Shape, square

Description automatically generated

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**MÔN HỌC: KHO DỮ LIỆU VÀ OLAP**

**BÁO CÁO ĐỒ ÁN**

**Lớp: IS217.N21.HTCL**

**GVHD: ThS. Nguyễn Thị Kim Phụng**

Sinh viên thực hiện

Nguyễn Hoàng Long - 20521568

Đặng Thị Tường Vy - 20522176

MỤC LỤC

[**CHƯƠNG 1: MÔ TẢ DỮ LIỆU BAN ĐẦU** 3](#_Toc137881448)

[**1.1** **Tổng quan về dữ liệu** 3](#_Toc137881449)

[**1.2** **Mô tả thuộc tính** 4](#_Toc137881450)

[**1.2.1** **Bảng Fact** 4](#_Toc137881451)

[**1.2.2** **Bảng Dim\_Company** 4](#_Toc137881452)

[**1.2.3** **Bảng Dim\_Time** 5](#_Toc137881453)

[**1.2.4** **Bảng Dim\_Tag** 5](#_Toc137881454)

[**1.2.5** **Bảng Dim\_Title** 5](#_Toc137881455)

[**1.2.6** **Bảng Dim\_Level** 5](#_Toc137881456)

[**1.2.7** **Bảng Dim\_City** 6](#_Toc137881457)

[**1.2.8** **Bảng Dim\_Education** 6](#_Toc137881458)

[**CHƯƠNG 2: XÂY DỰNG KHO DỮ LIỆU – QUÁ TRÌNH SSIS** 6](#_Toc137881459)

[**2.1** **Tạo mới Database** 6](#_Toc137881460)

[**2.2** **Tạo project mới với SQL Data Tool** 7](#_Toc137881461)

[**2.2** **Quá trình tích hợp dữ liệu** 13](#_Toc137881462)

[**2.2.1** **Tạo bảng Dim** 13](#_Toc137881463)

[**2.2.2** **Tạo Bảng Fact** 21](#_Toc137881464)

[**CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU – QUÁ TRÌNH SSAS** 30](#_Toc137881465)

[**3.1** **Cấu hình SSAS** 30](#_Toc137881466)

[**3.1.1** **Khởi tạo project SSAS** 30](#_Toc137881467)

[**3.1.2** **Tạo Data sources, Data source views, Cubes** 30](#_Toc137881468)

[**3.2** **Quá trình phân tích dữ liệu bằng ngôn ngữ MDX** 46](#_Toc137881469)

[**3.2.1** **Lấy thông tin về các vị trí công việc có mức lương trên 200,000** 46](#_Toc137881470)

[**3.2.2** **Tìm kiếm các vị trí công việc có mức lương trong nằm trong TOP15 ở thành phố New York** 49](#_Toc137881471)

[**3.2.3** **Đếm số nhân sự ở từng vai trò của từng công ty** 51](#_Toc137881472)

[**3.2.4** **Đếm số vị trí có title là Software Engineer mà công ty trả lương theo từng năm, từng tháng** 53](#_Toc137881473)

[**3.2.5** **Tại thành phố London thống kê xem top 5 công ty trả lương cao nhất** 53](#_Toc137881474)

[**3.2.6** **Thống kê số vị trí theo từng level của vai trò Data Scientist của từng công ty** 54](#_Toc137881475)

[**3.2.7** **Thống kê mức lương cơ bản theo từng vị trí mà Google đã trả cho nhân viên** 56](#_Toc137881476)

[**3.2.8** **Thống kê level, mức lương giảm dần vị trí full stack, của công ty Facebook ở thành phố San francisco** 57](#_Toc137881477)

[**3.2.9** **Thống kê mức lương cơ bản cho những vị trí học vấn tiến sĩ, giới tính nữ ở công ty Microsoft** 58](#_Toc137881478)

[**3.2.10** **Liệt kê top 3 vị trí có số năm kinh nghiệm lớn nhất của vai trò Data Scientist và mức lương của vị trí đó trong quý 2 năm 2020** 59](#_Toc137881479)

[**3.2.11** **Thống kê tổng tiền thưởng theo mỗi vai trò của từng công ty trong quý 2 năm 2020** 60](#_Toc137881480)

[**3.2.12** **Thống kê vị trí có số lương cơ bản cao nhất trong năm 2020 của công ty Microsoft** 63](#_Toc137881481)

[**3.2.13** **Thống kê số chi nhánh của công ty Facebook ở trong các thành phố** 64](#_Toc137881482)

[**3.2.14** **Liệt kê top 10 công ty có nhiều Product Manager có giới tính nữ nhất trong năm 2019** 65](#_Toc137881483)

[**CHƯƠNG 4: KHAI THÁC DỮ LIỆU – QUÁ TRÌNH DATAMINING** 67](#_Toc137881484)

[**4.1 PREPROCESSING** 67](#_Toc137881485)

[**4.2 Sử dụng Random Forest chọn các thuộc tính ảnh hưởng** 69](#_Toc137881486)

[**4.3 Trực quan dữ liệu** 73](#_Toc137881487)

[**4.4 Data mining** 75](#_Toc137881488)

[**4.4.1 Phân cụm dữ liệu bằng Kmeans** 75](#_Toc137881489)

[**TÀI LIỆU THAM** **KHẢO** 83](#_Toc137881490)

# **CHƯƠNG 1: MÔ TẢ DỮ LIỆU BAN ĐẦU**

## **Tổng quan về dữ liệu**

Dữ liệu mô tả bản ghi tiền lương của các công ty hàng đầu như Oracle, Ebay, Microsoft… không chỉ tiền lương mà thông qua đó người dùng còn biết được thông tin về trình độ học vấn, lương thưởng (lương cơ bản, tiền thưởng, …), chủng tộc, giới tính… của nhân viên.

Nội dung dữ liệu gồm 62643 bản ghi tiền lương, thông tin cá nhân của nhân viên từ 29 công ty được mô tả.

Link dữ liệu: [DataScience\_Stem\_Salary\_DataSet](https://www.kaggle.com/datasets/jackogozaly/data-science-and-stem-salaries)

## **Mô tả thuộc tính**

### **Bảng Fact**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Fact\_id | Khóa chính |
| Time | Thời gian dữ liệu được ghi lại |
| Company\_id | ID của công ty |
| Level\_id | ID của trình độ |
| Title\_id | ID của vai trò |
| City\_id | ID của thành phố |
| Basesalary | Lương cơ bản |
| Yearofexperience | Số năm kinh nghiệm |
| Bonus | Thưởng thêm |
| Gender | Giới tính |
| Tag\_id | ID vị trí công việc |
| education\_id | ID trình độ học vấn |
| Masters\_Degree | Có bằng thạc sĩ hay không |

### **Bảng Dim\_Company**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Company\_id | ID của company (Khóa chính) |
| Company | Tên của công ty |

### **Bảng Dim\_Time**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Time | Khóa chính |
| Month | Tháng |
| Quarter | Quý |
| Year | Năm |

### **Bảng Dim\_Tag**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Tag\_id | Khóa chính |
| Tag | Tên vị trí công việc |

### **Bảng Dim\_Title**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Title\_id | Khóa chính |
| Title | Tên vai trò |

### **Bảng Dim\_Level**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Level\_id | Khóa chính |
| Level | Trình độ |

### **Bảng Dim\_City**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| City\_id | Khóa chính |
| City\_name | Tên thành phố |
| Location | Tên vùng |

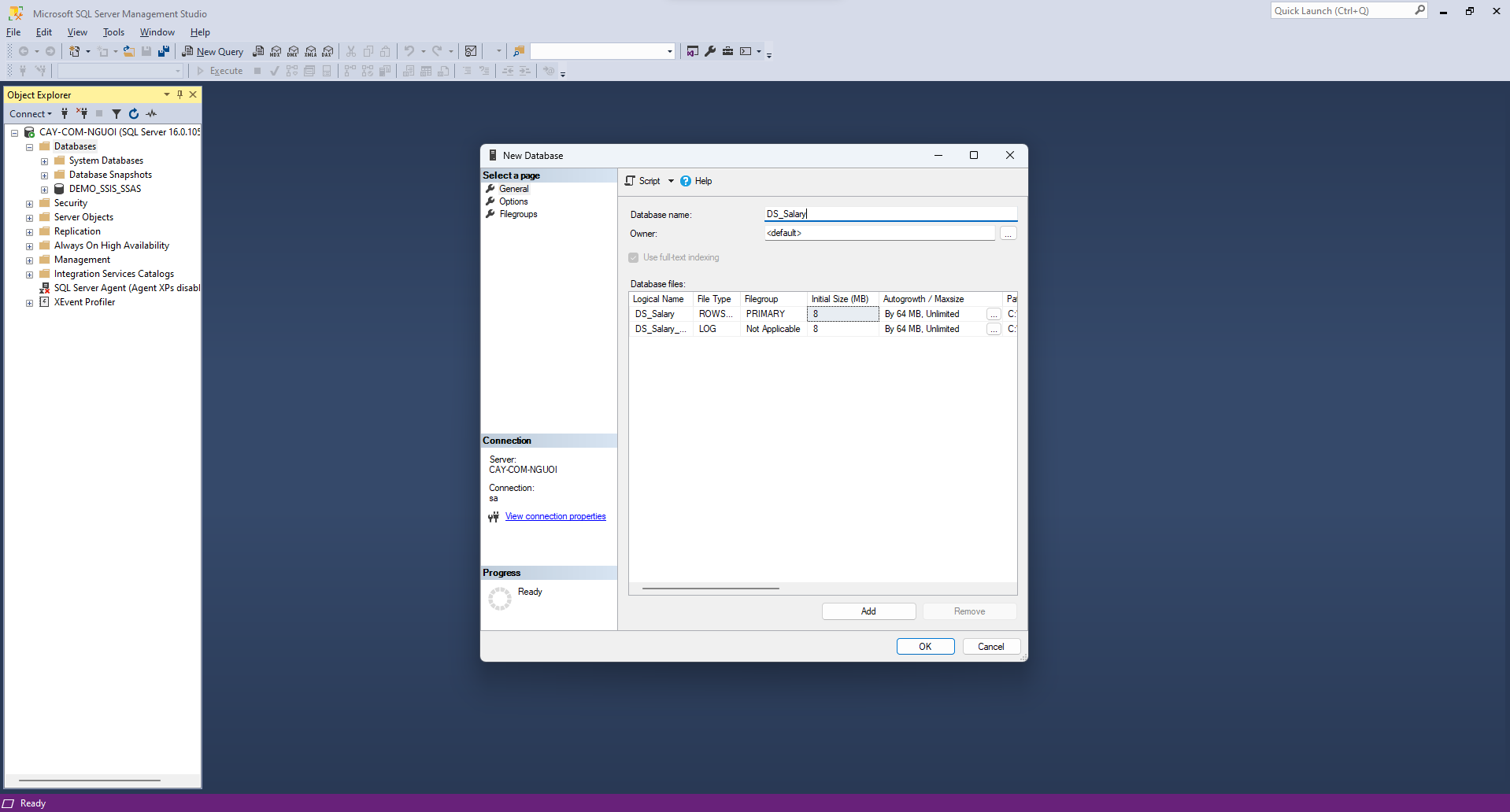
### **Bảng Dim\_Education**

|  |  |
| --- | --- |
| Tên thuộc tính | Mô tả |
| Education\_id | Khóa chính |
| Education | Trình độ học vấn |
|  |  |

# **CHƯƠNG 2: XÂY DỰNG KHO DỮ LIỆU – QUÁ TRÌNH SSIS**

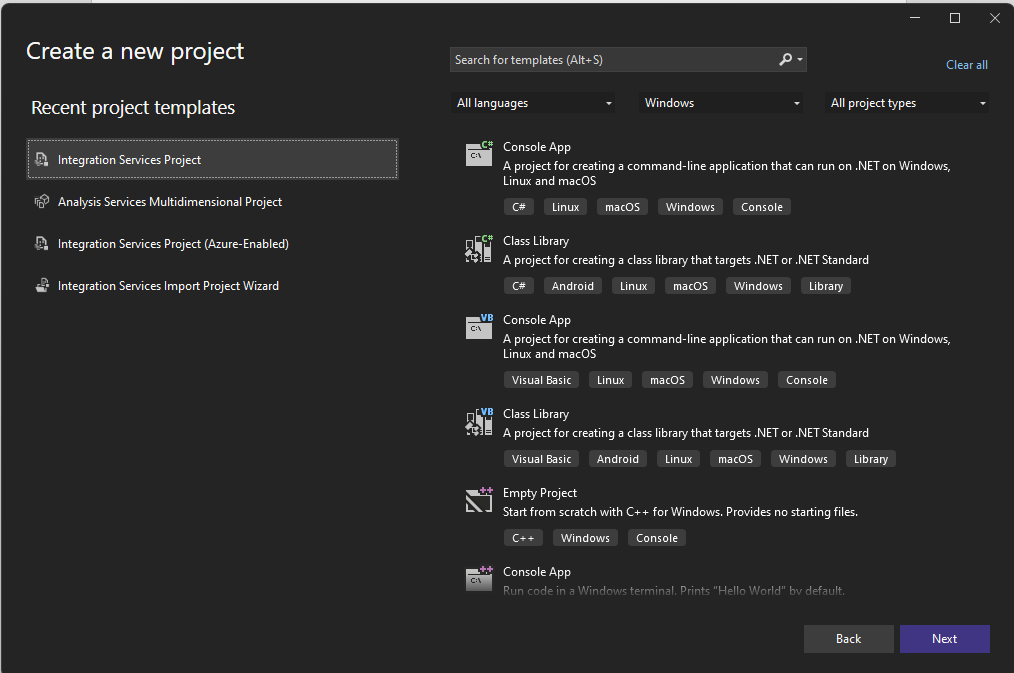
## **Tạo mới Database**

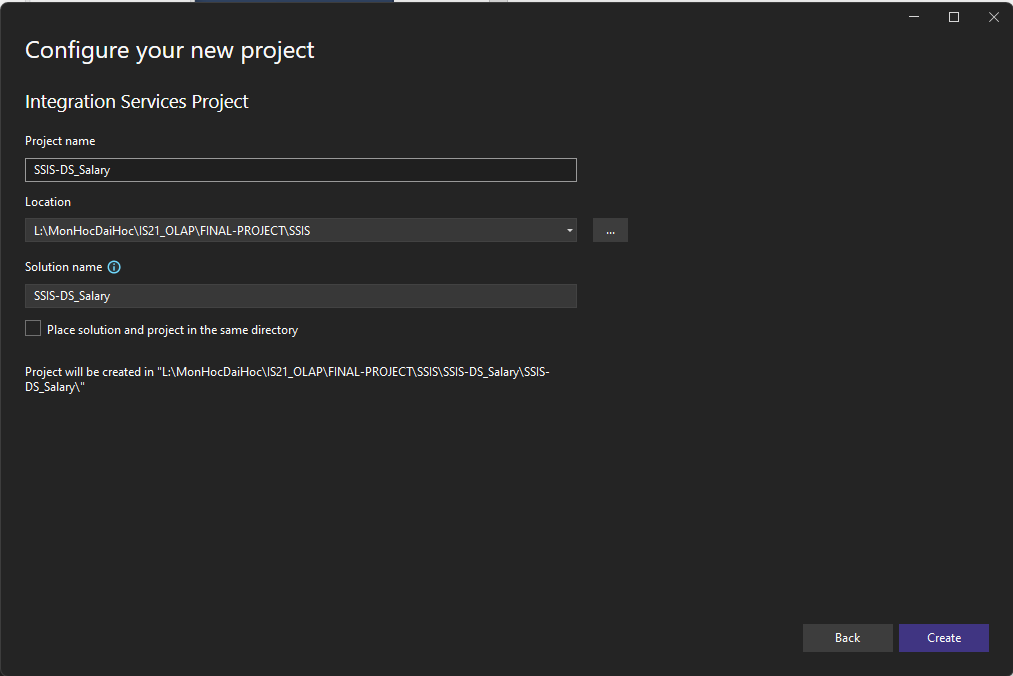
Dùng SQL Server 2019 tạo một Database với tên là *DS\_Salary* dùng để thực lưu trữ dữ liệu



## **2.2 Tạo project mới với SQL Data Tool**

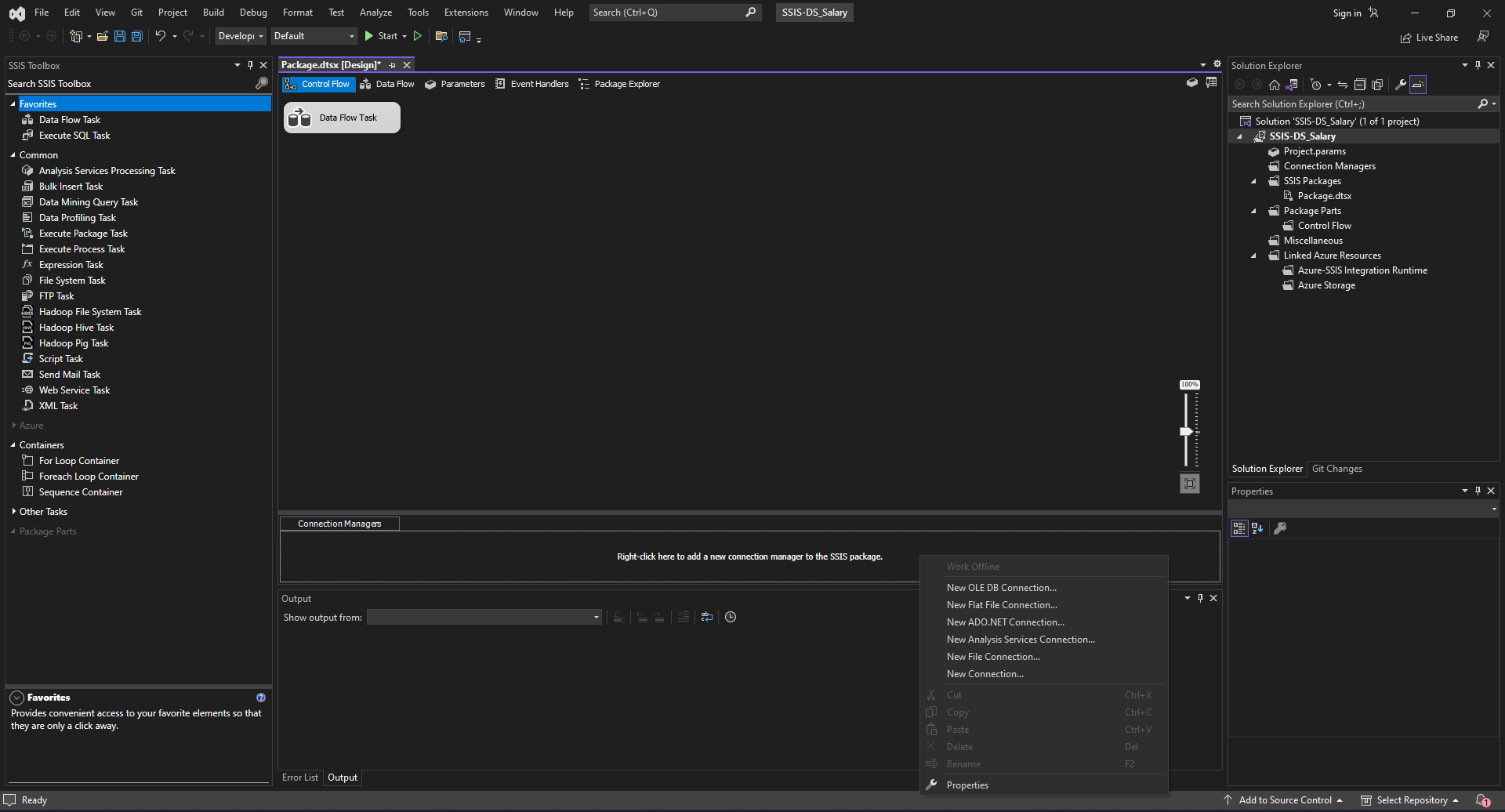
Dùng Visual Studio 2022 tạo một SSIS project với tên gọi là SSIS-DS\_Salary



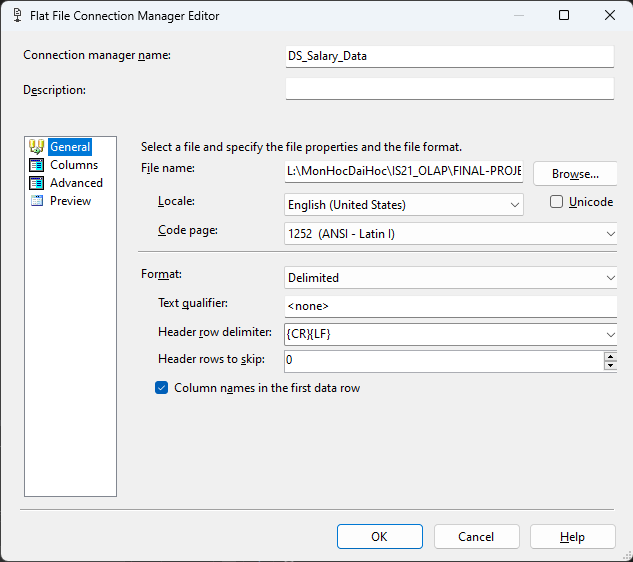


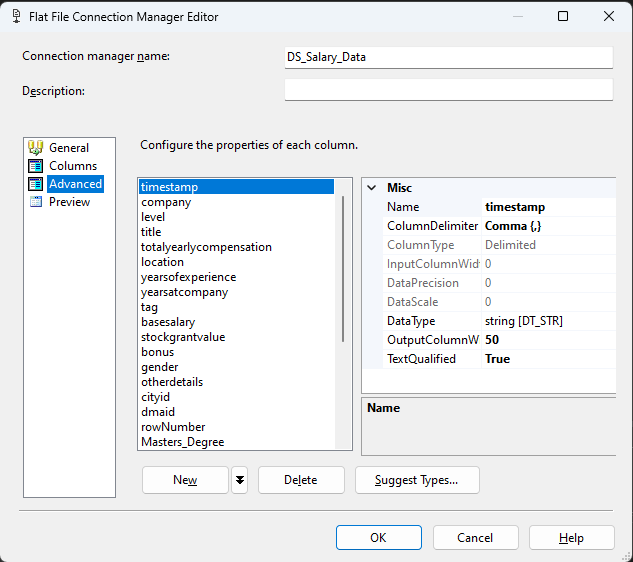
* **Import dữ liệu từ file CSV vào flat file source**

Click chuột phải vào ô “*Right-click here to add new connection manager to the SSIS pakage*” => chọn New Flat File connection

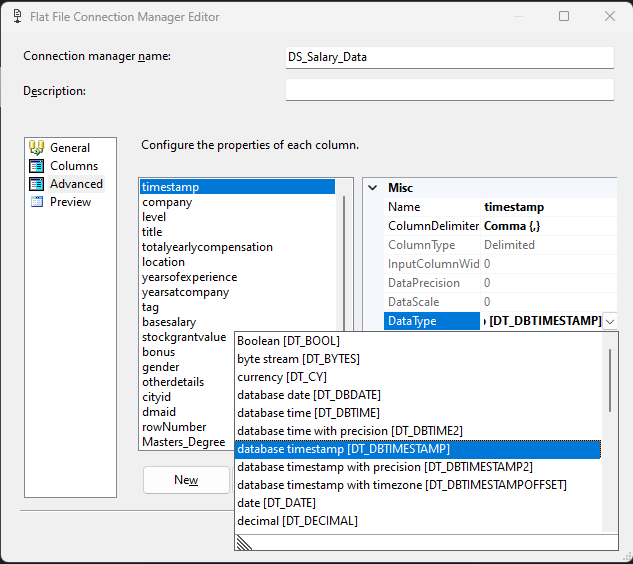


Đặt tên Data source là DS\_Salary\_Data

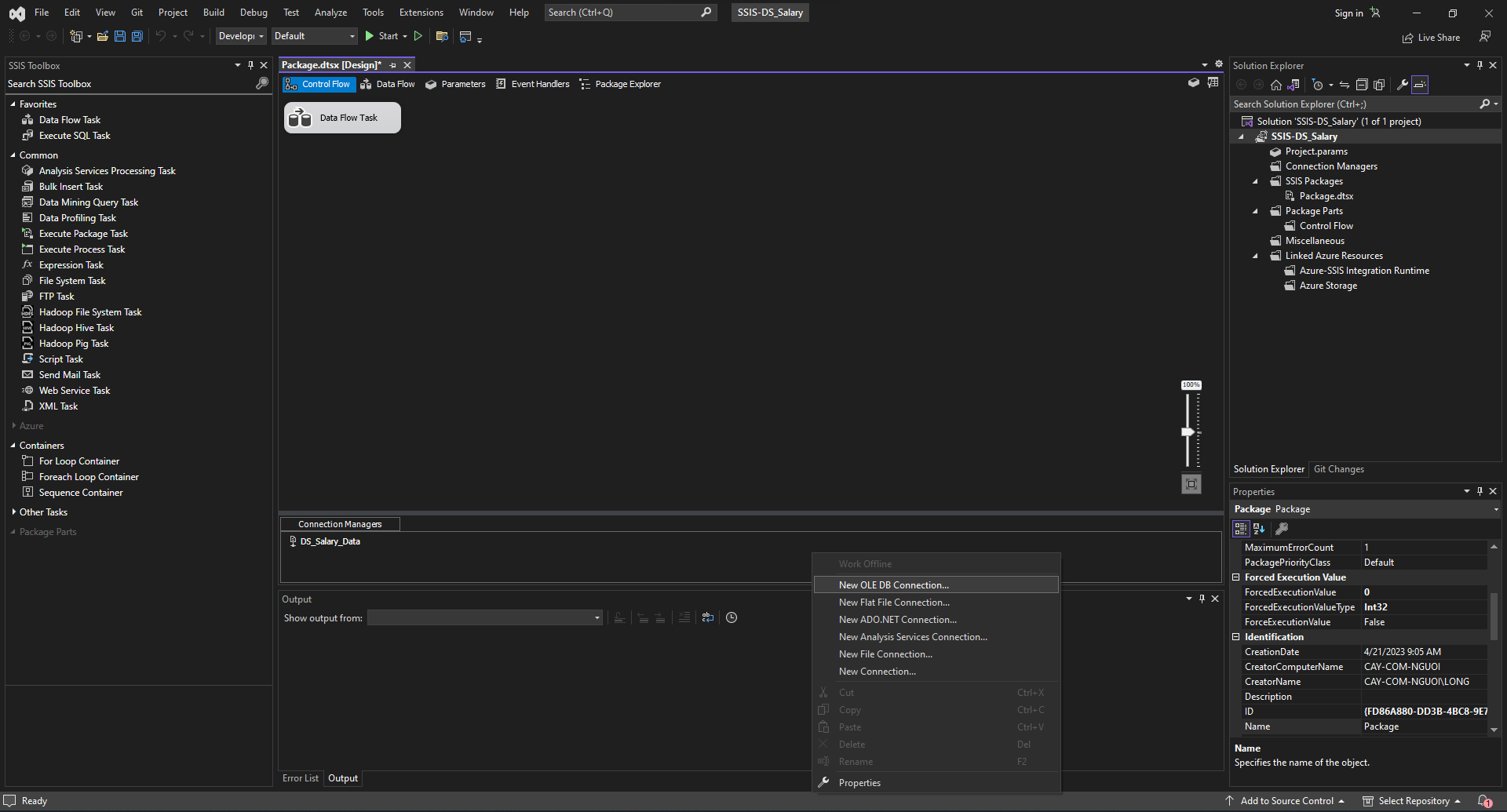




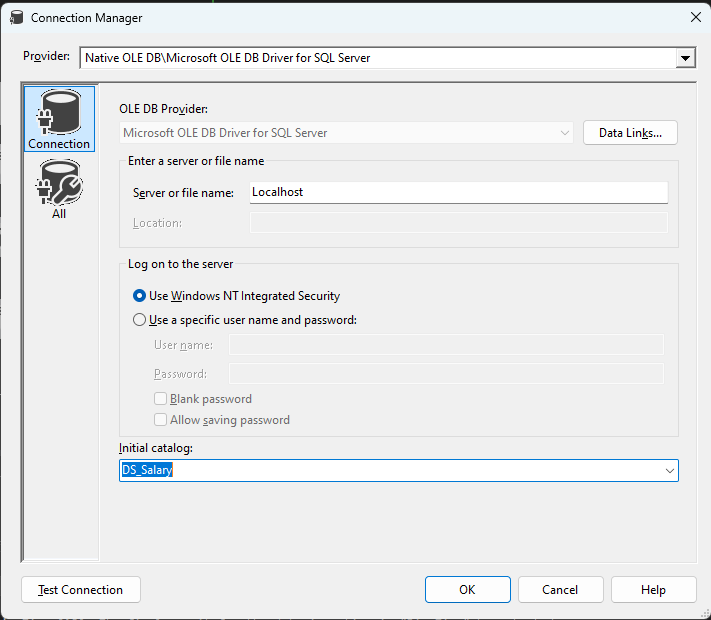
Chỉnh sửa kiểu dữ liệu (Data Type) từ *string[DT\_STR]* thành *Database timestamp [DT\_DBTIMESTAMP]*



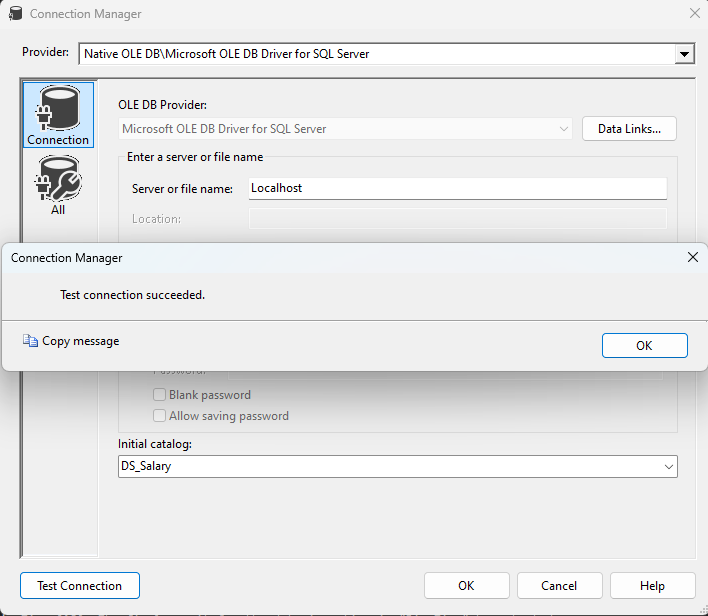
* Tạo mới Database để tích hợp dữ liệu



Tiếp theo nhập tên Server, tên Database (DS\_Salary) mà ta đã tạo trước.



Test thử xem đã connect được chưa



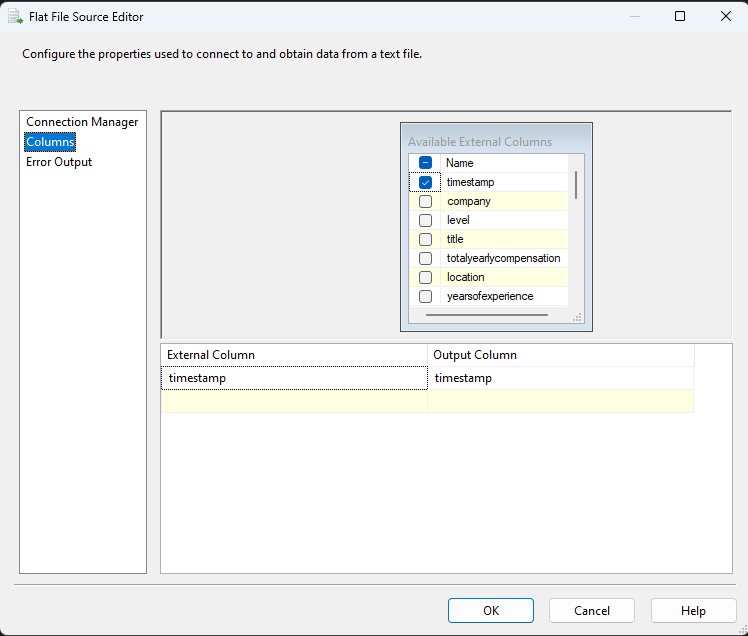
## **Quá trình tích hợp dữ liệu**

**2.2.1 Tạo bảng Dim**

* Bảng Dim\_Time

Trong tab *Control Flow* đổi tên *Data flow task* thành *Time* => double click vào Time

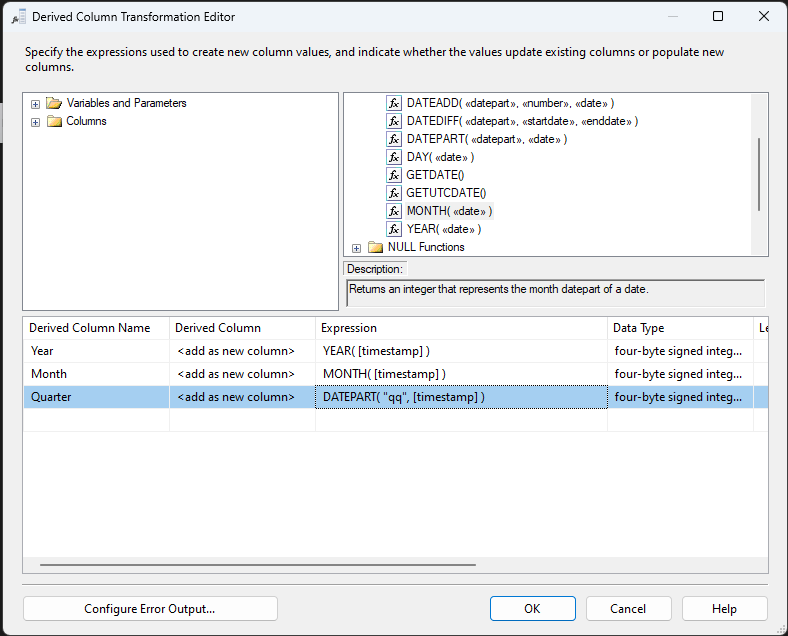
Kéo Flat file Source vào Data flow => Double click => Columns => chọn timestamp



Tiếp theo cần tách Time thành month, quarter, year, để làm điều này thì ta dùng *Derived Column*

Kéo Derived Column vào Data Flow => double click

Dùng những *Date/Time functions* để tách timestamp thành các cột như *Month, Year, Quarter*



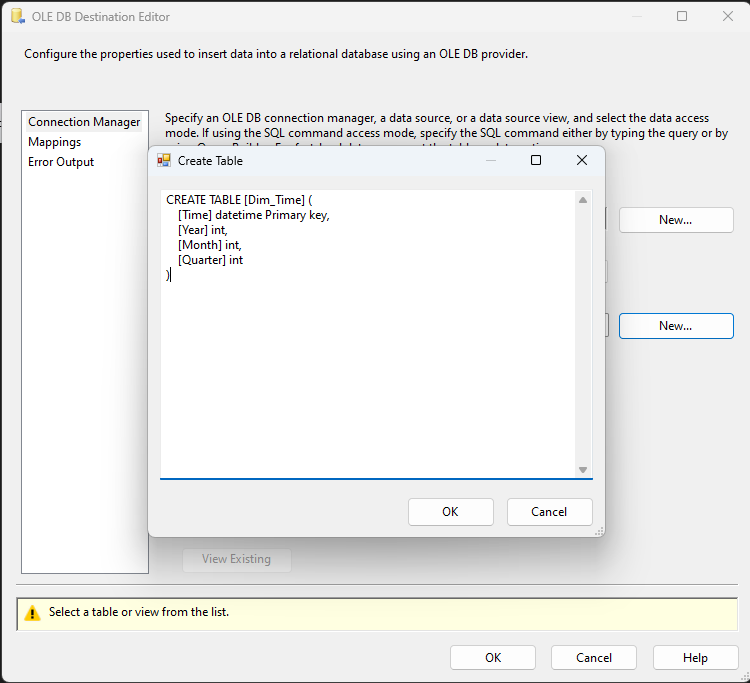
Tiếp theo kéo Sort và OLE DB Destination vào Data Flow

Double- click vào Sort => tích vào Remove rows with duplicate values và timestamp => OK

Đổi tên OLE DB Destination thành Dim\_Time

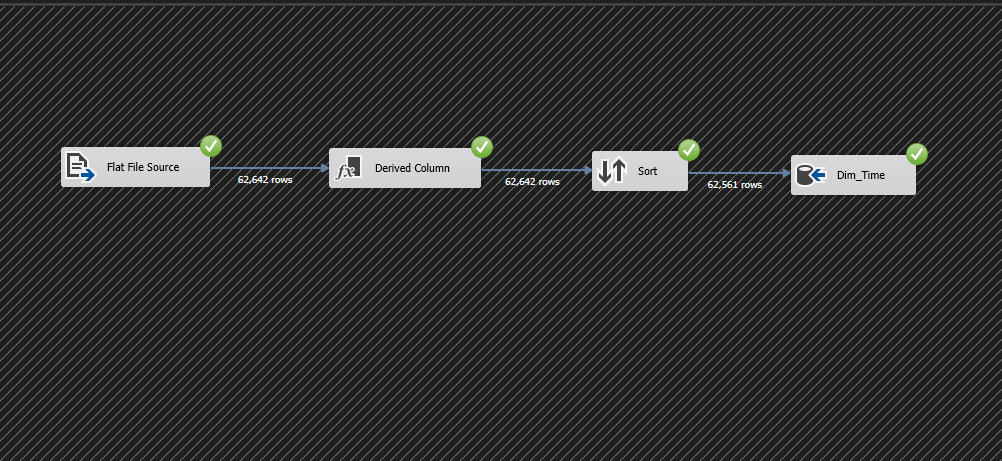
Double-click vào Dim\_time => Vào New… để tạo Bảng Dim\_time chứa dữ liệu trong database, thiết kế bảng như sau:

Sau đó vào mapping để chuyển đổi dữ liệu. => OK



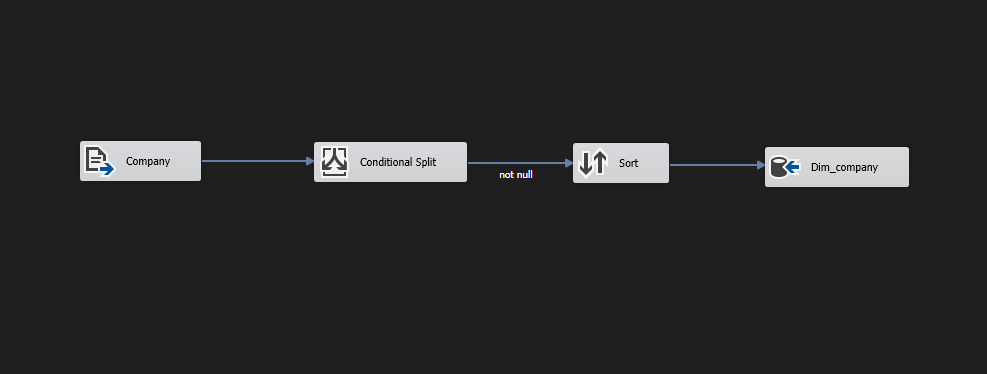
Sau đó ta chạy start để bắt đầu quá trình tích hợp dữ liều vào SQL Server

Đây là màn hình hiển thị khi quá trình tích hợp thành công

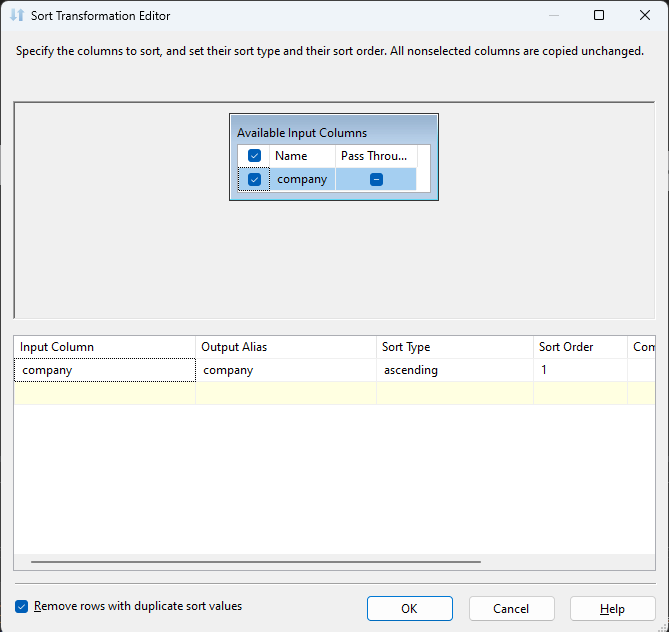


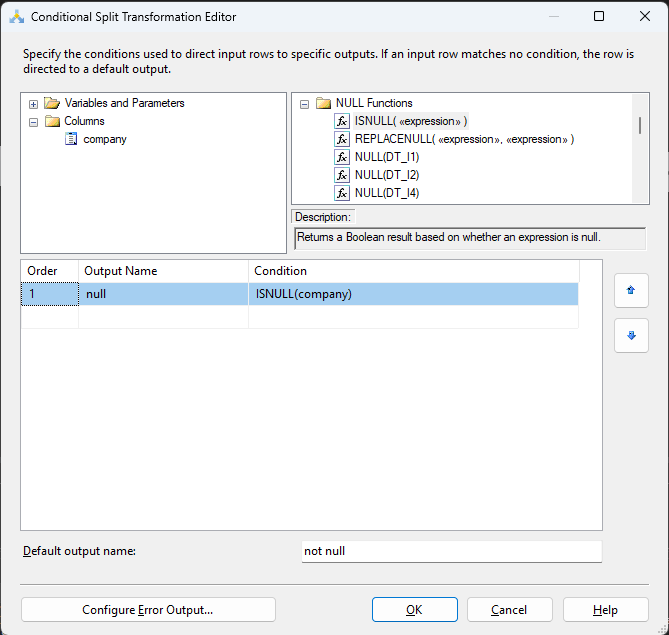
* Thực hiện Bảng Dim\_Company

Ta sẽ xây dựng Data Flow được minh họa như sau:

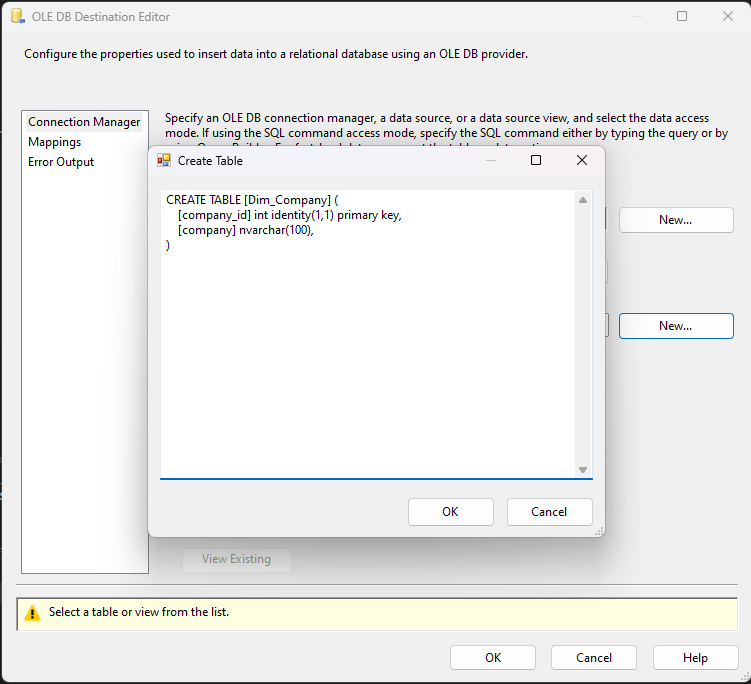


* Ở cửa sổ Control flow tạo một Data Flow Task => Đổi tên thành Company
  + Double-click vào Company
    - Kéo Flat File Source vào Data Flow => Đổi tên thành company
  + Ta làm tương tự với bảng Dim\_Time => chọn Flat file connection manager là DS\_Salary
  + Click vào Columns => Tích vào company. => OK
* Kéo Conditional split vào để xác định xem những hàng nào đã được định tuyến.
  + Double-click vào Conditional Split => vào NULL Functions => sử dụng ISNULL
  + Để Default output name là “not null” => OK

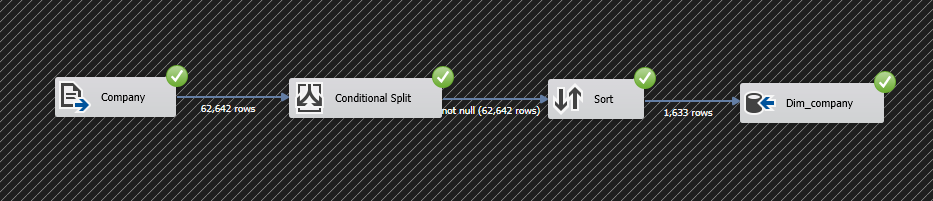




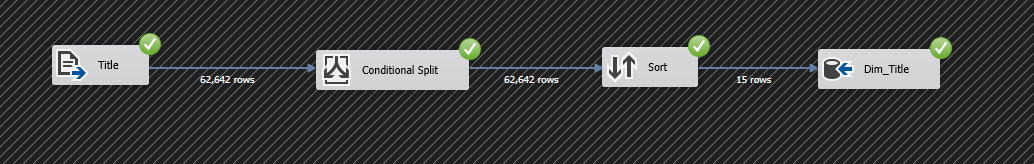
* Kéo Sort vào Data flow
  + Double-click vào Sort
  + Tích vào ô remove rows with duplicate sort value và ô company
* Kéo OLE DB Destination vào Data Flow
  + Đổi tên thành Dim\_Company
  + Double-click vào Dim\_Company => chọn new… để tạo bảng mới
  + Thiết kế bảng Dim\_Company giống như hình dưới => OK



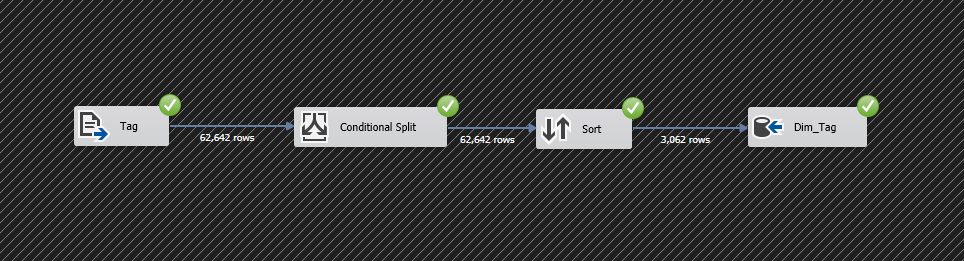
* Bắt đầu chạy start để chạy quá trình tích hợp dữ liệu



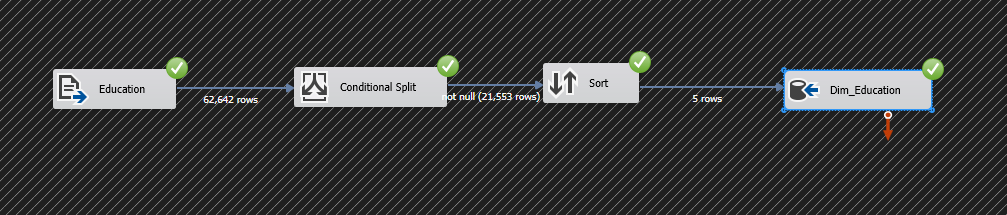
* **Tương tự với các bảng khác ta có**
* Bảng Dim\_Title



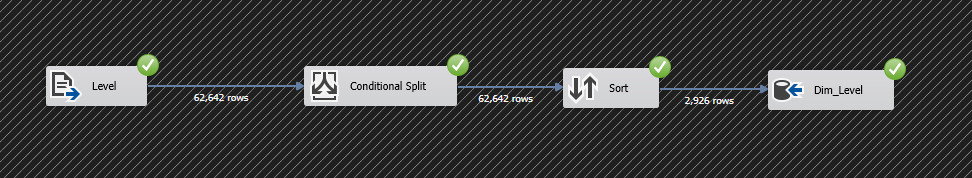
* Bảng Dim\_Tag



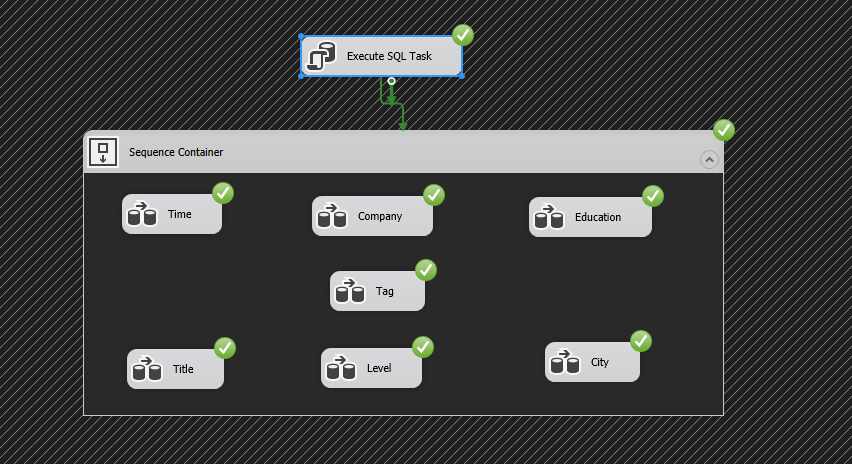
* Bảng Dim\_Education



* Bảng Dim\_Level



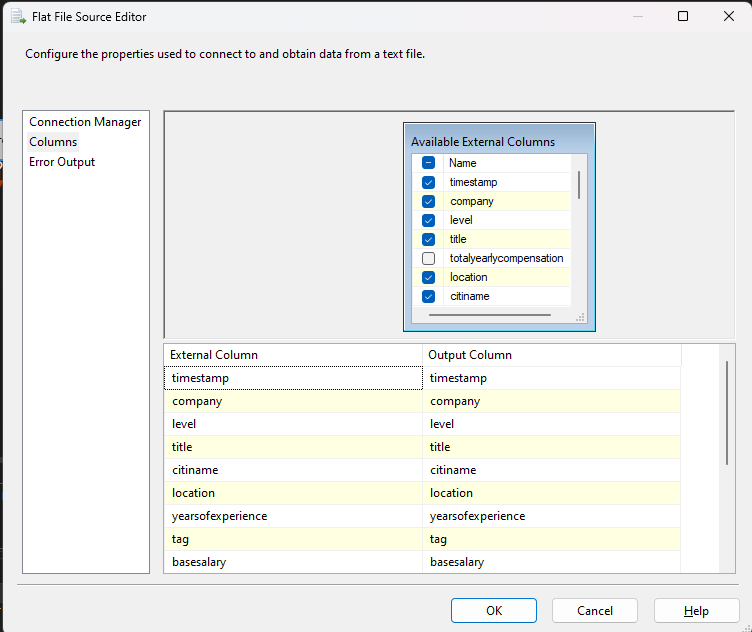
Gom những Data Flow Task như Title, company, … vào trong 1 Sequence Container sau đó chạy Excute SQL Task



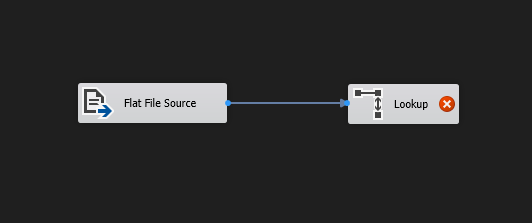
* + 1. **Tạo Bảng Fact**

Ta làm tương như các bảng khác

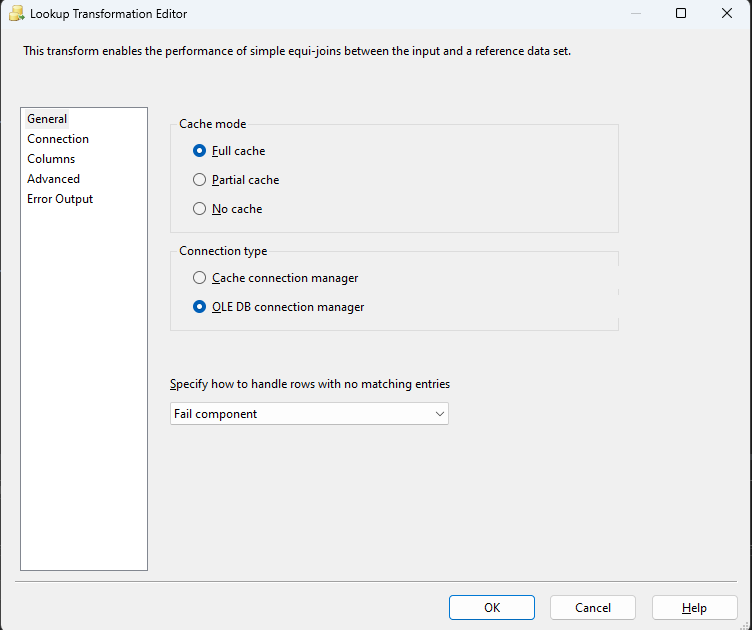
Kéo Flat file source để tạo dữ liệu đầu vào, sau đó chọn các thuộc tính cho bảng fact

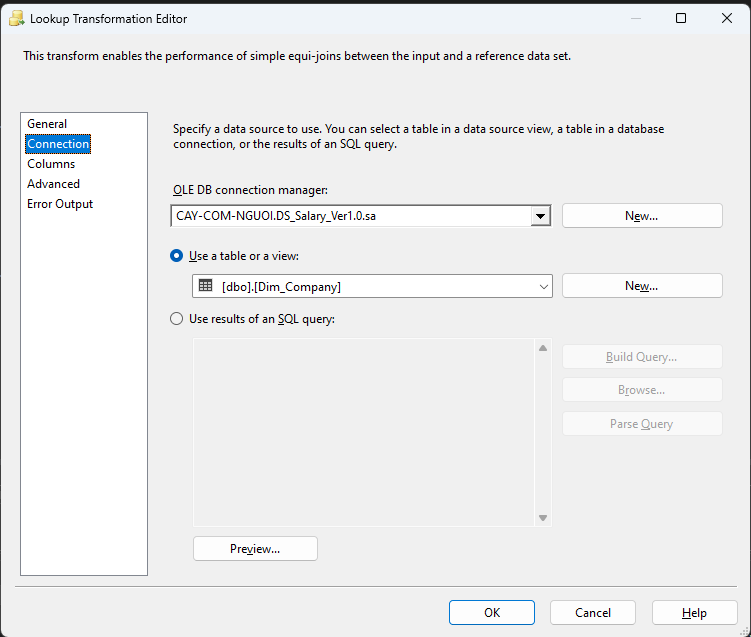


Thêm Lookup để tích hợp dữ liệu từ các bảng Time, Company, Education, … vào bảng Fact



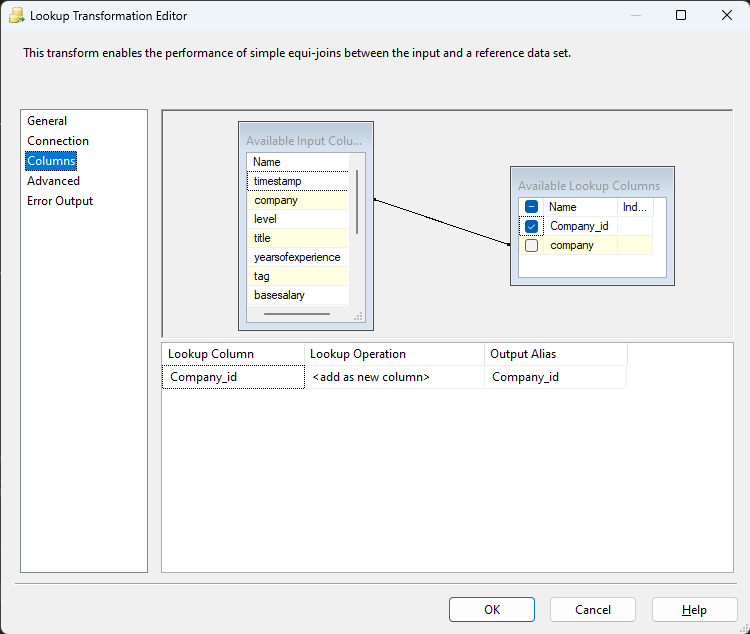
Double-click vào Lookup

 vào connection => chọn bảng cần tích hợp (ví dụ ở đây là bảng Company)



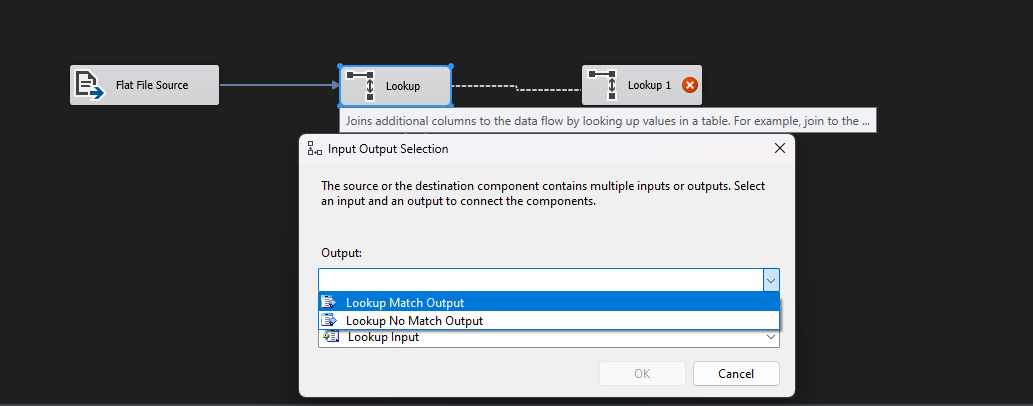
Vào columns => Tích chọn Company\_id

Kéo Company từ cột *Available input columns* sang Company ở cột *Available lookup columns*. => OK

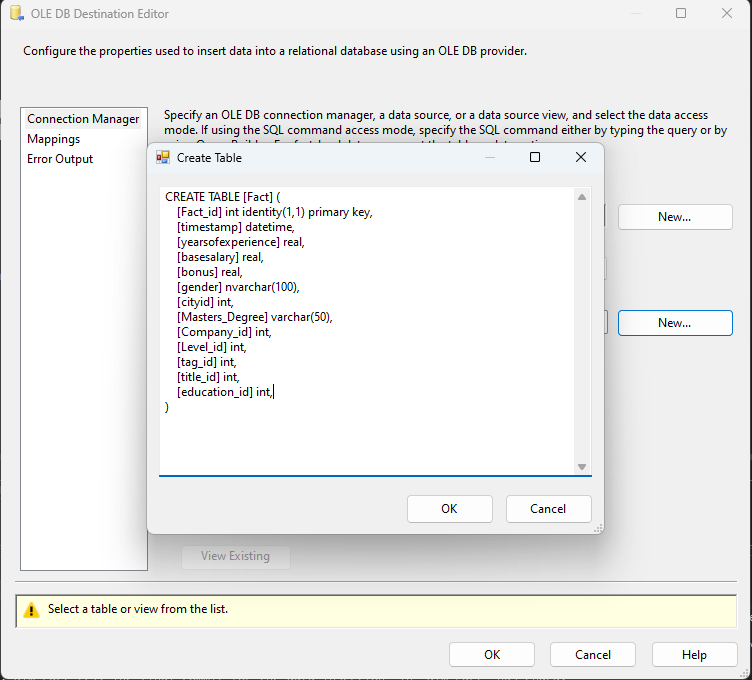


Làm tương tự với Look up cho các bảng khác. Nhưng vì city\_id ta đã có sẵn rồi nên không cần tạo Look up cho Dim\_City

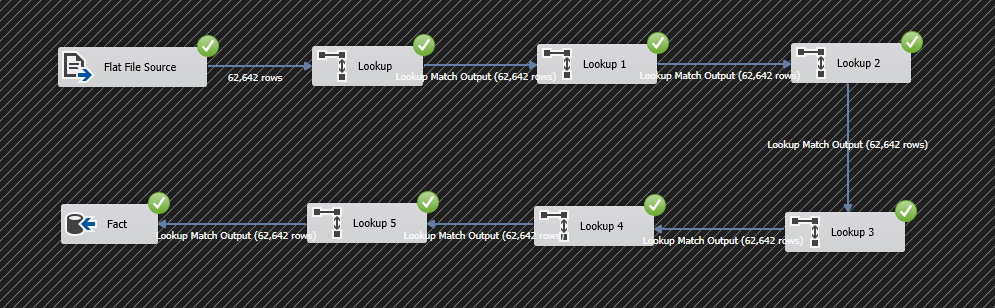
Khi nối các Look up lại với nhau ta chọn *Look up Match Output*



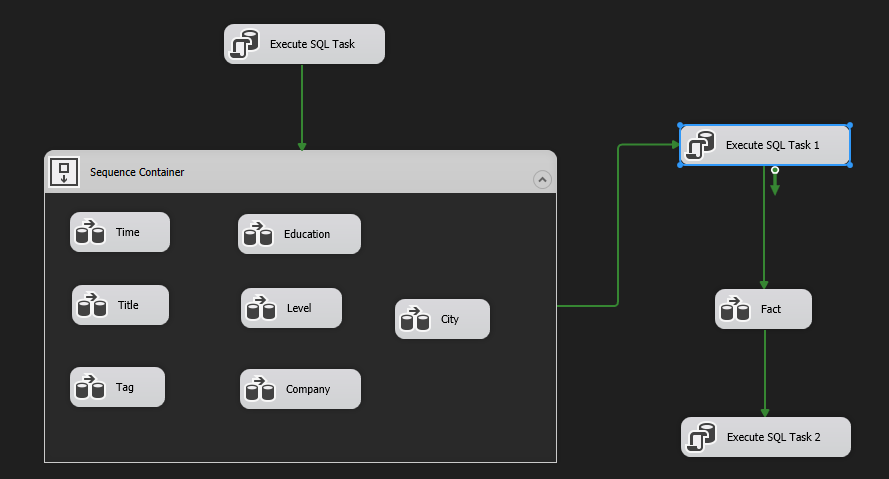
Cuối cùng Tạo 1 OLE DB Destination với tên là Dim\_Fact => Tạo bảng Dim\_Fact với thuộc tính như sau:



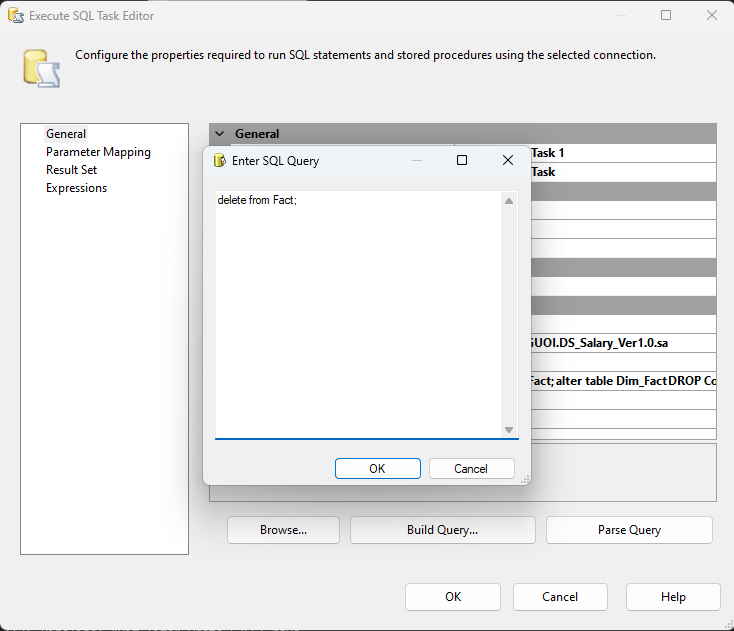
Cuối cùng ta có Data Flow của bảng Fact như sau:



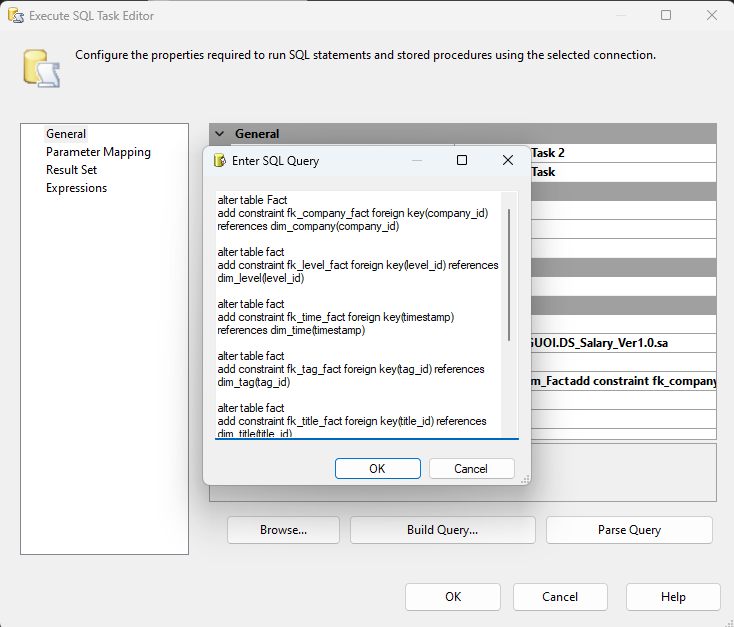
Thiết Kế mô hình Control Flow như hình



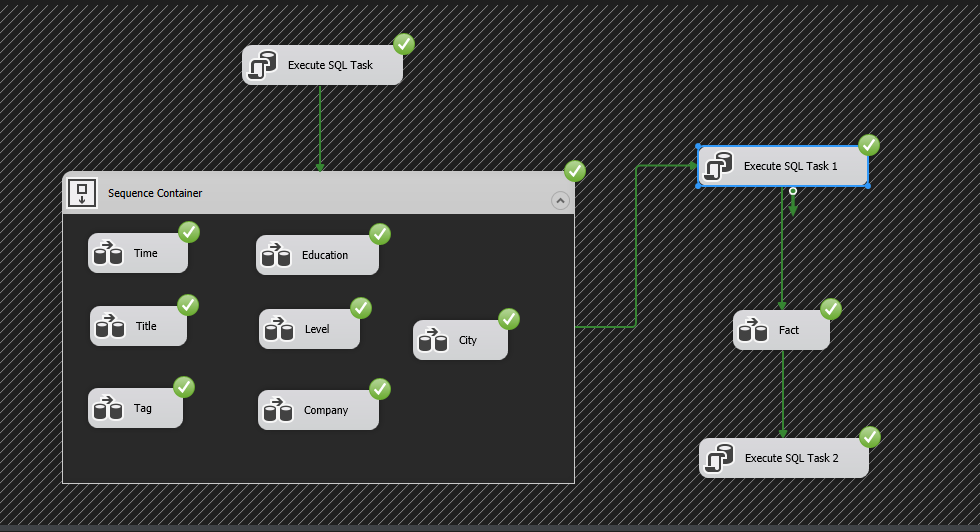
ở Excute SQL Task 1 ta cần làm sạch bảng Dim\_Fact



ở Excute SQL Task 2 thực hiện thêm khóa ngoại cho bảng Dim\_Fact

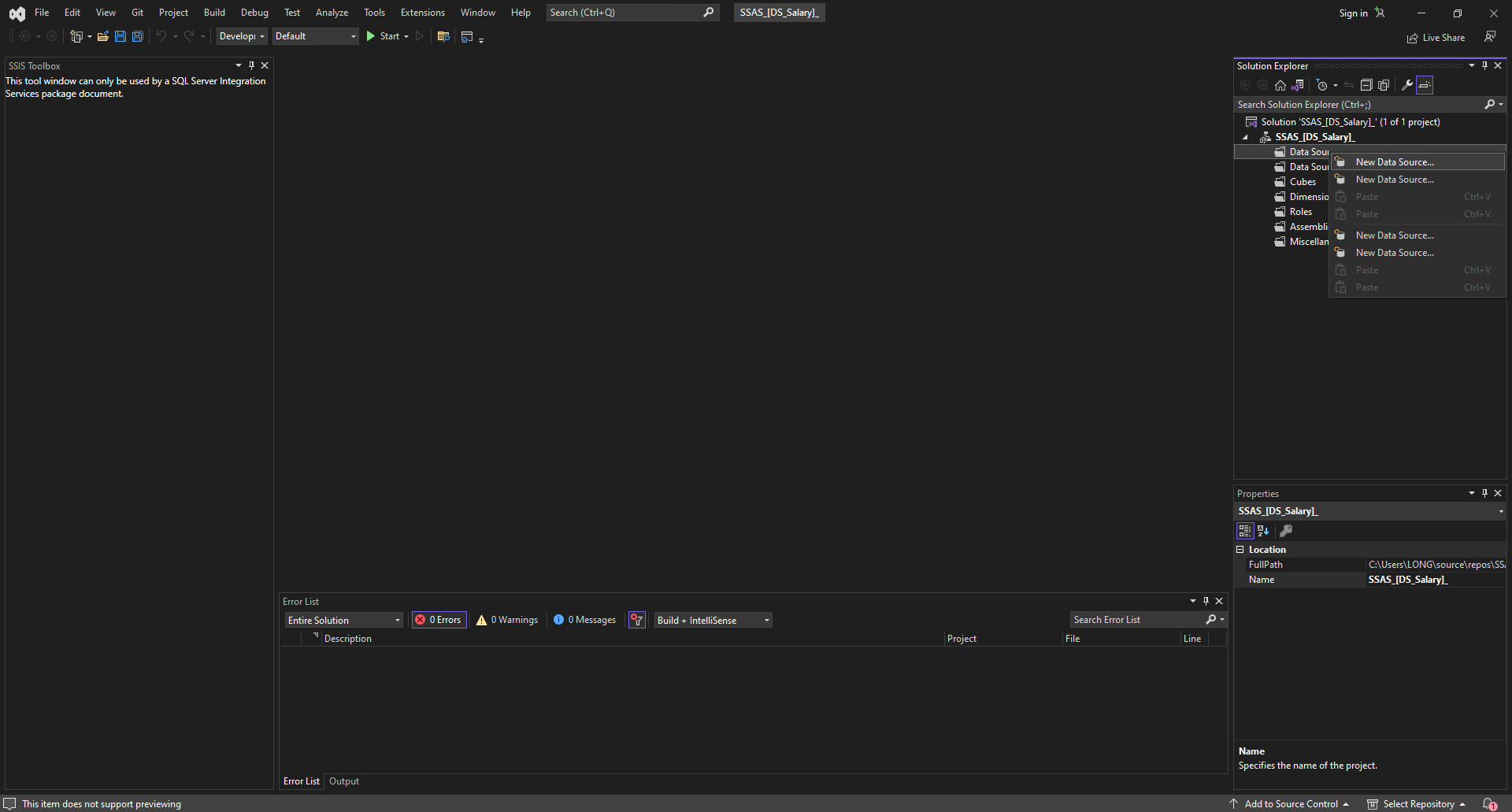


Như vậy là ta đã hoàn thành tiến trình SSIS cho các bảng Dim và bảng Fact



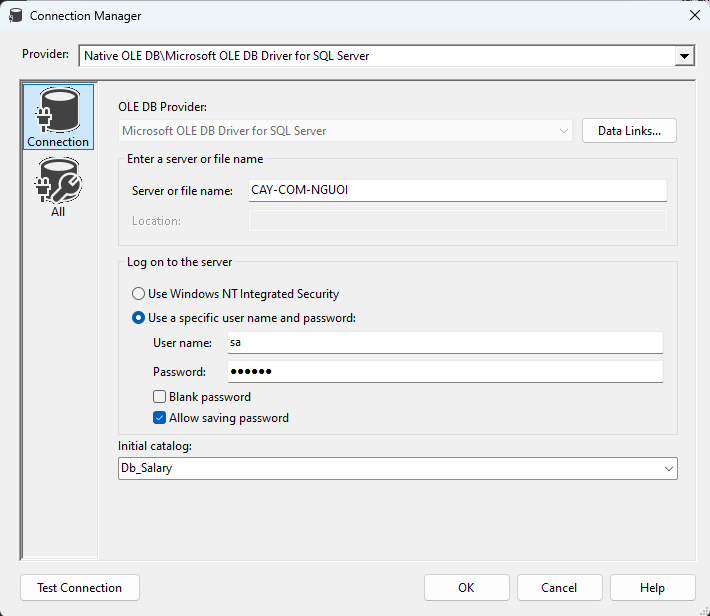
# **CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU – QUÁ TRÌNH SSAS**

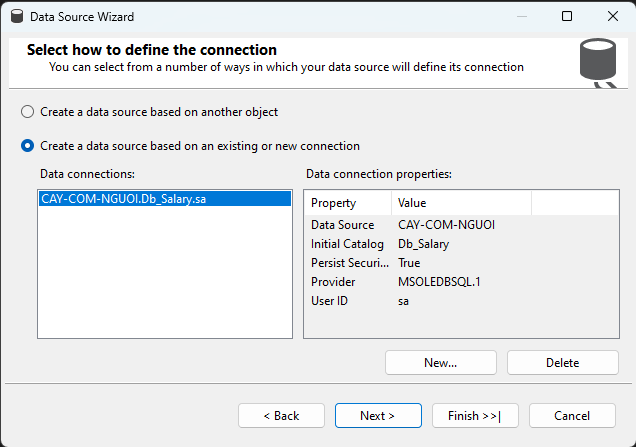
* 1. **Cấu hình SSAS**
     1. **Khởi tạo project SSAS**

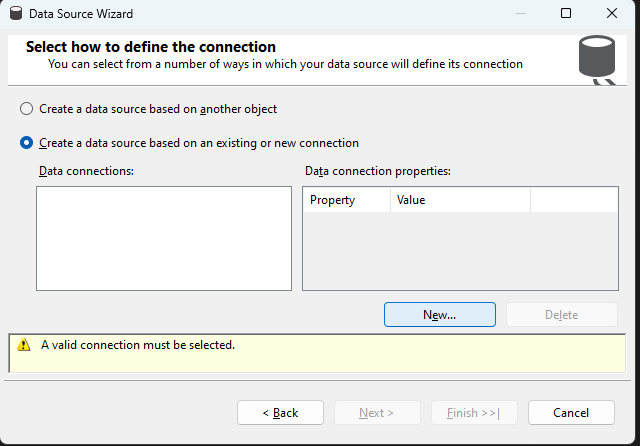
****

**3.1.2 Tạo Data sources, Data source views, Cubes**

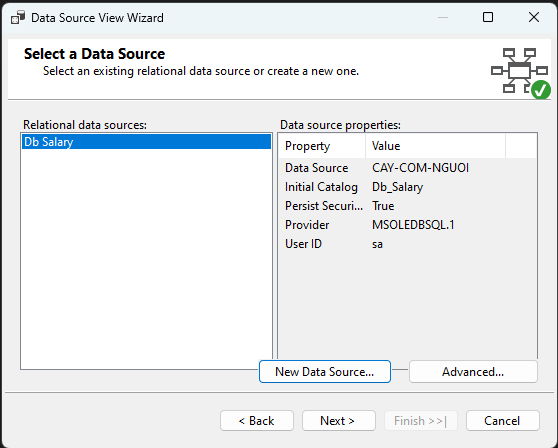
* **Data sources**

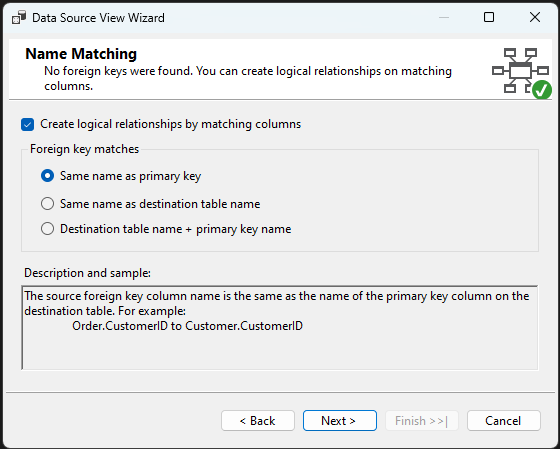
****

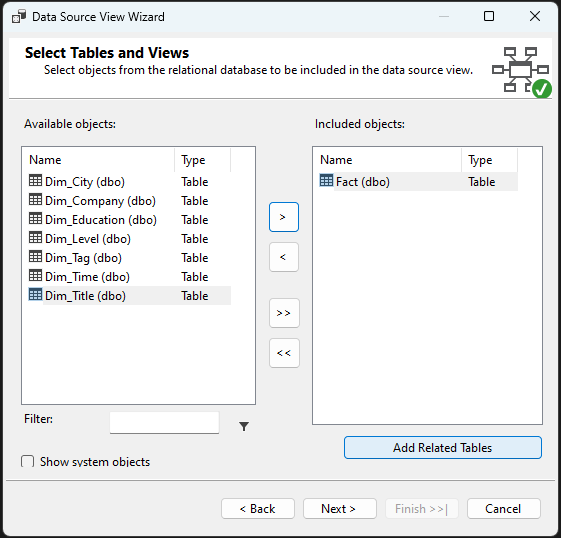
****

****

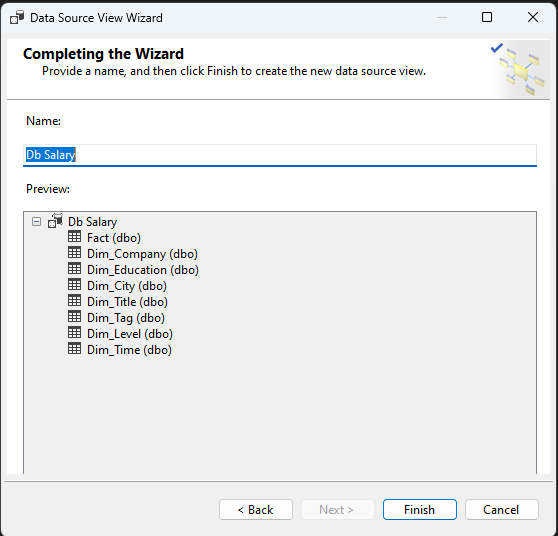
* **Data source views**

****

****

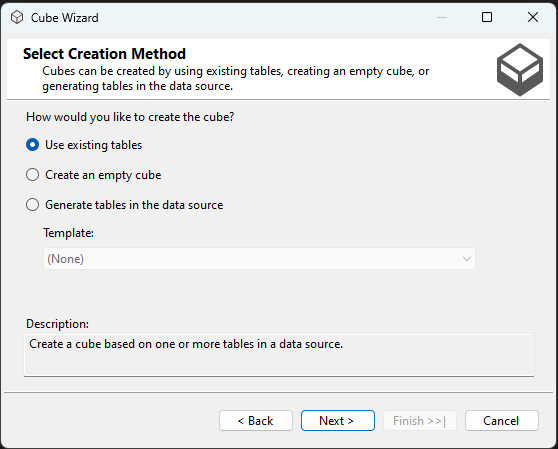
****

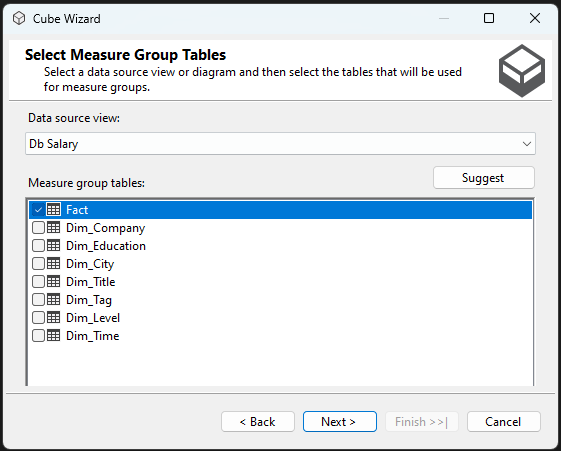
Chọn “Add Related Tables” sau đó chọn Next

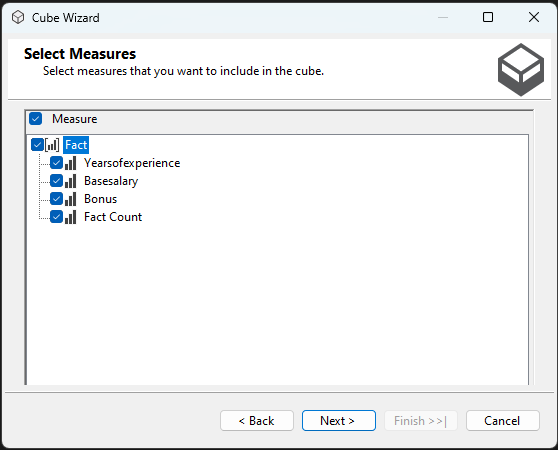


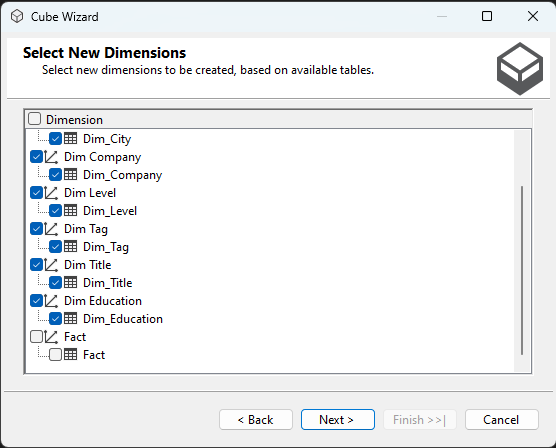
Chọn Finish

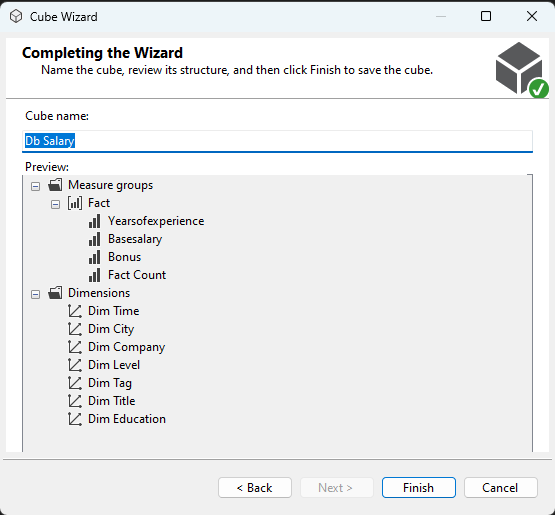
* **Cubes**

****

****

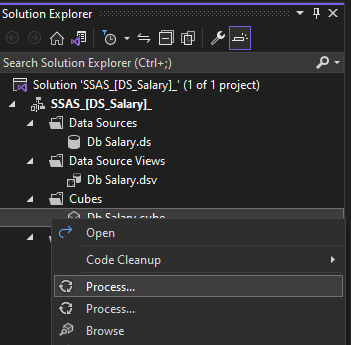
****

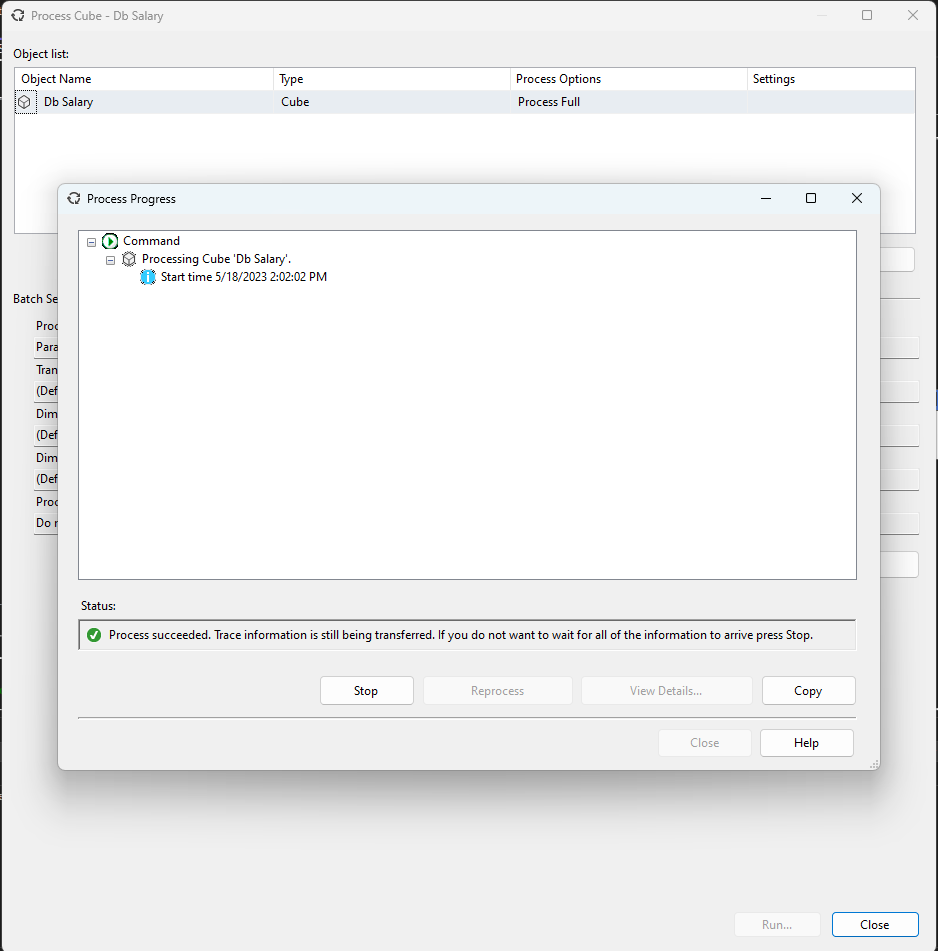
****

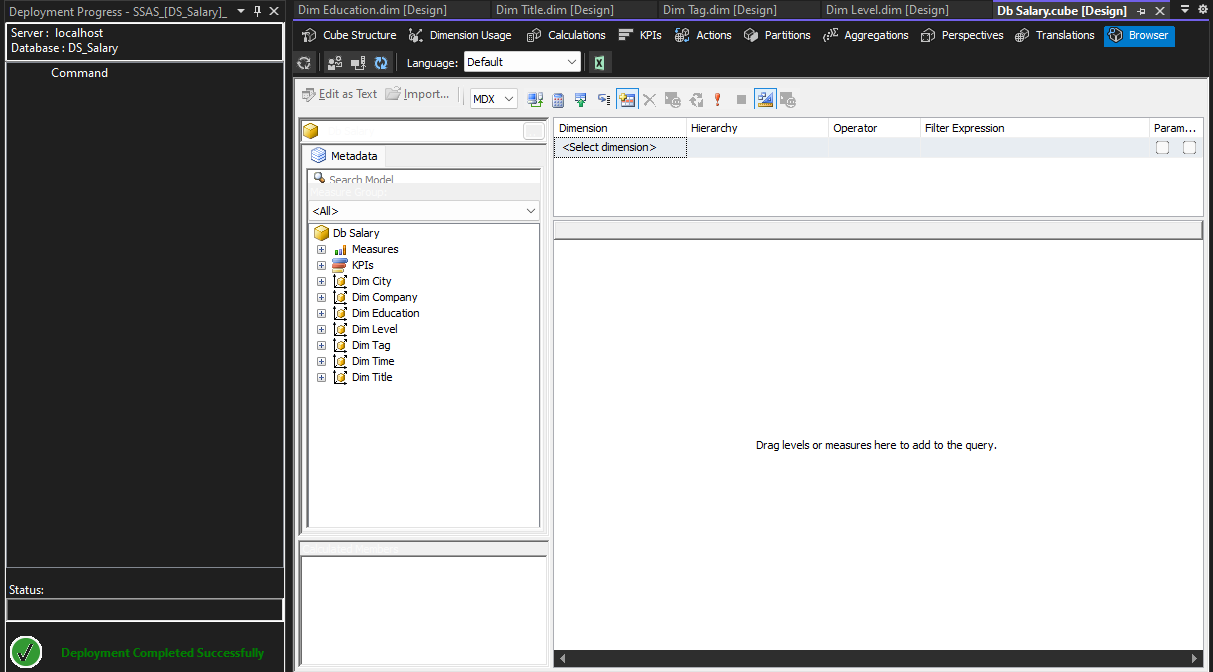
****

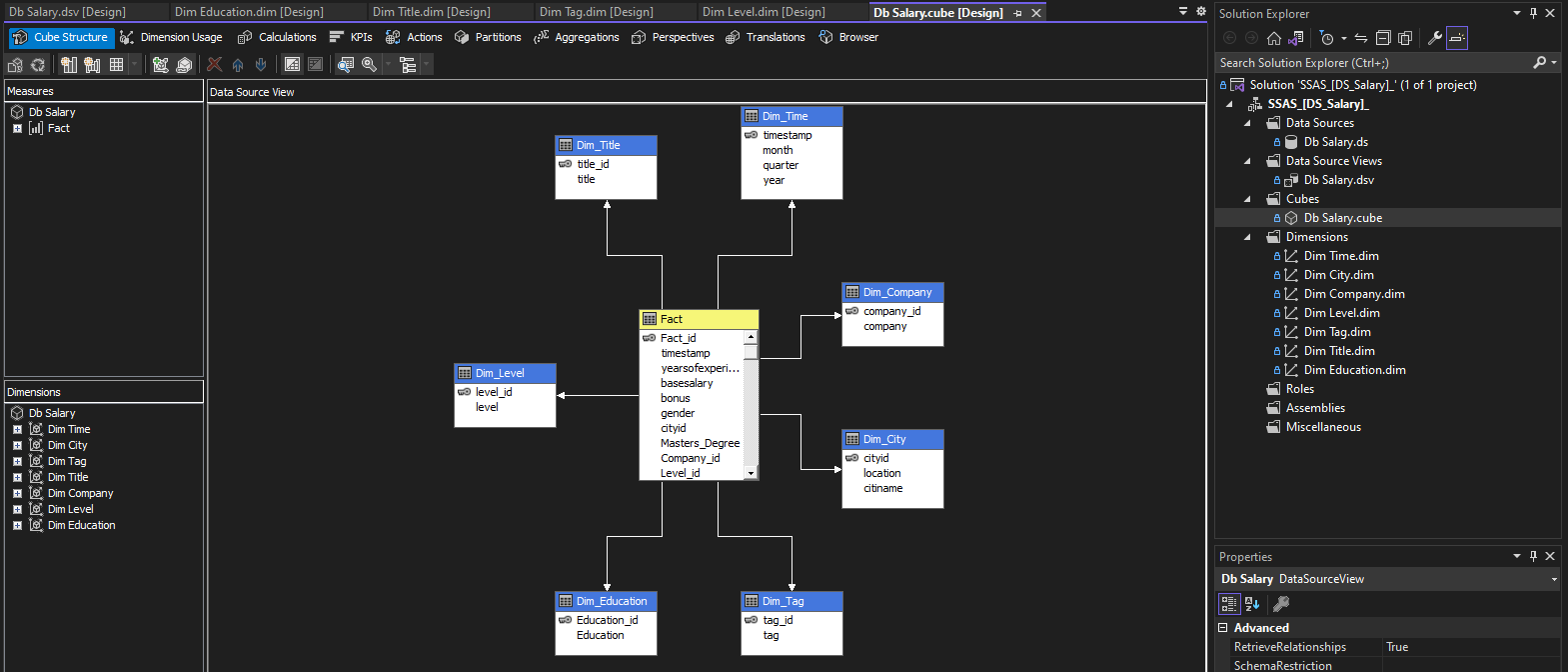
Chọn Finish

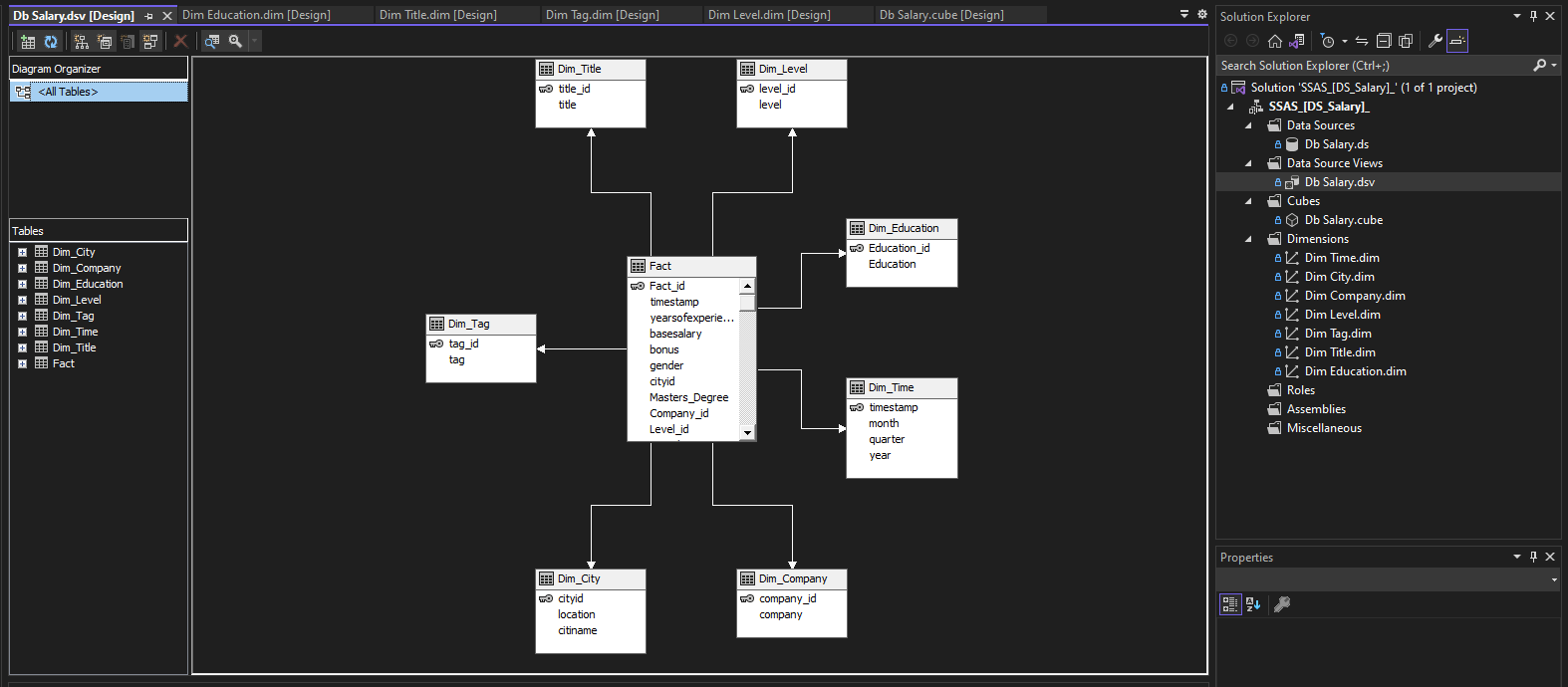
Sau khi đã tạo xong Cube ta chuột phải vào Cube đã tạo, sau đó chọn process



****

****

****

****

* 1. **Quá trình phân tích dữ liệu bằng ngôn ngữ MDX**
     1. **Lấy thông tin về các vị trí công việc có mức lương trên 200,000**
* MDX Query

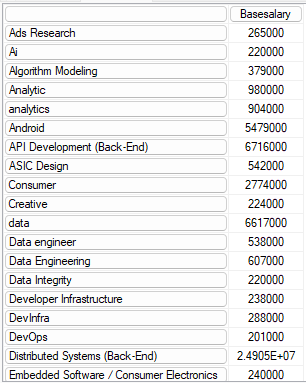
SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY (FILTER ([Dim Tag].[Tag].CHILDREN,[Measures].[Basesalary]>200000)) ON ROWS

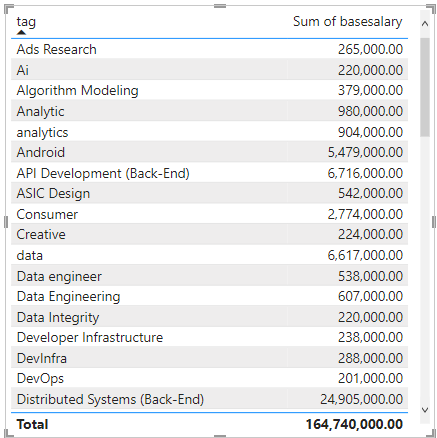
FROM [Db Salary]

WHERE [Dim Company].[Company].&[Facebook];

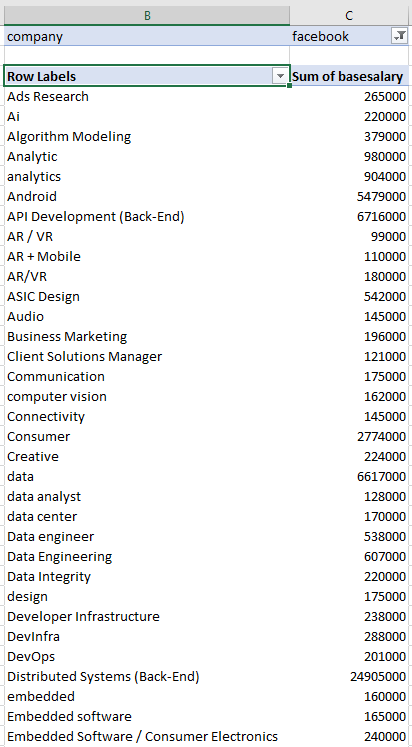
* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Tìm kiếm các vị trí công việc có mức lương trong nằm trong TOP15 ở thành phố New York**
* MDX query

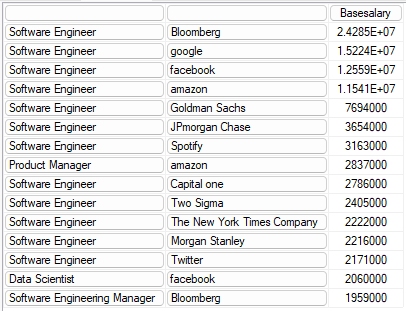
SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY (TOPCOUNT(({[Dim Title].[Title].CHILDREN},{[Dim Company].[Company].CHILDREN}),15,[Measures].[Basesalary])) ON ROWS

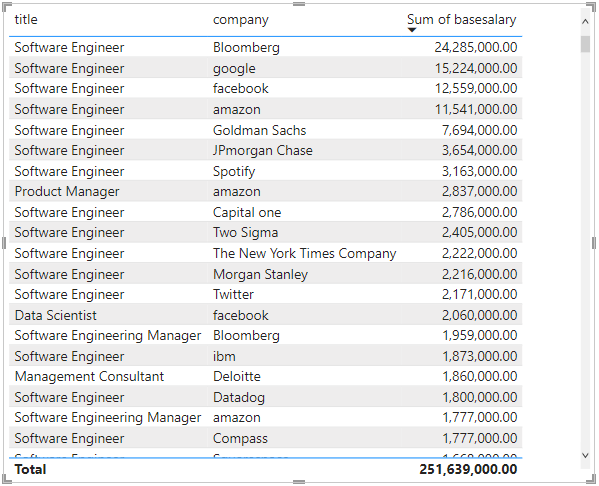
FROM [Db Salary]

WHERE [Dim City].[Location].&[New york];

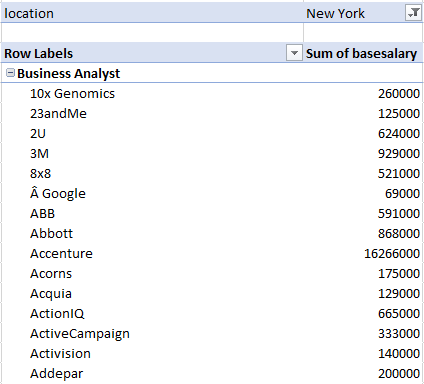
* Kết quả

****

* PowerBI

****

* Excel

****

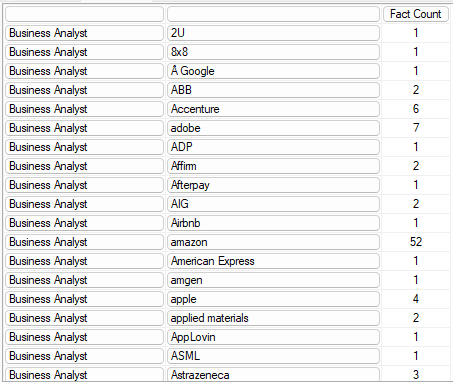
* + 1. **Đếm số nhân sự ở từng vai trò của từng công ty**
* MDX Query

SELECT NON EMPTY [Measures].[Fact Count] ON COLUMNS,

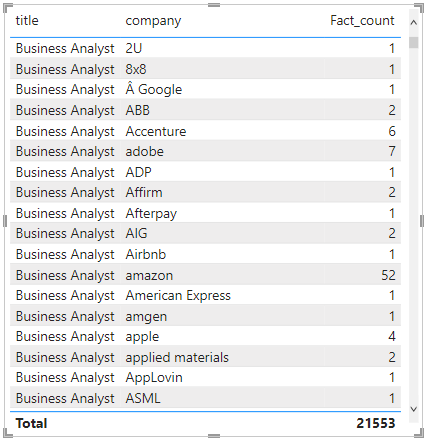
NON EMPTY ([Dim Title].[Title].CHILDREN,[Dim Company].[Company].CHILDREN) ON ROWS

FROM [Db Salary];

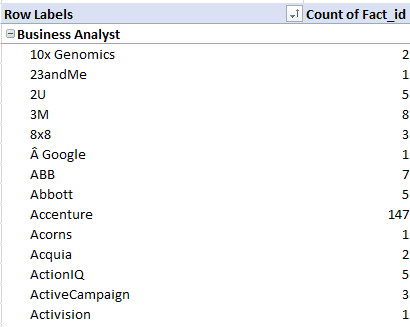
* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Đếm số vị trí có title là Software Engineer mà công ty trả lương theo từng năm, từng tháng**
* MDX Query

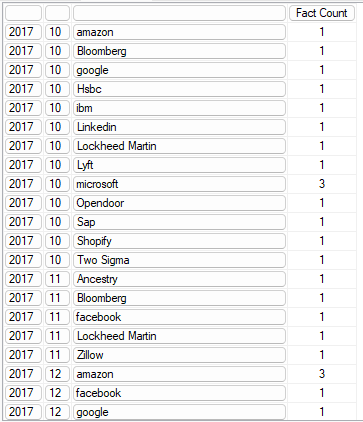
SELECT NON EMPTY [Measures].[Fact Count] ON COLUMNS,

NON EMPTY ([Dim Time].[Year].CHILDREN,[Dim Time].[Month].CHILDREN,[Dim Company].[Company].CHILDREN) ON ROWS

FROM [Db Salary]

WHERE [Dim Title].[Title].&[Software Engineer];

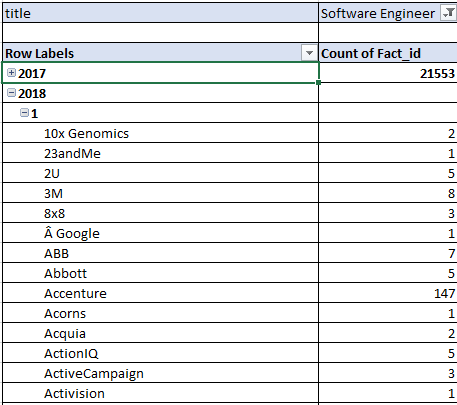
* Kết quả



* PowerBI



* Excel



* + 1. **Tại thành phố London thống kê xem top 5 công ty trả lương cao nhất**
* MDX Query

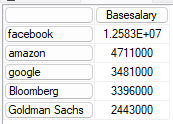
SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY (TOPCOUNT([Dim Company].[Company].CHILDREN,5,[Measures].[Basesalary])) ON ROWS

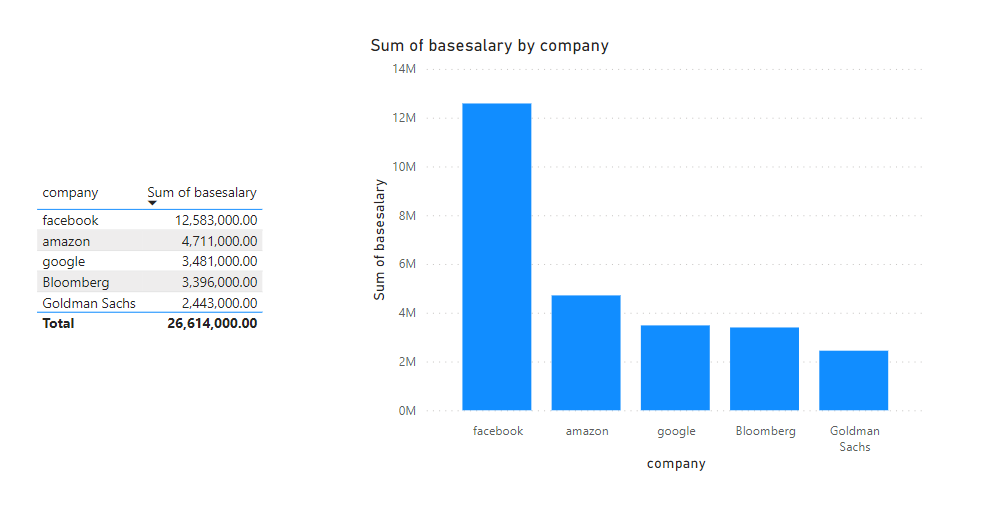
FROM [Db Salary]

WHERE [Dim City].[Location].&[London]

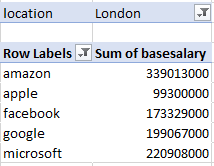
* Kết quả



* PowerBI



* Excel



* + 1. **Thống kê số vị trí theo từng level của vai trò Data Scientist của từng công ty**
* QUERY MDX

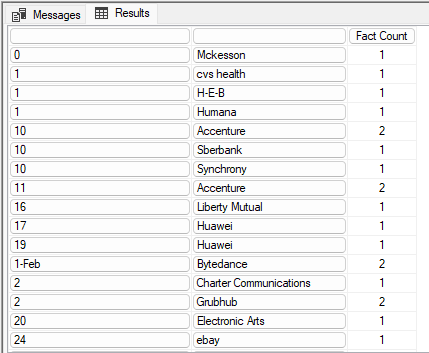
SELECT NON EMPTY [Measures].[Fact Count] ON COLUMNS,

NON EMPTY ([Dim Level].[Level].CHILDREN AS [LEVEL],[Dim Company].[Company].CHILDREN AS [COMPANY]) ON ROWS

FROM [Db Salary]

WHERE [Dim Title].[Title].&[Data Scientist]

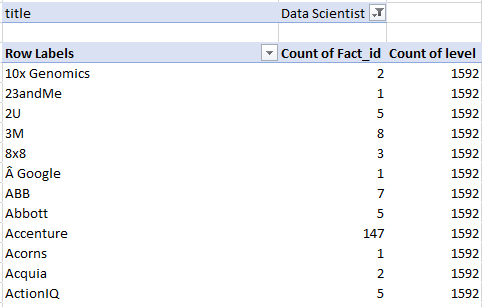
* Kết quả



* PowerBI



* Excel



* + 1. **Thống kê mức lương cơ bản theo từng vị trí mà Google đã trả cho nhân viên**
* Query MDX

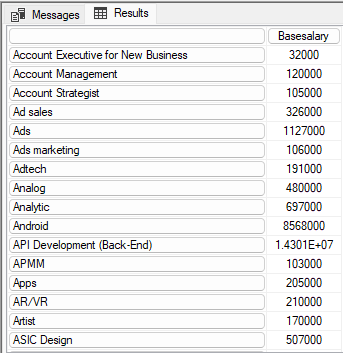
SELECT NON EMPTY [Measures].[Basesalary] AS [Base Salary] ON COLUMNS,

NON EMPTY ([Dim Tag].[Tag].CHILDREN AS [Position]) ON ROWS

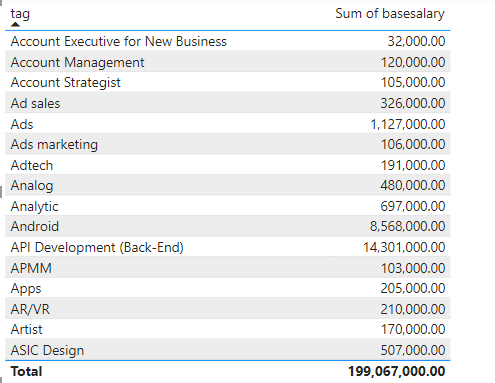
FROM [Db Salary]

WHERE [Dim Company].[Company].&[Google]

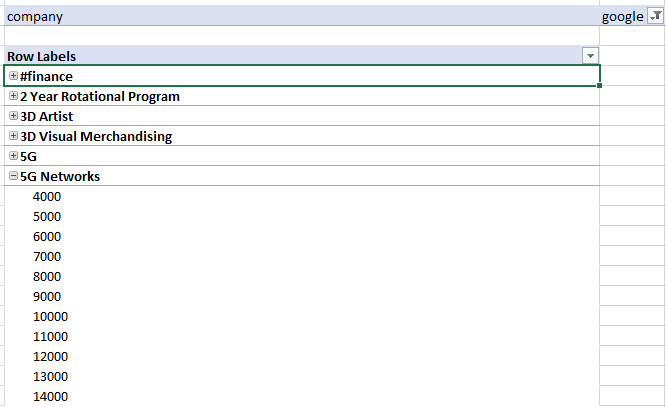
* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Thống kê level, mức lương giảm dần vị trí full stack, của công ty Facebook ở thành phố San francisco**
* Query MDX

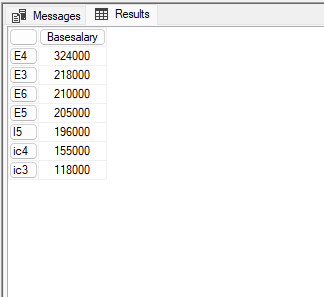
SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY ORDER([Dim Level].[Level].CHILDREN,[Measures].[Basesalary], DESC) ON ROWS

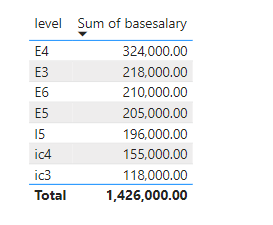
FROM [Db Salary]

WHERE ({[Dim Company].[Company].&[Facebook]},{[Dim Tag].[Tag].&[full stack]},{[Dim City].[Location].&[San Francisco]});

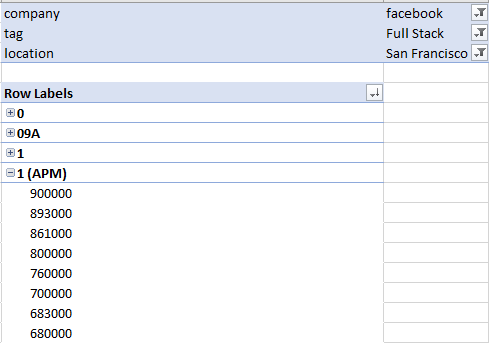
* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Thống kê mức lương cơ bản cho những vị trí học vấn tiến sĩ, giới tính nữ ở công ty Microsoft**
* MDX Query

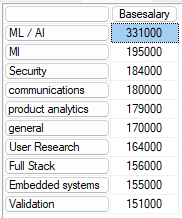
SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY ORDER([Dim Tag].[Tag].CHILDREN,[Measures].[Basesalary], DESC) ON ROWS

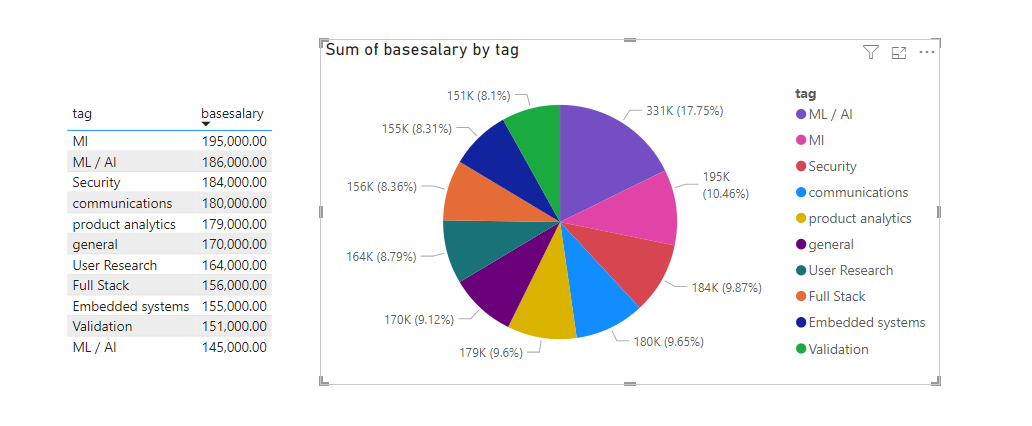
FROM [Db Salary]

WHERE ({[Dim Company].[Company].&[Microsoft]},{[Fact].[Gender].&[Female]},{[Dim Education].[Education].&[PhD]});

* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Liệt kê top 3 vị trí có số năm kinh nghiệm lớn nhất của vai trò Data Scientist và mức lương của vị trí đó trong quý 2 năm 2020**
* MDX Query

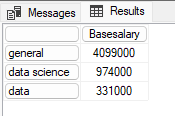
SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY TOPCOUNT ([Dim Tag].[Tag].CHILDREN, 3, [Measures].[Yearsofexperience]) ON ROWS

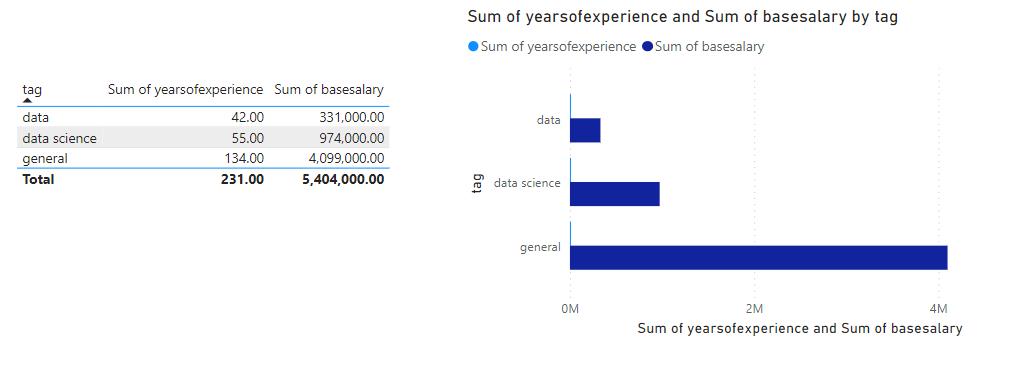
FROM [Db Salary]

WHERE ( {[Dim Title].[Title].&[Data Scientist]}, {[Dim Time].[Quarter].&[2]}, {[Dim Time].[Year].&[2020]});

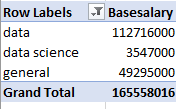
* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Thống kê tổng tiền thưởng theo mỗi vai trò của từng công ty trong quý 2 năm 2020**
* MDX Query

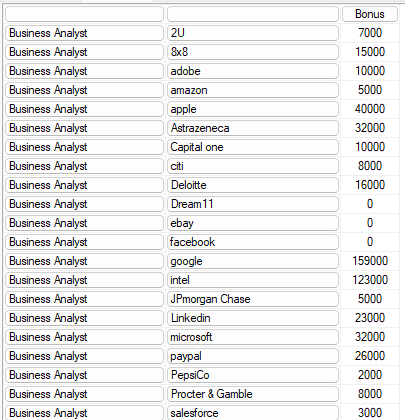
SELECT NON EMPTY [Measures].[Bonus] ON COLUMNS,

NON EMPTY ([Dim Title].[Title].CHILDREN,[Dim Company].[Company].CHILDREN) ON ROWS

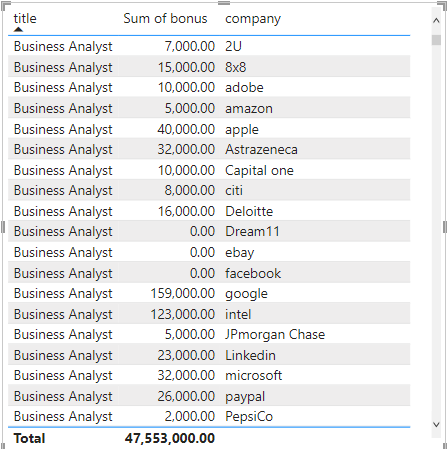
FROM [Db Salary]

WHERE ( {[Dim Time].[Quarter].&[2]}, {[Dim Time].[Year].&[2020]});

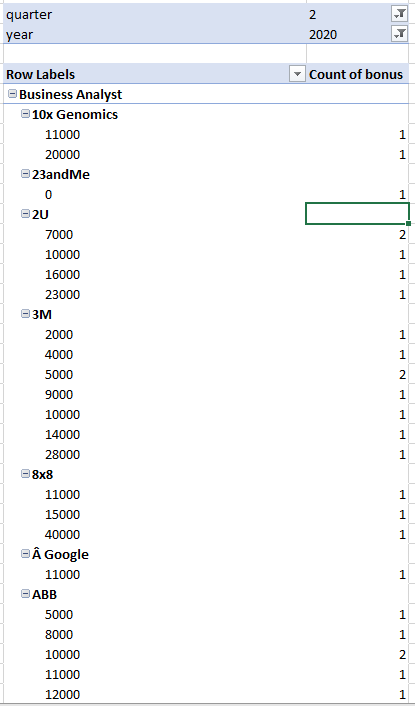
* Kết Quả

****

* PowerBI

****

* Excel

****

* + 1. **Thống kê vị trí có số lương cơ bản cao nhất trong năm 2020 của công ty Microsoft**
* MDX Query

SELECT NON EMPTY [Measures].[Basesalary] ON COLUMNS,

NON EMPTY TOPCOUNT ([Dim Tag].[Tag].CHILDREN, 1,[Measures].[Basesalary] ) ON ROWS

FROM [Db Salary]

WHERE ({[Dim Time].[Year].&[2020]},{[Dim Company].[Company].&[Microsoft]});

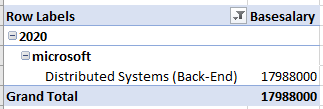
* Kết quả

****

* PowerBI

****

* Excel

****

* + 1. **Thống kê số chi nhánh của công ty Facebook ở trong các thành phố**
* MDX Query

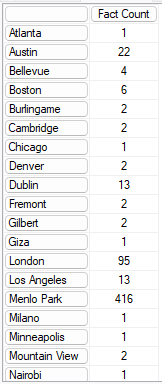
SELECT NON EMPTY [Measures].[Fact Count] ON COLUMNS,

NON EMPTY [Dim City].[Location].CHILDREN ON ROWS

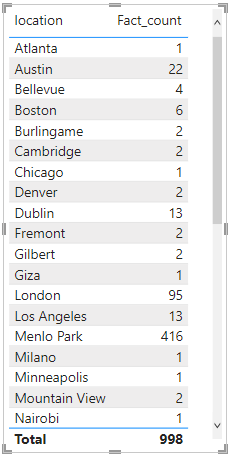
FROM [Db Salary]

WHERE ({[Dim Company].[Company].&[Facebook]});

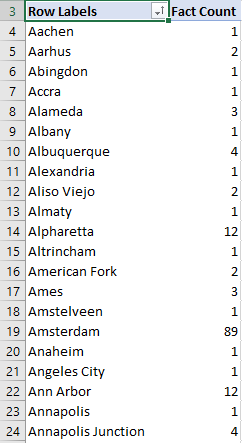
* Kết quả



* PowerBI



* Excel



* + 1. **Liệt kê top 10 công ty có nhiều Product Manager có giới tính nữ nhất trong năm 2019**
* MDX Query

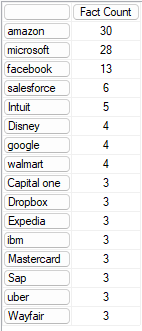
SELECT NON EMPTY [Measures].[Fact Count] ON COLUMNS,

NON EMPTY TOPCOUNT ([Dim Company].[Company].CHILDREN, 10,[Measures].[Fact Count] ) ON ROWS

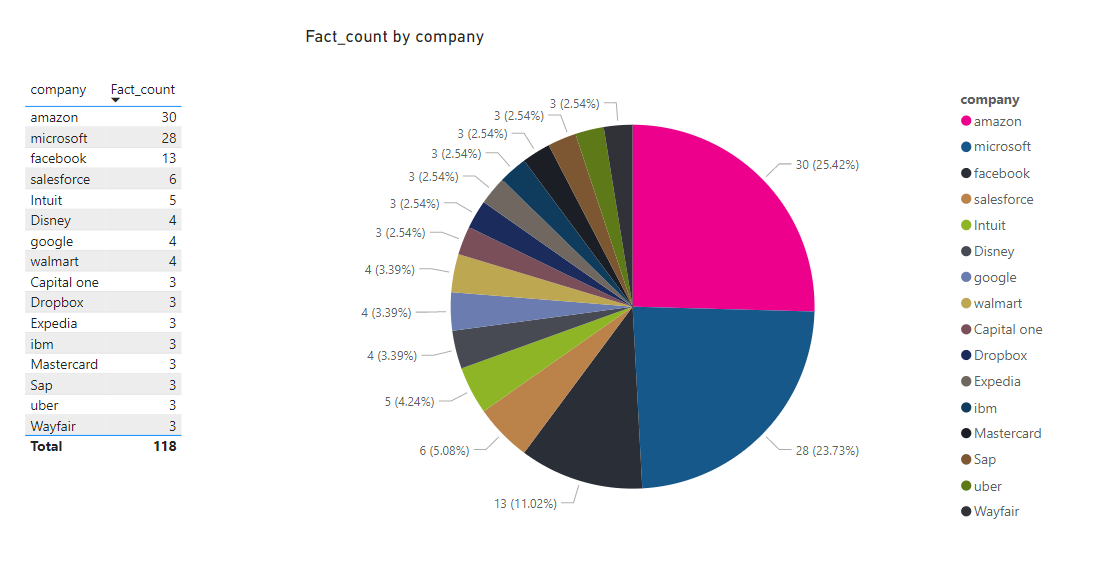
FROM [Db Salary]

WHERE ({[Dim Time].[Year].&[2019]},{[Fact].[Gender].&[Female]},{[Dim Title].[Title].&[Product Manager]});

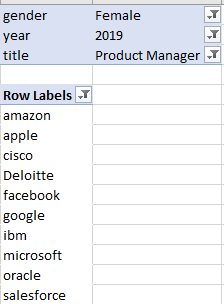
* Kết quả

****

* PowerBI

****

* Excel

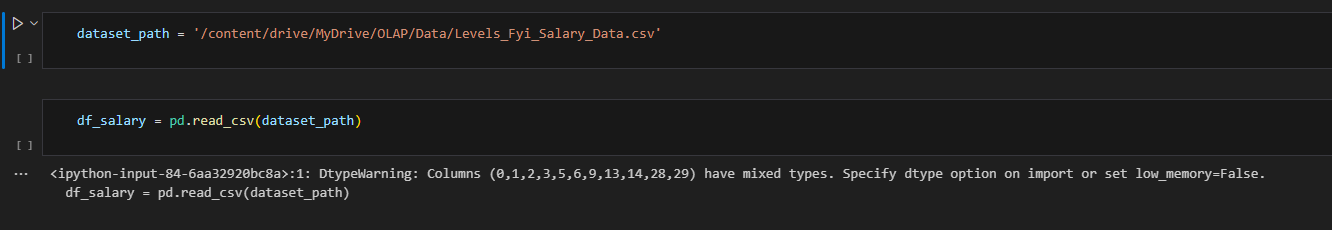
****

# **CHƯƠNG 4: KHAI THÁC DỮ LIỆU – QUÁ TRÌNH DATAMINING**

## **4.1 PREPROCESSING**

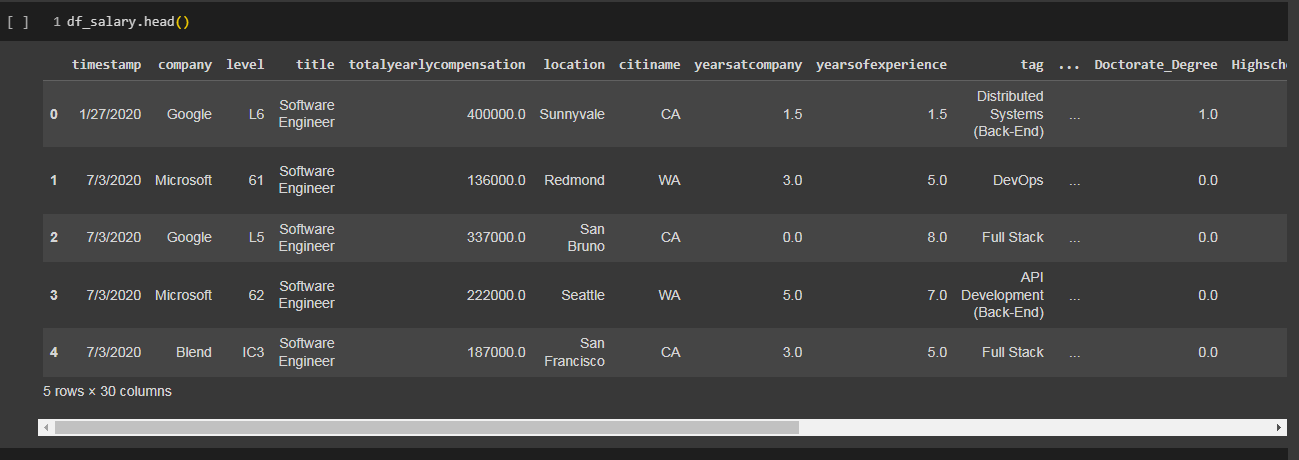
* Load data

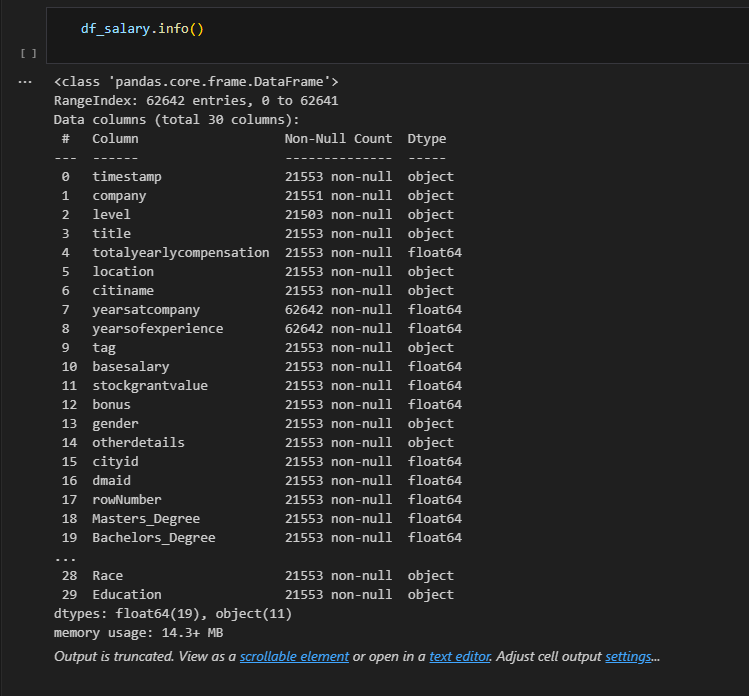
Đầu tiên, để tiến hành khai thác dữ liệu (data mining), chúng ta dùng thư viện pandas đọc file dữ liệu csv từ drive.



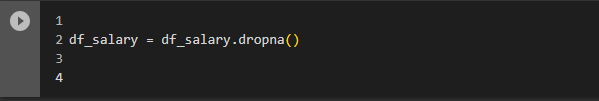
* Thực hiện kiểm tra dữ liệu

Sau đó, tiến hành in dữ liệu để kiểm tra sơ bộ các thuộc tính.



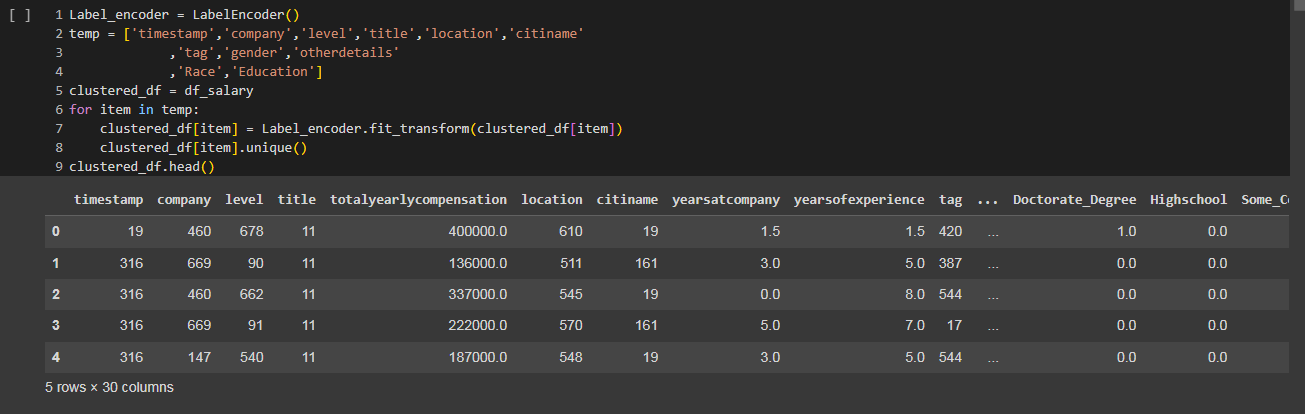


Trong bộ dữ liệu, không tránh khỏi những ô bị dữ liệu thiếu. Tiếp theo cần xử lý những record bị thiếu hoặc null hoặc NaN



* Gán nhãn dữ liệu

Sau khi kiểm tra dữ liệu, ta nhận thấy có một vài thuộc tính được dùng ở dạng chuỗi. Do đó, chúng ta phải sử dụng Label encoding chuyển đổi các giá trị chuỗi thành các giá trị số nguyên. Khi làm việc với các mô hình máy học, chúng ta cần đưa dữ liệu vào dưới dạng số để thuật toán có thể hiểu và xử lý các giá trị này.

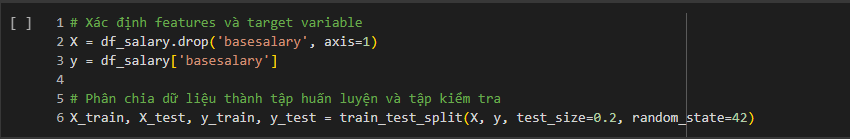


**4.2 Sử dụng Random Forest chọn các thuộc tính ảnh hưởng**

* Chọn các thuộc tính X, y, chia tập X\_train, y\_train để sử dụng Random Forest lựa chọn các thuộc tính quan trọng.

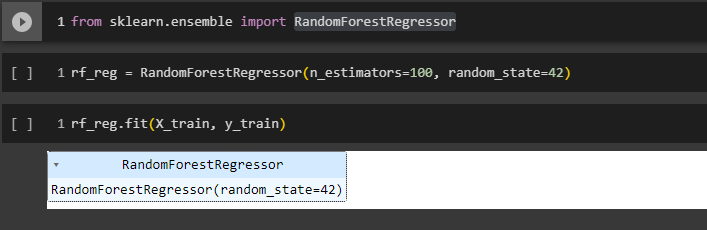
Ở bài này, chúng em sẽ dự đoán giá trị “basesalary” từ các thuộc tính liên quan. Tuy nhiên, dữ liệu này có nhiều các thuộc tính khác nhau, rất khó để lựa chọn những features nào có ảnh hưởng nhiều nhất đến giá trị dự đoán là output của chúng ta. Do đó, sẽ sử dụng RandomForest và hàm features\_importance để tính toán điểm số, mức độ quan trọng của các features.

Để sử dụng được Random forest, chúng ta phải có các tập X,y. Với X hiện tại là toàn bộ thuộc tính có trong tập dữ liệu, y là giá trị “basesalary”. Tiếp tục chia tập train/test theo test\_size = 0.2, tức tập test chiếm 20%, tập train là 80%.



* Training model RandomForest.

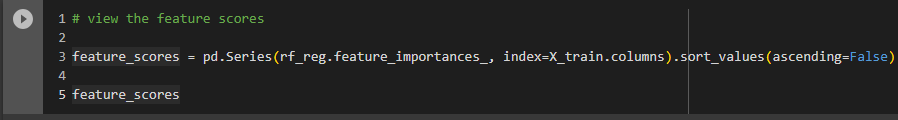
Sau khi chia được tập X\_train, X\_test, y\_train, y\_test. Ta sử dụng RandomForestRegressor từ thư viện sklearn với cái tham số n\_estimator = 1000, random\_state = 42 để traning model.



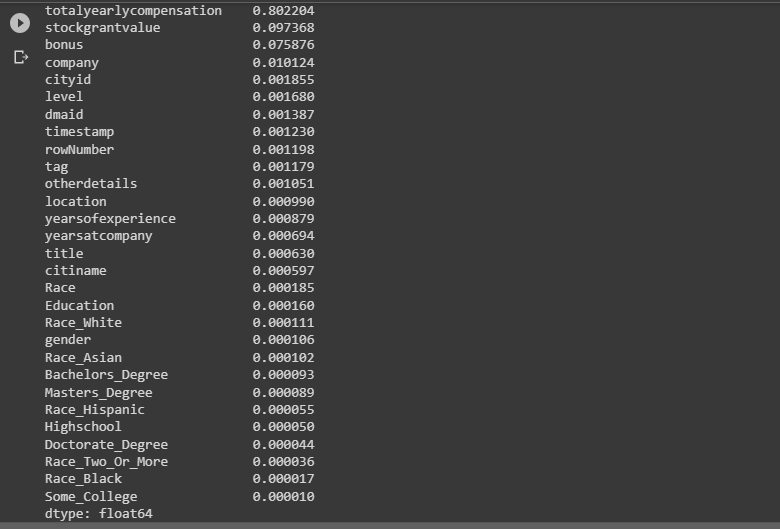
* Tính toán và hiển thị độ quan trọng của các thuộc tính trong mô hình.

Trong phần này, chúng ta sử dụng feature\_importances để tính toán và hiển thị được mức độ quan trọng của các features. Cụ thể, nó tính toán mức độ đóng góp của từng feature trong việc dự đoán “basesalary” và sắp xếp theo thứ tự giảm dần của mức độ quan trọng.

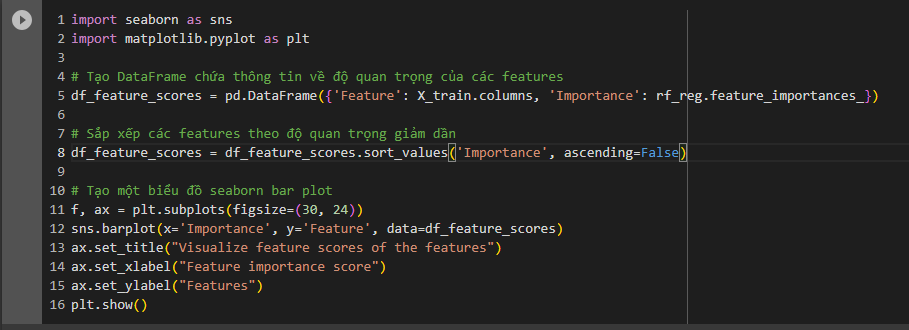
Đễ dễ hiểu hơn, với mô hình Random Forest, độ quan trọng của mỗi đặc trưng được tính bằng cách tính toán tổng số lần một đặc trưng được sử dụng để chia nhánh trong các cây quyết định trong rừng. Các đặc trưng có đóng góp cao hơn sẽ được xem là quan trọng hơn trong việc dự đoán kết quả của mô hình.



Kết quả thu được như sau:



Đễ dễ dàng chọn ra những thuộc tính quan trọng, chúng em sẽ trực quan hóa độ quan trọng của các đặc trưng trong mô hình học máy bằng cách tạo một biểu đồ thanh.



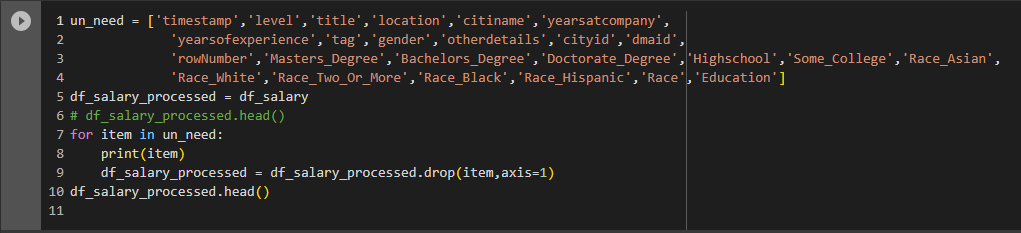
Vì kích thước chữ khá nhỏ, chúng em chỉ mô phỏng và chụp một vài thuộc tính có mức độ ảnh hưởng cao nhất, các thuộc tính bên dưới có mức độ ảnh hưởng không đáng kể.



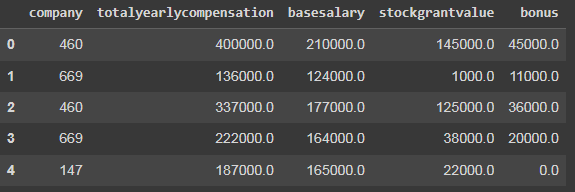
Biểu đồ thanh sẽ hiển thị tên của các đặc trưng trên trục y và độ quan trọng của chúng trên trục x. Các thanh được sắp xếp theo thứ tự giảm dần của độ quan trọng để giúp người dùng dễ dàng nhìn thấy đặc trưng quan trọng nhất. Nhìn vào biểu đồ trên, ta dễ dàng phân tích được: “Thuộc tính quan trọng nhất là **totalyearlycompensation** với features\_score là 0,802204. Các thuộc tính quan trọng kế tiếp là **stockgrantvalue 0.097368, bonus 0.075876, company 0.010124.**

* Ta sẽ chọn 4 features có mức ảnh hưởng nhất để tiến hành dự đoán “basesalary” là : *company, totalyearlycompensation, stockgrantvalue, bonus.*
* Loại bỏ các thuộc tính có mức độ ảnh hưởng thấp ra khỏi tập dữ liệu.

Sau khi có được 4 thuộc tính quan trọng, ta sẽ loại bỏ, xóa các thuộc tính khác trong tập dữ liệu để dễ dàng xử lý.

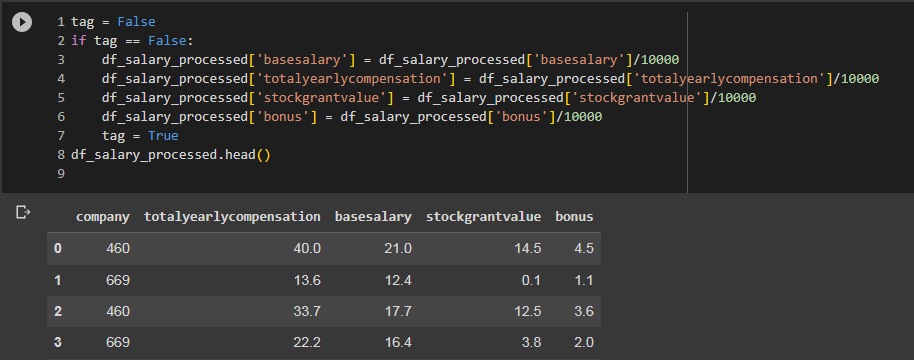


Bộ dữ liệu sau khi xử lý thu được:



* Scale lại giá trị của dữ liệu.

Scale lại giá trị basesalary, totalyearlycompensation, stockgrantvalue bonus cho phù hợp khi chia các giá trị cho 10000. Ta được kết quả như sau.

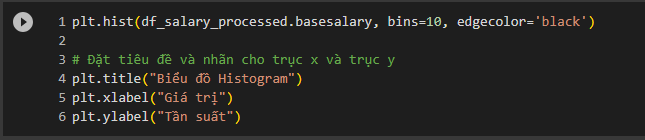


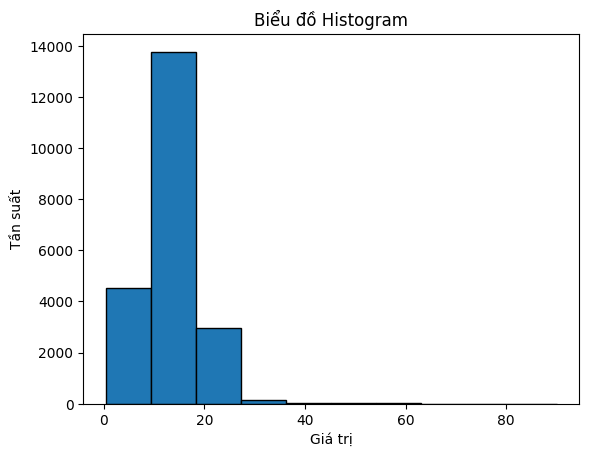
**4.3 Trực quan dữ liệu**

Để tiến hành khai thác dữ liệu, chúng ta phải hiểu rõ về dữ liệu và tiến hành trực quan chúng.

* Biểu đồ histogram

Để kiểm ra mức lương thường nằm trong giá trị bao nhiêu, xuất hiện bao nhiêu lần thì ta sẽ plot biểu đồ histogram – là biểu đồ tần suất giá trị.





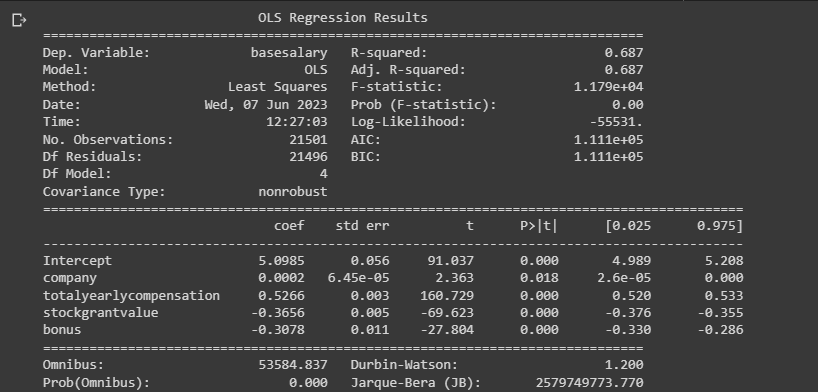
Phân tích kết quả biểu đồ : Ta thấy mức lương mà nhiều ngược có được nhất nằm ở khoảng từ 8-18, tiếp theo là <8. Số lượng người có mức lương tầm 20 cũng rơi vào khoảng gần 3000 người. Các mức lương còn lại như 40 60 v..v chiếm số lượng rất nhỏ.

* Phương trình hồi quy.

Chúng em sử dụng ols để tìm ra được phương trình hồi quy của “basesalary” từ các thuộc tính trên. Nói rõ hơn, Phương trình hồi quy được sử dụng để mô hình hóa mối quan hệ giữa biến đầu vào và biến đầu ra trong một tập dữ liệu. Nó cho phép dự đoán giá trị của biến đầu ra dựa trên giá trị của biến đầu vào, và có thể được sử dụng để giải quyết các vấn đề trong lĩnh vực dự đoán và phân tích dữ liệu.

Các tham số hồi quy được ước tính bằng các phương pháp định lượng, như phương pháp bình phương tối thiểu, để tối ưu hóa độ chính xác của phương trình hồi quy.





Nhìn vào bảng kết quả khi sử dụng OLS Regression, ta có thể dễ dàng có được phương trình hồi quy của “basesalary” từ 4 thuộc tính quan trọng trên như sau:

**Basesalary = 0,0002\*company + 0,5266\*totalyearcompensation + 0,3656\*stockgrantvalue – 0,3078\*bonus**

Từ phương trình trên, khi có bất kỳ giá trị company, totalyearcompensation, stockgrantvalue, bonus chúng ta có thể dự đoán được basesalary.

**4.4 Data mining**

**4.4.1 Phân cụm dữ liệu bằng Kmeans**

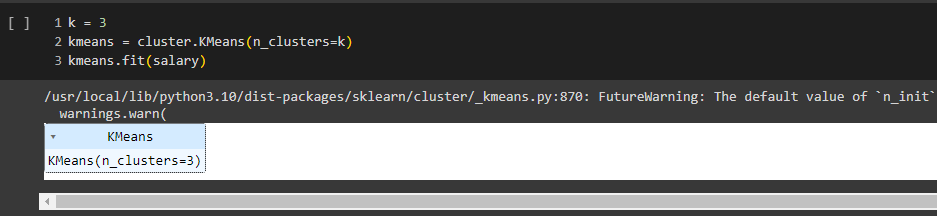
Trong phần này, chúng em tiếp tục sử dụng các thuộc tính đã chọn ở phần trước để tiến hành phân cụm dữ liệu. Các thuộc tính đó bao gồm: company, totalyearlycompensation, basesalary, stockgrantvalue, bonus.

* Sử dụng Kmeans từ thư viện sklearn để phân cụm dữ liệu.

Phân cụm bằng kmeans là một phương pháp trong học không giám sát (unsupervised learning) để phân nhóm các điểm dữ liệu vào các cụm dữ liệu khác nhau. Phương pháp này được sử dụng rộng rãi trong các ứng dụng khai thác dữ liệu và máy học.

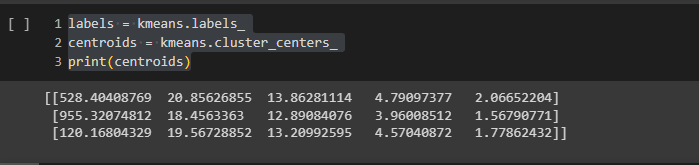
Phương pháp kmeans hoạt động bằng cách chọn ngẫu nhiên k điểm dữ liệu ban đầu để đại diện cho các cụm dữ liệu. Sau đó, mỗi điểm dữ liệu được gán vào cụm gần nhất với nó dựa trên khoảng cách Euclid giữa điểm dữ liệu và các điểm đại diện của các cụm dữ liệu. Sau đó, các điểm đại diện của các cụm được cập nhật bằng cách tính trung bình của tất cả các điểm thuộc cụm đó. Quá trình này được lặp đi lặp lại cho đến khi không còn sự thay đổi nào trong việc phân cụm.

Nhóm chúng em tiến hành chọn k = 3, tức sẽ phân cụm dữ liệu thành 3 nhóm.



* Lấy ra các nhãn (labels) và các điểm trung tâm của các cụm dữ liệu (centroids).

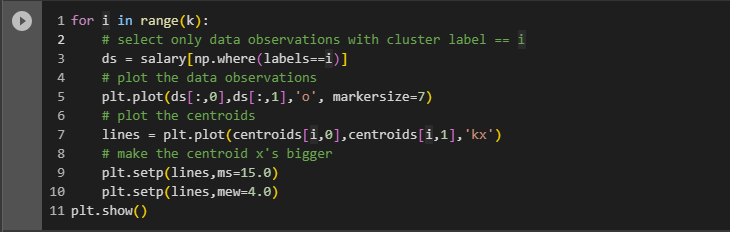
Cụ thể, kmeans.labels\_ là một mảng chứa các nhãn của các điểm dữ liệu, tức là nhãn của cụm mà mỗi điểm dữ liệu đã được phân vào kmeans.cluster\_centers\_ là một mảng chứa tọa độ của các điểm trung tâm của các cụm dữ liệu, tức là các điểm được chọn để đại diện cho mỗi cụm.

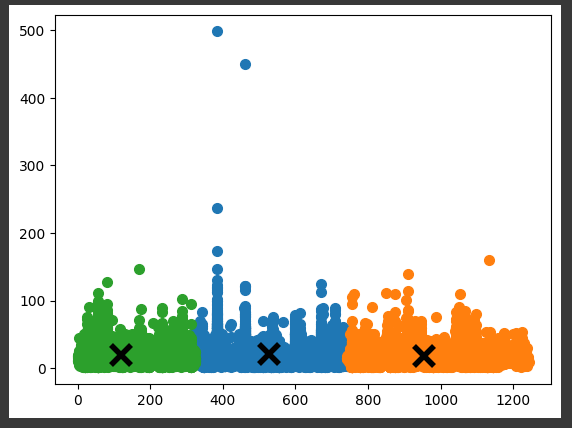


Thứ tự của các thuộc tính trong mảng: company, totalyearlycompensation, basesalary, stockgrantvalue, bonus. Các điểm trung tâm này được chọn để đại diện cho các cụm dữ liệu. Mỗi điểm trung tâm là trung bình của tất cả các điểm dữ liệu thuộc cùng một cụm.

Các điểm trung tâm có ý nghĩa quan trọng rong việc phân cụm vì chúng được sử dụng để tính toán khoảng cách giữa các điểm dữ liệu và các cụm dữ liệu. Khi một điểm dữ liệu mới được đưa vào, nó sẽ được gán vào cụm có điểm trung tâm gần nhất.

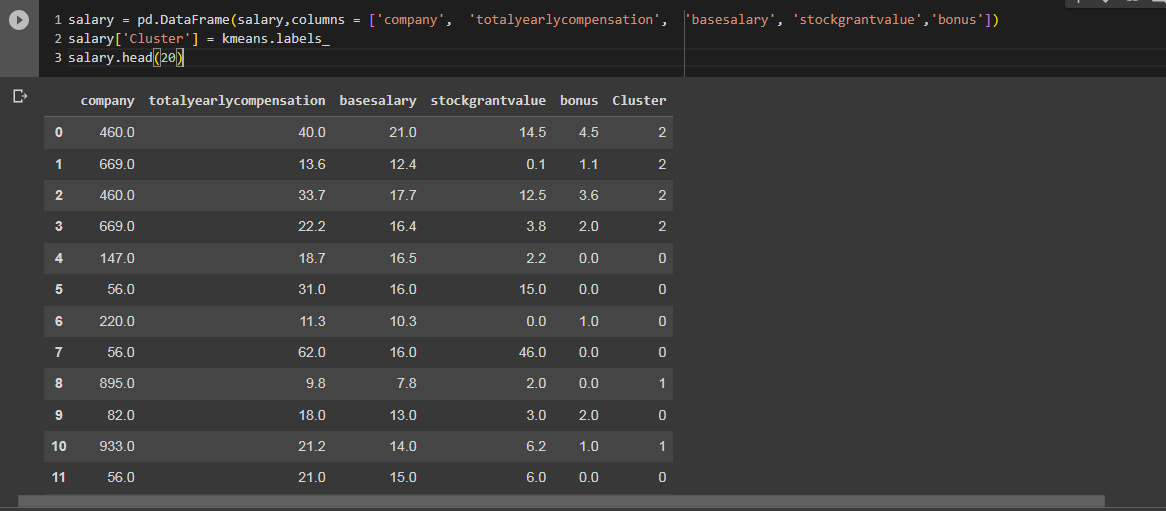
* Hiển thị biểu đồ kết quả của thuât toán Kmeans clustering.





Nhìn vào kết quả phân cụm, ta có thể nhận thấy ở cột x là company, do đó giá trị trung bình của cụm 1 là company xấp xỉ bằng 120, của cụm 2 là xấp xỉ 530 và ở cụm 3 là 955.

Thử nghiệm với bộ dữ liệu của chúng ta, thu được kết quả ở các mẫu dữ liệu như sau, với ba cụm được gán lables là 0;1;2.

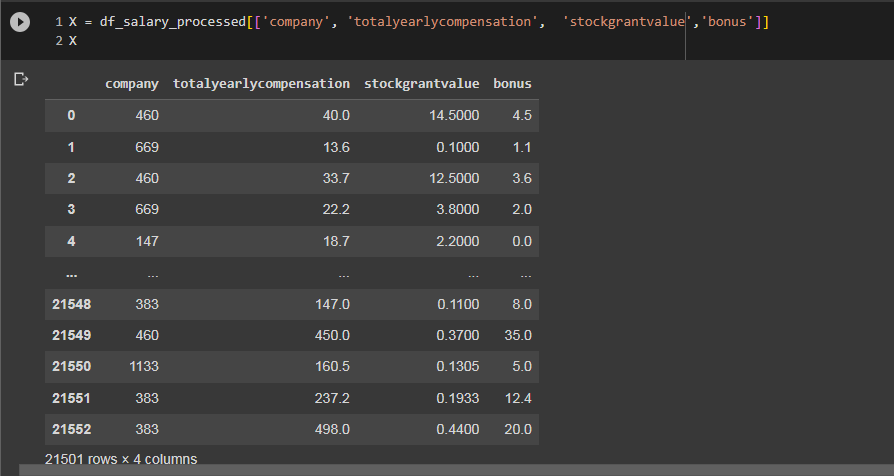


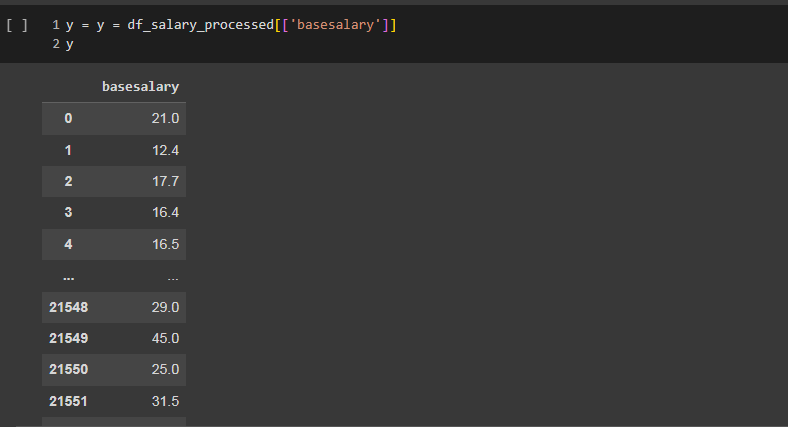
**4.4.3 Dùng các thuật toán Machine Learning dự đoán giá trị “basesalary”.**

Trong phần này, chúng em sử dụng một vài thuật toán cơ bản của Machine Learning để dự đoán giá trị “basesalary” như : Linear Regression, KNN, SVR, Random Forest. Vì cách xử lý và huấn luyện tương tự nhau, em sẽ tiến hành phân tích rõ ở mô hình Random Forest, do kết quả thu được là tốt nhất.

* Chia dữ liệu tập train/test

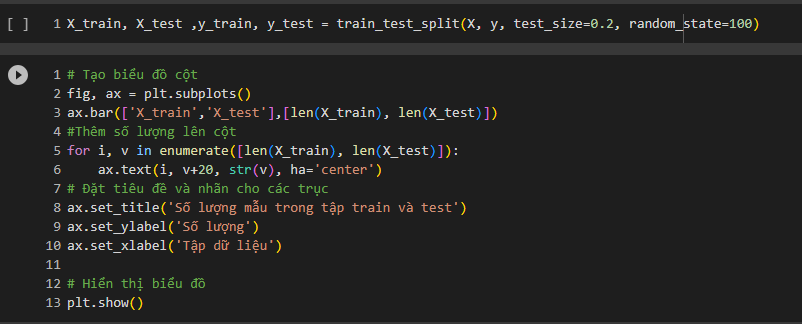
Như đã trình bày ở phần trước, ta đã chọn được 4 thuộc tính ảnh hưởng đến kết quả dự đoán của “basesalary”, do đó ta sẽ tiếp tục sử dụng 4 thuộc tính này trong bài toán dự đoán.

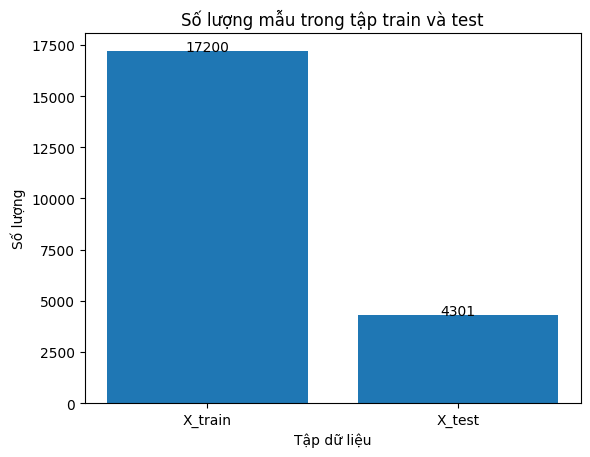




Dễ dàng nhận ra: X gồm các thuộc tính: company, totalyearlycompensation, tockgrantvalue, bonus. Y là thuộc tính giá trị basesalary.

Sau đó tiến hành chia tập train/test theo tỉ lệ 8:2. Ta thu được X\_train, X\_test, y\_train, y\_test.

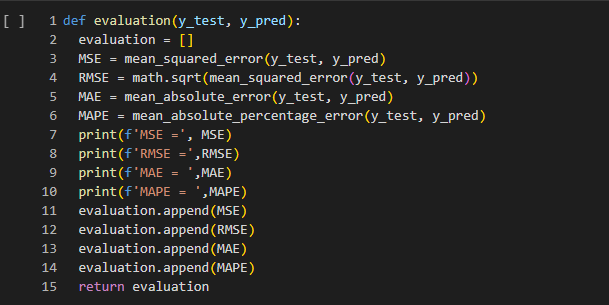


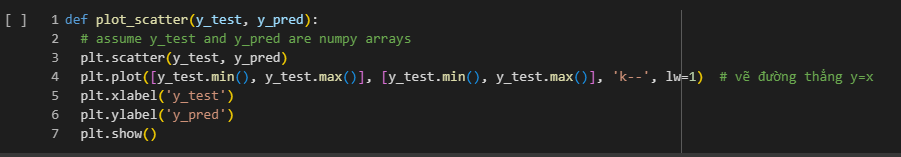


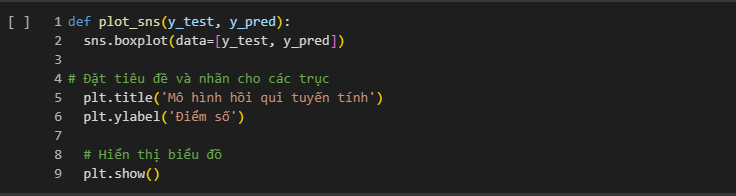
Nhìn vào biểu đồ bên trên, dễ dàng nhận ra sau khi xử lý dữ liệu và tiến hành chia tập train/test ta thu được : Số lượng mẫu dữ liệu trong tập train là 17200. Số lượng dữ liệu trong tập test là 4301.

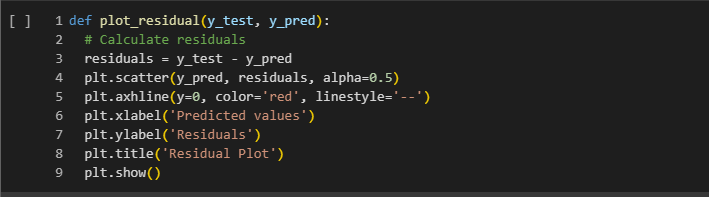
* Một số hàm hỗ trợ việc plot kết quả và phân tích dữ liệu.

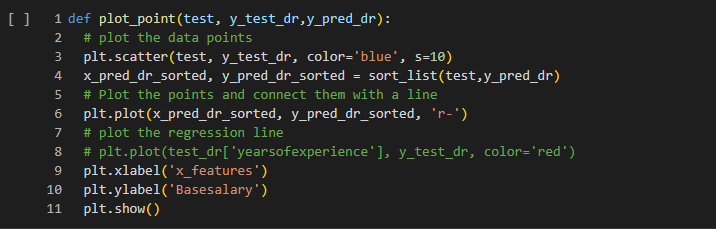
Nhóm chúng em sử dụng đến 4 thuật toán Machine learning nên khá mất thời gian nếu với mỗi mô hình, lại viết lại những hàm giống nhau. Do đó, trước khi tiến hành dự đoán, chúng ta có một số hàm hỗ trợ như sau. Ý nghĩa của các hàm sẽ được phân tích thông qua kết quả ở phần bên dưới.







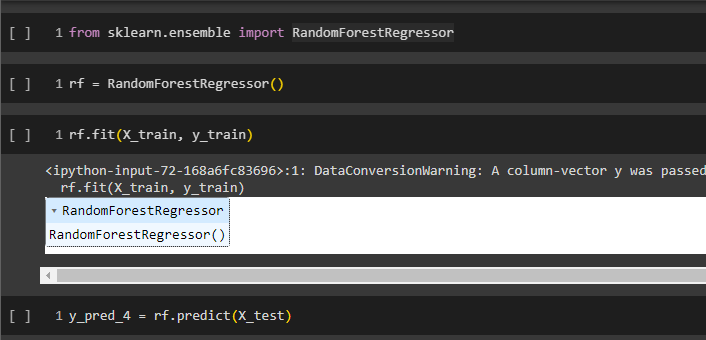




* Sử dụng RandomForestRegressor dự đoán giá trị “basesalary”

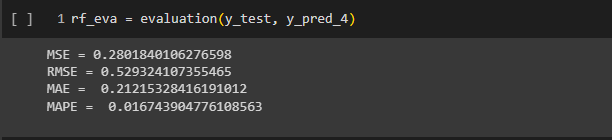
Random Forest Regressor là một thuật toán học máy dựa trên cây quyết định (decision tree) và kỹ thuật ensemble learning. Đây là một trong những thuật toán phổ biến và hiệu quả trong bài toán dự đoán và hồi quy. Ý tưởng của Random Forest Regressor là xây dựng nhiều cây quyết định độc lập và kết hợp chúng lại để tạo thành một mô hình dự đoán.

Khi thực hiện dự đoán cho một điểm dữ liệu mới, Random Forest Regressor sử dụng kết quả dự đoán của nhiều cây quyết định để đưa ra kết quả dự đoán cuối cùng. Kỹ thuật này giúp giảm thiểu các vấn đề liên quan đến overfitting, tăng tính ổn định của mô hình và độ chính xác của dự đoán.

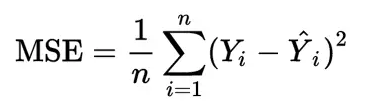


* Phân tích kết quả dự đoán.

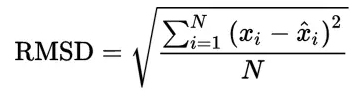
+ Phân tích kết quả qua độ đo



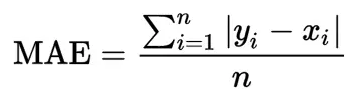
* MSE là trung bình của bình phương sai số giữa giá trị dự đoán và giá trị thực tế. MSE càng nhỏ thì mô hình dự đoán càng chính xác.



* RMSE là căn bậc hai của MSE. RMSE cũng được sử dụng để đánh giá hiệu quả của các mô hình dự đoán. RMSE càng nhỏ thì mô hình dự đoán càng chính xác.

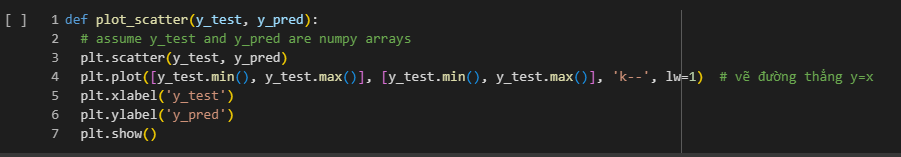


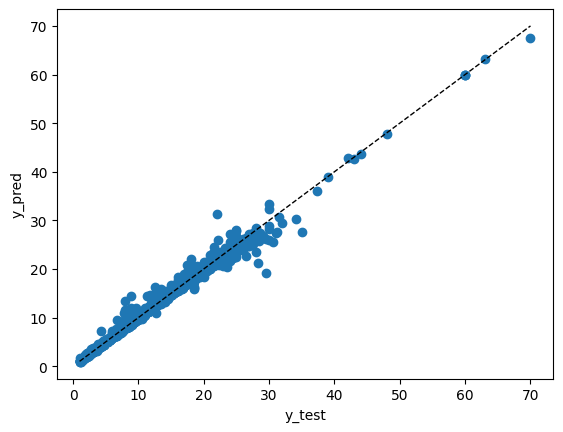
* MAE là trung bình của độ lệch tuyệt đối giữa giá trị dự đoán và giá trị thực tế. MAE càng nhỏ thì mô hình dự đoán càng chính xác.



* MAPE là trung bình của tỷ lệ độ lệch tuyệt đối giữa giá trị dự đoán và giá trị thực tế với giá trị thực tế. MAPE được tính bằng phần trăm và được sử dụng để đánh giá hiệu quả của các mô hình dự đoán với các giá trị dự đoán có giá trị lớn hơn không. MAPE càng thấp thì mô hình dự đoán càng chính xác.
* => Khi sử dụng RandomForest ta thu được các chỉ số độ đo khá nhỏ như MSE = 0,28, RMSE = 0,53, MAE = 0,212 và MAPE = 0,016, điều này cho thấy khi sử dụng Random Forest cùng với những feature có độ ảnh hướng lớn đến output đầu ra, ta thu được kết quả dự đoán trên tập test khá tốt. Do đó, chúng ta có thể sử dụng RandomForest cho bài toán dự đoán “basesalary”.

+ Phân tích kết quả qua hàm plot\_scatter



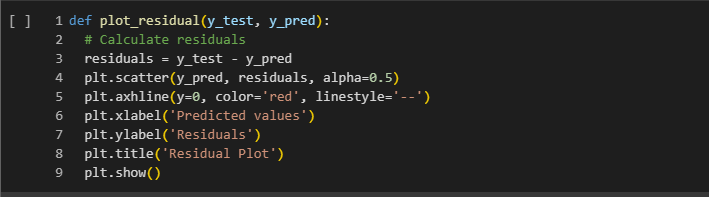


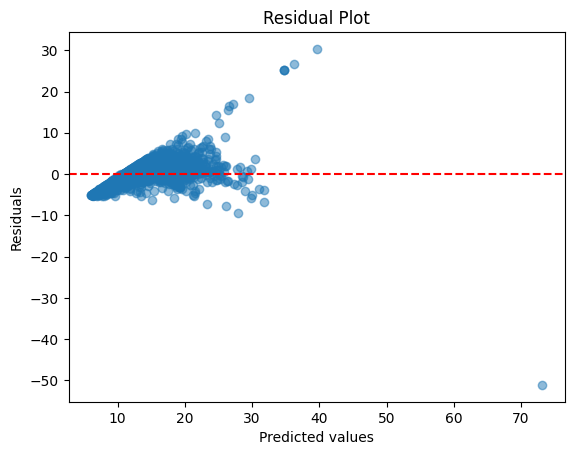
Đây là biểu đồ phân tán (scatter plot) của y\_test và y\_pred trên cùng một trục tọa độ, với trục x là giá trị của y\_test và trục y là giá trị của y\_pred.

Ngoài ra, đường thẳng y = x sẽ được vẽ bằng cách sử dụng hàm plot() của thư viện matplotlib, và đường ký hiệu bằng đường kẻ đứt. Đường kẻ đứt thường được sử dụng để biểu diễn giá trị thực tế và giá trị dự đoán trùng nhau. Khi các điểm nằm gần đường kẻ đứt như hình trên thì có nghĩa là giá trị dự đoán của mô hình gần giống với giá trị thực tế tương ứng. Nếu các điểm nằm rải rác khắp biểu đồ thì có thể cho thấy mô hình dự đoán không chính xác và cần được cải thiện.

Do đó, nếu các điểm càng gần đường kẻ đứt thì có thể cho thấy mô hình dự đoán đang hoạt động tốt và có độ chính xác cao hơn so với các điểm cách xa đường kẻ đứt. Tuy nhiên, việc đánh giá chính xác một mô hình dựa trên biểu đồ phân tán cần được kết hợp với việc sử dụng các độ đo như MSE, RMSE, MAE, MAPE để đánh giá hiệu quả của mô hình.

+ Phân tích kết quả bằng residual\_plot.

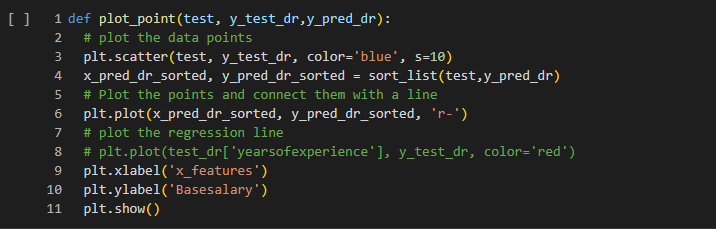




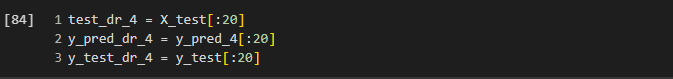
Trong biểu đồ residual plot, trục x là giá trị dự đoán và trục y là giá trị residual (y\_test – y\_pred) . Biểu đồ sẽ biểu diễn sự phân bố của các điểm trên mặt phẳng tọa độ và cho phép đánh giá mức độ chính xác của mô hình. Nếu các điểm được phân bố đều quanh đường kẻ đứt đỏ (đường y=0), thì mô hình được coi là chính xác và residual plot sẽ có hình dáng tròn. Nếu các điểm được phân bố không đều và có xu hướng tập trung ở một vùng trên hoặc dưới đường kẻ đứt, thì mô hình có thể cần được cải thiện.

Do đó khi nhìn vào biểu đồ kết quả trên, ta nhận thấy các điểm hầu hết phân bổ xung quanh đường thẳng y =0, tuy nhiên, vẫn còn một số điểm mà giá trị dự đoán có độ chênh lệch nhiều, hay cũng có thể nói đó là những điểm dữ liệu ngoại lai (outlier).

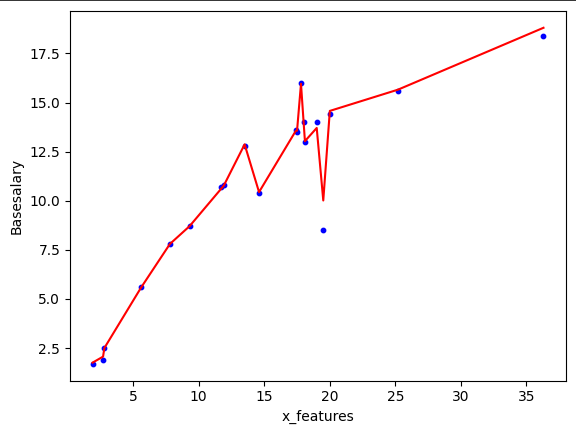
+ Phân tích kết quả qua biểu đồ đường thẳng hồi quy.



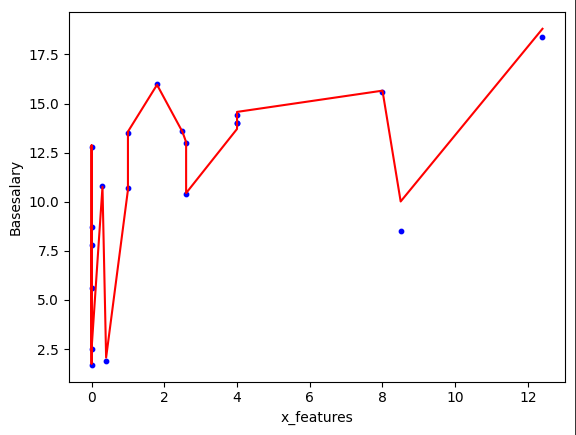
Biểu đồ này giúp người dùng có thể trực quan hóa mối quan hệ giữa các features và biến phụ thuộc và đánh giá mức độ phù hợp của mô hình dự đoán. Theo lẽ thông thường, chúng ta sẽ plot tất cả các điểm trong tập dữ liệu. Tuy nhiên, như vậy khá khó nhìn và nhiều, nên chúng em sẽ sử dụng 20 điểm để plot biểu đồ nhằm dễ dành đánh giá.



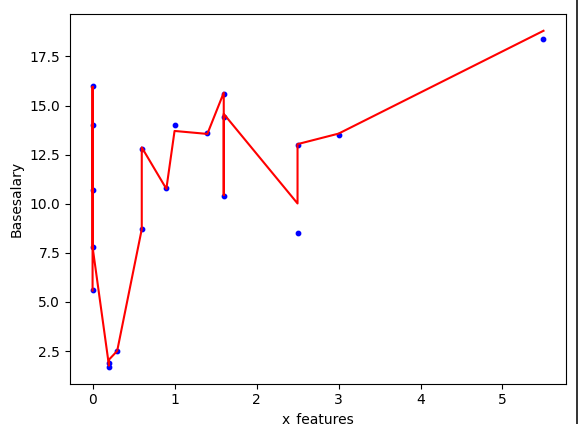
Đối với feature = ‘totalyearlycompensation”:



Đối với feature = ‘tockgrantvalue’:



Đối với feature = ‘bonus’:



Nhìn chung, Biểu đồ này là một biểu đồ plot trục x là các giá trị feature và trục y là giá trị của biến phụ thuộc (basesalary). Mỗi điểm trên biểu đồ biểu thị một giá trị feature và giá trị biến phụ thuộc tương ứng.

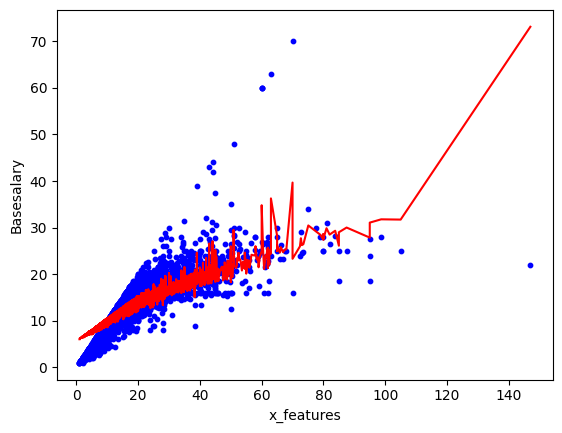
Đường màu đỏ trên biểu đồ là đường nối các điểm được sắp xếp theo giá trị feature tăng dần và tương ứng với giá trị dự đoán của biến phụ thuộc. Đường này được gọi là regression line (đường hồi quy), và thể hiện mối quan hệ tuyến tính giữa các features và biến phụ thuộc.

Biểu đồ này giúp người dùng có thể trực quan hóa mối quan hệ giữa các features và biến phụ thuộc và đánh giá mức độ phù hợp của mô hình dự đoán. Nếu các điểm trên biểu đồ phân bố gần đường regression line và không có sự tập trung lớn của các điểm ở bất kỳ phần nào của biểu đồ, thì mô hình dự đoán có thể được coi là khá chính xác. Ngược lại, nếu các điểm phân bố rộng và không có sự tập trung đáng kể, thì mô hình có thể cần được cải thiện.

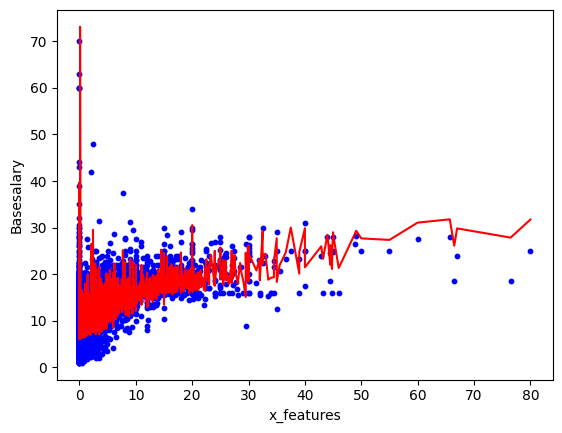
Vì vậy, khi nhìn vào biểu đồ giữa biến phụ thuộc “basesalary” và các biến đầu vào như “totalyearlycompensation”, “stockgrantvalue”, “bonus” với 20 điểm đầu tiên, ta thấy phần lớn các điểm nằm gần hoặc nằm trên đường thẳng hồi quy => Mô hình đang dự đoán khá tốt dựa trên những biến đầu vào.

Sau khi xét 20 điểm, chúng ta cũng cần xem xét trên toàn bộ tập dữ liệu. Tuy nhiên, vì nó quá nhiều điểm dữ liệu. Nên nhìn chung sẽ hơi khó cho việc phân tích.

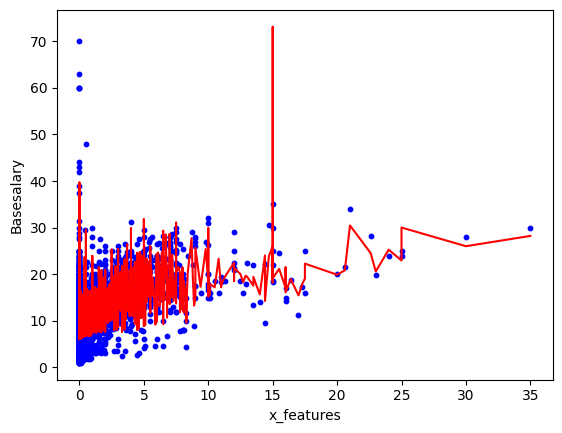
+ Với thuộc tính “totalyearlycompensation”:



+ Với thuộc tính “stockgrantvalue”



+ Với thuộc tính bonus:



Nhìn vào kết quả biểu diễn của sơ đồ, ta có thể rút ra thêm được một vài nhận xét và xu hướng của dữ liệu như sau:

+ Đối với x\_features = ‘totalyearlycompensation’, khi giá trị của totalyearlycompensation tăng dần thì giá trị basesalary có xu hướng tăng theo, và dường như sẽ tăng mạnh. Ngoài ra, ta thấy giá trị x\_features cũng nằm nhiều trong khoảng từ 0-> 60. Điều này cho thấy, totalyearlycompensation có sự ảnh hưởng to lớn đến kết quả dự đoán giá trị lương “basesalary”.

+ Đối với x\_features = ‘stockgrantvalue”, khi giá trị stockgrantvalue tăng thì giá trị basesalary có xu hướng tăng nhẹ, tuy nhiên vẫn tồn tại những điểm có mức độ chệnh lệch tương đối. Và giá trị stockgrantvalue nằm nhiều trong khoảng từ 0-30. Nhưng nhìn chung, đặc trưng stockgrantvalue vẫn có mức ảnh hưởng đáng kể lên kết quả dự đoán basesalary.

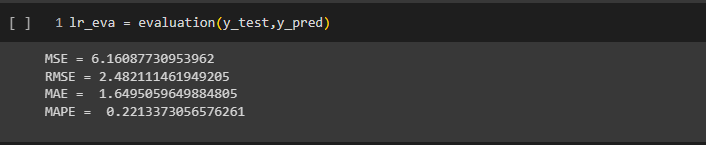
+ Đối với x\_features = ‘bonus’, dường như xu hướng của giá trị dự đoán “basesaly” không tăng nhiều khi giá trị ‘bonus’ tăng, có thiên hướng nằm ngang, tăng rất nhẹ. Ngoài ra khi nhìn vào biểu đồ trên, giá trị bonus nằm nhiều ở khoảng từ 0-10 và giữa đầu vào bonus và giá trị dự đoán basesalary cũng có mức độ tương quan nhất định.

* Sử dụng một số model khác

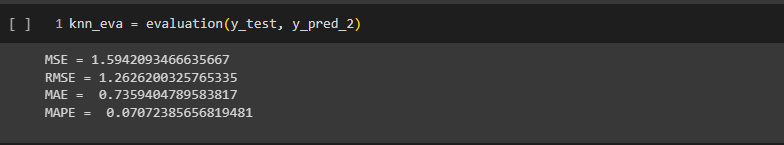
Ngoài sử dụng RandomForest cho bài toán dự đoán giá trị này, nhóm chúng em có tiến hành thử nghiệm và chạy trên một vài mô hình máy học khác. Tuy nhiên, do kết quả Random Forest là tốt nhất và cho ra xu hướng, đặc trưng dữ liệu là chính xác nên chúng em chỉ tập trung phân tích các biểu đồ và dữ liệu thông qua mô hình RandomForest. Tuy nhiên, trong file source code đính kèm, nhóm cũng plot ra tất cả các biểu đồ trên cho từng mô hình.

Thay vào đó, em sẽ trình bày kết quả các độ đo dự đoán trên từng mô hình như sau.

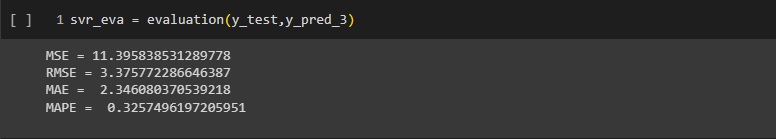
+ Với mô hình Linear Regression



+ Với mô hình KNN



+ Với mô mình SVR



Thông qua độ do đánh giá trong mô hình hồi quy, dễ dàng nhận thấy đối với tập dữ liệu của chúng ta, RandomForest sẽ có kết quả tốt nhất. Tiếp theo sau đó sẽ là các mô hình máy học khác như KNN, Linear Regression và cuối cùng sẽ là SVR.

# **TÀI LIỆU THAM** **KHẢO**

[1] <https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16>

[2] <https://learn.microsoft.com/en-us/analysis-services/analysis-services-tutorials-ssas?view=asallproducts-allversions>

[3] <https://learn.microsoft.com/en-us/analysis-services/multidimensional-tutorial/multidimensional-modeling-adventure-works-tutorial?view=asallproducts-allversions>

[4] <https://tedu.com.vn/co-so-du-lieu/mo-hinh-data-warehouse-so-do-ngoi-sao-148.html>

[5] <https://bootcamp.pe.gatech.edu/blog/10-key-data-mining-techniques-and-how-businesses-use-them/>

[6] <https://www.kaggle.com/code/carolzhangdc/predict-imdb-score-with-data-mining-algorithms>