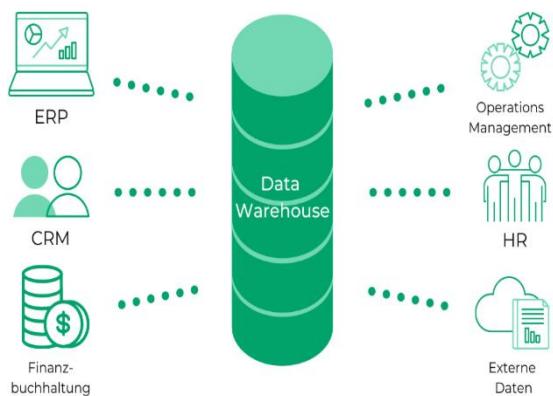




# BÁO CÁO ĐỒ ÁN KHO DỮ LIỆU

## XÂY DỰNG KHO DỮ LIỆU QUẢN LÝ CHO CỬA HÀNG OSLIST



### Nhóm 15:

Lê Tuấn Nghĩa	20133072
Lê Phúc Hậu	20110278
Phan Tân Thành	19110457

GVHD: Nguyễn Văn Thành

Nhóm: DAWH430784\_22\_2\_01

# Mục lục

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI VÀ DATASET SỬ DỤNG.....	3
1.1.    Giới thiệu lí do chọn đề tài:.....	3
1.1.1.    Vấn đề nhận thấy: .....	3
1.1.2.    Giải pháp: .....	3
1.2.    Giới thiệu dataset sử dụng:.....	3
1.2.1.    Nguồn dữ liệu sử dụng:.....	3
1.2.2.    Giới thiệu nguồn sử dụng dữ liệu: .....	3
1.2.3.    Giới thiệu chi tiết dữ liệu: .....	4
1.2.4.    Thông số dataset:.....	4
1.3.    Giới thiệu công cụ sử dụng: .....	5
1.3.1.    Visual Studio 2019:.....	5
1.3.2.    SQL Server 2019:.....	6
1.3.3.    Giới thiệu ngôn ngữ truy vấn SQL: .....	6
CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU (OLAP) CHO CÁC TẬP DỮ LIỆU.....	8
2.1.    Thêm data từ excel: .....	8
2.2.    Thực hiện tiền xử lý : .....	10
2.2.1.    Xử lý và làm sạch dữ liệu: .....	10
2.2.2.    Các bảng Fact con thu được:.....	35
2.2.2.1.    Fact_Order: .....	35
2.2.2.2.    Fact_Order_items:.....	35
2.2.2.3.    Fact_Order_payments:.....	35
2.2.2.4.    Fact_Order_Reviews: .....	36
2.2.3.    Các bảng Dim thu được: .....	36
2.2.3.1.    DimCustomers:.....	36
2.2.3.2.    DimDate: .....	36
2.2.3.3.    DimGeolocation: .....	37
2.2.3.4.    DimSellers: .....	37
2.2.3.5.    DimProducts:.....	37
2.3.    Lược đồ Diagram:.....	38
CHƯƠNG 3: TÍCH HỢP DỮ LIỆU VÀO KHO (SSIS) .....	39
3.1.    Tạo mới project: .....	39
3.2.    Insert dữ liệu vào kho từ excel:.....	39
3.3.    Tạo ADO.NET Connection:.....	40
3.4.    Tạo Data Flow:.....	43

3.4.1.    Data Flow Task:.....	43
3.4.2.    Data Flow Task 1:.....	48
3.5.    Khởi tạo Execute SQL Task: .....	50
3.6.    Khởi tạo các Execute SQL Task create table: .....	51
3.7.    Khởi động project và xem xét kết quả thu được:.....	55
3.7.1.    Data Flow Task:.....	55
3.7.2.    Data Flow Task 1:.....	55
3.8.    Quá trình đổ dữ liệu từ database vào kho dữ liệu:.....	56
3.8.1.    Tạo Connection Management: .....	56
3.8.2.    Khởi tạo các Data Flow Task: .....	57
3.8.2.1.    Data Flow Task:.....	57
3.8.2.2.    Data Flow Task 1:.....	61
3.8.2.3.    Data Flow Task 2:.....	63
3.8.3.    Tạo Execute SQL Task: .....	64
3.8.4.    Khởi tạo các EXECUTE SQL TASK CREATE TABLE: .....	65
3.8.5.    Khởi tạo EXECUTE SQL TASK ADD FOREIGN KEY: .....	69
3.8.6.    Chạy Project và xem xét kết quả thu được: .....	70
3.8.6.1.    Data Flow Task:.....	71
3.8.6.2.    Data Flow Task 1:.....	71
3.8.6.3.    Data Flow Task 2:.....	71
CHƯƠNG 4: PHÂN TÍCH DỮ LIỆU VỚI SSAS .....	72
4.1.    Danh sách các câu truy vấn:.....	72
4.2.    Giới thiệu các phương pháp để sử dụng truy vấn: .....	72
4.2.1.    Sử dụng SSAS, Pivot table: .....	72
4.2.2.    Câu lệnh truy vấn SQL: .....	72
4.3.    Xây dựng mô hình SSAS và Pivot table:.....	72
4.4.    Quá trình xây dựng khối:.....	78
4.5.    Thực hiện các câu truy vấn: .....	84
4.5.1.    Thống kê doanh thu theo khu vực: .....	84
4.5.2.    Thống kê doanh thu theo ngày, tháng: .....	87
4.5.3.    Thống kê số điểm đánh giá thông qua các order: .....	89
4.5.4.    Thống kê sản phẩm bán chạy:.....	90
4.5.5.    Thống kê tổng tiền thông qua các hình thức thanh toán:.....	92
CHƯƠNG 5: KẾT LUẬN .....	94
5.1.    Kết quả đạt được: .....	94

5.2.	Những hạn chế:.....	94
5.3.	Phân công công việc:.....	94
5.4.	Tài liệu tham khảo:.....	95

---

## CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI VÀ DATASET SỬ DỤNG

---

### 1.1. Giới thiệu lí do chọn đề tài:

#### 1.1.1. Vấn đề nhận thấy:

- Hiện nay, theo hướng phát triển công nghệ 4.0, các ngành trong nước và quốc tế có những sự biến đổi nhằm thích nghi với thời đại. Cùng với sự phát triển công nghệ của các nước, đi đôi với đó là những chuyển biến của các ngành kinh tế, trong đó, không thể không nhắc đến nhu cầu đặt hàng trực tuyến của con người.
- Không quá xa lạ đối với chúng ta, cụm từ “đặt hàng trực tuyến” đã trở nên quá quen thuộc. Hiện nay, mọi người có thể chỉ cần 1 click chuột hay 1 cái bấm tay là có thể chọn cho mình những món hàng ưng ý, dù cho là xa hay gần, thì mọi thứ đều sẽ được vận chuyển đến bạn.
- Nhưng với vai trò là 1 người bán hàng thì chúng ta cần biết những gì để có thể thu lại nguồn kinh tế lớn nhất cho bản thân. Vì vậy, nhóm em quyết định thực hiện xây dựng kho dữ liệu phân tích các đơn hàng của một cửa hàng để mọi người có thể hiểu được những bước cơ bản để có thể kinh doanh mang lại lợi nhuận.

#### 1.1.2. Giải pháp:

- Dựa trên nhu cầu thống kê, phân tích và khai thác dữ liệu các nhân sự trong các đơn hàng. Giải pháp là xây dựng kho dữ liệu phục vụ mục đích phân tích, khai thác, và tạo báo cáo tổng. Đưa ra các dự đoán có thể xuất hiện trong năm sau.

### 1.2. Giới thiệu dataset sử dụng:

#### 1.2.1. Nguồn dữ liệu sử dụng:

- Nguồn dữ liệu được thu thập từ nguồn [kaggle.com](https://www.kaggle.com), dataset “**Brazilian E-Commerce Public Dataset by Olist**”
- Link dataset: <https://www.kaggle.com/datasets/olistbr/brazilian-e-commerce>

#### 1.2.2. Giới thiệu nguồn sử dụng dữ liệu:

- “**Kaggle**” được thành lập với hoạt động chủ yếu là một cộng đồng trực tuyến và dành cho những nhà khoa học dữ liệu cùng mọi đối tượng có thể thực hành học máy.

Trong đó có thể hiểu về khoa học dữ liệu chính là một lĩnh vực liên ngành và có sử dụng đến các phương pháp, các quy trình hay thuật toán cùng với hệ thống khoa học công nghệ nhất định nhằm mang đến những kiến thức, những hiểu biết cần thiêу có liên quan đến vấn đề cấu trúc và phi cấu trúc.

### 1.2.3. Giới thiệu chi tiết dữ liệu:

- Đây là bộ dữ liệu công khai về thương mại điện tử của Brazil về các đơn đặt hàng được thực hiện tại Cửa hàng Olist. Bộ dữ liệu có thông tin về 100.000 đơn đặt hàng từ năm 2016 đến năm 2018 được thực hiện tại nhiều thị trường ở Brazil. Các tính năng của nó cho phép xem một đơn đặt hàng từ nhiều chiều: từ trạng thái đơn hàng, giá cả, thanh toán và vận chuyển hàng hóa đến vị trí của khách hàng, thuộc tính sản phẩm và cuối cùng là đánh giá được viết bởi khách hàng. Chúng tôi cũng đã phát hành bộ dữ liệu vị trí địa lý liên quan đến mã zip của Brazil với tọa độ lat/lng.

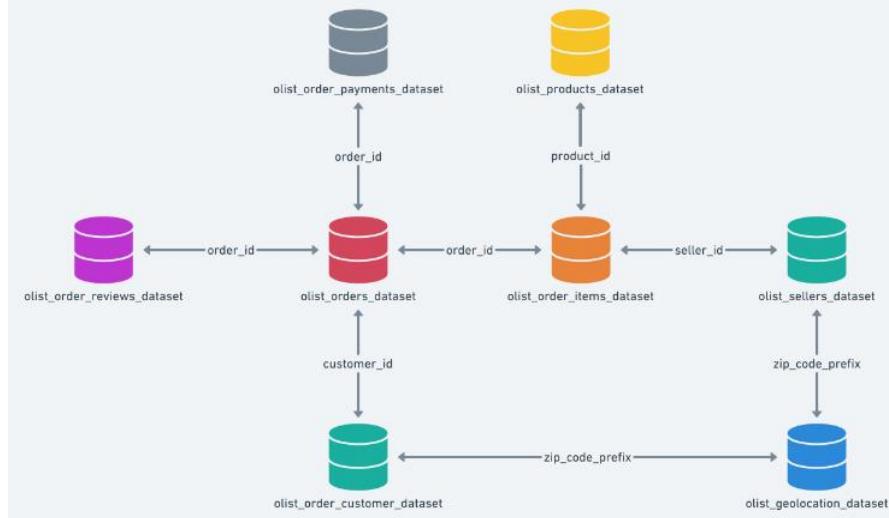
- Sau khi khách hàng mua sản phẩm từ Olist Store, người bán sẽ được thông báo để thực hiện đơn hàng đó. Sau khi khách hàng nhận được sản phẩm hoặc đến hạn giao hàng dự kiến, khách hàng sẽ nhận được bản khảo sát về mức độ hài lòng qua email, nơi họ có thể ghi chú về trải nghiệm mua hàng và viết ra một số nhận xét.

- Chú ý
  - o Một đơn hàng có thể có nhiều mặt hàng.
  - o Mỗi mặt hàng có thể được thực hiện bởi một người bán riêng biệt.

### 1.2.4. Thông số dataset:

- Dữ liệu gồm 9 bảng:
  - o olist\_customers\_dataset: 99441(dòng)\*5(cột), mỗi dòng tương ứng với thông tin của 1 customer.
  - o olist\_geolocation\_dataset: 1000163(dòng)\*5(cột), mỗi dòng là tọa độ của các cửa hàng ở các thành phố.
  - o olist\_order\_items\_dataset: 112650(dòng)\*7(cột), mỗi dòng tương ứng với các item order.
  - o olist\_order\_payments\_dataset: 103886(dòng)\*5(cột), với mỗi dòng tương ứng với hình thức và tổng tiền thanh toán.

- olist\_order\_reviews\_dataset: 99224(dòng)\*7(cột), mỗi dòng tương ứng với các đánh giá của khách hàng.
  - olist\_orders\_dataset: 99441(dòng)\*8(cột), mỗi dòng tương ứng với các order của các customer.
  - olist\_products\_dataset: 32951(dòng)\*9(cột), mỗi dòng là mô tả chi tiết của sản phẩm.
  - olist\_sellers\_dataset: 3095(dòng)\*4(cột), mỗi dòng tương ứng với các seller ở các thành phố.
  - product\_category\_name\_translation: 71(dòng)\*2(cột), mỗi dòng tương ứng với các category khác nhau.
- Data Schema:



### 1.3. Giới thiệu công cụ sử dụng:

- Visual Studio 2019
- SQL Server 2019
- Ngôn ngữ lập trình: SQL

#### 1.3.1. Visual Studio 2019:

- Visual Studio 2019 là một môi trường phát triển tích hợp (IDE) từ Microsoft. Visual Studio 2019 bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cài tiến mã nguồn. Trình gõ lỗi tích hợp hoạt động cả về trình gõ lỗi mức độ mã nguồn và gõ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế

các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu.

- Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI, VB.NET, C# và F#. Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

- Trong Visual Studio 2019 có 3 phiên bản:

- Visual Studio 2019 Community: Có IDE miễn phí nên phù hợp cho học sinh và sinh viên, những cá nhân chưa có điều kiện kinh tế. Tuy nhiên phiên bản này có ít chức năng nên không được dùng phổ biến.
- Visual Studio 2019 Professional: Đây là công cụ chuyên nghiệp có nhiều chức năng. Phần mềm phù hợp cho các nhóm nhỏ.
- Visual Studio 2019 Enterprise: Đây là bản cao cấp và có đầy đủ tính năng cho phép bạn và nhóm làm việc tạo ra các dự án, trò chơi tuyệt vời trên máy tính.

### 1.3.2. SQL Server 2019:

- SQL Server 2019 là bộ phận quản lý cơ sở dữ liệu, được xây dựng dựa trên khái niệm trí tuệ nhân tạo nhằm tạo điều kiện thuận lợi, cải tiến dịch vụ cơ sở dữ liệu, bảo mật và giảm bớt các khó khăn gặp phải khi phát triển các ứng dụng và lưu trữ dữ liệu.

- SQL Server 2019 được tích hợp với Cloud, điều này đồng nghĩa các tổ chức có thể hưởng lợi từ tính năng bảo mật cao, vừa đồng bộ được dữ liệu trên nhiều máy tính và các thiết bị hiện đại khác.

- SQL Server 2019 tạo ra nền tảng dữ liệu hợp nhất đi kèm với Hệ thống tệp phân tán Apache Spark và Hadoop (HDFS) để trở nên thông minh hơn với tất cả dữ liệu.

### 1.3.3. Giới thiệu ngôn ngữ truy vấn SQL:

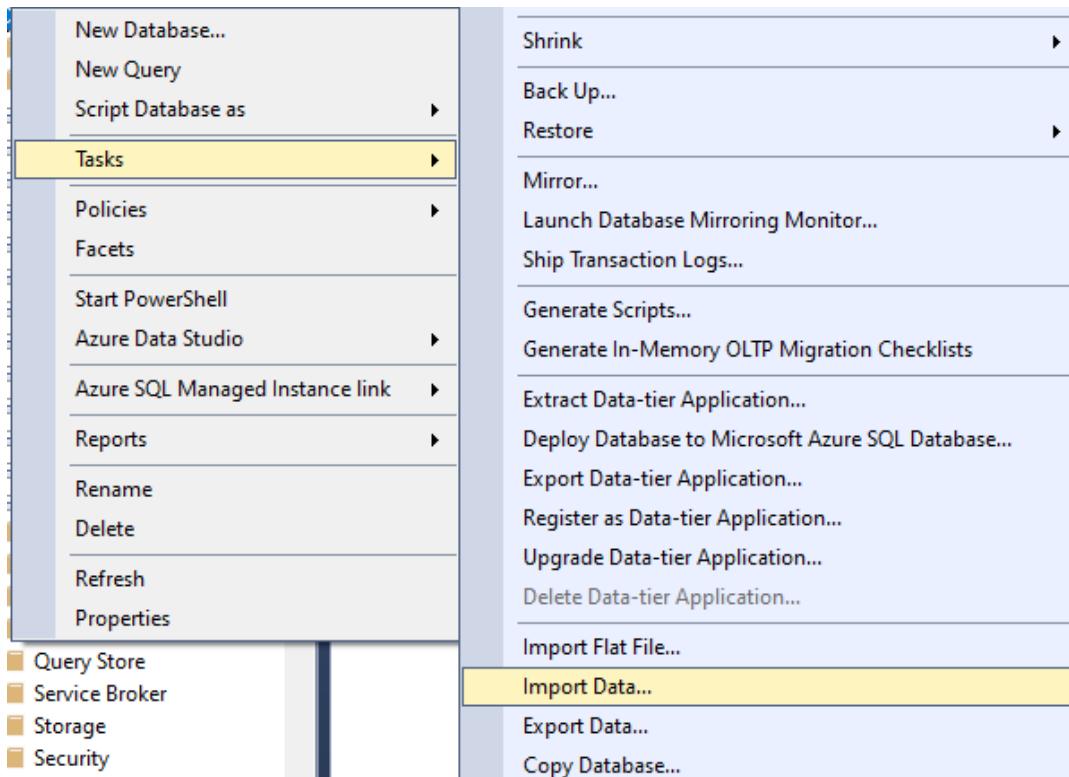
- Ngôn ngữ truy vấn có cấu trúc (SQL) là một ngôn ngữ lập trình phục vụ việc lưu trữ và xử lý thông tin trong cơ sở dữ liệu quan hệ. Cơ sở dữ liệu quan hệ lưu

trữ thông tin dưới dạng bảng có các hàng và cột đại diện cho những thuộc tính dữ liệu và nhiều mối quan hệ khác nhau giữa các giá trị dữ liệu. Bạn có thể sử dụng các câu lệnh SQL để lưu trữ, cập nhật, loại bỏ, tìm kiếm và truy xuất thông tin từ cơ sở dữ liệu. Bạn cũng có thể sử dụng SQL để duy trì và tối ưu hóa hiệu suất cơ sở dữ liệu.

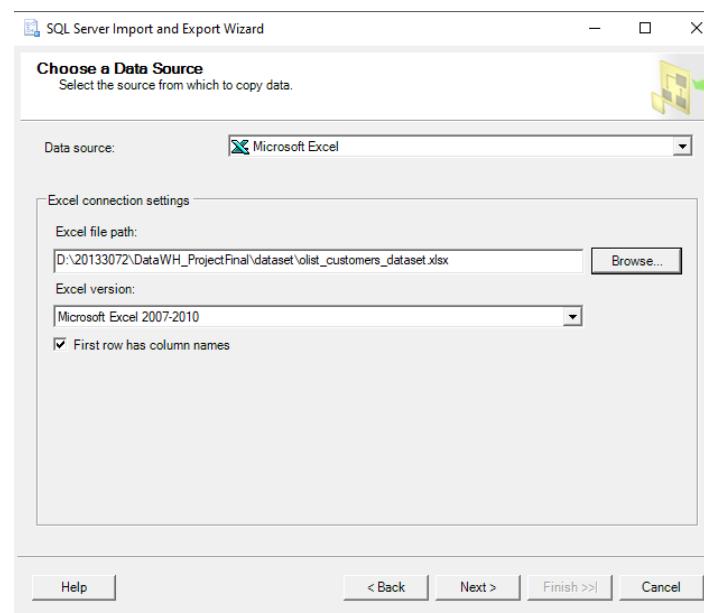
## CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU (OLAP) CHO CÁC TẬP DỮ LIỆU

### 2.1. Thêm data từ excel:

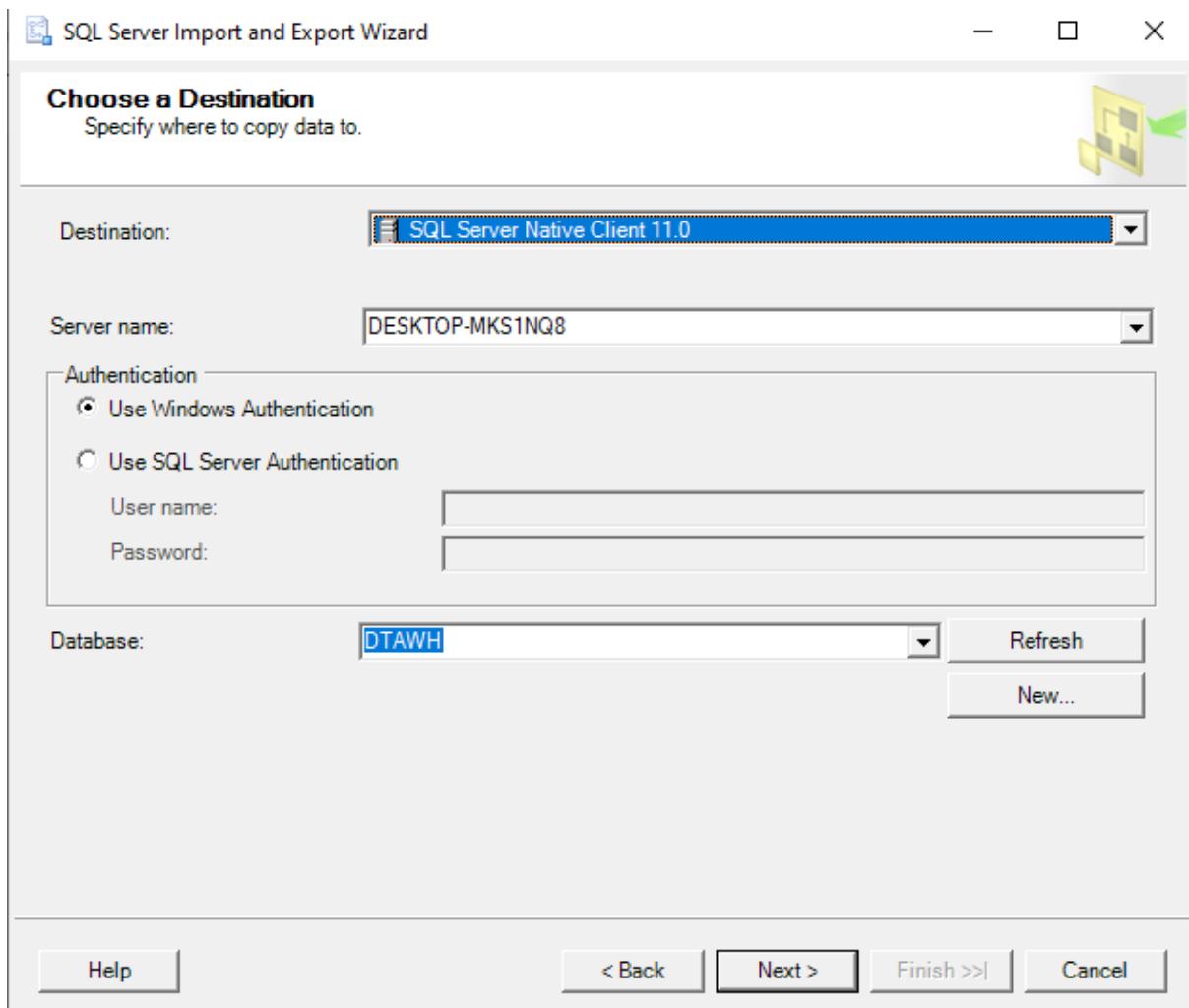
- Bước 1: Chuột phải vào database, chọn Task → Import Data...



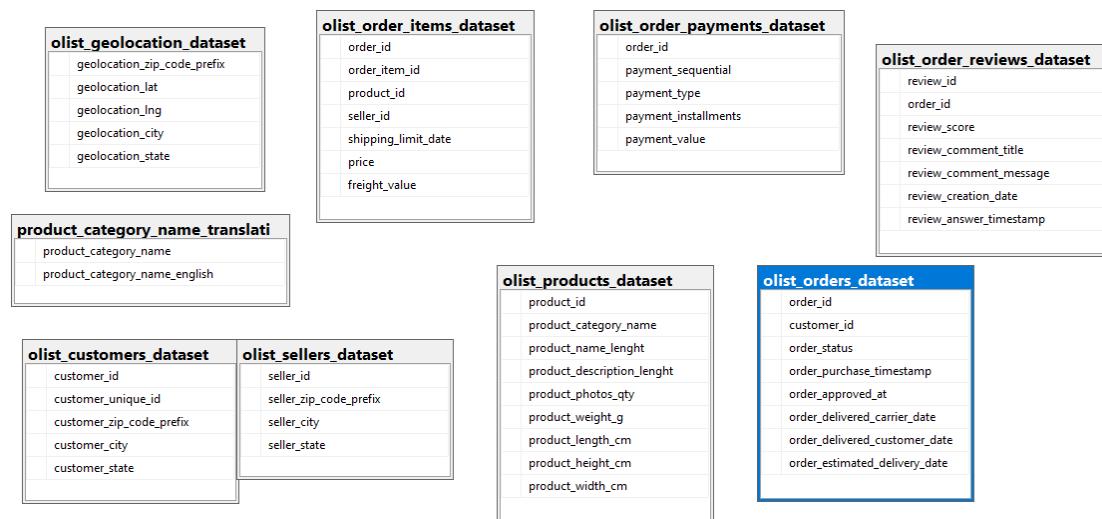
- Bước 2: Tại Data source, chọn Microsoft Excel, và tại Excel file path chọn đường dẫn tới địa chỉ file data..., sau đó chọn Next>



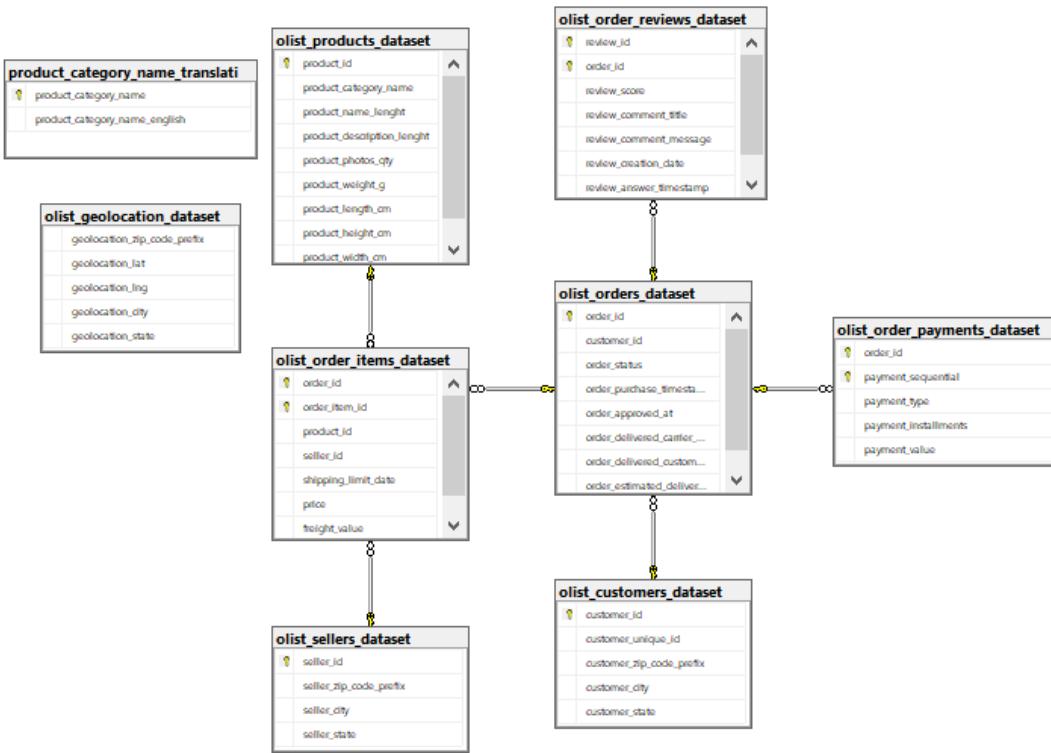
- Bước 3: Tại **Destination**, chọn **SQL Server Native Client 11.0**, và điền các thông tin cần thiết,... Tiếp tục chọn **Next >** cho đến cuối và chọn **Finish** để đổ dữ liệu vào database.



- Sau khi thêm dữ liệu từ các tập excel ban đầu, ta thu được các bảng như sau:



- Thực hiện thêm các ràng buộc khóa chính và khóa ngoại cho các bảng:



## 2.2. Thực hiện tiền xử lý :

### 2.2.1. Xử lý và làm sạch dữ liệu:

```

-- tạo cơ sở dữ liệu BrzShopDW để import các file CSV từ olist dataset
-- sử dụng task -> import flat file để import , chú ý kiểu dữ liệu của các trường
trong file
--create database BrzShopDW
--go
--use BrzShopDW
--go

-- Dùng code này, tạo datawarehouse cho olist dataset

-- Tao data warehouse
create database DA_DATAWH

```

```
go
-- Use datawh
use DA_DATAWH
go

-- DataWareHouse cho shopping datamart

-- Bang DimDate
create table DimDate(
    datekey int IDENTITY not null,
    -- attributes
    [Date] [datetime] NULL,
    [DayOfWeek] [tinyint] NOT NULL,
    [DayName] [varchar](9) NOT NULL,
    [DayOfMonth] [tinyint] NOT NULL,
    [DayOfYear] [smallint] NOT NULL,
    [WeekOfYear] [tinyint] NOT NULL,
    [MonthName] [varchar](9) NOT NULL,
    [MonthOfYear] [tinyint] NOT NULL,
    [Quarter] [tinyint] NOT NULL,
    [QuarterName] [varchar](9) NOT NULL,
    [Year] [smallint] NOT NULL,
    [IsAWeekday] varchar(1) NOT NULL DEFAULT ('N'),
    constraint pkDimDate PRIMARY KEY ([DateKey])
)
go

-- Bang DimLocation
create table DimGeolocation (
    geolocation_key int IDENTITY not null,
```

```
--attributes
geolocation_zip_code_prefix varchar(50),
-- geolocation_lat float,
-- geolocation_lng float,
geolocation_city nvarchar(50),
geolocation_state varchar(50),

constraint pkDimGeolocation primary key (geolocation_key)
)

go

--Bang DimCustomer
create table DimCustomers (
    customer_key int IDENTITY not null,
    geolocation_key int not null,
    --attributes
    customer_id varchar(50),
    customer_unique_id varchar(50),
    --customer_zip_code_prefix varchar,
    --customer_city nvarchar,
    --customer_state varchar
    --metadata
    CONSTRAINT [pkDimCustomer] PRIMARY KEY ( customer_key ),
    constraint fkDimCustomers_geolocation_Key foreign key
    (geolocation_key)
        references DimGeolocation(geolocation_key),
)
go
```

```
-- Bang DimSeller
create table DimSellers (
    seller_key int IDENTITY not null,
    geolocation_key int not null,
    --attributes
    seller_id nvarchar(50),
    --seller_zip_code_prefix nvarchar,
    --seller_city nvarchar,
    --seller_state nvarchar,
    --metadata
    CONSTRAINT [pkDimSellers] PRIMARY KEY ( seller_key ),
    constraint fkDimSellers_geolocation_Key foreign key (geolocation_key)
        references DimGeolocation(geolocation_key),
)
go

--Bang DimProduct
create table DimProducts (
    product_key int IDENTITY not null,
    --attributes
    product_id nvarchar(50),
    product_category_name nvarchar(50),
    product_name_lenght int,
    product_description_lenght int,
    product_photos_qty int,
    product_weight_g int,
    product_length_cm int,
```

```
product_height_cm int,  
product_width_cm int,  
--metadata  
CONSTRAINT [pkDimProducts] PRIMARY KEY ( product_key )  
)  
go  
  
--Bang Fact Order  
create table Fact_Orders (  
    order_key int IDENTITY not null,  
  
    customer_key int not null,  
    order_purchase_timestamp_key int not null,  
    order_approved_at_key int not null,  
    order_delivered_carrier_date_key int not null,  
    order_delivered_customer_date_key int not null,  
    order_estimated_delivery_date_key int not null,  
    --attributes  
    order_id varchar(50),  
    --customer_id varchar,  
    order_status varchar(50),  
    --order_purchase_timestamp datetime,  
    --order_approved_at datetime,  
    --order_delivered_carrier_date datetime,  
    --order_delivered_customer_date datetime,  
    --order_estimated_delivery_date datetime,  
  
    constraint pkOrders primary key (order_key),  
    constraint fkFact_Orders_Customer_Key foreign key (customer_key)  
        references DimCustomers(customer_key),
```

```
constraint fkFact_Orders_order_purchase_timestamp_key foreign key  
(order_purchase_timestamp_key)  
    references DimDate(datekey),  
constraint fkFact_Orders_order_approved_at_key foreign key  
(order_approved_at_key)  
    references DimDate(datekey),  
constraint fkFact_Orders_order_delivered_carrier_date_key foreign key  
(order_delivered_carrier_date_key)  
    references DimDate(datekey),  
constraint fkFact_Orders_order_delivered_customer_date_key foreign key  
(order_delivered_customer_date_key)  
    references DimDate(datekey),  
constraint fkFact_Orders_order_estimated_delivery_date_key foreign key  
(order_estimated_delivery_date_key)  
    references DimDate(datekey)  
)  
go  
  
-- Bang Fact Order Payments  
create table Fact_Order_payments (  
    order_payment_key int IDENTITY not null,  
  
    order_key int not null,  
    --order_id nvarchar primary key,  
    payment_sequential int,  
    payment_type nvarchar(50),  
    payment_installments int,  
    payment_value float  
  
    constraint pkOrder_payments primary key (order_payment_key),
```

```
constraint fkFact_Order_payments_Fact_Orders foreign key (order_key)
    references Fact_Orders(order_key)
)

go

-- Bang Fact Order Review

create table Fact_Order_reviews (
    review_key int IDENTITY not null,
    order_key int not null,
    --attributes
    review_id varchar(50),
    --order_id nvarchar,
    review_score tinyint,
    review_comment_title nvarchar(255),
    review_comment_message nvarchar(255),
    review_creation_date_key int not null,
    review_answer_timestamp_key int not null,
    constraint pkOrder_reviews primary key (review_key),
    constraint fkFact_Order_reviews_order_key foreign key (order_key)
        references Fact_Orders(order_key),
    constraint fkFact_Order_reviews_review_creation_date_key foreign key
    (review_creation_date_key)
        references DimDate(datekey),
    constraint fkFact_Order_reviews_order_review_answer_timestamp_key
foreign key (review_answer_timestamp_key)
        references DimDate(datekey)
)
```

```
go
```

```
-- Bang Fact Order Items
```

```
create table Fact_Order_items (
    order_item_key int IDENTITY not null,
    product_key int not null,
    seller_key int not null,
    shipping_limit_date_key int not null,
    --attributes
    order_id varchar(50),
    order_item_id int,
    --product_id varchar,
    --seller_id varchar,
    --shipping_limit_date datetime,
    price float,
    freight_value float,
    constraint pkOrder_items primary key (order_item_key),
    constraint fkFact_Order_items_product_key foreign key (product_key)
        references DimProducts(product_key),
    constraint fkFact_Order_items_seller_key foreign key (seller_key)
        references DimSellers(seller_key),
    constraint fkFact_Order_items_shipping_limit_date_key foreign key
    (shipping_limit_date_key)
        references DimDate(datekey)
)
```

```
go
```

-- Chèn các giá trị Unknown vào các bảng Dim để không xung đột Khóa Ngoại

-- Unknown Date Value

```
SET IDENTITY_INSERT [DimDate] ON
```

```
go
```

```
INSERT INTO [DimDate]
```

```
    ([DateKey]  
     ,[Date]  
     ,[DayOfWeek]  
     ,[DayName]  
     ,[DayOfMonth]  
     ,[DayOfYear]  
     ,[WeekOfYear]  
     ,[MonthName]  
     ,[MonthOfYear]  
     ,[Quarter]  
     ,[QuarterName]  
     ,[Year]  
     ,[IsAWeekday])
```

```
VALUES
```

```
(-1  
 ,null  
 ,0  
 ,'Unknown'  
 ,0  
 ,0  
 ,0  
 ,'Unknown'  
 ,0
```

```
,0
,'Unknown'

,0
,'?')

GO
SET IDENTITY_INSERT [DimDate] OFF
go
-- Unknown Location
SET IDENTITY_INSERT [DimGeolocation] ON
go

Insert into DimGeolocation
(geolocation_key,
geolocation_zip_code_prefix,
--geolocation_lat,
--geolocation_lng,
geolocation_city,
geolocation_state)
values
(-1
,'Unknown'
-- ,-1
-- ,-1
,'Unknown'
,'Unknown'
)

SET IDENTITY_INSERT [DimGeolocation] OFF
go

-- Unknown Customer
```

```
SET IDENTITY_INSERT [DimCustomers] ON
go
INSERT INTO [DimCustomers]
    (customer_key,
     geolocation_key,
     customer_id,
     customer_unique_id)
VALUES
    (-1
     ,-1
     ,'Unknown Customer'
     ,'Unknown Customer'
     )
GO
SET IDENTITY_INSERT [DimCustomers] OFF
go

-- Unknown Product
SET IDENTITY_INSERT [DimProducts] ON
GO
INSERT INTO [DimProducts]
    (product_key
     ,product_id
     ,product_category_name
     ,product_name_lenght
     ,product_height_cm
     ,product_length_cm
     ,product_weight_g
     ,product_photos_qty
     ,product_description_lenght)
```

## VALUES

```
(-1  
'Unknown'  
'Unknown'  
,-1  
,-1  
,-1  
,-1  
,-1  
,-1  
,-1)
```

GO

SET IDENTITY\_INSERT [DimProducts] OFF

GO

-- Unknown Sellers

SET IDENTITY\_INSERT [DimSellers] ON

GO

insert into DimSellers

```
(seller_key,  
geolocation_key,  
seller_id  
)  
values (  
-1,  
-1,  
'Unknown'  
)
```

SET IDENTITY\_INSERT [DimSellers] OFF

GO

- Đồ Data:

```
- -- Cơ sở dữ liệu StageData chứ dữ liệu Stage
- create database StageData
- go
- use StageData
- go
-
-
-
- create or alter function laChuoiThuong(@chuoi nvarchar(50)) returns int
- begin
-     declare @flag int
-     set @chuoi = REPLACE(@chuoi, ',')
-     while (Unicode(@chuoi) BETWEEN 97 and 122)
-         set @chuoi = SUBSTRING(@chuoi,2,LEN(@chuoi)-1)
-     if LEN(@chuoi) = 0
-         set @flag = 0
-     else
-         set @flag = 1
-     return @flag
- end;
- go
-
-
- select
-     distinct geolocation_zip_code_prefix, geolocation_city,
-     geolocation_state
- into dbo.DA_StageGeolocation
- from BrzShopDW.dbo.olist_geolocation_dataset
- where dbo.laChuoiThuong(geolocation_city) = 0
- order by geolocation_zip_code_prefix
- go
```

```
-  
- -- Stage Date  
- select *  
- into [dbo].[DA.StageDate]  
- from [Temp].[dbo].[Date_Dimension]  
- where year between 2016 and 2020  
- go  
  
-  
- -- Stage Customer  
- select  
-     customer_id,  
-     customer_unique_id,  
-     customer_zip_code_prefix  
- into dbo.DA.StageCustomer  
- from BrzShopDW.dbo.olist_customers_dataset  
- go  
  
-  
- -- Stage Seller  
- select  
-     seller_id,  
-     seller_zip_code_prefix  
- into dbo.DA.StageSeller  
- from BrzShopDW.dbo.olist_sellers_dataset  
- go  
  
-  
- -- Stage Product  
- select  
-     product_id,  
-     product_category_name,  
-
```

```
- product_name_lenght,
- product_description_lenght,
- product_photos_qty,
- product_weight_g,
- product_length_cm,
- product_height_cm,
- product_width_cm

-
- into dbo.DA_StageProduct
- from BrzShopDW.dbo.olist_products_dataset
- go

-
- --select *
- -- from BrzShopDW.dbo.olist_products_dataset,
BrzShopDW.dbo.product_category_name_translation
- -- where BrzShopDW.dbo.olist_products_dataset.product_category_name
= BrzShopDW.dbo.product_category_name_translation

-
- -- Stage order_payment
- select
-     order_id,
-     payment_sequential,
-     payment_type,
-     payment_installments,
-     payment_value
- into dbo.DA_StageOrder_payment
- from BrzShopDW.dbo.olist_order_payments_dataset
- go

-
- -- Stage Order_review
```

```
- select  
-     review_id,  
-     order_id,  
-     review_score,  
-     review_comment_title,  
-     review_comment_message,  
-     review_creation_date,  
-     review_answer_timestamp  
-   into dbo.DA_StageOrder_review  
-   from BrzShopDW.dbo.olist_order_reviews_dataset  
-   go  
-  
-  
-  
- -- Stage Orders  
- select  
-     order_id,  
-     customer_id,  
-     order_status,  
-     order_purchase_timestamp,  
-     order_approved_at,  
-     order_delivered_carrier_date,  
-     order_delivered_customer_date,  
-     order_estimated_delivery_date  
-   into dbo.DA_StageOrders  
-   from BrzShopDW.dbo.olist_orders_dataset  
-   go  
-  
- -- Stage order_items  
- select  
-     order_id,
```

```
- order_item_id,  
- product_id,  
- seller_id,  
- shipping_limit_date,  
- price,  
- freight_value  
- into dbo.DA_StageOrder_items  
- from BrzShopDW.dbo.olist_order_items_dataset  
- go  
  
-  
- -- Load Data to DataWareHouse  
  
-  
- use DA_DATAWH  
- go  
  
-  
- -- Load DimDate  
- SET IDENTITY_INSERT [DimDate] ON  
- go  
  
-  
- insert into DA_DATAWH.dbo.DimDate  
- (datekey, Date, DayOfWeek, DayName, DayOfMonth, DayOfYear,  
- WeekOfYear, MonthName, MonthOfYear, Quarter, QuarterName, Year,  
IsAWeekday)  
- select  
- date_key, full_date, day_of_week, day_name, day_num_in_month,  
- day_num_overall, week_num_in_year, month_name, month, quarter,  
- case  
- when month >= 1 and month <= 3 then 'First'  
- when month >= 4 and month <= 6 then 'Second'  
- when month >= 7 and month <= 9 then 'Third'
```

```
-           when month >= 10 and month <= 12 then 'Fourth'  
-           end,  
-           year, weekday_flag  
-           from StageData.dbo.DA_StageDate  
-           go  
-  
-           SET IDENTITY_INSERT [DimDate] OFF  
-           go  
-  
-           -- Load Location  
-           --SET IDENTITY_INSERT [DimGeolocation] ON  
-           --go  
-  
-           insert into DA_DATAWH.dbo.DimGeolocation  
-           (geolocation_zip_code_prefix, geolocation_city, geolocation_state)  
-           select  
-           geolocation_zip_code_prefix, geolocation_city, geolocation_state  
-           from StageData.dbo.DA_StageGeolocation  
-           go  
-  
-           --SET IDENTITY_INSERT [DimGeolocation] OFF  
-           --go  
-           -- Load DimProducts  
-           --SET IDENTITY_INSERT [DimProducts] ON  
-           --go  
-  
-           insert into DA_DATAWH.dbo.DimProducts  
-           (product_id, product_category_name, product_name_lenght,  
product_description_lenght, product_photos_qty,
```

```
- product_weight_g,  
product_length_cm,product_height_cm,product_width_cm)  
- select  
- product_id, product_category_name,  
product_name_lenght,product_description_lenght,product_photos_qty,  
product_weight_g,  
- product_length_cm,product_height_cm,product_width_cm  
- from StageData.dbo.DA_StageProduct  
  
-  
- --SET IDENTITY_INSERT [DimProducts] OFF  
- --go  
  
-  
- -- Load Dim Seller  
- --SET IDENTITY_INSERT [DimSellers] ON  
- --go  
  
-  
- --DBCC CHECKIDENT ('DimSellers',RESEED,0)  
  
-  
- insert into DA_DATAWH.dbo.DimSellers  
- (geolocation_key,seller_id)  
- select geolocation_key, seller_id  
- from StageData.dbo.DA_StageSeller a join  
DA_DATAWH.dbo.DimGeolocation b  
- on a.seller_zip_code_prefix = b.geolocation_zip_code_prefix  
  
-  
- --SET IDENTITY_INSERT [DimSellers] OFF  
- --go  
  
-  
- --delete from DimSellers  
-
```

```
- -- Load Customers  
  
-  
  
- --SET IDENTITY_INSERT [DimCustomers] ON  
  
- --go  
  
-  
  
- -- các câu lệnh để test ko chạy  
  
- --delete from DimCustomers  
  
- --DBCC CHECKIDENT ('DimCustomers',RESEED,0)  
  
-  
  
- insert into DA_DATAWH.dbo.DimCustomers  
    (geolocation_key, customer_id, customer_unique_id)  
- select geolocation_key, customer_id, customer_unique_id  
-         from StageData.dbo.DA_StageCustomer a join  
    DA_DATAWH.dbo.DimGeolocation b  
        on a.customer_zip_code_prefix = b.geolocation_zip_code_prefix  
- go  
- --SET IDENTITY_INSERT [DimCustomers] OFF  
- --go  
  
-  
  
-  
  
-  
  
- -- Load Facts Orders  
  
- -- xem dữ liệu  
  
- select  
    MIN(a.order_purchase_timestamp) as  
    order_purchase_timestamp_mindate,  
    MAX(a.order_purchase_timestamp) as  
    order_purchase_timestamp_maxdate,  
    MIN(a.order_approved_at) as order_approved_at_mindate,  
    MAX(a.order_approved_at) as order_approved_at_maxdate,
```

```
- MIN(a.order_delivered_carrier_date) as  
    order_delivered_carrier_date_mindate,  
- MAX(a.order_delivered_carrier_date) as  
    order_delivered_carrier_date_maxdate,  
- MIN(a.order_delivered_customer_date) as  
    order_delivered_customer_date_mindate,  
- MAX(a.order_delivered_customer_date) as  
    order_delivered_customer_date_maxdate,  
- MIN(a.order_estimated_delivery_date) as  
    order_estimated_delivery_date_mindate,  
- MAX(a.order_estimated_delivery_date) as  
    order_estimated_delivery_date_maxdate  
- from BrzShopDW.dbo.olist_orders_dataset a  
- go  
  
-  
- insert into DA_DATAWH.dbo.Fact_Orders  
- (customer_key, order_purchase_timestamp_key,  
order_approved_at_key, order_delivered_carrier_date_key,  
order_delivered_customer_date_key,  
order_estimated_delivery_date_key, order_id, order_status)  
- select a.customer_key,  
- case when s.order_purchase_timestamp is null then -1  
- else Day(s.order_purchase_timestamp) +  
MONTH(s.order_purchase_timestamp) * 100 +  
YEAR(s.order_purchase_timestamp) * 10000  
- end As order_purchase_timestamp_key,  
- case when s.order_approved_at is null then -1  
- else Day(s.order_approved_at) + MONTH(s.order_approved_at) * 100  
+ YEAR(s.order_approved_at) * 10000  
- end As order_approved_at_key,
```

```
- case when s.order_delivered_carrier_date is null then -1
- else Day(s.order_delivered_carrier_date) +
MONTH(s.order_delivered_carrier_date) * 100 +
YEAR(s.order_delivered_carrier_date) * 10000
- end As order_delivered_carrier_date_key,
- case when s.order_delivered_customer_date is null then -1
- else Day(s.order_delivered_customer_date) +
MONTH(s.order_delivered_customer_date) * 100 +
YEAR(s.order_delivered_customer_date) * 10000
- end As order_delivered_customer_date_key,
- case when s.order_estimated_delivery_date is null then -1
- else Day(s.order_estimated_delivery_date) +
MONTH(s.order_estimated_delivery_date) * 100 +
YEAR(s.order_estimated_delivery_date) * 10000
- end As order_estimated_delivery_date_key,
- s.order_id,
- s.order_status
- from StageData.dbo.DA_StageOrders s
- join DA_DATAWH.dbo.DimCustomers a
- on s.customer_id = a.customer_id
- go
-
- -- Load Facts Orders Items
- select
-     min(a.shipping_limit_date) as mindate,
-     max(a.shipping_limit_date) as maxdate
- from BrzShopDW.dbo.olist_order_items_dataset a
- go
-
- insert into DA_DATAWH.dbo.Fact_Order_items
```

```
- (product_key, seller_key, shipping_limit_date_key, order_id,
order_item_id, price, freight_value)
- select
- a.product_key,
- b.seller_key,
- case when s.shipping_limit_date is null then -1
- else Day(s.shipping_limit_date) + MONTH(s.shipping_limit_date) *
100 + YEAR(s.shipping_limit_date) * 10000
- end As shipping_limit_date_key,
- s.order_id,
- s.order_item_id,
- s.price,
- s.freight_value
- from StageData.dbo.DA_StageOrder_items s
- join DA_DATAWH.dbo.DimProducts a
- on s.product_id = a.product_id
- join DA_DATAWH.dbo.DimSellers b
- on s.seller_id = b.seller_id
- go
-
-
- -- Load Facts Orders Review
- select
- min(a.review_creation_date) as review_creation_date_min,
- max(a.review_creation_date) as review_creation_date_max,
- min(a.review_answer_timestamp) as
review_answer_timestamp_min ,
- max(a.review_answer_timestamp) as
review_answer_timestamp_max
- from BrzShopDW.dbo.olist_order_reviews_dataset a
- go
```

```
-  
- insert into DA_DATAWH.dbo.Fact_Order_reviews  
- (order_key,review_id,review_score,review_comment_title,review_com  
ment_message,review_creation_date_key,review_answer_timestamp_key)  
- select  
- a.order_key,  
- s.review_id,  
- s.review_score,  
- s.review_comment_title,  
- s.review_comment_message,  
- case when s.review_creation_date is null then -1  
- else Day(s.review_creation_date) + MONTH(s.review_creation_date) *  
100 + YEAR(s.review_creation_date) * 10000  
- end As review_creation_date_key,  
- case when s.review_answer_timestamp is null then -1  
- else Day(s.review_answer_timestamp) +  
MONTH(s.review_answer_timestamp) * 100 +  
YEAR(s.review_answer_timestamp) * 10000  
- end As review_answer_timestamp_key  
- from StageData.dbo.DA_StageOrder_review s  
- join DA_DATAWH.dbo.Fact_Orders a  
- on s.order_id = a.order_id  
-  
-  
- go  
-  
- -- Load Facts Orders Payments  
- insert into DA_DATAWH.dbo.Fact_Order_payments  
- (order_key, payment_sequential, payment_type, payment_installments,  
payment_value)
```

```
- select  
-     a.order_key,  
-     s.payment_sequential,  
-     s.payment_type,  
-     s.payment_installments,  
-     s.payment_value  
- from StageData.dbo.DA_StageOrder_payment s  
- join DA_DATAWH.dbo.Fact_Orders a  
- on s.order_id = a.order_id  
- go  
-  
- --  
- --  
- -- Test Data  
-  
- -- các code sql để test dữ liệu thì code ở đây  
- --select distinct geolocation_zip_code_prefix , geolocation_city,  
    geolocation_state from BrzShopDW.dbo.olist_geolocation_dataset  
- --select * from BrzShopDW.dbo.olist_geolocation_dataset  
- --select * from BrzShopDW.dbo.olist_geolocation_dataset  
- --go  
-  
- --use StageData  
- --go  
-  
- --select distinct customer_zip_code_prefix, customer_city  
- --from BrzShopDW.dbo.olist_customers_dataset  
- --order by customer_zip_code_prefix  
- --go  
- --select distinct seller_zip_code_prefix, seller_city
```

- --from BrzShopDW.dbo.olist\_sellers\_dataset
- --go

## 2.2.2. Các bảng Fact con thu được:

### 2.2.2.1. Fact\_Order:

Tên thuộc tính	Ý Nghĩa
order_key	Key Order
customer_key	Tình trạng của đơn hàng
order_purchase_timestamp_key	Mã thời gian đặt hàng
order_approved_at_key	Mã thời gian đơn hàng được xác nhận
order_delivered_carrier_date_key	Mã thời gian đơn vị vận chuyển nhận hàng
order_delivered_customer_date_key	Mã thời gian khách hàng nhận hàng
order_estimated_delivery_date_key	Mã thời gian ước lượng nhận hàng
order_id	Mã đơn hàng
order_status	Tình trạng đơn hàng

### 2.2.2.2. Fact\_Order\_items:

Tên thuộc tính	Ý Nghĩa
order_item_key	Key Order_item
product_key	Key Product
seller_key	Key Seller
shipping_limit_date_key	Key ngày giao hàng
order_key	Key Order
order_id	Mã Order
order_item_id	Mã Order_item
price	Giá sản phẩm
freight_value	Phí ship

### 2.2.2.3. Fact\_Order\_payments:

Tên thuộc tính	Ý Nghĩa

order_payment_key	Key Payment
order_key	Key Order
payment_sequential	Trình tự thanh toán
payment_type	Kiểu thanh toán
payment_installments	Trả góp
payment_value	Trị giá của các kiểu thanh toán

#### 2.2.2.4. Fact\_Order\_Reviews:

Tên thuộc tính	Ý Nghĩa
review_key	Key Review
order_key	Key Order
review_id	Mã Review
review_score	Điểm đánh giá của Review
review_comment_title	Title của Review
review_comment_message	Bình luận
review_creation_date_key	Ngày review được tạo
review_answer_timestamp_key	Ngày review được đánh giá

#### 2.2.3. Các bảng Dim thu được:

##### 2.2.3.1. DimCustomers:

Tên thuộc tính	Ý Nghĩa
customer_key	Key Customer
geolocation_key	Key Geolocation
customer_id	Mã khách hàng
customer_unique_id	Định danh của khách hàng

##### 2.2.3.2. DimDate:

Tên thuộc tính	Ý Nghĩa
datekey	Key Date
Date	Ngày
DayOfWeek	Ngày thứ mấy trong tuần

DayName	Tên ngày
DayOfMonth	Ngày trong tháng
DayOfYear	Ngày trong năm
WeekOfYear	Tuần trong năm
MonthName	Tên Tháng
MonthOfYear	Tháng mấy trong năm
Quarter	Quý thứ
QuarterName	Tên Quý
Year	Năm
IsAWeekday	Ngày cuối tuần ?

### 2.2.3.3. DimGeolocation:

Tên thuộc tính	Ý Nghĩa
geolocation_key	Key Geolocation
geolocation_zip_code_prefix	Mã vùng
geolocation_city	Thành phố
geolocation_state	Tên vùng

### 2.2.3.4. DimSellers:

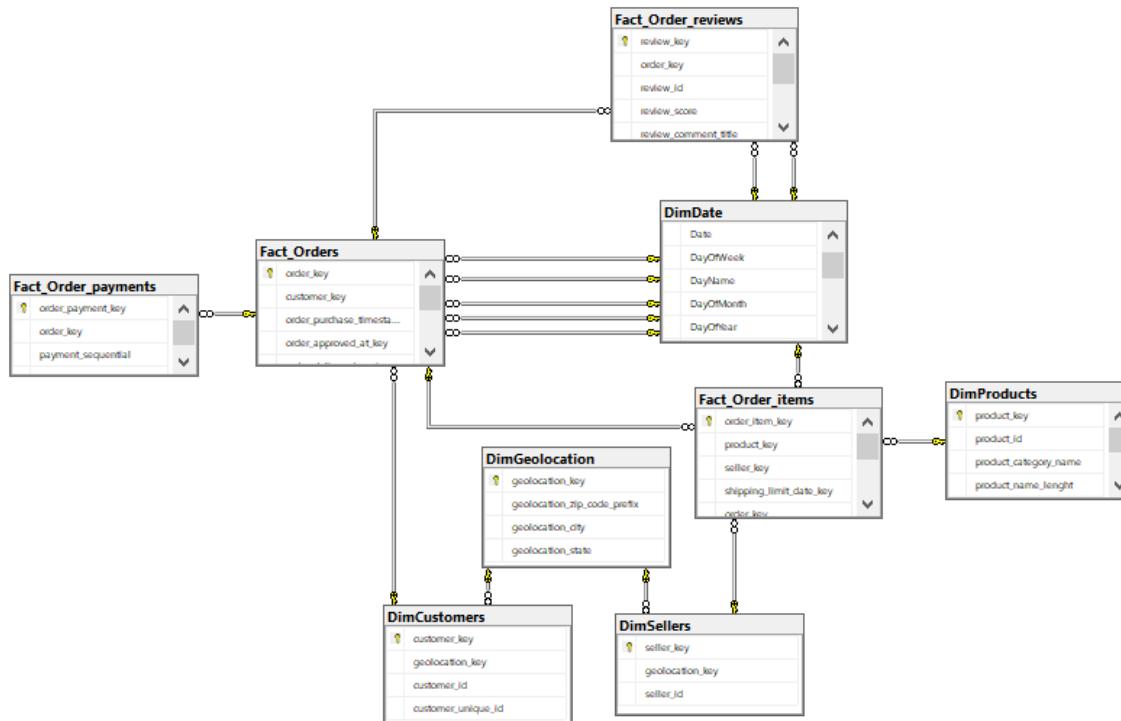
Tên thuộc tính	Ý Nghĩa
seller_key	Key Seller
geolocation_key	Key Geolocation
seller_id	Mã Seller

### 2.2.3.5. DimProducts:

Tên thuộc tính	Ý Nghĩa
product_key	Key Product
product_id	Mã sản phẩm
product_category_name	Loại sản phẩm
product_name_lenght	Chiều dài tên sản phẩm
product_description_lenght	Chiều dài mô tả sản phẩm

product_photos_qty	Số ảnh của sản phẩm
product_weight_g	Cân nặng sản phẩm
product_length_cm	Chiều dài sản phẩm
product_height_cm	Chiều cao sản phẩm
product_width_cm	Chiều rộng sản phẩm

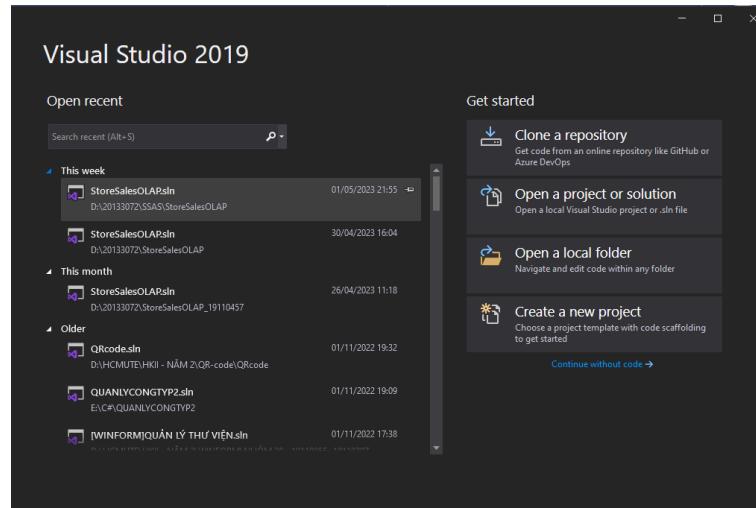
### 2.3. Lược đồ Diagram:



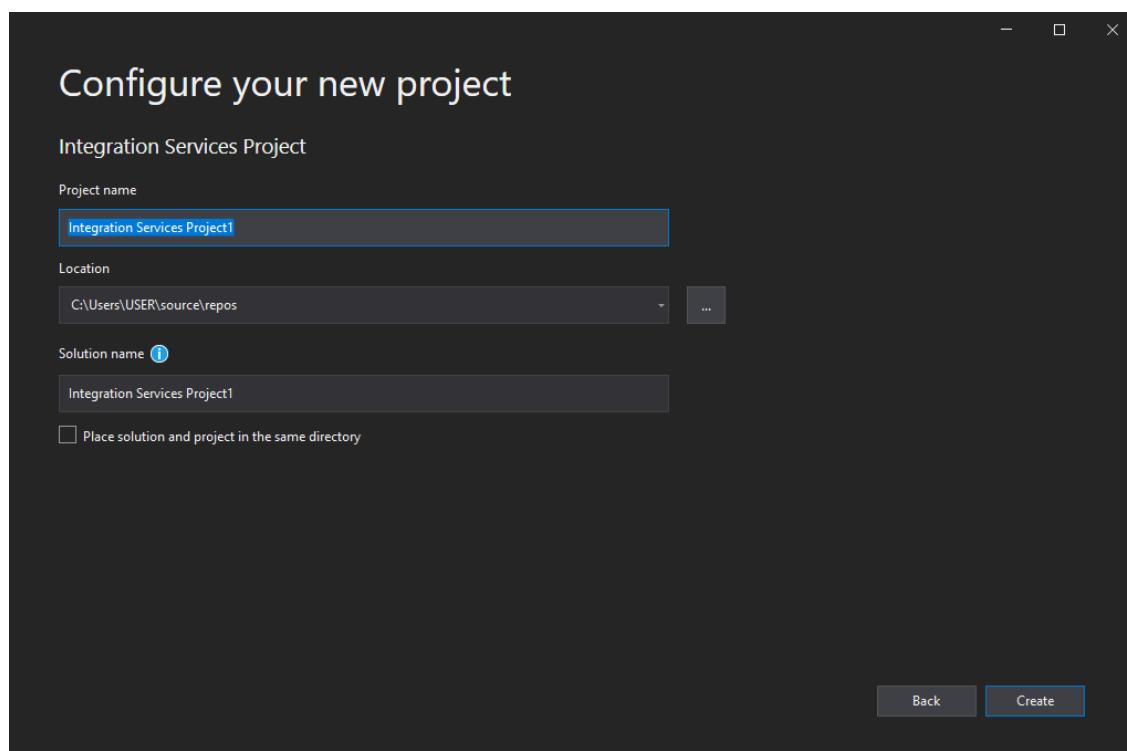
## CHƯƠNG 3: TÍCH HỢP DỮ LIỆU VÀO KHO (SSIS)

### 3.1. Tạo mới project:

- Tại giao diện ban đầu, chọn **Create New Project → Integration Services Project → Next**

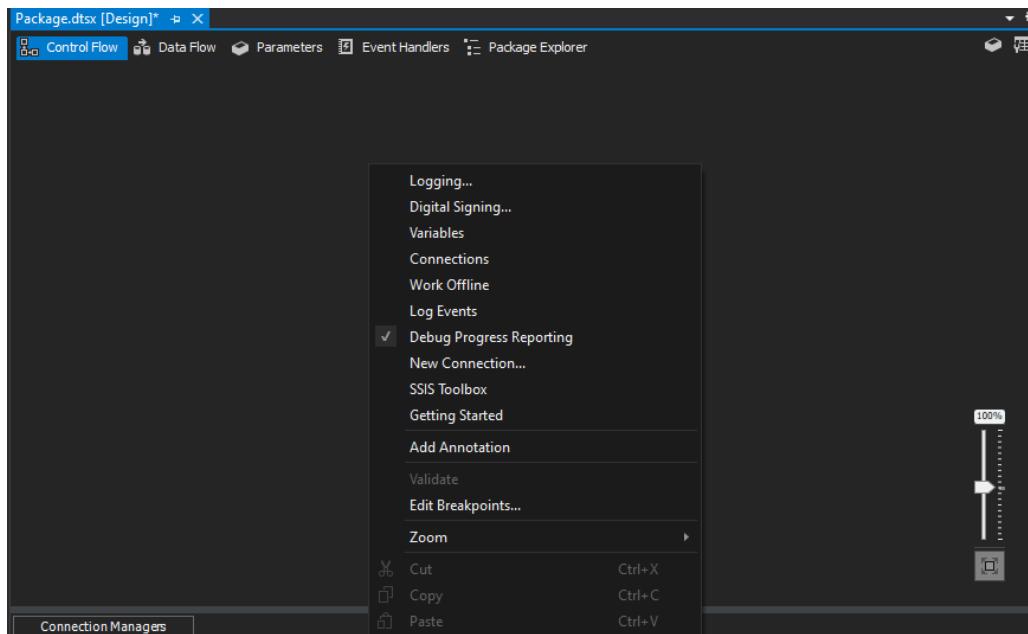


- Nhập **Project Name** và **Solution name** → **Create**

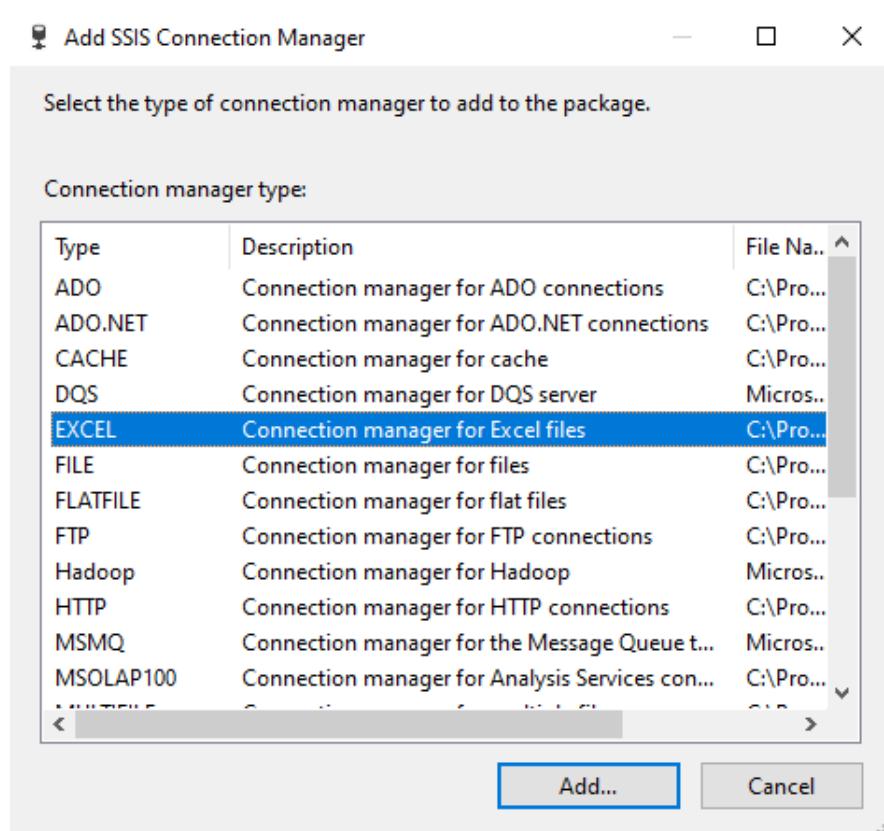


### 3.2. Insert dữ liệu vào kho từ excel:

- Chuột phải chọn **New Connection...**

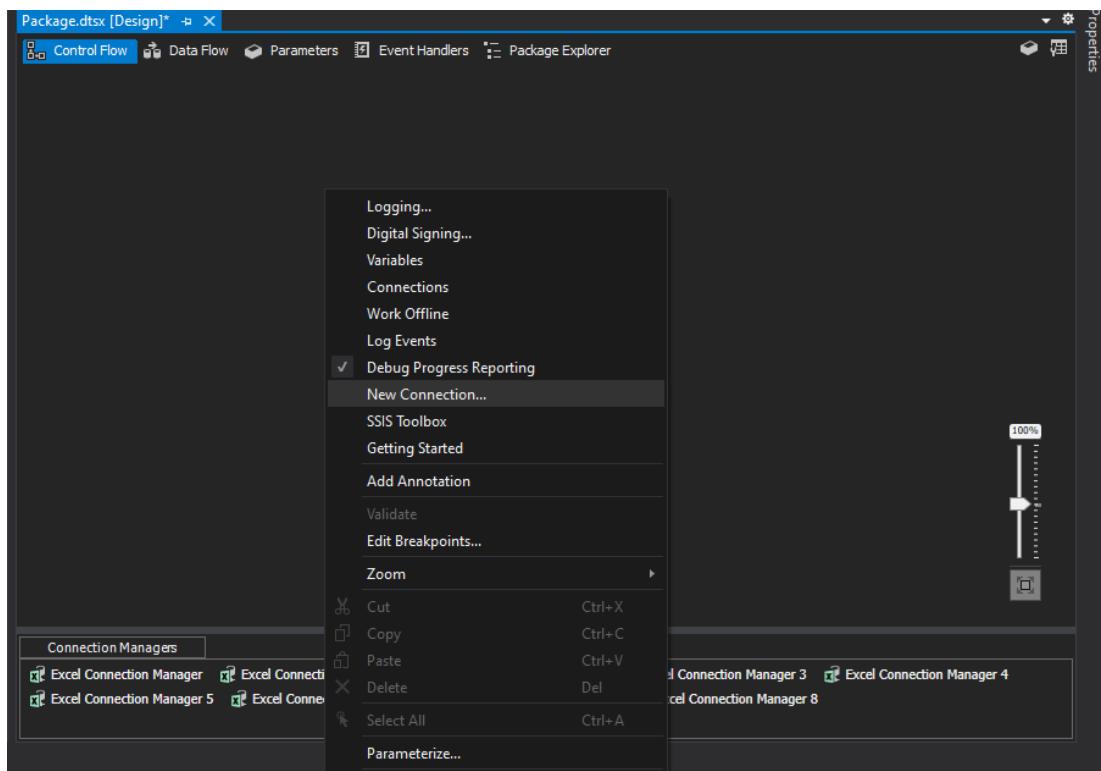


- Chọn **EXCEL** → **Add...** → điền đường dẫn đến file → **OK**

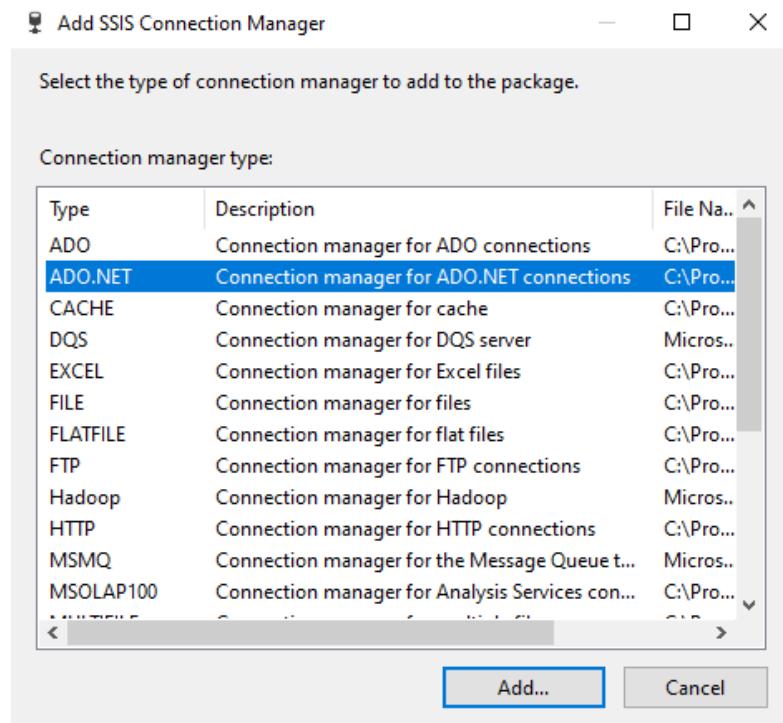


### 3.3. Tạo ADO.NET Connection:

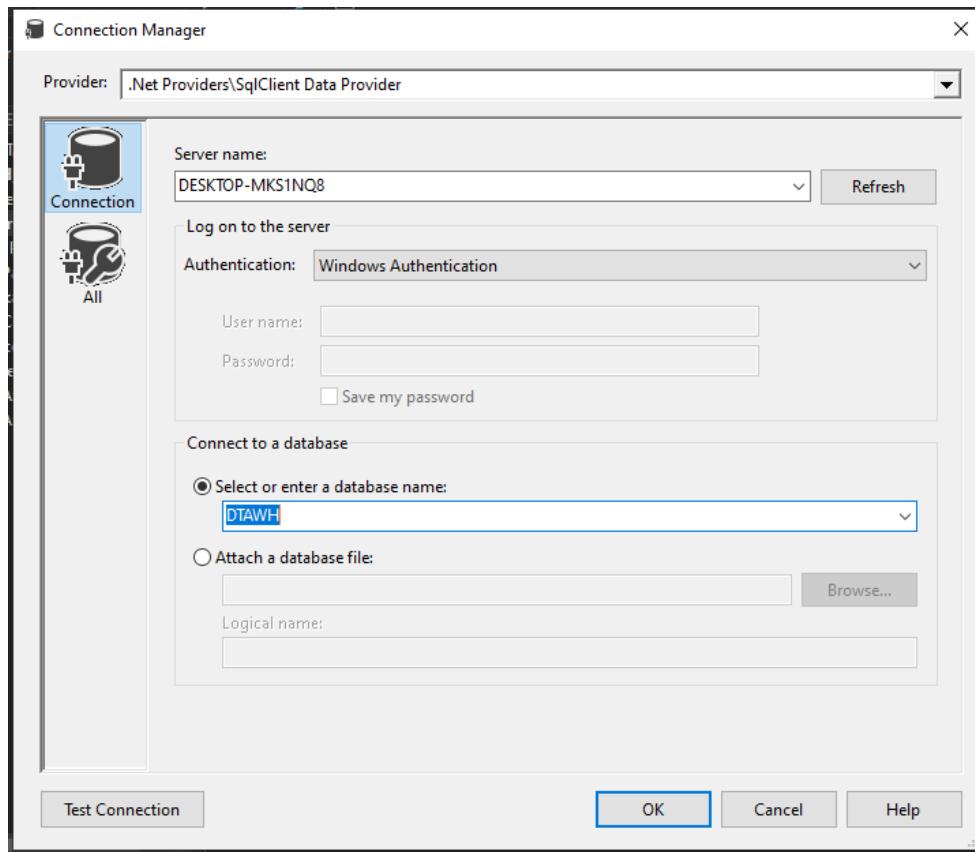
- Chuột phải chọn **New Connection...**



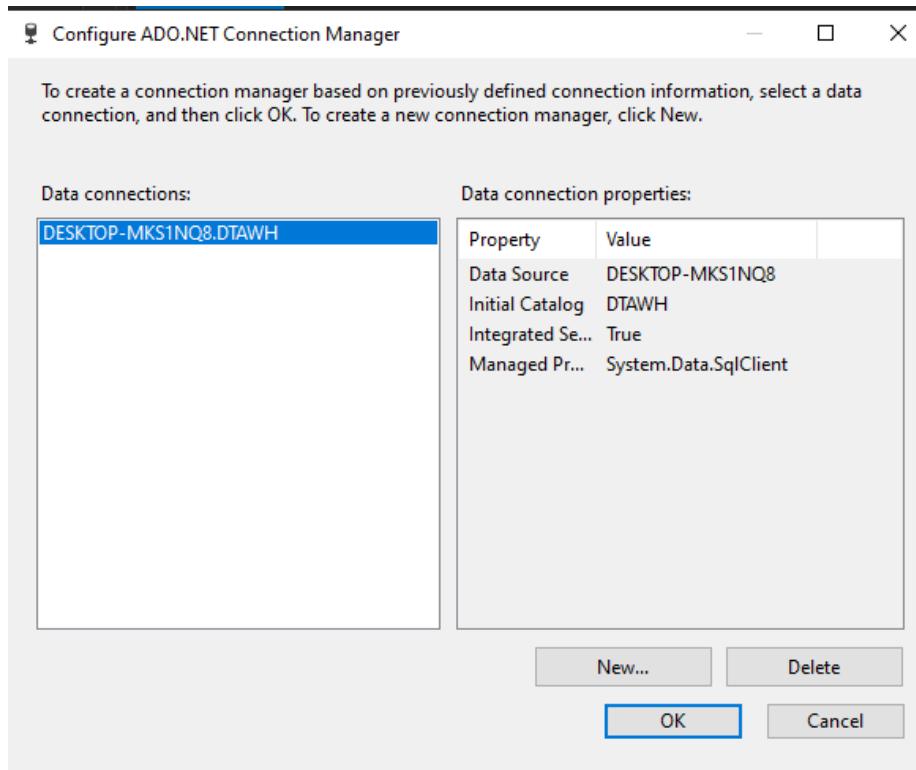
- Chọn ADO.NET



- Điền Sever name và database name thích hợp → **OK**.



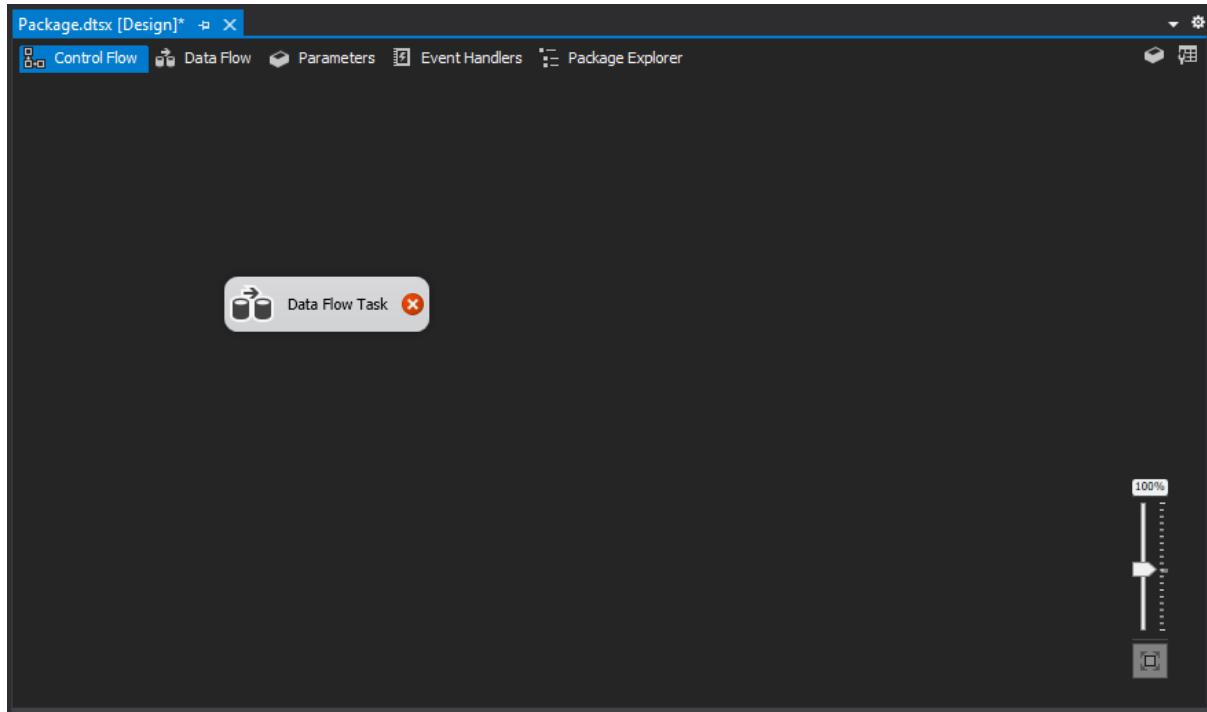
- Tiếp tục ấn **OK**.



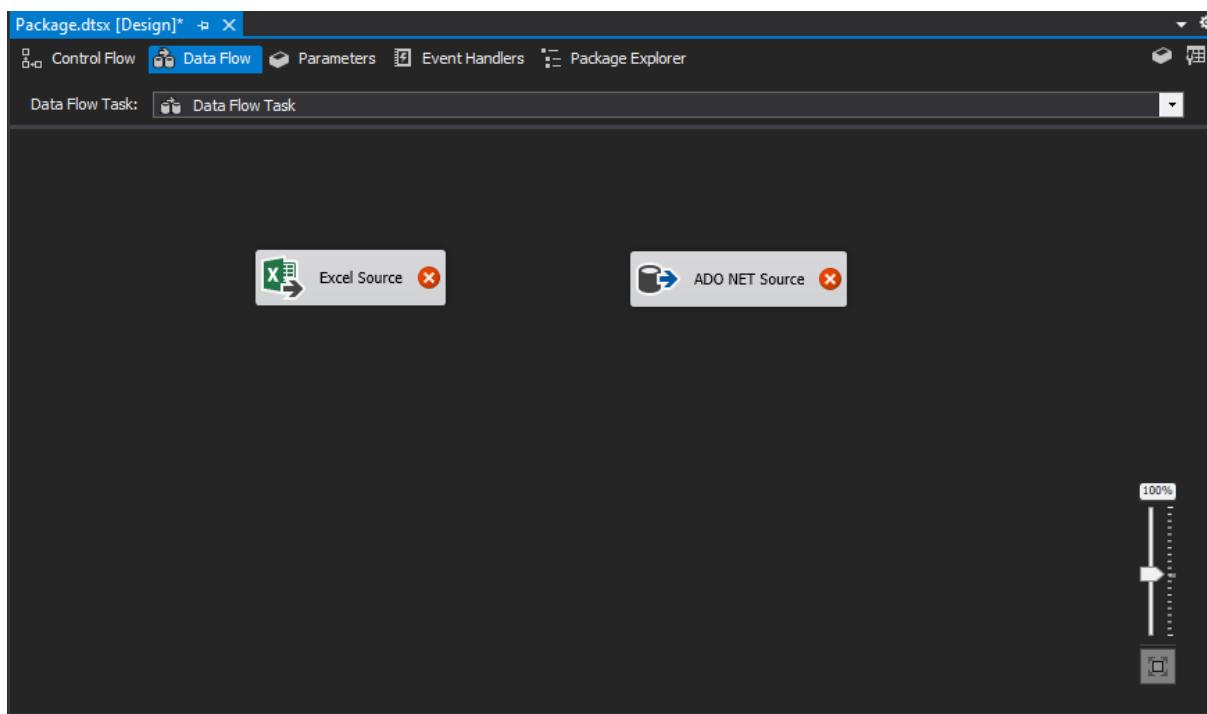
### 3.4. Tạo Data Flow:

#### 3.4.1. Data Flow Task:

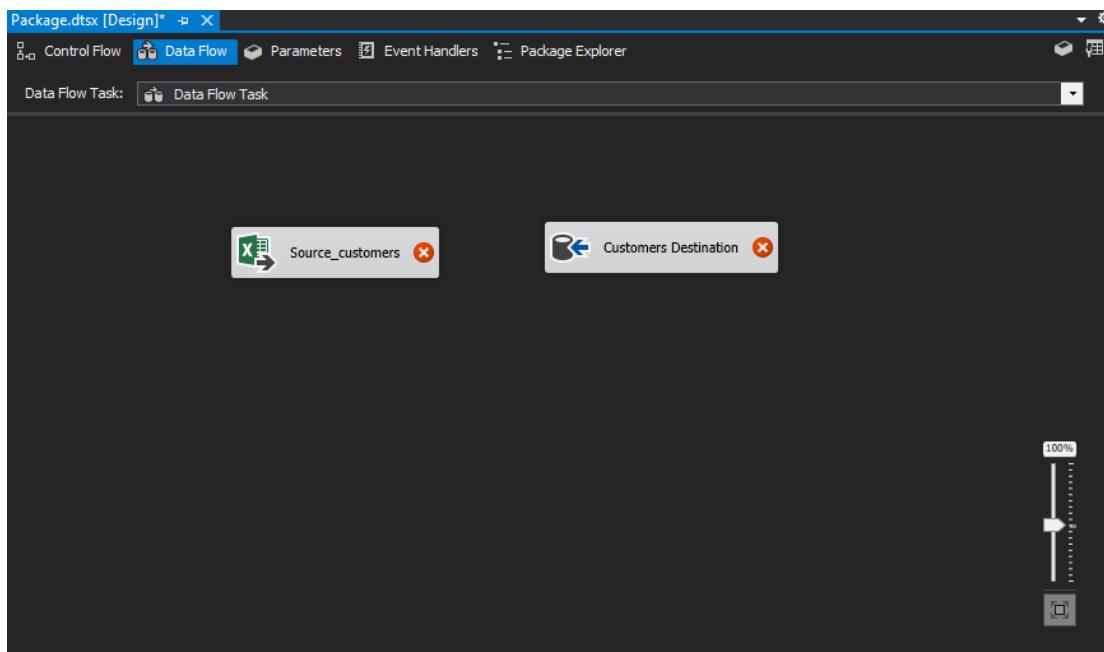
- Từ thanh ToolBox, kéo thả 1 **Data Flow Task**



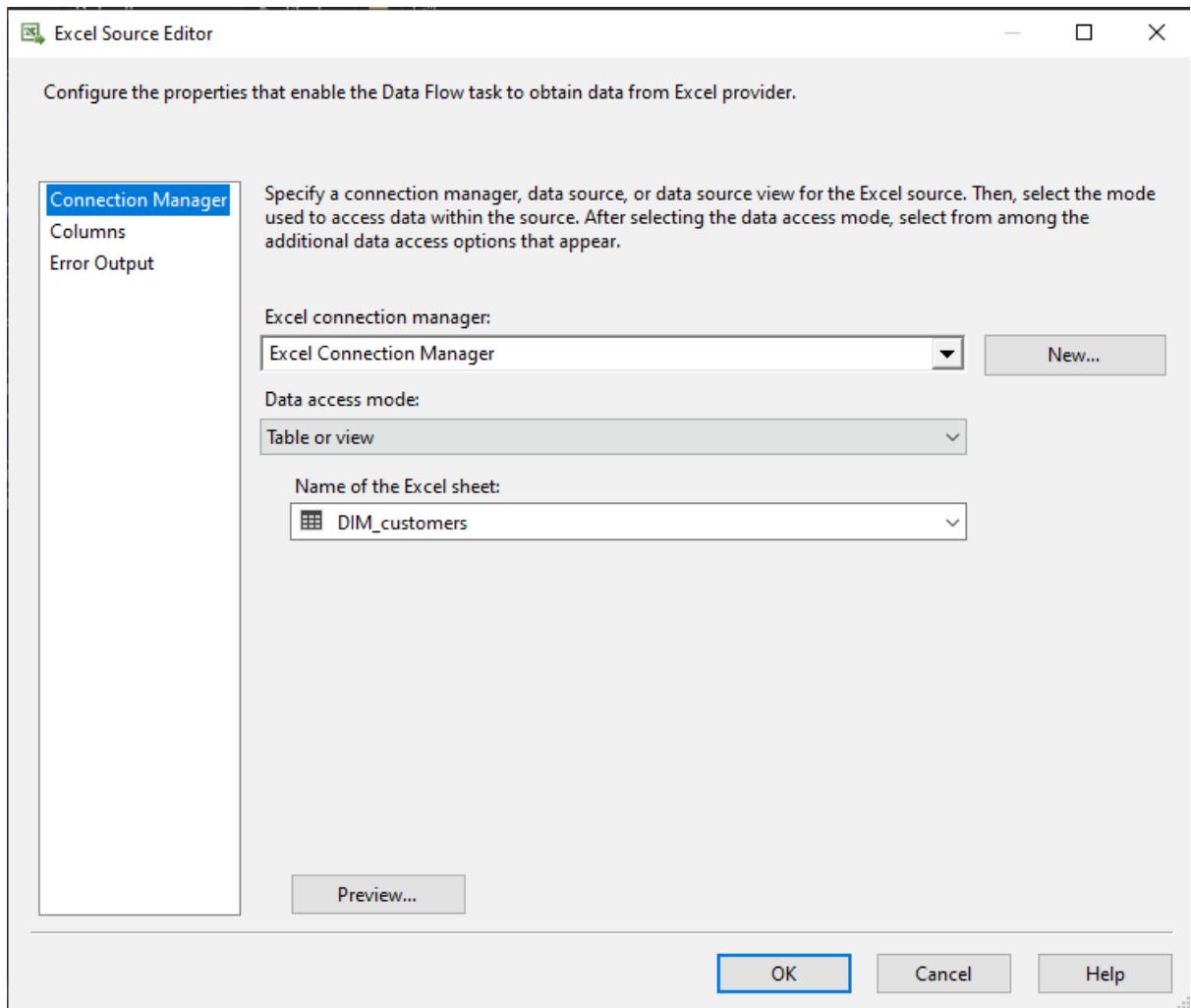
- Định nghĩa **Data Flow Task** bằng cách nháy chuột vào và tiếp tục kéo thả **Excel Source** và **ADO NET Destination** như hình:



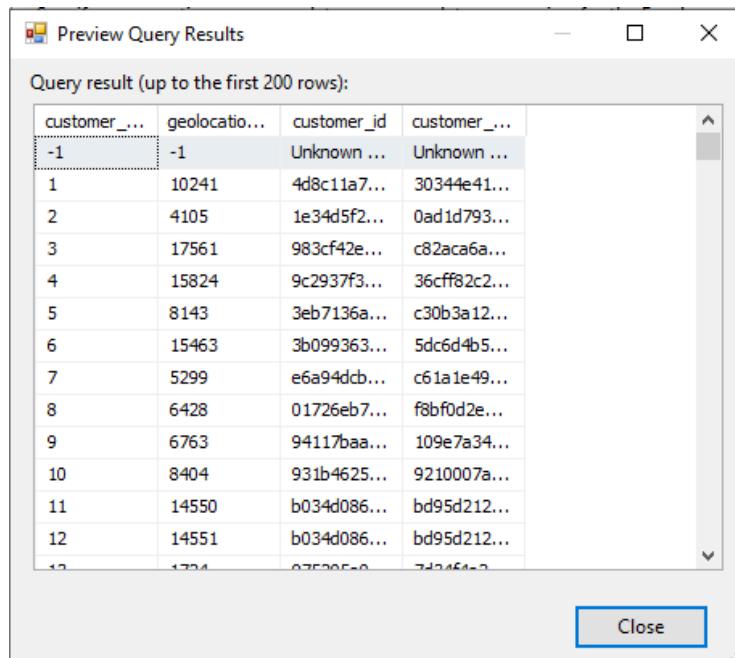
- Đặt tên 2 control:



- Nhấn đúp chuột vào **Source customers** -> Chọn **Excel Connection Management** đã tạo -> Chọn **Excel Sheet Customers**:



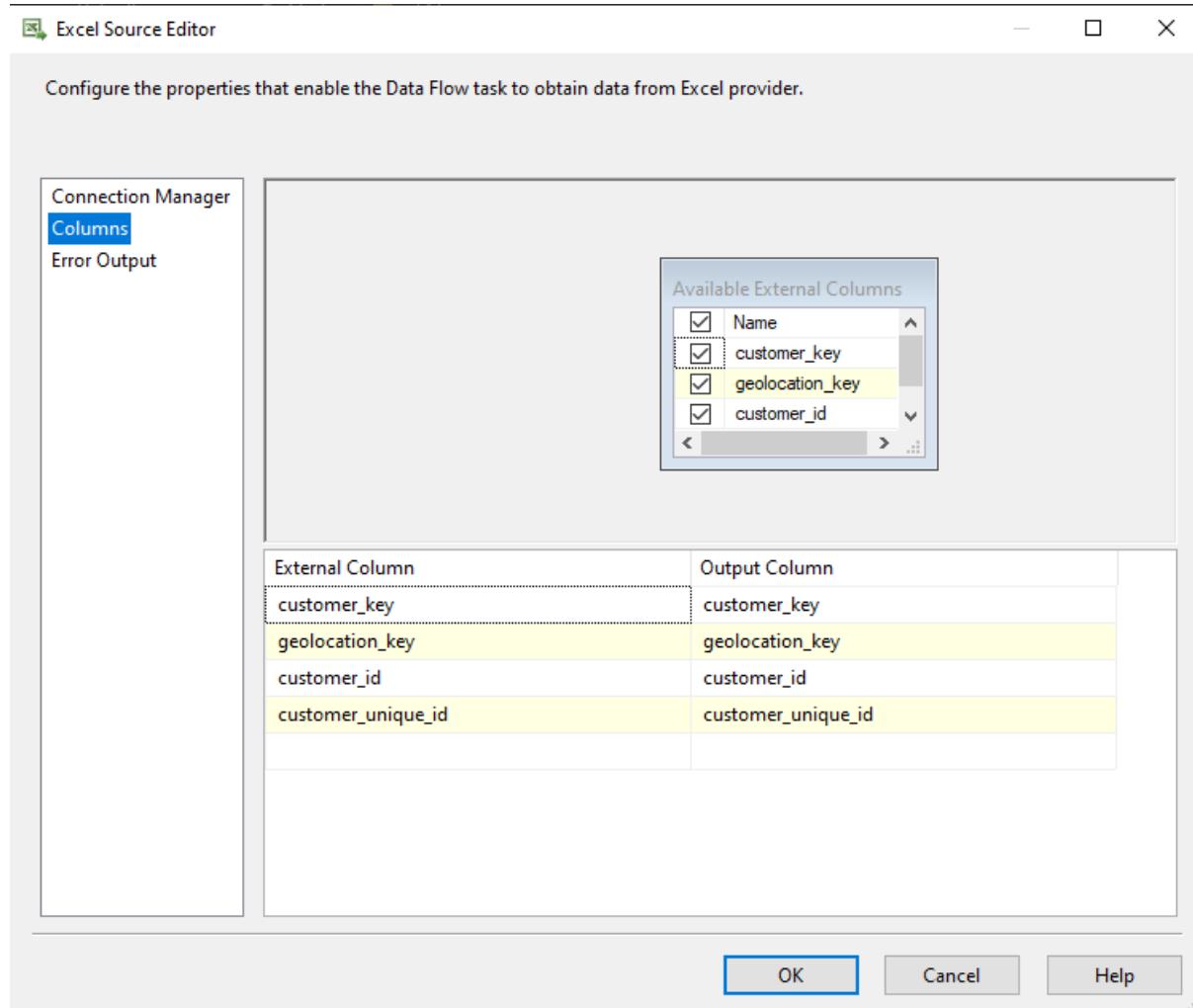
- Preview:



customer_id	geolocation_key	customer_id	customer_unique_id
-1	-1	Unknown ...	Unknown ...
1	10241	4d8c11a7...	30344e41...
2	4105	1e34d5f2...	0ad1d793...
3	17561	983cf42e...	c82aca6a...
4	15824	9c2937f3...	36cff82c...
5	8143	3eb7136a...	c30b3a12...
6	15463	3b099363...	5dc6d4b5...
7	5299	e6a94dcb...	c61a1e49...
8	6428	01726eb7...	f8bf0d2e...
9	6763	94117baa...	109e7a34...
10	8404	931b4625...	9210007a...
11	14550	b034d086...	bd95d212...
12	14551	b034d086...	bd95d212...
13	1724	075205e0...	7d2464-2

Close

- Columns:



Configure the properties that enable the Data Flow task to obtain data from Excel provider.

**Connection Manager**  
**Columns** (selected)  
Error Output

**Available External Columns**

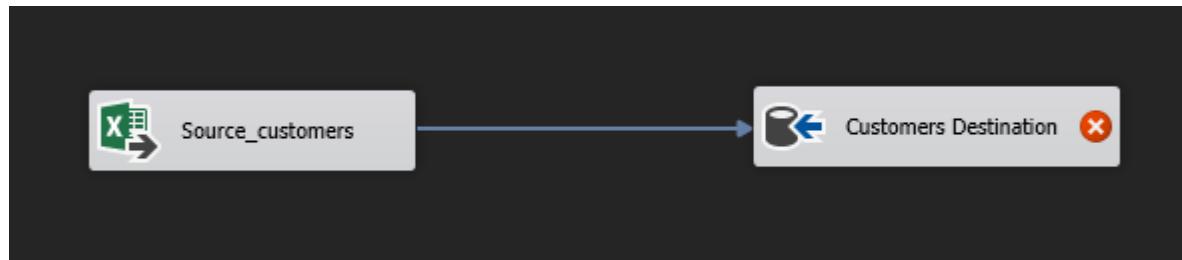
Name
customer_key
geolocation_key
customer_id
customer_unique_id

**External Column**      **Output Column**

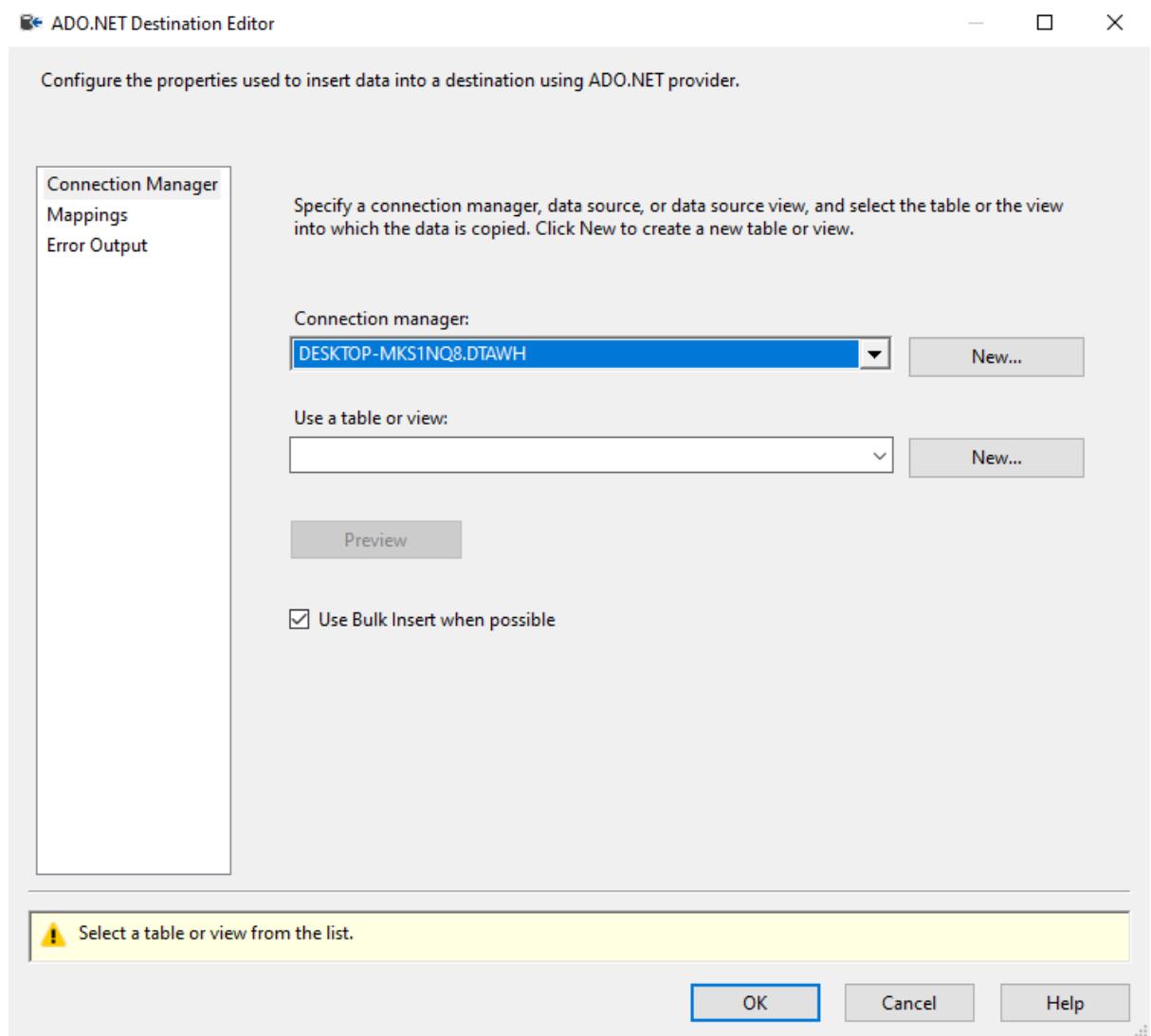
customer_key	customer_key
geolocation_key	geolocation_key
customer_id	customer_id
customer_unique_id	customer_unique_id

OK Cancel Help

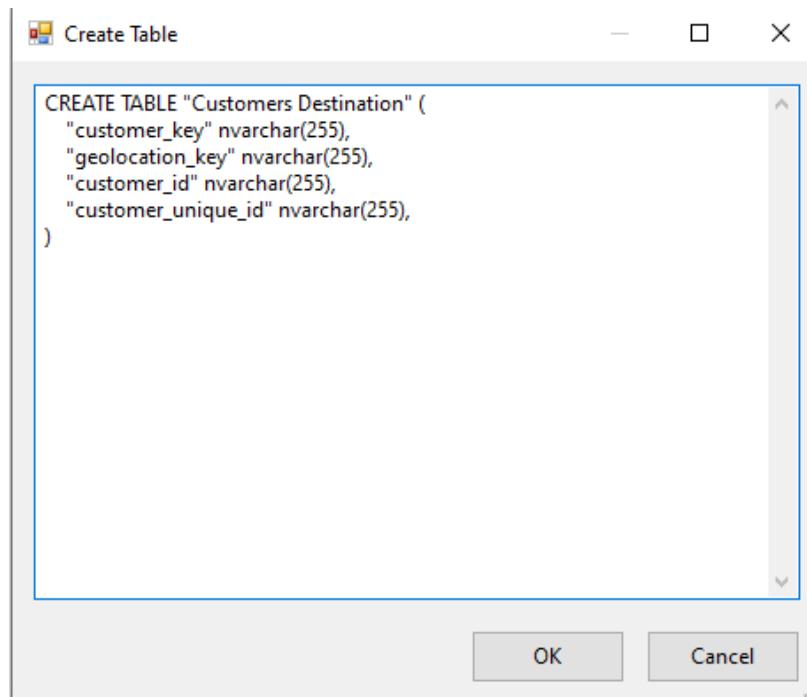
- Nhấn **OK**. Và kéo liên kết xanh đến **Customers Destination**:



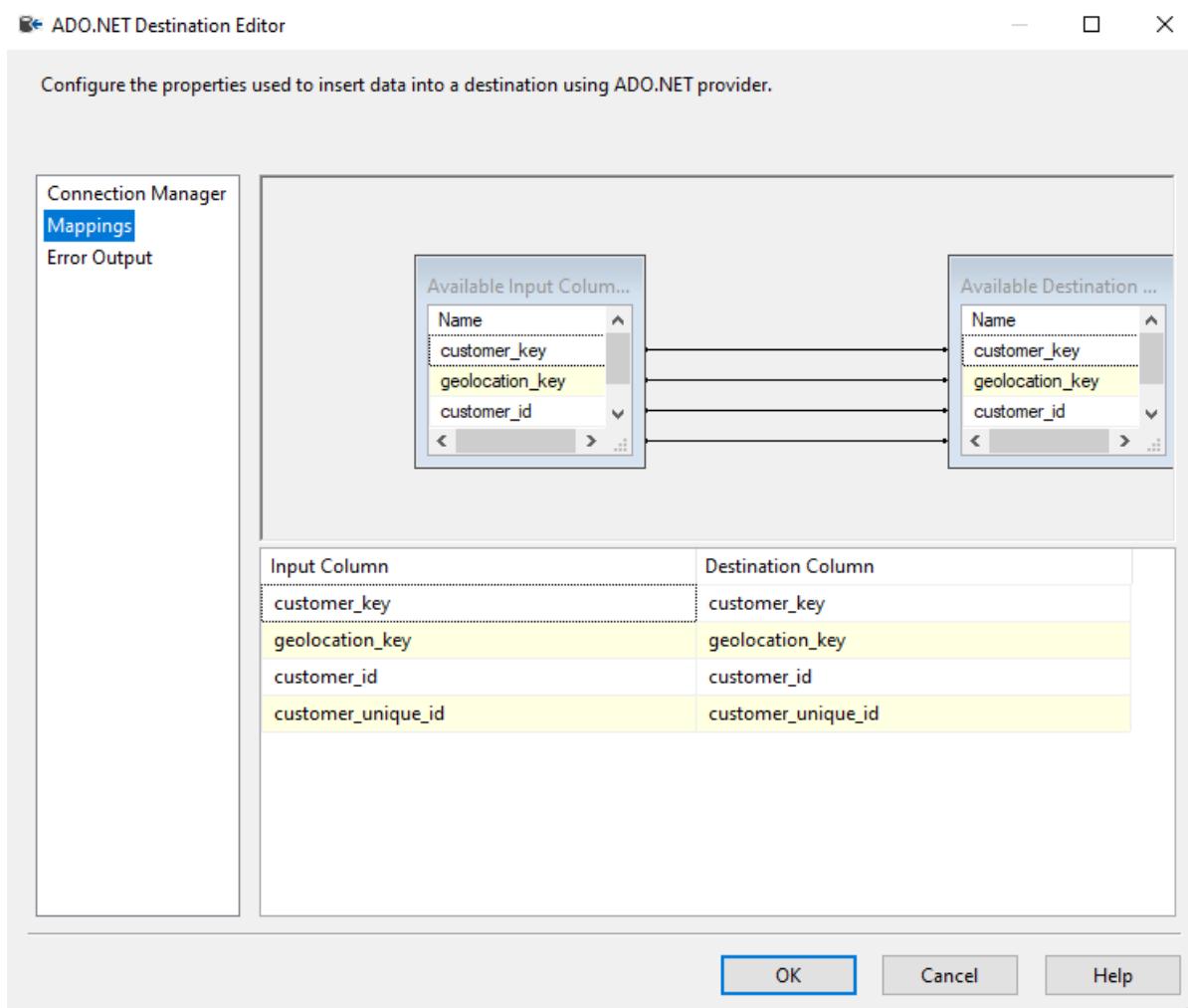
- Nhấn đúp chuột vào **Customers Destination** :



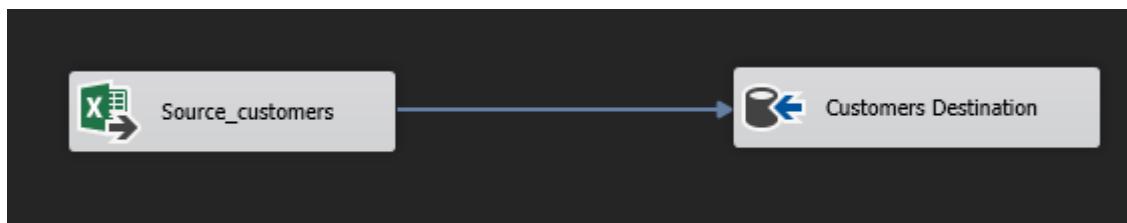
- Án **New...** → **OK** để tạo table mới



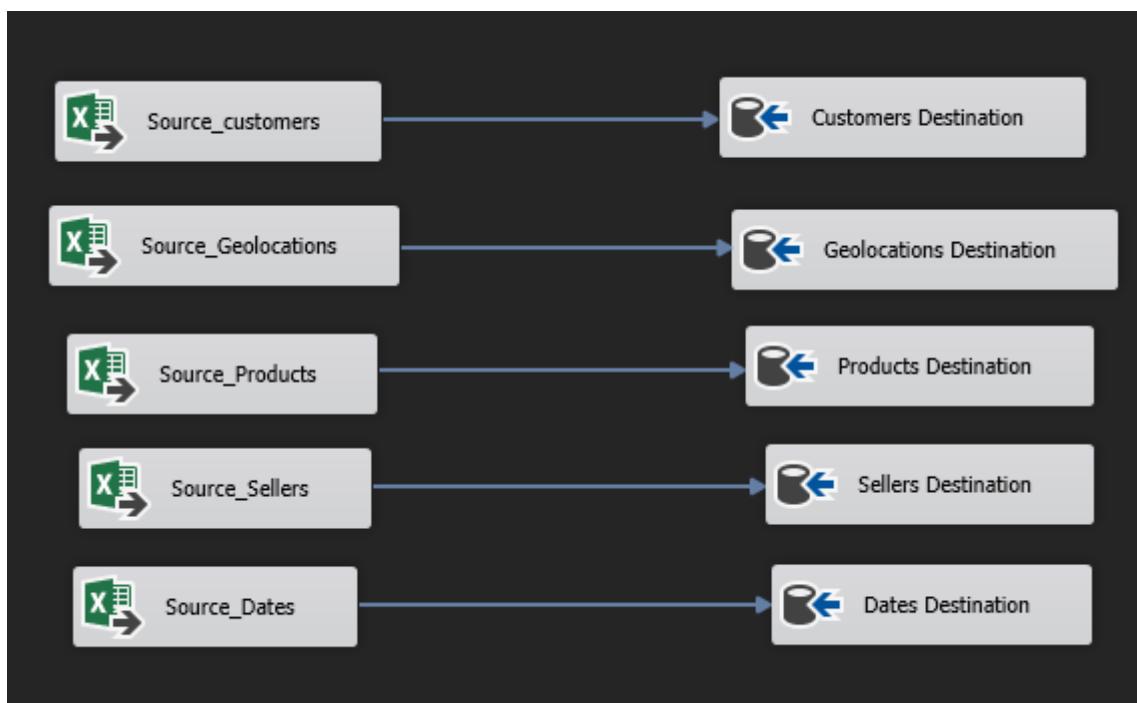
- **Mapping:**



- Nhấn **OK** để hoàn thành. Và ta được 1 cặp nguồn – đích Customers



- Thực hiện tương tự với các cặp nguồn – đích khác:



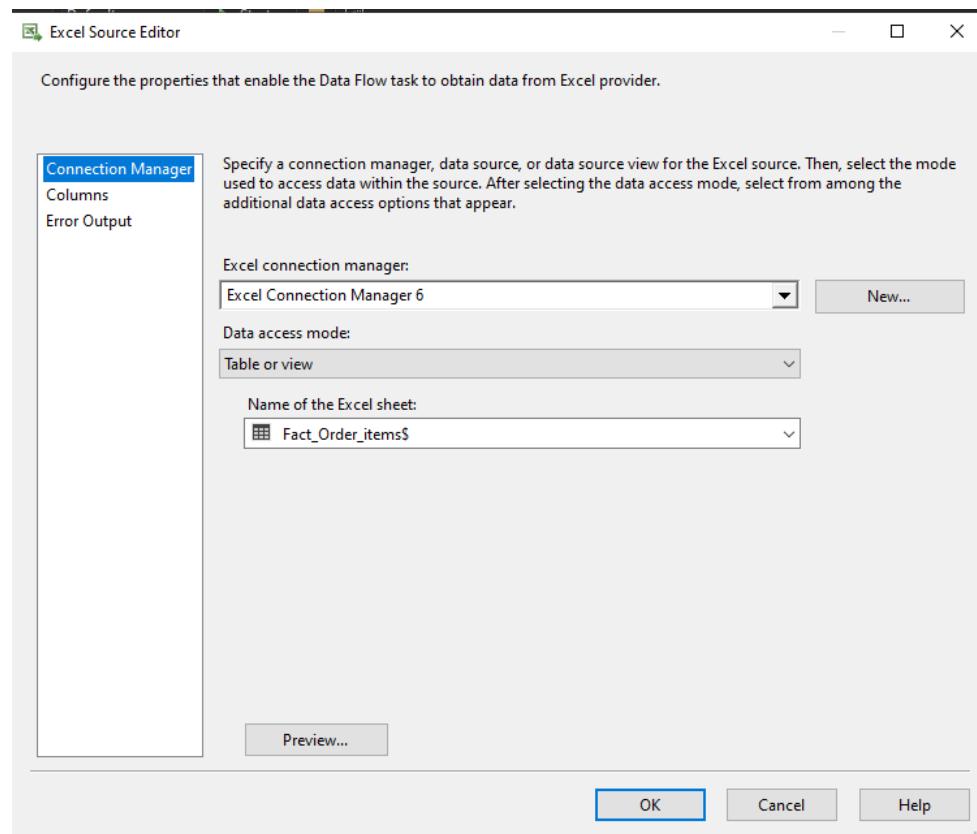
### 3.4.2. Data Flow Task 1:

- Từ thanh ToolBox, kéo thả 1 **Data Flow Task 1**

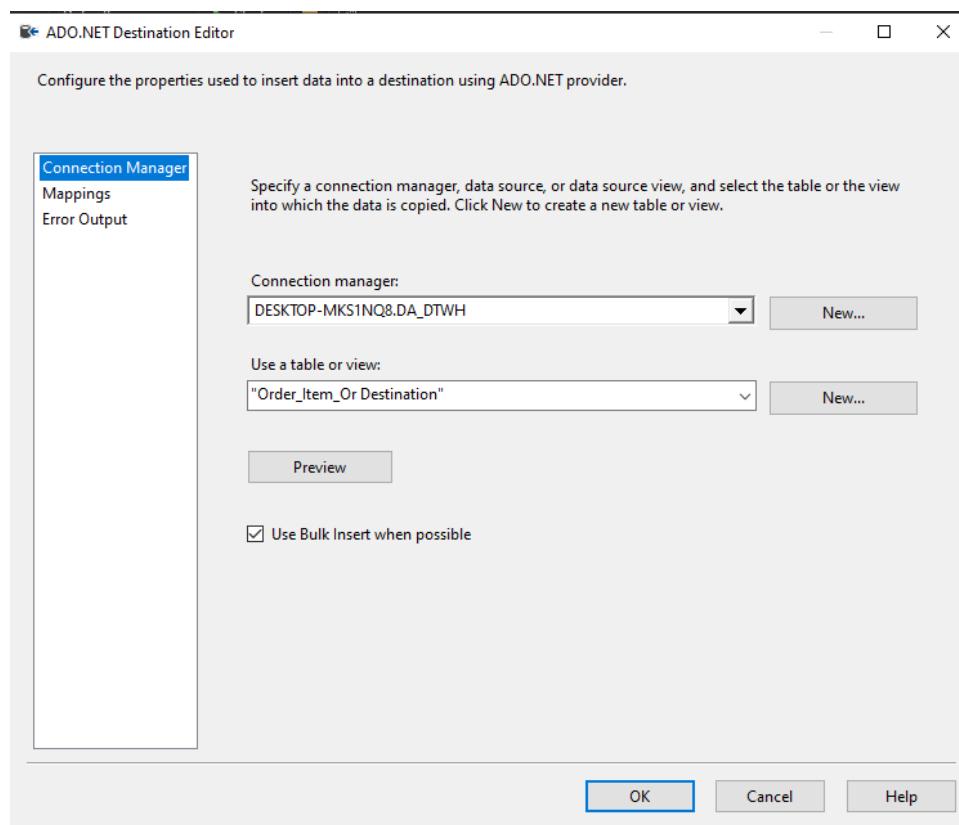


- Nháy chuột vào **Data Flow Task1** để định nghĩa, tương tự **Data flow Task**.

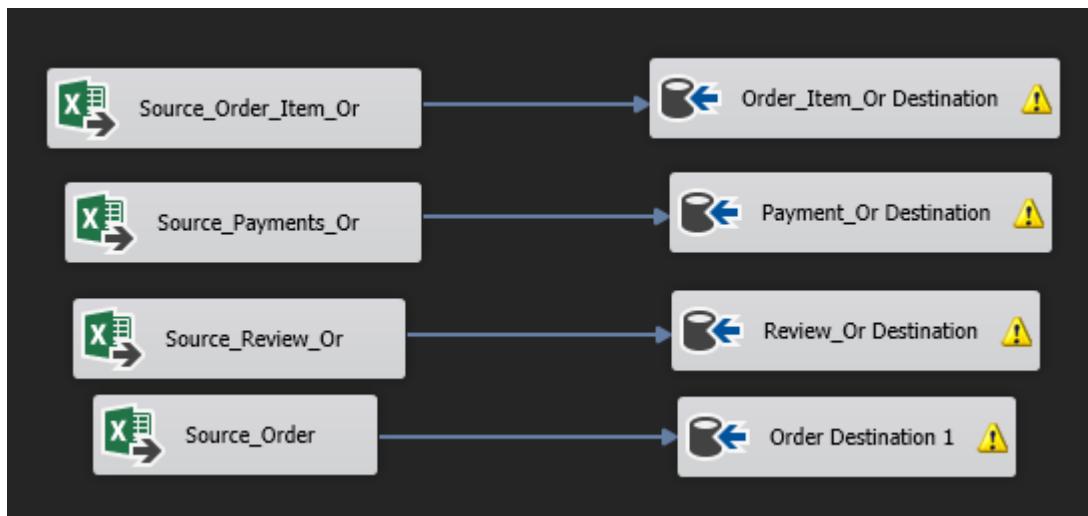
- **Source\_Order\_Item\_Or:**



- **Order\_Item\_Or Destination:**

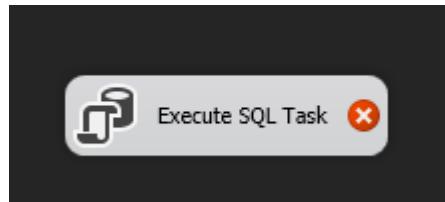


- Tương tự cho cặp nguồn – đích **Source\_Payments\_Or – Payments\_Or Destination** và **Source\_Review\_Or – Review\_Or Destination**. Ta được:

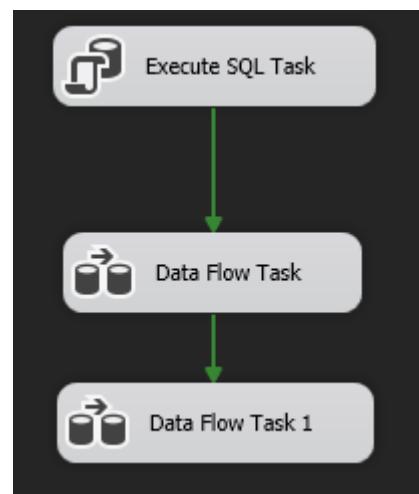


### 3.5. Khởi tạo Execute SQL Task:

- Kéo thả **Excute SQL Task** ở thanh công cụ **ToolBox**:



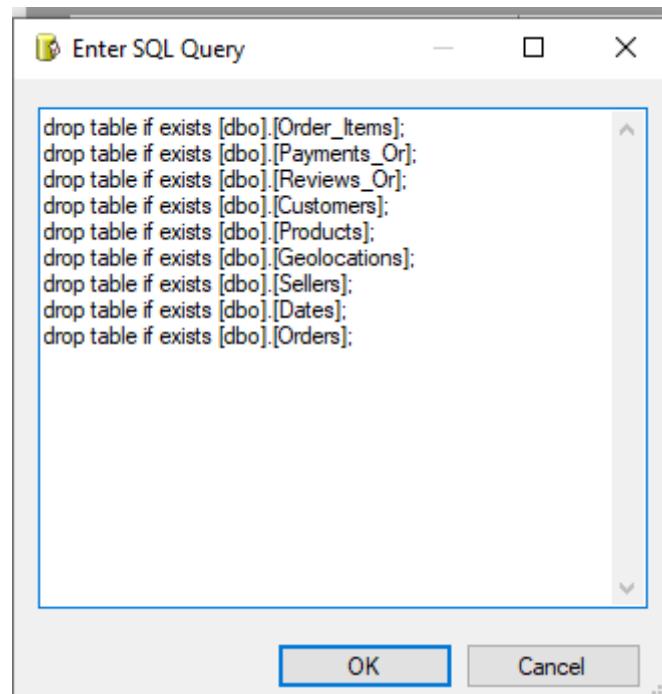
- Kéo đường dẫn (mũi tên xanh) đến Data Flow Task, tương tự kết nối các Data Flow Task với nhau:



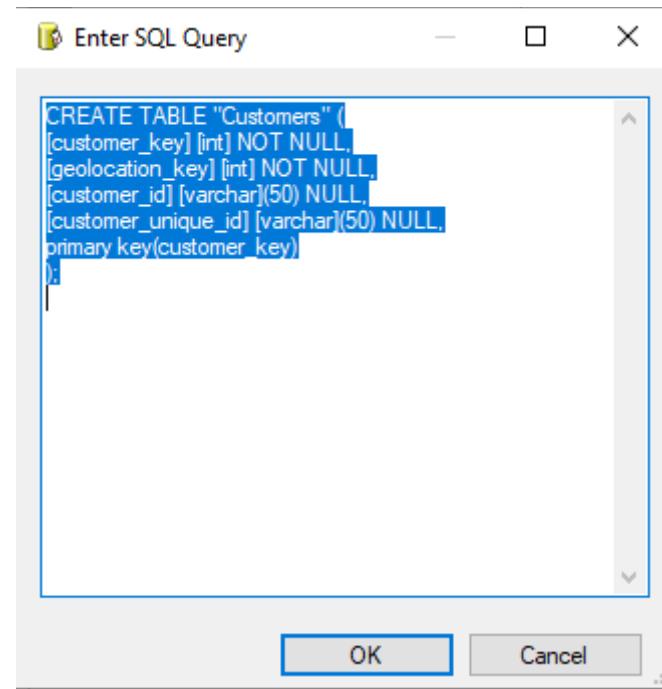
- Nhấn đúp **Execute SQL Task** → Chọn **Connection Type** là **ADO.NET**, Với Connection là:

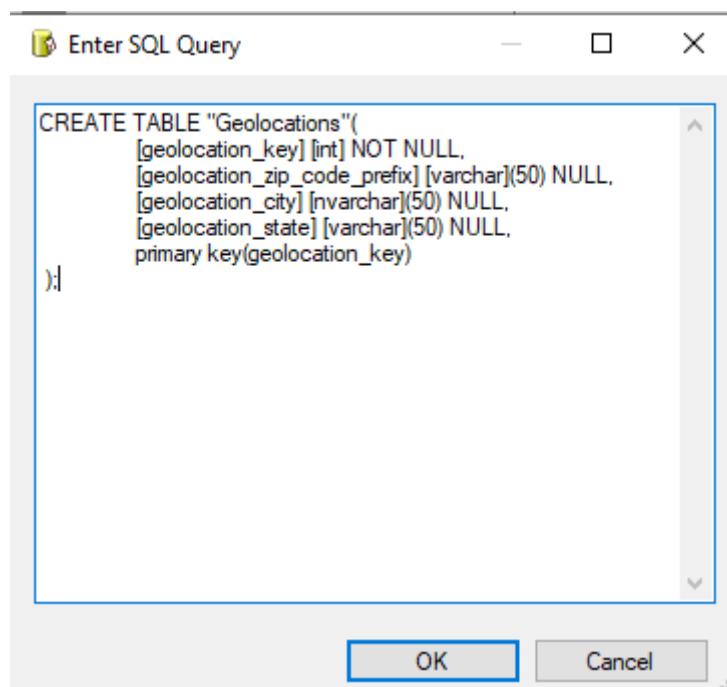
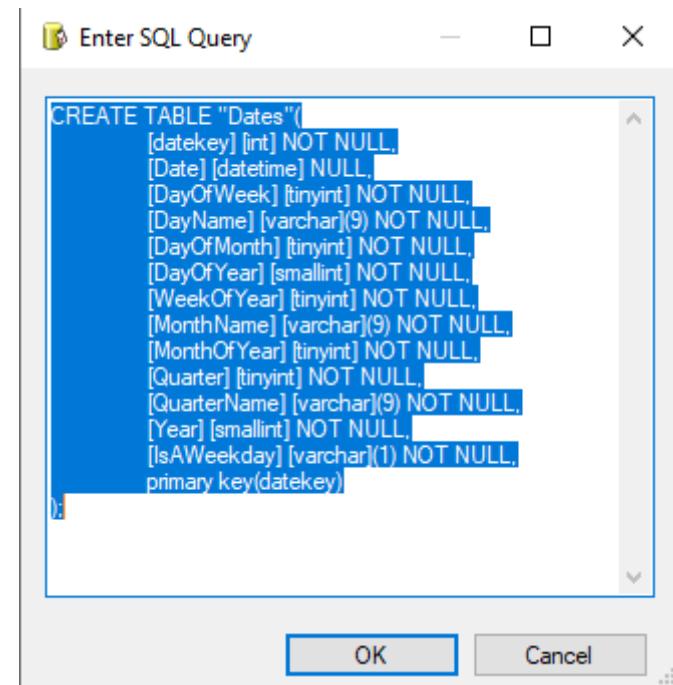
## DESKTOP-MKS1NQ8.DTAWH

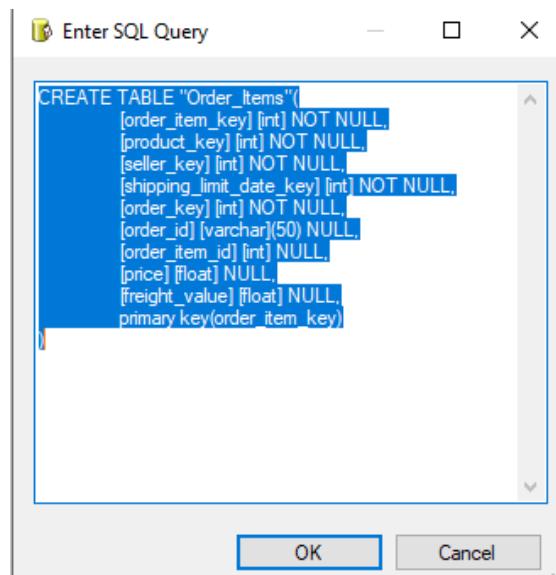
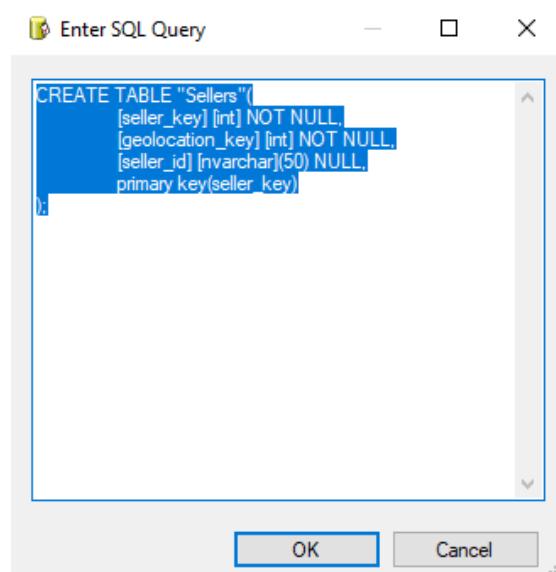
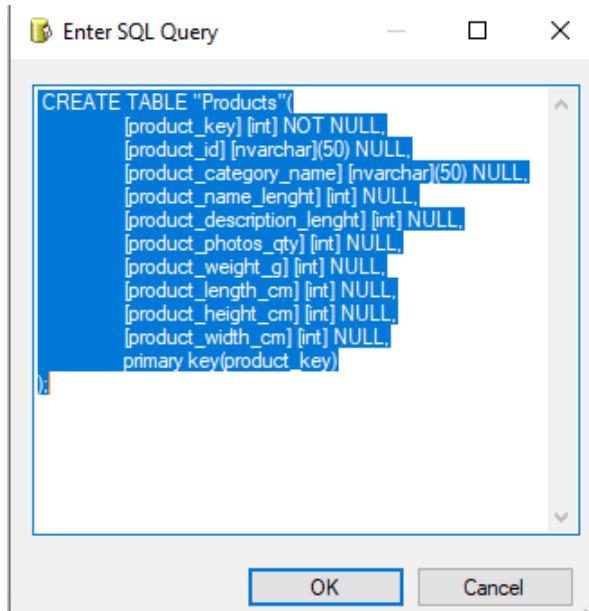
- Nhập SQLStatement → OK

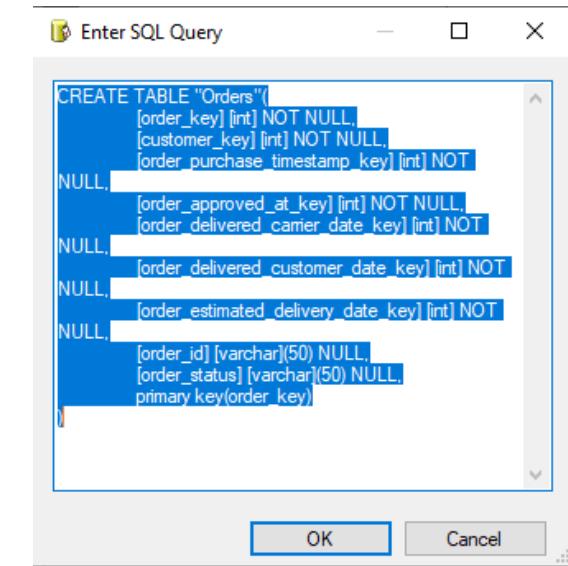
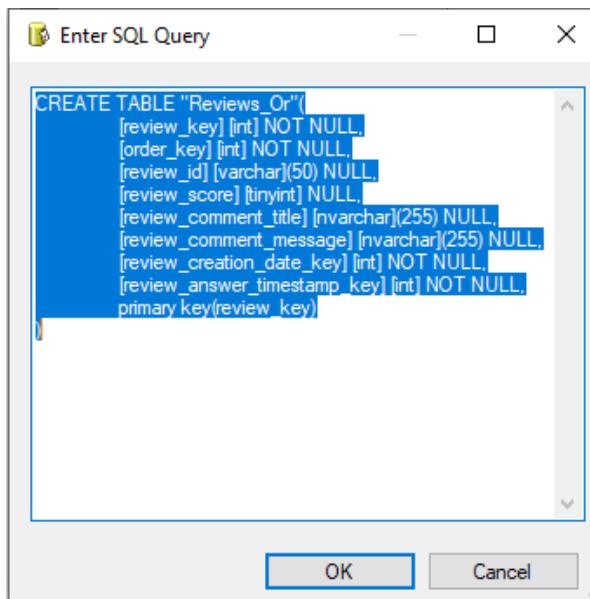
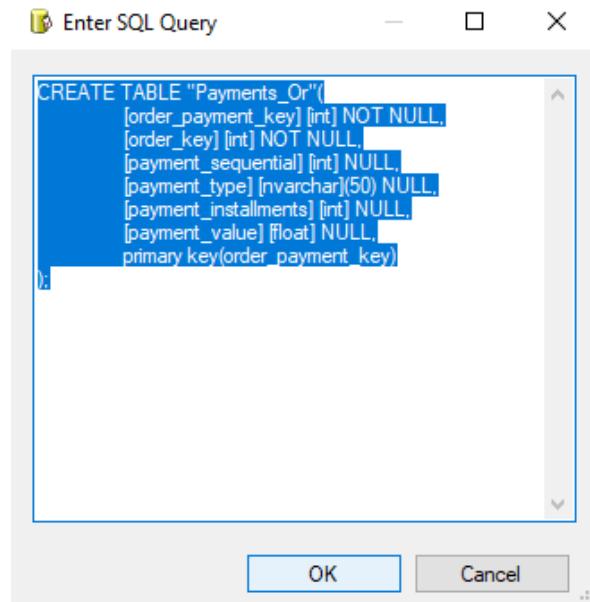


### 3.6. Khởi tạo các Execute SQL Task create table:

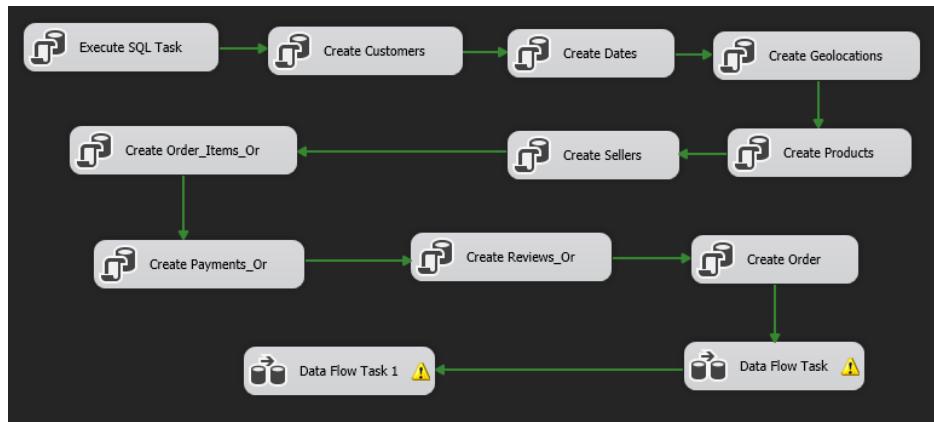




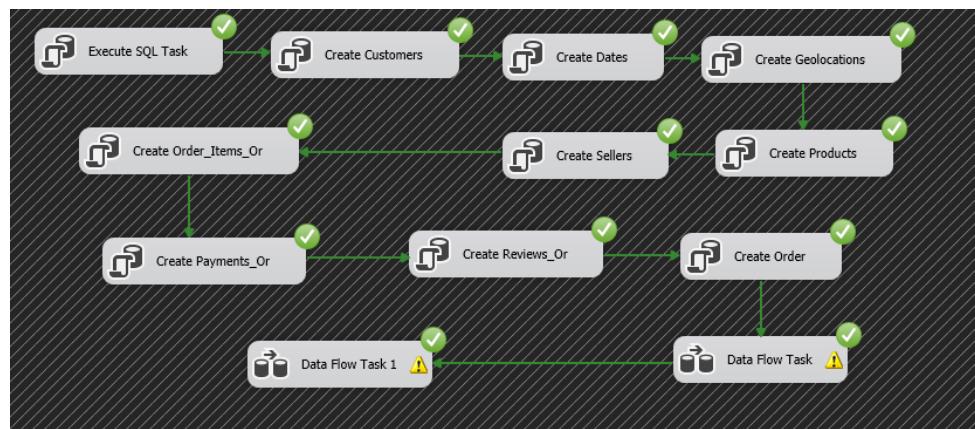




- Sau khi thực hiện thêm các SQL Execute Task:



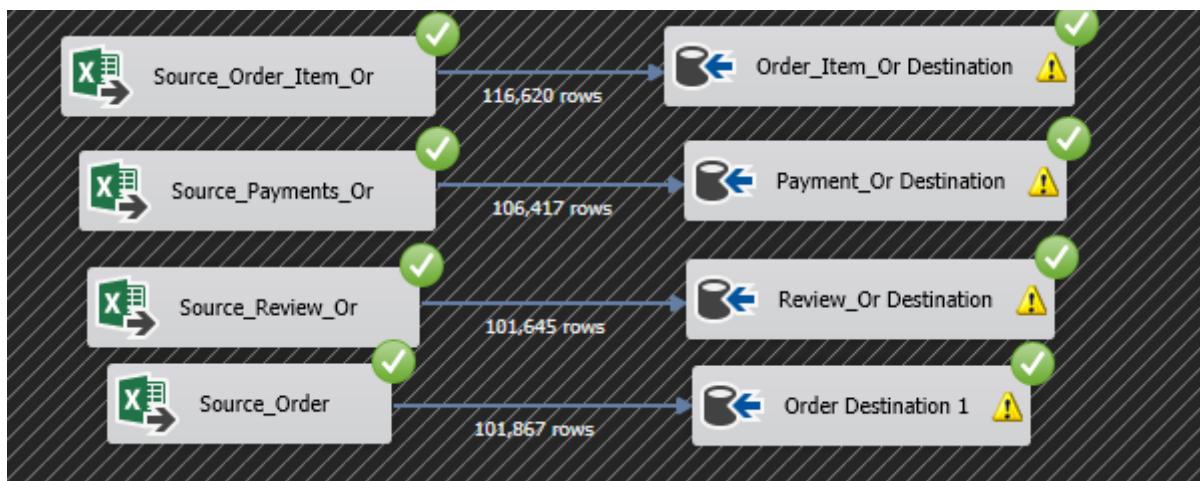
### 3.7. Khởi động project và xem xét kết quả thu được:



#### 3.7.1. Data Flow Task:



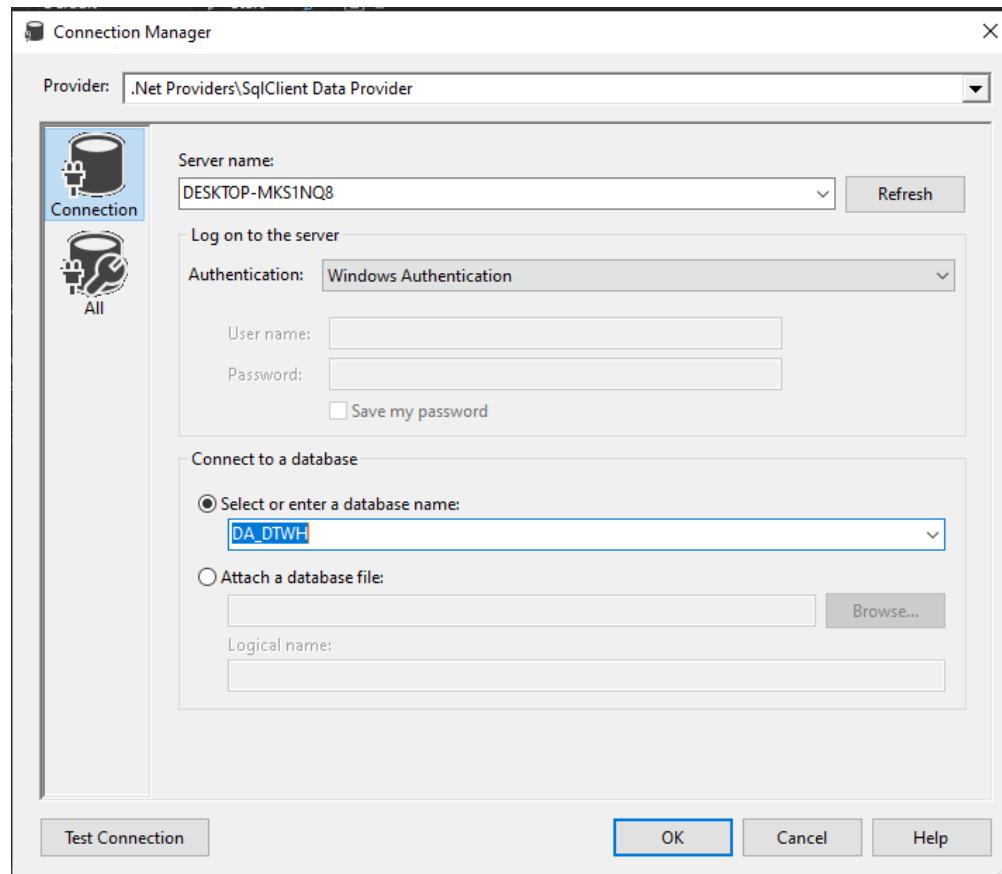
#### 3.7.2. Data Flow Task 1:



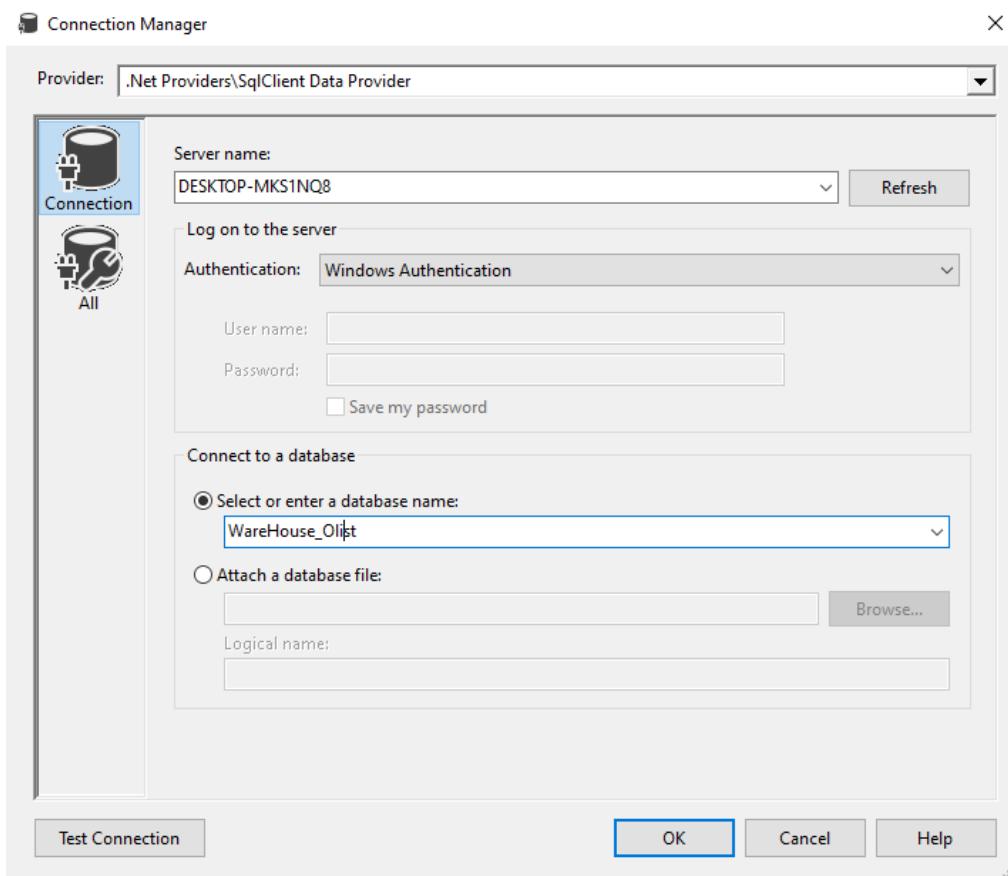
### 3.8. Quá trình đổ dữ liệu từ database vào kho dữ liệu:

#### 3.8.1. Tạo Connection Management:

- Tạo link nguồn dữ liệu:

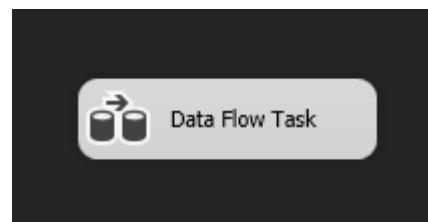


- Tạo link đích nhận dữ liệu:



### 3.8.2. Khởi tạo các Data Flow Task:

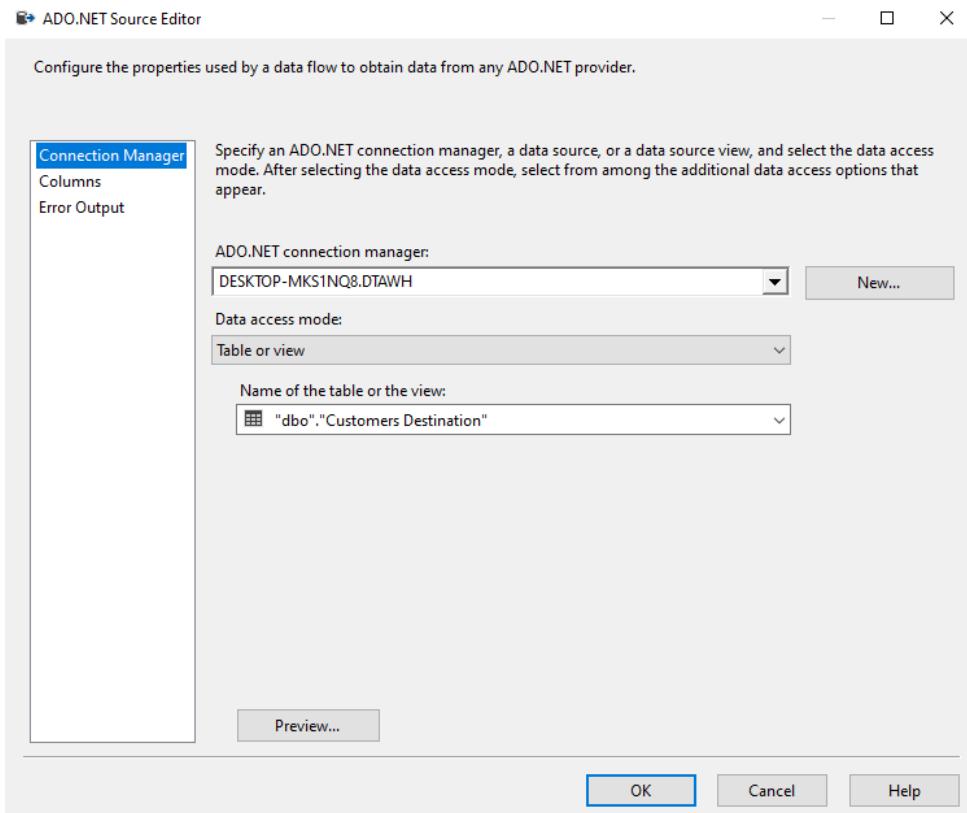
#### 3.8.2.1. Data Flow Task:



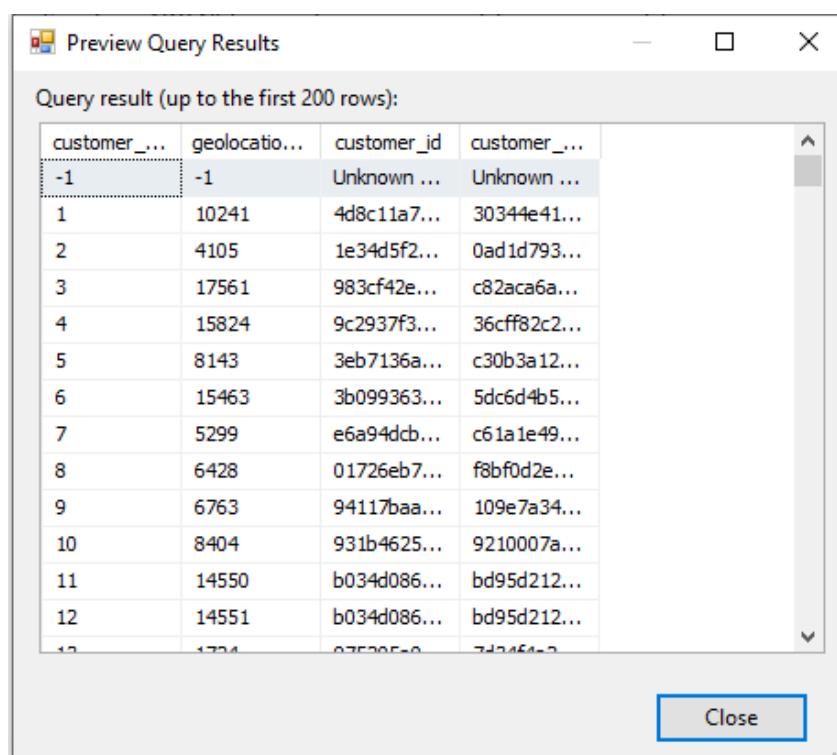
- Nháy chuột và thực hiện định nghĩa cho **Data Flow Task**.
- Kéo thả 1 **ADO Net Source** và **ADO NET Destination** từ thanh công cụ **Toolbox**.
- Thực hiện kéo dấu mũi tên xanh từ **ADO Net Source** đến **ADO NET Destination** và thực hiện đổi tên như hình:



- Nháy chuột vào **Source\_DimCustomer**. Tại **Connection Manager**, chọn đường dẫn đến link nguồn dữ liệu và chọn bảng nguồn.

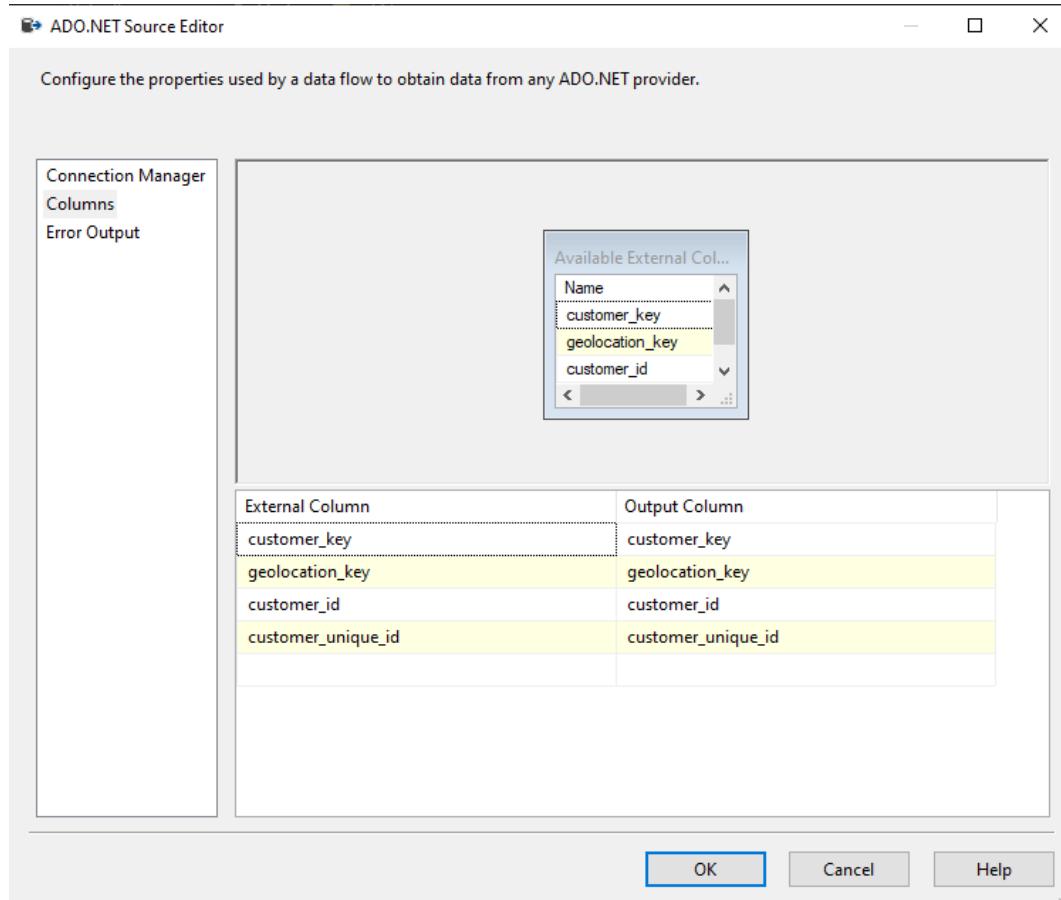


- Preview:

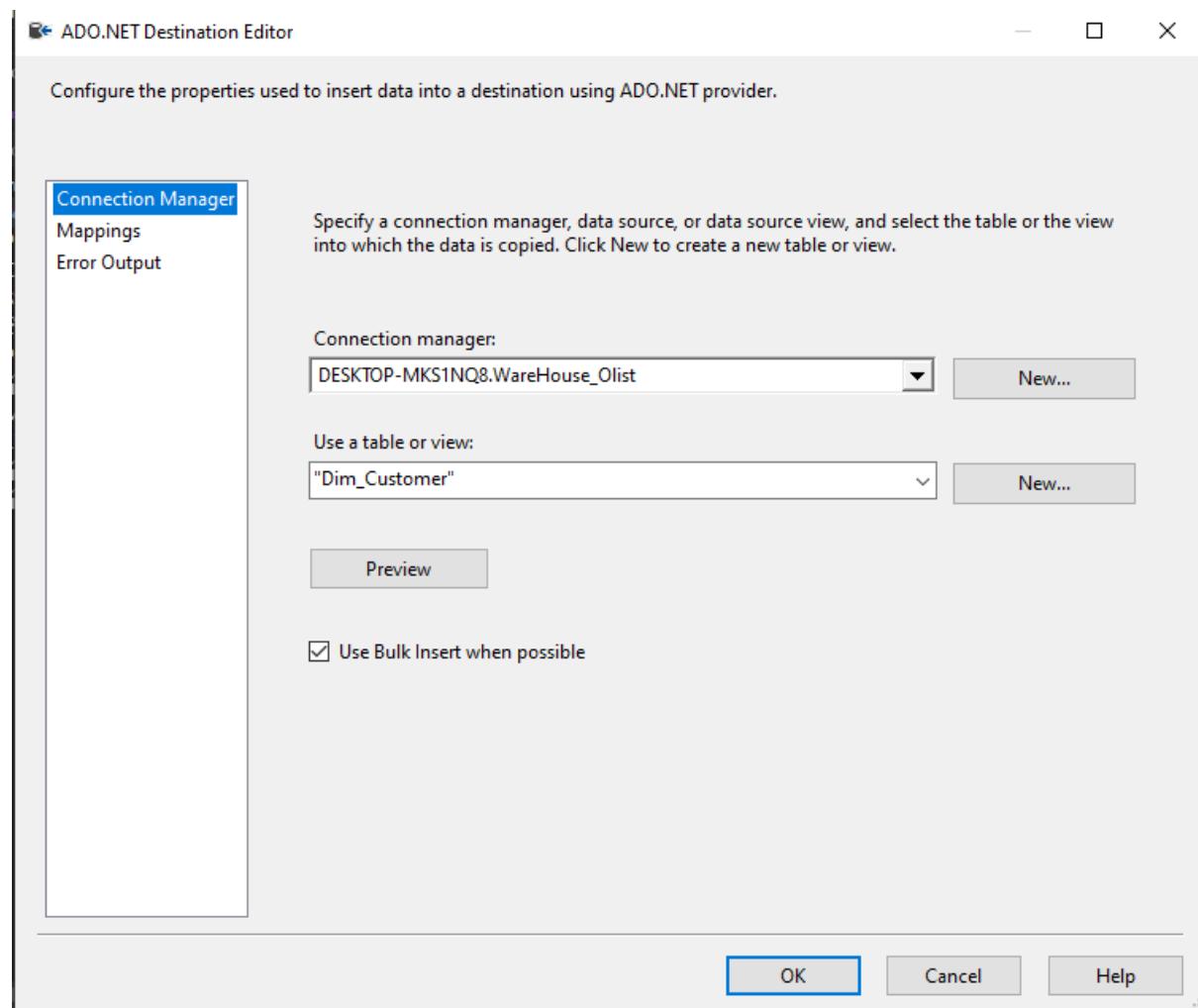


customer_id	geolocation_id	customer_id	customer_id
-1	-1	Unknown ...	Unknown ...
1	10241	4d8c11a7...	30344e41...
2	4105	1e34d5f2...	0ad1d793...
3	17561	983cf42e...	c82aca6a...
4	15824	9c2937f3...	36cff82c2...
5	8143	3eb7136a...	c30b3a12...
6	15463	3b099363...	5dc6d4b5...
7	5299	e6a94dc...	c61a1e49...
8	6428	01726eb7...	f8bf0d2e...
9	6763	94117baa...	109e7a34...
10	8404	931b4625...	9210007a...
11	14550	b034d086...	bd95d212...
12	14551	b034d086...	bd95d212...
13	1724	07e205e...	7d2454...

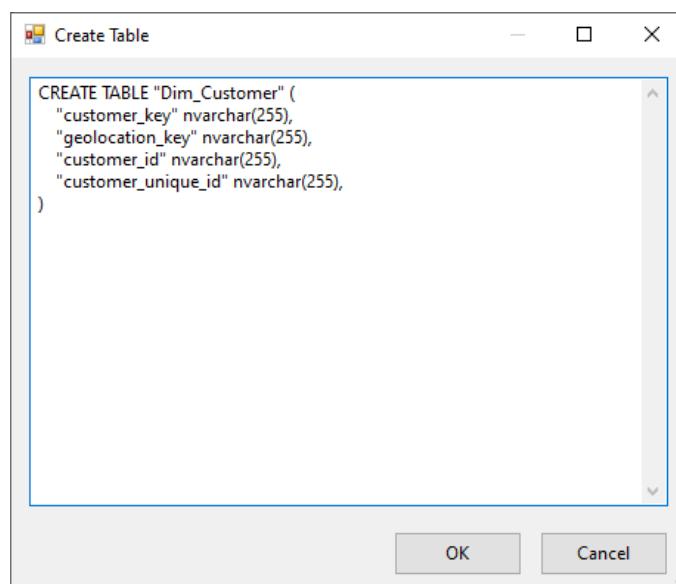
o Columns:



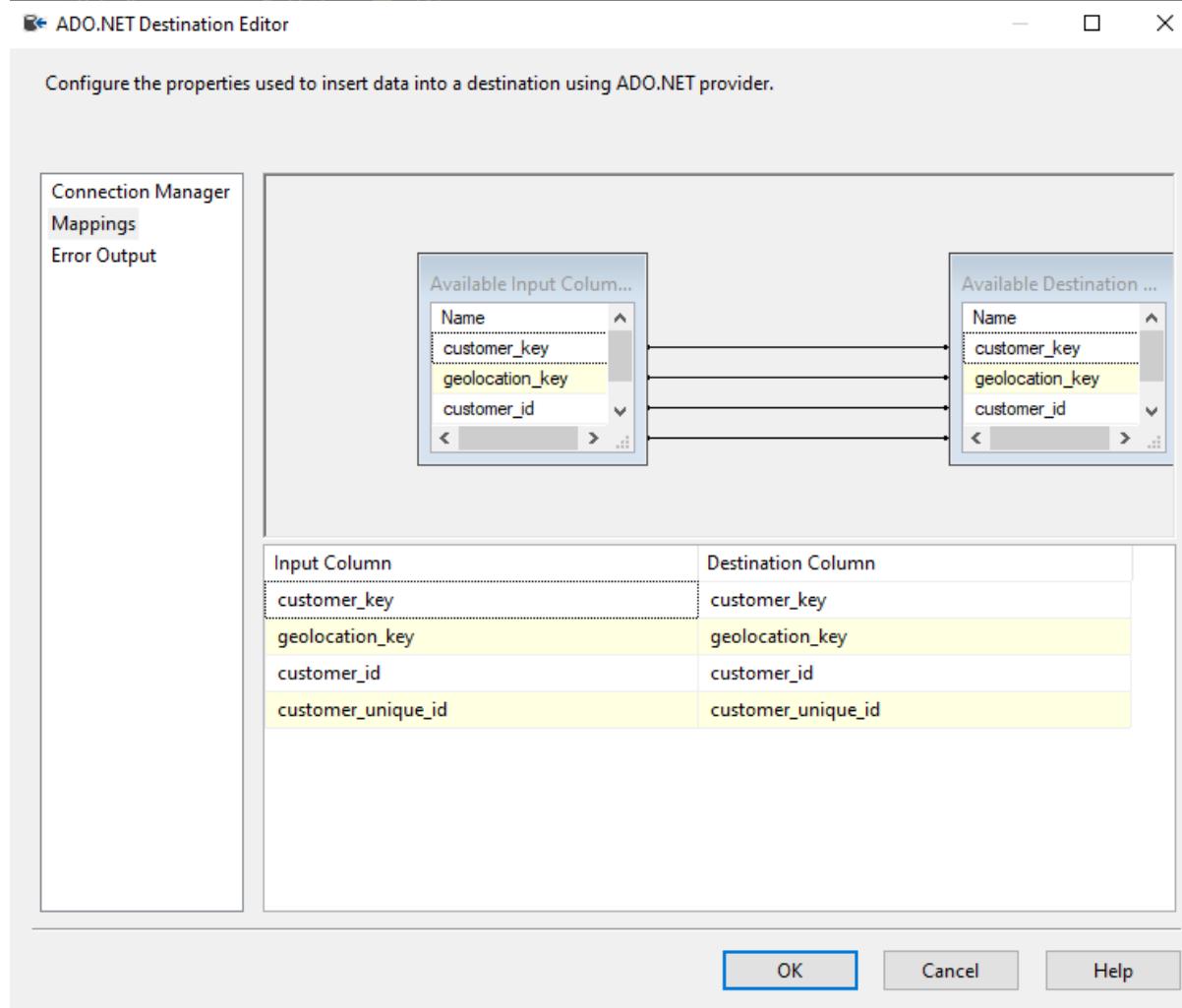
- Nháy đúp vào **Dim\_Customer**. Tại ô **Connection Manager**, chọn đường dẫn đến link dữ liệu đích nhận dữ liệu.



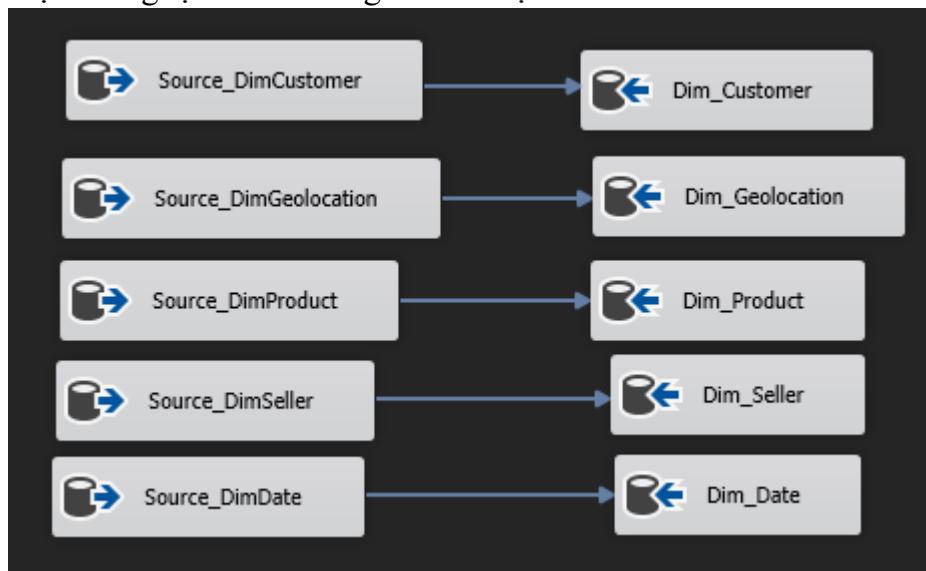
- Tại **Use a table or view**, chọn **New...** → **OK** để tạo bảng mới:



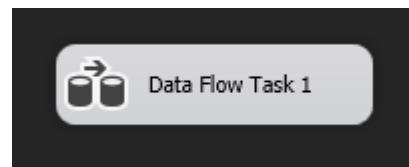
- o **Mappings:**



- Thực hiện tương tự với các bảng khác để tạo hoàn thành Data Flow Task:



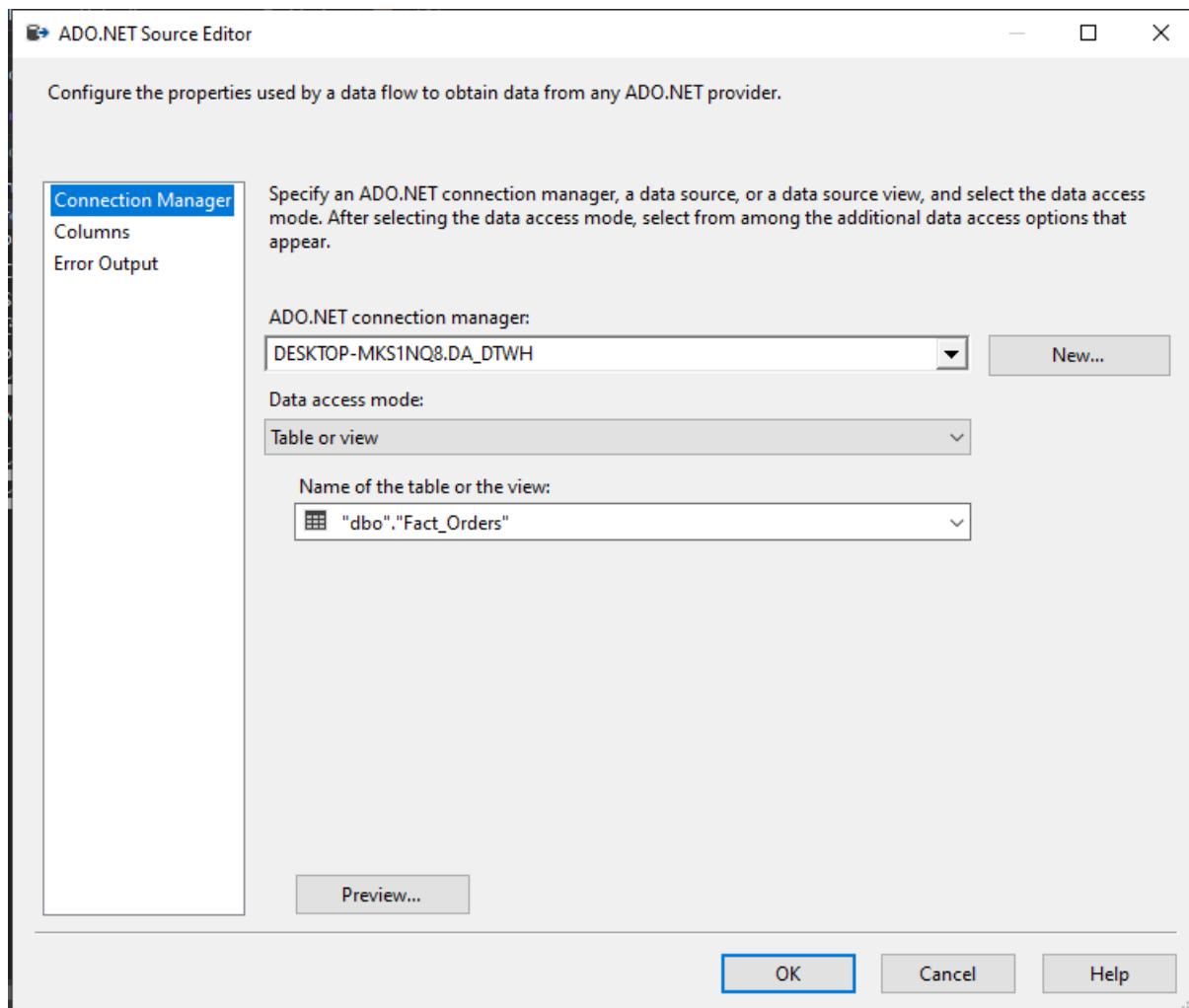
### 3.8.2.2. Data Flow Task 1:



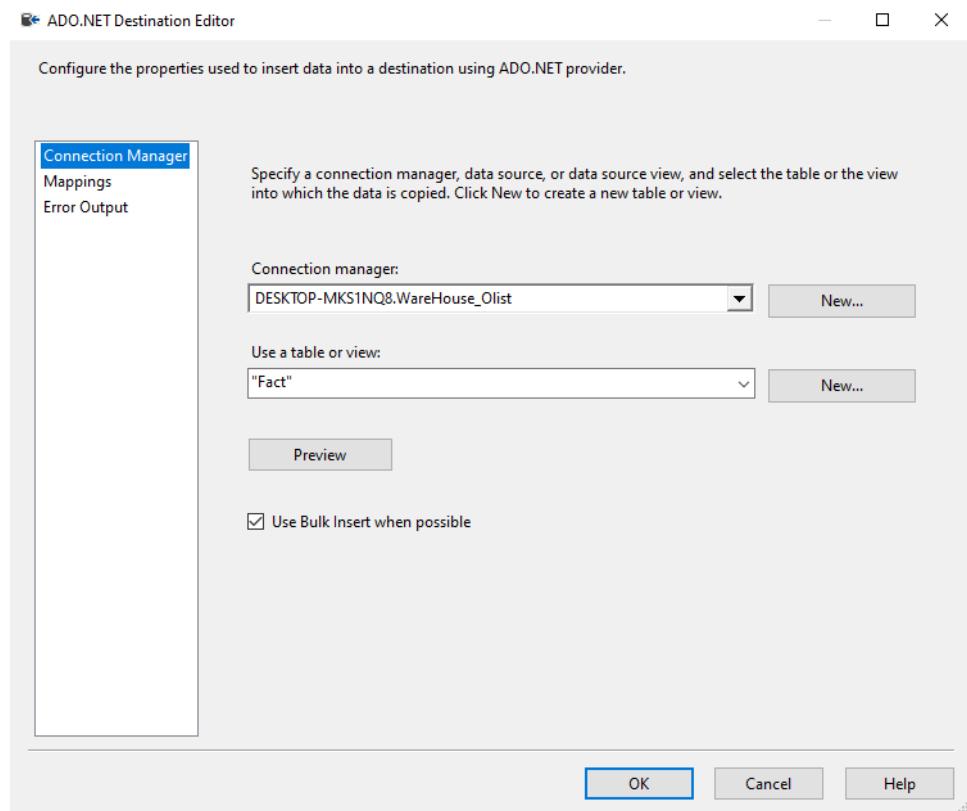
- Các bước thực hiện tương tự với **Data Flow Task**. Ta tạo **Data Flow Task 1** với mục đích để truyền dữ liệu cho bảng fact chính:



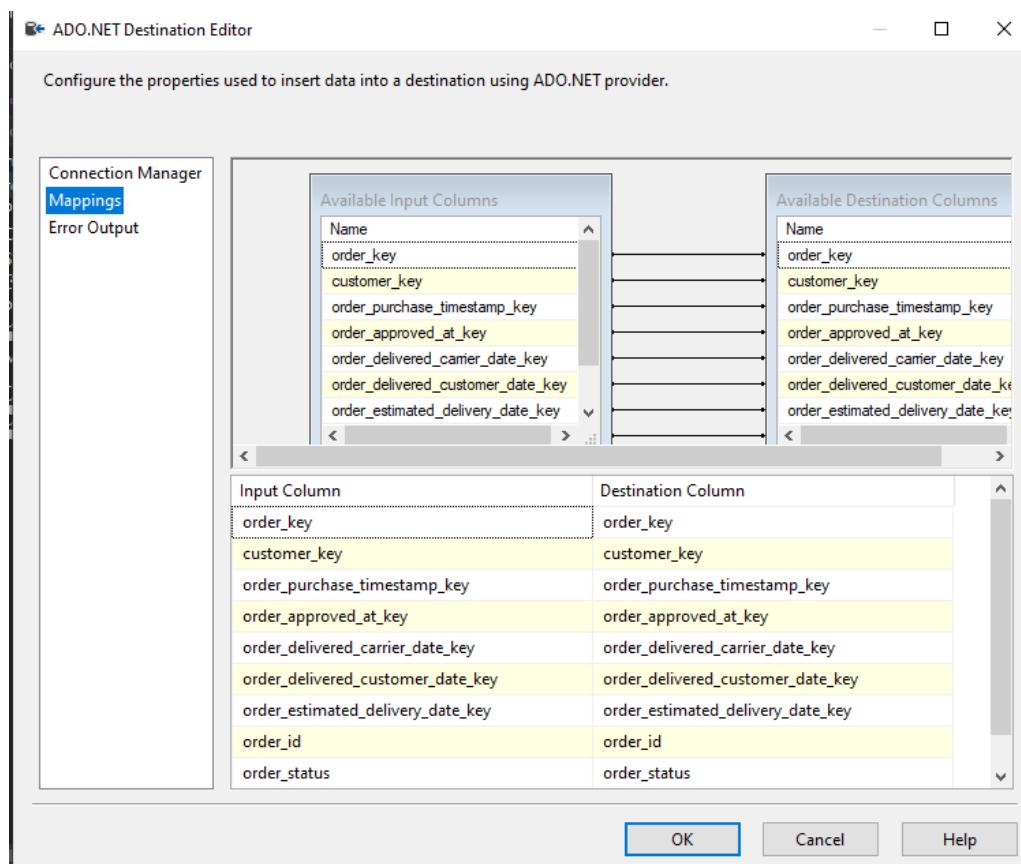
- **Source\_Order:**



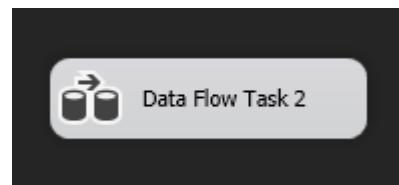
- **Fact:**



- Mappings:

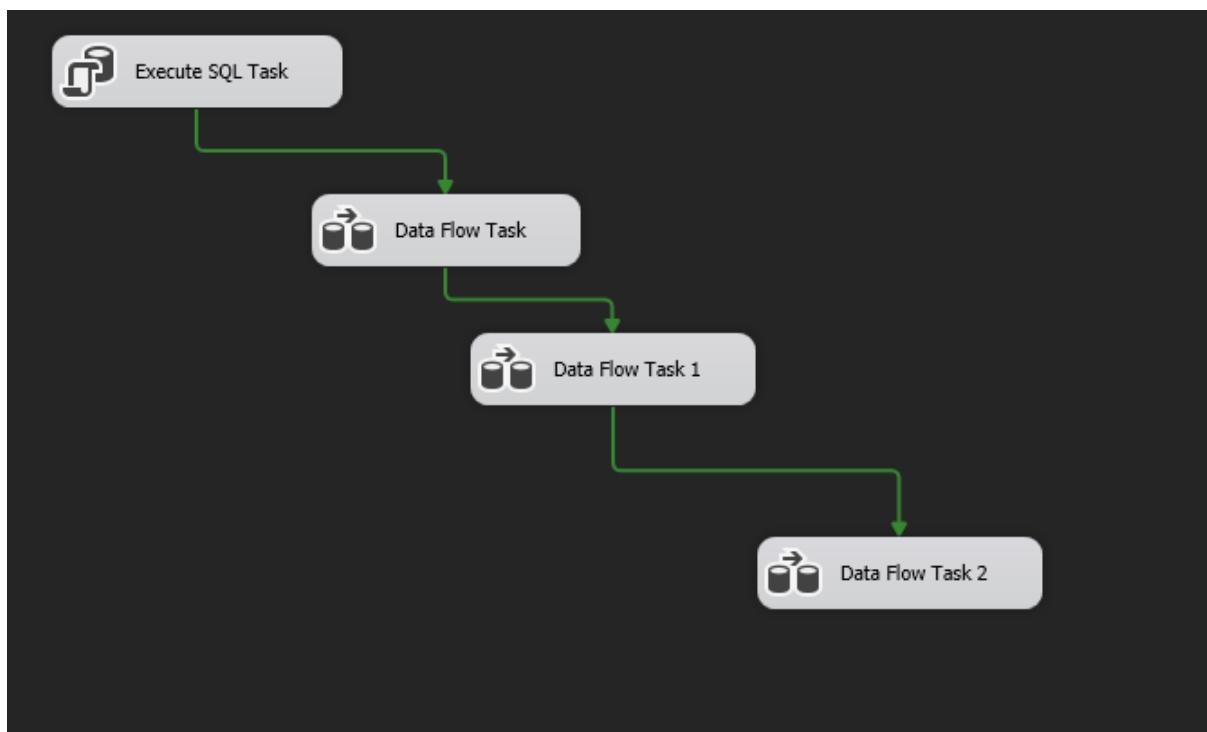


### 3.8.2.3. Data Flow Task 2:

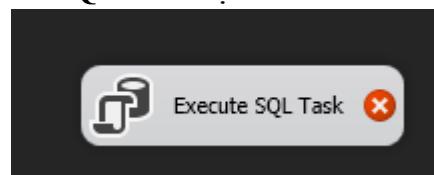


- Các bước thực hiện tương tự **Data Flow Task** và **Data Flow Task 1**. **Data Flow Task 2** được tạo ra nhằm mục đích khởi tạo và truyền dữ liệu cho các bảng fact con, hỗ trợ quá trình phân tích dữ liệu:

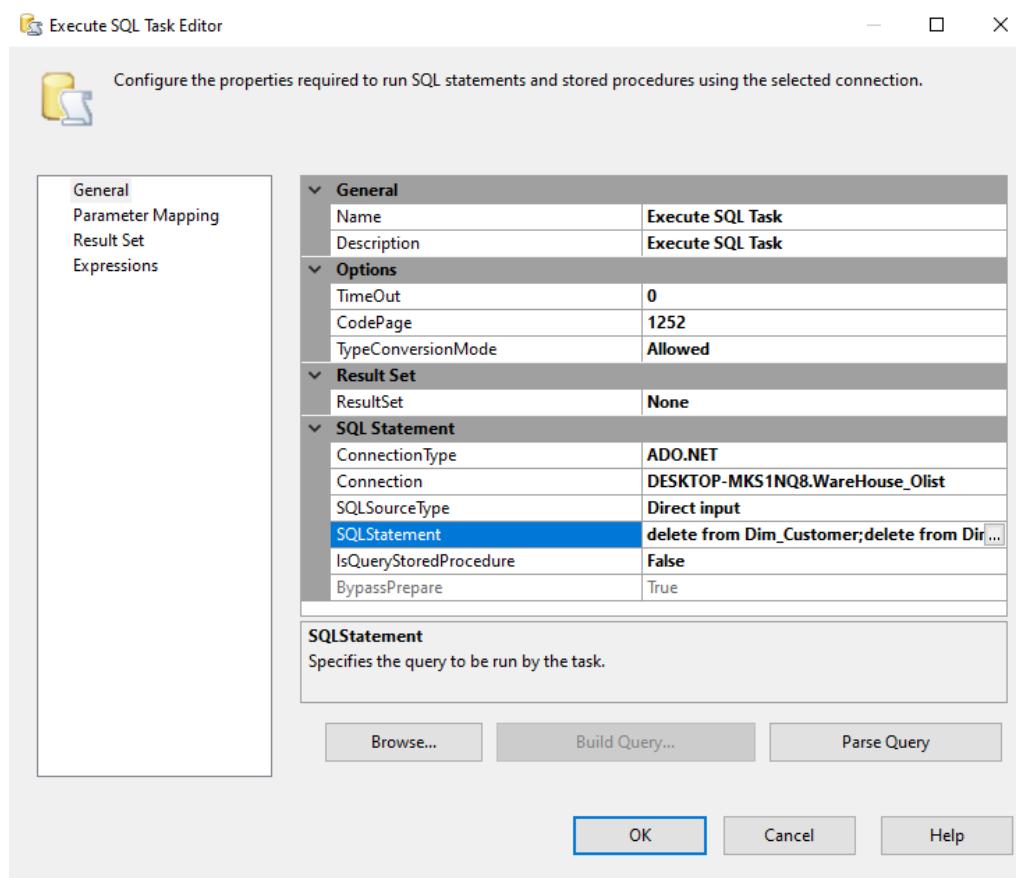
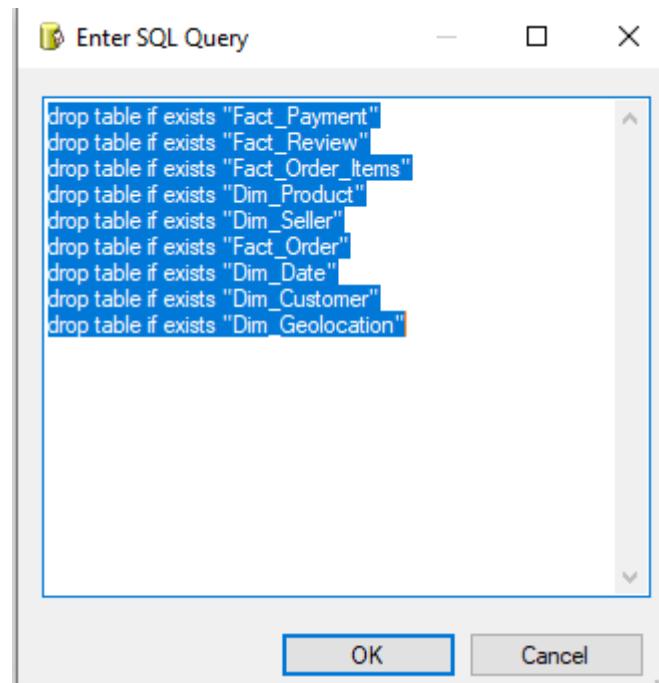
### 3.8.3. Tạo Execute SQL Task:



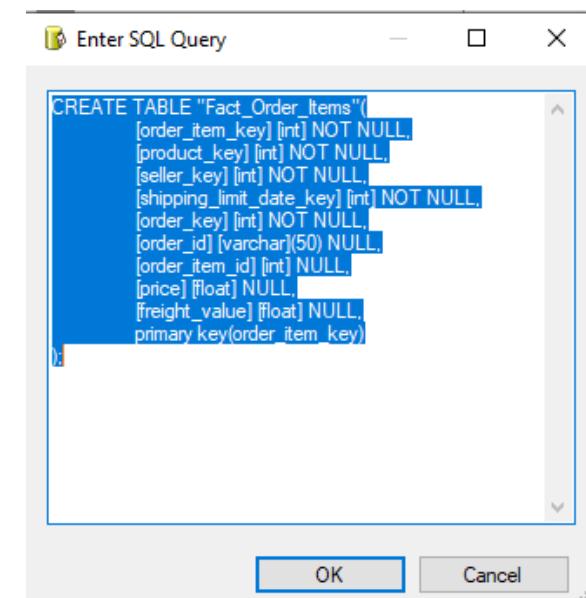
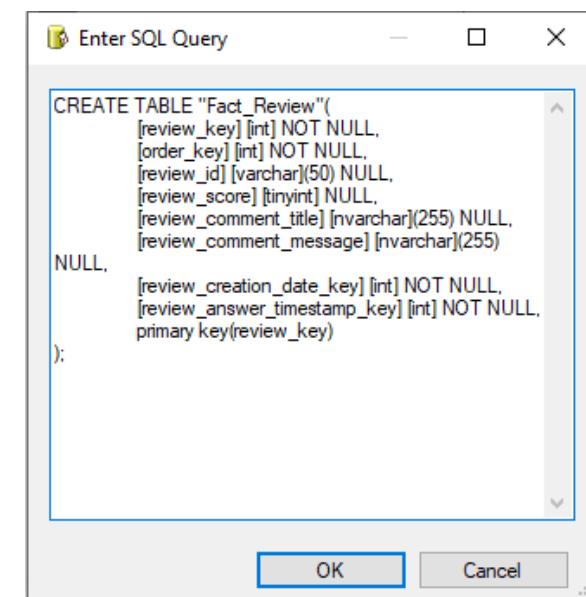
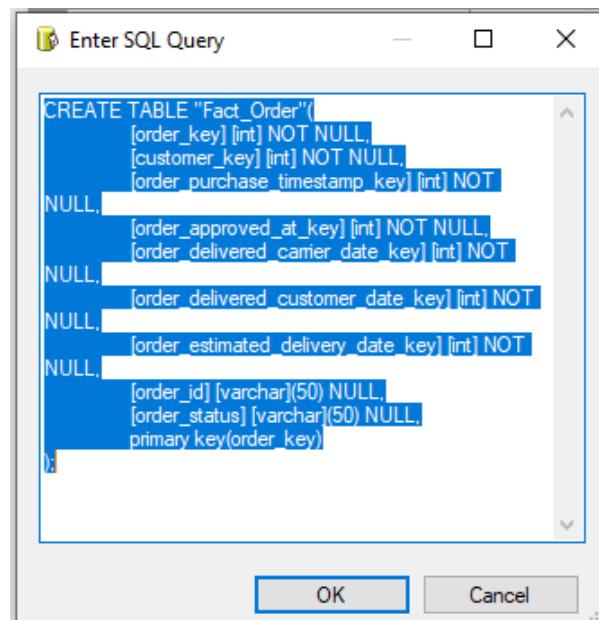
- Thực hiện kéo thả **Execute SQL Task** tại thanh **ToolBox**:



- Thực hiện định nghĩa **Execute SQL Task** bằng cách nháy chuột. Tại ô **Connection Type**, ta chọn **ADO.Net**. Tại ô **Connection**, ta thực hiện chọn đến link nguồn đích nhận dữ liệu.
- Điền dữ liệu vào **SQLStatement**:



### 3.8.4. Khởi tạo các EXECUTE SQL TASK CREATE TABLE:



Enter SQL Query

```
CREATE TABLE "Fact_Payment"(
    [order_payment_key] [int] NOT NULL,
    [order_key] [int] NOT NULL,
    [payment_sequential] [int] NULL,
    [payment_type] [nvarchar](50) NULL,
    [payment_installments] [int] NULL,
    [payment_value] [float] NULL,
    primary key(order_payment_key)
);
```

OK Cancel

Enter SQL Query

```
CREATE TABLE "Dim_Date"(
    [datekey] [int] NOT NULL,
    [Date] [datetime] NULL,
    [DayOfWeek] [tinyint] NOT NULL,
    [DayName] [varchar](9) NOT NULL,
    [DayOfMonth] [tinyint] NOT NULL,
    [DayOfYear] [smallint] NOT NULL,
    [WeekOfYear] [tinyint] NOT NULL,
    [MonthName] [varchar](9) NOT NULL,
    [MonthOfYear] [tinyint] NOT NULL,
    [Quarter] [tinyint] NOT NULL,
    [QuarterName] [varchar](9) NOT NULL,
    [Year] [smallint] NOT NULL,
    [IsAWeekday] [varchar](1) NOT NULL,
    primary key(datekey)
);
```

OK Cancel

Enter SQL Query

```
CREATE TABLE "Dim_Customer" (
    [customer_key] [int] NOT NULL,
    [geolocation_key] [int] NOT NULL,
    [customer_id] [varchar](50) NULL,
    [customer_unique_id] [varchar](50) NULL,
    primary key(customer_key)
);
```

OK Cancel

Enter SQL Query

```
CREATE TABLE "Dim_Geolocation"(
    [geolocation_key] [int] NOT NULL,
    [geolocation_zip_code_prefix] [varchar](50),
    NULL,
    [geolocation_city] [nvarchar](50) NULL,
    [geolocation_state] [varchar](50) NULL,
    primary key(geolocation_key)
);
```

OK Cancel

Enter SQL Query

```
CREATE TABLE "Dim_Product"(
    [product_key] [int] NOT NULL,
    [product_id] [nvarchar](50) NULL,
    [product_category_name] [nvarchar](50) NULL,
    [product_name_length] [int] NULL,
    [product_description_length] [int] NULL,
    [product_photos_qty] [int] NULL,
    [product_weight_g] [int] NULL,
    [product_length_cm] [int] NULL,
    [product_height_cm] [int] NULL,
    [product_width_cm] [int] NULL,
    primary key(product_key)
);
```

OK Cancel

Enter SQL Query

```
CREATE TABLE "Dim_Seller"(
    [seller_key] [int] NOT NULL,
    [geolocation_key] [int] NOT NULL,
    [seller_id] [nvarchar](50) NULL,
    primary key(seller_key)
);
```

OK Cancel

### 3.8.5. Khởi tạo EXECUTE SQL TASK ADD FOREIGN KEY:



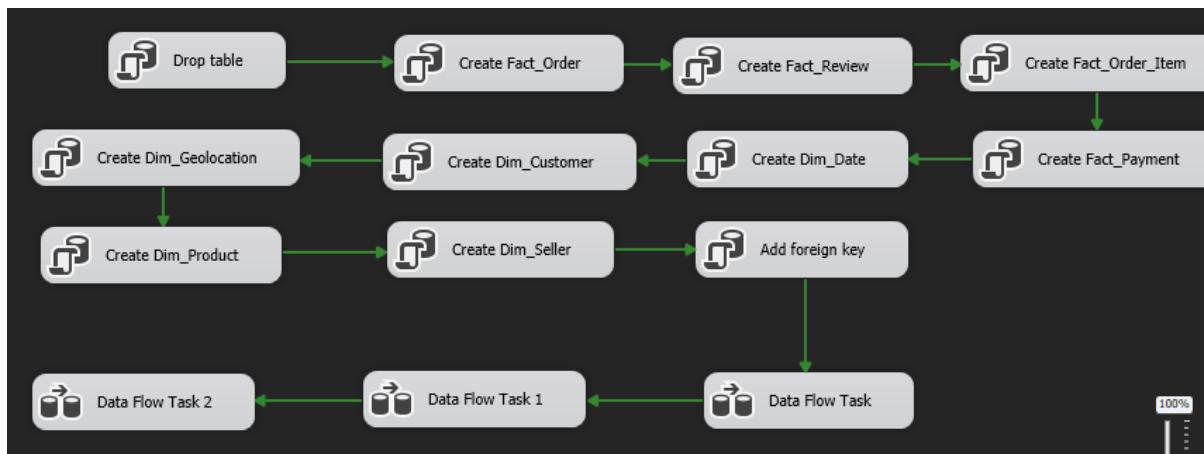
```

ALTER TABLE [dbo].[Dim_Date] ADD DEFAULT ('N') FOR [IsAWeekday]
GO
ALTER TABLE [dbo].[Dim_Customer] WITH CHECK ADD CONSTRAINT
[fkDimCustomers_geolocation_Key] FOREIGN KEY([geolocation_key])
REFERENCES [dbo].[Dim_Geolocation] ([geolocation_key])
GO
ALTER TABLE [dbo].[Dim_Customer] CHECK CONSTRAINT fkDimCustomers_geolocation_Key
GO
ALTER TABLE [dbo].[Dim_Seller] WITH CHECK ADD CONSTRAINT
[fkDimSellers_geolocation_Key] FOREIGN KEY([geolocation_key])
REFERENCES [dbo].[Dim_Geolocation] ([geolocation_key])
GO
ALTER TABLE [dbo].[Dim_Seller] CHECK CONSTRAINT fkDimSellers_geolocation_Key
GO
ALTER TABLE [dbo].[Fact_Order_Items] WITH CHECK ADD CONSTRAINT
[fkFact_Order_items_product_key] FOREIGN KEY([product_key])
REFERENCES [dbo].[Dim_Product] ([product_key])
GO
ALTER TABLE [dbo].[Fact_Order_Items] CHECK CONSTRAINT
fkFact_Order_items_product_key
GO
ALTER TABLE [dbo].[Fact_Order_Items] WITH CHECK ADD CONSTRAINT
[fkFact_Order_items_seller_key] FOREIGN KEY([seller_key])
REFERENCES [dbo].[Dim_Seller] ([seller_key])
GO
ALTER TABLE [dbo].[Fact_Order_Items] CHECK CONSTRAINT fkFact_Order_items_seller_key
GO
ALTER TABLE [dbo].[Fact_Order_Items] WITH CHECK ADD CONSTRAINT
[fkFact_Order_items_shipping_limit_date_key] FOREIGN KEY([shipping_limit_date_key])
REFERENCES [dbo].[Dim_Date] ([datekey])
GO
ALTER TABLE [dbo].[Fact_Order_Items] CHECK CONSTRAINT
fkFact_Order_items_shipping_limit_date_key
GO
ALTER TABLE [dbo].[Fact_Payment] WITH CHECK ADD CONSTRAINT
[fkFact_Order_payments_Fact_Orders] FOREIGN KEY([order_key])
REFERENCES [dbo].[Fact_Order] ([order_key])
GO
ALTER TABLE [dbo].[Fact_Payment] CHECK CONSTRAINT
fkFact_Order_payments_Fact_Orders
GO
ALTER TABLE [dbo].[Fact_Review] WITH CHECK ADD CONSTRAINT
[fkFact_Order_reviews_order_key] FOREIGN KEY([order_key])
REFERENCES [dbo].[Fact_Order] ([order_key])
GO
ALTER TABLE [dbo].[Fact_Review] CHECK CONSTRAINT fkFact_Order_reviews_order_key
GO
ALTER TABLE [dbo].[Fact_Review] WITH CHECK ADD CONSTRAINT
[fkFact_Review_reviewer_order_reviewer_timestamp_key] FOREIGN KEY([reviewer_timestamp_key])
REFERENCES [dbo].[Dim_Timestamp] ([timestamp])
GO

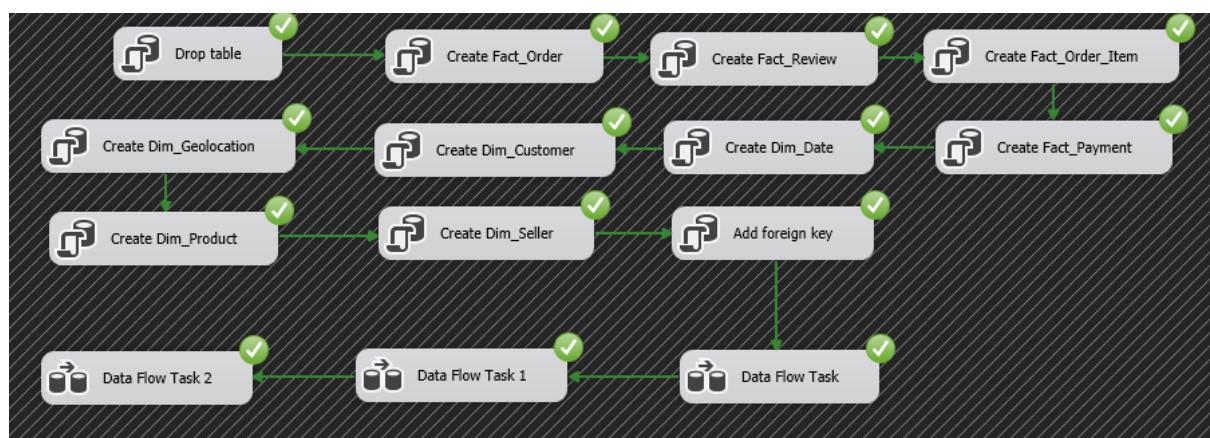
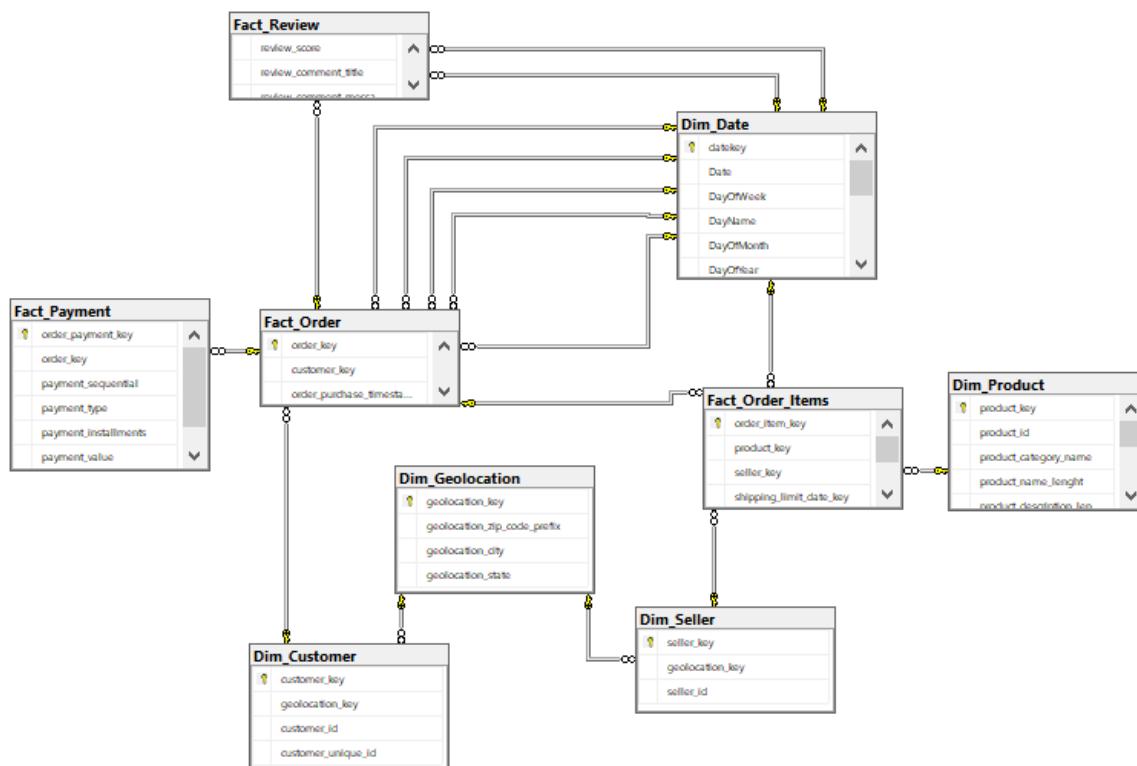
```

OK Cancel

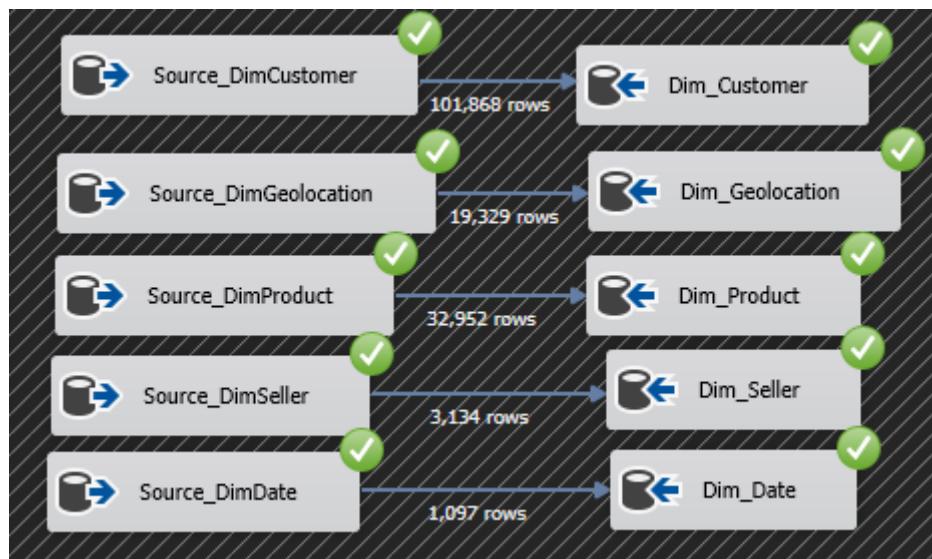
- Sau khi thêm các Data flow task và Execute SQL Task:



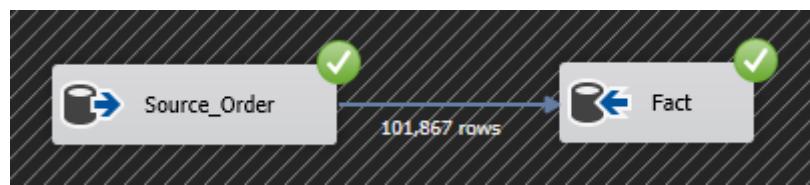
### 3.8.6. Chạy Project và xem xét kết quả thu được:



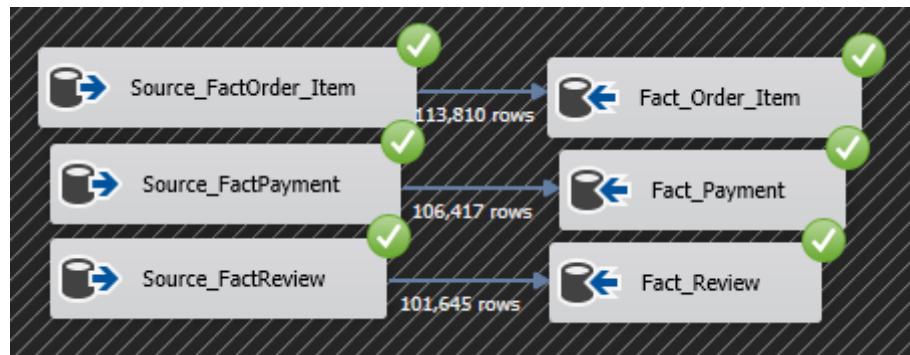
### 3.8.6.1. Data Flow Task:



### 3.8.6.2. Data Flow Task 1:



### 3.8.6.3. Data Flow Task 2:



## CHƯƠNG 4: PHÂN TÍCH DỮ LIỆU VỚI SSAS

### 4.1. Danh sách các câu truy vấn:

- Thống kê số điểm đánh giá (review\_score) trên các order.
- Thống kê doanh thu theo vùng.
- Thống kê doanh thu theo các tháng, năm.
- Thống kê các hình thức thanh toán mà khách hàng sử dụng.
- Thống kê sản phẩm bán chạy

### 4.2. Giới thiệu các phương pháp để sử dụng truy vấn:

#### 4.2.1. Sử dụng SSAS, Pivot table:

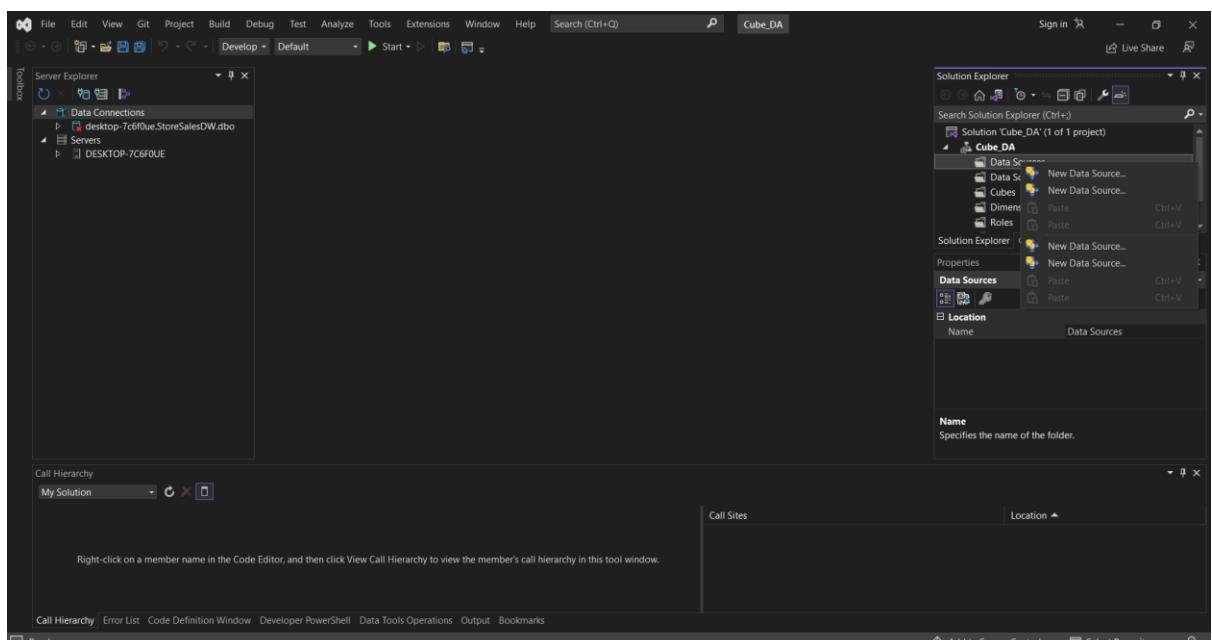
- Analysis Services là một công cụ khai thác dữ liệu và xử lý phân tích trực tuyến trong Microsoft SQL Server. SSAS được các tổ chức sử dụng như một công cụ để phân tích và hiểu ý nghĩa của thông tin có thể trải rộng trên nhiều cơ sở dữ liệu hoặc trong các bảng hoặc tệp khác nhau.

#### 4.2.2. Câu lệnh truy vấn SQL:

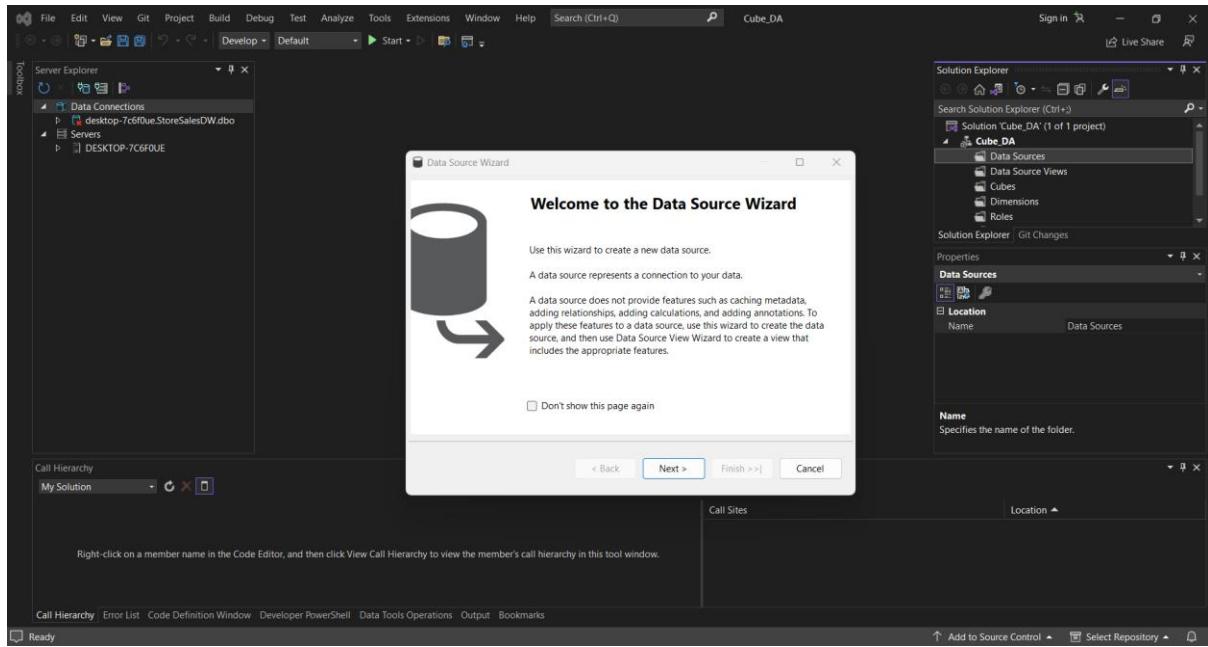
- Câu lệnh SQL, hoặc truy vấn SQL, là các lệnh hướng dẫn hợp lệ mà hệ thống quản lý cơ sở dữ liệu quan hệ hiểu được. Nhà phát triển phần mềm xây dựng các câu lệnh SQL bằng nhiều phần tử ngôn ngữ SQL khác nhau. Phần tử ngôn ngữ SQL là các thành phần như mã định danh, biến và điều kiện tìm kiếm tạo thành một câu lệnh SQL đúng.

### 4.3. Xây dựng mô hình SSAS và Pivot table:

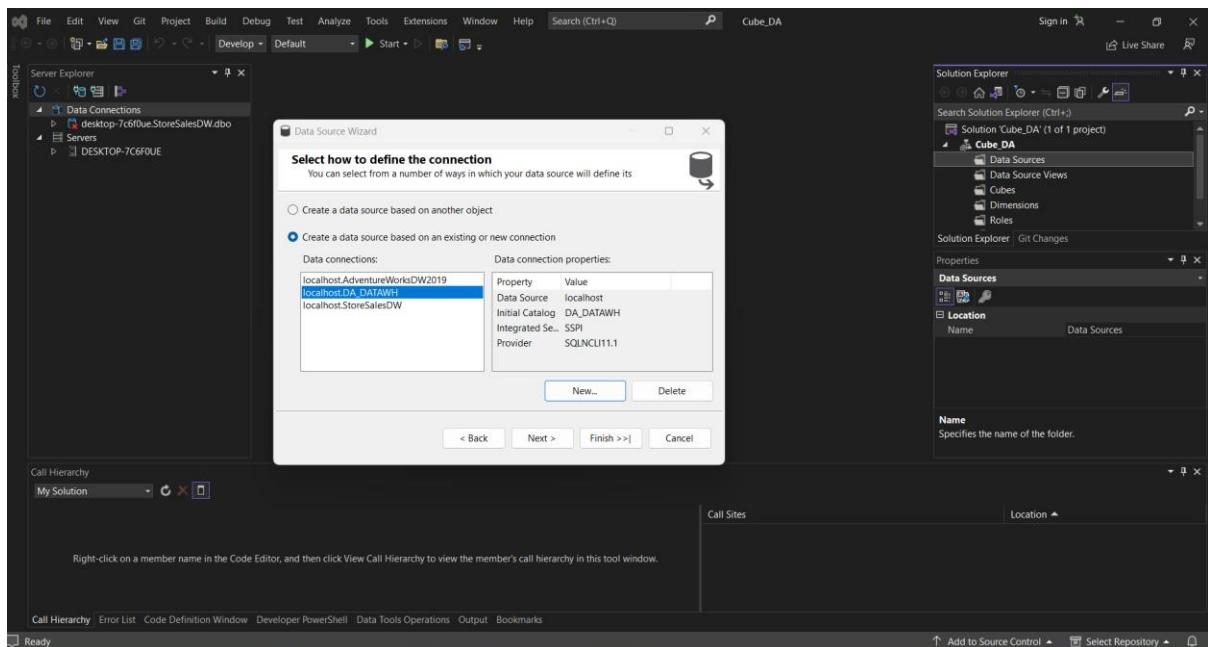
- Chọn New Data Source... để tạo Data Source mới:



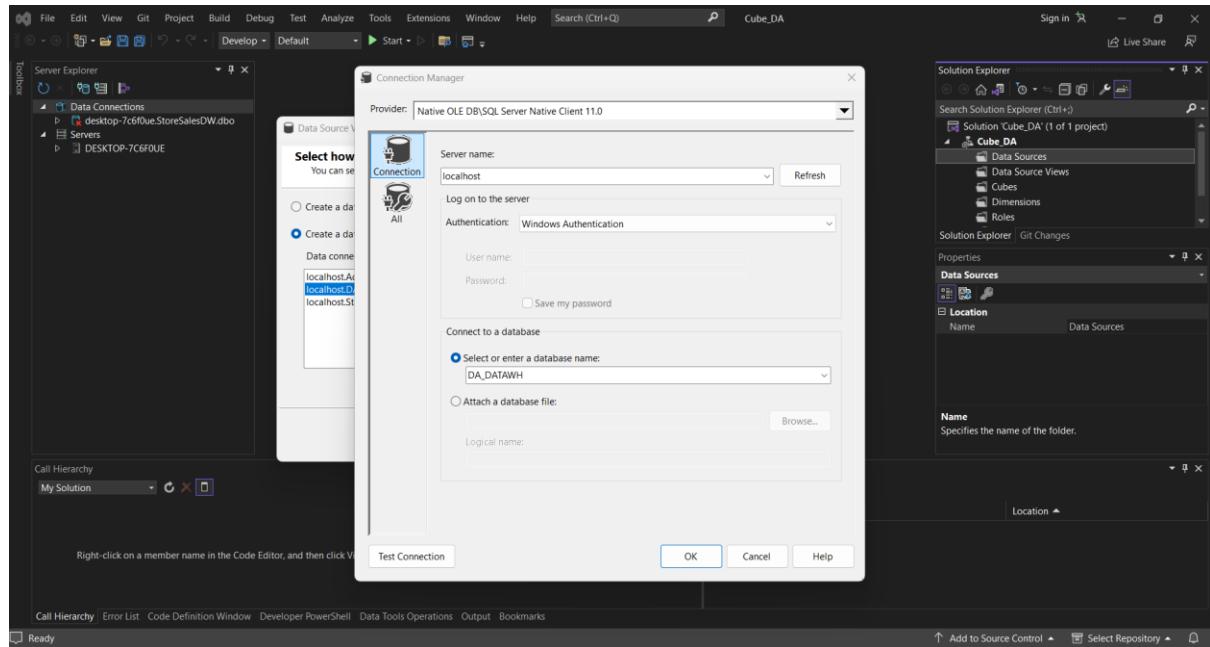
- Bấm Next



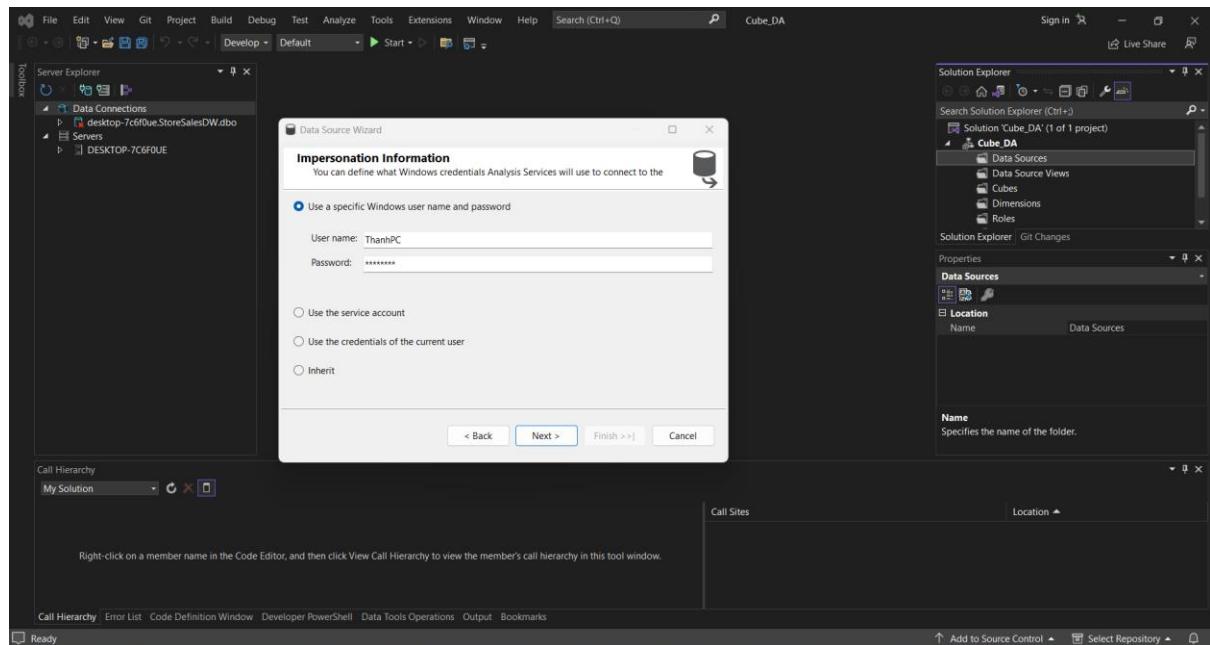
- Tích chọn **Create a data source based on an existing or new connection** và chọn **New...**



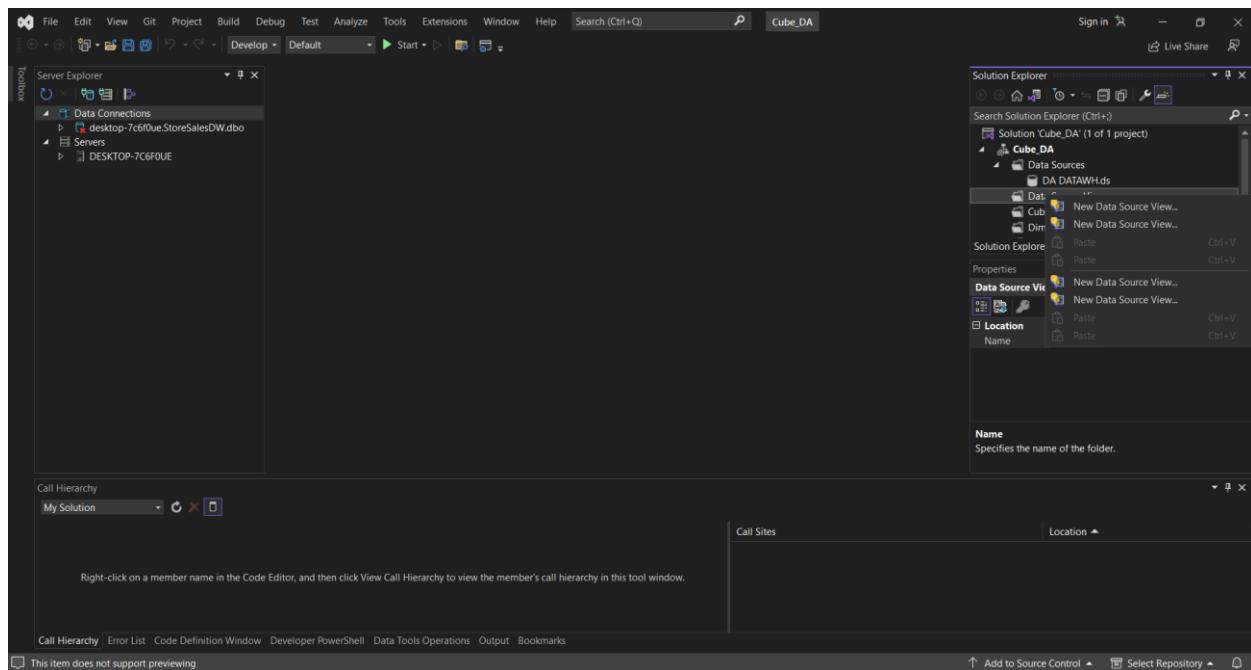
- Trong **Connection Manager** ở mục **Provider** chọn **Native OLE DB\SQL Server Native Client 11.0** với server name là **localhost** và **database**. Click **Test Connection** và click **OK**.



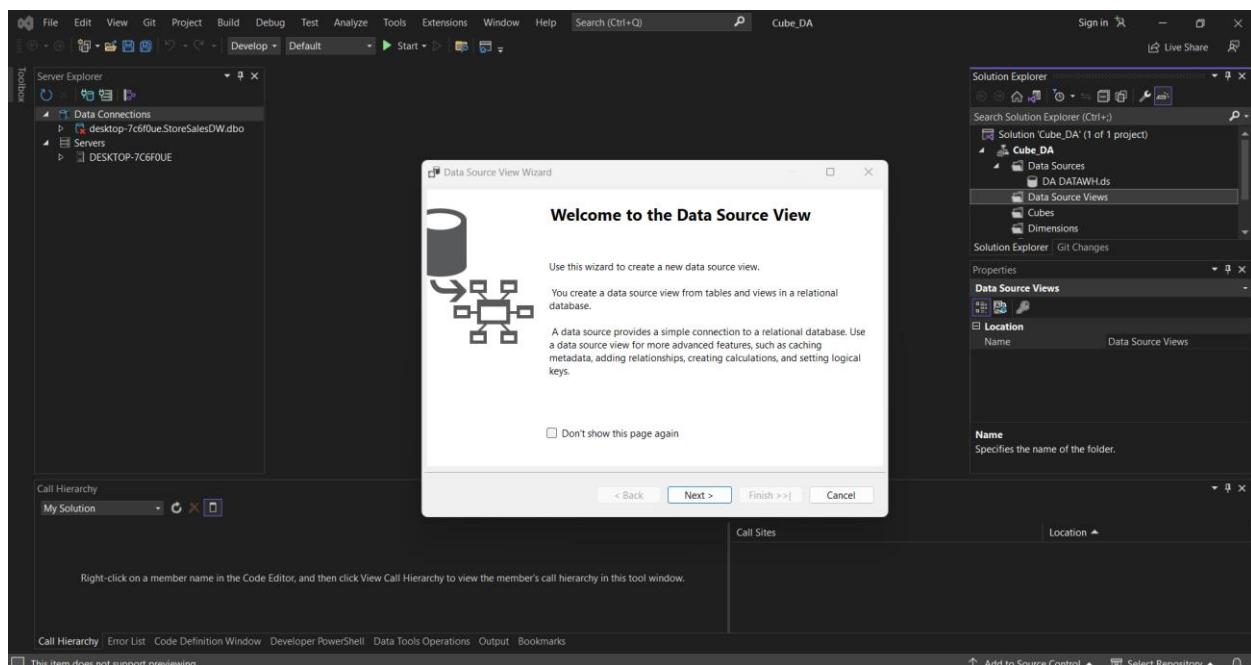
- **Impersonation Information** chọn **Use a specific Windows username and password**, nhập **username** và **password**, click **Next**



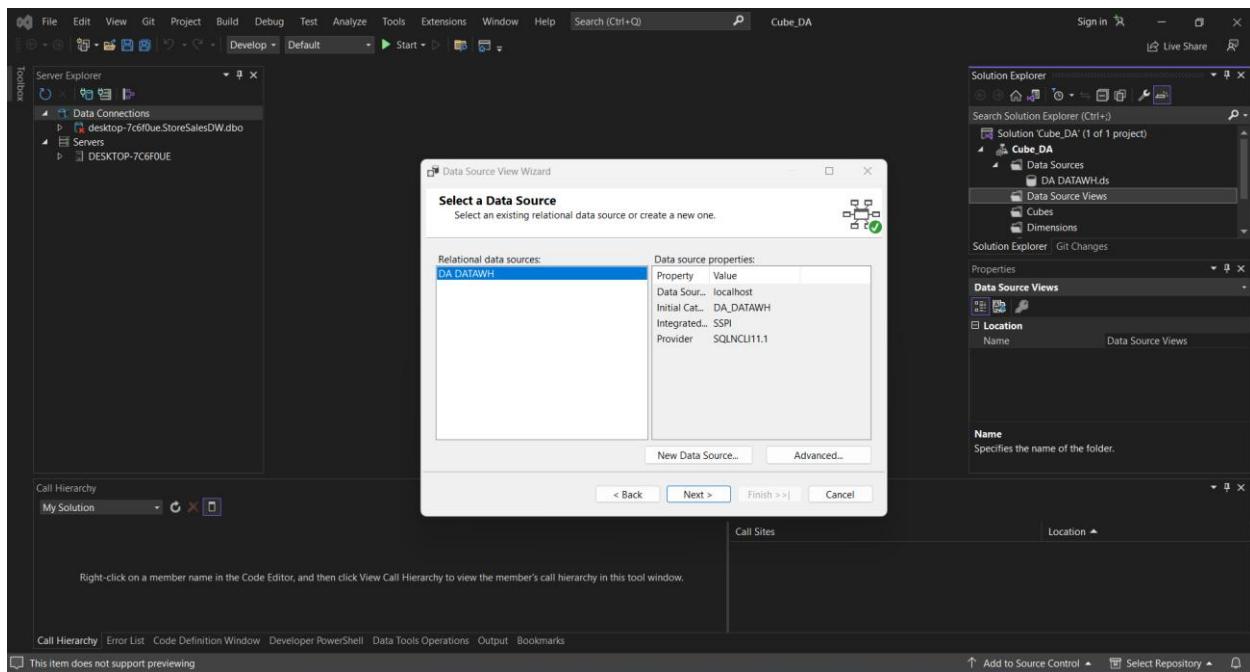
- Đặt tên **Data Source Name** và click **Finish**
- Tạo **Data Source View**:
  - o Chọn **Data Source Views** và chọn **New Data Source View...**



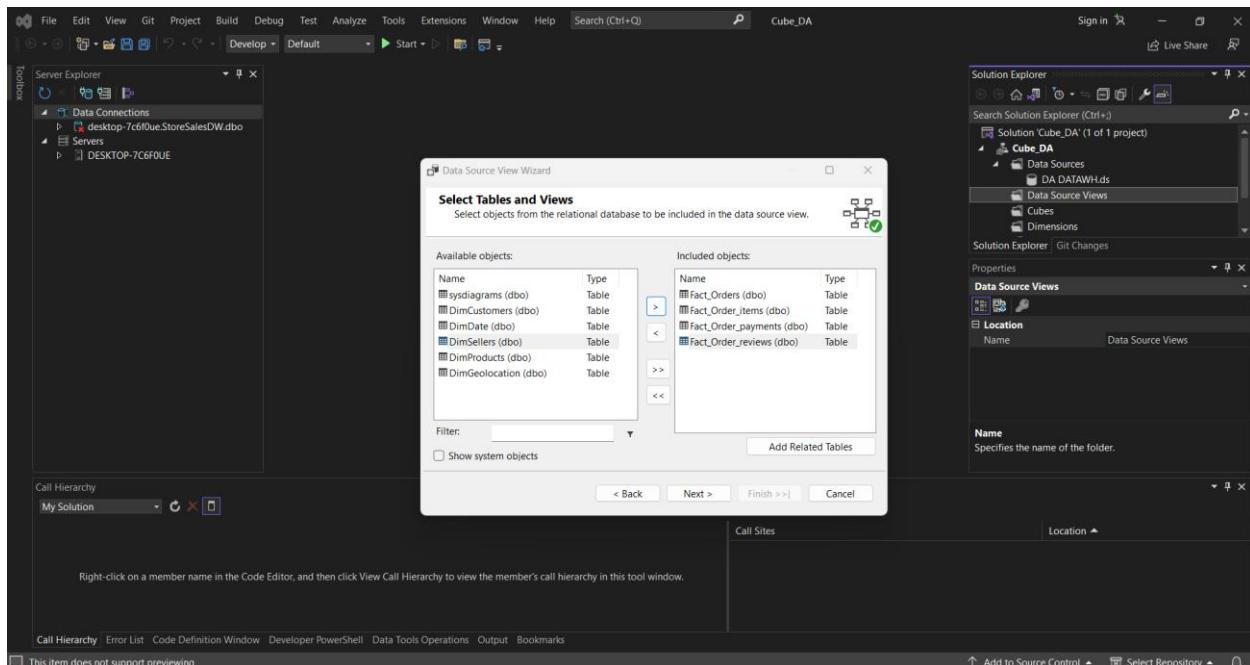
- Bấm Next



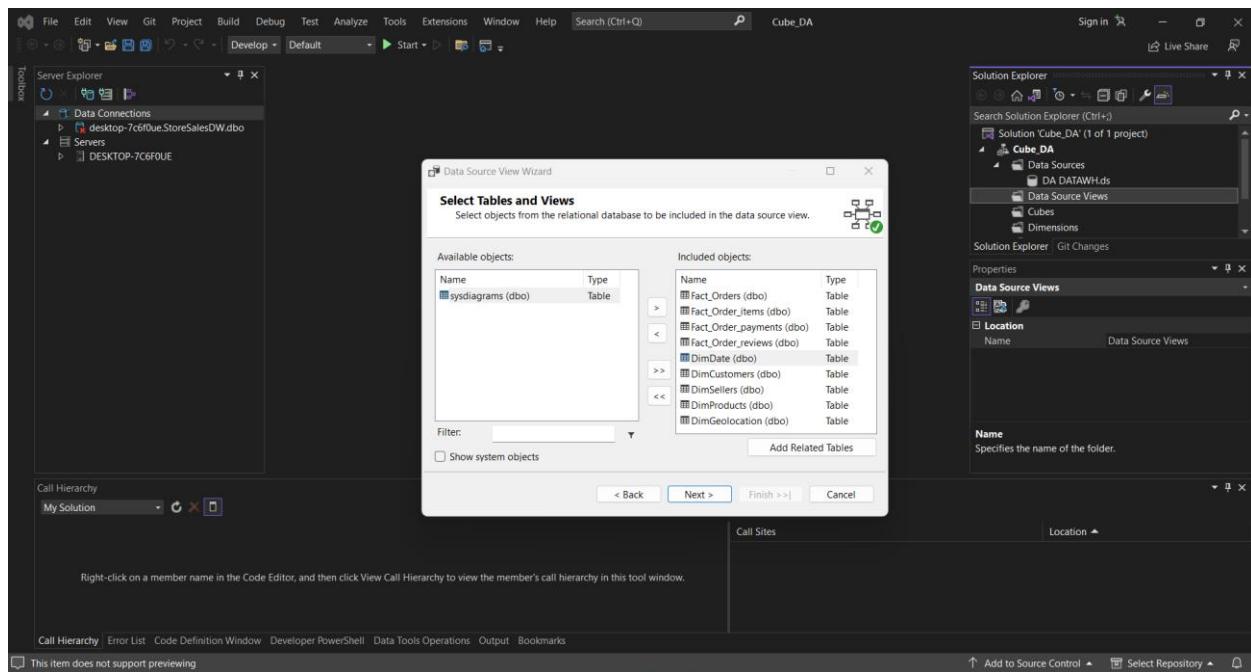
- Chọn Data Source vừa tạo



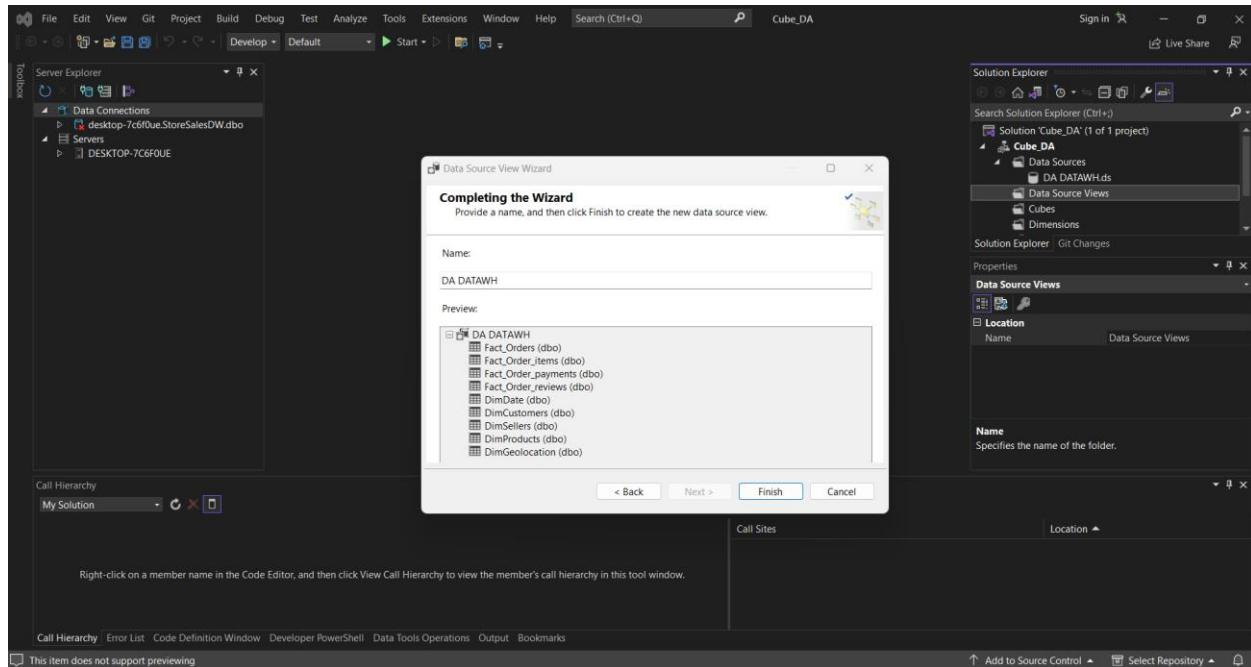
- Thêm các Fact Table vào



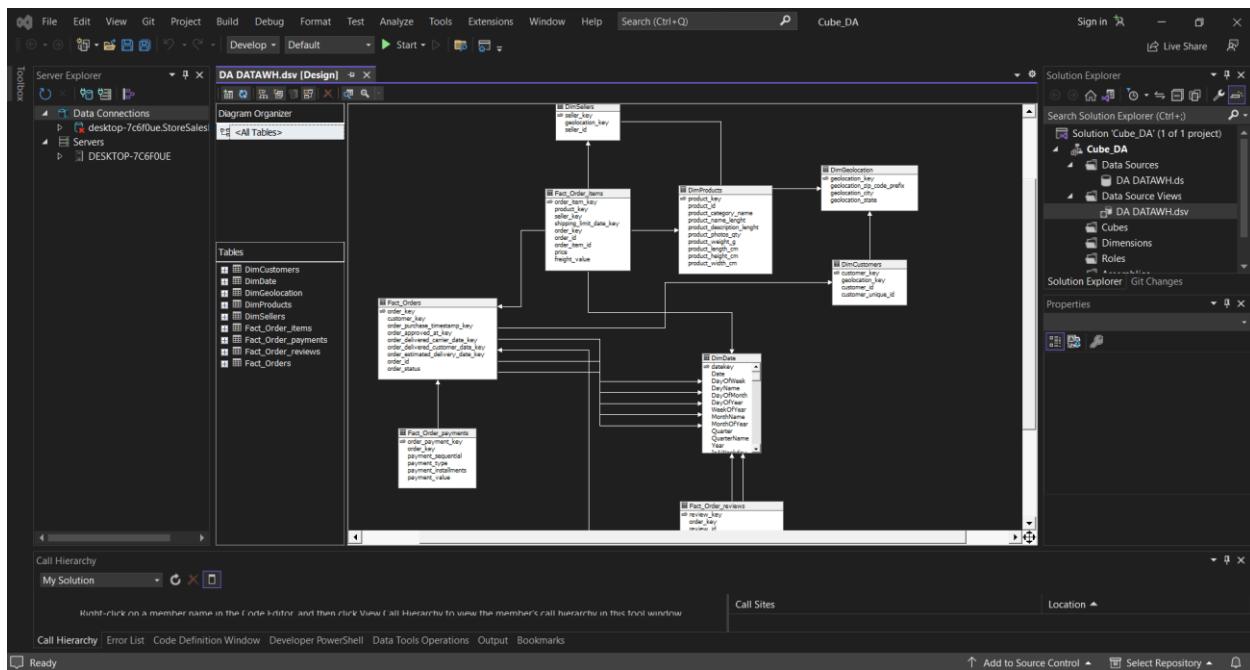
- Sau đó bấm Add Related Tables để thêm tất cả bảng Dim vào



- Đặt tên cho Data Source View

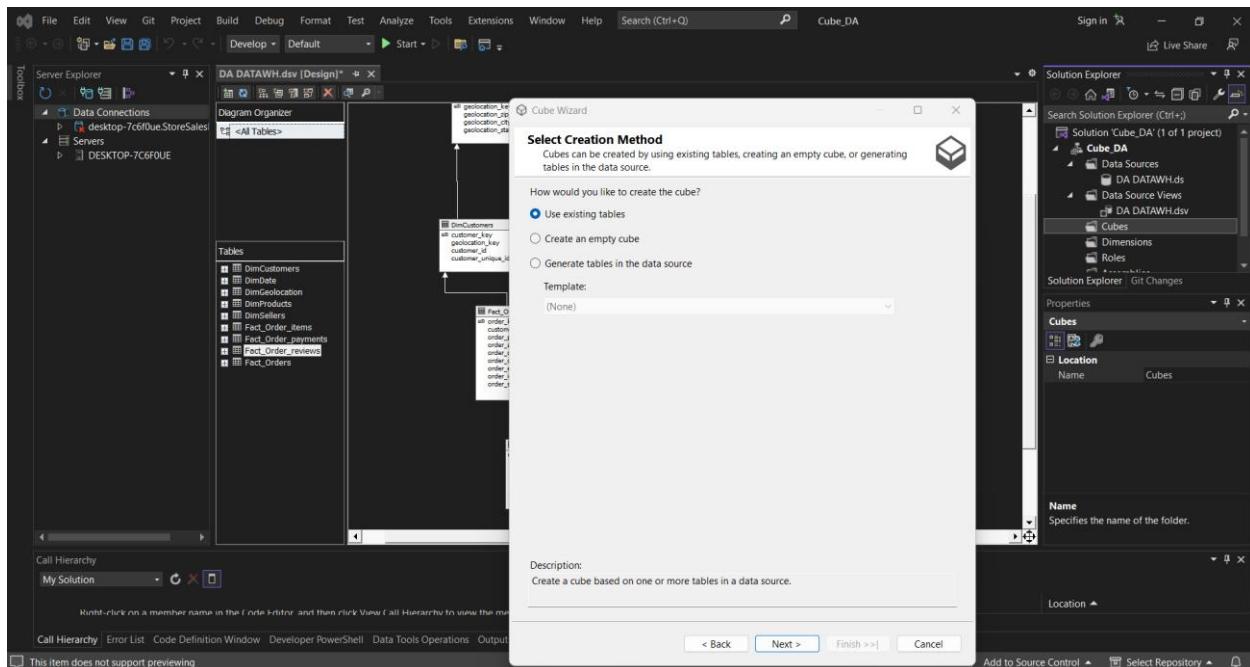


- Bấm Finish, click vào data source view vừa tạo để xem lược đồ của các tables

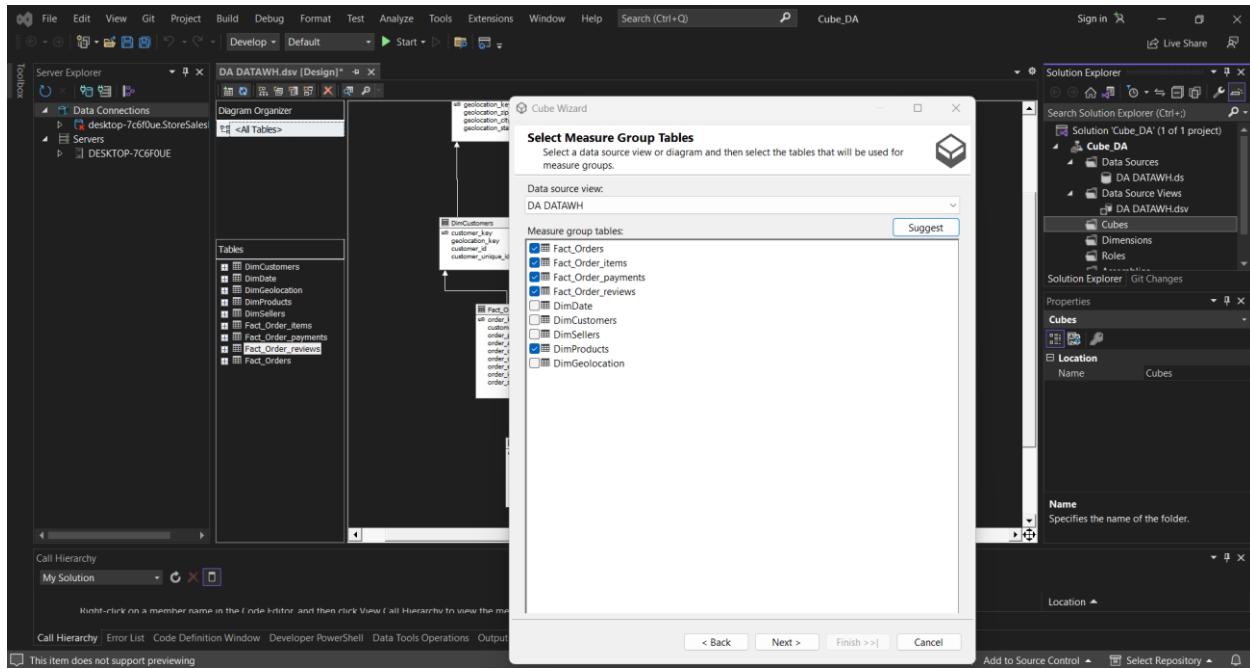


#### 4.4. Quá trình xây dựng khối:

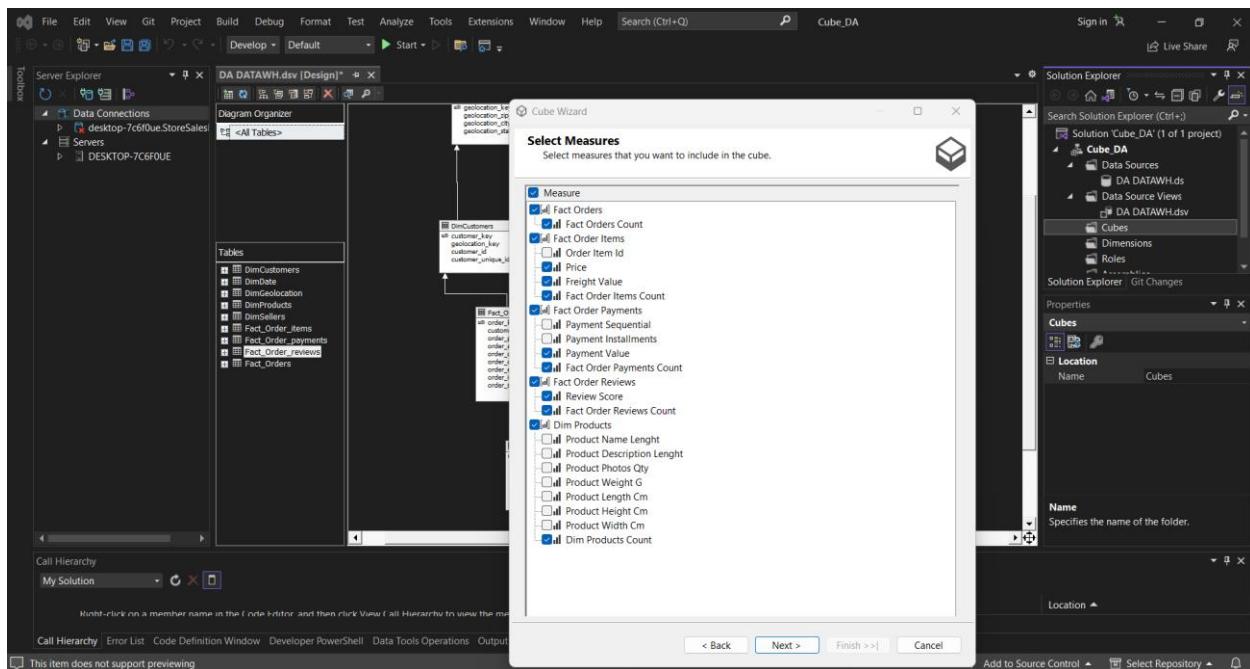
- Chọn vào Cubes, chọn New Cube



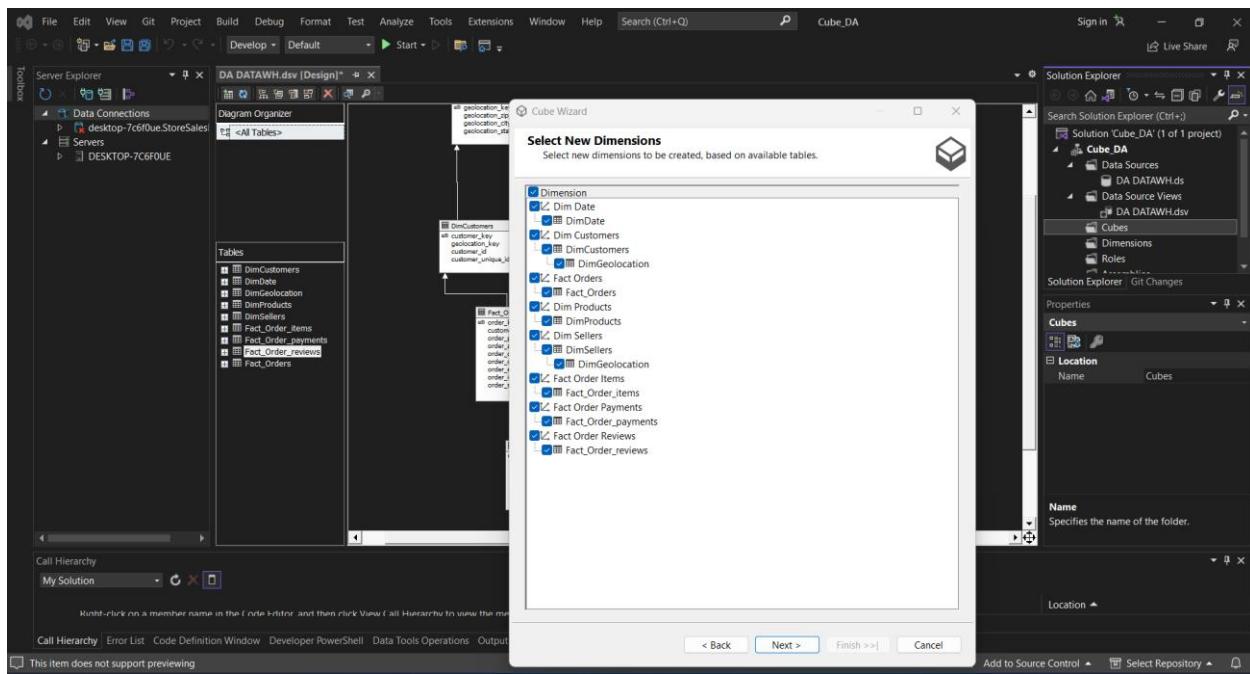
- Bấm Suggest để chọn các Measure Group Tables



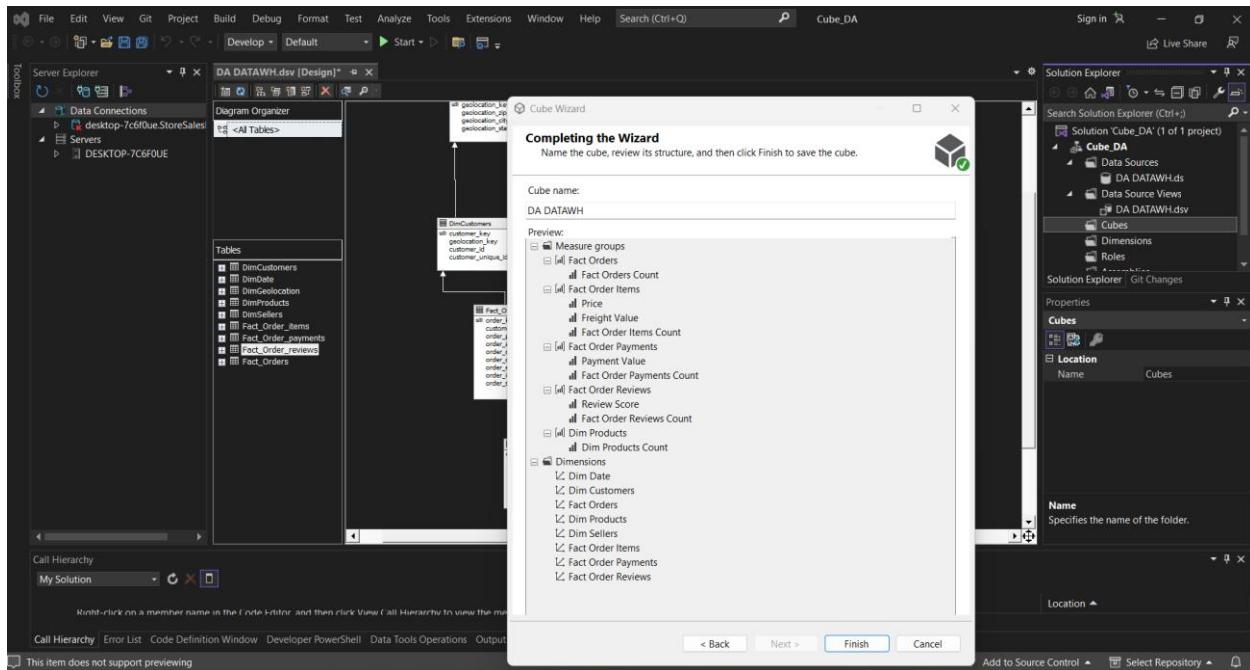
- Tích chọn các measure cần loại bỏ các Measure không cần thiết



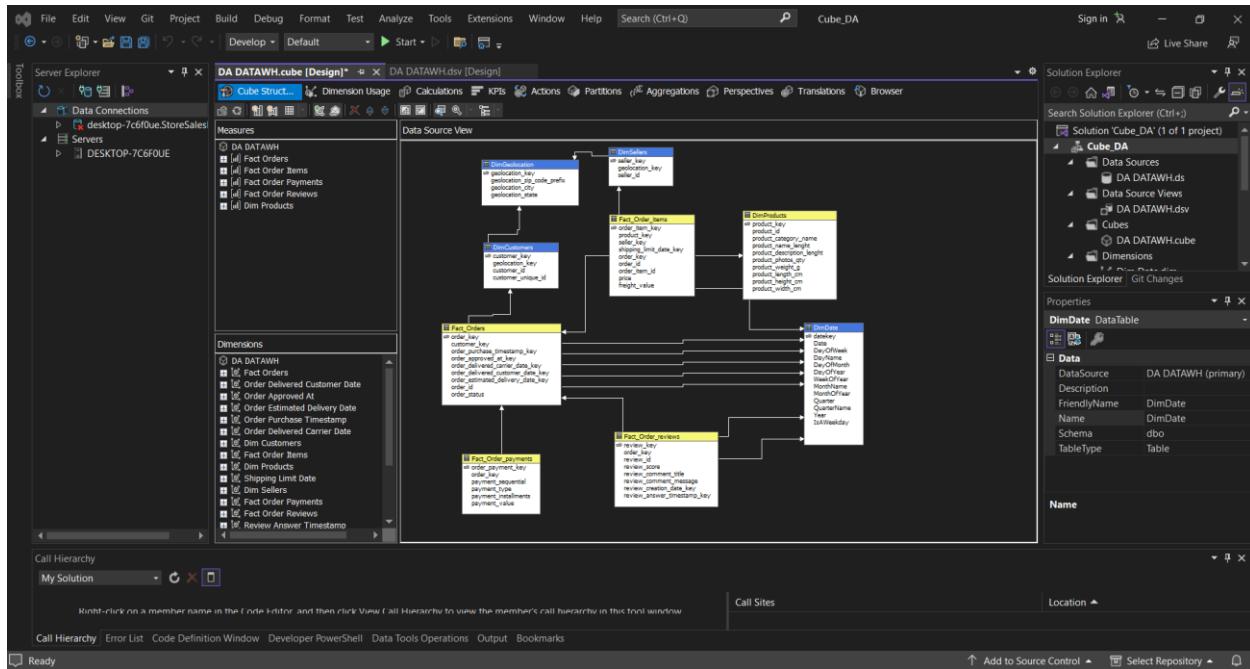
- Chọn các Dimension



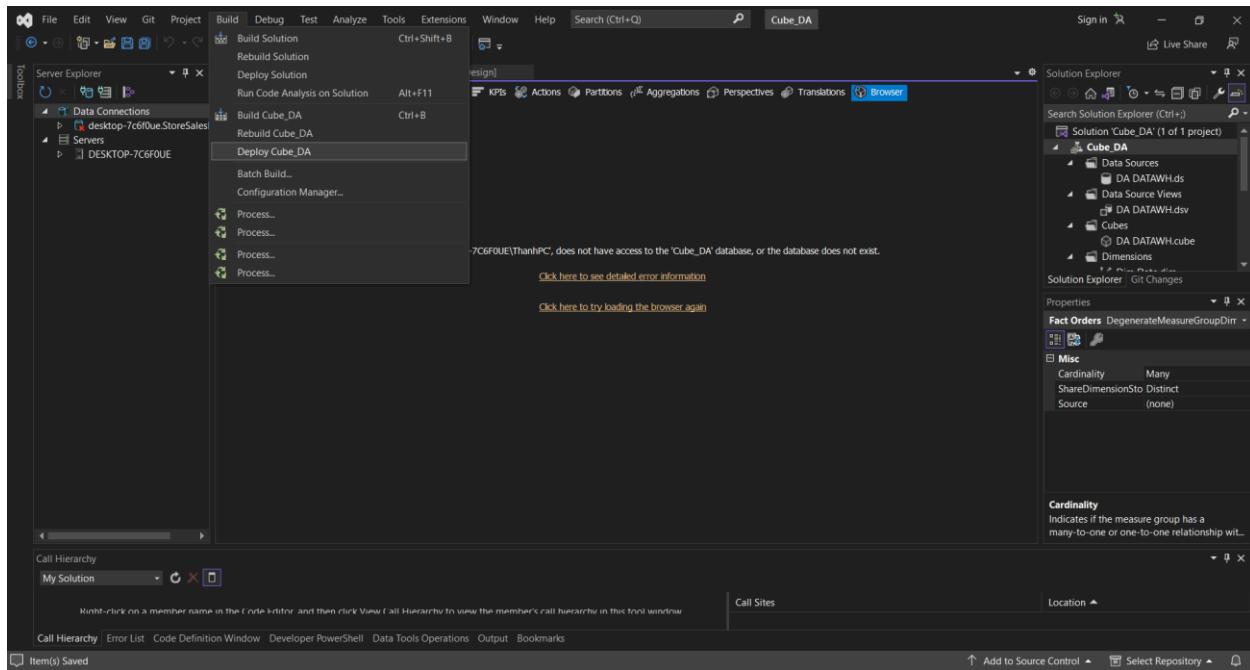
- Đặt tên Cube và click Finish



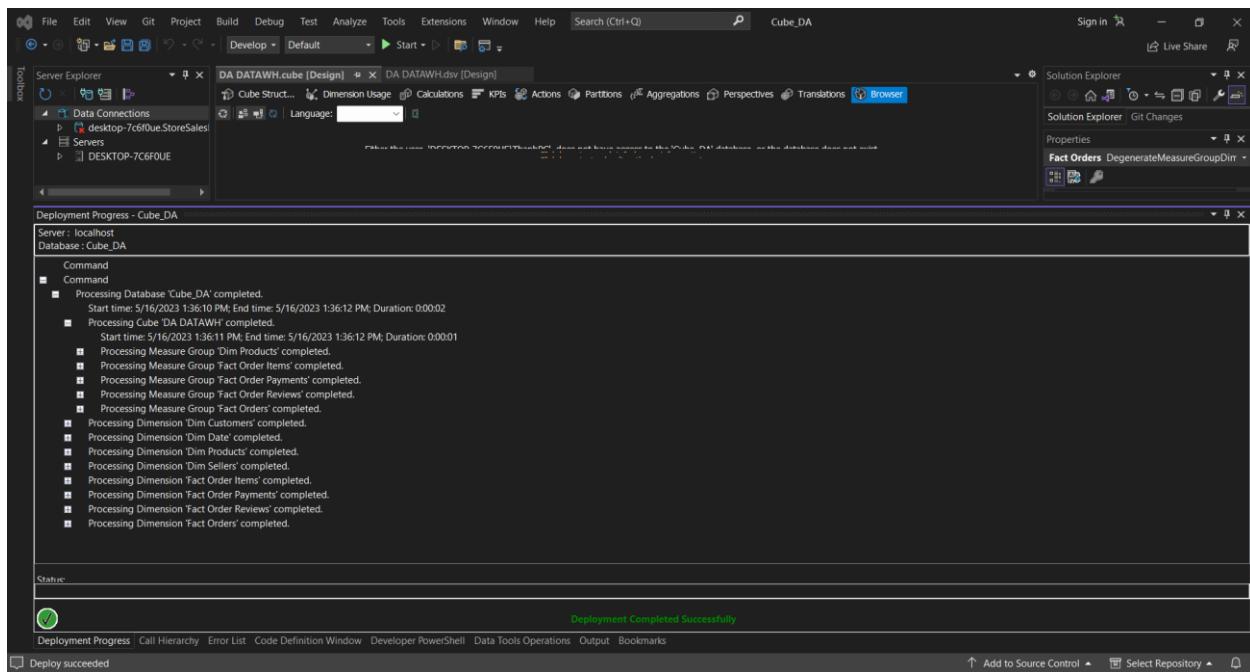
- Cube Structure vừa tạo



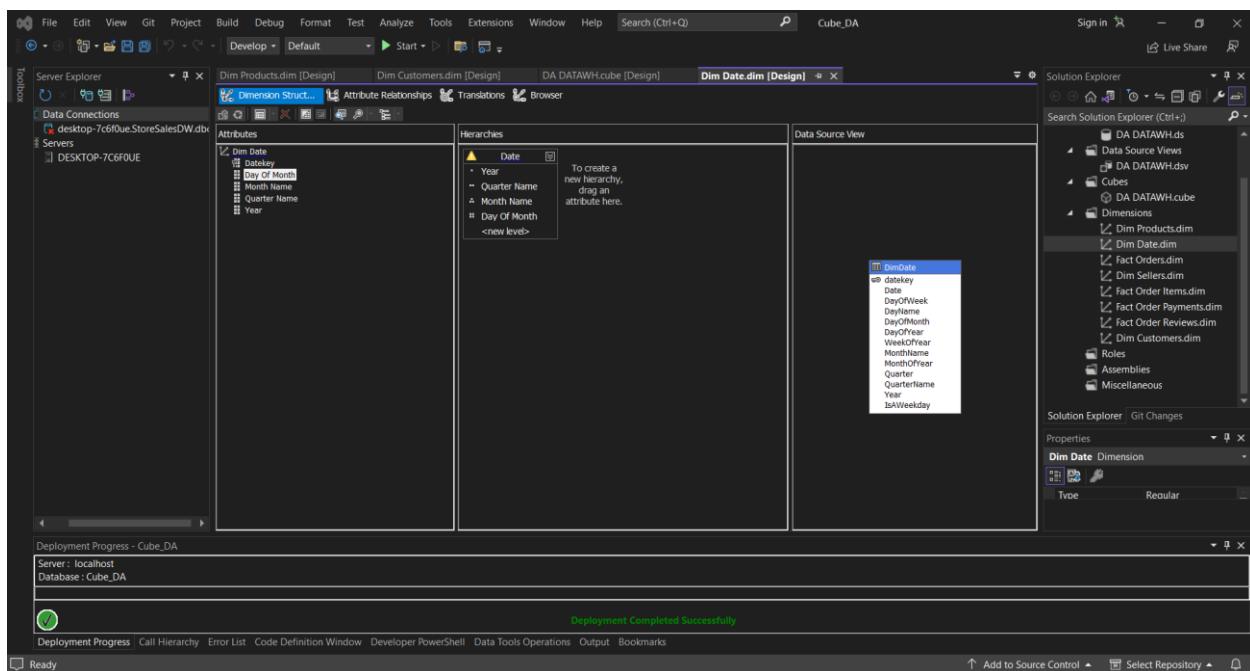
### - Bấm Deploy lên Analysis Service Server



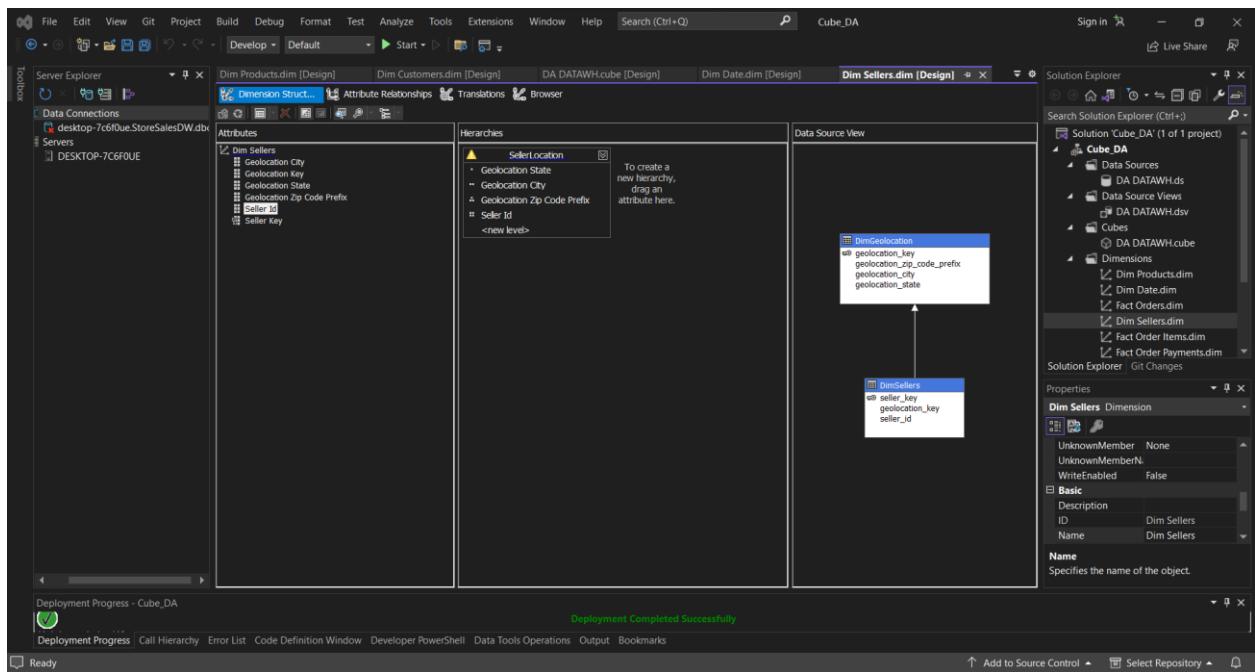
### - Sau khi Deploy xong xem hiển thị thông báo, kiểm tra nếu báo lỗi



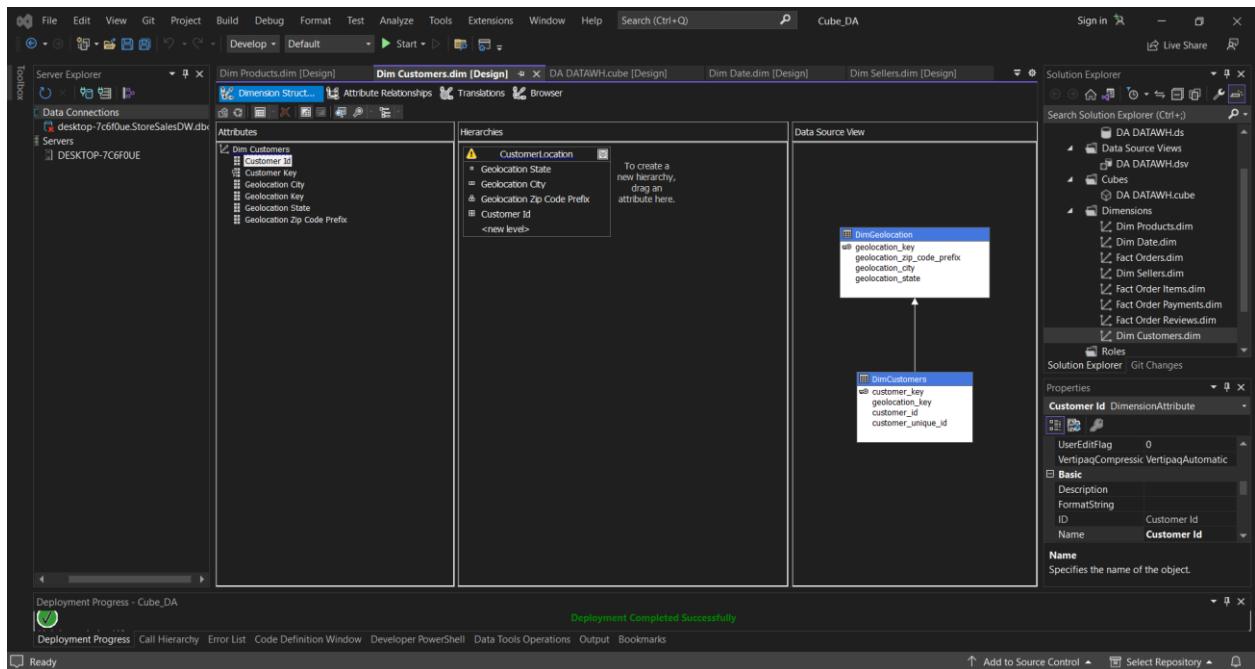
### - Tạo Phân cấp cho DimDate



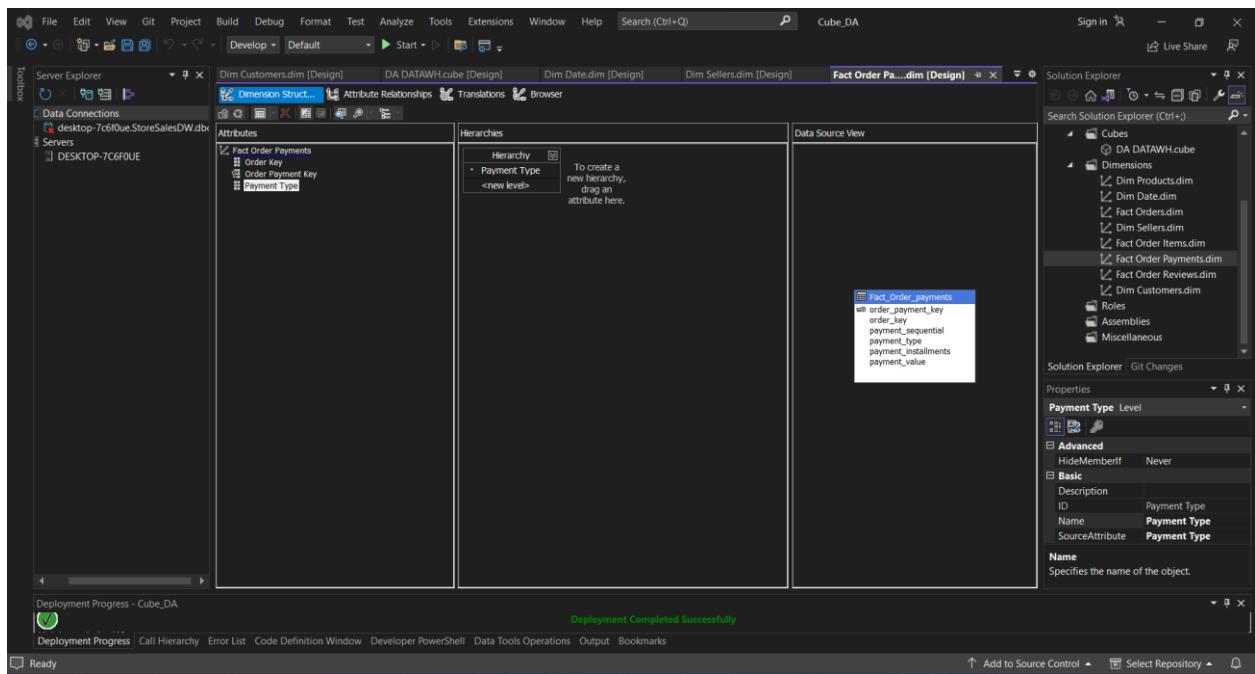
### - Tạo phân cấp cho Seller



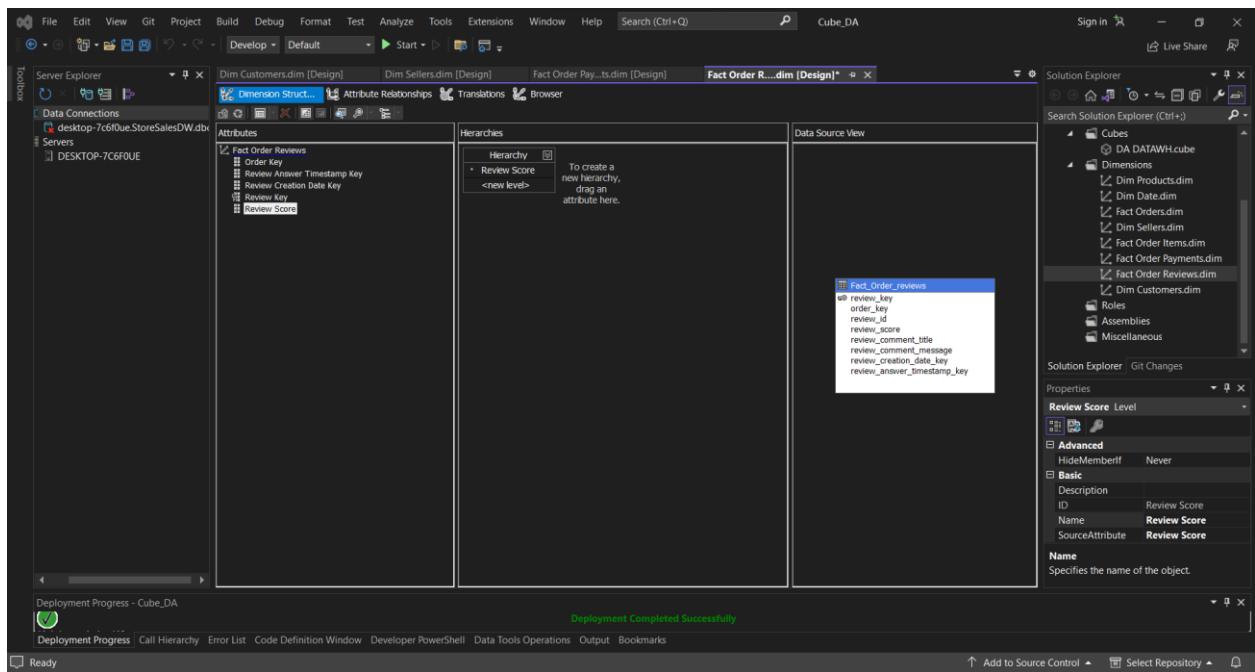
- Tạo phân cấp cho Customer



- Tạo phân cấp cho Payment Value



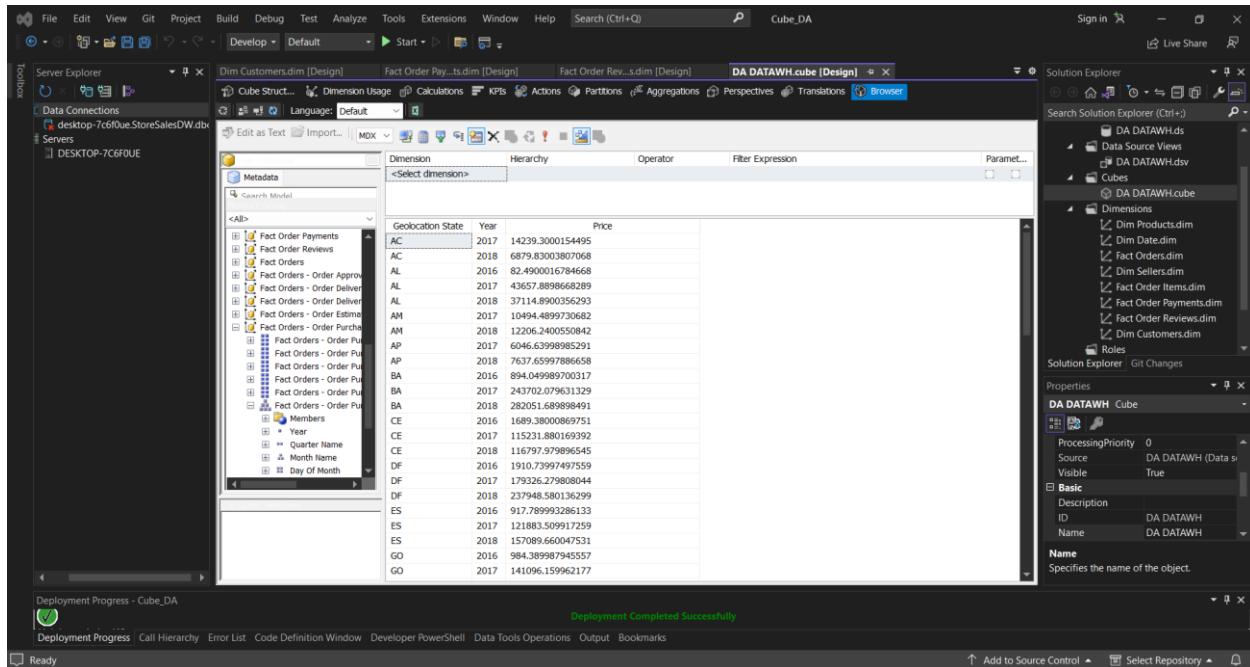
- Phân cấp cho Review Score



## 4.5. Thực hiện các câu truy vấn:

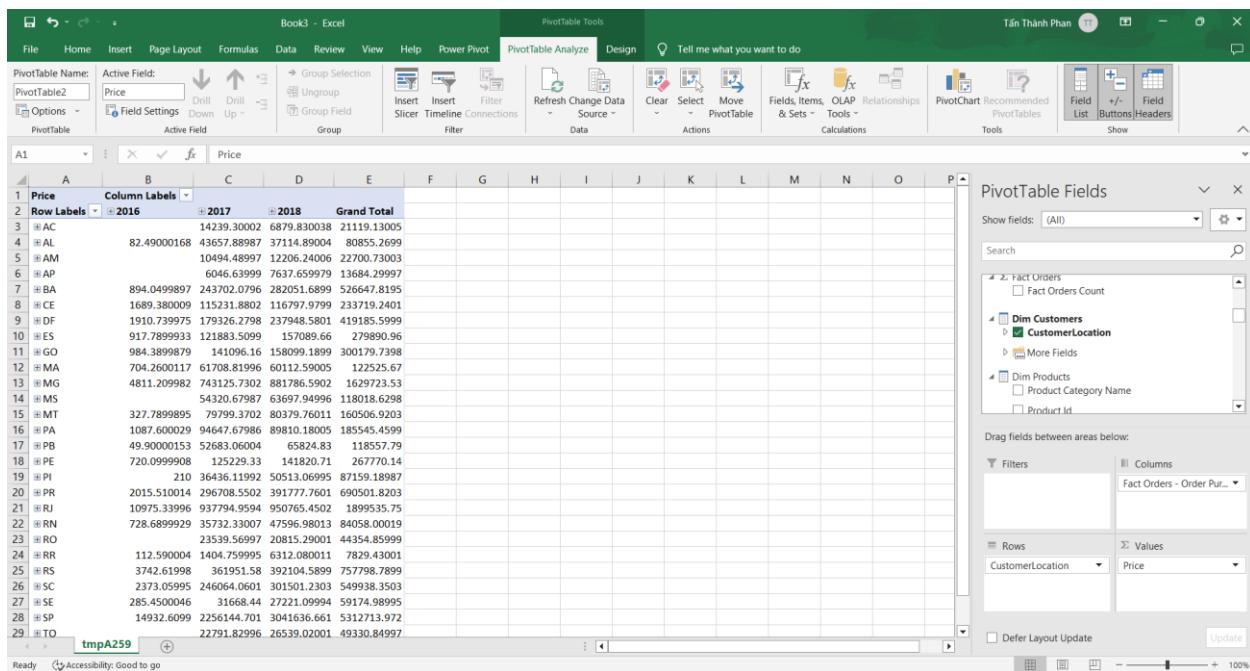
### 4.5.1. Thống kê doanh thu theo khu vực:

- Sử dụng công cụ SSAS:



The screenshot shows the Microsoft Analysis Services (SSMS) interface. The main area displays the 'DA DATAWH.cube [Design]' tab, which contains a grid of data with columns for 'Geolocation State', 'Year', and 'Price'. The sidebar on the left shows the 'Server Explorer' and 'Solution Explorer' panes. The 'Solution Explorer' pane lists the project structure, including 'DA DATAWH.ds' and 'DA DATAWH.cube' under 'Cubes'. The 'Properties' pane on the right shows the cube's properties: 'ProcessingPriority' is set to 0, 'Source' is 'DA DATAWH (Data Source)', and 'Visible' is set to True. The status bar at the bottom indicates a successful deployment.

- Sử dụng Pivot Table trong Excel:



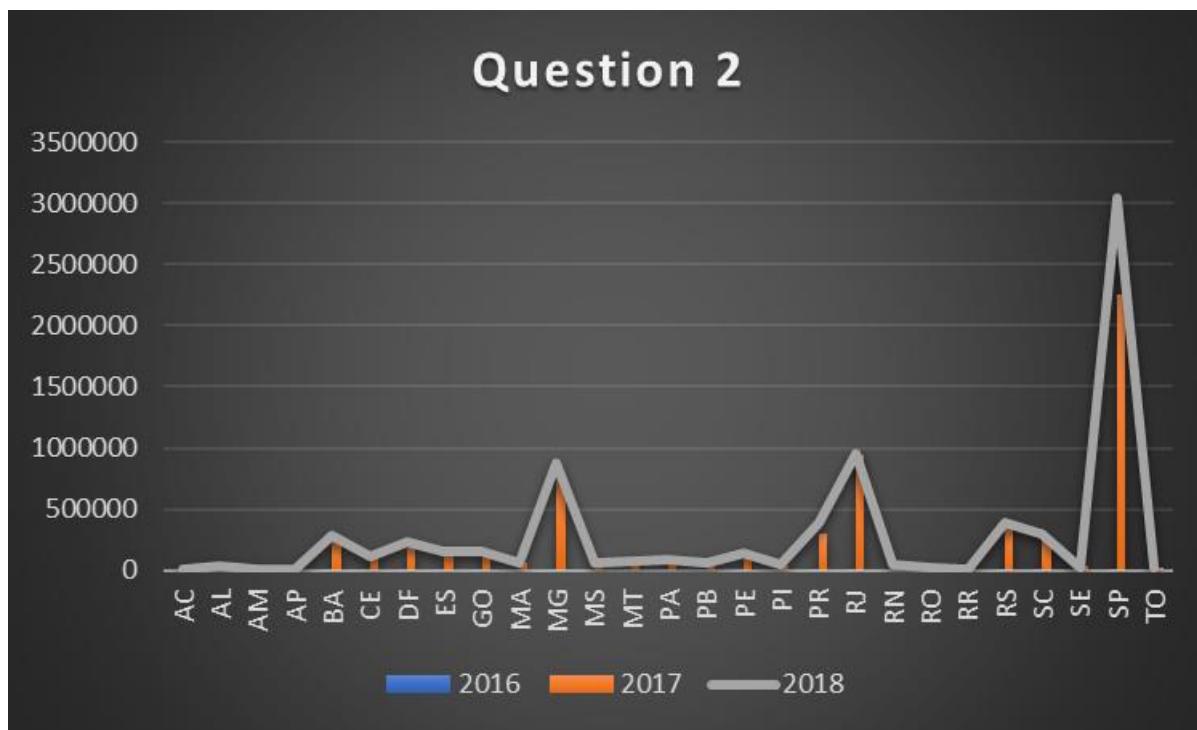
The screenshot shows a Microsoft Excel spreadsheet titled 'Book3 - Excel'. A PivotTable is displayed in the range A1:P29, with the formula bar showing 'tmpA259'. The PivotTable has 'Price' as the column label and '2016' and '2018' as row labels. The 'Grand Total' cell contains the value 21119.13005. The 'PivotTable Tools' ribbon is open, showing various options for data grouping, filtering, and calculations. The 'PivotTable Fields' pane on the right lists fields from the 'Dim Customers' and 'Fact Orders' dimensions, such as 'CustomerLocation' and 'Fact Orders - Order Purchases'. The status bar at the bottom indicates 'Ready'.

- Sử dụng ngôn ngữ truy vấn SQL:

```
select a.geolocation_state, sum(p.price) as Total_Price,d.Year
from [Dim_Geolocation] as a
join [Dim_Customer] as c on a.geolocation_key = c.geolocation_key
join [Fact_Order] as o on o.customer_key = c.customer_key
join [Fact_Order_Items] as p on o.order_key = p.order_key
join [Dim_Date] as d on d.datekey = o.order_purchase_timestamp_key
group by a.geolocation_state, d.Year
order by a.geolocation_state asc, d.Year asc
```

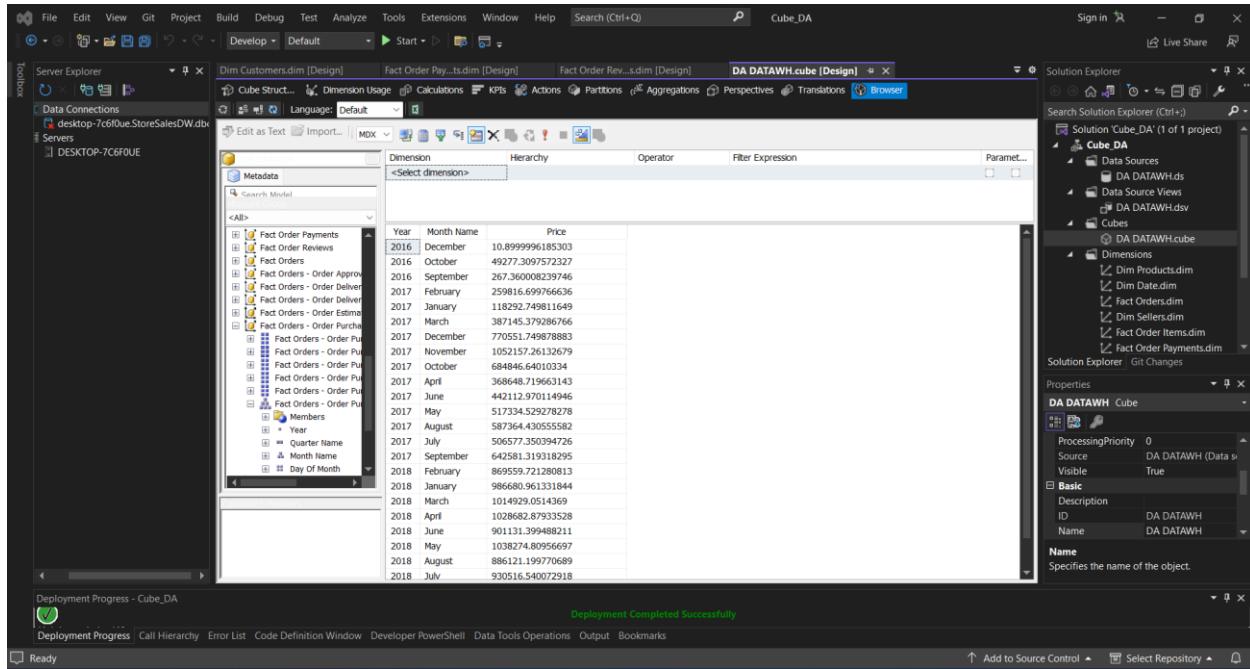
	geolocation_state	Total_Price	Year
1	AC	14239.3000154495	2017
2	AC	6879.83003807068	2018
3	AL	82.4900016784668	2016
4	AL	43657.8898668289	2017
5	AL	37114.8900356293	2018
6	AM	10494.4899730682	2017
7	AM	12206.2400550842	2018
8	AP	6046.63998985291	2017
9	AP	7637.65997886658	2018
10	BA	894.049989700317	2016
11	BA	243702.079631329	2017
12	BA	282051.689898491	2018
13	CE	1689.38000869751	2016
14	CE	115231.880169392	2017
15	CE	116797.979896545	2018
16	DF	1910.73997497559	2016
17	DF	179326.279808044	2017
18	DF	237948.580136299	2018
19	ES	917.789993286133	2016
20	ES	121883.509917259	2017
21	ES	157089.660047531	2018
22	GO	984.389987945557	2016
23	GO	141096.159962177	2017
24	GO	158099.189874649	2018
25	MA	704.260011672974	2016

- Trực quan hóa trên biểu đồ:



### 4.5.2. Thống kê doanh thu theo ngày, tháng:

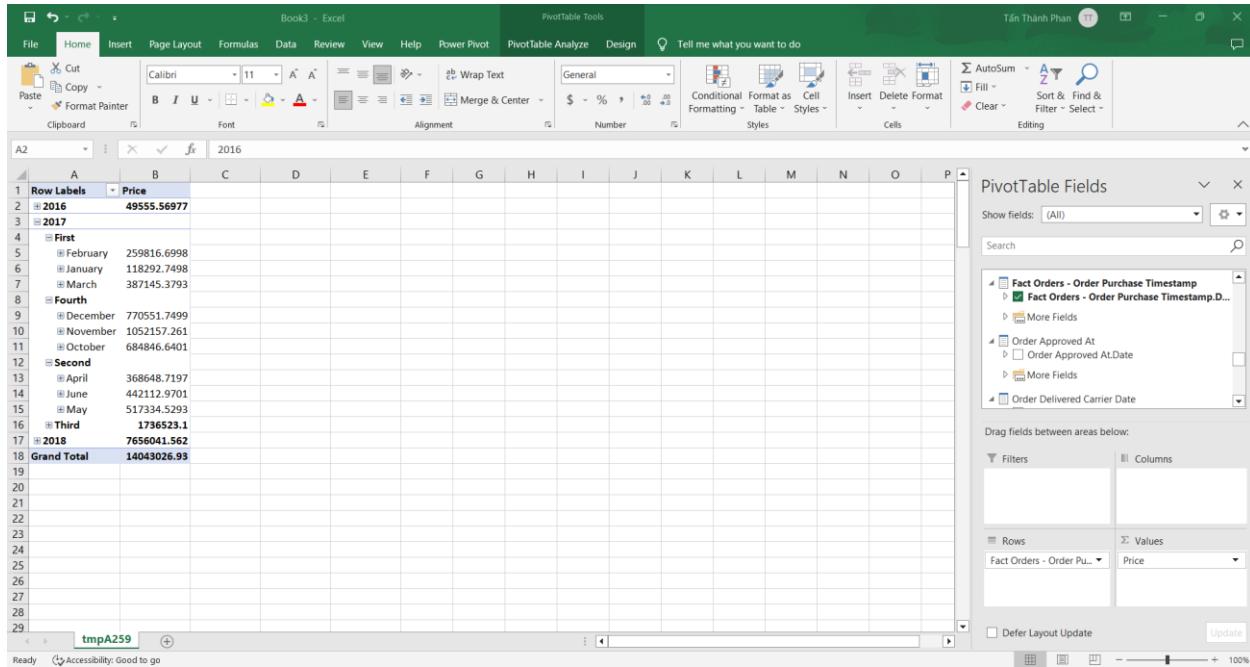
- Sử dụng công cụ SSAS:



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the left sidebar, the 'Server Explorer' shows a connection to 'DESKTOP-7C6F0UE'. In the center, the 'DA DATAWH.cube [Design]' tab is active, displaying a table with columns 'Year', 'Month Name', and 'Price'. The data is as follows:

Year	Month Name	Price
2016	December	10.8999996185303
2016	October	49277.3097572327
2016	September	267.360008239746
2017	February	259816.69976663
2017	January	118292.749811649
2017	March	387145.379286766
2017	December	770551.749878883
2017	November	105215.26132679
2017	October	684846.64010334
2017	April	368648.719663143
2017	June	442112.97011494
2017	May	517334.529278278
2017	August	587364.430555582
2017	July	506577.350394726
2017	September	642581.219318295
2018	February	869559.721280813
2018	January	986680.961331844
2018	March	1014929.0514369
2018	April	1028682.8793352
2018	June	901131.399488211
2018	May	1038274.80956697
2018	August	886121.199770689
2018	July	930516.540072918

- Sử dụng Pivot Table trong Excel:



The screenshot shows Microsoft Excel with a PivotTable set up. The PivotTable Fields pane on the right lists fields such as 'Fact Orders - Order Purchase Timestamp', 'Order Approved At', 'Order Delivered Carrier Date', etc. The main table area shows sales data grouped by year and month, with a total for each year and a grand total at the bottom.

Row Labels	Price
2016	49555.56977
2017	
First	
February	259816.6998
January	118292.7498
March	387145.3793
Fourth	
December	770551.7499
November	105215.261
October	684846.6401
Second	
April	368648.7197
June	442112.9701
May	517334.5293
Third	1736523.1
2018	7656041.562
Grand Total	14043026.93
Grand Total	14043026.93

- Sử dụng ngôn ngữ truy vấn SQL:

```

select sum(a.price) as Total_price, d.MonthOfYear,d.Year
  from [Fact_Order_Items] as a
  join [Fact_Order] as o on a.order_id=o.order_id
  join [Dim_Date] as d on d.datekey =
o.order_purchase_timestamp_key
 group by d.Year, d.MonthOfYear
order by d.Year asc, d.MonthOfYear asc
    
```

	Total_price	MonthOfYear	Year
1	267.360008239746	9	2016
2	51646.8297157288	10	2016
3	10.8999996185303	12	2016
4	124086.929823399	1	2017
5	282054.159647465	2	2017
6	413757.998847008	3	2017
7	386307.879655361	4	2017
8	541526.609226704	5	2017
9	465622.770098925	6	2017
10	530923.710464001	7	2017
11	612811.270637751	8	2017
12	686508.139190197	9	2017
13	726705.060177803	10	2017
14	1129790.5415144	11	2017
15	823572.330052376	12	2017
16	1047791.92139196	1	2018
17	909228.861192465	2	2018
18	1068580.79160261	3	2018
19	1075807.87927997	4	2018
20	1109436.38946581	5	2018
21	961556.279514074	6	2018
22	986283.300132275	7	2018
23	935201.159920454	8	2018
24	145	9	2018

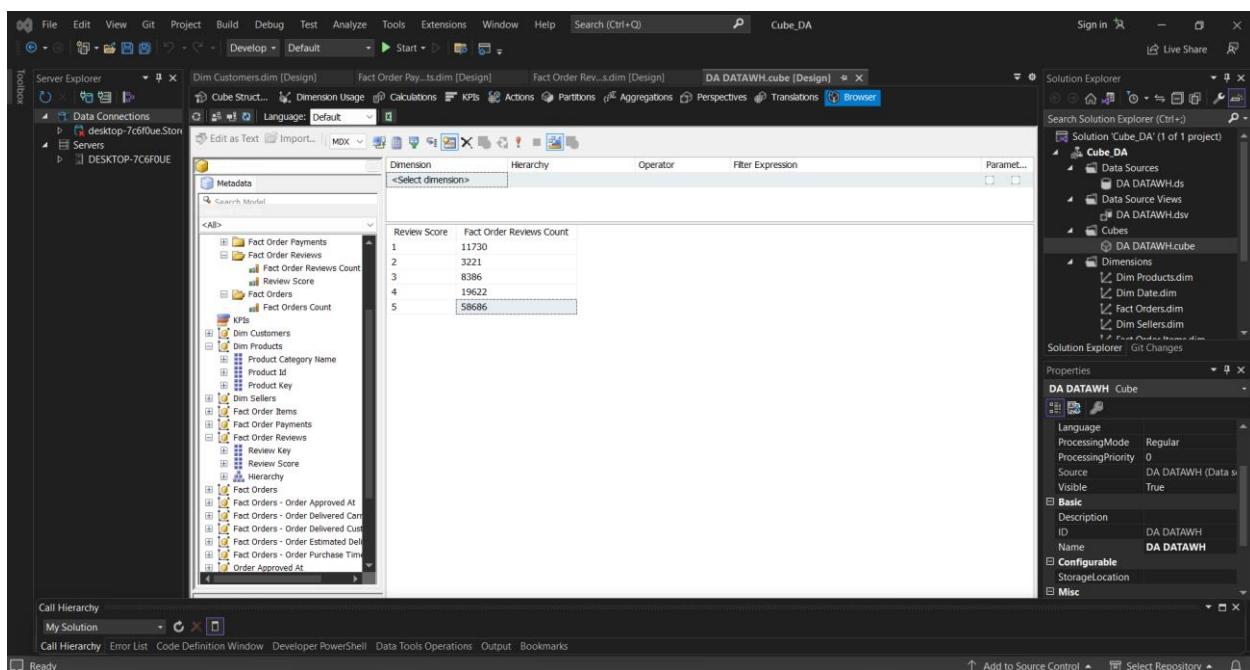
- Trực quan trên biểu đồ:

### Question 3

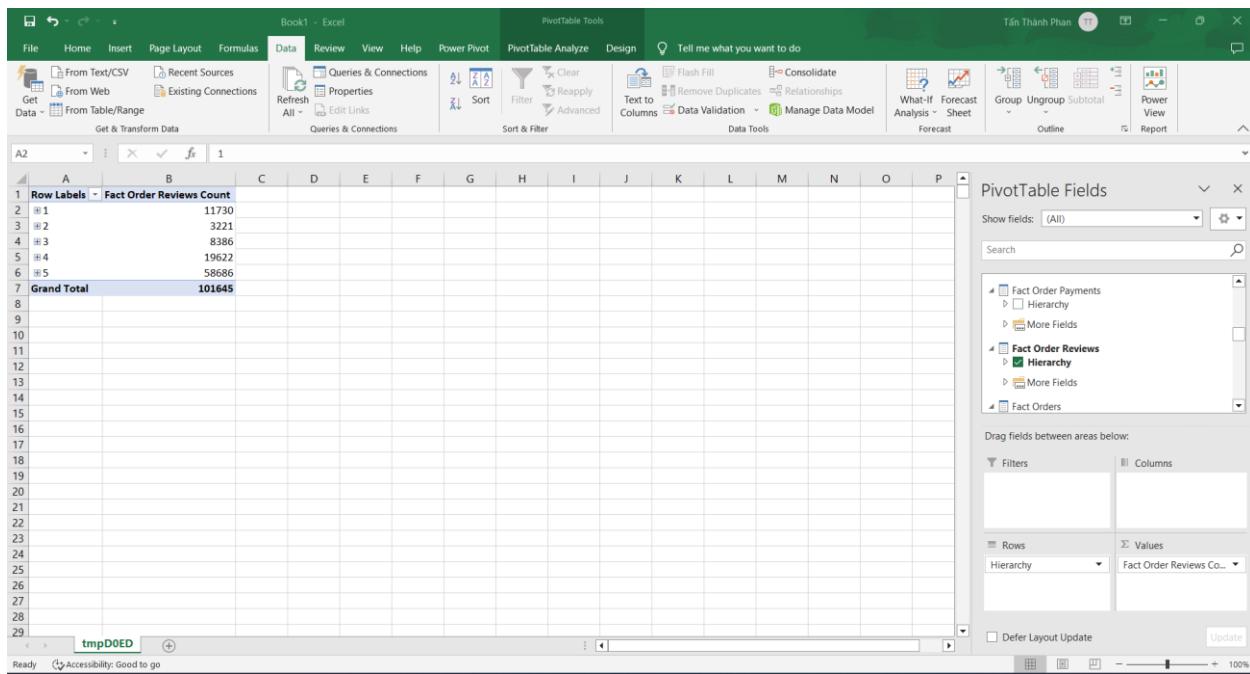


#### 4.5.3. Thống kê số điểm đánh giá thông qua các order:

- Sử dụng công cụ SSAS:



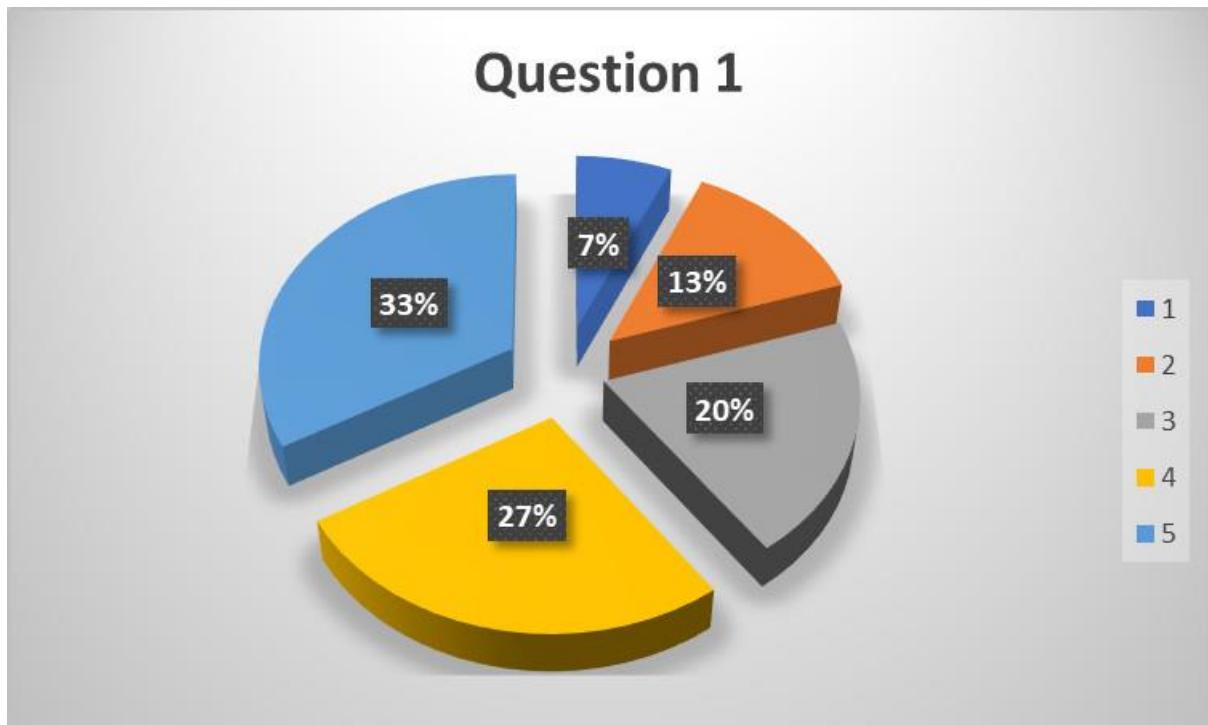
- Sử dụng Pivot Table trong Excel:



- Sử dụng ngôn ngữ truy vấn SQL:

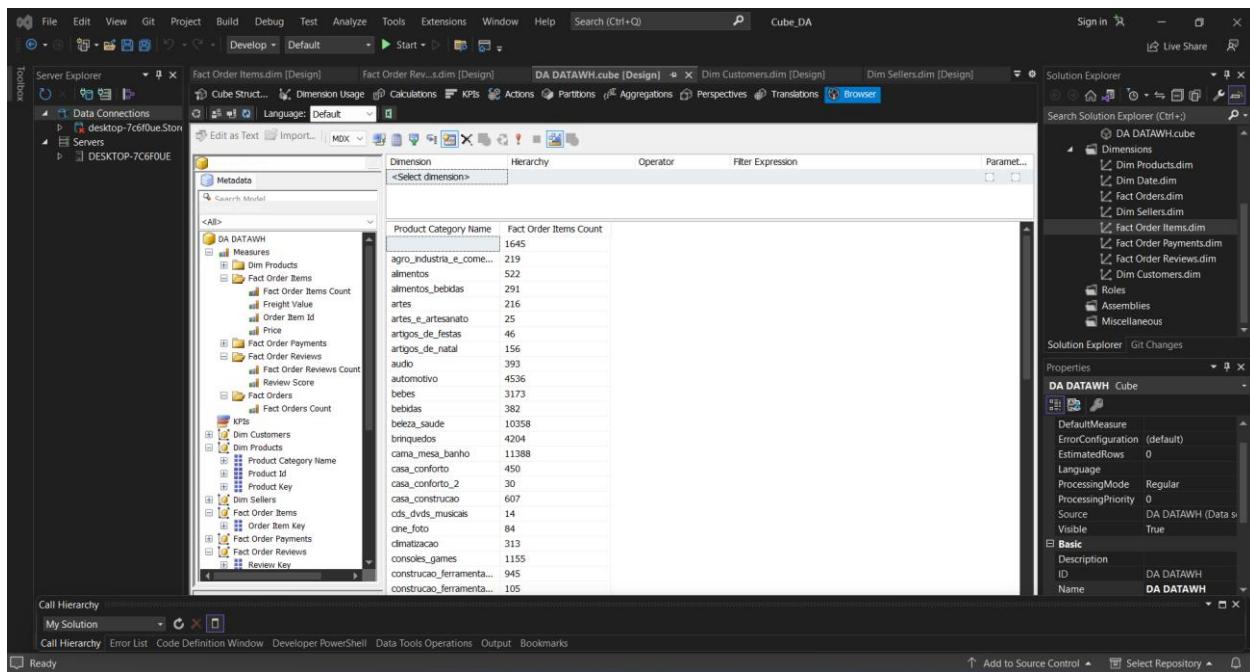
```
select review_score, count(review_score) as
number_of_votes
from [Fact_Review]
group by review_score
order by review_score
```

- Trực quan dữ liệu trên biểu đồ:



#### 4.5.4. Thống kê sản phẩm bán chạy:

- Sử dụng công cụ SSAS:



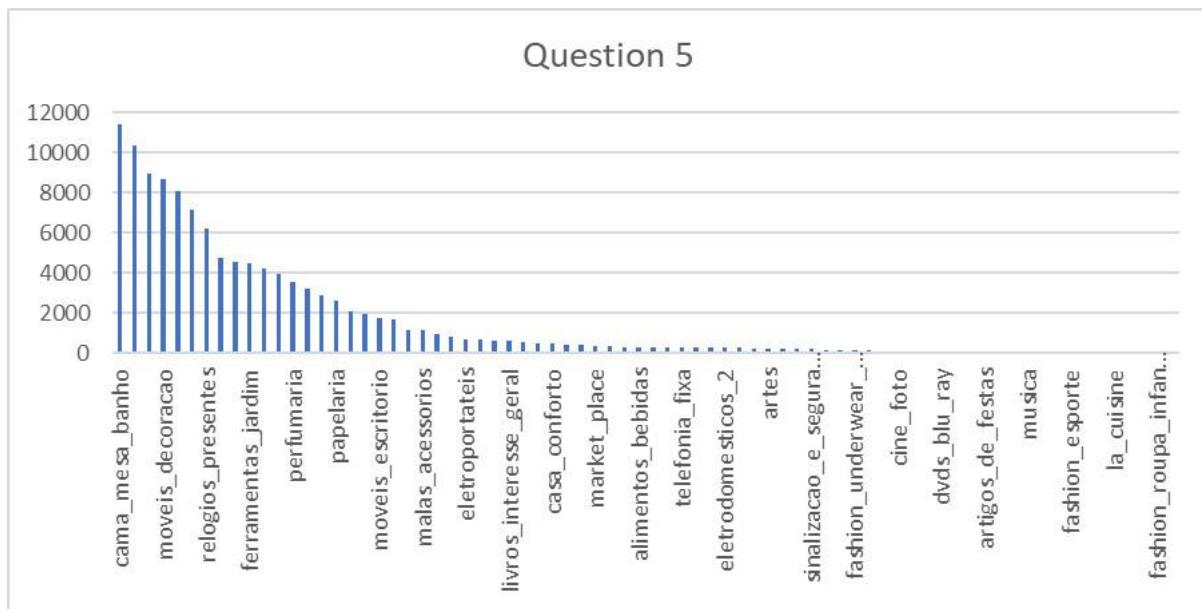
- Sử dụng Pivot Table trong Excel:

Row Labels	Fact Order Items Count
cama_mesa_banho	11388
beleza_saudé	10358
esporte_lazer	8910
moveis_decoracao	8659
informatica_acessorios	8067
utilidades_domesticas	7157
relogios_presentes	6233
telefonia	4727
automotivo	4536
fermentandas_jardim	4471
brinquedos	4204
cool_stuff	3915
perfumaria	3537
bebés	3173
eletronicos	2898
papelaria	2591
fashion_bolsas_e_acessorios	2085
pet_shop	1958
moveis_escritorio	1745
	1645
consoles_games	1155
malas_acessorios	1113
construcao_ferramentas_construcao	945
eletrodomesticos	794
eletroporcelais	705
instrumentos_musicais	689
casa_construcao	607
Início. Interesse geral	577

- Sử dụng ngôn ngữ truy vấn SQL:

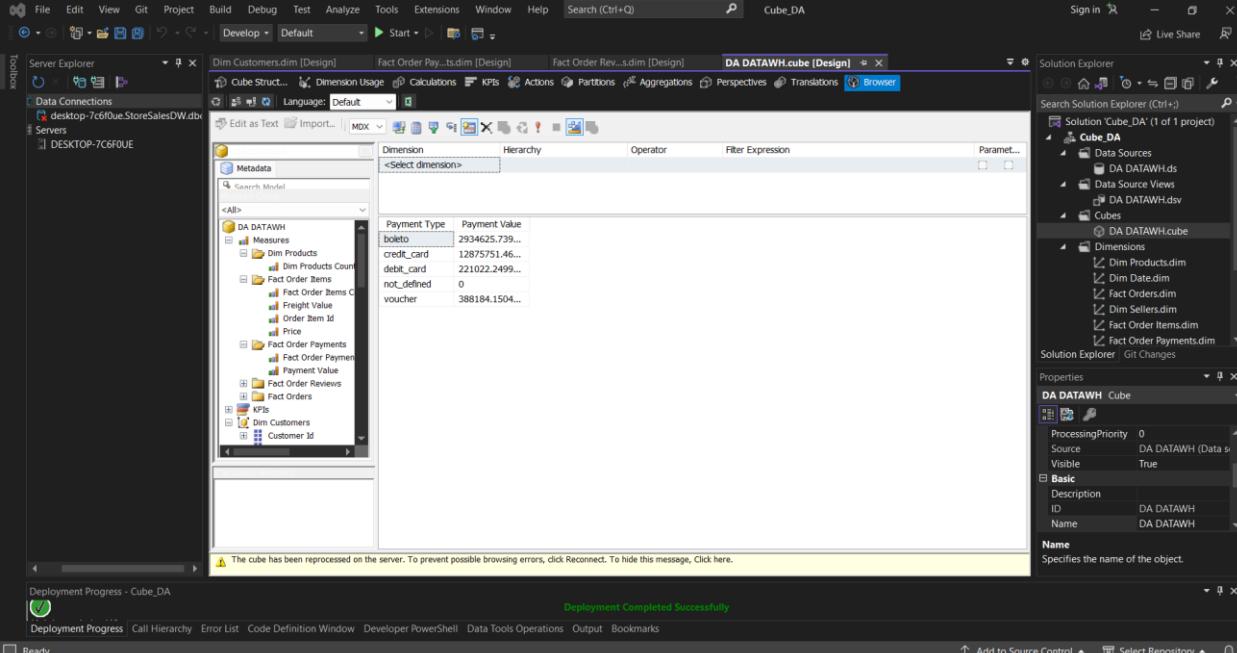
```
select top(10) p.product_category_name, count(i.product_key) as Quantity
    from [Dim_Product] as p
    join [Fact_Order_Items] as i on p.product_key=i.product_key
    group by p.product_category_name
    order by count(i.product_key) desc
```

- Trực quan dữ liệu trên biểu đồ:



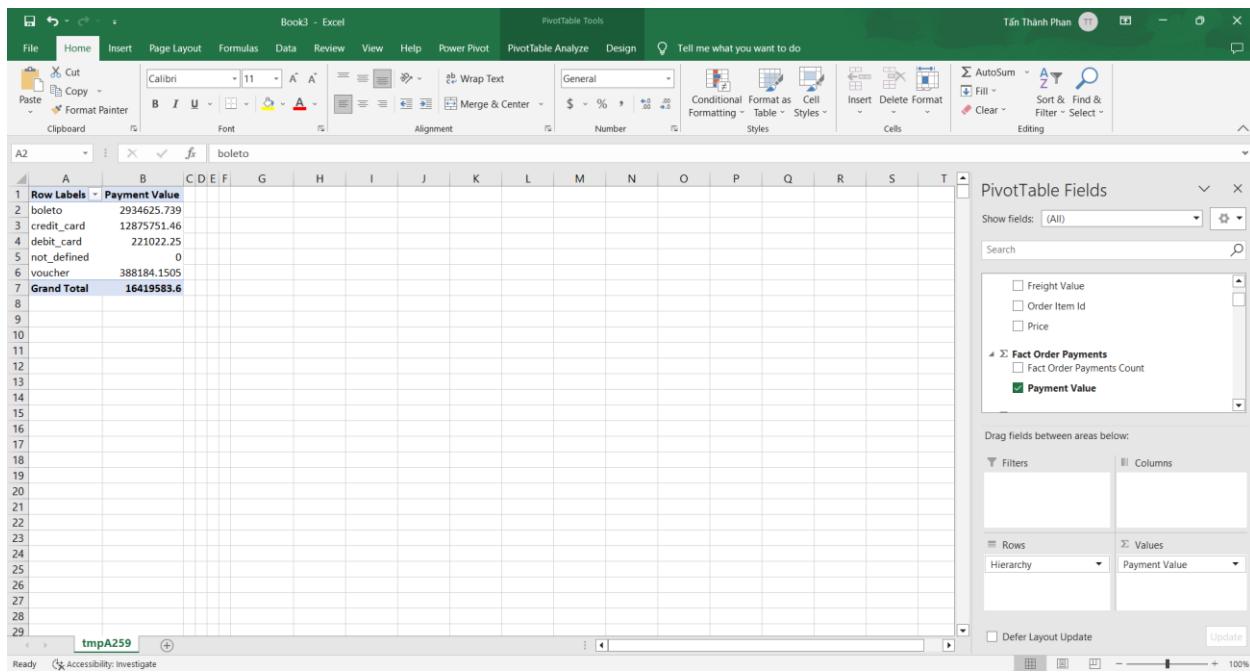
#### 4.5.5. Thống kê tổng tiền thông qua các hình thức thanh toán:

- Sử dụng công cụ SSAS:



Payment Type	Payment Value
boleto	2934625.739...
credit_card	12875751.46...
debit_card	221022.2499...
not_defined	0
voucher	388184.1504...

- Sử dụng Pivot Table trong Excel:



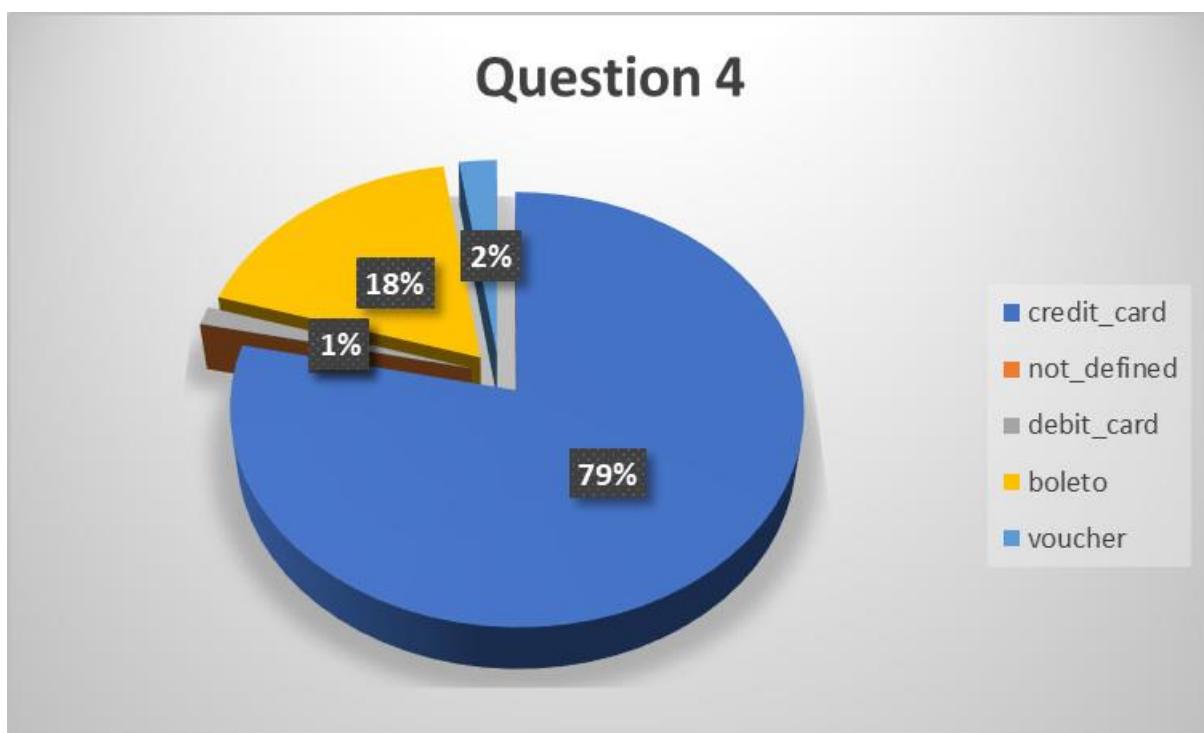
A screenshot of Microsoft Excel showing a PivotTable. The PivotTable Fields pane on the right shows fields from the Fact Order Payments table, with 'Payment Value' selected. The data grid on the left shows payment types and their values:

	Row Labels - Payment Value
1	boleto
2	2934625.739
3	credit_card
4	12875751.46
5	debit_card
6	221022.25
7	not_defined
8	0
9	voucher
10	388184.1505
11	<b>Grand Total</b>
12	<b>16419583.6</b>

- Sử dụng ngôn ngữ truy vấn SQL:

```
select distinct(payment_type), sum(payment_value) as payment_value
from Fact_Payment
group by payment_type
```

- Trực quan hóa dữ liệu trên biểu đồ:



## CHƯƠNG 5: KẾT LUẬN

### 5.1. Kết quả đạt được:

- Trong kỳ học vừa qua, nhóm đã tìm hiểu và vận dụng kiến thức về xây dựng kho dữ liệu và OLAP và đạt được các kết quả như sau:
  - o Nắm rõ các khái niệm cơ bản về kho dữ liệu và OLAP, các tính chất của một kho dữ liệu cần có.
  - o Có thể vận dụng được những kiến thức đã học để có thể xác định Business Process cho 1 kho dữ liệu.
  - o Nắm vững kiến thức và có thể vận dụng, xây dựng một kho dữ liệu hoàn chỉnh dùng để khai thác dữ liệu.
  - o Trang bị kiến thức về các công cụ SSIS, SSAS.
  - o Xây dựng được kho dữ liệu hoàn chỉnh.
  - o Trình bày tối ưu hóa câu truy vấn.

### 5.2. Những hạn chế:

- Do thời gian hạn ngắn cộng với khối lượng công việc nhiều nên trong quá trình thực hiện đồ án nhóm còn gặp phải một số vấn đề:
  - o Quá trình SSIS chưa được tối ưu, chưa thể tận dụng được hết các tool của SSIS.
  - o Quá trình phân tích bằng Pivot Table còn gặp một số lỗi trong quá trình thiết kế Cube.
  - o Tập dữ liệu còn ít và hạn chế nên chưa thể khai thác hết được các tiện ích của các công cụ.

### 5.3. Phân công công việc:

Công việc	Lê Tuấn Nghĩa	Lê Phúc Hậu	Phan Tấn Thành
Chọn dataset	X	X	X
Xác định các câu hỏi truy vấn	X	X	X
Xây dựng Business Process		X	X
Tạo kho dữ liệu OLAP	X	X	
Đổ data vào kho		X	
Tích hợp dữ liệu (SSIS)	X		

Xây dựng mô hình, khối – Cube (SSAS)			X
Truy vấn Pivot table			X
Truy vấn SQL	X	X	
Trực quan dữ liệu	X		X
Viết báo cáo	X	X	X

#### 5.4. Tài liệu tham khảo:

- Công cụ sử dụng:

- o Visual Studio 2019: <https://tinhte.vn/thread/download-visual-studio-2019-full-crack-huong-dan-cai-dat-chi-tiet.3562902/>
- o SQL Server 2019: <https://timoday.edu.vn/bai-1-tong-quan-ve-sql-server/>
- o Ngôn ngữ truy vấn SQL: <https://aws.amazon.com/vi/what-is/sql/>