ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO: ĐỒ ÁN THỰC HÀNH

MÔN: HỆ THỐNG THÔNG TIN PHỤC VỤ TRÍ TUỆ KINH DOANH

GVLT: cô Hồ Thị Hoàng Vy - hthvy@fit.hcmus.edu.vn

GVTH: cô Tiết Gia Hồng - tghong@fit.hcmus.edu.vn

GVTH: thầy Lê Nguyễn Hoài Nam - lnhnam@fit.hcmus.edu.vn

HỌC KỲ I – NĂM HỌC 2023 - 2024

A. THÔNG TIN NHÓM

Mã nhóm	20HTTT2.BI.07	
MSSV	**Họ và tên**	**Tỉ lệ hoàn thành công việc**
20127064	Nguyễn Trần Mai Phương	33%
20127250	Hồ Thị Thanh Ngân	34%
20127573	Nguyễn Thị Hồng Ngọc	33%

MỤC LỤC

A. THÔNG TIN NHÓM **2**
B. SUPERMARKET_SALES DATASET **6**
I. Attribute Information 6
II. Data warehouse design 7
1. Design data flow 8
2. Design NDS 8
3. Design DDS 20
III. ETL 26
1. Project Param: 26
2. Set up METADATA: 30
3. Thông tin NDS: 32
- Product (Master) 33
- ProductLine (Master) 36
- City (Master) 39
- CustomerType (Master) 42
- DATE_MASTER 43
- TIME_MASTER 44
- Payment (Master) 44
- Supermarket_sales (Transaction) 46
4. Thông tin DDS 51
- Product (Dimension) 51
- ProductLine (Dimension) 54
- City (Dimension) 56
- CustomerType (Dimension) 59
- Payment (Dimension) 61
- DATETIME_MASTER (Dimension) 63
- supermarket_sales (Fact) 64
4. Danh sách các table thuộc về các database 67
I. Hình các package 69
II. Analysis Service 75
1. Cài đặt server: 75
2. Tạo mới Data Source: 75
3. Tạo mới Data Source Views 76
4. Tạo Cube 79
5. Phân chiều cho các bảng Dim 83

6. Tạo measure: 88
- Invoice ID SK_Count, Supermarket Sale Count 88
- Rating Sum 89
- Rating_Count 89
- Rating_aver 90
III. Report 90
IV. Olap 98
V. Data Mining 101
Câu 1 101
Câu 2 103
Câu 3 103
VI. Dashboard 105
C. MOVIE DATASET 106
I. Attribute Information 106
a. Source 1: film_sales.xlsx 106
b. Source 2: film_sales.csv 109
II. Data warehouse design 110
1. Design Data Flow 110
2. Design NDS 110
3. Design DDS 120
III. ETL 122
1. Set up Project Param 122
2. Set up MetaData 124
3. Thông tin tạo NDS 126
- film (Master) 127
- film_cate (Master) 130
- City (Master) 132
- cinema (Master) 135
- DATE_MASTER 137
- film_sales (Transaction) 138
4. Thông tin tạo DDS: 142
- film (Dimension) 142
- film_cate (Dimension) 146
- city (Dimension) 149
- cinema (Dimension) 152
- film_sales (Fact) 155
4. Danh sách các table thuộc về các database 157

IV. Hình các package 159
V. Analysis Service 165
1. Cài đặt server: 165
2. Tạo mới Data Source 165
3. Tạo mới Data Source Views 167
4. Tạo Cube 168
5. Cấp chiều Cube 172
6. Tạo measure 176
VI. Report 180
VII. Olap 186
VIII. Data Mining 190
Câu 1 190
Câu 2 192
IX. Dashboard 194
D. Information 195
I. Link 195
II. Reference: 195

B. SUPERMARKET_SALES DATASET

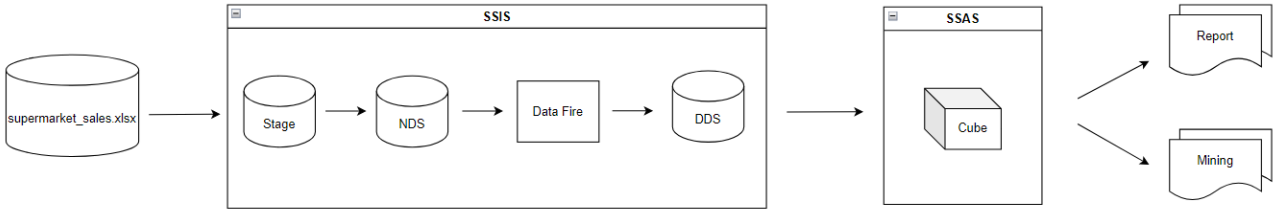
I. Attribute Information

Bảng	Thuộc tính	Ý nghĩa	Phụ thuộc	Định dạng
supermarket_sales	Invoice ID	Mã hoá đơn mua hàng		char
	Branch	Mã thành phố dùng để phân biệt các chi nhánh	supermarket_Sales.Branch tham chiếu đến thuộc tính Branch trong bảng city	char
	Customer type	Phân loại người mua hàng		char
	Gender	Xác định giới tính người mua hàng	Hiện gồm 4 định dạng (Female, Male, F,M)	char
	ProductID	Mã sản phẩm	supermarket_Sales.ProductID tham chiếu đến thuộc tính ProductID trong bảng Product	char
	Quantity	Số lượng của sản phẩm ProductID mà khách hàng mua	Quantity > 0	int
	Tax 5%	Tiền thuế của sản phẩm mà khách hàng mua	Tax 5% = Cogs * 5%	float
	Total	Tổng tiền sản phẩm mà khách hàng phải trả	Total = Cogs - Tax 5%	float
	Date	Ngày xuất hoá đơn	01/01/2019 <= Date <= 31/03/2019	date
	Time	Thời gian xuất hoá đơn	tính theo 24h 1 ngày 00:00 <= Time <= 23:59	time

	Payment	Phương thức thanh toán của khách hàng		char
	Cogs	Tổng giá tiền của sản phẩm (chưa gồm thuế)	Cogs = Quantity*(Product.Unit price)	float
	Gross margin percentage	% lợi nhuận mà công ty nhận được từ 1 sản phẩm	Gross margin percentage = 4,761904762	float
	Gross income	Tổng số lợi nhuận mà công ty nhận được từ số tiền mà khách hàng trả	Gross income = Total * gross margin percentage	float
	Rating	Đánh giá của khách hàng	0,0 <= rating <= 10.0	float
product	ProductID	Mã sản phẩm		char
	Unit price	Đơn giá sản phẩm	Unit price >= 0,0	float
	ProductLine	Mã loại sản phẩm	Product.Product tham chiếu đến thuộc tính ProductLineID trong bảng ProductLine	char
ProductLine	Product line	Mã loại sản phẩm		char
	ProductLineID	Tên loại sản phẩm		char
city	Branch	Mã thành phố/chi nhánh		char
	City	Tên thành phố/chi nhánh		char

II. Data warehouse design

1. Design data flow



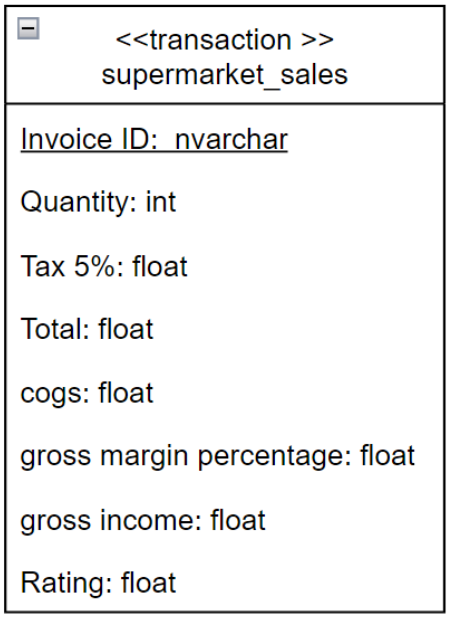
Dataflow Supermarket_sales

2. Design NDS

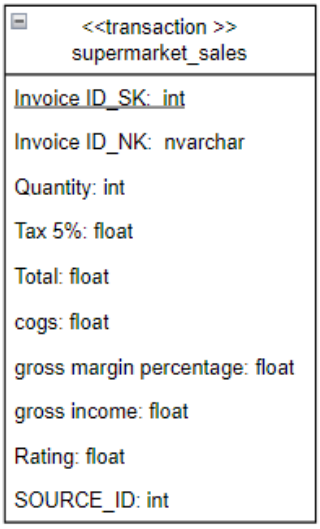
***Stage* supermarket_sales:**

	Invoice ID	Branch	Customer type	Gender	ProductID	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating	SOURCE_ID
1	750-67-8428	A	Member	Female	PID01	7	26.1415	548.9715	2019-01-05	13:08:00.0000000	Ewallet	522.83	4.761904762	26.1415	9.1	1
2	123-19-1176	A	Member	Male	PID02	8	23.288	489.048	2019-01-27	20:33:00.0000000	Ewallet	465.76	4.761904762	23.288	8.4	1
3	355-53-5943	A	Member	Female	PID03	6	20.652	433.692	2019-02-25	14:36:00.0000000	Ewallet	413.04	4.761904762	20.652	5.8	1
4	665-32-9167	A	Member	Female	PID04	2	3.626	76.146	2019-01-10	17:15:00.0000000	Credit card	72.52	4.761904762	3.626	7.2	1
5	656-95-9349	A	Member	Female	PID05	7	24.1255	506.6355	2019-03-11	11:03:00.0000000	Credit card	482.51	4.761904762	24.1255	4.6	1
6	549-59-1358	A	Member	Male	PID06	3	13.2945	279.1845	2019-03-02	17:36:00.0000000	Ewallet	265.89	4.761904762	13.2945	6	1
7	227-03-5010	A	Member	Female	PID07	8	21.036	441.756	2019-03-22	19:20:00.0000000	Credit card	420.72	4.761904762	21.036	8.5	1
8	129-29-8530	A	Member	Male	PID08	5	15.655	328.755	2019-03-10	19:15:00.0000000	Ewallet	313.1	4.761904762	15.655	7	1
9	162-48-8011	A	Member	Female	PID09	5	11.1475	234.0975	2019-02-10	15:10:00.0000000	Cash	222.95	4.761904762	11.1475	8.5	1
10	106-35-6779	A	Member	Male	PID10	2	4.434	93.114	2019-03-27	11:26:00.0000000	Cash	88.68	4.761904762	4.434	5.8	1

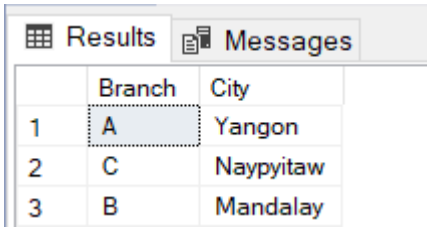
Vì supermarket_sales là bảng lưu giá trị doanh thu, ta sẽ tiến hành tạo 1 bảng **supermarket_sales** trong ***NDS*** với loại là Transaction với các thuộc tính Measure:



Từ source ta có thể thấy được InvoiceID là khoá chính của source => Tạo khoá thay thế Invoice ID_SK để tạo khoá tự tăng dần trong ***NDS,*** đổi tên cột Invoice ID thành Invoice ID_NK để lưu giá trị khoá chính cũ, thêm thuộc tính SOURCE_ID để lưu thông tin source:



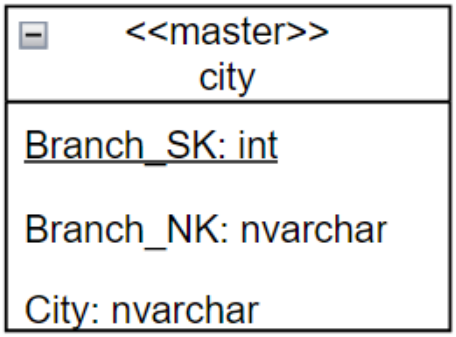
Trong thuộc tính Branch ở **supermarket_sales *Stage***, nó lưu trữ giá trị thông tin về chi nhánh xuất hoá đơn, trước hết nhìn vào bảng city trong Stage:



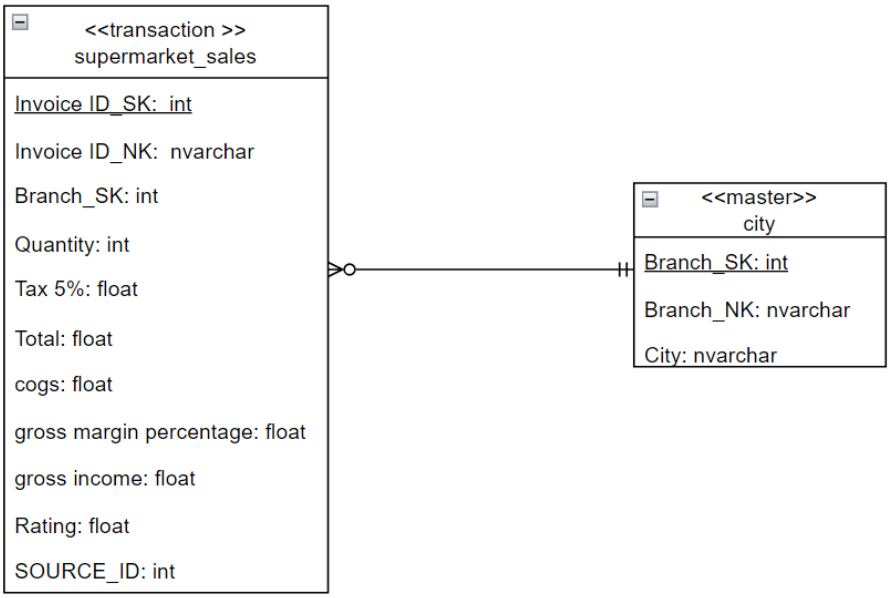
Results | Messages

	Branch	City
1	A	Yangon
2	C	Naypyitaw
3	B	Mandalay

Bảng city *Stage* lưu thông tin về chi nhánh và thành phố của chi nhánh đó, vì mỗi 1 chi nhánh chỉ nằm năm nhất trong 1 thành phố, thế nên Branch có thể được xem là khóa chính của bảng city, 1 Branch sẽ tìm ra một tên City duy nhất. Ở đây, ta có thể tách tên một bảng City để lưu thông tin về thành phố tuy nhiên vì yêu cầu đồ án thì việc tạo thêm bảng là không cần thiết và tốn chi phí, thế nên tôi chỉ tạo duy nhất một bảng city và NDS lưu giá trị về thông tin Branch: với Branch_SK là khoá thay thế tự tăng, Branch đổi thành Branch_NK để lưu thông tên Branch và City để lưu thông tin về tên thành phố mà Branch được mở tại đó.



Tiếp đó ta sẽ thêm cột Branch_SK trong **supermarket_sales *NDS*** và tham chiếu đến cột Branch_SK của bảng **city *NDS*** vừa tạo:



Customer type, Payment

Xét thuộc tính miền giá trị của thuộc tính Payment và Customer type trong **supermarket_sales *Stage***:

```
select distinct [Customer type] from supermarket_sales
```

138 %

Results | Messages

	Customer type
1	Normal
2	Member

```
select distinct [Payment] from supermarket_sales
```

138 %

Results | Messages

	Payment
1	Ewallet
2	Cash
3	Credit card

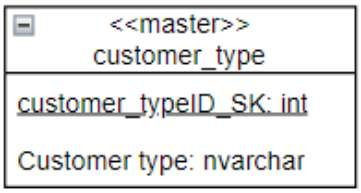
Ta nhận thấy chỉ có 2 loại giá trị trong Customer type và 3 loại giá trị Payment, và đồng thời, theo như yêu cầu report:

Report:

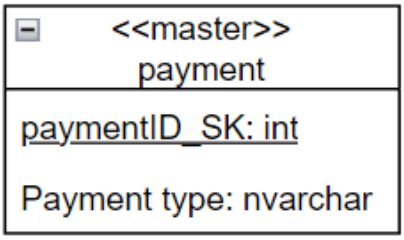
1. Thống kê số thành viên mua hàng theo ngày, tháng, năm
2. Thống kê doanh thu của khách hàng (member, normal) theo ngày, tháng, năm và theo chi nhánh.
3. Thống kê số lượng khách thanh toán theo cash/ debit/... ở từng chi nhánh theo từng tháng trong năm
4. Thống kê lượng rating của khách hàng (member, normal) theo từng loại sản phẩm (ProductLine)
5. Thống kê số lượng sản phẩm bán được theo từng thời điểm (time / date)
6. Thống kê số lượng khách hàng nữ đã mua theo từng loại sản phẩm

Thì việc thống kê sẽ chủ yếu xoay quanh loại khách hàng và loại hình thanh toán, thế nên nhóm quyết định sẽ tạo thêm 2 bảng vào ***NDS***:

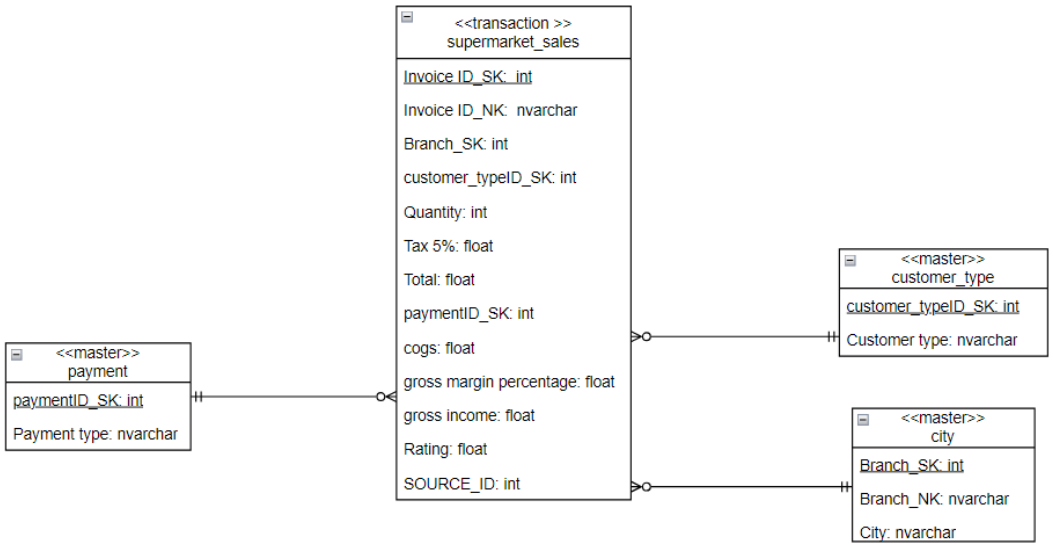
+ customer_type: với customer_typeID_SK làm khoá chính tự tăng, [Customer Type] lưu giá trị thông tin về các loại khách hàng.



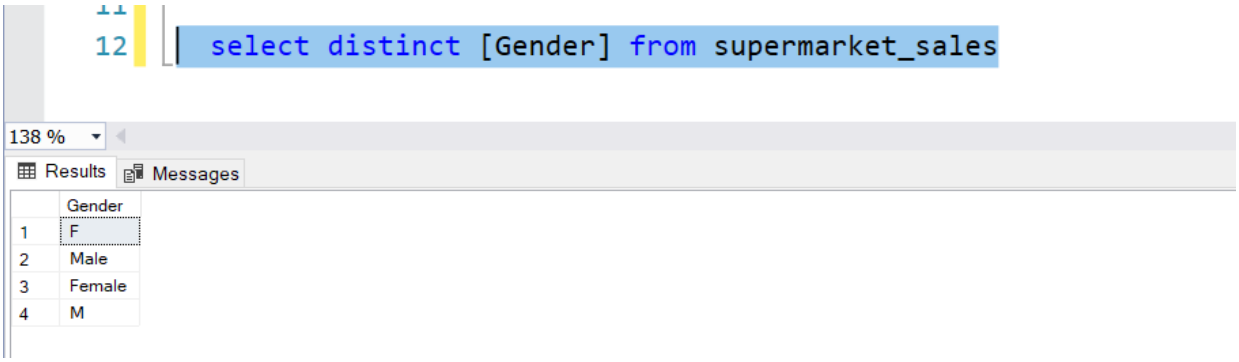
+ payment: paymentID_SK làm khoá chính tự tăng, [Payment type] lưu giá trị thông tin về các loại hình thanh toán:



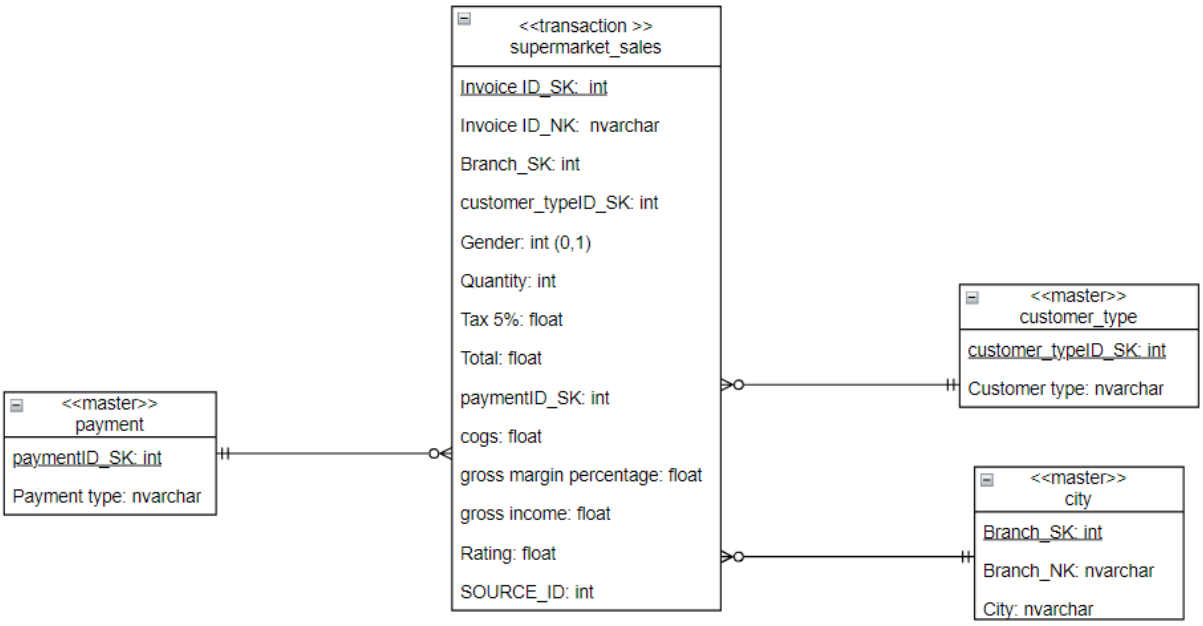
Sau đó, ta sẽ thêm thuộc tính customer_typeID_SK & paymentID_SK vào bảng **supermarket_sales** ***NDS***, và tham chiếu chúng đến 2 bảng vừa tạo:



Về thuộc tính Gender trong **supermarket_sales** ***Stage***, vì trong thực tế Gender chỉ có 2 loại là Female và Male, tuy nhiên ở trong Source lại có nhiều giá trị hơn nhưng lại chỉ cùng ý nghĩa:



Thế nên ta sẽ chuyển đổi và thêm cột Gender vào **supermarket_sales** trong ***NDS*** Type với int (0,1): Male hoặc M là 0 và Female hoặc F là 1.

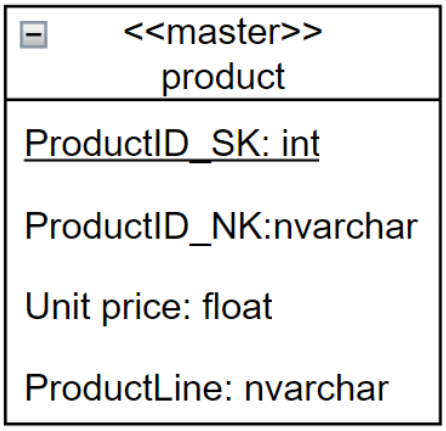


Với thuộc tính ProductID trong **supermarket_sales** ở ***Stage***, nó lưu khoá của Product, trước hãy nhìn vào bảng **product** trong ***Stage***:

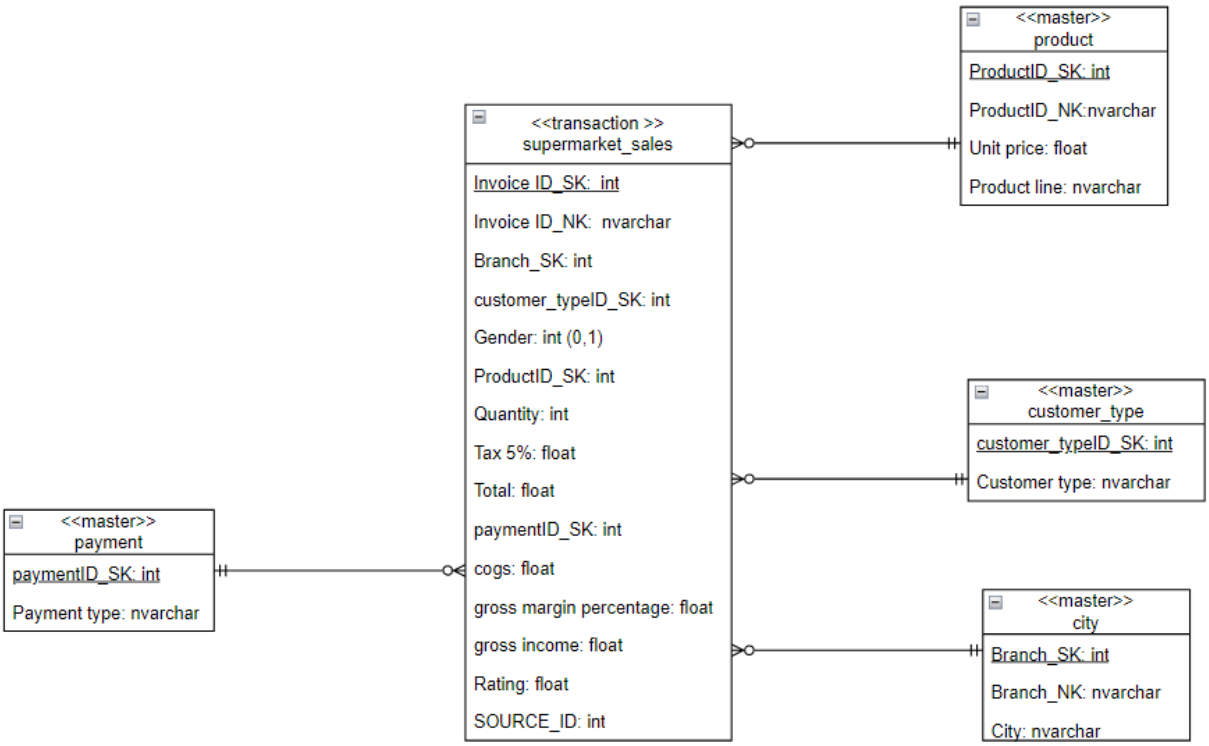
Stage product:

	ProductID	Unit price	ProductLine
1	PID01	74.69	PD001
2	PID02	58.22	PD002
3	PID03	68.84	PD003
4	PID04	36.26	PD001
5	PID05	68.93	PD004
6	PID06	88.63	PD002
7	PID07	52.59	PD002
8	PID08	62.62	PD003
9	PID09	44.59	PD001
10	PID10	44.34	PD005

Từ bảng ***Stage* product**, ta có thể thấy ***Stage* product** có khóa chính là ProductID, từ ProductID ta có thể biết được Unit Price và ProductLine của product đó. Như vậy trước hết ta sẽ thêm một bảng **Product dạng master** trong ***NDS*** với khoá ProductID_SK là khóa thay thế tự tăng, ProductID được đổi tên thành ProductID_NK để lưu giá trị khoá cũ từ ***stage***:



Sau đó ta sẽ thêm thuộc tính ProductID vào **supermarket_sales *NDS*** tham chiếu đến với ProductID_SK của **bảng product *NDS*** mà ta vừa tạo:

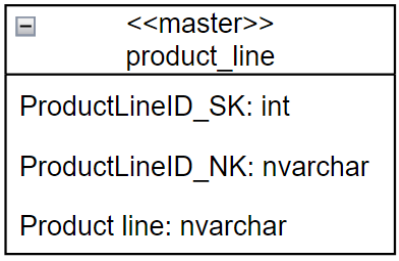


Vì trong **product *stage*** có chứa thuộc tính ProductLineID được lấy từ bảng **product_line**. Nhìn vào bảng **product_line** trong ***Stage***:

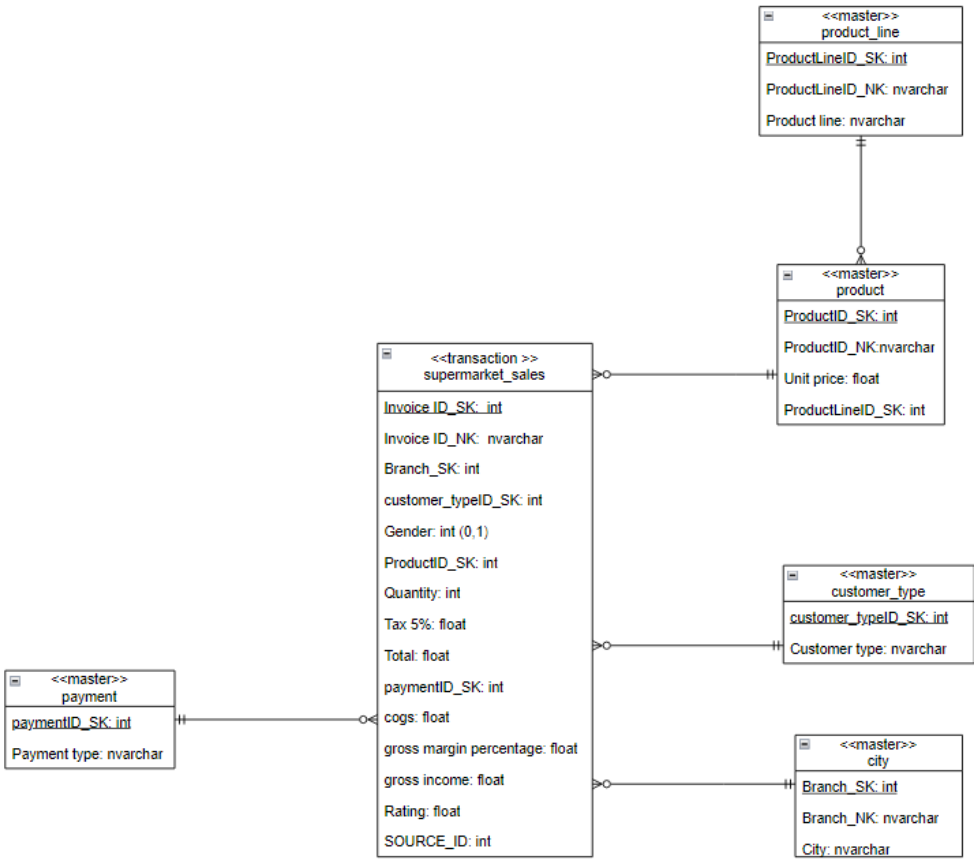
Stage product_line:

	Product line	ProductLineID
1	Health and beauty	PD001
2	Electronic accessories	PD002
3	Home and lifestyle	PD003
4	Sports and travel	PD004
5	Food and beverages	PD005
6	Fashion accessories	PD006

Ta có thể thấy 1 khoá chính là ProductLineID, từ 1 ProductLineID ta chỉ tìm được một Product Line duy nhất, như vậy ta sẽ tiến hành tạo **bảng product_line** trong ***NDS*** để lưu thông tin product line, với ProductLineID_SK là khoá thay thế tăng dần, ProductLineID thì được đổi thành ProductLineID_NK để lưu giá trị cũ:

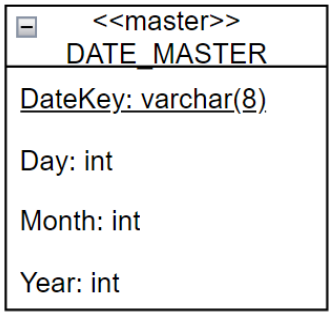


Tiếp đó ta sẽ đổi thay thế cột ProductLine trong **product *NDS*** thành ProductLineID_SK để tham chiếu đến cột ProductID_NK **bảng product_line** vừa tạo:



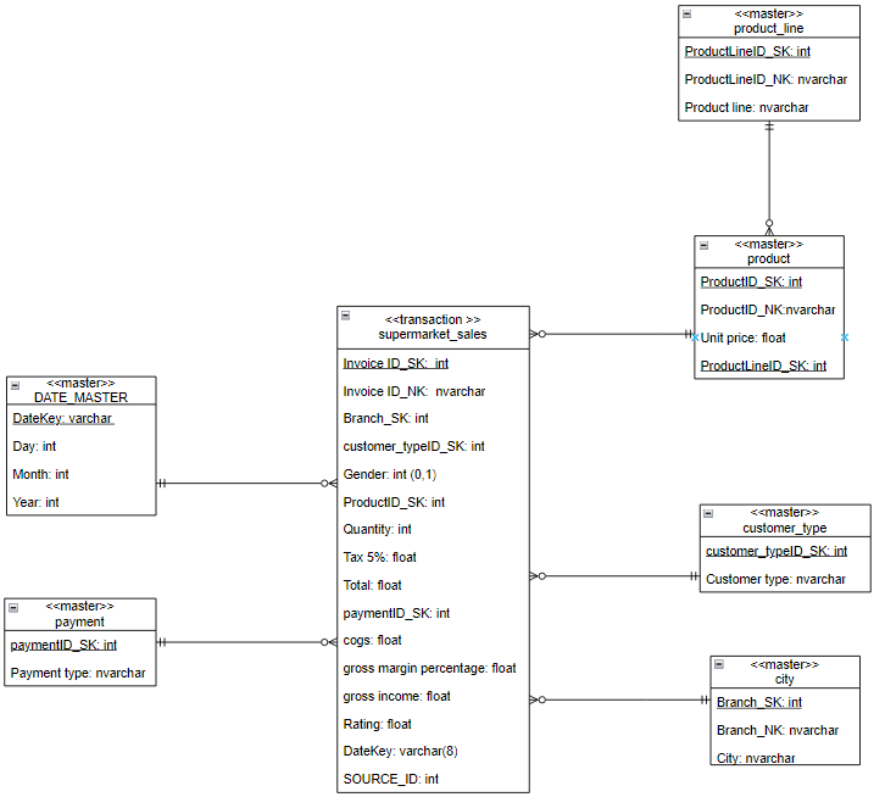
Tiếp đến việc xử lý ngày tháng năm, vì đây là thiết kế để phục vụ cho data warehouse, thế nên ta cần tạo một **DATE_MASTER** trong ***NDS*** để lưu giá trị ngày tháng năm và thêm một cột DATEKEY có dạng: 'YYYYMMDD' để làm khóa chính.

Vì bảng **DATE_MASTER** là bảng về datetime thế nên ta sẽ tự tạo trước dữ liệu trong DATE_MASTER thông qua script theo yêu cầu BI.



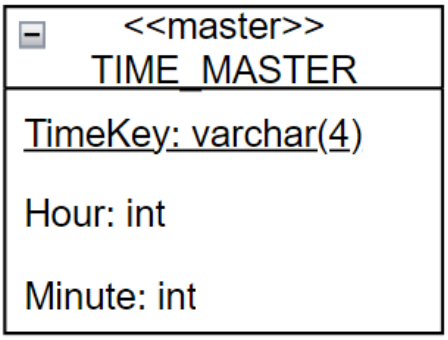
	DateKey	Day	Month	Year
1	20190101	1	1	2019
2	20190102	2	1	2019
3	20190103	3	1	2019
4	20190104	4	1	2019
5	20190105	5	1	2019
6	20190106	6	1	2019
7	20190107	7	1	2019
8	20190108	8	1	2019
9	20190109	9	1	2019
10	20190110	10	1	2019

Ta sẽ thêm thuộc tính DATEKEY vào bảng **supermarket_sales** ***NDS*** và tham chiếu đến DATEKEY trong **DATE_MASTER**:



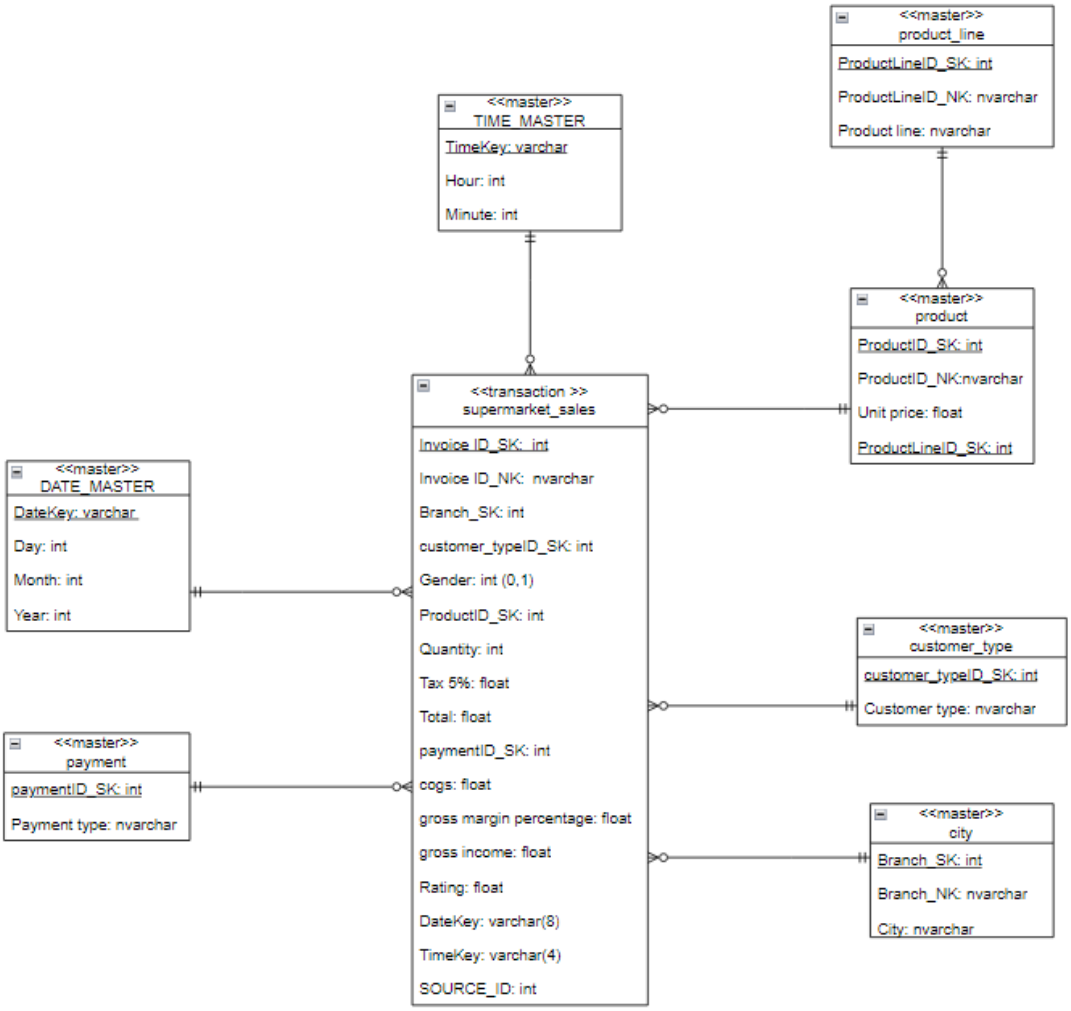
Tạo một **TIME_MASTER** trong ***NDS*** để lưu giá trị giờ, phút và thêm một cột TIMEKEY có dạng: 'HHMM' để làm khóa chính.

Vì bảng **TIME_MASTER** là bảng về datetime thế nên ta sẽ tự tạo trước dữ liệu trong **TIME_MASTER** thông qua script theo yêu cầu BI.

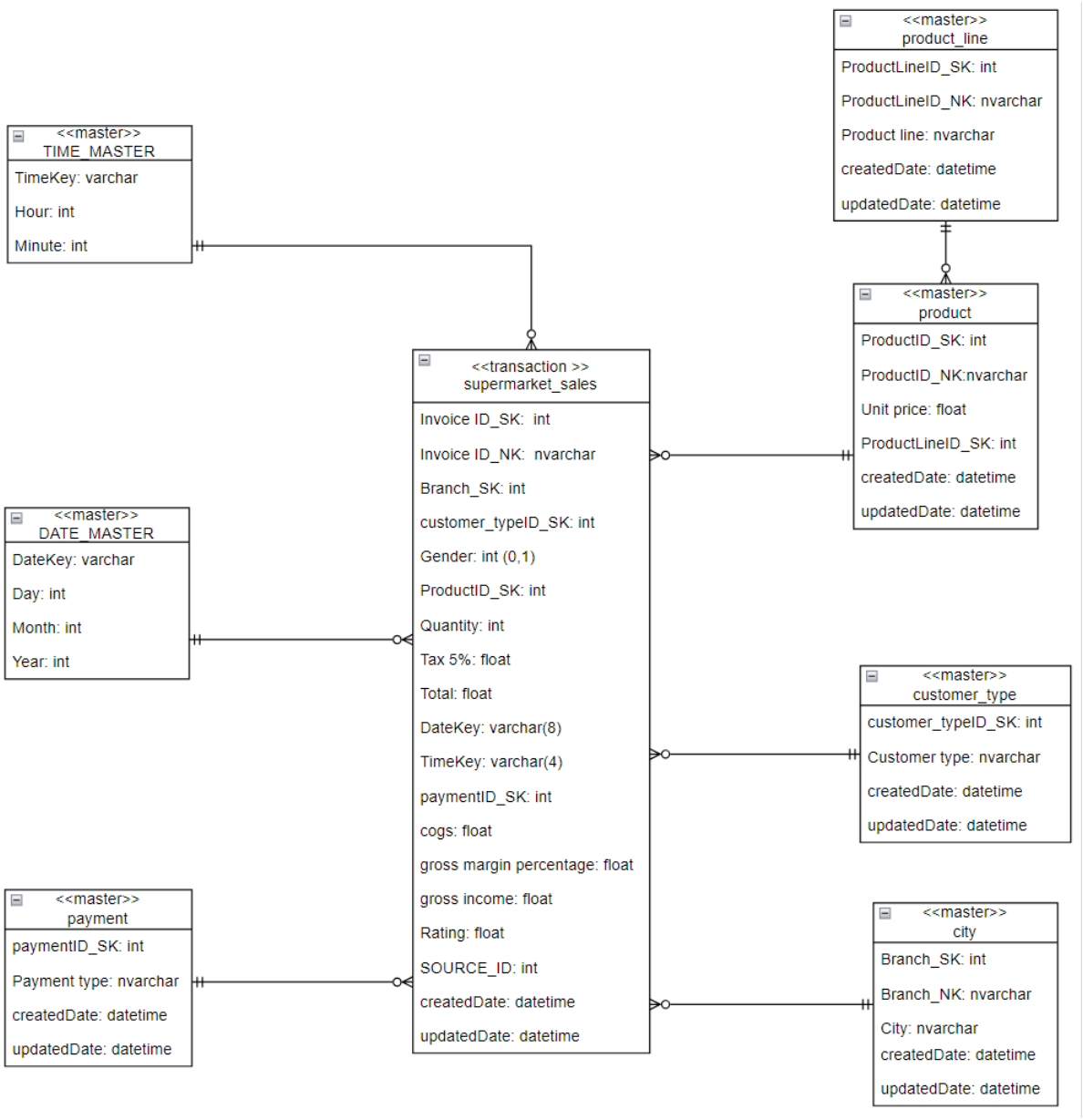


	TimeKey	Hour	Minute
1	0000	0	0
2	0001	0	1
3	0002	0	2
4	0003	0	3
5	0004	0	4
6	0005	0	5
7	0006	0	6
8	0007	0	7
9	0008	0	8
10	0009	0	9

Ta sẽ thêm thuộc tính TIMEKEY vào bảng **supermarket_sales *NDS*** và tham chiếu đến TIMEKEY trong **TIME_MASTER**:



Ở ***NDS*** ta sẽ thực hiện Incremental Extract, tức là cần so sánh với LSET, CET, ta sẽ thêm ở mỗi bảng Transaction & Master (Trừ bảng **DATE_MASTER** và **TIME_MASTER**) thuộc tính createdDate, updatedDate để tiện theo dõi quá trình ETL.

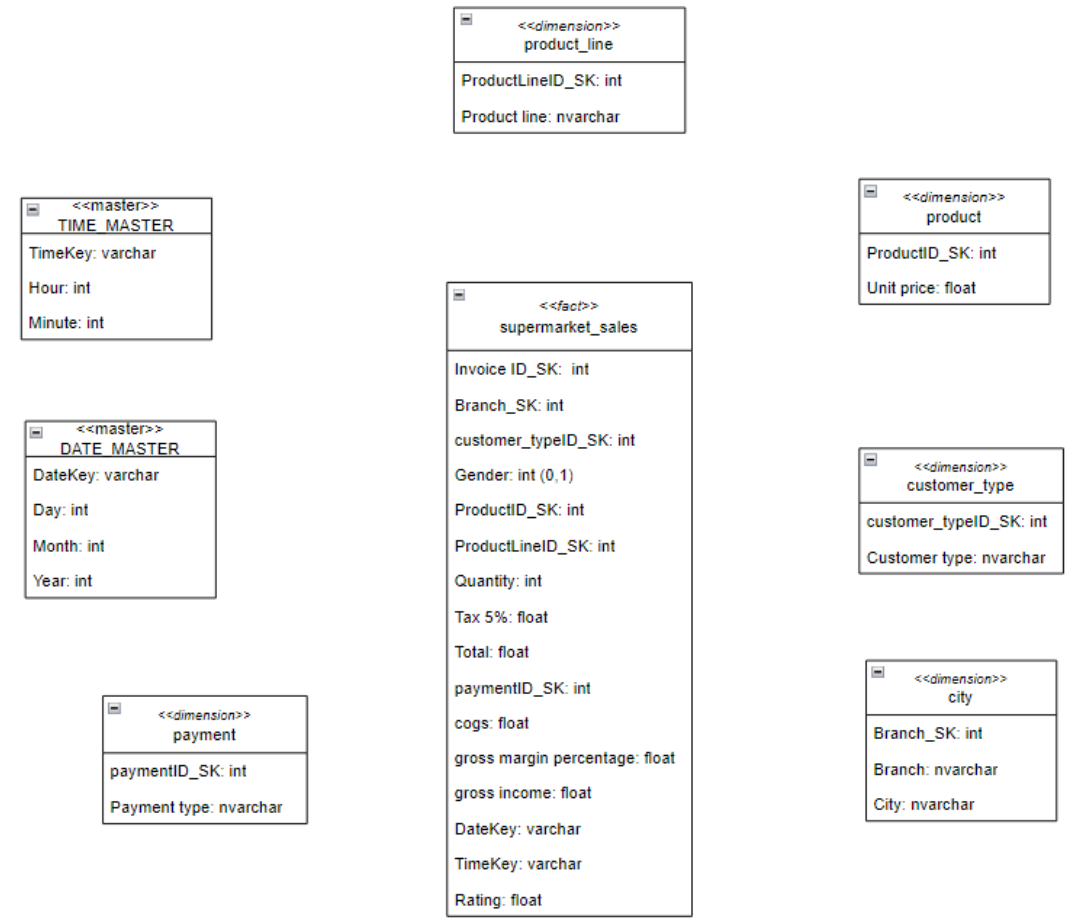


NDS

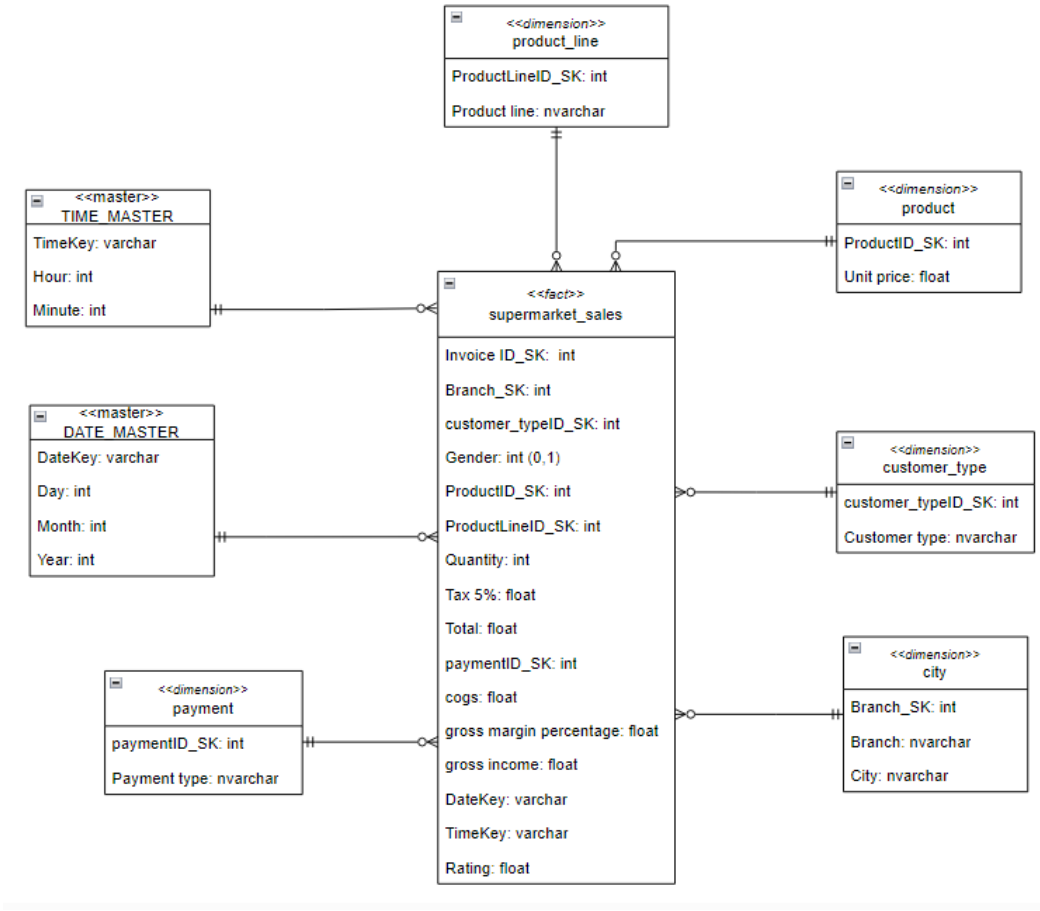
3. Design DDS

Ở ***DDS***, nhóm lựa chọn thiết kế Star Schema.

Trước hết, ta sẽ lấy lại thiết kế của ***NDS***, chuyển các bảng Master thành Dimension và bảng Transaction thành bảng fact, sau đó sẽ loại bỏ tất cả các khoá NK (trừ bảng city chỉ đổi tên Branch_NK thành Branch để lưu trữ tên chi nhánh) & createdDate, updatedDate ở mỗi bảng và xóa thuộc tính SOURCE_ID trong bảng Fact:



Tiếp đó ta sẽ đem tất cả những thuộc tính có khóa ngoại trong các bảng Dimension bỏ vào bảng fact, nối tham chiếu, và loại bỏ các thuộc tính khóa ngoại đó trong các bảng Dimension.

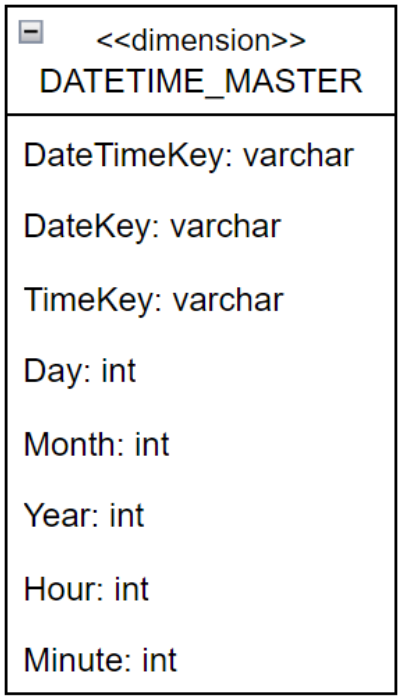


Sau cùng ta sẽ kiểm tra yêu cầu về phân cấp chiều, vì yêu cầu project là:

Thiết kế kho dữ liệu (KDL), chuẩn hóa, tổng hợp, nạp dữ liệu các nguồn vào KDL, thiết kế và xây dựng cube.

- Chuyển đổi dữ liệu ngày tháng sao cho có thể tạo được Date dimension với phân cấp chiều: năm > tháng > ngày > giờ
- Xác định và thiết kế các phân cấp chiều khác để đáp ứng yêu cầu OLAP và report.

Trước về chiều Date dimension, để có thể phân cấp chiều: năm > tháng > ngày > giờ, ta cần tạo 1 bảng **DATETIME_MASTER** tổng hợp cả **DATE_MASTER & TIME_MASTER** với kiểu như sau:

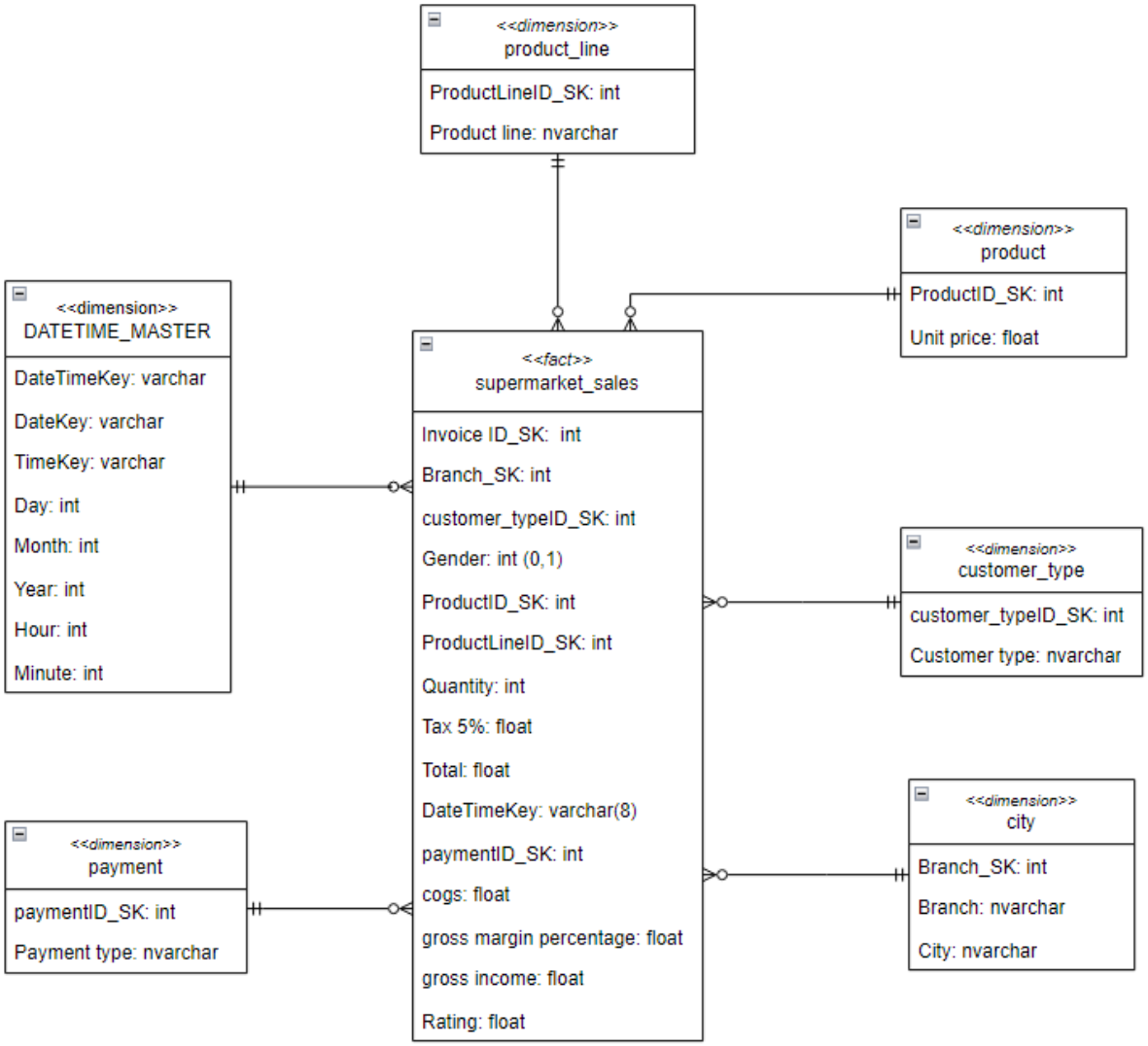


Với DateTimeKey có kiểu là 'YYYYMMDDHHSS' làm khoá chính, vì đây là chiều Datetime thế nên ta sẽ tiến hành tạo bảng vào bỏ giá trị vào bảng thông qua script đính kèm để được dữ liệu như sau:

Results | Messages

	DateTimeKey	DateKey	Day	Month	Year	TimeKey	Hour	Minute
1	201901010000	20190101	1	1	2019	0000	0	0
2	201901010001	20190101	1	1	2019	0001	0	1
3	201901010002	20190101	1	1	2019	0002	0	2
4	201901010003	20190101	1	1	2019	0003	0	3
5	201901010004	20190101	1	1	2019	0004	0	4
6	201901010005	20190101	1	1	2019	0005	0	5
7	201901010006	20190101	1	1	2019	0006	0	6
8	201901010007	20190101	1	1	2019	0007	0	7
9	201901010008	20190101	1	1	2019	0008	0	8
10	201901010009	20190101	1	1	2019	0009	0	9
11	201901010010	20190101	1	1	2019	0010	0	10
12	201901010011	20190101	1	1	2019	0011	0	11
13	201901010012	20190101	1	1	2019	0012	0	12
14	201901010013	20190101	1	1	2019	0013	0	13
15	201901010014	20190101	1	1	2019	0014	0	14
16	201901010015	20190101	1	1	2019	0015	0	15
17	201901010016	20190101	1	1	2019	0016	0	16

Sau đó ta sẽ bỏ thuộc tính DateKey & TimeKey trong bảng fact mà thay thế bằng DateTimeKey và tham chiếu đến cột DateTimeKey của bảng **DATETIME_MASTER**. Ta sẽ tiến hành loại bỏ 2 bảng **DATE_MASTER & TIME_MASTER**.



Tiếp đến ta sẽ kiểm tra độ mịn dữ liệu từ các yêu cầu report, Olap và Mining:

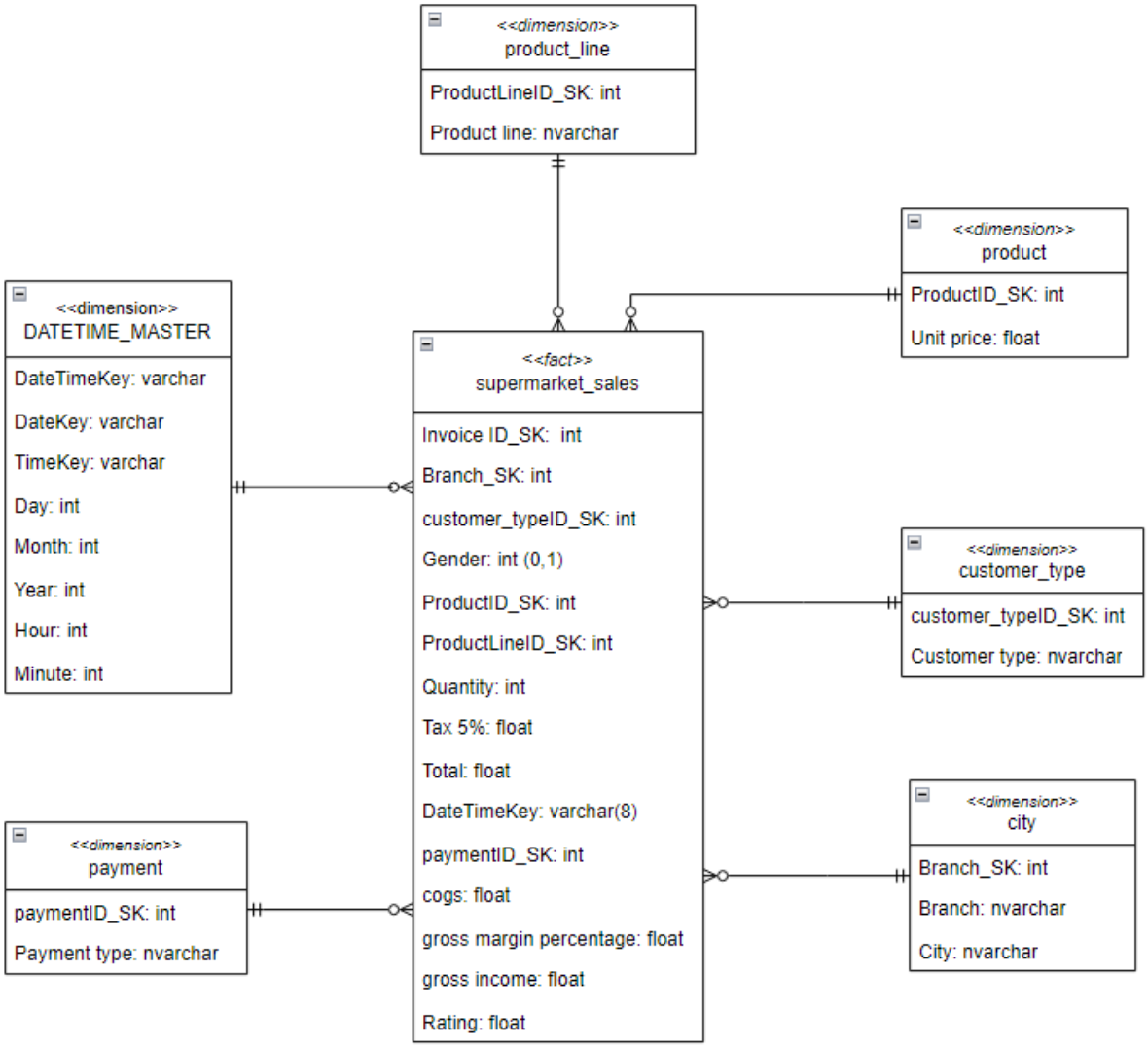
+ Report: vì ta đã check phần này ở ***DDS*** nên không cần thực hiện, nếu cần thêm bất kỳ Measure nào, ta có thể tự tạo ở SSAS.
+ Olap:

- **<u>OLAP</u>**:
  Tình hình mua hàng của khách hàng theo từng chi nhánh, từng loại sản phẩm, theo thời gian, hình thức thanh toán

Vì hiện tại ta đã có đủ chiều theo chi nhánh (city Dim), loại sản phẩm (product_line Dim), thời gian (DATETIME_MASTER Dim), hình thức thanh toán (payment Dim), nên ta không cần thay đổi mô hình.

+ Mining: Vì yêu cầu đề là tự đề xuất 1 trường hợp Mining, thế nên ở đây ta cũng không cần kiểm tra.

Và đây là mô hình DDS mà nhóm đã thiết kế:

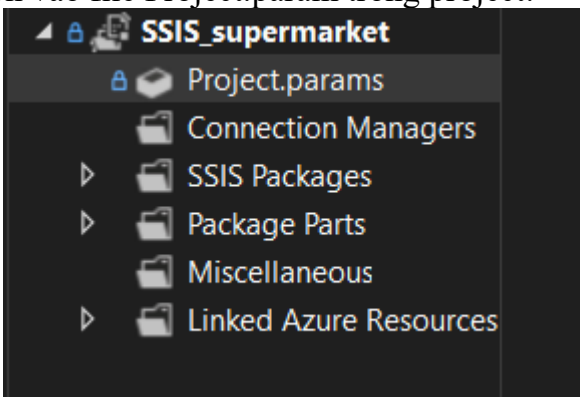


DDS

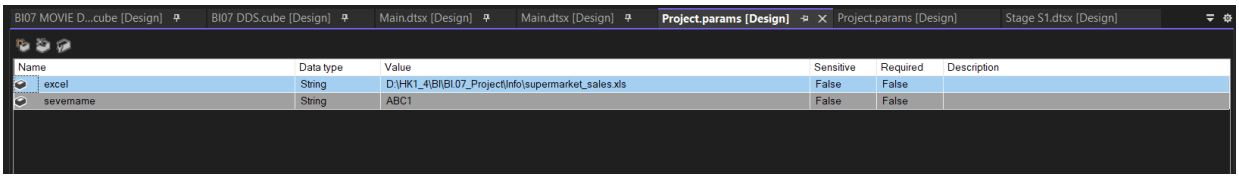
III. ETL

1. Project Param:

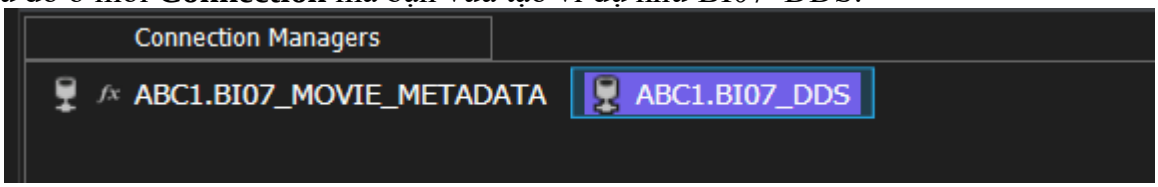
Để thuận tiện cho việc chạy ETL nhiều máy, cài đặt Project.param để lưu giá trị connection và tên server vào các biến trên toàn project.
Trước hết hãy click chọn vào file Project.param trong project:



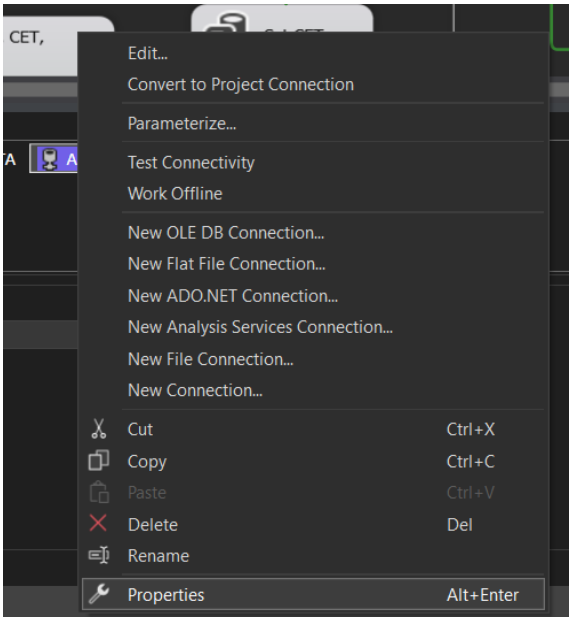
Sau đó ta sẽ tiến hành thêm các biến như sau:



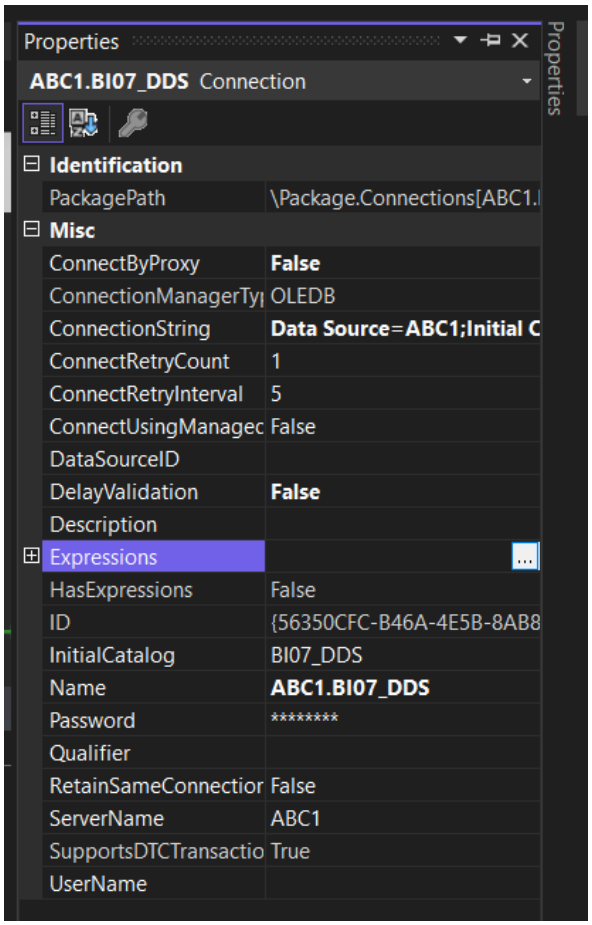
Với **excel** Value là path của source, **severname** là tên sever thiết bị của bạn.
Sau đó ở mỗi **Connection** mà bạn vừa tạo ví dụ như BI07 DDS:



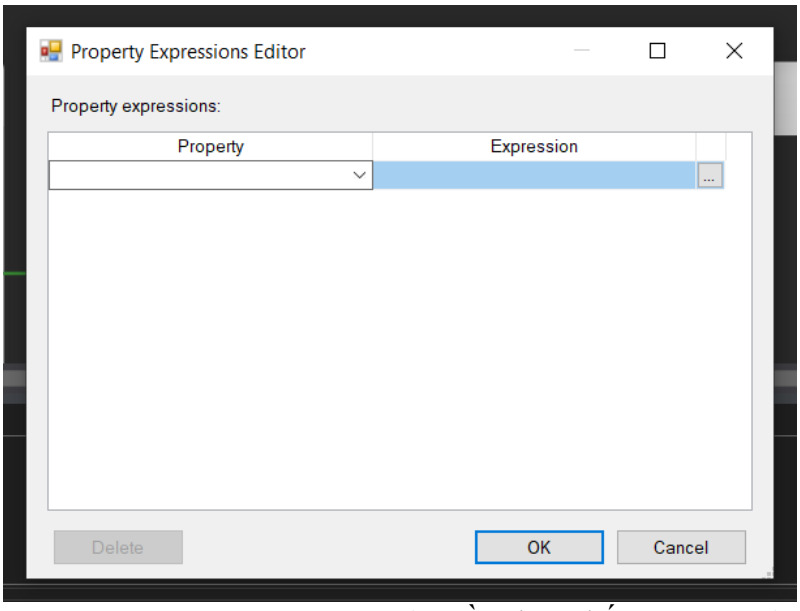
Ta click chuột phải vào **Connection**, chọn Properties:



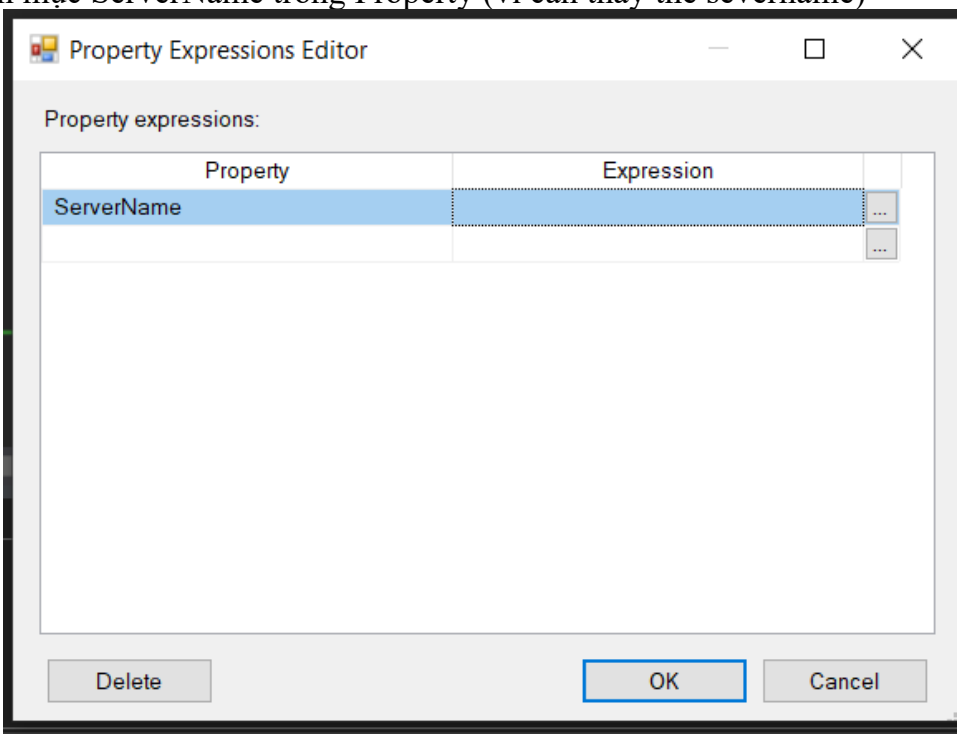
Chọn Expressions



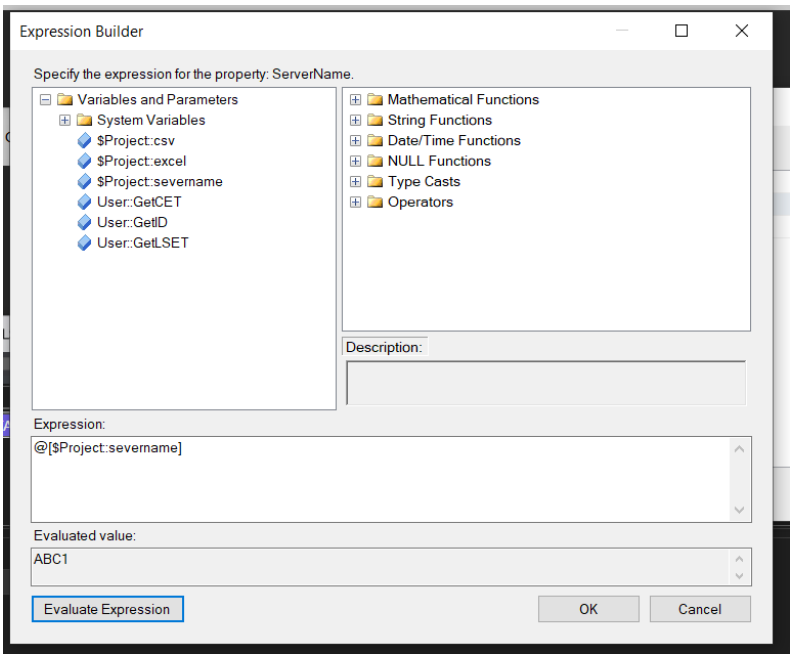
Màn hình sẽ nhảy ra một box như hình bên dưới:



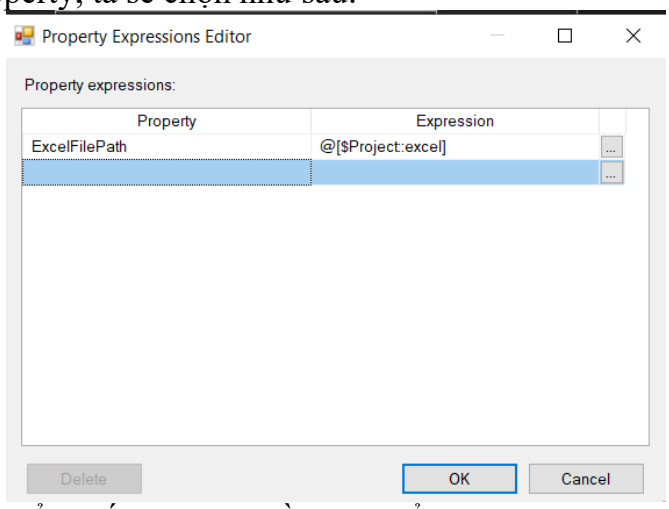
Ta sẽ chọn mục ServerName trong Property (vì cần thay thế severname)



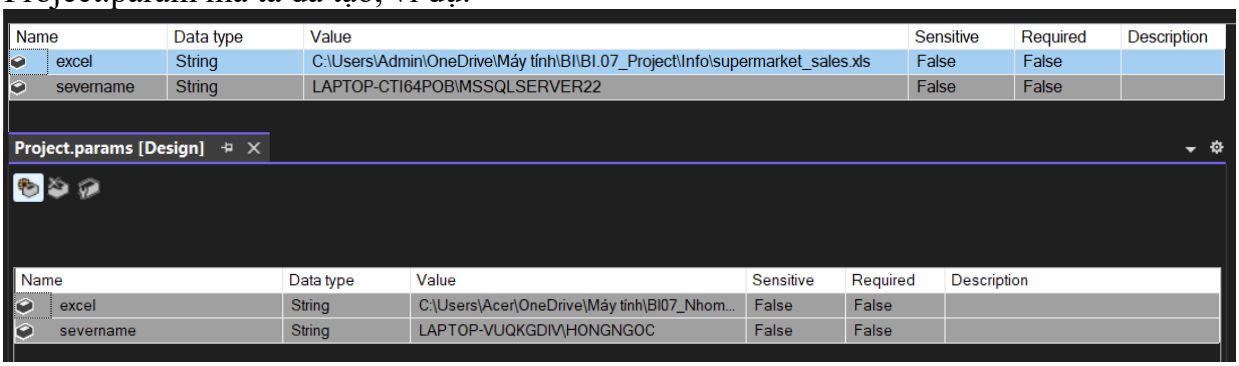
Sau đó ở expression ta sẽ ấn chọn biến severname vừa tạo như hình, lưu ý rằng cần nhấn Evaluated Value để có thể run thành công.



Tiếp đến connection String của excel, nó tương tự với những bước làm ở severname tuy nhiên trong mục Property, ta sẽ chọn như sau:



Và như vậy, mỗi khi đổi thiết bị, ta chỉ cần thay đổi giá trị Value trong các biến của Project.param mà ta đã tạo, ví dụ:



2. Set up METADATA:

DATA_FLOW:

	ID	SOURCE_NAME	LSET	CET
1	1	supermarket_sales	2023-12-29 08:10:25.627	2023-12-29 08:10:25.627

Bảng DATA_FLOW dùng để lưu thông tin Source, SourceID và thời gian LSET CET mà hệ thống ETL đối với Source đó, mỗi lần ETL xong thì LSET và CET sẽ được cập nhật lại giá trị mới

DATA_STORE_INFO

	DBS_ID	DBS_NAME	DESCRIPTIONS
1	1	BI07_METADATA	Describes whole data information in system
2	2	BI07_STAGE	An internal data store used for transforming and...
3	3	BI07_NDS	An internal master data store in the form of nor...
4	4	BI07_DDS	A user-facing data store, in the form of relational...

Bảng DATA_STORE_INFO lưu ID và thông tin các database được sử dụng trong hệ thống

DATA_TABLE_INFO

	TABLE_ID	DBS_ID	TABLE_NAME	TABLE_TYPE	DESCRIPTIONS	UPDATEDATE
1	1	1	DATA_FLOW	METADATA	Store the timestamps of data source flow that ET...	2023-12-29 08:10:26.000
2	2	2	city	Stage	Store City data about the city from source	2023-12-29 08:10:26.000
3	3	2	product_line	Stage	Store Product Line data from source	2023-12-29 08:10:26.000
4	4	2	product	Stage	Store Product data from source	2023-12-29 08:10:26.000
5	5	2	supermarket_sales	Stage	Store the Invoice data of supermarket_sales from...	2023-12-29 08:10:26.000
6	6	3	DATE_MASTER	Master tables(NDS)	Store date data	2023-12-11 16:52:47.103
7	7	3	TIME_MASTER	Master tables(NDS)	Store time data	2023-12-11 16:52:47.103
8	8	3	payment	Master tables(NDS)	Store payment type	2023-12-16 18:50:14.000
9	9	3	city	Master tables(NDS)	Store city data	2023-12-29 08:10:26.000
10	10	3	customer_type	Master tables(NDS)	Store customer type	2023-12-16 18:50:14.000
11	11	3	product_line	Master tables(NDS)	Store product line data	2023-12-29 08:10:26.000
12	12	3	product	Master tables(NDS)	Store product info data	2023-12-29 08:10:26.000
13	13	3	supermarket_sales	Transaction table(NDS)	Store the invoice data of supermarket_sales	2023-12-29 08:10:26.000
14	14	4	TIME	dimension	Store Time Dimension	2023-12-13 11:15:21.077
15	15	4	DATE	dimension	Store Date Dimension	2023-12-13 11:15:21.080
16	16	4	payment	dimension	Store payment Dimension	2023-12-29 08:10:26.000
17	17	4	city	dimension	Store city Dimension	2023-12-29 08:10:26.000
18	18	4	product_line	dimension	Store product_line Dimension	2023-12-22 17:03:30.000
19	19	4	product	dimension	Store product Dimension	2023-12-29 08:10:26.000
20	20	4	customer_type	dimension	Store customer Dimension	2023-12-29 08:10:26.000

Bảng DATA_TABLE_INFO lưu thông tin các table có trong nguyên hệ thống, DBS_ID chứa Table đó và ngày mà Table được UPDATE. Sau mỗi lần ETL, nếu table được cập nhật thì UPDATEDATE của table đó sẽ thay đổi theo giá trị ngày mới.

COLUMN_INFO

	COLUMN_ID	TABLE_ID	COLUMN_NAME	COLUMN_TYPE	DESCRIPTIONS	SAMPLE_VALUES	SOURCE_COLUMN_KEY_ID
37	37	12	ProductID_NK	nvarchar	Store productline nature key	PID01	NULL
38	38	12	Unit price	float	Store unit price	74.69	NULL
39	39	12	ProductLineID_SK	int	Store productline surrogate k...	1	37
40	40	13	InvoiceID_SK	int	Store invoice ID surrogate key	1	NULL
41	41	13	Invoice ID_NK	nvarchar	Store invoice ID surrogate key	1	NULL
42	42	13	Branch_SK	int	Store branch ID surrogate key	1	32
43	43	13	customer_typeID_SK	int	Store cuustomer type ID surr...	1	35
44	44	13	Gender	int	Store customer gender	1	NULL
45	45	13	ProductID_SK	int	Store product ID surrogate k...	1	40
46	46	13	Quantity	int	Store product quantity	2	NULL
47	47	13	Tax 5%	float	Store tax amount	26.1415	NULL
48	48	13	Total	float	Store Total amouunt of invoice	548.9715	NULL
49	49	13	DateKey	varchar	Store invoice create Date key	20231223	23
50	50	13	TimeKey	varchar	Store invoice create Time key	2359	27
51	51	13	paymentID_SK	int	Store payment ID surrogate ...	1	30
52	52	13	cogs	float	Store cost of goods sold	522.83	NULL
53	53	13	gross margin percentage	float	Store gross margin percenta...	4.761904762	NULL
54	54	13	gross income	float	Store gross income amount	26.1415	NULL
55	55	13	Rating	float	Store ratings statistics	9.1	NULL

Bảng COLUMN_INFO lưu thông tin từng cột trong các bảng, TABLE_ID chứa cột đó và ID của cột là giá trị tham chiếu đến nếu có

DATA_FIRE

	ID	COLUMN_ID	RULE
1	1	24	([Day]>0 && [Day]<32 && [Month]==1,3,5,5,7,8,10,12)|...
2	2	25	[Month]>=1 || [Month]<=12 ? [Month] : 1
3	3	26	[Year]<1 || [Year]>Year(GETDATE()) ? Year(GETDAT...
4	4	28	[Hour]<0 || [Hour]>23 ? 12 : [Hour]
5	5	29	[Minute]<0 || [Minute]>59 ? 0 : [Minute]
6	6	42	[Unit price]<0 ? 1 : [Unit price]
7	7	50	[Quantity]<0 ? 1 : [Quantity]
8	8	51	[Tax 5%]!=[Cogs]*0.05 ? [Cogs]*0.05 : [Tax 5%]
9	9	52	[Total]!=[Cogs]-[Tax 5%] ? [Cogs]-[Tax 5%] : [Total]
10	10	42	[Unit price] < 0 || ISNULL([Unit price]) ? 1 : [Unit price]
11	11	50	[Quantity] < 0 || ISNULL([Quantity]) ? 1 : [Total]
12	12	51	[Tax 5%] < 0 || ISNULL([Tax 5%]) ? 1 : [Tax 5%] = [Cog...
13	13	52	[Total] < 0 || ISNULL([Total]) ? 1 : [Total] = [Cogs] - [Tax...
14	14	56	[cogs] < 0 || ISNULL([cogs]) ? 1 : [cogs] = [Quantity] * [...
15	15	57	[gross margin percentage] < 0 || ISNULL([gross margin...
16	16	58	[gross income] < 0 || ISNULL([gross income]) ? 1 : [Tot...
17	17	59	[Rating] <= 0 || [Rating] >= 10 || ISNULL([Rating]) ? 0 : ...

Bảng DATA_FIRE lưu thông tin biểu thức tính toán được áp dụng cho những cột trong NDS và ID của cột được áp dụng công thức đó

3. **Thông tin NDS:**

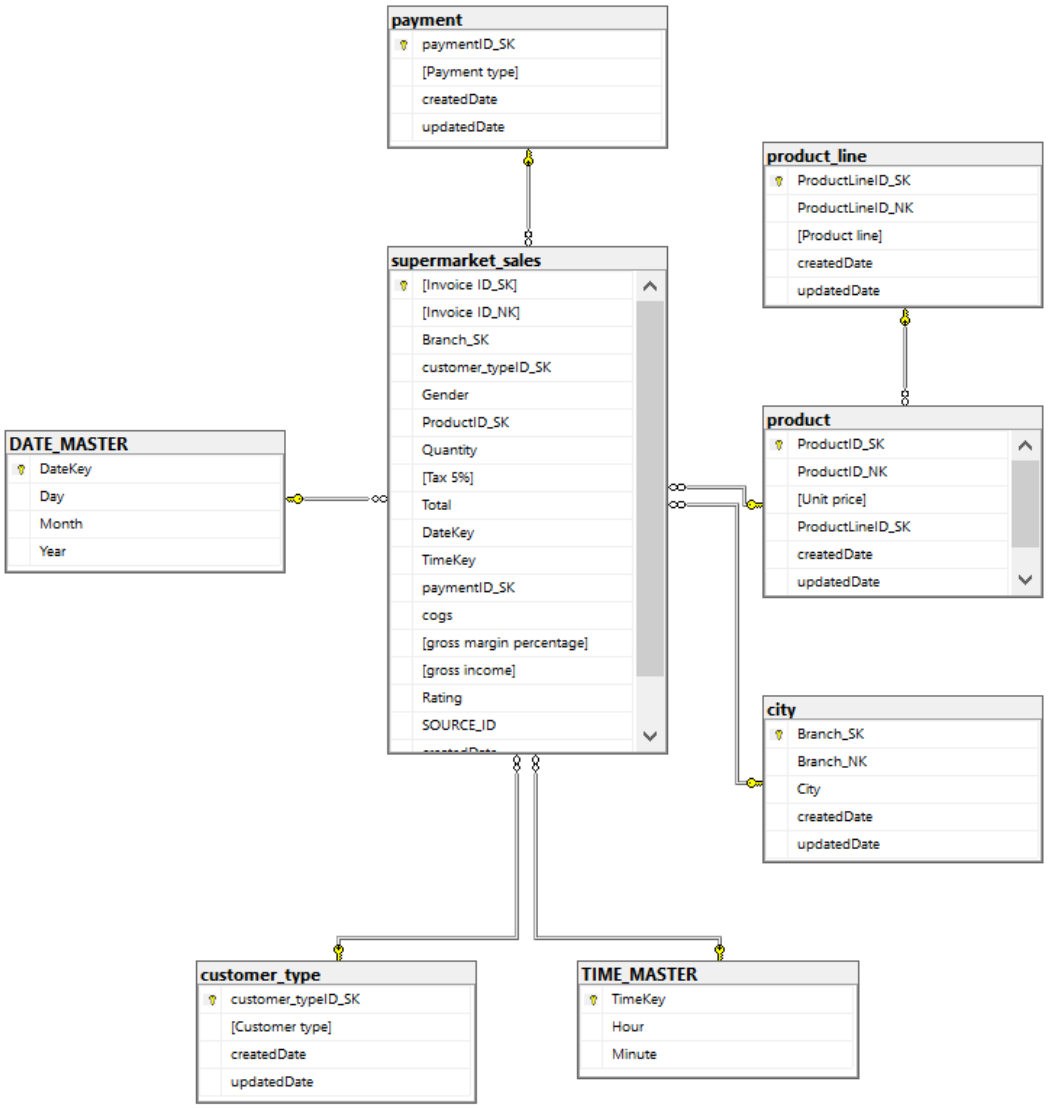
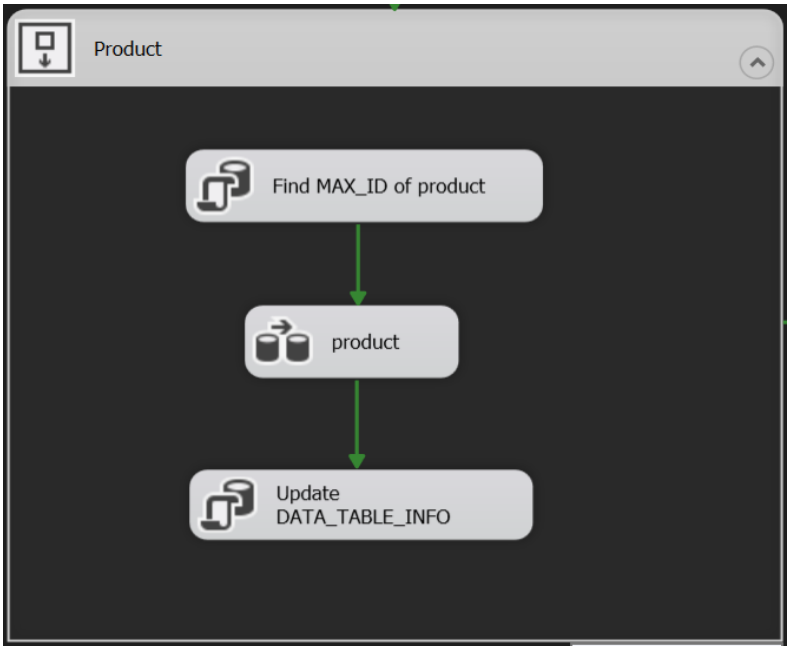


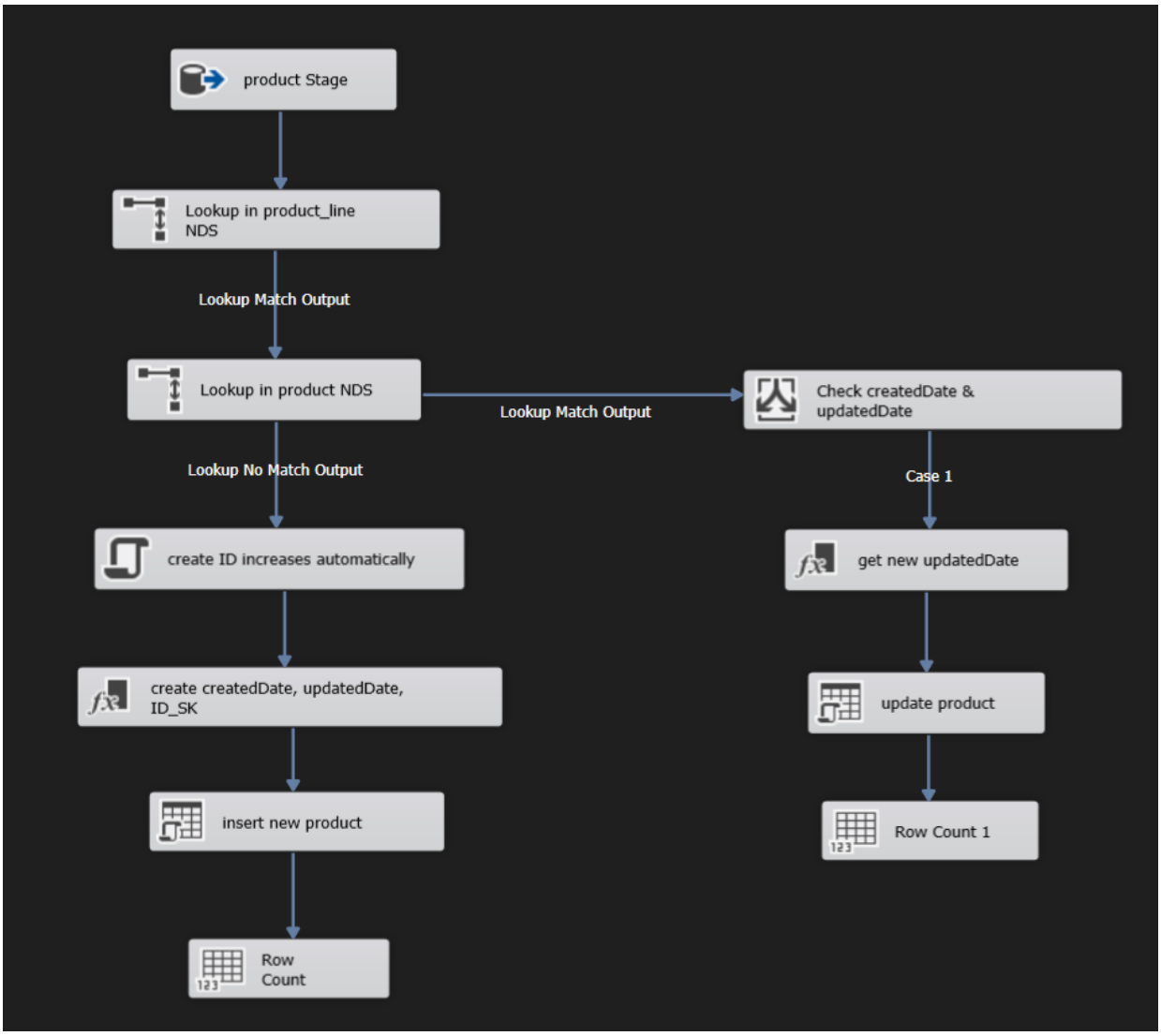
Diagram Supermarket_Sales NDS

- Product (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
ProductID_SK		int	1
ProductID_NK	product Stage (ProductID)	nvarchar	PID01
Unit price	product Stage (Unit Price)	float	74.69

ProductLineID_SK	product_line NDS (ProductLineID_SK)	int	1
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





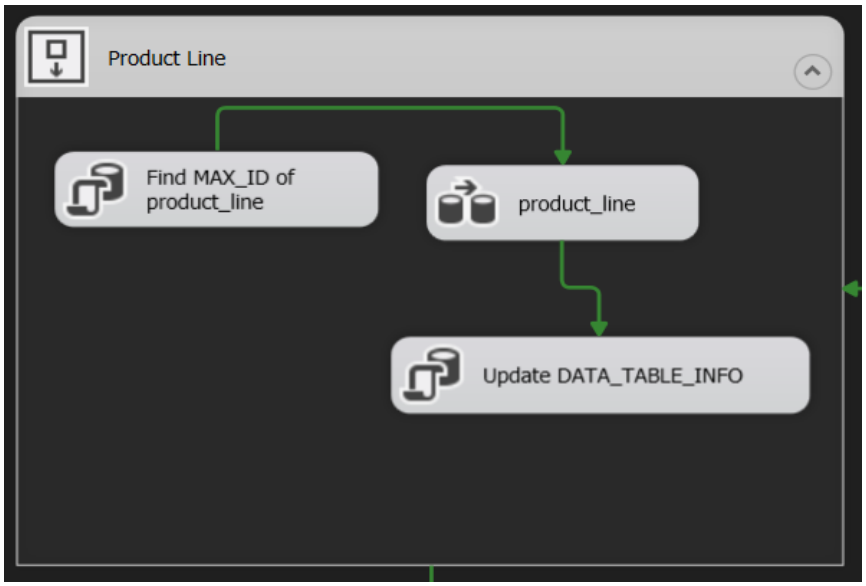
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột ProductID_SK từ bảng product trong NDS	Execute SQL Task
1.2		Tạo một Data Flow cho việc ETL bảng product từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ product Stage	OLEDB Source

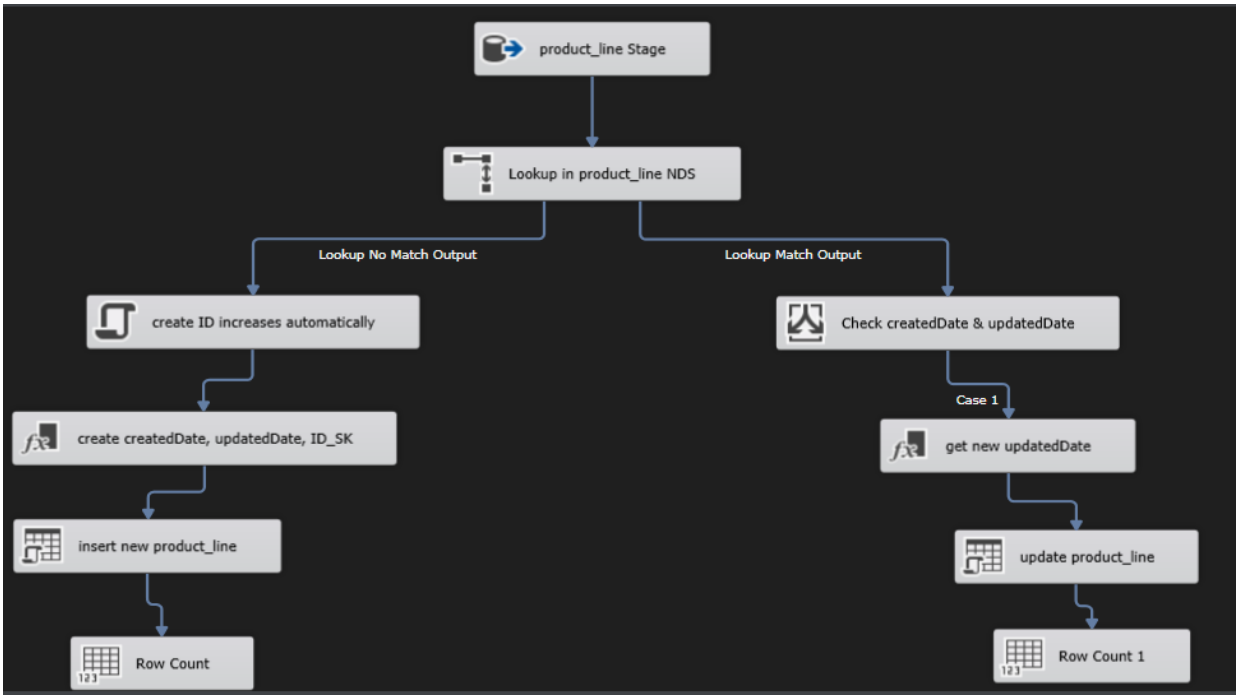
2.2	-	Lookup giá trị ProductLineID trong bảng ProductLine NDS	Lookup
2.2.1	no match	Bỏ qua row và không tiếp tục ETL	
2.2.2	match	Lookup giá trị ProductID trong bảng product NDS	Lookup
2.2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của ProductLine vừa tìm được xem có thoả điều kiện ETL không	Conditional Spilt
		Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng product	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2.2	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
		Tạo createdDate & updatedDate = CET, và ProductLineID_SK = MAX_ID + ID	Derived Column
		Thực hiện thêm row vào product	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng product NDS trong METADATA	Execute SQL Task

- ProductLine (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
ProductLineID_SK		int	1

ProductLineID _NK	ProductLine_Stage (ProductLineID)	nvarchar	PD001
ProductLine	ProductLine_Stage (product line)	nvarchar	Health and beauty
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





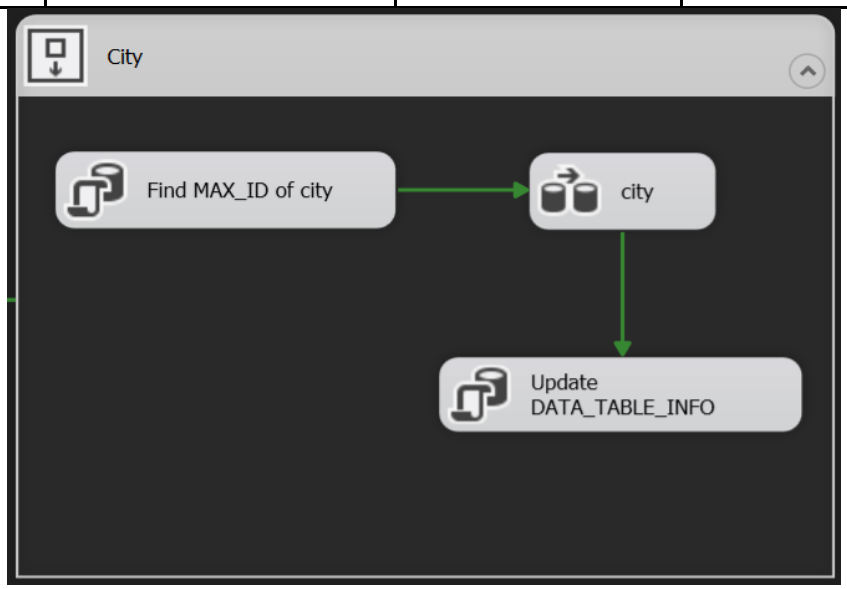
Các bước thực hiện:

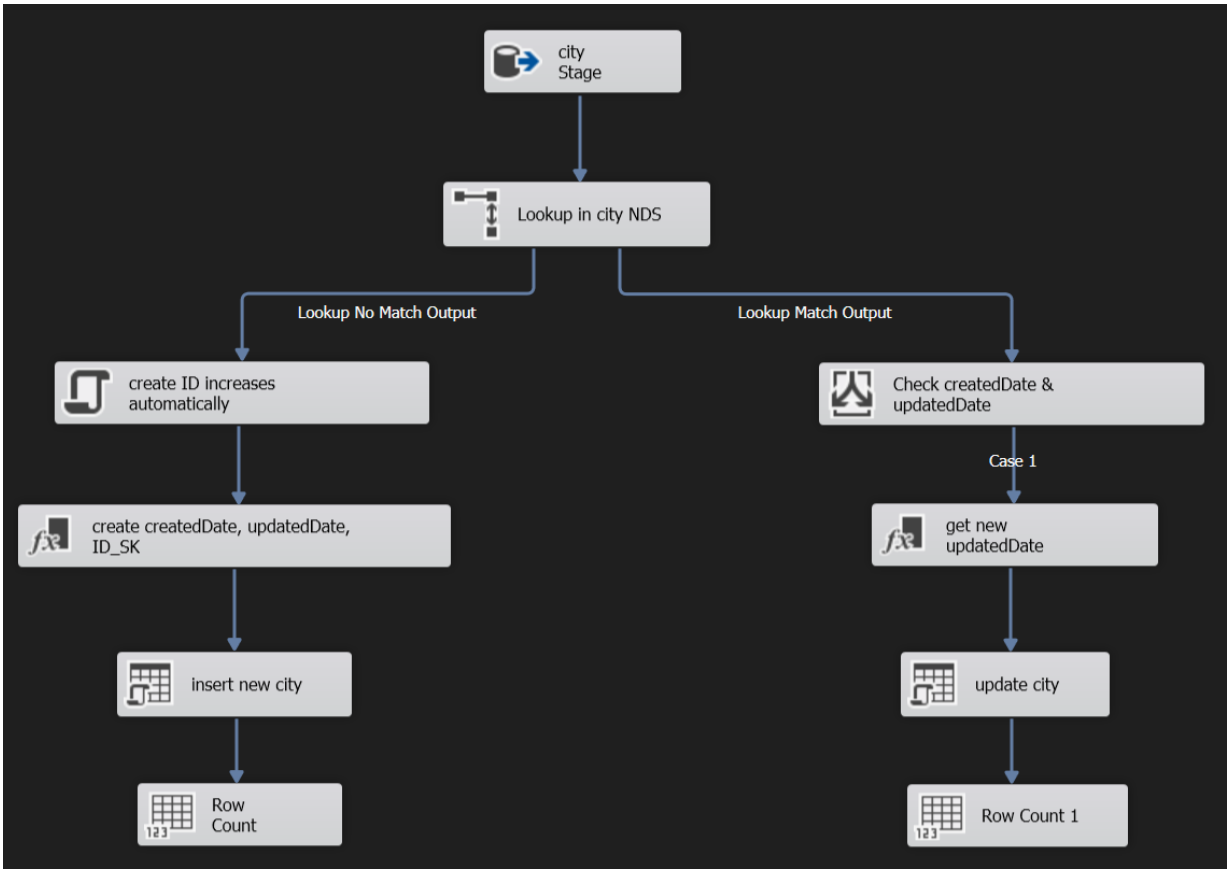
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột ProductLineID_SK từ bảng ProductLine trong NDS	Execute SQL
1.2		Tạo một Data Flow cho việc ETL ProductLineID từ Stage vào NDS	Data Flow Task
2. Data Flow Task			
2.1		Lấy từ bộ dữ liệu từ ProductLine (stage)	OLE DB Source
2.2		Kiểm tra xem ProductLineID lấy từ stage có tồn tại trong NDS chưa thông qua việc so sánh ProductLineID với ProductLineID_NK	Lookup
2.3.1	match	tiến hành kiểm tra createdDate & updatedDate của ProductLine vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
		Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng ProductLine	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.3.2	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
		Tạo createdDate & updatedDate = CET, và ProductLineID_SK = MAX_ID + ID	Derived Column
		Thực hiện thêm những hàng ProductLine	OLE DB Command

		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng ProductLine NDS vào METADATA	Execute SQL Task/

- City (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
Branch_SK		int	1
Branch_NK	City (Branch)	nvarchar	A
City	City (City)	nvarchar	Yangon
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36



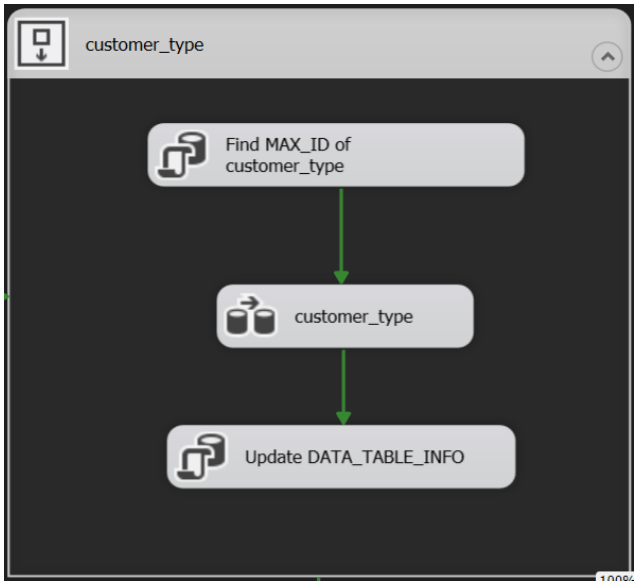


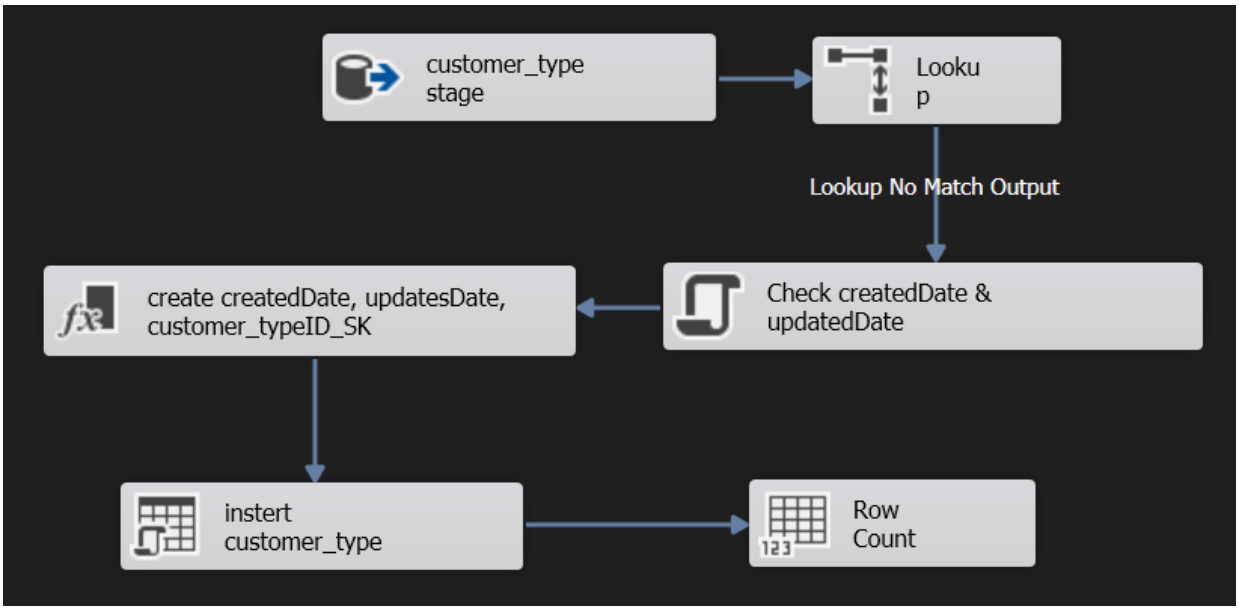
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột Branch_SK từ bảng city trong NDS	Execute SQL Task
1.2		Tạo Data Flow cho việc ETL trong bảng city từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ city Stage	OLEDB Source
2.2	-	Lookup giá trị Branch_NK trong bảng city NDS	Lookup
2.2.1	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component

		Tạo createdDate & updatedDate = CET, và ProductLineID_SK = MAX_ID + ID	Derived Column
		Thực hiện insert row vào city NDS	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
2.2.2	match	tiến hành kiểm tra createdDate & updatedDate của ProductLine vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
		Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng city	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng city NDS trong METADATA	Execute SQL Task

- CustomerType (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
customer_typeID_SK		int	1
Customer type	supermarket_sales Stage (Customer type)	nvarchar	Member
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột customer_typeID_SK từ bảng customer_type trong NDS	Execute SQL Task
1.2		Tạo Data Flow cho việc ETL dữ liệu liên quan đến customer type trong bảng	Data Flow Task

		supermarket_sales từ Stage vào NDS	
2. Data Flow			
2.1	-	Lấy dữ liệu từ supermarket_sales Stage	OLEDB Source
2.2	-	Lookup giá trị customer type trong bảng customer_type NDS	Lookup
2.2.1	no match	tiến hành kiểm tra createdDate & updatedDate của customer_type vừa tìm được xem có thoả điều kiện ETL không	Script Component
		Tạo createdDate & updatedDate = CET, và ProductLineID_SK = MAX_ID + ID	Derived Column
		Thực hiện insert row vào customer_type NDS	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng customer_type NDS trong METADATA	Execute SQL Task

- DATE_MASTER

DATE_MASTER lưu trữ ngày tháng năm (đã tạo sẵn trong DB)

Không thực hiện ETL trên bảng này vì đã thiết lập sẵn trên database

Tên cột	Nguồn	Kiểu DL	Value mẫu
DateKey		varchar(8)	20232412
Day		int	24
Month		int	12
Year		int	2023

- TIME_MASTER

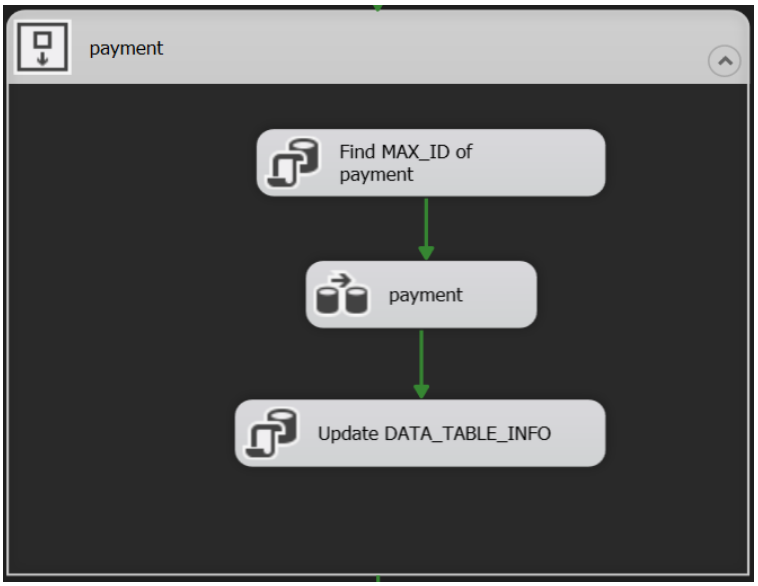
TIME_MASTER lưu thông tin giờ - phút (đã tạo sẵn trong DB)

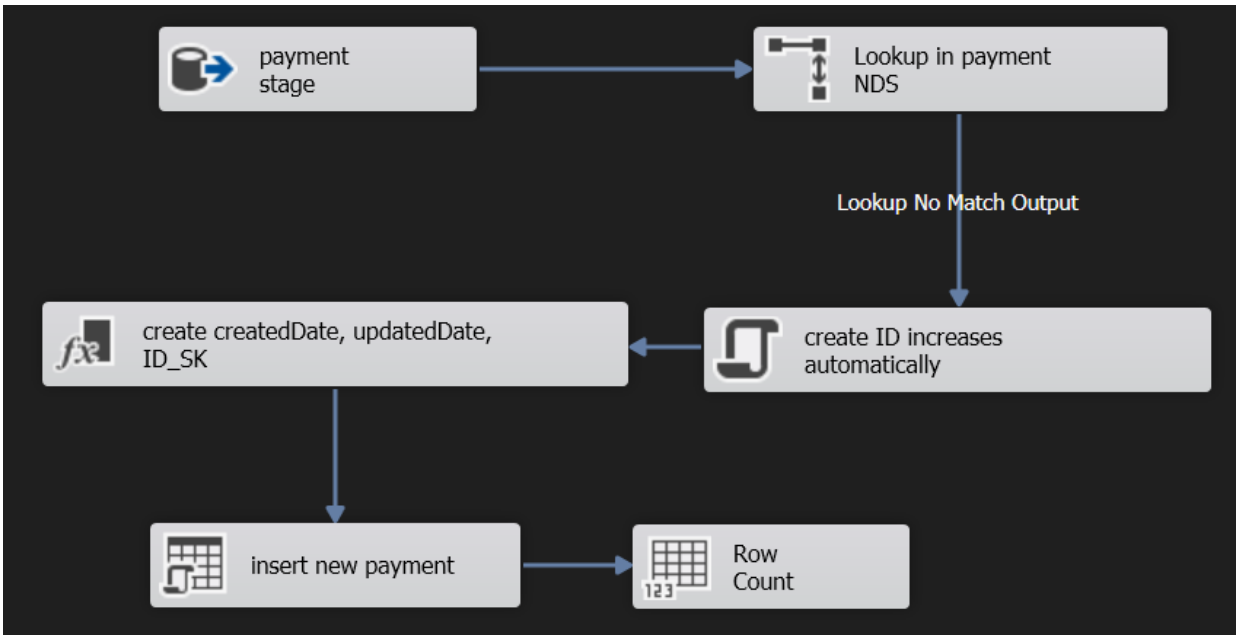
Không thực hiện ETL trên bảng này vì đã thiết lập sẵn trên database

Tên cột	Nguồn	Kiểu DL	Value mẫu
TimeKey		varchar(4)	2359
Hour		int	23
Minute		int	59

- Payment (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
Payment_SK		int	1
Payment type	supermarket_sales (Payment)	nvarchar	Ewallet
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





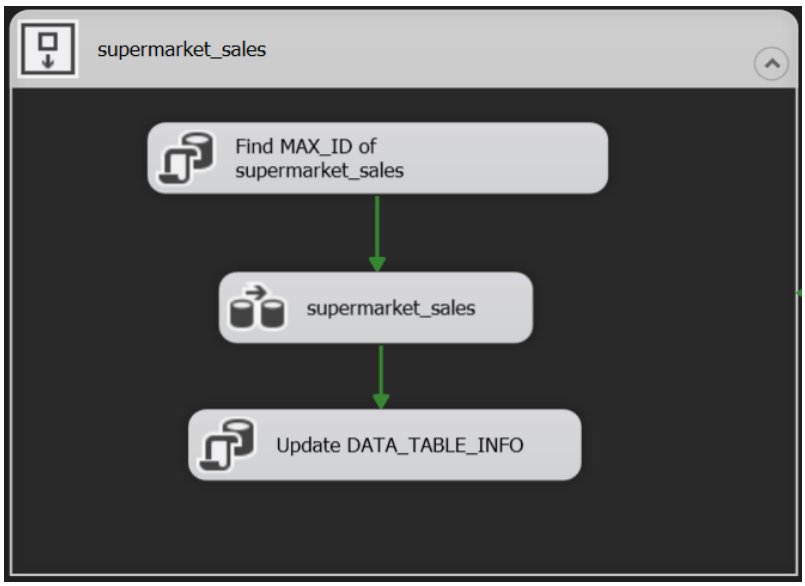
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột paymentID_SK từ bảng payment trong NDS	Execute SQL Task
1.2		Tạo Data Flow cho việc ETL dữ liệu liên quan đến payment type trong bảng supermarket_sales từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ supermarket_sales Stage	OLEDB Source
2.2	-	Lookup giá trị payment type trong bảng payment NDS	Lookup
2.2.1	match	Bỏ qua row	
2.2.2	no match	tiến hành kiểm tra createdDate & updatedDate của payment type vừa tìm được xem có thoả điều kiện ETL không	Script
		Tạo createdDate & updatedDate = CET,	Derived Column

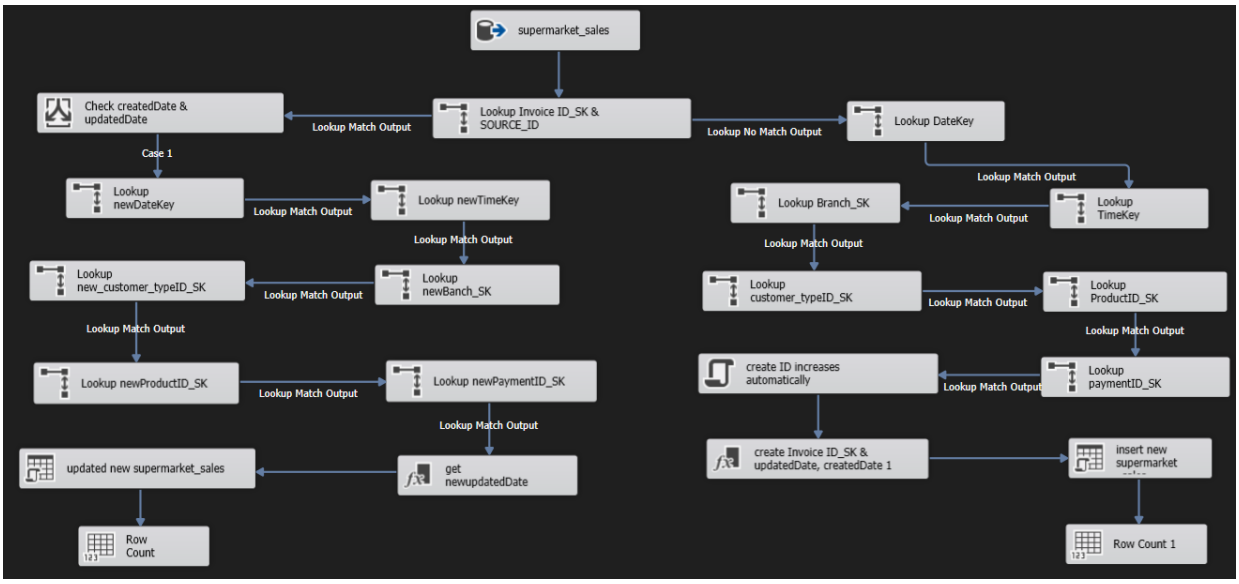
		và ProductLineID_SK = MAX_ID + ID	
		Thực hiện thêm row vào payment NDS	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng customer_type NDS trong METADATA	Execute SQL Task

- Supermarket_sales (Transaction)

Tên cột	Nguồn	Kiểu DL	Value mẫu
InvoiceID_SK		int	1
Invoice ID_NK	supermarket_sales Stage (Invoice ID)	nvarchar	750-67-8428
Branch_SK	city NDS (Branch_SK)	int	1
customer_typeID_SK	customer_type NDS(customer_typeID_SK)	int	1
Gender		int	1
ProductID_SK	product NDS (ProductID_SK)	int	1
Quantity	supermarket_sales Stage (Quantity)	int	2
Tax 5%	supermarket_sales Stage (Tax 5%)	float	26.1415
Total	supermarket_sales Stage (Total)	float	548.9715
DateKey	DATE_MASTER	varchar	20231223

	(DateKey)		
TimeKey	TIME_MASTER (TimeKey)	varchar	2359
paymentID_SK	supermarket_sales Stage (paymentID_SK)	int	1
cogs	supermarket_sales Stage (cogs)	float	522.83
gross margin percentage	supermarket_sales Stage (gross margin percentage)	float	4.761904762
gross income	supermarket_sales Stage (gross income)	float	26.1415
Rating	supermarket_sales Stage (Rating)	float	9.1
SOURCE_ID		int	1
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột InvoiceID_SK từ bảng supermarket_sales trong NDS	Execute SQL Task
1.2		Tạo Data Flow cho việc ETL dữ liệu bảng supermarket_sales từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ supermarket_sales Stage	OLEDB Source
2.2	-	Lookup giá trị InvoiceID trong bảng supermarket_sales NDS	Lookup
2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của ProductLine vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
	thỏa điều kiện	Lookup giá trị date có tồn tại trong DATE_MASTER không	Lookup
	match	Lookup giá trị time có tồn tại trong	Lookup

		TIME_MASTER không	
	match	Lookup giá trị BranchID trong bảng city NDS và lấy newBranchID_SK	Lookup
	match	Lookup giá trị customer type trong bảng customer_type NDS và lấy newcustomer_typeID_SK	Lookup
	match	Lookup giá trị ProductID trong bảng product NDS và lấy newProductID_SK	Lookup
	match	Lookup giá trị payment type trong bảng payment NDS và lấy newPaymentID_SK	Lookup
	match	Tạo ngày cập nhật mới bằng CET	Derived Column
	-	Thực hiện cập nhật dữ liệu mới vào bảng city	OLE DB Command
	-	Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2	no match	Lookup giá trị date có tồn tại trong DATE_MASTER không	Lookup
	match	Lookup giá trị time có tồn tại trong TIME_MASTER không	Lookup
	match	Lookup giá trị BranchID trong bảng city NDS và lấy BranchID_SK	Lookup
	match	Lookup giá trị customer type trong bảng customer_type NDS và lấy customer_typeID_SK	Lookup
	match	Lookup giá trị ProductID trong bảng product NDS và lấy ProductID_SK	Lookup
	match	Lookup giá trị payment type trong bảng payment NDS và lấy PaymentID_SK	Lookup
	match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component

	-	Tạo createdDate & updatedDate = CET, và InvoiceID_SK = MAX_ID + ID	Derived Column
	-	Thực hiện insert dữ liệu vào supermarket_sales	OLE DB Command
	-	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng supermarket_sales NDS trong METADATA	Execute SQL Task

4. Thông tin DDS

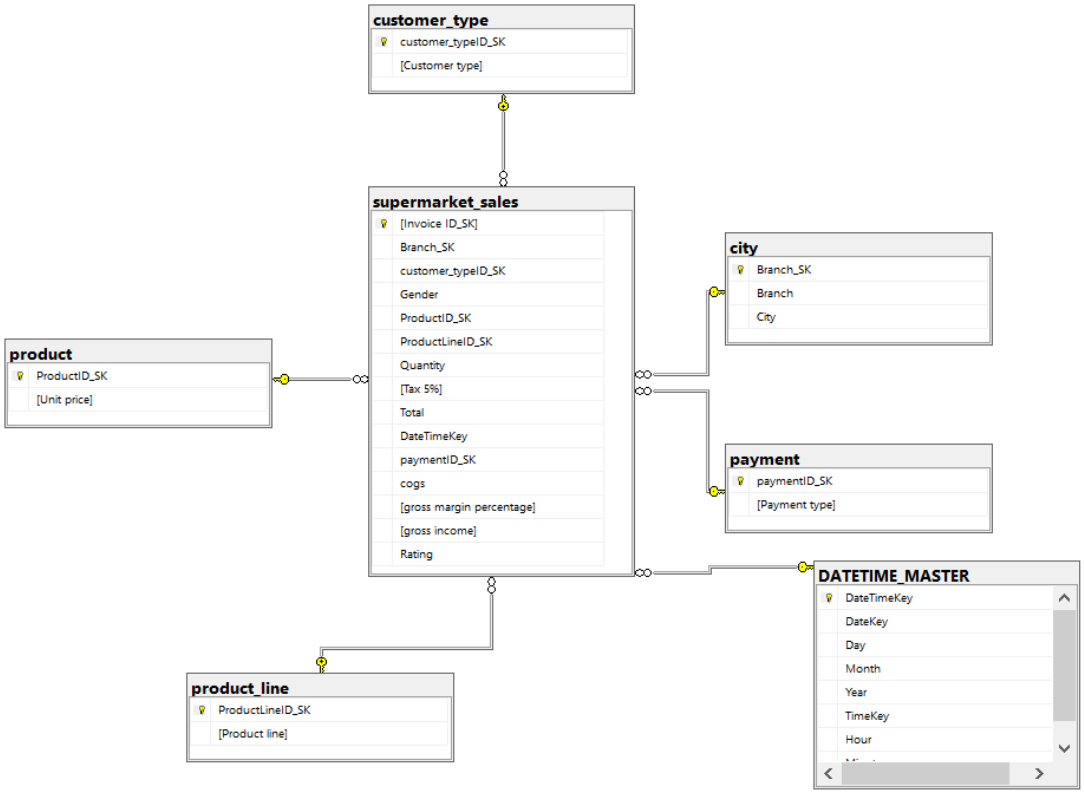
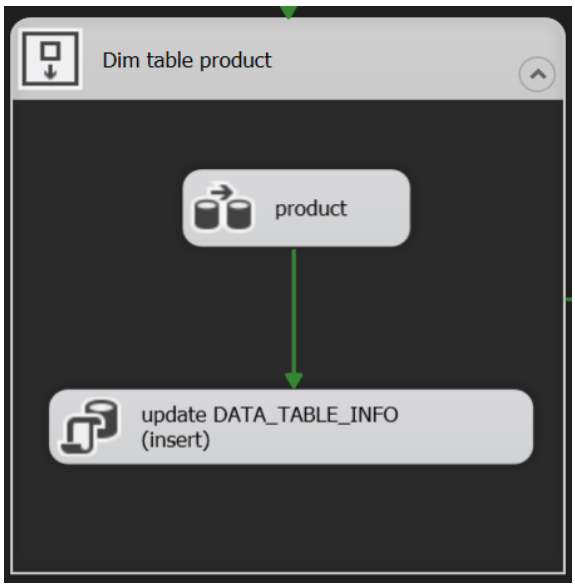
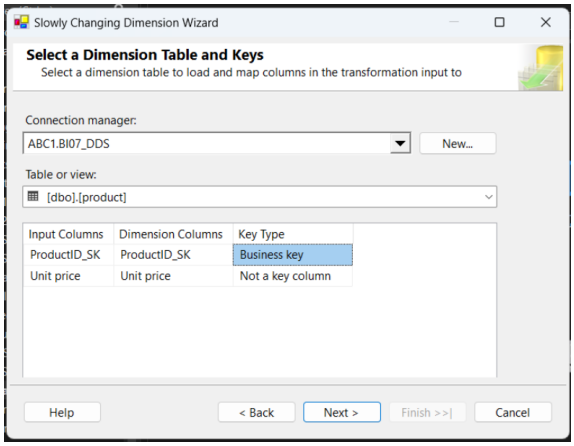
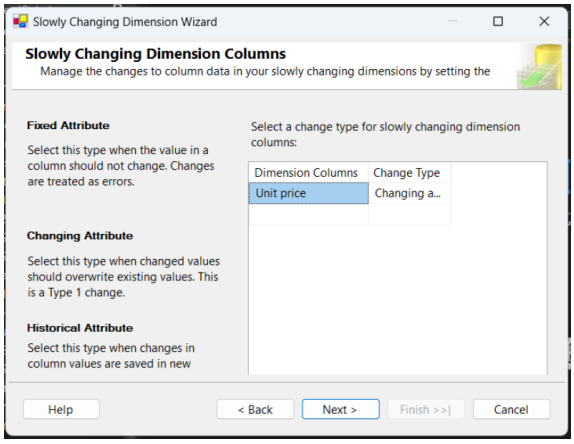


Diagram Supermarket_Sales DDS

- Product (Dimension)

Tên cột	Nguồn	Kiểu DL	Value mẫu
ProductID_SK		int	1
Unit price	product Stage (Unit Price)	float	74.69

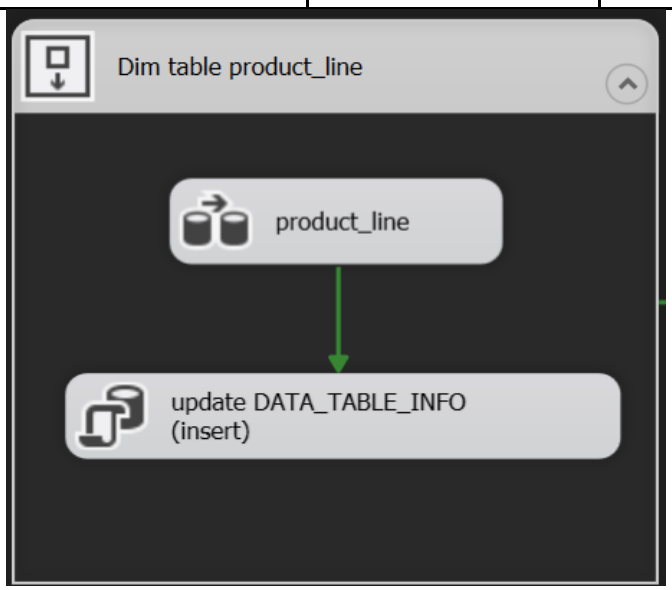


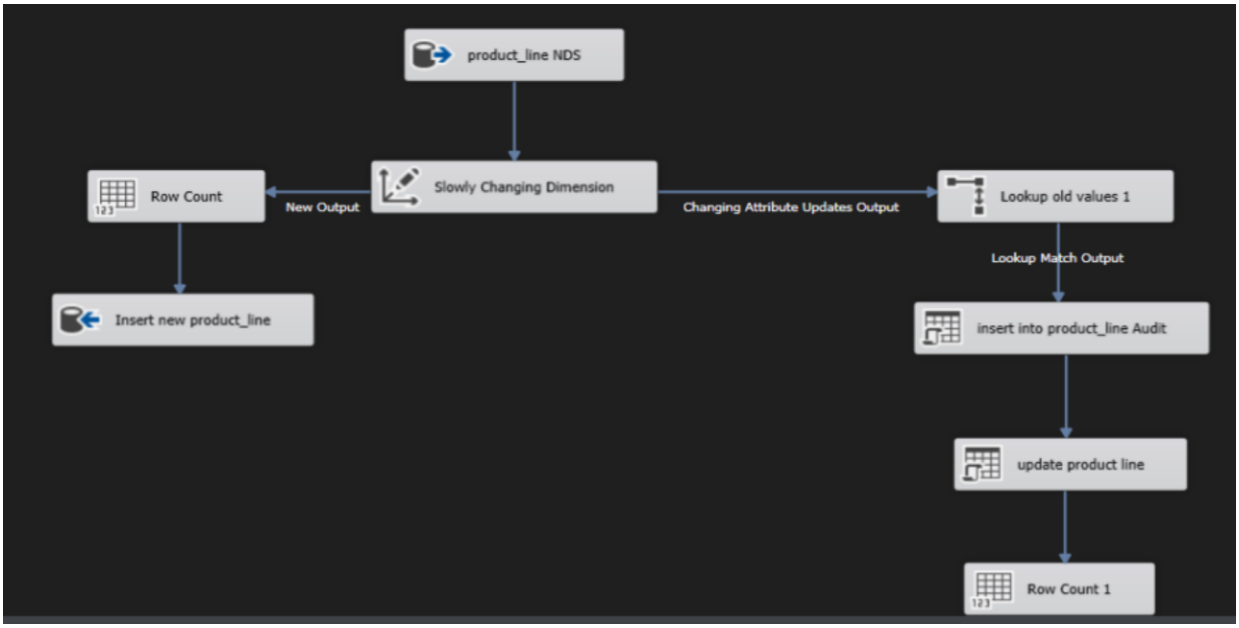
STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ product NDS	OLEDB Source
1.2		Tạo SCD	
			
			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào product DDS	OLEDB Destination
	Thay đổi giá trị cũ	Lookup các dòng dữ liệu cũ trong bảng DDS	LookUp

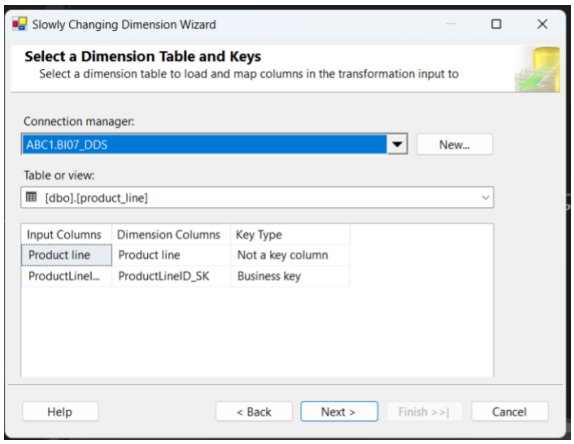
	match	Insert dòng dữ liệu vào bảng product_Audit	OLE DB Command
		Update dòng dữ liệu mới vào bảng product DDS	OLE DB Command
		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
2. Execute SQL Task			
2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng product DDS trong METADATA	Execute SQL Task

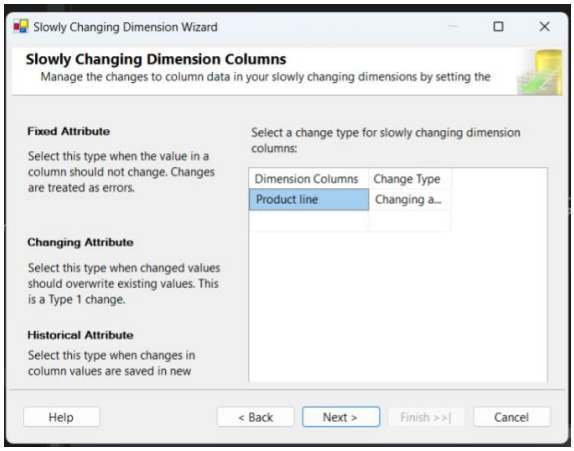
- ProductLine (Dimension)

Tên cột	Nguồn	Kiểu DL	Value mẫu
ProductLineID_SK		int	1
ProductLine	ProductLine_Stage (product line)	nvarchar	Health and beauty



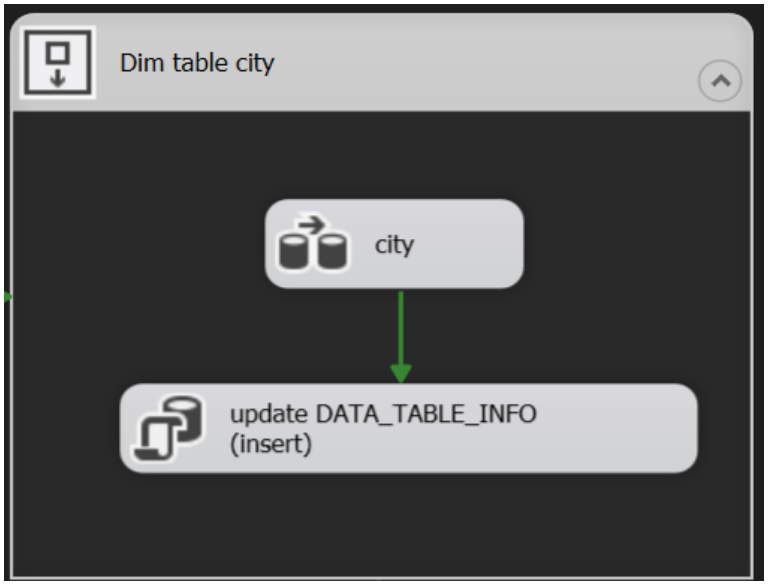


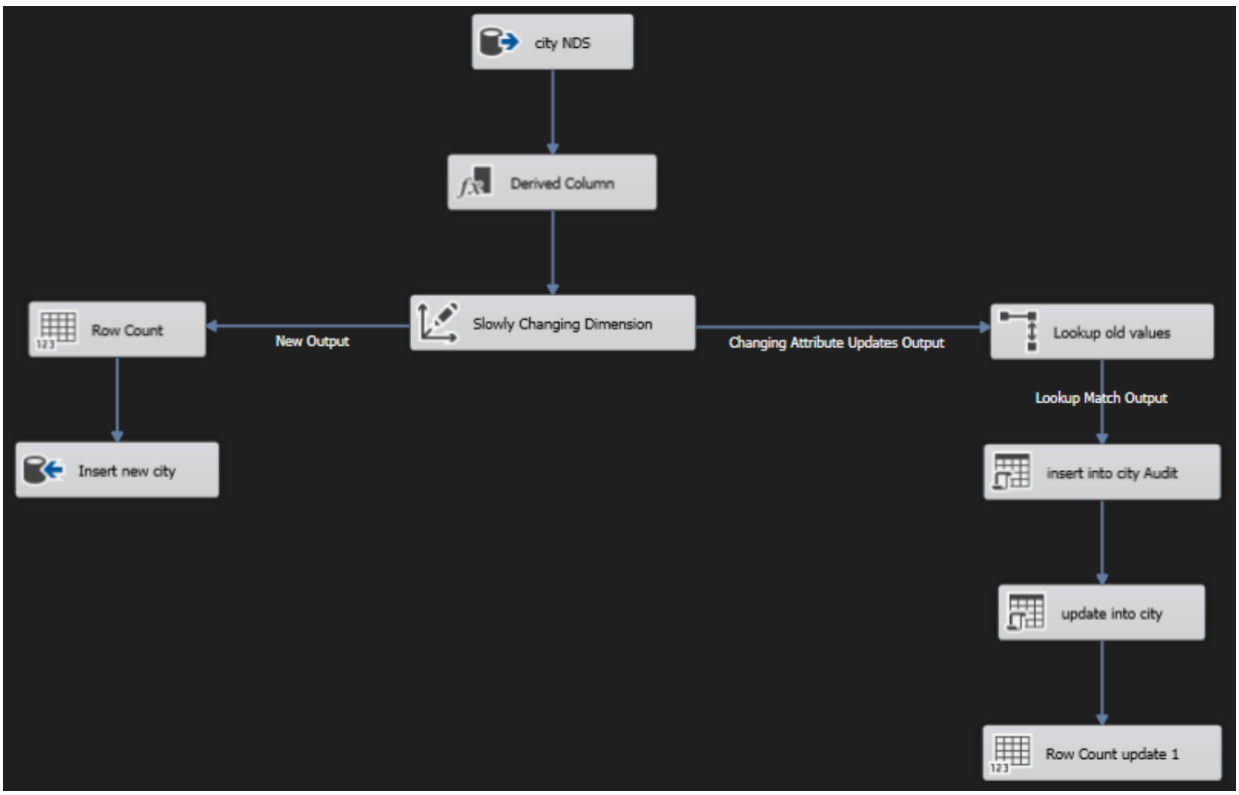
STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ product_line NDS	OLEDB Source
1.2		Tạo SCD	
			

			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào product_line DDS	OLEDB Destination
	Thay đổi giá trị cũ	Lookup các dòng dữ liệu cũ trong bảng DDS	LookUp
	match	Insert dòng dữ liệu vào bảng product_line_Audit	OLE DB Command
		Update dòng dữ liệu mới vào bảng product DDS	OLE DB Command
		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
2. Execute SQL Task			
2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng product_line DDS trong METADATA	Execute SQL Task

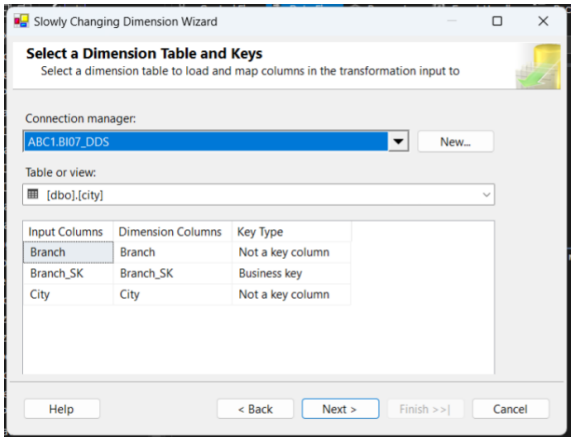
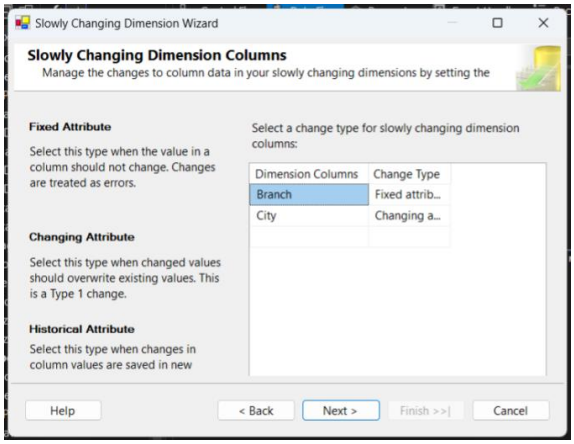
- City (Dimension)

Tên cột	Nguồn	Kiểu DL	Value mẫu
Branch_SK		int	1
City	City (City)	nvarchar	Yangon





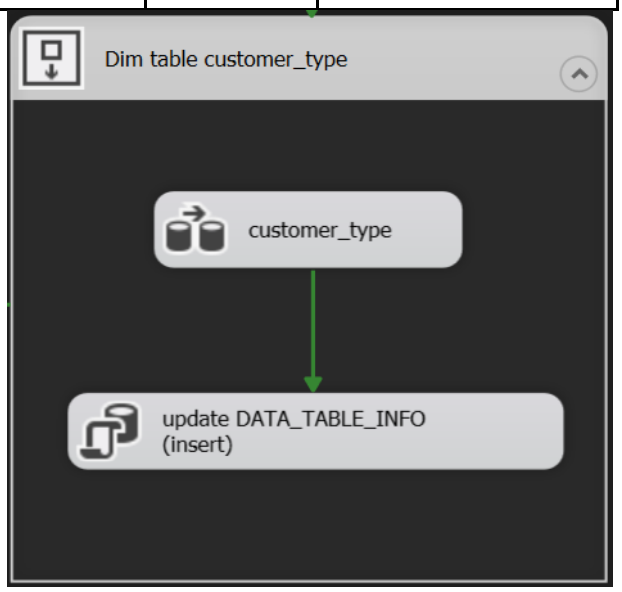
STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			

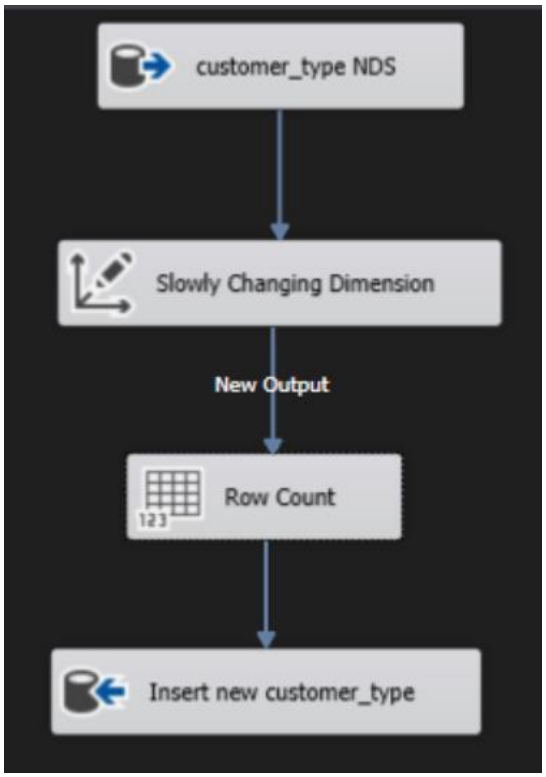
1.1	-	Lấy dữ liệu từ city NDS	OLEDB Source
1.2		Tạo SCD	
			
			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào city DDS	OLEDB Destination
	Thay đổi giá trị cũ	Lookup các dòng dữ liệu cũ trong bảng DDS	LookUp
	match	Insert dòng dữ liệu vào bảng city_Audit	OLE DB Command
		Update dòng dữ liệu mới vào bảng product DDS	OLE DB Command

		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
Execute SQL Task			
2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng city DDS trong METADATA	Execute SQL Task

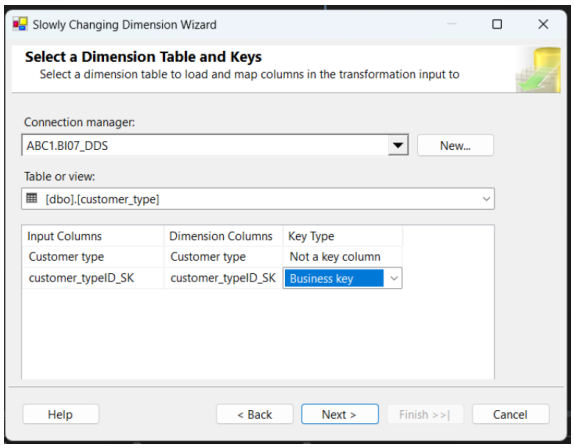
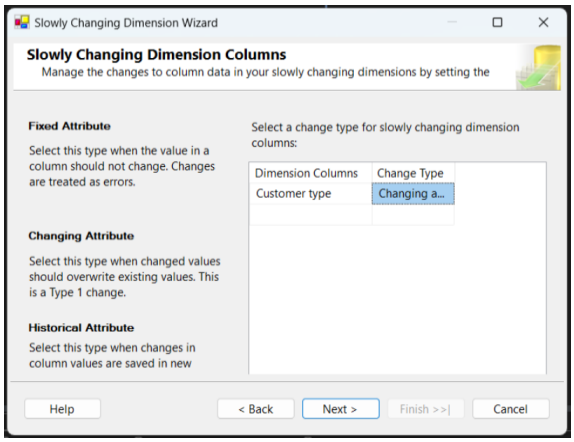
- CustomerType (Dimension)

Tên cột	Nguồn	Kiểu DL	Value mẫu
customer_type ID_SK		int	1
Customer type	supermarket _sales Stage (Customer type)	nvarchar	Member



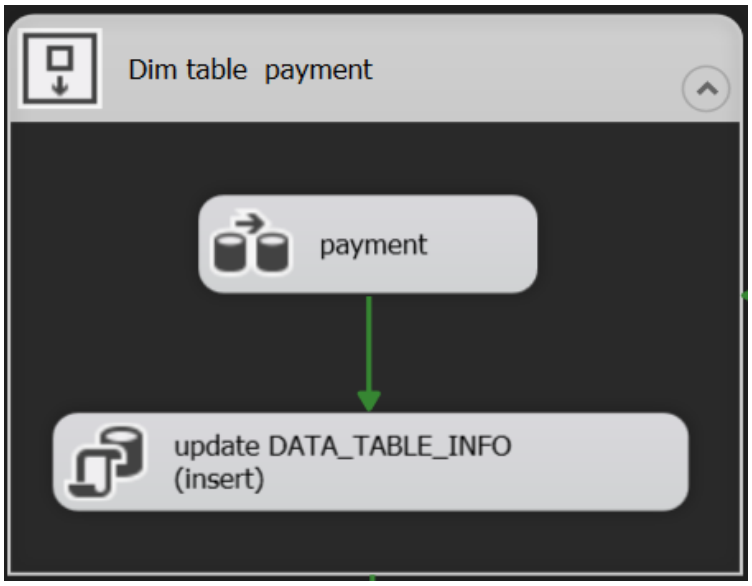


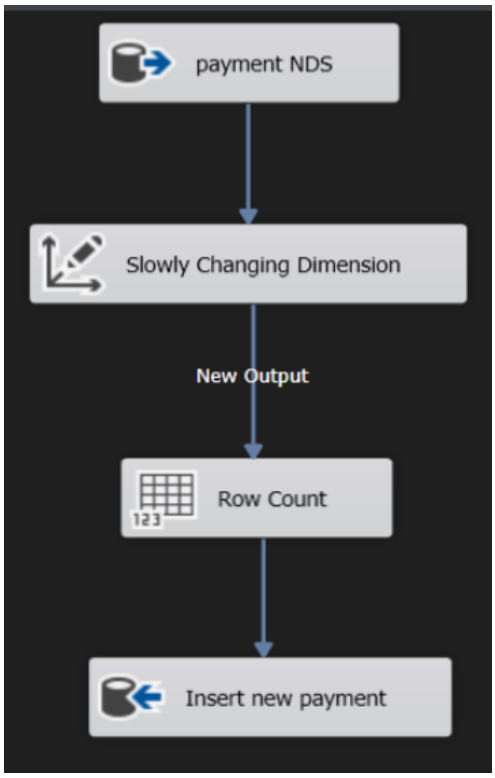
STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source
1.2		Tạo SCD	

			
			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào city DDS	OLEDB Destination
2. Execute SQL Task			
2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng city DDS trong METADATA	Execute SQL Task

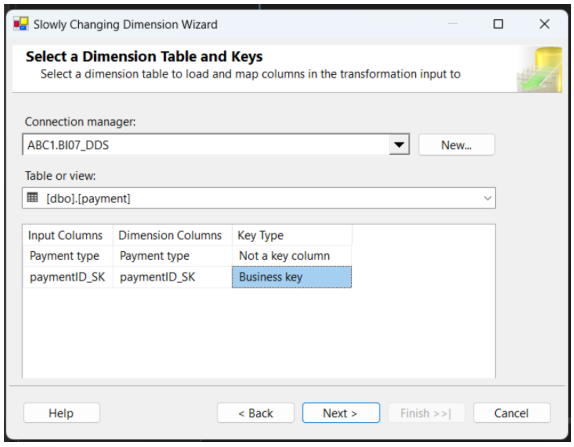
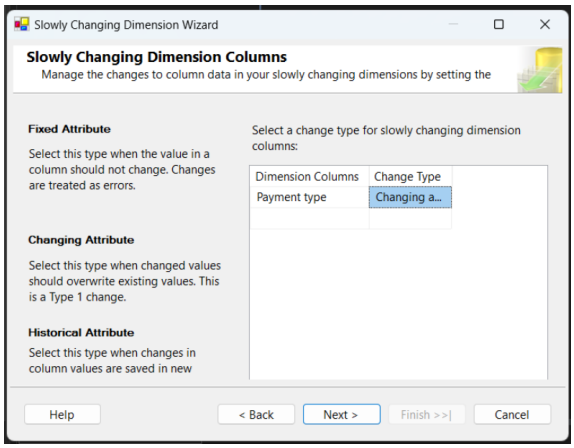
- Payment (Dimension)

Tên cột	Nguồn	Kiểu DL	Value mẫu
Payment_SK		int	1
Payment type	supermarket_sales (Payment)	nvarchar	Ewallet





STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source

1.2		Tạo SCD	
			
			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào city DDS	OLEDB Destination
2. Execute SQL Task			
2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng city DDS trong METADATA	Execute SQL Task

- DATETIME_MASTER (Dimension)

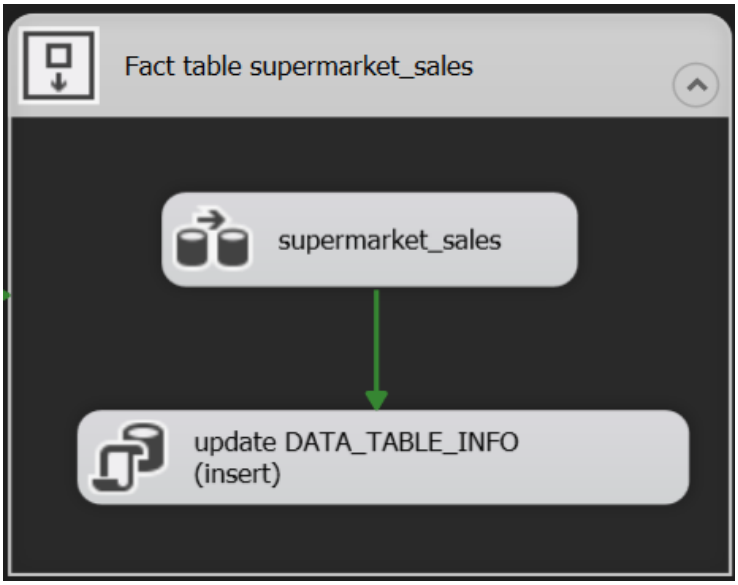
Tên cột	Nguồn	Kiểu DL	Value mẫu
DateTimeKey		varchar	202312242359

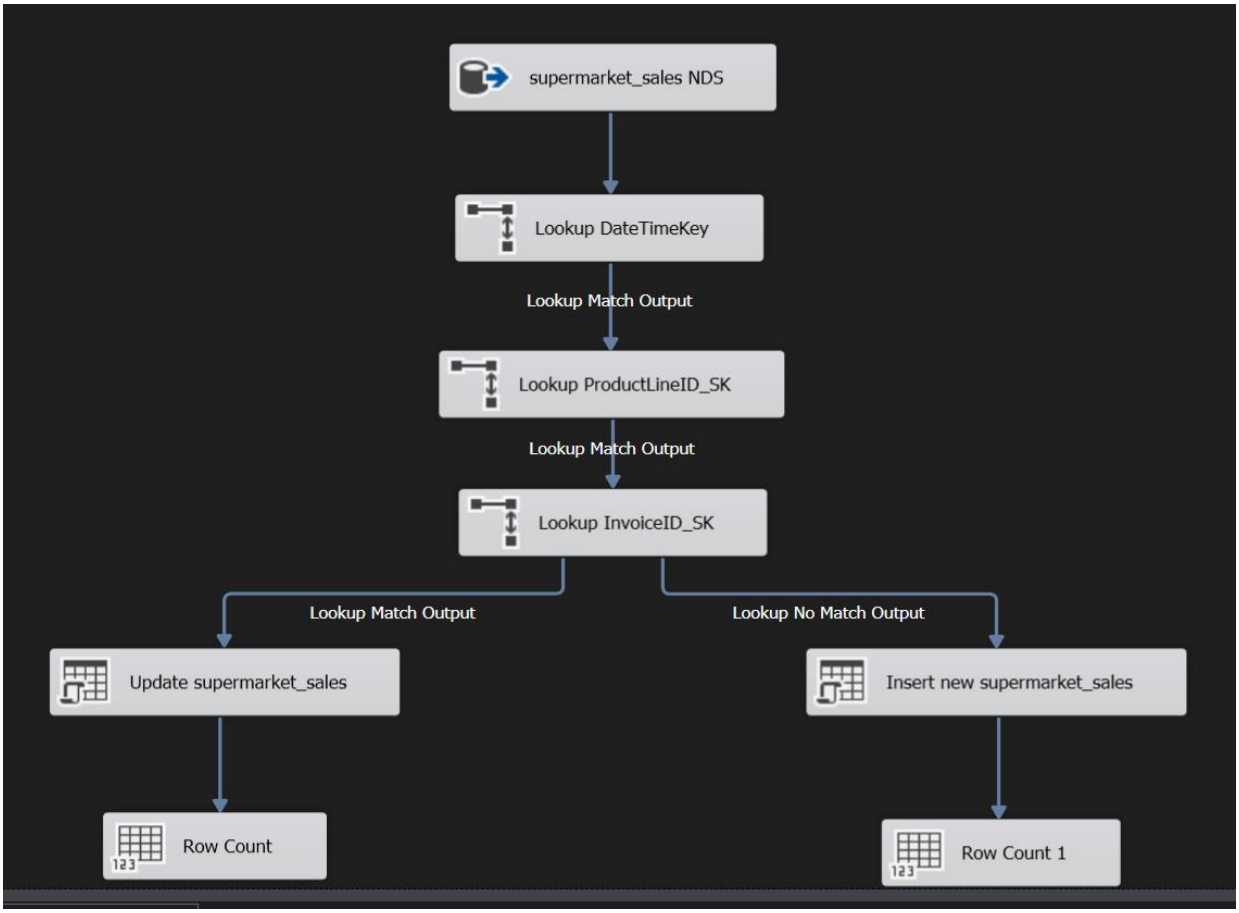
DateKey	DATE_MASTER (NDS)	varchar	20231224
TimeKey	TIME_MASTER (NDS)	varchar	2359
Day	DATE_MASTER (NDS)	int	24
Month	DATE_MASTER (NDS)	int	12
Year	DATE_MASTER (NDS)	int	2023
Hour	TIME_MASTER (NDS)	int	23
Minute	TIME_MASTER (NDS)	int	59

- supermarket_sales (Fact)

Tên cột	Nguồn	Kiểu DL	Value mẫu
InvoiceID_SK		int	1
Branch_SK	city NDS (Branch_SK)	int	1
customer_typeID_SK	customer_type NDS(customer_typeID_SK)	int	1
Gender		int	1
ProductID_SK	product NDS (ProductID_SK)	int	1
ProductLineID_SK	product_line NDS (ProductLineID_SK)	int	1
Quantity	supermarket_sales Stage (Quantity)	int	2
Tax 5%	supermarket_sales Stage (Tax 5%)	float	26.1415
Total	supermarket_sales Stage (Total)	float	548.9715
DateTimeKey	DATE_MASTER (DateKey)	varchar	202312232359

	TIME_MASTER (TimeKey)		
paymentID_SK	supermarket_sales Stage (paymentID_SK)	int	1
cogs	supermarket_sales Stage (cogs)	float	522.83
gross margin percentage	supermarket_sales Stage (gross margin percentage)	float	4.761904762
gross income	supermarket_sales Stage (gross income)	float	26.1415
Rating	supermarket_sales Stage (Rating)	float	9.1





STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ supermarket_sales NDS	OLEDB Source
1.2		Lookup DateTime và TimeKey trong bảng DateTimeKey	Lookup
	match	Lookup giá trị ProductLineID_SK trong bảng product_line NDS	Lookup
	match	Lookup giá trị InvoiceID_SK trong bảng supermarket_sales NDS	Lookup
	nomatch	Thực hiện insert dữ liệu vào supermarket_sales	OLE DB Command

		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	match	Thực hiện cập nhật dữ liệu mới vào bảng supermarket	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2. Execute SQL Task			
2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng supermarket DDS trong METADATA	Execute SQL Task

4. Danh sách các table thuộc về các database

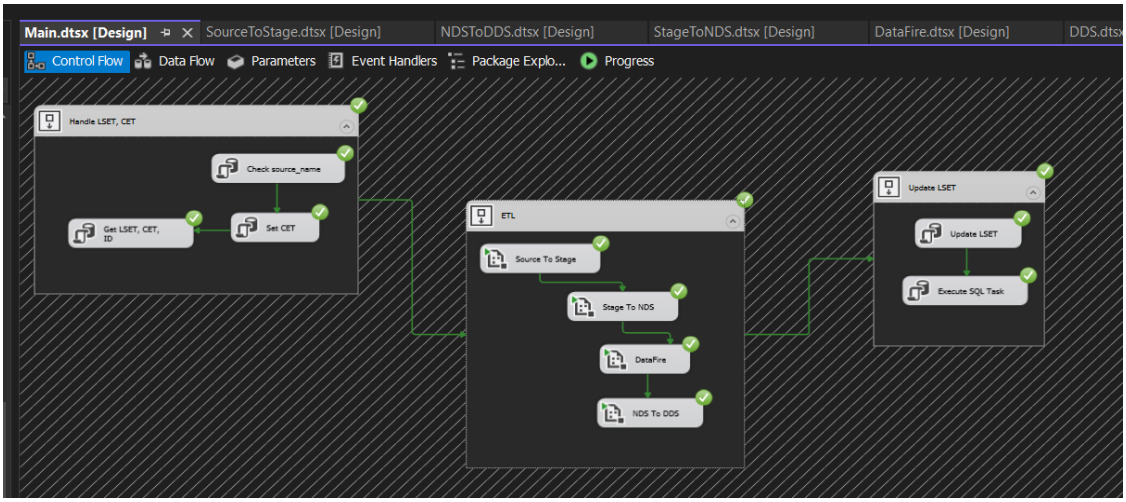
Database	**Table**	**Ý nghĩa**
BI07_METADATA	[dbo].[DATA_FLOW]	Lưu trữ thông tin ETL của các source
	[dbo].[DATA_STORE_INFO]	Lưu trữ thông tin các database trong hệ thống
	[dbo].[DATA_TABLE_INFO]	Lưu trữ thông tin các table trong từng database
	[dbo].[COLUMN_INFO]	Lưu trữ thông tin từng cột trong table
	[dbo].[DATA_FIRE]	Lưu trữ thông tin về ràng buộc dữ liệu
BI07_STAGE	[dbo].[city]	Lưu trữ thông tin Stage của bảng city
	[dbo].[product]	Lưu trữ thông tin Stage của bảng product
	[dbo].[product_line]	Lưu trữ thông tin Stage của

		bảng product_line
	[dbo].[supermarket_sales]	Lưu trữ thông tin Stage của bảng supermarket_sales
BI07_NDS	[dbo].[DATE_MASTER]	Lưu thông tin ngày, tháng, năm
	[dbo].[TIME_MASTER]	Lưu thông tin giờ, phút
	[dbo].[city]	Lưu trữ dữ liệu sau khi format lại từ city stage
	[dbo].[product]	Lưu trữ dữ liệu sau khi format lại từ product stage
	[dbo].[product_line]	Lưu trữ dữ liệu sau khi format lại từ product_line stage
	[dbo].[customer_type]	Lưu trữ dữ liệu sau khi format lại từ supermarket_sales stage
	[dbo].[payment]	Lưu trữ dữ liệu sau khi format lại từ supermarket_sales stage
	[dbo].[supermarket_sales]	Lưu trữ dữ liệu sau khi format lại từ supermarket_sales stage
BI07_DDS	[dbo].[DATE_MASTER]	Lưu thông tin ngày, tháng, năm
	[dbo].[TIME_MASTER]	Lưu thông tin giờ, phút
	[dbo].[city]	Lưu trữ dữ liệu city trong DWH
	[dbo].[city_Audit]	Lưu trữ các dữ liệu cũ của city trong DWH
	[dbo].[product]	Lưu trữ dữ liệu product trong DWH
	[dbo].[product_Audit]	Lưu trữ các dữ liệu cũ của product trong DWH

	[dbo].[product_line]	Lưu trữ dữ liệu product_line trong DWH
	[dbo].[product_line_Audit]	Lưu trữ các dữ liệu cũ của product_line trong DWH
	[dbo].[customer_type]	Lưu trữ dữ liệu customer_type trong DWH
	[dbo].[customer_type_Audit]	Lưu trữ các dữ liệu cũ của customer_type trong DWH
	[dbo].[payment]	Lưu trữ dữ liệu payment trong DWH
	[dbo].[payment_Audit]	Lưu trữ các dữ liệu cũ của payment trong DWH
	[dbo].[supermarket_sales]	

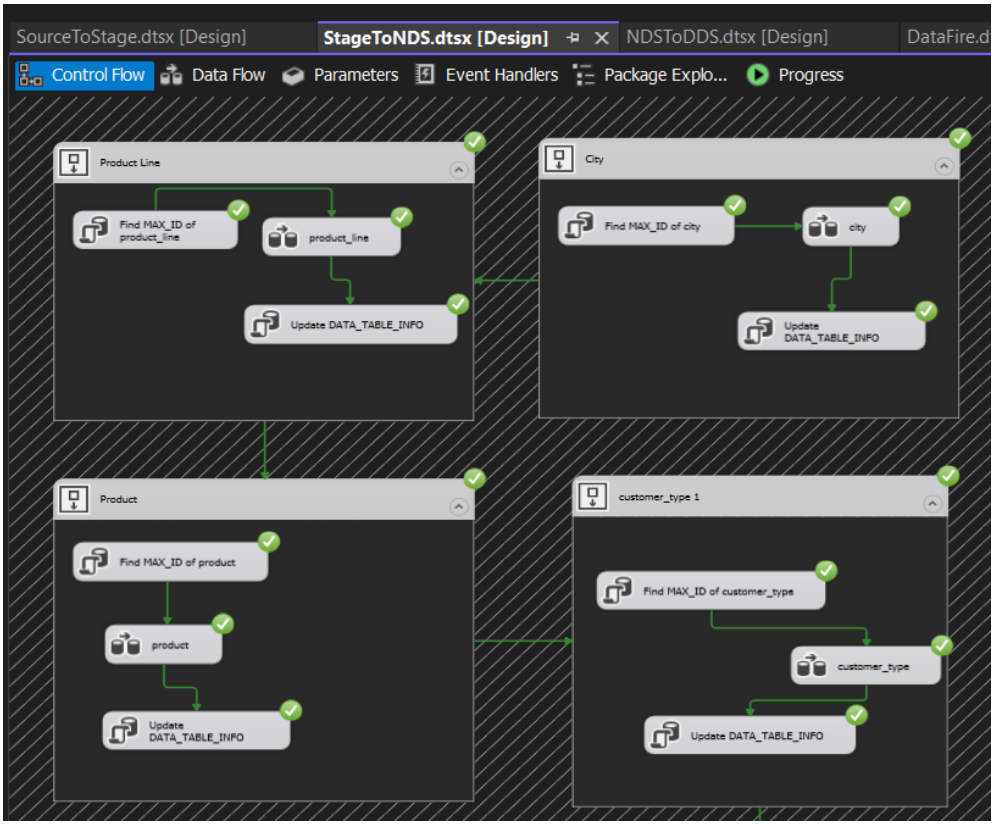
I. Hình các package

- Main



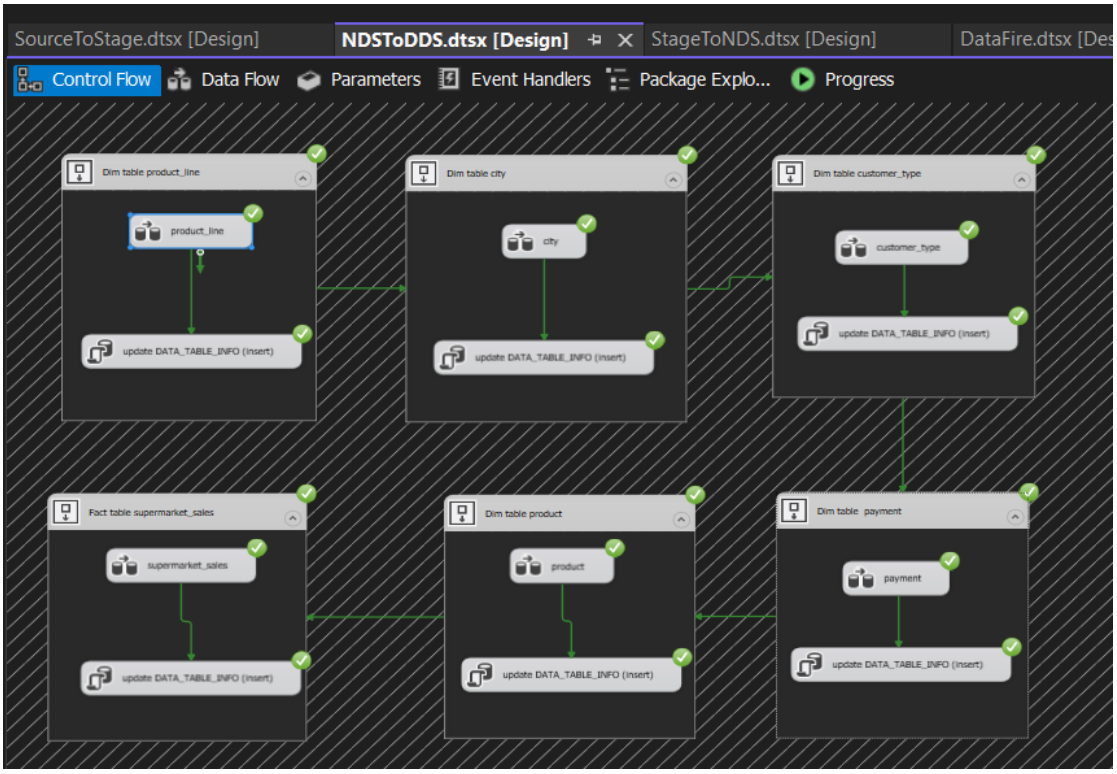
- SourceToStage

- StageToNDS

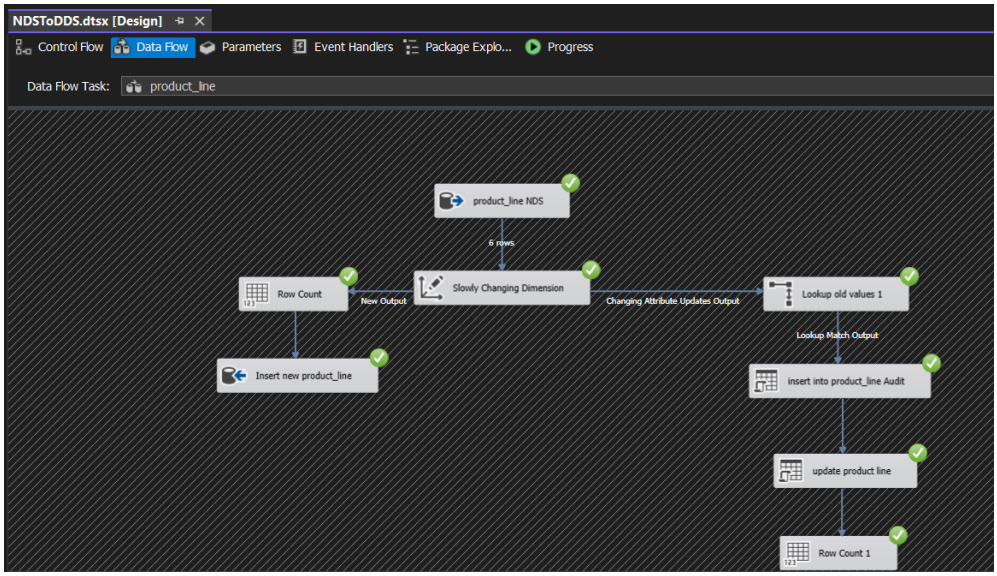


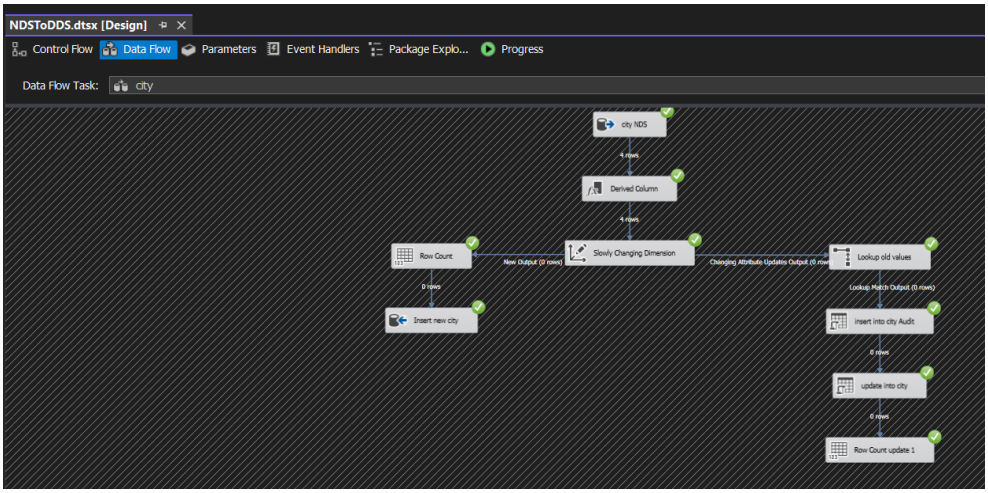
- Data Fire

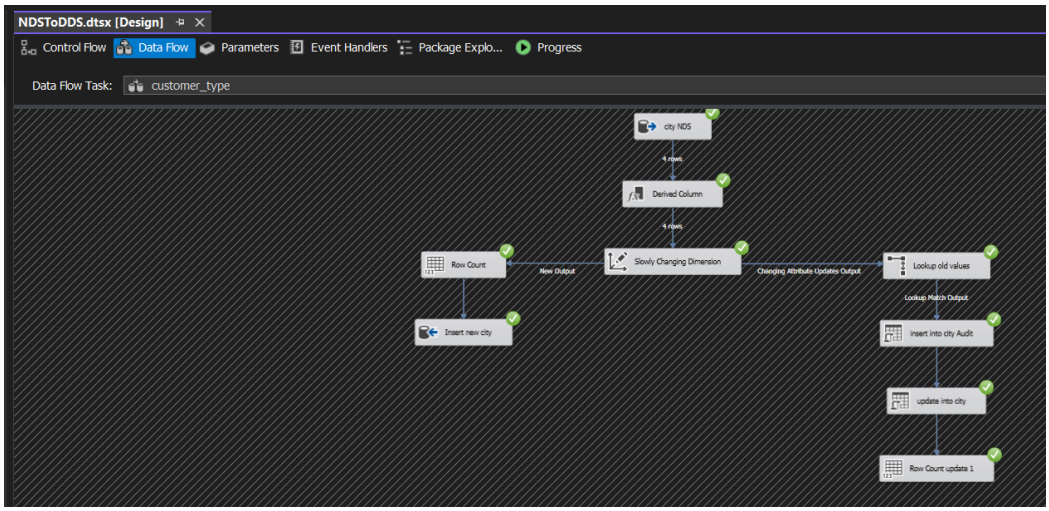
- NDSToDDS

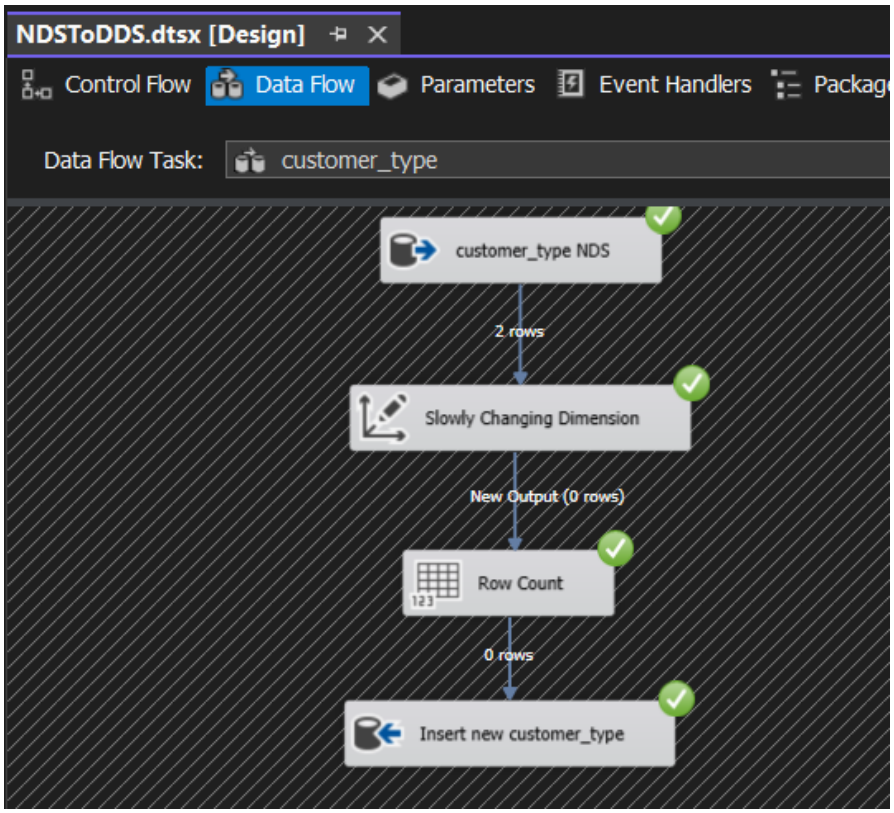


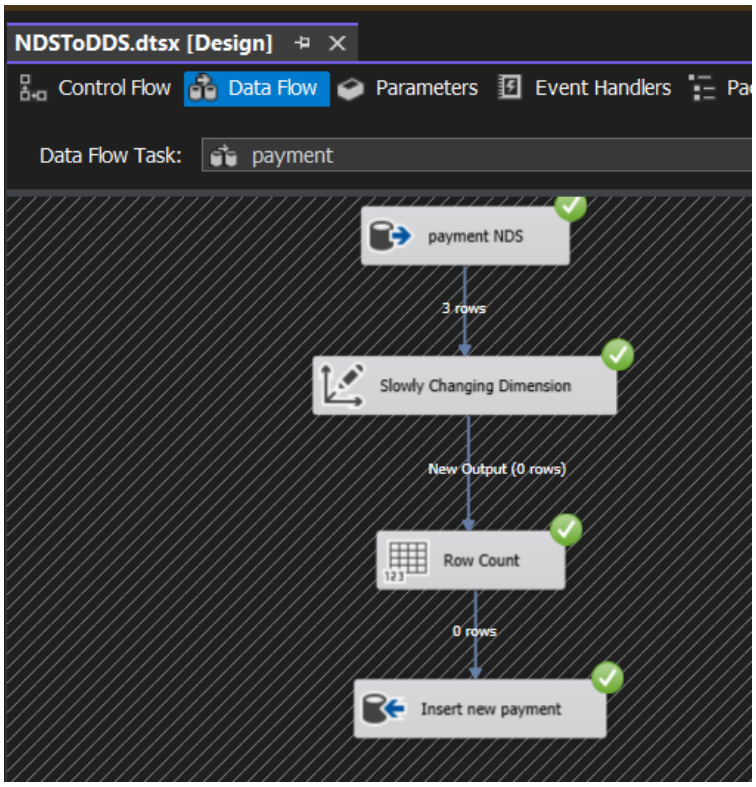
- Chụp chi tiết 1 flow nạp bảng chiều & fact

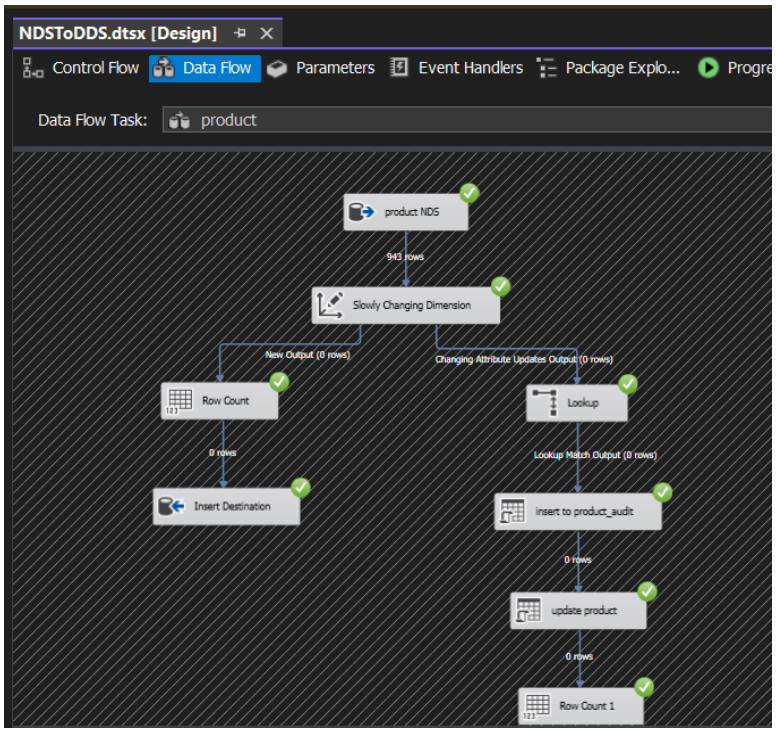

NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
product_line
product_line NDS
6 rows
Row Count
New Output
Slowly Changing Dimension
Changing Attribute Updates Output
Lookup old values 1
Lookup Match Output
Insert new product_line
insert into product_line Audit
update product line
Row Count 1

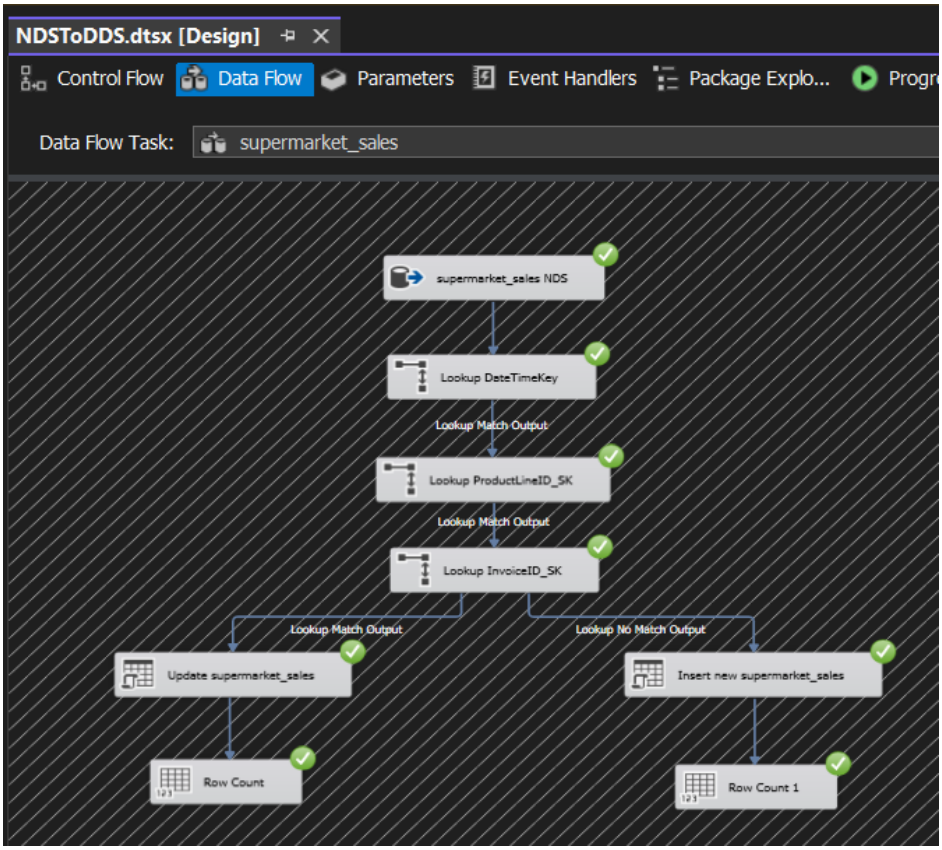

NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
city
city NDS
4 rows
Derived Column
4 rows
Row Count
New Output (0 rows)
Slowly Changing Dimension
Changing Attribute Updates Output (0 rows)
Lookup old values
0 rows
Lookup Match Output (0 rows)
Insert new city
insert into city Audit
0 rows
update into city
6 rows
Row Count update 1


NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
customer_type
city NDS
4 rows
Derived Column
4 rows
Row Count
New Output
Slowly Changing Dimension
Changing Attribute Updates Output
Lookup old values
Lookup Match Output
Insert new city
insert into city Audit
update into city
Row Count update 1


NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Data Flow Task:
customer_type
customer_type NDS
2 rows
Slowly Changing Dimension
New Output (0 rows)
Row Count
0 rows
Insert new customer_type


NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Data Flow Task:
payment
payment NDS
3 rows
Slowly Changing Dimension
New Output (0 rows)
Row Count
0 rows
Insert new payment

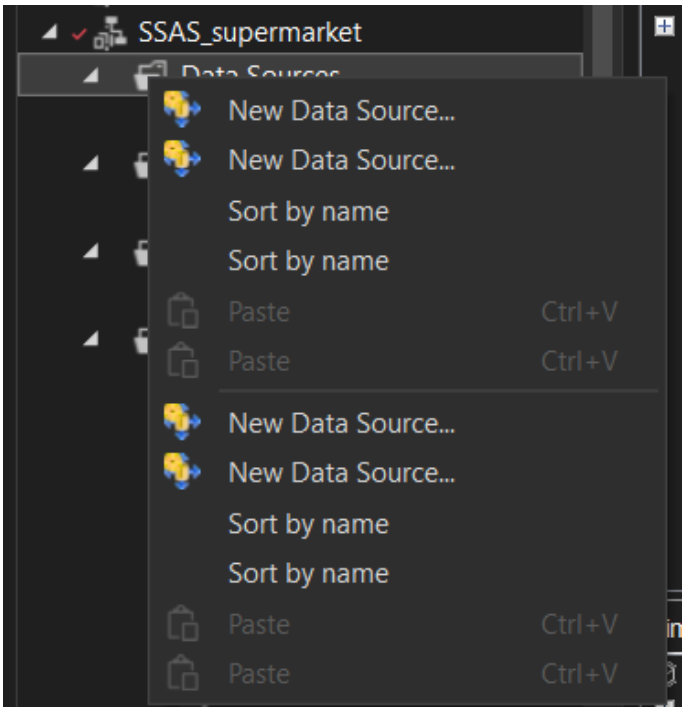

NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progre
Data Flow Task:
product
product NDS
943 rows
Slowly Changing Dimension
New Output (0 rows)
Changing Attribute Updates Output (0 rows)
Row Count
Lookup
0 rows
Lookup Match Output (0 rows)
Insert Destination
insert to product_audit
0 rows
update product
0 rows
Row Count 1


NDSToDDS.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progr
Data Flow Task:
supermarket_sales
supermarket_sales NDS
Lookup DateTimeKey
Lookup Match Output
Lookup ProductLineID_SK
Lookup Match Output
Lookup InvoiceID_SK
Lookup Match Output
Lookup No Match Output
Update supermarket_sales
Insert new supermarket_sales
Row Count
Row Count 1

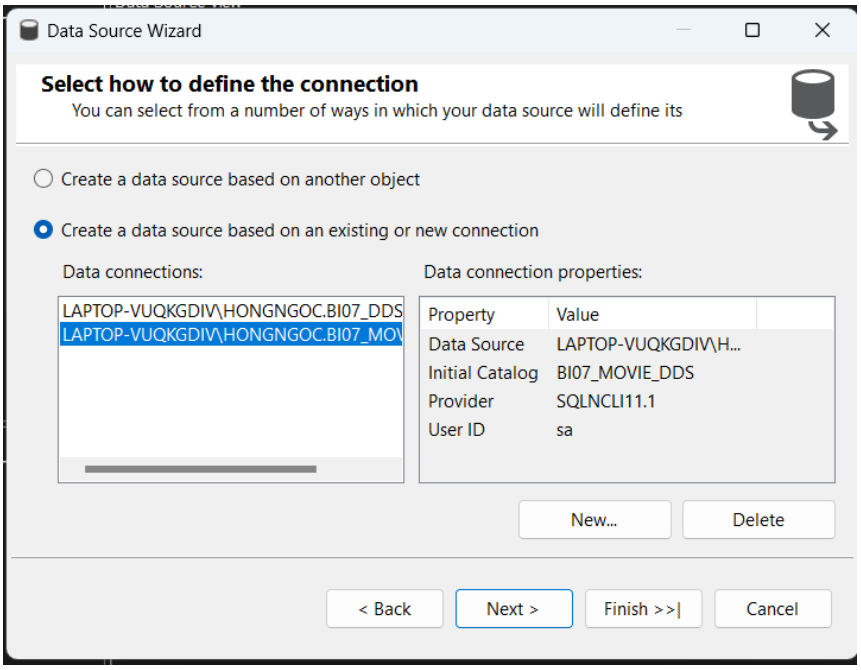
II. Analysis Service

1. Cài đặt server:
2. Tạo mới Data Source:

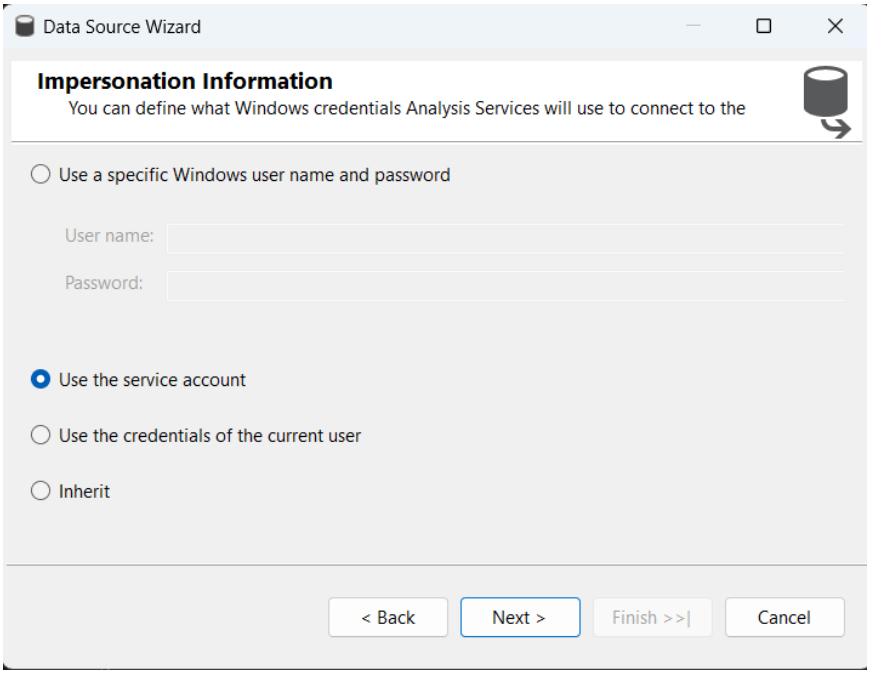
- Click phải và chọn “New Data Source”



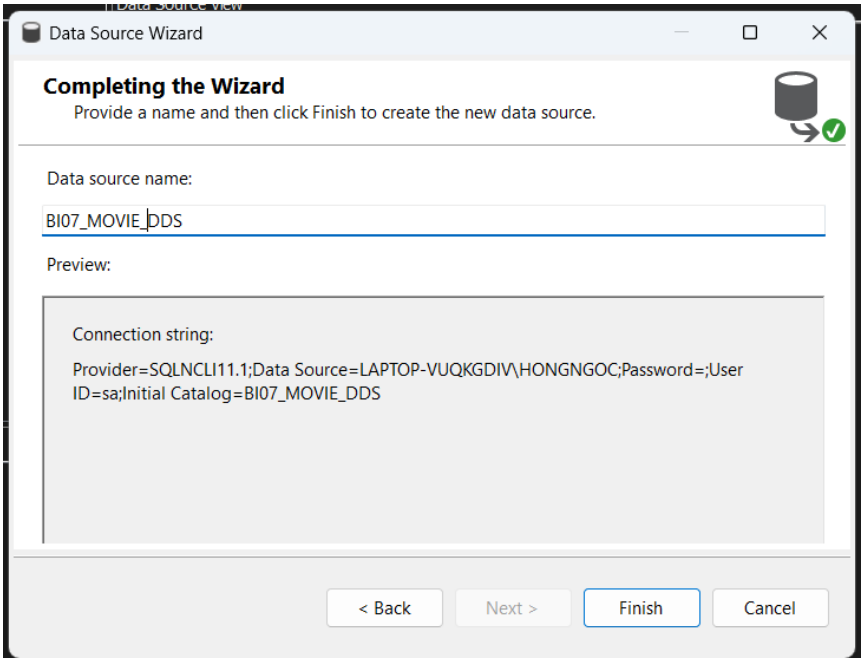
- Sau đó chọn tới chỗ connect lưu dữ liệu trong SSMS



- Tại đây nếu ta không muốn nhập username và password của máy thì có thể chọn mục thứ 2 thì sẽ sử dụng service account

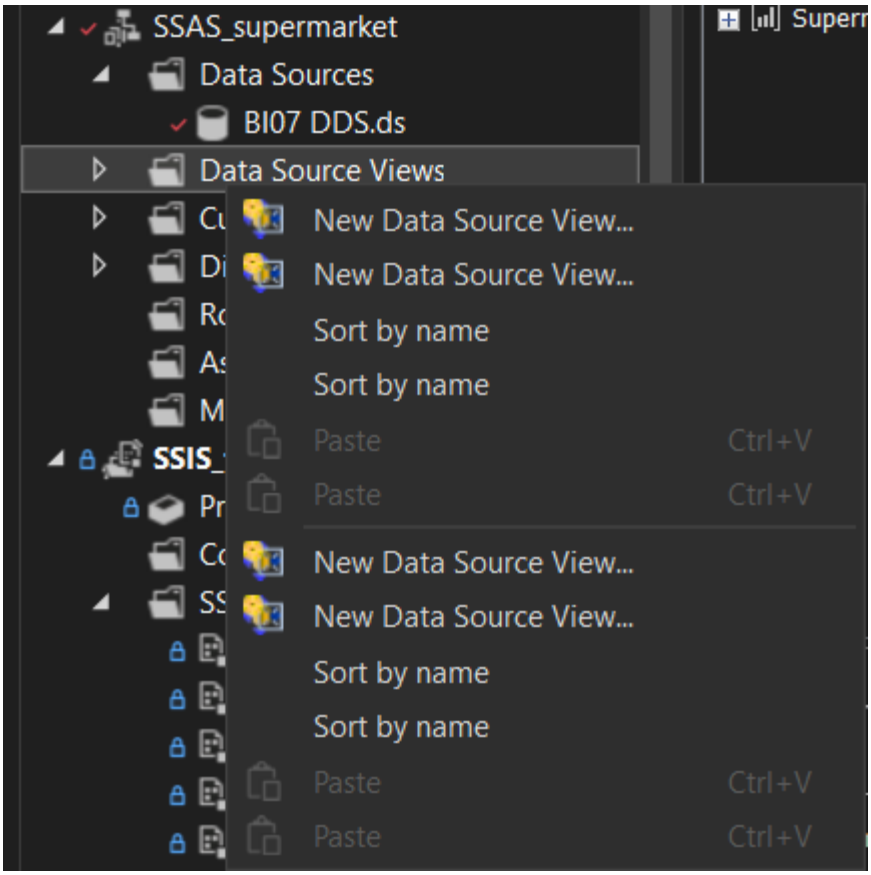


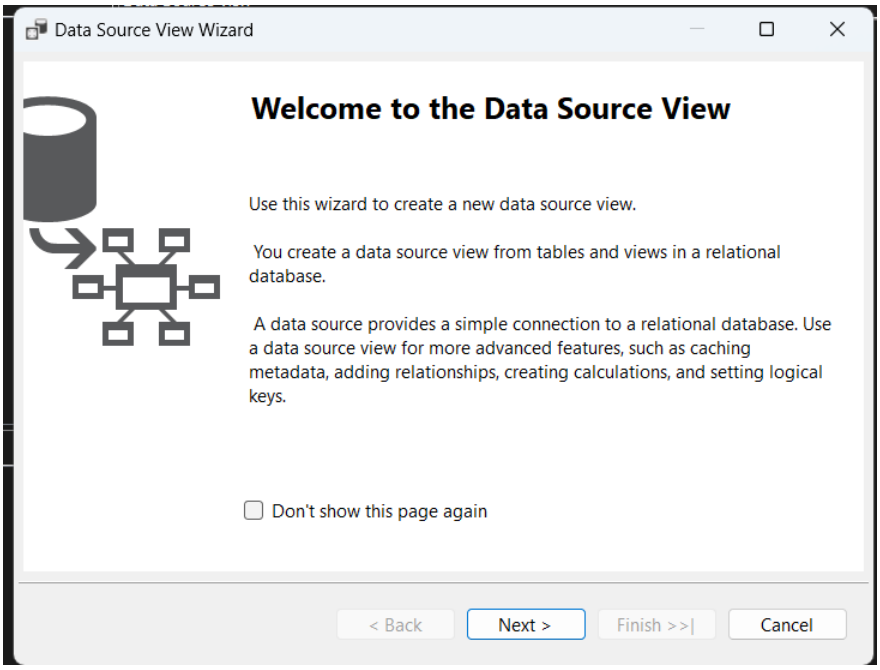
- Và đặt tên cho data source



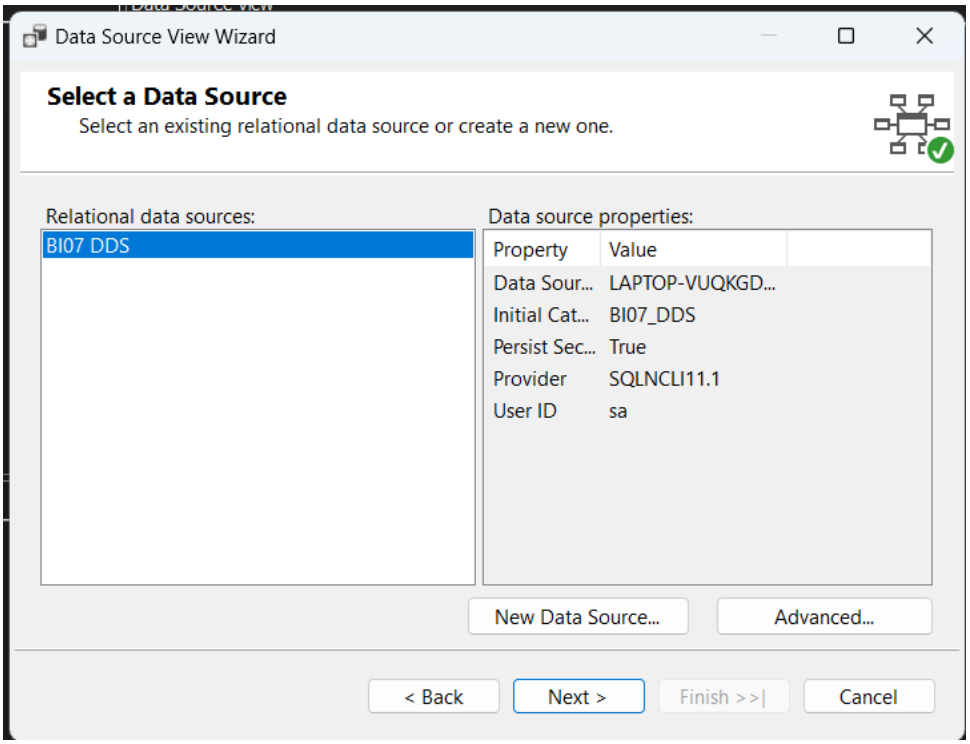
3. Tạo mới Data Source Views

- Click phải và chọn “New Data Source View”

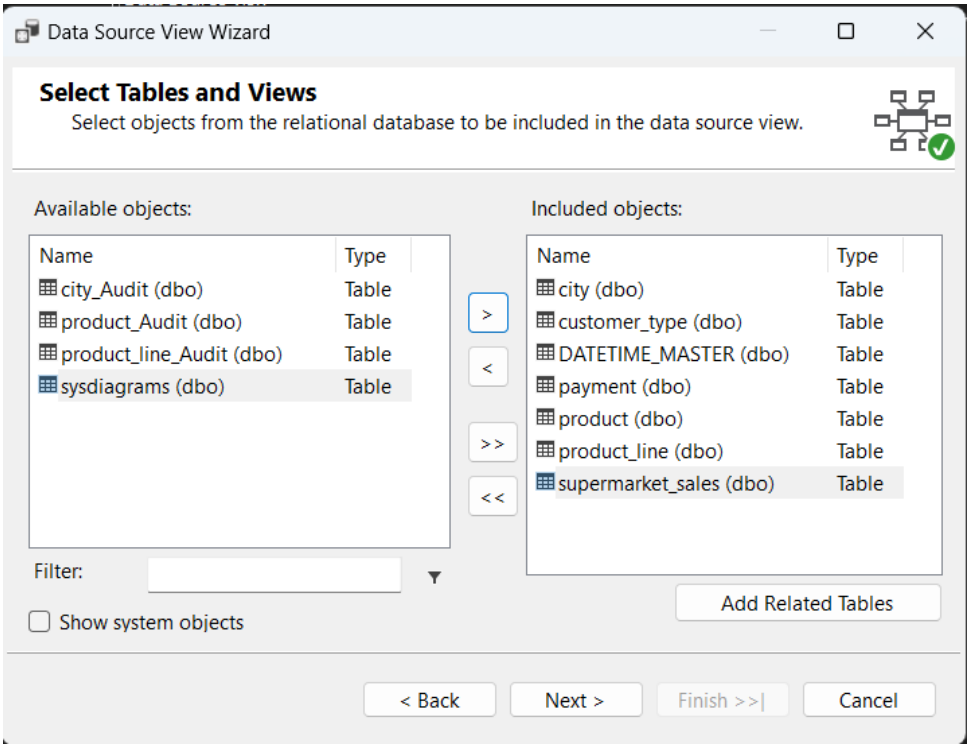




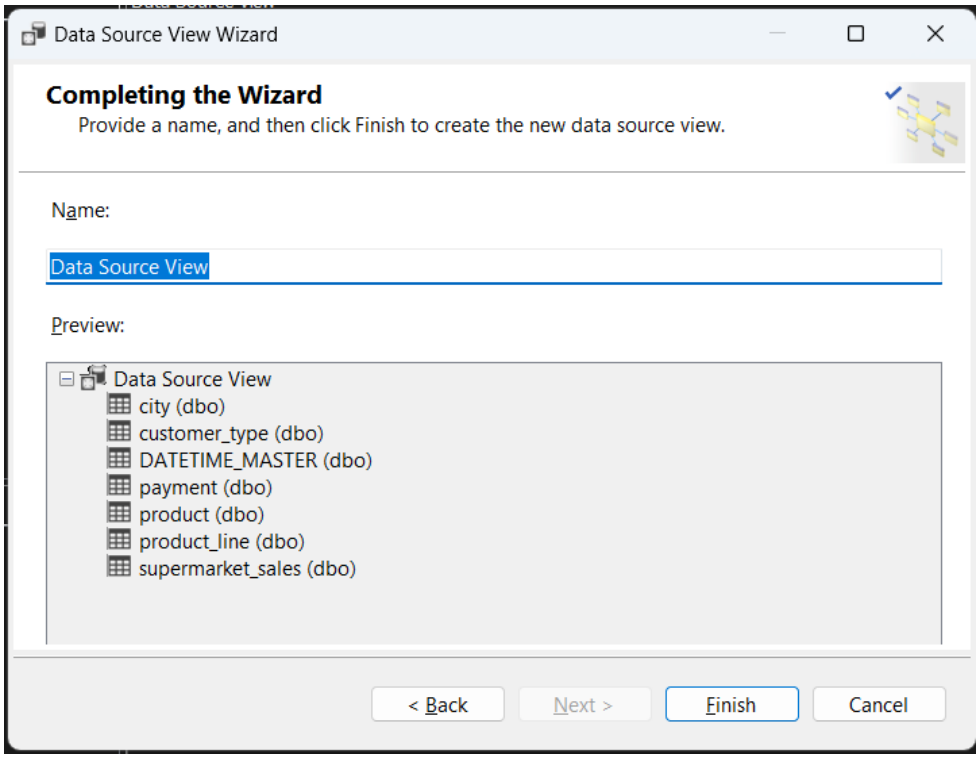
- Sau đó chọn tới chỗ connect lưu dữ liệu trong SSMS



- Chọn các bảng liên quan

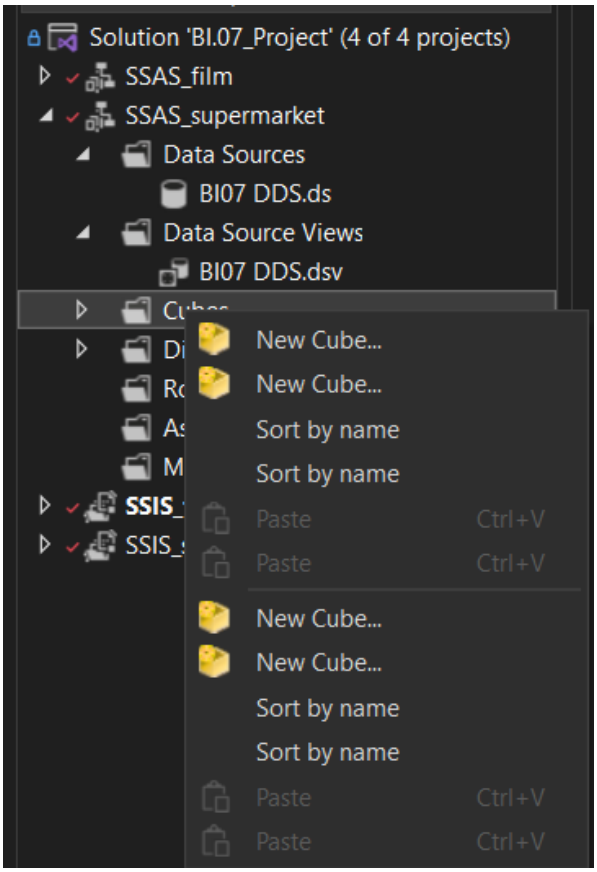


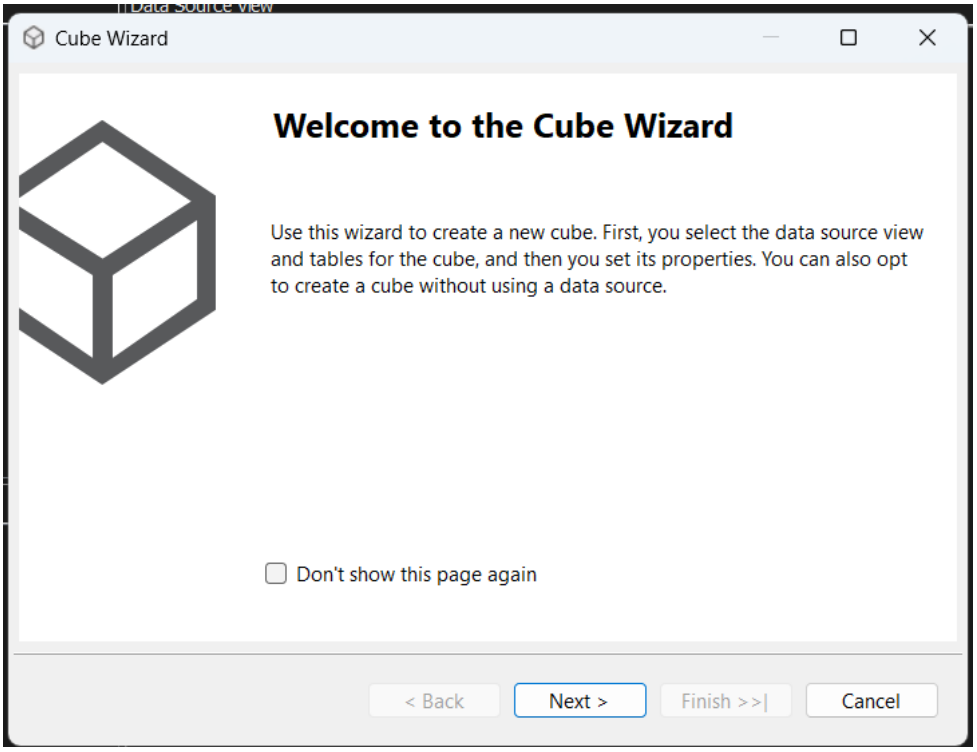
- Đặt tên cho data source view

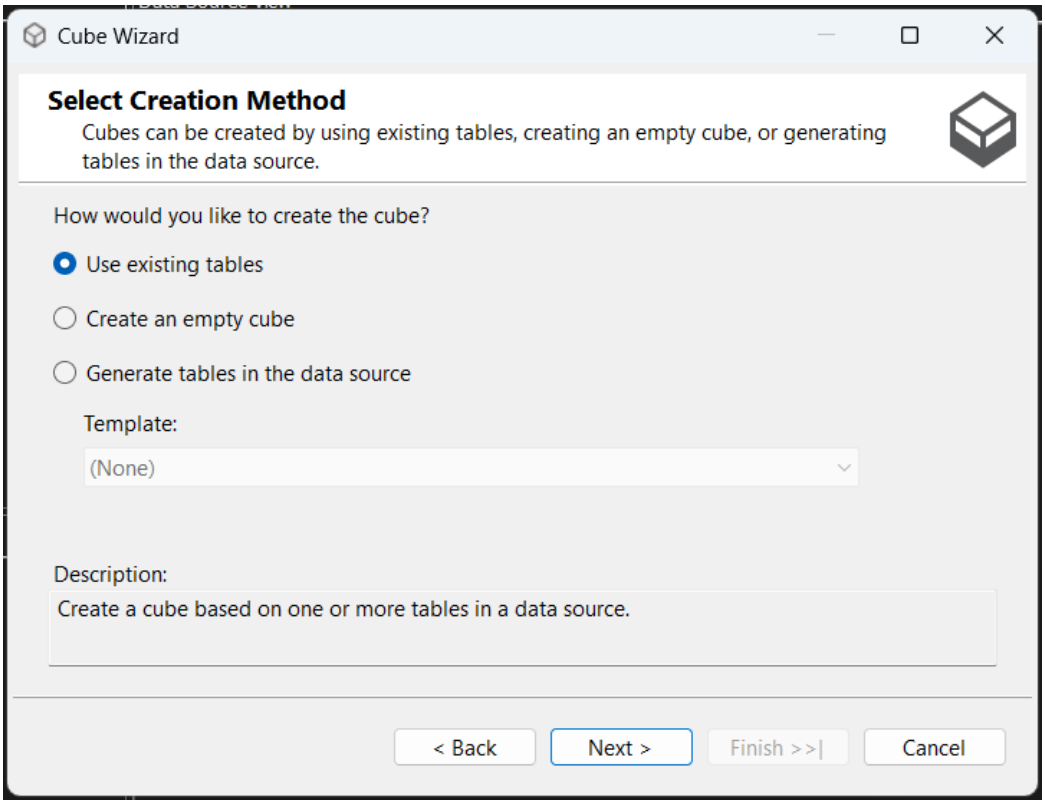


4. Tạo Cube

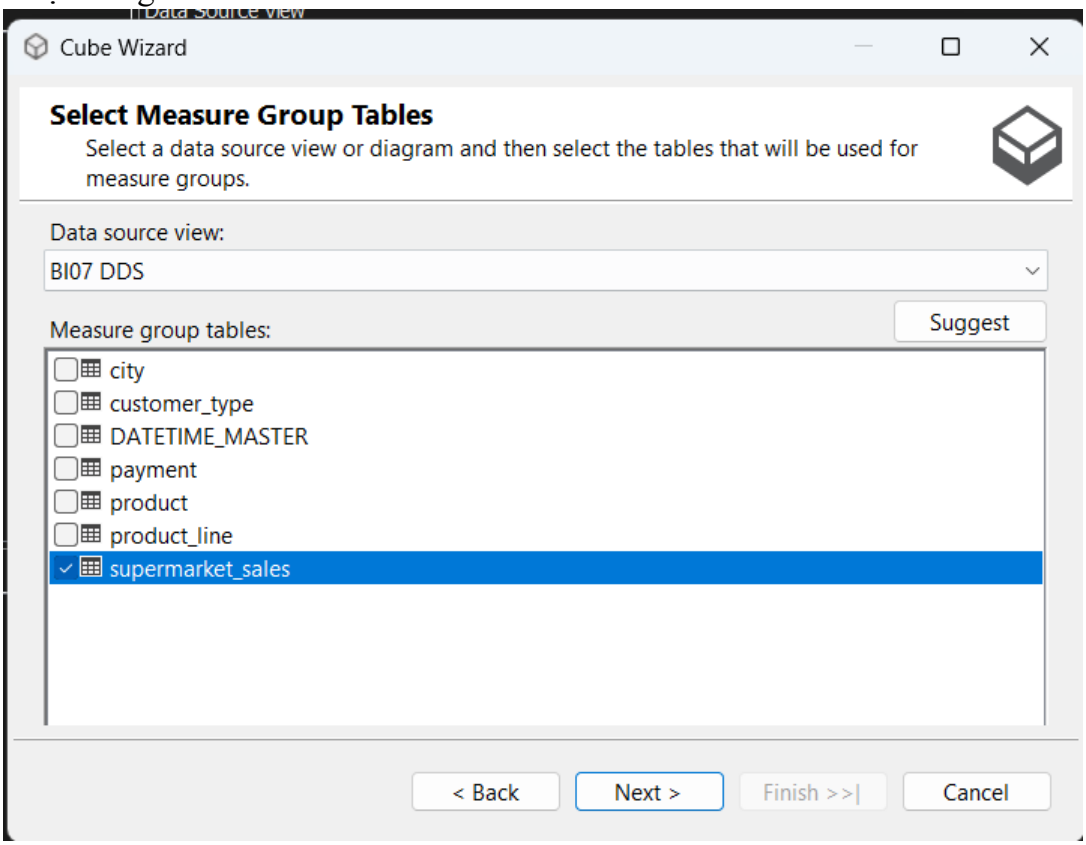
- Click phải và chọn “New Cube”


Solution 'BI.07_Project' (4 of 4 projects)
SSAS_film
SSAS_supermarket
Data Sources
BI07 DDS.ds
Data Source Views
BI07 DDS.dsv
New Cube...
New Cube...
Sort by name
Sort by name
Paste Ctrl+V
Paste Ctrl+V
New Cube...
New Cube...
Sort by name
Sort by name
Paste Ctrl+V
Paste Ctrl+V

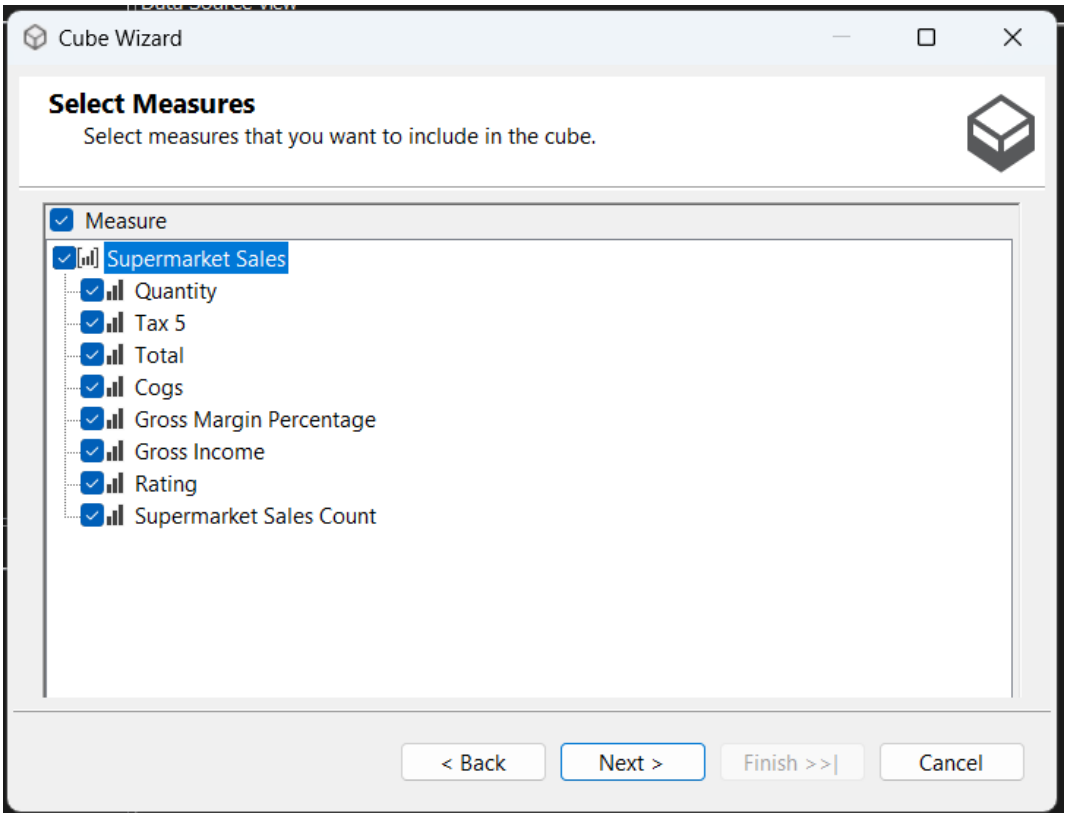

Cube Wizard
Welcome to the Cube Wizard
Use this wizard to create a new cube. First, you select the data source view and tables for the cube, and then you set its properties. You can also opt to create a cube without using a data source.
Don't show this page again
< Back
Next >
Finish >>|
Cancel

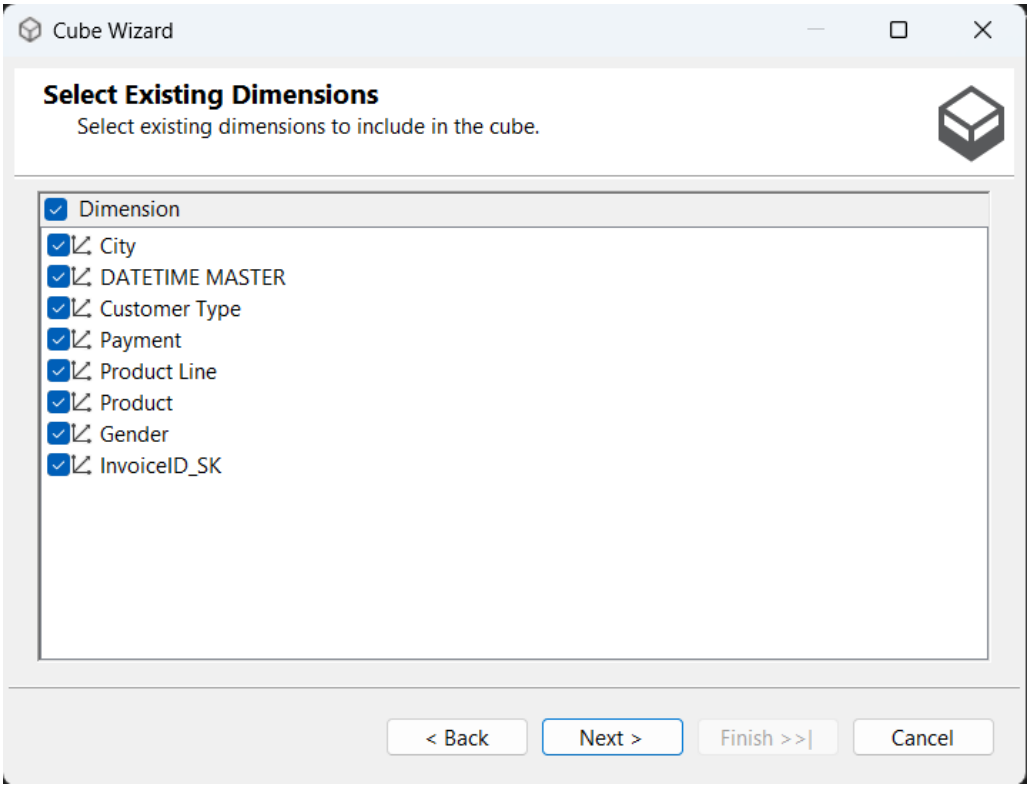


- Chọn bảng fact

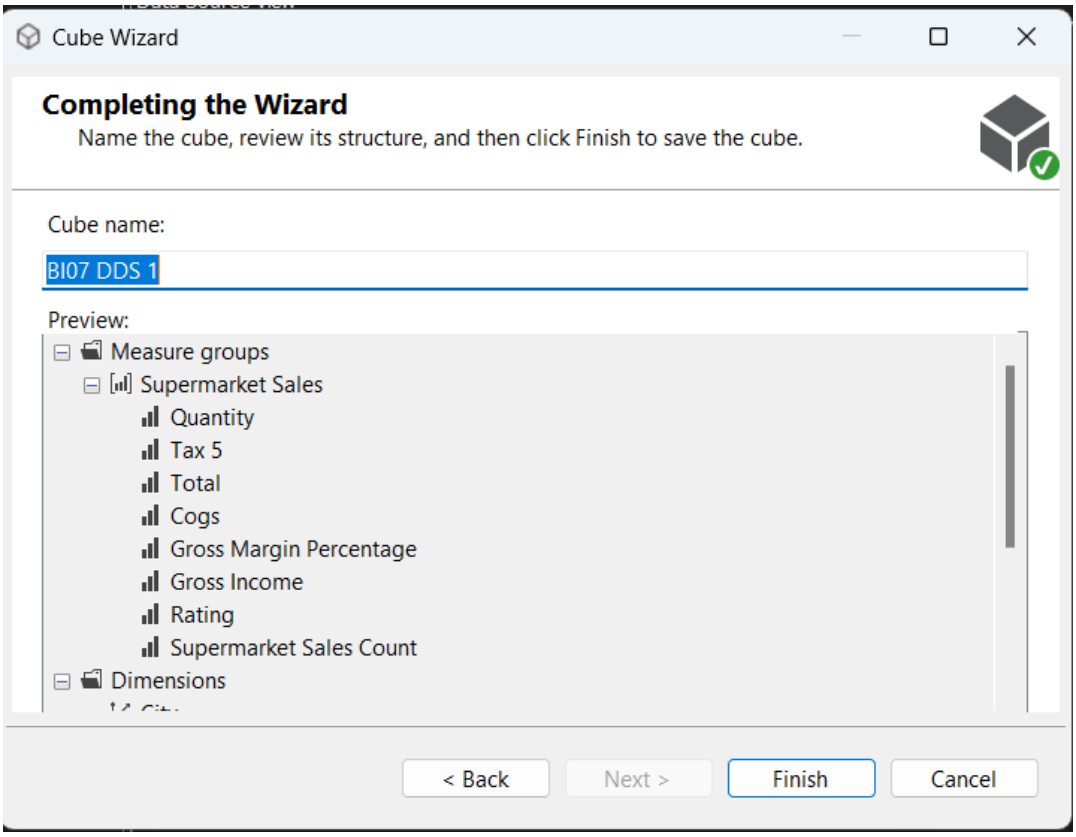


- Chọn các measure và bảng dim



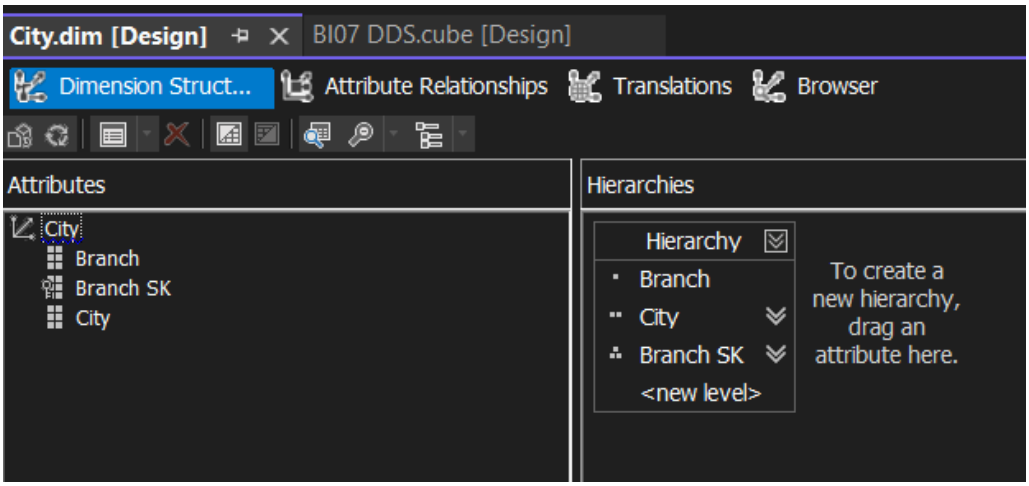


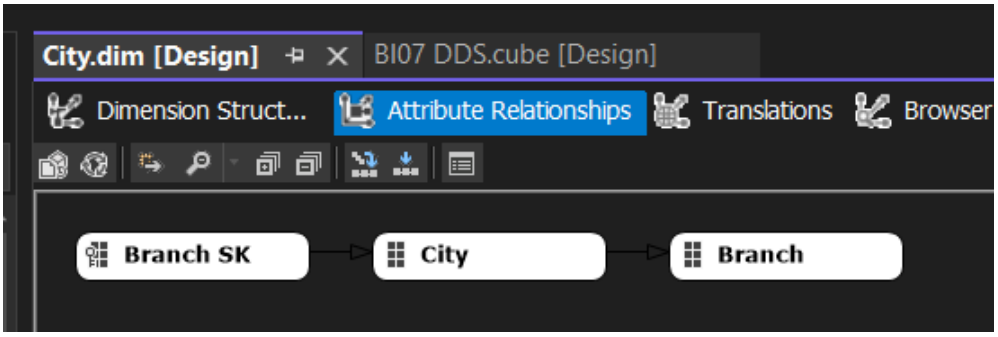
- Đặt tên cho cube



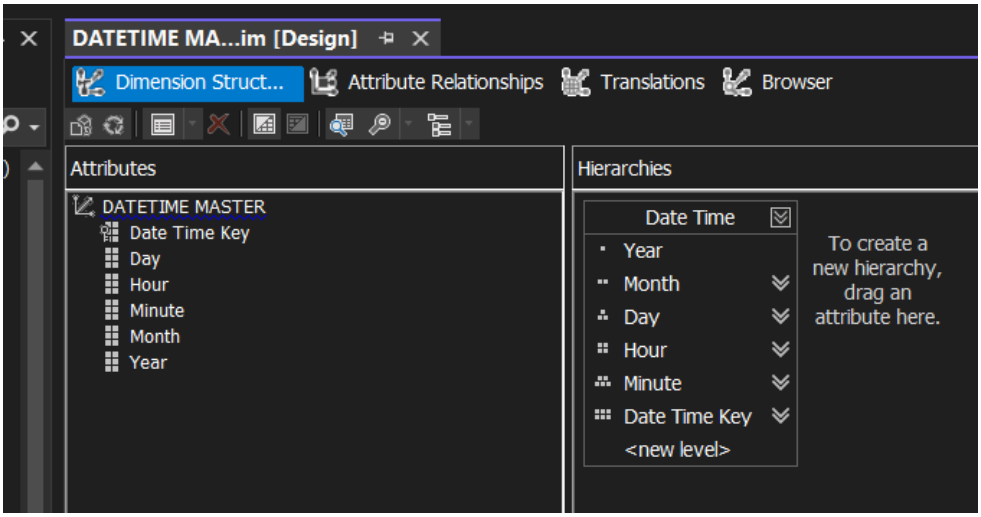
5. Phân chiều cho các bảng Dim

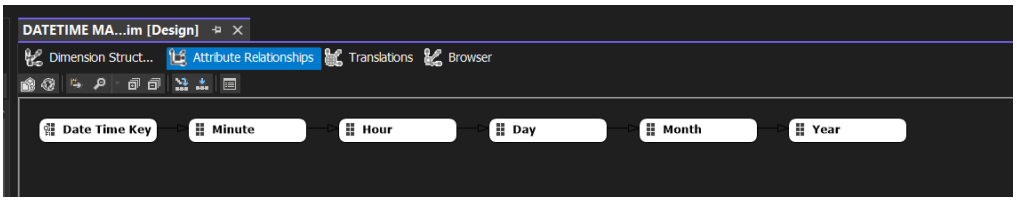
- city



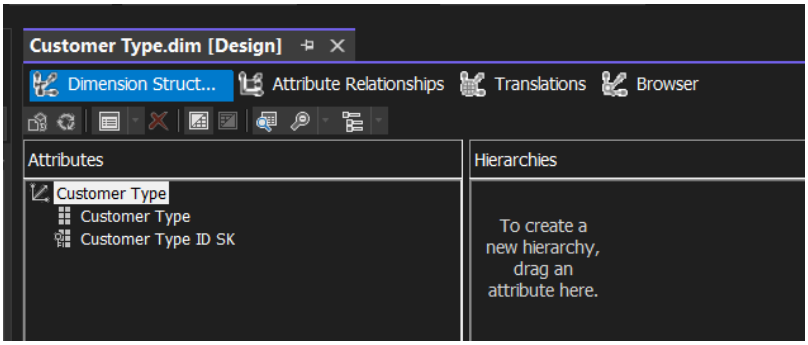


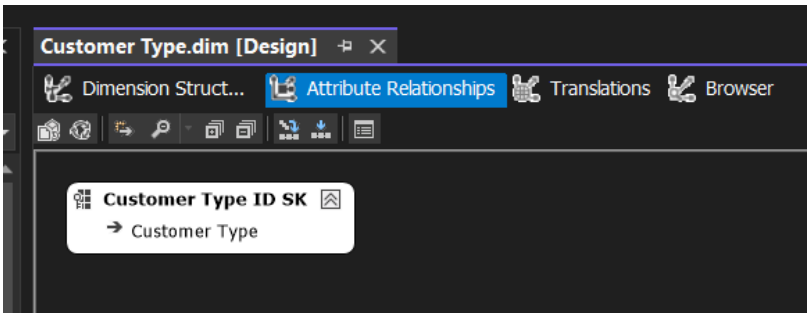
- datetime



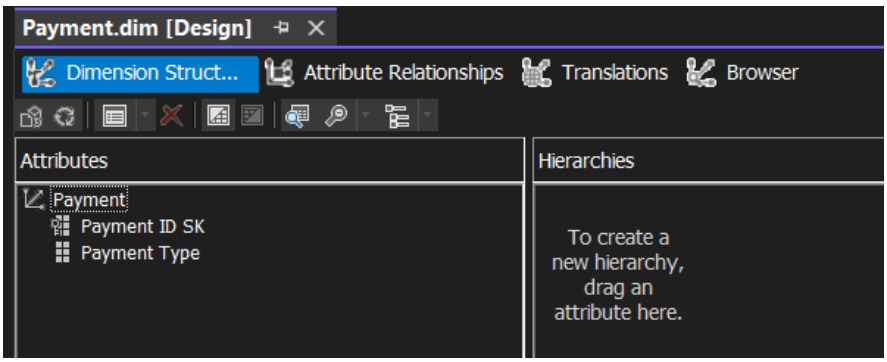


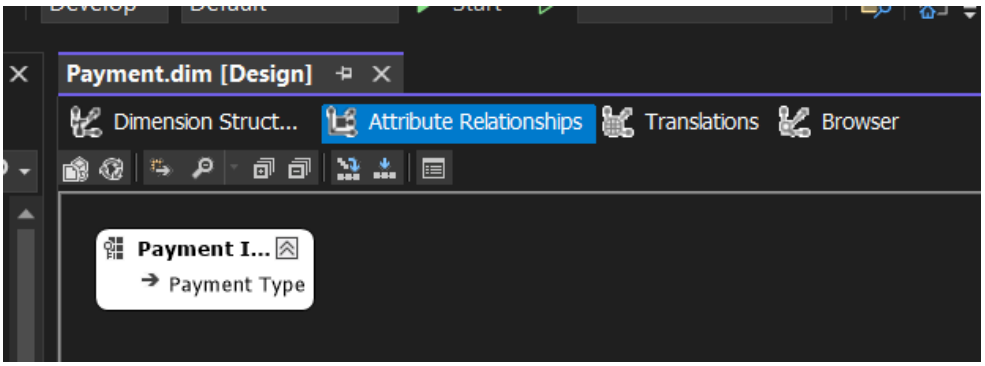
- customer type



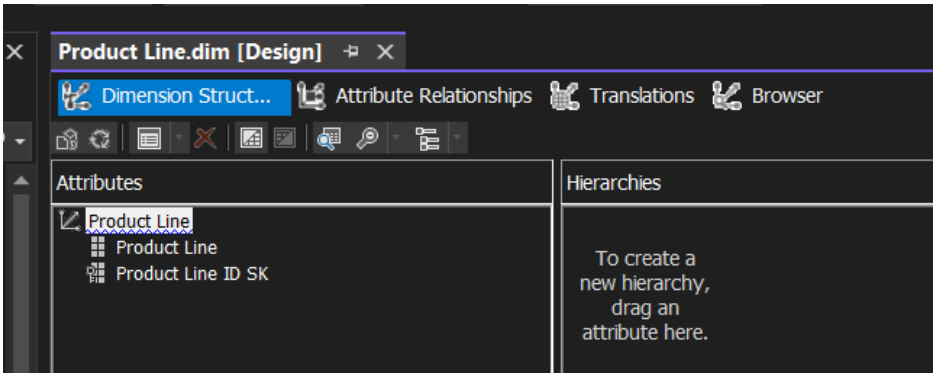


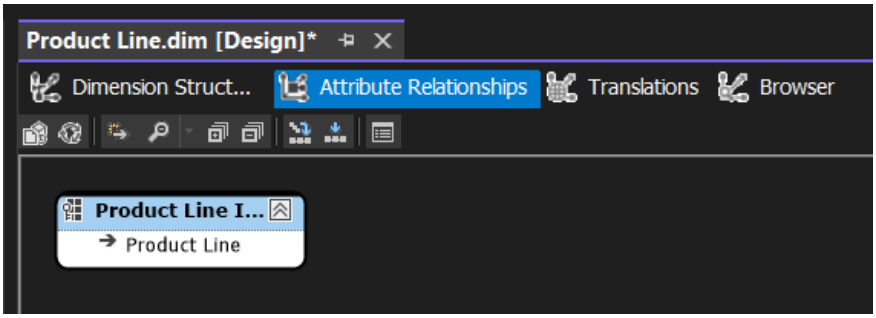
- payment



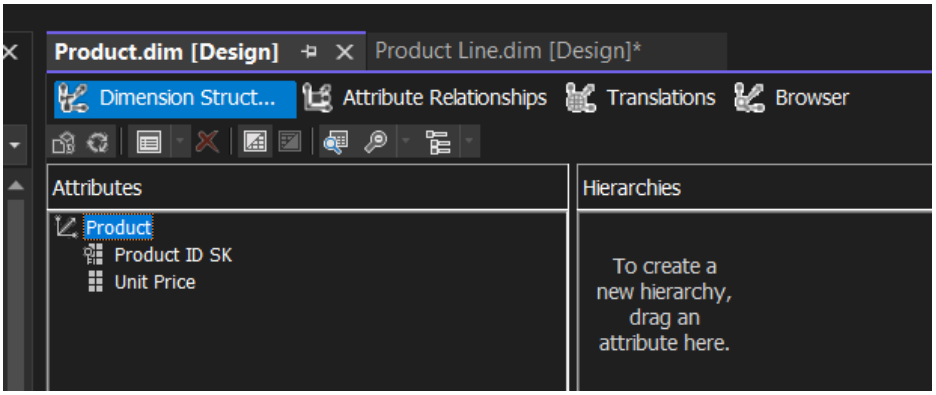


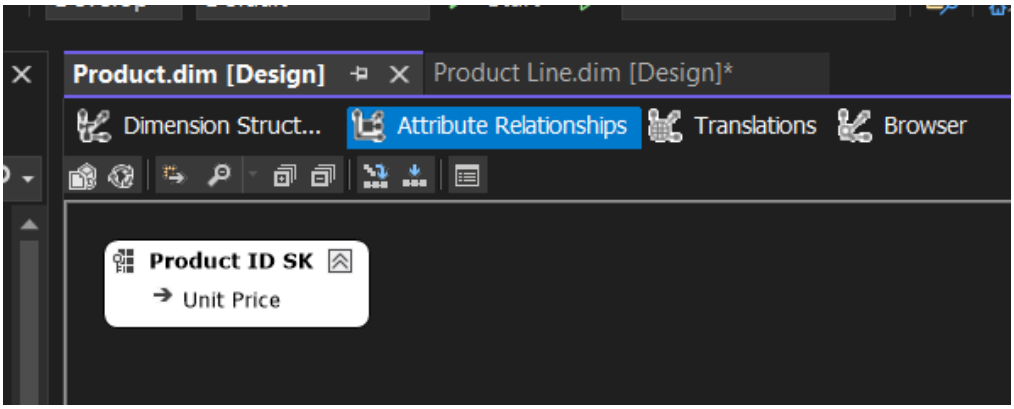
- product line



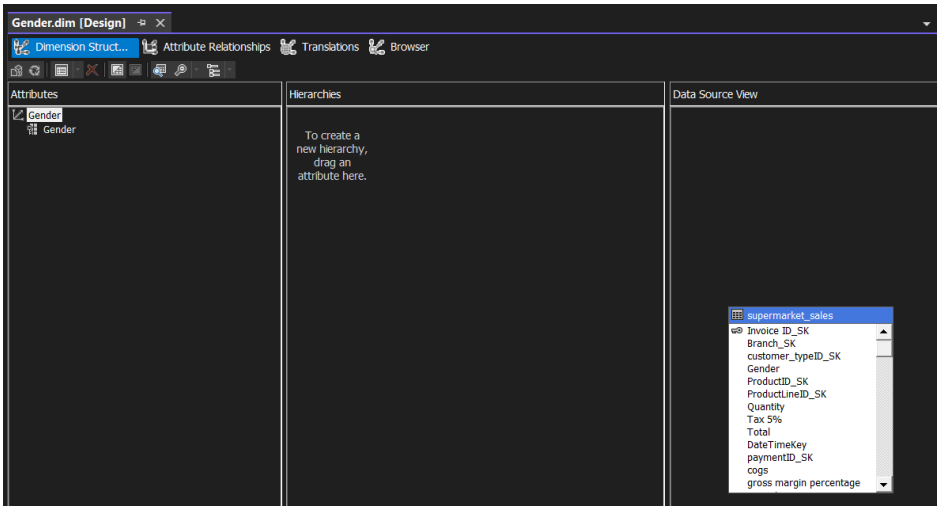


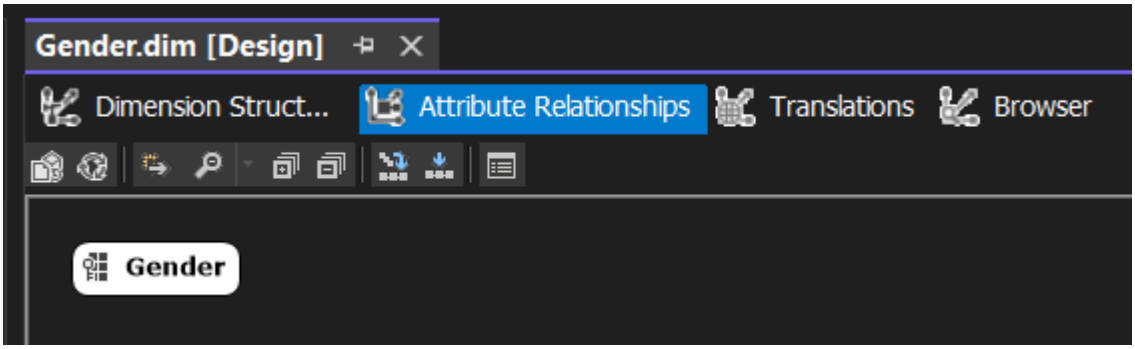
- product



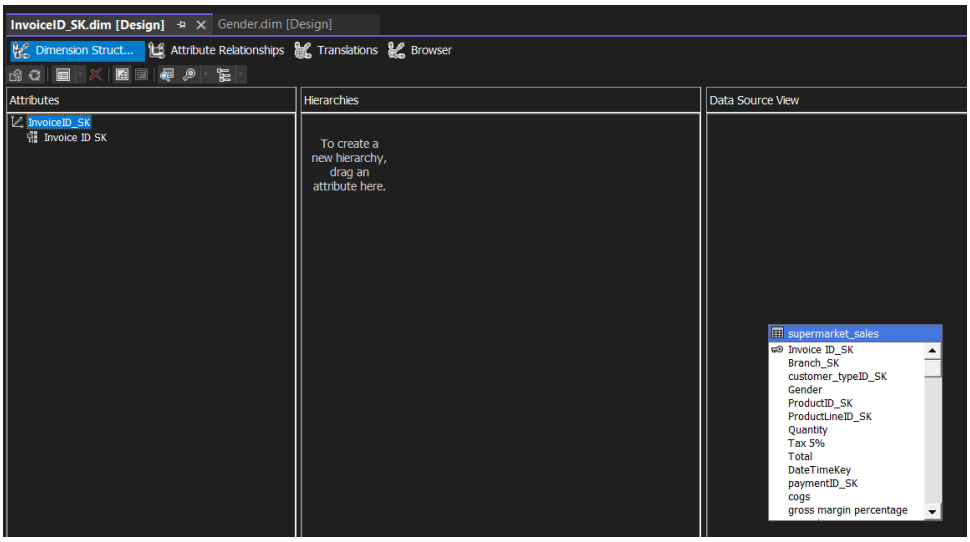


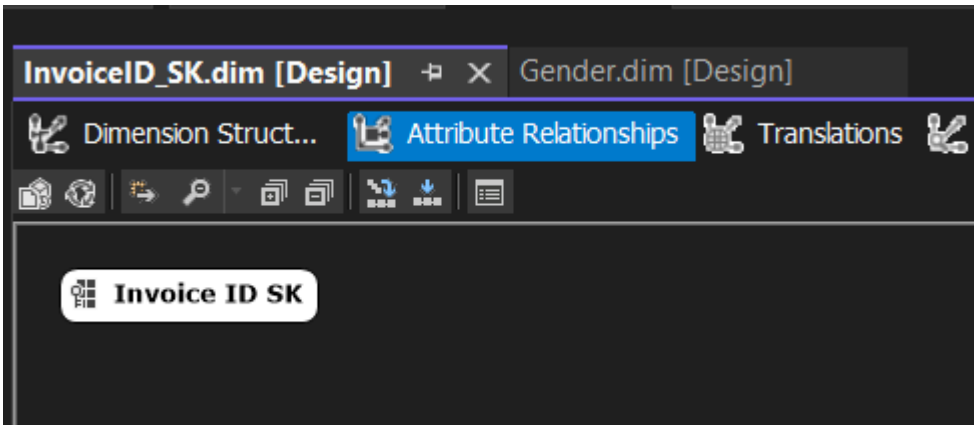
- gender (lấy thuộc tính gender trong bảng supermarket_sales để tạo thành dim)





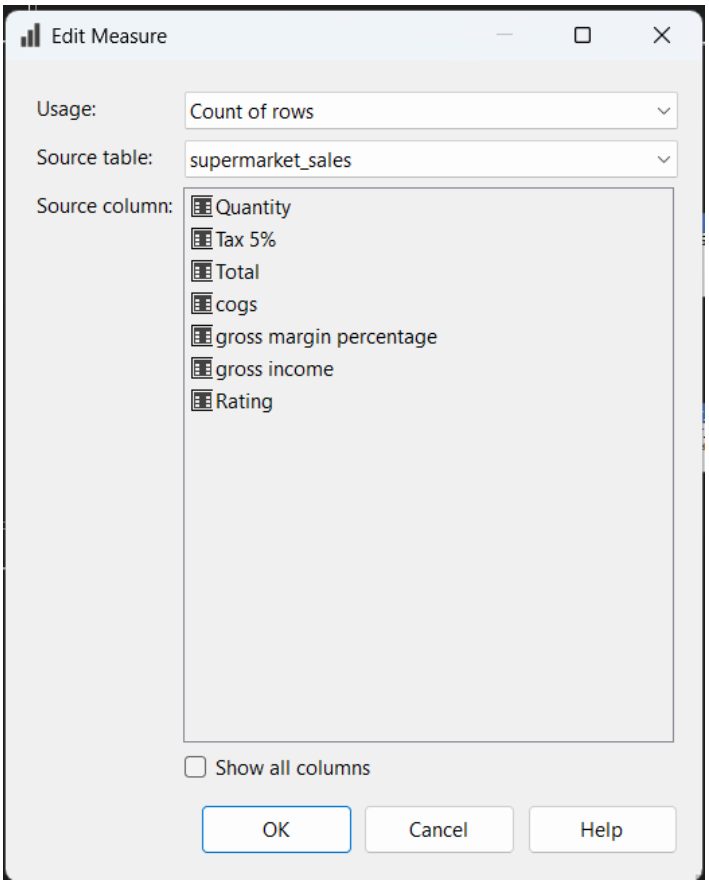
- invoiceID_SK (Lấy thuộc tính invoiceID_SK trong bảng supermarket_sales tạo thành dim)



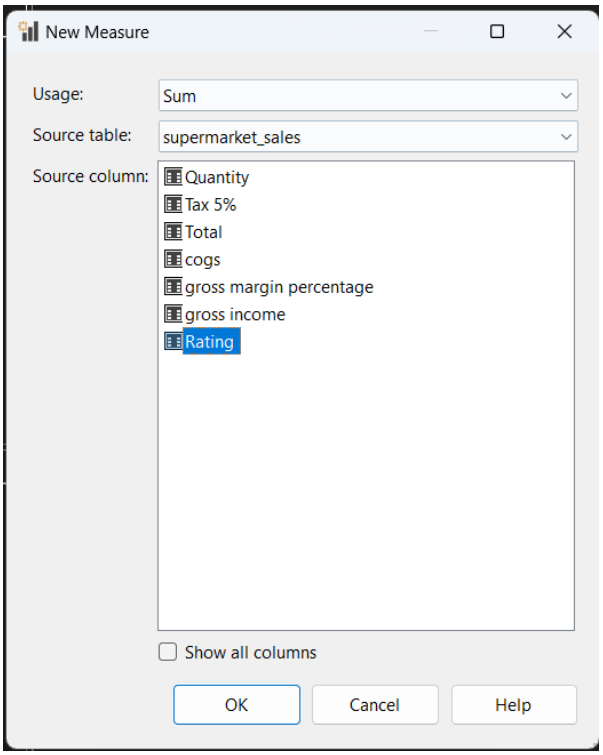


6. Tạo measure:

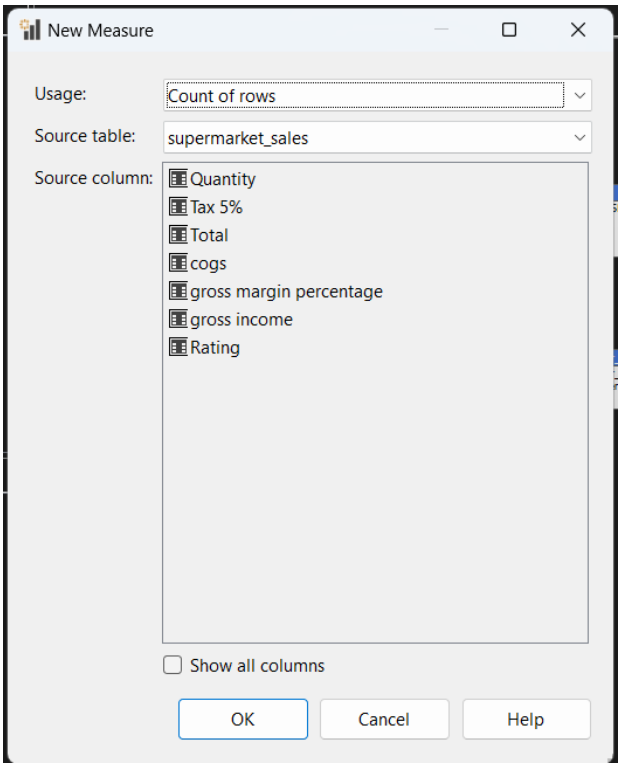
- Invoice ID SK_Count, Supermarket Sale Count



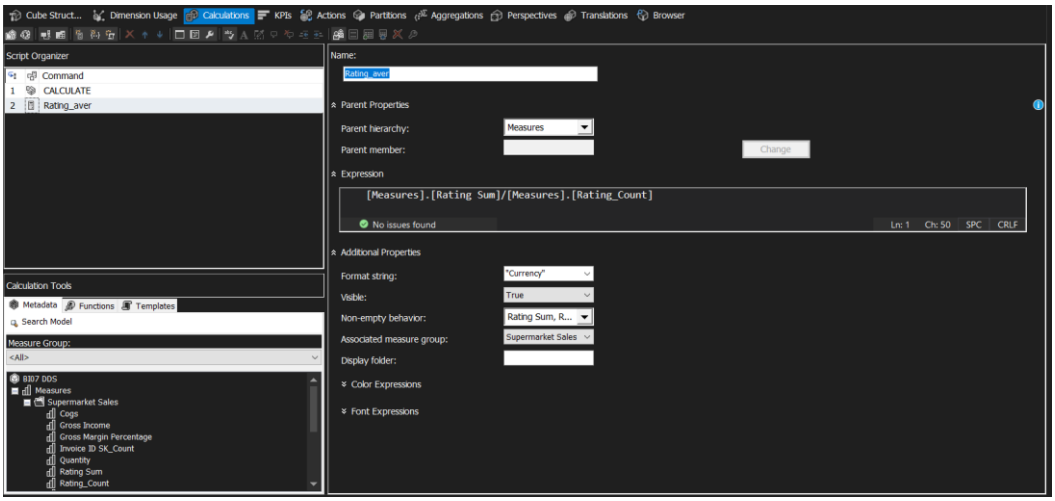
- Rating Sum

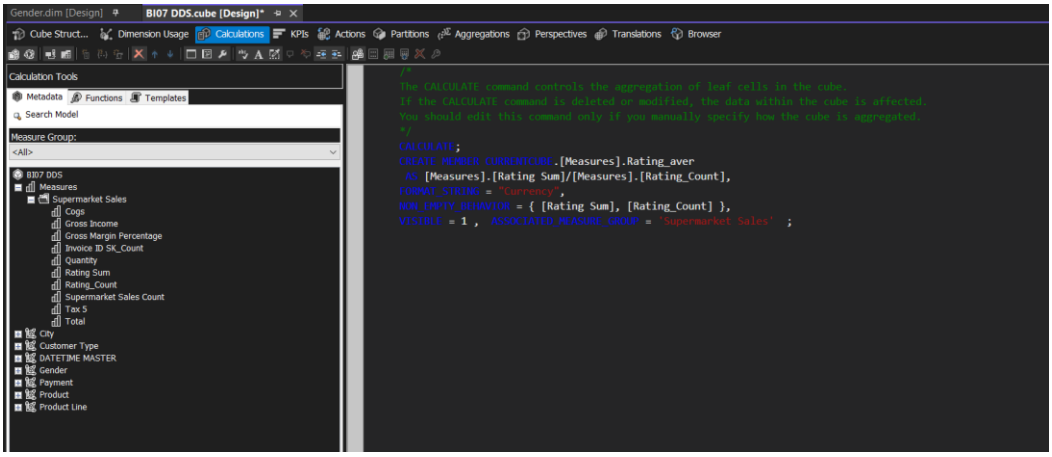


- Rating_Count



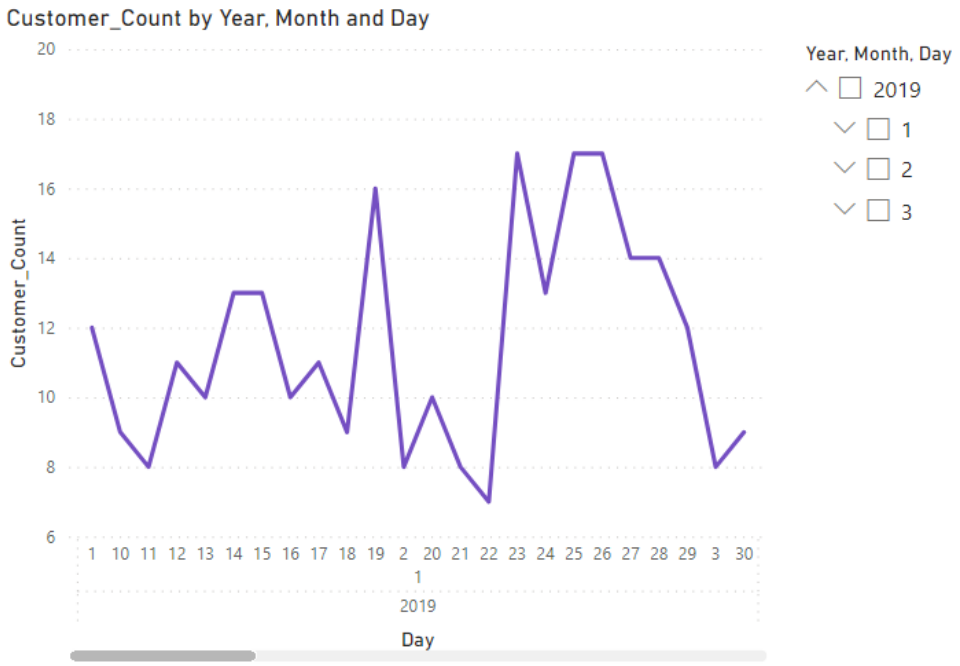
- Rating_aver





III. Report

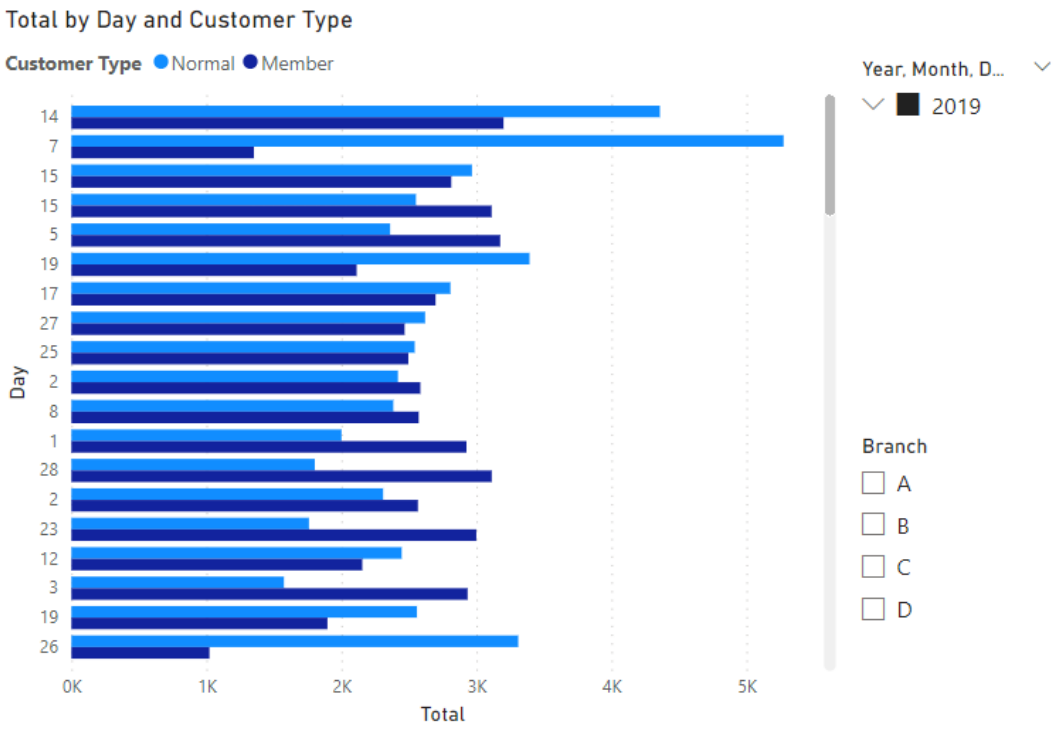
1. Thống kê số thành viên mua hàng theo ngày, tháng, năm



Loại biểu đồ: Line Chart

- Trục X: Day (DATE MASTER - Date time - Day)
- Trục Y: Số lượt khách hàng (Invoice ID SK - supermarket_sales measure)
- Slicer field: Day (DATE MASTER - Date time - Day)

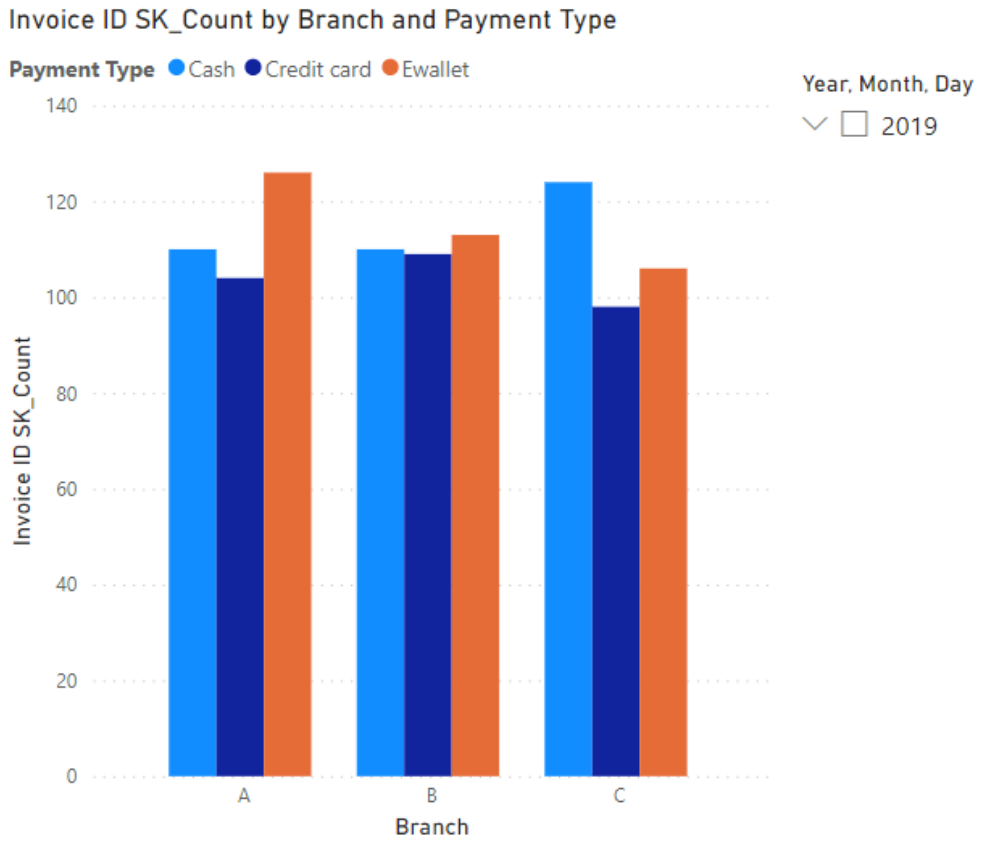
2. Thống kê doanh thu của khách hàng (member, normal) theo ngày, tháng, năm và theo chi nhánh.



Loại biểu đồ: Clustered Bar Chart

- Trục X: Day (DATE MASTER - Date time - Day)
- Trục Y: Số doanh thu (Total - supermarket_sales measure)
- Slicer field: Day (DATE MASTER - Date time - Day); Branch (City - Hierarchy - Branch)

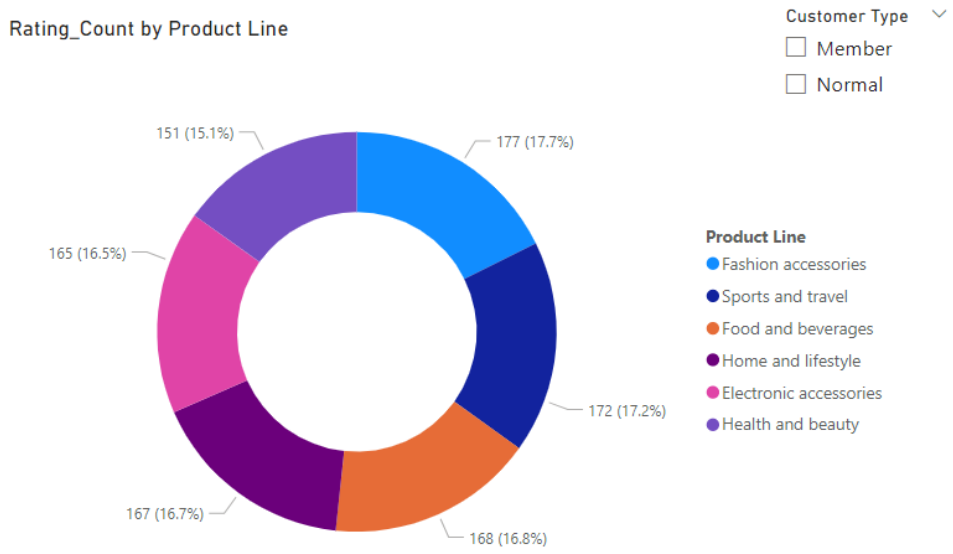
3. Thống kê số lượng khách thanh toán theo cash/ debit/… ở từng chi nhánh theo từng tháng trong năm



Loại biểu đồ: Clustered Column Chart

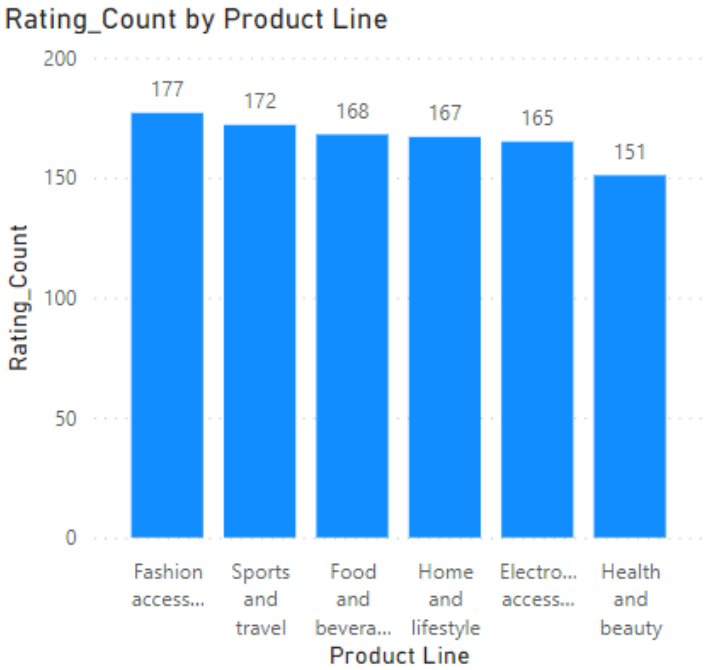
- Trục X: Branch (City - Hierarchy - Branch)
- Trục Y: Số lượt khách hàng (Invoice ID SK - supermarket_sales measure)
- Legend: Tên phương thức thanh toán (Payment Type -Payment dimension)
- Thực hiện filter các table theo từng Legend
- Slicer field: Day (DATE MASTER - Date time - Day)

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



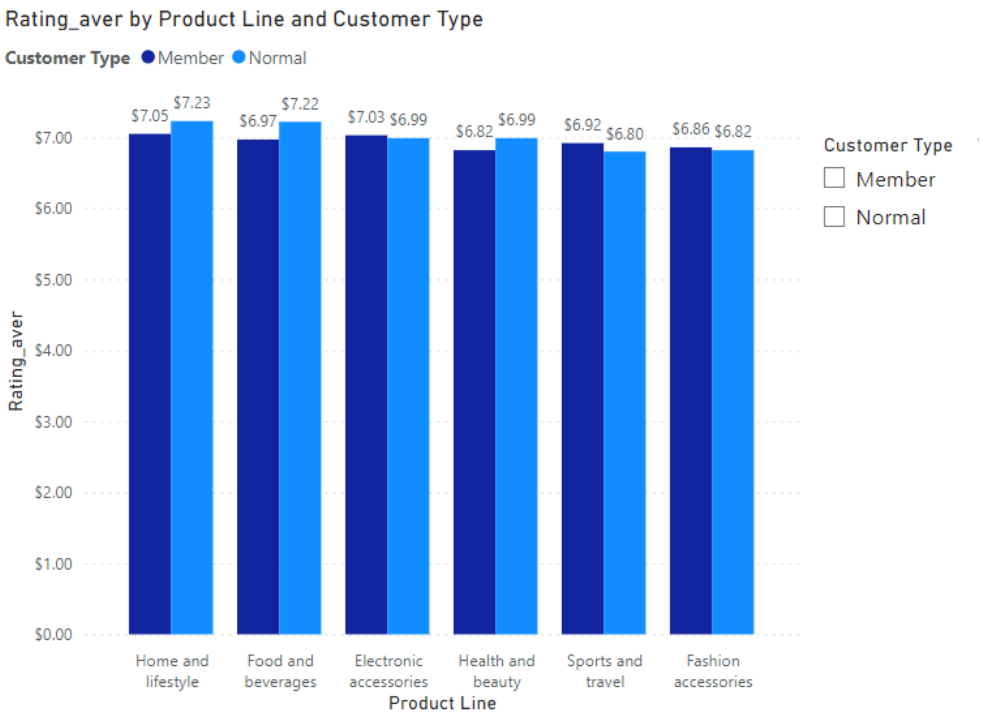
Loại biểu đồ: Donut Chart

- Legend: ProductLine (ProductLine - Product Line dimension)
- Values: Số lượng rating của khách hàng (Rating_Count - supermarket_sales measure)
- Slicer field: Customer Type - Customer Type dimension



Loại biểu đồ: Stacked Column Chart

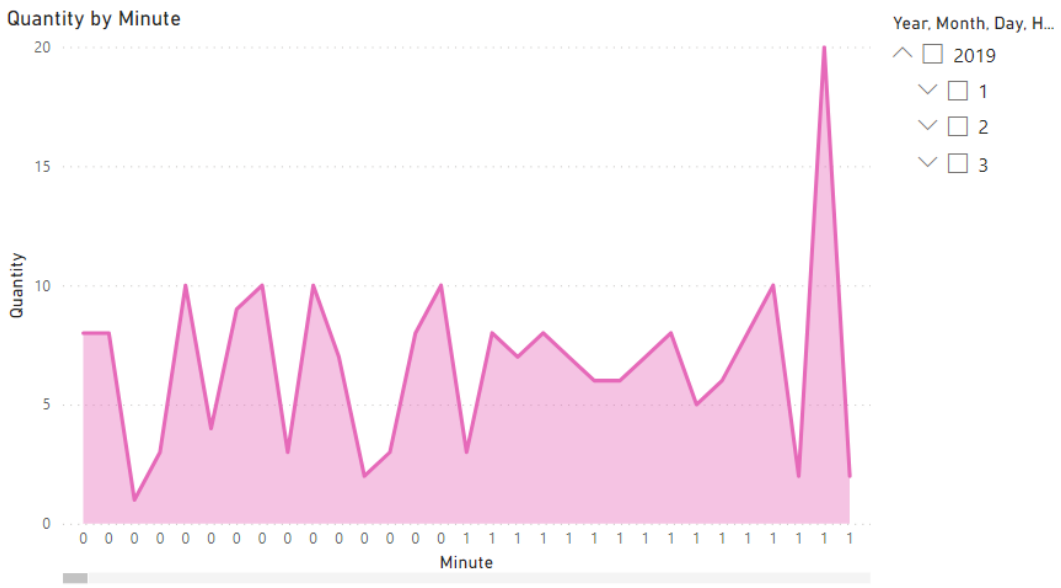
- Trục X: ProductLine (ProductLine - Product Line dimension)
- Trục Y: Số lượng rating (Rating_Count - supermarket_sales measure)
- Slicer field: Customer Type - Customer Type dimension



Loại biểu đồ: Clustered Column Chart

- Trục X: ProductLine (ProductLine - Product Line dimension)
- Trục Y: Giá trị trung bình rating của khách hàng (Rating_aver - supermarket_sales measure)
- Legend: Tên loại khách hàng (Customer Type - Customer Type dimension)
- Thực hiện filter các table theo từng Legend
- Slicer field: Customer Type - Customer Type dimension

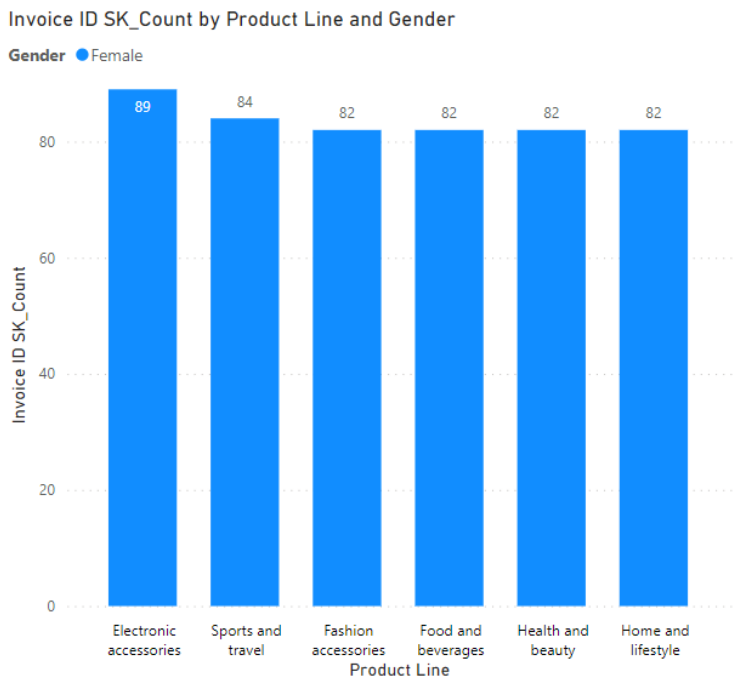
5. Thống kê số lượng sản phẩm bán được theo từng thời điểm (time / date)



Loại biểu đồ: Area Chart

- Trục X: Day (DATE MASTER dimension)
- Trục Y: Quantity (Quantity - supermarket_sales measure)
- Slicer field: Minute (DATE MASTER - Date time - Minute)

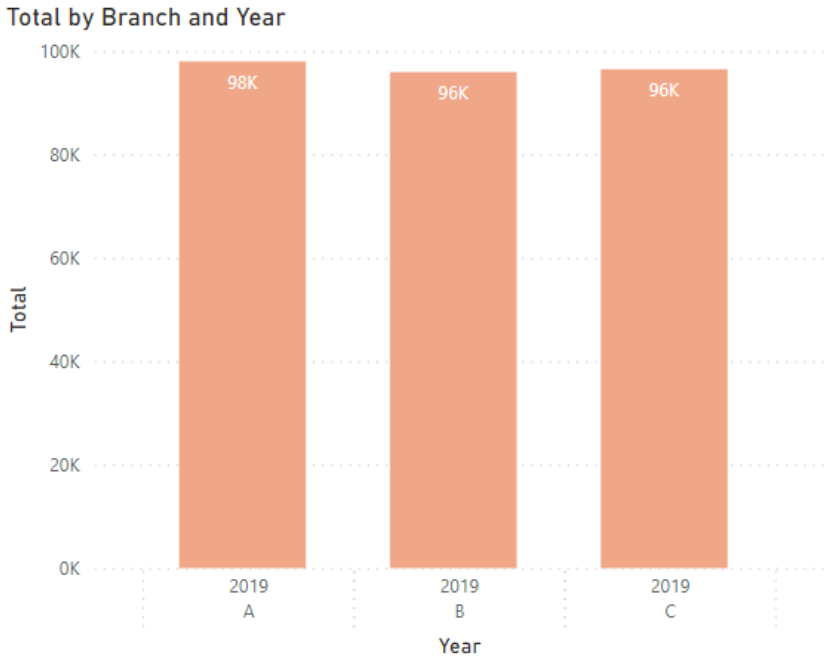
6. Thống kê số lượng khách hàng nữ đã mua theo từng loại sản phẩm



Loại biểu đồ: Stacked Column Chart

- Trục X: ProductLine (ProductLine - Product Line dimension)
- Trục Y: Số lượt khách hàng (Invoice ID SK - supermarket_sales measure)
- Legend: Khách hàng nữ (Gender - Gender dimension)
- Thực hiện filter các table theo từng Legend

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



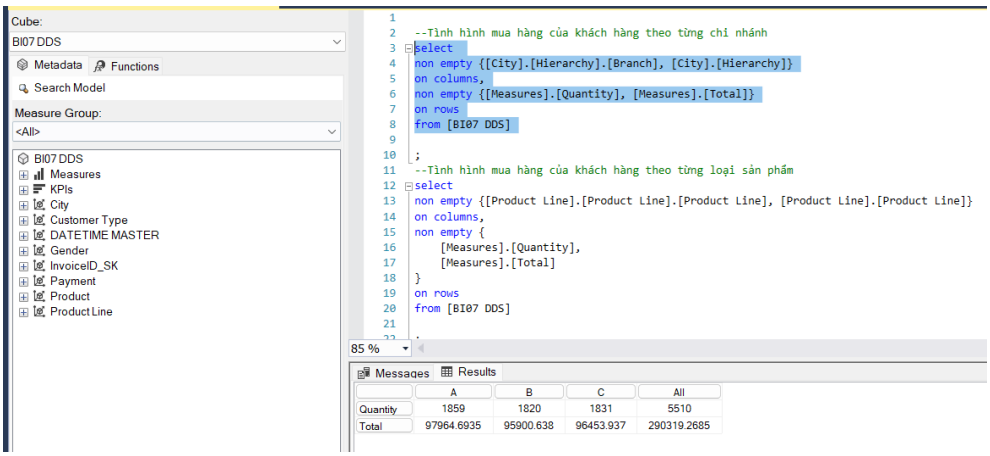
Loại biểu đồ: Stacked Column Chart

- Trục X: Branch (City - Hierarchy - Branch); Year (DATE MASTER - Date time - Year)
- Trục Y: Số doanh thu (Total - supermarket_sales measure)

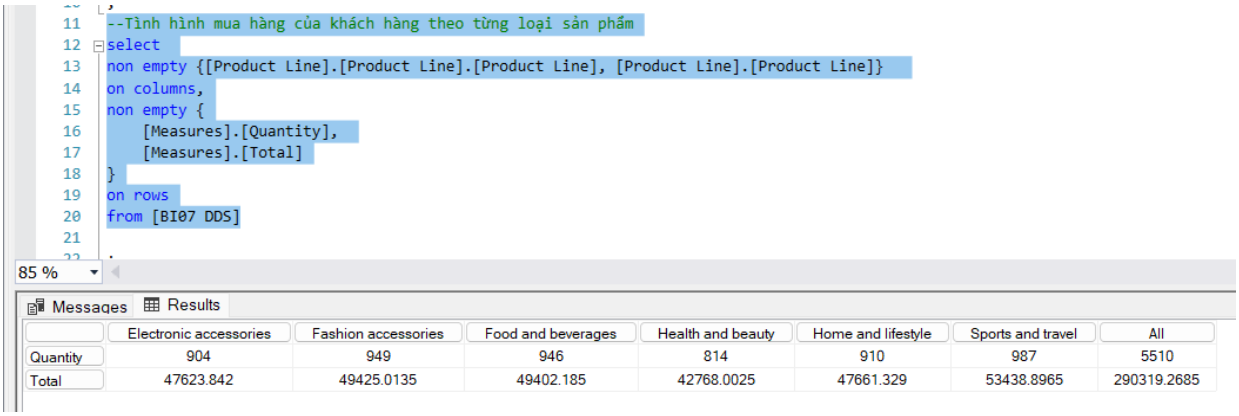
IV. Olap

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

- Tình hình mua hàng của khách hàng theo từng chi nhánh



- Tình hình mua hàng của khách hàng theo từng loại sản phẩm



- Tình hình mua hàng của khách hàng theo thời gian

```
;
--Tình hình mua hàng của khách hàng theo thời gian
select
non empty {[Measures].[Quantity], [Measures].[Total]}
on columns,
non empty CROSSJOIN(
    [DATETIME MASTER].[Date Time].[Month].Members,
    [DATETIME MASTER].[Year].[Year].Members
)
on rows
from [BI07 DDS]

;
--Tình hình mua hàng của khách hàng theo hình thức thanh toán
select
```

85 %

Messages | Results

		Quantity	Total
1	2019	1965	101728.7075
2	2019	1654	87802.5245
3	2019	1891	100788.0365

- Tình hình mua hàng của khách hàng theo hình thức thanh toán

```
--Tình hình mua hàng của khách hàng theo hình thức thanh toán
select
non empty {[Payment].[Payment Type].[Payment Type], [Payment].[Payment Type]}
on columns,
non empty  {[Measures].[Quantity], [Measures].[Total]}
on rows
from [BI07 DDS]

;
--Tình hình mua hàng của khách hàng từng chi nhánh theo hình thức thanh toán
select
```

85 %

Messages | Results

	Cash	Credit card	Ewallet	All
Quantity	1896	1722	1892	5510
Total	97651.963	92252.049	100415.2565	290319.2685

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

```sql
--Tình hình mua hàng của khách hàng theo từng chi nhánh, từng loại sản phẩm, theo thời gian, hình thức thanh toán
SELECT
non empty  {[Measures].[Quantity], [Measures].[Total]} ON COLUMNS,
non empty  CROSSJOIN(
    {[City].[Hierarchy].[Branch].Members},
    {[DATETIME MASTER].[Date Time].[Month].Members},
    {[DATETIME MASTER].[Year].[Year].Members},
    {[Product Line].[Product Line].[Product Line].Members}
    ,{[Payment].[Payment Type].[Payment Type].Members}
  ) ON ROWS
FROM
```

85 %

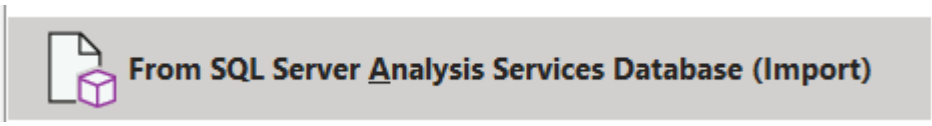
Messages | Results

					Quantity	Total
A	1	2019	Electronic accessories	Cash	26	1529.006
A	1	2019	Electronic accessories	Credit card	8	251.218
A	1	2019	Electronic accessories	Ewallet	40	1145.89
A	1	2019	Fashion accessories	Cash	39	2462.001
A	1	2019	Fashion accessories	Credit card	38	2109.494
A	1	2019	Fashion accessories	Ewallet	64	3393.4665
A	1	2019	Food and beverages	Cash	10	726.389
A	1	2019	Food and beverages	Credit card	53	2478.2555
A	1	2019	Food and beverages	Ewallet	25	1418.901
A	1	2019	Health and beauty	Cash	59	1975.525
A	1	2019	Health and beauty	Credit card	23	1438.319
A	1	2019	Health and beauty	Ewallet	47	2367.1625
A	1	2019	Home and lifestyle	Cash	54	3635.498
A	1	2019	Home and lifestyle	Credit card	54	3609.7435
A	1	2019	Home and lifestyle	Ewallet	48	2836.719
A	1	2019	Sports and travel	Cash	31	1819.535
A	1	2019	Sports and travel	Credit card	15	776.0645
A	1	2019	Sports and travel	Ewallet	51	2696.8505
A	2	2019	Electronic accessories	Cash	26	1562.522
A	2	2019	Electronic accessories	Credit card	23	1037.533
A	2	2019	Electronic accessories	Ewallet	25	1624.804
A	2	2019	Fashion accessories	Cash	22	1198.444
A	2	2019	Fashion accessories	Credit card	29	1093.6115
A	2	2019	Fashion accessories	Ewallet	31	1518.9265
A	2	2019	Food and beverages	Cash	31	1568.108

V. Data Mining

1. Liệu giới tính có liên quan đến loại sản phẩm được mua ? Nếu có dự đoán loại sản phẩm mà người phụ nữ tiếp theo sẽ mua.

   Trước hết ta sẽ tiến hành lấy dữ liệu từ



   Sau đó sẽ tiến hành lựa chọn các dữ liệu từ các dim & fact phù hợp:

Câu 1

Display Options

```
let
    Source = AnalysisServices.Database("ABC1", "SSAS", [TypedMeasureColumns=true, Implementation="2.0"]),
    #"BI07 DDS1" = Source{[Id="BI07 DDS"]}[Data],
    #"BI07 DDS2" = #"BI07 DDS1"{[Id="BI07 DDS"]}[Data],
    #"Added Items" = Cube.Transform(#"BI07 DDS2",
        {
            {Cube.AddAndExpandDimensionColumn, "[Gender]", {"[Gender].[Gender].[Gender]"}, {"Gender.Gender"}},
            {Cube.AddAndExpandDimensionColumn, "[InvoiceID_SK]", {"[InvoiceID_SK].[Invoice ID SK].[Invoice ID SK]"}, {"InvoiceID_SK.Invoice ID SK"}},
            {Cube.AddAndExpandDimensionColumn, "[Product Line]", {"[Product Line].[Product Line].[Product Line]"}, {"Product Line.Product Line"}},
            {Cube.AddMeasureColumn, "Quantity", "[Measures].[Quantity]"}
        }),
    #"Added Items1" = Cube.Transform(#"Added Items",
        {
            {Cube.AddAndExpandDimensionColumn, "[Customer Type]", {"[Customer Type].[Customer Type].[Customer Type]"}, {"Customer Type.Customer Type"}},
            {Cube.AddAndExpandDimensionColumn, "[Payment]", {"[Payment].[Payment Type].[Payment Type]"}, {"Payment.Payment Type"}}
        }),
    #"Added Items2" = Cube.Transform(#"Added Items1",
        {
            {Cube.AddMeasureColumn, "Total", "[Measures].[Total]"}
        })
in
    #"Added Items2"
```

Sau đó ta sẽ tiến hành gọi dữ liệu vào python để thực hiện thuật toán predict vì Visual Studio 2022 không hỗ trợ Mining

```python
df1 = pd.read_excel("supermarket_mining.xlsx", sheet_name= "Câu 1")
df1 = df1.drop('InvoiceID_SK.Invoice ID SK', axis=1)
mapping = {'Gender.Gender': 'Gender','Product Line.Product Line': 'Product Line',
           'Customer Type.Customer Type': 'Customer Type',
           'Payment.Payment Type':'Payment'}
df1 = df1.rename(columns= mapping)
df1.head()
```

Đây là kết quả ta nhận được:

	Gender	Product Line	Quantity	Customer Type	Payment	Total
0	Female	Health and beauty	7	Member	Ewallet	496.6885
1	Female	Home and lifestyle	6	Member	Ewallet	392.3880
2	Female	Health and beauty	2	Member	Credit card	68.8940
3	Female	Sports and travel	7	Member	Credit card	458.3845
4	Female	Electronic accessories	8	Member	Credit card	399.6840

Tiếp theo đó ta sẽ sử dụng Chi-square để xem liệu giới tính có liên quan đến sản phẩm được mua của khách hàng hay không. Và cuối cùng đây là kết quả trả về:

```
Sử dụng Chi-square để xem giới tính và sản phẩm được mua có liên quan đến nhau hay không
H0: Không có sự liên quan đáng kể giữa giới tính và loại sản phẩm được mua
H1: Có ít nhất một sự khác bị đáng kể giữa giới tính và loại sản phẩm được mua
Chi-square statistic: 3.3364
P-value: 0.6483
Không đủ bằng chứng để bác bỏ giả thiết H0: Không có sự ảnh hưởng đáng kể của giới tính đối với loại sản phẩm được mua.
Giới tính và loại sản phẩm được mua không có sự liên quan đáng kể đến nhau
```

2. Cho một khách hàng có giới tính là Female, hiện đang là Member của siêu thị, và cô ấy có thói quen sử dụng Creadit card, dự đoán lần tiếp theo cô ấy mua hàng thì tổng giá trị đơn hàng là bao nhiêu ?

Với câu này tôi vẫn sẽ sử dụng dữ liệu đã lấy từ câu 1, sau đó tiến hành sử dụng LinearRegression để giải quyết mining, và đây chính là kết quả thu được:

```
Độ lỗi: 43781.107192545
Dự đoán cô ấy sẽ mua hàng có tổng giá tiền là: 288.0

Ở mô hình hiện tại có độ lỗi rất cao, thế nên đây không phải là mô hình hoàn hảo, sẽ có sự chênh lệch lớn giữa giá trị dự đoán và giá trị thực tế
Tuy nhiên, với dataset hiện tại ta không tìm được mô hình phù hợp hơn
```

Như ta đã thấy ở đây, mô hình dự đoán có độ lỗi rất cao, tuy nhiên vì dữ liệu ít nên ta sẽ tạm thời chấp nhận kết quả dự đoán này.

3. Siêu thị chuẩn bị mở 1 chi nhánh D mới ở thành phố HCM, dự đoán doanh thu chi nhánh mới với dữ liệu support cho sẵn

Trước hết tôi sẽ tiến hành bỏ thêm dữ liệu support về thông tin các thành phố như sau:

City.City	Population	Female	Male	Rival
HCM	7200	49	51	9
Mandalay	5000	30	70	5
Naypyitaw	6000	42	48	9
Yangon	7300	59	41	2

Với City.City là tên thành phố, Population là dân số, Female là tỷ lệ nữ giới, Male là tỷ lệ nam mới và Rival là số đối thủ cạnh tranh. Rồi tôi tiến hành đọc dữ liệu vừa tạo vào python

```python
df2_sp = pd.read_excel("supermarket_mining.xlsx", sheet_name= "Câu 2 support")
df2_sp = df2_sp.rename(columns= {'City.City':'City'})
df2_sp.head()
```

✓ 0.0s Python

	City	Population	Female	Male	Rival
0	HCM	7200	49	51	9
1	Mandalay	5000	30	70	5
2	Naypyitaw	6000	42	48	9
3	Yangon	7300	59	41	2

Sau đó tôi tiến hành lấy dữ liệu mình cần từ các bảng dim & fact:

Câu 2

```
let
    Source = AnalysisServices.Database("ABC1", "SSAS", [TypedMeasureColumns=true, Implementation="2.0"]),
    #"BI07 DDS1" = Source{[Id="BI07 DDS"]}[Data],
    #"BI07 DDS2" = #"BI07 DDS1"{[Id="BI07 DDS"]}[Data],
    #"Added Items" = Cube.Transform(#"BI07 DDS2",
        {
            {Cube.AddAndExpandDimensionColumn, "[City]", {"[City].[Hierarchy].[Branch]", "[City].[Hierarchy].[City]"}, {"City.Branch", "City.City"}},
            {Cube.AddMeasureColumn, "Gross Income", "[Measures].[Gross Income]"}
        })
in
    #"Added Items"
```

Tôi tiến hành merge 2 dữ liệu lại với nhau:

```
df_combined = pd.merge(df2, df2_sp, on='City')
df_combined.head()
```

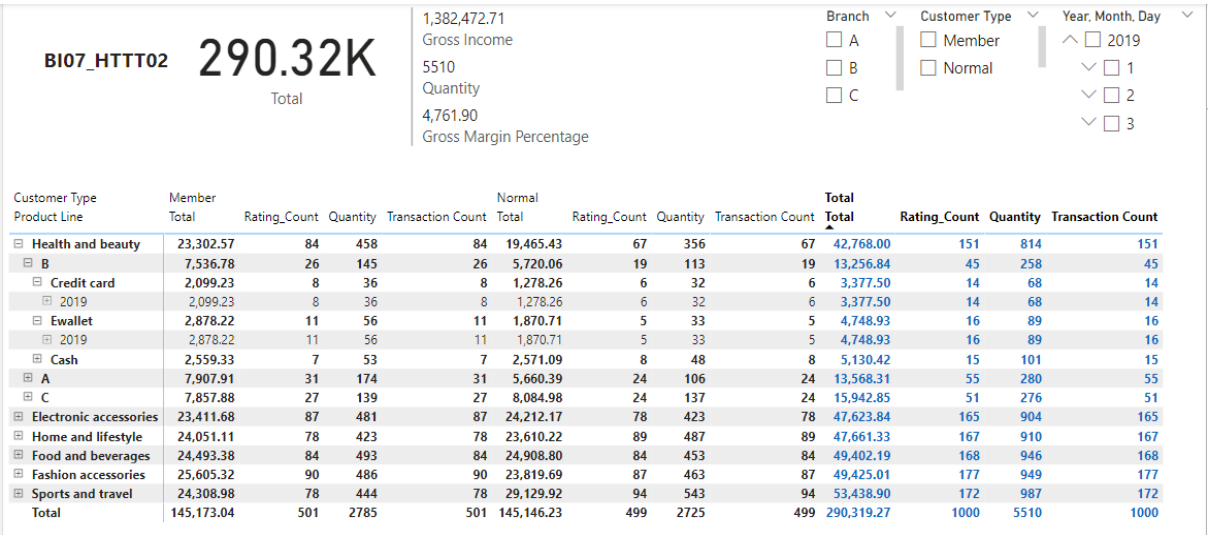
✓ 0.0s

	Branch	City	Gross Income	Population	Female	Male	Rival
0	A	Yangon	466498.540486	7300	59	41	2
1	B	Mandalay	456669.704771	5000	30	70	5
2	C	Naypyitaw	459304.461914	6000	42	48	9

Và sau khi thực hiện thuật toán LinearRegression, đây là kết quả tôi thu được:

```
Độ lỗi: 72444902.69759203
Dự đoán doanh thu của chi nhánh mới: 457987.0833424933

 Tuy độ lỗi quá cao nhưng tạm chấp nhận vì dữ liệu không đầy đủ
```

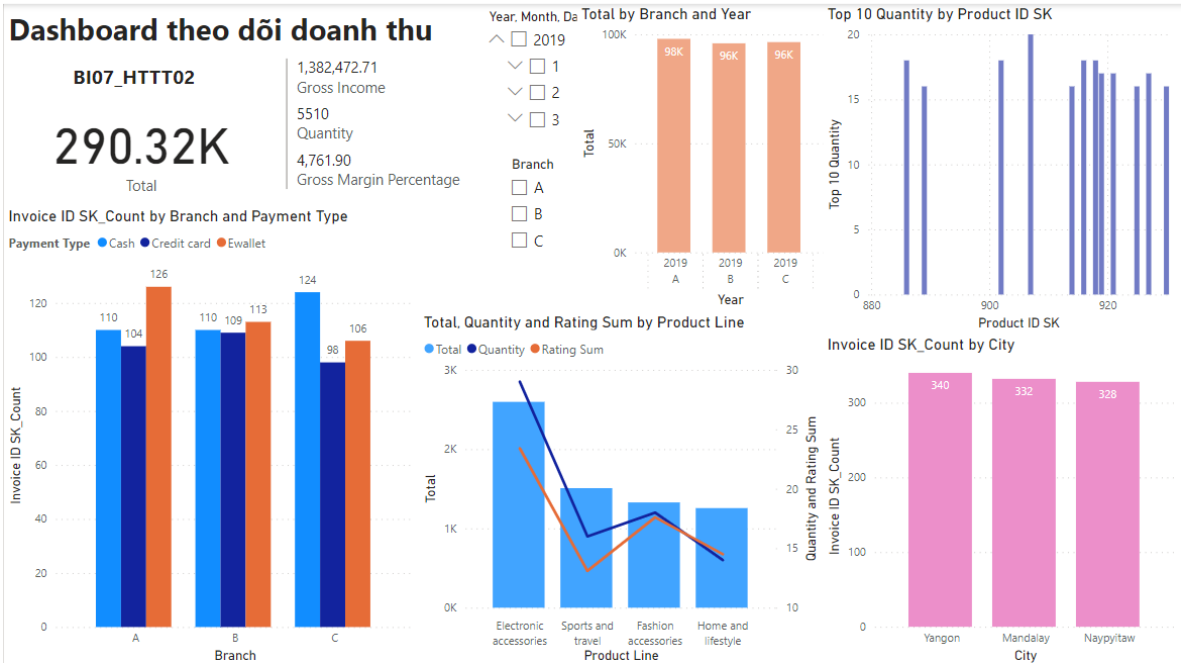
VI. Dashboard



Customer Type / Product Line	Member Total	Rating_Count	Quantity	Transaction Count	Normal Total	Rating_Count	Quantity	Transaction Count	Total Total	Rating_Count	Quantity	Transaction Count
Health and beauty	23,302.57	84	458	84	19,465.43	67	356	67	42,768.00	151	814	151
B	7,536.78	26	145	26	5,720.06	19	113	19	13,256.84	45	258	45
Credit card	2,099.23	8	36	8	1,278.26	6	32	6	3,377.50	14	68	14
2019	2,099.23	8	36	8	1,278.26	6	32	6	3,377.50	14	68	14
Ewallet	2,878.22	11	56	11	1,870.71	5	33	5	4,748.93	16	89	16
2019	2,878.22	11	56	11	1,870.71	5	33	5	4,748.93	16	89	16
Cash	2,559.33	7	53	7	2,571.09	8	48	8	5,130.42	15	101	15
A	7,907.91	31	174	31	5,660.39	24	106	24	13,568.31	55	280	55
C	7,857.88	27	139	27	8,084.98	24	137	24	15,942.85	51	276	51
Electronic accessories	23,411.68	87	481	87	24,212.17	78	423	78	47,623.84	165	904	165
Home and lifestyle	24,051.11	78	423	78	23,610.22	89	487	89	47,661.33	167	910	167
Food and beverages	24,493.38	84	493	84	24,908.80	84	453	84	49,402.19	168	946	168
Fashion accessories	25,605.32	90	486	90	23,819.69	87	463	87	49,425.01	177	949	177
Sports and travel	24,308.98	78	444	78	29,129.92	94	543	94	53,438.90	172	987	172
Total	145,173.04	501	2785	501	145,146.23	499	2725	499	290,319.27	1000	5510	1000

Cho biết doanh thu của từng chi nhánh của các năm, cho biết lượng khách hàng thanh toán theo mỗi hình thức cash, debit... của mỗi chi nhánh / của tất cả chi nhánh theo từng năm...

Dashboard theo dõi doanh thu, top sp bán chạy, chi nhánh đạt doanh thu cao nhất, city có đông lượt mua hàng nhất, theo dõi doanh thu của 1 chi nhánh bất kỳ



C. MOVIE DATASET

I. Attribute Information

Nhóm tiến hành kết hợp 2 Data từ 2 source:

a. Source 1: [film_sales.xlsx]

Bảng	Thuộc tính	Ý nghĩa	Phụ thuộc	Định dạng
Film	FilmID	Mã phim		char
	Film Name	Tên phim		char
	FilmCategoryID	Mã thể loại phim	Film.FilmCategoryID tham chiếu từ thuộc tính FilmCategoryID trong bảng Film Category	char
	Price	Đơn giá của phim	Price >= 0	char

Film Category	FilmCategoryID	Mã thể loại phim		char
	FilmCategory	Tên thể loại phim		char
Cinema	CinemaID	Mã rạp phim		char
	CinemaName	Tên rạp phim		char
	CityID	Mã thành phố	Cinema.CityID tham chiếu từ thuộc tính CityID trong bảng City	char
	Capacity	Sức chứa của rạp		int
	Standard_Seats_price	Giá của ghế hạng phổ thông		int
	VIP_Seats_price	Giá của ghế hạng cao cấp		int
City	CityID	Mã thành phố		char
	CityName	Tên thành phố/chi nhánh		char
MovieSales	RevenueID	Mã doanh thu		char
	FilmID	Mã phim	MovieSales.FilmID tham chiếu từ thuộc tính FilmID trong bảng Film	char
	CinemaName	Tên rạp phim	MovieSales.CinemaName tham chiếu từ thuộc tính CinemaName trong bảng Cinema	char
	total_sales	Doanh thu bán	total_sales = (Standard_Seats_sold-Standard_Seats_out) * (Price +	int

			Standard_Seats_price) + (VIP_Seats_sold - VIP_Seats_out) * (Price + VIP_Seats_price)	
	tickets_sold	Số lượng vé đã bán	tickets_sold = Standard_Seats_sold + VIP_Seats_sold	int
	Standard_Seats_sold	Số lượng vé bán ra của ghế hạng phổ thông		int
	VIP_Seats_sold	Số lượng vé bán ra của ghế hạng cao cấp		int
	Standard_Seats_out	Số lượng vé bán bị hủy của ghế hạng phổ thông	Standard_Seats_out <= Standard_Seats_sold	int
	VIP_Seats_out	Số lượng vé bán bị hủy của ghế hạng cao cấp	VIP_Seats_out <= VIP_Seats_sold	int
	tickets_out	Số lượng vé bán bị hủy	tickets_out = Standard_Seats_out + VIP_Seats_out; tickets_out <= tickets_sold	int
	occu_perc	Phần trăm rạp chiếu phim theo công suất sẵn có	occu_perc <= 100%; occu_perc = ticket_use*100/Capacity	float
	ticket_use	Vé được sử dụng	ticket_use = tickets_sold - tickets_out; ticket_use <= tickets_sold	int
	Date	Ngày xuất vé	3/2018 <= Date <= 10/2018	date

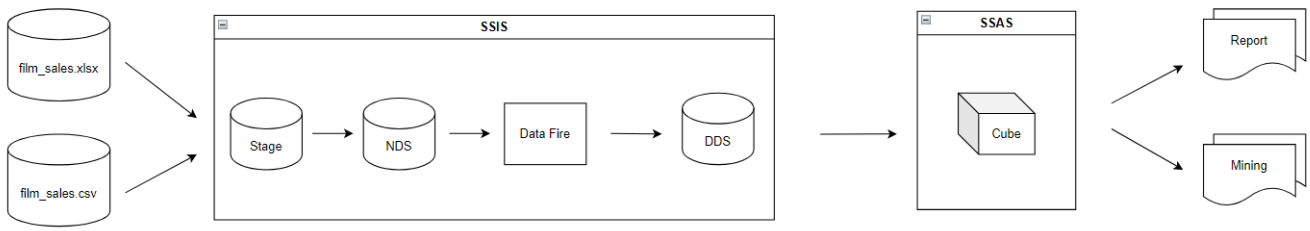
b. Source 2: film_sales.csv

Bảng	Thuộc tính	Ý nghĩa	Phụ thuộc	Định dạng
film_sales	RevenueID	Mã doanh thu		char
	Film Name	Tên phim		char
	CinemaID	Mã rạp phim		char
	total_sales	Doanh thu bán	total_sales = (Standard_Seats_sold-Standard_Seats_out) * (Price + Standard_Seats_price) + (VIP_Seats_sold - VIP_Seats_out) * (Price + VIP_Seats_price)	int
	Standard_Seats_sold	Số lượng vé bán ra của ghế hạng phổ thông		int
	VIP_Seats_sold	Số lượng vé bán ra của ghế hạng cao cấp		int
	Standard_Seats_out	Số lượng vé bán bị hủy của ghế hạng phổ thông	Standard_Seats_out <= Standard_Seats_sold	int
	VIP_Seats_out	Số lượng vé bán bị hủy của ghế hạng cao cấp	VIP_Seats_out <= VIP_Seats_sold	int
	occu_perc	Phần trăm rạp chiếu phim theo công suất sẵn có	occu_perc <= 100%; occu_perc = ticket_use*100/Capacity	float
	ticket_use	Vé được sử dụng		int

	year	Năm xuất vé	year = 2018	int
	quarter	Quý xuất vé	quarter = {1, 2, 3, 4}	int
	month	Tháng xuất vé	3<= month <= 10	int
	day	Ngày xuất vé	1 <= day <=30	int

II. Data warehouse design

1. Design Data Flow



Dataflow film

2. Design NDS

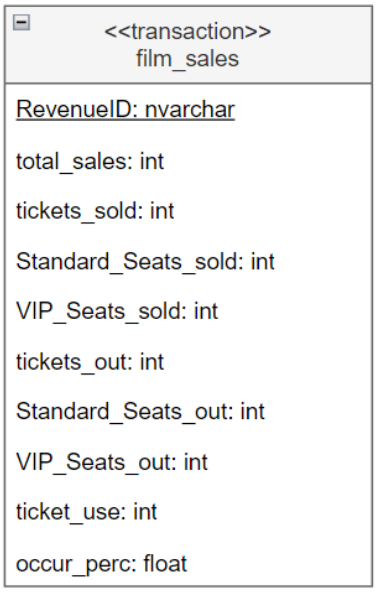
***Stage:* film_sales_S1:**

	RevenueID	FilmID	Cinema Name	total_sales	tickets_sold	Standard_Seats_sold	VIP_Seats_sold	Standard_Seats_out	VIP_Seats_out	tickets_out	occur_perc	ticket_use	date	SOURCE_ID
1	RV1	F2	Mega Cinema	4160	208	101	107	0	0	0	16	208	2018-09-18	1
2	RV2	F2	Cinestar	19980	999	828	171	0	1	1	76.77	998	2018-09-16	1
3	RV3	F2	Cinestar	25280	1264	1133	131	1	8	9	96.54	1255	2018-09-14	1
4	RV4	F2	Cinestar	4400	220	108	112	1	8	9	16.23	211	2018-09-13	1
5	RV5	F2	Cinestar	8520	426	122	304	1	1	2	32.62	424	2018-12-09	1
6	RV6	F2	Cinestar	1420	71	71	0	1	8	9	4.77	62	2018-11-09	1
7	RV7	F2	Mega Cinema	2620	131	43	88	1	1	2	9.92	129	2018-09-30	1
8	RV8	F2	Mega Cinema	4000	200	112	88	0	1	1	15.31	199	2018-09-26	1
9	RV9	F2	Cinestar	7440	372	115	257	0	0	0	28.62	372	2018-09-25	1
10	RV10	F2	Mega Cinema	20320	1016	861	155	1	2	3	77.92	1013	2018-09-25	1

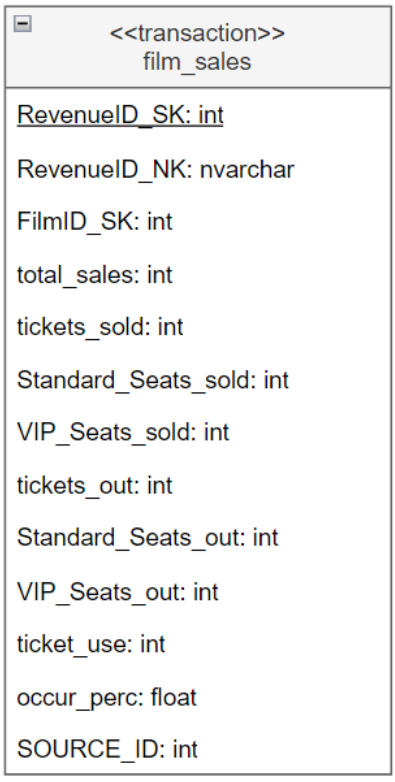
***Stage:* film_sales_S2:**

	RevenueID	Film Name	Cinema ID	total_sales	Standard_Seats_sold	VIP_Seats_sold	Standard_Seats_out	VIP_Seats_out	occur_perc	ticket_use	year	quarter	month	day	SOURCE_ID
1	RV1	Avengers: Endgame	C1	29125	517	648	2	2	64.5	1161	2018	2	5	5	2
2	RV2	Avengers: Endgame	C1	36450	791	667	2	5	80.61	1451	2018	2	5	4	2
3	RV3	Avengers: Endgame	C1	2350	62	32	0	6	4.89	88	2018	2	5	3	2
4	RV4	Avengers: Endgame	C1	4300	143	29	0	10	9	162	2018	2	5	2	2
5	RV5	Avengers: Endgame	C1	26375	894	161	3	3	58.28	1049	2018	2	5	6	2
6	RV6	Avengers: Endgame	C1	37625	1374	131	0	3	83.44	1502	2018	2	6	26	2
7	RV7	Avengers: Endgame	C2	61825	1315	1158	0	2	98.84	2471	2018	2	6	26	2
8	RV8	Avengers: Endgame	C2	24525	177	804	0	2	39.16	979	2018	2	6	25	2
9	RV9	Avengers: Endgame	C2	48250	417	1513	1	2	77.08	1927	2018	2	6	24	2
10	RV10	Avengers: Endgame	C2	9000	170	190	3	5	14.08	352	2018	2	6	23	2

Vì **film_sales** là bảng lưu giá trị doanh thu, ta sẽ tiến hành tạo 1 bảng **film_sales** trong ***NDS*** với loại là Transaction với các thuộc tính Measure có chung giữa 2 source:



Từ 2 source ta có thể thấy được RevenueID là khoá chính của 2 source và hiện đang có một bộ đếm độc lập với nhau => Như vậy cần tạo khoá thay thế Invoice ID_SK để tạo khoá tự tăng dần trong NDS, đổi tên cột Invoice ID thành Invoice ID_NK để lưu giá trị khoá chính cũ, thêm thuộc tính SOURCE_ID để lưu thông tin source:



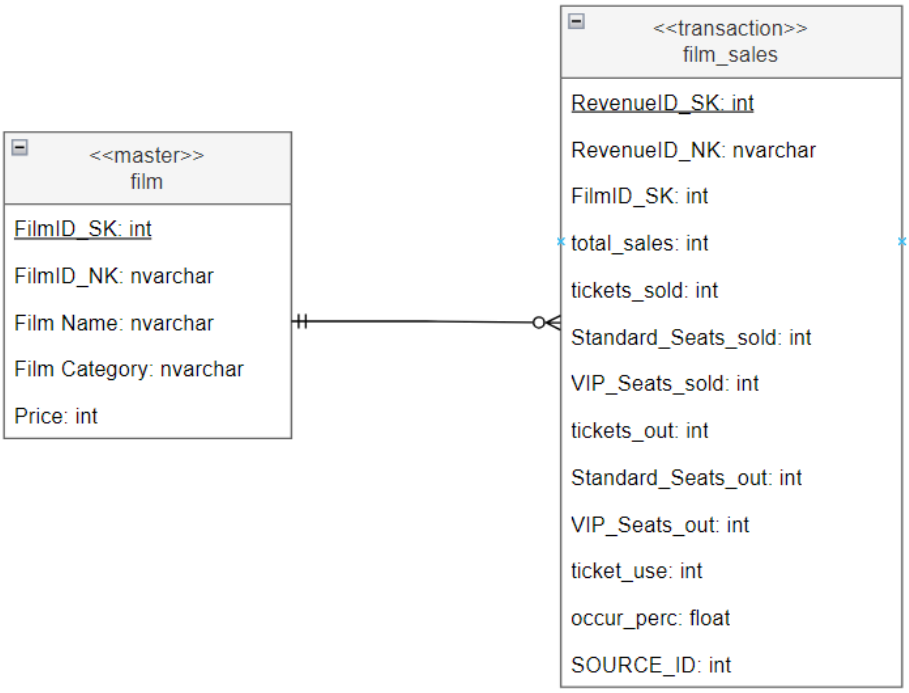
Tiếp đến thông tin Film trong 2 source, ta có thể thấy 2 source đang lưu 2 loại dữ liệu khác nhau, **film_sales_S1**: FilmID, **film_sales_S2**: Film Name. Hãy nhìn đến bảng **film** trong ***Stage***:

***Stage* film:**

	FilmID	Film Name	FilmCategoryID	Price
1	F1	Avengers: Endgame	FC1	25
2	F2	Deadpool 2	FC1	20
3	F3	Home Alone	FC3	18
4	F4	One Piece Film: Red	FC2	26
5	F5	Avatar 2	FC1	25
6	F6	Elemental	FC2	20
7	F7	Wonka	FC3	25

Từ bảng ***Stage* film**, ta có thể thấy ***Stage* film** có khóa chính là FilmID, từ FilmID ta có thể tìm được một Film Name. Như vậy trước hết ta sẽ thêm một bảng **film** dạng master trong ***NDS*** với khoá FilmID_SK là khoá thay thế tự tăng, FilmID được đổi tên thành FilmID_NK để lưu giá trị khoá cũ từ stage:

Sau đó ta sẽ thêm thuộc tính FilmID vào **film_sales *NDS*** tham chiếu đến với FilmID_SK của bảng **film** master mà ta vừa tạo:

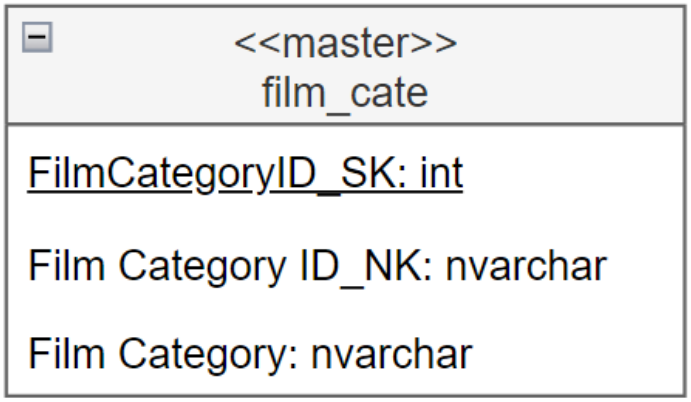


Vì trong **film *stage*** có chứa thuộc tính FilmCategoryID được lấy từ bảng **film_cate**. Nhìn vào bảng **film_cate trong *Stage***:

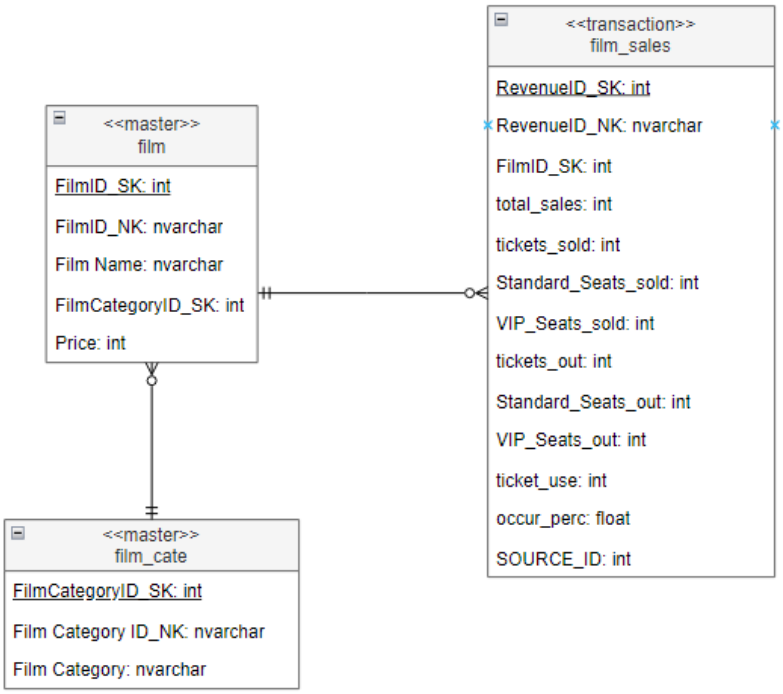
***Stage* film_cate:**

	FilmCategoryID	Film Category
1	FC1	Action
2	FC2	Animation
3	FC3	Kid

Ta có thể thấy 1 khoá chính là FilmCategoryID, từ 1 FilmCategoryID ta chỉ tìm được một Film Category duy nhất, như vậy ta sẽ tiến hành tạo **bảng film_cate trong *NDS*** để lưu thông tin thể loại film, với FilmCategoryID_SK là khoá thay thế tăng dần, FilmCategoryID thì được đổi thành FilmCategoryID_NK để lưu giá trị cũ



Tiếp đó ta sẽ đổi tên cột Film Category trong **film *NDS*** thành FilmCategoryID_SK để tham chiếu đến **bảng film_cate** vừa tạo:

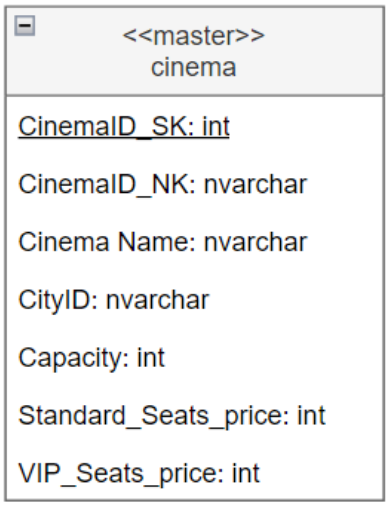


Tương tự với thuộc tính cinema, trong bảng film_sales của 2 source, ta có thể thấy 2 source đang lưu 2 loại dữ liệu khác nhau, **film_sales_S1**: Cinema Name, **film_sales_S2**: CinemaID. Hãy nhìn đến **bảng Cinema trong *Stage***:

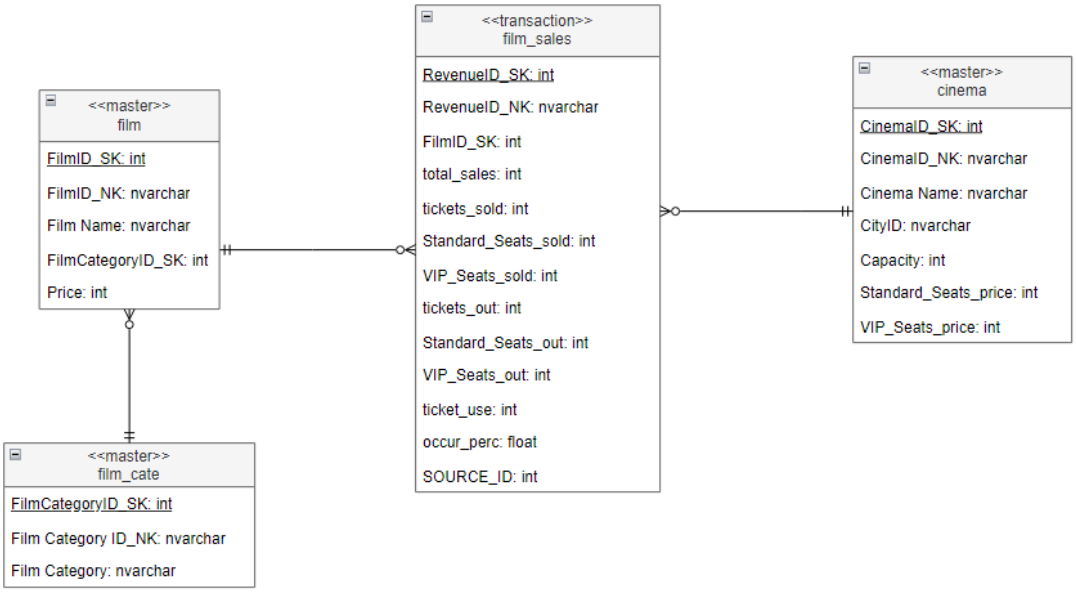
***Stage* cinema:**

	CinemaID	Cinema Name	CityID	Capacity	Standard_Seats_price	VIP_Seats_price
1	C1	Galaxy Cinema	HCM	1800	30	50
2	C2	CGV Cinema	HCM	2500	25	40
3	C3	Mega Cinema	HN	1300	20	30
4	C4	Cinestar	DN	1300	27	39
5	C5	BHD Cinema	HN	1500	25	45

Ta có thể thấy từ **bảng cinema trong *stage*** thì CinemaID là khoá chính, 1 CinemaID có thể tìm được một Cinema Name duy nhất, thế nên ta sẽ tiến hành tạo 1 **bảng cinema trong *NDS*** để lưu trữ thông tin rạp phim với CinemaID_SK là khoá thay thế, CinemaID đổi tên thành CinemaID_NK để lưu trữ giá trị khoá từ ***Stage***.



Sau đó ta sẽ tiến hành thêm cột CinemaID_SK vào bảng **film_sales** trong ***NDS*** và tham chiếu đến với CinemaID_SK của **bảng cimena trong *NDS***:

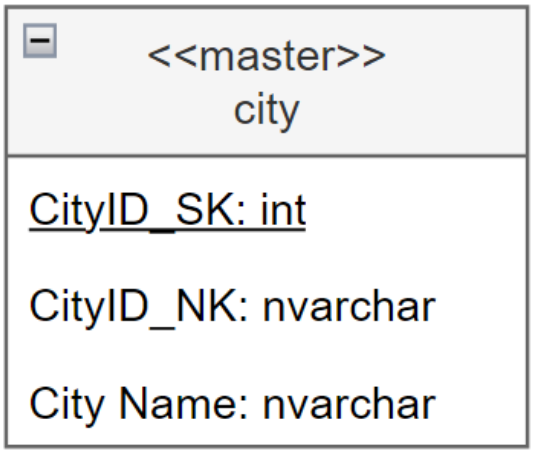


Vì trong bảng **cinema *stage*** có cứ thông tin về thành phố CityID, ta sẽ xem thử bảng city trong stage:

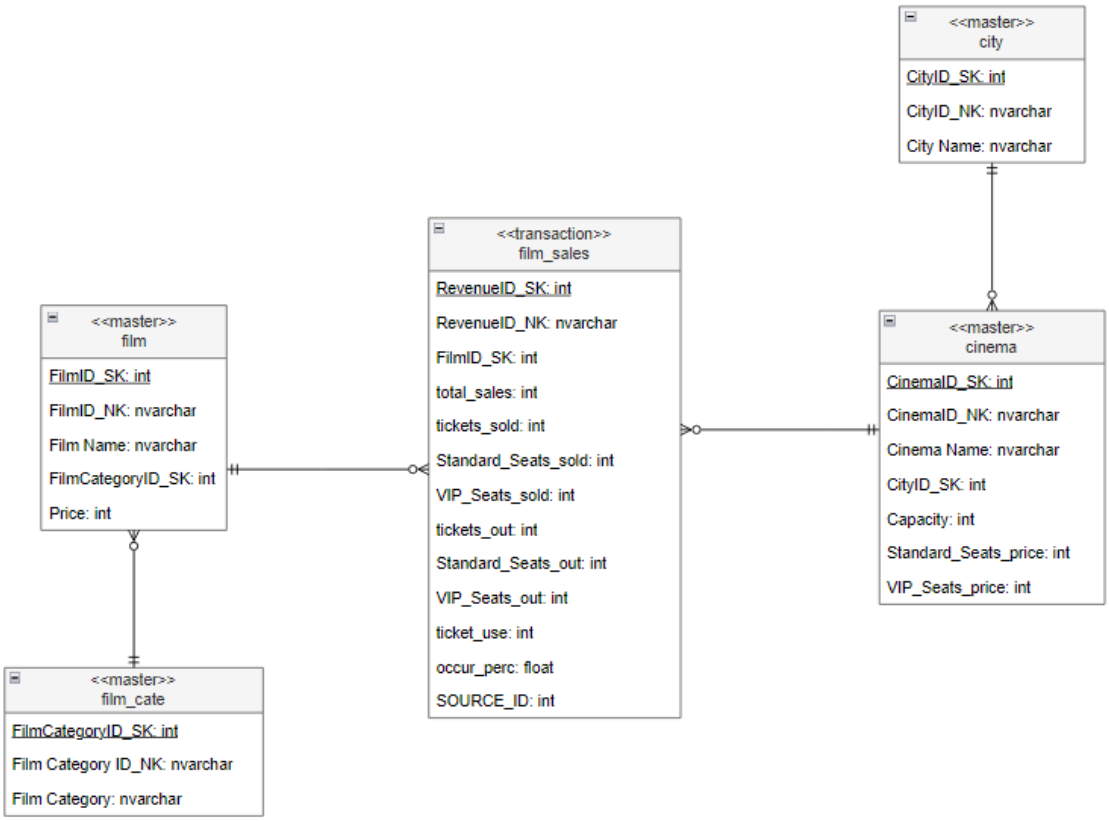
***Stage* city:**

	CityID	CityName
1	HCM	Ho Chi Minh
2	HN	Ha Noi
3	DN	Da Nang

Từ ảnh ta có thể thấy CityID là khoá chính của bảng **city trong *stage***, thế nên ta sẽ tạo thêm một bảng **city trong *NDS*** để lưu thông tin về city, với CityID_SK là khoá thay thế tự tăng, CityID thì được đổi thành CityID_NK để lưu trữ giá trị khoá cũ:

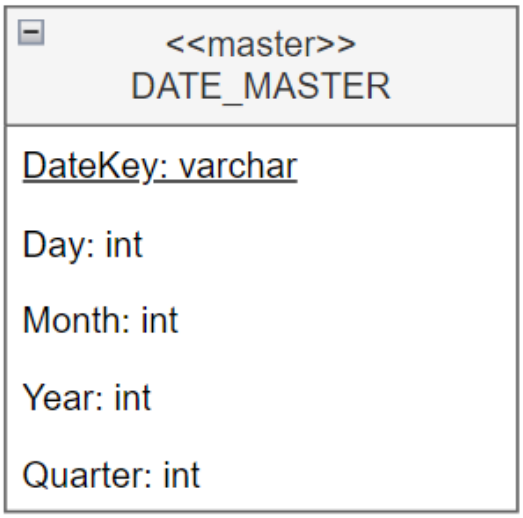


Sau đó ta sẽ đổi cột CityID trong bảng **cinema *NDS*** thành CityID_SK và tham chiếu đến cột CityID_SK trong bảng **city *NDS***:



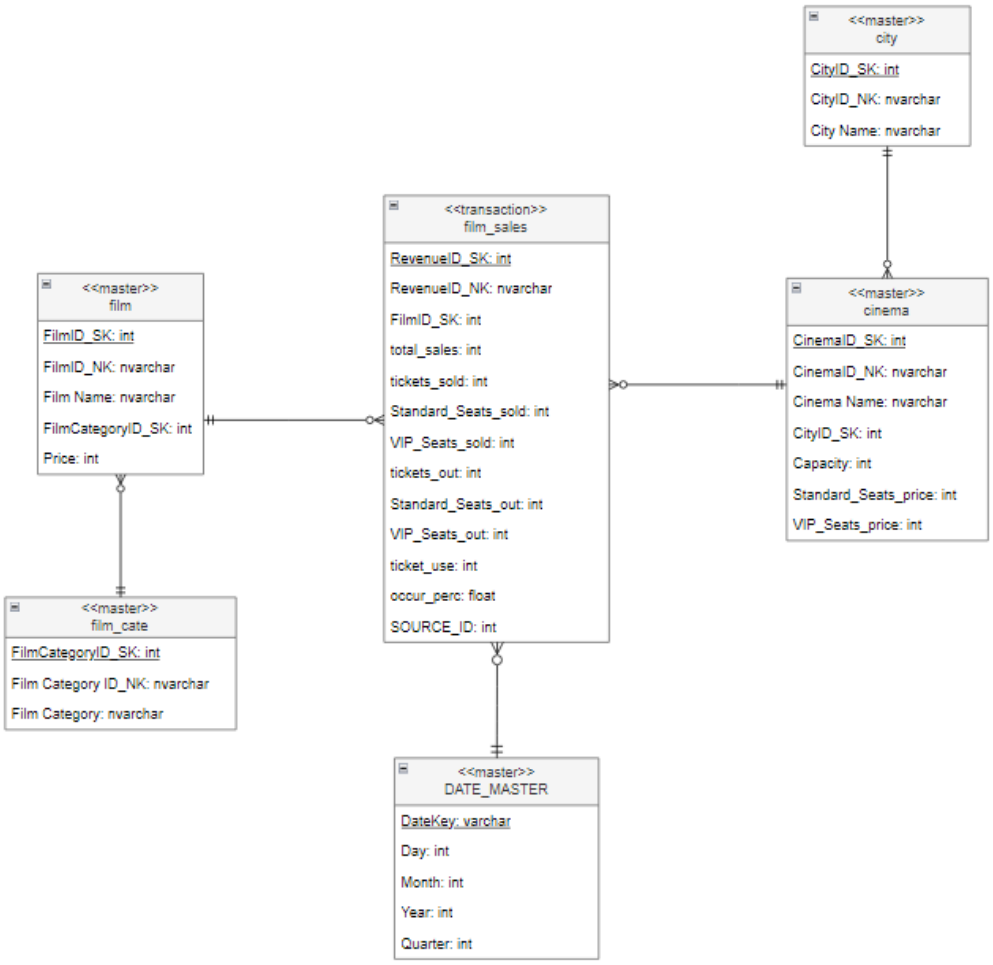
Tiếp đến việc xử lý ngày tháng năm, vì đây là thiết kế để phục vụ cho data warehouse, thế nên ta cần tạo một **DATE_MASTER trong *NDS*** để lưu giá trị ngày tháng năm, đồng thời vì ở film_sales_S2 trong stage có lưu thêm thuộc tính quarter là quý trong năm, thế nên ta sẽ tạo bảng **DATE_MASTER** gồm Date, Month, Quarter, Year và thêm một cột DATEKEY có dạng: 'YYYYMMDD' để làm khóa chính.

Vì bảng **DATE_MASTER** là bảng về datetime thế nên ta sẽ tự tạo trước dữ liệu trong **DATE_MASTER** thông qua script theo yêu cầu BI.



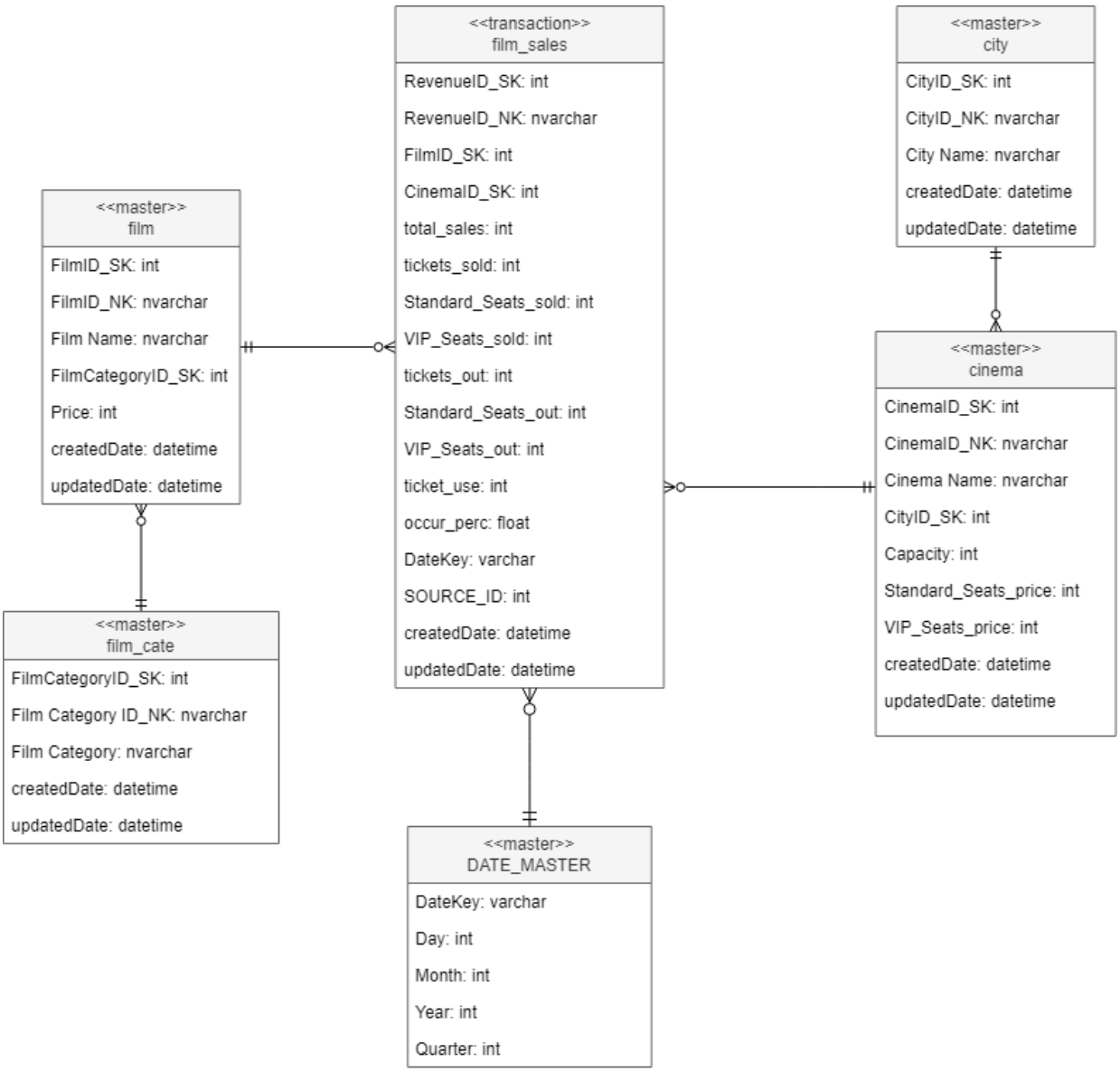
	DateKey	Day	Month	Year	Quarter
1	20180301	1	3	2018	1
2	20180302	2	3	2018	1
3	20180303	3	3	2018	1
4	20180304	4	3	2018	1
5	20180305	5	3	2018	1
6	20180306	6	3	2018	1
7	20180307	7	3	2018	1
8	20180308	8	3	2018	1
9	20180309	9	3	2018	1
10	20180310	10	3	2018	1
11	20180311	11	3	2018	1
12	20180312	12	3	2018	1
13	20180313	13	3	2018	1
14	20180314	14	3	2018	1
15	20180315	15	3	2018	1
16	20180316	16	3	2018	1
17	20180317	17	3	2018	1

Ta sẽ thêm thuộc tính DATEKEY vào bảng **film *NDS*** và tham chiếu đến DATEKEY trong **DATE_MASTER**:



Ở ***NDS*** ta sẽ thực hiện Incremental Extract, tức là cần so sánh với LSET, CET, ta sẽ thêm ở mỗi bảng Transaction & Master (Trừ bảng DATE_MASTER) thuộc tính createdDate, updatedDate để tiện theo dõi quá trình ETL.

Như vậy đây là thiết kế ***NDS*** của film:

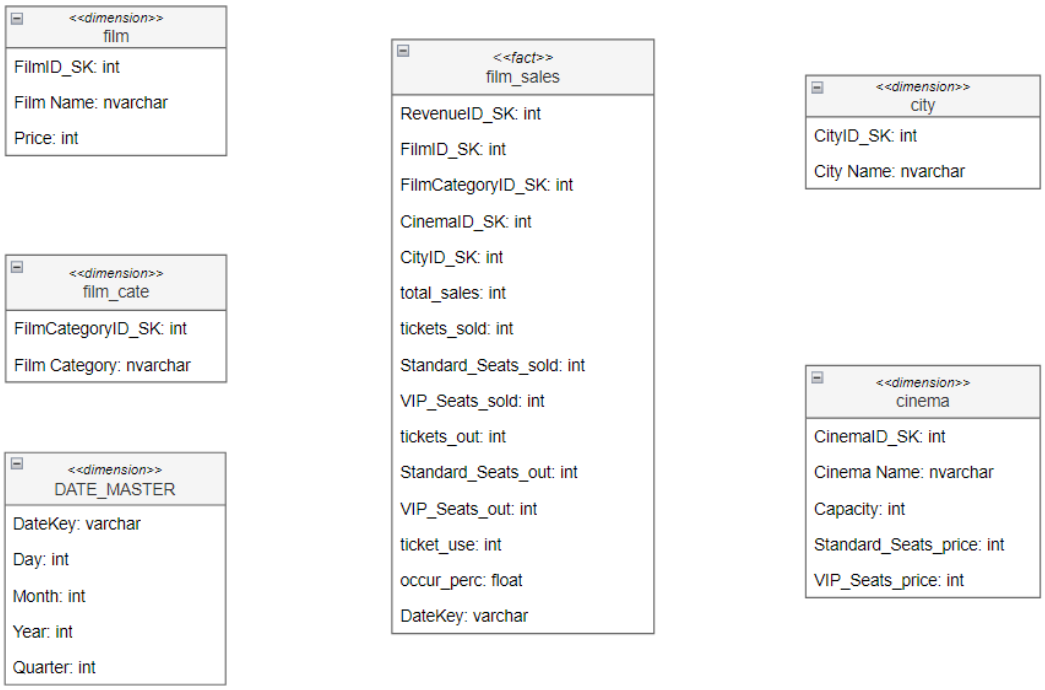


NDS

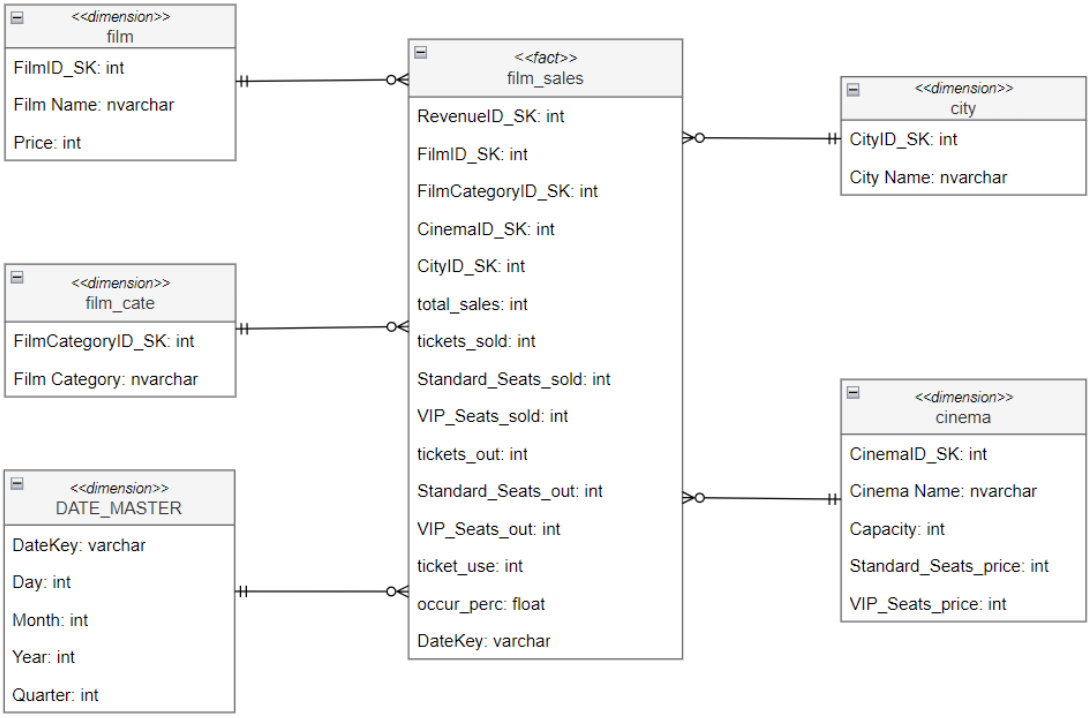
3. Design DDS

Ở DDS, nhóm lựa chọn thiết kế Star Schema.

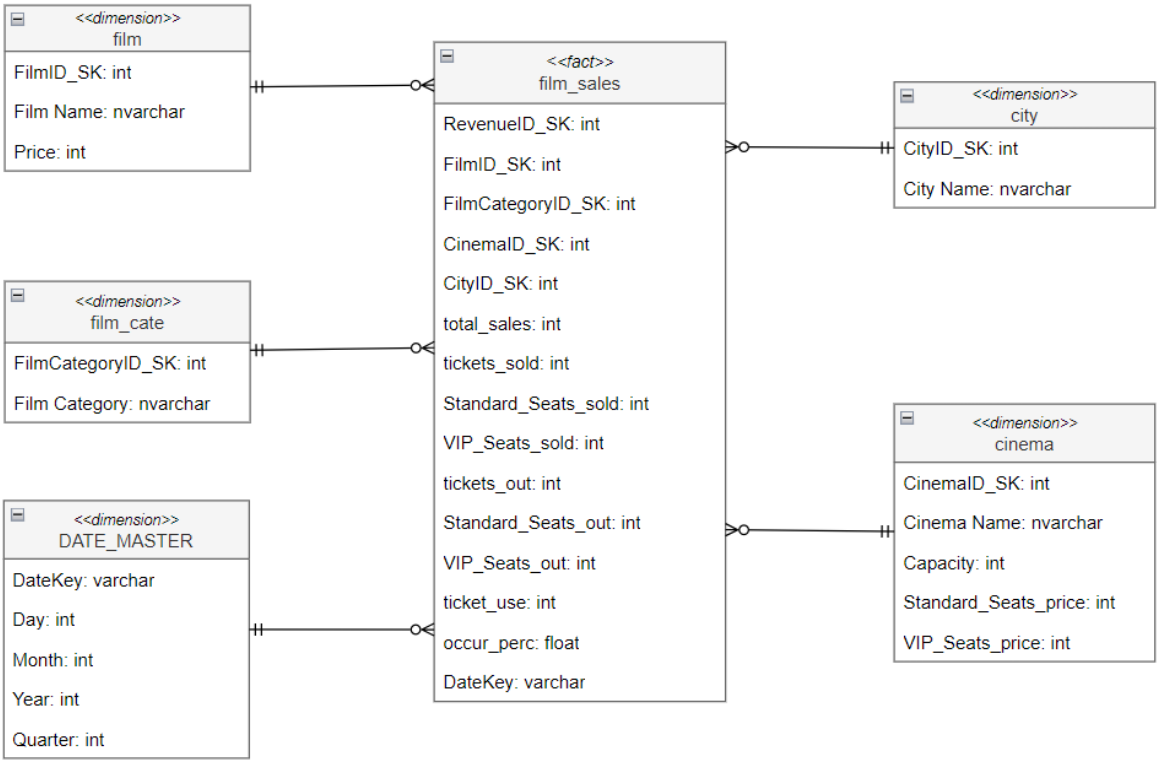
Trước hết, ta sẽ lấy lại thiết kế của NDS, chuyển các bảng Master thành Dimension và bảng Transaction thành bảng fact, sau đó sẽ loại bỏ tất cả các khoá NK & createdDate, updatedDate ở mỗi bảng và xóa thuộc tính SOURCE_ID trong bảng Fact:



Tiếp đó ta sẽ đem tất cả những thuộc tính có khóa ngoại trong các bảng Dimension bỏ vào bảng fact, nối tham chiếu, và loại bỏ các thuộc tính khóa ngoại đó trong các bảng Dimension.



Sau cùng ta sẽ kiểm tra yêu cầu về phân cấp chiều, chi tiết dữ liệu dựa theo yêu cầu Report và Mining, nhưng vì đây là DataSet tự chọn, nên hiện thiết kế DDS đã hoàn toàn phù hợp.

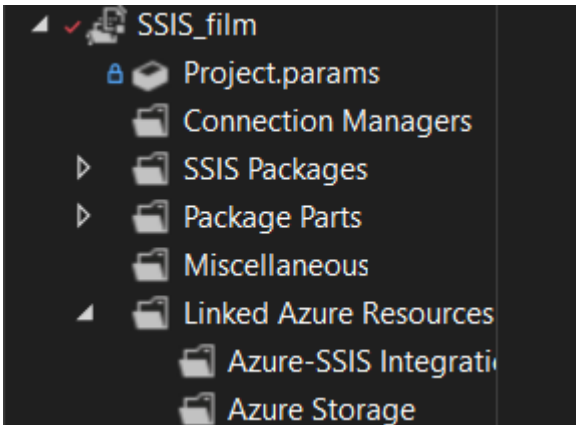


DDS

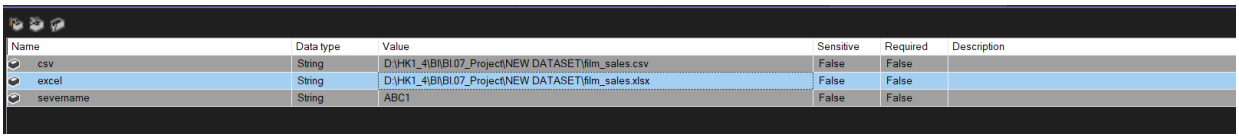
III. ETL

1. Set up Project Param

Để thuận tiện cho việc chạy ETL nhiều máy, cài đặt Project.param để lưu giá trị connection và tên server vào các biến trên toàn project.
Trước hết hãy click chọn vào file Project.param trong project:

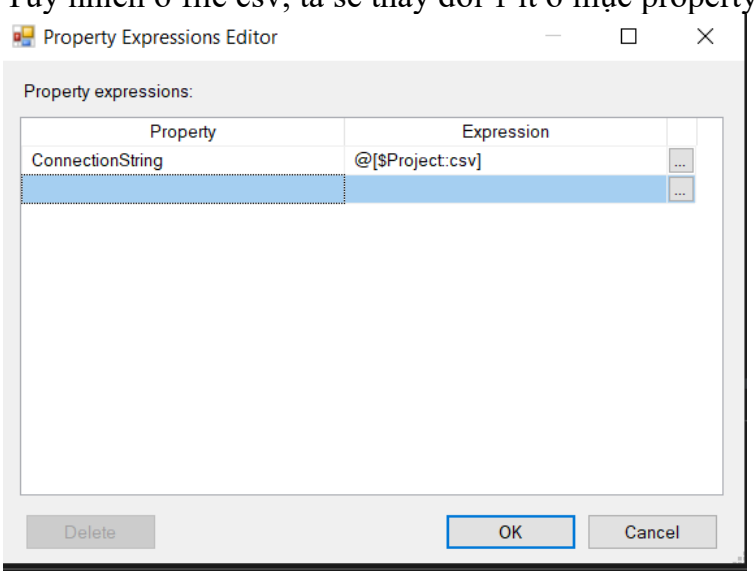


Sau đó ta sẽ tiến hành thêm các biến như sau:

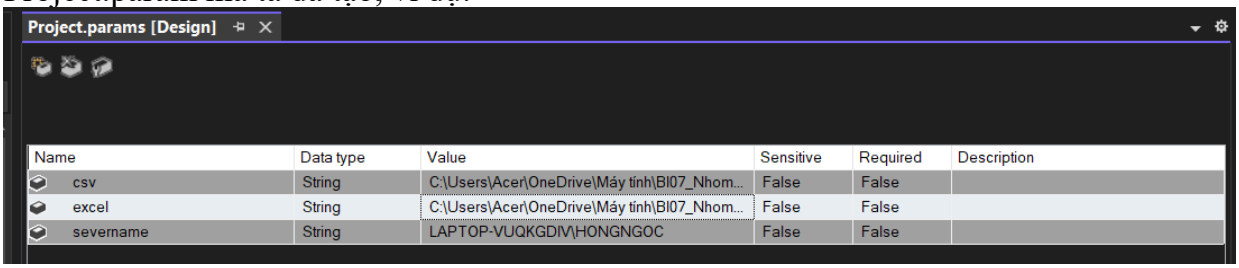


Name	Data type	Value	Sensitive	Required	Description
csv	String	D:\HK1_4\BI\BI.07_Project\NEW DATASET\film_sales.csv	False	False	
excel	String	D:\HK1_4\BI\BI.07_Project\NEW DATASET\film_sales.xlsx	False	False	
severname	String	ABC1	False	False	

Sau đó ta sẽ tiến hành set up thay đổi Connection tương tự với Supermarket_sales, Tuy nhiên ở file csv, ta sẽ thay đổi 1 ít ở mục property như sau:



Và như vậy, mỗi khi đổi thiết bị, ta chỉ cần thay đổi giá trị Value trong các biến của Project.param mà ta đã tạo, ví dụ:



Name	Data type	Value	Sensitive	Required	Description
csv	String	C:\Users\Acer\OneDrive\Máy tính\BI07_Nhom...	False	False	
excel	String	C:\Users\Acer\OneDrive\Máy tính\BI07_Nhom...	False	False	
severname	String	LAPTOP-VUQKGDIV\HONGNGOC	False	False	

Name	Data type	Value	Sensitive	Required	Description
csv	String	C:\Users\Admin\OneDrive\Máy tính\BI\BI.07_Project\NEW DATASET\film_sales.csv	False	False	
excel	String	C:\Users\Admin\OneDrive\Máy tính\BI\BI.07_Project\NEW DATASET\film_sales.xlsx	False	False	
severname	String	LAPTOP-CTI64POB\MSSQLSERVER22	False	False	

2. Set up MetaData

DATA_FLOW:

	ID	SOURCE_NAME	LSET	CET
1	1	film_sales.xls	2023-12-29 11:36:31.423	2023-12-29 11:36:31.423
2	2	film_sales.csv	2023-12-29 11:00:50.020	2023-12-29 11:00:50.020

Bảng DATA_FLOW dùng để lưu thông tin Source, SourceID và thời gian LSET CET mà hệ thống ETL đối với Source đó, mỗi lần ETL xong thì LSET và CET sẽ được cập nhật lại giá trị mới

DATA_STORE_INFO

	DBS_ID	DBS_NAME	DESCRIPTIONS
1	1	BI07_MOVIE_METADATA	Describes whole data information in system
2	2	BI07_MOVIE_STAGE	An internal data store used for transforming and...
3	3	BI07_MOVIE_NDS	An internal master data store in the form of nor...
4	4	BI07_MOVIE_DDS	A user-facing data store, in the form of relational...

Bảng DATA_STORE_INFO lưu ID và thông tin các database được sử dụng trong hệ thống

DATA_TABLE_INFO

	TABLE_ID	DBS_ID	TABLE_NAME	TABLE_TYPE	DESCRIPTIONS	UPDATEDATE
1	1	1	DATA_FLOW	METADATA	Store the timestamps of data source flow that E...	2023-12-29 11:36:31.000
2	2	1	DATA_TABLE_INFO	METADATA	Store table information	2023-12-20 23:19:24.713
3	3	1	DATA_STORE_INFO	METADATA	Store database information	2023-12-20 23:19:24.713
4	4	2	film	Stage	Store film data from source	2023-12-29 11:36:31.000
5	5	2	film_category	Stage	Store film category data from source	2023-12-29 11:36:31.000
6	6	2	cinema	Stage	Store Cinema data about the city from source	2023-12-29 11:36:31.000
7	7	2	city	Stage	Store City data from source	2023-12-29 11:36:31.000
8	8	2	film_sales_S1	Stage	Store film_sales_S1 data from source	2023-12-29 11:36:31.000
9	9	2	film_sales_S2	Stage	Store Ho Chi Minh city film sales data from source	2023-12-29 11:00:50.000
10	10	3	DATE_MASTER	NDS	Store date data from source	2023-12-28 16:59:28.333
11	11	3	film	Master tables(NDS)	Store film data	2023-12-29 11:36:31.000
12	12	3	film_cate	Master tables(NDS)	Store film category data	2023-12-29 11:36:31.000
13	13	3	city	Master tables(NDS)	Store city data	2023-12-29 11:36:31.000
14	14	3	cinema	Master tables(NDS)	Store cinema data	2023-12-29 11:36:31.000
15	15	3	film_sales	NDS	Store film sale data from source	2023-12-29 11:02:02.000
16	16	4	DATE_MASTER	dimension	Store date data from source	2023-12-28 16:59:49.950
17	17	4	film	dimension	Store film Dimension	2023-12-22 11:49:12.000
18	18	4	film_cate	dimension	Store film category Dimension	2023-12-29 11:02:02.000
19	19	4	city	dimension	Store city Dimension	2023-12-28 20:43:18.000
20	20	4	cinema	dimension	Store cinema Dimension	2023-12-22 11:49:12.000

Bảng DATA_TABLE_INFO lưu thông tin các table có trong nguyên hệ thống, DBS_ID chứa Table đó và ngày mà Table được UPDATE. Sau mỗi lần ETL, nếu table được cập nhật thì UPDATEDATE của table đó sẽ thay đổi theo giá trị ngày mới.

COLUMN_INFO

	COLUMN_ID	TABLE_ID	COLUMN_NAME	COLUMN_TYPE	DESCRIPTIONS	SAMPLE_VALUES	SOURCE_COLUMN_KEY_ID
1	1	5	FilmCategoryID	nvarchar	store Film Category Key	FC1	NULL
2	2	5	Film Category	nvarchar	store Film Category	Action	NULL
3	3	4	FilmID	nvarchar	store film ID	F7	NULL
4	4	4	Film Name	nvarchar	store film name	Wonka	NULL
5	5	4	FilmCategoryID	nvarchar	store film category ID	FC1	1
6	6	4	Price	int	store film price	25000	NULL
7	7	7	CityID	nvarchar	store City Key	HCM	NULL
8	8	7	CityName	nvarchar	store City Name	Ho Chi Minh	NULL
9	9	6	CinemaID	nvarchar	store Cinema Key	C1	NULL
10	10	6	Cinema Name	nvarchar	store Cinema Name	Galaxy Cinema	NULL
11	11	6	CityID	nvarchar	store City Key	HCM	7
12	12	6	Capacity	int	store Capacity	1800	NULL
13	13	6	Standard_Seats_price	int	store Standard Seats Price	30000	NULL
14	14	6	VIP_Seats_price	int	store VIP Seats Price	50000	NULL
15	15	8	RevenueID	nvarchar	store revenue ID	RV1	NULL
16	16	8	FilmID	nvarchar	store film ID	F2	3
17	17	8	Cinema Name	nvarchar	store cinema name	Mega Cinema	NULL
18	18	8	total_sales	int	store total sale of revenue	4160000	NULL
19	19	8	tickets_sold	int	store number of ticket sold	208	NULL
20	20	8	Standard_Seats_sold	int	store number of standard seat sold	101	NULL

Bảng COLUMN_INFO lưu thông tin từng cột trong các bảng, TABLE_ID chứa cột đó và ID của cột là giá trị tham chiếu đến nếu có

DATA_FIRE

	ID	COLUMN_ID	RULE
1	1	43	([Day]>0 && [Day]<32 && [Month]==1,3,5,5,7,8,10,12)|...
2	2	44	[Month]>=1 || [Month]<=12 ? [Month] : 1
3	3	45	[Year]<1 || [Year]>Year(GETDATE()) ? Year(GETDAT...
4	4	46	[Quarter] < 1 || [Quarter] > 4 ? 1 : [Quarter]
5	5	56	[Price] < 0 ? 0 : [Price]
6	6	77	[total_sales] < 0 ? 0 : [total_sales]
7	7	78	[tickets_sold] < 0 ? 0 : [tickets_sold]
8	8	81	[tickets_out] < 0 || [tickets_out] > [tickets_sold] ? 0 : [tic...
9	9	82	[Standard_Seats_out] < 0 || [Standard_Seats_out] > [...
10	10	83	[VIP_Seats_out] < 0 || [VIP_Seats_out] > [VIP_Seats_...
11	11	84	[ticket_use] < 0 || [ticket_use] > [tickets_sold] ? 0 : [tick...
12	12	85	[occur_perc] < 0 || [occur_perc] > 100 ? 0 : [occur_perc]

Bảng DATA_FIRE lưu thông tin biểu thức tính toán được áp dụng cho những cột trong NDS và ID của cột được áp dụng công thức đó

3. Thông tin tạo NDS

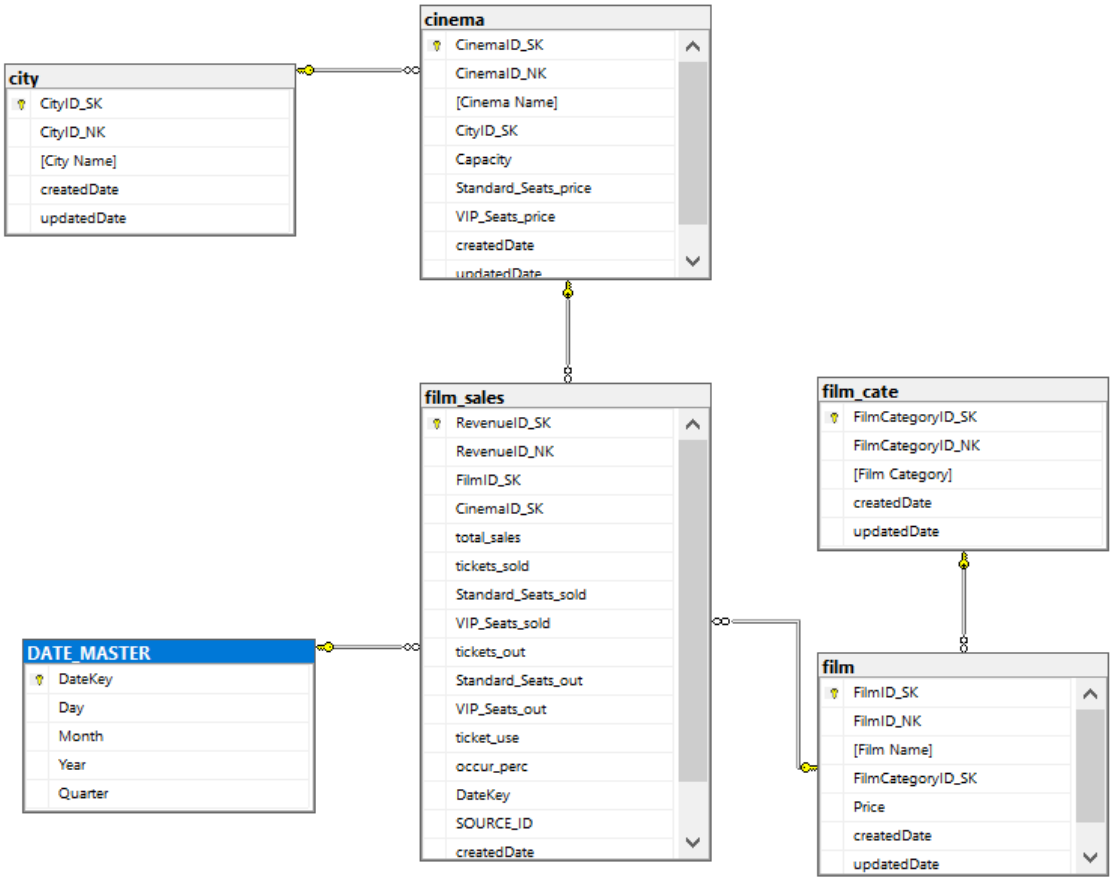
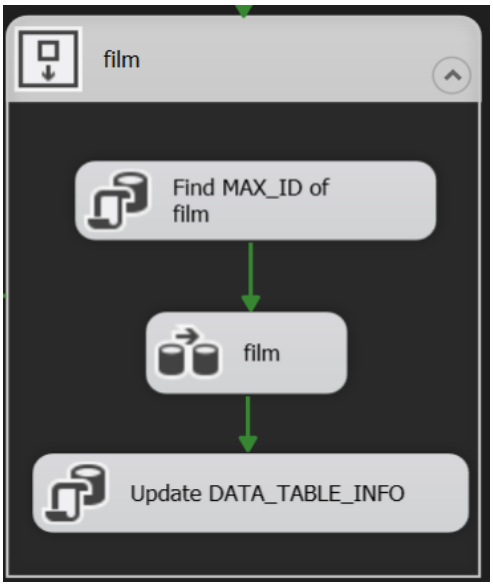


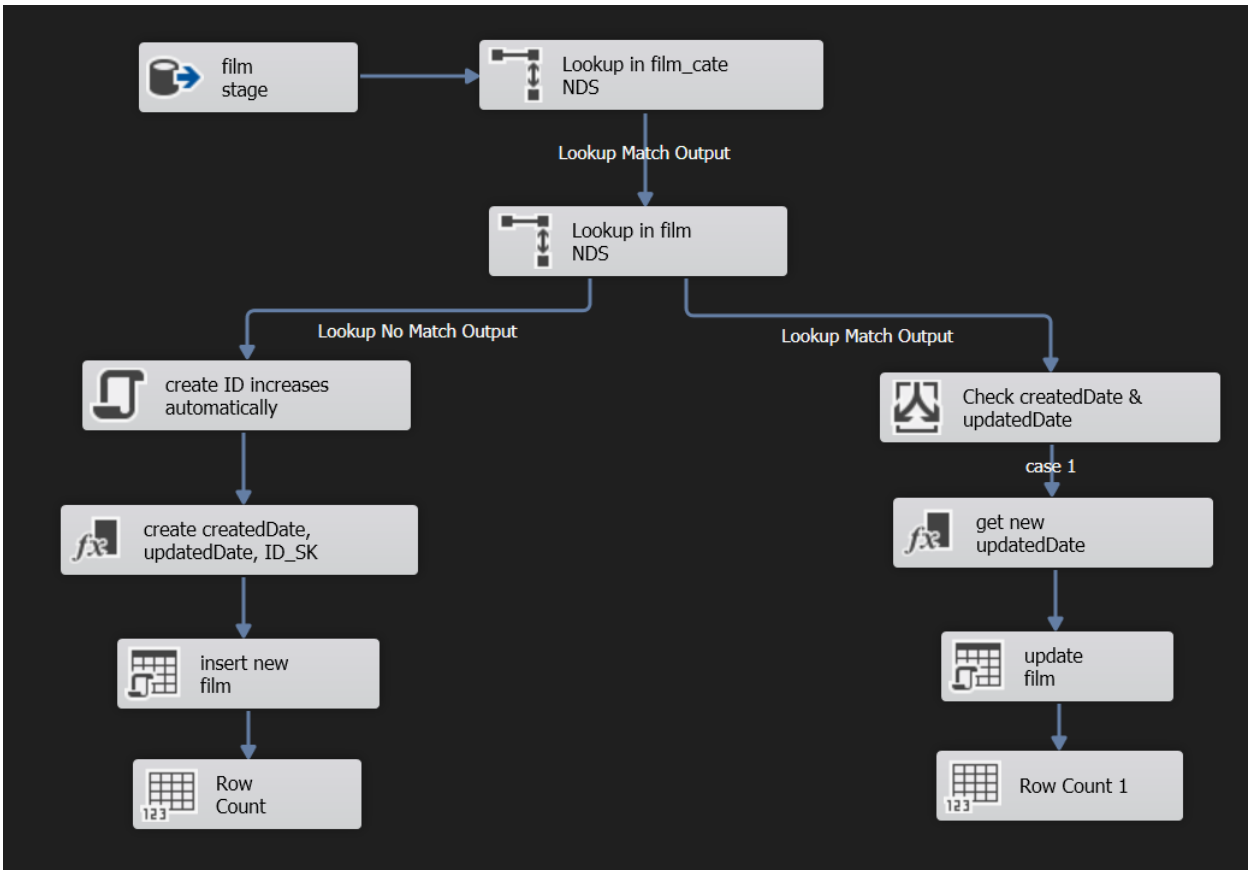
Diagram Film_Sales NDS

- film (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
FilmID_SK		int	1
FilmID_NK	Film Stage (FilmID)	nvarchar	F2
Film Name	Film Stage (Film Name)	nvarchar	Wonka
FilmCategoryID_SK	FilmCategory NDS (FilmCategoryID_SK)	int	1
Price	Film Stage (Price)	int	250

CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36



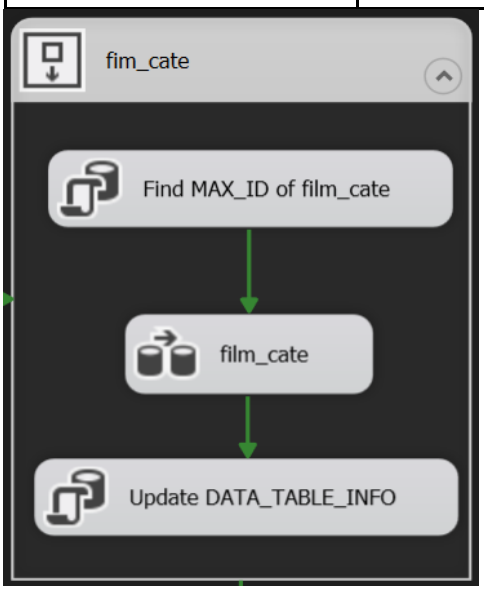


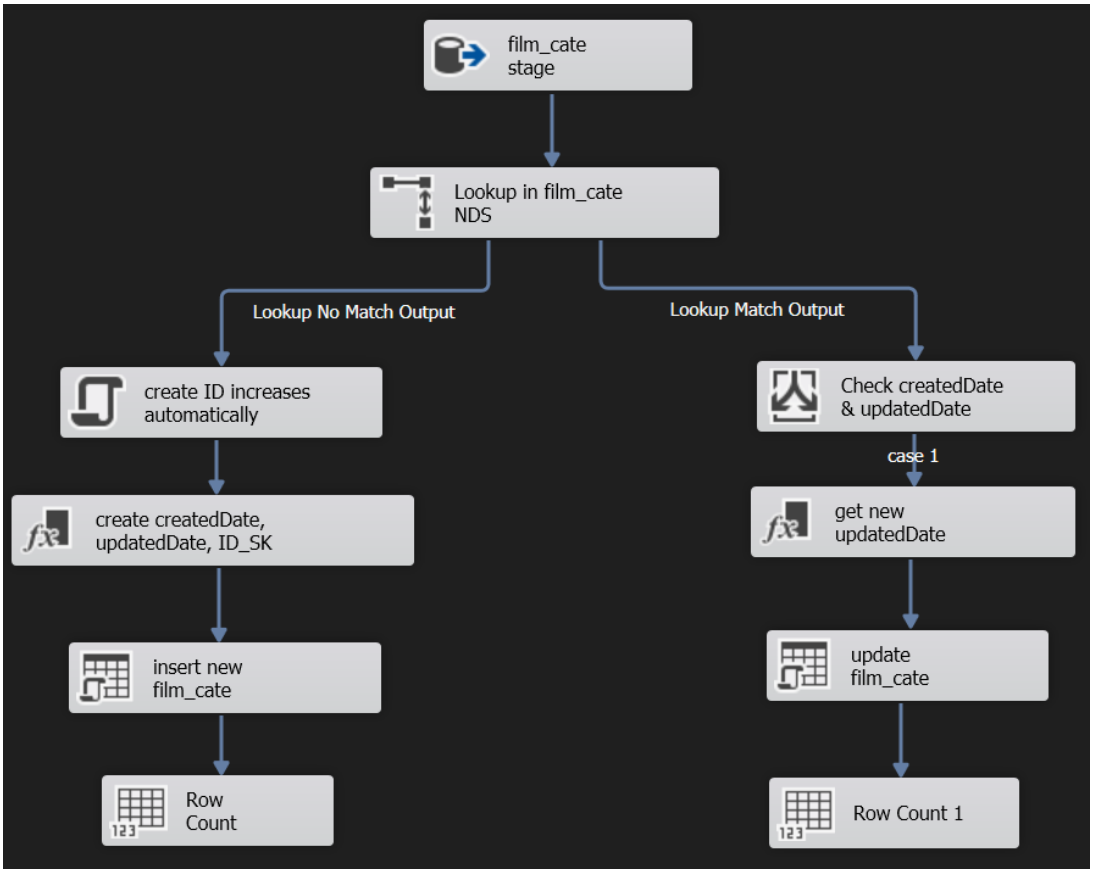
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột FilmID_SK từ bảng film trong NDS	Execute SQL Task
1.2		Tạo một Data Flow cho việc ETL bảng film từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ film Stage	OLEDB Source
2.2	-	Lookup giá trị FilmCategoryID trong bảng film_cate NDS	Lookup
2.2.1	no match	Bỏ qua row và không tiếp tục ETL	
2.2.2	match	Lookup giá trị FilmID trong bảng film NDS	Lookup
2.2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của Film vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
		Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng product	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2.2	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
		Tạo createdDate & updatedDate = CET, và FilmID_SK = MAX_ID + ID	Derived Column
		Thực hiện thêm row vào film	OLE DB Command
		Đếm số dòng vừa insert và lưu vào	Row Count

		Count_Row	
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng film NDS trong METADATA	Execute SQL Task

- film_cate (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
FilmCategoryID_SK		int	1
FilmCategoryID_NK	film_cate Stage (FilmCategoryID)	nvarchar	FC1
Film Category	film_cate Stage (FilmCategory)	nvarchar	Action
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36



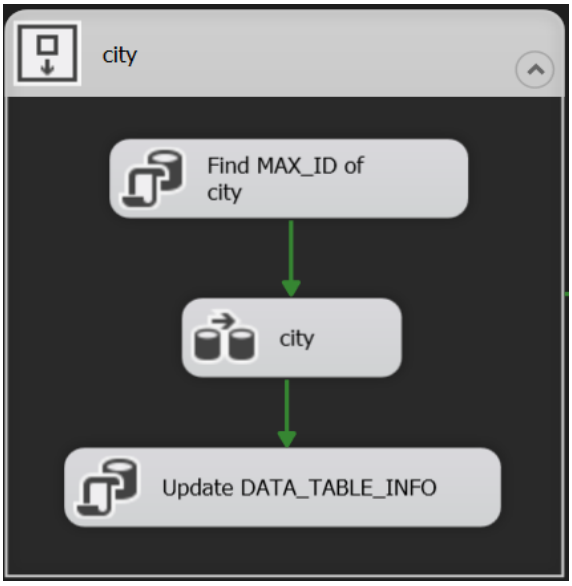


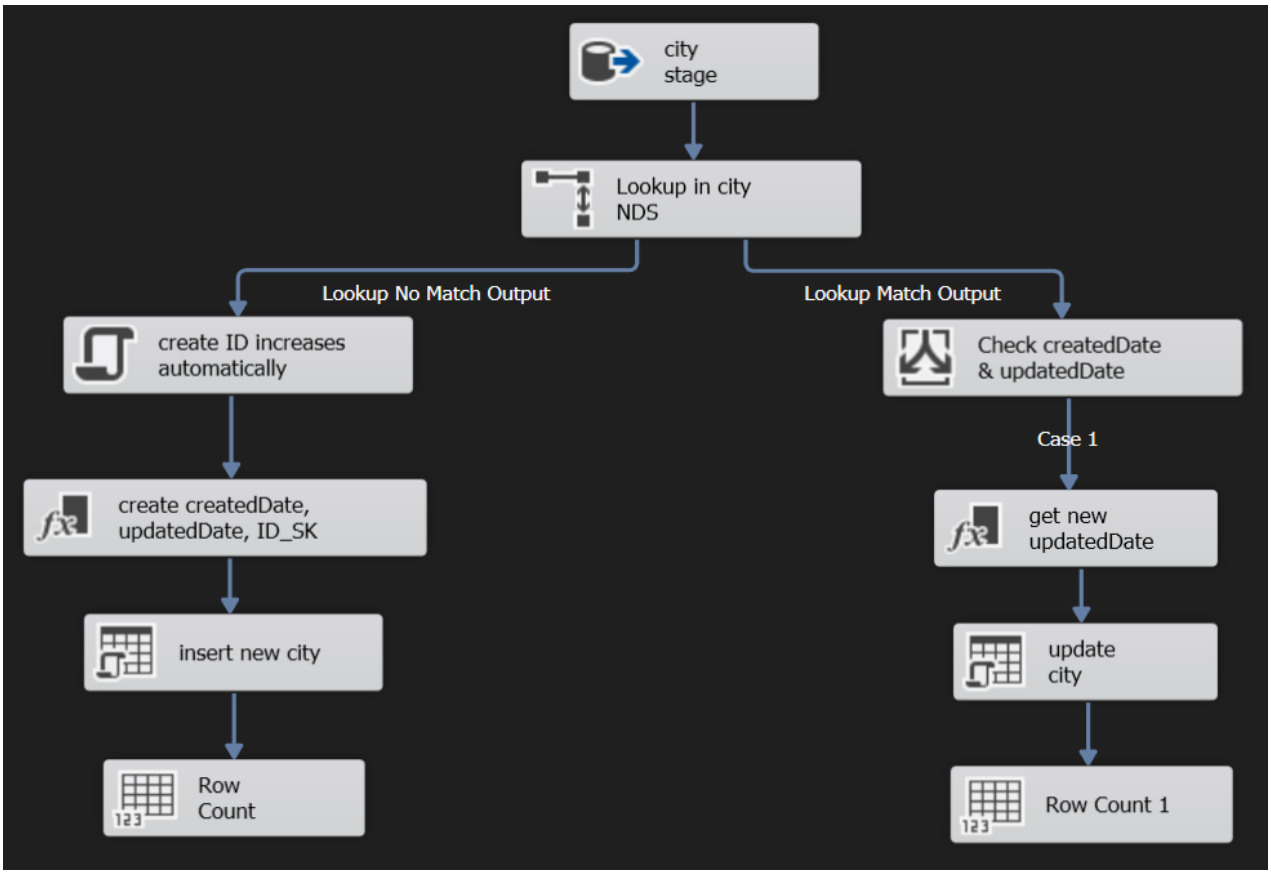
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột FilmID_SK từ bảng film_cate trong NDS	Execute SQL Task
1.2		Tạo một Data Flow cho việc ETL bảng film_cate từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ film_cate Stage	OLEDB Source
2.2.	-	Lookup giá trị FilmCategoryID trong bảng film_cate NDS	Lookup
2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của FilmCategory vừa tìm	Conditional Split

		được xem có thoả điều kiện ETL không	
	match	Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng film_cate	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
		Tạo createdDate & updatedDate = CET, và FilmCategoryID_SK = MAX_ID + ID	Derived Column
		Thực hiện thêm row vào film_cate	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng film_cate NDS trong METADATA	Execute SQL Task

- City (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
CityID_SK		int	1
CityID_NK	city Stage (CityID)	nvarchar	HCM
CityName	city Stage (CityName)	nvarchar	Ho Chi Minh
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36



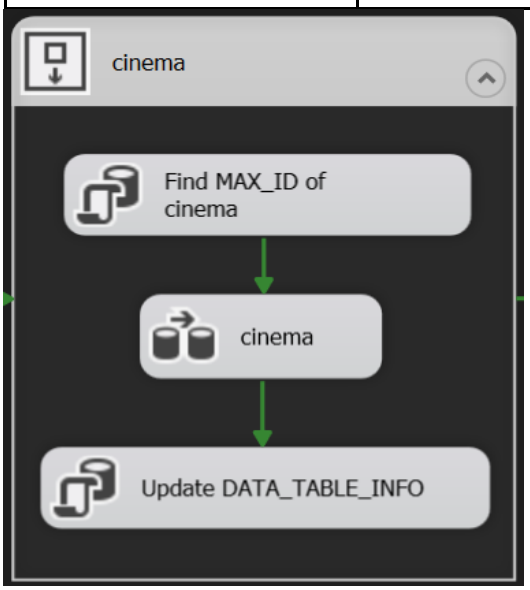


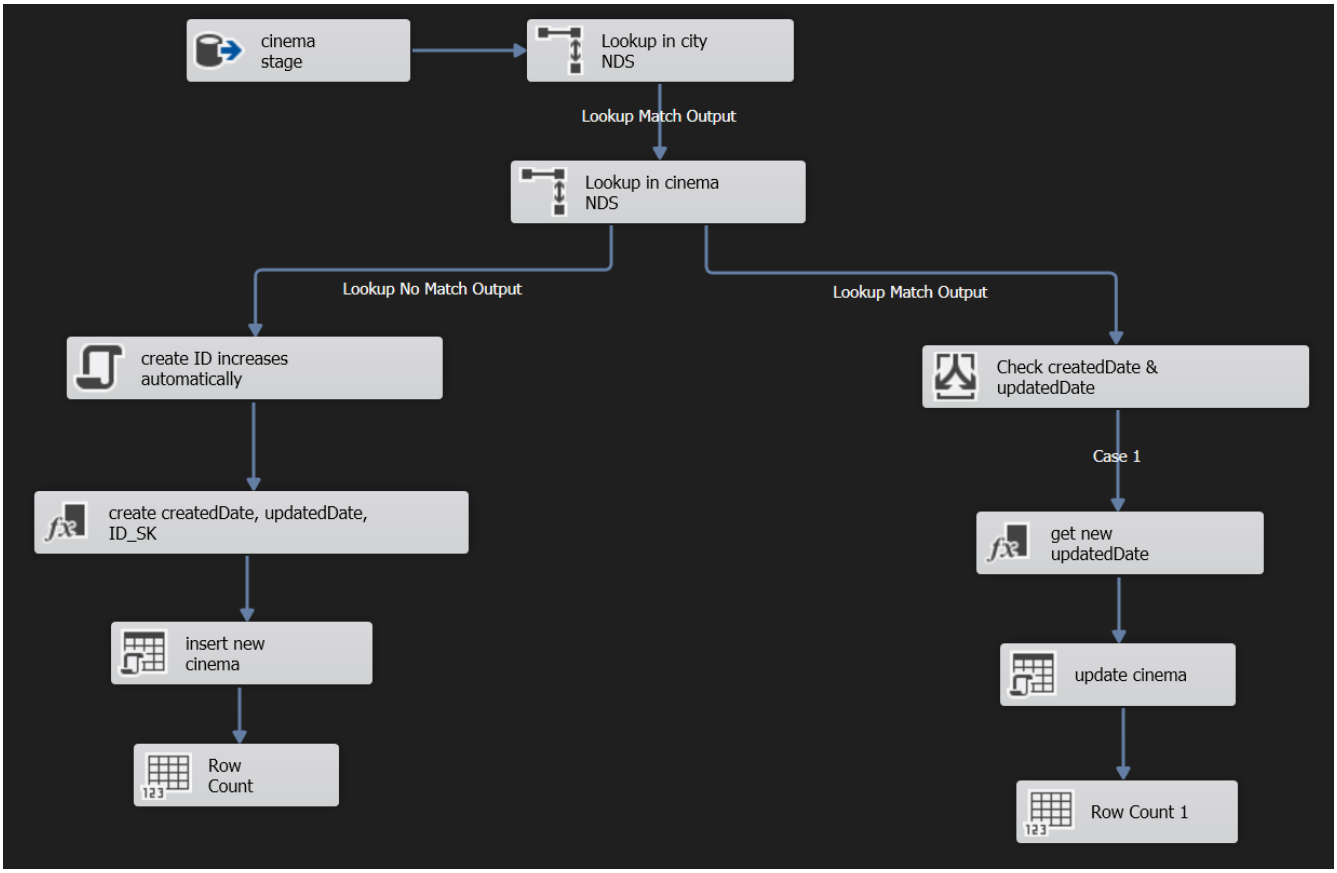
STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			

1.1		Tìm MAX_ID của cột CityID_SK từ bảng city trong NDS	Execute SQL Task
1.2		Tạo một Data Flow cho việc ETL bảng city từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ city Stage	OLEDB Source
2.2.	-	Lookup giá trị CityID trong bảng city NDS	Lookup
2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của City vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
		Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng city	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
		Tạo createdDate & updatedDate = CET, và CityID_SK = MAX_ID + ID	Derived Column
		Thực hiện thêm row vào city	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng city NDS trong METADATA	Execute SQL Task

- cinema (Master)

Tên cột	Nguồn	Kiểu DL	Value mẫu
CinemaID_SK		int	1
CinemaID_NK	cinema Stage (CinemaID)	nvarchar	C1
Cinema Name	cinema Stage (Cinema Name)	nvarchar	Galaxy Cinema
CityID_SK	city NDS (CityID_SK)	int	1
Capacity	cinema Stage (Capacity)	int	1800
Standard_Seats_price	cinema Stage (Standard_Seats_price)	int	30000
VIP_Seats_price	cinema Stage (VIP_Seats_price)	int	50000
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





STT	Trường hợp	Mô tả	SSIS tool
1. Execute SQL Task			
1.1		Tìm MAX_ID của cột CinemaID_SK từ bảng cinema trong NDS	Execute SQL Task
1.2		Tạo một Data Flow cho việc ETL bảng cinema từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ cinema Stage	OLEDB Source
2.2	-	Lookup giá trị CityID trong bảng cinema NDS	Lookup
2.2.1	no match	Bỏ qua row và không tiếp tục ETL	

2.2.2	match	Lookup giá trị CinemaID trong bảng cinema NDS	Lookup
2.2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của Cinema vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
		Tạo ngày cập nhật mới bằng CET	Derived Column
		Thực hiện cập nhật dữ liệu mới vào bảng cinema	OLE DB Command
		Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2.2	no match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
		Tạo createdDate & updatedDate = CET, và CinemaID_SK = MAX_ID + ID	Derived Column
		Thực hiện thêm row vào cinema	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng cinema NDS trong METADATA	Execute SQL Task

- DATE_MASTER

DATE_MASTER lưu trữ ngày tháng năm

Không thực hiện ETL trên bảng này vì đã thiết lập sẵn trên database

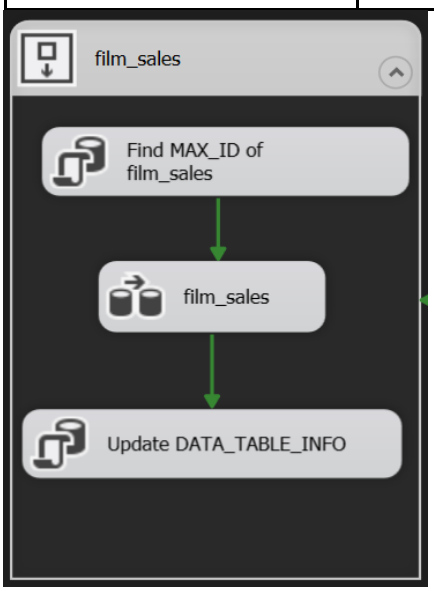
Tên cột	Nguồn	Kiểu DL	Value mẫu
DateKey		varchar(8)	20232412
Day		int	24

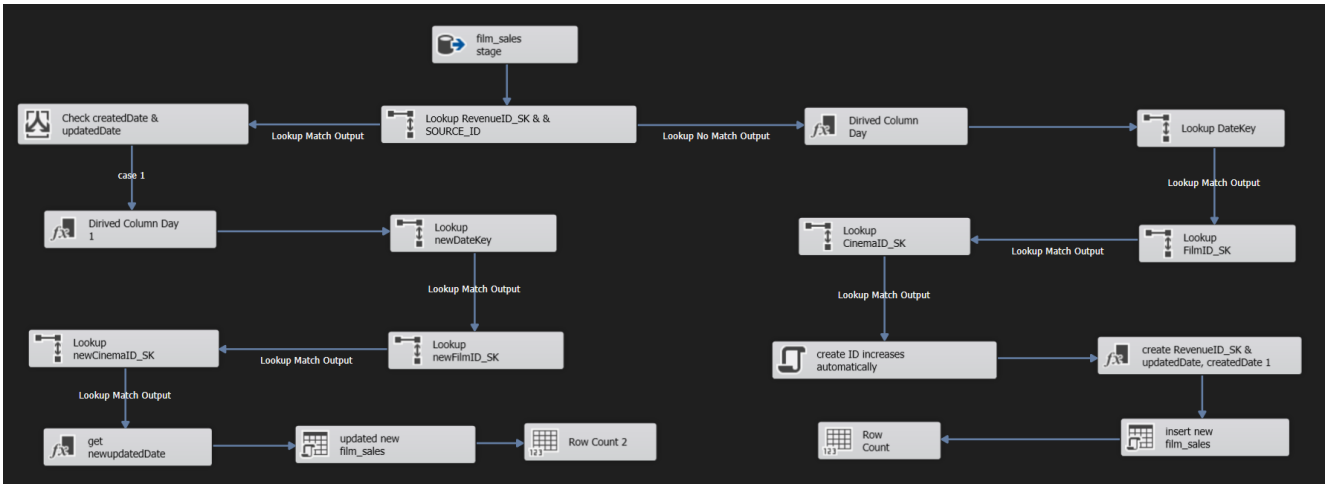
Month		int	12
Year		int	2023
Quarter		int	4

- film_sales (Transaction)

Tên cột	Nguồn	Kiểu DL	Value mẫu
RevenueID_SK		int	1
RevenueID_NK	film_sales Stage (RevenueID)	nvarchar	RV1
FilmID_SK	film NDS (FilmID_SK)	int	1
CinemaID_SK	cinema NDS (CinemaID_SK)	int	1
total_sales	film_sales Stage (total_sales)	int	4160000
tickets_sold	film_sales Stage (tickets_sold)	int	208
Standard_Seats_sold	film_sales Stage (Standard_Seats_sold)	int	517
VIP_Seats_sold	film_sales Stage (VIP_Seats_sold)	int	648
tickets_out	film_sales Stage (tickets_out)	int	0
Standard_Seats_out	film_sales Stage (Standard_Seats_out)	int	0
VIP_Seats_out	film_sales Stage (VIP_Seats_out)	int	0
ticket_use	film_sales Stage (ticket_use)	int	1161

occur_perc	film_sales Stage (occur_perc)	float	64.5
DateKey	DATE_MASTER NDS (DateKey)	nvarchar	20190105
SOURCE_ID		int	1
CreateDate		Datetime	08/12/23 12:36
UpdateDate		Datetime	08/12/23 12:36





STT	Trường hợp	Mô tả	SSIS tool

1. Execute SQL Task			
1.1		Tìm MAX_ID của cột RevenueID_SK từ bảng cinema trong NDS	Execute SQL Task
1.2		Tạo một Data Flow cho việc ETL bảng film_sales từ Stage vào NDS	Data Flow Task
2. Data Flow			
2.1	-	Lấy dữ liệu từ film_sales Stage	OLEDB Source
2.2	-	Lookup giá trị RevenueID, SOURCE_ID trong bảng film_sales NDS	Lookup
2.2.1	match	tiến hành kiểm tra createdDate & updatedDate của film_sales vừa tìm được xem có thoả điều kiện ETL không	Conditional Split
	thỏa điều kiện	Lấy giá trị ngày, tháng, năm từ Date và lưu vào 3 cột Day, Month, Year	Derived Column
	match	Lookup giá trị Day, Month, Year có tồn tại trong DATE_MASTER không	Lookup
	match	Lookup giá trị FilmID trong bảng city NDS và lấy newFilmID_SK	Lookup
	match	Lookup giá trị CinemaID trong bảng cinema NDS và lấy newCinemaID_SK	Lookup
	match	Tạo ngày cập nhật mới bằng CET	Derived Column
	-	Thực hiện cập nhật dữ liệu mới vào bảng film_sales	OLE DB Command
	-	Đếm số dòng vừa cập nhật và lưu vào Count_Row	Row Count
2.2.2	no match	Lấy giá trị ngày, tháng, năm từ Date và lưu vào 3 cột Day, Month, Year	Derived Column
		Lookup giá trị Day, Month, Year có tồn tại trong DATE_MASTER không	Lookup

	match	Lookup giá trị FilmID trong bảng city NDS và lấy FilmID_SK	Lookup
	match	Lookup giá trị CinemaID trong bảng cinema NDS và lấy CinemaID_SK	Lookup
	match	Tạo biến ID tự tăng theo mỗi dòng được insert vào	Script Component
	-	Tạo createdDate & updatedDate = CET, và RevenueID_SK = MAX_ID + ID	Derived Column
	-	Thực hiện insert dữ liệu vào film_sales	OLE DB Command
	-	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
3. Execute SQL Task			
3.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng film_sales NDS trong METADATA	Execute SQL Task

4. Thông tin tạo DDS:

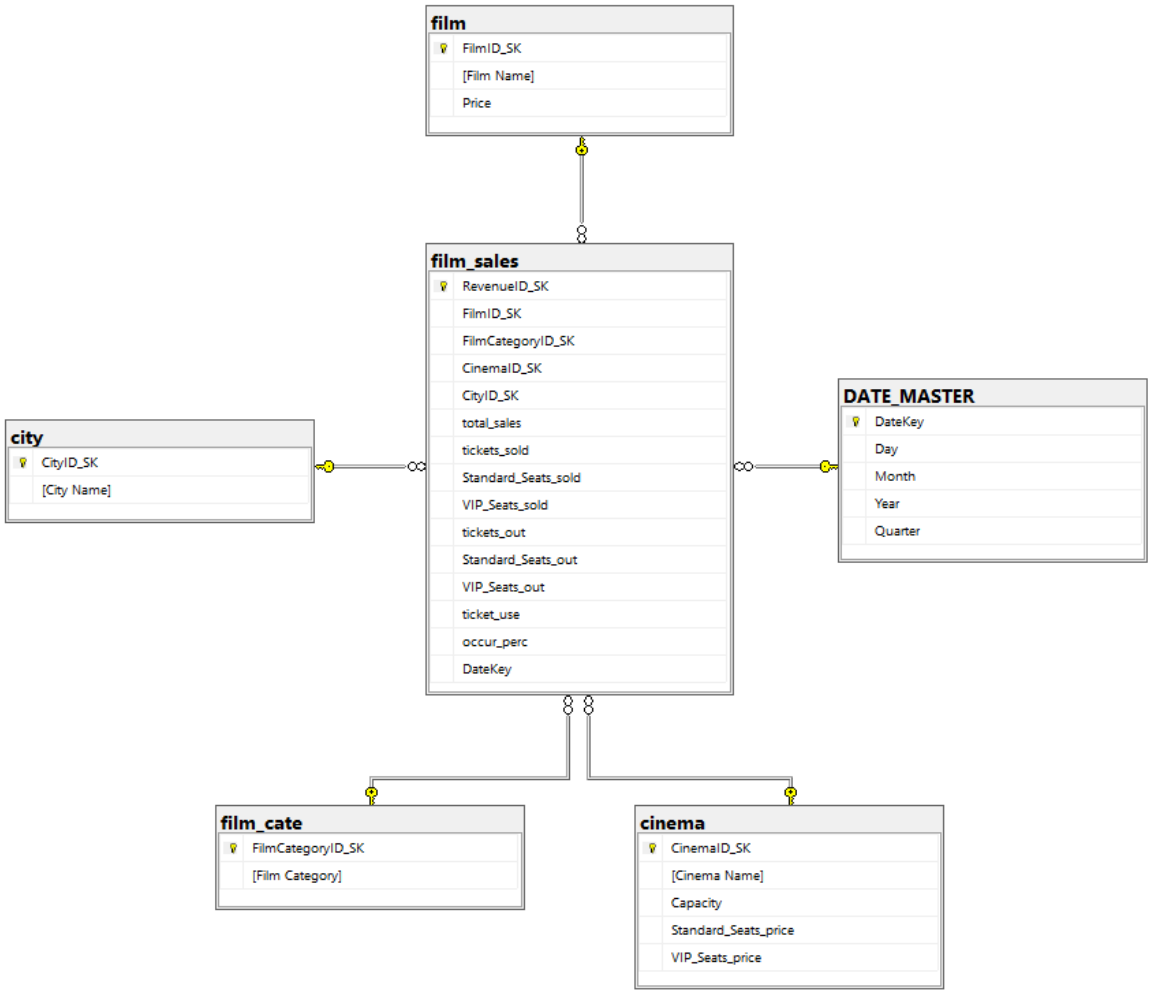
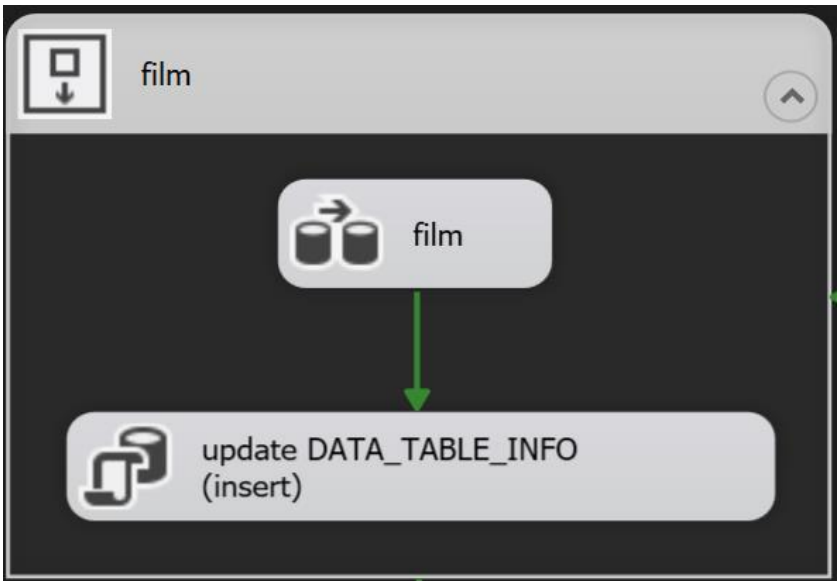
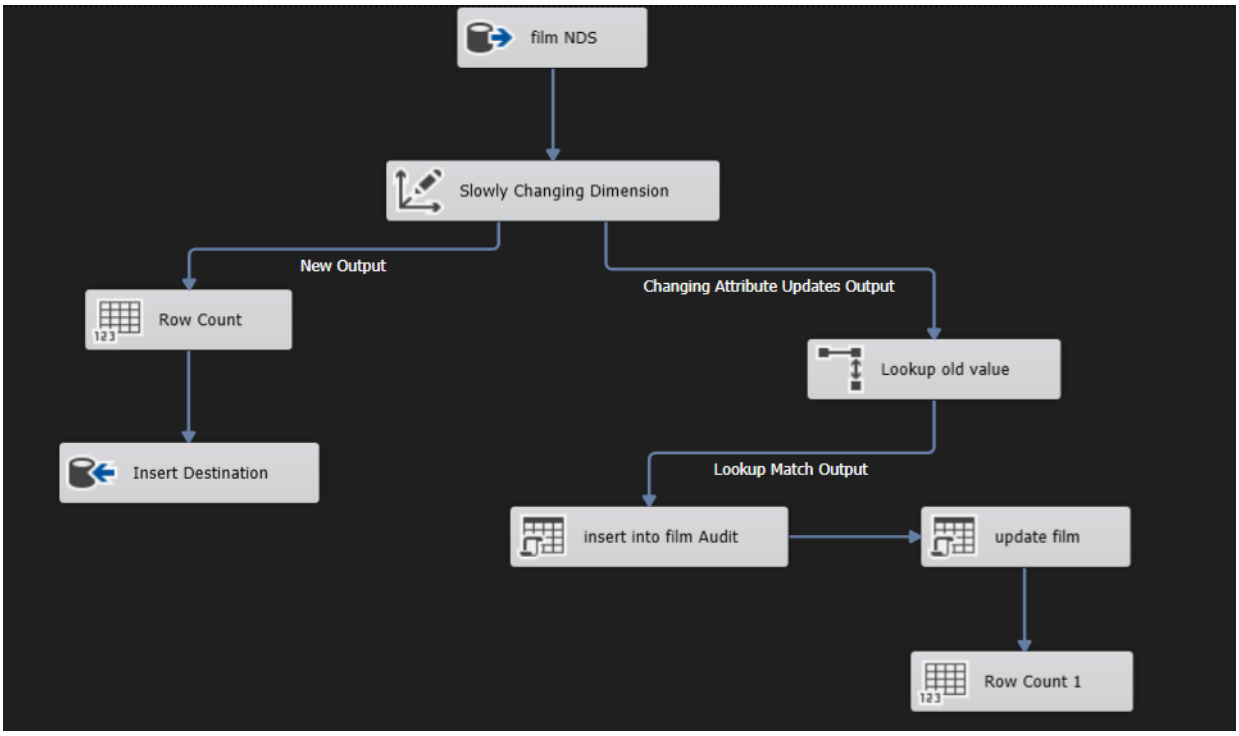


Diagram Film DDS

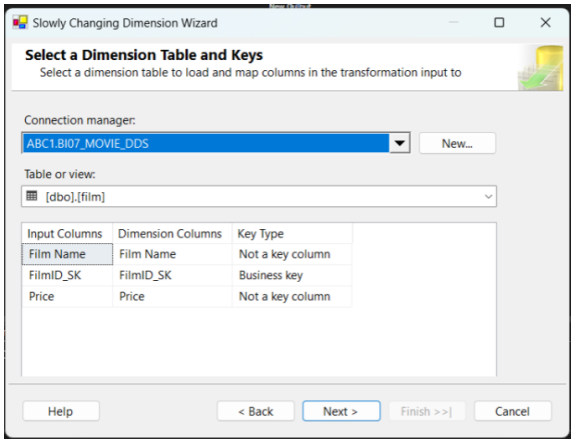
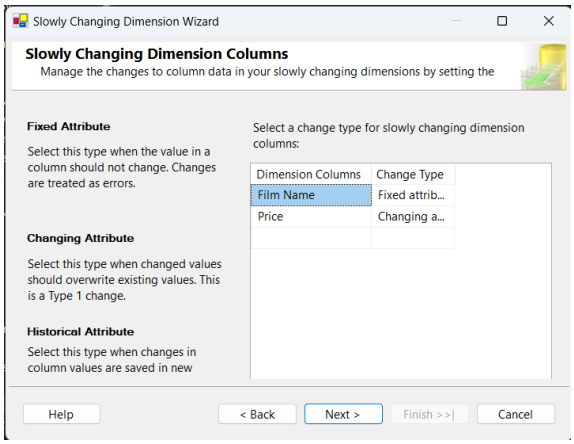
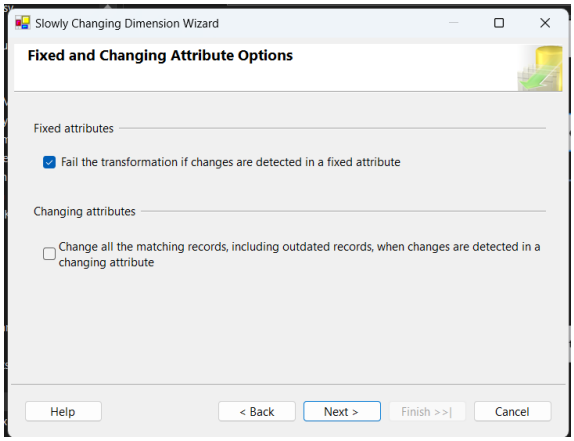
- film (Dimension)

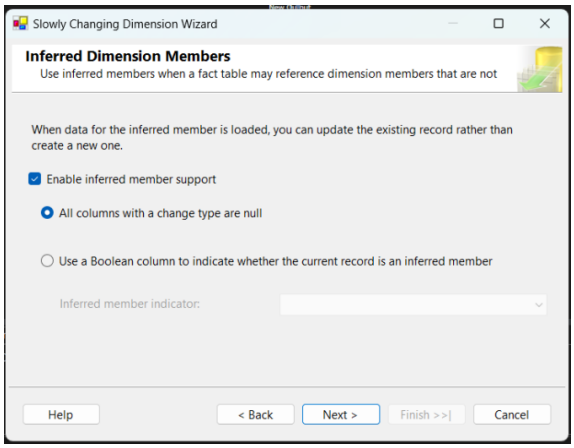
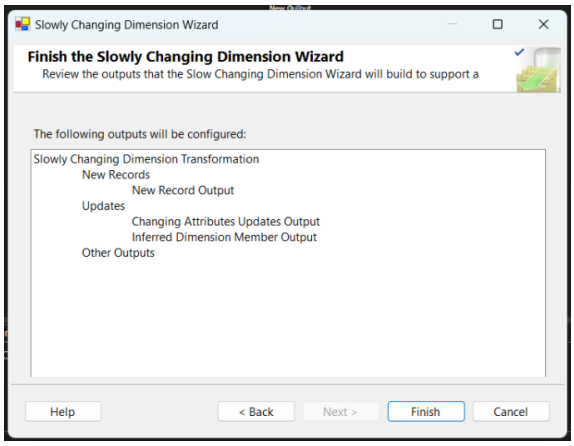
Tên cột	Nguồn	Kiểu DL	Value mẫu
FilmID_SK		int	1
Film Name	Film Stage (Film Name)	nvarchar	Wonka
Price	Film Stage (Price)	int	250





STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source

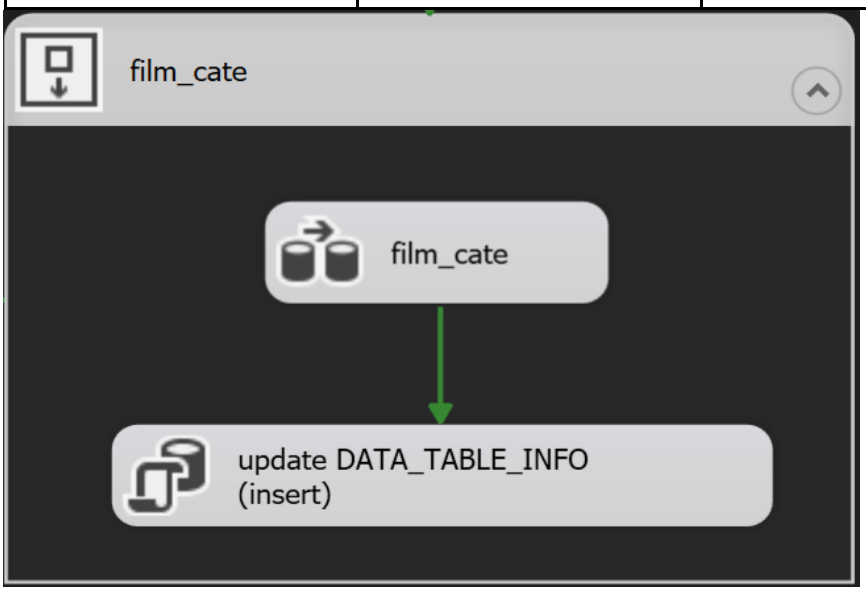
1.2	-	Tạo SCD	
	-	 Slowly Changing Dimension Wizard Select a Dimension Table and Keys Select a dimension table to load and map columns in the transformation input to Connection manager: ABC1.BI07_MOVIE_DDS New... Table or view: [dbo].[film] Input Columns Dimension Columns Key Type Film Name Film Name Not a key column FilmID_SK FilmID_SK Business key Price Price Not a key column Help < Back Next > Finish >>| Cancel	
	-	 Slowly Changing Dimension Wizard Slowly Changing Dimension Columns Manage the changes to column data in your slowly changing dimensions by setting the Fixed Attribute Select this type when the value in a column should not change. Changes are treated as errors. Changing Attribute Select this type when changed values should overwrite existing values. This is a Type 1 change. Historical Attribute Select this type when changes in column values are saved in new Select a change type for slowly changing dimension columns: Dimension Columns Change Type Film Name Fixed attrib... Price Changing a... Help < Back Next > Finish >>| Cancel	
	-	 Slowly Changing Dimension Wizard Fixed and Changing Attribute Options Fixed attributes Fail the transformation if changes are detected in a fixed attribute Changing attributes Change all the matching records, including outdated records, when changes are detected in a changing attribute Help < Back Next > Finish >>| Cancel	

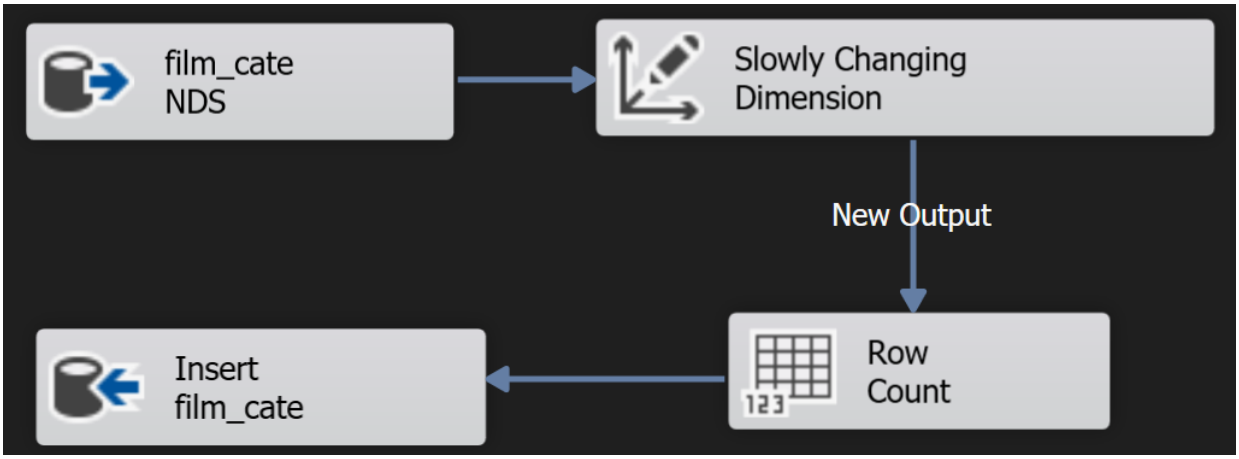
	-		
	-		
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào film DDS	OLEDB Destination
	Thay đổi giá trị cũ	Lookup các dòng dữ liệu cũ trong bảng DDS	LookUp
	match	Insert dòng dữ liệu vào bảng film_Audit	OLE DB Command
		Update dòng dữ liệu mới vào bảng film DDS	OLE DB Command
		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
2. Execute SQL Task			

2.1	-	Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng film DDS trong METADATA	Execute SQL Task

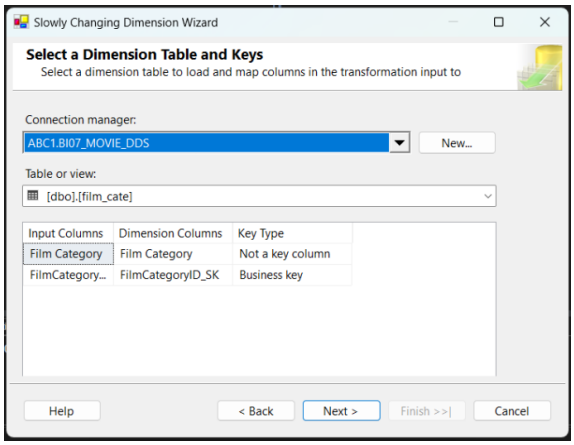
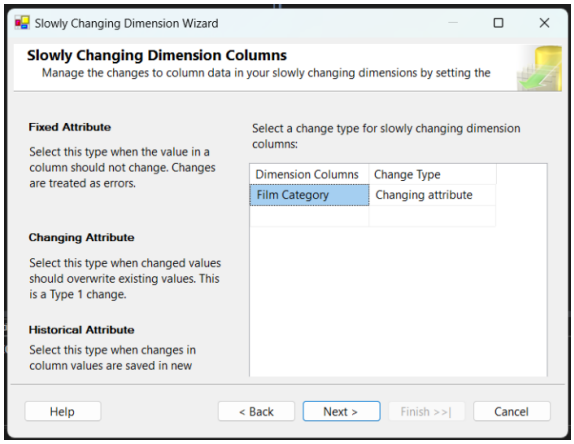
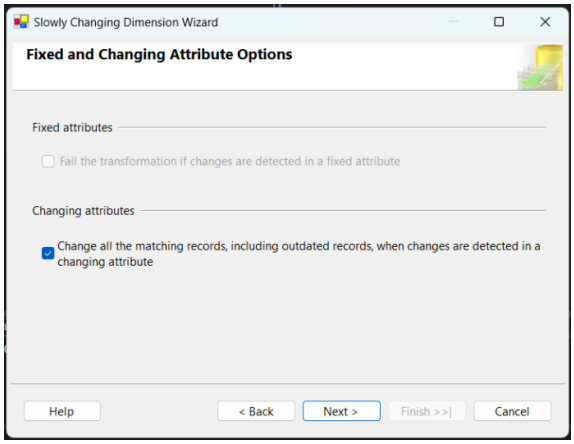
- film_cate (Dimension)

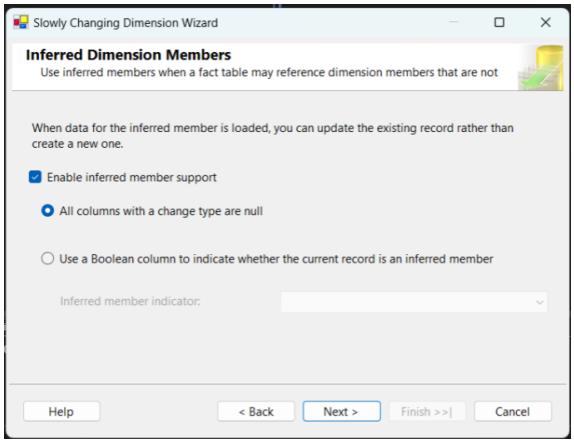
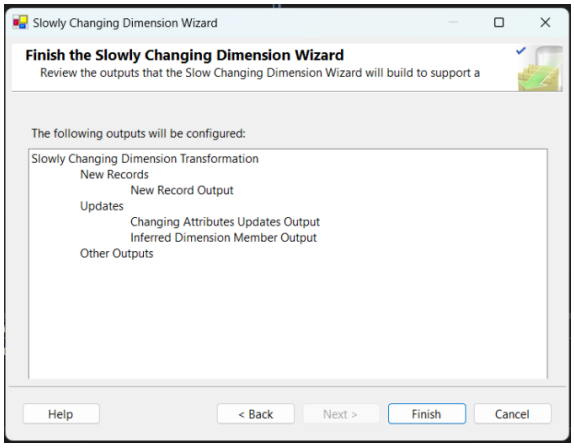
Tên cột	Nguồn	Kiểu DL	Value mẫu
FilmCategoryID_SK		int	1
Film Category	film_cate Stage (FilmCategory)	nvarchar	Action



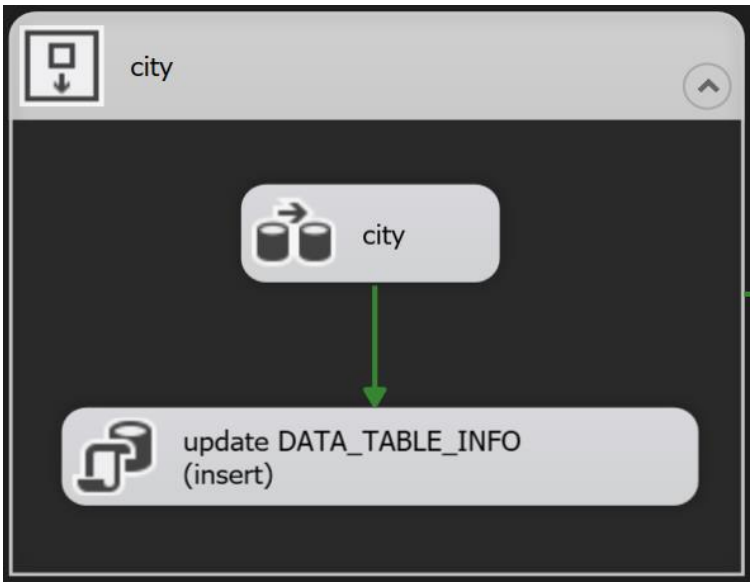


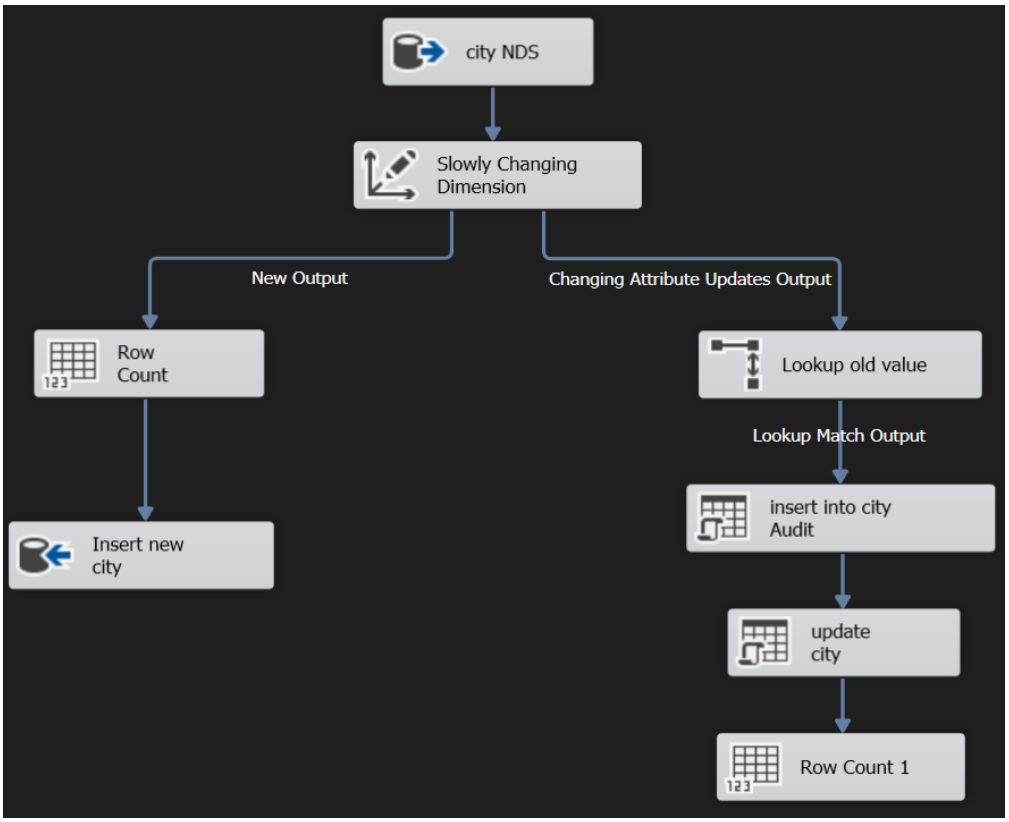
STT	Trường hợp	Mô tả	SSIS tool

1. Data Flow Task			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source
1.2	-	Tạo SCD	
	-		
	-		
	-		

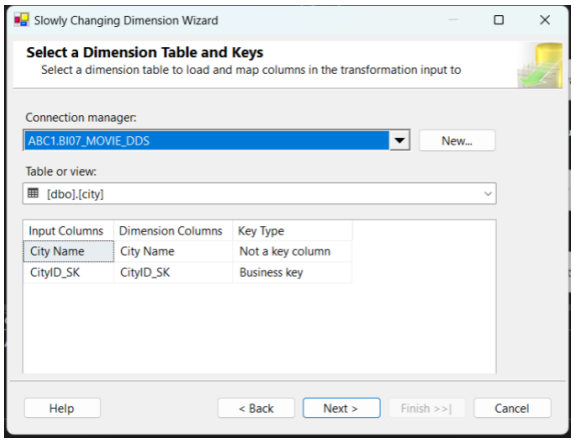
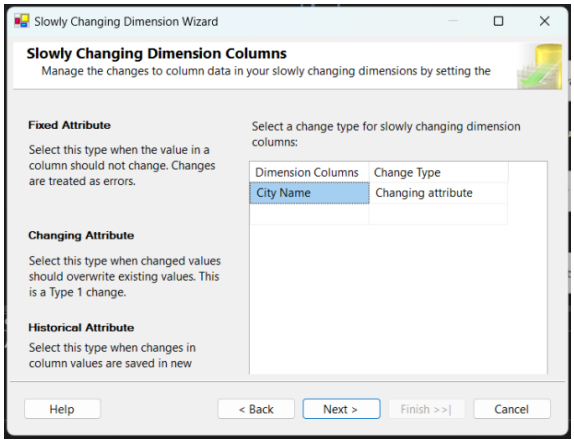
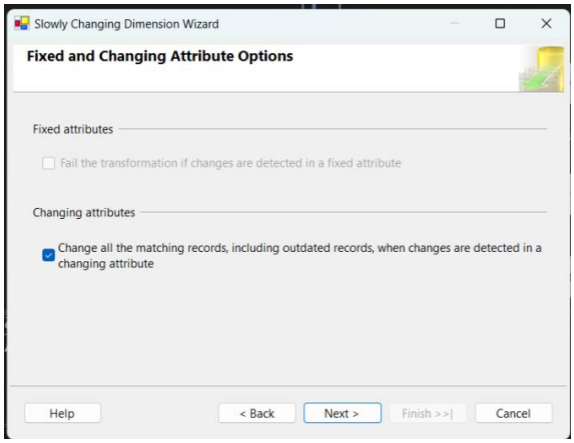
	-		
	-		
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào film DDS	OLEDB Destination
2. Execute SQL Task			
2.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng film DDS trong METADATA	Execute SQL Task

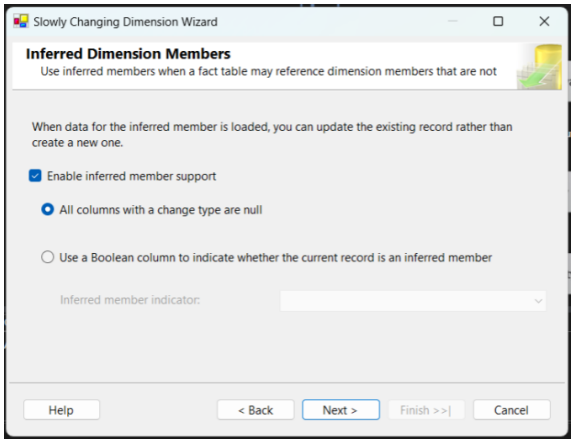
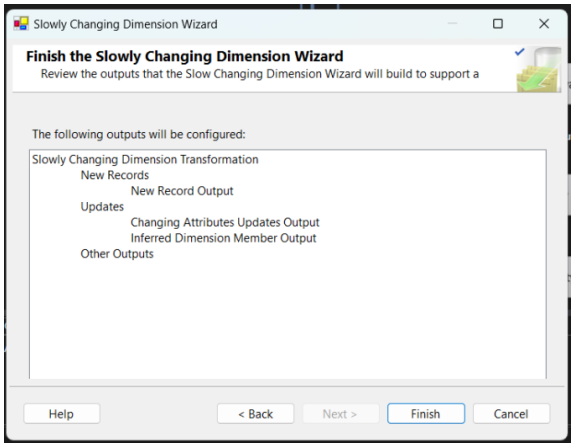
- city (Dimension)





STT	Trường hợp	Mô tả	SSIS tool

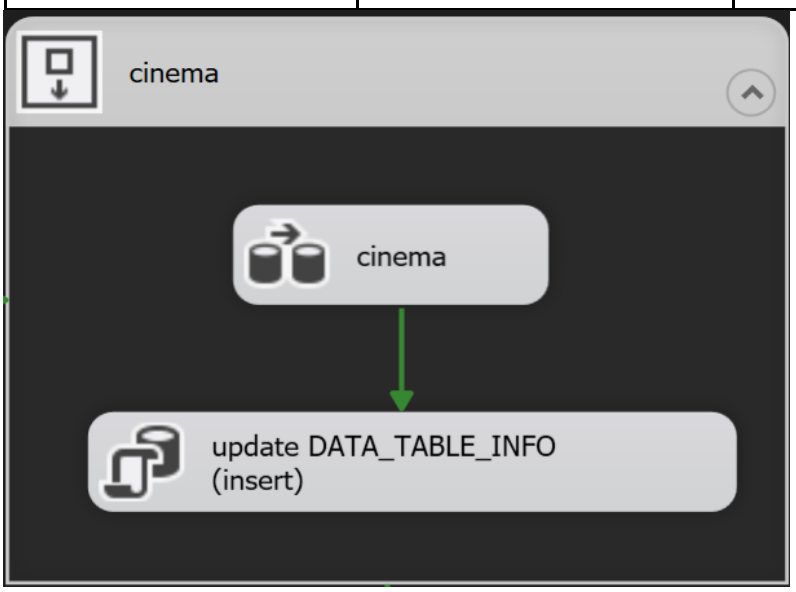
1. **Data Flow Task**			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source
1.2		Tạo SCD	
			
			
			

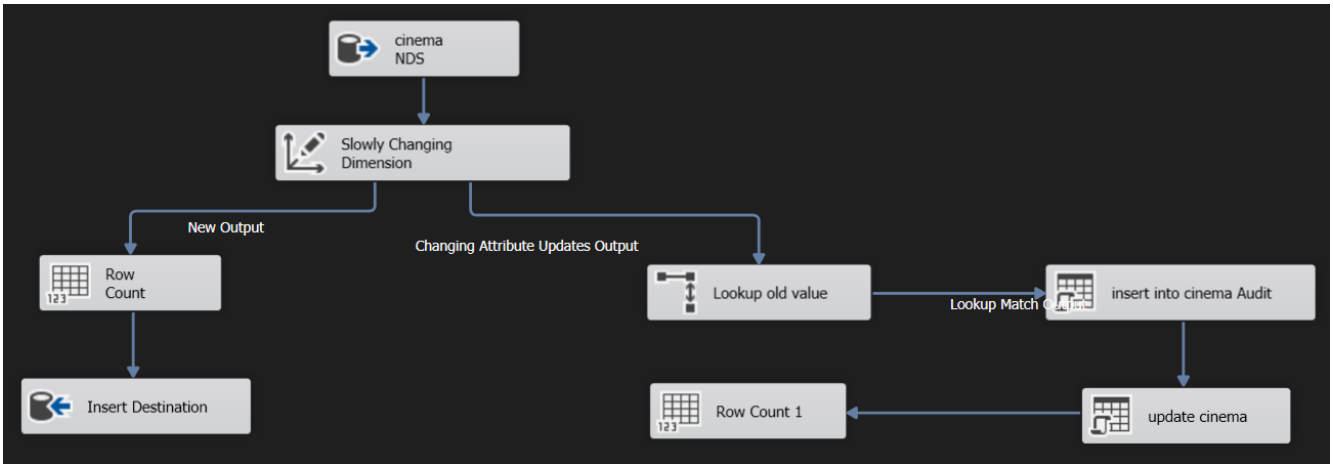
			
			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào city DDS	OLEDB Destination
	Thay đổi giá trị cũ	Lookup các dòng dữ liệu cũ trong bảng DDS	LookUp
	match	Insert dòng dữ liệu vào bảng city_Audit	OLE DB Command
		Update dòng dữ liệu mới vào bảng city DDS	OLE DB Command
		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
2. Execute SQL Task			

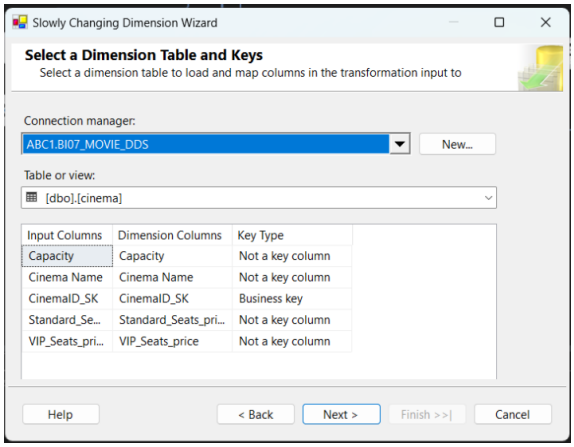
2.1		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng city DDS trong METADATA	Execute SQL Task

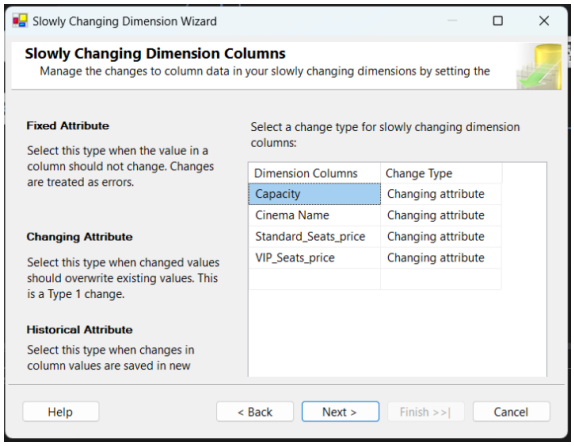
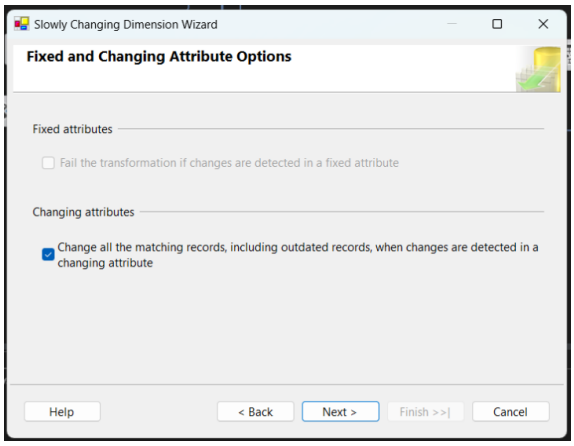
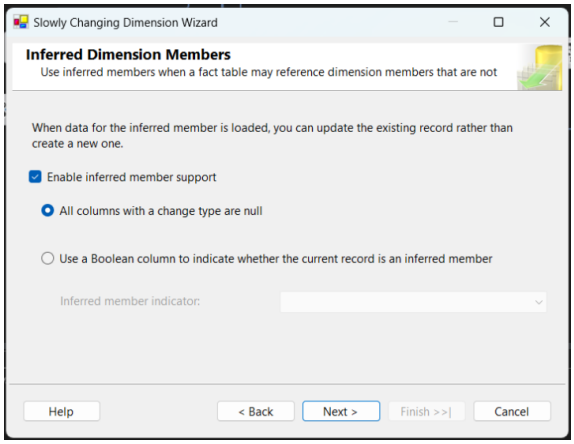
- cinema (Dimension)

Tên cột	Nguồn	Kiểu DL	Value mẫu
CinemaID_SK		int	1
Cinema Name	cinema Stage (Cinema Name)	nvarchar	Galaxy Cinema
Capacity	cinema Stage (Capacity)	int	1800
Standard_Seats_price	cinema Stage (Standard_Seats_price)	int	30000
VIP_Seats_price	cinema Stage (VIP_Seats_price)	int	50000





STT	Trường hợp	Mô tả	SSIS tool
1. Data Flow Task			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source
1.2		Tạo SCD	
			

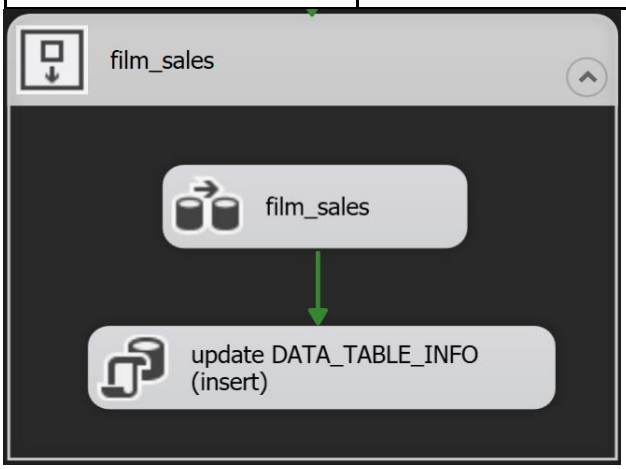
			
			
			
	Có giá trị mới	Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
	-	Truyền dữ liệu vào cinema DDS	OLEDB Destination
	Thay đổi giá trị cũ	Lookup các dòng dữ liệu cũ trong bảng cinema DDS	LookUp

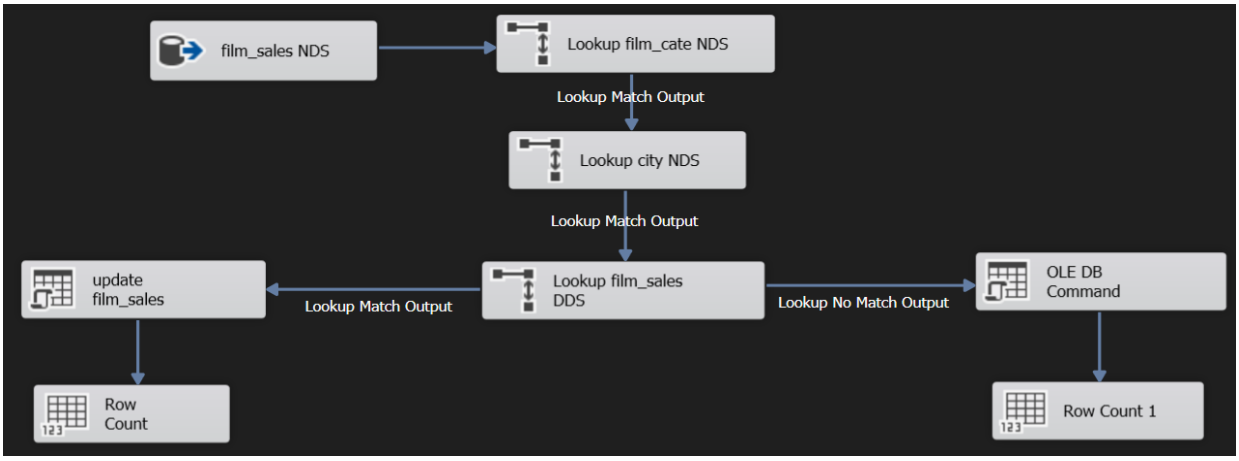
	match	Insert dòng dữ liệu vào bảng cinema_Audit	OLE DB Command
		Update dòng dữ liệu mới vào bảng cinema DDS	OLE DB Command
		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
2. Execute SQL Task			
		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng cinema DDS trong METADATA	Execute SQL Task

- film_sales (Fact)

Tên cột	Nguồn	Kiểu DL	Value mẫu
RevenueID_SK		int	1
FilmID_SK	film NDS (FilmID_SK)	int	1
FilmCategoryID_SK	film_cate NDS (FilmCategoryID_SK)	int	1
CinemaID_SK	cinema NDS (CinemaID_SK)	int	1
CityID_SK	city NDS (CityID_SK)	int	1
total_sales	film_sales Stage (total_sales)	int	4160000
tickets_sold	film_sales Stage (tickets_sold)	int	208
Standard_Seats_sold	film_sales Stage (Standard_Seats_sold)	int	517
VIP_Seats_sold	film_sales Stage (VIP_Seats_sold)	int	648
tickets_out	film_sales Stage	int	0

	(tickets_out)		
Standard_Seats_out	film_sales Stage (Standard_Seats_out)	int	0
VIP_Seats_out	film_sales Stage (VIP_Seats_out)	int	0
ticket_use	film_sales Stage (ticket_use)	int	1161
occur_perc	film_sales Stage (occur_perc)	float	64.5
DateKey	DATE_MASTER NDS (DateKey)	nvarchar	20190105





STT	Trường hợp	Mô tả	SSIS tool

1. Data Flow Task			
1.1	-	Lấy dữ liệu từ film_sales NDS	OLEDB Source
		Lookup giá trị FilmCategoryID_SK trong bảng film_cate NDS	Lookup
	match	Lookup giá trị CityID_SK trong bảng city NDS	Lookup
	match	Lookup giá trị RevenueID_SK trong bảng film_sales NDS	Lookup
	match	Update dòng dữ liệu mới vào bảng film_sales DDS	OLE DB Command
		Đếm số dòng vừa Update và lưu vào Count_Row	Row Count
	no match	Insert dòng dữ liệu mới vào bảng film_sales DDS	OLE DB Command
		Đếm số dòng vừa insert và lưu vào Count_Row	Row Count
2. Execute SQL Task			
		Kiểm tra Row Count > 0 thì lưu việc cập nhật bảng cinema DDS trong METADATA	Execute SQL Task

4. Danh sách các table thuộc về các database

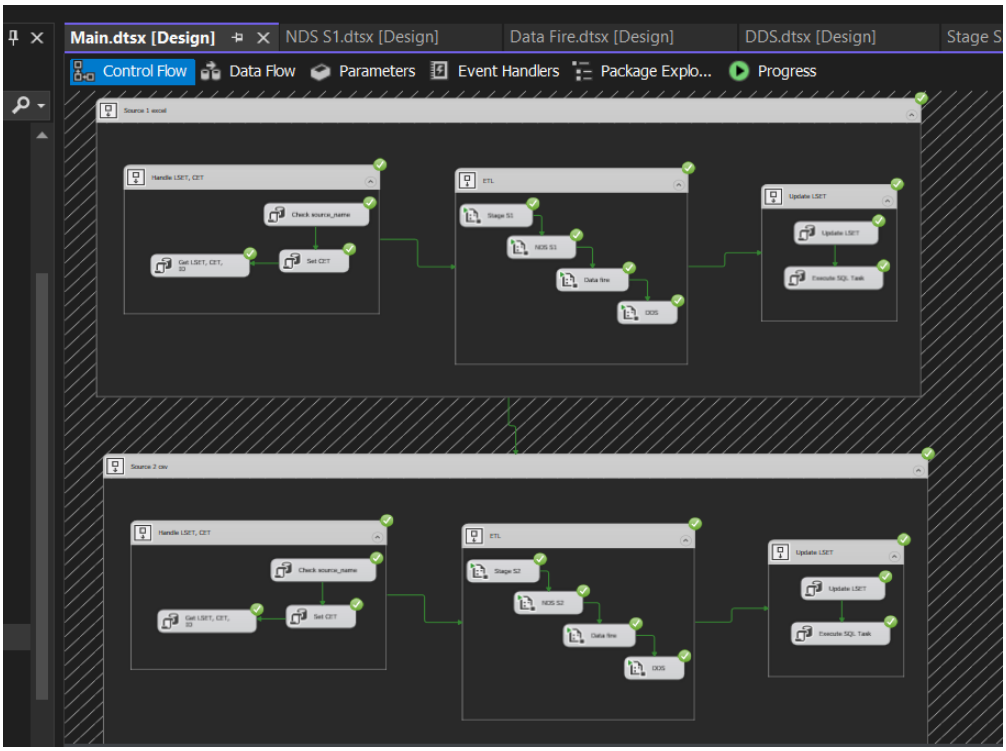
Database	**Table**	**Ý nghĩa**
BI07_METADATA	[dbo].[DATA_FLOW]	Lưu trữ thông tin ETL của các source
	[dbo].[DATA_STORE_INFO]	Lưu trữ thông tin các database trong hệ thống

	[dbo].[DATA_TABLE_INFO]	Lưu trữ thông tin các table trong từng database
BI07_STAGE	[dbo].[city]	Lưu trữ thông tin Stage của bảng city
	[dbo].[cinema]	Lưu trữ thông tin Stage của bảng cinema
	[dbo].[film]	Lưu trữ thông tin Stage của bảng film
	[dbo].[film_cate]	Lưu trữ thông tin Stage của bảng film_cate
	[dbo].[film_sales_S1]	Lưu trữ thông tin Stage của bảng film_sales_S1
	[dbo].[film_sales_S2]	Lưu trữ thông tin Stage của bảng film_sales_S2
BI07_NDS	[dbo].[DATE_MASTER]	Lưu thông tin ngày, tháng, năm
	[dbo].[city]	Lưu trữ dữ liệu sau khi format lại từ city stage
	[dbo].[cinema]	Lưu trữ dữ liệu sau khi format lại từ cinema stage
	[dbo].[film]	Lưu trữ dữ liệu sau khi format lại từ film stage
	[dbo].[film_cate]	Lưu trữ dữ liệu sau khi format lại từ film_cate stage
	[dbo].[film_sales]	Lưu trữ dữ liệu sau khi format lại từ film_sales_S1 stage & film_sales_S2 stage
BI07_DDS	[dbo].[DATE_MASTER]	Lưu thông tin ngày, tháng, năm
	[dbo].[city]	Lưu trữ dữ liệu city trong DWH

	[dbo].[city_Audit]	Lưu trữ các dữ liệu cũ của city trong DWH
	[dbo].[cinema]	Lưu trữ dữ liệu cinema trong DWH
	[dbo].[cinema_Adit]	Lưu trữ các dữ liệu cũ của cinema trong DWH
	[dbo].[film]	Lưu trữ dữ liệu film trong DWH
	[dbo].[film_Adit]	Lưu trữ các dữ liệu cũ của film trong DWH
	[dbo].[film_cate]	Lưu trữ dữ liệu film_cate trong DWH
	[dbo].[film_cate_Adit]	Lưu trữ các dữ liệu cũ của film_cate trong DWH
	[dbo].[film_sales]	Lưu trữ dữ liệu film_sales trong DWH

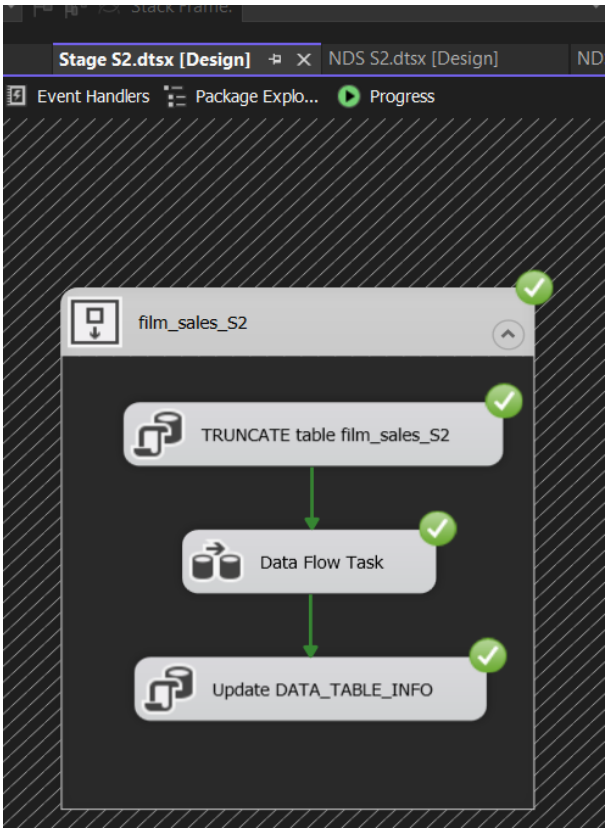
IV. Hình các package

- Main

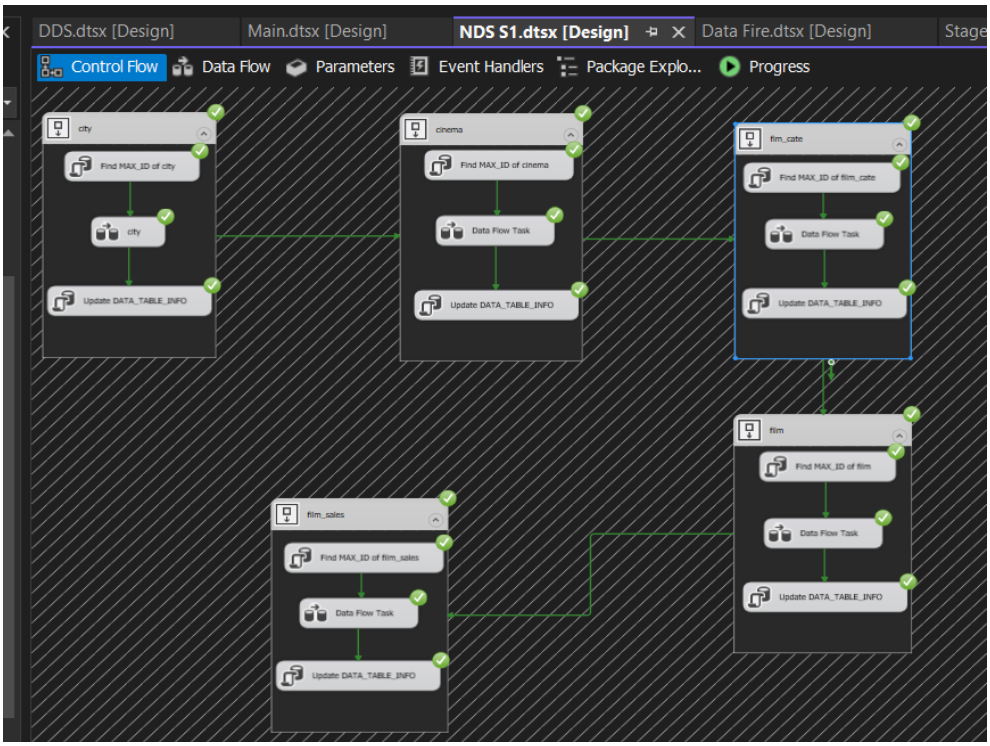


- Stage 1

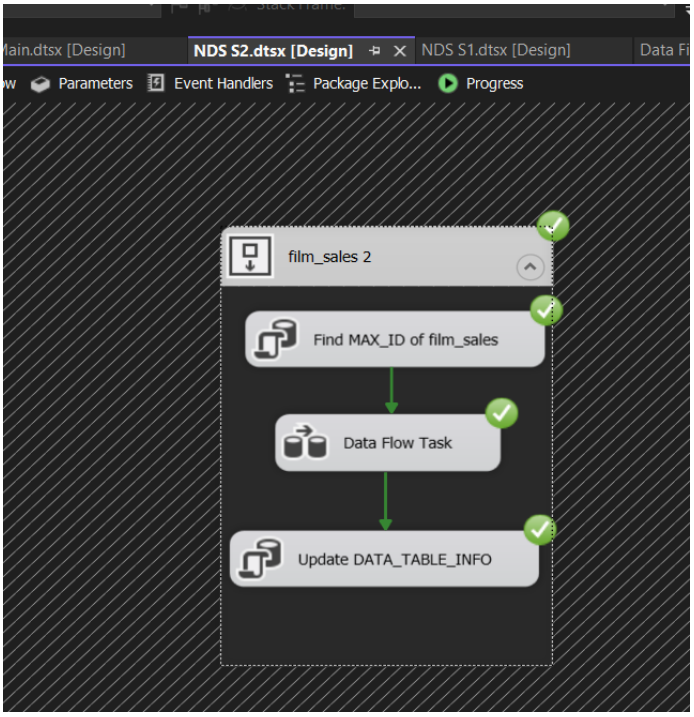
- Stage 2


Stage S2.dtsx [Design]
NDS S2.dtsx [Design]
Event Handlers
Package Explo...
Progress
film_sales_S2
TRUNCATE table film_sales_S2
Data Flow Task
Update DATA_TABLE_INFO

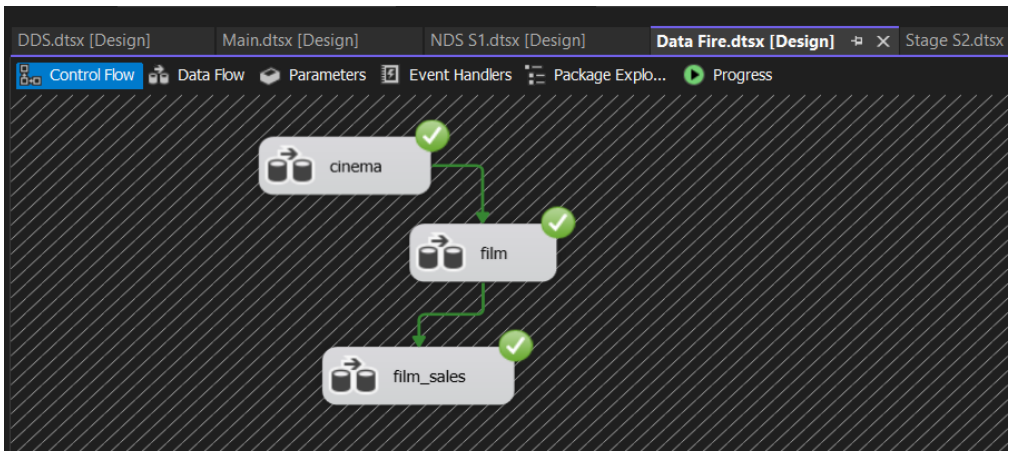
- NDS 1


DDS.dtsx [Design]
Main.dtsx [Design]
NDS S1.dtsx [Design]
Data Fire.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress

- NDS 2

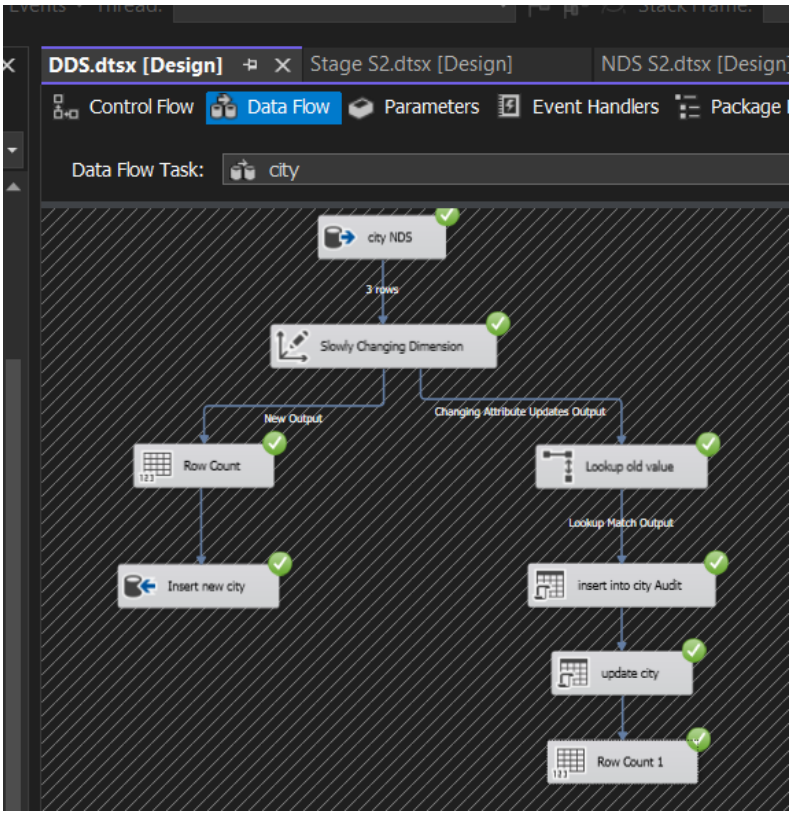


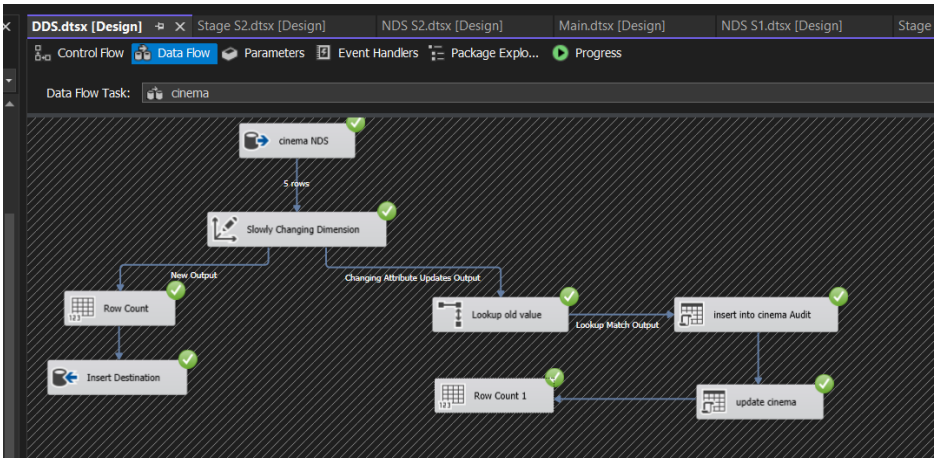
- Data Fire

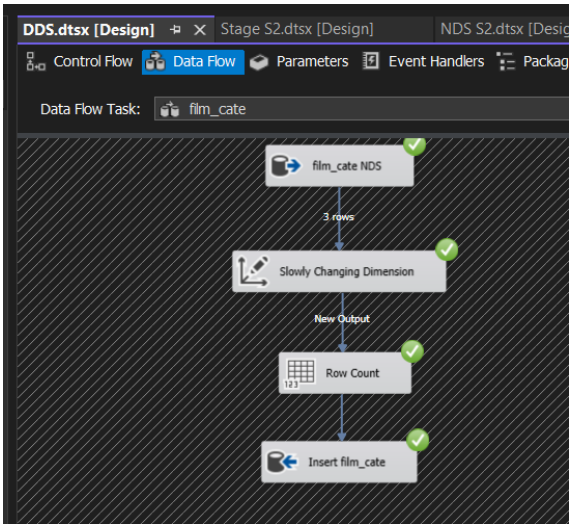


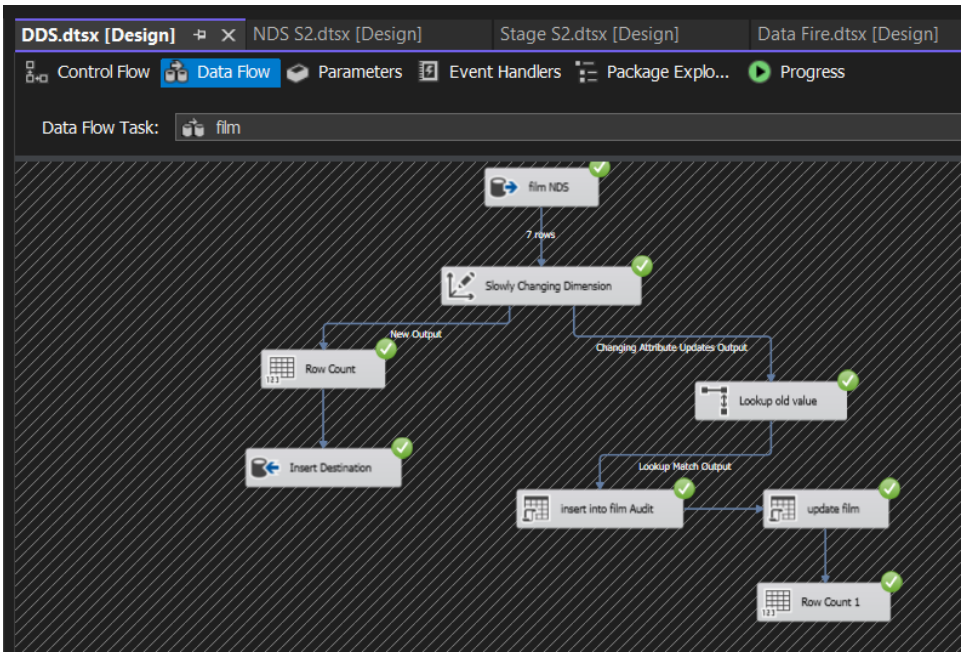
- DDS

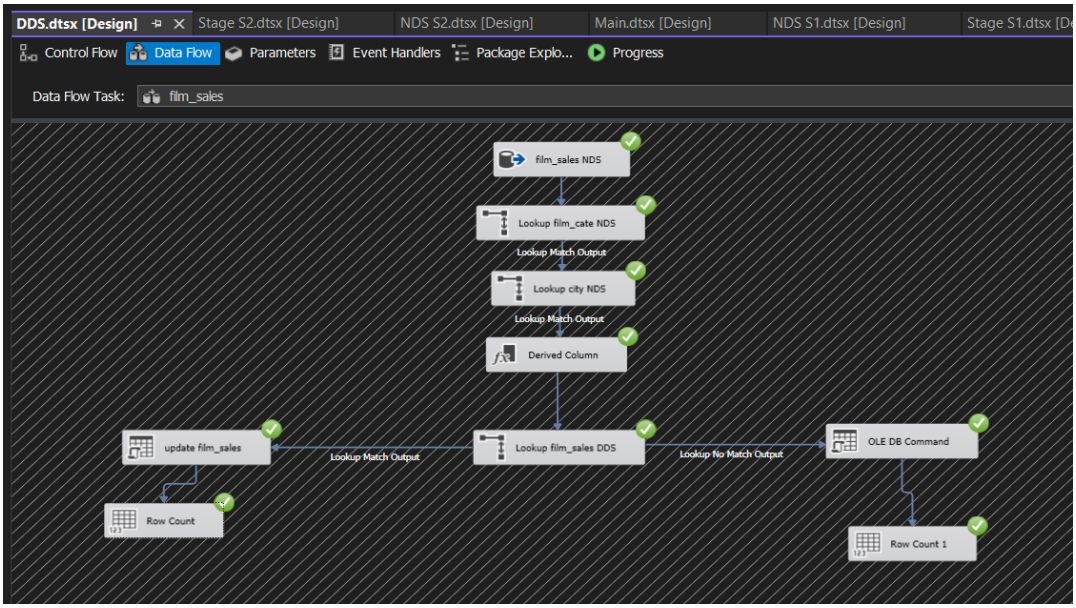
- Chụp chi tiết 1 flow nạp bảng chiều & fact




DDS.dtsx [Design]
Stage S2.dtsx [Design]
NDS S2.dtsx [Design]
Main.dtsx [Design]
NDS S1.dtsx [Design]
Stage
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
cinema
cinema NDS
5 rows
Slowly Changing Dimension
New Output
Changing Attribute Updates Output
Row Count
Lookup old value
Lookup Match Output
insert into cinema Audit
Insert Destination
Row Count 1
update cinema


DDS.dtsx [Design]
Stage S2.dtsx [Design]
NDS S2.dtsx [Desig
Control Flow
Data Flow
Parameters
Event Handlers
Packag
Data Flow Task:
film_cate
film_cate NDS
3 rows
Slowly Changing Dimension
New Output
Row Count
Insert film_cate


DDS.dtsx [Design]
NDS S2.dtsx [Design]
Stage S2.dtsx [Design]
Data Fire.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
film
film NDS
7 rows
Slowly Changing Dimension
New Output
Changing Attribute Updates Output
Row Count
Lookup old value
Insert Destination
Lookup Match Output
insert into film Audit
update film
Row Count 1

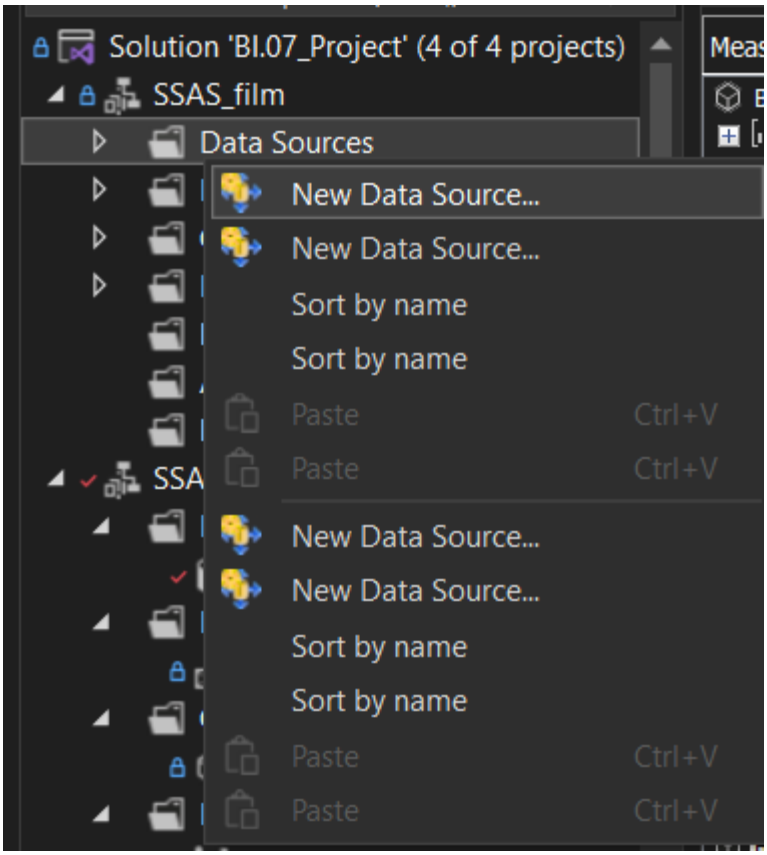


V. Analysis Service

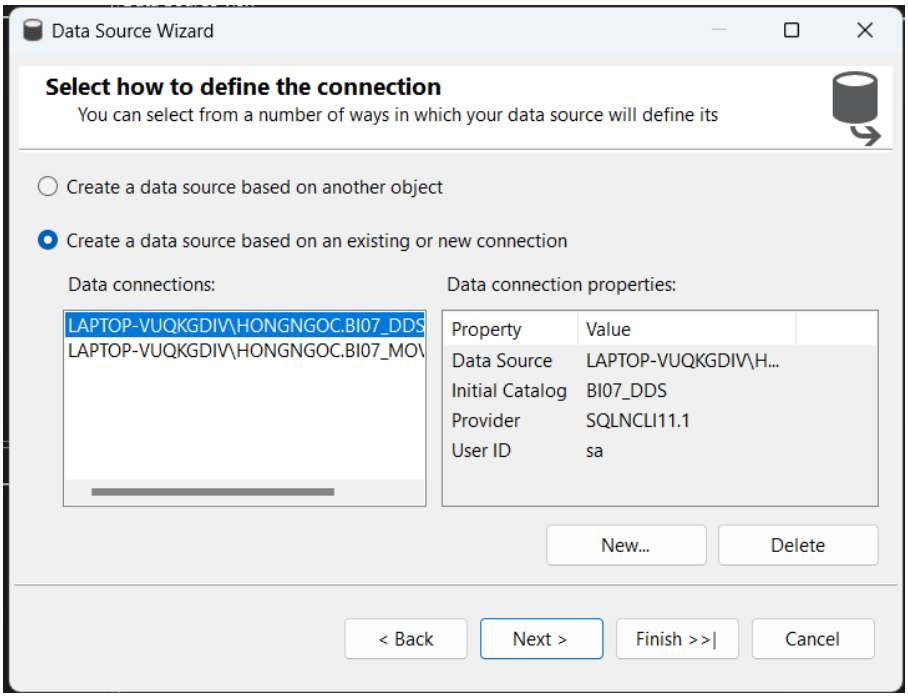
1. Cài đặt server:

2. Tạo mới Data Source

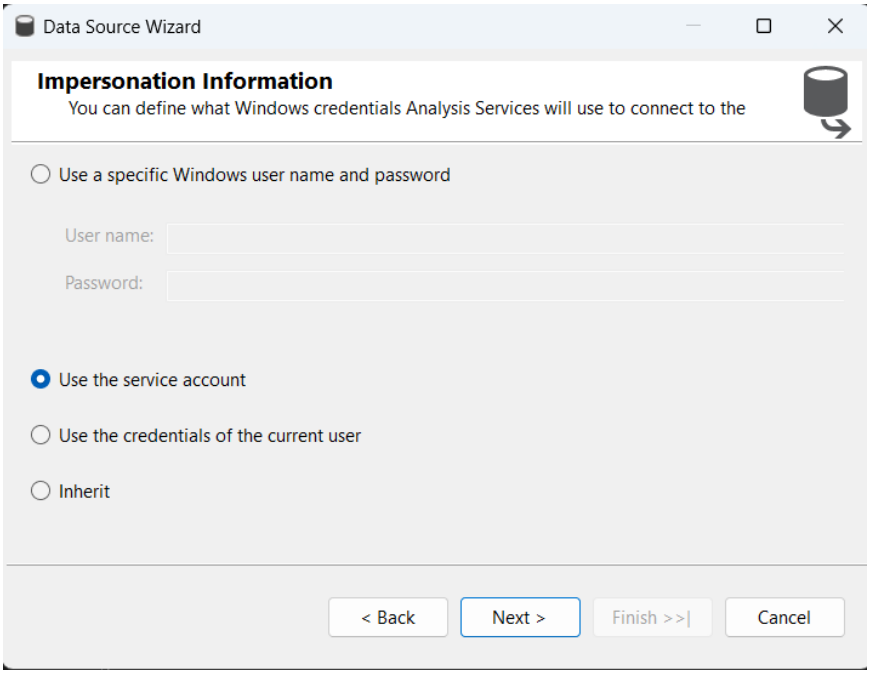
- Click phải và chọn “New Data Source”



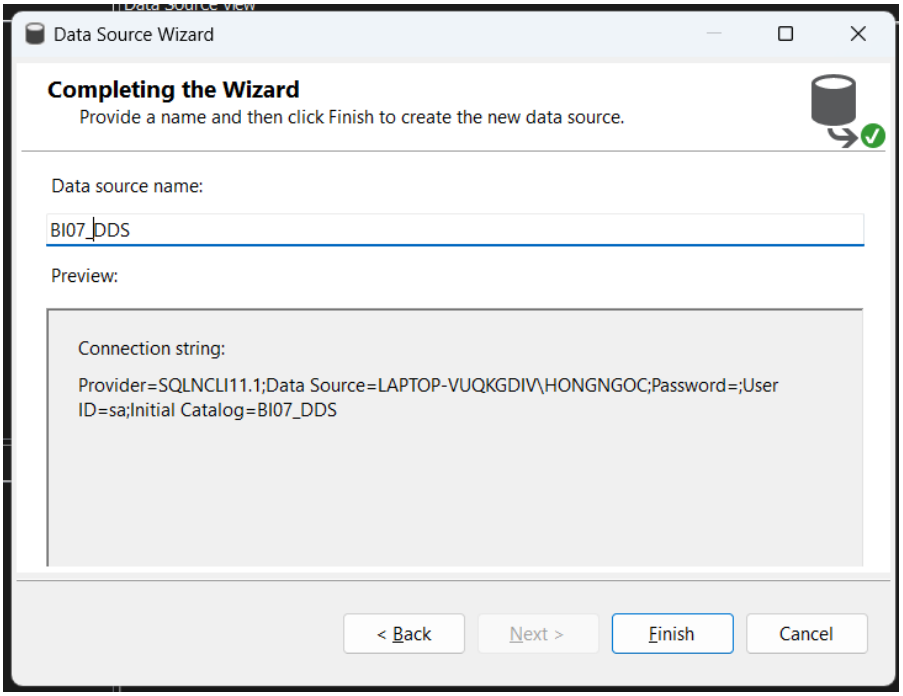
- Sau đó chọn tới chỗ connect lưu dữ liệu trong SSMS



- Tại đây nếu ta không muốn nhập username và password của máy thì có thể chọn mục thứ 2 thì sẽ sử dụng service account



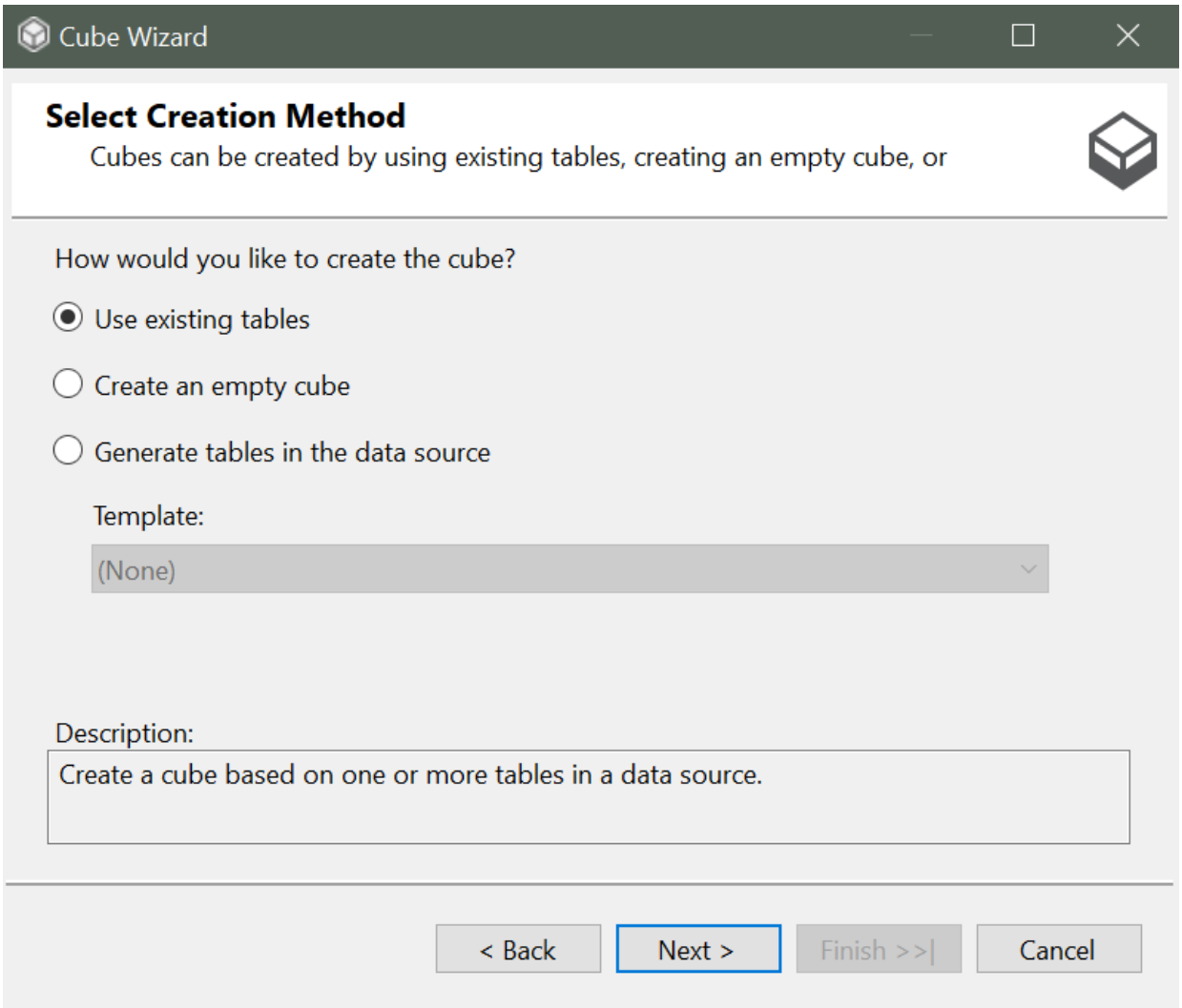
- Và đặt tên cho data source



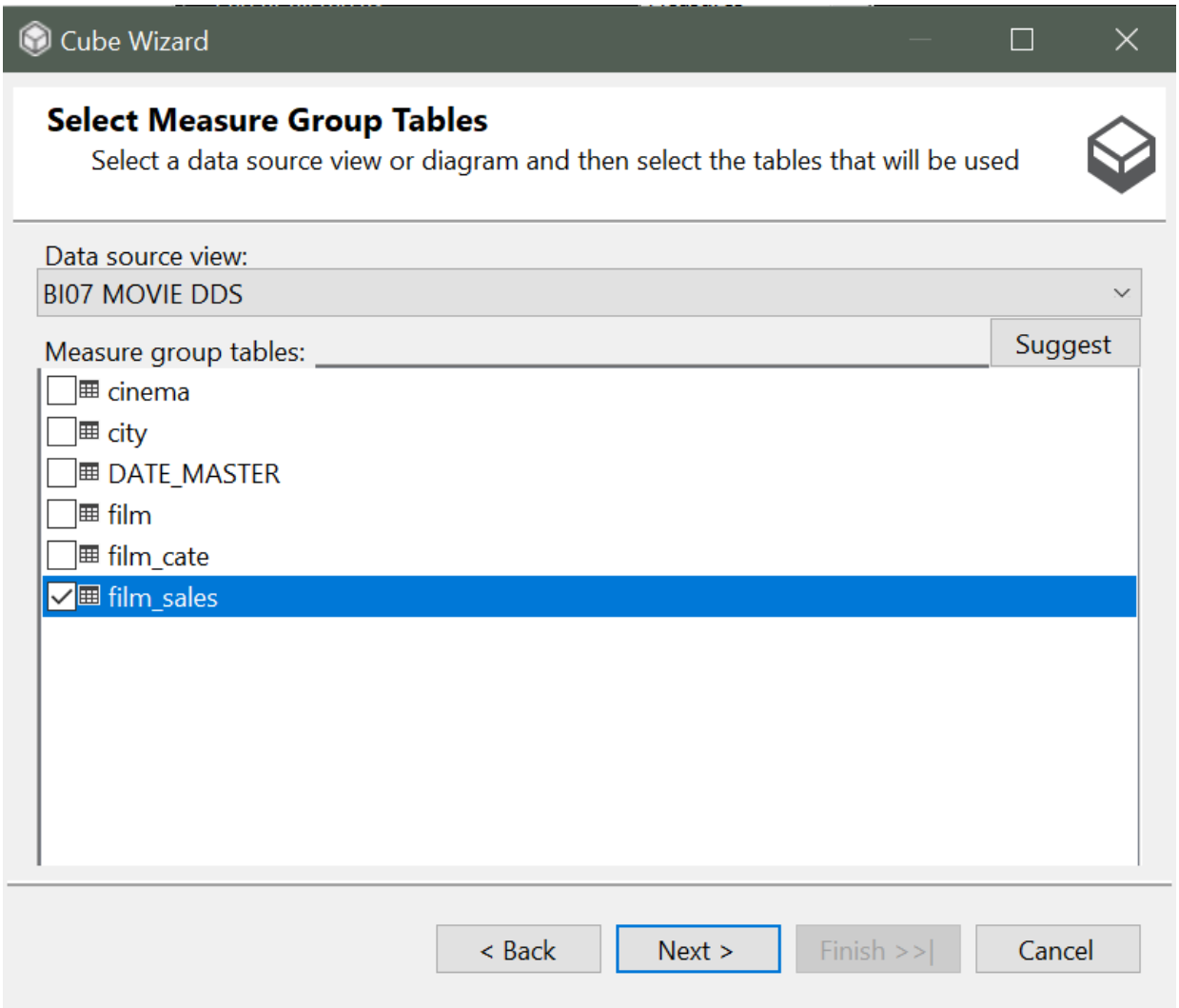
3. Tạo mới Data Source Views

4. Tạo Cube

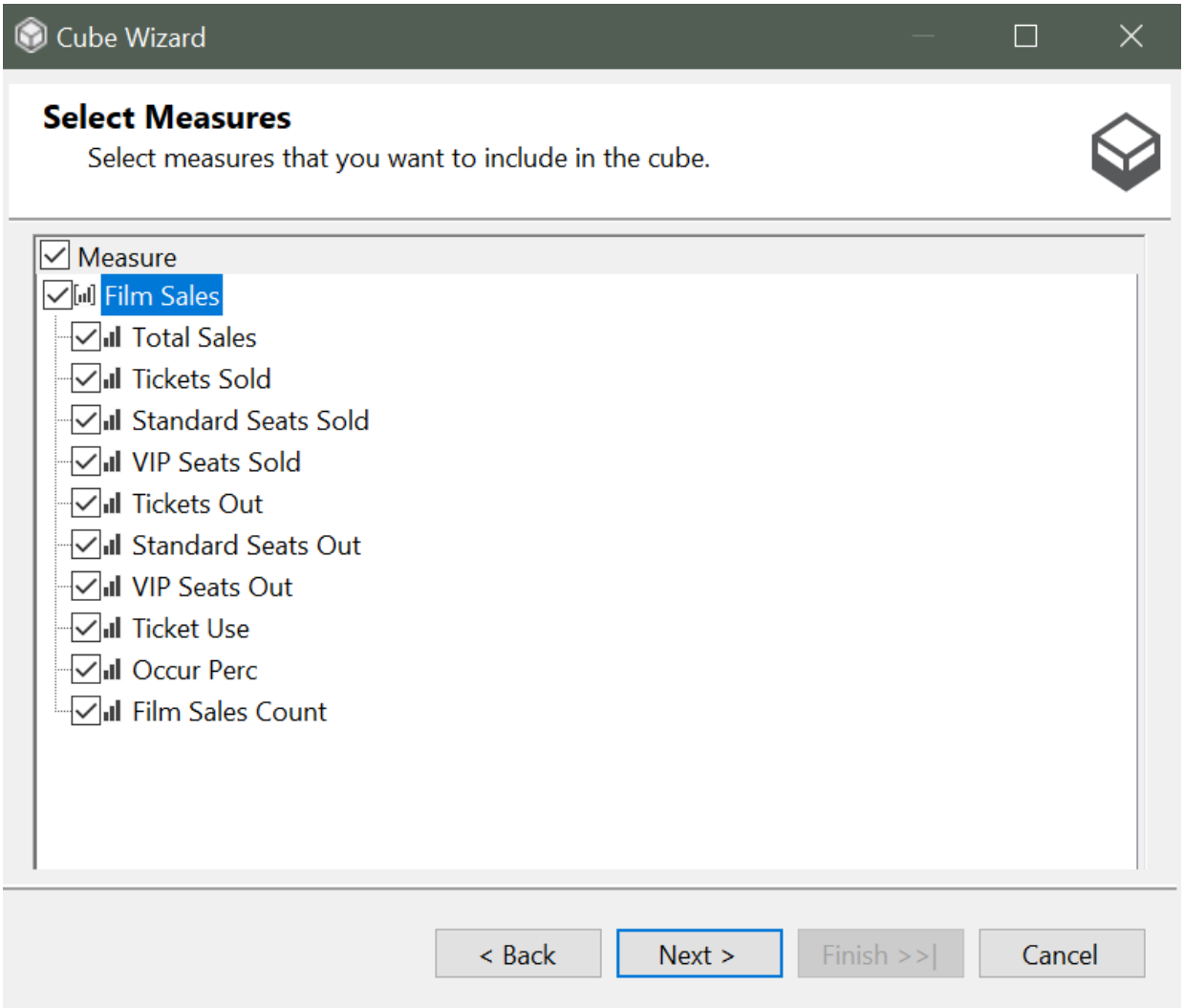
- Creation Method: Use existing tables (vì đã có tạo bảng trong DB BI07_MOVIE_DDS)



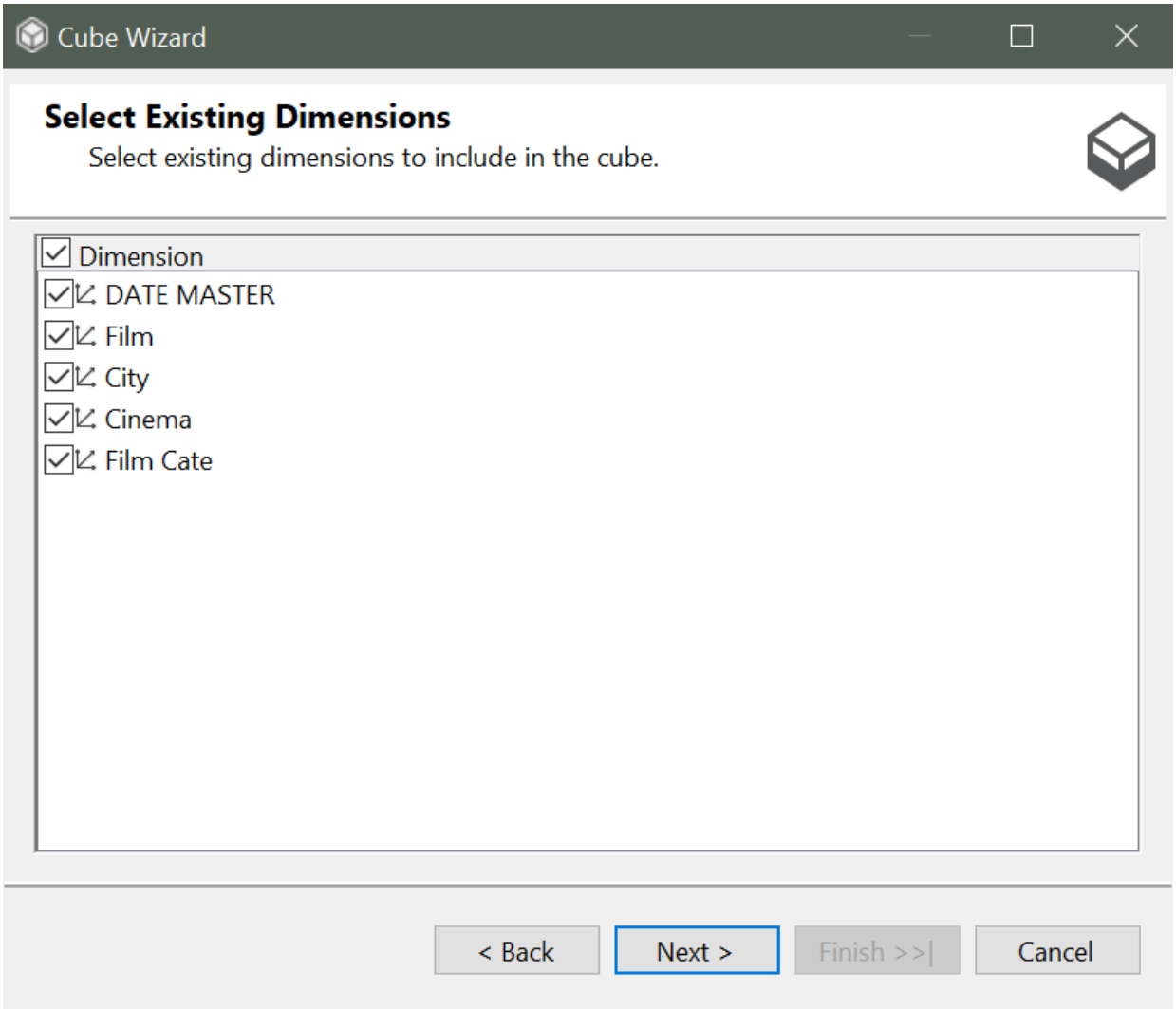
- Measure Group Tables: chọn film_sale (bảng Fact)



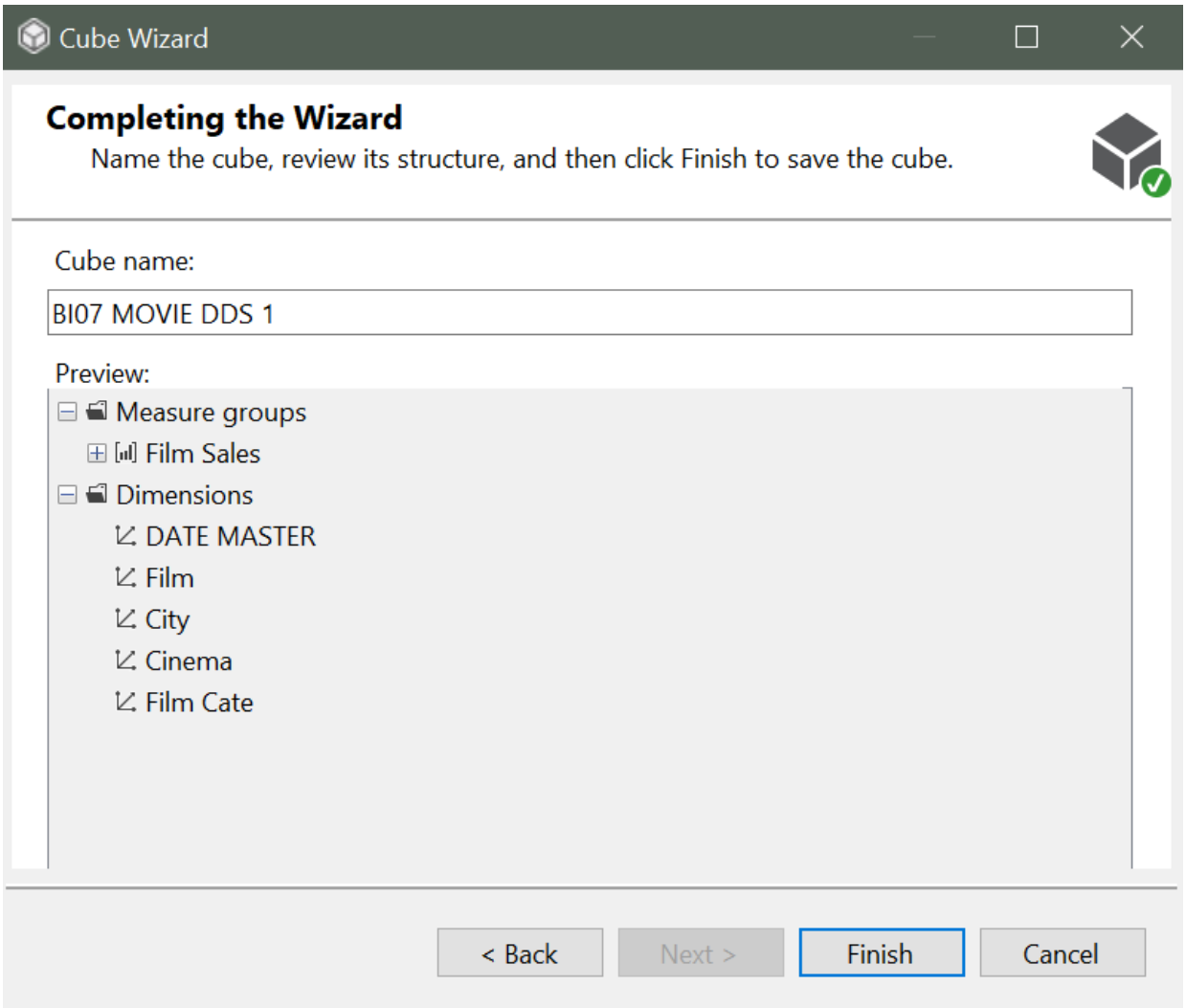
- Measures: Chọn tất cả Measure cần để tính toán và thống kê



- Select Existing Dimension: Chọn các chiều đã có sẵn trong View

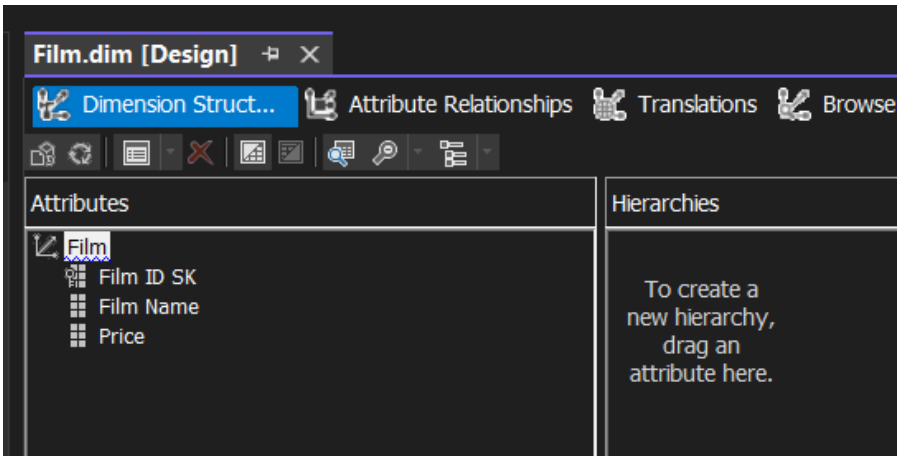


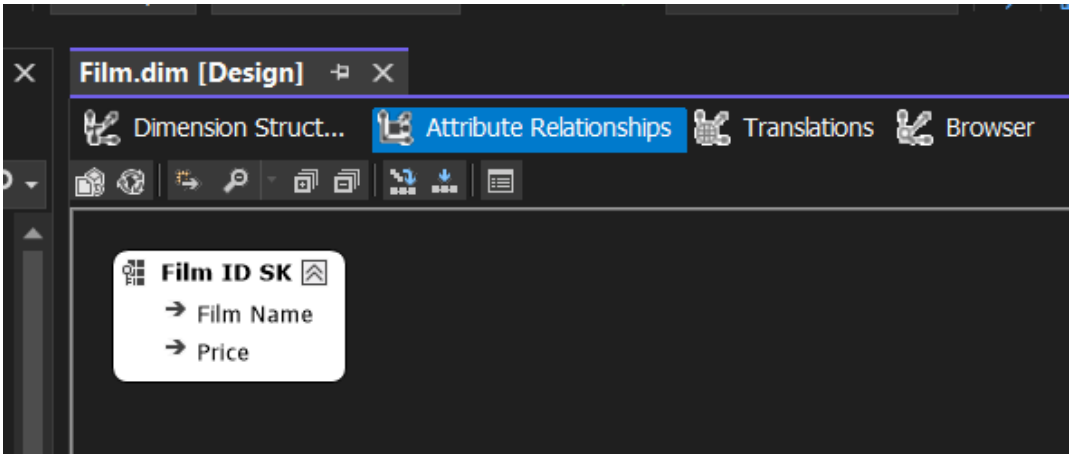
- Complete: Kiểm tra lại các Measures, Dimension và đặt tên Cube => Finish để tạo Cube



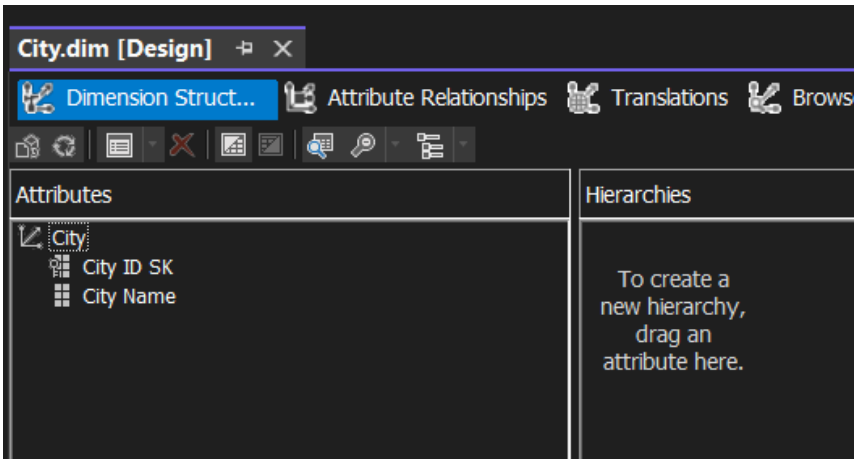
5. Cấp chiều Cube

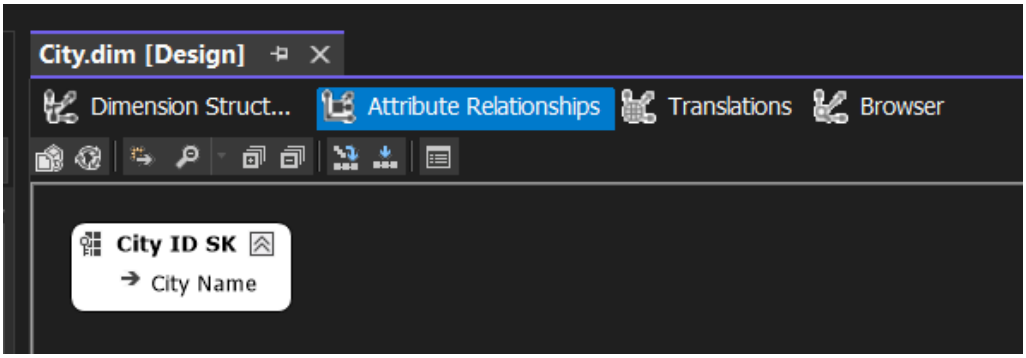
- Film



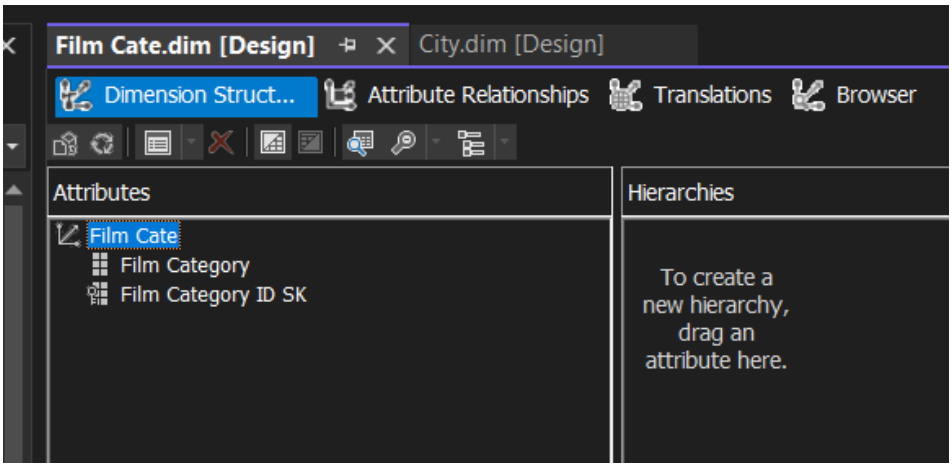


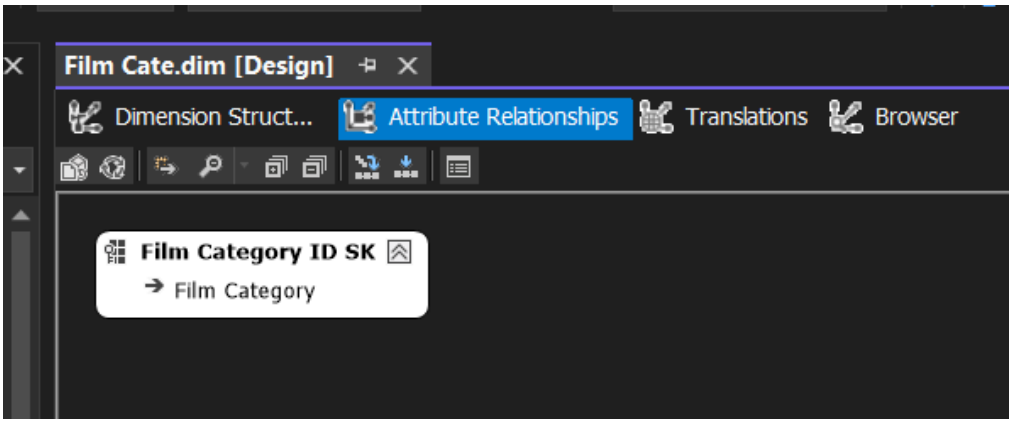
- City



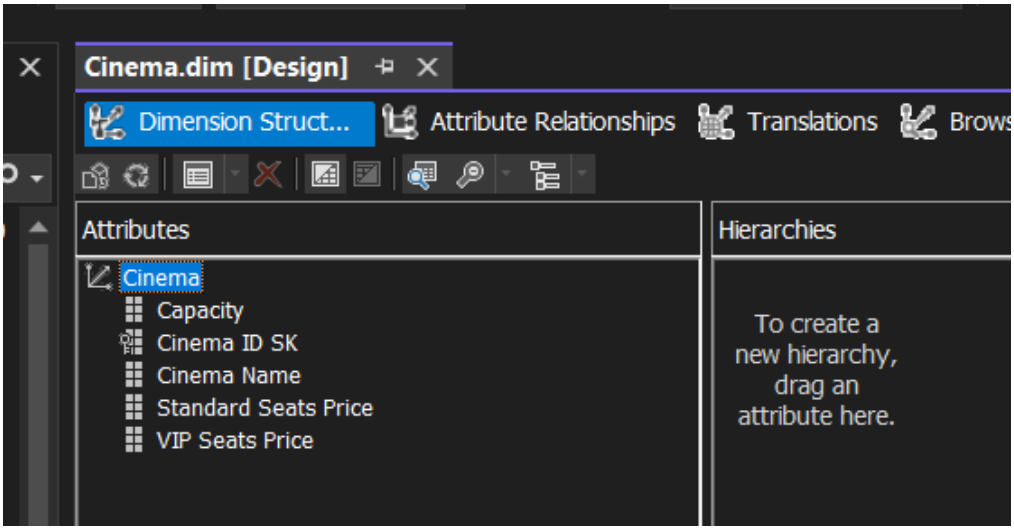


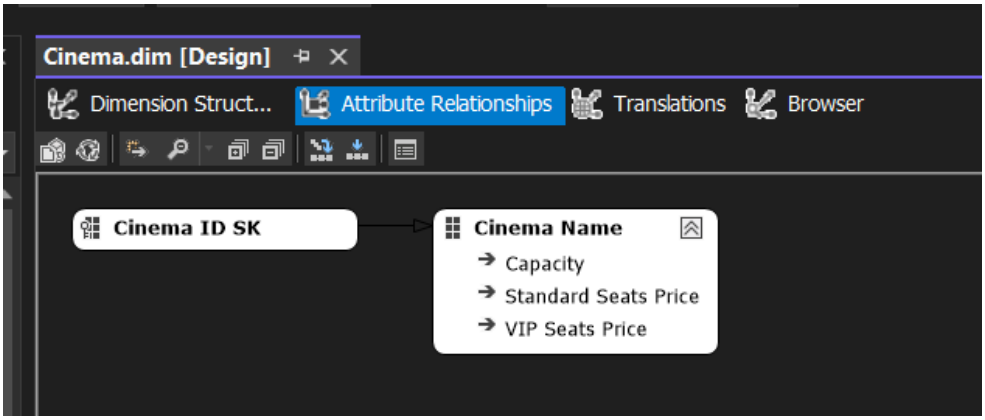
- Film Cate


Film Cate.dim [Design]
City.dim [Design]
Dimension Struct...
Attribute Relationships
Translations
Browser
Attributes
Hierarchies
Film Cate
Film Category
Film Category ID SK
To create a new hierarchy, drag an attribute here.

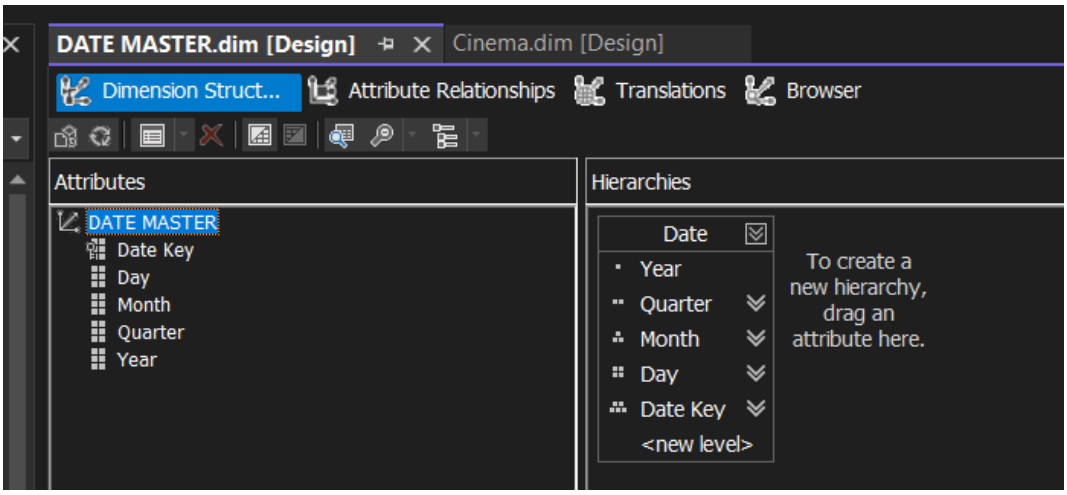

Film Cate.dim [Design]
Dimension Struct...
Attribute Relationships
Translations
Browser
Film Category ID SK
Film Category

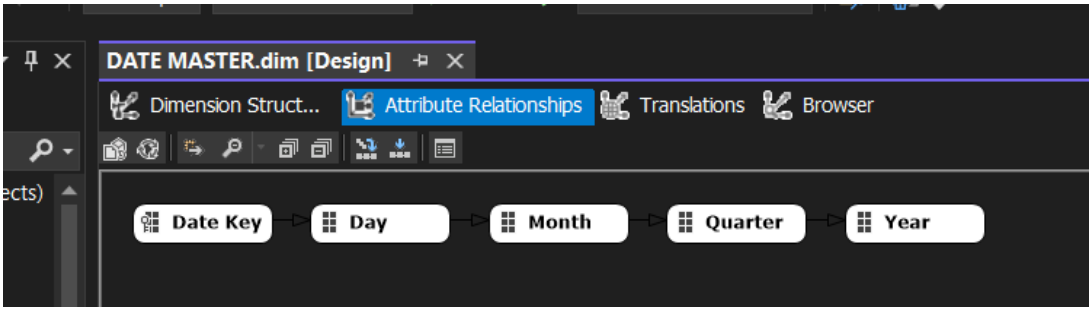
- Cinema


Cinema.dim [Design]
Dimension Struct...
Attribute Relationships
Translations
Attributes
Hierarchies
Cinema
Capacity
Cinema ID SK
Cinema Name
Standard Seats Price
VIP Seats Price
To create a new hierarchy, drag an attribute here.

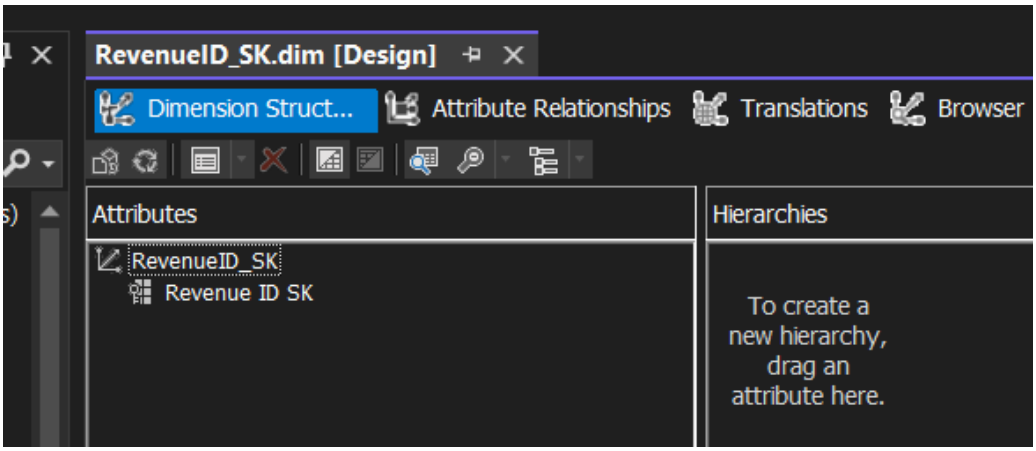


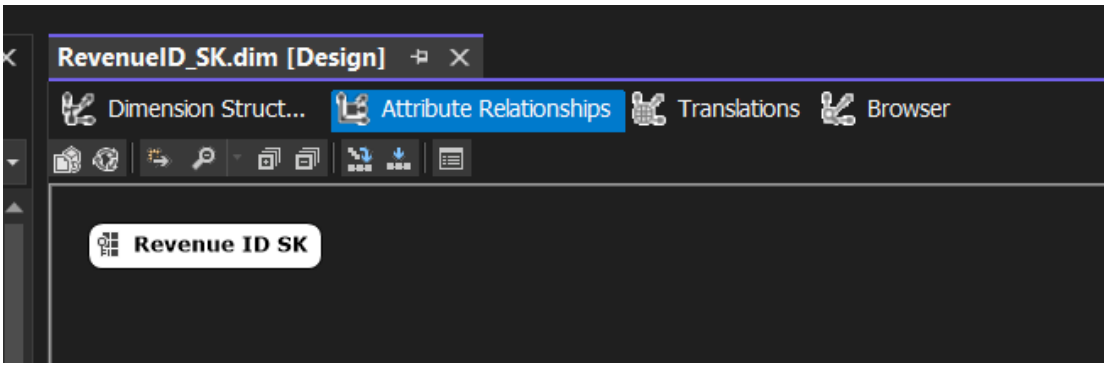
- Date Master





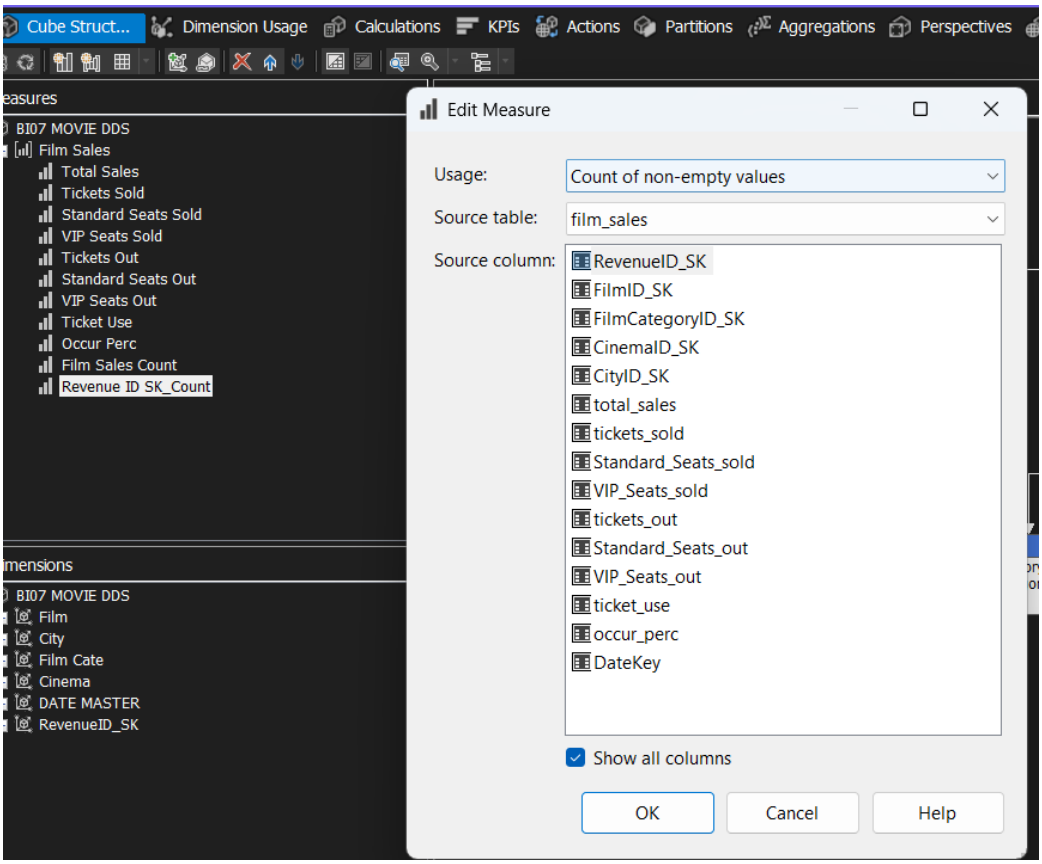
- RevenueID_SK (lấy ReveueID_SK từ bảng film_sales ra làm dim)



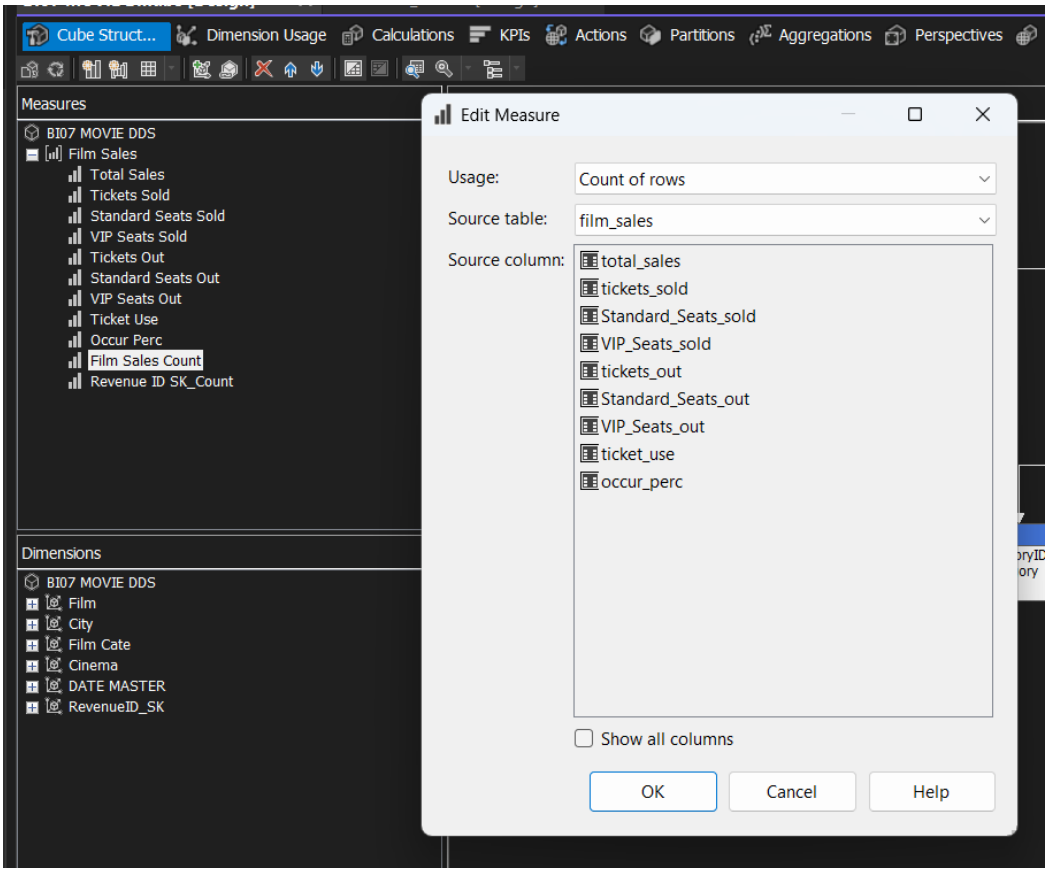


6. Tạo measure

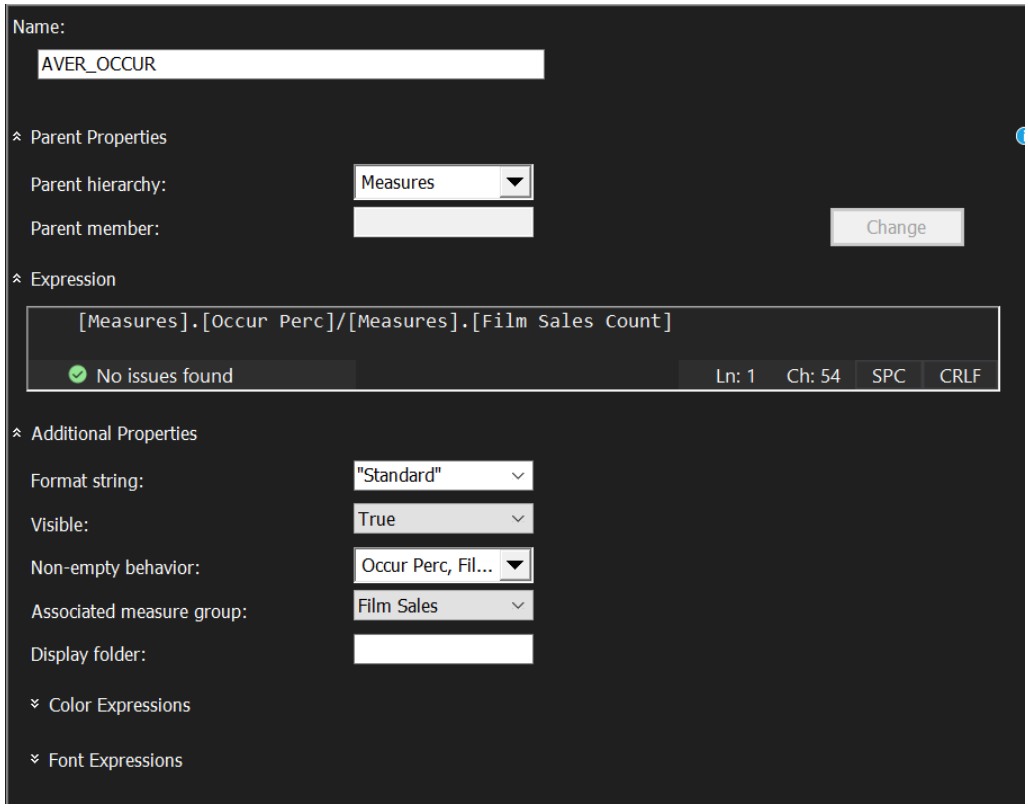
- RevenueID SK_Count



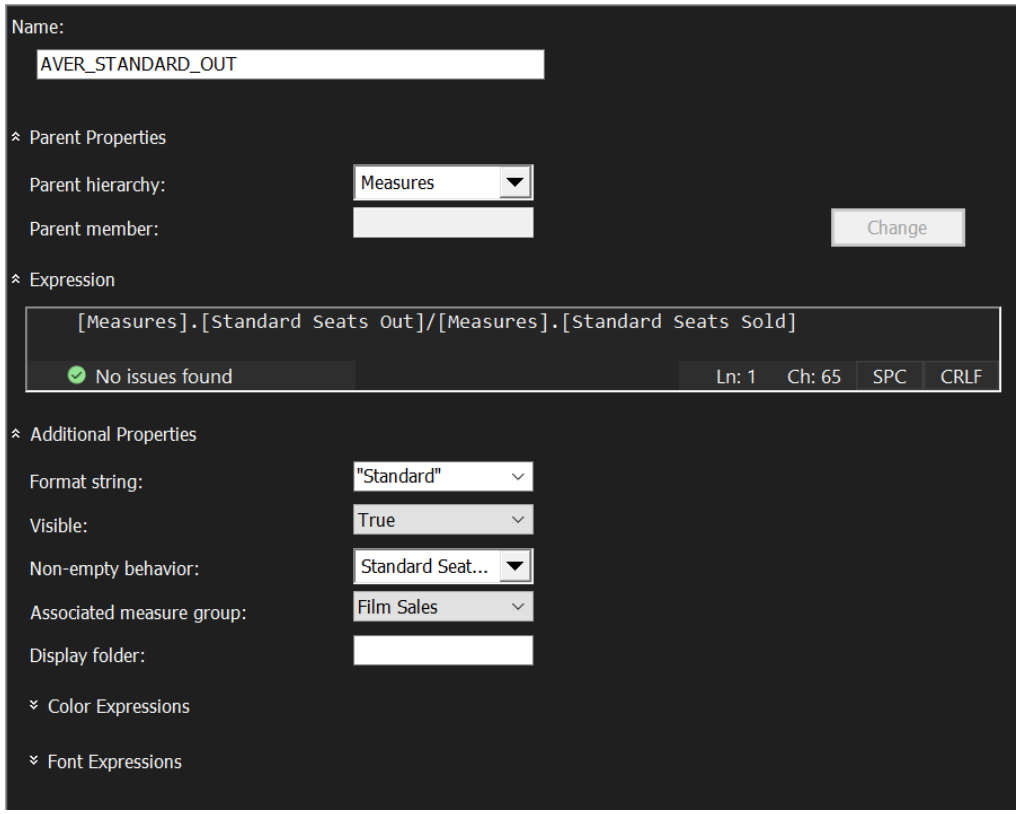
- Film Sales Count


Cube Struct...
Dimension Usage
Calculations
KPIs
Actions
Partitions
Aggregations
Perspectives
Measures
BI07 MOVIE DDS
Film Sales
Total Sales
Tickets Sold
Standard Seats Sold
VIP Seats Sold
Tickets Out
Standard Seats Out
VIP Seats Out
Ticket Use
Occur Perc
Film Sales Count
Revenue ID SK_Count
Dimensions
BI07 MOVIE DDS
Film
City
Film Cate
Cinema
DATE MASTER
RevenueID_SK
Edit Measure
Usage:
Count of rows
Source table:
film_sales
Source column:
total_sales
tickets_sold
Standard_Seats_sold
VIP_Seats_sold
tickets_out
Standard_Seats_out
VIP_Seats_out
ticket_use
occur_perc
Show all columns
OK
Cancel
Help

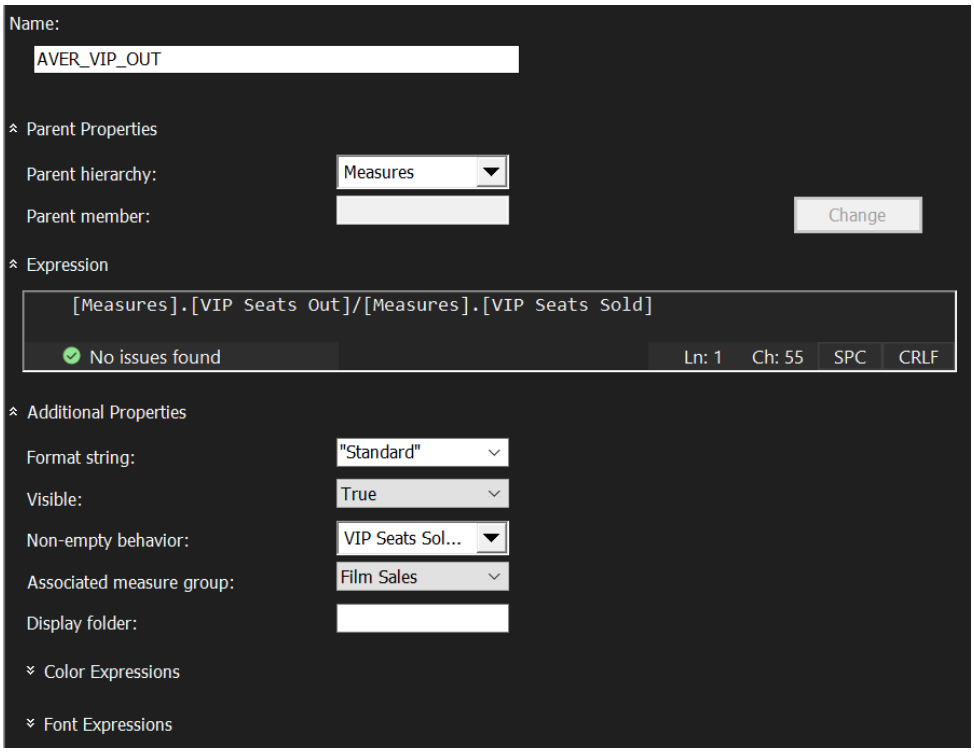
- AVER_OCCUR



- AVER_STANDARD_OUT

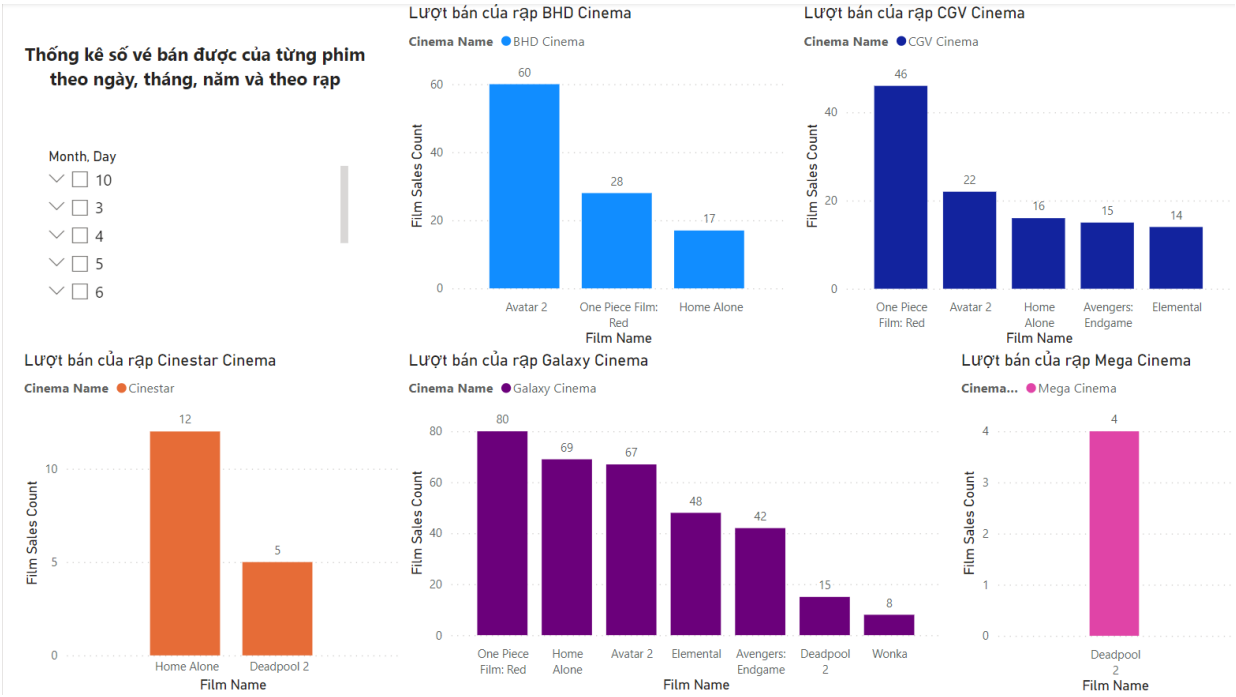


- AVER_VIP_OUT



VI. Report

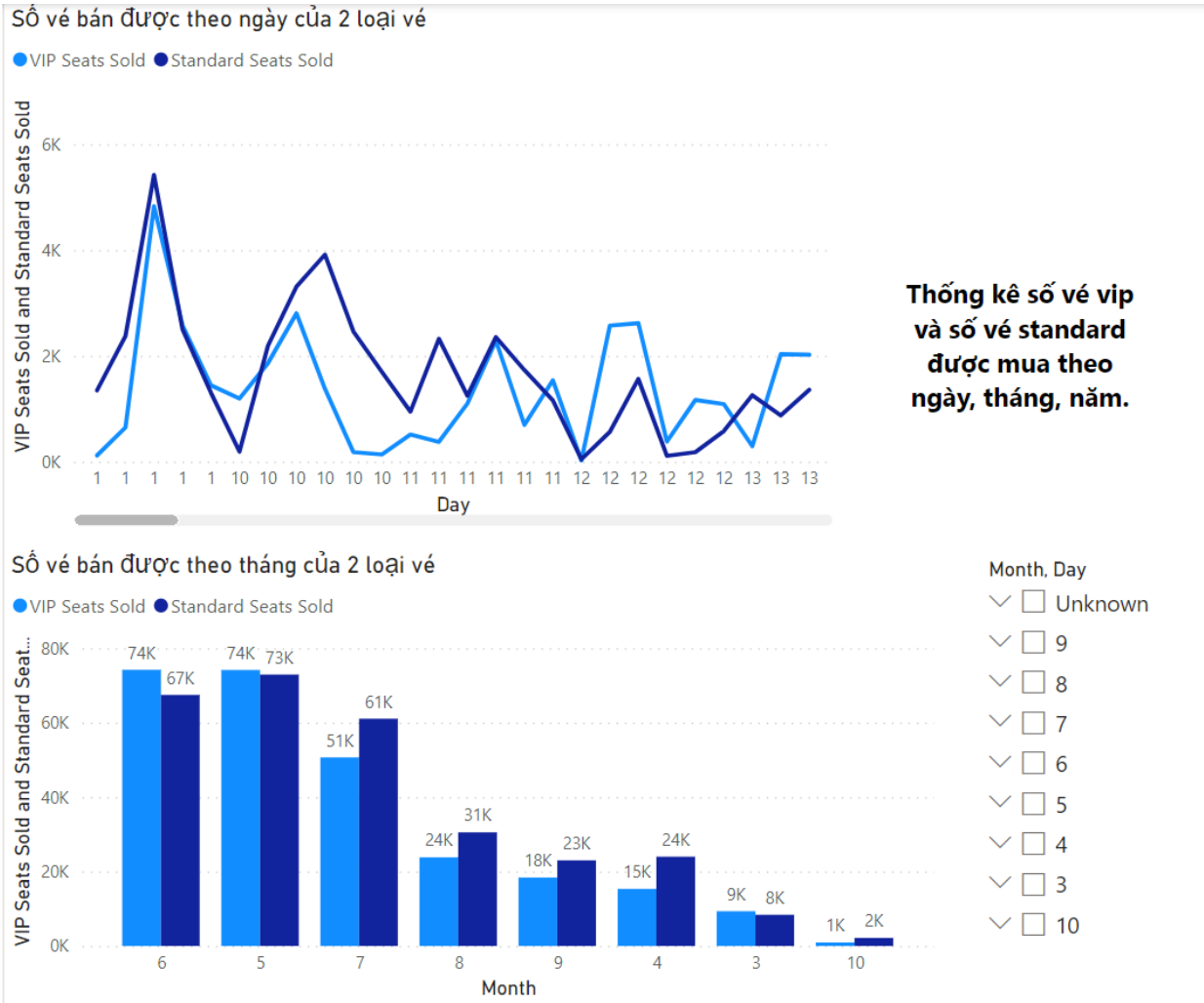
1. Thống kê số vé bán được của từng phim theo ngày, tháng, năm và theo rạp.



Loại biểu đồ: Clustered Column Chart

- Trục X: Tên phim (Film Name - film dimension)
- Trục Y: Số lượt bán (Film Sales Count - film_sales measure)
- Legend: Tên rạp phim (Cinema Name - Cinema dimension)
- Thực hiện filter các table theo từng Legend
- Slicer field: Month (DATE MASTER dimension), Day (DATE MASTER dimension)

2. Thống kê số vé vip và số vé standard được mua hàng theo ngày, tháng, năm.



Bảng Số vé bán được theo ngày của 2 loại vé

Loại biểu đồ: Line Chart

- Trục X: Day (Day - DATE MASTER dimension)
- Trục Y: Số lượt bán của 2 loại vé (VIP Seats Sold - film_sales measure, Standard Seats Sold - film_sales measure)

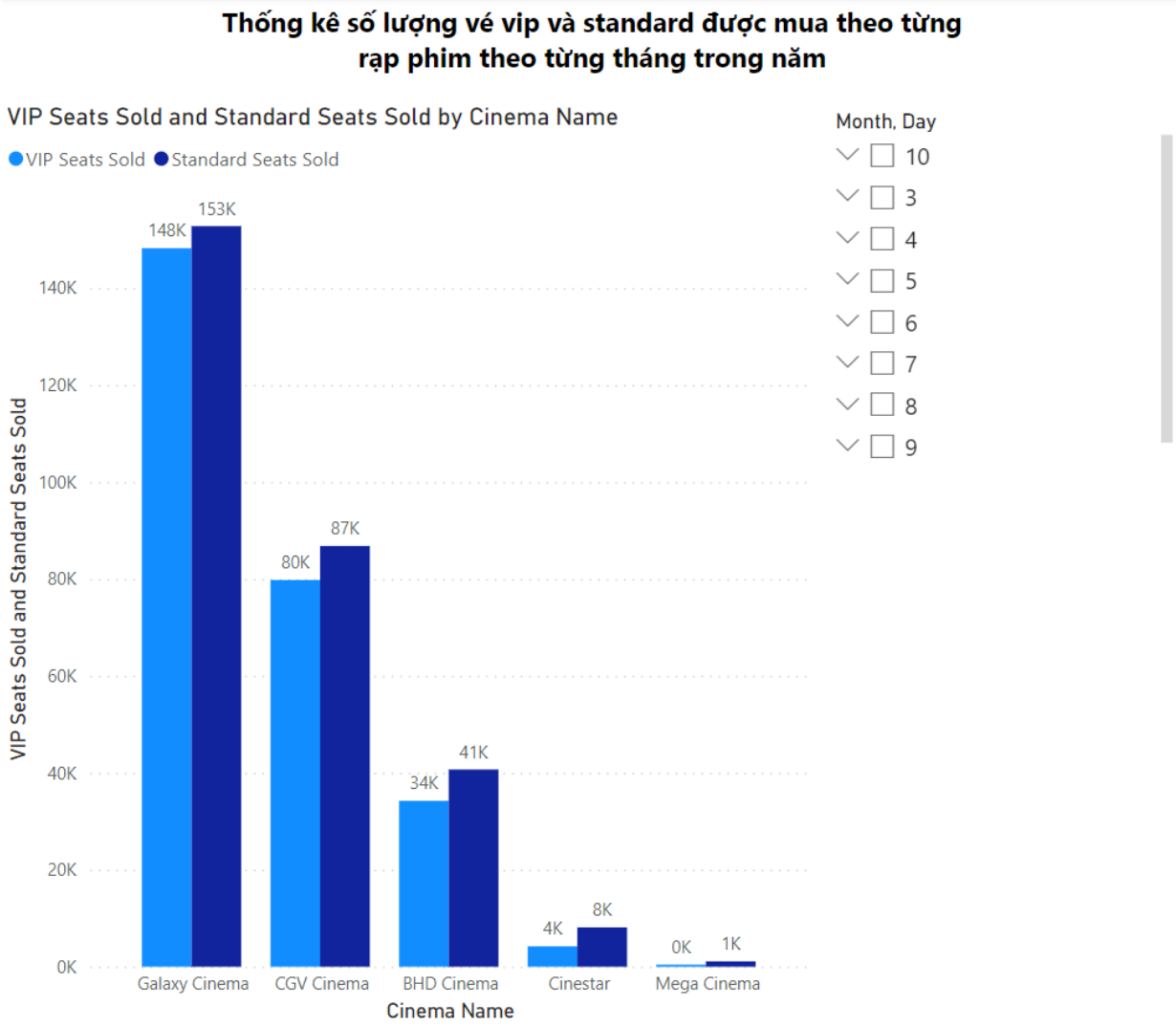
Bảng Số vé bán được theo tháng của 2 loại vé:

Loại biểu đồ: Clustered Column Chart

- Trục X: Month (Month - DATE MASTER dimension)
- Trục Y: Số lượt bán của 2 loại vé (VIP Seats Sold - film_sales measure, Standard Seats Sold - film_sales measure)

Slicer field: Month (DATE MASTER dimension), Day (DATE MASTER dimension)

3. Thống kê số lượng vé vip và standard được mua theo từng rạp phim theo từng tháng trong năm.

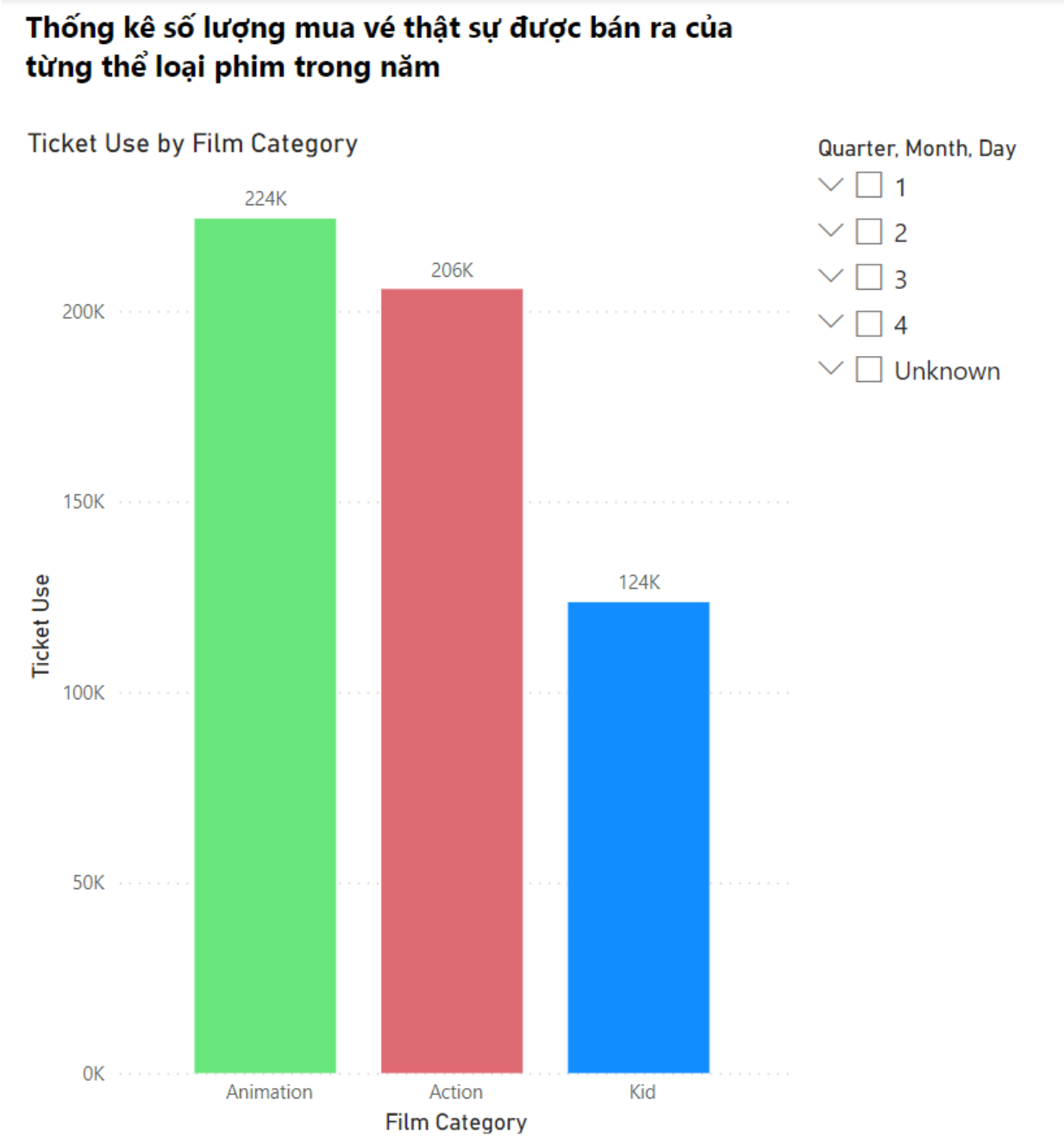


Loại biểu đồ: Clustered Column Chart

- Trục X: Tên rạp phim (Cinema Name - Cinema dimension)

- Trục Y: Số lượt bán của 2 loại vé (VIP Seats Sold - film_sales measure, Standard Seats Sold - film_sales measure)
- Slicer field: Month (DATE MASTER dimension), Day (DATE MASTER dimension)

4. Thống kê số lượng mua vé thật sự được bán ra của từng thể loại phim trong năm.

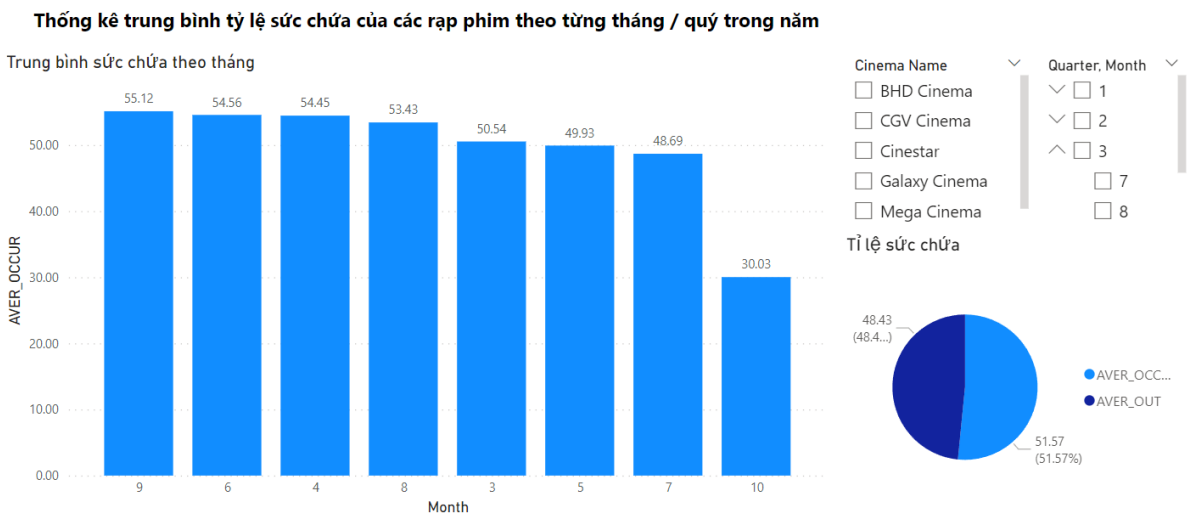


Loại biểu đồ: Clustered Column Chart

- Trục X: Tên thể loại phim (Film Category - film_cate dimension)

- Trục Y: Số vé được sử dụng (Ticket Use - film_sales measure)
- Slicer field: Quarter (DATE MASTER dimension), Month (DATE MASTER dimension), Day (DATE MASTER dimension)

5. Thống kê trung bình tỷ lệ sức chứa của các rạp phim theo từng tháng / quý trong năm.



Bảng Trung bình sức chứa theo tháng

Loại biểu đồ: Clustered Column Chart

- Trục X: Month (Month - DATE MASTER dimension)
- Trục Y: Trung bình sức chứa của các rạp (AVER_OCCUR - film_sales measure)

Biểu đồ Tỉ lệ sức chứa

Loại biểu đồ: Pie Chart

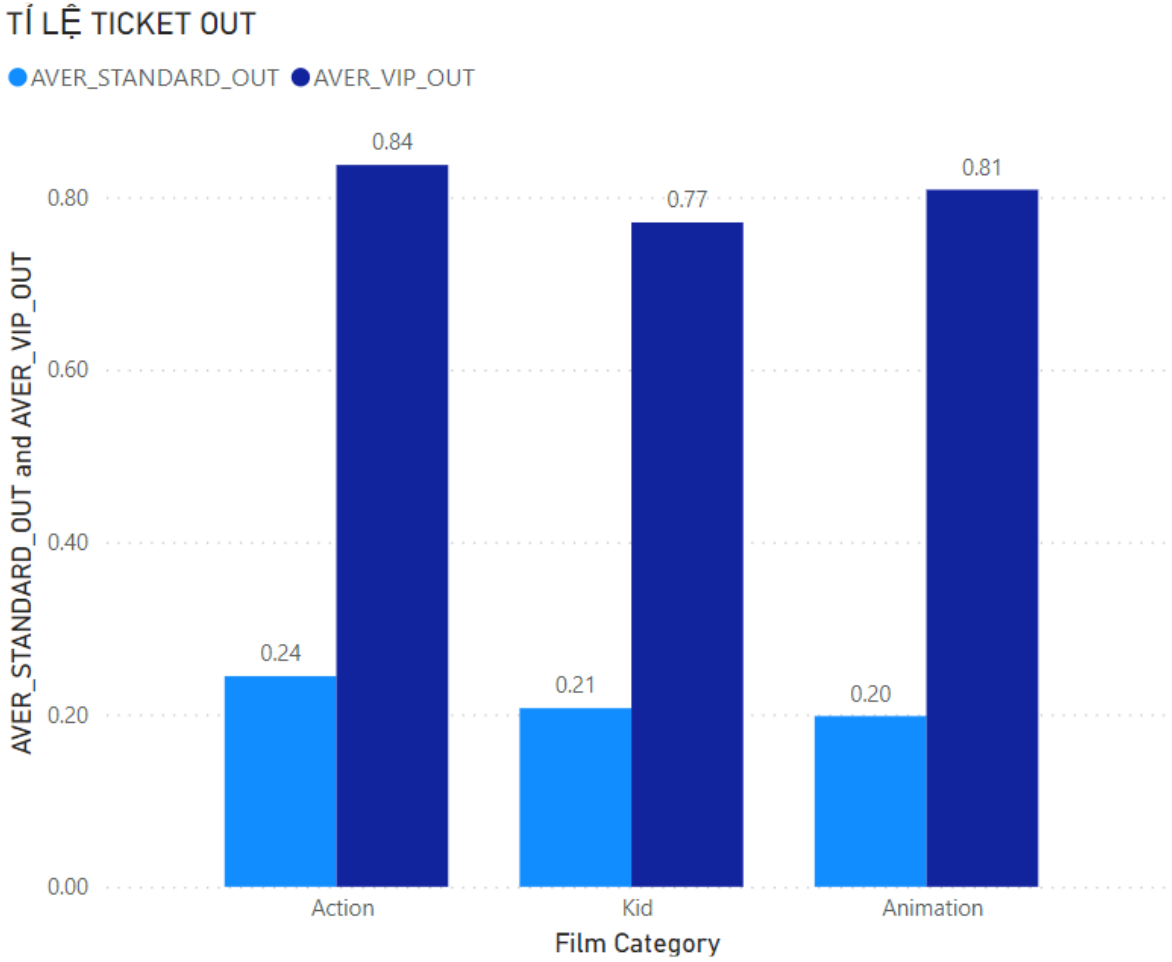
- Values: Trung bình sức chứa của các rạp (AVER_OCCUR - film_sales measure), Trung bình khoảng trống của các rạp (AVER_OUT - film_sales measure)

Slicer Cinema Name Field: Cinema Name (cinema dimension)

Slicer Quarter, Month Field: Quarter (DATE MASTER dimension), Month (DATE MASTER dimension)

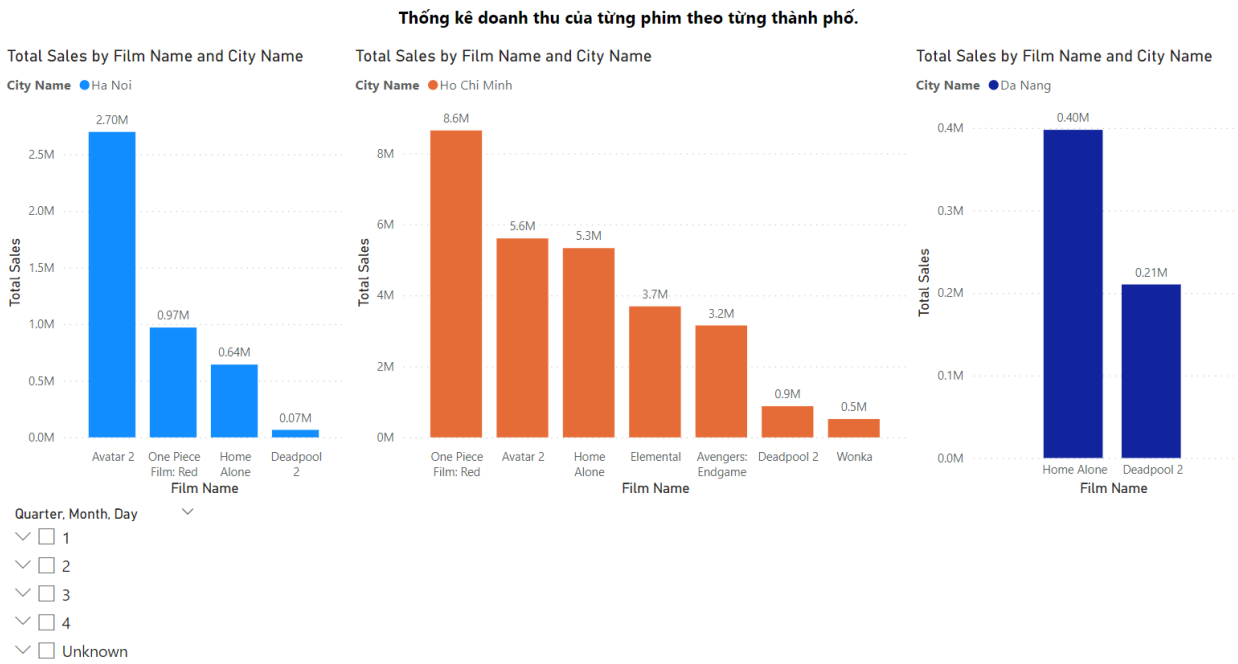
6. Thống kê tỷ lệ vé được trả lại của từng loại vé theo thể loại phim.

Thống kê tỷ lệ vé được trả lại của từng loại vé theo thể loại phim.



- Trục X: Tên thể loại phim (Film Category - film_cate dimension)
- Trục Y: Trung bình vé trả lại của loại vé Standard (AVER_STANDARD_OUT - film_sales measure), trung bình vé trả lại của loại vé VIP (AVER_VIP_OUT - film_sales measure)

7. Thống kê doanh thu của từng phim theo từng thành phố.



Loại biểu đồ: Clustered Column Chart

- Trục X: Tên phim (Film Name - film dimension)
- Trục Y: Doanh thu (Total Sales - film_sales measure)
- Legend: Tên thành phố (City Name - City dimension)
- Thực hiện filter các table theo từng Legend
- Slicer field: Quarter (DATE MASTER dimension), Month (DATE MASTER dimension), Day (DATE MASTER dimension)

VII. Olap

MDX scripts để xem kết quả Report ở dạng bảng:

```
-- Thống kê số vé bán được của từng phim theo ngày, tháng, năm và theo rạp
SELECT NONEMPTY ({[Measures].[Film Sales Count]}*
[Cinema].[Cinema Name].[Cinema Name].ALLMEMBERS) ON COLUMNS,
NONEMPTY(   CROSSJOIN ( [DATE MASTER].[Hierarchy].[Year],
                        [DATE MASTER].[Month].[Month],
                        [DATE MASTER].[Day].[Day]))
ON ROWS
FROM [BI07 MOVIE DDS]
```

			Film Sales Count	Film Sales Count	Film Sales Count	Film Sales Count	Film Sales Count
			BHD Cinema	CGV Cinema	Cinestar	Galaxy Cinema	Mega Cinema
2018	10	6	2	(null)	(null)	(null)	(null)
2018	10	7	2	(null)	1	1	(null)
2018	10	8	1	(null)	(null)	(null)	(null)
2018	3	14	(null)	(null)	(null)	1	(null)
2018	3	15	(null)	(null)	(null)	1	(null)
2018	3	16	(null)	(null)	(null)	1	(null)
2018	3	17	(null)	(null)	(null)	1	(null)
2018	3	18	(null)	(null)	(null)	1	(null)
2018	3	19	(null)	(null)	(null)	1	(null)
2018	3	21	(null)	(null)	(null)	1	(null)
2018	3	22	(null)	(null)	(null)	1	(null)
2018	3	23	(null)	(null)	(null)	1	(null)
2018	3	24	(null)	(null)	(null)	1	(null)
2018	3	25	(null)	(null)	(null)	1	(null)
2018	3	26	(null)	(null)	(null)	1	(null)
2018	3	27	(null)	(null)	(null)	1	(null)
2018	3	28	(null)	(null)	(null)	1	(null)
2018	3	29	(null)	(null)	(null)	1	(null)
2018	3	30	(null)	(null)	(null)	1	(null)

```
-- Thống kê số vé vip và số vé standard được mua theo ngày, tháng, năm.
SELECT NONEMPTY ({  [Measures].[Standard Seats Sold],
                    [Measures].[VIP Seats Sold]}) ON COLUMNS,
NONEMPTY(CROSSJOIN ([DATE MASTER].[Hierarchy].[Year],
                    [DATE MASTER].[Month].[Month],
                    [DATE MASTER].[Day].[Day]))
ON ROWS
FROM [BI07 MOVIE DDS]
```

			Standard Seats Sold	VIP Seats Sold
2018	10	6	652	250
2018	10	7	1250	455
2018	10	8	266	249
2018	3	14	825	883
2018	3	15	1171	626
2018	3	16	257	1161
2018	3	17	98	313
2018	3	18	65	37
2018	3	19	970	175
2018	3	21	303	1081
2018	3	22	538	202
2018	3	23	229	48

```
-- Thống kê số lượng vé vip và standard được mua theo từng rạp
SELECT NONEMPTY ({  [Measures].[Standard Seats Sold],
                    [Measures].[VIP Seats Sold]}) ON COLUMNS,
NONEMPTY([Cinema].[Cinema Name].[Cinema Name])
ON ROWS
FROM [BI07 MOVIE DDS]
```

	Standard Seats Sold	VIP Seats Sold
BHD Cinema	40697	34230
CGV Cinema	86813	79784
Cinestar	8113	4210
Galaxy Cinema	152796	148232
Mega Cinema	1116	438

```
-- Thống kê số lượng mua vé thật sự được bán ra của từng
SELECT NONEMPTY ({[Measures].[Ticket Use]}) ON COLUMNS,
NONEMPTY([Film Cate].[Film Category].[Film Category])
ON ROWS
FROM [BI07 MOVIE DDS]
```

	Ticket Use
Action	205782
Animation	224259
Kid	123596

```
-- Thống kê trung bình tỷ lệ sức chứa của các rạp phim theo từng tháng
-- / quý trong năm
SELECT NONEMPTY ({[Measures].[AVER_OCCUR]}) ON COLUMNS,
NONEMPTY(
CROSSJOIN ( [DATE MASTER].[Hierarchy].[Year],
            [DATE MASTER].[Quarter].[Quarter],
            [DATE MASTER].[Month].[Month]))
ON ROWS
FROM [BI07 MOVIE DDS]
```

			AVER_OCCUR
2018	1	3	50.54
2018	2	4	54.45
2018	2	5	49.93
2018	2	6	54.56
2018	3	7	48.69
2018	3	8	53.43
2018	3	9	55.12
2018	4	10	30.03

```
-- Thống kê tỷ lệ vé được trả lại của từng loại vé theo thể loại phim
SELECT NONEMPTY ({  [Measures].[AVER_STANDARD_OUT],
                    [Measures].[AVER_VIP_OUT]}) ON COLUMNS,
NONEMPTY([Film Cate].[Film Category].[Film Category])
ON ROWS
FROM [BI07 MOVIE DDS]
```

	AVER_STANDARD_OUT	AVER_VIP_OUT
Action	0.00	0.01
Animation	0.00	0.01
Kid	0.00	0.01

```sql
-- Thống kê doanh thu của từng phim theo từng thành phố.
SELECT NONEMPTY ({  [Measures].[Total Sales]}*
                    [Film].[Film Name].[Film Name].ALLMEMBERS) ON COLUMNS,
NONEMPTY([City].[City Name].[City Name])
ON ROWS
FROM [BI07 MOVIE DDS]
```

	Total Sales	Total Sales	Total Sales	Total Sales	Total Sales	Total Sales	Total Sales
	Avatar 2	Avengers: Endgame	Deadpool 2	Elemental	Home Alone	One Piece Film: Red	Wonka
Da Nangg	(null)	(null)	210325	(null)	397740	(null)	(null)
Ha Noi	2698130	(null)	66260	(null)	643251	970310	(null)
Ho Chi Minh	5604195	3152860	878710	3689905	5330991	8643805	517282

VIII. Data Mining

1. Dự đoán doanh thu từng thể loại phim nếu mỗi thể loại có tổng lượt chiếu ở từng rạp là 35 suất.

Trước hết tôi sẽ tiến hành kết nối và lấy dữ liệu mình cần tương tự như khi làm supermarket_mining:

Câu 1

```
let
    Source = AnalysisServices.Database("ABC1", "SSAS_film", [TypedMeasureColumns=true, Implementation="2.0"]),
    #"BI07 MOVIE DDS1" = Source{[Id="BI07 MOVIE DDS"]}[Data],
    #"BI07 MOVIE DDS2" = #"BI07 MOVIE DDS1"{[Id="BI07 MOVIE DDS"]}[Data],
    #"Added Items" = Cube.Transform(#"BI07 MOVIE DDS2",
        {
            {Cube.AddAndExpandDimensionColumn, "[Cinema]", {"[Cinema].[Cinema Name].[Cinema Name]"}, {"Cinema.Cinema Name"}},
            {Cube.AddAndExpandDimensionColumn, "[City]", {"[City].[City Name].[City Name]"}, {"City.City Name"}},
            {Cube.AddAndExpandDimensionColumn, "[Film Cate]", {"[Film Cate].[Film Category].[Film Category]"}, {"Film Cate.Film Category"}},
            {Cube.AddMeasureColumn, "Film Sales Count", "[Measures].[Film Sales Count]"},
            {Cube.AddMeasureColumn, "Total Sales", "[Measures].[Total Sales]"}
        })
in
    #"Added Items"
```

Đây là dữ liệu mà tôi nhận được:

	A	B	C	D	E
1	Cinema.Cinema Name	City.City Name	Film Cate.Film Category	Film Sales Count	Total Sales
2	BHD Cinema	Ha Noi	Action	60	2698130
3	BHD Cinema	Ha Noi	Animation	28	970310
4	BHD Cinema	Ha Noi	Kid	17	643251
5	CGV Cinema	Ho Chi Minh	Action	37	3011090
6	CGV Cinema	Ho Chi Minh	Animation	60	4928320
7	CGV Cinema	Ho Chi Minh	Kid	16	1271783
8	Cinestar	Da Nang	Action	5	210325
9	Cinestar	Da Nang	Kid	12	397740
10	Galaxy Cinema	Ho Chi Minh	Action	124	6624675
11	Galaxy Cinema	Ho Chi Minh	Animation	128	7405390
12	Galaxy Cinema	Ho Chi Minh	Kid	77	4576490
13	Mega Cinema	Ha Noi	Action	4	66300
14					

Sau đó tôi sẽ tiến hành đọc dữ liệu vào python

```python
df1 = pd.read_excel("film_mining.xlsx", sheet_name= "Câu 1")

mapping = {'Cinema.Cinema Name': 'Cinema Name','City.City Name': 'City Name',
           'Film Cate.Film Category': 'Film Category',
           'Film Sales Count':'total screening capacity'}
df1 = df1.rename(columns= mapping)
df1.head()
```

[5] Python

	Cinema Name	City Name	Film Category	total screening capacity	Total Sales
0	BHD Cinema	Ha Noi	Action	60	2698130
1	BHD Cinema	Ha Noi	Animation	28	970310
2	BHD Cinema	Ha Noi	Kid	17	643251
3	CGV Cinema	Ho Chi Minh	Action	37	3011090
4	CGV Cinema	Ho Chi Minh	Animation	60	4928320

Để dự đoán doanh thu dễ dàng hơn tôi sẽ tính doanh thu của từng rạp ở từng thể loại của mỗi suất chiếu, sau đó sẽ tiến hành thực hiện LinearRegression để dự đoán:

```python
df1['Revenue_Per_Screening'] = df1['Total Sales'] / df1['total screening capacity']
features = ['total screening capacity']
target = 'Revenue_Per_Screening'
model = LinearRegression()
train_data = df1[features + [target]]

model.fit(train_data[features], train_data[target])

df1['Predicted_Revenue'] = model.predict(df1[features] * 35)

print(df1[['Cinema Name', 'Film Category', 'Predicted_Revenue']])
```

Và đây là kết quả dự đoán:

	Cinema Name	Film Category	Predicted_Revenue
0	BHD Cinema	Action	377870.362829
1	BHD Cinema	Animation	200018.162239
2	BHD Cinema	Kid	138881.468286
3	CGV Cinema	Action	250039.093655
4	CGV Cinema	Animation	377870.362829
5	CGV Cinema	Kid	133323.587018
6	Cinestar	Action	72186.893065
7	Cinestar	Kid	111092.061944
8	Galaxy Cinema	Action	733574.764009
9	Galaxy Cinema	Animation	755806.289082
10	Galaxy Cinema	Kid	472354.344392
11	Mega Cinema	Action	66629.011796

2. Dự đoán giá vé của BHD Cinema nếu mỗi loại ghế giá vé sẽ tăng 5 đồng

Trước hết tôi sẽ tiến hành lấy dữ liệu mình cần từ các bảng Dim & Fact

Advanced Editor

Câu 2

Display Options

```
let
    Source = AnalysisServices.Database("ABC1", "SSAS_film", [TypedMeasureColumns=true, Implementation="2.0"]),
    #"BI07 MOVIE DDS1" = Source{[Id="BI07 MOVIE DDS"]}[Data],
    #"BI07 MOVIE DDS2" = #"BI07 MOVIE DDS1"{[Id="BI07 MOVIE DDS"]}[Data],
    #"Added Items" = Cube.Transform(#"BI07 MOVIE DDS2",
        {
            {Cube.AddAndExpandDimensionColumn, "[Cinema]", {"[Cinema].[Cinema Name].[Cinema Name]", "[Cinema].[Standard Seats Price].[Standard Seats Price]", "[Cinema].[VIP Seats Price].[VIP Seats Price]"}, {"Cinema.Cine
            {Cube.AddMeasureColumn, "Standard Seats Out", "[Measures].[Standard Seats Out]"},
            {Cube.AddMeasureColumn, "Standard Seats Sold", "[Measures].[Standard Seats Sold]"},
            {Cube.AddMeasureColumn, "VIP Seats Out", "[Measures].[VIP Seats Out]"},
            {Cube.AddMeasureColumn, "VIP Seats Sold", "[Measures].[VIP Seats Sold]"}
        }),
    #"Added Items1" = Cube.Transform(#"Added Items",
        {
            {Cube.AddAndExpandDimensionColumn, "[DATE MASTER]", {"[DATE MASTER].[Month].[Month]"}, {"DATE MASTER.Month"}}
        })
in
    #"Added Items1"
```

Sau đó đọc dữ liệu mình vừa lấy vào python:

```python
df2 = pd.read_excel("film_mining.xlsx", sheet_name= "Câu 2")

mapping = {'Cinema.Cinema Name': 'Cinema Name','Cinema.Standard Seats Price': 'Standard Seats Price','Cinema.VIP Seats Price': 'VIP Seats Price','DATE MASTER.Month':'Month'}
df2 = df2.rename(columns= mapping)
df2['Standard Seats Use'] = df2['Standard Seats Sold']- df2['Standard Seats Out']
df2['VIP Seats Use'] = df2['VIP Seats Sold'] - df2['VIP Seats Out']
df2.head()
```

[36] Python

	Cinema Name	Standard Seats Price	VIP Seats Price	Standard Seats Out	Standard Seats Sold	VIP Seats Out	VIP Seats Sold	Month	Standard Seats Use	VIP Seats Use
0	BHD Cinema	25	45	3	1196	16	569	10	1193	553
1	BHD Cinema	25	45	0	895	22	2018	3	895	1996
2	BHD Cinema	25	45	3	1319	7	779	4	1316	772
3	BHD Cinema	25	45	28	9264	84	7595	5	9236	7511
4	BHD Cinema	25	45	35	13233	115	9885	6	13198	9770

Sau đó tôi tiến hành dự đoán số lượt vé Standard Seat Use & VIP Seat Use thông qua LinearRegression sau khi tăng giá vé lên:

```
current_bhd_standard_seats_use = df2[df2['Cinema Name'] == 'BHD Cinema']['Standard Seats Use'].sum()
current_bhd_vip_seats_use = df2[df2['Cinema Name'] == 'BHD Cinema']['VIP Seats Use'].sum()

features = ['Standard Seats Price']
target_standard_seats_use = 'Standard Seats Use'
target_vip_seats_use = 'VIP Seats Use'

df_bhd = df2[df2['Cinema Name'] == 'BHD Cinema']

model_bhd_standard_seats = LinearRegression()
model_bhd_standard_seats.fit(df_bhd[features], df_bhd[target_standard_seats_use])

model_bhd_vip_seats = LinearRegression()
model_bhd_vip_seats.fit(df_bhd[features], df_bhd[target_vip_seats_use])

price_increase = 5
prediction_data = pd.DataFrame({
    'Standard Seats Price': [df_bhd['Standard Seats Price'].iloc[0] + price_increase],
})

predicted_bhd_standard_seats_use = model_bhd_standard_seats.predict(prediction_data)[0]
predicted_bhd_vip_seats_use = model_bhd_vip_seats.predict(prediction_data)[0]
```

Tuy nhiên để dự đoán doanh thu thì tôi dần lấy số lượng vé đã mua nhân giá vé:

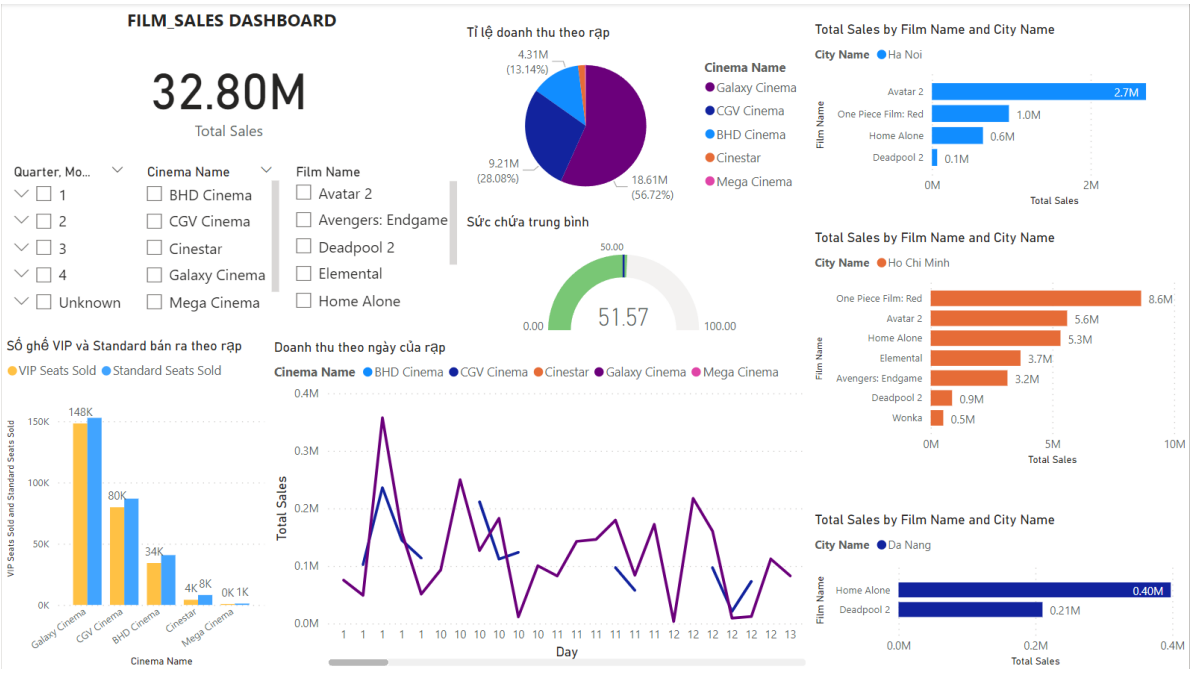
```
# Tính toán doanh thu dự kiến
new_predicted_revenue = predicted_bhd_standard_seats_use * prediction_data['Standard Seats Price'].iloc[0] + predicted_bhd_vip_seats_use * (df_bhd['VIP Seats Price'].iloc[0] + price

print(f'Dự đoán số lượt vé Standard được bán ra của rạp BHD Cinema sau khi tăng giá là: {predicted_bhd_standard_seats_use}')
print(f'Dự đoán số lượt vé VIP được bán ra của rạp BHD Cinema sau khi tăng giá là: {predicted_bhd_vip_seats_use}')
print(f'Dự đoán doanh thu của rạp BHD Cinema sau khi tăng giá là: {new_predicted_revenue}')
```

Và đây là kết quả trả về

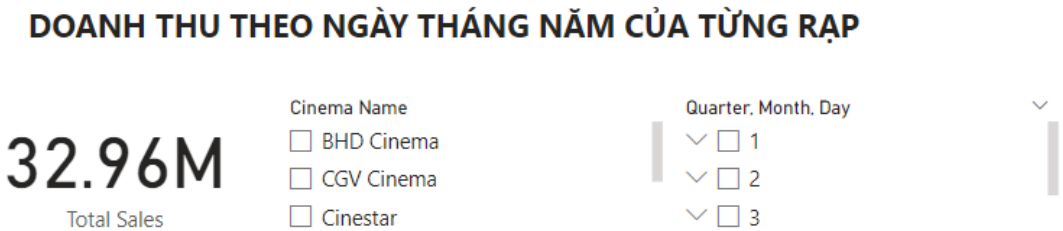
```
Dự đoán số lượt vé Standard được bán ra của rạp BHD Cinema sau khi tăng giá là: 5071.375
Dự đoán số lượt vé VIP được bán ra của rạp BHD Cinema sau khi tăng giá là: 4227.75
Dự đoán doanh thu của rạp BHD Cinema sau khi tăng giá là: 363528.75
```

IX. Dashboard

Dashboard tổng kết các report:



Dashboard doanh thu theo ngày tháng năm của từng rạp



Day	BHD Cinema	CGV Cinema	Cinestar	Galaxy Cinema	Mega Cinema	Total
26	69910	231538		302591		**604039**
1		236089		357473		**593562**
27	54670	196465		291668		**542803**
29	51470	264523		203062		**519055**
31	110544	81368		311493		**503405**
7	113600	129756		250313		**493669**
2		217969		251066		**469035**
29	57890	131340		252026		**441256**
7	138232	121200		172478		**431910**
7	76195	152159	8250	194524		**431128**
14	51402	111084	25089	242653		**430228**
28	69540	127143		226545		**423228**
26	80969	140890		183021		**404880**
28		111673		287072		**398745**
25	11060	168420		216042		**397332**
Total	**4327912**	**9278338**	**608065**	**18681993**	**66300**	**32962608**

D. Information

I. Link

BI07_Project

Demo

Trello

Drawio

II. Reference:

New Dataset source: https://www.kaggle.com/datasets/arashnic/cinema-ticket?fbclid=IwAR03Bcd_gIuXjmayAeEE5RpiLkHAHVztFr7k1zI0hHSbOHTLwIIBF3eZH4M