

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

----



**BÁO CÁO ĐỒ ÁN**  
**Môn học: Kho dữ liệu và OLAP**

**ĐỀ TÀI: PHÂN TÍCH VÀ XÂY DỰNG KHO DỮ  
LIỆU VỀ DỮ LIỆU BÁN HÀNG Ở SIÊU THỊ  
TRÊN TOÀN CẦU**

**Sinh viên thực hiện:**

Dương Văn Nhật Long	20521561
Võ Đoàn Tố Loan	20521544

*Giảng viên hướng dẫn: ThS. Nguyễn Thị Kim Phụng*

*Thành phố Hồ Chí Minh tháng 06 năm 2024*

## LỜI MỞ ĐẦU

Trong bối cảnh của sự phát triển không ngừng của công nghệ thông tin và dữ liệu lớn, việc hiểu và áp dụng các khái niệm và công nghệ trong lĩnh vực Kho dữ liệu và OLAP (Online Analytical Processing) trở nên vô cùng quan trọng. Đồ án này được thực hiện nhằm mục đích nghiên cứu và triển khai một hệ thống kho dữ liệu và OLAP có khả năng xử lý, phân tích và trực quan hóa dữ liệu một cách hiệu quả.

Trên nền tảng của những kiến thức được học trong môn học, chúng em đã tiến hành nghiên cứu sâu hơn về các phương pháp thiết kế, triển khai và tối ưu hóa hệ thống kho dữ liệu. Đồ án này đưa ra một quy trình cụ thể từ việc thu thập dữ liệu, chuẩn hóa và lưu trữ, cho đến việc xây dựng các cube OLAP và phát triển các báo cáo, trực quan hóa dữ liệu để hỗ trợ quá trình ra quyết định.

Qua việc thực hiện đồ án, chúng em đã đạt được một số mục tiêu quan trọng như: Hiểu rõ hơn về cách tổ chức và quản lý dữ liệu trong một môi trường kho dữ liệu. Nắm vững các nguyên lý và phương pháp OLAP để phân tích dữ liệu đa chiều. Áp dụng kiến thức lý thuyết vào thực tiễn thông qua việc xây dựng một hệ thống kho dữ liệu và OLAP thực tế. Phát triển kỹ năng làm việc nhóm, quản lý dự án và giải quyết vấn đề.

Qua quá trình thực hiện đồ án, chúng em hi vọng rằng kết quả đạt được sẽ đóng góp vào việc nâng cao hiệu quả quản lý và ra quyết định trong lĩnh vực này. Đồng thời, kinh nghiệm và kiến thức tích lũy từ đồ án sẽ là nền tảng cho sự phát triển và tiếp tục nghiên cứu trong tương lai.

## LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới cô Nguyễn Thị Kim Phụng – giảng viên hướng dẫn môn học “Kho dữ liệu và OLAP” tại trường Đại học Công nghệ Thông tin. Cô đã cung cấp cho chúng em cơ hội để có được những kỹ năng và kiến thức cần thiết để hoàn thành chủ đề đồ án này.

Trong suốt quá trình học tập và trong quá trình thực hiện đề tài, nhóm chúng em đã vận dụng và tích lũy những kiến thức quý báu từ các bài giảng đồng thời kết hợp những kiến thức mới thông qua nghiên cứu. Tuy nhiên, do hiểu biết về lý luận và kinh nghiệm thực tế còn hạn chế nên trong đồ án của chúng em không tránh khỏi những sai sót và thiếu sót. Chúng em rất mong nhận được những phản hồi và góp ý từ cô, điều này sẽ giúp chúng em có thêm kinh nghiệm và hoàn thiện hơn trong tương lai, không chỉ trong giai đoạn tới mà còn trong các dự án sau này.

Xin chân thành cảm ơn !

**Nhóm sinh viên thực hiện:**

Dương Văn Nhật Long

Võ Đoàn Tố Loan

## NHẬN XÉT CỦA GIẢNG VIÊN

....., ngày ....., tháng ....., năm 2024

## Người nhận xét

(Ký tên và ghi rõ họ tên)

## MỤC LỤC

<b>LỜI MỞ ĐẦU .....</b>	<b>2</b>
<b>LỜI CẢM ƠN .....</b>	<b>3</b>
<b>NHẬN XÉT CỦA GIẢNG VIÊN .....</b>	<b>4</b>
<b>MỤC LỤC .....</b>	<b>5</b>
<b>DANH MỤC BẢNG, HÌNH ẢNH.....</b>	<b>9</b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....</b>	<b>16</b>
1.1    Lý do chọn đề tài.....	16
1.2    Mô tả bài toán .....	16
<b>CHƯƠNG 2: GIỚI THIỆU KHO DỮ LIỆU .....</b>	<b>17</b>
2.1    Giới thiệu bộ dữ liệu .....	17
2.1.1    Thông tin nguồn dữ liệu .....	17
2.1.2    Thông tin chi tiết .....	17
2.1.3    Mô tả thuộc tính .....	18
2.1.4    Mô tả chi tiết các thuộc tính danh mục .....	19
2.2    Xây dựng kho dữ liệu .....	25
2.2.1    Lược đồ hình sao (Star Schema) .....	25
2.2.2    Mô tả chi tiết bảng FACT và các bảng DIM .....	25
<b>CHƯƠNG 3: QUÁ TRÌNH XÂY DỰNG KHO DỮ LIỆU (SSIS).....</b>	<b>30</b>
3.1    Chuẩn bị các công cụ .....	30
3.2    Chuẩn bị cơ sở dữ liệu.....	32
3.3    Tạo mới project SSIS.....	36
3.4    Cấu hình và thực hiện quá trình SSIS .....	39
3.4.1    Mô hình thực hiện quá trình SSIS .....	39
3.4.2    Load Dimension Tables.....	40

3.4.3	Load Fact Table .....	93
3.4.4	Execute SQL Task .....	102
<b>3.5</b>	<b>Chạy Project SSIS.....</b>	<b>106</b>
<b>CHƯƠNG 4:</b>	<b>QUÁ TRÌNH PHÂN TÍCH DỮ LIỆU (SSAS).....</b>	<b>108</b>
<b>4.1</b>	<b>Thực hiện quá trình SSAS .....</b>	<b>108</b>
4.1.1	Chuẩn bị các công cụ.....	108
4.1.2	Tạo mới project SSAS .....	108
4.1.3	Cấu hình và thực hiện quá trình SSAS.....	110
4.1.4	Chạy project SSAS.....	138
<b>4.2</b>	<b>Thực thi 15 câu truy vấn trên Visual Studio (Thao tác bằng tay).....</b>	<b>139</b>
4.2.1	Roll Up Queries.....	139
4.2.2	Drill Down Queries .....	140
4.2.3	Slice and Dice Queries .....	142
4.2.4	Pivot.....	145
<b>4.3</b>	<b>Thực thi 15 câu truy vấn bằng ngôn ngữ MDX.....</b>	<b>146</b>
4.3.1	Roll Up Queries.....	146
4.3.2	Drill Down Queries .....	149
4.3.3	Slice and Dice Queries .....	150
4.3.4	Pivot.....	153
<b>4.4</b>	<b>Thực thi 15 câu truy vấn bằng Excel .....</b>	<b>155</b>
4.4.1	Thực hiện kết nối Microsoft Analysis Services .....	155
4.4.2	Thực hiện các truy vấn .....	157
<b>4.5</b>	<b>Thực thi 15 câu truy vấn bằng Power BI .....</b>	<b>167</b>
4.5.1	Thực hiện kết nối Microsoft Analysis Services .....	167
4.5.2	Thực hiện các truy vấn .....	168

<b>CHƯƠNG 5: QUÁ TRÌNH KHAI THÁC DỮ LIỆU (DATA MINING) .....</b>	<b>181</b>
5.1    Chủ đề khai thác .....	181
5.2    Import dữ liệu và thư viện cần thiết .....	181
5.3    Thực hiện EDA.....	182
5.4    Tiền xử lý dữ liệu .....	184
5.4.1    Tạo thuộc tính quyết định Profit_Positive.....	184
5.4.2    Xóa các thuộc tính không cần thiết .....	185
5.4.3    Chuyển đổi kiểu dữ liệu .....	185
5.4.4    Phân tích tương quan.....	186
5.4.5    Xử lý thuộc tính danh mục .....	188
5.4.6    Tìm 4 thuộc tính quan trọng .....	190
5.5    Huấn luyện mô hình .....	192
5.5.1    Giới thiệu phương pháp GridSearchCV .....	192
5.5.2    Thuật toán Random Forest .....	193
5.5.3    Thuật toán Logistic Regression.....	196
5.5.4    Thuật toán Naive Bayes .....	199
5.5.5    Thuật toán Perceptron .....	202
5.6    Đánh giá mô hình.....	205
5.6.1    Thuật toán Random Forest .....	205
5.6.2    Thuật toán Logistic Regression.....	206
5.6.3    Thuật toán Naïve Bayes .....	207
5.6.4    Thuật toán Perceptron .....	208
5.6.5    Đánh giá chung.....	208
5.7    Phân tích kết quả thuật toán.....	209
5.7.1    Thuật toán Random Forest .....	210

5.7.2	Thuật toán Logistic Regression .....	219
5.7.3	Thuật toán Naïve Bayes .....	227
5.7.4	Thuật toán Perceptron .....	234
<b>5.8</b>	<b>Deploy mô hình và thử nghiệm.....</b>	<b>242</b>
<b>NGUỒN THAM KHẢO.....</b>		<b>247</b>

## **DANH MỤC BẢNG, HÌNH ẢNH**

### **1. Bảng**

Bảng 2.1: Mô tả các thuộc tính của bộ dữ liệu Global Superstore .....	19
Bảng 2.2: Mô tả chi tiết giá trị của thuộc tính Category .....	20
Bảng 2.3: Mô tả chi tiết giá trị của thuộc tính Sub_Category .....	21
Bảng 2.4: Mô tả chi tiết giá trị của thuộc tính Market .....	21
Bảng 2.5: Mô tả chi tiết giá trị của thuộc tính Order_Priority .....	21
Bảng 2.6: Mô tả chi tiết giá trị của thuộc tính Ship_Mode .....	22
Bảng 2.7: Mô tả chi tiết giá trị của thuộc tính Segment .....	22
Bảng 2.8: Mô tả chi tiết giá trị của thuộc tính City .....	23
Bảng 2.9: Mô tả chi tiết giá trị của thuộc tính Segment .....	23
Bảng 2.10: Mô tả chi tiết giá trị của thuộc tính Country .....	24
Bảng 2.11: Mô tả chi tiết giá trị của thuộc tính Region .....	24
Bảng 2.12: Lược đồ hình sao của kho dữ liệu .....	25
Bảng 2.13: Mô tả chi tiết bảng Fact .....	26
Bảng 2.14: Mô tả chi tiết bảng Dim_Customer .....	27
Bảng 2.15: Mô tả chi tiết bảng Dim_Product .....	27
Bảng 2.16: Mô tả chi tiết bảng Dim_Location .....	28
Bảng 2.17: Mô tả chi tiết bảng Dim_Market .....	28
Bảng 2.18: Mô tả chi tiết bảng Dim_OrderDate .....	28
Bảng 2.19: Mô tả chi tiết bảng Dim_ShipDate .....	28
Bảng 2.20: Mô tả chi tiết bảng Dim_OderPriority .....	29
Bảng 2.21: Mô tả chi tiết bảng Dim_ShipMode .....	29

### **2. Hình ảnh**

Hình 2.1: Bộ dữ liệu Global Superstore .....	18
Hình 3.1: Visual Studio Community 2022 .....	30
Hình 3.2: SQL Server Integration Services Projects .....	31
Hình 3.3: Microsoft SQL Server 2022 .....	32
Hình 3.4: Kết nối tới SQL Server 2022 .....	33
Hình 3.5: Khởi tạo 2 database có tên là GlobalSuperstore và WH_GlobalSuperstore	34

Hình 3.6: Mô hình thực hiện quá trình SSIS .....	39
Hình 3.7: Data Flow Task “Dim_Customer” .....	42
Hình 3.8: Data Flow Task “Dim_Product” .....	53
Hình 3.9: Cấu hình Data Flow Task “Dim_Location” .....	58
Hình 3.10: Cấu hình Data Flow Task “Dim_Market” .....	64
Hình 3.11: Cấu hình Data Flow Task “Dim_OrderDate” .....	69
Hình 3.12: Cấu hình Data Flow Task “Dim_ShipDate” .....	76
Hình 3.13: Cấu hình Data Flow Task “Dim_OrderPriority” .....	83
Hình 3.14: Cấu hình Data Flow Task “Dim_ShipMode” .....	88
Hình 3.15: Data Flow Task “Fact” .....	95
Hình 3.16: Execute SQL Task để xóa dữ liệu sau mỗi lần chạy .....	102
Hình 3.17: Execute SQL Task để thêm khóa ngoại vào bảng Fact .....	104
Hình 4.1: Microsoft Analysis Services Projects 2022 .....	108
Hình 4.2: Data Source View .....	117
Hình 4.3: Kết quả quá trình tạo Cubes .....	121
Hình 4.4: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 1 .....	139
Hình 4.5: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 2 .....	140
Hình 4.6: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 3 .....	140
Hình 4.7: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 4 .....	141
Hình 4.8: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 5 .....	141
Hình 4.9: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 6 .....	142
Hình 4.10: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 7 .....	142
Hình 4.11: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 8 .....	143
Hình 4.12: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 9 .....	143
Hình 4.13: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 10 .....	144
Hình 4.14: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 11 .....	144
Hình 4.15: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 13 .....	145
Hình 4.16: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 14 .....	146
Hình 4.17: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 15 .....	146
Hình 4.18: MDX Query – câu 1 .....	147
Hình 4.19: MDX Query – câu 2 .....	148

Hình 4.20: MDX Query – Câu 3 .....	148
Hình 4.21: MDX Query – Câu 4 .....	149
Hình 4.22: MDX Query – câu 5 .....	150
Hình 4.23: MDX Query – câu 6 .....	150
Hình 4.24: MDX Query – câu 7 .....	151
Hình 4.25: MDX Query – câu 8 .....	151
Hình 4.26: MDX Query – câu 9 .....	152
Hình 4.27: MDX Query – câu 10 .....	152
Hình 4.28: MDX Query – câu 11 .....	152
Hình 4.29: MDX Query – câu 12 .....	153
Hình 4.30: MDX Query – câu 13 .....	153
Hình 4.31: MDX Query – câu 14 .....	154
Hình 4.32: MDX Query – câu 15 .....	155
Hình 4.33: Kết quả truy vấn trên Excel – câu 1 .....	158
Hình 4.34: Kết quả truy vấn trên Excel – câu 1 – Chart .....	158
Hình 4.35: Kết quả truy vấn trên Excel – câu 2 .....	159
Hình 4.36: Kết quả truy vấn trên Excel – câu 2 – Chart .....	159
Hình 4.37: Kết quả truy vấn trên Excel – câu 3 .....	160
Hình 4.38: Kết quả truy vấn trên Excel – câu 3 – Chart .....	160
Hình 4.39: Kết quả truy vấn trên Excel – câu 4 .....	161
Hình 4.40: Kết quả truy vấn trên Excel – câu 4 – Chart .....	161
Hình 4.41: Kết quả truy vấn trên Excel – câu 5 .....	162
Hình 4.42: Kết quả truy vấn trên Excel – câu 6 .....	162
Hình 4.43: Kết quả truy vấn trên Excel – câu 6 – Chart .....	163
Hình 4.44: Kết quả truy vấn trên Excel – câu 7 .....	163
Hình 4.45: Kết quả truy vấn trên Excel – câu 7 – Chart .....	164
Hình 4.46: Kết quả truy vấn trên Excel – câu 8 .....	164
Hình 4.47: Kết quả truy vấn trên Excel – câu 9 .....	164
Hình 4.48: Kết quả truy vấn trên Excel – câu 9 – Chart .....	165
Hình 4.49: Kết quả truy vấn trên Excel – câu 10 .....	165
Hình 4.50: Kết quả truy vấn trên Excel – câu 11 .....	165

Hình 4.51: Kết quả truy vấn trên Excel – câu 12 .....	166
Hình 4.52: Kết quả truy vấn trên Excel – câu 13 .....	166
Hình 4.53: Kết quả truy vấn trên Excel – câu 14 .....	166
Hình 4.54: Kết quả truy vấn trên Excel – câu 15 .....	166
Hình 4.55: Kết quả truy vấn trên Power BI – câu 1 .....	169
Hình 4.56: Kết quả truy vấn trên Power BI – câu 2 – Table .....	169
Hình 4.57: Kết quả truy vấn trên Power BI – câu 2 – Chart .....	170
Hình 4.58: Kết quả truy vấn trên Power BI – câu 3 .....	170
Hình 4.59: Kết quả truy vấn trên Power BI – câu 4 .....	171
Hình 4.60: Kết quả truy vấn trên Power BI – câu 5 .....	171
Hình 4.61: Kết quả truy vấn trên Power BI – câu 6 .....	172
Hình 4.62: Kết quả truy vấn trên Power BI – câu 7 – Table .....	172
Hình 4.63: Kết quả truy vấn trên Power BI – câu 7 – Chart .....	173
Hình 4.64: Kết quả truy vấn trên Power BI – câu 8 .....	173
Hình 4.65: Kết quả truy vấn trên Power BI – câu 9 .....	174
Hình 4.66: Kết quả truy vấn trên Power BI – câu 10 .....	174
Hình 4.67: Kết quả truy vấn trên Power BI – câu 11 .....	175
Hình 4.68: Kết quả truy vấn trên Power BI – câu 12 .....	175
Hình 4.69: Kết quả truy vấn trên Power BI – câu 13 – Table .....	176
Hình 4.70: Kết quả truy vấn trên Power BI – câu 13 – Matrix .....	176
Hình 4.71: Kết quả truy vấn trên Power BI – câu 13 – Chart .....	177
Hình 4.72: Kết quả truy vấn trên Power BI – câu 14 – Table .....	177
Hình 4.73: Kết quả truy vấn trên Power BI – câu 14 – Matrix .....	178
Hình 4.74: Kết quả truy vấn trên Power BI – câu 14 – Chart .....	178
Hình 4.75: Kết quả truy vấn trên Power BI – câu 15 – Table .....	179
Hình 4.76: Kết quả truy vấn trên Power BI – câu 15 – Matrix .....	179
Hình 4.77: Kết quả truy vấn trên Power BI – câu 15 – Chart .....	180
Hình 5.1: Chủ đề cho bài toán khai thác dữ liệu .....	181
Hình 5.2: Ma trận tương quan giữa các thuộc tính .....	187
Hình 5.3: Phương pháp GridSearchCV .....	193
Hình 5.4: Thuật toán Rừng Ngẫu Nhiên (Random Forest) .....	194

Hình 5.5: Thuật toán Logistic Regression.....	196
Hình 5.6: Thuật toán Naïve Bayes.....	200
Hình 5.7: Thuật toán Perceptron.....	202
Hình 5.8: Vẽ một trong những cây trong Rừng Ngẫu nhiên .....	205
Hình 5.9: Ma trận nhầm lẫn thuật toán Random Forest .....	205
Hình 5.10: Ma trận nhầm lẫn thuật toán Logistic Regression.....	206
Hình 5.11: Ma trận nhầm lẫn thuật toán Naïve Bayes.....	207
Hình 5.12: Ma trận nhầm lẫn thuật toán Perceptron .....	208
Hình 5.13: Biểu đồ Histogram của thuộc tính Discount (RF-0) .....	210
Hình 5.14: Phần trăm của các mức điểm lợi nhuận theo thành phố (RF-0).....	211
Hình 5.15: Số lượng và tần suất các thành phố theo điểm lợi nhuận (RF-0) .....	211
Hình 5.16: Phần trăm của các mức điểm lợi nhuận theo bang (RF-0) .....	212
Hình 5.17: Số lượng và tần suất các bang theo điểm lợi nhuận (RF-0) .....	212
Hình 5.18: Phần trăm của các mức điểm lợi nhuận theo quốc gia (RF-0) .....	213
Hình 5.19: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (RF-0) .....	213
Hình 5.20: Biểu đồ Histogram của thuộc tính Discount (RF-1) .....	214
Hình 5.21: Phần trăm của các mức điểm lợi nhuận theo thành phố (RF-1).....	214
Hình 5.22: Số lượng và tần suất các thành phố theo điểm lợi nhuận (RF-1) .....	215
Hình 5.23: Phần trăm của các mức điểm lợi nhuận theo bang (RF-1) .....	215
Hình 5.24: Số lượng và tần suất các bang theo điểm lợi nhuận (RF-1) .....	216
Hình 5.25: Phần trăm của các mức điểm lợi nhuận theo quốc gia (RF-1) .....	216
Hình 5.26: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (RF-1) .....	217
Hình 5.27: Biểu đồ Histogram của thuộc tính Discount (LR-0) .....	219
Hình 5.28: Phần trăm của các mức điểm lợi nhuận theo thành phố (LR-0).....	220
Hình 5.29: Số lượng và tần suất các thành phố theo điểm lợi nhuận (LR-0) .....	220
Hình 5.30: Phần trăm của các mức điểm lợi nhuận theo bang (LR-0).....	221
Hình 5.31: Số lượng và tần suất các bang theo điểm lợi nhuận (LR-0) .....	221
Hình 5.32: Phần trăm của các mức điểm lợi nhuận theo quốc gia (LR-0).....	222
Hình 5.33: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (LR-0) .....	222
Hình 5.34: Biểu đồ Histogram của thuộc tính Discount (LR-1) .....	223
Hình 5.35: Phần trăm của các mức điểm lợi nhuận theo thành phố (LR-1).....	223

Hình 5.36: Số lượng và tần suất các thành phố theo điểm lợi nhuận (LR-1).....	224
Hình 5.37: Phần trăm của các mức điểm lợi nhuận theo bang (LR-1).....	224
Hình 5.38: Số lượng và tần suất các bang theo điểm lợi nhuận (LR-1).....	225
Hình 5.39: Phần trăm của các mức điểm lợi nhuận theo quốc gia (LR-1).....	225
Hình 5.40: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (LR-1).....	226
Hình 5.41: Biểu đồ Histogram của thuộc tính Discount (NB-0).....	227
Hình 5.42: Phần trăm của các mức điểm lợi nhuận theo thành phố (NB-0) .....	227
Hình 5.43: Số lượng và tần suất các thành phố theo điểm lợi nhuận (NB-0) .....	228
Hình 5.44: Phần trăm của các mức điểm lợi nhuận theo bang (NB-0) .....	228
Hình 5.45: Số lượng và tần suất các bang theo điểm lợi nhuận (NB-0) .....	229
Hình 5.46: Phần trăm của các mức điểm lợi nhuận theo quốc gia (NB-0) .....	229
Hình 5.47: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (NB-0).....	230
Hình 5.48: Biểu đồ Histogram của thuộc tính Discount (NB-1).....	230
Hình 5.49: Phần trăm của các mức điểm lợi nhuận theo thành phố (NB-1) .....	231
Hình 5.50: Số lượng và tần suất các thành phố theo điểm lợi nhuận (NB-1) .....	231
Hình 5.51: Phần trăm của các mức điểm lợi nhuận theo bang (NB-1) .....	232
Hình 5.52: Số lượng và tần suất các bang theo điểm lợi nhuận (NB-1) .....	232
Hình 5.53: Phần trăm của các mức điểm lợi nhuận theo quốc gia (NB-1) .....	233
Hình 5.54: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (NB-1).....	233
Hình 5.55: Biểu đồ Histogram của thuộc tính Discount (P-0) .....	234
Hình 5.56: Phần trăm của các mức điểm lợi nhuận theo thành phố (P-0) .....	235
Hình 5.57: Số lượng và tần suất các thành phố theo điểm lợi nhuận (P-0).....	235
Hình 5.58: Phần trăm của các mức điểm lợi nhuận theo bang (P-0).....	236
Hình 5.59: Số lượng và tần suất các bang theo điểm lợi nhuận (P-0).....	236
Hình 5.60: Phần trăm của các mức điểm lợi nhuận theo quốc gia (P-0) .....	237
Hình 5.61: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (P-0) .....	237
Hình 5.62: Biểu đồ Histogram của thuộc tính Discount (P-1) .....	238
Hình 5.63: Phần trăm của các mức điểm lợi nhuận theo thành phố (P-1) .....	238
Hình 5.64: Số lượng và tần suất các thành phố theo điểm lợi nhuận (P-1).....	239
Hình 5.65: Phần trăm của các mức điểm lợi nhuận theo bang (P-1).....	239
Hình 5.66: Số lượng và tần suất các bang theo điểm lợi nhuận (P-1).....	240

Hình 5.67: Phần trăm của các mức điểm lợi nhuận theo quốc gia (P-1).....	240
Hình 5.68: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (P-1).....	241
Hình 5.69: Mở terminal và chạy lệnh để thực thi ứng dụng .....	242
Hình 5.70: Giao diện web trên trình duyệt của người dùng .....	243
Hình 5.71: Nhập thông tin của order .....	243
Hình 5.72: Hiển thị kết quả dự đoán từ mô hình.....	244
Hình 5.73: Bấm vào nút “Browse files” hoặc drag files trong khu vực tô đỏ.....	244
Hình 5.74: Chọn file csv cần upload để dự đoán và nhấp Open .....	245
Hình 5.75: Ứng dụng nhận diện file đúng format và tải lên tập tin CSV thành công	245
Hình 5.76: Thông báo lỗi nếu file không đúng định dạng.....	246
Hình 5.77: Nhấn button “Download Predictions” để tải xuống kết quả dự đoán .....	246
Hình 5.78: File csv “ModelPredictions.csv” chứa kết quả dự đoán được tải về .....	246

## CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1 Lý do chọn đề tài

Trong thời đại công nghệ thông tin và dữ liệu lớn phát triển mạnh mẽ, việc quản lý và phân tích dữ liệu trở thành nhu cầu thiết yếu đối với các doanh nghiệp. Dữ liệu bán hàng ở các siêu thị là một nguồn thông tin quan trọng giúp doanh nghiệp hiểu rõ hơn về thị trường, khách hàng và các xu hướng kinh doanh. Tuy nhiên, việc xử lý và phân tích một khối lượng dữ liệu lớn đòi hỏi các công cụ và phương pháp tiên tiến. Do đó, nhóm chúng em quyết định nghiên cứu và triển khai hệ thống kho dữ liệu và OLAP để hỗ trợ doanh nghiệp trong việc xử lý, phân tích và trực quan hóa dữ liệu bán hàng một cách hiệu quả. Đồ án này giúp chúng em áp dụng kiến thức lý thuyết vào thực tiễn, phát triển kỹ năng làm việc nhóm, quản lý dự án và giải quyết vấn đề. Kết quả đạt được từ đồ án sẽ hỗ trợ các doanh nghiệp nâng cao hiệu quả quản lý và ra quyết định dựa trên dữ liệu, góp phần vào sự phát triển bền vững của doanh nghiệp.

### 1.2 Mô tả bài toán

Đề tài tập trung vào việc phân tích và xây dựng kho dữ liệu về doanh số bán hàng của chuỗi siêu thị toàn cầu. Bộ dữ liệu bao gồm các thông tin về khách hàng (mã ID, tên, phân khúc), sản phẩm (loại sản phẩm, tên sản phẩm, số lượng, lãi/lỗ), đơn hàng (mã ID, ngày đặt hàng, ngày giao hàng, phương thức vận chuyển, chi phí vận chuyển, giảm giá), và vị trí (quốc gia, thành phố, khu vực). Chúng em sẽ giải quyết các câu hỏi như: mặt hàng nào bán chạy nhất, lợi nhuận từ các loại sản phẩm khác nhau như thế nào, phân khúc khách hàng nào chi tiêu nhiều nhất, các chiến dịch giảm giá có hiệu quả không, và phí vận chuyển ảnh hưởng đến lợi nhuận như thế nào và nhiều các câu hỏi phức tạp khác đồng thời áp dụng Data Mining để khai thác tiềm năng từ dữ liệu. Trực quan hóa dữ liệu để tìm ra các thông tin quan trọng ảnh hưởng đến hoạt động kinh doanh của các siêu thị. Mục tiêu của chúng em là xây dựng hệ thống kho dữ liệu và OLAP để phân tích các khía cạnh này, từ đó hỗ trợ quá trình ra quyết định kinh doanh.

## CHƯƠNG 2: GIỚI THIỆU KHO DỮ LIỆU

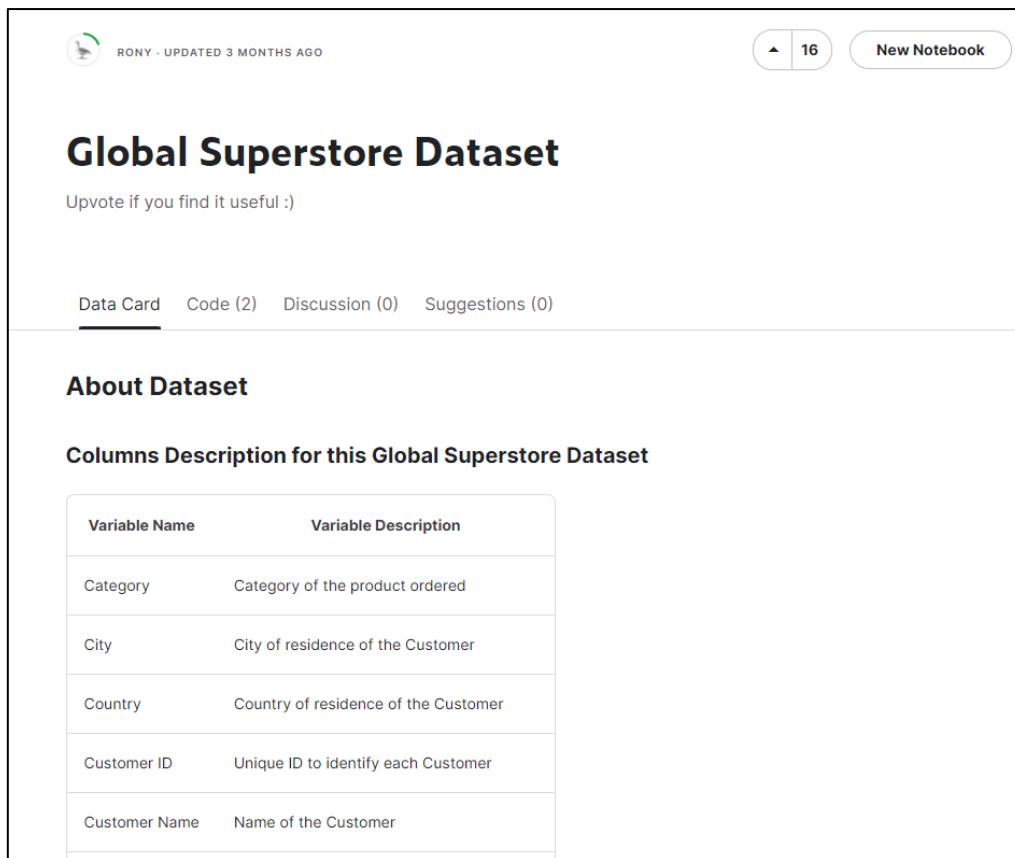
### 2.1 Giới thiệu bộ dữ liệu

#### 2.1.1 Thông tin nguồn dữ liệu

Bộ dữ liệu Global Superstore này cung cấp thông tin chi tiết về doanh số bán hàng của một chuỗi siêu thị toàn cầu. Nó bao gồm các thông tin về khách hàng (mã ID, tên, phân khúc), sản phẩm (loại sản phẩm, tên sản phẩm, số lượng, lãi/ lỗ), đơn hàng (mã ID, ngày đặt hàng, ngày giao hàng, phương thức vận chuyển, chi phí vận chuyển, giảm giá), và vị trí (quốc gia, thành phố, khu vực). Những thông tin này cho phép phân tích nhiều khía cạnh của hoạt động kinh doanh của chuỗi siêu thị.

#### 2.1.2 Thông tin chi tiết

- Data source [1] : <https://www.kaggle.com/datasets/ronysoliman/global-superstore-dataset>
- Chủ sở hữu: ronysoliman
- Ngày cập nhập gần nhất: 3 tháng trước
- Tên file: Global\_Superstore.csv
- Kích thước: 12.12 MB
- Dataset bao gồm 51290 dòng với 23 thuộc tính mô tả.



**Global Superstore Dataset**

Upvote if you find it useful :)

Data Card    Code (2)    Discussion (0)    Suggestions (0)

### About Dataset

**Columns Description for this Global Superstore Dataset**

Variable Name	Variable Description
Category	Category of the product ordered
City	City of residence of the Customer
Country	Country of residence of the Customer
Customer ID	Unique ID to identify each Customer
Customer Name	Name of the Customer

Hình 2.1: Bộ dữ liệu Global Superstore

### 2.1.3 Mô tả thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	Global_Orders_ID	Integer	Mã số đơn hàng toàn cầu, là một định danh duy nhất cho mỗi đơn hàng.
2	Order_ID	String	Mã số đơn hàng, định danh duy nhất cho từng đơn hàng.
3	Category	String	Danh mục sản phẩm, ví dụ như Technology (Công nghệ), Office Supplies (Văn phòng phẩm), Furniture (Nội thất).
4	City	String	Thành phố nơi đơn hàng được đặt.
5	Country	String	Quốc gia nơi đơn hàng được đặt.
6	Customer_Name	String	Tên khách hàng.

7	Market	String	Thị trường, ví dụ như USCA (Hoa Kỳ và Canada).
8	Customer_ID	String	Mã số khách hàng, định danh duy nhất cho mỗi khách hàng.
9	Order_Date	Date	Ngày đặt hàng.
10	Ship_Date	Date	Ngày giao hàng.
11	Order_Priority	String	Mức độ ưu tiên của đơn hàng.
12	Product_ID	String	Mã sản phẩm, định danh duy nhất cho mỗi sản phẩm.
13	Product_Name	String	Tên sản phẩm.
14	Region	String	Khu vực, ví dụ như Central (Trung tâm), East (Đông).
15	Segment	String	Phân khúc khách hàng, ví dụ như Home Office (Văn phòng tại nhà), Corporate (Doanh nghiệp).
16	Ship_Mode	String	Phương thức vận chuyển, ví dụ như Standard Class (Lớp tiêu chuẩn).
17	State	String	Địa chỉ đơn hàng được đặt.
18	Sub_Category	String	Tiêu danh mục sản phẩm, ví dụ như Accessories (Phụ kiện), Binders (Bìa hồ sơ).
19	Discount	Float	Mức giảm giá áp dụng cho đơn hàng.
20	Profit	Float	Lợi nhuận thu được từ đơn hàng.
21	Quantity	Integer	Số lượng sản phẩm được đặt.
22	Sales	Float	Doanh số bán hàng.
23	Shipping_Cost	Float	Chi phí vận chuyển.

Bảng 2.1: Mô tả các thuộc tính của bộ dữ liệu Global Superstore

#### 2.1.4 Mô tả chi tiết các thuộc tính danh mục

##### 1. Category

Mô tả: Danh mục sản phẩm.

Ý nghĩa giá trị: Giá trị chuỗi cho biết danh mục của sản phẩm, bao gồm:

Category		
STT	Giá trị	Ý nghĩa
1	Technology	Sản phẩm thuộc mặt hàng công nghệ như máy tính, điện thoại, thiết bị điện tử.
2	Office Supplies	Sản phẩm thuộc mặt hàng văn phòng phẩm như bút, giấy, tập vở, dụng cụ văn phòng.
3	Furniture	Sản phẩm thuộc mặt hàng nội thất như bàn, ghế, tủ, kệ.

Bảng 2.2: Mô tả chi tiết giá trị của thuộc tính Category

## 2. Sub\_Category

Mô tả: Tiêu danh mục sản phẩm.

Ý nghĩa giá trị: Giá trị chuỗi xác định tiêu danh mục của sản phẩm, chi tiết hơn so với Category. Các giá trị bao gồm:

Sub_Category		
STT	Giá trị	Ý nghĩa
1	Accessories	Phụ kiện
2	Appliances	Thiết bị
3	Art	Nghệ thuật
4	Binders	Bìa hồ sơ.
5	Bookcases	Kệ sách.
6	Chairs	Ghế
7	Copiers	Máy photocopy.
8	Envelopes	Phong bì.
9	Fasteners	Đồ gá
10	Furnishings	Đồ nội thất.
11	Labels	Nhãn
12	Machines	Máy móc
13	Paper	Giấy

14	Phones	Điện thoại.
15	Storage	Lưu trữ
16	Supplies	Vật tư
17	Tables	Bàn

Bảng 2.3: Mô tả chi tiết giá trị của thuộc tính Sub\_Category

### 3. Market

Mô tả: Thị trường.

Ý nghĩa giá trị: Giá trị chuỗi cho biết thị trường mà đơn hàng thuộc về, bao gồm:

Market		
STT	Giá trị	Ý nghĩa
1	USCA	Thị trường Hoa Kỳ và Canada
2	APAC	Thị trường Châu Á - Thái Bình Dương
3	EMEA	Thị trường Châu Âu, Trung Đông và Châu Phi
4	LATAM	Thị trường Châu Mỹ Latin

Bảng 2.4: Mô tả chi tiết giá trị của thuộc tính Market

### 4. Order\_Priority

Mô tả: Mức độ ưu tiên của đơn hàng.

Ý nghĩa giá trị: Giá trị chuỗi xác định mức độ ưu tiên của đơn hàng, bao gồm:

Order_Priority		
STT	Giá trị	Ý nghĩa
1	Critical	Mức độ ưu tiên quan trọng
2	High	Mức độ ưu tiên cao
3	Medium	Mức độ ưu tiên trung bình
4	Low	Mức độ ưu tiên thấp

Bảng 2.5: Mô tả chi tiết giá trị của thuộc tính Order\_Priority

### 5. Ship\_Mode

Mô tả: Phương thức vận chuyển.

Ý nghĩa giá trị: Giá trị chuỗi xác định phương thức vận chuyển. Các giá trị bao gồm:

Ship_Mode		
STT	Giá trị	Ý nghĩa
1	First Class	Phương thức vận chuyển ưu tiên nhất với thời gian giao hàng nhanh nhất, thường được sử dụng cho các đơn hàng quan trọng hoặc cần giao gấp.
2	Second Class	Phương thức vận chuyển nhanh, nhưng không nhanh bằng hạng nhất. Thích hợp cho các đơn hàng cần giao trong thời gian tương đối ngắn nhưng không khẩn cấp.
3	Standard Class	Phương thức vận chuyển tiêu chuẩn, thường có thời gian giao hàng lâu hơn nhưng với chi phí thấp hơn. Thích hợp cho các đơn hàng không khẩn cấp.
4	Same Day	Phương thức vận chuyển rất nhanh, đảm bảo đơn hàng được giao trong cùng ngày đặt hàng. Thích hợp cho các đơn hàng cần giao ngay lập tức.

Bảng 2.6: Mô tả chi tiết giá trị của thuộc tính Ship\_Mode

## 6. Segment

Mô tả: Phân khúc khách hàng.

Ý nghĩa giá trị: Giá trị chuỗi xác định phân khúc của khách hàng. Các giá trị bao gồm:

Segment		
STT	Giá trị	Ý nghĩa
1	Consumer	Khách hàng thuộc phân khúc tiêu dùng
2	Corporate	Khách hàng thuộc phân khúc doanh nghiệp
3	Home Office	Khách hàng thuộc phân khúc văn phòng tại nhà

Bảng 2.7: Mô tả chi tiết giá trị của thuộc tính Segment

## 7. City

Mô tả: Thành phố nơi đơn hàng được đặt.

Ý nghĩa giá trị: Giá trị chuỗi cho biết tên thành phố mà đơn hàng được đặt. Một số ví dụ về giá trị bao gồm:

City		
STT	Giá trị	Ý nghĩa
1	New York	Thành phố New York
2	Los Angeles	Thành phố Los Angeles
3	Chicago	Thành phố Chicago
...	...	Tên các thành phố khác

Bảng 2.8: Mô tả chi tiết giá trị của thuộc tính City

## 8. State

Mô tả: Bang nơi đơn hàng được đặt.

Ý nghĩa giá trị: Giá trị chuỗi cho biết tên bang mà đơn hàng được đặt. Một số ví dụ về giá trị bao gồm:

State		
STT	Giá trị	Ý nghĩa
1	California	Bang California
2	Texas	Bang Texas
3	Florida	Bang Florida
...	...	Tên các bang khác

Bảng 2.9: Mô tả chi tiết giá trị của thuộc tính Segment

## 9. Country

Mô tả: Quốc gia nơi đơn hàng được đặt.

Ý nghĩa giá trị: Giá trị chuỗi cho biết tên quốc gia mà đơn hàng được đặt. Một số ví dụ về giá trị bao gồm:

Country		
STT	Giá trị	Ý nghĩa
1	United States	Quốc gia United States

2	Canada	Quốc gia Canada
3	Mexico	Quốc gia Mexico
...	...	Tên các quốc gia khác

Bảng 2.10: Mô tả chi tiết giá trị của thuộc tính Country

## 10. Region

Mô tả: Khu vực địa lý nơi đơn hàng được đặt.

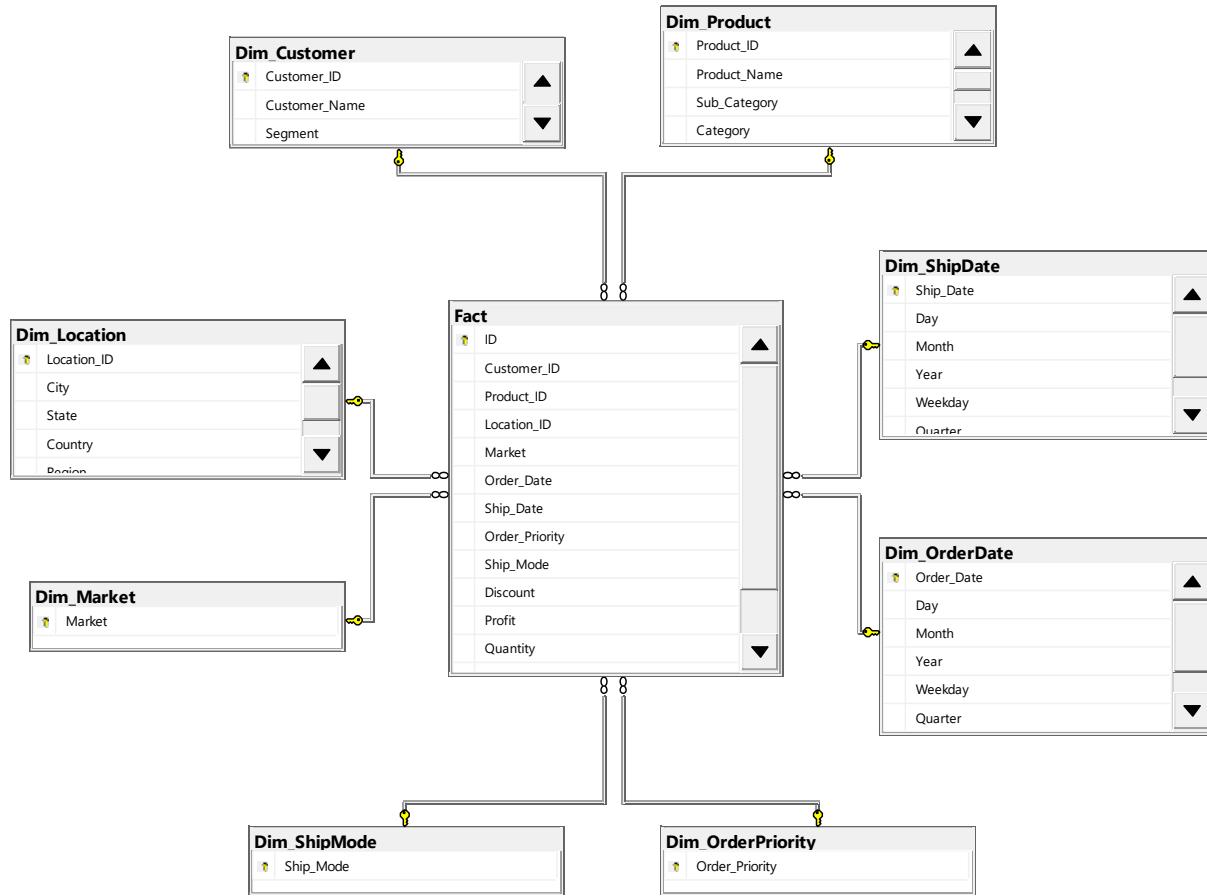
Ý nghĩa giá trị: Giá trị chuỗi cho biết khu vực địa lý mà đơn hàng thuộc về. Các giá trị cụ thể bao gồm:

Region		
STT	Giá trị	Ý nghĩa
1	Central	Khu vực trung tâm của các quốc gia, thường bao gồm các thành phố lớn và trung tâm kinh tế
2	East	Khu vực phía đông
3	Africa	Lục địa Châu Phi
4	EMEA	(Europe, Middle East, and Africa) Châu Âu, Trung Đông và Châu Phi
5	South	Khu vực phía nam, nổi bật với nền văn hóa độc đáo và các lĩnh vực kinh tế như năng lượng và nông nghiệp
6	Southeast Asia	Khu vực Đông Nam Á
7	West	Khu vực phía tây, nổi bật với nền kinh tế công nghệ và giải trí
8	North	Khu vực phía bắc, thường có khí hậu lạnh hơn và trung tâm công nghiệp quan trọng
9	Oceania	Khu vực Châu Đại Dương
10	Central Asia	Khu vực Trung Á
11	North Asia	Khu vực Bắc Á
12	Canada	Toàn bộ quốc gia Canada.
13	Caribbean	Khu vực Caribbean

Bảng 2.11: Mô tả chi tiết giá trị của thuộc tính Region

## 2.2 Xây dựng kho dữ liệu

### 2.2.1 Lược đồ hình sao (Star Schema)



Bảng 2.12: Lược đồ hình sao của kho dữ liệu

### 2.2.2 Mô tả chi tiết bảng FACT và các bảng DIM

#### 2.2.2.1 Bảng FACT

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	ID	Integer	PK	Mã số đơn hàng toàn cầu, là một định danh duy nhất cho mỗi đơn hàng.
2	Customer_ID	String	FK	Mã số khách hàng, định danh duy nhất cho mỗi khách hàng.

3	Product_ID	Integer	FK	Mã sản phẩm, định danh duy nhất cho mỗi sản phẩm.
4	Location_ID	Integer	FK	Mã số định danh vị trí giao hàng.
5	Market	String	FK	Thị trường, ví dụ như USCA (Hoa Kỳ và Canada).
6	Order_Date	Date	FK	Ngày đặt hàng
7	Ship_Date	Date	FK	Ngày giao hàng
8	Order_Priority	String	FK	Mức độ ưu tiên của đơn hàng.
9	Ship_Mode	String	FK	Phương thức vận chuyển, ví dụ như Standard Class (Lớp tiêu chuẩn).
10	Discount	Float		Mức giảm giá áp dụng cho đơn hàng.
11	Profit	Float		Lợi nhuận thu được từ đơn hàng.
12	Quantity	Integer		Số lượng sản phẩm được đặt.
13	Sales	Float		Doanh số bán hàng.
14	Shipping_Cost	Float		Chi phí vận chuyển.

Bảng 2.13: Mô tả chi tiết bảng Fact

#### 2.2.2.2 Bảng Dim\_Customer

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Customer_ID	String	PK	Mã số khách hàng, định danh duy nhất cho mỗi khách hàng.
2	Customer_Name	String		Tên khách hàng.

3	Segment	String		Phân khúc khách hàng, ví dụ như Home Office (Văn phòng tại nhà), Corporate (Doanh nghiệp).
---	---------	--------	--	--

Bảng 2.14: Mô tả chi tiết bảng Dim\_Customer

#### 2.2.2.3 Bảng Dim\_Product

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Product_ID	Integer	PK	Mã sản phẩm, định danh duy nhất cho mỗi sản phẩm.
2	Product_Name	String		Tên sản phẩm.
3	Sub_Category	String		Tiêu danh mục sản phẩm, ví dụ như Accessories (Phụ kiện), Binders (Bìa hồ sơ).
4	Category	String		Danh mục sản phẩm, ví dụ như Technology (Công nghệ), Office Supplies (Văn phòng phẩm), Furniture (Nội thất).

Bảng 2.15: Mô tả chi tiết bảng Dim\_Product

#### 2.2.2.4 Bảng Dim\_Location

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Location_ID	Integer	PK	Mã số định danh duy nhất cho mỗi vị trí
2	City	String		Thành phố nơi đơn hàng được đặt.
3	State	String		bang nơi đơn hàng được đặt.
4	Country	String		Quốc gia nơi đơn hàng được đặt.
5	Region	String		Khu vực, ví dụ như Central (Trung tâm), East (Đông).

Bảng 2.16: Mô tả chi tiết bảng Dim\_Location

#### 2.2.2.5 Dim\_Market

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Ship_Mode	String	PK	Phương thức vận chuyển

Bảng 2.17: Mô tả chi tiết bảng Dim\_Market

#### 2.2.2.6 Bảng Dim\_OrderDate

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Order_Date	Date	PK	Ngày đặt hàng
2	Day	Integer		Ngày trong tháng khi đơn hàng được đặt
3	Month	Integer		Tháng khi đơn hàng được đặt.
4	Year	Integer		Năm khi đơn hàng được đặt
5	Weekday	Integer		Ngày trong tuần khi đơn hàng được đặt
6	Quarter	Integer		Quý trong năm khi đơn hàng được đặt

Bảng 2.18: Mô tả chi tiết bảng Dim\_OrderDate

#### 2.2.2.7 Bảng Dim\_ShipDate

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Ship_Date	Date	PK	Ngày giao hàng
2	Day	Integer		Ngày trong tháng khi đơn hàng được giao
3	Month	Integer		Tháng khi đơn hàng được giao.
4	Year	Integer		Năm khi đơn hàng được giao.
5	Weekday	Integer		Ngày trong tuần khi đơn hàng được giao
6	Quarter	Integer		Quý trong năm khi đơn hàng được giao

Bảng 2.19: Mô tả chi tiết bảng Dim\_ShipDate

#### 2.2.2.8 Dim\_OrderPriority

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Order_Priority	String	PK	Mức độ ưu tiên của đơn hàng

Bảng 2.20: Mô tả chi tiết bảng Dim\_OrderPriority

#### 2.2.2.9 Bảng Dim\_ShipMode

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Ship_Mode	String	PK	Phương thức vận chuyển

Bảng 2.21: Mô tả chi tiết bảng Dim\_ShipMode

## CHƯƠNG 3: QUÁ TRÌNH XÂY DỰNG KHO DỮ LIỆU (SSIS)

### 3.1 Chuẩn bị các công cụ

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

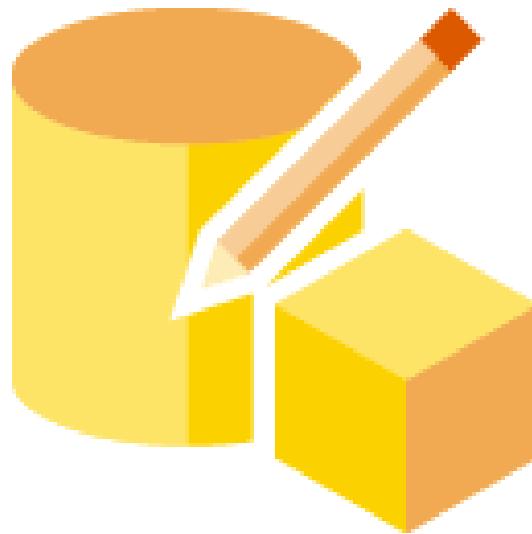
#### 1. Visual Studio Community 2022



Hình 3.1: Visual Studio Community 2022

Link Download [2]: <https://visualstudio.microsoft.com/vs/community/>

#### 2. SQL Server Integration Services Projects



*Hình 3.2: SQL Server Integration Services Projects*

Link Download [3]:

<https://marketplace.visualstudio.com/items?itemName=SSIS.SqlServerIntegrationServicesProjects>

### 3. Microsoft SQL Server 2022

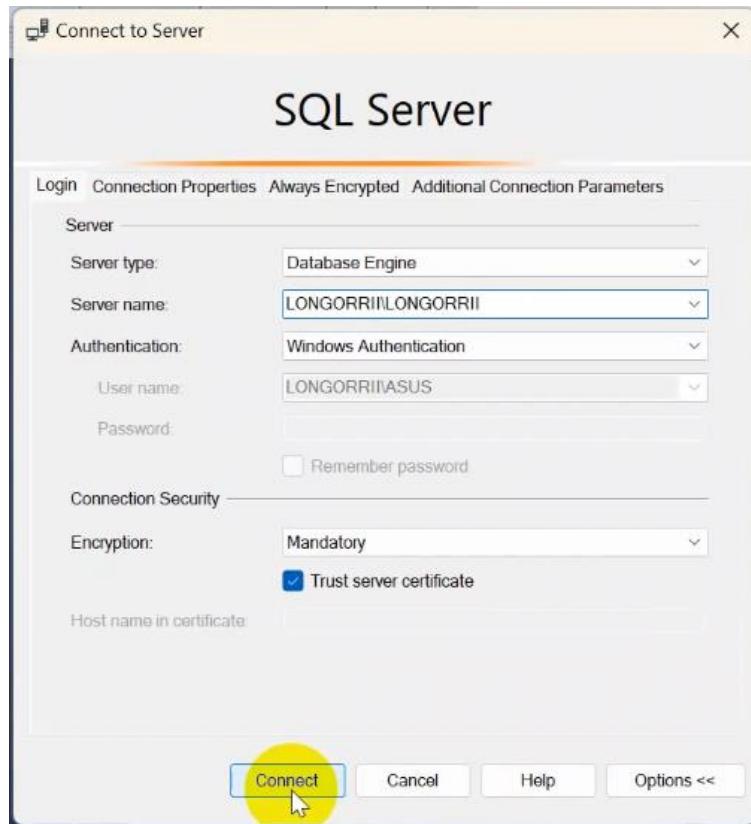


*Hình 3.3: Microsoft SQL Server 2022*

Link Download [4]: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>  
(Chọn phiên bản Developer)

### 3.2 Chuẩn bị cơ sở dữ liệu

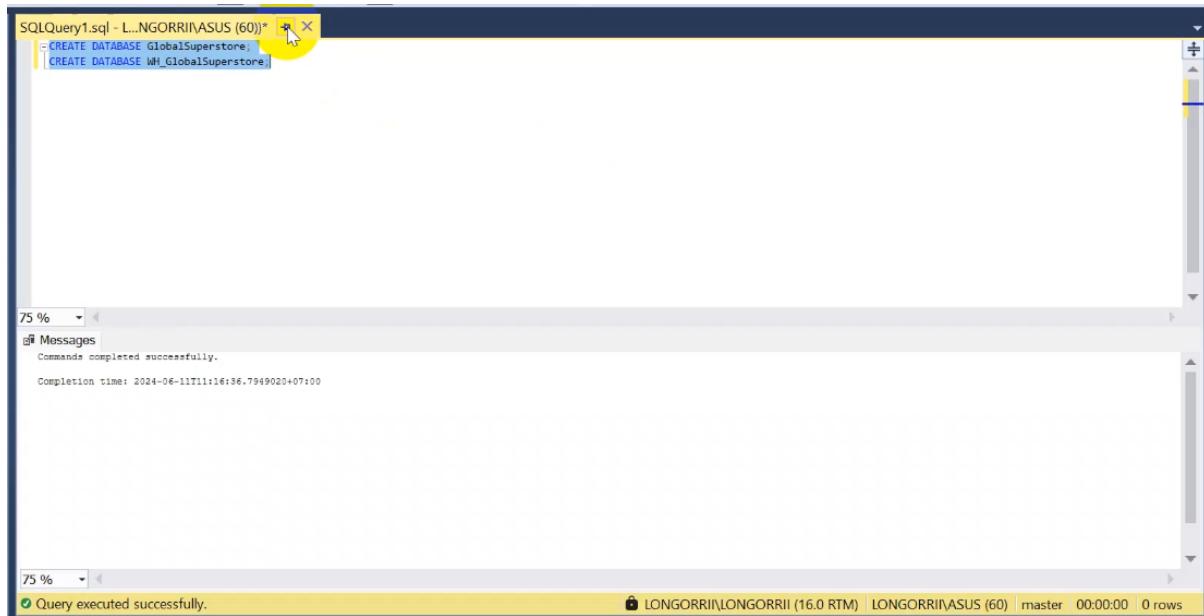
**Bước 1:** Mở SQL Server 2022 và kết nối với server bằng tài khoản user của window (Windows Authentication).



Hình 3.4: Kết nối tới SQL Server 2022

**Bước 2:** Khởi tạo 2 database có tên là **GlobalSuperstore** và **WH\_GlobalSuperstore** trong đó:

- **GlobalSuperstore:** Là database chứa dữ liệu gốc ban đầu
- **WH\_GlobalSuperstore:** Là database chứa dữ liệu các bảng Dim và bảng Fact sau quá trình thực hiện SSIS



SQLQuery1.sql - L...NGORRI\ASUS (60)\*

```

CREATE DATABASE GlobalSuperstore;
CREATE DATABASE WH_GlobalSuperstore;

```

75 %

Messages

Commands completed successfully.

Completion time: 2024-06-11T11:16:36.7949020+07:00

75 %

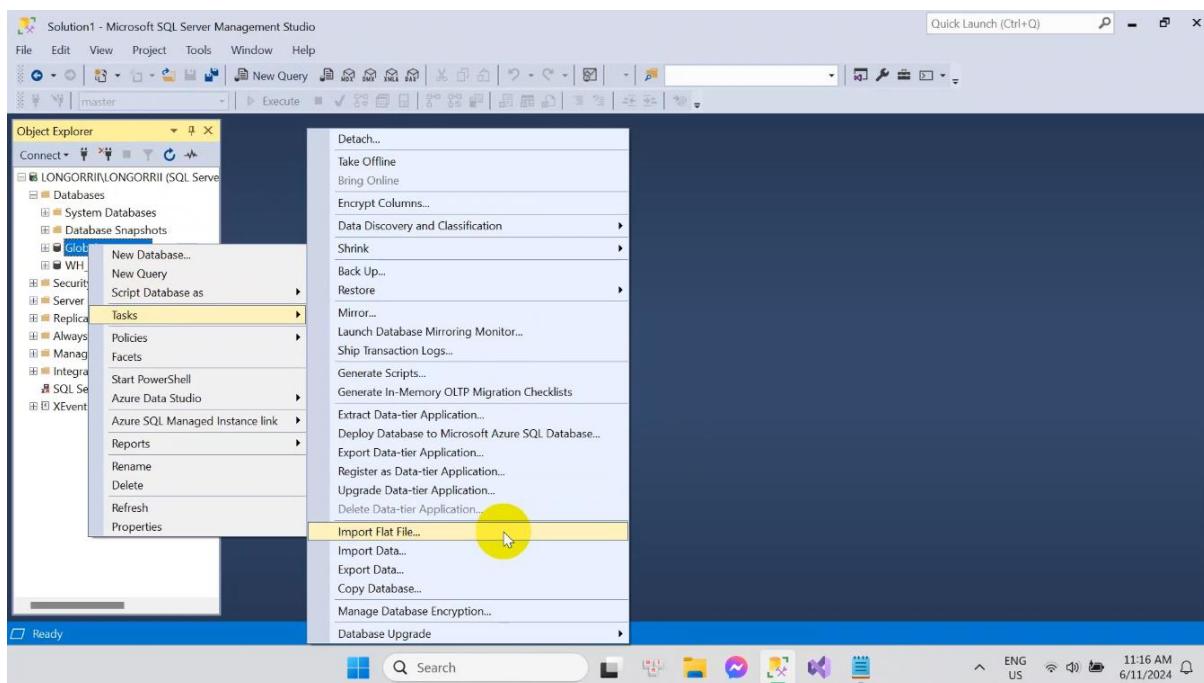
Query executed successfully.

LONGORRI\LONGORRI (16.0 RTM) | LONGORRI\ASUS (60) | master | 00:00:00 | 0 rows

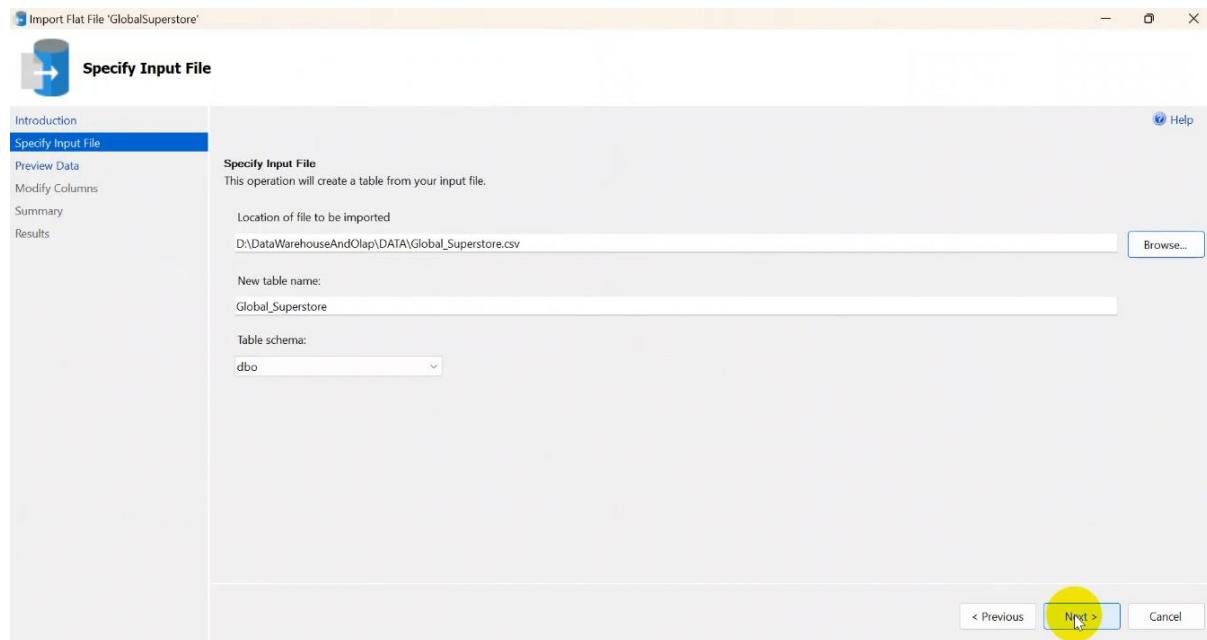
Hình 3.5: Khởi tạo 2 database có tên là GlobalSuperstore và WH\_GlobalSuperstore

### Bước 3: Import dữ liệu từ file csv gốc vào database **GlobalSuperstore**

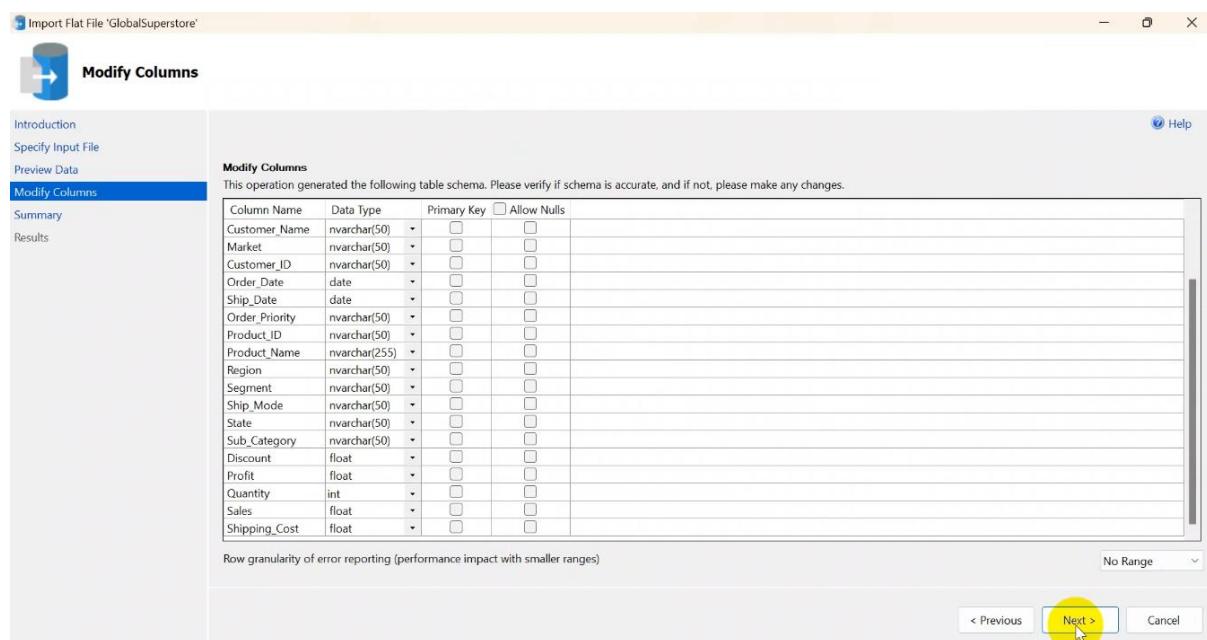
- Click chuột phải vào database GlobalSuperstore -> Tasks -> Import Flat File



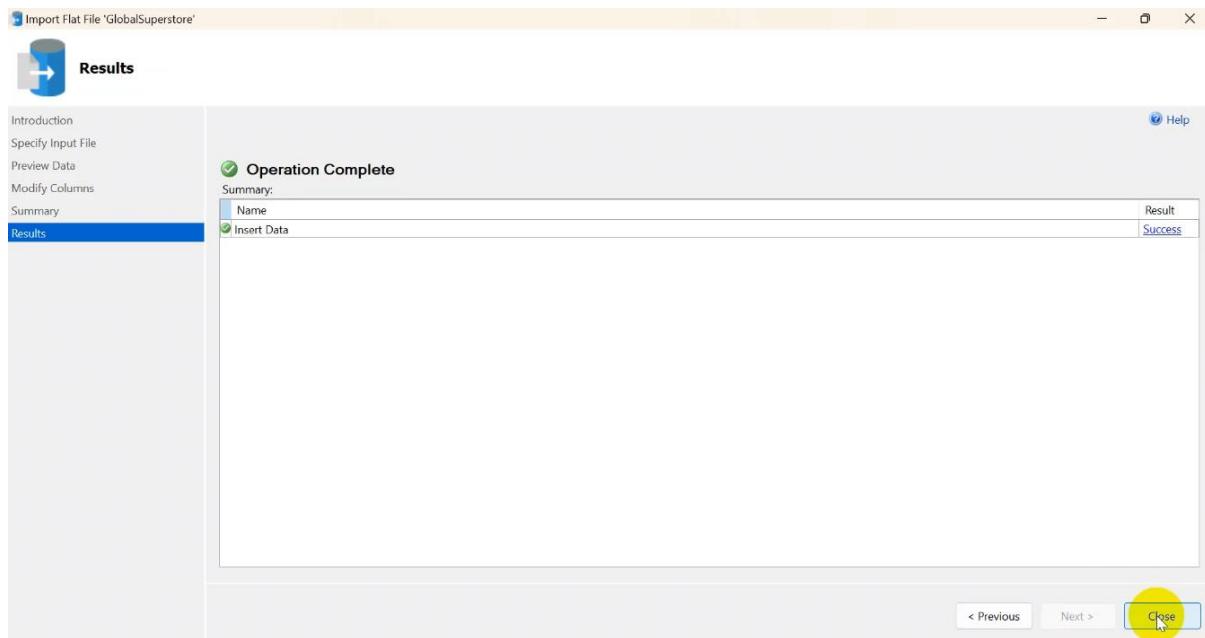
- Ở mục Specify Input File tiến hành chọn đường dẫn tới file csv gốc và nhấn Next



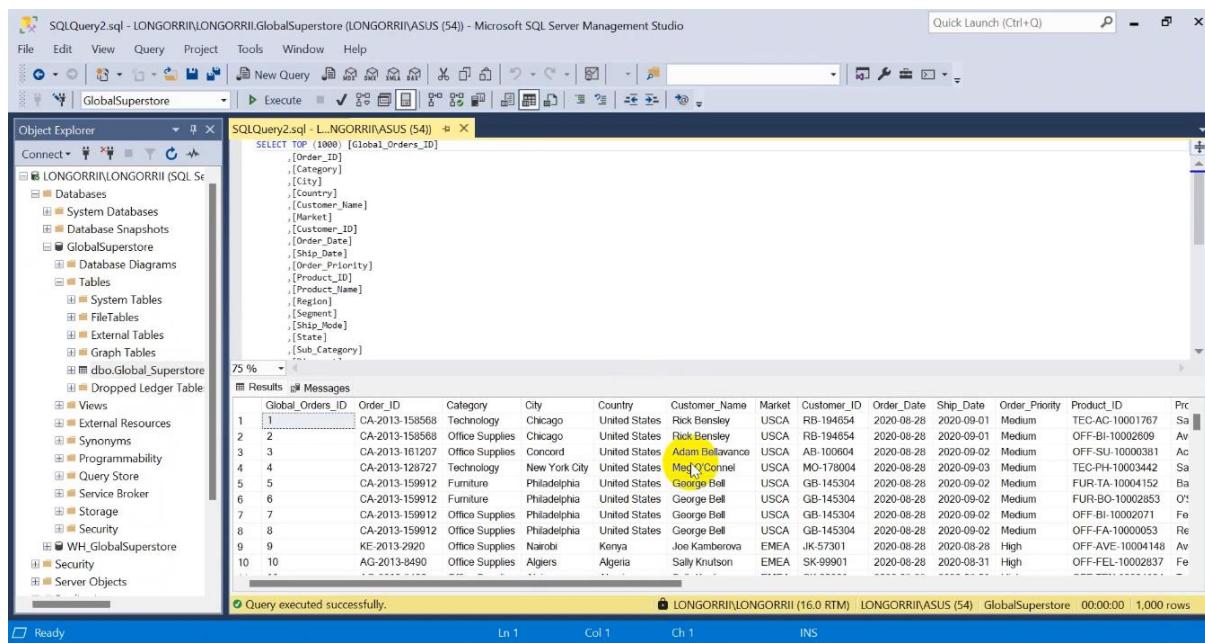
- Ở mục Modify Columns ta thay đổi dữ liệu cho phù hợp với các thuộc tính và nhấn Next



- Sau khi import dữ liệu vào database GlobalSuperstore thành công, nhấn “Close” để hoàn tất

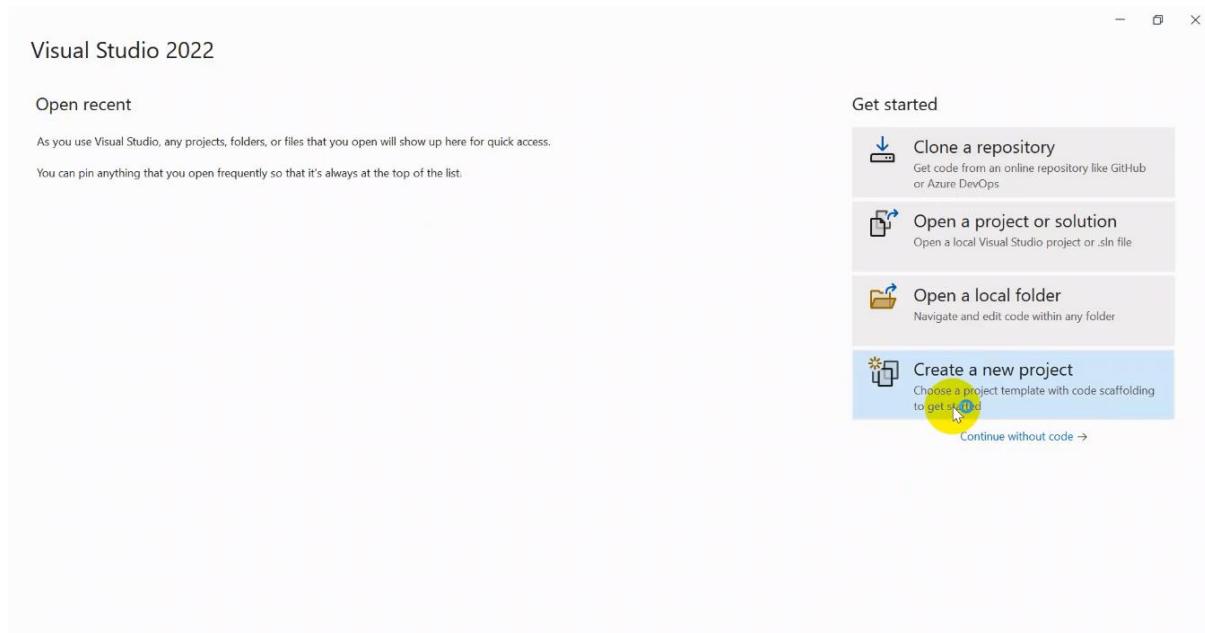


- Kiểm tra dữ liệu xem đã import thành công hay không, ta thấy dữ liệu đã được import thành công vào database GlobalSuperstore

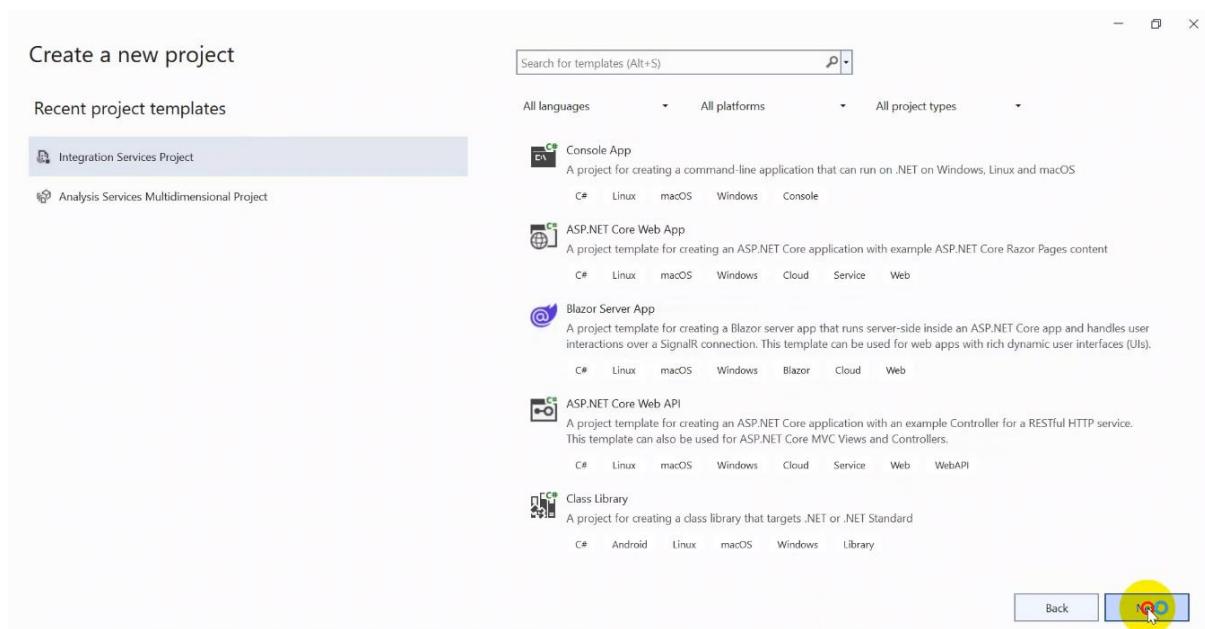


### 3.3 Tạo mới project SSIS

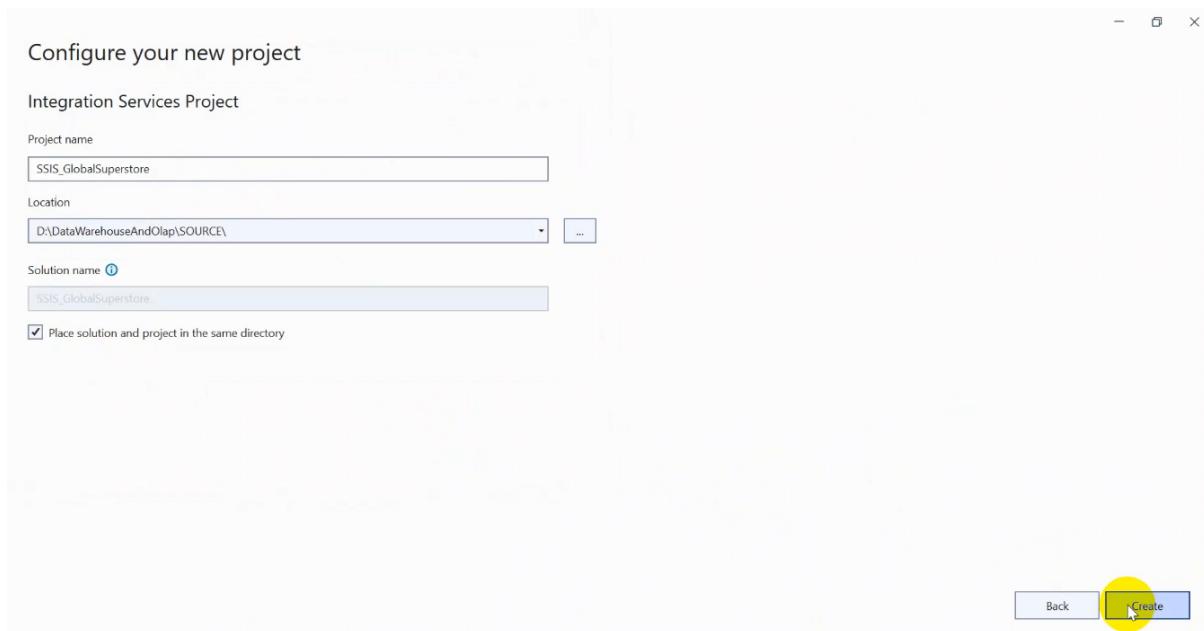
Bước 1: Mở Visual Studio 2022 và nhấn chọn “Create a new project”



## Bước 2: Chọn “Integration Services Projects” và nhấn Next

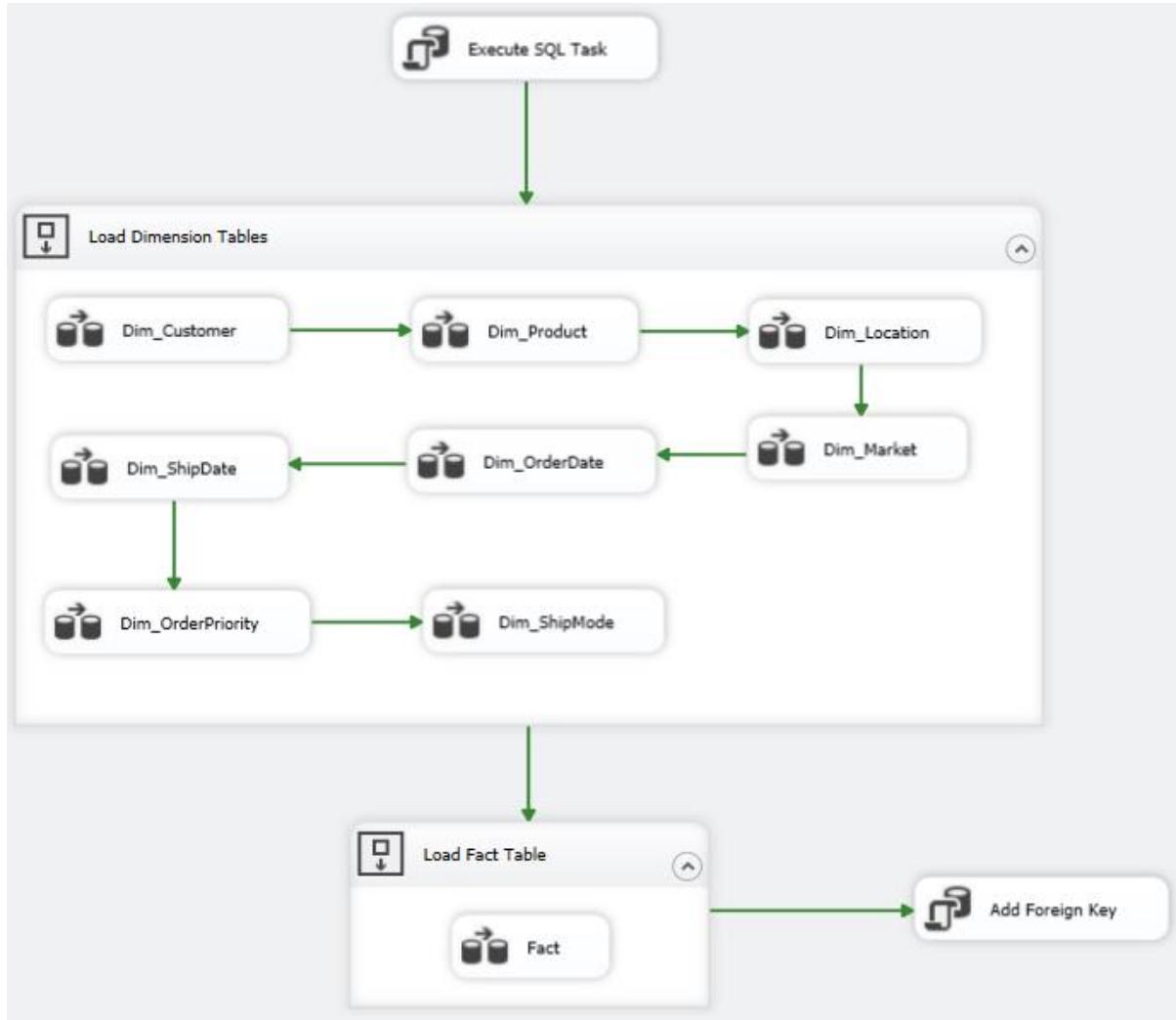


## Bước 3: Tiến hành đặt tên project là “SSIS\_GlobalSuperstore”, chọn đường dẫn lưu thư mục dự án và nhấn “Create” để hoàn tất việc tạo mới



### 3.4 Cấu hình và thực hiện quá trình SSIS

#### 3.4.1 Mô hình thực hiện quá trình SSIS



Hình 3.6: Mô hình thực hiện quá trình SSIS

**Bước 1: Execute SQL Task:** Xóa dữ liệu các bảng Dim và bảng Fact để đảm bảo mỗi lần chạy project không gây ra lỗi.

**Bước 2: Sequence Container (Load Dimension Tables):** Thực hiện việc cấu hình các data flow task để load dữ liệu vào các bảng Dim.

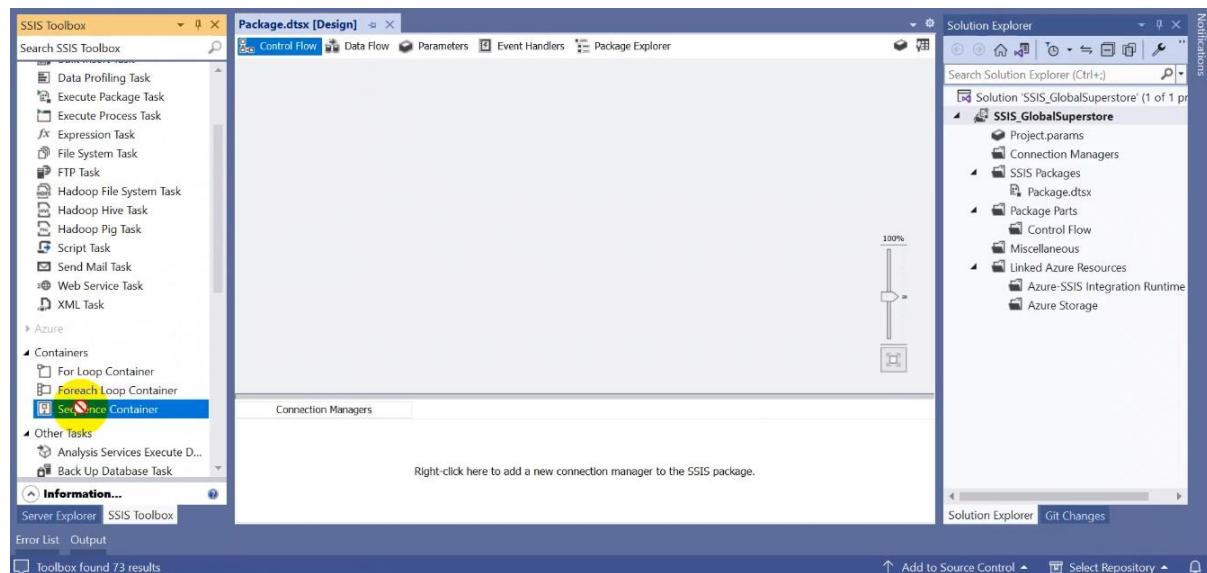
**Bước 3: Sequence Container (Load Fact Table):** Thực hiện việc cấu hình Data Flow Task để load dữ liệu vào bảng Fact.

**Bước 4: Execute SQL Task (Add Foreign Key):** Tạo các khóa ngoại

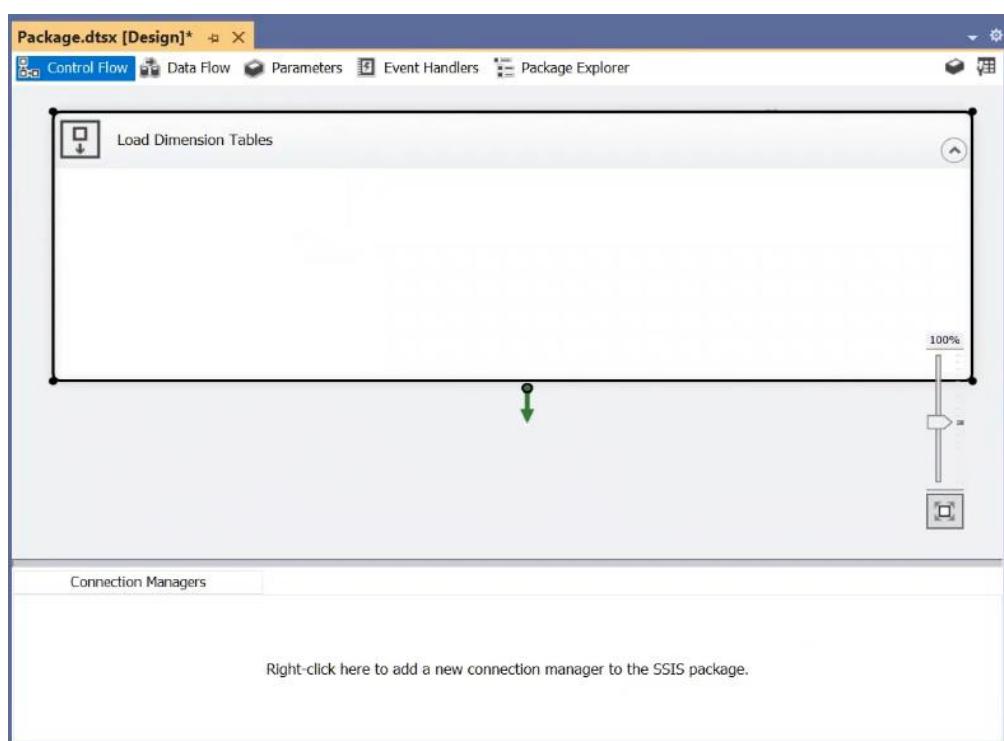
### 3.4.2 Load Dimension Tables

#### 3.4.2.1 Tạo mới Sequence Container

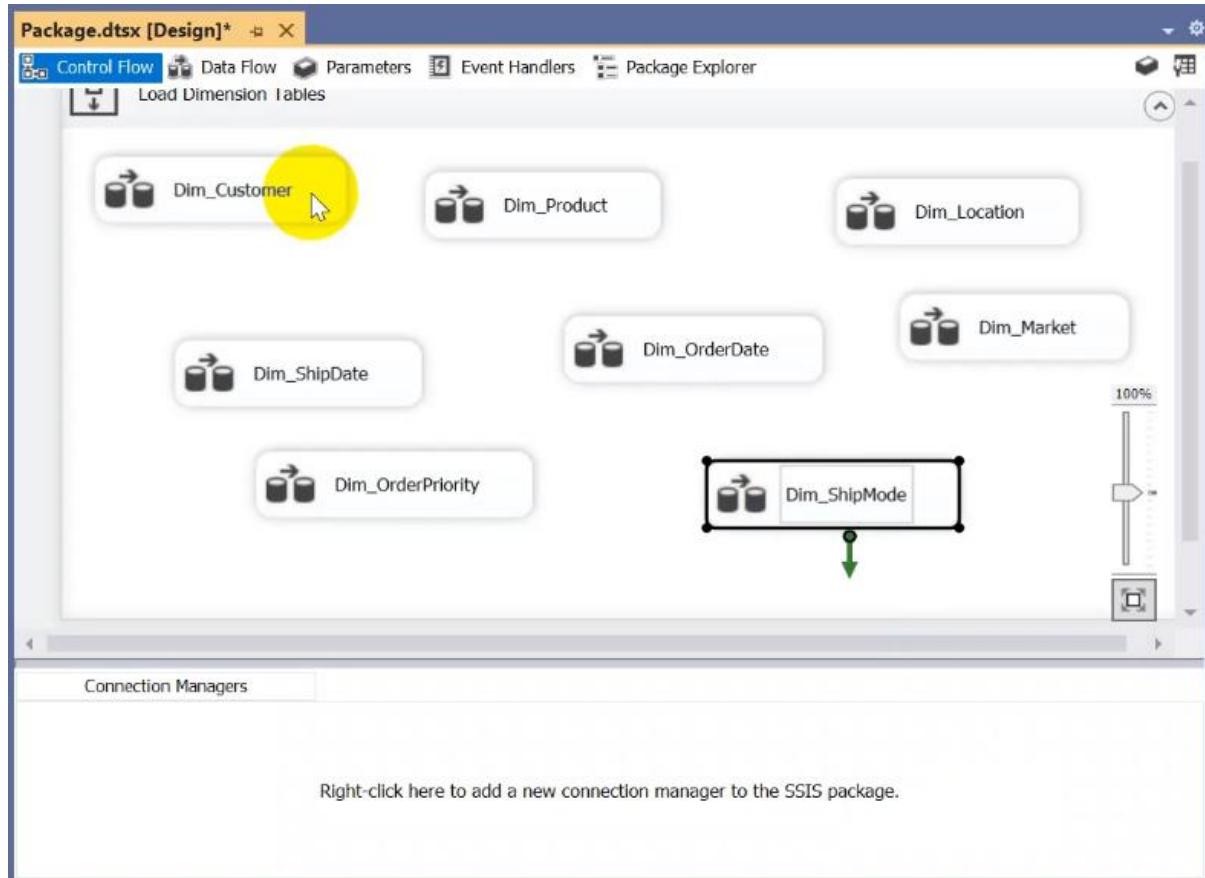
**Bước 1:** Trên thanh công cụ SSIS Toolbox, trong mục “Containers” ta chọn “Sequence Container”



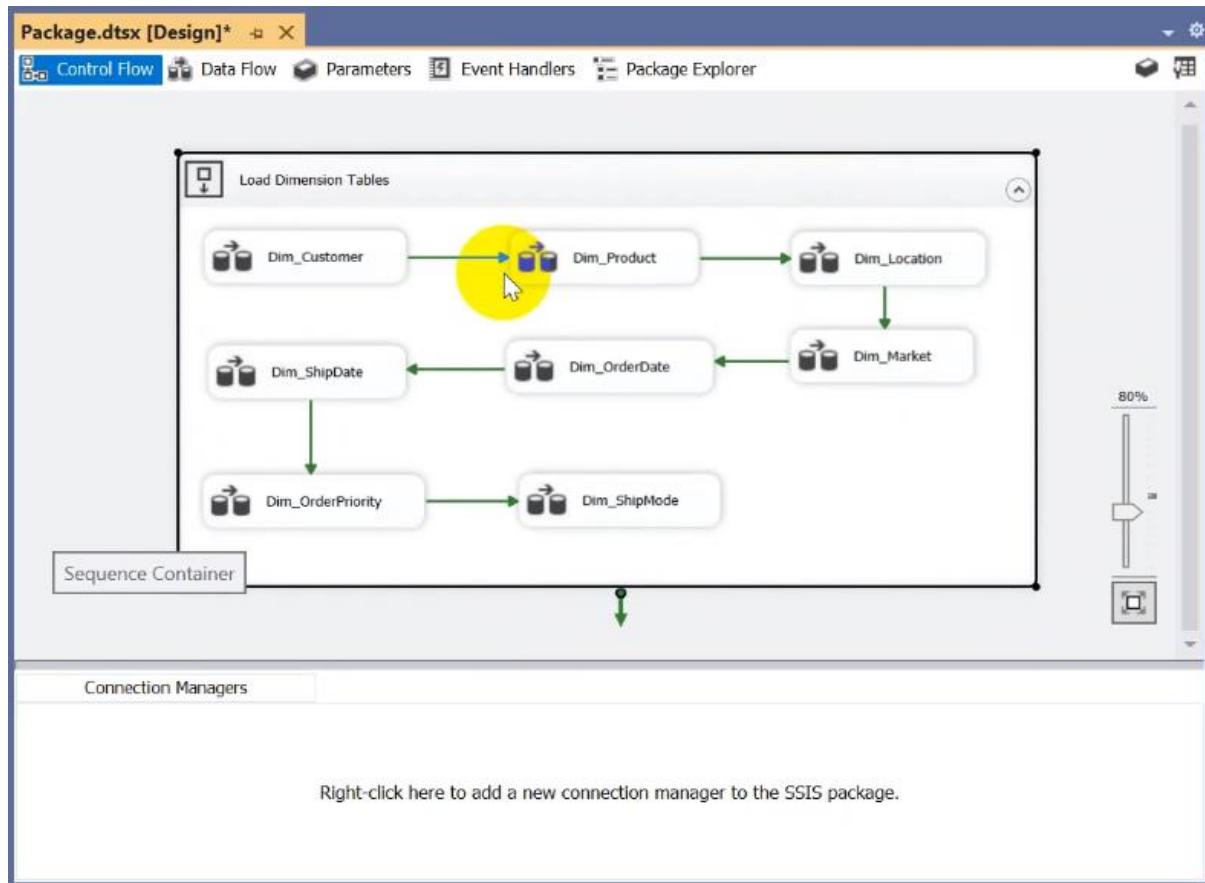
**Bước 2:** Kéo thả vào giao diện làm việc và đổi tên Sequence Container thành “Load Dimension Tables”



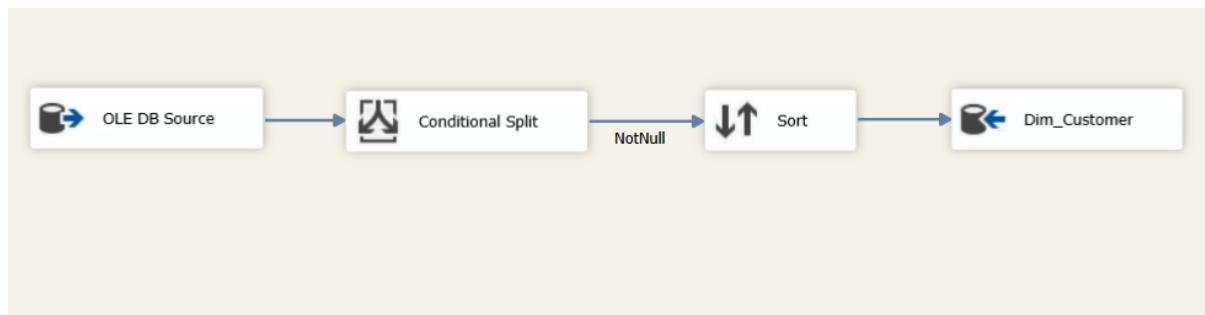
**Bước 3:** Lần lượt thêm các “**Data Flow Task**” vào sequence container “**Load Dimension Tables**”, có tổng cộng 8 Data Flow Task mỗi Data Flow Task tương ứng với mỗi bảng Dim. Đổi tên từng Data Flow Task cho tương ứng với mỗi bảng Dim.



**Bước 4:** Kết nối các Data Flow Task để thực hiện theo thứ tự chỉ định.



### 3.4.2.2 Cấu hình Data Flow Task “Dim\_Customer”

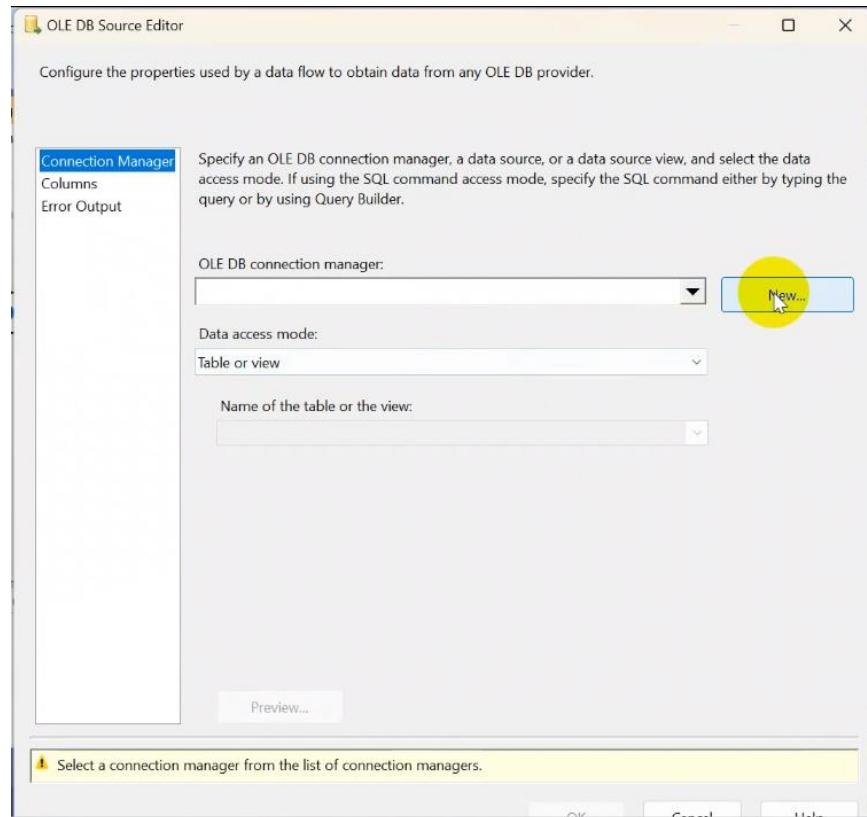


Hình 3.7: Data Flow Task “Dim\_Customer”

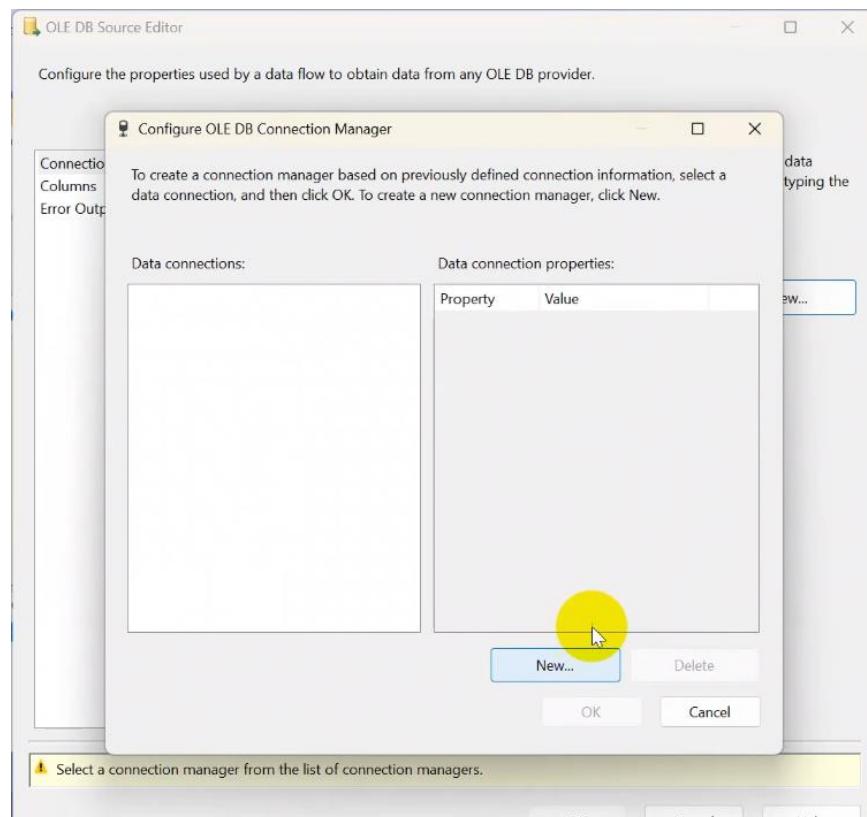
**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn Edit để đến giao diện “OLE DB Source Editor”.

\* Thiết lập connection đến database chứa dữ liệu gốc “Global Superstore”

- Trong mục Connection Manager. Nhấn New để thiết lập “OLE DB connection manager”.

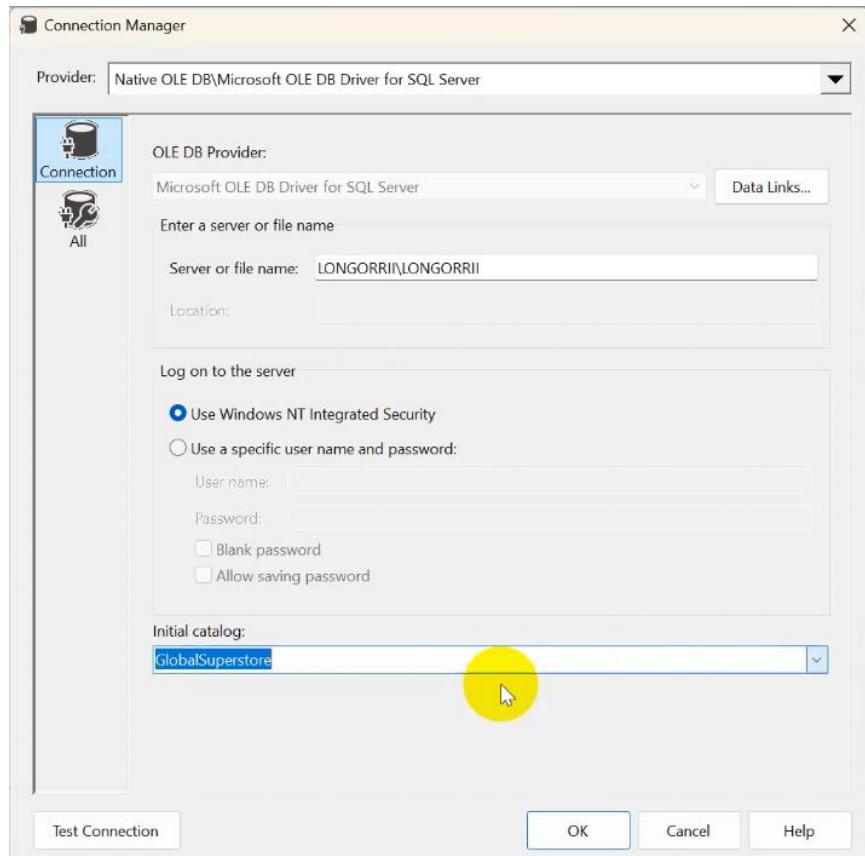


- Trong “Configure OLE DB Connection Manager” tiếp tục nhấn New

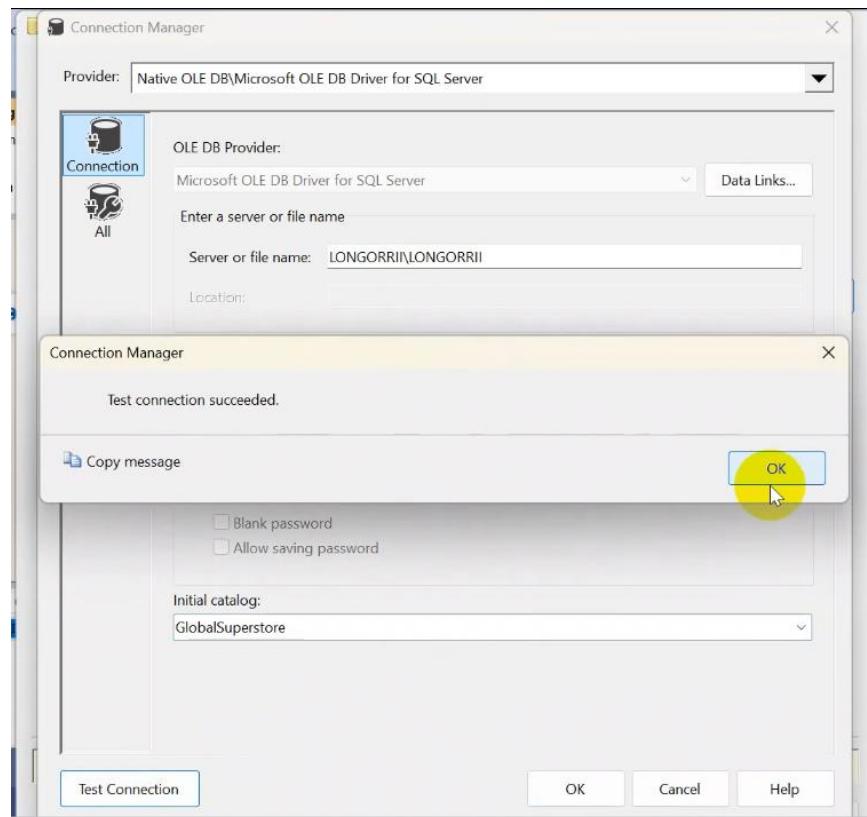


- Tiếp tục trong Connection Manager ta thiết lập các thông số sau:

- **Provider:** Microsoft OLE DB Driver for SQL Server
- **Server or file name:** LONGORRII\LONGORRII (tên server của SQL Server)
- **Initial catalog:** GlobalSuperstor (tên database chứa dữ liệu gốc)

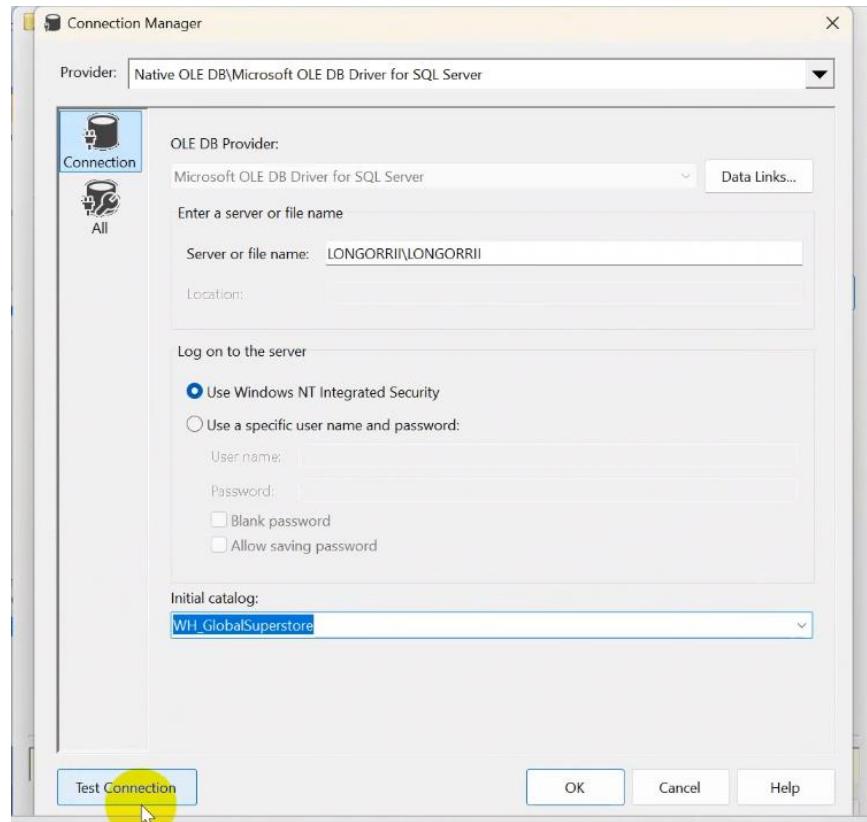


- Nhấn **Test Connection** để kiểm tra kết nối.

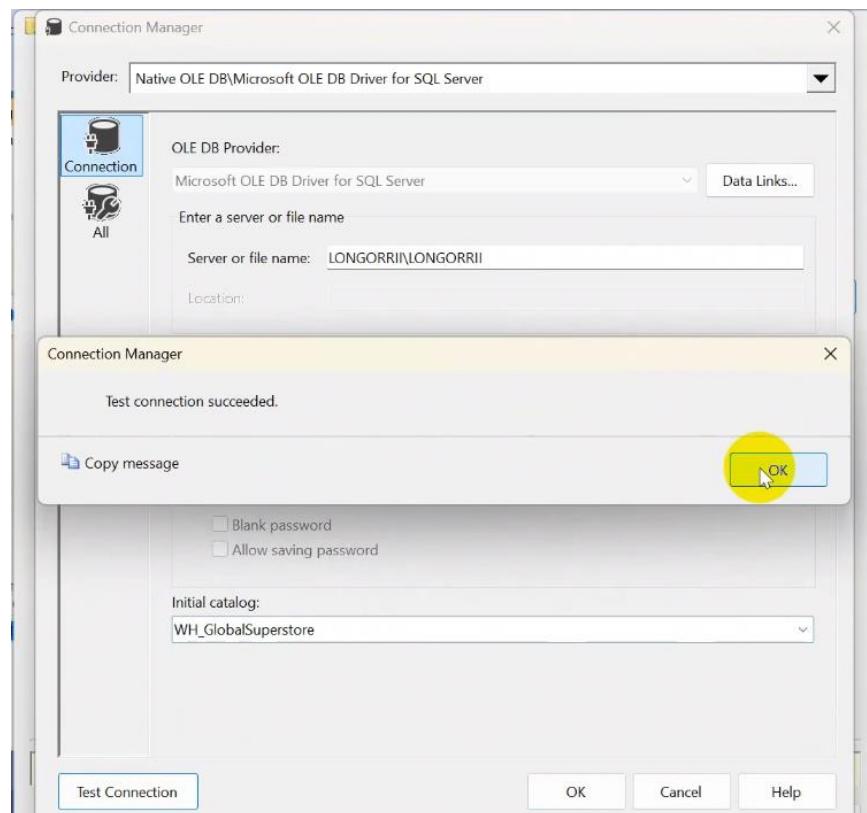


\* Thiết lập connection đến warehouse database “WH\_Global Superstore”

- Các bước thực hiện kết nối trước giống với bước thiết lập đến database “GlobalSuperstore”
- Trong **Connection Manager** ta thiết lập các thông số sau:
  - o **Provider:** Microsoft OLE DB Driver for SQL Server
  - o **Server or file name:** LONGORRII\LONGORRII (tên server của SQL Server)
  - o **Initial catalog:** WH\_GlobalSuperstor (tên database chứa dữ liệu bảng Dim và bảng Fact)

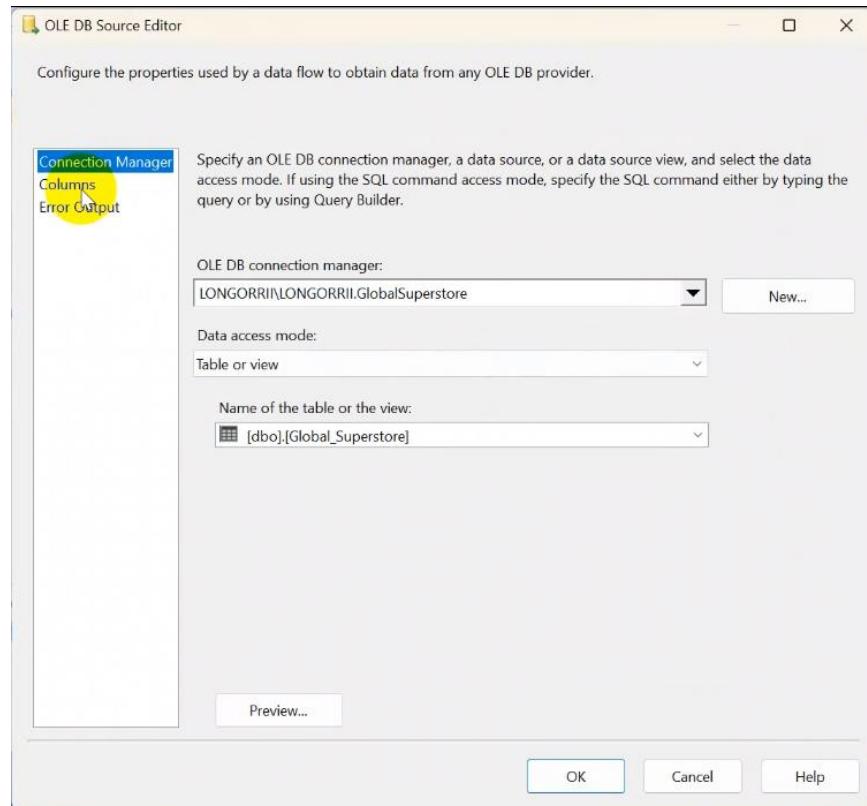


- Nhấn **Test Connection** để kiểm tra kết nối.

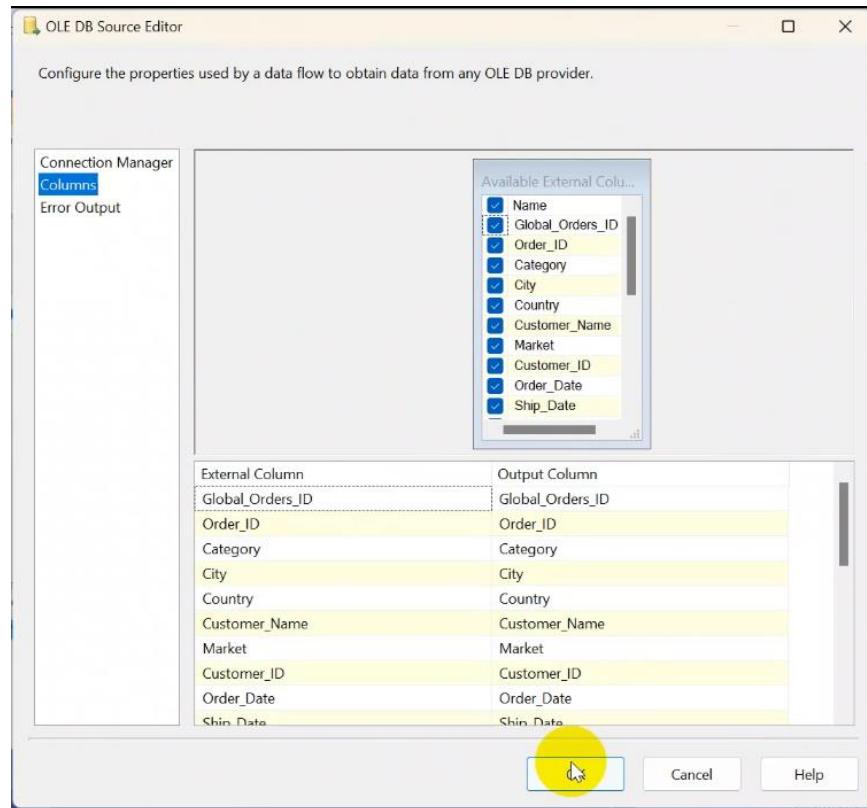


#### \* Thiết lập OLE DB Source Editor:

- Trong mục **Connection Manager** ta chọn:
  - o **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORRII\LONGORRII.GlobalSuperstore.
  - o **Name of the table or the view:** [dbo].[ GlobalSuperstore].



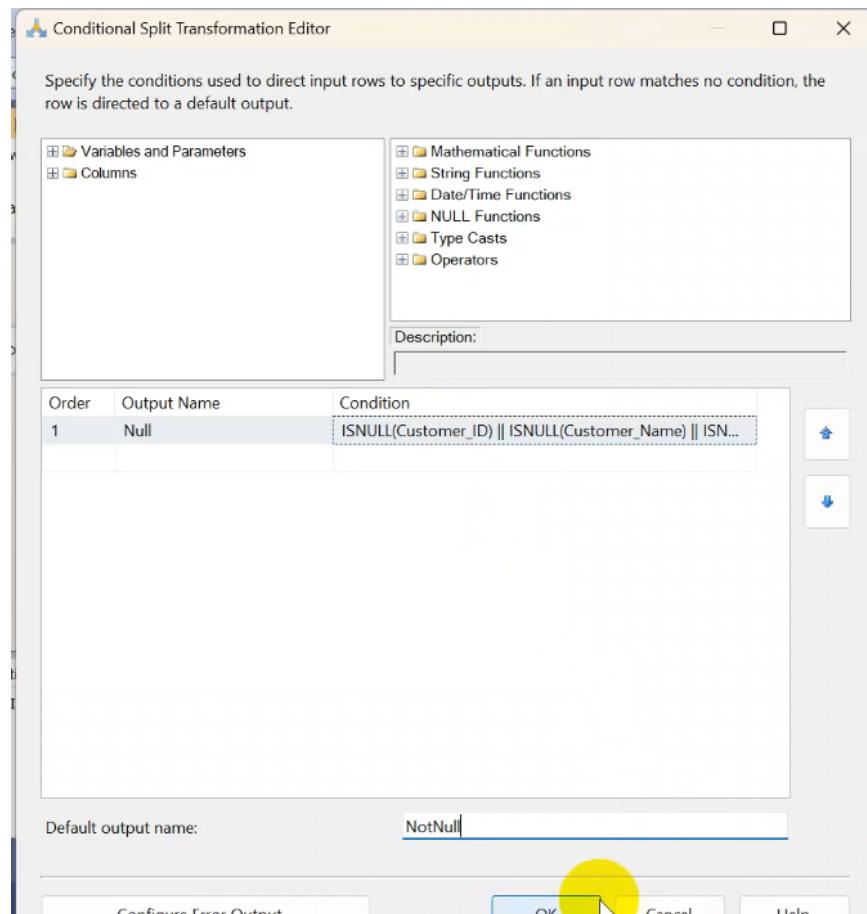
- Trong mục **Columns**, ta kiểm tra lại lần nữa các thuộc tính và nhấn **OK**



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* **Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.**

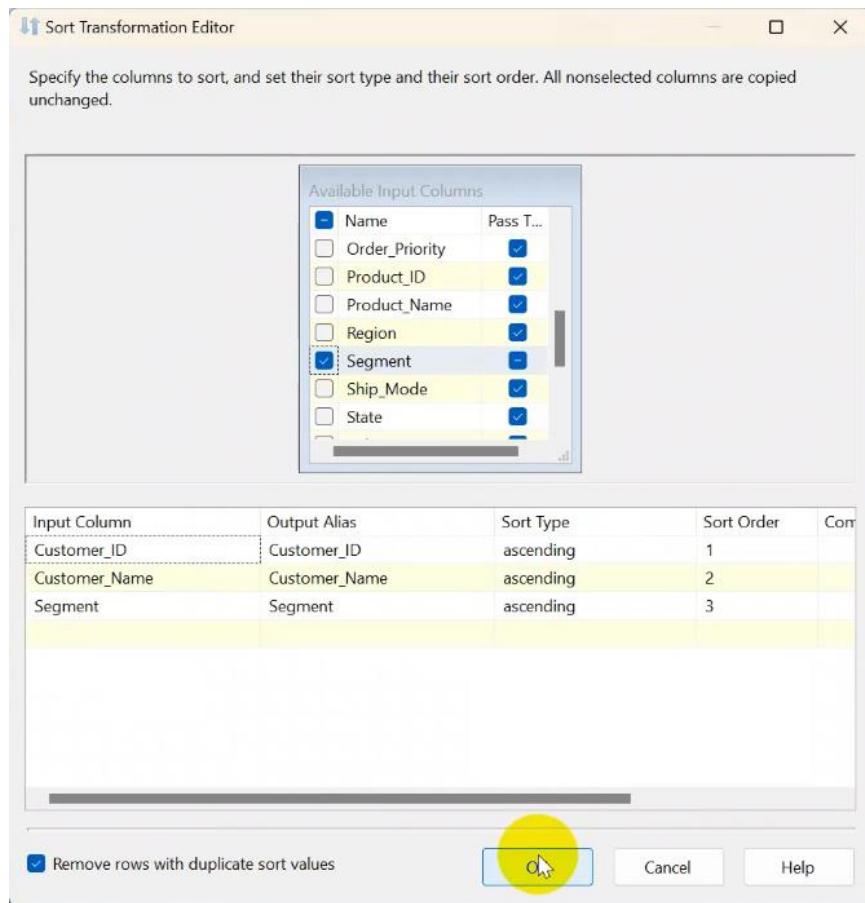
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Customer\_ID) || ISNULL(Customer\_Name) || ISNULL(Segment)
  - **Output Name:** Null
  - **Default output name:** NotNull



**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

**\* Thiết lập Sort Transformation Editor:**

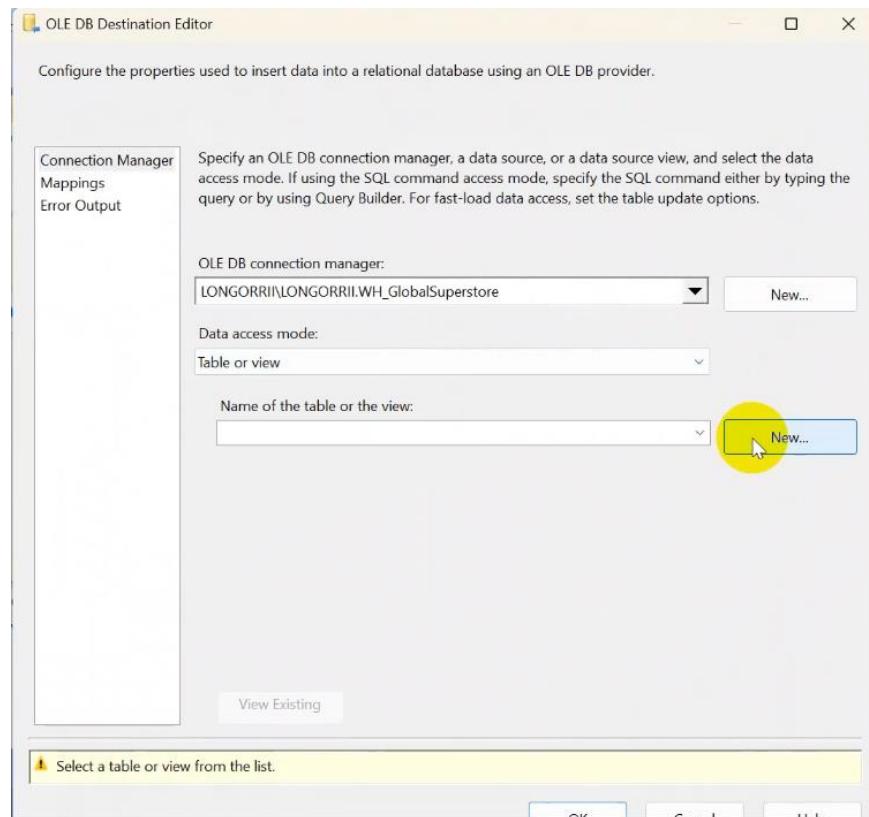
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_Customer như sau:
  - Customer\_ID: Ta lấy thuộc tính này làm khóa chính cho bảng Dim\_Customer
  - Customer\_Name
  - Segment
- Tick chọn “**Remove rows with duplicate sort values**” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**



**Bước 4: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “OLE DB Destination” và chọn **Edit** để đến giao diện “OLE DB Destination Editor”.

#### \* Thiết lập OLE DB Destination Editor

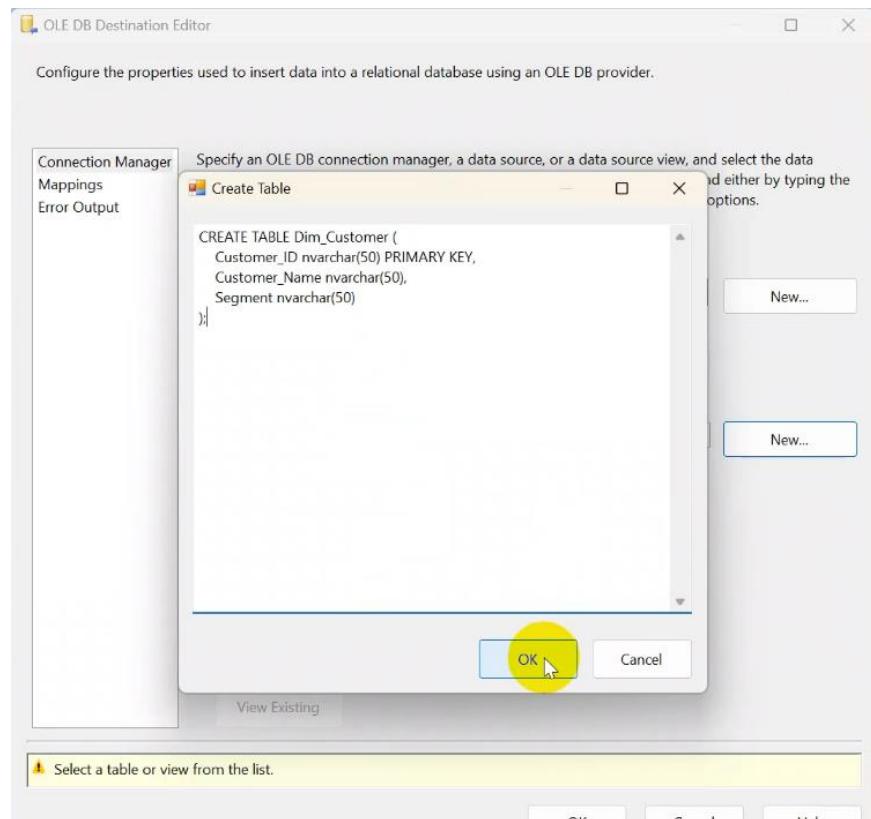
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORII\LONGORII.WH\_GlobalSuperstore.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_Customer**



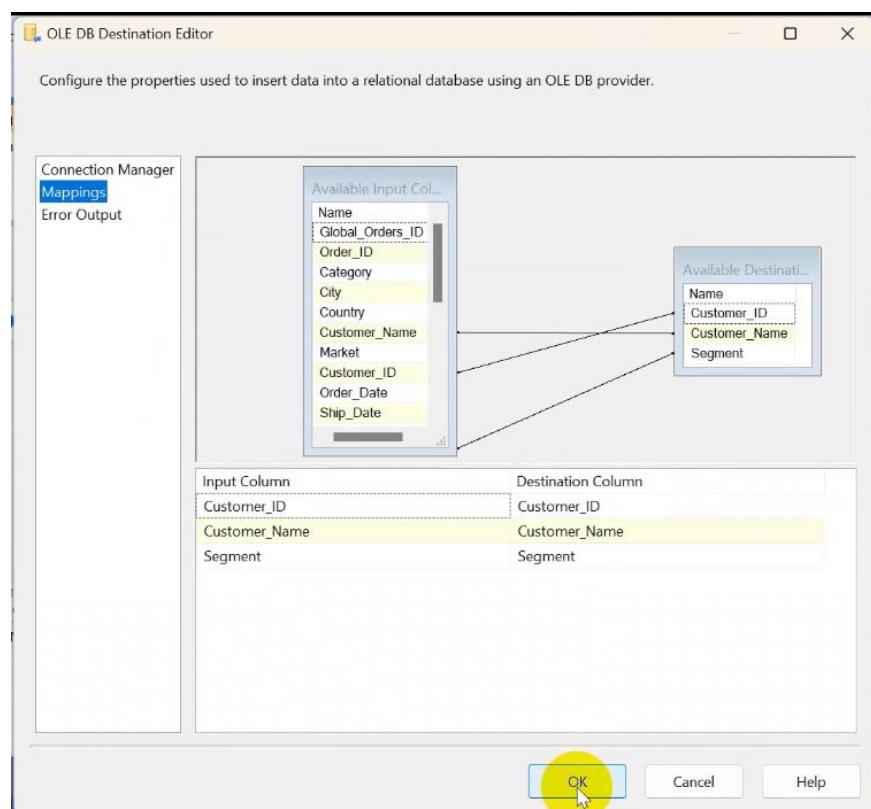
- Dán câu lệnh SQL sau để tạo bảng **Dim\_Customer** và nhấn **OK**

**Nội dung câu lệnh SQL:**

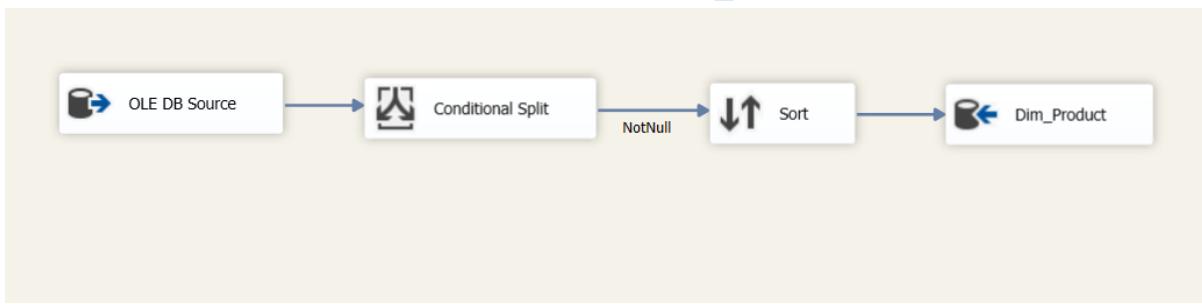
```
CREATE TABLE Dim_Customer (
    Customer_ID nvarchar(50) PRIMARY KEY,
    Customer_Name nvarchar(50),
    Segment nvarchar(50)
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



### 3.4.2.3 Cấu hình Data Flow Task “Dim\_Product”

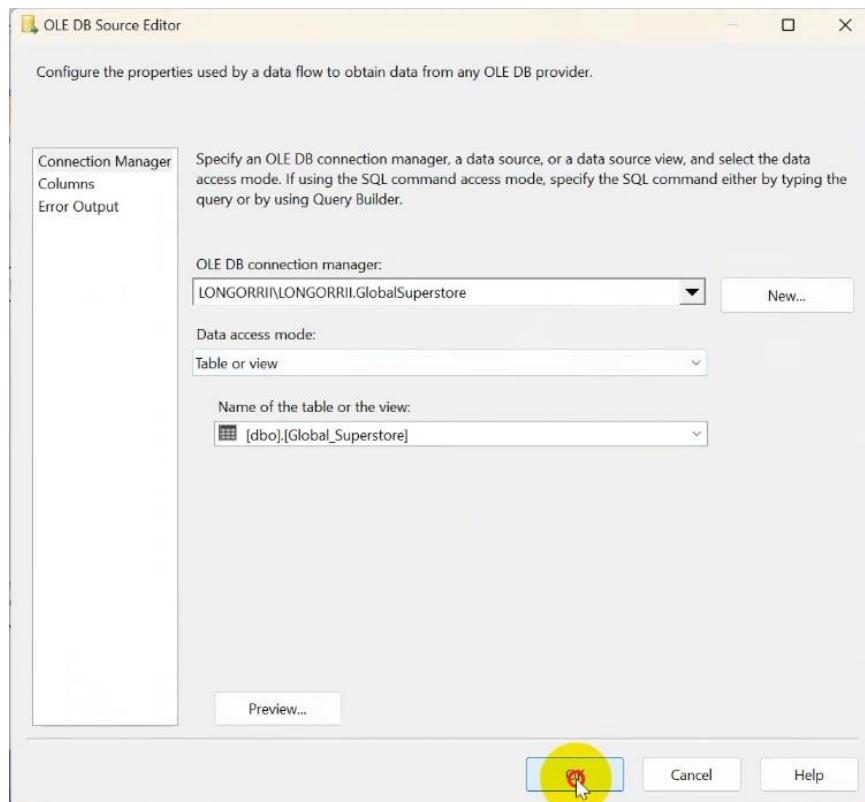


Hình 3.8: Data Flow Task “Dim\_Product”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

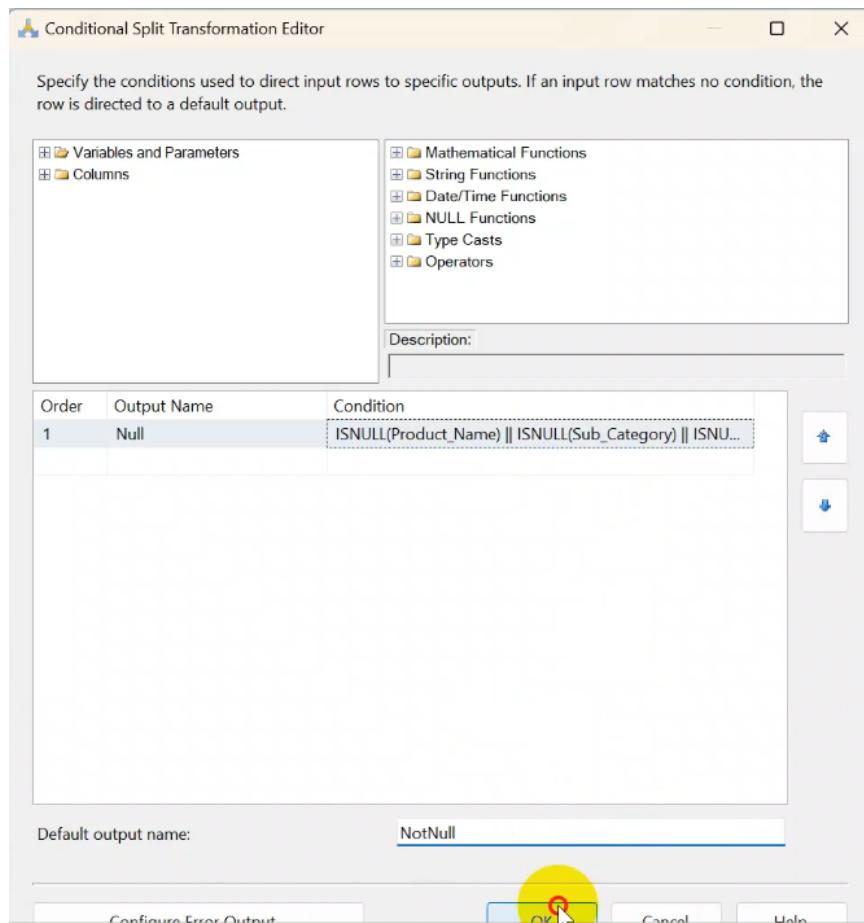
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORII\LONGORII.GlobalSuperstore.
  - **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.

- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Product\_Name) || ISNULL(Sub\_Category) || ISNULL(Category)
  - **Output Name:** Null
  - **Default output name:** NotNull

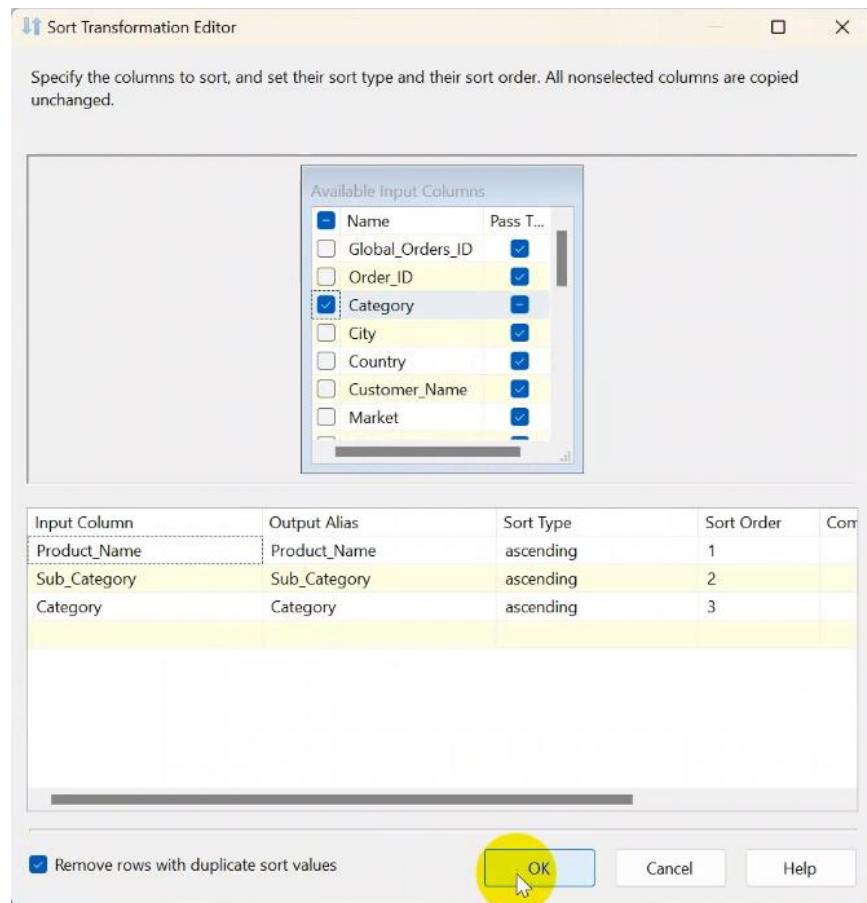


**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* Thiết lập Sort Transformation Editor:

- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_Product như sau:
  - Product\_Name
  - Sub\_Category
  - Category

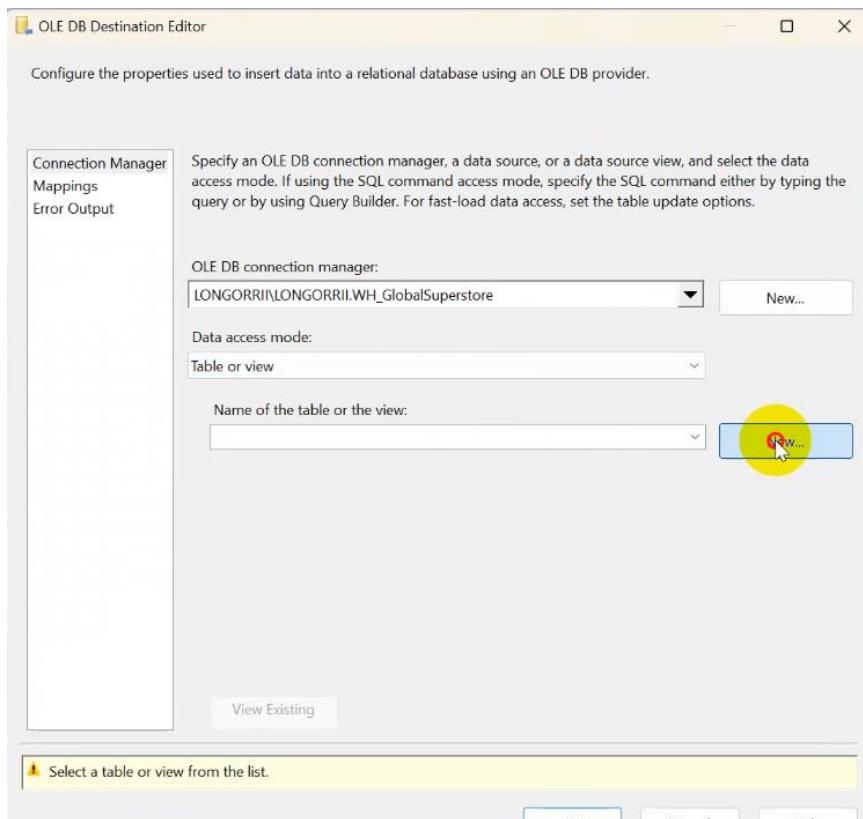
- Tick chọn “Remove rows with duplicate sort values” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**



**Bước 4: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “OLE DB Destination” và chọn **Edit** để đến giao diện “OLE DB Destination Editor”.

#### \* Thiết lập OLE DB Destination Editor

- Trong mục **Connection Manager**, ta chọn các thông số sau
  - **OLE DB connection manager:** chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORII\LONGORII.WH\_GlobalSuperstore.
  - **Data access mode:** chọn **Table or view**
  - **Name of the table or the view:** nhấn **New** để tạo bảng **Dim\_Product**



- Dán câu lệnh SQL sau để tạo bảng **Dim\_Product** và nhấn **OK**

**Note:** Mặc dù data gốc có thuộc tính Product\_ID nhưng chúng em không lấy thuộc tính này làm khóa chính bởi vì có 1 số dòng dữ liệu trùng Product\_ID nhưng Product\_Name khác nhau. Lý do có thể do một sản phẩm có thể đã thay đổi tên hoặc có các phiên bản khác nhau theo thời gian, nhưng vẫn giữ nguyên mã Product\_ID

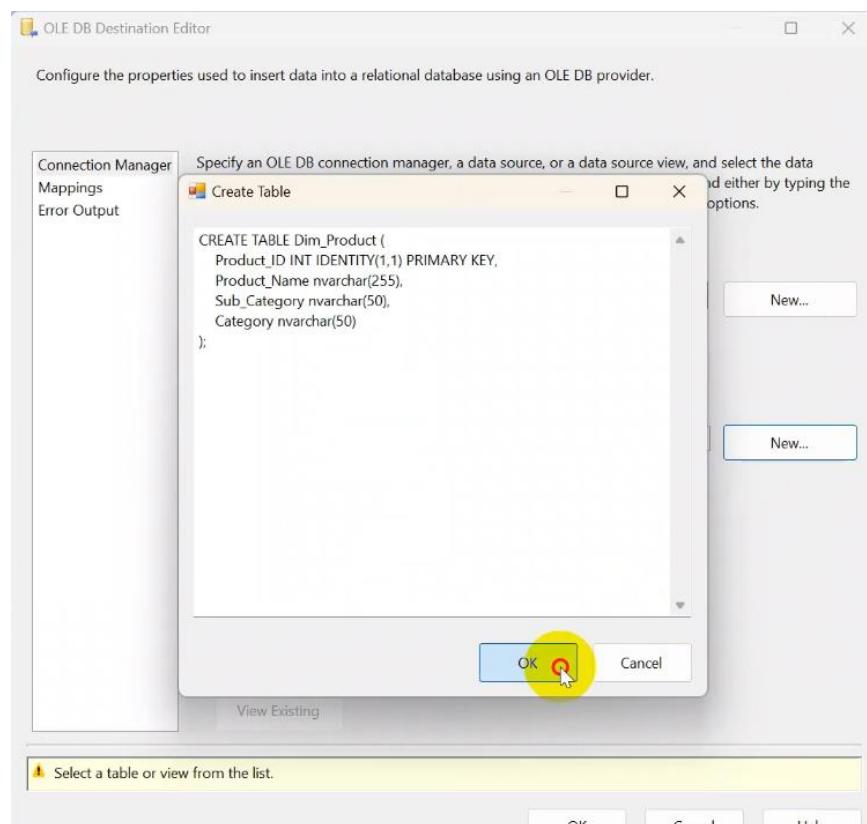
**Ví dụ:** Product\_ID “FUR-BO-10000087” có Product\_Name là “Dania Classic Bookcase, Mobile” ở 1 record nhưng lại có 1 record khác có Product\_Name là “Sauder Corner Shelving, Pine”

- Nên chúng em sẽ tạo thuộc tính Product\_ID mới sử dụng IDENTITY để giải quyết vấn đề này.

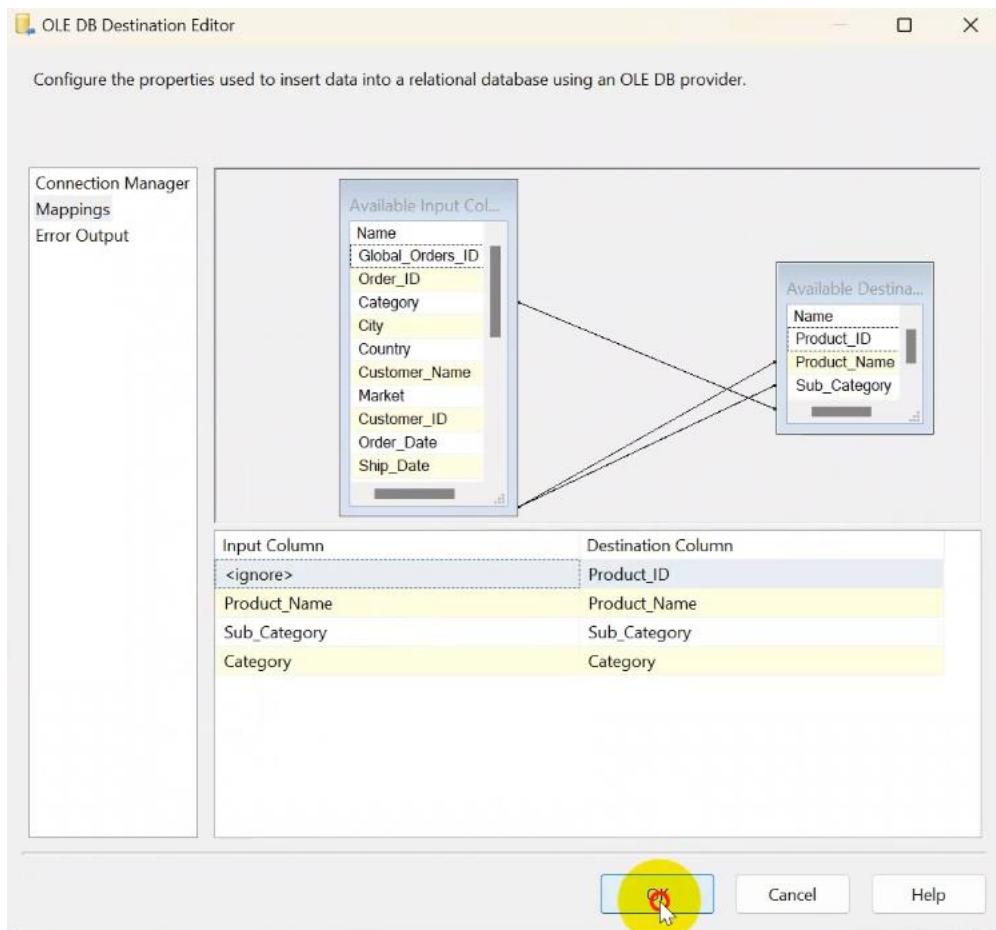
**Nội dung câu lệnh SQL:**

```
CREATE TABLE Dim_Product (
    Product_ID INT IDENTITY(1,1) PRIMARY KEY,
    Product_Name nvarchar(255),
```

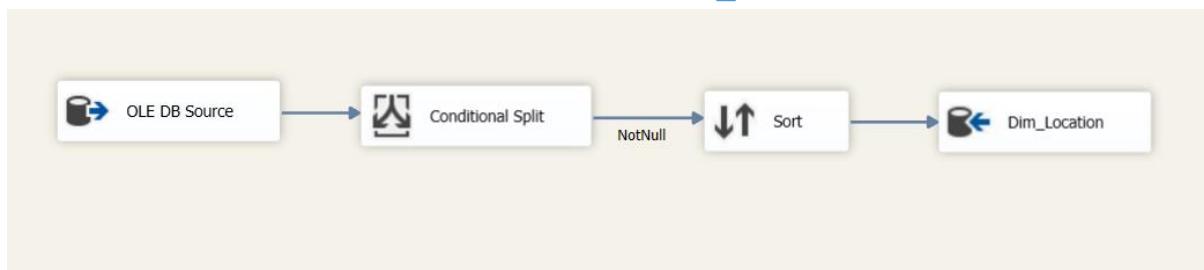
```
Sub_Catgeory nvarchar(50),  
Category nvarchar(50)  
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



#### 3.4.2.4 Cấu hình Data Flow Task “Dim\_Location”

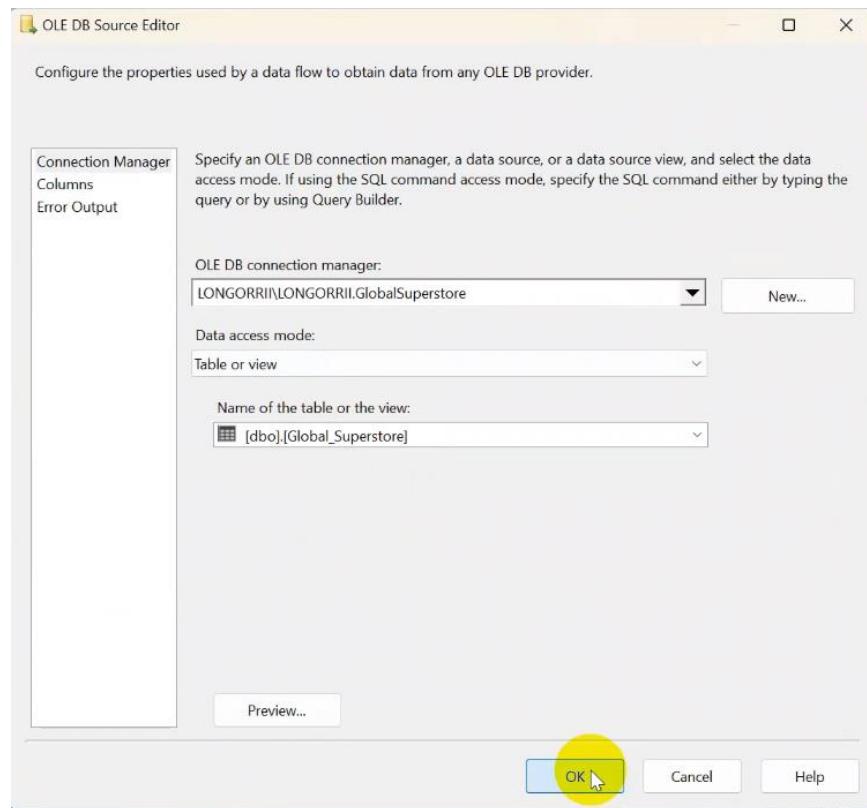


Hình 3.9: Cấu hình Data Flow Task “Dim\_Location”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn Edit để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

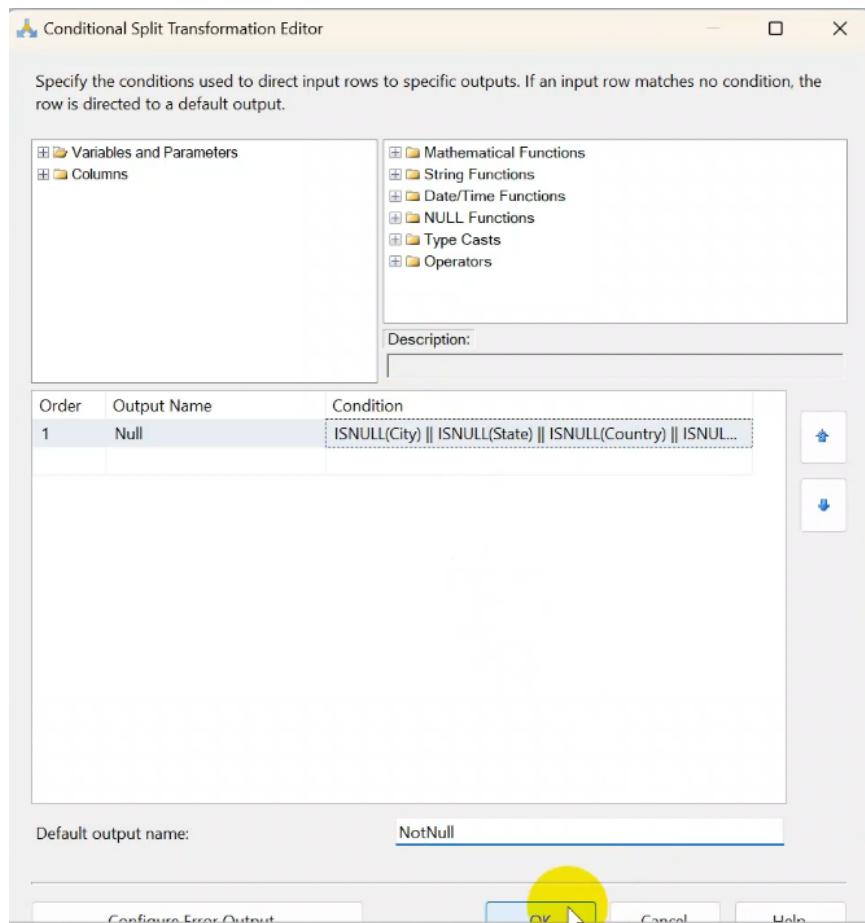
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - o **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORRI\LONGORRII.GlobalSuperstore.
  - o **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* **Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.**

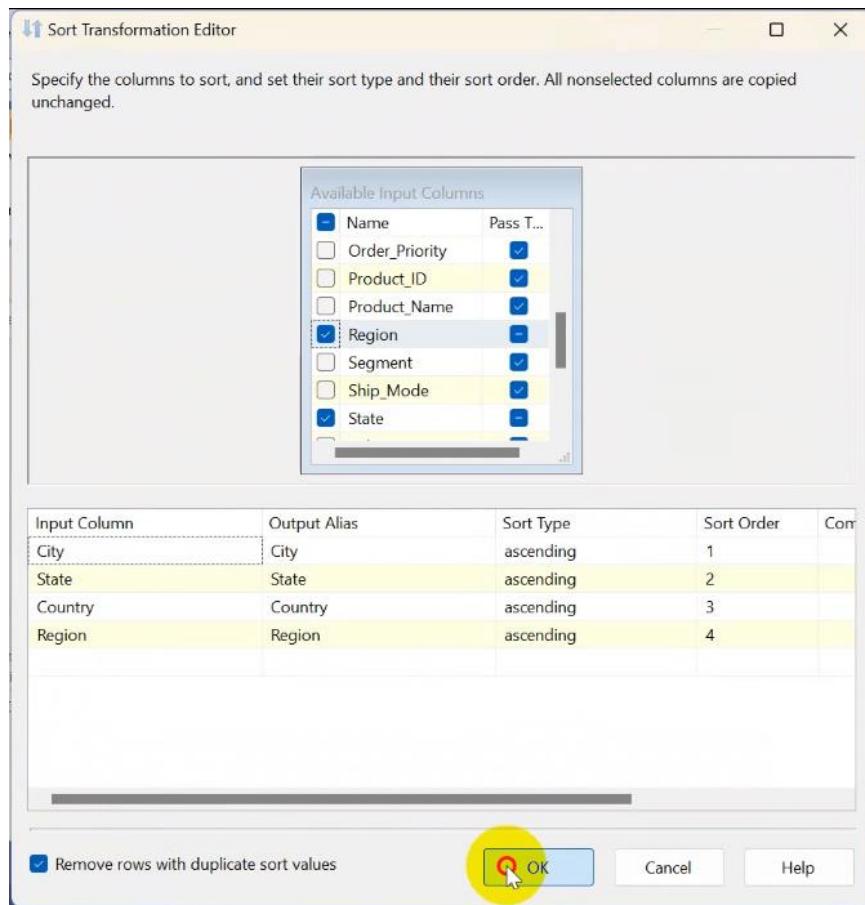
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(City) || ISNULL(State) || ISNULL(Country) || ISNULL(Region)
  - **Output Name:** Null
  - **Default output name:** NotNull



**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* **Thiết lập Sort Transformation Editor:**

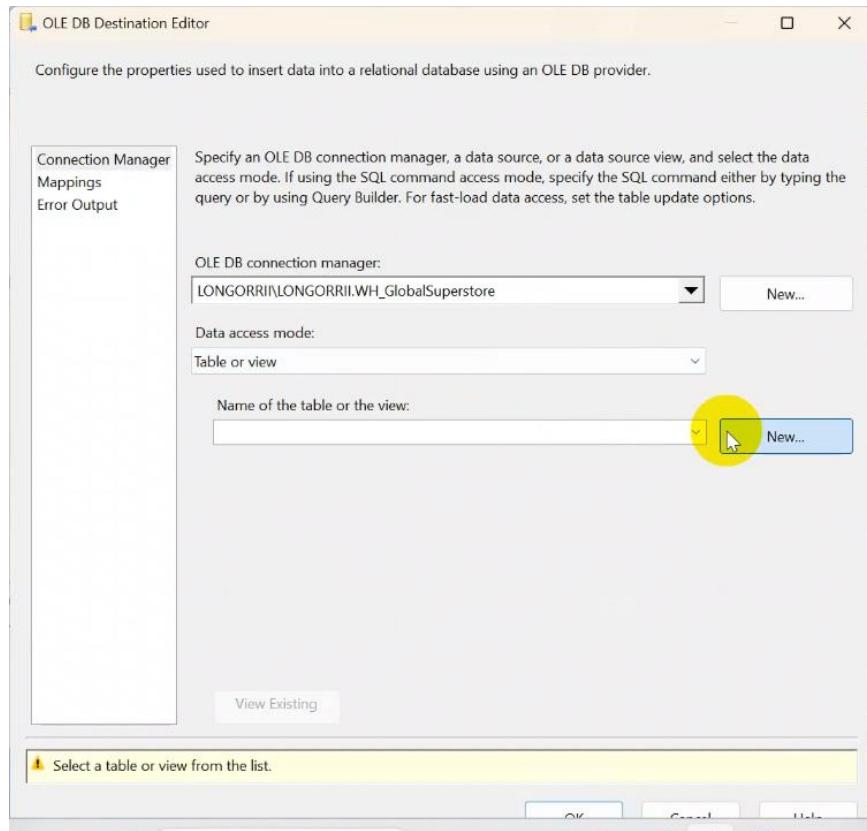
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_Location như sau:
  - o City
  - o State
  - o Country
  - o Region
- Tick chọn “Remove rows with duplicate sort values” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**.



**Bước 4: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “OLE DB Destination” và chọn Edit để đến giao diện “OLE DB Destination Editor”.

#### \* Thiết lập OLE DB Destination Editor

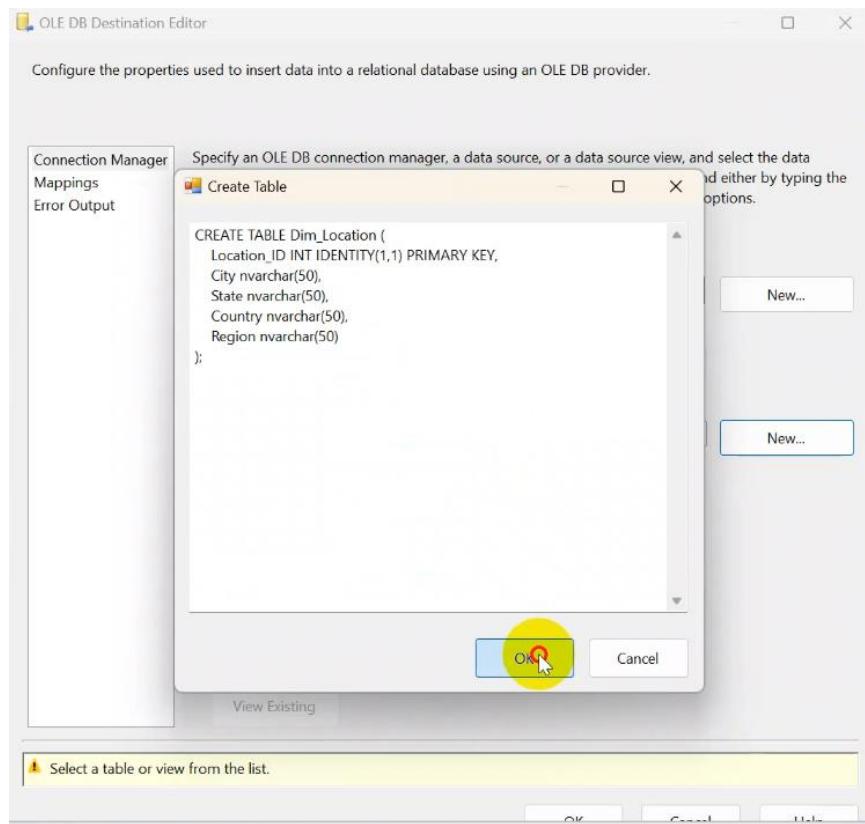
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORRII\LONGORRII.WH\_GlobalSuperstore.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_Location**



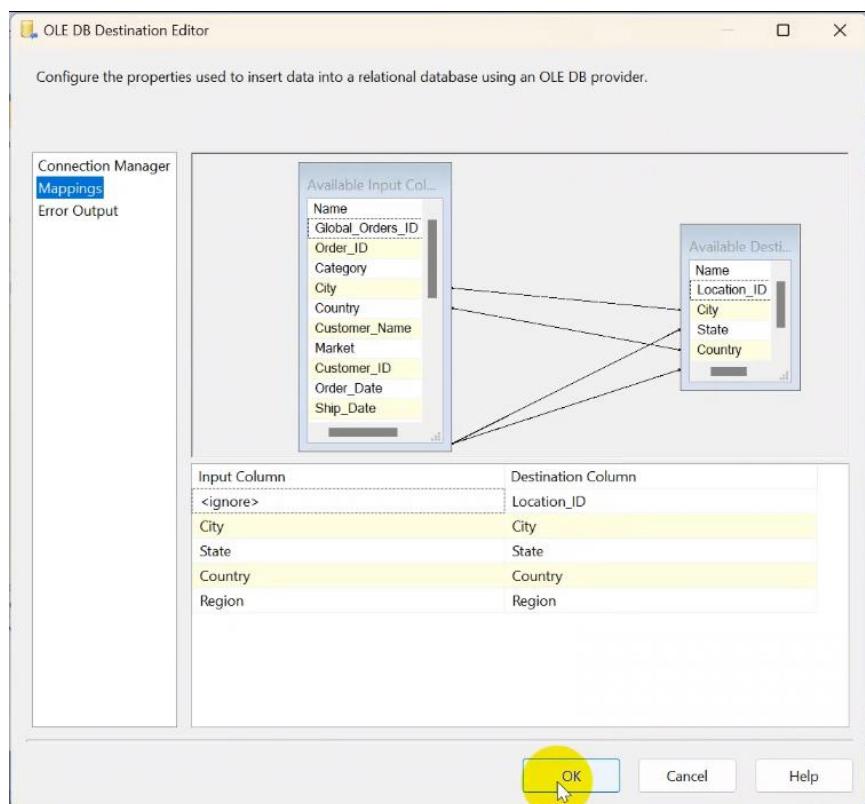
- Dán câu lệnh SQL sau để tạo bảng **Dim\_Location** và nhấn **OK**

**Nội dung câu lệnh SQL:**

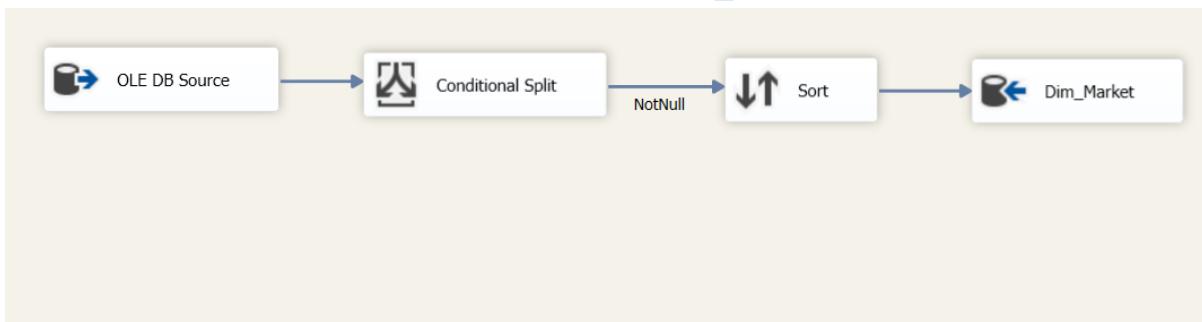
```
CREATE TABLE Dim_Location (
    Location_ID INT IDENTITY(1,1) PRIMARY KEY,
    City nvarchar(50),
    State nvarchar(50),
    Country nvarchar(50),
    Region nvarchar(50)
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



### 3.4.2.5 Cấu hình Data Flow Task “Dim\_Market”

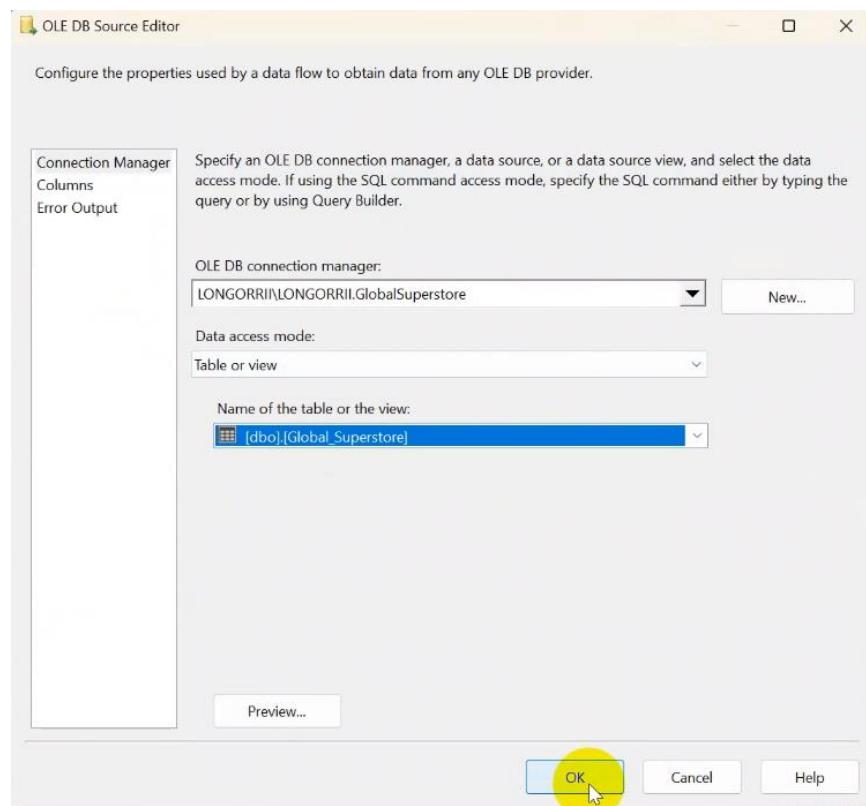


Hình 3.10: Cấu hình Data Flow Task “Dim\_Market”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

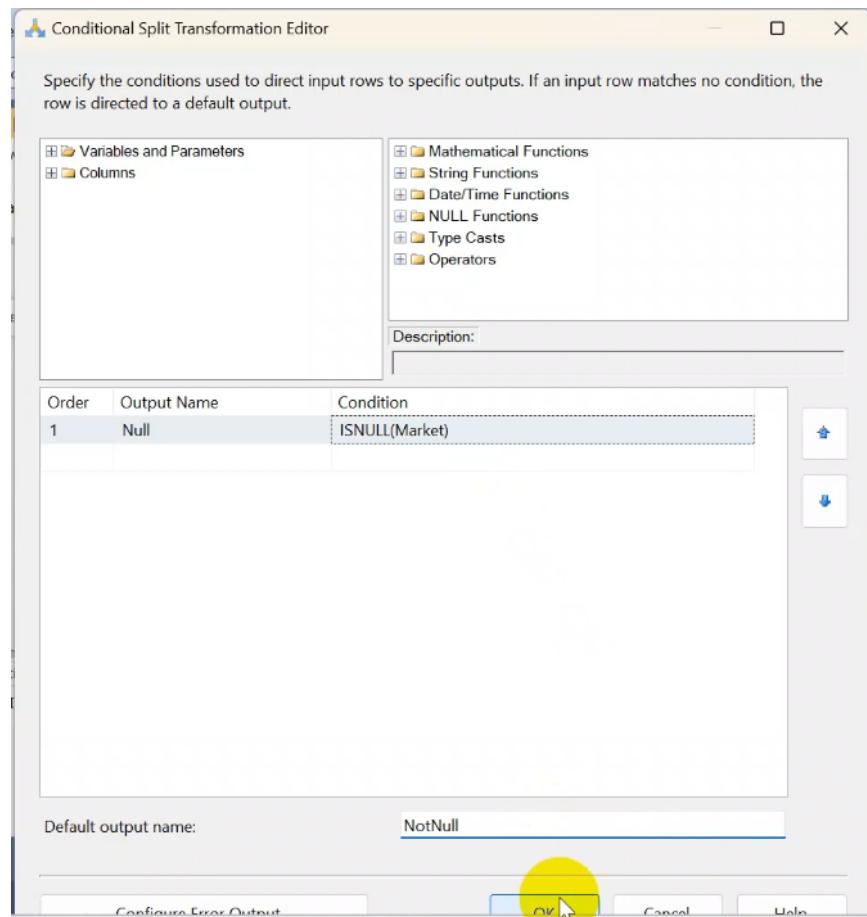
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORRI\LONGORRII.GlobalSuperstore.
  - **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.

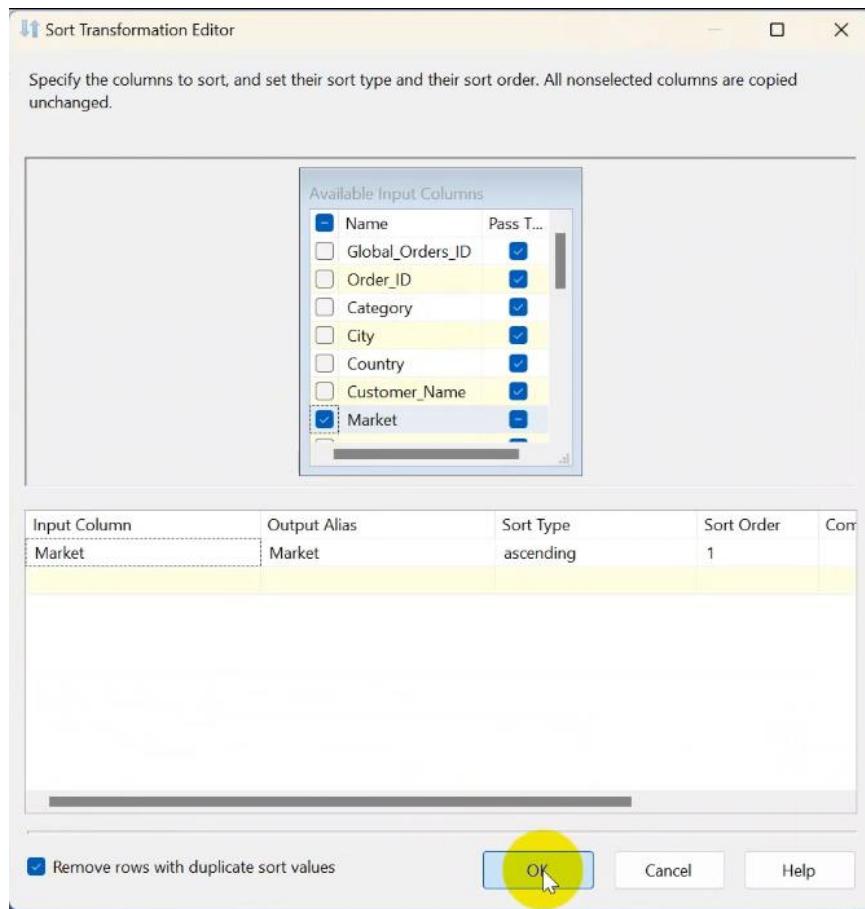
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Market)
  - **Output Name:** Null
  - **Default output name:** NotNull



**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* Thiết lập Sort Transformation Editor:

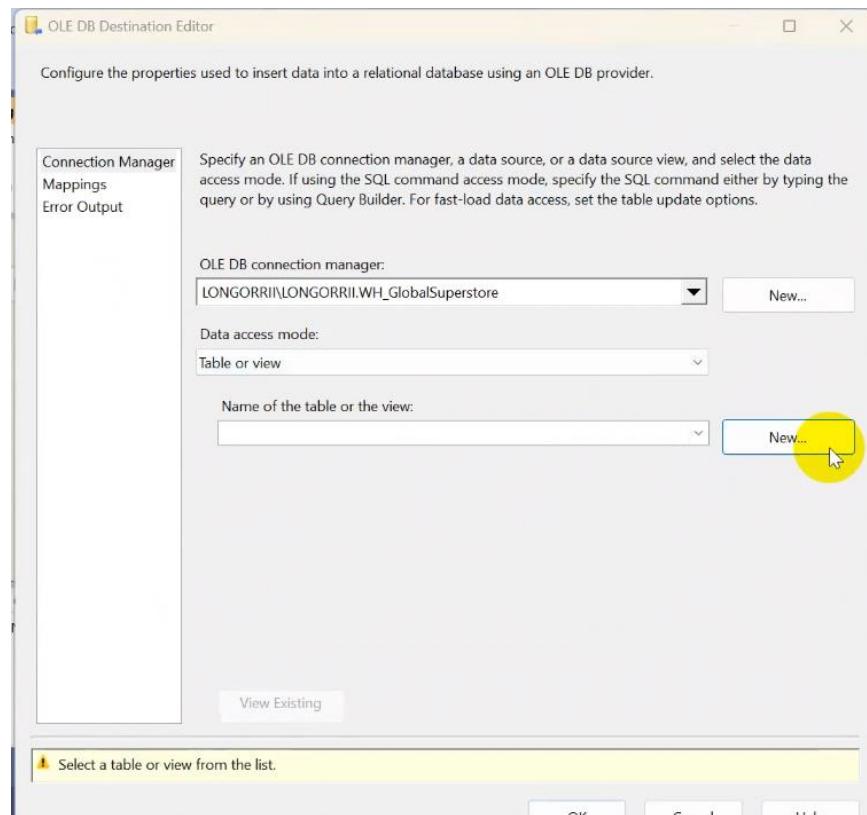
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_Market như sau:
  - Market
- Tick chọn “**Remove rows with duplicate sort values**” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**.



**Bước 4: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “**OLE DB Destination**” và chọn **Edit** để đến giao diện “**OLE DB Destination Editor**”.

#### \* Thiết lập OLE DB Destination Editor

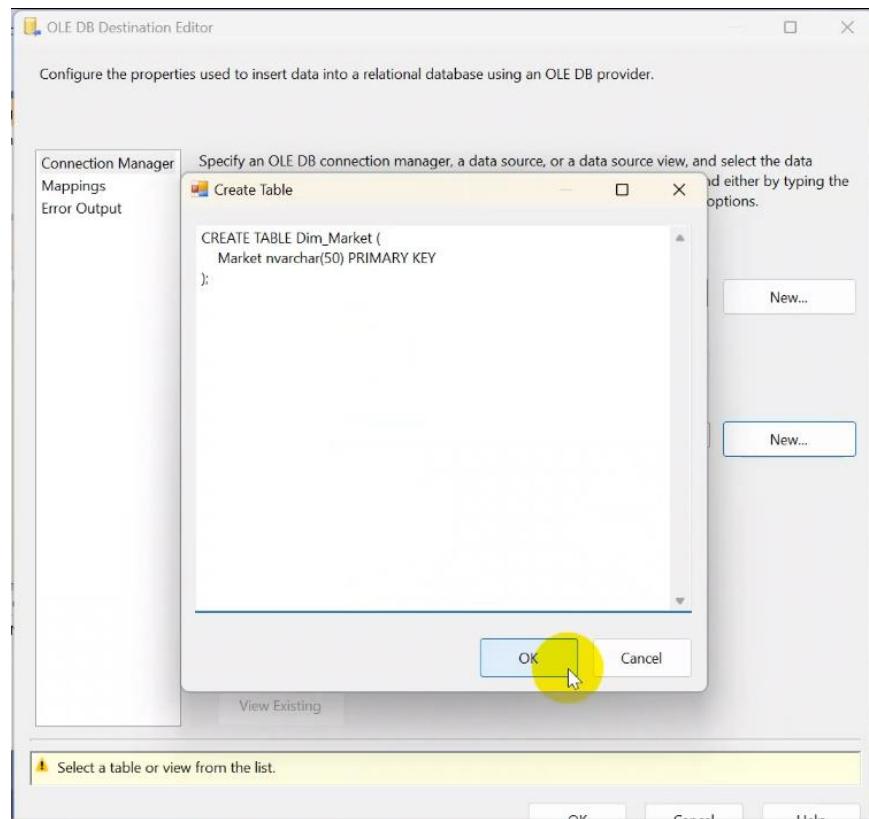
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORRII\LONGORRII.WH\_GlobalSuperstore.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_Market**



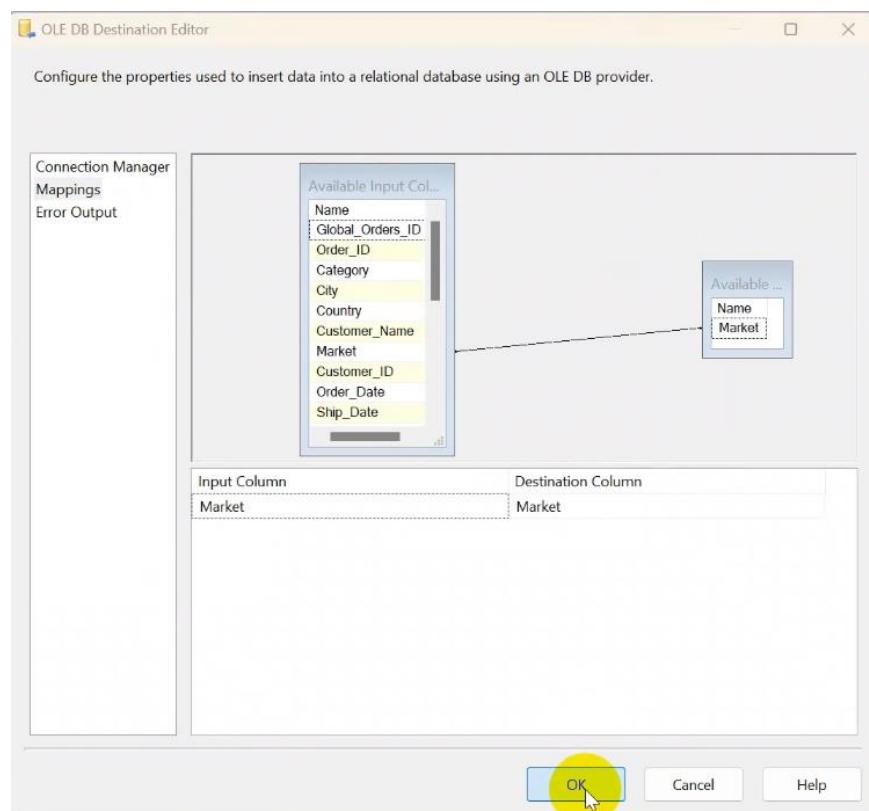
- Dán câu lệnh SQL sau để tạo bảng **Dim\_Market** và nhấn **OK**

**Nội dung câu lệnh SQL:**

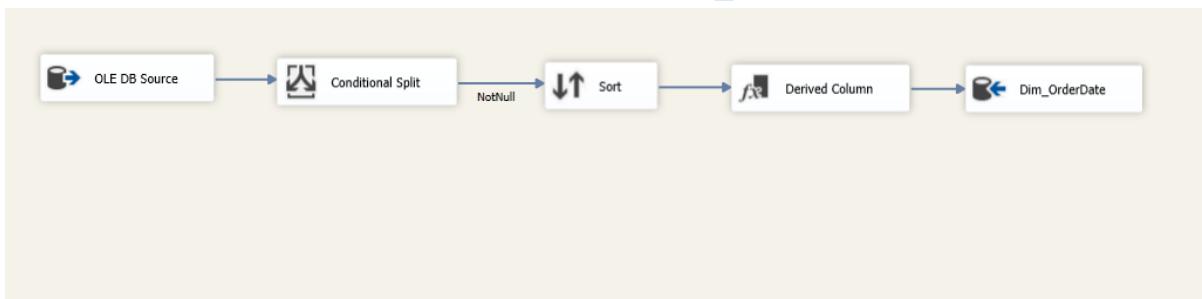
```
CREATE TABLE Dim_Market (
    Market nvarchar(50) PRIMARY KEY
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



### 3.4.2.6 Cấu hình Data Flow Task “Dim\_OrderDate”

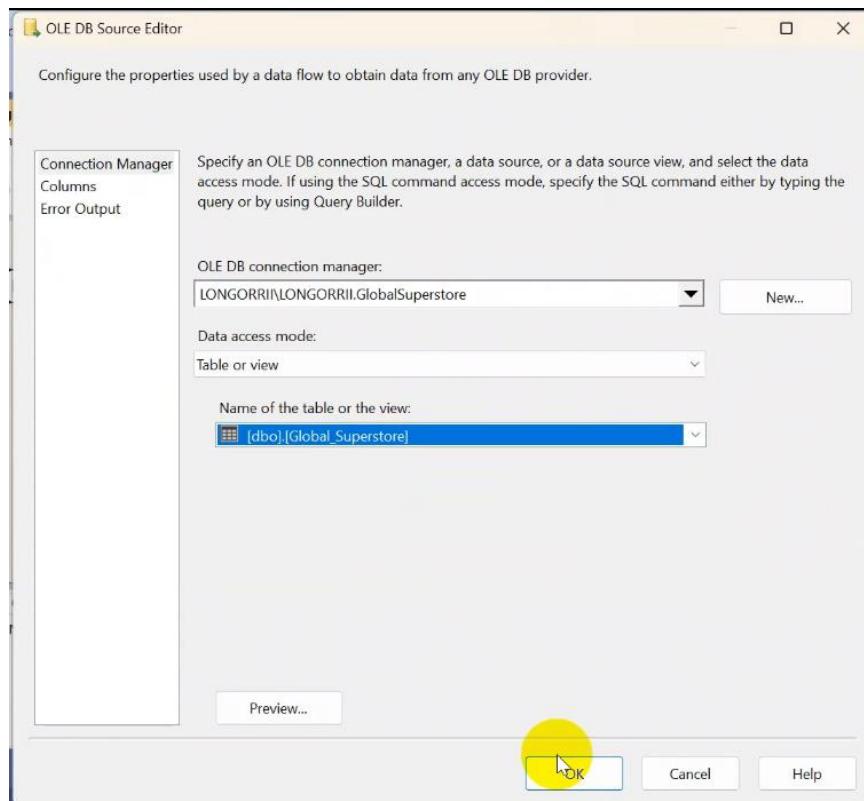


Hình 3.11: Cấu hình Data Flow Task “Dim\_OrderDate”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

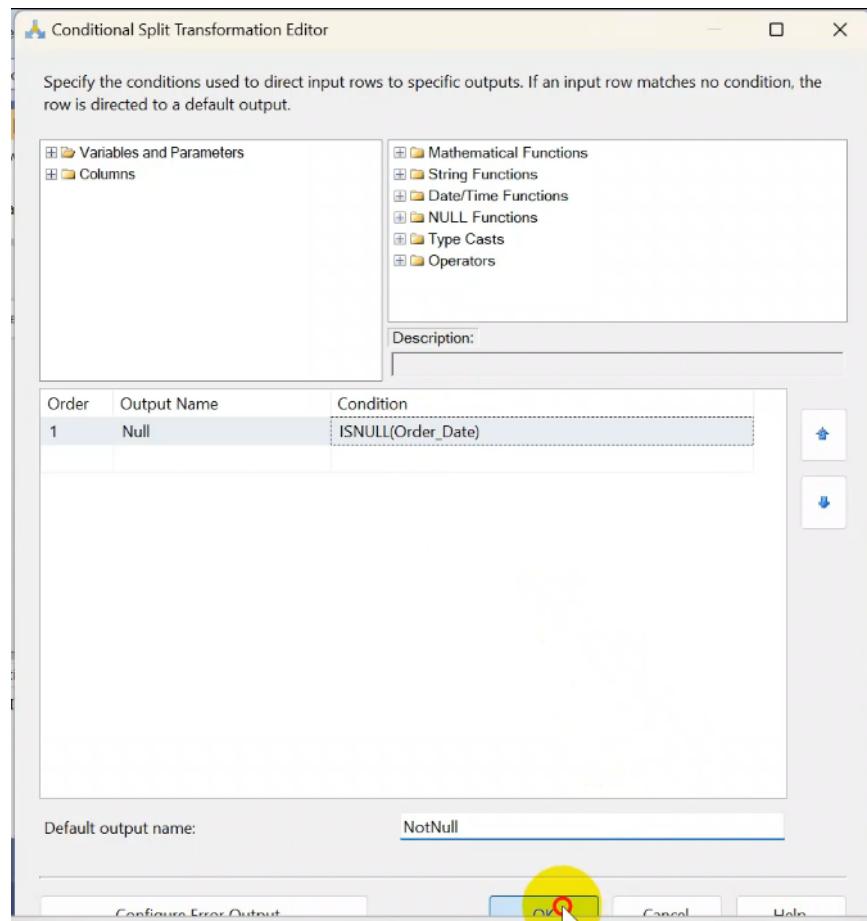
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORII\LONGORII.GlobalSuperstore.
  - **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.

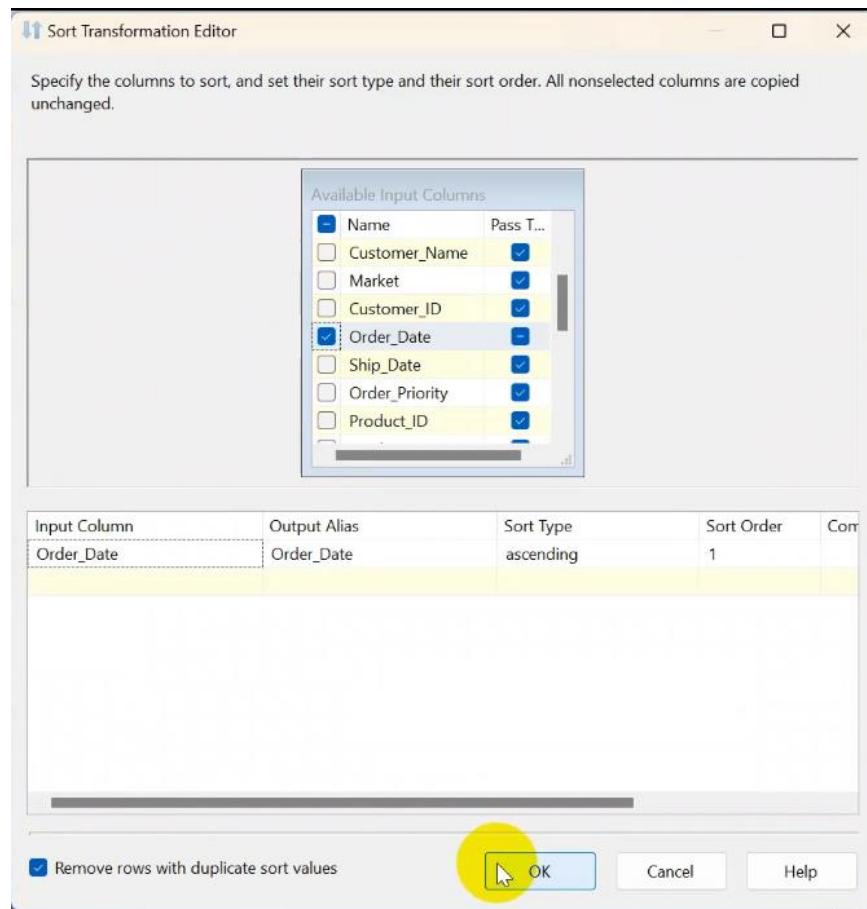
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Order\_Date)
  - **Output Name:** Null
  - **Default output name:** NotNull



Bước 3: Thiết lập Sort. Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* Thiết lập Sort Transformation Editor:

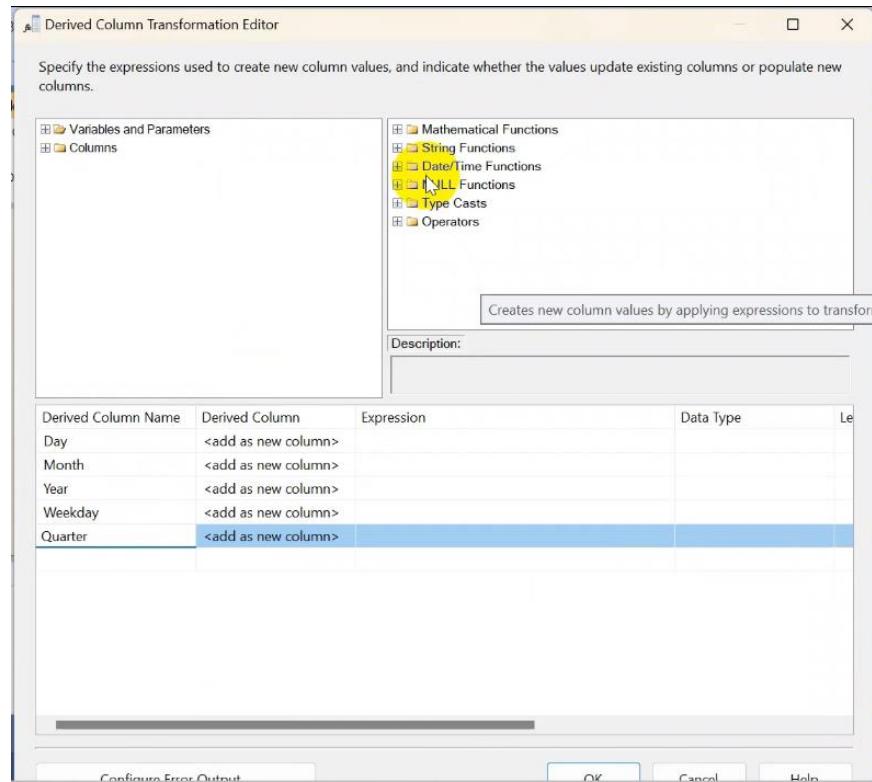
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_OrderDate như sau:
  - Order\_Date
- Tick chọn “**Remove rows with duplicate sort values**” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**.



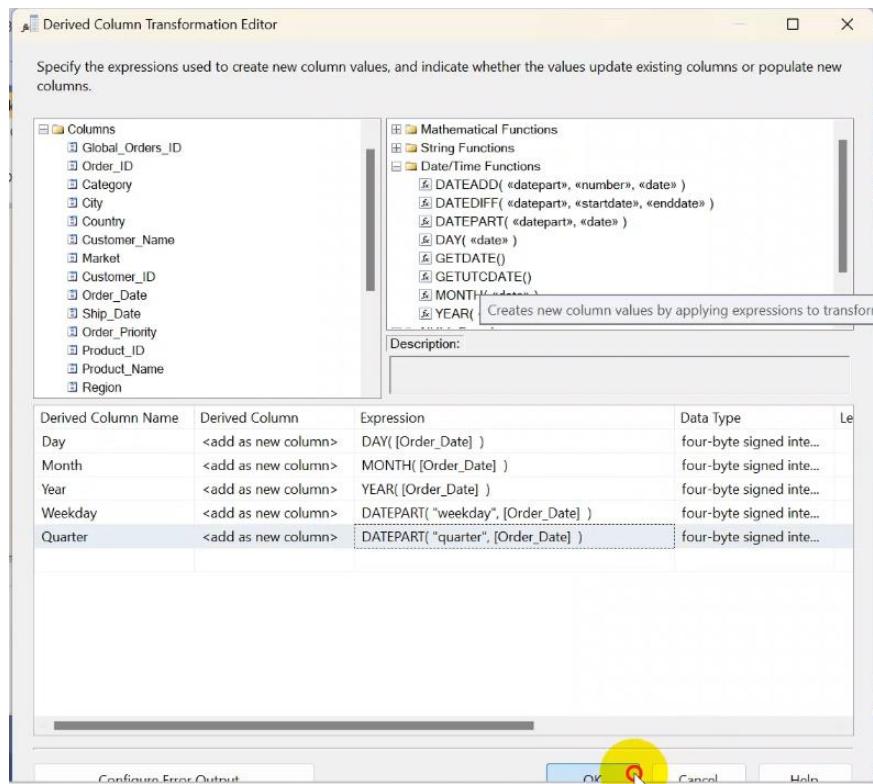
**Bước 4: Thiết lập Divered Column.** Nhấp chuột phải vào “Divered Column” và chọn **Edit** để đến giao diện “Divered Column Transformation Editor”.

#### \* Thiết lập Divered Column Transformation Editor

- Chia dữ liệu từ cột Order\_Date có kiểu dữ liệu dd/MM/yyyy thành các cột Day, Month, Year, Weekday, Quarter. Ở mục “Divered Column Name”. Ta điền tên các thuộc tính tạo mới



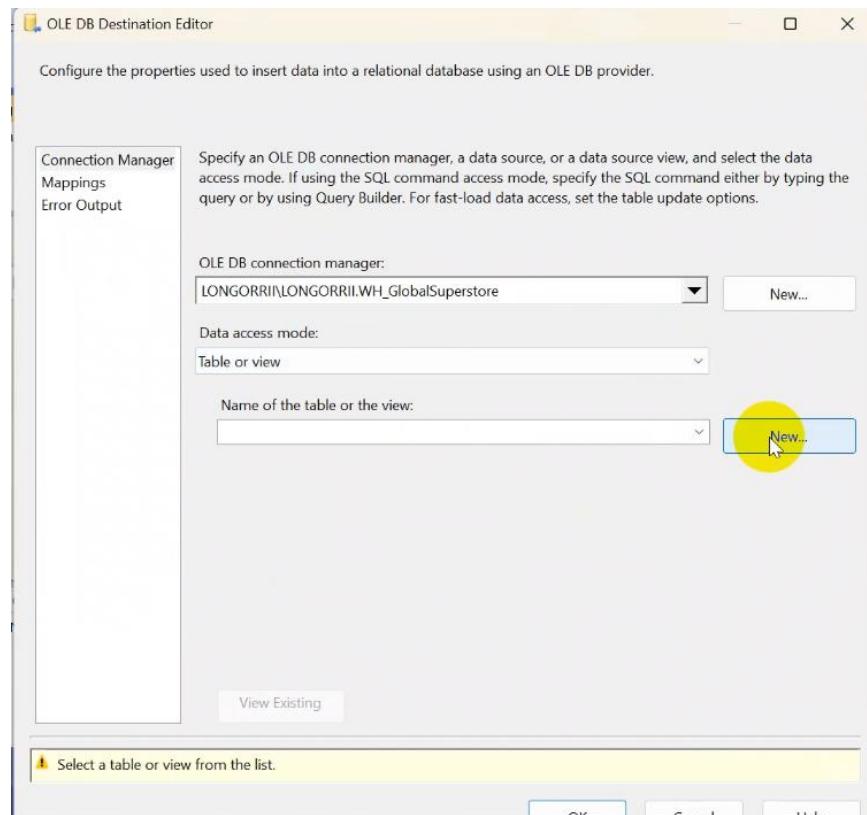
- Ở mục **Expression** ta kéo các hàm để chia dữ liệu tương ứng với mỗi thuộc tính như sau và nhấn **OK**:
  - o **Day:** DAY([Order\_Date])
  - o **Month:** MONTH([Order\_Date])
  - o **Year:** YEAR([Order\_Date])
  - o **Weekday:** DATEPART("weekday", [Order\_Date]).
  - o **Quarter:** DATEPART("quarter", [Order\_Date]).



**Bước 5: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “OLE DB Destination” và chọn **Edit** để đến giao diện “OLE DB Destination Editor”.

#### \* Thiết lập OLE DB Destination Editor

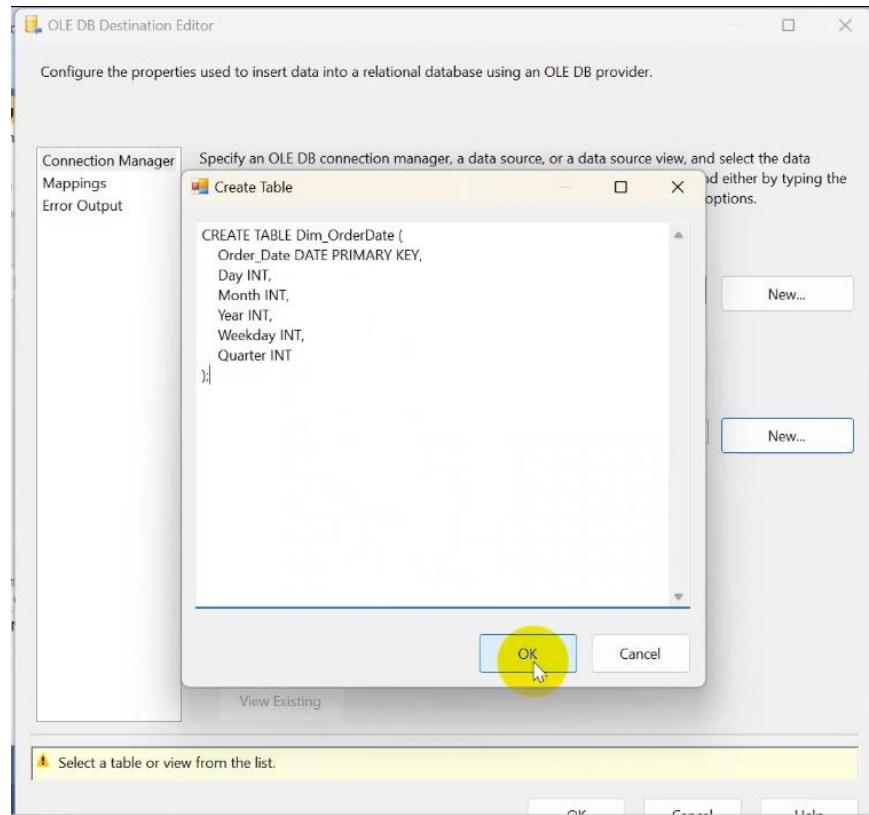
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORRII\LONGORRII.WH\_GlobalSuperstore.
  - **Data access mode**: chọn **Table or view**
  - **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_OrderDate**



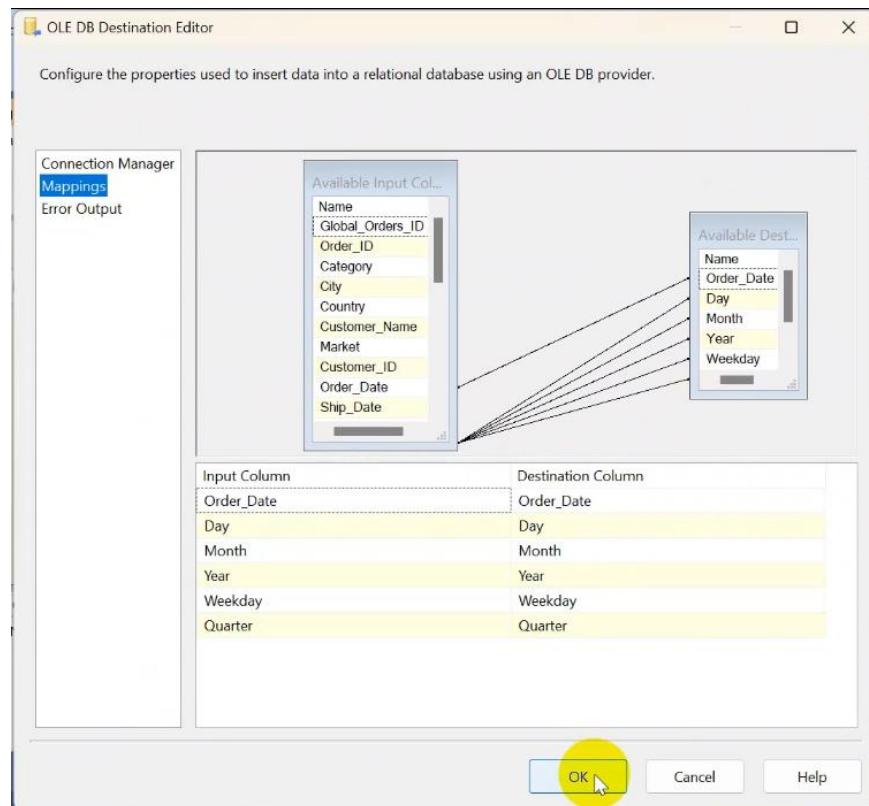
- Dán câu lệnh SQL sau để tạo bảng **Dim\_OrderDate** và nhấn **OK**

**Nội dung câu lệnh SQL:**

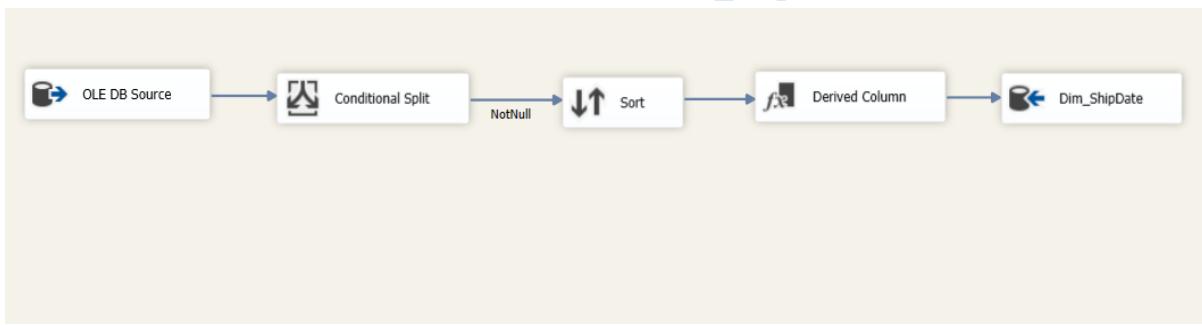
```
CREATE TABLE Dim_OrderDate (
    Order_Date DATE PRIMARY KEY,
    Day INT,
    Month INT,
    Year INT,
    Weekday INT,
    Quarter INT
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



### 3.4.2.7 Cấu hình Data Flow Task “Dim\_ShipDate”

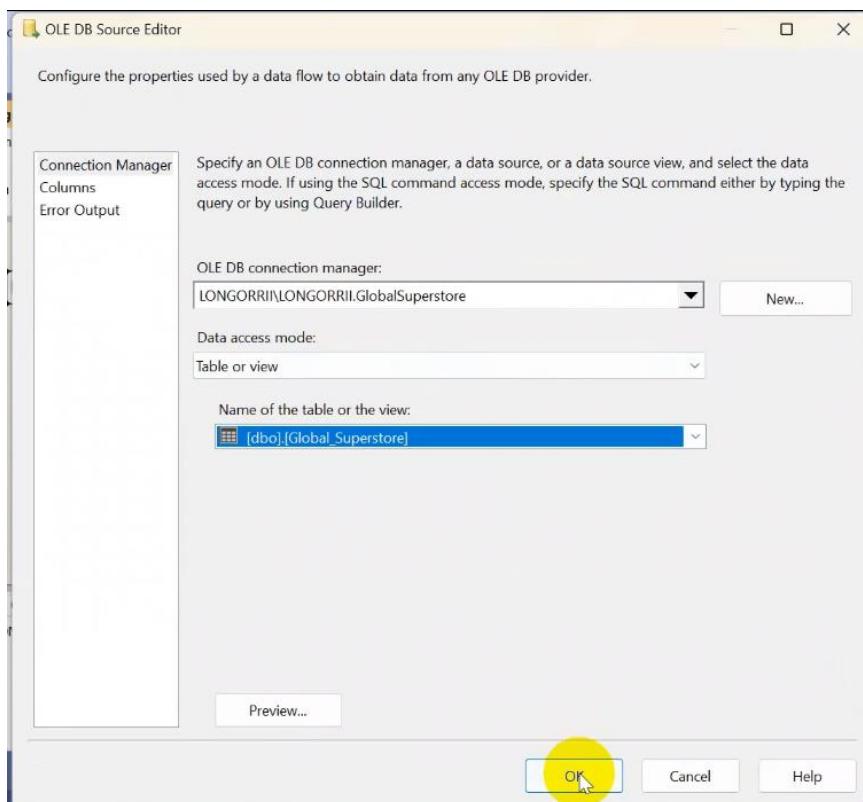


Hình 3.12: Cấu hình Data Flow Task “Dim\_ShipDate”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

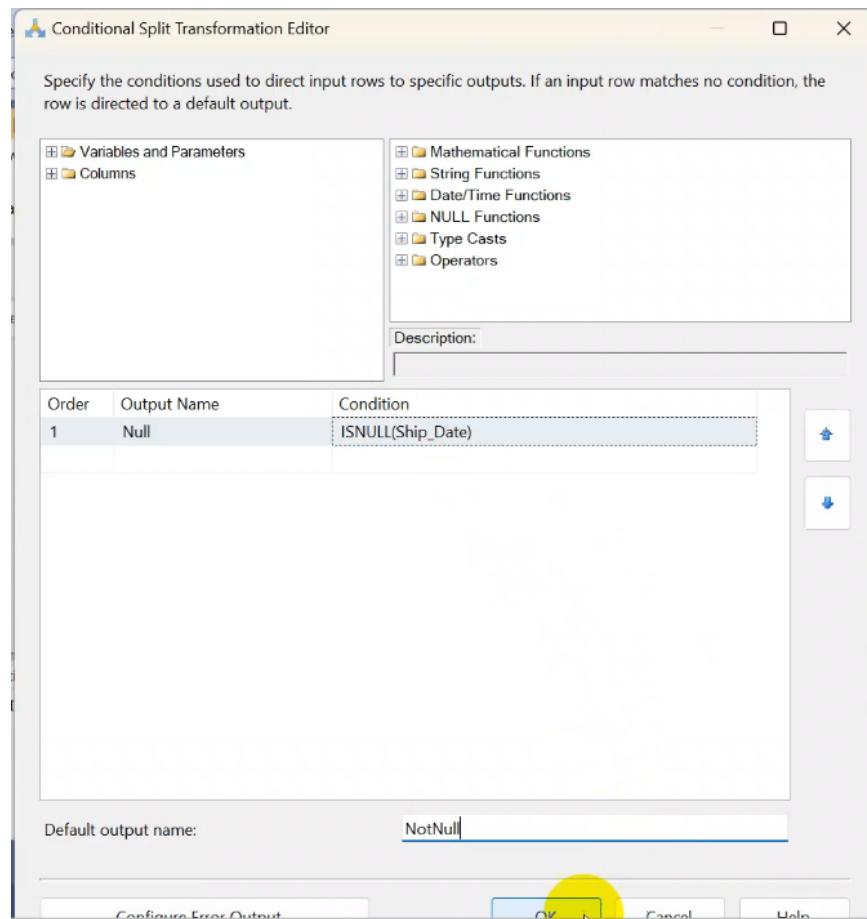
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - o **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORRII\LONGORRII.GlobalSuperstore.
  - o **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.

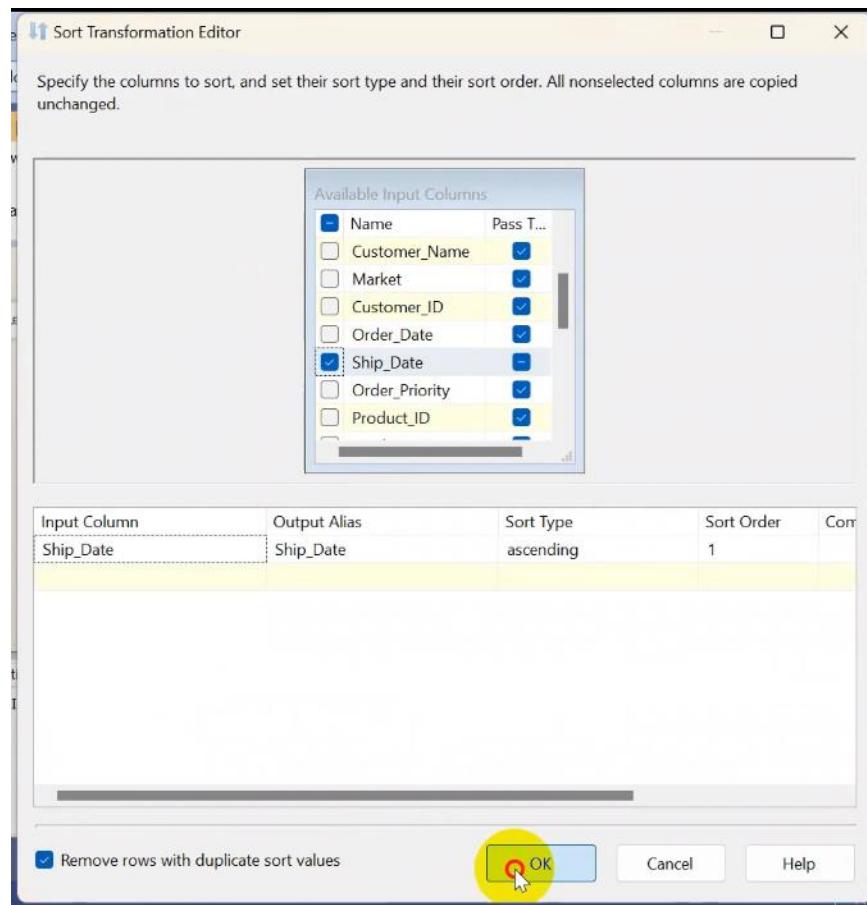
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Ship\_Date)
  - **Output Name:** Null
  - **Default output name:** NotNull



**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* Thiết lập Sort Transformation Editor:

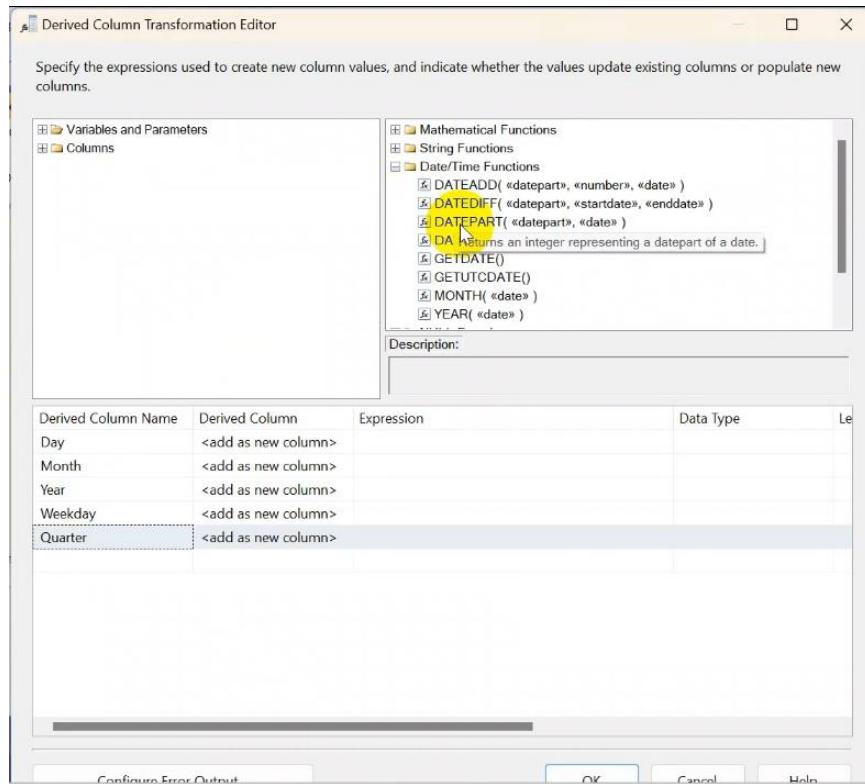
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_ShipDate như sau:
  - Ship\_Date
- Tick chọn “**Remove rows with duplicate sort values**” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**.



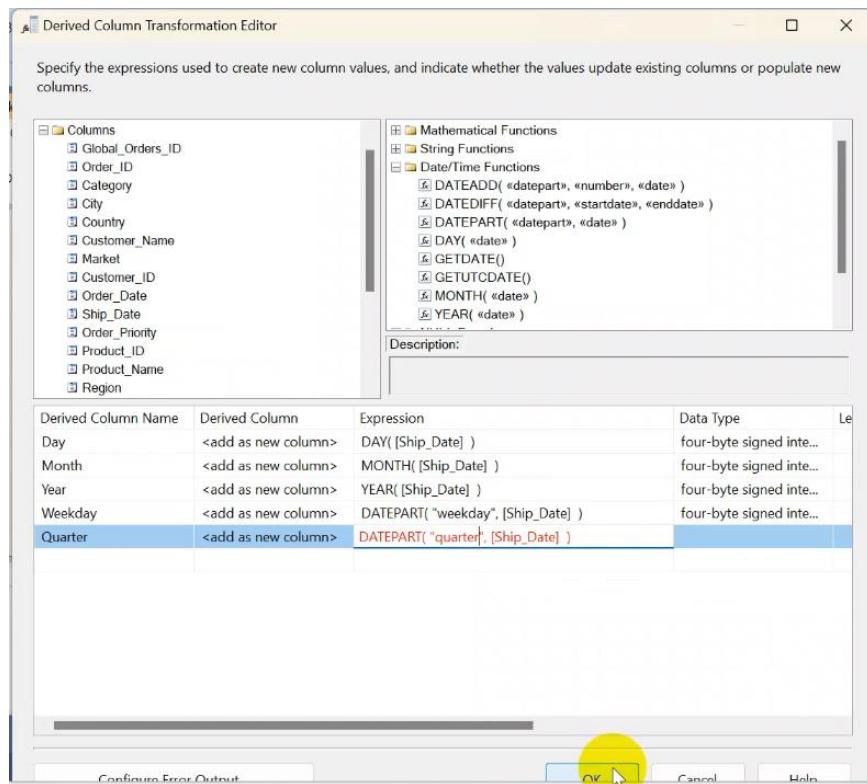
**Bước 4: Thiết lập Diveded Column.** Nhấp chuột phải vào “Diveded Column” và chọn Edit để đến giao diện “Diveded Column Transformation Editor”.

#### \* Thiết lập Diveded Column Transformation Editor

- Chia dữ liệu từ cột Ship\_Date có kiểu dữ liệu dd/MM/yyyy thành các cột Day, Month, Year, Weekday, Quarter. Ở mục “Diveded Column Name”. Ta điền tên các thuộc tính tạo mới



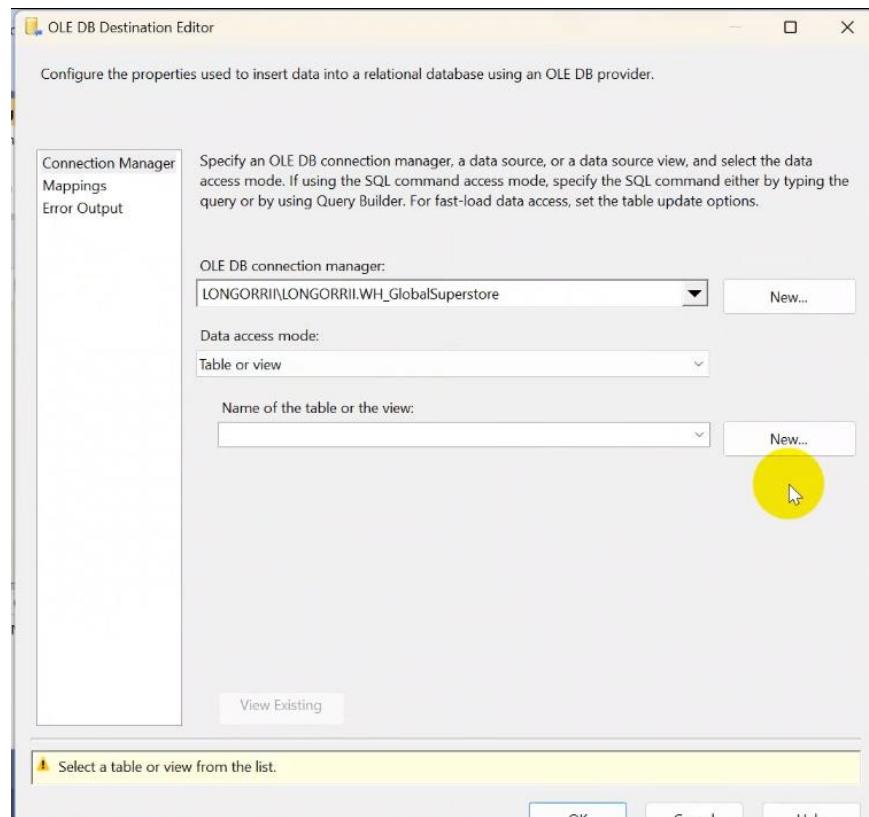
- Ở mục **Expression** ta kéo các hàm để chia dữ liệu tương ứng với mỗi thuộc tính như sau và nhấn **OK**:
  - o **Day:** DAY([Ship\_Date])
  - o **Month:** MONTH([Ship\_Date])
  - o **Year:** YEAR([Ship\_Date])
  - o **Weekday:** DATEPART("weekday", [Ship\_Date]).
  - o **Quarter:** DATEPART("quarter", [Ship\_Date]).



**Bước 5: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “OLE DB Destination” và chọn **Edit** để đến giao diện “OLE DB Destination Editor”.

#### \* Thiết lập OLE DB Destination Editor

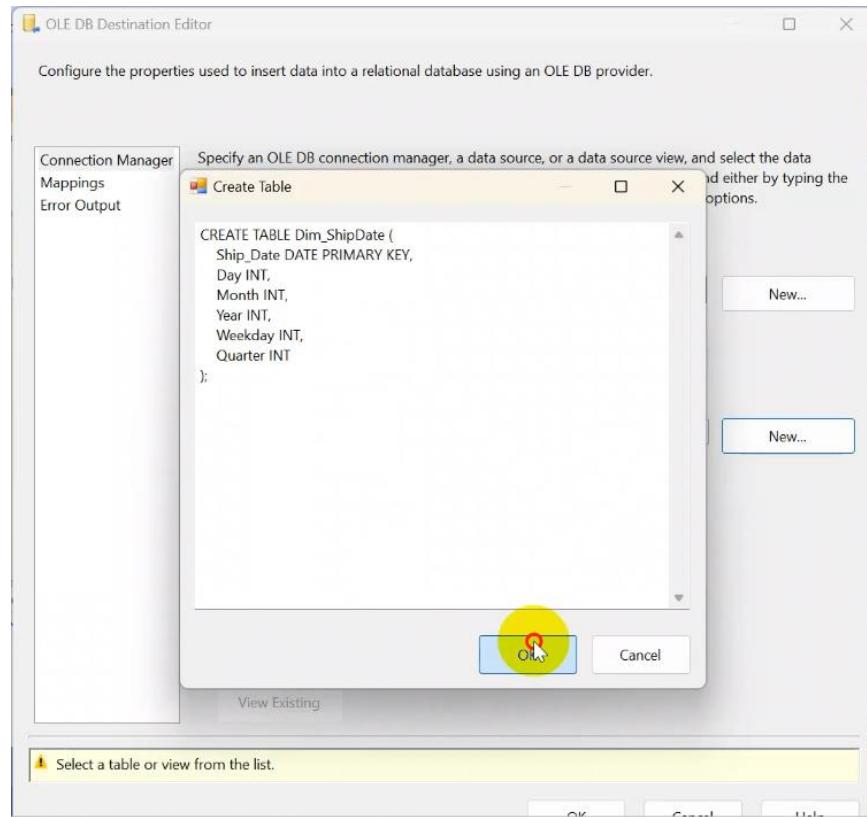
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORRII\LONGORRII.WH\_GlobalSuperstore.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_ShipDate**



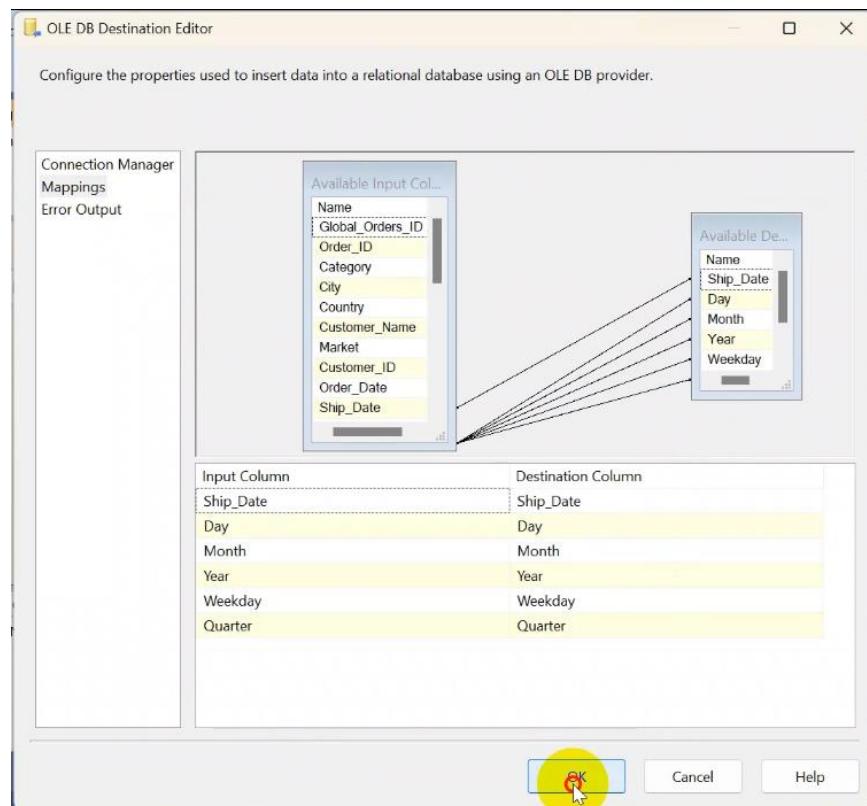
- Dán câu lệnh SQL sau để tạo bảng **Dim\_ShipDate** và nhấn **OK**

**Nội dung câu lệnh SQL:**

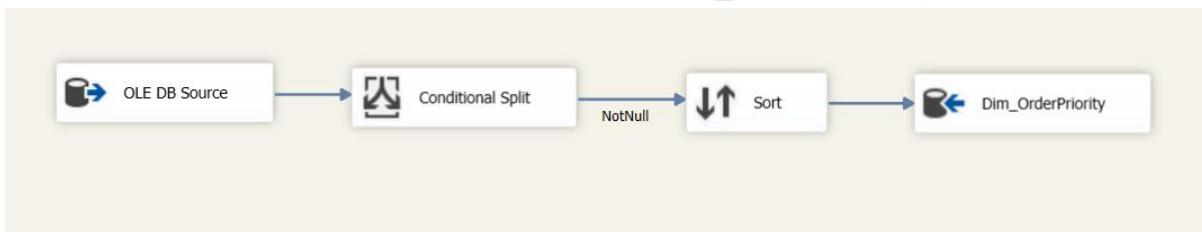
```
CREATE TABLE Dim_ShipDate (
    Ship_Date DATE PRIMARY KEY,
    Day INT,
    Month INT,
    Year INT,
    Weekday INT,
    Quarter INT
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



### 3.4.2.8 Cấu hình Data Flow Task “Dim\_OrderPriority”

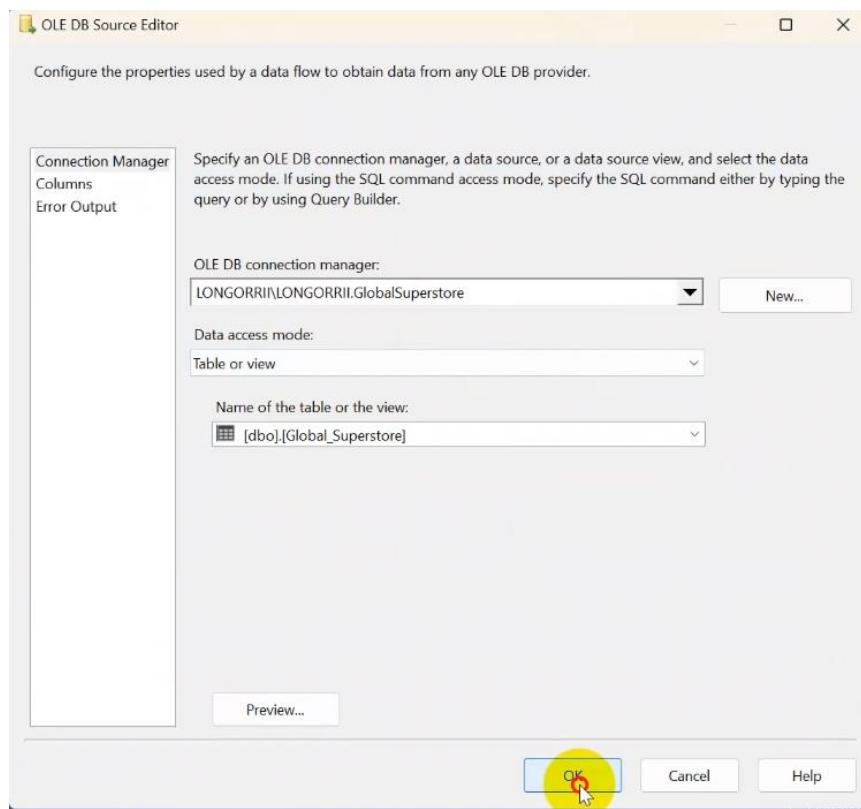


Hình 3.13: Cấu hình Data Flow Task “Dim\_OrderPriority”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

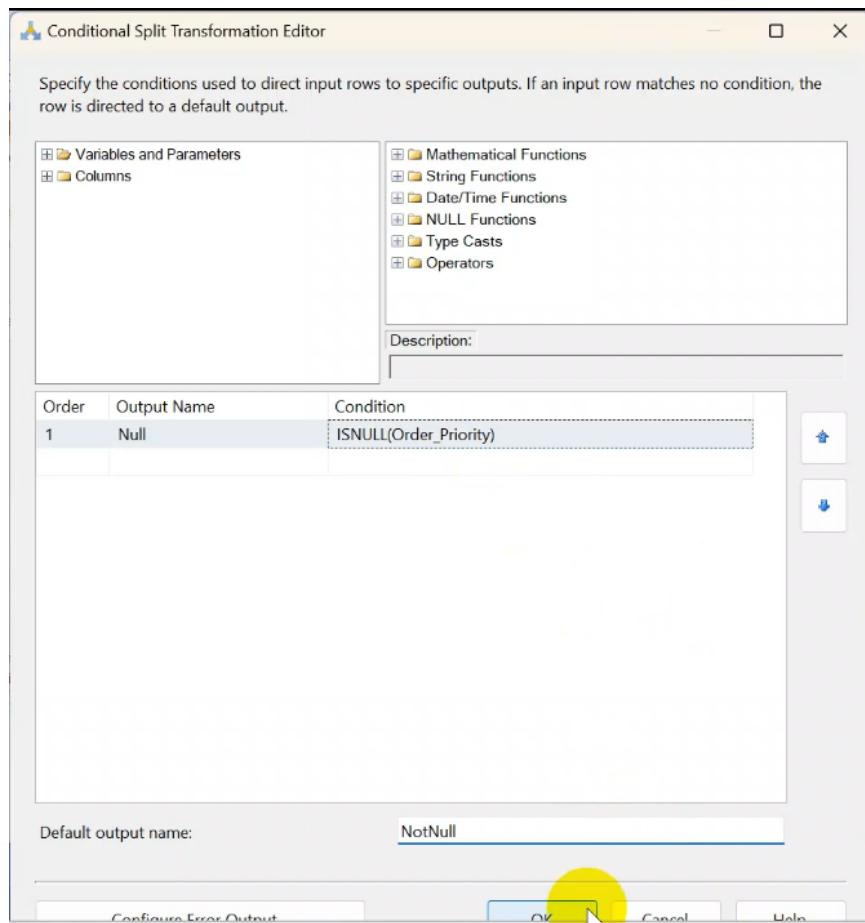
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - o **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORII\LONGORII.GlobalSuperstore.
  - o **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.

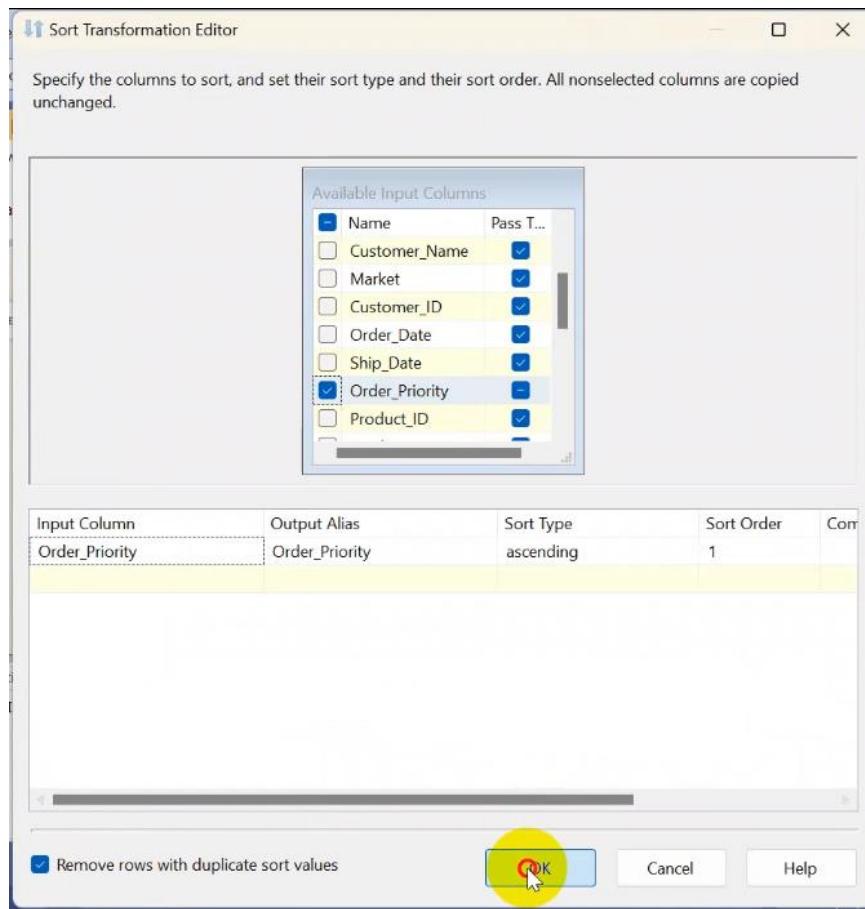
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Order\_Priority)
  - **Output Name:** Null
  - **Default output name:** NotNull



**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* Thiết lập Sort Transformation Editor:

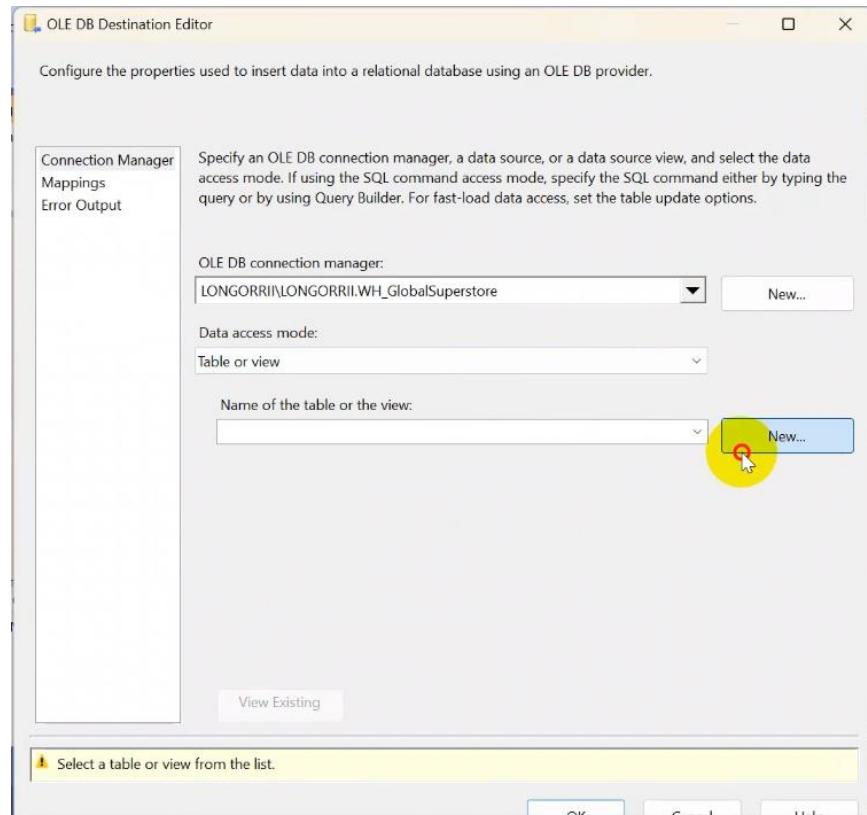
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_OrderPriority như sau:
  - Order\_Priority
- Tick chọn “**Remove rows with duplicate sort values**” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**.



**Bước 4: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “**OLE DB Destination**” và chọn **Edit** để đến giao diện “**OLE DB Destination Editor**”.

#### \* Thiết lập OLE DB Destination Editor

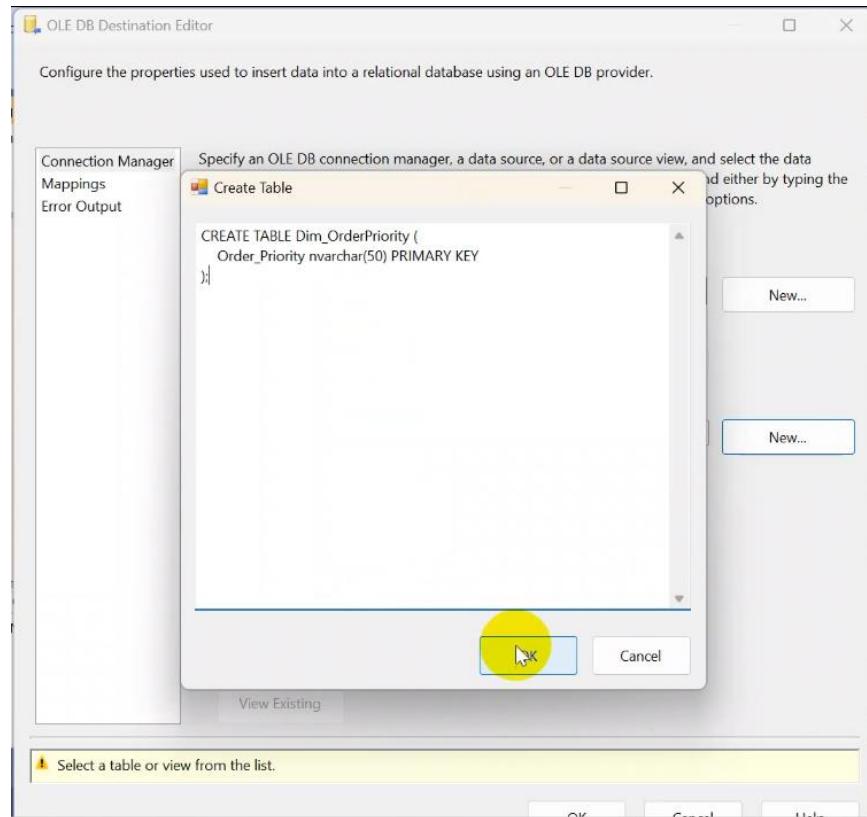
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORRII\LONGORRII.WH\_GlobalSuperstore.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_OrderPriority**



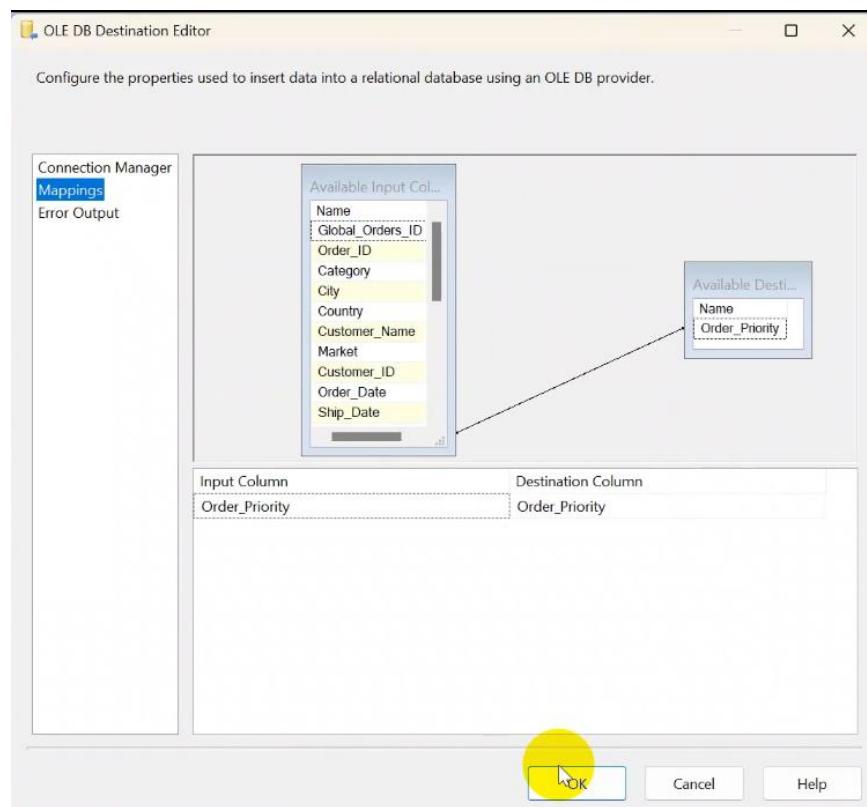
- Dán câu lệnh SQL sau để tạo bảng **Dim\_OrderPriority** và nhấn **OK**

**Nội dung câu lệnh SQL:**

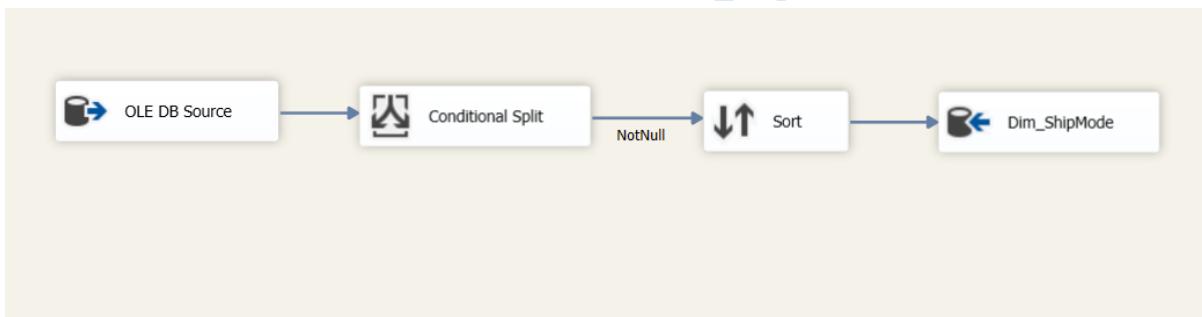
```
CREATE TABLE Dim_OrderPriority (
    Order_Priority nvarchar(50) PRIMARY KEY
);
```



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



### 3.4.2.9 Cấu hình Data Flow Task “Dim\_ShipMode”

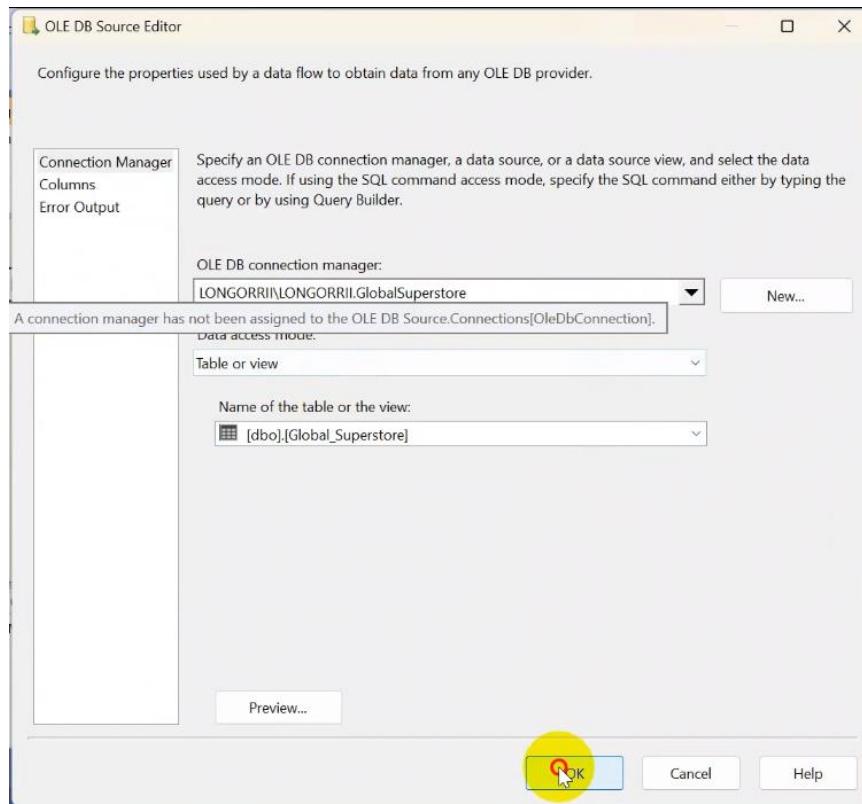


Hình 3.14: Cấu hình Data Flow Task “Dim\_ShipMode”

**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

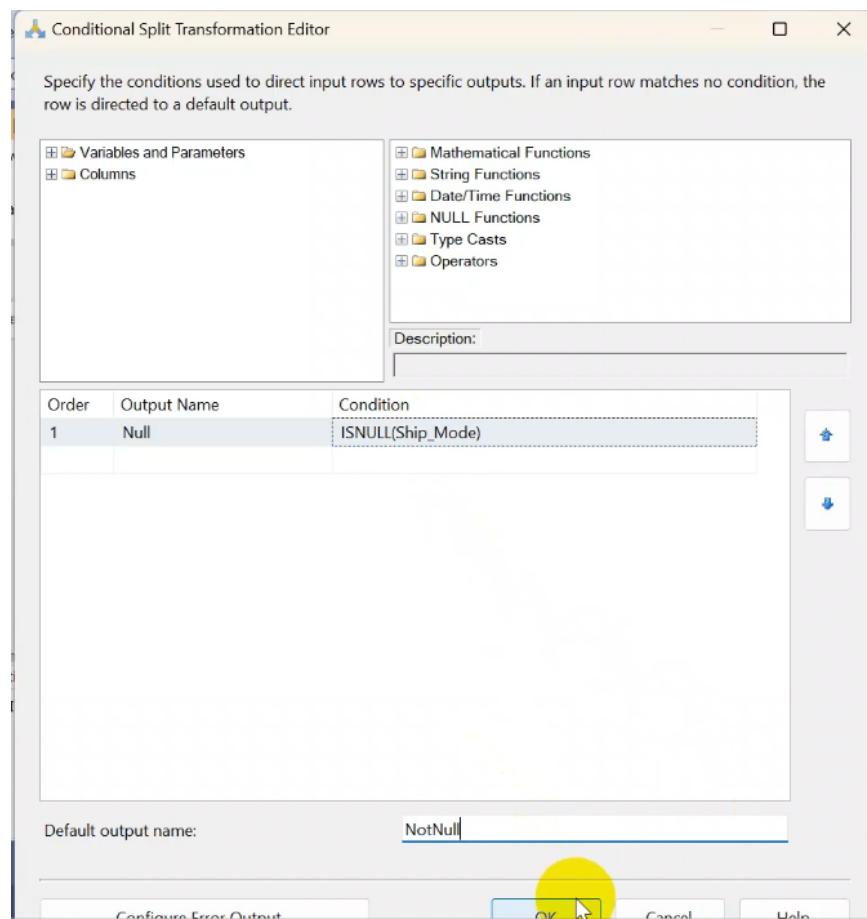
- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - o **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORII\LONGORII.GlobalSuperstore.
  - o **Name of the table or the view:** [dbo].[ GlobalSuperstore].



**Bước 2: Thiết lập Conditional Split.** Nhấp chuột phải vào “Conditional Split” và chọn **Edit** để đến giao diện “Conditional Split Transformation Editor”.

\* Thiết lập Conditional Split Transformation Editor: mục đích nhằm loại bỏ các dòng chứa dữ liệu null.

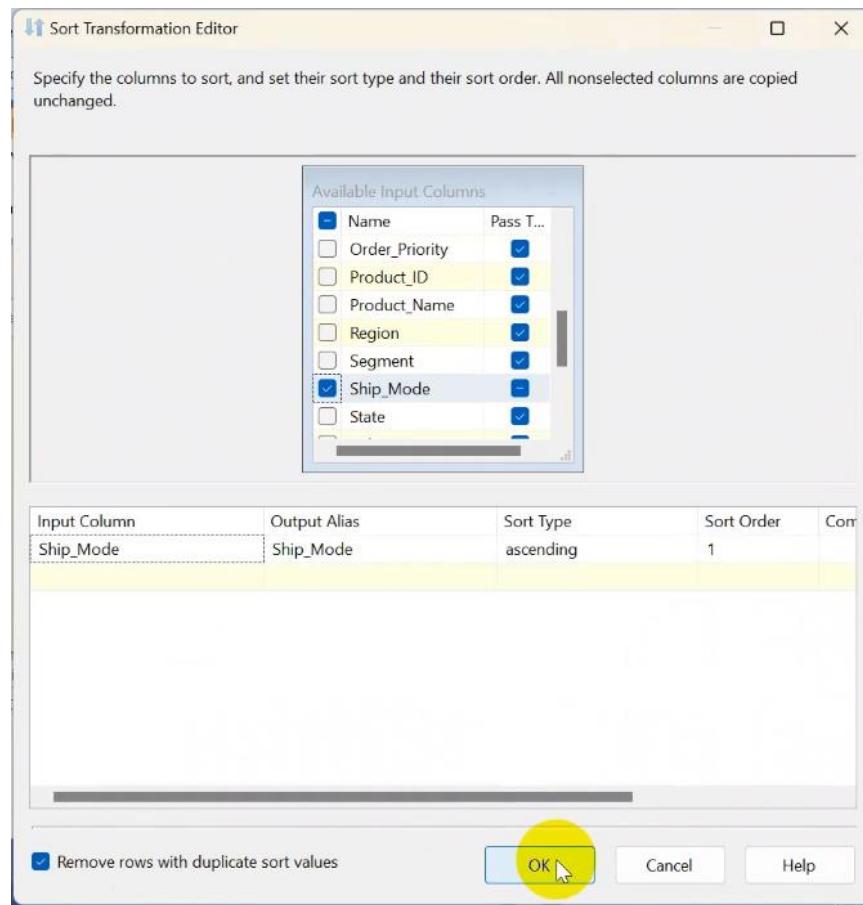
- Thiết lập các thông số sau và nhấn OK:
  - **Condition:** ISNULL(Ship\_Mode)
  - **Output Name:** Null
  - **Default output name:** NotNull



**Bước 3: Thiết lập Sort.** Nhấp chuột phải vào “Sort” và chọn **Edit** để đến giao diện “Sort Transformation Editor”.

\* Thiết lập Sort Transformation Editor:

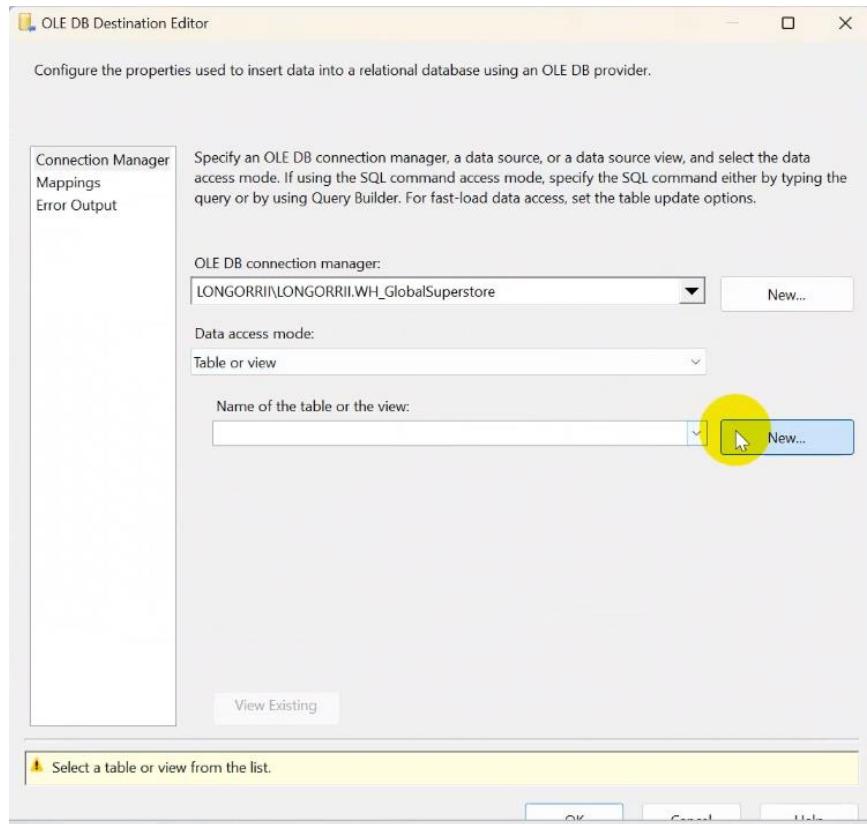
- Trong khung box **Available Input Columns**: Chọn các thuộc tính cần thiết cho Dim\_ShipMode như sau:
  - Ship\_Mode
- Tick chọn “**Remove rows with duplicate sort values**” để loại bỏ các dòng dữ liệu trùng lặp và nhấn **OK**.



**Bước 4: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “OLE DB Destination” và chọn **Edit** để đến giao diện “OLE DB Destination Editor”.

#### \* Thiết lập OLE DB Destination Editor

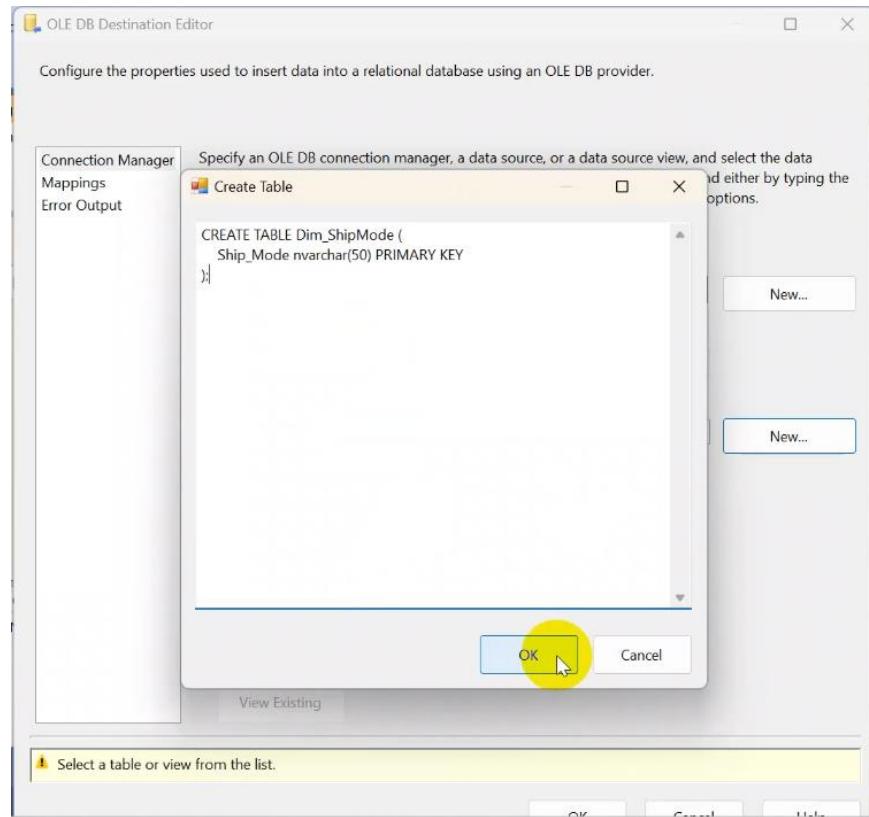
- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - LONGORRII\LONGORRII.WH\_GlobalSuperstore.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Dim\_ShipMode**



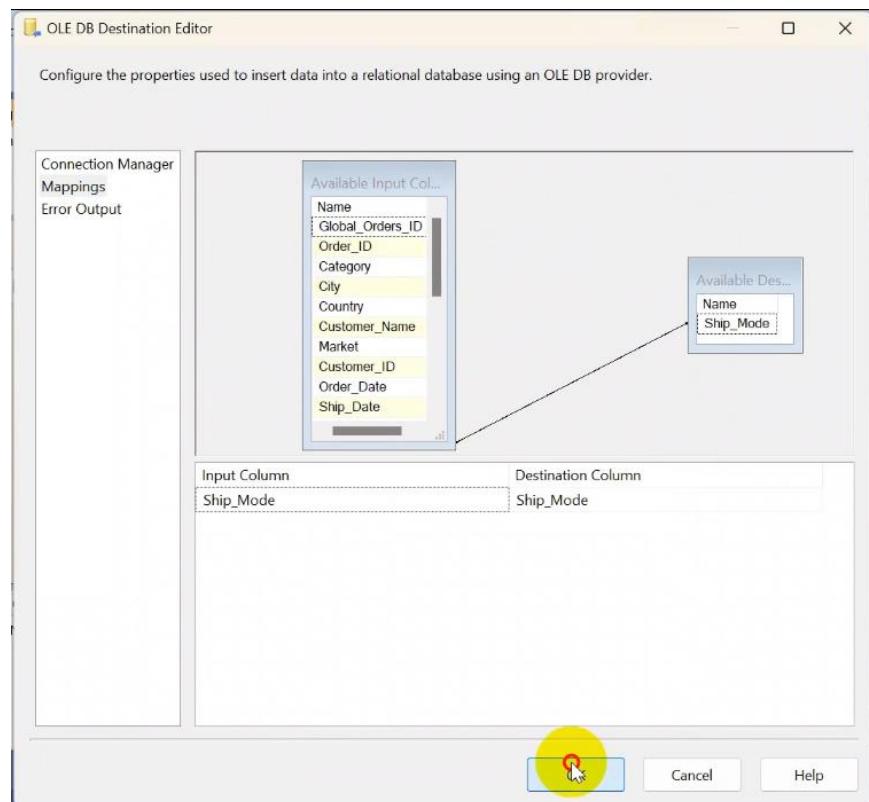
- Dán câu lệnh SQL sau để tạo bảng **Dim\_ShipMode** và nhấn **OK**

**Nội dung câu lệnh SQL:**

```
CREATE TABLE Dim_ShipMode (
    Ship_Mode nvarchar(50) PRIMARY KEY
);
```



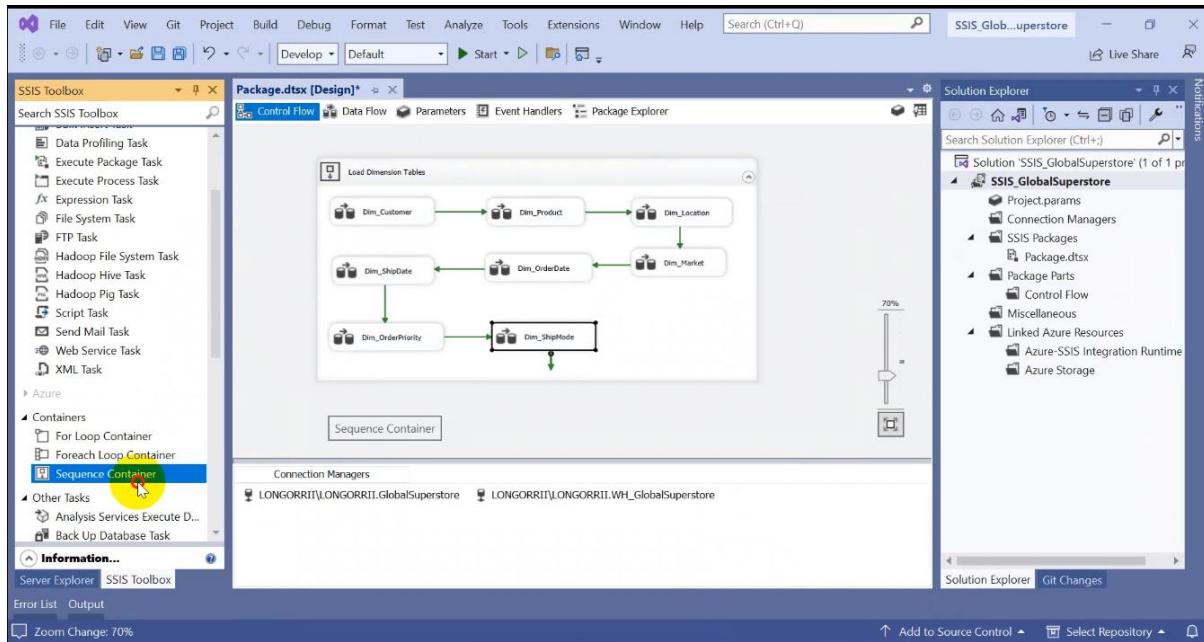
- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**.



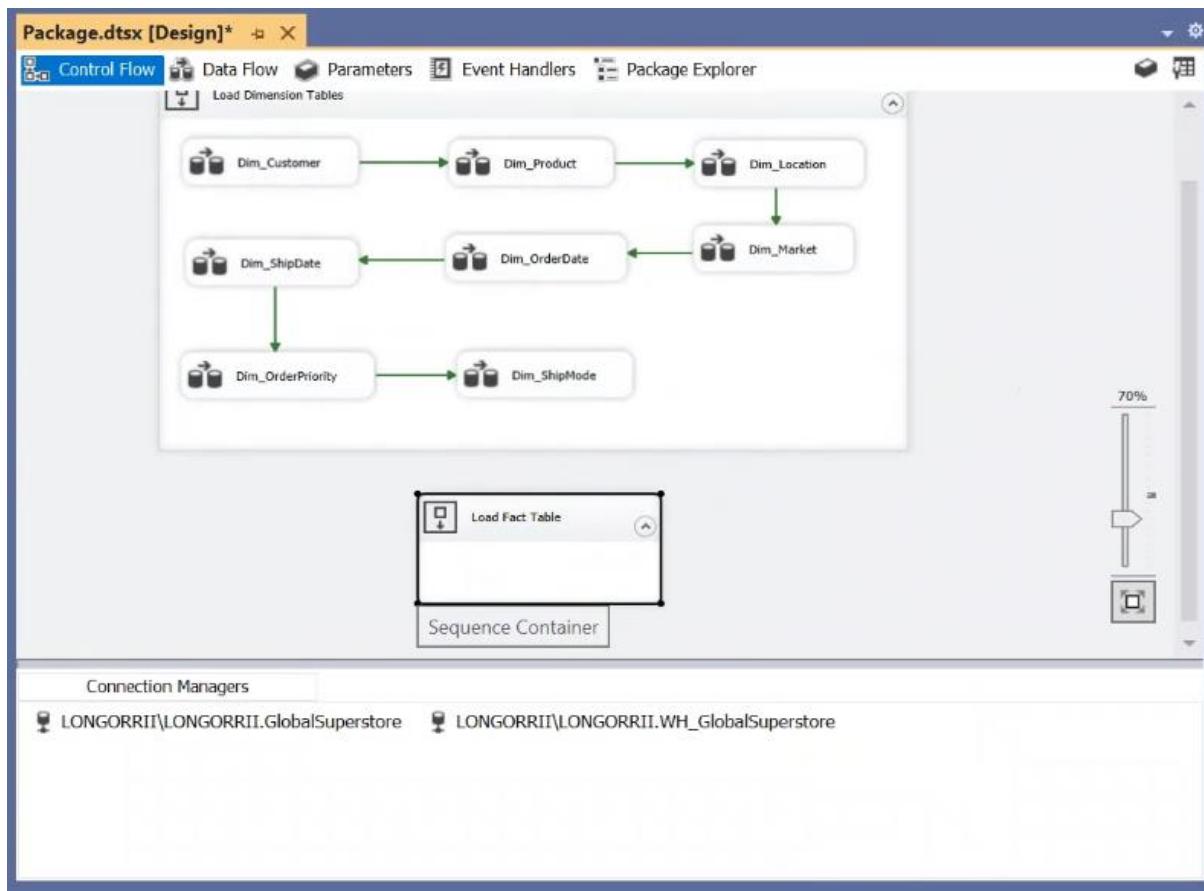
### 3.4.3 Load Fact Table

#### 3.4.3.1 Tạo mới Sequence Container

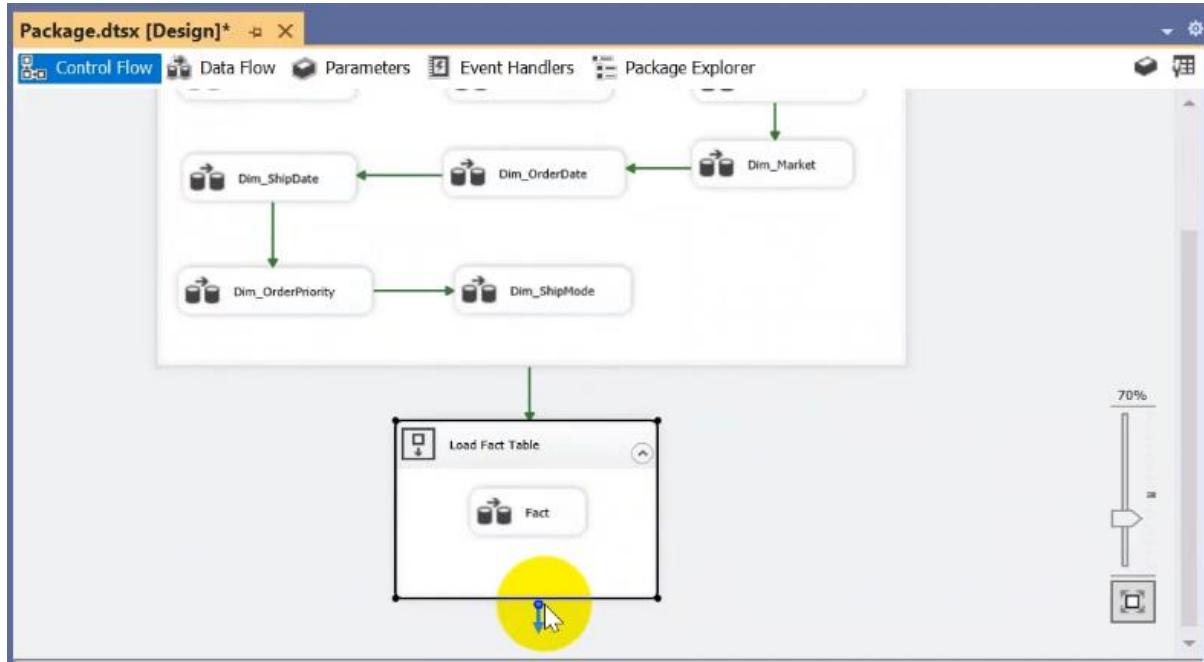
**Bước 1:** Trên thanh công cụ **SSIS Toolbox**, trong mục “Containers” ta chọn “Sequence Container”



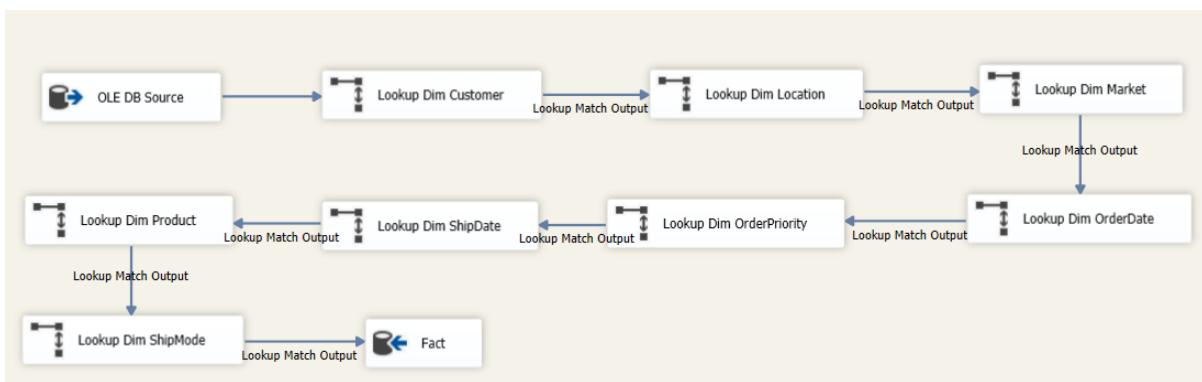
**Bước 2:** Kéo thả vào giao diện làm việc và đổi tên Sequence Container thành “Load Fact Table”



Bước 3: Thêm Data Flow Task “Fact” vào sequence container “Load Fact Table”.



### 3.4.3.2 Cấu hình Data Flow Task “Fact”

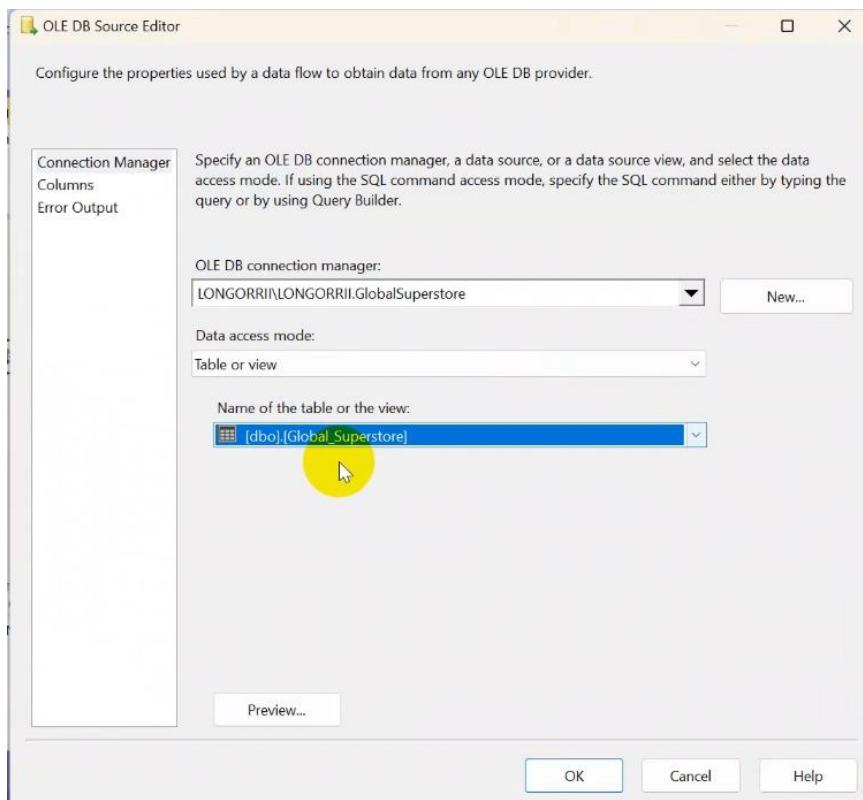


Hình 3.15: Data Flow Task “Fact”

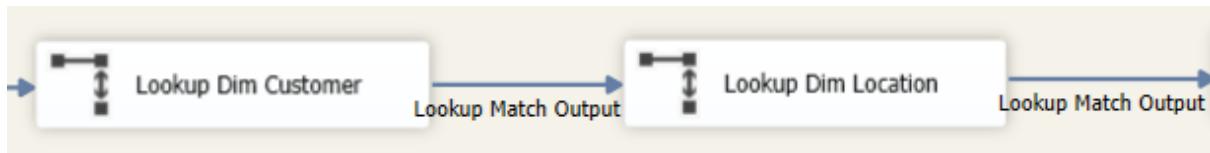
**Bước 1: Thiết lập OLE DB Source.** Nhấp chuột phải vào “OLE DB Source” và chọn **Edit** để đến giao diện “OLE DB Source Editor”.

#### \* Thiết lập OLE DB Source Editor:

- Trong mục **Connection Manager** ta chọn các thông số sau và nhấn **OK**:
  - **OLE DB connection manager:** chọn connection đến database chứa dữ liệu gốc - LONGORII\LONGORII.GlobalSuperstore.
  - **Name of the table or the view:** [dbo].[ GlobalSuperstore].



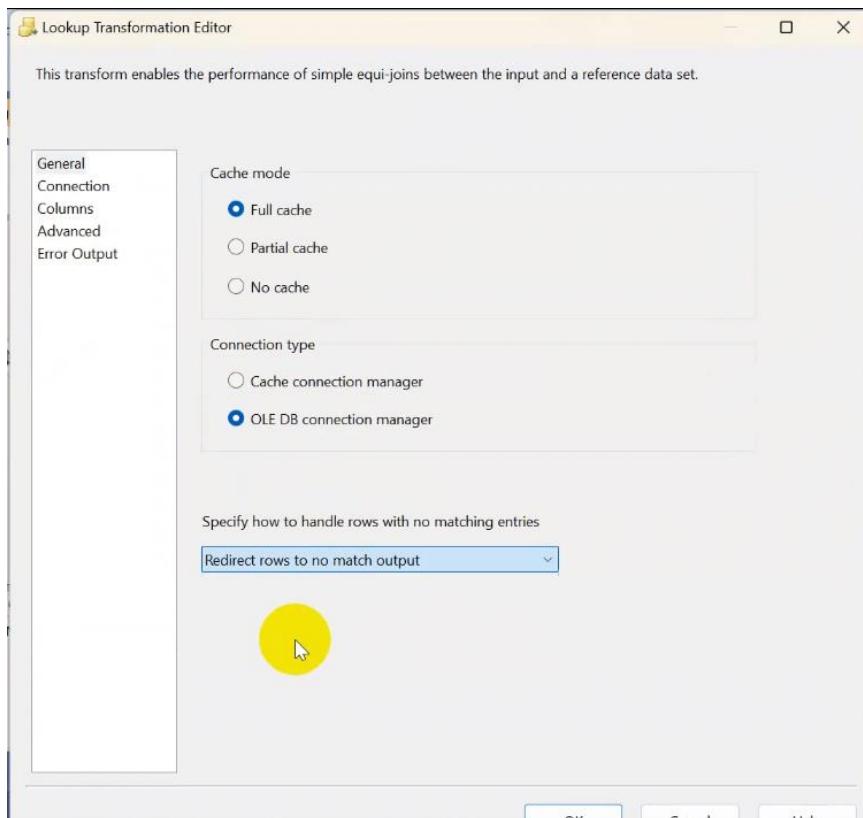
**Bước 2: Thiết lập Lookup.**



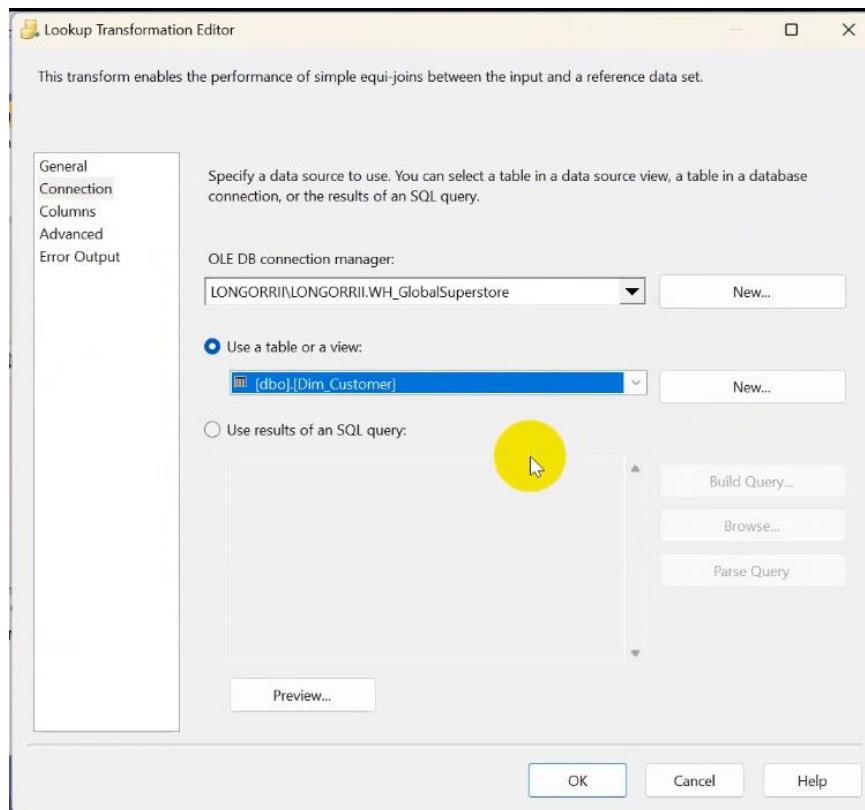
Nhấp chuột phải vào “**Lookup Dim Customer**” và chọn **Edit** để đến giao diện “**Lookup Transformation Editor**”.

\* **Thiết lập Lookup Transformation Editor:** đảm bảo rằng dữ liệu được ghi vào bảng **Fact** từ các bảng **Dimension** là chính xác, nhất quán và không bị thiếu dữ liệu.

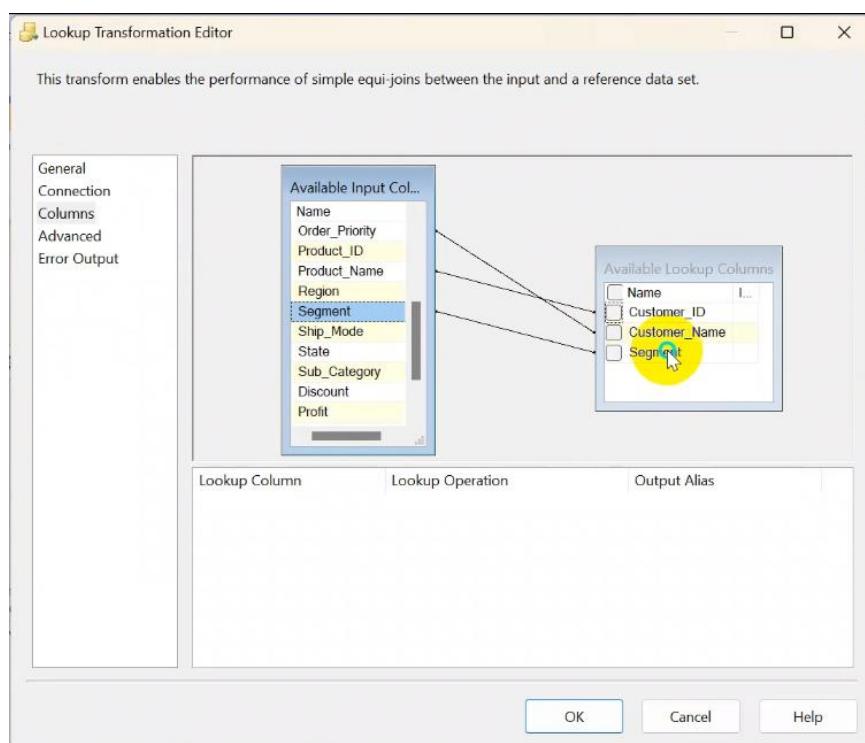
- Tại mục “**General**” ta thiết lập các thông số sau:
  - **Cache mode:** Full cache
  - **Connection type:** OLE DB connection manager
  - **Specify how to handle rows with no matching entries:** Redirect rows to no match output.



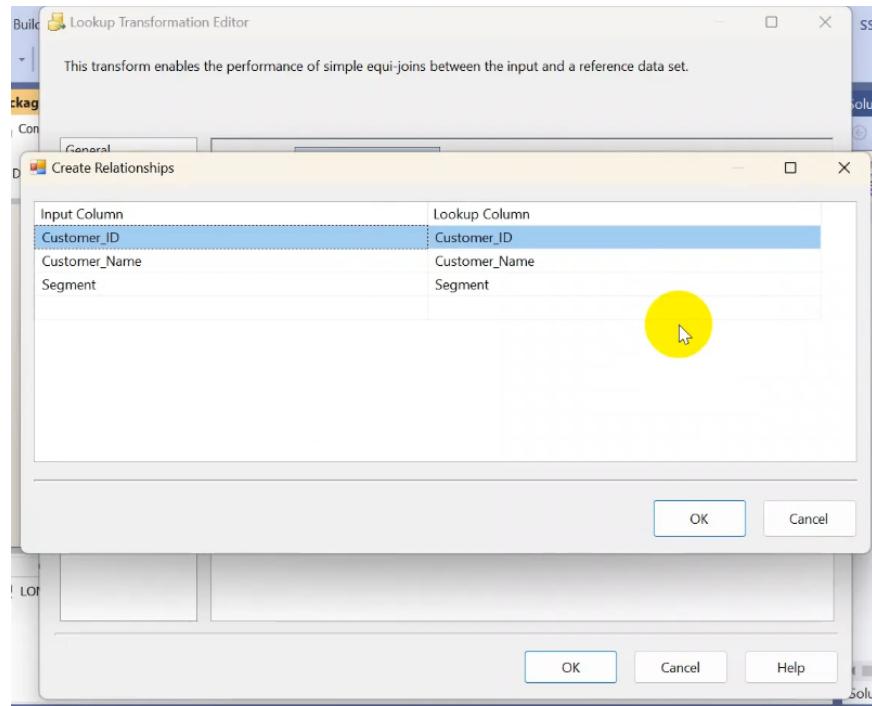
- Tại mục “**Connection**” ta chọn
  - **OLE DB connection manager:** LONGORRII\LONGORRII.WH\_GlobalSuperstore
  - **Use a table or a view:** chọn bảng Dim tương ứng với Lookup. Trong trường hợp này là **[dbo].[Dim\_Customer]**



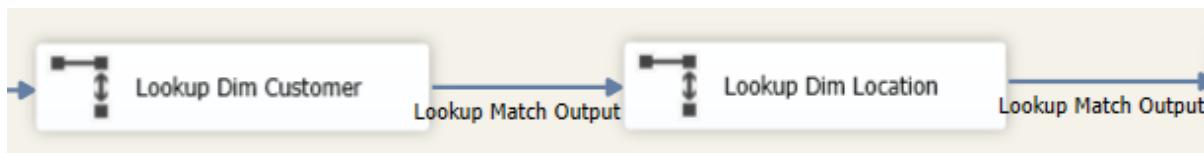
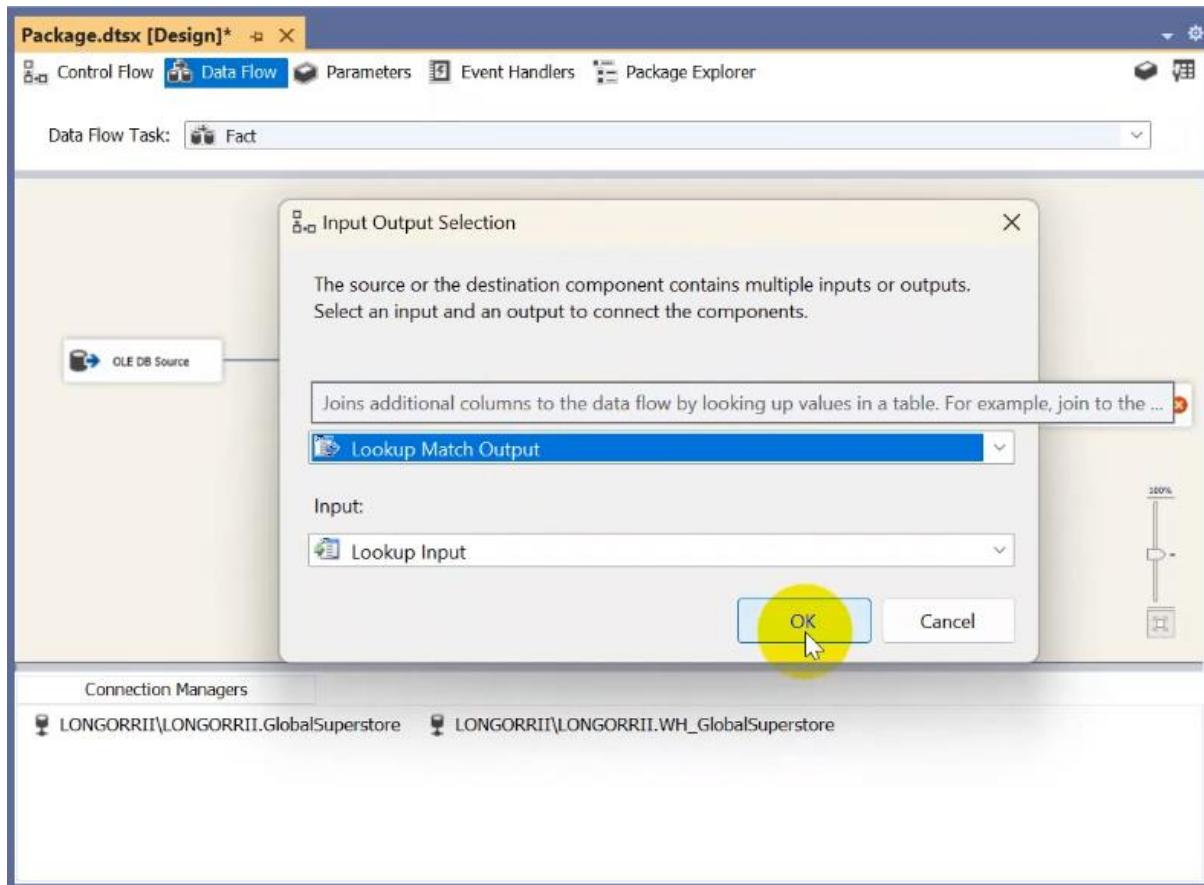
- Tại mục “**Columns**” ta kéo thả các thuộc tính ở khung box “**Available Input columns**” sao cho khớp với các thuộc tính ở “**Available Lookup Columns**”



- Nhấp chuột phải ở “Available Lookup Columns” chọn “Edit Mappings” để kiểm tra xem các thuộc tính có match chính xác hay không sau đó nhấn OK để hoàn tất.



- Khi kết nối giữa 2 Lookup, trong trường hợp này là giữa **Lookup Dim Customer** và **Lookup Dim Location**. Ta chọn “**Lookup Match Output**” và nhấn **OK**

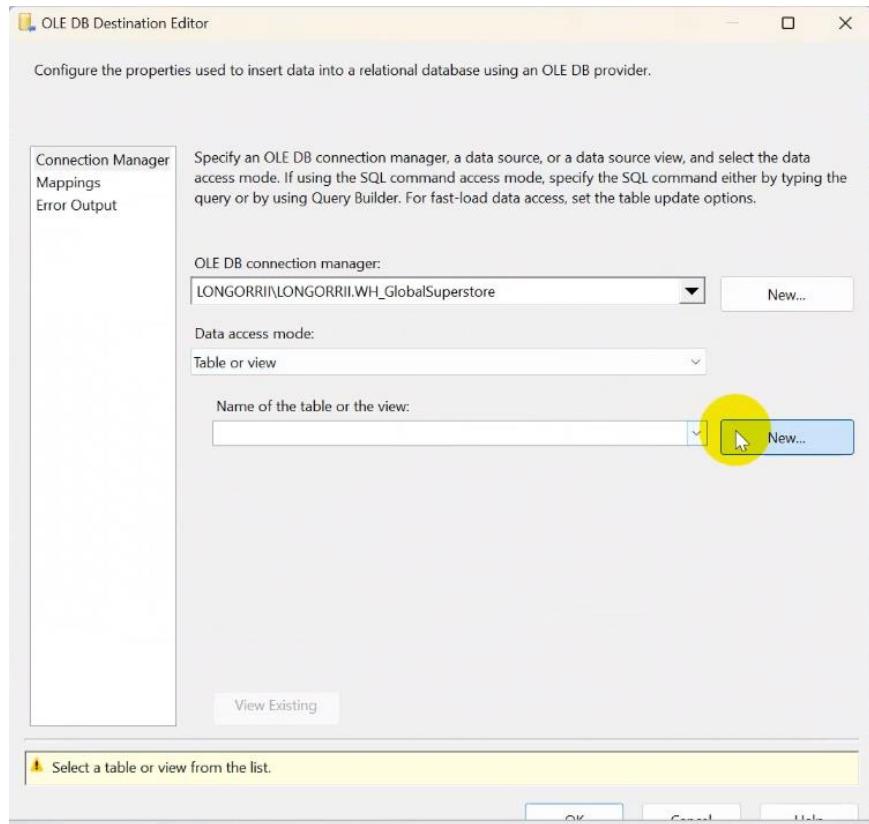


=> Ta thực hiện tương tự các bước trên với các **Lookup** ở các bảng **Dim** khác còn lại: *Market, Product, ShipDate, OrderDate, OrderPriority, OrderDate, ShipMode*.

**Bước 3: Thiết lập OLE DB Destination.** Nhấp chuột phải vào “**OLE DB Destination**” và chọn **Edit** để đến giao diện “**OLE DB Destination Editor**”.

#### \* Thiết lập OLE DB Destination Editor

- Trong mục **Connection Manager**, ta chọn các thông số sau
  - o **OLE DB connection manager**: chọn connection đến database chứa dữ liệu các bảng Dim và bảng Fact - *LONGORRII\LONGORRII.WH\_GlobalSuperstore*.
  - o **Data access mode**: chọn **Table or view**
  - o **Name of the table or the view**: nhấn **New** để tạo bảng **Fact**



- Dán câu lệnh SQL sau để tạo bảng **Dim\_Fact** và nhấn **OK**

#### **Nội dung câu lệnh SQL:**

```
CREATE TABLE Fact (
```

```
    ID INT PRIMARY KEY,
```

```
    Customer_ID nvarchar(50),
```

```
    Product_ID INT,
```

```
    Location_ID INT,
```

```
    Market nvarchar(50),
```

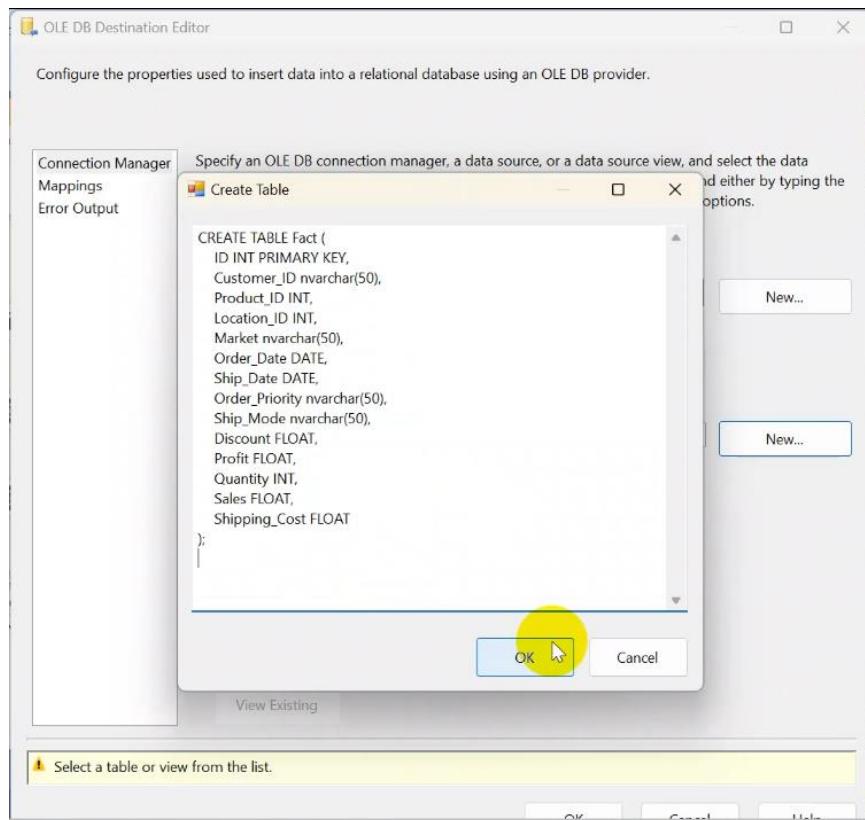
```
    Order_Date DATE,
```

```
    Ship_Date DATE,
```

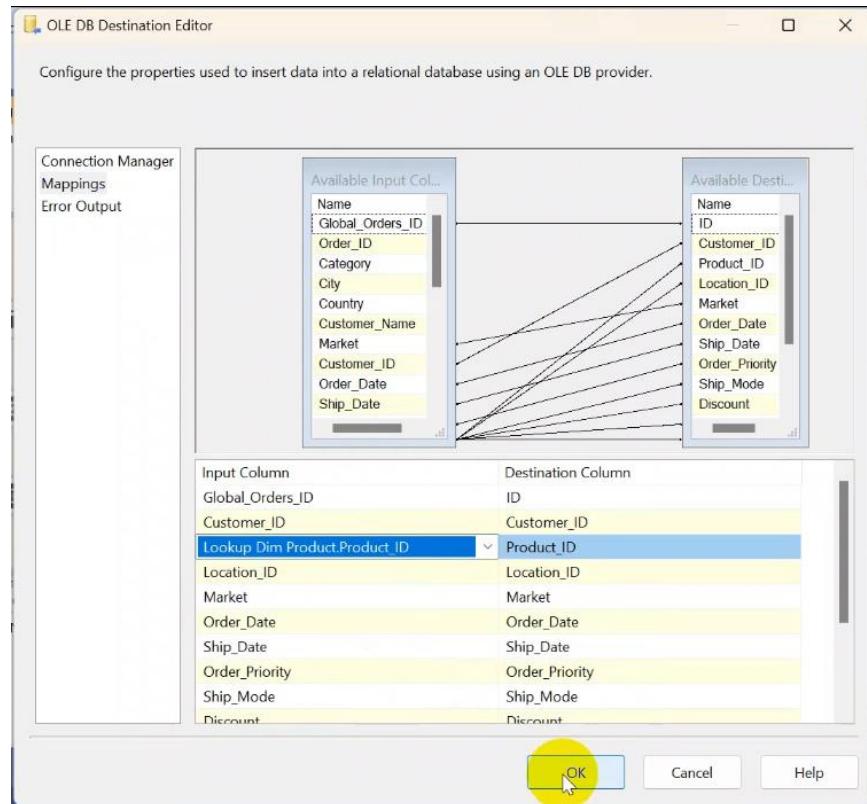
```
    Order_Priority nvarchar(50),
```

```
    Ship_Mode nvarchar(50),
```

Discount FLOAT,  
Profit FLOAT,  
Quantity INT,  
Sales FLOAT,  
Shipping\_Cost FLOAT  
);



- Trong mục **Mappings** ta xem xét việc ánh xạ các cột dữ liệu có đúng không và nhấn **OK**. Trong **Input Column**. Chọn thuộc tính “**Global\_Orders\_ID**” là thuộc tính **ID (khóa chính)** của bảng **Fact**. Lookup Dim Product.ProductID là thuộc tính cho Product\_ID của bảng Fact



### 3.4.4 Execute SQL Task

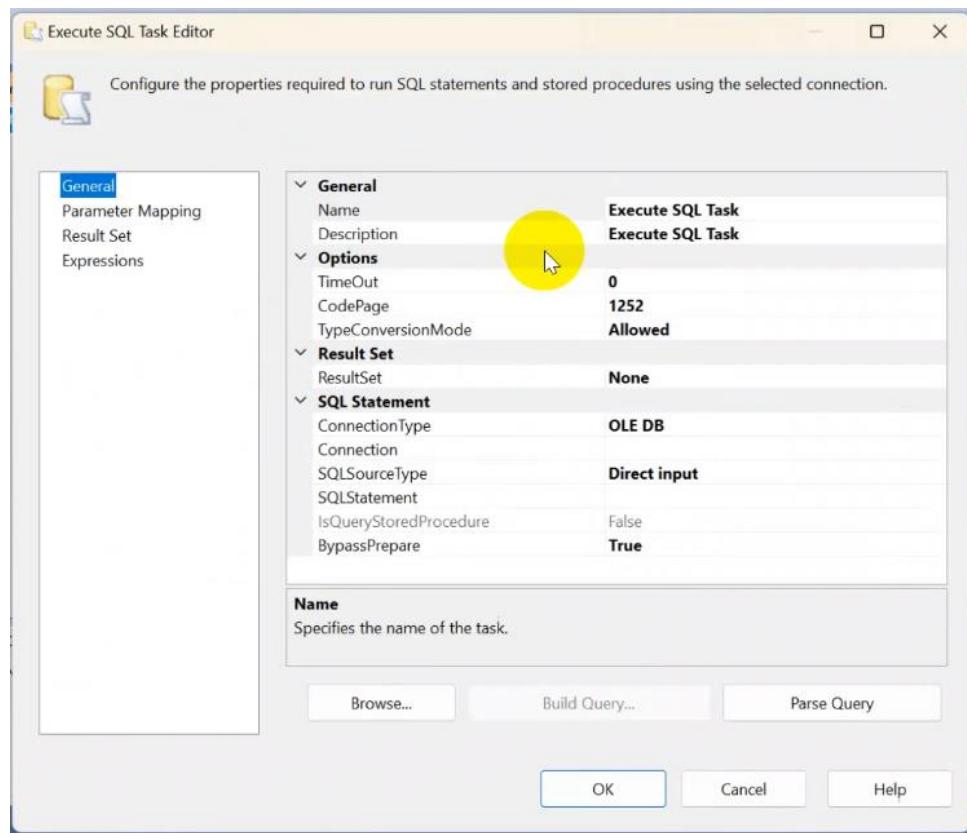
#### 3.4.4.1 Cấu hình Execute SQL Task để xóa dữ liệu sau mỗi lần chạy



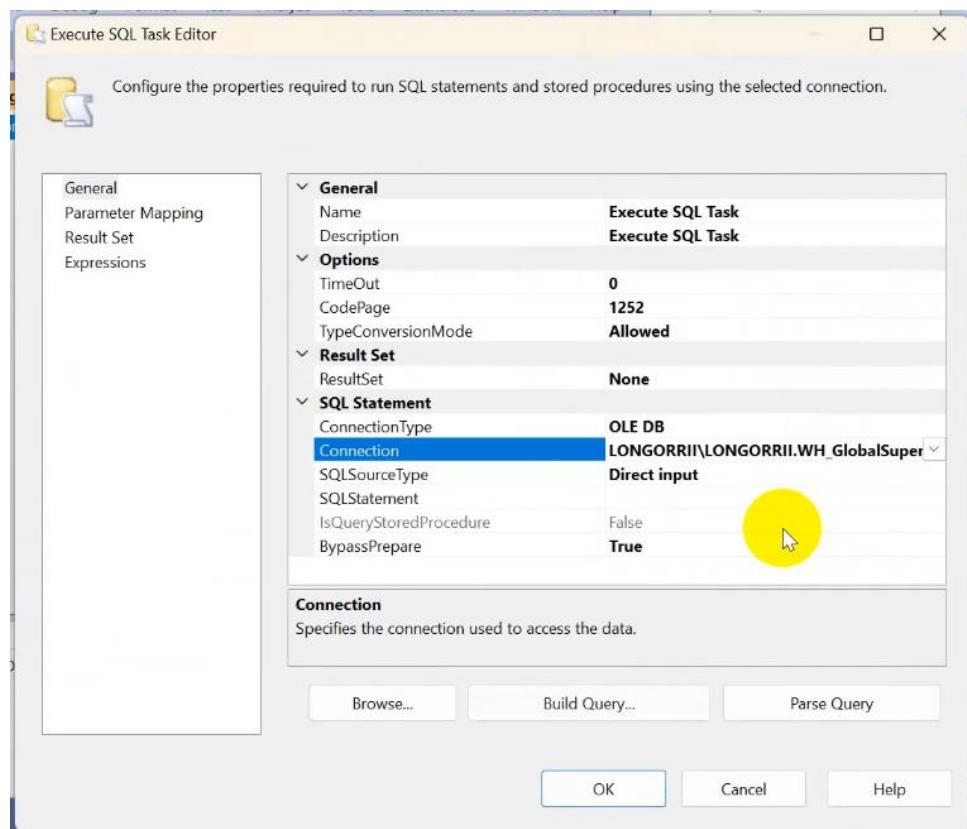
Hình 3.16: Execute SQL Task để xóa dữ liệu sau mỗi lần chạy

Để đảm bảo rằng mỗi lần chạy project đều đỗ dữ liệu mới hoàn toàn và không bị chồng chéo với dữ liệu cũ, ta thêm một Execute SQL Task vào quy trình SSIS. Nhiệm vụ của Execute SQL Task này là xóa dữ liệu cũ trong bảng Fact và các bảng Dimension trước khi bắt đầu quá trình chia tách và tải dữ liệu mới. Dưới đây là cách thực hiện:

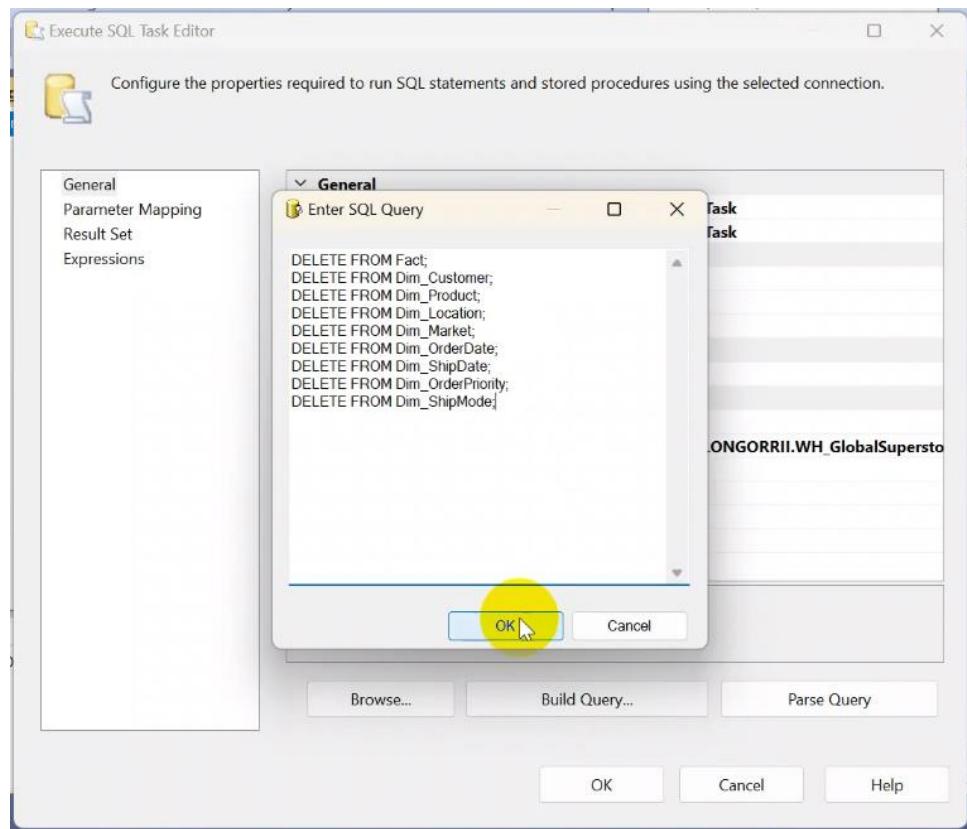
**Bước 1:** Nhấn chuột phải vào **Execute SQL Task** này và chọn **Edit** để mở giao diện “**Execute SQL Task Editor**”



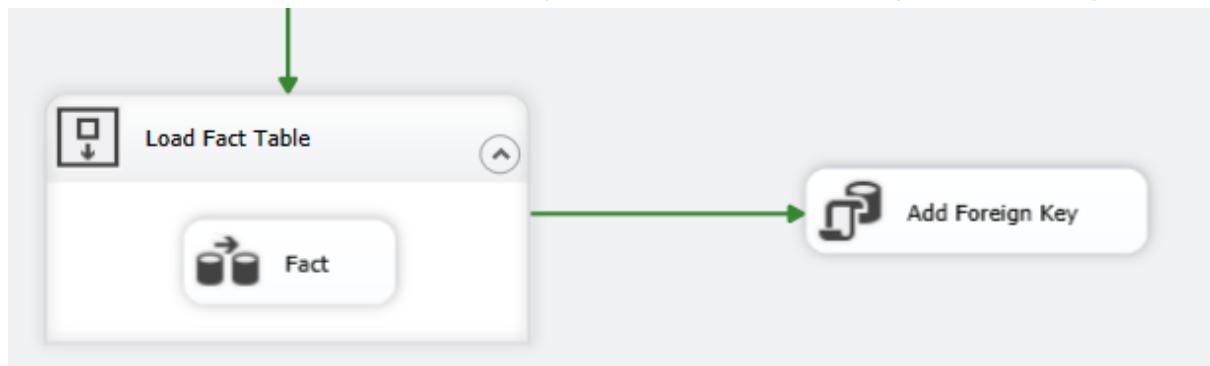
**Bước 2:** Thiết lập connection. Chọn connection đã thiết lập đến data warehouse trong SQL Server



**Bước 3:** Ở ô SQLStatement, thêm các câu truy vấn SQL thực hiện xóa dữ liệu cũ trong các bảng Dim và bảng Fact mỗi khi chạy project và nhấn OK để hoàn thành.



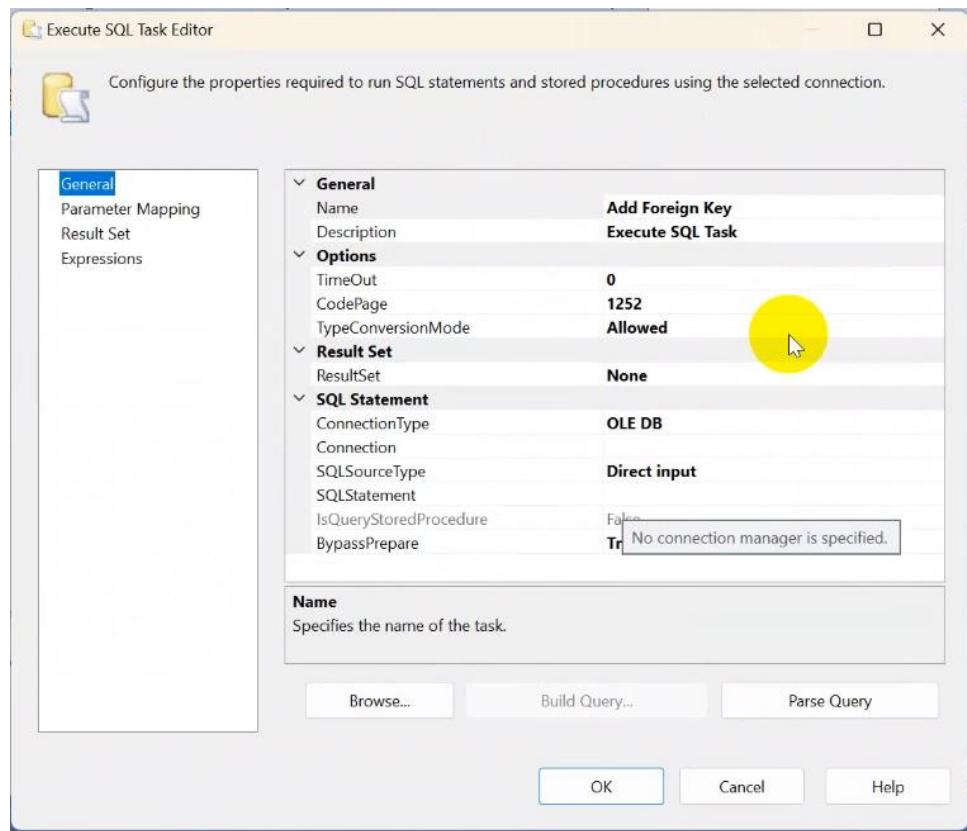
#### 3.4.4.2 Cấu hình Execute SQL Task để thêm khóa ngoại vào bảng Fact



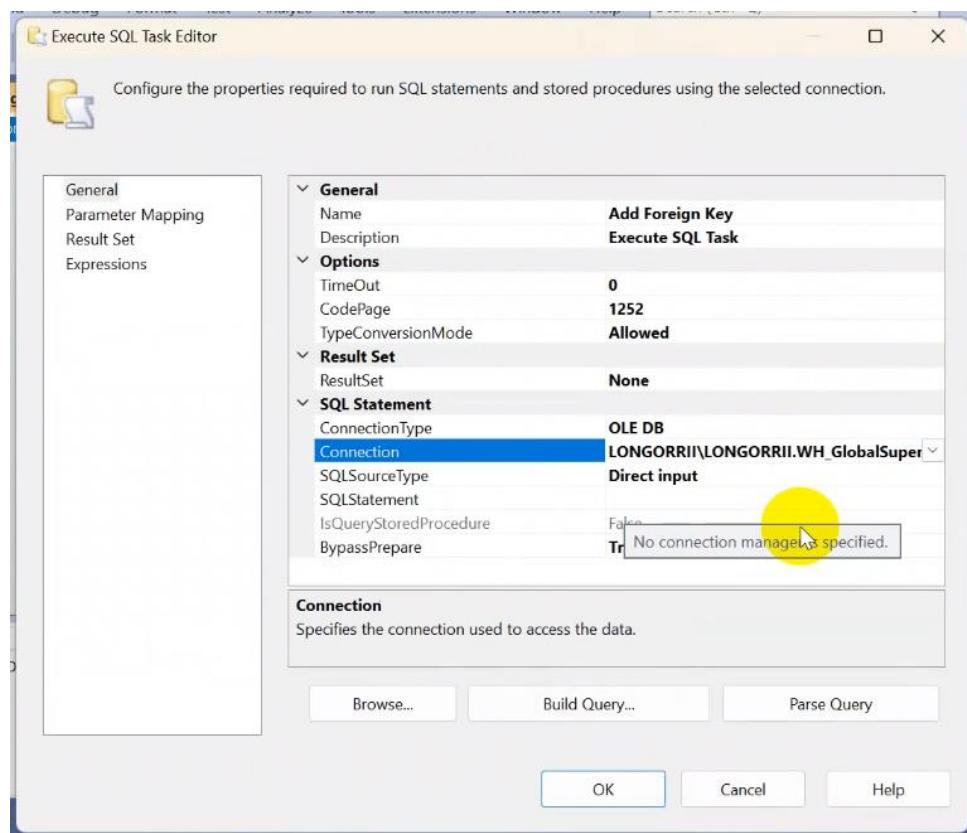
Hình 3.17: Execute SQL Task để thêm khóa ngoại vào bảng Fact

Sử dụng Execute SQL Task để tiến hành tạo các khóa ngoại theo đúng thiết kế. Ta thực hiện các bước như sau:

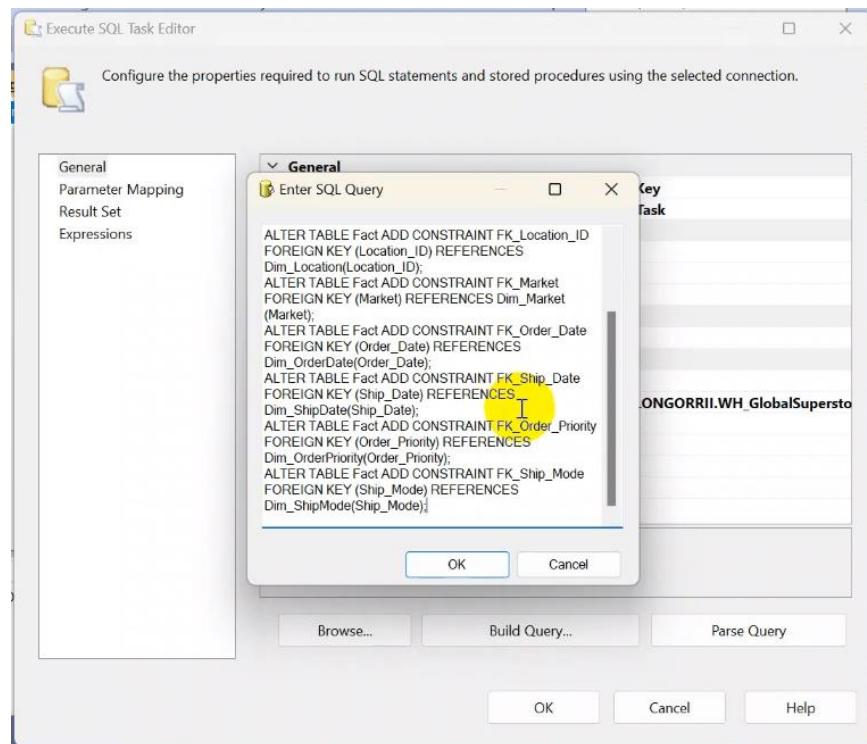
**Bước 1:** Nhấn chuột phải vào Execute SQL Task “Add Foreign Key” và chọn Edit để mở giao diện “Execute SQL Task Editor”



**Bước 2:** Thiết lập connection. Chọn connection đã thiết lập đến data warehouse trong SQL Server



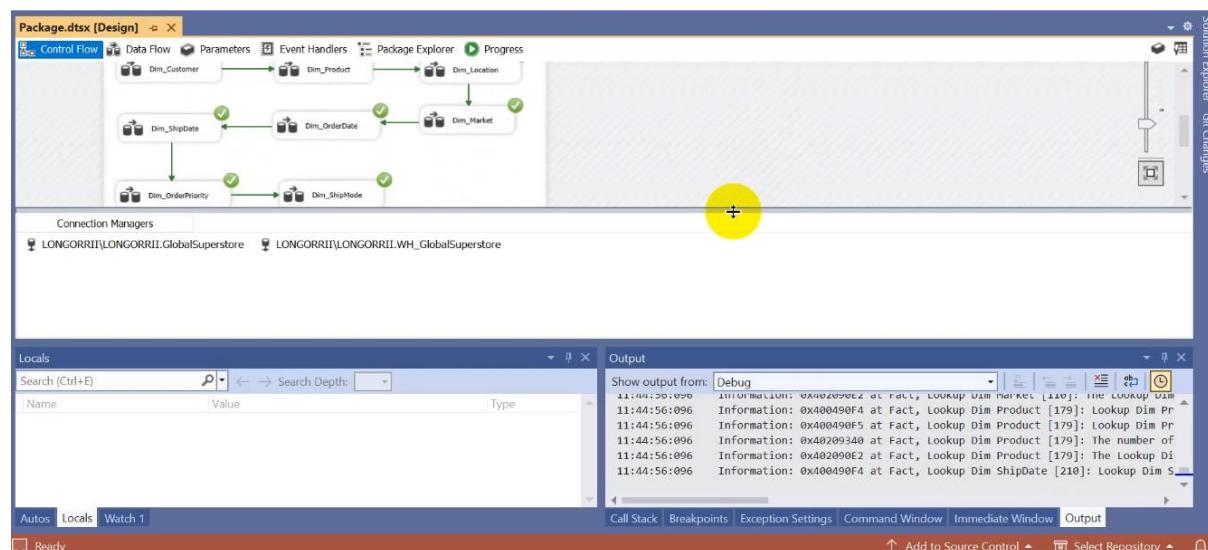
**Bước 3:** Ở ô SQLStatement ta tiến hành thêm các lệnh SQL để thêm các khóa ngoại vào mỗi lần khởi chạy lại project tiếp theo và nhấn OK để hoàn thành.

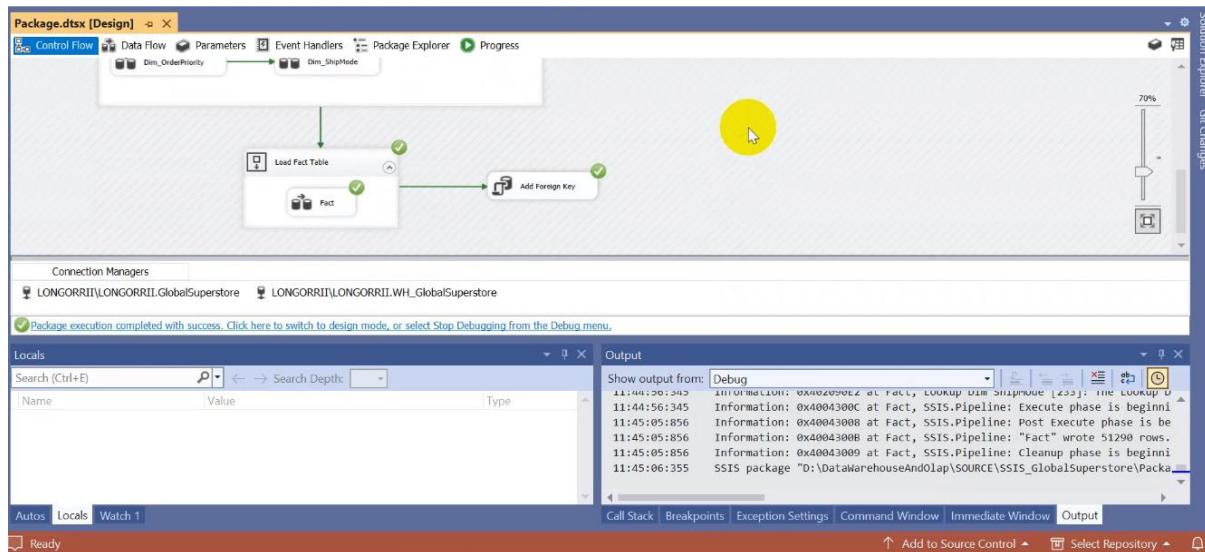


### 3.5 Chạy Project SSIS

Sau khi cấu hình xong tất cả các bước ta tiến hành nhấn “Start” để tiến hành chạy khởi động project **SSIS\_GlobalSuperstore**

Kết quả chạy project:





## CHƯƠNG 4: QUÁ TRÌNH PHÂN TÍCH DỮ LIỆU (SSAS)

### 4.1 Thực hiện quá trình SSAS

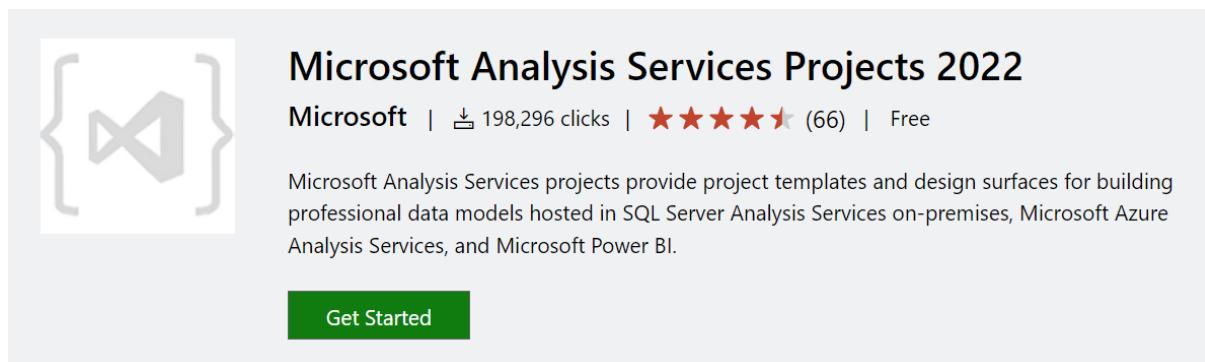
#### 4.1.1 Chuẩn bị các công cụ

Ngoài các công cụ đã đề cập ở trước như là:

1. **Visual Studio Community 2022**

2. **Microsoft SQL Server 2022** (có cài đặt Analysis Services.)

Ta cần cài đặt thêm **Microsoft Analysis Services Projects**



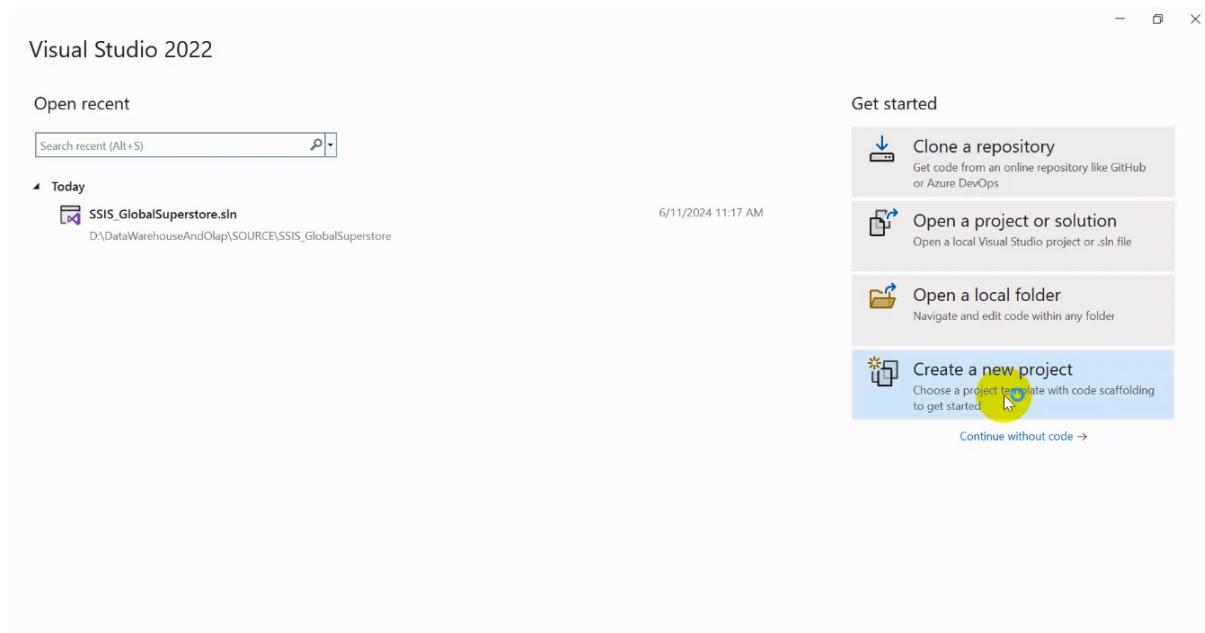
Hình 4.1: Microsoft Analysis Services Projects 2022

Link download [5]:

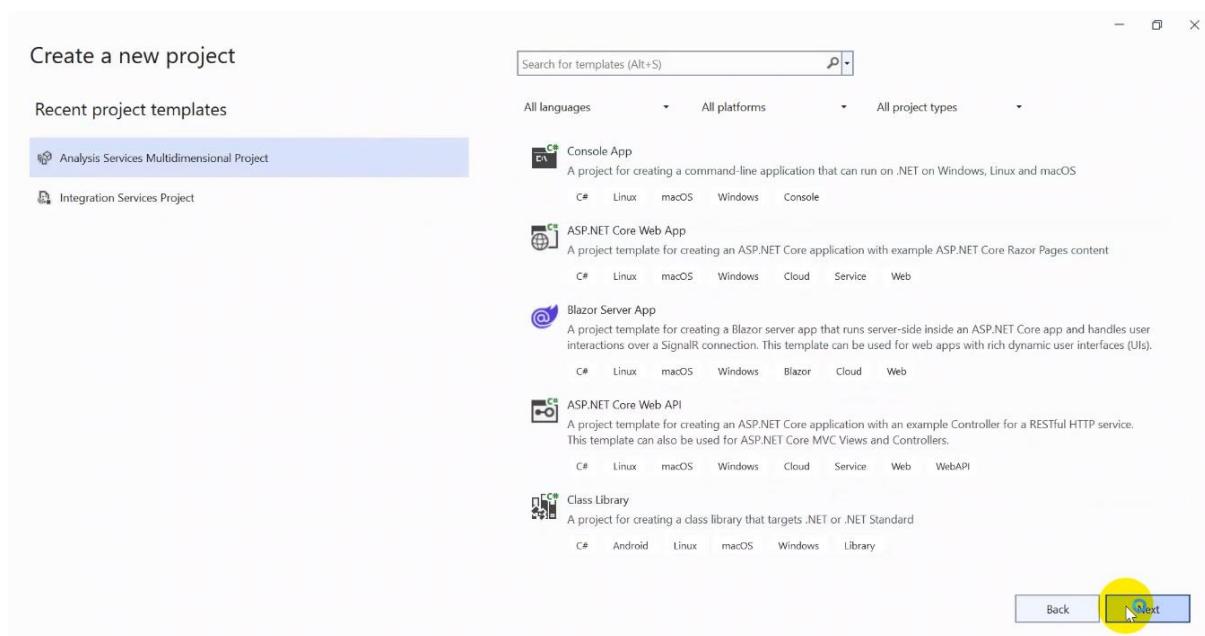
<https://marketplace.visualstudio.com/items?itemName=ProBITools.MicrosoftAnalysisServicesModelingProjects2022>

#### 4.1.2 Tạo mới project SSAS

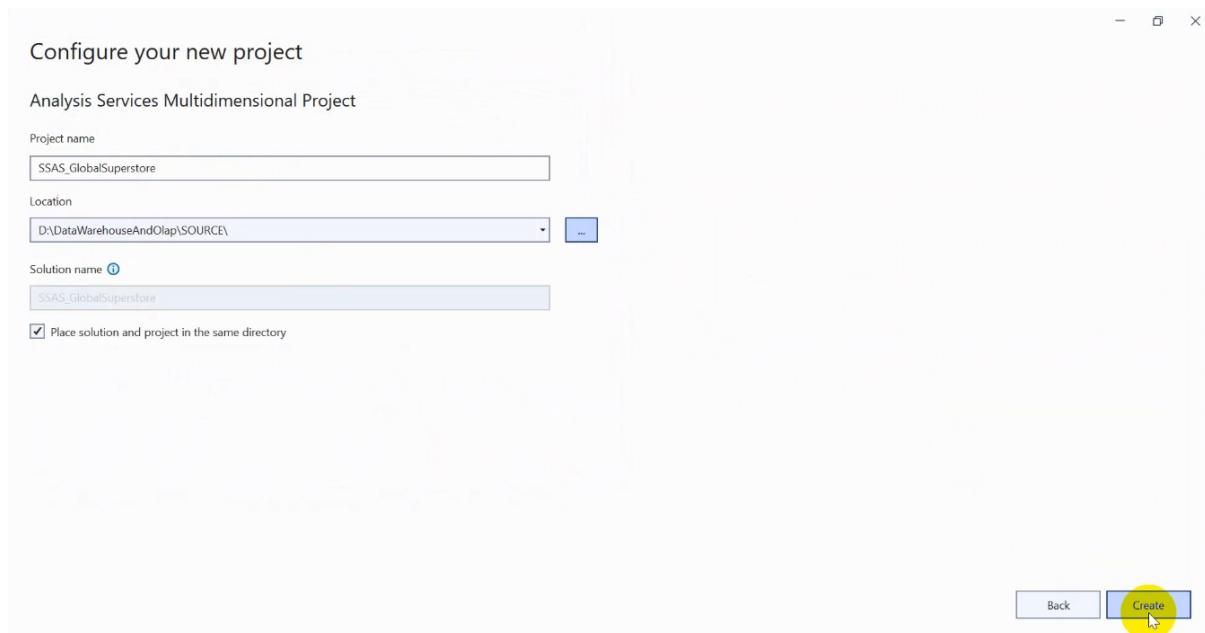
**Bước 1:** Mở Visual Studio 2022 và nhấn chọn “Create a new project”



## Bước 2: Chọn “Analysis Services Multidimensional Project” và nhấn Next



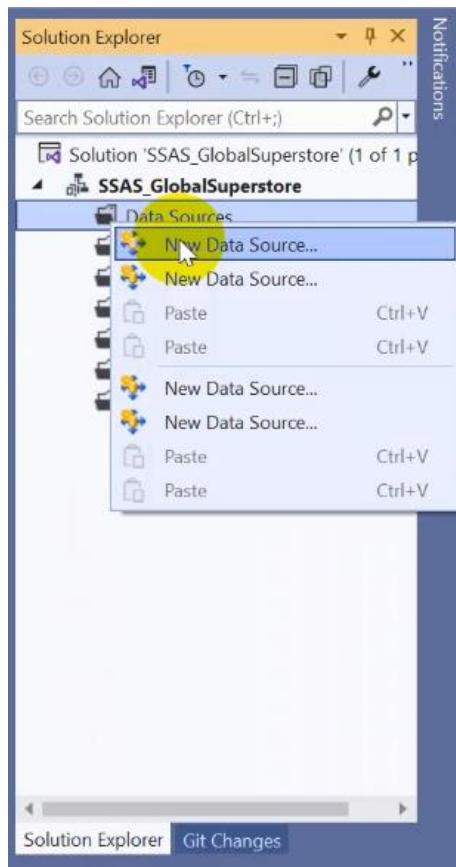
## Bước 3: Tiến hành đặt tên project là “SSAS\_GlobalSuperstore”, chọn đường dẫn lưu thư mục dự án và nhấn “Create” để hoàn tất việc tạo mới



### 4.1.3 Cấu hình và thực hiện quá trình SSAS

#### 4.1.3.1 Tạo Data Sources

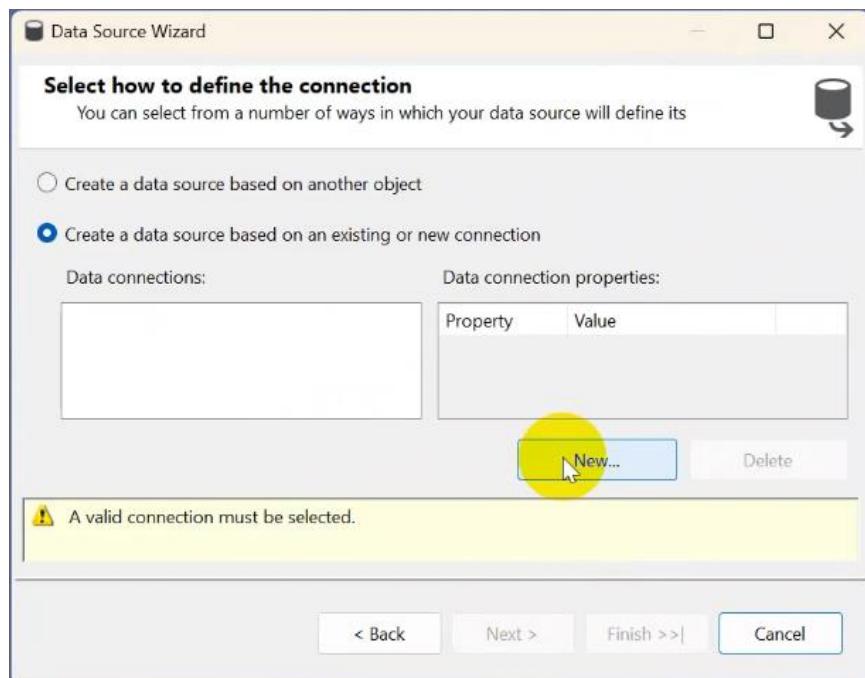
**Bước 1:** Trên thanh công cụ “Solution Explorer”. Click chuột phải vào “Data Sources” và nhấn chọn “New Data Source”



**Bước 2:** Màn hình thông báo “**Data Source Wizard**” hiện ra và chọn “**Next**” để tiếp tục.

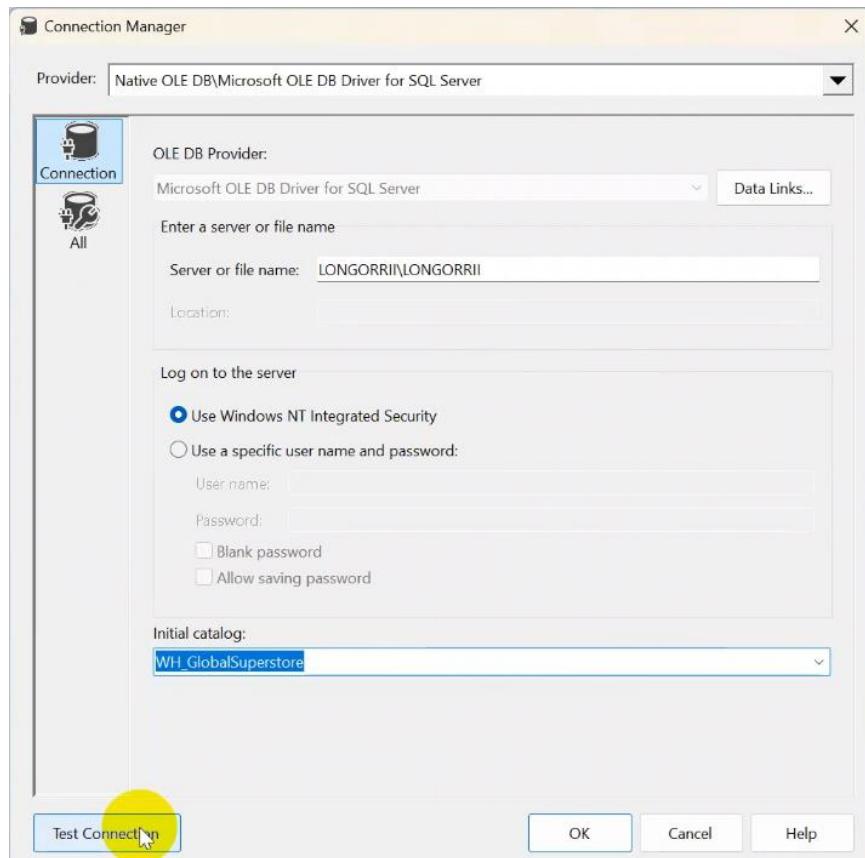


**Bước 3:** Trong **Data Source Wizard**. Chọn “Create a data source based on an existing or new connection” sau đó chọn **New**.

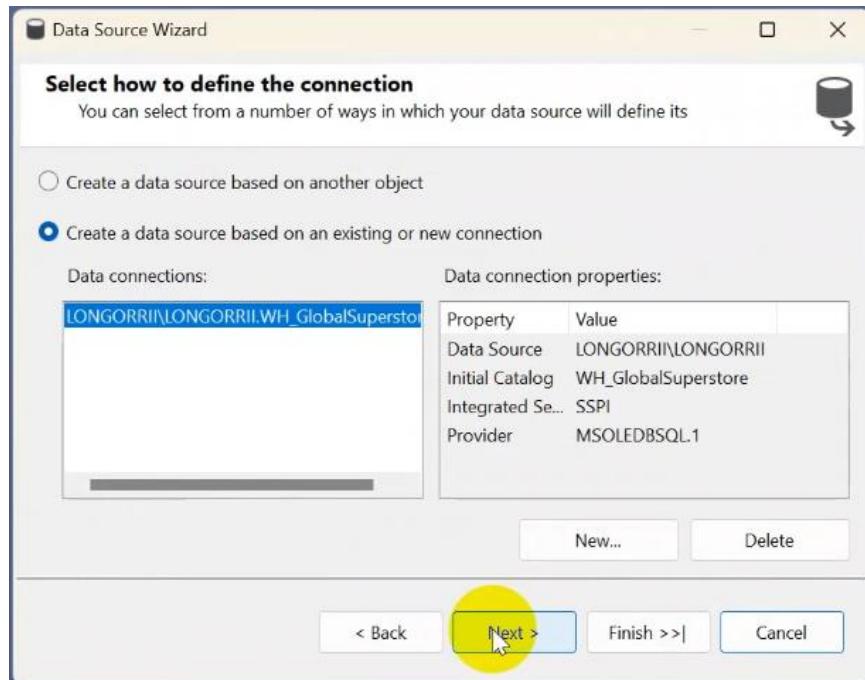


**Bước 4:** Thiết lập kết nối. Trong **Connection Manager** ta thiết lập các thông số dưới và nhấn **Test Connection** để kiểm tra kết nối.

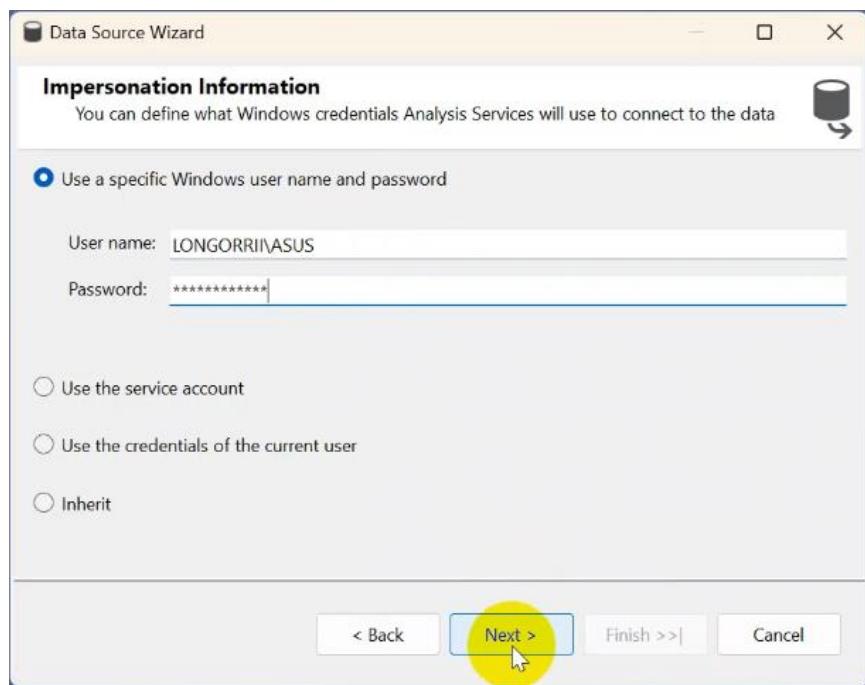
- **Provider:** Microsoft OLE DB Driver for SQL Server
- **Server or file name:** LONGORRII\LONGORRII (tên server của SQL Server)
- **Initial catalog:** WH\_GlobalSuperstor (tên cơ sở dữ liệu đã được tạo từ quá trình SSIS)



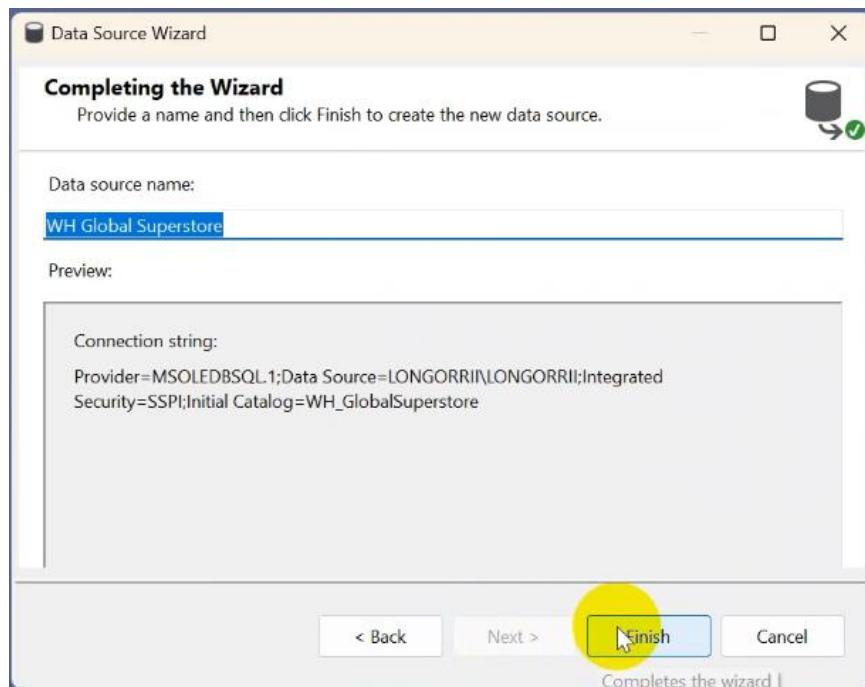
**Bước 5:** Chọn Data Connection vừa tạo và chọn Next để tiếp tục.



**Bước 6:** Chọn Use a specific Windows user name and password. Sau đó nhập User name và Password và nhấn Next.

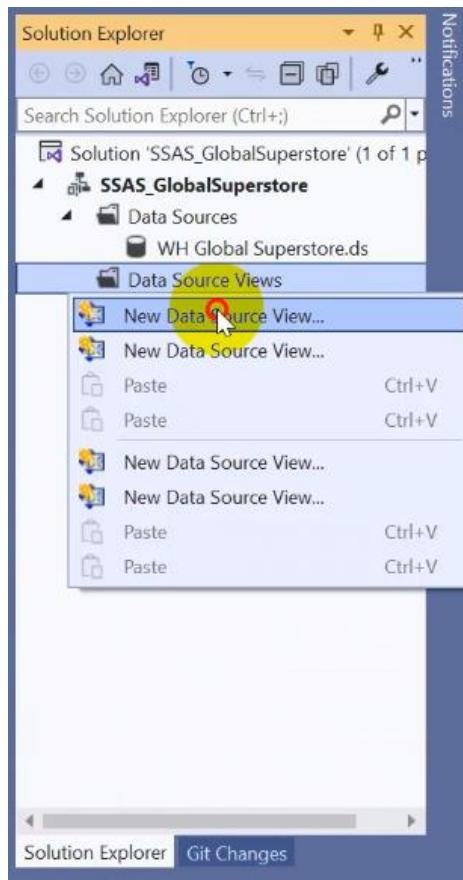


**Bước 7:** Đặt tên Data source name và nhấn “Finish” để kết thúc quá trình tạo Data Source

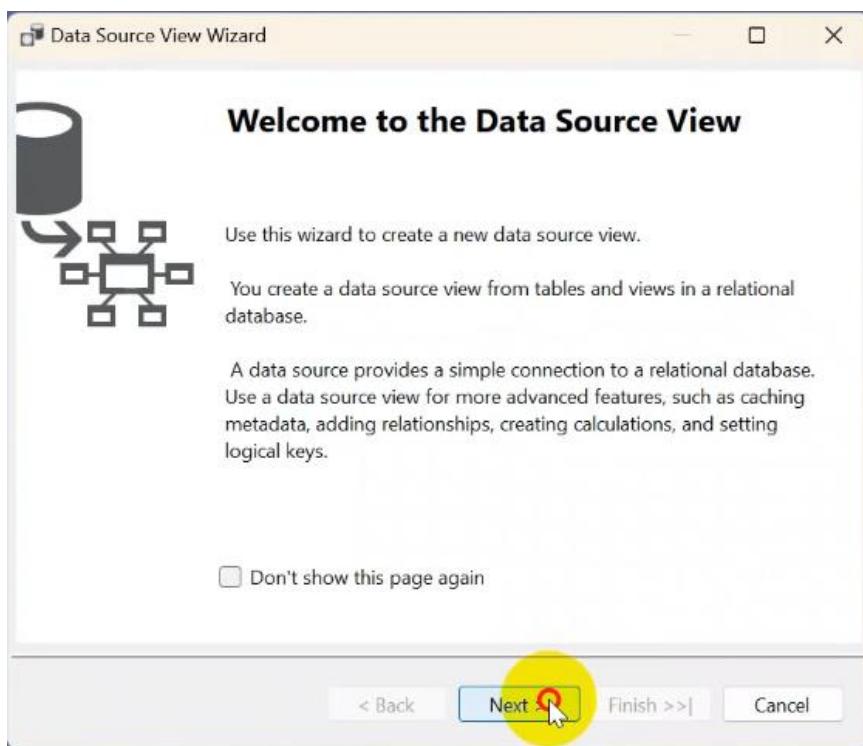


#### 4.1.3.2 Tạo Data Source Views

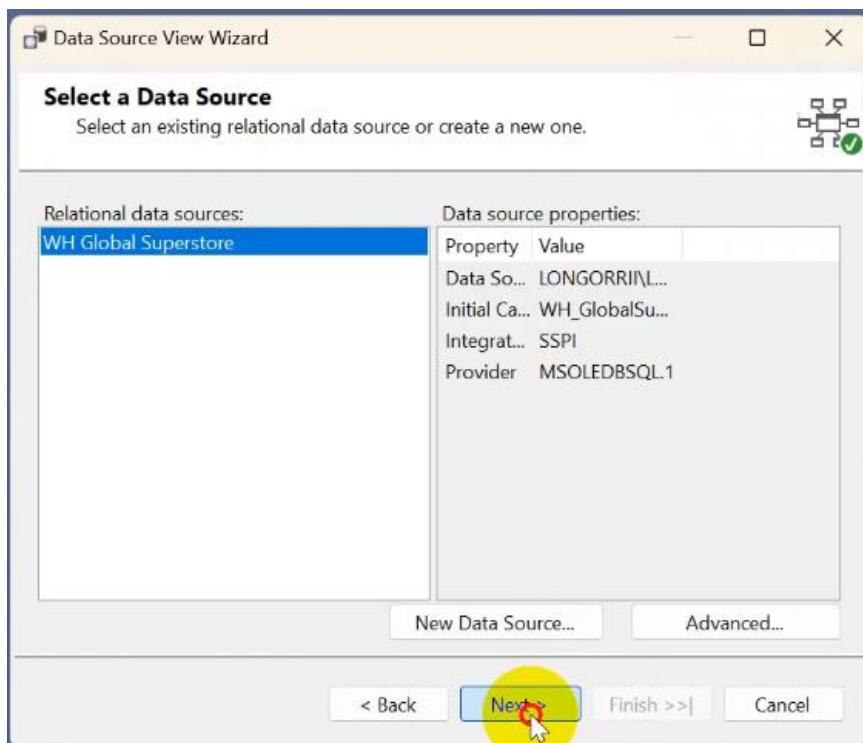
**Bước 1:** Trên thanh công cụ “Solution Explorer”. Click chuột phải vào “Data Source View” và nhấp chọn “New Data Source View”



**Bước 2:** Nhấn Next để tiếp tục.

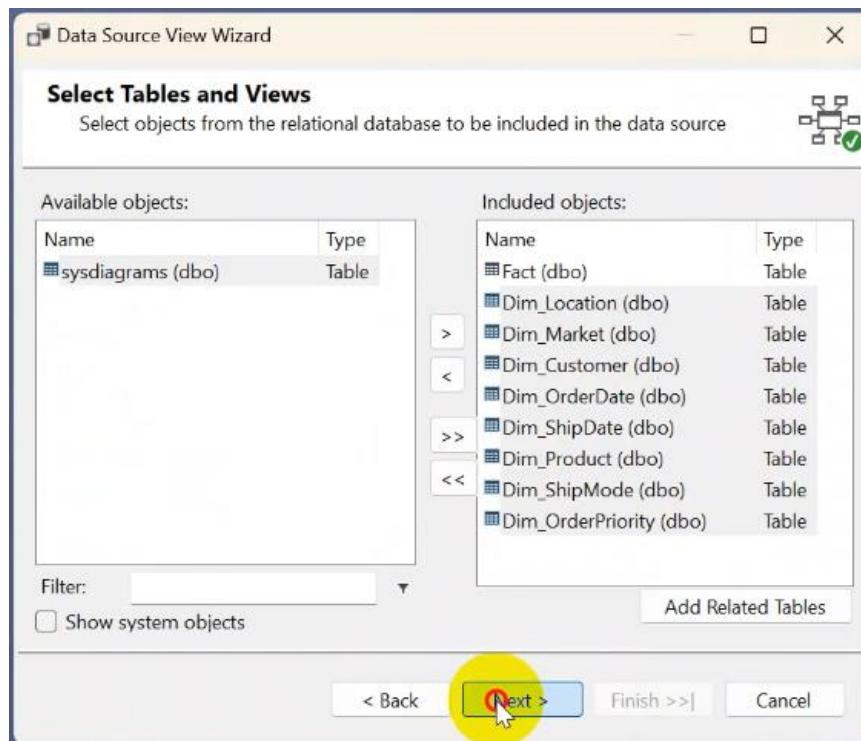


**Bước 3:** Chọn Data source vừa tạo và nhấn “Next”

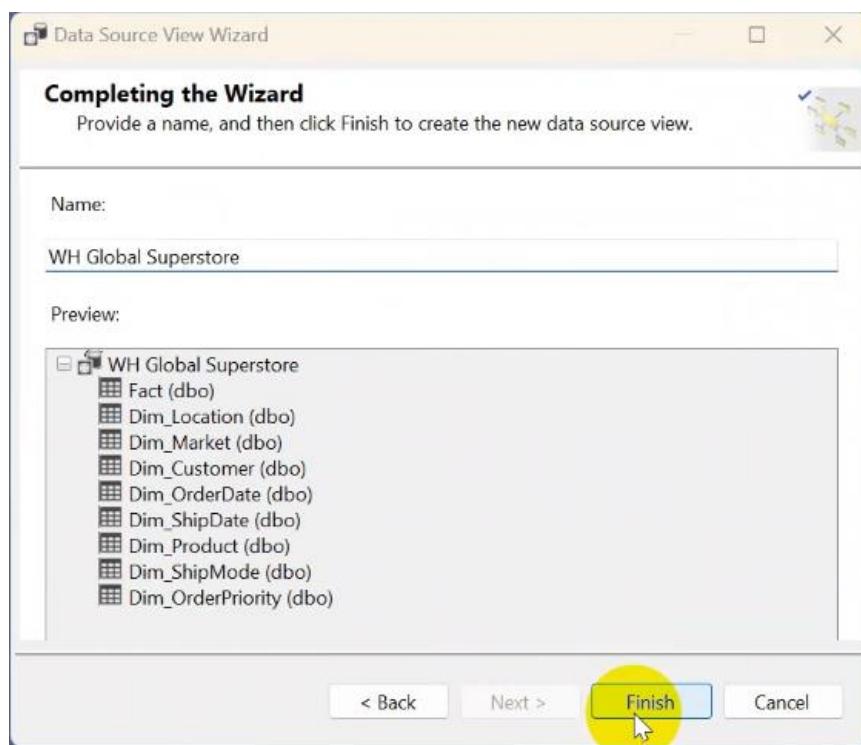


**Bước 4:** Chọn bảng Fact từ Available objects sau đó chuyển sang Included objects.

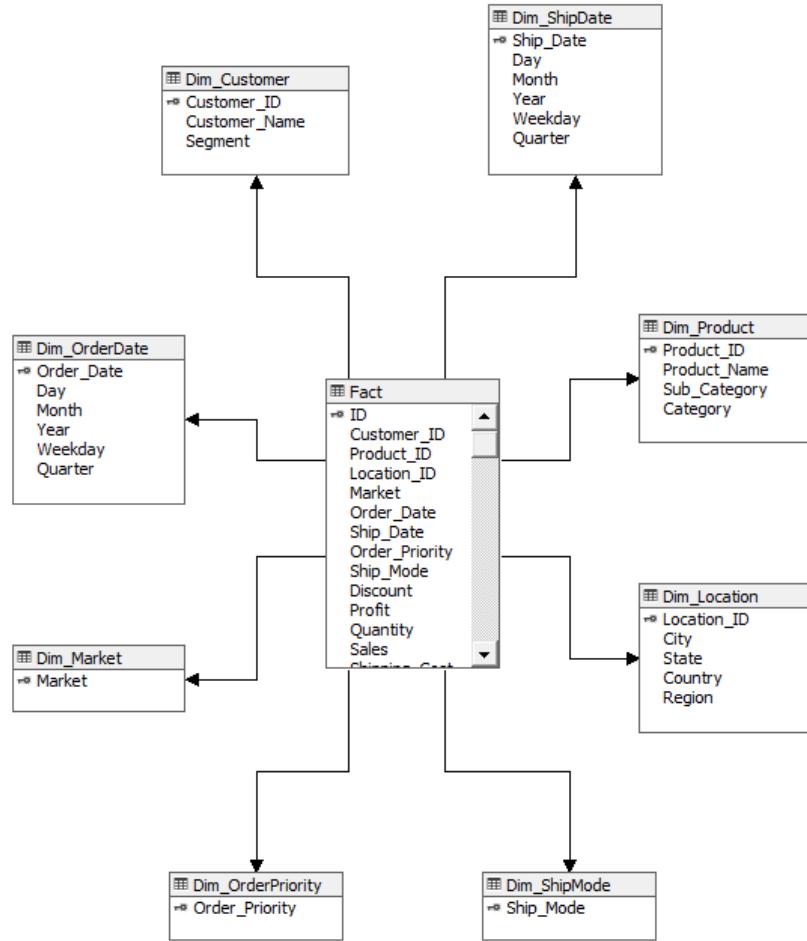
Nhấn Add Related Tables để thêm các bảng Dim. Sau đó nhấn “Next”



**Bước 5:** Nhấn “Finish” để hoàn tất quá trình tạo Data Source View



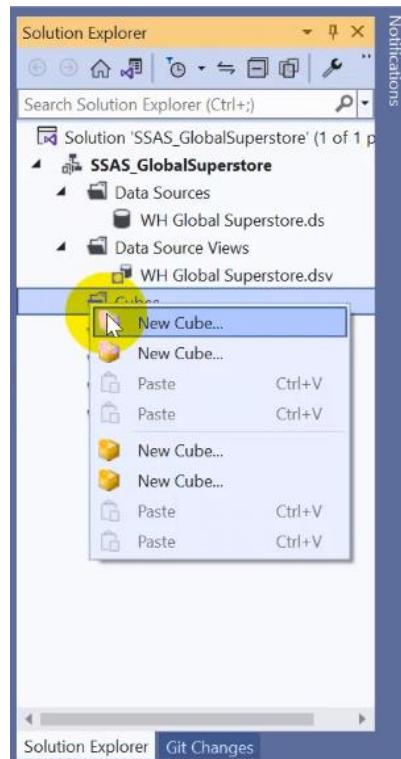
Sau khi hoàn tất ta sẽ có Data Source View như hình dưới đây:



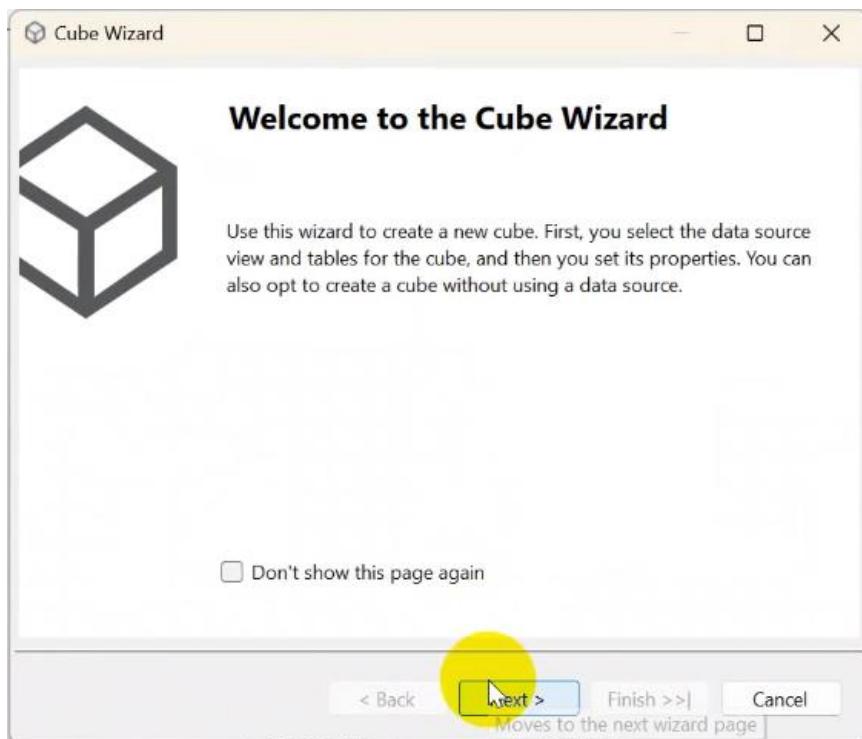
Hình 4.2: Data Source View

#### 4.1.3.3 Tạo Cubes

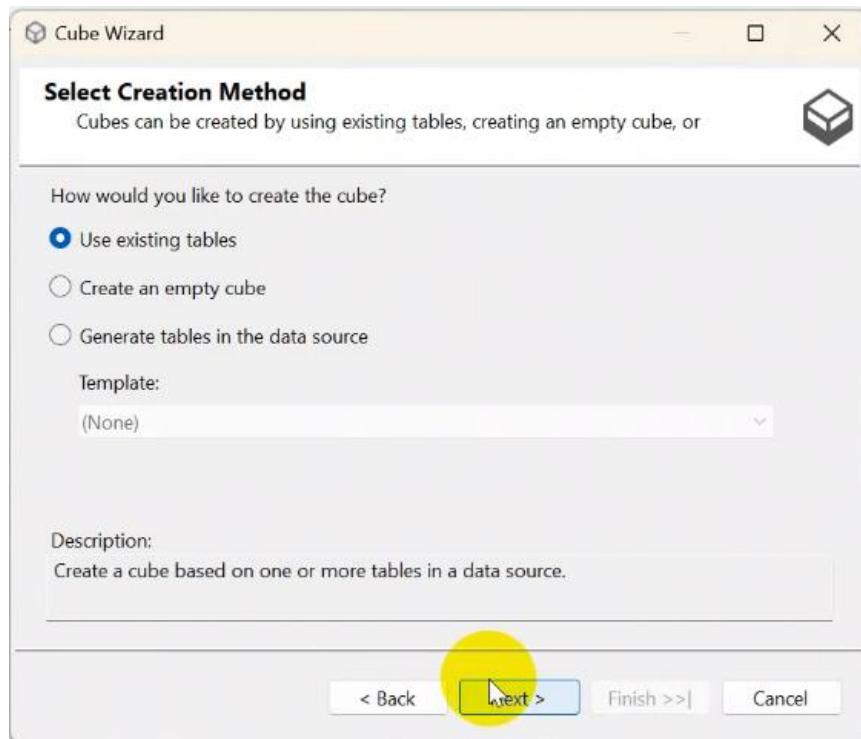
**Bước 1:** Trên thanh công cụ “Solution Explorer”. Click chuột phải vào “Cubes” và nhấn chọn “New Cube”



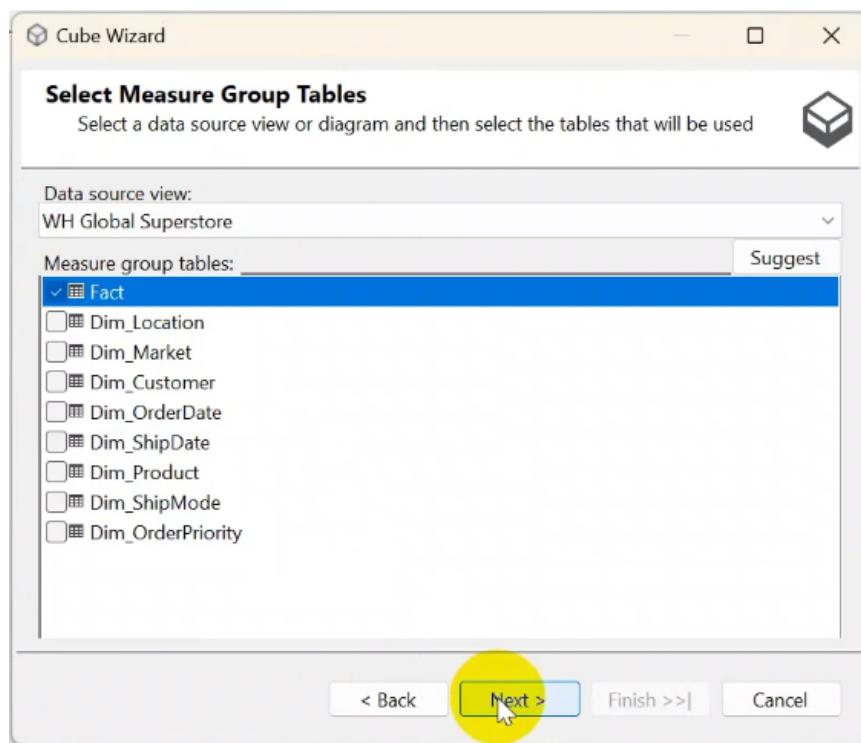
**Bước 2:** Màn hình thông báo **Cube Wizard** hiện ra. Nhấn **Next** để tiếp tục



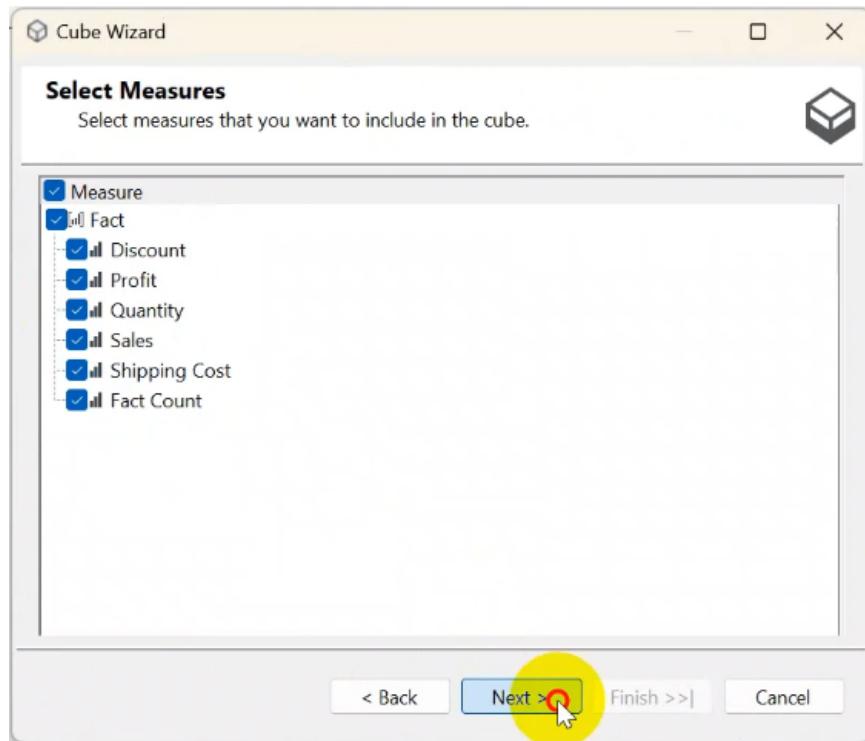
**Bước 3:** Chọn **Use existing tables**, sau đó chọn **Next** để tiếp tục



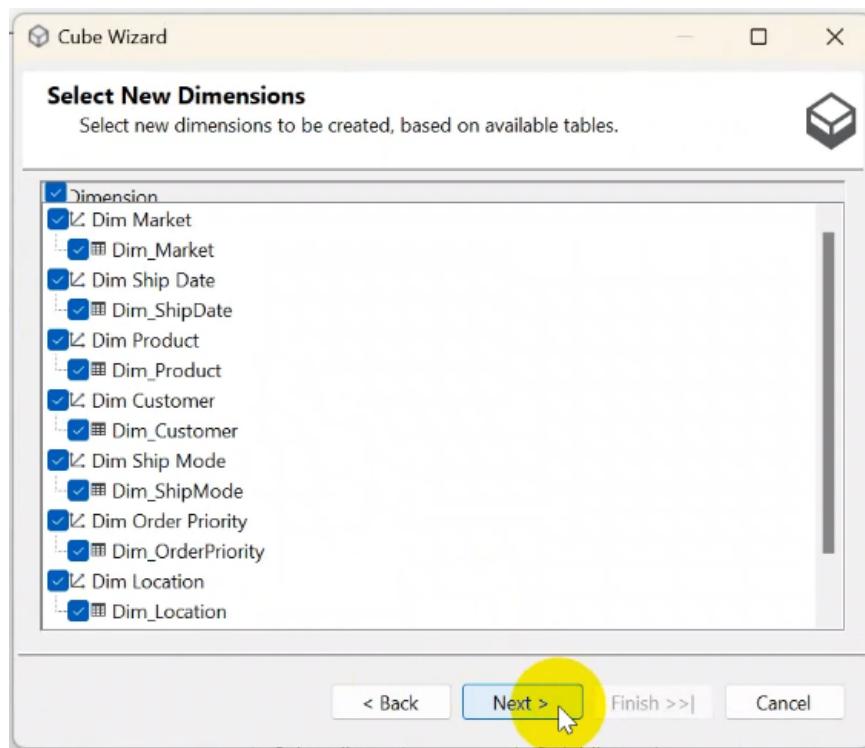
**Bước 4:** Ở mục **Measure group tables**. Chọn bảng **Fact** và nhấn “Next”



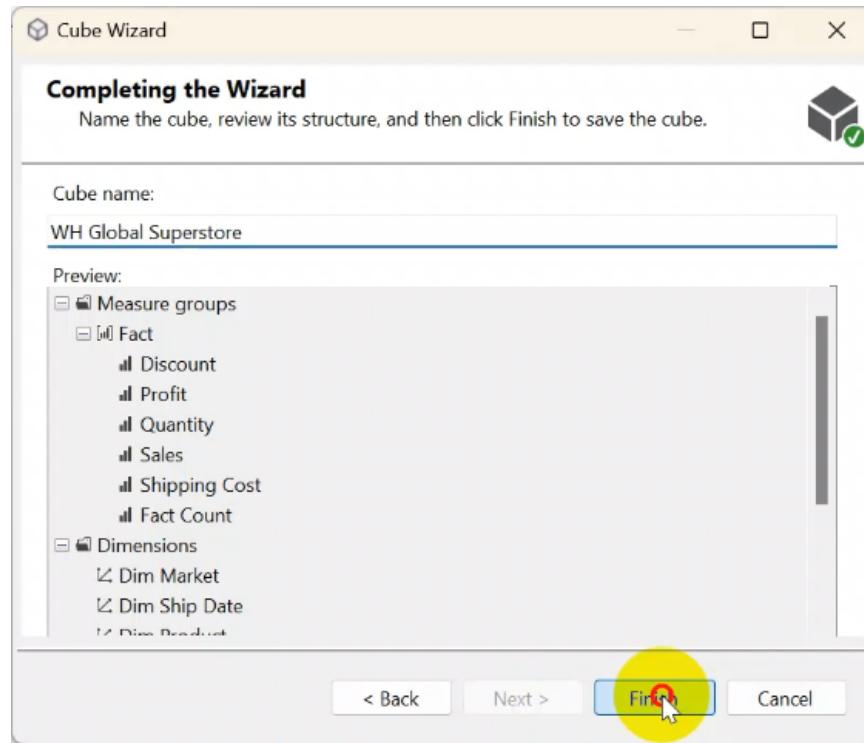
**Bước 5:** Chọn các độ đo **Measure** và nhấn “Next”



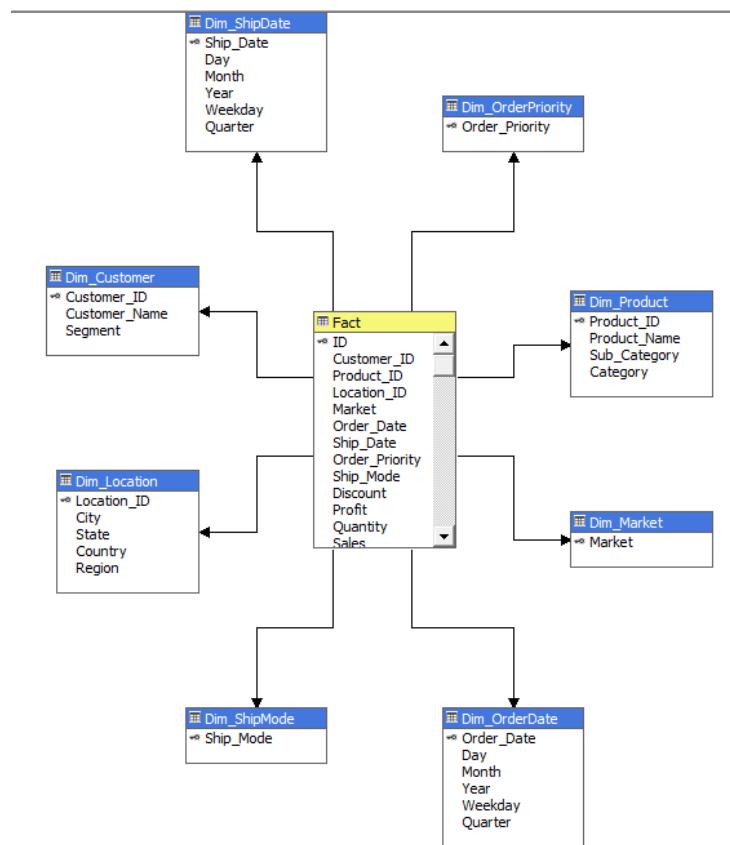
**Bước 6:** Chọn các bảng **Dimension**, sau đó chọn **Next** để tiếp tục.



**Bước 7:** Đặt tên **Cube** và nhấn “Finish” để hoàn tất quá trình tạo **Cubes**



Sau khi hoàn tất ta có kết quả như hình dưới đây:

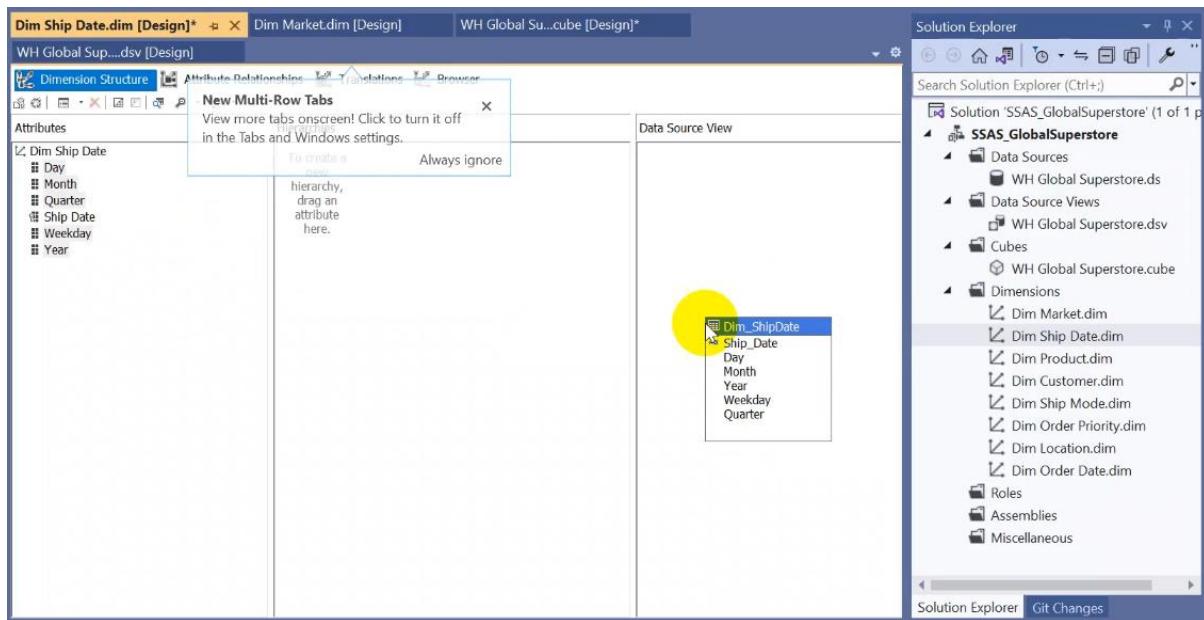


Hình 4.3: Kết quả quá trình tạo Cubes

#### 4.1.3.4 Xác định thuộc tính các Dimensions

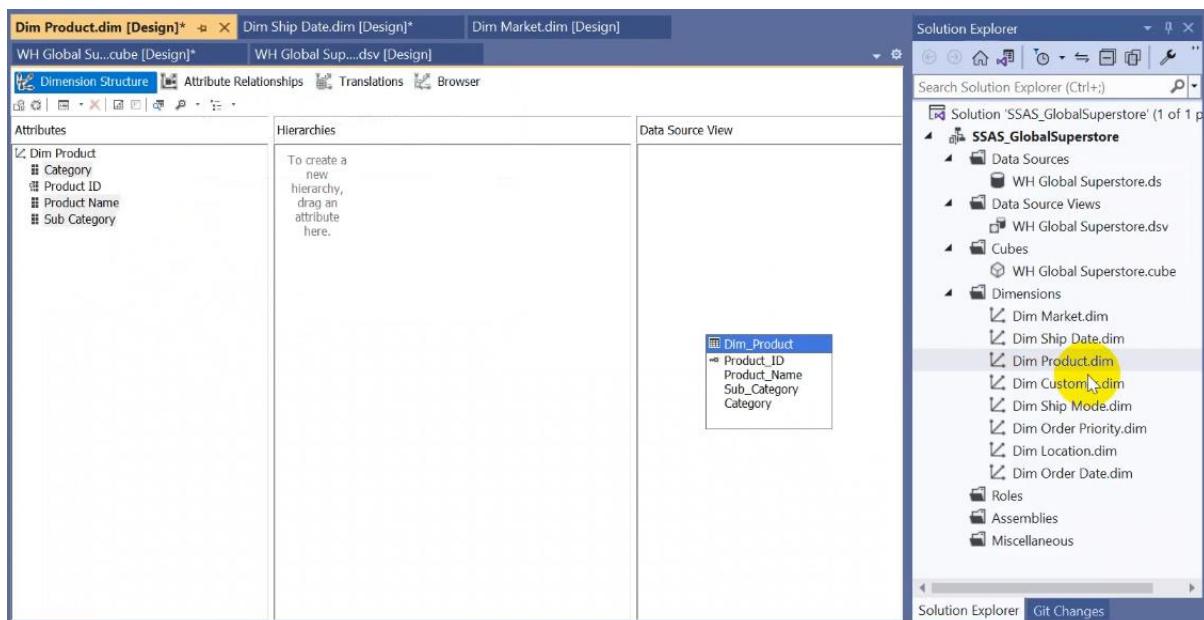
##### Bước 1: Xác định thuộc tính bảng Dim\_ShipDate

Tại thư mục Dimensions ta nhấn vào Dim Ship Date.dim và kéo các thuộc tính Day, Month, Year, Weekday, Quarter từ Data Source View vào Attributes



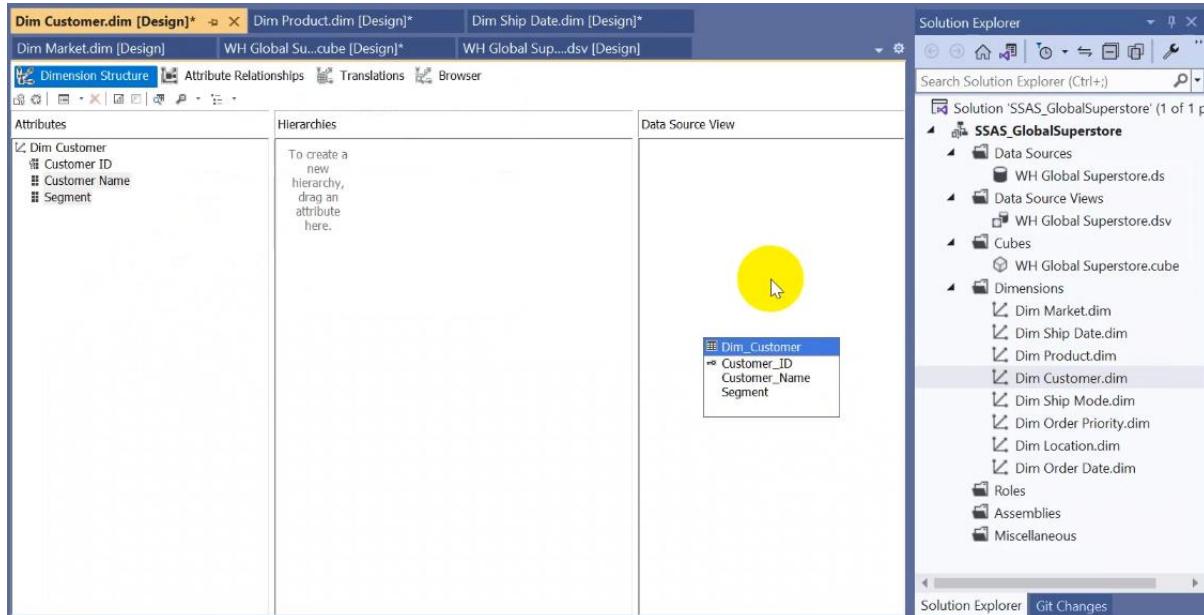
##### Bước 2: Xác định thuộc tính bảng Dim\_Product

Tại thư mục Dimensions ta nhấn vào Dim Product.dim và kéo các thuộc tính Product\_Name, Sub\_Category, Category từ Data Source View vào Attributes



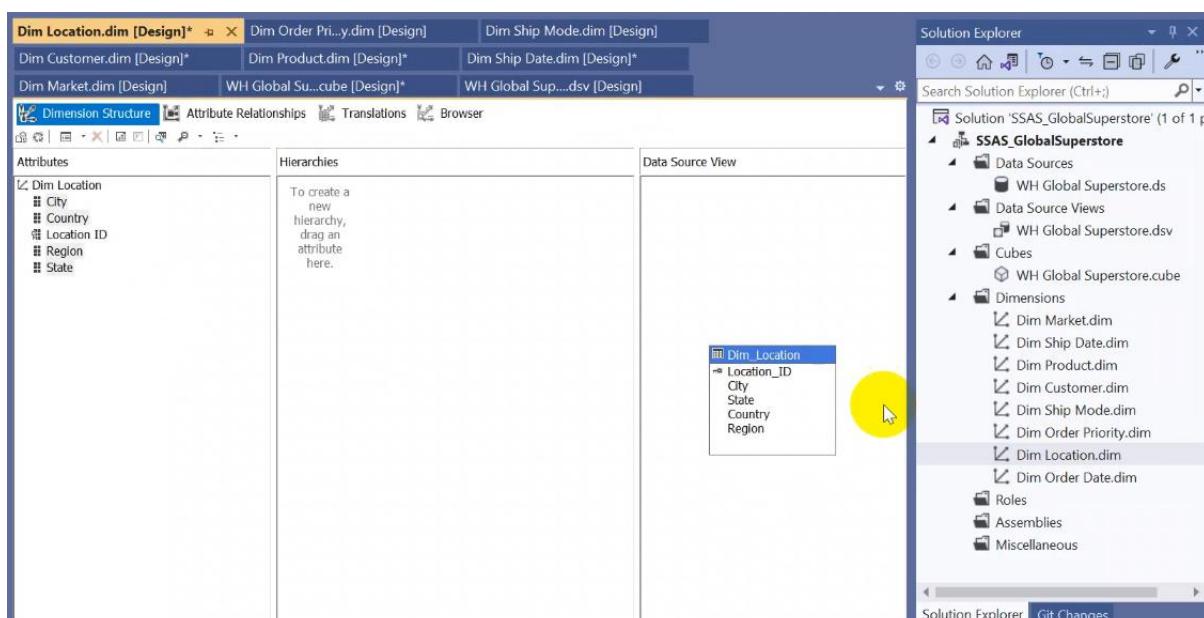
### Bước 3: Xác định thuộc tính bảng Dim\_Customer

Tại thư mục Dimensions ta nhấn vào Dim\_Customer.dim và kéo các thuộc tính Customer\_Name, Segment từ Data Source View vào Attributes



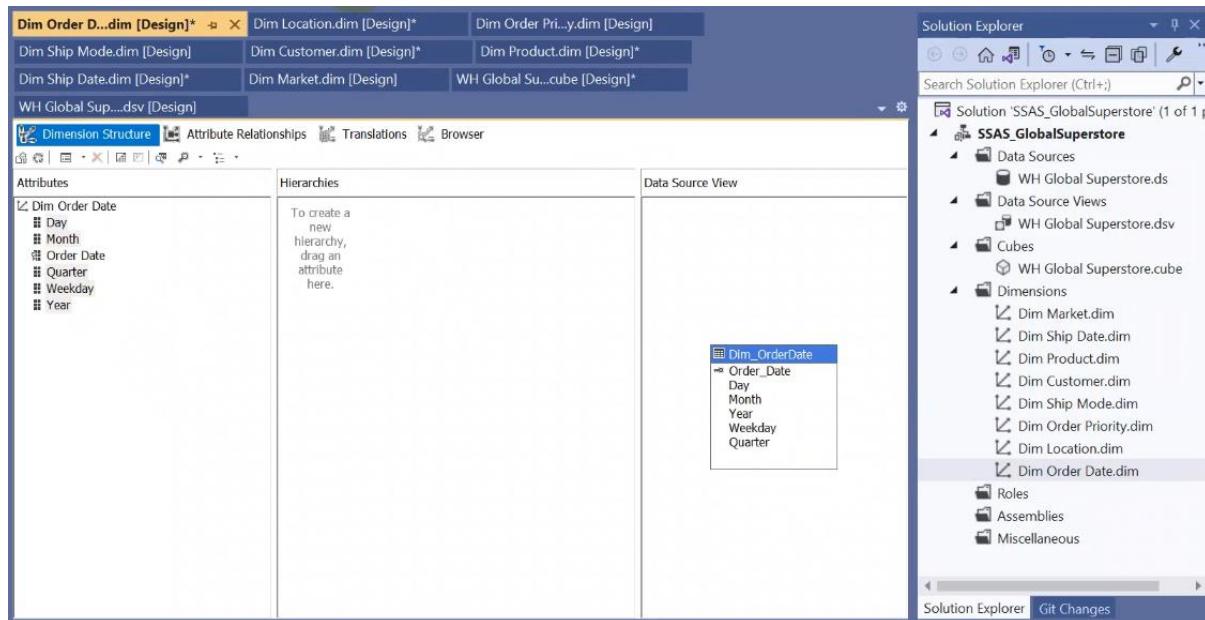
### Bước 4: Xác định thuộc tính bảng Dim\_Location

Tại thư mục Dimensions ta nhấn vào Dim\_Location.dim và kéo các thuộc tính City, State, Country, Region từ Data Source View vào Attributes

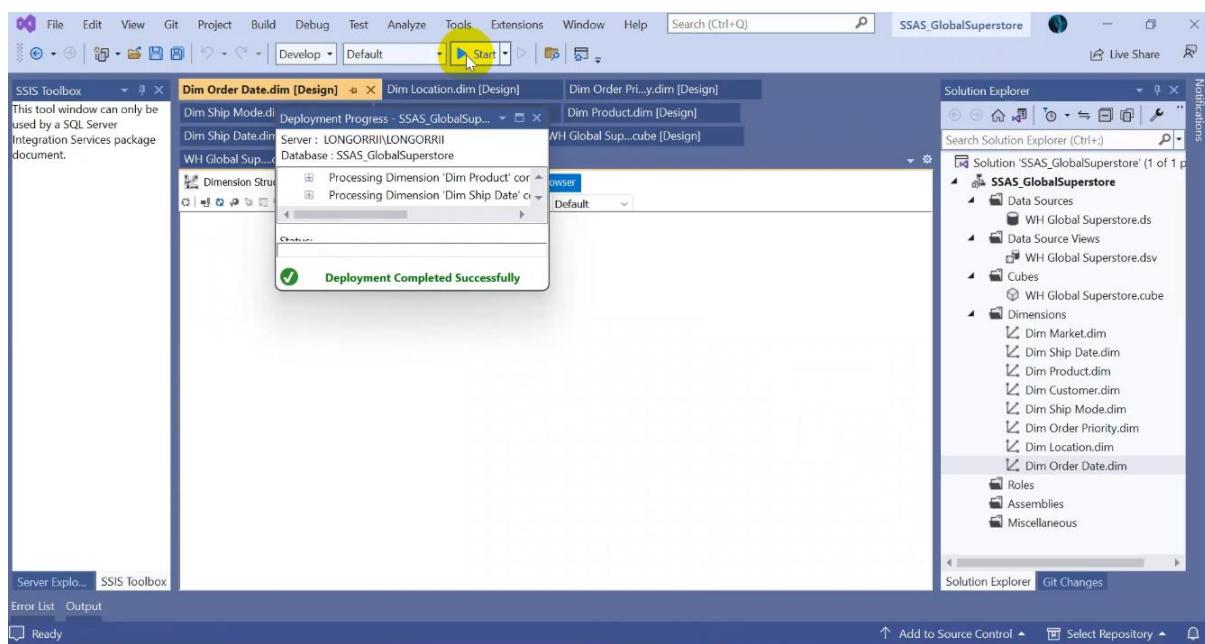


### Bước 5: Xác định thuộc tính bảng Dim\_OrderDate

Tại thư mục **Dimensions** ta nhấn vào **Dim Orders Date.dim** và kéo các thuộc tính **Day, Month, Year, Weekday, Quarter** từ **Data Source View** vào **Attributes**



**Bước 6:** Ta nhấn Start để deploy project. Khi không xảy ra lỗi ta sẽ nhận được kết quả như sau:

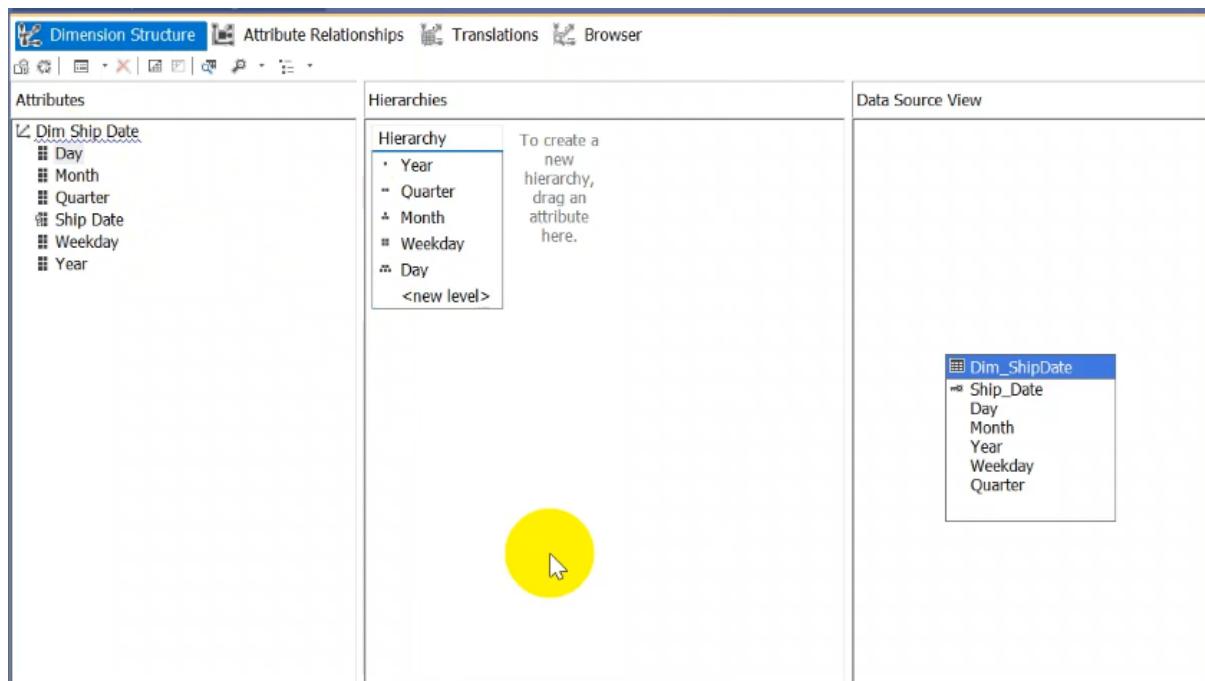


#### 4.1.3.5 Phân cấp (hierarchy) trong bảng Dimensions

##### 4.1.3.5.1 Phân cấp bảng Dim\_ShipDate và Dim\_OrderDate

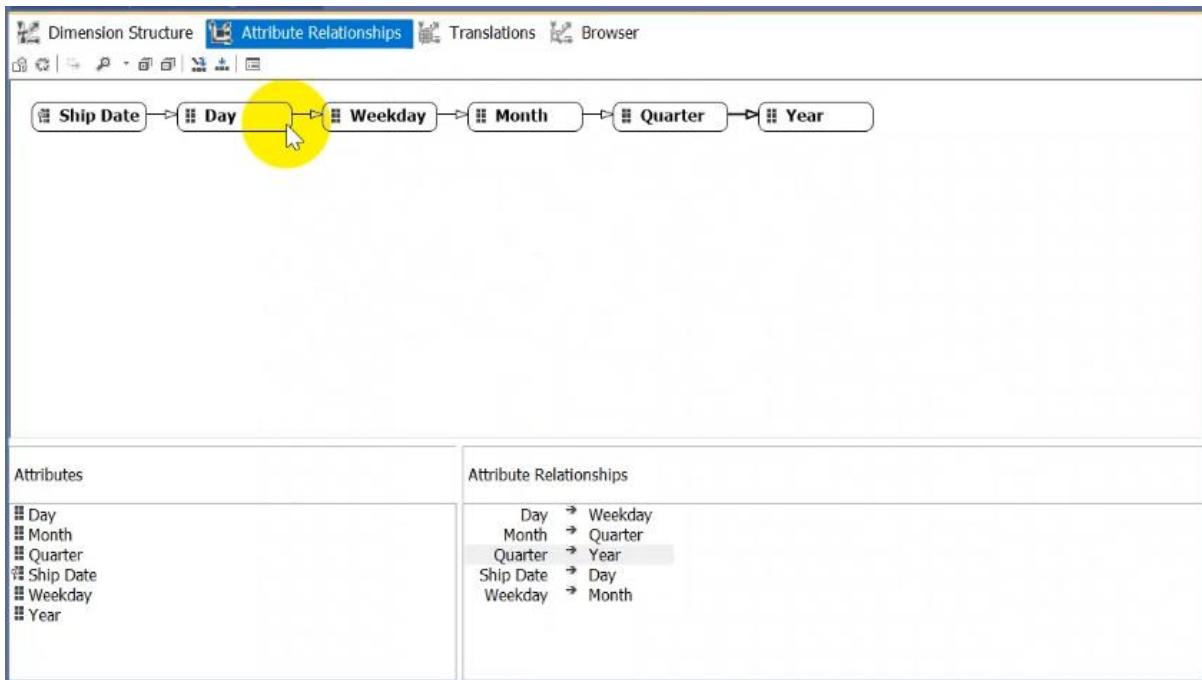
Vì hai bảng **Dim Ship Date** và **Dim Order Date** này có cấu trúc phân cấp thuộc tính giống nhau, nên chúng ta sẽ mô tả chi tiết phân cấp của một trong hai bảng. Dưới đây là mô tả phân cấp của bảng **Dim Ship Date**:

**Bước 1:** Tại **Dimension Structure** ta kéo những thuộc tính cần phân cấp qua cửa sổ Hierarchies. Và sắp xếp theo thứ tự phân cấp như sau: Year -> Quarter -> Month -> Weekday -> Day



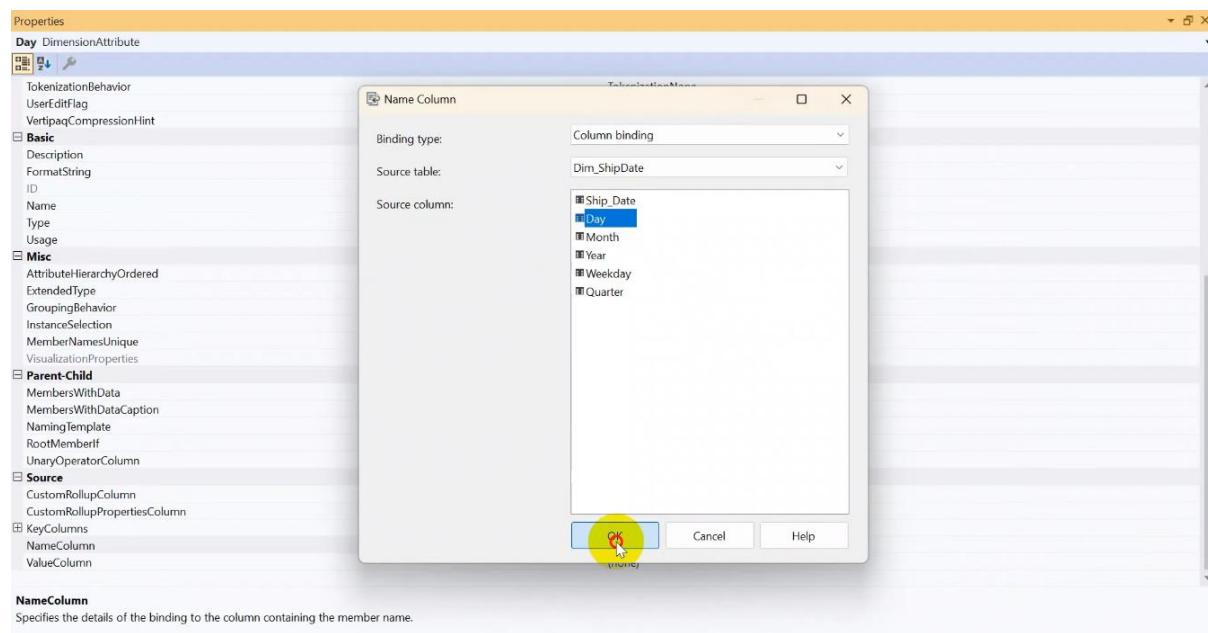
The screenshot shows the 'Dimension Structure' interface. The 'Hierarchies' pane has a 'Hierarchy' button highlighted with a yellow circle. A tooltip says: 'To create a new hierarchy, drag an attribute here.' The 'Attributes' pane lists attributes for 'Dim\_ShipDate': Day, Month, Quarter, Ship Date, Weekday, and Year. The 'Data Source View' pane shows the structure of 'Dim\_ShipDate' with attributes: Ship Date, Day, Month, Year, Weekday, and Quarter.

**Bước 2:** Tại **Attribute Relationships** ta tạo sơ đồ mối quan hệ như sau.

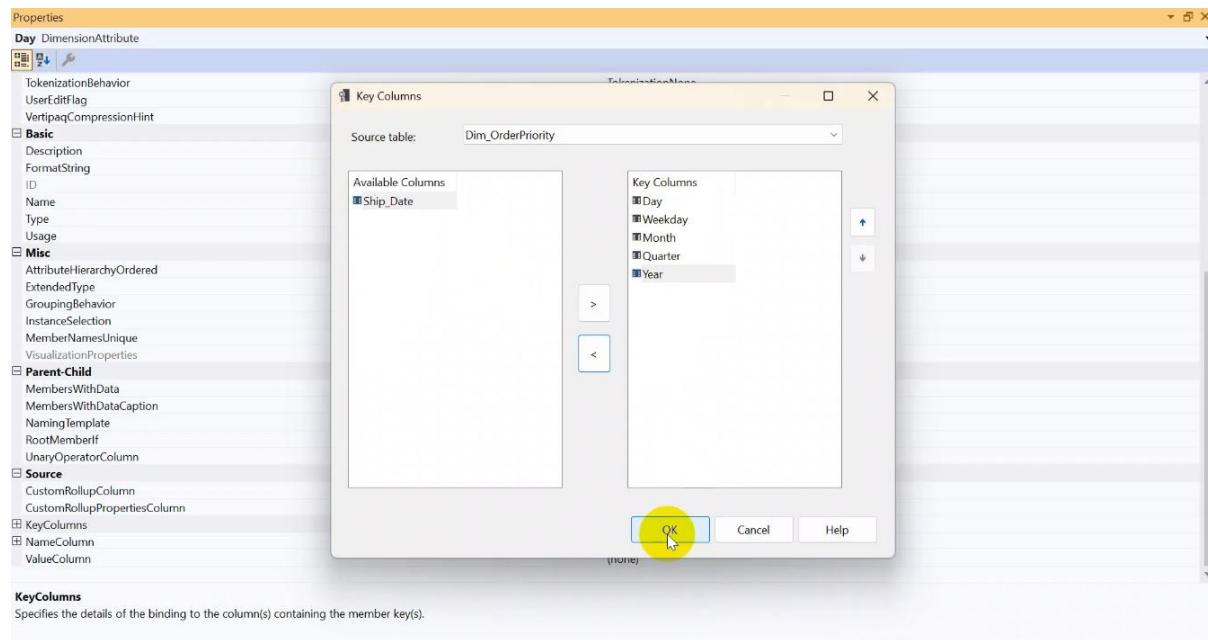


**Bước 3: Chỉnh KeyColumns và Name Column** của thuộc tính **Day**. Cấu hình **KeyColumns** bao gồm các cột từ các cấp cao hơn như **Year**, **Quarter**, **Month**, **Weekday**, và thiết lập **NameColumn** cho thuộc tính **Day**.

Tại cửa sổ **Properties** của thuộc tính **Day**, ta chọn **Name Column** là **Day** và nhấn **OK**

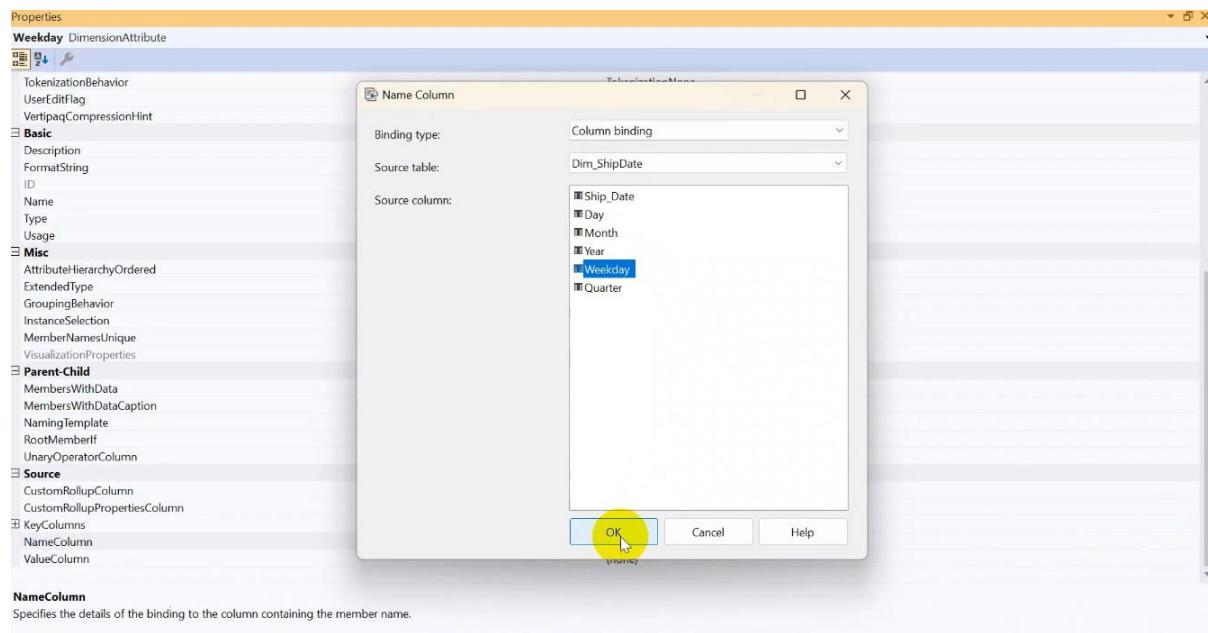


Tại cửa sổ **Properties** ta thêm các những thuộc tính cấp cao hơn **Day** vào **KeyColumns**, sau đó chọn **OK**

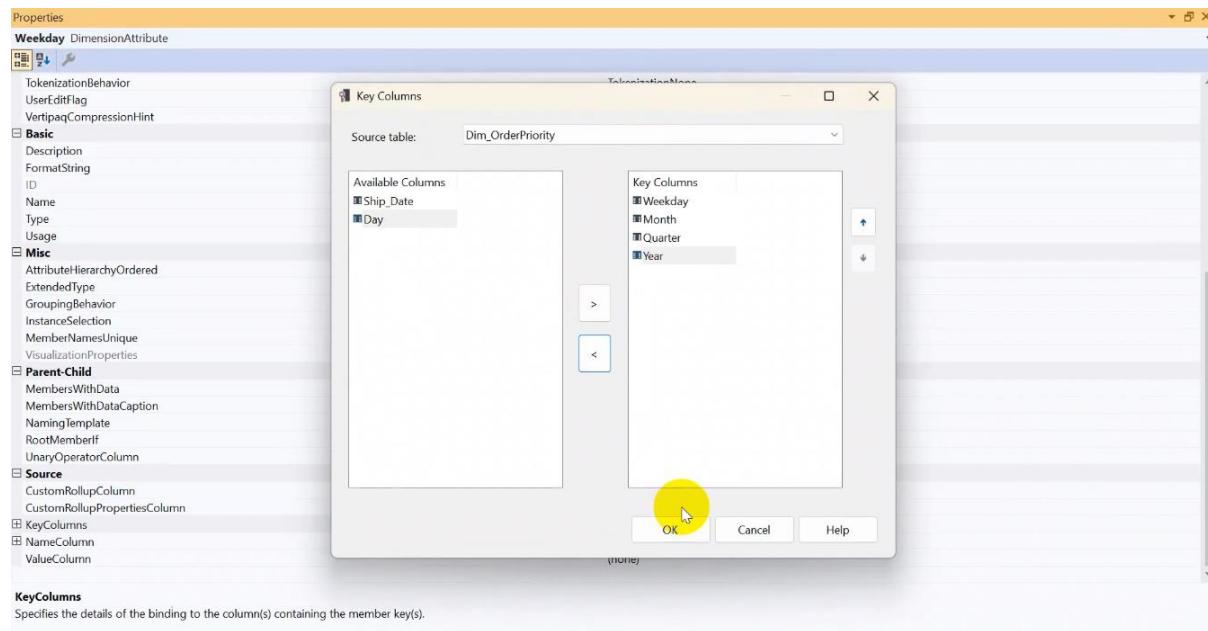


**Bước 4:** Chính **KeyColumns** và **Name Column** của thuộc tính **Weekday**. Cấu hình **KeyColumns** bao gồm các cột từ các cấp cao hơn như **Year**, **Quarter**, và **Month**, và thiết lập **NameColumn** cho thuộc tính **Weekday**.

Tại cửa sổ **Properties** của thuộc tính **Weekday**, ta chọn **Name Column** là **Weekday** và nhấn **OK**

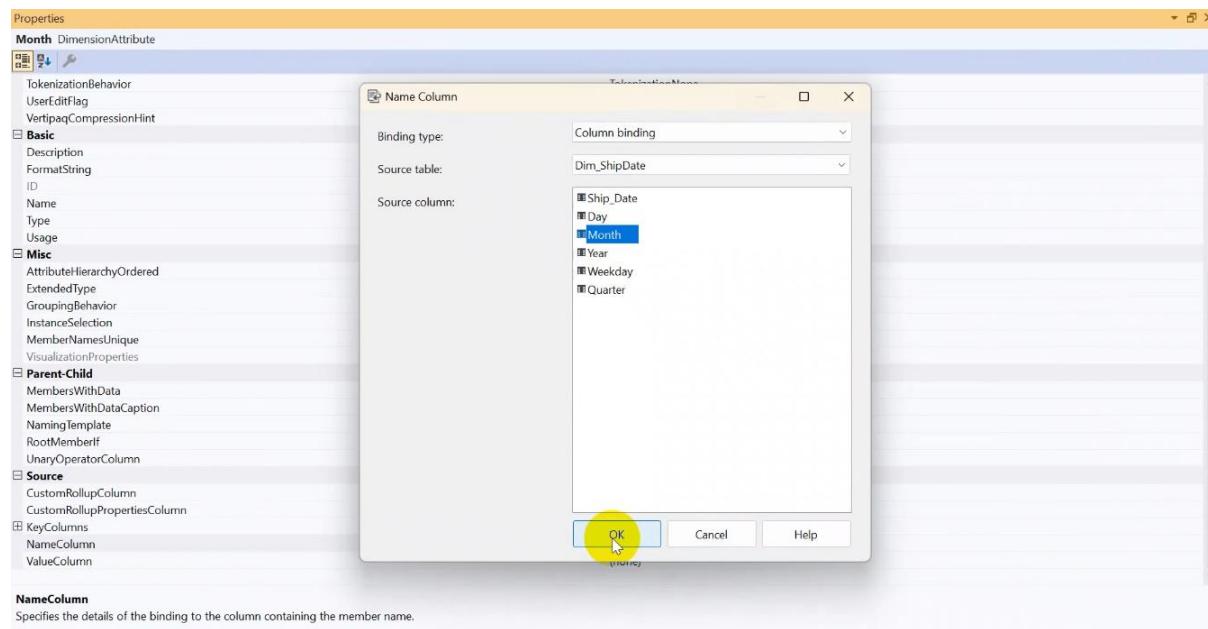


Tại cửa sổ **Properties** ta thêm các những thuộc tính cấp cao hơn **Weekday** vào **KeyColumns**, sau đó chọn **OK**

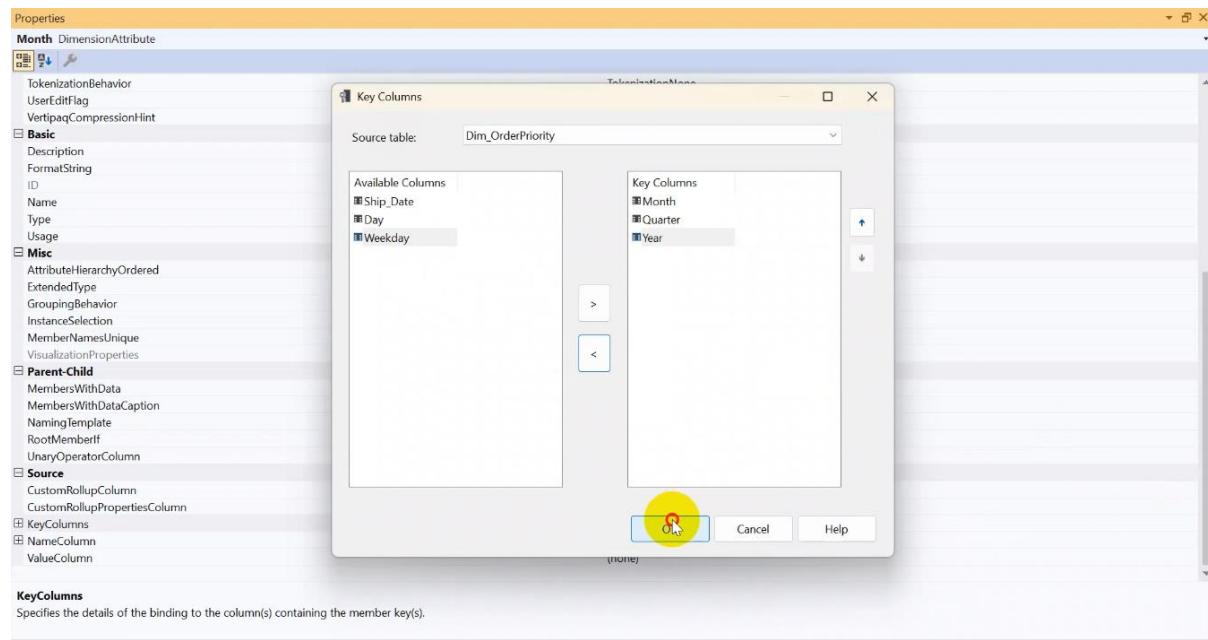


**Bước 5: Chỉnh KeyColumns và Name Column của thuộc tính Month. Cấu hình KeyColumns bao gồm các cột từ các cấp cao hơn như Year, Quarter, và thiết lập NameColumn cho thuộc tính Month.**

Tại cửa sổ Properties của thuộc tính Month, ta chọn Name Column là Month và nhấn OK

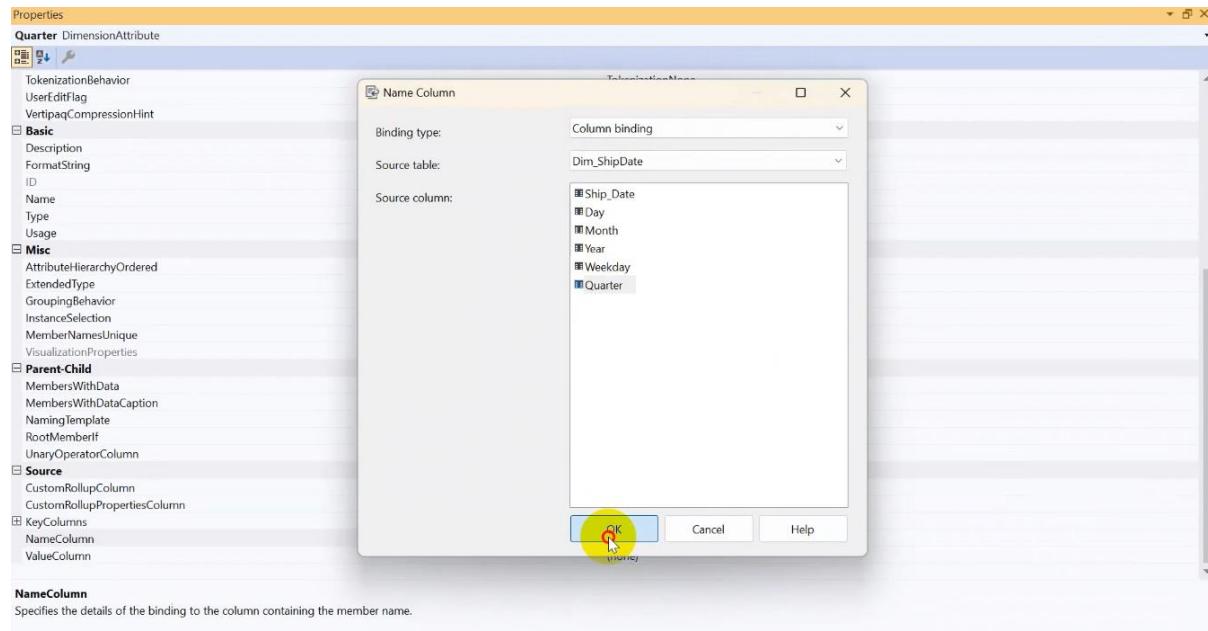


Tại cửa sổ Properties ta thêm các những thuộc tính cấp cao hơn Month vào KeyColumns, sau đó chọn OK

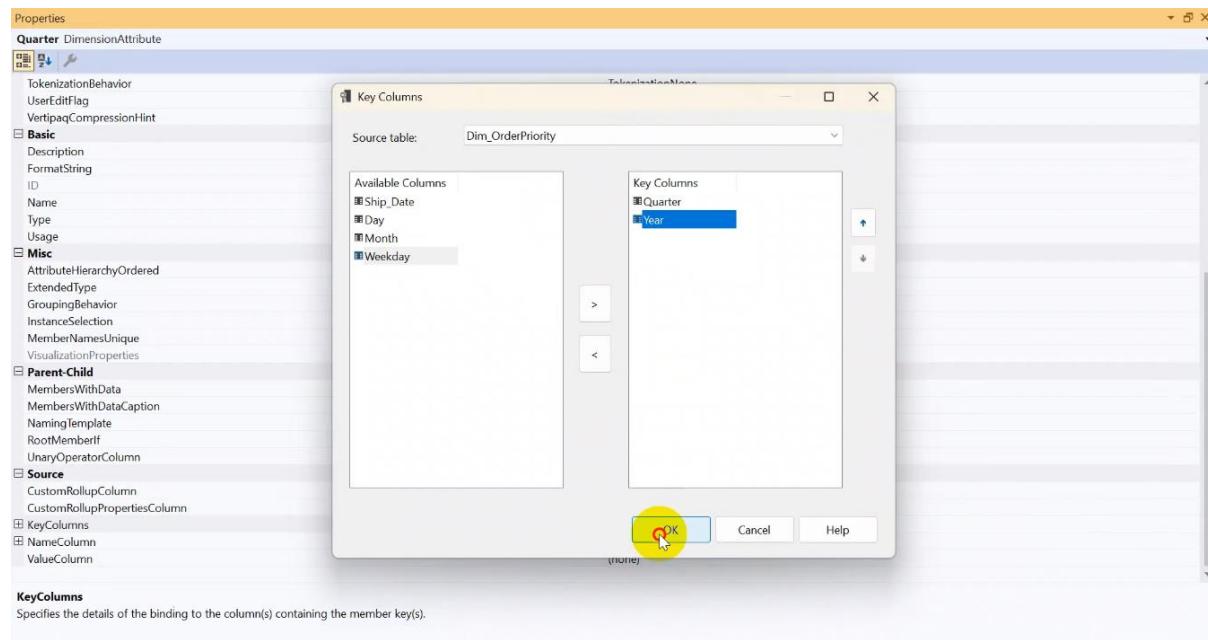


**Bước 6: Chính KeyColumns và Name Column của thuộc tính Quarter. Cấu hình KeyColumns bao gồm các cột từ các cấp cao hơn như Year và thiết lập NameColumn cho thuộc tính Quarter.**

Tại cửa sổ Properties của thuộc tính Quarter, ta chọn Name Column là Quarter và nhấn OK

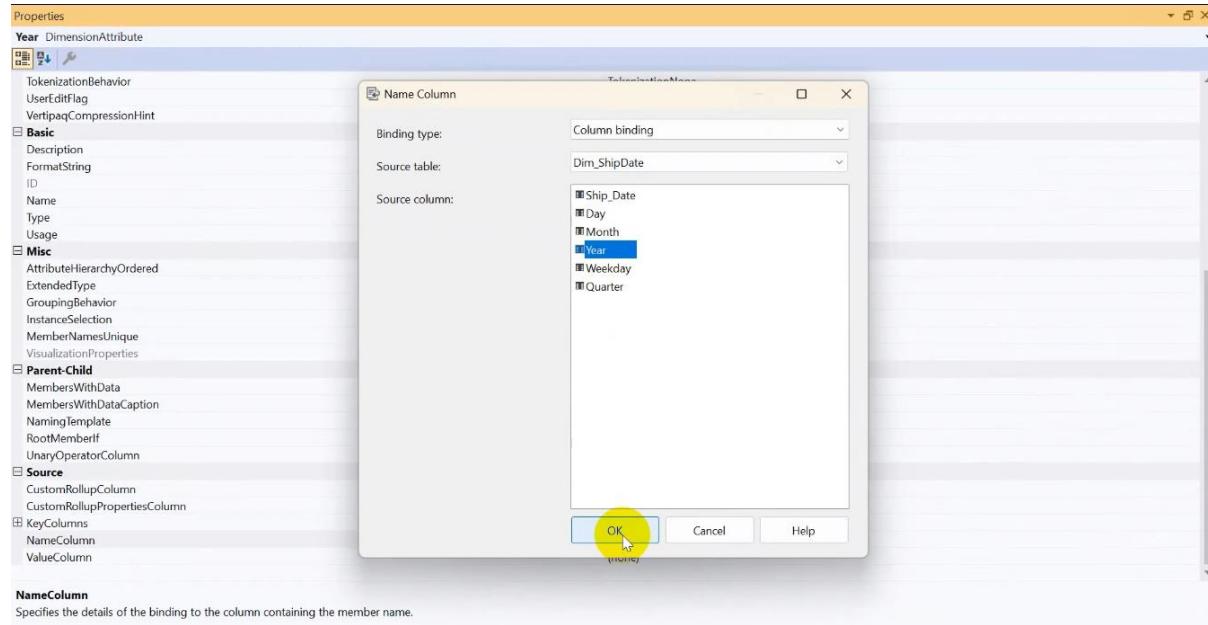


Tại cửa sổ Properties ta thêm các những thuộc tính cấp cao hơn Quarter vào KeyColumns, sau đó chọn OK



**Bước 7: Chính Name Column của thuộc tính Year.** Bởi vì Year là thuộc tính cấp cao nhất nên không cần thiết lập KeyColumns

Tại cửa sổ **Properties** của thuộc tính Year, ta chọn **Name Column** là Year và nhấn **OK**



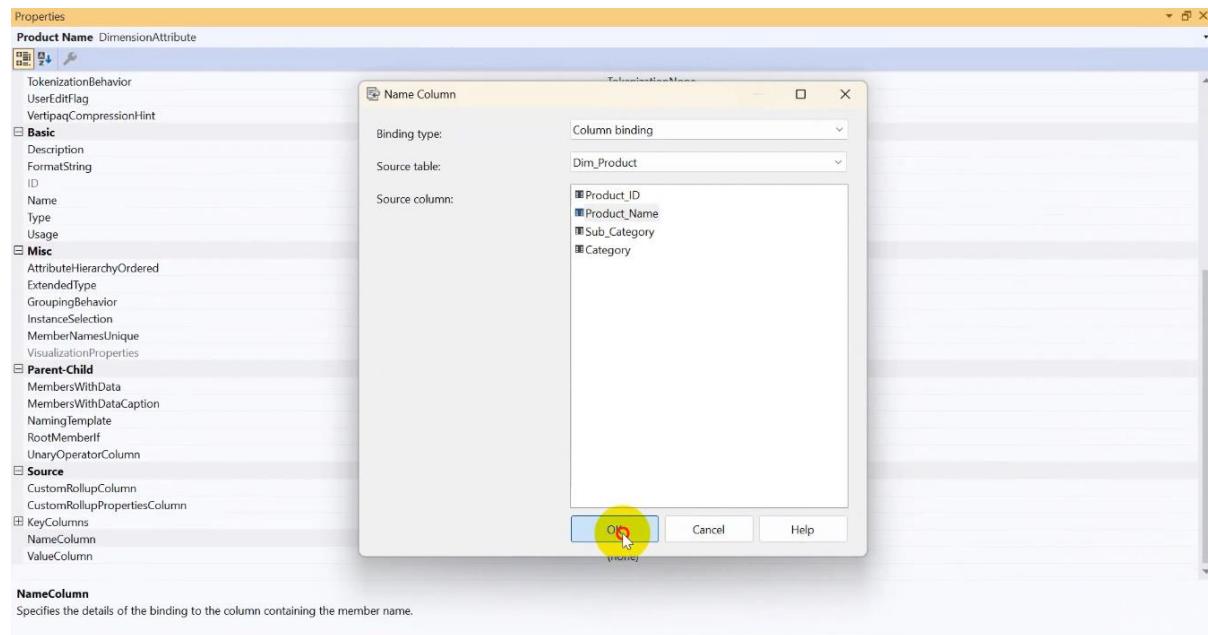
#### 4.1.3.5.2 Phân cấp bảng Dim\_Product

**Bước 1:** Tại **Dimension Structure** ta kéo những thuộc tính cần phân cấp qua cửa sổ Hierarchies. Và sắp xếp theo thứ tự phân cấp như sau: Category -> Sub Category -> Product Name

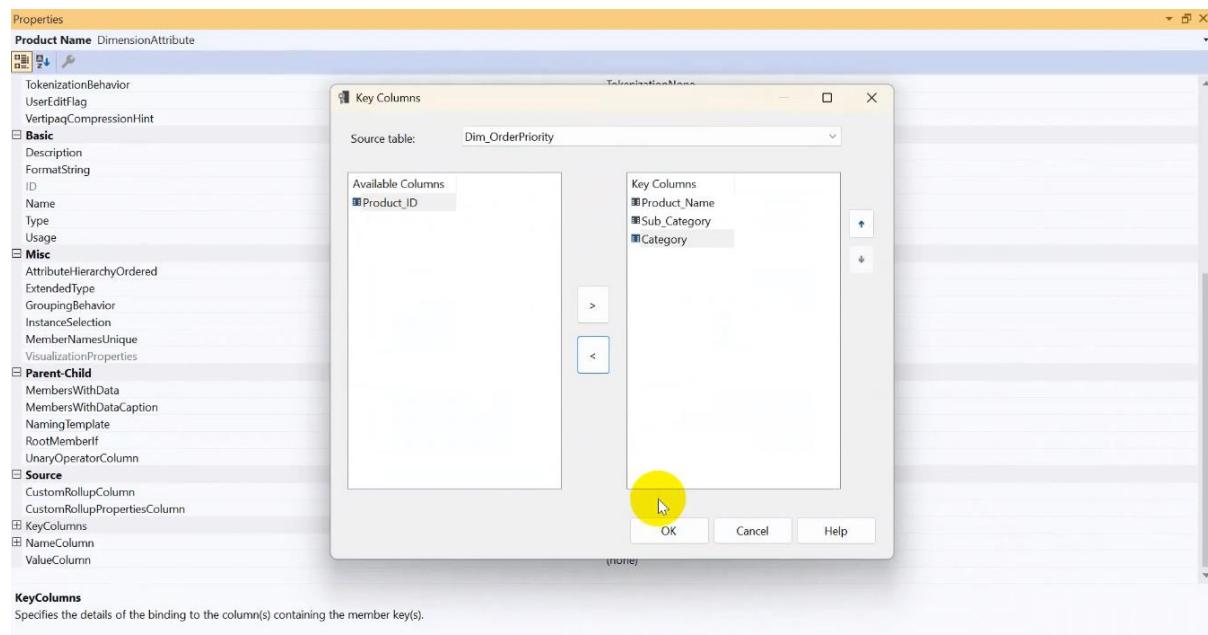
**Bước 2:** Tại **Attribute Relationships** ta tạo sơ đồ mối quan hệ như sau.

**Bước 3:** Chỉnh **KeyColumns** và **Name Column** của thuộc tính **Product Name**. Cấu hình **KeyColumns** bao gồm các cột từ các cấp cao hơn như **Category**, **Sub Category** và thiết lập **NameColumn** cho thuộc tính **Product Name**.

Tại cửa sổ **Properties** của thuộc tính **Product Name**, ta chọn **Name Column** là **Product Name** và nhấn **OK**

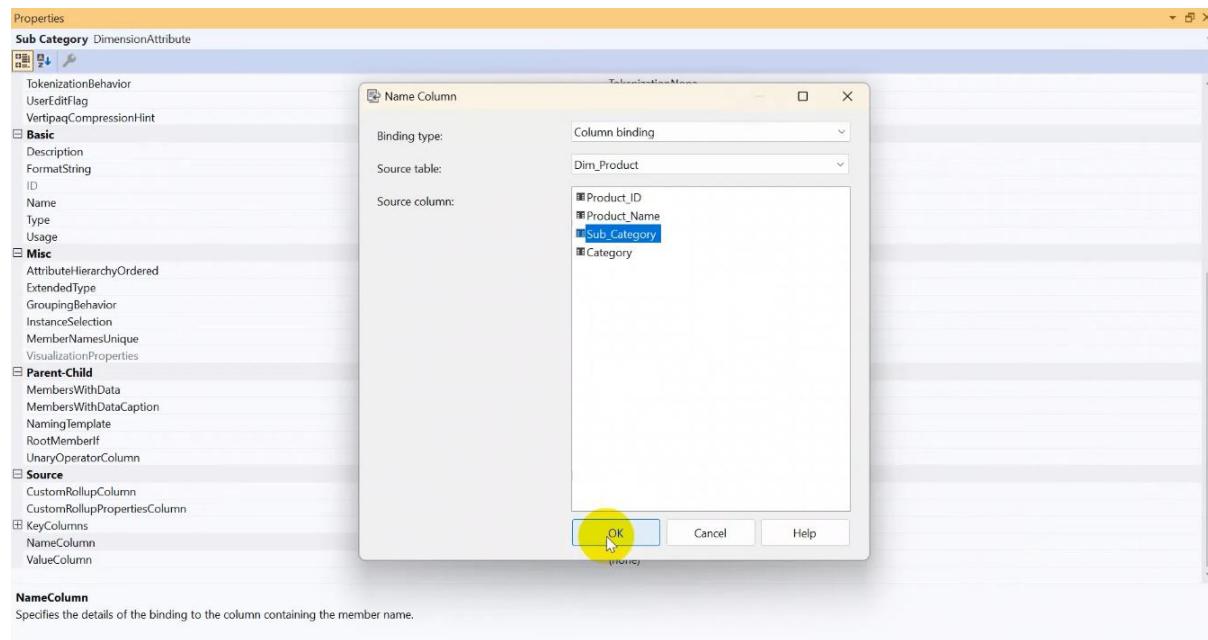


Tại cửa sổ **Properties** ta thêm các những thuộc tính cấp cao hơn **Product Name** vào **KeyColumns**, sau đó chọn **OK**

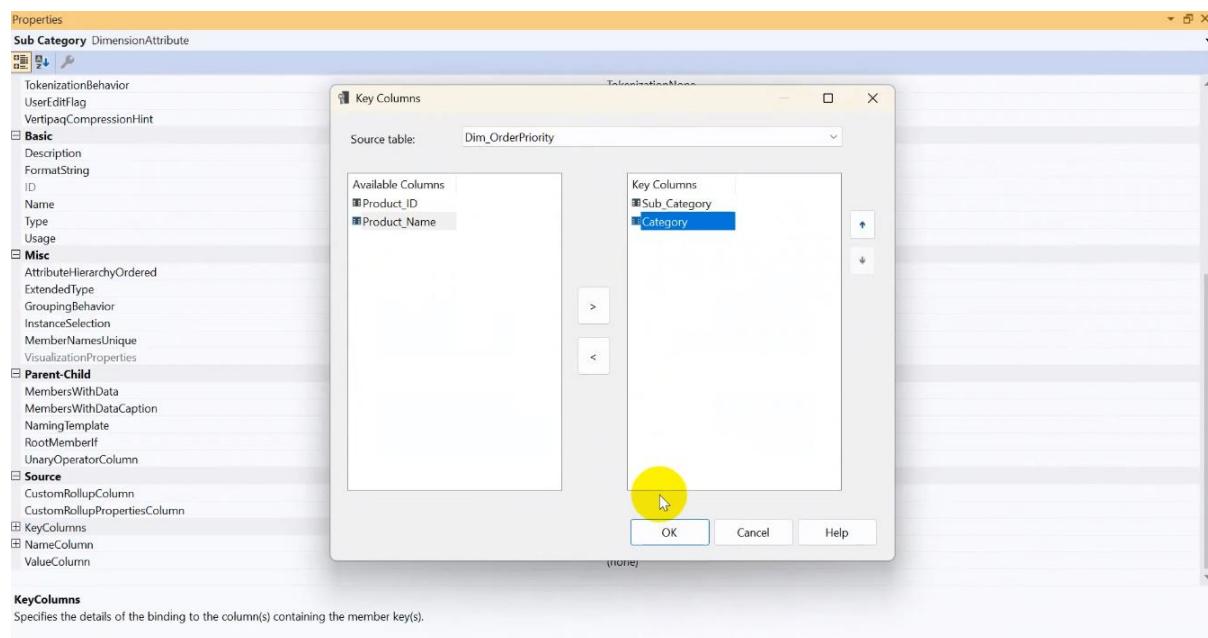


**Bước 4:** Chỉnh **KeyColumns** và **Name Column** của thuộc tính **Sub Category**. Cấu hình **KeyColumns** bao gồm các cột từ các cấp cao hơn như **Category** và thiết lập **NameColumn** cho thuộc tính **Sub Category**.

Tại cửa sổ **Properties** của thuộc tính **Sub Category**, ta chọn **Name Column** là **Sub Category** và nhấn **OK**

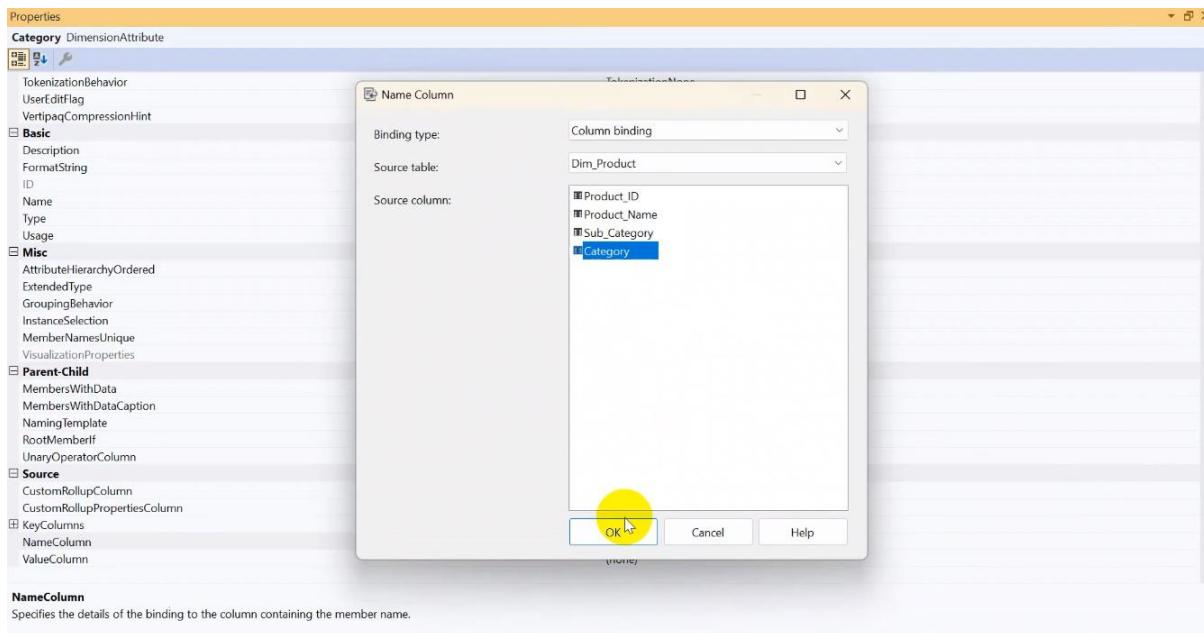


Tại cửa sổ **Properties** ta thêm các những thuộc tính cấp cao hơn **Sub Category** vào **KeyColumns**, sau đó chọn **OK**



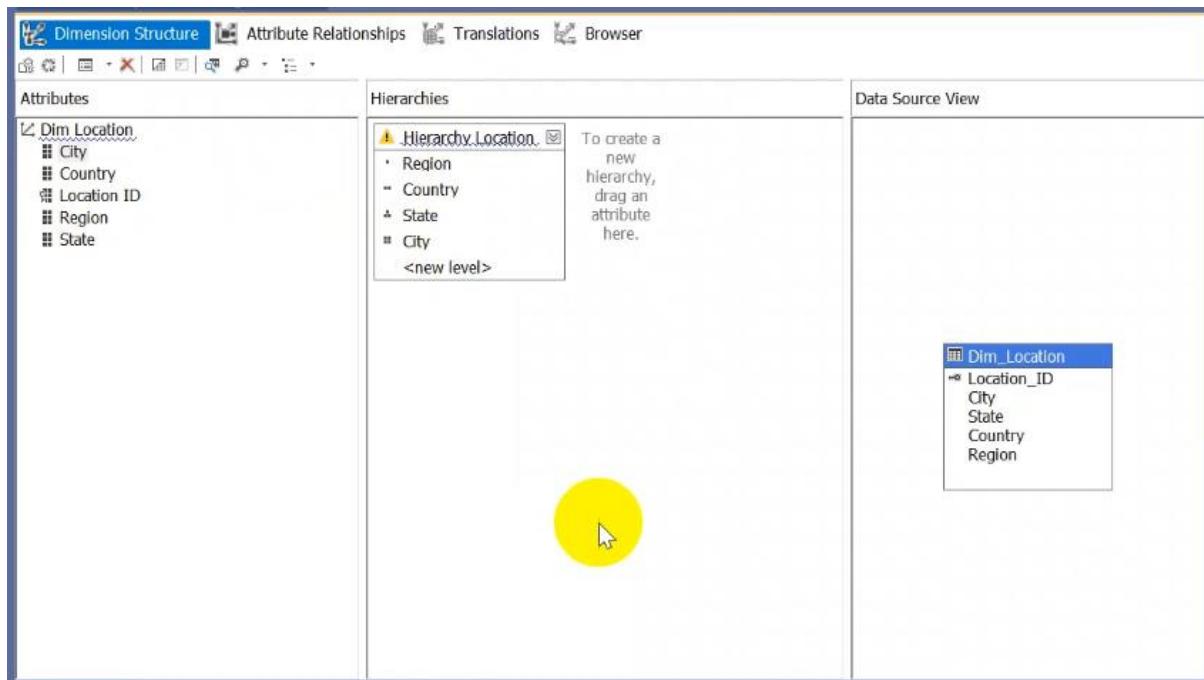
**Bước 5:** Chỉnh **Name Column** của thuộc tính **Category**. Bởi vì **Category** là thuộc tính cấp cao nhất nên không cần thiết lập **KeyColumns**

Tại cửa sổ **Properties** của thuộc tính **Category**, ta chọn **Name Column** là **Category** và nhấn **OK**

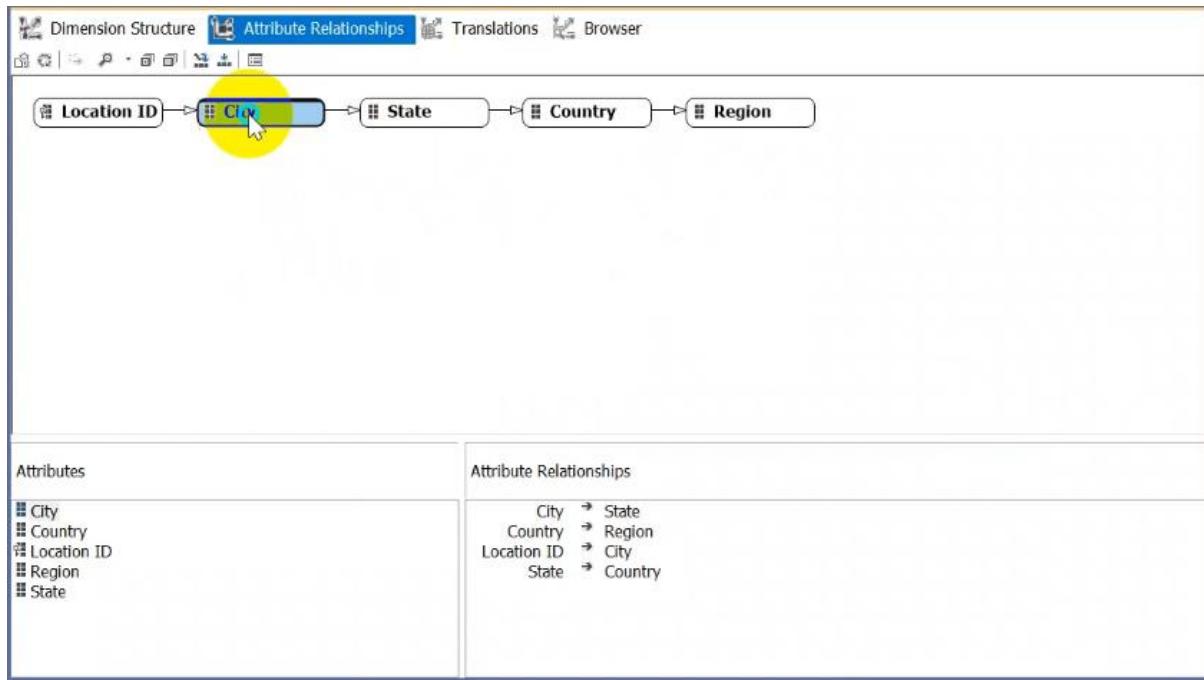


#### 4.1.3.5.3 Phân cấp bảng Dim\_Location

**Bước 1:** Tại **Dimension Structure** ta kéo những thuộc tính cần phân cấp qua cửa sổ **Hierarchies**. Và sắp xếp theo thứ tự phân cấp như sau: Region -> Country -> State -> City

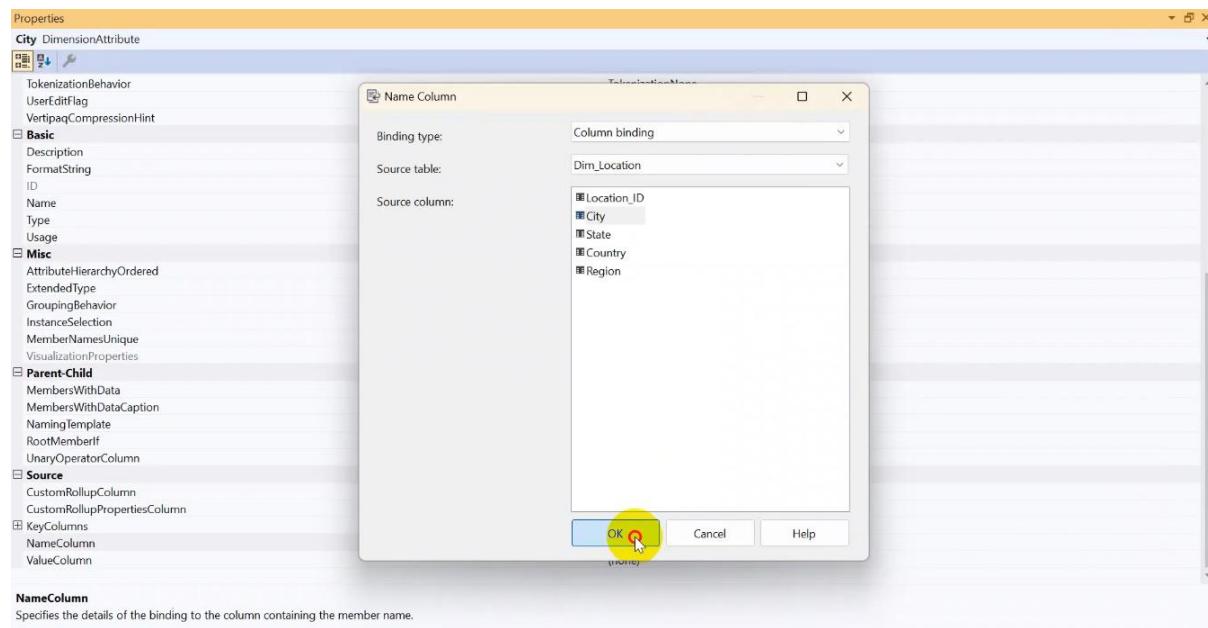


**Bước 2:** Tại **Attribute Relationships** ta tạo sơ đồ mối quan hệ như sau.

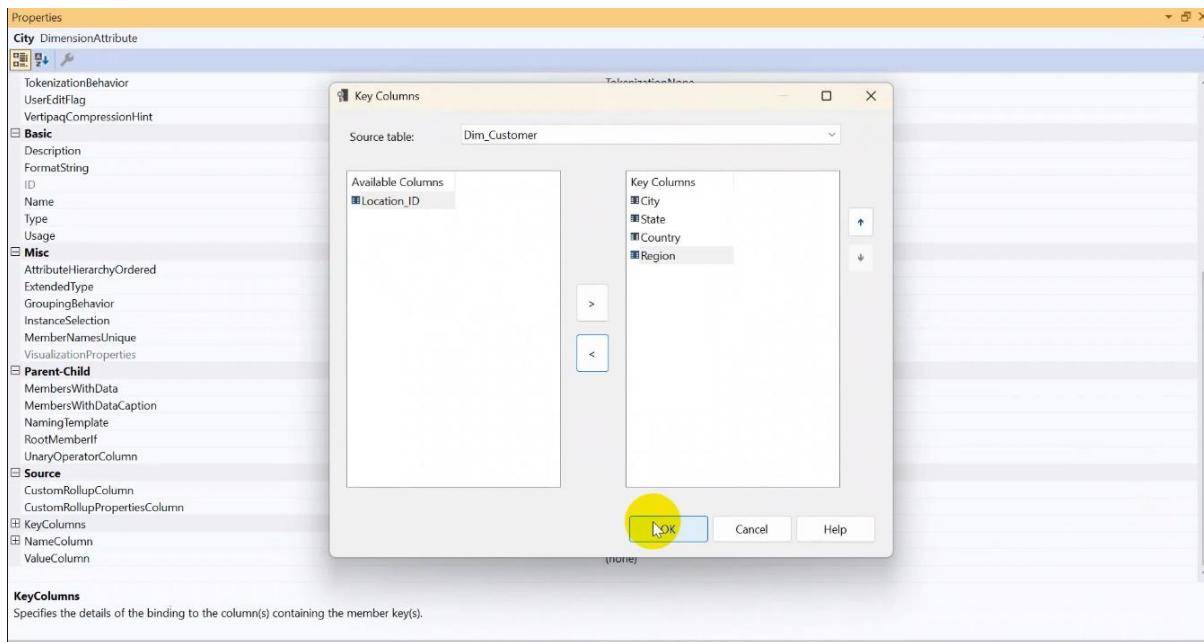


**Bước 3: Chỉnh KeyColumns và Name Column** của thuộc tính City. Cấu hình KeyColumns bao gồm các cột từ các cấp cao hơn như Region, Country, State, và thiết lập NameColumn cho thuộc tính City.

Tại cửa sổ **Properties** của thuộc tính City, ta chọn **Name Column** là City và nhấn **OK**

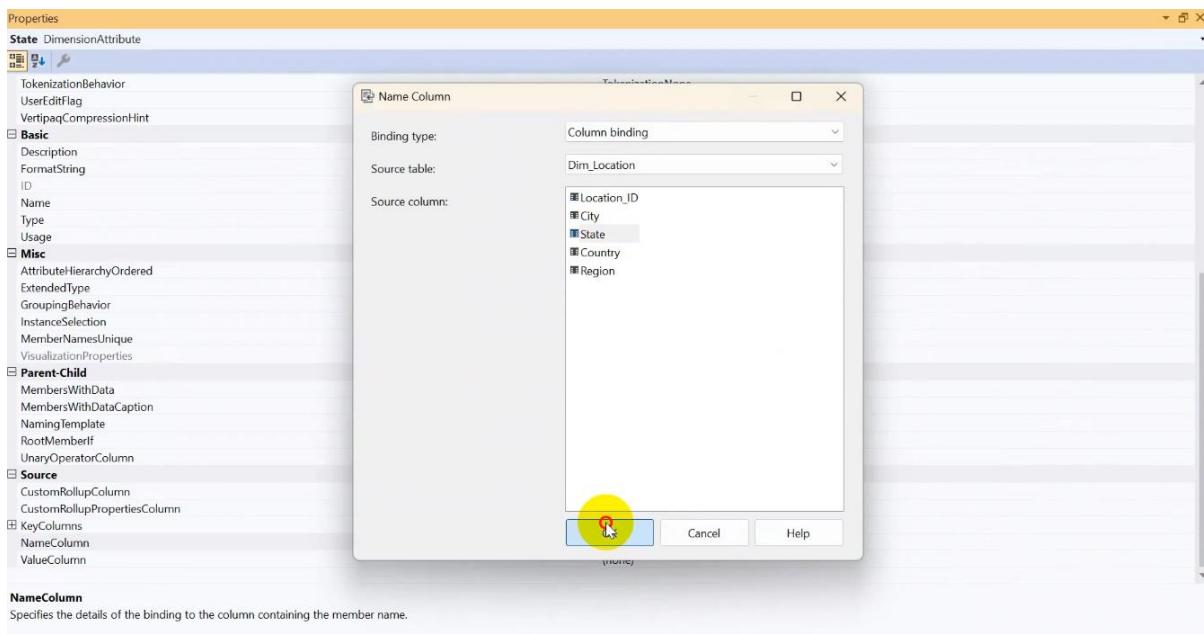


Tại cửa sổ **Properties** ta thêm các những thuộc tính cấp cao hơn City vào KeyColumns, sau đó chọn **OK**

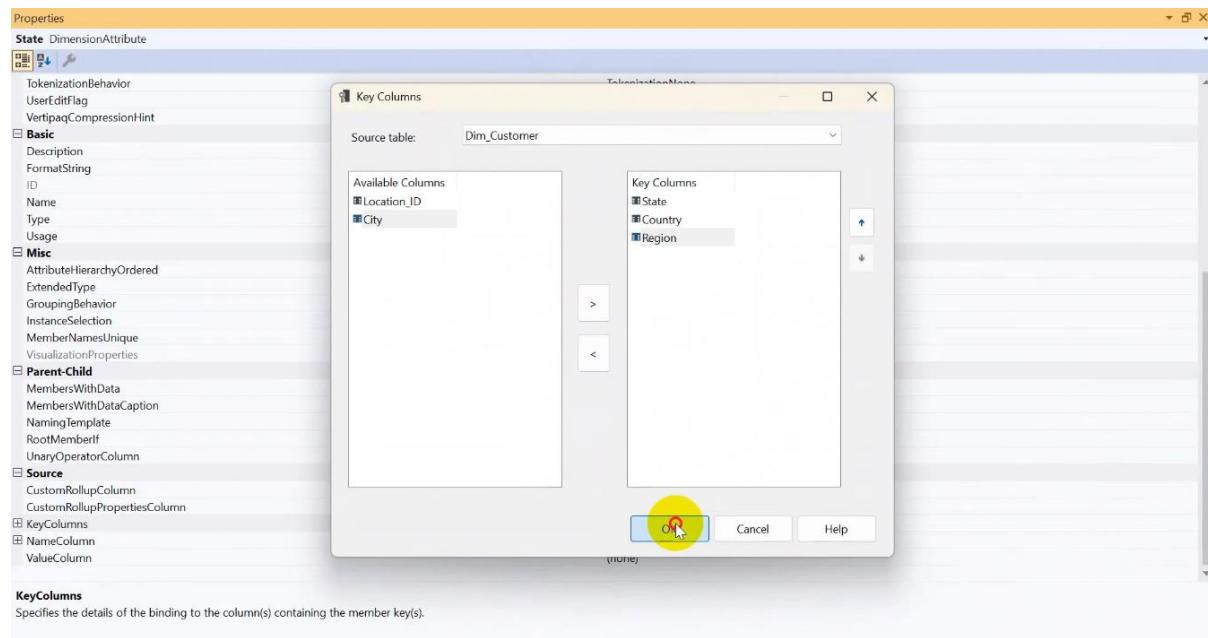


**Bước 4: Chỉnh KeyColumns và Name Column của thuộc tính State. Cấu hình KeyColumns bao gồm các cột từ các cấp cao hơn như Country, Region, và thiết lập NameColumn cho thuộc tính State.**

Tại cửa sổ **Properties** của thuộc tính State, ta chọn **Name Column** là State và nhấn **OK**

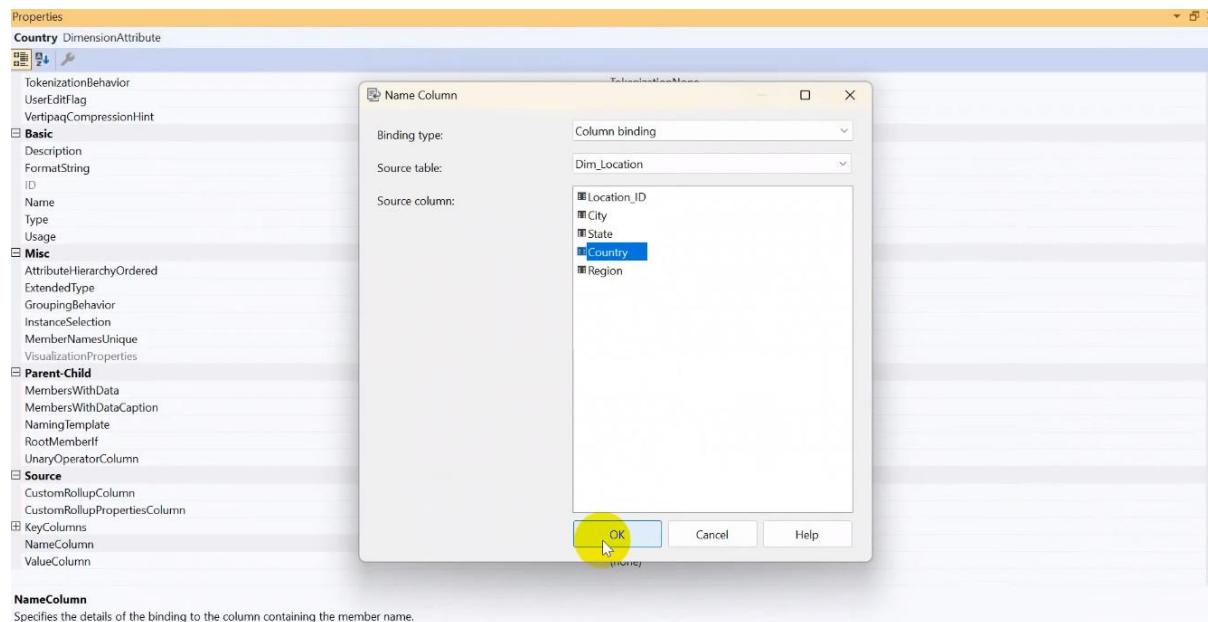


Tại cửa sổ **Properties** ta thêm các những thuộc tính cấp cao hơn State vào **KeyColumns**, sau đó chọn **OK**

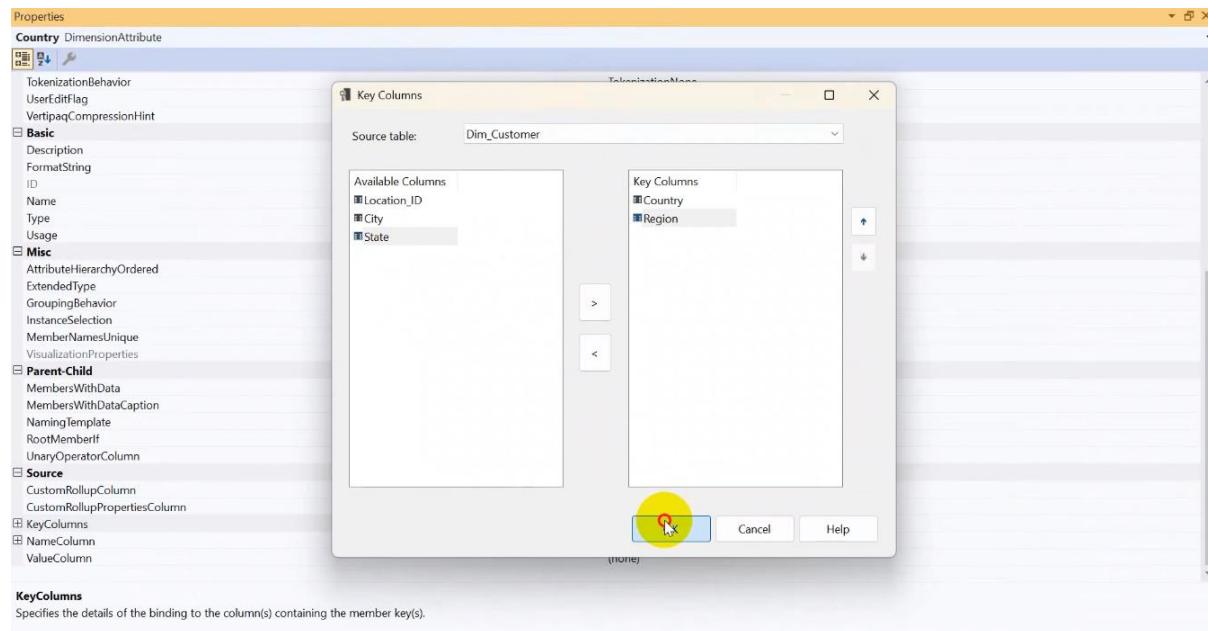


**Bước 5: Chính KeyColumns và Name Column của thuộc tính Country. Cấu hình KeyColumns bao gồm các cột từ các cấp cao hơn như Region, và thiết lập NameColumn cho thuộc tính Country.**

Tại cửa sổ Properties của thuộc tính Country, ta chọn Name Column là Country và nhấn OK

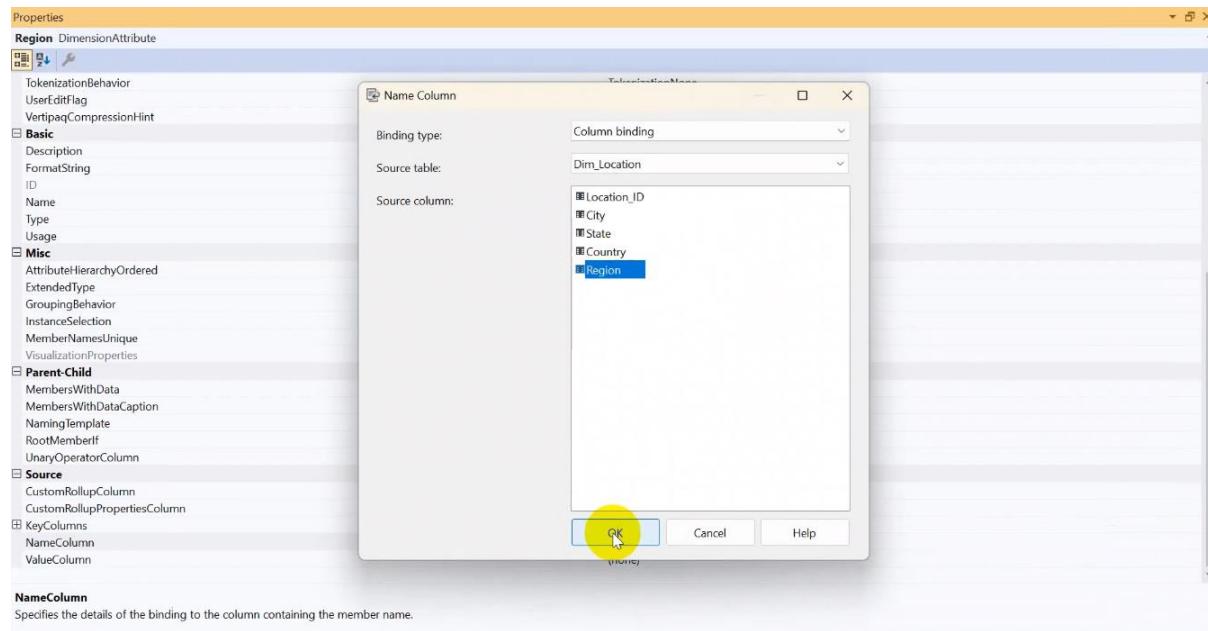


Tại cửa sổ Properties ta thêm các những thuộc tính cấp cao hơn Country vào KeyColumns, sau đó chọn OK



**Bước 6: Chỉnh Name Column** của thuộc tính **Region**. Bởi vì **Region** là thuộc tính cấp cao nhất nên không cần thiết lập **KeyColumns**

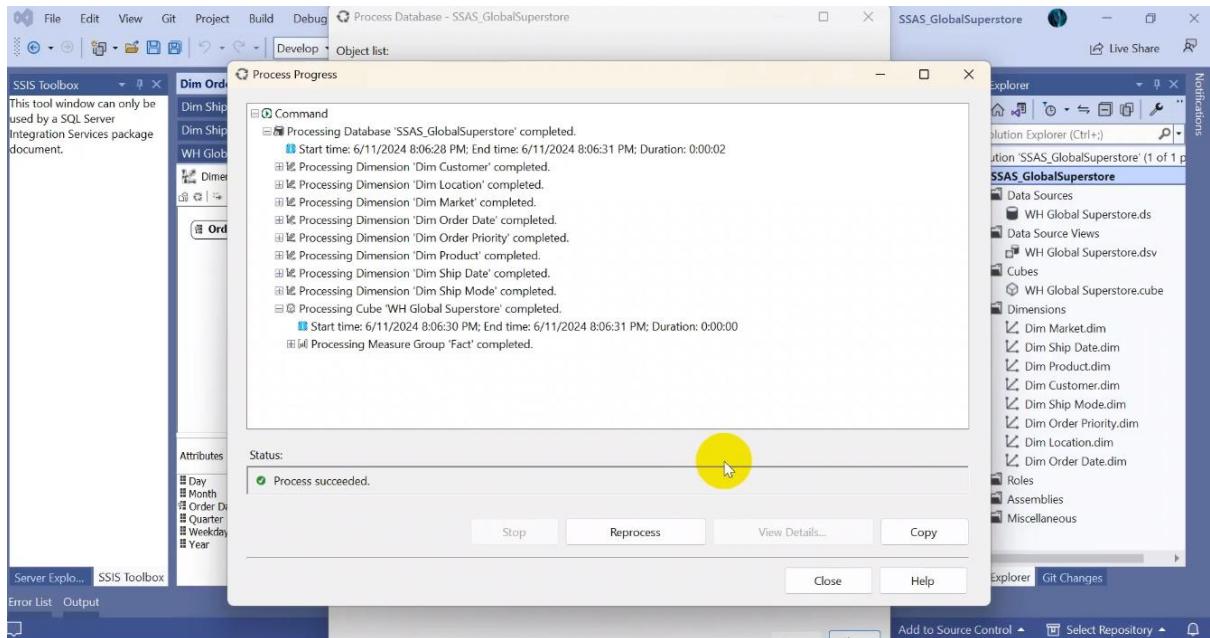
Tại cửa sổ **Properties** của thuộc tính **Region**, ta chọn **Name Column** là **Region** và nhấn **OK**



#### 4.1.4 Chạy project SSAS

Sau khi cấu hình xong tất cả các bước ta nhấn “Start” để tiến hành chạy khởi động project **SSAS\_GlobalSuperstore**

Kết quả chạy project thành công.



## 4.2 Thực thi 15 câu truy vấn trên Visual Studio (Thao tác bằng tay)

### 4.2.1 Roll Up Queries

Câu 1: Thống kê tổng số lượng sản phẩm bán ra theo từng tháng, năm.

Month	Year	Quantity
1	2020	2401
1	2021	2950
1	2018	1583
1	2019	1874
10	2019	3557
10	2020	4024
10	2018	2981
10	2021	5494
11	2020	5444
11	2021	8277
11	2018	3924
11	2019	5168
12	2021	6858
12	2018	4326
12	2019	4549
12	2017	20

Hình 4.4: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 1

Câu 2: Tính tổng doanh thu theo từng danh mục con, danh mục.

Sub Category	Category	Sales
Accessories	Technology	749237.018398404
Appliances	Office Supplies	1011064.30433428
Art	Office Supplies	372091.966175199
Binders	Office Supplies	461911.504749358
Bookcases	Furniture	1466572.23922729
Chairs	Furniture	1501681.76200581
Copiers	Technology	1509436.26990509
Envelopes	Office Supplies	170904.301549435
Fasteners	Office Supplies	83242.3158375025
Furnishings	Furniture	385578.255224824
Labels	Office Supplies	73404.0299957991
Machines	Technology	779060.066156387
Paper	Office Supplies	244291.719033718
Phones	Technology	1706824.13945794
Storage	Office Supplies	1127085.86116481
Supplies	Office Supplies	243074.219475508

Hình 4.5: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 2

### Câu 3: Tổng chi phí vận chuyển theo từng quý, từng năm của ngày vận chuyển.

Quarter	Year	Shipping Cost
1	2020	61640.9988196082
1	2021	72333.4986132048
1	2018	32783.8064686544
1	2019	42169.1519335187
1	2022	2143.5819940269
2	2018	52280.3669776376
2	2021	99178.3720165491
2	2019	66887.429197287
2	2020	88814.6448328029
3	2019	78685.271900123
3	2020	99417.2425013427
3	2018	63049.9289330952
3	2021	129752.683192471
4	2020	115183.133105343
4	2021	158895.061152527
4	2018	94086.8449031059

Hình 4.6: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 3

#### 4.2.2 Drill Down Queries

### Câu 4: Thống kê tổng lợi nhuận từng năm, từng quý.

The screenshot shows the SSAS Design Studio interface with the title bar "WH Global Superstore [Design]". The menu bar includes "Cube Structure", "Dimension Usage", "Calculations", "KPIs", "Actions", "Partitions", "Aggregations", "Perspectives", "Translations", and "Browser". The toolbar includes "Edit as Text", "Import...", "MDX", and various icons for saving, opening, and navigating. The left pane shows the "Metadata" view for the "WH Global Superstore" cube, with a tree structure of Measure Groups (e.g., Measures, Fact, KPIs), Dimensions (e.g., Dim Customer, Dim Location, Dim Market, Dim Order Date, Day, Month, Order Data), and Calculated Members. The right pane displays a query results grid with columns "Year", "Quarter", and "Profit". The data shows quarterly profits from 2017 to 2021.

Year	Quarter	Profit
2017	4	201.992998123169
2018	1	35532.4959853953
2018	2	48795.7924011853
2018	3	65538.1269418891
2018	4	98872.4029092286
2019	1	43424.2872604989
2019	2	82260.9088358823
2019	3	88533.7385690878
2019	4	94542.8675745551
2020	1	75872.1050965423
2020	2	93333.5748184575
2020	3	98092.3907546946
2020	4	138691.752374099
2021	1	86465.6992090177
2021	2	102245.112087431
2021	3	151997.124027718

Hình 4.7: Kết quả truy vấn thông tin tài chính

### Câu 5: Thống kê tổng chiết khấu theo từng khu vực, từng quốc gia

The screenshot shows the SSAS Design Studio interface with the title bar "WH Global Superstore [Design]". The menu bar includes "Cube Structure", "Dimension Usage", "Calculations", "KPIs", "Actions", "Partitions", "Aggregations", "Perspectives", "Translations", and "Browser". The toolbar includes "Edit as Text", "Import...", "MDX", and various icons for saving, opening, and navigating. The left pane shows the "Metadata" view for the "WH Global Superstore" cube, with a tree structure of Measure Groups (e.g., Measures, Fact, KPIs), Dimensions (e.g., Dim Customer, Dim Location, Dim Market, Dim Order Date, Day, Month, Order Data), and Calculated Members. The right pane displays a query results grid with columns "Region", "Country", and "Discount". The data shows discount rates for various countries within the Africa region.

Region	Country	Discount
Africa	Algeria	0
Africa	Angola	0
Africa	Benin	0
Africa	Burundi	0
Africa	Cameroon	0
Africa	Central African Republic	0
Africa	Chad	0
Africa	Cote d'Ivoire	0
Africa	Democratic Republic of the Congo	0
Africa	Djibouti	0
Africa	Egypt	0
Africa	Equatorial Guinea	0
Africa	Eritrea	0
Africa	Ethiopia	0
Africa	Gabon	0
Africa	Ghana	0

Hình 4.8: Kết quả truy vấn thông tin tài chính

### Câu 6: Thống kê tổng lợi nhuận theo từng danh mục sản phẩm, danh mục con sản phẩm.

The screenshot shows the SSAS Design Studio interface with the following details:

- Toolbar:** Cube Structure, Dimension Usage, Calculations, KPIs, Actions, Partitions, Aggregations, Perspectives, Translations, Browser.
- Language:** Default.
- Query Editor:** MDX, showing the query: `SELECT Category, Sub Category, Profit FROM [WH Global Superstore].[Measures].[Fact] WHERE Category = 'Furniture'`.
- Left pane:** Metadata browser showing the cube structure. It includes a tree view of measures (Fact, Discount, Fact Count, Profit, Quantity, Sales, Shipping Cost), KPIs, dimensions (Dim Customer, Dim Location, Dim Market, Dim Order Date, Dim Order Priority, Dim Product, Category), and calculated members.
- Right pane:** A table showing the results of the query. The table has columns: Category, Sub Category, and Profit. The data is as follows:

Category	Sub Category	Profit
Furniture	Bookcases	161924.419882886
Furniture	Chairs	140396.267090578
Furniture	Furnishings	46967.4253476417
Furniture	Tables	-64083.3889139369
Office Supplies	Appliances	141680.589207367
Office Supplies	Art	57953.9107327349
Office Supplies	Binders	72449.8461364489
Office Supplies	Envelopes	29601.116192224
Office Supplies	Fasteners	11525.4241089309
Office Supplies	Labels	15010.5120192845
Office Supplies	Paper	59207.6824752083
Office Supplies	Storage	108461.4895204
Office Supplies	Supplies	22583.2631223099
Technology	Accessories	129626.306129094
Technology	Copiers	258567.54822829
Technology	Machines	58867.8726928122

Hình 4.9: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 6

#### 4.2.3 Slice and Dice Queries

Câu 7: Trong năm 2020 thống kê 3 quốc gia có tổng số lượng sản phẩm bán ra nhiều nhất.

The screenshot shows the SSAS Design Studio interface with the following details:

- Toolbar:** Cube Structure, Dimension Usage, Calculations, KPIs, Actions, Partitions, Aggregations, Perspectives, Translations, Browser.
- Language:** Default.
- Query Editor:** MDX, showing the query: `SELECT Country, Quantity FROM [WH Global Superstore].[Measures].[Fact] WHERE Dim Order Date = { 2020 }`.
- Left pane:** Metadata browser showing the cube structure. It includes a tree view of measures (Fact, Discount, Fact Count, Profit, Quantity, Sales, Shipping Cost), KPIs, dimensions (Dim Customer, Dim Location, Dim Market, Dim Order Date, Dim Order Priority, Dim Product, Category), and calculated members.
- Right pane:** A table showing the results of the query. The table has columns: Country and Quantity. The data is as follows:

Country	Quantity
Australia	3124
France	2874
United States	3036

Hình 4.10: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 7

Câu 8: Thống kê các sản phẩm (id, tên sản phẩm) có tổng doanh thu lớn hơn 75000.

Product ID	Product Name	Sales
7895	Apple Smart Phone, Full Size	86935.7790527344
8553	Cisco Smart Phone, Full Size	76441.5316314697

Hình 4.11: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 8

**Câu 9: Thống kê tổng chi phí vận chuyển theo mức độ ưu tiên của đơn hàng thuộc khu vực Africa trong năm 2021 (theo ngày ship hàng).**

Order Priority	Shipping Cost
Critical	4785.68999290466
High	10607.8000076786
Low	1123.94000607729
Medium	13467.6500359252

Hình 4.12: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 9

**Câu 10: Thống kê Top 3 tên sản phẩm có số lượng được bán ra nhiều nhất.**

WH Global Superstore

Metadata

Measure Group: <All>

WH Global Superstore

Measures

- Fact
  - Discount
  - Fact Count
  - Profit
  - Quantity
  - Sales
  - Shipping Cost

KPIs

Dim Customer

Dim Location

Dim Market

Calculated Members

Dimension Hierarchy Operator Filter Expression Parameters

Dim Product Product Name Custom TopCount([Dim Product].[Product ...] Parameters

Product Name Quantity

Cardinal Index Tab, Clear	337
Eldon File Cart, Single Width	321
Rogers File Cart, Single Width	262

Hình 4.13: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 10

**Câu 11: Thống kê tổng số lượng sản phẩm được bán ra thuộc danh mục “Technology” cho các khách hàng “Corporate” theo loại thị trường (Market).**

WH Global Superstore

Metadata

Measure Group: <All>

WH Global Superstore

Measures

- Profit
- Quantity
- Sales
- Shipping Cost

KPIs

Dim Customer

Dim Location

Dim Market

Market

Dim Order Date

Dim Order Priority

Dim Product

Calculated Members

Dimension Hierarchy Operator Filter Expression Parameters

Dim Product Category Equal { Technology } Parameters

Dim Customer Segment Equal { Corporate } Parameters

Market Quantity

APAC	2580
EMEA	3665
LATAM	2218
USCA	2131

Hình 4.14: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 11

**Câu 12: Thống kê top 2 khách hàng mua hàng mang lại tổng lợi nhuận cao nhất trong quý 4 năm 2020.**

The screenshot shows the SSAS Management Studio interface with the title bar "WH Global Superstore [Design]". The main area displays a query results grid. The grid has columns: "Customer Name" and "Profit". Two rows are visible: "Adrian Barton" with a profit of "5025.65011674166" and "Tamara Chand" with a profit of "8451.14387667924". To the left of the grid is a tree view of the cube structure, showing dimensions like Dim Customer, Dim Location, and Dim Order Date. A toolbar with various icons is at the top, and a menu bar with "Edit as Text", "Import...", and "MDX" is visible.

#### 4.2.4 Pivot

Câu 13: Thống kê tổng lợi nhuận theo quý và khu vực.

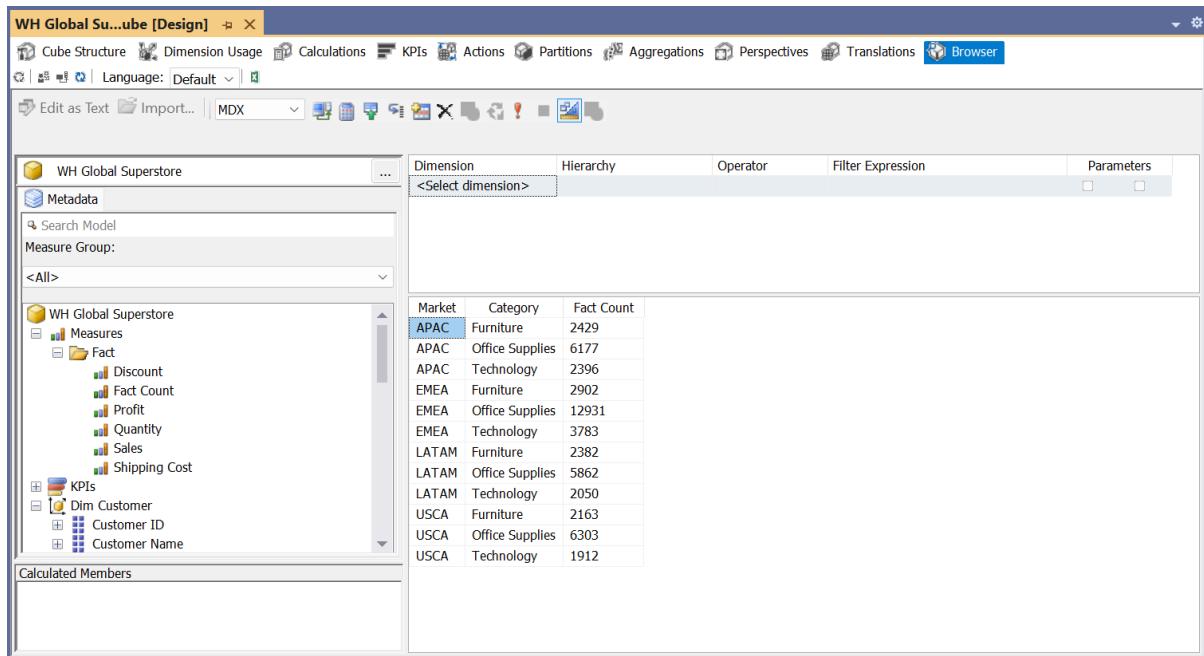
The screenshot shows the SSAS Management Studio interface with the title bar "WH Global Superstore [Design]". The main area displays a query results grid. The grid has columns: "Quarter", "Region", and "Profit". The data is as follows:

Quarter	Region	Profit
1	Africa	7632.48298739642
1	Canada	697.88996990561
1	Carib...	506.887263421901
1	Central	10458.4794477532
1	Centr...	6416.40600349009
1	East	4286.43668067455
1	EMEA	3453.17400631309
1	North	14517.651743114
1	North...	6456.91200441122
1	Oceania	7809.32700830698
1	South	11745.0979475081
1	South...	-364.378895883448
1	West	2255.73890304565
1	Africa	4332.15901833773
1	Canada	1917.06001621485
1	Carib...	1786.75464296341

The left pane shows the cube structure with dimensions like Dim Customer, Dim Location, and Dim Market. A toolbar and menu bar are visible at the top.

Hình 4.15: Kết quả truy vấn thao tác bằng tay trên khối cube – câu 13

Câu 14: Thống kê tổng số đơn hàng được bán ra theo từng loại thị trường và danh mục sản phẩm.

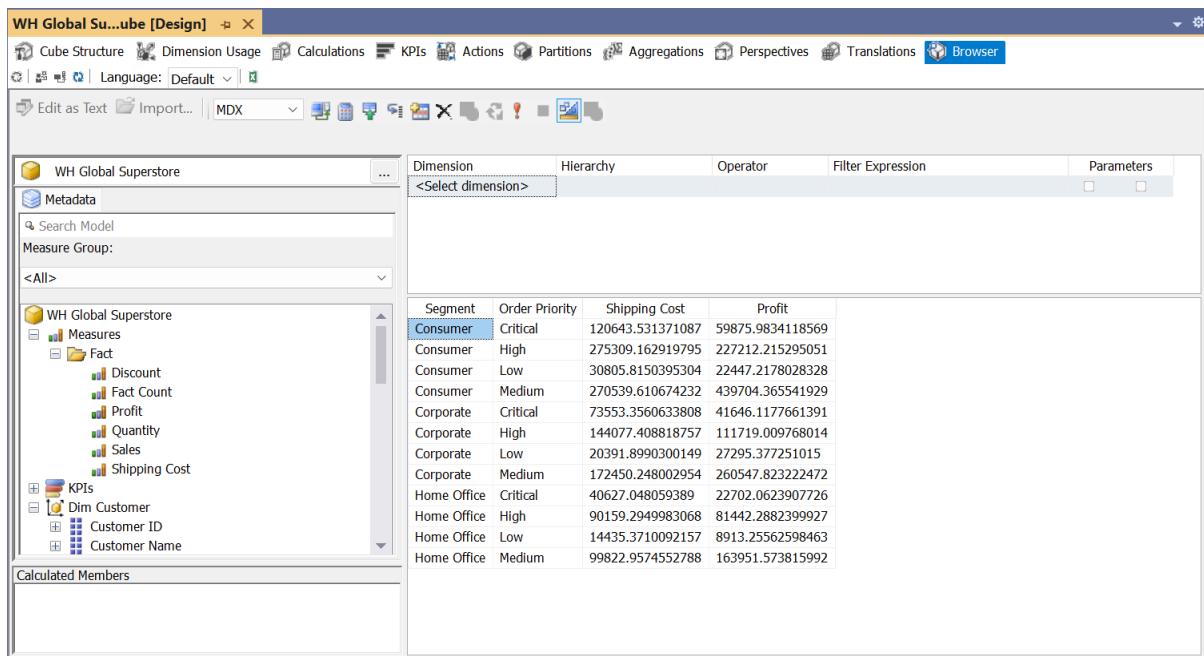


The screenshot shows the SSAS Management Studio interface in 'Design' mode. The left pane displays the cube structure with nodes like 'WH Global Superstore', 'Metadata', and 'Measure Group'. The right pane shows a query results grid with the following data:

Market	Category	Fact Count
APAC	Furniture	2429
APAC	Office Supplies	6177
APAC	Technology	2396
EMEA	Furniture	2902
EMEA	Office Supplies	12931
EMEA	Technology	3783
LATAM	Furniture	2382
LATAM	Office Supplies	5862
LATAM	Technology	2050
USCA	Furniture	2163
USCA	Office Supplies	6303
USCA	Technology	1912

Hình 4.16: Kết quả truy vấn thông tin chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng

### Câu 15: Thống kê tổng chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng



The screenshot shows the SSAS Management Studio interface in 'Design' mode. The left pane displays the cube structure with nodes like 'WH Global Superstore', 'Metadata', and 'Measure Group'. The right pane shows a query results grid with the following data:

Segment	Order Priority	Shipping Cost	Profit
Consumer	Critical	120643.531371087	59875.9834118569
Consumer	High	275309.162919795	227212.215295051
Consumer	Low	30805.8150395304	22447.2178028328
Consumer	Medium	270539.610674232	439704.365541929
Corporate	Critical	73553.3560633808	41646.1177661391
Corporate	High	144077.408818757	111719.009768014
Corporate	Low	20391.8990300149	27295.377251015
Corporate	Medium	172450.248002954	260547.823222472
Home Office	Critical	40627.048059389	22702.0623907726
Home Office	High	90159.2949983068	81442.2882399927
Home Office	Low	14435.3710092157	8913.25562598463
Home Office	Medium	99822.9574552788	163951.573815992

Hình 4.17: Kết quả truy vấn thông tin chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng

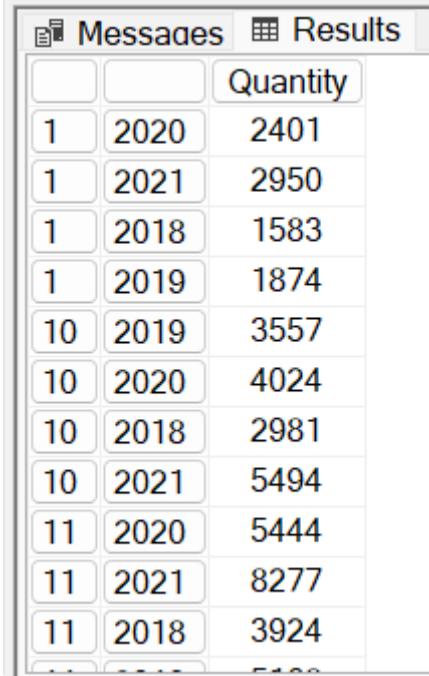
### 4.3 Thực thi 15 câu truy vấn bằng ngôn ngữ MDX

#### 4.3.1 Roll Up Queries

##### Câu 1: Thống kê tổng số lượng sản phẩm bán ra theo từng tháng, năm.

```
SELECT NON EMPTY [Measures].[Quantity] ON COLUMNS,
```

```
NON EMPTY CrossJoin(
    [Dim Order Date].[Month].children,
    [Dim Order Date].[Year].children
) ON ROWS
FROM [WH Global Superstore];
```



		Quantity
1	2020	2401
1	2021	2950
1	2018	1583
1	2019	1874
10	2019	3557
10	2020	4024
10	2018	2981
10	2021	5494
11	2020	5444
11	2021	8277
11	2018	3924

Hình 4.18: MDX Query – câu 1

## Câu 2: Tính tổng doanh thu theo từng danh mục con, danh mục.

```
SELECT NON EMPTY [Measures].[Sales] ON COLUMNS,
NON EMPTY CrossJoin(
    [Dim Product].[Sub Category].children,
    [Dim Product].[Category].children
) ON ROWS
FROM [WH Global Superstore];
```

		Sales
Accessories	Technology	749237.018398404
Appliances	Office Supplies	1011064.30433428
Art	Office Supplies	372091.966175199
Binders	Office Supplies	461911.504749358
Bookcases	Furniture	1466572.23922729
Chairs	Furniture	1501681.76200581
Copiers	Technology	1509436.26990509
Envelopes	Office Supplies	170904.301549435
Fasteners	Office Supplies	83242.3158375025
Furnishings	Furniture	385578.255224824
Labels	Office Supplies	73404.0299957991

Hình 4.19: MDX Query – câu 2

**Câu 3: Tổng chi phí vận chuyển theo từng quý và từng năm của ngày vận chuyển.**

```
SELECT NON EMPTY [Measures].[Shipping Cost] ON COLUMNS,
NON EMPTY CrossJoin(
    [Dim Ship Date].[Quarter].Members,
    [Dim Ship Date].[Year].children
) ON ROWS
FROM [WH Global Superstore];
```

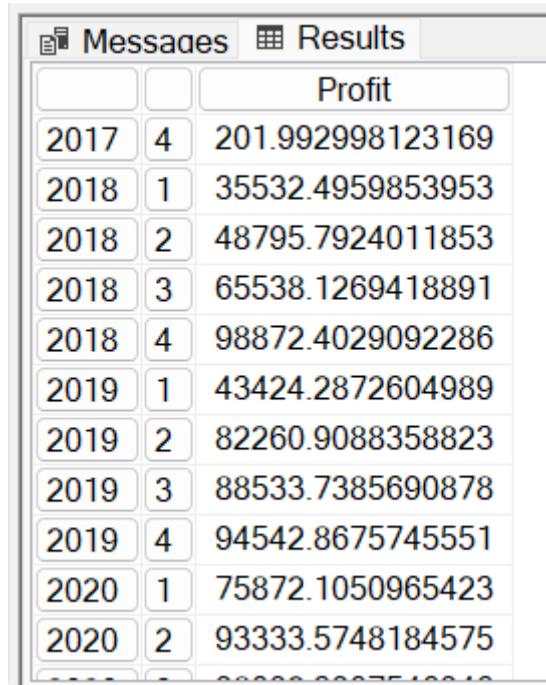
		Shipping Cost
1	2020	61640.9988196082
1	2021	72333.4986132048
1	2018	32783.8064686544
1	2019	42169.1519335187
1	2022	2143.5819940269
2	2018	52280.3669776376
2	2021	99178.3720165491
2	2019	66887.429197287
2	2020	88814.6448328029
3	2019	78685.271900123
3	2020	99417.2425013427

Hình 4.20: MDX Query – Câu 3

### 4.3.2 Drill Down Queries

**Câu 4: Thống kê tổng lợi nhuận từng năm, từng quý.**

```
Select NON EMPTY [Measures].[Profit] on COLUMNS,  
NON EMPTY [Dim Order Date].[Year].children * [Dim Order Date].[Quarter].children on  
ROWS  
FROM [WH Global Superstore];
```



		Profit
2017	4	201.992998123169
2018	1	35532.4959853953
2018	2	48795.7924011853
2018	3	65538.1269418891
2018	4	98872.4029092286
2019	1	43424.2872604989
2019	2	82260.9088358823
2019	3	88533.7385690878
2019	4	94542.8675745551
2020	1	75872.1050965423
2020	2	93333.5748184575

Hình 4.21: MDX Query – Câu 4

**Câu 5: Thống kê tổng chiết khấu theo từng khu vực, từng quốc gia trong khu vực đó.**

```
Select NON EMPTY [Measures].[Discount] on COLUMNS,  
NON EMPTY [Dim Location].[Region].children * [Dim Location].[Country].children on  
ROWS  
FROM [WH Global Superstore];
```

		Discount
Africa	Algeria	0
Africa	Angola	0
Africa	Benin	0
Africa	Burundi	0
Africa	Cameroon	0
Africa	Central African Republic	0
Africa	Chad	0
Africa	Cote d'Ivoire	0
Africa	Democratic Republic of the Congo	0
Africa	Djibouti	0
Africa	Egypt	0
...	...	...

Hình 4.22: MDX Query – câu 5

**Câu 6: Thống kê tổng lợi nhuận theo từng danh mục sản phẩm, danh mục con sản phẩm.**

```
Select NON EMPTY [Measures].[Profit] on COLUMNS,
NON EMPTY [Dim Product].[Category].children * [Dim Product].[Sub Category].children
on ROWS
FROM [WH Global Superstore];
```

		Profit
Furniture	Bookcases	161924.419882886
Furniture	Chairs	140396.267090578
Furniture	Furnishings	46967.4253476417
Furniture	Tables	-64083.3889139369
Office Supplies	Appliances	141680.589207367
Office Supplies	Art	57953.9107327349
Office Supplies	Binders	72449.8461364489
Office Supplies	Envelopes	29601.116192224
Office Supplies	Fasteners	11525.4241089309
Office Supplies	Labels	15010.5120192845
Office Supplies	Paper	59207.6824752083

Hình 4.23: MDX Query – câu 6

### 4.3.3 Slice and Dice Queries

**Câu 7: Trong năm 2020 thống kê 3 quốc gia có tổng số lượng sản phẩm bán ra nhiều nhất.**

```

SELECT NON EMPTY [Measures].[Quantity] ON COLUMNS,
NON EMPTY TopCount(
    [Dim Location].[Country].children,
    3,
    [Measures].[Quantity]
) ON ROWS
FROM [WH Global Superstore]
WHERE ([Dim Order Date].[Year].& [2020]);

```

Messages		Results
		Quantity
	Australia	3124
	United States	3036
	France	2874

Hình 4.24: MDX Query – câu 7

**Câu 8: Thống kê các sản phẩm (id, tên sản phẩm) có tổng doanh thu lớn hơn 75000.**

```

SELECT NON EMPTY [Measures].[Sales] ON COLUMNS,
NON EMPTY Filter(
    CrossJoin(
        [Dim Product].[Product ID].children,
        [Dim Product].[Product Name].children
    ),
    [Measures].[Sales] > 75000
) ON ROWS
FROM [WH Global Superstore];

```

Messages		Results
		Sales
7895	Apple Smart Phone, Full Size	86935.7790527344
8553	Cisco Smart Phone, Full Size	76441.5316314697

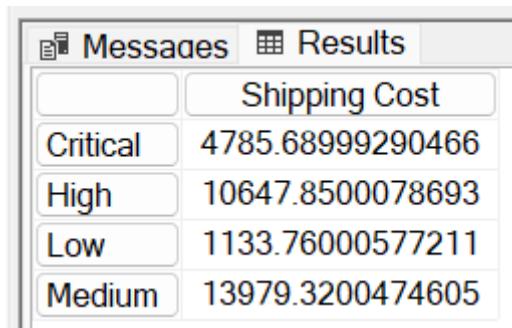
Hình 4.25: MDX Query – câu 8

**Câu 9: Thống kê tổng chi phí vận chuyển theo mức độ ưu tiên của đơn hàng thuộc khu vực Africa trong năm 2021 (theo ngày ship hàng).**

```

SELECT NON EMPTY [Measures].[Shipping Cost] ON COLUMNS,
NON EMPTY [Dim Order Priority].[Order Priority].children ON ROWS
FROM [WH Global Superstore]
WHERE (
    [Dim Ship Date].[Year].& [2021],
    [Dim Location].[Region].& [Africa]
);

```

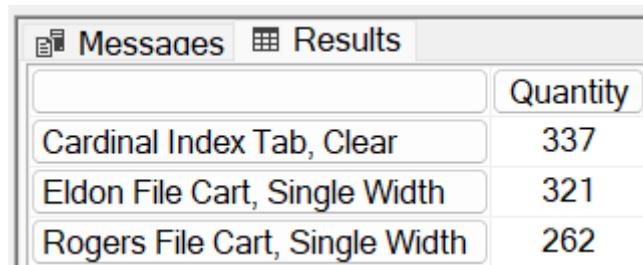


	Shipping Cost
Critical	4785.68999290466
High	10647.8500078693
Low	1133.76000577211
Medium	13979.3200474605

Hình 4.26: MDX Query – câu 9

**Câu 10: Thống kê Top 3 tên sản phẩm có số lượng được bán ra nhiều nhất.**

```
SELECT NON EMPTY [Measures].[Quantity] ON COLUMNS,
NON EMPTY TopCount(
    [Dim Product].[Product Name].children,
    3,
    [Measures].[Quantity]
) ON ROWS
FROM [WH Global Superstore];
```

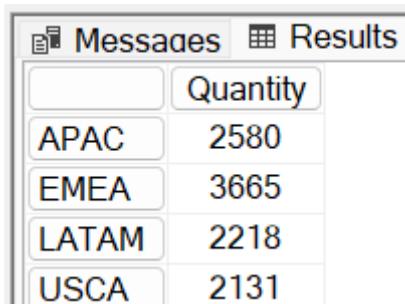


	Quantity
Cardinal Index Tab, Clear	337
Eldon File Cart, Single Width	321
Rogers File Cart, Single Width	262

Hình 4.27: MDX Query – câu 10

**Câu 11: Thống kê tổng số lượng sản phẩm được bán ra thuộc danh mục “Technology” cho các khách hàng “Corporate” theo loại thị trường (Market)**

```
SELECT NON EMPTY [Measures].[Quantity] ON COLUMNS,
NON EMPTY [Dim Market].[Market].children ON ROWS
FROM [WH Global Superstore]
WHERE (
    { [Dim Product].[Category].& [Technology] },
    { [Dim Customer].[Segment].& [Corporate] }
);
```



	Quantity
APAC	2580
EMEA	3665
LATAM	2218
USCA	2131

Hình 4.28: MDX Query – câu 11

**Câu 12: Thống kê top 2 khách hàng mua hàng mang lại tổng lợi nhuận cao nhất trong quý 4 năm 2020**

```
SELECT NON EMPTY [Measures].[Profit] ON COLUMNS,
NON EMPTY TopCount(
    [Dim Customer].[Customer Name].children,
    2,
    [Measures].[Profit]
) ON ROWS
FROM [WH Global Superstore]
WHERE (
    { [Dim Order Date].[Year].& [2020] },
    { [Dim Order Date].[Quarter].[4] }
);

```

Results	
	Profit
Tamara Chand	8451.14387667924
Adrian Barton	5025.65011674166

Hình 4.29: MDX Query – câu 12

#### 4.3.4 Pivot

**Câu 13: Thống kê tổng lợi nhuận theo quý và khu vực**

```
SELECT NON EMPTY [Measures].[Profit] ON COLUMNS,
NON EMPTY (
    [Dim Order Date].[Quarter].children,
    [Dim Location].[Region].children
) ON ROWS
FROM [WH Global Superstore];
```

Results	
	Profit
1 Africa	7632.48298739642
1 Canada	697.889996990561
1 Caribbean	506.887263421901
1 Central	10458.4794477532
1 Central Asia	6416.40600349009
1 East	4286.43668067455
1 EMEA	3453.17400631309
1 North	14517.651743114
1 North Asia	6456.91200441122
1 Oceania	7809.32700830698
1 South	11745.0979475081
1 Southeast Asia	364.378895883448
1 West	2255.73890304565
1 Africa	4332.15901833773
1 Canada	1917.06001621485
1 Caribbean	1786.75464296341
1 Central	20169.9783377424

Hình 4.30: MDX Query – câu 13

**Câu 14: Thống kê tổng số đơn hàng được bán ra theo từng loại thị trường và danh mục sản phẩm.**

```
SELECT NON EMPTY { [Measures].[Fact Count] } ON COLUMNS,
NON EMPTY CrossJoin(
    [Dim Market].[Market].children,
    [Dim Product].[Category].children
) ON ROWS
FROM [WH Global Superstore];
```

		Messages	Results	Fact Count
APAC	Furniture			2429
APAC	Office Supplies			6177
APAC	Technology			2396
EMEA	Furniture			2902
EMEA	Office Supplies			12931
EMEA	Technology			3783
LATAM	Furniture			2382
LATAM	Office Supplies			5862
LATAM	Technology			2050
USCA	Furniture			2163
USCA	Office Supplies			6303
USCA	Technology			1912

Hình 4.31: MDX Query – câu 14

**Câu 15: Thống kê tổng chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng**

```
SELECT NON EMPTY { [Measures].[Shipping Cost],
[Measures].[Profit] } ON COLUMNS,
NON EMPTY CrossJoin(
    [Dim Customer].[Segment].children,
    [Dim Order Priority].[Order Priority].children
) ON ROWS
FROM [WH Global Superstore];
```

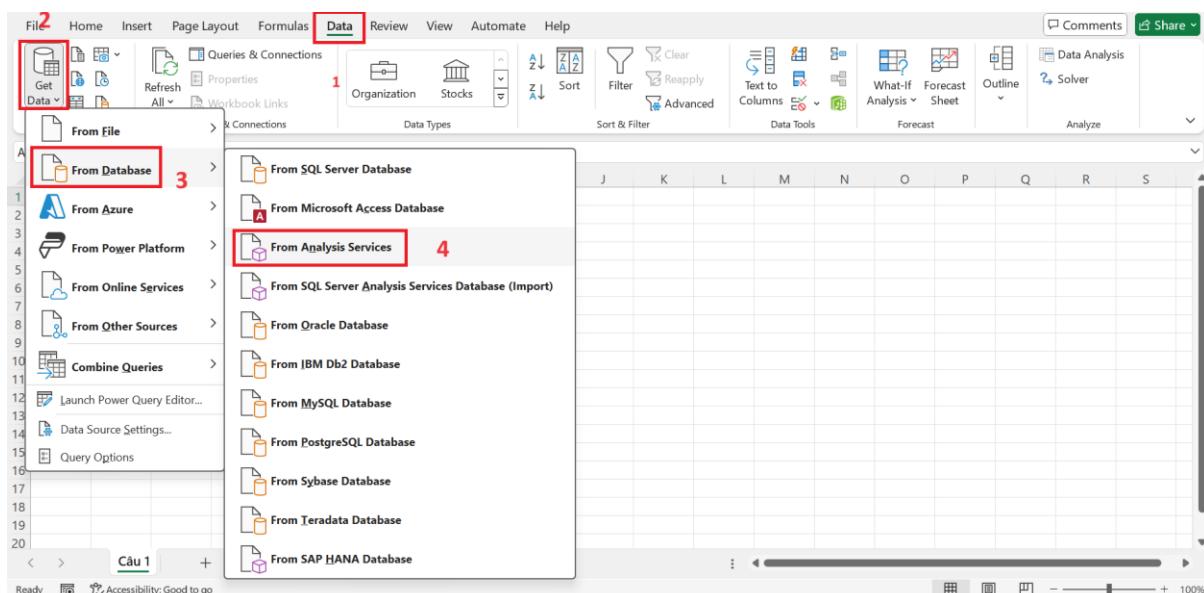
		Shipping Cost	Profit
Consumer	Critical	120643.531371087	59875.9834118569
Consumer	High	275309.162919795	227212.215295051
Consumer	Low	30805.8150395304	22447.2178028328
Consumer	Medium	270539.610674232	439704.365541929
Corporate	Critical	73553.3560633808	41646.1177661391
Corporate	High	144077.408818757	111719.009768014
Corporate	Low	20391.8990300149	27295.377251015
Corporate	Medium	172450.248002954	260547.823222472
Home Office	Critical	40627.048059389	22702.0623907726
Home Office	High	90159.2949983068	81442.2882399927
Home Office	Low	14435.3710092157	8913.25562598463
Home Office	Medium	99822.9574552788	163951.573815992

Hình 4.32: MDX Query – câu 15

## 4.4 Thực thi 15 câu truy vấn bằng Excel

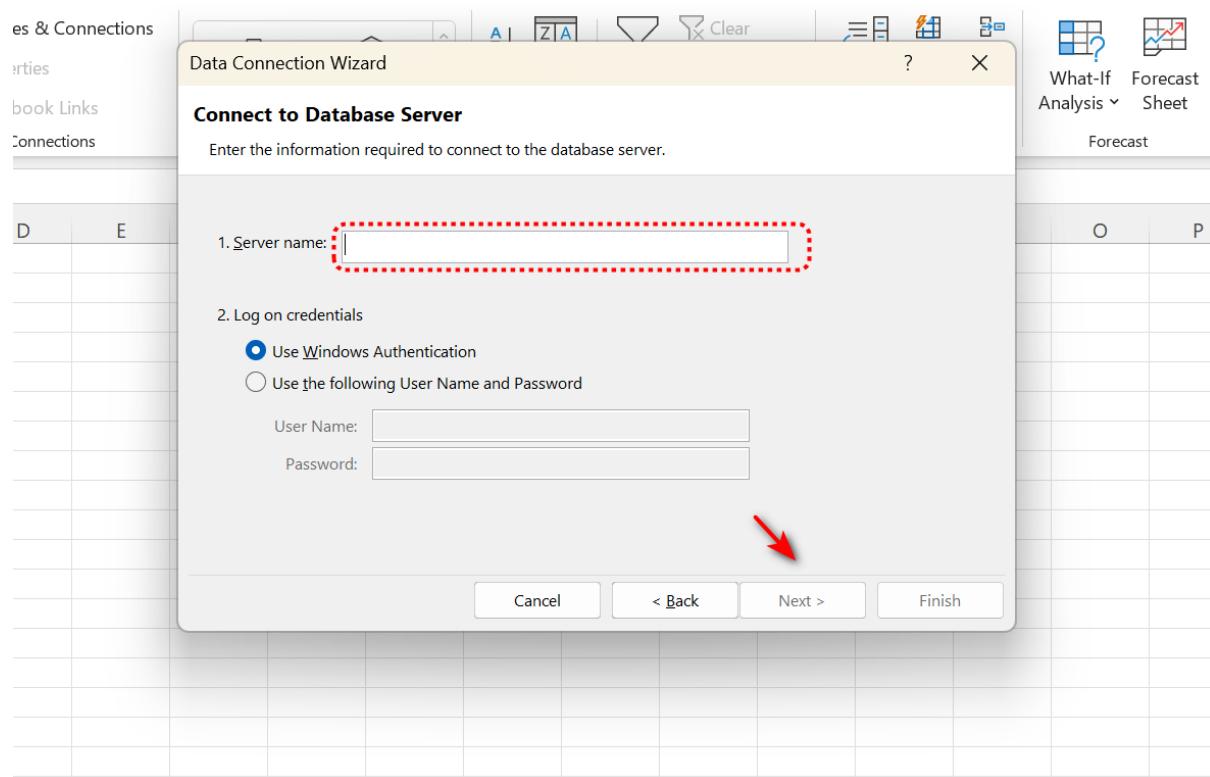
### 4.4.1 Thực hiện kết nối Microsoft Analysis Services

Bước 1: Mở Excel -> Get Data -> From Database -> From Analysis Services.

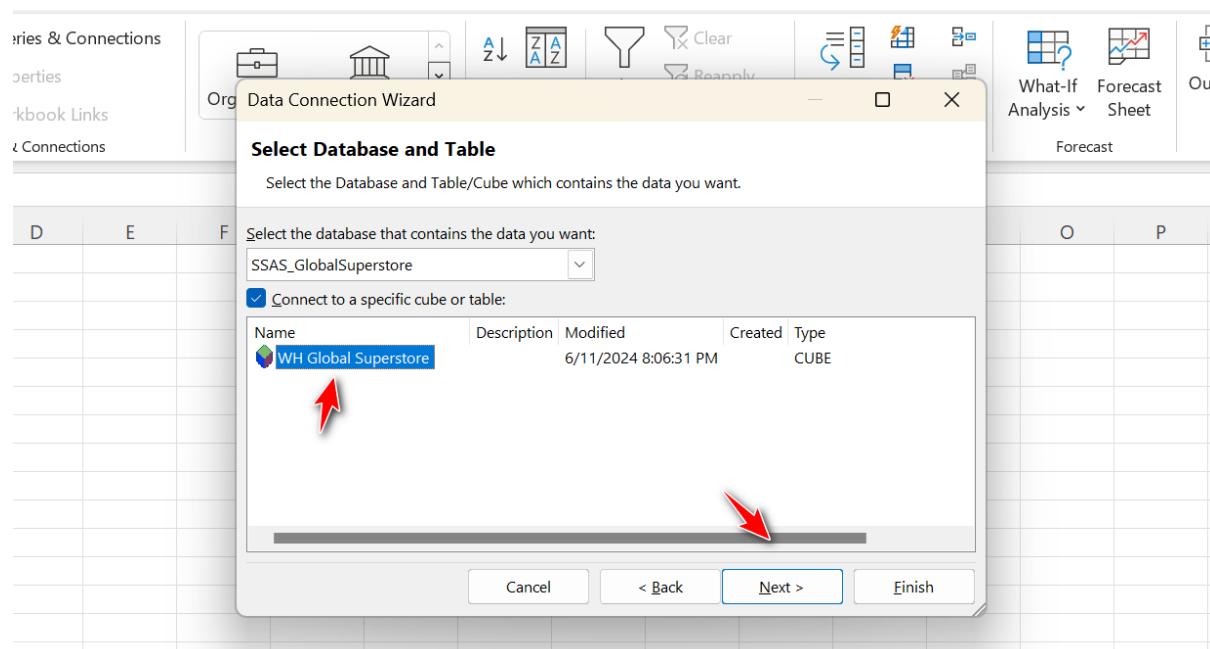


Bước 2: Điền các thông tin của SQL Server Analysis Services database vào:

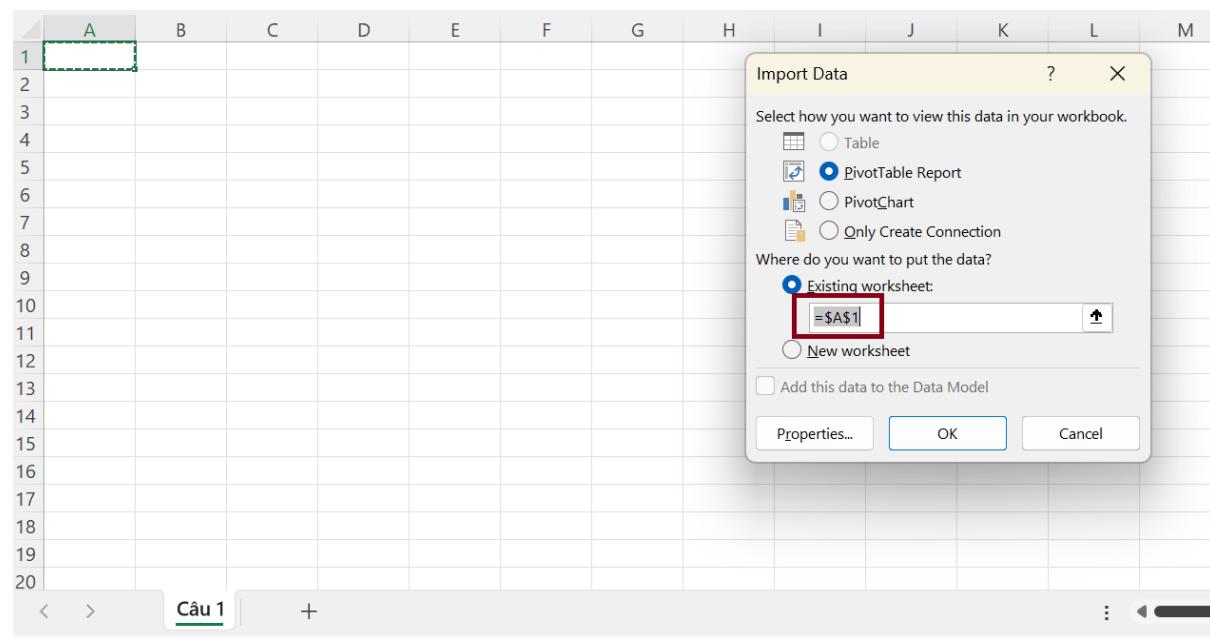
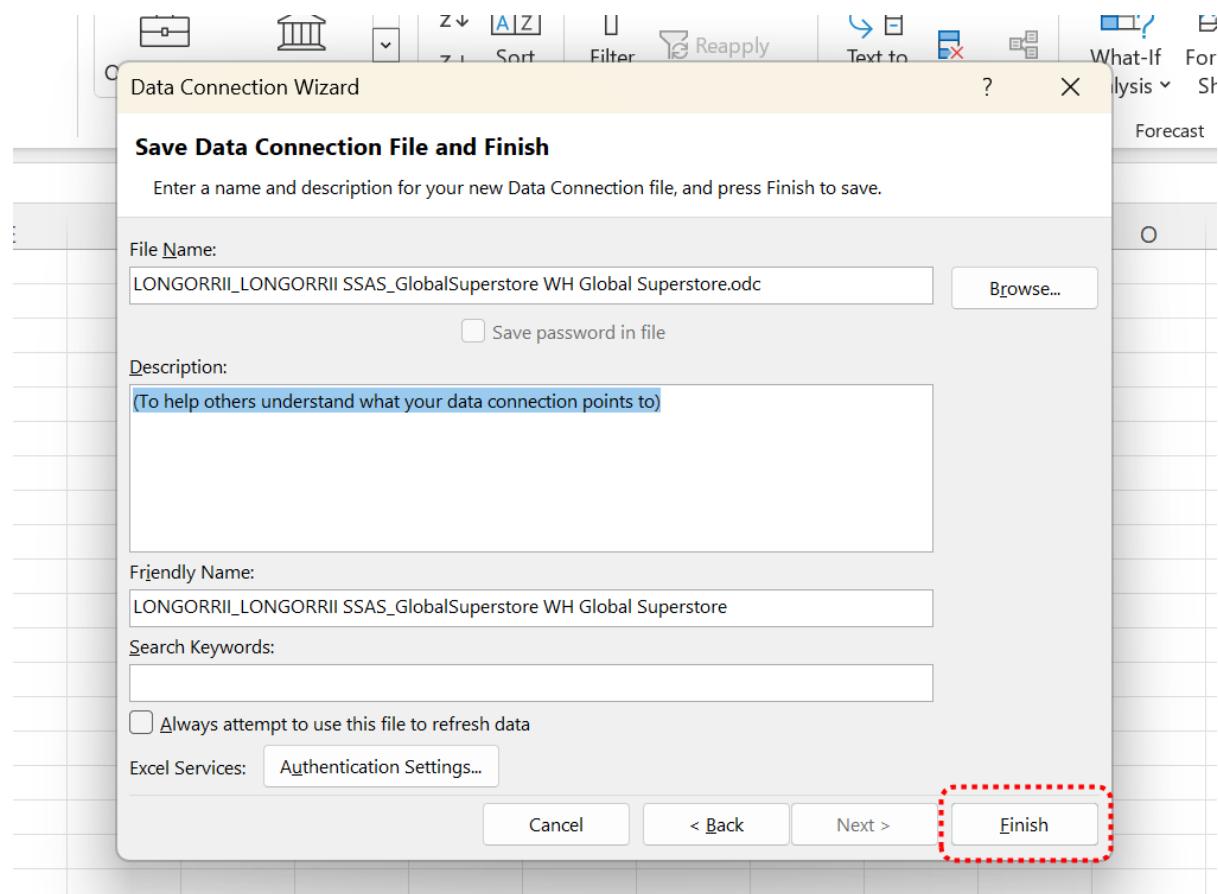
- Tên Server
- Chọn Use Windows Authentication và nhấn Next



**Bước 3:** Chọn database SSAS\_GlobalSuperstore và nhấn “Next”



**Bước 4:** Nhấn “Finish” để tiến hành chọn Cell nhập dữ liệu sau đó nhập “OK” để hoàn tất.



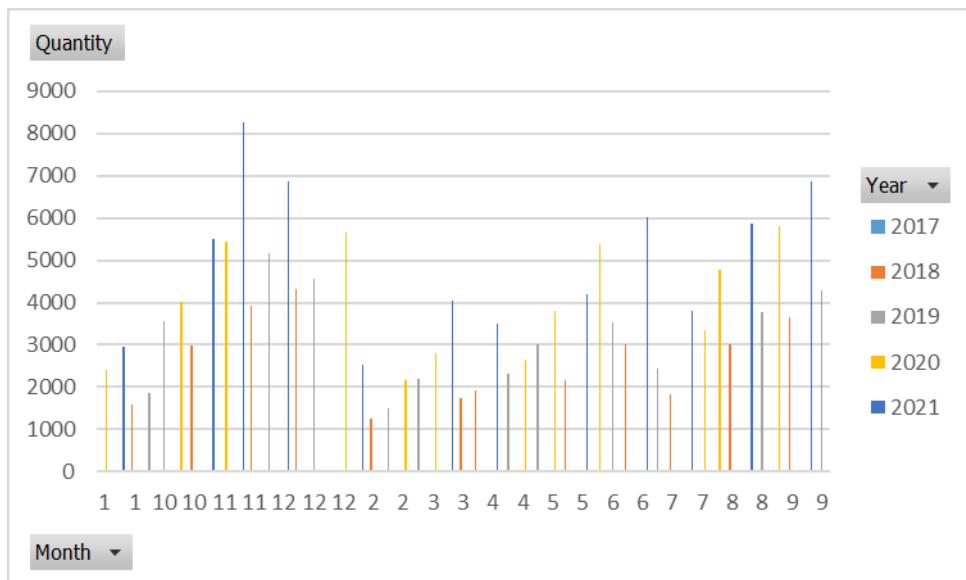
## 4.4.2 Thực hiện các truy vấn

### 4.4.2.1 Roll Up Queries

**Câu 1: Thống kê tổng số lượng sản phẩm bán ra theo từng tháng, năm.**

Câu 1: Thống kê tổng số lượng sản phẩm báu ra theo từng tháng, năm.						
Quantity	Row Labels	Column Labels	2018	2019	2020	2021
						Grand Total
1	1				2401	2401
5	1				2950	2950
6	1		1583			1583
7	1			1874		1874
8	10			3557		3557
9	10				4024	4024
10	10		2981			2981
11	10				5494	5494
12	11			5444		5444
13	11				8277	8277
14	11		3924			3924
15	11			5168		5168
16	12				6858	6858
17	12		4326			4326
18	12			4549		4549
19	12	20				20
20	12			5650		5650
21	2				2524	2524
22	2		1261			1261
23	2			1504		1504
24	2				2175	2175
25	3		2189			2189
26	3				2787	2787

Hình 4.33: Kết quả truy vấn trên Excel – câu 1

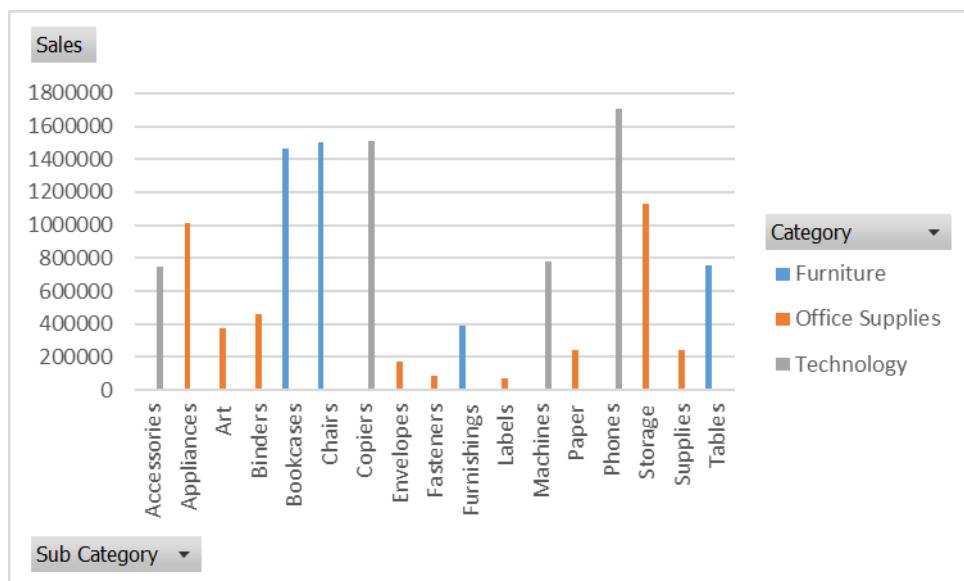


Hình 4.34: Kết quả truy vấn trên Excel – câu 1 – Chart

**Câu 2: Tính tổng doanh thu theo từng danh mục con, danh mục.**

Câu 2: Tính tổng doanh thu theo từng danh mục con, danh mục.				
Sales	Column Labels	Office Supplies	Technology	Grand Total
Row Labels	Furniture			
Accessories		749237.018	749237.018	
Appliances		1011064.304		1011064.3
Art		372091.9662		372091.966
Binders		461911.5047		461911.505
Bookcases	1466572.239			1466572.24
Chairs	1501681.762			1501681.76
Copiers		1509436.27	1509436.27	
Envelopes		170904.3015		170904.302
Fasteners		83242.31584		83242.3158
Furnishings	385578.2552			385578.255
Labels		73404.03		73404.03
Machines		779060.066	779060.066	
Paper		244291.719		244291.719
Phones		1706824.14	1706824.14	
Storage		1127085.861		1127085.86
Supplies		243074.2195		243074.219
Tables	757041.9226			757041.923
<b>Grand Total</b>	<b>4110874.179</b>	<b>3787070.222</b>	<b>4744557.49</b>	<b>12642501.9</b>

Hình 4.35: Kết quả truy vấn trên Excel – câu 2

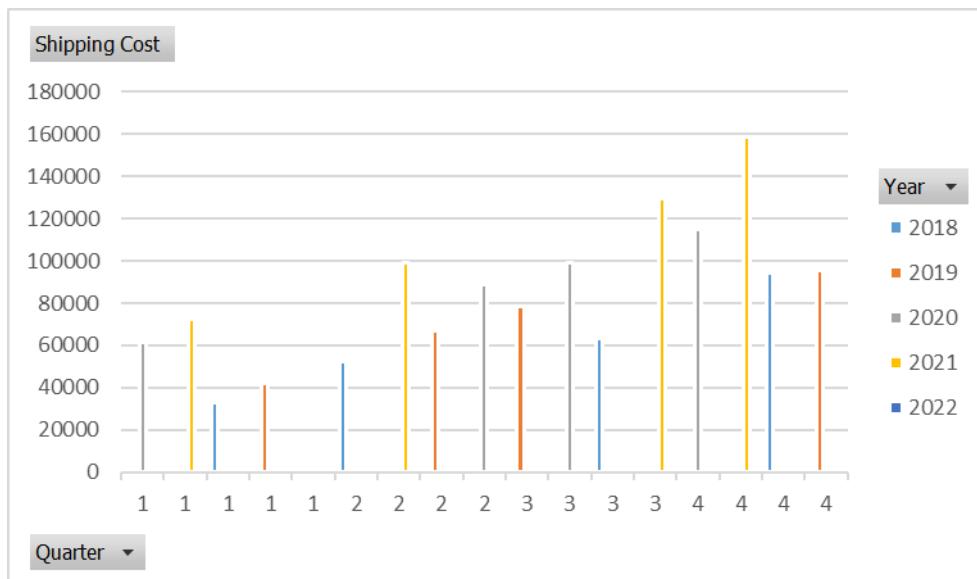


Hình 4.36: Kết quả truy vấn trên Excel – câu 2 – Chart

Câu 3: Tổng chi phí vận chuyển theo từng quý, từng năm của ngày vận chuyển.

Câu 3: Tổng chi phí vận chuyển theo từng quý, từng năm của ngày vận chuyển.						
Row Label	2018	2019	2020	2021	2022	Grand Total
1			61640.9988			61640.9988
1				72333.4986		72333.4986
1	32783.80647					32783.8065
1		42169.1519				42169.1519
1				2143.58199	2143.58199	
2	52280.36698					52280.367
2			99178.372			99178.372
2		66887.4292				66887.4292
2			88814.6448			88814.6448
3		78685.2719				78685.2719
3			99417.2425			99417.2425
3	63049.92893					63049.9289
3			129752.683			129752.683
4		115183.133				115183.133
4				158895.061		158895.061
4	94086.8449					94086.8449
4		95513.6869				95513.6869
Grand Total		242200.9473	283255.54	365056.019	460159.615	2143.58199
						1352815.7

Hình 4.37: Kết quả truy vấn trên Excel – câu 3



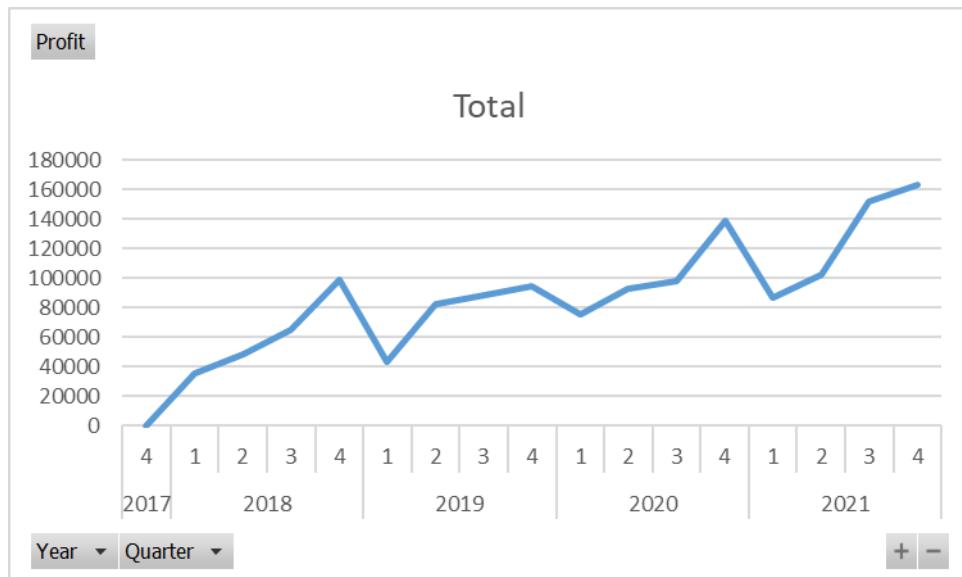
Hình 4.38: Kết quả truy vấn trên Excel – câu 3 – Chart

#### 4.4.2.2 Drill Down Queries

**Câu 4: Thống kê tổng lợi nhuận theo từng năm, từng quý.**

Câu 4: Thống kê tổng lợi nhuận theo từng năm, từng quý	
Row Labels	Profit
2017	
4	201.992998
2018	
1	35532.496
2	48795.7924
3	65538.1269
4	98872.4029
2019	
1	43424.2873
2	82260.9088
3	88533.7386
4	94542.8676
2020	
1	75872.1051
2	93333.5748
3	98092.3908
4	138691.752
2021	
1	86465.6992
2	102245.112
3	151997.124
4	163056.918
Grand Total	1467457.29

Hình 4.39: Kết quả truy vấn trên Excel – câu 4



Hình 4.40: Kết quả truy vấn trên Excel – câu 4 – Chart

Câu 5: Thống kê tổng chiết khấu theo từng khu vực, từng quốc gia.

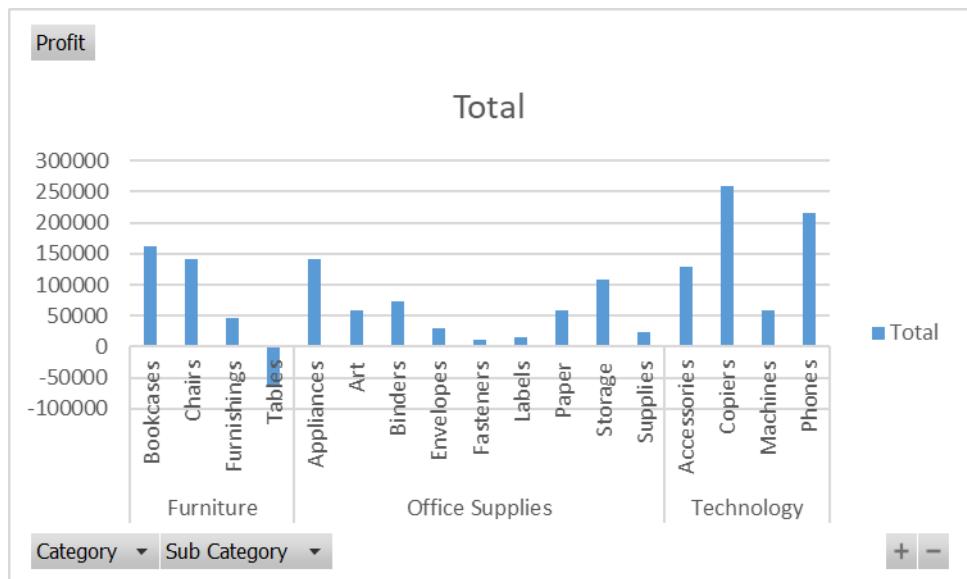
Câu 5: Thống kê tổng chiết khấu theo từng khu vực, từng quốc gia.	
Row Labels	Discount
■ Africa	
Algeria	0
Angola	0
Benin	0
Burundi	0
Cameroon	0
Central African Republic	0
Chad	0
Cote d'Ivoire	0
Democratic Republic of the Congo	0
Djibouti	0
Egypt	0
Equatorial Guinea	0
Eritrea	0
Ethiopia	0
Gabon	0
Ghana	0
Guinea	0
Guinea-Bissau	0
Kenya	0
Lesotho	0
Liberia	0
Libya	0
Madagascar	0
...	...

Hình 4.41: Kết quả truy vấn trên Excel – câu 5.

## Câu 6: Thống kê tổng lợi nhuận theo từng danh mục sản phẩm, danh mục con sản phẩm.

Câu 6: Thống kê tổng lợi nhuận theo từng danh mục sản phẩm, danh mục con sản phẩm.	
Row Labels	Profit
<input type="checkbox"/> Furniture	
Bookcases	161924.42
Chairs	140396.267
Furnishings	46967.4253
Tables	-64083.3889
<input type="checkbox"/> Office Supplies	
Appliances	141680.589
Art	57953.9107
Binders	72449.8461
Envelopes	29601.1162
Fasteners	11525.4241
Labels	15010.512
Paper	59207.6825
Storage	108461.49
Supplies	22583.2631
<input type="checkbox"/> Technology	
Accessories	129626.306
Copiers	258567.548
Machines	58867.8727
Phones	216717.006
<b>Grand Total</b>	<b>1467457.29</b>

Hình 4.42: Kết quả truy vấn trên Excel – câu 6



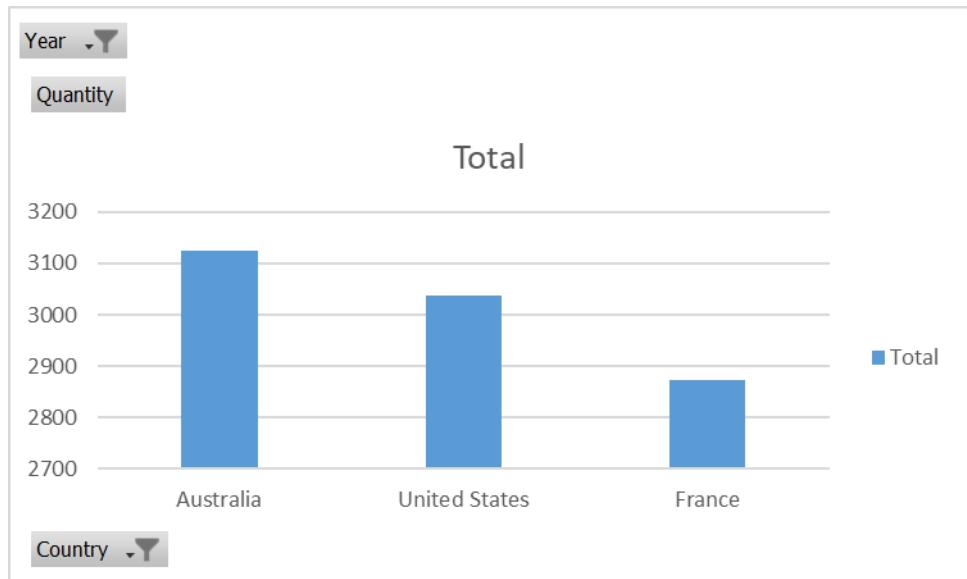
Hình 4.43: Kết quả truy vấn trên Excel – câu 6 – Chart

#### 4.4.2.3 Slice and Dice Queries

Câu 7: Trong năm 2020 thống kê 3 quốc gia có tổng số lượng sản phẩm bán ra nhiều nhất.

Câu 7: Trong năm 2020 thống kê 3 quốc gia có tổng số lượng sản phẩm bán ra nhiều nhất.	
Year	2020
Row Labels	
Australia	3124
United States	3036
France	2874
<b>Grand Total</b>	<b>9034</b>

Hình 4.44: Kết quả truy vấn trên Excel – câu 7



Hình 4.45: Kết quả truy vấn trên Excel – câu 7 – Chart

**Câu 8: Thống kê các sản phẩm (id, tên sản phẩm) có tổng doanh thu lớn hơn 75000.**

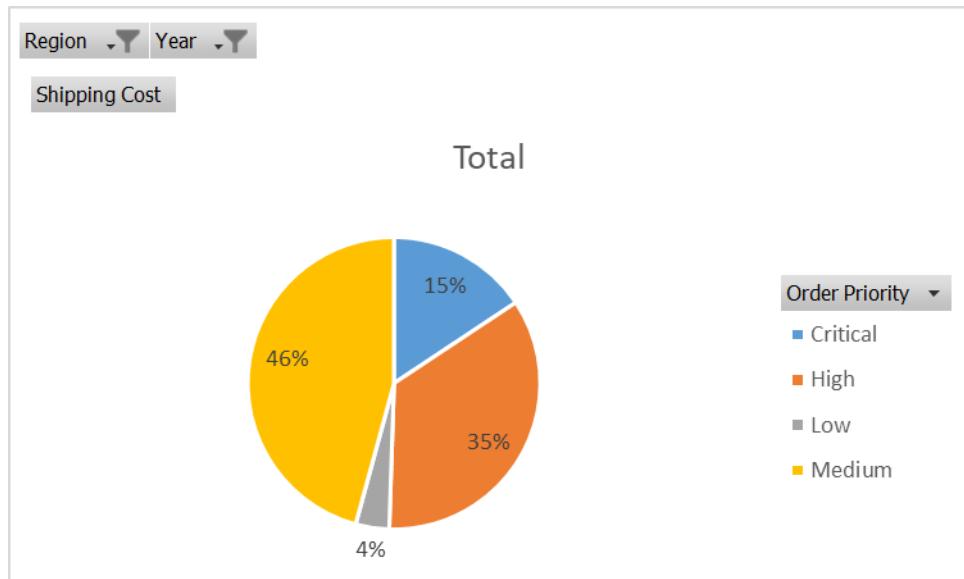
Câu 8: Thống kê các sản phẩm (id, tên sản phẩm) có tổng doanh thu lớn hơn 75000.		
Row Labels	Sales	
Apple Smart Phone, Full Size	86935.77905	
Cisco Smart Phone, Full Size	76441.53163	
<b>Grand Total</b>	<b>163377.3107</b>	

Hình 4.46: Kết quả truy vấn trên Excel – câu 8

**Câu 9: Thống kê tổng chi phí vận chuyển theo mức độ ưu tiên của đơn hàng thuộc khu vực Africa trong năm 2021 (theo ngày ship hàng).**

Câu 9: Thống kê tổng chi phí vận chuyển theo mức độ ưu tiên của đơn hàng thuộc khu vực Africa trong năm 2021 (theo ngày ship hàng)		
Region	Africa	<input type="button" value="▼"/>
Year	2021	<input type="button" value="▼"/>
Row Labels	Shipping Cost	
Critical	4785.689993	
High	10647.85001	
Low	1133.760006	
Medium	13979.32005	
<b>Grand Total</b>	<b>30546.62005</b>	

Hình 4.47: Kết quả truy vấn trên Excel – câu 9



Hình 4.48: Kết quả truy vấn trên Excel – câu 9 – Chart

**Câu 10: Thống kê Top 3 tên sản phẩm có số lượng được bán ra nhiều nhất.**

Câu 10: Thống kê Top 3 tên sản phẩm có số lượng được bán ra nhiều nhất.	
Row Labels	Quantity
Cardinal Index Tab, Clear	337
Eldon File Cart, Single Width	321
Rogers File Cart, Single Width	262
<b>Grand Total</b>	<b>920</b>

Hình 4.49: Kết quả truy vấn trên Excel – câu 10

**Câu 11: Thống kê tổng số lượng sản phẩm được bán ra thuộc danh mục “Technology” cho các khách hàng “Corporate” theo loại thị trường (Market)**

Câu 11: Thống kê tổng số lượng sản phẩm được bán ra thuộc danh mục “Technology” cho các khách hàng “Corporate” theo loại thị trường (Market)	
Category	Technology
Segment	Corporate
<b>Row Labels</b> <b>Quantity</b>	
APAC	2580
EMEA	3665
LATAM	2218
USCA	2131
<b>Grand Total</b>	<b>10594</b>

Hình 4.50: Kết quả truy vấn trên Excel – câu 11

**Câu 12: Thống kê top 2 khách hàng mua hàng mang lại tổng lợi nhuận cao nhất trong quý 4 năm 2020**

Câu 12: Thống kê top 2 khách hàng mua hàng mang lại tổng lợi nhuận cao nhất trong quý 4 năm 2020											
Quarter	4	Year	2020								
Row Labels	Profit										
Tamara Chand	8451.143877										
Adrian Barton	5025.650117										
<b>Grand Total</b>	<b>13476.79399</b>										

Hình 4.51: Kết quả truy vấn trên Excel – câu 12

#### 4.4.2.4 Pivot

### Câu 13: Thống kê tổng lợi nhuận theo quý và khu vực

Câu 13: Thống kê tổng lợi nhuận theo quý và khu vực															
Profit	Column Labels														
Row Labels	Africa	Canada	Caribbean	Central	Central Asia	East	EMEA	North	North Asia	Oceania	South	Southeast Asia	West	Grand Total	
1	7632.48298	697.889997	506.8872634	10458.47945	6416.406003	4286.436681	3453.174006	14517.65174	6456.912004	7809.327008	11745.09797	-364.3788959	2255.738903	75872.1051	
1	4332.159018	1917.056016	1786.754643	20169.97834	6608.465917	2049.212109	5824.997993	6182.873495	8044.068087	3950.748044	10568.04054	773.0769289	14258.26408	86465.69921	
1	3019.770052	536.5799942	794.6068057	8046.379332	2215.283971	-780.0422926	3076.541955	4878.486205	7503.498036	2703.596999	1988.967535	-255.555001	1804.382395	35532.49599	
1	1603.178899	128.490007	315.0346087	8778.950682	4851.839972	1647.202361	956.27405	5393.9793	6531.509992	1380.981033	4951.447215	1607.554505	5277.844552	43424.28726	
2	-434.3849711	204.0299982	1088.049409	10843.1222	4382.474906	1508.276472	-383.8649836	7754.097237	6296.105882	7241.196026	4488.696474	1546.541711	4261.452035	48795.7924	
2	11731.35903	396.5100061	2149.231404	19687.61873	9774.815793	5125.865188	2760.161957	16146.59091	8671.008041	12337.45505	12716.01119	-2448.401701	3196.886486	102245.1121	
2	2737.716006	2724.210013	663.6775929	12315.25059	8548.211919	1798.947111	14865.53407	3735.635941	14076.91785	10540.350553	1887.889822	4008.292223	82260.09884		
2	5349.936016	536.2199955	2211.76076	17719.29291	9513.629939	8253.412262	5637.242999	10267.33629	11592.26383	9191.628051	6754.138247	1752.184703	4554.528750	9333.57482	
3	4219.410131	107.460014	2637.086963	27727.8955	5192.283001	5878.772903	3153.43203	13503.72199	8574.053944	4758.155994	7311.718101	697.0191126	3781.985487	88533.73857	
3	7810.493959	2408.370031	1789.458218	21885.53435	8200.100881	3118.755366	-2137.529921	12588.17689	16254.60914	9634.934983	8439.586084	33.83190087	8066.068873	98092.39075	
3	5180.592051	801.5699842	-88.40856656	14114.10808	4004.804987	5823.756093	894.92407	10836.10693	8469.186036	1828.73091	5053.261693	1043.356791	7576.137888	65538.12694	
3	10689.59702	1112.280004	3484.309974	29201.5466	15647.50205	6560.88984	8643.471074	19225.05776	13996.53295	3358.397972	22263.88218	2359.75018	15453.90643	151997.124	
4	6388.602094	1485.269997	4465.43307	46756.35707	8979.044985	4491.548458	3673.415976	13725.36576	14072.27097	10650.96609	12874.40101	1754.546113	9374.530782	138691.7524	
4	12096.37513	2567.160025	5112.319808	29079.32161	15516.69397	19486.03762	5338.931968	15103.82648	21956.71187	11950.6231	6137.681421	8010.596726	10700.63855	163056.9183	
4	3071.997007	261.7799931	2564.661254	19674.86329	12243.88201	10507.61924	1662.789123	12423.59176	13244.58287	9566.087953	6318.176712	906.6529158	6423.718779	98872.40291	
4	3336.506948	939.3899895	5089.737758	14945.2827	10384.74589	9206.462354	488.5779932	17211.61087	10179.47102	9560.216986	8204.285761	-1454.337082	7424.072364	9452.86757	
4	106.1399994	3.119999886						29.63999939	-26.05500031	89.14799976				201.9929981	
<b>Grand Total</b>	<b>88871.63147</b>	<b>17817.39007</b>	<b>34571.32096</b>	<b>311403.9812</b>	<b>132408.1862</b>	<b>91522.77982</b>	<b>43897.97142</b>	<b>194597.9527</b>	<b>165578.4206</b>	<b>120089.1121</b>	<b>140355.7663</b>	<b>17852.32873</b>	<b>108418.4486</b>	<b>1467457.29</b>	

Hình 4.52: Kết quả truy vấn trên Excel – câu 13

### Câu 14: Thống kê tổng số đơn hàng được bán ra theo từng loại thị trường và danh mục sản phẩm.

Câu 14: Thống kê tổng số đơn hàng được bán ra theo từng loại thị trường và danh mục sản phẩm.					
Fact Count	Column Labels				
Row Labels	Furniture	Office Supplies	Technology	Grand Total	
APAC	2429	6177	2396	11002	
EMEA	2902	12931	3783	19616	
LATAM	2382	5862	2050	10294	
USCA	2163	6303	1912	10378	
<b>Grand Total</b>	<b>9876</b>	<b>31273</b>	<b>10141</b>	<b>51290</b>	

Hình 4.53: Kết quả truy vấn trên Excel – câu 14

### Câu 15: Thống kê tổng chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng

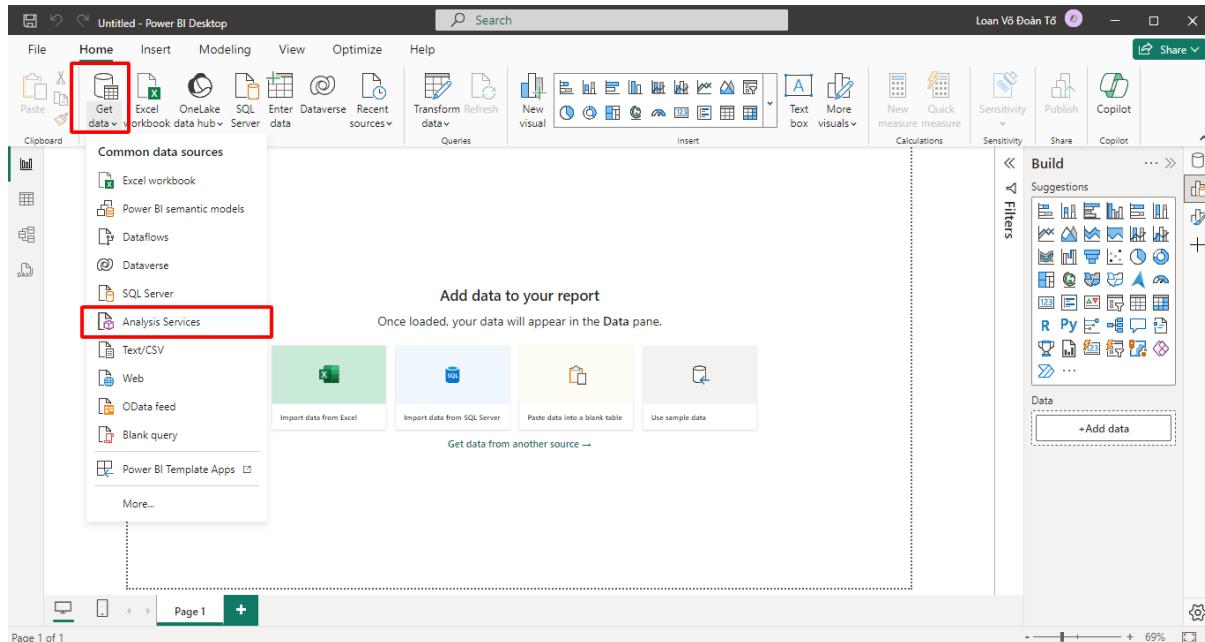
Câu 15: Thống kê tổng chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng								
Column Labels								
Row Labels	Critical	High	Low	Medium	Total Profit	Total Shipping Cost		
Consumer	59875.98341	120643.5314	227212.2153	275309.1629	22447.2178	30805.81504	439704.3655	27059.6107
Corporate	41646.11777	73553.35606	111719.0098	144077.4088	27295.37725	20391.89903	260547.8232	172450.248
Home Office	22702.06239	40627.04806	81442.28824	90159.295	8913.255626	14435.37101	163951.5738	9982.95746
<b>Grand Total</b>	<b>124224.1636</b>	<b>234823.9355</b>	<b>420373.5133</b>	<b>509545.8667</b>	<b>58655.85068</b>	<b>65633.08508</b>	<b>864203.7626</b>	<b>542812.8161</b>
								<b>1467457.29</b>
								<b>1352815.703</b>

Hình 4.54: Kết quả truy vấn trên Excel – câu 15

## 4.5 Thực thi 15 câu truy vấn bằng Power BI

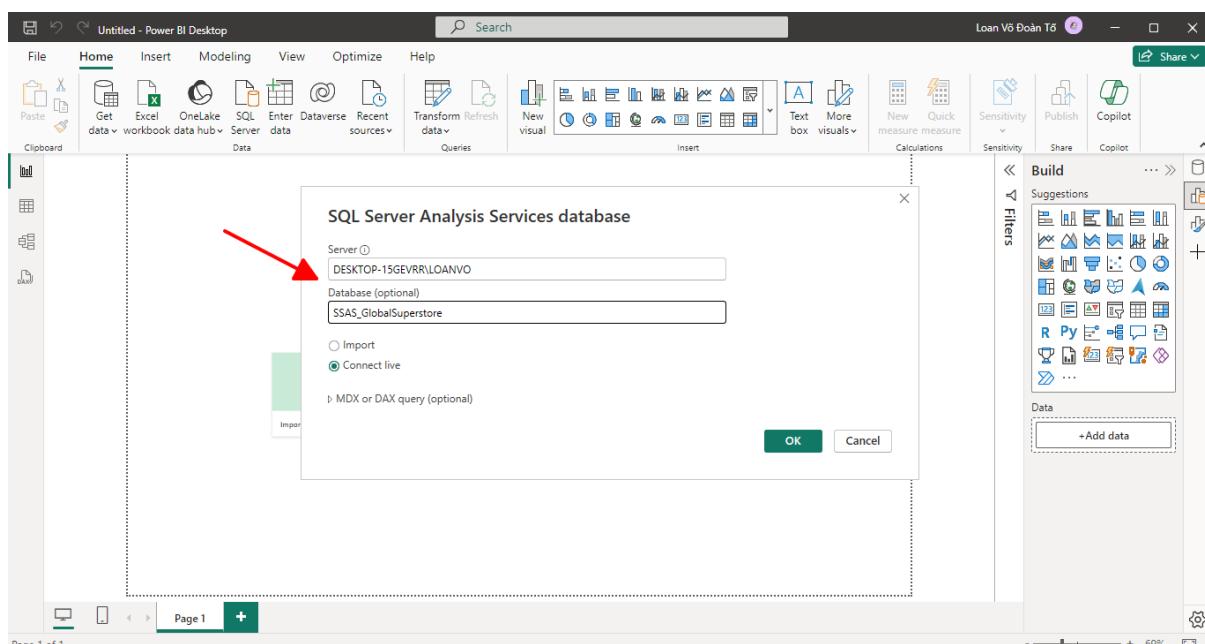
### 4.5.1 Thực hiện kết nối Microsoft Analysis Services

Bước 1: Mở Power BI -> Get Data -> Analysis Services

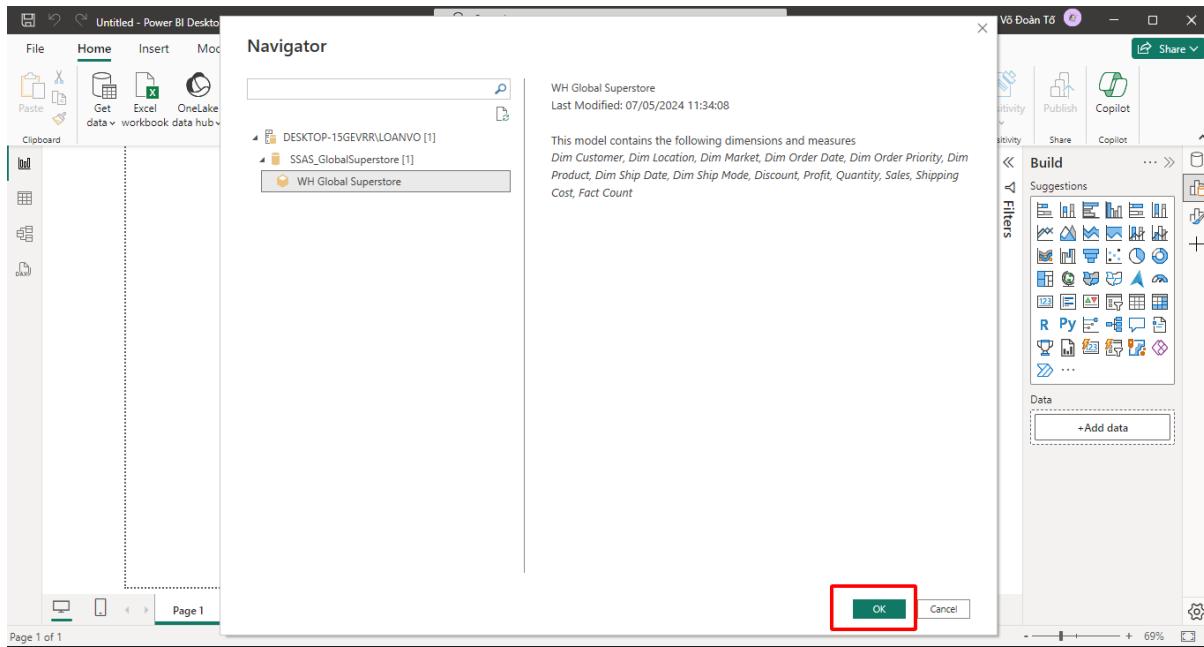


Bước 2: Điền các thông tin của SQL Server Analysis Services database vào:

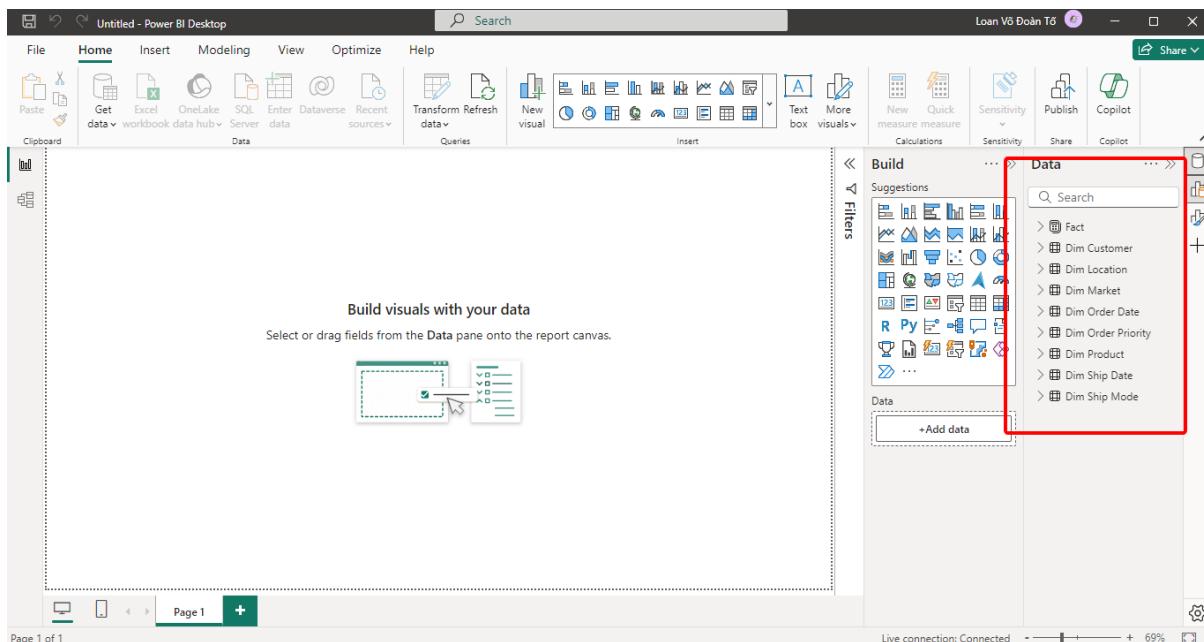
- Tên Server
- Tên Database



Bước 3: Xem các bảng dữ liệu có trong database đó -> chọn OK



#### Bước 4: Power BI đã lấy các bảng Fact và Dim thành công từ Analysis Services



## 4.5.2 Thực hiện các truy vấn

### 4.5.2.1 Roll Up Queries

Câu 1: Thống kê tổng số lượng sản phẩm bán ra theo từng tháng, năm.

Kết quả truy vấn trên Power BI:

The screenshot shows the Power BI interface with a table visual on the left. The table has columns: Month, Year, and Quantity. The data includes rows for months from 1 to 12 and years from 2017 to 2021, with a total row at the bottom. On the right, the 'Build' pane is open, specifically the 'Columns' section. A red box highlights the 'Columns' section, which contains three columns: Month, Year, and Quantity, each with a 'X' button to remove. Below this is a '+Add data' button. The 'Fact' section of the data pane is also visible, showing various measures and dimensions.

Hình 4.55: Kết quả truy vấn trên Power BI – câu 1

## Câu 2: Tính tổng doanh thu theo từng danh mục con, danh mục.

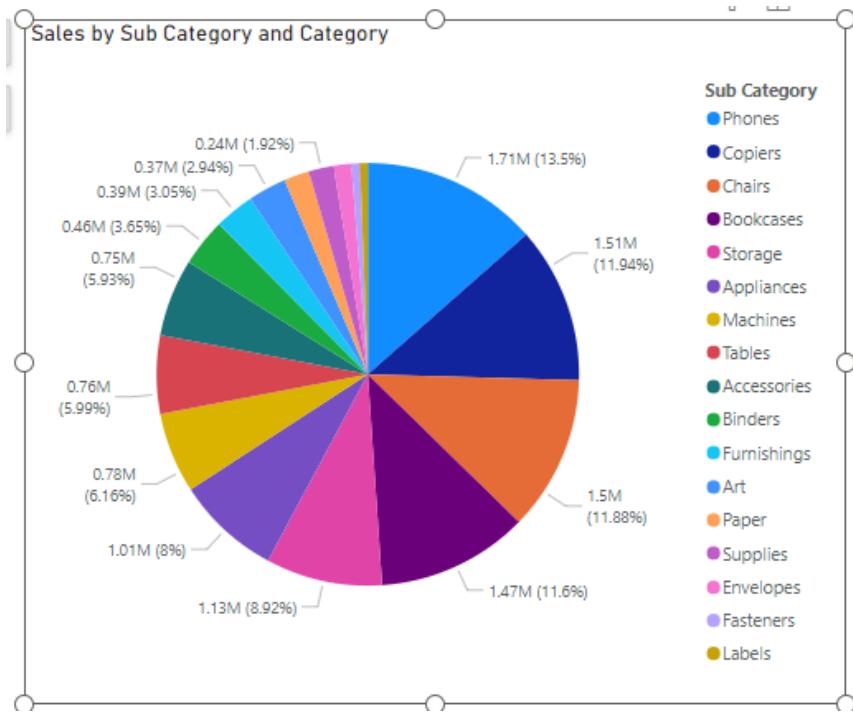
Kết quả truy vấn trên Power BI:

+ Dạng table:

The screenshot shows the Power BI interface with a table visual on the left. The table has columns: Sub Category, Category, and Sales. The data includes rows for various sub-categories and categories, with a total row at the bottom. On the right, the 'Build' pane is open, specifically the 'Columns' section. A red box highlights the 'Columns' section, which contains three columns: Sub Category, Category, and Sales, each with a 'X' button to remove. Below this is a '+Add data' button. The 'Fact' section of the data pane is also visible, showing various measures and dimensions.

Hình 4.56: Kết quả truy vấn trên Power BI – câu 2 – Table

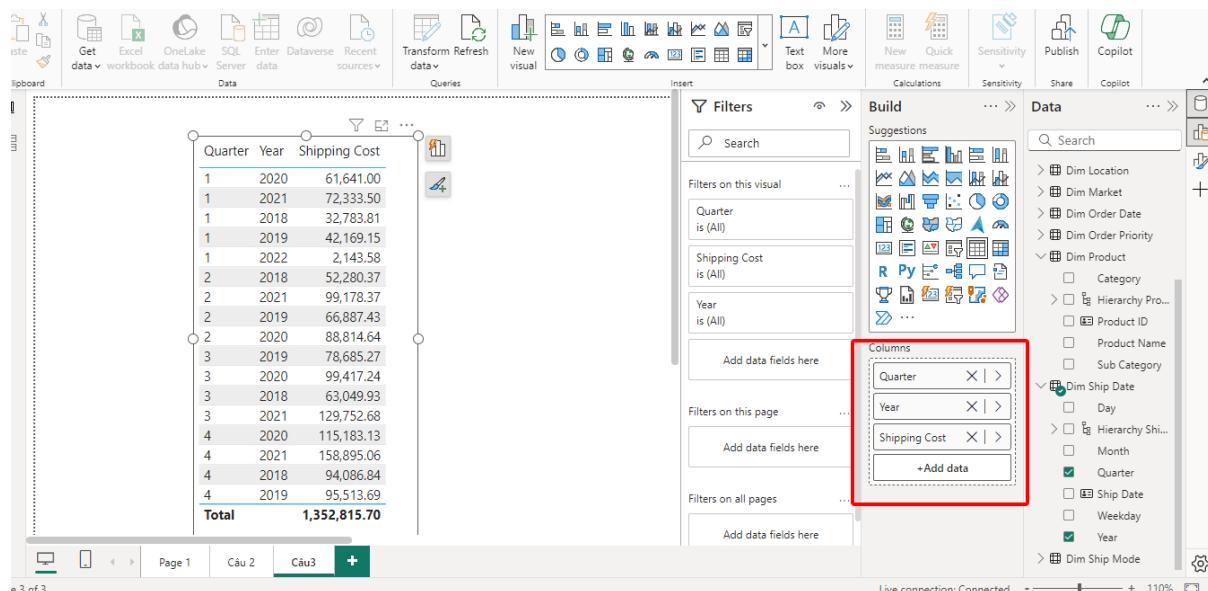
+ Dạng chart:



Hình 4.57: Kết quả truy vấn trên Power BI – câu 2 – Chart

### Câu 3: Tổng chi phí vận chuyển theo từng quý, từng năm của ngày vận chuyển.

Kết quả truy vấn trên Power BI:



Hình 4.58: Kết quả truy vấn trên Power BI – câu 3

#### 4.5.2.2 Drill Down Queries

### Câu 4: Thống kê tổng lợi nhuận từng năm, từng quý.

Kết quả truy vấn trên Power BI:

Year    Quarter    Profit

2017	4	201.99
2018	1	35,532.50
2018	2	48,795.79
2018	3	65,538.13
2018	4	98,872.40
2019	1	43,424.29
2019	2	82,260.91
2019	3	88,533.74
2019	4	94,542.87
2020	1	75,872.11
2020	2	93,333.57
2020	3	98,092.39
2020	4	138,691.75
2021	1	86,465.70
2021	2	102,245.11
2021	3	151,997.12
2021	4	163,056.92
<b>Total</b>		<b>1,467,457.29</b>

Hình 4.59: Kết quả truy vấn trên Power BI – câu 4

### Câu 5: Thống kê tổng chiết khấu theo từng khu vực, từng quốc gia.

Kết quả truy vấn trên Power BI:

Region    Country    Discount

Africa	Guinea	0.00
Africa	Guinea-Bissau	0.00
Africa	Kenya	0.00
Africa	Lesotho	0.00
Africa	Liberia	0.00
Africa	Libya	0.00
Africa	Madagascar	0.00
Africa	Mali	0.00
Africa	Mauritania	0.00
Africa	Morocco	0.00
Africa	Mozambique	0.00
Africa	Namibia	0.00
Africa	Niger	0.00
Africa	Nigeria	633.50
Africa	Republic of the Congo	0.00
Africa	Rwanda	0.00
<b>Total</b>		<b>7,329.73</b>

Hình 4.60: Kết quả truy vấn trên Power BI – câu 5

### Câu 6: Thống kê tổng lợi nhuận theo từng danh mục sản phẩm, danh mục con sản phẩm.

Kết quả truy vấn trên Power BI:

Table visual showing data for Category, Sub Category, and Profit. The total profit is 1,467,457.29.

Category	Sub Category	Profit
Furniture	Bookcases	161,924.42
Furniture	Chairs	140,396.27
Furniture	Furnishings	46,967.43
Furniture	Tables	-64,083.39
Office Supplies	Appliances	141,680.59
Office Supplies	Art	57,953.91
Office Supplies	Binders	72,449.85
Office Supplies	Envelopes	29,601.12
Office Supplies	Fasteners	11,525.42
Office Supplies	Labels	15,010.51
Office Supplies	Paper	59,207.68
Office Supplies	Storage	108,461.49
Office Supplies	Supplies	22,583.26
Technology	Accessories	129,626.31
Technology	Copiers	258,567.55
Technology	Machines	58,867.87
<b>Total</b>		<b>1,467,457.29</b>

Hình 4.61: Kết quả truy vấn trên Power BI – câu 6

#### 4.5.2.3 Slice and Dice Queries

Câu 7: Trong năm 2020 thống kê 3 quốc gia có tổng số lượng sản phẩm bán ra nhiều nhất.

Kết quả truy vấn trên Power BI:

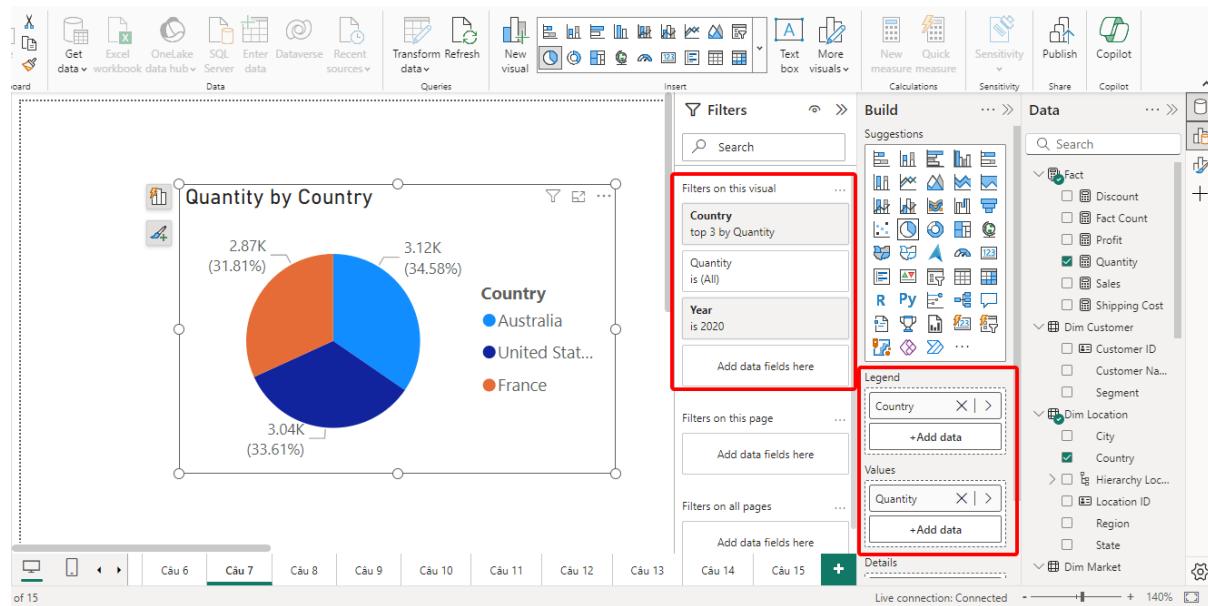
+ Dạng matrix:

Matrix visual showing data for Country and Quantity. The total quantity is 9034.

Country	Quantity
Australia	3124
France	2874
United States	3036
<b>Total</b>	<b>9034</b>

Hình 4.62: Kết quả truy vấn trên Power BI – câu 7 – Table

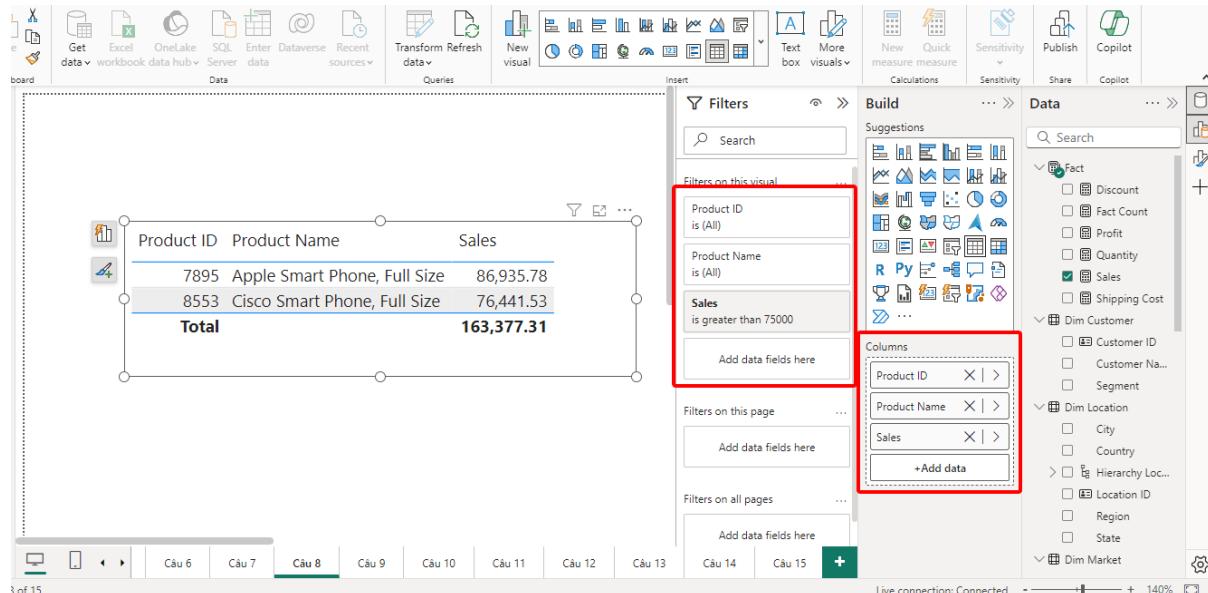
+ Dạng Chart:



Hình 4.63: Kết quả truy vấn trên Power BI – câu 7 – Chart

**Câu 8: Thống kê các sản phẩm (id, tên sản phẩm) có tổng doanh thu lớn hơn 75000.**

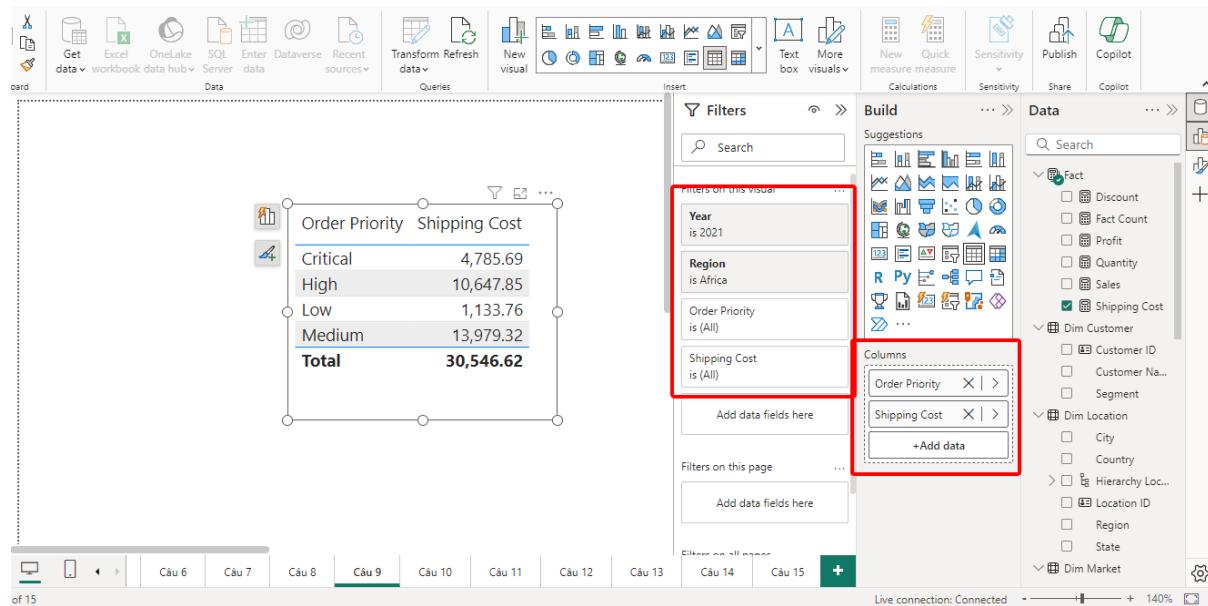
Kết quả truy vấn trên Power BI:



Hình 4.64: Kết quả truy vấn trên Power BI – câu 8

**Câu 9: Thống kê tổng chi phí vận chuyển theo mức độ ưu tiên của đơn hàng thuộc khu vực Africa trong năm 2021 (theo ngày ship hàng).**

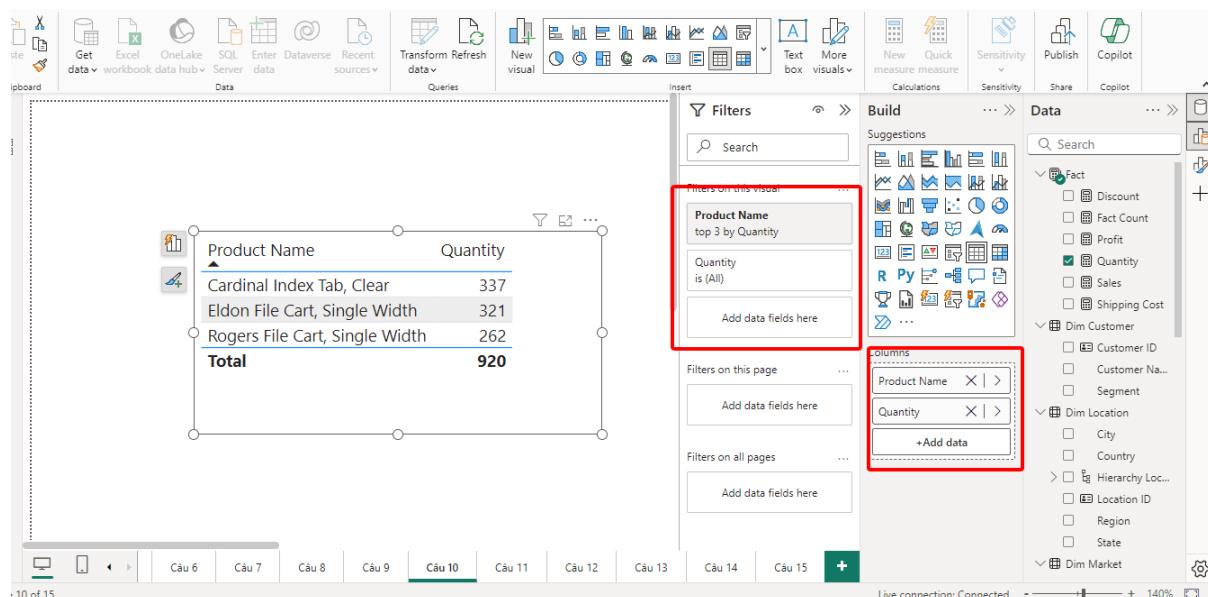
Kết quả truy vấn trên Power BI:



Hình 4.65: Kết quả truy vấn trên Power BI – câu 9

### Câu 10: Thống kê Top 3 tên sản phẩm có số lượng được bán ra nhiều nhất.

Kết quả truy vấn trên Power BI:

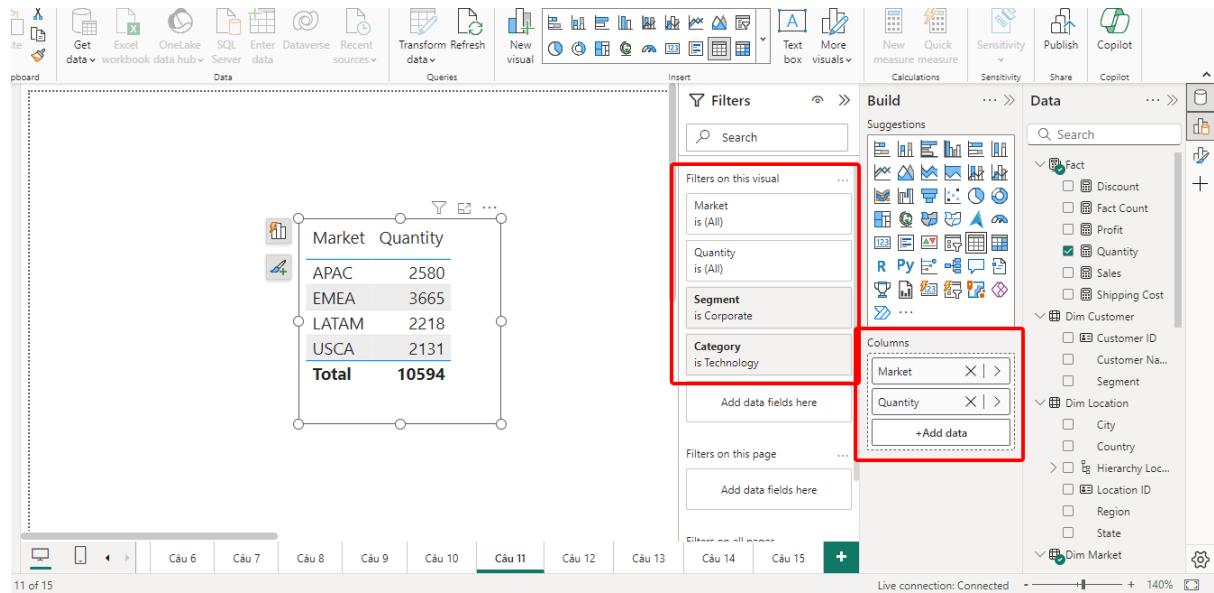


Hình 4.66: Kết quả truy vấn trên Power BI – câu 10

### Câu 11: Thống kê tổng số lượng sản phẩm được bán ra thuộc danh mục

“Technology” cho các khách hàng “Corporate” theo loại thị trường (Market)

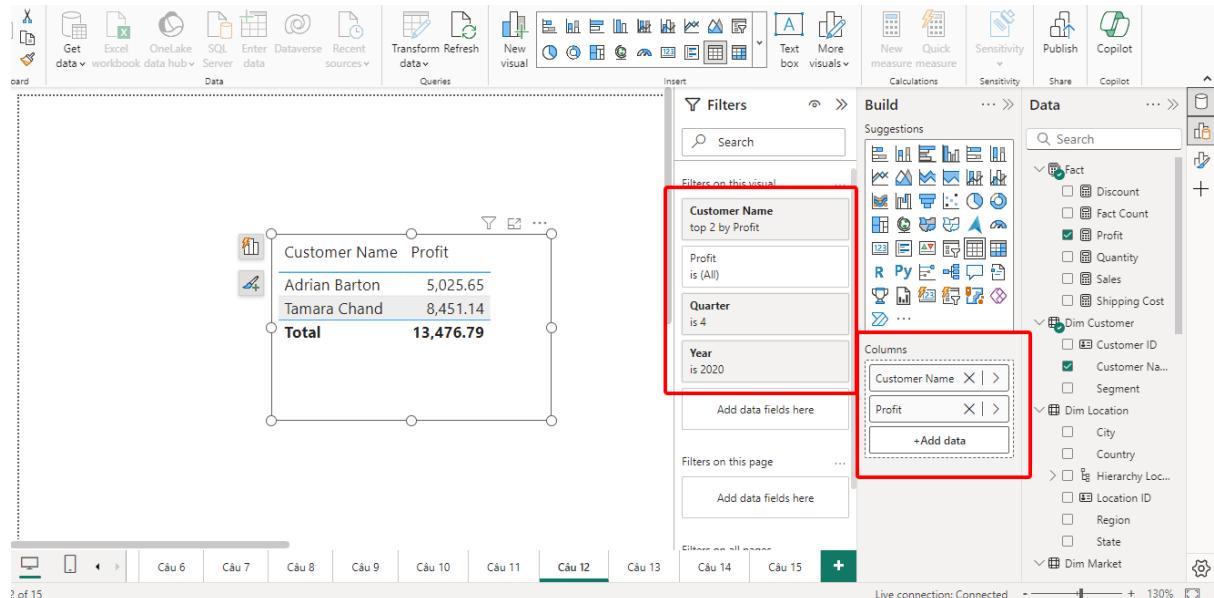
Kết quả truy vấn trên Power BI:



Hình 4.67: Kết quả truy vấn trên Power BI – câu 11

**Câu 12: Thống kê top 2 khách hàng mua hàng mang lại tổng lợi nhuận cao nhất trong quý 4 năm 2020**

Kết quả truy vấn trên Power BI:



Hình 4.68: Kết quả truy vấn trên Power BI – câu 12

#### 4.5.2.4 Pivot

**Câu 13: Thống kê tổng lợi nhuận theo quý và khu vực**

Kết quả truy vấn trên Power BI:

+ **Dạng Table**

Quarter Region Profit

1	Africa	7,632.48
1	Africa	4,332.16
1	Africa	3,019.77
1	Africa	1,603.18
2	Africa	-434.38
2	Africa	11,731.36
2	Africa	2,737.42
2	Africa	5,349.94
3	Africa	4,219.41
3	Africa	7,810.49
3	Africa	5,180.59
3	Africa	10,689.60
4	Africa	6,388.60
4	Africa	12,096.38
4	Africa	3,072.00
4	Africa	3,336.51
4	Africa	106.14
1	Canada	697.89
1	Canada	1,917.06
1	Canada	536.58
<b>Total</b>		<b>1,467,457.29</b>

Hình 4.69: Kết quả truy vấn trên Power BI – câu 13 – Table

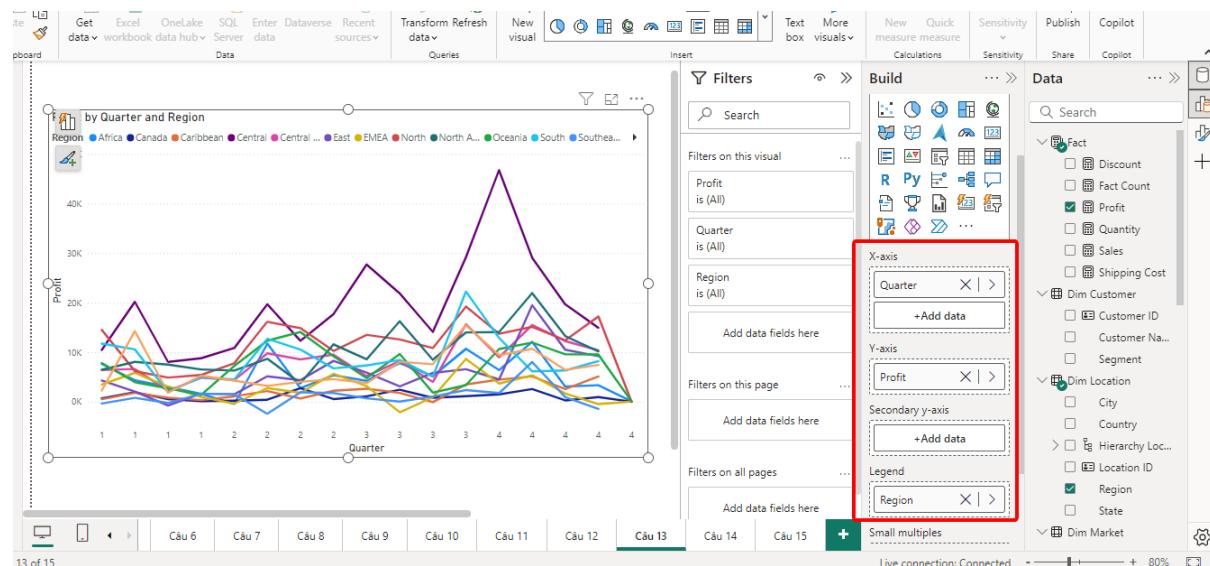
+ Dạng Matrix:

Quarter Central Central Asia East EMEA North North Asia Oceania South Southeast Asia West Total

1	10,458.48	6,416.41	4,286.44	3,453.17	14,517.65	6,456.91	7,809.33	11,745.10	-364.38	2,255.74	<b>75,872.11</b>	
1	20,169.98	6,608.47	2,049.21	5,825.00	6,182.87	8,044.07	3,950.75	10,568.04	773.08	14,258.26	<b>86,465.70</b>	
1	8,046.38	2,215.28	-780.04	3,076.54	4,878.49	7,503.50	2,703.60	1,988.97	-255.56	1,804.38	<b>35,532.50</b>	
1	8,778.95	4,651.84	1,647.20	956.27	5,393.98	6,531.51	1,380.98	4,951.45	1,607.55	5,277.84	<b>43,424.29</b>	
2	10,843.12	4,382.47	1,503.28	-383.86	7,754.10	6,296.11	7,241.20	4,488.70	1,546.54	4,261.45	<b>48,795.79</b>	
2	19,687.62	9,774.82	5,125.87	2,760.16	16,146.59	8,671.01	12,337.46	12,716.01	-2,448.40	3,196.89	<b>102,245.11</b>	
2	12,315.25	6,548.21	4,358.58	1,798.95	14,865.53	3,735.64	14,076.92	10,540.35	1,887.89	4,008.29	<b>82,260.91</b>	
2	17,719.29	9,513.63	8,253.41	5,637.24	10,267.34	11,592.26	9,191.63	6,754.14	1,752.18	4,554.53	<b>93,333.57</b>	
3	27,727.98	5,192.28	5,878.77	3,153.43	13,503.72	8,574.05	4,758.16	7,311.74	697.02	3,781.99	<b>88,533.74</b>	
3	21,885.53	8,200.10	3,118.77	-2,137.53	12,586.18	16,254.61	9,634.93	8,439.59	33.83	8,066.07	<b>98,092.39</b>	
3	14,114.11	4,004.80	5,823.76	894.92	10,836.11	8,469.19	1,828.73	5,053.26	1,043.36	7,576.14	<b>65,538.13</b>	
3	29,201.55	15,647.50	6,560.88	8,643.47	19,225.06	13,996.53	3,358.40	22,263.88	2,359.75	15,453.91	<b>151,997.12</b>	
4	46,756.38	8,979.04	4,491.55	3,673.42	13,723.57	14,072.27	10,650.97	12,874.40	1,754.55	9,374.53	<b>138,691.75</b>	
4	29,079.32	15,516.69	19,486.03	5,338.93	10,507.62	21,956.71	11,950.62	6,137.68	8,010.60	10,700.64	<b>163,056.92</b>	
4	19,674.86	12,243.88	1,662.79	12,423.59	13,244.58	9,566.09	6,318.18	908.65	6,423.72	98,872.40		
4	14,945.28	10,394.75	9,206.46	-484.58	17,211.87	10,179.47	9,560.22	8,204.29	-1,454.34	7,424.07	<b>94,542.87</b>	
					29.64	-26.06	89.15				201.99	
<b>Total</b>		<b>311,403.98</b>	<b>132,480.19</b>	<b>91,522.78</b>	<b>43,879.97</b>	<b>194,597.95</b>	<b>165,578.42</b>	<b>120,089.11</b>	<b>140,355.77</b>	<b>17,852.33</b>	<b>108,418.45</b>	<b>1,467,457.29</b>

Hình 4.70: Kết quả truy vấn trên Power BI – câu 13 – Matrix

+ Dạng Chart:

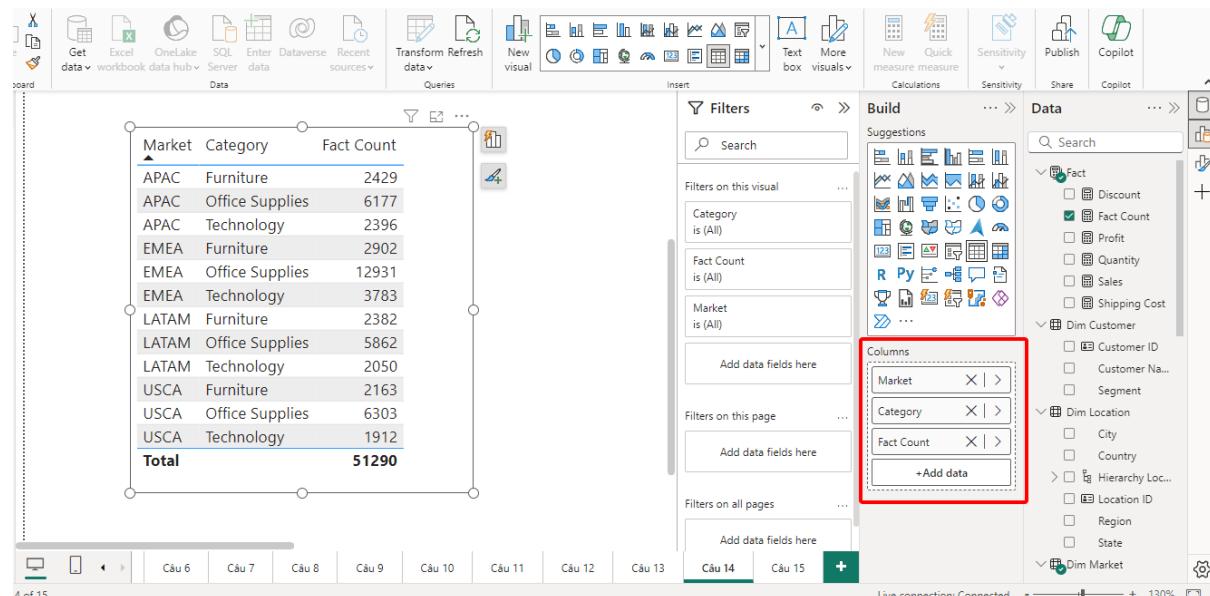


Hình 4.71: Kết quả truy vấn trên Power BI – câu 13 – Chart

**Câu 14: Thông kê tổng số đơn hàng được bán ra theo từng loại thị trường và danh mục sản phẩm.**

Kết quả truy vấn trên Power BI:

+ **Dạng Table**



Hình 4.72: Kết quả truy vấn trên Power BI – câu 14 – Table

+ **Dạng Matrix:**

	Market	Furniture	Office Supplies	Technology	Total
EMEA	2902	12931	3783	19616	
APAC	2429	6177	2396	11002	
USCA	2163	6303	1912	10378	
LATAM	2382	5862	2050	10294	
<b>Total</b>	<b>9876</b>	<b>31273</b>	<b>10141</b>	<b>51290</b>	

Hình 4.73: Kết quả truy vấn trên Power BI – câu 14 – Matrix

+ Dạng Chart:

Fact Count by Market and Category

Category: Furniture (blue), Office Supplies (dark blue), Technology (orange)

Market	Furniture	Office Supplies	Technology
EMEA	~2900	~12931	~3783
APAC	~2429	~6177	~2396
USCA	~2163	~6303	~1912
LATAM	~2382	~5862	~2050

Hình 4.74: Kết quả truy vấn trên Power BI – câu 14 – Chart

**Câu 15: Thống kê tổng chi phí vận chuyển và lợi nhuận theo phân loại khách hàng và mức độ ưu tiên đơn hàng**

Kết quả truy vấn trên Power BI:

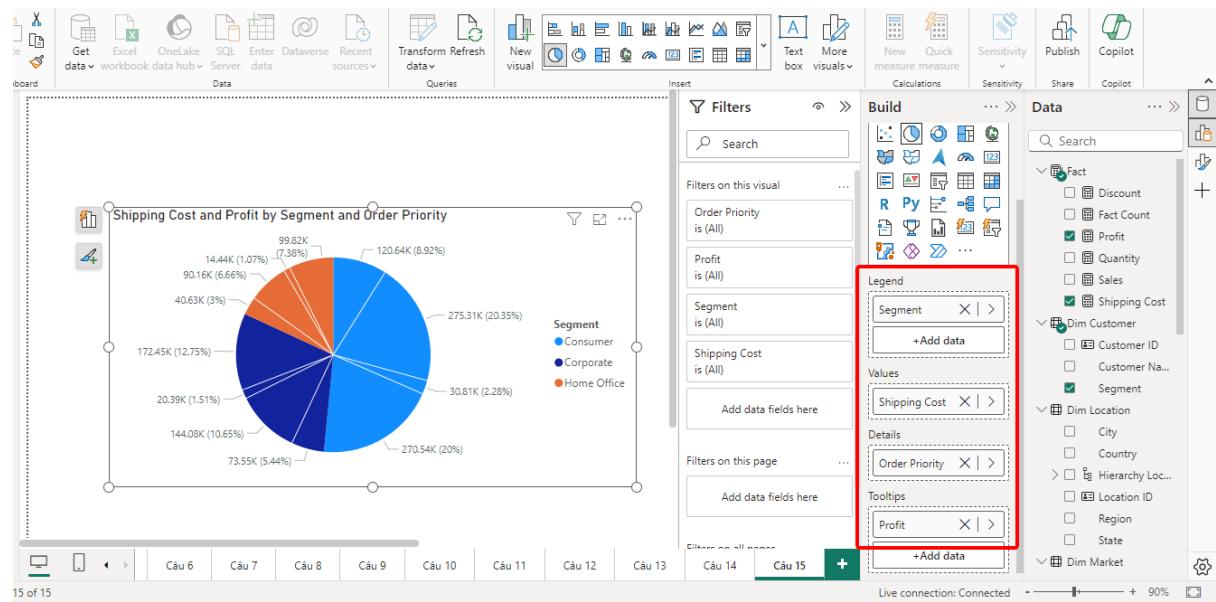
+ Dạng Table:

Hình 4.75: Kết quả truy vấn trên Power BI – câu 15 – Table

+ Dạng Matrix:

Hình 4.76: Kết quả truy vấn trên Power BI – câu 15 – Matrix

+ Dạng Chart:



Hình 4.77: Kết quả truy vấn trên Power BI – câu 15 – Chart

## CHƯƠNG 5: QUÁ TRÌNH KHAI THÁC DỮ LIỆU (DATA MINING)

### 5.1 Chủ đề khai thác

Mục tiêu của bài toán là dự đoán khả năng sinh lợi nhuận của một đơn hàng dựa trên các đặc trưng như giá bán, ngày đặt hàng, ngày giao hàng, địa chỉ đơn hàng và các thông tin liên quan khác. Việc dự đoán này giúp các doanh nghiệp tối ưu hóa chiến lược kinh doanh, quản lý tồn kho, và nâng cao hiệu quả hoạt động.



Hình 5.1: Chủ đề cho bài toán khai thác dữ liệu

### 5.2 Import dữ liệu và thư viện cần thiết

Import các thư viện cần thiết như sau:

- **pandas** và **numpy** dùng để xử lý và phân tích dữ liệu.
- **matplotlib.pyplot** và **seaborn** giúp trực quan hóa dữ liệu.
- Các thuật toán học máy như **Perceptron**, **LogisticRegression**, **GaussianNB** và **RandomForestClassifier** từ **scikit-learn** được sử dụng để xây dựng mô hình dự đoán.
- **GridSearchCV** để tìm tham số tối ưu, **train\_test\_split** để chia dữ liệu, và các công cụ như **accuracy\_score** và **classification\_report** để đánh giá mô hình.
- **SelectKBest** và **f\_classif** giúp chọn ra các đặc trưng quan trọng.
- **pickle** dùng để lưu trữ và tải lại các mô hình đã huấn luyện.

### Import libraries

```
[1]: # Import libraries for dataset manipulation and numerical operations
import pandas as pd
import numpy as np

# Import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Import Libraries for machine Learning models and evaluation
from sklearn.linear_model import Perceptron, LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier

# Import Libraries for model selection and evaluation
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Import Libraries for feature selection
from sklearn.feature_selection import SelectKBest, f_classif

# Import library for saving and loading models
import pickle
```

Sử dụng pandas để đọc tệp csv và truyền vào địa chỉ tệp file .csv chứa dữ liệu

### Import dataset

```
[2]: df = pd.read_csv("data/Global_Superstore.csv")
```

## 5.3 Thực hiện EDA

Xem 10 dòng dữ liệu đầu tiên của tập dữ liệu.

```
[4]: # View 10 samples first
df.head(10)
```

	Global_Orders_ID	Order_ID	Category	City	Country	Customer_Name	Market	Customer_ID	Order_Date	Ship_Date	...	Region	Segment	Ship_Mode
0	1	CA-2013-158568	Technology	Chicago	United States	Rick Bensley	USCA	RB-194654	8/28/2020	9/1/2020	...	Central	Home Office	Standard Class
1	2	CA-2013-158568	Office Supplies	Chicago	United States	Rick Bensley	USCA	RB-194654	8/28/2020	9/1/2020	...	Central	Home Office	Standard Class
2	3	CA-2013-161207	Office Supplies	Concord	United States	Adam Bellavance	USCA	AB-100604	8/28/2020	9/2/2020	...	East	Home Office	Standard Class
3	4	CA-2013-128727	Technology	New York City	United States	Meg O'Connel	USCA	MO-178004	8/28/2020	9/3/2020	...	East	Home Office	Standard Class
4	5	CA-2013-159912	Furniture	Philadelphia	United States	George Bell	USCA	GB-145304	8/28/2020	9/2/2020	...	East	Corporate	Standard Class
5	6	CA-2013-159912	Furniture	Philadelphia	United States	George Bell	USCA	GB-145304	8/28/2020	9/2/2020	...	East	Corporate	Standard Class
6	7	CA-2013-159912	Office Supplies	Philadelphia	United States	George Bell	USCA	GB-145304	8/28/2020	9/2/2020	...	East	Corporate	Standard Class
7	8	CA-2013-159912	Office Supplies	Philadelphia	United States	George Bell	USCA	GB-145304	8/28/2020	9/2/2020	...	East	Corporate	Standard Class
8	9	KE-2013-2920	Office Supplies	Nairobi	Kenya	Joe Kamberova	EMEA	JK-57301	8/28/2020	8/28/2020	...	Africa	Consumer	Same Day
9	10	AG-2013-8490	Office Supplies	Algiers	Algeria	Sally Knutson	EMEA	SK-99901	8/28/2020	8/31/2020	...	Africa	Consumer	Second Class

10 rows × 23 columns

Xem thông tin cơ bản của tập dữ liệu: Gồm 51290 dòng dữ liệu và 23 thuộc tính.

```
[5]: # View basic information
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Global_Orders_ID  51290 non-null   int64  
 1   Order_ID          51290 non-null   object  
 2   Category          51290 non-null   object  
 3   City              51290 non-null   object  
 4   Country            51290 non-null   object  
 5   Customer_Name     51290 non-null   object  
 6   Market             51290 non-null   object  
 7   Customer_ID        51290 non-null   object  
 8   Order_Date         51290 non-null   object  
 9   Ship_Date          51290 non-null   object  
 10  Order_Priority    51290 non-null   object  
 11  Product_ID        51290 non-null   object  
 12  Product_Name      51290 non-null   object  
 13  Region             51290 non-null   object  
 14  Segment            51290 non-null   object  
 15  Ship_Mode          51290 non-null   object  
 16  State              51290 non-null   object  
 17  Sub_Category       51290 non-null   object  
 18  Discount           51290 non-null   float64 
 19  Profit              51290 non-null   float64 
 20  Quantity            51290 non-null   int64  
 21  Sales              51290 non-null   float64 
 22  Shipping_Cost      51290 non-null   float64 
dtypes: float64(4), int64(2), object(17)
memory usage: 9.0+ MB
```

Sử dụng phương thức describe() của pandas để tạo ra các thống kê mô tả cho các thuộc tính định lượng trong DataFrame.

```
[7]: # Statistical statistics of quantitative attributes such as: count the number of values, maximum, minimum, mean, standard deviation, quartiles...
df.describe(include='all')

[7]:
```

	Global_Orders_ID	Order_ID	Category	City	Country	Customer_Name	Market	Customer_ID	Order_Date	Ship_Date	...	Region	Segment	Ship_Mode
count	51290.00000	51290	51290	51290	51290	51290	51290	51290	51290	51290	...	51290	51290	51290
unique	NaN	25035	3	3636	147	795	4	4873	1429	1463	...	13	3	4
top	NaN	CA-100111	Office Supplies	New York City	United States	Muhammed Yedwab	EMEA	JG-158051	6/16/2021	11/20/2021	...	Central	Consumer	Standard Class
freq	NaN	14	31273	915	9994	108	19616	40	135	130	...	11117	26518	30775
mean	25645.50000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
std	14806.29199	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
min	1.00000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
25%	12823.25000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
50%	25645.50000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
75%	38467.75000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
max	51290.00000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN

11 rows × 23 columns

Đếm số lượng các kiểu dữ liệu khác nhau trong tập dữ liệu. Ta thấy có 17 thuộc tính object, 4 thuộc tính float, 2 thuộc tính integer.

```
[9]: # Count data type
      dtype.value_counts()
```

```
[9]: types
      object      17
      float64     4
      int64       2
      dtype: int64
```

Kiểm tra và đếm số lượng giá trị thiếu (missing values) trong tập dữ liệu. Ta thấy tập dữ liệu không xuất hiện missing values.

```
[10]: # Check if dataset have missing values
      df.isna().sum()
```

```
[10]: Global_Orders_ID      0
      Order_ID            0
      Category            0
      City                0
      Country             0
      Customer_Name       0
      Market              0
      Customer_ID          0
      Order_Date          0
      Ship_Date           0
      Order_Priority       0
      Product_ID           0
      Product_Name          0
      Region              0
      Segment              0
      Ship_Mode            0
      State                0
      Sub_Category          0
      Discount             0
      Profit               0
      Quantity             0
      Sales                0
      Shipping_Cost         0
      dtype: int64
```

Kiểm tra dữ liệu trùng, ta thấy tập dữ liệu không có dữ liệu trùng nhau.

```
[11]: # Check if dataset have duplicate values
      df.duplicated().sum()
```

```
[11]: 0
```

## 5.4 Tiết xử lý dữ liệu

### 5.4.1 Tạo thuộc tính quyết định Profit\_Positive

Tạo thuộc tính **Profit\_Positive**: Sử dụng hàm apply để áp dụng hàm lambda cho từng giá trị trong cột Profit. Nếu giá trị lớn hơn 0, gán giá trị 1 (có lợi nhuận) cho cột Profit\_Positive, ngược lại gán giá trị 0 (lỗ).

Create the target column with a more descriptive name

```
[12]: df['Profit_Positive'] = df['Profit'].apply(lambda x: 1 if x > 0 else 0)
```

### 5.4.2 Xóa các thuộc tính không cần thiết

Tạo danh sách các thuộc tính không cần thiết cho quá trình dự đoán và loại bỏ chúng khỏi tập dữ liệu. Các thuộc tính này bao gồm “Global\_Orders\_ID”, “Order\_ID”, “Customer\_Name”, “Customer\_ID”, “Product\_ID”, “Product\_Name”.

#### Drop useless column

```
[13]: drop_columns = ['Global_Orders_ID', 'Order_ID', 'Customer_Name', 'Customer_ID', 'Product_ID', 'Product_Name']
# Drop columns that are not meaningful for prediction
df = df.drop(columns=drop_columns)
```

Đếm và in ra số lượng giá trị khác biệt trong mỗi thuộc tính của tập dữ liệu. Kết quả cho thấy số lượng giá trị khác biệt trong từng cột, từ đó cung cấp cái nhìn tổng quan về độ đa dạng của dữ liệu trong mỗi cột.

#### Check unique values in each columns

```
[14]: for column in df.columns:
    num_distinct_values = len(df[column].unique())
    print(f"{column}: {num_distinct_values} distinct values")

Category: 3 distinct values
City: 3636 distinct values
Country: 147 distinct values
Market: 4 distinct values
Order_Date: 1429 distinct values
Ship_Date: 1463 distinct values
Order_Priority: 4 distinct values
Region: 13 distinct values
Segment: 3 distinct values
Ship_Mode: 4 distinct values
State: 1086 distinct values
Sub_Category: 17 distinct values
Discount: 27 distinct values
Profit: 24575 distinct values
Quantity: 14 distinct values
Sales: 22995 distinct values
Shipping_Cost: 16877 distinct values
Profit_Positive: 2 distinct values
```

### 5.4.3 Chuyển đổi kiểu dữ liệu

Chuyển đổi kiểu dữ liệu của thuộc tính “Sales” sang kiểu số nguyên và chuyển đổi các cột **Order\_Date** và **Ship\_Date** sang kiểu ngày tháng. Sau đó, nó trích xuất các thành phần ngày tháng như ngày, ngày trong tuần, tháng, quý, và năm từ các cột này và lưu vào các thuộc tính mới. Cuối cùng, nó loại bỏ các cột **Order\_Date** và **Ship\_Date**.

### Data Type Conversion

```
[15]: df['Sales'] = df['Sales'].astype(int)

# Change the type of 'Order_Date' and 'Ship_Date'
df['Order_Date'] = pd.to_datetime(df['Order_Date'])
df['Ship_Date'] = pd.to_datetime(df['Ship_Date'])

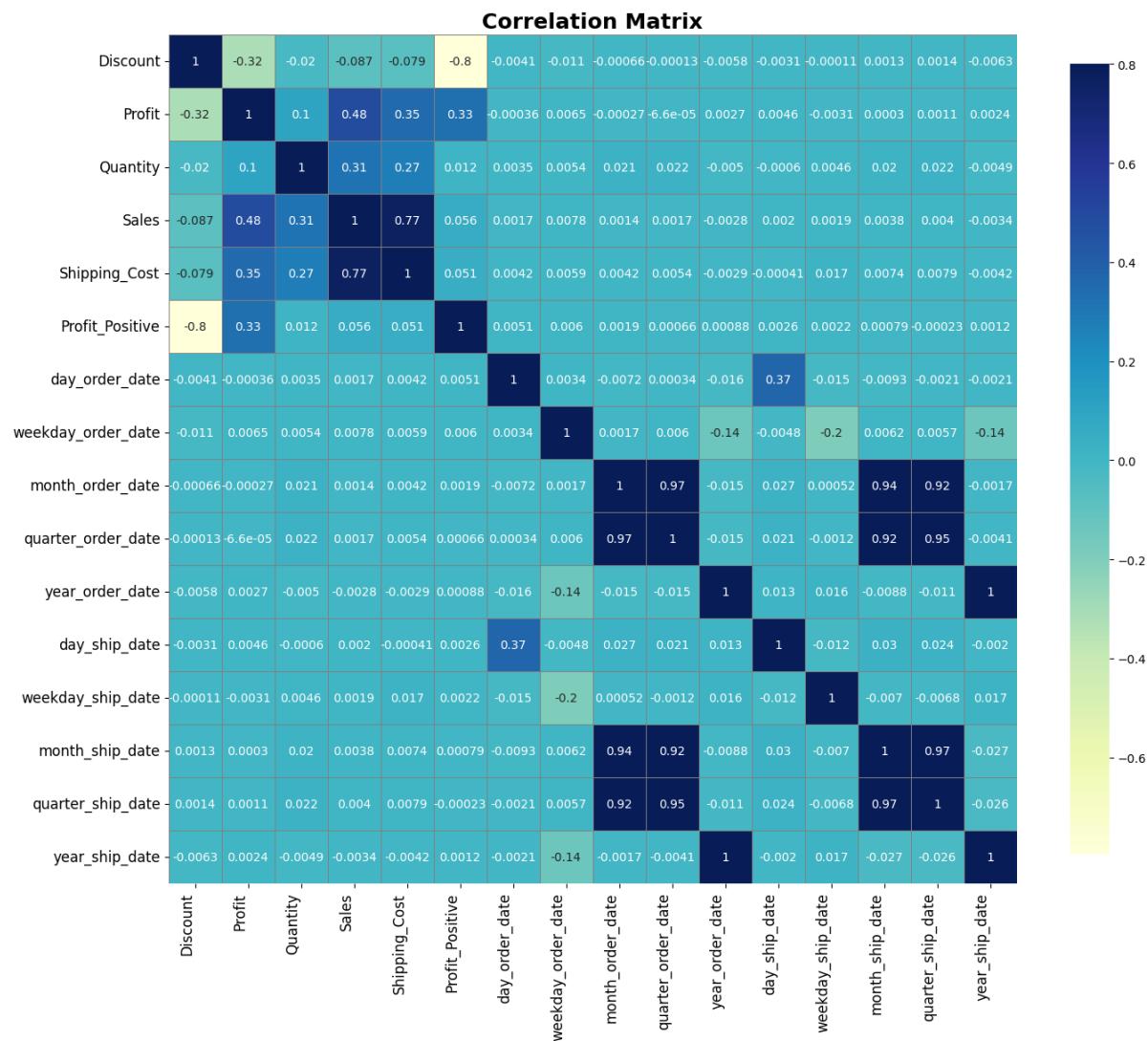
# Extract day, weekday, month, quarter, year from Order_Date
df['day_order_date'] = df['Order_Date'].dt.day
df['weekday_order_date'] = df['Order_Date'].dt.weekday
df['month_order_date'] = df['Order_Date'].dt.month
df['quarter_order_date'] = df['Order_Date'].dt.quarter
df['year_order_date'] = df['Order_Date'].dt.year

# Extract day, weekday, month, quarter, year from Ship_Date
df['day_ship_date'] = df['Ship_Date'].dt.day
df['weekday_ship_date'] = df['Ship_Date'].dt.weekday
df['month_ship_date'] = df['Ship_Date'].dt.month
df['quarter_ship_date'] = df['Ship_Date'].dt.quarter
df['year_ship_date'] = df['Ship_Date'].dt.year

# Drop column Order_Date and Ship_Date
df = df.drop(columns=['Order_Date', 'Ship_Date'])
```

#### 5.4.4 Phân tích tương quan

Thực hiện phân tích tương quan giữa các thuộc tính (Correlation Analysis) bằng cách tạo ma trận tương quan giữa các cột số trong df và hiển thị nó dưới dạng biểu đồ heatmap



Hình 5.2: Ma trận tương quan giữa các thuộc tính

- **Discount** và **Profit\_Positive**: Có một mối tương quan quan âm mạnh (-0.796437), cho thấy mối liên hệ nghịch giữa tỷ lệ giảm giá và việc có lợi nhuận dương. Điều này có nghĩa là khi giảm giá cao hơn, khả năng có lợi nhuận dương sẽ giảm đi.
- **Profit** và **Sales**: Mối tương quan dương mạnh (0.484911) giữa lợi nhuận và doanh số bán hàng. Điều này ngũ ý rằng khi doanh số tăng, có xu hướng lợi nhuận cũng tăng.
- **Profit** và **Shipping\_Cost**: Mối tương quan dương (0.354441) cũng khá mạnh giữa lợi nhuận và chi phí vận chuyển.
- Các thành phần ngày tháng của ngày vận chuyển và ngày đặt hàng cũng có mối tương quan cao với nhau

- **Discount** và **Shipping\_Cost, Sales**: Discount có mối tương quan âm với Shipping\_Cost và Sales (-0.079055 và -0.086715 lần lượt), tuy nhiên, mối tương quan này không quá mạnh.
- **Profit\_Positive** và các biến số khác: Mối tương quan của Profit\_Positive với các biến số khác (ngoài Discount) thường rất thấp, cho thấy sự độc lập của biến số này với các biến số khác trong ma trận.

#### 5.4.5 Xử lý thuộc tính danh mục

Xử lý các thuộc tính thuộc loại danh mục: **Order\_Priority, Ship\_Mode, Category, Sub\_Category, City, Country, State, Region, Market, Segment**

##### \* *Order\_Priority* và *Ship\_Mode*

Chuyển đổi các cột phân loại Order\_Priority và Ship\_Mode thành các giá trị số để thuận tiện cho việc phân tích và dự đoán. Order\_Priority được chuyển đổi dựa trên mức độ ưu tiên (Low: 1, Medium: 2, High: 3, Critical: 4), thuộc tính Ship\_Mode được chuyển đổi dựa trên phương thức vận chuyển (Standard Class: 1, Second Class: 2, First Class: 3, Same Day: 4).

```
[21]: priority_mapping = {
    'Low': 1,
    'Medium': 2,
    'High': 3,
    'Critical': 4
}

ship_mode_mapping = {
    'Standard Class': 1,
    'Second Class': 2,
    'First Class': 3,
    'Same Day': 4
}

df['Order_Priority_Numeric'] = df['Order_Priority'].map(priority_mapping)
df['Ship_Mode_Numeric'] = df['Ship_Mode'].map(ship_mode_mapping)
```

##### \* *Category, Sub\_Category, City, Country, State, Region, Market, Segment*

Đối với các thuộc tính trên ta sẽ không chọn cách xử lý như hai thuộc tính Order\_Priority và Ship\_Mode vì số lượng giá trị của mỗi thuộc tính quá nhiều dẫn đến sẽ không mang lại nhiều lợi ích cho việc dự đoán nên ta sẽ thực hiện định nghĩa hàm

**calculate\_profit\_score** tính điểm lợi nhuận cho các giá trị trong cột phân loại dựa trên lợi nhuận trung bình. Các giá trị lợi nhuận được chia thành 5 mức như sau:

- Score 4: Very High Profit (Lợi nhuận rất cao)
- Score 3: High Profit (Lợi nhuận cao)
- Score 2: Average Profit (Lợi nhuận trung bình)
- Score 1: Low Profit (Lợi nhuận thấp)
- Score 0: Loss (Lợi nhuận âm)

Các cột mới với hậu tố `_Score` sẽ được thêm vào tập dữ liệu, chứa điểm lợi nhuận tương ứng cho từng giá trị trong các cột phân loại trên.

```
[31]: def calculate_profit_score(df, column, target_column='Profit'):  
    """  
    Score the values in the categorical column based on average return.  
  
    Args:  
    df (DataFrame): Original data.  
    column (str): Name of the classification column to calculate score.  
    target_column (str): Target column name (Profit).  
  
    Returns:  
    dict: Mapping from values in categorical columns to scores.  
    """  
  
    # Calculate the average profit for each value in the categorical column  
    mean_profit = df.groupby(column)[target_column].mean()  
  
    # Calculate the percentiles of the average return  
    quantiles = mean_profit.quantile([0.25, 0.5, 0.75])  
  
    # Function assigns scores based on percentiles  
    def assign_score(profit):  
        if profit > quantiles[0.75]:  
            return 4  
        elif profit > quantiles[0.5]:  
            return 3  
        elif profit > quantiles[0.25]:  
            return 2  
        elif profit > 0:  
            return 1  
        else:  
            return 0  
  
    # Create mapping from value to point  
    score_mapping = mean_profit.apply(assign_score).to_dict()  
  
    return score_mapping
```

Thực hiện mapping giá trị.

```
[32]: # Apply scoring to classification columns and assign scores
for col in categorical_columns:
    score_mapping = calculate_profit_score(df, col)
    df[f'{col}_Score'] = df[col].map(score_mapping)
```

### 5.4.6 Tìm 4 thuộc tính quan trọng

Chia dữ liệu thành hai phần: biến độc lập (X) và biến mục tiêu (y) để phục vụ cho việc tìm các thuộc tính quan trọng, xóa các thuộc tính danh mục đã xử lý ở bước trước khỏi tập X.

```
[33]: from sklearn.model_selection import train_test_split

drop_columns = ['Profit', 'Ship_Mode', 'Order_Priority', 'Category', 'Sub_Category', 'City', 'Country', 'State', 'Region', 'Market', 'Segment']

df_copy = df.copy()

df_copy = df_copy.drop(columns=drop_columns)

X = df_copy.drop(columns=['Profit_Positive'])
y = df_copy['Profit_Positive']

# Split data into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

#### 5.4.6.1 Sử dụng SelectKBest với f\_classif

Using SelectKBest

```
[34]: from sklearn.feature_selection import SelectKBest, f_classif

[35]: # Use SelectKBest to select the 4 most important attributes
selector = SelectKBest(score_func=f_classif, k=4)
X_new = selector.fit_transform(X_train, y_train)

# Get the names of the most important attributes
selected_features = X_train.columns[selector.get_support()]

print("The 4 most important features:", selected_features)

The 4 most important features: Index(['Discount', 'City_Score', 'Country_Score', 'State_Score'], dtype='object')
```

Sử dụng phương pháp SelectKBest với hàm điểm f\_classif, chúng ta đã xác định được 4 thuộc tính quan trọng nhất cho việc dự đoán Profit\_Positive. Các thuộc tính này là Discount, City\_Score, Country\_Score, và State\_Score.

#### 5.4.6.2 Sử dụng thuật toán Random Forest

```
[37]: # Create the model pipeline
model = RandomForestClassifier(random_state=42)

# Train the model
model.fit(X, y)

# Extract feature importances
feature_importances = model.feature_importances_

# Get the feature names
feature_names = X.columns

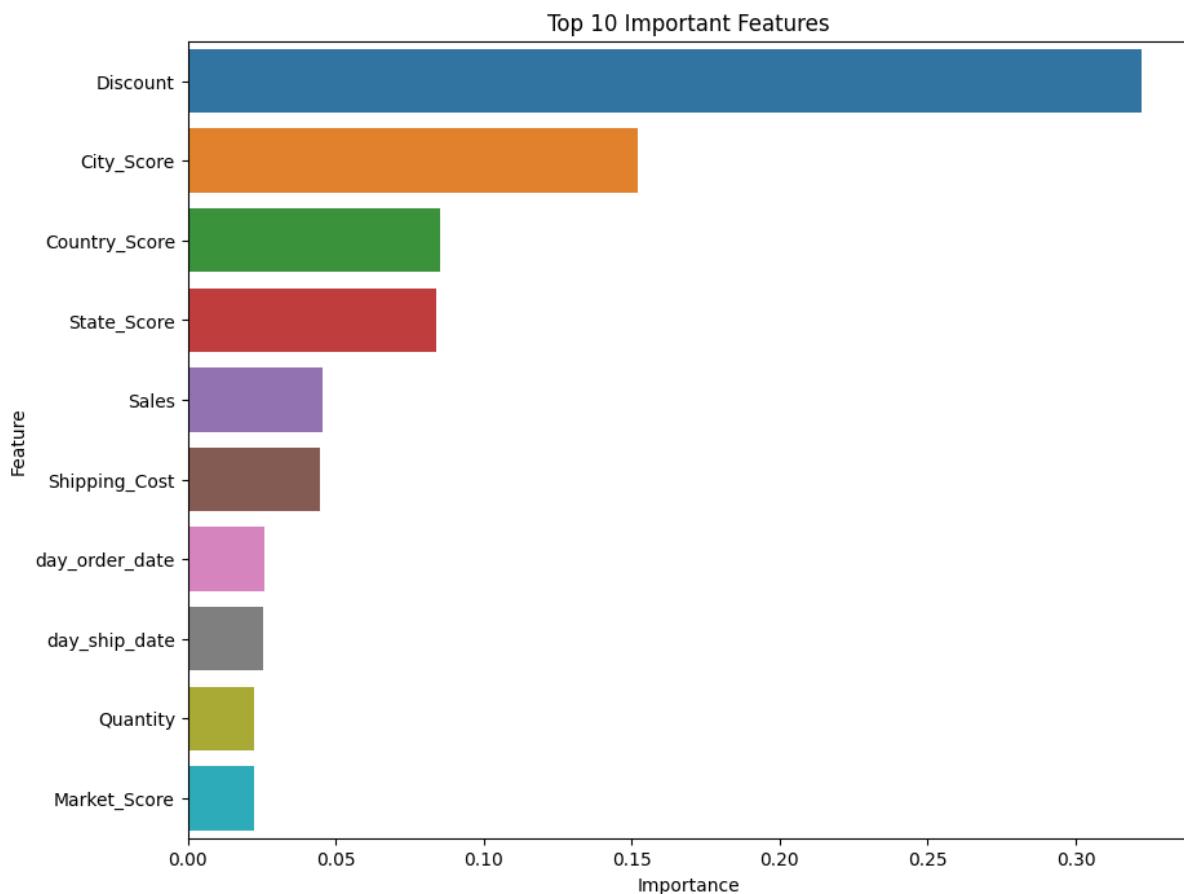
# Create a DataFrame for feature importances
feature_importances_df = pd.DataFrame({'Feature': feature_names, 'Importance': feature_importances})

# Sort the features by importance
feature_importances_df = feature_importances_df.sort_values(by='Importance', ascending=False)

# Visualize the top features
plt.figure(figsize=(10, 8))
sns.barplot(x='Importance', y='Feature', data=feature_importances_df.head(10))
plt.title('Top 10 Important Features')
plt.show()

# Print the top 4 important features
print("Top 4 Important Features:")
print(feature_importances_df.head(4))
```

Kết quả:



Sử dụng mô hình Random Forest, chúng ta đã xác định được 4 thuộc tính quan trọng nhất để dự đoán Profit\_Positive. Các thuộc tính này là Discount, City\_Score, Country\_Score, và State\_Score.

=> **Kết luận:** Sử dụng cả hai phương pháp (SelectKBest và Random Forest) đều chỉ ra rằng các thuộc tính **Discount, City\_Score, Country\_Score, và State\_Score** đều được xác định là 4 thuộc tính quan trọng nhất trong cả hai phương pháp, cho thấy rằng chúng có ảnh hưởng đáng kể đến việc dự đoán **Profit\_Positive**. Và qua đó ta sẽ sử dụng 4 thuộc tính này để đưa vào các mô hình dưới đây để dự đoán biến mục tiêu **Profit\_Positive**

## 5.5 Huấn luyện mô hình

### 5.5.1 Giới thiệu phương pháp GridSearchCV

GridSearchCV là một kỹ thuật tìm kiếm tham số tốt nhất cho mô hình máy học bằng cách thực hiện một tìm kiếm toàn diện trên một lưới (grid) các tham số được chỉ định. Nó tự động hóa quá trình tìm kiếm tham số tối ưu bằng cách kiểm tra tất cả các kết hợp có thể của các tham số và chọn ra sự kết hợp mang lại hiệu suất tốt nhất cho mô hình.

Lợi ích của GridSearchCV:

1. Tự động hóa quá trình tìm kiếm tham số: Thay vì thử từng tham số một cách thủ công, GridSearchCV kiểm tra tất cả các kết hợp có thể của các tham số.
2. Đảm bảo tìm được tham số tối ưu: GridSearchCV giúp tìm ra sự kết hợp tốt nhất của các tham số mang lại hiệu suất cao nhất cho mô hình.
3. Sử dụng cross-validation: GridSearchCV sử dụng k-fold cross-validation để đánh giá hiệu suất của mỗi bộ tham số, đảm bảo rằng kết quả không bị lệch và mô hình có khả năng tổng quát hóa tốt.

Cách Thức Hoạt Động:

1. Xác định lưới tham số (Parameter Grid): Người dùng cung cấp một tập hợp các tham số và các giá trị tương ứng mà họ muốn thử nghiệm.

2. Tạo GridSearchCV: GridSearchCV sẽ tạo ra tất cả các kết hợp có thể của các tham số trong lưới.
3. Thực hiện Cross-Validation: Mỗi bộ tham số được đánh giá bằng cách sử dụng k-fold cross-validation trên tập dữ liệu huấn luyện.
4. Chọn tham số tốt nhất: Bộ tham số có hiệu suất tốt nhất (thường dựa trên một thước đo như độ chính xác) sẽ được chọn.



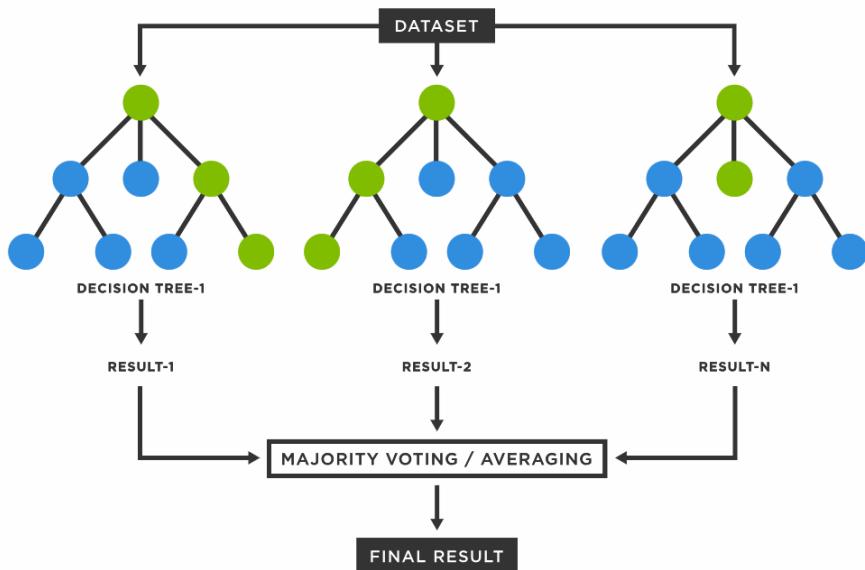
Hình 5.3: Phương pháp GridSearchCV

## 5.5.2 Thuật toán Random Forest

### 5.5.2.1 Giới thiệu

Thuật toán Rừng Ngẫu Nhiên (Random Forest) là một kỹ thuật học máy mạnh mẽ dựa trên cây quyết định. Nó hoạt động bằng cách tạo ra một số lượng lớn cây quyết định trong giai đoạn huấn luyện. Mỗi cây được xây dựng bằng cách sử dụng một tập con ngẫu nhiên của tập dữ liệu để đo lường một tập con ngẫu nhiên của các đặc trưng trong mỗi phần. Sự ngẫu nhiên này giới thiệu sự biến đổi giữa các cây đơn lẻ, giảm nguy cơ overfitting và cải thiện hiệu suất dự đoán tổng thể. Trong giai đoạn dự đoán, thuật toán kết hợp kết quả của tất cả các cây, bằng cách bỏ phiếu (cho các bài toán phân loại) hoặc tính trung bình (cho các bài toán hồi quy). Quá trình quyết định hợp tác này, được hỗ trợ bởi nhiều cây với những hiểu biết của chúng, cung cấp các kết quả ổn định và chính xác. Rừng ngẫu nhiên được sử dụng rộng rãi cho các chức năng

phân loại và hồi quy, nổi tiếng với khả năng xử lý dữ liệu phức tạp, giảm overfitting và cung cấp dự báo đáng tin cậy trong các môi trường khác nhau. [6]



Hình 5.4: Thuật toán Rừng Ngẫu Nhiên (Random Forest)

### 5.5.2.2 Triển khai

Code triển khai thuật toán:

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report

# Define the parameter grid for Random Forest
param_grid_rf = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30]
}

# Create a Grid Search CV for Random Forest
grid_search_rf = GridSearchCV(
    estimator=RandomForestClassifier(random_state=42),
    param_grid=param_grid_rf,
    cv=5,
    scoring='accuracy',
    n_jobs=-1
)

# Train the model
grid_search_rf.fit(X_train, y_train)
  
```

```
# Get the best model
best_rf = grid_search_rf.best_estimator_

# Make predictions
y_pred_rf = best_rf.predict(X_test)

# Evaluate the model
accuracy_rf = accuracy_score(y_test, y_pred_rf)
report_rf = classification_report(y_test, y_pred_rf)

print(f'Best Parameters for Random Forest:
{grid_search_rf.best_params_}')
print(f'Accuracy for Random Forest: {accuracy_rf}')
print(f'Classification Report for Random Forest:\n{report_rf}')
```

## 1. Import các thư viện cần thiết:

- RandomForestClassifier từ sklearn.ensemble để xây dựng mô hình Random Forest.
- GridSearchCV từ sklearn.model\_selection để tìm kiếm siêu tham số tốt nhất.
- accuracy\_score và classification\_report từ sklearn.metrics để đánh giá mô hình.

## 2. Định nghĩa lưới tham số cho Random Forest:

- **n\_estimators**: số lượng cây
- **max\_depth**: độ sâu tối đa của cây

## 3. GridSearchCV được khởi tạo với các tham số như sau:

- estimator=RandomForestClassifier(random\_state=42): Mô hình sử dụng là RandomForestClassifier với random\_state=42 để đảm bảo kết quả có thể tái sản sinh.
- param\_grid=param\_grid\_rf: Lưới các tham số cần tối ưu hóa.
- cv=5: Sử dụng phương pháp cross-validation với 5 fold.
- scoring='accuracy': Đánh giá hiệu suất bằng độ chính xác.

## 4. Huấn luyện mô hình: Gọi phương thức .fit(X\_train, y\_train) trên grid\_search\_rf để huấn luyện mô hình trên tập dữ liệu huấn luyện (X\_train, y\_train).

## 5. Lấy mô hình tốt nhất: best\_rf = grid\_search\_rf.best\_estimator\_: Lấy mô hình tốt nhất từ kết quả tối ưu hóa.

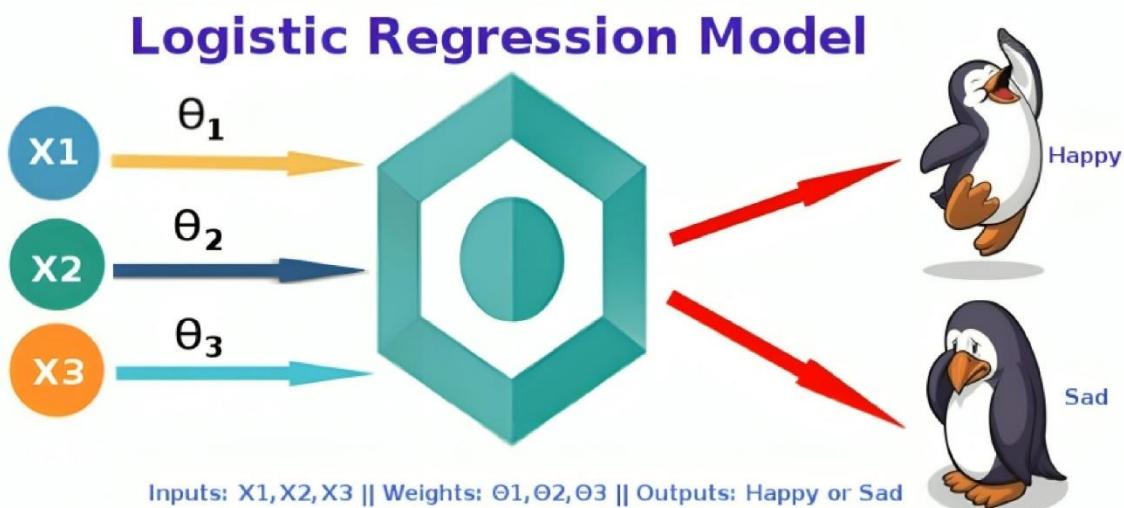
Ta được các parameters tốt nhất cho mô hình Random Forest như sau:

```
{'max_depth': 10, 'n_estimators': 50}
```

### 5.5.3 Thuật toán Logistic Regression

#### 5.5.3.1 Giới thiệu

**Hồi quy Logistic** là một phương pháp phân tích hồi quy phù hợp khi biến phụ thuộc là nhị phân (binary). Giống như tất cả các phân tích hồi quy, hồi quy logistic là một phân tích dự đoán. Nó được sử dụng để mô tả dữ liệu và giải thích mối quan hệ giữa một biến phụ thuộc nhị phân và một hoặc nhiều biến độc lập ở mức độ danh nghĩa, thứ tự, khoảng hoặc tỷ lệ. Nó là một phương pháp phân tích thống kê được học máy mượn để sử dụng. Nó được dùng khi biến phụ thuộc của chúng ta là nhị phân hoặc có hai giá trị. Điều này có nghĩa là biến chỉ có hai kết quả, ví dụ như một người sẽ sống sót sau tai nạn hay không, học sinh sẽ đậu kỳ thi hay không. Kết quả có thể là có hoặc không (hai đầu ra). Kỹ thuật hồi quy này tương tự như hồi quy tuyến tính và có thể được sử dụng để dự đoán xác suất cho các vấn đề phân loại. [7]



Hình 5.5: Thuật toán Logistic Regression

Các loại Hồi quy Logistic:

- **Binary Logistic Regression:** Dùng để dự đoán xác suất của một kết quả nhị phân, chẳng hạn như có hoặc không, đúng hoặc sai, hoặc 0 hoặc 1. Ví dụ, nó có thể được sử dụng để dự đoán liệu khách hàng sẽ rời bỏ dịch vụ hay không, bệnh nhân có mắc bệnh hay không, hoặc khoản vay có được hoàn trả hay không.

- **Multinomial Logistic Regression:** Dùng để dự đoán xác suất của một trong ba hoặc nhiều kết quả có thể xảy ra, chẳng hạn như loại sản phẩm khách hàng sẽ mua, đánh giá mà khách hàng sẽ đưa ra cho sản phẩm, hoặc đảng phái chính trị mà một người sẽ bầu chọn.
- **Ordinal Logistic Regression:** Dùng để dự đoán xác suất của một kết quả rơi vào một trật tự đã được xác định trước, chẳng hạn như mức độ hài lòng của khách hàng, mức độ nghiêm trọng của một căn bệnh, hoặc giai đoạn của bệnh ung thư.

#### 5.5.3.2 Triển khai

Code triển khai thuật toán:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report

# Define the parameter grid for Logistic Regression
param_grid_lr = [
    {'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000], 'solver': ['liblinear'], 'penalty': ['l1', 'l2'], 'class_weight': [None, 'balanced']},
    {'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000], 'solver': ['lbfgs', 'newton-cg', 'sag'], 'penalty': ['l2'], 'class_weight': [None, 'balanced']},
    {'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000], 'solver': ['saga'], 'penalty': ['l1', 'l2', 'elasticnet'], 'class_weight': [None, 'balanced'], 'l1_ratio': [0.0, 0.5, 1.0] }
]

# Create a Grid Search CV for Logistic Regression
grid_search_lr = GridSearchCV(
    estimator=LogisticRegression(max_iter=200, random_state=42),
    param_grid=param_grid_lr,
    cv=5,
    scoring='accuracy',
    n_jobs=-1
)

# Train the model
grid_search_lr.fit(X_train, y_train)

# Get the best model
best_lr = grid_search_lr.best_estimator_

# Make predictions
```

```
y_pred_lr = best_lr.predict(X_test)

# Evaluate the model
accuracy_lr = accuracy_score(y_test, y_pred_lr)
report_lr = classification_report(y_test, y_pred_lr)

print(f'Best Parameters for Logistic Regression:
{grid_search_lr.best_params_}')
print(f'Accuracy for Logistic Regression: {accuracy_lr}')
print(f'Classification Report for Logistic Regression:\n{report_lr}'')
```

## 1. Import các thư viện cần thiết:

- LogisticRegression từ sklearn.linear\_model để xây dựng mô hình Logistic Regression.
- GridSearchCV từ sklearn.model\_selection để tìm kiếm siêu tham số tốt nhất.
- accuracy\_score và classification\_report từ sklearn.metrics để đánh giá mô hình.

## 2. Định nghĩa lưới tham số cho Logistic Regression:

- **C**: tham số nghịch đảo của độ giảm (regularization strength), với các giá trị [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000].
- **solver**: thuật toán tối ưu hóa mà mô hình sử dụng, bao gồm 'liblinear', 'lbfgs', 'newton-cg', 'sag', 'saga'.
- **penalty**: hình phạt được sử dụng để giảm overfitting, có thể là 'l1', 'l2', hoặc 'elasticnet'.
- **class\_weight**: cân bằng trọng số cho các lớp trong mô hình, bao gồm None hoặc 'balanced'.
- **l1\_ratio**: tỷ lệ của hình phạt L1 so với tổng hình phạt (chỉ có hiệu lực khi solver='saga' và penalty='elasticnet'), với các giá trị [0.0, 0.5, 1.0].

## 3. Tạo Grid Search CV:

- estimator=LogisticRegression(max\_iter=200, random\_state=42): sử dụng mô hình Logistic Regression với giới hạn số lần lặp max\_iter=200 và random\_state=42.
- param\_grid=param\_grid\_lr: sử dụng lưới tham số đã định nghĩa ở trên.
- cv=5: sử dụng cross-validation với 5 fold để đánh giá hiệu suất.
- scoring='accuracy': đánh giá hiệu suất của mô hình dựa trên độ chính xác.

## 4. Huấn luyện mô hình:

Gọi phương thức .fit(X\_train, y\_train) trên grid\_search\_lr để huấn luyện mô hình trên tập dữ liệu huấn luyện (X\_train, y\_train).

**5. Lấy mô hình tốt nhất:** `best_lr = grid_search_lr.best_estimator_` : lấy ra mô hình tối ưu nhất từ kết quả tối ưu hóa.

Ta được các parameters tốt nhất cho mô hình Logistic Regression như sau:

```
{'C': 1, 'class_weight': None, 'penalty': 'l1', 'solver': 'liblinear'}
```

## 5.5.4 Thuật toán Naive Bayes

### 5.5.4.1 Giới thiệu

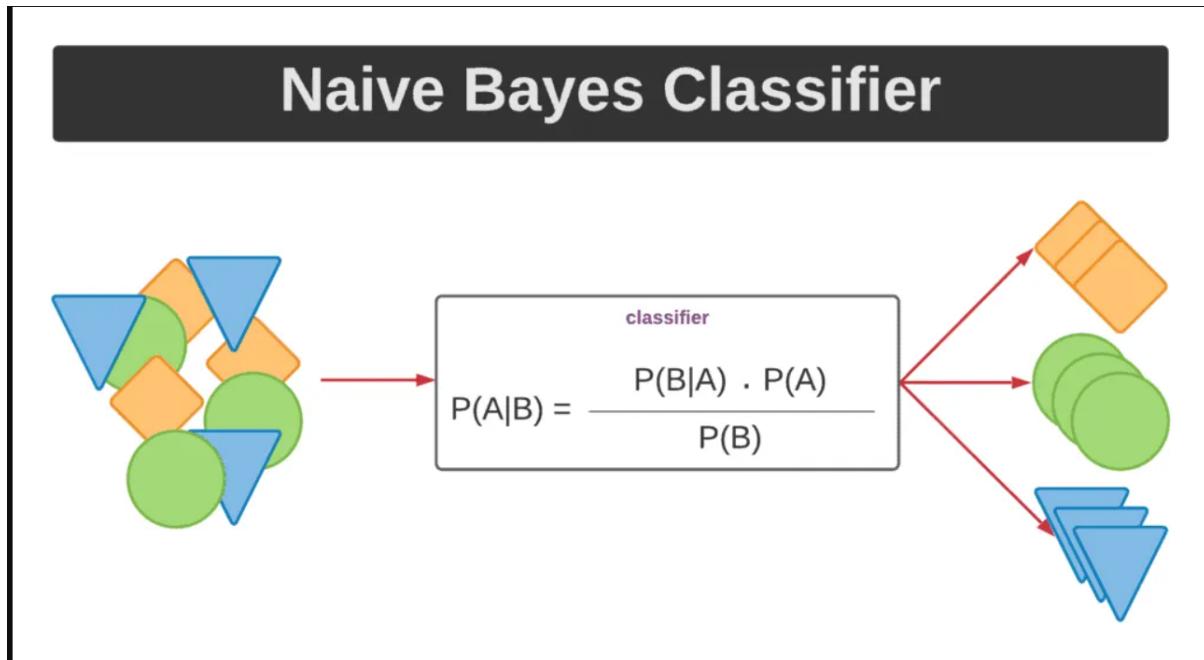
**Naïve Bayes** là một trong những thuật toán phân loại dữ liệu, thuộc top 10 thuật toán trong khai thác dữ liệu. **Naïve Bayes** là một thuật toán phân loại xác suất đơn giản, tính toán một tập hợp các xác suất bằng cách tính tần suất và sự kết hợp của các giá trị trong một tập dữ liệu cụ thể. Xác suất của các đặc trưng trong dữ liệu xuất hiện dưới dạng các thành viên trong một chuỗi xác suất và được tính bằng cách tính tần suất của mỗi giá trị đặc trưng trong lớp từ tập dữ liệu huấn luyện. Tập dữ liệu huấn luyện là một tập con được sử dụng để huấn luyện các thuật toán phân loại. Quá trình huấn luyện sử dụng các giá trị đã biết để dự đoán các giá trị chưa biết. [8]

Công thức tổng quát của **Naïve Bayes** được xác định dựa trên định lý Bayes. Giả sử chúng ta có một tập dữ liệu huấn luyện với  $N$  mẫu và  $K$  lớp khác nhau, công thức tổng quát của Naïve Bayes để tính xác suất hậu nghiệm của một lớp cụ thể cho một mẫu dữ liệu mới được biểu diễn như sau:

$$P(C|X) = \frac{P(C) \times P(X|C)}{P(X)}$$

Trong đó :

- $P(C | X)$  là xác suất hậu nghiệm của lớp  $C$  cho mẫu dữ liệu  $X$ .
- $P(C)$  là xác suất tiên nghiệm của lớp  $C$ , đại diện cho xác suất của lớp  $C$  trong tập dữ liệu huấn luyện.
- $P(X | C)$  là xác suất có điều kiện của mẫu dữ liệu  $X$  với lớp  $C$ . Trong Naïve Bayes, giả định rằng các đặc trưng là độc lập, do đó xác suất có điều kiện có thể được tính bằng cách nhân các xác suất của từng đặc trưng.
- $P(X)$  là xác suất của mẫu dữ liệu  $X$ .



Hình 5.6: Thuật toán Naïve Bayes

#### 5.5.4.2 Triển khai

Code triển khai thuật toán:

```
from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report

# Define the parameter grid for GridSearchCV
param_grid_nb = {
    'var_smoothing': [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2,
1e-1, 1.0]
}

# Initialize the Naive Bayes classifier
nb_classifier = GaussianNB()

# Initialize GridSearchCV with the Naive Bayes classifier and parameter
# grid
grid_search_nb = GridSearchCV(estimator=nb_classifier,
param_grid=param_grid_nb, cv=5, scoring='accuracy')

# Train the classifier on the training data
grid_search_nb.fit(X_train, y_train)

# Get the best model
best_nb = grid_search_nb.best_estimator_
```

```
# Make predictions on the test data using the best model
y_pred_nb = best_nb.predict(X_test)

# Evaluate the performance of the classifier
accuracy_nb = accuracy_score(y_test, y_pred_nb)
classification_report_nb = classification_report(y_test, y_pred_nb)

print(f"Best Naive Bayes parameters: {grid_search_nb.best_params_}")
print(f"Naive Bayes Classifier Accuracy: {accuracy_nb}")
print(f"Classification Report:\n{classification_report_nb}")
```

## 1. Import các thư viện cần thiết:

GridSearchCV từ sklearn.model\_selection để tìm kiếm siêu tham số tốt nhất.

GaussianNB từ sklearn.naive\_bayes để sử dụng mô hình Naive Bayes.

accuracy\_score và classification\_report từ sklearn.metrics để đánh giá hiệu suất của mô hình.

## 2. Định nghĩa lưới tham số cho GridSearchCV:

- **var\_smoothing**: một phần tử quan trọng trong phương pháp ước lượng dựa vào Gaussian Naive Bayes. Các giá trị được thử nghiệm là [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1.0].

## 3. Khởi tạo GridSearchCV:

- estimator=nb\_classifier: sử dụng mô hình Naive Bayes đã khởi tạo.
- param\_grid=param\_grid\_nb: sử dụng lưới tham số đã định nghĩa ở trên.
- cv=5: sử dụng cross-validation với 5 fold để đánh giá hiệu suất.
- scoring='accuracy': đánh giá hiệu suất của mô hình dựa trên độ chính xác.

**4. Huấn luyện mô hình:** Gọi phương thức .fit(X\_train, y\_train) trên grid\_search\_nb để huấn luyện mô hình Naive Bayes trên tập dữ liệu huấn luyện (X\_train, y\_train).

**5. Lấy mô hình tối ưu nhất:** best\_nb = grid\_search\_nb.best\_estimator\_: lấy ra mô hình Naive Bayes tối ưu nhất từ kết quả tối ưu hóa.

Ta được các parameters tốt nhất cho mô hình Naïve Bayes như sau:

```
{'var_smoothing': 1e-09}
```

## 5.5.5 Thuật toán Perceptron

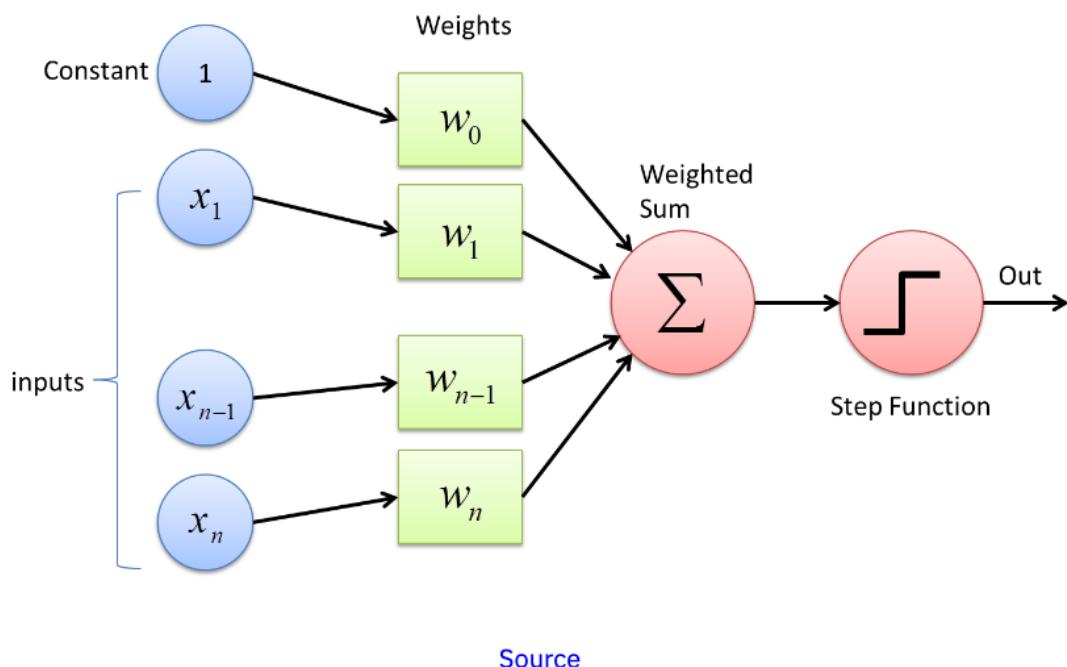
### 5.5.5.1 Giới thiệu

Perceptron là một trong những thuật toán học máy cơ bản và đầu tiên, được phát triển bởi Frank Rosenblatt vào năm 1958. Đây là một thuật toán học có giám sát được sử dụng chủ yếu cho các bài toán phân loại nhị phân, nơi mà mục tiêu là phân loại dữ liệu vào một trong hai lớp. [9]

Thuật toán Perceptron mô phỏng hoạt động của một nơ-ron thần kinh trong não, với các đầu vào (inputs), trọng số (weights), và một hàm kích hoạt (activation function).

Mô hình Perceptron đơn giản bao gồm:

- **Đầu vào (Input Layer):** Tập hợp các đặc trưng của dữ liệu được đưa vào mô hình.
- **Trọng số (Weights):** Mỗi đặc trưng được gán một trọng số tương ứng.
- **Hàm kích hoạt (Activation Function):** Tổng các đầu vào có trọng số được tính toán và đưa qua hàm kích hoạt để đưa ra quyết định phân loại.



Hình 5.7: Thuật toán Perceptron

Perceptron là nền tảng cho nhiều mô hình học máy phức tạp hơn và đã đặt nền móng cho sự phát triển của các mạng nơ-ron nhân tạo (Artificial Neural Networks). Mặc dù đơn giản, Perceptron đã mở ra một kỷ nguyên mới trong nghiên cứu về trí tuệ nhân tạo và học máy, dẫn đến sự phát triển của nhiều thuật toán và kỹ thuật tiên tiến trong lĩnh vực này.

#### 5.5.5.2 Triển khai

Code triển khai thuật toán:

```
from sklearn.linear_model import Perceptron
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report

# Define the parameter grid for the Perceptron
param_grid_perceptron = {
    'penalty': [None, 'l2', 'l1', 'elasticnet'],
    'alpha': [0.00001, 0.0001, 0.001, 0.01, 0.1], # alpha values
    'max_iter': [1000, 2000, 3000, 4000], # iterations
    'tol': [1e-3, 1e-4, 1e-5, 1e-6], # stricter tolerance values
    'eta0': [1.0, 0.5, 0.1, 0.05, 0.01, 0.005] # learning rates
}

# Create Grid Search CV for Perceptron
grid_search_perceptron = GridSearchCV(
    estimator=Perceptron(random_state=42),
    param_grid=param_grid_perceptron,
    cv=5,
    scoring='accuracy',
    n_jobs=-1
)

# Train the Perceptron model
grid_search_perceptron.fit(X_train, y_train)

# Get the best Perceptron model
best_perceptron = grid_search_perceptron.best_estimator_

# Prediction with the best Perceptron model
y_pred_perceptron = best_perceptron.predict(X_test)

# Evaluate the Perceptron model
accuracy_perceptron = accuracy_score(y_test, y_pred_perceptron)
report_perceptron = classification_report(y_test, y_pred_perceptron)
```

```
print(f'Best Parameters for Perceptron:\n{grid_search_perceptron.best_params_}')
print(f'Accuracy for Perceptron: {accuracy_perceptron}')
print(f'Classification Report for Perceptron:\n{report_perceptron}'
```

## 1. Import các thư viện cần thiết:

- Perceptron từ sklearn.linear\_model để xây dựng mô hình Perceptron.
- GridSearchCV từ sklearn.model\_selection để tìm kiếm siêu tham số tốt nhất.
- accuracy\_score và classification\_report từ sklearn.metrics để đánh giá hiệu suất của mô hình.

## 2. Định nghĩa lưới tham số cho Perceptron:

- **penalty**: hình phạt được áp dụng cho hàm mất mát, có thể là None, 'l2', 'l1', hoặc 'elasticnet'.
- **alpha**: hệ số học (learning rate) của hàm mất mát, với các giá trị [0.00001, 0.0001, 0.001, 0.01, 0.1].
- **max\_iter**: số lần lặp tối đa để huấn luyện mô hình, với các giá trị [1000, 2000, 3000, 4000].
- **tol**: ngưỡng chấp nhận được để dừng quá trình huấn luyện, với các giá trị [1e-3, 1e-4, 1e-5, 1e-6].
- **eta0**: tỷ lệ học ban đầu, là một tham số quan trọng trong quá trình học của Perceptron, với các giá trị [1.0, 0.5, 0.1, 0.05, 0.01, 0.005].

## 3. Khởi tạo Grid Search CV cho Perceptron:

- estimator=Perceptron(random\_state=42): sử dụng mô hình Perceptron với random\_state=42 để có kết quả nhất quán.
- param\_grid=param\_grid\_perceptron: sử dụng lưới tham số đã định nghĩa ở trên.
- cv=5: sử dụng cross-validation với 5 fold để đánh giá hiệu suất.
- scoring='accuracy': đánh giá hiệu suất của mô hình dựa trên độ chính xác.

## 4. Huấn luyện mô hình Perceptron:

Gọi phương thức .fit(X\_train, y\_train) trên grid\_search\_perceptron để huấn luyện mô hình Perceptron trên tập dữ liệu huấn luyện (X\_train, y\_train).

## 5. Lấy mô hình Perceptron tốt nhất:

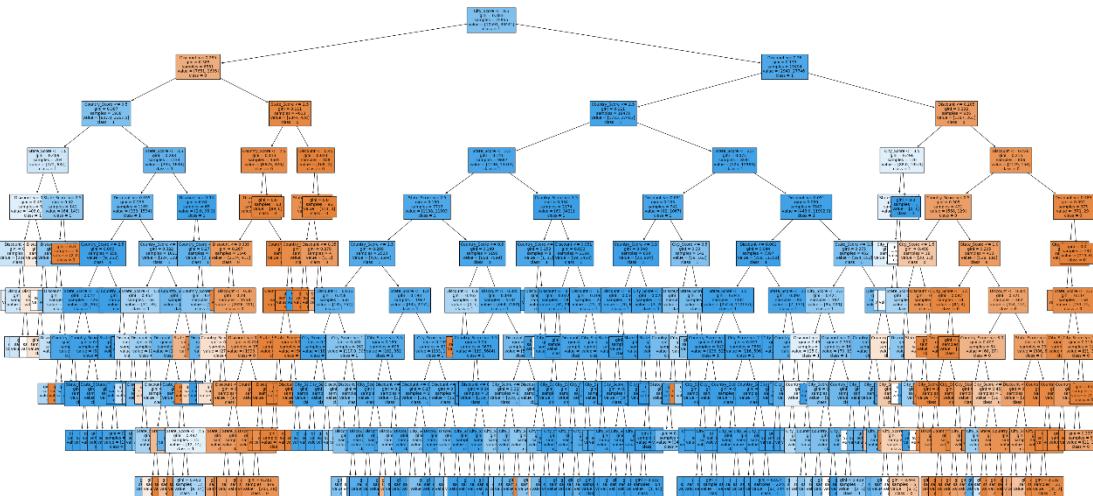
best\_perceptron = grid\_search\_perceptron.best\_estimator\_ : lấy ra mô hình Perceptron tối ưu nhất từ kết quả tối ưu hóa.

Ta được các parameters tốt nhất cho mô hình Perceptron như sau:

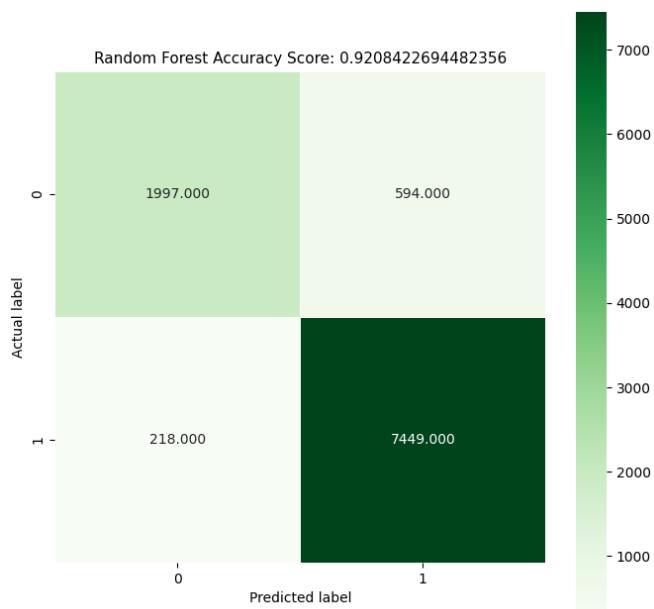
```
{'alpha': 1e-05, 'eta0': 0.005, 'max_iter': 1000, 'penalty': 'elasticnet', 'tol': 1e-05}
```

## 5.6 Đánh giá mô hình

### 5.6.1 Thuật toán Random Forest



Hình 5.8: Vẽ một trong những cây trong Rừng Ngẫu nhiên



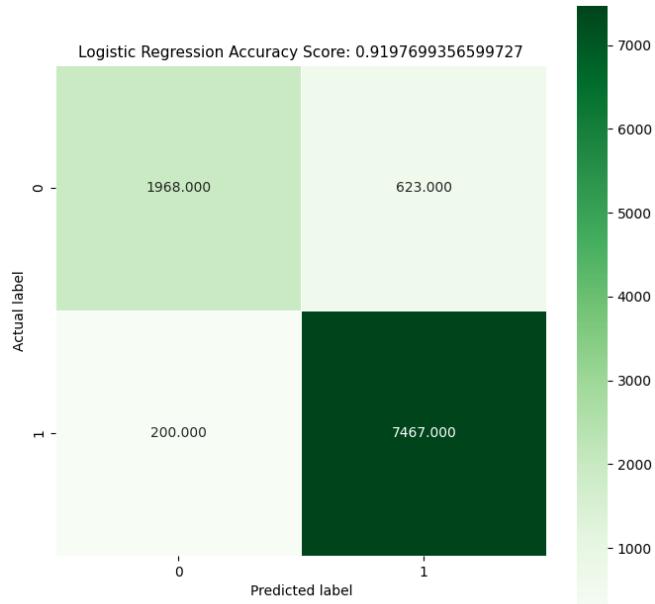
Hình 5.9: Ma trận nhầm lẫn thuật toán Random Forest

- **Accuracy:** 0.9208422694482356
- **Precision:** Class 0: 0.90 – Class 1: 0.93
- **Recall:** Class 0: 0.77 – Class 1: 0.97
- **F1-score:** Class 0: 0.83 – Class 1: 0.95

- **Support:** Class 0: 2591 – Class 1: 7667
- **Weighted Avg:** Precision: 0.92 – Recall: 0.92 – F1-score: 0.92

**Nhận xét:** Random Forest đạt được độ chính xác rất cao (92%) với sự cân bằng tốt giữa các lớp. Precision và Recall của lớp 1 rất tốt, cho thấy mô hình này hoạt động tốt trong việc phát hiện các mẫu của lớp 1. Tuy nhiên, recall của lớp 0 còn thấp, chỉ đạt 77%.

### 5.6.2 Thuật toán Logistic Regression

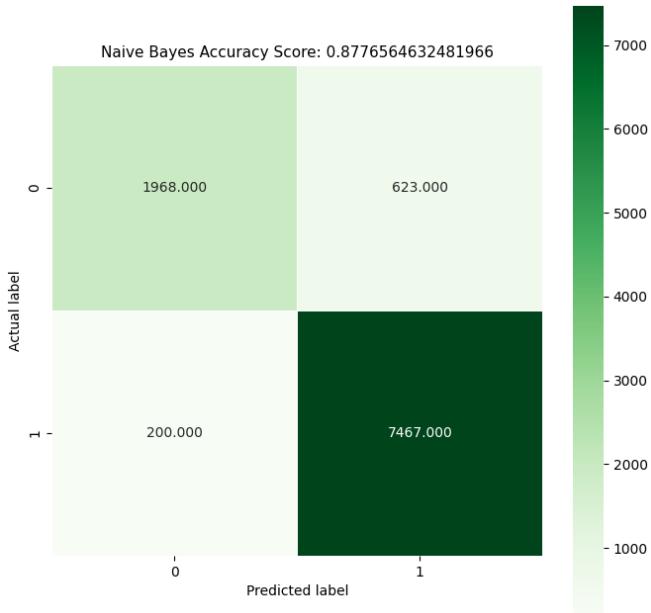


Hình 5.10: Ma trận nhầm lẫn thuật toán Logistic Regression

- **Accuracy:** 0.9197699356599727
- **Precision:** Class 0: 0.91 – Class 1: 0.92
- **Recall:** Class 0: 0.76 – Class 1: 0.97
- **F1-score:** Class 0: 0.83 – Class 1: 0.95
- **Support:** Class 0: 2591 – Class 1: 7667
- **Weighted Avg:** Precision: 0.92 – Recall: 0.92 – F1-score: 0.92

**Nhận xét:** Logistic Regression cũng đạt được độ chính xác cao (91.97%), gần bằng với Random Forest. Precision và Recall của lớp 1 rất tốt, nhưng recall của lớp 0 còn thấp (76%), điều này giống với mô hình Random Forest.

### 5.6.3 Thuật toán Naïve Bayes

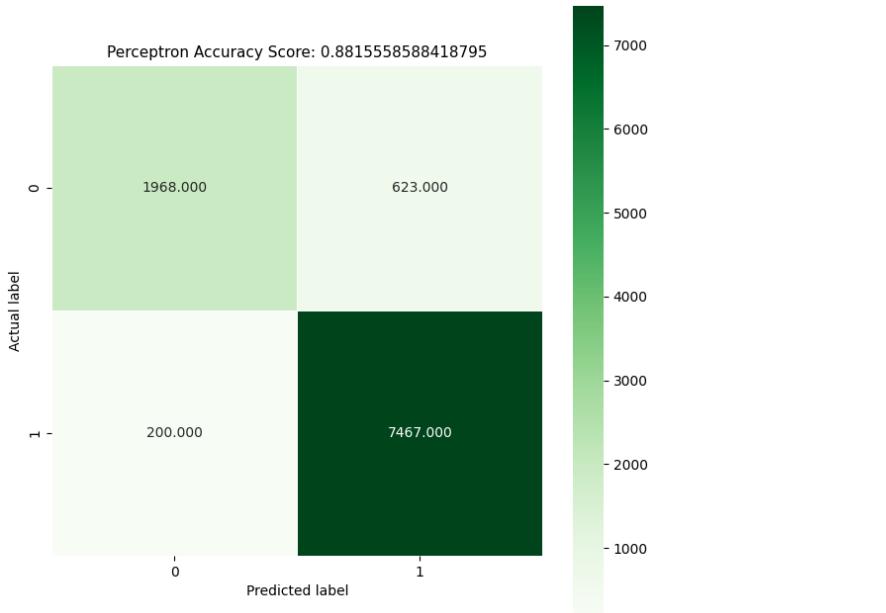


Hình 5.11: Ma trận nhầm lẫn thuật toán Naïve Bayes

- Accuracy: 0.8776564632481966
- Precision: Class 0: 0.73 – Class 1: 0.94
- Recall: Class 0: 0.83 – Class 1: 0.89
- F1-score: Class 0: 0.77 – Class 1: 0.92
- Support: Class 0: 2591 – Class 1: 7667
- Weighted Avg: Precision: 0.89 – Recall: 0.88 – F1-score: 0.88

**Nhận xét:** Naïve Bayes có độ chính xác thấp hơn (87.76%) so với Random Forest và Logistic Regression. Precision và Recall của lớp 1 rất tốt, nhưng Precision của lớp 0 còn thấp (73%). Điều này cho thấy mô hình này có xu hướng dự đoán nhiều mẫu thuộc về lớp 1 hơn.

#### 5.6.4 Thuật toán Perceptron



Hình 5.12: Ma trận nhầm lẫn thuật toán Perceptron

Accuracy: 0.8815558588418795

Precision: Class 0: 0.74 – Class 1: 0.94

Recall: Class 0: 0.83 – Class 1: 0.90

F1-score: Class 0: 0.78 – Class 1: 0.92

Support: Class 0: 2591 – Class 1: 7667

Weighted Avg: Precision: 0.89 – Recall: 0.88 – F1-score: 0.88

**Nhận xét:** Perceptron có độ chính xác (88.15%) cao hơn so với Naive Bayes, nhưng thấp hơn Random Forest và Logistic Regression. Precision và Recall của lớp 1 rất tốt, nhưng Precision của lớp 0 còn thấp (74%).

#### 5.6.5 Đánh giá chung

Trong cả bốn mô hình, Random Forest và Logistic Regression đều có độ chính xác cao (trên 91%) và sự cân bằng tốt giữa các lớp. Naive Bayes và Perceptron có độ chính xác thấp hơn (khoảng 88%).

- **Random Forest:** Mô hình này nổi bật với độ chính xác cao nhất và hiệu suất tốt nhất trên cả hai lớp, nhưng có thể cần cải thiện recall của lớp 0.
- **Logistic Regression:** Gần đạt độ chính xác của Random Forest, với sự cân bằng tốt giữa precision và recall của cả hai lớp.
- **Naive Bayes:** Có độ chính xác thấp hơn 2 thuật toán Random Forest và Logistic Regression, thích hợp khi cần một mô hình đơn giản và nhanh chóng nhưng không yêu cầu hiệu suất cao.
- **Perceptron:** Nhỉnh hơn Naive Bayes 1 xíu về độ chính xác, có thể sử dụng khi cần một mô hình tuyến tính đơn giản.

## 5.7 Phân tích kết quả thuật toán

Lưu trữ kết quả dự đoán vào một DataFrame mới và gắn nhãn dự đoán trên toàn bộ dữ liệu như sau:

1. Dự đoán nhãn cho toàn bộ dữ liệu với từng mô hình.

```
[52]: df_predicted_labels_rf = best_rf.predict(df_important_feature)
df_predicted_labels_lr = best_lr.predict(df_important_feature)
df_predicted_labels_nb = best_nb.predict(df_important_feature)
df_predicted_labels_p = best_perceptron.predict(df_important_feature)
```

2. Tạo một DataFrame mới để lưu trữ các giá trị dự đoán cùng với các thuộc tính gốc.

```
[53]: # Tạo DataFrame mới để lưu trữ các giá trị dự đoán và các thuộc tính gốc
df_predicted_data = pd.DataFrame({
    'Discount': df['Discount'],
    'City': df['City'],
    'City_Score': df['City_Score'],
    'State': df['State'],
    'State_Score': df['State_Score'],
    'Country': df['Country'],
    'Country_Score': df['Country_Score'],
    'Predicted_Label_RF': df_predicted_labels_rf,
    'Predicted_Label_LR': df_predicted_labels_lr,
    'Predicted_Label_NB': df_predicted_labels_nb,
    'Predicted_Label_P': df_predicted_labels_p,
})
```

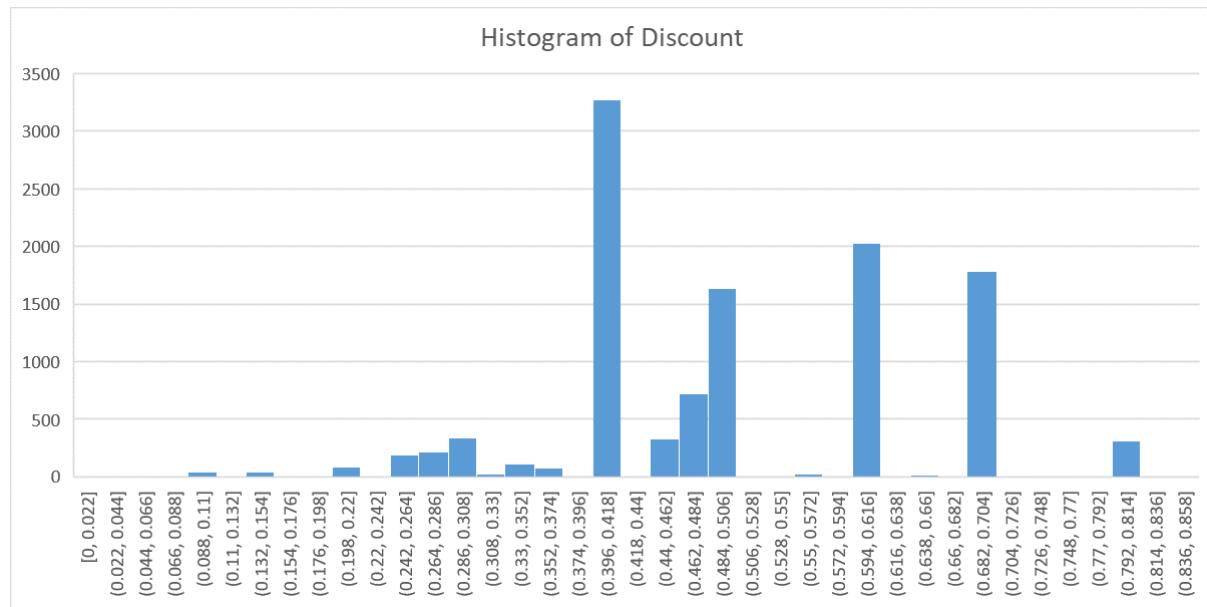
3. Lưu DataFrame này vào một tệp CSV có tên “predicted\_data\_with\_df.csv” để sử dụng phân tích trên Excel.

```
[54]: # Lưu DataFrame này xuống file CSV
df_predicted_data.to_csv('predicted_data_with_df.csv', index=False)
```

## 5.7.1 Thuận toán Random Forest

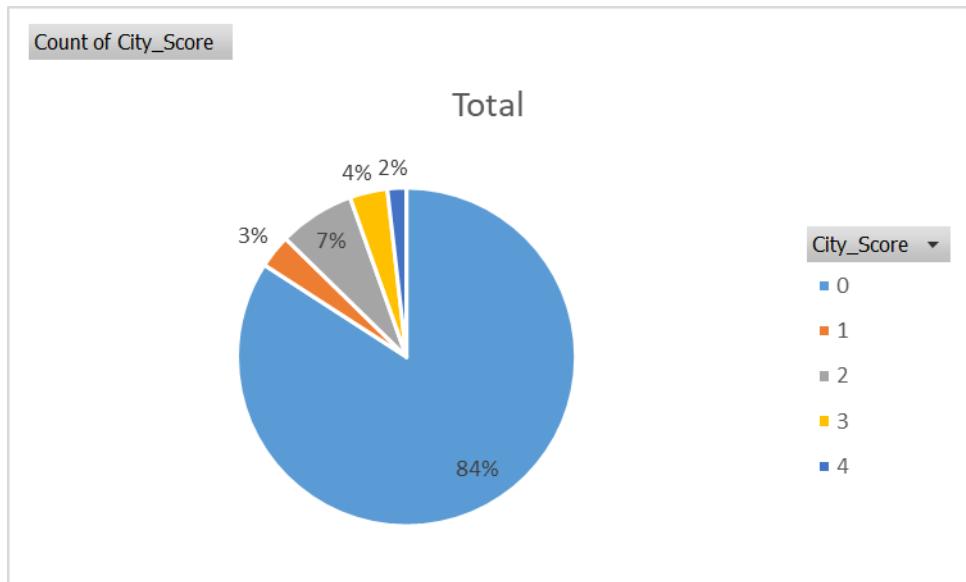
### 5.7.1.1 Class 0: (Lô)

- Thuộc tính **Discount**:

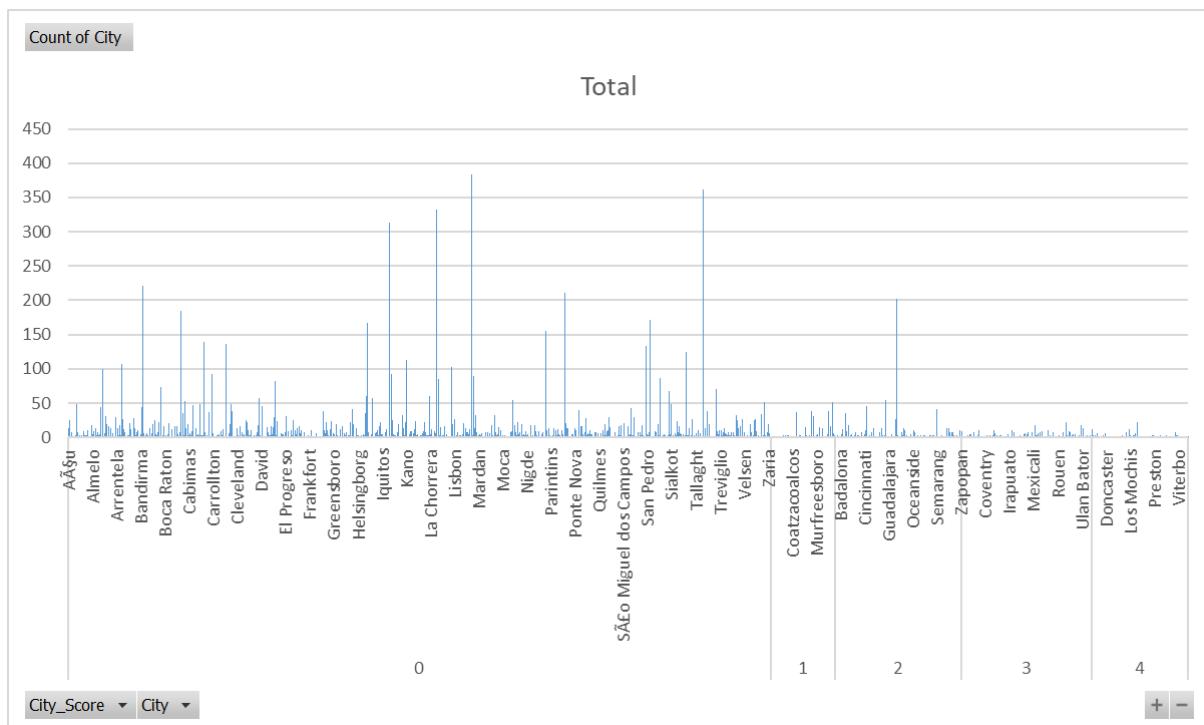


Hình 5.13: Biểu đồ Histogram của thuộc tính Discount (RF-0)

- Thuộc tính **City** and **City\_score**:

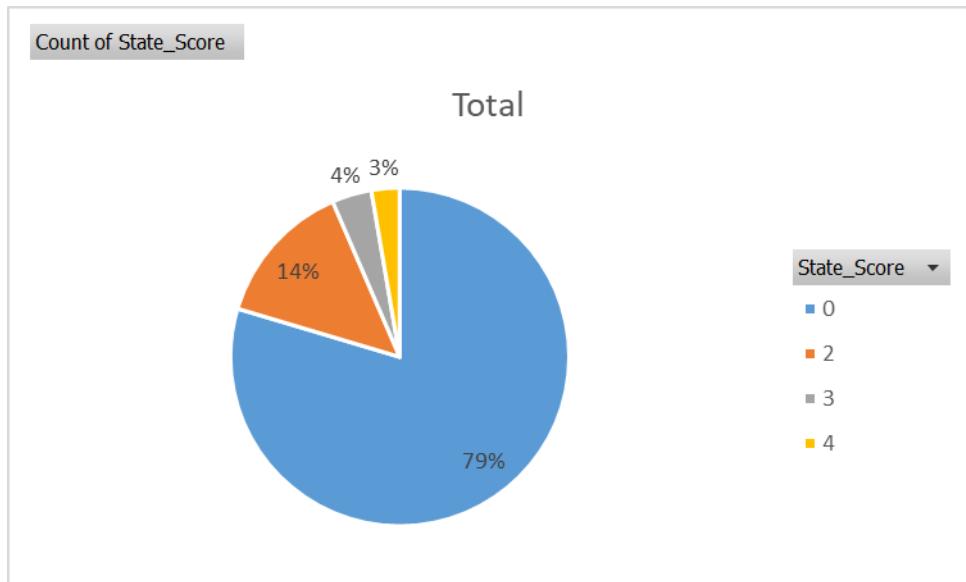


Hình 5.14: Phản trăng của các mức điểm lợi nhuận theo thành phố (RF-0)

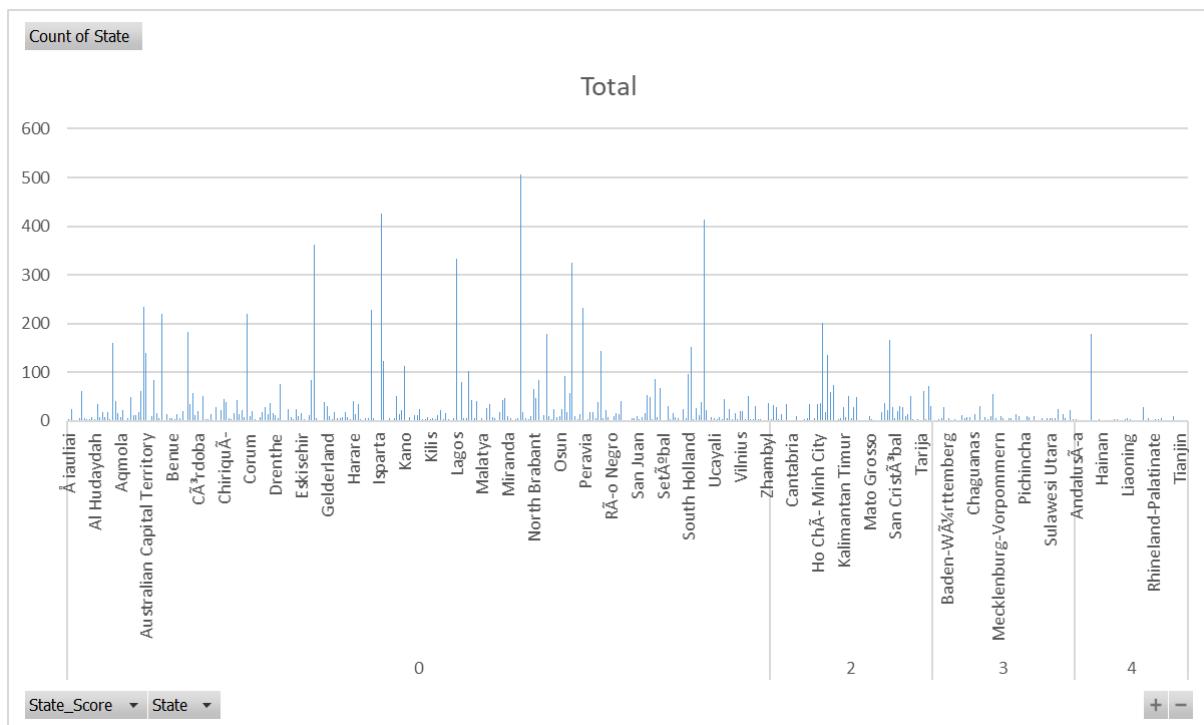


Hình 5.15: Số lượng và tần suất các thành phố theo điểm lợi nhuận (RF-0)

- Thuộc tính State and State\_score:

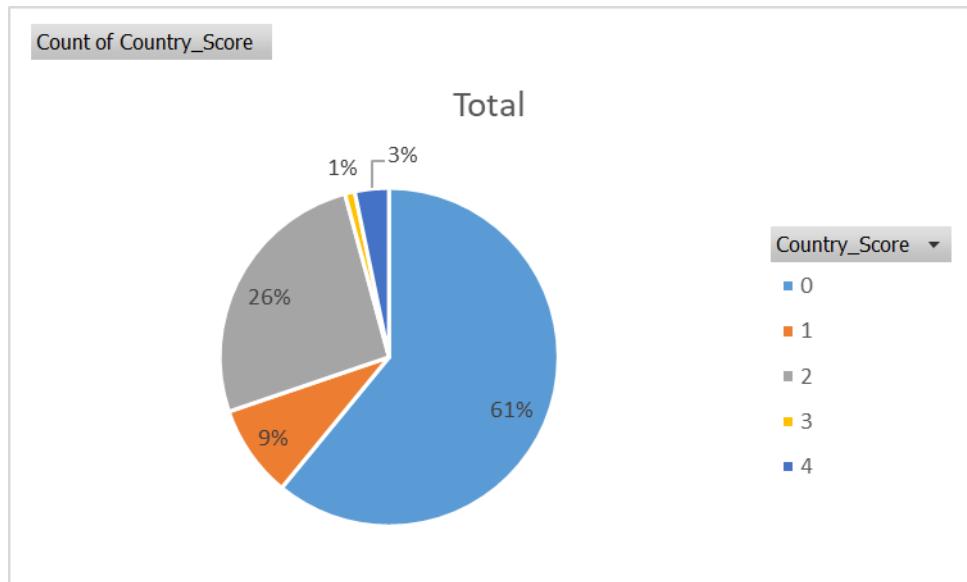


Hình 5.16: Phản trăng của các mức điểm lợi nhuận theo bang (RF-0)

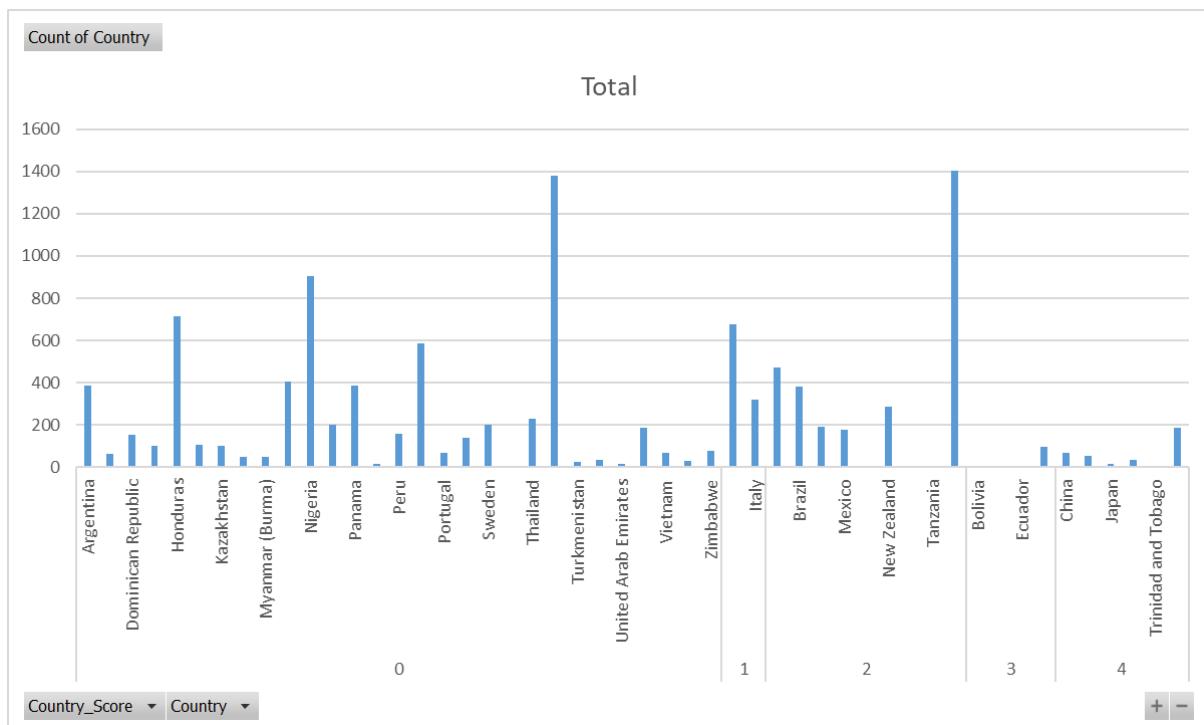


Hình 5.17: Số lượng và tần suất các bang theo điểm lợi nhuận (RF-0)

- Thuộc tính Country and Country\_score:



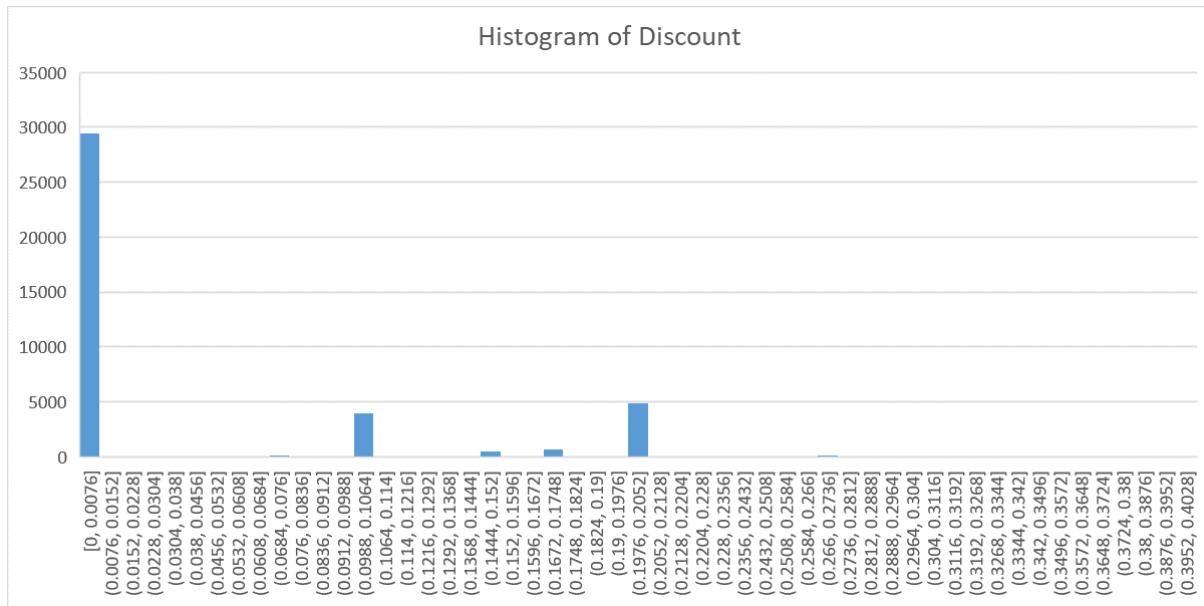
Hình 5.18: Phản trăng của các mức điểm lợi nhuận theo quốc gia (RF-0)



Hình 5.19: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (RF-0)

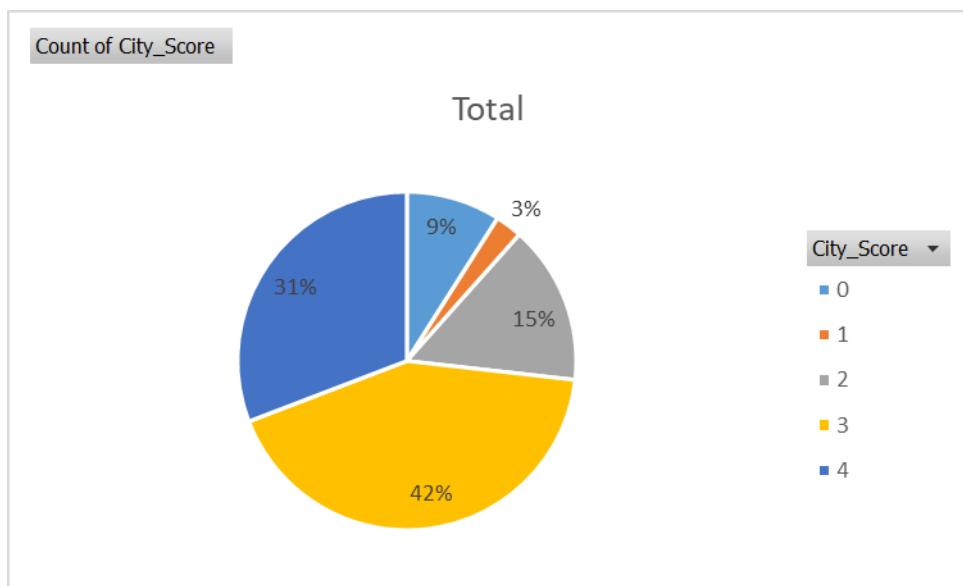
#### 5.7.1.2 Class 1: (Có lợi nhuận)

- Thuộc tính **Discount**:

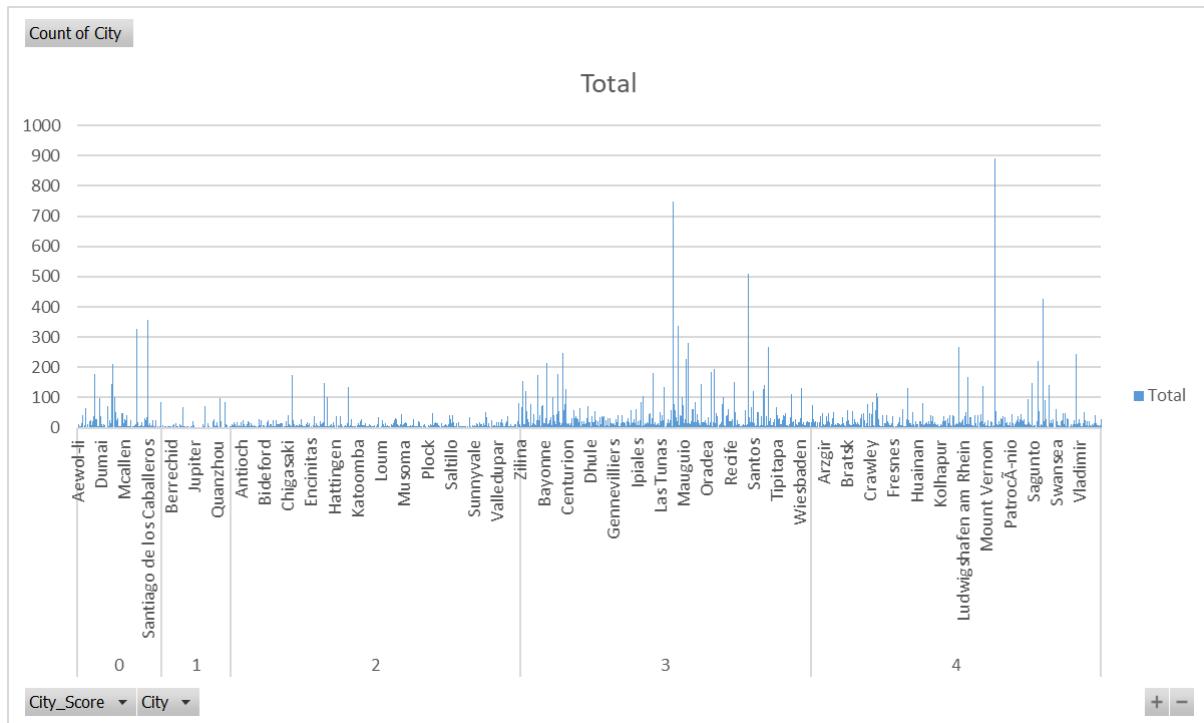


Hình 5.20: Biểu đồ Histogram của thuộc tính Discount (RF-1)

- Thuộc tính City and City\_score:

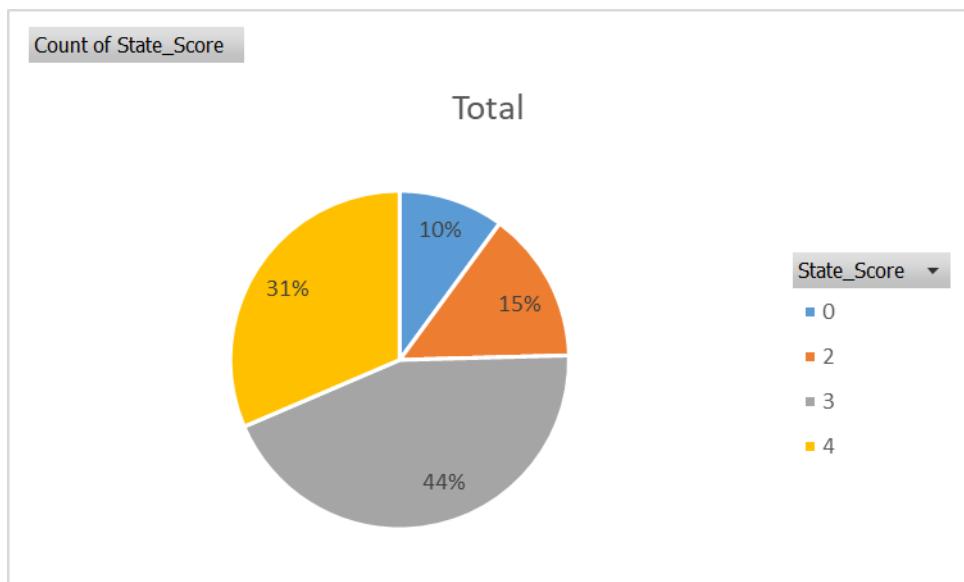


Hình 5.21: Phân trăm của các mức điểm lợi nhuận theo thành phố (RF-1)

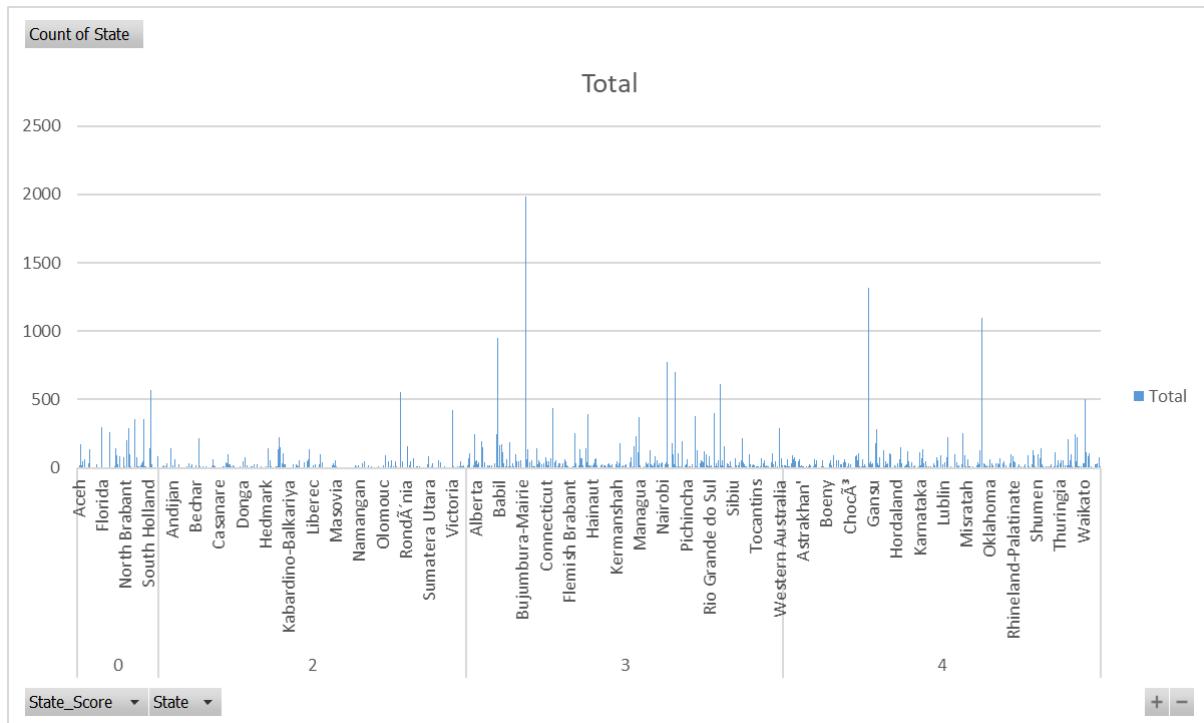


Hình 5.22: Số lượng và tần suất các thành phố theo điểm lợi nhuận (RF-1)

- Thuộc tính State and State\_score:

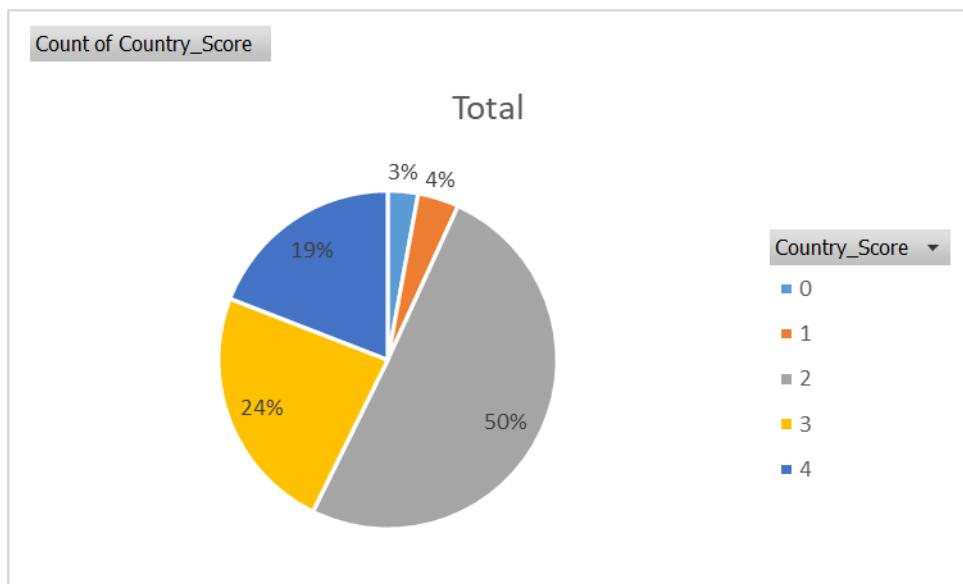


Hình 5.23: Phản trăng của các mức điểm lợi nhuận theo bang (RF-1)

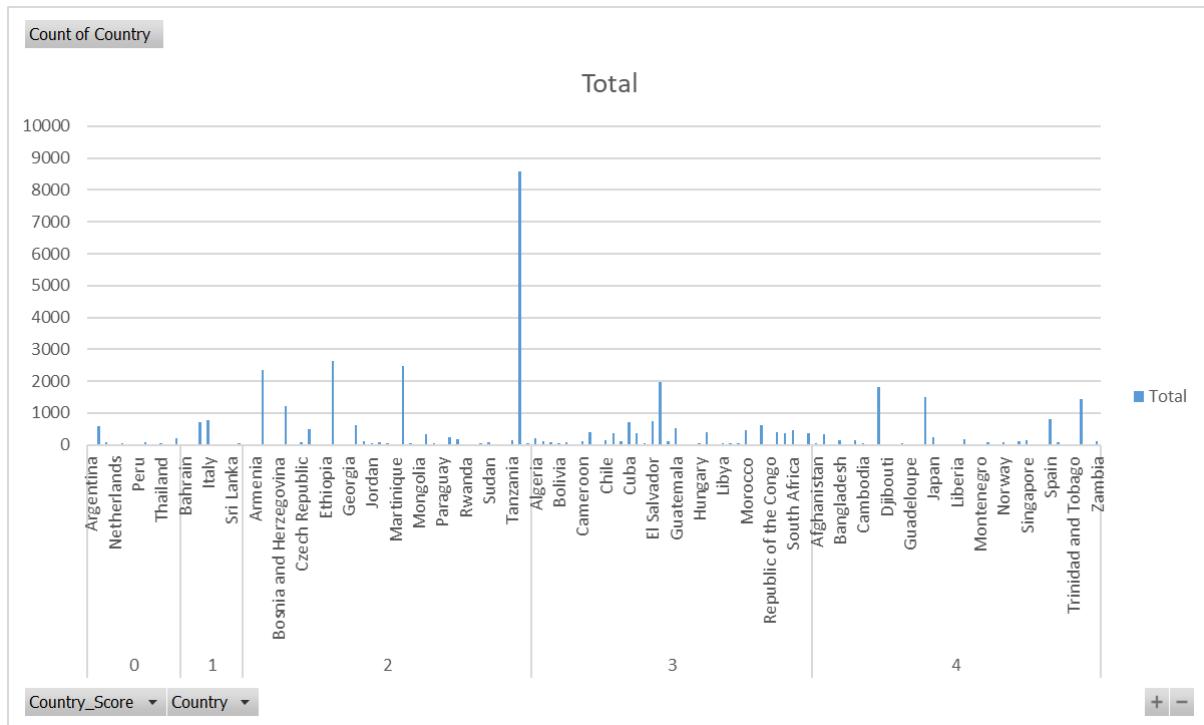


Hình 5.24: Số lượng và tần suất các bang theo điểm lợi nhuận (RF-1)

- Thuộc tính **Country** and **Country\_score**:



Hình 5.25: Phản trăm của các mức điểm lợi nhuận theo quốc gia (RF-1)



Hình 5.26: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (RF-1)

#### 5.7.1.3 Nhận xét

##### Class 0: Lỗ

- **Discount:** Phần lớn các đơn hàng trong Class 0 mức chiết khấu cao, tập trung chủ yếu từ 0.4 đến 0.7, các mức chiết khấu thấp hơn chiếm tỷ lệ ít. Điều này cho thấy các đơn hàng có chiết khấu cao thường rơi vào lớp lỗ.
- **Thành phố:** Khoảng 84% thành phố có điểm lợi nhuận âm (0). Với tỷ lệ thành phố có điểm lợi nhuận rất cao (4), cao (3), trung bình (2), và thấp (1) chiếm rất ít lần lượt là 2%, 4%, 7%, 3% và chiếm tổng cộng 16%. Các thành phố có điểm lợi nhuận âm (0) chiếm đa số và xuất hiện dày đặc.
- **Bang:** Khoảng 79% bang có điểm lợi nhuận âm (0). Với tỷ lệ bang có điểm lợi nhuận rất cao (4), cao (3), và trung bình (2) lần lượt là 3%, 4%, và 14% và chiếm tổng 21%. Các bang có điểm lợi nhuận âm (0) chiếm đa số và xuất hiện dày đặc.
- **Quốc gia:** Khoảng 61% quốc gia có điểm lợi nhuận âm (0). tỷ lệ quốc gia có điểm lợi nhuận rất cao (4), cao (3), trung bình (2), và thấp (1) lần lượt là 3%,

1%, 26%, và 9% chiếm tổng cộng 39%. Các quốc gia có lợi nhuận âm chiếm ưu thế và xuất hiện nhiều hơn so với các quốc gia có lợi nhuận.

- Phần lớn các địa điểm trên thuộc về Châu Âu (Denmark, Sweden, Portugal, Ireland,...), Châu Á (Tajikistan, Turkmenistan, Pakistan, Kazakhstan, Yemen, Turkey), Châu Phi (Haiti, Nigeria, Uganda, Zimbabwe, Papua New Guinea, Panama). Đặc điểm chung các khu vực trên là thu nhập trung bình thấp.

➔ **Tóm lại:** Đặc điểm chung của các đơn hàng trong Class 0 cho thấy các đơn hàng có chiết khấu cao thường rơi vào lớp 0 (lỗ) và phần lớn các thành phố, bang, và quốc gia có điểm lợi nhuận âm (0), với tần suất xuất hiện dày đặc. Địa điểm tập trung ở các khu vực kinh tế tương đối không ổn định và thuộc loại hình nền kinh tế đang phát triển, thu nhập trung bình thấp.

### Class 1: Có lợi nhuận

- **Discount:** Các đơn hàng thường có mức chiết khấu rất thấp, gần như bằng 0. Các mức chiết khấu lớn hơn ( $> 0.3$ ) hầu như không xuất hiện.
- **Thành phố:** 91% thành phố có điểm lợi nhuận dương, với tỷ lệ thành phố có lợi nhuận rất cao (4), cao (3), trung bình (2), và thấp (1) lần lượt là 31%, 42%, 15%, và 3%. Thành phố có điểm lợi nhuận âm chiếm rất ít khoảng 9%. Phần lớn các thành phố có lợi nhuận với các thành phố có lợi nhuận âm xuất hiện ít và thưa thớt.
- **Bang:** Khoảng 90% bang có lợi nhuận dương, với tỷ lệ bang có lợi nhuận rất cao (4), cao (3), và trung bình (2) lần lượt là 31%, 44%, 15%. Bang có điểm lợi nhuận âm chiếm rất ít khoảng 10%. Các bang có lợi nhuận âm xuất hiện ít và thưa thớt.
- **Quốc gia:** Khoảng 97% quốc gia có lợi nhuận dương, với tỷ lệ quốc gia có điểm lợi nhuận rất cao (4), cao (3), trung bình (2), và thấp (1) lần lượt là 19%, 24%, 50%, và 4%. Quốc gia có điểm lợi nhuận âm chiếm rất ít khoảng 3%. Phần lớn các quốc gia có lợi nhuận dương, với các quốc gia có lợi nhuận âm xuất hiện ít và thưa thớt.

- Các địa điểm trên tập trung khá đa dạng Đông Âu, Nam Á, Trung Đông, Châu Phi, Tây Âu, Bắc Âu, Châu Á, Châu Mỹ. Khác với các quốc gia trong class 0 (lõi). Các quốc gia trong class này thường là quốc gia phát triển, đa dạng, ổn định. Thu nhập tương đối cao và ổn định.

➔ **Tóm lại:** Đặc điểm chung của các đơn hàng trong Class 1 cho thấy cho thấy đơn hàng có chiết khấu thấp thường có lợi nhuận và phần lớn các thành phố, bang, và quốc gia có điểm lợi nhuận dương, với tần suất xuất hiện dày đặc và địa điểm tập trung ở các khu vực phát triển và thị trường kinh tế năng động.

## 5.7.2 Thuật toán Logistic Regression

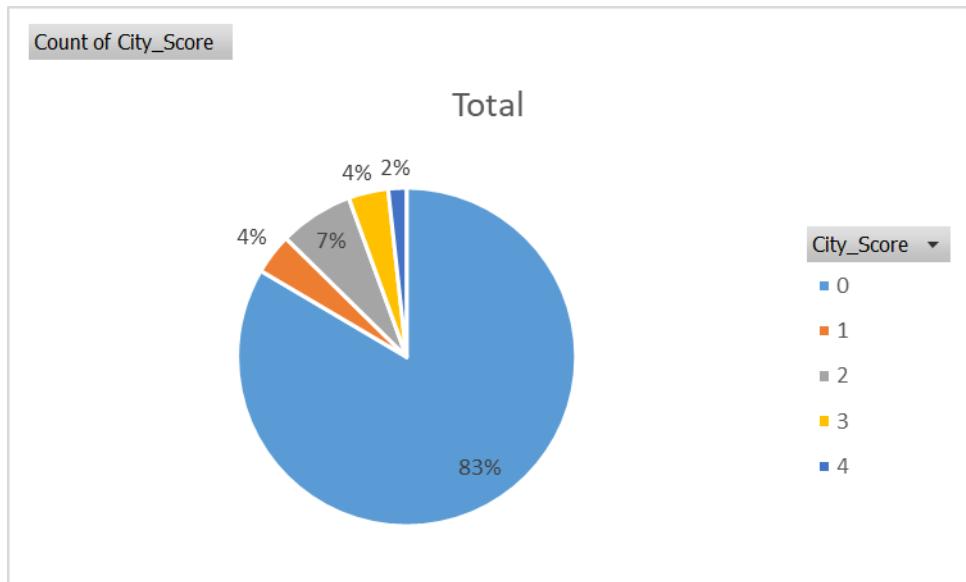
### 5.7.2.1 Class 0: ( $\tilde{L_0}$ )

- Thuộc tính **Discount**:

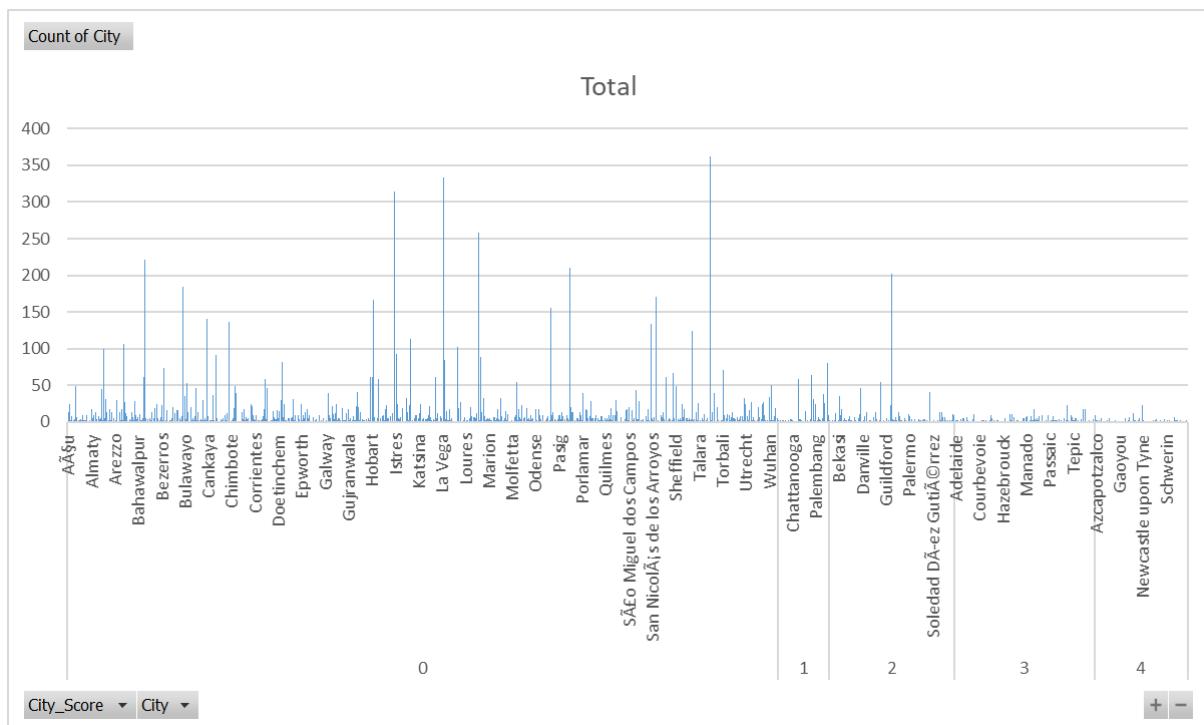


Hình 5.27: Biểu đồ Histogram của thuộc tính Discount (LR-0)

- Thuộc tính **City** and **City\_score**:

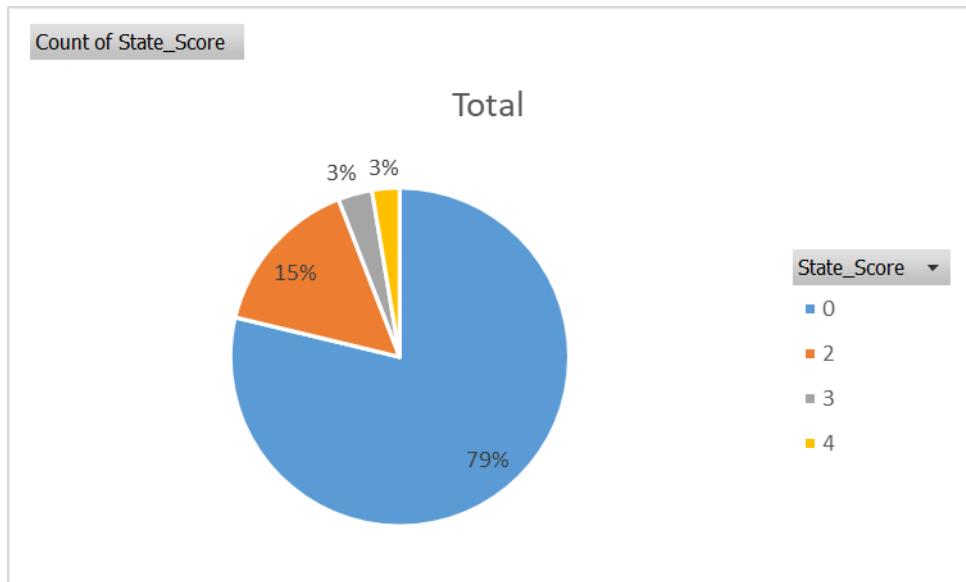


Hình 5.28: Phân trăm của các mức điểm lợi nhuận theo thành phố (LR-0)

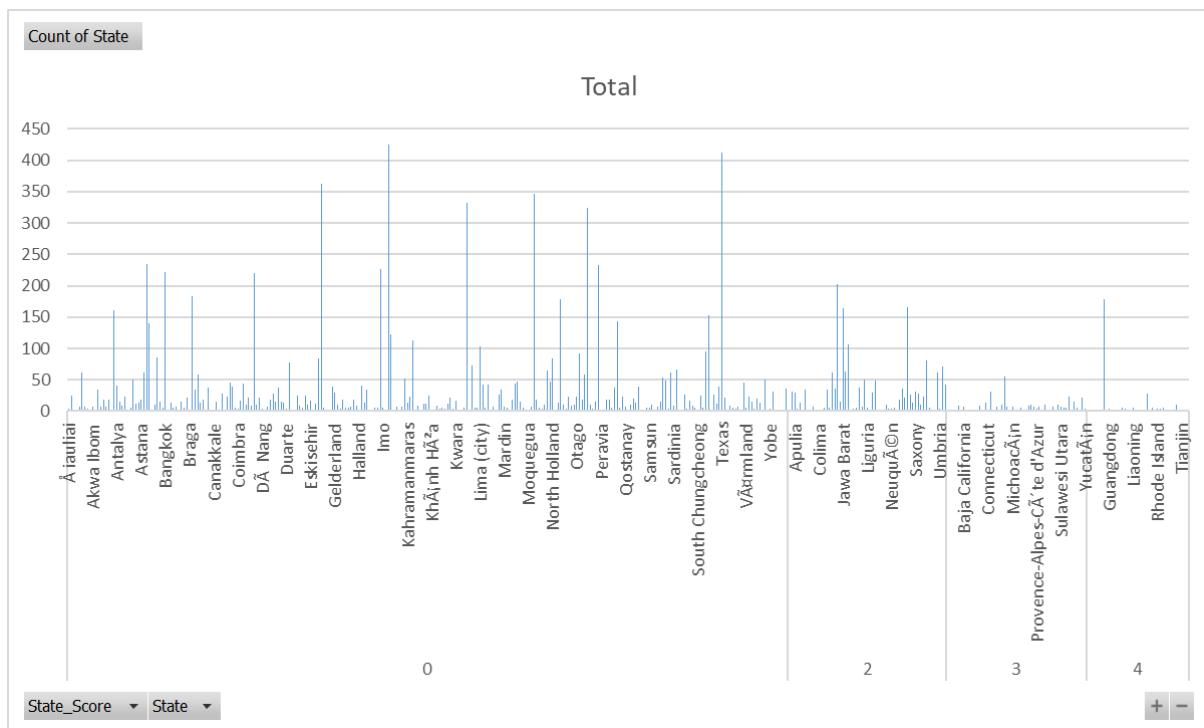


Hình 5.29: Số lượng và tần suất các thành phố theo điểm lợi nhuận (LR-0)

- Thuộc tính State and State\_score:

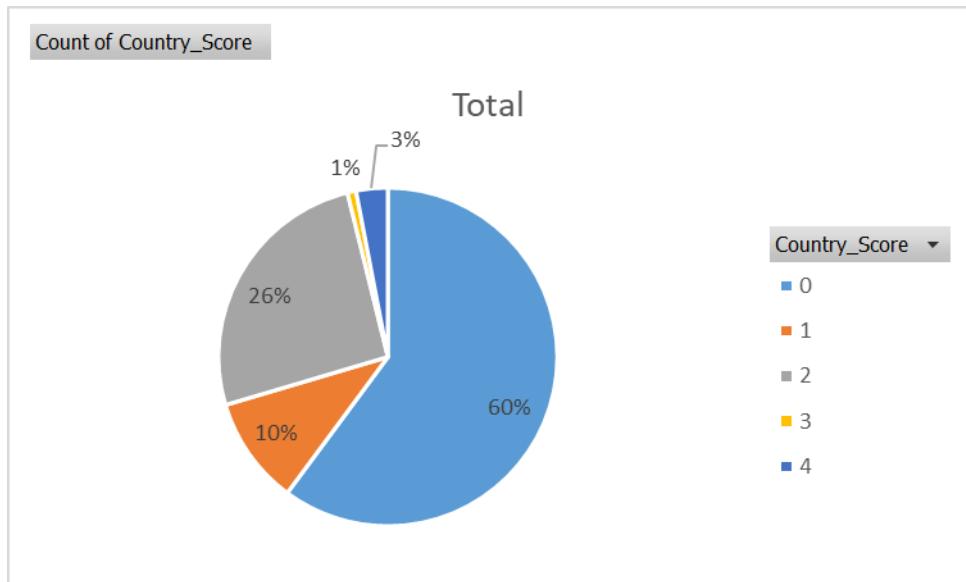


Hình 5.30: Phần trăm của các mức điểm lợi nhuận theo bang (LR-0)

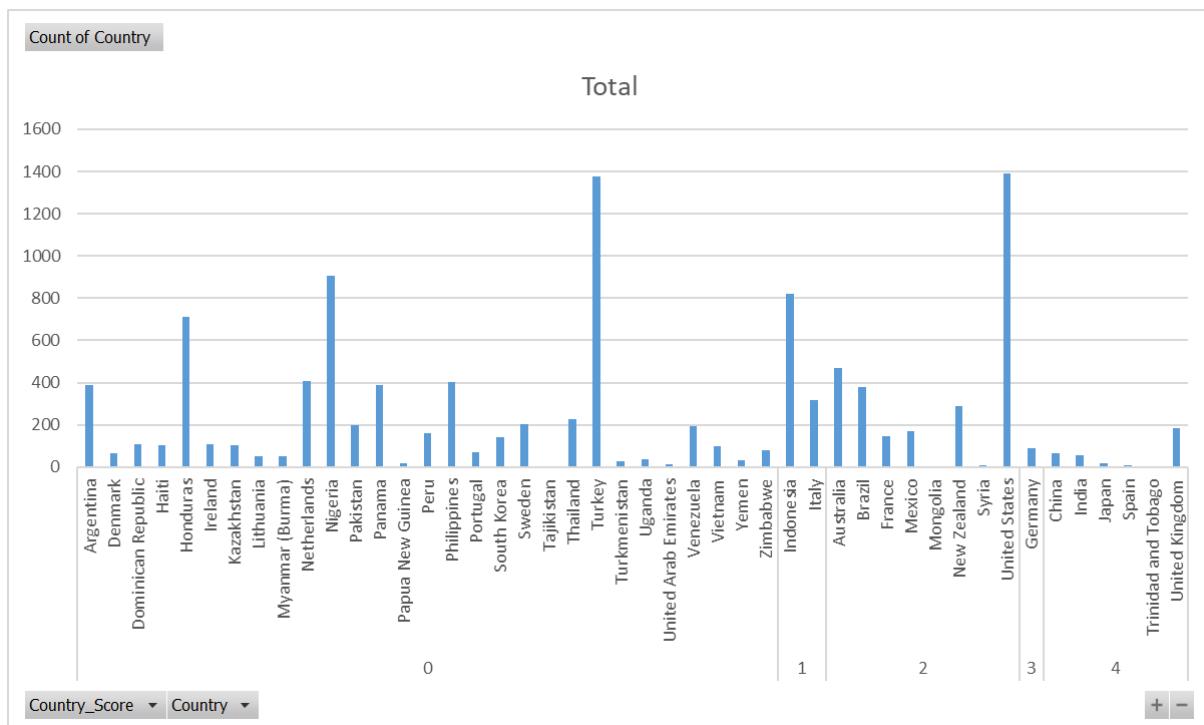


Hình 5.31: Số lượng và tần suất các bang theo điểm lợi nhuận (LR-0)

- Thuộc tính **Country** and **Country\_score**:



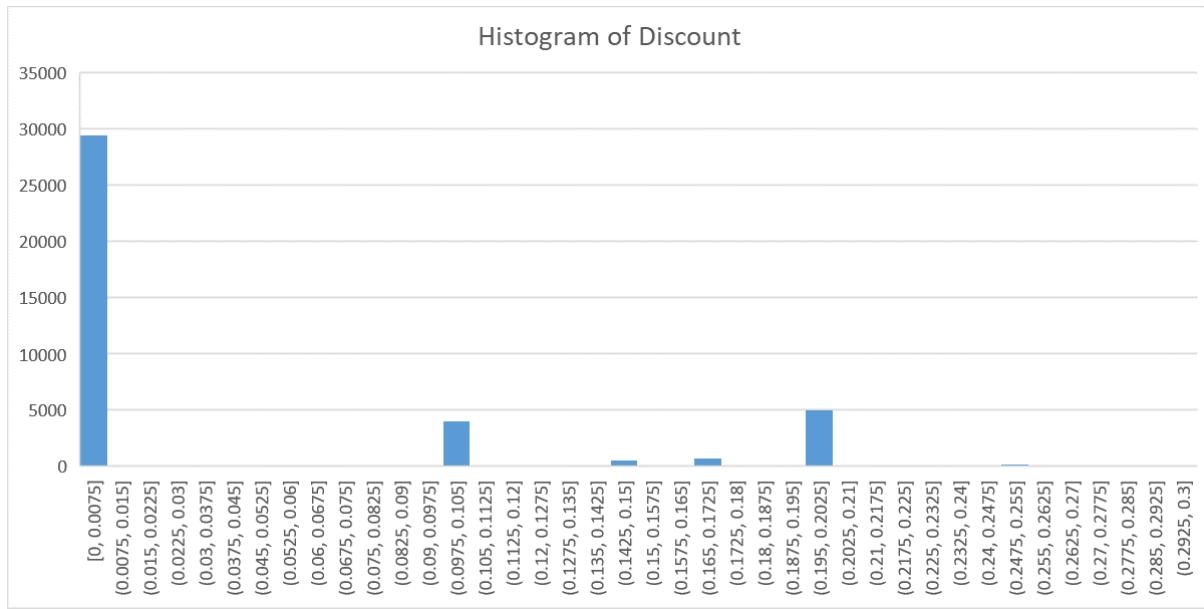
Hình 5.32: Phân trăm của các mức điểm lợi nhuận theo quốc gia (LR-0)



Hình 5.33: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (LR-0)

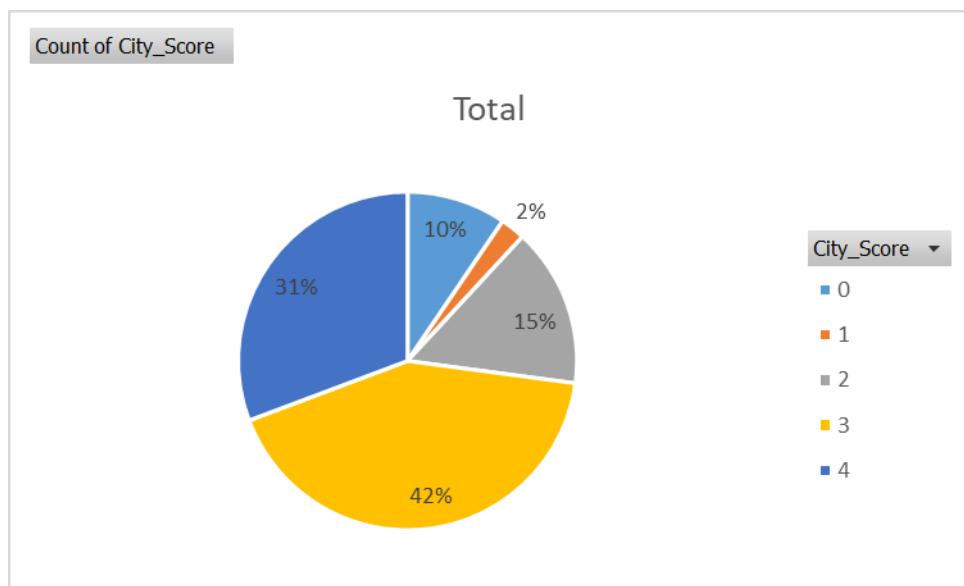
### 5.7.2.2 Class 1: (Có lợi nhuận)

- Thuộc tính **Discount**:

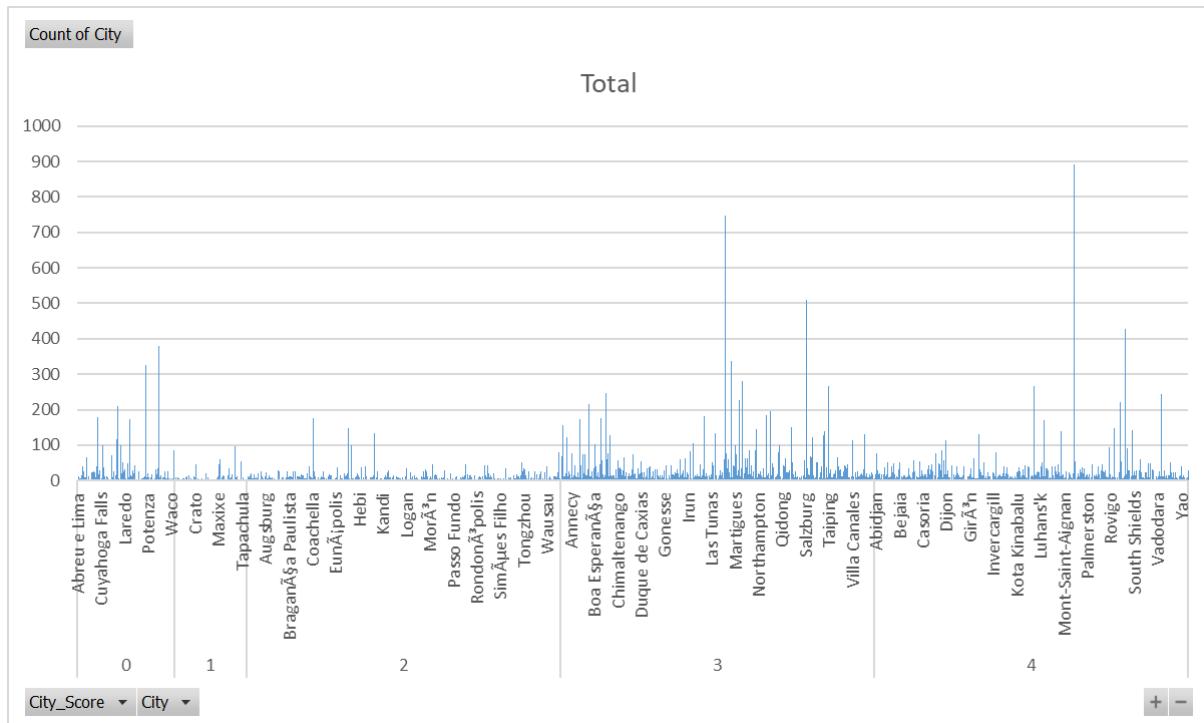


Hình 5.34: Biểu đồ Histogram của thuộc tính Discount (LR-1)

- Thuộc tính City and City\_score:

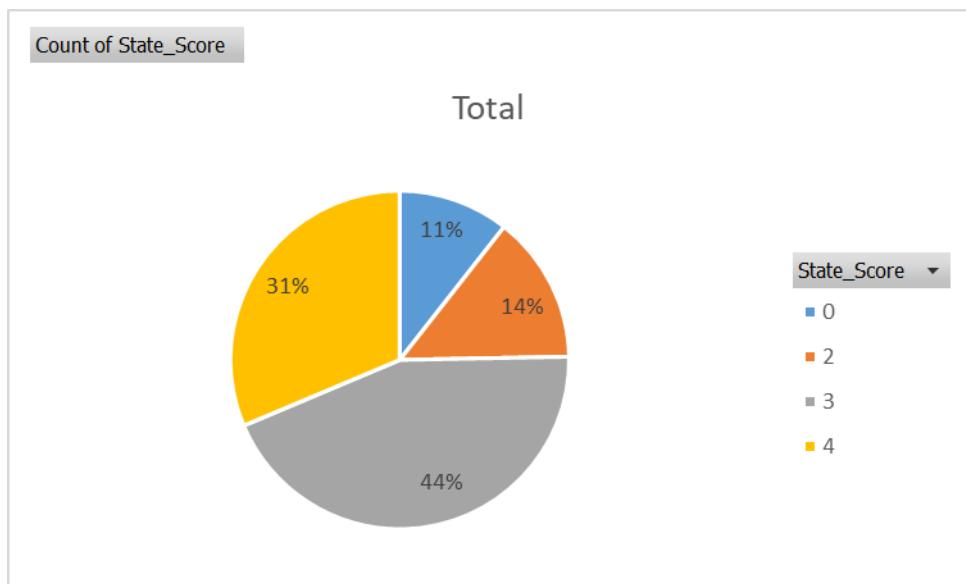


Hình 5.35: Phân trăm của các mức điểm lợi nhuận theo thành phố (LR-1)

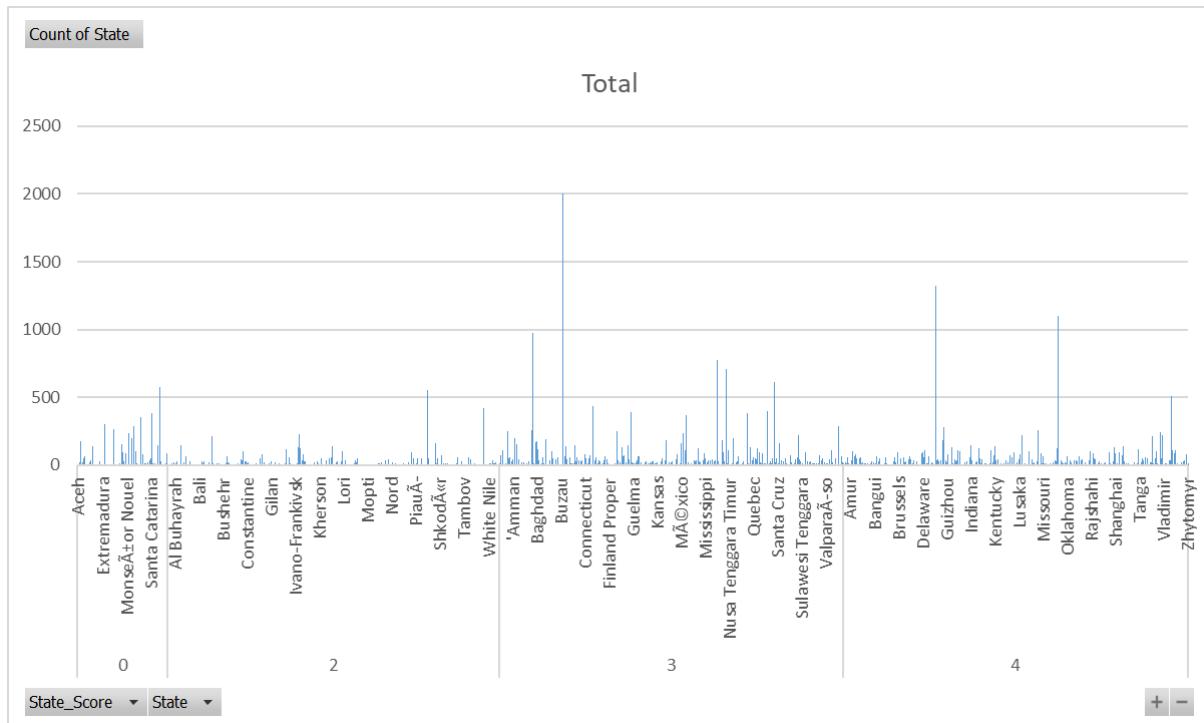


Hình 5.36: Số lượng và tần suất các thành phố theo điểm lợi nhuận (LR-1)

- Thuộc tính State and State\_score:

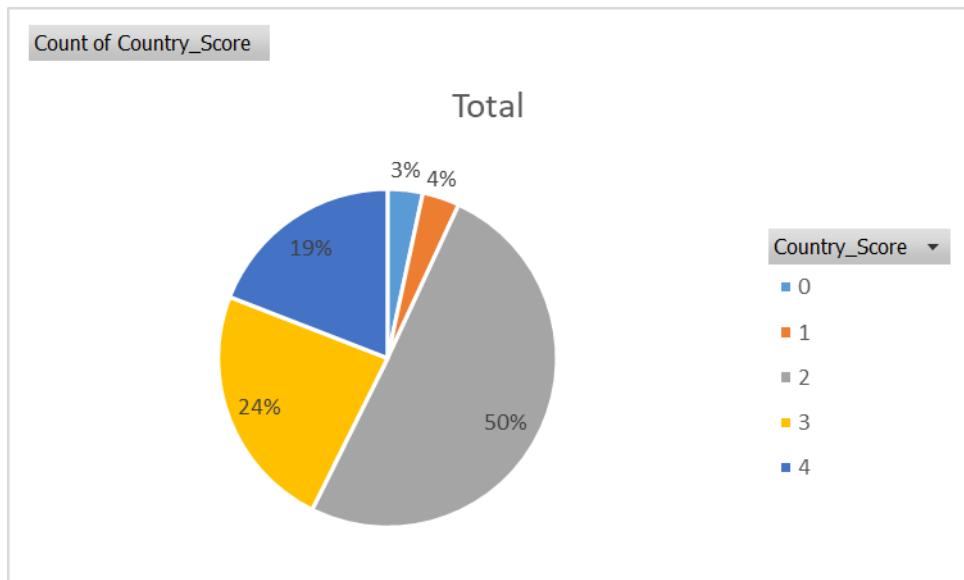


Hình 5.37: Phản trăng của các mức điểm lợi nhuận theo bang (LR-1)

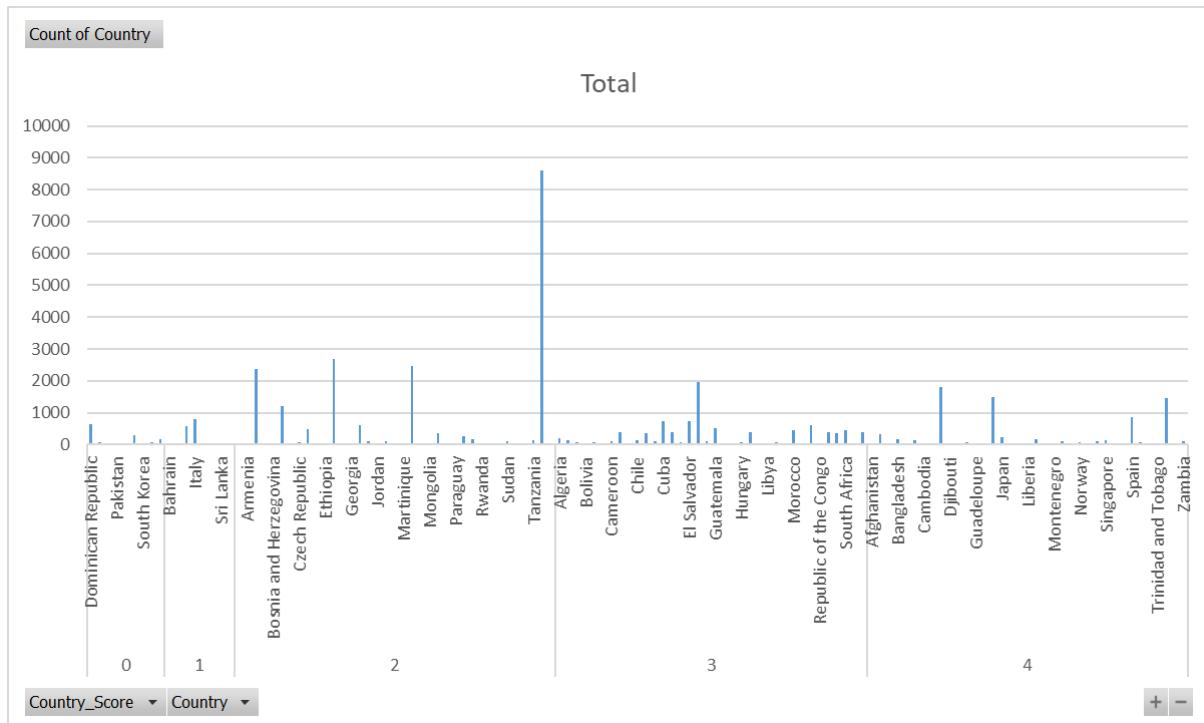


Hình 5.38: Số lượng và tần suất các bang theo điểm lợi nhuận (LR-1)

- Thuộc tính **Country** and **Country\_score**:



Hình 5.39: Phân trăm của các mức điểm lợi nhuận theo quốc gia (LR-1)



Hình 5.40: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (LR-1)

### 5.7.2.3 Nhận xét

Thuật toán Logistic Regression có kết quả tương tự với Random Forest, điểm khác biệt chính nằm ở tỷ lệ phần trăm ở các loại điểm lợi nhuận. Xu hướng chung và đặc điểm của các class vẫn nhất quán:

Class 0: Lỗ

- Discount: Đơn hàng có chiết khấu cao (0.4 - 0.7) thường rơi vào lớp lỗ.
- Thành phố: 83% thành phố có điểm lợi nhuận âm.
- Bang: 79% bang có điểm lợi nhuận âm.
- Quốc gia: 60% quốc gia có điểm lợi nhuận âm.
- Khu vực: Đặc điểm chung các khu vực trên là thu nhập trung bình đến thấp.

Class 1: Có lợi nhuận

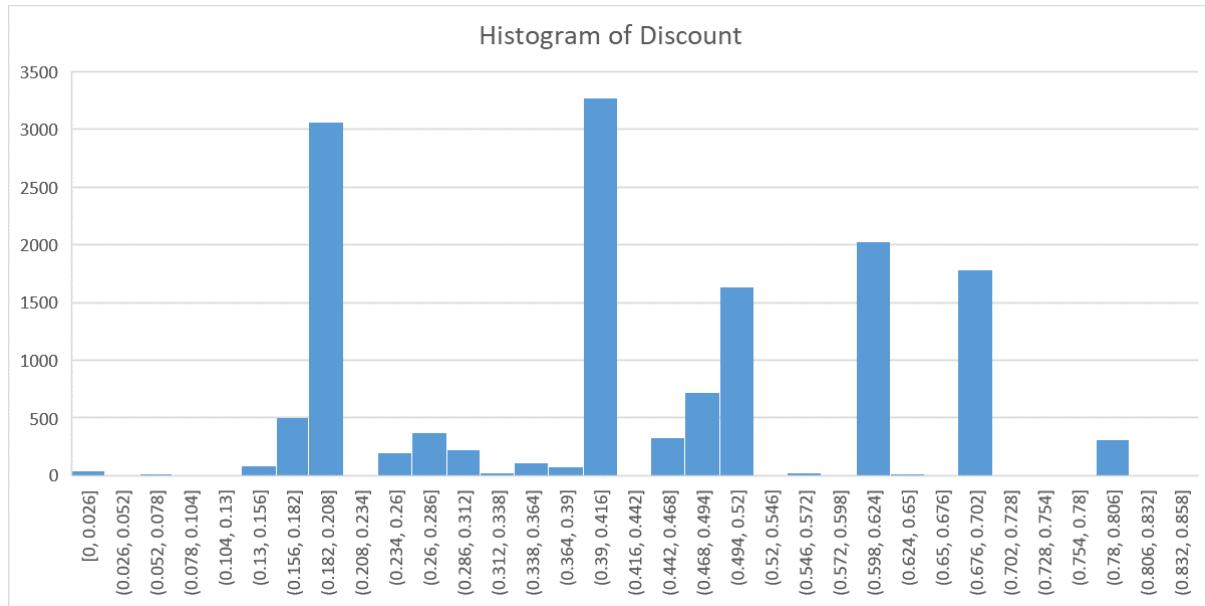
- Discount: Đơn hàng có chiết khấu rất thấp hoặc gần như bằng 0.
- Thành phố: 90% thành phố có điểm lợi nhuận dương.
- Bang: 89% bang có lợi nhuận dương.
- Quốc gia: 97% quốc gia có lợi nhuận dương.

- Khu vực: Đa dạng ở các khu vực phát triển và ổn định kinh tế.

### 5.7.3 Thuật toán Naïve Bayes

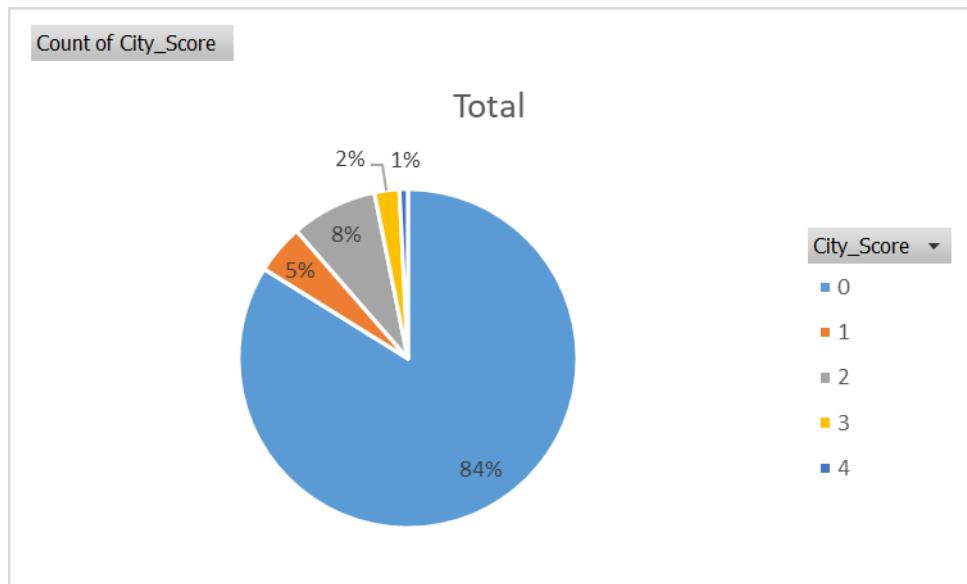
#### 5.7.3.1 Class 0: ( $\tilde{L_0}$ )

- Thuộc tính **Discount**:

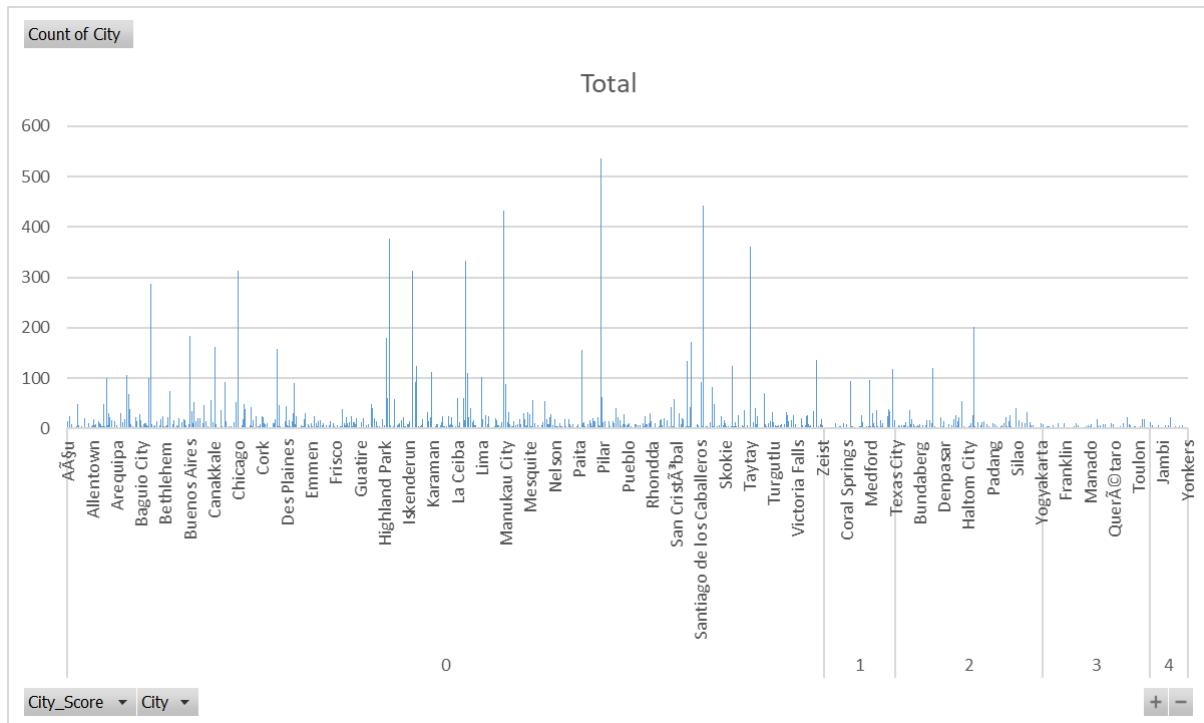


Hình 5.41: Biểu đồ Histogram của thuộc tính Discount (NB-0)

- Thuộc tính **City** and **City\_score**:

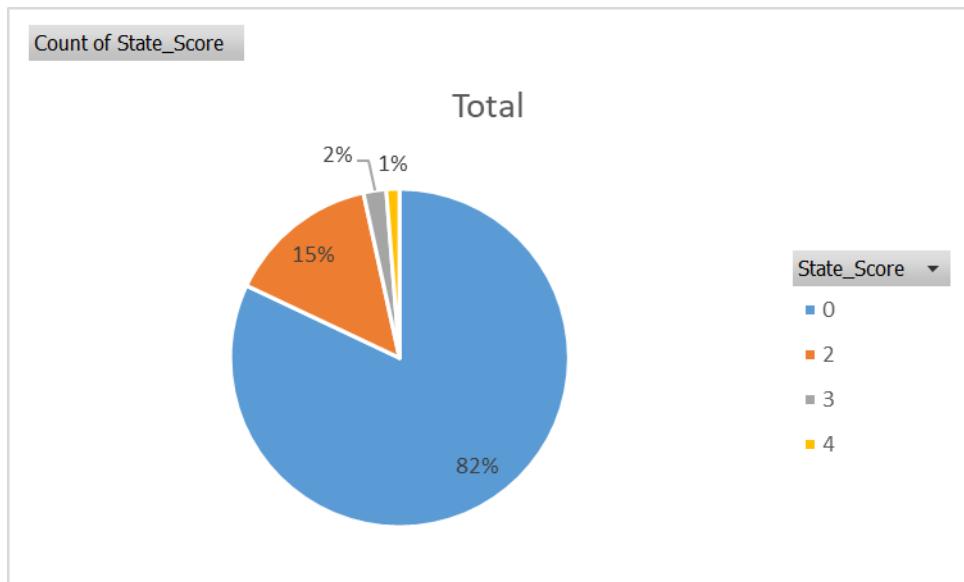


Hình 5.42: Phản trăng của các mức điểm lợi nhuận theo thành phố (NB-0)

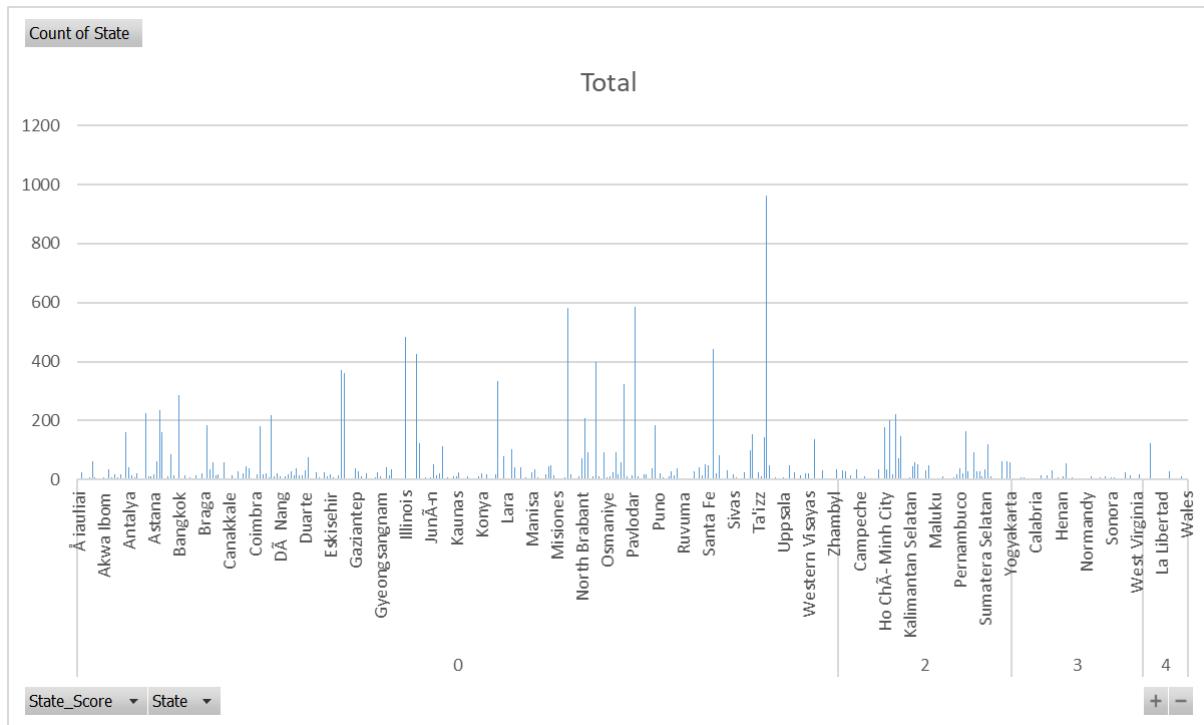


Hình 5.43: Số lượng và tần suất các thành phố theo điểm lợi nhuận (NB-0)

- Thuộc tính State and State\_score:

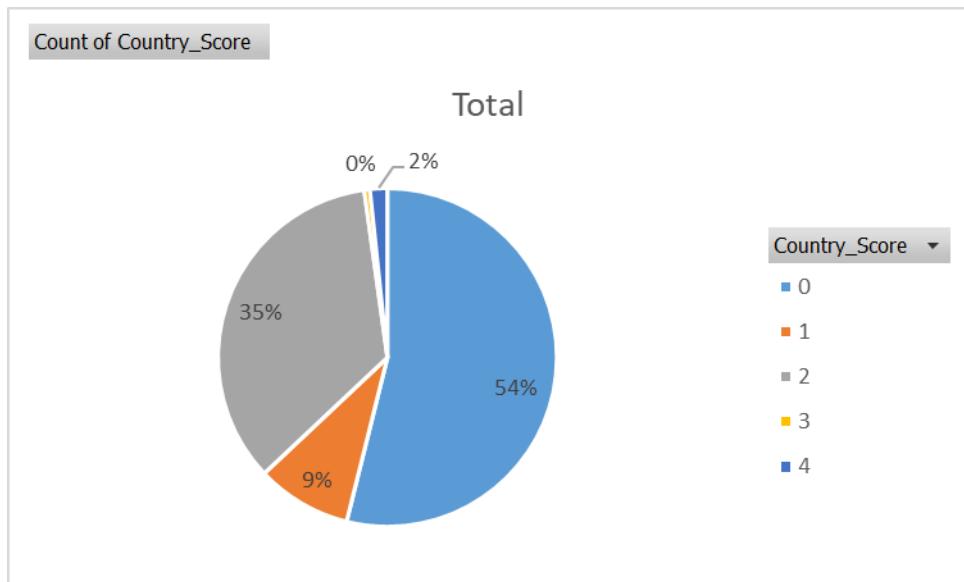


Hình 5.44: Phản trăng của các mức điểm lợi nhuận theo bang (NB-0)

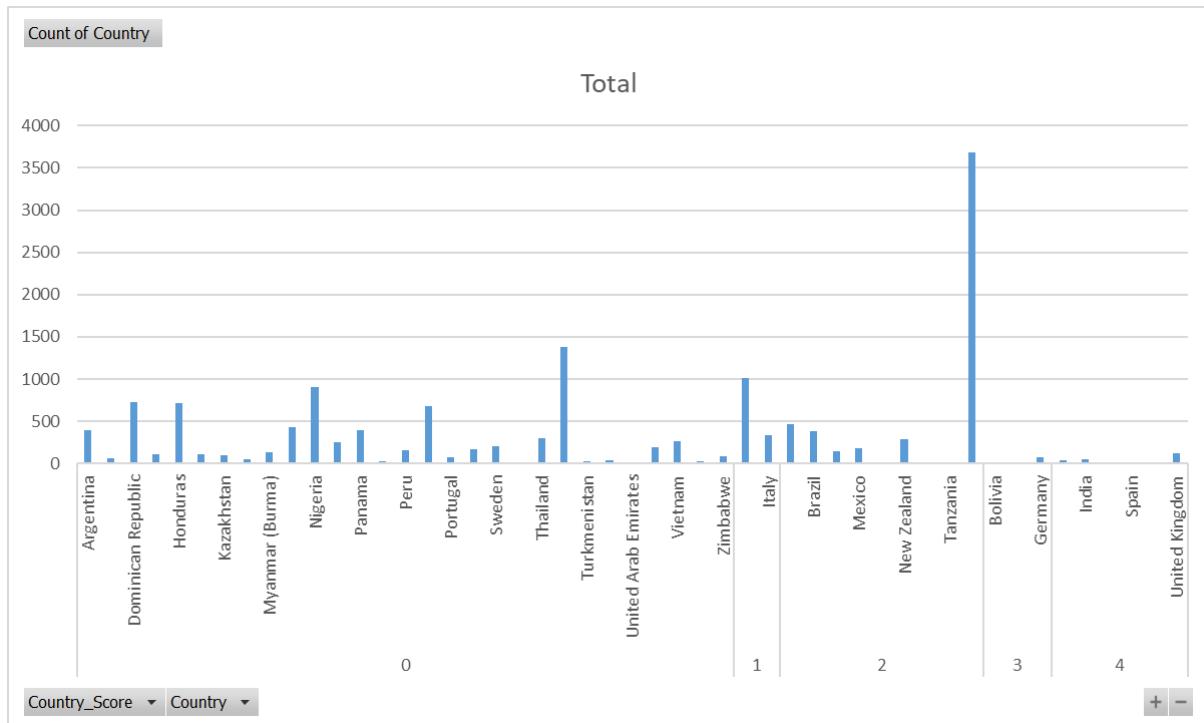


Hình 5.45: Số lượng và tần suất các bang theo điểm lợi nhuận (NB-0)

- Thuộc tính **Country** and **Country\_score**:



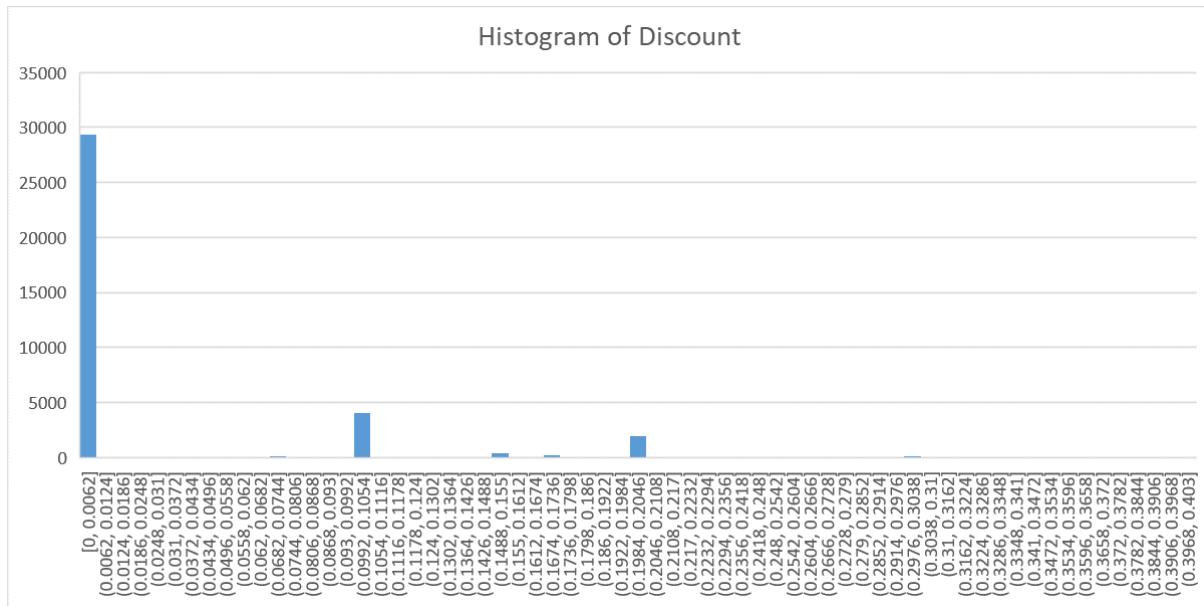
Hình 5.46: Phần trăm của các mức điểm lợi nhuận theo quốc gia (NB-0)



Hình 5.47: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (NB-0)

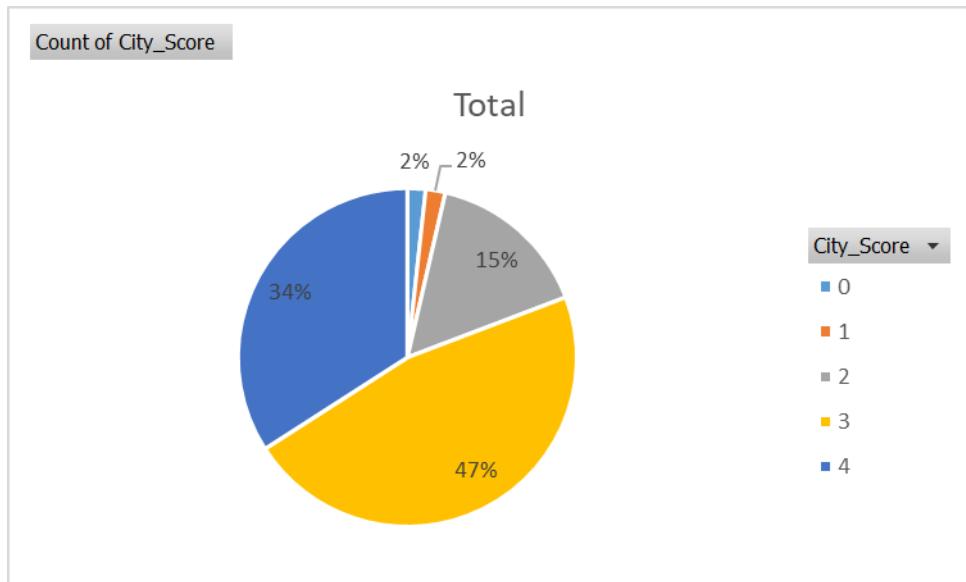
### 5.7.3.2 Class 1: (Có lợi nhuận)

- Thuộc tính Discount:

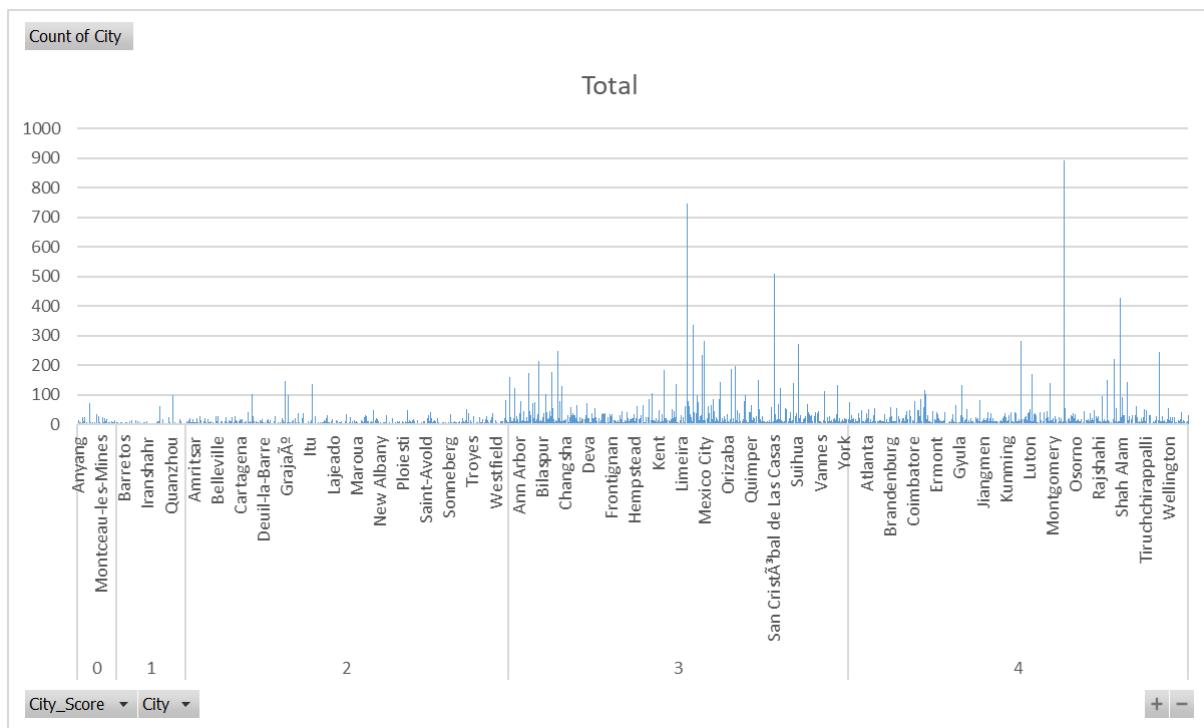


Hình 5.48: Biểu đồ Histogram của thuộc tính Discount (NB-1)

- Thuộc tính City and City\_score:

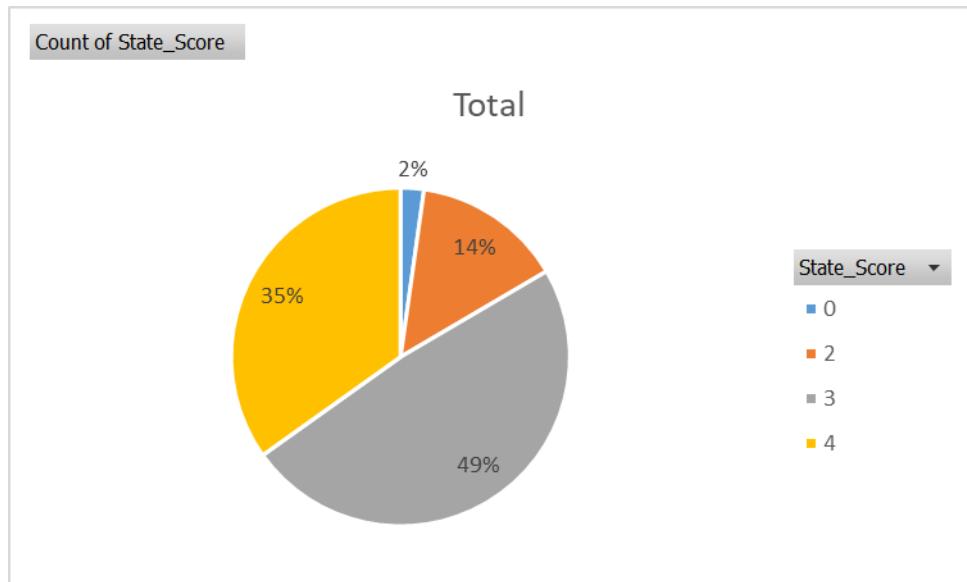


Hình 5.49: Phân trăm của các mức điểm lợi nhuận theo thành phố (NB-1)

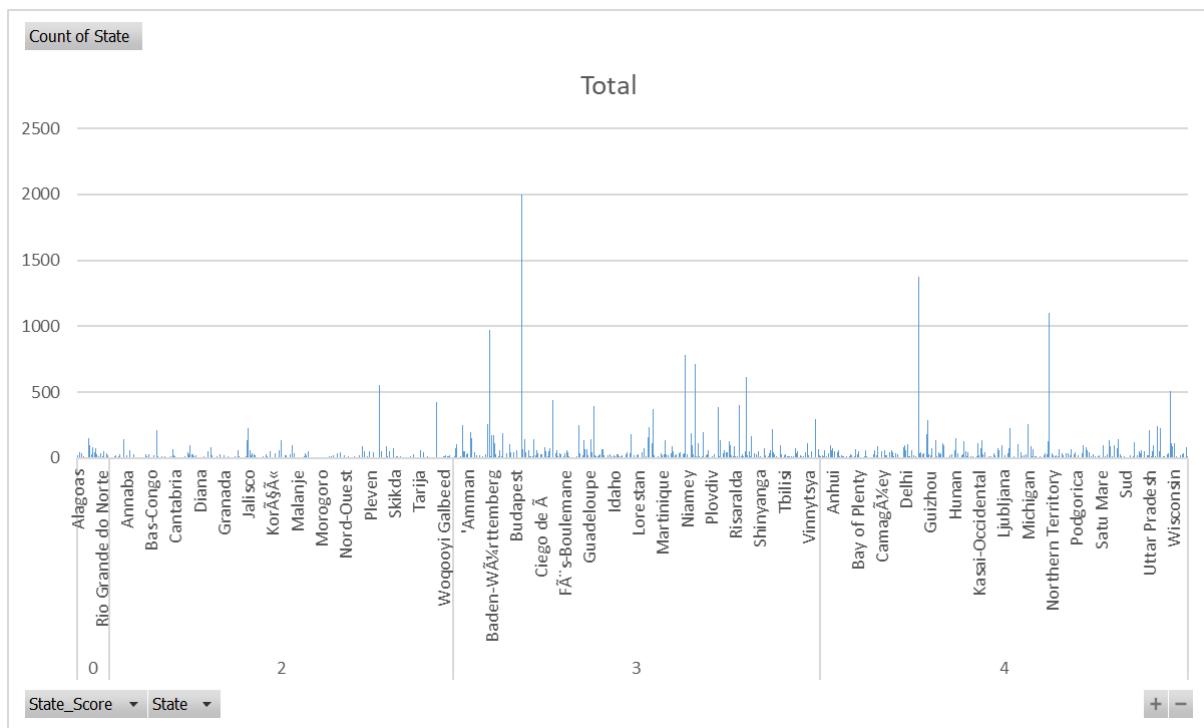


Hình 5.50: Số lượng và tần suất các thành phố theo điểm lợi nhuận (NB-1)

- Thuộc tính State and State\_score:

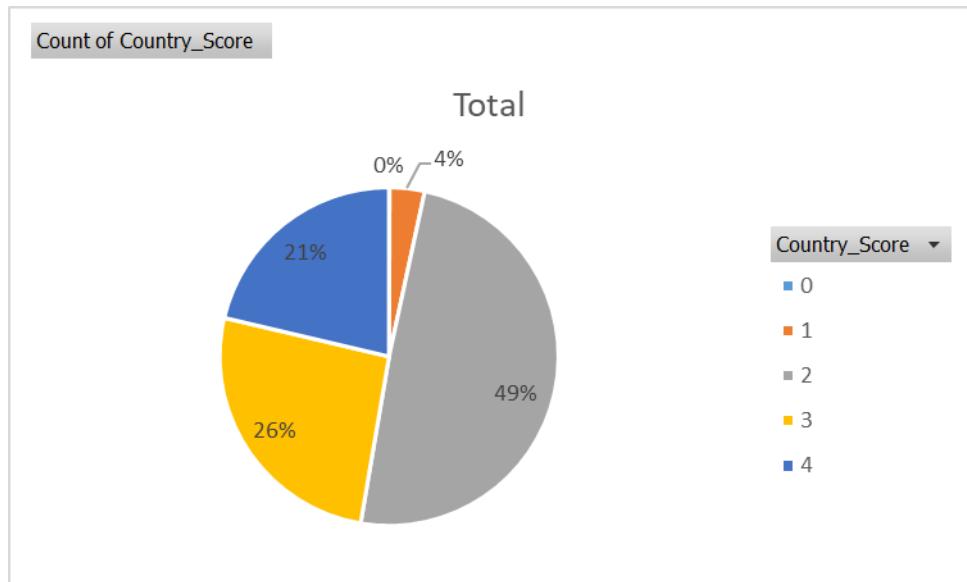


Hình 5.51: Phản trăng của các mức điểm lợi nhuận theo bang (NB-1)

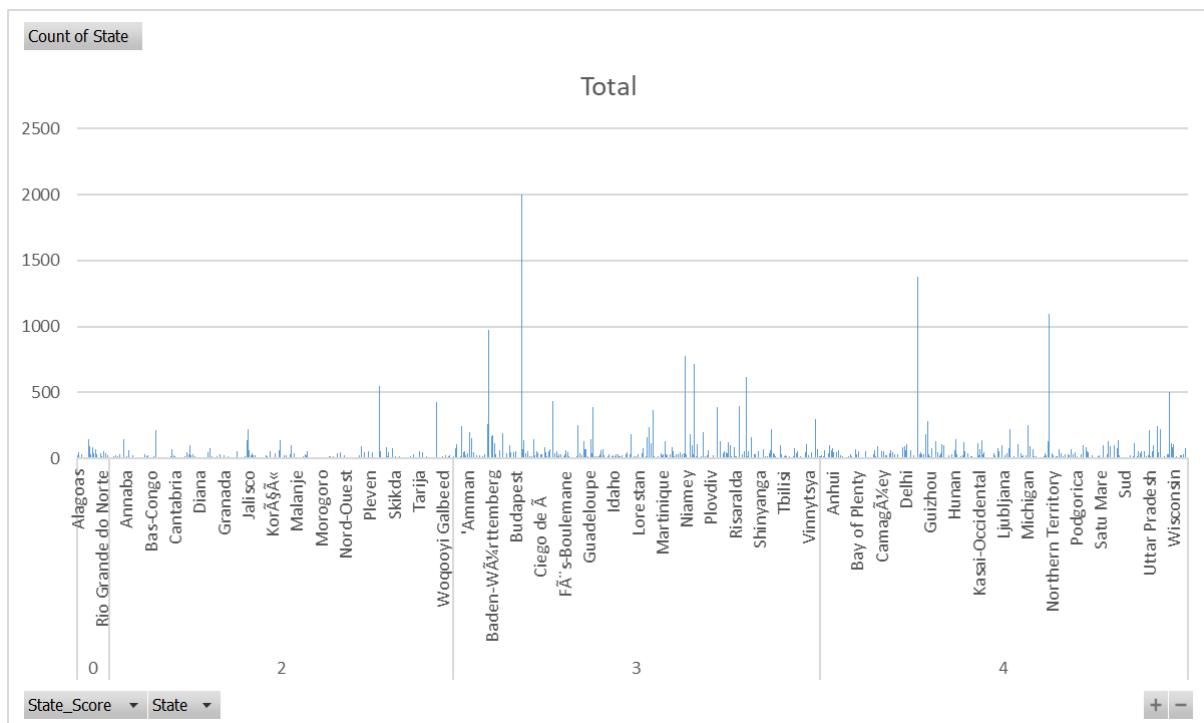


Hình 5.52: Số lượng và tần suất các bang theo điểm lợi nhuận (NB-1)

- Thuộc tính **Country** and **Country\_score**:



Hình 5.53: Phần trăm của các mức điểm lợi nhuận theo quốc gia (NB-1)



Hình 5.54: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (NB-1)

### 5.7.3.3 Nhận xét

Thuật toán Naïve Bayes vẫn cho kết quả tốt và xu hướng vẫn rõ ràng. Xu hướng chung và đặc điểm của các class vẫn nhất quán tuy nhiên vẫn kém hơn các thuật RF, LR, P.

## Class 0: L $\tilde{o}$

- Discount: Đơn hàng có chiết khấu cao (0.4 - 0.7) thường rơi vào lớp lõi.
- Thành phố: 84% thành phố có điểm lợi nhuận âm.
- Bang: 82% bang có điểm lợi nhuận âm.
- Quốc gia: 54% quốc gia có điểm lợi nhuận âm.
- Khu vực: Đặc điểm chung các khu vực trên là thu nhập trung bình đênh thấp.

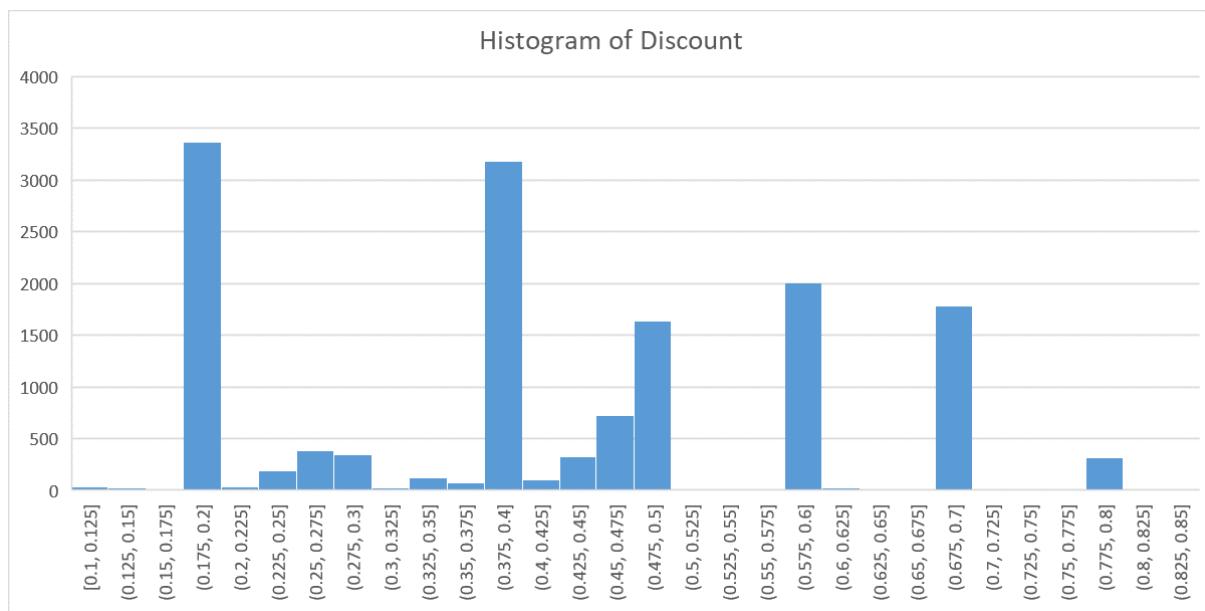
### Class 1: Có lợi nhuận

- Discount: Đơn hàng có chiết khấu rất thấp hoặc gần như bằng 0.
- Thành phố: 98% thành phố có điểm lợi nhuận dương.
- Bang: 98% bang có lợi nhuận dương.
- Quốc gia: 100% quốc gia có lợi nhuận dương, tỷ lệ quốc gia có lợi nhuận âm xấp xỉ 0%.
- Khu vực: Đa dạng ở các khu vực phát triển và ổn định kinh tế.

#### 5.7.4 Thuật toán Perceptron

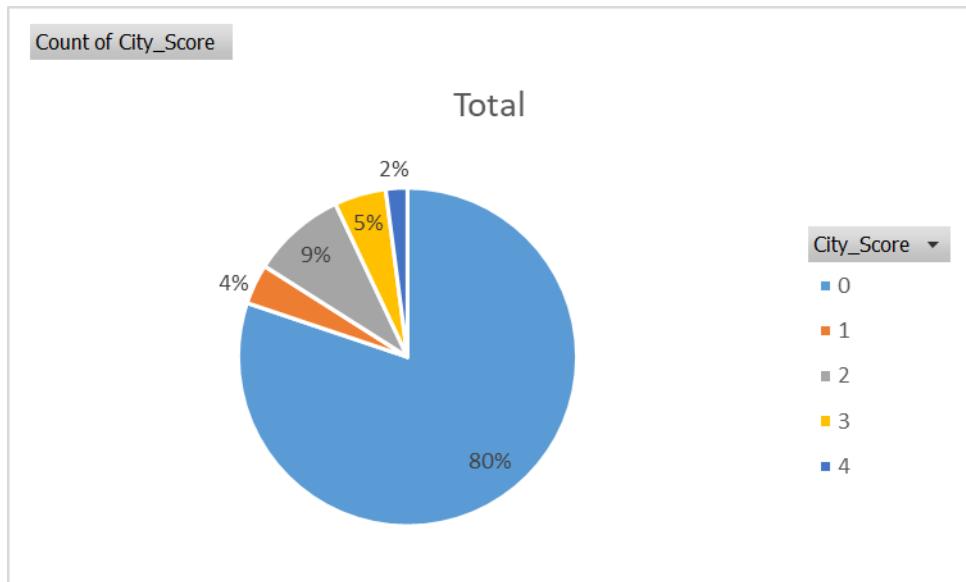
##### 5.7.4.1 Class 0: ( $\tilde{L}$ )

- Thuộc tính **Discount**:

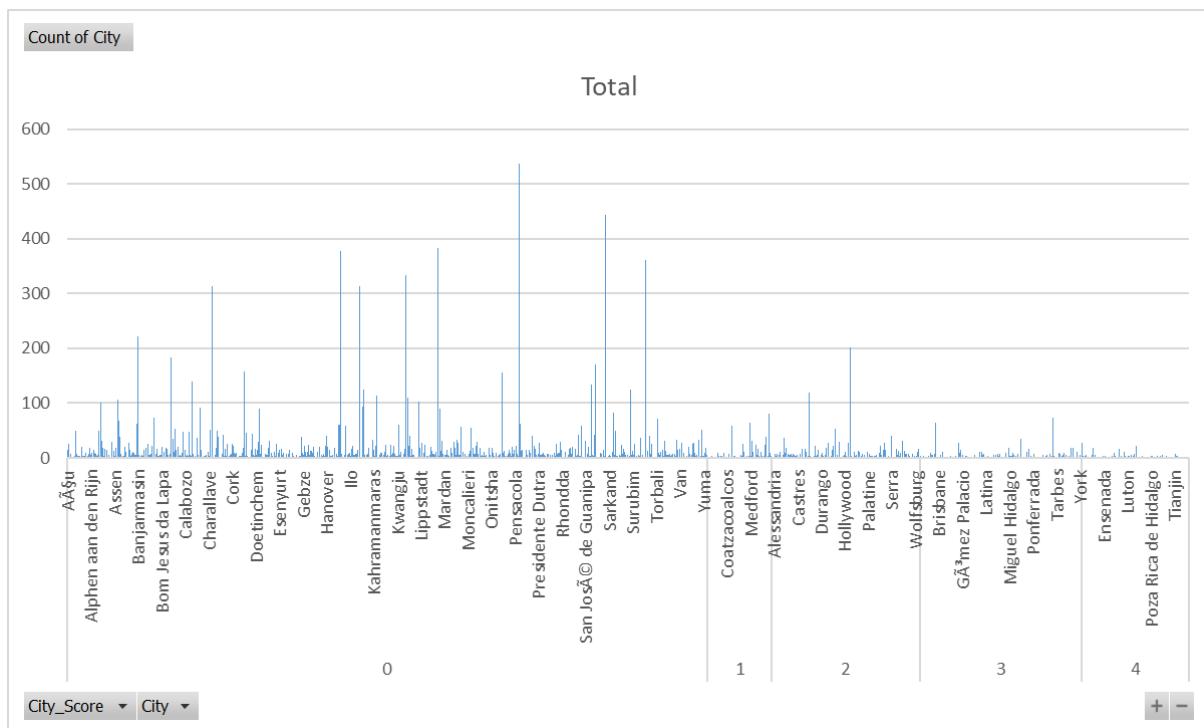


Hình 5.55: Biểu đồ Histogram của thuộc tính Discount (P-0)

- Thuộc tính **City and City\_score**:

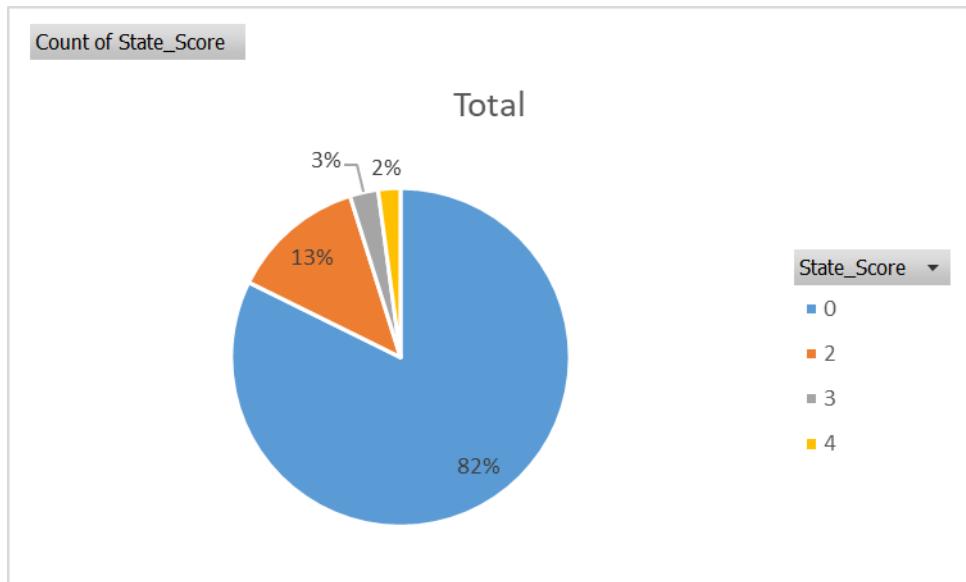


Hình 5.56: Phân trăm của các mức điểm lợi nhuận theo thành phố (P-0)

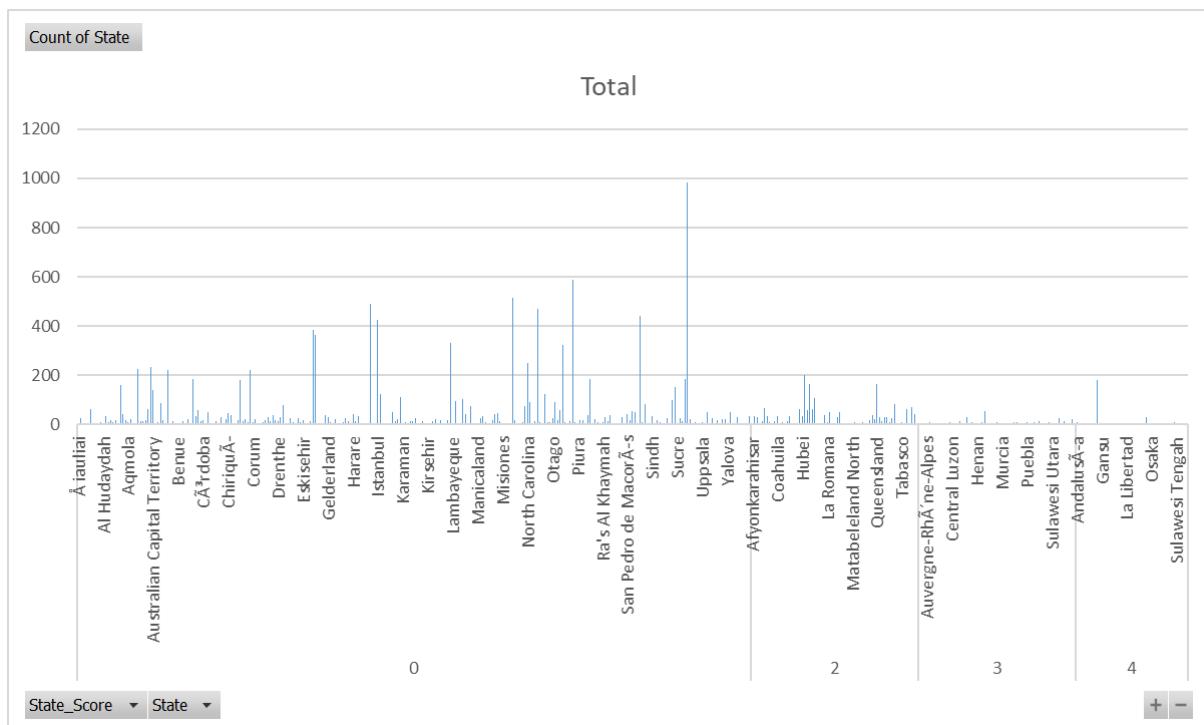


Hình 5.57: Số lượng và tần suất các thành phố theo điểm lợi nhuận (P-0)

- Thuộc tính State and State\_score:

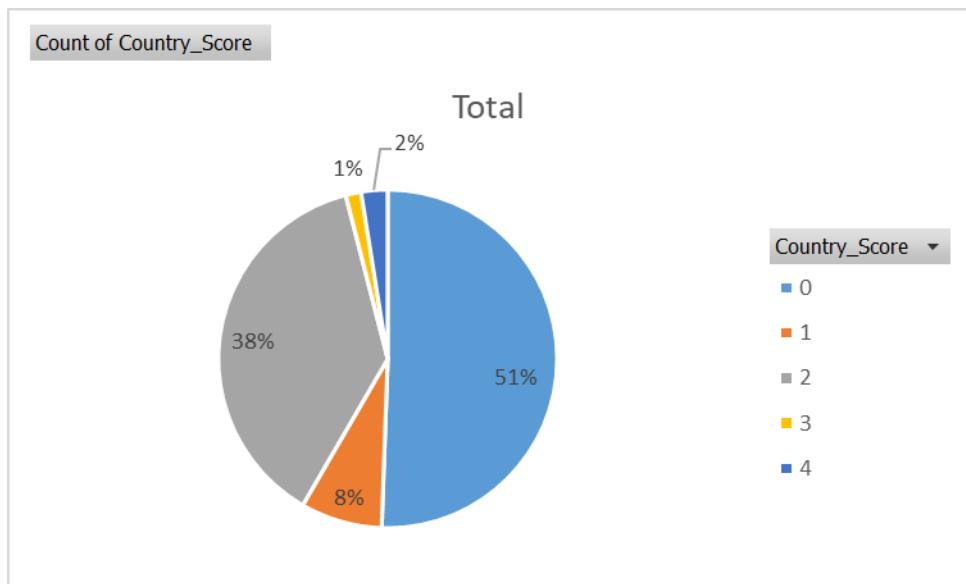


Hình 5.58: Phản trăng của các mức điểm lợi nhuận theo bang (P-0)

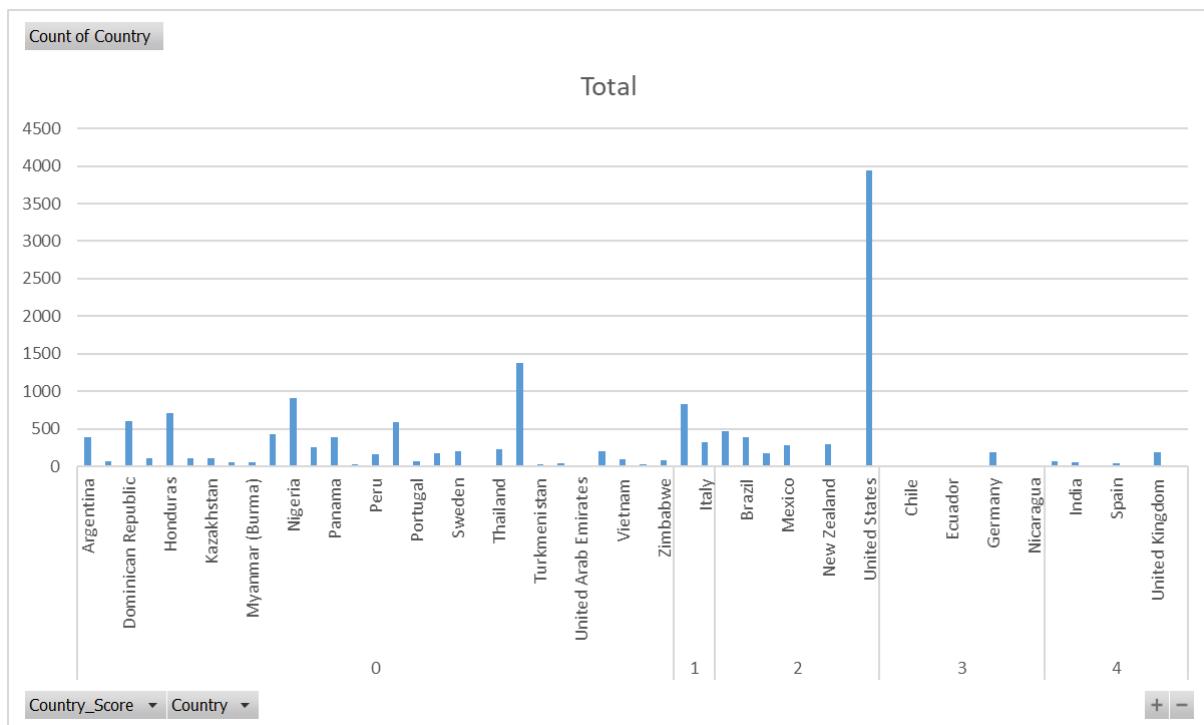


Hình 5.59: Số lượng và tần suất các bang theo điểm lợi nhuận (P-0)

- Thuộc tính Country and Country\_score:



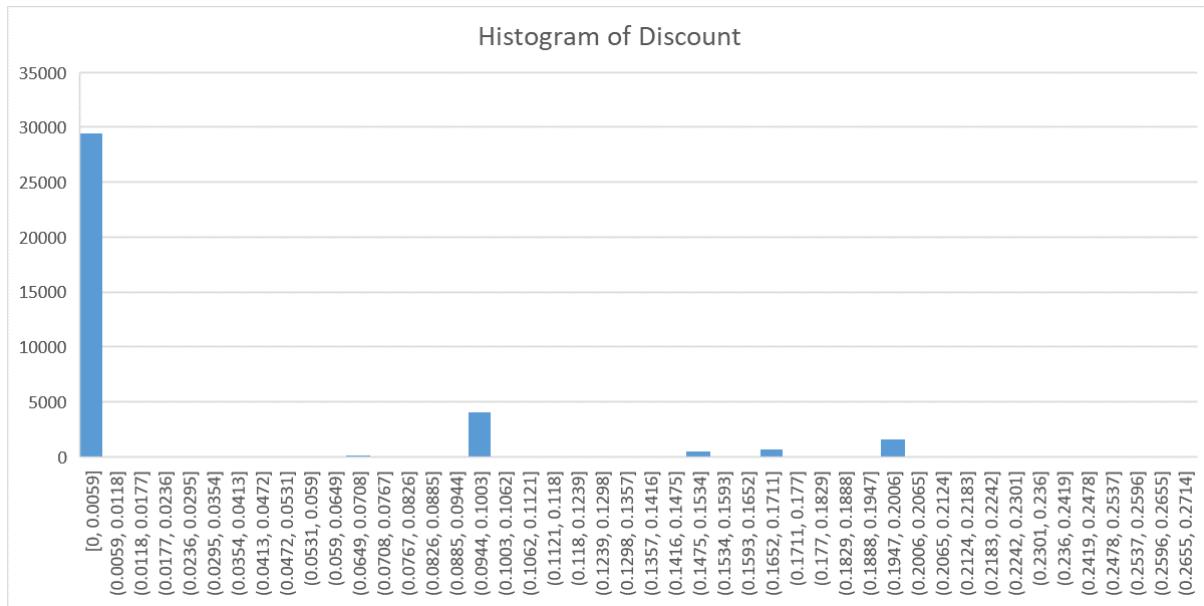
Hình 5.60: Phản trăng của các mức điểm lợi nhuận theo quốc gia (P-0)



Hình 5.61: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (P-0)

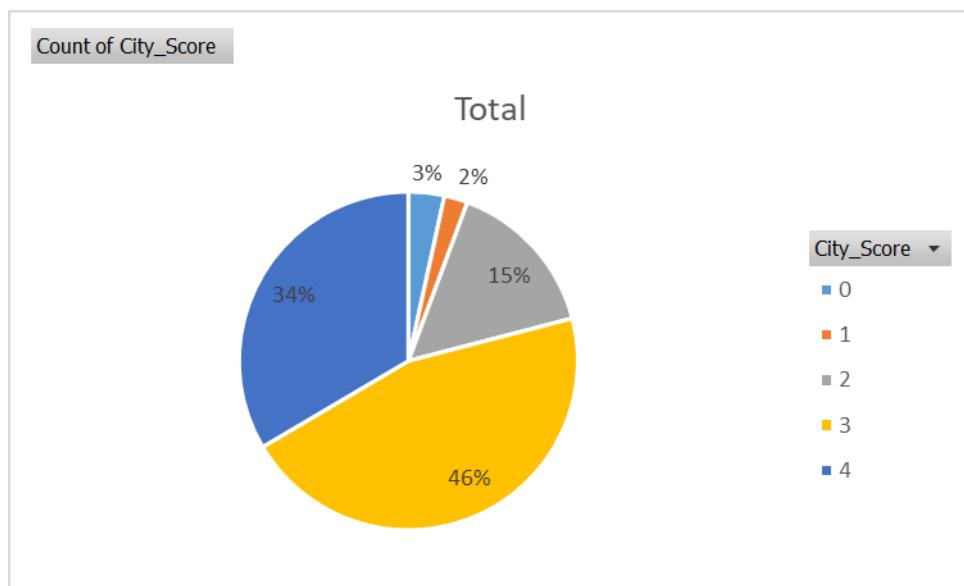
#### 5.7.4.2 Class 1: (Có lợi nhuận)

- Thuộc tính **Discount**:

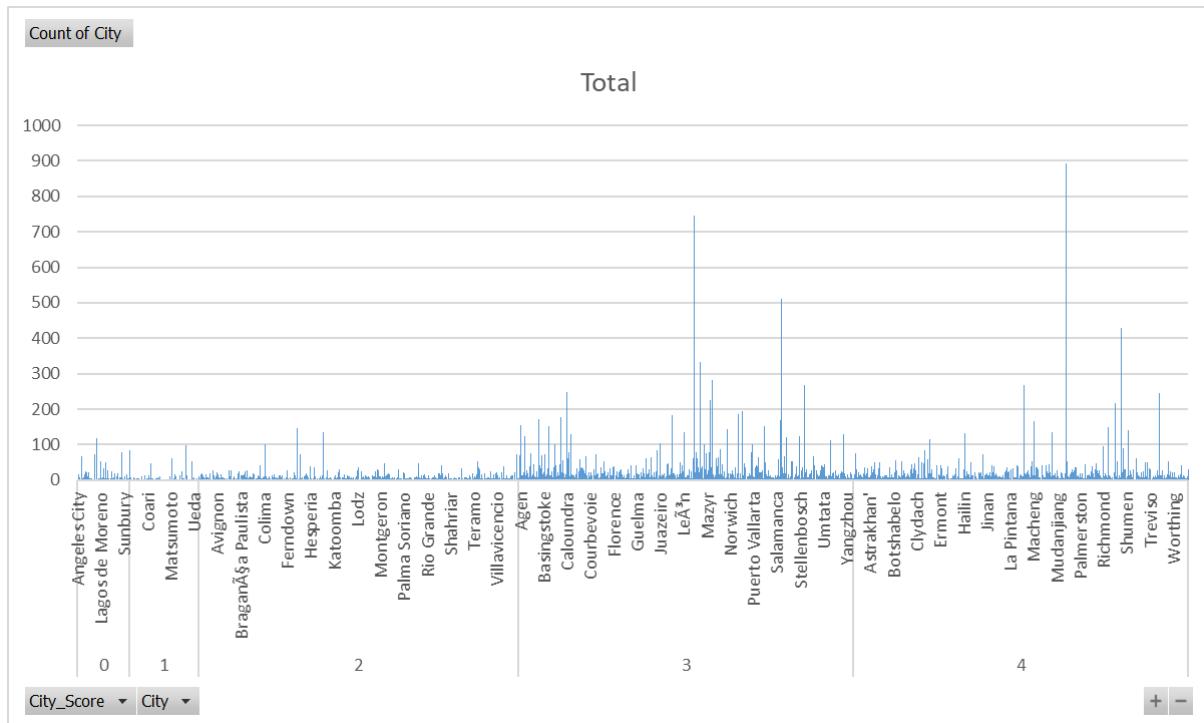


Hình 5.62: Biểu đồ Histogram của thuộc tính Discount (P-1)

- Thuộc tính **City** and **City\_score**:

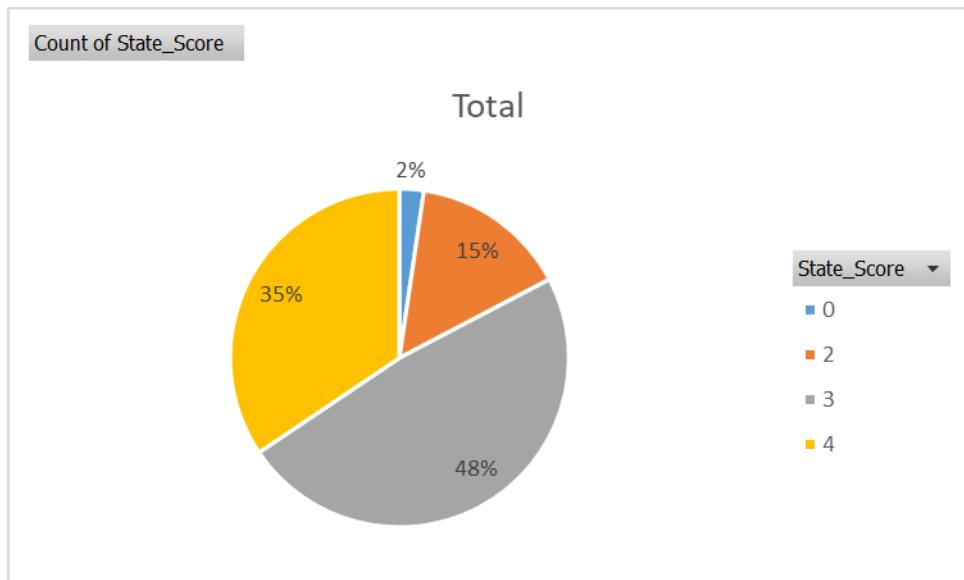


Hình 5.63: Phân trăm của các mức điểm lợi nhuận theo thành phố (P-1)

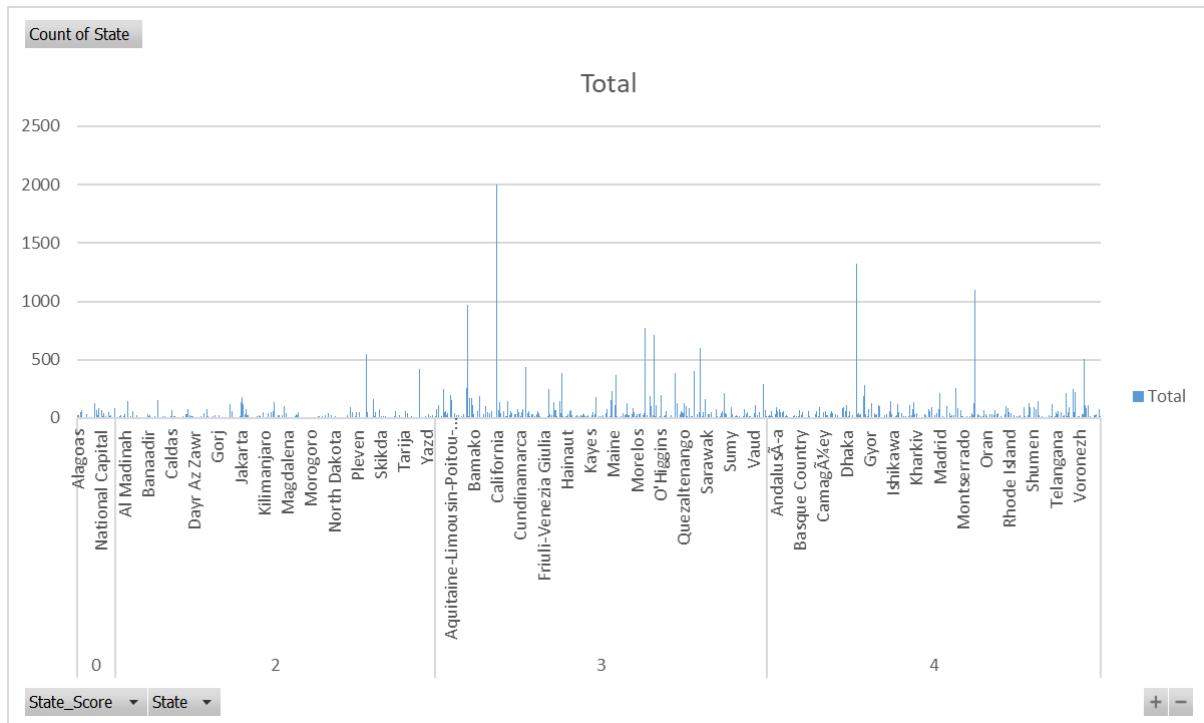


Hình 5.64: Số lượng và tần suất các thành phố theo điểm lợi nhuận (P-1)

- Thuộc tính State and State\_score:

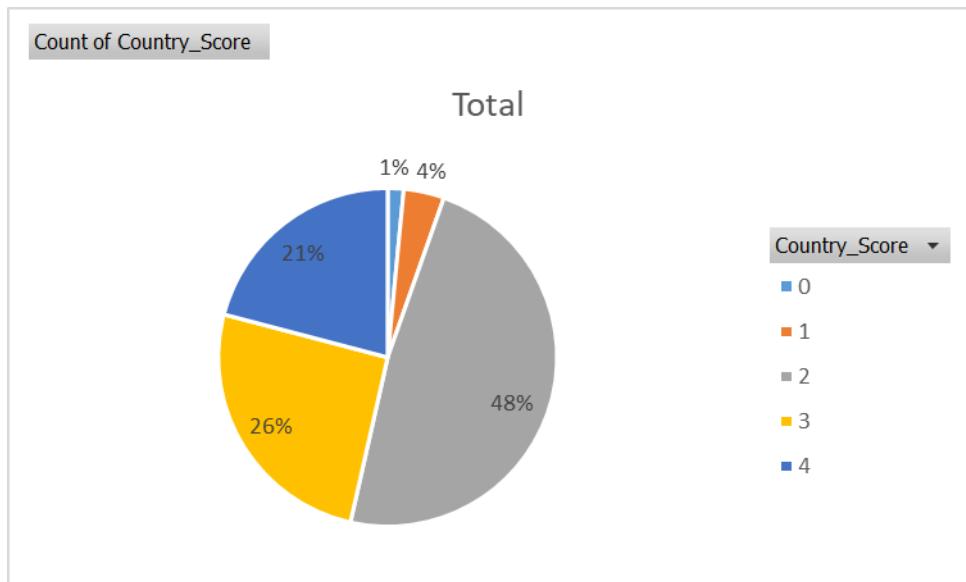


Hình 5.65: Phản trăm của các mức điểm lợi nhuận theo bang (P-1)

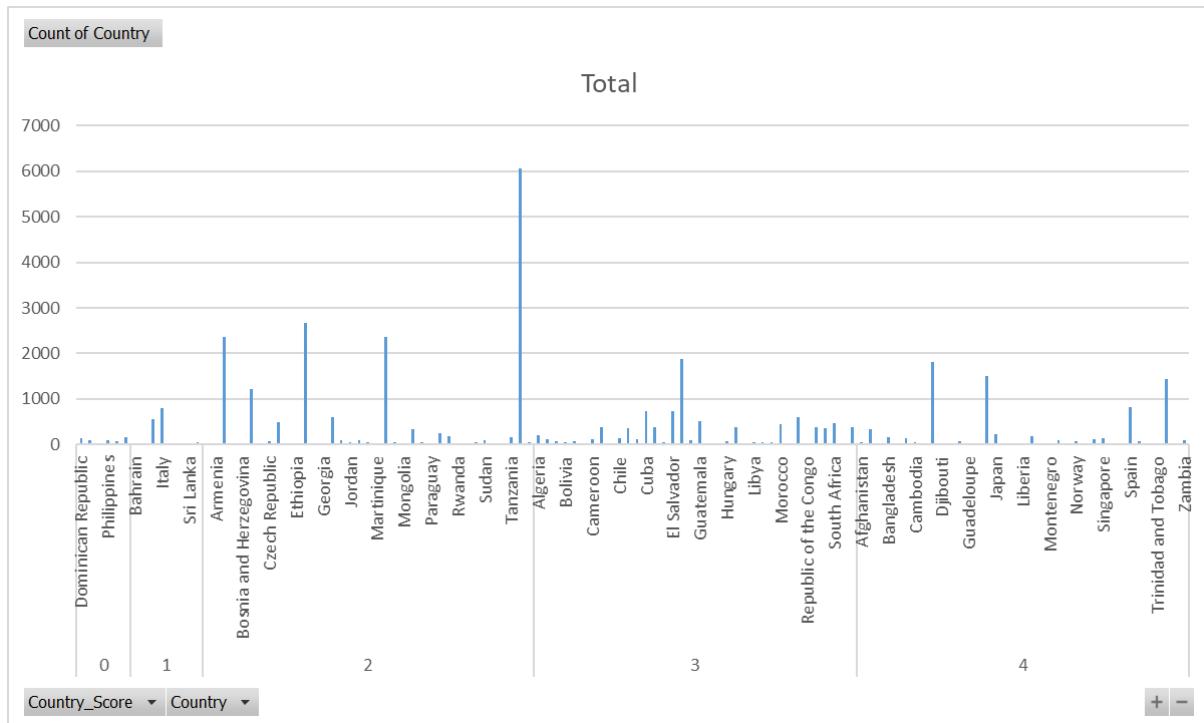


Hình 5.66: Số lượng và tần suất các bang theo điểm lợi nhuận (P-1)

- Thuộc tính **Country** and **Country\_score**:



Hình 5.67: Phản trăm của các mức điểm lợi nhuận theo quốc gia (P-1)



Hình 5.68: Số lượng và tần suất các quốc gia theo điểm lợi nhuận (P-1)

#### 5.7.4.3 Nhận xét

Thuật toán Perceptron cho kết quả tốt và xu hướng vẫn rõ ràng hơn Naïve Bayes tuy nhiên vẫn kém hơn Random Forest và Logistic Regression. Xu hướng chung và đặc điểm của các class vẫn thể hiện rõ ràng.

#### Class 0: Lỗ

- Discount: Đơn hàng có chiết khấu cao (0.2 - 0.8) thường rơi vào lớp lỗ.
- Thành phố: 80% thành phố có điểm lợi nhuận âm.
- Bang: 82% bang có điểm lợi nhuận âm.
- Quốc gia: 51% quốc gia có điểm lợi nhuận âm.
- Khu vực: Đặc điểm chung các khu vực trên là thu nhập trung bình đến thấp.

#### Class 1: Có lợi nhuận

- Discount: Đơn hàng có chiết khấu rất thấp hoặc gần như bằng 0.
- Thành phố: 97% thành phố có điểm lợi nhuận dương.
- Bang: 98% bang có lợi nhuận dương.
- Quốc gia: 99% quốc gia có lợi nhuận dương

- Khu vực: Đa dạng ở các khu vực phát triển và ổn định kinh tế.

## 5.8 Deploy mô hình và thử nghiệm

Trong cả bốn mô hình trên, ta sẽ chọn mô hình có hiệu suất tốt nhất đó là **Random Forest** với độ chính xác 92.08 % để deploy trên ứng dụng Streamlit đơn giản để dự đoán khả năng sinh lợi nhuận của đơn hàng dựa trên 4 đặc trưng quan trọng như giảm giá, thành phố, tiểu bang và quốc gia. Sử dụng mô hình Random Forest đã được huấn luyện trước để thực hiện dự đoán. Bên cạnh đó, ứng dụng cũng cho phép tải lên tập tin CSV để dự đoán hàng loạt và tải xuống kết quả dưới dạng CSV.

### Các thư viện sử dụng:

- streamlit để xây dựng giao diện web.
- pandas để xử lý dữ liệu.
- pickle để tải mô hình đã lưu.

### Bước 1: Chạy ứng dụng bằng câu lệnh sau:

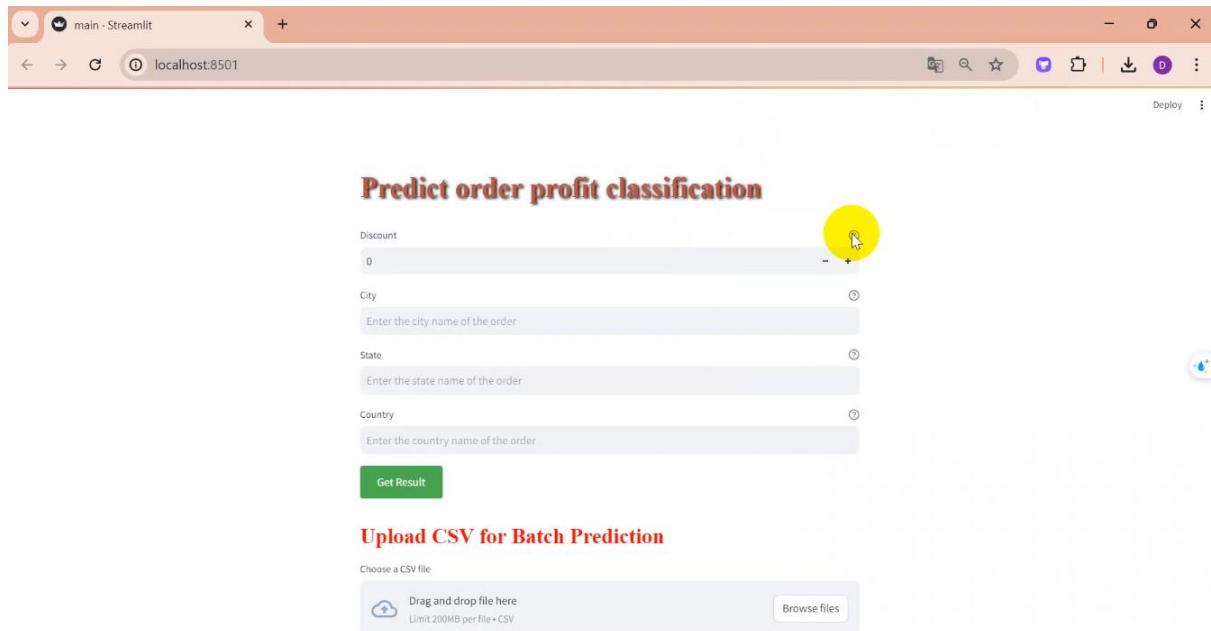
Để khởi chạy ứng dụng, mở terminal và chạy lệnh sau để sử dụng Streamlit và thực thi tập tin main.py:

```
streamlit run main.py
```



Hình 5.69: Mở terminal và chạy lệnh để thực thi ứng dụng

Sau khi chạy lệnh sẽ mở giao diện web trên trình duyệt của người dùng để tương tác với ứng dụng.



**Predict order profit classification**

Discount: 0

City: Enter the city name of the order

State: Enter the state name of the order

Country: Enter the country name of the order

Get Result

**Upload CSV for Batch Prediction**

Choose a CSV file

Drag and drop file here Limit 200MB per file • CSV

Browse files

Hình 5.70: Giao diện web trên trình duyệt của người dùng

## **Bước 2: Dự đoán thủ công**

Nhập thông tin đơn đặt hàng:

- Nhập giảm giá của đơn đặt hàng vào ô "Discount".
- Nhập tên thành phố vào ô "City".
- Nhập tên tiểu bang vào ô "State".
- Nhập tên quốc gia vào ô "Country".



**Predict order profit classification**

Discount: 0.2

City: Los Angeles

State: California

Country: United States

Get Result

Hình 5.71: Nhập thông tin của order

Sau khi điền đầy đủ thông tin và nhấn nút “Get Result”, ứng dụng sẽ hiển thị lại thông tin mà người dùng đã nhập và kết quả dự đoán.

#### User Input The Order Information !

Discount	City	Country	State
0	0.2	Los Angeles	United States

Profit forecast results: The order is profitable

City Los Angeles has a High Profit

State California has a High Profit

Country United States has a Average Profit



Hình 5.72: Hiển thị kết quả dự đoán từ mô hình

Như hình trên, người dùng sẽ thấy kết quả dự đoán lợi nhuận của đơn hàng là “The order is profitable” hoặc “The order is unprofitable”.

#### Bước 3: Dự đoán hàng loạt

Tải lên tập tin CSV:

- Bấm vào nút “Browse files” hoặc drag files trong khu vực tô đỏ như hình bên dưới để chọn tập tin CSV từ máy tính cá nhân.

**Get Result**

User Input The Order Information !

Discount	City	Country	State
0	0.2	Los Angeles	United States

Profit forecast results: The order is profitable

City Los Angeles has a High Profit

State California has a High Profit

Country United States has a Average Profit

**♂ Upload CSV for Batch Prediction**

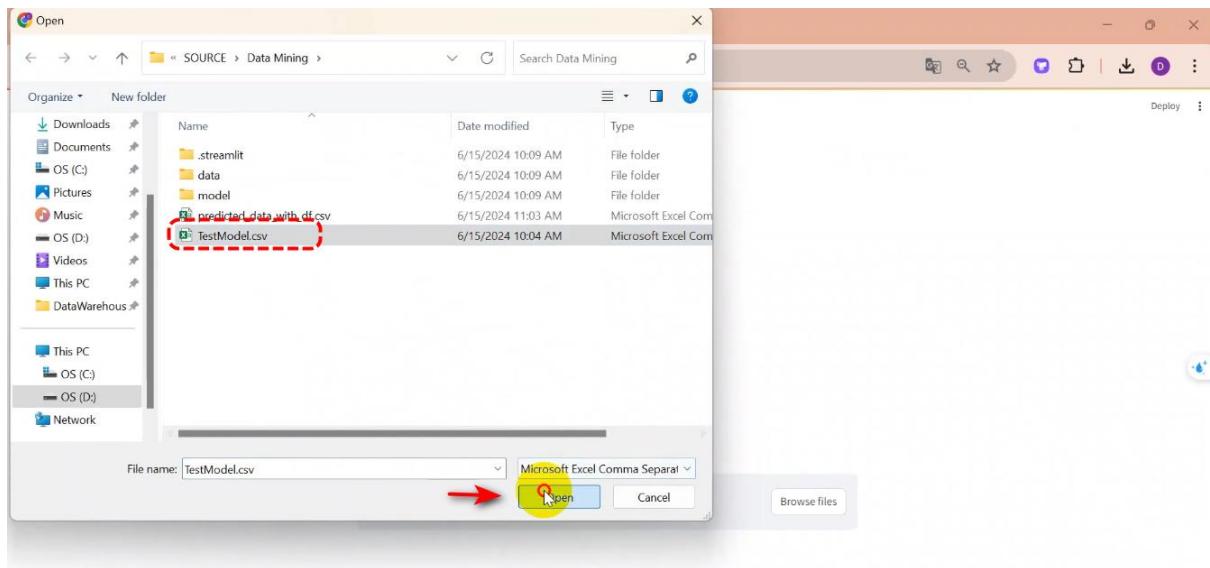
Choose a CSV file

Drag and drop file here  
Limit 200MB per file • CSV

**Browse files**

Hình 5.73: Bấm vào nút “Browse files” hoặc drag files trong khu vực tô đỏ

- Ứng dụng sẽ tự động nhận diện và tải lên tập tin CSV.



Hình 5.74: Chọn file csv cần upload để dự đoán và nhấp Open

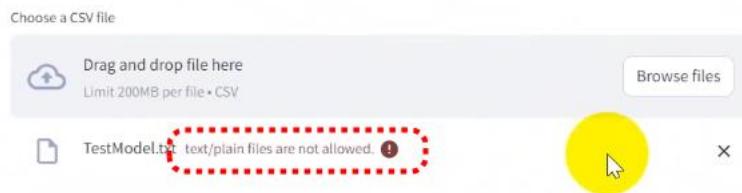
### Upload CSV for Batch Prediction



Hình 5.75: Ứng dụng nhận diện file đúng format và tải lên tập tin CSV thành công

- Nếu tập tin không đúng định dạng hoặc thiếu các cột cần thiết như "Discount", "City", "State", "Country", người dùng sẽ nhận được cảnh báo.

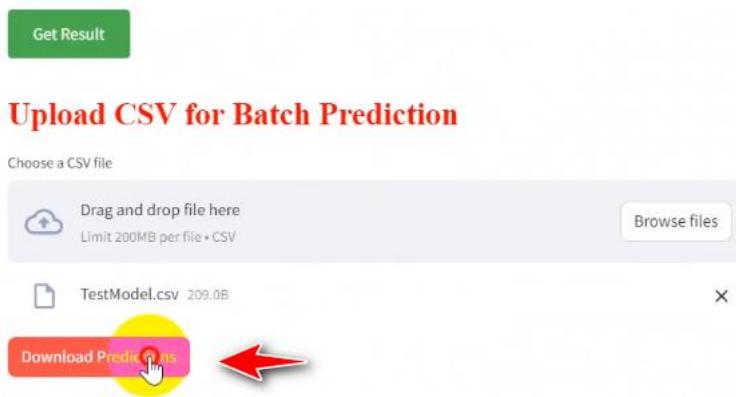
### Upload CSV for Batch Prediction



Hình 5.76: Thông báo lỗi nếu file không đúng định dạng

### Dự đoán và cung cấp kết quả để tải xuống:

- Sau khi tải lên thành công, nhấn button “Download Predictions” để ứng dụng tiến hành dự đoán lợi nhuận cho từng đơn đặt hàng trong tập tin CSV và tải xuống tập tin CSV chứa kết quả dự đoán..



Hình 5.77: Nhấn button “Download Predictions” để tải xuống kết quả dự đoán

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Discount	City_Score	Country_Score	State_Score	Predicted_Label_RF								
2	0.2	0	2	0	Profitable								
3	0.4	0	2	0	Unprofitable								
4	0	3	2	3	Profitable								
5	0.1	4	3	0	Profitable								
6	0.17	0	0	0	Profitable								
7													
8													
9													
10													
11													
12													
13													
14													

Hình 5.78: File csv “ModelPredictions.csv” chứa kết quả dự đoán được tải về

## NGUỒN THAM KHẢO

- [1] R. "Global Superstore Dataset," 03 March 2024. [Online]. Available: <https://www.kaggle.com/datasets/ronysoliman/global-superstore-dataset>. [Accessed 01 May 2024].
- [2] "Visual Studio 2022 community edition – download latest free version," 09 March 2023. [Online]. Available: <https://visualstudio.microsoft.com/vs/community/>. [Accessed 20 April 2024].
- [3] "SQL Server Integration Services Projects - Visual Studio Marketplace," [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=SSIS.SqlServerIntegrationServicesProjects>. [Accessed 23 April 2024].
- [4] "SQL Server downloads," [Online]. Available: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>. [Accessed 21 March 2024].
- [5] "Microsoft Analysis Services Projects 2022 - Visual Studio Marketplace," [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=ProBITools.MicrosoftAnalysisServicesModelingProjects2022>. [Accessed 28 March 2024].
- [6] "Random Forest algorithm in machine learning," [Online]. Available: <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>. [Accessed 01 June 2024].
- [7] A. Saini, "What is logistic regression?," [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/>. [Accessed 01 June 2024].
- [8] IBM, "What is Naïve Bayes," [Online]. Available: <https://www.ibm.com/topics/naive-bayes?fbclid=IwAR3q-I9ss8LXIjJnsITjRXc->

OPKgd2YTGrzwIzZB5OVnwiNY-  
4zTPpEBVX0#:~:text=The%20Na%C3%AFve%20Bayes%20classifier%20is,a%  
20given%20class%20or%20category.

- [9] "Perceptron in Machine Learning - Javatpoint," [Online]. Available: <https://www.javatpoint.com/perceptron-in-machine-learning>. [Accessed 4 June 2024].