Hệ thống thông tin phục vụ trí tuệ kinh doanh

ĐỒ ÁN MÔN HỌC

REPORT

Lớp: 20HTTT01

Nhóm *<20HTTT1_4>*

<20127233> - <Huỳnh Thế Long>

<20127432> - <Nguyễn Hoài An>

Khoa Công nghệ Thông tin

Đại học Khoa học Tự nhiên TP HCM

Tháng Thg12-23

MỤC LỤC

1 Tổng quan

Thông tin nhóm

MSSV	Họ tên
20127233	Huỳnh Thế Long
20127432	Nguyễn Hoài An

Thông tin đồ án

Phiên bản sử dụng:

- SQL Server Integration Services Projects 2022
- Microsoft Analysis Services Projects 2022
- Visual Studio 2022

Cấu trúc thư mục

- Resource
    - Script SQL
        - Other
            - DDSwithdata.sql (DDS có dữ liệu, có thể dùng file này để build cube, viết MDX)
        - 1.Metadata.sql (cấu trúc và dữ liệu)
        - 2. Source (chỉ cấu trúc)
        - 3. Stage (chỉ cấu trúc)
        - 4. NDS (cấu chúc, có dữ liệu bảng source, status)
        - 5. DDS (chỉ cấu trúc)
    - Data-Mining.ipynb (file jupyter notebook bằng ngôn ngữ Python trong phần data mining của nhóm)
    - Dashboard.pbix (file power bi phần dashboard của nhóm)
    - MDXQuery.mdx (file mdx chứa toàn bộ mdx query của nhóm)
- SSAS (Project phần SSAS của nhóm)
- SSIS (Project phần SSIS của nhóm)

Tỷ lệ tham gia đóng góp công việc

MSSV	Họ tên	% Đóng góp
20127233	Huỳnh Thế Long	100%
20127432	Nguyễn Hoài An	100%

Phân công công việc

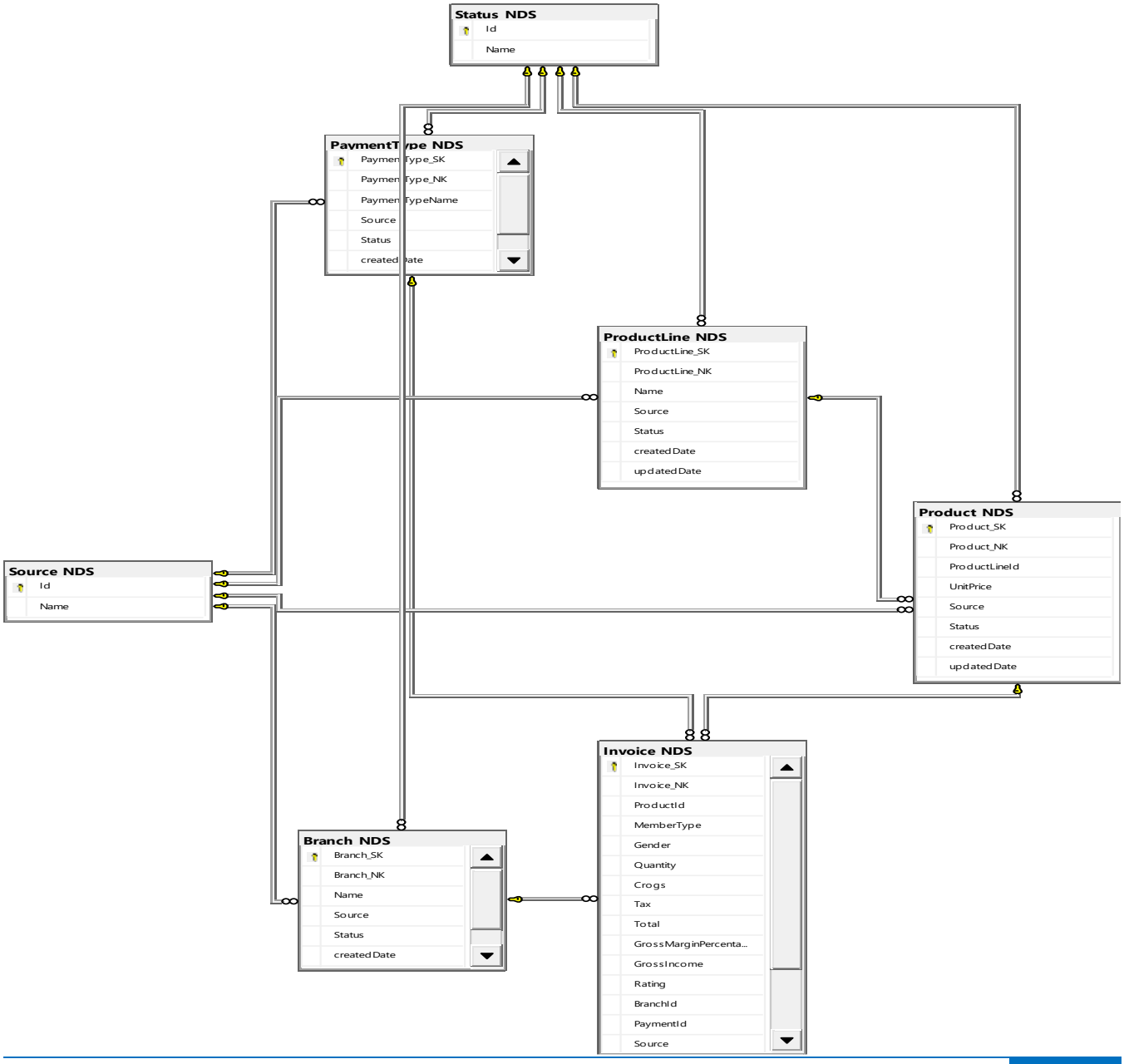
Phân công	Phân công cụ thể	Tên người tham gia	Đánh giá
Thiết kế & cài đặt CSDL NDS, DDS	Thiết kế NDS	20127233	100%
		20127432	100%
	Thiết kế DDS	20127233	100%
		20127432	100%
ETL	Tổng phần ETL	20127233	100%
	Tổng phần ETL	20127432	100%
ETL Source to Source SQL Server	ETL Source Excel to Source OLE DB	20127233	
ETL Source to Stage	ETL Source to Stage	20127233	
ETL Stage to NDS	ProductLine_Stage to ProductLine_NDS	20127432	
	Product_Stage to Product_NDS	20127432	
	Brach_Stage to Branch_NDS	20127233	
	Sales_Stage to PaymentType_NDS	20127233	
	Sales_Stage to Invoice_NDS	20127233	
ETL NDS to DDS	Dim Branch	20127432	
	Dim Payment Type	20127432	
	Dim Product	20127432	
	Dim Date	20127233	
	Fact Sales	20127233	
MDX	Tổng phần MDX	20127233	100%

	Tổng phần MDX	20127432	100%
	MDX nhu cầu 1,2,3,4	20127233	
	MDX nhu cầu 4,5,6	20127432	
OLAP	Tổng phần OLAP	20127233	100%
	Phân cấp chiều Date	20127233	
	Phân cấp chiều Product	20127432	
	Dashboard Power Bi	20127233	
	Phân cấp chiều Customer	20127233	
	Thêm measure Quantity cho Cube	20127233	
Mining	Tổng phần Mining	20127432	100%
Report	Viết báo cáo word	20127233	100%
	Viết cáo cáo word	20127432	100%
	Quay video ETL process	20127233	100%

2 Nội dung đồ án

2.1 Thiết kế

Thiết kế NDS



Cấu trúc bảng Branch_NDS

Column Name	Data Type	Description	Tranformation
Branch_SK	int	Khóa chính	
Branch_NK	nvarchar(255)	Mã chi nhánh	
Name	nvarchar(255)	Tên chi nhánh	
Source	int	Nguồn	
Status	int	Trạng thái	
createdDate	datetime	Ngày tạo	
updatedDate	datetime	Ngày cập nhật	

Cấu trúc bảng Product_NDS

Column Name	Data Type	Description	Tranformation
Product_SK	int	Khóa chính	
Product_NK	nvarchar(255)	Mã sản phẩm	
ProductLineID	int	Mã dòng sản phẩm	
UnitPrice	money	Giá trên một sản phẩm	
Source	int	Nguồn	
Status	int	Trạng thái	
createdDate	datetime	Ngày tạo	
updatedDate	datetime	Ngày cập nhật	

Cấu trúc bảng ProductLine_NDS

Column Name	Data Type	Description	Tranformation
ProductLine_SK	int	Khóa chính	
ProductLine_NK	nvarchar(255)	Mã dòng sản phẩm	
Name	nvarchar(255)	Tên dòng sản phẩm	
Source	int	Nguồn	
Status	int	Trạng thái	
createdDate	datetime	Ngày tạo	
updatedDate	datetime	Ngày cập nhật	

Cấu trúc bảng PaymentType_NDS

Column Name	Data Type	Description	Tranformation
PaymentType_SK	int	Khóa chính	
PaymentType_NK	int	Mã Thanh toán	PaymenType_NK được tạo theo công thức: "PMT" + ${PaymenType_SK} Ví dụ nếu PaymentType_SK = 1 thì PaymentType_NK = "PMT1"
Name	nvarchar(255)	Tên phương thức thanh toán	
Source	int	Nguồn	
Status	int	Trạng thái	
createdDate	datetime	Ngày tạo	
updatedDate	datetime	Ngày cập nhật	

Cấu trúc bảng Invoice_NDS

Column Name	Data Type	Description	Tranformation
Invoice_SK	int	Khóa chính	
Invoice_NK	nvarchar(255)	Mã hóa đơn	
ProductId	int	Mã sản phẩm	
MemberType	nvarchar(255)	Loại khách hàng	
Gender	nvarchar(6)	Giới tính	Chuyển đổi 'F' sang 'Female', 'M' sang 'Male'
Quantity	int	Số lượng mua	
Corgs	money	Chi phí sản phẩm bán được	
Tax	money	Thuế 5%	
Total	money	Tổng giá trị đơn hàng	
GrossMarginPercentage	money	Tỉ lệ lợi nhuận	
GrossIncome	money	Thu nhập	
Rating	float	Đánh giá trên đơn hàng	
BranchId	int	Mã chi nhánh	
PaymentId	int	Mã thanh toán	
Source	int	Nguồn	
Status	int	Trạng thái	
createdDate	datetime	Ngày tạo	Được lấy từ 2 cột Date và Time từ Source. Sẽ thực hiện việc cộng 2 cột thành một chuỗi, sau đó ép kiểu của chuỗi đó sang datetime. Dữ liệu mẫu của cột createdDate: "2019-01-30 14:43:00.000"
updatedDate	datetime	Ngày cập nhật	

Cấu trúc bảng Source

Column Name	Data Type	Description	Tranformation
ID	int	Khóa chính	
Name	nvarchar(255)	Tên nguồn	

Dữ liệu trong bản Source của NDS được thêm sẵn, vì đồ án chỉ có 1 dataset nên bảng source có 1 dòng

	Id	Name
1	1	supermarket_sales

Cấu trúc bảng Status

Column Name	Data Type	Description	Tranformation
ID	int	Khóa chính	
Name	nvarchar(255)	Tên trạng thái	

Dữ liệu trong bảng Status của NDS được nhóm thêm sẵn

	Id	Name
1	0	Inactive
2	1	Active

Thiết kế DDS


Dim_Product
Product_SK
Product_NK
ProductLineId
ProductLineName
UnitPrice
Source
Status
Fact_Sales
Date_Key
Product_Key
PaymentType_Key
Branch_Key
InvoiceId
Invoice_NK
UnitPrice
Quantity
Crogs
Tax
Total
GrossMarginPercentange
GrossIncome
Rating
Gender
MemberType
Dim_Date
Date_SK
Date_NK
Hour
Day
Month
EnglishMonthName
VietnameseMonthName
BurmeseMonthName
Year
Dim_Branch
Branch_SK
Branch_NK
Name
Source
Status
Dim_PaymentType
PaymentType_SK
PaymentType_NK
PaymentType_Name
Source
Status

Cấu trúc bảng Dim_Product

Column Name	Data Type	Description	Tranformation
Product_SK	int	Khóa chính	
Product_NK	nvarchar(255)	Mã sản phẩm	
ProductLineID	int	Mã dòng sản phẩm	
ProductLineName	nvarchar(255)	Tên mã dòng sản phẩm	
UnitPrice	money	Giá trên một sản phẩm	
Source	int	Nguồn	
Status	int	Trạng thái	

Cấu trúc bảng Dim_Branch

Column Name	Data Type	Description	Tranformation
Branch_SK	int	Khóa chính	
Branch_NK	nvarchar(255)	Mã chi nhánh	
Name	nvarchar(255)	Tên chi nhánh	
Source	int	Nguồn	
Status	int	Trạng thái	

Cấu trúc bảng Dim_PaymentMethod

Column Name	Data Type	Description	Tranformation
PaymentType_SK	int	Khóa chính	
PaymentType_NK	int	Mã Thanh toán	
Name	nvarchar(255)	Tên phương thức thanh toán	
Source	Int	Nguồn	
Status	int	Trạng thái	

Cấu trúc bảng Dim_Date

Column Name	Data Type	Description	Tranformation
Date_SK	int	Khoá chính	
Date_NK	datetime		
Hour	time (0)	Giờ và phút. Dữ liệu mẫu: "14:43:00"	
Day	Int	Ngày trong tháng	
Month	Int	Tháng trong năm	
Year	Int	Năm	
EnglishMonthName	nvarchar(255)	Tên tháng bằng tiếng Anh dựa trên cột "Month". Dữ liệu mẫu: "January"	Biến đổi dựa trên cột Month, cột Month có giá trị từ 1->12 sẽ ra tên tháng bằng tiếng Anh tương ứng
VietnameseMonthName	nvarchar(255)	Tên tháng bằng tiếng Việt dựa trên cột "Month" Dữ liệu mẫu: "Tháng một"	Biến đổi dựa trên cột Month, cột Month có giá trị từ 1->12 sẽ ra tên tháng bằng tiếng Việt tương ứng
BurmeseMonthName	nvarchar(255)	Tên tháng bằng tiếng Miến Điện dựa trên cột "Month" Dữ liệu mẫu: "ဇန်နဝါရီ"	Biến đổi dựa trên cột Month, cột Month có giá trị từ 1->12 sẽ ra tên tháng bằng tiếng Miến Điện tương ứng

Cấu trúc bảng Fact_sale

Column Name	Data Type	Description	Tranformation
Date_Key	int	Khoá ngoại tham chiếu đến Dim_Date	
Prodcut_Key	int	Khoá ngoại tham chiếu đến Dim_Product	
PaymentType_Key	int	Khoá ngoại tham chiếu đến Dim_PaymentType	
InvoiceId	int	Khoá chính	
Invoice_NDS	nvarchar(255)	Khoá tự nhiên của hoá đơn	
UnitPrice	money	Đơn giá sản phẩm	
Quantity	int	Số lượng sản phẩm mua trên 1 hoá đơn	
Crogs	money	Chi phí của sản phẩm	
Tax	money	Thuế 5% trên tổng (UnitPrice * Quantity)	
Total	money	Tổng tiền của một hoá đơn (UnitPrice * Quantity) + Tax	
GrossMarginPercentange	money	Tỉ lệ lợi nhuận	
GrossIncome	money	Thu nhập	
Rating	float	Đánh giá của khách hang	
Gender	nvarchar(6)	Giới tính của khách hang	
MemberType	nvarchar(255)	Loại khách hang của khách hàng	

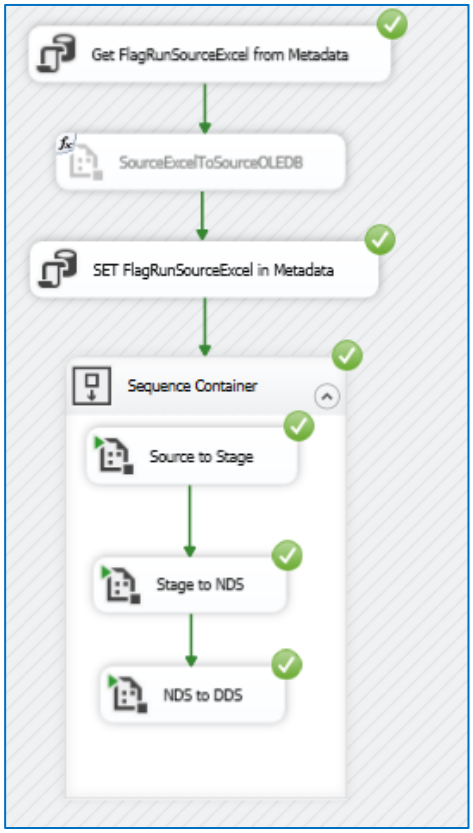
MetaData

Cấu trúc bảng Data_Flow

Column Name	Data Type	Description	Tranformation
ID	int	Khóa chính	
Name	nvarchar(255)	Tên table	
LSET	datetime	Last Successful Extraction Time	
CET	datetime	Current Extraction Time	

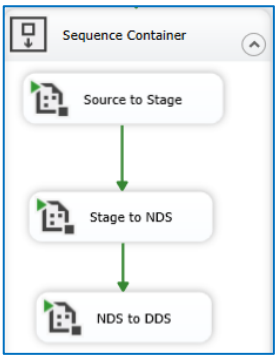
2.2 ETL

Main Package

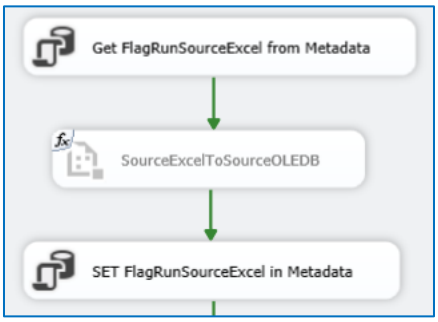


Giải thích:

Main Package đây là Entry Point Package. Nó sẽ chạy các package khác theo tuần tự: Source -> Stage -> NDS -> DDS



Ngoài ra project còn có thêm bước giả lập nạp dữ liệu nguồn Excel vào Sql Server, mục đích xây dựng 1 process xuyên suốt. Như vậy mong muốn khi lần đầu ETL. Nó sẽ nạp dữ liệu từ source Excel và source OLE DB 1 lần duy nhất.

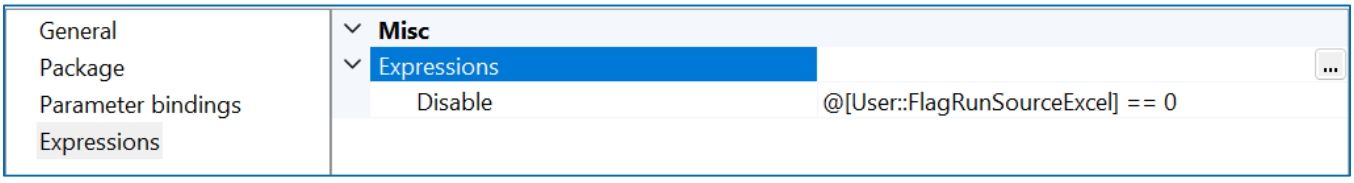


Cách làm: Sử dụng Variable và Expression trong SSIS. Và có 1 flag trong metadata để kiểm tra kiểm tra xem có cần nạp dữ liệu từ Excel hay không. Nếu flag = 1 thì sẽ nạp dữ liệu

	Id	Name	FlagRunOnce
1	1	FlagRunSourceExcel	0

Dữ liệu trong bảng ETL_RunOnceControl của Metadata

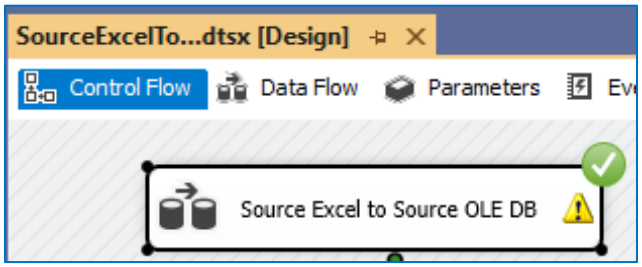
- Đầu tiên nó sẽ lấy ra dữ liệu FlagRunSourceExcel từ Metadata. Sau khi lấy được dữ liệu nó sẽ gán dữ liệu đó vào variable [User:FlagRunSourceExcel]
- Ở Component SourceExcelToSourceOLEDB nó sẽ kiểm tra điều kiện, nếu User:FlagRunSourceExcel = 0 thì sẽ không chạy component đó
- Sau khi chạy trong component SourceExcelToSourceOLEDB, sẽ thực hiện việc gán FlagRunSourceExcel = 0 vào Metadata



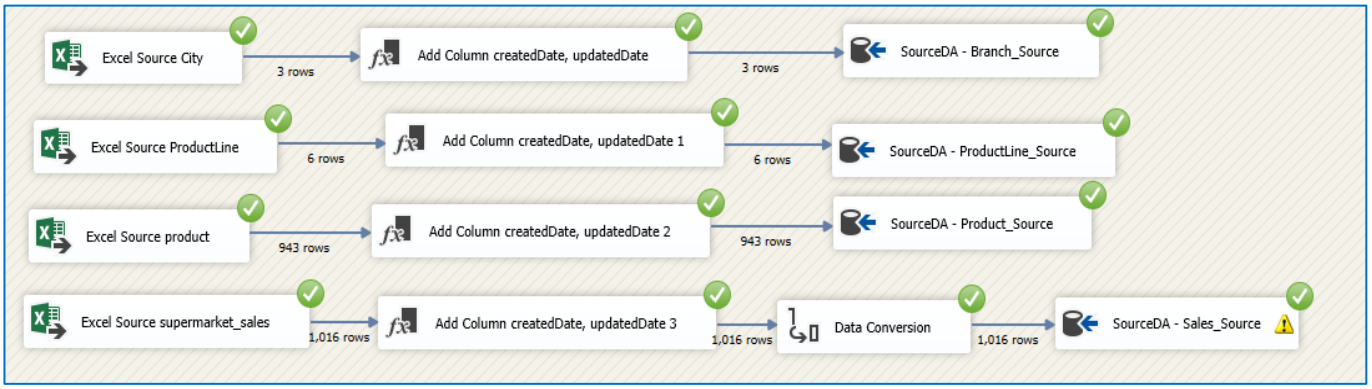
Expression của component SourceExceltoSourceOLEDB

ETL Excel Source to OLE DB Source

Control Flow



Data Flow



Data Flow of Package Source Excel to Source OLE DB

Giải thích:

- Đọc dữ liệu từ nguồn Excel, lấy ra các bảng, các cột cần thiết
- Vì ở giai đoạn ETL từ Source->Stage sẽ dùng Incremental extract nên sẽ thêm cột 'createdDate' và 'updatedDate' cho các cột chưa có

Derived Column Name	Derived Column	Expression
createdDate	<add as new column>	GETDATE()
updatedDate	<add as new column>	GETDATE()

- Ở bảng 'supermarket_sales' có time, và date nên cột creadtedDate và updatedDate sẽ sử dụng giá trị đó
    - Thực hiện một số bước chuyển đổi cho phù hợp:

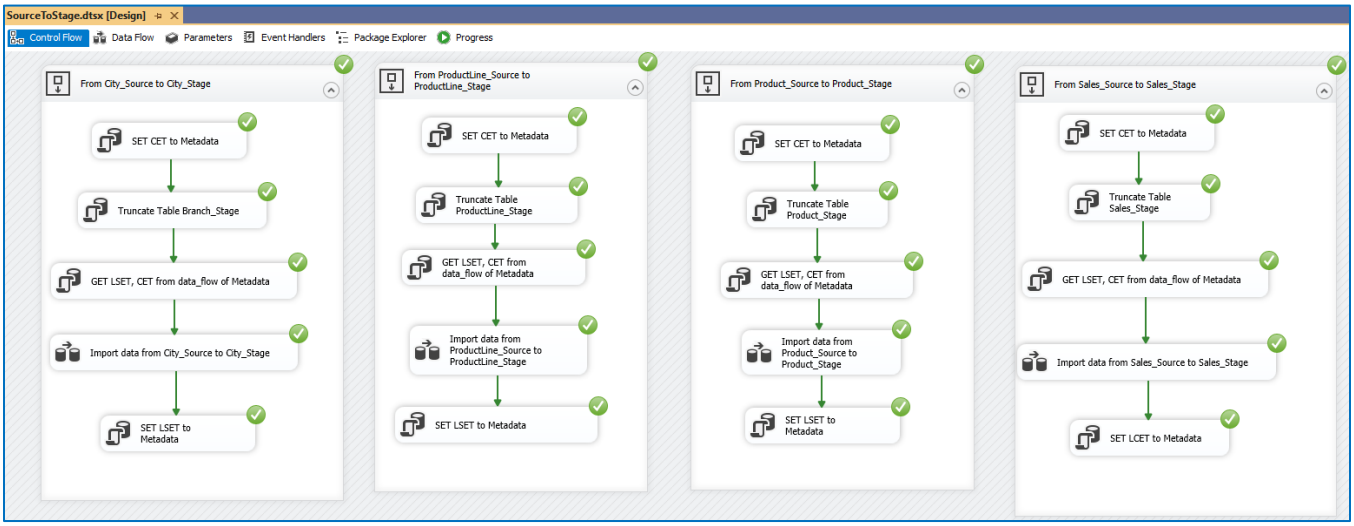
Derived Column Name	Derived Column	Expression	Data Type
createdDate	<add as new column>	(DT_WSTR,50)(Date) + " " + (DT_WSTR,50)(Time)	Unicode string [DT_WSTR]
updatedDate	<add as new column>	(DT_WSTR,50)(Date) + " " + (DT_WSTR,50)(Time)	Unicode string [DT_WSTR]

Nối 2 cột Date và Time trong 'supermarket_sales' lại với nhau. Tuy nhiên kiểu dữ liệu trả về của nó là `Unicode string` vì vậy cần phải chuyển đổi nó sang dạng database timestamp để cho phù hợp với kiểu dữ liệu trong source OLE DB

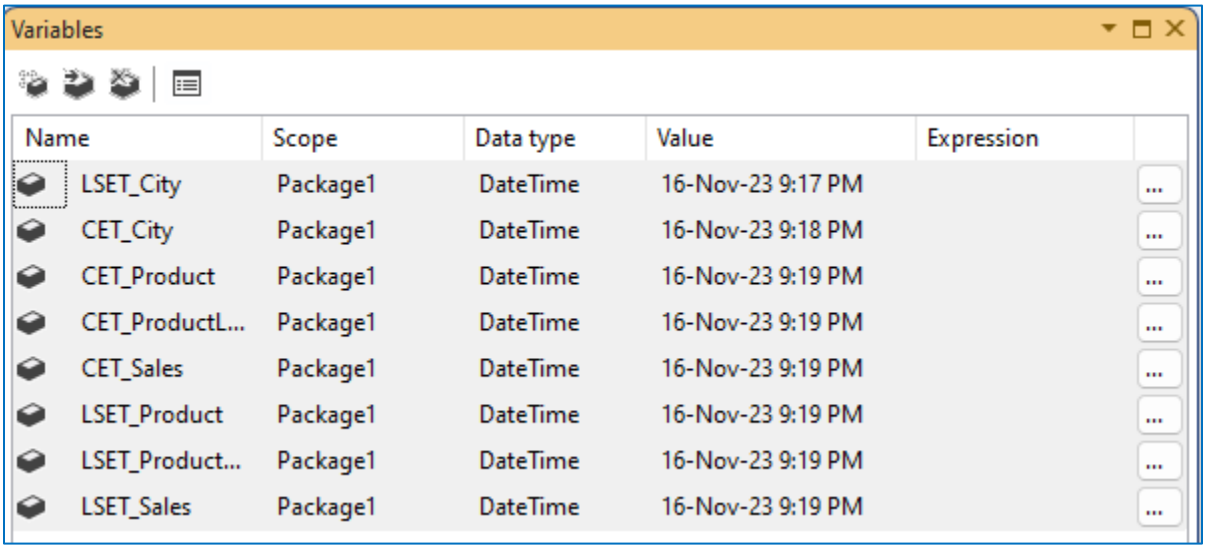
Input Column	Output Alias	Data Type
createdDate	createdDate	database timestamp [DT_DBTIM...
updatedDate	updatedDate	database timestamp [DT_DBTIM...

ETL Source to Stage

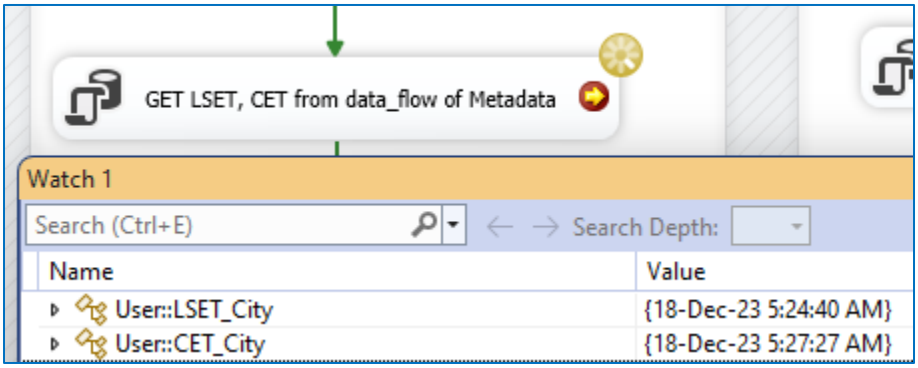
Control Flow



Variables được sử dụng trong package Source to Stage

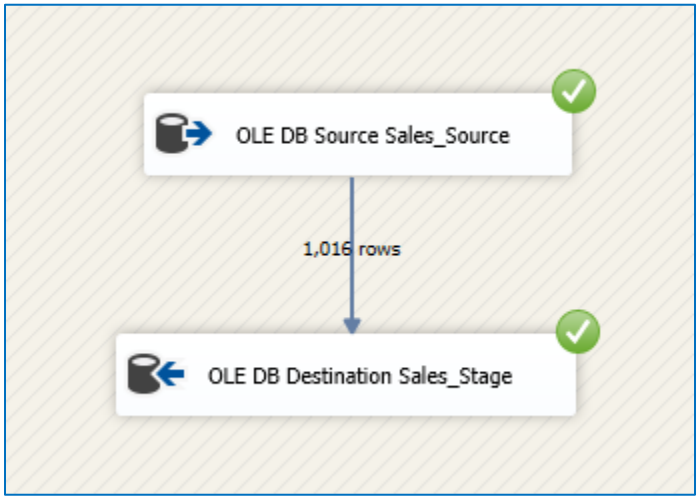


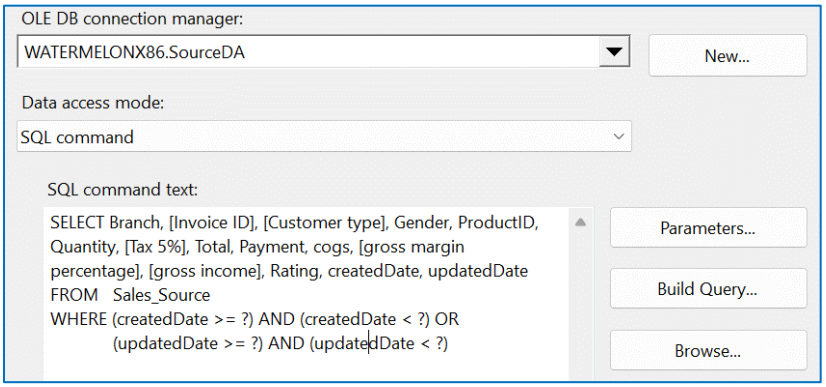
Kiểm tra xem các biến LSET, CET có được thiết lập và lấy ra đúng giá trị hay không

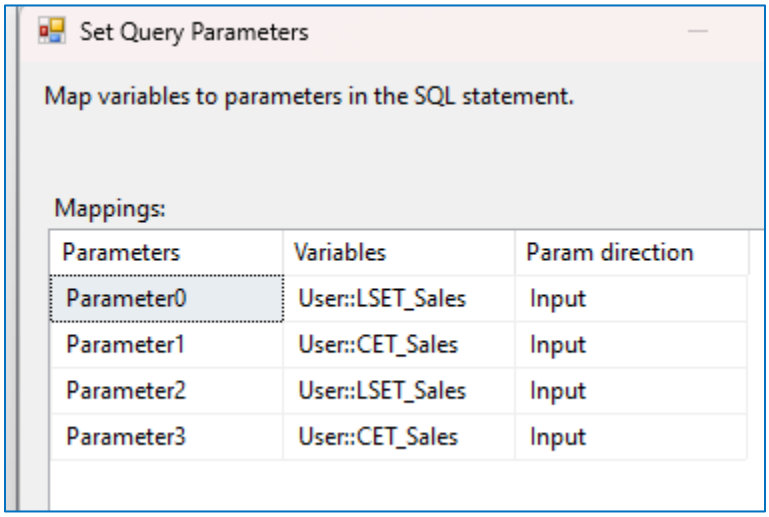


Data Flow

Ví dụ cho bảng Sales, các bảng khác làm tương tự:

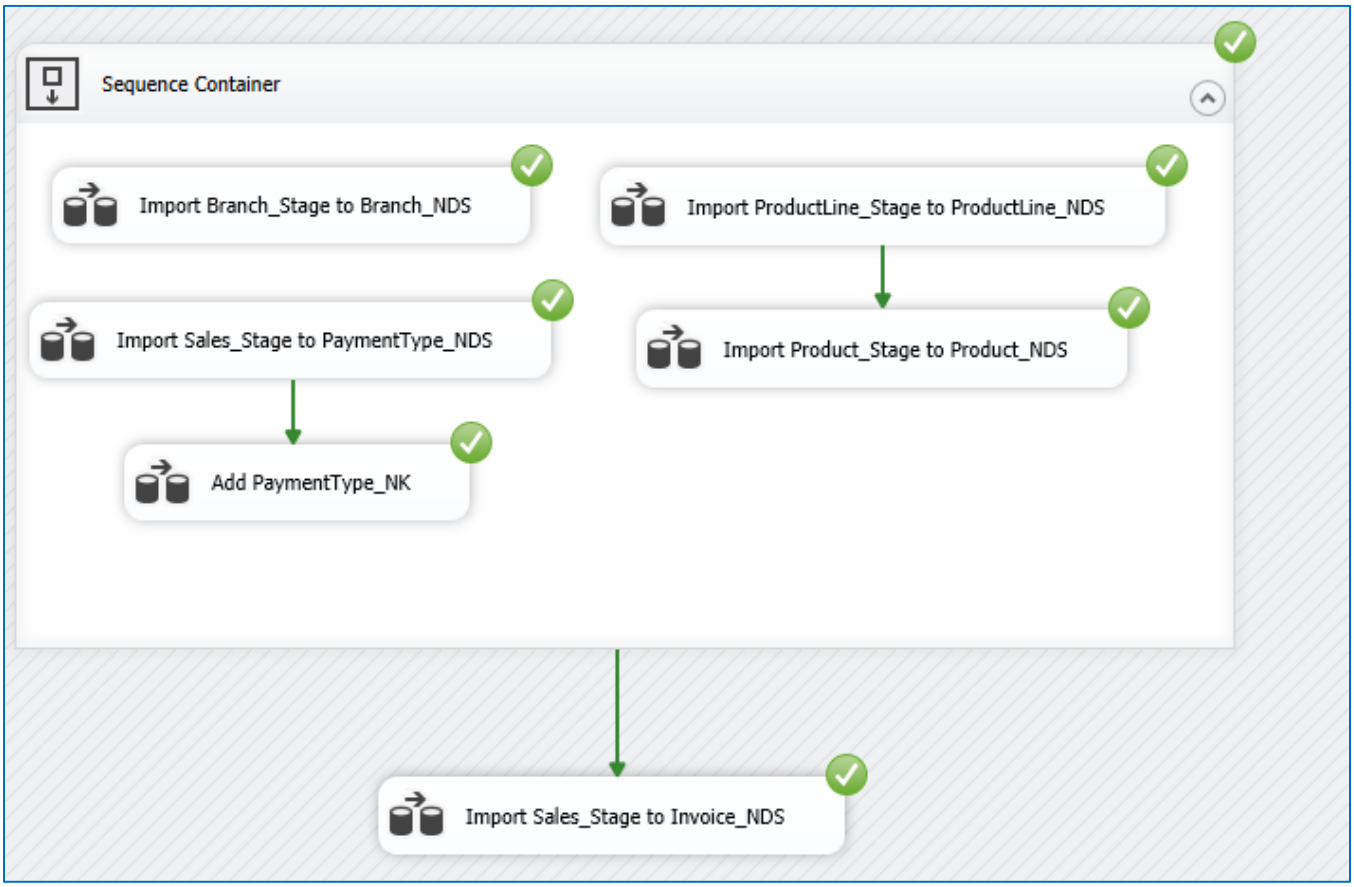






ETL Stage to NDS

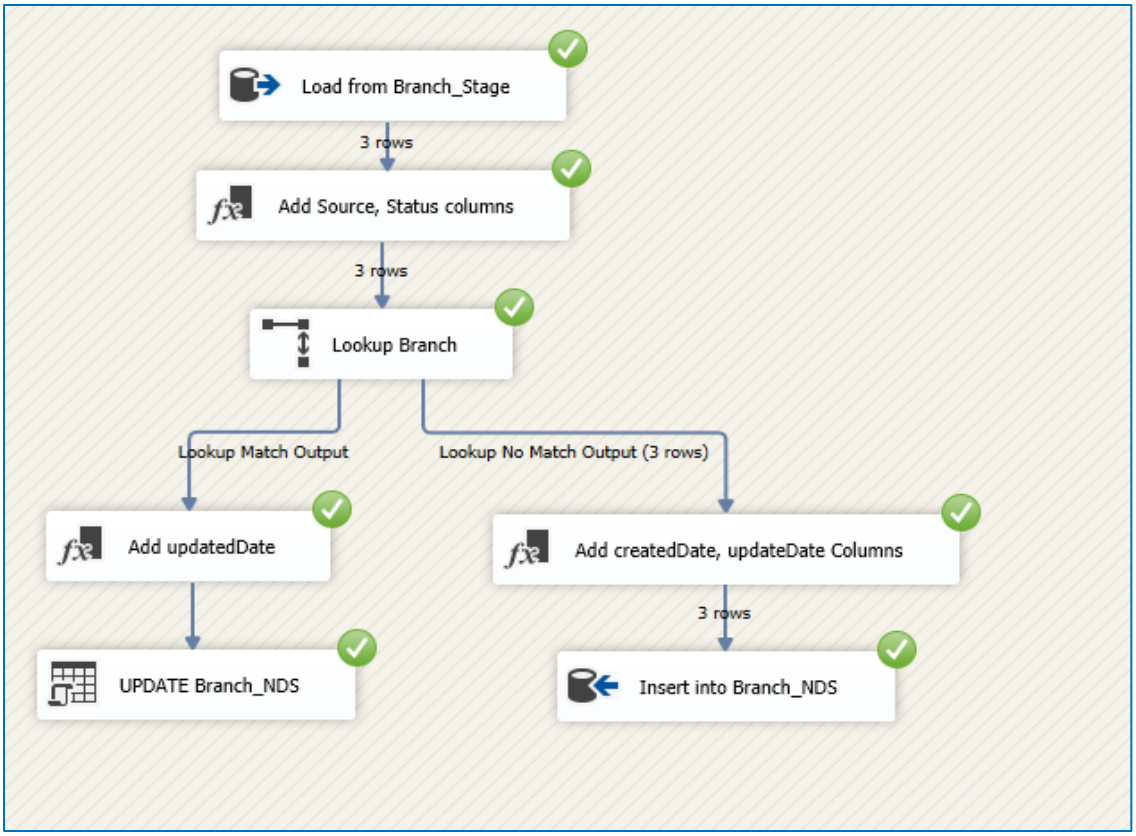
Control Flow



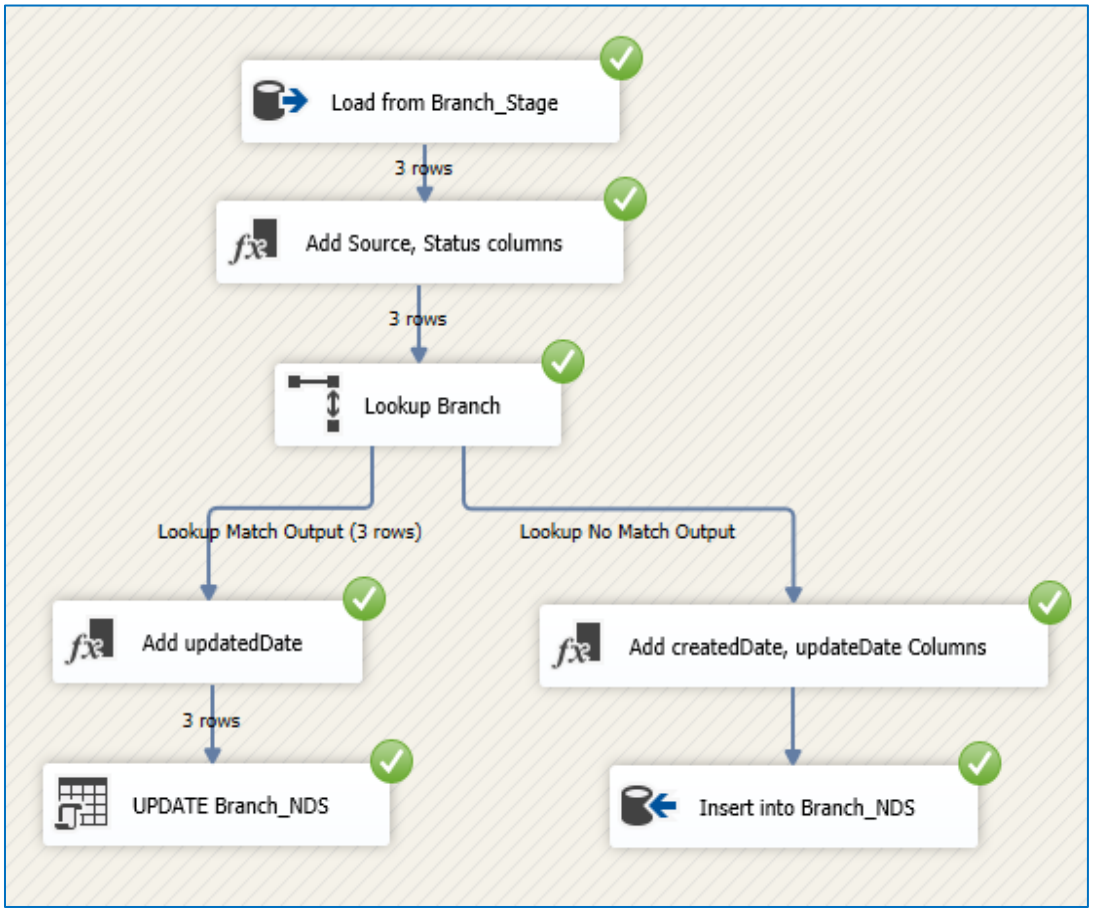
Data Flow

Brach_Stage to Branch_NDS

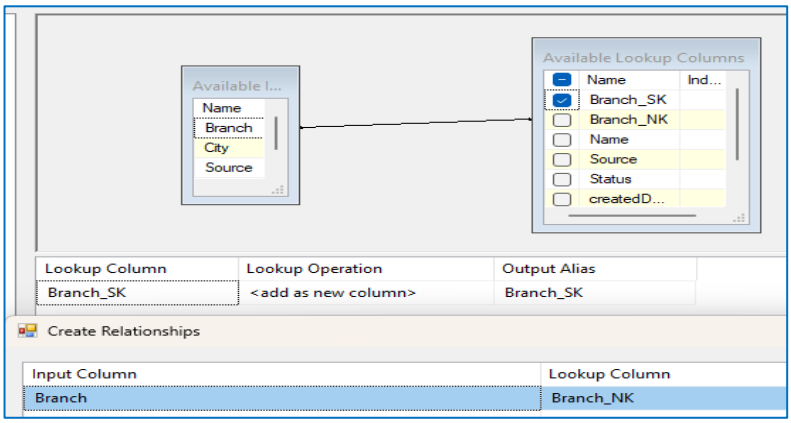
Giải thích: Ban đầu ở Metadata LSET, CET đang bằng null, vì vậy khi ETL lần đầu toàn bộ dữ liệu sẽ nạp qua stage. Và ở gia đoạn Stage qua NDS. Nó sẽ lookup để tìm xem là những dòng dữ liệu trong bảng stage đã có trong bảng NDS chưa ? Nếu chưa có thì sẽ thêm vào bảng NDS



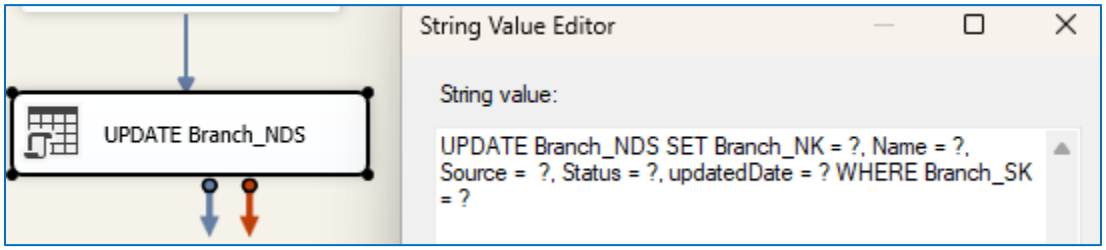
Ở lần ETL thứ 2 sau khi lookup, biết được rằng các dòng trong stage đã có trong NDS, vì vậy thay vì insert nó sẽ chuyển sang update các dòng dữ liệu trong NDS và cũng như cập nhập lại trường updatedDate



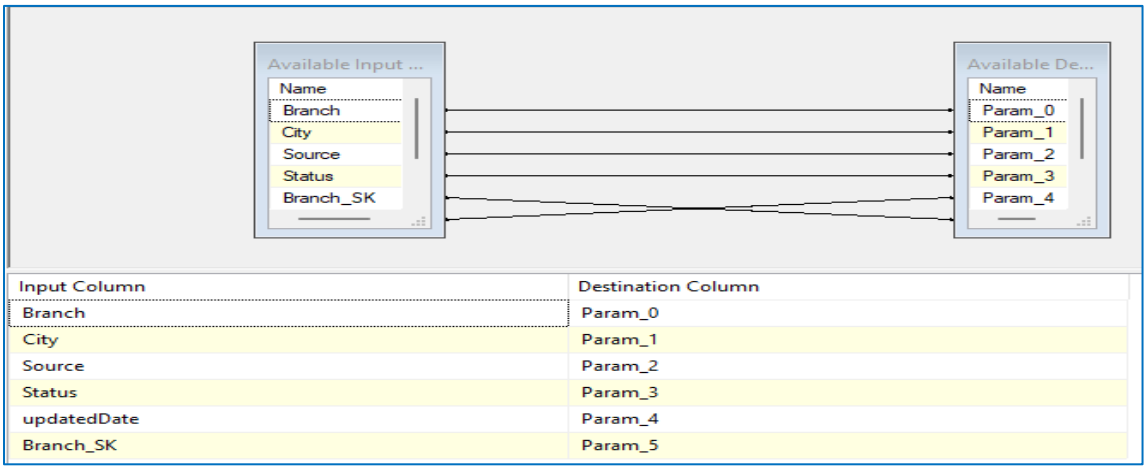
Cấu hình lookup:



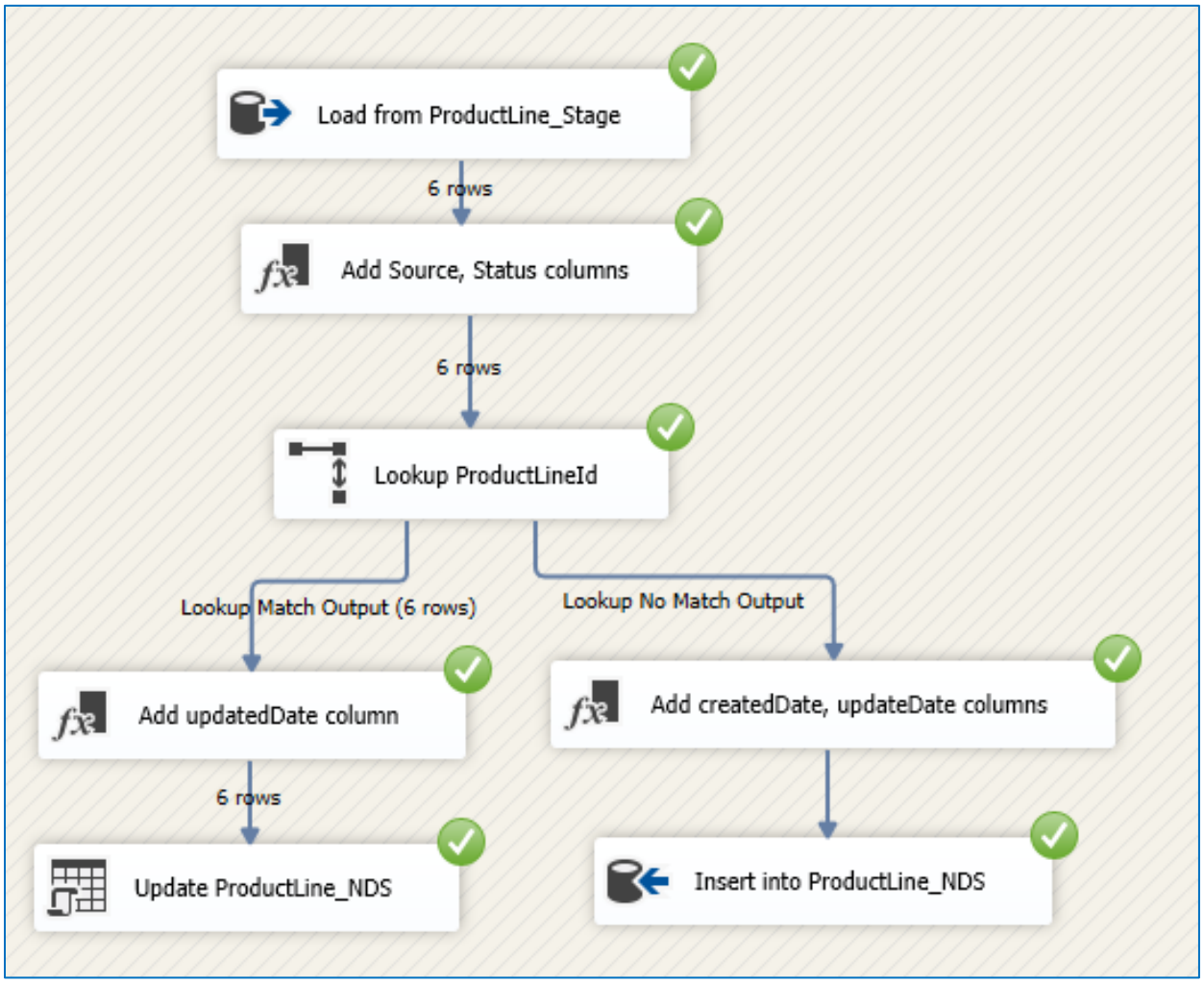
Cấu hình của UPDATE Branch_NDS



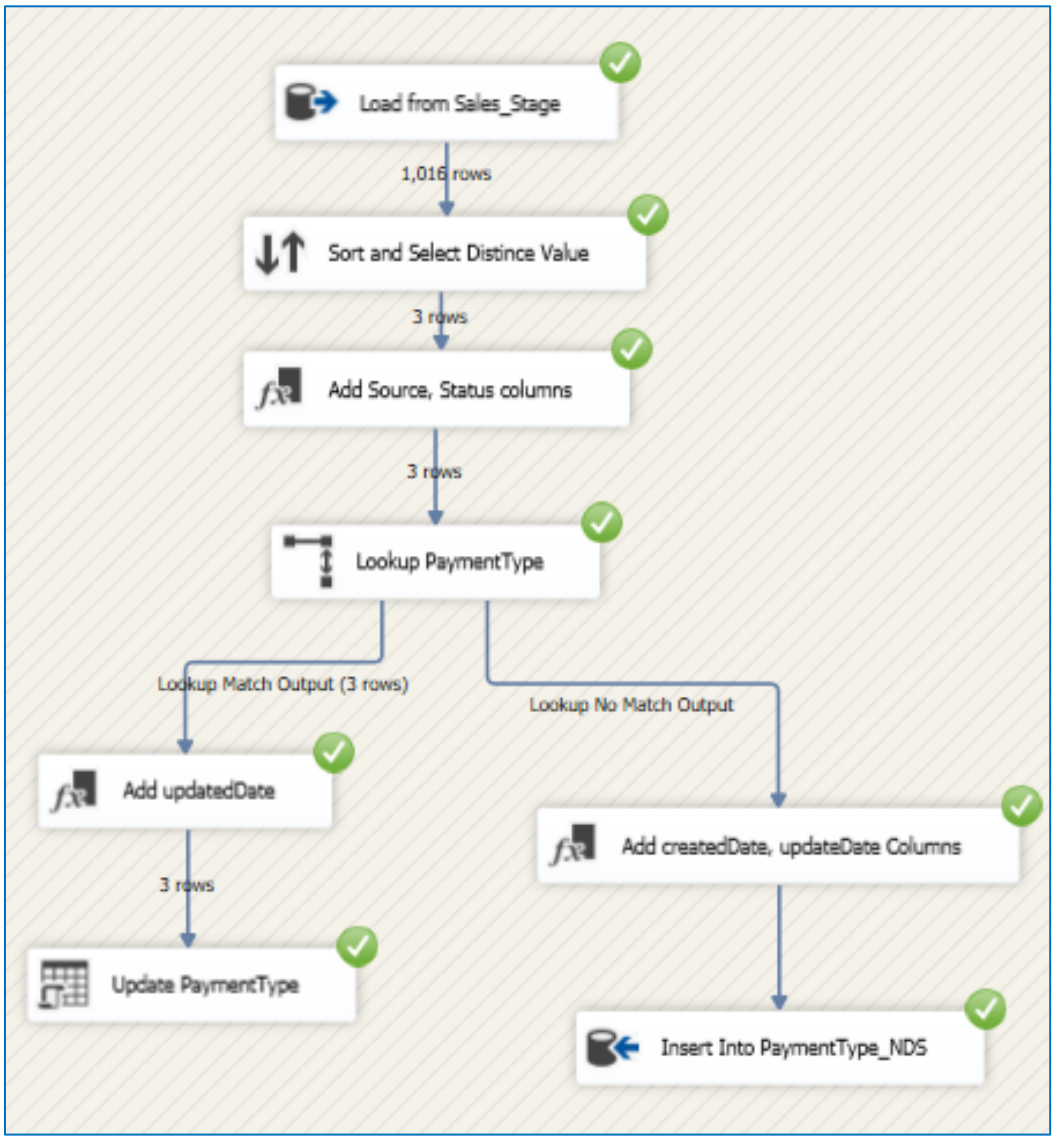
Mapping các thuộc tính của Branch_NDS



ProductLine_Stage to ProductLine_NDS

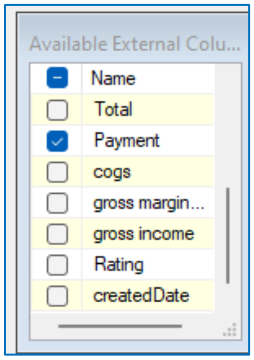


Sales_Stage to PaymentType_NDS

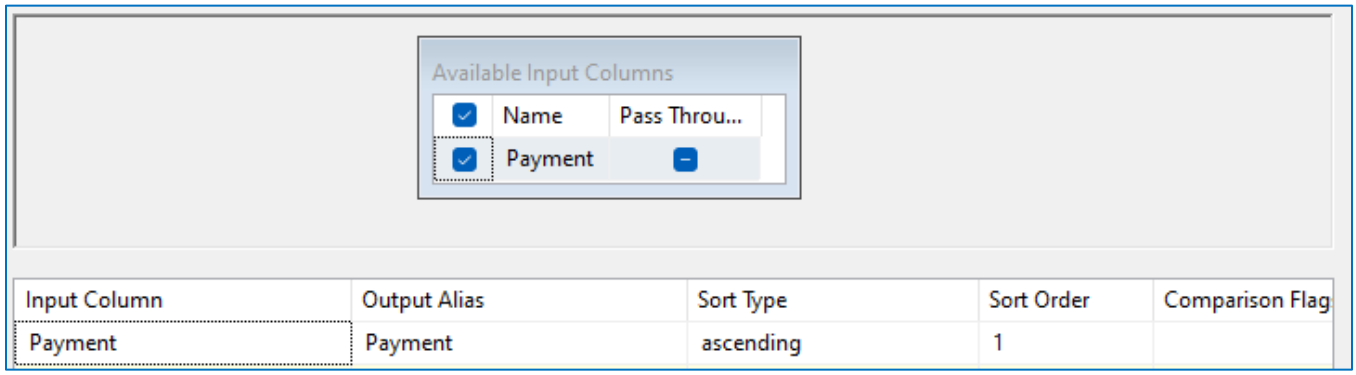


Trong dữ liệu Source không có sẵn bảng PaymentType, thuộc tính của nó nằm trong bảng Sales_Stage vì vậy cần phải sort là vấy các giá trị không trùng sau đó mới loopup và làm như thông thường

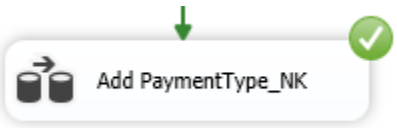
Ngoài ra không cần lấy hết dữ liệu từ bảng Sales_Stage, chỉ lấy dữ liệu mình cần

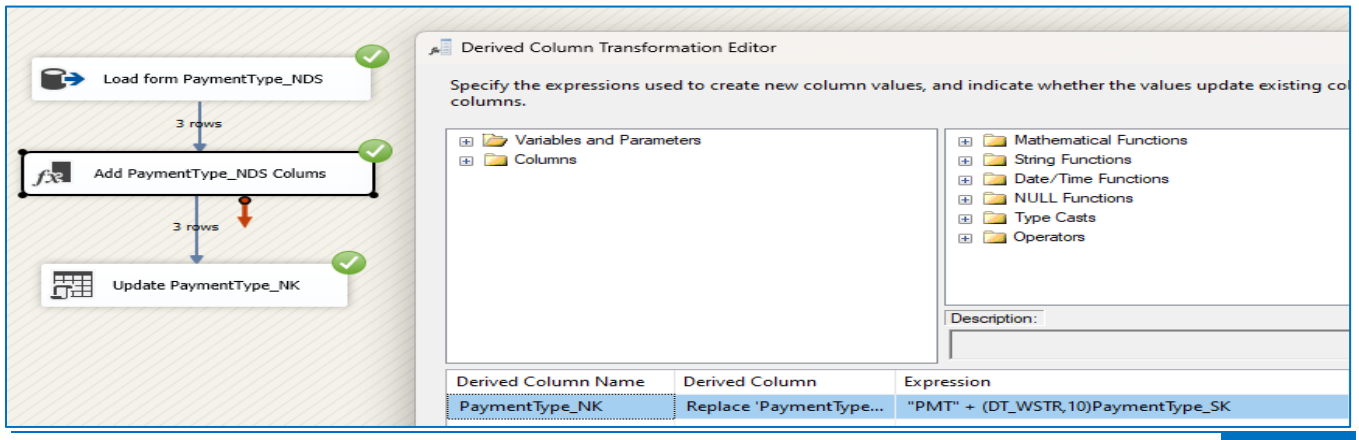


Cấu hình sắp xếp và lấy các giá trị không trùng

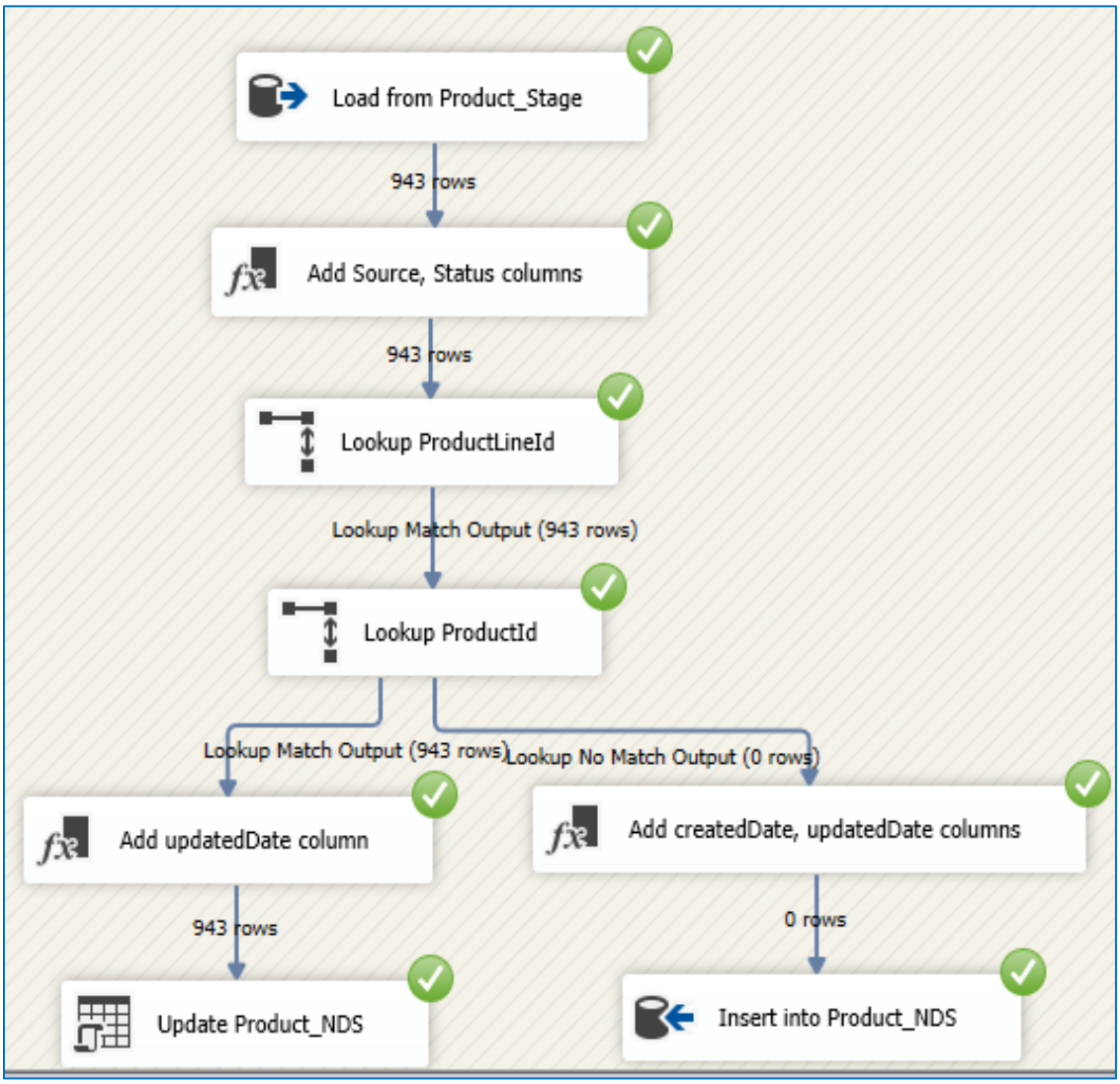


Bản thân PaymentType_NDS không không có khoá tự nhiên, vì vậy nhóm đã tự thêm khoá tự nhiên cho PaymentType_NDS theo công thức: "PMT" + ${PaymementType_SK}



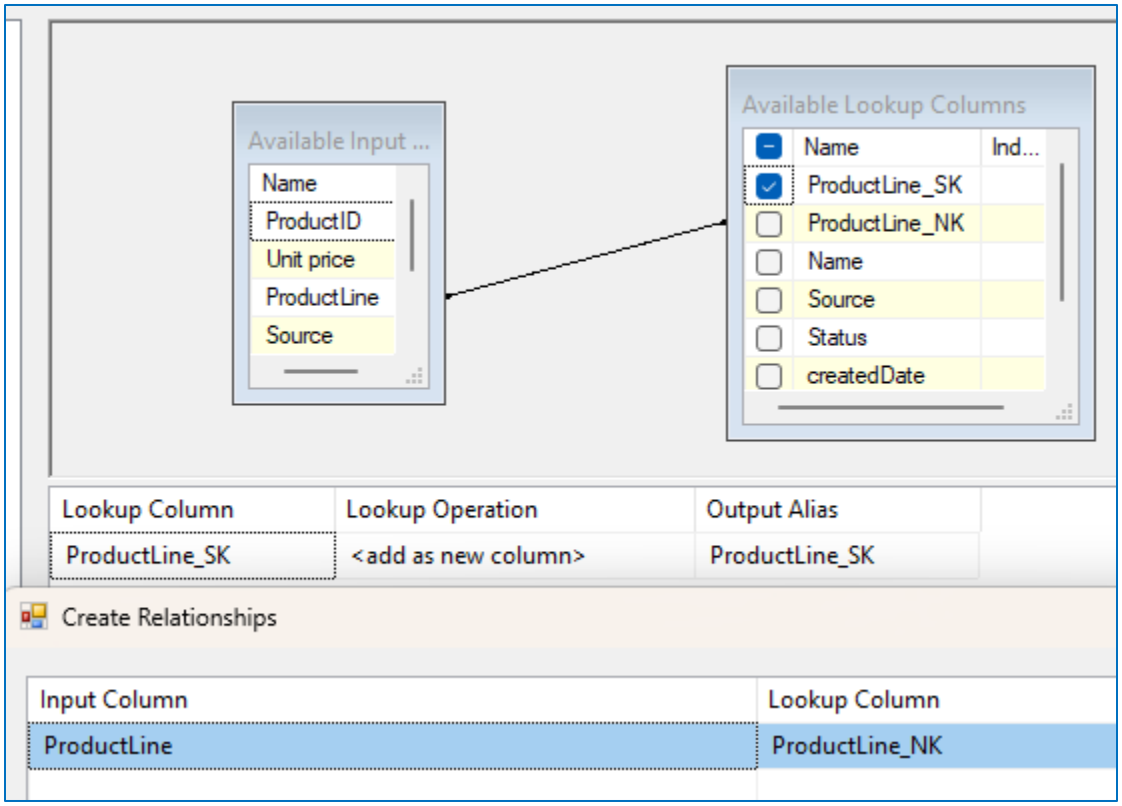


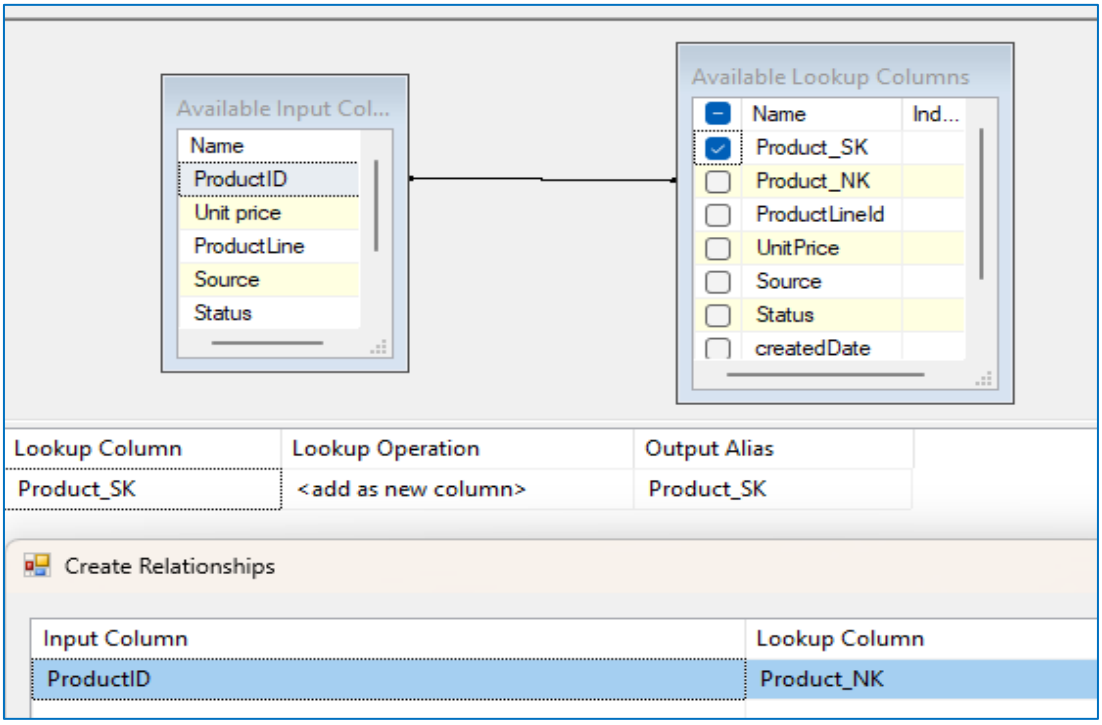
Product_Stage to Product_NDS



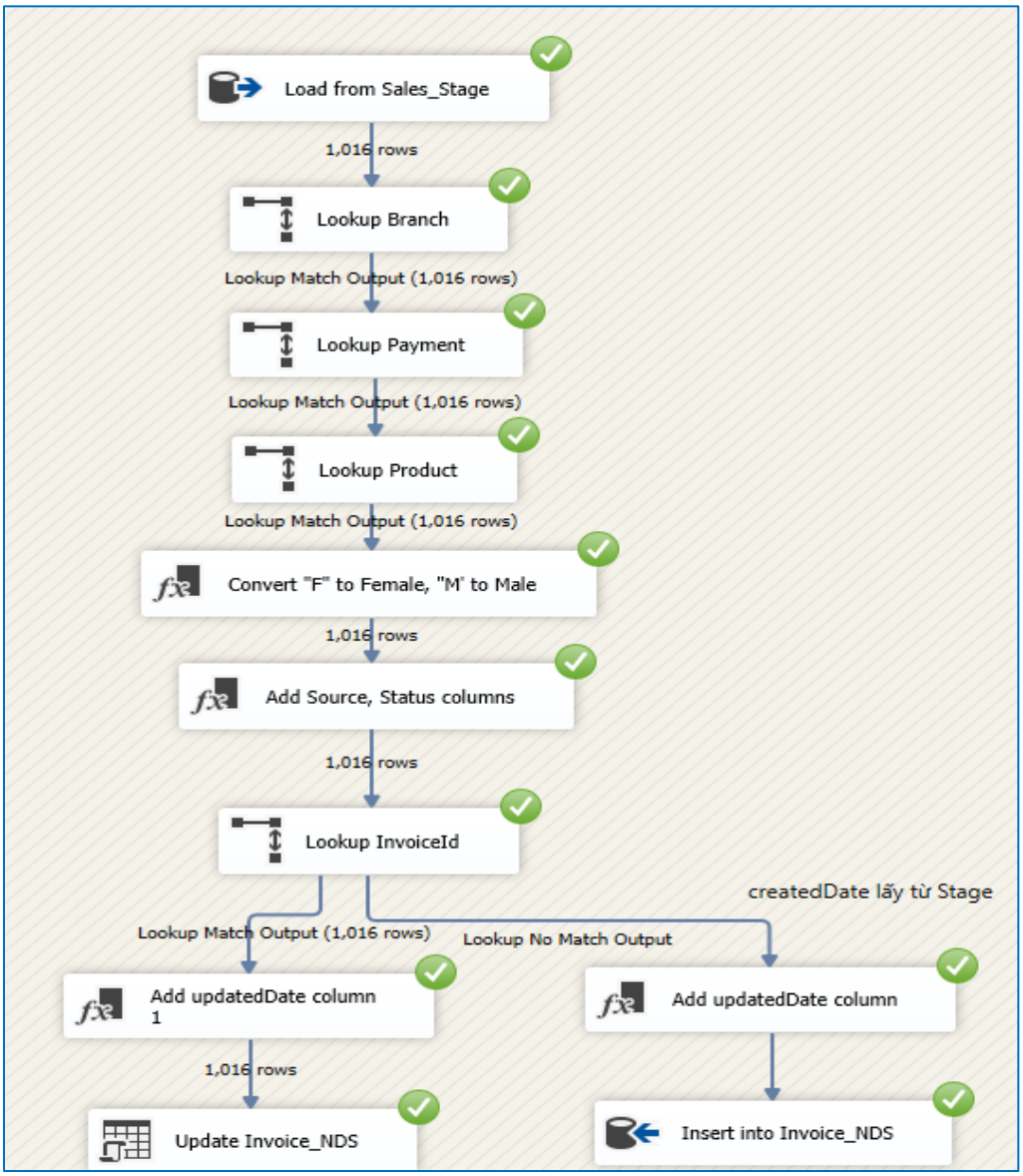
Vì trong thiết kế NDS, Product tham chiếu đến ProductLine nên cần phải Lookup ProductLine trước

Cấu hình lookup



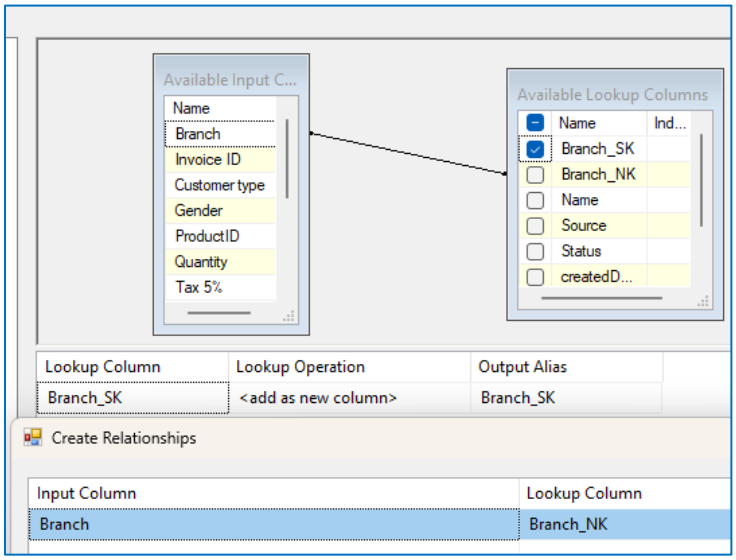


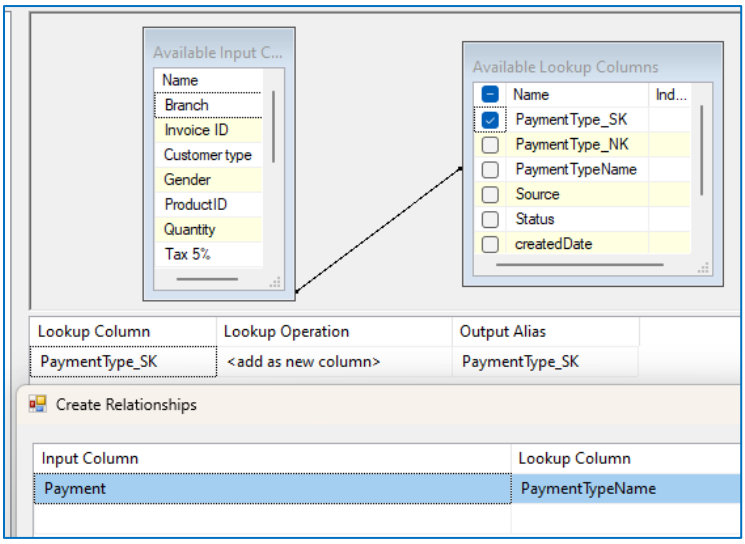
Sales_Stage to Invoice_NDS

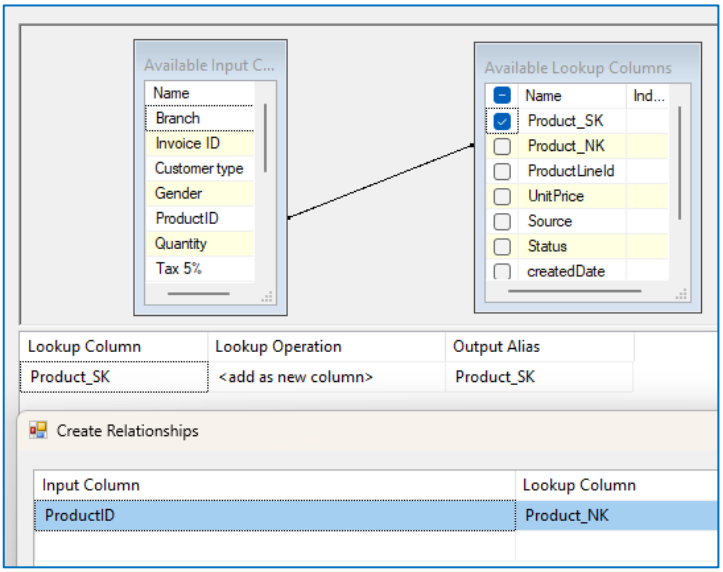


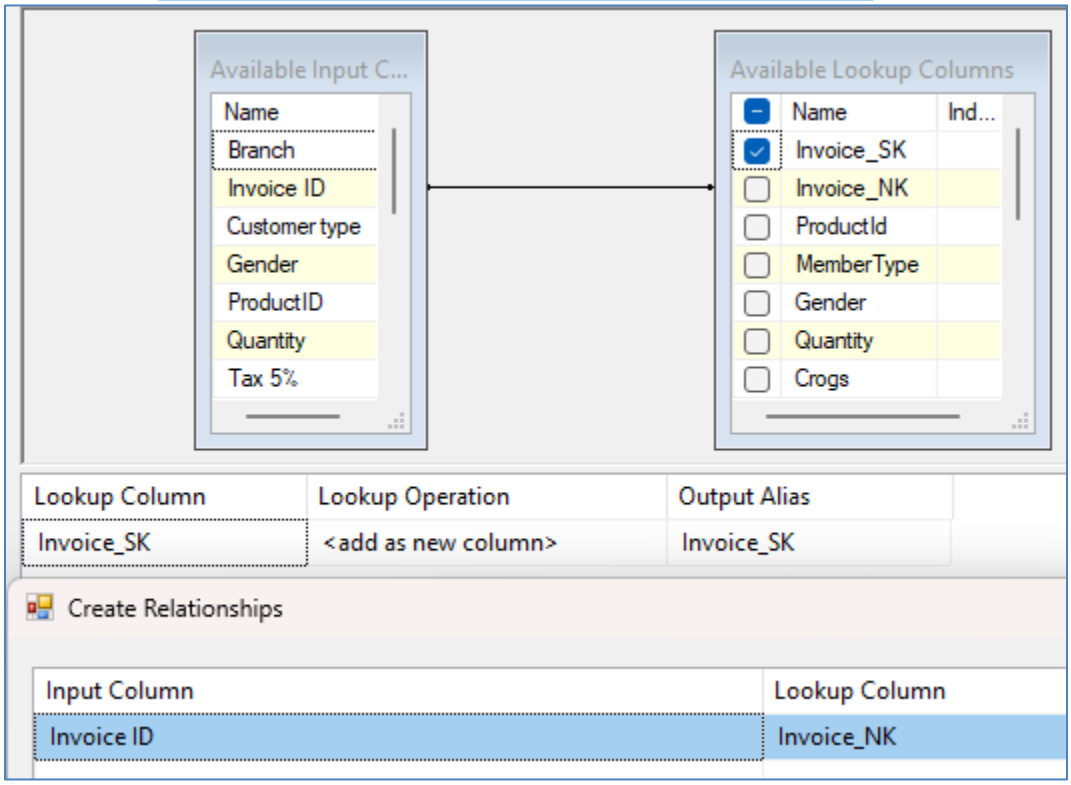
Lookup các bảng đó trước trước khi Lookup xem rằng dữ liệu trong Stage đã có trog NDS chưa Trong thiết kế NDS, Invoice tham chiếu đến Branch_NDS, Payment_NDS, Product_NDS, vì vậy cần phải

Cấu hình lookup



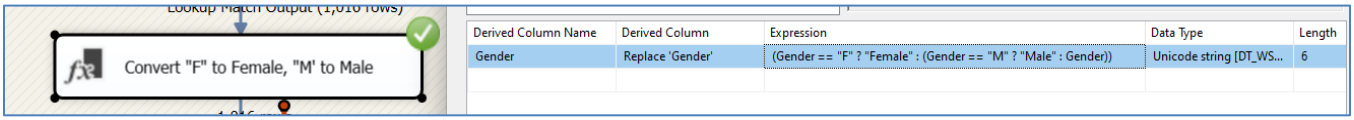



Available Input C...
Name
Branch
Invoice ID
Customer type
Gender
ProductID
Quantity
Tax 5%
Available Lookup Columns
Name
Ind...
Product_SK
Product_NK
ProductLineId
UnitPrice
Source
Status
createdDate
Lookup Column
Lookup Operation
Output Alias
Product_SK
<add as new column>
Product_SK
Create Relationships
Input Column
Lookup Column
ProductID
Product_NK


Available Input C...
Name
Branch
Invoice ID
Customer type
Gender
ProductID
Quantity
Tax 5%
Available Lookup Columns
Name
Ind...
Invoice_SK
Invoice_NK
ProductId
MemberType
Gender
Quantity
Crogs
Lookup Column
Lookup Operation
Output Alias
Invoice_SK
<add as new column>
Invoice_SK
Create Relationships
Input Column
Lookup Column
Invoice ID
Invoice_NK

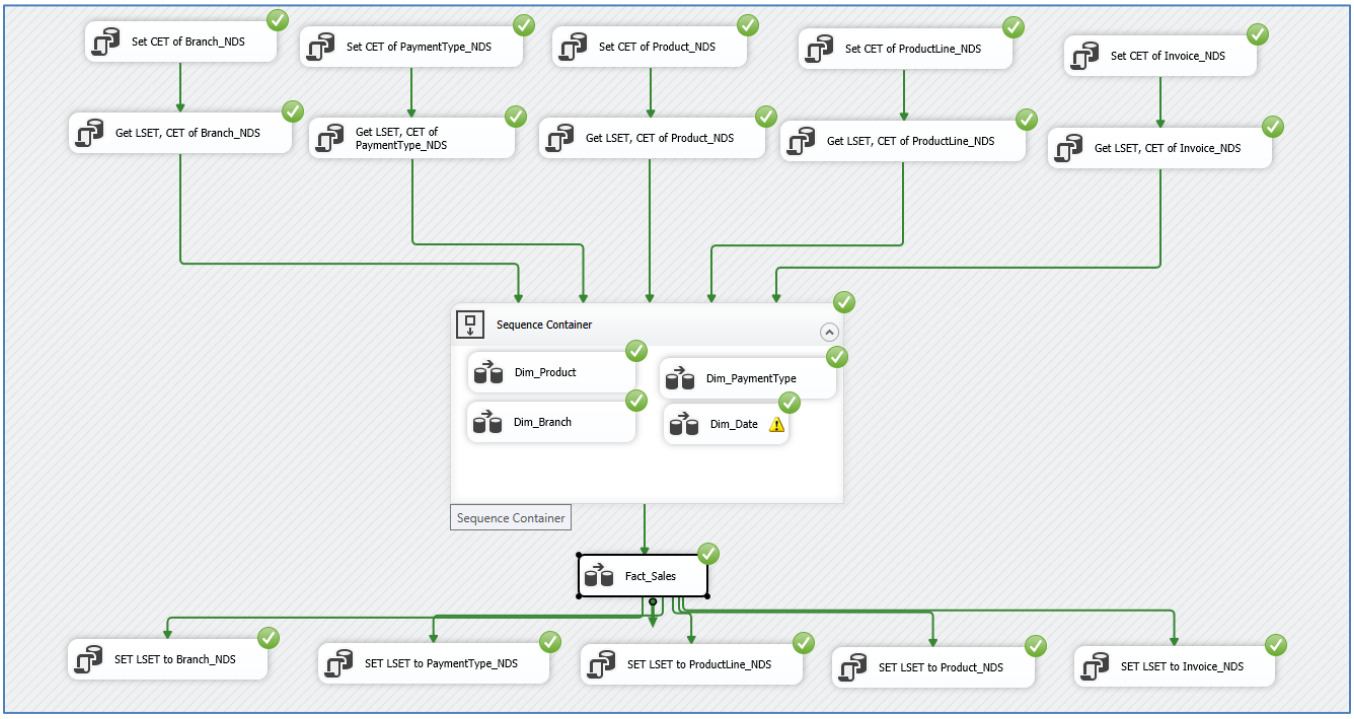
Ngoài ra trong dữ liệu từ Source không đồng nhất, cụ thể là trường "Gender" có các dòng "F", "Female", "M", "Male" vì vậy cần phải chuyển đổi trường "Gender" đồng nhất. Nhóm quyết định dữ liệu đồng nhất là: "Female" và "Male"

Cấu hình chuyển đổi:

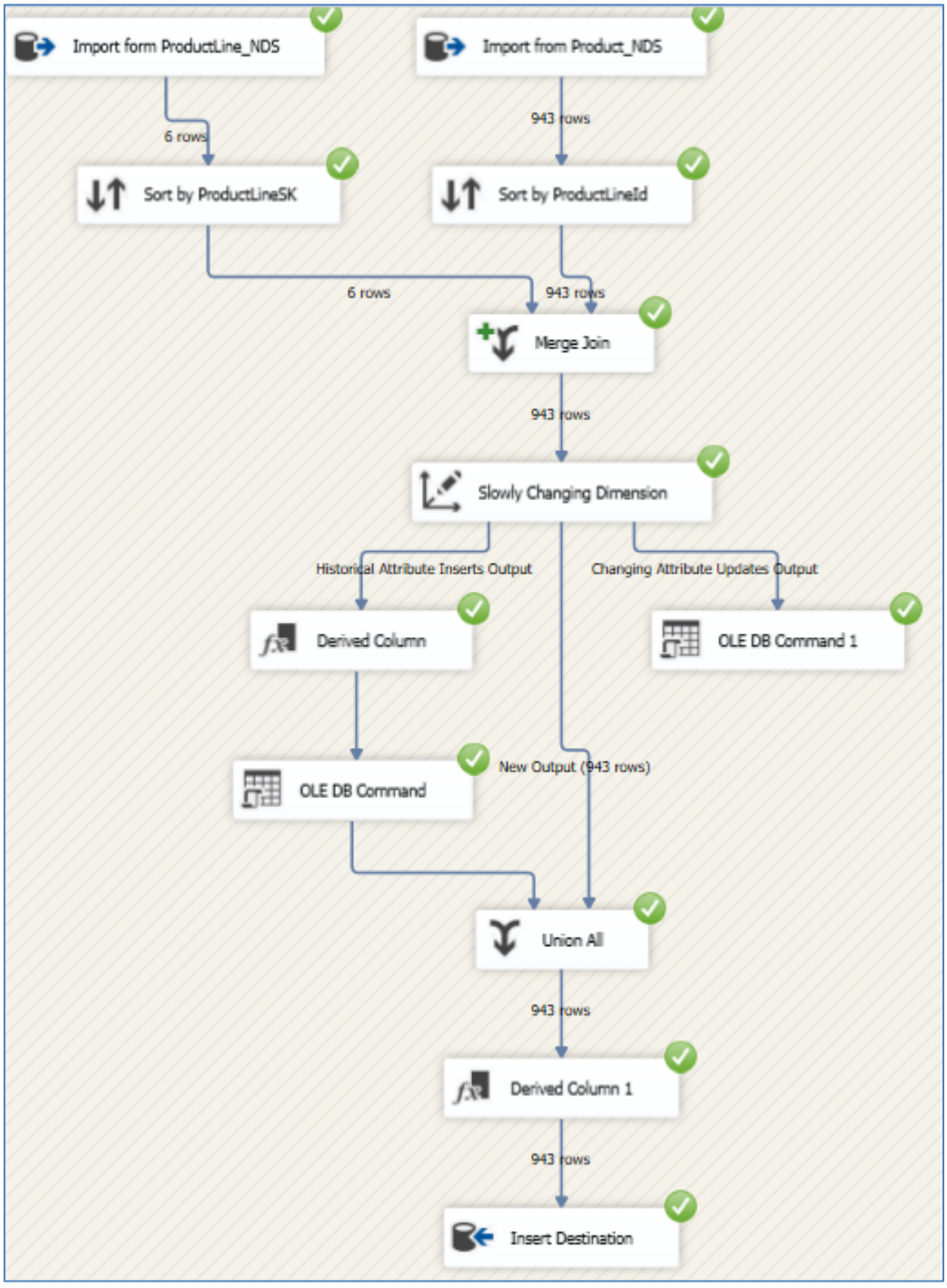


ETL NDS to DDS

Control Flow

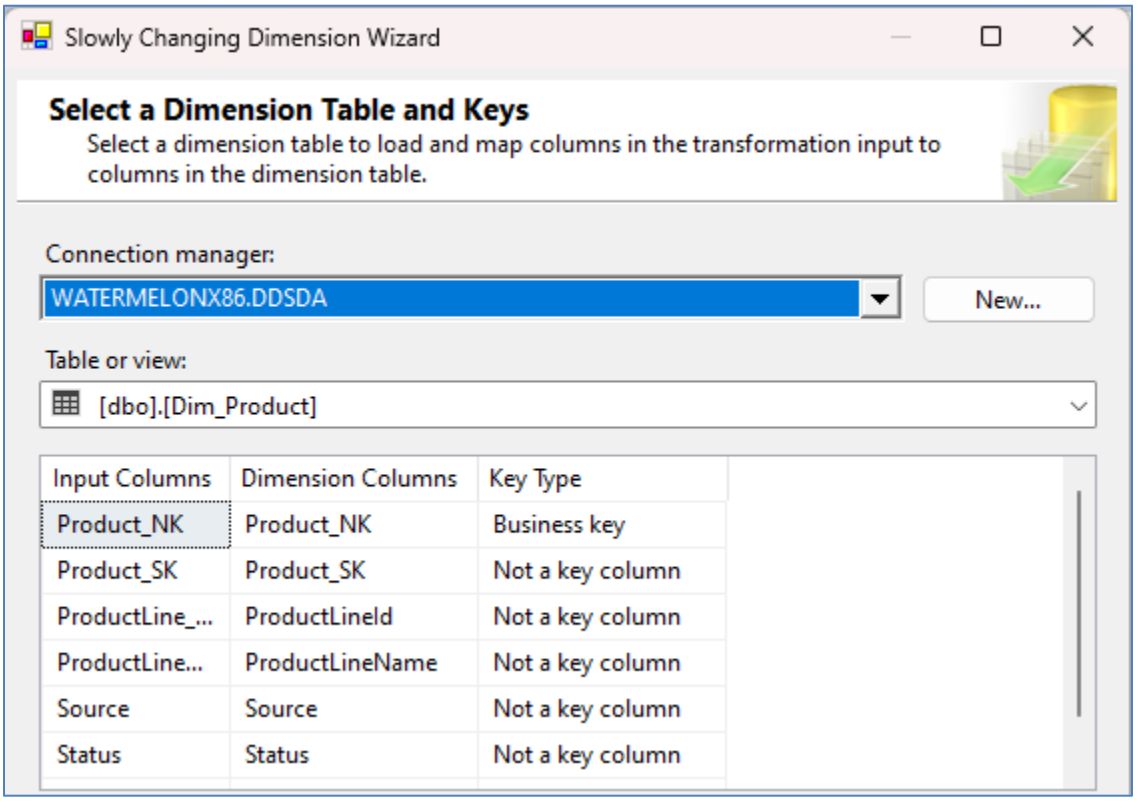


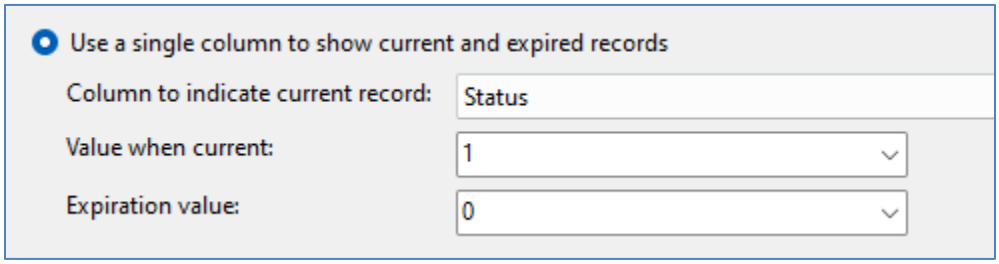
Data Flow

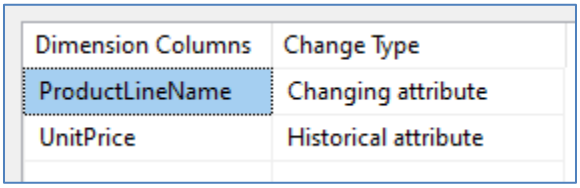


Dim_Product

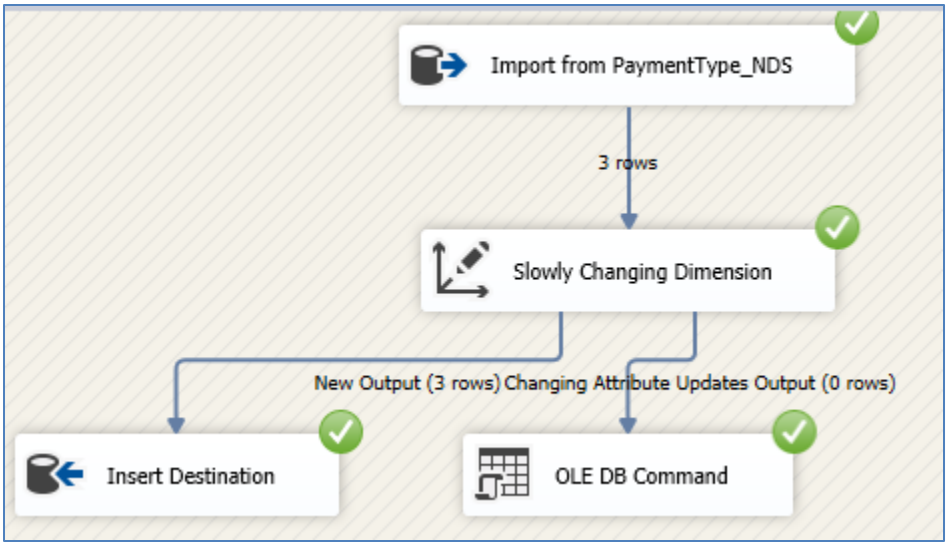
Cấu hình chiều thay đổi chậm của Dim_Product



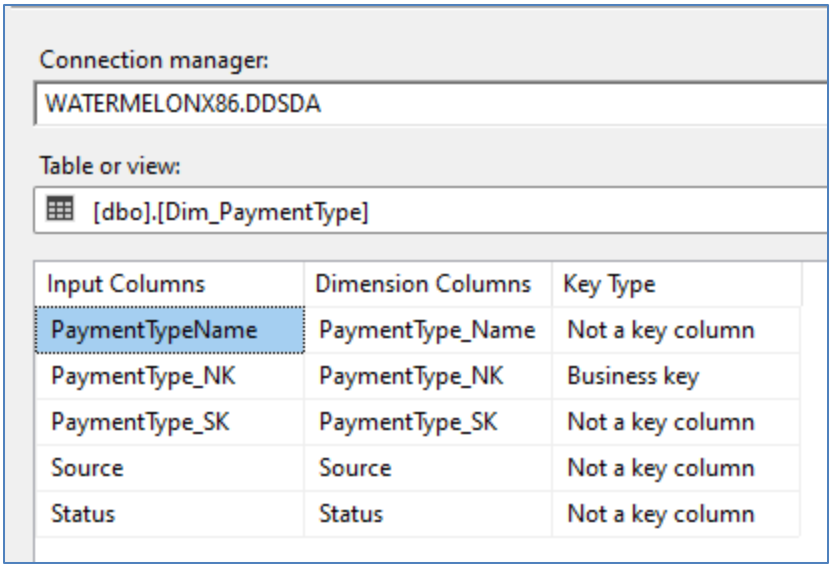




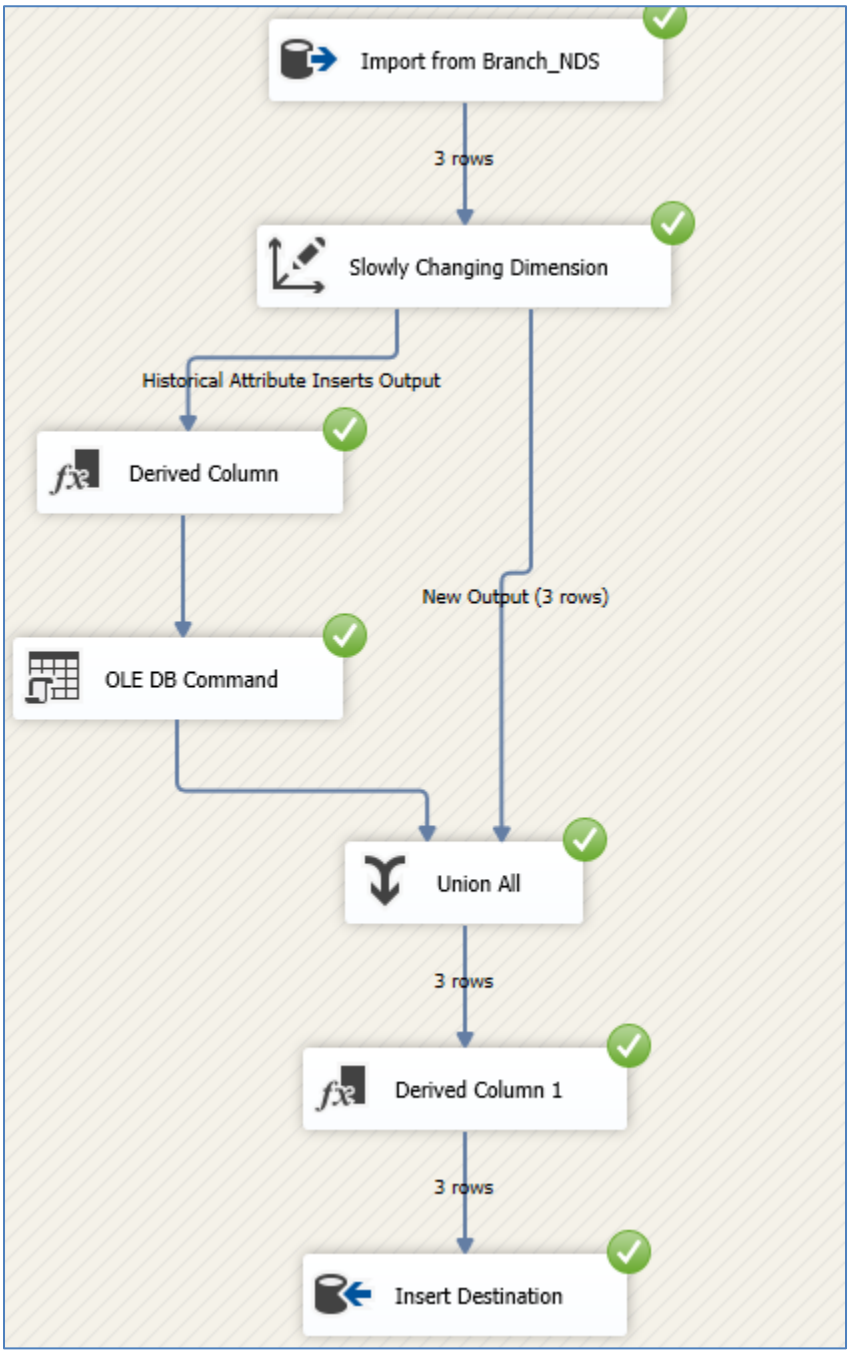
Dim PaymentType



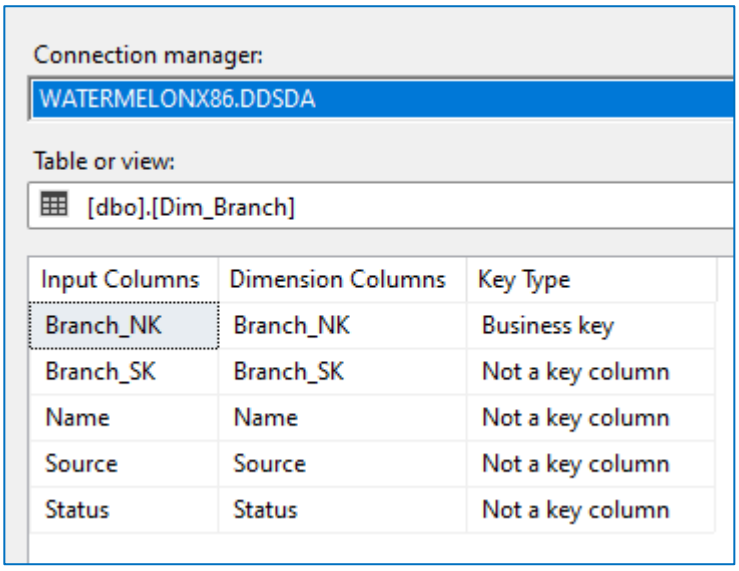
Cấu hình chiều hay đổi chậm của Dim PaymentType

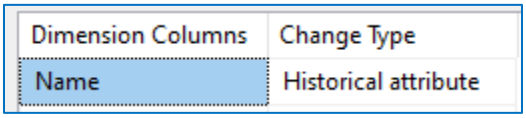


Dim Branch

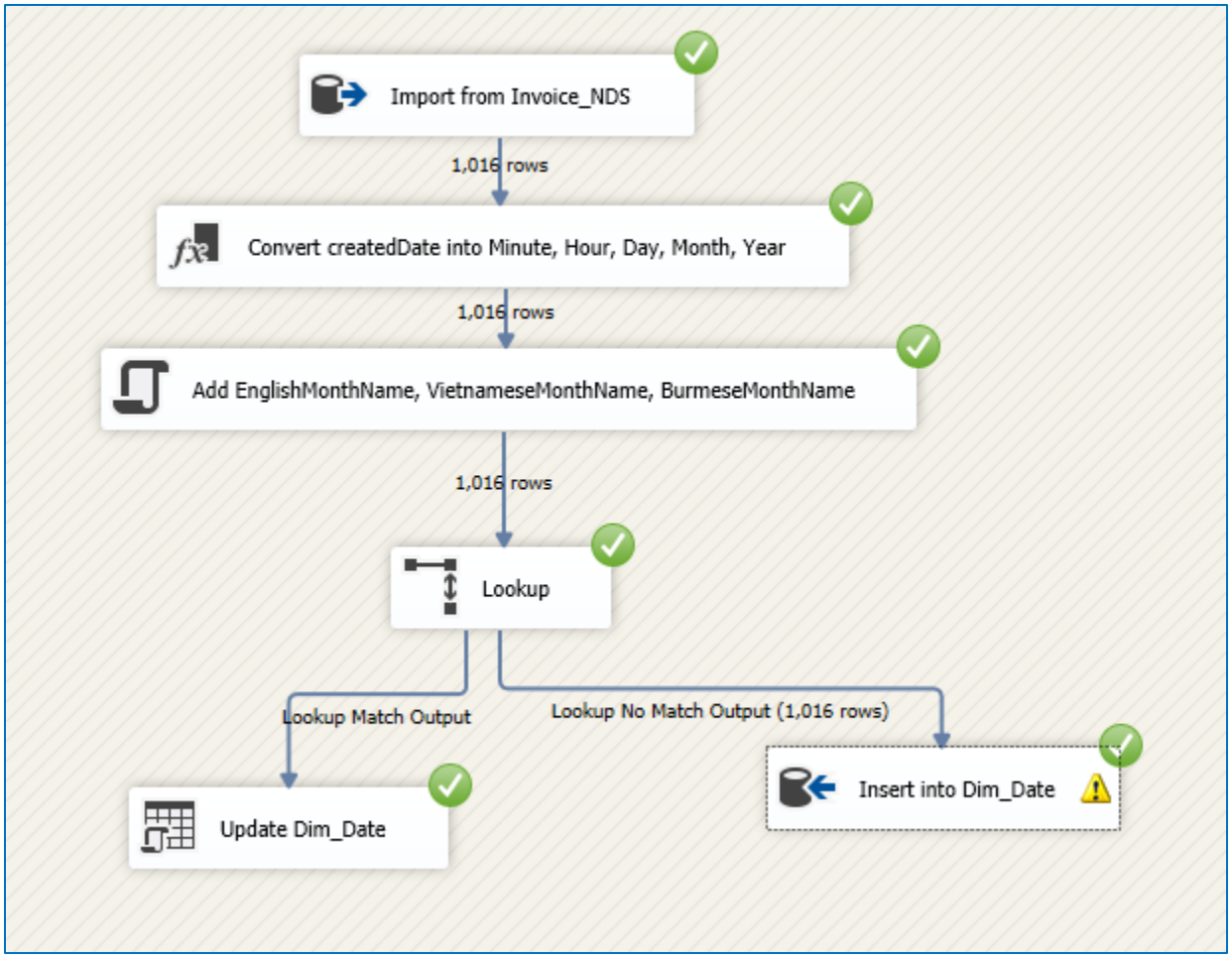

Import from Branch_NDS
3 rows
Slowly Changing Dimension
Historical Attribute Inserts Output
Derived Column
New Output (3 rows)
OLE DB Command
Union All
3 rows
Derived Column 1
3 rows
Insert Destination

Cấu hình chiều thay đổi chậm của Dim Branch

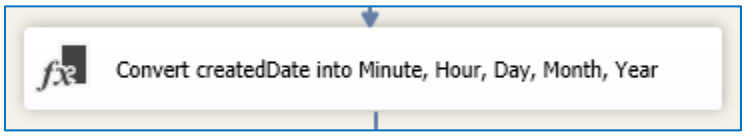




Dim Date

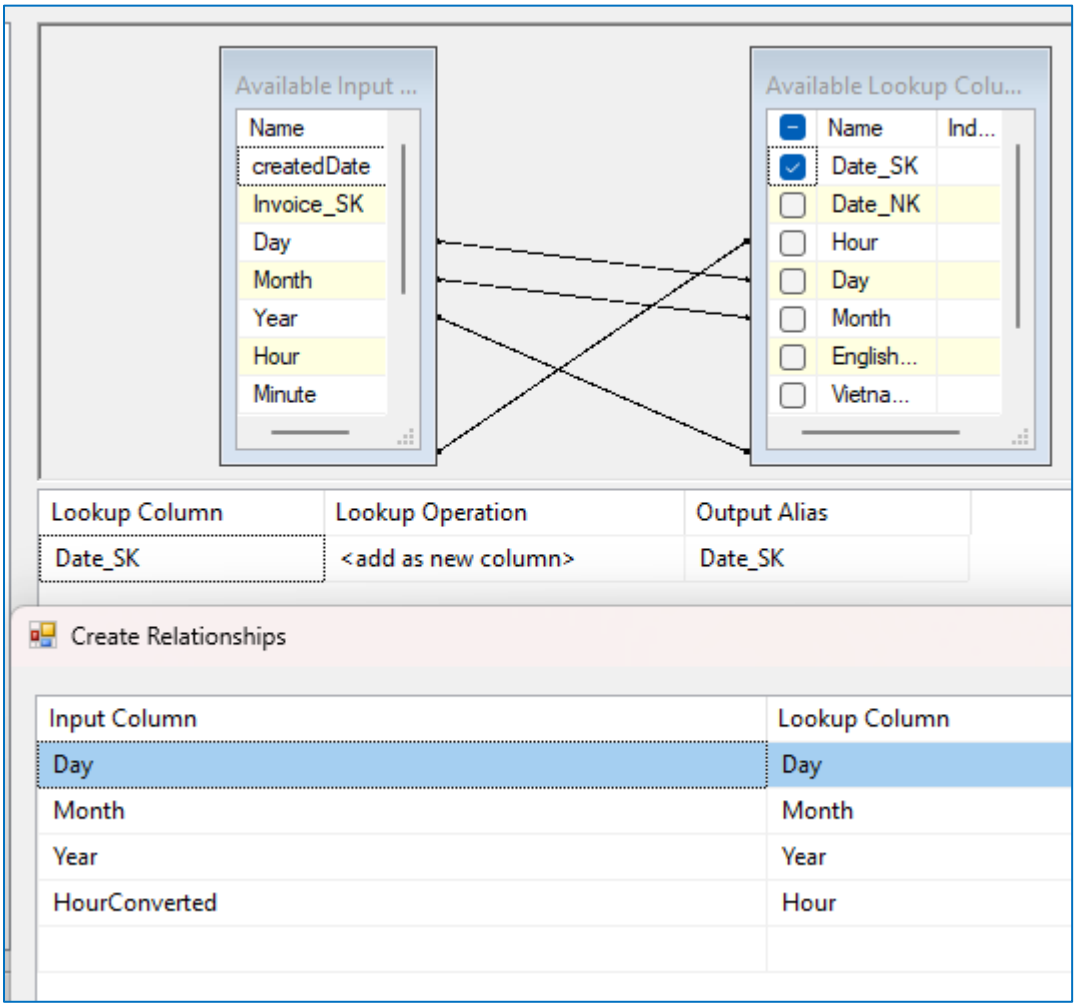


- Dim Date không có chiều thay đổi chậm vì nhóm suy nghĩ rằng trong thực tế khi 1 hoá đơn được tạo ra thì trường date sẽ được hệ thống tạo và vì vậy gần như không có sai sót và nhu cầu thảy đổi
- Sử dụng trường "createdDate" và các hàm về date để chuyển đổi dữ liệu theo mong muốn

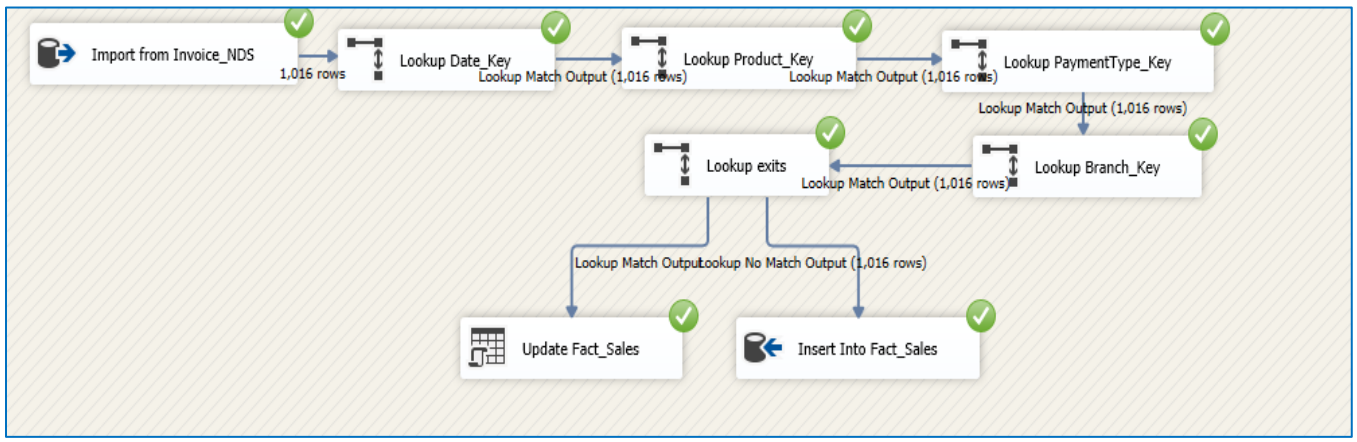


Derived Column Name	Derived Column	Expression
Day	<add as new column>	DAY(createdDate)
Month	<add as new column>	MONTH(createdDate)
Year	<add as new column>	YEAR(createdDate)
Hour	<add as new column>	DATEPART("hh",createdDate)
Minute	<add as new column>	DATEPART("mi",createdDate)

Lookup của Dim Date



Fact_Sales

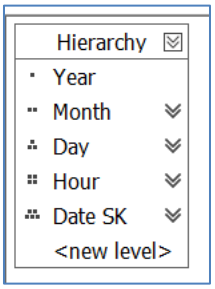


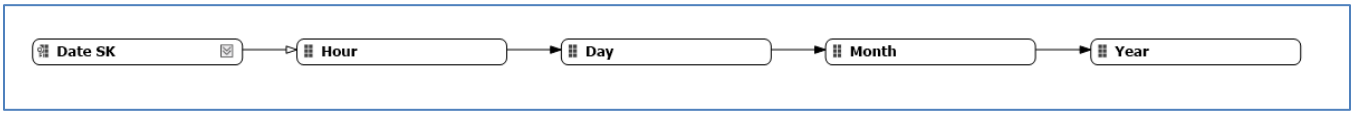
Trong thiết kế DDS bảng Fact tham chiếu đến các Dim vì vậy phải lookup toàn bộ các Dim mà bảng Fact tham chiếu, sau đó mới lookup để tìm xem các dòng dữ liệu từ NDS đã có trong DDS chưa

2.3 MDX, OLAP

OLAP

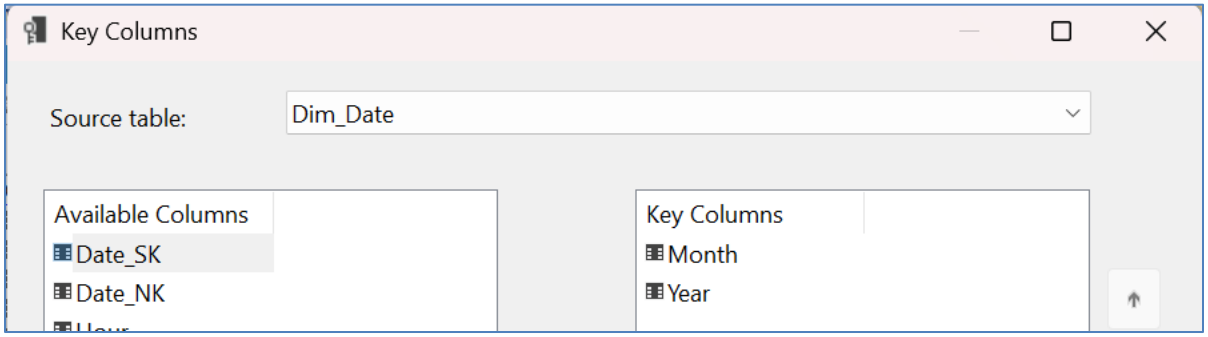
Phân cấp chiều Dim_Date



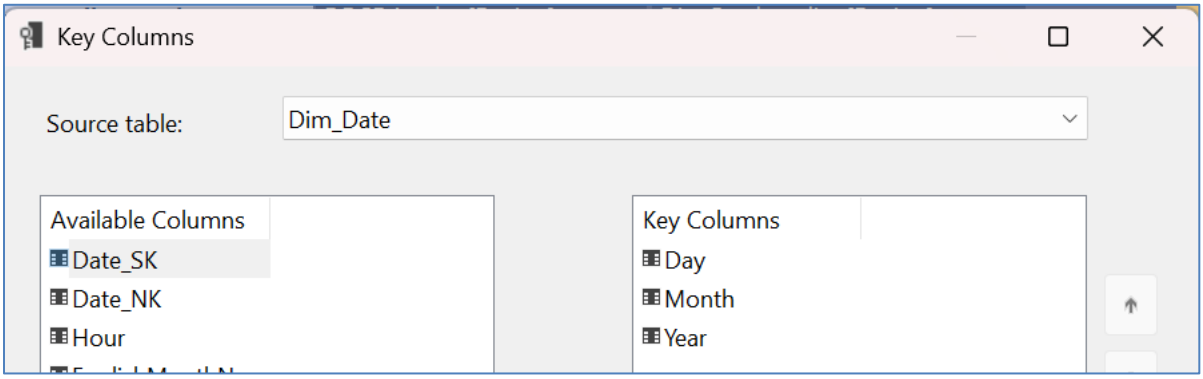


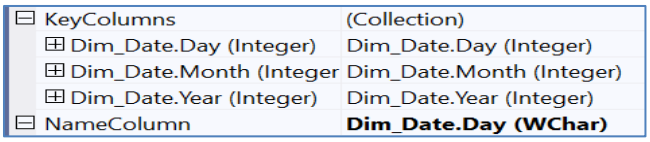
Cấu hình phân cấp của Dim_Date

Thuộc tính Month:

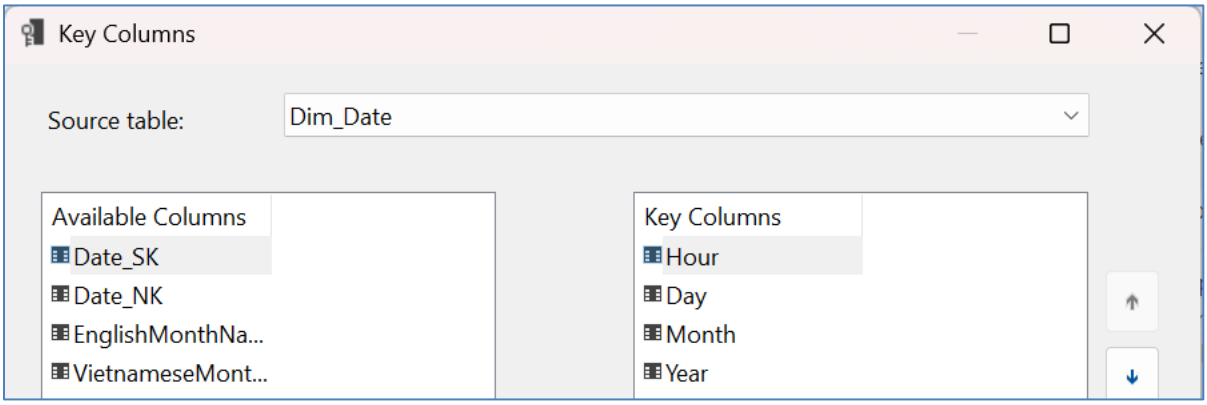


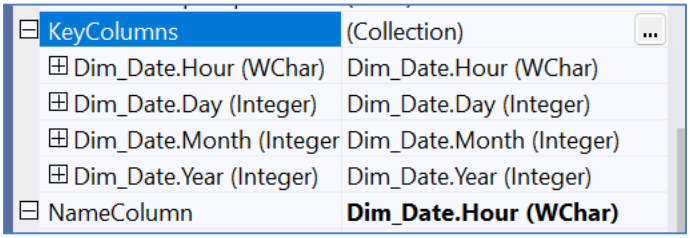
Thuộc tính Day:





Thuộc tính Hour



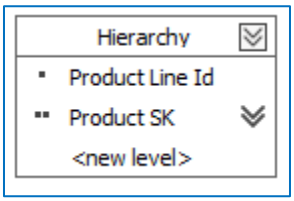


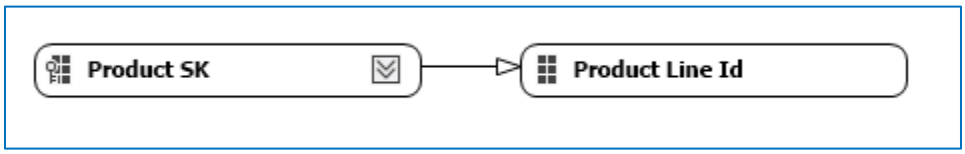
Kiểm tra phân cấp date

Year	Month	Day	Hour
2019	January	1	10:39:00
2019	January	1	11:36:00
2019	January	1	11:40:00
2019	January	1	11:43:00
2019	January	1	13:55:00
2019	January	1	14:42:00
2019	January	1	14:47:00
2019	January	1	15:51:00
2019	January	1	19:07:00
2019	January	1	19:31:00
2019	January	1	19:48:00
2019	January	1	20:26:00
2019	January	2	13:00:00
2019	January	2	13:40:00
2019	January	2	15:24:00
2019	January	2	16:19:00
2019	January	2	16:57:00

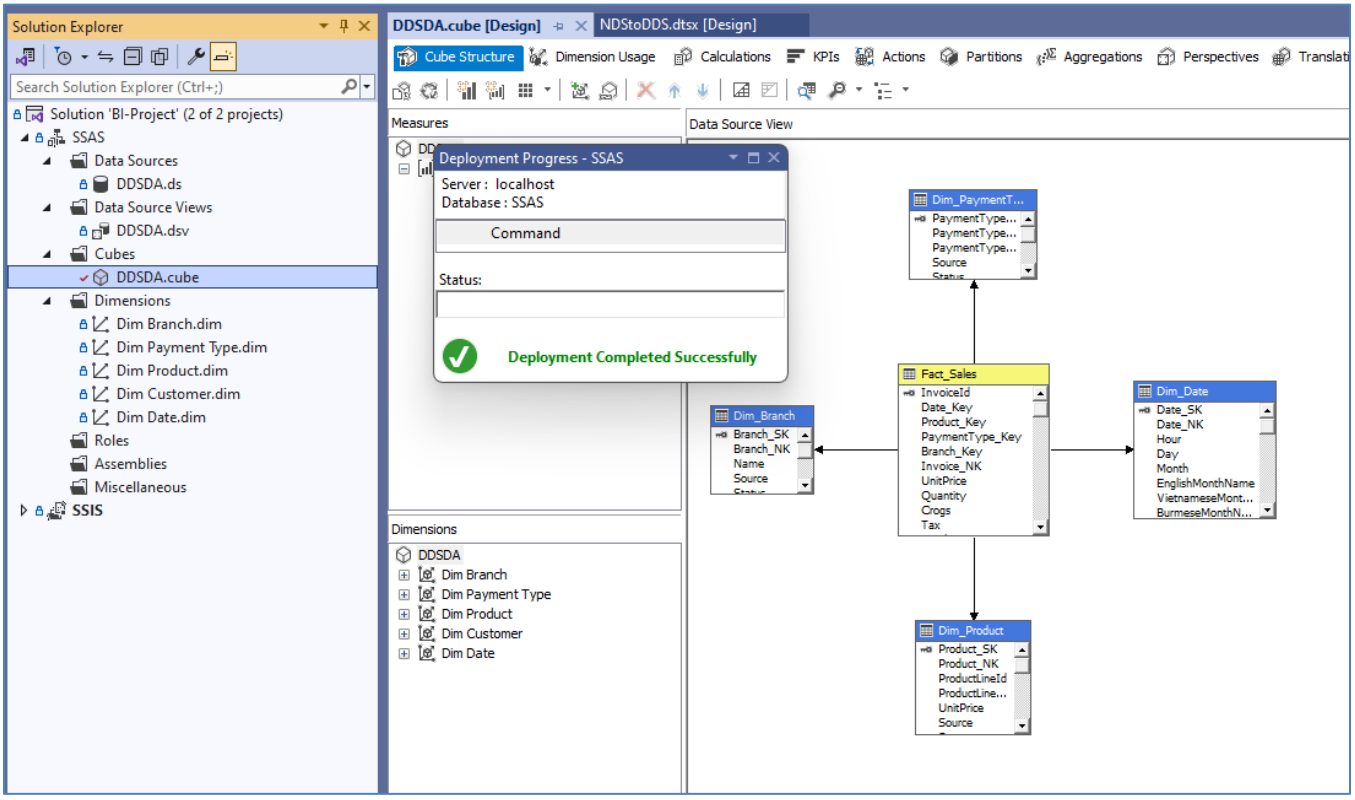
Nhóm phân cấp theo: giờ -> ngày -> tháng -> năm

Phân cấp chiều Dim_Product



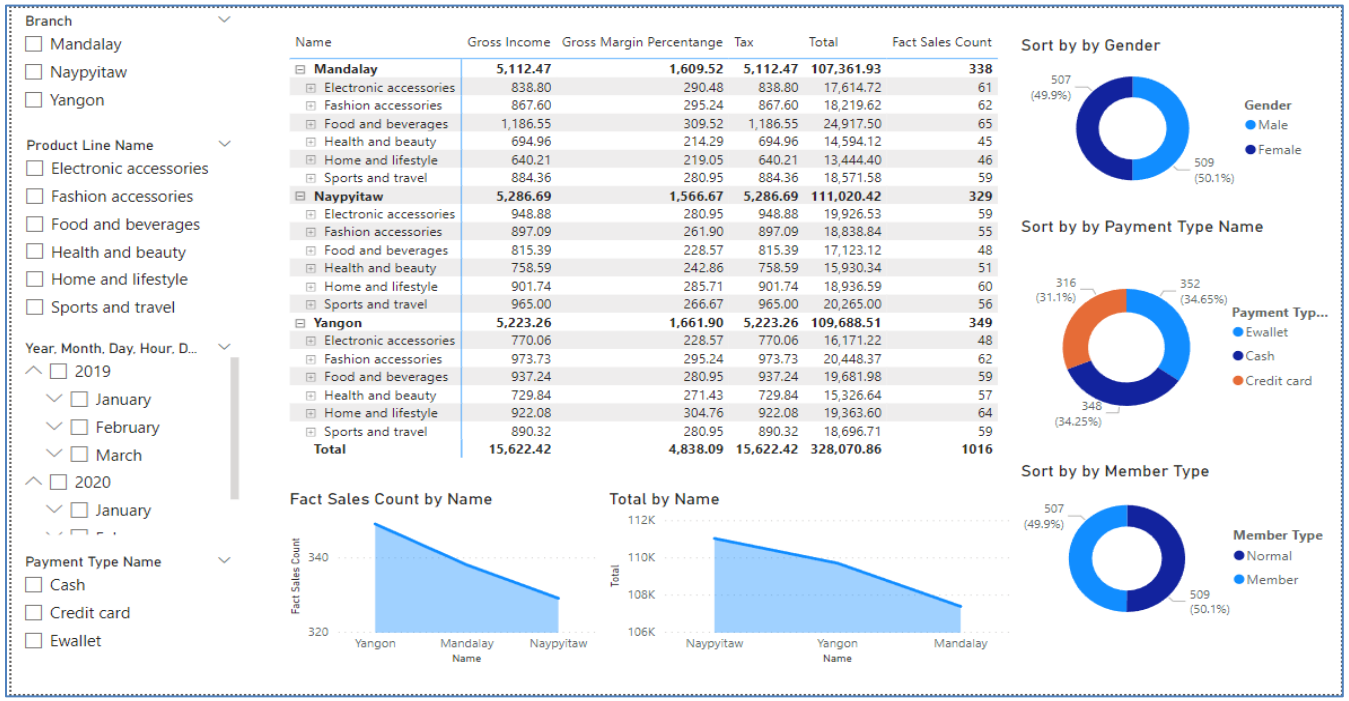


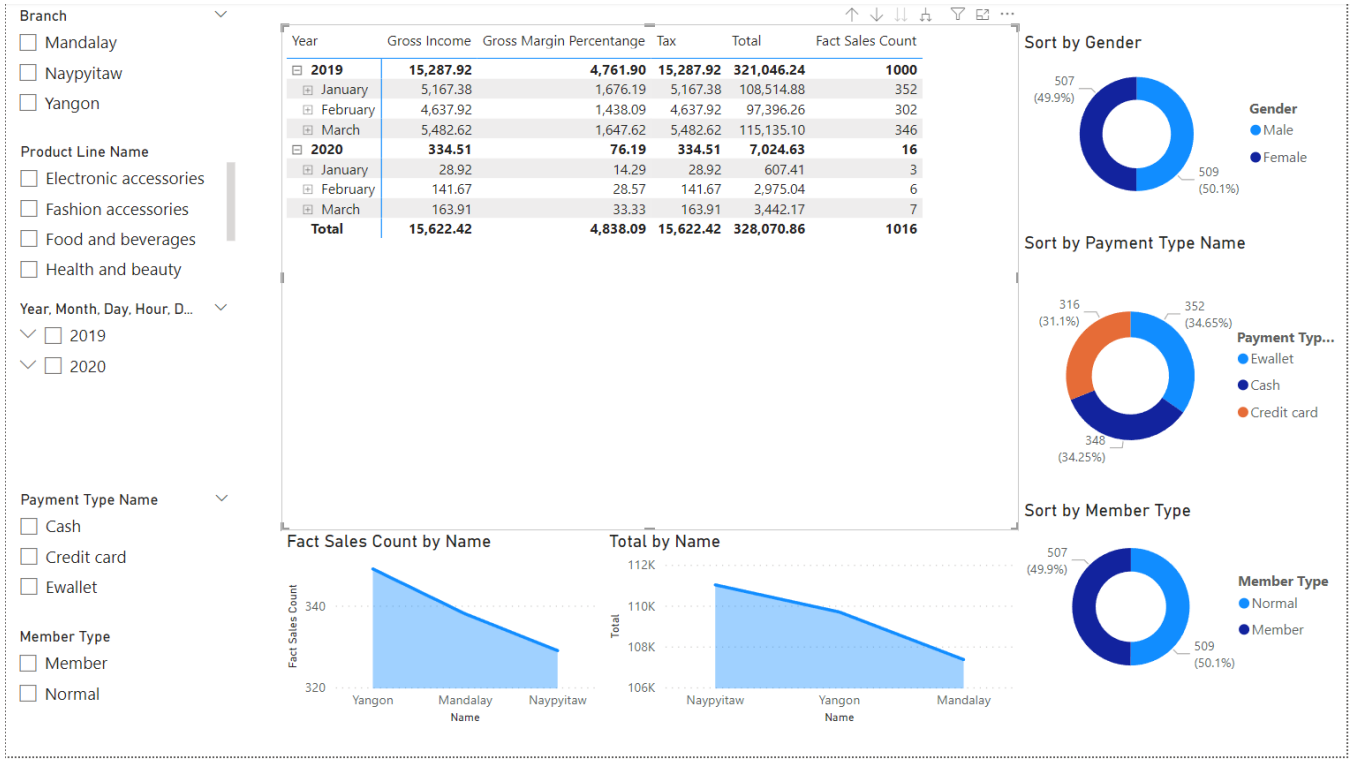
Kết quả process Cube



Dashboard

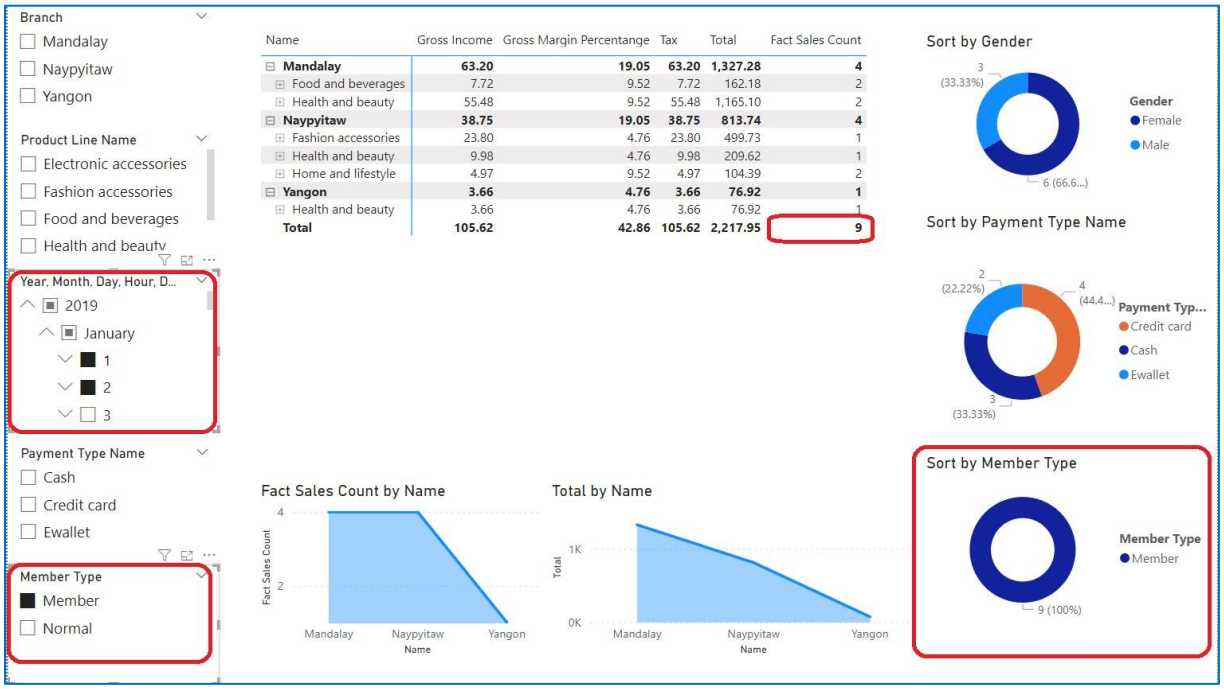
Tình hình mua hàng của khách hàng theo từng chi nhánh, từng loại sản phẩm, theo thời gian, hình thức thanh toán



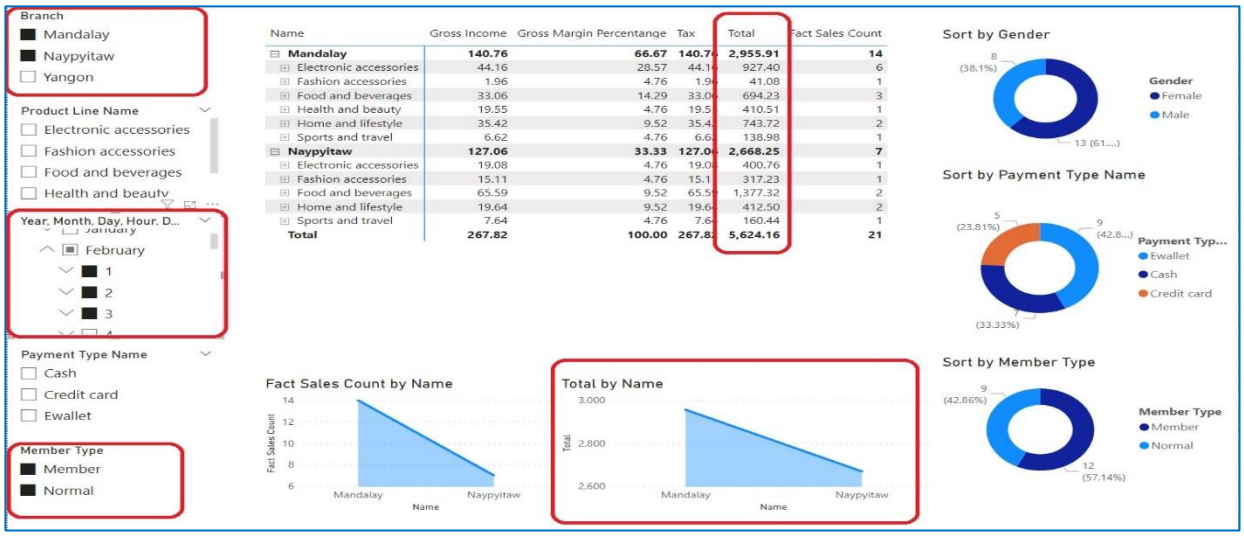


Một các các nhu cầu truy vấn có thể xem từ Dashboard

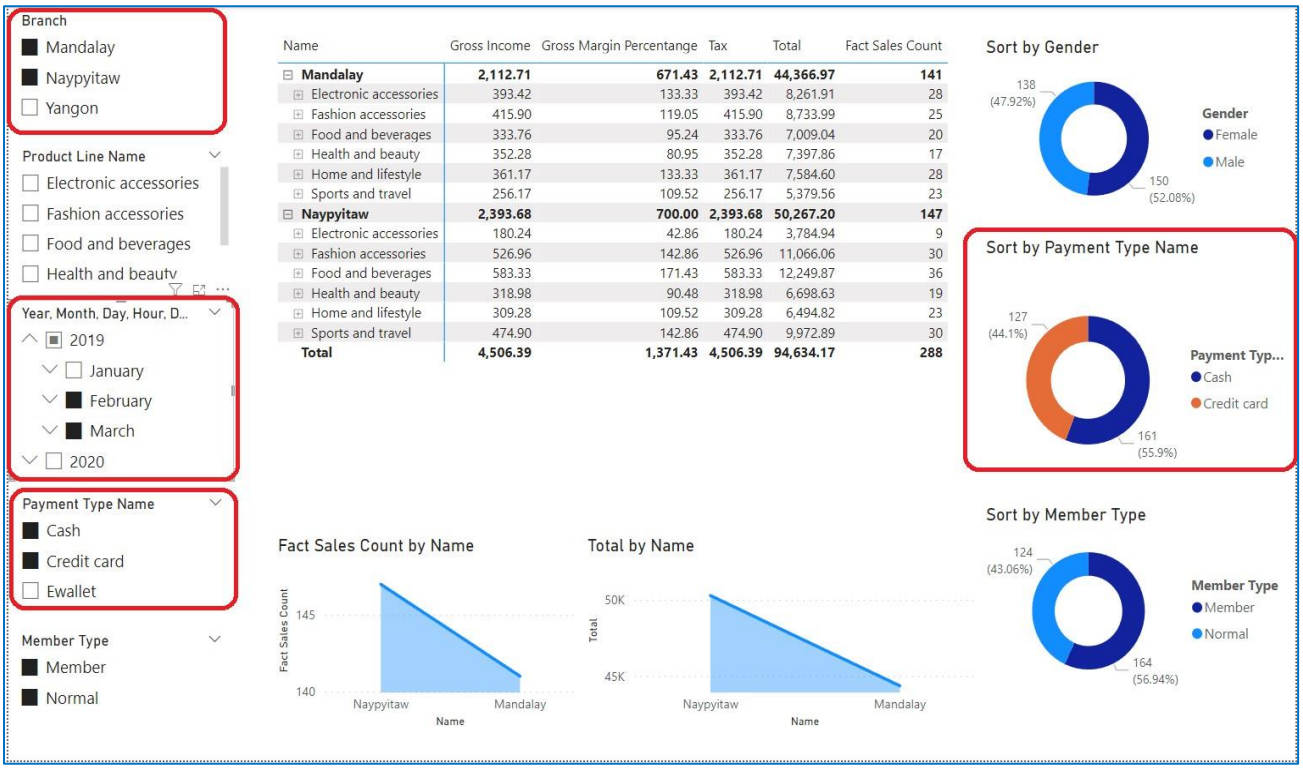
- Thống kế số thành viên mua hàng theo ngày tháng năm: Chọn slicer Date, chọn slicer MemberTyper và theo dõi Donut chart: Sort by Member Type



- Thống kê doanh thu của khách hàng (member, normal) theo ngày, tháng, năm và theo chi nhánh: Chọn slicer Date, chọn slicer Branch, xem thông tin trên cột Total và chart Total by Name



- Thống kê số lượng khách thanh toán theo cash/ debit/… ở từng chi nhánh theo từng tháng trong năm. Chọn slicer Branch, chọn slicer Date, xem thông tin cột Fact Sales Count và donut chart Sort by Payment Type Name



MDX

1. Thống kê số thành viên mua hàng theo ngày, tháng, năm

- MDX query

```
SELECT
  NON EMPTY { [Measures].[Fact Sales Count] } ON COLUMNS,
  NON EMPTY {
    (
      [Dim Date].[Year].[Year].ALLMEMBERS *
      [Dim Date].[Month].[Month].ALLMEMBERS *
      [Dim Date].[Day].[Day].ALLMEMBERS *
      [Dim Customer].[Member Type].[Member Type].ALLMEMBERS
    )
  } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
FROM (
  SELECT
    ( { [Dim Customer].[Member Type].&[Member] } ) ON COLUMNS
  FROM [DDSDA]
)
```

- Kết quả:

				Fact Sales Count
2019	January	27	Member	6
2019	January	28	Member	8
2019	January	29	Member	6
2019	January	30	Member	3
2019	January	31	Member	8
2019	February	1	Member	5
2019	February	2	Member	8
2019	February	3	Member	8
2019	February	4	Member	6
2019	February	5	Member	11
2019	February	6	Member	9
2019	February	7	Member	5
2019	February	8	Member	6
2019	February	9	Member	7
2019	February	10	Member	8
2019	February	11	Member	2
2019	February	12	Member	5
2019	February	13	Member	6
2019	February	14	Member	5

2. Thống kê doanh thu của khách hàng (member, normal) theo ngày, tháng, năm và theo chi nhánh.

- MDX query

```
SELECT
  NON EMPTY { [Measures].[Total] } ON COLUMNS,
  NON EMPTY {
    (
      [Dim Date].[Year].[Year].ALLMEMBERS *
      [Dim Date].[Month].[Month].ALLMEMBERS *
      [Dim Date].[Day].[Day].ALLMEMBERS *
      [Dim Branch].[Name].[Name].ALLMEMBERS *
      [Dim Customer].[Member Type].[Member Type].ALLMEMBERS
    )
  } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
FROM [DDSDA]
```

- Kết quả

					Total
2019	January	1	Mandalay	Member	888.615
2019	January	1	Mandalay	Normal	648.081
2019	January	1	Naypyitaw	Member	572.376
2019	January	1	Naypyitaw	Normal	264.789
2019	January	1	Yangon	Member	1292.634
2019	January	1	Yangon	Normal	1078.686
2019	January	2	Mandalay	Member	383.7645
2019	January	2	Mandalay	Normal	779.037
2019	January	2	Naypyitaw	Member	266.028
2019	January	2	Naypyitaw	Normal	209.622
2019	January	2	Yangon	Member	262.458
2019	January	2	Yangon	Normal	44.5935
2019	January	3	Mandalay	Member	520.8
2019	January	3	Mandalay	Normal	495.894
2019	January	3	Naypyitaw	Member	124.026
2019	January	3	Yangon	Normal	937.4085
2019	January	4	Mandalay	Member	364.3605
2019	January	4	Mandalay	Normal	146.223
2019	January	4	Naypyitaw	Member	629.8425

3. Thống kê số lượng khách thanh toán theo cash/ debit/... ở từng chi nhánh theo từng tháng trong năm

- MDX query

```
SELECT
  NON EMPTY { [Measures].[Fact Sales Count] } ON COLUMNS,
  NON EMPTY {
    (
      [Dim Payment Type].[Payment Type Name].[Payment Type Name].ALLMEMBERS *
      [Dim Branch].[Name].[Name].ALLMEMBERS *
      [Dim Date].[Month].[Month].ALLMEMBERS *
      [Dim Date].[Year].[Year].ALLMEMBERS
    )
  } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
FROM [DDSDA]
```

- Kết quả

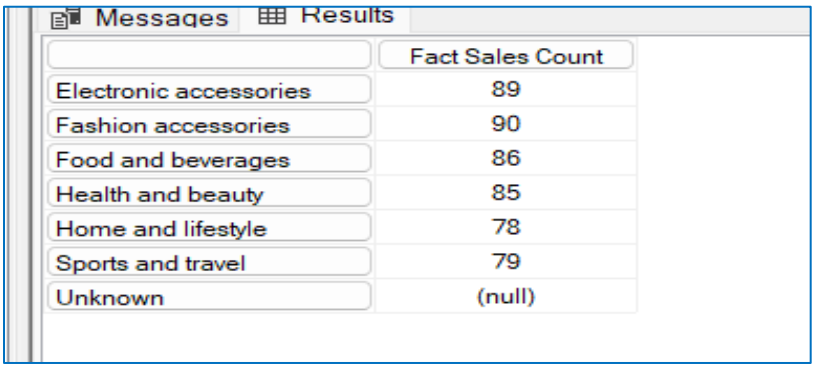
				Fact Sales Count
Cash	Yangon	January	2019	39
Cash	Yangon	February	2019	32
Cash	Yangon	February	2020	1
Cash	Yangon	March	2019	39
Credit card	Mandalay	January	2019	44
Credit card	Mandalay	February	2019	30
Credit card	Mandalay	March	2019	35
Credit card	Mandalay	March	2020	1
Credit card	Naypyitaw	January	2019	35
Credit card	Naypyitaw	February	2019	32
Credit card	Naypyitaw	March	2019	31
Credit card	Yangon	January	2019	34
Credit card	Yangon	January	2020	2
Credit card	Yangon	February	2019	28
Credit card	Yangon	March	2019	42
Credit card	Yangon	March	2020	2
Ewallet	Mandalay	January	2019	32
Ewallet	Mandalay	February	2019	35
Ewallet	Mandalay	March	2019	46

4. Thống kê lượng rating của khách hàng (member, normal) theo từng loại sản phẩm (ProductLine)

- Lượng rating của khách hàng member:
  - MDX query:

```
  SELECT
 [Measures].[Fact Sales Count] ON COLUMNS,
 [Dim Product].[Product Line Id].[Product Line Id].Members ON ROWS
FROM
        [DDSDA]
WHERE
       [Dim Customer].[Member Type].&[Member]
```

  - Kết quả:



Messages | Results

	Fact Sales Count
Electronic accessories	89
Fashion accessories	90
Food and beverages	86
Health and beauty	85
Home and lifestyle	78
Sports and travel	79
Unknown	(null)

- Lượng rating của khách hàng Normal:
  - MDX query:

```
SELECT
[Measures].[Fact Sales Count] ON COLUMNS,
[Dim Product].[Product Line Id].[Product Line Id].Members ON ROWS
FROM
[DDSDA]
WHERE
[Dim Customer].[Member Type].&[Normal]
```

  - Kết quả:

	Fact Sales Count
Electronic accessories	79
Fashion accessories	89
Food and beverages	86
Health and beauty	68
Home and lifestyle	92
Sports and travel	95
Unknown	(null)

- Lượng rating của khách hàng Normal và Member:
  - MDX query:

```
SELECT
  {[Measures].[Fact Sales Count]} ON COLUMNS,
  NON EMPTY{
    [Dim Product].[Product Line Name].Members *
    {[Dim Customer].[Member Type].[Member], [Dim Customer].[Member Type].[Normal]}
  } ON ROWS
FROM
  [DDSDA]
WHERE
  {
    [Dim Product].[Product Line Id].Members
  }
```

  - Kết quả:

		Fact Sales Count
All	Member	507
All	Normal	509
Electronic accessories	Member	89
Electronic accessories	Normal	79
Fashion accessories	Member	90
Fashion accessories	Normal	89
Food and beverages	Member	86
Food and beverages	Normal	86
Health and beauty	Member	85
Health and beauty	Normal	68
Home and lifestyle	Member	78
Home and lifestyle	Normal	92
Sports and travel	Member	79
Sports and travel	Normal	95

5. Thống kê số lượng sản phẩm bán được theo từng thời điểm (time / date)

- Thống kê số lượng sản phẩm theo Date:
  - MDX query:

```
SELECT
  NON EMPTY { [Measures].[Quantity] } ON COLUMNS,
  NON EMPTY {
    (
      [Dim Date].[Year].[Year].ALLMEMBERS *
      [Dim Date].[Month].[Month].ALLMEMBERS *
      [Dim Date].[Day].[Day].ALLMEMBERS *
      [Dim Date].[Hour].[Hour].ALLMEMBERS
    )
  } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
FROM [DDSDA]
```

  - Kết quả:

				Quantity
2019	January	1	10:39:00	6
2019	January	1	11:36:00	10
2019	January	1	11:40:00	6
2019	January	1	11:43:00	2
2019	January	1	13:55:00	9
2019	January	1	14:42:00	10
2019	January	1	14:47:00	8
2019	January	1	15:51:00	2
2019	January	1	19:07:00	6
2019	January	1	19:31:00	8
2019	January	1	19:48:00	10

6. Thống kê số lượng khách hàng nữ đã mua theo từng loại sản phẩm

- MDX query:

```
SELECT
  [Measures].[Fact Sales Count] ON COLUMNS,
  NON EMPTY [Dim Product].[Product Line Id].[Product Line Id].Members ON
ROWS
FROM
  [DDSDA]
WHERE

    [Dim Customer].[Gender].[Female]
```

- Kết quả:

	Fact Sales Count
PD001	83
PD002	90
PD003	83
PD004	85
PD005	83
PD006	83

7. Cho biết doanh thu của từng chi nhánh của các năm

- MDX query

```
SELECT
  NON EMPTY { [Measures].[Total] } ON COLUMNS,
  NON EMPTY {
    (
      [Dim Branch].[Name].[Name].ALLMEMBERS *
      [Dim Date].[Year].[Year].ALLMEMBERS
    )
  } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
FROM [DDSDA]
```

- Kết quả

		Total
Mandalay	2019	104336.862
Mandalay	2020	3025.071
Naypyitaw	2019	109418.085
Naypyitaw	2020	1602.3315
Yangon	2019	107291.289
Yangon	2020	2397.2235

2.4 Mining

Đề xuất trường hợp: Dự đoán doanh thu khi ra mắt một dòng sản phẩm mới

- Sử dụng mô hình hồi quy tuyến tính, ngôn ngữ Python trên môi trường Jupiter Notebook
- Load, EDA dữ liệu từ DDS

EDA

```
import pandas as pd
from sklearn.linear_model import LinearRegression

server = 'PC\SQLSERVERA'
database = 'DDSDA'
trusted_connection = 'yes'

conn_str = (
    f'DRIVER={{SQL Server}};'
    f'SERVER={server};'
    f'DATABASE={database};'
    f'Trusted_Connection={trusted_connection};'
)

conn = pyodbc.connect(conn_str)
sql_query = """
SELECT
    p.ProductLineId,
    SUM(f.Quantity) AS total_quantity,
    AVG(p.UnitPrice) as average_unit_price,
    AVG(f.Rating) AS average_rating,
    SUM(f.GrossIncome) as Total_income,
    SUM(f.Total) AS total_sales
FROM Fact_Sales f
INNER JOIN Dim_Date d ON f.Date_Key = d.Date_SK
INNER JOIN Dim_Product p ON f.Product_Key = p.Product_SK
GROUP BY
    p.ProductLineId
ORDER BY p.ProductLineId
"""
df = pd.read_sql(sql_query, conn)
conn.close()
```

- Dữ liệu cần thiết bao gồm các dữ liệu thống kê liên quan danh số và rating từ bảng fact nhưng do muốn theo 1 dòng sản phẩm nên có kết với bản Dim_Product để group theo dòng sản phẩm và tính toán các giá trị cho phù hợp để chọn làm biến độc lập

Data sau khi EDA

```
df
```

[110] ✓ 0.0s Python

	ProductLineId	total_quantity	average_unit_price	average_rating	Total_income	total_sales
0	PD001	826	56.0149	6.901961	2183.3855	45851.0955
1	PD002	918	56.2830	6.968452	2557.7365	53712.4665
2	PD003	922	54.3010	7.110588	2464.0280	51744.5880
3	PD004	1005	55.2208	6.871839	2739.6805	57533.2905
4	PD005	959	54.9429	7.085465	2939.1710	61722.5910
5	PD006	962	54.5154	6.843575	2738.4205	57506.8305

- Dữ liệu có các biến độc lập bao gồm : total_quantity, average_unit_price, average_rating, Total_income
- Biến phụ thuộc : total_sale
- Xây dựng mô hình từ dữ liệu và kiểm tra mô hình có tin cậy hay không

Xây dựng và kiểm tra mô hình

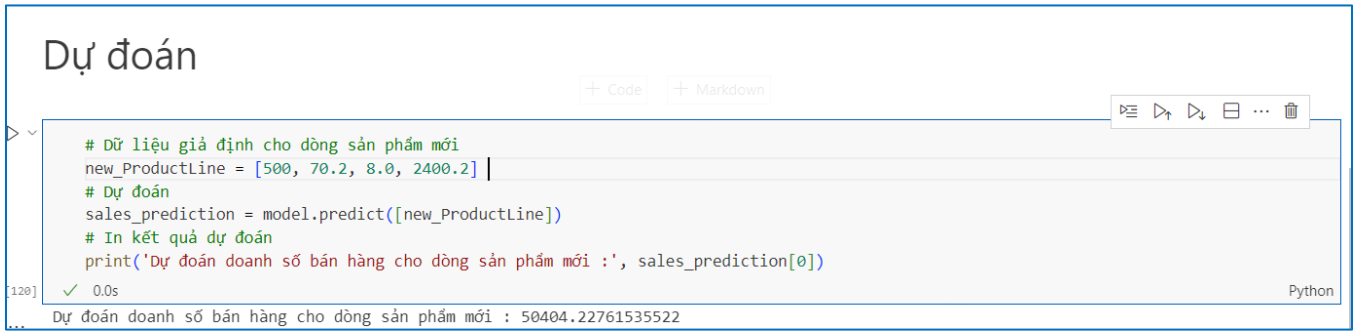
```
features = df[[ 'total_quantity', 'average_unit_price', 'average_rating','Total_income']]
target = df['total_sales']

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Khởi tạo mô hình và huấn luyện
model = LinearRegression()
model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred = model.predict(X_test)
# Đánh giá mô hình
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

[115] ✓ 0.0s Python

```
Mean Squared Error: 1.6038697984605902e-05
```

Với Mean Squared nhỏ tiệm cận bằng 0 có thể thấy mô hình có ý nghĩa thống kê và chính xác cao khi dùng để đánh giá , dự đoán

- Áp dụng mô hình đã xây dựng để dự đoán doanh thu cho 1 sản phẩm mới có dữ liệu mẫu là :
  - total_quantity : 500
  - average_unit_price: 70.2
  - average_rating: 8.0
  - Total_income : 2400.2



- Kết quả doanh thu dự đoán cho 1 sản phẩm ví dụ trên là 50404.2276