

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO ĐÖ ÁN
KHO DỮ LIỆU VÀ OLAP**

ĐỀ TÀI:

PHÂN TÍCH DỮ LIỆU GIÁ CHO THUÊ NHÀ Ở ẨN ĐỘ

GVHD:

Lớp:

Sinh viên thực hiện:

Nguyễn Nhật Phương Huy - 21522156

TP. HCM, ngày 18 tháng 5 năm 2024

MỤC LỤC

| | |
|--|-----------|
| CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI..... | 2 |
| 1.1. Lý do chọn đề tài | 2 |
| 1.2. Nội dung đề tài..... | 2 |
| 1.3. Mô tả dữ liệu | 2 |
| 1.3.1. Thuộc tính của kho dữ liệu gốc..... | 2 |
| 1.3.2. Thuộc tính của kho dữ liệu sau khi tiền xử lý | 3 |
| 1.4. Thiết kế kho dữ liệu | 5 |
| 1.4.1. Lược đồ hình sao..... | 5 |
| 1.4.2. Bảng Fact | 5 |
| 1.4.3. Các bảng chiều | 6 |
| CHƯƠNG 2. XÂY DỰNG KHO DỮ LIỆU (SSIS) | 8 |
| 2.1. Chuẩn bị công cụ | 8 |
| 2.2. Tạo cơ sở dữ liệu và thiết lập kết nối..... | 8 |
| 2.3. Chuẩn bị dữ liệu gốc, import dữ liệu gốc..... | 9 |
| 2.4. Tạo project SSIS trong Visual Studio 2022..... | 13 |
| 2.5. Quá trình tạo các bảng Dimension | 18 |
| 2.5.1. Tạo bảng Dim_Location | 18 |
| 2.5.2. Tạo bảng Dim_Status | 24 |
| 2.5.3. Tạo bảng Dim_Building | 32 |
| 2.5.4. Tạo bảng Dim_Date | 33 |
| 2.6. Quá trình tạo bảng Fact | 42 |
| 2.7. Tạo các ràng buộc khóa ngoại cho bảng Fact | 54 |
| 2.8. Chạy project và kiểm tra dữ liệu | 58 |
| CHƯƠNG 3. PHÂN TÍCH KHO DỮ LIỆU (SSAS)..... | 64 |
| 3.1. Chuẩn bị các công cụ | 64 |
| 3.1.1. Cài đặt Microsoft Analysis Services Projects | 64 |
| 3.1.2. Cài đặt Analysis Services..... | 65 |
| 3.2. Tạo mới project SSAS..... | 67 |
| 3.3. Xác định dữ liệu nguồn (Data Sources)..... | 68 |

| | |
|--|-----------|
| 3.4. Xác định khung nhìn dữ liệu nguồn (Data Source Views) | 70 |
| 3.5. Xây dựng các khối (Cubes) và xác định các độ đo (Measures)..... | 73 |
| 3.6. Xác định các chiều (Dimensions) | 75 |
| 3.6.1. Dim_Date | 75 |
| 3.6.2. Dim_Building | 75 |
| 3.6.3. Dim_Status | 76 |
| 3.6.4. Dim_Location | 76 |
| 3.7. Xác định các độ đo (Measures) | 77 |
| 3.8. Phân cấp trong các bảng chiều | 78 |
| 3.8.1. Phân cấp bảng Dim_Date..... | 78 |
| 3.9. Chạy dữ án SSAS | 83 |
| 3.10. Thực hiện 15 câu truy vấn – Quá trình phân tích dữ liệu bằng thao tác tay trên các khối CUBE | 84 |
| 3.10.1. Câu truy vấn 1: Liệt kê 5 tòa nhà cho thuê có giá cao nhất..... | 84 |
| 3.10.2. Câu truy vấn 2: Cho biết có bao nhiêu nhà cho thuê có 1 phòng tắm và không có nội thất ở Thành phố “Mumbai” | 84 |
| 3.10.3. Câu truy vấn 3: Liệt kê các nhà có giá thuê trên 10000, có 1 phòng tắm và không có nội thất..... | 85 |
| 3.10.4. Câu truy vấn 4: Cho biết 2 giá thuê cao nhất trong số các chung cư được cho thuê trong tháng 6 và ở tầng 2 | 86 |
| 3.10.5. Câu truy vấn 5: Liệt kê giá những căn nhà có diện tích từ “1010 đến 1030” và “3 phòng tắm” | 86 |
| 3.10.6. Câu truy vấn 6: Nhà nào có 1 phòng ngủ, 1 phòng tắm ở thành phố “Bangalore” có giá thấp nhất | 87 |
| 3.10.7. Câu truy vấn 7: Căn nhà nào có giá thuê cao nhất ở thành phố “Mumbai ở tầng trệt” | 88 |
| 3.10.8. Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi” | 89 |
| 3.10.9. Câu truy vấn 9: Ngày nào số lượng đăng nhà cho thuê ít nhất tháng 7 .. | 89 |
| 3.10.10. Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai” | 90 |
| 3.10.11. Câu truy vấn 11: Lấy ra 5 nhà có nội thất với giá cao nhất trong mỗi tháng..... | 91 |

| | |
|--|------------|
| 3.12.12. Câu truy vấn 12: Lấy 5 nhà có giá lớn nhất có Area Locality ở Behala theo từng tháng | 92 |
| 3.12.13. Câu truy vấn 13: Lấy ra những nhà có 4 phòng tắm và có Area Locality bắt đầu bằng chữ “M” | 93 |
| 3.12.14. Câu truy vấn 14: Mỗi thành phố tìm ra 3 nhà có giá cao nhất có nội thất cơ bản | 94 |
| 3.12.15. Câu truy vấn 15: Với mỗi thành phố lấy ra 5 nhà có diện tích lớn nhất có Area Type là “Super Area” | 95 |
| 3.11. Thực hiện 10 câu truy vấn – Quá trình phân tích dữ liệu bằng Pivot Excel . | 95 |
| 3.11.1. Câu truy vấn 1: Liệt kê 5 tòa nhà cho thuê với giá trên 300000..... | 95 |
| 3.11.2. Câu truy vấn 2: Cho biết có bao nhiêu nhà cho thuê có 1 phòng tắm và không có nội thất ở Thành phố “Mumbai” | 97 |
| 3.11.3. Câu truy vấn 3: Liệt kê các nhà có giá thuê trên 10000, có 1 phòng tắm và không có nội thất..... | 98 |
| 3.11.4. Câu truy vấn 4: Cho biết 2 giá thuê cao nhất trong số các chung cư được cho thuê trong tháng 6 và ở tầng 2 | 99 |
| 3.11.5. Câu truy vấn 5: Liệt kê giá những căn nhà có diện tích từ “1010 đến 1030” và “3 phòng tắm” | 100 |
| 3.11.6. Câu truy vấn 6: Nhà nào có 1 phòng ngủ, 1 phòng tắm ở thành phố “Bangalore” có giá thấp nhất | 102 |
| 3.11.7. Câu truy vấn 7: Căn nhà nào có giá thuê cao nhất ở thành phố “Mumbai ở tầng trệt” | 103 |
| 3.11.8. Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi” | 105 |
| 3.11.9. Câu truy vấn 9: Ngày nào số lượng đăng nhà cho thuê ít nhất tháng 7 | 107 |
| 3.11.10. Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai” | 108 |
| 3.12. Thực hiện 15 câu truy vấn – Quá trình phân tích dữ liệu bằng ngôn ngữ truy vấn MDX | 110 |
| 3.12.1. Câu truy vấn 1: Liệt kê 5 tòa nhà cho thuê có giá cao nhất..... | 110 |
| 3.12.2. Câu truy vấn 2: Cho biết có bao nhiêu nhà cho thuê có 1 phòng tắm và không có nội thất ở Thành phố “Mumbai” | 111 |
| 3.12.3. Câu truy vấn 3: Liệt kê các nhà có giá thuê trên 10000, có 1 phòng tắm và không có nội thất..... | 111 |

| | |
|---|-----|
| 3.12.4. Câu truy vấn 4: Cho biết 2 giá thuê cao nhất trong số các chung cư được cho thuê trong tháng 6 và ở tầng 2 | 112 |
| 3.12.5. Câu truy vấn 5: Liệt kê giá những căn nhà có diện tích từ “1010 đến 1030” và “3 phòng tắm” | 112 |
| 3.12.6. Câu truy vấn 6: Nhà nào có 1 phòng ngủ, 1 phòng tắm ở thành phố “Bangalore” có giá thấp nhất | 113 |
| 3.12.7. Câu truy vấn 7: Căn nhà nào có giá thuê cao nhất ở thành phố “Mumbai ở tầng trệt” | 113 |
| 3.12.8. Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi” | 114 |
| 3.12.9. Câu truy vấn 9: Ngày nào số lượng đăng nhà cho thuê ít nhất tháng 7 | 114 |
| 3.12.10. Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai” | 115 |
| 3.12.11. Câu truy vấn 11: Lấy ra 5 nhà có nội thất với giá cao nhất trong mỗi tháng..... | 115 |
| 3.12.12. Câu truy vấn 12: Lấy 5 nhà có giá thuê lớn nhất có Area Locality ở Behala theo từng tháng..... | 116 |
| 3.12.13. Câu truy vấn 13: Lấy ra những nhà có 4 phòng tắm và có Area Locality bắt đầu bằng chữ “M” theo thứ tự tăng dần giá thuê | 117 |
| 3.12.14. Câu truy vấn 14: Mỗi thành phố tìm ra 3 nhà có giá cao nhất có nội thất cơ bản | 118 |
| 3.12.15. Câu truy vấn 15: Với mỗi thành phố lấy ra 5 nhà có diện tích lớn nhất có Area Type là “Super Area” | 118 |
| CHƯƠNG 4. SSRS | 120 |
| 4.1. Report với Visual Studio..... | 120 |
| 4.1.1. Cấu hình Report Server Configuration Manager | 120 |
| 4.1.2. Tạo project SSRS trên Visual Studio | 124 |
| 4.1.3. Report 1: Câu truy vấn 3: Liệt kê các nhà có diện tích trên 600, có 1 phòng tắm và không có nội thất. | 136 |
| 4.1.4. Report 2: Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”..... | 138 |
| 4.1.5. Report 3: Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”..... | 140 |

| | |
|--|------------|
| 4.1.6. Report 4: Câu truy vấn: Top 3 nhà có giá cao nhất có Semi-Furnished ở mỗi thành phố..... | 142 |
| 4.1.7. Report 5: Câu truy vấn: Liệt kê những nhà có 4 phòng tắm và Area Locality bắt đầu bằng chữ M..... | 143 |
| 4.1.8. Triển khai SSRS trên Visual Studio | 143 |
| 4.2. Report với Power BI | 147 |
| 4.2.1. Cài đặt Power BI..... | 147 |
| 4.2.2. Tạo project SSRS trên Power BI..... | 148 |
| 4.2.3. Report 6: Câu truy vấn 3: Liệt kê các nhà có diện tích trên 600, có 1 phòng tắm và không có nội thất. | 151 |
| 4.2.4. Report 7: Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”..... | 154 |
| 4.2.5. Report 8: Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”..... | 156 |
| 4.2.6. Triển khai SSRS trên Power BI | 160 |
| CHƯƠNG 5. DATA MINING..... | 163 |
| 5.1. Tiền xử lý dữ liệu..... | 163 |
| 5.2. Phân tích dữ liệu..... | 165 |
| 5.2.1. Thống kê mô tả..... | 165 |
| 5.2.2. Trực quan hóa dữ liệu | 165 |
| 5.3. Ứng dụng mô hình thuật toán khai thác dữ liệu | 167 |
| 5.3.1. Tiền xử lý dữ liệu trước khi áp dụng các mô hình khai thác dữ liệu..... | 168 |
| 5.3.2. Chia dữ liệu trước khi xây dựng mô hình thuật toán | 169 |
| 5.3.3. Decision Tree | 170 |
| 5.3.4. Random Forest | 175 |
| 5.3.5. Gaussian Naïve Bayes..... | 179 |
| 5.3.6. Bernoulli Naïve Bayes..... | 180 |
| 5.4. Đánh giá thuật toán và dự báo..... | 181 |
| 5.4.1. Các độ đo dùng để đánh giá thuật toán | 181 |
| 5.4.2. Đánh giá, so sánh các mô hình thuật toán dựa trên các độ đo và tiến hành dự báo..... | 183 |
| 5.5. Tập luật cho người dùng cuối | 187 |

| | |
|--|-----|
| DANH MỤC TÀI LIỆU THAM KHẢO | 189 |
| CÁC TÀI LIỆU LIÊN QUAN | 190 |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1. Lý do chọn đề tài

Giá cho thuê nhà phản ánh trực tiếp đến sự biến động trong thị trường bất động sản và cung cấp thông tin về sự đắt đỏ, tính cạnh tranh tại từng phân khúc của việc chọn thuê và cho thuê nhà ở.

Để hiểu rõ hơn về thị trường cho thuê và dự đoán hay lựa chọn được giá thuê phù hợp, nhóm quyết định chọn đề tài này để phân tích.

1.2. Nội dung đề tài

Dataset về dữ liệu tiền thuê nhà được lấy từ trang tổng hợp dataset [Kaggle.com](https://www.kaggle.com).

Bộ dữ liệu này rất phong phú đầy đủ, gồm gần 4700+ căn nhà ở, bao gồm những căn hộ, chung cư và nhà phố được cung cấp để cho thuê, thông qua kho dữ liệu người dùng có thể biết được thông tin về số phòng, giá cho thuê, quy mô bất động sản, số tầng, loại khu vực, địa phương, thành phố, tình trạng nội thất, sở thích người thuê, số lượng phòng tắm và thông tin liên hệ với người cho thuê.

Dataset gồm 12 cột và 4746 hàng dữ liệu, được thu thập vào năm 2022. Kho dữ liệu được xây dựng theo hướng chủ đề: House Rent (Cho thuê nhà).

Link dataset: <https://www.kaggle.com/datasets/iamsouravbanerjee/house-rent-prediction-dataset>

The screenshot shows the Kaggle platform interface. On the left, there's a sidebar with navigation links like 'Create', 'Home', 'Competitions', 'Datasets' (which is selected), 'Models', 'Code', 'Discussions', 'Learn', 'More', 'Your Work', and 'VIEWED'. The main content area has a search bar at the top. Below it, a dataset card for 'House Rent Prediction Dataset' by Sourav Banerjee is displayed. The card includes a thumbnail image of a house, a file size of 84 kB, and download and notebook creation buttons. The dataset title is 'House Rent Prediction Dataset', described as 'Renting Insights: House Rent Prediction Dataset with 4700+ Listings'. Below the title, there are tabs for 'Data Card', 'Code (165)', 'Discussion (7)', and 'Suggestions (0)'. The 'About Dataset' section contains a 'Context' paragraph detailing the diversity of housing options in India, from palaces to modest huts. It also mentions the Human Rights Measurement Initiative and the concept of a gross lease. To the right of the dataset card, there are sections for 'Usability' (10.00), 'License' (Other), 'Expected update frequency' (Never), and 'Tags' (Tabular, Beginner).

1.3. Mô tả dữ liệu

1.3.1. Thuộc tính của kho dữ liệu gốc

| STT | Tên thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|-----|-------------------|--|----------------|
| 1 | Posted On | Ngày đăng | Date |
| 2 | BKH | Số phòng ngủ, sảnh, bếp | Int |
| 3 | Rent | Tiền thuê Nhà/Chung cư/Căn hộ | Decimal(10, 2) |
| 4 | Size | Kích thước của Nhà/Chung cư/Căn hộ bằng Feet vuông | Int |
| 5 | Floor | Nhà/Chung cư/Căn hộ ở tầng nào và tổng số tầng | Varchar |
| 6 | Area Type | Loại diện tích của Nhà/Chung cư/Căn hộ | Varchar |
| 7 | Area Locality | Vị trí của Nhà/Chung cư/Căn hộ | Varchar |
| 8 | City | Thành phố | Varchar |
| 9 | Furnishing Status | Tình trạng nội thất | Varchar |
| 10 | Tenant Preferred | Loại người thuê nhà ưa thích của chủ sở hữu hoặc đại lý | Varchar |
| 11 | Bathroom | Số phòng tắm | Int |
| 12 | Point of Contact | Người mà người thuê cần liên hệ để có thông tin về thuê nhà | Varchar |

1.3.2. Thuộc tính của kho dữ liệu sau khi tiền xử lý

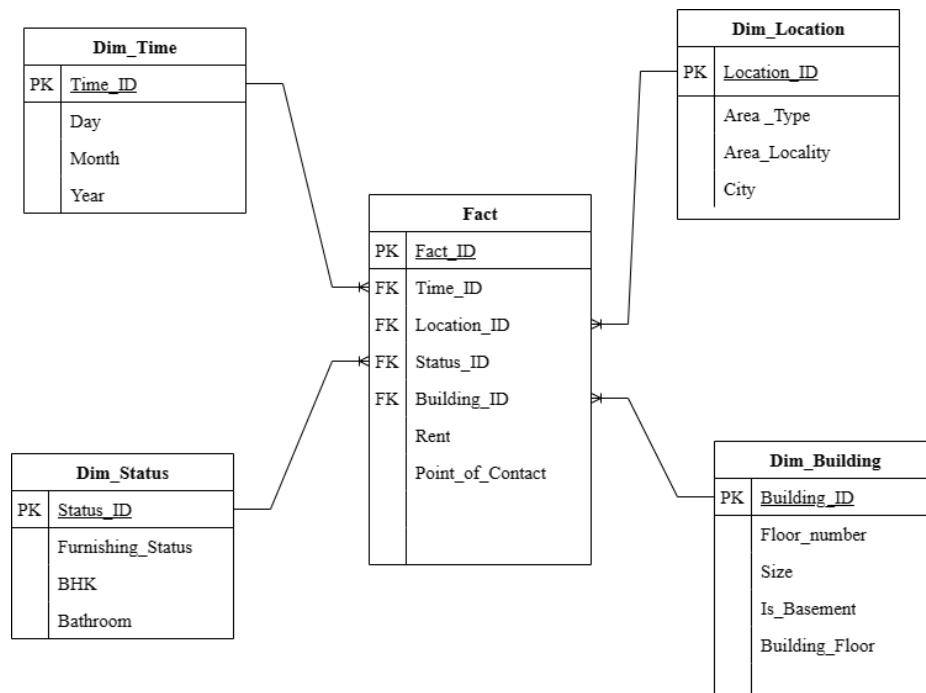
- Bảng cột dữ liệu sau khi tiền xử lý dữ liệu

| STT | Tên thuộc tính | Ý nghĩa | Kiểu dữ liệu |
|-----|-------------------|--|----------------|
| 1 | Day | Ngày đăng | Date |
| 2 | Month | Tháng đăng | Date |
| 3 | Year | Năm đăng | Date |
| 4 | BKH | Số phòng ngủ, sảnh, bếp | Int |
| 5 | Rent | Tiền thuê Nhà/Chung cư/Căn hộ | Decimal(10, 2) |
| 6 | Size | Kích thước của Nhà/Chung cư/Căn hộ bằng Feet vuông | Int |
| 7 | Area_Type | Loại diện tích của Nhà/Chung cư/Căn hộ | Varchar |
| 8 | Area_Locality | Vị trí của Nhà/Chung cư/Căn hộ | Varchar |
| 9 | City | Thành phố | Varchar |
| 10 | Furnishing_Status | Tình trạng nội thất | Varchar |
| 11 | Bathroom | Số phòng tắm | Int |
| 12 | Point_of_Contact | Người mà người thuê cần liên hệ để có thông tin về thuê nhà | Varchar |
| 13 | Floor_number | Nhà/Chung cư/Căn hộ ở tầng nào và tổng số tầng | Int |

| | | | |
|----|-----------------|--------------------------------------|-----|
| 14 | Is_Basement | Tầng hầm | Bit |
| 15 | Building_Floors | Tổng số tầng của Nhà/Chung cư/Căn hộ | Int |

1.4. Thiết kế kho dữ liệu

1.4.1. Lược đồ hình sao



1.4.2. Bảng Fact

| STT | Tên thuộc tính | Kiểu | Ràng buộc | Ý nghĩa |
|-----|----------------|---------|-------------|---------------|
| 1 | Fact_ID | Int | Primary key | Mã Fact |
| 2 | Time_ID | Varchar | Foreign key | Mã ngày |
| 3 | Location_ID | Varchar | Foreign key | Mã vị trí |
| 4 | Status_ID | Varchar | Foreign key | Mã trạng thái |

| | | | | |
|---|------------------|----------------|-------------|---|
| 5 | Building_ID | Varchar | Foreign key | Mã nhà |
| 6 | Rent | Decimal(10, 2) | | Giá cho thuê |
| 7 | Point_of_contact | Varchar | | Người liên hệ (có giá trị là: owner và agent) |

1.4.3. Các bảng chiều

- Bảng Dim_Time

| STT | Tên thuộc tính | Kiểu | Ràng buộc | Ý nghĩa |
|-----|----------------|---------|-------------|---------|
| 1 | Time_ID | Varchar | Primary key | Mã ngày |
| 2 | Day | Int | | Ngày |
| 3 | Month | Int | | Tháng |
| 4 | Year | Int | | Năm |

- Bảng Dim_Location

| STT | Tên thuộc tính | Kiểu | Ràng buộc | Ý nghĩa |
|-----|----------------|---------|-------------|--|
| 1 | Location_ID | Varchar | Primary key | Mã vị trí |
| 2 | Area_Type | Varchar | | Loại diện tích (Giá trị là: SuperArea, CarpetArea) |
| 3 | Area_Locality | Varchar | | Vị trí cụ thể |

| | | | | |
|---|------|---------|--|---|
| 4 | City | Varchar | | Thành phố (Các thành phố: Mumbai, Kolkata, ...) |
|---|------|---------|--|---|

- Bảng Dim_Status

| STT | Tên thuộc tính | Kiểu | Ràng buộc | Ý nghĩa |
|-----|-------------------|---------|-------------|---|
| 1 | Status_ID | Varchar | Primary key | Mã trạng thái |
| 2 | Furnishing_Status | Varchar | | Tình trạng nội thất (gồm: Unfurnished, semi-furnished, furnished) |
| 3 | BKH | Int | | Số phòng ngủ, sảnh, bếp (miền giá trị: 1 – 6) |
| 4 | Bathroom | Int | | Số phòng tắm (miền giá trị: 1 – 10) |

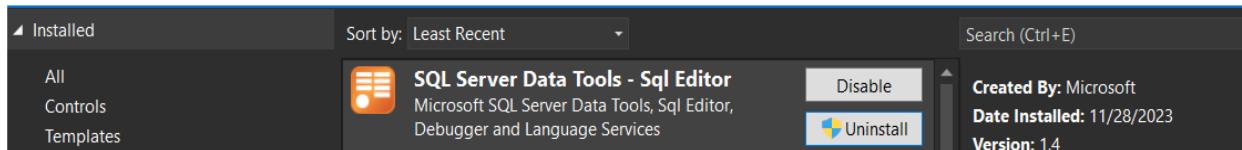
- Bảng Dim_Building

| STT | Tên thuộc tính | Kiểu | Ràng buộc | Ý nghĩa |
|-----|----------------|---------|-------------|--|
| 1 | Building_ID | Varchar | Primary key | Mã nhà |
| 2 | Floor_number | Int | | Nhà/Chung cư/Căn hộ ở tầng nào và tổng số tầng |
| 3 | Is_Basement | Int | | Có tầng hầm không |
| 4 | Building_Floor | Int | | Tổng số tầng của Nhà/Chung cư/Căn hộ |
| 5 | Size | Int | | Kích thước của Nhà/Chung cư/Căn hộ |

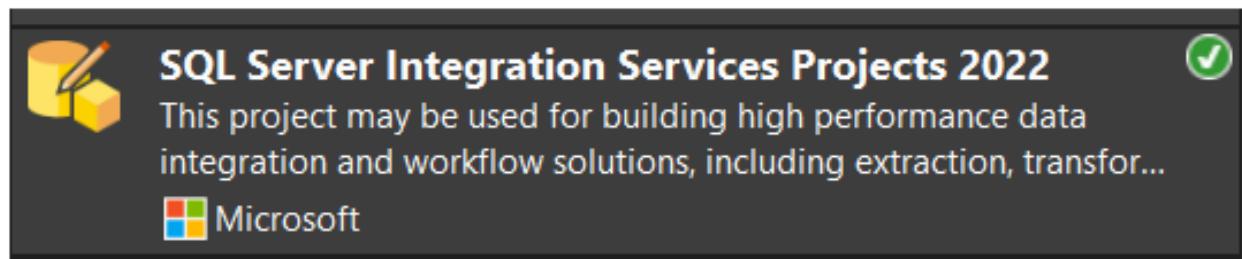
CHƯƠNG 2. XÂY DỰNG KHO DỮ LIỆU (SSIS)

2.1. Chuẩn bị công cụ

- Cài đặt công cụ SQL server Data Tools

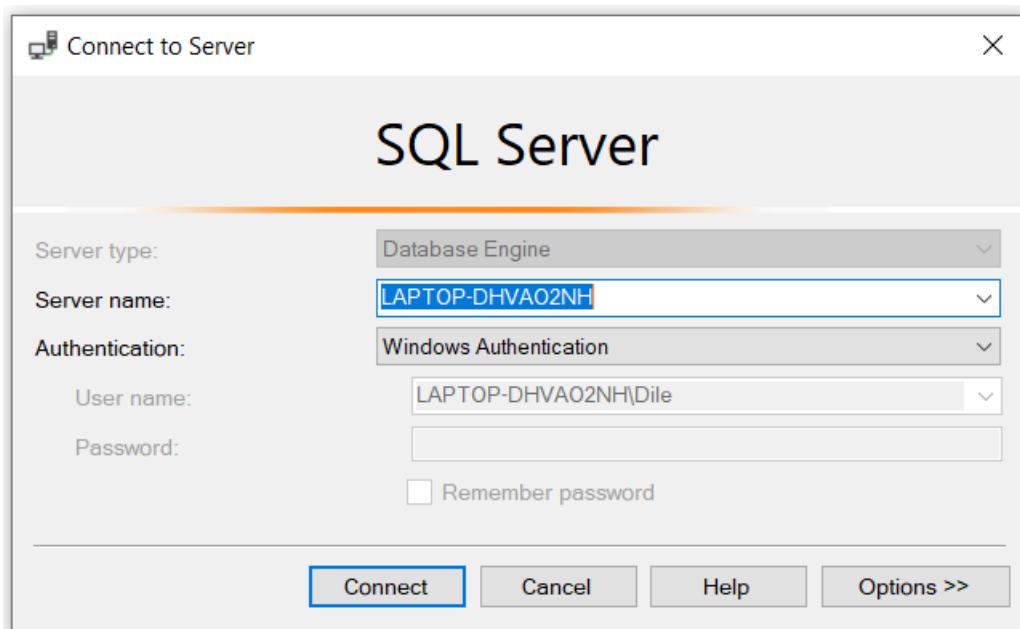


- Cài đặt công cụ cho việc thực hiện SSIS



2.2. Tạo cơ sở dữ liệu và thiết lập kết nối

- Kết nối SQL Server và tạo database

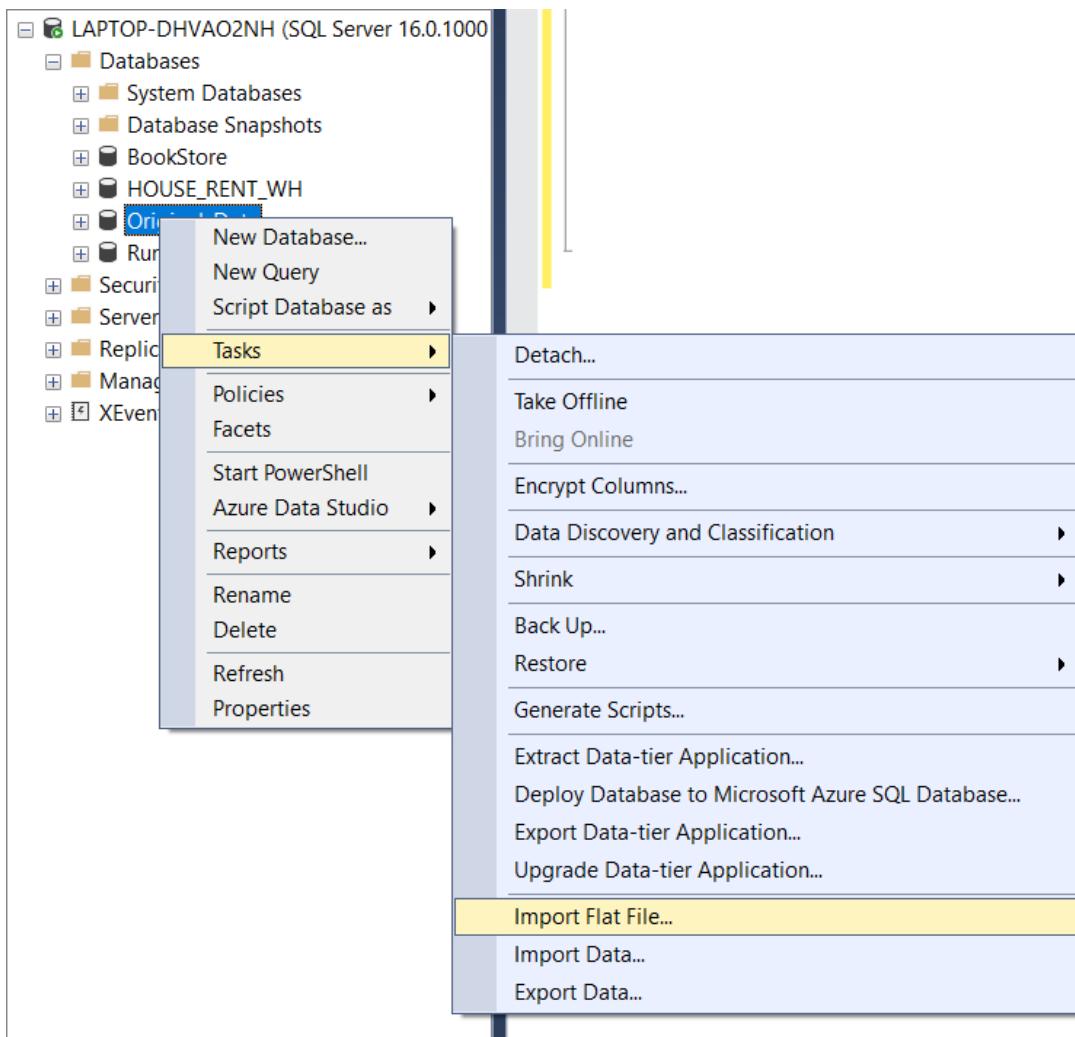


```

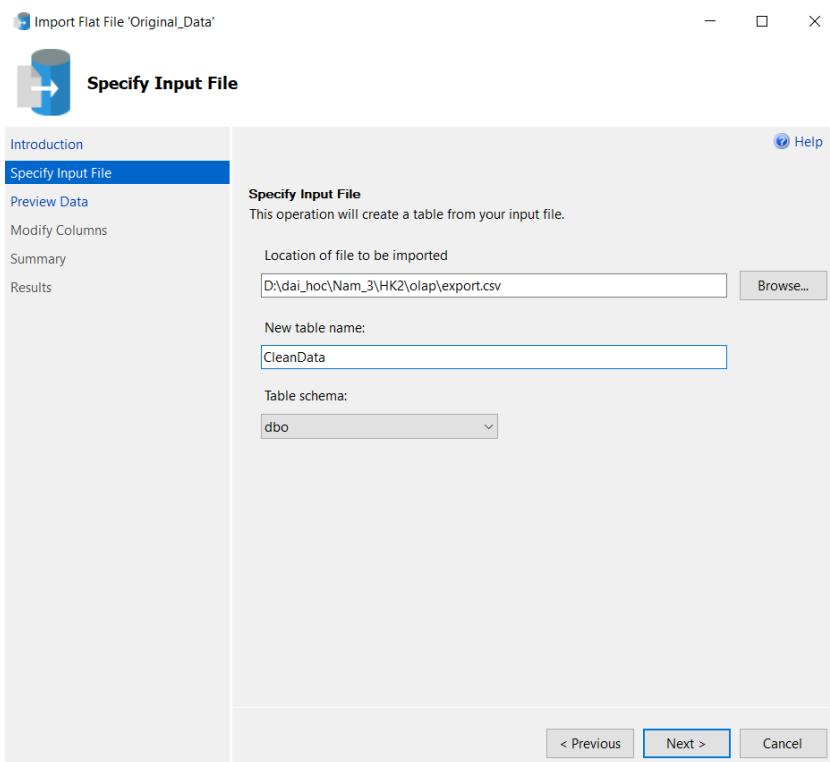
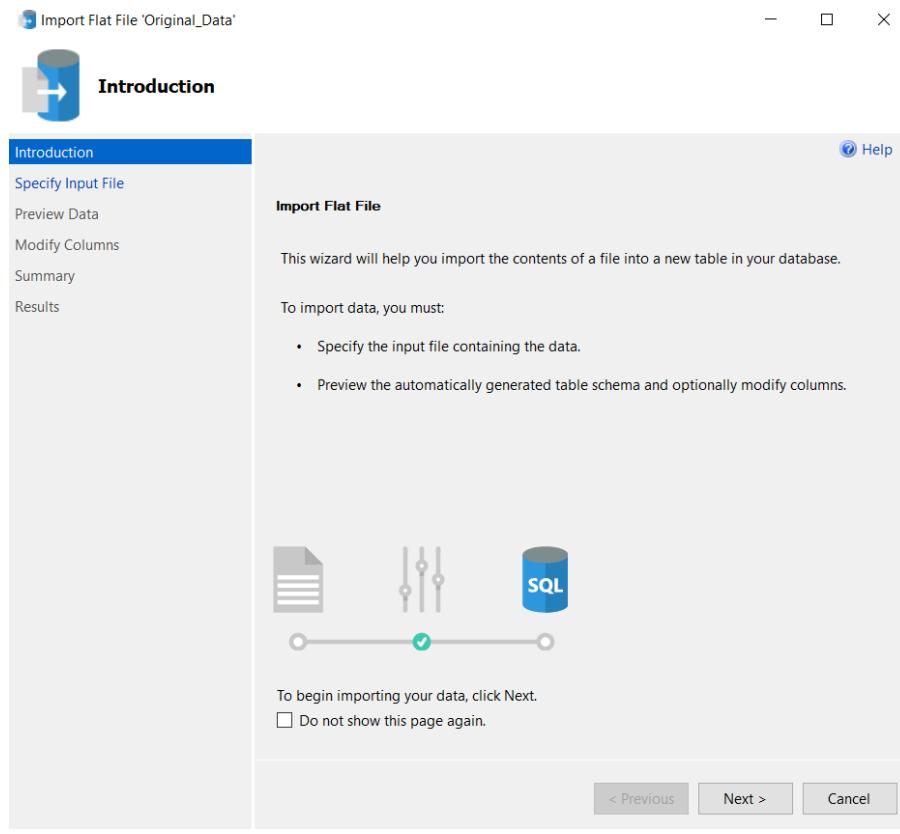
Object Explorer    SQLQuery4.sql - L...HVAO2NH\file (55)   SQLQuery1.sql - L...HVAO2NH\file (55)
Connect  X  Databases  CREATE DATABASE HOUSE_RENT_WH
          System Databases  CREATE DATABASE Original_Data
          Database Snapshots
  
```

2.3. Chuẩn bị dữ liệu gốc, import dữ liệu gốc

- Import Flat File dataset (.csv)



- Original_Data: là database chứa dữ liệu gốc, bao gồm các bảng CleanData chứa các dữ liệu đã được làm sạch.



Import Flat File 'Original_Data'

Preview Data

This operation analyzed the input file structure to generate the preview below for up to the first 50 rows.

| ID | Posted_On | BHK | Rent | Size | Area_Type | A ^ |
|----|-----------|-----|-------|------|-------------|-----|
| 1 | 5/18/2022 | 2 | 10000 | 1100 | Super Area | Ba |
| 2 | 5/13/2022 | 2 | 20000 | 800 | Super Area | Ph |
| 3 | 5/16/2022 | 2 | 17000 | 1000 | Super Area | Sa |
| 4 | 7/4/2022 | 2 | 10000 | 800 | Super Area | Du |
| 5 | 5/9/2022 | 2 | 7500 | 850 | Carpet Area | Sc |
| 6 | 4/29/2022 | 2 | 7000 | 600 | Super Area | Th |
| 7 | 6/21/2022 | 2 | 10000 | 700 | Super Area | Mi |
| 8 | 6/21/2022 | 1 | 5000 | 250 | Super Area | Mi |
| 9 | 6/7/2022 | 2 | 26000 | 800 | Carpet Area | Pa |
| 10 | 6/20/2022 | 2 | 10000 | 1000 | Carpet Area | N |
| 11 | 5/23/2022 | 3 | 25000 | 1200 | Carpet Area | Ac |
| 12 | 6/7/2022 | 1 | 5000 | 400 | Carpet Area | Ke |
| 13 | 5/14/2022 | 4 | 8000 | 700 | Carpet Area | Tc |

Column names changed due to invalid characters, duplication, etc. Column names can be edited in Modify Columns page.

Use Rich Data Type Detection - may provide a closer type fit. However, cells with anomalous values may be dropped.

< Previous Next > Cancel

Import Flat File 'Original_Data'

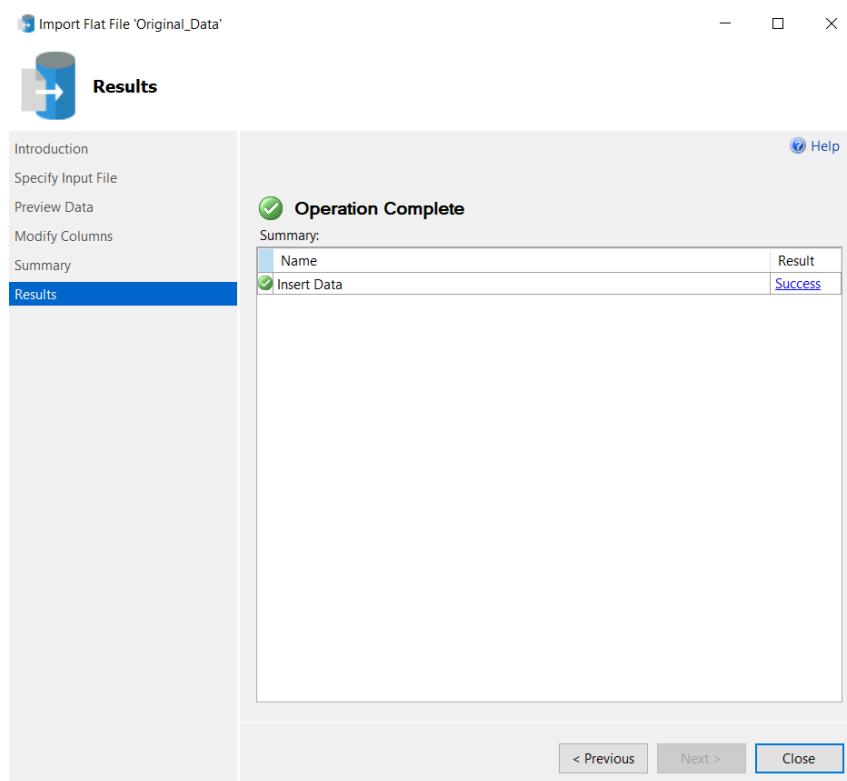
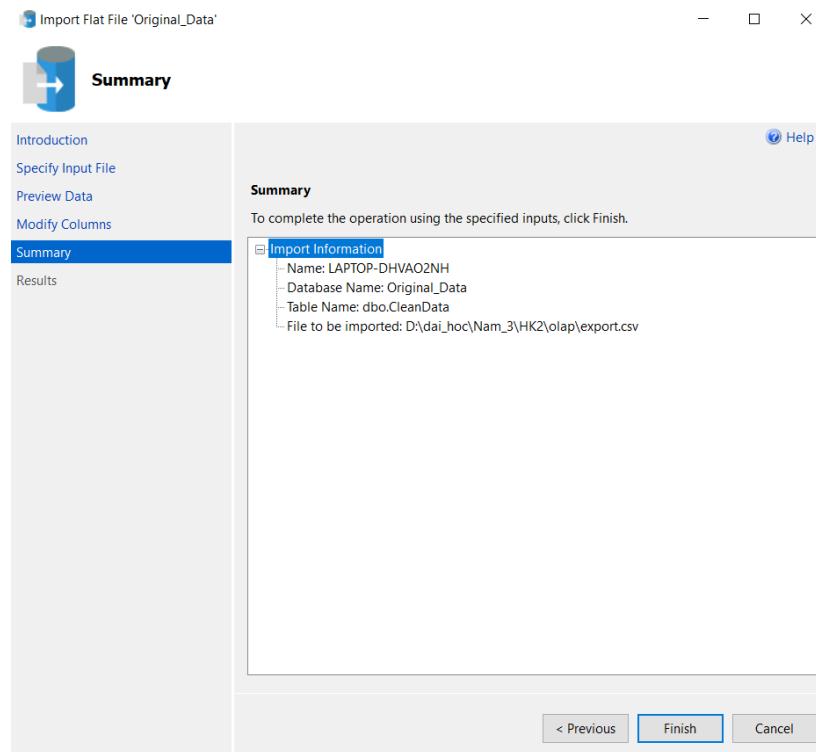
Modify Columns

This operation generated the following table schema. Please verify if schema is accurate, and if not, please make any changes.

| Column Name | Data Type | Primary Key | <input type="checkbox"/> Allow Nulls |
|-------------------|---------------|--------------------------|--------------------------------------|
| ID | smallint | <input type="checkbox"/> | <input type="checkbox"/> |
| Posted_On | date | <input type="checkbox"/> | <input type="checkbox"/> |
| BHK | tinyint | <input type="checkbox"/> | <input type="checkbox"/> |
| Rent | int | <input type="checkbox"/> | <input type="checkbox"/> |
| Size | smallint | <input type="checkbox"/> | <input type="checkbox"/> |
| Area_Type | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| Area_Locality | nvarchar(100) | <input type="checkbox"/> | <input type="checkbox"/> |
| City | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| Furnishing_Status | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| Bathroom | tinyint | <input type="checkbox"/> | <input type="checkbox"/> |
| Point_of_Contact | nvarchar(50) | <input type="checkbox"/> | <input type="checkbox"/> |
| is_basement | bit | <input type="checkbox"/> | <input type="checkbox"/> |
| floor_number | smallint | <input type="checkbox"/> | <input type="checkbox"/> |
| building_floors | tinyint | <input type="checkbox"/> | <input type="checkbox"/> |

Row granularity of error reporting (performance impact with smaller ranges) No Range

< Previous Next > Cancel



- Sau khi import thành công

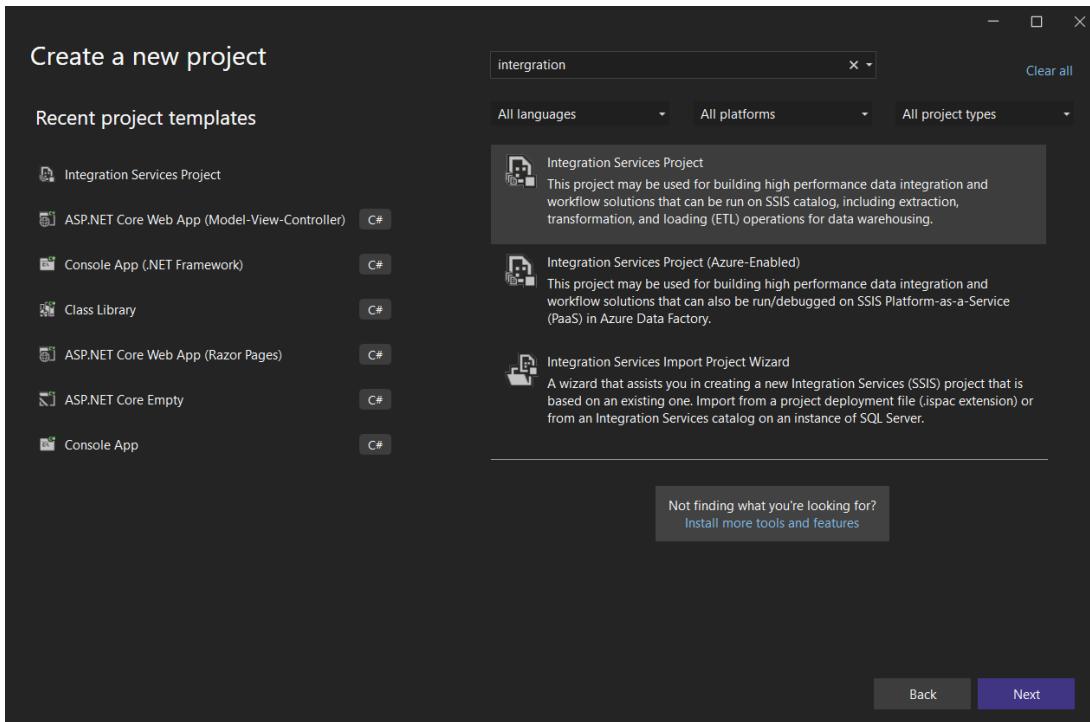
| | ID | Posted_On | BHK | Rent | Size | Area_Type | Area_Locality | City | Furnishing_Status | Bathroom | Point_of_Contact | is_basement | floor_number | building_floors |
|----|----|------------|-----|-------|------|-------------|----------------------------------|---------|-------------------|----------|------------------|-------------|--------------|-----------------|
| 28 | 28 | 2022-06-25 | 2 | 6000 | 1000 | Super Area | Kodalia, Hooghly-Chinsurah | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 0 | 3 |
| 29 | 29 | 2022-06-22 | 2 | 8500 | 800 | Super Area | Baguiati | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 4 | 5 |
| 30 | 30 | 2022-06-25 | 2 | 12500 | 850 | Super Area | Rabindra Sarobar Area, Dhakuria | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 2 |
| 31 | 31 | 2022-05-21 | 1 | 7500 | 350 | Carpet Area | Baghajatin | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 0 | 2 |
| 32 | 32 | 2022-06-26 | 2 | 15000 | 900 | Carpet Area | Project Kaikhali, Vip Road | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 2 |
| 33 | 33 | 2022-06-16 | 2 | 6000 | 550 | Super Area | Vip Road | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 1 |
| 34 | 34 | 2022-06-29 | 2 | 5000 | 500 | Carpet Area | Baruipur | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 2 |
| 35 | 35 | 2022-05-10 | 3 | 22000 | 1100 | Carpet Area | Dum Dum Park | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 2 | 3 |
| 36 | 36 | 2022-05-12 | 2 | 15000 | 850 | Carpet Area | Sreebhumi | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 2 |
| 37 | 37 | 2022-06-03 | 2 | 12500 | 800 | Carpet Area | Shyam Bazar | Kolkata | Unfurnished | 1 | Contact Agent | 0 | 2 | 2 |
| 38 | 38 | 2022-05-31 | 2 | 7000 | 630 | Carpet Area | Birati | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 0 | 3 |
| 39 | 39 | 2022-06-10 | 2 | 21000 | 900 | Carpet Area | Bansdroni | Kolkata | Semi-Furnished | 2 | Contact Owner | 0 | 1 | 2 |
| 40 | 40 | 2022-06-24 | 1 | 10000 | 270 | Carpet Area | Jadavpur University | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 2 |
| 41 | 41 | 2022-06-29 | 2 | 7200 | 630 | Carpet Area | Birati | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 0 | 1 |
| 42 | 42 | 2022-05-12 | 3 | 12000 | 1500 | Carpet Area | Bhadrakali | Kolkata | Furnished | 2 | Contact Owner | 0 | 0 | 2 |
| 43 | 43 | 2022-06-26 | 1 | 5000 | 600 | Super Area | Behala | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 2 | 5 |
| 44 | 44 | 2022-04-30 | 2 | 8500 | 700 | Carpet Area | Baruipur | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 4 | 14 |
| 45 | 45 | 2022-05-27 | 1 | 6000 | 300 | Super Area | Ballygunge | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 0 | 3 |
| 46 | 46 | 2022-06-06 | 2 | 10000 | 1300 | Super Area | Kalikapur | Kolkata | Semi-Furnished | 2 | Contact Owner | 0 | 0 | 1 |
| 47 | 47 | 2022-05-20 | 2 | 4600 | 400 | Carpet Area | Behala | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 3 | 3 |
| 48 | 48 | 2022-07-02 | 2 | 12000 | 650 | Super Area | Baishnabghata Patuli Township... | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 1 |
| 49 | 49 | 2022-05-26 | 3 | 30000 | 2000 | Super Area | Behala | Kolkata | Semi-Furnished | 2 | Contact Owner | 0 | 0 | 2 |
| 50 | 50 | 2022-05-12 | 3 | 15000 | 1068 | Super Area | Bansdroni | Kolkata | Semi-Furnished | 2 | Contact Owner | 0 | 0 | 3 |
| 51 | 51 | 2022-06-22 | 1 | 7500 | 150 | Super Area | Salt Lake City | Kolkata | Furnished | 1 | Contact Owner | 0 | 1 | 1 |
| 52 | 52 | 2022-06-25 | 1 | 7500 | 450 | Carpet Area | New Town | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 2 | 2 |
| 53 | 53 | 2022-07-03 | 2 | 5000 | 800 | Carpet Area | Baruipur | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 0 | 1 |
| 54 | 54 | 2022-04-23 | 2 | 15000 | 1000 | Super Area | Bansdroni | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 2 |
| 55 | 55 | 2022-05-14 | 2 | 11000 | 900 | Super Area | Pancha Sayar | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 5 | 5 |

Query executed successfully.

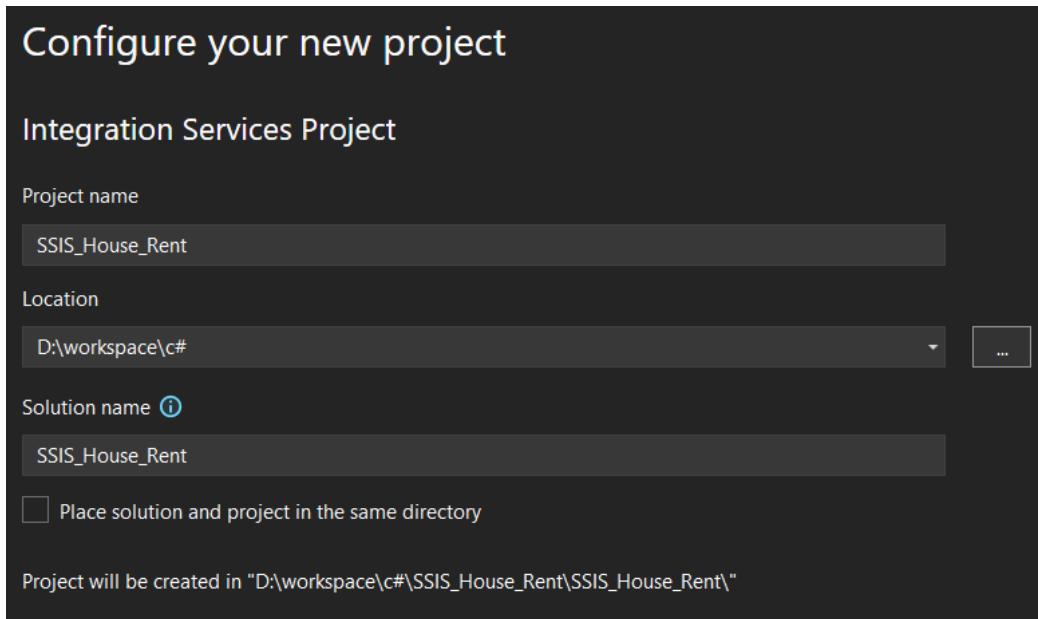
LAPTOP-DHVAO2NH (16.0 RTM) | LAPTOP-DHVAO2N

2.4. Tạo project SSIS trong Visual Studio 2022

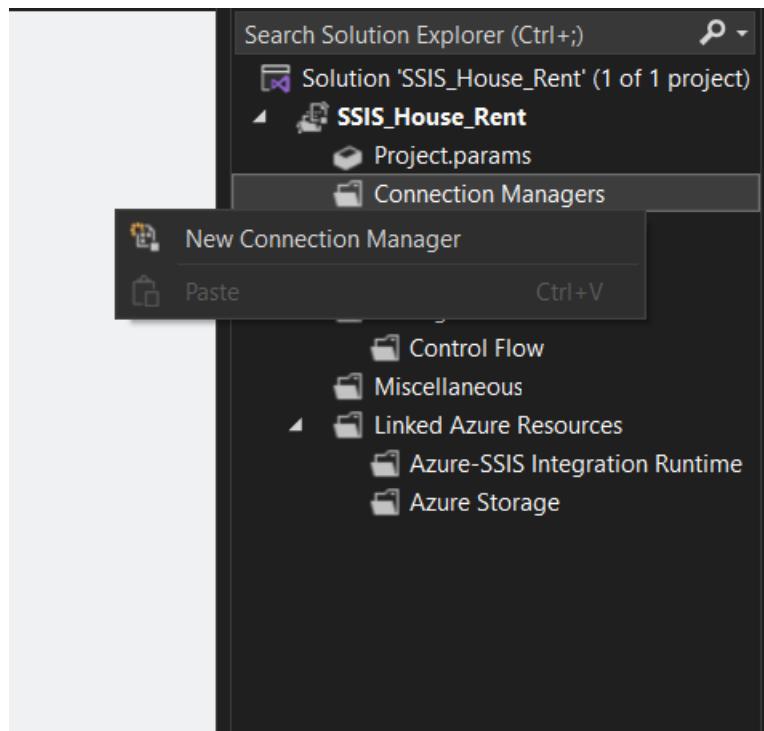
- Chọn File - New project – Hộp thoại sẽ hiển thị ra nhu hình và chọn Integration Services Project.



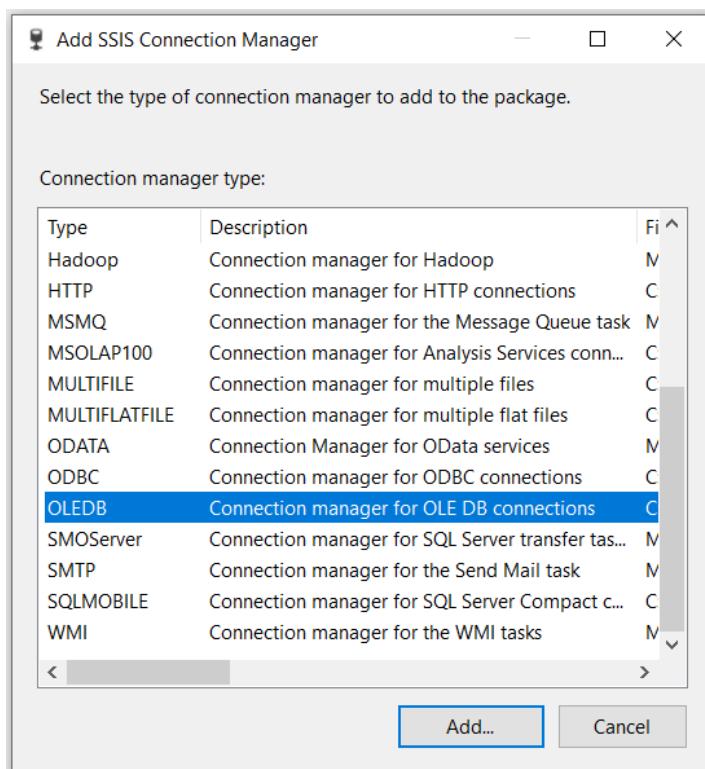
- Đặt tên cho project

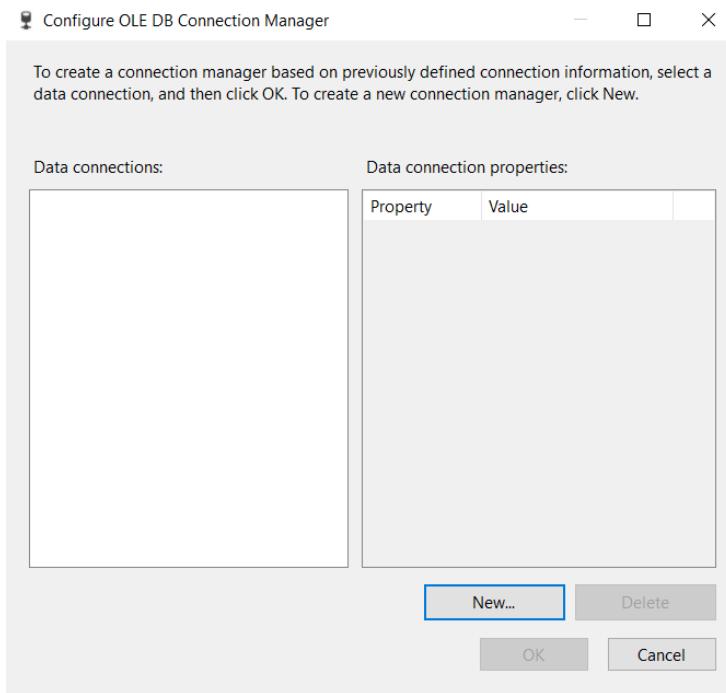


- Tạo kết nối đến database Original_Data

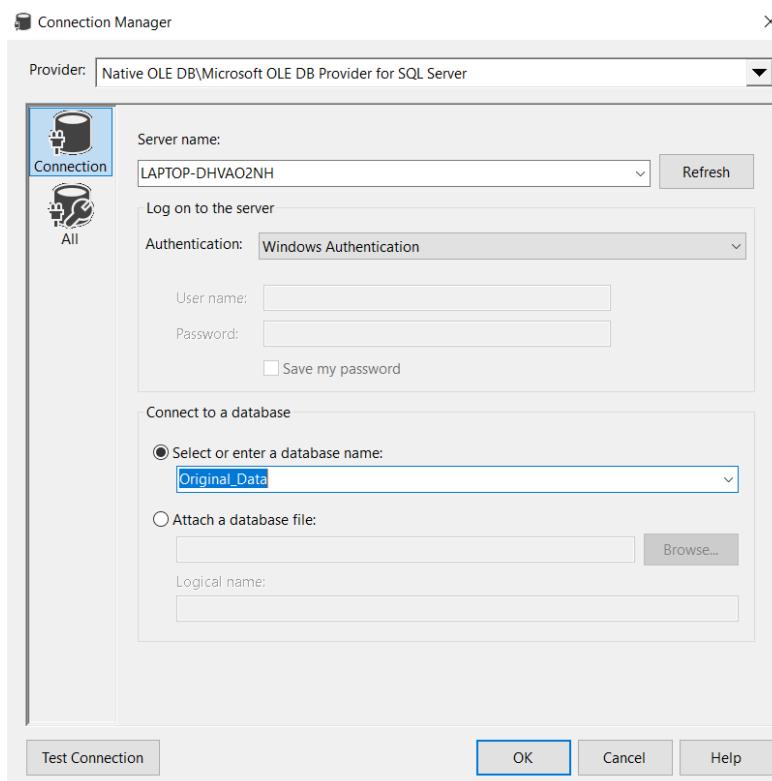


- Chọn OLEDB và nhấp New

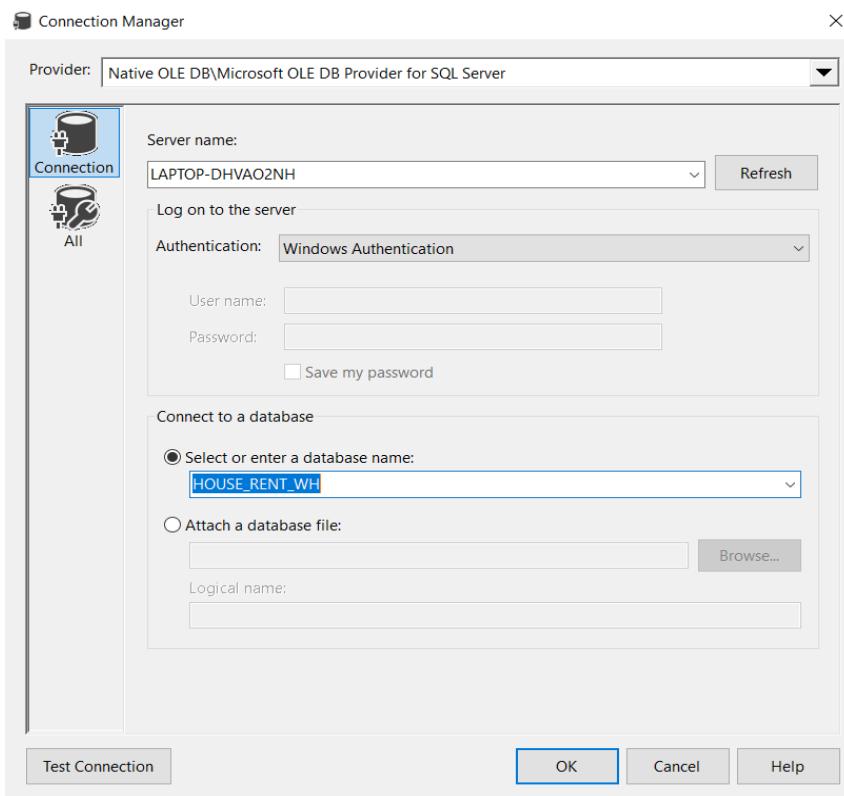
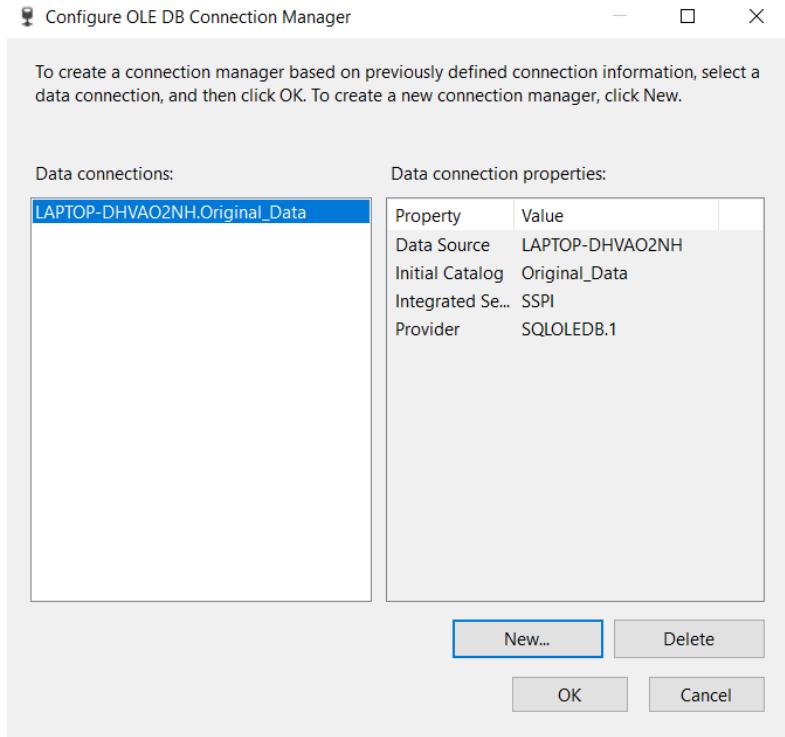




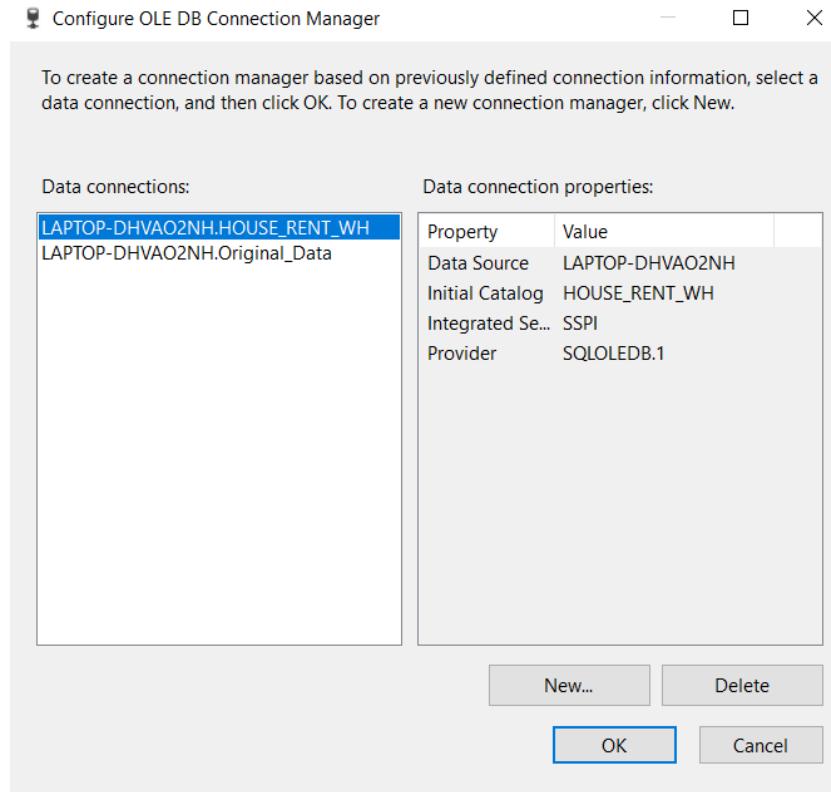
- Nhấn OK và hoàn tất quá trình kết nối đến Original_Data



- Tương tự tạo kết nối đến HOUSE_RENT_WH

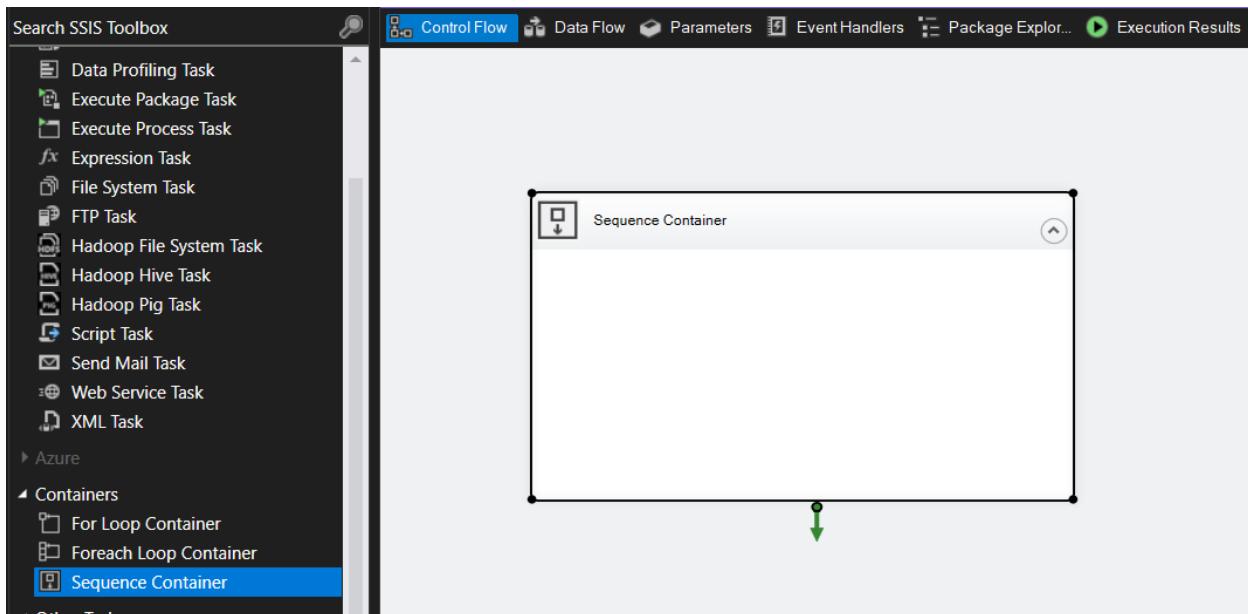


- Thiết lập thành công

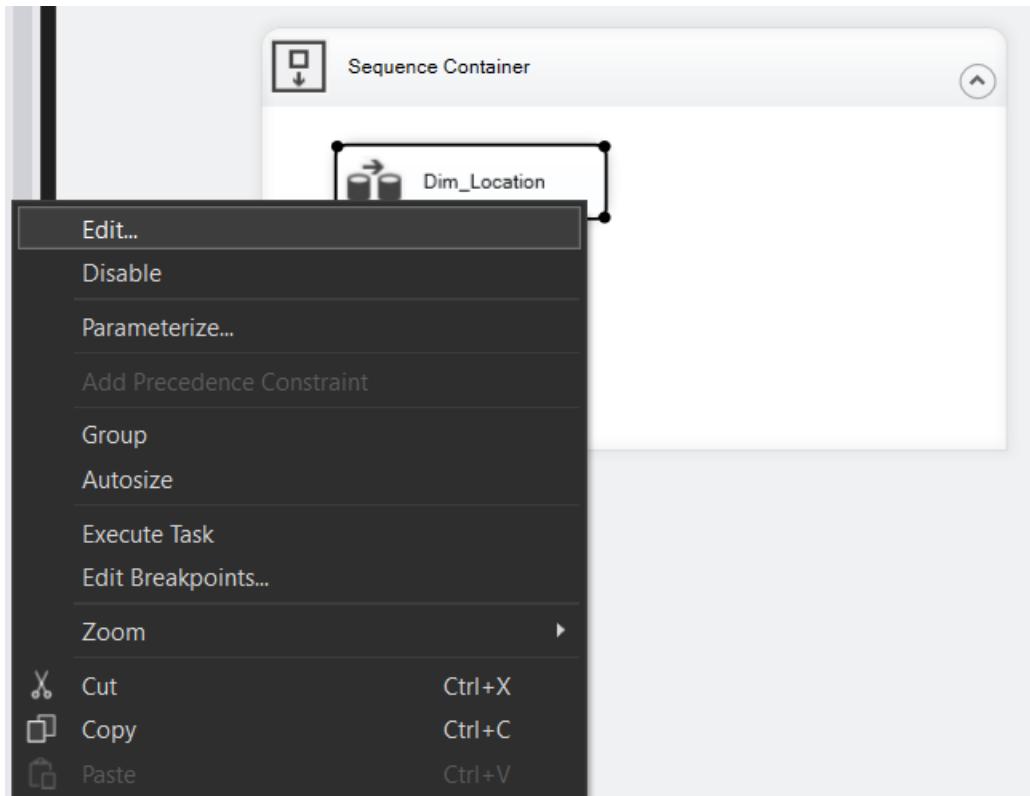


2.5. Quá trình tạo các bảng Dimension

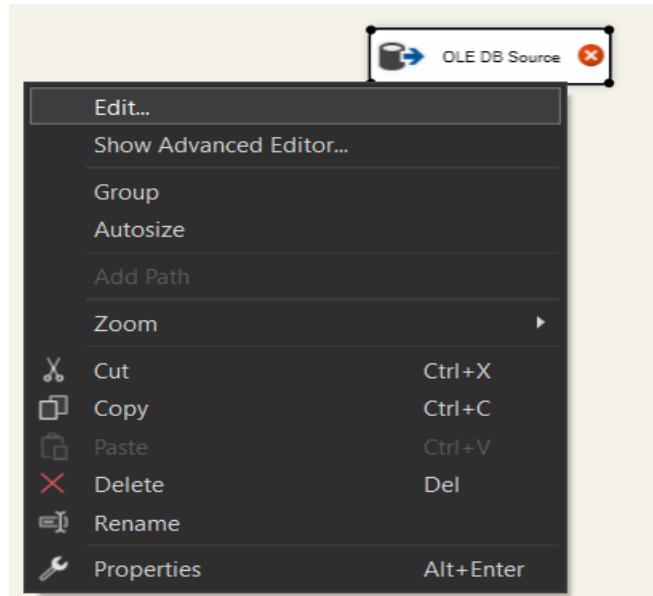
- Tạo Sequence Container tiến hành đồ đồng loạt các bảng Dimensions



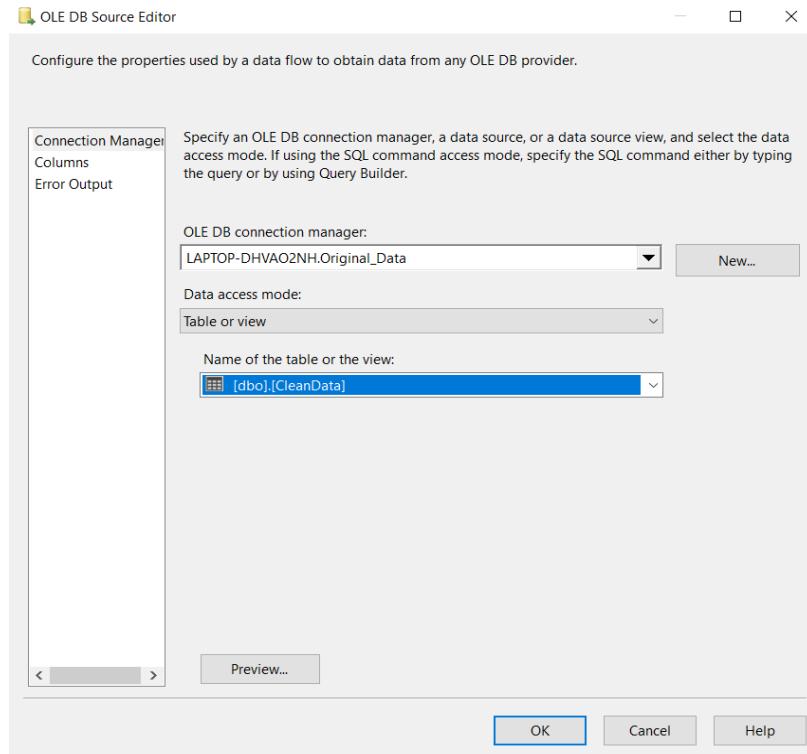
2.5.1. Tạo bảng Dim_Location



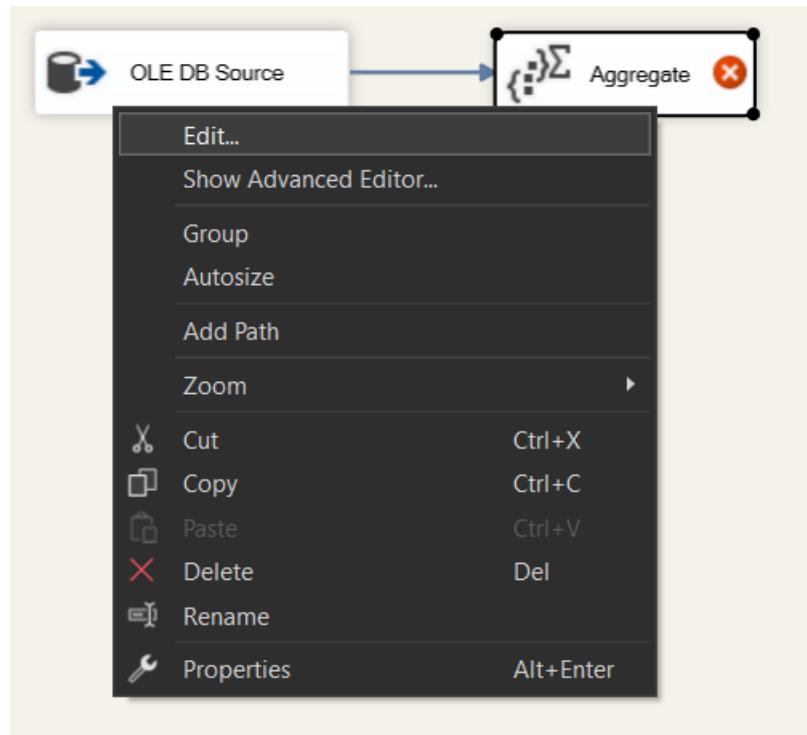
- Kéo thả OLE DB Source

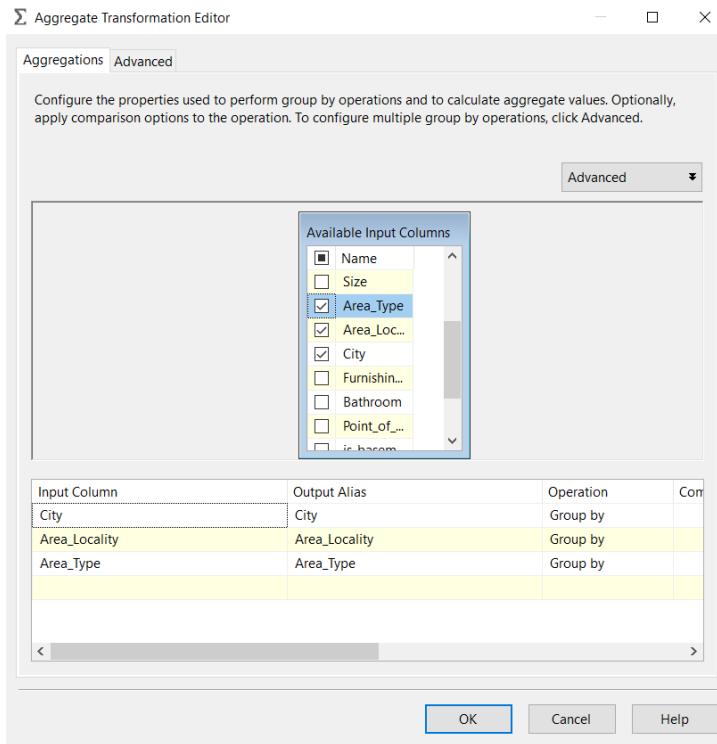


- Chọn bảng CleanData từ Original_Data

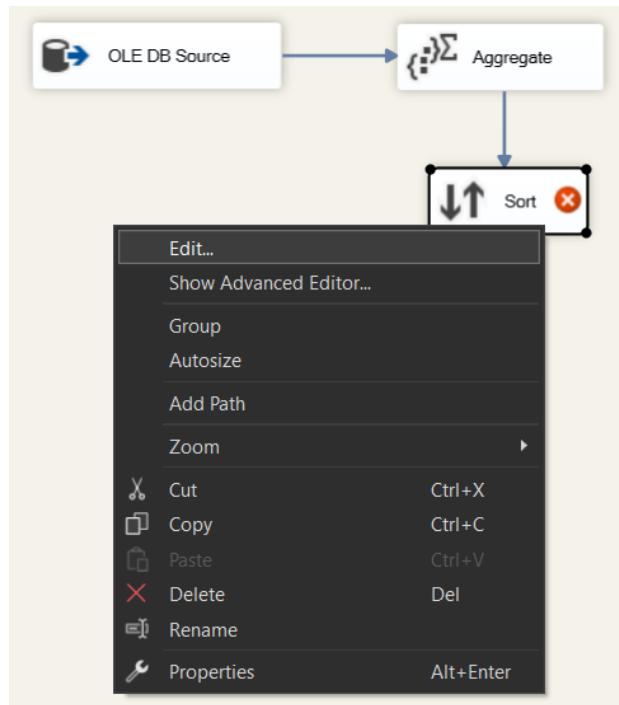


- Sử dụng **Aggregate** để chọn các thuộc tính cần thiết cho bảng.

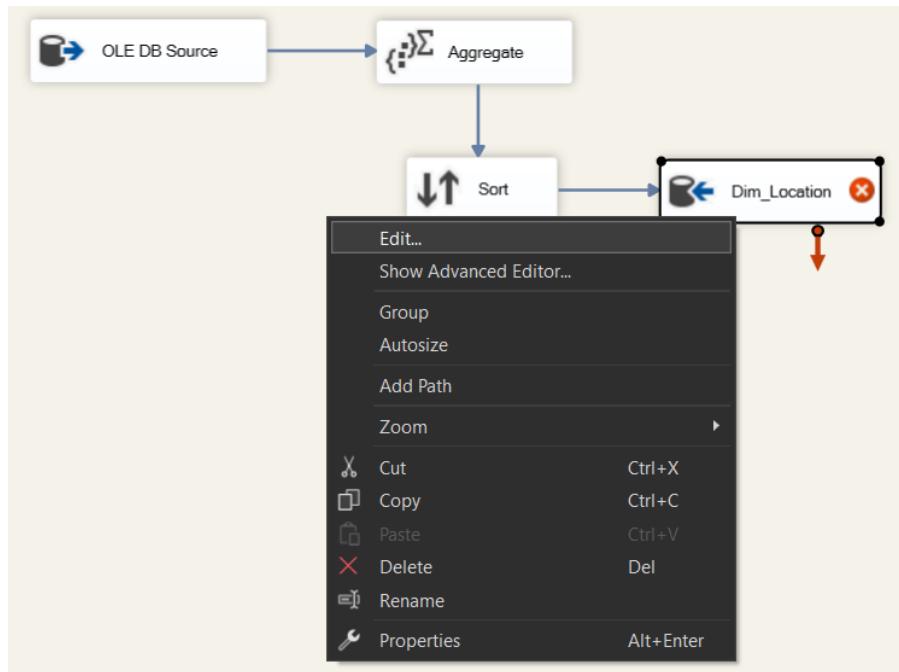




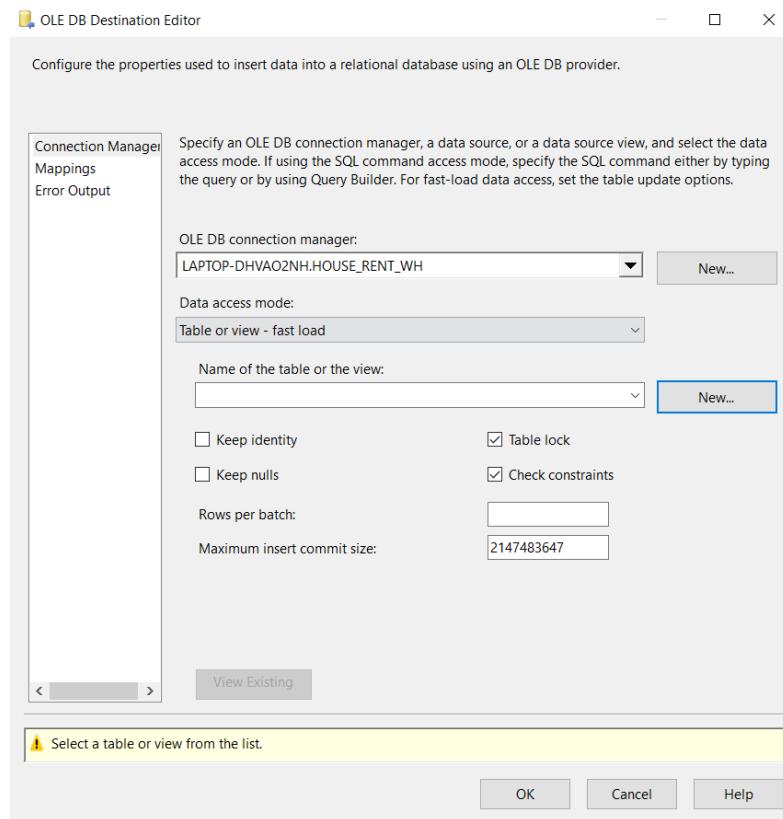
- Sử dụng Sort để sắp xếp lại các thuộc tính.

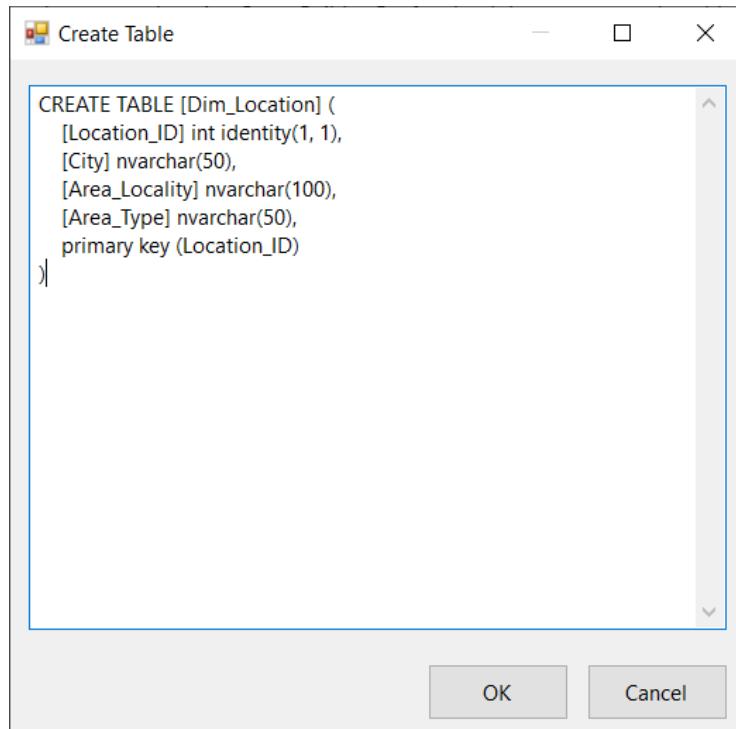


- Kéo thả OLE DB Destination tạo bảng Dim_Location.

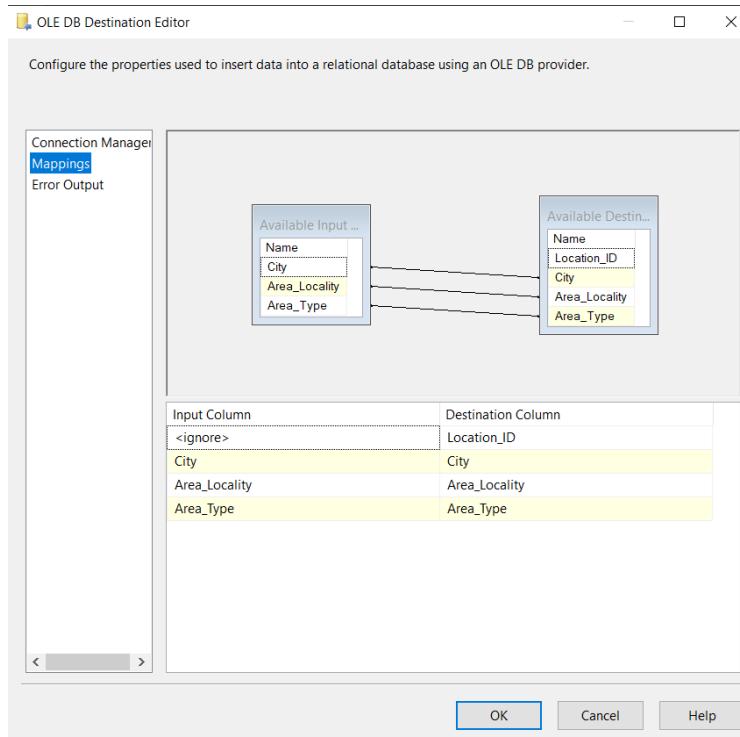


- Chọn New và tạo khóa chính Location_ID

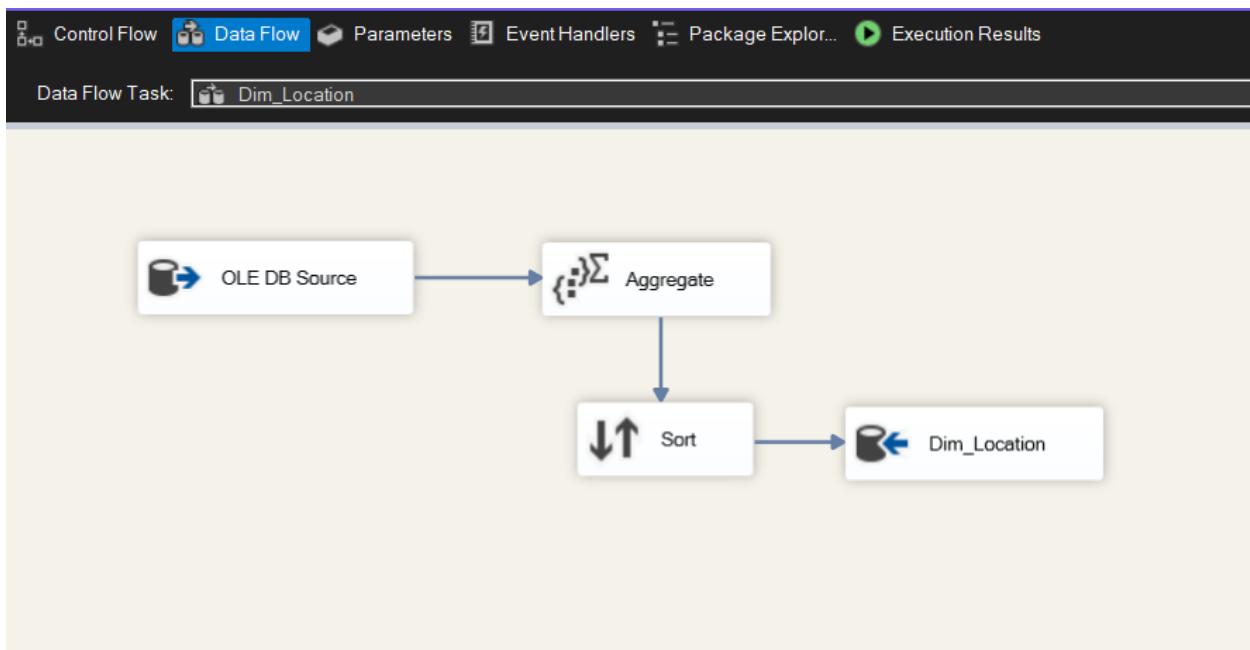




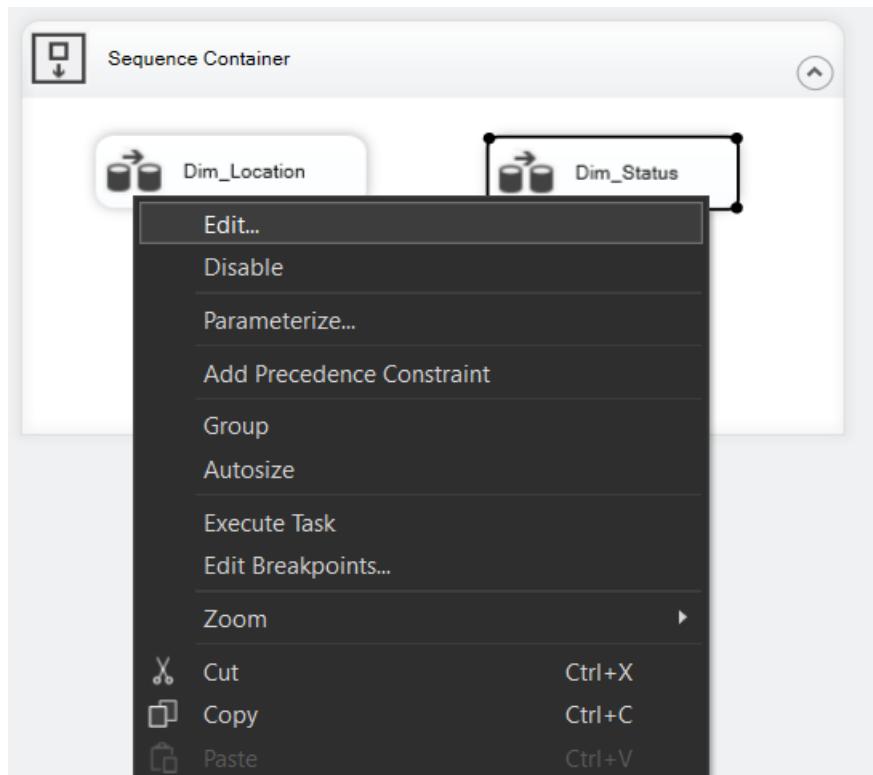
- Kiểm tra các thuộc tính trước khi nhấn OK.



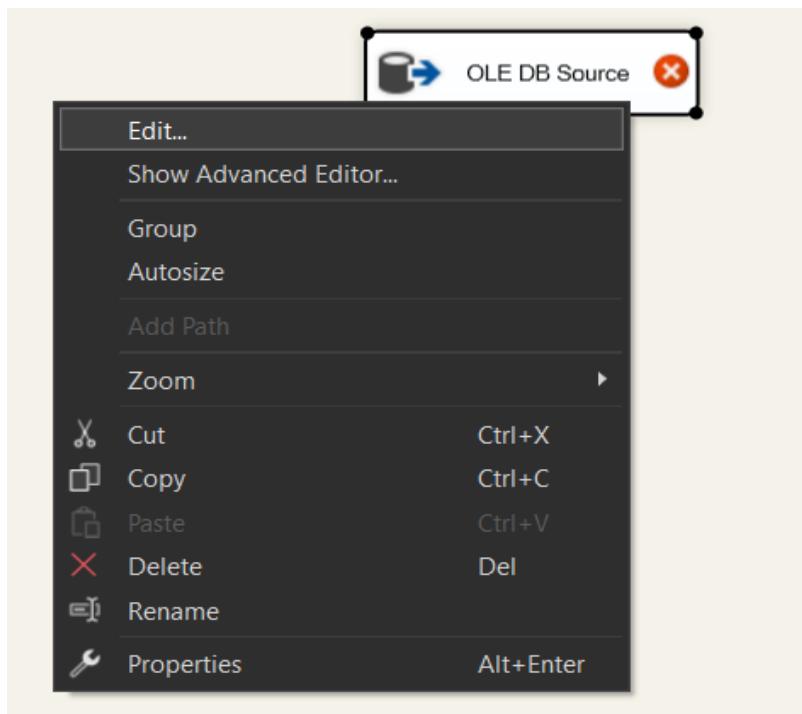
- Data Flow quá trình tạo bảng Dim_Location sau khi hoàn thành.

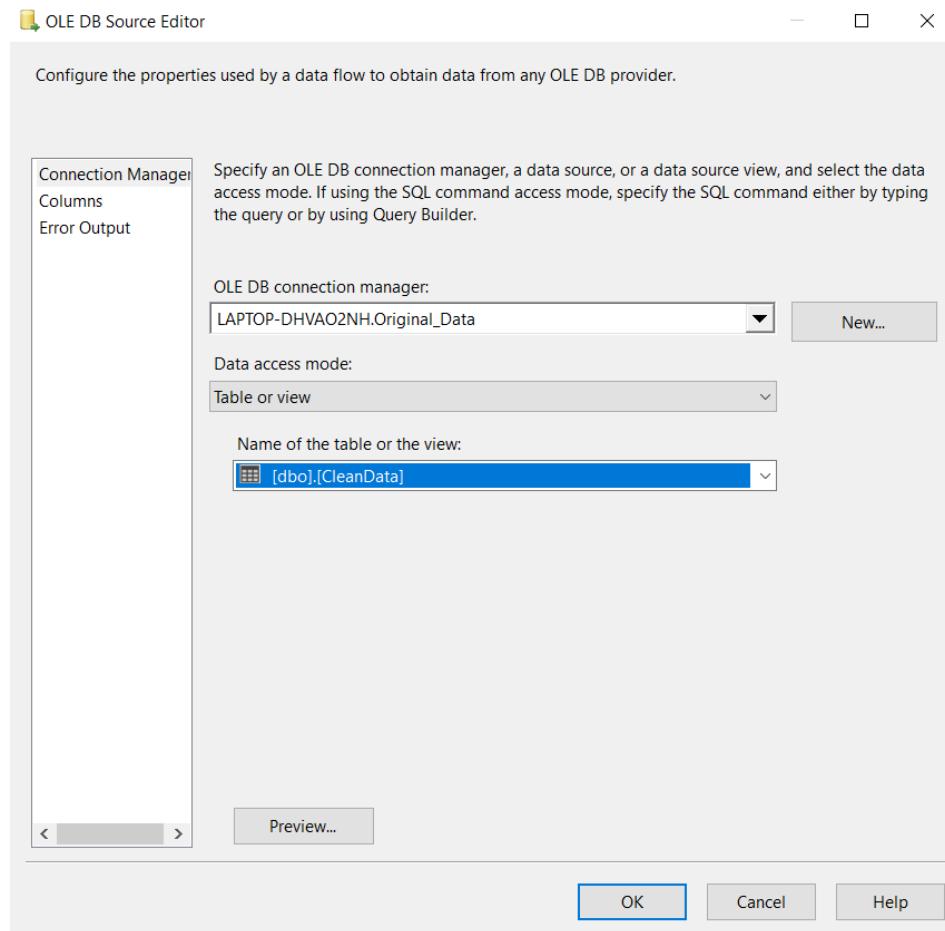


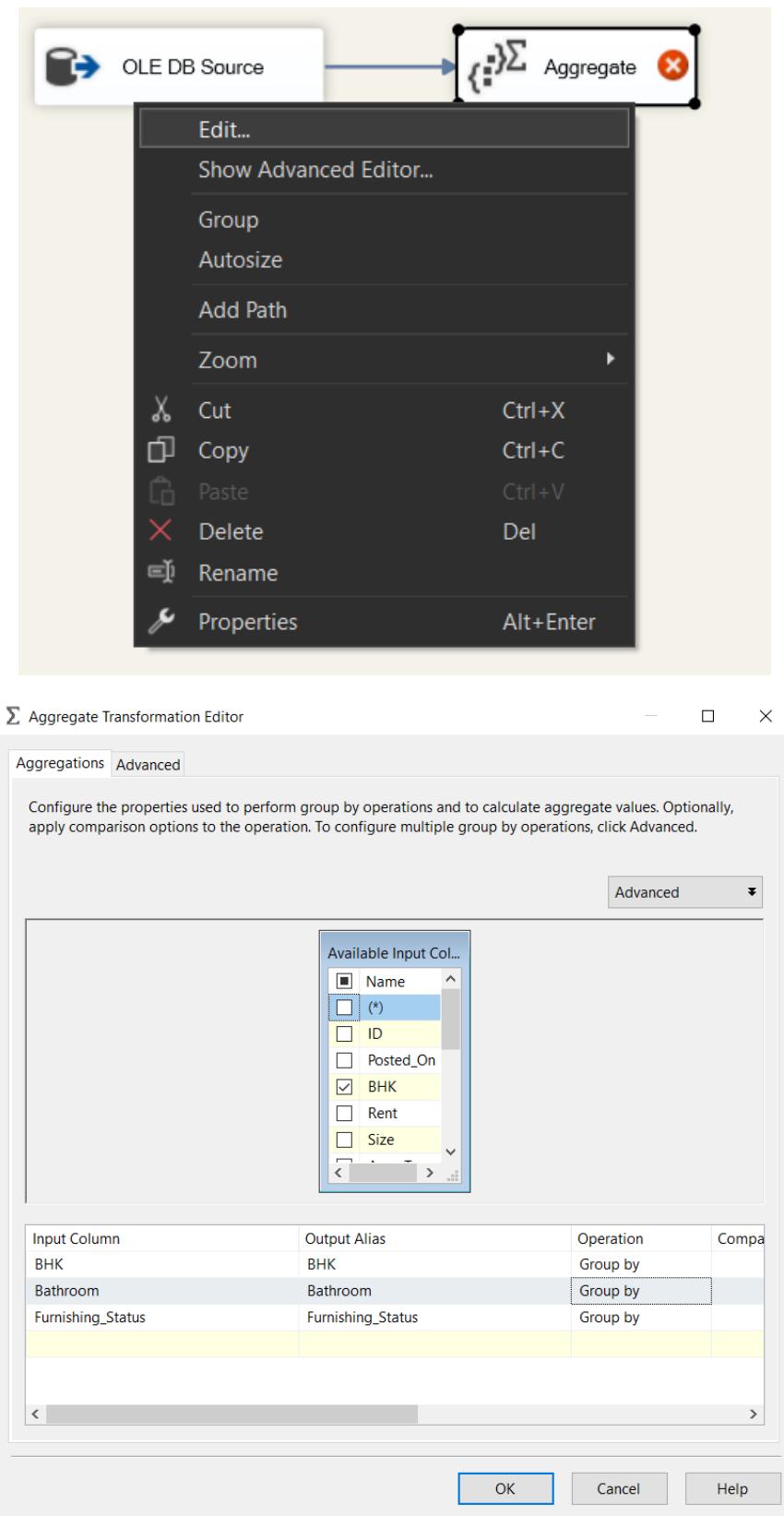
2.5.2. Tạo bảng Dim_Status

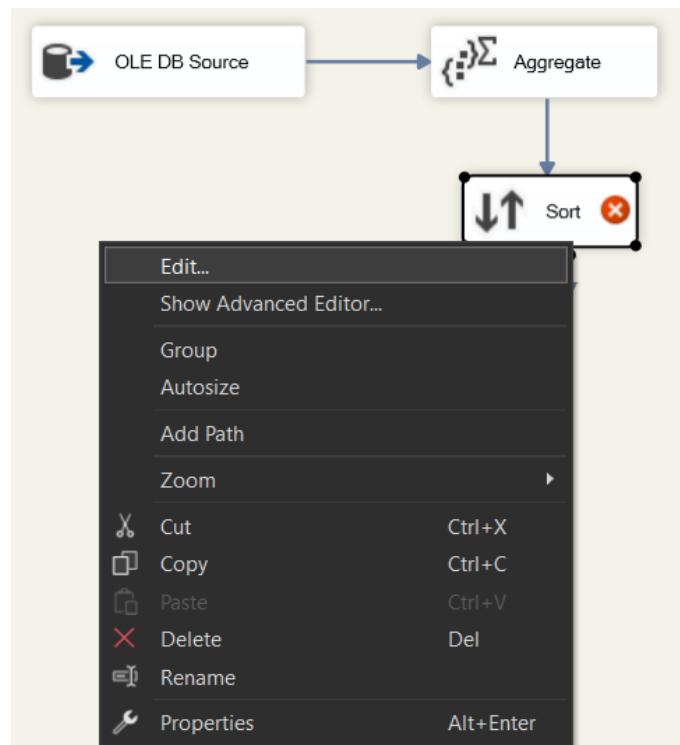


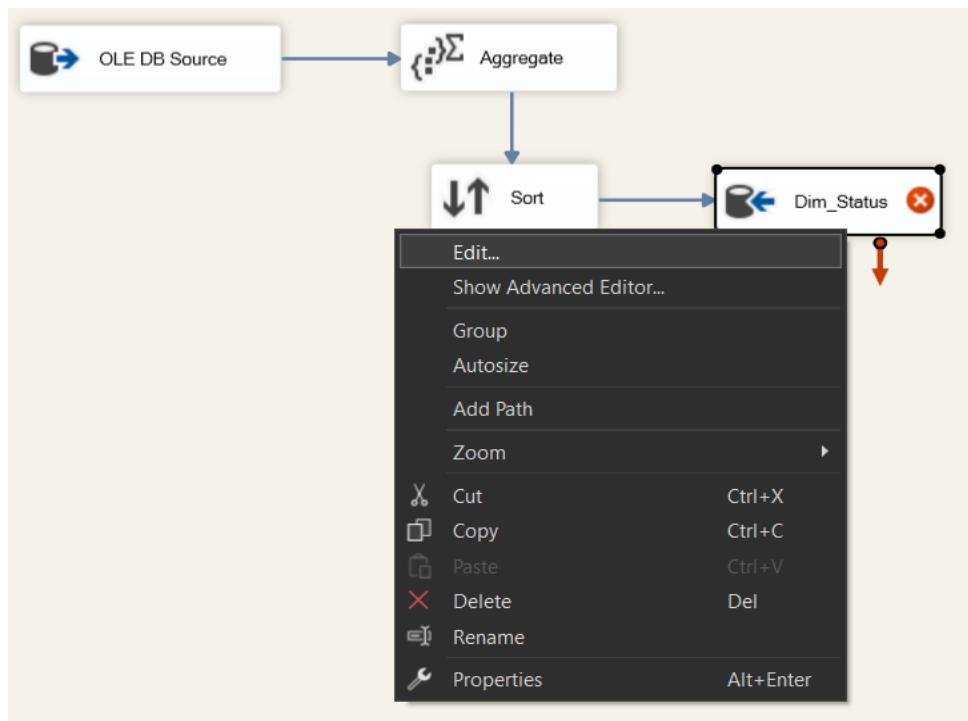
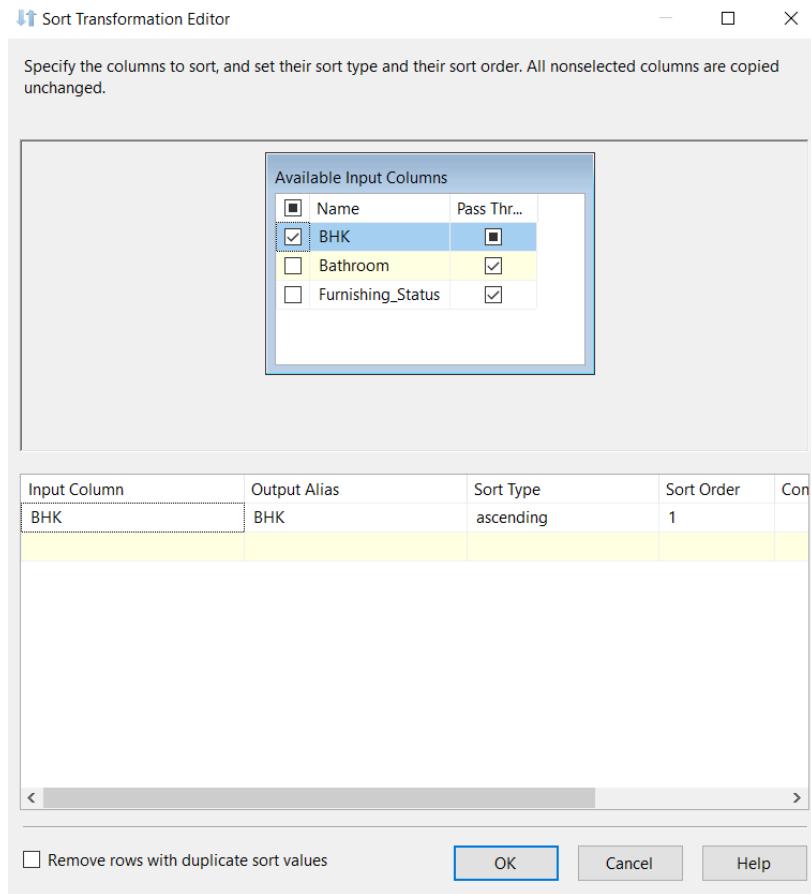
- Thao tác tương tự quá trình tạo bảng Dim_Location

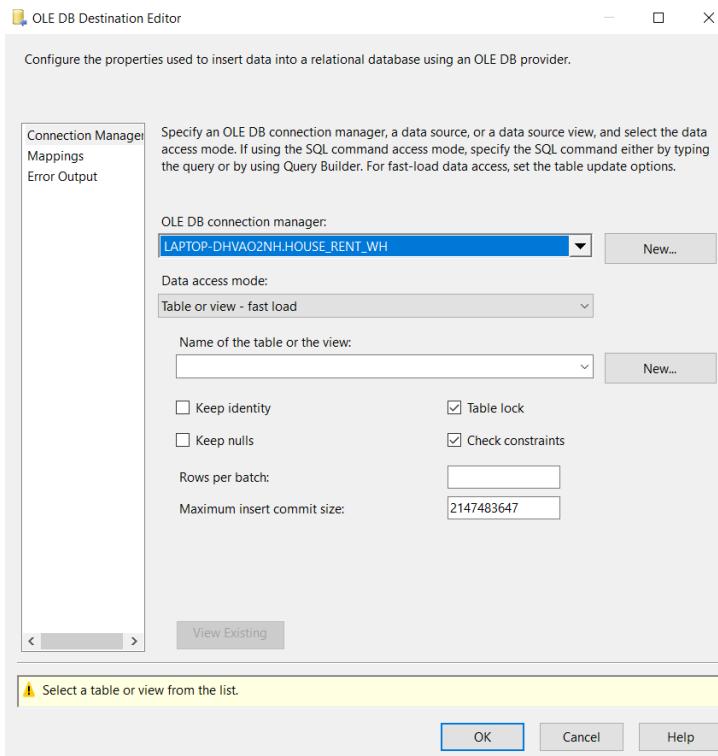




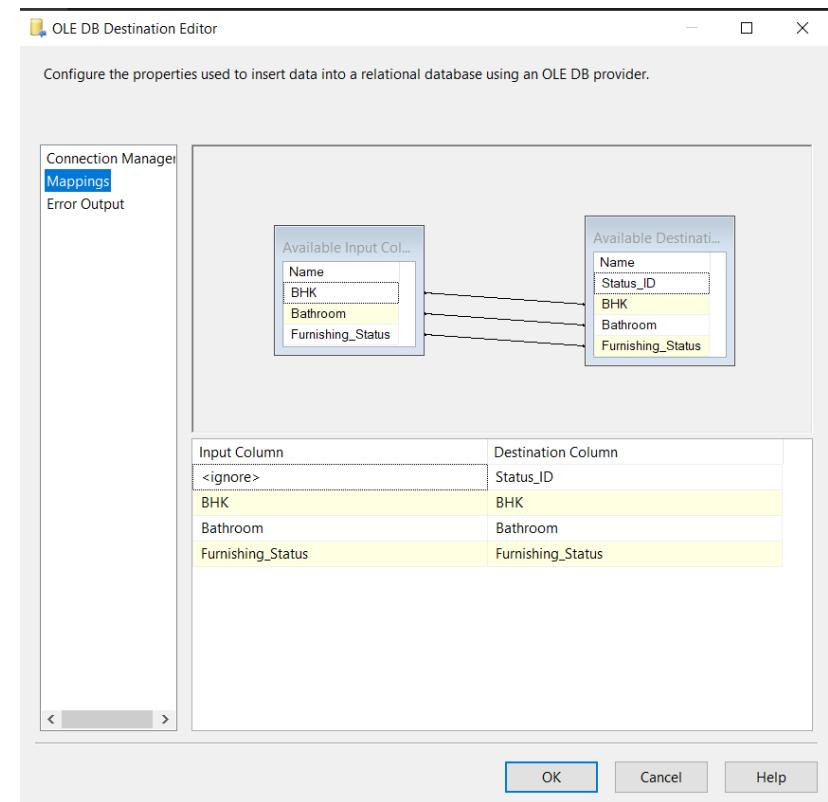
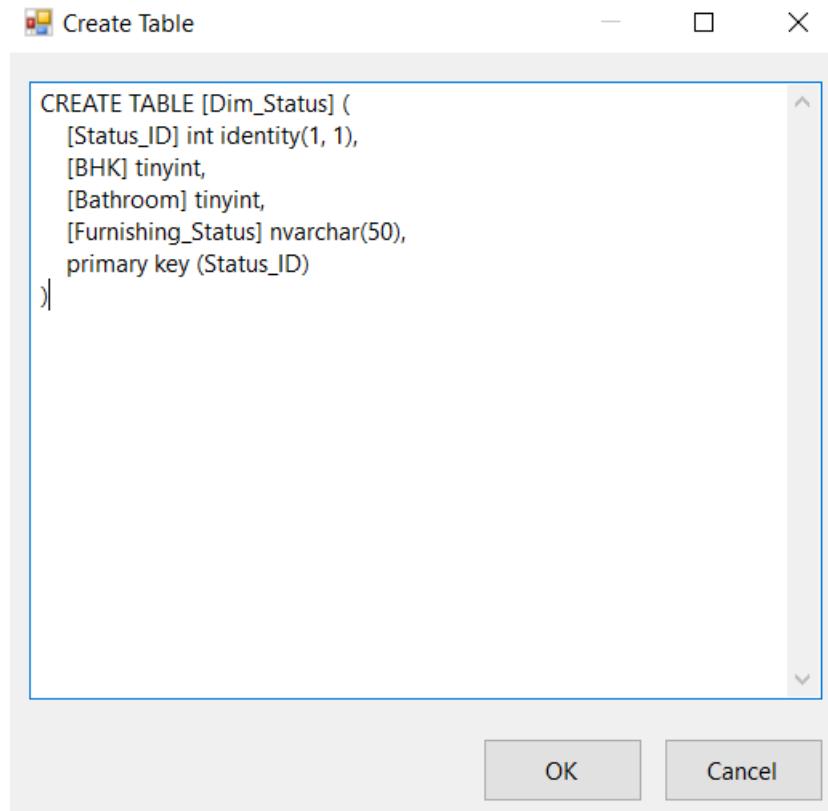




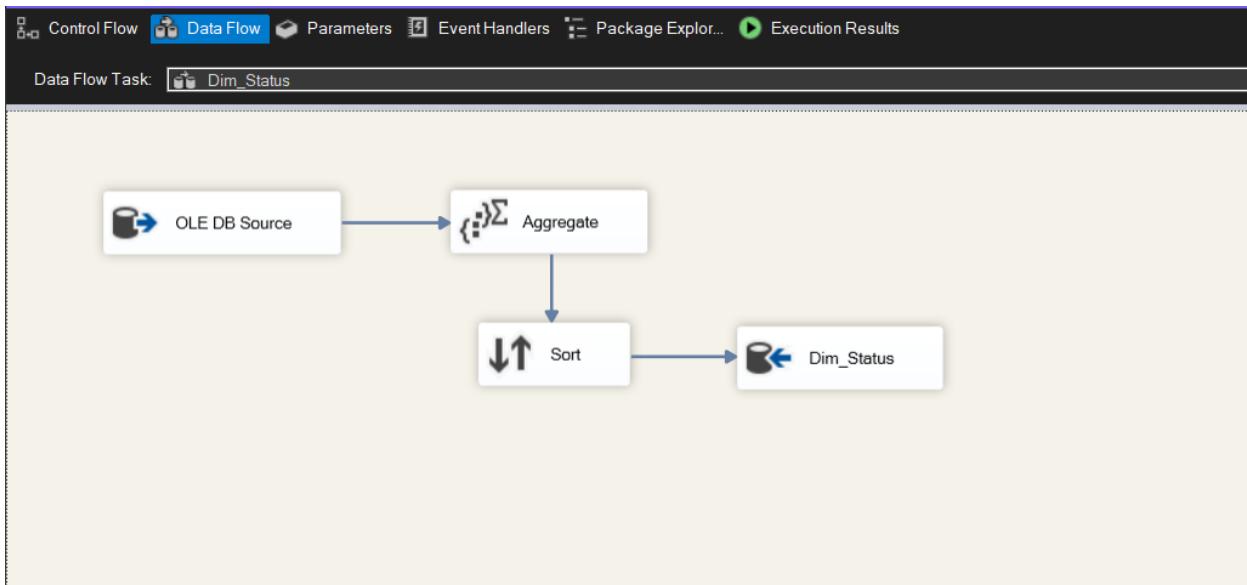




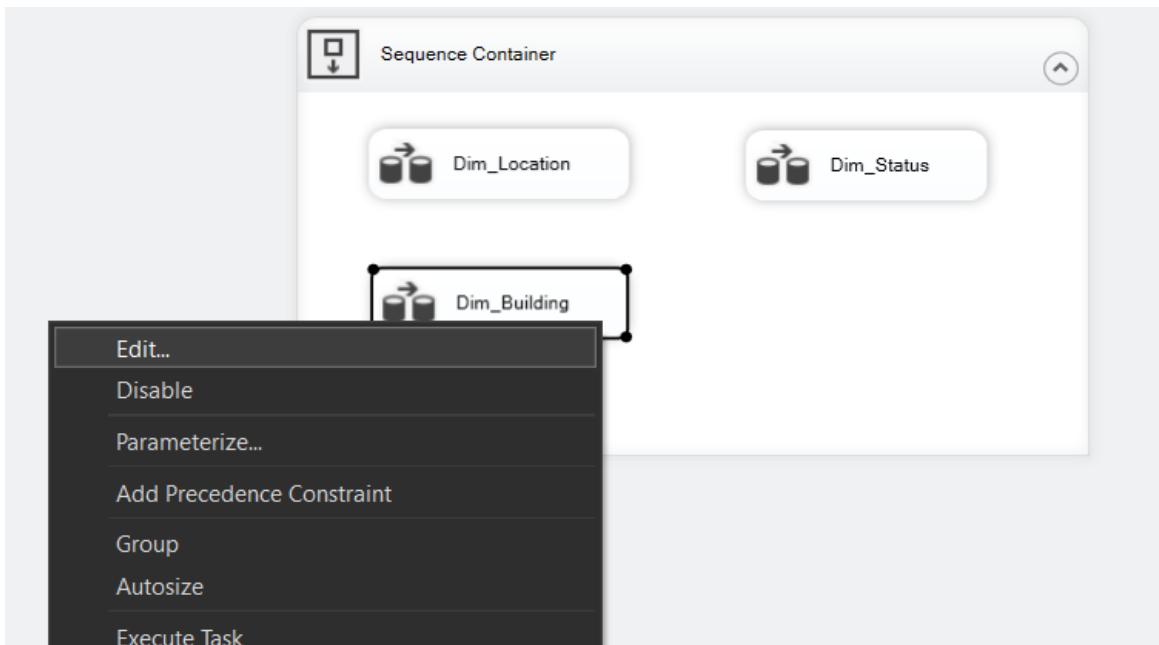
- Tạo khóa chính Status_ID.



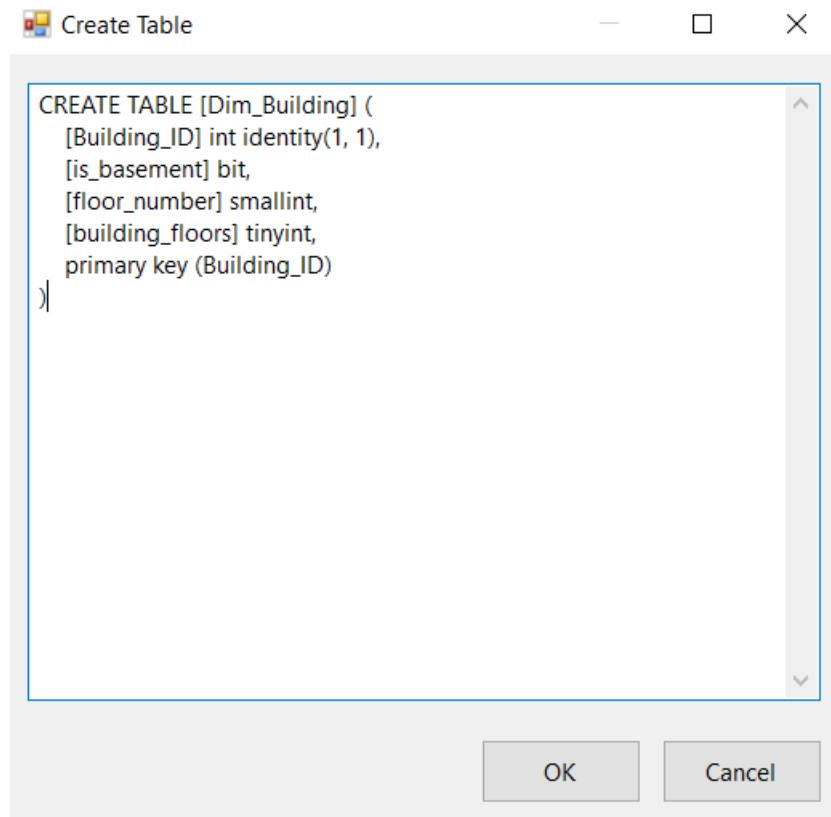
- Data Flow quá trình tạo bảng Dim_Status sau khi hoàn thành.



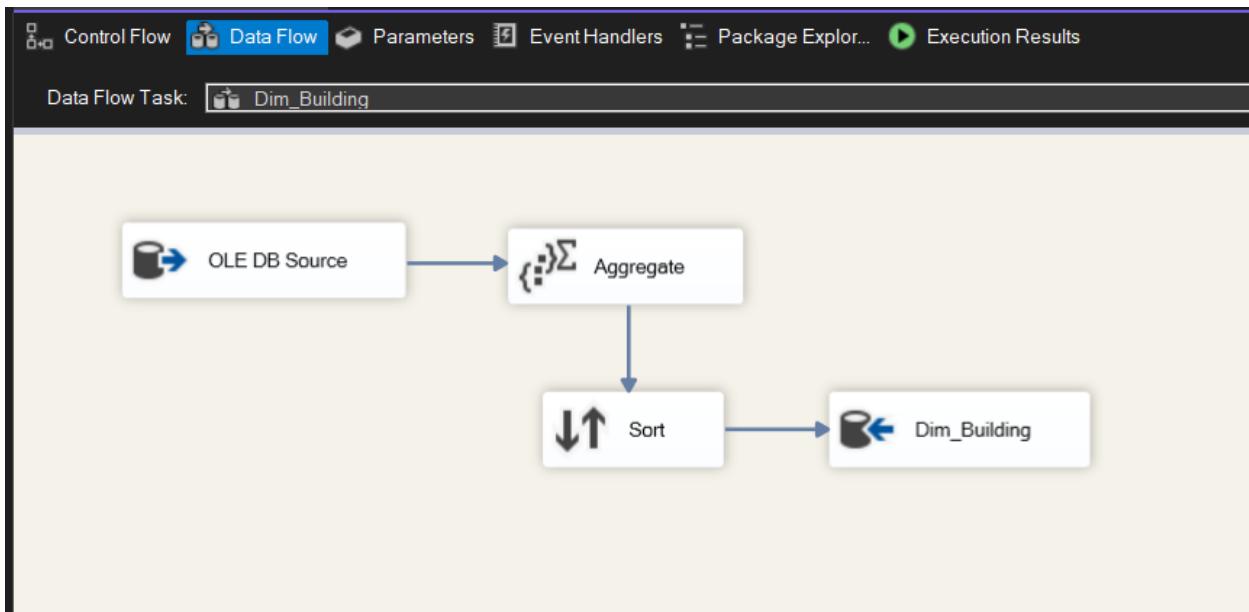
2.5.3. Tạo bảng Dim_Building



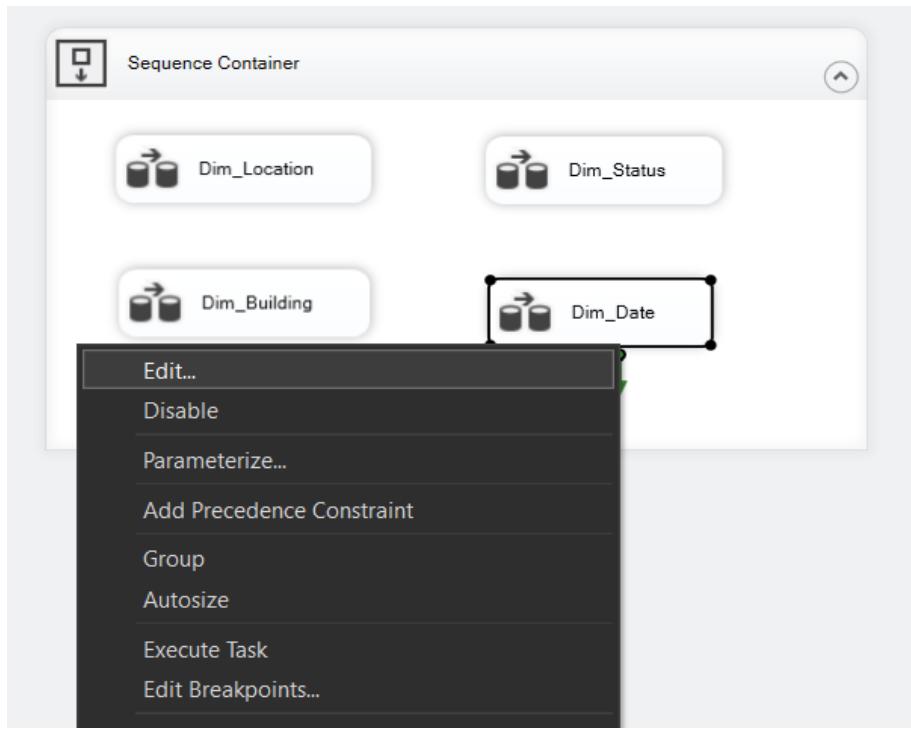
- Thao tác tương tự quá trình tạo bảng Dim_Location
- Tạo khóa chính Building_ID



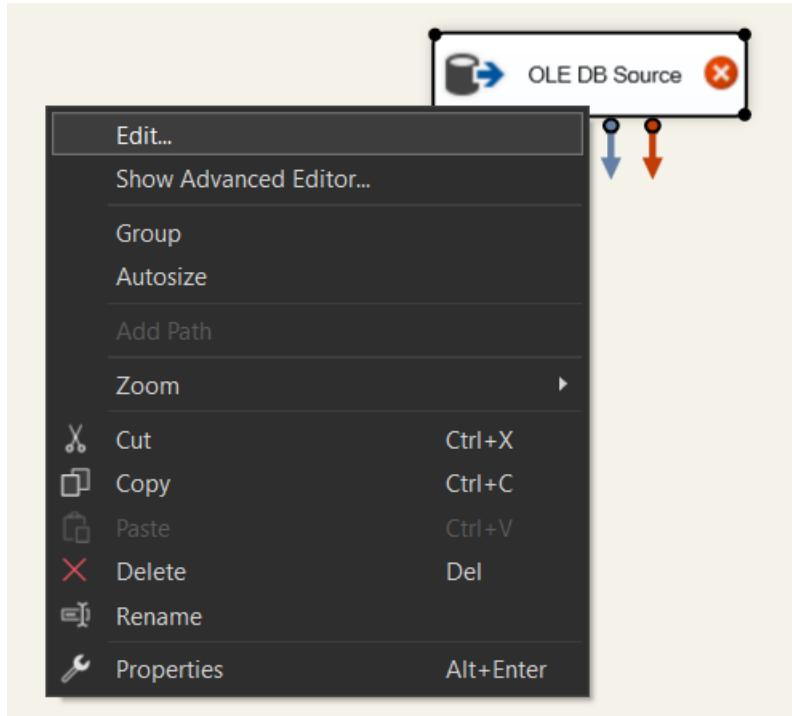
- Data Flow quá trình tạo bảng Dim_Building sau khi hoàn thành.



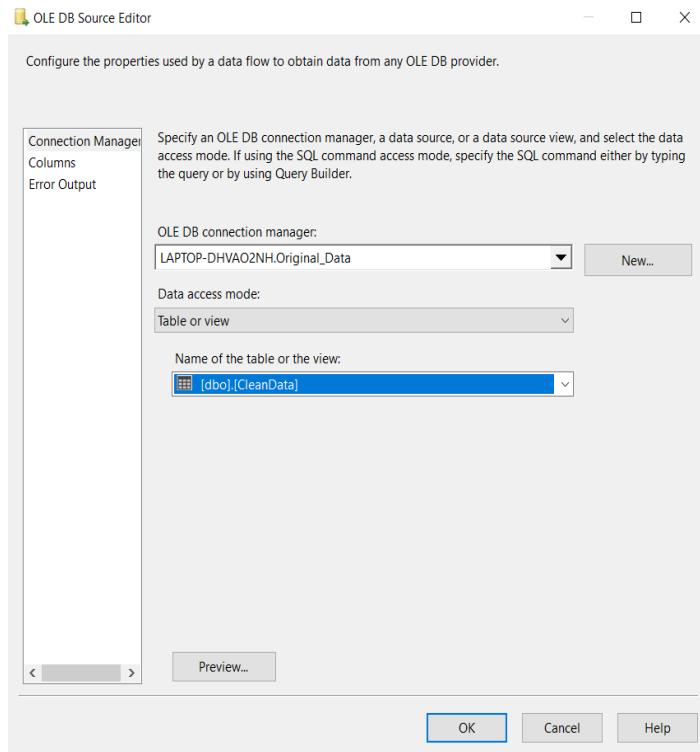
2.5.4. Tạo bảng Dim_Date



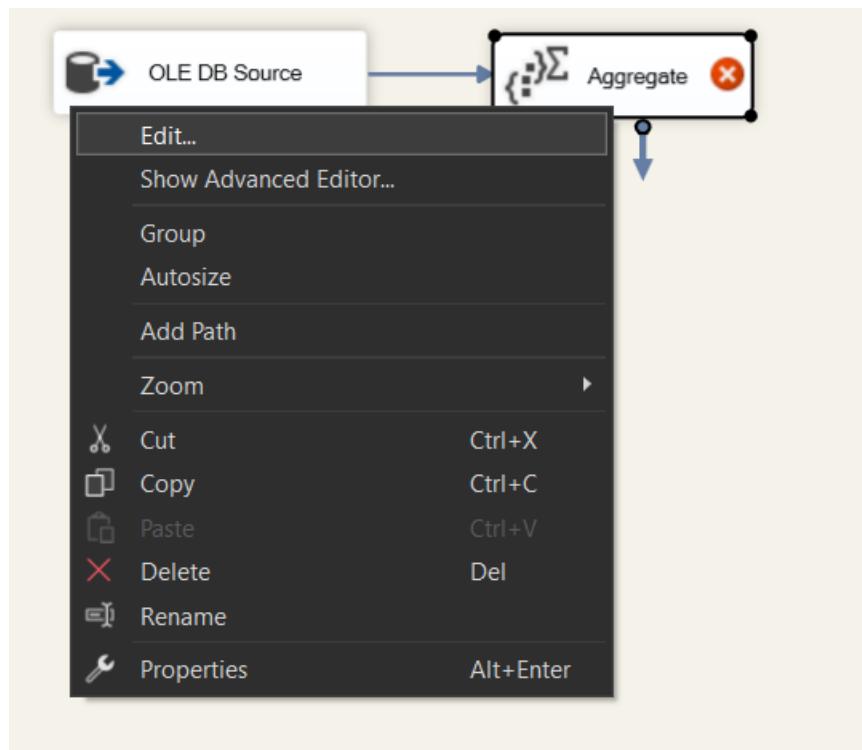
- Kéo thả OLE DB Source.

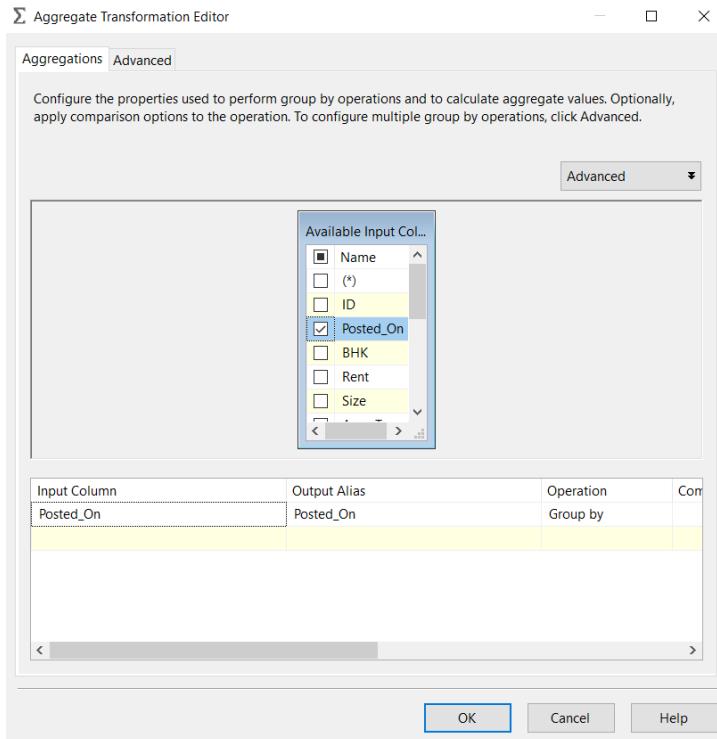


- Chọn bảng CleanData từ Original_Data.

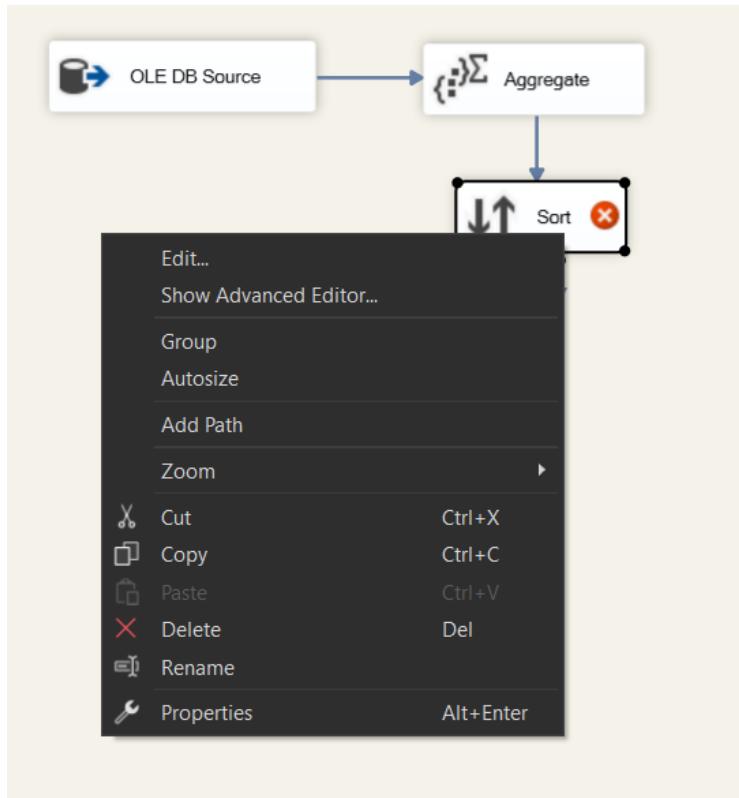


- Sử dụng **Aggregate** để lấy các thuộc tính cần thiết cho bảng.

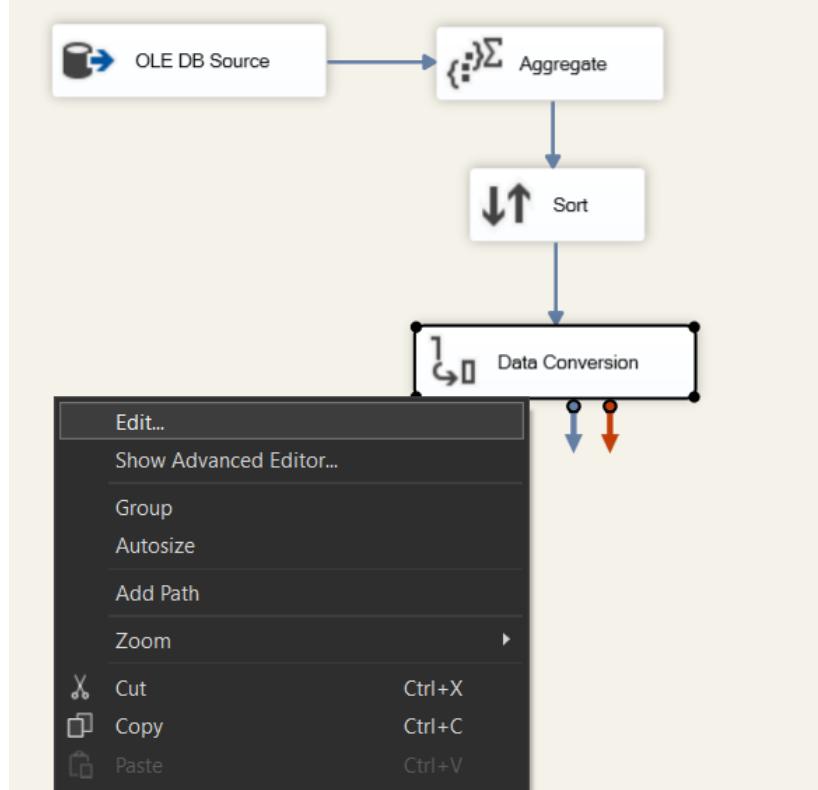


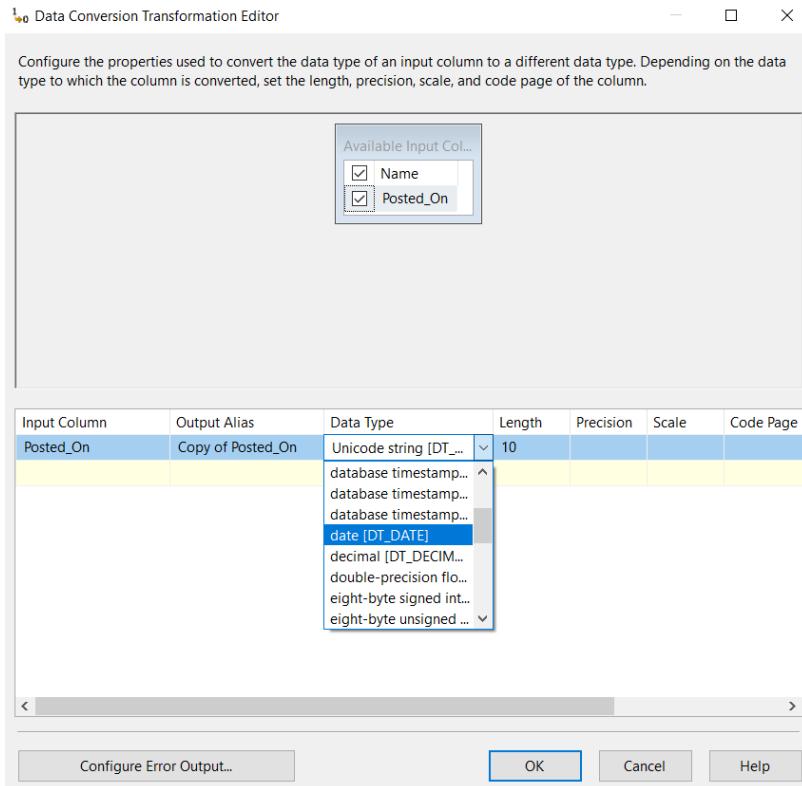


- Sử dụng Sort sắp xếp lại các thuộc tính.

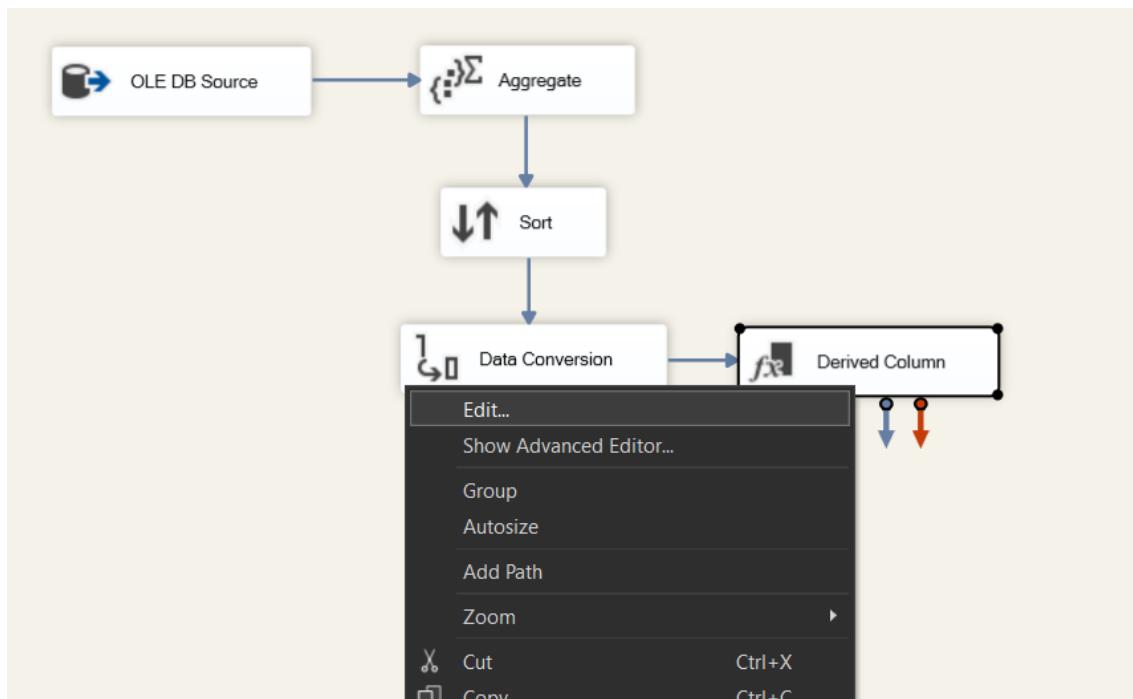


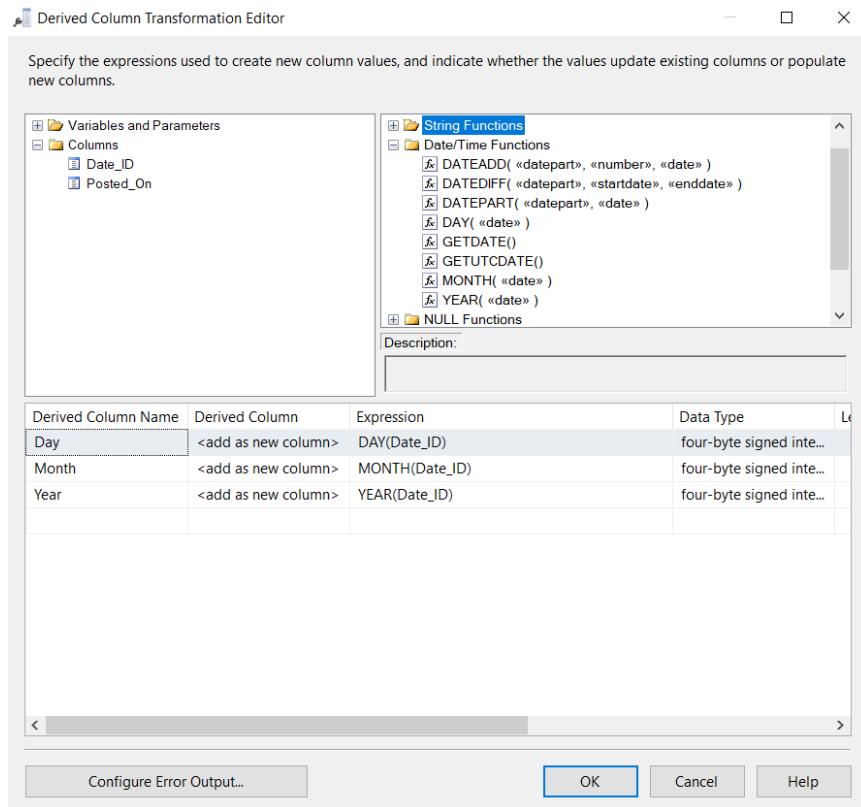
- Sử dụng **Data Conversion** chuyển đổi về kiểu dữ liệu cho thuộc tính Posted_On



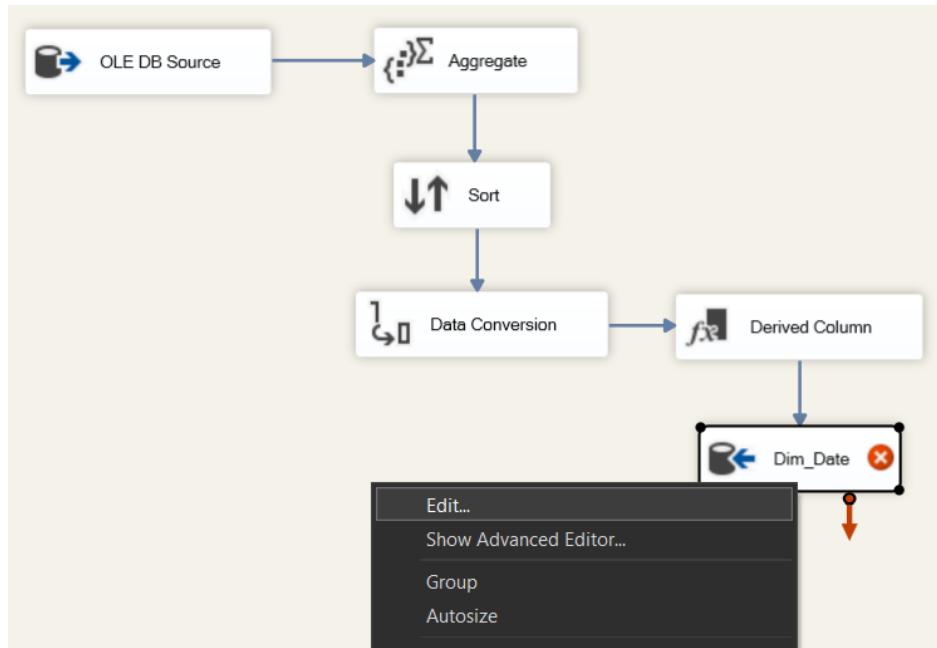


- Sử dụng **Derived Column** để tách cột Posted_On thành các cột dữ liệu có tên Day, Month, Year.

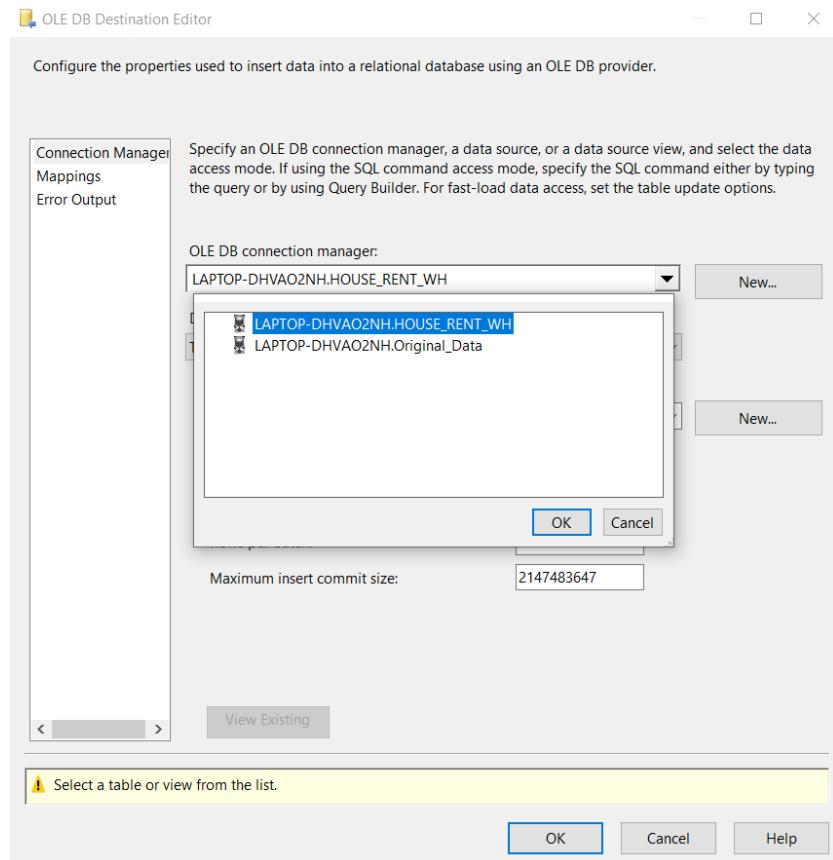


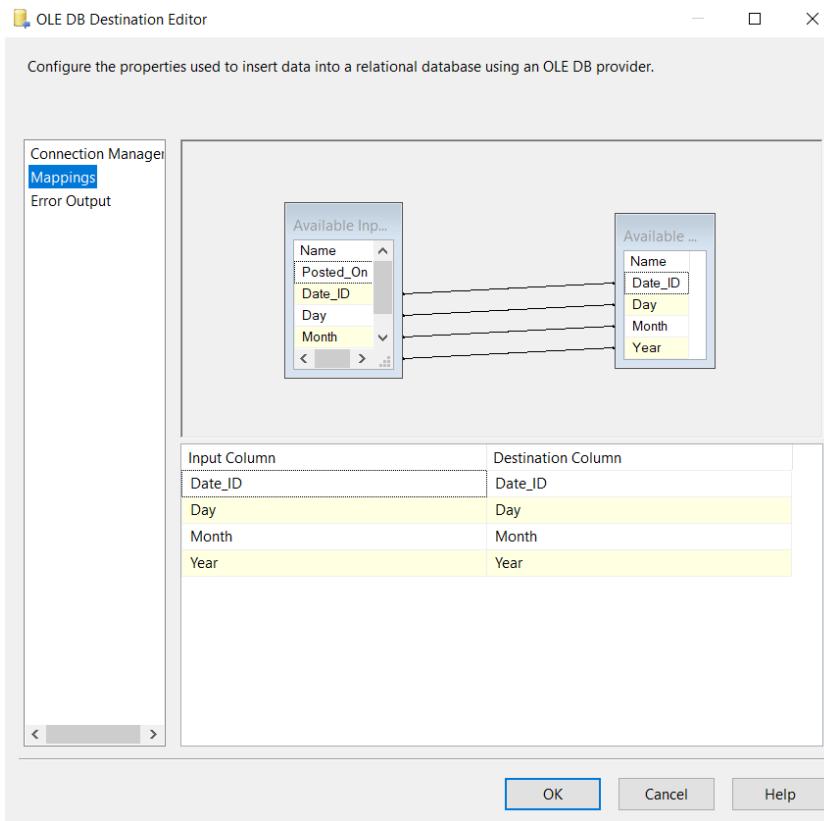
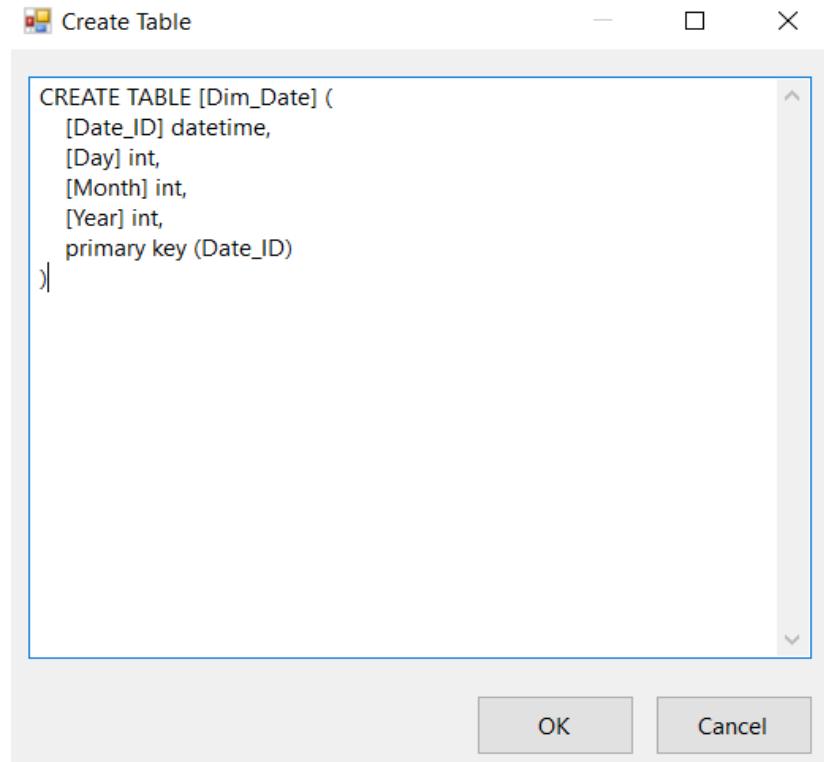


- Kéo thả OLE DB Destination tạo bảng Dim_Date.

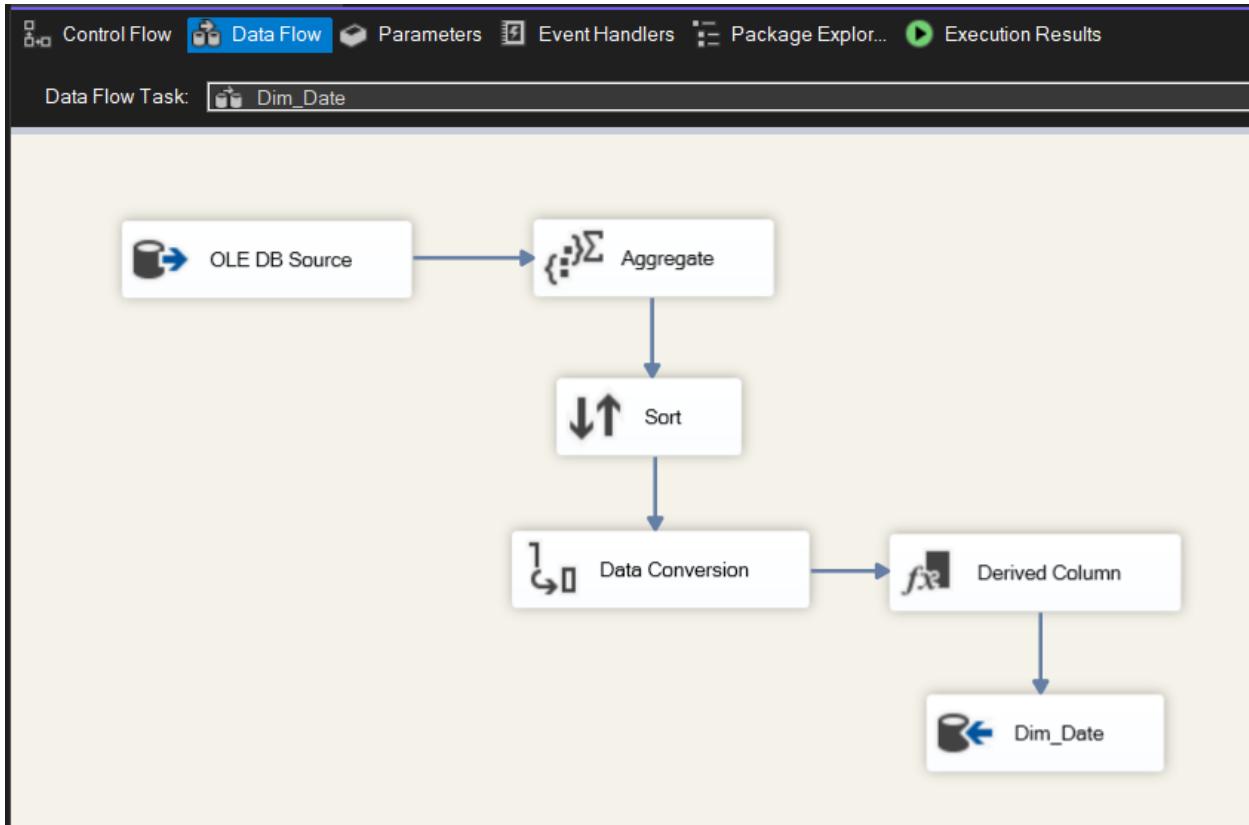


- Chọn House_Rent_WH

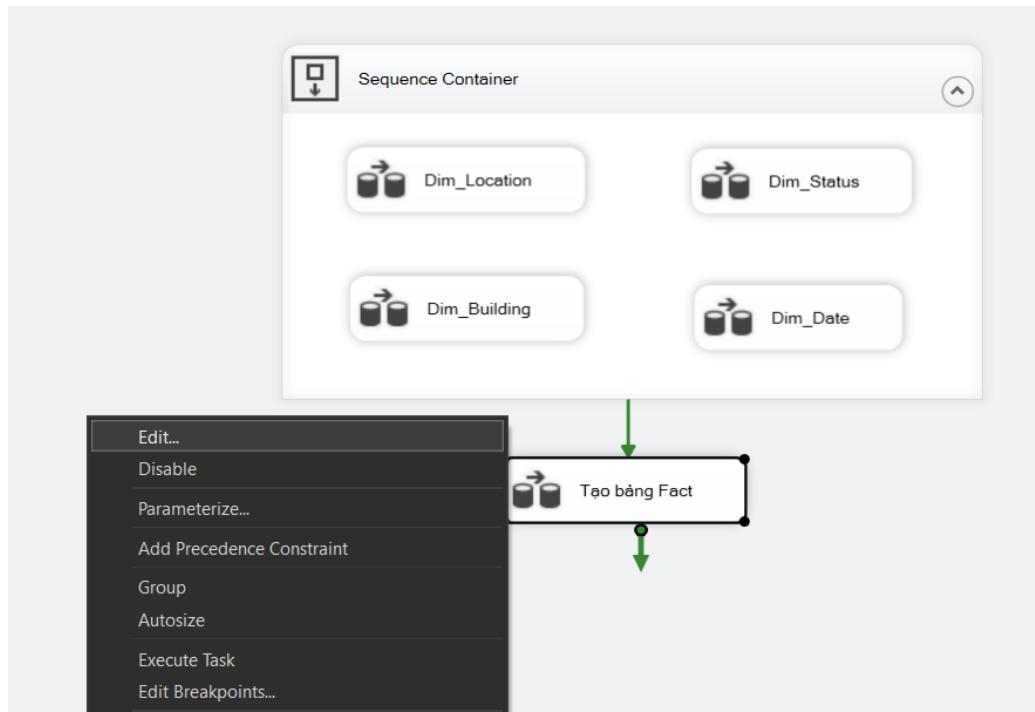




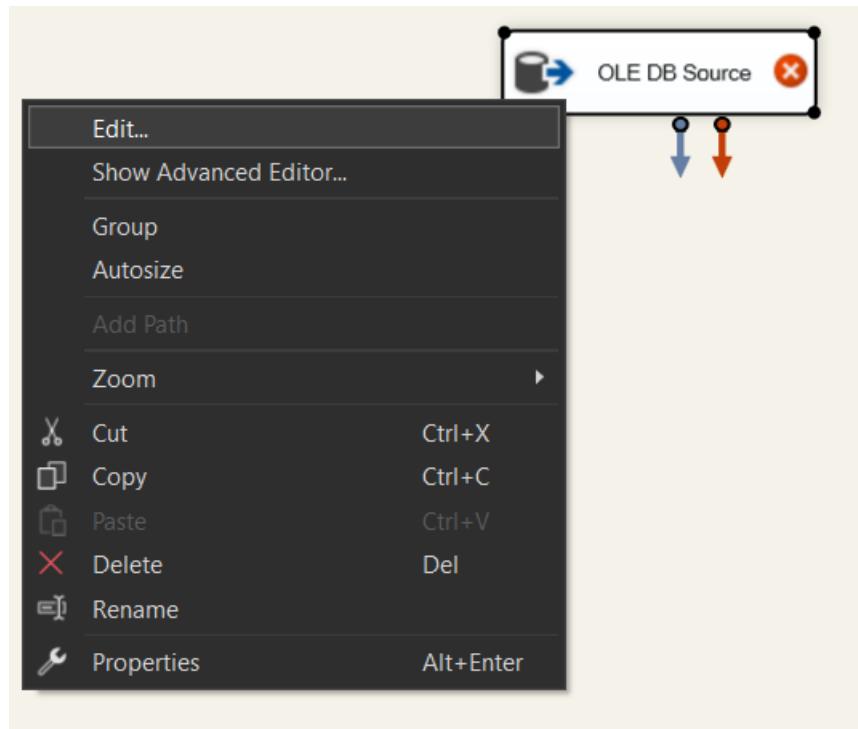
- Data Flow quá trình tạo bảng Dim_Date sau khi hoàn thành.



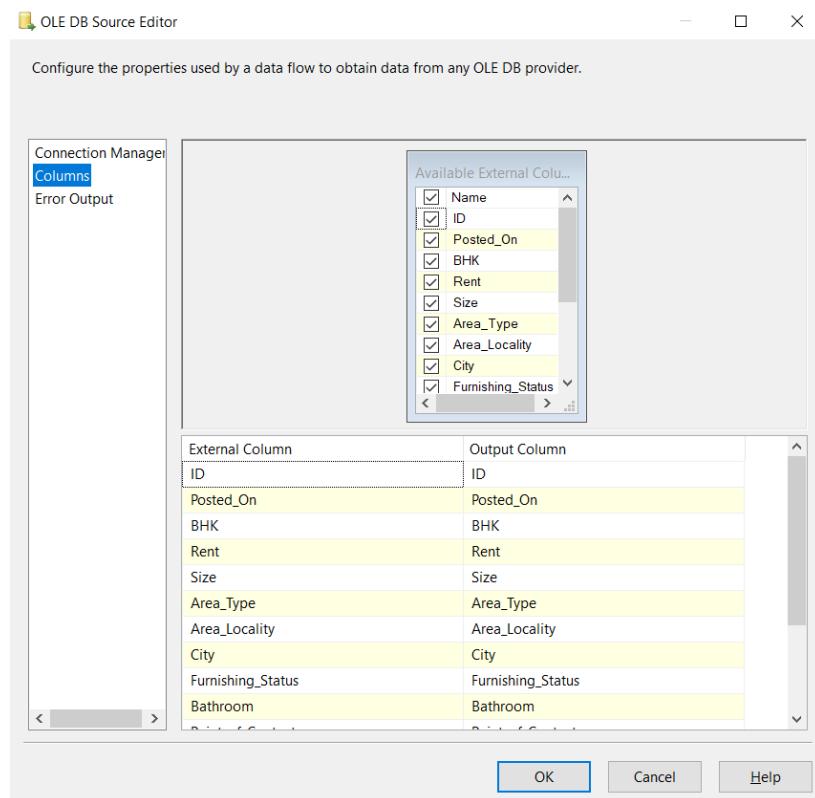
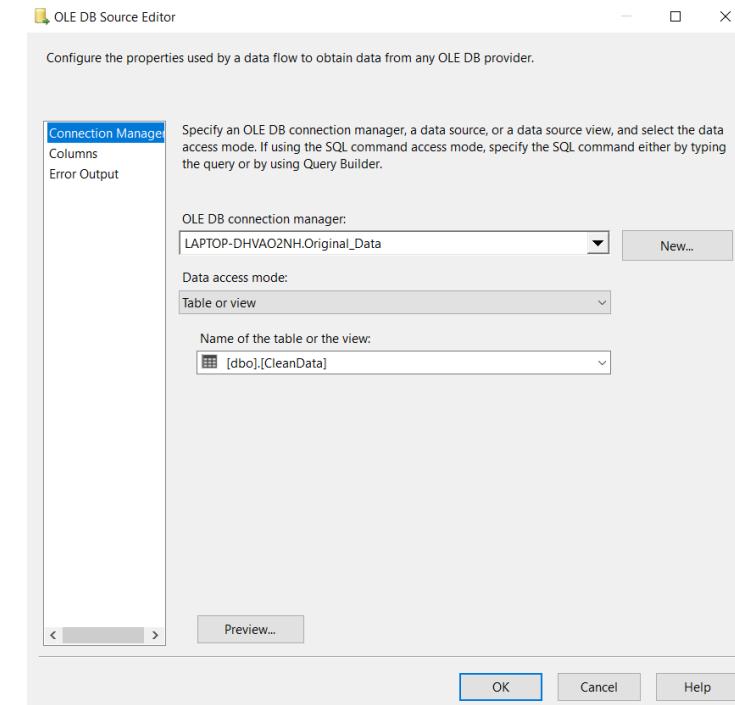
2.6. Quá trình tạo bảng Fact



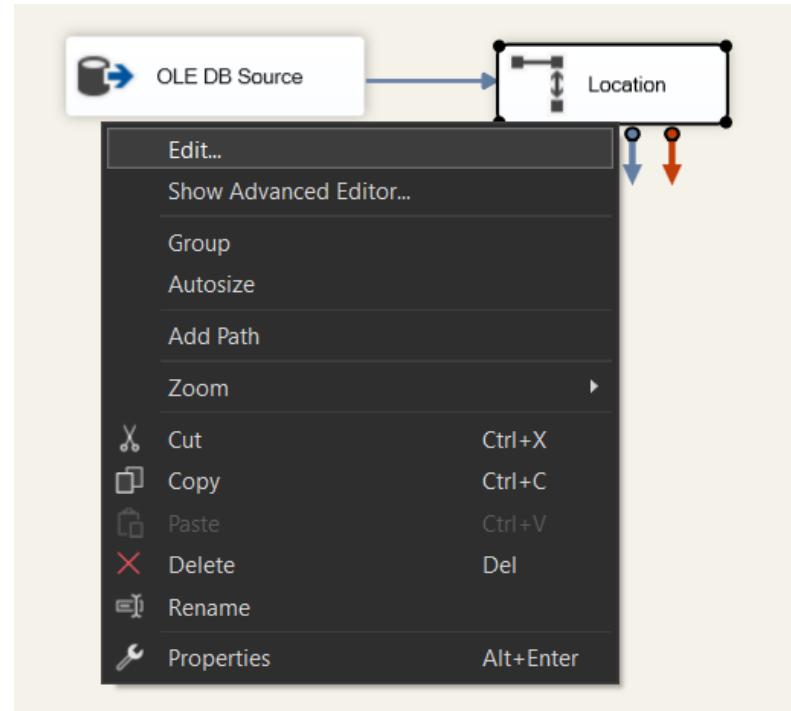
- Kéo thả OLE DB Source



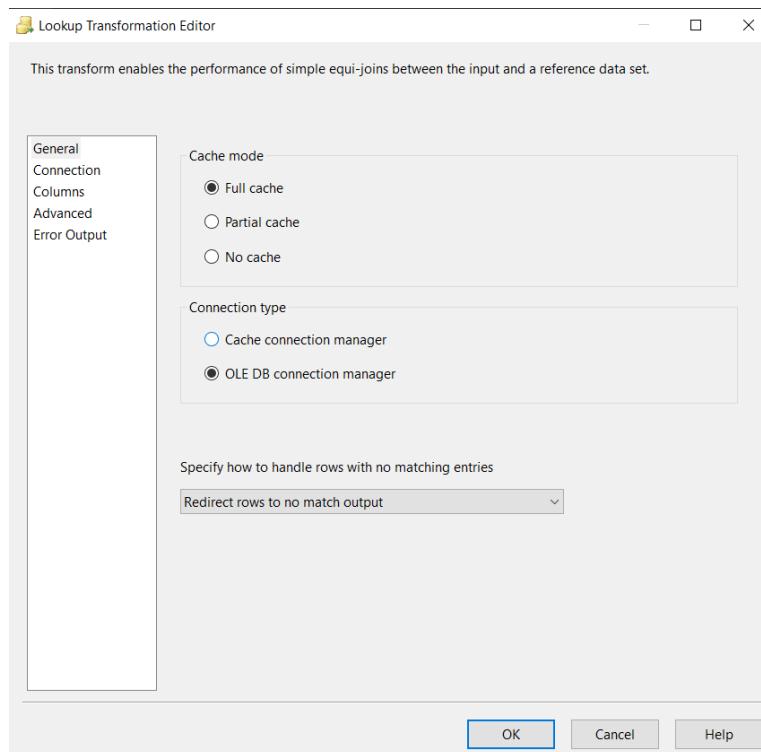
- Chọn bảng CleanData từ Original_Data

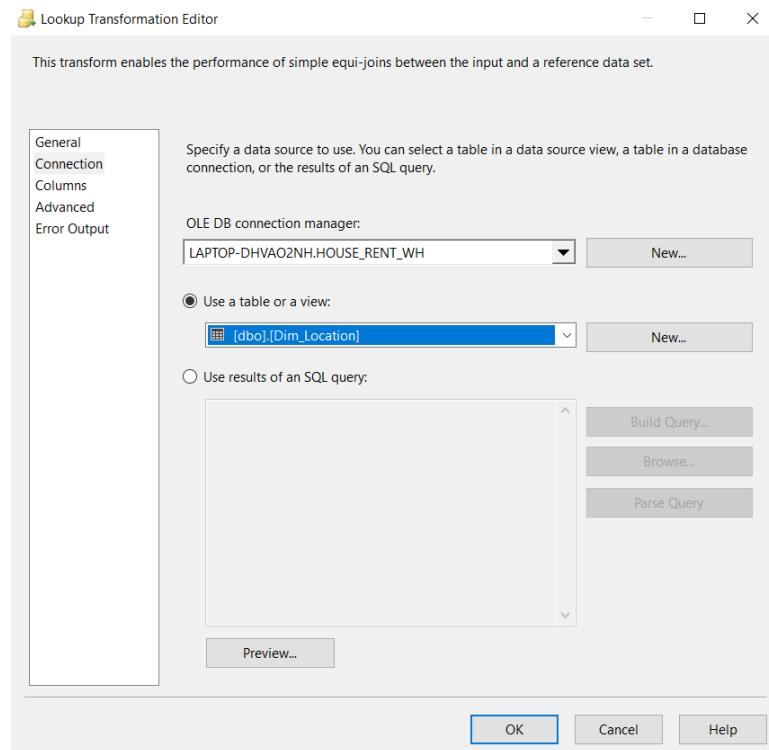


- Sử dụng **Look Up** các thuộc tính từ bảng Dim_Location và chọn **Edit**.

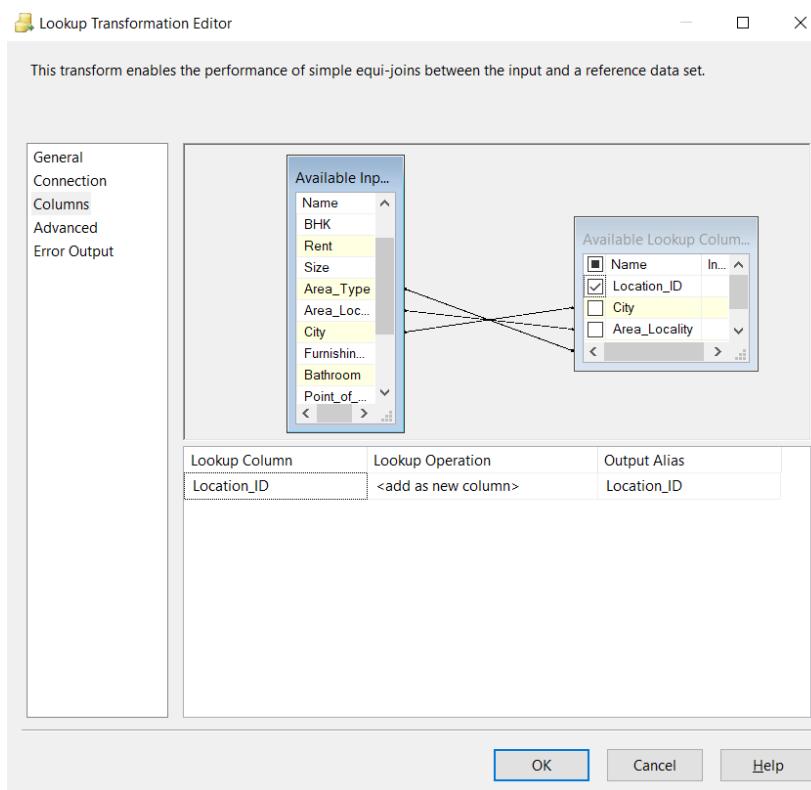


- Chọn **Redirect rows to no match output**.

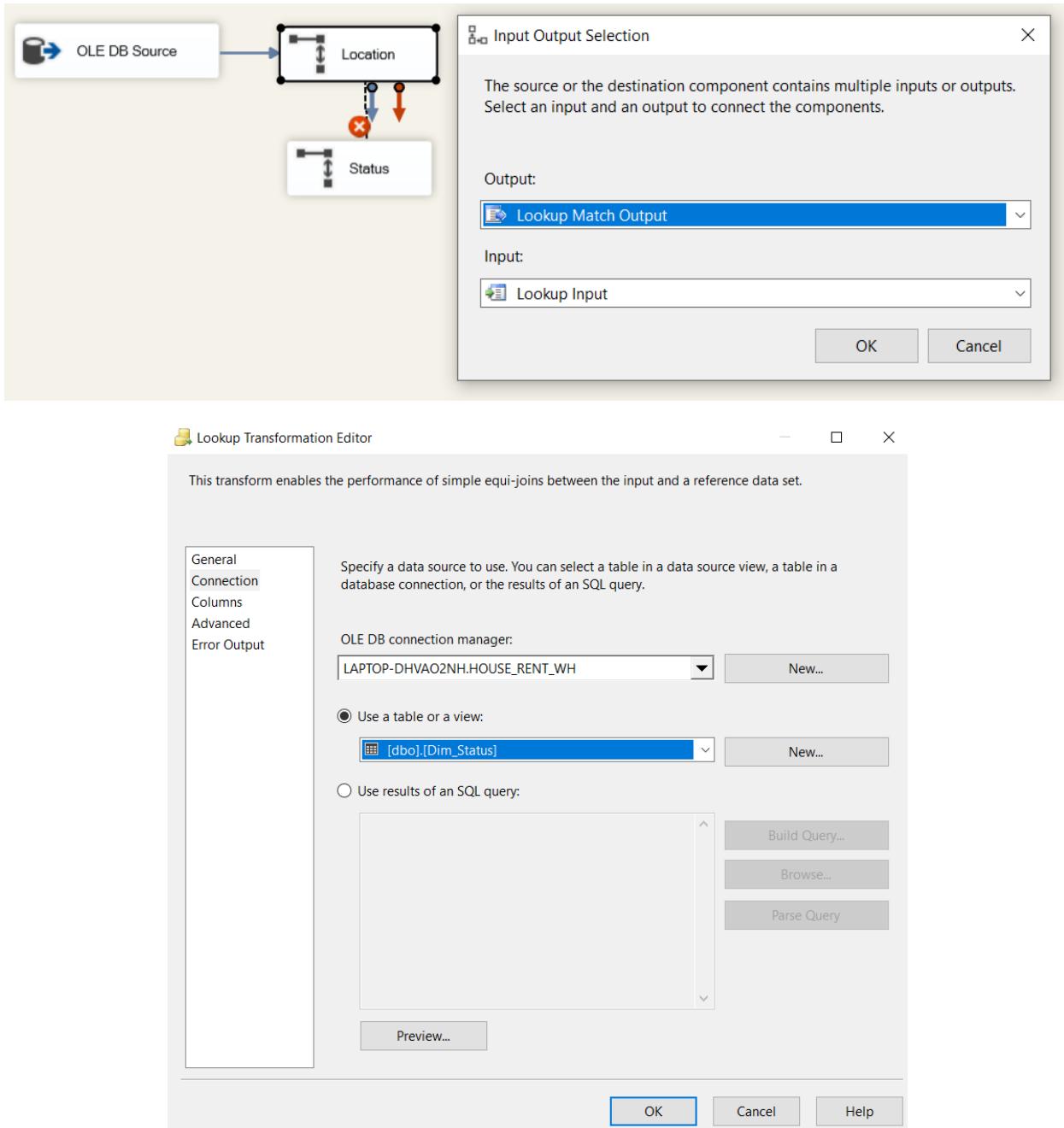




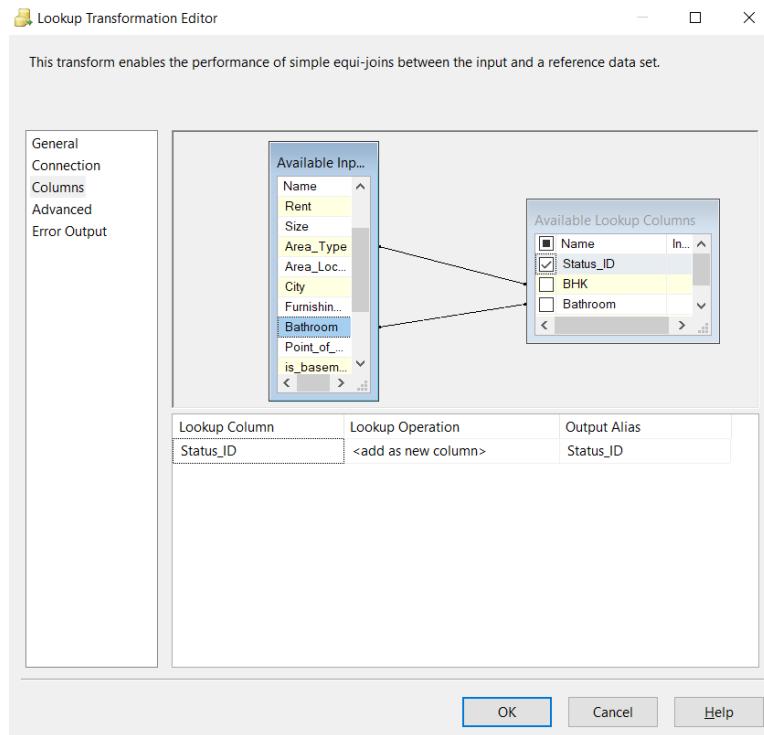
- Kiểm tra, liên kết các thuộc tính, nhấn OK để hoàn tất.



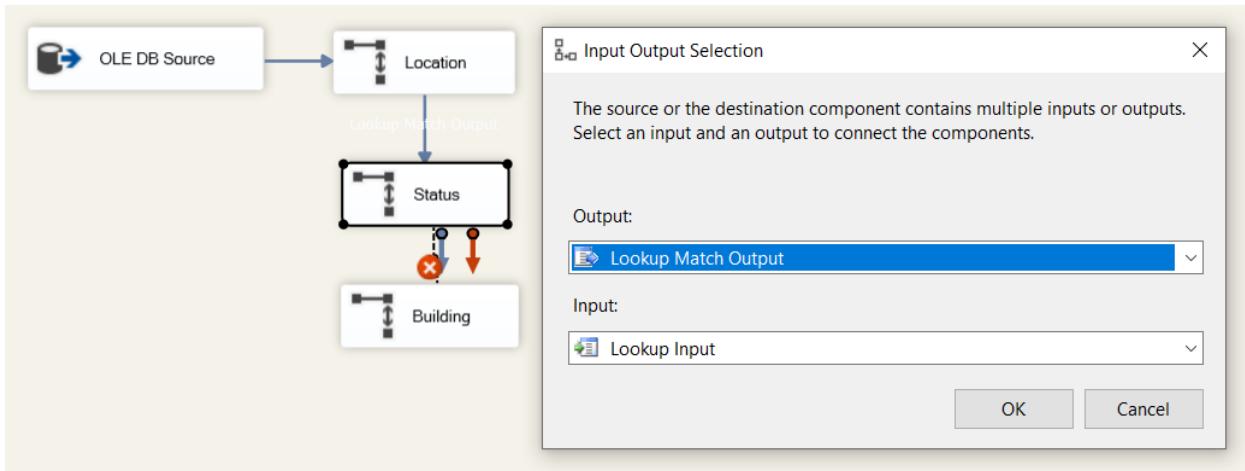
- Kéo thả tạo mới Look up Status

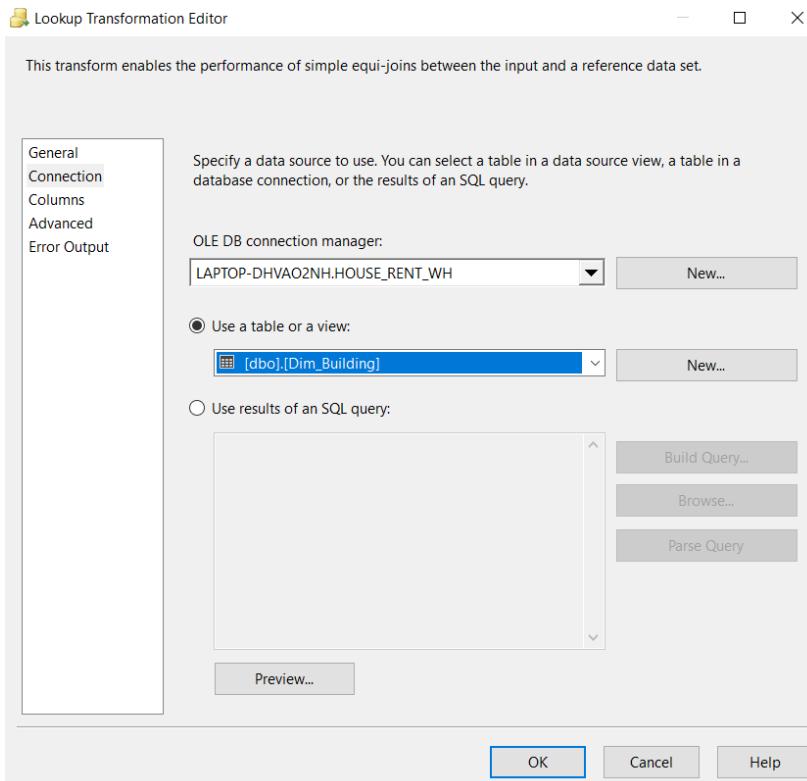


- Kiểm tra, liên kết các thuộc tính, nhấn OK để hoàn tất

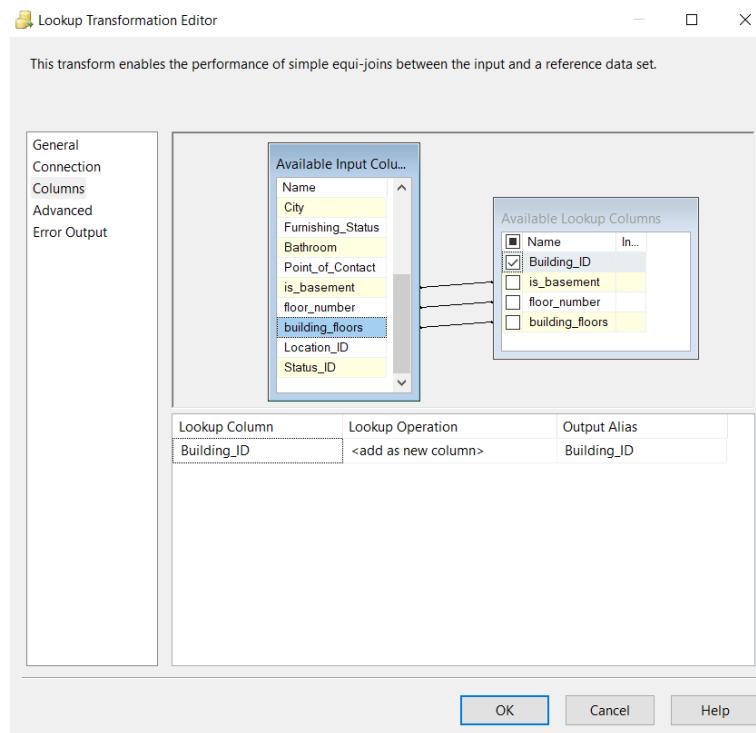


- Kéo thả tạo mới Look up Building

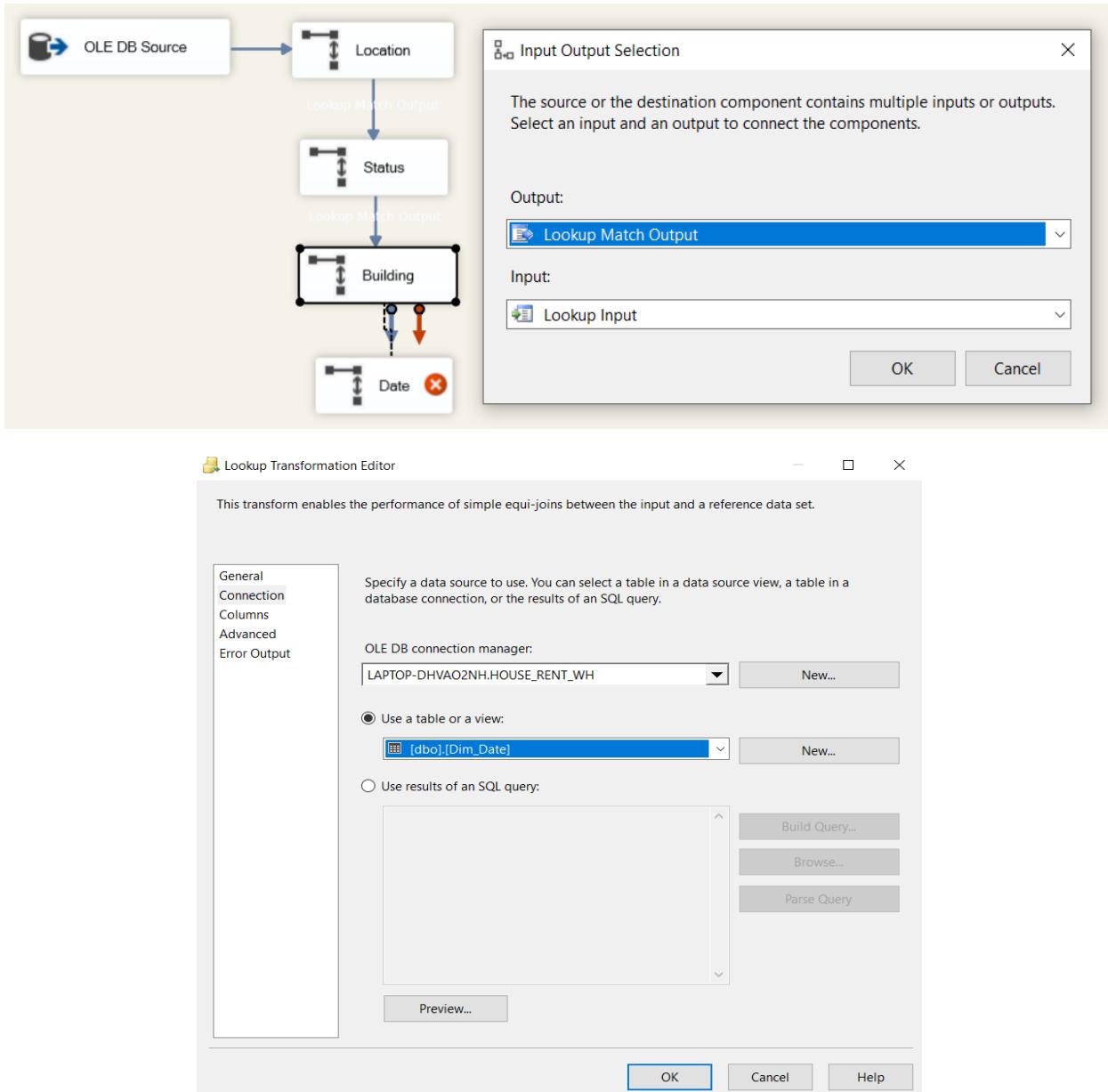




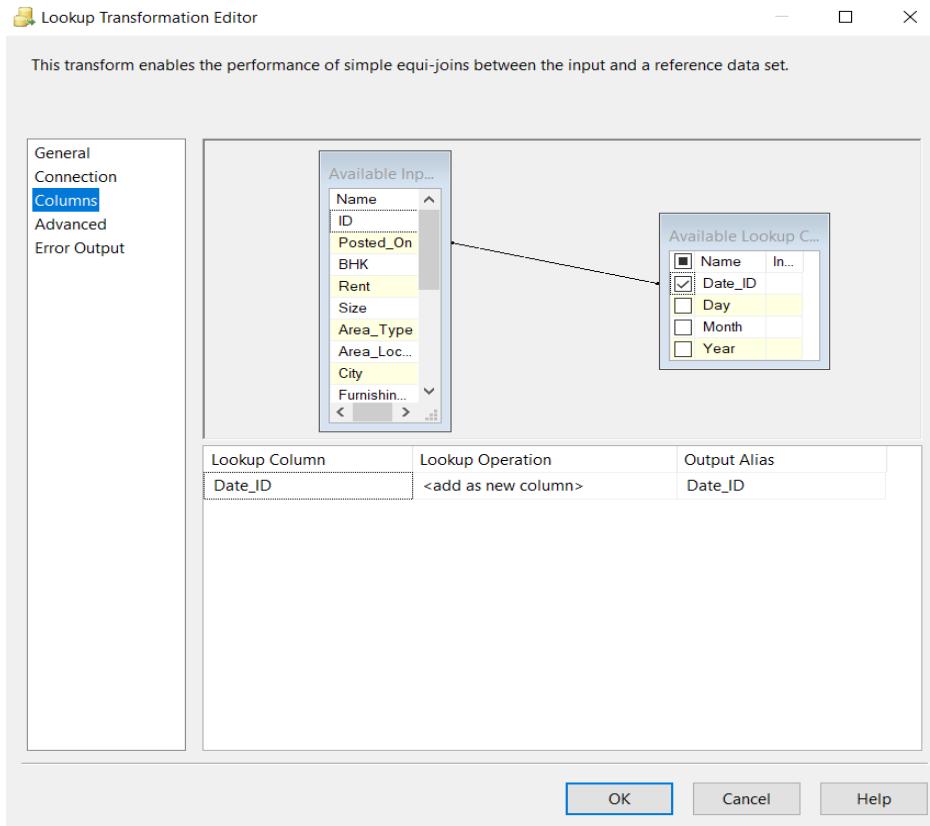
- Kiểm tra, liên kết các thuộc tính, nhấn OK để hoàn tất.



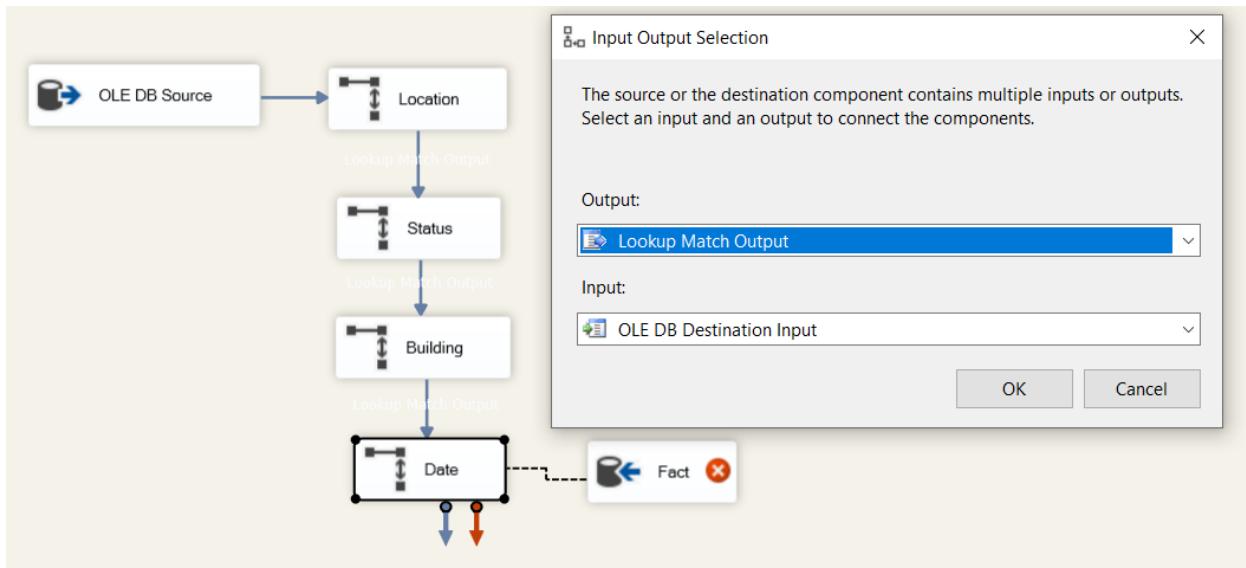
- Kéo thả tạo mới Look up Date



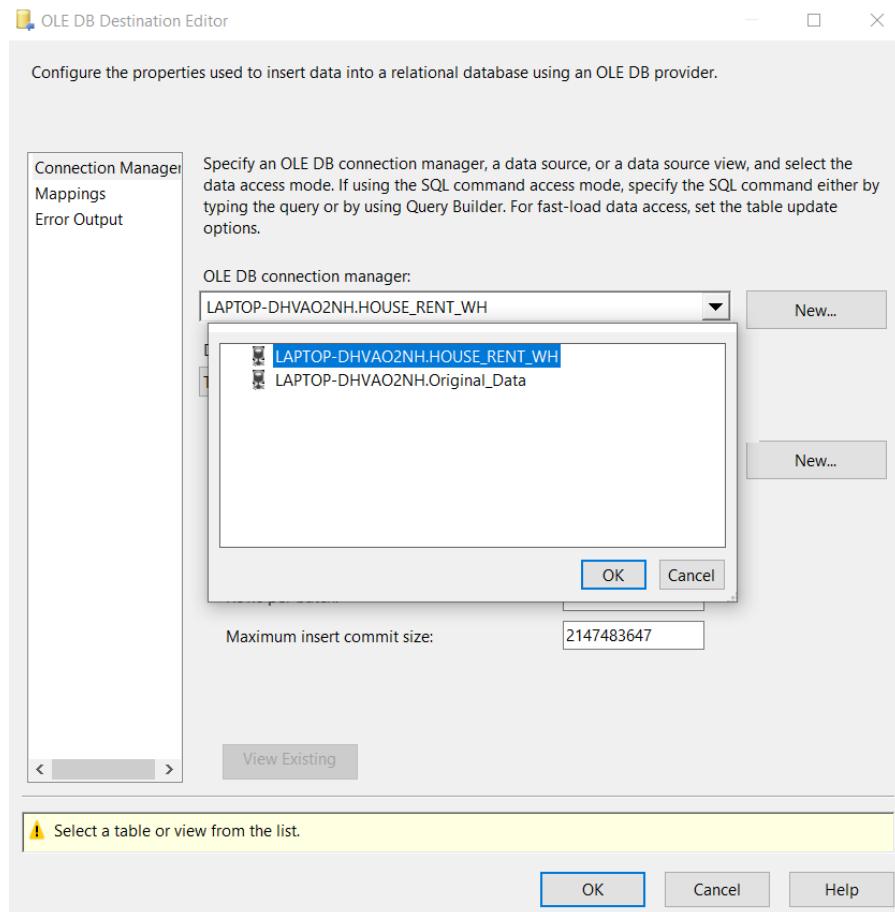
- Kiểm tra, liên kết các thuộc tính, nhấn OK để hoàn tất.

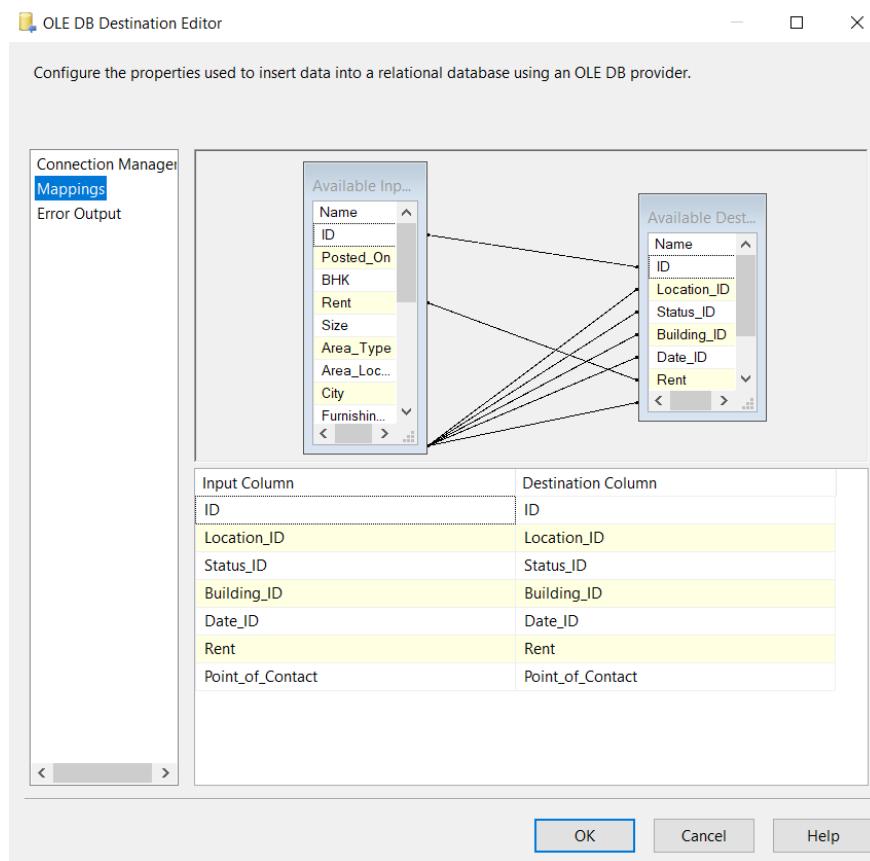
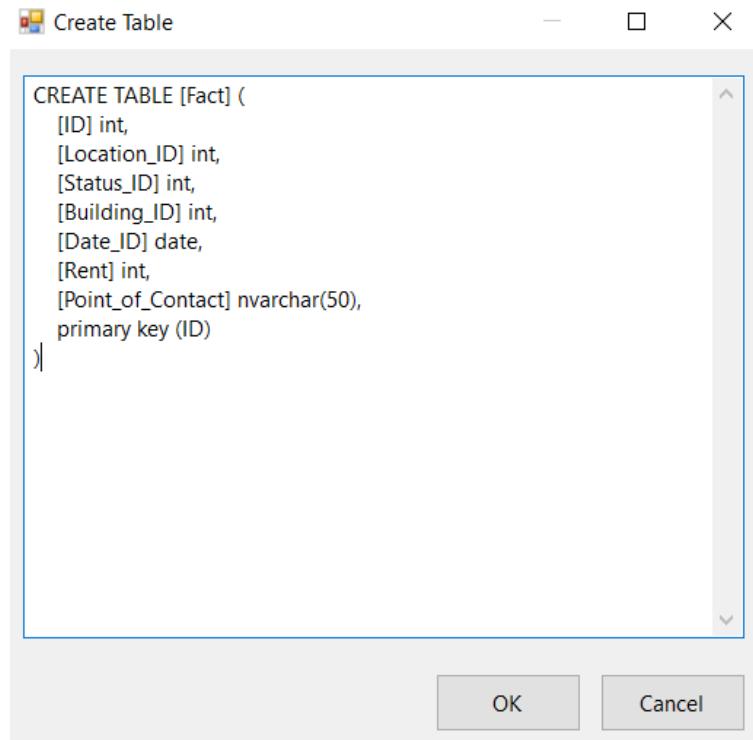


- Kéo thả OLE DB Destination tạo bảng Fact

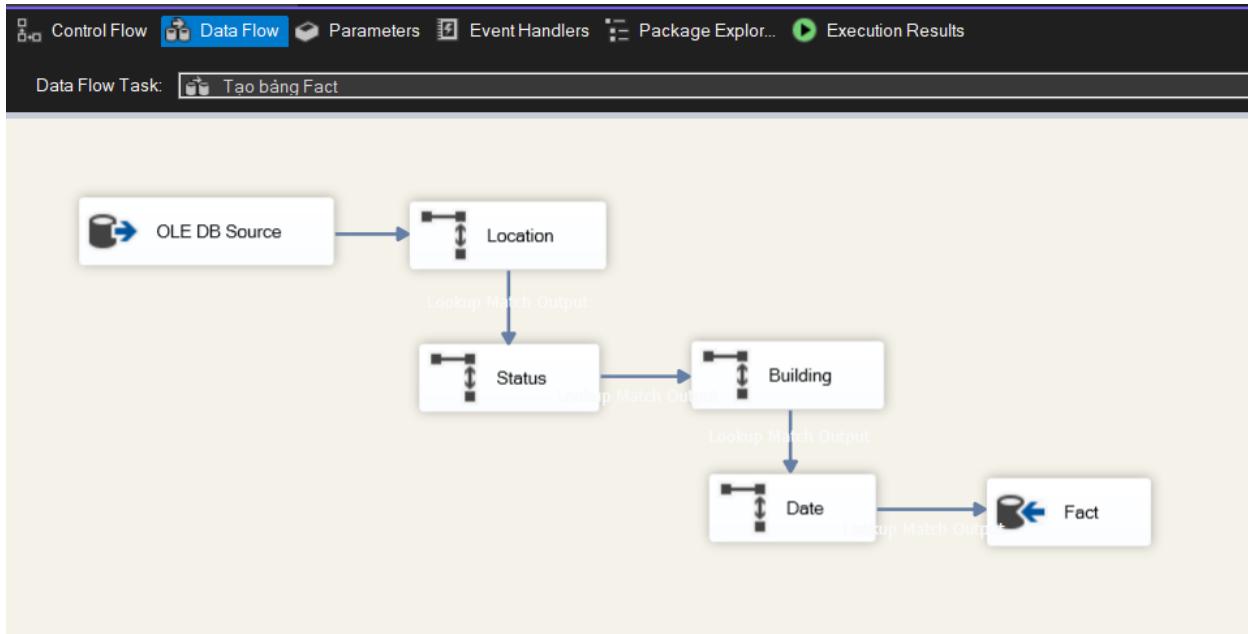


- Chọn House_Rent_WH.

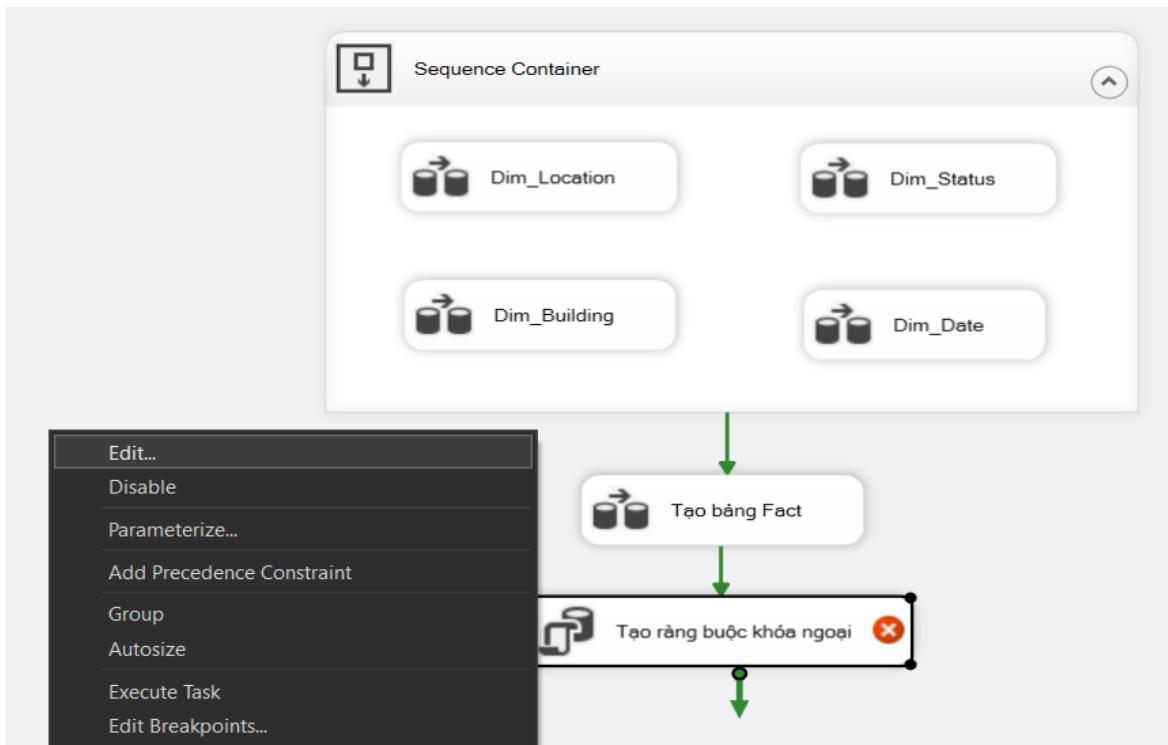


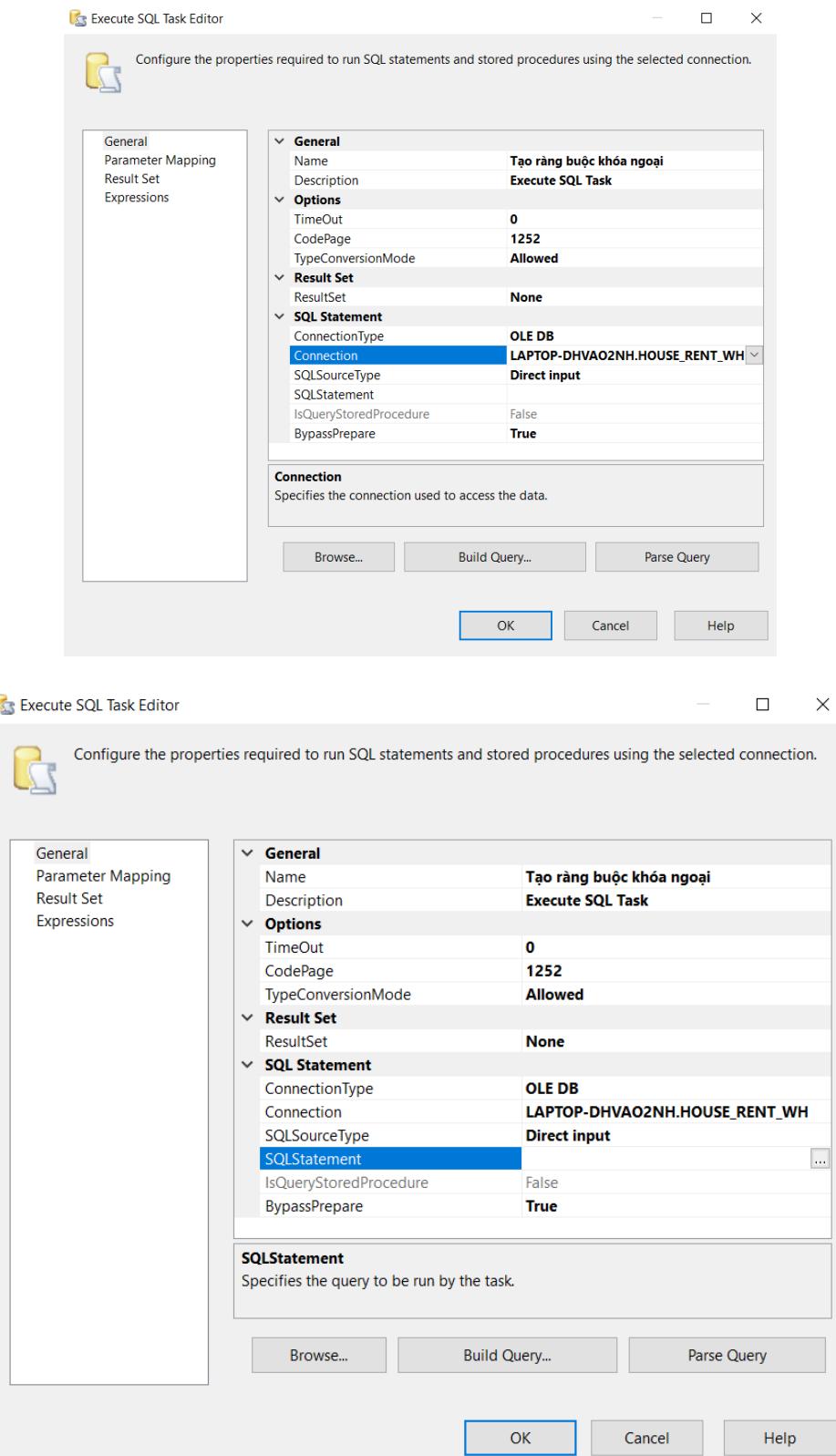


- Data Flow quá trình tạo bảng Fact sau khi hoàn thành

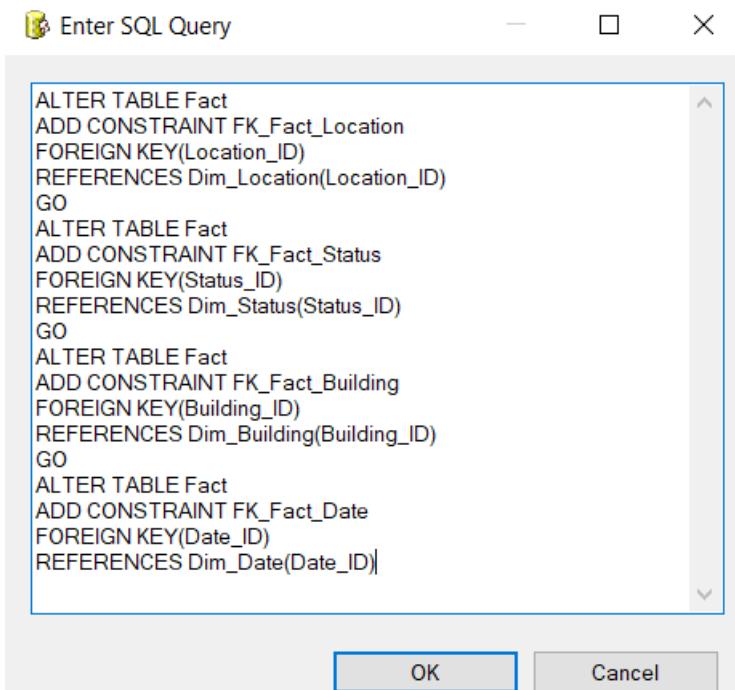


2.7. Tạo các ràng buộc khóa ngoại cho bảng Fact

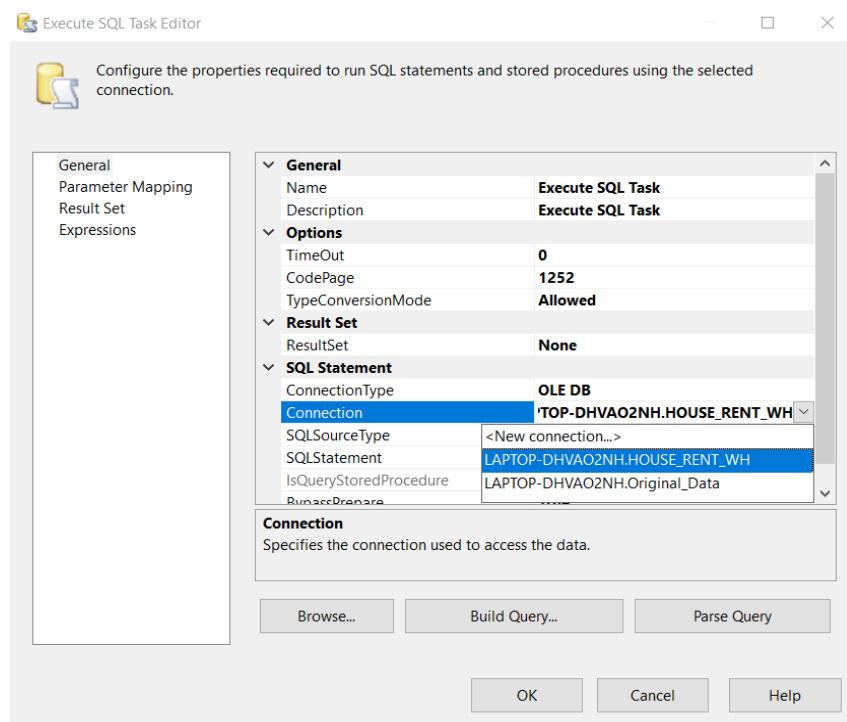
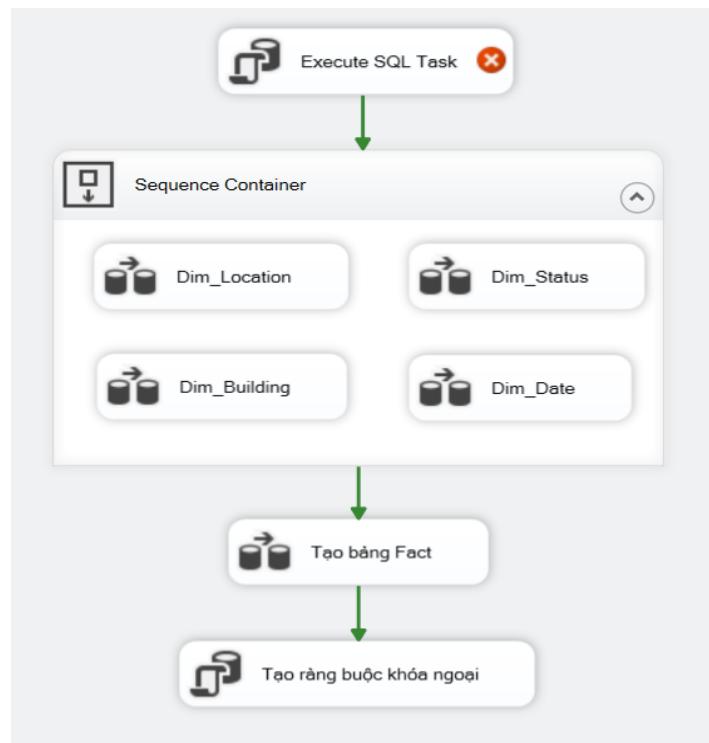


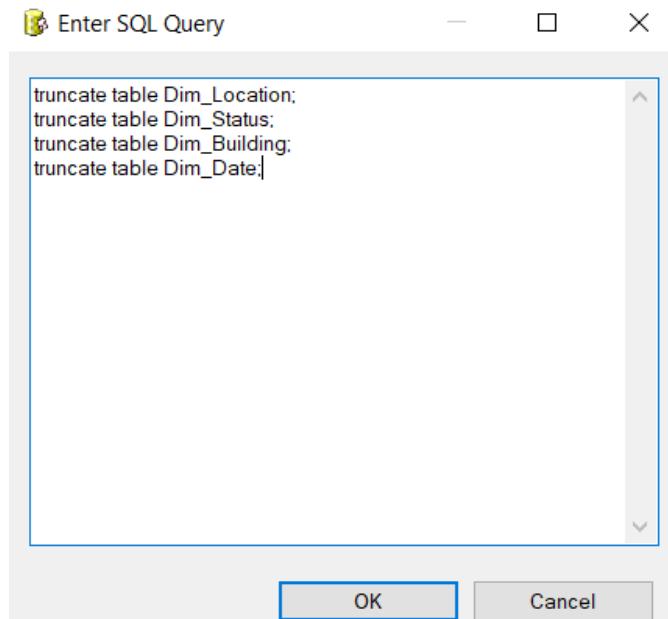


- Chọn SQL Statement

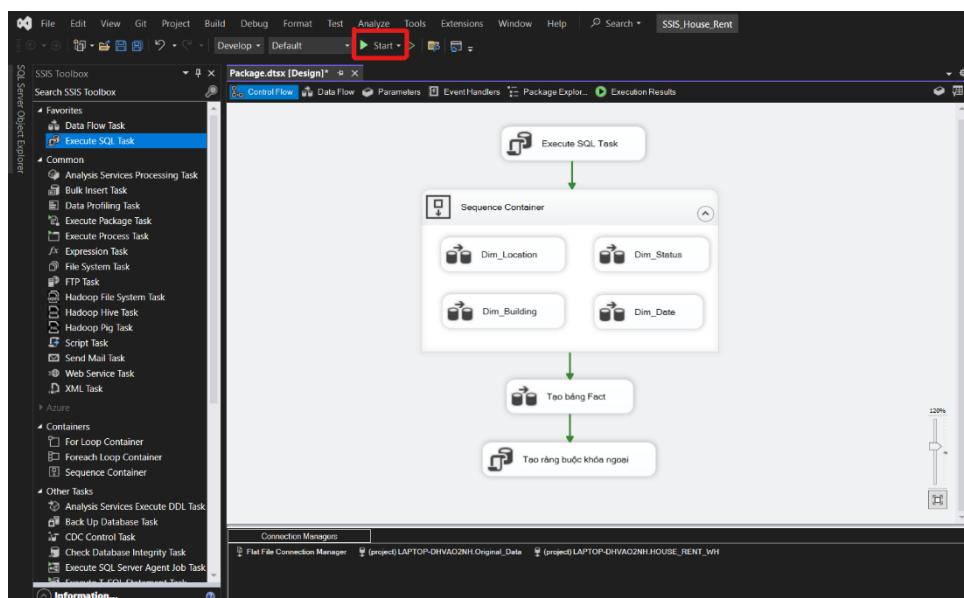


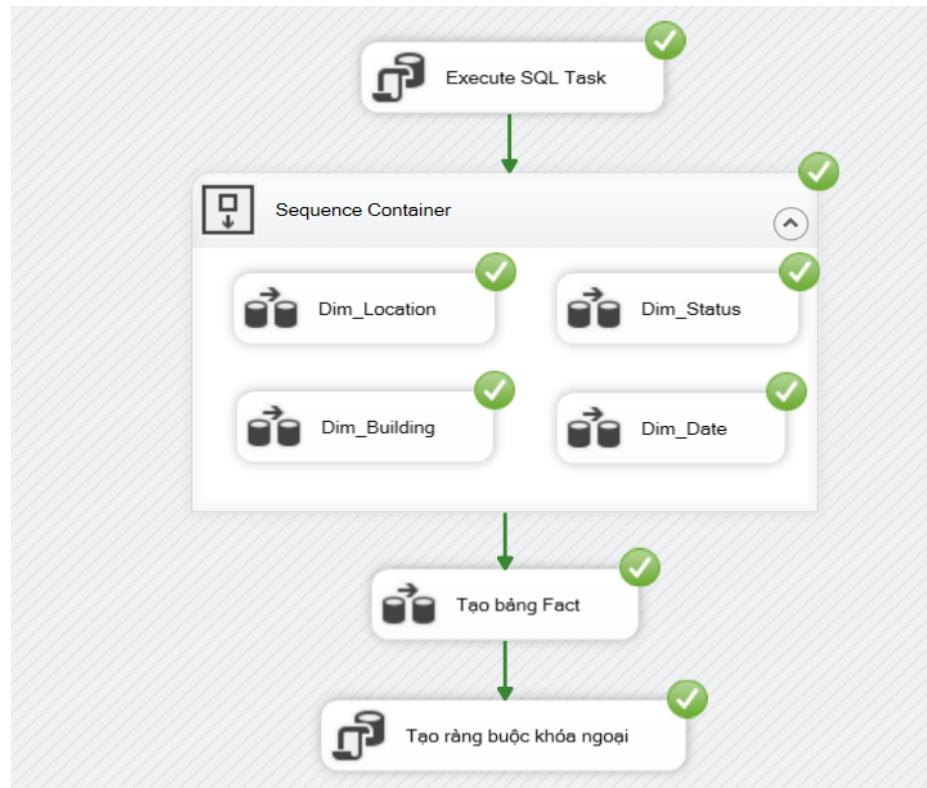
- Thêm một Execute SQL Task nhằm thực hiện nhiệm vụ đảm bảo dữ liệu mới hoàn toàn không bị chồng chéo dữ liệu cũ mỗi khi chạy project, trước quá trình chia bảng Fact và Dimension.





2.8. Chạy project và kiểm tra dữ liệu





- Khi chạy project thành công, khóa ngoại của bảng Fact tham chiếu đến các bảng Dimension được tạo. Tiến hành thêm các câu lệnh SQL để xóa các khóa ngoại cho những lần chạy tiếp theo.

Enter SQL Query

```

ALTER TABLE Fact
DROP CONSTRAINT FK_Fact_Location;
ALTER TABLE Fact
DROP CONSTRAINT FK_Fact_Status;
ALTER TABLE Fact
DROP CONSTRAINT FK_Fact_Building;
ALTER TABLE Fact
DROP CONSTRAINT FK_Fact_Date;
truncate table Fact;
truncate table Dim_Location;
truncate table Dim_Status;
truncate table Dim_Building;
truncate table Dim_Date;
  
```

OK Cancel

- Kiểm tra dữ liệu bảng Dim_Location.

| | Location_ID | City | Area_Locality | Area_Type |
|----|-------------|-----------|--|-------------|
| 1 | 1 | Bangalore | Vijay Enclave | Super Area |
| 2 | 2 | Bangalore | Lingarajapuram, Lingarajapuram, Hennur Main Road | Super Area |
| 3 | 3 | Bangalore | Vishweshwariah Layout, Annapurneshwari Nagar | Carpet Area |
| 4 | 4 | Bangalore | Amrutahalli | Super Area |
| 5 | 5 | Bangalore | Jnana Ganga Nagar | Super Area |
| 6 | 6 | Bangalore | Electronics City Phase 1, Electronic City | Carpet Area |
| 7 | 7 | Bangalore | Jyothi Nagar, Dooravani Nagar | Super Area |
| 8 | 8 | Bangalore | whitefield | Super Area |
| 9 | 9 | Bangalore | Basavanagudi | Carpet Area |
| 10 | 10 | Bangalore | Bannerghatta Main Road | Carpet Area |
| 11 | 11 | Bangalore | Srinivaspura | Super Area |
| 12 | 12 | Bangalore | Mallathahalli, Outer Ring Road | Super Area |
| 13 | 13 | Bangalore | Shampura, Kaval Byrasandra | Super Area |
| 14 | 14 | Bangalore | Margondanahalli | Super Area |
| 15 | 15 | Bangalore | Mudhaliar layout | Carpet Area |
| 16 | 16 | Bangalore | Airport Road | Super Area |
| 17 | 17 | Bangalore | Fci Layout, Deepanjali Nagar | Carpet Area |
| 18 | 18 | Bangalore | Kaggadasapura, Indira nagar | Carpet Area |
| 19 | 19 | Bangalore | Madanayakahalli | Super Area |
| 20 | 20 | Bangalore | Rmv Extension, Armane Nagar | Super Area |
| 21 | 21 | Bangalore | Banaswadi | Super Area |
| 22 | 22 | Bangalore | Doddaballapur, Electronics City | Carpet Area |

✓ Query executed successfully.

- Kiểm tra dữ liệu bảng Dim_Status.

| | Status_ID | BHK | Bathroom | Furnishing_Status |
|----|-----------|-----|----------|-------------------|
| 1 | 1 | 2 | 1 | Unfurnished |
| 2 | 2 | 4 | 1 | Unfurnished |
| 3 | 3 | 2 | 1 | Semi-Furnished |
| 4 | 4 | 3 | 1 | Semi-Furnished |
| 5 | 5 | 2 | 1 | Furnished |
| 6 | 6 | 1 | 1 | Unfurnished |
| 7 | 7 | 3 | 1 | Unfurnished |
| 8 | 8 | 3 | 1 | Furnished |
| 9 | 9 | 1 | 1 | Furnished |
| 10 | 10 | 1 | 1 | Semi-Furnished |
| 11 | 11 | 3 | 2 | Unfurnished |
| 12 | 12 | 4 | 2 | Semi-Furnished |
| 13 | 13 | 4 | 2 | Furnished |
| 14 | 14 | 4 | 2 | Unfurnished |
| 15 | 15 | 1 | 2 | Semi-Furnished |
| 16 | 16 | 2 | 2 | Furnished |
| 17 | 17 | 2 | 2 | Unfurnished |
| 18 | 18 | 1 | 2 | Unfurnished |
| 19 | 19 | 2 | 2 | Semi-Furnished |

✓ Query executed successfully.

- Kiểm tra dữ liệu bảng Building.

| | Building_ID | is_basement | floor_number | building_floors |
|----|-------------|-------------|--------------|-----------------|
| 1 | 1 | 0 | 1 | 1 |
| 2 | 2 | 0 | 2 | 3 |
| 3 | 3 | 0 | 5 | 6 |
| 4 | 4 | 0 | 2 | 5 |
| 5 | 5 | 0 | 3 | 4 |
| 6 | 6 | 0 | 4 | 6 |
| 7 | 7 | 0 | 2 | 3 |
| 8 | 8 | 0 | 1 | 5 |
| 9 | 9 | 0 | 0 | 2 |
| 10 | 10 | 0 | 3 | 3 |
| 11 | 11 | 0 | 0 | 3 |
| 12 | 12 | 0 | 0 | 1 |
| 13 | 13 | 0 | 2 | 3 |
| 14 | 14 | 0 | 1 | 5 |
| 15 | 15 | 0 | 1 | 3 |
| 16 | 16 | 0 | 1 | 2 |
| 17 | 17 | 0 | 1 | 4 |
| 18 | 18 | 0 | 2 | 4 |
| 19 | 19 | 0 | 2 | 5 |
| 20 | 20 | 0 | 3 | 4 |
| 21 | 21 | 0 | 4 | 5 |
| 22 | 22 | 0 | 0 | 4 |

✓ Query executed successfully.

- Kiểm tra dữ liệu bảng Dim_Date.

| | Date_ID | Day | Month | Year |
|----|------------|-----|-------|------|
| 1 | 2022-04-13 | 13 | 4 | 2022 |
| 2 | 2022-04-23 | 23 | 4 | 2022 |
| 3 | 2022-04-24 | 24 | 4 | 2022 |
| 4 | 2022-04-25 | 25 | 4 | 2022 |
| 5 | 2022-04-26 | 26 | 4 | 2022 |
| 6 | 2022-04-27 | 27 | 4 | 2022 |
| 7 | 2022-04-28 | 28 | 4 | 2022 |
| 8 | 2022-04-29 | 29 | 4 | 2022 |
| 9 | 2022-04-30 | 30 | 4 | 2022 |
| 10 | 2022-05-01 | 1 | 5 | 2022 |
| 11 | 2022-05-02 | 2 | 5 | 2022 |
| 12 | 2022-05-03 | 3 | 5 | 2022 |
| 13 | 2022-05-04 | 4 | 5 | 2022 |
| 14 | 2022-05-05 | 5 | 5 | 2022 |
| 15 | 2022-05-06 | 6 | 5 | 2022 |
| 16 | 2022-05-07 | 7 | 5 | 2022 |
| 17 | 2022-05-08 | 8 | 5 | 2022 |
| 18 | 2022-05-09 | 9 | 5 | 2022 |
| 19 | 2022-05-10 | 10 | 5 | 2022 |
| 20 | 2022-05-11 | 11 | 5 | 2022 |
| 21 | 2022-05-12 | 12 | 5 | 2022 |
| 22 | 2022-05-13 | 13 | 5 | 2022 |
| 23 | 2022-05-14 | 14 | 5 | 2022 |

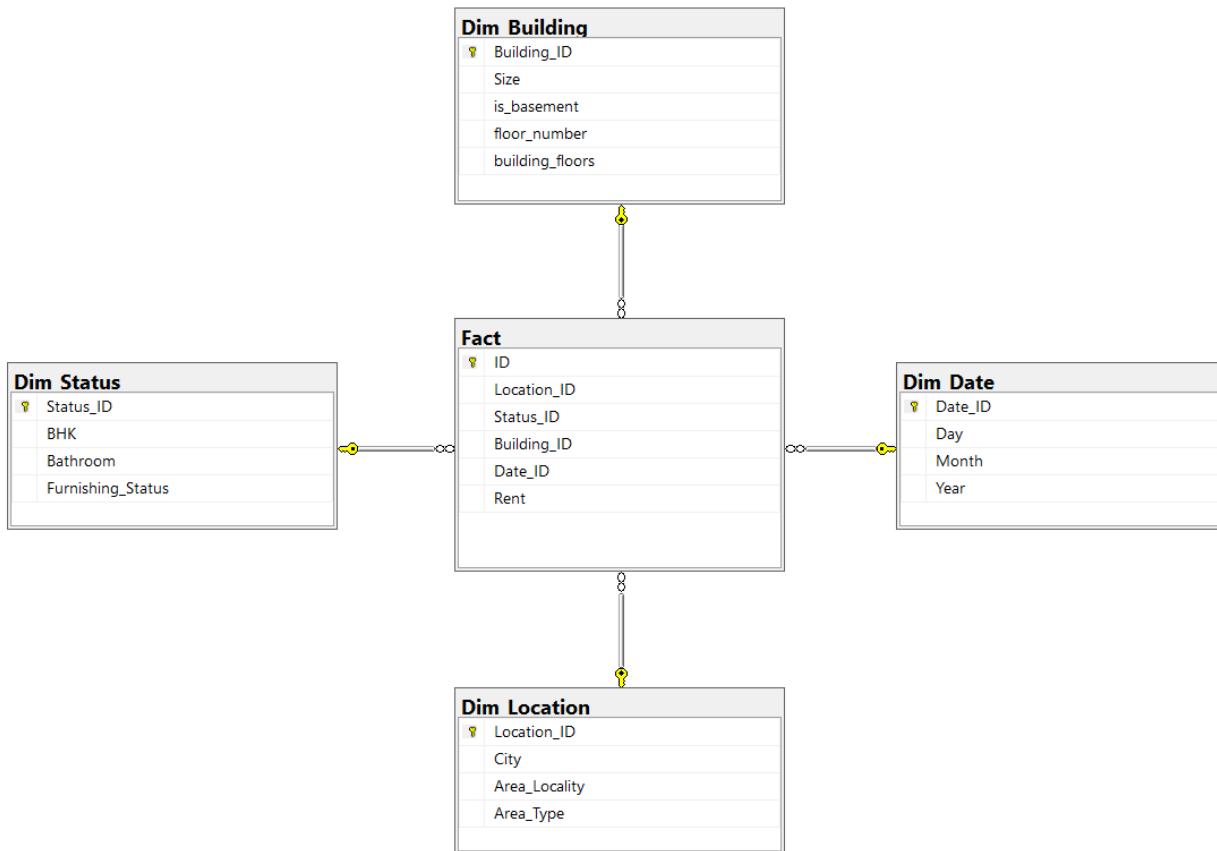
✓ Query executed successfully.

- Kiểm tra dữ liệu bảng Fact.

| | ID | Location_ID | Status_ID | Building_ID | Date_ID | Rent | Point_of_Contact |
|----|----|-------------|-----------|-------------|------------|-------|------------------|
| 1 | 1 | 1879 | 16 | 9 | 2022-05-18 | 10000 | Contact Owner |
| 2 | 2 | 1727 | 1 | 15 | 2022-05-13 | 20000 | Contact Owner |
| 3 | 3 | 1755 | 1 | 15 | 2022-05-16 | 17000 | Contact Owner |
| 4 | 4 | 1969 | 1 | 16 | 2022-07-04 | 10000 | Contact Owner |
| 5 | 5 | 1843 | 1 | 16 | 2022-05-09 | 7500 | Contact Owner |
| 6 | 6 | 1798 | 16 | 12 | 2022-04-29 | 7000 | Contact Owner |
| 7 | 7 | 1720 | 16 | 22 | 2022-06-21 | 10000 | Contact Agent |
| 8 | 8 | 1720 | 6 | 16 | 2022-06-21 | 5000 | Contact Agent |
| 9 | 9 | 1954 | 16 | 16 | 2022-06-07 | 26000 | Contact Agent |
| 10 | 10 | 1931 | 16 | 15 | 2022-06-20 | 10000 | Contact Owner |
| 11 | 11 | 1858 | 11 | 17 | 2022-05-23 | 25000 | Contact Agent |
| 12 | 12 | 1845 | 6 | 1 | 2022-06-07 | 5000 | Contact Agent |
| 13 | 13 | 1867 | 6 | 17 | 2022-05-14 | 6500 | Contact Owner |
| 14 | 14 | 1729 | 6 | 16 | 2022-05-09 | 5500 | Contact Agent |
| 15 | 15 | 1752 | 11 | 9 | 2022-05-05 | 8500 | Contact Owner |
| 16 | 16 | 1941 | 11 | 1 | 2022-06-01 | 40000 | Contact Owner |
| 17 | 17 | 1913 | 1 | 16 | 2022-05-17 | 6000 | Contact Owner |
| 18 | 18 | 1870 | 1 | 9 | 2022-06-20 | 10000 | Contact Owner |
| 19 | 19 | 1710 | 1 | 11 | 2022-06-09 | 11000 | Contact Owner |

✓ Query executed successfully.

Diagram đồ án sau khi hoàn thành:

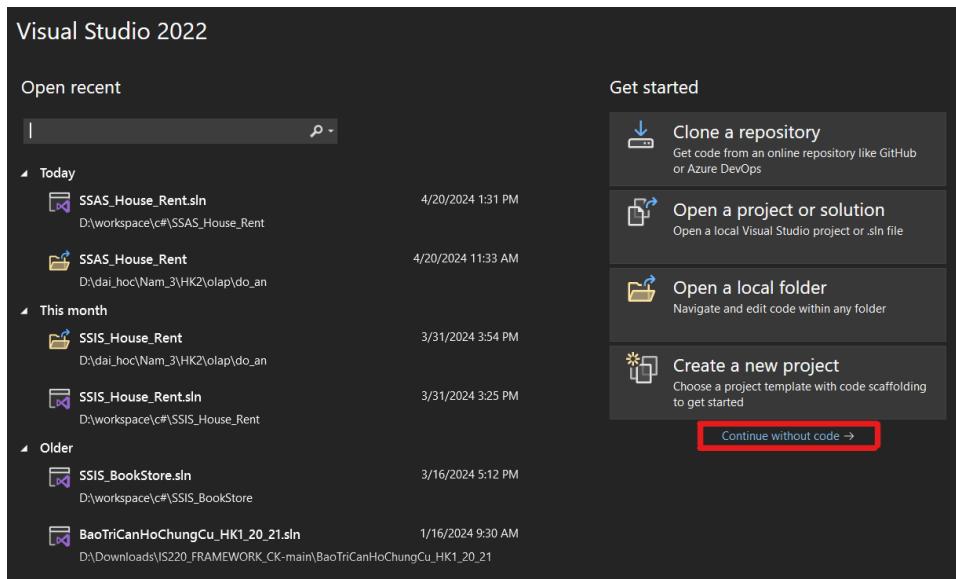


CHƯƠNG 3. PHÂN TÍCH KHO DỮ LIỆU (SSAS)

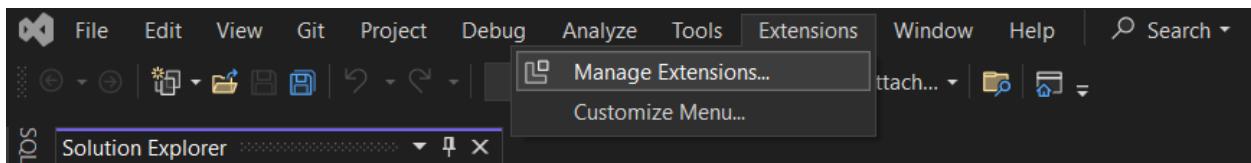
3.1. Chuẩn bị các công cụ

3.1.1. Cài đặt Microsoft Analysis Services Projects

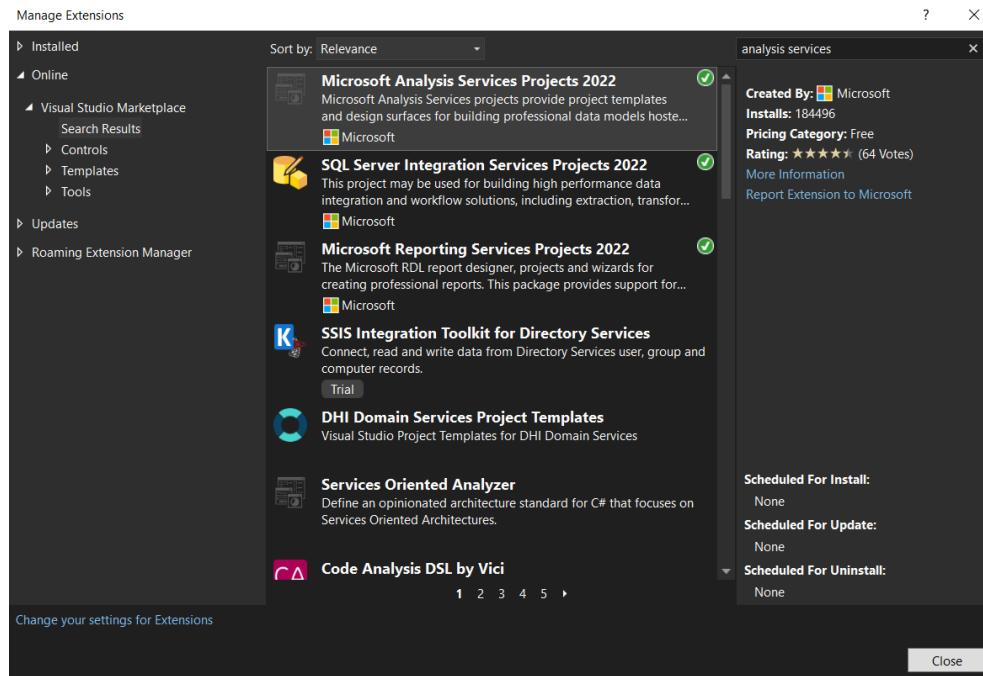
- Vào giao diện code của Visual Studio



- Chọn Extensions -> Manage Extensions



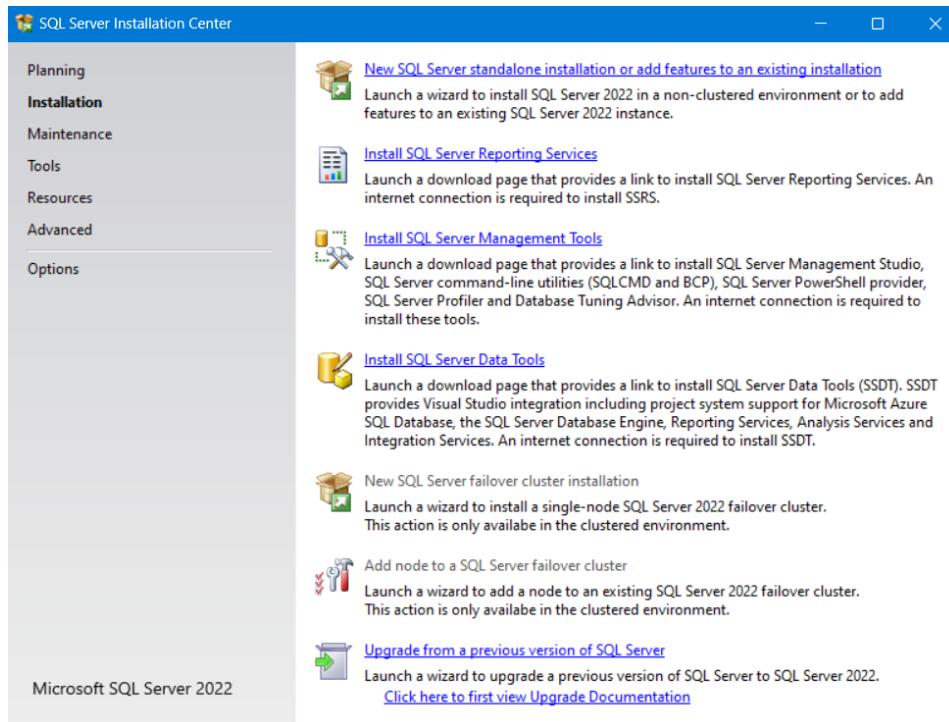
- Tìm kiếm Analysis Services và tiến hành download



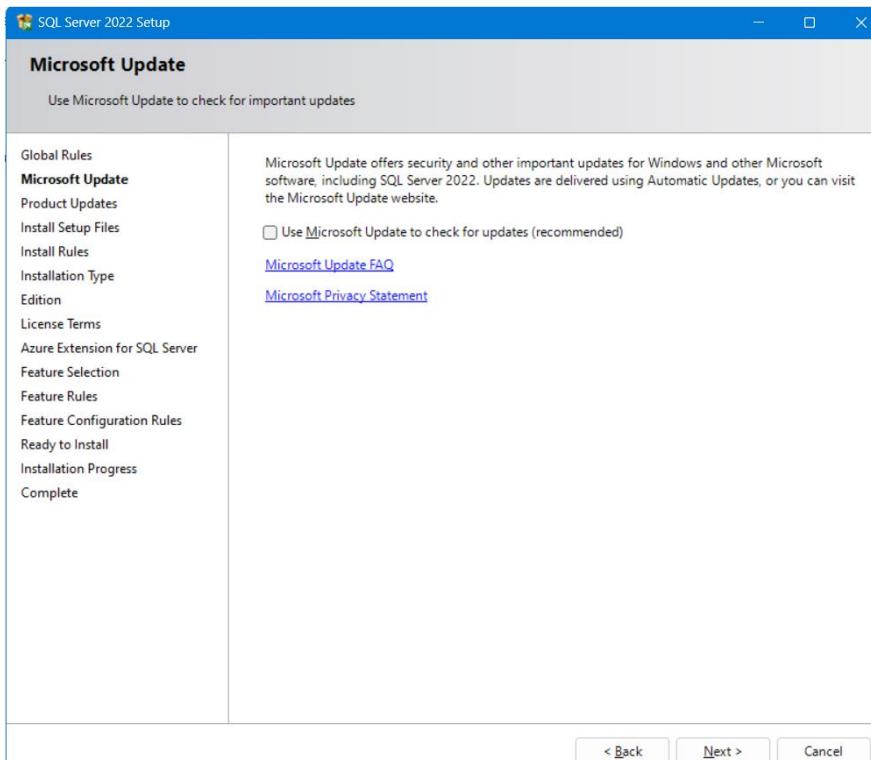
- Khi download xong nhấn đúp vào file đã download để tiến hành cài đặt Microsoft Analysis Services Projects.

3.1.2. Cài đặt Analysis Services

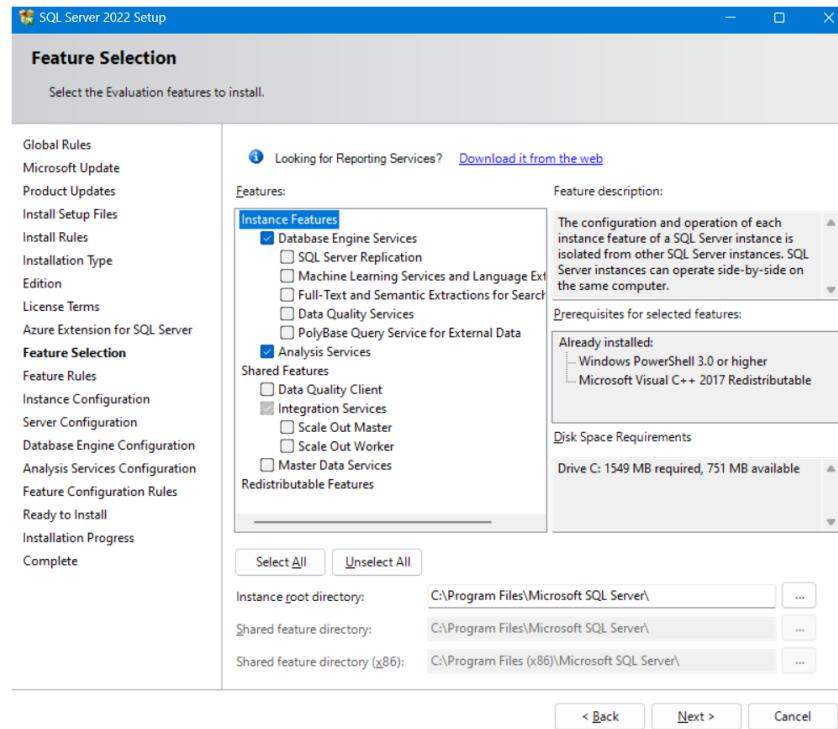
- Mở File SETUP.EXE ở thư mục đã cài SQL SERVER, chọn mục Installation, tiếp theo chọn “New SQL Server standalone installation or add features to an existing installation”.



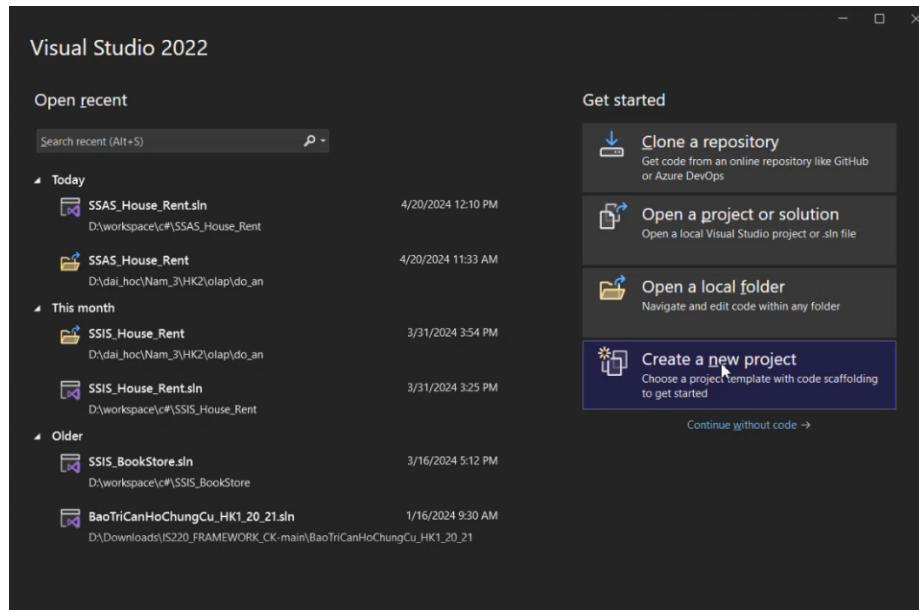
- Hộp thoại hiển thị như sau

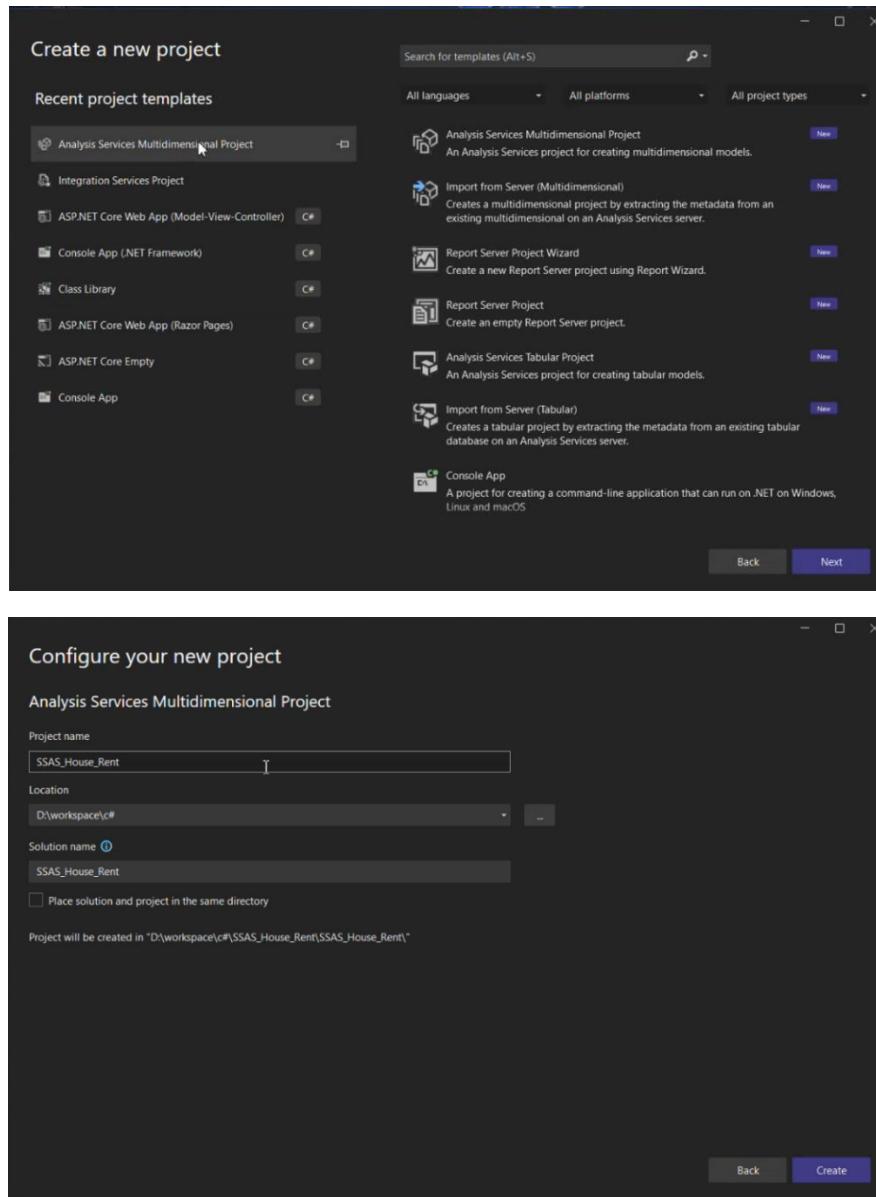


- Tiếp tục nhấn Next, chọn các thông tin server thích hợp và đến mục Feature Selection tích chọn Analysis Services. Và tiến hành cài đặt.

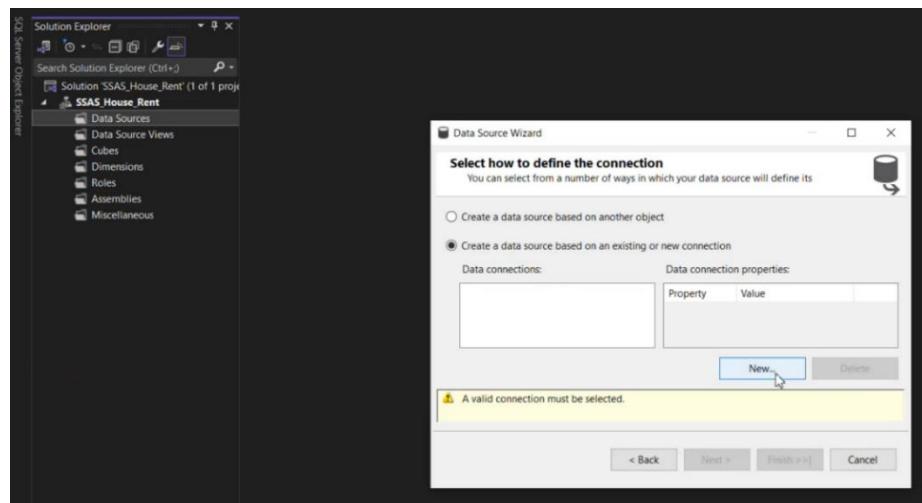
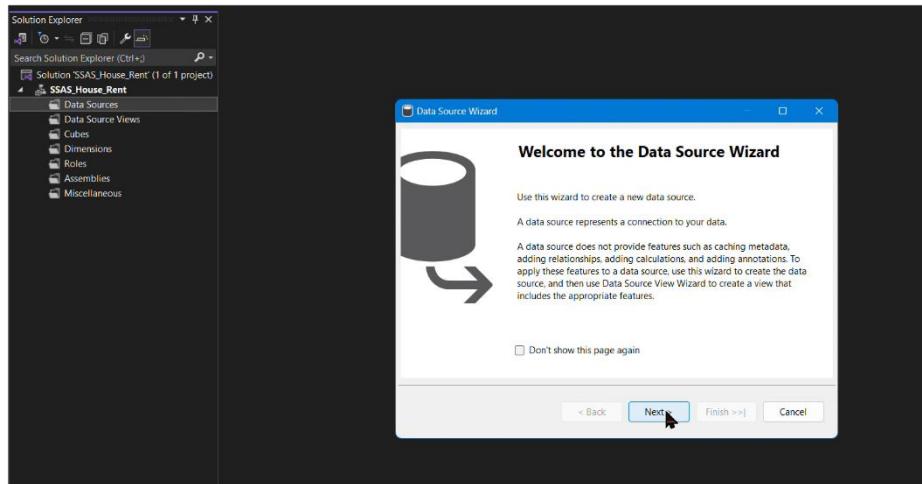


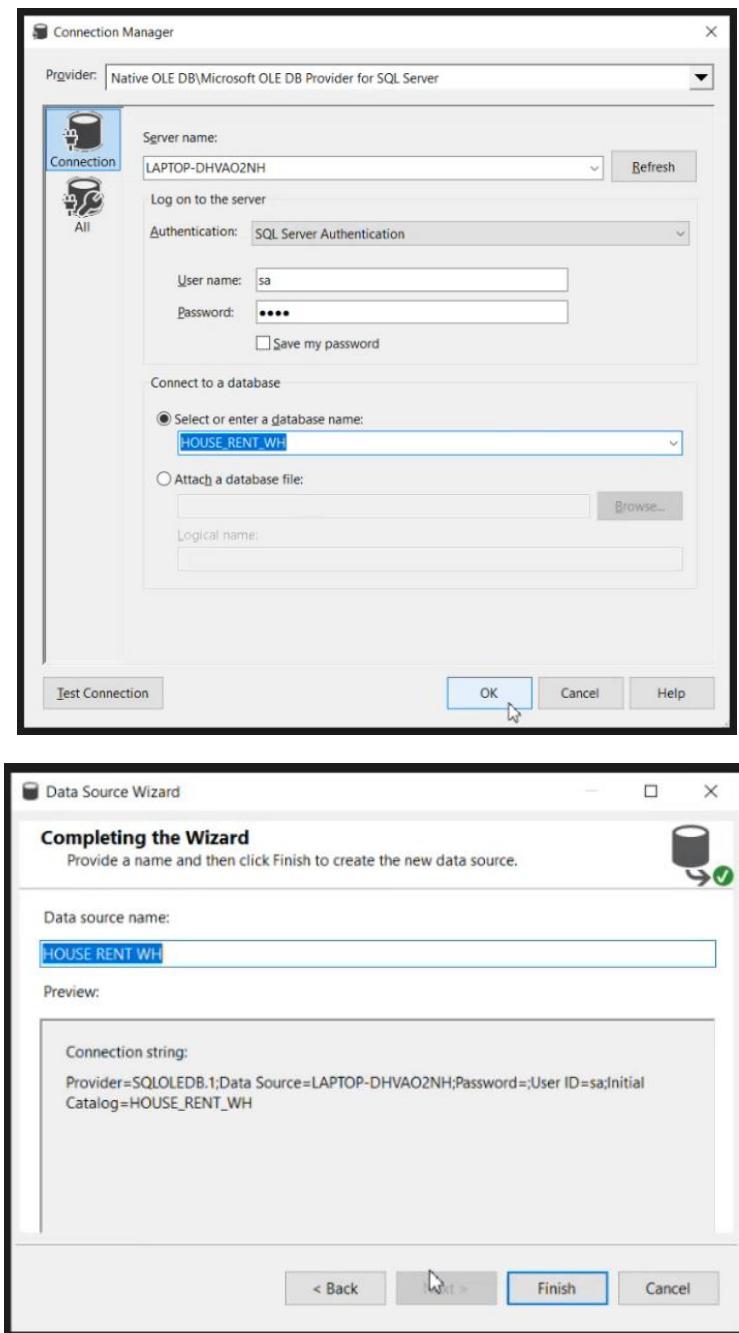
3.2. Tạo mới project SSAS



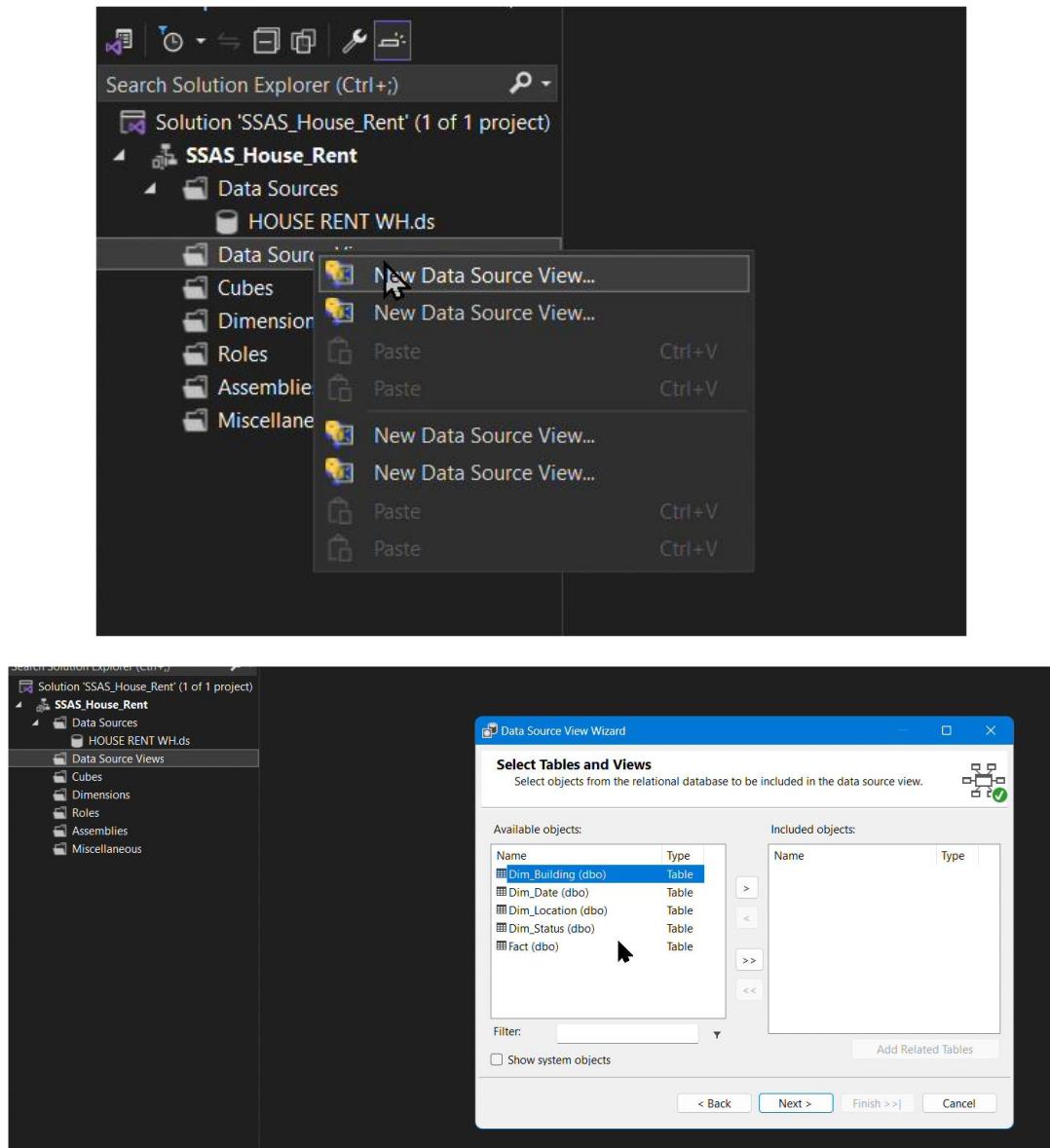


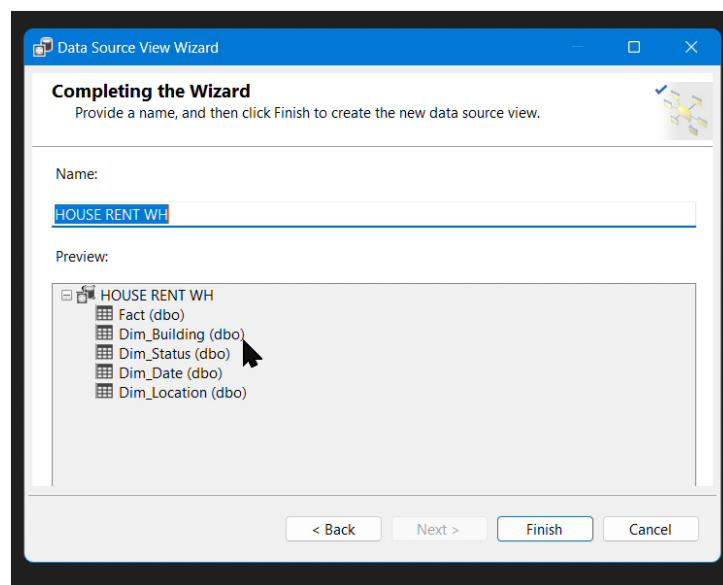
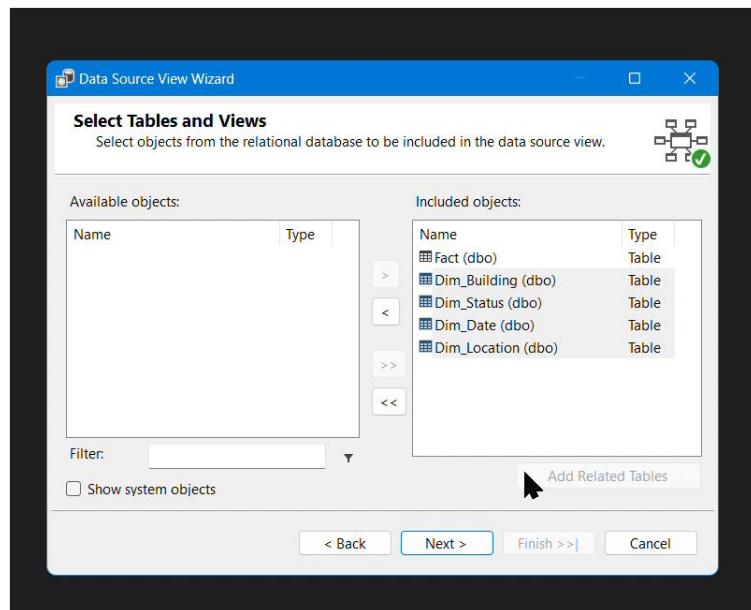
3.3. Xác định dữ liệu nguồn (Data Sources)

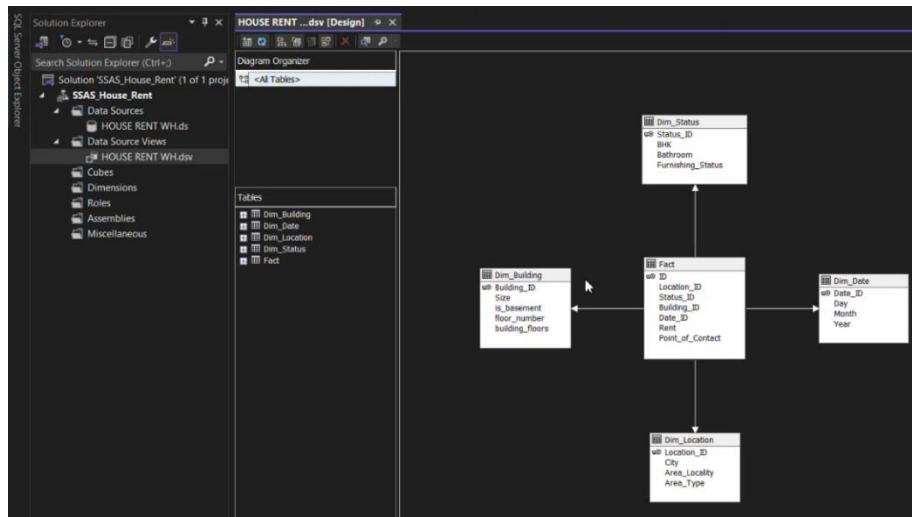




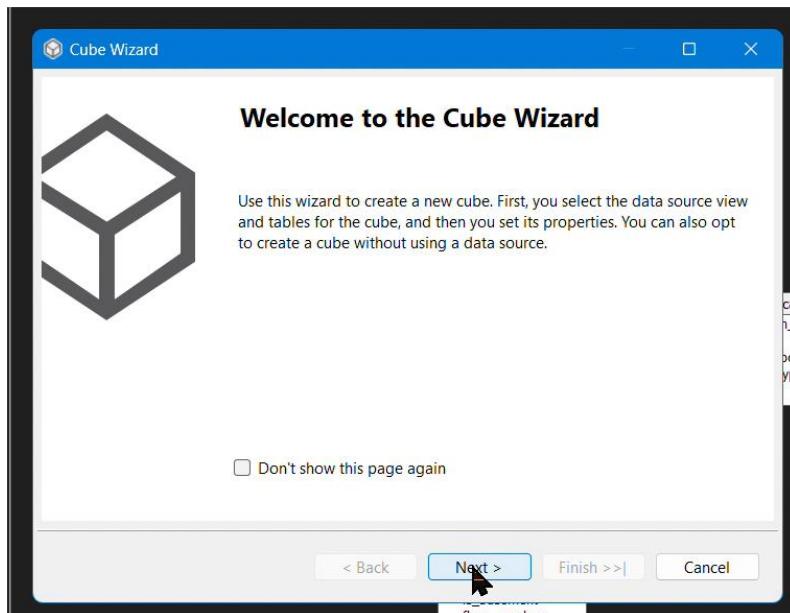
3.4. Xác định khung nhìn dữ liệu nguồn (Data Source Views)

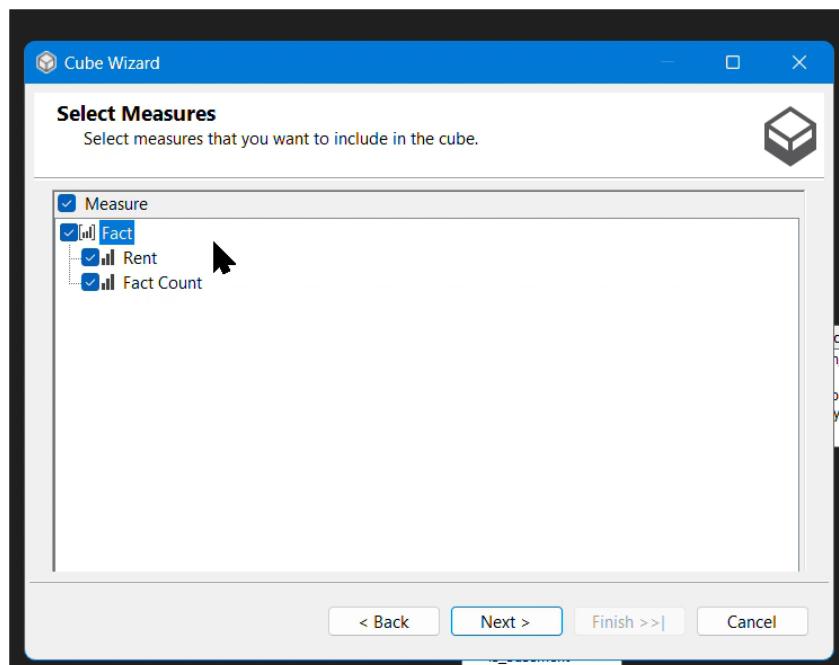
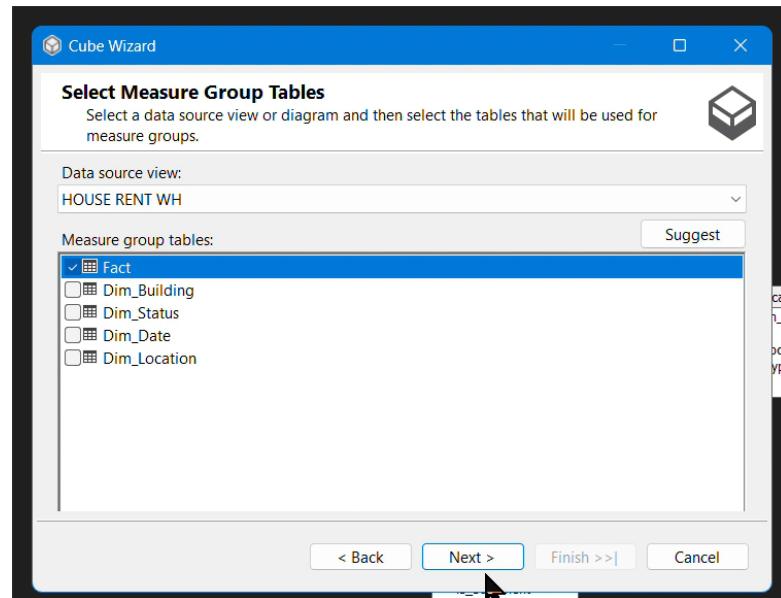


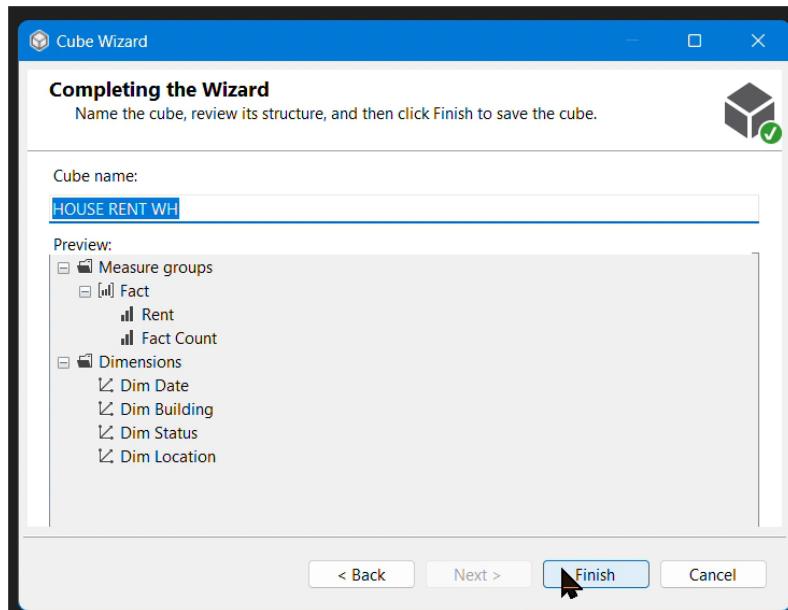




3.5. Xây dựng các khối (Cubes) và xác định các độ đo (Measures)







3.6. Xác định các chiều (Dimensions)

3.6.1. Dim_Date

| Dim Date.dim [Design] | | |
|--|--|------------------|
| Attributes | Hierarchies | Data Source View |
| <input checked="" type="checkbox"/> Dim Date <input type="checkbox"/> Date ID | To create a new hierarchy, drag an attribute here. | |

| Dim Date.dim [Design] | | |
|---|--|------------------|
| Attributes | Hierarchies | Data Source View |
| <input checked="" type="checkbox"/> Dim Date <input type="checkbox"/> Date ID <input type="checkbox"/> Day <input type="checkbox"/> Month <input type="checkbox"/> Year | To create a new hierarchy, drag an attribute here. | |

3.6.2. Dim_Building

The screenshot shows the Data Source View for the Dim_Building dimension. The interface is divided into three main sections: Attributes, Hierarchies, and Data Source View.

- Attributes:** A list of attributes for the Dim_Building dimension, including Building Floors, Building ID, Floor Number, Is Basement, and Size.
- Hierarchies:** A section for creating new hierarchies by dragging attributes here. It contains the placeholder text: "To create a new hierarchy, drag an attribute here."
- Data Source View:** A preview of the data source showing the Dim_Building table with columns: Building_ID, is_basement, floor_number, building_floors, and Size.

3.6.3. Dim_Status

The screenshot shows the Data Source View for the Dim_Status dimension. The interface is divided into three main sections: Attributes, Hierarchies, and Data Source View.

- Attributes:** A list of attributes for the Dim_Status dimension, including Status_ID.
- Hierarchies:** A section for creating new hierarchies by dragging attributes here. It contains the placeholder text: "To create a new hierarchy, drag an attribute here."
- Data Source View:** A preview of the data source showing the Dim_Status table with columns: Status_ID, BHK, Bathroom, and Furnishing_Status.

The screenshot shows the Data Source View for the Dim_Status dimension. The interface is divided into three main sections: Attributes, Hierarchies, and Data Source View.

- Attributes:** A list of attributes for the Dim_Status dimension, including Status_ID, Bathroom, BHK, and Furnishing_Status.
- Hierarchies:** A section for creating new hierarchies by dragging attributes here. It contains the placeholder text: "To create a new hierarchy, drag an attribute here."
- Data Source View:** A preview of the data source showing the Dim_Status table with columns: Status_ID, BHK, Bathroom, and Furnishing_Status.

3.6.4. Dim_Location

Dim Location.dim [Design] * Dim Status.dim [Design]* Dim Building.dim [Design]* Dim Date.dim [Design]*

Dimension Structure Attribute Relationships Translations Browser

Attributes Hierarchies Data Source View

To create a new hierarchy, drag an attribute here.

Dim_Location location_ID City Area_Locality Area_Type

Dim Location.dim [Design]* Dim Status.dim [Design]* Dim Building.dim [Design]* Dim Date.dim [Design]*

Dimension Structure Attribute Relationships Translations Browser

Attributes Hierarchies Data Source View

To create a new hierarchy, drag an attribute here.

Dim_Location location_ID City Area_Locality Area_Type

Deployment Progress - SSAS_House_Rent

Server : LAPTOP-DHVAO2NHNMSSQLAS
Database : SSAS_House_Rent

Command

- Processing Database 'SSAS_House_Rent' completed.
Start time: 4/20/2024 1:05:06 PM; End time: 4/20/2024 1:05:23 PM; Duration: 0:00:17
 - Processing Dimension 'Dim Building' completed.
 - Processing Dimension 'Dim Date' completed.
 - Processing Dimension 'Dim Location' completed.

Status

Deployment Completed Successfully

3.7. Xác định các độ đo (Measures)

- Đổi thuộc tính các độ đo ban đầu và thêm độ đó SIZE từ bảng Dim Building

| Measures | | | | |
|----------|--------------------|---------------|-----------|-------------|
| | Name | Measure Group | Data Type | Aggregation |
| Bar | Rent | Fact | Integer | Sum |
| Bar | Fact Count | Fact | Integer | Count |
| Bar | Size | Dim Building | Integer | Sum |
| | Add new measure... | | | |

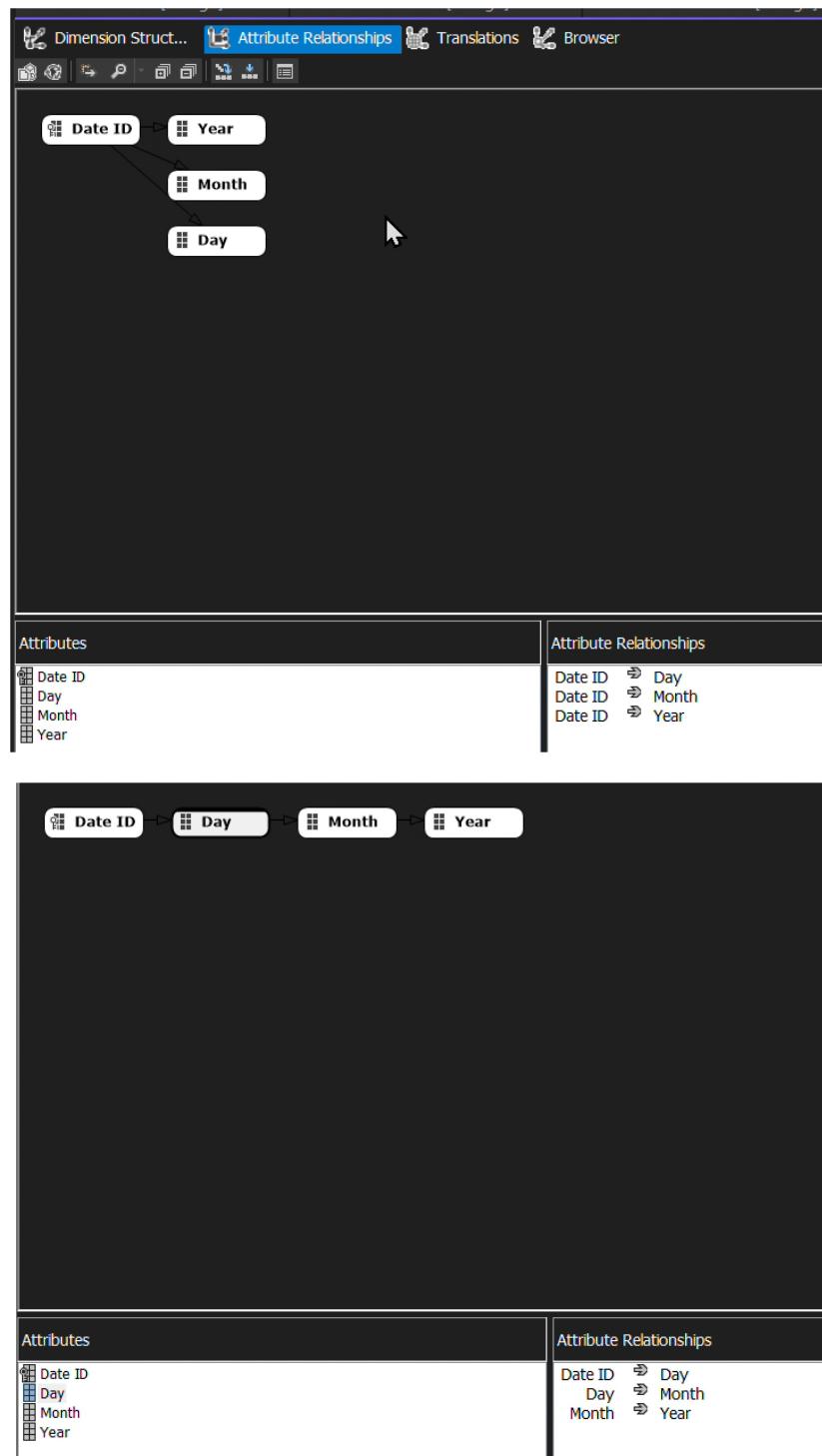
| Measures | | | | |
|----------|--------------------|---------------|-----------|-------------|
| | Name | Measure Group | Data Type | Aggregation |
| Bar | Rent | Fact | Integer | Min |
| Bar | Fact Count | Fact | Integer | Count |
| Bar | Size | Dim Building | Integer | Min |
| | Add new measure... | | | |

3.8. Phân cấp trong các bảng chiều

3.8.1. Phân cấp bảng Dim_Date

The screenshot shows the 'Dimension Structure' tab of the Analysis Services Management Studio. At the top, there are tabs for Dimension Structure, Attribute Relationships, Translations, and Browser. Below the tabs is a toolbar with various icons. On the left, under 'Attributes', the 'Dim Date' dimension is expanded, showing attributes: Date ID, Day, Month, and Year. On the right, under 'Hierarchies', there is a placeholder message: 'To create a new hierarchy, drag an attribute here.'.

This screenshot shows the same interface after a hierarchy has been created. The 'Hierarchies' pane now contains a single hierarchy named 'Hierarchy'. This hierarchy includes levels for Year, Month, Day, and a level labeled '<new level>'. The 'Attributes' pane remains the same, showing the expanded 'Dim Date' dimension with its attributes.



The screenshot shows the Vertipaq Designer application interface. At the top, there is a navigation bar with three tabs: "Day" (selected), "Month", and "Year". Below the tabs is a context menu with four items: "New Attribute Relationship...", "Properties" (highlighted with a cursor), and "New Attribute Relationship..." (repeated twice). The main workspace is dark, and the properties pane on the right is open for the "Day" dimension.

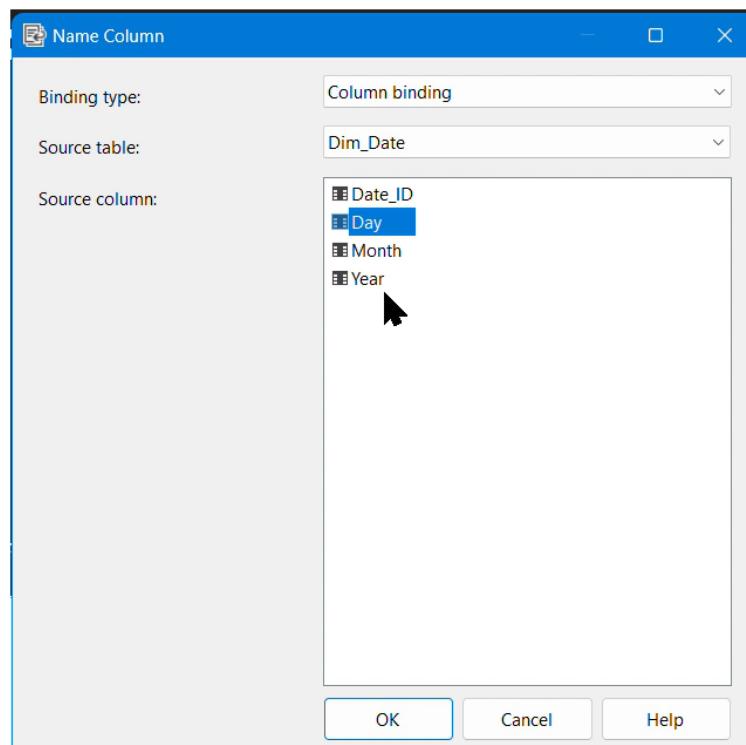
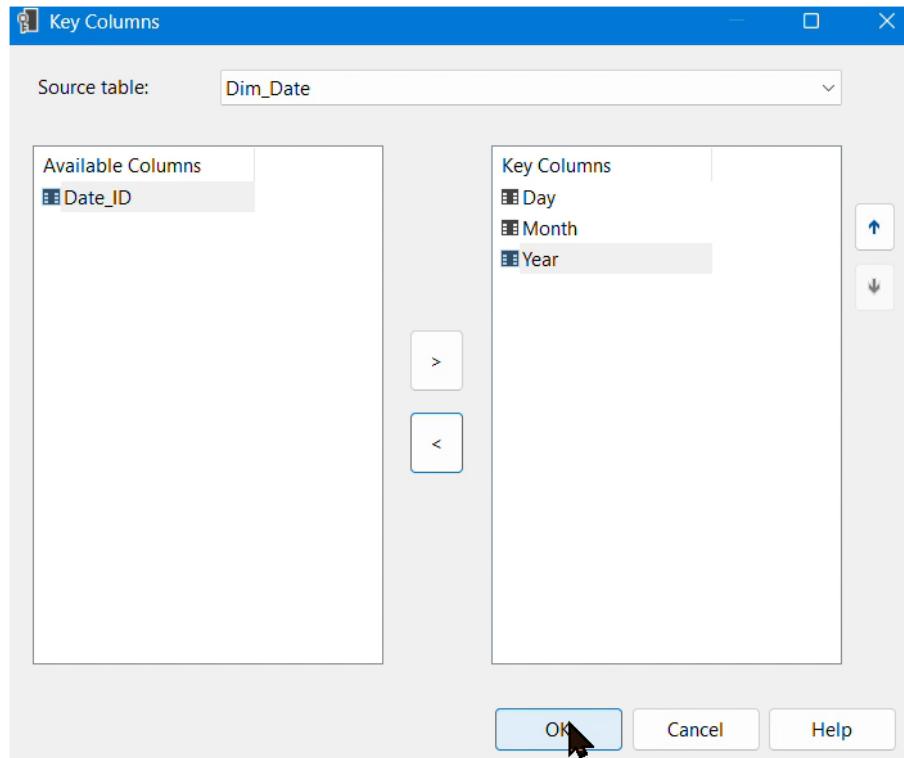
| ProcessingState | Unprocessed |
|-------------------------|----------------------------|
| TokenizationBehavior | TokenizationNone |
| UserEditFlag | 0 |
| VertipaqCompression | VertipaqAutomatic |
| Basic | |
| Description | |
| FormatString | Day |
| ID | Day |
| Name | Day |
| Type | Regular |
| Usage | Regular |
| Misc | |
| AttributeHierarchyOrder | True |
| ExtendedType | |
| GroupingBehavior | EncourageGrouping |
| InstanceSelection | None |
| MemberNameUnique | False |
| VisualizationProperties | |
| Parent-Child | |
| MembersWithData | NonLeafDataVisible |
| MembersWithDataCap | |
| NamingTemplate | |
| RootMemberIf | ParentIsBlankSelfOrMissing |
| UnaryOperatorColumn | (none) |
| Source | |
| CustomRollupColumn | (none) |
| CustomRollupProperties | (none) |
| Key Columns | |
| Dim_Date.Day | (Integer) |
| NameColumn | (none) |
| ValueColumn | (none) |

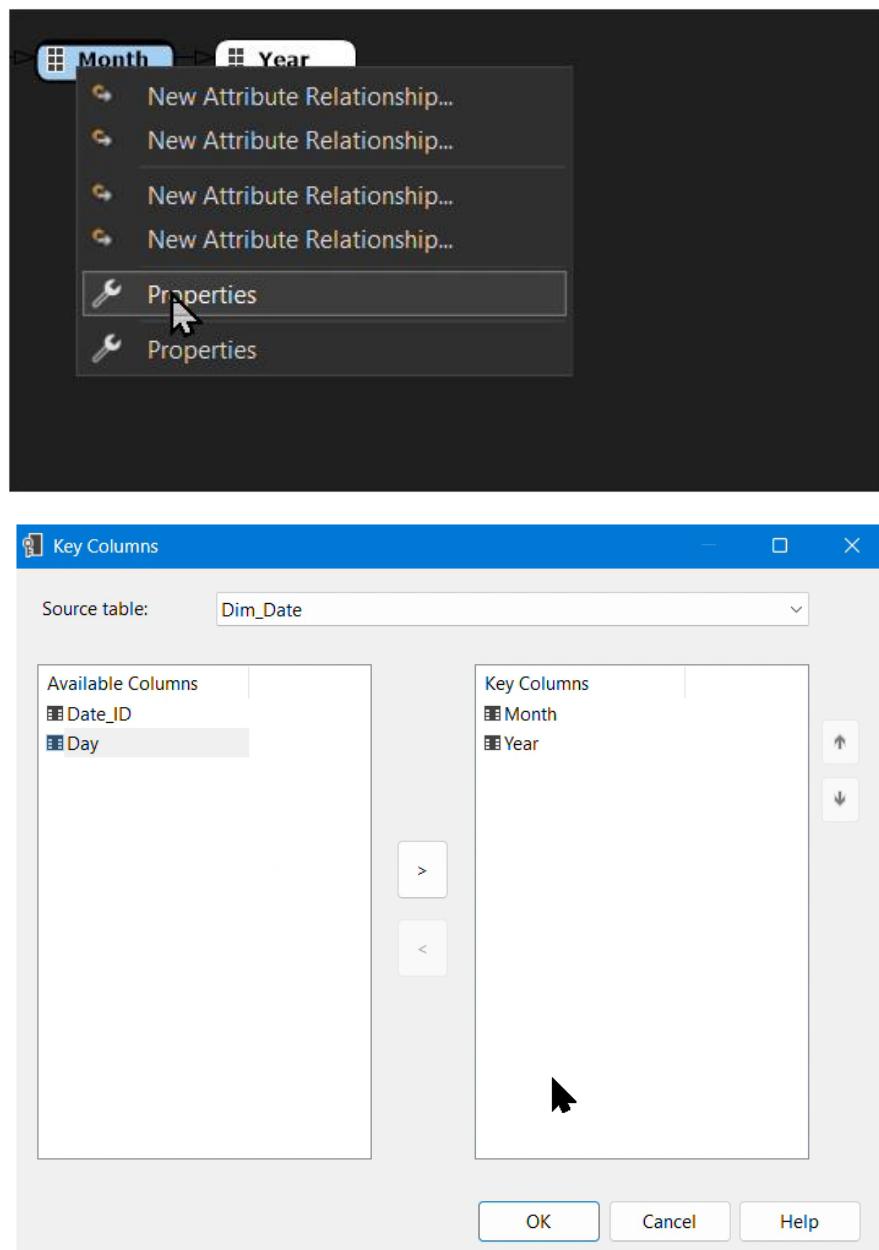
Attributes

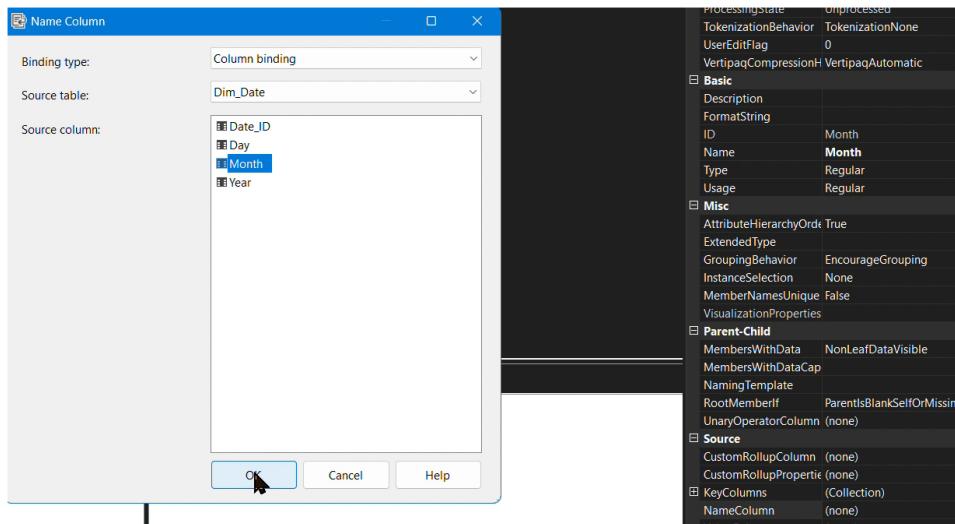
- Date ID
- Day
- Month
- Year

Attribute Relationships

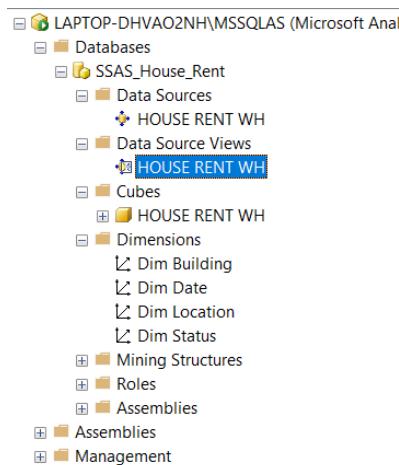
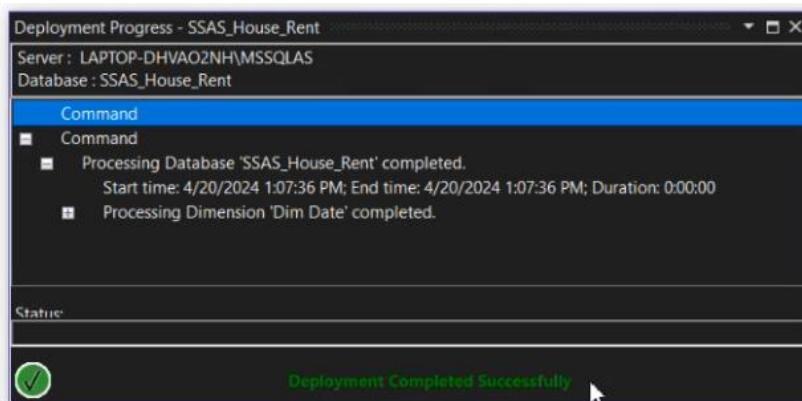
- Date ID → Day
- Day → Month
- Month → Year







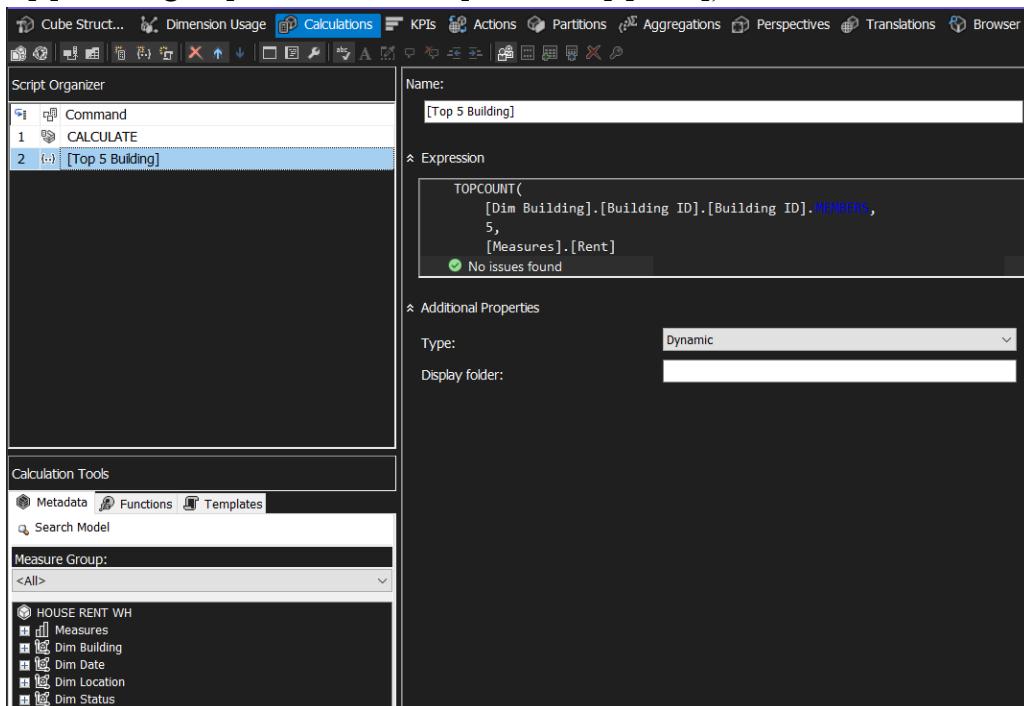
3.9. Chạy dữ án SSAS



3.10. Thực hiện 15 câu truy vấn – Quá trình phân tích dữ liệu bằng thao tác tay trên các khối CUBE

3.10.1. Câu truy vấn 1: Liệt kê 5 tòa nhà cho thuê có giá cao nhất

- Trong Calculations, tạo new name set [Top 5 Building] ở Script Organizer Panel và tạo bộ dữ liệu trong ô Expression như sau: TOPCOUNT([Dim Building].[Building ID].[Building ID].MEMBERS,5, [Measures].[Rent])



- Chọn Dim Building, Operator chọn In và ở Filter Expression chọn name set [Top 5 Building] vừa tạo.

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------|----------|-------------------|
| Dim Building | Building ID | In | Top 5 Building |
| <Select dimension> | | | |
| Building ID | Rent | | |
| 2383 | 3500000 | | |
| 2447 | 1000000 | | |
| 2459 | 700000 | | |
| 2460 | 850000 | | |
| 2508 | 1200000 | | |

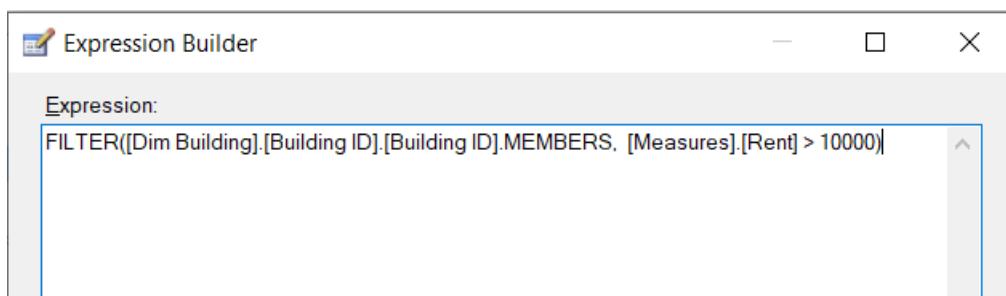
3.10.2. Câu truy vấn 2: Cho biết có bao nhiêu nhà cho thuê có 1 phòng tắm và không có nội thất ở Thành phố “Mumbai”

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------------|----------|-------------------|
| Dim Status | Furnishing Status | Equal | { Unfurnished } |
| Dim Status | Bathroom | Equal | { 1 } |
| Dim Location | City | Equal | { Mumbai } |
| <Select dimension> | | | |

| | |
|------------|----|
| Fact Count | 96 |
|------------|----|

3.10.3. Câu truy vấn 3: Liệt kê các nhà có giá thuê trên 10000, có 1 phòng tắm và không có nội thất

- Trong Operator chọn Custom và Sử dụng Expression sau:



| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------------|----------|---|
| Dim Status | BHK | Equal | { 1 } |
| Dim Status | Furnishing Status | Equal | { Unfurnished } |
| Dim Building | Building ID | Custom | FILTER([Dim Building].[Building ID].[Building ID].MEMBERS, [Measures].[Rent] > 1..) |
| <Select dimension> | | | |

| Building ID | Rent |
|-------------|-------|
| 32 | 14000 |
| 46 | 15000 |
| 56 | 20000 |
| 84 | 11000 |
| 119 | 10500 |
| 132 | 16000 |
| 141 | 13000 |
| 169 | 45000 |
| 170 | 15000 |
| 172 | 19000 |
| 176 | 12000 |
| 177 | 21000 |
| 178 | 18000 |
| 187 | 15500 |
| 188 | 16000 |
| 200 | 20000 |
| 201 | 14000 |
| 202 | 22000 |
| 204 | 15000 |
| 205 | 27000 |
| 208 | 20000 |

3.10.4. Câu truy vấn 4: Cho biết 2 giá thuê cao nhất trong số các chung cư được cho thuê trong tháng 6 và ở tầng 2

- Trong Calculations tạo name set [Top 2 Building]:

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|--------------|--------------|-------------------|
| Dim Date | Month | Equal | { 6 } |
| Dim Building | Floor Number | Equal | { 2 } |
| Dim Building | Building ID | In | Top 2 Building |
| <Select dimension> | | | |
| Building ID | Month | Floor Number | Rent |
| 2460 | 6 | 2 | 850000 |
| 2474 | 6 | 2 | 380000 |

3.10.5. Câu truy vấn 5: Liệt kê giá những căn nhà có diện tích từ “1010 đến 1030” và “3 phòng tắm”

- Trong Calculations tạo name set [Filter Size >= 1010 && <= 1030]:

The screenshot shows the Microsoft Analysis Services Script Organizer interface. In the left pane, under 'Script Organizer' > 'Command', there is a list of items:

- 1 CALCULATE
- 2 (...) [Top 5 Building]
- 3 (...) [Top 2 Building]
- 4 (...) [Filter Size >= 1010 && <= 1030] (highlighted)

In the right pane, the selected item is detailed:

- Name:** [Filter Size >= 1010 && <= 1030]
- Expression:**

```
FILTER(
    [Dim Building].[Building ID].[Building ID].MEMBERS,
    [Dim Building].[Size].CurrentMember.MemberValue >= 1010 AND
    [Dim Building].[Size].CurrentMember.MemberValue <= 1030
)
No issues found
```
- Additional Properties:**
 - Type: Dynamic
 - Display folder: (empty)

Below the script organizer, there is a table titled 'Dimension' showing building data:

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------|----------|--------------------------------|
| Dim Status | Bathroom | Equal | { 3 } |
| Dim Building | Building ID | In | Filter Size >= 1010 && <= 1030 |
| <Select dimension> | | | |

At the bottom, there is a detailed table of building data:

| Building ID | Bathroom | Size | Rent |
|-------------|----------|------|--------|
| 1408 | 3 | 1010 | (null) |
| 1409 | 3 | 1010 | (null) |
| 1410 | 3 | 1012 | (null) |
| 1411 | 3 | 1015 | 10000 |
| 1412 | 3 | 1016 | 45000 |
| 1413 | 3 | 1020 | (null) |
| 1414 | 3 | 1020 | (null) |
| 1415 | 3 | 1020 | (null) |
| 1416 | 3 | 1020 | (null) |
| 1417 | 3 | 1020 | (null) |
| 1418 | 3 | 1020 | (null) |
| 1419 | 3 | 1020 | (null) |
| 1420 | 3 | 1021 | (null) |
| 1421 | 3 | 1021 | 28000 |
| 1422 | 3 | 1025 | (null) |
| 1423 | 3 | 1025 | (null) |
| 1424 | 3 | 1025 | (null) |
| 1425 | 3 | 1027 | (null) |
| 1426 | 3 | 1030 | (null) |
| 1427 | 3 | 1030 | (null) |
| 1428 | 3 | 1030 | (null) |
| 1429 | 3 | 1030 | 40000 |

3.10.6. Câu truy vấn 6: Nhà nào có 1 phòng ngủ, 1 phòng tắm ở thành phố “Bangalore” có giá thấp nhất

- Trong Operator chọn Custom và Sử dụng Expression sau:

```
BOTTOMCOUNT( FILTER(
    [Dim Building].[Building ID].[Building ID].MEMBERS, [Measures].[Rent]
).1, [Measures].[Rent])
```

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------|----------|---|
| Dim Location | City | Equal | { Bangalore } |
| Dim Building | Building ID | Custom | BOTTOMCOUNT(FILTER([Dim Building].[B... |
| Dim Status | BHK | Equal | { 1 } |
| Dim Status | Bathroom | Equal | { 1 } |
| <Select dimension> | | | |

| Building ID | City | Bathroom | BHK | Rent |
|-------------|-----------|----------|-----|------|
| 125 | Bangalore | 1 | 1 | 3500 |

3.10.7. Câu truy vấn 7: Căn nhà nào có giá thuê cao nhất ở thành phố “Mumbai ở tầng trệt”

- Trong Calculations tạo name set [Top 1 Building]:

The screenshot shows the Analysis Services Script Organizer interface. The 'Calculations' tab is selected. In the 'Script Organizer' pane, there is a list of existing calculations, including 'CALCULATE', 'Top 5 Building', 'Top 2 Building', 'Filter Size >= 1010 && <= 1030', and 'Top 1 Building'. The 'Expression' pane shows the formula: `TOPCOUNT([Dim Building].[Building ID].[Building ID].MEMBERS, 2, [Measures].[Rent])`. The 'Additional Properties' pane shows 'Type: Dynamic'.

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------|----------|-------------------|
| Dim Location | City | Equal | { Mumbai } |
| Dim Building | Is Basement | Equal | { True } |
| Dim Building | Building ID | In | Top 1 Building |
| <Select dimension> | | | |

| Building ID | City | Is Basement | Size | Rent |
|-------------|--------|-------------|------|--------|
| 2393 | Mumbai | True | 2600 | 310000 |

3.10.8. Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”

- Trong Operator chọn Custom và Sử dụng Expression sau:

```
TOPCOUNT(
    [Dim Building].[Building ID].[Building ID].MEMBERS * [Dim Location].[City].[Delhi],
    3,
    [Measures].[Rent]
)
```

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------|----------|--|
| Dim Location | City | Equal | { Delhi } |
| Dim Building | Building ID | Custom | TOPCOUNT([Dim Building].[Building ID].[B...] |
| <Select dimension> | | | |

| Building ID | City | SIZE | Rent |
|-------------|-------|------|--------|
| 775 | Delhi | 3200 | 350000 |
| 779 | Delhi | 3800 | 280000 |
| 1460 | Delhi | 4000 | 530000 |

3.10.9. Câu truy vấn 9: Ngày nào số lượng đăng nhà cho thuê ít nhất tháng 7

- Trong Calculations tạo name set [BottomCount FactCount]

The screenshot shows the Microsoft Analysis Services Script Organizer interface. The 'Calculations' tab is selected. A new name set is being created with the following details:

- Name:** [BottomCount FactCount]
- Expression:** BOTTOMCOUNT([Dim Date].[Date ID].[Date ID], 1, [Measures].[Fact Count])
- Type:** Dynamic
- Display folder:** (empty)

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|------------|----------|-----------------------|
| Dim Date | Hierarchy | Equal | { 7 } |
| Dim Date | Date ID | In | BottomCount FactCount |
| <Select dimension> | | | |
| Date ID | Fact Count | | |
| 2022-07-11 | 1 | | |

3.10.10. Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”

Expression:
**TOPCOUNT([Dim Building].[Building ID].[Building ID].MEMBERS
 .10, [Measures].[Rent])**

| Dimension | Hierarchy | Operator | Filter Expression |
|--------------------|-------------------|----------|---|
| Dim Location | City | Equal | { Chennai } |
| Dim Status | Furnishing Status | Equal | { Furnished } |
| Dim Building | Building ID | Custom | TOPCOUNT([Dim Building].[Building ID].[Buildi...] |
| <Select dimension> | | | |
| Building ID | City | Rent | |
| 1916 | Chennai | 110000 | |
| 1986 | Chennai | 70000 | |
| 2146 | Chennai | 85000 | |
| 2209 | Chennai | 60000 | |
| 2342 | Chennai | 100000 | |
| 2368 | Chennai | 150000 | |
| 2403 | Chennai | 160000 | |
| 2416 | Chennai | 130000 | |
| 2439 | Chennai | 200000 | |
| 2444 | Chennai | 130000 | |

3.12.11. Câu truy vấn 11: Lấy ra 5 nhà có nội thất với giá cao nhất trong mỗi tháng

```

SELECT [Measures].[Rent] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Date].[Month].Children,
    TOPCOUNT(
        [Dim Date].[Month].CurrentMember * [Dim Building].[Building ID].Children,
        5,
        [Measures].[Rent]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Status].[Furnishing Status].&[Furnished];

```

| Month | Building ID | Rent |
|-------|-------------|--------|
| 4 | 2189 | 260000 |
| 4 | 2148 | 100000 |
| 4 | 1276 | 50000 |
| 4 | 1548 | 40000 |
| 4 | 724 | 33000 |
| 5 | 2501 | 600000 |
| 5 | 2328 | 400000 |
| 5 | 2332 | 300000 |
| 5 | 2421 | 280000 |
| 5 | 1355 | 230000 |
| 6 | 2460 | 850000 |
| 6 | 2459 | 700000 |
| 6 | 2385 | 400000 |

3.12.12. Câu truy vấn 12: Lấy 5 nhà có giá lớn nhất có Area Locality ở Behala theo từng tháng

```

SELECT [Measures].[Rent] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Date].[Month].Children,
    TOPCOUNT(
        [Dim Date].[Month].CurrentMember * [Dim Building].[Building ID].Children,
        5,
        [Measures].[Rent]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Location].[Area Locality].&[Behala];

```

| Month | Building ID | Rent |
|-------|-------------|-------|
| 5 | 2297 | 30000 |
| 5 | 1362 | 12000 |
| 5 | 677 | 9000 |
| 5 | 856 | 8500 |
| 5 | 659 | 7000 |
| 6 | 1039 | 35000 |
| 6 | 417 | 10000 |
| 6 | 1038 | 10000 |
| 6 | 788 | 6000 |
| 6 | 768 | 5500 |

3.12.13. Câu truy vấn 13: Lấy ra những nhà có 4 phòng tắm và có Area Locality bắt đầu bằng chữ “M”

```

SELECT [Measures].[Rent] ON COLUMNS,
FILTER(
    ORDER(
        [Dim Location].[Area Locality].[Area Locality].MEMBERS ,
        [Measures].[Rent],
        ASC
    ),
    LEFT([Dim Location].[Area Locality].CurrentMember.Name, 1) = "M"
    AND [Dim Status].[Bathroom].&[4]
) ON ROWS
FROM [HOUSE RENT WH];

```

| Area Locality | Rent |
|---------------------------------------|--------|
| Manikonda, Outer Ring Road | 9000 |
| Mylapore | 9000 |
| Madhapur | 10500 |
| Mehdipatnam | 11000 |
| Mallikarjuna Nagar, Secunderabad | 20000 |
| Manigunda | 20000 |
| Model Town | 39000 |
| Magnum Tower CHS, Lokhandwala Complex | 50000 |
| MLA Colony, Banjara Hills | 120000 |
| Madras Boat Club Road | 200000 |
| Mount Marry, Bandra West | 600000 |

3.12.14. Câu truy vấn 14: Mô hình thành phố tìm ra 3 nhà có giá cao nhất có nội thất cơ bản

```

SELECT [Measures].[Rent] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Location].[City].children,
    TOPCOUNT(
        [Dim Location].[City].CurrentMember * [Dim Building].[Building ID].children,
        3,
        [Measures].[Rent]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Status].[Furnishing Status].&[Semi-Furnished];

```

| City | Building ID | Rent |
|-----------|-------------|--------|
| Bangalore | 2383 | 350... |
| Bangalore | 2474 | 380... |
| Bangalore | 2509 | 250... |
| Chennai | 2479 | 330... |
| Chennai | 2502 | 280... |
| Chennai | 2473 | 250... |
| Delhi | 2492 | 530... |
| Delhi | 2454 | 350... |
| Delhi | 2484 | 280... |
| Hyderabad | 2511 | 400... |
| Hyderabad | 2503 | 250... |
| Hyderabad | 2506 | 200... |
| Kolkata | 2126 | 600... |
| Kolkata | 2471 | 400... |
| Kolkata | 1039 | 350... |
| Mumbai | 2508 | 120... |
| Mumbai | 2447 | 100... |
| Mumbai | 2264 | 680... |

3.12.15. Câu truy vấn 15: Với mỗi thành phố lấy ra 5 nhà có diện tích lớn nhất có Area Type là “Super Area”

```

SELECT [Measures].[Size] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Location].[City].children,
    TOPCOUNT(
        [Dim Location].[City].CurrentMember * [Dim Building].[Building ID].children,
        5,
        [Measures].[Size]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Location].[Area Type].&[Super Area];

```

| City | Building ID | Size |
|-----------|-------------|------|
| Bangalore | 2512 | 8000 |
| Bangalore | 2511 | 7000 |
| Bangalore | 2510 | 6000 |
| Bangalore | 2509 | 5700 |
| Bangalore | 2508 | 5000 |
| Chennai | 2512 | 8000 |
| Chennai | 2511 | 7000 |
| Chennai | 2510 | 6000 |
| Chennai | 2509 | 5700 |
| Chennai | 2508 | 5000 |
| Delhi | 2512 | 8000 |
| Delhi | 2511 | 7000 |
| Delhi | 2510 | 6000 |

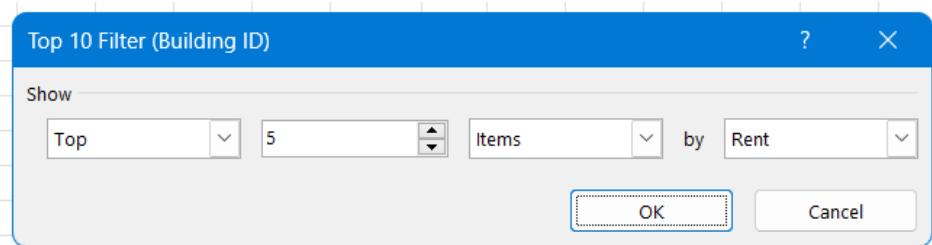
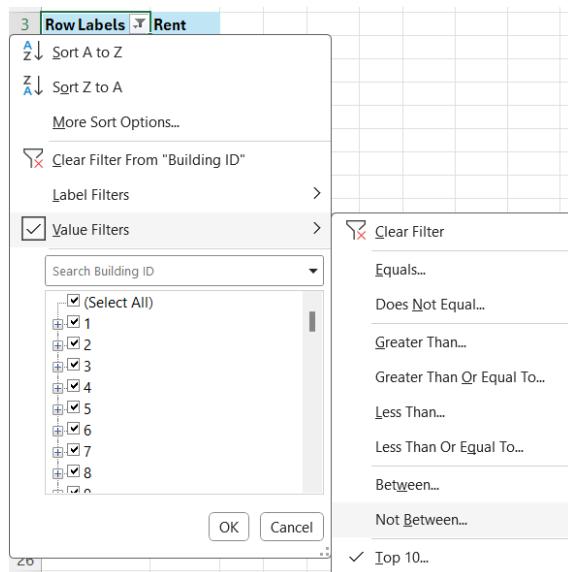
3.11. Thực hiện 10 câu truy vấn – Quá trình phân tích dữ liệu bằng Pivot Excel

3.11.1. Câu truy vấn 1: Liệt kê 5 tòa nhà cho thuê với giá trên 300000

- Kéo thả Building ID vào Rows và Rent vào Values

The screenshot shows the Power BI Data Model view. At the top, under the Fact table, 'Rent' is selected. Below it, under the Dim Building table, 'Building ID' is selected. In the 'Rows' section, 'Building ID' is listed. In the 'Values' section, 'Rent' is listed.

- Tiếp theo chọn Value Filters, chọn Top 10.



- Kết quả:

| Row Labels | Rent |
|------------|---------|
| 1167 | 850000 |
| 1658 | 3500000 |
| 1664 | 1200000 |
| 2394 | 1000000 |
| 2447 | 700000 |

3.11.2. Câu truy vấn 2: Cho biết có bao nhiêu nhà cho thuê có 1 phòng tắm và không có nội thất ở Thành phố “Mumbai”

- Kéo thả thuộc tính Bathroom và Furnishing Status vào Filters, City vào Rows, Fact Count vào Values.
- Chọn lọc số phòng ngủ là 1, tình trạng chưa có nội thất và thành phố là Mumbai.

The screenshot shows a data analysis interface with the following configuration:

- Filters:** Contains two checked filters: "Bathroom" and "Furnishing Status".
- Rows:** Contains one selected filter: "City".
- Values:** Contains one selected measure: "Fact Count".
- Dimensions:** On the left, under "Dim Status", "Bathroom" and "Furnishing Status" are checked. Under "Location ID", "City" is selected.

- Kết quả:

| Bathroom | 1 |
|-------------------|-------------|
| Furnishing Status | Unfurnished |
| Row Labels | Fact Count |
| Mumbai | 96 |

3.11.3. Câu truy vấn 3: Liệt kê các nhà có giá thuê trên 10000, có 1 phòng tắm và không có nội thất

- Kéo thả thuộc tính BHK và Furnishing Status vào Filters, Building ID vào Rows và SIZE vào Values

The screenshot shows a data visualization interface with the following configuration:

- Filters:** Contains 'BHK' and 'Furnishing Status'.
- Rows:** Contains 'Building ID'.
- Values:** Contains 'SIZE'.
- Dim Status:** Contains 'Bathroom' and 'BHK' (both checked).

- Chọn lọc số phòng ngủ là 1, tình trạng chưa có nội thất

The screenshot shows a filtered result with the following data:

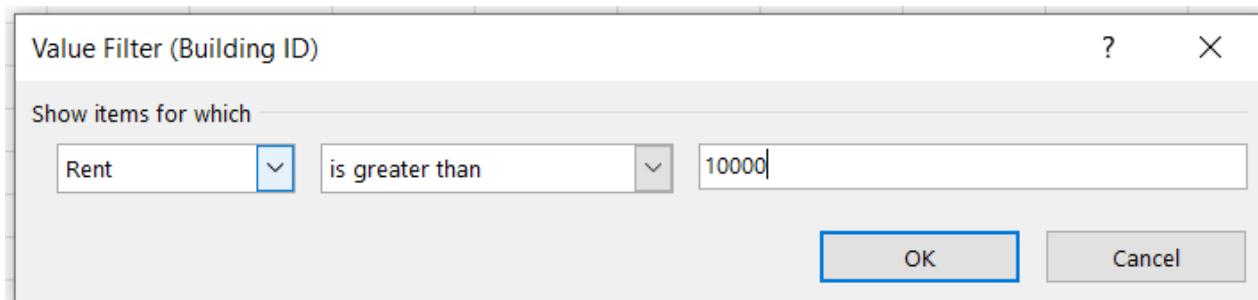
| | |
|-------------------|-------------|
| BHK | 1 |
| Furnishing Status | Unfurnished |

- Chọn Value Filters – Greater Than

The screenshot shows the 'Value Filters' menu with the following configuration:

- Value Filters:** Contains 'Search Building ID' with options '(Select All)', '1', '2', '3', '4', and '5'.
- Greater Than...** is selected.

- Chọn giá trị cho SIZE lớn hơn 600



- Kết quả:

| Row Labels | Rent |
|------------|-------|
| 32 | 14000 |
| 46 | 15000 |
| 56 | 20000 |
| 84 | 11000 |
| 119 | 10500 |
| 132 | 16000 |
| 141 | 13000 |
| 169 | 45000 |
| 170 | 15000 |
| 172 | 19000 |
| 176 | 12000 |
| 177 | 21000 |
| 178 | 18000 |
| 187 | 15500 |
| 188 | 16000 |

3.11.4. Câu truy vấn 4: Cho biết 2 giá thuê cao nhất trong số các chung cư được cho thuê trong tháng 6 và ở tầng 2

- Kéo thả Hierachy và Floor Number vào Filters, Building ID vào Rows và Rent vào Values

The screenshot shows the Power BI Data view. In the top pane, there is a tree structure with the following items:

- $\sum \text{Dim Building}$
 - SIZE
- $\sum \text{Fact}$
 - Fact Count
 - Rent
- $\sum \text{Dim Building}$

Below the tree structure, there is a message: "Drag fields between areas below:". The bottom pane is divided into two sections: "Filters" and "Columns".

Filters:

- Hierarchy (selected value: Month 6)
- Floor Number (selected value: 2)

Columns:

- Values (selected value: Rent)

- Chọn lọc Hierarchy Month là tháng 6, số tầng là 2

- Kết quả:

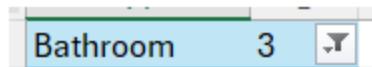
| Rent |
|--------|
| 850000 |
| 380000 |

3.11.5. Câu truy vấn 5: Liệt kê giá những căn nhà có diện tích từ “1010 đến 1030” và “3 phòng tắm”

- Kéo thả thuộc tính Bathroom vào Filters, Building ID vào Rows và Rent vào Values

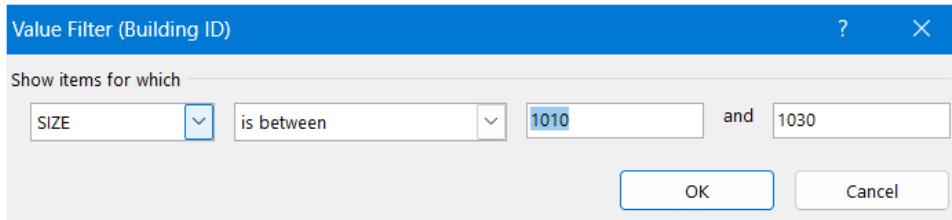
The screenshot shows a data visualization interface. At the top, there's a legend with a green square labeled 'Rent'. Below it, under 'Dim Building', 'Building ID' is checked. In the 'Filters' section, 'Bathroom' is set to 3. The 'Rows' section shows 'Building ID' and the 'Values' section shows 'Rent'.

- Chọn lọc Bathroom là 3



- Chọn Value Filters – Between

The screenshot shows the 'Value Filters' dialog for 'Building ID'. It lists building IDs 1 through 8, each with a checkbox. On the right, a sidebar shows filter options: 'Clear Filter', 'Equals...', 'Does Not Equal...', 'Greater Than...', 'Greater Than Or Equal To...', 'Less Than...', 'Less Than Or Equal To...', and 'Between...'. The 'Between...' option is currently selected.



- Kết quả:

| Row Labels | Rent |
|------------|-------|
| 238 | 10000 |
| 1336 | 45000 |
| 2127 | 28000 |
| 2128 | 40000 |

3.11.6. Câu truy vấn 6: Nhà nào có 1 phòng ngủ, 1 phòng tắm ở thành phố “Bangalore” có giá thấp nhất

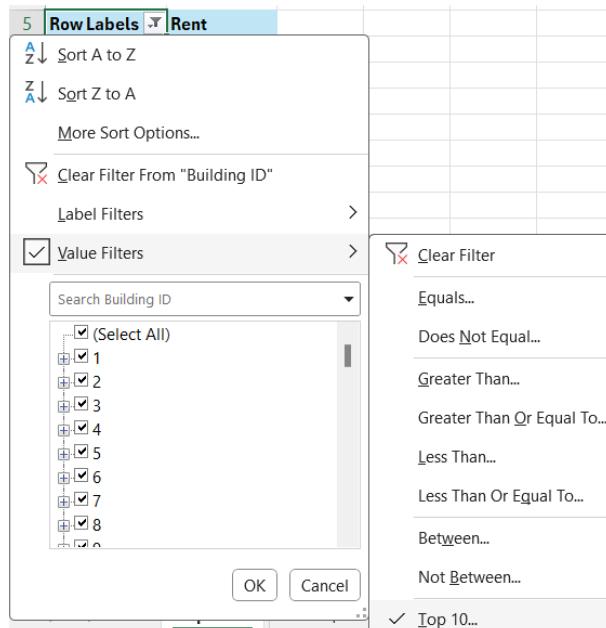
- Kéo thả BHK, Bathroom và City vào Filters, Building ID vào Rows và Rent vào Values

The screenshot shows the Power BI Filter pane and Fields pane. In the Filter pane, 'City' and 'Bathroom' are checked. In the Fields pane, 'BHK', 'Bathroom', and 'City' are listed under 'Filters'. 'Building ID' is listed under 'Rows'. Under 'Values', 'Rent' is listed.

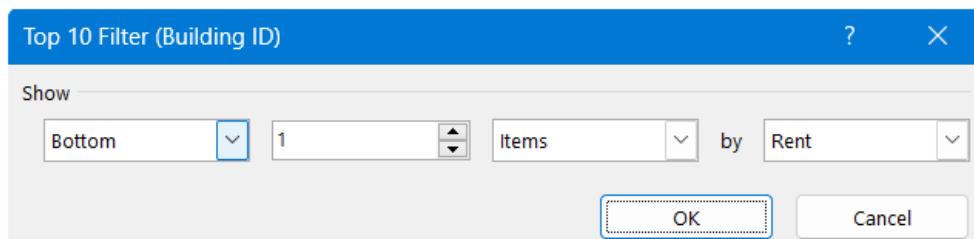
- Chọn lọc 1 phòng ngủ, 1 phòng tắm, ở thành phố Bangalore

| | | |
|----------|-----------|--|
| BHK | 1 | |
| Bathroom | 1 | |
| City | Bangalore | |

- Chọn Value Filters – Top 10



- Chọn Show Bottom



- Kết quả:

| Row Labels | Rent |
|------------|------|
| 814 | 3500 |

3.11.7. Câu truy vấn 7: Căn nhà nào có giá thuê cao nhất ở thành phố “Mumbai ở tầng trệt”

- Kéo thả City và Is Basement vào Filters, Rent vào Values, Building ID vào Rows.

The screenshot shows the Power BI Data View interface. At the top, there is a tree view of the 'Dim Building' table with several columns listed:

- Building Floors (unchecked)
- Building ID** (checked)
- Floor Number (unchecked)
- Is Basement** (checked)
- Size (unchecked)

Below the table, there is a section titled "Drag fields between areas below:" which contains four main sections:

- Filters**: Contains dropdowns for "City" and "Is Basement".
- Columns**: An empty list.
- Rows**: Contains a dropdown for "Building ID".
- Values**: Contains a dropdown for "Rent".

At the bottom of the interface are two buttons: "Defer Layout Update" and "Update".

- Chọn thành phố “Mumbai” ở City và True ở Is Basement

| | |
|-------------|--------|
| City | Mumbai |
| Is Basement | True |

The screenshot shows the 'Row Labels' context menu open over a data table. The 'Value Filters' option is selected. A sub-menu for 'Building ID' is displayed, showing checkboxes for values 1 through 8. To the right of the main menu is a 'Clear Filter' dialog with various comparison operators. Below these is a 'Top 10 Filter (Building ID)' dialog, which has 'Top' set to 1, 'by' set to 'Rent', and 'Items' set to '1'. The 'OK' button is highlighted.

- Kết quả:

The screenshot shows the Power BI interface with the results of the top 10 filter applied. Row 31 is selected, and its value '310000' is displayed under the 'Rent' column.

3.11.8. Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”

- Kéo thả City vào Filters, Rent và SIZE vào Values, Building ID vào Rows.

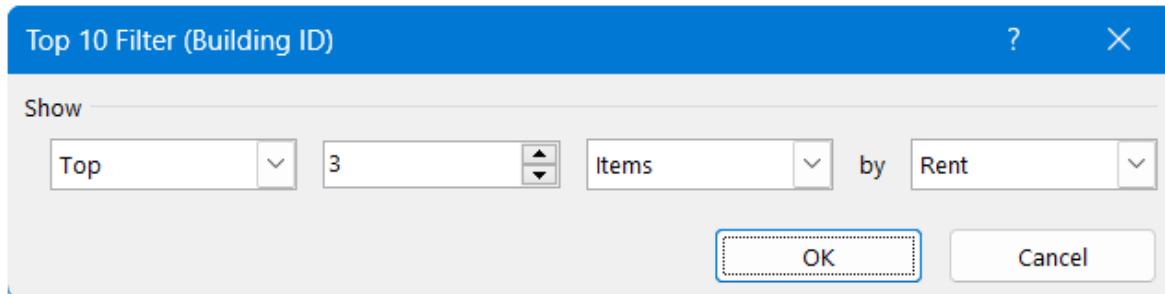
The screenshot shows a dimension hierarchy pane at the top with 'Dim Location' expanded, listing 'Area Locality', 'Area Type', 'City' (selected), and 'Location ID'. Below this is a 'Dim Status' node. A message 'Drag fields between areas below:' is displayed. Underneath, there are two main sections: 'Filters' and 'Columns'. In the 'Filters' section, 'City' is selected. In the 'Columns' section, the aggregation 'Σ Values' is selected. Below these are 'Rows' and 'Values' sections. In the 'Rows' section, 'Building ID' is selected. In the 'Values' section, 'SIZE' and 'Rent' are listed.

- Chọn lọc thành phố “Delhi”



- Chọn Top 10 trong Value Filters

The screenshot shows a 'Value Filters' dialog box. At the top, it displays 'Row Labels' and 'Rent'. Below this are sorting options: 'Sort A to Z' and 'Sort Z to A'. There is also a 'More Sort Options...' link. A 'Clear Filter From "Building ID"' button is present. The 'Value Filters' section is checked and expanded, showing a search bar 'Search Building ID' and a list of building IDs from 1 to 8, all of which are selected ('Select All'). To the right of the dialog, a list of filter conditions is shown, with 'Top 10...' selected.



- Kết quả:

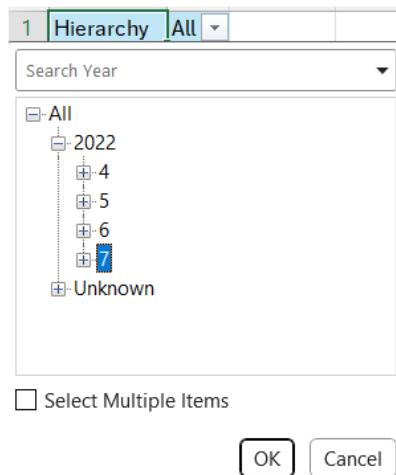
| Row Labels | SIZE | Rent |
|------------|------|--------|
| 775 | 3200 | 350000 |
| 779 | 3800 | 280000 |
| 1460 | 4000 | 530000 |

3.11.9. Câu truy vấn 9: Ngày nào số lượng đăng nhà cho thuê ít nhất tháng 7

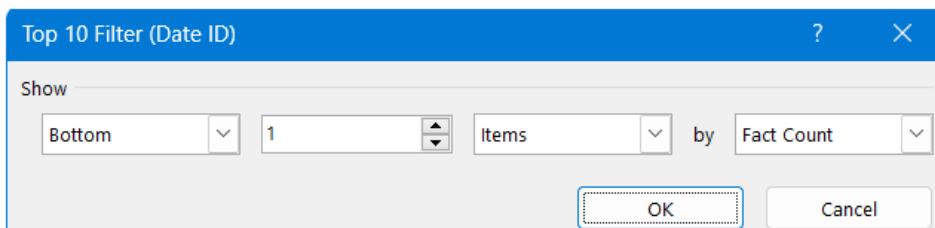
- Kéo thả Hierarchy vào Filters, Fact Count vào Values và Date ID vào Rows

The screenshot shows the Power BI interface. In the top left, the Fields pane displays "Month", "Day", and a collapsed "More Fields" section containing "Date ID" (which is checked). In the bottom right, the ribbon is divided into three sections: "Rows" (set to "Date ID"), "Filters" (set to "Hierarchy"), and "Values" (set to "Fact Count"). A message "Drag fields between areas below:" is visible between the Fields pane and the ribbon.

- Lọc Hierarchy chọn lấy tháng 7



- Chọn Top 10 Filter, Show Bottom và chọn trường Fact Count



- Kết quả:

| Row Labels | Fact Count |
|------------|------------|
| 2022-07-11 | 1 |

3.11.10. Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”

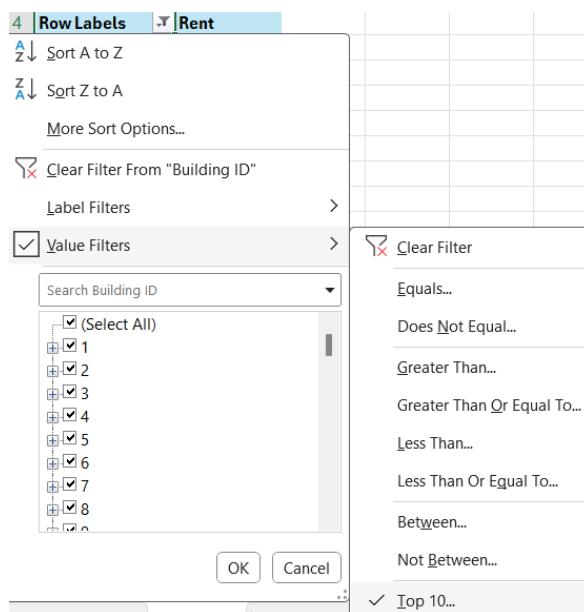
- Kéo thả City và Furnishing Status vào Filters, Rent vào Values và Building ID vào Rows

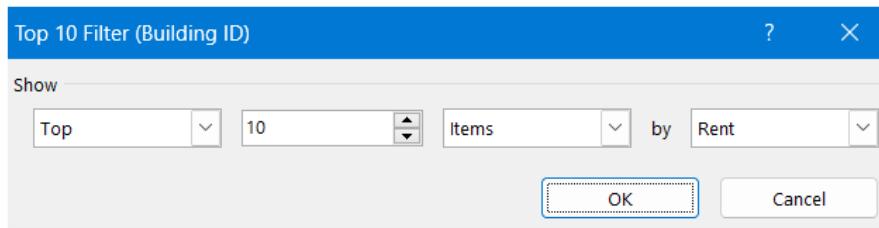
The screenshot shows the Power BI Data Model view. In the top pane, under the 'Dim Building' table, the 'Building ID' column is selected. Below this, the Power BI Query Editor interface is visible, featuring a 'Filters' section with dropdowns for 'City' and 'Furnishing Status', and a 'Columns' section with 'Rows' set to 'Building ID' and 'Values' set to 'Rent'.

- Lọc thành phố Chennai, có nội thất

A screenshot of a Power BI visualization showing a matrix. The columns are labeled 'City' (Chennai) and 'Furnishing Status' (Furnished). The rows are implicitly defined by the data points in the matrix cells.

- Chọn Top 10 Filter, lọc theo Giá





- Kết quả:

| Row Labels | Rent |
|------------|--------|
| 335 | 100000 |
| 702 | 70000 |
| 735 | 60000 |
| 770 | 130000 |
| 772 | 200000 |
| 1149 | 150000 |
| 1774 | 110000 |
| 1802 | 130000 |
| 2024 | 160000 |
| 2143 | 85000 |

3.12. Thực hiện 15 câu truy vấn – Quá trình phân tích dữ liệu bằng ngôn ngữ truy vấn MDX

3.12.1. Câu truy vấn 1: Liệt kê 5 tòa nhà cho thuê có giá cao nhất

MDX Query:

```

SELECT {[Measures].[Rent]} ON COLUMNS,
TOPCOUNT([Dim Building].[Building ID].[Building ID].MEMBERS, 5, [Measures].[Rent]) ON ROWS
FROM [HOUSE RENT WH]
  
```

Results:

| | Rent |
|------|---------|
| 2383 | 3500000 |
| 2508 | 1200000 |
| 2447 | 1000000 |
| 2460 | 850000 |
| 2459 | 700000 |

3.12.2. Câu truy vấn 2: Cho biết có bao nhiêu nhà cho thuê có 1 phòng tắm và không có nội thất ở Thành phố “Mumbai”

The screenshot shows the SSMS interface with the following details:

- Cube:** HOUSE RENT WH
- Measure Group:** <All>
- Query:**

```

SELECT {[Measures].[Fact Count]} ON COLUMNS
FROM (
    SELECT [Dim Location].[City].&[Mumbai] AS [Selected City]
    ON COLUMNS
    FROM [HOUSE RENT WH]
)
WHERE (
    [Dim Status].[Bathroom].[1],
    [Dim Status].[Furnishing Status].[Unfurnished]
)

```
- Results:** Fact Count = 96

3.12.3. Câu truy vấn 3: Liệt kê các nhà có giá thuê trên 10000, có 1 phòng tắm và không có nội thất

The screenshot shows the SSMS interface with the following details:

- Query:**

```

SELECT [Measures].[Rent] ON COLUMNS,
FILTER(
    [Dim Building].[Building ID].[Building ID],
    [Measures].[Rent] > 10000
) ON ROWS
FROM [HOUSE RENT WH]
WHERE ([Dim Status].[Bathroom].&[1], [Dim Status].[Furnishing Status].&[Unfurnished]);

```
- Results:**

| Rent | |
|------|-------|
| 32 | 14000 |
| 41 | 11000 |
| 44 | 12000 |
| 46 | 15000 |
| 49 | 13000 |
| 80 | 14000 |
| 84 | 11000 |
| 119 | 10500 |
| 132 | 16000 |
| 141 | 13000 |
| 164 | 21467 |
| 169 | 45000 |
| 170 | 15000 |

3.12.4. Câu truy vấn 4: Cho biết 2 giá thuê cao nhất trong số các chung cư được cho thuê trong tháng 6 và ở tầng 2

```

SELECT [Measures].[Rent] ON COLUMNS,
TOPCOUNT(
    [Dim Building].[Building ID].[Building ID].MEMBERS, 2,
    [Dim Building].[Floor Number].[2] *
    [Dim Date].[Month].[6] *
    [Measures].[Rent]
) ON ROWS
FROM [HOUSE RENT WH];

```

| Rent | 2460 | 850000 |
|------|------|--------|
| 2474 | | 380000 |

3.12.5. Câu truy vấn 5: Liệt kê giá những căn nhà có diện tích từ “1010 đến 1030” và “3 phòng tắm”

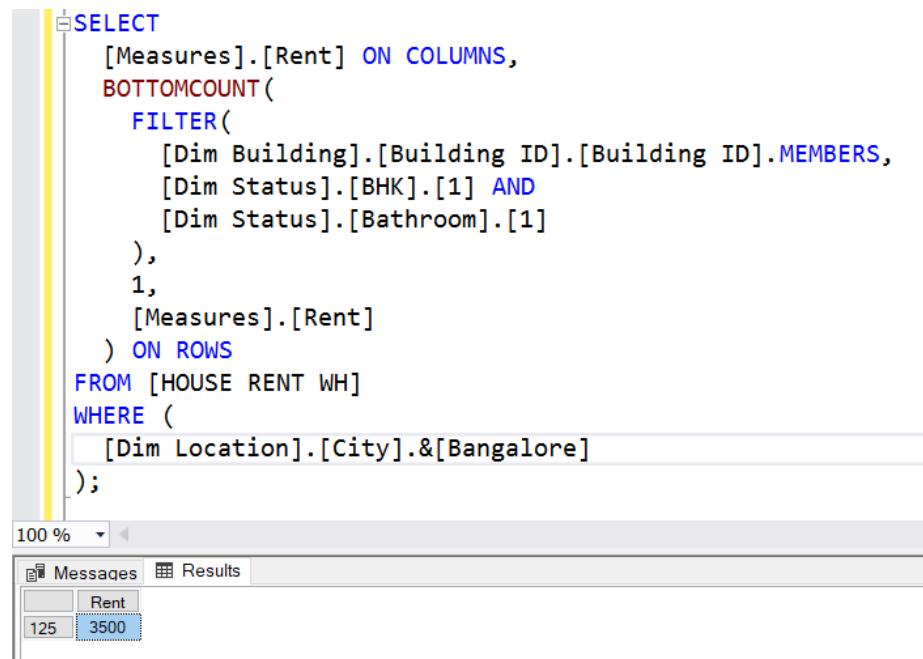
```

SELECT {[Measures].[Rent], [Measures].[Size]} ON COLUMNS,
FILTER(
    [Dim Building].[Building ID].[Building ID].MEMBERS,
    [Dim Building].[Size].CurrentMember.MemberValue >= 1010 AND
    [Dim Building].[Size].CurrentMember.MemberValue <= 1030 AND
    [Dim Status].[Bathroom].&[3]
) ON ROWS
FROM [HOUSE RENT WH];

```

| Rent | Size | |
|------|-------|------|
| 1411 | 10000 | 1015 |
| 1412 | 45000 | 1016 |
| 1421 | 28000 | 1021 |
| 1429 | 40000 | 1030 |

3.12.6. Câu truy vấn 6: Nhà nào có 1 phòng ngủ, 1 phòng tắm ở thành phố “Bangalore” có giá thấp nhất



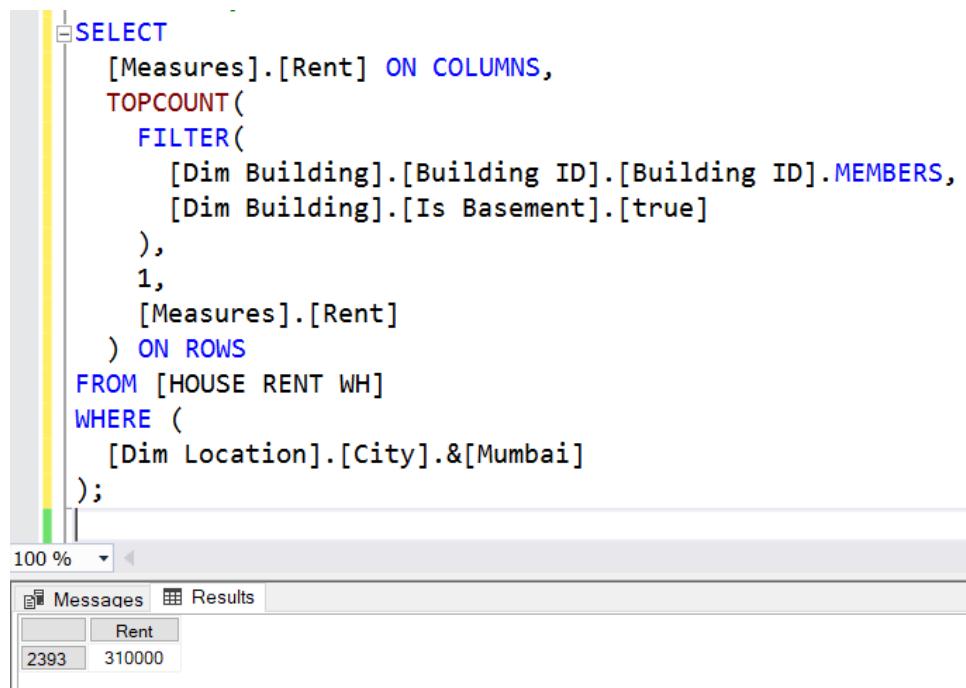
```

SELECT
    [Measures].[Rent] ON COLUMNS,
    BOTTOMCOUNT(
        FILTER(
            [Dim Building].[Building ID].[Building ID].MEMBERS,
            [Dim Status].[BHK].[1] AND
            [Dim Status].[Bathroom].[1]
        ),
        1,
        [Measures].[Rent]
    ) ON ROWS
FROM [HOUSE RENT WH]
WHERE (
    [Dim Location].[City].&[Bangalore]
);
  
```

The screenshot shows the SQL query for question 6. The results pane displays a single row with the value 125 under the Rent column.

| Rent |
|------|
| 125 |

3.12.7. Câu truy vấn 7: Căn nhà nào có giá thuê cao nhất ở thành phố “Mumbai ở tầng trệt”



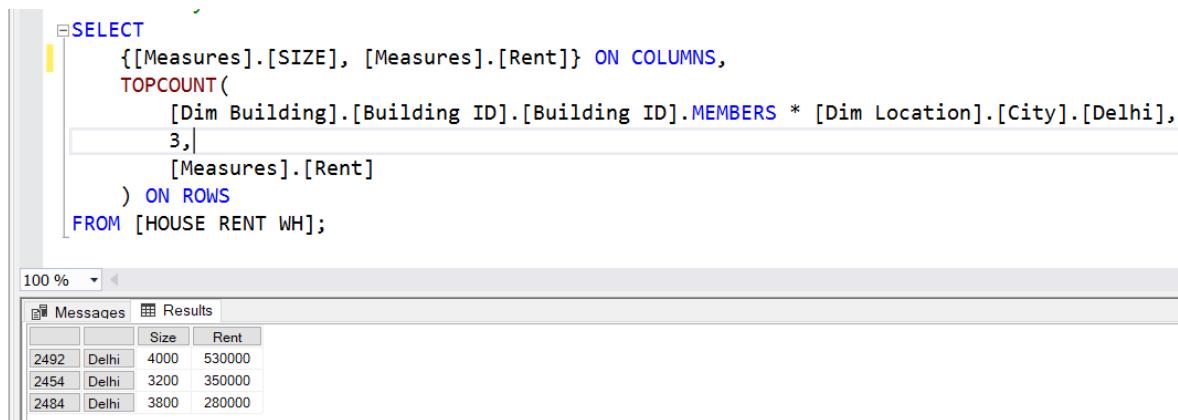
```

SELECT
    [Measures].[Rent] ON COLUMNS,
    TOPCOUNT(
        FILTER(
            [Dim Building].[Building ID].[Building ID].MEMBERS,
            [Dim Building].[Is Basement].[true]
        ),
        1,
        [Measures].[Rent]
    ) ON ROWS
FROM [HOUSE RENT WH]
WHERE (
    [Dim Location].[City].&[Mumbai]
);
  
```

The screenshot shows the SQL query for question 7. The results pane displays a single row with the value 310000 under the Rent column.

| Rent | |
|------|--------|
| 2393 | 310000 |

3.12.8. Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”



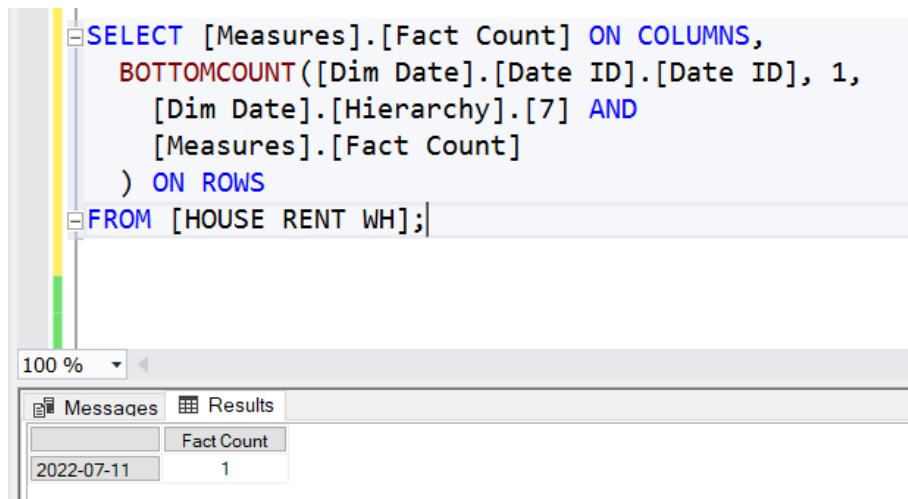
```

SELECT {[Measures].[SIZE], [Measures].[Rent]} ON COLUMNS,
TOPCOUNT(
    [Dim Building].[Building ID].[Building ID].MEMBERS * [Dim Location].[City].[Delhi],
    3,
    [Measures].[Rent]
) ON ROWS
FROM [HOUSE RENT WH];
  
```

The screenshot shows the SQL query above and its results. The results table has columns 'Size' and 'Rent'. It lists three rows for 'Delhi':

| | Size | Rent |
|------|------|--------|
| 2492 | 4000 | 530000 |
| 2454 | 3200 | 350000 |
| 2484 | 3800 | 280000 |

3.12.9. Câu truy vấn 9: Ngày nào số lượng đăng nhà cho thuê ít nhất tháng 7



```

SELECT [Measures].[Fact Count] ON COLUMNS,
BOTTOMCOUNT([Dim Date].[Date ID].[Date ID], 1,
[Dim Date].[Hierarchy].[7] AND
[Measures].[Fact Count]
) ON ROWS
FROM [HOUSE RENT WH];
  
```

The screenshot shows the SQL query above and its results. The results table has columns 'Fact Count'. It lists one row for '2022-07-11' with a value of 1.

| | Fact Count |
|------------|------------|
| 2022-07-11 | 1 |

3.12.10. Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”

```

SELECT [Measures].[Rent] ON COLUMNS,
TOPCOUNT(
    FILTER(
        [Dim Building].[Building ID].[Building ID].MEMBERS * [Dim Location].[City].&[Chennai],
        [Dim Status].[Furnishing Status].&[Furnished]
    ),
    10,
    [Measures].[Rent]
) ON ROWS
FROM [HOUSE RENT WH];

```

100 %

| | | Rent |
|------|---------|--------|
| 2403 | Chennai | 160000 |
| 2368 | Chennai | 150000 |
| 2416 | Chennai | 130000 |
| 2444 | Chennai | 130000 |
| 1916 | Chennai | 110000 |
| 2342 | Chennai | 100000 |
| 2439 | Chennai | 100000 |
| 2146 | Chennai | 85000 |
| 1986 | Chennai | 70000 |
| 2209 | Chennai | 60000 |

3.12.11. Câu truy vấn 11: Lấy ra 5 nhà có nội thất với giá cao nhất trong mỗi tháng

```

SELECT [Measures].[Rent] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Date].[Month].Children,
    TOPCOUNT(
        [Dim Date].[Month].CurrentMember * [Dim Building].[Building ID].Children,
        5,
        [Measures].[Rent]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Status].[Furnishing Status].&[Furnished];

```

| | | Rent |
|---|------|--------|
| 4 | 2189 | 260000 |
| 4 | 2148 | 100000 |
| 4 | 1276 | 50000 |
| 4 | 1548 | 40000 |
| 4 | 724 | 33000 |
| 5 | 2501 | 600000 |
| 5 | 2328 | 400000 |
| 5 | 2332 | 300000 |
| 5 | 2421 | 280000 |
| 5 | 1355 | 230000 |
| 6 | 2460 | 850000 |
| 6 | 2459 | 700000 |
| 6 | 2385 | 400000 |
| 6 | 2229 | 360000 |
| 6 | 2187 | 300000 |
| 7 | 2488 | 500000 |

3.12.12. Câu truy vấn 12: Lấy 5 nhà có giá thuê lớn nhất có Area Locality ở Behala theo từng tháng

```

SELECT [Measures].[Rent] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Date].[Month].Children,
    TOPCOUNT(
        [Dim Date].[Month].CurrentMember * [Dim Building].[Building ID].Children,
        5,
        [Measures].[Rent]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Location].[Area Locality].&[Behala];

```

| | | Rent |
|---|------|-------|
| 5 | 2297 | 30000 |
| 5 | 1362 | 12000 |
| 5 | 677 | 9000 |
| 5 | 856 | 8500 |
| 5 | 659 | 7000 |
| 6 | 1039 | 35000 |
| 6 | 417 | 10000 |
| 6 | 1038 | 10000 |
| 6 | 788 | 6000 |
| 6 | 768 | 5500 |

3.12.13. Câu truy vấn 13: Lấy ra những nhà có 4 phòng tắm và có Area Locality bắt đầu bằng chữ “M” theo thứ tự tăng dần giá thuê

```

SELECT [Measures].[Rent] ON COLUMNS,
FILTER(
    ORDER(
        [Dim Location].[Area Locality].[Area Locality].MEMBERS ,
        [Measures].[Rent],
        ASC
    ),
    LEFT([Dim Location].[Area Locality].CurrentMember.Name, 1) = "M"
    AND [Dim Status].[Bathroom].&[4]
) ON ROWS
FROM [HOUSE RENT WH];

```

100 % ▶

| | Rent |
|---------------------------------------|--------|
| Manikonda, Outer Ring Road | 9000 |
| Mylapore | 9000 |
| Madhapur | 10500 |
| Mehdipatnam | 11000 |
| Malikarjuna Nagar, Secunderabad | 20000 |
| Manigunda | 20000 |
| Model Town | 39000 |
| Magnum Tower CHS, Lokhandwala Complex | 50000 |
| MLA Colony, Banjara Hills | 120000 |
| Madras Boat Club Road | 200000 |
| Mount Mary, Bandra West | 600000 |

3.12.14. Câu truy vấn 14: Mô hình thành phố tìm ra 3 nhà có giá cao nhất có nội thất cơ bản

```

SELECT [Measures].[Rent] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Location].[City].children,
    TOPCOUNT(
        [Dim Location].[City].CurrentMember * [Dim Building].[Building ID].children,
        3,
        [Measures].[Rent]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Status].[Furnishing Status].&[Semi-Furnished];

```

| City | Building ID | Rent |
|-----------|-------------|--------|
| Bangalore | 2383 | 350... |
| Bangalore | 2474 | 380... |
| Bangalore | 2509 | 250... |
| Chennai | 2479 | 330... |
| Chennai | 2502 | 280... |
| Chennai | 2473 | 250... |
| Delhi | 2492 | 530... |
| Delhi | 2454 | 350... |
| Delhi | 2484 | 280... |
| Hyderabad | 2511 | 400... |
| Hyderabad | 2503 | 250... |
| Hyderabad | 2506 | 200... |
| Kolkata | 2126 | 600... |
| Kolkata | 2471 | 400... |
| Kolkata | 1039 | 350... |
| Mumbai | 2508 | 120... |
| Mumbai | 2447 | 100... |
| Mumbai | 2264 | 680... |

3.12.15. Câu truy vấn 15: Với mô hình thành phố lấy ra 5 nhà có diện tích lớn nhất có Area Type là “Super Area”

```
SELECT [Measures].[Size] ON COLUMNS,
NON EMPTY GENERATE(
    [Dim Location].[City].children,
    TOPCOUNT(
        [Dim Location].[City].CurrentMember * [Dim Building].[Building ID].children,
        5,
        [Measures].[Size]
    )
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Location].[Area Type].&[Super Area];
```

100 % ▶

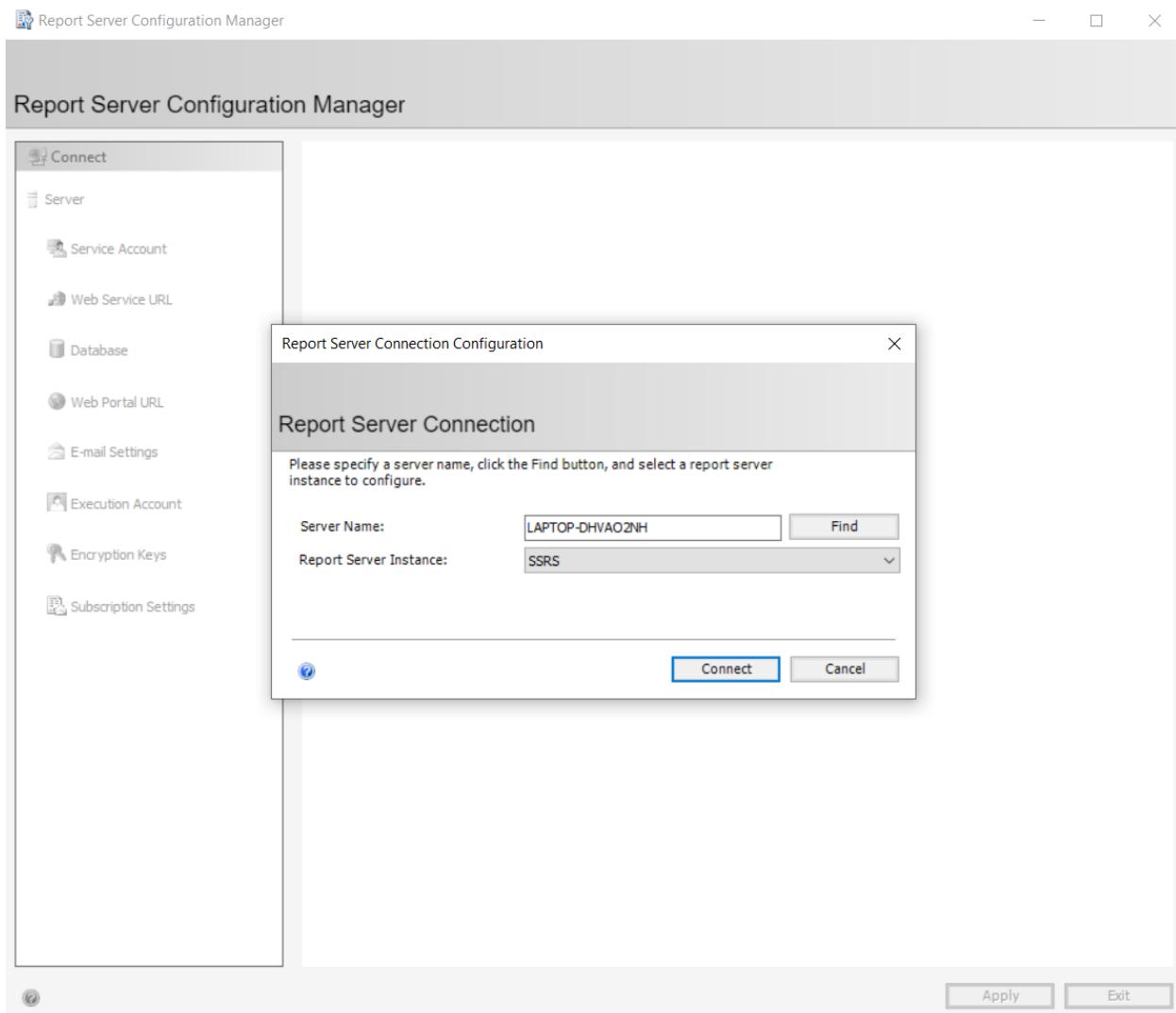
| | | Size |
|-----------|------|------|
| Bangalore | 2512 | 8000 |
| Bangalore | 2511 | 7000 |
| Bangalore | 2510 | 6000 |
| Bangalore | 2509 | 5700 |
| Bangalore | 2508 | 5000 |
| Chennai | 2512 | 8000 |
| Chennai | 2511 | 7000 |
| Chennai | 2510 | 6000 |
| Chennai | 2509 | 5700 |
| Chennai | 2508 | 5000 |
| Delhi | 2512 | 8000 |
| Delhi | 2511 | 7000 |
| Delhi | 2510 | 6000 |
| Delhi | 2509 | 5700 |
| Delhi | 2508 | 5000 |
| Hyderabad | 2512 | 8000 |

CHƯƠNG 4. SSRS

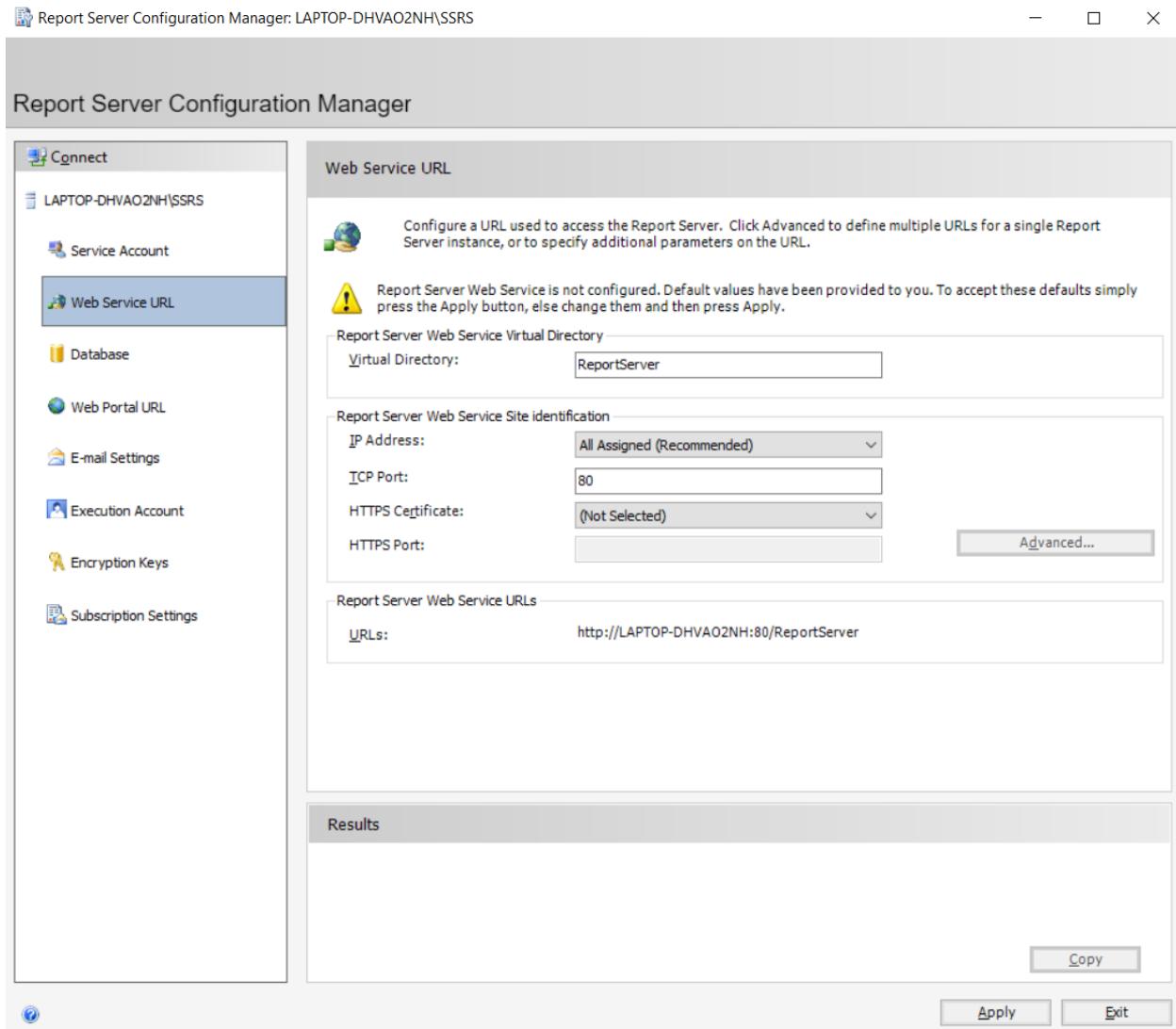
4.1. Report với Visual Studio

4.1.1. Cấu hình Report Server Configuration Manager

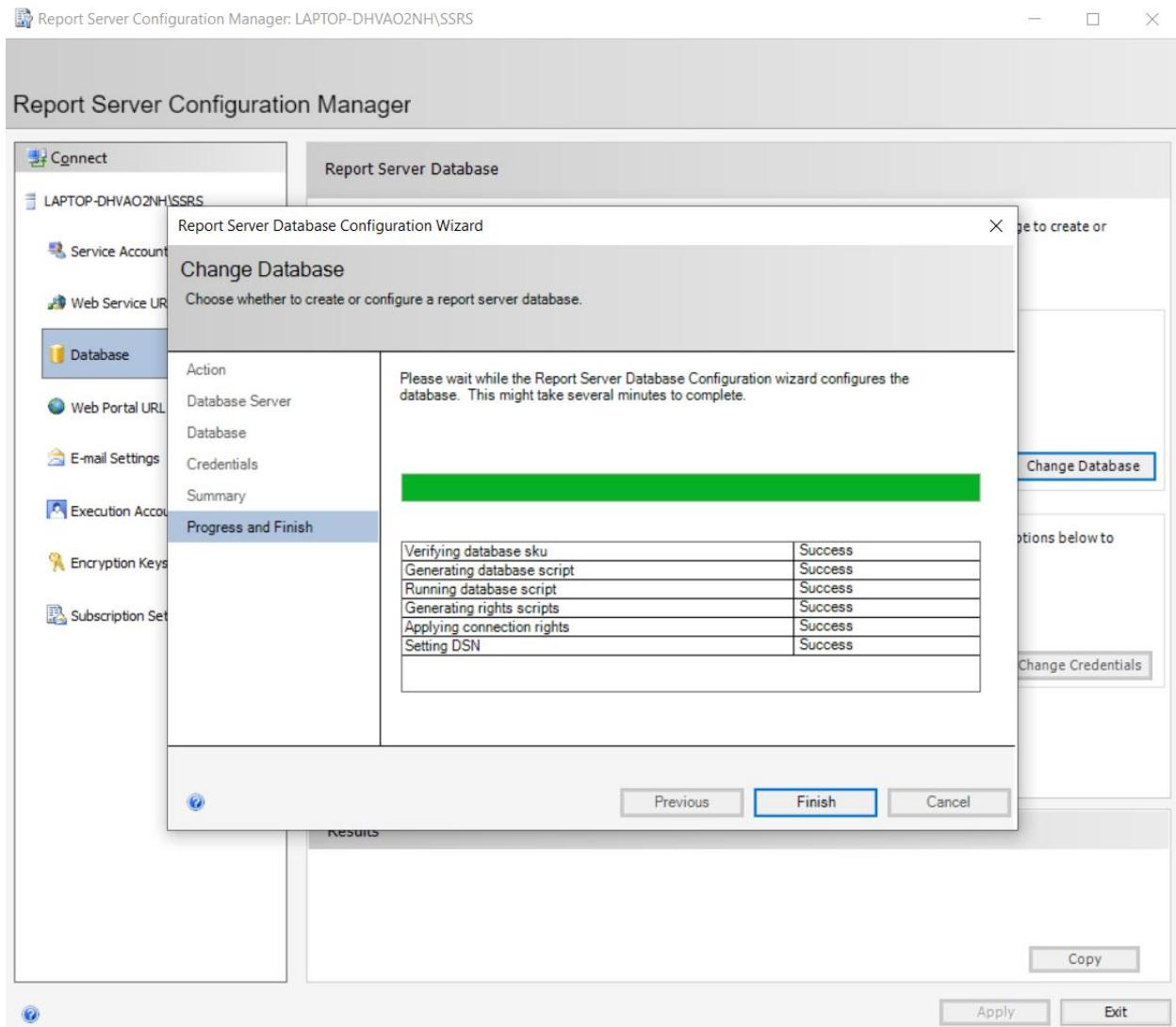
Tải về máy SQLReportingServices.exe. Chạy file .exe và cài đặt theo chỉ dẫn. Sau khi cài đặt xong, mở Report Server Configuration Manager và nhấp Connect đến Server

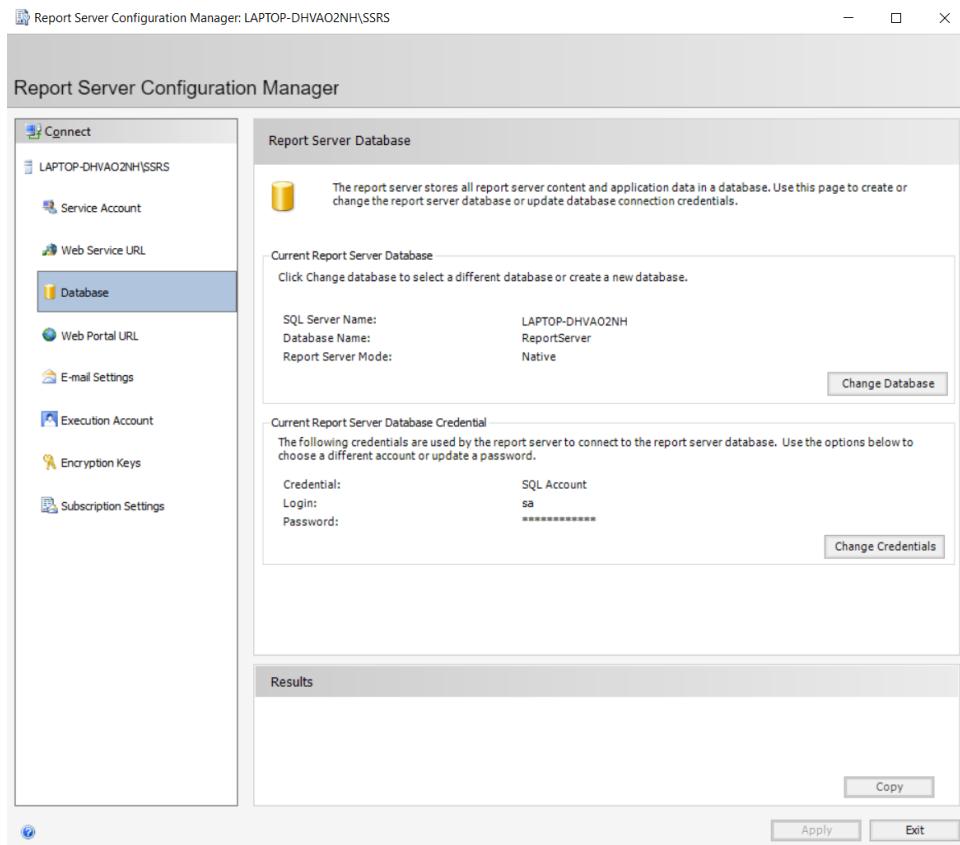


Trong phần Web Service URL, giữ nguyên các mặc định và nhấn Apply.

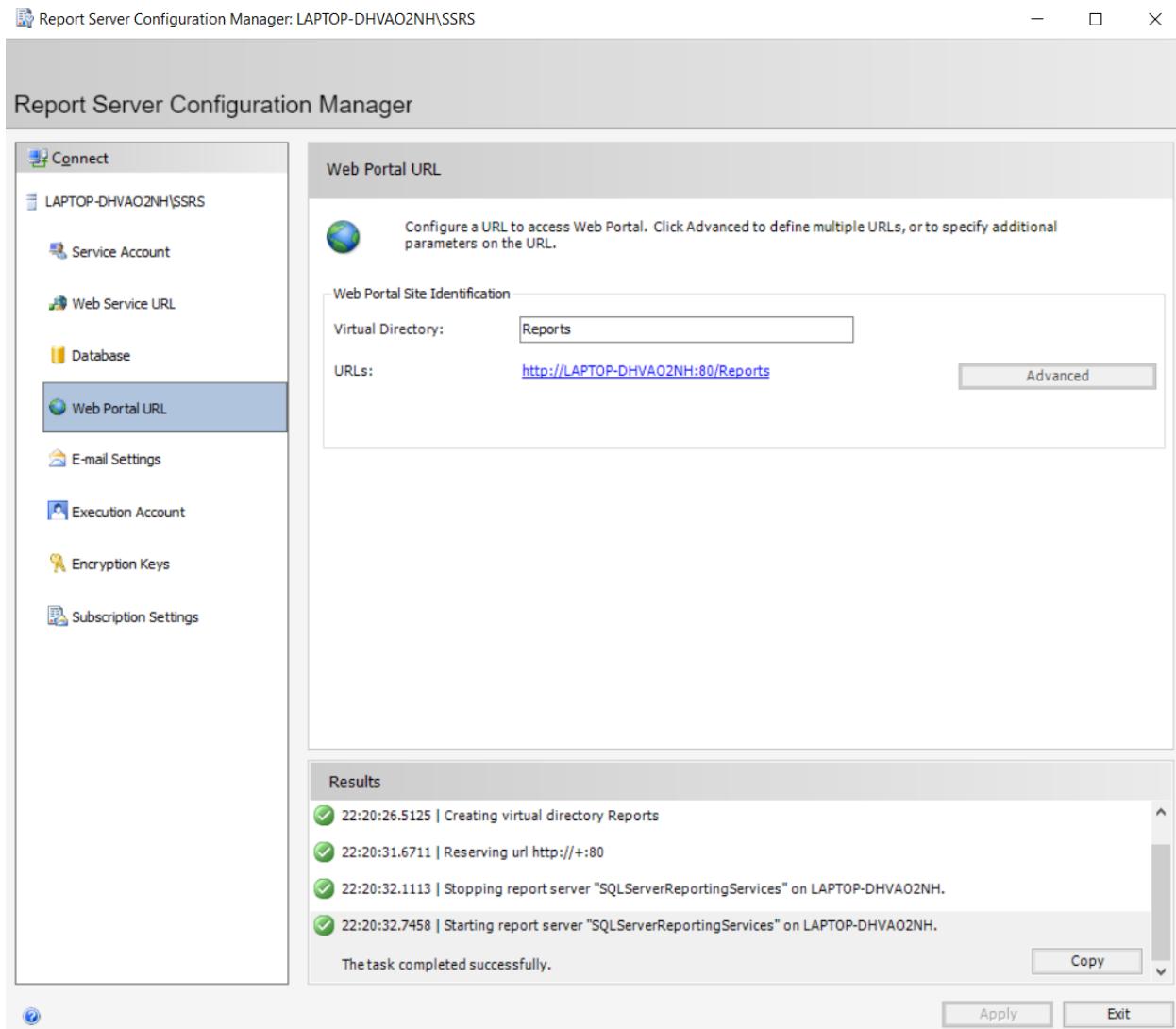


Chọn vào phần Database và nhấn Change Database để thay đổi cơ sở dữ liệu. Tiến hành thay đổi Database theo chỉ dẫn.



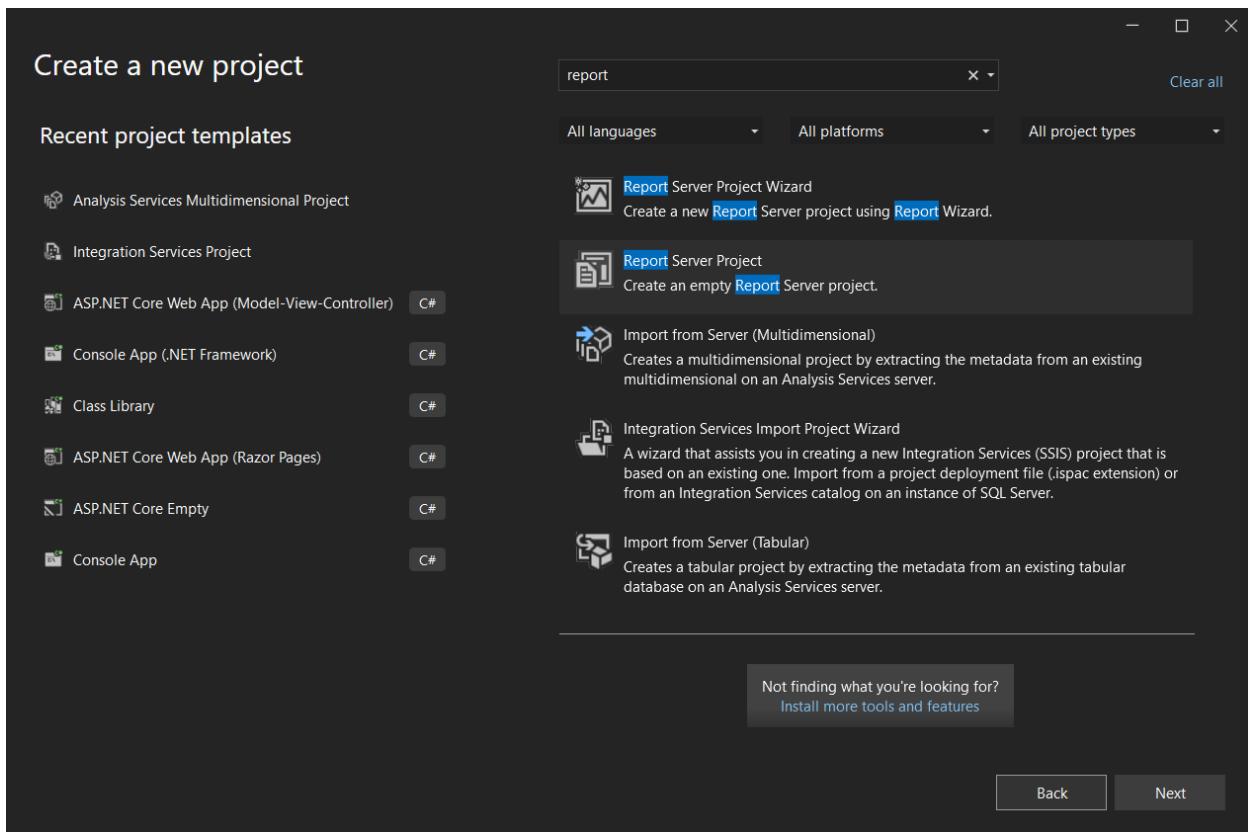


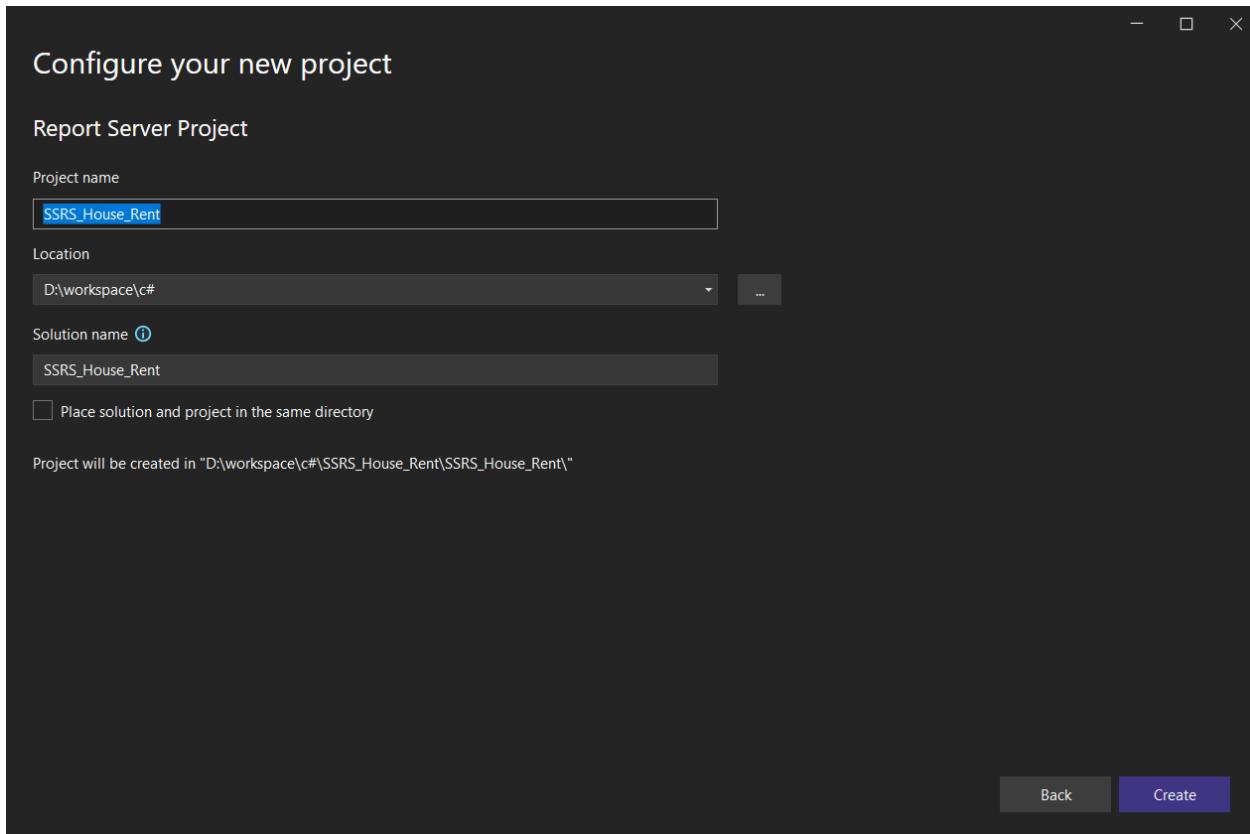
Trong phần Web Portal URL, kiểm tra các giá trị có trong hình dưới. Sau đó nhấn Apply để xác nhận cấu hình.



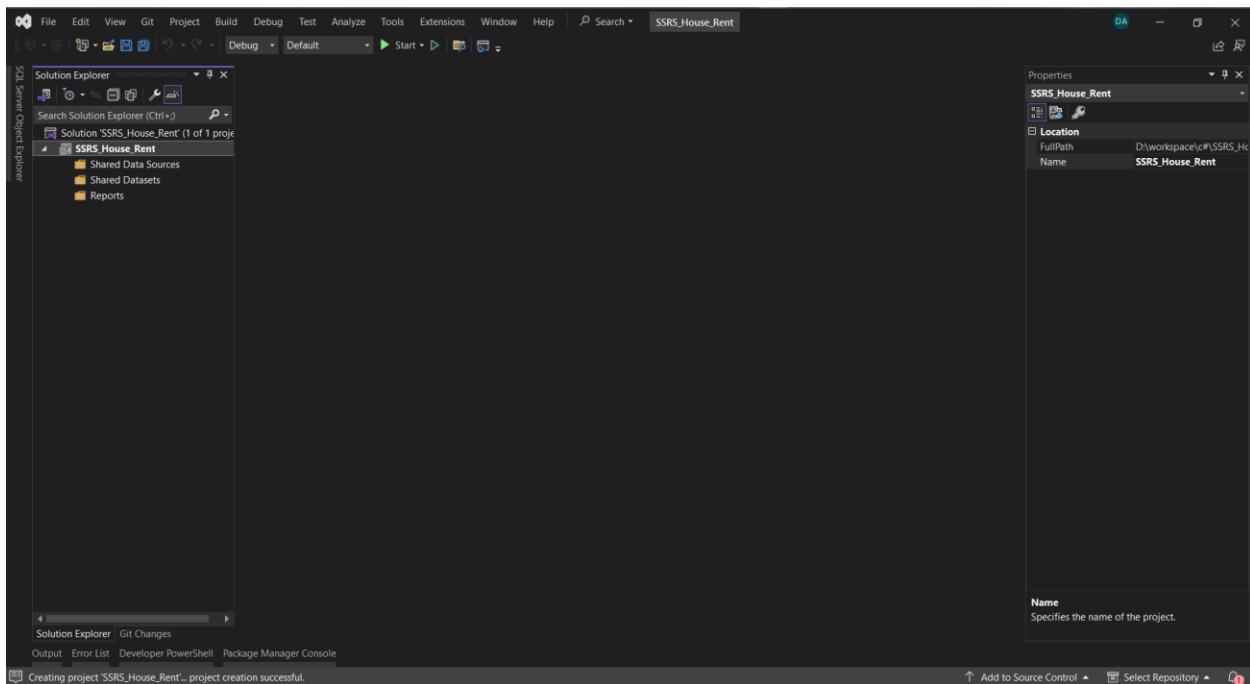
4.1.2. Tạo project SSRS trên Visual Studio

Mở Visual Studio, chọn Create a new project -> tìm và chọn Report Server Project -> Create project.

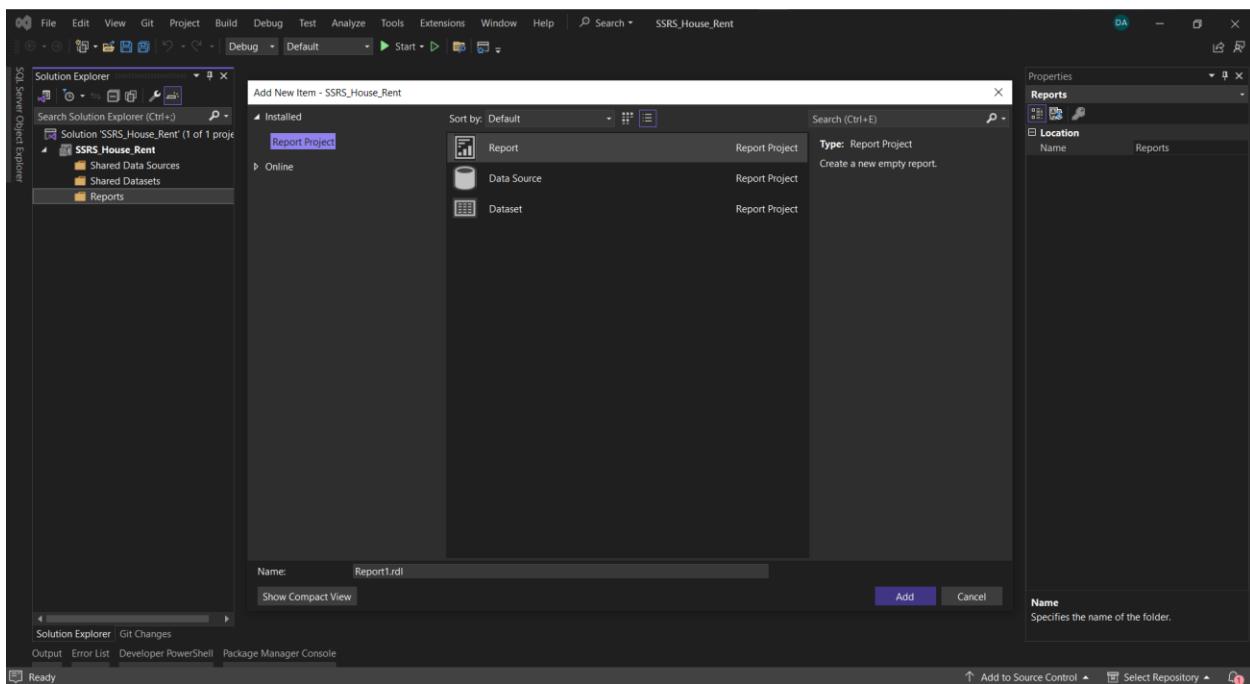
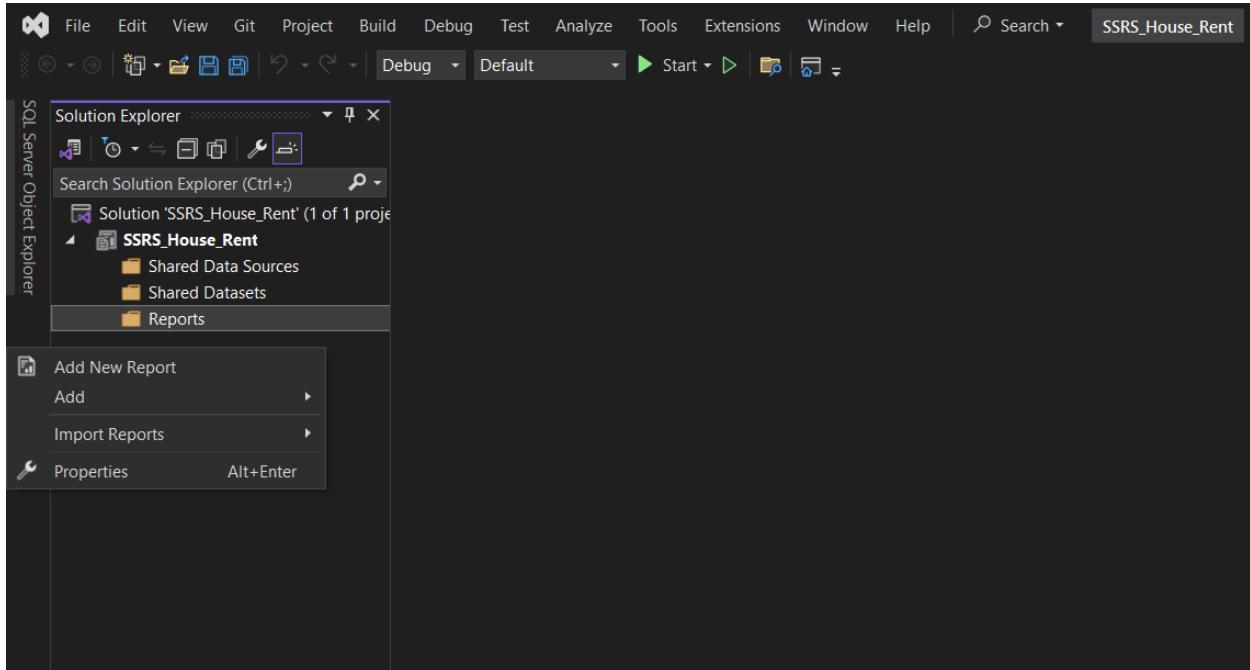




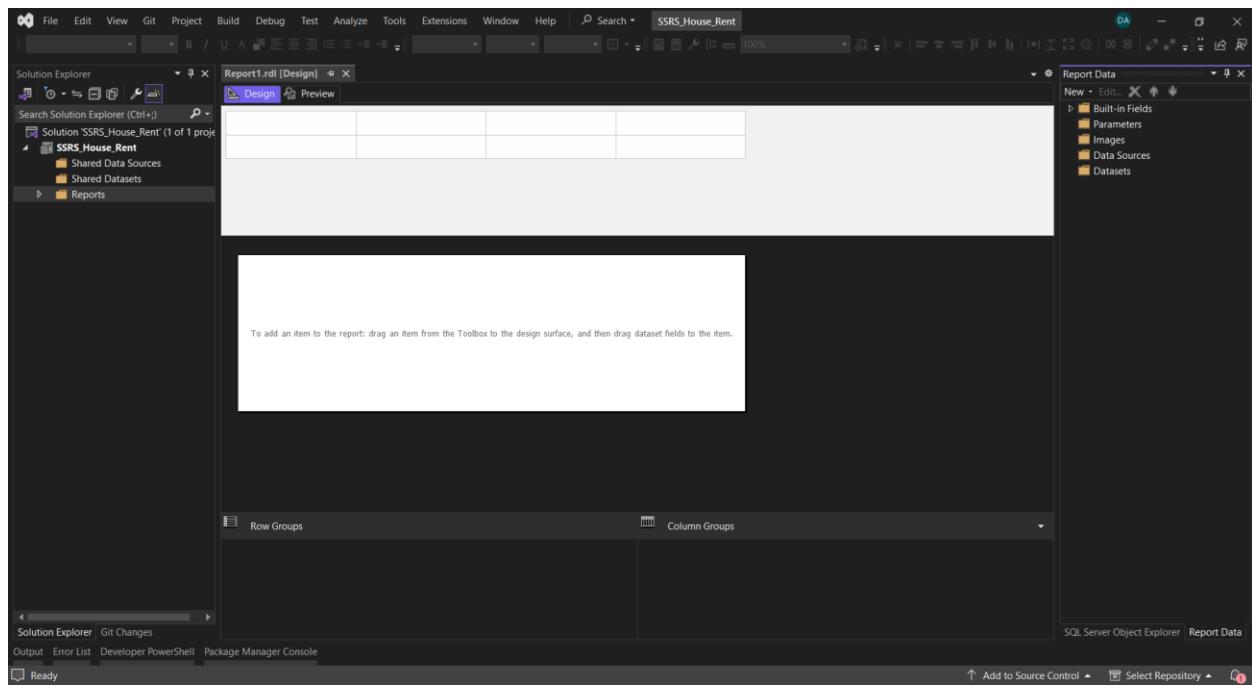
Xuất hiện giao diện làm việc như hình dưới.



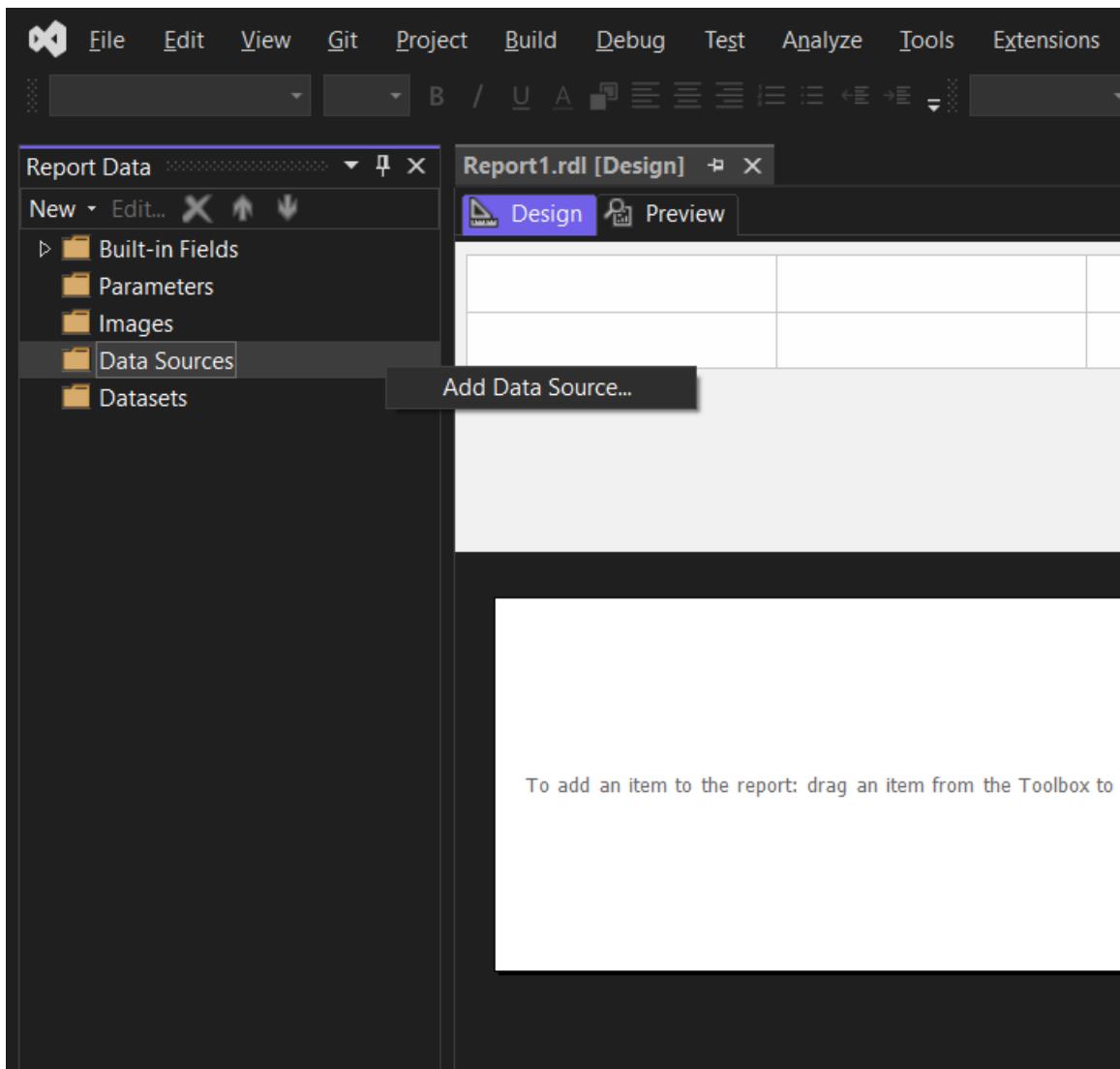
Tạo report bằng cách nhấn chuột phải vào Reports tại Solution Explorer, chọn Add -> New Item -> Report. Đặt tên và nhấn Add để thêm mới một Report.



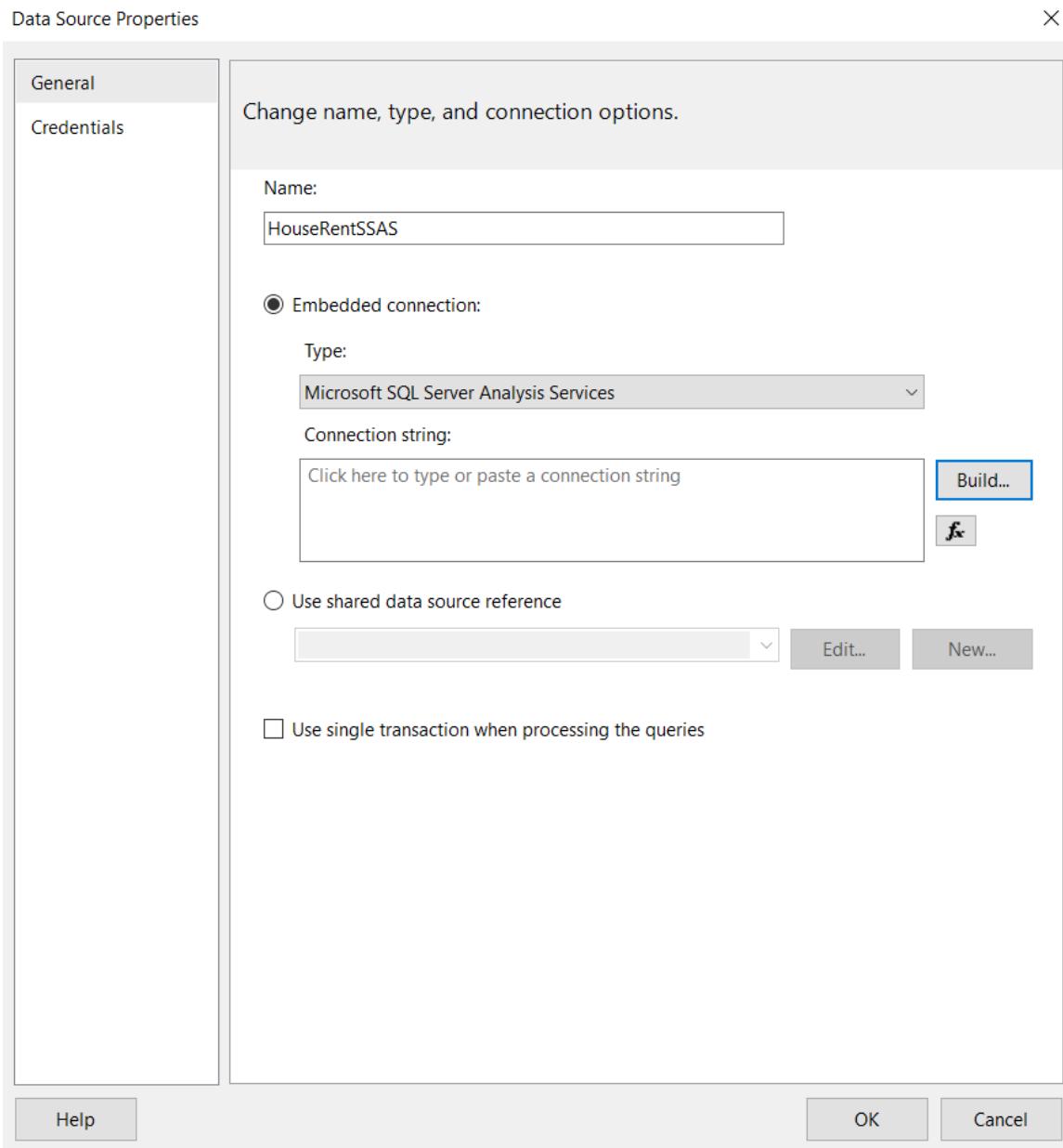
Xuất hiện màn hình giao diện làm việc của Report như hình dưới.



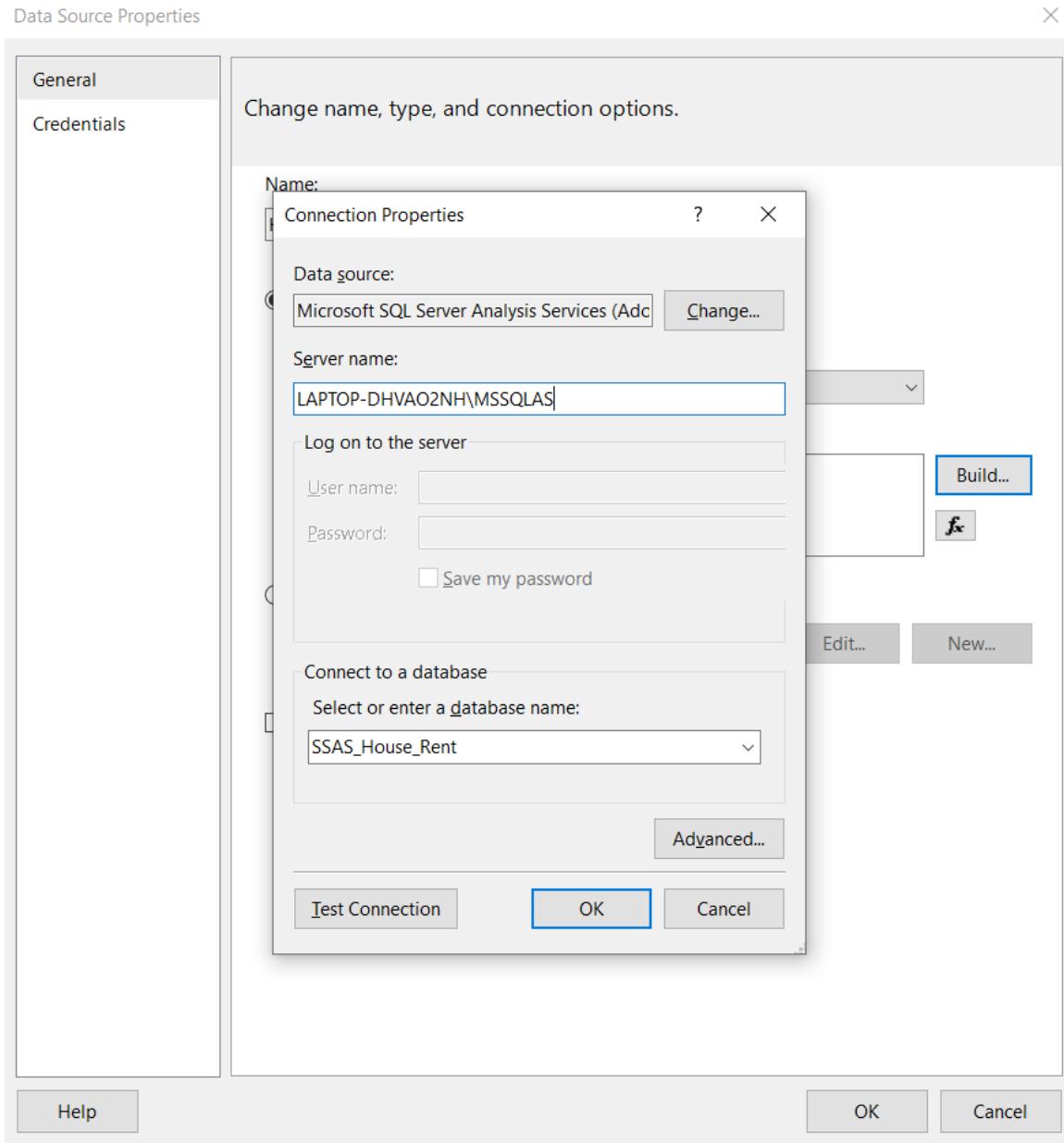
Nhấn chuột phải vào Data Source -> Add Data Source để thêm nguồn vào nguồn dữ liệu.

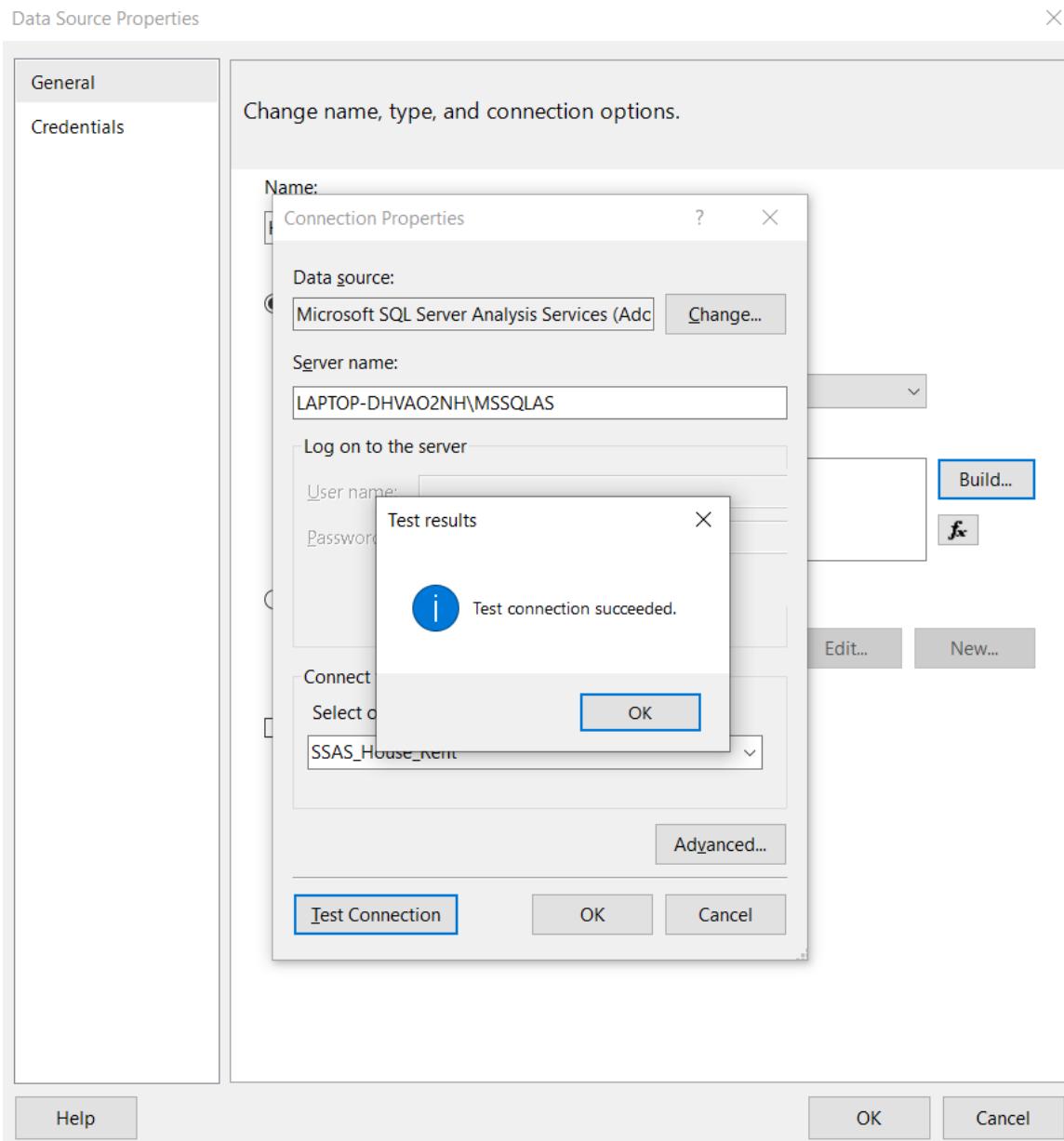


Tại phần General, đặt tên cho Data Source tại mục Name. Mặc định chọn Embedded connections, Type SQL Server Analysis. Nhấn nút Build để tạo Connection string.

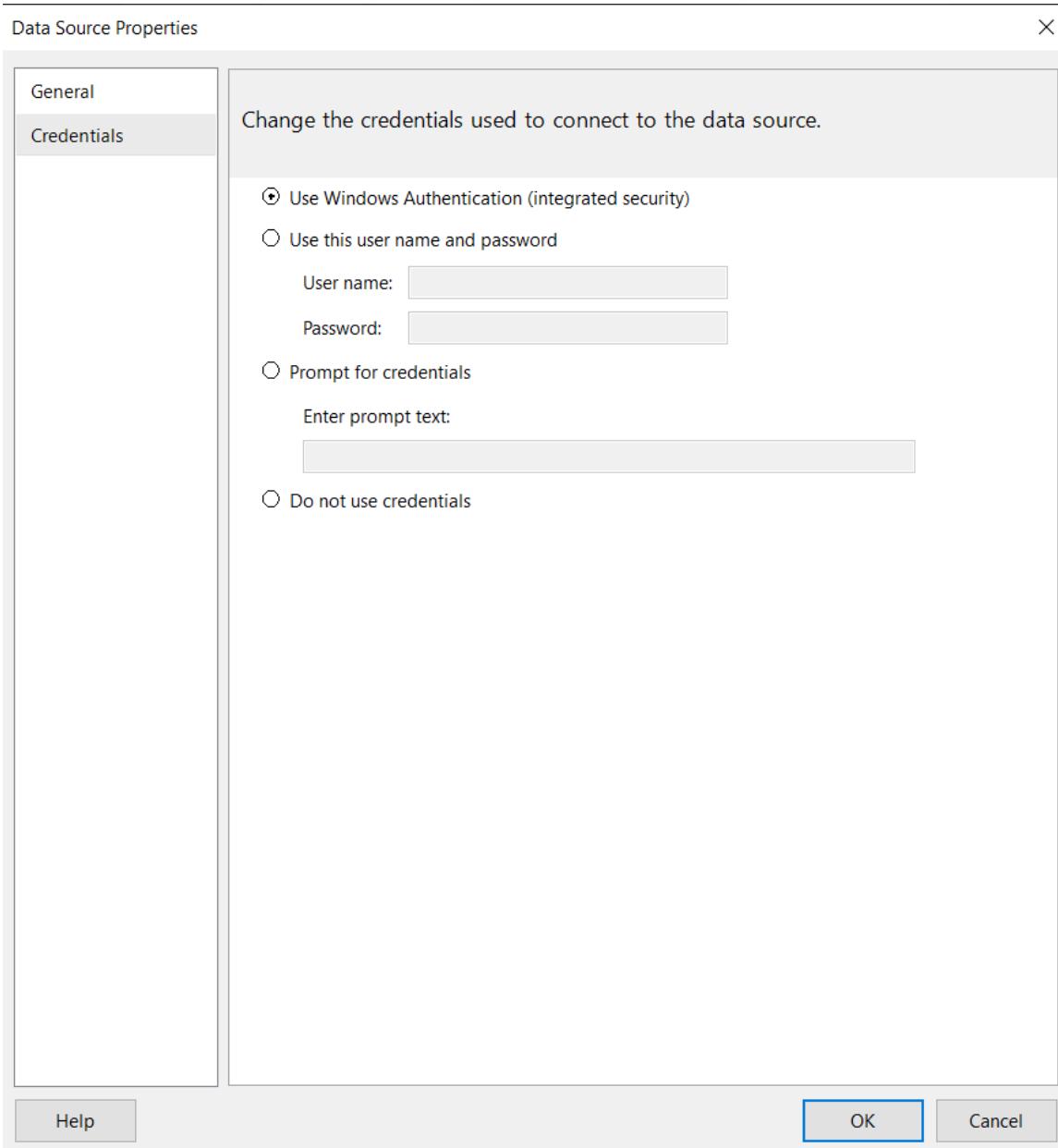


Nhấn vào Build, tại Server name gõ tên LAPTOP-DHVAO2NH\MSSQLAS. Bên dưới mục Select or enter a database name chọn SSAS_House_Rent. Nhấn Test Connection để kiểm tra kết nối.

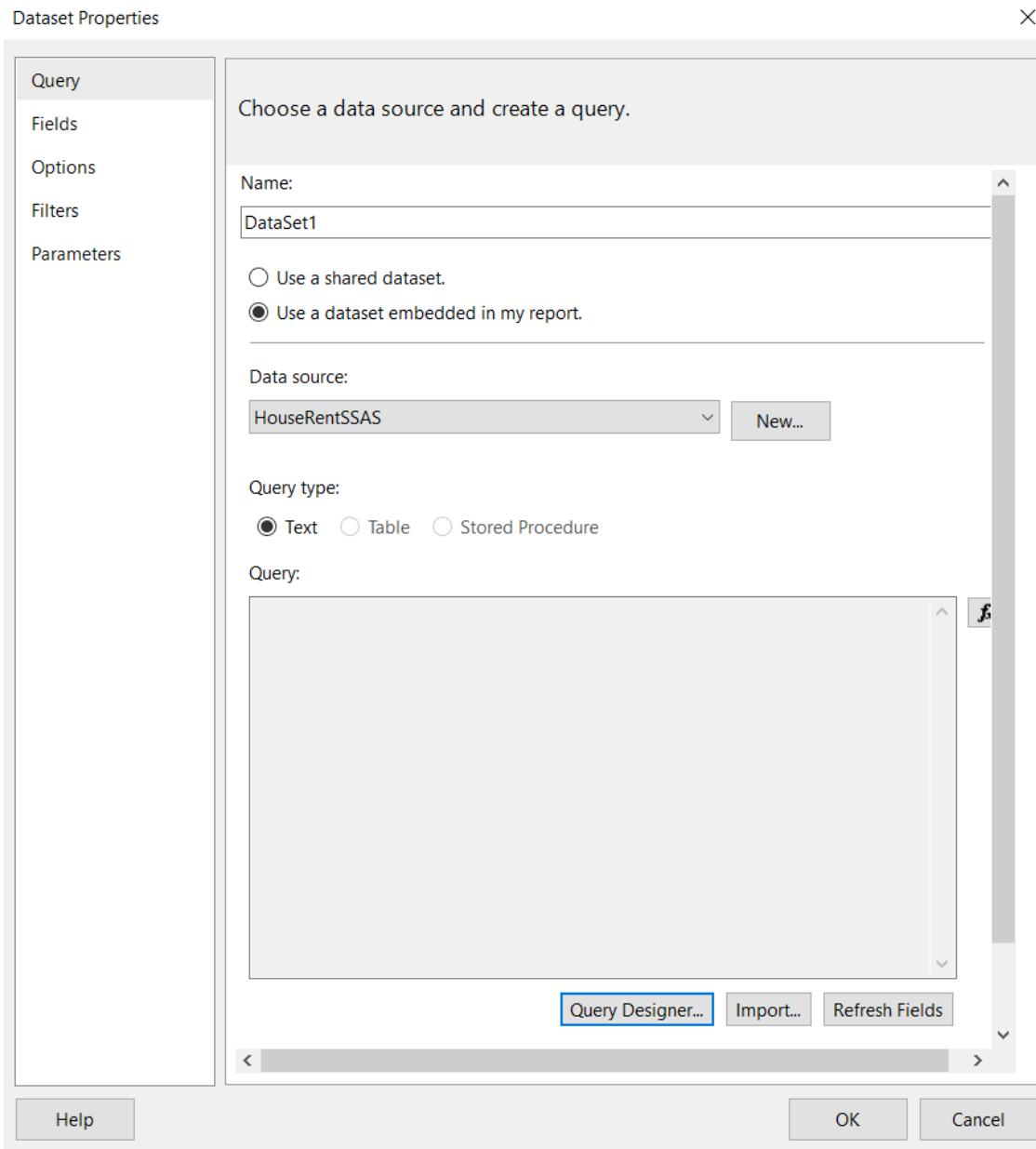




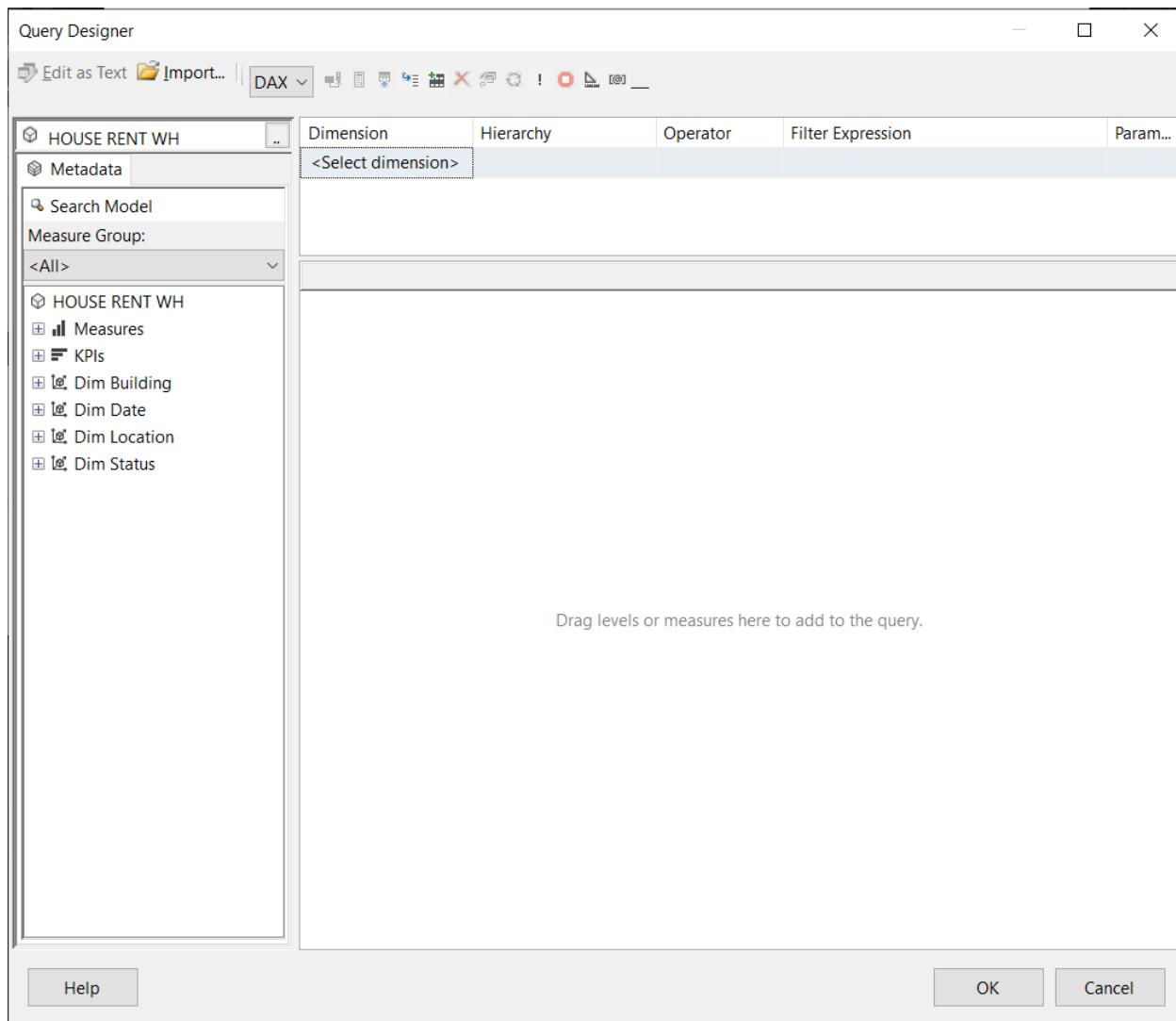
Tại phần Credentials, kiểm tra đã chọn Use Window Authentication chưa. Sau đó nhấn Ok để xác nhận thêm Data Source.



Nhấn chuột phải vào Datasets -> Add Dataset. Cửa sổ Data Properties hiện ra, đặt tên cho dataset. Chọn Use a dataset embedded in my report. Chọn Data Source vừa tạo ở trên. Nhấn Query Designer để truy suất dữ liệu.



Cửa sổ Query Designer hiện ra. Thực hiện truy vấn tương tự như công cụ Manual của SSAS. Nhấn OK để xác nhận chọn kết quả của câu truy vấn làm Database.



4.1.3. Report 1: Câu truy vấn 3: Liệt kê các nhà có diện tích trên 600, có 1 phòng tắm và không có nội thất.

Query Designer

The screenshot shows the Microsoft Query Designer interface. On the left, there's a navigation pane with a tree view of the data source 'HOUSE RENT WH'. The tree includes 'Metadata', 'Functions', 'Search Model', 'Measure Group' (set to '<All>'), and several dimension groups like 'HOUSE RENT WH', 'Measures', 'KPIs', 'Dim Building', 'Dim Date', 'Dim Location', and 'Dim Status'. The main area contains an MDX query:

```
SELECT [Measures].[Size] ON COLUMNS,  
FILTER(  
    [Dim Building].[Building ID].[Building ID].MEMBERS,  
    [Dim Status].[Bathroom].[1] AND  
    [Measures].[Size] > 600  
) ON ROWS  
FROM [HOUSE RENT WH]  
WHERE (  
    [Dim Status].[Furnishing Status].[Unfurnished]  
)
```

Below the query, there's a preview table with columns 'Building ID' and 'Size'. At the bottom, there's a link 'Click to run the query.' and buttons for 'Help', 'OK', and 'Cancel'.

Report1.rdl [Design]

Report2.rdl [Design]

| Building ID | Size |
|---------------|--------|
| [Building_ID] | [Size] |

| Building ID | Size |
|-------------|------|
| 696 | 620 |
| 704 | 625 |
| 718 | 630 |
| 728 | 641 |
| 729 | 647 |
| 732 | 649 |
| 766 | 650 |
| 767 | 650 |
| 768 | 650 |
| 770 | 650 |
| 771 | 650 |
| 772 | 650 |
| 773 | 650 |
| 775 | 650 |
| 776 | 650 |
| 777 | 650 |
| 778 | 650 |
| 779 | 650 |

4.1.4. Report 2: Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”.

Query Designer

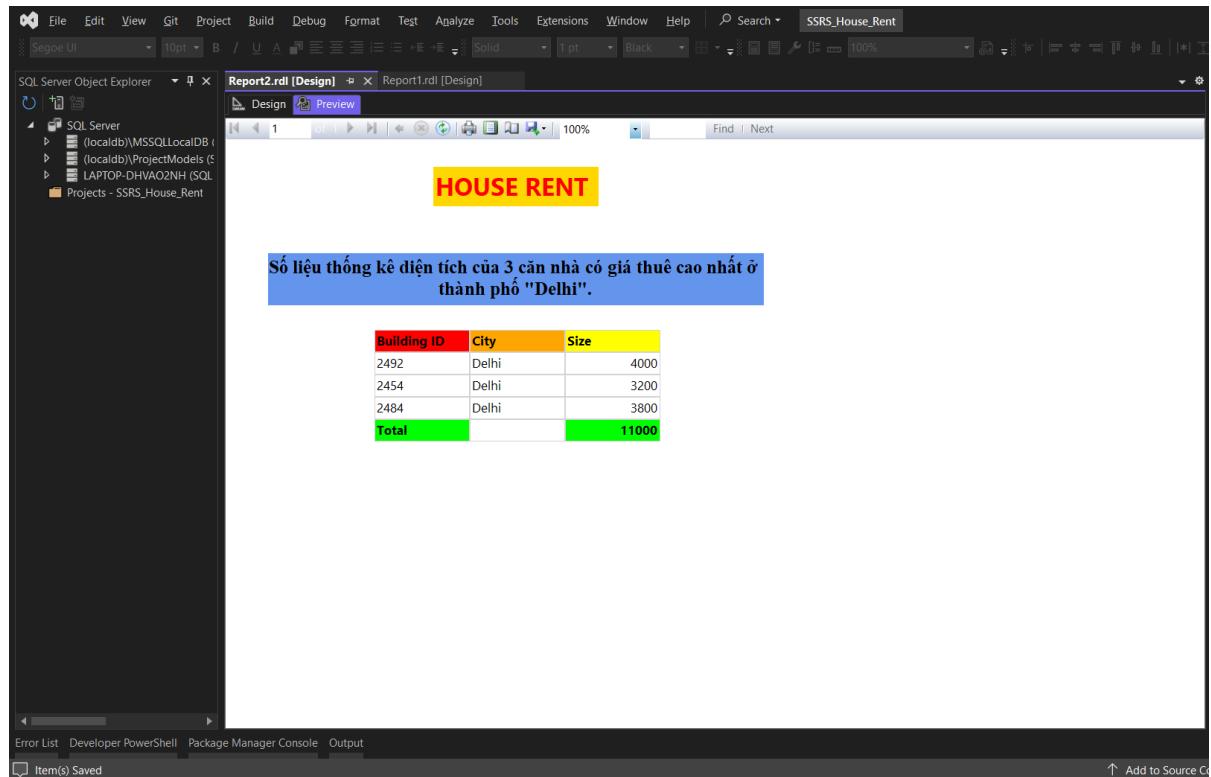
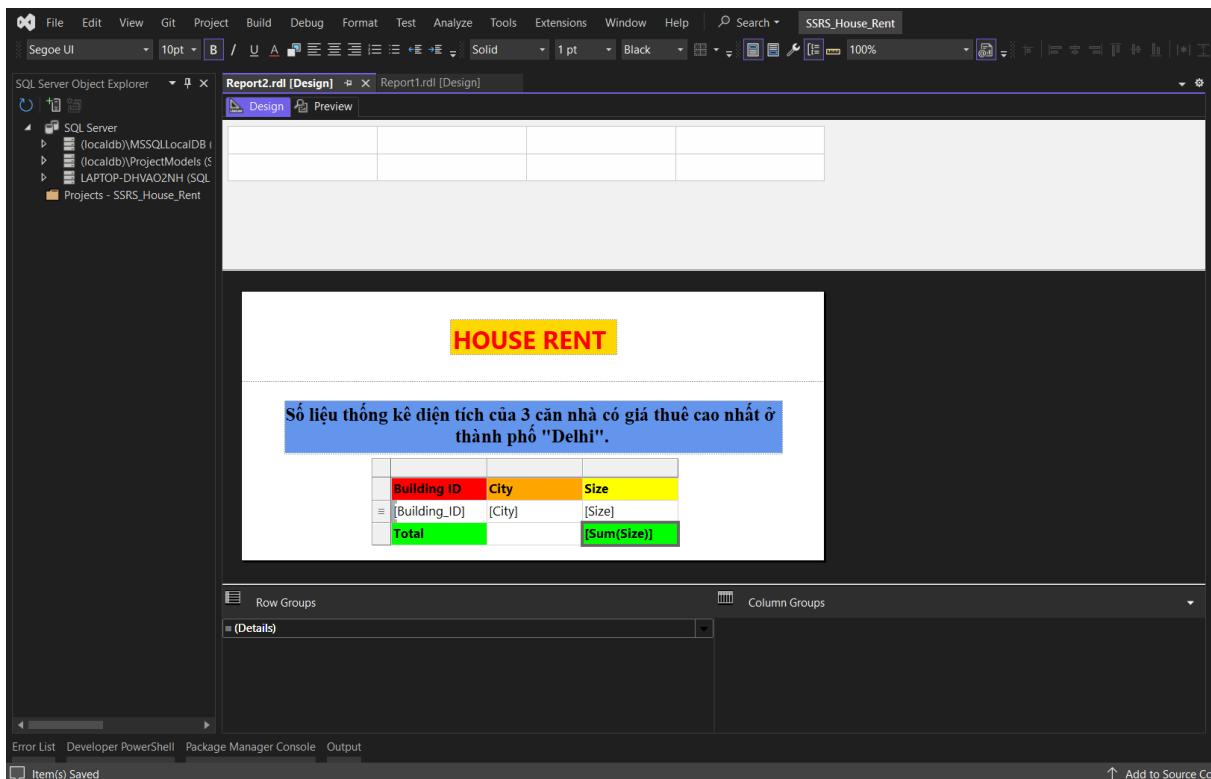
The screenshot shows the Microsoft Analysis Services Query Designer interface. On the left, there's a navigation pane with a tree view of the data source, currently expanded to show 'HOUSE RENT WH' and its children: 'Metadata', 'Functions', 'Search Model', 'Measure Group', and several dimension tables like 'Dim Building', 'Dim Date', etc. A dropdown menu under 'Measure Group' is set to '<All>'. The main area contains a T-SQL query:

```
SELECT  
    [Measures].[SIZE] ON COLUMNS,  
    TOPCOUNT(  
        [Dim Building].[Building ID].[Building ID].MEMBERS * [Dim Location].[City].[Delhi],  
        3,  
        [Measures].[Rent]  
    ) ON ROWS  
FROM [HOUSE RENT WH]
```

Below the query, a preview grid shows the results:

| Building ID | City | Size |
|-------------|-------|------|
| 2492 | Delhi | 4000 |
| 2454 | Delhi | 3200 |
| 2484 | Delhi | 3800 |

At the bottom right are 'OK' and 'Cancel' buttons.



4.1.5. Report 3: Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”.

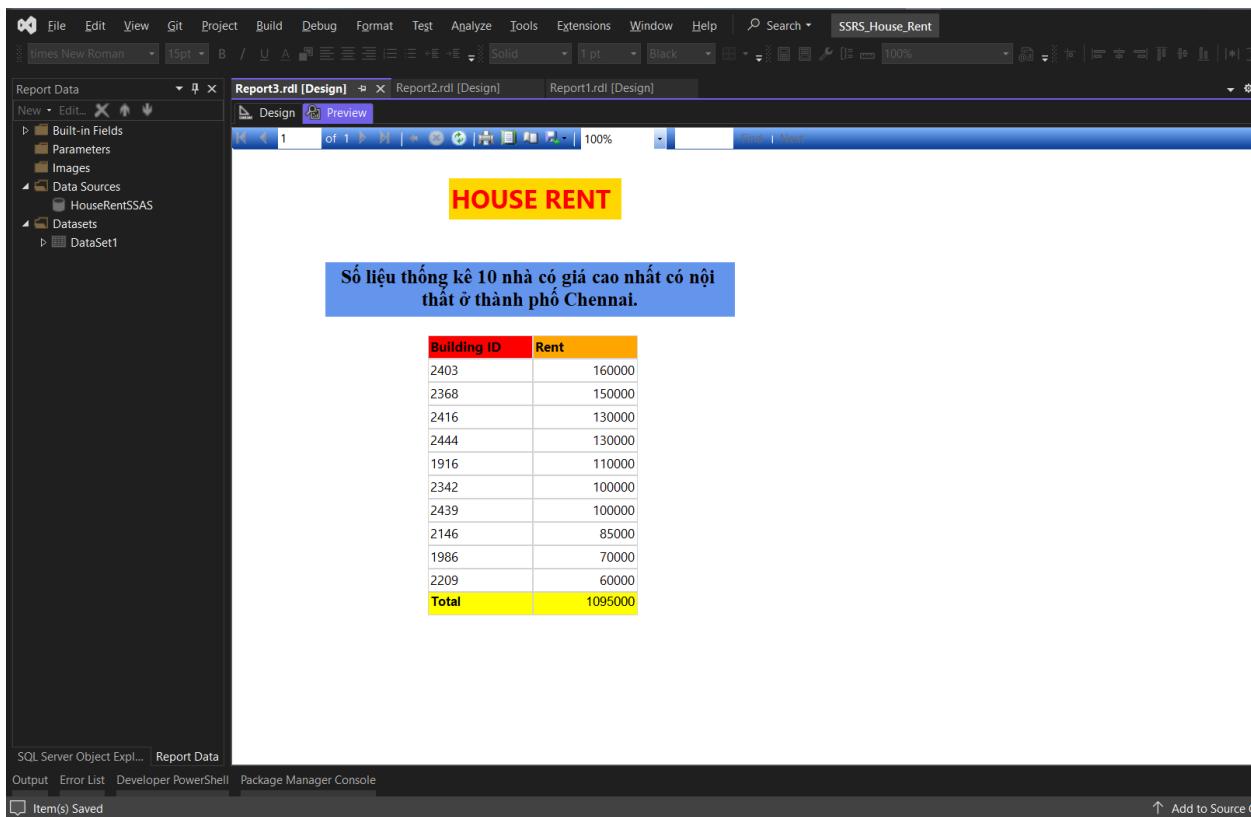
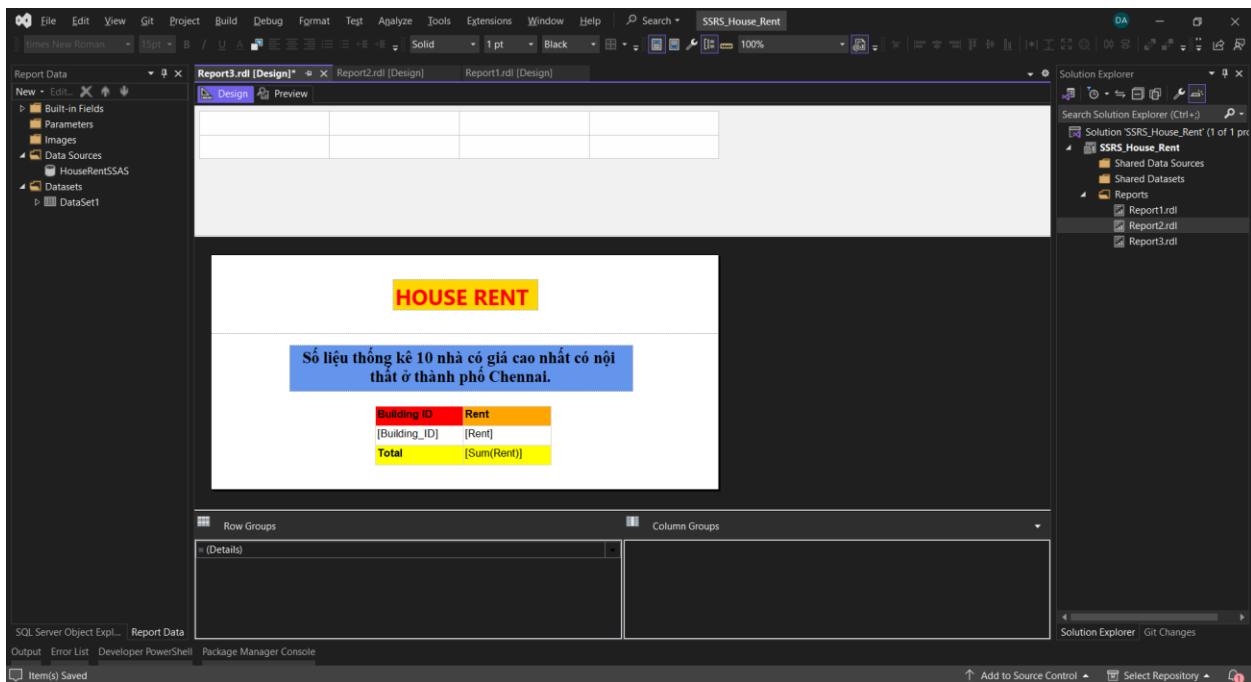
Query Designer

```

SELECT [Measures].[Rent] ON COLUMNS,
TOPCOUNT(
  FILTER(
    [Dim Building].[Building ID].[Building ID].MEMBERS,
    [Dim Status].[Furnishing Status].&[Furnished]
  ),
  10,
  [Measures].[Rent]
) ON ROWS
FROM [HOUSE RENT WH]
WHERE [Dim Location].[City].&[Chennai];
  
```

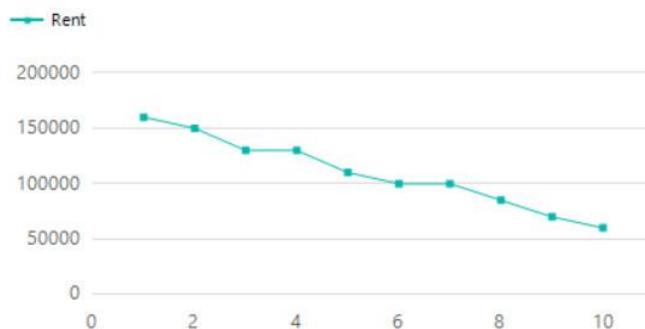
| Building ID | Rent |
|-------------|--------|
| 2403 | 160000 |
| 2368 | 150000 |
| 2416 | 130000 |
| 2444 | 130000 |
| 1916 | 110000 |
| 2342 | 100000 |
| 2439 | 100000 |
| 2146 | 85000 |
| 1986 | 70000 |
| 2209 | 60000 |

Help OK Cancel



| City | Building ID | Rent |
|--------------|-------------|----------------|
| Chennai | 2403 | 160000 |
| Chennai | 2368 | 150000 |
| Chennai | 2416 | 130000 |
| Chennai | 2444 | 130000 |
| Chennai | 1916 | 110000 |
| Chennai | 2342 | 100000 |
| Chennai | 2439 | 100000 |
| Chennai | 2146 | 85000 |
| Chennai | 1986 | 70000 |
| Chennai | 2209 | 60000 |
| Total | | 1095000 |

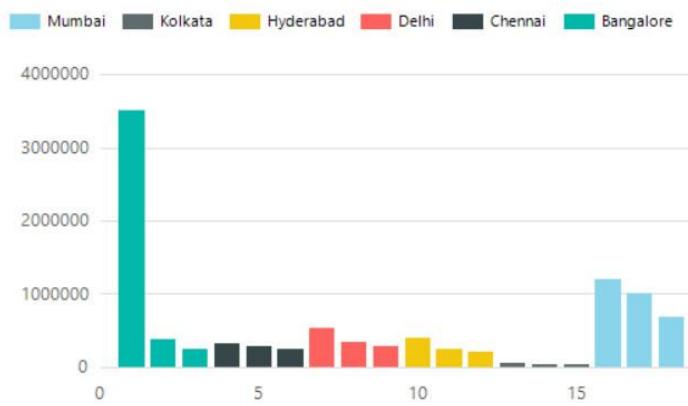
Top 10 căn nhà có Furnished với giá thuê cao nhất ở Chennai.



4.1.6. Report 4: Câu truy vấn: Top 3 nhà có giá cao nhất có Semi-Furnished ở mỗi thành phố

| City | Building ID | Rent |
|-----------|-------------|---------|
| Bangalore | 2383 | 3500000 |
| Bangalore | 2474 | 380000 |
| Bangalore | 2509 | 250000 |
| Chennai | 2479 | 330000 |
| Chennai | 2502 | 280000 |
| Chennai | 2473 | 250000 |
| Delhi | 2492 | 530000 |
| Delhi | 2454 | 350000 |
| Delhi | 2484 | 280000 |
| Hyderabad | 2511 | 400000 |
| Hyderabad | 2503 | 250000 |
| Hyderabad | 2506 | 200000 |
| Kolkata | 2126 | 60000 |
| Kolkata | 2471 | 40000 |
| Kolkata | 1039 | 35000 |
| Mumbai | 2508 | 1200000 |
| Mumbai | 2447 | 1000000 |
| Mumbai | 2264 | 680000 |

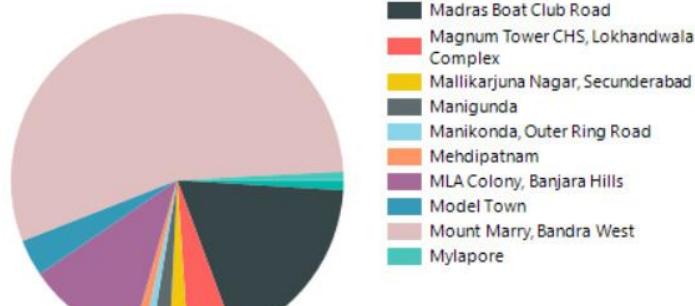
Top 3 nhà có giá cao nhất có Semi-Furnished ở mỗi thành phố.



4.1.7. Report 5: Câu truy vấn: Liệt kê những nhà có 4 phòng tắm và Area Locality bắt đầu bằng chữ M.

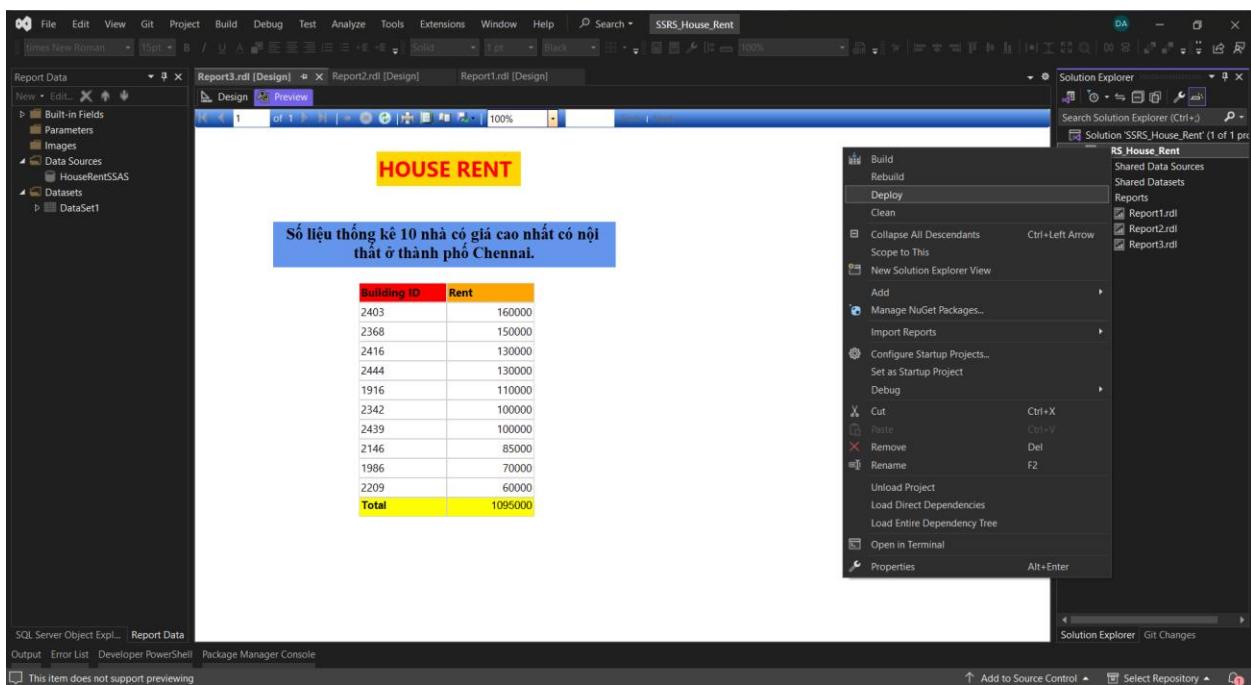
| Area Locality | Rent |
|---------------------------------------|----------------|
| Manikonda, Outer Ring Road | 9000 |
| Mylapore | 9000 |
| Madhapur | 10500 |
| Mehdipatnam | 11000 |
| Mallikarjuna Nagar, Secunderabad | 20000 |
| Manigunda | 20000 |
| Model Town | 39000 |
| Magnum Tower CHS, Lokhandwala Complex | 50000 |
| MLA Colony, Banjara Hills | 120000 |
| Madras Boat Club Road | 200000 |
| Mount Marry, Bandra West | 600000 |
| Total | 1088500 |

Những nhà có 4 phòng tắm và có Area Locality bắt đầu bằng chữ M.

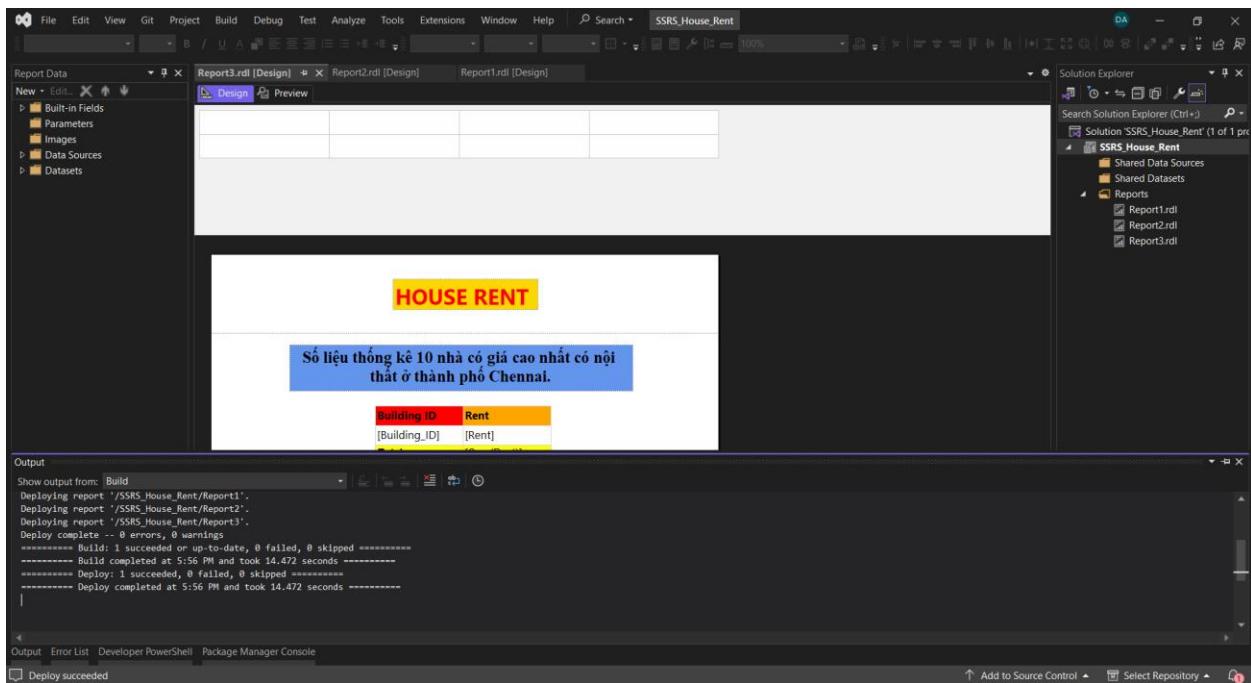


4.1.8. Triển khai SSRS trên Visual Studio

Nhấn chuột phải vào tên project -> chọn Deploy để triển khai các báo cáo.



Nếu deploy thành công thì, Output hiện ra thông báo như dưới.



Truy cập vào đường link được cấu hình từ trước để truy cập các báo cáo
<http://localhost/reportserver>

The screenshot shows a Microsoft Report Server interface with four tabs at the top: 'localhost/ReportServer - /SSRS', 'Report1 - Report Viewer', 'Report2 - Report Viewer', and 'Report3 - Report Viewer'. The main content area displays a report titled 'HOUSE RENT' in a yellow header box. Below the title is a blue box containing the text: 'Số liệu thống kê các nhà có diện tích trên 600, có 1 phòng tắm và không nội thất.' A table below lists building IDs and sizes, with rows numbered 696 to 778. The table has two columns: 'Building ID' and 'Size'.

| Building ID | Size |
|-------------|------|
| 696 | 620 |
| 704 | 625 |
| 718 | 630 |
| 728 | 641 |
| 729 | 647 |
| 732 | 649 |
| 766 | 650 |
| 767 | 650 |
| 768 | 650 |
| 770 | 650 |
| 771 | 650 |
| 772 | 650 |
| 773 | 650 |
| 775 | 650 |
| 776 | 650 |
| 777 | 650 |
| 778 | 650 |

The screenshot shows a Microsoft Report Server interface with a single report titled "Report2 - Report Viewer". The report has a yellow header bar with the text "HOUSE RENT". Below it is a blue box containing the text: "Số liệu thống kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố 'Delhi'." A table follows, showing the area of three houses in Delhi:

| Building ID | City | Size |
|--------------|-------|--------------|
| 2492 | Delhi | 4000 |
| 2454 | Delhi | 3200 |
| 2484 | Delhi | 3800 |
| Total | | 11000 |

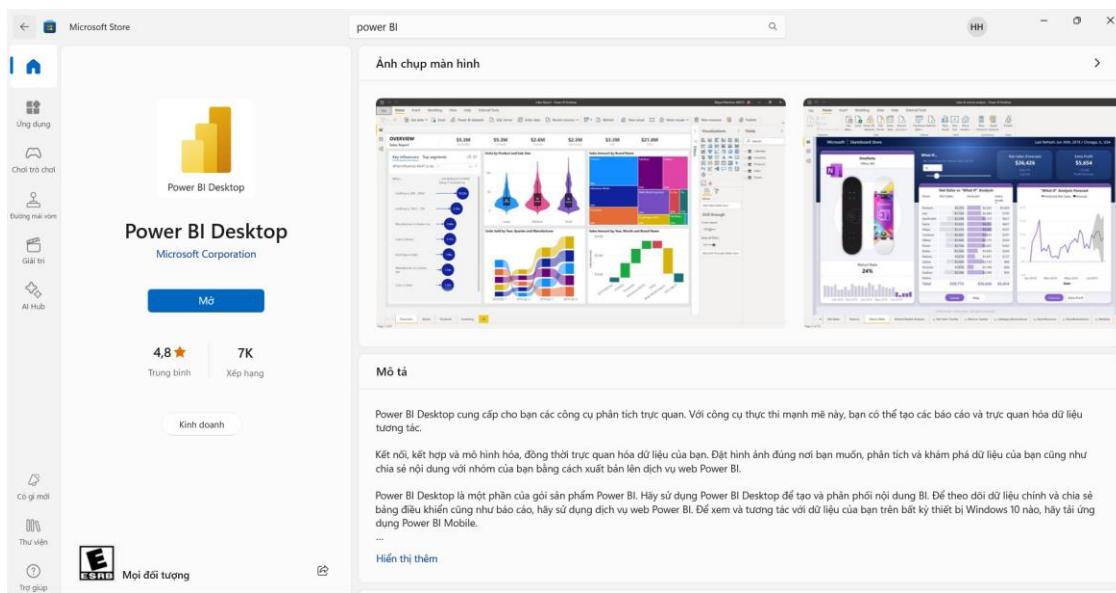
The screenshot shows a Microsoft Report Server interface with four reports listed in the title bar: "ReportServer - /SSRS", "Report1 - Report Viewer", "Report2 - Report Viewer", and "Report3 - Report Viewer". The report viewer for "Report3" is active, showing a yellow header bar with the text "HOUSE RENT". Below it is a blue box containing the text: "Số liệu thống kê 10 nhà có giá cao nhất có nội thất ở thành phố Chennai." A table follows, showing the rent of ten houses in Chennai:

| Building ID | Rent |
|--------------|----------------|
| 2403 | 160000 |
| 2368 | 150000 |
| 2416 | 130000 |
| 2444 | 130000 |
| 1916 | 110000 |
| 2342 | 100000 |
| 2439 | 100000 |
| 2146 | 85000 |
| 1986 | 70000 |
| 2209 | 60000 |
| Total | 1095000 |

4.2. Report với Power BI

4.2.1. Cài đặt Power BI

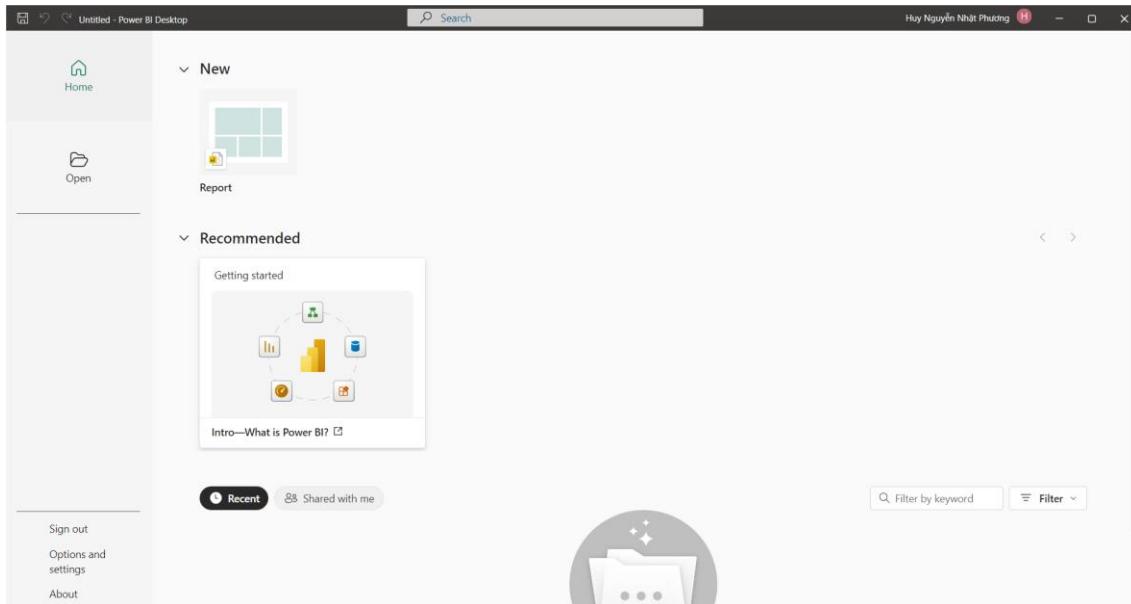
- Truy cập vào Microsoft Store, tìm ứng dụng Power BI, tiến hành tải về và cài đặt.



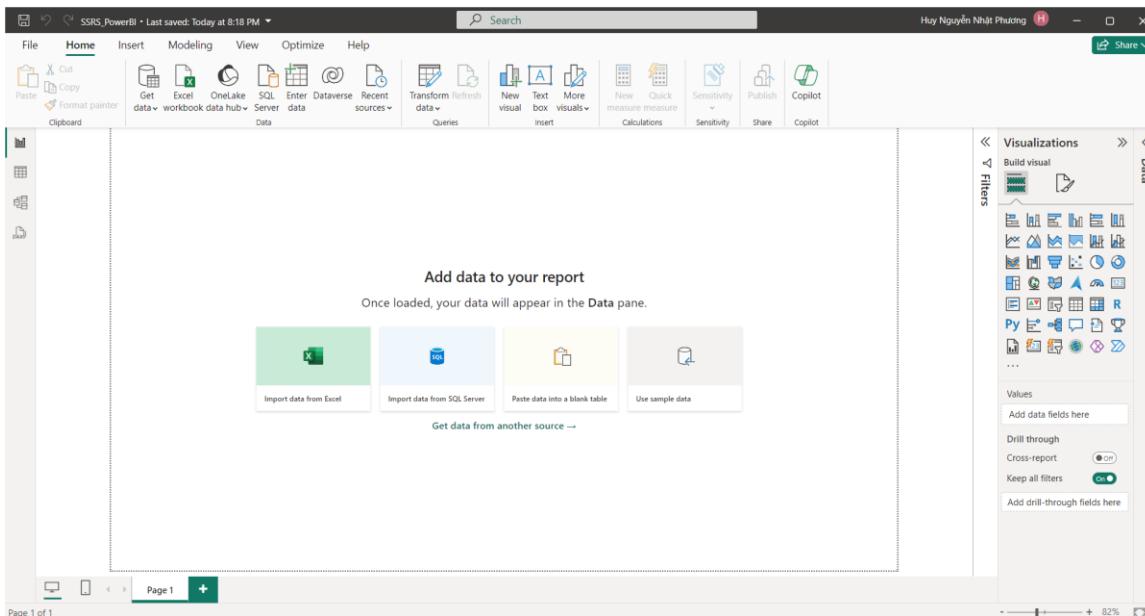
- Sau khi cài đặt, đăng nhập bằng tài khoản Microsoft đã đăng ký.

4.2.2. Tạo project SSRS trên Power BI

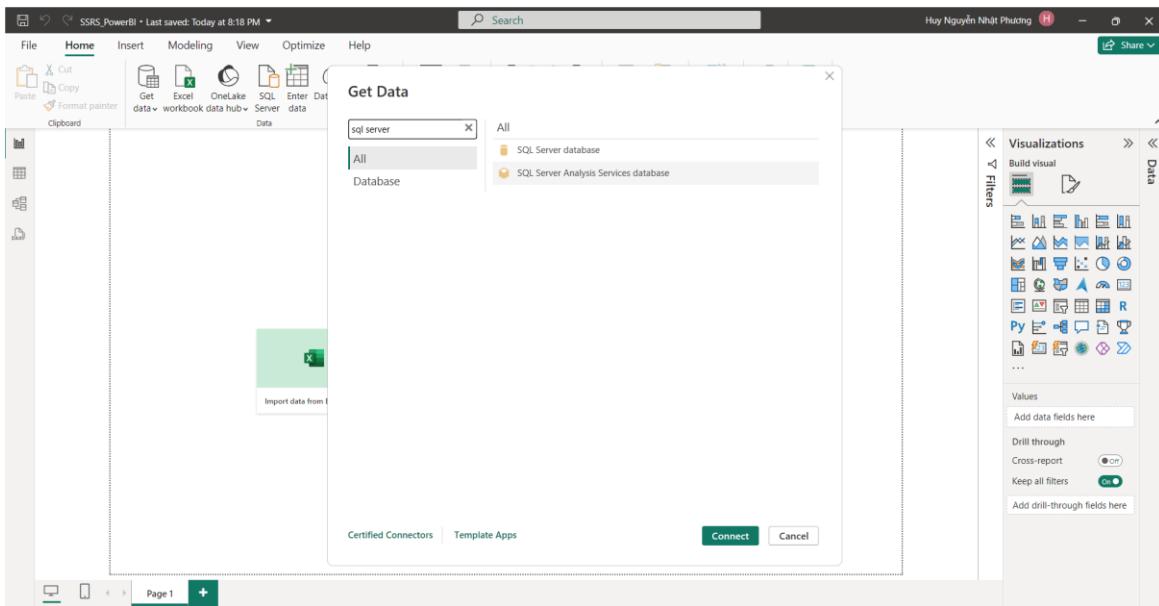
- Mở Power BI, giao diện cửa sổ của phần mềm hiển thị như sau:



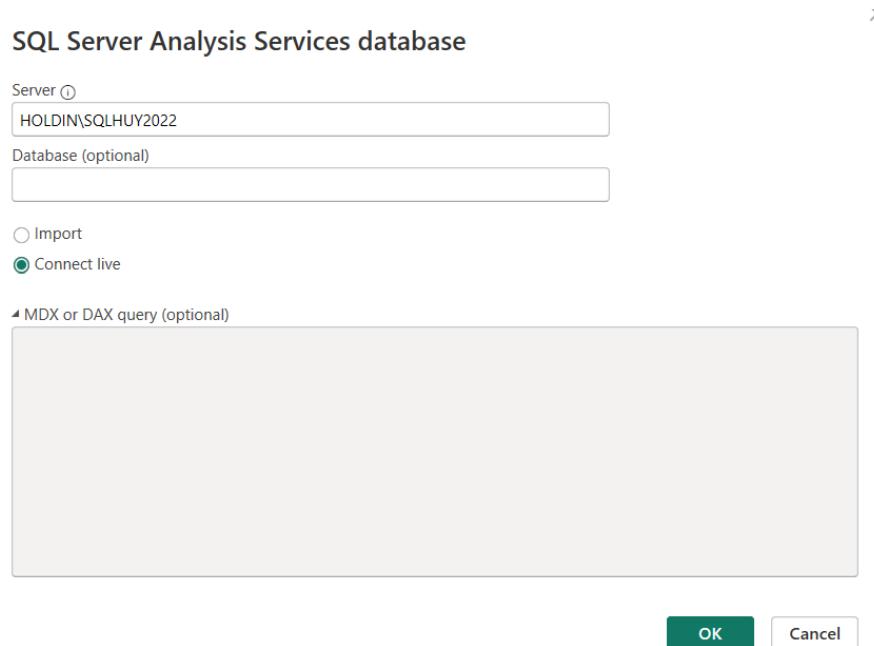
- Chọn New, tạo mới một project và lưu.



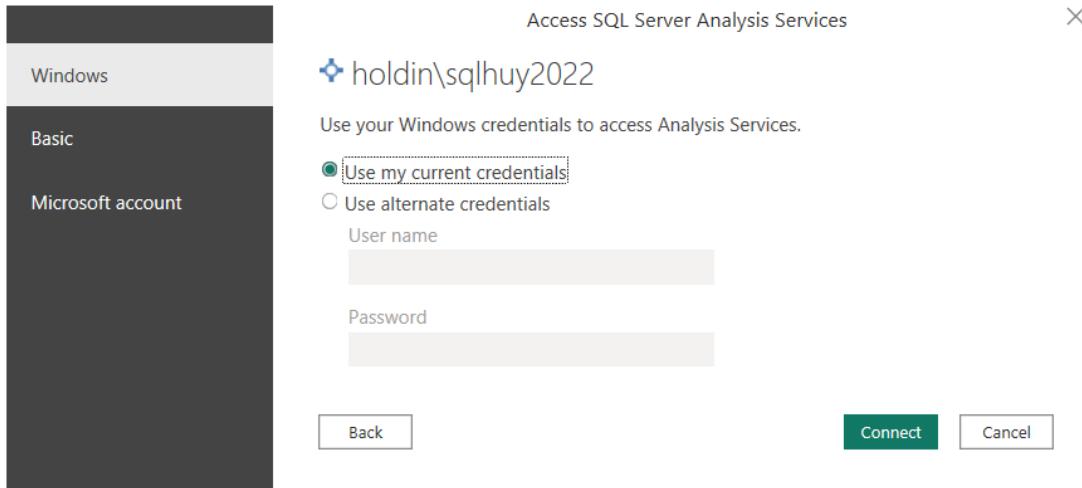
- Để tạo dataset, chọn Get data – tìm kiếm và chọn Sql Server Analysis Services database. Nhấn Connect.



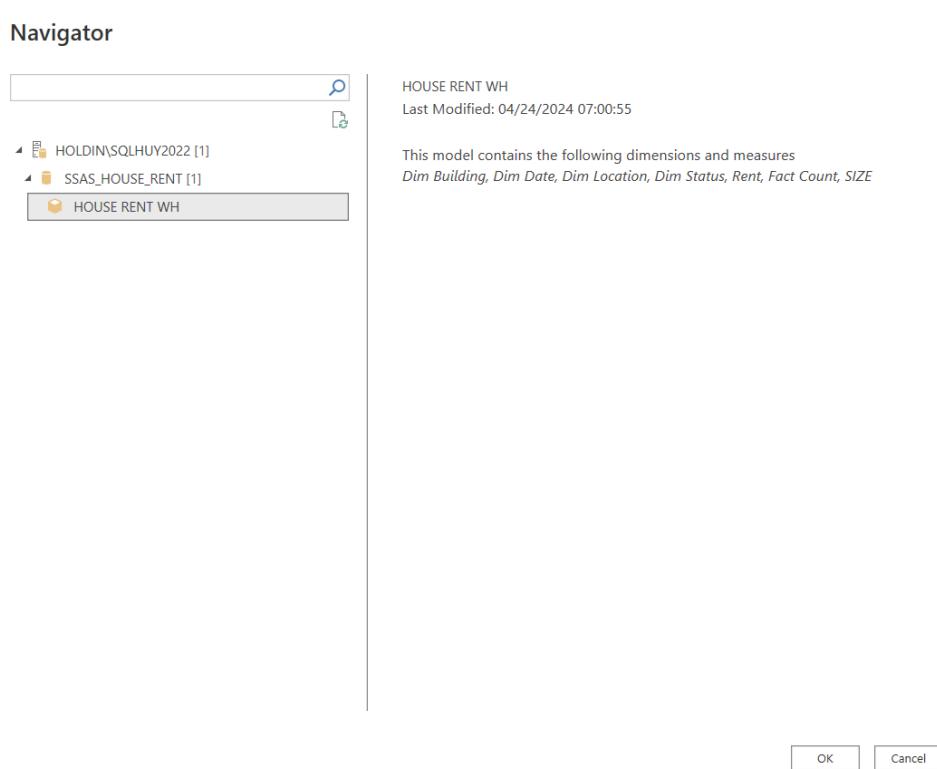
- Nhập tên Server tương ứng, chọn Connect Live, nhấn OK.



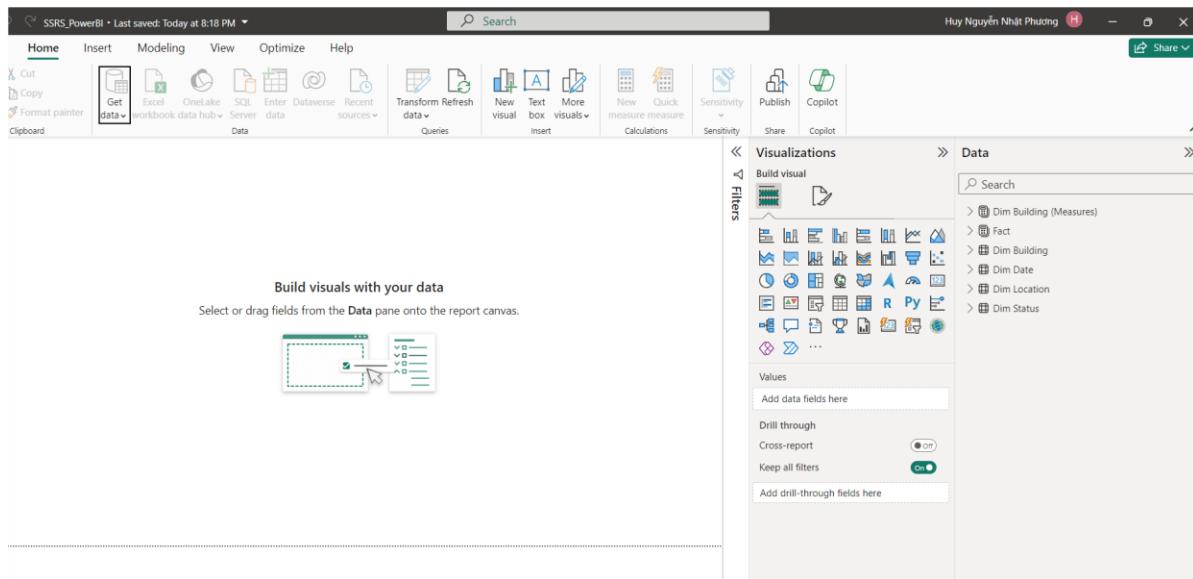
- Chọn Use my current credentials và nhấn Connect.



- Tiếp theo hộp thoại Navigator hiện ra, chọn Cube cần thiết và nhấn OK.



- Sau khi kết nối thành công thì các Measure và Dimension sẽ xuất hiện.



4.2.3. Report 6: Câu truy vấn 3: Liệt kê các nhà có diện tích trên 600, có 1 phòng tắm và không có nội thất.

- Dữ liệu của Report 4 được lấy từ câu truy vấn 3.

SQL Server Analysis Services database

Server ①
localhost

Database
SSAS_HOUSE_RENT

Import
 DirectQuery

▲ MDX or DAX query (optional)

```
SELECT [Measures].[Size] ON COLUMNS,
    FILTER(
        [Dim Building].[Building ID].[Building ID].MEMBERS,
        [Dim Status].[Bathroom].[1] AND
        [Measures].[Size] > 600
    ) ON ROWS
FROM [HOUSE RENT WH]
WHERE (
    [Dim Status].[Furnishing Status].[Unfurnished]
);
```

OK Cancel

- Chọn Transform Data, chỉnh lại tên cột tương ứng sau đó ấn Close và Apply và tiến hành điều chỉnh trong Visualizations.

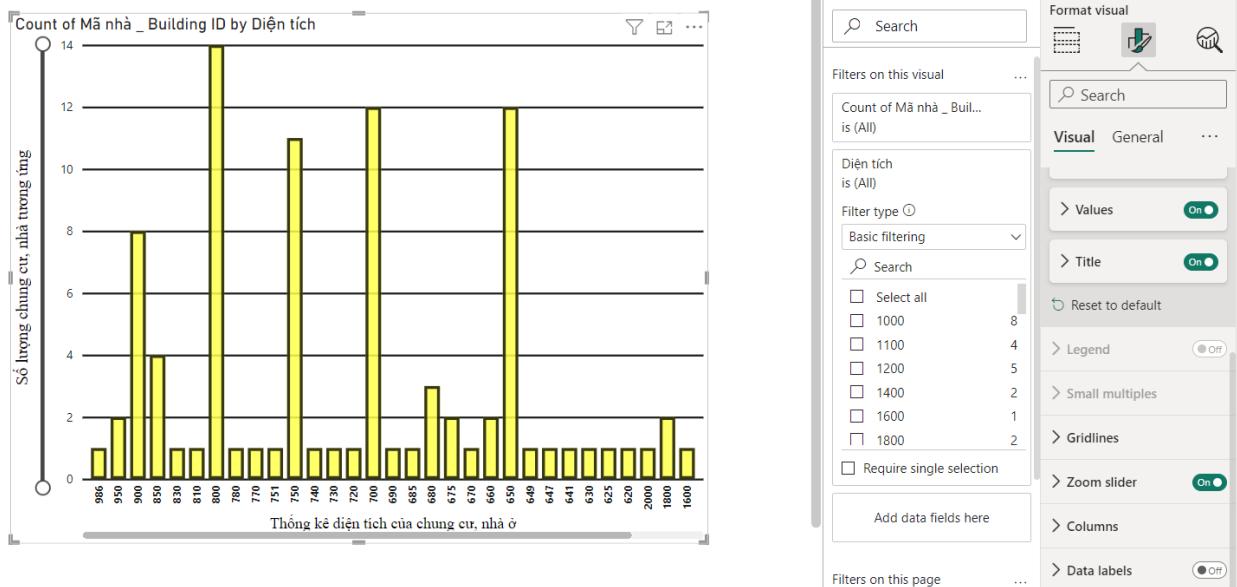
localhost: SSAS_HOUSE_RENT

| [Dim Building].[Building ID] | [Measures].[SIZE 1] |
|------------------------------|---------------------|
| 696 | 620 |
| 704 | 625 |
| 718 | 630 |
| 728 | 641 |
| 729 | 647 |
| 732 | 649 |
| 766 | 650 |
| 767 | 650 |
| 768 | 650 |
| 770 | 650 |
| 771 | 650 |
| 772 | 650 |
| 773 | 650 |
| 775 | 650 |
| 776 | 650 |
| 777 | 650 |
| 778 | 650 |
| 779 | 650 |
| 787 | 660 |
| 788 | 660 |

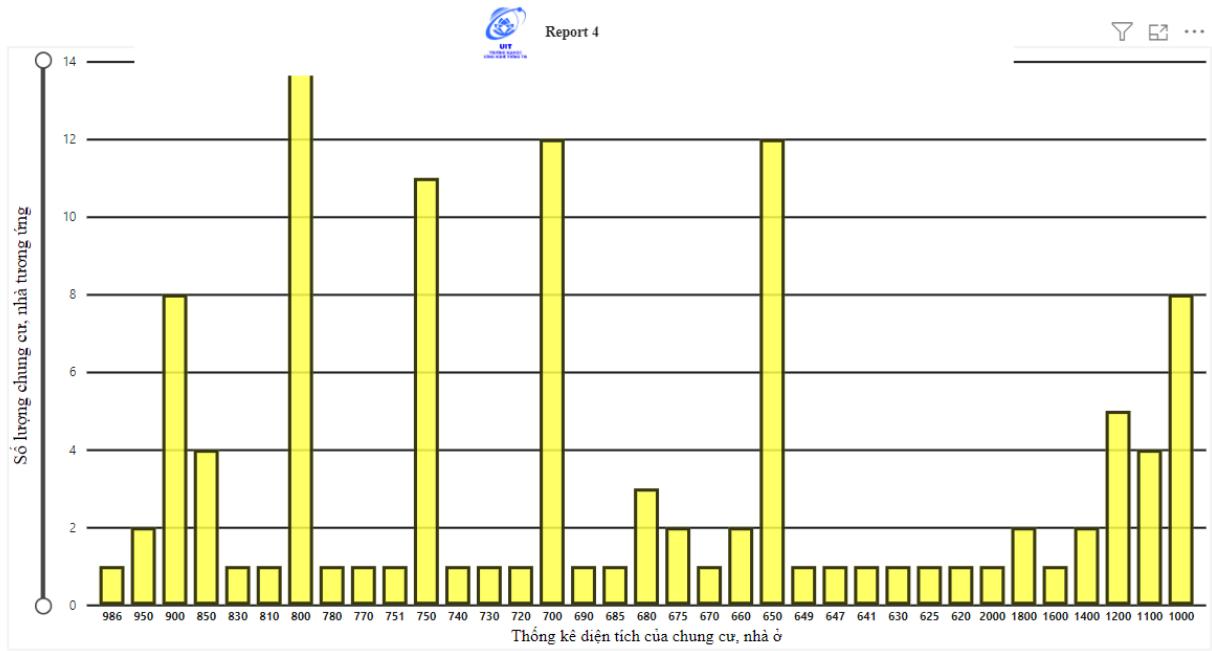
💡 The data in the preview has been truncated due to size limits.

Load Transform Data Cancel

- Chọn Clustered column Chart trong Build Visual.
- Trong Format Visual điều chỉnh biểu đồ để hiển thị trực quan dữ liệu.
- Điều chỉnh trục X-axis, Y-axis, Range, Values, Title, Column...



- Trong Insert Textbox và Image để tạo tiêu đề cho báo cáo.
- Chọn File – Export to PDF để xuất báo cáo.



4.2.4. Report 7: Câu truy vấn 8: Liệt kê diện tích của 3 căn nhà có giá thuê cao nhất ở thành phố “Delhi”

- Đối với phần lấy dữ liệu của Report 5, nhập vào MDX của câu truy vấn số 8 Nhấn OK.

The screenshot shows the 'SQL Server Analysis Services database' interface. In the 'Server' dropdown, 'localhost' is selected. The 'Database' dropdown shows 'HOUSE RENT WH'. Below these, there are two radio buttons: 'Import' (selected) and 'Connect live'. A large text area contains the following MDX query:

```

SELECT
    [Measures].[SIZE_1] ON COLUMNS,
    TOPCOUNT(
        [Dim Building].[Building ID].[Building ID].MEMBERS * [Dim Location].[City].[Delhi],
        3,
        [Measures].[Rent]
    ) ON ROWS
FROM [HOUSE RENT WH]
  
```

- Cửa sổ dữ liệu hiện ra, nhấn Transform Data để biến đổi tên cột dữ liệu.

The screenshot shows the 'Transform Data' editor in Power BI. The top ribbon has tabs like File, Home, Transform, Add Column, View, Tools, and Help. The main area displays a table with three columns: 'Building ID', 'Thành phố', and 'Diện tích'. The 'Thành phố' column is highlighted. The right side of the screen shows the 'Query Settings' pane, which includes sections for 'PROPERTIES' (Name: Query8) and 'APPLIED STEPS' (Query1, Renamed Columns, Changed Type).

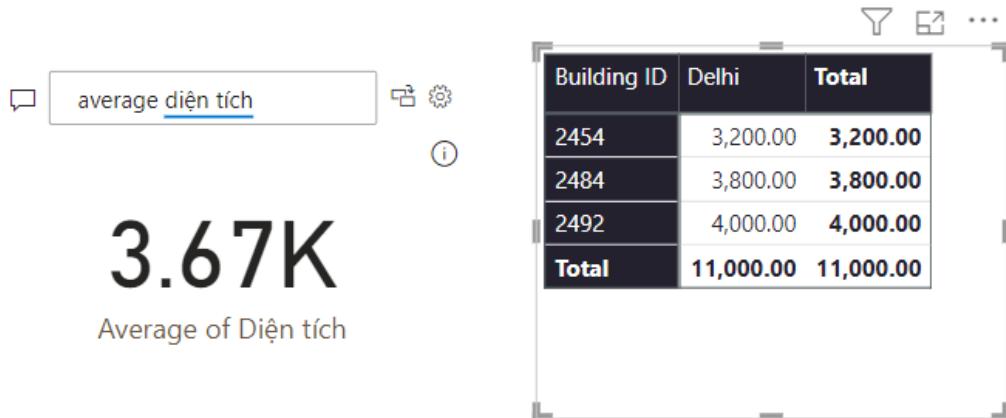
| | Building ID | Thành phố | Diện tích |
|---|-------------|-----------|-----------|
| 1 | 2492 | Delhi | 4000 |
| 2 | 2454 | Delhi | 3200 |
| 3 | 2484 | Delhi | 3800 |

- Sau khi đổi tên cột nhấn Close và Apply. Để trở về màn hình làm việc.
- Trong Visualizations chọn Matrix và phân bố các thuộc tính của Query8 ở Fields panel, ta thu được một ma trận dữ liệu như hình bên dưới:

The screenshot shows the Power BI desktop interface. On the left is a table visualization titled "Delhi" with columns "Building ID", "Delhi", and "Total". The data rows are: 2454 (3,200.00), 2484 (3,800.00), 2492 (4,000.00), and Total (11,000.00). The "Total" column is bolded. The ribbon on the right is visible, showing sections like "Visualizations" and "Data". The "Data" section shows filters for "Building ID", "Σ Diện tích", and "Thành phố". The "Rows" section shows "Building ID" and "Thành phố". The "Values" section shows "Sum of Diện tích". The "Drill through" section has "Cross-report" set to "Off". The "Keep all filters" button is turned on.

- Nhấn chuột 2 lần vào bảng dữ liệu ta có thể sử dụng tính năng gợi ý câu hỏi cho bộ dữ liệu.

The screenshot shows the Power BI desktop interface. On the left, there is a "Ask a question about your data" button. Below it are two blue buttons: "average diện tích" and "number of thành phố". On the right is a table visualization titled "Delhi" with columns "Building ID", "Delhi", and "Total". The data rows are: 2454 (3,200.00), 2484 (3,800.00), 2492 (4,000.00), and Total (11,000.00). The "Total" column is bolded.

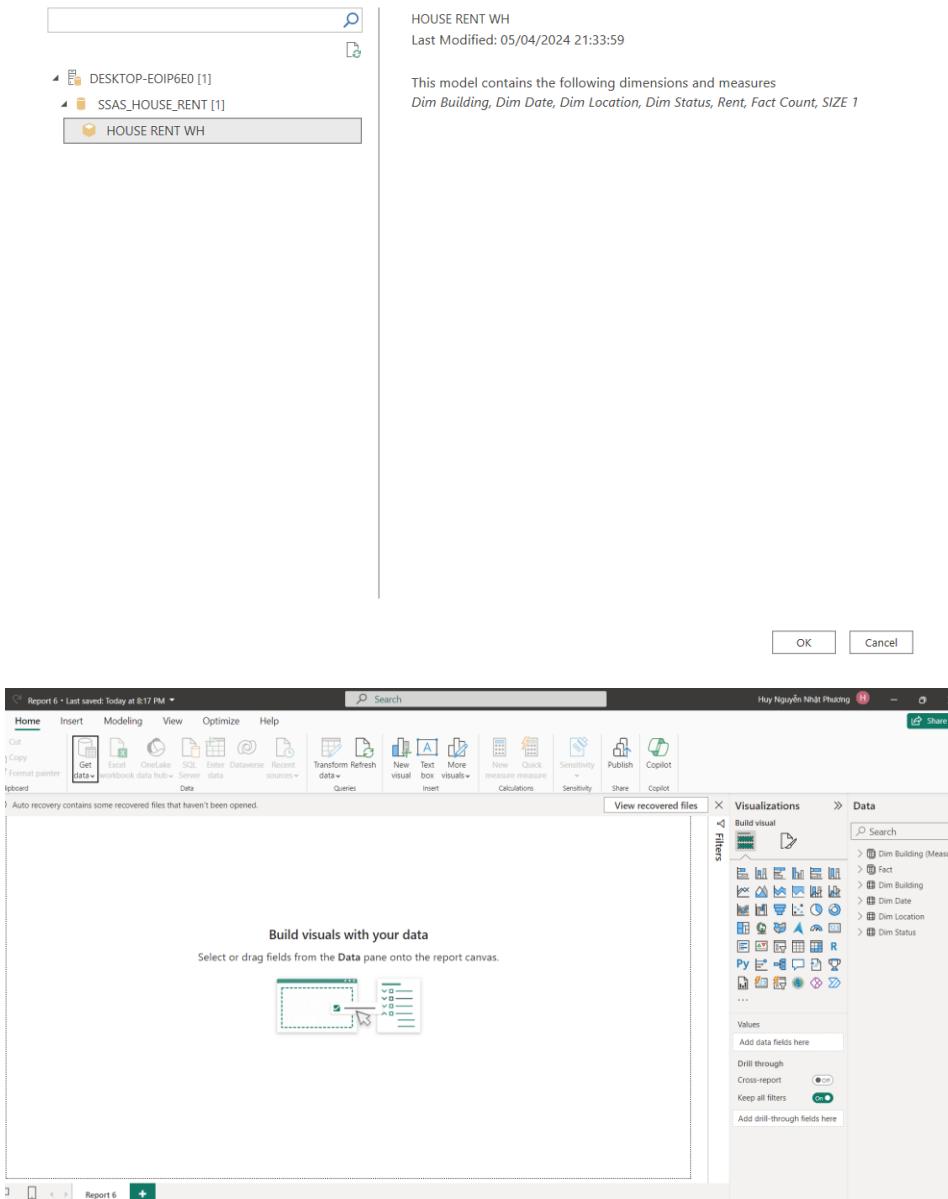


4.2.5. Report 8: Câu truy vấn 10: Cho biết 10 nhà có giá thuê cao nhất có “Furnished” ở thành phố “Chennai”.

- Đối với phần lấy dữ liệu của Report 6, ta chọn Get Data – Sql Server Analysis Services database. Nhập tên Server, chọn Connect Live và Nhấn OK.



- Chọn SSAS HOUSE RENT WH.



- Trong tab Data chọn Measure Rent, các thuộc tính cần thiết ở các bảng Dim: Building ID, City, Furnishing Status.

The screenshot shows the Power BI interface with the 'Visualizations' pane on the left and the 'Data' pane on the right.

Visualizations:

- Build visual: Includes icons for Bar, Line, Stacked Bar, Stacked Line, Heatmap, Map, Treemap, Gauge, Timeline, Card, Table, and Radar.
- Rows: Contains filters for 'Building ID' and 'Furnishing Status'.
- Columns: Contains a filter for 'City'.
- Values: Contains a filter for 'Rent'.
- Drill through: Shows 'Cross-report' with a 'Off' button.

Data:

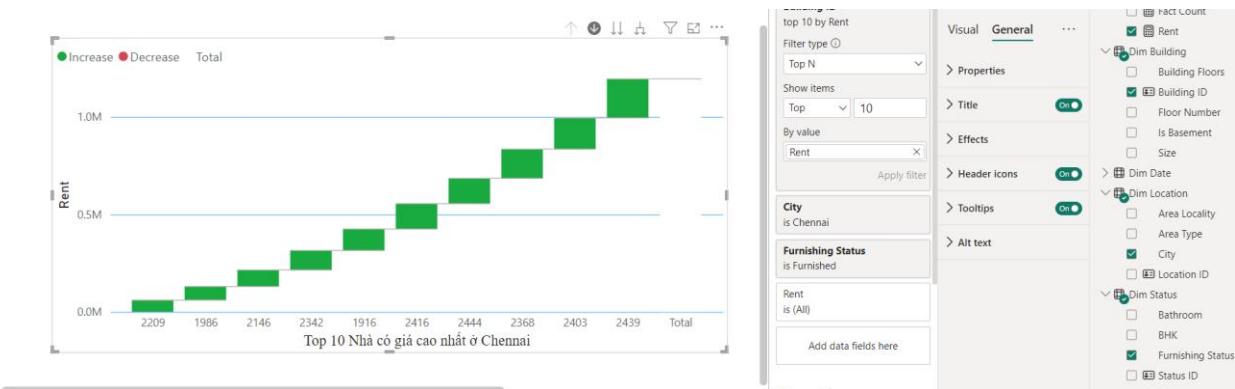
- Search bar: 'Search'.
- Fact: Contains 'Fact Count' (unchecked) and 'Rent' (checked).
- Dim Building: Contains 'Building Floors' (unchecked), 'Building ID' (checked), 'Floor Number' (unchecked), 'Is Basement' (unchecked), and 'Size' (unchecked).
- Dim Date: Contains 'Area Locality' (unchecked), 'Area Type' (unchecked), 'City' (checked), and 'Location ID' (unchecked).
- Dim Location: Contains 'Bathroom' (unchecked), 'BHK' (unchecked), 'Furnishing Status' (checked), and 'Status ID' (unchecked).

- Trong Filters, Building ID chọn Filter Type là Top N lọc theo Rent, chọn City là Chennai, Furnishing Status là Furnished.
- Tiếp theo trong Visualizations chọn Table.
- Thực hiện Format Visual.

The screenshot shows the Microsoft Power BI desktop interface. On the left, there is a table visualization titled "Building ID, City, Furnishing Status, Rent". The table lists 10 buildings in Chennai with furnished status and their rents. A "Total" row at the bottom shows a sum of 60000. To the right of the table are several filter panes and a visualizations pane.

- Filters pane:** Shows filters applied to the table, including "Building ID top 10 by Rent", "Top N", "Show items Top 10", "By value Rent", and specific filters for "City is Chennai", "Furnishing Status is Furnished", and "Rent is (All)".
- Visualizations pane:** Displays a list of available visualizations such as Waterfall chart, Bar chart, Line chart, etc.
- Data pane:** Shows the data source structure, including Fact, Dim Building, Dim Floors, Dim Location, Dim Date, and Dim Status dimensions.

- Chọn Waterfall chart, tiến hành tương tự và format được kết quả như hình sau:

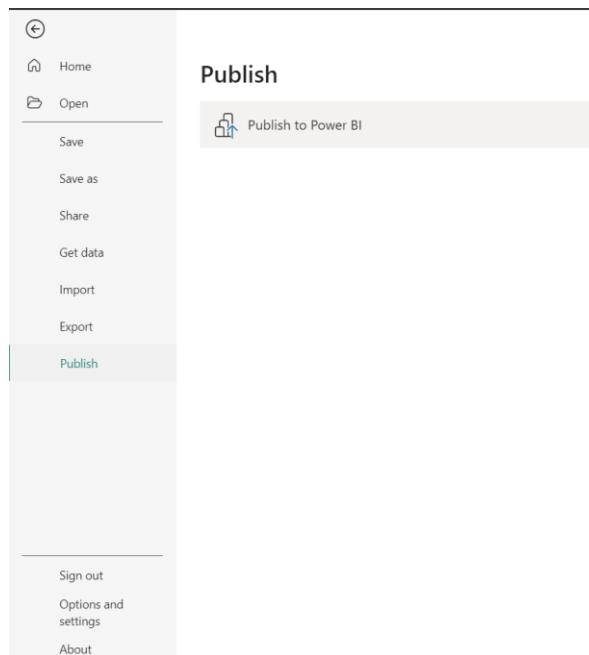


- Xuất báo cáo:



4.2.6. Triển khai SSRS trên Power BI

- Chọn File – Publish – Publish to Power BI để đưa report lên Server.



- Chọn My workspace, xác nhận nơi lưu trữ dữ liệu các report trên server.

Publish to Power BI

X

Select a destination

 Search

My workspace

Select

Cancel

- Sau khi public thành công, màn hình hiện thông báo như hình bên dưới. Chọn Open file.pbix để xem các report đã tạo.

Publishing to Power BI

X

 Success!

[Open 'Report 5.pbix' in Power BI](#)

[Get Quick Insights](#)

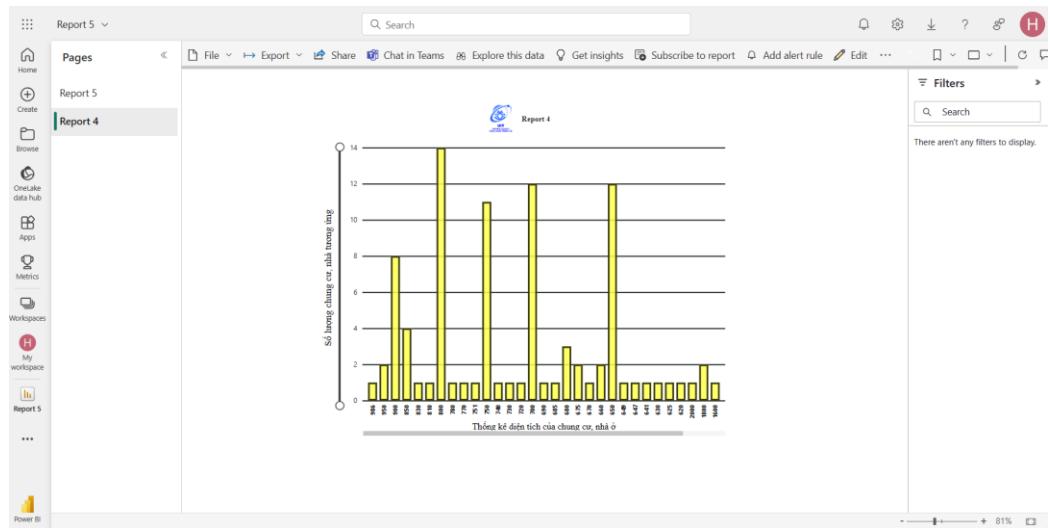


Did you know?

You can create a portrait view of your report, tailored for mobile phones.

On the **View** tab, select **Mobile Layout**. [Learn more](#)

Got it



CHƯƠNG 5. DATA MINING

5.1. Tiết xuât lý dữ liệu

- Import dữ liệu từ file .csv

```
[3]: df = pd.read_csv('House_Rent_Dataset.csv')
df.head()
```

| | Posted On | BHK | Rent | Size | Floor | Area Type | Area Locality | City | Furnishing Status | Tenant Preferred | Bathroom | Point of Contact |
|---|-----------|-----|-------|------|-----------------|-------------|--------------------------|---------|-------------------|------------------|----------|------------------|
| 0 | 5/18/2022 | 2 | 10000 | 1100 | Ground out of 2 | Super Area | Bandel | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner |
| 1 | 5/13/2022 | 2 | 20000 | 800 | 1 out of 3 | Super Area | Phool Bagan, Kankurgachi | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner |
| 2 | 5/16/2022 | 2 | 17000 | 1000 | 1 out of 3 | Super Area | Salt Lake City Sector 2 | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner |
| 3 | 7/4/2022 | 2 | 10000 | 800 | 1 out of 2 | Super Area | Dum Dum Park | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner |
| 4 | 5/9/2022 | 2 | 7500 | 850 | 1 out of 2 | Carpet Area | South Dum Dum | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner |

```
[4]: pd.set_option('display.max_rows', None)
pd.DataFrame(df.groupby('Floor')['Floor'].count())
pd.set_option('display.max_rows', 50)
```

- Bổ sung các thuộc tính is_basement, floor_number, building_floors được phân tách từ cột Floor

```
[5]: def feature_floor_split(r: pd.core.series.Series) -> Tuple[int, int]:
    if ' out of ' in r['Floor']:
        floor, max_floors = r['Floor'].split(' out of ')
        return [
            -1 if 'Basement' in floor else int(floor),
            int(max_floors)
        ]
    # This accounts for the values where instead of having a floor like "1 out of 3", it just has "1".
    return [int(r['Floor'].split(' out of ')[0]), None]

def feature_is_basement(r: pd.core.series.Series) -> str:
    if 'Basement' in r['Floor']:
        return 1
    return 0

# Replace all instances of "Ground" with 0
df['Floor'] = df['Floor'].str.replace('Ground', '0')

# Create `is_basement` + floor cols
df['is_basement'] = df.apply(lambda row: feature_is_basement(row), axis=1)
df['floor_number'] = df.apply(lambda row: feature_floor_split(row)[0], axis=1)
df['building_floors'] = df.apply(lambda row: feature_floor_split(row)[1], axis=1)

df.head()
```

| [5]: | Posted On | BHK | Rent | Size | Floor | Area Type | Area Locality | City | Furnishing Status | Tenant Preferred | Bathroom | Point of Contact | is_basement | floor_number | building_floors |
|------|-----------|-----|-------|------|------------|-------------|--------------------------|---------|-------------------|------------------|----------|------------------|-------------|--------------|-----------------|
| 0 | 5/18/2022 | 2 | 10000 | 1100 | 0 out of 2 | Super Area | Bandel | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 0 | 0 | 2.0 |
| 1 | 5/13/2022 | 2 | 20000 | 800 | 1 out of 3 | Super Area | Phool Bagan, Kankurgachi | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 0 | 1 | 3.0 |
| 2 | 5/16/2022 | 2 | 17000 | 1000 | 1 out of 3 | Super Area | Salt Lake City Sector 2 | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 0 | 1 | 3.0 |
| 3 | 7/4/2022 | 2 | 10000 | 800 | 1 out of 2 | Super Area | Dum Dum Park | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 0 | 1 | 2.0 |
| 4 | 5/9/2022 | 2 | 7500 | 850 | 1 out of 2 | Carpet Area | South Dum Dum | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner | 0 | 1 | 2.0 |

- Xóa các thuộc tính không cần thiết

```
[6]: for col in ['Tenant Preferred', 'Floor']:
    df.drop(col, inplace=True, axis=1)

df.head()
```

| [6]: | Posted On | BHK | Rent | Size | Area Type | Area Locality | City | Furnishing Status | Bathroom | Point of Contact | is_basement | floor_number | building_floors |
|------|-----------|-----|-------|------|-------------|--------------------------|---------|-------------------|----------|------------------|-------------|--------------|-----------------|
| 0 | 5/18/2022 | 2 | 10000 | 1100 | Super Area | Bandel | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 2.0 |
| 1 | 5/13/2022 | 2 | 20000 | 800 | Super Area | Phool Bagan, Kankurgachi | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 3.0 |
| 2 | 5/16/2022 | 2 | 17000 | 1000 | Super Area | Salt Lake City Sector 2 | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 3.0 |
| 3 | 7/4/2022 | 2 | 10000 | 800 | Super Area | Dum Dum Park | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 1 | 2.0 |
| 4 | 5/9/2022 | 2 | 7500 | 850 | Carpet Area | South Dum Dum | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 1 | 2.0 |

- Lưu Dataset đã được xử lý

```
[6]: df.to_excel(r'D:\Full_final_dataset.xlsx', index=False)

[7]: df
```

| [7]: | Posted On | BHK | Rent | Size | Area Type | Area Locality | City | Furnishing Status | Bathroom | Point of Contact | is_basement | floor_number | building_floors |
|------|-----------|-----|-------|------|-------------|--------------------------|-----------|-------------------|----------|------------------|-------------|--------------|-----------------|
| 0 | 5/18/2022 | 2 | 10000 | 1100 | Super Area | Bandel | Kolkata | Unfurnished | 2 | Contact Owner | 0 | 0 | 2.0 |
| 1 | 5/13/2022 | 2 | 20000 | 800 | Super Area | Phool Bagan, Kankurgachi | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 3.0 |
| 2 | 5/16/2022 | 2 | 17000 | 1000 | Super Area | Salt Lake City Sector 2 | Kolkata | Semi-Furnished | 1 | Contact Owner | 0 | 1 | 3.0 |
| 3 | 7/4/2022 | 2 | 10000 | 800 | Super Area | Dum Dum Park | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 1 | 2.0 |
| 4 | 5/9/2022 | 2 | 7500 | 850 | Carpet Area | South Dum Dum | Kolkata | Unfurnished | 1 | Contact Owner | 0 | 1 | 2.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 4741 | 5/18/2022 | 2 | 15000 | 1000 | Carpet Area | Bandam Kommu | Hyderabad | Semi-Furnished | 2 | Contact Owner | 0 | 3 | 5.0 |

5.2. Phân tích dữ liệu

5.2.1. Thống kê mô tả

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

```
[8]: numeric_cols = df.select_dtypes(include=['int', 'float']).columns
```

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

```
[9]: stats = df[numeric_cols].describe().T
```

- Tính toán thêm các giá trị mode, range, and variance cho tất cả các cột có kiểu dữ liệu là Int hoặc Float

```
[12]: for col in df[numeric_cols].columns:
    stats.loc[col, 'mode'] = df[col].mode()[0]
for col in df[numeric_cols].columns:
    stats.loc[col, 'range'] = df[col].max() - df[col].min()
for col in df[numeric_cols].columns:
    stats.loc[col, 'variance'] = df[col].var()
display(stats)
```

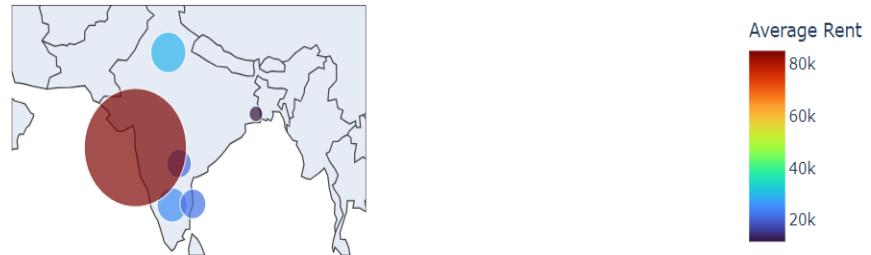
- Xem kết quả thống kê dữ liệu

| | count | mean | std | min | 25% | 50% | 75% | max | mode | range | variance |
|------------------------|--------|--------------|--------------|--------|---------|---------|---------|-----------|---------|-----------|--------------|
| BHK | 4746.0 | 2.083860 | 0.832256 | 1.0 | 2.0 | 2.0 | 3.0 | 6.0 | 2.0 | 5.0 | 6.926499e-01 |
| Rent | 4746.0 | 34993.451327 | 78106.412937 | 1200.0 | 10000.0 | 16000.0 | 33000.0 | 3500000.0 | 15000.0 | 3498800.0 | 6.100612e+09 |
| Size | 4746.0 | 967.490729 | 634.202328 | 10.0 | 550.0 | 850.0 | 1200.0 | 8000.0 | 1000.0 | 7990.0 | 4.022126e+05 |
| Bathroom | 4746.0 | 1.965866 | 0.884532 | 1.0 | 1.0 | 2.0 | 2.0 | 10.0 | 2.0 | 9.0 | 7.823963e-01 |
| is_basement | 4746.0 | 0.007164 | 0.084345 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 7.114105e-03 |
| floor_number | 4746.0 | 3.438475 | 5.771967 | -1.0 | 1.0 | 2.0 | 3.0 | 76.0 | 1.0 | 77.0 | 3.331560e+01 |
| building_floors | 4742.0 | 6.973429 | 9.469727 | 1.0 | 2.0 | 4.0 | 6.0 | 89.0 | 4.0 | 88.0 | 8.967573e+01 |

5.2.2. Trực quan hóa dữ liệu

- Trực quan hóa giá nhà trung bình mở mỗi thành phố

Average Rent in Indian Cities

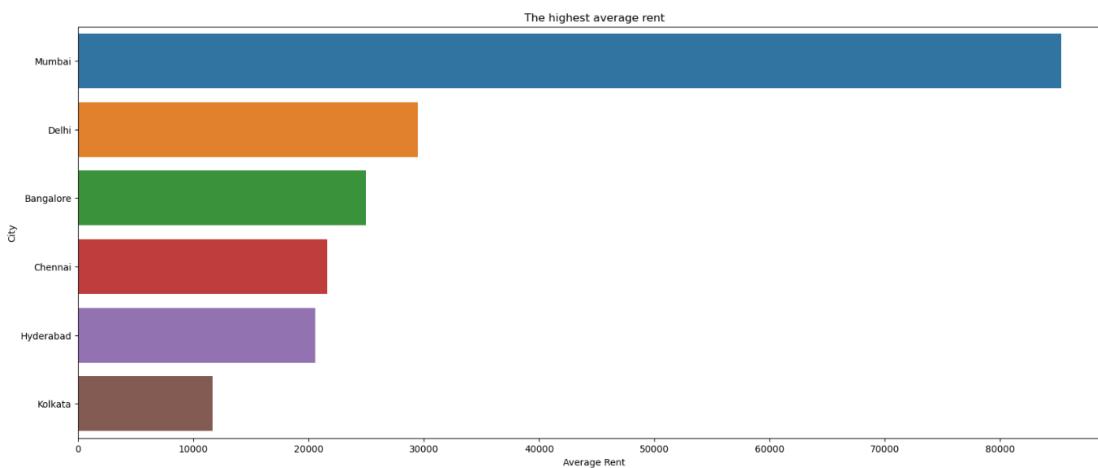


Nhìn vào bản đồ, ta có thể thấy được hầu hết các thành phố đều có màu xanh đậm đến nhạt (biểu thị giá nhà trung bình thấp), chỉ có một thành phố có màu đỏ (biểu thị giá nhà trung bình cao).

- Xếp hạng trung bình giá thuê theo từng thành phố

```
[22]: import matplotlib.pyplot as plt
import seaborn as sns

top_cities = city_rent_avg.sort_values(by='Rent', ascending=False).head(10)
plt.figure(figsize=(20, 8))
plt.title("The highest average rent")
sns.barplot(x=top_cities['Rent'], y=top_cities['City'], orient="h")
plt.xlabel("Average Rent")
plt.ylabel("City")
plt.show()
```

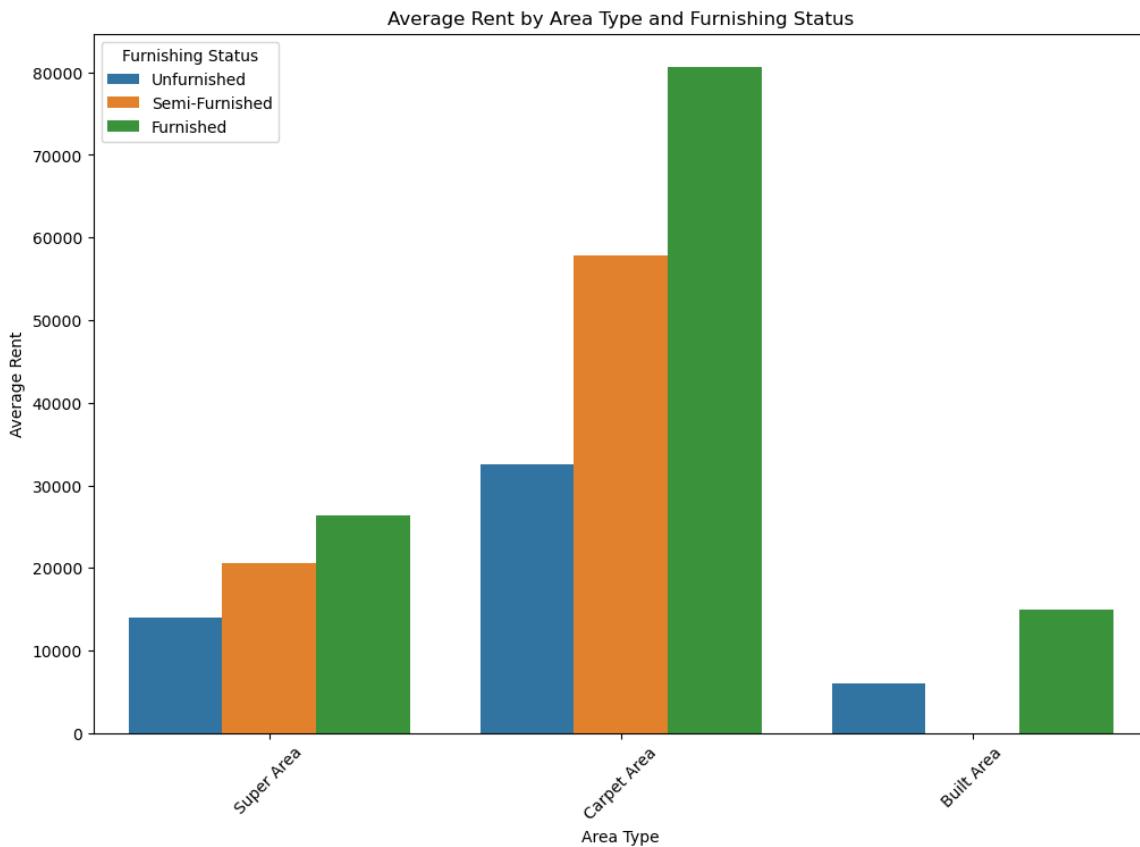


Quan sát từ biểu đồ trên, ta thấy Mumbai là thành phố có giá thuê nhà trung bình cao nhất trên 80000. Hầu hết các thành phố còn lại đều có giá thuê trung bình dưới 30000.

- Phân tích giá thuê trung bình dựa trên tình trạng “Furnishing Status” và “Area Type”

```
[16]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 8))
sns.barplot(data=df, x='Area Type', y='Rent', hue='Furnishing Status', errorbar=None)
plt.title('Average Rent by Area Type and Furnishing Status')
plt.xlabel('Area Type')
plt.ylabel('Average Rent')
plt.xticks(rotation=45)
plt.show()
```



Từ biểu đồ trên, ta có thể thấy được rằng giá thuê trung bình ở Carpet Area cao hơn hẳn Super Area và Built Area bất kể tình trạng nội thất như thế nào.

5.3. Ứng dụng mô hình thuật toán khai thác dữ liệu

Khai báo 6 từ điển (dictionary) trống, mỗi từ điển có tên là accuracy, precision, recall, f1, fpr và tpr.

- accuracy: Lưu trữ kết quả độ chính xác (accuracy) của mô hình trên từng lớp.
- precision: Lưu trữ kết quả độ chính xác dự báo (precision) mô hình trên từng lớp.

- recall: Lưu trữ kết quả độ phủ sóng (recall) của mô hình trên từng lớp.
- f1: Lưu trữ kết quả độ F1 (F1-score) của mô hình trên từng lớp.

Mỗi từ điển sẽ được sử dụng để lưu kết quả đánh giá cho một lớp (class) trong bài toán phân loại đa lớp (multi-class classification). Các giá trị sẽ được cập nhật trong quá trình đánh giá mô hình.

```
[134]: accuracy = dict()
precision = dict()
recall = dict()
f1 = dict()
```

5.3.1. Tiền xử lý dữ liệu trước khi áp dụng các mô hình khai thác dữ liệu

- Lựa chọn các thuộc tính

```
[21]: # Tạo DataFrame mới chỉ chứa các thuộc tính cần thiết
X = df[['BHK', 'Rent', 'Size', 'Bathroom', 'is_basement', 'floor_number', 'building_floors']]
```

- Xử lý các giá trị thiếu trong DataFrame X

```
print("Các lớp duy nhất cho từng thuộc tính phân loại: ")
for cat in X:
    print("{:15s}".format(cat), "\t", len(X[cat].unique()))
```

Các lớp duy nhất cho từng thuộc tính phân loại:

| | |
|-----------------|-----|
| BHK | 6 |
| Rent | 243 |
| Size | 615 |
| Bathroom | 8 |
| is_basement | 2 |
| floor_number | 53 |
| building_floors | 67 |

```
unique_values = df['BHK'].unique()
print("Unique values in 'column1':", unique_values)
```

Unique values in 'column1': [2 1 3 6 4 5]

```

print(X.isna().sum())

BHK          0
Rent         0
Size         0
Bathroom     0
is_basement  0
floor_number 0
building_floors 4
dtype: int64

# Xác định giá trị phổ biến nhất trong cột 'building_floors'
mode_building_floors = X['building_floors'].mode()[0]

# Thay thế các giá trị thiếu bằng giá trị phổ biến nhất
X['building_floors'].fillna(mode_building_floors, inplace=True)

```

- Chuẩn hóa dữ liệu

```
[31]: scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
```

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

Chia DataFrame X thành 2 phần: tập huấn luyện (X) và tập kiểm tra (X_test)

```

# Train/Validation - Test split
X, X_test = train_test_split(X, test_size=.2, random_state=42)
print(X.shape, X_test.shape)

(3796, 7) (950, 7)

```

- Tham số test_size bằng 0.2, có nghĩa 20% dữ liệu được dùng để kiểm tra và 80% dữ liệu còn lại được sử dụng để huấn luyện và đánh giá.

Gán DataFrame X vào một biến mới là sample.

```

sample = X
y_sample = sample["BHK"]
X_sample = sample.drop("BHK", axis=1)

X_train, X_validate, y_train, y_validate = train_test_split(X_sample, y_sample, random_state=42)
print(X_train.shape, y_train.shape)
print(X_validate.shape, y_validate.shape)

(2847, 6) (2847,)
(949, 6) (949,)

```

- Trích xuất thuộc tính mục tiêu Rent từ sample và gán cho y_sample, còn lại các thuộc tính được gán cho X_sample.
- Sử dụng hàm train_test_split() để chia X_sample và y_sample thành tập huấn luyện và tập đánh giá. Thiết lập tham số random_state bằng 42 để đảm bảo việc chia là có thể tái lập.
- Cuối cùng, in ra kích thước của hai tập kết quả sử dụng thuộc tính shape của mỗi DataFrame hoặc Series.

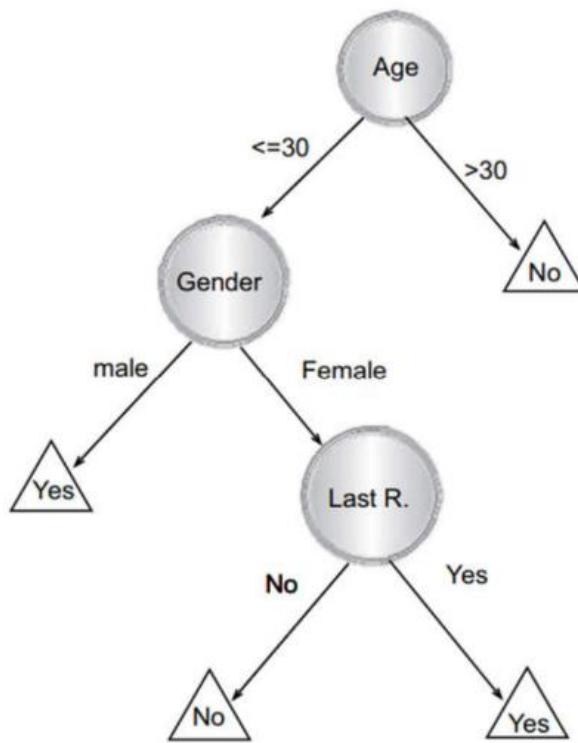
5.3.3. Decision Tree

5.3.3.1. Giới thiệu thuật toán Decision Tree

Cây quyết định là một mô hình phân loại được biểu diễn dưới dạng phân chia đệ quy của không gian các ví dụ. Cây quyết định bao gồm các nút, trong đó có một nút gốc không có cạnh nào, các nút nội có đúng một cạnh vào và thực hiện kiểm tra trên các thuộc tính, và các nút lá (nút cuối) đại diện kết quả phân loại. Tại các nút nội, không gian ví dụ được phân chia thành các không gian con dựa trên các giá trị của một hoặc nhiều thuộc tính.

Trong trường hợp đơn giản, mỗi kiểm tra chỉ xét một thuộc tính. Các ví dụ được phân loại bằng cách điều hướng từ nút gốc xuống các nút lá, theo kết quả của các kiểm tra trên đường đi. Mỗi nút lá được gán cho một lớp đại diện cho giá trị mục tiêu phù hợp nhất, hoặc chứa một vector xác suất cho các giá trị mục tiêu.

Ví dụ, hình ảnh này mô tả một cây quyết định dùng để suy luận xem một khách hàng tiềm năng có phản hồi với thư email trực tiếp hay không. Các nút nội được biểu diễn dưới dạng các hình tròn, trong khi các lá được ký hiệu là các tam giác.



5.3.3.2. Lý do chọn thuật toán

Một số lợi ích của thuật toán Cây quyết định (Decision tree):

- Dễ hiểu và dễ giải thích. Cây có thể được trực quan hóa.
- Có khả năng xử lý cả dữ liệu số và dữ liệu phân loại.
- Có khả năng xử lý dữ liệu không cân bằng (imbalanced data). Điều này có ý nghĩa rằng cây quyết định không bị ảnh hưởng nghiêm trọng bởi sự chênh lệch về tỷ lệ các lớp trong dữ liệu đào tạo.
- Tuân thủ nguyên tắc Occam's Razor, tức là ưu tiên chọn mô hình đơn giản hơn trong trường hợp có nhiều mô hình tương đồng. Điều này giúp tránh overfitting và tạo ra các mô hình khái quát hóa tốt.
- Khả năng xử lý dữ liệu có tính chất phi tuyến (non-linear). Cây quyết định có thể tạo ra các quyết định phân chia dựa trên các quan hệ phi tuyến giữa thuộc tính và mục tiêu.
- Khả năng xử lý các giá trị ngoại lệ (outlier values). Điều này cho phép cây quyết định duy trì tính ổn định và hiệu quả trong việc phân loại dữ liệu, ngay cả khi có một số giá trị ngoại lệ xuất hiện trong tập dữ liệu.

5.3.3.3. Xây dựng và dự đoán mô hình Decision Tree

- Tạo mô hình cây quyết định (dtc) sử dụng DecisionTreeClassifier từ sklearn.tree, với tham số Random_state được đặt thành 42.
- Tạo danh sách các tham số để thử trong tìm kiếm dạng lưới. Danh sách này bao gồm hai tham số: criterion (gồm hai phương pháp gini index và entropy đo độ tạp chất của nút) và max_depth (độ sâu tối đa của cây).
- Tạo đối tượng tìm kiếm dạng lưới với mô hình cây quyết định ‘dtc’, các tham số để thử ‘parameters’, ‘verbose = 5’ để in kết quả tìm kiếm dạng lưới và n_jobs = -1 để sử dụng tất cả các lõi CPU có sẵn.
- Huấn luyện mô hình trên tập X_train và y_train, đồng thời thực hiện tìm kiếm dạng lưới để tìm các tham số tốt nhất cho mô hình.

```

dtc = DecisionTreeClassifier(random_state=42)
parameters = [{"criterion": ["gini", "entropy"], "max_depth": [5, 10, 15, 30]}]
grid = GridSearchCV(dtc, parameters, verbose=5, n_jobs=-1)
grid.fit(X_train, y_train)

print("Best parameters scores:")
print(grid.best_params_)
print("Train score:", grid.score(X_train, y_train))
print("Validation score:", grid.score(X_validate, y_validate))

Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best parameters scores:
{'criterion': 'gini', 'max_depth': 5}
Train score: 0.821917808219178
Validation score: 0.8229715489989463

```

- Đầu ra có nghĩa là mô hình Cây quyết định đã được đào tạo và điểm số tham số tốt nhất là “criterion: gini” và “max_depth: 5”.
- Điểm chính xác trên dữ liệu huấn luyện (train data) xấp xỉ 0.821, cho thấy mô hình đã dự đoán chính xác nhãn lớp cho 82,1% mẫu huấn luyện.
- Điểm chính xác trên dữ liệu kiểm định (validation data) là xấp xỉ 0.822, cho biết mô hình đã dự đoán chính xác nhãn lớp cho 82,2% tập kiểm định.

Khớp mô hình (Fit model) Cây quyết định (dtc) với tập dữ liệu huấn luyện (X_train và y_train) nhằm tìm tham số có khả năng dự đoán tốt trên dữ liệu mới và in ra điểm đánh giá hiệu suất mô hình.

```

print("Default scores:")
dtc.fit(X_train, y_train)
print("Train score:", dtc.score(X_train, y_train))
print("Validation score:", dtc.score(X_validate, y_validate))

```

```

Default scores:
Train score: 0.9859501229364243
Validation score: 0.743940990516333

```

- Điểm huấn luyện xác xỉ 0.98 cho thấy mô hình dự đoán đúng 98% mẫu huấn luyện.
- Điểm kiểm chứng xác xỉ 0.74 cho thấy mô hình đã dự đoán đúng 74 % số mẫu kiểm chứng.

Tạo DataFrame từ kết quả tìm kiếm dạng lưới (grid search) và sắp xếp kết quả theo thứ tự tăng dần của điểm xác thực.

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | param_criterion | param_max_depth | params | split0_test_score | split1_test_score | split2_test_score | split3_test_score |
|---|---------------|--------------|-----------------|----------------|-----------------|-----------------|---|-------------------|-------------------|-------------------|-------------------|
| 0 | 0.008818 | 0.001994 | 0.003827 | 0.001000 | gini | 5 | {"criterion": "gini", "max_depth": 5} | 0.807018 | 0.759649 | 0.818981 | 0.818981 |
| 4 | 0.008503 | 0.001382 | 0.002746 | 0.000912 | entropy | 5 | {"criterion": "entropy", "max_depth": 5} | 0.792982 | 0.757895 | 0.811951 | 0.811951 |
| 1 | 0.013989 | 0.002785 | 0.005127 | 0.003180 | gini | 10 | {"criterion": "gini", "max_depth": 10} | 0.773684 | 0.745614 | 0.801406 | 0.801406 |
| 5 | 0.011481 | 0.001160 | 0.003207 | 0.001347 | entropy | 10 | {"criterion": "entropy", "max_depth": 10} | 0.754386 | 0.733333 | 0.787346 | 0.787346 |
| 7 | 0.016011 | 0.004345 | 0.002360 | 0.000762 | entropy | 30 | {"criterion": "entropy", "max_depth": 30} | 0.715789 | 0.728070 | 0.750439 | 0.750439 |
| 6 | 0.014480 | 0.004502 | 0.001400 | 0.001148 | entropy | 15 | {"criterion": "entropy", "max_depth": 15} | 0.724561 | 0.712281 | 0.743409 | 0.743409 |
| 2 | 0.011546 | 0.003043 | 0.003863 | 0.004697 | gini | 15 | {"criterion": "gini", "max_depth": 15} | 0.726316 | 0.722807 | 0.739895 | 0.739895 |

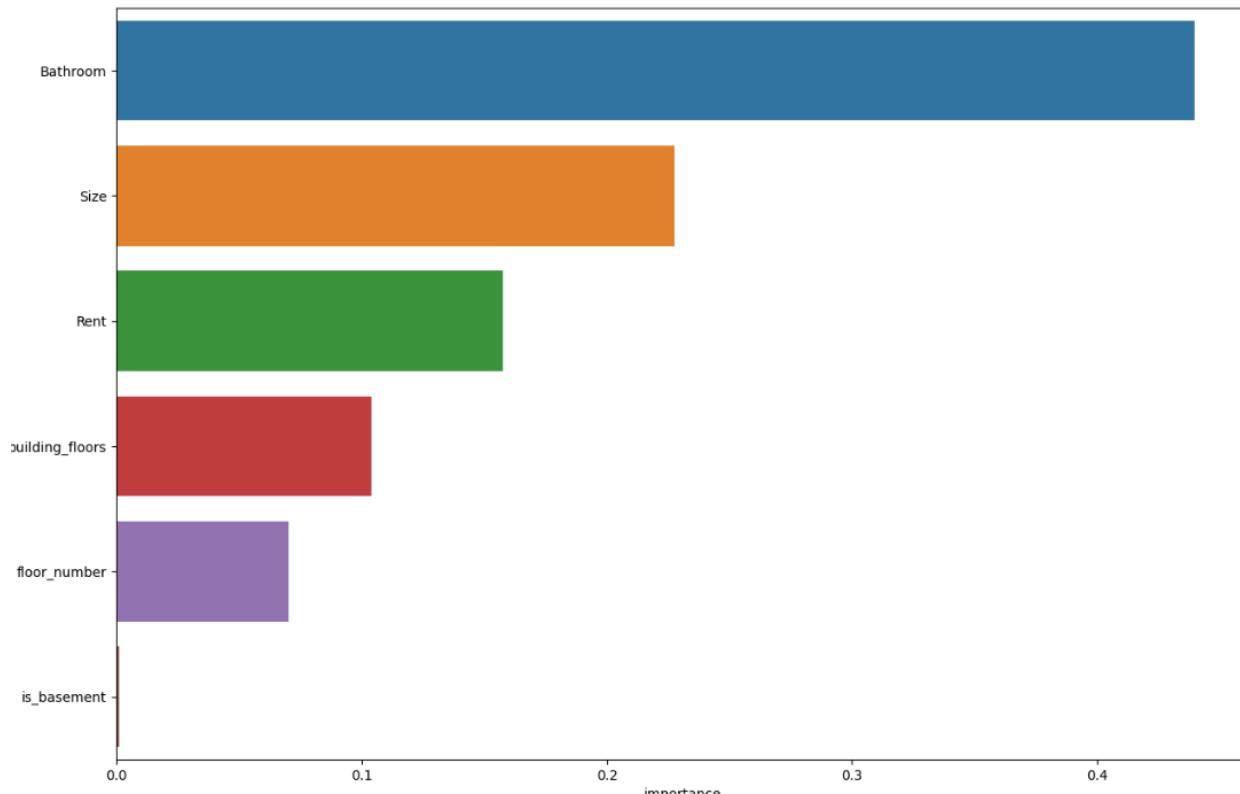
Tìm ra 30 tính năng quan trọng hàng đầu bằng cách tạo DataFrame có số hàng bằng số cột trong X_train và một cột có tên là “quan trọng”, sử dụng tên thuộc tính trong X_train làm chỉ mục.

```
[115]: importances = pd.DataFrame(np.zeros((X_train.shape[1], 1)), columns=["importance"], index=X_train.columns)

importances.iloc[:,0] = dtc.feature_importances_

importances = importances.sort_values(by="importance", ascending=False)[:30]

plt.figure(figsize=(15, 10))
sns.barplot(x="importance", y=importances.index, data=importances)
plt.show()
```

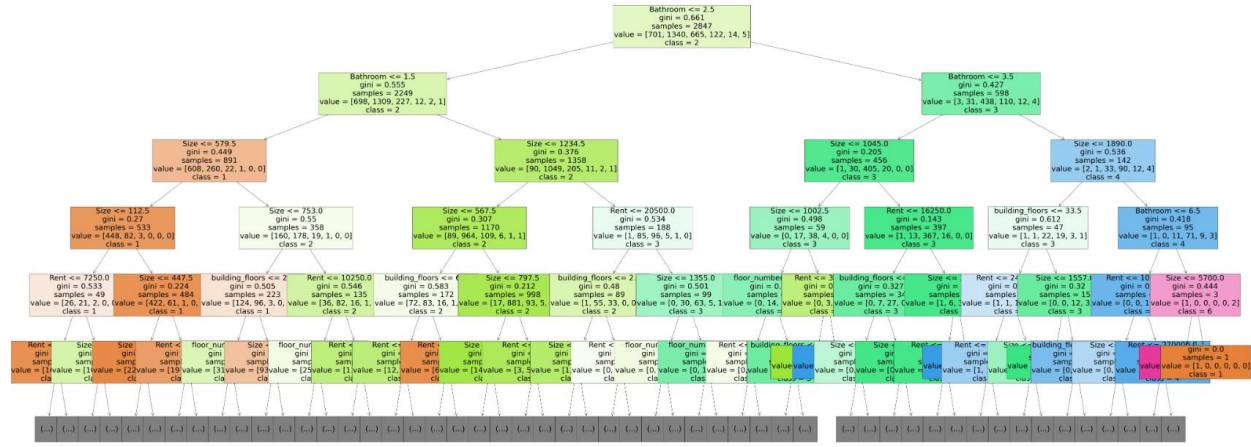


Tạo trực quan hóa mô hình Cây quyết định ‘dtc’ bằng cách sử dụng hàm plot_tree() từ sklearn.tree, với max_depth = 5.

```
fig, ax = plt.subplots(figsize=(100, 40))
plot_tree(dtc, max_depth=5, fontsize=15, feature_names=X_train.columns.to_list(), class_names=[str(cls) for cls in dtc.classes_], filled=True)
plt.savefig('decision_tree.png', format='png', bbox_inches='tight')
plt.show()

classes = [str(cls) for cls in dtc.classes_]
colors = [plt.cm.tab10(i / len(classes)) for i in range(len(classes))]

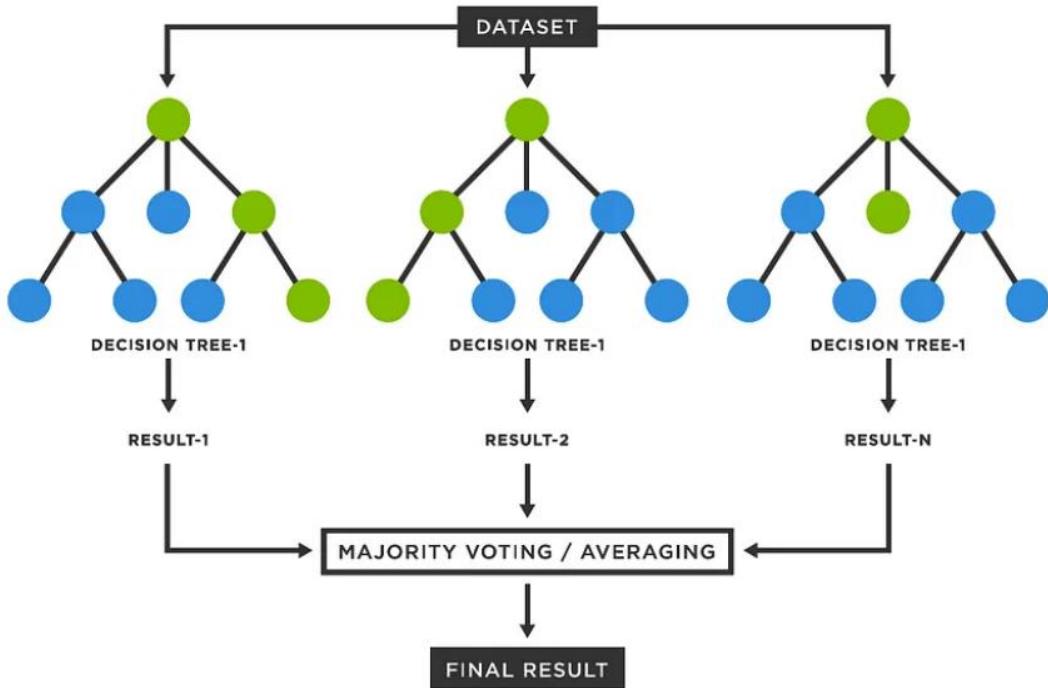
# Tạo chú giải
handles = [mpatches.Patch(color=colors[i], label=f'Class {classes[i]}') for i in range(len(classes))]
plt.legend(handles, bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



5.3.4. Random Forest

5.3.4.1. Giới thiệu thuật toán Random Forest

Nhằm khắc phục tình trạng overfitting của mô hình cây quyết định, mô hình Random forest được giới thiệu lần đầu bởi Breiman (2001) [2], tác giả đã đề xuất tạo một tập hợp các cây quyết định được xây dựng dựa trên một tập con ngẫu nhiên của các mẫu dữ liệu với các thuộc tính quyết định cũng được lựa chọn ngẫu nhiên. Dữ liệu của mỗi cây được lấy ngẫu nhiên và có thể trùng lặp(Bootstrapping) để đếm lại các kết quả dự báo từ bộ dữ liệu huấn luyện độc lập nhau. Từ đó, kết quả dự báo là giá trị trung bình của tập hợp cây quyết định.



5.3.4.2. Lý do chọn thuật toán

- Bởi vì đây là một thuật toán phân loại dữ liệu, nên nó phù hợp cho vấn đề được đặt ra.
- Thuật toán cung cấp độ chính xác cao và thời gian chạy nhanh.

5.3.4.3. Xây dựng và dự đoán với mô hình Random Forest

- Thực hiện bộ phân loại Random Forest: Đoạn mã khởi tạo một bộ phân loại Random Forest (rfc) bằng cách sử dụng lớp Random Forest Classifier.
- Đôi tượng bộ phân loại Random Forest được khởi tạo với `n_jobs` và `random_state`.
- Các siêu tham số cần được điều chỉnh được chỉ định trong một từ điển được gọi là `parameters`.
- Một đôi tượng GridSearchCV được tạo ra và phù hợp với dữ liệu huấn luyện.
- Các siêu tham số tốt nhất được xuất ra cùng với các điểm số độ chính xác của mô hình trên dữ liệu huấn luyện và dữ liệu xác thực.

```

rfc = RandomForestClassifier(n_jobs=1, random_state=42)
parameters = [{"n_estimators": [50, 100, 200, 500], "max_depth": [5, 10, 15, 30]}]
grid = GridSearchCV(rfc, parameters, verbose=5, n_jobs=1)
grid.fit(X_train, y_train)

print("Best parameters scores:")
print(grid.best_params_)
print("Train score:", grid.score(X_train, y_train))
print("Validation score:", grid.score(X_validate, y_validate))

Fitting 5 folds for each of 16 candidates, totalling 80 fits
Best parameters scores:
{'max_depth': 10, 'n_estimators': 200}
Train score: 0.898840885142255
Validation score: 0.827186512118019

```

- Kết quả đầu ra cho thấy rằng mô hình bộ phân loại Random Forest đã được huấn luyện và đánh giá bằng cách sử dụng GridSearchCV với 5-fold cross-validation trên 2 tố hợp siêu tham số (n_estimators, max_depth).
- Điểm số độ chính xác trên dữ liệu huấn luyện là 0,898, cho thấy rằng mô hình đã dự đoán đúng nhãn lớp cho 89.8% các ví dụ trong tập huấn luyện.
- Điểm số độ chính xác trên dữ liệu xác thực là 0.8271, cho thấy rằng mô hình đã dự đoán đúng nhãn lớp cho 82,71% các ví dụ trong tập xác thực.

Sử dụng mô hình bộ phân loại Random Forest (rfc) để phù hợp với tập dữ liệu huấn luyện (`X_train` và `y_train`) và in ra các điểm số huấn luyện và xác thực với các siêu tham số mặc định.

```

print("Default scores:")
rfc.fit(X_train, y_train)
print("Train score:", rfc.score(X_train, y_train))
print("Validation score:", rfc.score(X_validate, y_validate))

Default scores:
Train score: 0.9859501229364243
Validation score: 0.8008429926238145

```

- Điểm số huấn luyện là 0,98 cho thấy rằng mô hình đã dự đoán đúng 98% các mẫu huấn luyện.
- Điểm số xác thực là 0,8 cho thấy rằng mô hình đã dự đoán đúng 80% các mẫu xác thực.

Tạo một DataFrame Pandas chứa kết quả của cross-validation lưới tìm kiếm và sắp xếp theo rank_test_score.

| | mean_fit_time | std_fit_time | mean_score_time | std_score_time | param_max_depth | param_n_estimators | params | split0_test_score | split1_test_score | split2_test_score |
|---|---------------|--------------|-----------------|----------------|-----------------|--------------------|--|-------------------|-------------------|-------------------|
| 6 | 0.885550 | 0.099001 | 0.112666 | 0.025270 | 10 | 200 | {'max_depth': 10, 'n_estimators': 200} | 0.812281 | 0.764912 | 0.8295 |
| 7 | 2.284733 | 0.308644 | 0.219827 | 0.027349 | 10 | 500 | {'max_depth': 10, 'n_estimators': 500} | 0.815789 | 0.757895 | 0.8330 |
| 5 | 0.613435 | 0.177724 | 0.098418 | 0.031521 | 10 | 100 | {'max_depth': 10, 'n_estimators': 100} | 0.810526 | 0.759649 | 0.8330 |
| 1 | 0.355216 | 0.049651 | 0.041277 | 0.005419 | 5 | 100 | {'max_depth': 5, 'n_estimators': 100} | 0.805263 | 0.761404 | 0.8312 |
| 4 | 0.157161 | 0.019788 | 0.037829 | 0.008357 | 10 | 50 | {'max_depth': 10, 'n_estimators': 50} | 0.815789 | 0.759649 | 0.8260 |
| 2 | 0.702082 | 0.038550 | 0.081804 | 0.019439 | 5 | 200 | {'max_depth': 5, 'n_estimators': 200} | 0.805263 | 0.763158 | 0.8295 |

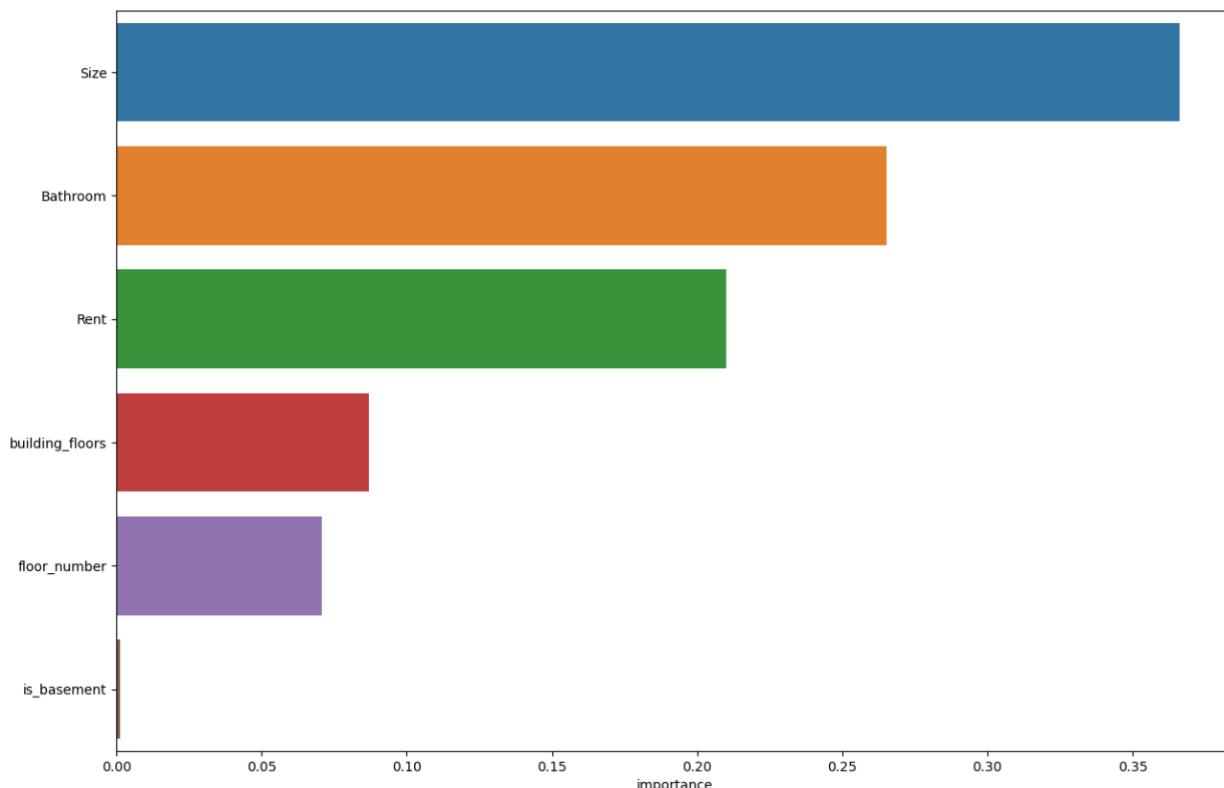
Tạo biểu đồ cột để đại diện cho độ quan trọng của tính năng, được xác định bởi bộ phân loại Random Forest cho 30 tính năng hàng đầu. Điểm số độ quan trọng của một tính năng càng cao thì tính năng đó càng ảnh hưởng đến việc dự đoán với mô hình Random Forest.

```
[90]: importances = pd.DataFrame(np.zeros((X_train.shape[1], 1)), columns=["importance"], index=X_train.columns)

importances.iloc[:,0] = rfc.feature_importances_

importances = importances.sort_values(by="importance", ascending=False)[:30]

plt.figure(figsize=(15, 10))
sns.barplot(x="importance", y=importances.index, data=importances)
plt.show()
```



5.3.5. Gaussian Naïve Bayes

5.3.5.1. Giới thiệu thuật toán Gaussian Naïve Bayes

Gaussian Naive Bayes (GNB) là một biến thể của thuật toán phân loại Naive Bayes, được sử dụng khi các đặc trưng đầu vào có phân phối Gaussian (chuẩn).

Trong Naive Bayes cổ điển, giả định rằng các đặc trưng đều độc lập và có phân phối rời rạc (ví dụ: Bernoulli, Multinomial). Tuy nhiên, khi các đặc trưng có phân phối liên tục như Gaussian, GNB là một lựa chọn phù hợp hơn.

Cụ thể, GNB giả định rằng các đặc trưng có phân phối Gaussian với trung bình và phương sai khác nhau cho mỗi lớp. Quá trình phân loại sẽ dựa vào việc tính toán xác suất của mẫu dữ liệu với từng lớp, dựa trên những thông số thống kê (trung bình và phương sai) của phân phối Gaussian.

5.3.5.2. Lý do chọn thuật toán

- Xử lý đặc trưng liên tục có thể làm việc hiệu quả với các đặc trưng có phân phối Gaussian (chuẩn), không giới hạn ở các đặc trưng rời rạc như Naive Bayes cổ điển. Giúp mở rộng phạm vi ứng dụng so với các thuật toán phân loại khác.

- Tính hiệu quả cao nhờ có cấu trúc đơn giản và yêu cầu tính toán ít, do đó có tốc độ xử lý nhanh. Rất hữu ích khi làm việc với dữ liệu lớn hoặc trong các ứng dụng yêu cầu phản hồi nhanh.
- Khả năng xử lý đa lớp có thể áp dụng cho bài toán phân loại đa lớp một cách tự nhiên.
- Khả năng kết hợp với các kỹ thuật khác như chọn đặc trưng, cải thiện dữ liệu,... để nhằm tăng cường hiệu suất.

5.3.5.3. Xây dựng và dự đoán với mô hình Gaussian Naïve Bayes

```
from datetime import timedelta
import time
gnb_start_time = time.time()
gnb = GaussianNB()
gnb.fit(X_train, y_train)
gnb_end_time = time.time()
gnb_time = gnb_end_time - gnb_start_time
print("Train score:", gnb.score(X_train, y_train))
print("Validation score:", gnb.score(X_validate, y_validate))
print("Time is: ", timedelta(seconds=round(gnb_time,4)))
```

Train score: 0.6111696522655427
 Validation score: 0.6322444678609063
 Time is: 0:00:00.012800

- Điểm số huấn luyện là 0,611 cho thấy rằng mô hình đã dự đoán đúng 61,1% các mẫu huấn luyện.
- Điểm số xác thực là 0,632 cho thấy rằng mô hình đã dự đoán đúng 63,22% các mẫu xác thực.

5.3.6. Bernoulli Naïve Bayes

5.3.6.1. Giới thiệu thuật toán Bernoulli Naïve Bayes

Bernoulli Naive Bayes (BNB) là một biến thể của thuật toán Naive Bayes, được sử dụng khi các đặc trưng của dữ liệu là nhị phân (binary) hoặc định danh (categorical).

Trong BNB, mỗi đặc trưng được coi là một biến ngẫu nhiên độc lập theo phân phối Bernoulli, với giá trị 0 hoặc 1. Thay vì sử dụng phân phối Gaussian như trong Gaussian Naive Bayes, BNB mô hình hóa xác suất xuất hiện của mỗi đặc trưng trong từng lớp.

Cụ thể, BNB tính toán xác suất có điều kiện của mỗi đặc trưng nhị phân, dựa trên tần suất xuất hiện của chúng trong tập dữ liệu huấn luyện. Sau đó, nó sử dụng định lý Bayes để tính xác suất của mỗi lớp dự đoán, dựa trên tích của các xác suất có điều kiện này.

5.3.6.2. Lý do chọn thuật toán

- Yêu cầu ít dữ liệu huấn luyện so với một số thuật toán phức tạp khác.
- Đơn giản và dễ hiểu, giúp người dùng có thể hiểu được cách nó đưa ra dự đoán.
- Hiệu suất tốt cho kết quả phân loại tốt, đặc biệt là trên các tập dữ liệu nhỏ hoặc đặc trưng ít. Có thể đặt độ chính xác cao, đặc biệt khi các đặc trưng được coi là độc lập.

5.3.6.3. Xây dựng và dự đoán với mô hình Bernoulli Naïve Bayes

```
bnb_start_time = time.time()
bnb = BernoulliNB()
bnb.fit(X_train, y_train)
bnb_end_time = time.time()
bnb_time = bnb_end_time - bnb_start_time
print("Train score:", bnb.score(X_train, y_train))
print("Validation score:", bnb.score(X_validate, y_validate))
print("Time is: ", timedelta(seconds=round(bnb_time,4)))
```

```
Train score: 0.47067088162978576
Validation score: 0.48472075869336145
Time is: 0:00:00.034900
```

- Điểm số huấn luyện là 0,47 cho thấy rằng mô hình đã dự đoán đúng 47% các mẫu huấn luyện.
- Điểm số xác thực là 0,48 cho thấy rằng mô hình đã dự đoán đúng 48% các mẫu xác thực.

5.4. Đánh giá thuật toán và dự báo

5.4.1. Các độ đo dùng để đánh giá thuật toán

Đánh giá mô hình phân loại trong máy học, ta cần biết được các chỉ số: TP, FP, TN, FN.

- TP (True Positive): Tổng số trường hợp dự báo khớp Positive.
- TN (True Negative): Tổng số trường hợp dự báo khớp Negative.
- FP (False Positive): Tổng số trường hợp dự báo các quan sát thuộc nhãn Negative thành Positive.

- FN (False Negative): Tổng số trường hợp dự báo các quan sát thuộc nhãn Positive thành Negative.

Những chỉ số trên sẽ là cơ sở để tính toán những metric như accuracy, precision, recall, f1 score.

5.4.1.1. Accuracy

Accuracy là một chỉ số giúp đánh giá các mô hình phân loại. Chưa chính thức, độ chính xác là tỷ lệ dự đoán mà mô hình của chúng tôi đã có chính xác. Chính thức, độ chính xác có định nghĩa sau:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Đối với cách phân loại nhị phân, độ chính xác cũng có thể được tính theo dạng dương và âm:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5.4.1.2. Đường cong Precision-Recall

Precision (độ chính xác của dương tính): Là tỷ lệ giữa số lượng dự đoán dương tính đúng và tổng số lượng dự đoán dương tính. Độ chính xác này đánh giá khả năng mô hình đưa ra các dự đoán chính xác khi dự đoán một lớp cụ thể.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (độ phục hồi): Là tỷ lệ giữa số lượng dự đoán dương tính đúng và tổng số lượng thực tế dương tính trong dữ liệu. Độ phục hồi đo lường khả năng của mô hình trong việc tìm ra tất cả các mẫu thuộc lớp dương.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Đường cong Precision-Recall: thể hiện mối quan hệ giữa độ chính xác (precision) và độ hoàn thiện (recall) của mô hình cây quyết định. Đường cong PR càng gần góc trên bên trái (precision và recall cao), thì mô hình có hiệu suất tốt hơn trong việc phân loại các mẫu. Biểu đồ này hữu ích để đánh giá hiệu suất của mô hình trong các tác vụ phân loại không cân bằng, nơi số lượng mẫu thuộc các lớp khác nhau không đồng đều.

5.4.2. Đánh giá, so sánh các mô hình thuật toán dựa trên các độ đo và tiến hành dự báo

5.4.2.1. Đánh giá, so sánh các mô hình thuật toán

Độ đo Accuracy trên tập xác thực cho mỗi mô hình:

a/ **Decision Tree:**

Accuracy - Decision Tree

```
accuracy["Decision Tree"] = accuracy_score(y_validate, y_pred)
print("Accuracy - Decision Tree: ", accuracy["Decision Tree"])
```

Accuracy - Decision Tree: 0.743940990516333

b/ **Random Forest:**

Accuracy - Random Forest

```
accuracy["Random Forest"] = accuracy_score(y_validate, y_pred)
print("accuracy - Random Forest: ", accuracy["Random Forest"])
```

accuracy - Random Forest: 0.8008429926238145

c/ **Gaussian Naïve Bayes:**

Accuracy - Gaussian Naive Bayes

```
accuracy["Gaussian Naive Bayes"] = accuracy_score(y_validate, y_pred)
print("Accuracy - Gaussian Naive Bayes: ", accuracy["Gaussian Naive Bayes"])
```

Accuracy - Gaussian Naive Bayes: 0.6322444678609063

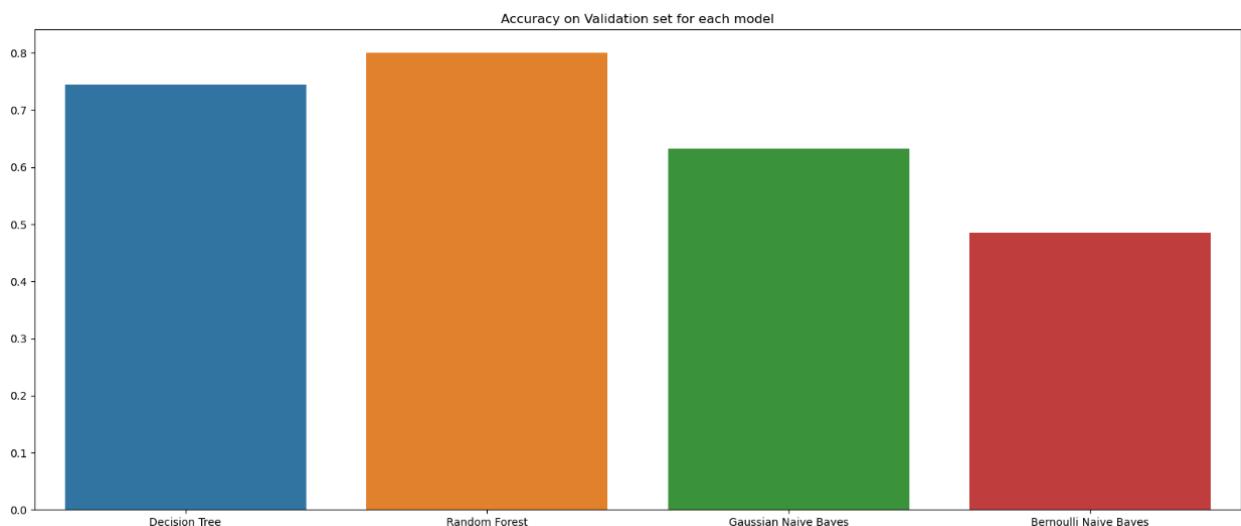
d/ **Bernoulli Naïve Bayes**

Accuracy - Bernoulli Naive Bayes

```
accuracy["Bernoulli Naive Bayes"] = accuracy_score(y_validate, y_pred)
print("Accuracy - Bernoulli Naive Bayes: ", accuracy["Bernoulli Naive Bayes"])
```

Accuracy - Bernoulli Naive Bayes: 0.48472075869336145

```
[111]: plt.figure(figsize=(20, 8))
plt.title("Accuracy on Validation set for each model")
sns.barplot(x = list(range(len(accuracy))), y = list(accuracy.values()))
plt.xticks(range(len(accuracy)), labels=accuracy.keys())
plt.show()
```



Độ đo F1 Score trên tập Validation cho mỗi mô hình:

```
f1["Decision Tree"] = f1_score(y_validate, y_pred, average="macro")
precisionScore = precision_score(y_validate, y_pred, average="macro")
recallScore = recall_score(y_validate, y_pred, average="macro")
print("f1-Score - Decision Tree: ", f1["Decision Tree"])
print("precision - Decision Tree: ", precisionScore)
print("recall - Decision Tree: ", recallScore)

f1-Score - Decision Tree: 0.4977437032138347
precision - Decision Tree: 0.4853480812755732
recall - Decision Tree: 0.514901394204453
```

- F1-Score - Decision Tree: 0.4977437032138347

```
f1["Random Forest"] = f1_score(y_validate, y_pred, average="macro")
precisionScore = precision_score(y_validate, y_pred, average="macro")
recallScore = recall_score(y_validate, y_pred, average="macro")
print("f1-Score - Random Forest: ", f1["Random Forest"])
print("precision - Random Forest: ", precisionScore)
print("recall - Random Forest: ", recallScore)

f1-Score - Random Forest: 0.501859993493743
precision - Random Forest: 0.4948757379826554
recall - Random Forest: 0.5105477778999109
```

- F1-Score - Random Forest: 0.501859993493743

```
f1["Gaussian Naive Bayes"] = f1_score(y_validate, y_pred, average="macro")
precisionScore = precision_score(y_validate, y_pred, average="macro")
recallScore = recall_score(y_validate, y_pred, average="macro")
print("f1-Score - Gaussian Naive Bayes: ", f1["Gaussian Naive Bayes"])
print("precision - Gaussian Naive Bayes: ", precisionScore)
print("recall - Gaussian Naive Bayes: ", recallScore)

f1-Score - Gaussian Naive Bayes: 0.4934940138304739
precision - Gaussian Naive Bayes: 0.5029670054561982
recall - Gaussian Naive Bayes: 0.5336230107577024
```

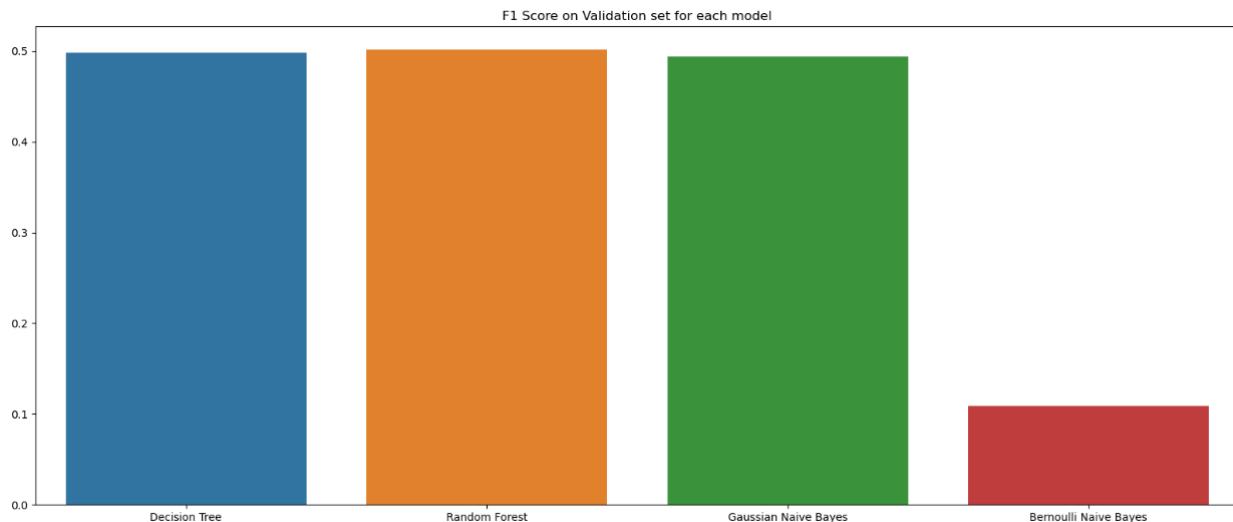
- F1-Score - Gaussian Naive Bayes: 0.4934940138304739

```
f1["Bernoulli Naive Bayes"] = f1_score(y_validate, y_pred, average="macro")
precisionScore = precision_score(y_validate, y_pred, average="macro")
recallScore = recall_score(y_validate, y_pred, average="macro")
print("f1-Score - Bernoulli Naive Bayes: ", f1["Bernoulli Naive Bayes"])
print("precision - Bernoulli Naive Bayes: ", precisionScore)
print("recall - Bernoulli Naive Bayes: ", recallScore)

f1-Score - Bernoulli Naive Bayes: 0.10882422521883132
precision - Bernoulli Naive Bayes: 0.08078679311556024
recall - Bernoulli Naive Bayes: 0.16666666666666666666
```

- F1-Score – Bernoulli Naïve Bayes: 0.10882422521883132

```
[112]: plt.figure(figsize=(20, 8))
plt.title("F1 Score on Validation set for each model")
sns.barplot(x = list(range(len(f1))), y = list(f1.values()))
plt.xticks(range(len(f1)), labels=f1.keys())
plt.show()
```



Kết luận:

Dựa trên các đồ thị và kết quả trên, ta có thể kết luận rằng thuật toán Random Forest là thuật toán tốt và phù hợp với bộ dữ liệu, vì các giá trị độ đo cao và ổn định hơn so với ba thuật toán còn lại.

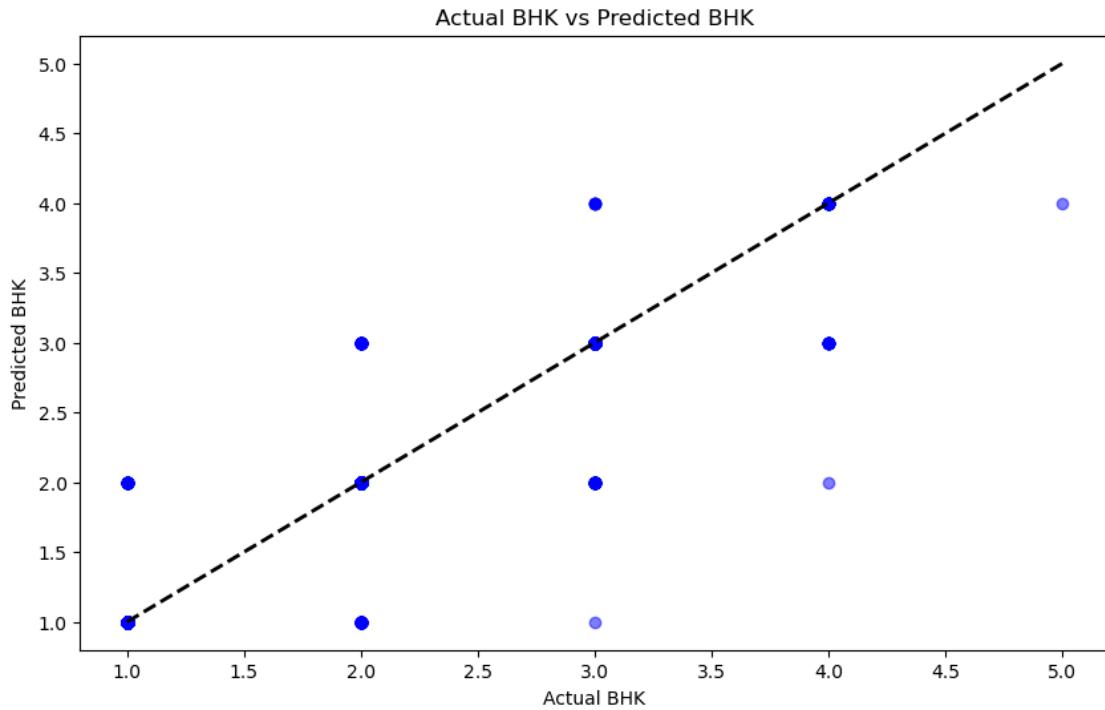
Nên ta dùng Random Forest để dự báo và đưa ra tập luật cho người dùng cuối.

5.4.2.2. Dự báo

```
[127]: sample = X_test
y_test_sample = sample["Rent"]
X_test_sample = sample.drop("Rent", axis=1)

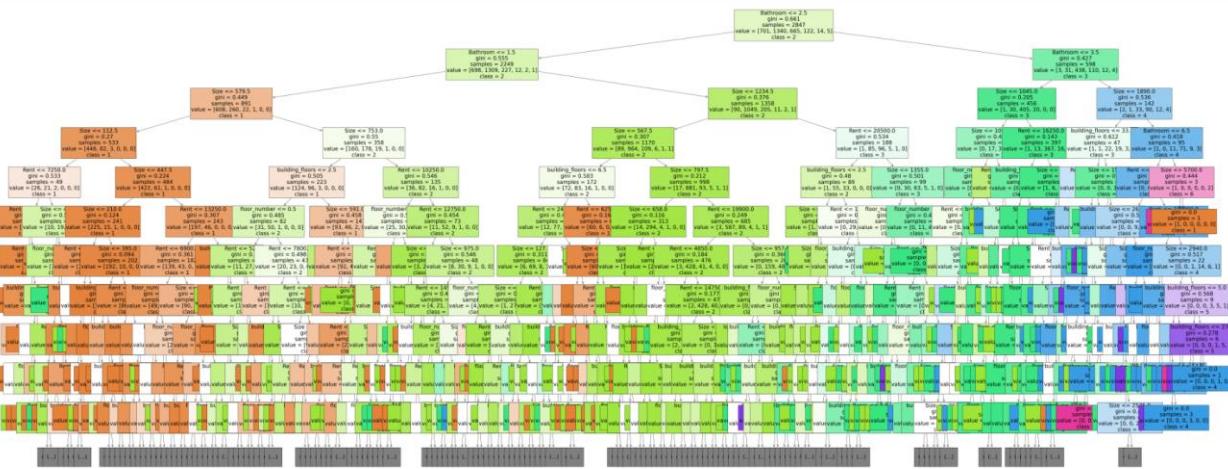
y_pred = rfc.predict(X_test_sample)

print(classification_report(y_test_sample, y_pred))
```



5.5. Tập luật cho người dùng cuối

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



- Nếu lần lượt Bathroom ≤ 2.5 , Bathroom $\leq 1.5 \leq 1.5$, Size ≤ 579.5 , Size ≤ 112.5 , Rent ≤ 7250.0 , Rent $\leq 2500.0 \Rightarrow$ **class 1: BHK = 1.**
- Nếu lần lượt Bathroom ≤ 2.5 , Bathroom $\leq 1.5 \leq 1.5$, Size ≤ 579.5 , Size ≤ 753.0 , Rent ≤ 10250.0 , floor_number $\leq 1.5 \Rightarrow$ **class 2: BHK = 2.**

- Nếu lần lượt Bathroom ≤ 2.5 , Bathroom ≤ 3.5 , Size ≤ 1045.0 , Rent $\leq 16250.0 \Rightarrow$ **class 3: BHK = 3.**
- Nếu lần lượt Bathroom ≤ 2.5 , Bathroom ≤ 3.5 , Size ≤ 1890.0 , Bathroom ≤ 6.5 , Rent $\leq 105000 \Rightarrow$ **class 4: BHK = 4.**
- Nếu lần lượt Bathroom ≤ 2.5 , Bathroom ≤ 3.5 , Size ≤ 1890.0 , Bathroom ≤ 6.5 , Size $\leq 5700 \Rightarrow$ **class 6: BHK = 6.**
- Nếu lần lượt Bathroom ≤ 2.5 , Bathroom ≤ 3.5 , Size ≤ 1890.0 , Bathroom ≤ 6.5 , Rent ≤ 105000.0 , Size ≤ 2940.0 , building_floor $\leq 5.0 \Rightarrow$ **class 5: BHK = 5.**

DANH MỤC TÀI LIỆU THAM KHẢO

[Online] <http://iase-web.org/documents/papers/icots5/Topic3c.pdf>

[Online] <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning>

Pfannkuch, C. W. a. M. "WHAT IS STATISTICAL THINKING? (Page 335)," 1998.

[Online] <http://iase-web.org/documents/papers/icots5/Topic3c.pdf>

Power BI Documentation. [Online] <https://learn.microsoft.com/en-us/power-bi/>

Random Forest. [Online] <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>

SQL Server 2012 Tutorials: Analysis Services - Data Mining File.

SQL Server 2012 Tutorials: Analysis Services - Multidimensional Modeling File.

CÁC TÀI LIỆU LIÊN QUAN