

**Brazilian E-commerce
public dataset
by Olist**

Assignment 1

IT Number – IT19157306

Submitted by – P.A.N.D. Panditharathna

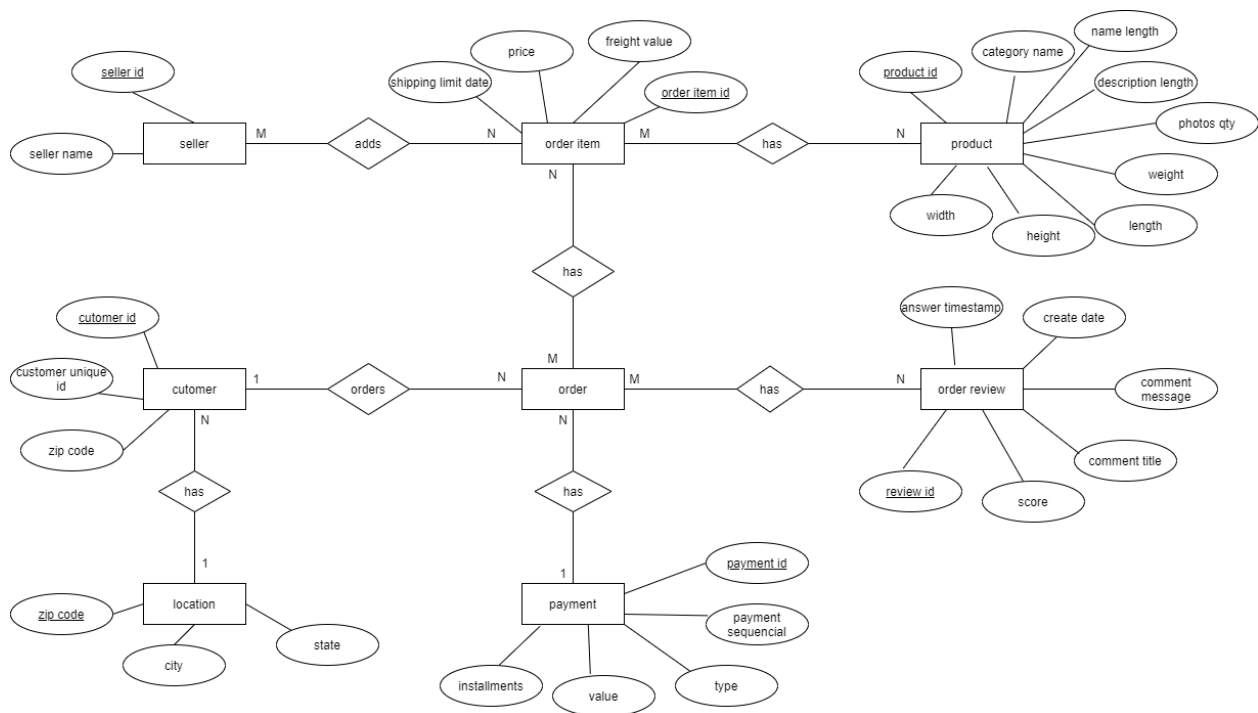
Data set selection

The selected data set source is a collection of transactional data. The link to the source data set mentioned below.

[Brazilian E-Commerce Public Dataset by Olist | Kaggle](#)

This data set shows about orders made at online store named Olist. Olist connects small businesses from all over Brazil. Those merchants are able to sell their products through the olist store and ship them directly to the customers location. After a customer purchases the product from olist store, a seller gets notified to fulfill that order. Once the customer receives the product or estimate delivery date is due, the customer gets a satisfaction survey by email where he can give a note for the purchase experience and write down some comments.

ER Diagram



Preparation of Data source

This data set gives all files are CSV files. I decided to change their file type to CSV, Excel, Text format.

1. CSV files

- olist_customers_dataset
- olist_geolocation_dataset
- olist_order_items_dataset

2. Excel files

- olist_order_payments_dataset
- olist_order_reviews_dataset
- olist_orders_dataset

3. Text files

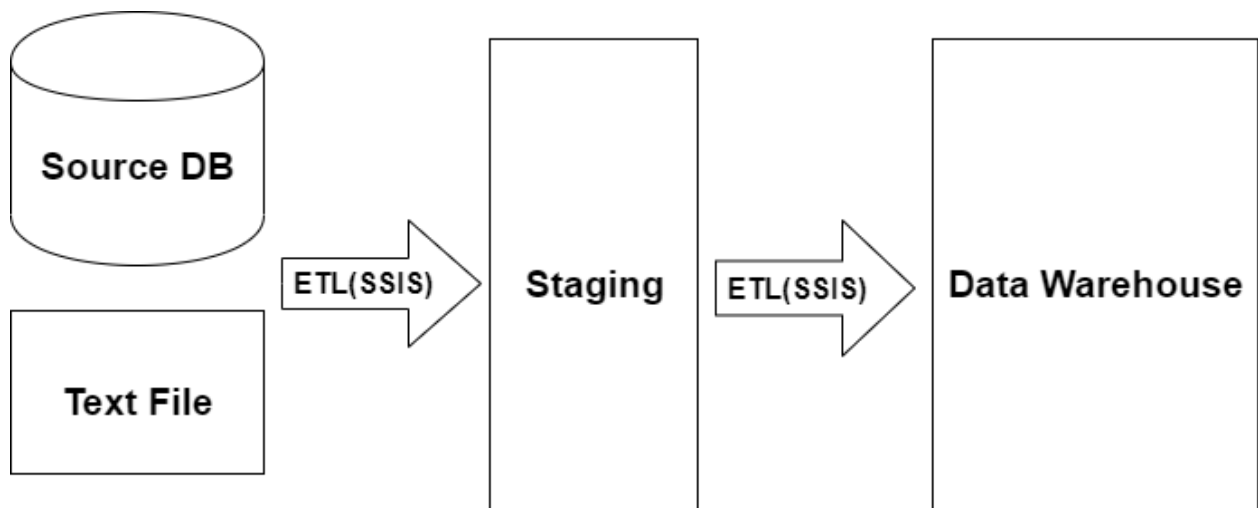
- olist_products_dataset
- olist_sellers_dataset

Description of the data set

Table Name	Column Name	Data Type	Description
product	Product_id	nvarchar(50)	Details of products
	Product_category_name	nvarchar(50)	
	product_name_lenght	nvarchar(50)	
	product_description_lenght	nvarchar(50)	
	product_photos_qty	nvarchar(50)	
	product_weight_g	nvarchar(50)	
	product_length_cm	nvarchar(50)	
	product_height_cm	nvarchar(50)	
	product_width_cm	nvarchar(50)	
orders	order_id	nvarchar(100)	Details of orders
	customer_id	nvarchar(255)	
	order_status	nvarchar(255)	
	order_purchase_timestamp	datetime	
	order_approved_at	datetime	
	order_delivered_carrier_date	datetime	
	order_delivered_customer_date	datetime	
	order_estimated_delivery_date	datetime	
	order_id	nvarchar(100)	

order_items	order_item_id	nvarchar(50)	Details of order items
	product_id	nvarchar(50)	
	seller_id	nvarchar(50)	
	shipping_limit_date	datetime	
	price	nvarchar(50)	
	freight_value	nvarchar(50)	
payment	Payment_id	float	Details of order purchases
	order_id	int	
	payment_sequential	nvarchar(100)	
	payment_type	float	
	payment_installments	float	
	payment_value	float	
order_review	review_id	nvarchar(255)	Details of customer reviews about product
	order_id	nvarchar(100)	
	review_score	nvarchar(255)	
	review_comment_title	nvarchar(255)	
	review_comment_message	nvarchar(255)	
	review_creation_date	datetime	
	review_answer_timestamp	datetime	
customer	customer_id	nvarchar(50)	Details of customers
	customer_unique_id	nvarchar(50)	
	customer_zip_code_prefix	int	
	customer_city	nvarchar(50)	
	customer_state	nvarchar(50)	
seller	seller_id	nvarchar(50)	Details of sellers
	seller_name	nvarchar(50)	
geolocation	zip_code_prefix	int	Location details
	city	nvarchar(50)	
	state	nvarchar(50)	

Solution Architecture



Components

Data sources

Data sources can be operational system, disintegrated data, external sources, non-digitalized data, etc. I import Excel files and CSV files into the database. Text files are imported to the database through ETL process.

- Text file
- CSV files
- Excel files

Processing

In processing component, we are doing ETL task.

ETL- Extract, Transform and Load

- Extract- Reading data from the data sources.
- Transform- The process of converting extracted data to the new form, it can be placed into a new database.
- Load- The process of writing data to the target database.
 - ❖ Surrogate key assign
 - ❖ Indexing
 - ❖ FK constraint check

Data Warehouse

In this have dimensional modelling

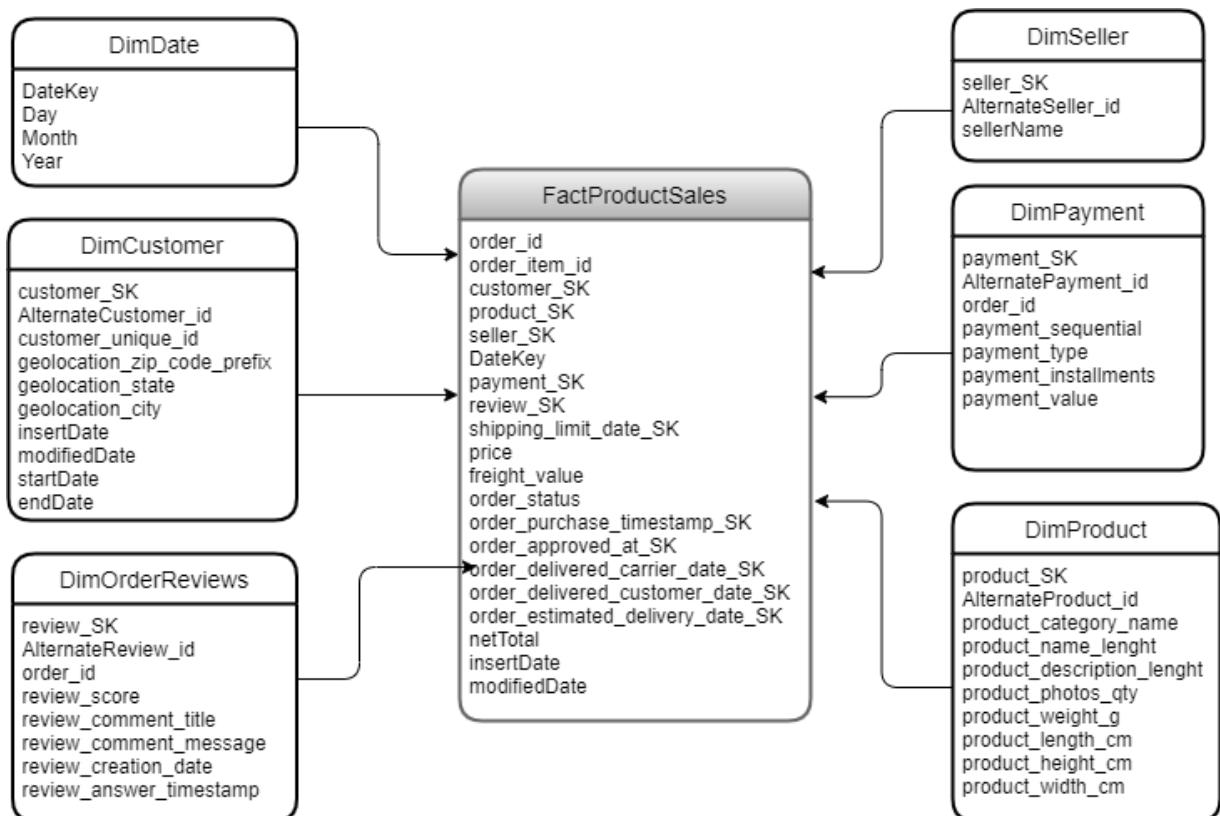
- Describe about facts and dimensions.

Have many schemas

- Star
- Snowflake
- Galaxy

Data Warehouse design development

we can select fact tables and dimension tables from the source files. I decided to seller, customer, payment, product, order review files get as dimension tables in addition we need date dimension table to get date details. So, I get that too. In my fact table, there are order details and order item details. In my scenario I decided to get customer dimension as slowly changing dimension. Because customer has location details. I have only one fact table so, this is star schema dimensional model.



Assumptions –

- Customer is slowly changing dimension.
- I get netTotal as function which has sum of price and freight value.

Steps in ETL process

Step 1- Import excel and CSV files to the Olist_DB in SQL server.

Step 2 - Create new databases Olist_Staging and Olist_DW in SQL server.

Step 3 - Create SSIS solution called SSISOlist.

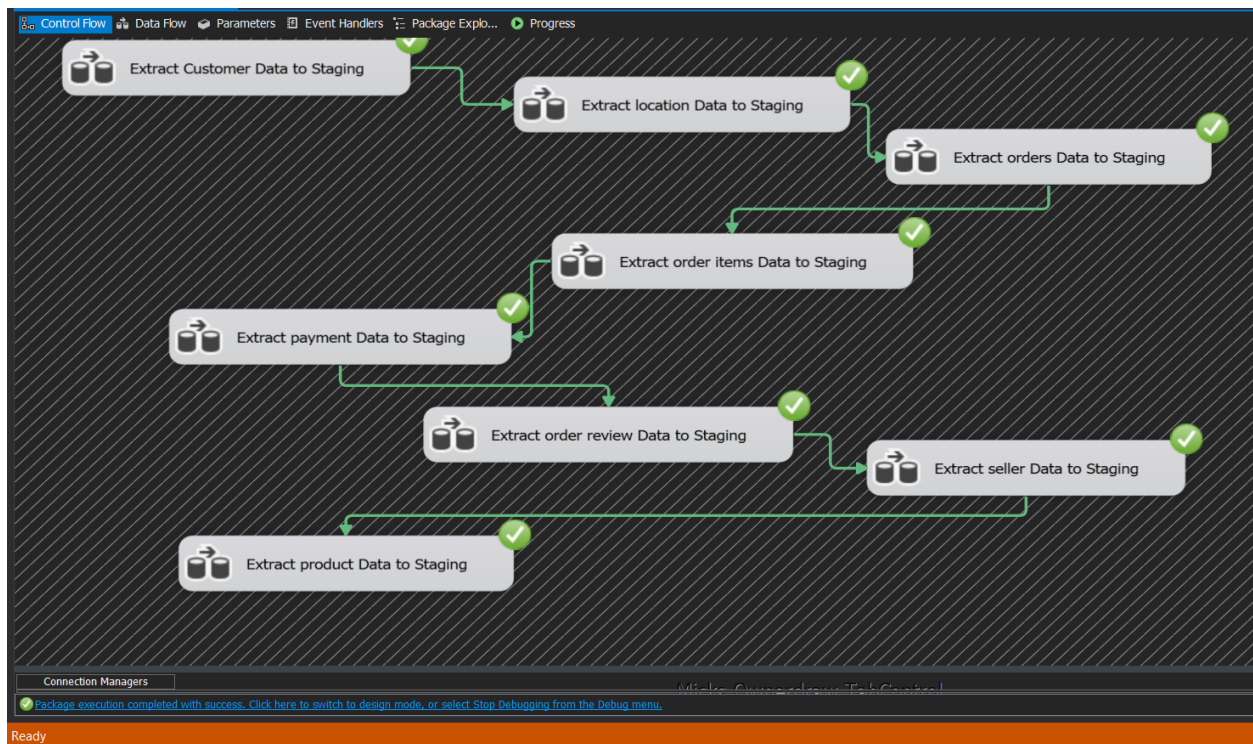
Rename package called package.dtsx to Olist ETL Staging.dtsx.

Extract data from Olist_DB to Olist_Staging.

Extract Flat Files (seller,product) to Olist_Staging.

Used OLE DB Source and OLE DB Destinations to extract data.

Used Flat File Source and OLE DB Destinations to extract data in olist_products_dataset.txt and olist_sellers_dataset .txt



Step 4 – We use a code to get Date dimension. It shows below.

1

```
CREATE TABLE [dbo].[DimDate]
(
    [DateKey] INT primary key,
    [Date] DATETIME,
    [FullDateUK] CHAR(10), -- Date in dd-MM-yyyy format
    [FullDateUSA] CHAR(10), -- Date in MM-dd-yyyy format
    [DayOfMonth] VARCHAR(2), -- Field will hold day number of Month
    [DaySuffix] VARCHAR(4), -- Apply suffix as 1st, 2nd, 3rd etc
    [DayName] VARCHAR(9), -- Contains name of the day, Sunday, Monday
    [DayOfWeekUSA] CHAR(1), -- First Day Sunday=1 and Saturday=7
    [DayOfWeekUK] CHAR(1), -- First Day Monday=1 and Sunday=7
    [DayOfWeekInMonth] VARCHAR(2), -- 1st Monday or 2nd Monday in Month
    [DayOfWeekInYear] VARCHAR(2),
    [DayOfQuarter] VARCHAR(3),
    [DayOfYear] VARCHAR(3),
    [WeekOfMonth] VARCHAR(1), -- Week Number of Month
    [WeekOfQuarter] VARCHAR(2), -- Week Number of the Quarter
    [WeekOfYear] VARCHAR(2), -- Week Number of the Year
    [Month] VARCHAR(2), -- Number of the Month 1 to 12
    [MonthName] VARCHAR(9), -- January, February etc
    [MonthOfQuarter] VARCHAR(2), -- Month Number belongs to Quarter
    [Quarter] CHAR(1),
    [QuarterName] VARCHAR(9), -- First, Second..
    [Year] CHAR(4), -- Year value of Date stored in Row
    [YearName] CHAR(7), -- CY 2012, CY 2013
    [MonthYear] CHAR(10), -- Jan-2013, Feb-2013
    [MMYYYY] CHAR(6),
    [FirstDayOfMonth] DATE,
    [LastDayOfMonth] DATE,
    [FirstDayOfQuarter] DATE,
    [LastDayOfQuarter] DATE,
    [FirstDayOfYear] DATE,
    [LastDayOfYear] DATE,
    [IsHolidaySL] BIT, -- Flag 1-National Holiday, 0-No National Holiday
    [IsWeekday] BIT, -- 0-Week End, 1-Week Day
    [HolidaySL] VARCHAR(50), -- Name of Holiday in US
    [IsCurrentDay] int, -- Current day=1 else = 0
    [IsDataAvailable] int, -- data available for the day = 1, no data ava.
    [IsLatestDataAvailable] int
)
```

2

```

/*****
--Specify Start Date and End date here
--Value of Start Date Must be Less than Your End Date

DECLARE @StartDate DATETIME = '01/01/1990' --Starting value of Date Range
DECLARE @EndDate DATETIME = '01/01/2099' --End Value of Date Range

--Temporary Variables To Hold the Values During Processing of Each Date of Year
DECLARE
    @DayOfWeekInMonth INT,
    @DayOfWeekInYear INT,
    @DayOfQuarter INT,
    @WeekOfMonth INT,
    @CurrentYear INT,
    @CurrentMonth INT,
    @CurrentQuarter INT

/*Table Data type to store the day of week count for the month and year*/
DECLARE @DayOfWeek TABLE (DOM INT, MonthCount INT, QuarterCount INT, YearCount INT)

INSERT INTO @DayOfWeek VALUES (1, 0, 0, 0)
INSERT INTO @DayOfWeek VALUES (2, 0, 0, 0)
INSERT INTO @DayOfWeek VALUES (3, 0, 0, 0)
INSERT INTO @DayOfWeek VALUES (4, 0, 0, 0)
INSERT INTO @DayOfWeek VALUES (5, 0, 0, 0)
INSERT INTO @DayOfWeek VALUES (6, 0, 0, 0)
INSERT INTO @DayOfWeek VALUES (7, 0, 0, 0)

--Extract and assign various parts of Values from Current Date to Variable

DECLARE @CurrentDate AS DATETIME = @StartDate
SET @CurrentMonth = DATEPART(MM, @CurrentDate)
SET @CurrentYear = DATEPART(YY, @CurrentDate)
SET @CurrentQuarter = DATEPART(QQ, @CurrentDate)

```



```

/*****
--Proceed only if Start Date(Current date ) is less than End date you specified above

WHILE @CurrentDate < @EndDate
BEGIN

/*Begin day of week logic*/

/*Check for Change in Month of the Current date if Month changed then
Change variable value*/
IF (@CurrentMonth != DATEPART(MM, @CurrentDate))
BEGIN
    UPDATE @DayOfWeek
    SET MonthCount = 0
    SET @CurrentMonth = DATEPART(MM, @CurrentDate)
END

/* Check for Change in Quarter of the Current date if Quarter changed then change
Variable value*/

IF (@CurrentQuarter != DATEPART(QQ, @CurrentDate))
BEGIN
    UPDATE @DayOfWeek
    SET QuarterCount = 0
    SET @CurrentQuarter = DATEPART(QQ, @CurrentDate)
END

/* Check for Change in Year of the Current date if Year changed then change
Variable value*/

IF (@CurrentYear != DATEPART(YY, @CurrentDate))
BEGIN
    UPDATE @DayOfWeek
    SET YearCount = 0
    SET @CurrentYear = DATEPART(YY, @CurrentDate)
END

-- Set values in table data type created above from variables

```

```

UPDATE @DayOfWeek
SET
    MonthCount = MonthCount + 1,
    QuarterCount = QuarterCount + 1,
    YearCount = YearCount + 1
WHERE DOW = DATEPART(DW, @CurrentDate)

SELECT
    @DayOfWeekInMonth = MonthCount,
    @DayOfQuarter = QuarterCount,
    @DayOfWeekInYear = YearCount
FROM @DayOfWeek
WHERE DOW = DATEPART(DW, @CurrentDate)

/*End day of week logic*/

/* Populate Your Dimension Table with values*/

INSERT INTO [dbo].[DimDate]
SELECT
    CONVERT (char(8),@CurrentDate,112) as DateKey,
    @CurrentDate AS Date,
    CONVERT (char(10),@CurrentDate,103) as FullDateUK,
    CONVERT (char(10),@CurrentDate,101) as FullDateUSA,
    DATEPART(DO, @CurrentDate) AS DayOfMonth,
    --Apply Suffix values like 1st, 2nd 3rd etc..
    CASE
        WHEN DATEPART(DO,@CurrentDate) IN (11,12,13)
        THEN CAST(DATEPART(DO,@CurrentDate) AS VARCHAR) + 'th'
        WHEN RIGHT(DATEPART(DO,@CurrentDate),1) = 1
        THEN CAST(DATEPART(DO,@CurrentDate) AS VARCHAR) + 'st'
        WHEN RIGHT(DATEPART(DO,@CurrentDate),1) = 2
        THEN CAST(DATEPART(DO,@CurrentDate) AS VARCHAR) + 'nd'
        WHEN RIGHT(DATEPART(DO,@CurrentDate),1) = 3
        THEN CAST(DATEPART(DO,@CurrentDate) AS VARCHAR) + 'rd'
        ELSE CAST(DATEPART(DO,@CurrentDate) AS VARCHAR) + 'th'
    END AS DaySuffix,
    DATENAME(DW, @CurrentDate) AS DayName,

```

```

CASE DATEPART(DW, @CurrentDate)
    WHEN 1 THEN 7
    WHEN 2 THEN 1
    WHEN 3 THEN 2
    WHEN 4 THEN 3
    WHEN 5 THEN 4
    WHEN 6 THEN 5
    WHEN 7 THEN 6
END
AS DayOfWeekUK,

@DayOfWeekInMonth AS DayOfWeekInMonth,
@DayOfWeekInYear AS DayOfWeekInYear,
@DayOfQuarter AS DayOfQuarter,
DATEPART(DY, @CurrentDate) AS DayOfYear,
DATEPART(WK, @CurrentDate) + 1 - DATEPART(WK, CONVERT(VARCHAR,
DATEPART(MM, @CurrentDate)) + '/1/' + CONVERT(VARCHAR,
DATEPART(YY, @CurrentDate))) AS WeekOfMonth,
(DATEIFF(DO, DATEADD(QQ, DATEIFF(QQ, 0, @CurrentDate), 0),
@CurrentDate) / 7) + 1 AS WeekOfQuarter,
DATEPART(WK, @CurrentDate) AS WeekOfYear,
DATEPART(MM, @CurrentDate) AS Month,
DATENAME(MM, @CurrentDate) AS MonthName,
CASE
    WHEN DATEPART(MM, @CurrentDate) IN (1, 4, 7, 10) THEN 1
    WHEN DATEPART(MM, @CurrentDate) IN (2, 5, 8, 11) THEN 2
    WHEN DATEPART(MM, @CurrentDate) IN (3, 6, 9, 12) THEN 3
    END AS MonthOfQuarter,
DATEPART(QQ, @CurrentDate) AS Quarter,
CASE DATEPART(QQ, @CurrentDate)
    WHEN 1 THEN 'First'
    WHEN 2 THEN 'Second'
    WHEN 3 THEN 'Third'
    WHEN 4 THEN 'Fourth'
    END AS QuarterName,
DATEPART(YEAR, @CurrentDate) AS Year,
'CY ' + CONVERT(VARCHAR, DATEPART(YEAR, @CurrentDate)) AS YearName,
LEFT(DATENAME(MM, @CurrentDate), 3) + '-' + CONVERT(VARCHAR,
DATEPART(YY, @CurrentDate)) AS MonthYear,
RIGHT('0' + CONVERT(VARCHAR, DATEPART(MM, @CurrentDate)), 2) +
CONVERT(VARCHAR, DATEPART(YY, @CurrentDate)) AS MMYYYY,

```

```

    WHEN 3 THEN 'Third'
    WHEN 4 THEN 'Fourth'
    END AS QuarterName,
DATEPART(YEAR, @CurrentDate) AS Year,
'CY ' + CONVERT(VARCHAR, DATEPART(YEAR, @CurrentDate)) AS YearName,
LEFT(DATENAME(MM, @CurrentDate), 3) + '-' + CONVERT(VARCHAR,
DATEPART(YY, @CurrentDate)) AS MonthYear,
RIGHT('0' + CONVERT(VARCHAR, DATEPART(MM, @CurrentDate)), 2) +
CONVERT(VARCHAR, DATEPART(YY, @CurrentDate)) AS MMYYYY,
CONVERT(DATETIME, CONVERT(DATE, DATEADD(DO, - (DATEPART(DO,
@CurrentDate) - 1), @CurrentDate))) AS FirstDayOfMonth,
CONVERT(DATETIME, CONVERT(DATE, DATEADD(DO, - (DATEPART(DO,
(DATEADD(MM, 1, @CurrentDate))), DATEADD(MM, 1,
@CurrentDate)))) AS LastDayOfMonth,
DATEADD(QQ, DATEIFF(QQ, 0, @CurrentDate), 0) AS FirstDayOfQuarter,
DATEADD(QQ, DATEIFF(QQ, -1, @CurrentDate), -1) AS LastDayOfQuarter,
CONVERT(DATETIME, '01/01/' + CONVERT(VARCHAR, DATEPART(YY,
@CurrentDate))) AS FirstDayOfYear,
CONVERT(DATETIME, '12/31/' + CONVERT(VARCHAR, DATEPART(YY,
@CurrentDate))) AS LastDayOfYear,
NULL AS IsHolidaySL,
CASE DATEPART(DW, @CurrentDate)
    WHEN 1 THEN 0
    WHEN 2 THEN 1
    WHEN 3 THEN 1
    WHEN 4 THEN 1
    WHEN 5 THEN 1
    WHEN 6 THEN 1
    WHEN 7 THEN 0
    END AS IsWeekday,
NULL AS HolidaySL, (case when (@CurrentDate = convert(date, sysdatetime())) then 1 else 0 end), 0, 0

SET @CurrentDate = DATEADD(DO, 1, @CurrentDate)
END

/*****
/*****/

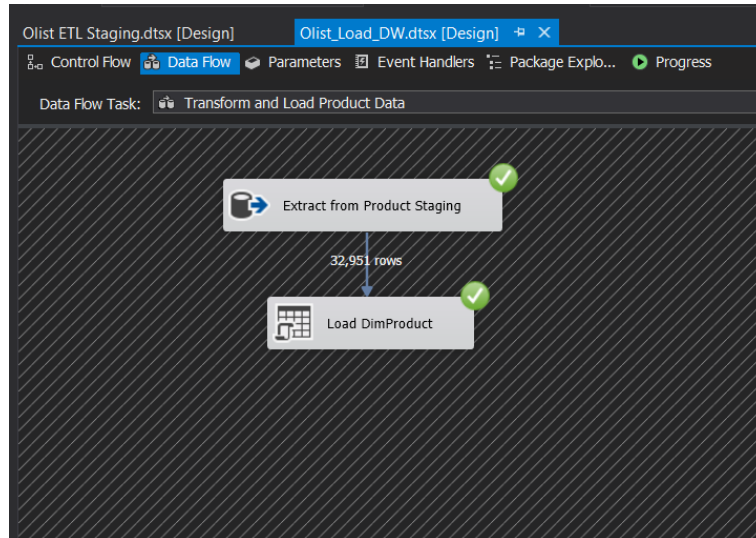
SELECT * FROM [dbo].[DimDate]

```

Step 5 - Load the data from Olist_Staging to the dimensions in Olist_DW using transformations.

(I used transformation task as lookups, derived columns, merge, sort)

- Product transformation



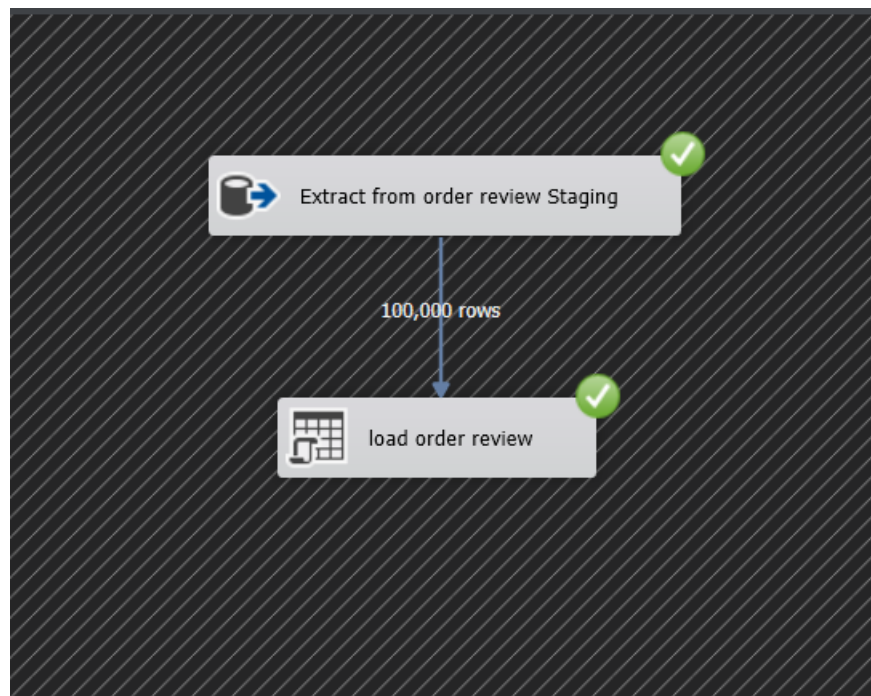
DimProduct Table -

```
***** Script for SelectTopNRows command from SSMS *****
SELECT TOP 1000 [product_SK]
,[AlternateProduct_id]
,[product_category_name]
,[product_name_lenght]
,[product_description_lenght]
,[product_photos_qty]
,[product_weight_g]
,[product_length_cm]
,[product_height_cm]
,[product_width_cm]
FROM [Olist_DW].[dbo].[DimProduct]
```

	product_SK	AlternateProduct_id	product_category_name	product_name_lenght	product_description_lenght	product_photos_qty	product_weight_g	product_length_cm	product_heig
7	7	3394bd1ad2fa9278c9da6724be73c60	alimentos_bebidas	36	352	3	650	16	23
8	8	fd37eb547d2c63b60bcefd68951ee54d	moveis_decoracao	55	238	1	2700	50	12
9	9	415dfa912acf02a7360d3f4cf2f9b06	ferramentas_jardim	58	1063	1	472	18	19
10	10	80665d04b7d8216835a0ad52625c7039	papelaria	59	227	3	900	38	30
11	11	e7467a3d68aba3dce30c17002bccd8da	esporte_lazer	59	772	1	2450	20	11
12	12	#39fa912acf02a733de1b0cf84e779e	pet_shop	51	728	1	5850	46	13
13	13	119addec70b398c88bec023de4b8786	moveis_decoracao	59	1180	2	1950	43	14
14	14	70afaf5a0fe1e5947d15e7f375f1390f	papelaria	37	151	1	200	16	10
15	15	acc8648b140588f8aa6cce2905ff6872	utilidades_domesticas	45	381	1	400	26	12
16	16	9917cf3117ebf28fa2264443c7c6a55	informatica_acessorios	45	461	2	800	16	29
17	17	6525c63f85b031e963d95bb3a07e541	consoles_games	43	675	3	650	26	17
18	18	1b861948326326b19c7442789ef59ebb	beleza_saude	53	144	1	100	16	11
19	19	0540951fb114c6a30ef6a43567983ce9f	telefonia	58	1551	1	100	20	20

Query executed successfully. DESKTOP-DADSR34 (13.0 SP1) | DESKTOP-DADSR34\aveen... | Olist_DW | 00:00:00 | 1000 rows

- Order review data transformation



DimOrderReview Table -

***** Script for SelectTopNRows command from SSMS *****

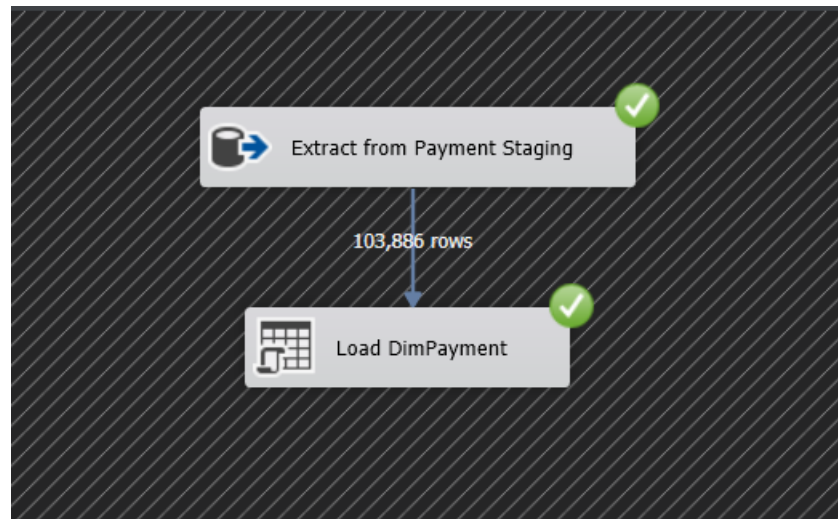
```

SELECT TOP 1000 [review_SK]
      ,[AlternateReview_id]
      ,[order_id]
      ,[review_score]
      ,[review_comment_title]
      ,[review_comment_message]
      ,[review_creation_date]
      ,[review_answer_timestamp]
FROM [Olist_DW].[dbo].[DimOrderReview]
  
```

review_SK	AlternateReview_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_date
1	fa58cab7240294da0ecbb1d389fe64e5	d4b4b8d59b203dc85ccad823014fbb34	4	NULL	NULL	2018-08-03 00:00:00.00
2	f4d8572b570b51c323f9a20383fbedee	eeecbb8240f0eb71bec072e86dc50d8be	1	NULL	comprei trÃs luminÃrias e sÃ³ foi entregue atÃ...	2018-03-24 00:00:00.00
3	18f49afebdbcb6c70c87771c385480f	a0dddfb69e48d83c22789bd03521a53	5	NULL	Cumpriu o prazo certinho e o produto Ã lindo. R...	2017-10-15 01:00:00.00
4	6ae1e5a9b79624789e303773a438f02e	892b6462174cfb942fb8d9c8e0287c11	5	NULL	NULL	2018-05-05 00:00:00.00
5	fee30fc36f39354563cd7d087376b120	b7664cfd894b776f4c7743e5e2fb917b	4	NULL	Boa logÃstica	2017-04-05 00:00:00.00
6	1808e984fe5a3d41d80cf4a57599bf47	cbe90389edee454fc22a84aacee69e6d	4	NULL	NULL	2018-03-09 00:00:00.00
7	294d7dc59926e45e9b0b9496fc4b59	2936486c289152580e86cde8051de57c	5	NULL	NULL	2018-01-09 00:00:00.00
8	38158c51671499b0b33186b6506341e1	2f5b7d28366c2cee7d7d9be0c23136e5	5	Ãtima	Bem embalado e chegou antes do prazo...	2018-05-06 00:00:00.00
9	b3cc50376d5cb6ae0aa935a8179d96fb	555905de95610046be84df6313b3328e	1	NULL	Produto NÃo Entregue	2017-11-09 00:00:00.00
10	aa15b2bf0d4372b9c38bf582113da977	b4eabf9a64099f1413105754e3d1343f	5	NULL	NULL	2017-03-03 00:00:00.00
11	c0c0ecd7efd59bb1e29173784f2abcf0	cdac2dd3643b01d8c79cb8bb30998c6a	1	NULL	Quero saber se foi cancelada sÃ isso!	2017-04-04 00:00:00.00
12	6cc130828b852a08f285b4580c79cb82	4cc7a837ccbd410b6a7ebe3e15dc8800	5	NULL	Todo processo foi rÃpido, pratico e seguro. Obrig...	2017-11-29 00:00:00.00
13	9b531hf717668993fe632bac8755e921	a32c1ae6b544c2e72c53dh079e7a3af6	5	NULL	Linda e era como eu esperava Amei	2017-07-11 00:00:00.00

Query executed successfully. DESKTOP-DADSR34 (13.0 SP1) DESKTOP-DADSR34\naveen... Olist_DW 00:00:00 1000 rows

- Payment data transformation



DimPayment Table

***** Script for SelectTopNRows command from SSMS *****

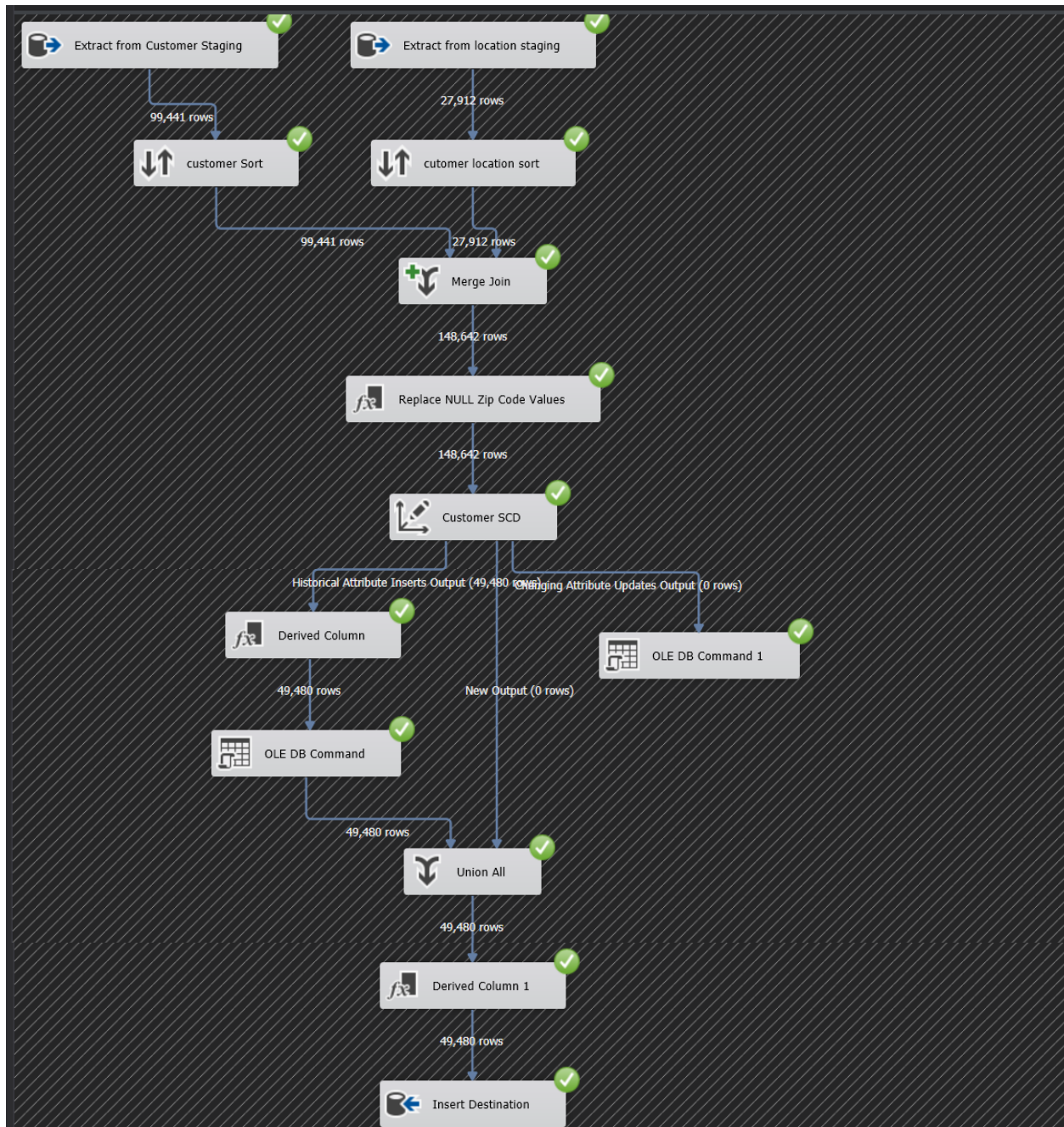
```

SELECT TOP 1000 [payment_SK]
      ,[AlternatePayment_id]
      ,[order_id]
      ,[payment_sequential]
      ,[payment_type]
      ,[payment_installments]
      ,[payment_value]
FROM [Olist_DW].[dbo].[DimPayment]
  
```

	payment_SK	AlternatePayment_id	order_id	payment_sequential	payment_type	payment_installments	payment_value
1	1	20000000	b81ef226f3e1789b1e8b2acac839d17	1	credit_card	8	99.33
2	2	20000001	a9810da82917af2d9aefd1278f1dcfa0	1	credit_card	1	24.39
3	3	20000002	25e8ea4e93396b6fa0d3dd708e76c1bd	1	credit_card	1	65.71
4	4	20000003	ba78997921bbcdc1373bb41e913ab953	1	credit_card	8	107.78
5	5	20000004	42df880ba16b47b59251dd489d4441a	1	credit_card	2	128.45
6	6	20000005	298fcd1f73eb413e4d26d01b25bc1cd	1	credit_card	2	96.12
7	7	20000006	771ee386b001f06208a7419e4fc1bbd7	1	credit_card	1	81.16
8	8	20000007	3d7239c394a212faae122962df514ac7	1	credit_card	3	51.84
9	9	20000008	1f78449c87a54faf9e96e88ba1491fa9	1	credit_card	6	341.09
10	10	20000009	0573b5e23cbd798006520e1d5b4c6714	1	boleto	1	51.95
11	11	20000010	d88e0d5fa41661ce03cf6cf336527646	1	credit_card	8	188.73
12	12	20000011	2480f727e869fdeb397244a21b721b67	1	credit_card	1	141.9
13	13	20000012	616105c9352a9668c38303ad44e056cd	1	credit_card	1	75.78
14	14	20000013	cf95215a722f3e9f29a6b9bah87a29a61	1	credit_card	5	102.66

Query executed successfully. DESKTOP-DADSR34 (13.0 SP1) DESKTOP-DADSR

- Customer data transformation



DimCustomer table

Script for SelectTopNRows command from SSMS

```

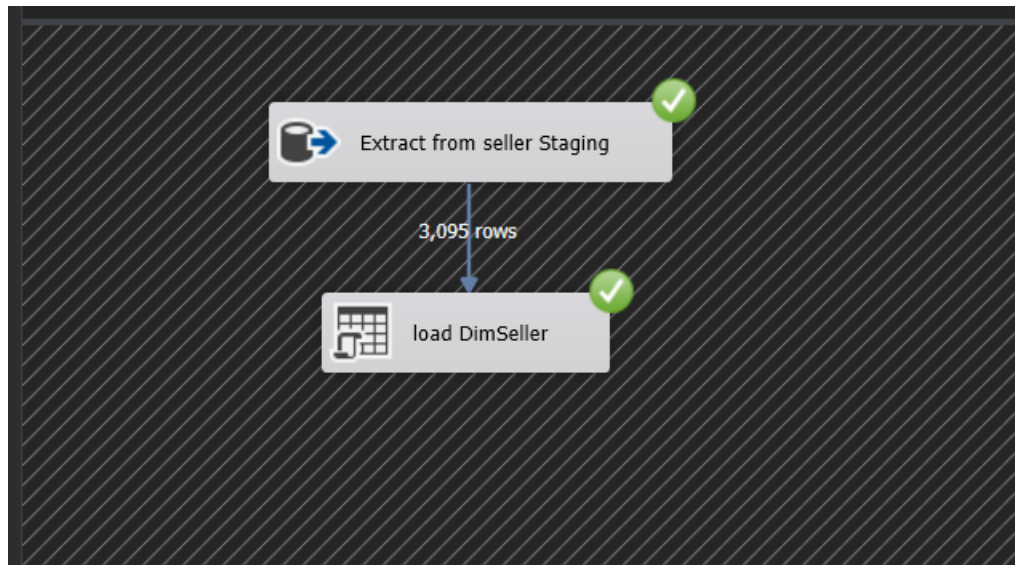
SELECT TOP 1000 [customer_SK]
,[AlternateCustomer_id]
,[customer_unique_id]
,[geolocation_zip_code_prefix]
,[geolocation_state]
,[geolocation_city]
,[insertDate]
,[modifiedDate]
,[startDate]
,[endDate]
FROM [Olist_DW].[dbo].[DimCustomer]

```

customer_SK	AlternateCustomer_id	customer_unique_id	geolocation_zip_code_prefix	geolocation_state	geolocation_city	insertDate
1	7ae2a9337aa4bc799723511faa1d6830	0c1a20644f0dc126c3eaff8dbc1bd12c	1003	SP	sao paulo	2021-05-13 19:42
2	7ae2a9337aa4bc799723511faa1d6830	0c1a20644f0dc126c3eaff8dbc1bd12c	1003	SP	são paulo	2021-05-13 19:42
3	a09edf8c1e842e94805a206b3d73eed5	968f6d2f674977d88a4b445a5117ccd8	1004	SP	são paulo	2021-05-13 19:42
4	a09edf8c1e842e94805a206b3d73eed5	968f6d2f674977d88a4b445a5117ccd8	1004	SP	sao paulo	2021-05-13 19:42
5	ee9b73e88afb4904ee2322cfc89cf638	095e7c124c5c1ccb1eb9f731152eae6a	1004	SP	são paulo	2021-05-13 19:42
6	ee9b73e88afb4904ee2322cfc89cf638	095e7c124c5c1ccb1eb9f731152eae6a	1004	SP	sao paulo	2021-05-13 19:42
7	5a8b64ee6ccdae09ea823e6aa00e9517	9c84e5193d6ee59b3870e0e4e3a2dad8	1005	SP	sao paulo	2021-05-13 19:42
8	5a8b64ee6ccdae09ea823e6aa00e9517	9c84e5193d6ee59b3870e0e4e3a2dad8	1005	SP	são paulo	2021-05-13 19:42
9	6ec2b4682814cfdac8d92bad42b3ddab	57f0ea1c7f6b9ef8615c0a0b8f06fe57	1005	SP	sao paulo	2021-05-13 19:42
10	6ec2b4682814cfdac8d92bad42b3ddab	57f0ea1c7f6b9ef8615c0a0b8f06fe57	1005	SP	são paulo	2021-05-13 19:42
11	f3e31afdae80581be48ce94e7b0f3366	ded4351942c7fc292b88e5b090af2b46	1005	SP	sao paulo	2021-05-13 19:42
12	f3e31afdae80581be48ce94e7b0f3366	ded4351942c7fc292b88e5b090af2b46	1005	SP	são paulo	2021-05-13 19:42
13	c46061cd515f35604f9e9a2130df702	84a7776f914ff19505e44effba86455f	1005	SP	sao paulo	2021-05-13 19:42
14	c46061cd515f35604f9e9a2130df702	84a7776f914ff19505e44effba86455f	1005	SP	são paulo	2021-05-13 19:42
15	15d7dbcd027b5b24866db33e2b819021	ddab5650ba76e2fa2d7e25ed3343bd92	1005	SP	sao paulo	2021-05-13 19:42
16	15d7dbcd027b5b24866db33e2b819021	ddab5650ba76e2fa2d7e25ed3343bd92	1005	SP	são paulo	2021-05-13 19:42
17	6e4eb34e74d526a82726712aa17c02b	ded4351942c7fc292b88e5b090af2b46	1005	SP	sao paulo	2021-05-13 19:42

Query executed successfully. DESKTOP-DADSR34 (13.0 SP1) DESKTOP-DADSR34\aveen... Olist_DW 00:00:00 1000 rows

- Seller data transformation

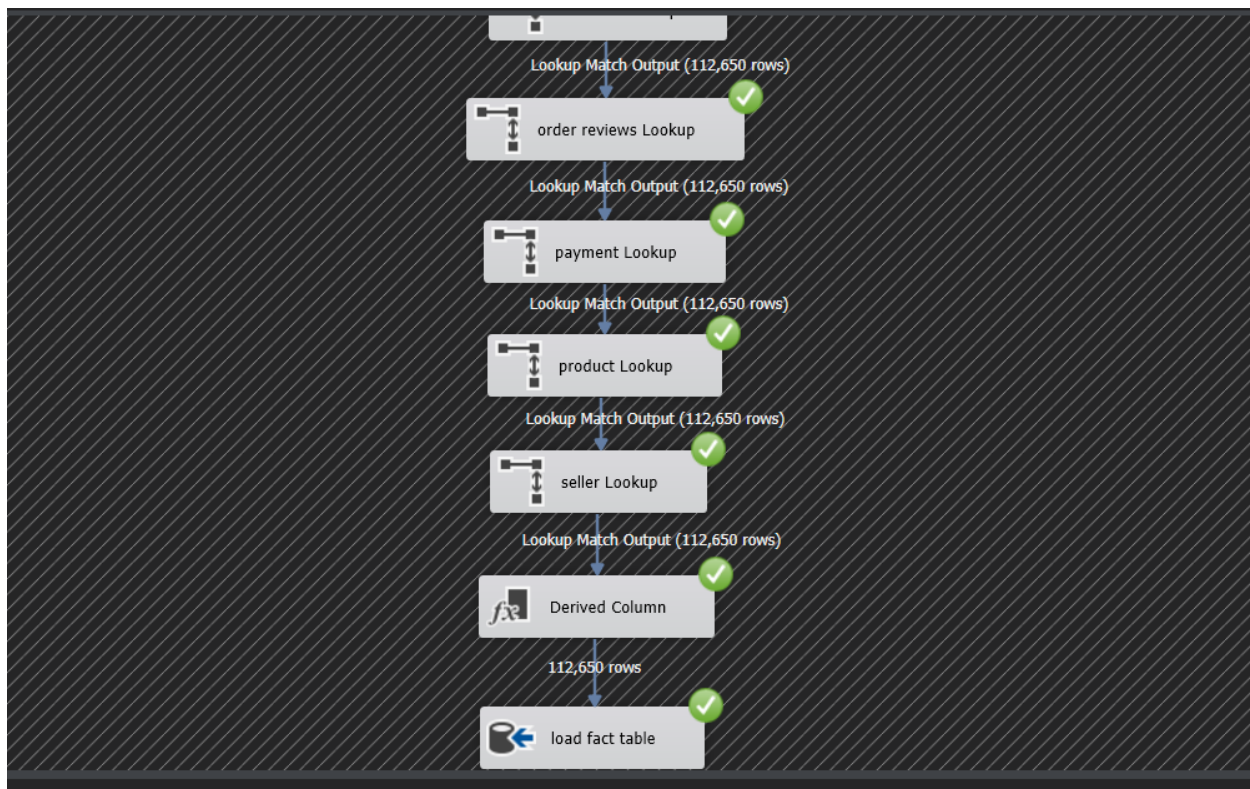


DimSeller table

Results			
	seller_SK	AlternateSeller_id	sellerName
1	1	8602a61d680a10a82cceeda0d99ea3d	Mary
2	2	e0487761face83d64cada2408959a36	Annie
3	3	dd55f1bb788714a40e7954c3be6df745	Mattie
4	4	09bad886111255c5b5030314fc71a4a	Ruby
5	5	f049a72cf58fd31b11f8919cade515e7	Willie
6	6	82921991ff5b557b045605b8bbf08d49	Louise
7	7	c1dde11f12d05c478f5de2d7319ad3b2	Lillie
8	8	14a08204d03bb6b6bde8029f801ae0eb	Ethel
9	9	c84592044b180dec206770c38603814b	Bessie
10	10	6902157ee16728322d61a1321fa79e58	Rosa
11	11	6da1992f915d77be95d7fa48b36904af	Ruth
12	12	5d378b73ab7dd6f0418d743e5dcb0bd1	Elizabeth
13	13	630008e0f062605a415d694489f6f82e	Emma
14	14	7040-026000-04114-4341-705-4214017	Mary

- FactProductSale transformation





FactProductSale Table -

***** Script for SelectTopNRows command from SSMS *****

```

SELECT TOP 1000 [order_id]
,[order_item_id]
,[customer_SK]
,[product_SK]
,[seller_SK]
,[payment_SK]
,[review_SK]
,[shipping_limit_date]
,[price]
,[freight_value]
,[order_status]
,[order_purchase_timestamp]
,[order_approved_at]
,[order_delivered_carrier_date]
,[order_delivered_customer_date]
,[order_estimated_delivery_date]
,[netTotal]
,[insertDate]
,[modifiedDate]

```

	order_id	order_item_id	customer_SK	product_SK	seller_SK	payment_SK	review_SK	shipping_limit_date	price	freight_value
1	8cfd5f5c3bd7bfdd1788b72b5bb9341e	1	45433	5061	1939	77093	48400	2017-09-21 14:10:00.000	359.90	18.28
2	8cfd096b874503d795af7dbb5ac758fc	1	58969	9399	1502	89334	13352	2018-07-24 19:30:00.000	25.99	12.84
3	8cfd0b9c9223d214d619f483f51b8979	1	89935	30025	324	95622	41304	2017-06-18 23:15:00.000	38.99	14.10
4	8d00078c87467e65796aa4032f0a0011	1	32753	9922	2561	51569	23459	2017-11-03 10:48:00.000	198.00	15.14
5	8d009dd4bfa1f56a289579b630fa3b83	1	68056	30831	782	24514	73928	2018-03-13 19:15:00.000	49.77	12.79
6	8d00e06d3fa925f408fd289f9805a765	1	89401	28247	2260	38454	67379	2018-02-26 17:10:00.000	109.95	13.79
7	8d0104be66d2f2828b6dd123117d7c22	1	86259	10240	782	9579	38527	2017-12-08 17:32:00.000	259.77	15.57
8	8d02084669f5b4bd784dba858f55bcd6	1	106995	32216	710	67422	36240	2018-04-16 22:29:00.000	35.00	22.06
9	8d032d2c84787aedd32fc2d5014954	1	43241	24402	1519	50688	19638	2017-10-20 11:56:00.000	13.90	11.85
10	8d041d799681ab5221f1fd956d1d59f7	1	133575	16413	2437	50481	71904	2018-07-27 15:05:00.000	119.00	35.90
11	8d04cc6c75bba21874268deda821055d	1	103810	16148	2533	30415	87235	2017-07-24 06:10:00.000	193.00	23.67
12	8d053738b817a245da596c4aac057923	1	98309	8712	131	38689	33569	2017-02-15 12:25:00.000	49.90	14.52
13	8d0584fef2164e00c8eb23abhd187ef7	1	130968	32332	2416	73894	64906	2017-08-17 20:15:00.000	69.90	8.11

Query executed successfully.

DESKTOP-DADSR34 (13.0 SP1) | DESKTOP-DADSR34\ naveen... | Olist_DW | 00:00:00 | 1000