Cairo University
Faculty of Computers and Artificial Intelligence

Third Year

Data Warehouse Project

Name	**ID**
Mariam Essam Mohamed Abdelaziz	20220511
Momen Mohamed Salem	20221258
Mai Hassan Awad	20221172
Ahmed Hossam Samir El Alfy	20220016
Habiba Alaa Eldin Mahfouz	20220104

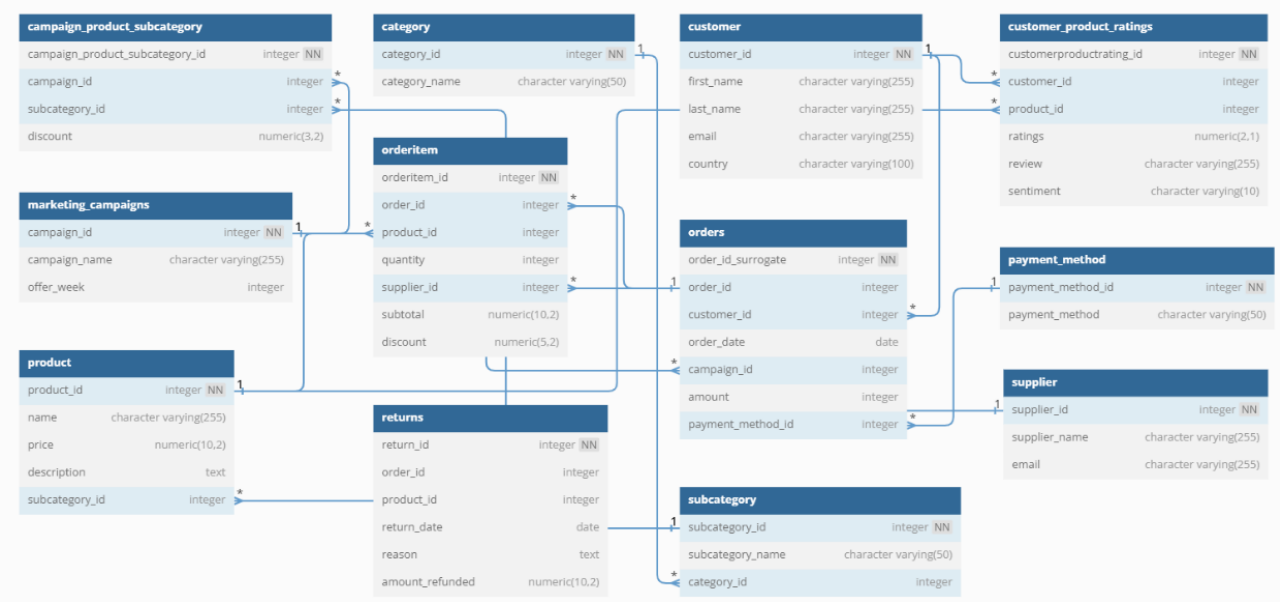
Sec: 3IS-S1

Cairo University
Faculty of Computers and Artificial Intelligence

E-Commerce Data Warehouse

I. Source System:

- **Source Data:**
  OLTP-Ecommerce-Data
- **Physical Model:**



II. Dimensional Model:

➢ Business Processes Being Modeled

1. **Sales Transactions** – Capturing product sales by customers over time, including quantities, discounts, and net-revenue.
2. **Campaign Performance** – Tracking customer engagement or purchases related to marketing campaigns.
3. **Product Returns** – Capturing information about returned products, reasons, refunded amounts, and quantities.

➢ Grain of Each Fact Table

1. **Sales Fact Table**
   Grain: very fine: One record per product per customer per order line item on a specific date.

Each row represents: A line item in a customer order including product, quantity, and net revenue.

2. **Campaign Performance Fact Table**
   Grain: One record per customer, per campaign, per subcategory, per date.
   Each row represents: Aggregated campaign performance metrics like total sales and returns for that customer and campaign context.

3. **Returns Fact Table**
   Grain: One record per product return transaction.
   Each row represents: A returned item including refund amount, reason, return date, and original order information.

➢ Type of Each Fact Table

1. **Sales Fact Table**
   - **Type:** Transaction fact table.
   - **Reason:** Records individual sales transactions at the most granular level.
2. **Campaign Performance Fact Table**
   - **Type:** Periodic snapshot fact table.
   - **Reason:** Aggregates data (orders, sales, discounts, returns) across a period, likely daily, by campaign and customer.
3. **Returns Fact Table**
   - **Type:** Transaction fact table.
   - **Reason:** Each record corresponds to a specific return event with all transactional details.

➢ Dimensions:

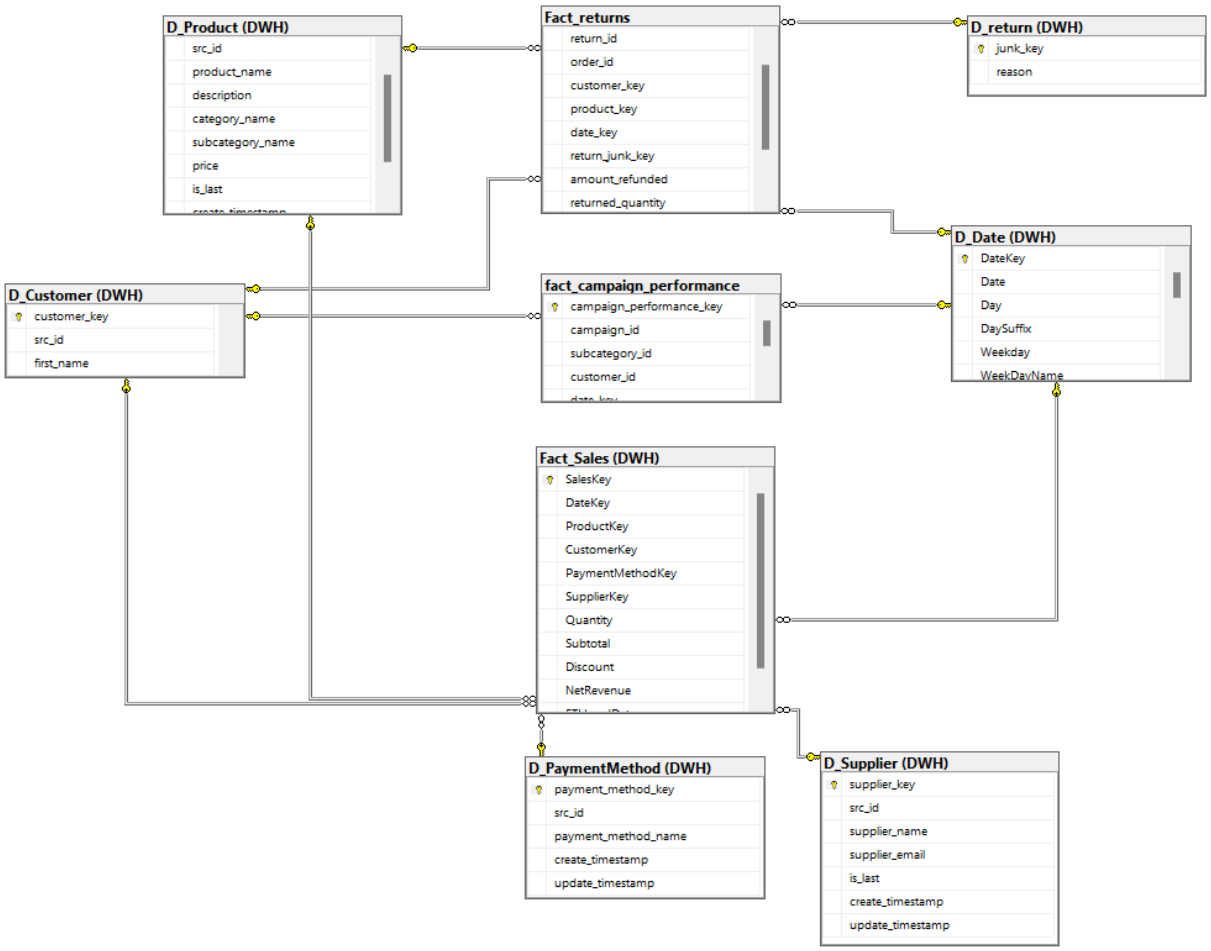
Dimension Name	Dimension Type
D_Date	Role-Playing Dimension
D_Customer	Slowly Changing Dimension (SCD) + Conformed Dimension
D_PaymentMethod	Static Dimension
D_Supplier	Slowly Changing Dimension (SCD)
D_Product	Slowly Changing Dimension (SCD) + Conformed Dimension
D_return	Junk Dimension

Dimension Name	Dimension Type
Order_id	Degenerate Dimension

- **Measures:**

Fact Table	Measures	Type
Sales Fact	Net Revenue	Fully Additive
Returns Fact	Quantity Returned , Processing Days	Fully Additive
Campaign Performance	Total quantity ,total sales, net sales	Fully Additive

- **Physical Model:**



III. Control flow & Data flow:

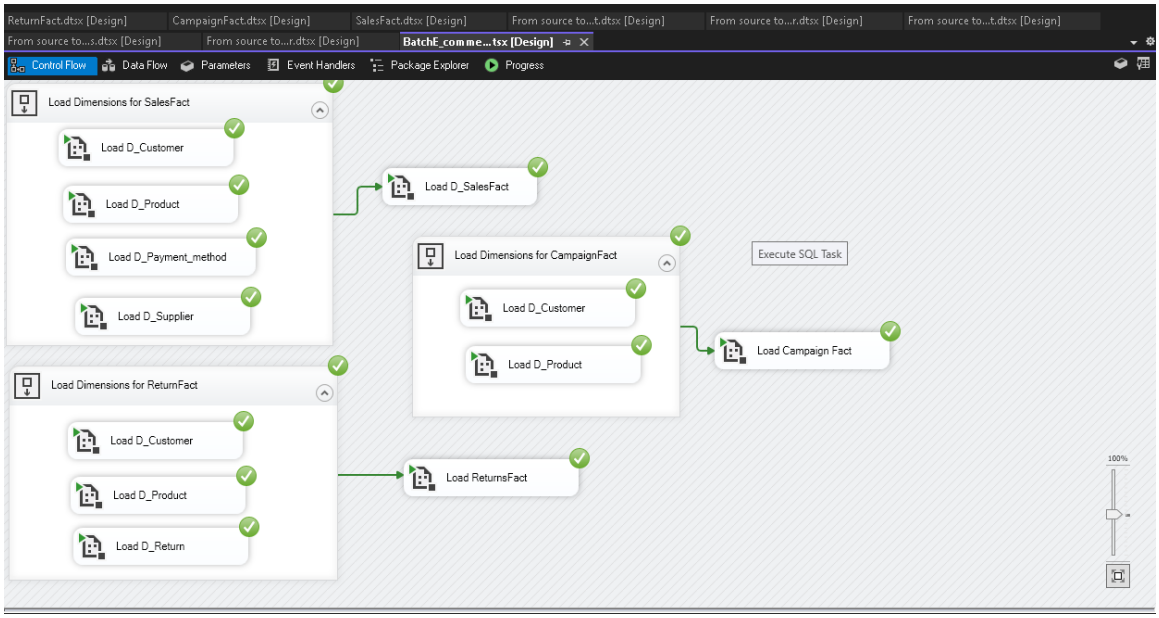


Figure 1 Batch of E-commerce

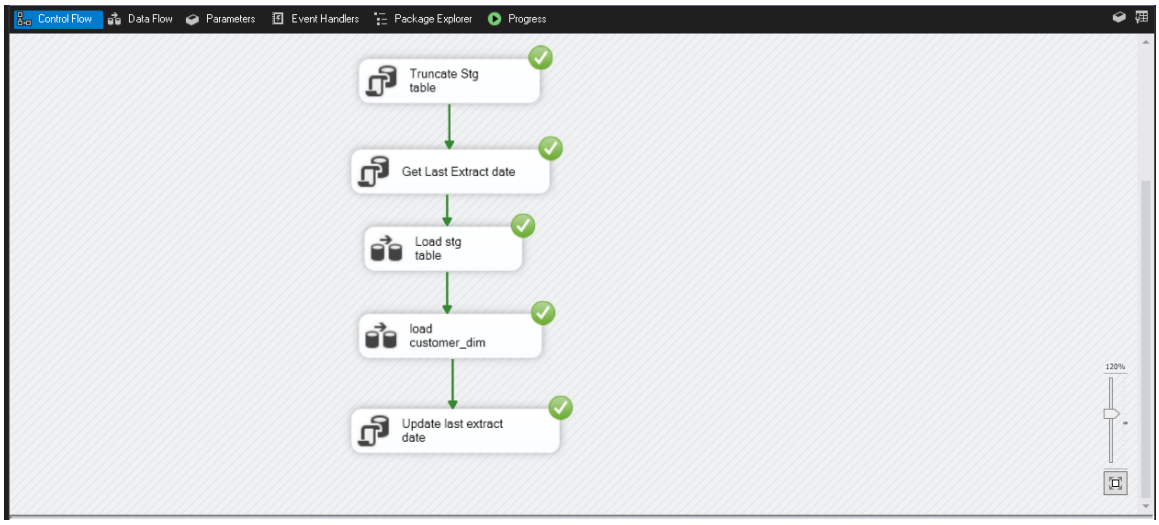


Figure 2 Control Flow of From Source to Customer Dimension

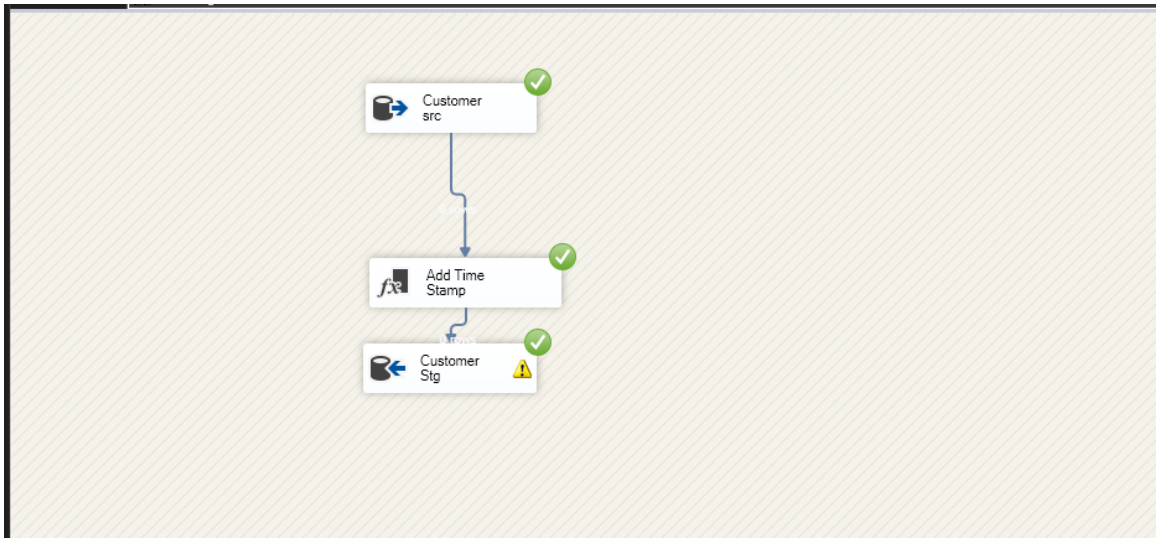


Figure 3 Data Flow of From Source to Customer Dimension (Load Customer Staging Table)

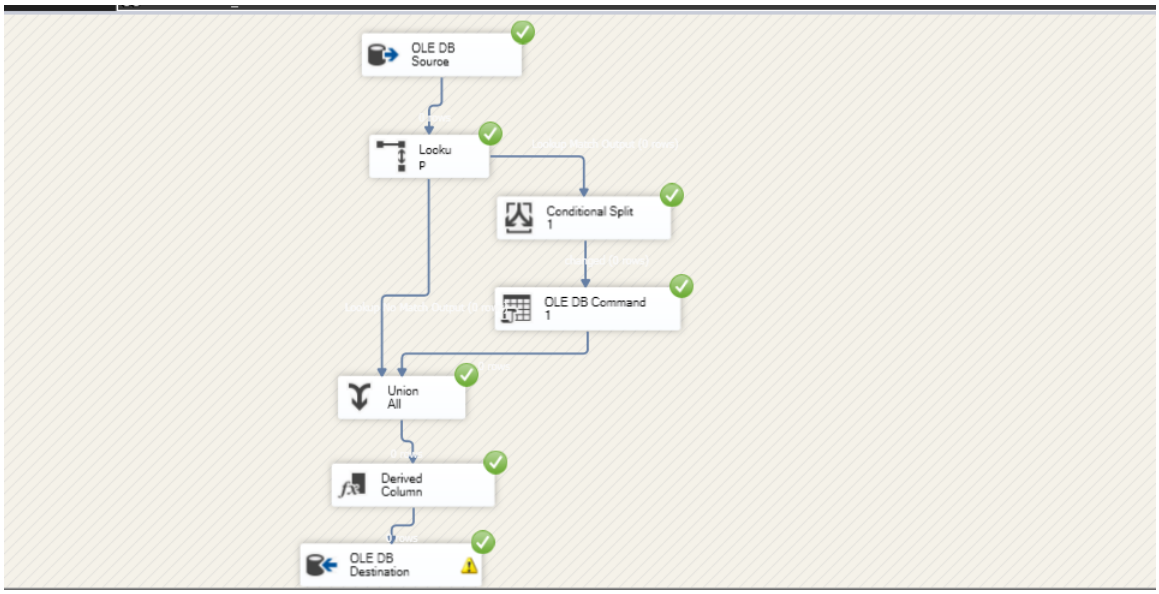


Figure 4 Data Flow of From Source to Customer Dimension (Load Customer Dimension)

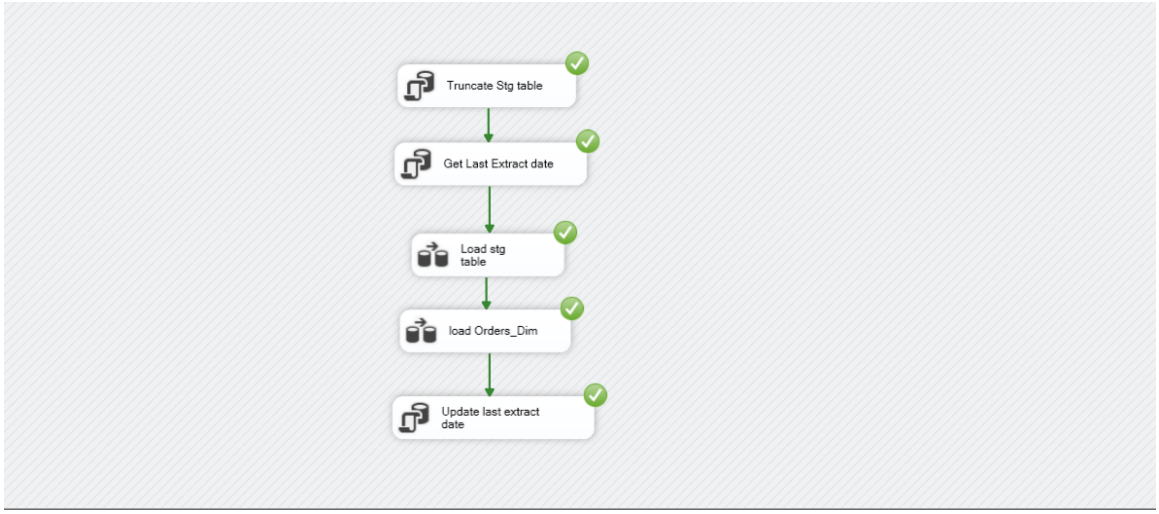


Figure 5 Control Flow of From Source to Orders Dimension

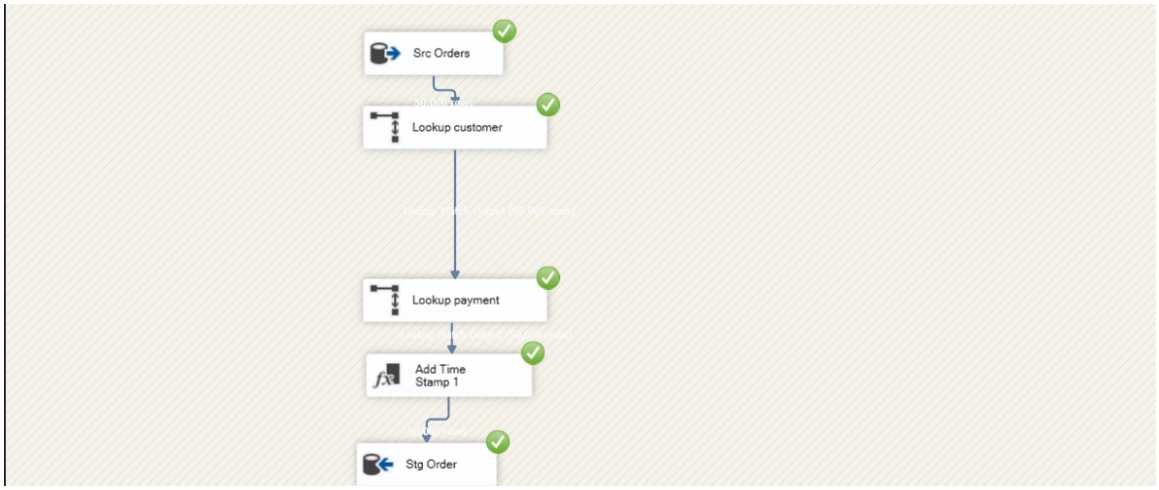


Figure 6 Data Flow of From Source to Orders Dimension (Load Orders Staging Table)

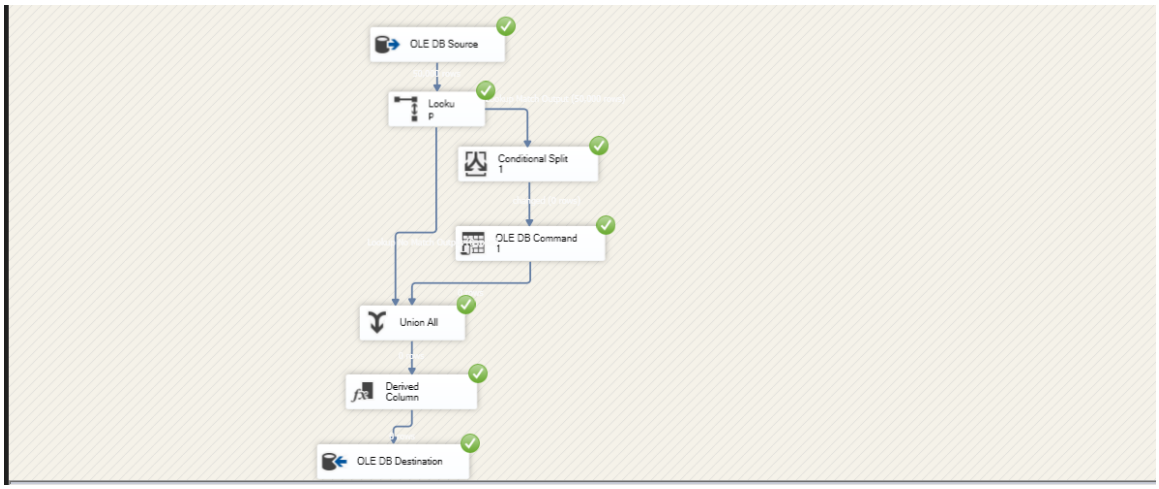


Figure 7 Data Flow of From Source to Orders Dimension (Load Orders Dimension)

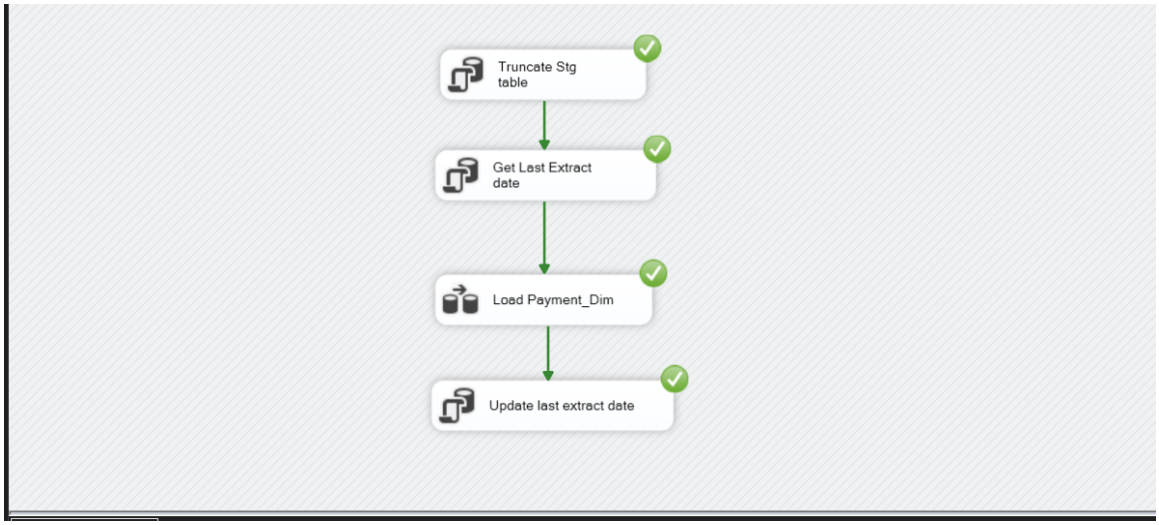


Figure 8 Control Flow of From Source to Payment Method Dimension

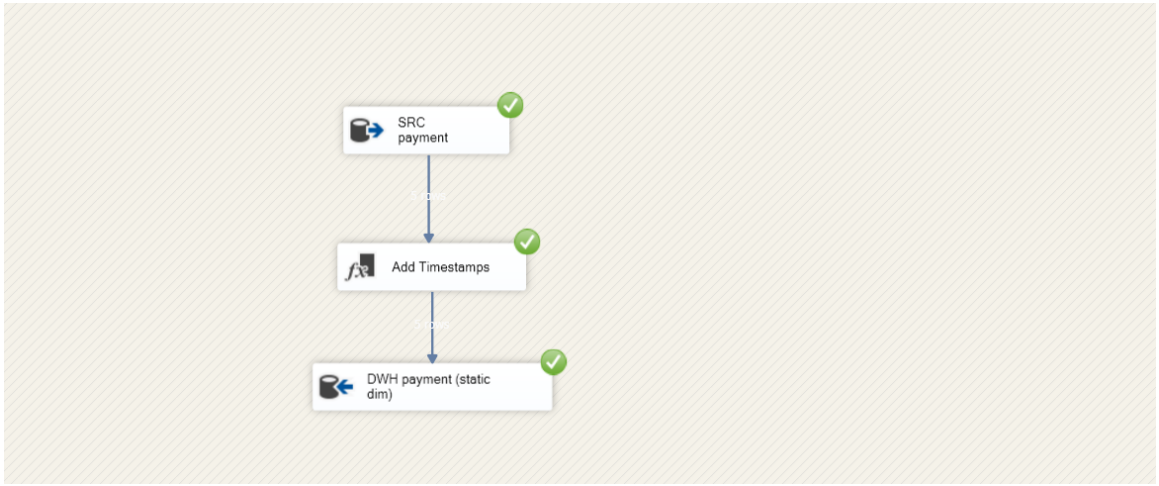


Figure 9 Data Flow of From Source to Payment Method Dimension (Load Payment Method Dimension)

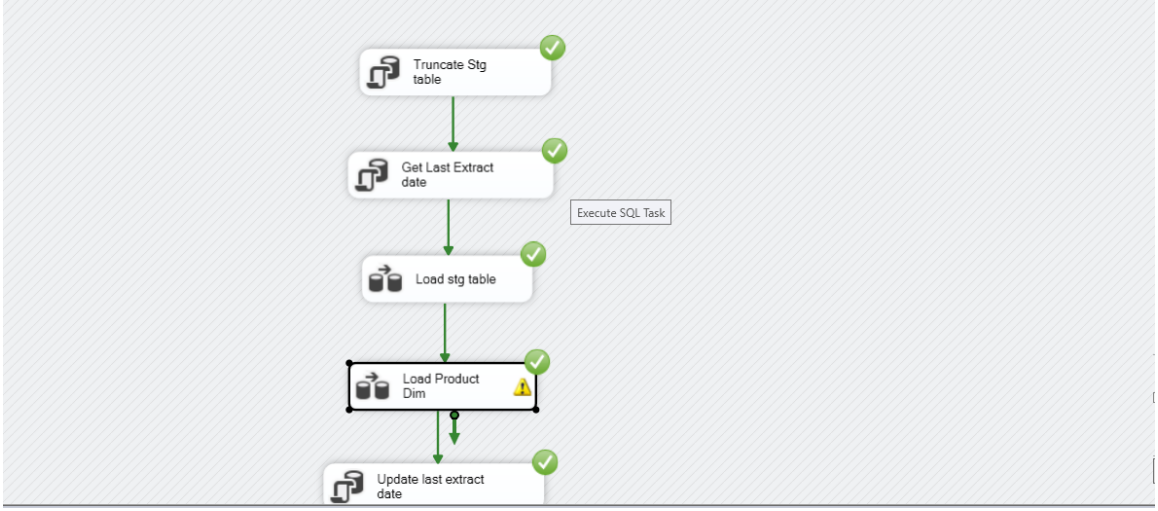


Figure 10 Control Flow of From Source to Product Dimension

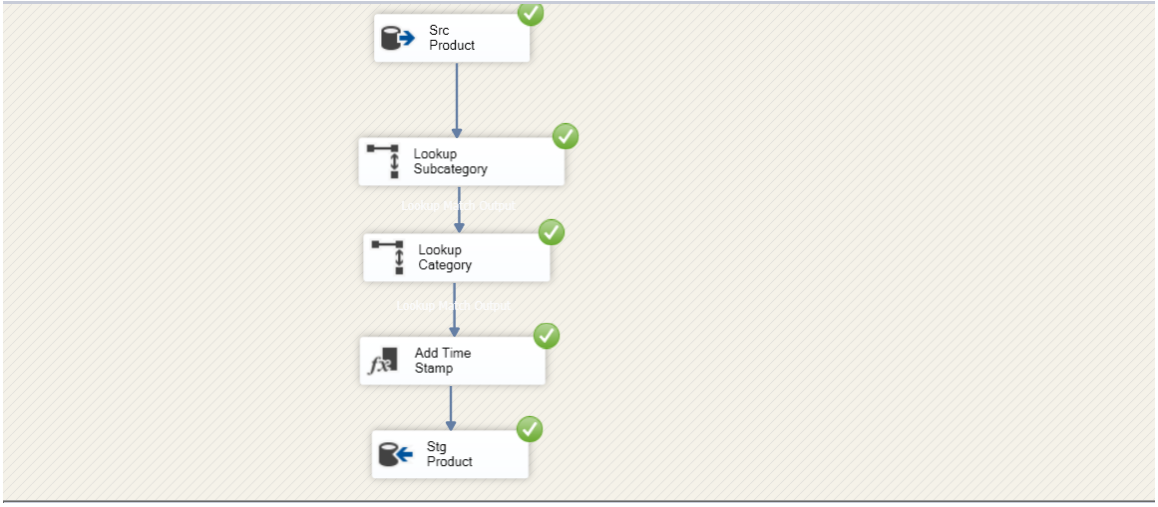


Figure 11 Data Flow of From Source to Product Dimension (Load Product Staging Table)

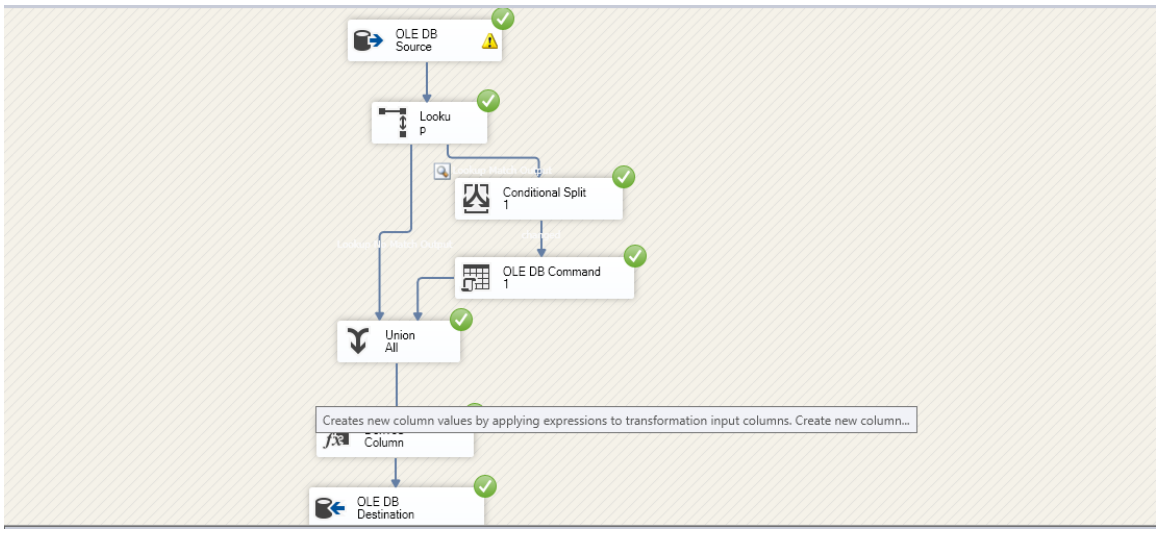


Figure 12 Data Flow of From Source to Product Dimension (Load Product Dimension)

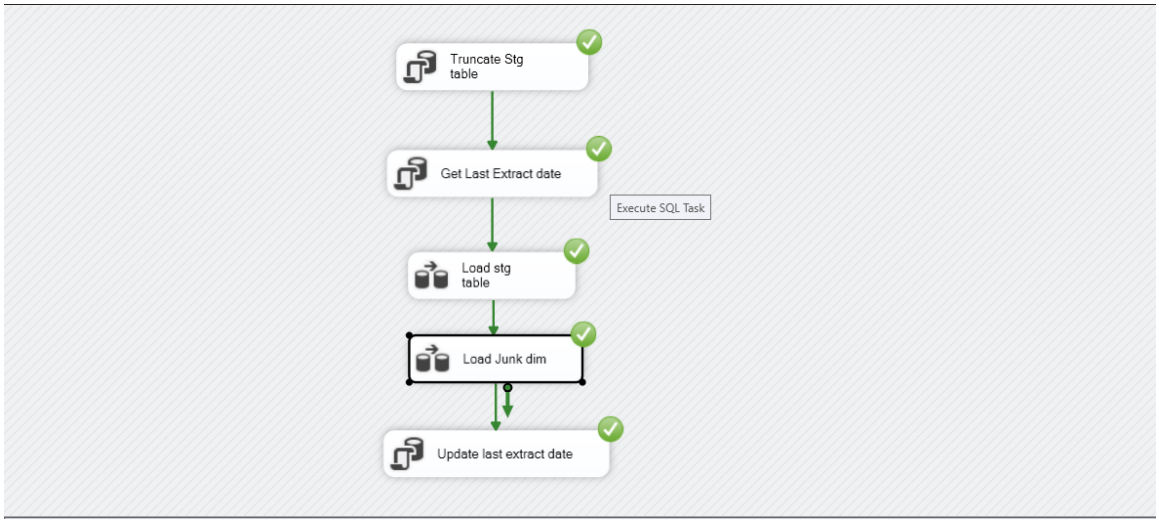


Figure 13 Control Flow of From Source to Returns Dimension

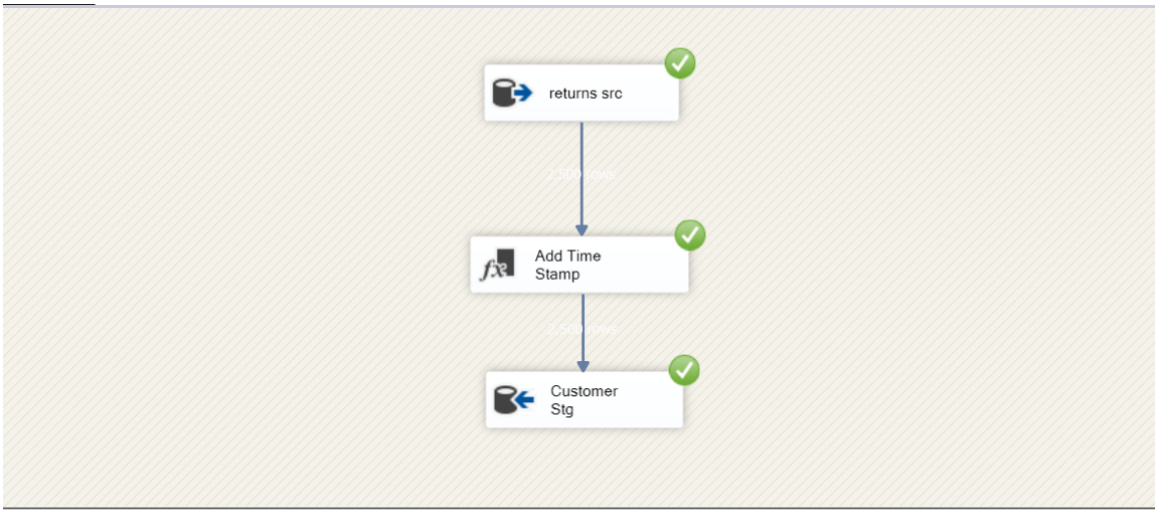


Figure 14 Data Flow of From Source to Returns Dimension (Load Return Staging Table)

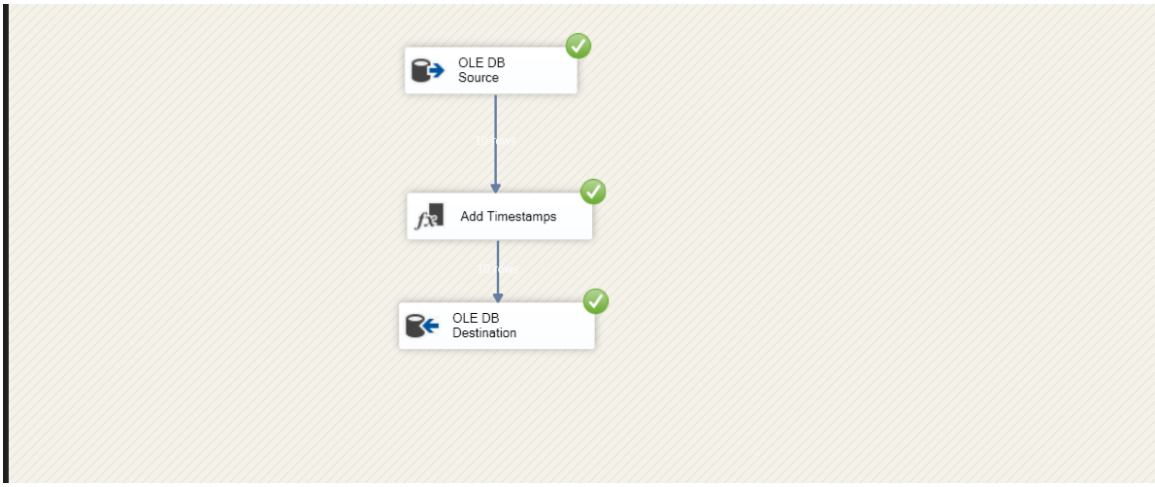


Figure 15 Data Flow of From Source to Returns Dimension (Load Returns Dimension)

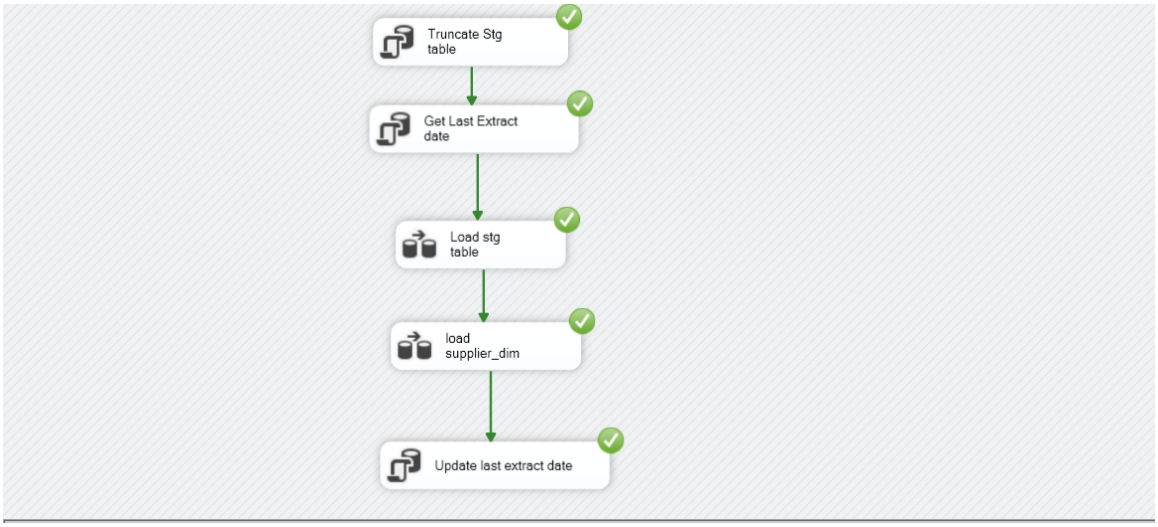


Figure 16 Control Flow of From Source to Supplier Dimension

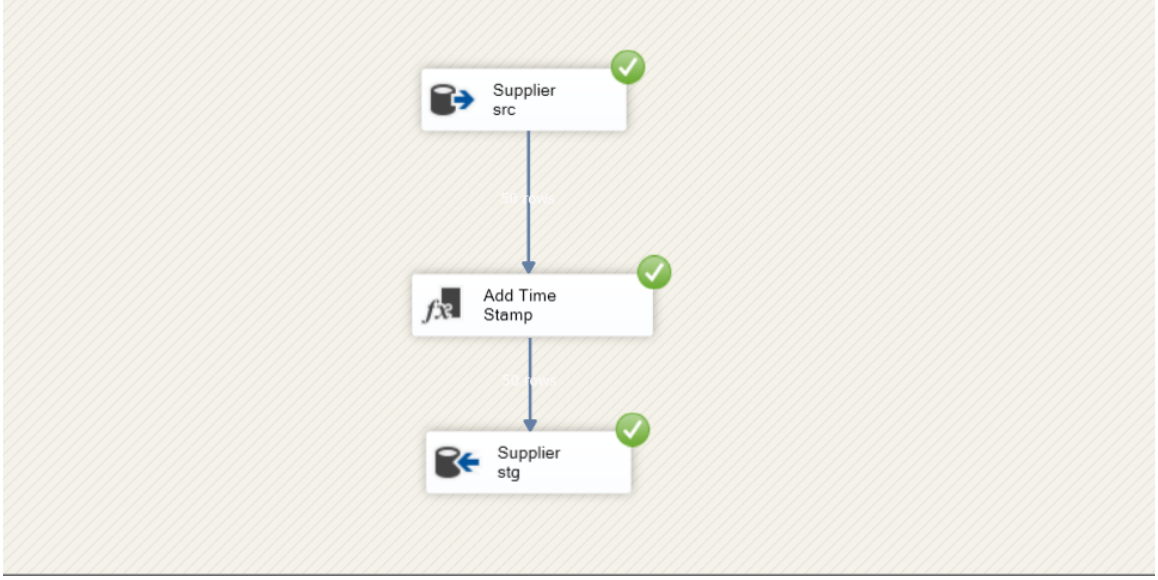


Figure 17 Data Flow of From Source to Supplier Dimension (Load Supplier Staging Table)

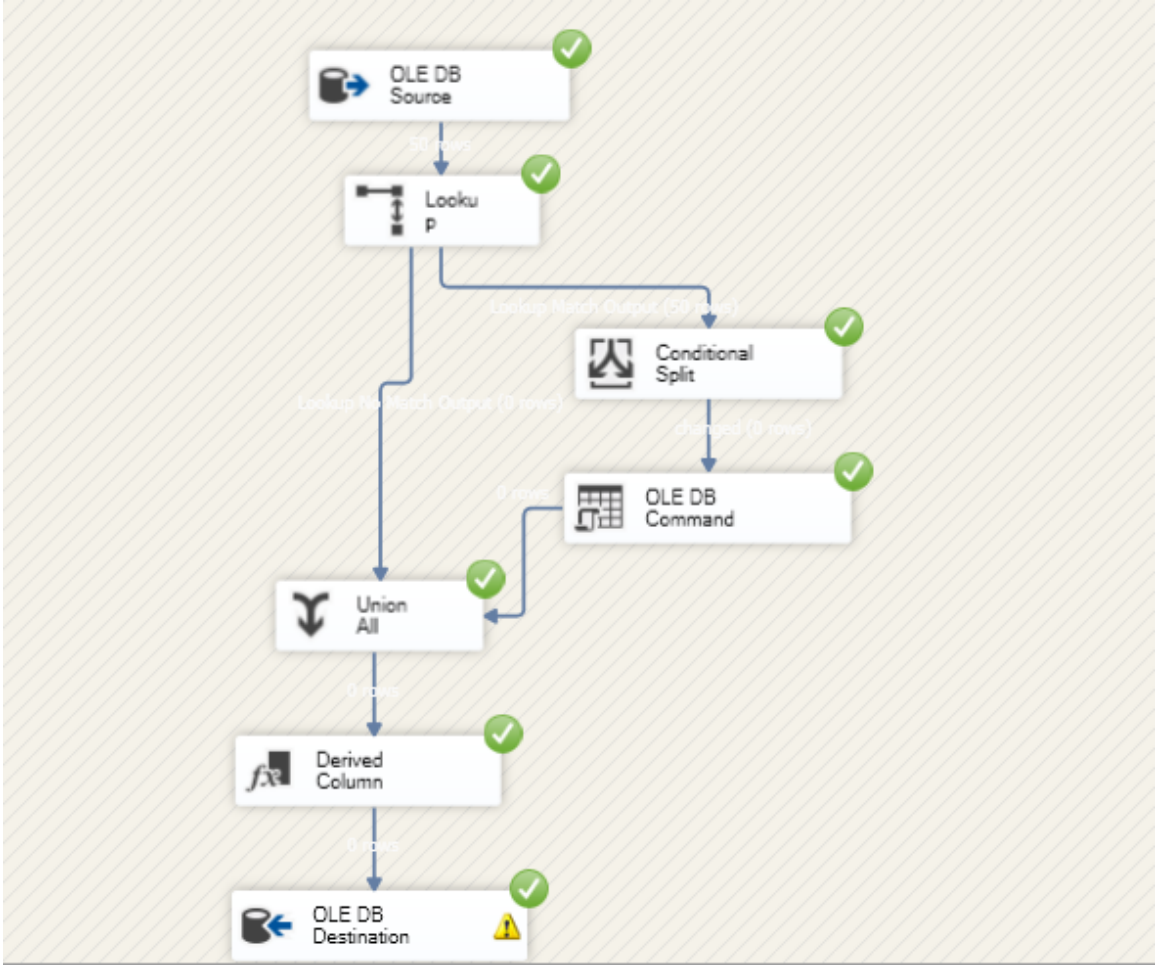


Figure 18 Data Flow of From Source to Supplier Dimension (Load Supplier Dimension)

Figure 19 Control Flow of Returns Fact Table

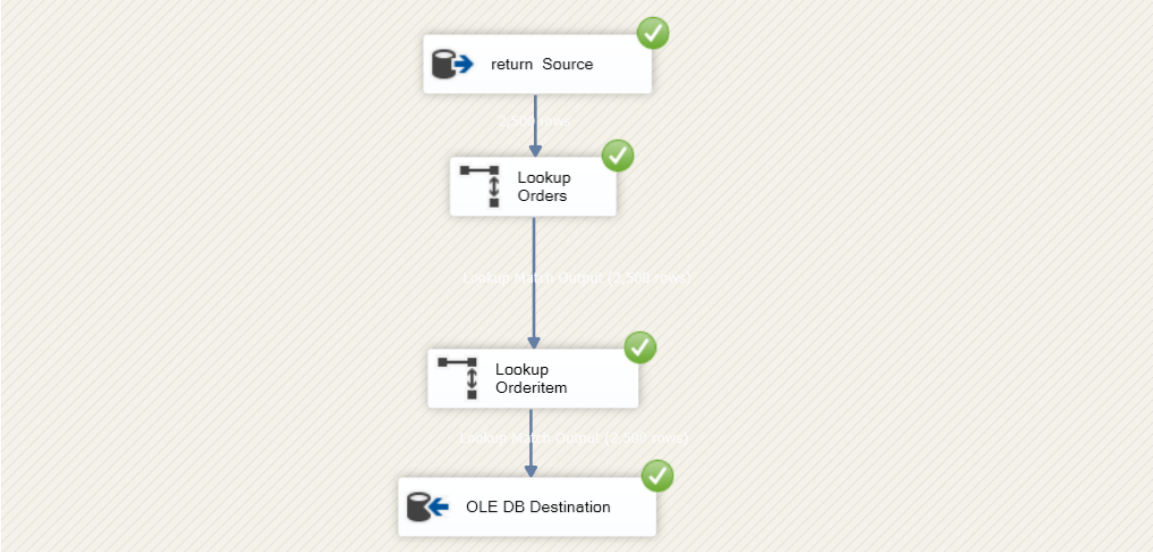


Figure 20 Data Flow of Returns Fact Table (Load Returns Fact Staging Table)

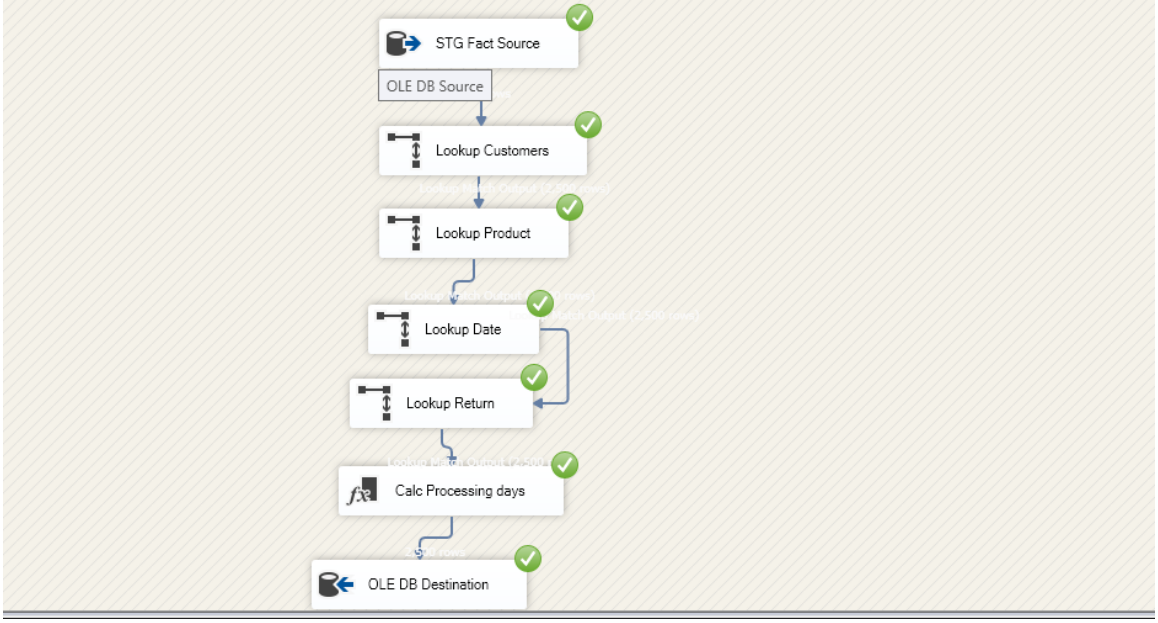


Figure 21 Data Flow of Returns Fact Table (Load Returns Fact Table)

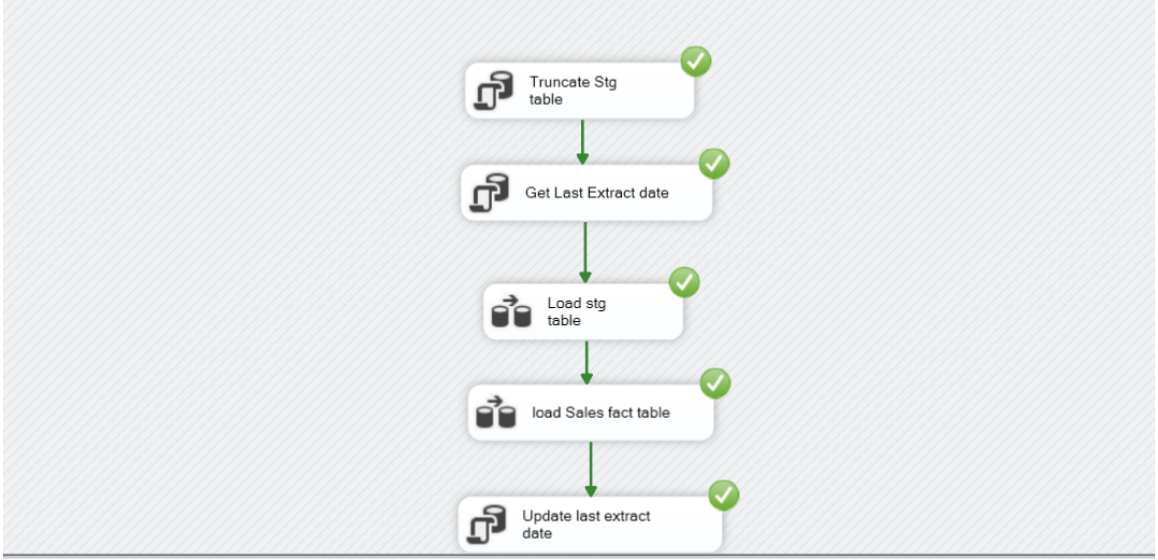


Figure 22 Control Flow of Sales Fact Table

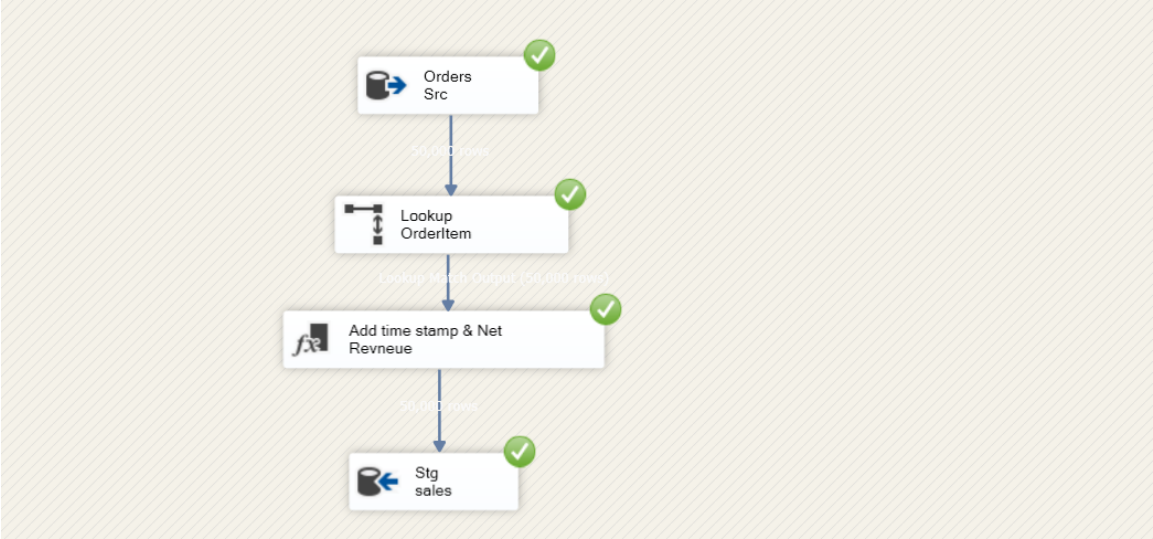


Figure 23 Data Flow of Sales Fact Table (Load Sales Fact Staging Table)

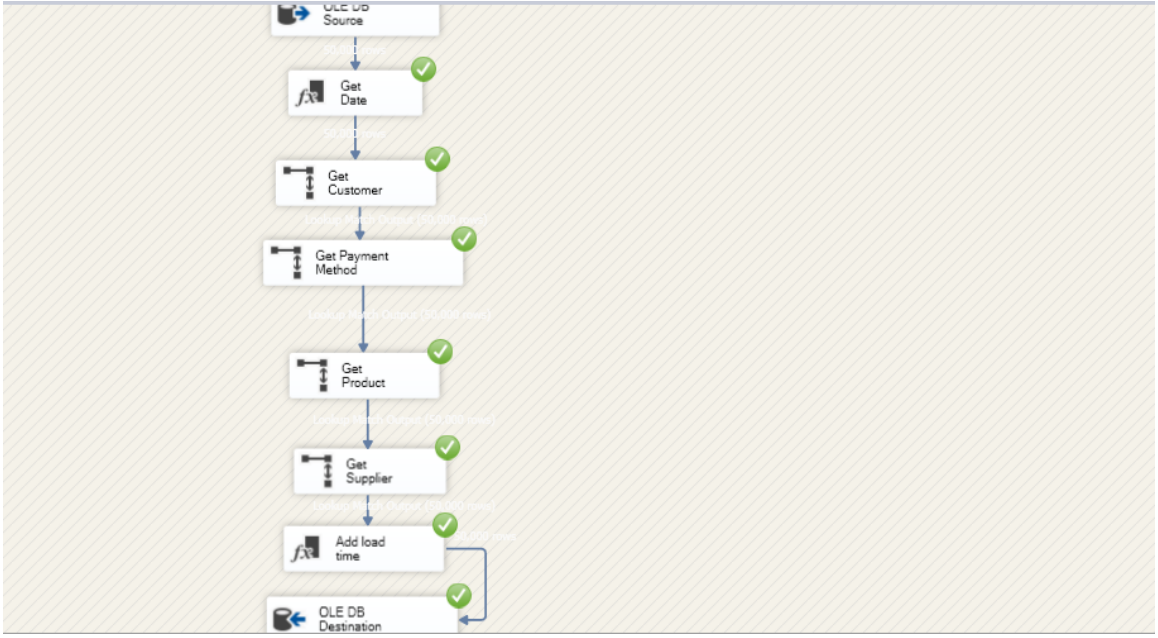


Figure 24 Data Flow of Sales Fact Table (Load Sales Fact Table)

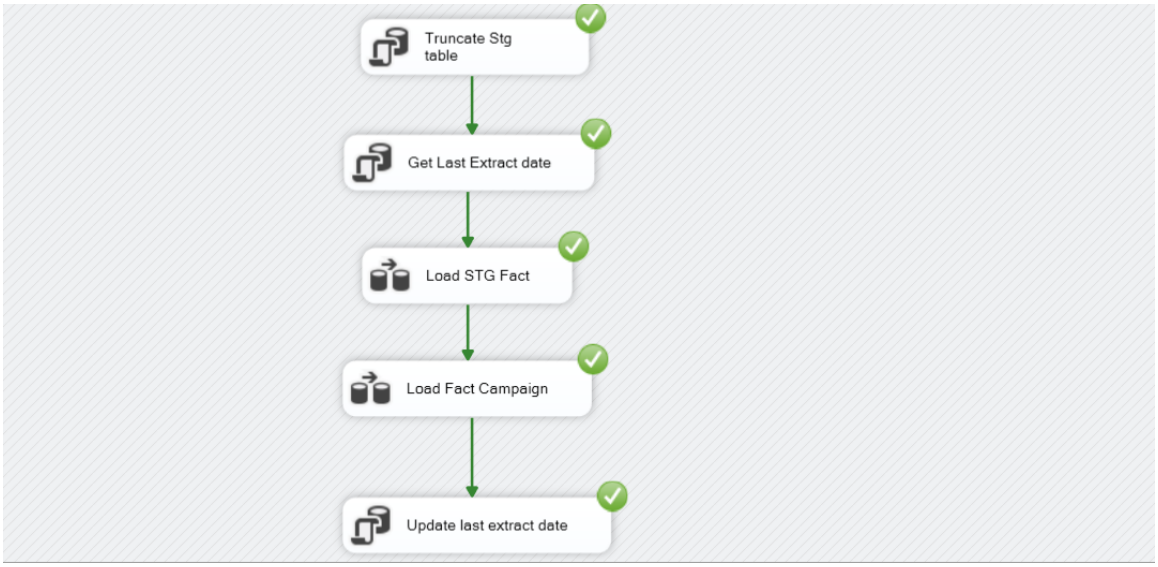


Figure 25 Control Flow of Campaign Performance Fact Table

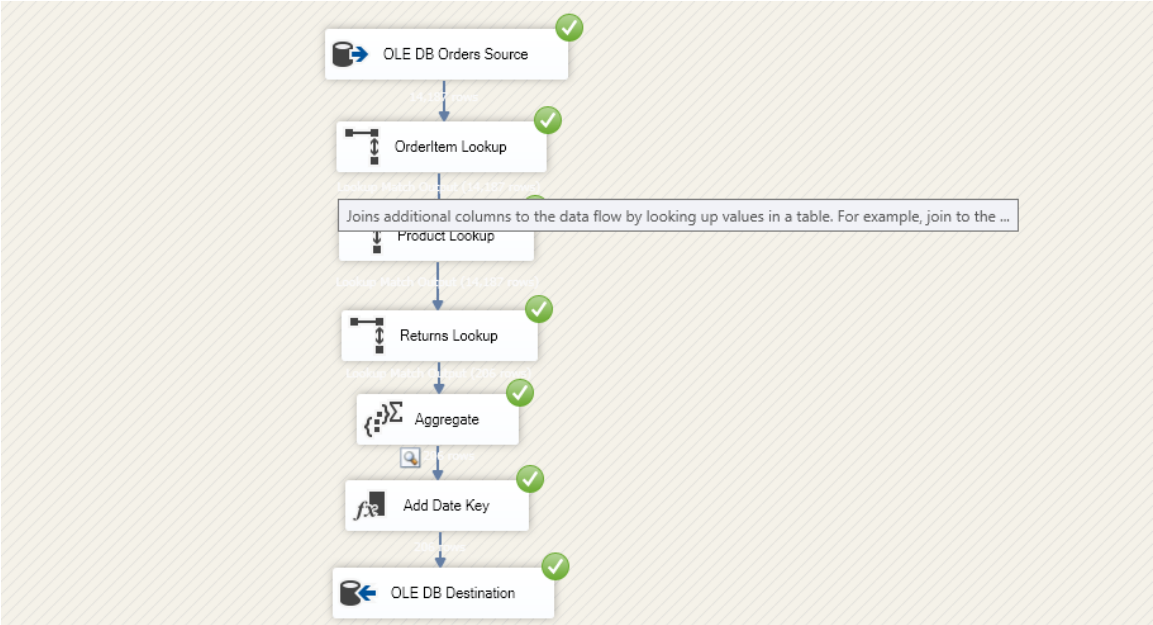


Figure 26 Data Flow of Campaign Performance Fact Table (Load Campaign Performance Fact Staging Table)

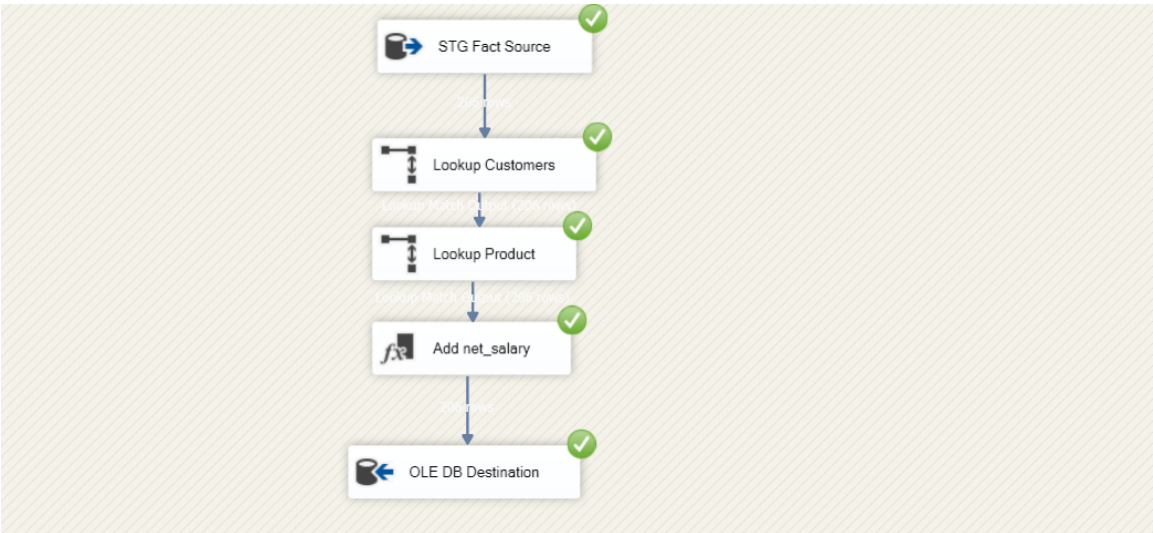


Figure 27 Data Flow of Campaign Performance Fact Table (Load Campaign Performance Fact Table)

IV. Queries:

1. Queries on Sales Fact:

```
--Get Most Used Method to payment with the total amount paid
SELECT TOP 1
    PM.payment_method_name,
    COUNT(*) AS Usage_Count,
    SUM(FS.NetRevenue) AS Total_Amount
FROM
    DWH.Fact_Sales AS FS
JOIN
    DWH.D_PaymentMethod AS PM ON FS.PaymentMethodKey = PM.payment_method_key
GROUP BY
    PM.payment_method_name
ORDER BY
    Usage_Count DESC;
```

100 %

Results | Messages

	payment_method_name	Usage_Count	Total_Amount
1	debit card	20138	40276.00

```sql
--Get Top 10 Products in revenue
SELECT TOP 10
    P.product_key,
    P.product_name,
    SUM(FS.NetRevenue) AS Total_Sales
FROM
    DWH.Fact_Sales AS FS
JOIN
    DWH.D_Product AS P ON FS.ProductKey = P.product_key
GROUP BY
    P.product_key,
    P.product_name
ORDER BY
    Total_Sales DESC


--Get Most Used Method to payment with the total amount paid
```

100 %

Results | Messages

	product_key	product_name	Total_Sales
1	238	Home Security - Product 3	794.00
2	161	Makeup - Product 1	784.00
3	170	Fragrances - Product 5	784.00
4	28	T-Shirts - Product 3	780.00
5	233	Hand Tools - Product 3	760.00
6	26	T-Shirts - Product 1	740.00
7	423	Watches - Product 3	738.00
8	332	Keyboards - Product 2	738.00
9	393	Sports - Product 3	736.00
10	81	Non-Fiction - Product 1	736.00

```sql
--Identify most valuable Customers
SELECT C.customer_email, SUM(FS.NetRevenue) AS Total_Spent
FROM DWH.Fact_Sales FS
JOIN DWH.D_Customer C ON FS.CustomerKey = C.customer_key
GROUP BY C.customer_email
ORDER BY Total_Spent DESC;
```

100 %

Results | Messages

	customer_email	Total_Spent
1	brian28@example.com	806.00
2	robertserin@example.com	802.00
3	gutierrezvictoria@example.net	786.00
4	morrismichelle@example.org	774.00
5	dakota62@example.net	764.00
6	morrisryan@example.net	762.00
7	garysandoval@example.org	734.00
8	kirbymary@example.com	734.00
9	leesteele@example.net	732.00
10	jamiepreston@example.com	730.00
11	zburton@example.net	730.00
12	kelly87@example.net	726.00
13	deniseyoung@example.org	724.00
14	evaldez@example.net	724.00

```sql
--Trends over time (better used in graphcical view)
SELECT D.Month, D.Year, SUM(FS.NetRevenue) AS Monthly_Sales
FROM DWH.Fact_Sales FS
JOIN DWH.D_Date D ON FS.DateKey = D.DateKey
GROUP BY D.Year, D.Month
ORDER BY D.Year, D.Month;
```

109 %

Results | Messages

	Month	Year	Monthly_Sales
1	1	2016	3832.00
2	2	2016	3230.00
3	3	2016	3910.00
4	4	2016	3492.00
5	5	2016	3864.00
6	6	2016	3746.00
7	7	2016	3662.00
8	8	2016	3416.00
9	9	2016	3554.00
10	10	2016	3502.00
11	11	2016	3398.00
12	12	2016	3536.00
13	1	2017	3698.00
14	2	2017	3026.00
15	3	2017	3524.00
16	4	2017	3634.00
17	5	2017	3896.00
18	6	2017	3730.00
19	7	2017	3318.00
20	8	2017	3620.00
21	9	2017	3514.00
22	10	2017	3584.00
23	11	2017	3404.00
24	12	2017	3548.00
25	1	2018	3466.00

2. Queries on Campaign Performance Fact:

```sql
--Identify top 5 successed campgains
SELECT TOP 5
    campaign_id,
    SUM(net_sales) AS net_sales
FROM fact_campaign_performance
GROUP BY campaign_id
ORDER BY net_sales DESC;
```

109 %

Results | Messages

	campaign_id	net_sales
1	12	46382.83
2	5	43742.35
3	4	40558.91
4	15	39511.78
5	8	37301.00

SQLQuery3.sql - D...MIQOPLK\DELL (82))* | queries.sql - DESK...MIQOPLK\DELL (71))* | DESKTOP-MIQOPLK...e_DW - Diagram_1* | SQLQuery1.sql

```sql
--Summary of campgain performance
SELECT
    campaign_id,
    SUM(total_sales) AS total_sales,
    SUM(returned_amount) AS total_returns,
    SUM(net_sales) AS net_sales,
    ROUND((SUM(returned_amount) * 100.0) / NULLIF(SUM(total_sales), 0), 2) AS return_rate_percent
FROM fact_campaign_performance
GROUP BY campaign_id
ORDER BY net_sales DESC;
```

109 %

Results | Messages

	campaign_id	total_sales	total_returns	net_sales	return_rate_percent
1	12	54185.18	7802.35	46382.83	14.400000
2	5	49252.71	5510.36	43742.35	11.190000
3	4	47289.03	6730.12	40558.91	14.230000
4	15	46931.24	7419.46	39511.78	15.810000
5	8	42477.94	5176.94	37301.00	12.190000
6	9	42438.13	5778.31	36659.82	13.620000
7	1	40562.48	5160.64	35401.84	12.720000
8	14	43189.91	7953.56	35236.35	18.420000
9	16	38430.49	5557.21	32873.28	14.460000
10	6	36561.36	4891.95	31669.41	13.380000
11	2	34398.65	4499.27	29899.38	13.080000
12	13	32623.54	4735.03	27888.51	14.510000
13	11	31214.63	5879.53	25335.10	18.840000
14	3	28389.28	3958.19	24431.09	13.940000
15	7	26047.57	3777.61	22269.96	14.500000

3. Queries on Returns Fact:

```sql
SELECT
    AVG(processing_days) AS avg_processing_days
FROM fact_returns;
```

109 %

Results | Messages

	avg_processing_days
1	6

SQLQuery3.sql - D...MIQOPLK\DELL (82))* | queries.sql - DESK...MIQOPLK\DELL (71)) | DESKTOP-MIQOPLK...e_DW - Diagr

```sql
--Identify customers who return very frequntly
SELECT
    c.customer_key,
    c.customer_email,
    COUNT(r.return_fact_key) AS total_returns
FROM fact_returns r
JOIN DWH.D_Customer c ON r.customer_key = c.customer_key
GROUP BY c.customer_key,
c.customer_email
HAVING COUNT(r.return_fact_key) >= 9  -- Threshold of frequent returns
ORDER BY total_returns DESC;
```

109 %

Results | Messages

	customer_key	customer_email	total_returns
1	45	sarahlong@example.com	11
2	326	smithnicole@example.com	11
3	343	susan73@example.org	11
4	417	cmason@example.com	11
5	488	yorksteven@example.net	11
6	444	cherrera@example.com	10
7	469	xabbott@example.com	10
8	348	rodriguezelizabeth@example.com	10
9	357	hlevy@example.org	10
10	359	brandon07@example.com	10
11	398	morrisjustin@example.com	10
12	302	glenn66@example.net	10
13	309	stevenwatson@example.org	10
14	224	jacoblopez@example.org	10
15	16	brian28@example.com	10
16	36	sarah82@example.net	10
17	125	christinaarmstrong@example.com	10
18	127	melissa73@example.net	9
19	138	thomasjohnson@example.net	9
20	154	caitlin53@example.com	9
21	162	karen09@example.com	9

```sql
--Identify the Top 3 reason for return products
SELECT TOP 3
R.reason, COUNT(R.reason) As Reason_count
FROM Fact_returns FR
JOIN DWH.D_return R ON FR.return_junk_key= R.junk_key
GROUP BY R.reason
ORDER BY Reason_count DESC;
```

09 %

Results | Messages

	reason	Reason_count
1	Changed my mind	278
2	Color doesn't match	274
3	Found a better deal	266

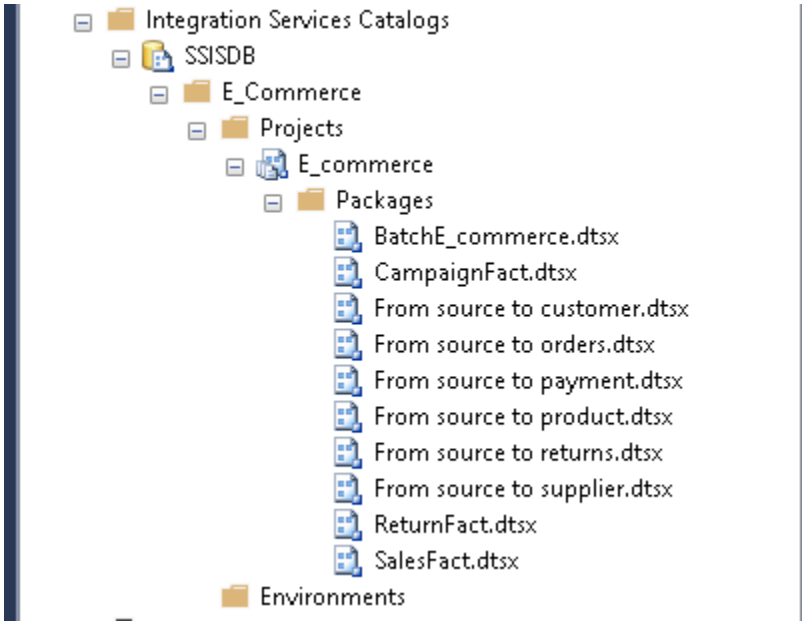
```sql
--Show return trend ( better visualized)
SELECT
    d.Year,
    d.Month,
    COUNT(r.return_fact_key) AS total_returns
FROM fact_returns r
JOIN DWH.D_Date d ON r.date_key = d.DateKey
GROUP BY d.Year, d.Month
ORDER BY d.Year, d.Month;
```

109 %

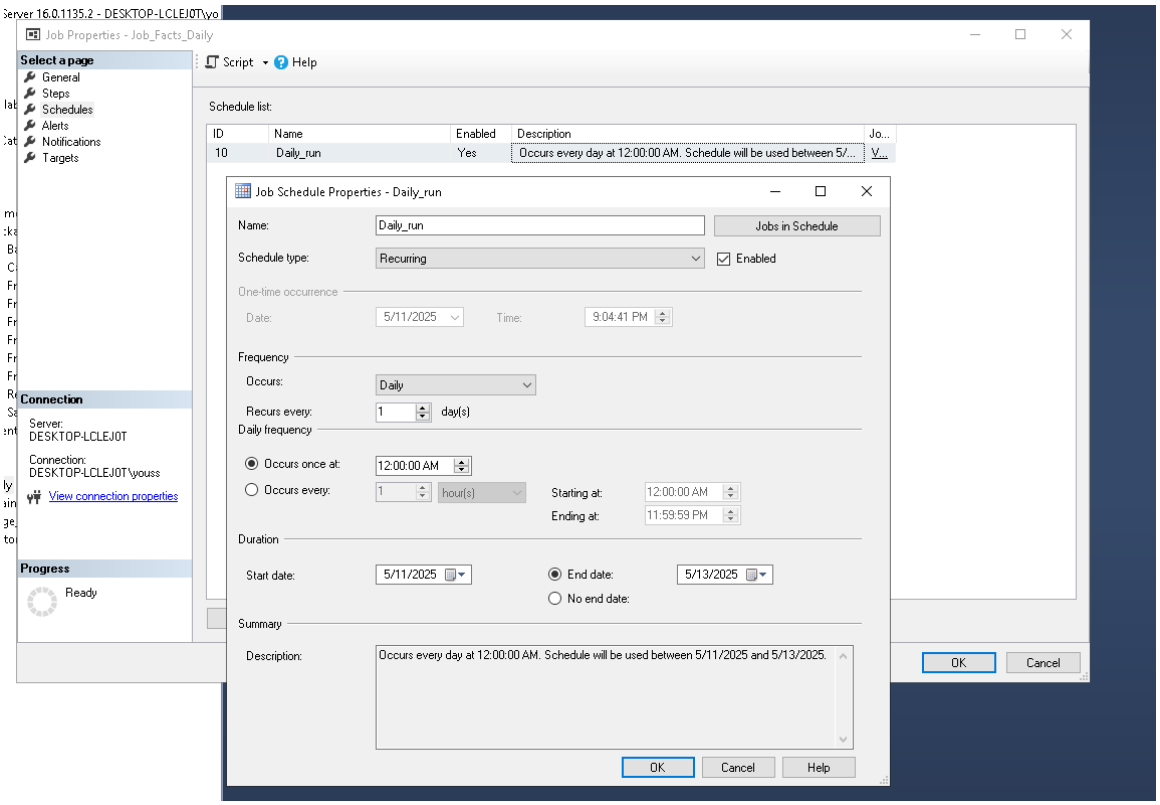
Results | Messages

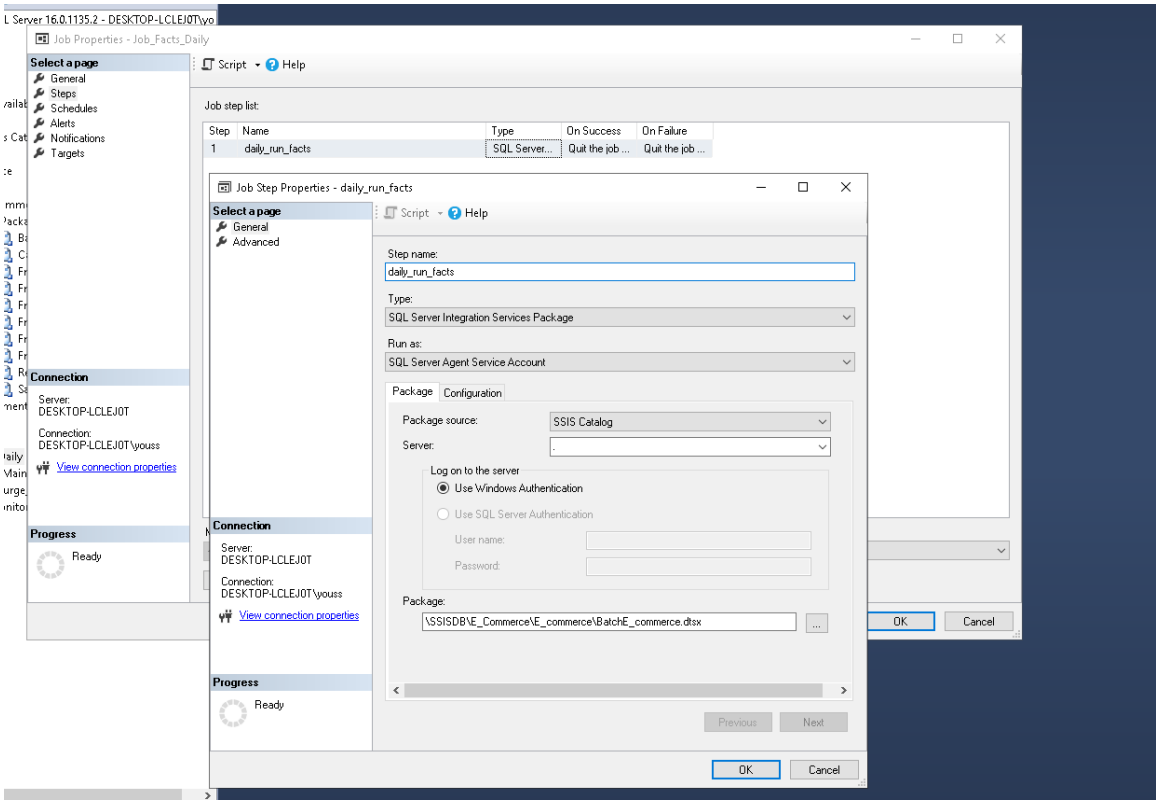
	Year	Month	total_returns
1	2016	1	25
2	2016	2	37
3	2016	3	32
4	2016	4	25
5	2016	5	27
6	2016	6	21
7	2016	7	34
8	2016	8	26
9	2016	9	30
10	2016	10	26
11	2016	11	30
12	2016	12	31
13	2017	1	28
14	2017	2	38
15	2017	3	29
16	2017	4	40
17	2017	5	34
18	2017	6	27
19	2017	7	34
20	2017	8	27

V. Deployed Packages:

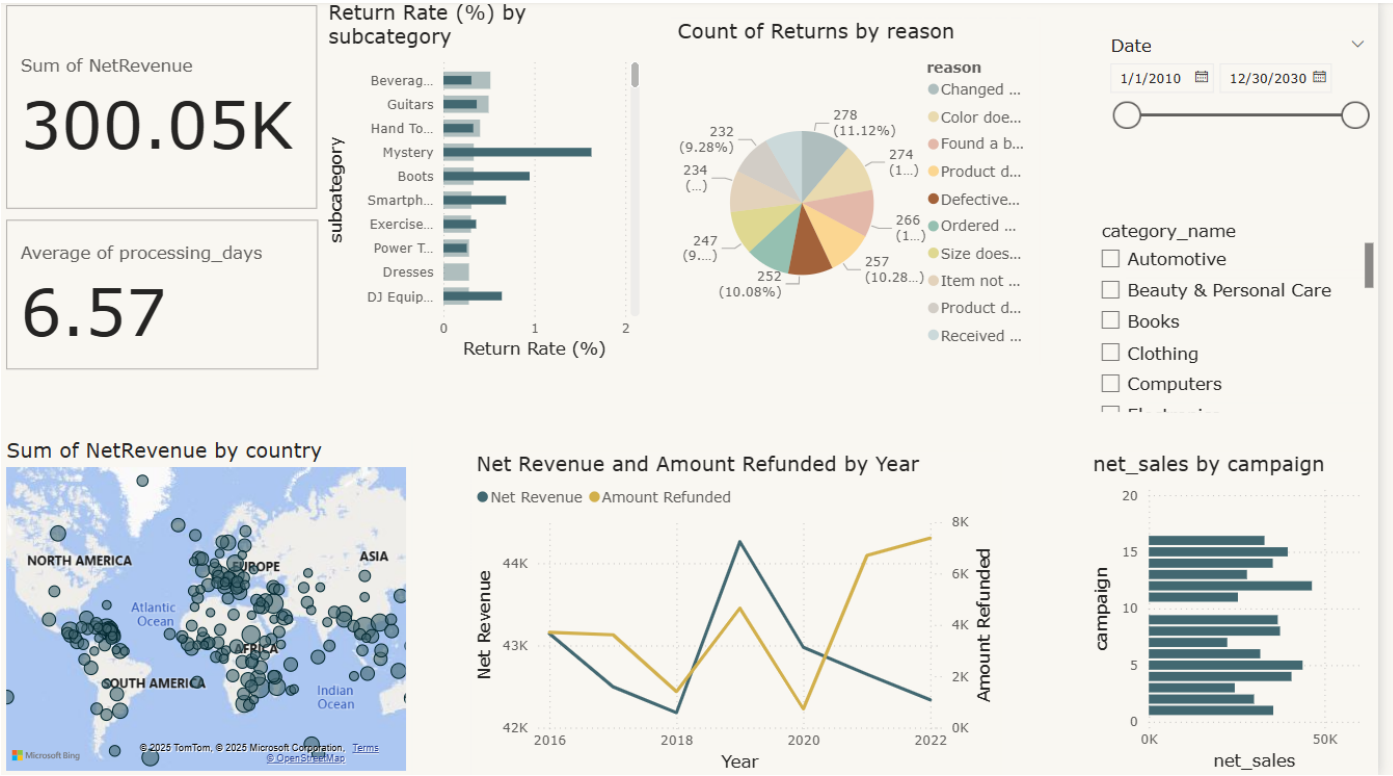


Their Schedules:




Job Properties - Job_Facts_Daily
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Job step list:
Step
Name
Type
On Success
On Failure
1
daily_run_facts
SQL Server...
Quit the job ...
Quit the job ...
Job Step Properties - daily_run_facts
Select a page
General
Advanced
Step name:
daily_run_facts
Type:
SQL Server Integration Services Package
Run as:
SQL Server Agent Service Account
Package
Configuration
Package source:
SSIS Catalog
Server:
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Package:
\SSISDB\E_Commerce\E_commerce\BatchE_commerce.dtsx
Connection
Server:
DESKTOP-LCLEJ0T
Connection:
DESKTOP-LCLEJ0T\youss
View connection properties
Progress
Ready
OK
Cancel
Previous
Next

VI. Power BI:


Sum of NetRevenue
300.05K
Average of processing_days
6.57
Return Rate (%) by subcategory
Beverag...
Guitars
Hand To...
Mystery
Boots
Smartph...
Exercise...
Power T...
Dresses
DJ Equip...
subcategory
0
1
2
Return Rate (%)
Count of Returns by reason
reason
Changed ...
Color doe...
Found a b...
Product d...
Defective...
Ordered ...
Size does...
Item not ...
Product d...
Received ...
278 (11.12%)
274 (1...)
266 (1...)
257 (10.28...)
252 (10.08%)
247 (9....)
234 (...)
232 (9.28%)
Date
1/1/2010
12/30/2030
category_name
Automotive
Beauty & Personal Care
Books
Clothing
Computers
Sum of NetRevenue by country
NORTH AMERICA
EUROPE
ASIA
Atlantic Ocean
AFRICA
SOUTH AMERICA
Indian Ocean
© 2025 TomTom, © 2025 Microsoft Corporation, Terms
© OpenStreetMap
Net Revenue and Amount Refunded by Year
Net Revenue
Amount Refunded
44K
43K
42K
8K
6K
4K
2K
0K
2016
2018
2020
2022
Year
net_sales by campaign
campaign
20
15
10
5
0
0K
50K
net_sales