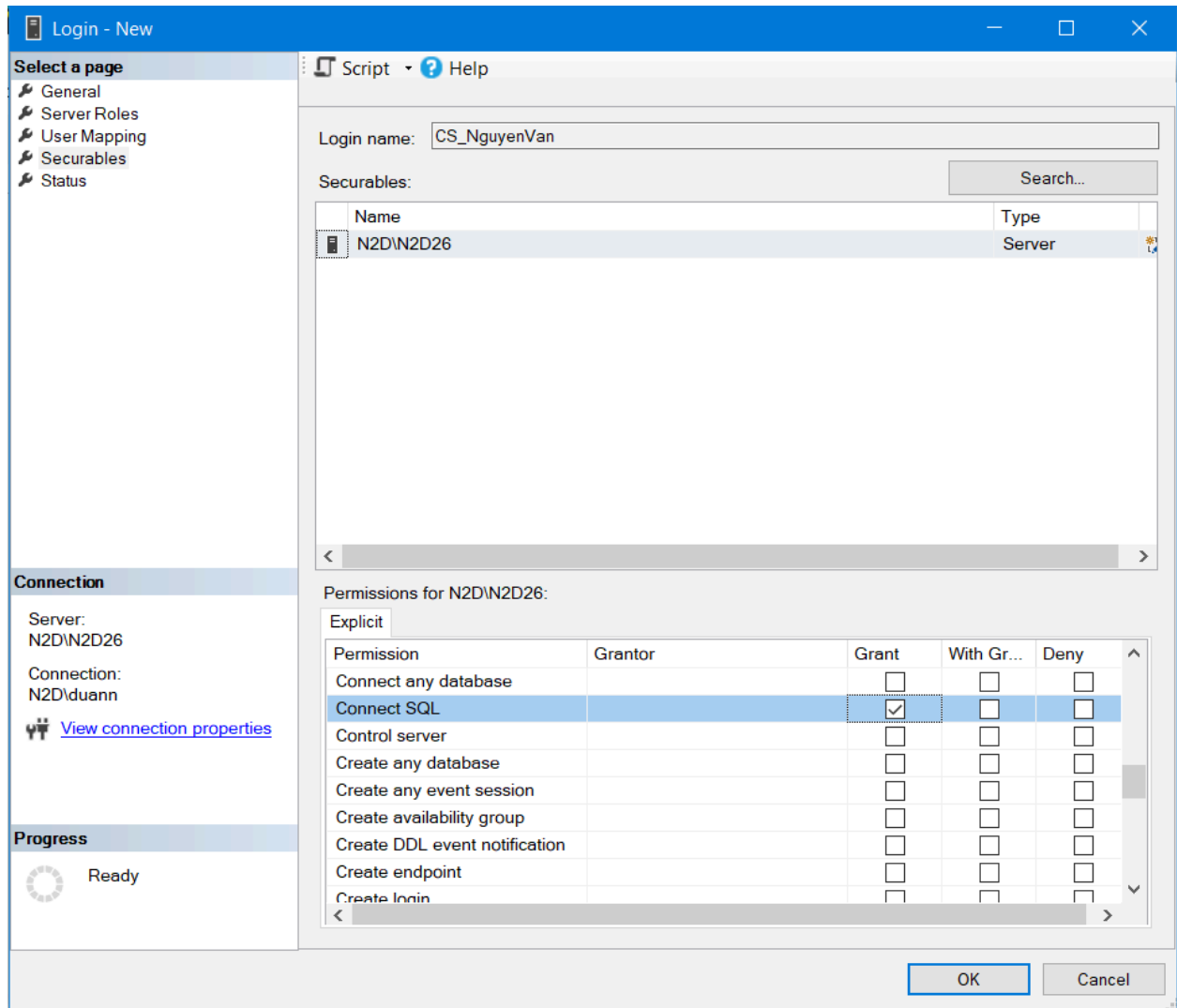
Ready

Hình 22: Tạo Login CS\_NguyenVan





Hình 23: Cấu hình User Mapping CS\_NguyenVan



Hình 24: Gán quyền cho Login

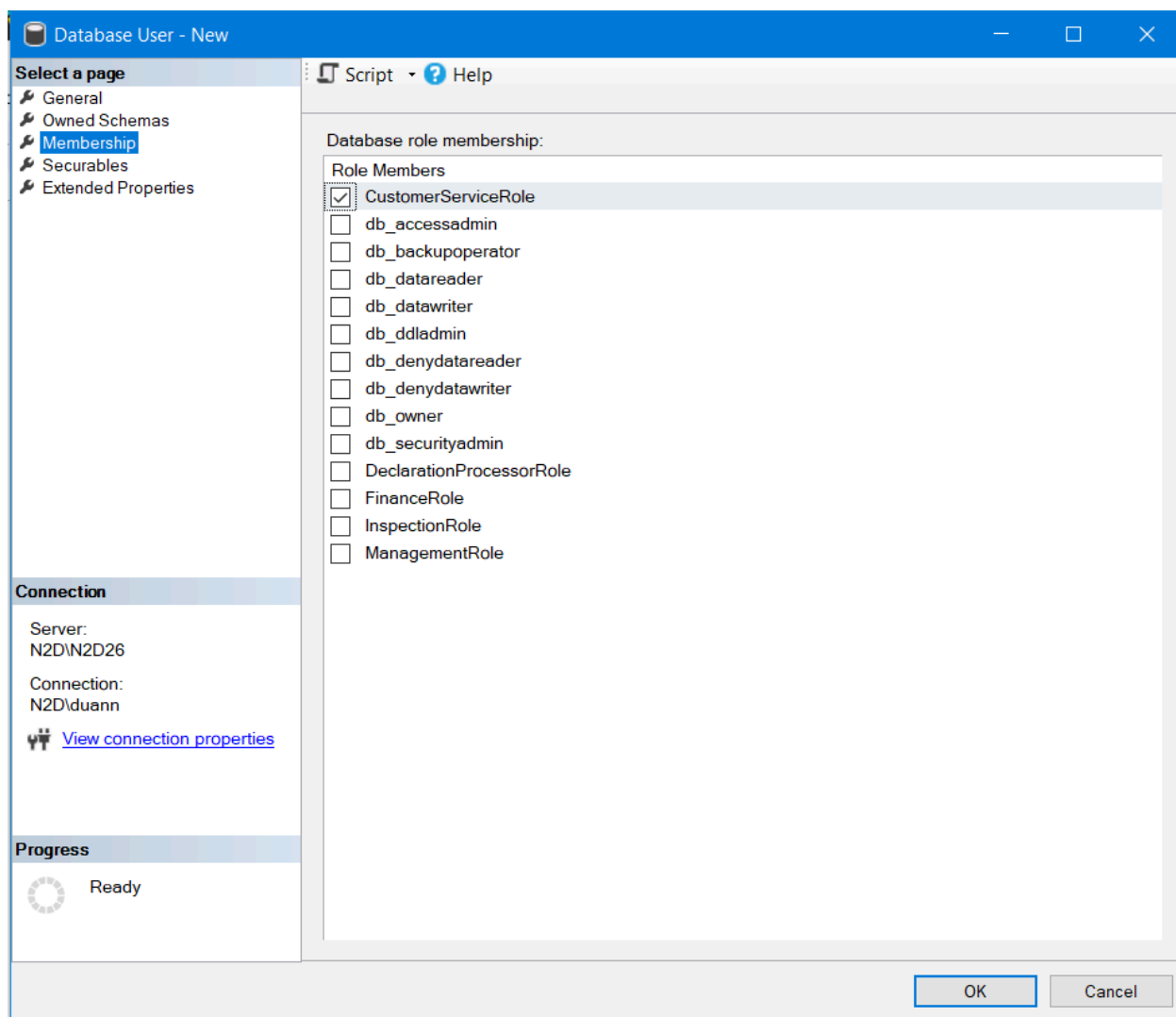
Khởi tạo User:

The screenshot shows the 'Database User - New' dialog box with the following details:

- General Tab:**
  - User type:** SQL user with login
  - User name:** CS\_NguyenVan
  - Login name:** CS\_NguyenVan
  - Default schema:** dbo
- Connection:**
  - Server: N2D\N2D26
  - Connection: N2D\duann
  - [View connection properties](#)
- Progress:** Ready

Buttons: OK, Cancel

Hình 25: Tạo User CS\_NguyenVan



Hình 26: Chọn Role Membership cho User CS\_NguyenVan

Thực hiện tương tự với các User khác bằng thao tác dòng lệnh:

```
-- Tạo login và user cho Declaration Processor
CREATE LOGIN DP_TranThi WITH PASSWORD = 'P@ssw0rd123DP';
CREATE USER DP_TranThi FOR LOGIN DP_TranThi;
ALTER ROLE DeclarationProcessorRole ADD MEMBER DP_TranThi;

-- Tạo login và user cho Inspection
CREATE LOGIN INSP_LeMinhWithPassword = 'P@ssw0rd123INSP';
CREATE USER INSP_LeMinh FOR LOGIN INSP_LeMinh;
ALTER ROLE InspectionRole ADD MEMBER INSP_LeMinh;

-- Tạo login và user cho Finance
CREATE LOGIN FIN_PhamHong WITH PASSWORD = 'P@ssw0rd123FIN';
CREATE USER FIN_PhamHong FOR LOGIN FIN_PhamHong;
ALTER ROLE FinanceRole ADD MEMBER FIN_PhamHong;

-- Tạo login và user cho quản lý cao cấp
CREATE LOGIN MGT_DoanhThanh WITH PASSWORD = 'P@ssw0rd123MGT';
CREATE USER MGT_DoanhThanh FOR LOGIN MGT_DoanhThanh;
ALTER ROLE ManagementRole ADD MEMBER MGT_DoanhThanh;

-- Thêm cấu hình bảo mật cho các tài khoản
ALTER LOGIN DP_TranThi WITH CHECK_POLICY = ON, CHECK_EXPIRATION = ON;
ALTER LOGIN INSP_LeMinh WITH CHECK_POLICY = ON, CHECK_EXPIRATION = ON;
ALTER LOGIN FIN_PhamHong WITH CHECK_POLICY = ON, CHECK_EXPIRATION = ON;
ALTER LOGIN MGT_DoanhThanh WITH CHECK_POLICY = ON, CHECK_EXPIRATION = ON;
```

Hình 27: Tạo các Login và User khác còn thiếu

### +Theo dõi truy cập (Audit)

Trong kiến trúc bảo mật cơ sở dữ liệu hiện đại, việc thiết lập một schema riêng biệt cho quản lý và một bảng ghi lại nhật ký hoạt động người dùng đóng vai trò then chốt. Vì vậy, cần tạo schema Management cùng bảng UserActivityLog nhằm đảm bảo tính toàn vẹn và bảo mật dữ liệu, dưới sự quản lý và truy cập của ManagementRole. Bảng UserActivityLog được thiết kế để lưu trữ thông tin chi tiết về mọi tương tác của người dùng với hệ thống, bao gồm tên người dùng, loại hành động, đối tượng tác động, thời gian, địa chỉ IP và mô tả chi tiết.

```

CREATE SCHEMA Management;
CREATE TABLE Management.UserActivityLog (
    LogID INT IDENTITY(1,1) PRIMARY KEY,
    UserName NVARCHAR(128),
    ActionType NVARCHAR(50),
    TableName NVARCHAR(128),
    ActionDateTime DATETIME DEFAULT GETDATE(),
    IPAddress NVARCHAR(50),
    Description NVARCHAR(MAX)
);
GRANT SELECT, INSERT, UPDATE, DELETE ON Management.UserActivityLog TO ManagementRole;

```

Hình 28: Tạo bảng log và gán quyền truy cập cho ManagementRole

Hệ thống được cấu hình để ghi nhận và theo dõi mọi hoạt động truy cập dữ liệu quan trọng:

```

CREATE SERVER AUDIT HaiQuanSystemAudit
TO FILE (FILEPATH = 'E:\SQLLog', MAXSIZE = 100MB);

CREATE DATABASE AUDIT SPECIFICATION HaiQuanDBChanges
FOR SERVER AUDIT HaiQuanSystemAudit
ADD (INSERT, UPDATE, DELETE, SELECT ON Customer BY CustomerServiceRole),
ADD (INSERT, UPDATE, DELETE ON Customer_Declaration BY DeclarationProcessorRole),
ADD (INSERT, UPDATE, DELETE ON Payment BY FinanceRole);

ALTER SERVER AUDIT HaiQuanSystemAudit WITH (STATE = ON);
ALTER DATABASE AUDIT SPECIFICATION HaiQuanDBChanges WITH (STATE = ON);

```

Hình 29: SQL Server Audit ghi nhận các hoạt động truy cập dữ liệu

Để truy vấn vào server audit, sử dụng lệnh sau:

```
SELECT * FROM sys.fn_get_audit_file('E:\SQLLog *.sqlaudit', DEFAULT, DEFAULT);
```

### + Encryption

Bảng giải pháp mã hóa

Loại dữ liệu	Giải pháp mã hóa	Lý do chọn
Thông tin khách hàng ( Customer_Phone, Bank_Account_Number, )	CLE (AES-256)	<ul style="list-style-type: none"> <li>Dữ liệu khách hàng như số tài khoản ngân hàng, mã số thuế, email có thể bị lạm dụng nếu rò rỉ.</li> <li>AES-256 là thuật toán mạnh mẽ, giúp bảo vệ</li> </ul>

		dữ liệu khỏi truy cập trái phép. <ul style="list-style-type: none"> <li>• CLE cho phép mã hóa từng cột trong bảng mà không ảnh hưởng đến toàn bộ database.</li> </ul>
Toàn bộ database	<b>TDE</b>	<ul style="list-style-type: none"> <li>• Bảo vệ database khỏi bị sao chép trái phép (data at rest encryption).</li> <li>• TDE mã hóa toàn bộ file dữ liệu và log file, giúp bảo vệ khi backup database.</li> <li>• Nếu hacker đánh cắp file .mdf hoặc .ldf, họ không thể đọc dữ liệu.</li> </ul>

Trước khi đi vào mã hóa bằng cấp độ DB và CLE, cần tạo master key, certificate, và backup file để đảm bảo hệ thống có lớp bảo vệ an toàn, cung cấp khóa chính để mã hóa và bảo vệ các khóa mã hóa khác, hỗ trợ khôi phục dữ liệu nếu xảy ra sự cố, tạo chứng chỉ giúp kiểm soát quyền truy cập vào dữ liệu đã mã hóa, đảm bảo tính toàn vẹn và bảo mật khi thực hiện mã hóa, cho phép giải mã dữ liệu an toàn khi cần thiết và đáp ứng các yêu cầu bảo mật cũng như quy định về bảo vệ dữ liệu trong doanh nghiệp.

*--Bước 1: Tạo Master Key*

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Pa$$w0rd';
```

*-- Bước 2: Tạo Certificate*

```
CREATE CERTIFICATE TDECert WITH SUBJECT = 'Certificate For TDE';
```

*-- Bước 3: Tạo Symmetric Key và Mã Hóa Dữ Liệu*

```
CREATE SYMMETRIC KEY MyAES256Key
WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE TDECert;
```

GO

-- Bước 4: Mã Hóa Dữ Liệu

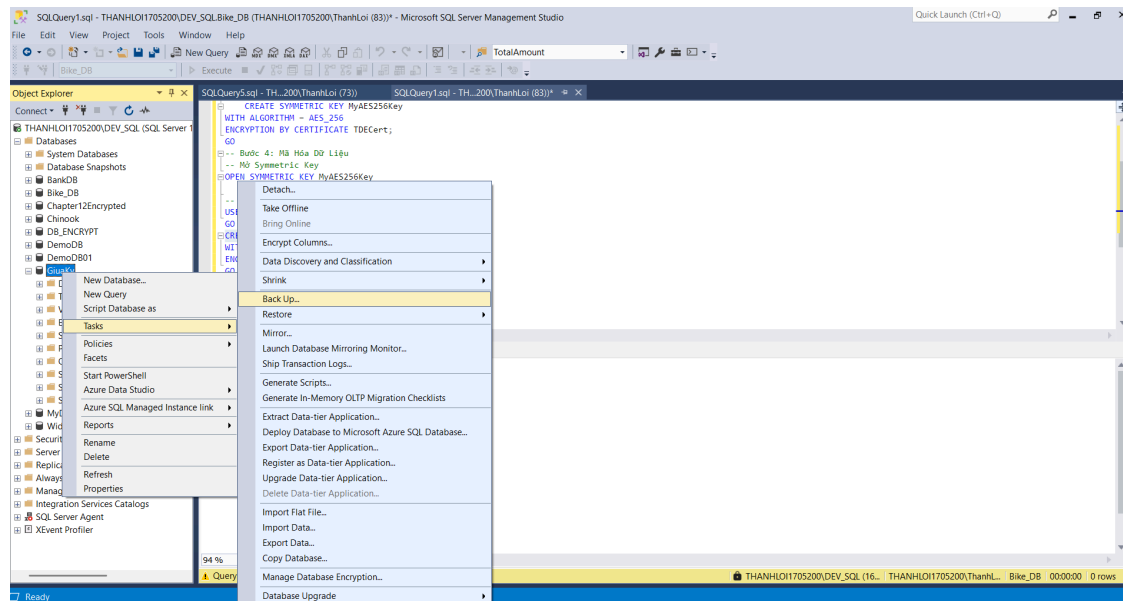
-- Mở Symmetric Key

OPEN SYMMETRIC KEY MyAES256Key

DECRYPTION BY CERTIFICATE TDECert;

*Bước 5: Backup Full và Backup Transaction Log*

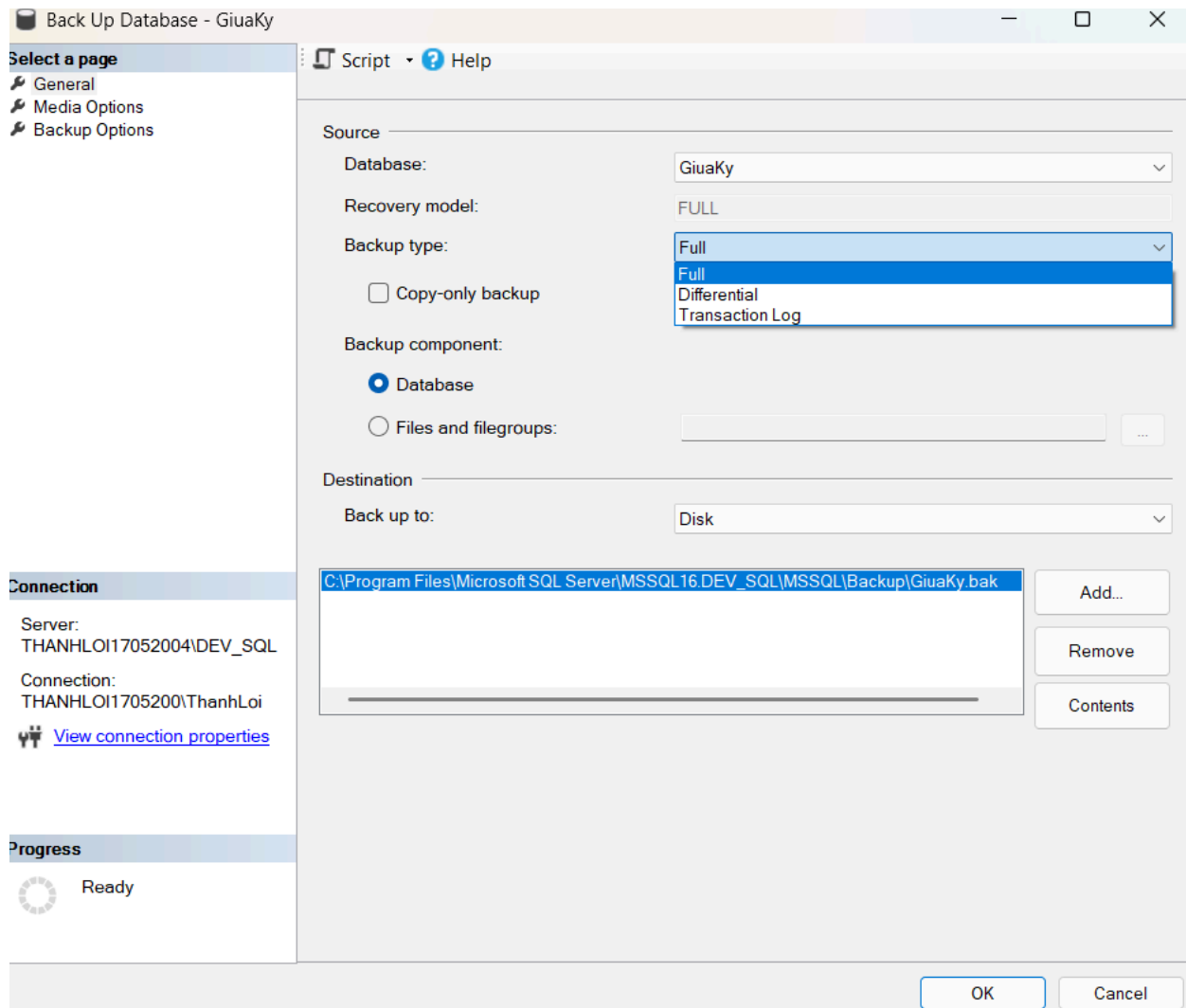
Chuột phải vào DB và vào task → backup



Hình 30: Thao tác back up

Thực hiện 2 lần backup “Full” và “Transaction Log” rồi nhấn Ok





Hình 31: Thực hiện 2 lần backup “Full” và “Transaction Log” rồi nhấn Ok

- -Bước 6 : Sao lưu Certificate và Private Key

BACKUP CERTIFICATE TDE\_Certificate

TO FILE = 'C:\Backup\TDE\_Certificate.bak'

WITH PRIVATE KEY (

FILE = 'C:\Backup\TDE\_PrivateKey.pvk',

ENCRYPTION BY PASSWORD = 'Mật\_khẩu\_Bảo\_mật\_Khóa');

-- Bước 7: Tạo DEK

USE GiuaKy;

GO

CREATE DATABASE ENCRYPTION KEY

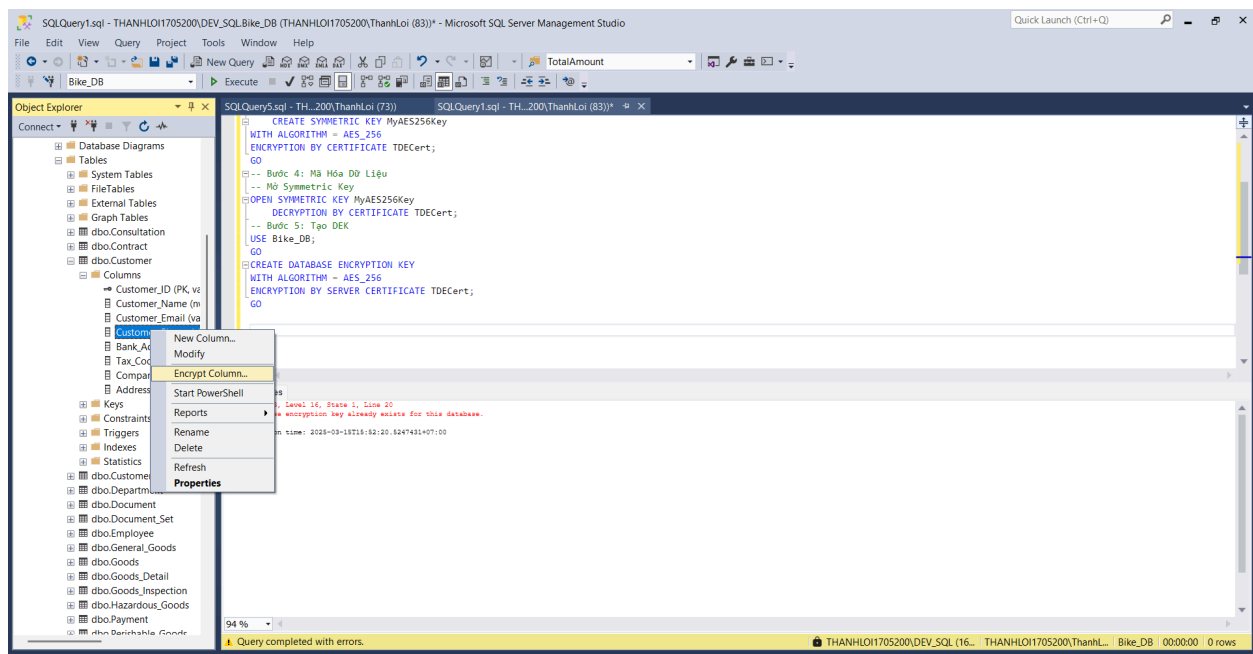
WITH ALGORITHM = AES\_256

ENCRYPTION BY SERVER CERTIFICATE TDECert;

GO

Tiến hành mã hóa CLE

– Chuột phải vào cột cần mã hóa và chọn Encrypt Column



Hình 32: chọn Encrypt Column

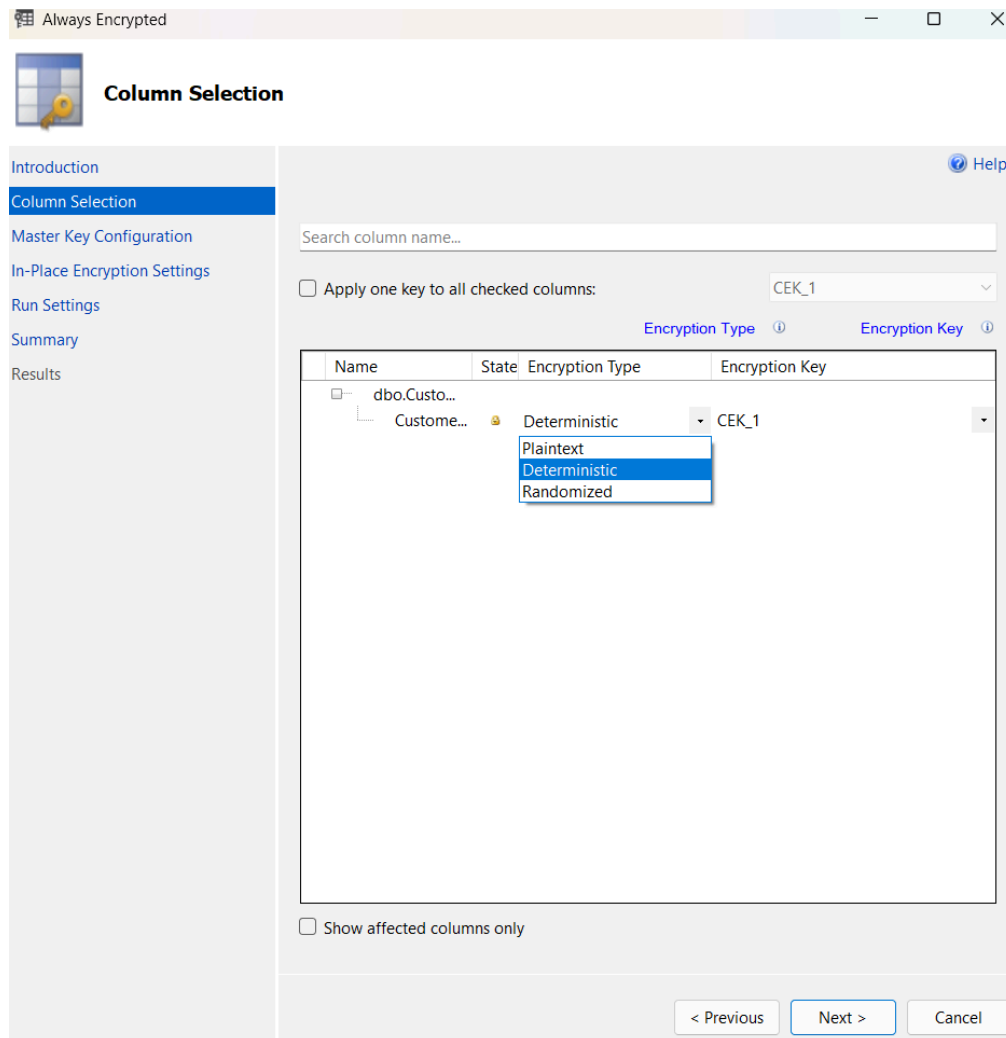
Chọn Encryption type và Encryption Key và ấn 5 lần next rồi Ok

Plaintext: Đây là dữ liệu không mã hóa, được lưu trữ ở dạng gốc và ai có quyền truy cập vào cơ sở dữ liệu cũng có thể đọc được dữ liệu này ngay lập tức cho nên an toàn cho các thông tin nhạy cảm.

Deterministic: Mã hóa dữ liệu theo cách mà giá trị giống nhau sẽ luôn được mã hóa thành một kết quả giống nhau. Cho phép thực hiện các truy vấn như tìm kiếm hoặc so sánh (ví dụ: WHERE column = 'value' vẫn hoạt động). Tuy nhiên, điều này dễ bị tấn công nếu kẻ xâm nhập đoán được các mẫu dữ liệu lặp lại.

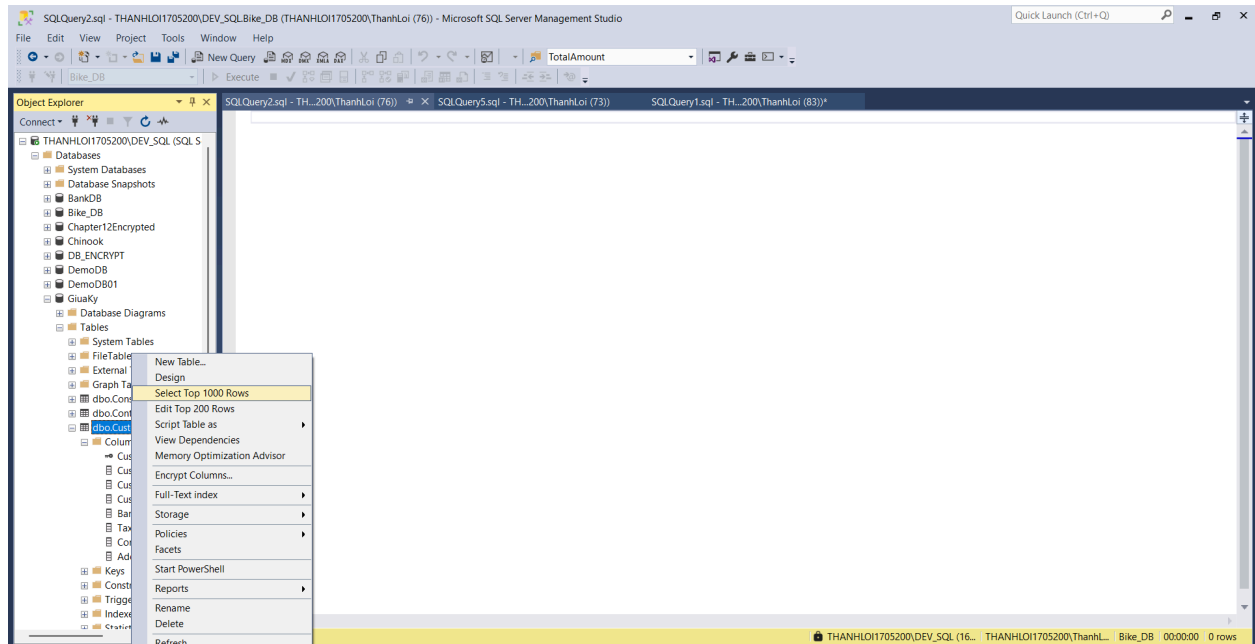
Randomized: Mã hóa dữ liệu sao cho giá trị giống nhau sẽ được mã hóa thành kết quả khác nhau mỗi lần. Kiểu mã hóa này an toàn hơn so với deterministic vì khó đoán được giá trị ban đầu. Tuy

nhiên, hạn chế là không thể thực hiện các truy vấn so sánh trực tiếp (ví dụ: không thể sử dụng WHERE với giá trị đã mã hóa).



Hình 33: Chọn Encryption type và Encryption Key

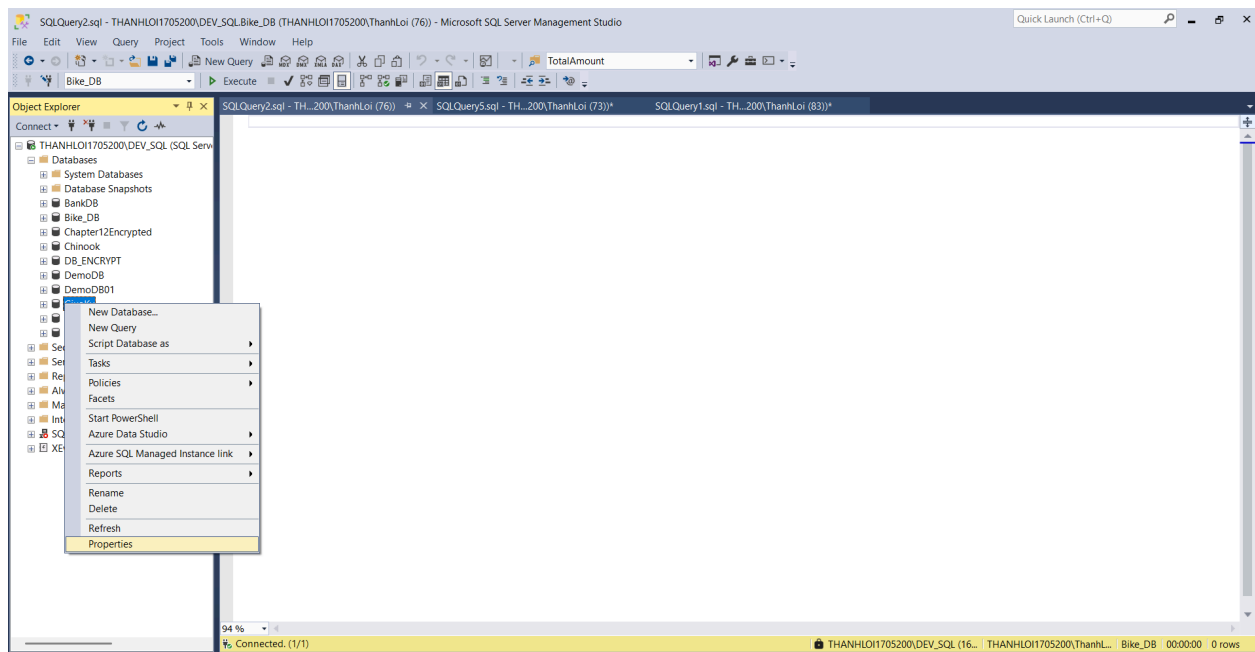
Truy vấn vào dữ liệu bảng Customer xem đã được mã hóa hay chưa, nếu dữ liệu được chuyển thành các ký tự mã hóa thì mã hóa thành công.



Hình 34: Truy vấn vào dữ liệu

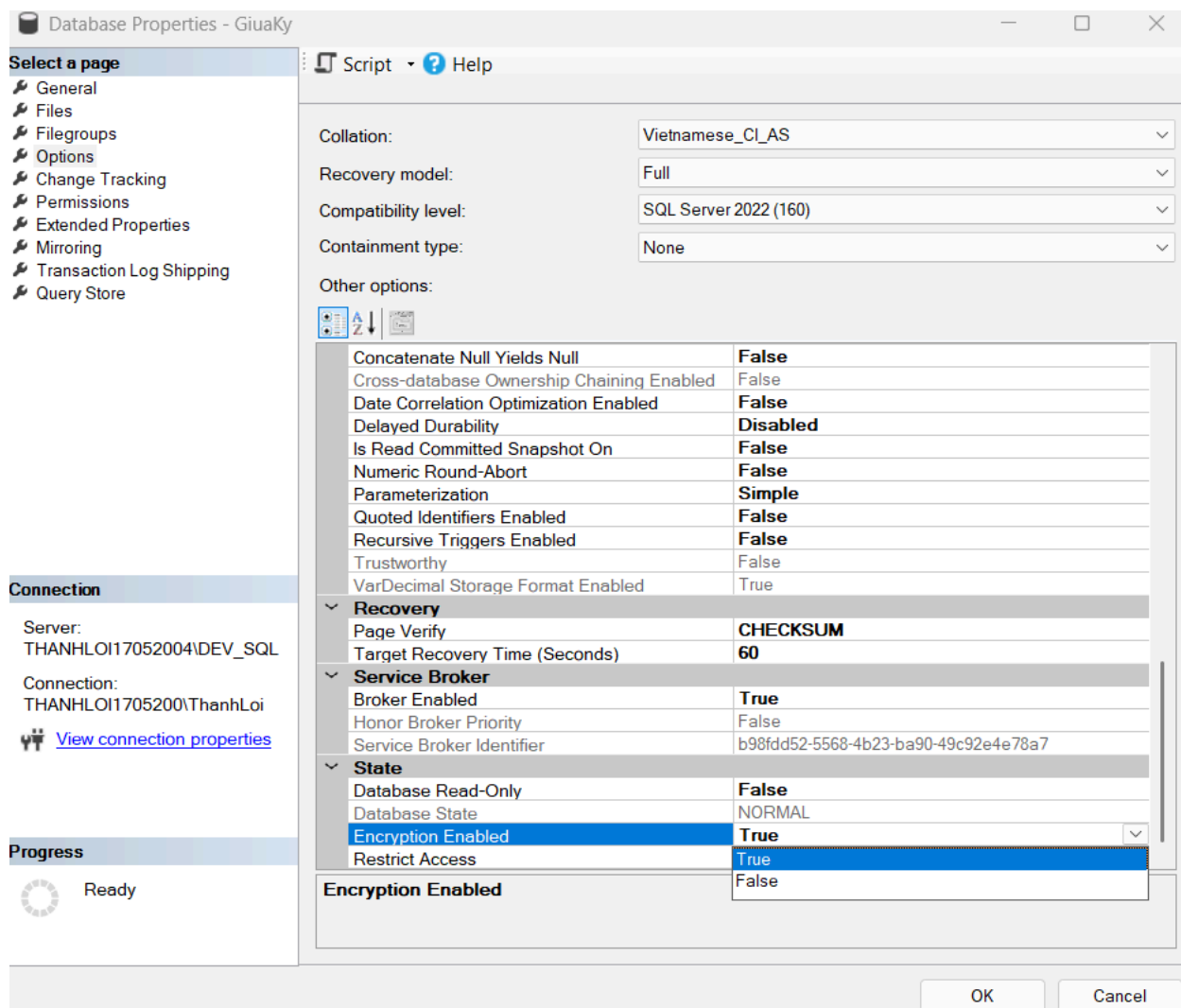
Tiến hành mã hóa Database

Chuột phải vào DB cần mã hóa và chọn properties



Hình 35: Thao tác mã hóa DB

Chọn Options, set True cho Encryption Enabled và ấn Ok

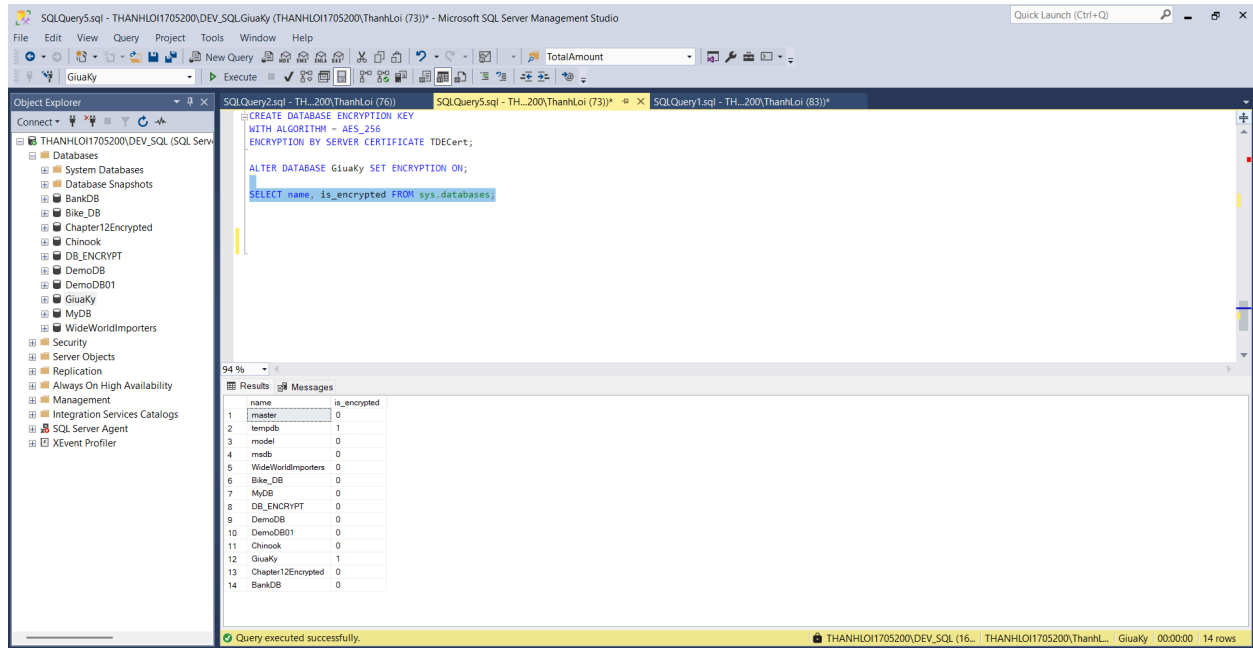


Hình 36: Chọn Options, set True cho Encryption Enabled

Kiểm tra xem DB đã được mã hóa chưa bằng cách chạy lệnh

SELECT name, is\_encrypted FROM sys.databases;

Nếu cột is\_encrypted thành số 1 thì mã hóa đã thành công



Hình 37: Kiểm tra xem DB đã được mã hóa chưa

## + Backups and Restores

Đảm bảo tính liên tục của dịch vụ là yếu tố quan trọng trong hệ thống quản lý thủ tục hải quan.

Sau đây là các chiến lược sao lưu và khôi phục được triển khai.

Chiến lược sao lưu phân lớp (Tiered Backup Strategy)

Hệ thống áp dụng chiến lược sao lưu phân lớp với ba cấp độ:

Sao lưu toàn bộ (Full backup) - Hàng tuần

New Job

Select a page

- General
- Steps
- Schedules
- Alerts
- Notifications
- Targets

Script ? Help

Name: GiuaKyFullBackupWeekly

Owner: MGT\_DoanhThanh

Category: [Uncategorized (Local)]

Description:

☒ Enabled

Connection

Server: N2D\N2D26

Connection: N2D\duann

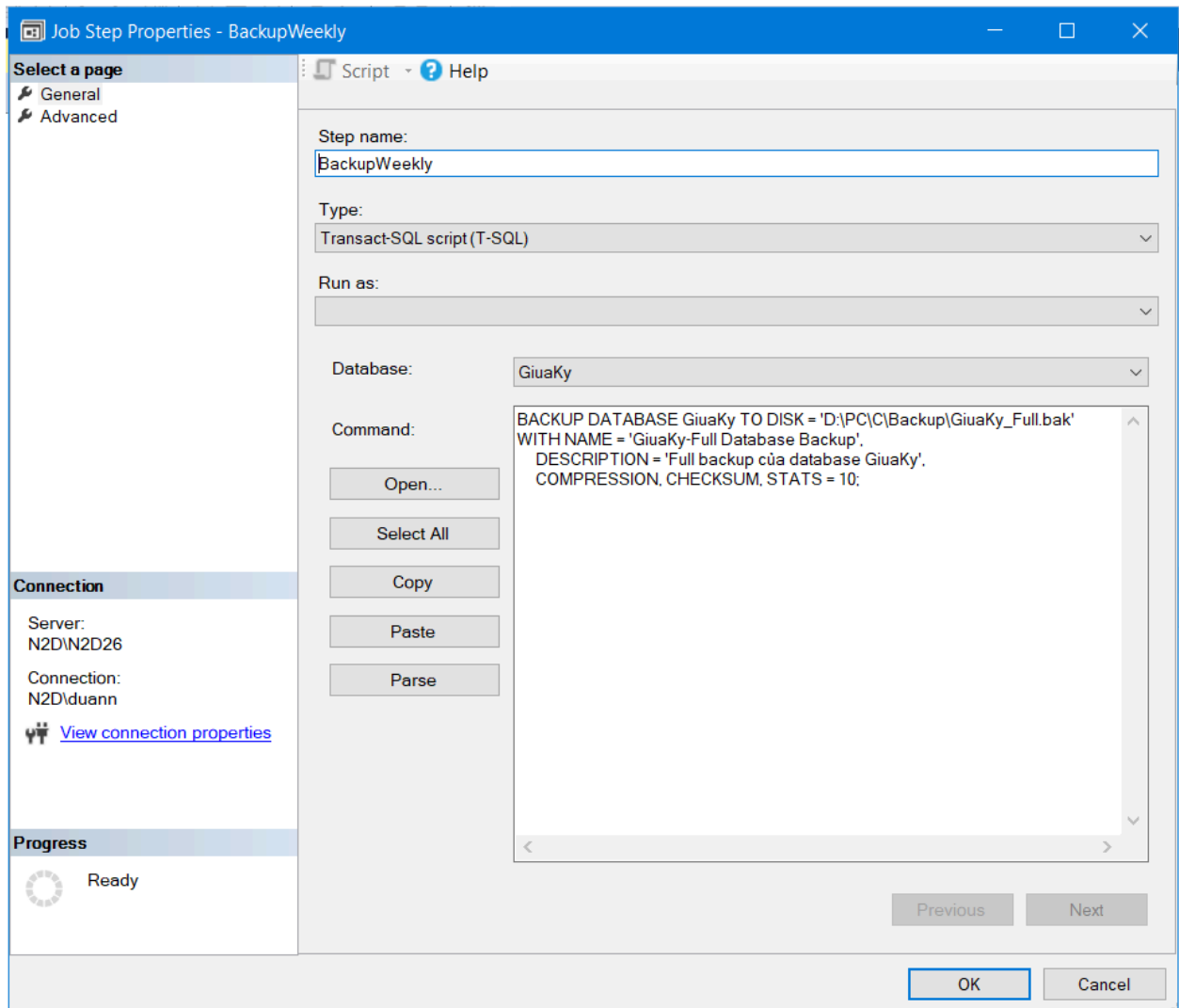
[View connection properties](#)

Progress

Ready

OK Cancel

Hình 38: Tạo job backup weekly cho account manager



Hình 39: Tạo step backup weekly



**Job Schedule Properties - Weekly**

Name:  Jobs in Schedule

Schedule type:  ☒ Enabled

One-time occurrence

Date:  Time:

Frequency

Occurs:

Recurs every:  week(s) on

☐ Monday ☐ Wednesday ☐ Friday ☐ Saturday

☒ Tuesday ☐ Thursday ☐ Sunday

Daily frequency

☒ Occurs once at:

☐ Occurs every:  hour(s) Starting at:  Ending at:

Duration

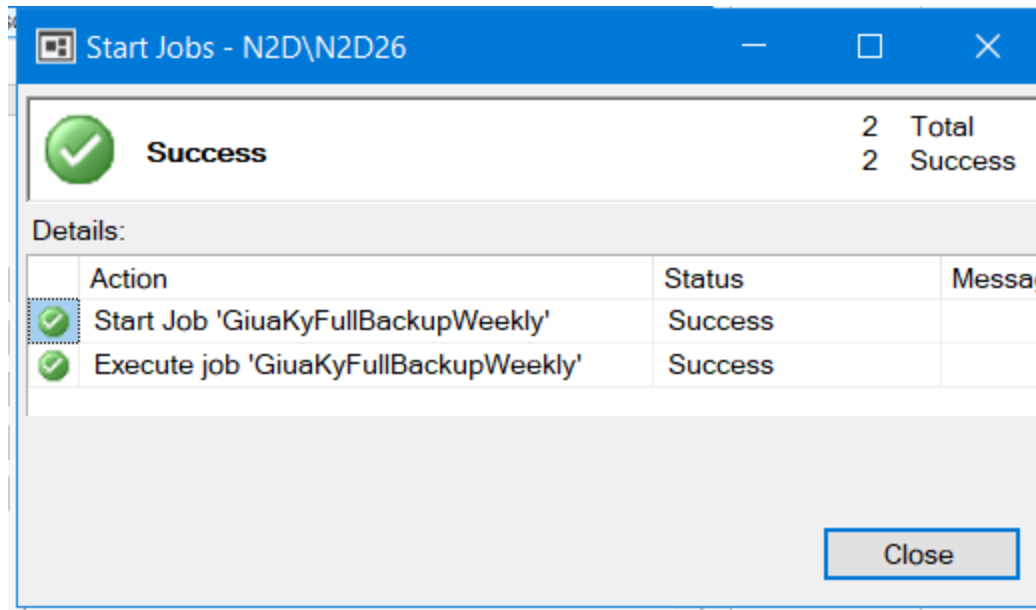
Start date:  ☐ End date:  ☒ No end date:

Summary

Description:

OK Cancel Help

Hình 40: Tạo schedule backup weekly



Hình 41: Kết quả sau khi bắt đầu khởi chạy job

Sao lưu khác biệt (Differential backup) - Hàng ngày

Thực hiện tương tự với job script sau:

*BACKUP DATABASE GiuaKy TO DISK = 'D:\PC\C\Backup\GiuaKy\_Diff.bak'*

*WITH DIFFERENTIAL,*

*NAME = 'GiuaKy-Differential Backup',*

*DESCRIPTION = 'Differential backup của database GiuaKy',*

*COMPRESSION, CHECKSUM, STATS = 10;*

Sao lưu nhật ký giao dịch (Transaction Log backup) - Mỗi giờ

Thực hiện tương tự với job script sau

*BACKUP LOG GiuaKy TO DISK = D:\PC\C\Backup\GiuaKy\_Log.trn'*

*WITH NAME = 'GiuaKy-Log Backup',*

*DESCRIPTION = 'Transaction log backup của database GiuaKy',*

*COMPRESSION, CHECKSUM, STATS = 10;*

Khôi phục dịch vụ nhanh chóng (Point-in-Time Recovery)

Hệ thống hỗ trợ khôi phục đến một thời điểm cụ thể, giúp phục hồi dữ liệu khi có lỗi nghiệp vụ thông qua dữ liệu đã backup thông qua T-SQL bên dưới:

```
RESTORE DATABASE GiuaKy  
FROM DISK = "D:\PC\C\Backup\GiuaKy_Full.bak"  
WITH NORECOVERY;
```

```
RESTORE DATABASE GiuaKy  
FROM DISK = "D:\PC\C\Backup\GiuaKy_Diff.bak"  
WITH NORECOVERY;
```

```
RESTORE LOG GiuaKy  
FROM DISK = "D:\PC\C\Backup\GiuaKy_Log.trn"  
WITH RECOVERY, STOPAT = '2025-03-13T08:30:00';
```