

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA HỆ THÔNG THÔNG TIN



BÁO CÁO ĐÒ ÁN  
KHO DỮ LIỆU VÀ OLAP  
ĐỀ TÀI  
PHÂN TÍCH DOANH SỐ BÁN XE VÀ  
XU HƯỚNG THỊ TRƯỜNG

Giảng viên hướng dẫn: Đỗ Thị Minh Phụng

Lớp: IS217.O22.HTCL

Sinh viên thực hiện:

Đặng Trần Tuấn Anh 20521058

Đỗ Đình Đăng Khoa 21522218

TP. Hồ Chí Minh, tháng 3 năm 2024

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

....., ngày.....tháng.....năm 2024

## Người nhận xét

(Ký tên và ghi rõ họ tên)

## MỤC LỤC

<b>TỔNG QUAN ĐỀ TÀI .....</b>	<b>6</b>
<b>CHƯƠNG 1. GIỚI THIỆU KHO DỮ LIỆU .....</b>	<b>7</b>
1.1.    MÔ TẢ DATASET .....	7
1.2.    LỌC DỮ LIỆU .....	8
1.3.    DANH SÁCH THUỘC TÍNH ĐƯỢC PHÂN TÍCH .....	10
1.4.    XÂY DỰNG KHO DỮ LIỆU .....	12
1.4.1.    SƠ ĐỒ HÌNH SAO MINH HỌA .....	12
1.4.2.    MÔ TẢ CHI TIẾT CÁC BẢNG DỮ LIỆU .....	12
1.4.2.1.    BẢNG FACT .....	12
1.4.2.2.    DIM_TIME .....	13
1.4.2.3.    DIM_VEHICLE .....	14
1.4.2.4.    DIM_SELLER .....	14
<b>CHƯƠNG 2. QUÁ TRÌNH XÂY DỰNG KHO DỮ LIỆU (SSIS) .....</b>	<b>16</b>
2.1.    CHUẨN BỊ CÁC CÔNG CỤ .....	16
2.2.    CHUẨN BỊ CƠ SỞ DỮ LIỆU .....	20
2.3.    TẠO MỚI PROJECT SSIS .....	22
2.4.    TẠO BẢNG DIM VÀ BẢNG FACT .....	23
2.4.1.    BẢNG DIM_SELLER .....	29
2.4.2.    BẢNG DIM_TIME .....	37
2.4.3.    BẢNG DIM_VEHICLE .....	45
2.4.4.    BẢNG FACT .....	49
2.4.4.1.    MERGE FACT_RAW VÀ DIM_SELLER VÀO FACT1 .....	52
2.4.4.2.    MERGE FACT1 VÀ DIM_VEHICLE VÀO FACT .....	62
2.4.4.3.    TẠO KHÓA NGOẠI TỪ BẢNG FACT ĐẾN CÁC DIMENSION .....	72
2.4.5.    CHẠY DỰ ÁN SSIS .....	75
2.4.6.    KIỂM TRA DỮ LIỆU CÁC BẢNG .....	80
<b>CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU .....</b>	<b>85</b>
3.1.    CHUẨN BỊ CÁC CÔNG CỤ .....	85
3.2.    TẠO MỚI PROJECT SSAS .....	85
3.3.    XÁC ĐỊNH DỮ LIỆU NGUỒN (DATA SOURCES) .....	86
3.4.    XÁC ĐỊNH KHUNG NHÌN DỮ LIỆU NGUỒN (DATA SOURCE VIEWS) .....	90
3.5.    XÂY DỰNG CÁC KHỐI (CUBES) VÀ XÁC ĐỊNH CÁC ĐỘ ĐO (MEASURES) .....	93
3.6.    XÁC ĐỊNH CÁC CHIỀU (DIMENSIONS) .....	97
3.7.    XÁC ĐỊNH CÁC ĐỘ ĐO (MEASURES) .....	99

IS217 – Kho dữ liệu và OLAP	
3.7.1. ĐỔI TÊN VÀ THUỘC TÍNH CÁC ĐỘ ĐO BAN ĐẦU .....	99
3.7.2. TẠO CÁC ĐỘ ĐO MỚI.....	101
3.7.3. TỔNG KẾT ĐỘ ĐO.....	102
3.8. PHÂN CẤP TRONG BẢNG DIM_TIME.....	102
3.10. THỰC HIỆN 15 CÂU TRUY VẤN.....	108
3.10.1. SỐ LƯỢNG GIAO DỊCH BÁN XE CŨ THEO TỪNG NĂM .....	108
3.10.2. TỔNG DOANH THU TỪ VIỆC BÁN XE THEO TỪNG NGƯỜI BÁN .....	109
3.10.3. GIÁ BÁN TRUNG BÌNH THEO TỪNG HÃNG SẮP XẾP THEO GIÁ TRỊ GIẢM DẦN .....	110
3.10.4. TRUNG BÌNH DOANH THU THEO TỪNG THÁNG TRONG NĂM 2015 .....	111
3.10.5. TỔNG DOANH THU TỪ VIỆC BÁN XE SEDAN CÓ MÀU ĐỎ .....	112
3.10.6. TRUNG BÌNH GIÁ THỊ TRƯỜNG ƯỚC TÍNH THEO MỖI HÃNG XE .....	113
3.10.7. GIÁ TRỊ MMR TRUNG BÌNH VÀ ODOMETER CAO NHẤT THEO TỪNG DÒNG XE .....	115
3.10.8. SỐ LƯỢNG GIAO DỊCH VÀ TỔNG DOANH THU THEO TỪNG QUÝ .....	117
3.10.9. TÌM CÁC HÃNG XE CÓ DOANH THU DƯỚI 100,000 .....	118
3.10.10. SỐ LƯỢNG GIAO DỊCH BÁN XE CỦA CÁC NGƯỜI BÁN HÀNG TOP 5 .....	120
3.10.11. LẤY TOP 10 HÃNG XE THEO DOANH THU BÁN HÀNG.....	121
3.10.12. THỐNG KÊ SỐ LƯỢNG XE BÁN RA THEO TỪNG HÃNG, MẪU XE, LOẠI THÂN XE .....	121
3.10.13. TỔNG DOANH THU, SỐ LƯỢNG GIAO DỊCH BÁN XE VÀ TRUNG BÌNH GIÁ BÁN CHO CÁC MẪU XE CỦA HÃNG PORSCHE, PHÂN LOẠI THEO LOẠI THÂN XE VÀ MẪU XE CỤ THỂ. .....	123
3.10.14. THỐNG KÊ SỐ LƯỢNG GIAO DỊCH BÁN XE VÀ TỔNG DOANH THU THEO TỪNG NGÀY .....	124
3.10.15. TRUNG BÌNH GIÁ BÁN VÀ GIÁ TRỊ MMR TRUNG BÌNH THEO TỪNG NĂM:.....	125
3.10.16. SỐ LƯỢNG GIAO DỊCH BÁN XE THEO TỪNG NĂM VÀ TỪNG THÁNG TRONG NĂM ĐÓ .....	126
3.10.17. TỔNG DOANH THU TỪ VIỆC BÁN XE TRONG NĂM 2015 VÀ SO SÁNH VỚI NĂM 2014 .....	127
3.10.18 CHO BIẾT ĐIỀU KIỆN XE TỐT NHẤT MÀ SELLER ĐÃ BÁN THEO TỪNG NĂM .....	128
3.11. PHÂN TÍCH BẰNG PIVOT TABLE TRONG EXCEL.....	129
3.12 QUÁ TRÌNH LẬP BÁO BIỂU BẰNG CÔNG CỤ POWER BI .....	131
3.12.1. KẾT NỐI POWER BI VỚI SQL SERVER .....	131
3.12.2. THỰC HIỆN REPORT BẰNG POWER BI .....	134
3.12.2.1. CÂU TRUY VẤN 1.....	134
3.12.2.2 CÂU TRUY VẤN 2.....	136
3.12.2.3 CÂU TRUY VẤN 3.....	137
3.12.3 THỰC HIỆN REPORT BẰNG VISUAL STUDIO.....	139
3.12.3.1 CÂU TRUY VẤN 4.....	139
3.12.3.2 CÂU TRUY VẤN 5.....	141
3.12.3.3 CÂU TRUY VẤN 6.....	142
CHƯƠNG 4. QUÁ TRÌNH KHAI THÁC DỮ LIỆU (DATA MINING) .....	145
4.1. PHÂN TÍCH DATASET GỐC.....	145
4.1.1. THỐNG KÊ MÔ TẢ.....	145

IS217 – Kho dữ liệu và OLAP	
4.1.2. TRỰC QUAN HÓA DỮ LIỆU .....	146
4.1.2.1. TẠO BIỂU ĐỒ ĐỂ HIỂN THỊ SỐ LƯỢNG XE TRONG TỪNG LOẠI TÌNH TRẠNG.....	146
4.1.2.2. TẠO BIỂU ĐỒ PHÂN TÁC GIỮA SỐ KM ĐÃ ĐI (ODOMETER) VÀ GIÁ BÁN (SELLING PRICE).....	148
4.1.2.3. TẠO BIỂU ĐỒ HIỂN THỊ GIÁ BÁN TRUNG BÌNH THEO NĂM SẢN XUẤT. ....	149
4.2. TIỀN XỬ LÝ DỮ LIỆU.....	150
4.2.1. KIỂM TRA MỐI TƯƠNG QUAN GIỮA CÁC THUỘC TÍNH.....	151
4.2.2. LỰA CHỌN THUỘC TÍNH.....	152
4.2.3. THÊM THUỘC TÍNH.....	153
4.2.4. XÓA TRÙNG LẶP .....	154
4.3. ỨNG DỤNG MÔ HÌNH THUẬT TOÁN KHAI THÁC DỮ LIỆU .....	155
4.3.1. CHIA DỮ LIỆU TRƯỚC KHI XÂY DỰNG MÔ HÌNH THUẬT TOÁN.....	155
4.3.2. THỰC HIỆN XÂY DỰNG MÔ HÌNH DECISION TREE VÀ RANDOM FOREST.....	156
4.3.3. KẾT QUẢ SAU KHI THỰC HIỆN.....	157
4.3.3.1.....	157
4.3.3.2. CÂY QUYẾT ĐỊNH DECISION TREE .....	158
4.3.3.3. CÂY QUYẾT ĐỊNH RANDOM FOREST .....	160
4.3.4. SỬ DỤNG MÔ HÌNH RANDOM FOREST VÀ VẼ CÁC ĐỒ THỊ LIÊN QUAN .....	163
4.3.5. SỬ DỤNG MÔ HÌNH RANDOM FOREST ĐỂ ĐƯA RA DỰ ĐOÁN “SELLINGPRICE” DỰA TRÊN CÁC THUỘC TÍNH “CONDITION”, “ODOMETER”, “SALEDAY”, “SALEMONT”, “SALEYEAR”. .....	164
4.3.6. SO SÁNH KẾT QUẢ DỰ ĐOÁN VỚI KẾT QUẢ THỰC TẾ TRONG DATASET.....	165
4.3.7. TẬP LUẬN DÀNH CHO NGƯỜI DÙNG CUỐI. .....	166
DANH MỤC TÀI LIỆU THAM KHẢO .....	168

## TỔNG QUAN ĐỀ TÀI

Phân tích doanh số bán xe và xu hướng thị trường là một lĩnh vực quan trọng trong ngành công nghiệp ô tô, đặc biệt là trong bối cảnh ngày càng tăng của sự cạnh tranh và sự phát triển của công nghệ. Bộ dữ liệu về doanh số bán xe và xu hướng thị trường cung cấp một tập hợp thông tin đa dạng và toàn diện về các giao dịch mua bán xe hơi. Từ các chi tiết như năm sản xuất, hãng sản xuất, mẫu xe, trang trí, loại thân xe cho đến các yếu tố như loại hộp số, số VIN, trạng thái đăng ký, và xếp hạng tình trạng, bộ dữ liệu này cho phép chúng ta phân tích sâu hơn về thị trường ô tô.

Thông qua việc phân tích dữ liệu, nhà nghiên cứu có thể đưa ra những nhận định quan trọng về xu hướng thị trường, sự thay đổi trong sở thích của người tiêu dùng, và các yếu tố ảnh hưởng đến quyết định mua bán xe hơi. Đồng thời, bằng cách áp dụng các phương pháp phân tích dữ liệu tiên tiến, như mô hình dự đoán và phân tích chuỗi thời gian, chúng ta có thể dự báo và đánh giá hiệu quả các chiến lược tiếp thị và kinh doanh trong ngành công nghiệp ô tô.

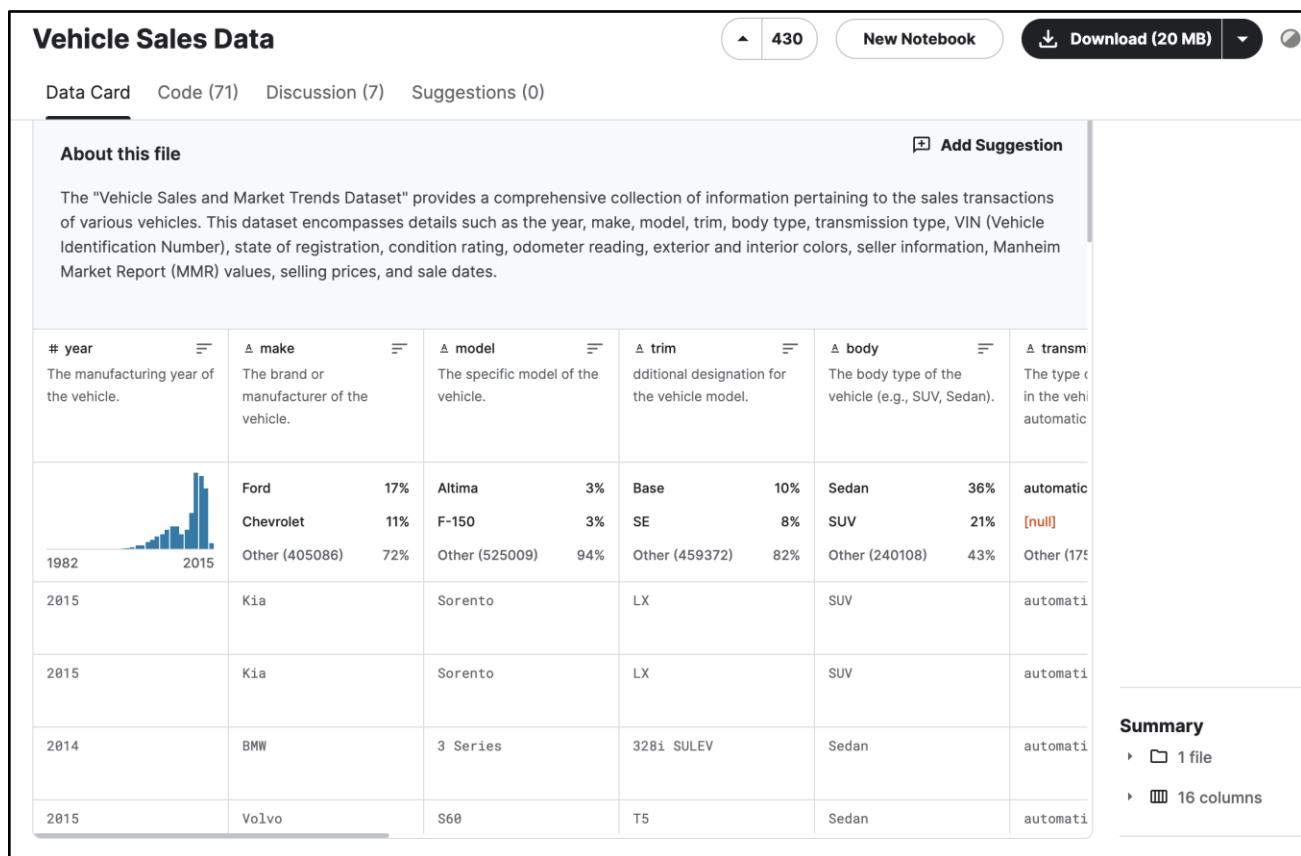
Mục tiêu của nghiên cứu này là không chỉ làm sáng tỏ những yếu tố ảnh hưởng đến doanh số bán xe, mà còn giúp các doanh nghiệp và nhà quản lý thị trường hiểu rõ hơn về nhu cầu và xu hướng của thị trường, từ đó đưa ra các quyết định kinh doanh và chiến lược phù hợp. Bằng cách này, chúng ta hứa hẹn có thể đóng góp vào việc nâng cao hiệu quả hoạt động kinh doanh và tạo ra giá trị cho ngành công nghiệp ô tô.

# CHƯƠNG 1. GIỚI THIỆU KHO DỮ LIỆU

*Giới thiệu tổng quan về Kho dữ liệu Vehicle Sales*

## 1.1. Mô tả dataset

- Tên bộ dữ liệu: Vehicle Sales (Bộ dữ liệu về doanh số bán xe và xu hướng thị trường).



- Đây là tập dữ liệu về doanh số bán xe và xu hướng thị trường.
- Cập nhật gần nhất vào tháng 3 năm 2024
- Tên tác giả: Syed Anwar

- Bộ dữ liệu gồm 558,838 dòng dữ liệu và 16 cột thuộc tính.
- Bộ dữ liệu được thu thập từ năm 1982 đến tháng năm 2015.
- Nguồn: <https://www.kaggle.com/datasets/syedanwarafridi/vehicle-sales-data/data>

## 1.2. Lọc dữ liệu

Bởi vì dữ liệu khá lớn, thiết bị nhóm khó có thể xử lý hết nên nhóm em quyết định sẽ lấy dữ liệu các xe được sản xuất từ năm 2005 -2015 để xây dựng cho kho dữ liệu.

Sử dụng python để lọc dữ liệu

### Bước 1: Nhập các thư viện cần thiết và đọc file dataset

```
✓ 21 phút [5] from google.colab import files
      uploaded = files.upload()

      Chọn tệp car_prices.csv
      • car_prices.csv(text/csv) - 88047552 bytes, last modified: 21/2/2024 - 100% done
      Saving car_prices.csv to car_prices.csv
      Chọn tệp Chưa có tệp nào được chọn

✓ 5 giây [6] import numpy as np
      import pandas as pd
      df=pd.read_csv('car_prices.csv')
      df.head()
```

year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate	
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america inc	20500.0	21500.0	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america inc	20800.0	21500.0	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45.0	1331.0	gray	black	financial services remarketing (lease)	31900.0	30000.0	Thu Jan 15 2015 04:30:00 GMT-0800 (PST)
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni	27500.0	27750.0	Thu Jan 29 2015 04:30:00 GMT-0800 (PST)
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43.0	2641.0	gray	black	financial services remarketing (lease)	66000.0	67000.0	Thu Dec 18 2014 12:30:00 GMT-0800 (PST)

## Bước 2: Xóa các dòng có dữ liệu bị thiếu hoặc null

1	[7]	df=df.dropna()														
gây																
year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate	
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america inc	20500.0	21500.0	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america inc	20800.0	21500.0	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45.0	1331.0	gray	black	financial services remarketing (lease)	31900.0	30000.0	Thu Jan 15 2015 04:30:00 GMT-0800 (PST)
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni	27500.0	27750.0	Thu Jan 29 2015 04:30:00 GMT-0800 (PST)
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43.0	2641.0	gray	black	financial services remarketing (lease)	66000.0	67000.0	Thu Dec 18 2014 12:30:00 GMT-0800 (PST)

## Bước 3: Chuyển đổi cột saledate sang định dạng datetime

[13]	df['saledate'] = pd.to_datetime(df['saledate'], utc=True, errors='coerce')															
df																
<ipython-input-13-9572d24e5ddb>:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back to 'dateutil'. To ensure parsing is consistent and : df['saledate'] = pd.to_datetime(df['saledate'], utc=True, errors='coerce')																
year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate	
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america inc	20500.0	21500.0	2014-12-16 04:30:00+00:00
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america inc	20800.0	21500.0	2014-12-16 04:30:00+00:00
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45.0	1331.0	gray	black	financial services remarketing (lease)	31900.0	30000.0	2015-01-14 20:30:00+00:00
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni	27500.0	27750.0	2015-01-28 20:30:00+00:00
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43.0	2641.0	gray	black	financial services remarketing (lease)	66000.0	67000.0	2014-12-18 04:30:00+00:00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
558831	2011	BMW	5 Series	528i	Sedan	automatic	wbafr1c53bc744672	fl	39.0	66403.0	white	brown	lauderdale imports ltd bmw pembrok pines	20300.0	22800.0	2015-07-06 23:15:00+00:00
558833	2012	Ram	2500	Power Wagon	Crew Cab	automatic	3c6td5et6cg112407	wa	5.0	54393.0	white	black	i-5 uhlmann rv	30200.0	30800.0	2015-07-08 02:30:00+00:00
558834	2012	BMW	X5	xDrive35d	SUV	automatic	5uxzw0c58cl668465	ca	48.0	50561.0	black	black	financial services remarketing (lease)	29800.0	34000.0	2015-07-08 02:30:00+00:00

## Bước 4: Xử lý để loại bỏ múi giờ +00:00

```

❶ df['saledate'] = df['saledate'].dt.strftime('%Y-%m-%d %H:%M:%S')

❷ # Hiển thị DataFrame sau khi xử lý
df

```

	year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg666472	ca	5.0	16639.0	white	black	kia motors america inc	20500.0	21500.0	2014-12-16 04:30:00
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america inc	20800.0	21500.0	2014-12-16 04:30:00
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45.0	1331.0	gray	black	financial services remarketing (lease)	31900.0	30000.0	2015-01-14 20:30:00
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612b4ff1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni	27500.0	27750.0	2015-01-28 20:30:00
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43.0	2641.0	gray	black	financial services remarketing (lease)	66000.0	67000.0	2014-12-18 04:30:00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
558831	2011	BMW	5 Series	528i	Sedan	automatic	wba1c53bc744672	fl	39.0	66403.0	white	brown	lauderdale imports ltd bmw pembrok pines	20300.0	22800.0	2015-07-06 23:15:00
558833	2012	Ram	2500	Power Wagon	Crew Cab	automatic	3c6td5et6cg112407	wa	5.0	54393.0	white	black	i-5 uhlmann rv	30200.0	30800.0	2015-07-08 02:30:00
558834	2012	BMW	X5	xDrive35d	SUV	automatic	5uxzw0c58cl668465	ca	48.0	50561.0	black	black	financial services remarketing (lease)	29800.0	34000.0	2015-07-08 02:30:00
558835	2015	Nissan	Altima	2.5 S	sedan	automatic	1n4al3ap0fc216050	ga	38.0	16658.0	white	black	enterprise vehicle exchange / tra/ rental /...	15100.0	11100.0	2015-07-08 23:45:00
558836	2014	Ford	F-150	XLT SuperCrew	automatic	1ftfw1et2eke87277	ca	34.0	15008.0	gray	gray	ford motor credit company llc pd	29600.0	26700.0	2015-05-27 22:30:00	

472325 rows × 16 columns

## Bước 5: Lưu dataset đã lọc thành file CSV “FINALDATA\_VEHICLE” để xây dựng kho dữ liệu sau này.

```
[17] df.to_csv('FINALDATA_VEHICLE.csv', index = False)
```

Dữ liệu sau khi lọc:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	year	make	model	trim	body	transmissi	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate
2	1990	Toyota	Camry	Deluxe	Sedan	automatic	j12sv21exl0345999	wa	2	214723	blue	blue	donate for	675	375	1/27/2015 20:30
3	1990	Honda	Accord	EX	Sedan	automatic	jhmc7661c036504	tx	2	19279	gray	tan	automotive	875	350	12/23/2014 2:00
4	1990	Toyota	Camry	Deluxe	Sedan	automatic	4t1sv21e0lu227097	ca	2	122877	blue	á€"	charitable	700	400	12/30/2014 5:00
5	1990	Chevrolet	Corvette	Base	Hatchback	automatic	1g1yv238751058284	oh	1	1	red	red	purple hea	7850	2800	12/23/2014 5:00
6	1990	Honda	Accord	LX	Sedan	automatic	jhmc765lx131957	rv	1	183366	gold	á€"	automotiv	775	400	1/21/2015 19:00
7	1990	Mercedes-Benz	300-Class	300E	Sedan	automatic	wdbea30d6b200847	rv	2	141799	white	á€"	automotive	425	300	1/21/2015 19:00
8	1990	Lexus	LS 400	Base	Sedan	automatic	jtufl11e410041243	ca	3	106472	white	tan	illest moto	550	700	12/31/2014 3:30
9	1990	Honda	Accord	EX	Sedan	automatic	jhmc76651c099475	az	1	247555	gray	gray	cash time	675	400	1/7/2015 3:00
10	1990	Toyota	Corolla	Base	Sedan	automatic	j12ae91a013295341	or	2	195851	red	gray	purple hea	475	200	1/5/2015 19:31
11	1990	Cadillac	DeVille	Base	Sedan	automatic	1g6cd53334326699	pa	2	120590	white	purple	purple hea	400	325	5/13/2015 5:01
12	1990	Toyota	Corolla	Deluxe	Sedan	automatic	21ae94a8l016373	fl	2	213005	white	gray	autonation	500	500	1/13/2015 10:00
13	1990	Lexus	LS 400	Base	Sedan	automatic	jtufl11e410028791	fl	1	102437	black	gray	autonation	600	400	1/13/2015 10:00
14	1990	Toyota	Camry	Deluxe	Sedan	automatic	4t1sv21e3u252866	ga	2	201659	black	gray	purple hea	675	375	1/14/2015 20:30
15	1990	Honda	Accord	LX	Sedan	manual	1hgb7554d154103	md	2	32286	blue	black	purple hea	725	300	1/19/2015 20:00
16	1990	Mazda	MX-5 Miata	Base	Convertible	manual	j11na35150123822	ga	2	129102	white	black	five star do	1400	900	1/29/2015 19:00
17	1990	Honda	Accord	LX	Sedan	automatic	jhmc7651c020391	ca	2	237586	gold	burgundy	honda of si	600	350	1/28/2015 20:00
18	1990	Lexus	LS 400	Base	Sedan	automatic	jtufl11e10039294	ca	2	249117	black	beige	purple hea	700	300	2/24/2015 23:10
19	1990	Lexus	LS 400	Base	Sedan	automatic	jtufl11e410008543	md	1	1	white	blue	axcess fina	700	225	2/16/2015 21:00
20	1990	Chevrolet	C/K 1500 Ser 454SS	Regular Cab	Automatic	1gcdc14n4l215783	fl	3	101927	black	red	american l	7225	8000	2/9/2015 17:30	
21	1990	Nissan	300ZX	GS	Hatchback	manual	j11r2441k006044	tx	1	181363	red	red	remarket	1500	200	3/4/2015 22:00
22	1990	Lexus	LS 400	Base	Sedan	automatic	jtufl11e410010629	fl	2	221945	white	tan	autonation	575	1100	2/17/2015 2:00
23	1990	Mazda	MX-5 Miata	Base	Convertible	manual	j11na35130126525	md	2	119379	blue	black	manheim t	1500	425	2/16/2015 21:00
24	1990	Chevrolet	C/K 1500 Ser 454SS	Regular Cab	Automatic	1gcdc14n52242061	tn	4	34266	black	red	t & s motor	9550	11500	3/3/2015 18:30	
25	1990	Mazda	MX-5 Miata	Base	Convertible	manual	j11na35100124909	fl	2	63300	red	black	mini of fort	2150	1600	3/3/2015 2:00
26	1990	Mazda	MX-5 Miata	Base	Convertible	manual	j11na35120109926	ca	2	70370	blue	gray	aero swe	2075	2500	3/3/2015 19:30
27	1990	Toyota	4Runner	SR5 V6	SUV	automatic	jtdvn39w10036787	ca	1	206522	red	gray	premium a	750	550	3/4/2015 20:00
28	1990	Mazda	300-Class	300E	Sedan	automatic	j11na35100124907	ca	2	117089	white	black	data	650	250	4/28/2015 22:00

### 1.3. Danh sách thuộc tính được phân tích

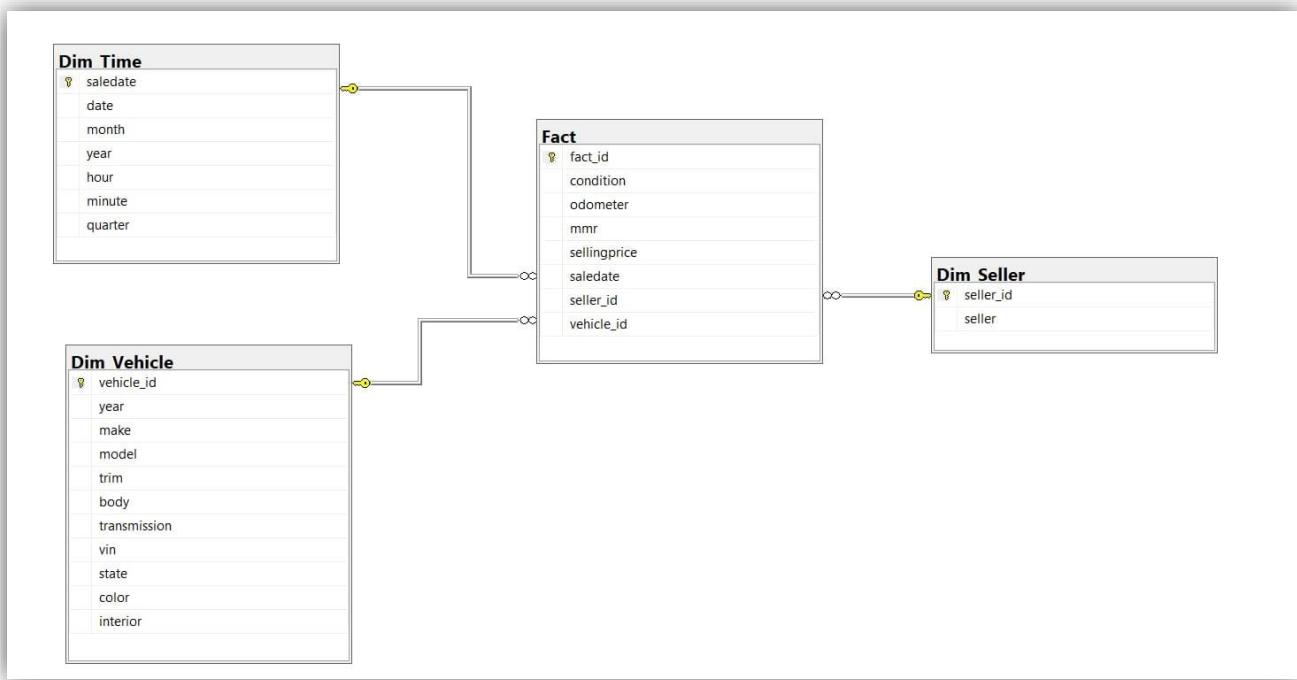
STT	Tên thuộc tính	Kiểu	Ý nghĩa
		dữ liệu	

1	Year	String	Năm sản xuất của xe.
2	Make	String	Thương hiệu hoặc nhà sản xuất xe.
3	Model	String	Model cụ thể của xe.
4	Trim	String	Chỉ định bổ sung cho mẫu xe.
5	Body	String	Loại thân xe (ví dụ: SUV, Sedan).
6	Transmission	String	Loại hộp số trên xe (ví dụ: Hộp số tự động).
7	Vin	String	Số nhận dạng phương tiện, mỗi mã duy nhất cho mỗi phương tiện.
8	State	String	Tiểu bang nơi chiếc xe hơi được đăng kí.
9	Condition	Int	Tình trạng của xe, có thể được đánh giá theo thang điểm từ 1 đến 49.
10	Odometer	Int	Số dặm hoặc quãng đường mà xe đã đi được.
11	Color	String	Màu sắc ngoại thất của xe.
12	Interior	String	Màu sắc nội thất của xe.
13	Seller	String	Đơn vị bán xe.
14	Mmr(Manheim market report)	Int	Báo cáo thị trường Manheim, có thể chỉ ra giá trị thị trường ước tính của xe).

15	Sellingprice	Int	Giá mà chiếc xe đã được bán.
16	Selldate	datetime	Thời gian xe được bán.

## 1.4. Xây dựng kho dữ liệu

### 1.4.1. Sơ đồ hình sao minh họa



### 1.4.2. Mô tả chi tiết các bảng dữ liệu

#### 1.4.2.1. Bảng Fact

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	Fact_id	String	PK	Mã của bảng fact.
2	Vehicle_id	String	FK	Mã xe.

3	Seller_id	Int	FK	Mã đơn vị bán xe.
4	Saledate	datetime	FK	Thời gian xe được bán.
5	Condition	Int		Tình trạng của xe, được đánh giá theo thang điểm từ 1 đến 49.
6	Odometer	Int		Số dặm hoặc quãng đường mà xe đã đi được.
7	Mmr	Int		Giá trị thị trường ước tính của xe.
8	Sellingprice	Int		Giá mà chiếc xe đã được bán.

#### 1.4.2.2. Dim\_Time

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
2	Saledate	datetime	PK	Thời gian xe được bán.
3	Day	Int		Ngày xe được bán.
4	Month	Int		Tháng xe được bán.
5	Year	Int		Năm xe được bán.
6	Hour	Int		Giờ xe được bán.
7	Minute	Int		Phút xe được bán.

8	Quarter	Int		Quý xe được bán.
---	---------	-----	--	------------------

#### 1.4.2.3. Dim\_Vehicle

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	Vehicle_id	String	PK	Mã xe.
2	Year	String		Năm sản xuất của xe.
3	Make	String		Thương hiệu hoặc nhà sản xuất.
4	Model	String		Model cụ thể của xe.
5	Trim	String		Chỉ định bổ sung cho mẫu xe.
6	Body	String		Loại thân xe (ví dụ: SUV, Sedan,...)
7	Transmission	String		Loại hộp số trên xe (ví dụ: hộp số sàn,...)
8	Vin	String		Số nhận dạng phương tiện.
9	State	String		Tiểu bang nơi chiếc xe được đăng ký.
10	Color	String		Màu sắc ngoại thất của xe.
11	Interior	String		Màu sắc nội thất của xe.

#### 1.4.2.4. Dim\_Seller

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	Seller_id	String	PK	Mã đơn vị bán xe.
2	Seller	String		Đơn vị bán xe.

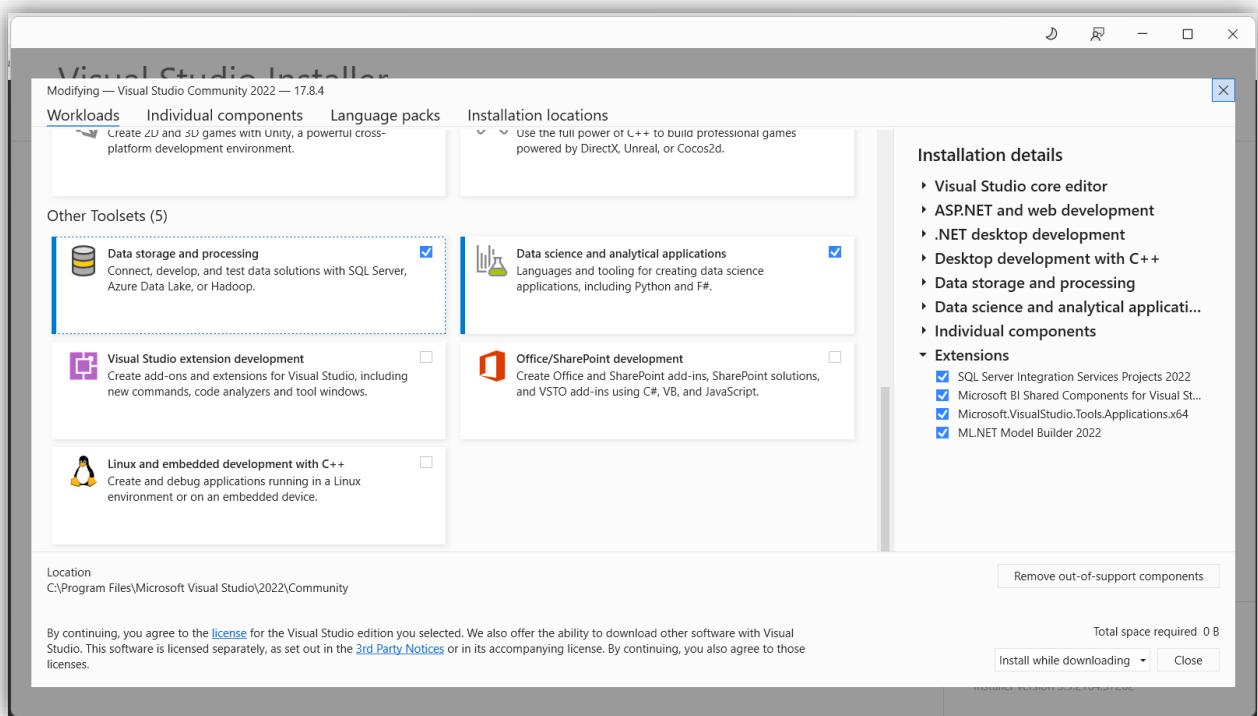
## CHƯƠNG 2. QUÁ TRÌNH XÂY DỰNG KHO DỮ LIỆU (SSIS)

### 2.1. Chuẩn bị các công cụ

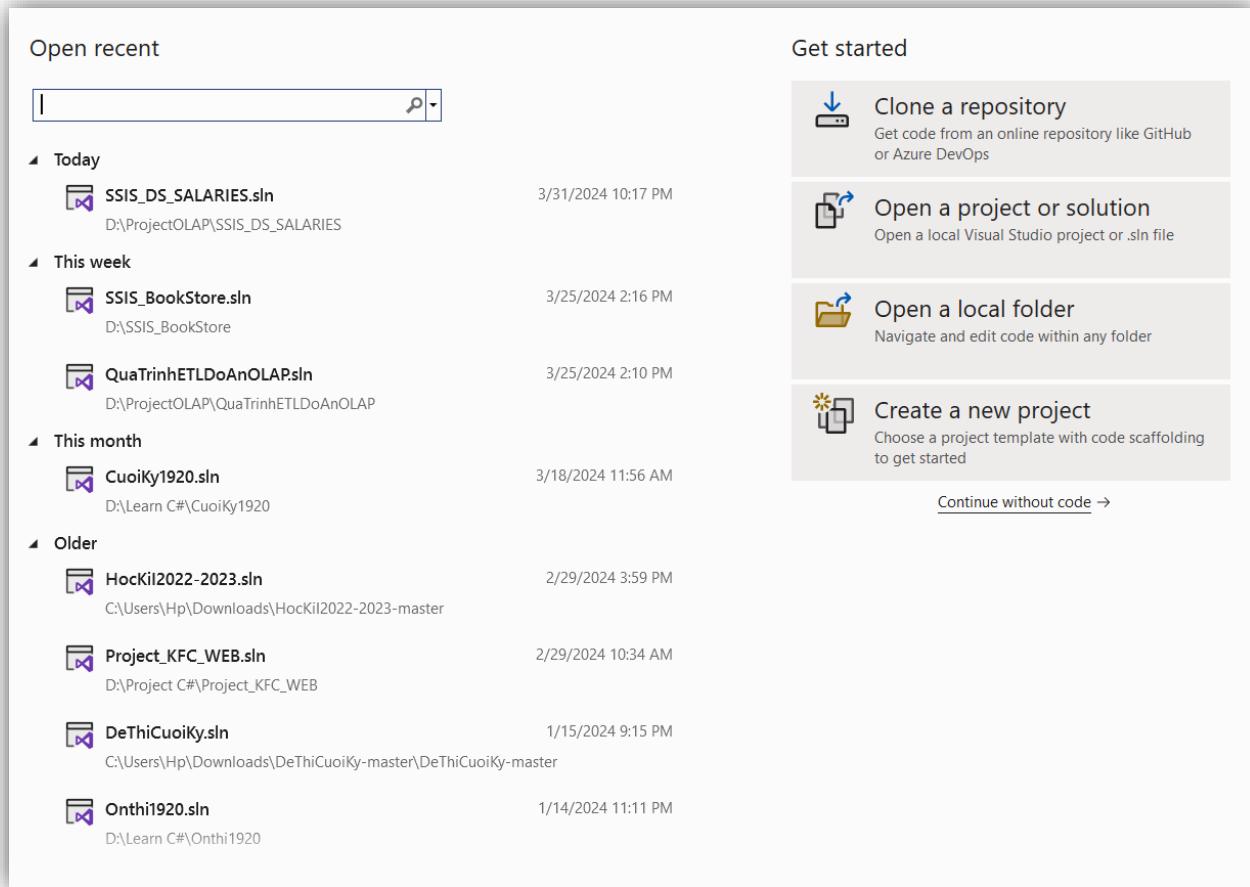
Để thực hiện được quá trình SSIS ta cần chuẩn bị và cài đặt các công cụ sau:

- Visual Studio Community 2022
- SQL Server Integration Services Project

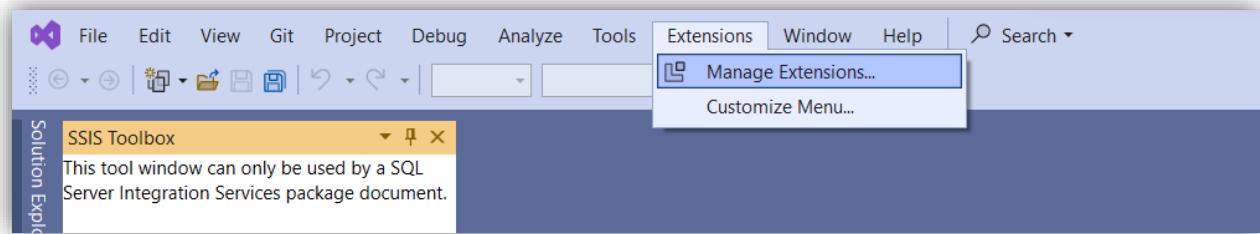
**Bước 1:** Tải Visual Studio Community 2022 về máy. Trong lúc cài đặt, chọn mục “Data storage and processing” để cài đặt SQL Server Data Tools. Sau đó chọn Install để tiến hành cài đặt



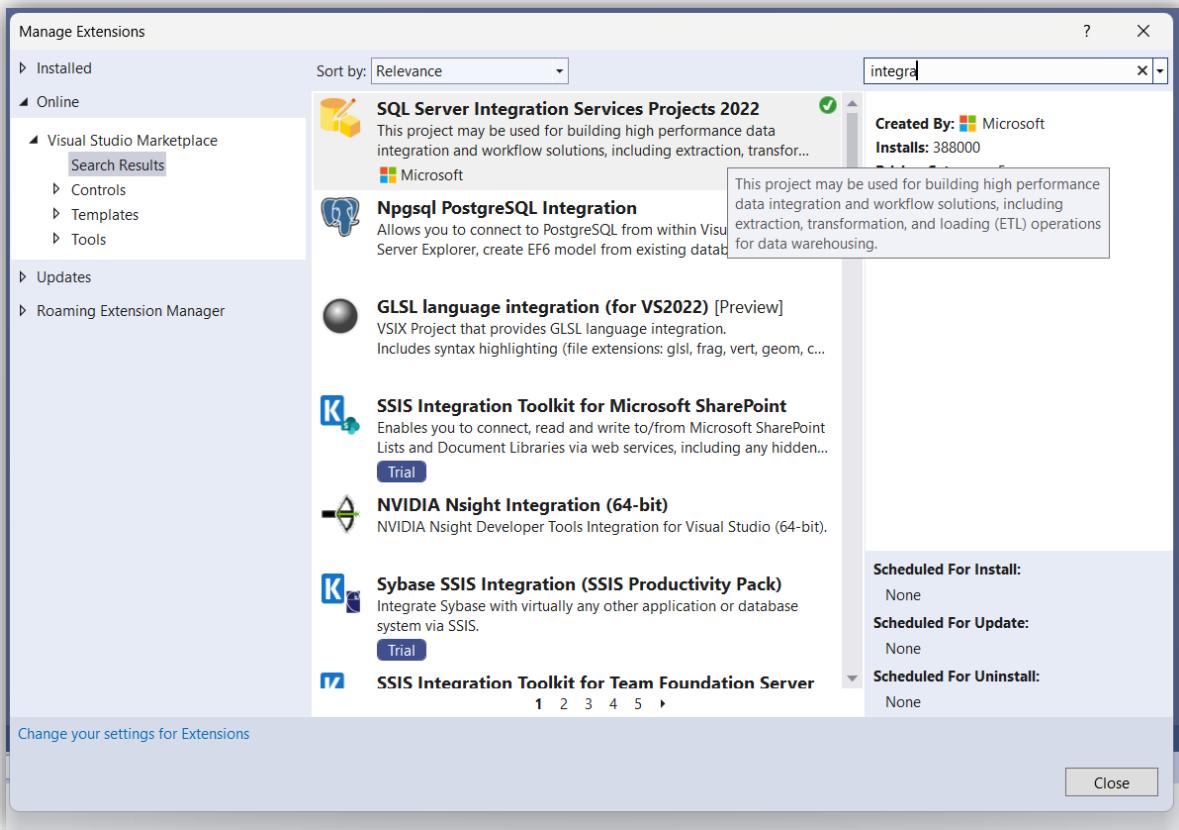
**Bước 2:** Mở Visual Studio 2022 và chọn “Continue without code”



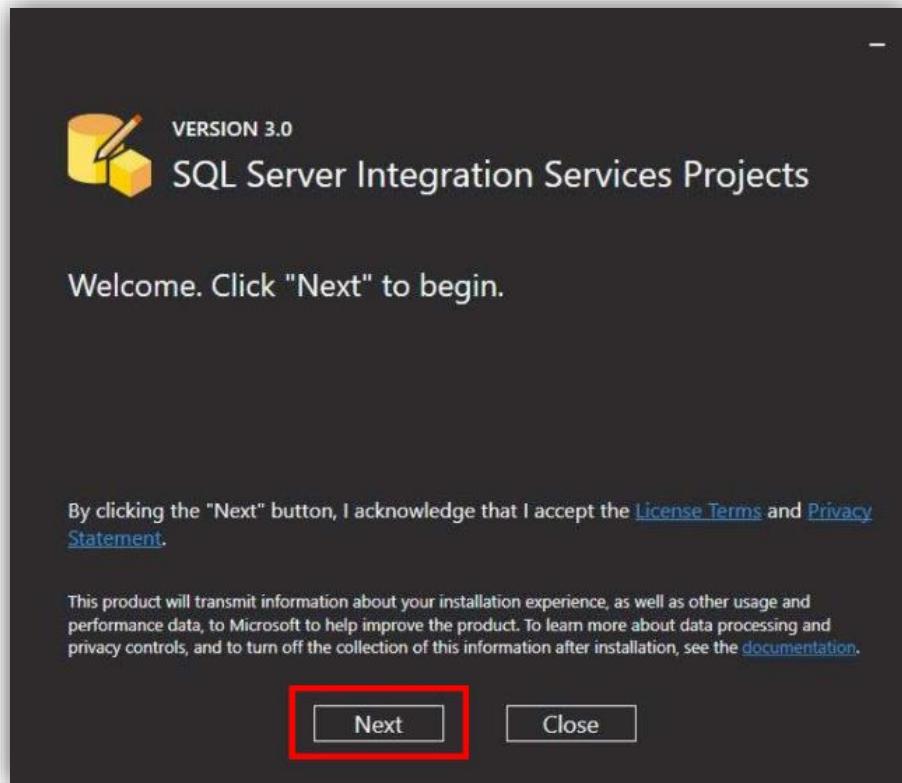
**Bước 3:** Trong giao diện chính, click chọn “Extensions” > “Manage Extensions”



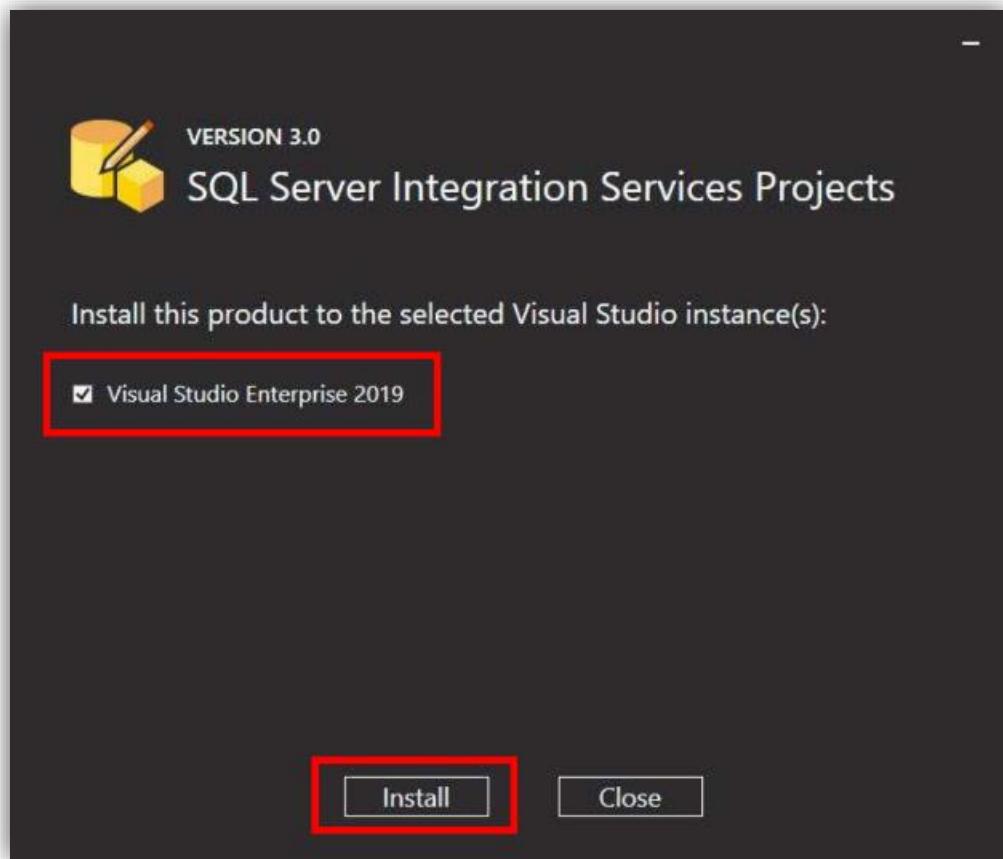
**Bước 4:** Tìm và tải về công cụ SQL Server Integration Services Projects.



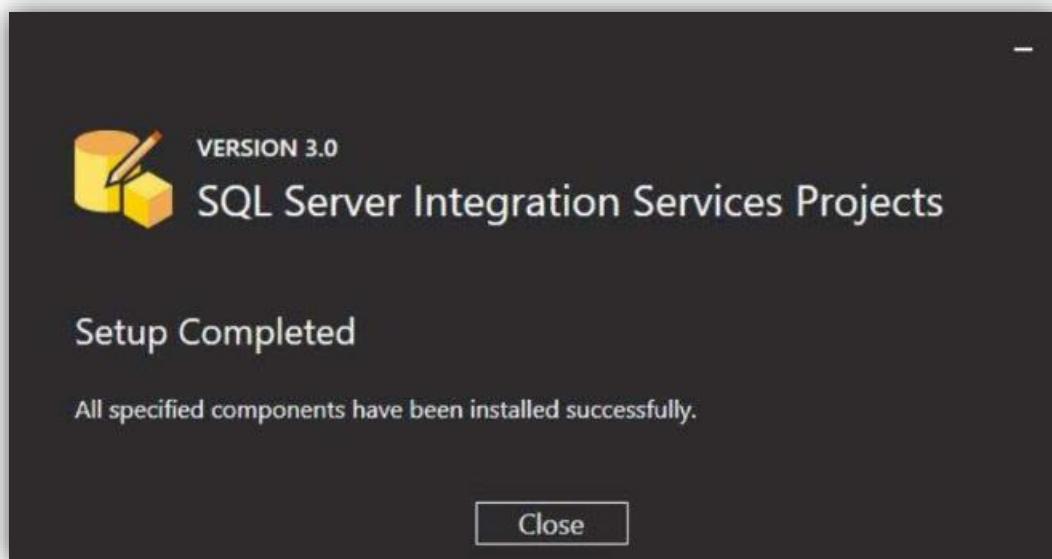
**Bước 5:** Mở file .exe vừa tải xuống. Chọn Next để tiếp tục



**Bước 6:** Tick chọn Visual Studio 2019 và chọn Install

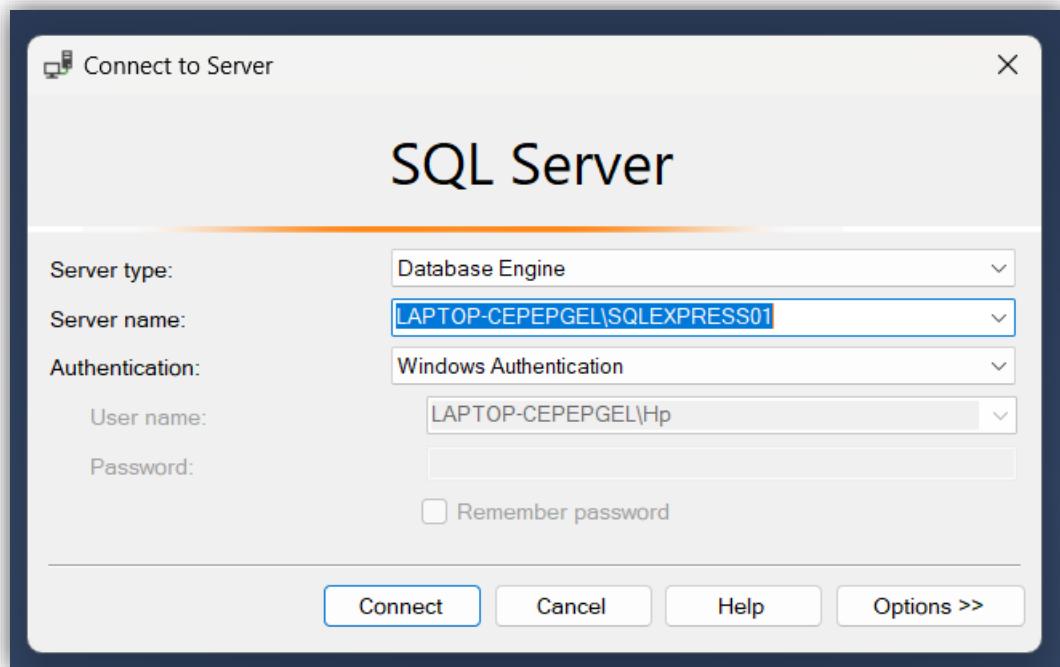


Sau khi cài đặt thành công sẽ nhận được thông báo:

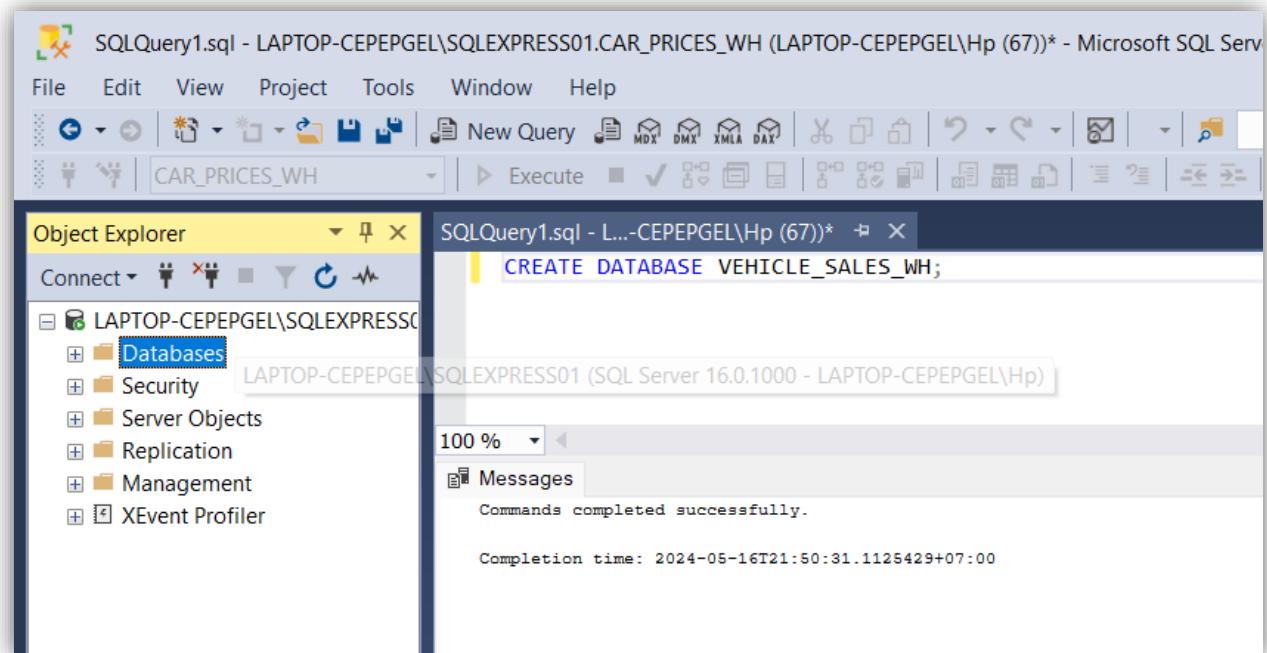


## 2.2. Chuẩn bị cơ sở dữ liệu

**Bước 1:** Mở SQL Server 2019 và kết nối với server bằng tài khoản user của window (Windows Authentication).

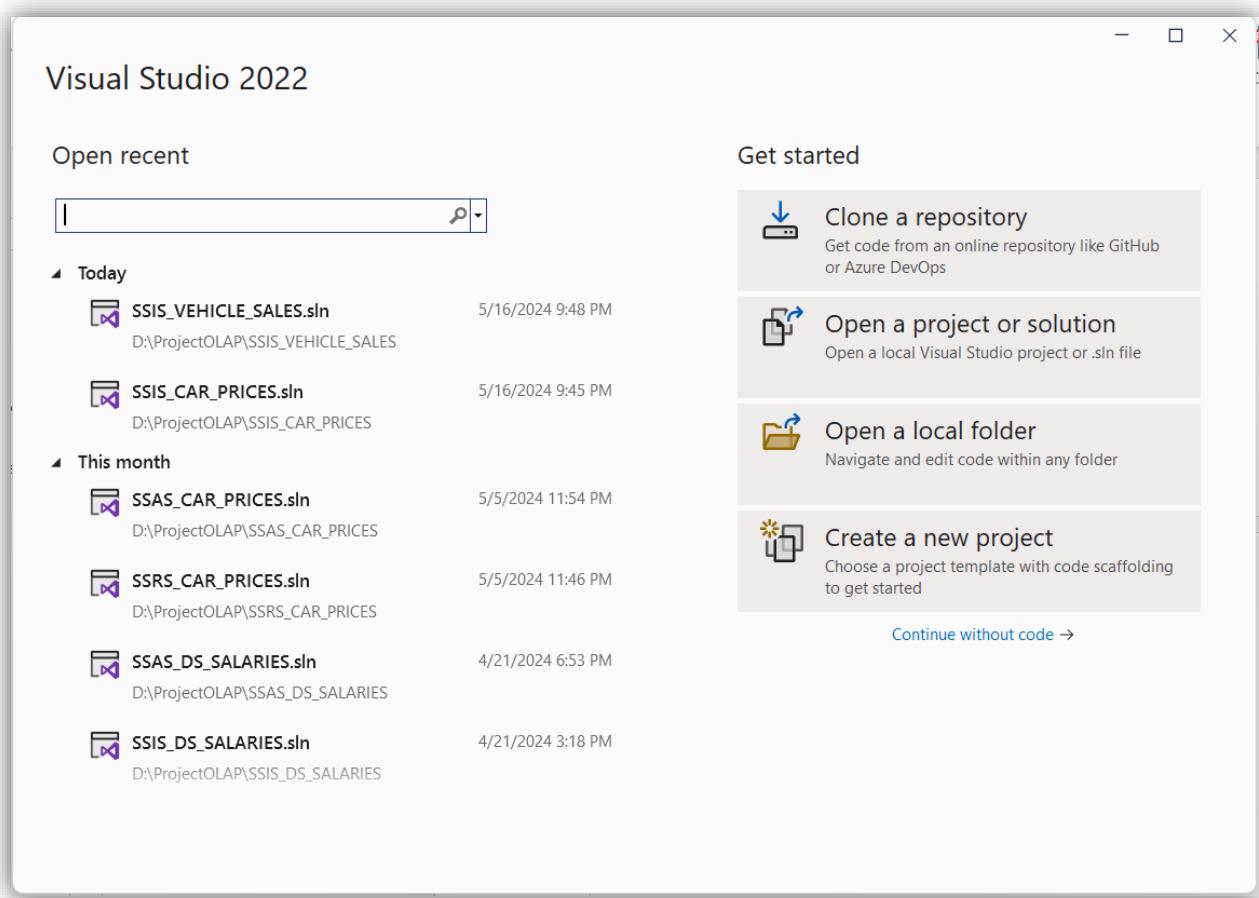


**Bước 2:** Khởi tạo một cơ sở dữ liệu có tên DS\_SALARIES\_WH, đây là nơi lưu các bảng Dim và bảng Fact cùng dữ liệu của các bảng đó.

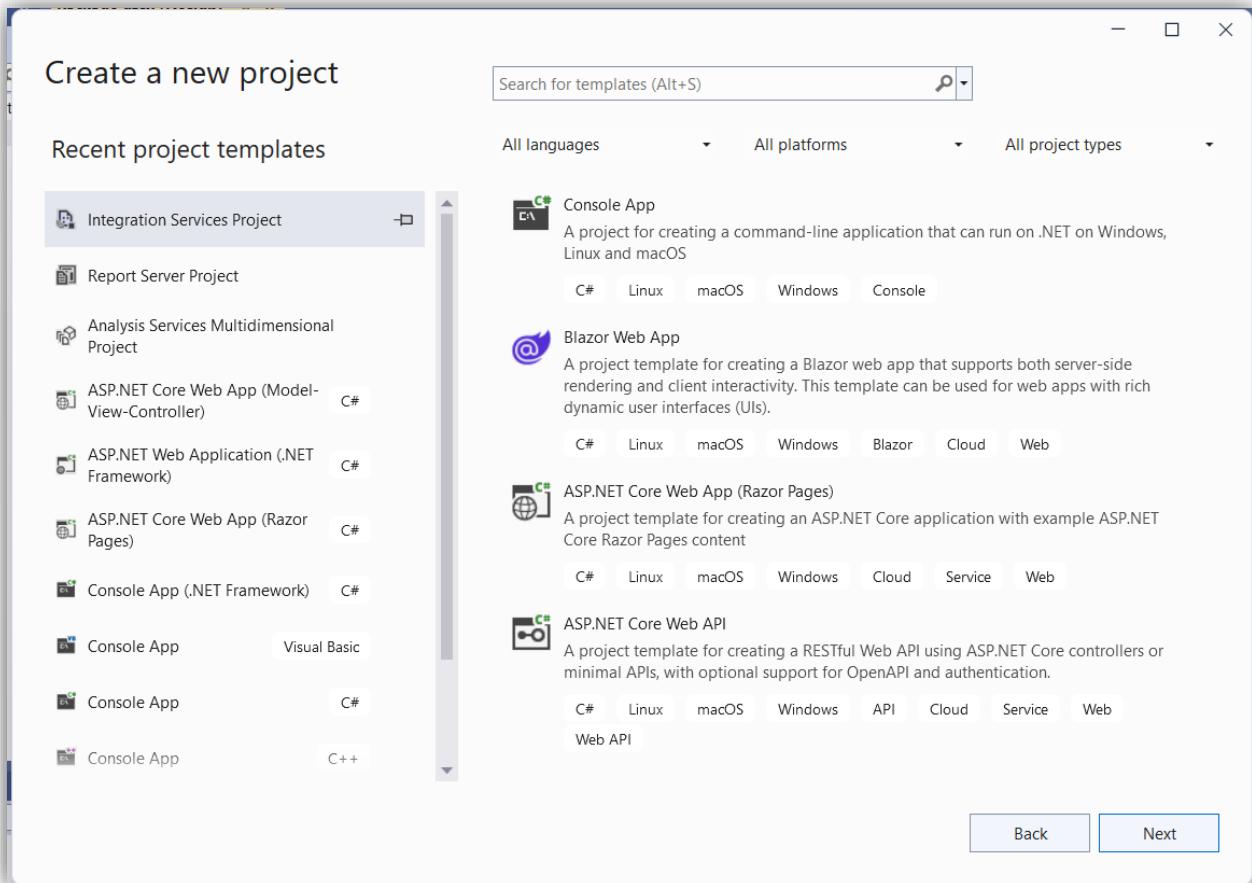


### 2.3. Tạo mới project SSIS

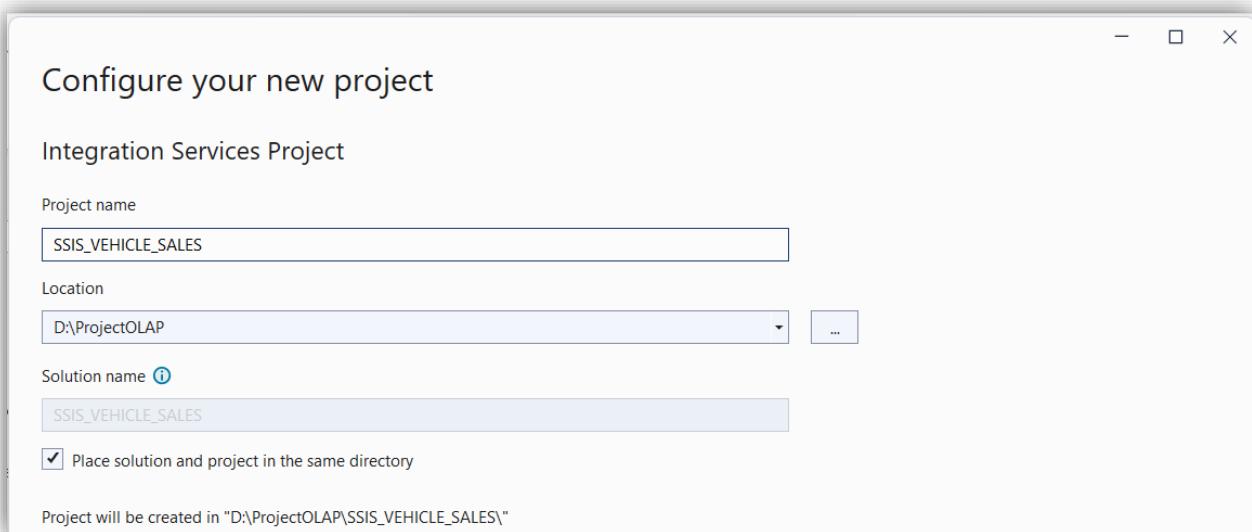
**Bước 1:** Mở Visual Studio 2022 và chọn “Create a new project”.



**Bước 2:** Chọn Integration Services Project và chọn Next



**Bước 3:** Đặt tên và thiết lập đường dẫn cho Project. Sau đó chọn Create.

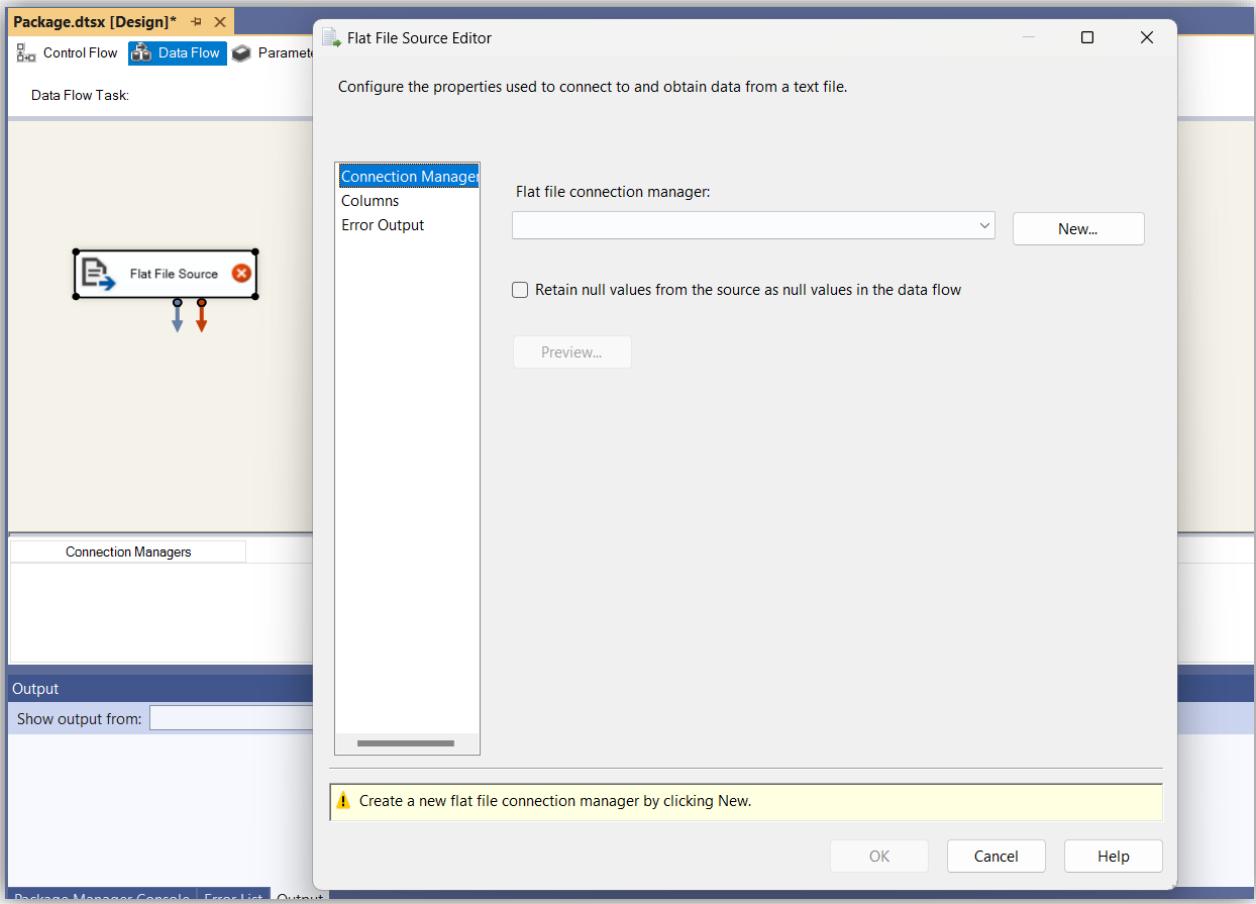


## 2.4. Tạo bảng Dim và bảng Fact

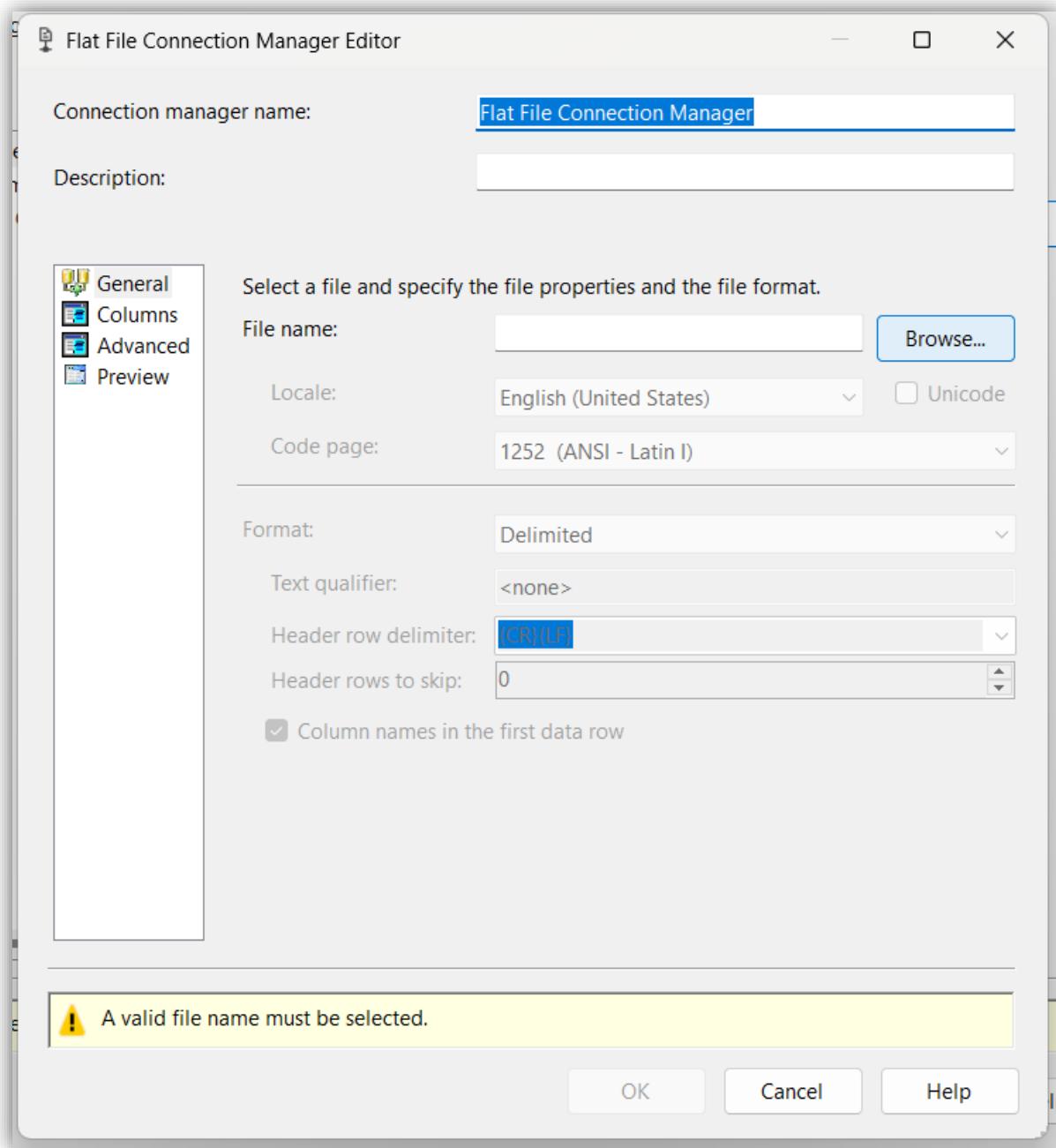
Trước khi tiến hành chia Dimension và bảng Fact, ta cần load dữ liệu gốc từ file

.csv vào Data Flow:

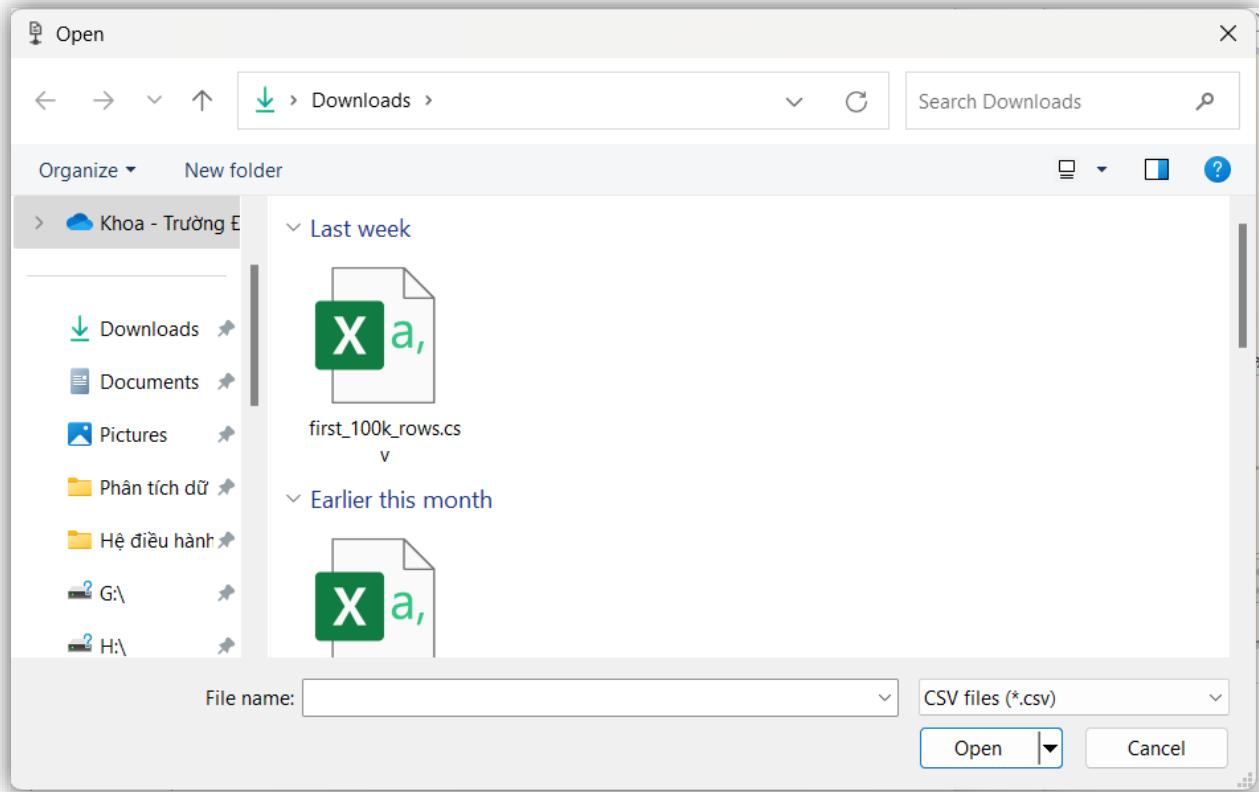
**Bước 1.** Trong Data Flow, tạo một đối tượng Flat File Source để lấy dữ liệu gốc từ file .csv. Chọn New để tạo một Flat File Connection Manager



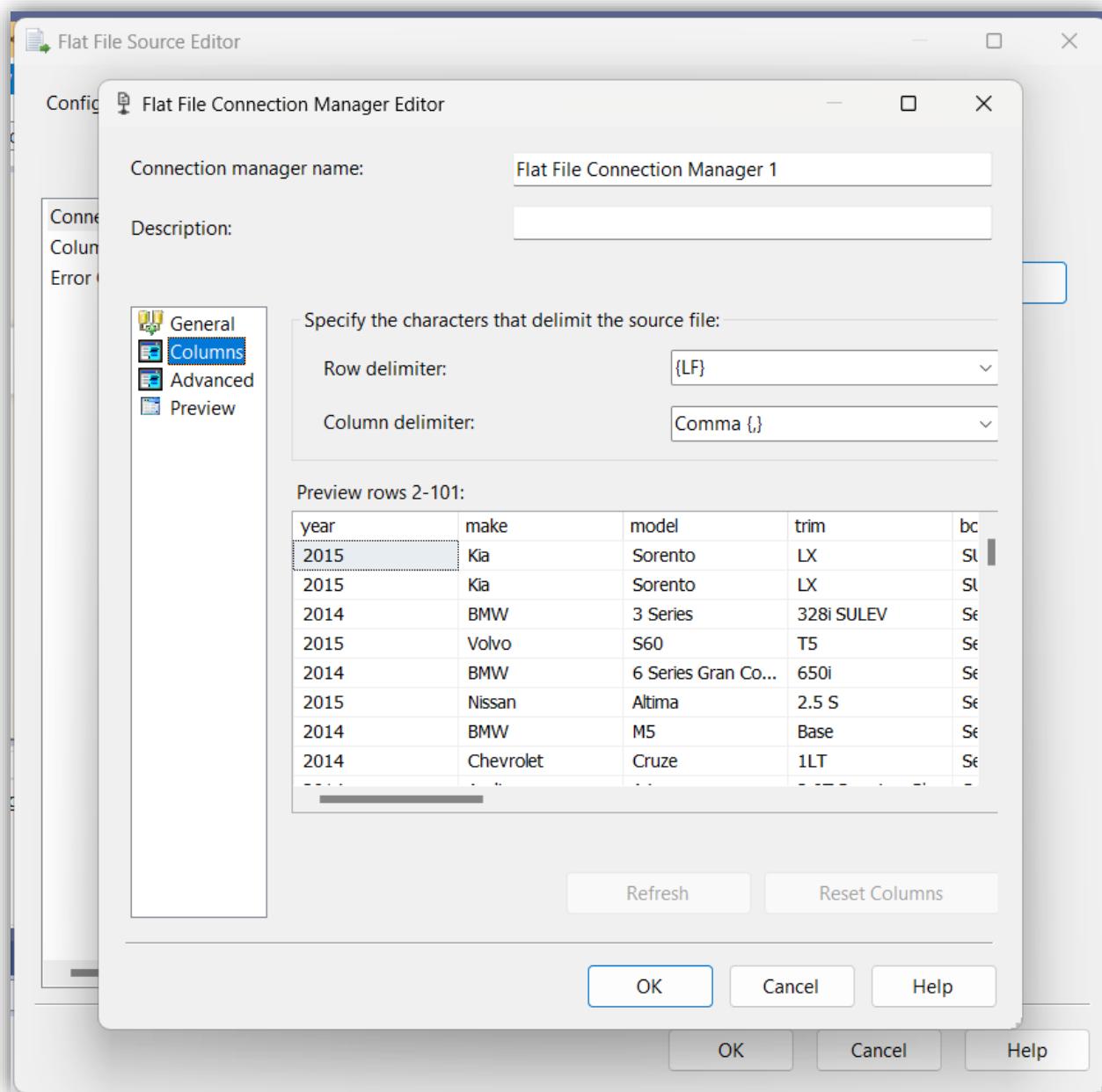
**Bước 2.** Chọn nút Browse để tải lên file dữ liệu gốc lưu trong máy



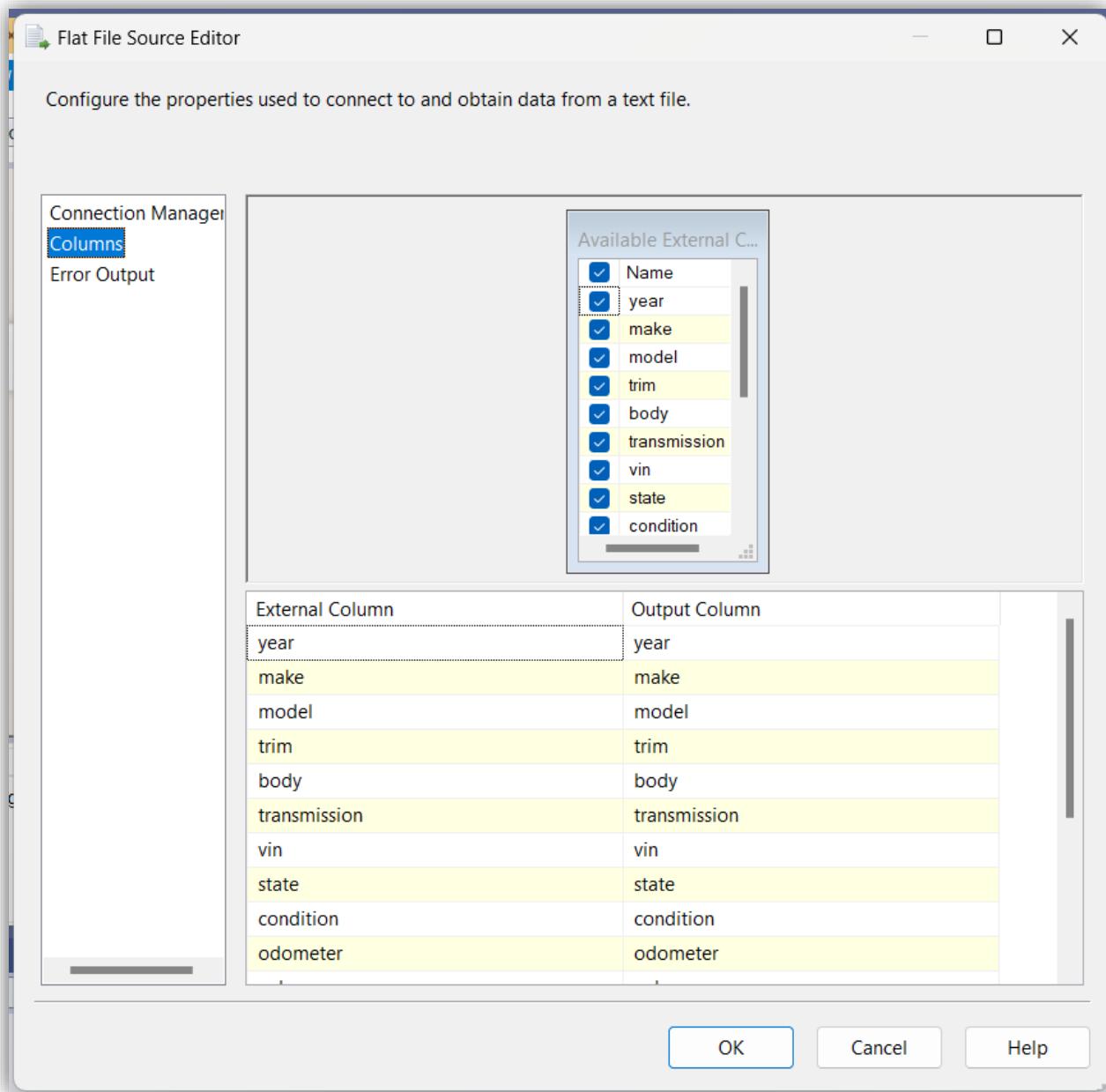
**Bước 3.** Chọn file dữ liệu .csv và chọn Open



**Bước 4.** Xem lại các cột dữ liệu trong file dữ liệu đã được tải lên ở menu Columns

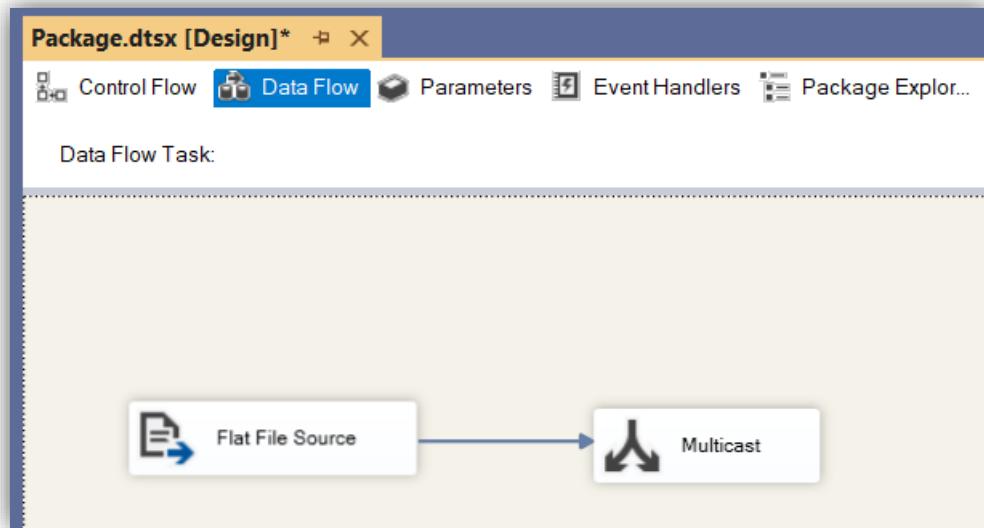


**Bước 5.** Click chọn OK và kiểm tra lần nữa các cột dữ liệu ở dạng danh sách. Nhấn OK lần nữa để tiến hành hoàn tất quá trình load dữ liệu vào Flat File Source



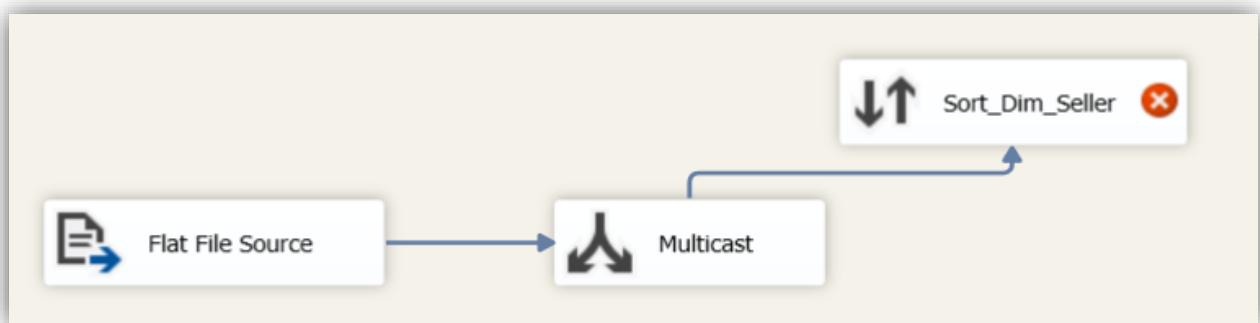
**Bước 6.** Tạo Multicast để phân tán dữ liệu từ Flat File Source đến các Dimension.

Tiến hành kết nối Flat File Source và Multicast.

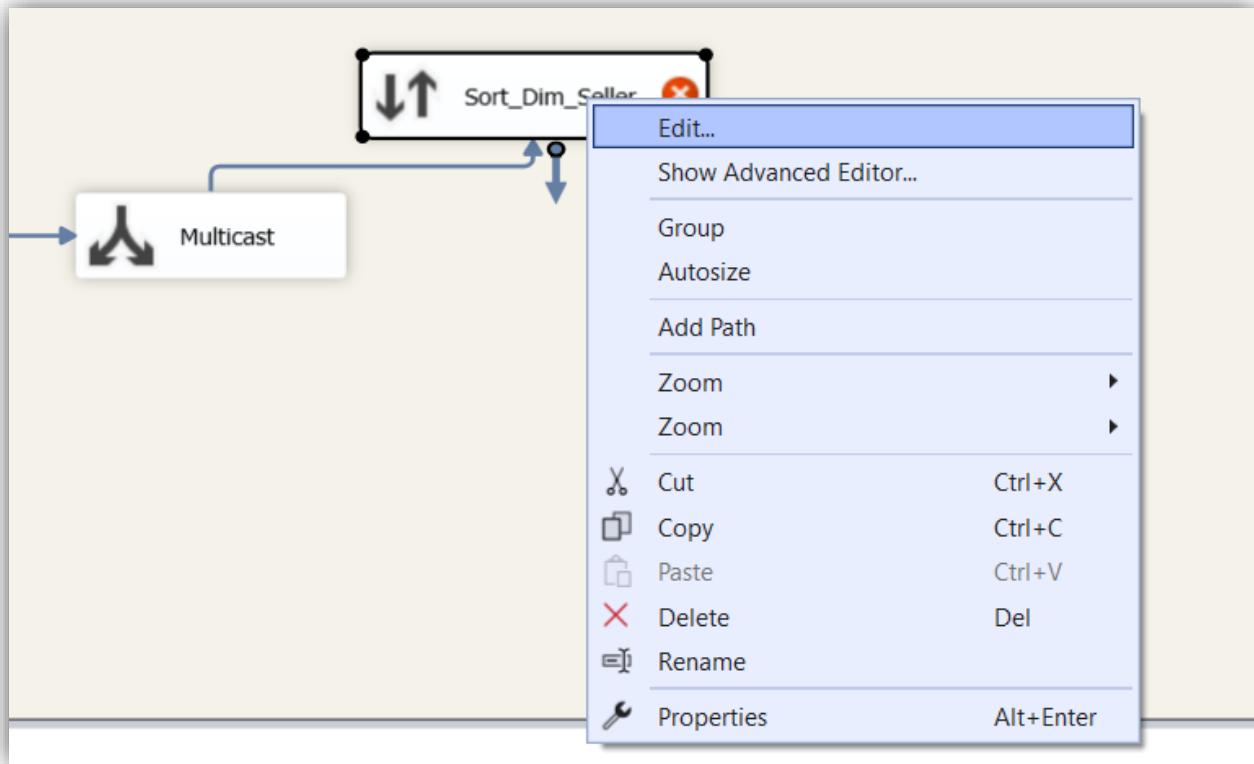


#### 2.4.1. Bảng Dim\_Seller

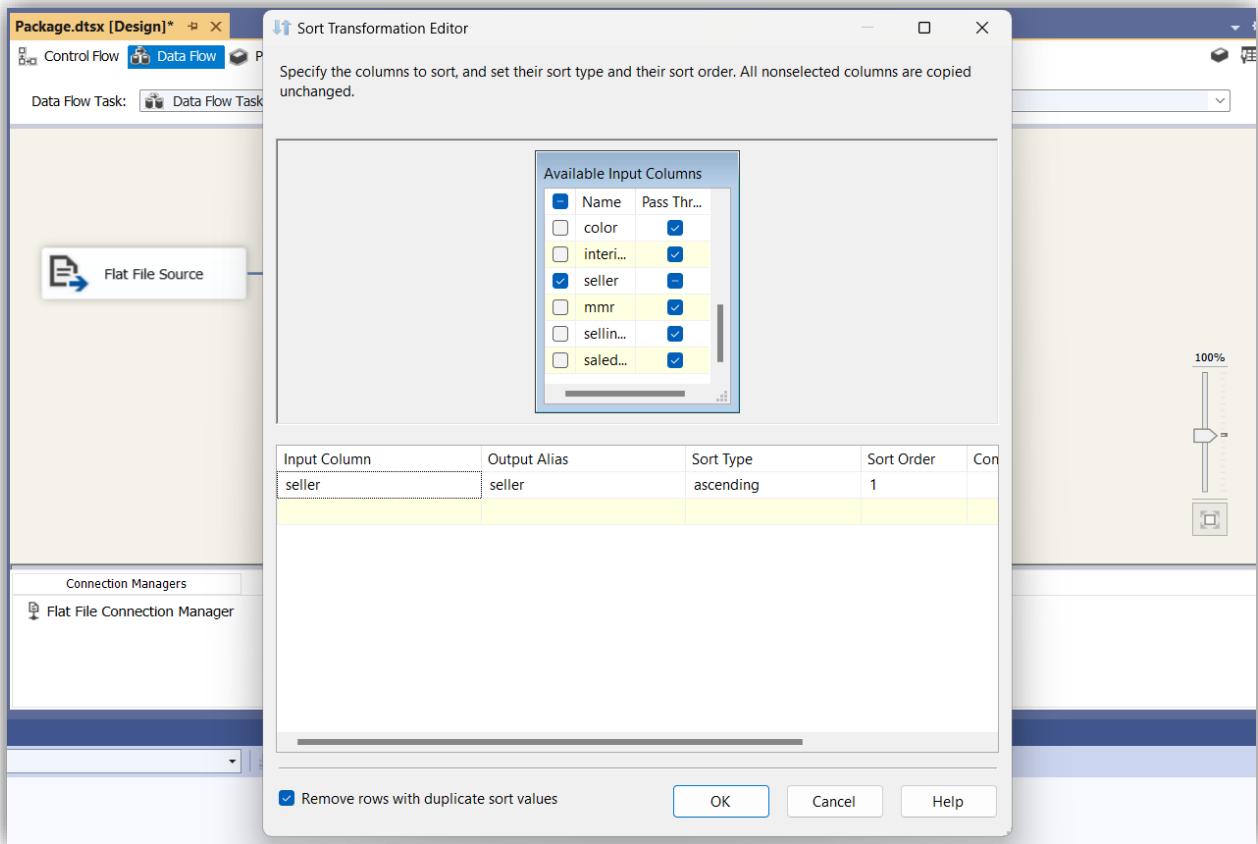
**Bước 1.** Chọn một Sort để tạo ra Sort\_Dim\_Seller cho Dim\_Seller



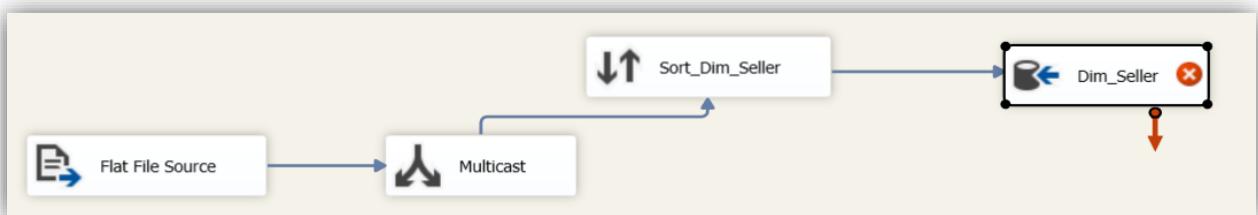
**Bước 2.** Click chuột phải vào Sort\_Dim\_Seller, chọn Edit: lần lượt chọn cột seller để đổ dữ liệu vào Sort\_Dim\_Seller



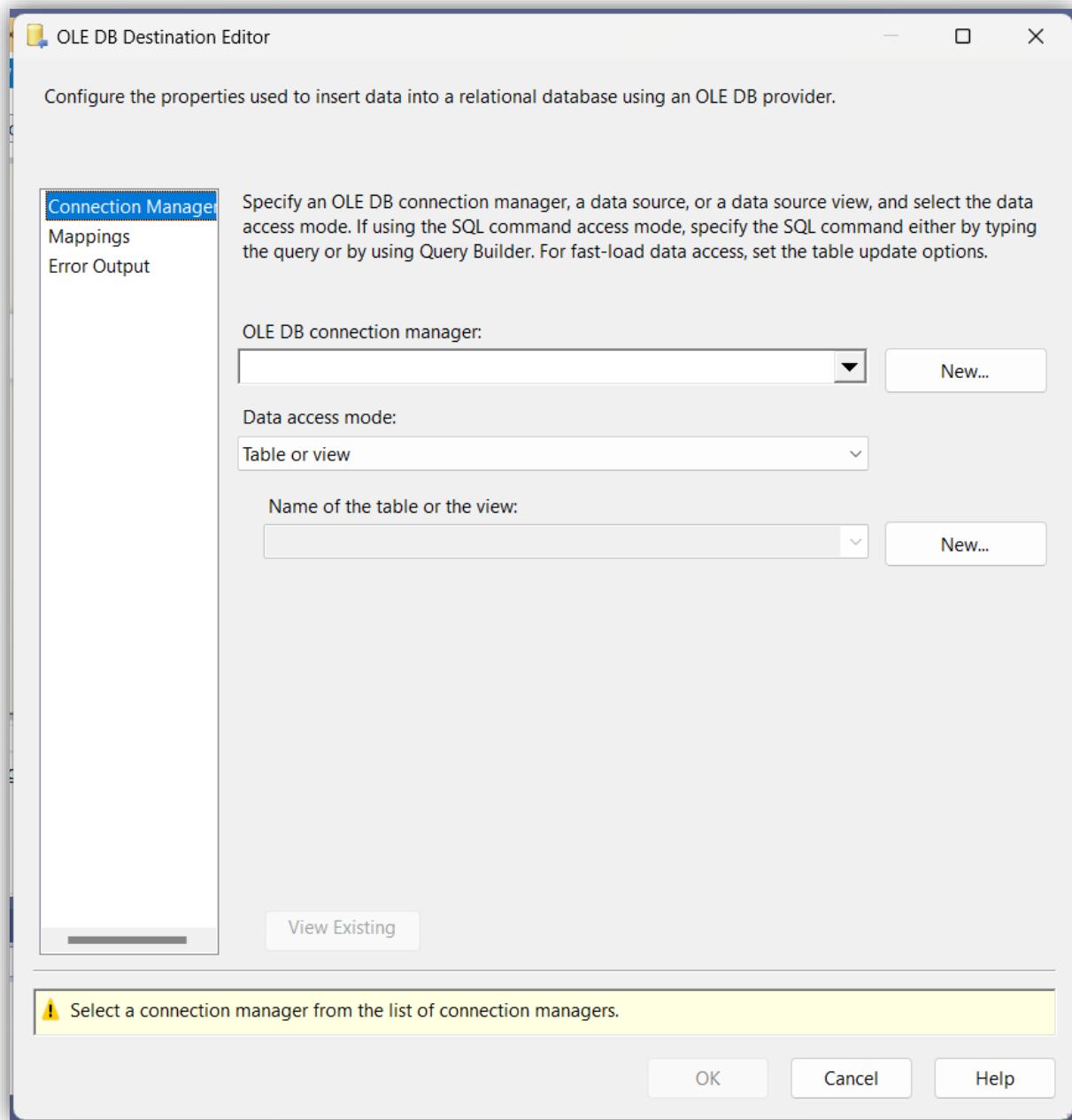
Tick chọn Remove rows with duplicate sort values xóa đi các dòng dữ liệu trùng nhau và sau đó chọn OK.



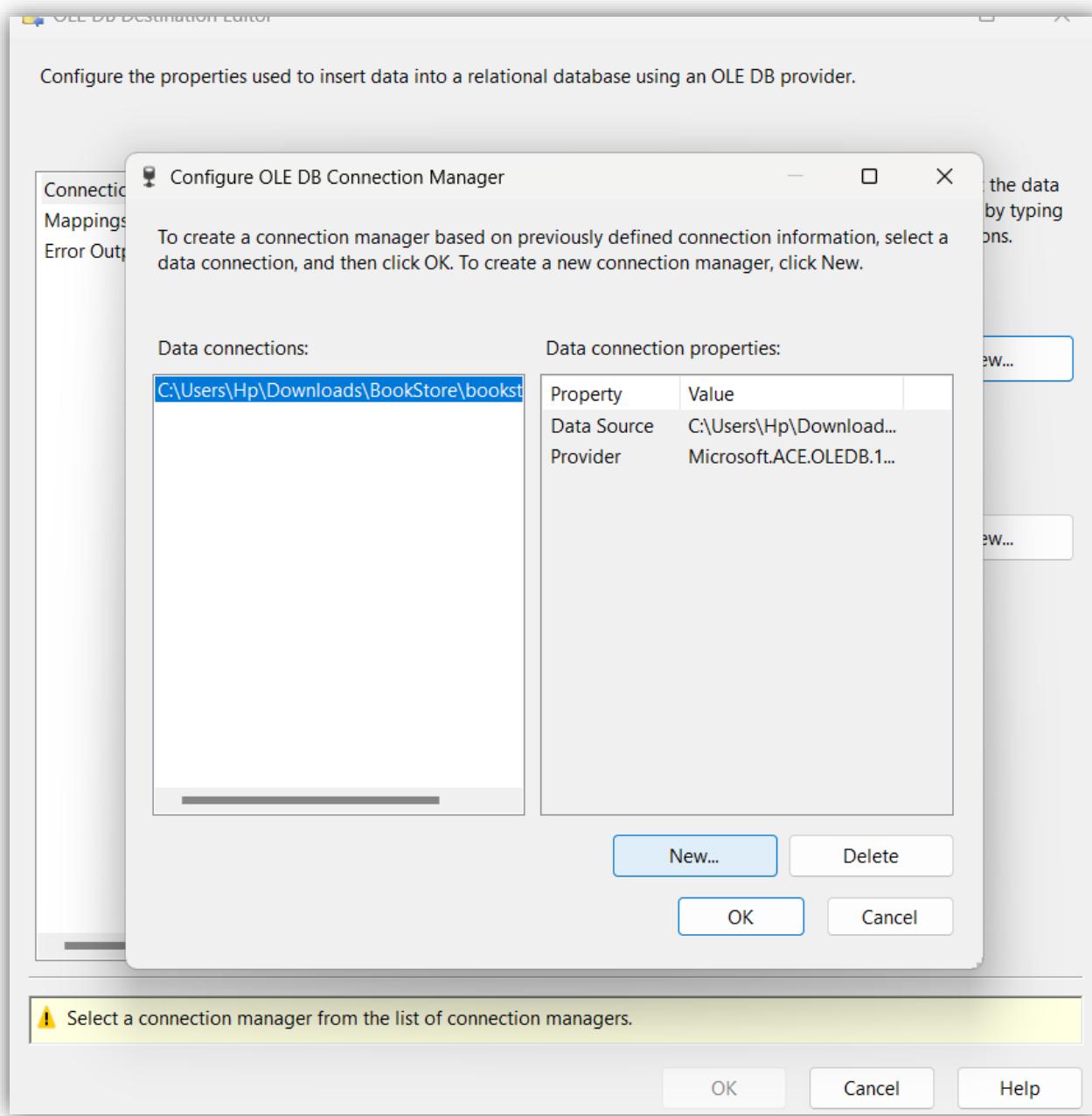
**Bước 3.** Tạo mới một OLE DB Destination để đỗ dữ liệu gốc sau khi đã được xử lý vào trong kho dữ liệu



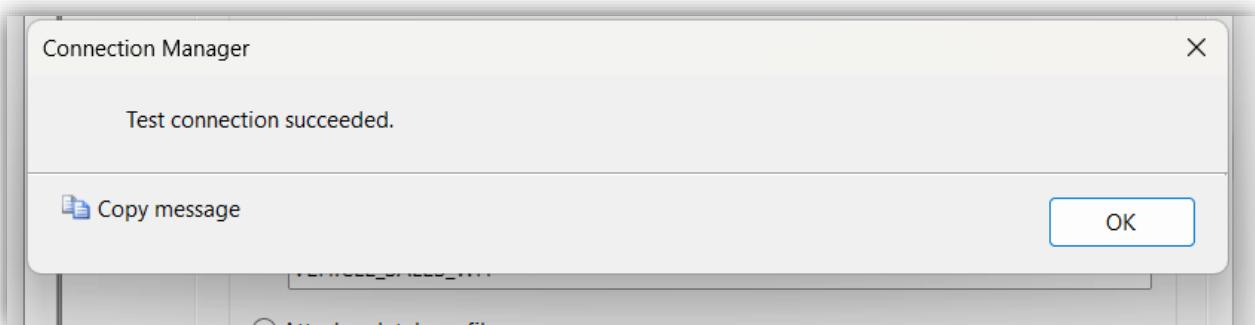
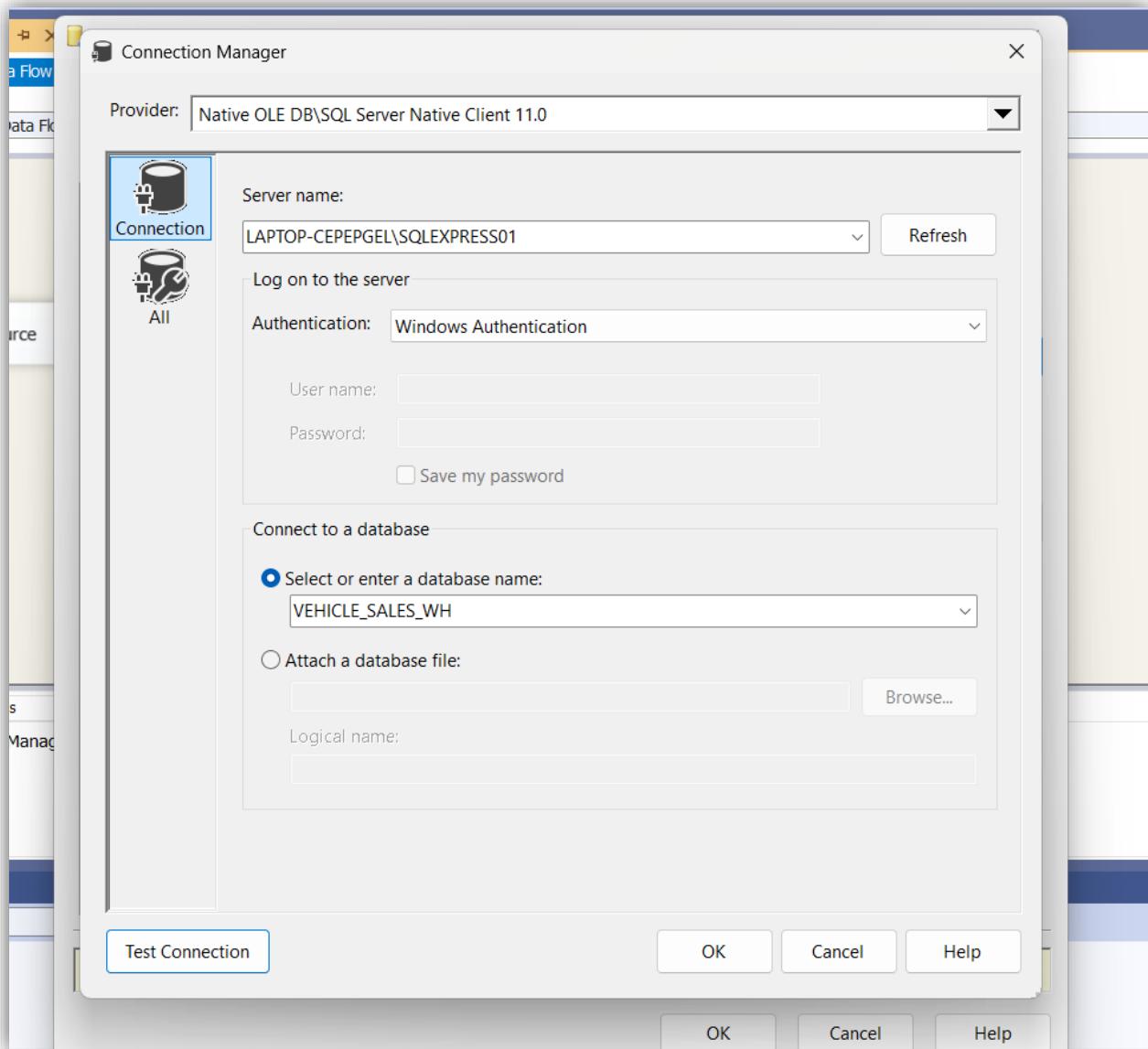
**Bước 4.** Tạo Dim\_Seller từ một OLE DB Destination. Double click vào OLE DB Destination này để tạo một connection mới đến MS SQL Server.



Tiếp tục chọn New... để tạo một connection mới:

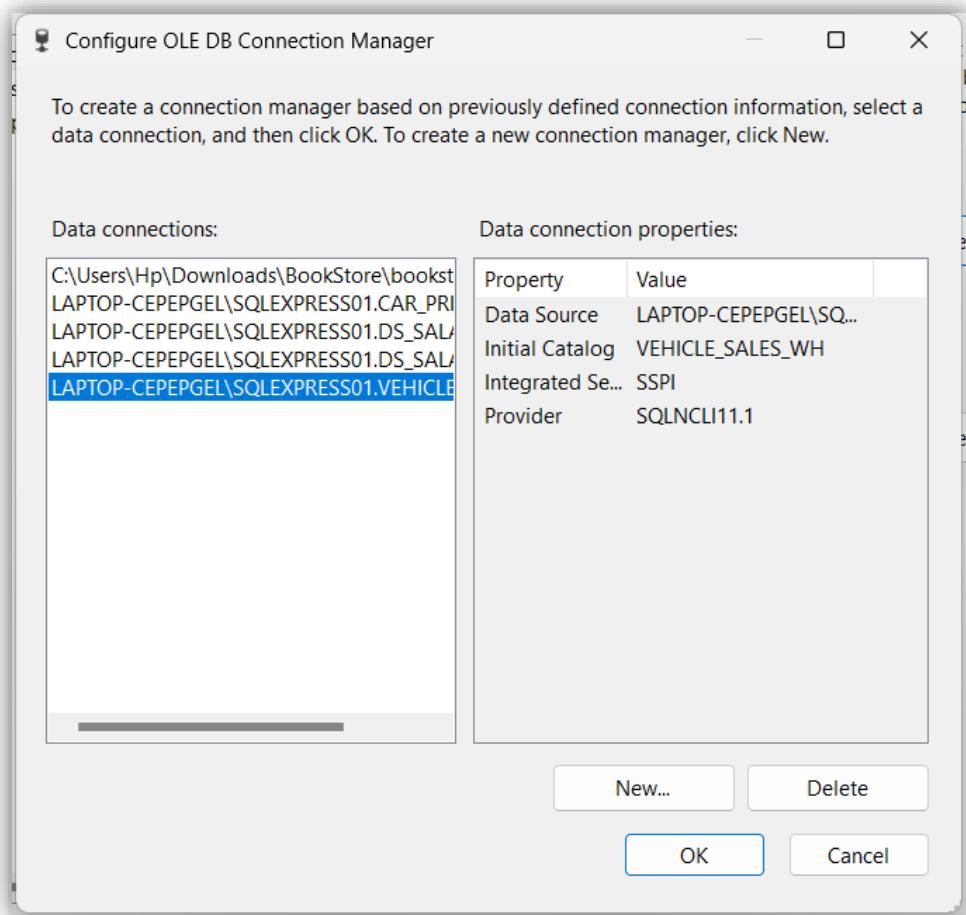


**Bước 5.** Chọn tên server name trùng với server name MS SQL Server để ta có thể kết nối đến datawarehouse VEHICLE\_SALES\_WH vừa tạo. Kết nối đến server bằng tài khoản window mặc định (Windows Authentication)  
 Nhấn Test Connection để kiểm tra kết nối

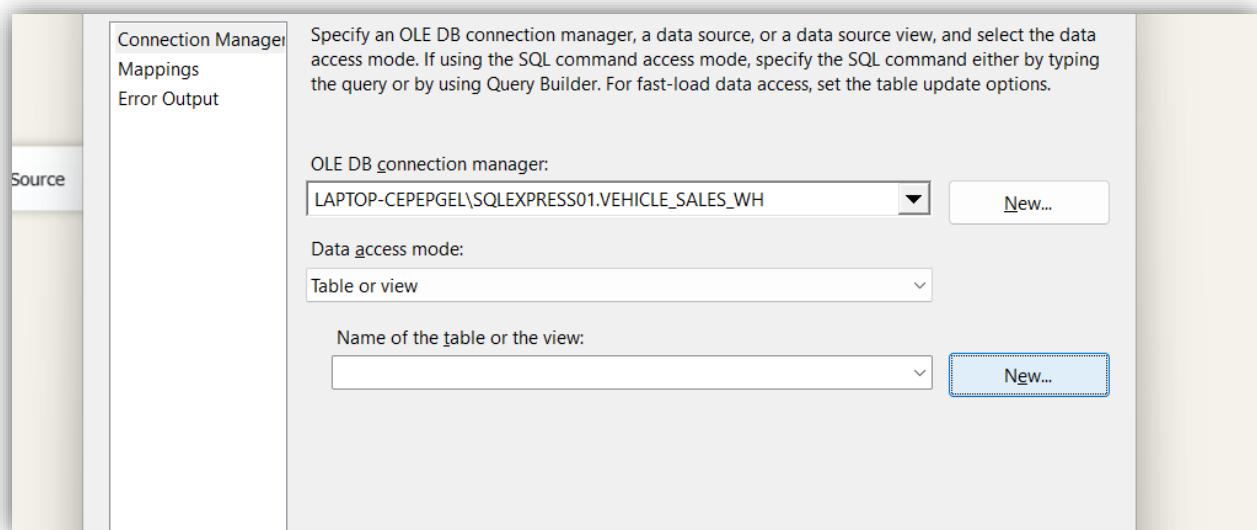


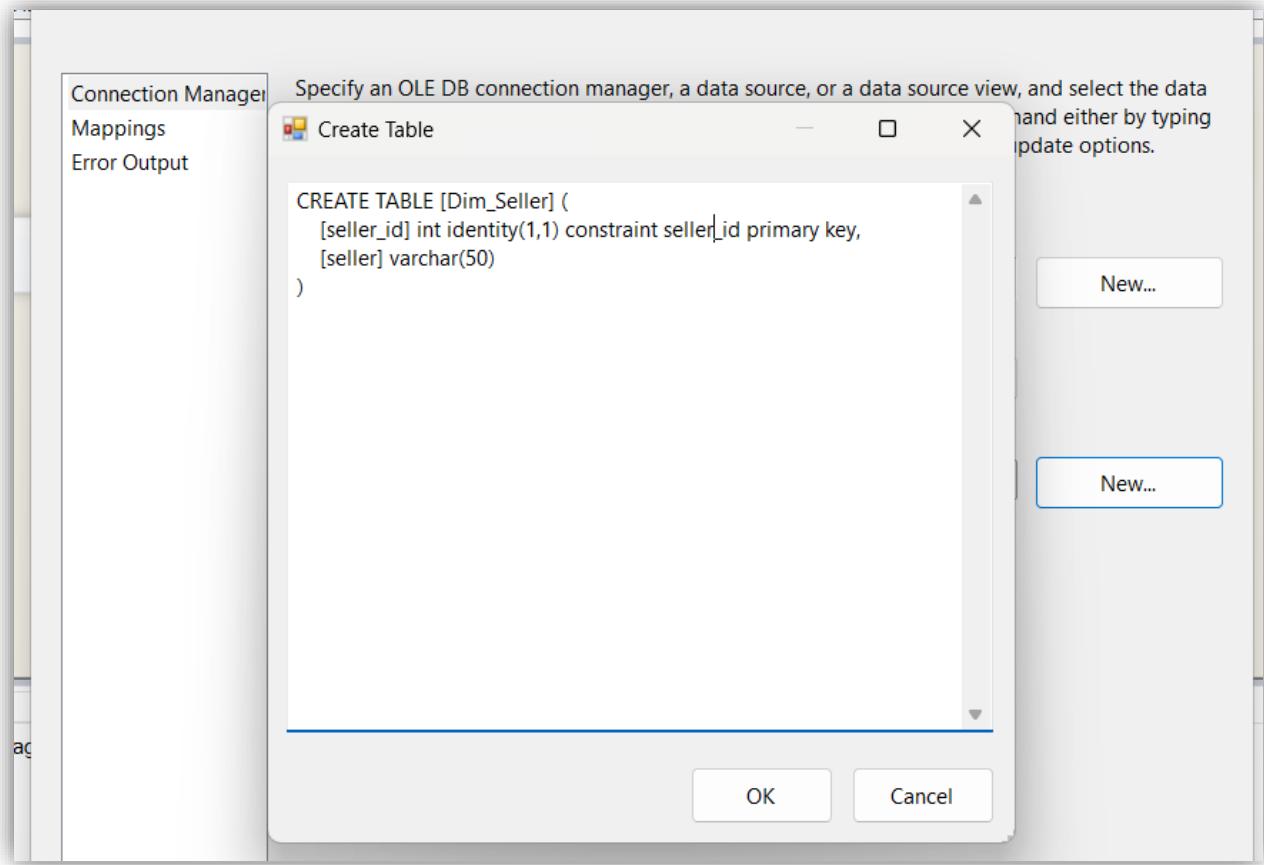
Ta tiến hành tạo bảng Dim\_Seller ở MS SQL Server

**Bước 6.** Chọn connection vừa tạo đến MS SQL Server và nhấn OK.



### Bước 7. Chọn New.. để tạo mới bảng

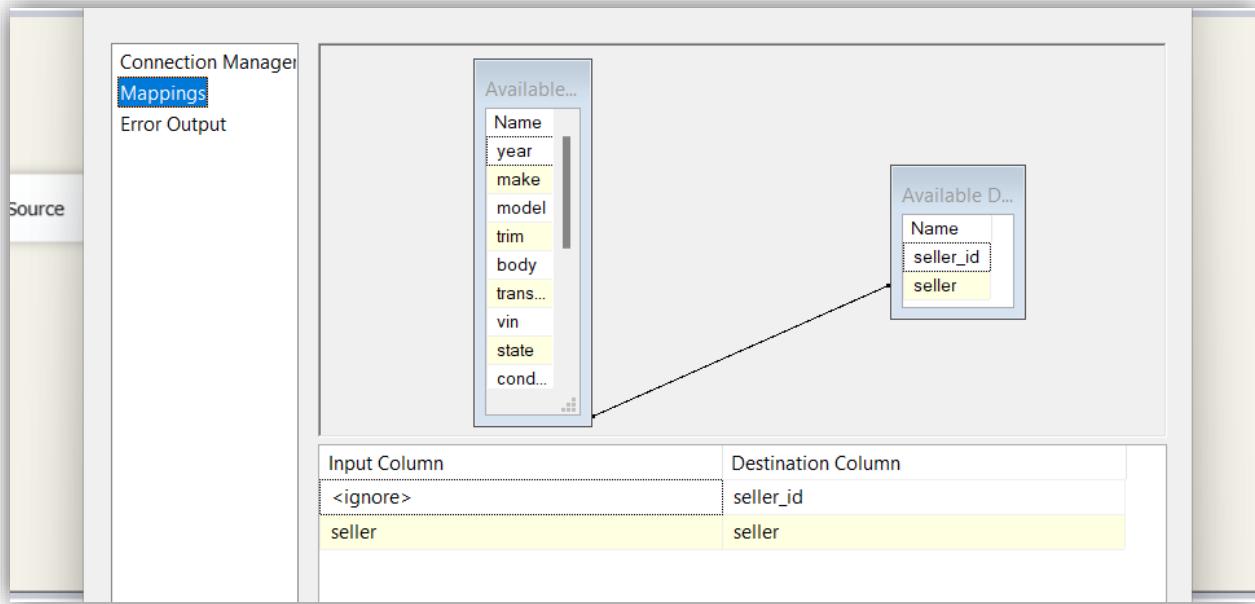




Nội dung câu lệnh SQL tạo bảng Dim\_Company như sau:

```
CREATE TABLE [Dim_Seller] (
    [seller_id] int identity(1,1) constraint seller_id primary key,
    [seller] varchar(50)
)
```

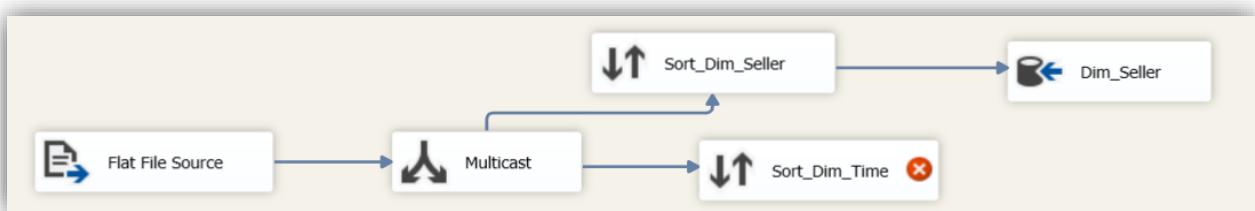
**Bước 8.** Tiếp đến ta cần chọn mục Mappings để xem xét việc ánh xạ các cột dữ liệu



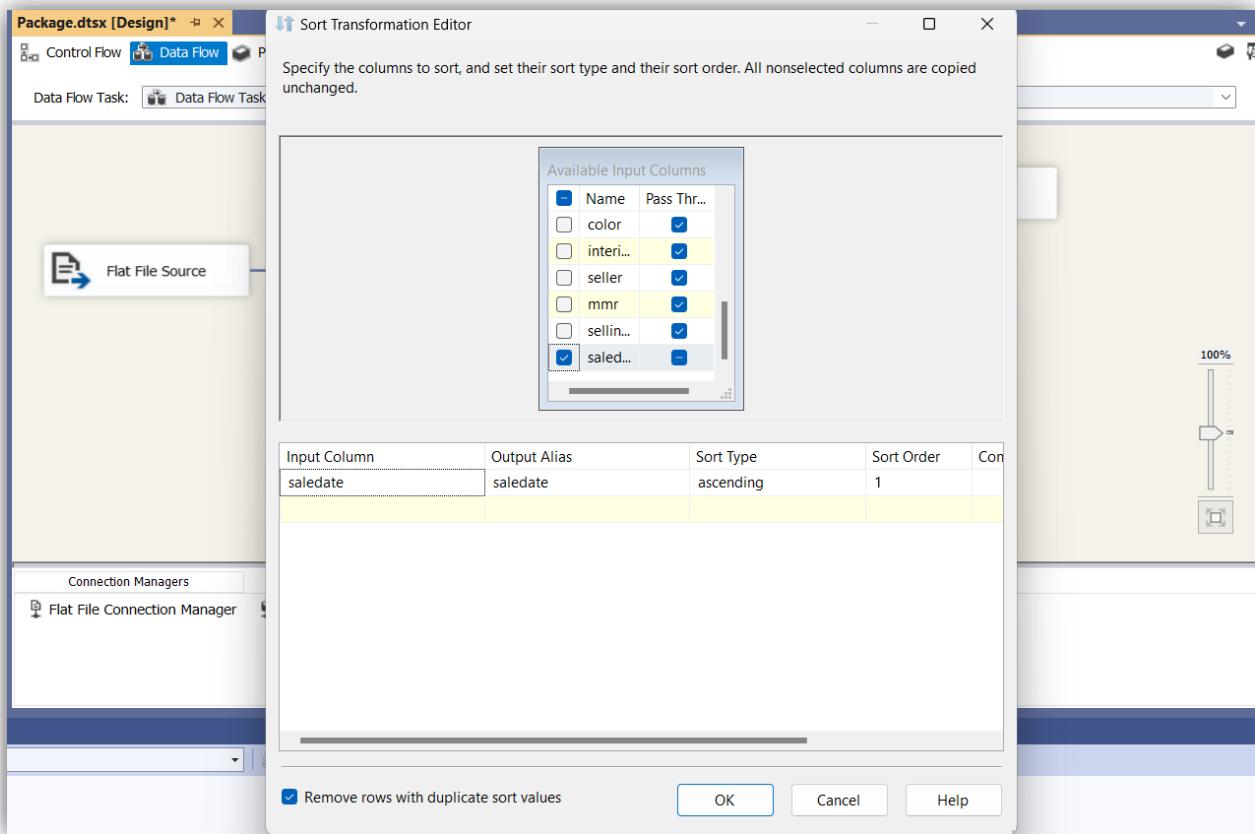
Chọn OK để hoàn tất thiết lập.

#### 2.4.2. Bảng Dim\_Time

**Bước 1.** Chọn một Sort để tạo ra Sort\_Dim\_Time cho Dim\_Time

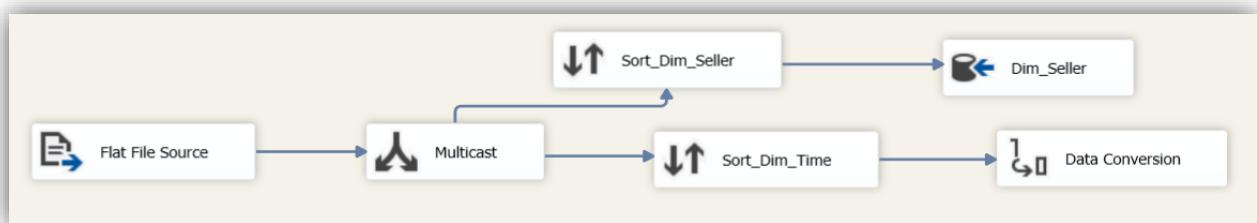


Click chuột phải vào Sort\_Dim\_Time, chọn Edit: lần lượt chọn cột saledate làm các cột để đổ dữ liệu vào Sort\_Dim\_Time



Tick chọn Remove rows with duplicate sort values xóa đi các dòng dữ liệu trùng nhau và sau đó chọn OK.

**Bước 2.** Kiểu dữ liệu DateTime khi lấy từ dữ liệu gốc sẽ được mặc định là kiểu string. Ta dùng Data Conversion để chuyển đổi kiểu string về dạng DateTime.

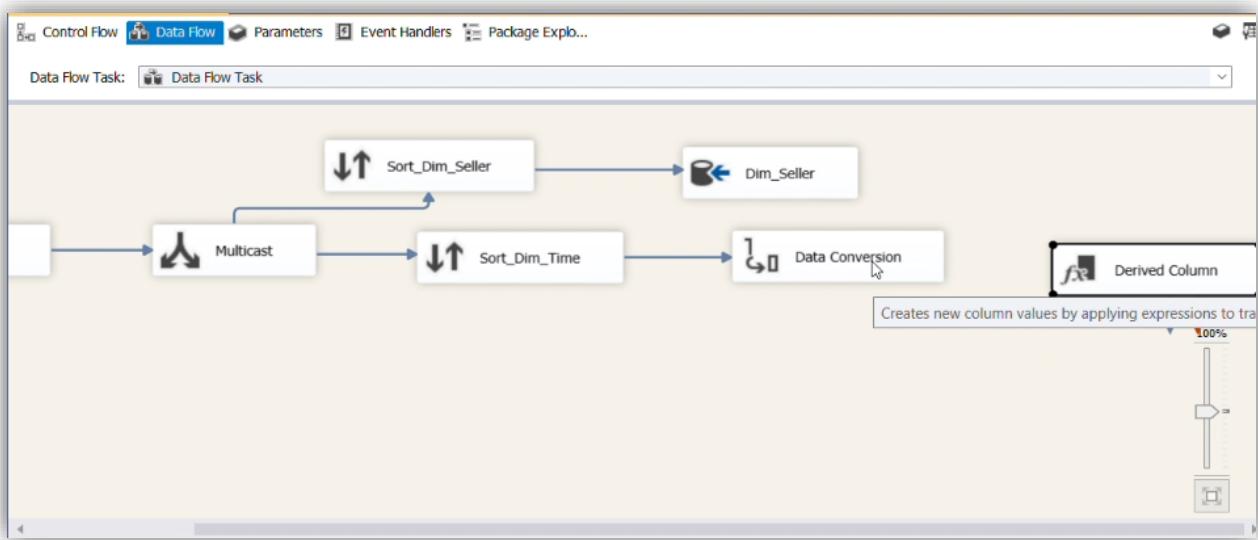


**Bước 3.** Click chuột phải vào Data conversion này và chọn Edit, click chọn cột saledate, ở cột Data Type ta thấy mặc định kiểu dữ liệu là string, ta chọn lại kiểu dữ liệu cho cột này date[DT\_DATE]

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
saledate	Copy of saledate	string [DT_STR]	50			1252 (ANSI)

**Bước 4.** Đặt lại alias cho saledate từ Copy of saledate thành saledate. Nhấn OK.

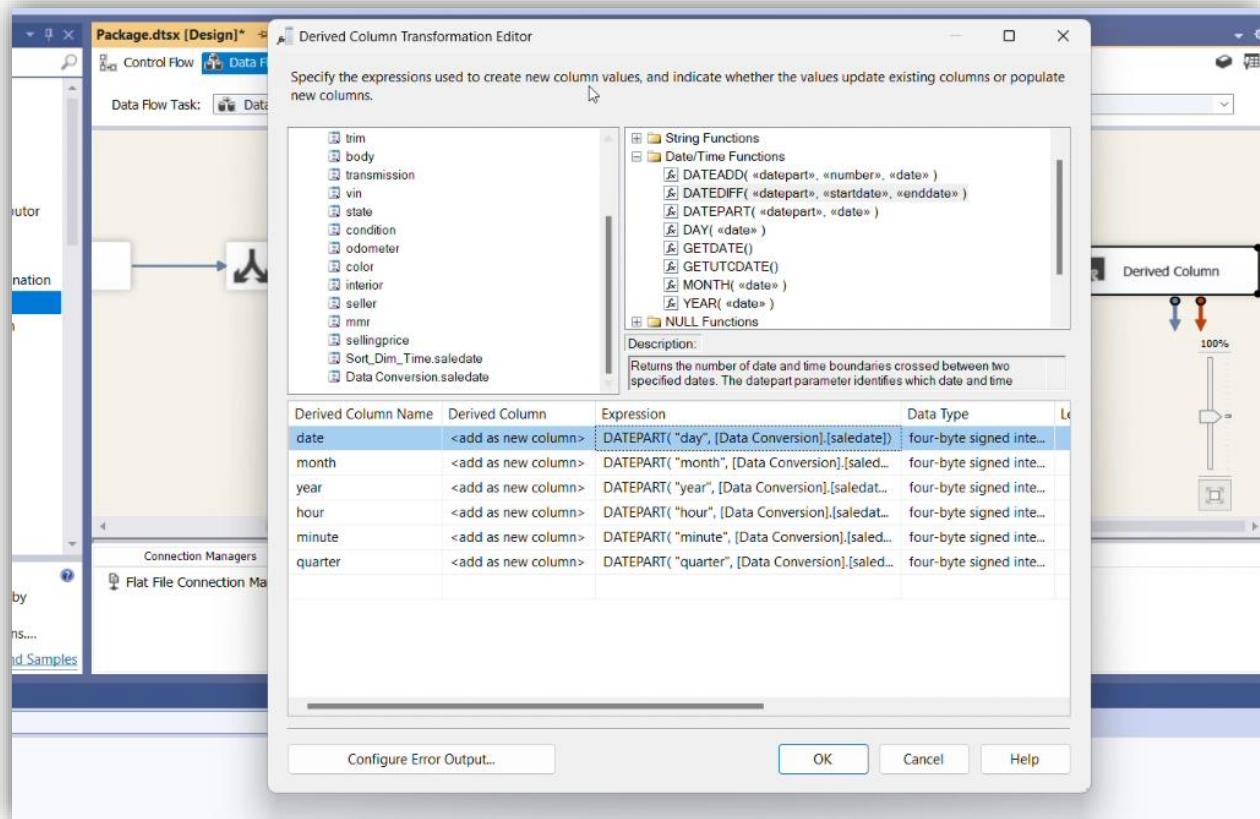
Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
saledate	saledate	date [DT_DATE]				

**Bước 5.** Thêm thành phần Derived Column và chọn Edit để chia cột dữ liệu

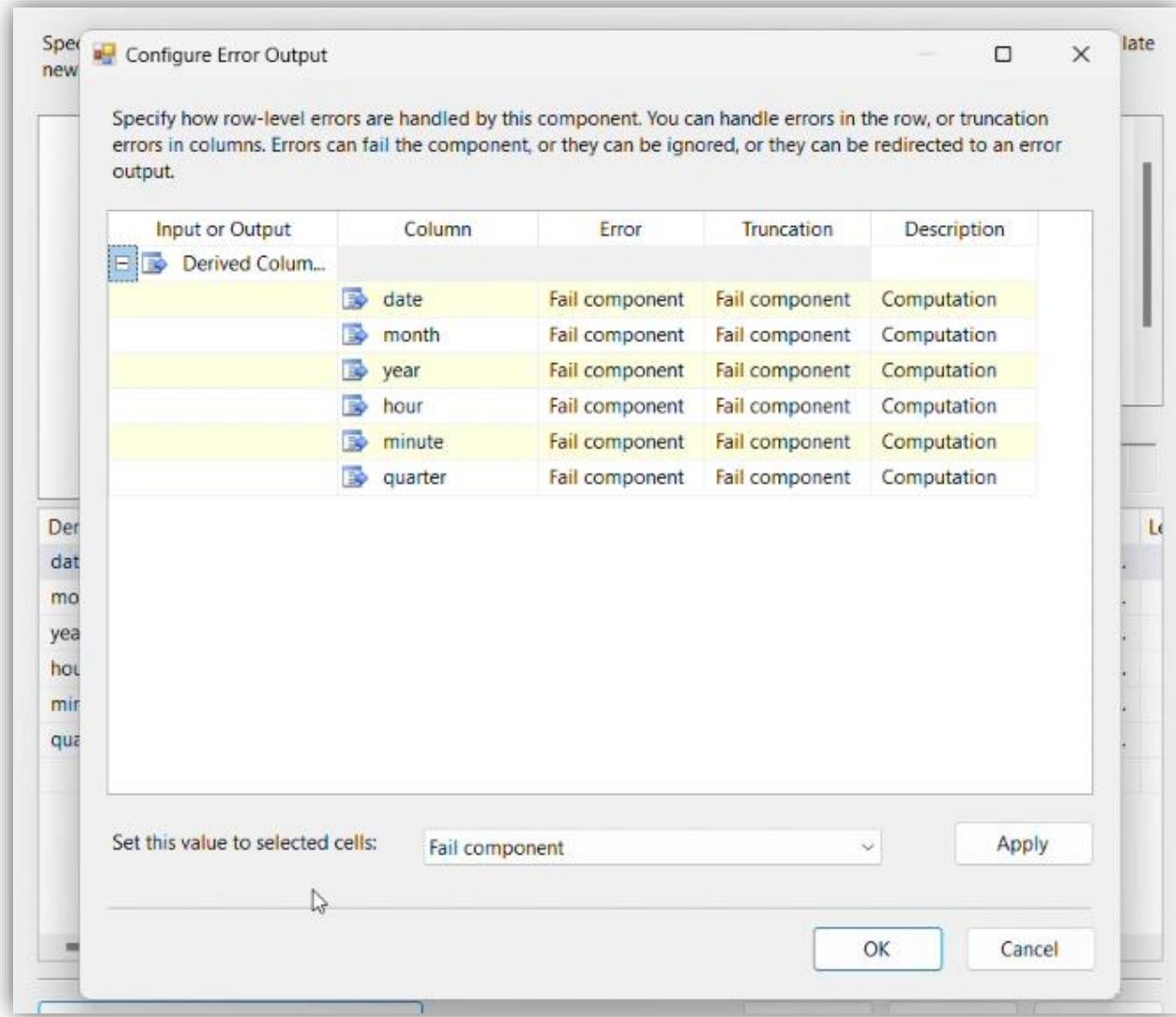
**Bước 6.** Chia dữ liệu từ cột saledate có kiểu dữ liệu dd/MM/yyyy HH:mm thành các cột date, month, year, hour, minute, quarter

Chọn phương thức DATEPART(<kiểu dữ liệu thời gian>, cột dữ liệu). Ở đây, ta chia saledate thành cột date nên ta cài đặt: DATEPART("day", [Data Conversation].[saledate]).

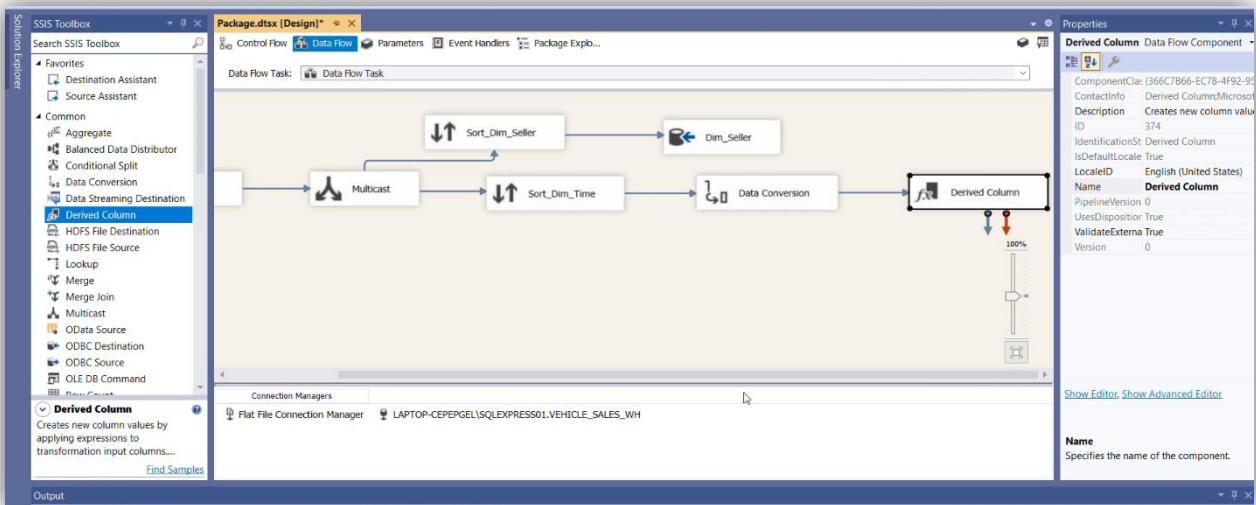
Tương tự chia saledate thành các cột date, month, year, hour, minute, quarter



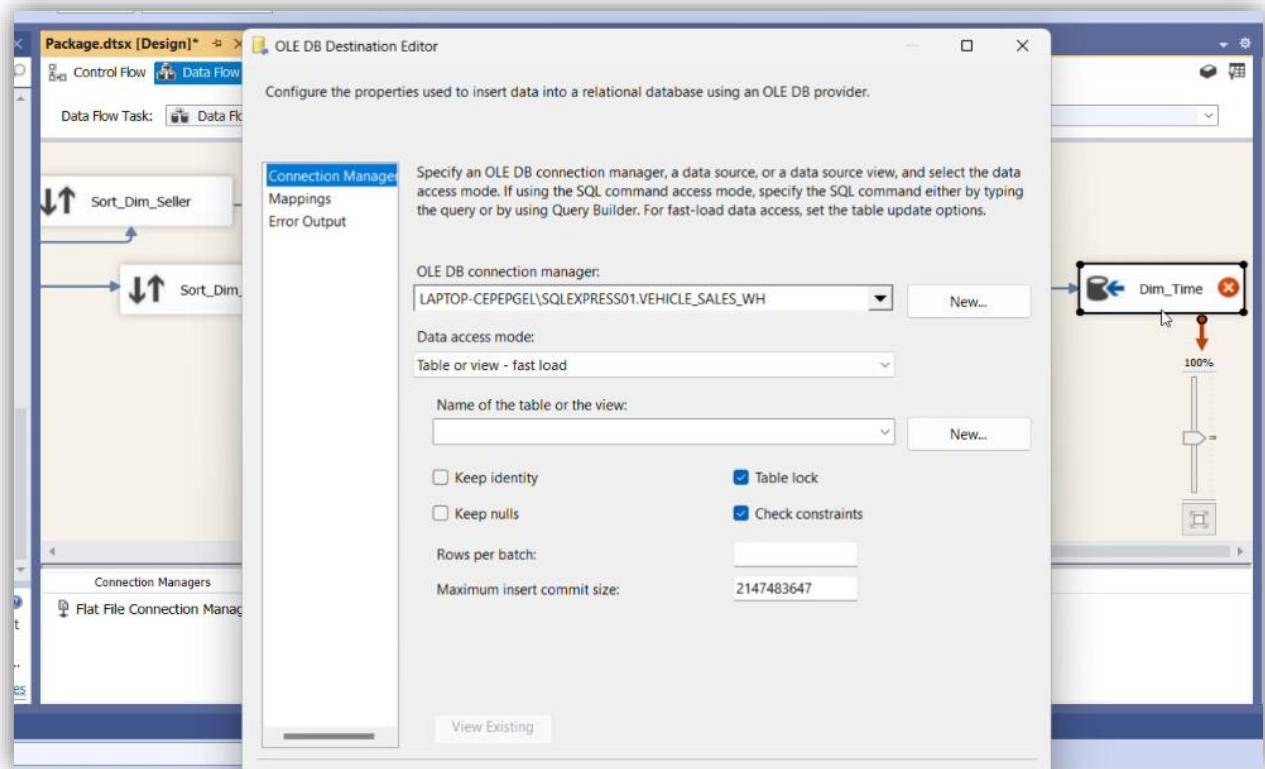
**Bước 7:** Nhấn nút Configure Error Output...



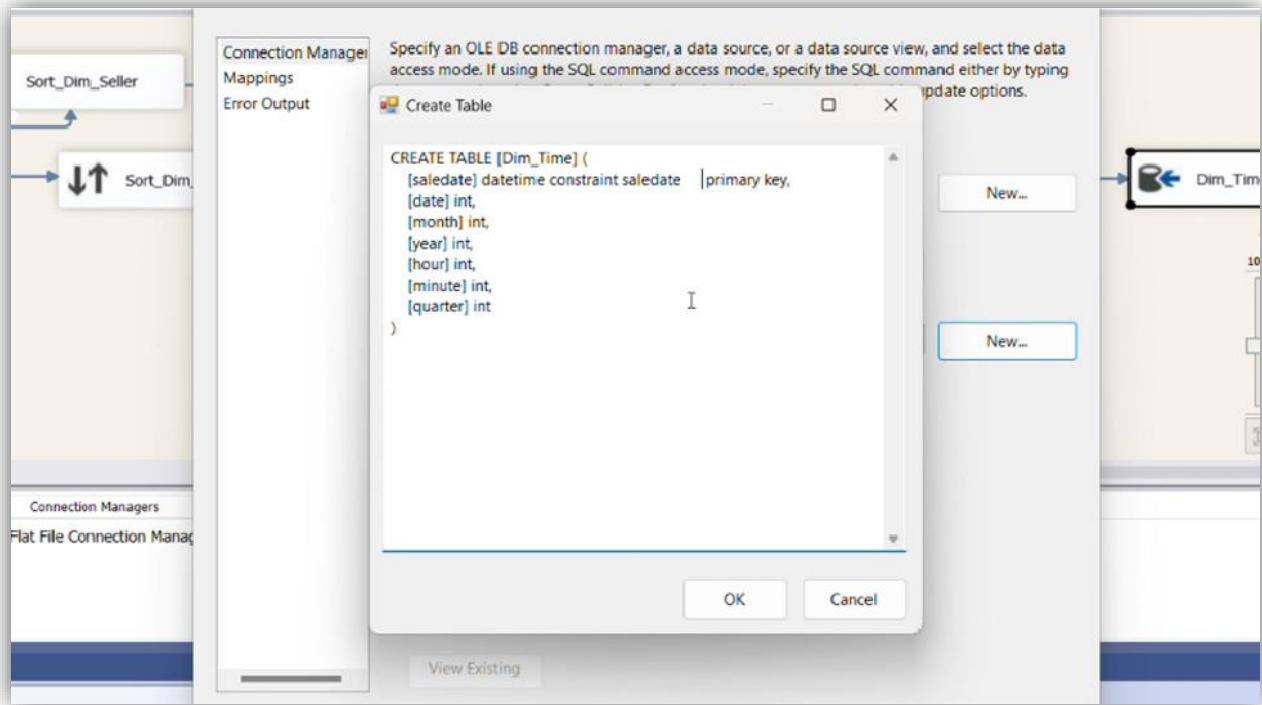
**Bước 8.** Ta thấy từ 1 cột saledate được chia thành 6 cột ứng với lược đồ dữ liệu bảng Dim\_Time. Nhấn OK để hoàn tất quá trình chia cột.



**Bước 9.** Tạo Dim\_Time từ một OLE DB Destination. Double click vào OLE DB Destination này để tạo một connection mới đến MS SQL Server.



**Bước 10.** Chọn New.. để tạo mới bảng



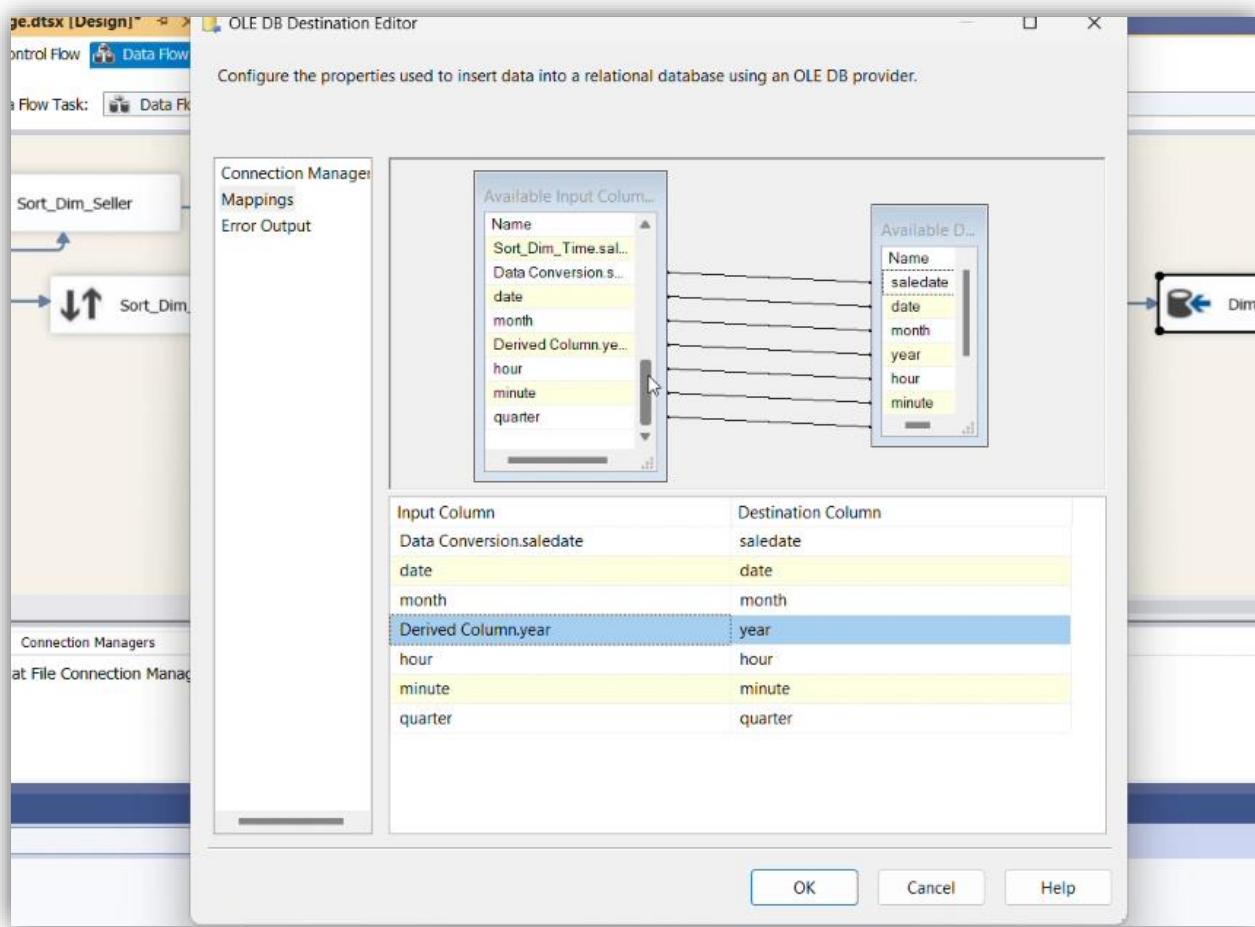
Nội dung câu lệnh SQL tạo bảng Dim\_Time như sau:

```

CREATE TABLE [Dim_Time] (
    [saledate] datetime constraint saledate primary key,
    [date] int,
    [month] int,
    [year] int,
    [hour] int,
    [minute] int,
    [quarter] int
)

```

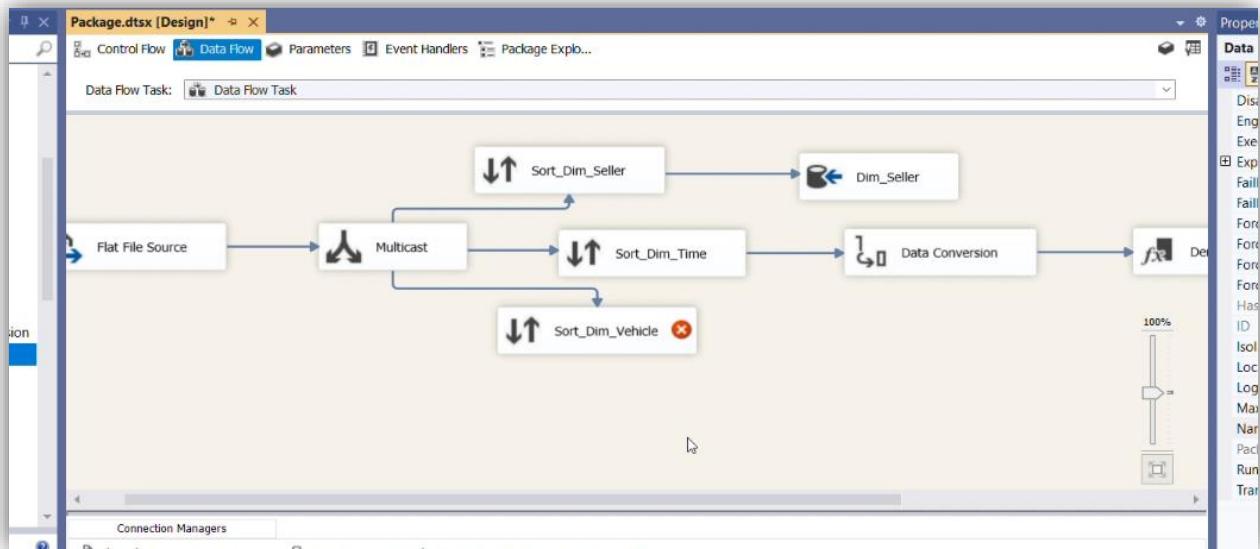
**Bước 11.** Tiếp đến ta cần chọn mục Mappings để xem xét việc ánh xạ các cột dữ liệu



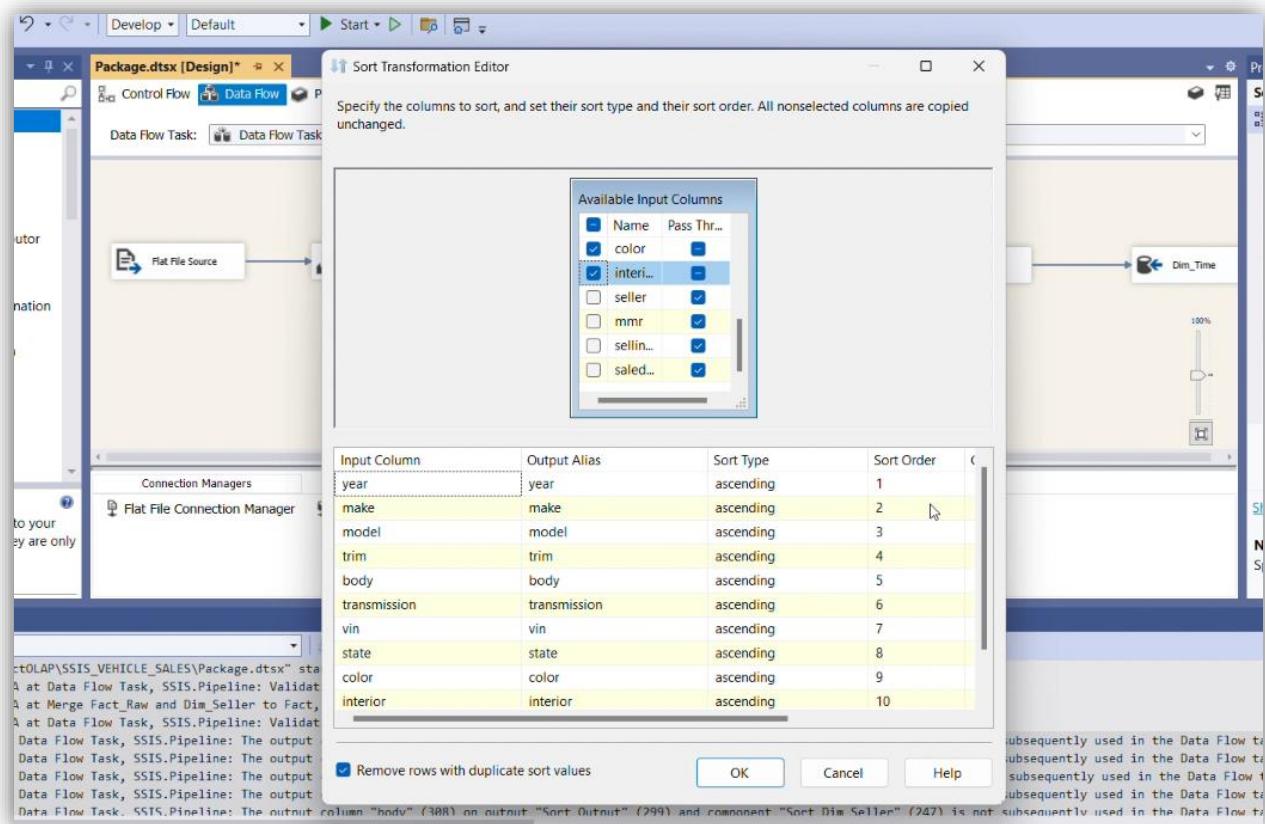
Chọn OK để hoàn tất thiết lập.

#### 2.4.3. Bảng Dim\_Vehicle

**Bước 1.** Chọn một Sort để tạo ra Sort\_Dim\_Vehicle cho Dim\_Vehicle



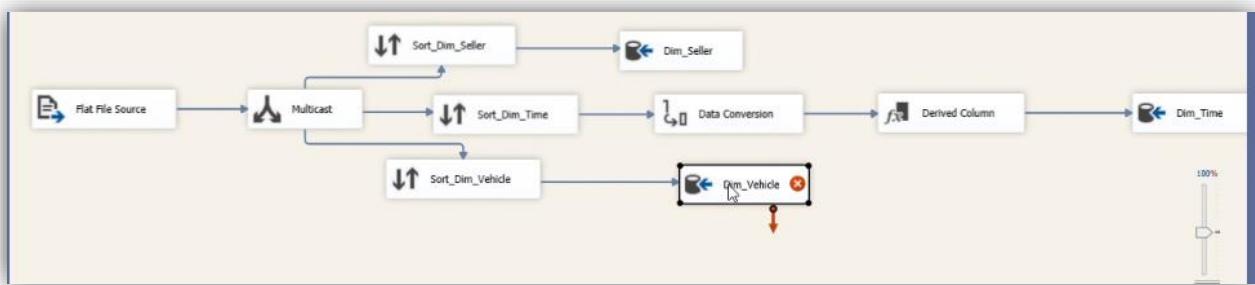
**Bước 2.** Click chuột phải vào Sort\_Dim\_Vehicle, chọn Edit: chọn các cột để đổ dữ liệu vào Sort\_Dim\_Vehicle



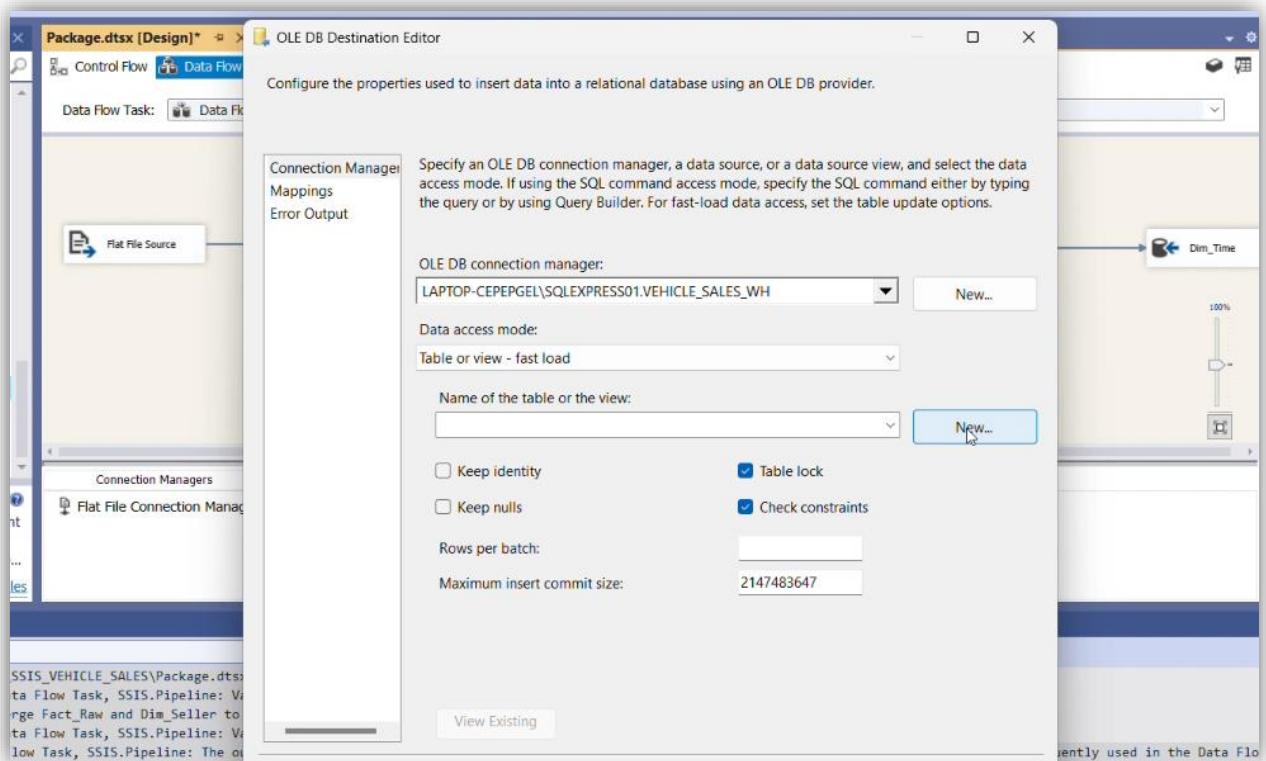
Click chọn Remove rows with duplicate sort values xóa đi các dòng dữ liệu trùng

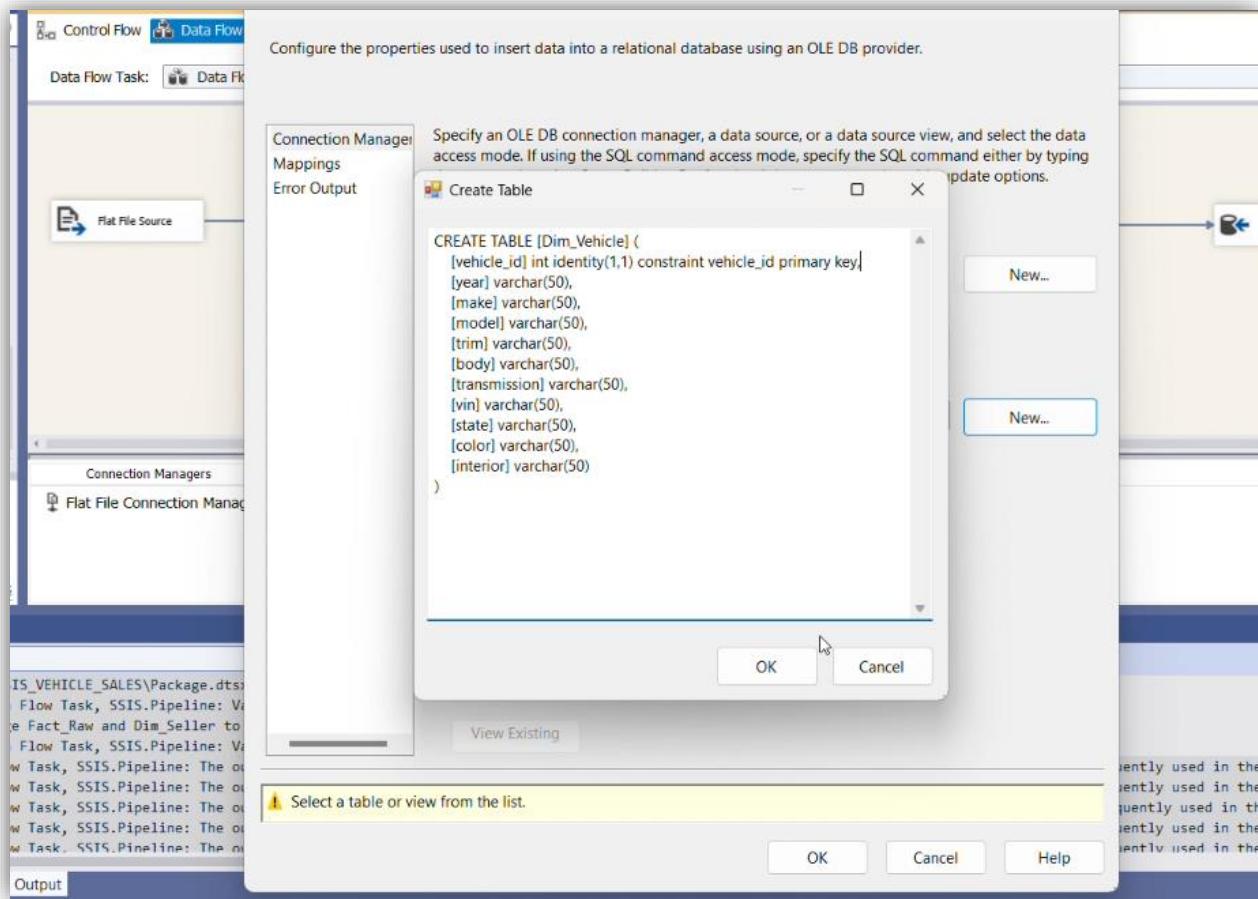
nhau và sau đó chọn OK.

**Bước 3.** Tạo mới một OLE DB Destination để đỗ dữ liệu gốc sau khi đã được xử lý vào trong bảng Dim\_Vehicle kho dữ liệu VEHICLE\_SALES\_WH



**Bước 4.** Chọn New... để tạo bảng Dim\_Vehicle



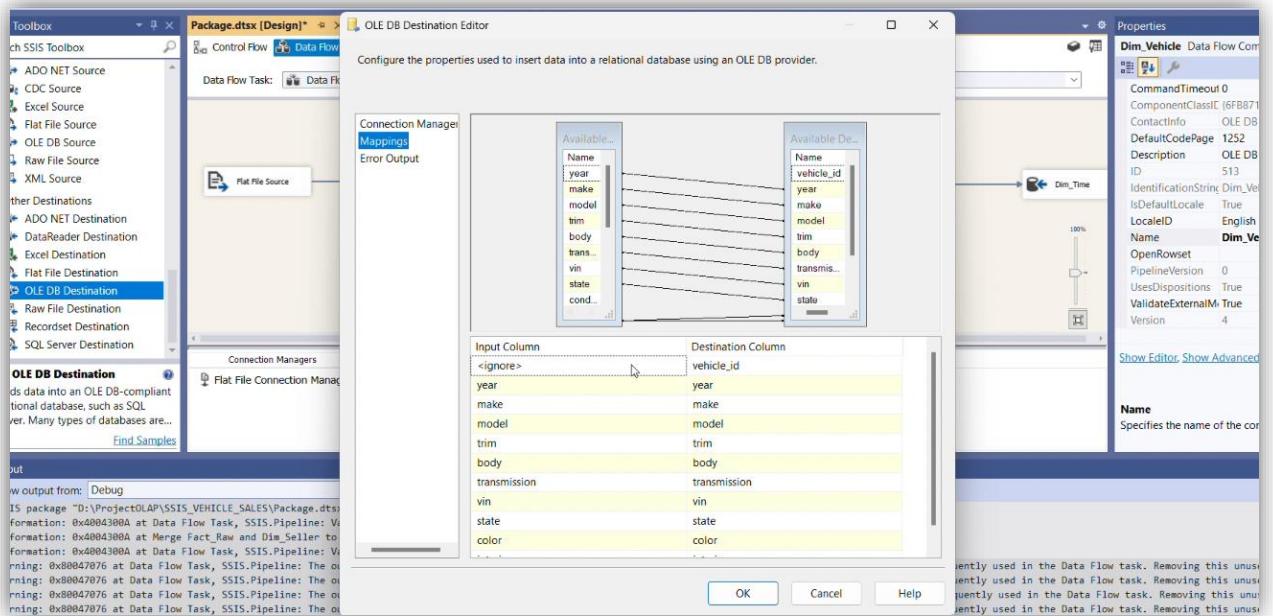


Nội dung câu lệnh SQL tạo bảng Dim\_Job\_Title như sau:

```
CREATE TABLE [Dim_Vehicle] (
    [vehicle_id] int identity(1,1) constraint vehicle_id primary key,
    [year] varchar(50),
    [make] varchar(50),
    [model] varchar(50),
    [trim] varchar(50),
    [body] varchar(50),
    [transmission] varchar(50),
    [vin] varchar(50),
    [state] varchar(50),
    [color] varchar(50),
    [interior] varchar(50)
)
```

```
[color] varchar(50),
[interior] varchar(50)
}
```

**Bước 5.** Tiếp đến ta cần chọn mục Mappings để xem xét việc ánh xạ các cột dữ liệu

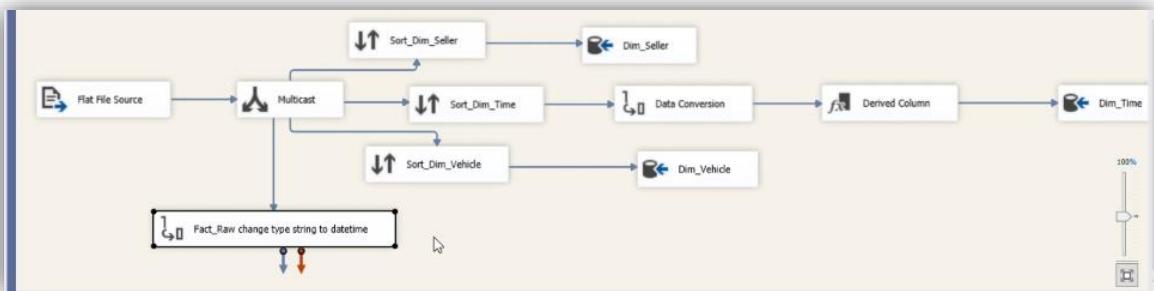


Chọn OK để hoàn tất thiết lập.

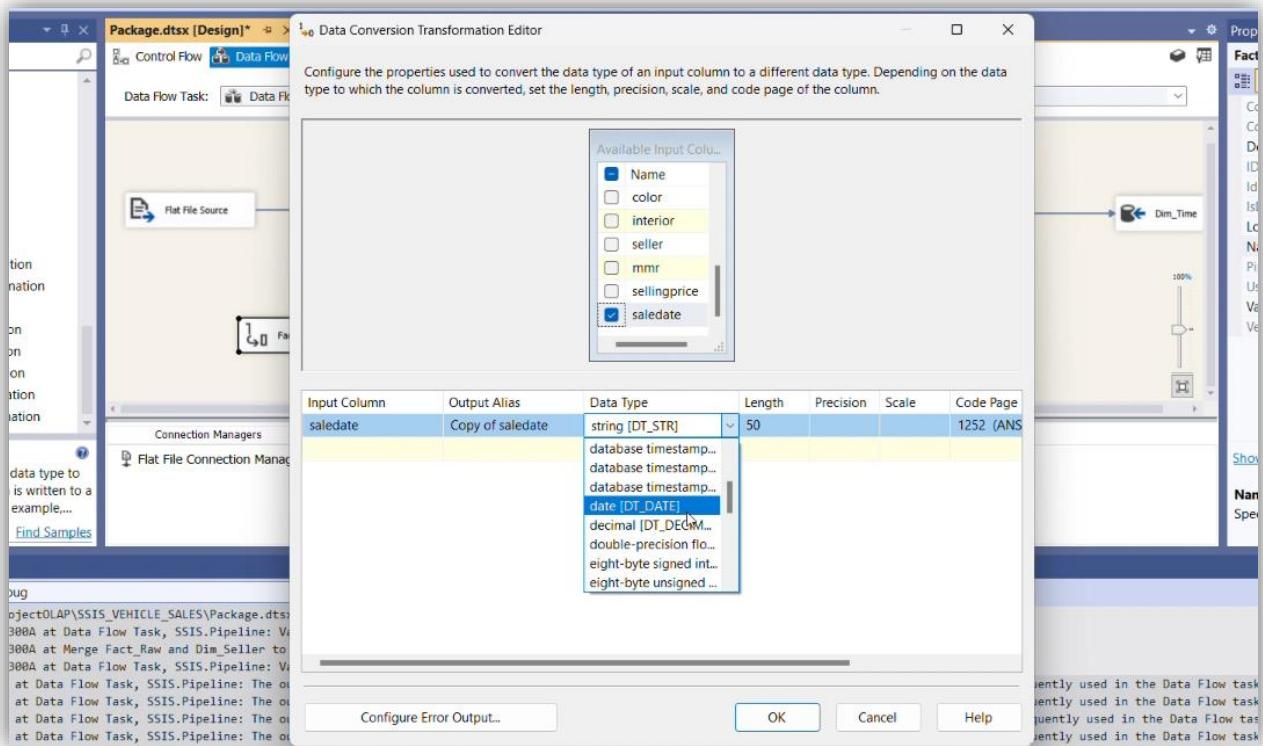
#### 2.4.4. Bảng Fact

Ta thấy bảng Fact có một cột saledate chứa kiểu dữ liệu DateTime, tuy nhiên kiểu dữ liệu DateTime khi lấy từ dữ liệu gốc (file .csv) sẽ được mặc định là kiểu string. Ta dùng Data Conversion để chuyển đổi kiểu string về dạng DateTime.

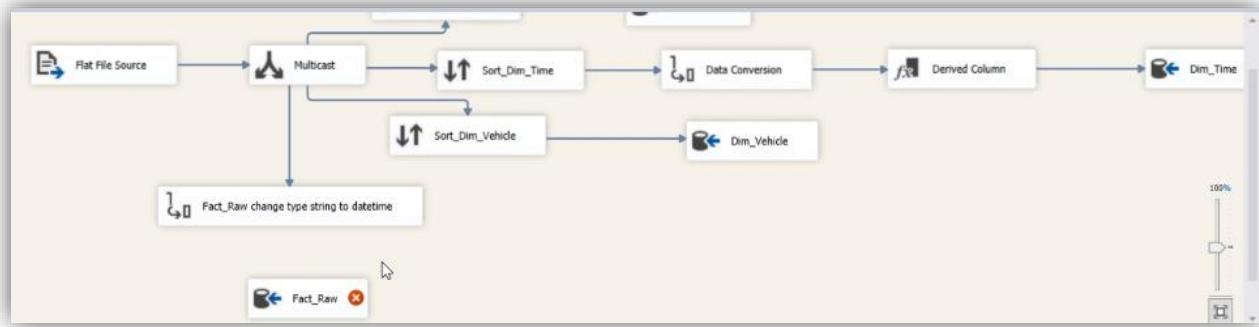
**Bước 1.** Tạo một Data Conversion và đặt tên là “Fact\_Raw change type string to datetime”



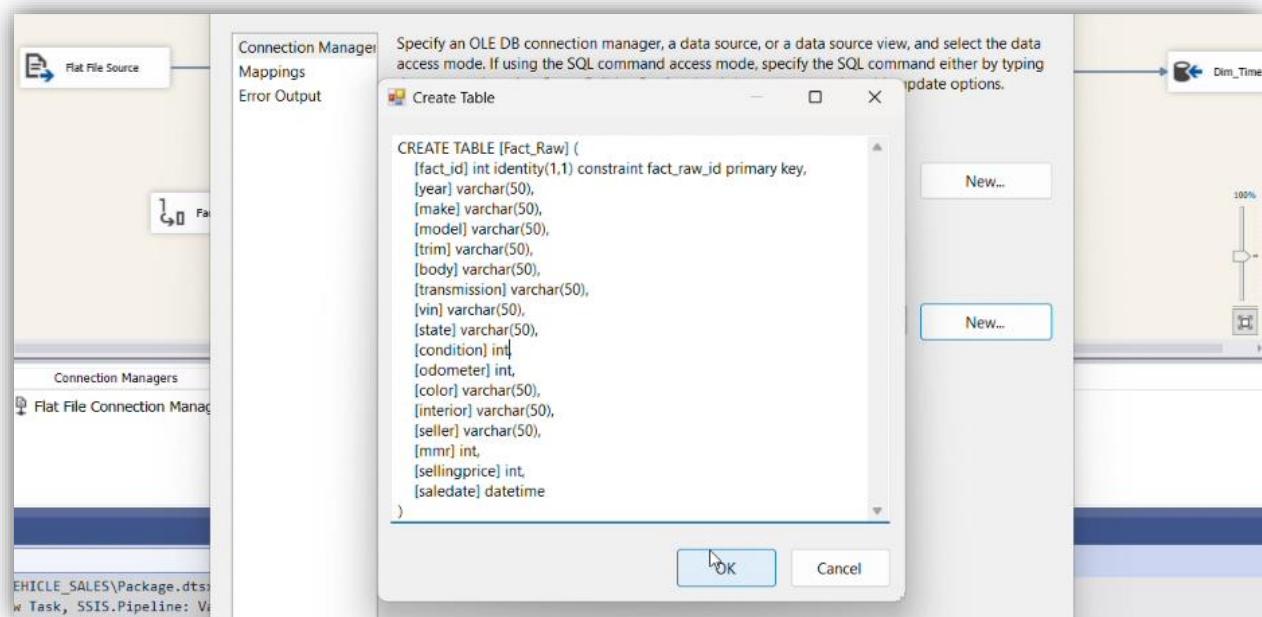
**Bước 2.** Click phải chuột vào Data Conversion trên và chọn Edit, tick chọn cột saledate, ở cột Data Type ta thấy mặc định kiểu dữ liệu là string, ta chọn lại kiểu dữ liệu cho cột này date[DT\_DATE] sau đó nhấn nút OK.



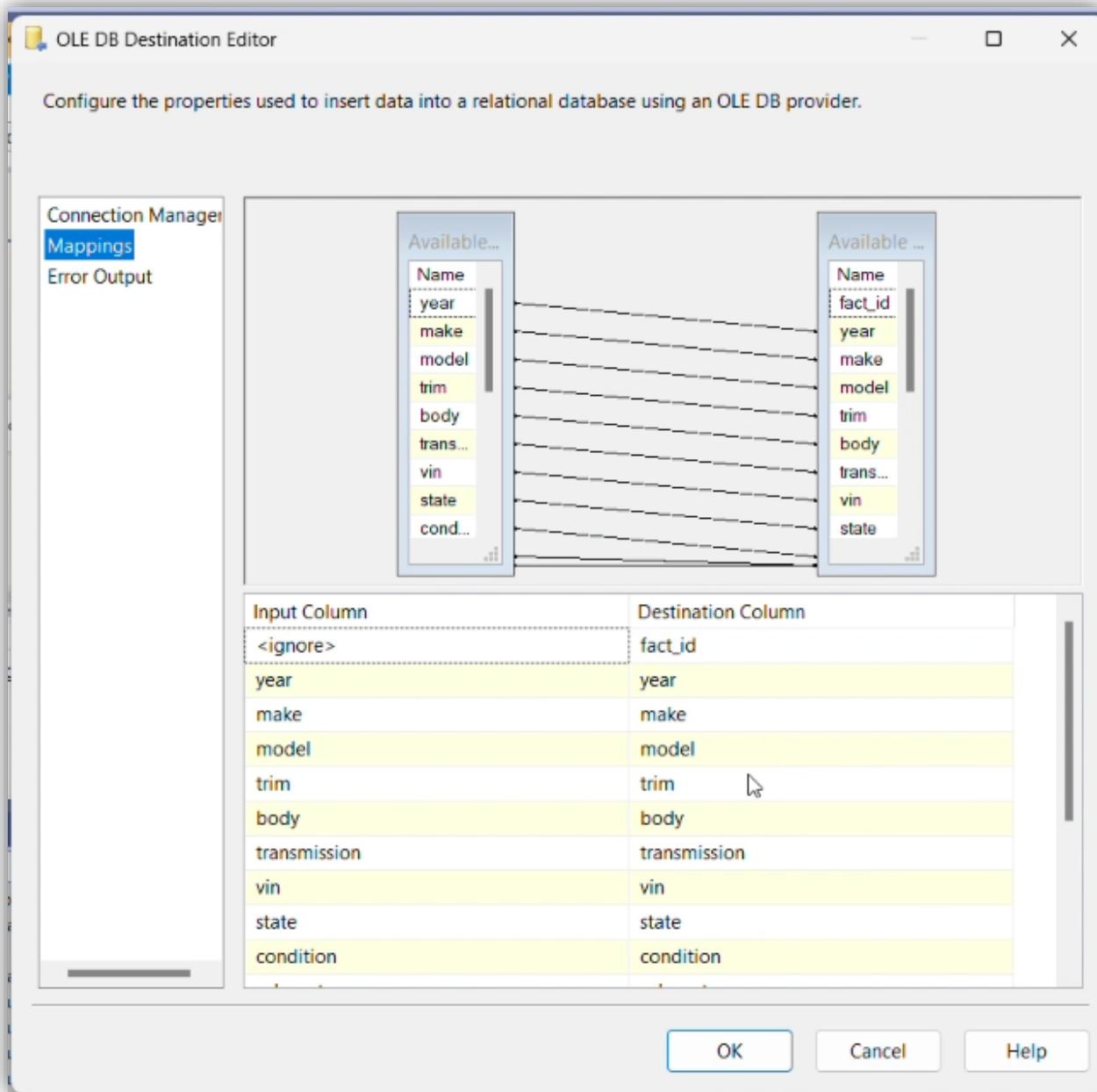
**Bước 3.** Tiến hành tạo bảng Fact, ta đặt tên là Fact\_Raw từ một OLE DB Destination



**Bước 4.** Click chuột phải và chọn Edit để tạo bảng Fact\_Raw có các cột là tất cả các cột từ dữ liệu gốc và chứa tất cả các dòng dữ liệu.



**Bước 3.** Tiếp đến ta cần chọn mục Mappings để xem xét việc ánh xạ các cột dữ liệu.

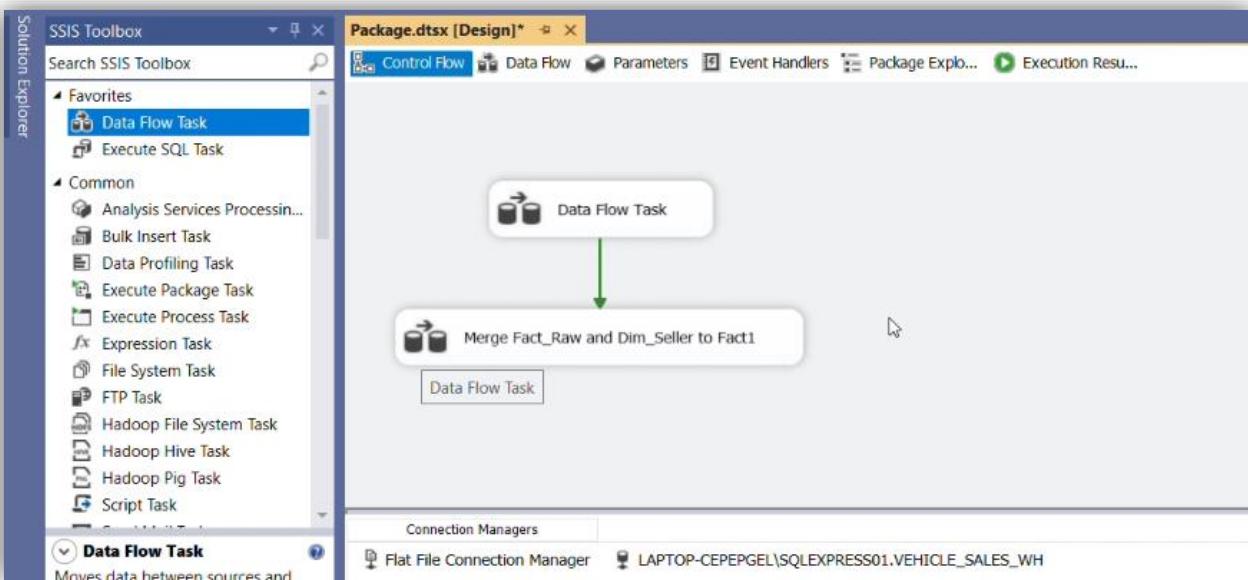


Cuối cùng nhấn nút OK để hoàn tất quá trình tạo bảng.

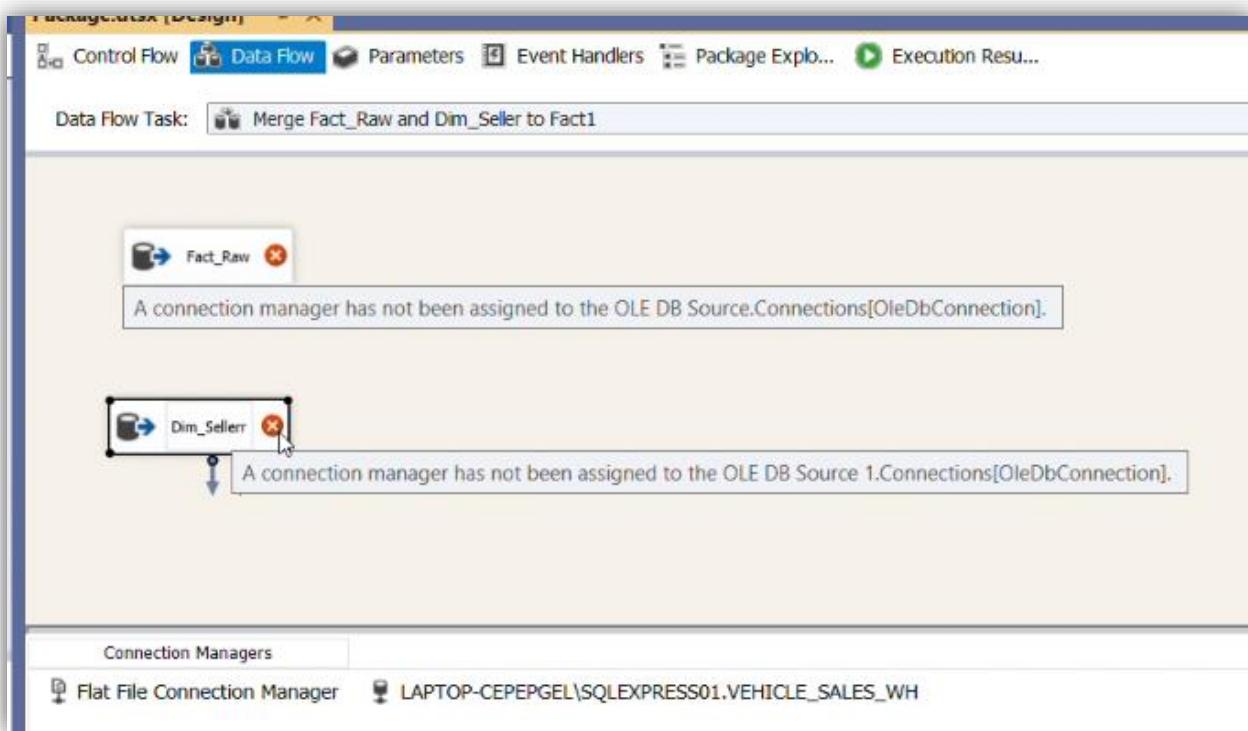
Tiếp theo đây ta sẽ thực hiện quá trình lần lượt loại bỏ các cột dữ liệu trùng của bảng Fact với các Dimension, thực hiện thêm khóa ngoại vào bảng Fact nhằm thu gọn bảng Fact, tối ưu hóa quá trình phân tích dữ liệu.

#### 2.4.4.1. Merge Fact\_Raw và Dim\_Seller vào Fact1

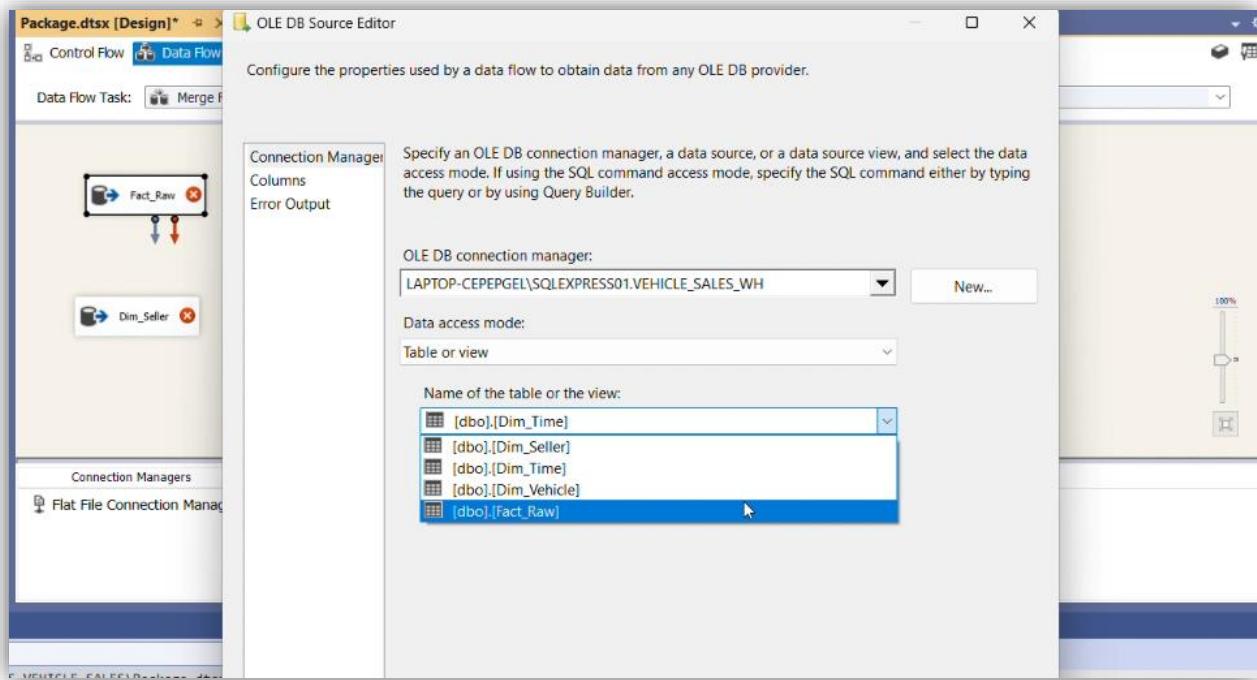
**Bước 1.** Ở tab Control Flow, tạo mới 1 Data Flow Task và đổi tên Data Flow Task thứ 2 là “Merge Fact\_Raw and Dim\_Seller to Fact1”



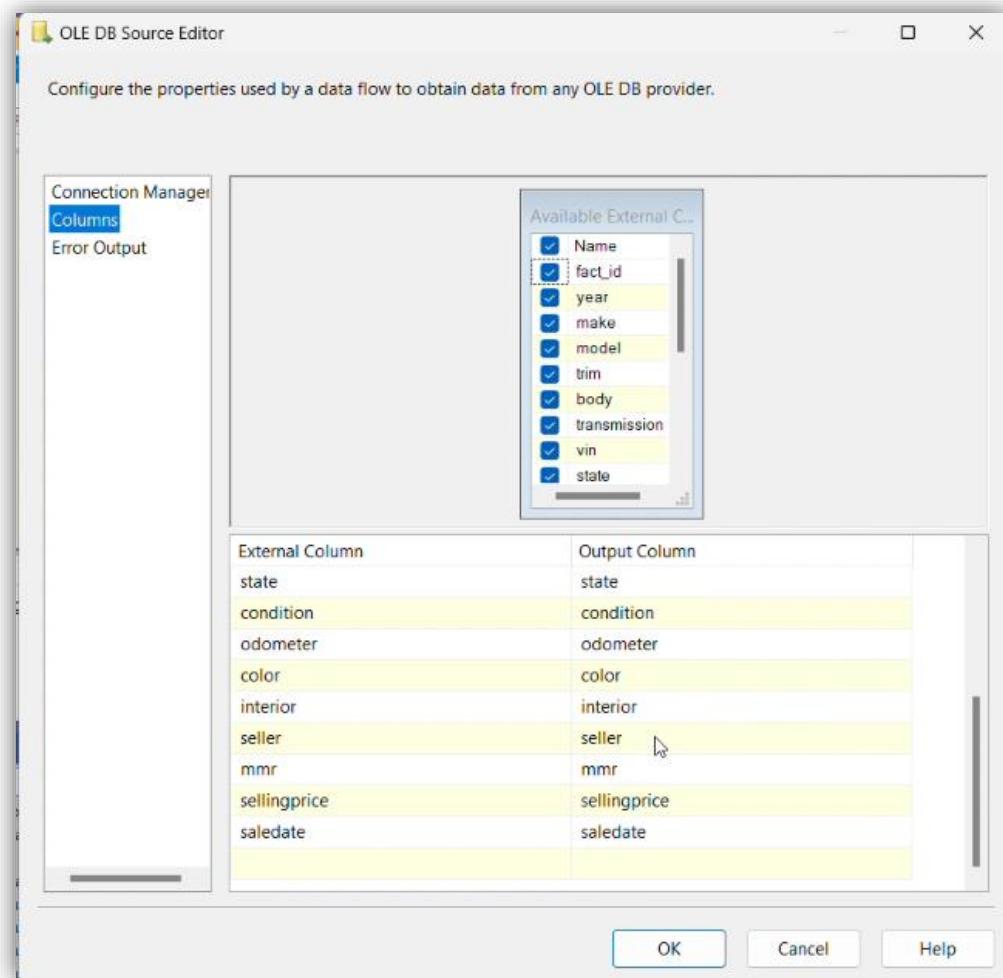
**Bước 2.** Click chuột phải vào Data Flow Task nói trên và chọn Edit, trong tab Data Flow ta tạo 2 OLE DB Source và đổi tên thành Fact\_Raw và Dim\_Seller



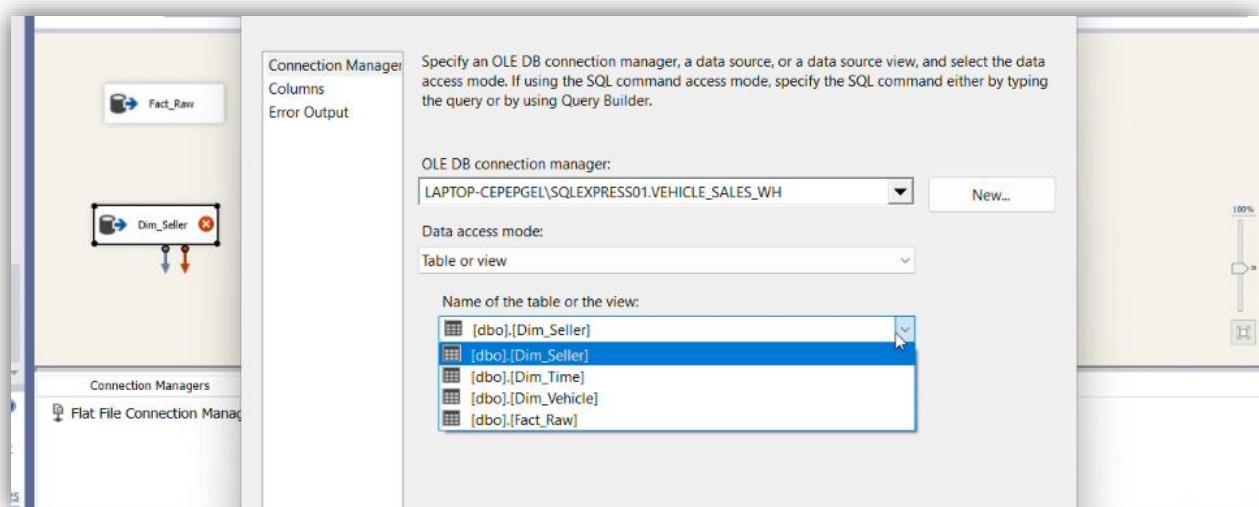
**Bước 3.** Click chuột phải chọn Edit, sau đó chọn bảng Fact\_Raw đã tạo trước đó làm data source cho bảng Fact\_Raw mới này.

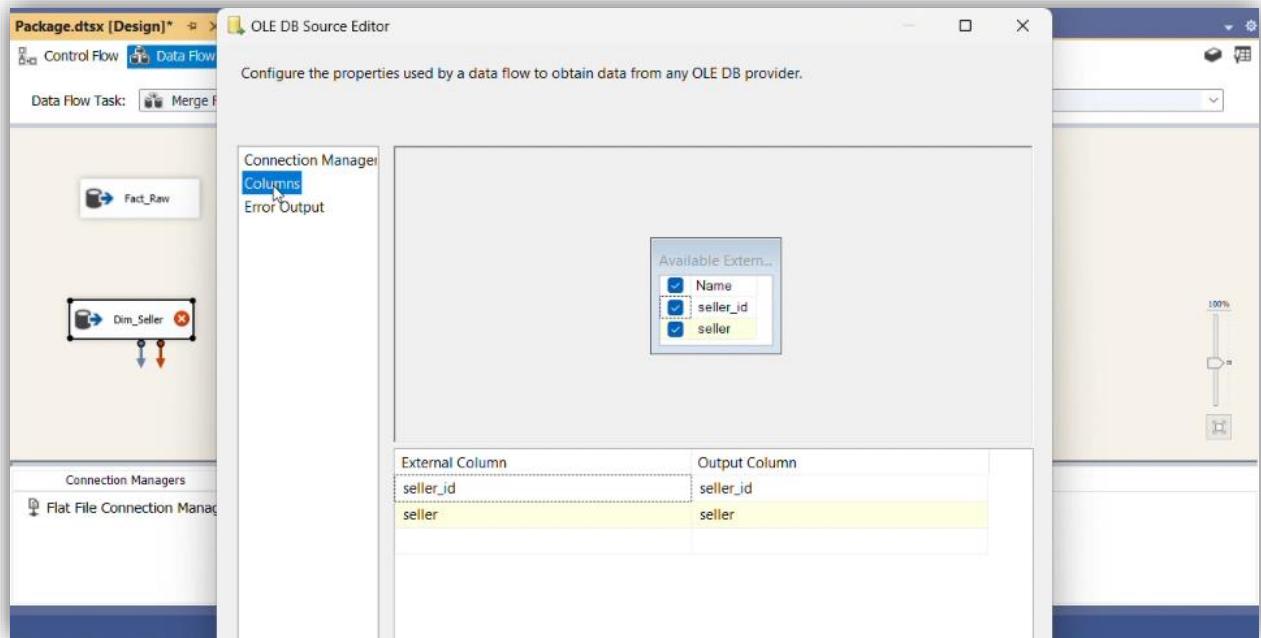


**Bước 4.** Chọn mục Columns để xem xét các cột được ánh xạ. Nhấn OK.

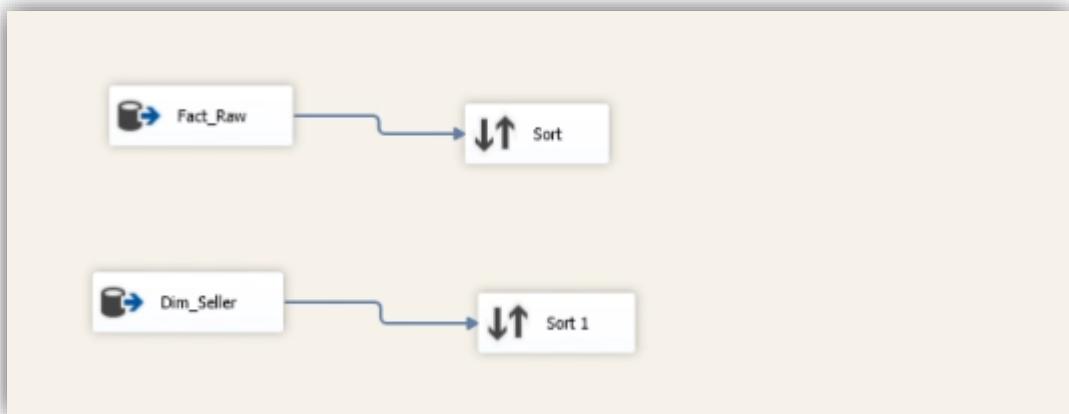


### Bước 5. Tương tự thực hiện chọn ánh xạ cột cho Dim\_Seller

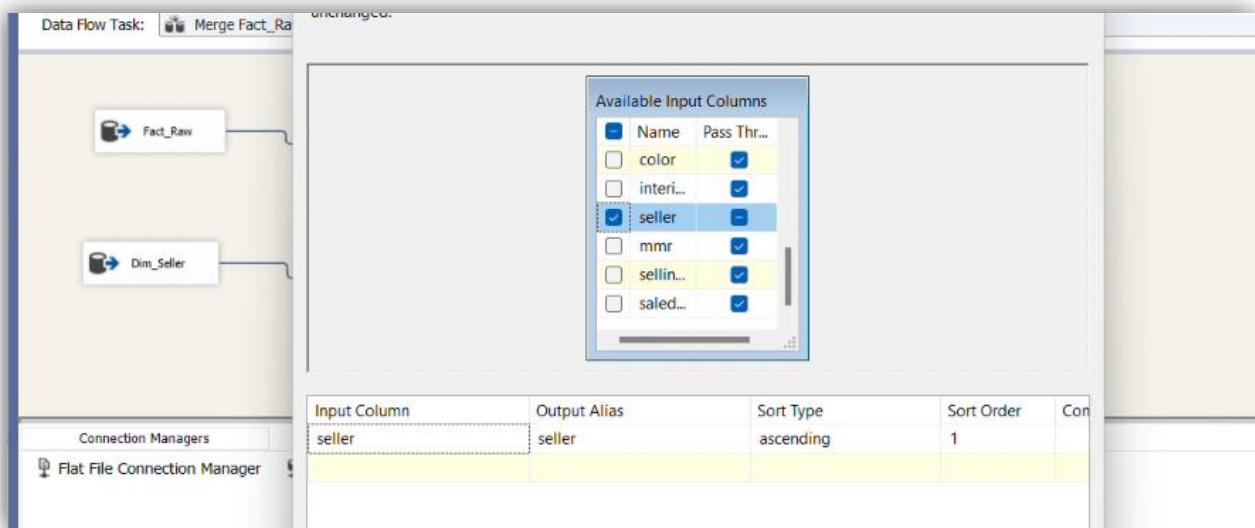




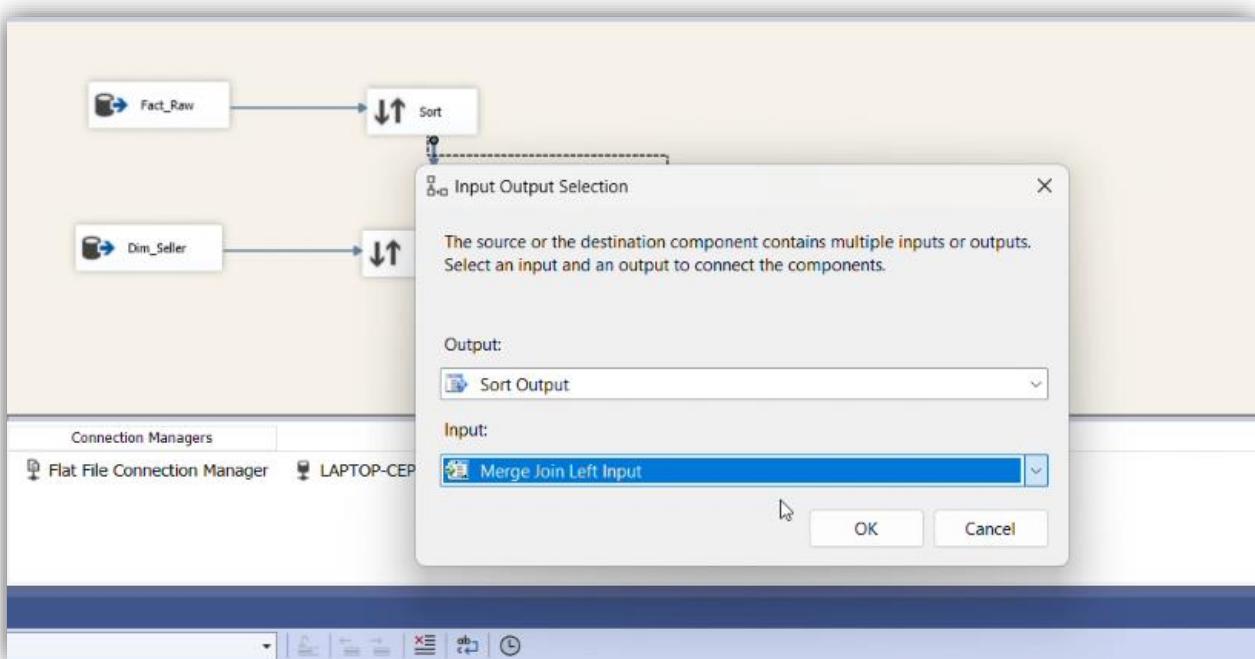
**Bước 6.** Tạo 2 Sort là Sort và Sort1 tương ứng với mỗi Source.



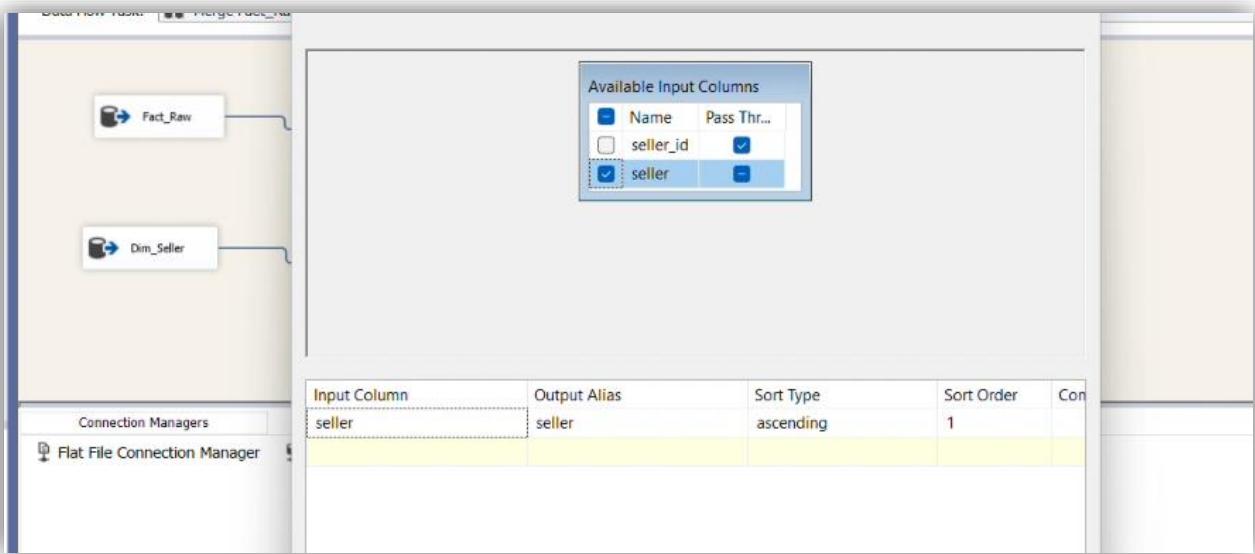
**Bước 7.** Ở Sort, click chuột phải chọn Edit và chọn cột với bảng Dim\_Seller để chuẩn bị cho quá trình merge.



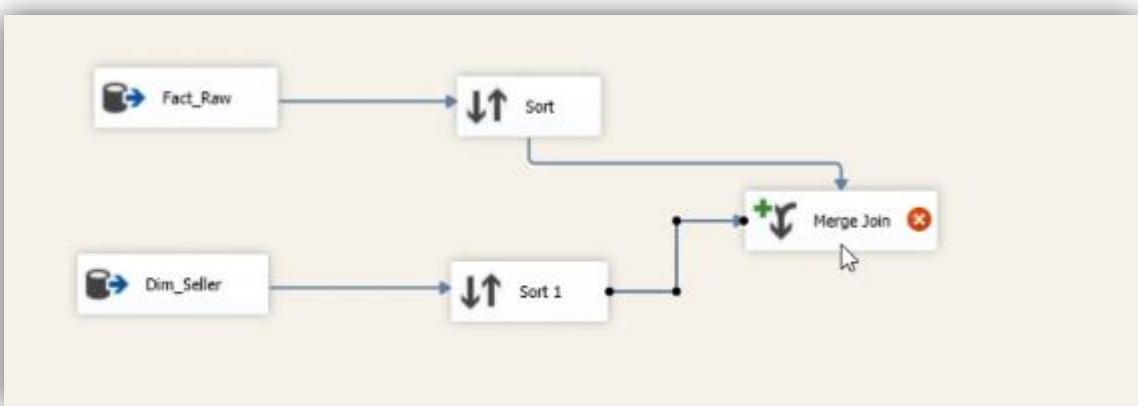
**Bước 8.** Tạo một Merge Join và nối với Sort, tiếp theo ta chọn Merge Join Left Input để giữ lại toàn bộ các dòng trong bảng Fact\_Raw bất kể có kết quả khi thực hiện phép kết trái với cột ID của bảng Dim\_Location hay không.



**Bước 9.** Tương tự ta chọn cột cho Sort1

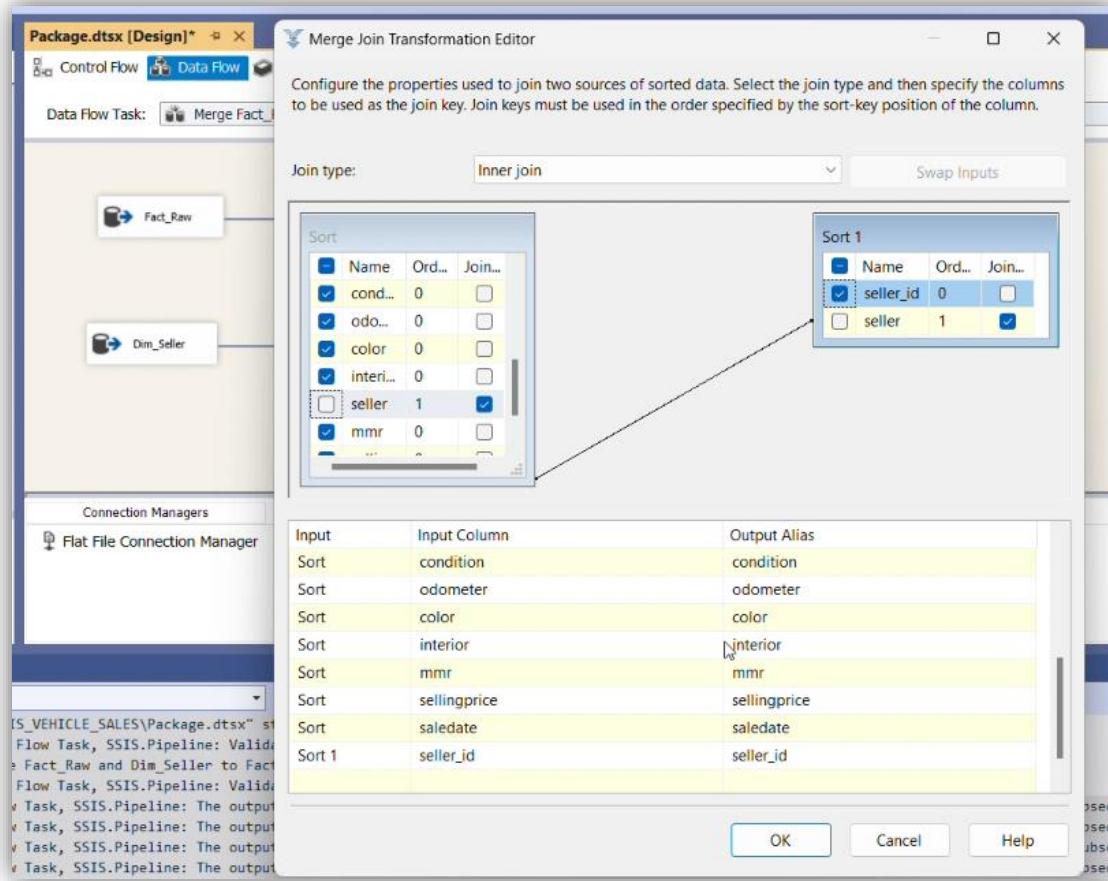


Nối Sort1 với Merge Join

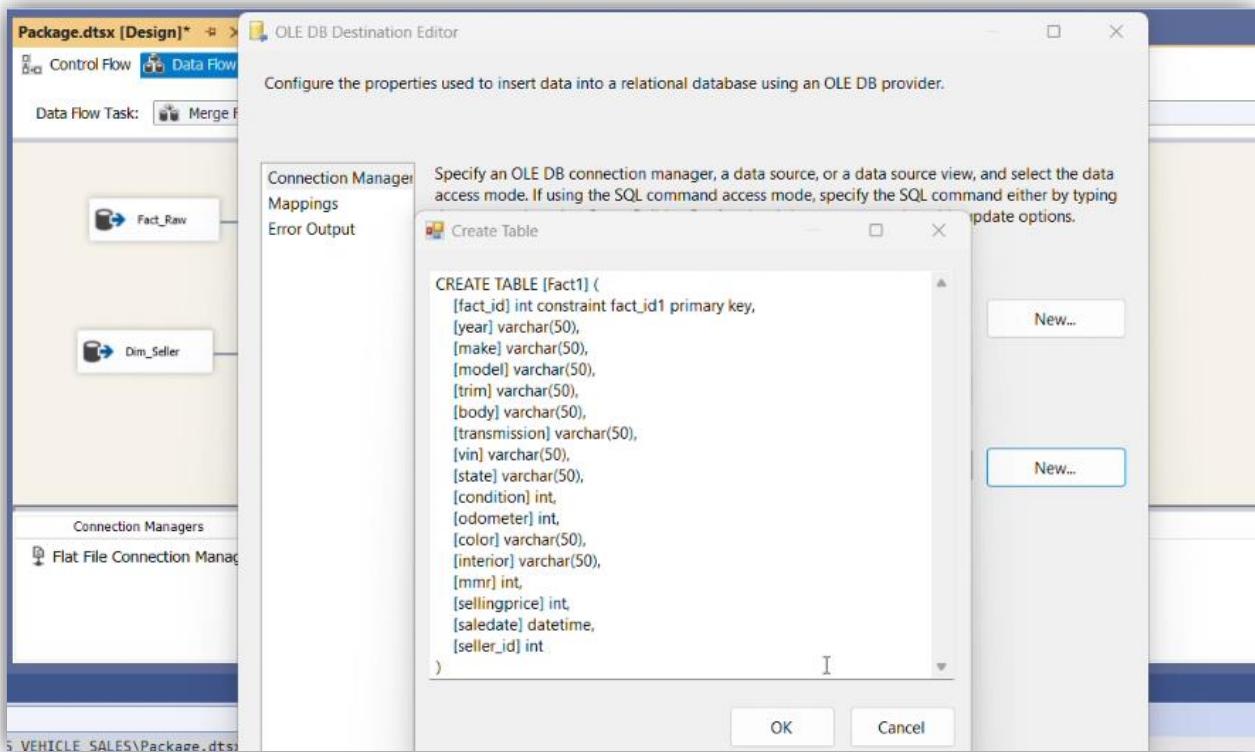


#### Bước 10.

- Chuột phải vào Merge Join và nhấn Edit, một hộp thoại merge editor xuất hiện: ở đây ta tick chọn tất cả các cột của Sort nhưng không lấy thuộc tính là seller.
- Tiếp theo ta chọn seller\_id ở Sort1 để merge vào Fact\_Raw
- Kết quả sau khi merge là bảng Fact\_Raw không còn thuộc tính seller và có thêm 1 thuộc tính mới là seller\_id



**Bước 11.** Tạo bảng Fact1 từ một OLE DB Destination để chứa tất cả những gì đã merge



Nội dung câu lệnh SQL tạo bảng Fact1 như sau:

```

CREATE TABLE [Fact1] (
    [fact_id] int constraint fact_id1 primary key,
    [year] varchar(50),
    [make] varchar(50),
    [model] varchar(50),
    [trim] varchar(50),
    [body] varchar(50),
    [transmission] varchar(50),
    [vin] varchar(50),
    [state] varchar(50),
    [condition] int,
    [odometer] int,
    [color] varchar(50),
    [interior] varchar(50),
    [mmr] int,
    [sellingprice] int,
    [saledate] datetime,
    [seller_id] int
)

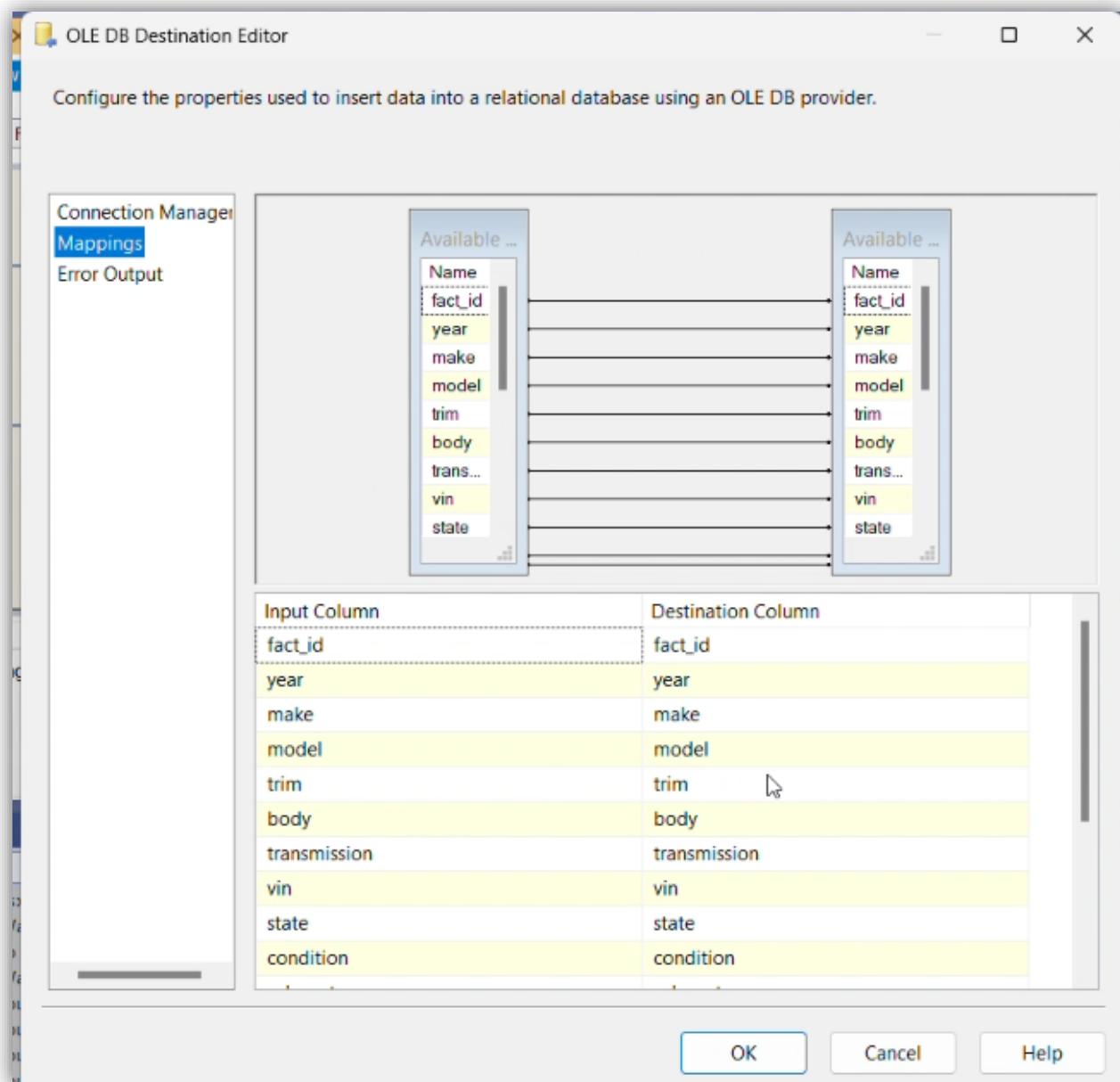
```

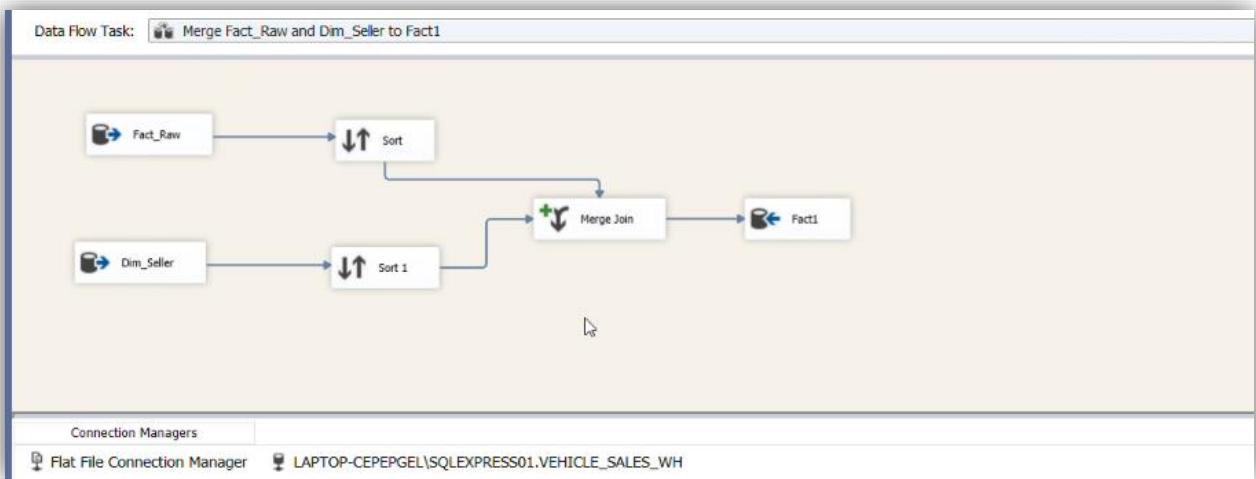
```

[sellingprice] int,
[saledate] datetime,
[seller_id] int
)

```

**Bước 12.** Chọn mục Mappings để xem xét việc ánh xạ các cột dữ liệu và nhấn OK

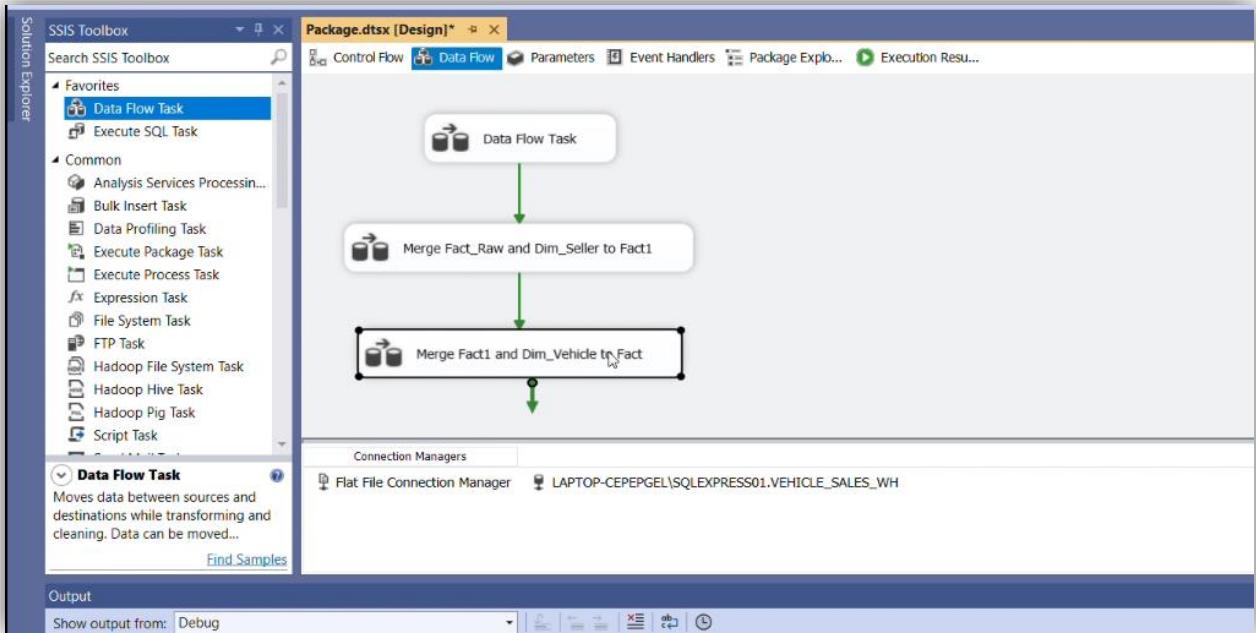




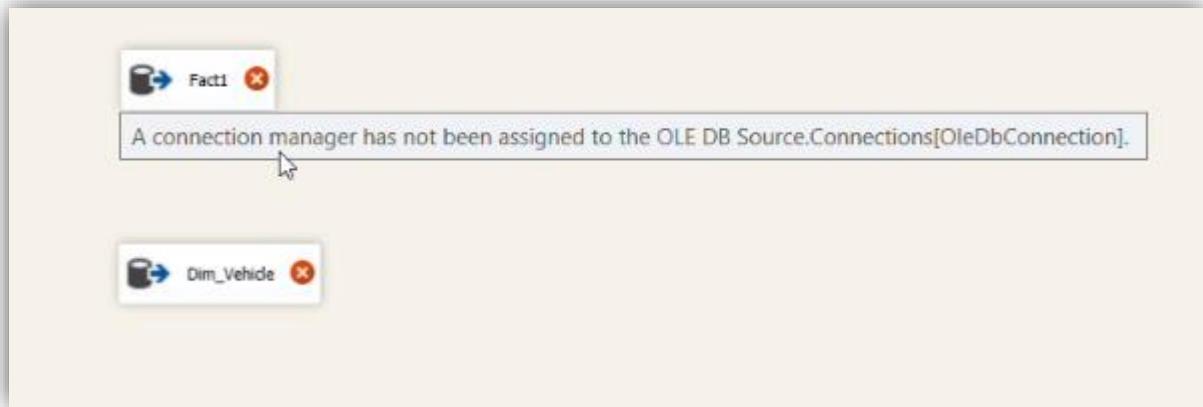
Tiếp tục quá trình merge bảng Fact1 với Dimension còn lại để có 1 bảng Fact hoàn chỉnh.

#### 2.4.4.2. Merge Fact1 và Dim\_Vehicle vào Fact

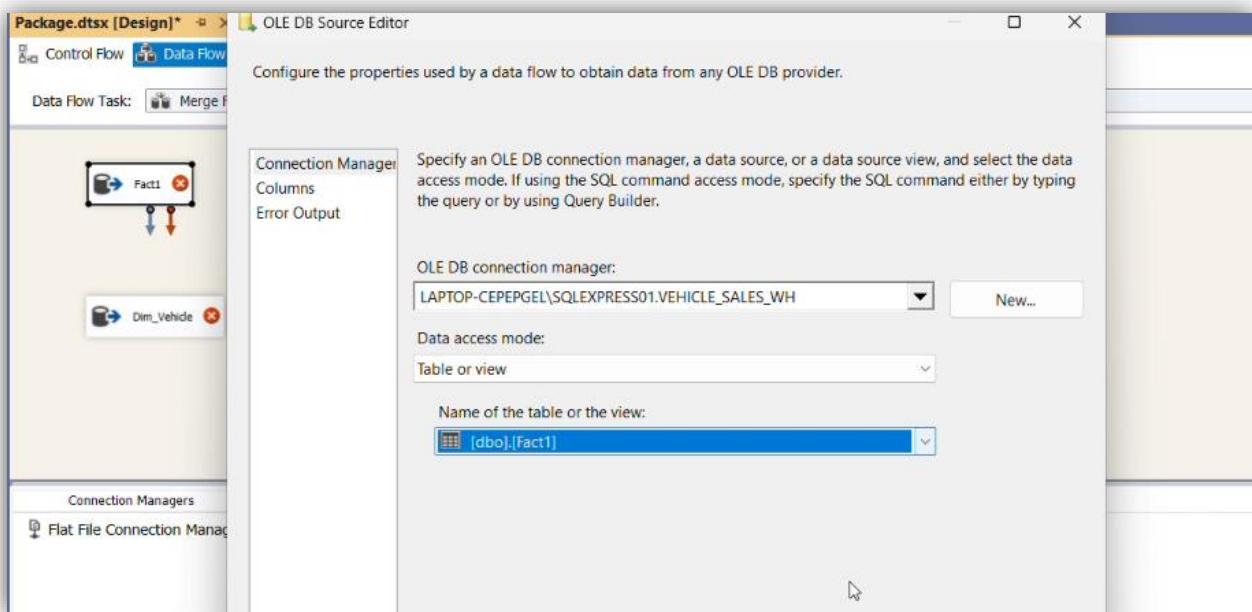
**Bước 1.** Ở tab Control Flow, tạo thêm một Data Flow Task và đổi tên Data Flow Task này là “Merge Fact1 and Dim\_Vehicle to Fact”



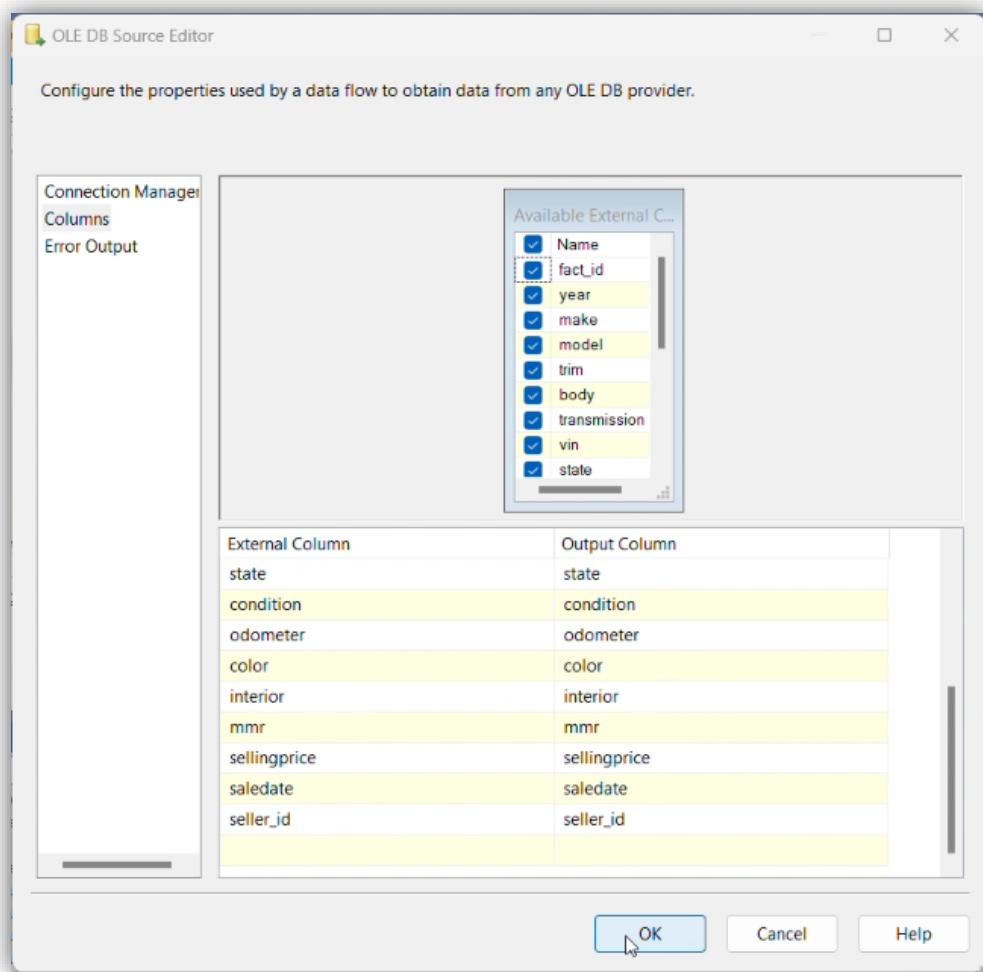
**Bước 2.** Click chuột phải vào Data Flow Task nói trên và chọn Edit, trong tab Data Flow ta tạo 2 OLE DB Source và đổi tên thành Fact1 và Dim\_Vehicle



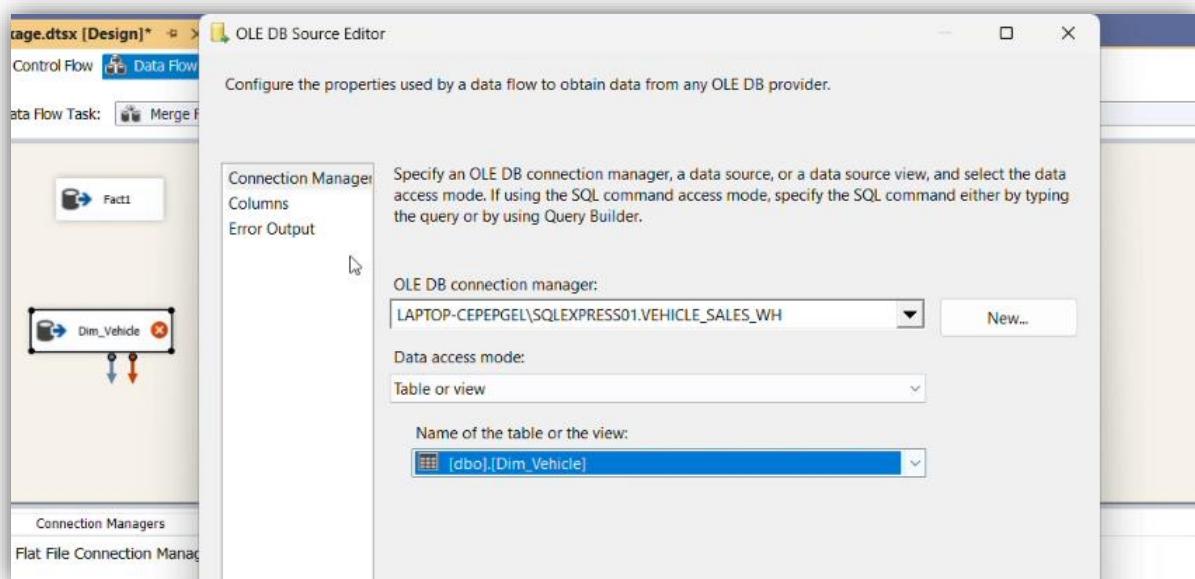
**Bước 3.** Click chuột phải vào Fact1 chọn Edit, sau đó chọn bảng Fact1 đã được tạo khi merge Fact\_Raw và Dim\_Seller làm data source.



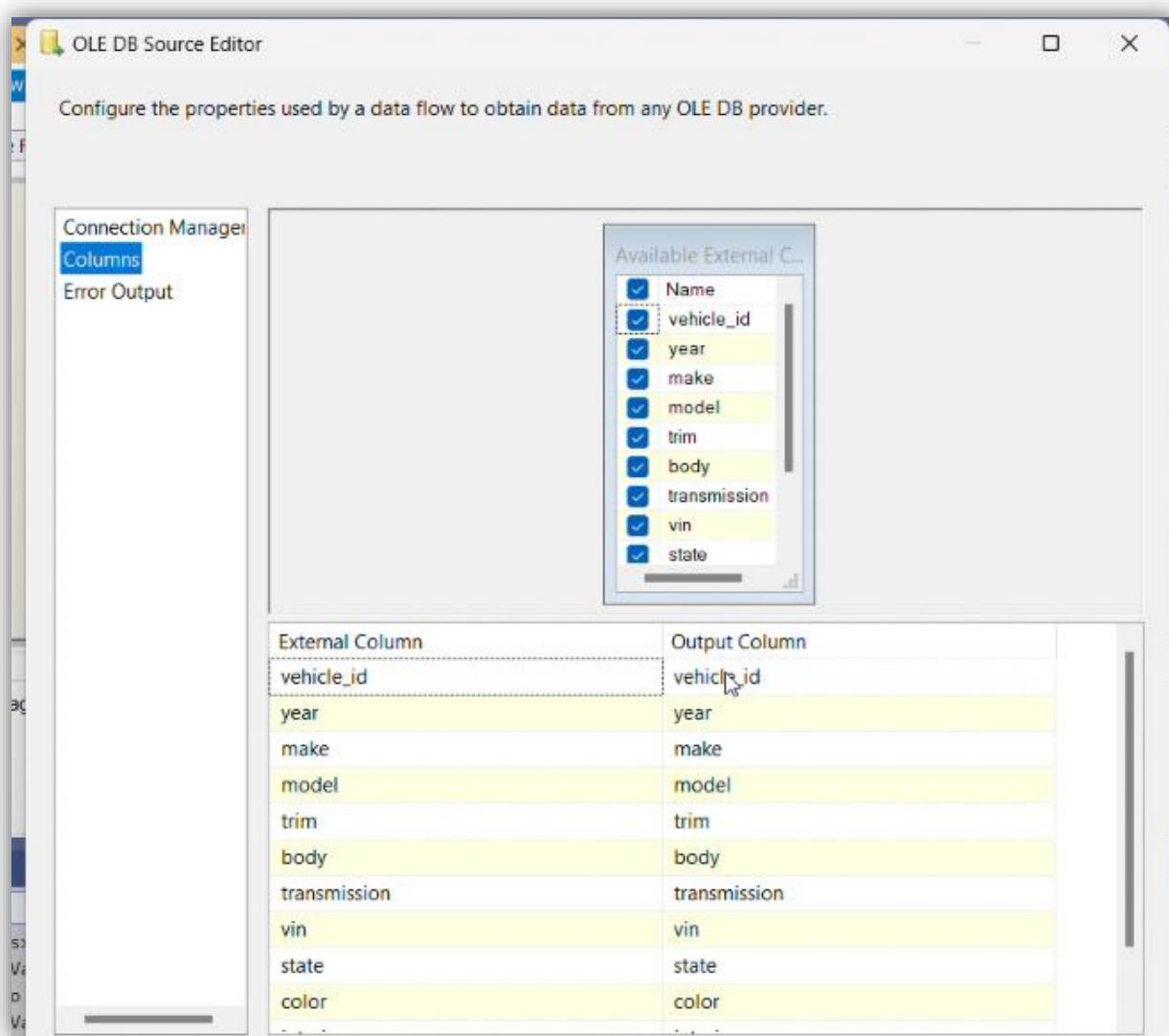
**Bước 4.** Chọn mục Columns để xem xét các cột được ánh xạ. Nhấn OK.



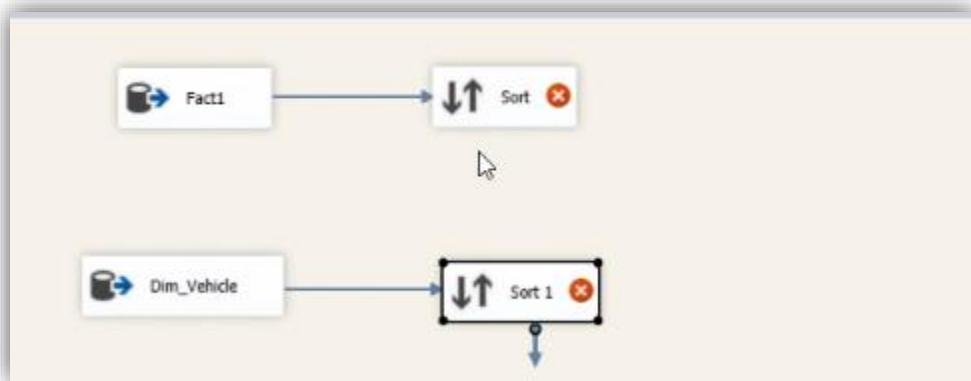
### Bước 5. Thực hiện chọn ánh xạ các cột cho Dim\_Vehicle



Chọn mục Columns để xem xét các cột được ánh xạ. Nhấn OK.

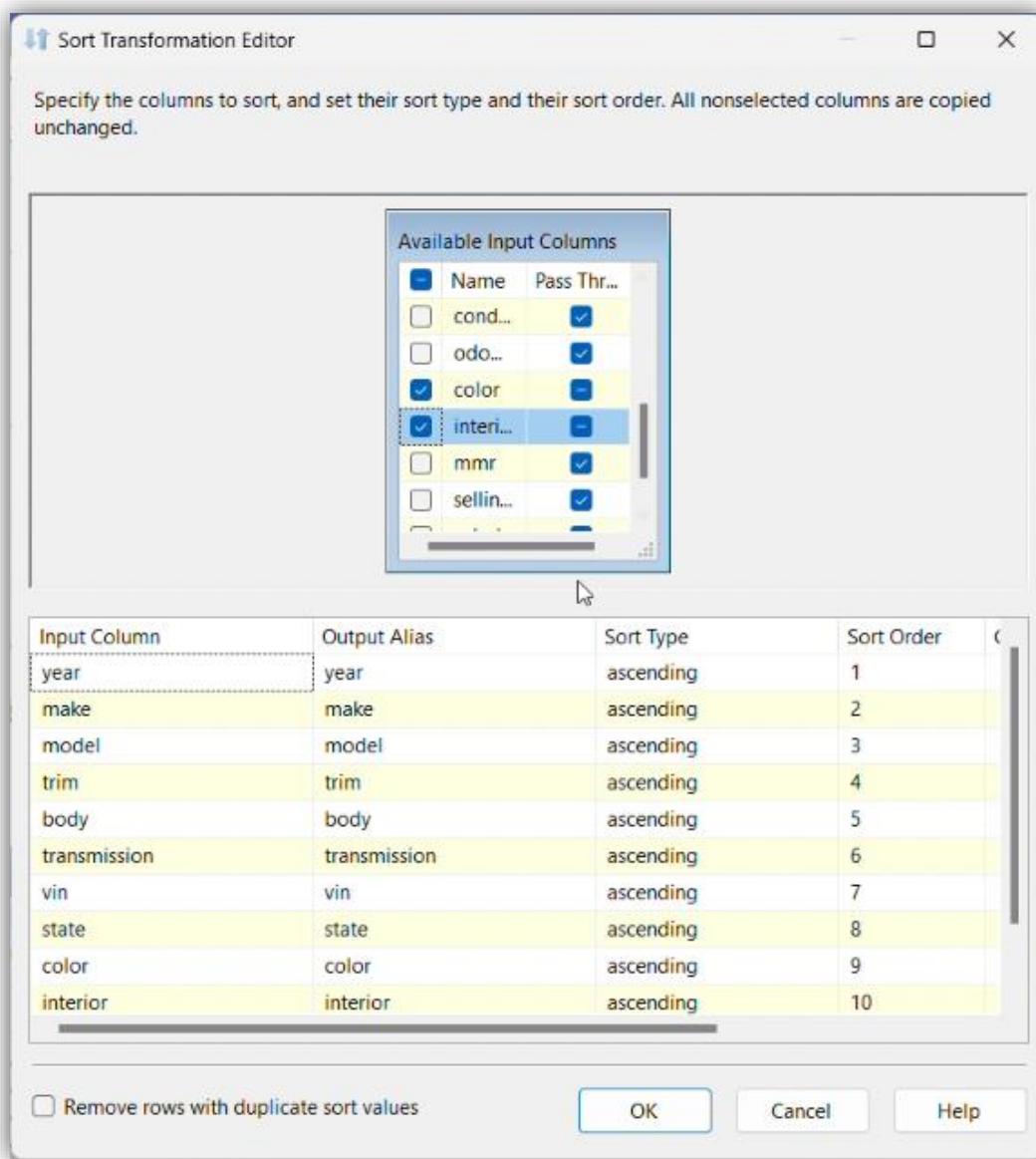


**Bước 6.** Tạo 2 Sort tương ứng với mỗi Source

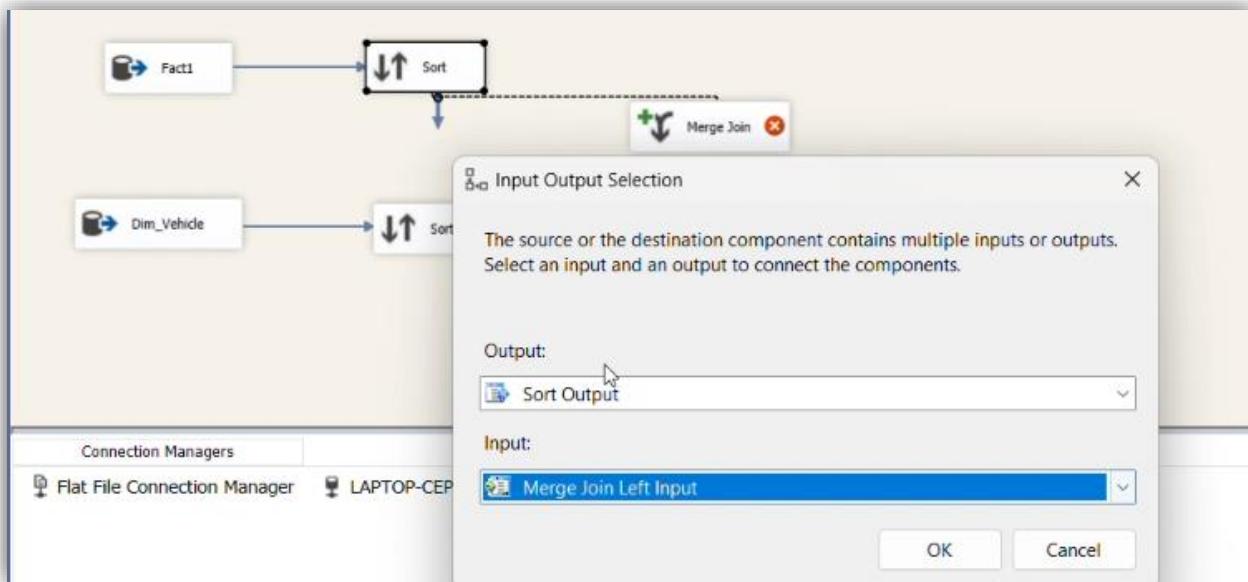


**Bước 7.** Tại Sort, click chuột phải chọn Edit và chọn các cột theo thứ tự giống với

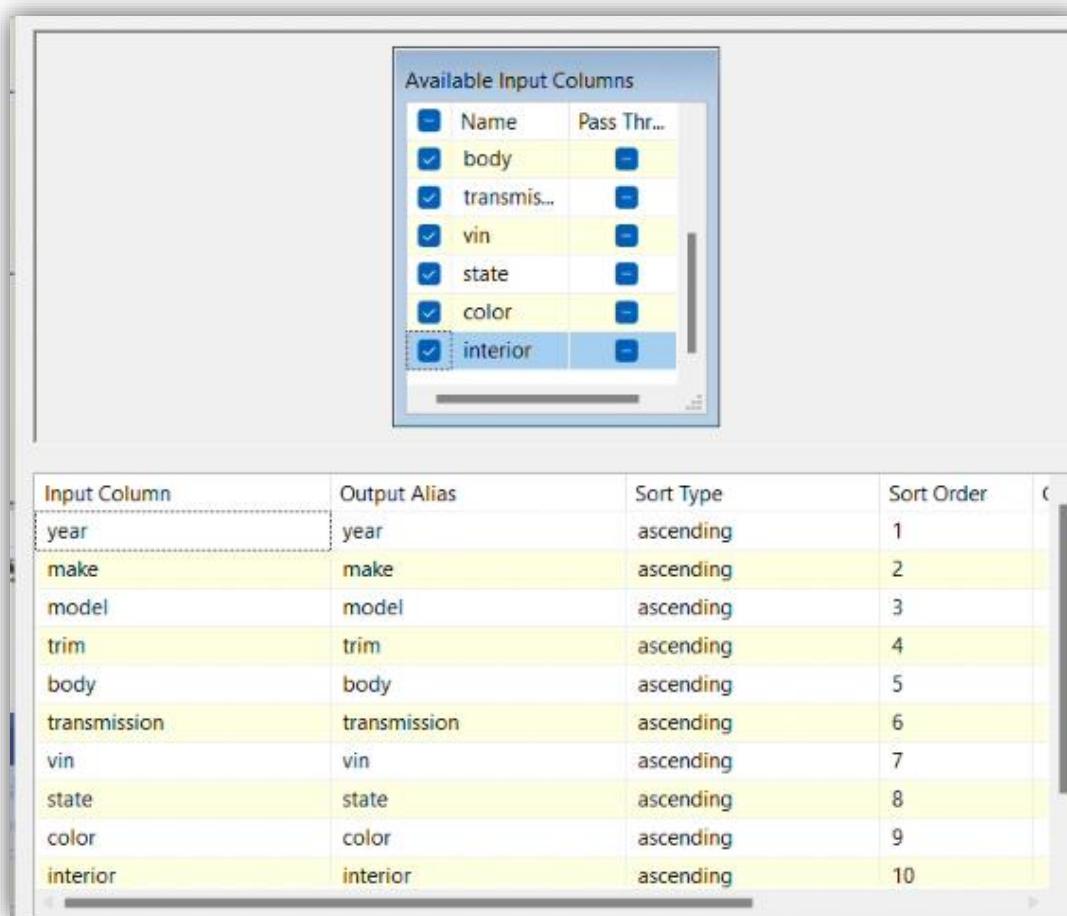
bảng Dim\_Vehicle để chuẩn bị cho quá trình merge.



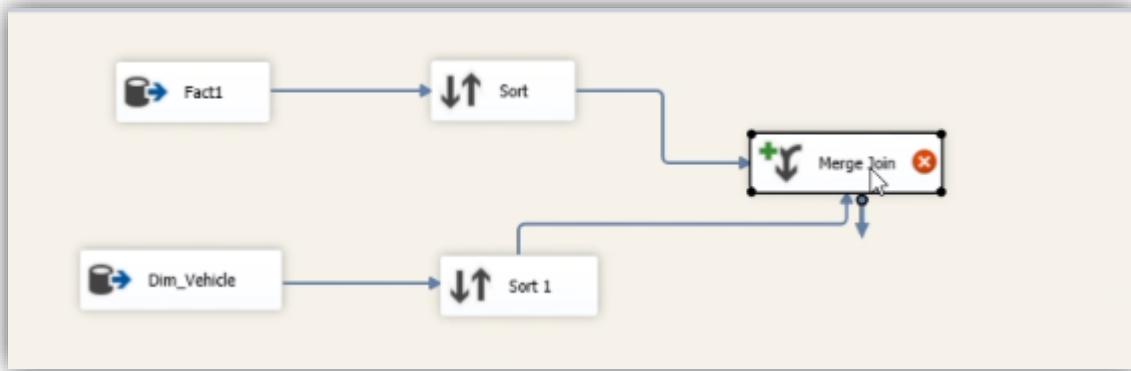
**Bước 8.** Tạo một Merge Join và nối với Sort, tiếp theo chọn Merge Join Left Input để giữ lại toàn bộ các dòng trong bảng Fact1 bất kể có kết quả khi thực hiện phép kết trái với cột ID của bảng Dim\_Vehicle hay không.



**Bước 9.** Tương tự ta chọn các cột (trừ vehicle\_id) cho Sort1

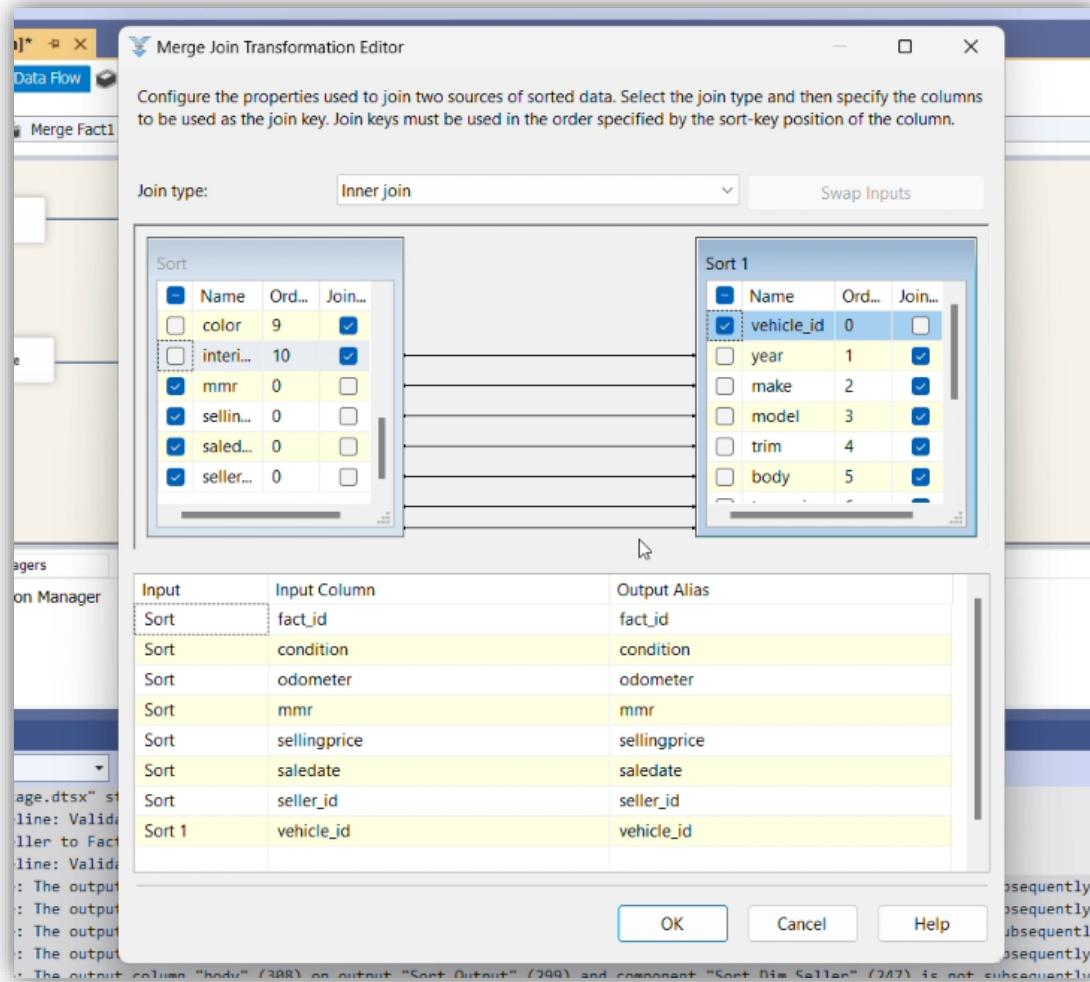


Nối Sort1 với Merge Join

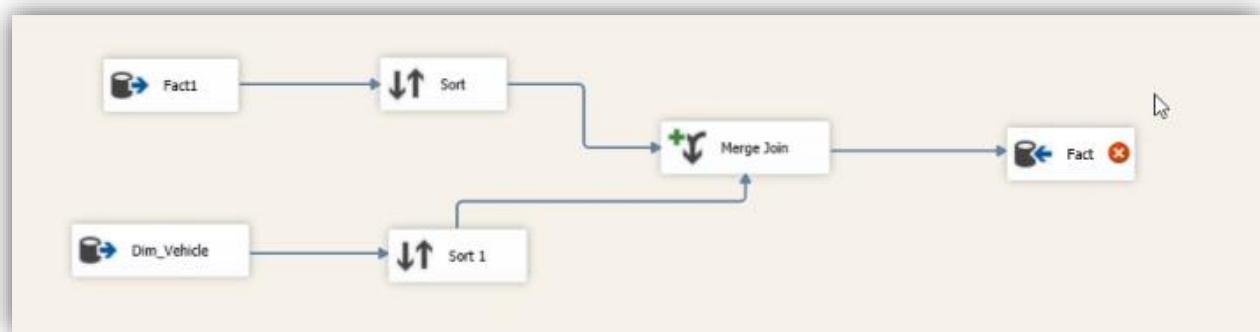


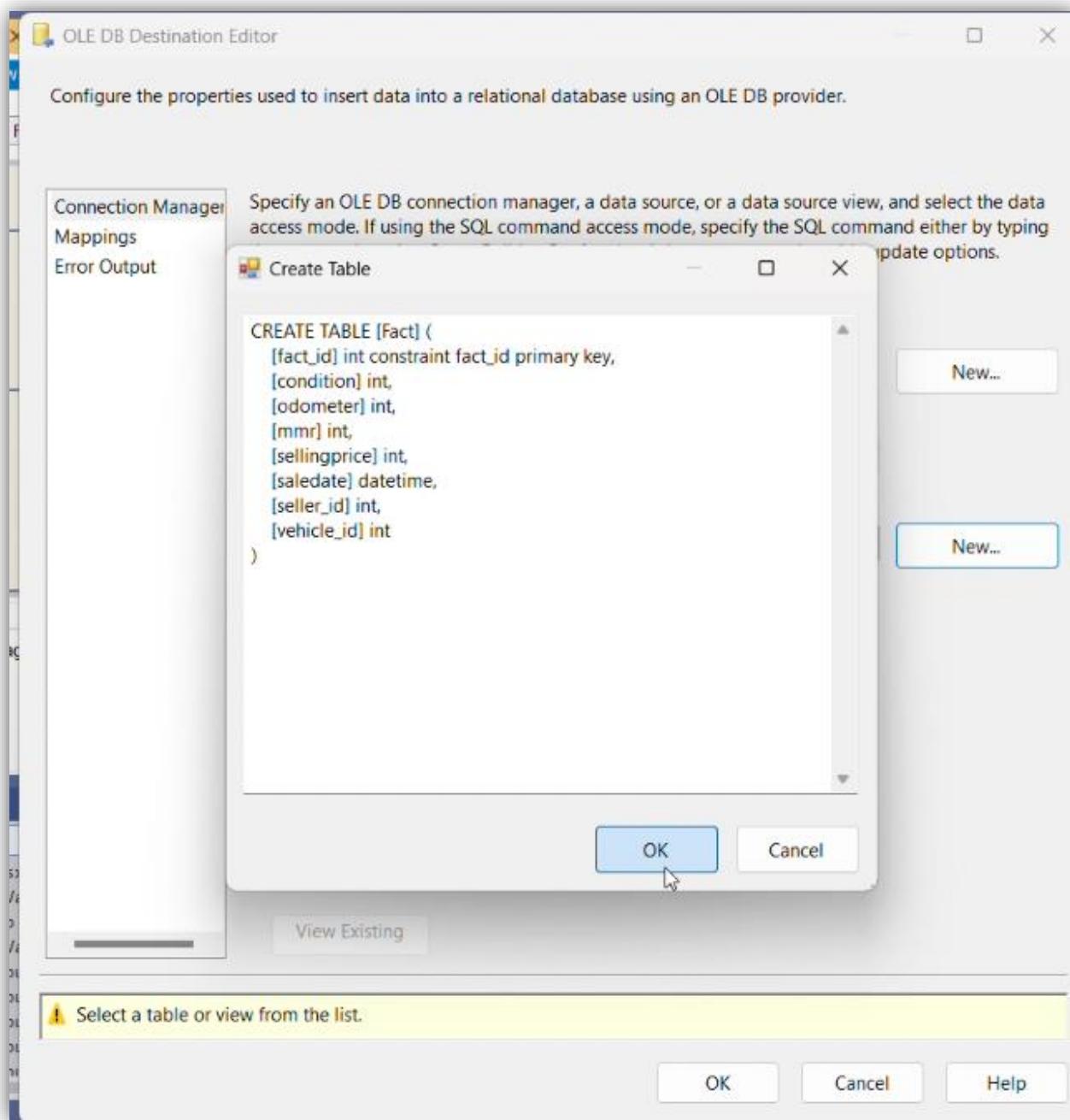
### Bước 10.

- Chuột phải vào Merge Join và nhấn Edit, một hộp thoại merge editor xuất hiện: ở đây ta tick chọn tất cả các cột của Sort nhưng không lấy 10 thuộc tính của bảng Dim\_Vehicle
- Tiếp theo ta chọn vehicle\_id ở Sort1 để merge vào Fact1
- Kết quả sau khi merge là bảng Fact1 không còn 10 thuộc tính của bảng Dim\_Vehicle và có thêm 1 thuộc tính mới là vehicle\_id



**Bước 11.** Tạo bảng Fact từ một OLE DB Destination để chứa tất cả những gì đã merge





Nội dung câu lệnh SQL tạo bảng Fact2 như sau:

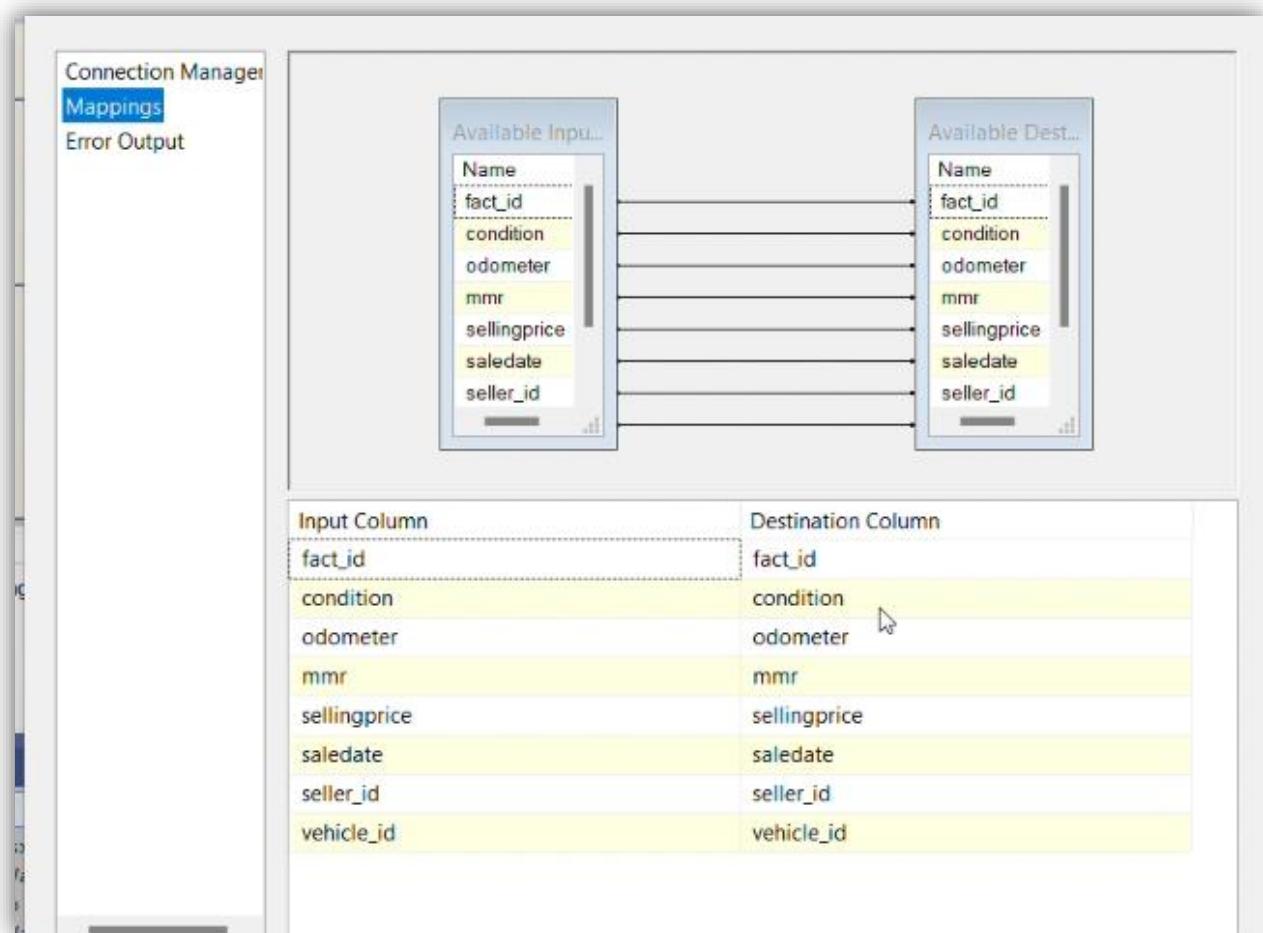
```
CREATE TABLE [Fact] (
    [fact_id] int constraint fact_id primary key,
    [condition] int,
    [odometer] int,
    [mmr] int,
```

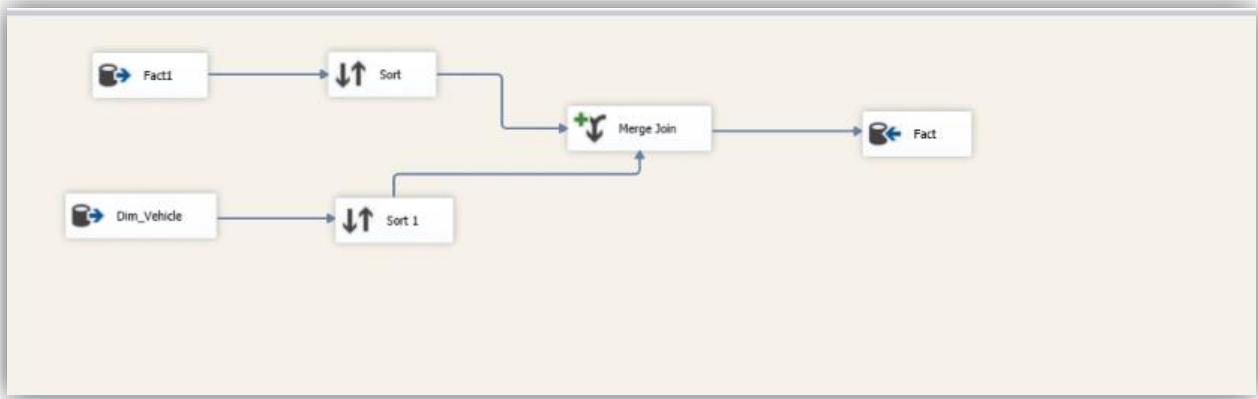
```

[sellingprice] int,
[saledate] datetime,
[seller_id] int,
[vehicle_id] int
)

```

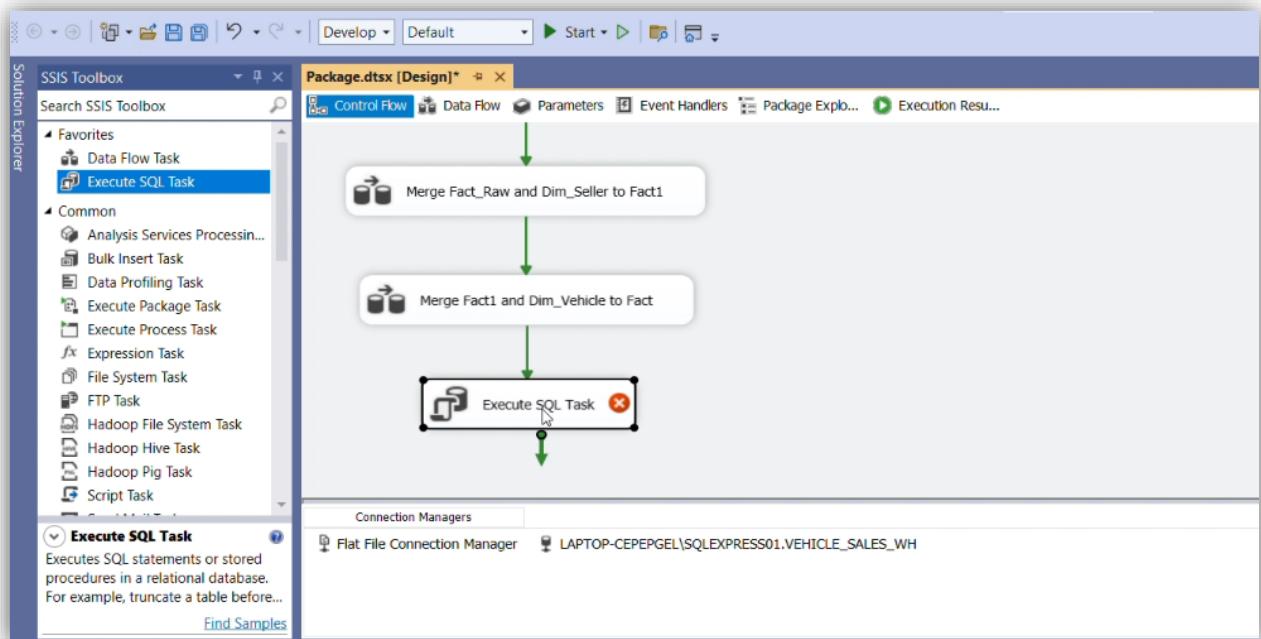
Chọn mục Mappings để xem xét việc ánh xạ các cột dữ liệu



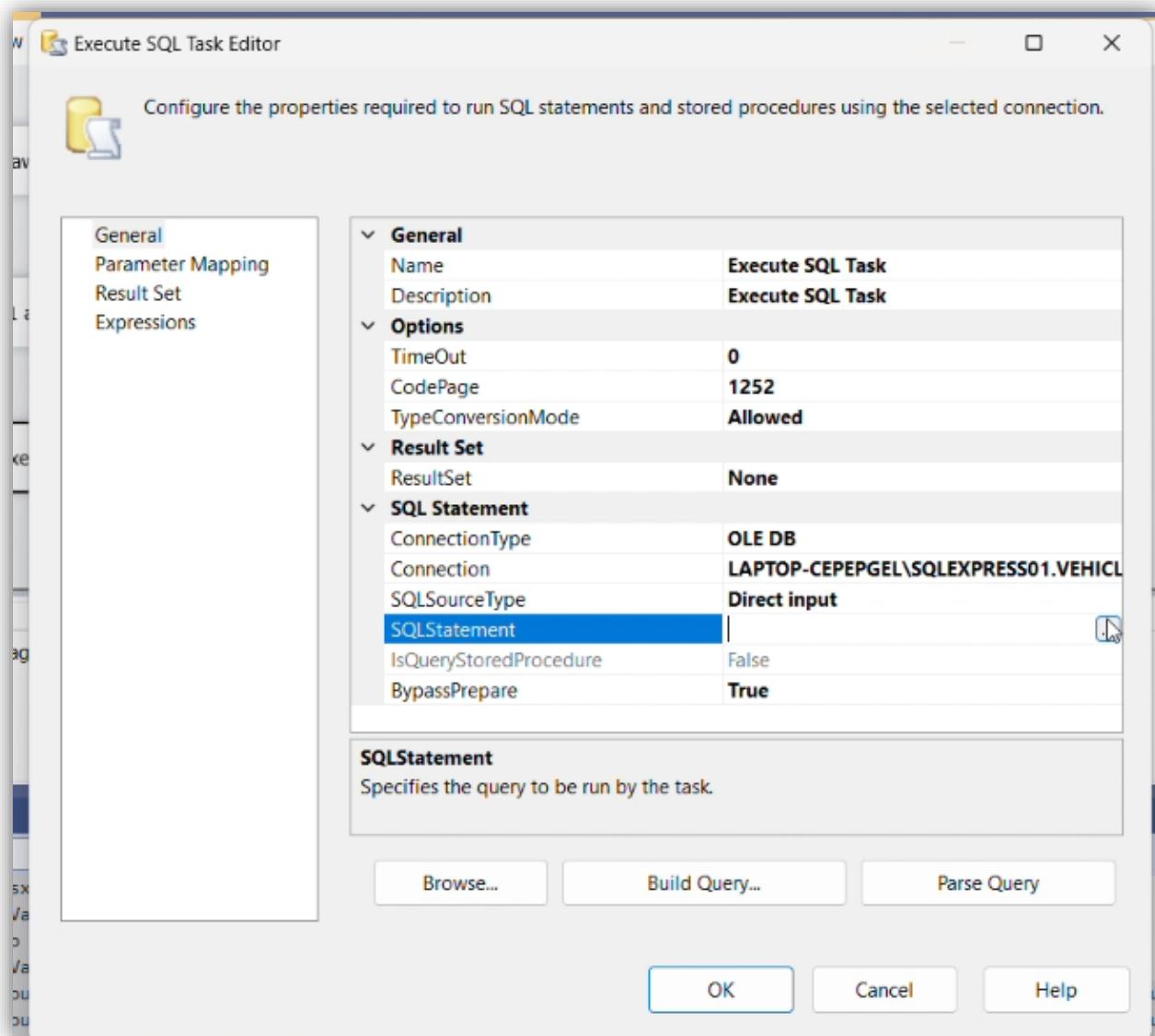


#### 2.4.4.3. Tạo khóa ngoại từ bảng Fact đến các Dimension

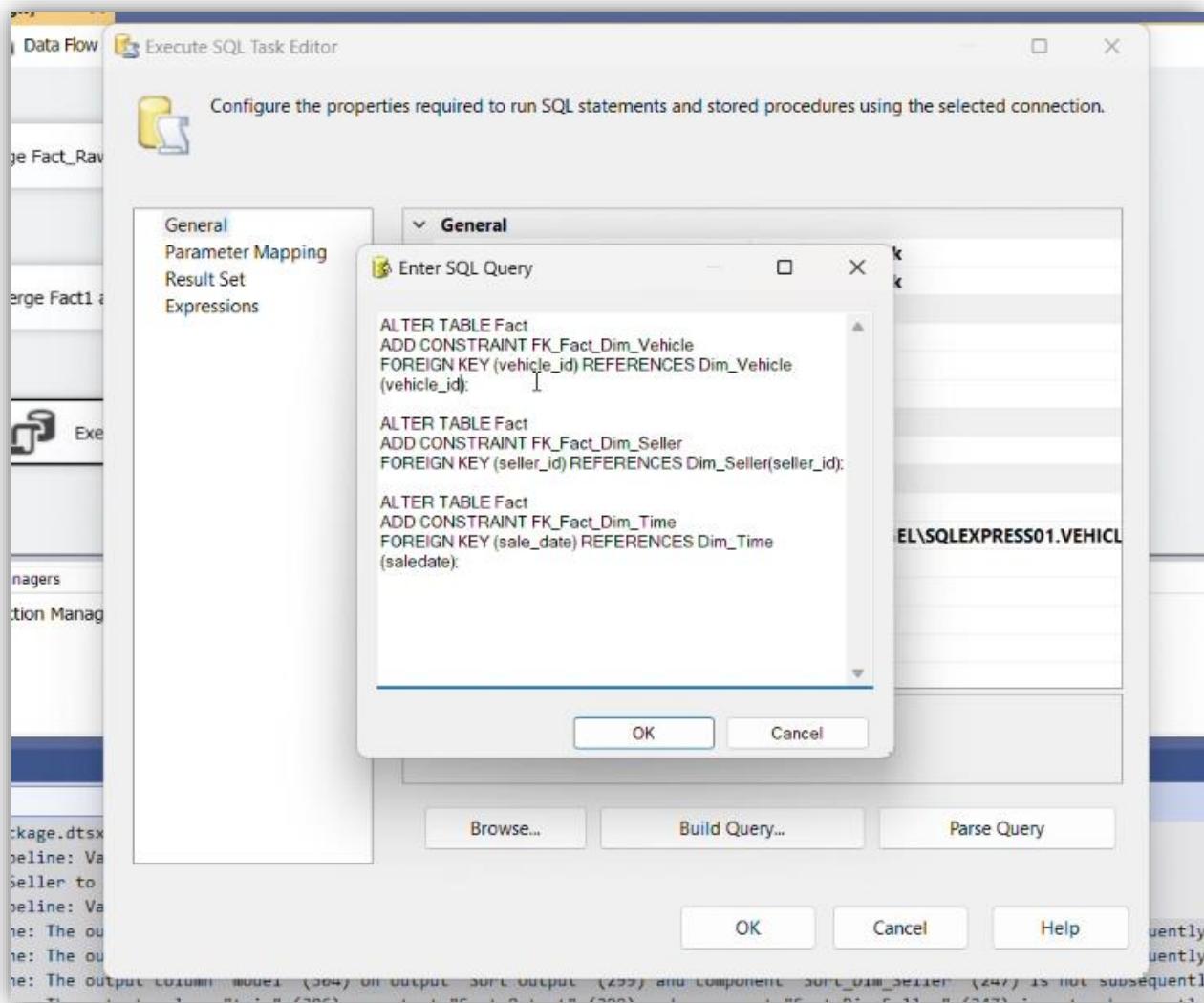
**Bước 1.** Tạo một Execute SQL Task để thực thi các câu lệnh SQL tạo các khóa ngoại từ các Dimension đến bảng Fact.



**Bước 2.** Nhấn chuột phải vào Execute SQL Task này và chọn Edit. Ở ô Connection, chọn connection đã thiết lập đến data warehouse trong SQL Server



**Bước 3.** Ở ô SQLStatement, thêm các câu truy vấn SQL thực hiện tạo các khóa ngoại từ các Dimension đến bảng Fact.



Các câu lệnh:

ALTER TABLE Fact

ADD CONSTRAINT FK\_Fact\_Dim\_Vehicle

FOREIGN KEY (vehicle\_id) REFERENCES Dim\_Vehicle(vehicle\_id);

ALTER TABLE Fact

ADD CONSTRAINT FK\_Fact\_Dim\_Seller

FOREIGN KEY (seller\_id) REFERENCES Dim\_Seller(seller\_id);

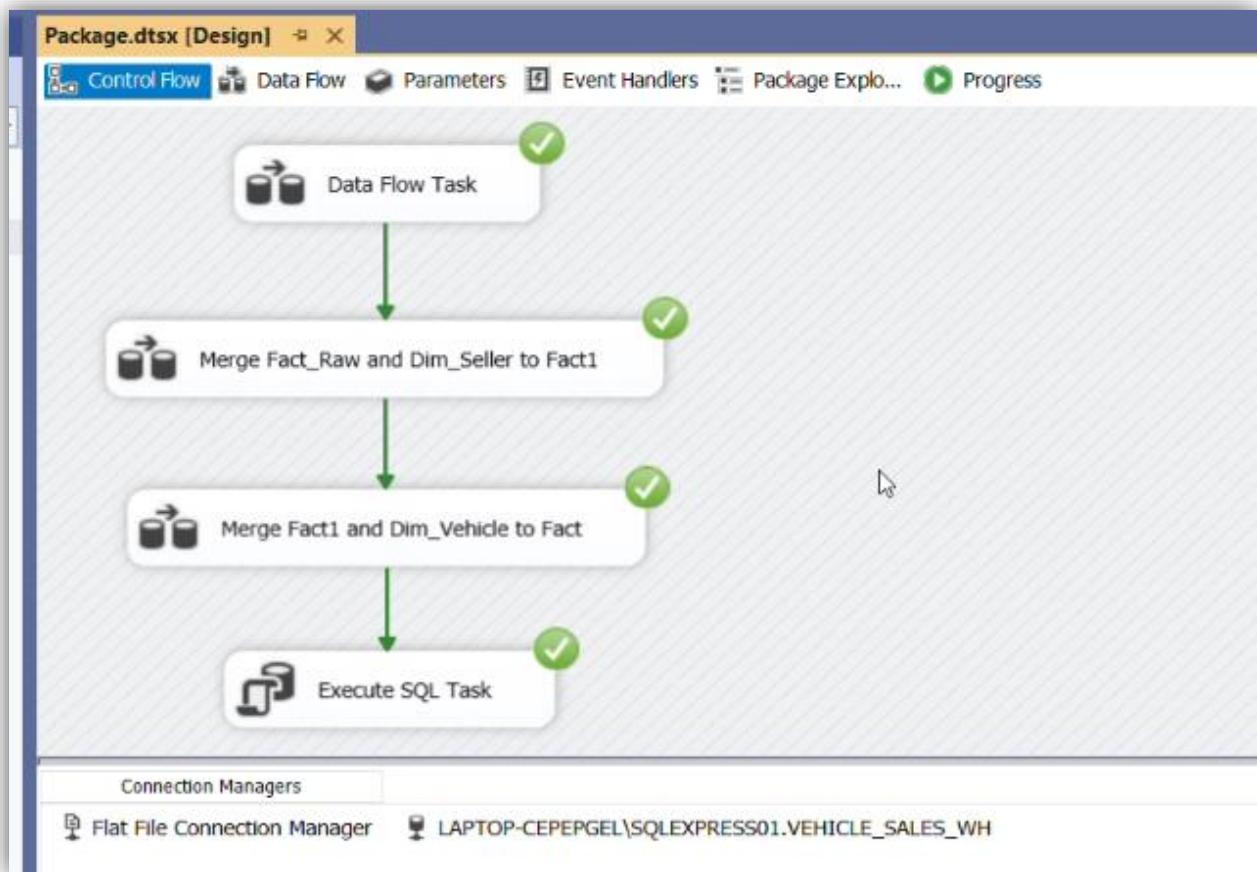
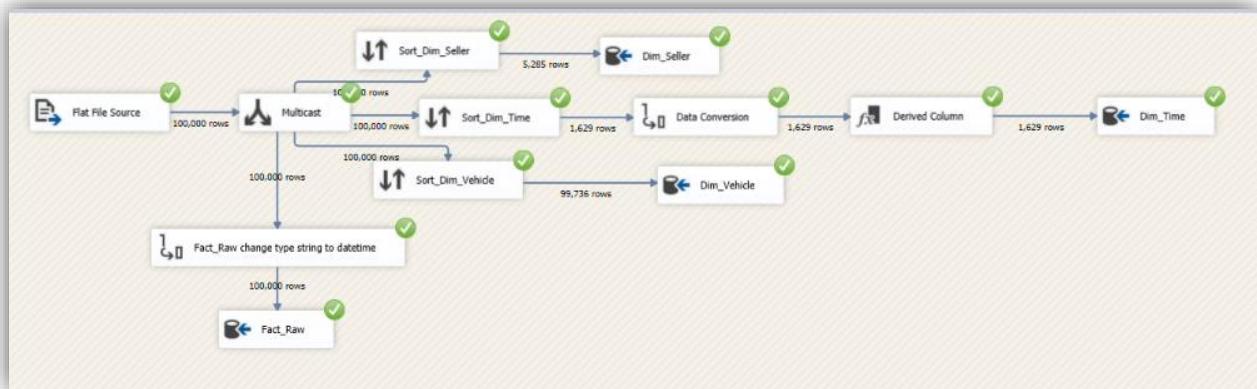
ALTER TABLE Fact

ADD CONSTRAINT FK\_Fact\_Dim\_Time

FOREIGN KEY (saledate) REFERENCES Dim\_Time(saledate);

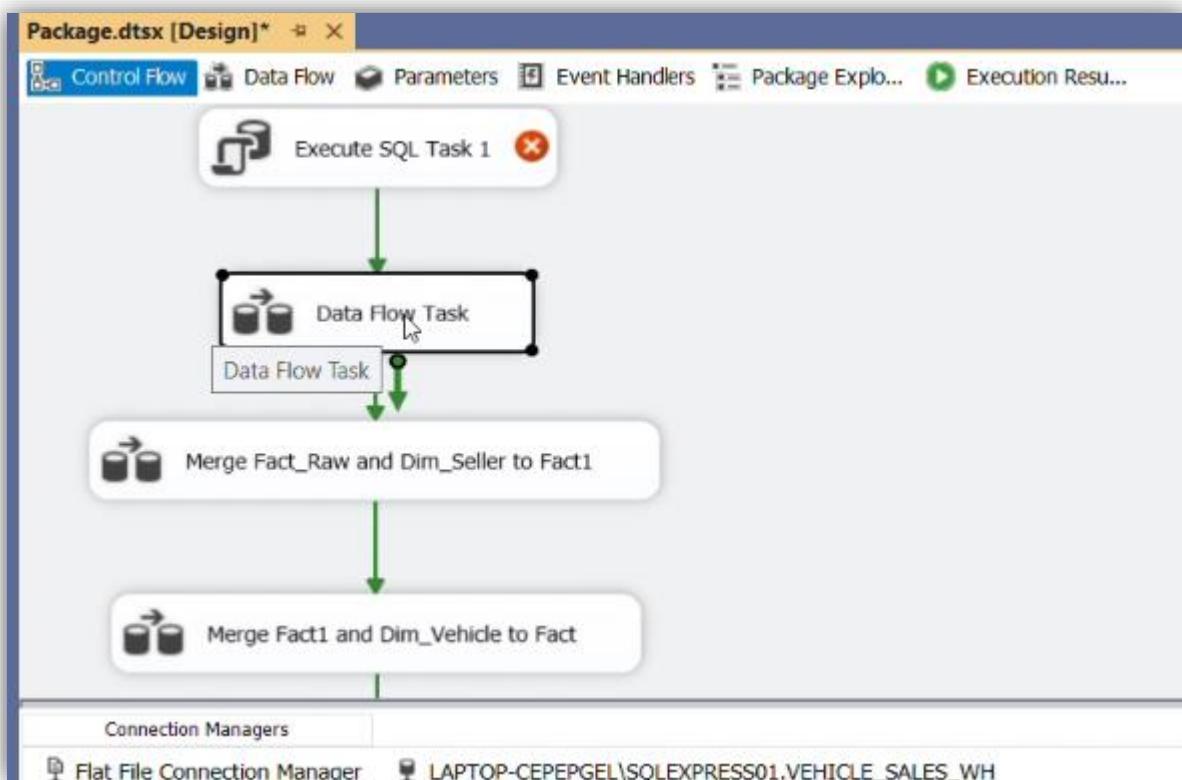
Nhấn OK để hoàn tất quá trình.

#### 2.4.5. Chạy dự án SSIS

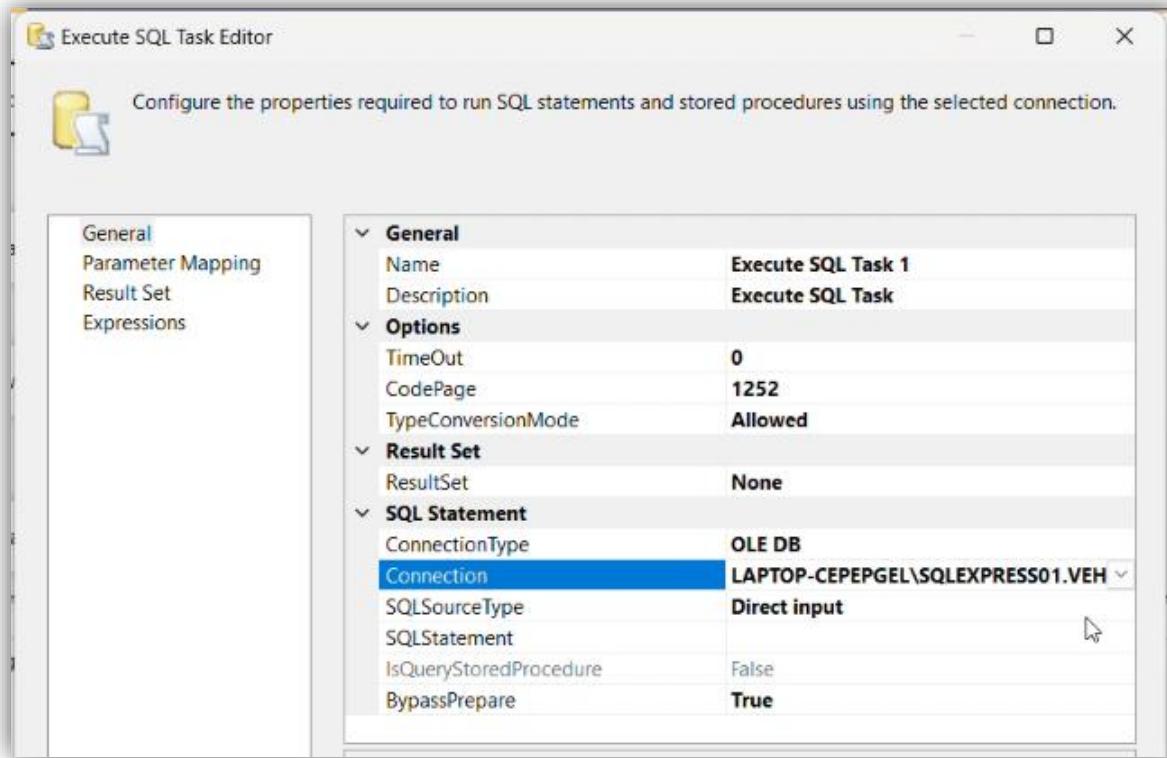


Nhằm thực hiện nhiệm vụ đảm bảo dữ liệu mới hoàn toàn (không bị chồng chéo dữ liệu cũ) mỗi khi chạy lại project, trước quá trình chia bảng Fact và các Dimension.

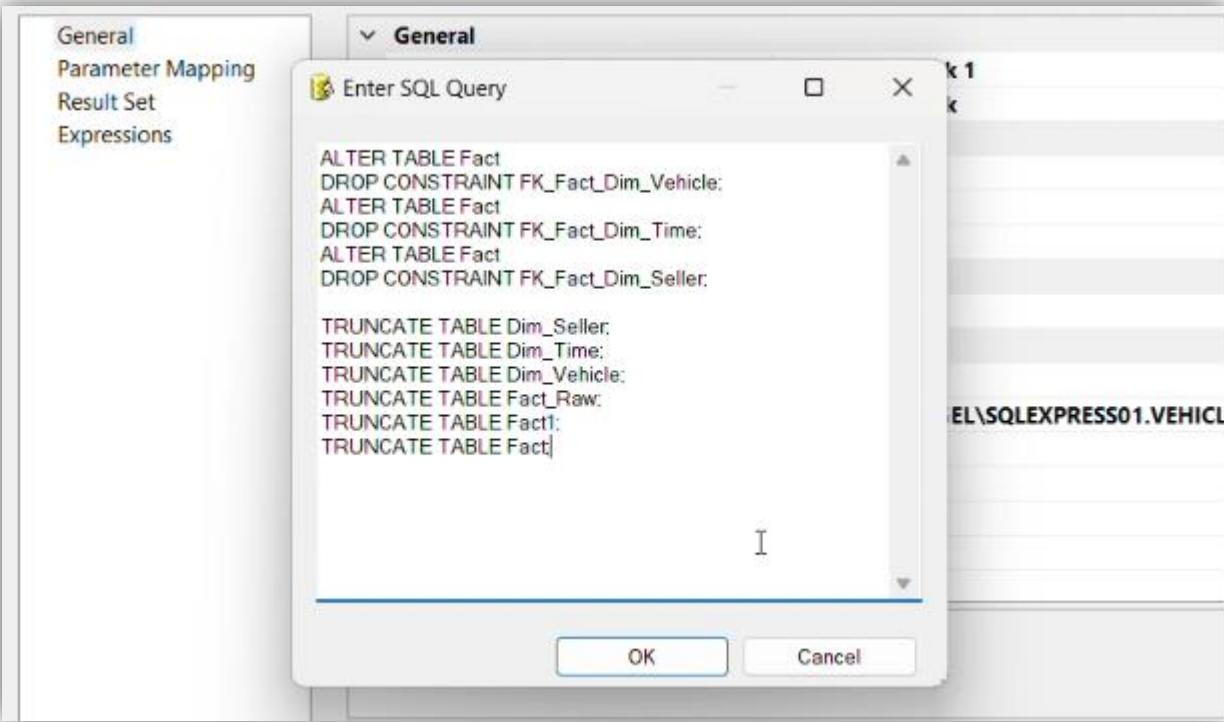
**Bước 1.** Thêm vào một Execute SQL Task.



**Bước 2.** Nhấn chuột phải vào Execute SQL Task này và chọn Edit. Ở ô Connection, chọn connection đã thiết lập đến data warehouse trong SQL Server

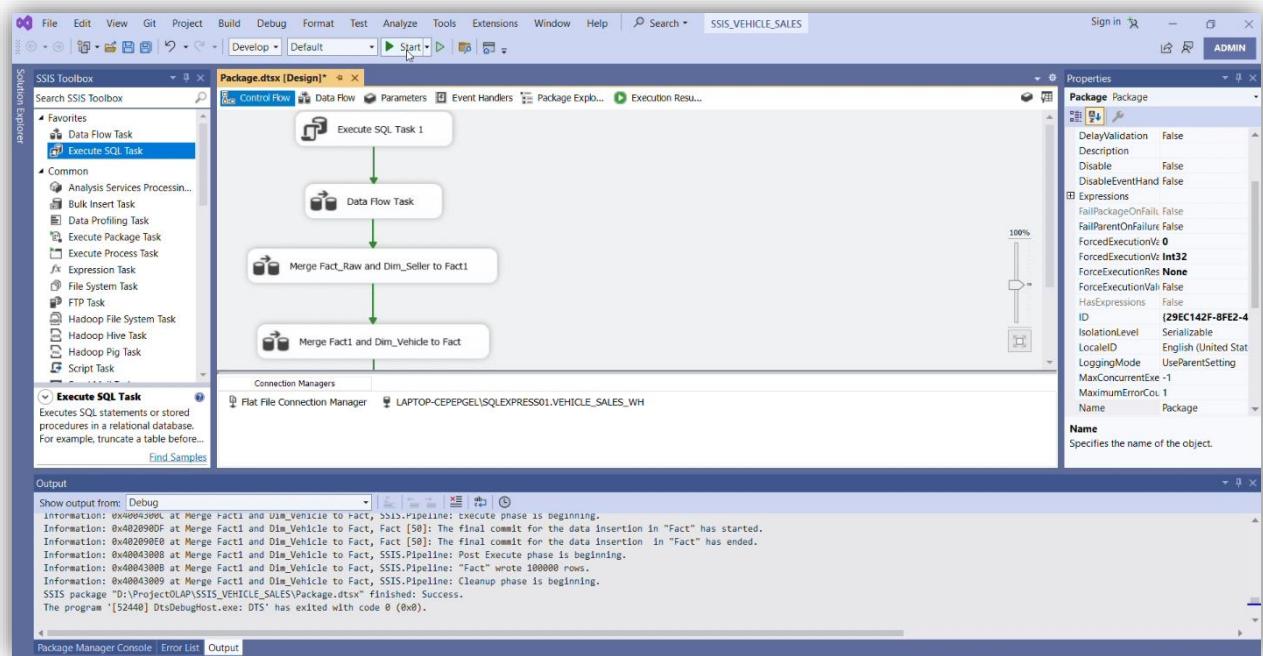


**Bước 3.** Ở ô SQLStatement, thêm các câu truy vấn SQL thực hiện xóa khoá ngoại và dữ liệu cũ trong các bảng mỗi khi chạy lại project.

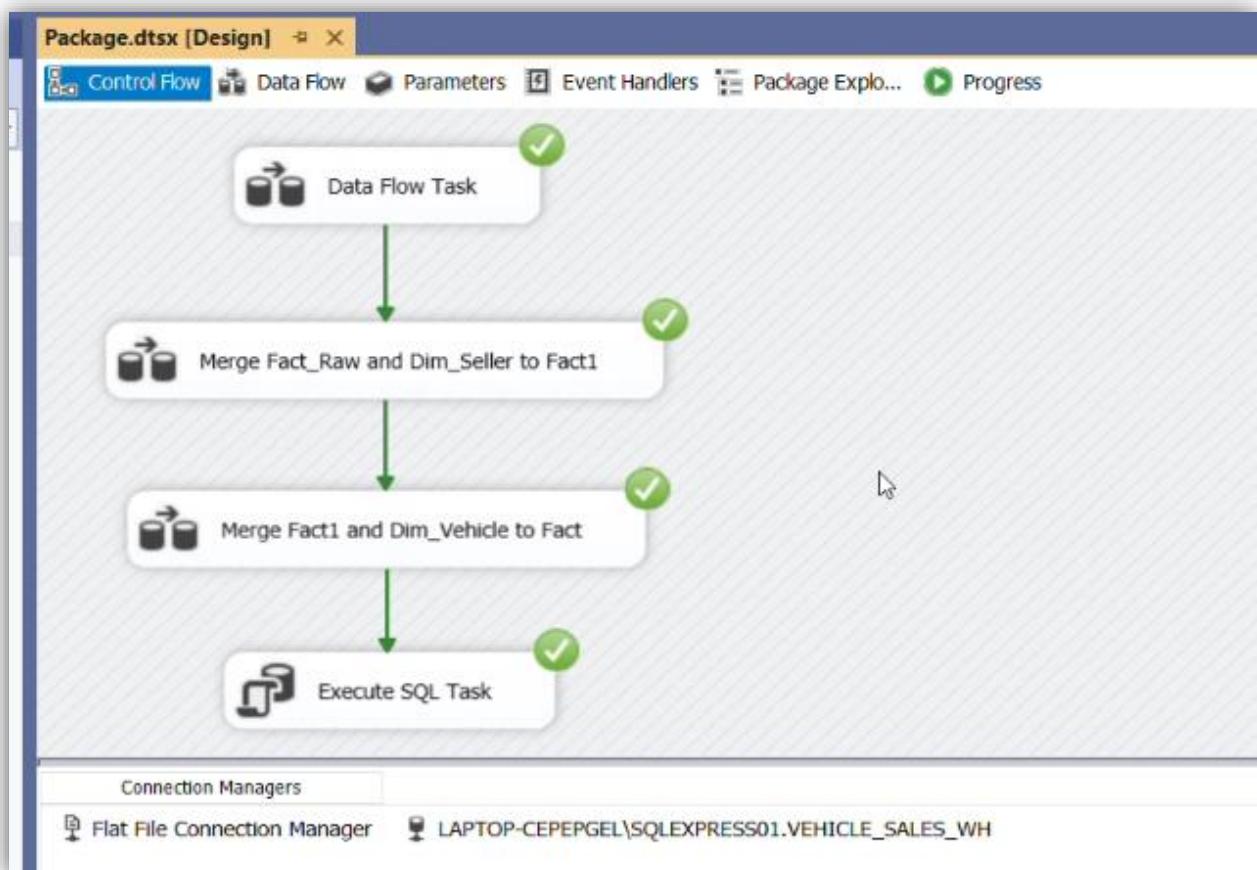
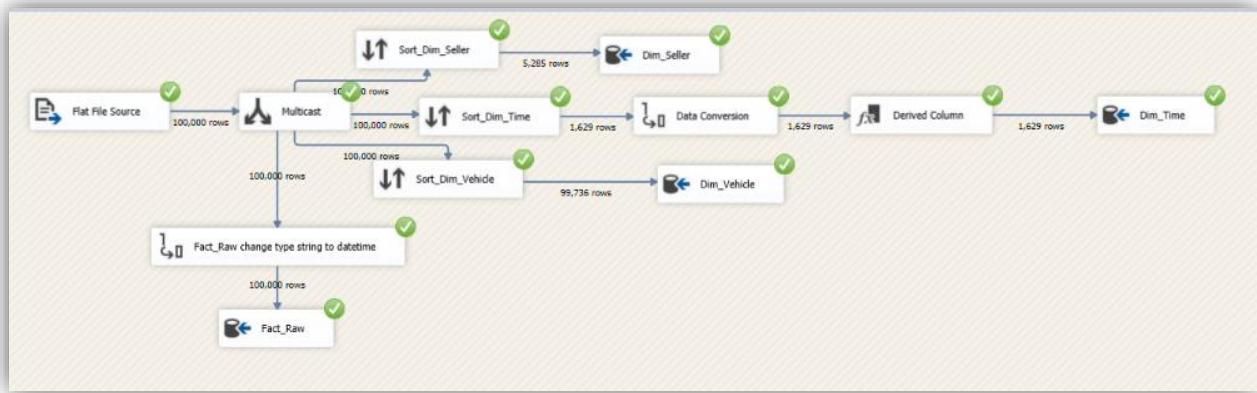


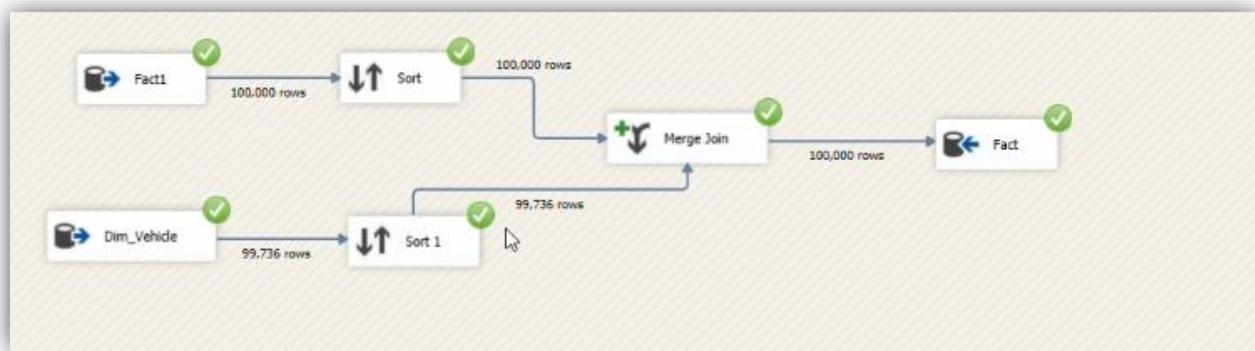
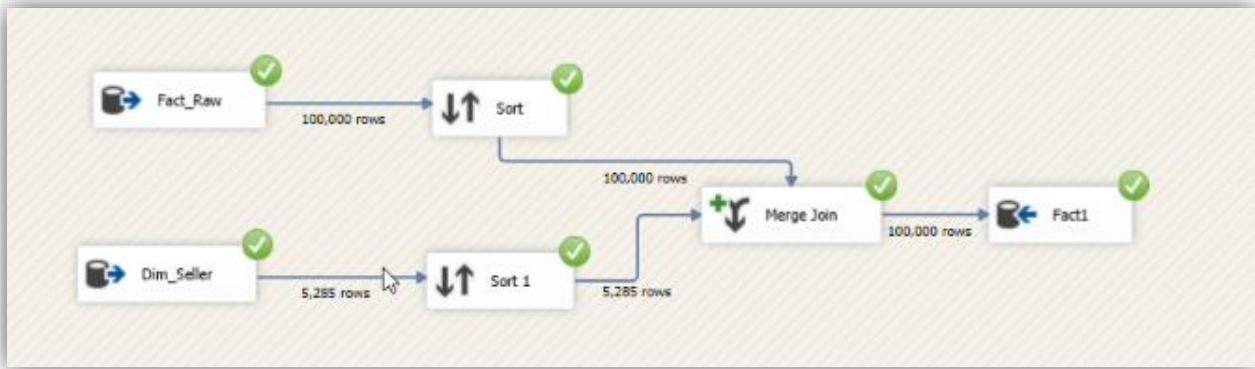
Nhấn OK để hoàn tất quá trình.

**Bước 4.** Nhấn nút Start trên thanh menu để tiến hành chạy project.



Kết quả chạy project:





#### 2.4.6. Kiểm tra dữ liệu các bảng

Kiểm tra dữ liệu bảng Dim\_Seller

	seller_id	seller
▶	1	1 cochran o...
	2	143 auto sal...
	3	159191 can...
	4	1st advanta...
	5	1st capital fi...
	6	1st choice a...
	7	1st commer...
	8	1st liberty fcu
	9	1st mid am...
	10	1st national ...
	11	22nd street ...
	12	231 car sale...
	13	281 truck sa...
	14	3 amigos au...
	15	3 in one aut...
	16	3 line motor...
	17	3:16 auto sa...
	18	355 toyota
	19	4 h auto sal...
	20	46 vans & tr...
	21	77th street ...
	22	800 loan m...
	23	800cash247
	24	888 motors
	25	91 automoti...
	26	9209 0851 q...
	27	924 auto co...
	28	a & a auto ...
	29	a & a auto s...

Kiểm tra dữ liệu bảng Dim\_Time

	saledate	date	month	year	hour	minute	quarter
1	2014-01-01 01:15:00.000	1	1	2014	1	15	1
2	2014-01-01 02:00:00.000	1	1	2014	2	0	1
3	2014-01-01 03:00:00.000	1	1	2014	3	0	1
4	2014-01-02 00:29:00.000	2	1	2014	0	29	1
5	2014-01-02 01:00:00.000	2	1	2014	1	0	1
6	2014-01-05 01:00:00.000	5	1	2014	1	0	1
7	2014-01-05 01:30:00.000	5	1	2014	1	30	1
8	2014-01-05 16:00:00.000	5	1	2014	16	0	1
9	2014-01-06 01:00:00.000	6	1	2014	1	0	1
10	2014-01-06 01:15:00.000	6	1	2014	1	15	1
11	2014-01-06 01:30:00.000	6	1	2014	1	30	1
12	2014-01-06 01:50:00.000	6	1	2014	1	50	1
13	2014-01-06 04:00:00.000	6	1	2014	4	0	1
14	2014-01-06 04:30:00.000	6	1	2014	4	30	1
15	2014-01-06 16:00:00.000	6	1	2014	16	0	1
16	2014-01-07 01:20:00.000	7	1	2014	1	20	1
17	2014-01-07 02:00:00.000	7	1	2014	2	0	1
18	2014-01-07 02:30:00.000	7	1	2014	2	30	1
19	2014-01-07 03:00:00.000	7	1	2014	3	0	1
20	2014-01-07 05:00:00.000	7	1	2014	5	0	1
21	2014-01-08 01:00:00.000	8	1	2014	1	0	1
22	2014-01-08 01:30:00.000	8	1	2014	1	30	1
23	2014-01-08 02:00:00.000	8	1	2014	2	0	1
24	2014-01-08 03:00:00.000	8	1	2014	3	0	1
25	2014-01-08 03:30:00.000	8	1	2014	3	30	1
26	2014-01-08 04:00:00.000	8	1	2014	4	0	1
27	2014-01-12 02:00:00.000	12	1	2014	2	0	1
28	2014-01-12 05:00:00.000	12	1	2014	5	0	1
29	2014-01-13 03:00:00.000	13	1	2014	3	0	1
30	2014-01-13 04:00:00.000	13	1	2014	4	0	1
31	2014-01-14 03:00:00.000	14	1	2014	3	0	1

Kiểm tra dữ liệu bảng Dim\_Vehicle

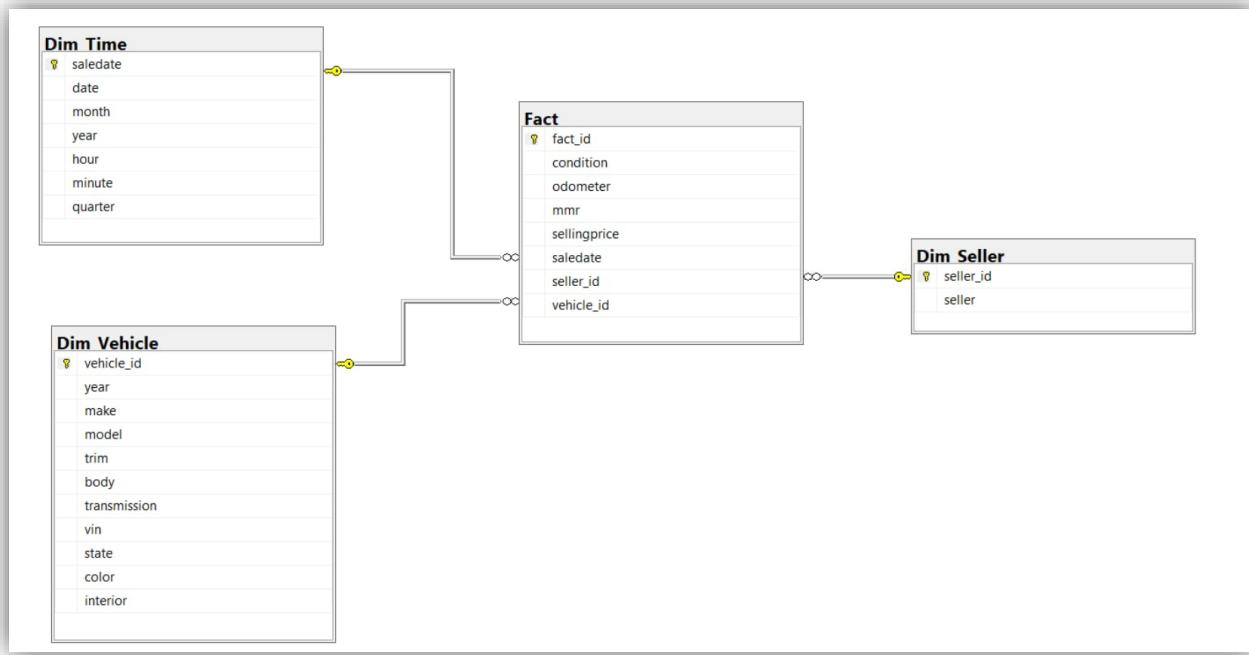
Results Messages

	vehicle_id	year	make	model	trim	body	transmission	vin	state	color	interior
1	1	1990	Cadillac	DeVille	Base	Sedan	automatic	1g6cd5333l4326699	pa	white	purple
2	2	1990	Chevrolet	Corvette	Base	Hatchback	automatic	1g1yy238715108284	oh	red	red
3	3	1990	Honda	Accord	EX	Sedan	automatic	jhmc7661lc036504	tx	gray	tan
4	4	1990	Honda	Accord	EX	Sedan	automatic	jhmc7665lc099475	az	gray	gray
5	5	1990	Lexus	LS 400	Base	Sedan	automatic	jt8uf11e5l0041243	ca	white	tan
6	6	1990	Toyota	Camry	Deluxe	Sedan	automatic	jt2av21exl0345999	wa	blue	blue
7	7	1990	Toyota	Corolla	Base	Sedan	automatic	jt2ae91a0l3295341	or	red	gray
8	8	1991	Chevrolet	Corvette	Base	Convertible	automatic	1g1yy3383m5102652	ca	black	black
9	9	1991	Honda	Accord	SE	Sedan	automatic	jhmc7683mc096190	md	gray	blue
10	10	1991	Lexus	LS 400	Base	Sedan	automatic	jt8uf11e5m0094008	ca	silver	silver
11	11	1991	Lexus	LS 400	Base	Sedan	automatic	jt8uf11e9m0096702	ga	black	gray
12	12	1991	Mazda	MX-5 Miata	Base	Convertible	automatic	jm1na3518m1211663	ga	red	gray
13	13	1991	Mercedes-Benz	500-Class	500SL	Convertible	automatic	wdbfa66e5mf030237	fl	blue	beige
14	14	1991	Mercedes-Benz	500-Class	500SL	Convertible	automatic	wdbfa66e7mf035858	ca	silver	gray
15	15	1991	Nissan	Maxima	GXE	Sedan	automatic	jn1hj01p5mt524361	fl	black	gray
16	16	1991	Toyota	Camry	Deluxe	Sedan	automatic	4t1av21e5mu453282	md	white	beige
17	17	1991	Toyota	Camry	Deluxe	Sedan	automatic	4t1av21e8mu304686	fl	white	blue
18	18	1991	Toyota	Camry	Deluxe	Sedan	automatic	jt2av21e3m3460184	ca	gray	gray
19	19	1991	Toyota	Corolla	Deluxe	Sedan	automatic	1nxa94a7mz175093	tx	white	blue
20	20	1991	Toyota	Corolla	Deluxe	Sedan	automatic	jt2ae94axm3443766	wa	silver	gray
21	21	1992	Buick	Park Aven...	Base	Sedan	automatic	1g4cw53l3n1604038	fl	gold	beige
22	22	1992	Buick	Park Aven...	Base	Sedan	automatic	1g4cw53l6n1658420	ga	blue	blue
23	23	1992	Cadillac	DeVille	Base	Sedan	automatic	1g6cd53b4n4338628	nc	silver	beige
24	24	1992	Cadillac	DeVille	Base	Sedan	automatic	1g6cd53b5n4310353	ca	blue	blue
25	25	1992	Cadillac	DeVille	Base	Sedan	automatic	1g6cd53b6n4290503	ca	white	blue
26	26	1992	Cadillac	DeVille	Base	Sedan	automatic	1g6cd53b8n4322741	ne	bur...	purple
27	27	1992	Ford	Explorer	XLT	SUV	automatic	1fmdu34x1nua10739	fl	gold	beige
28	28	1992	Honda	Accord	EX	Sedan	automatic	1hgcb767xna011644	ca	white	blue
29	29	1992	Honda	Accord	LX	Sedan	automatic	1hgcb7659na127352	fl	white	blue

Kiểm tra dữ liệu bảng Fact

	fact_id	condition	odometer	mmr	sellingprice	saledate	seller_id	vehicle_id
1	1	5	16639	20500	21500	2014-12-16 04:30:00.000	2657	99555
2	2	5	9393	20800	21500	2014-12-16 04:30:00.000	2657	99553
3	3	45	1331	31900	30000	2015-01-14 20:30:00.000	1814	85150
4	4	41	14282	27500	27750	2015-01-28 20:30:00.000	5111	99706
5	5	43	2641	66000	67000	2014-12-18 04:30:00.000	1814	85358
6	6	1	5554	15350	10900	2014-12-30 04:00:00.000	1669	99654
7	7	34	14943	69000	65000	2014-12-17 04:30:00.000	4747	85368
8	8	2	28617	11900	9800	2014-12-16 05:00:00.000	1669	85777
9	9	42	9557	32100	32250	2014-12-18 04:00:00.000	317	84995
10	10	3	4809	26300	17500	2015-01-19 20:00:00.000	1372	85609
11	11	48	14414	47300	49750	2014-12-16 04:30:00.000	1454	85024
12	12	48	2034	15150	17700	2014-12-16 04:00:00.000	2662	99399
13	13	2	5559	15350	12000	2015-01-13 04:00:00.000	1669	99319
14	14	5	14634	20600	21500	2014-12-16 04:30:00.000	2657	99505
15	15	2	11398	14750	14100	2014-12-23 04:00:00.000	1669	99665
16	16	49	7983	37100	40000	2014-12-18 04:30:00.000	320	85039
17	17	17	13441	17750	17000	2014-12-30 07:00:00.000	5162	85593
18	18	34	8819	68000	67200	2014-12-17 04:30:00.000	4747	85338
19	19	19	14538	24300	7200	2015-07-07 02:30:00.000	1669	99152
20	20	29	25969	34200	30000	2015-02-02 20:30:00.000	1814	85289
21	21	49	5826	24000	23750	2014-12-18 04:30:00.000	320	99110

#### 2.4.7. Lược đồ sau khi hoàn thành



## CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU

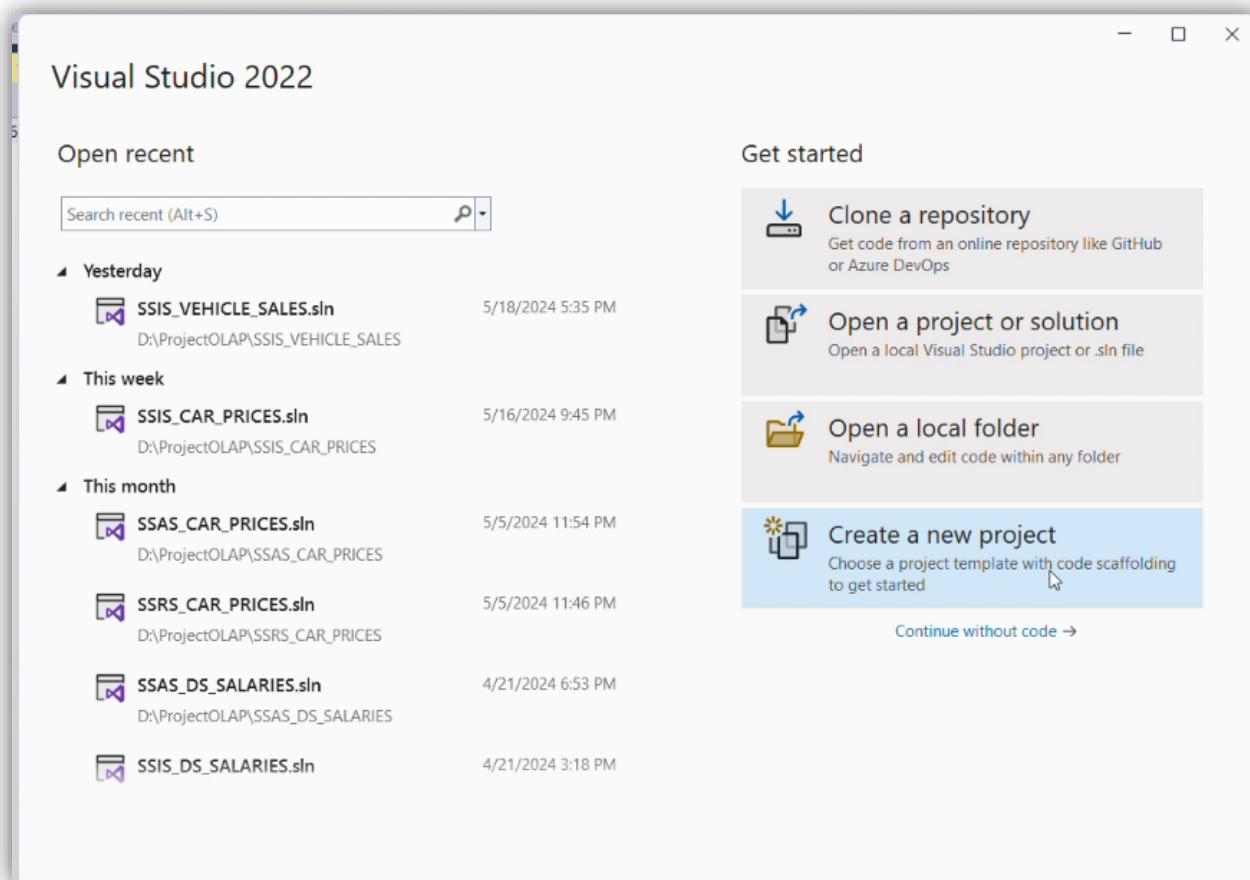
### 3.1. Chuẩn bị các công cụ

Để thực hiện được quá trình SSAS ta cần chuẩn bị và cài đặt các công cụ sau:

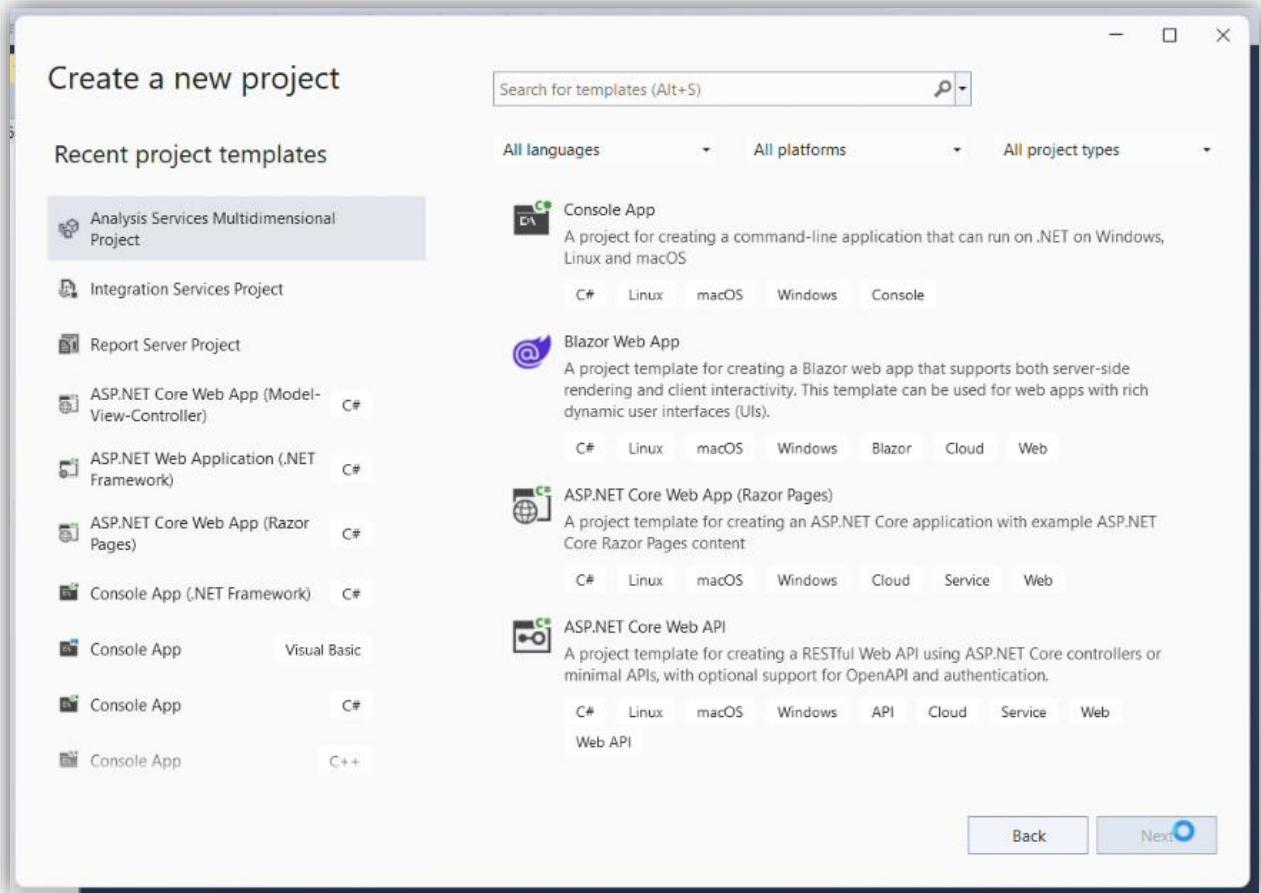
- Microsoft SQL Server có cài đặt Analysis Services.
- Microsoft Analysis Services Projects

### 3.2. Tạo mới Project SSAS

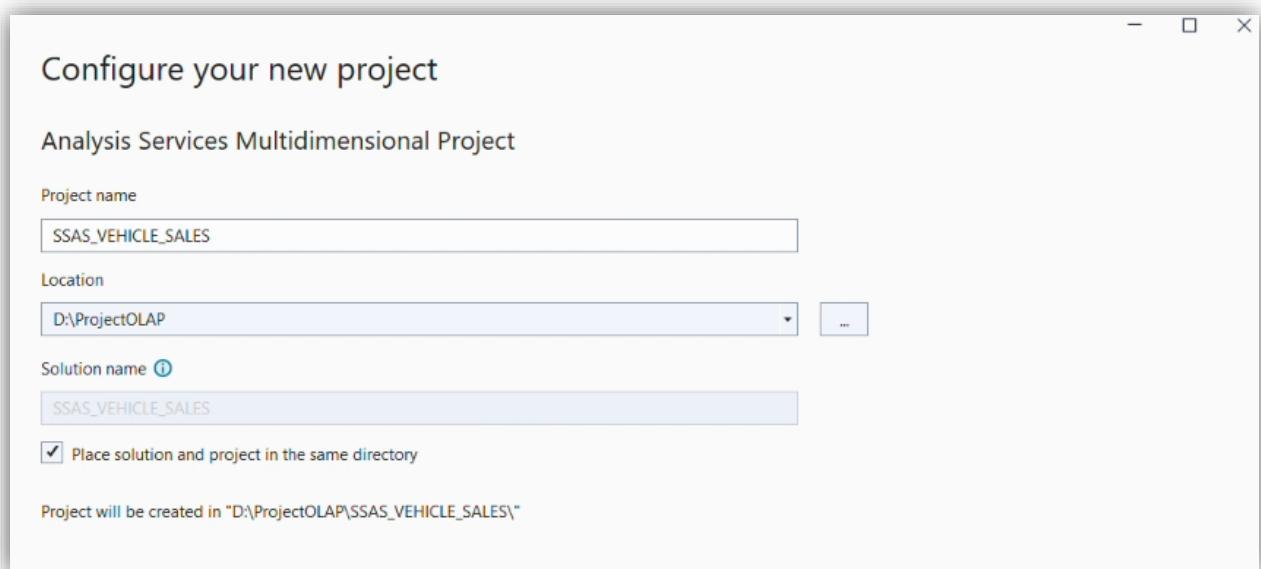
**Bước 1:** Mở Visual Studio và chọn “Create a new project”.



**Bước 2:** Chọn Analysis Services Multidimensional Project và chọn Next

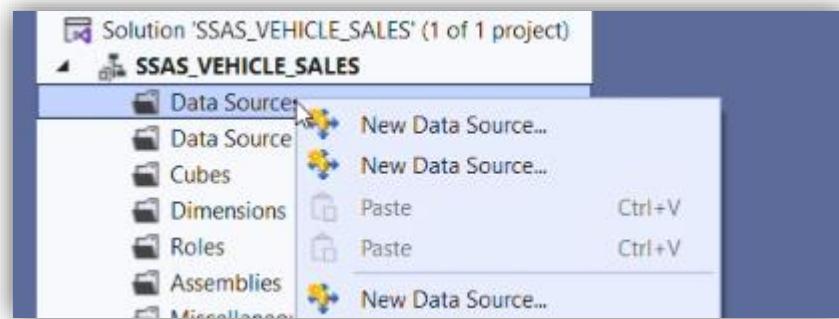


**Bước 3:** Đặt tên và thiết lập đường dẫn cho Project. Sau đó chọn Create.

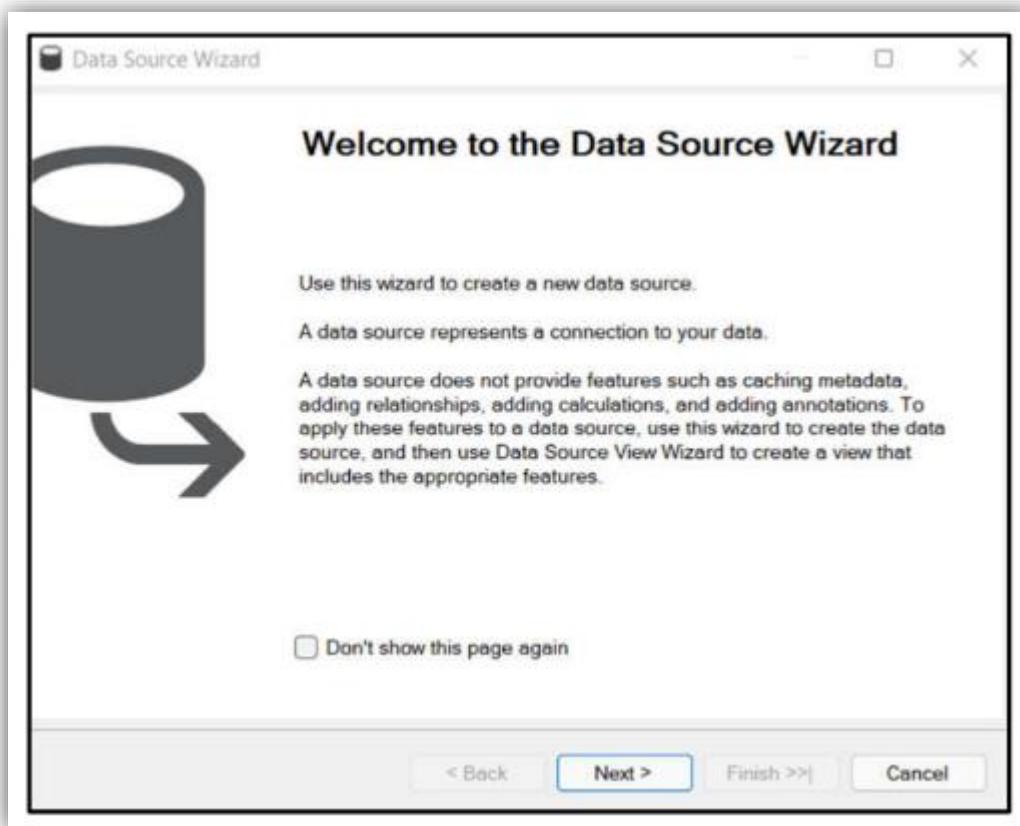


### 3.3. Xác định dữ liệu nguồn (Data Sources)

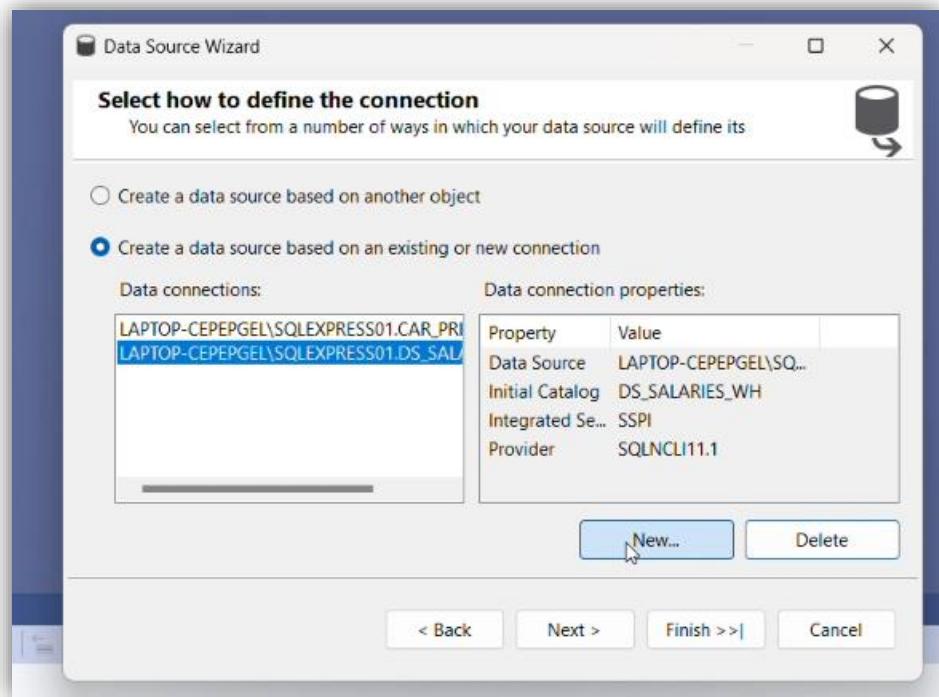
**Bước 1:** Tại Solution Explorer, ta click chuột phải vào thư mục Data Sources và chọn New Data Source.



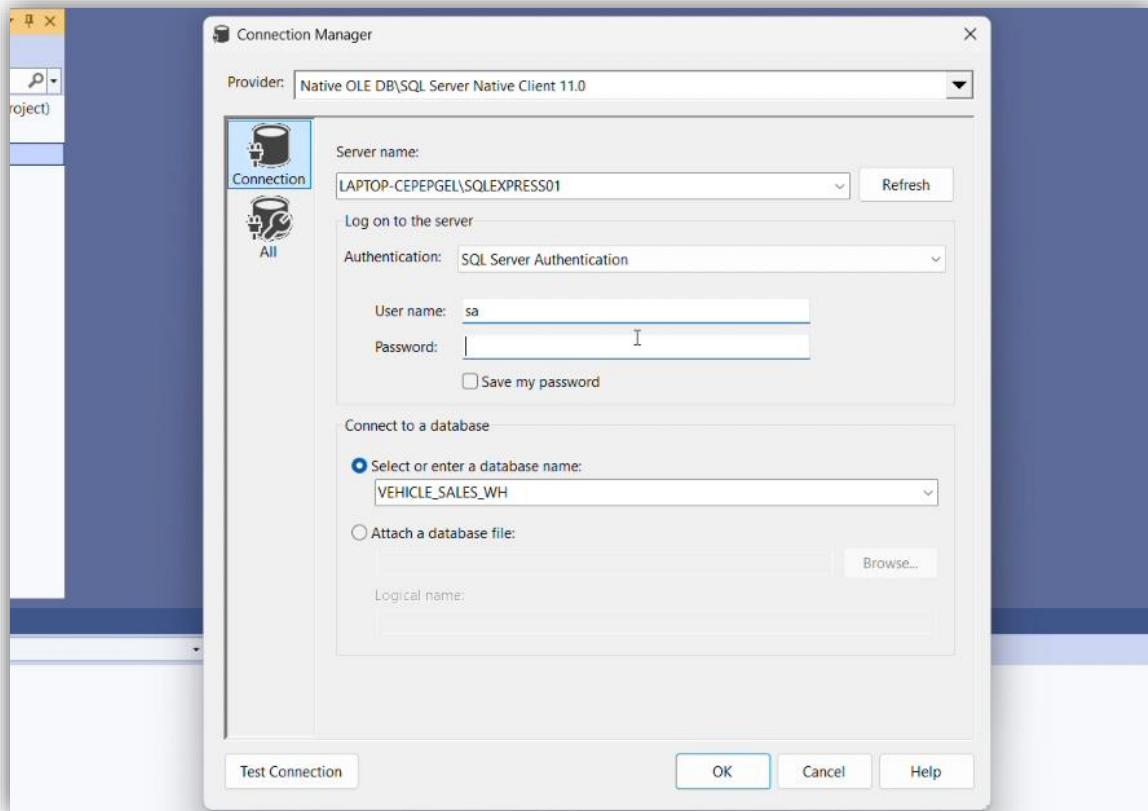
**Bước 2:** Hộp thoại Data Source Wizard xuất hiện, chọn Next để tiếp tục.



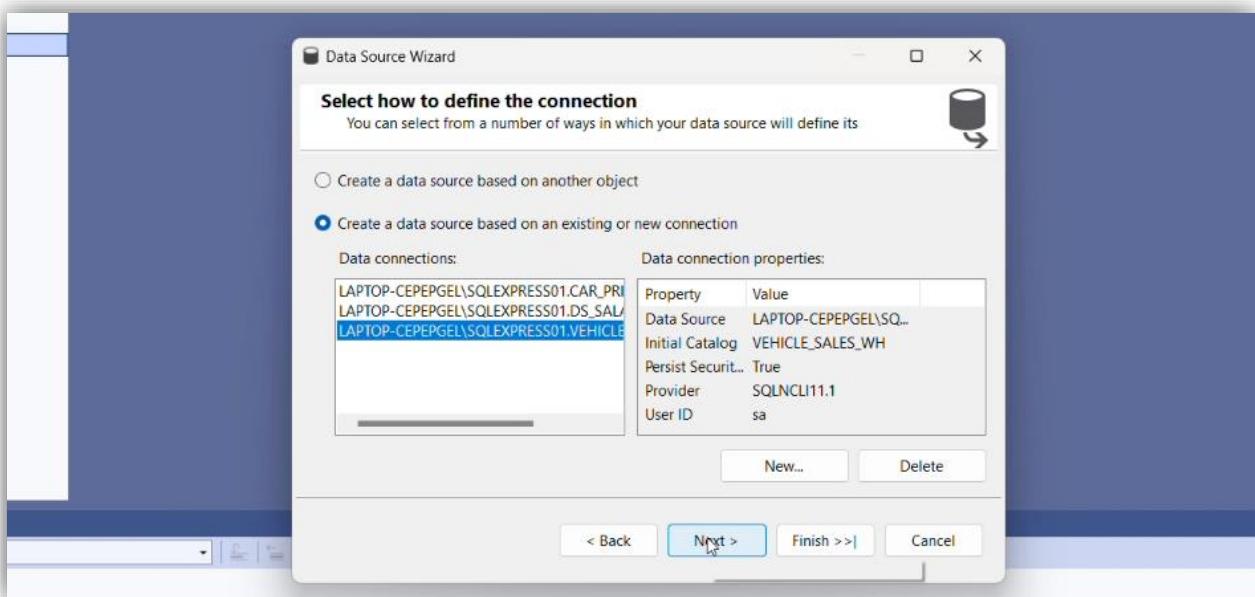
**Bước 3:** Chọn “Create a data source based on an existing or new connection” sau đó chọn New... để tạo kết nối với cơ sở dữ liệu đã được tạo từ quá trình SSIS.



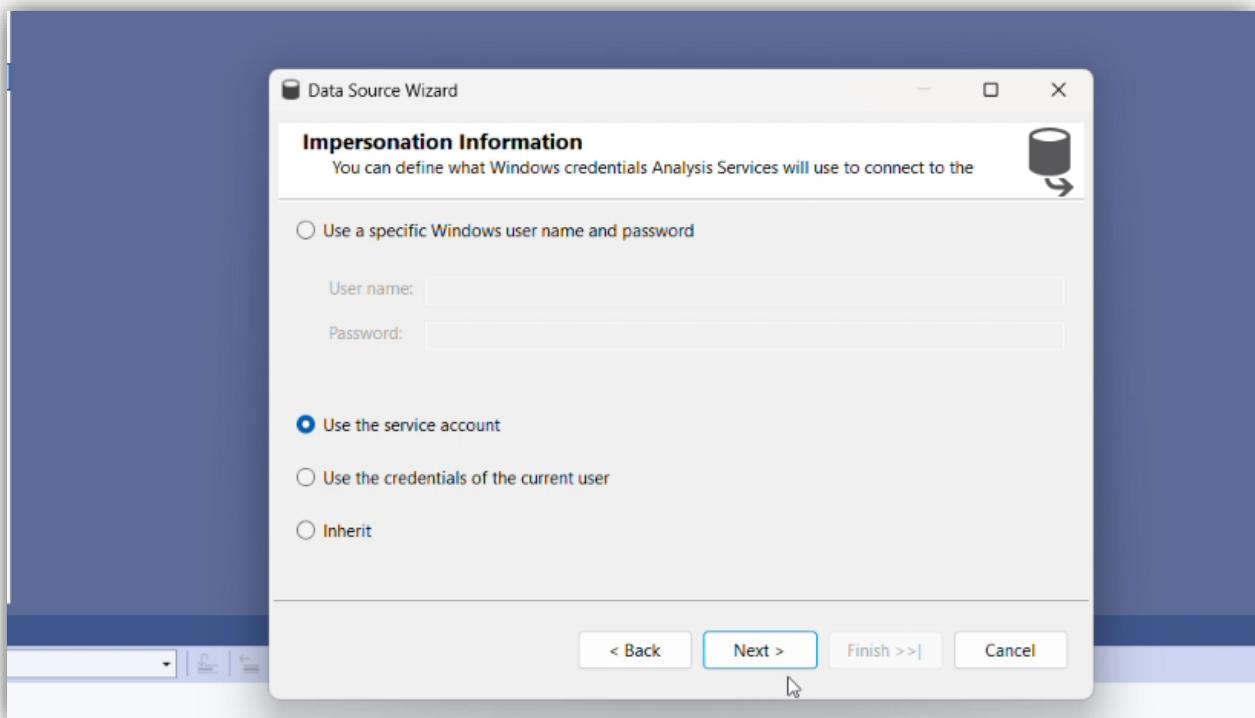
**Bước 4:** Hộp thoại Connection Manager xuất hiện, ta nhập Server name và chọn cơ sở dữ liệu mà ta đã tạo ra từ quá trình SSIS. Sau đó chọn OK.



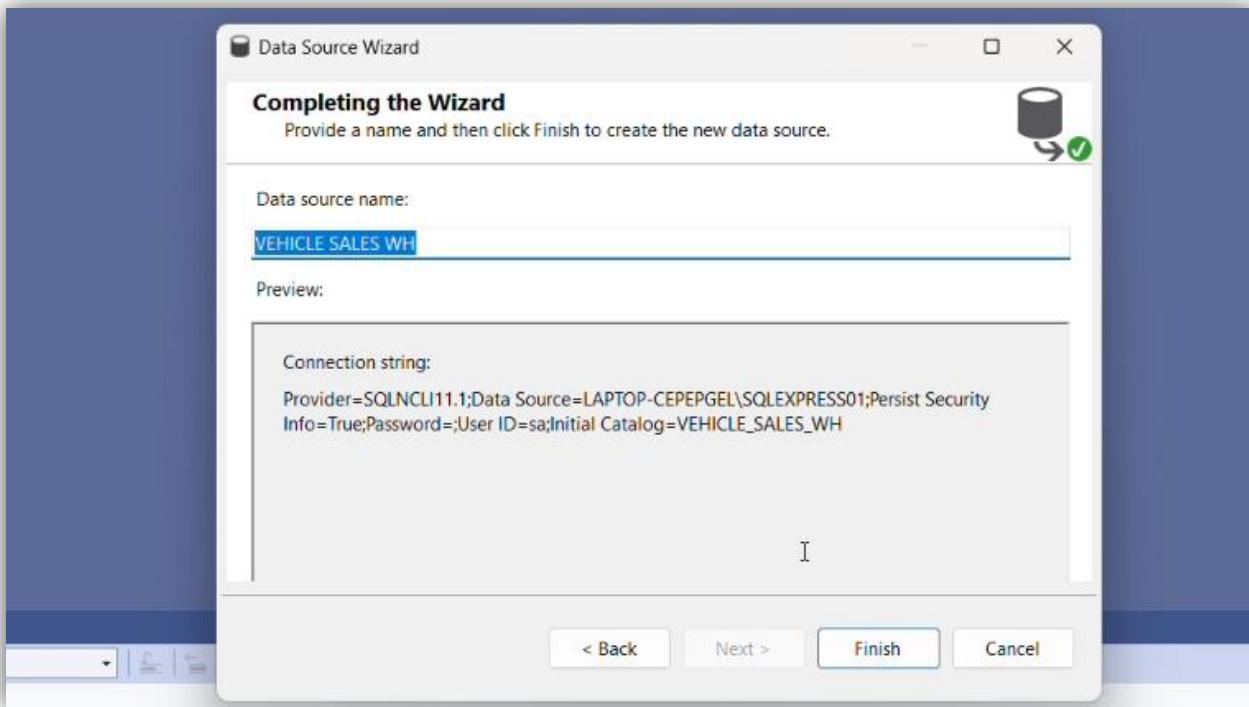
Chọn data source vừa tạo và chọn Next để tiếp tục.



**Bước 5:** Chọn “Use the service account”, sau đó chọn Next để tiếp tục.

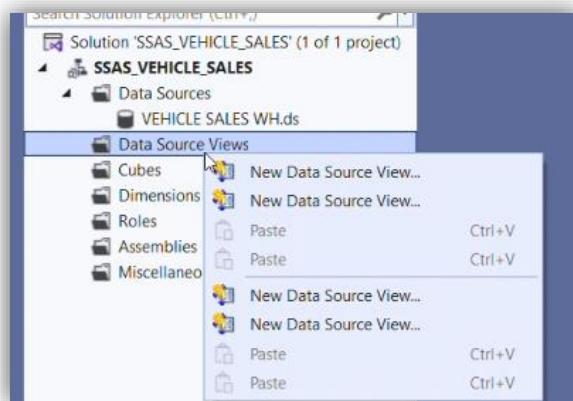


**Bước 6:** Cuối cùng ta chọn Finish để hoàn tất quy trình định nghĩa nguồn dữ liệu.

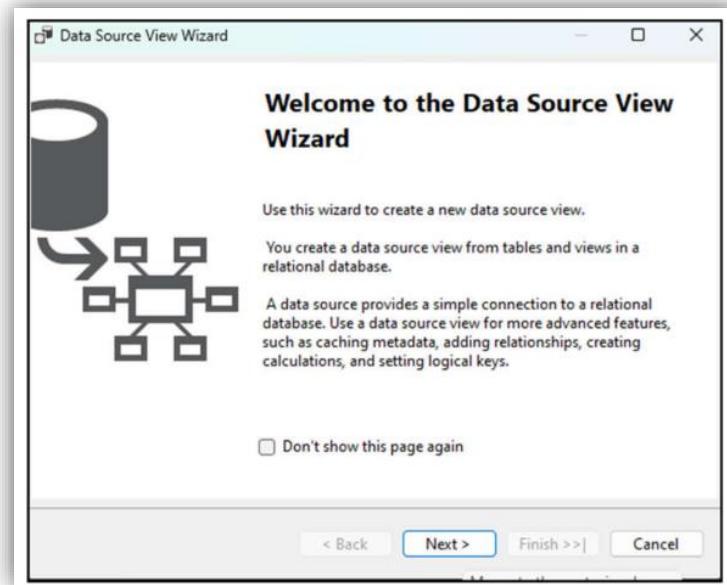


### 3.4. Xác định khung nhìn dữ liệu nguồn (Data Source Views)

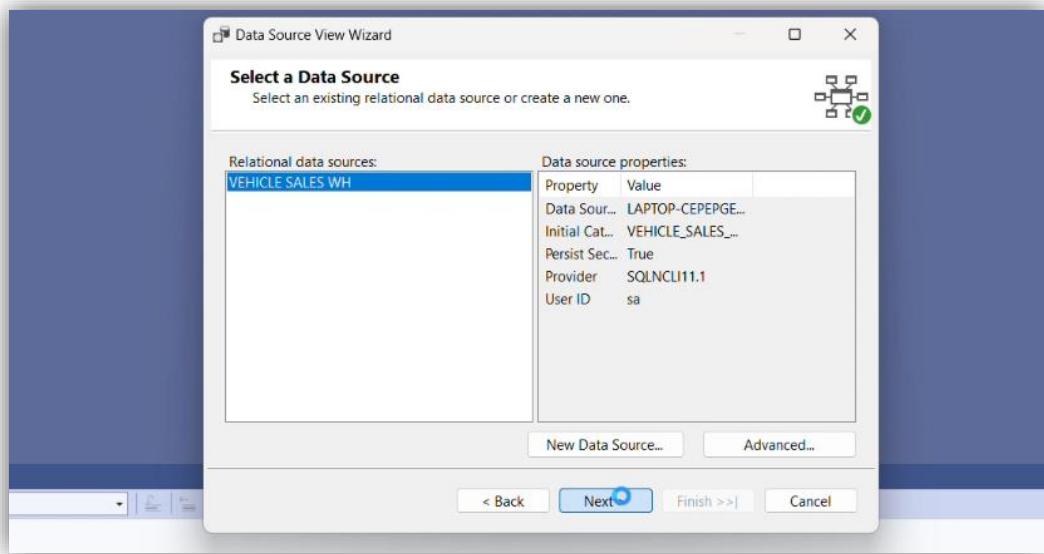
Bước 1: Tại Solution Explorer, ta click chuột phải vào thư mục Data Source Views và chọn New Data Source View.



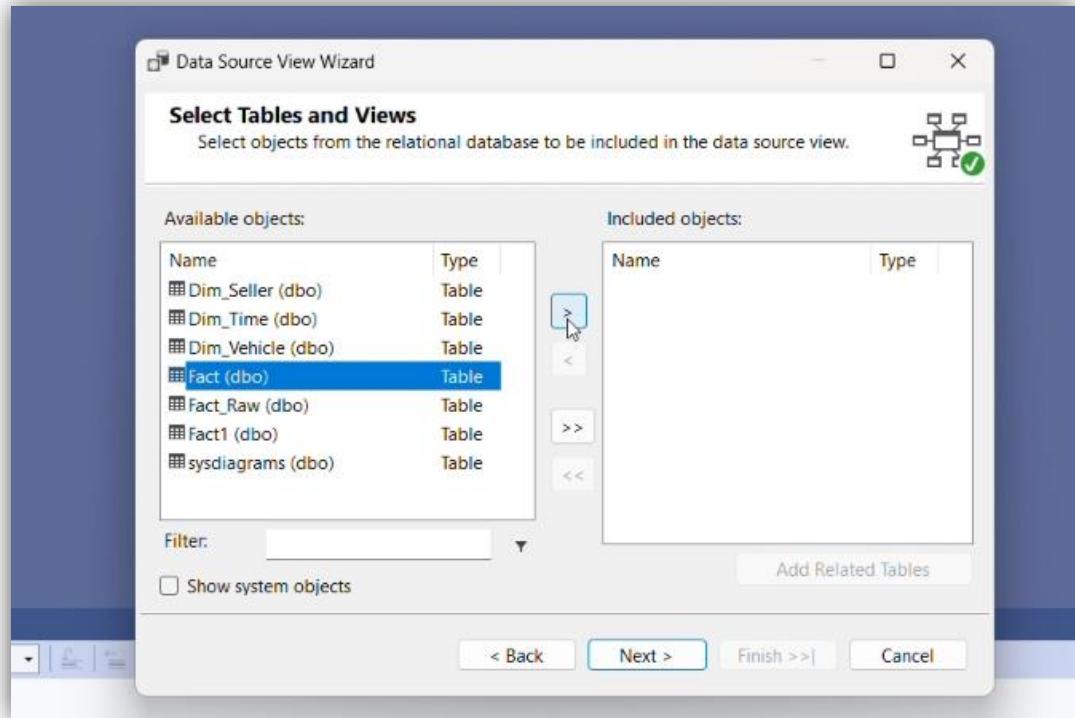
Bước 2: Hộp thoại Data Source View Wizard xuất hiện, chọn Next để tiếp tục.



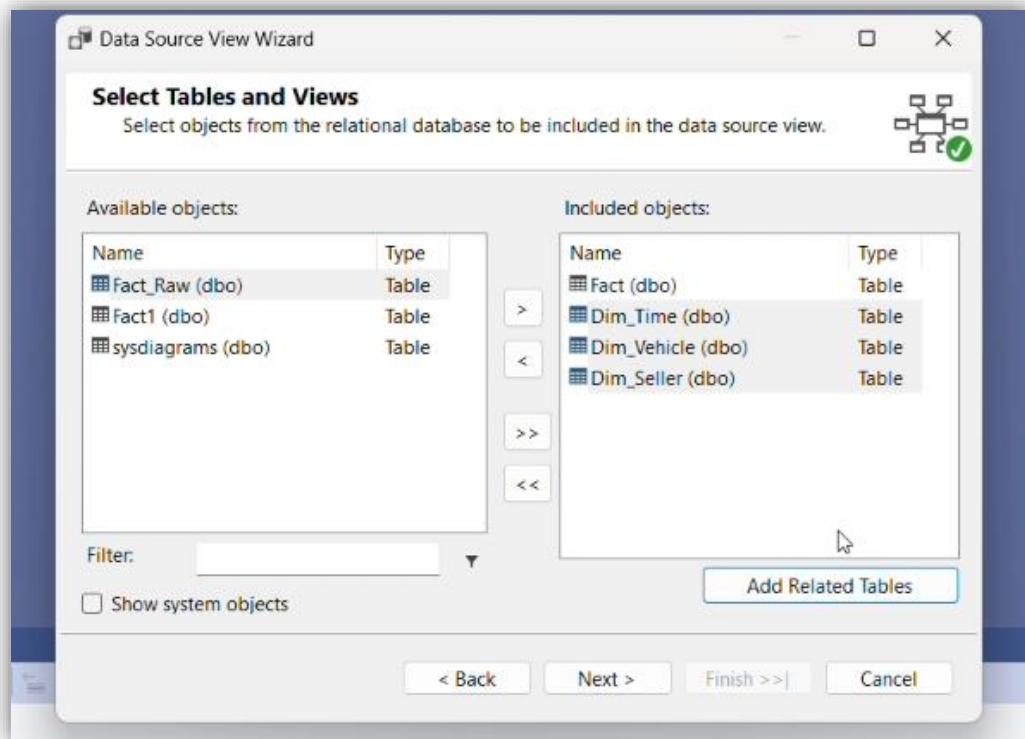
**Bước 3:** Chọn data source vừa tạo, sau đó chọn Next để tiếp tục.



**Bước 4:** Chọn bảng Fact, sau đó chọn nút > để thêm bảng Fact vào data source view.

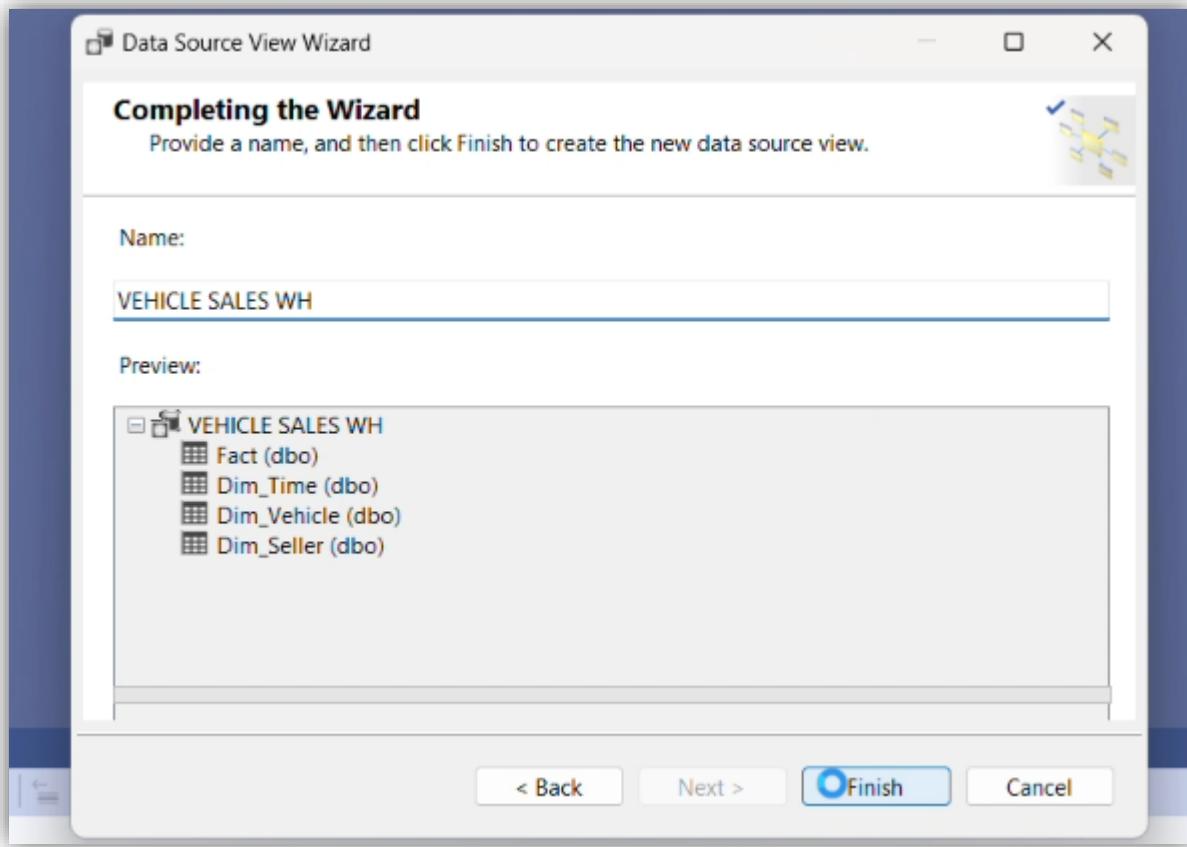


**Bước 5:** Tiếp theo, chọn nút Add Related Tables để thêm tất cả các bảng Dim vào data source view. Sau đó chọn Next để tiếp tục.

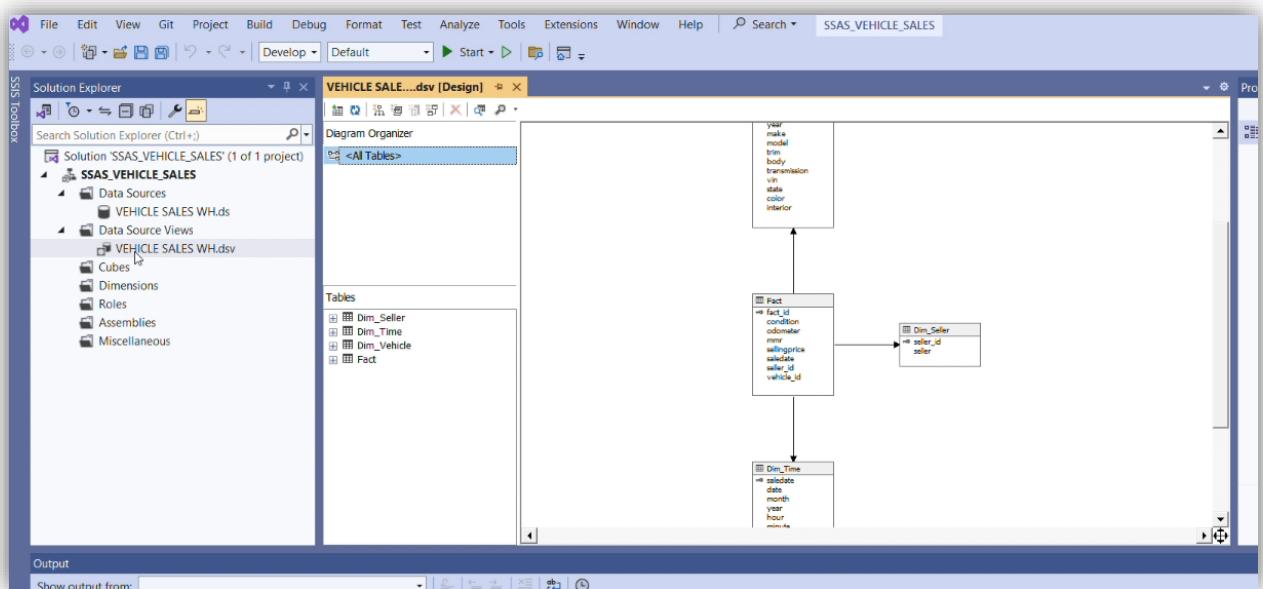


**Bước 6:** Chọn Finish để hoàn tất quá trình xác định khung nhìn dữ liệu nguồn

(Data Source View).

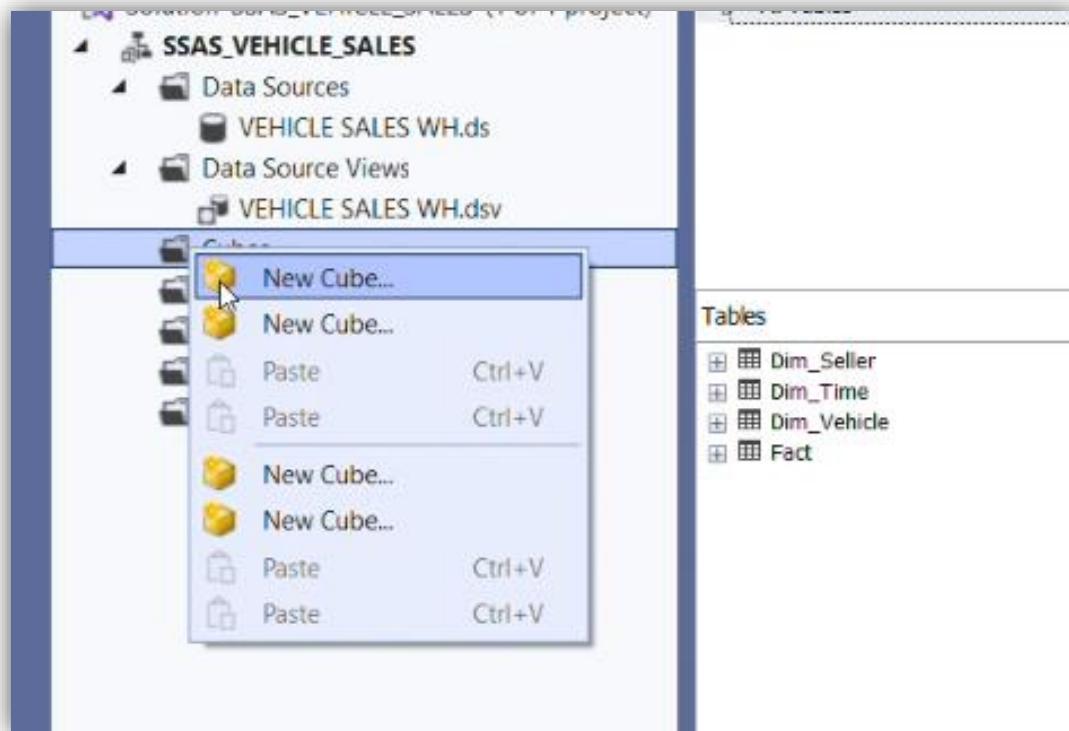


Sau khi kết thúc quá trình này, ta sẽ được data source view như hình sau

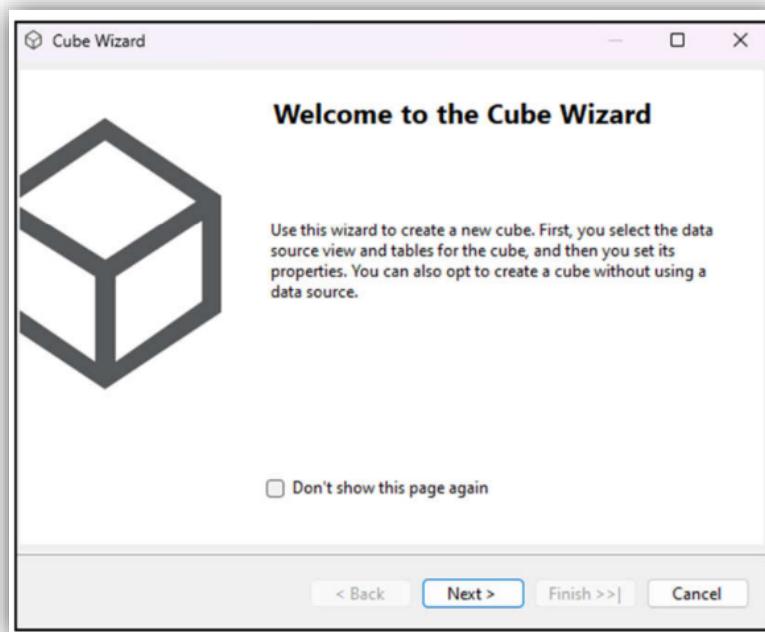


### 3.5. Xây dựng các khối (Cubes) và xác định các độ đo (Measures)

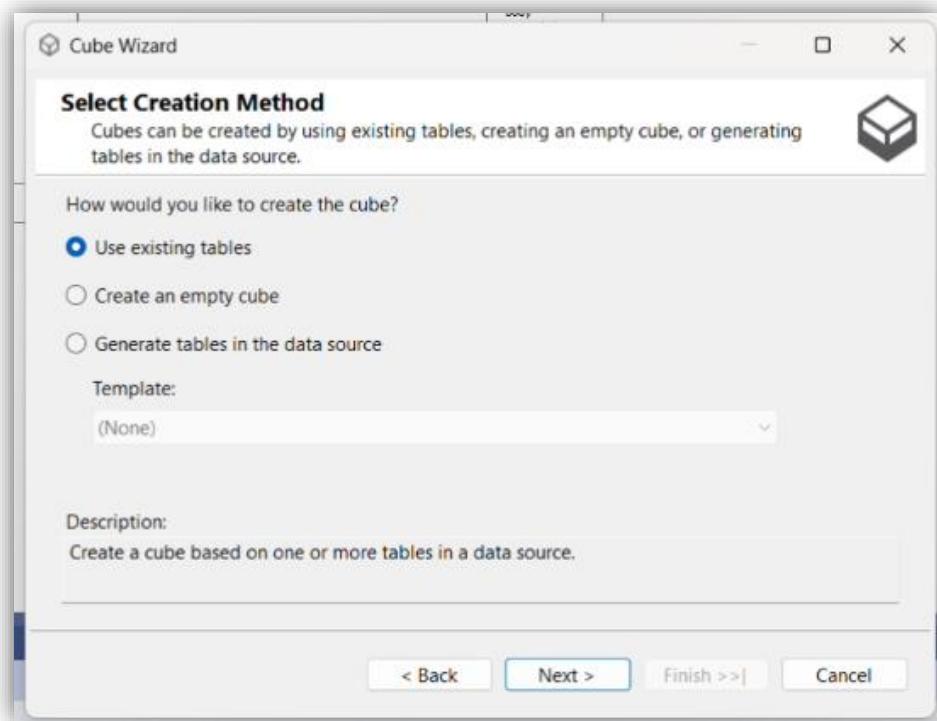
**Bước 1:** Tại Solution Explorer, ta click chuột phải vào thư mục Cubes và chọn New Cube.



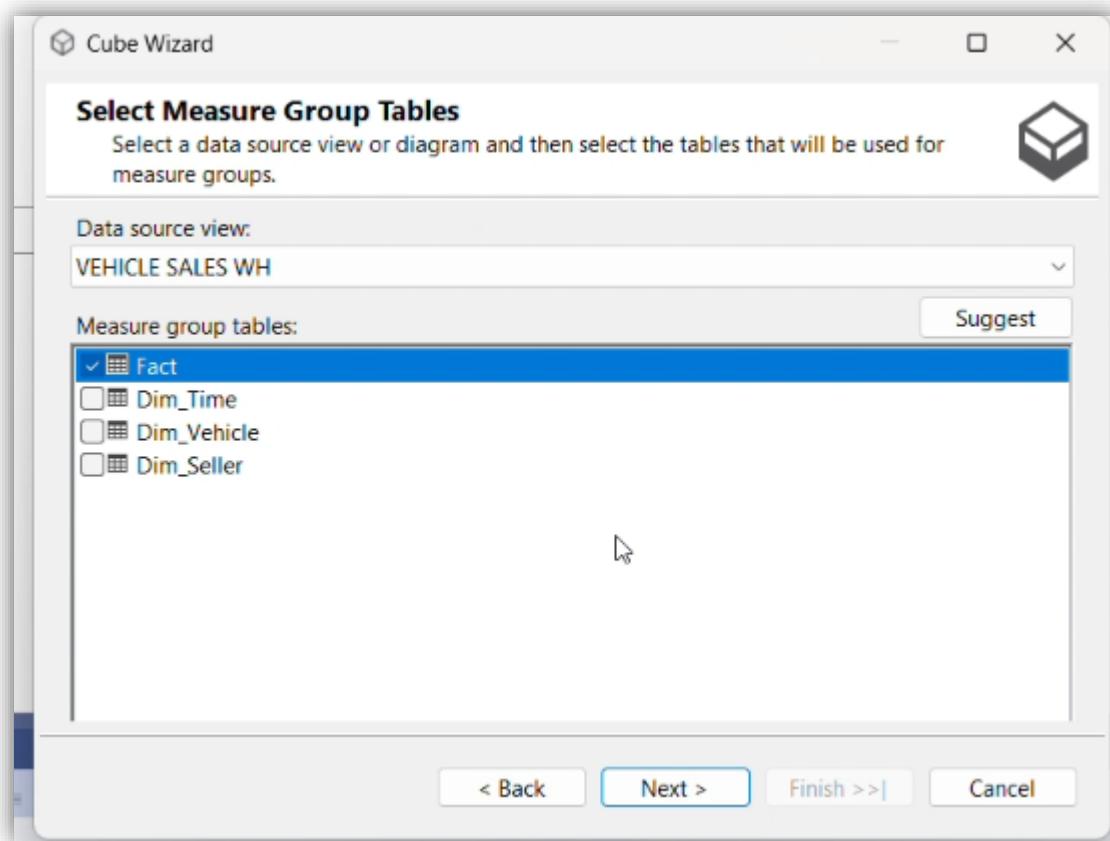
**Bước 2:** Hộp thoại Cube Wizard xuất hiện, chọn Next để tiếp tục.



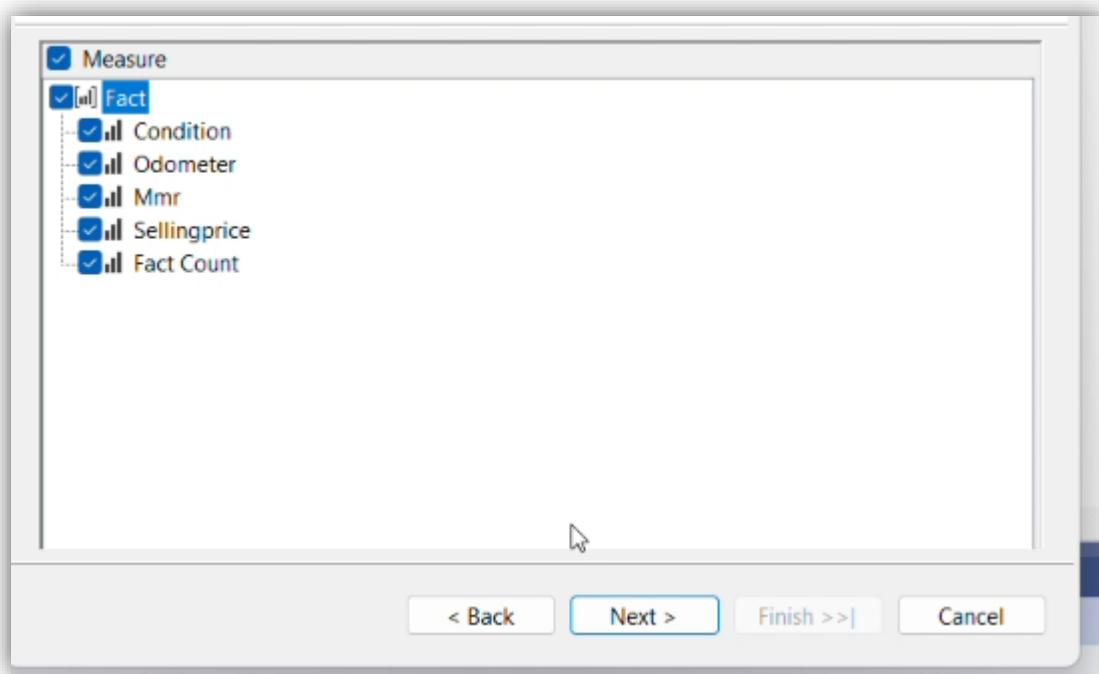
**Bước 3:** Chọn use existing tables, sau đó chọn Next để tiếp tục.



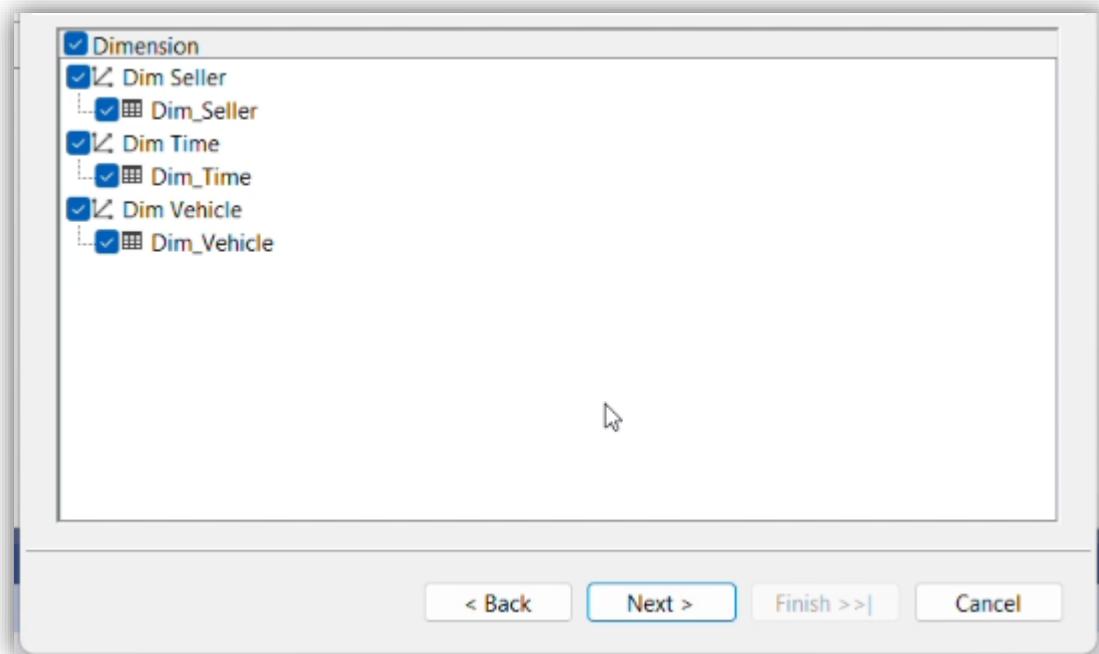
**Bước 4:** Chọn Fact để phân chia các measure group.



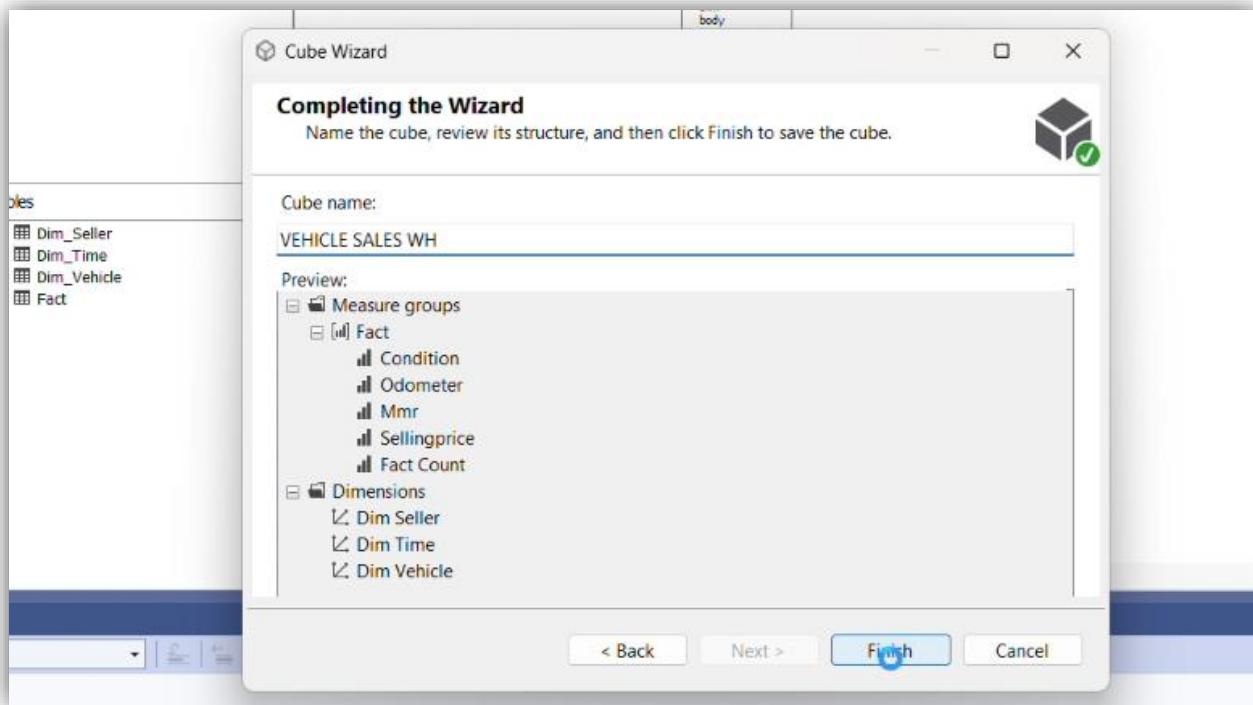
**Bước 5:** Chọn những độ đo để xuất, sau đó chọn Next để tiếp tục.



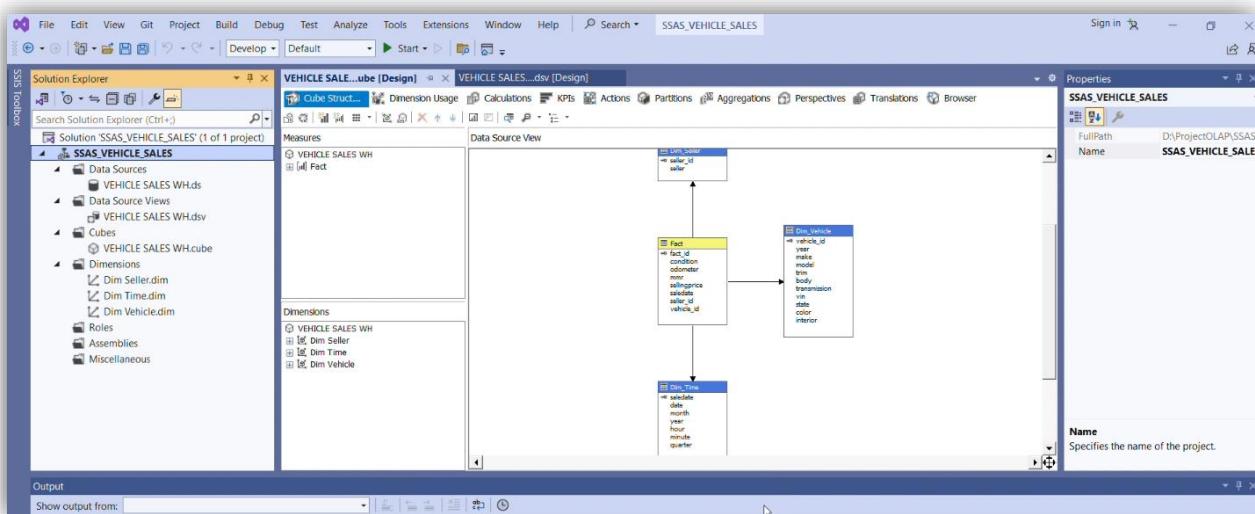
**Bước 6:** Chọn danh sách các bảng Dimension, sau đó chọn Next để tiếp tục.



**Bước 7:** Chọn Finish để hoàn tất quy trình xây dựng các khối (Cubes) và xác định các độ đo (Measures).

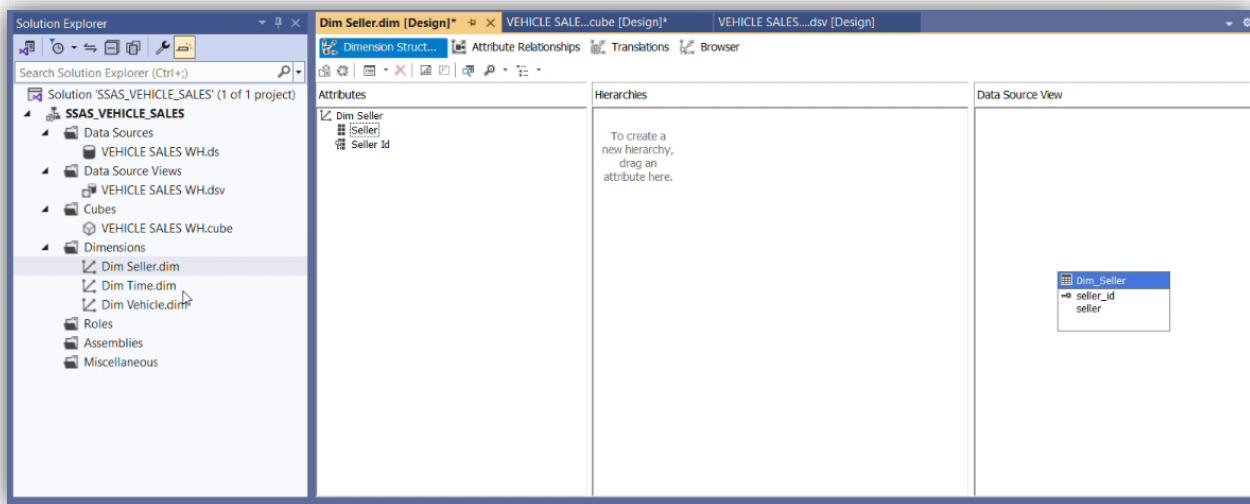


Sau khi kết thúc quá trình này, ta sẽ được kết quả như hình sau:

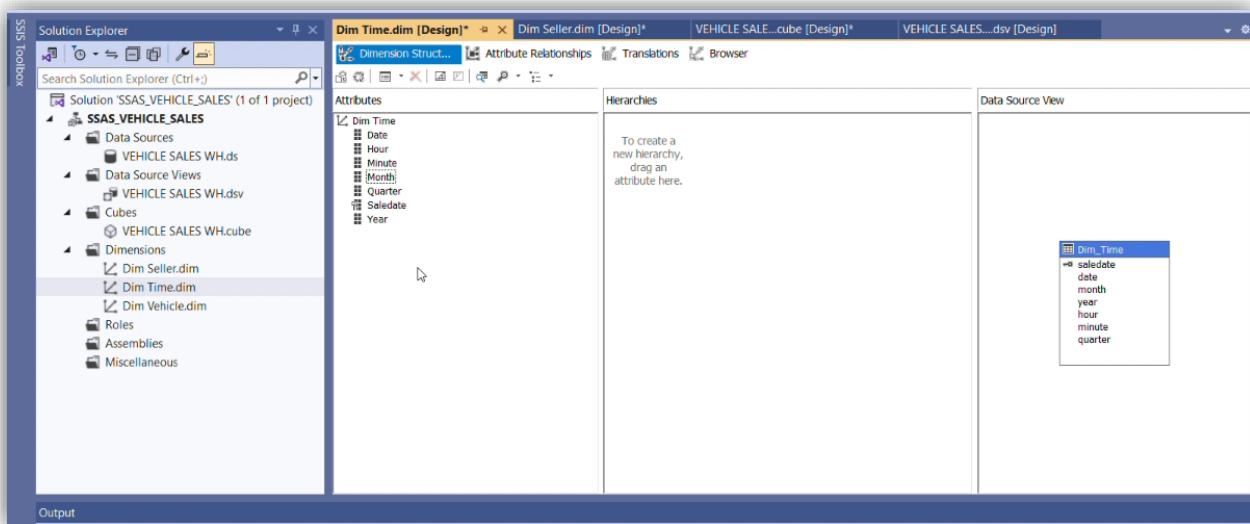


### 3.6. Xác định các chiều (Dimensions)

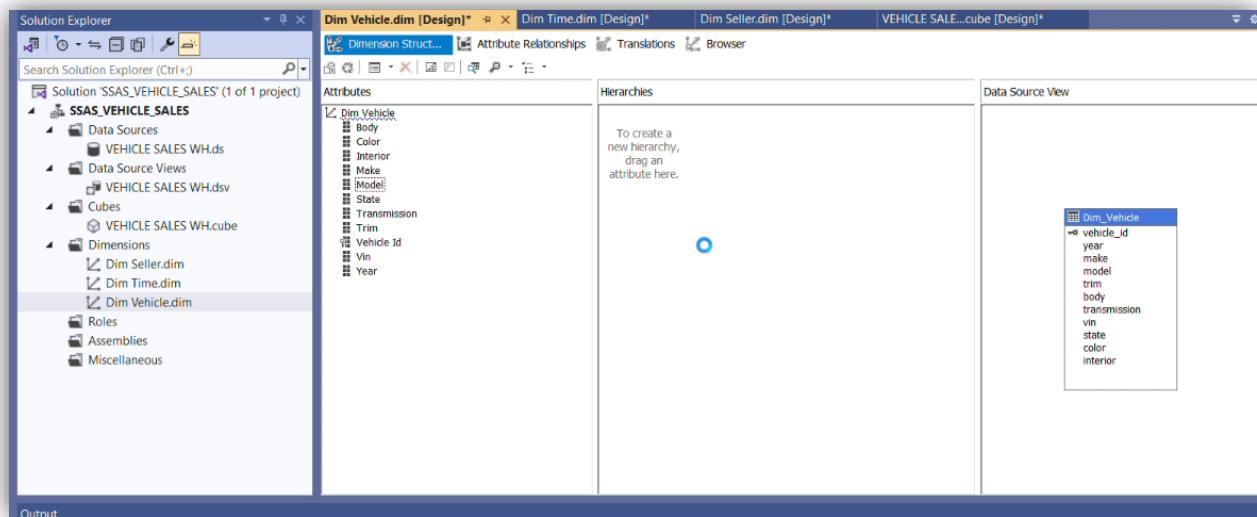
Bước 1: Tại folder Dimensions trong Solution Explorer, ta chọn Dim Seller.dim. Sau đó kéo thả các thuộc tính Seller từ Data Source View vào Attributes.



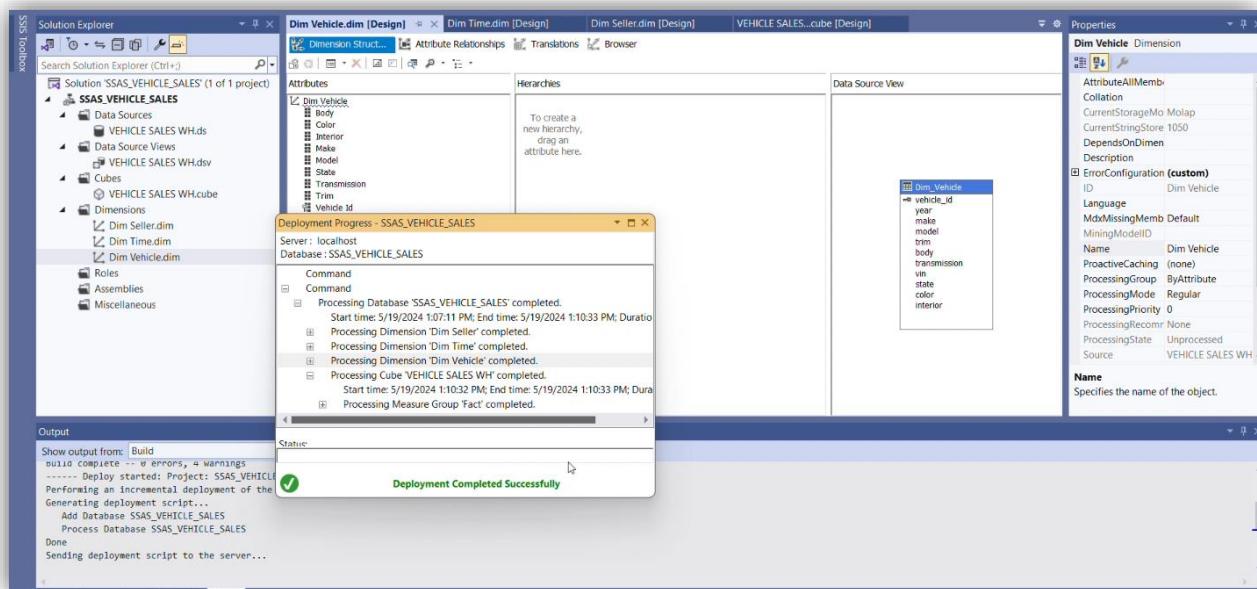
Bước 2: Tại folder Dimensions trong Solution Explorer, ta chọn Dim Time.dim. Sau đó kéo thả các thuộc tính từ Data Source View vào Attributes.



Bước 3. Tại folder Dimensions trong Solution Explorer, ta chọn Dim Vehicle.dim. Sau đó kéo thả các thuộc tính từ Data Source View vào Attributes.



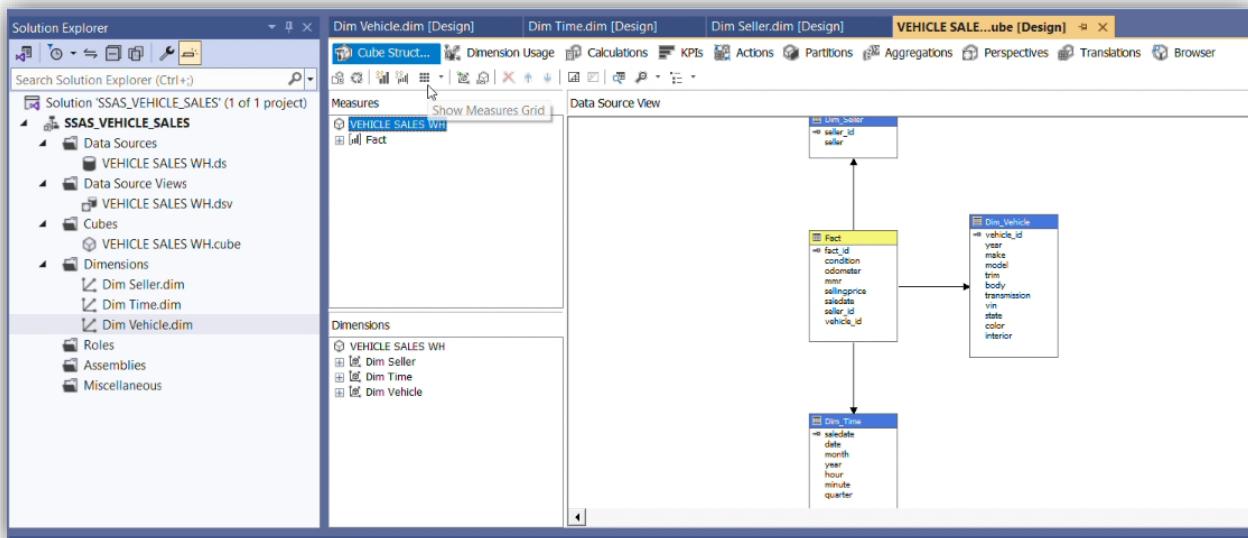
Bước 4: Ta chọn Start để deploy project. Khi deploy thành công ta sẽ nhận được kết quả như sau:



### 3.7. Xác định các độ đo (Measures)

#### 3.7.1. Đổi tên và thuộc tính các độ đo ban đầu

Bước 1: Tại khôi vừa tạo, chọn Show Measures Grid để hiện thị chi tiết các độ đo.



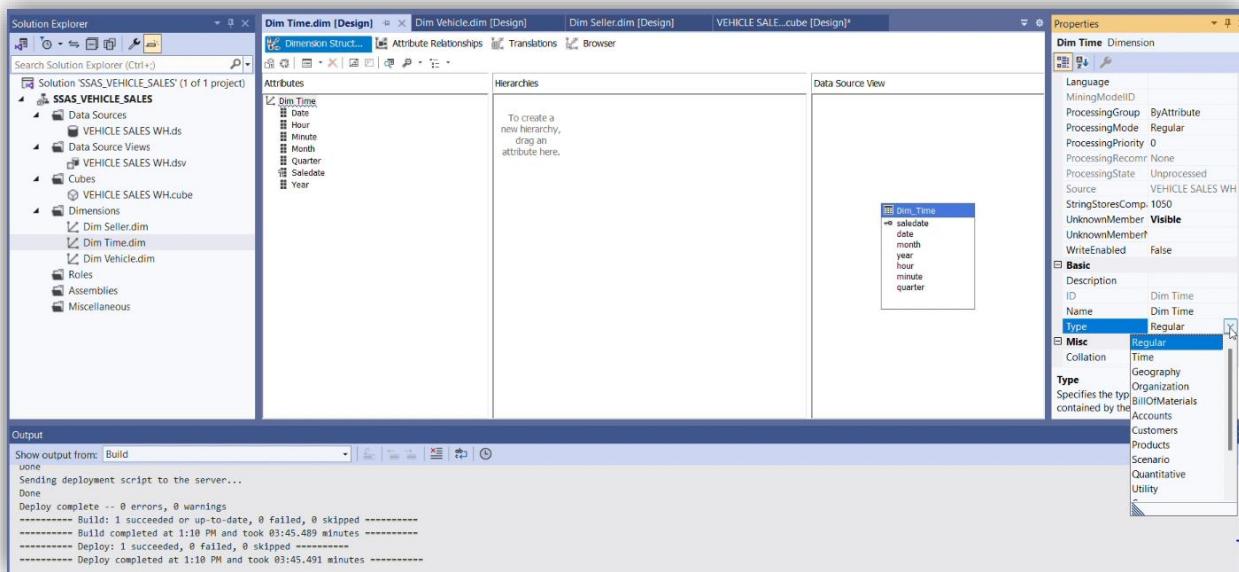
Chi tiết các độ đo sẽ hiển thị dưới dạng bảng, dễ dàng để tương tác.

Name	Measure Group	Data Type	Aggregation
Condition	Fact	Integer	Sum
Odometer		Integer	Sum
Mmr		Integer	Sum
Sellingprice		Integer	Sum
Fact Count		Integer	Count

**Bước 2:** Ta đổi tên và thuộc tính các độ đo hiện tại theo các hàm tổng hợp (aggregation) và thêm vào 1 độ đo nữa.

Sau khi đổi tên và thuộc tính các độ đo ban đầu. Một thông báo xuất hiện yêu cầu ta phải có một time dimension.

**Bước 3:** Mở Dim Time.dim. Tại cửa sổ Properties, ta đổi kiểu bảng từ Regular sang Time.



Quá trình hoàn tất, ta được 5 độ đo như hình

Measures				
	Name	Measure Group	Data Type	Aggregation
Condition			Integer	Max
Odometer			Integer	Max
Average Mmr			Double	AverageOfC...
Average Sellingprice			Double	AverageOfC...
Fact Vehicle Sales Co...			Integer	Count
Add new measure...				

### 3.7.2. Tạo các độ đo mới

Để tạo ra các độ đo mới, ta chọn Add new measure..., sau đó tạo ra thêm một độ đo mới là Total Sellingprice

Measures				
	Name	Measure Group	Data Type	Aggregation
Condition			Integer	Max
Odometer			Integer	Max
Average Mmr			Double	AverageOfC...
Average Sellingprice			Double	AverageOfC...
Fact Vehicle Sales Co...			Integer	Count
Total Sellingprice			Integer	Sum
Add new measure...				

### 3.7.3. Tổng kết độ đo

Stt	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	Fact Vehicle Sales Count	Int	Đếm số giao dịch bán xe.
2	Condition	Int	Điều kiện tối đa của phương tiện.
3	Odometer	Int	Số km tối đa mà phương tiện đã đi.
4	Average Mmr	Double	Trung bình giá trị thị trường ước tính của xe.
5	Average Sellingprice	Double	Trung bình giá bán thực của xe.
6	Total Sellingprice	Int	Tổng giá bán thực của xe

### 3.8. Phân cấp trong bảng Dim\_Time

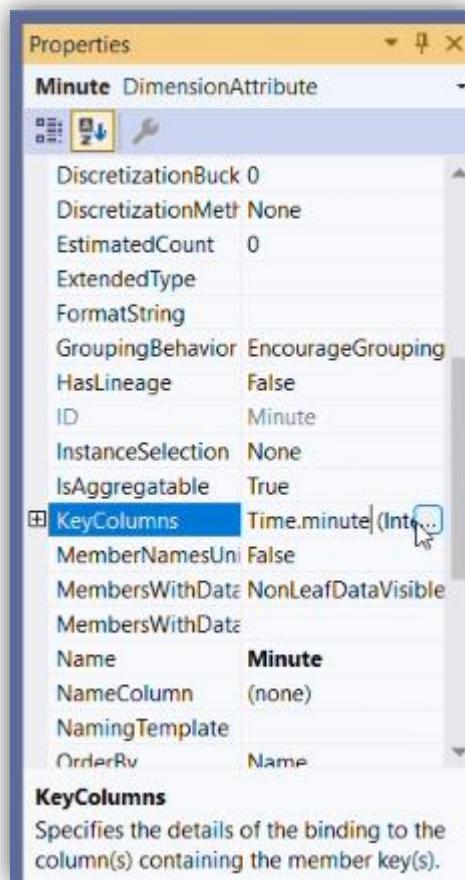
**Bước 1:** Kéo những thuộc tính cần phân cấp qua cửa sổ Hierarchies

**Bước 2:** Sắp xếp lại các thuộc tính phân cấp theo thứ tự: Year > Quarter > Month > Date > Hour > Minute

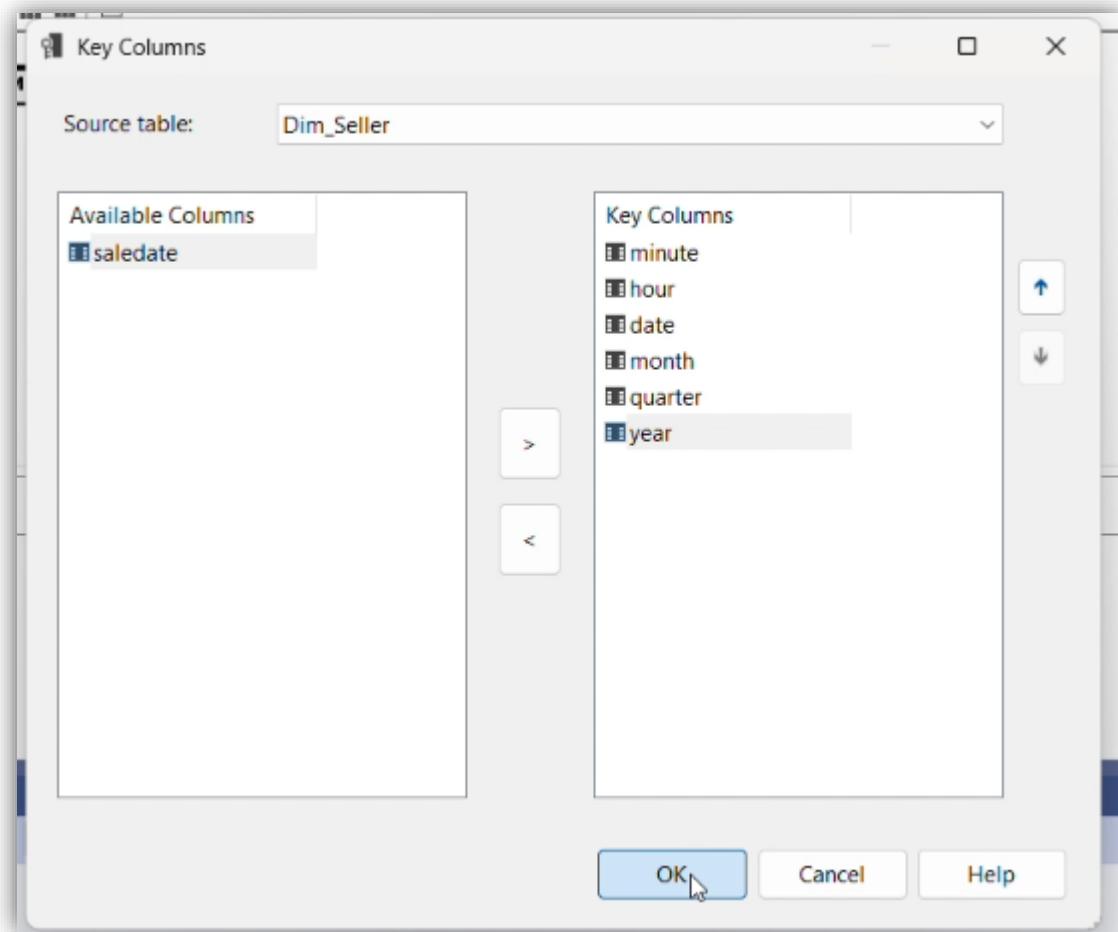
**Bước 3:** Tại panel Attribute Relationships, tạo mối quan hệ như sau



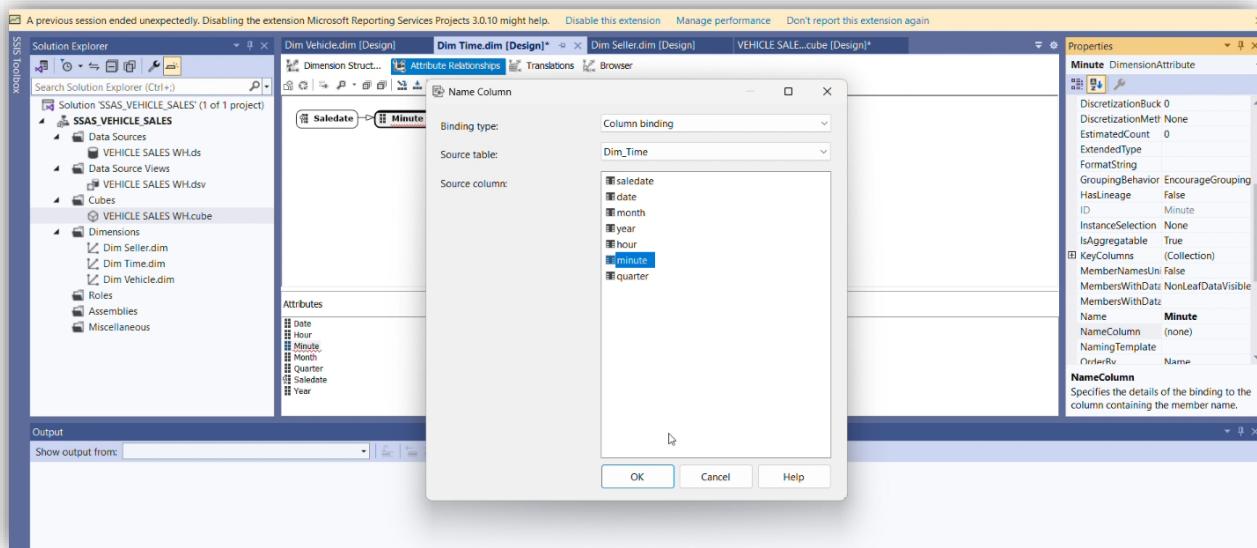
**Bước 4:** Chính khóa cột (KeyColumns) và tên cột (Name Column) của thuộc tính Minute. Vì thuộc tính Minute là thuộc tính cấp nhỏ nhất sẽ lấy khóa cột gồm chính nó và những thuộc tính cấp cao hơn. Tại cửa sổ Properties của thuộc tính Minute, chọn KeyColumns.



Thêm các thuộc tính cấp cao hơn vào KeyColumns, sau đó chọn OK để hoàn tất.

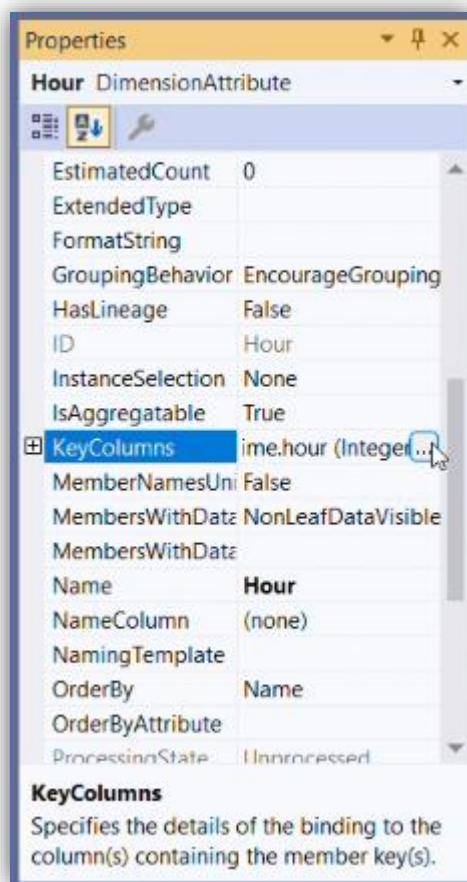


Tại cửa sổ Properties của thuộc tính Minute, ta chọn Name Column và chọn tên thuộc tính là Minute.

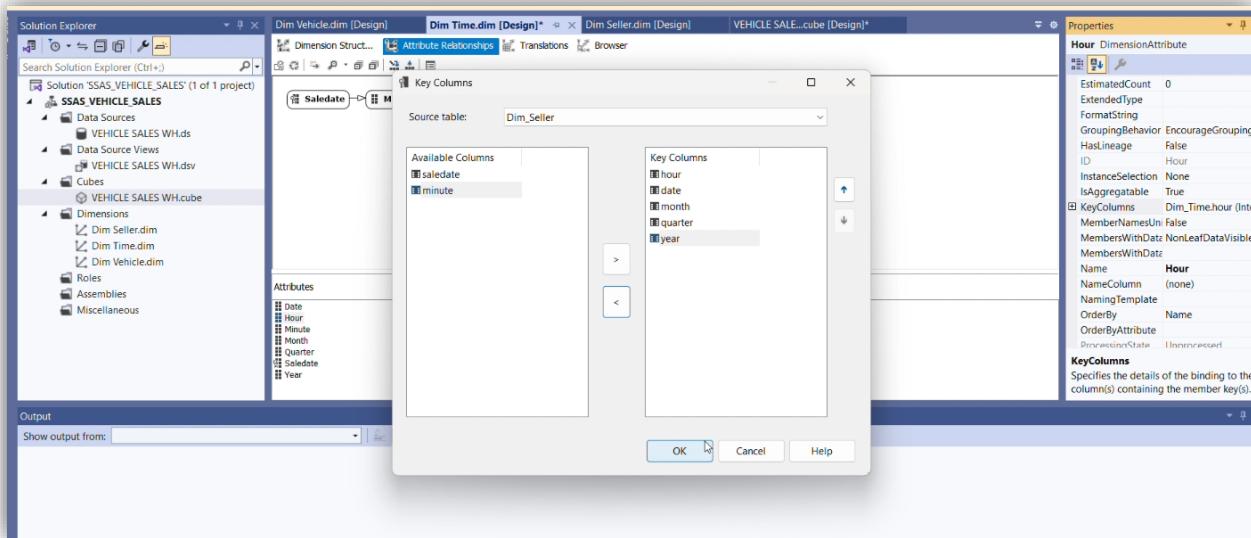


Bước 5: Chính khóa cột (KeyColumns) và tên cột (Name Column) của thuộc tính Hour.

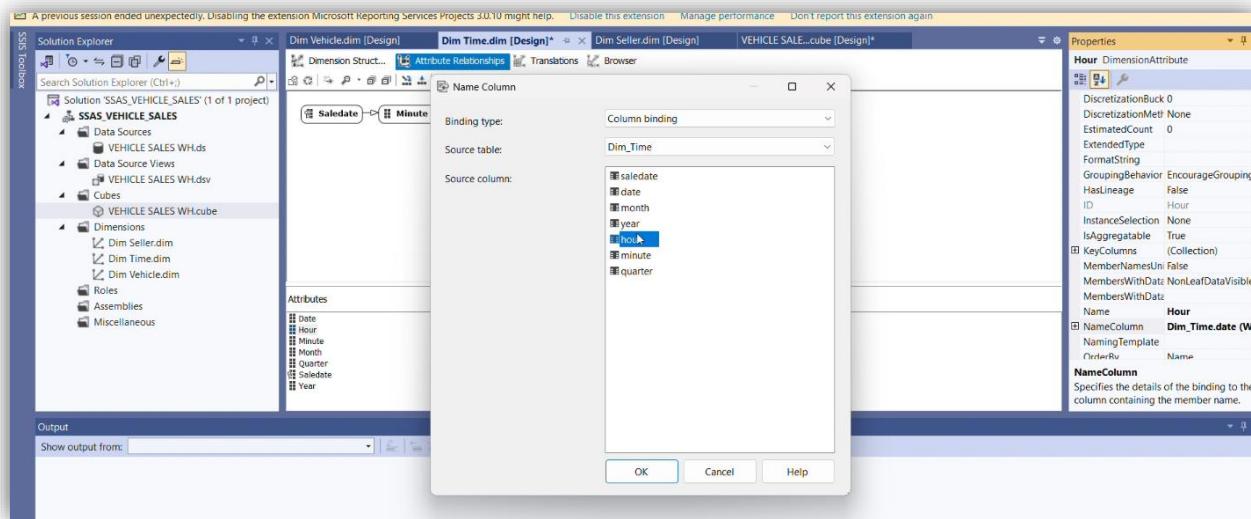
Vì thuộc tính Hour là thuộc tính cấp nhỏ hơn Date, Month, Quarter, Year nên sẽ lấy khóa cột gồm chính nó và những thuộc tính cấp cao hơn. Tại cửa sổ Properties của thuộc tính Hour, chọn KeyColumns.



Thêm các những thuộc tính cấp cao hơn vào KeyColumns, sau đó chọn OK để hoàn tất.



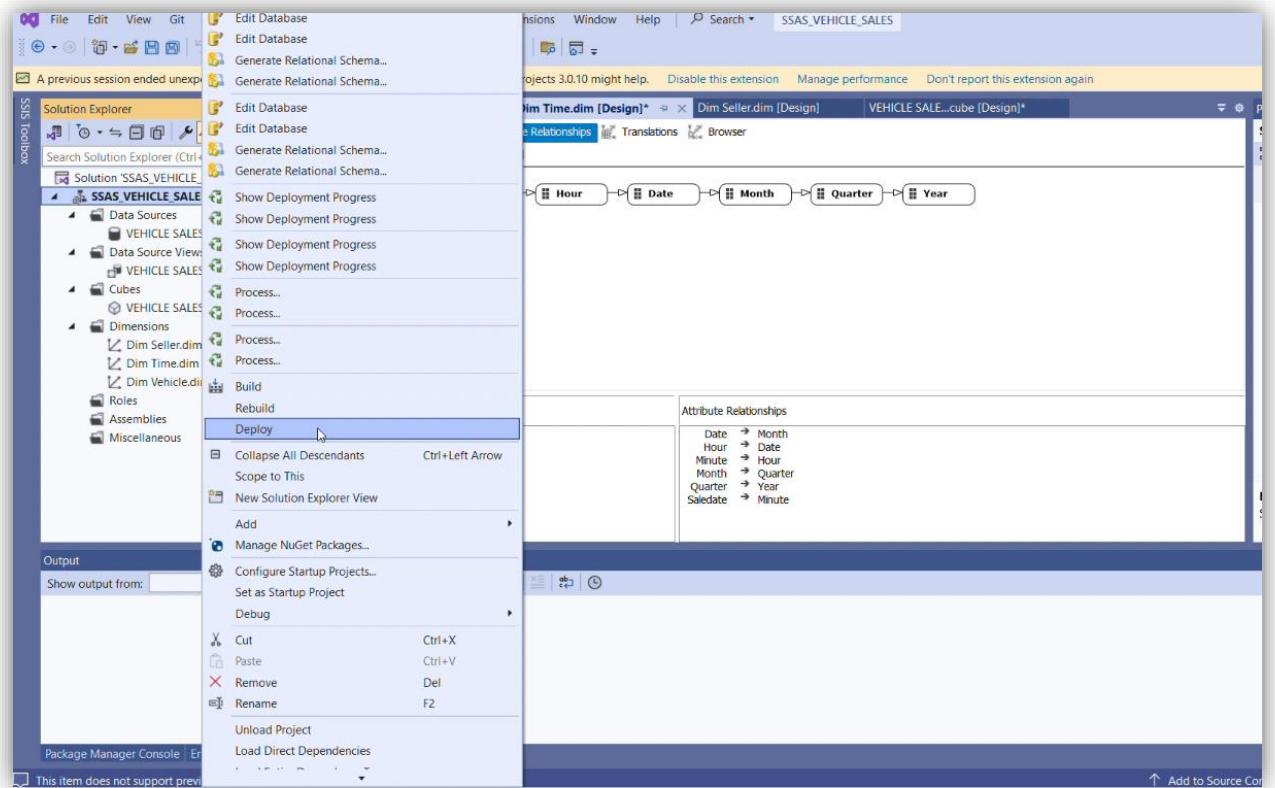
Tại cửa sổ Properties của thuộc tính Hour, ta chọn Name Column và chọn tên thuộc tính là Hour.



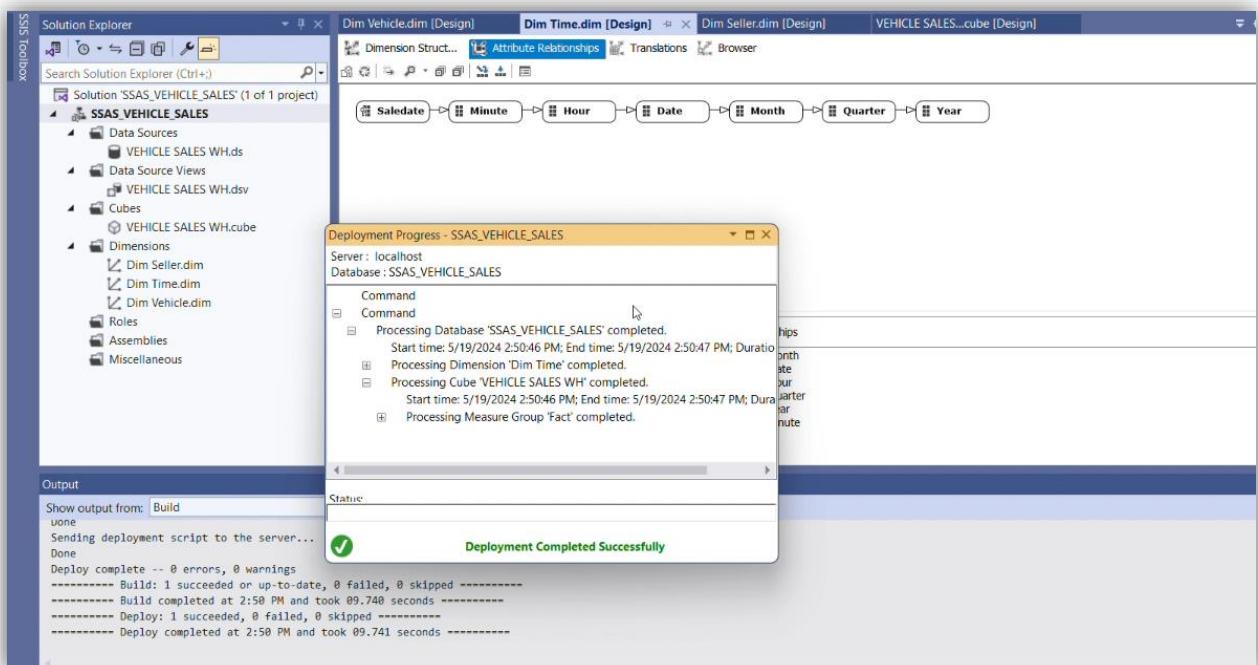
Tương tự cho các thuộc tính Date, Month, Quarter, Year

### 3.9. Chạy dự án SSAS

Sau khi quá trình phân cấp cho các bảng chiều hoàn tất, ta thực hiện deploy project để đảm bảo không có lỗi xảy ra sau quá trình phân cấp. Nhấn chuột phải vào tên project () nhấn Deploy như hình sau:



Khi deploy thành công, hệ thống sẽ hiển thị như hình sau và chúng ta bắt đầu thực hiện các câu truy vấn.



### 3.10. Thực hiện 15 câu truy vấn

### 3.10.1. Số lượng giao dịch bán xe cũ theo từng năm

- Trong MSSQ

MDXQuery2.mdx -...TOP-CEPEPGEL\Hp)\* timxetrungmavin.s...-CEPEPGEL\Hp (57)\*

Cube: CARSALES

Metadata Functions

Search Model

Measure Group: <All>

CARSales

Measures

Fact

- Average Mmr
- Average Sellingprice
- Condition
- Fact Vehicle Sales Count
- Odometer
- Total Sellingprice

KPIs

Dim Seller

Dim Time

Date

Hour

Minute

Month

Quarter

//1 Số lượng giao dịch bán xe cũ theo từng năm

```
SELECT [Measures].[Fact Vehicle Sales Count] ON COLUMNS,
       [Dim Time].[Year].[Year].Members ON ROWS
  FROM [CARSALES];
```

//2 Tổng doanh thu từ việc bán xe theo từng người bán

```
SELECT [Measures].[Total SellingPrice] ON COLUMNS,
       [Dim Seller].[Seller].[Seller].Members ON ROWS
  FROM [CARSALES];
```

//3 Report Giá bán trung bình theo từng hàng sắp xếp theo giá trị giảm dần

```
SELECT
```

100 %

Messages Results

	Fact Vehicle Sales Count
2014	34474
2015	65526
Unknown	(null)

- Trong Visual Studio Code

Dim Vehicle.dim [Design] Dim Time.dim [Design] Dim Seller.dim [Design] **CARSALES.cube [Design]** VEHICLE SALES....dsv [Design]

Cube Struct... Dimension Usage Calculations KPIs Actions Partitions Aggregations Perspectives Translations Browser

Language: Default

Edit as Text Import... MDX

Dimension Hierarchy Operator Filter Expression

<Select dimension>

Search Model

Measure Group: <All>

Measures

- Fact
  - Average Mmr
  - Average Sellingprice
  - Condition
  - Fact Vehicle Sales Count
  - Odometer
  - Total Sellingprice

KPIs

Dim Seller

- Seller
- Seller Id

Dim Time

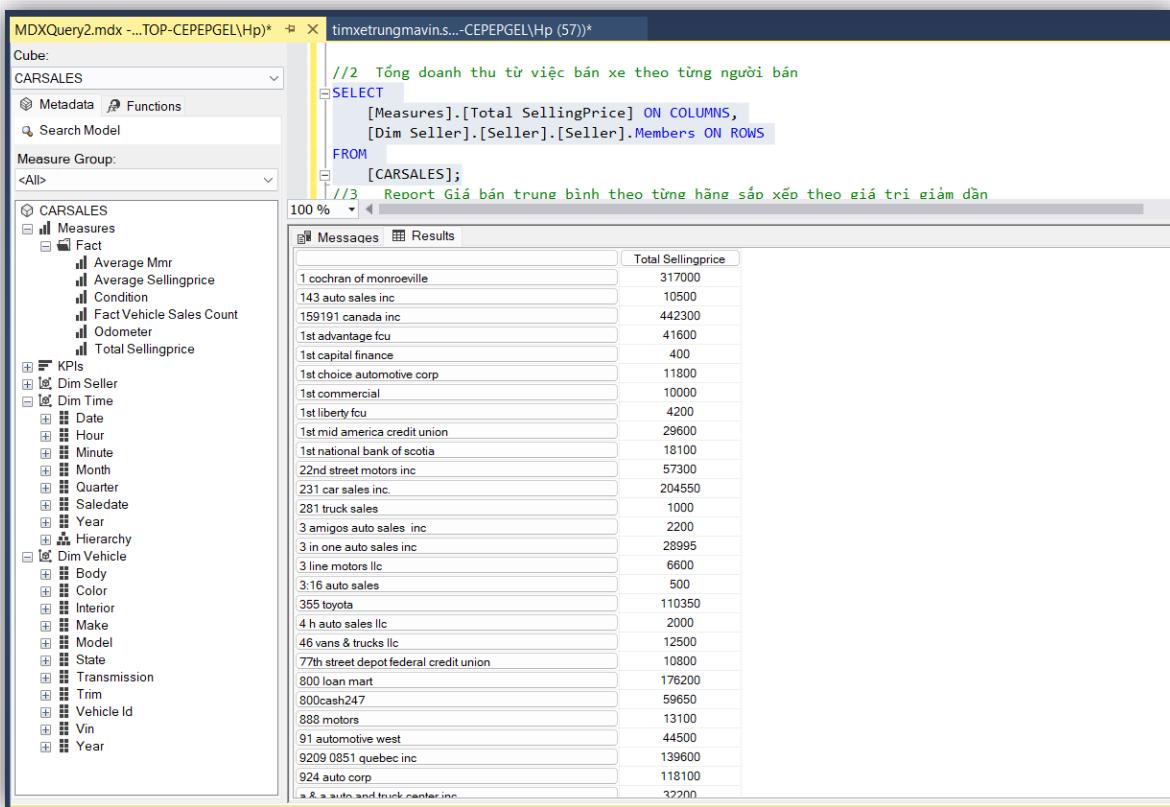
- Date

Calculated Members

Year	Fact Vehicle Sales Count
2014	34474
2015	65526

### 3.10.2. Tổng doanh thu từ việc bán xe theo từng người bán

- Trong MSSQ



```

MDXQuery2.mdx - ...TOP-CEPEPGEL\Hp*  x timxetrungmavins...-CEPEPGEL\Hp (57)*
Cube: CARSALES
Measure Group: <All>
  CARSALES
    Measures
      Fact
        Average Mmr
        Average Sellingprice
        Condition
        FactVehicle Sales Count
        Odometer
        Total Sellingprice
      KPIs
    Dim Seller
    Dim Time
      Date
      Hour
      Minute
      Month
      Quarter
      Saledate
      Year
      Hierarchy
    Dim Vehicle
      Body
      Color
      Interior
      Make
      Model
      State
      Transmission
      Trim
      Vehicle Id
      Vin
      Year
  [Measures].[Total SellingPrice] ON COLUMNS,
  [Dim Seller].[Seller].[Seller].Members ON ROWS
FROM [CARSALES];
//3 Report Giá bán trung bình theo từng hàn g sáo xép theo giá trị giảm dần
  
```

	Total Sellingprice
1 cochran of monroeville	317000
143 auto sales inc	10500
159191 canada inc	442300
1st advantage fcu	41600
1st capital finance	400
1st choice automotive corp	11800
1st commercial	10000
1st liberty fcu	4200
1st mid america credit union	29600
1st national bank of scota	18100
22nd street motors inc	57300
231 car sales inc.	204550
281 truck sales	1000
3 amigos auto sales inc	2200
3 in one auto sales inc	28995
3 line motors llc	6600
3:16 auto sales	500
355 toyota	110350
4 h auto sales llc	2000
46 vans & trucks llc	12500
77th street depot federal credit union	10800
800 loan mart	176200
800cash247	59650
888 motors	13100
91 automotive west	44500
9209 0851 quebec inc	139600
924 auto corp	118100
82 auto and truck center inc	32200

- Trong Visual Studio Code

The screenshot shows the SSAS Management Studio interface. The left pane displays the cube structure for 'CARSALES' with nodes for 'Metadata', 'Search Model', 'Measure Group', and 'Calculated Members'. The right pane shows a data grid titled 'Seller' with a column for 'Seller' and 'Total Sellingprice'. The data is sorted by 'Total Sellingprice' in descending order, with the top entry being '1 cochrane of monroeville' at 317000. The data grid includes a header row and 20 data rows.

Seller	Total Sellingprice
1 cochrane of monroeville	317000
143 auto sales inc	10500
159191 canada inc	442300
1st advantage fcu	41600
1st capital finance	400
1st choice automotive corp	11800
1st commercial	10000
1st liberty fcu	4200
1st mid america credit union	29600
1st national bank of scotia	18100
22nd street motors inc	57300
231 car sales inc.	204550
281 truck sales	1000
3 amigos auto sales inc	2200
3 in one auto sales inc	28995
3 line motors llc	6600
3:16 auto sales	500
355 toyota	110350
4 h auto sales llc	2000

### 3.10.3. Giá bán trung bình theo từng hàng sắp xếp theo giá trị giảm dần

- Trong MSSQ

```

SELECT
    [Measures].[Average SellingPrice] ON COLUMNS,
    ORDER([Dim Vehicle].[Make].[Make].Members, [Measures].[Average SellingPrice], DESC) ON ROWS
FROM
    [CARSALES];
//4 Trung bình doanh thu theo từng tháng trong năm 2015
  
```

100 %

Messages Results

	Average Sellingprice
Ford	244942.690448792
Infiniti	199531.07278481
Rolls-Royce	159650
BMW	156392.014084507
Nissan	151326.964630225
Chevrolet	150886.618531889
Bentley	145933.333333333
Mercedes-Benz	138937.938967136
Ferrari	137750
Toyota	134521.888888889
Honda	125695.027932961
Lexus	123813.389830508
Dodge	100360.902439024
Kia	99798.7402298851
Hyundai	94680.020242915
Ram	85847.3913043478
Land Rover	83799.4615384615
Jeep	81332.5492341357
Tesla	80000
Porsche	76022.8448275862
Audi	70911.8103448276
Subaru	67121.568627451
GMC	64630.1259259259
Chrysler	60306.3595706619
Acura	59102.6051672962

- Trong Visual Studio Code

Solution Explorer      Dim Vehicle.dim [Design]      Dim Time.dim [Design]      Dim Seller.dim [Design]      CARSALES.cube [Design]      VEHICLE SALES...dsv [Design]

Properties

**CARSALES Cube**

- AggregationPrefi
- Collation
- DefaultMeasure
- Description
- ErrorConfigurati
- EstimatedRows 0
- ID VEHICLE SALES WH
- Language
- Name **CARSALES**
- ProactiveCaching (none)
- ProcessingMode Regular
- ProcessingPriority 0
- ScriptCacheProce Regular
- ScriptErrorHandli IgnoreNone
- Source VEHICLE SALES WH (D
- StorageLocation
- StorageMode Molap
- Visible True

**CARSALES.cube [Design]**

Dimension: Dim Vehicle      Hierarchy: Make      Operator: Equal      Filter Expression: <Select dimension>

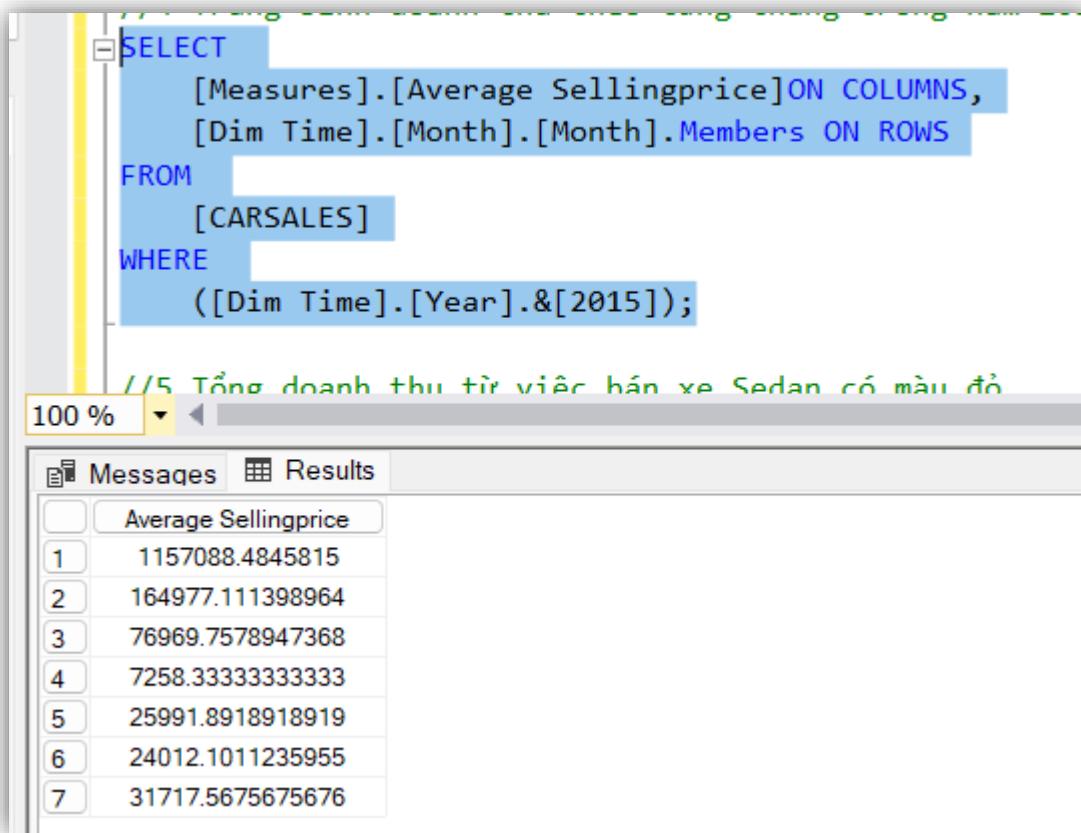
Measure Group: <All>

Calculated Members

Make	Average Sellingprice
Acura	59102.6051672962
Aston Martin	51000
Audi	70911.8103448276
Bentley	145933.333333333
BMW	156392.014084507
Buck	29478.3178807947
Cadillac	52717.322834657
Chevrolet	150886.618531889
Chrysler	60306.3595706619
Dodge	100360.902439024
Ferrari	137750
FIAT	19118.3139534884
Fisker	54500
Ford	244942.690448792
Geo	681.25
GMC	64630.1259259259
Honda	125695.027932961
HUMMER	18330.0925925926
Hyundai	94680.020242915

### 3.10.4. Trung bình doanh thu theo từng tháng trong năm 2015

- Trong MSSQ



```

SELECT
    [Measures].[Average Sellingprice] ON COLUMNS,
    [Dim Time].[Month].[Month].Members ON ROWS
FROM
    [CARSALES]
WHERE
    ([Dim Time].[Year].&[2015]);

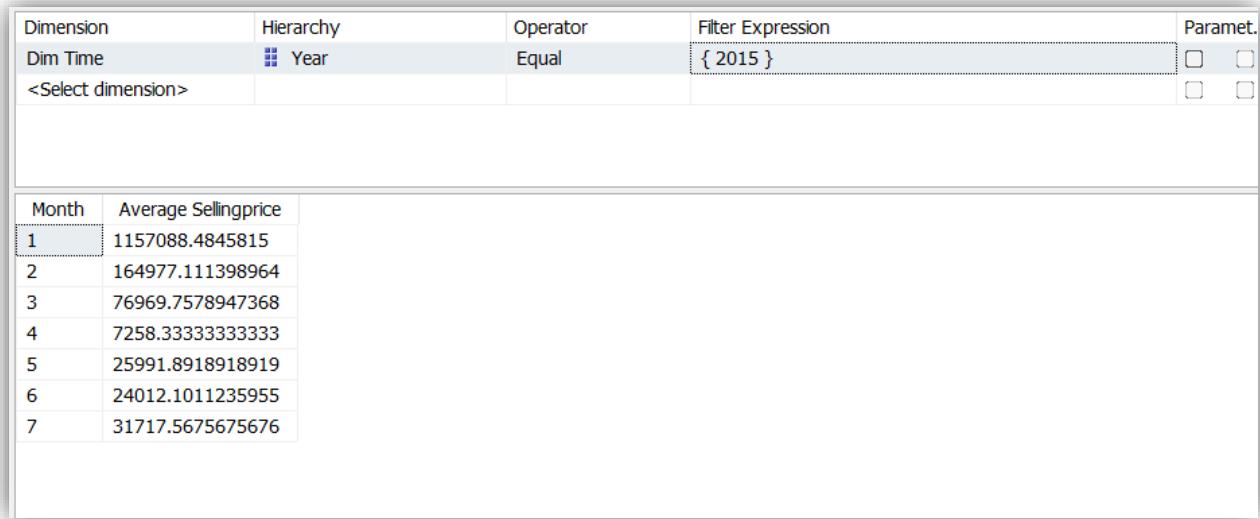
```

//5 Tổng doanh thu từ việc bán xe Sedan có màu đỏ

100 %

	Average Sellingprice
1	1157088.4845815
2	164977.111398964
3	76969.7578947368
4	7258.33333333333
5	25991.8918918919
6	24012.1011235955
7	31717.5675675676

- Trong Visual Studio Code



Dimension	Hierarchy	Operator	Filter Expression	Paramet.
Dim Time	Year	Equal	{ 2015 }	<input type="checkbox"/> <input type="checkbox"/>
<Select dimension>				

Month	Average Sellingprice
1	1157088.4845815
2	164977.111398964
3	76969.7578947368
4	7258.33333333333
5	25991.8918918919
6	24012.1011235955
7	31717.5675675676

### 3.10.5. Tổng doanh thu từ việc bán xe Sedan có màu đỏ

- Trong MSSQ

```

//5 Tổng doanh thu từ việc bán xe Sedan có màu đỏ
SELECT
    {[Measures].[Total SellingPrice]} ON COLUMNS
FROM
    [CARSALES]
WHERE
    ([Dim Vehicle].[Body].[Sedan], [Dim Vehicle].[Color].[Red]);

//6 Trung bình giá thị trường ước tính theo mỗi hãng xe
SELECT

```

100 %

Messages Results

Total Sellingprice	35431754
--------------------	----------

- Trong Visual Studio Code

The screenshot shows the Microsoft SQL Server Management Studio (MSSQ) interface. On the left, there is a tree view of the database structure for the 'CARSALES' database, including 'Metadata', 'Search Model', 'Measure Group', and 'Fact' tables. The 'Fact' table contains measures like 'Average Mmr' and 'Average Sellingprice'. On the right, there is a query editor window with the following content:

```

SELECT
    {[Measures].[Total SellingPrice]} ON COLUMNS
FROM
    [CARSALES]
WHERE
    ([Dim Vehicle].[Body].[Sedan], [Dim Vehicle].[Color].[Red]);

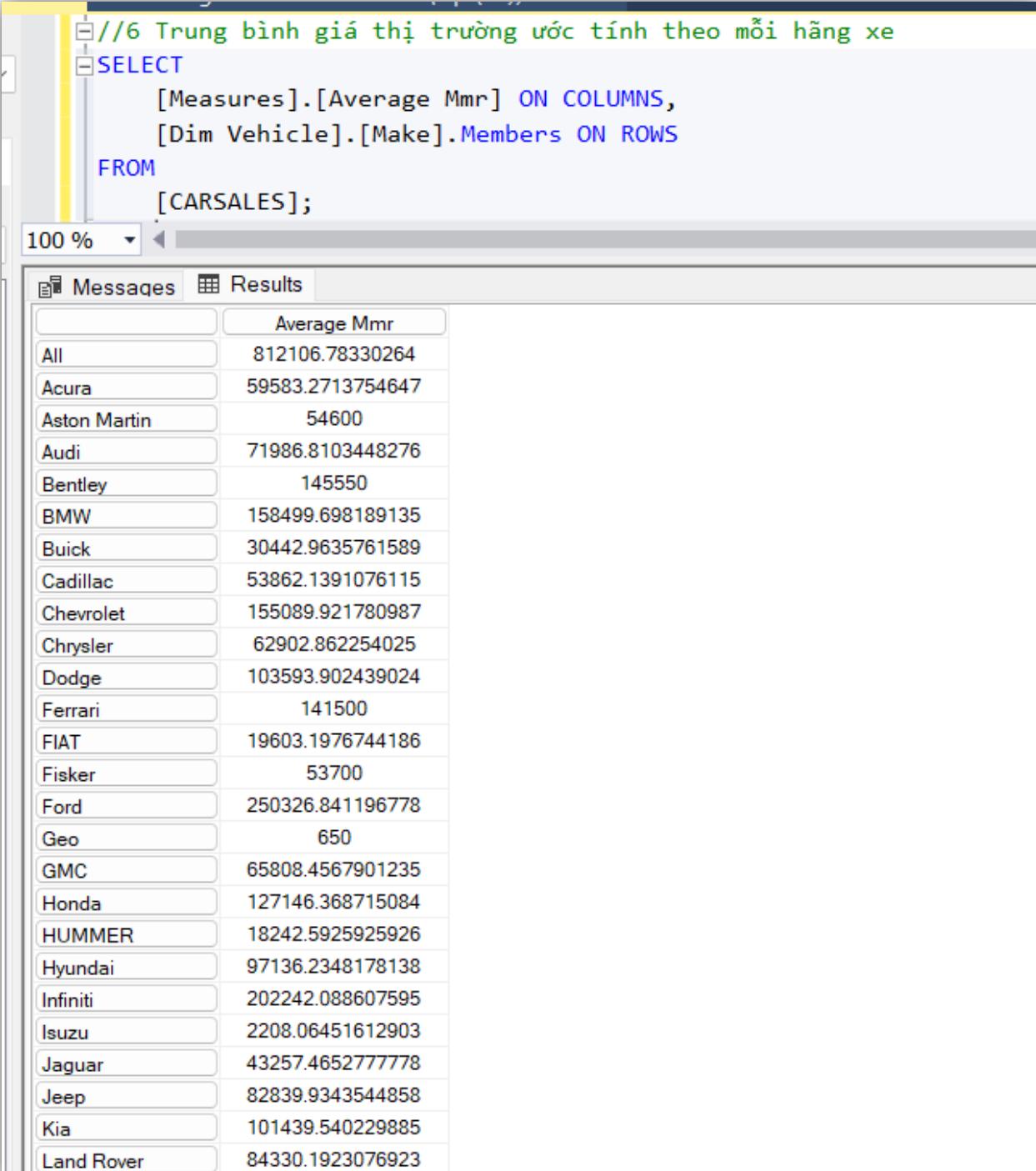
```

Below the query editor is a results grid with one row:

Total Sellingprice	35431754
--------------------	----------

### 3.10.6. Trung bình giá thị trường ước tính theo mỗi hãng xe

- Trong MSSQ



The screenshot shows a Microsoft Analysis Services (AS) query editor. The top pane displays an AS query:

```
//6 Trung bình giá thị trường ước tính theo mỗi hãng xe
SELECT
    [Measures].[Average Mmr] ON COLUMNS,
    [Dim Vehicle].[Make].Members ON ROWS
FROM
    [CARSALES];
```

The bottom pane shows the results of the query, which is a table of car makes and their average market price (Average Mmr). The table has two columns: 'Make' and 'Average Mmr'.

Make	Average Mmr
All	812106.78330264
Acura	59583.2713754647
Aston Martin	54600
Audi	71986.8103448276
Bentley	145550
BMW	158499.698189135
Buick	30442.9635761589
Cadillac	53862.1391076115
Chevrolet	155089.921780987
Chrysler	62902.862254025
Dodge	103593.902439024
Ferrari	141500
FIAT	19603.1976744186
Fisker	53700
Ford	250326.841196778
Geo	650
GMC	65808.4567901235
Honda	127146.368715084
HUMMER	18242.5925925926
Hyundai	97136.2348178138
Infiniti	202242.088607595
Isuzu	2208.06451612903
Jaguar	43257.4652777778
Jeep	82839.9343544858
Kia	101439.540229885
Land Rover	84330.1923076923

- Trong Visual Studio Code

The screenshot shows the SSAS Data Model Editor interface. On the left, the 'CARSALES' metadata is displayed, including a 'Dim Vehicle' dimension with various attributes like Body, Color, Interior, Make, Model, State, Transmission, and Trim. The 'Calculated Members' section is empty. On the right, a table lists car makes with their average MMR (Average Mmr). The table has two columns: 'Make' and 'Average Mmr'. The data is as follows:

Make	Average Mmr
Acura	59583.2713754647
Aston Martin	54600
Audi	71986.8103448276
Bentley	145550
BMW	158499.698189135
Buick	30442.9635761589
Cadillac	53862.1391076115
Chevrolet	155089.921780987
Chrysler	62902.862254025
Dodge	103593.902439024
Ferrari	141500
FIAT	19603.1976744186
Fisker	53700
Ford	250326.841196778
Geo	650
GMC	65808.4567901235
Honda	127146.368715084
HUMMER	18242.5925925926
Hyundai	97136.2348178138

### 3.10.7. Giá trị MMR trung bình và Odometer cao nhất theo từng dòng xe

- Trong MSSQ

```
//7 Giá trị MMR trung bình và Odometer cao nhất theo từng dòng xe
SELECT
    { [Measures].[Average Mmr], [Measures].[Odometer] } ON COLUMNS,
    {[Dim Vehicle].[Make].[Make].Members * [Dim Vehicle].[Model].[Model].Members} ON ROWS
FROM
    [CARSALES];
//8 Số lượng giao dịch và tổng doanh thu theo từng quý
SELECT
```

100 %

		Average Mmr	Odometer
Acura	CL	1766.666666666667	375363
Acura	ILX	21481.5789473684	41578
Acura	Integra	1229.166666666667	200636
Acura	Legend	900	153841
Acura	MDX	31947.0890410959	312886
Acura	RDX	24939.8305084746	159149
Acura	RL	5602.777777777778	314736
Acura	RLX	33300	18320
Acura	RSX	4502.083333333333	207575
Acura	TL	31638.4375	291489
Acura	TSX	43444.7674418605	226770
Acura	TSX Sport Wagon	20564.2857142857	64736
Acura	ZDX	33016.666666666667	45346
Aston Martin	V8 Vantage	54600	23479
Audi	A3	17493.75	226816
Audi	A4	28710.5113636364	277574
Audi	A5	43684.0909090909	168454
Audi	A6	32439.0957446808	195372
Audi	A7	62578.9473684211	55216
Audi	A8	29888.8888888889	206785
Audi	allroad	37242.8571428571	30270
Audi	allroad quattro	3429.166666666667	142940
Audi	Q5	39256.8965517241	165704

- Trong Visual Studio Code

Make	Model	Average Mmr	Odometer
Acura	CL	1766.666666666667	375363
Acura	ILX	21481.5789473684	41578
Acura	Integra	1229.166666666667	200636
Acura	Legend	900	153841
Acura	MDX	31947.0890410959	312886
Acura	RDX	24939.8305084746	159149
Acura	RL	5602.777777777778	314736
Acura	RLX	33300	18320
Acura	RSX	4502.083333333333	207575
Acura	TL	31638.4375	291489
Acura	TSX	43444.7674418605	226770
Acura	TSX Sport Wagon	20564.2857142857	64736
Acura	ZDX	33016.66666666667	45346
Aston Martin	V8 Vantage	54600	23479
Audi	A3	17493.75	226816
Audi	A4	28710.5113636364	277574
Audi	A5	43684.0909090909	168454
Audi	A6	32439.0957446808	195372
Audi	A7	62578.9473684211	55216

### 3.10.8. Số lượng giao dịch và tổng doanh thu theo từng quý

- Trong MSSQ

```

//8 Số lượng giao dịch và tổng doanh thu theo từng quý
SELECT
    { [Measures].[Fact Vehicle Sales Count], [Measures].[Total SellingPrice] } ON COLUMNS,
    {[Dim Time].[Year].Members * [Dim Time].[Quarter].Members} ON ROWS
FROM
    [CARSALES];

//9 Tìm các hãng xe có doanh thu dưới 100,000

```

Year	Quarter	Fact Vehicle Sales Count	Total Sellingprice
2014	1	161	2497425
2014	4	34313	430425783
2015	1	65219	858970550
2015	2	230	3185877
2015	3	77	1173550
Unknown	Unknown	(null)	(null)

- Trong Visual Studio Code

Year	Quarter	Fact Vehicle Sales Count	Total Sellingprice
2014	1	161	2497425
2014	4	34313	430425783
2015	1	65219	858970550
2015	2	230	3185877
2015	3	77	1173550

### 3.10.9. Tìm các hãng xe có doanh thu dưới 100,000

- Trong MSSQ

Make	Total Sellingprice
Aston Martin	51000
Fisker	54500
Geo	2725
Isuzu	61900
Oldsmobile	71000
Plymouth	1100
Tesla	80000
Unknown	(null)

- Trong Visual Studio Code

Bước 1: Tạo 1 Named Set [Make With Total Sellingprice Under 100000]

Expression:

```

EXCEPT([Dim Vehicle].[Make].Members,
        FILTER([Dim Vehicle].[Make].Members, [Measures].[Total Sellingprice] >
        100000))
  
```

Bước 2: Nhập chọn Process, sau đó ở tab Browser chọn Refresh

Bước 3: Chọn Filter Expression và kéo thả các trường, ta được kết quả dưới đây:

Make	Total Sellingprice
Aston Martin	51000
Fisker	54500
Geo	2725
Isuzu	61900
Oldsmobile	71000
Plymouth	1100
Tesla	80000

### 3.10.10. Số lượng giao dịch bán xe của các người bán hàng top 5

- Trong MSSQ

```
//10 Số lượng giao dịch bán xe của các người bán hàng top 5
SELECT
    { [Measures].[Fact Vehicle Sales Count] } ON COLUMNS,
    { TOPCOUNT([Dim Seller].[Seller].Members, 5, [Measures].[Fact Vehicle Sales Count]) } ON ROWS
FROM
    [CARSALES];
```

	Fact Vehicle Sales Count
ford motor credit company llc	3311
santander consumer	3281
nissan-infiniti lt	2813
wells fargo dealer services	2373
nissan infiniti lt	2110

- Trong Visual Studio Code

Tạo Named Set:

Name: [TOP 5 Seller by Sale Count]

Expression: TOPCOUNT([Dim Seller].[Seller].Members, 5, [Measures].[Fact Vehicle Sales Count])

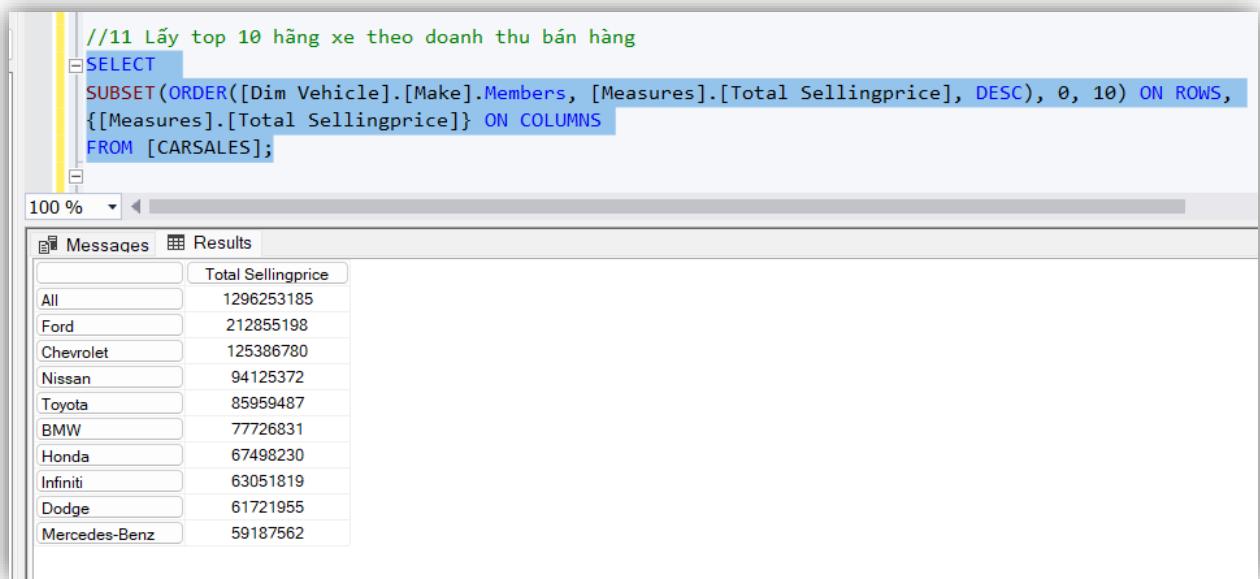
Type: Dynamic

Kết quả:

Seller	Fact Vehicle Sales Count
ford motor credit company llc	3311
nissan infiniti lt	2110
nissan-infiniti lt	2813
santander consumer	3281
wells fargo dealer services	2373

### 3.10.11. Lấy top 10 hãng xe theo doanh thu bán hàng

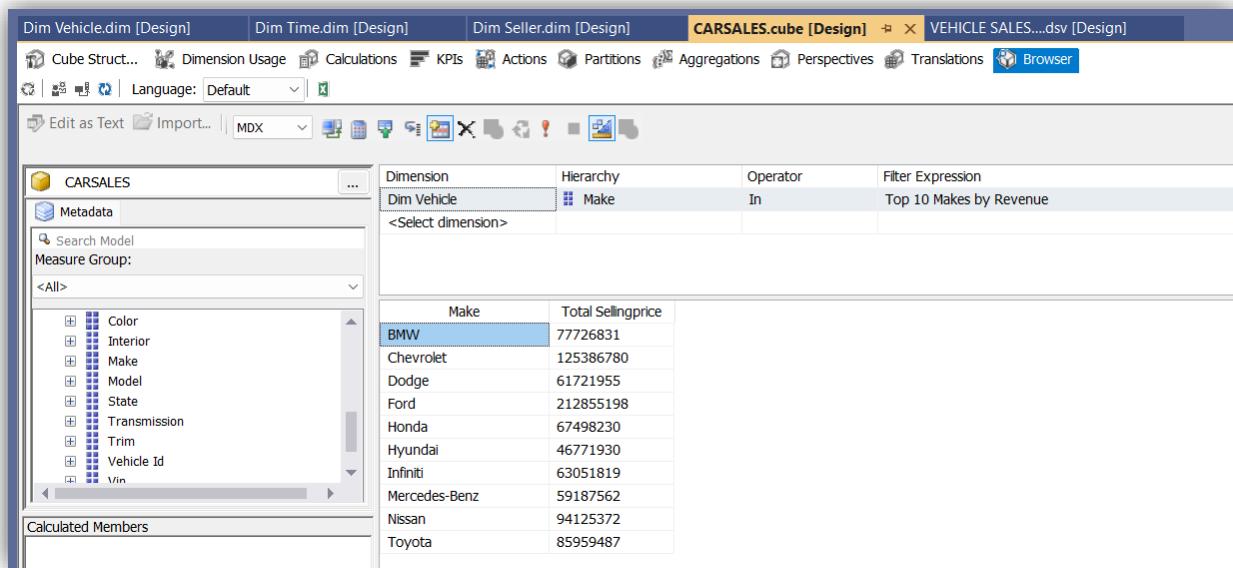
- Trong MSSQ



```
//11 Lấy top 10 hãng xe theo doanh thu bán hàng
SELECT
SUBSET(ORDER([Dim Vehicle].[Make].Members, [Measures].[Total Sellingprice], DESC), 0, 10) ON ROWS,
{[Measures].[Total Sellingprice]} ON COLUMNS
FROM [CARSALES];
```

	Total Sellingprice
All	1296253185
Ford	212855198
Chevrolet	125386780
Nissan	94125372
Toyota	85959487
BMW	77726831
Honda	67498230
Infiniti	63051819
Dodge	61721955
Mercedes-Benz	59187562

- Trong Visual Studio Code



Dimension	Hierarchy	Operator	Filter Expression
Dim Vehicle	Make	In	Top 10 Makes by Revenue

Make	Total Sellingprice
BMW	77726831
Chevrolet	125386780
Dodge	61721955
Ford	212855198
Honda	67498230
Hyundai	46771930
Infiniti	63051819
Mercedes-Benz	59187562
Nissan	94125372
Toyota	85959487

### 3.10.12. Thống kê số lượng xe bán ra theo từng hãng, mẫu xe, loại thân xe

- Trong MSSQ

```
//12 Thống kê số lượng xe bán ra theo từng hãng, mẫu xe, loại thân xe
SELECT
    [Measures].[Fact Vehicle Sales Count] ON COLUMNS,
    {
        [Dim Vehicle].[Make].[Make].Members *
        [Dim Vehicle].[Body].[Body].Members*
        [Dim Vehicle].[Model].[Model].Members*
        [Dim Vehicle].[Color].[Color].Members
    } ON ROWS
FROM
    [CARSALES];
```

100 %

Messages				Results
				Fact Vehicle Sales Count
Acura	Coupe	CL	black	3
Acura	Coupe	CL	blue	2
Acura	Coupe	CL	gold	2
Acura	Coupe	CL	green	1
Acura	Coupe	CL	red	1
Acura	Coupe	CL	silver	5
Acura	Coupe	CL	white	3
Acura	Hatchback	Integra	black	1
Acura	Hatchback	Integra	red	1
Acura	Hatchback	Integra	silver	2
Acura	Hatchback	RSX	black	7
Acura	Hatchback	RSX	blue	3
Acura	Hatchback	RSX	burgundy	1
Acura	Hatchback	RSX	gray	2
Acura	Hatchback	RSX	red	2
Acura	Hatchback	RSX	silver	8

- Trong Visual Studio Code

Make	Body	Model	Color	Fact Vehicle Sales Count
Acura	Coupe	CL	black	3
Acura	Coupe	CL	blue	2
Acura	Coupe	CL	gold	2
Acura	Coupe	CL	green	1
Acura	Coupe	CL	red	1
Acura	Coupe	CL	silver	5
Acura	Coupe	CL	white	3
Acura	Hatchback	Integra	black	1
Acura	Hatchback	Integra	red	1
Acura	Hatchback	Integra	silver	2
Acura	Hatchback	RSX	black	7
Acura	Hatchback	RSX	blue	3
Acura	Hatchback	RSX	burgundy	1
Acura	Hatchback	RSX	gray	2
Acura	Hatchback	RSX	red	2
Acura	Hatchback	RSX	silver	8
Acura	Hatchback	RSX	white	5
Acura	Hatchback	TSX	black	5

### 3.10.13. Tổng doanh thu, số lượng giao dịch bán xe và trung bình giá bán cho các mẫu xe của hãng Porsche, phân loại theo loại thân xe và mẫu xe cụ thể.

- Trong MSSQ

```

//13 Tổng doanh thu, số lượng giao dịch bán xe và trung bình giá bán cho các mẫu xe của hãng Porsche, phân loại theo loại thân xe và mẫu xe cụ thể
SELECT
{[Measures].[Total Sellingprice], [Measures].[Fact Vehicle Sales Count], [Measures].[Average Sellingprice]} ON COLUMNS,
{
    {[Dim Vehicle].[Make].[Make].&[Porsche],
    [Dim Vehicle].[Bcdy].[Body].Members,
    [Dim Vehicle].[Model].[Model].Members
} ON ROWS
FROM
[CARSALES];

//14 Thống kê số lượng giao dịch bán xe và tổng doanh thu theo từng ngày

```

Make	Body	Model	Total Sellingprice	Fact Vehicle Sales Count	Average Sellingprice
Porsche	Convertible	911	605150	12	75643.75
Porsche	Convertible	Boxster	1013100	49	27381.0810810811
Porsche	Coupe	911	1162750	17	105704.545454545
Porsche	Coupe	Cayman	533800	16	38128.5714285714
Porsche	Coupe	Cayman S	63000	3	21000
Porsche	Sedan	Panamera	2676100	49	121640.90909090909
Porsche	SUV	Cayenne	2709750	103	38165.49295774465
Porsche	SUV	Macan	55000	1	55000

- Trong Visual Studio Code

### 3.10.14. Thông kê số lượng giao dịch bán xe và tổng doanh thu theo từng ngày

- Trong MSSQ

			Fact Vehicle Sales Count	Total Sellingprice
2014	1	1	47	514600
2014	1	12	20	414500
2014	1	13	3	76725
2014	1	14	1	14700
2014	1	15	1	23000
2014	1	2	10	132300
2014	1	5	3	48000
2014	1	6	42	741500
2014	1	7	19	326200
2014	1	8	14	195400
2014	12	16	1296	18994950
2014	12	17	2006	23019780
2014	12	18	8387	120251458
2014	12	19	1500	21912150
2014	12	21	16	114525
2014	12	22	1172	14648207

- Trong Visual Studio Code

Year	Month	Date	Fact Vehicle Sales Count	Total Sellingprice
2014	1	1	47	514600
2014	1	12	20	414500
2014	1	13	3	76725
2014	1	14	1	14700
2014	1	15	1	23000
2014	1	2	10	132300
2014	1	5	3	48000
2014	1	6	42	741500
2014	1	7	19	326200
2014	1	8	14	195400
2014	12	16	1296	18994950
2014	12	17	2006	23019780
2014	12	18	8387	120251458
2014	12	19	1500	21912150
2014	12	21	16	114525
2014	12	22	1172	14648207
2014	12	23	7646	86510004
2014	12	24	39	411700

### 3.10.15. Trung bình giá bán và giá trị MMR trung bình theo từng năm:

- Trong MSSQ

```

//15 Trung bình giá bán và giá trị MMR trung bình theo từng năm:
SELECT
    { [Measures].[Average SellingPrice], [Measures].[Average Mmr] } ON COLUMNS,
    [Dim Time].[Year].[Year].Members ON ROWS
FROM
    [CARSALES];

//16 Số lượng giao dịch bán xe theo từng năm và từng tháng trong năm đó
SELECT

```

Year	Average Sellingprice	Average Mmr
2014	1482613.7260274	1508185.2739726
2015	645721.747943156	660083.657442034
Unknown	(null)	(null)

- Trong Visual Studio Code

### 3.10.16. Số lượng giao dịch bán xe theo từng năm và từng tháng trong năm đó

- Trong MSSQ

```
//16 Số lượng giao dịch bán xe theo từng năm và từng tháng trong năm đó
SELECT
    [Measures].[Fact Vehicle Sales Count] ON COLUMNS,
    [Dim Time].[Year].[Year].Members *
    [Dim Time].[Month].[Month].Members ON ROWS
FROM
    [CARSALES];
//17 Tổng doanh thu từ việc bán xe trong năm 2015 và so sánh với năm 2014
```

		Fact Vehicle Sales Count
2014	1	160
2014	12	34313
2014	2	1
2015	1	59873
2015	2	4771
2015	3	575
2015	4	12
2015	5	61
2015	6	157
2015	7	77
Unknown	Unknown	(null)

- Trong Visual Studio Code

### 3.10.17. Tổng doanh thu từ việc bán xe trong năm 2015 và so sánh với năm 2014

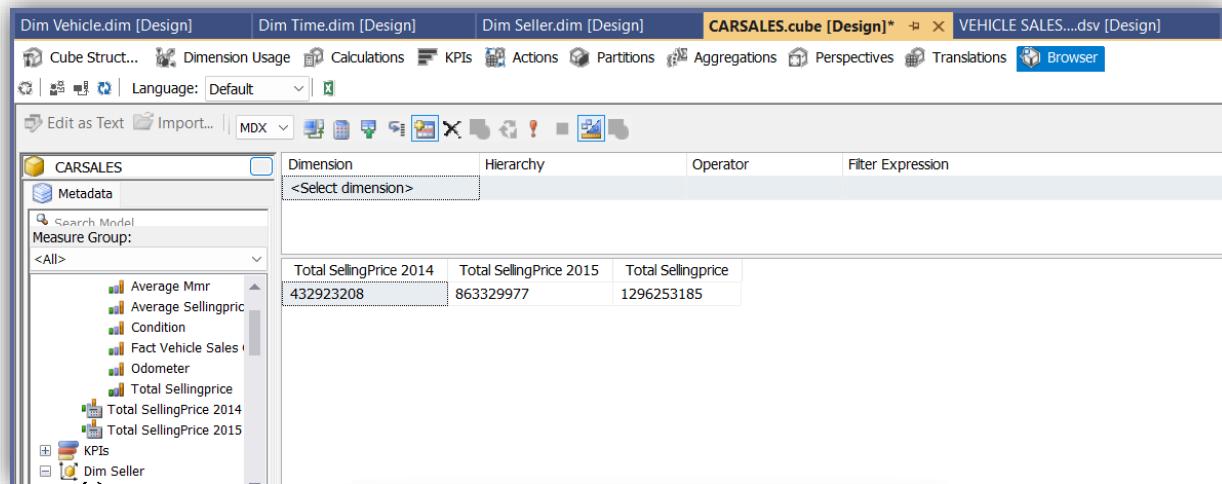
- Trong MSSQ

```

//17 Tổng doanh thu từ việc bán xe trong năm 2015 và so sánh với năm 2014
WITH
  MEMBER [Measures].[Total SellingPrice 2015] AS
    ([Measures].[Total SellingPrice], [Dim Time].[Year].&[2015])
  MEMBER [Measures].[Total SellingPrice 2014] AS
    ([Measures].[Total SellingPrice], [Dim Time].[Year].&[2014])
SELECT
  {
    [Measures].[Total SellingPrice 2015],
    [Measures].[Total SellingPrice 2014],
    [Measures].[Total SellingPrice]
  } ON COLUMNS
FROM
  [CARSALES];
  
```

Total SellingPrice 2015	Total SellingPrice 2014	Total Sellingprice
863329977	432923208	1296253185

- Trong Visual Studio Code



### 3.10.18 Cho biết điều kiện xe tốt nhất mà seller đã bán theo từng năm

- Trong MSSQ

Condition	Year	Value
1 cochranch monroeville	2014	43
1 cochranch monroeville	2015	48
1 cochranch monroeville	Unknown	(null)
143 auto sales inc	2014	(null)
143 auto sales inc	2015	38
143 auto sales inc	Unknown	(null)
159191 canada inc	2014	(null)
159191 canada inc	2015	49
159191 canada inc	Unknown	(null)
1st advantage fcu	2014	(null)
1st advantage fcu	2015	39
1st advantage fcu	Unknown	(null)
1st capital finance	2014	(null)
1st capital finance	2015	2
1st capital finance	Unknown	(null)
1st choice automotive corp	2014	36
1st choice automotive corp	2015	(null)
1st choice automotive corp	Unknown	(null)
1st commercial	2014	(null)
1st commercial	2015	28
1st commercial	Unknown	(null)
1st liberty fcu	2014	(null)
1st liberty fcu	2015	19
1st liberty fcu	Unknown	(null)
1st mid america credit union	2014	31
1st mid america credit union	2015	25
1st mid america credit union	Unknown	(null)

- Trong Visual Studio Code

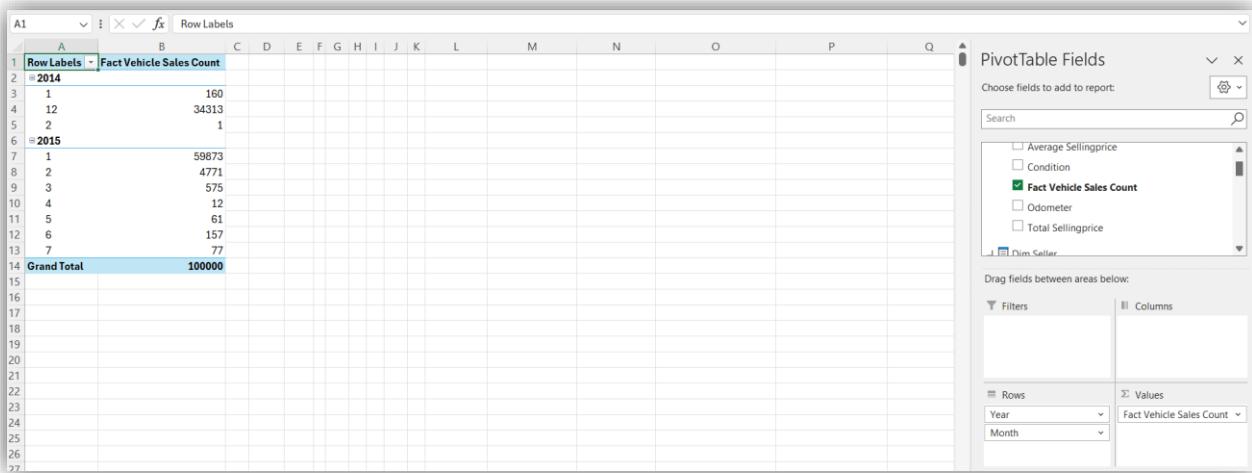
Seller	Year	Condition
1 cochranch of monroeville	2014	43
1 cochranch of monroeville	2015	48
143 auto sales inc	2015	38
159191 canada inc	2015	49
1st advantage fcu	2015	39
1st capital finance	2015	2
1st choice automotive corp	2014	36
1st commercial	2015	28
1st liberty fcu	2015	19
1st mid america credit union	2014	31
1st mid america credit union	2015	25
1st national bank of scotia	2014	29
1st national bank of scotia	2015	35
22nd street motors inc	2014	32
22nd street motors inc	2015	44
231 car sales inc.	2014	44
231 car sales inc.	2015	49
281 truck sales	2015	2

### 3.11. Phân tích bằng Pivot table trong Excel

**Câu 1:** Cho biết Tổng doanh thu từ việc bán xe của mỗi người bán

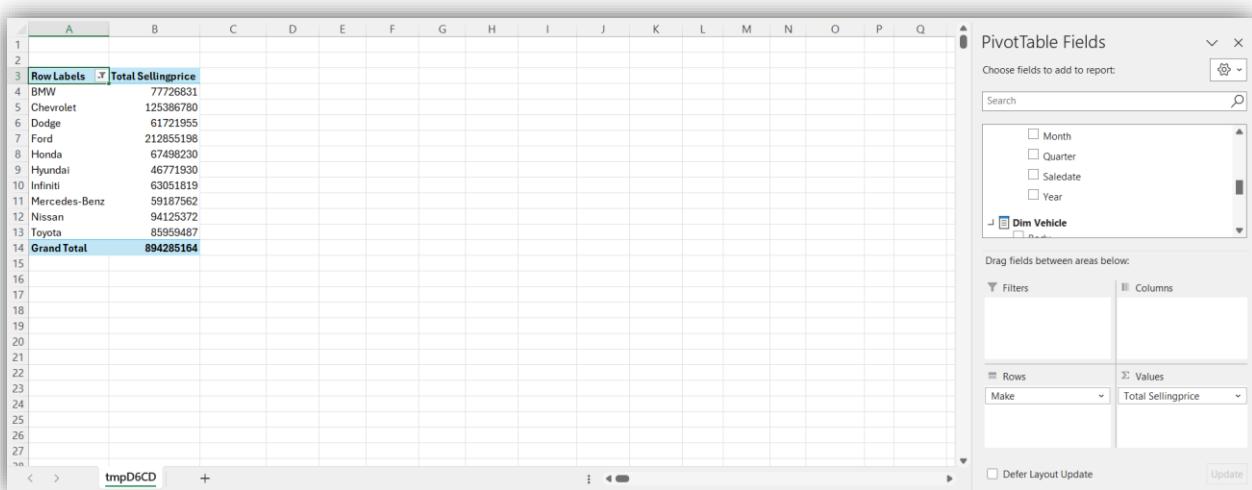
Row Labels	Total Sellingprice
1 cochranch of monroeville	317000
143 auto sales inc	10500
159191 canada inc	442300
1st advantage fcu	41600
1st capital finance	400
1st choice automotive corp	11800
1st commercial	10000
1st liberty fcu	4200
1st mid america credit union	29600
1st national bank of scotia	18100
22nd street motors inc.	57300
231 car sales inc.	204550
281 truck sales	1000
3 amigos auto sales inc	2200
3 in one auto sales inc	28995
3 line motors llc	6600
3:16 auto sales	500
355 toyota	110350
4h auto sales llc	2000
46 vans & trucks llc	12500
77th street depot federal credit union	10800
800 loan mart	176200
800cash247	59650
888 motors	13100
91 automotive west	44500
9209 0851 quebec inc	139600

**Câu 2:** Số lượng giao dịch bán xe theo từng năm và từng tháng trong năm đó



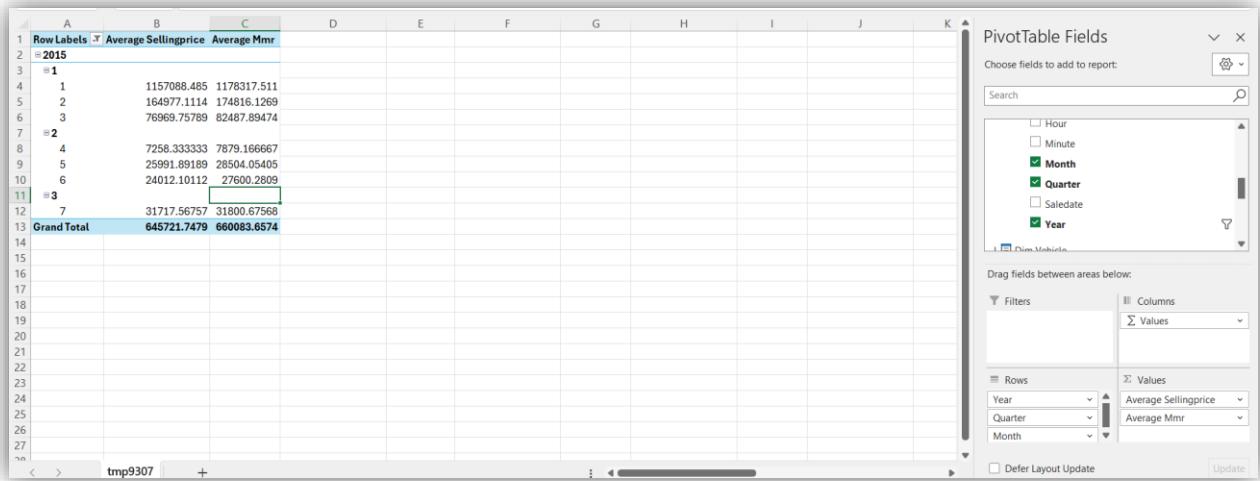
Row Labels	Fact Vehicle Sales Count
2014	
1	160
12	34313
2	1
2015	
1	59873
2	4771
3	575
4	12
5	61
6	157
7	77
Grand Total	1000000

**Câu 3:** Top 10 hãng xe có doanh thu bán xe cao nhất



Row Labels	Total Sellingprice
BMW	7726831
Chevrolet	125386780
Dodge	61721955
Ford	212855198
Honda	67498230
Hyundai	46771930
Infiniti	63051819
Mercedes-Benz	59187562
Nissan	94125372
Toyota	85959487
Grand Total	894285164

**Câu 4:** Trung bình doanh thu theo từng tháng, quý trong năm 2015



Row Labels: Average Sellingprice Average Mmr

2015

1 1 1157088.485 1178317.511

2 2 164977.1114 174816.1269

3 3 76969.75789 82487.89474

4 4 7258.33333 7879.166667

5 5 25991.89189 28504.05405

6 6 24012.10112 27600.2809

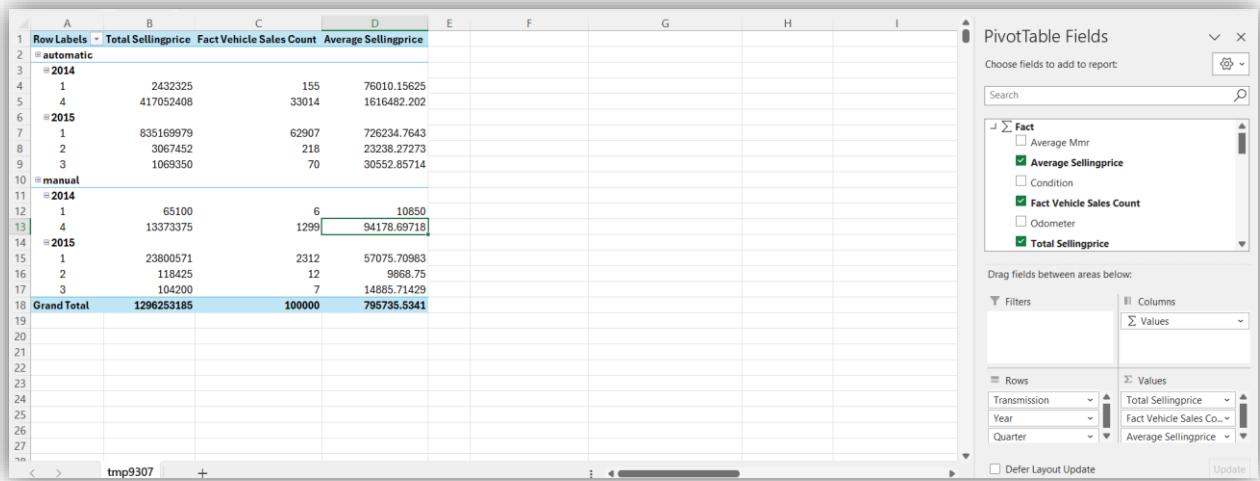
7 7 31717.56757 31800.67568

Grand Total 645721.7479 660083.6574

Rows: Year, Quarter, Month

Values: Average Sellingprice, Average Mmr

**Câu 5:** Cho biết tổng doanh thu, số lượng giao dịch và trung bình giá bán xe theo từng quý, năm của mỗi loại hộp số sàn và hộp số tự động



Row Labels: Total Sellingprice Fact Vehicle Sales Count Average Sellingprice

automatic

2014

1 2432325 155 76010.15625

4 417052408 33014 1616482.202

2015

1 835169979 62907 726234.7643

2 3067452 218 23238.27273

3 1069350 70 30552.85714

manual

2014

1 65100 6 10850

4 13373375 1299 94178.69718

2015

1 23800571 2312 57075.70983

2 118425 12 9868.75

3 104200 7 14885.71429

Grand Total 1296253185 100000 795735.5341

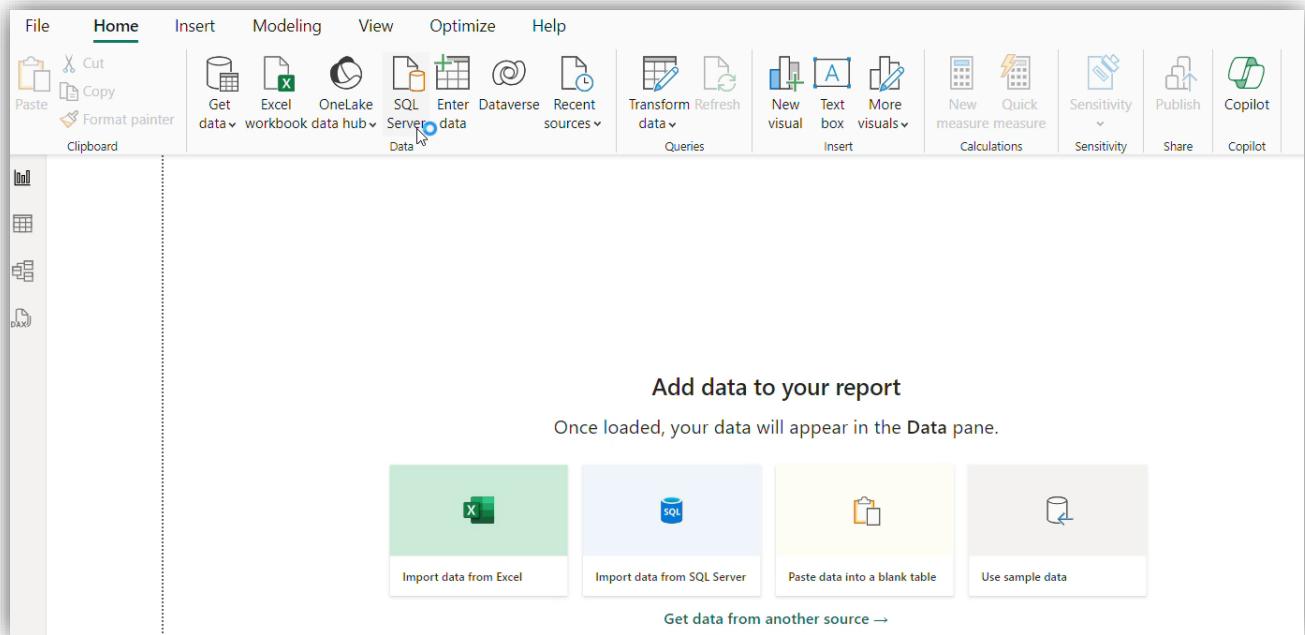
Rows: Transmission, Year, Quarter

Values: Total Sellingprice, Fact Vehicle Sales Count, Average Sellingprice

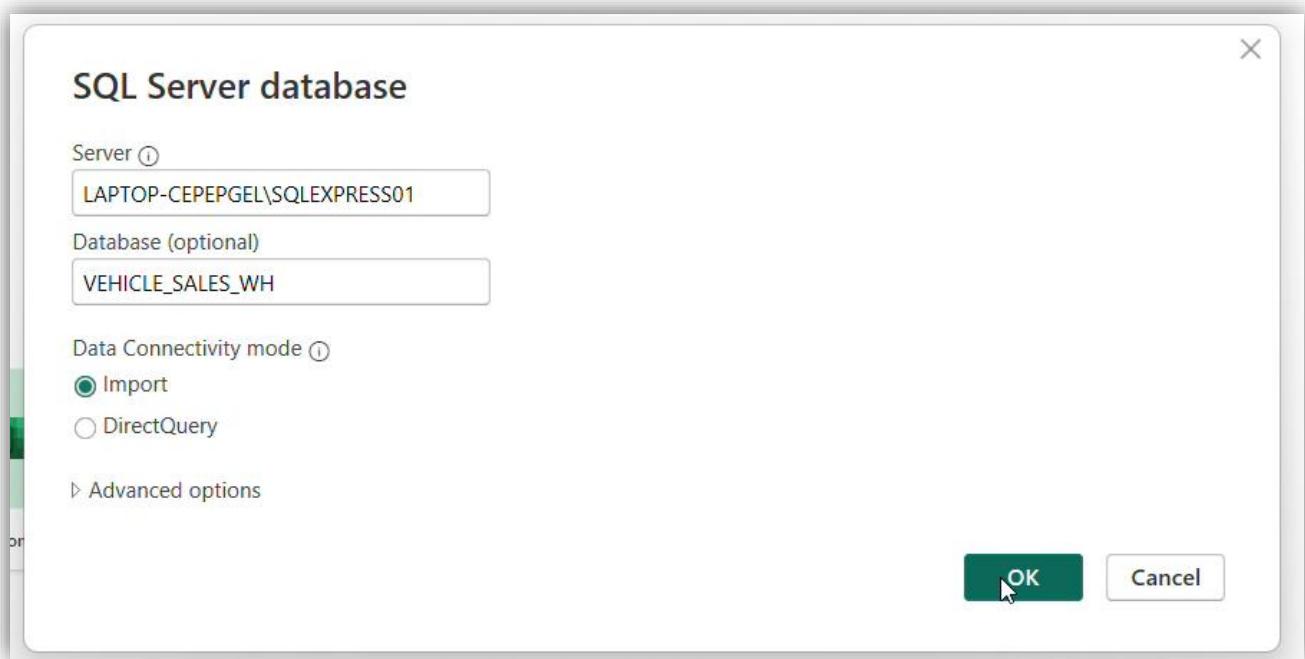
### 3.12 Quá trình lập báo biểu bằng công cụ Power BI

#### 3.12.1. Kết nối Power BI với SQL Server

**Bước 1.** Mở Power BI. Tại mục Data của tab Home, chọn SQL Server.

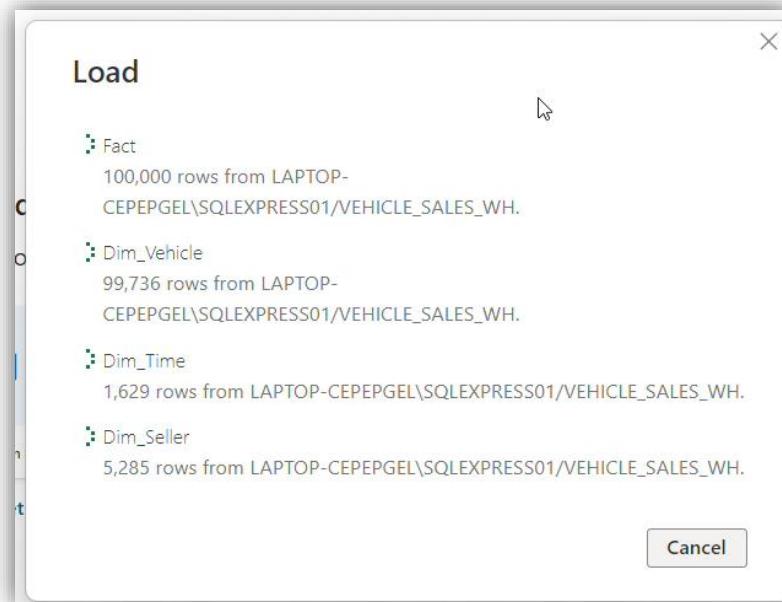
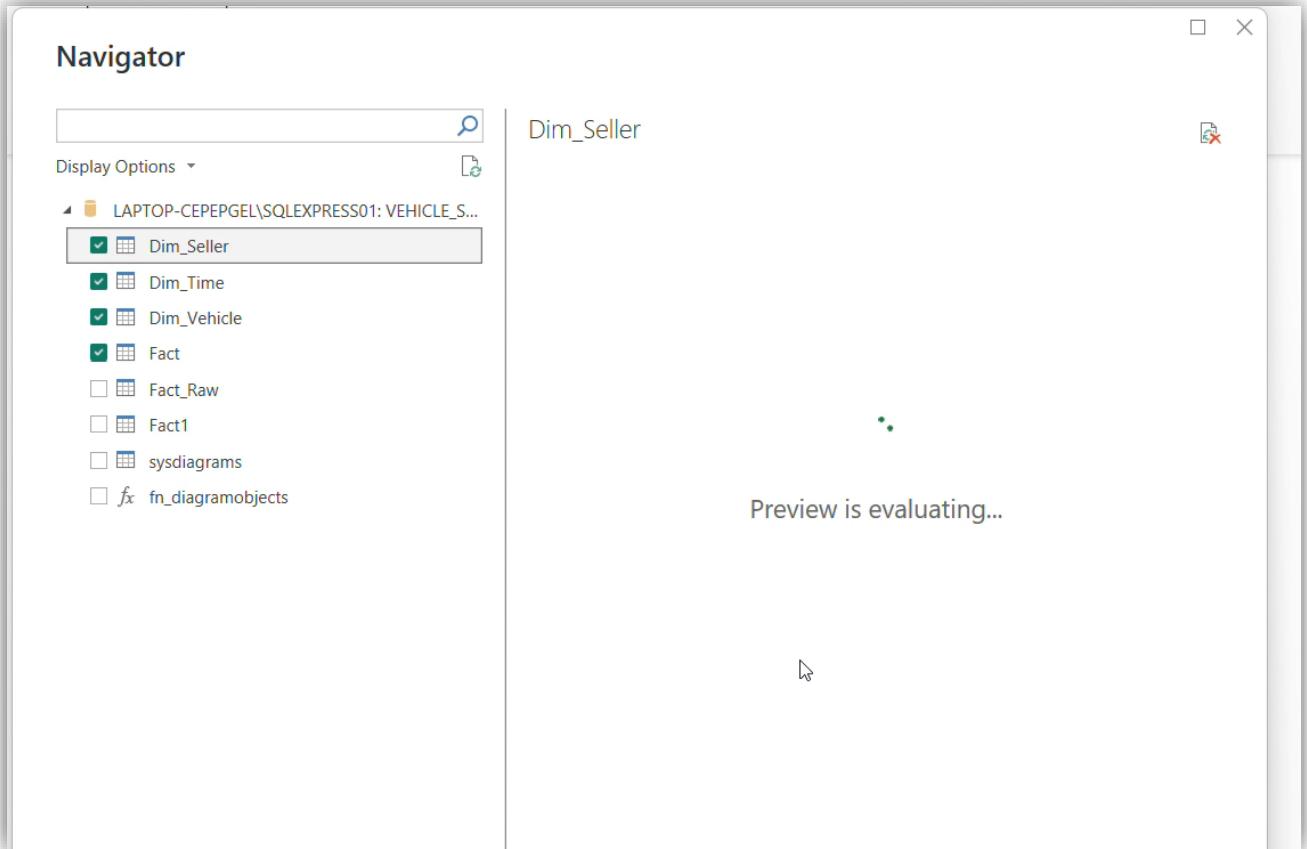


**Bước 2.** Nhập tên Server và tên Database rồi nhấn OK.

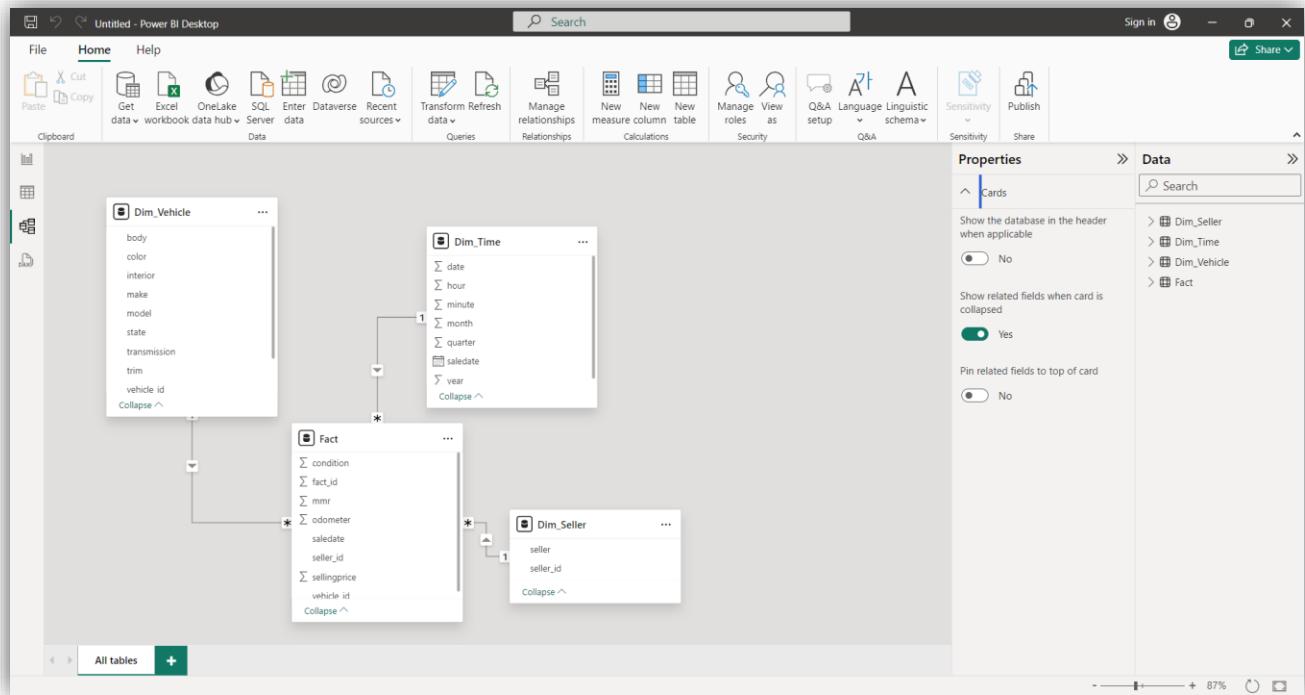


**Bước 3.** Xuất hiện một cửa sổ thực hiện đăng nhập vào Sql Server bằng 3 cách: user Windows, user SQL Server và Microsoft Account. Tại đây ta đăng nhập bằng tài khoản sa của database rồi nhấn nút Connect.

**Bước 4.** Thực hiện chọn bảng Fact và các bảng Dim và đợi Power BI load dữ liệu.



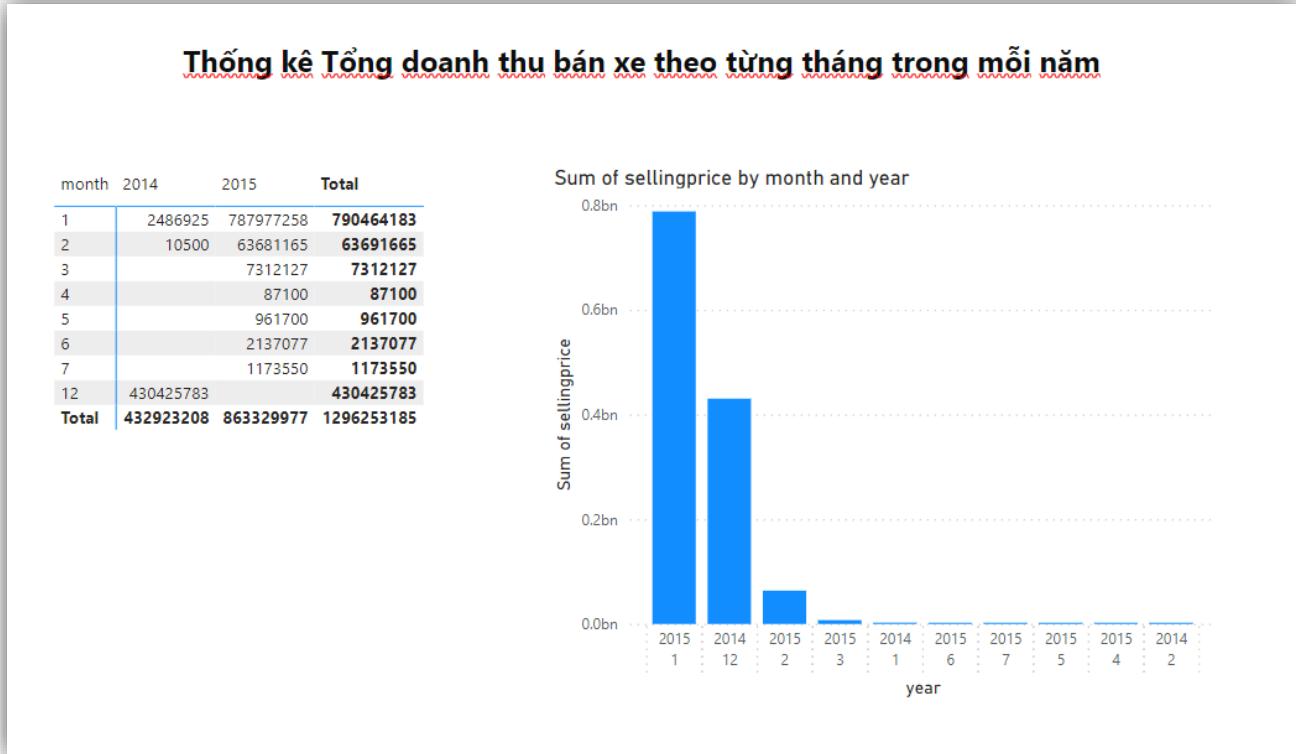
**Bước 5.** Sau khi Power BI load dữ liệu xong, nhấn vào biểu tượng Model, ta được Relationship như sau:



### 3.12.2. Thực hiện Report bằng POWER BI

#### 3.12.2.1. Câu truy vấn 1

Nội dung báo biểu: Thông kê Tổng doanh thu bán xe theo từng tháng trong mỗi năm



Đơn vị tiền tệ: đô

- **Nhận xét về biểu đồ:**

Năm 2015: Tháng 1 và tháng 12 có tổng doanh thu cao nhất, với tháng 1 đạt gần 0.8 tỷ đô và tháng 12 đạt khoảng 0.6 tỷ đô.

Năm 2014: Tháng 12 có doanh thu đáng kể, đạt gần 0.4 tỷ đô. Các tháng khác trong năm 2014 có doanh thu rất thấp so với tháng 12.

Sự chênh lệch giữa các tháng: Có sự chênh lệch rõ rệt về doanh thu giữa các tháng. Một số tháng như tháng 1 và tháng 12 có doanh thu rất cao, trong khi các tháng còn lại như tháng 2, 4, 5, và 7 có doanh thu rất thấp.

- **Kết luận:**

Tháng 1 và tháng 12 năm 2015 ghi nhận doanh thu cao nhất, đặc biệt tháng 1 đạt gần 0.8 tỷ và tháng 12 đạt gần 0.6 tỷ. Điều này có thể liên quan đến các chiến dịch khuyến mãi mùa vụ như giảm giá cuối năm hoặc đầu năm mới.

Tháng 12 năm 2014 cũng có doanh thu đáng kể, đạt gần 0.4 tỷ.

Các tháng còn lại trong cả hai năm có doanh thu thấp hơn nhiều, với một số tháng như

tháng 4, 5 và 7 năm 2015 có doanh thu dưới 1.5 triệu.

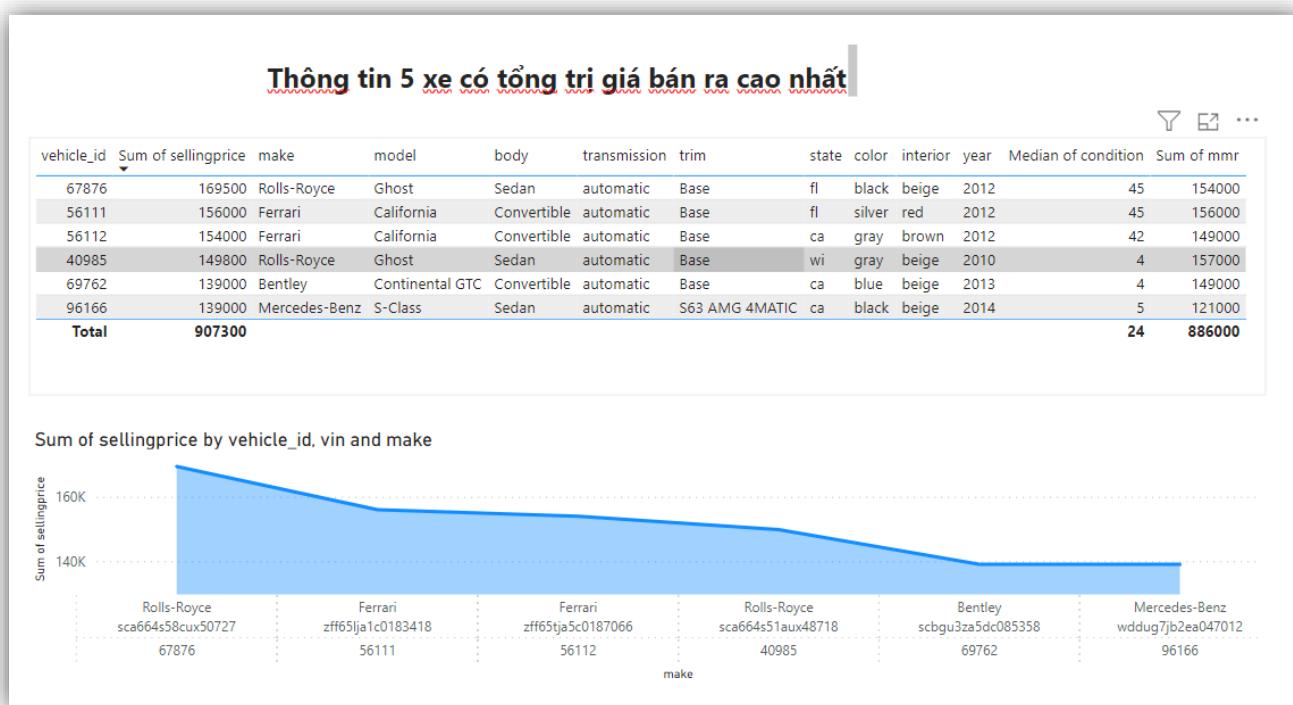
Sự chênh lệch doanh thu giữa các tháng cho thấy có sự ảnh hưởng lớn của mùa vụ và có thể là các yếu tố khác như chiến lược marketing, nhu cầu thị trường thay đổi theo thời gian.

Hiệu suất bán hàng tổng thể: Năm 2015 có tổng doanh thu là 863,299,977, cao hơn nhiều so với năm 2014 với tổng doanh thu là 432,923,208. Tổng cộng cả hai năm đạt 1,296,253,185.

Điều này cho thấy năm 2015 có chiến lược bán hàng hiệu quả hơn hoặc có thể do thị trường xe hơi có sự tăng trưởng mạnh mẽ trong năm này.

### 3.12.2.2 Câu truy vấn 2

Nội dung báo biểu: Thông tin chi tiết 5 xe có tổng trị giá bán ra cao nhất



Đơn vị tiền tệ: đô

- **Nhận xét:**

Bảng hiển thị chi tiết thông tin của 5 xe có tổng giá trị bán ra cao nhất, bao gồm các thông tin:

Vehicle ID, Sum of Selling Price, Make, Model, Body, Transmission, Trim, State, Color,

Interior, Year, Median of Condition, Sum of MMR.

Tổng giá trị bán ra (Sum of Selling Price) của 5 xe này là 907,300.

Tổng giá trị MMR (Sum of MMR) của 5 xe này là 886,000.

Trung bình điều kiện xe (Median of Condition): Giá trị trung bình là 24.

Xe có giá trị bán cao nhất là Rolls-Royce Ghost (Vehicle ID: 67876) với giá trị bán là 169,500.

Các xe còn lại có giá trị bán dao động từ 139,000 đến 156,000.

- **Kết luận:**

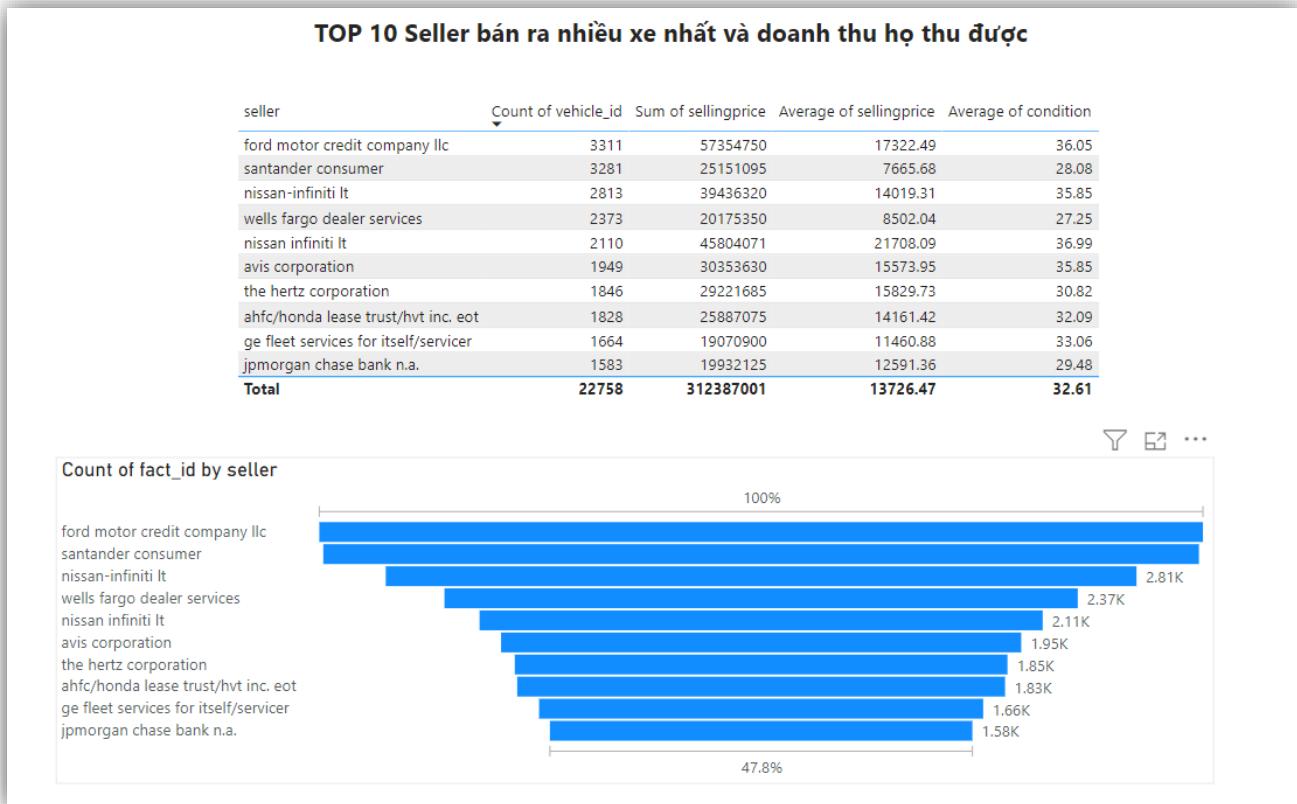
Đa số các xe có năm sản xuất từ năm 2012 đều có tình trạng tốt (Median of Condition từ 42 đến 45) và được bán với giá cao.

Xe có giá trị bán cao nhất là Xe Rolls-Royce Ghost (Vehicle ID: 67876) có giá trị bán cao nhất là 169,500 đô.

Các xe Ferrari California (Vehicle ID: 56111 và 56112) cũng có giá trị bán cao, lần lượt là 156,000 và 154,000. Các xe có giá trị bán cao đều là những dòng xe sang trọng và nổi tiếng như Rolls-Royce, Ferrari, Bentley, và Mercedes-Benz.

### 3.12.2.3 Câu truy vấn 3

Nội dung báo biểu: TOP 10 Seller bán ra nhiều xe nhất, doanh thu họ thu được và 1 vài chỉ số khác



- **Nhận xét về bảng dữ liệu:**

Về số lượng xe bán ra:

- + Ford Motor Credit Company LLC đứng đầu danh sách với 3,311 xe bán ra, theo sau là Santander Consumer với 3,281 xe.
- + JP Morgan Chase Bank N.A. có số lượng xe bán ít nhất trong top 10 với 1,583 xe.

Về tổng doanh thu:

- + Ford Motor Credit Company LLC cũng đứng đầu về tổng doanh thu từ việc bán xe với 573,547,50 USD.
- + Santander Consumer mặc dù bán số lượng xe gần tương đương với Ford Motor Credit Company LLC nhưng tổng doanh thu thấp hơn đáng kể, chỉ đạt 251,510,95 USD.

Về Hiệu quả kinh tế:

- + Nissan-Infiniti LT có doanh thu khá cao (394,363,20 USD) với số lượng xe bán ra 2,813 xe, cho thấy giá bán trung bình mỗi xe có thể cao hơn so với một số đối thủ

khác.

- + Avis Corporation và The Hertz Corporation, hai công ty cho thuê xe, cũng có doanh thu khá cao tương ứng với số lượng xe bán ra.

#### **Kết luận:**

- + Ford Motor Credit Company LLC và Santander Consumer chiếm thị phần lớn nhất về số lượng xe bán ra, cho thấy sự ảnh hưởng mạnh mẽ của hai công ty này trên thị trường.
- + Nissan-Infiniti LT đạt doanh thu cao nhờ vào giá bán trung bình mỗi xe cao hơn, mặc dù số lượng xe bán ra ít hơn. Điều này cho thấy công ty có thể tập trung vào phân khúc xe giá trị cao.
- + Ford Motor Credit Company LLC mặc dù bán được nhiều xe nhất, nhưng doanh thu trung bình mỗi xe không cao bằng một số đối thủ khác. Điều này có thể chỉ ra rằng họ tập trung vào phân khúc thị trường xe giá rẻ hoặc xe cũ.
- + Điều kiện trung bình của xe là một yếu tố quan trọng khi đánh giá chất lượng sản phẩm và có thể ảnh hưởng đến giá bán trung bình. Nissan Infiniti LT và Ford Motor Credit Company LLC có điều kiện xe tốt nhất, có thể giải thích vì sao giá bán trung bình của họ cao.

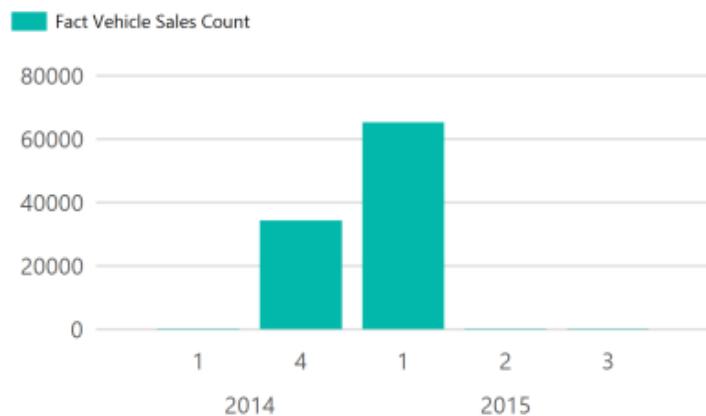
Tổng kết lại, Ford Motor Credit Company LLC và Santander Consumer dẫn đầu về số lượng xe bán ra, trong khi Nissan-Infiniti LT đạt doanh thu cao nhờ vào giá bán trung bình mỗi xe cao hơn và điều kiện xe tốt. Điều này cung cấp một cái nhìn toàn diện về tình hình bán xe và doanh thu trong ngành công nghiệp ô tô.

#### **3.12.3 Thực hiện Report bằng Visual Studio**

##### **3.12.3.1 Câu truy vấn 4**

### Số lượng giao dịch và tổng doanh thu theo từng quý

Year	Quarter	Fact Vehicle Sales Count	Total Sellingprice
2014	1	161	2497425
2014	4	34313	430425783
2015	1	65219	858970550
2015	2	230	3185877
2015	3	77	1173550



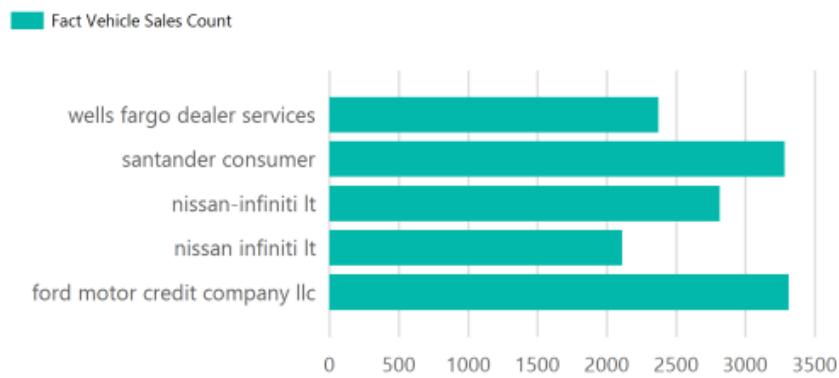
- Năm 2014 chứng kiến sự tăng trưởng mạnh mẽ về số lượng xe bán ra và tổng doanh thu, đặc biệt là trong quý 4.
- Quý 1 năm 2015 là thời điểm đỉnh cao nhất về số lượng xe bán ra và tổng doanh thu.

- Sau quý 1 năm 2015, số lượng xe bán ra và tổng doanh thu giảm mạnh, đặc biệt là trong quý 2 và quý 3 năm 2015.

### 3.12.3.2 Câu truy vấn 5

Thống kê số lượng xe bán ra theo từng quý của top 5 seller bán nhiều xe nhất

Seller	Year	Quarter	Fact Vehicle Sales Count
ford motor credit company llc	2014	1	18
ford motor credit company llc	2014	4	672
ford motor credit company llc	2015	1	2609
ford motor credit company llc	2015	2	7
ford motor credit company llc	2015	3	5
nissan infiniti lt	2014	4	637
nissan infiniti lt	2015	1	1468
nissan infiniti lt	2015	2	2
nissan infiniti lt	2015	3	3
nissan-infiniti lt	2014	1	1
nissan-infiniti lt	2014	4	818
nissan-infiniti lt	2015	1	1986
nissan-infiniti lt	2015	2	8
santander consumer	2014	4	1581
santander consumer	2015	1	1697
santander consumer	2015	2	2
santander consumer	2015	3	1
wells fargo dealer services	2014	1	4
wells fargo dealer services	2014	4	1203
wells fargo dealer services	2015	1	1164
wells fargo dealer services	2015	3	2



- Nhận xét:
  - Xu hướng tăng trưởng mạnh mẽ trong quý 1 năm 2015: Các seller hàng đầu như Ford Motor Credit Company LLC, Nissan Infiniti LT, Nissan-Infiniti LT,

Santander Consumer và Wells Fargo Dealer Services đều đạt đỉnh doanh số trong quý 1 năm 2015.

- + Giảm sút nghiêm trọng sau quý 1 năm 2015: Sau sự tăng trưởng mạnh mẽ, số lượng xe bán ra của tất cả các seller đều giảm mạnh trong các quý tiếp theo.

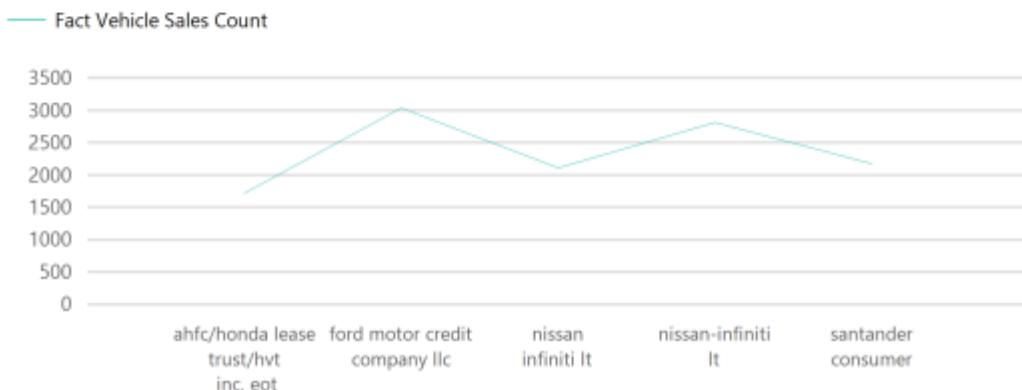
- Kết luận:

Nhìn chung, các seller hàng đầu này đều đạt đỉnh doanh số vào quý 1 năm 2015 sau một sự tăng trưởng ổn định trong năm 2014, nhưng sau đó số lượng xe bán ra giảm mạnh trong các quý tiếp theo. Điều này có thể do nhiều yếu tố như thay đổi trong chiến lược kinh doanh, thay đổi trong nhu cầu thị trường, hoặc sự cạnh tranh gia tăng. Cần phân tích thêm để hiểu rõ nguyên nhân cụ thể và điều chỉnh chiến lược kinh doanh cho phù hợp để cải thiện doanh số bán hàng trong các quý tiếp theo.

### 3.12.3.3 Câu truy vấn 6

## Thống kê số lượng xe bán được của 5 seller bán nhiều xe nhất trong 10 hãng xe có tổng doanh thu cao nhất

Make	Seller	Fact Vehicle Sales Count
BMW	santander consumer	40
Chevrolet	ford motor credit company llc	2
Chevrolet	santander consumer	574
Dodge	ford motor credit company llc	3
Dodge	santander consumer	408
Ford	ford motor credit company llc	3023
Ford	santander consumer	306
Honda	ahfc/honda lease trust/hvt inc. eot	1718
Honda	ford motor credit company llc	3
Honda	santander consumer	138
Hyundai	ford motor credit company llc	2
Hyundai	santander consumer	123
Infiniti	nissan infiniti lt	2110
Infiniti	santander consumer	20
Mercedes-Benz	ford motor credit company llc	1
Mercedes-Benz	santander consumer	28
Nissan	ford motor credit company llc	3
Nissan	nissan-infiniti lt	2813
Nissan	santander consumer	316
Toyota	ford motor credit company llc	1
Toyota	santander consumer	221



- **Nhận xét:**

- + Ford là hãng xe bán chạy nhất với tổng 3329 chiếc, bao gồm cả dòng xe Ford và các dòng khác do ford motor credit company llc phân phối.
- + Dodge cũng đạt doanh số cao với 411 chiếc, chủ yếu đến từ santander consumer

(408 chiếc).

- + Các hãng xe Nhật Bản như Honda, Hyundai và Nissan cũng có doanh số khá cao từ cả hai nhà bán lẻ chính.
- + Một số hãng xe cao cấp như Mercedes-Benz và Infiniti có doanh số thấp hơn so với phân khúc phổ thông.
- + Ford motor credit company llc và santander consumer là hai nhà bán lẻ lớn nhất, chiếm phần lớn doanh số bán xe trong top 5.
- + Có sự khác biệt đáng kể về doanh số giữa hai nhà bán lẻ hàng đầu và các nhà bán lẻ khác trong top 5.

- **Kết luận:**

Dữ liệu cho thấy ford motor credit company llc và santander consumer là hai "ông lớn" thống trị thị trường bán lẻ ô tô, đặc biệt là với các dòng xe phổ thông và bình dân như Ford và Dodge. Các hãng xe Nhật với thương hiệu vững chắc cũng có doanh số cao nhờ chất lượng và uy tín. Trong khi đó, các hãng xe cao cấp có doanh số khiêm tốn hơn, phù hợp với đối tượng khách hàng hẹp hơn. Xu hướng này phản ánh nhu cầu và sức tiêu thụ chính của thị trường tiêu dùng hiện nay. Các nhà bán lẻ cần dựa vào dữ liệu này để có chiến lược kinh doanh và hoạch định nguồn cung phù hợp.

## CHƯƠNG 4. QUÁ TRÌNH KHAI THÁC DỮ LIỆU (DATA MINING)

**Chủ đề:** Sử dụng các thuật toán học máy để xây dựng mô hình dự đoán giá bán của xe Volvo S60 dựa trên các yếu tố như năm sản xuất, tình trạng, số km đã đi (odometer), giá trị MMR, và các đặc điểm khác của xe.

### 4.1. Phân tích dataset gốc

#### 4.1.1. Thông kê mô tả

Tính toán đại lượng thống kê mô tả: Count, Min, Max, Mean, Median, Quantile, Range, Mode và Variance trên tập dữ liệu.

**Bước 1:** Import các thư viện cần thiết và đọc dữ liệu từ file csv.

df = pd.read_csv('/content/drive/MyDrive/olap/FINALDATA_VEHICLE.csv')																			year		make		model		trim		body		transmission		vin		state		condition		odometer		color		interior		seller		mmr		sellingprice		saledate	
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america inc	20500.0	21500.0	2014-12-16 04:30:00																																		
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america inc	20800.0	21500.0	2014-12-16 04:30:00																																		
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45.0	1331.0	gray	black	financial services remarketing (lease)	31900.0	30000.0	2015-01-14 20:30:00																																		
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni	27500.0	27750.0	2015-01-28 20:30:00																																		
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43.0	2641.0	gray	black	financial services remarketing (lease)	66000.0	67000.0	2014-12-18 04:30:00																																		
filtered_df = df[(df['make'] == 'Volvo') & (df['model'] == 'S60')]																																																		
# Hiển thị dữ liệu đã lọc																																																		
year		make		model		trim		body		transmission		vin		state		condition		odometer		color		interior		seller		mmr		sellingprice		saledate																				
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni	27500.0	27750.0	2015-01-28 20:30:00																																		
558	2013	Volvo	S60	T5	Sedan	automatic	yv1612fs2d2200121	ca	42.0	19942.0	white	beige	volvo car financial service/world omni	15550.0	20000.0	2014-12-18 04:30:00																																		
632	2013	Volvo	S60	T5	Sedan	automatic	yv1612fs1d2225611	ca	37.0	27368.0	white	black	volvo car financial service/world omni	18550.0	19000.0	2014-12-18 04:30:00																																		
1470	2012	Volvo	S60	T5	Sedan	automatic	yv1622fsxc2098367	ca	29.0	32782.0	gray	black	fiserv/usb dealer services northstar exchange	16250.0	15700.0	2014-12-18 04:30:00																																		
1480	2012	Volvo	S60	T5	Sedan	automatic	yv1622fs5c2098583	ca	41.0	60252.0	white	black	us bank	13900.0	14750.0	2014-12-18 04:30:00																																		

**Bước 2:** Nhận thấy rằng dataset có giá trị các cột bao gồm hai loại dữ liệu số là Int và Float. Ta tiến hành lọc ra các cột có kiểu Int và Float để tính toán.

```
[119] numeric_cols = filtered_df.select_dtypes(include=['int', 'float']).columns
```

**Bước 3:** Sử dụng phương thức describe() để tính toán thống kê mô tả cơ bản về số lượng, trung bình, độ lệch chuẩn, giá trị tối thiểu, giá trị tối đa và các phân vị(quantiles) cho tất cả các cột có kiểu dữ liệu là Int hoặc Float.



```
stats = filtered_df[numeric_cols].describe().T
```

**Bước 4:** Tính toán thêm các giá trị mode, range và variance cho tất cả các cột có kiểu dữ liệu là Int hoặc Float.

```
for col in filtered_df[numeric_cols].columns:
    stats.loc[col, 'mode'] = filtered_df[col].mode()[0]

for col in filtered_df[numeric_cols].columns:
    stats.loc[col, 'range'] = filtered_df[col].max() - filtered_df[col].min()

for col in filtered_df[numeric_cols].columns:
    stats.loc[col, 'variance'] = filtered_df[col].var ()
```

**Bước 5:** Xem kết quả thống kê mô tả dữ liệu.

	count	mean	std	min	25%	50%	75%	max	mode	range	variance
year	886.0	2009.980813	4.613255	2001.0	2006.00	2012.0	2014.0	2015.0	2012.0	14.0	2.128212e+01
condition	886.0	32.291196	11.860911	1.0	26.00	36.0	42.0	49.0	44.0	48.0	1.406812e+02
odometer	886.0	64343.363431	57415.098251	1.0	18934.50	39780.5	102421.0	355696.0	1.0	355695.0	3.296494e+09
mmr	886.0	13146.444695	8357.172994	250.0	3781.25	15450.0	21000.0	28500.0	22800.0	28250.0	6.984234e+07
sellingprice	886.0	12999.463883	8363.004969	300.0	3625.00	15200.0	20300.0	28400.0	23500.0	28100.0	6.993985e+07

#### 4.1.2. Trực quan hóa dữ liệu

##### 4.1.2.1. Tạo biểu đồ để hiển thị số lượng xe trong từng loại tình trạng.

```

import matplotlib.pyplot as plt
import seaborn as sns

condition_counts = filtered_df.groupby("condition")["sellingprice"].count()

plt.figure(figsize=(14, 8))
sns.set(style="whitegrid")

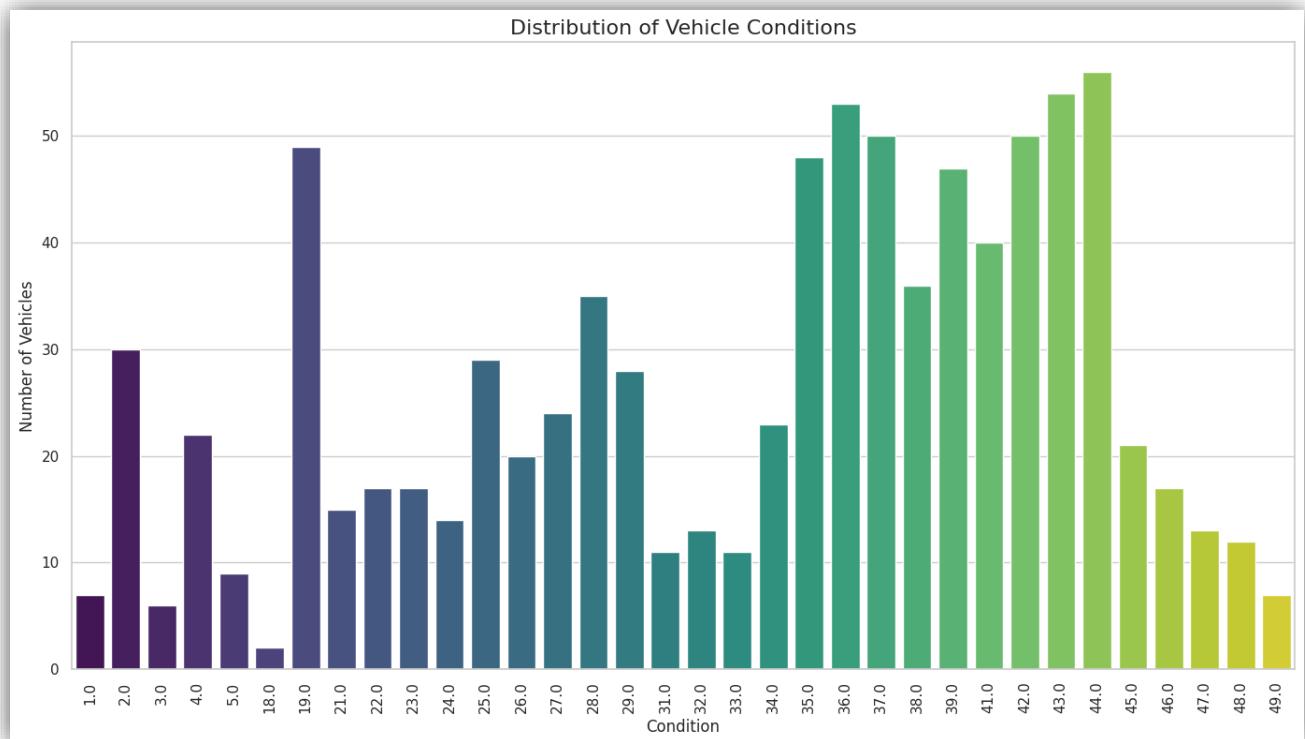
ax = sns.barplot(x=condition_counts.index, y=condition_counts.values, palette="viridis")

ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

plt.xlabel("Condition", fontsize=12)
plt.ylabel("Number of Vehicles", fontsize=12)
plt.title("Distribution of Vehicle Conditions", fontsize=16)

plt.tight_layout()
plt.show()

```



- Với tình trạng xe được đánh giá theo thang điểm từ 1 đến 49, ta có thể chia thành các nhóm để đánh giá mức độ tình trạng của xe:
  - 1-10: Rất kém, xe có nhiều vấn đề cần sửa chữa.
  - 11-20: Kém, xe có một số vấn đề cần chú ý.

- 21-30: Trung bình, xe có tình trạng tương đối ổn định nhưng có thể cần một số sửa chữa nhỏ.
- 31-40: Tốt, xe trong tình trạng tốt, ít vấn đề.
- 41-49: Rất tốt, xe trong tình trạng gần như mới.
- Dựa vào biểu đồ, ta có thể đưa ra một số đánh giá sau:
  - **Tình trạng phổ biến nhất:** Từ biểu đồ, ta thấy rằng có nhiều xe ở tình trạng tốt và rất tốt (điểm từ 31 đến 49). Điều này có thể dẫn đến giá trung bình trên thị trường xe cao hơn.
  - **Tình trạng ảnh hưởng đến giá xe:** Xe ở các điểm thấp (1-20) có thể kéo giá trung bình xuống, nhưng vì số lượng xe ở các mức này ít hơn so với các mức điểm cao hơn, nên ảnh hưởng của chúng đến giá chung có thể không quá lớn.
  - **Nhu cầu và cung cấp:** Số lượng xe trong tình trạng tốt và rất tốt cao cho thấy cung cấp xe chất lượng cao trên thị trường là đủ lớn, điều này có thể giúp ổn định giá ở mức cao. Ngược lại, số lượng xe trong tình trạng kém và rất kém thấp, có thể gây ra một số biến động về giá trong phân khúc xe giá rẻ.

- **Kết luận:**

**Tình** trạng xe ảnh hưởng mạnh mẽ đến giá bán của chúng. Xe ở tình trạng càng tốt sẽ có giá càng cao và dễ bán hơn. Biểu đồ cho thấy đa số các xe ở tình trạng trung bình đến rất tốt, điều này cho thấy thị trường xe có thể có xu hướng giá cao hơn, với một số ngoại lệ ở các xe tình trạng kém và rất kém có giá thấp hơn.

#### 4.1.2.2. Tạo biểu đồ phân tán giữa số km đã đi (odometer) và giá bán (selling price).

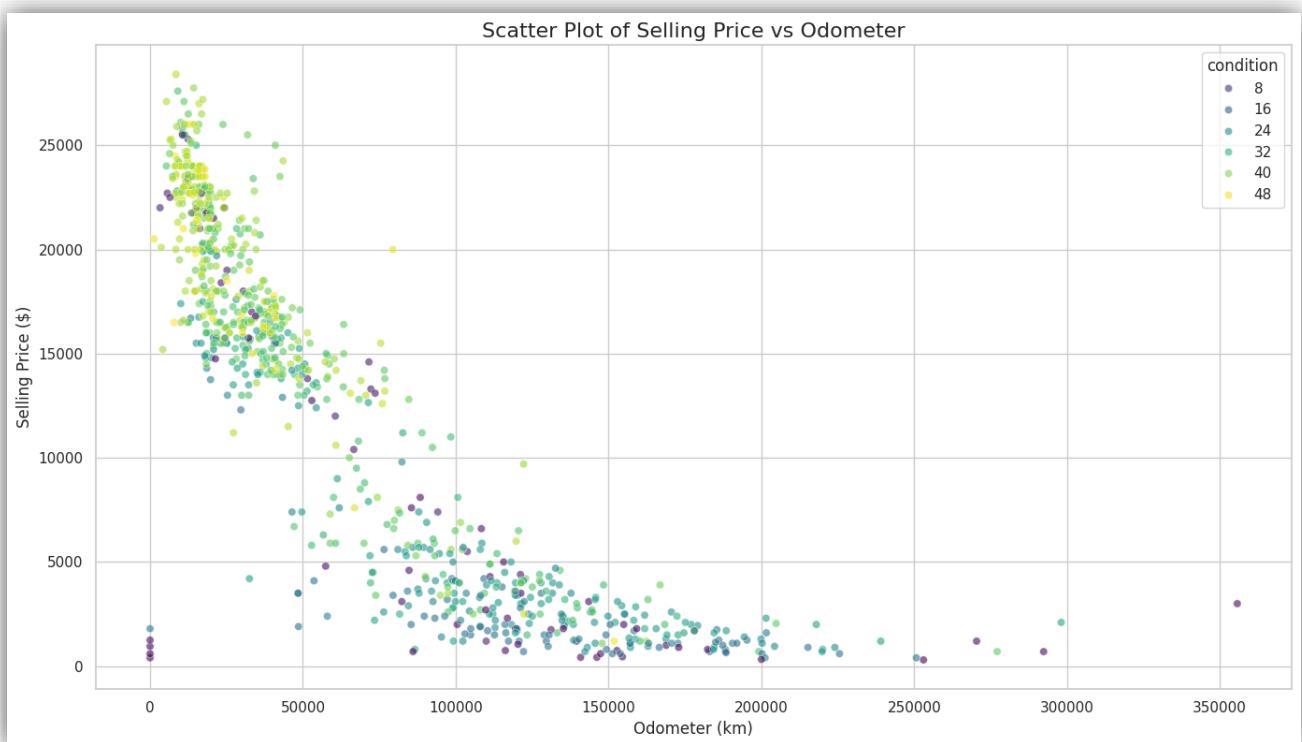
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(14, 8))
sns.set(style="whitegrid")

ax = sns.scatterplot(data=filtered_df, x="odometer", y="sellingprice", hue="condition", palette="viridis", alpha=0.6)

plt.xlabel("Odometer (km)", fontsize=12)
plt.ylabel("Selling Price ($)", fontsize=12)
plt.title("Scatter Plot of Selling Price vs Odometer", fontsize=16)

plt.tight_layout()
plt.show()
```



- Dựa vào biểu đồ, ta có thể đưa ra một số đánh giá như sau:
  - Xu hướng giảm giá bán theo số km đã đi: Rõ ràng và dễ nhận thấy, với giá bán giảm mạnh khi số km đã đi tăng lên.
  - Ảnh hưởng của tình trạng xe: Tình trạng tốt hơn giúp duy trì giá bán cao hơn, nhưng không thể ngăn chặn hoàn toàn sự giảm giá khi số km đã đi tăng.

#### 4.1.2.3. Tạo biểu đồ hiển thị giá bán trung bình theo năm sản xuất.

```

import matplotlib.pyplot as plt
import seaborn as sns

average_selling_price_by_year = filtered_df.groupby("year")["sellingprice"].mean().reset_index()

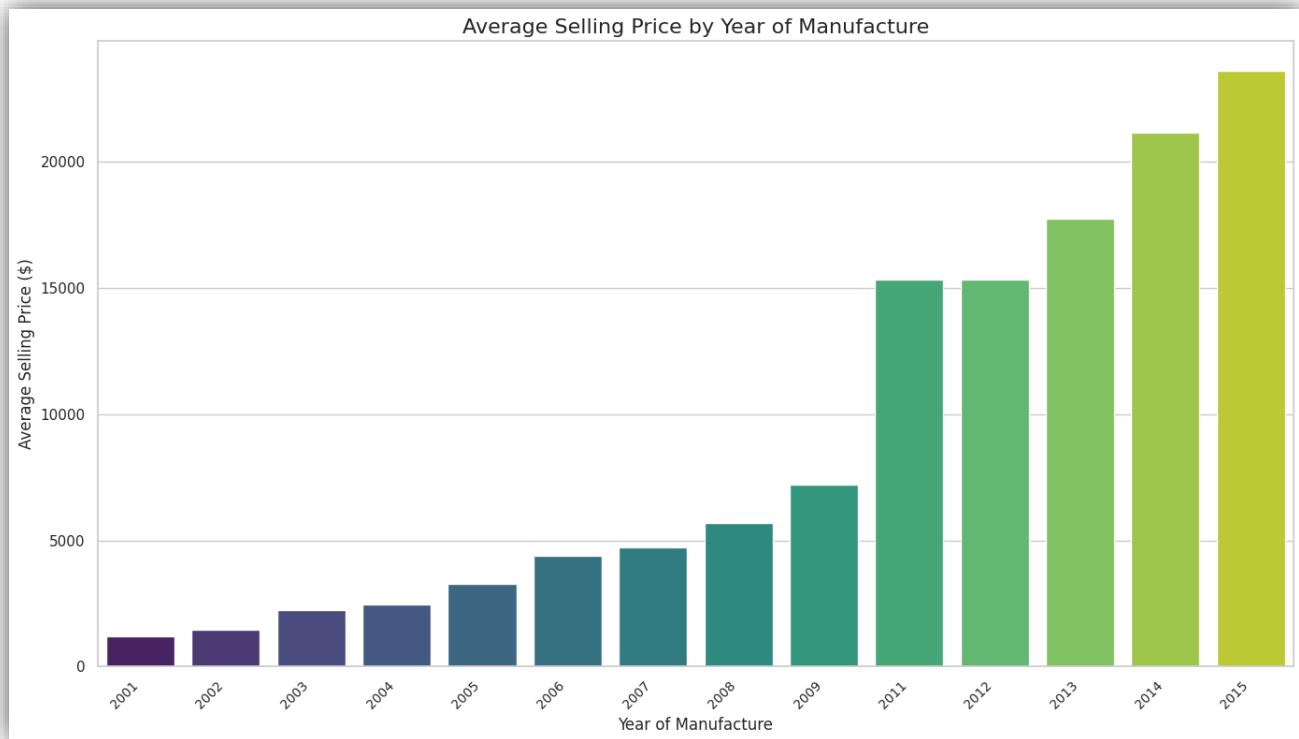
plt.figure(figsize=(14, 8))
sns.set(style="whitegrid")

ax = sns.barplot(data=average_selling_price_by_year, x="year", y="sellingprice", palette="viridis")

plt.xlabel("Year of Manufacture", fontsize=12)
plt.ylabel("Average Selling Price ($)", fontsize=12)
plt.title("Average Selling Price by Year of Manufacture", fontsize=16)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right", fontsize=10)

plt.tight_layout()
plt.show()

```



- Dựa vào biểu đồ, ta có thể đưa ra một số nhận xét như sau:  
Nhìn chung, biểu đồ thể hiện một bức tranh **tăng trưởng mạnh về giá trong các năm về sau**. Điều này có thể phản ánh nhiều yếu tố, bao gồm sự cải tiến trong công nghệ, tăng chi phí sản xuất, hoặc giá trị thị trường tăng lên của sản phẩm.

#### 4.2. Tiền xử lý dữ liệu

### 4.2.1. Kiểm tra mối tương quan giữa các thuộc tính

```

numeric_columns = ['year', 'condition', 'odometer', 'mmr', 'sellingprice']

missing_columns = [col for col in numeric_columns if col not in filtered_df.columns]
if missing_columns:
    print(f"The following columns are missing in the DataFrame: {missing_columns}")
else:
    correlation_matrix = filtered_df[numeric_columns].corr()

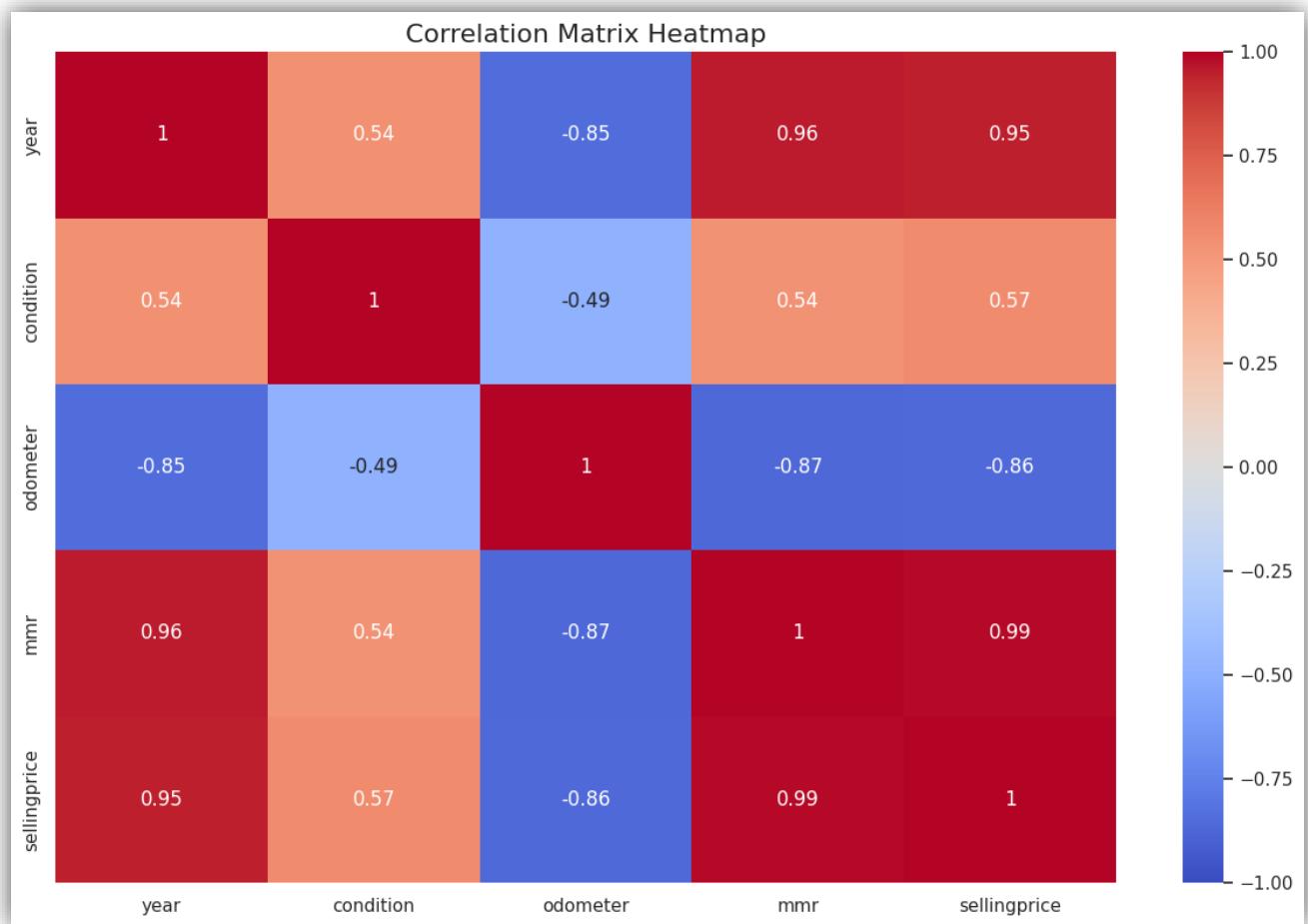
plt.figure(figsize=(12, 8))
sns.set(style="whitegrid")

ax = sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", vmin=-1, vmax=1, center=0)

plt.title("Correlation Matrix Heatmap", fontsize=16)

plt.tight_layout()
plt.show()

```



- Phân tích các mối tương quan:

- Year **và SellingPrice (0.95)**: Mối tương quan rất mạnh mẽ và tích cực. Điều này cho thấy năm sản xuất của xe càng mới, giá bán của xe càng cao. Đây là điều hợp lý vì xe mới thường có giá trị cao hơn do ít hao mòn và công nghệ mới.
- Year **và MMR (0.96)**: Mối tương quan rất mạnh mẽ và tích cực. MMR (Manheim Market Report) cũng là một yếu tố đánh giá giá trị xe. Xe sản xuất năm càng mới thì MMR càng cao, phản ánh giá trị xe trên thị trường cao hơn.
- Condition **và SellingPrice (0.57)**: Mối tương quan tích cực và khá mạnh. Tình trạng xe càng tốt thì giá bán càng cao. Điều này cũng hợp lý vì xe ở tình trạng tốt ít cần sửa chữa và bảo dưỡng, do đó có giá trị cao hơn.
- Odometer **và SellingPrice (-0.86)**: Mối tương quan rất mạnh mẽ và tiêu cực. Số km đi được càng nhiều thì giá bán xe càng thấp. Xe đi nhiều thường bị hao mòn nhiều hơn, do đó có giá trị thấp hơn.
- MMR **và SellingPrice (0.99)**: Mối tương quan cực kỳ mạnh mẽ và tích cực. Giá trị MMR gần như trực tiếp ảnh hưởng đến giá bán xe. Điều này cho thấy MMR là một chỉ số quan trọng trong việc định giá xe.
- Year **và Odometer (-0.85)**: Mối tương quan rất mạnh mẽ và tiêu cực. Xe càng mới thì số km đi được thường ít hơn, và ngược lại xe cũ thường đã đi được nhiều km hơn.

#### 4.2.2. Lựa chọn thuộc tính.

##### - **Loại bỏ các thuộc tính không cần thiết:**

Dựa vào ma trận tương quan, chúng ta có thể thấy một số thuộc tính có mối tương quan rất cao với nhau. Điều này có nghĩa là chúng có thể cung cấp thông tin tương tự và có thể được loại bỏ.

##### - **Tiến hành loại bỏ thuộc tính MMR và Year.** Giữ lại thuộc tính Condition và Odometer.

```
df_numeric = filtered_df[numERIC_columns]
x_data = df_numeric.drop(columns=['mmr', 'year'])
x_data.head()
```

	condition	odometer	sellingprice
3	41.0	14282.0	27750.0
558	42.0	19942.0	20000.0
632	37.0	27368.0	19000.0
1470	29.0	32782.0	15700.0
1480	41.0	60252.0	14750.0

#### 4.2.3. Thêm thuộc tính

Chuyển đổi cột saledate thành các cột day, month, year và đưa vào mô hình

```

x_data['saledate'] = pd.to_datetime(x_data['saledate'])

x_data['saleday'] = x_data['saledate'].dt.day
x_data['salemmonth'] = x_data['saledate'].dt.month
x_data['saleyear'] = x_data['saledate'].dt.year

x_data.drop('saledate', axis=1, inplace=True)

x_data.head(10)

```

	condition	odometer	sellingprice	saleday	salemmonth	saleyear	grid icon
218744		35.0	9088.0	27600.0	11	2	2015
178461		2.0	120373.0	1050.0	2	2	2015
176132		19.0	99851.0	4100.0	3	2	2015
33805		39.0	31904.0	25500.0	6	1	2015
413502		33.0	36034.0	14100.0	15	6	2015
181180		44.0	18069.0	22200.0	3	2	2015
274298		27.0	111030.0	2500.0	25	2	2015
308437		32.0	178058.0	1700.0	4	3	2015
378562		19.0	89706.0	2400.0	28	5	2015
396380		43.0	33124.0	16500.0	4	6	2015

#### 4.2.4. Xóa trùng lặp

Kiểm tra xem có trùng lặp trong tập dữ liệu hay không.

```

print("Number of rows: ", len(x_data.index))
x_data.drop_duplicates(inplace=True)
print("Number of rows after drop duplicates: ", len(x_data.index))

```

Number of rows: 886  
Number of rows after drop duplicates: 886

### 4.3. Ứng dụng mô hình thuật toán khai thác dữ liệu

#### 4.3.1. Chia dữ liệu trước khi xây dựng mô hình thuật toán.

Sau khi tiền xử lý dữ liệu, chúng ta thu được DataFrame cuối cùng X với các thuộc tính phân loại được mã hóa.

DataFrame X được chia thành các tập lần lượt Train-Validate-Test

**Bước 1:** Chia DataFrame X thành hai phần: tập huấn luyện/đánh giá (X) và tập kiểm tra (X\_test).

```
from sklearn.model_selection import train_test_split

# Train/Validation - Test split
x_data, x_data_test = train_test_split(x_data, test_size=0.2, random_state=42)
print(x_data.shape, x_data_test.shape)

(708, 6) (178, 6)
```

- Sử dụng hàm train\_test\_split() từ thư viện scikit-learn để thực hiện việc chia. Thiết lập tham số test\_size bằng 0.2, có nghĩa là 20% dữ liệu sẽ được dùng cho tập kiểm tra và 80% còn lại sẽ được sử dụng cho huấn luyện và đánh giá.
- Thiết lập tham số random\_state bằng 42 để đảm bảo việc chia là có thể tái lập. In ra kích thước của hai tập kết quả sử dụng thuộc tính shape của mỗi DataFrame, trả về một tuple chứa số hàng và số cột (theo thứ tự đó).

#### Bước 2:

Tạo các biến “sample”, “y\_sample”, và “x\_sample”:

```
sample = x_data
y_sample = sample["sellingprice"]
x_sample = sample.drop("sellingprice", axis=1)
```

- “sample”: Chứa dữ liệu từ “x\_data”. “y\_sample”: Chứa cột mục tiêu “sellingprice”.
- “x\_sample”: Chứa các đặc trưng (features) ngoại trừ “sellingprice”.

- Hàm “preprocess\_input” để tiền xử lý dữ liệu

```
def preprocess_input(df,scaler):
    y=y_sample
    x=x_sample

    #Chia dữ liệu thành tập huấn luyện và tập kiểm tra (90% huấn luyện, 10% kiểm tra)
    x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.9)

    #Huấn luyện bộ chuẩn hóa trên tập huấn luyện bằng scaler
    scaler.fit(x_train)

    #Biến đổi tập huấn luyện và tập kiểm tra theo bộ chuẩn hóa đã được huấn luyện,
    #Chuyển kết quả về DataFrame và gán lại các tên cột
    x_train=pd.DataFrame(scaler.transform(x_train),columns=x_train.columns)
    x_test=pd.DataFrame(scaler.transform(x_test),columns=x_test.columns)

    #Trả về các tập dữ liệu đã được chuẩn hóa
    return x_train,x_test,y_train,y_test
```

Quy trình này giúp đảm bảo rằng dữ liệu được chuẩn bị và sẵn sàng cho việc huấn luyện mô hình học máy, với các giá trị đặc trưng được chuẩn hóa để cải thiện hiệu suất của mô hình.

**Bước 3:** Sau khi tiền xử lý xong, việc in ra kích thước của các tập dữ liệu giúp bạn kiểm tra xem quá trình chia và chuẩn hóa dữ liệu đã được thực hiện đúng hay không.

```
x_train,x_test,y_train,y_test=preprocess_input(df,MinMaxScaler())
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

(637, 5)
(71, 5)
(637,)
(71,)
```

#### 4.3.2. Thực hiện xây dựng mô hình Decision Tree và Random Forest

**Bước 1: Khởi tạo từ điển “model” chứa các mô hình hồi quy.**

```
models = {
    'Random Forest Regressor': RandomForestRegressor(random_state=42),
    'Decision Tree Regressor': tree.DecisionTreeRegressor(random_state=42)
}
```

### Bước 2: Huấn luyện mô hình và vẽ cây quyết định:

Vòng lặp qua các mô hình trong từ điển models, huấn luyện chúng và vẽ cây quyết định nếu mô hình là Decision Tree hoặc Random Forest.

```
# Train and plot the trees
for name, model in models.items():
    model.fit(x_train, y_train)

    # Limit depth for visualization
    max_depth = 3

    if name == 'Random Forest Regressor':
        fn = x_train.columns
        fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(20, 10), dpi=800)
        plot_tree(model.estimators_[0], feature_names=fn, filled=True, max_depth=max_depth)
        fig.savefig('rf_individualtree_pruned.png')

    if name == 'Decision Tree Regressor':
        fn = x_train.columns
        fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(20, 10), dpi=800)
        plot_tree(model, feature_names=fn, filled=True, max_depth=max_depth)
        fig.savefig('dct_individualtree_pruned.png')
```

### Bước 3: Đánh giá hiệu suất mô hình

```
print(name, "R-squared on test data:", model.score(x_test, y_test))

y_pred = model.predict(x_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"{name} Mean Squared Error: {mse}")
print(f"{name} R-squared: {r2}")
```

#### 4.3.3. Kết quả sau khi thực hiện

##### 4.3.3.1.

```

Random Forest Regressor R-squared on test data: 0.8984115260860758
Random Forest Regressor Mean Squared Error: 7391035.966549296
Random Forest Regressor R-squared: 0.8984115260860758
Decision Tree Regressor R-squared on test data: 0.8570208198471413
Decision Tree Regressor Mean Squared Error: 10402403.169014085
Decision Tree Regressor R-squared: 0.8570208198471413

```

- Đánh giá:

- Random Forest:

**Hiệu suất:** R-squared ( $R^2$ ) là một chỉ số đo lường mức độ phù hợp của mô hình với dữ liệu. Với giá trị  $R^2$  xấp xỉ 0.898, mô hình Random Forest Regressor cho thấy khả năng giải thích gần 90% sự biến thiên trong dữ liệu kiểm tra. Đây là một hiệu suất khá cao, cho thấy mô hình có thể dự đoán giá bán của xe với độ chính xác tốt.

**Sai số:** Mean Squared Error (MSE) là chỉ số đo lường sai số trung bình của các dự đoán so với giá trị thực tế. Với MSE khoảng 7.39 triệu, mô hình vẫn có một mức độ sai số nhất định, nhưng vẫn thấp hơn so với Decision Tree Regressor.

- Decision Tree:

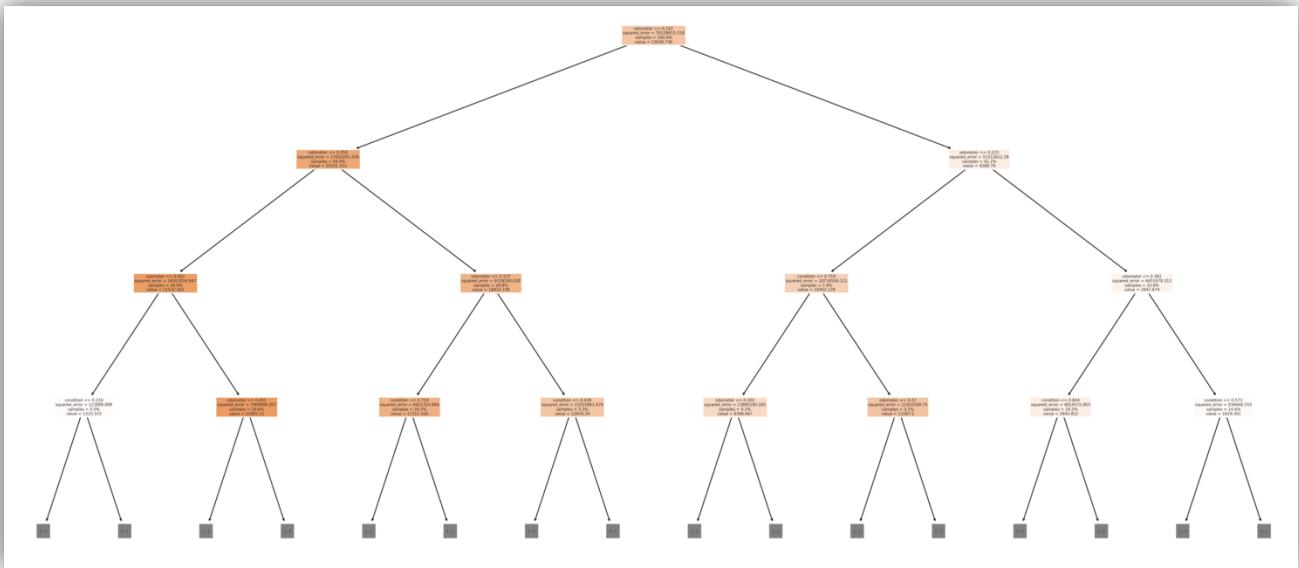
**Hiệu suất:** Với  $R^2$  khoảng 0.857, mô hình Decision Tree Regressor cũng cho thấy khả năng giải thích tốt, nhưng thấp hơn so với Random Forest Regressor. Điều này có nghĩa là Decision Tree Regressor không hiệu quả bằng Random Forest Regressor trong việc dự đoán giá bán xe.

**Sai số:** MSE của Decision Tree Regressor cao hơn, khoảng 10.4 triệu, cho thấy mức độ sai số của mô hình này lớn hơn. Điều này có thể do mô hình đơn cây quyết định dễ bị overfitting (quá khớp) với dữ liệu huấn luyện.

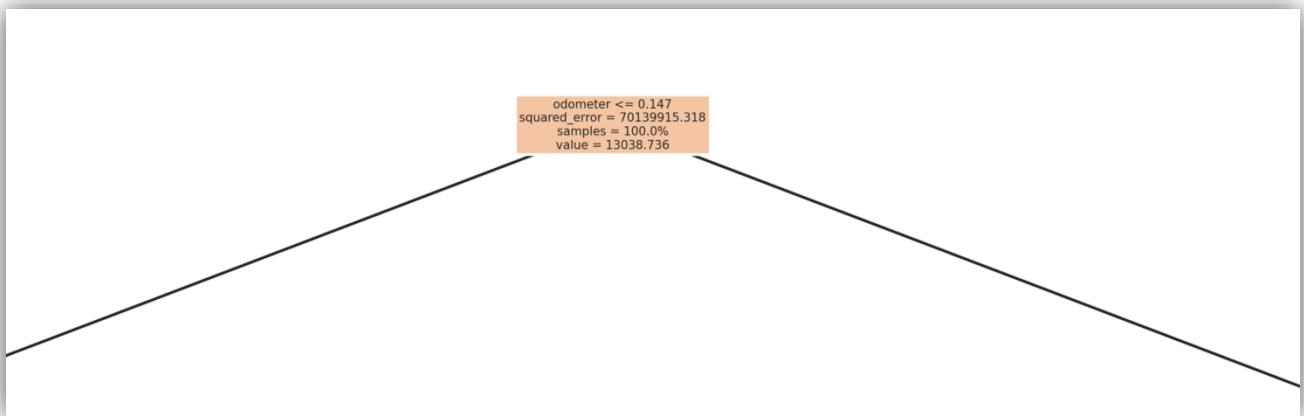
- **Kết luận:**

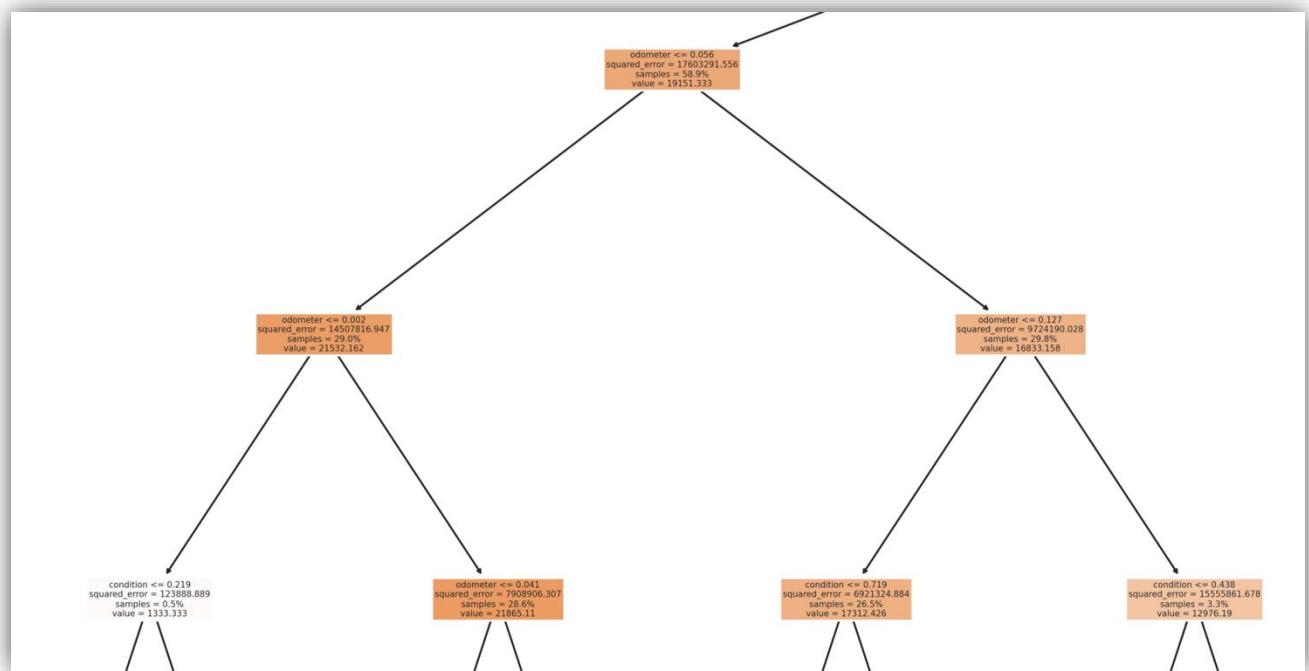
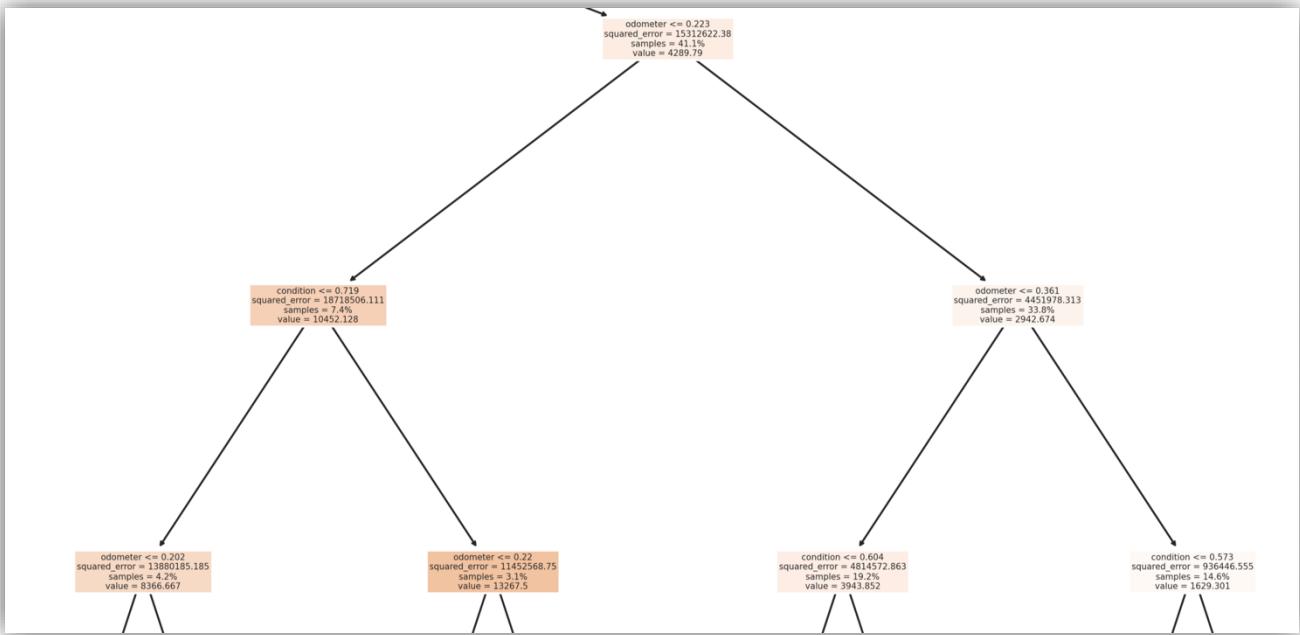
Random Forest Regressor cho kết quả tốt hơn với  $R^2$  cao hơn và MSE thấp hơn so với Decision Tree Regressor. Điều này là do Random Forest sử dụng nhiều cây quyết định và trung bình các dự đoán của chúng, giúp giảm overfitting và tăng độ chính xác của dự đoán.

#### 4.3.3.2. Cây quyết định Decision Tree

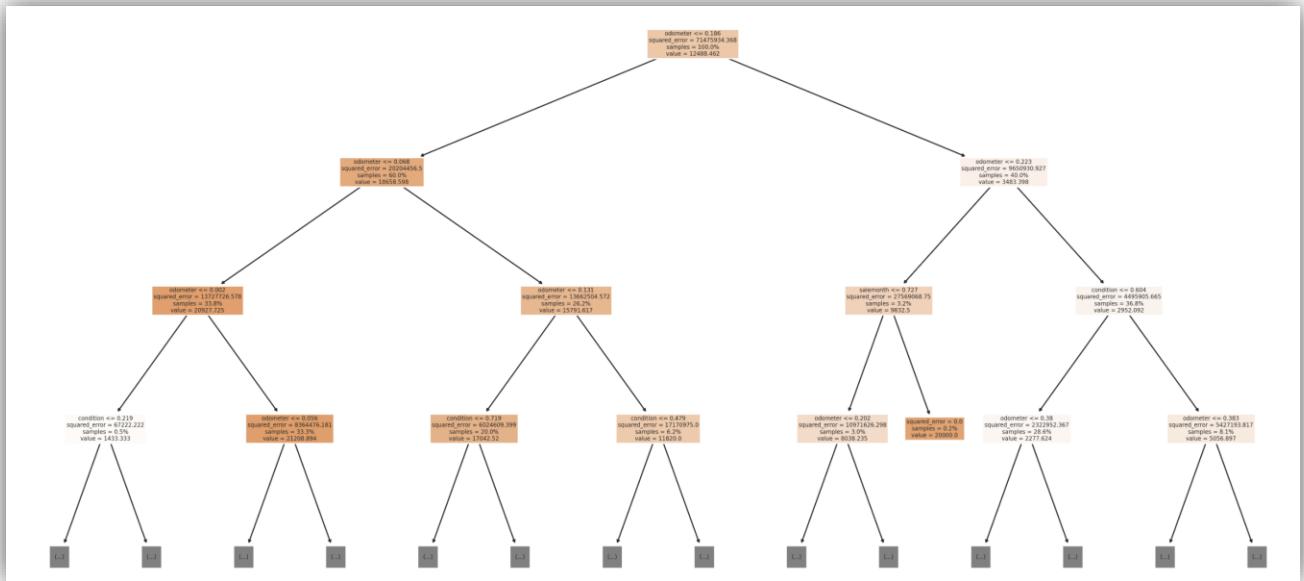


Phóng to:

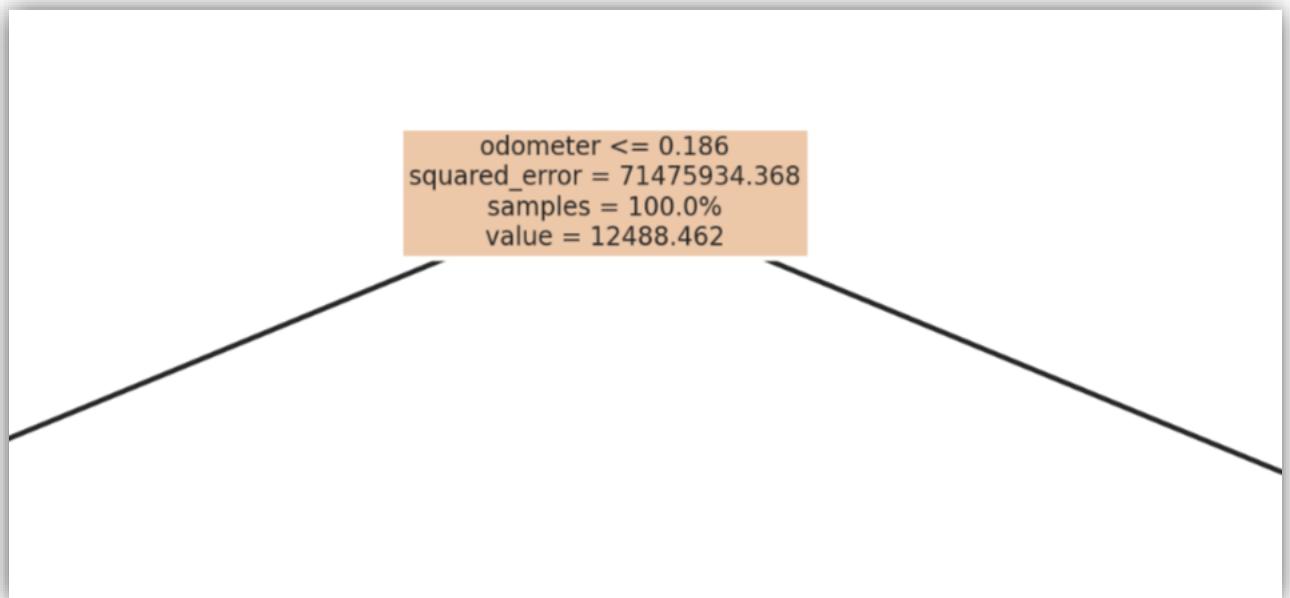


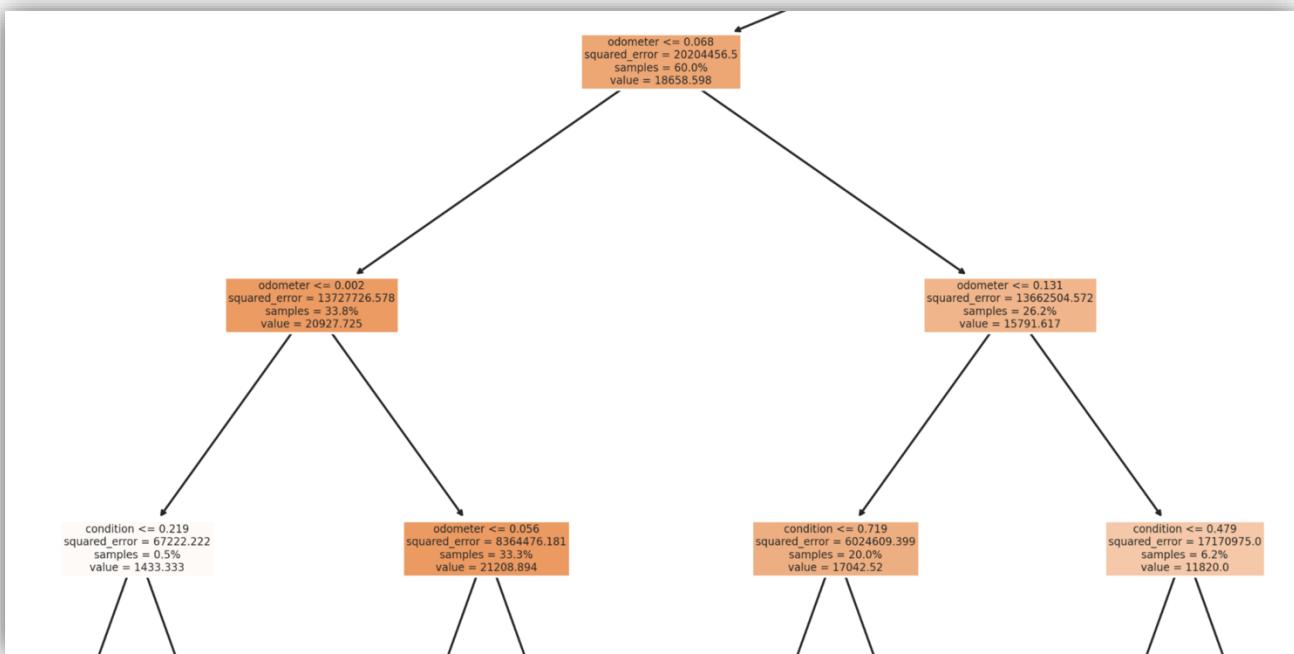
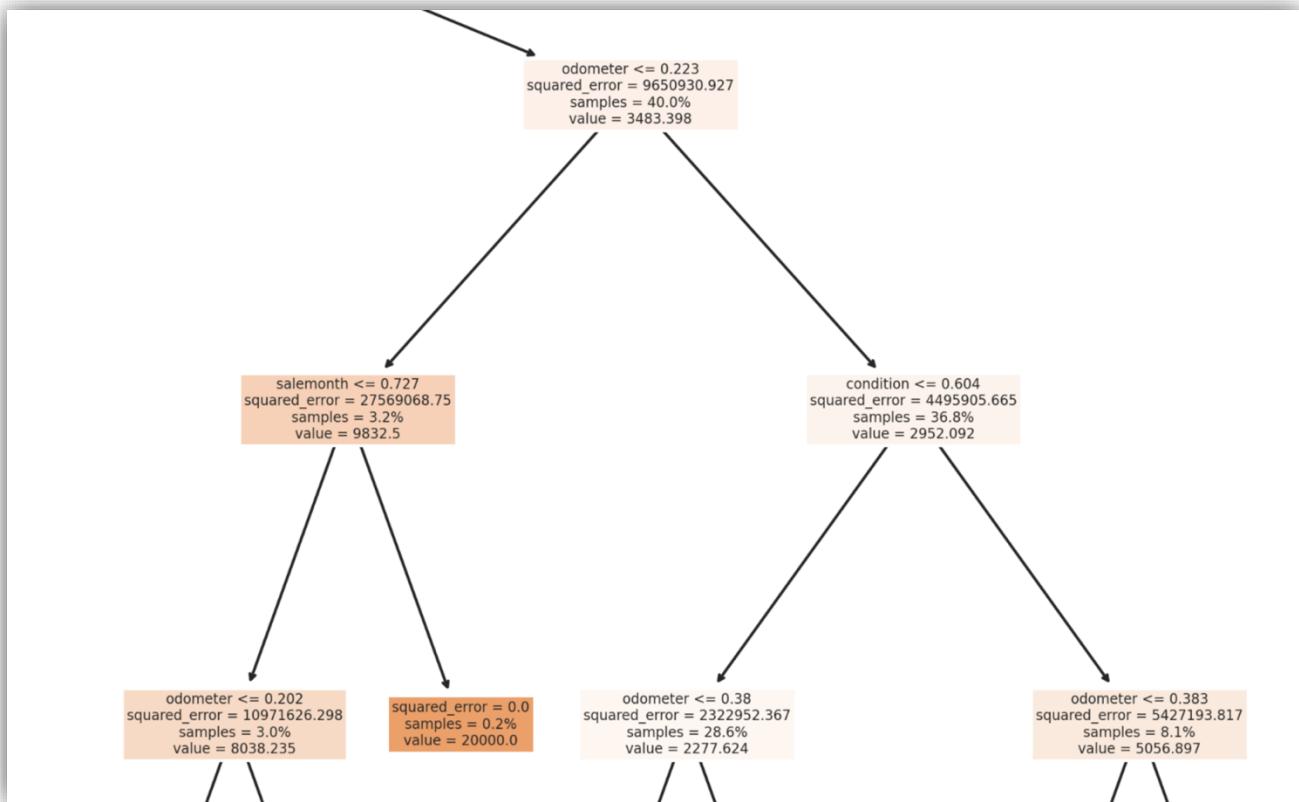


### 4.3.3.3. Cây quyết định Random Forest



Phóng to:



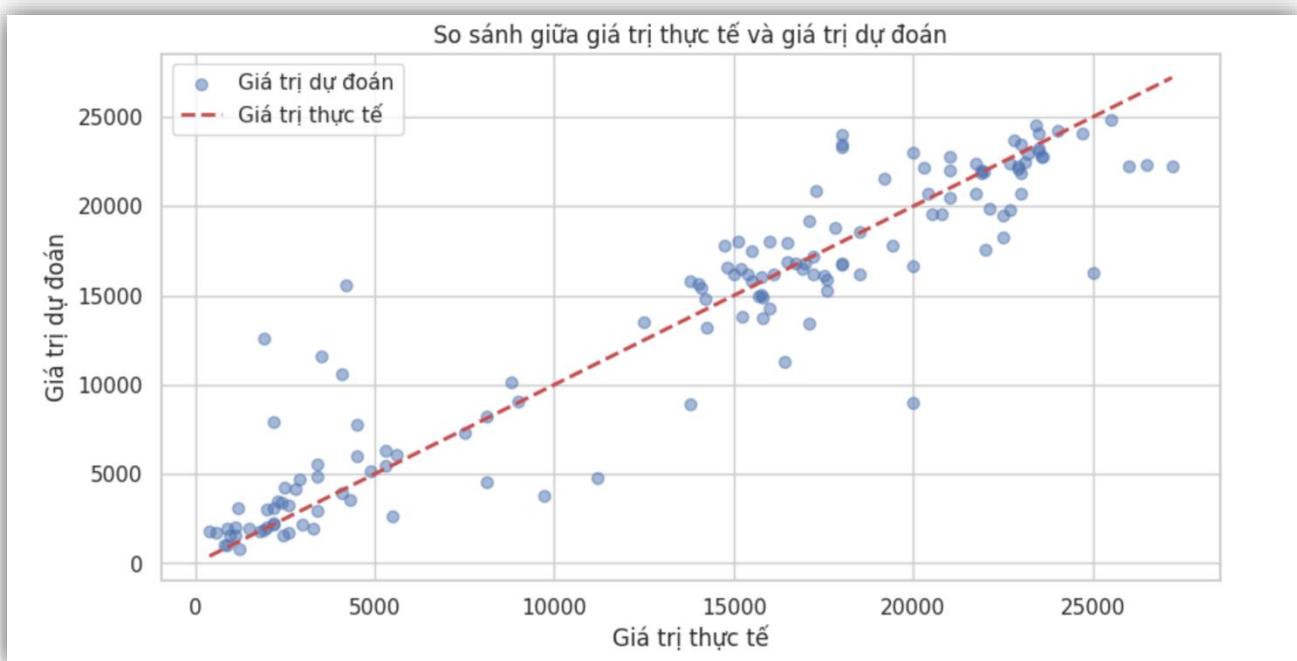


- **Giải thích:** Kết quả của cây quyết định từ mô hình Random Forest Regressor có một số liệu tại nút ban đầu (nút gốc) với các thông tin như sau:
    - odometer  $\leq 0.201$ : Đây là điều kiện đầu tiên mà cây quyết định kiểm tra để phân

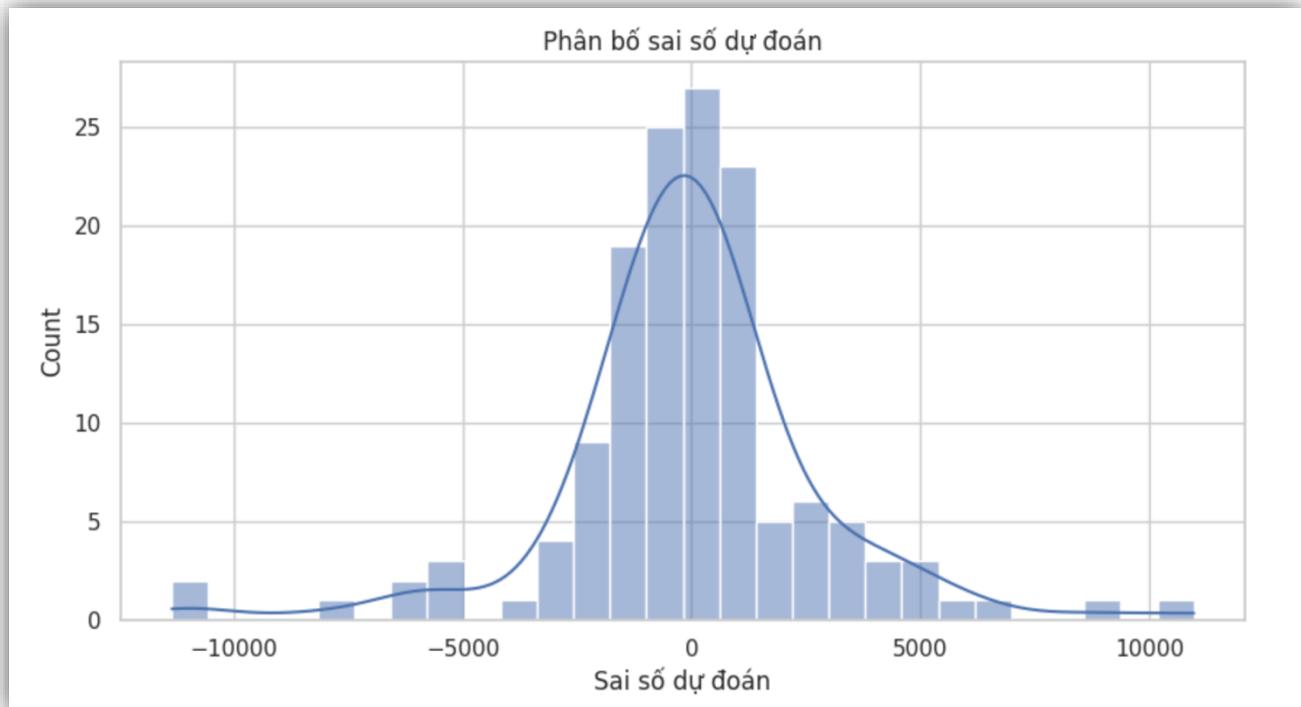
chia dữ liệu. Nếu giá trị odometer (số km đã đi) nhỏ hơn hoặc bằng 0.201 (giá trị này có thể đã được chuẩn hóa hoặc chia tỷ lệ), dữ liệu sẽ đi theo nhánh bên trái của cây. Ngược lại, nếu giá trị lớn hơn 0.201, dữ liệu sẽ đi theo nhánh bên phải.

- squared\_error = 68648947.428: Đây là tổng sai số bình phương (squared error) tại nút gốc. Nó biểu thị tổng độ lệch bình phương giữa giá trị dự đoán và giá trị thực tế của tất cả các mẫu dữ liệu tại nút này. Sai số này càng thấp thì mô hình càng tốt.
- samples = 100%: Đây là tỷ lệ mẫu dữ liệu tại nút gốc, ở đây là 100% vì tất cả dữ liệu huấn luyện đều được sử dụng để xây dựng cây quyết định.
- value = 13381.554: Đây là giá trị trung bình dự đoán của biến mục tiêu (giá bán) tại nút gốc. Nó biểu thị giá trị trung bình của tất cả các giá trị mục tiêu của mẫu dữ liệu tại nút này.

#### 4.3.4. Sử dụng mô hình Random Forest và vẽ các đồ thị liên quan



- Đánh giá: Đường chấm màu đỏ đại diện cho giá trị thực tế, và các điểm màu xanh đại diện cho giá trị dự đoán. Các điểm dữ liệu nằm gần đường chấm đỏ cho thấy mô hình dự đoán rất sát với giá trị thực tế. Tuy nhiên, có một số điểm dữ liệu xa đường chấm đỏ, chỉ ra một số lỗi dự đoán.



- Đánh giá: Biểu đồ này cho thấy phân phối của sai số dự đoán (giá trị thực tế trừ giá trị dự đoán). Phần lớn các sai số nằm gần giá trị 0, cho thấy mô hình dự đoán khá chính xác. Tuy nhiên, có một số sai số lớn, điều này có thể chỉ ra các ngoại lệ hoặc các trường hợp mà mô hình không dự đoán tốt.

#### 4.3.5. Sử dụng mô hình Random Forest để đưa ra dự đoán “sellingprice” dựa trên các thuộc tính “condition”, “odometer”, “saleday”, “salemonth”, “saleyear”.

```

# Hàm để dự đoán giá bán mới
def predict_selling_price(condition, odometer, saleday, salemonth, saleyear):
    # Tạo DataFrame mới từ các giá trị đầu vào
    input_data = pd.DataFrame({
        'condition': [condition],
        'odometer': [odometer],
        'saleday': [saleday],
        'salemmonth': [salemmonth],
        'saleyear': [saleyear]
    })

    # Dự đoán giá bán sử dụng mô hình Random Forest đã huấn luyện
    predicted_price = rf.predict(input_data)
    return predicted_price[0]

# Ví dụ dự đoán giá bán mới
condition = 4.0
odometer = 50000
saleday = 15
salemmonth = 5
saleyear = 2024

predicted_price = predict_selling_price(condition, odometer, saleday, salemonth, saleyear)
print(f"Dự đoán giá bán: ${predicted_price:.2f}")

```

- Kết quả dự đoán: \$12131.50

#### 4.3.6. So sánh kết quả dự đoán với kết quả thực tế trong dataset.

- Dự đoán

```

def predict_selling_price(condition, odometer, saleday, salemonth, saleyear):
    # Tạo DataFrame mới từ các giá trị đầu vào
    input_data = pd.DataFrame({
        'condition': [condition],
        'odometer': [odometer],
        'saleday': [saleday],
        'salemmonth': [salemmonth],
        'saleyear': [saleyear]
    })

    # Dự đoán giá bán sử dụng mô hình Random Forest đã huấn luyện
    predicted_price = rf.predict(input_data)
    return predicted_price[0]

# Ví dụ dự đoán giá bán mới
condition = 35.0
odometer = 9088.0
saleday = 11
salemmonth = 2
saleyear = 2015

predicted_price = predict_selling_price(condition, odometer, saleday, salemonth, saleyear)
print(f"Dự đoán giá bán: ${predicted_price:.2f}")

```

Dự đoán giá bán: \$26502.00

- Thực tế:

	condition	odometer	sellingprice	saleday	salemmonth	saleyear
218744	35.0	9088.0	27600.0	11	2	2015

**Kết luận:** Giá bán dự đoán được là 26502.00 gần đúng so với giá trị thực tế là 27600.0

#### 4.3.7. Tập luận dành cho người dùng cuối.

Dựa vào cây quyết định có được từ việc huấn luyện mô hình thuật toán random forest ta có 1 số tập luận sau:

- Nếu lần lượt odometer  $\leq 0.002$ , condition  $\leq 0.219$  thì giá trị sellingprice sẽ rơi vào khoảng 866.667
- Nếu  $0.002 \leq \text{odometer} \leq 0.045$  thì giá trị sellingprice sẽ rơi vào khoảng 22115.946
- Nếu  $0.045 \leq \text{odometer} \leq 0.083$  thì giá trị sellingprice sẽ rơi vào khoảng

17163.143

- Nếu odometer  $\leq 0.134$  và condition  $\leq 0.49$  thì giá trị sellingprice sẽ rơi vào khoảng 12343.137

## **DANH MỤC TÀI LIỆU THAM KHẢO**

<https://drive.google.com/file/d/1ob7Qdtf0OTCZFzuvHLj4MVhT8PGU44Lo/view>

[https://drive.google.com/file/d/1SFQZXLDmsU0q5d\\_r8qTavVkh8HMhkTjH/view?usp=sharing](https://drive.google.com/file/d/1SFQZXLDmsU0q5d_r8qTavVkh8HMhkTjH/view?usp=sharing)

<https://learn.microsoft.com/en-us/power-bi/visuals/power-bi-report-visualizations>

