NCPL SALES

Microsoft Fabric End-to-End Data Pipeline

Table Of Content		

Introduction

The goal of this project is to design and implement a fully functional end-to-end data engineering and analytics solution using Microsoft Fabric. The solution will ingest raw data from source systems, process and refine it through multiple transformation layers (such as bronze, silver, and gold in a medallion architecture), implement scalable data pipelines using Fabric components (e.g., pipelines, notebooks), and finally deliver actionable insights through reporting and analytics. The project will demonstrate how to build an integrated data platform that supports data cleansing, transformation, storage, and visualization at scale, ensuring data quality, governance, and performance throughout the pipeline.

Project Scope

1. Data Ingestion

- Collect data from various structured and unstructured data sources.
- Implement automated ingestion using Microsoft Fabric data pipelines.
- Ensure ingestion supports incremental and batch loads.

2. Medallion Architecture Implementation

- **Bronze Layer:** Store raw ingested data in its original form.
- **Silver Layer:** Clean, join, filter, and enrich the raw data.
- **Gold Layer:** Aggregate refined data to support analytics and reporting use cases.

3. Data Transformation and Processing

- Use PySpark notebooks within Microsoft Fabric to apply business logic to data.
- Manage transformation workflows and dependencies between pipeline stages.

4. Data Storage and Optimization

- Establish Lakehouse storage with structured tables optimized for querying.
- Implement partitioning, schema enforcement, and performance optimization practices.

5. Reporting & Visualization

- Build dashboards and reports (e.g., using Power BI connected to Fabric outputs).
- Ensure insights are consumable by business users.

Technologies used

🎯 Key Technologies

- **Microsoft Fabric** platform (Lakehouse, Data Pipelines, Notebooks)
- **PySpark** for data transformation and enrichment
- **Power BI** analytics
- Medallion architecture for data refinement

🥇 What is Medallion Architecture?

Medallion Architecture is a **layered data design pattern** used in modern data platforms (like Microsoft Fabric, Databricks, Azure) to progressively refine data from **raw to business-ready** formats.

🔁 Core Idea

Data quality and value **increase** as data moves through layers.

🥉🥈🥇 Medallion Layers

🥉 Bronze Layer (Raw Data)

- Stores data **as-is** from source systems

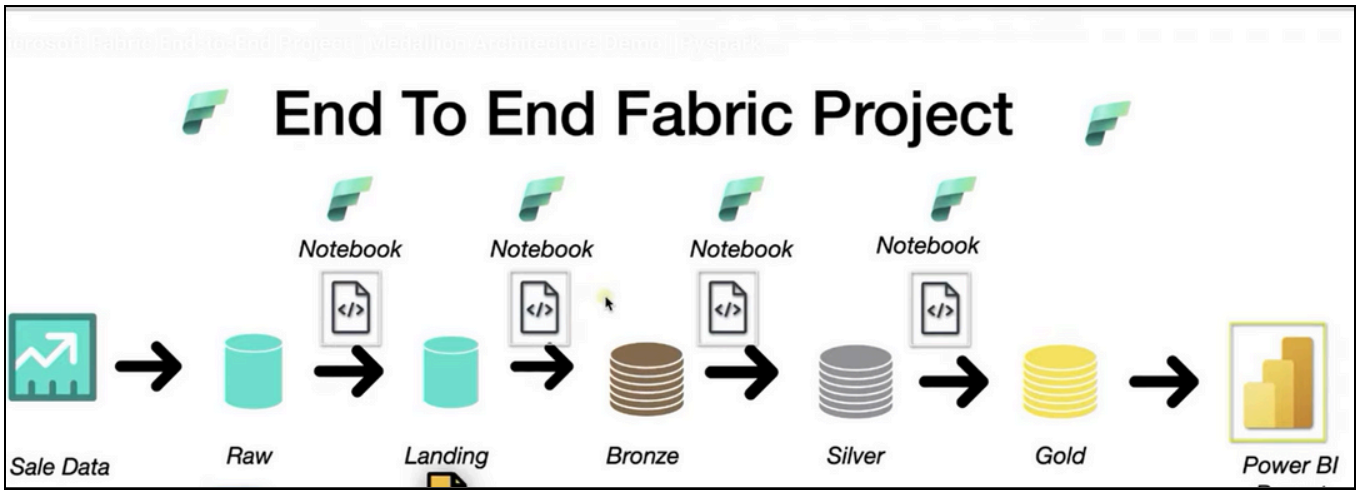
🥈 Silver Layer (Cleaned Data)

- Data is cleaned, validated, and enriched
- Duplicates removed, nulls handled
- Business rules applied

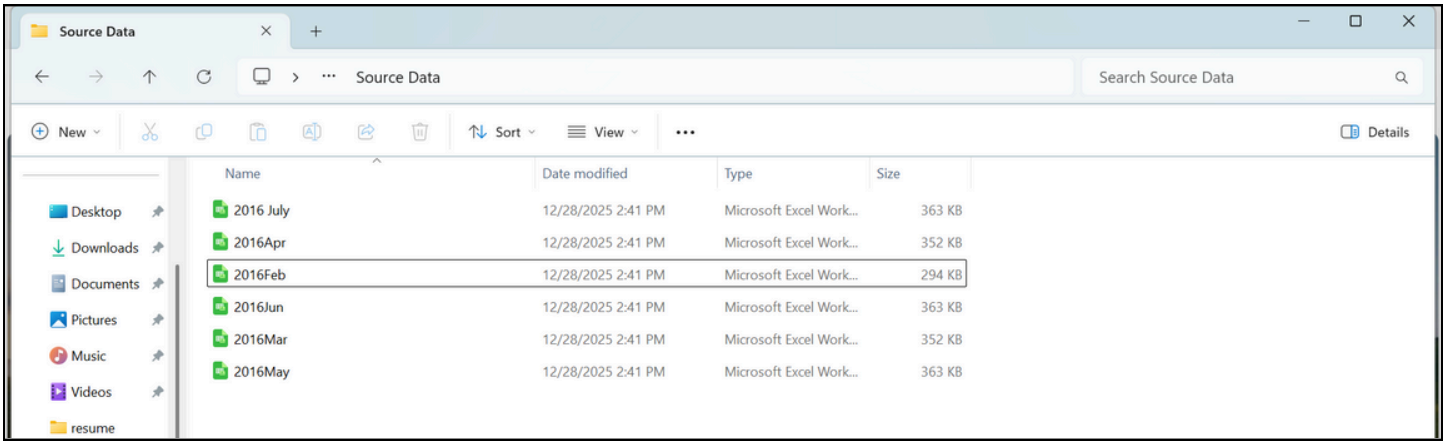
🥇 Gold Layer (Business Data)

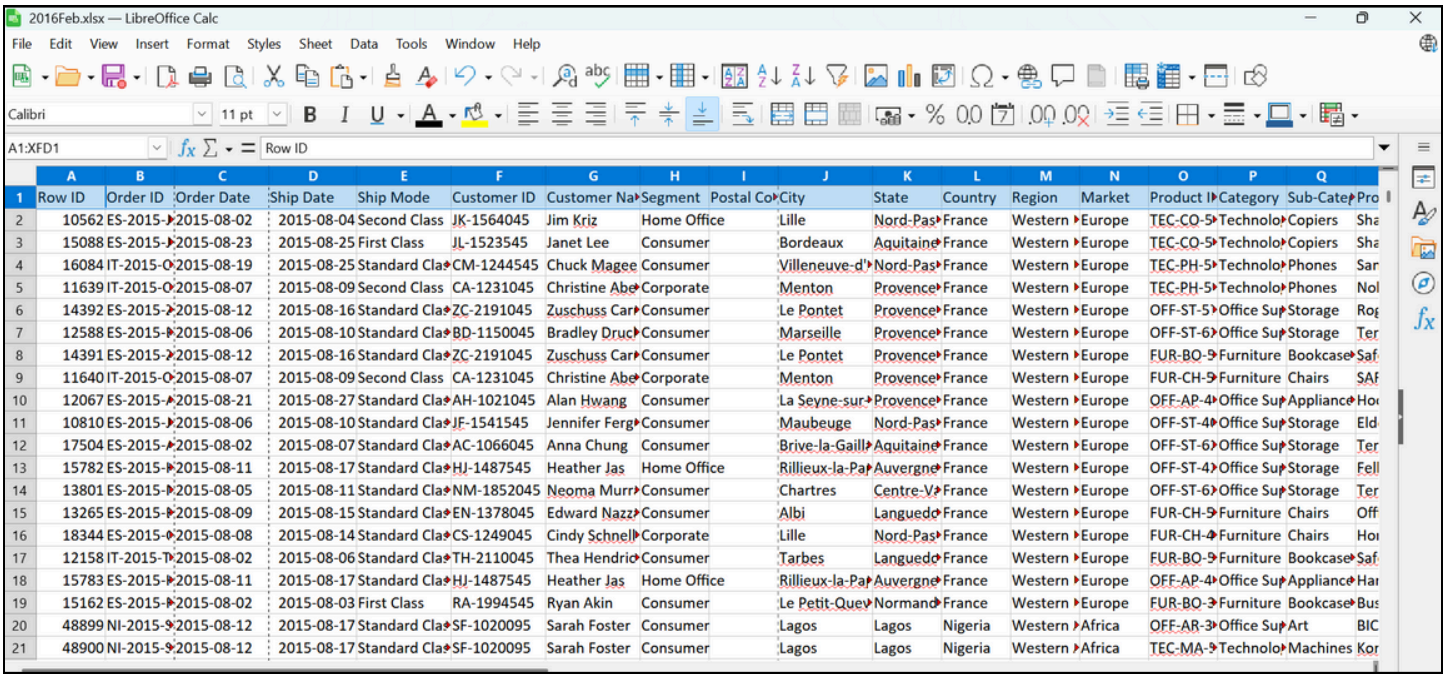
- Aggregated and curated data
- Optimized for reporting and analytics
- Used by BI tools

Architecture Diagram



Datasets

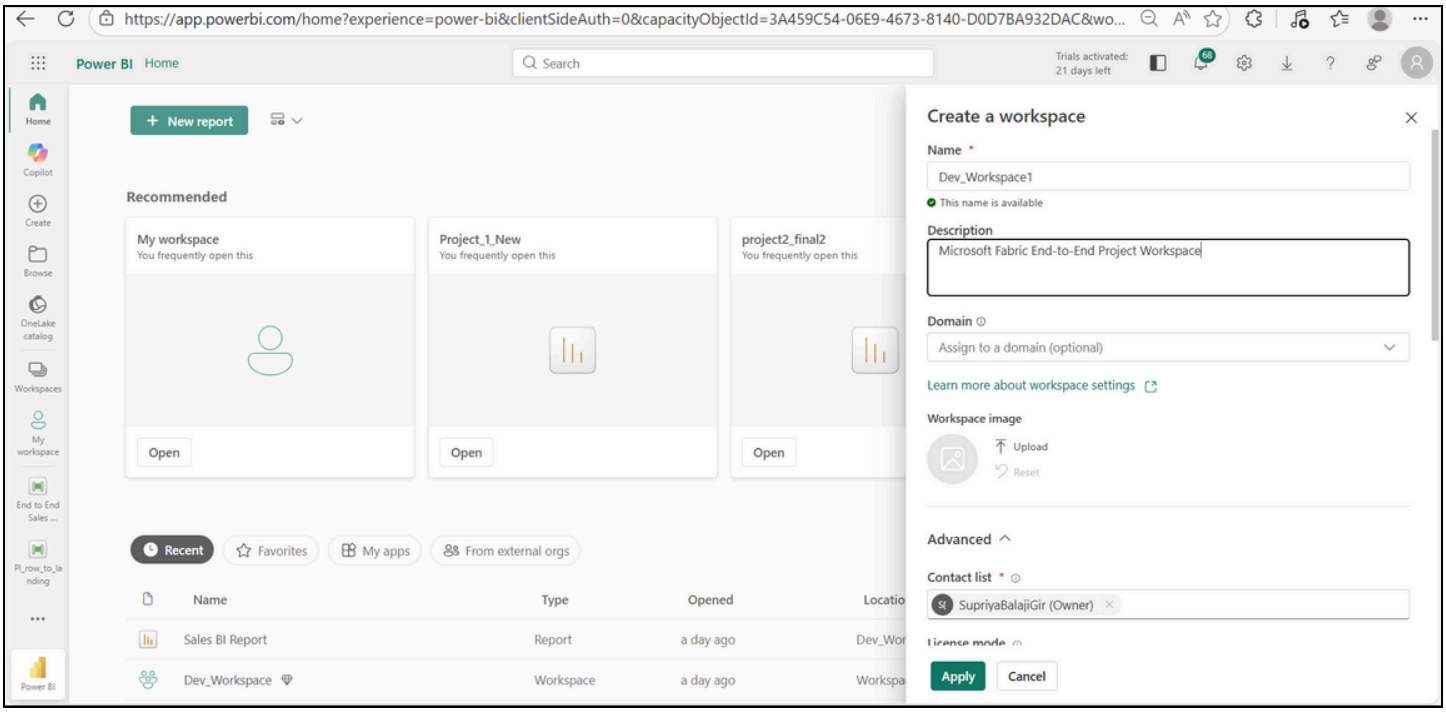




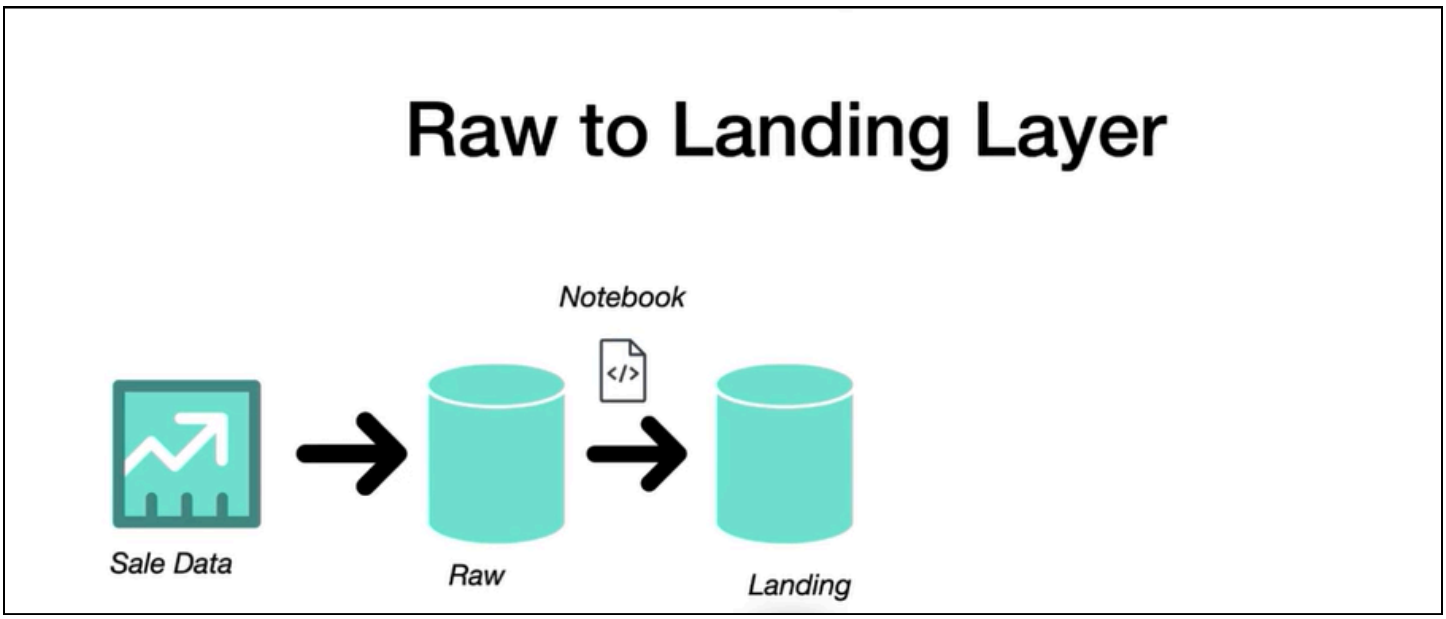
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Na	Segment	Postal Co	City	State	Country	Region	Market	Product I	Category	Sub-Cate
2	10562	ES-2015-	2015-08-02	2015-08-04	Second Class	JK-1564045	Jim Kriz	Home Office		Lille	Nord-Pas	France	Western	Europe	TEC-CO-5	Technolo	Copiers
3	15088	ES-2015-	2015-08-23	2015-08-25	First Class	JL-1523545	Janet Lee	Consumer		Bordeaux	Aquitaine	France	Western	Europe	TEC-CO-5	Technolo	Copiers
4	16084	IT-2015-0	2015-08-19	2015-08-25	Standard Cla	CM-1244545	Chuck Magee	Consumer		Villeneuve-d'	Nord-Pas	France	Western	Europe	TEC-PH-5	Technolo	Phones
5	11639	IT-2015-0	2015-08-07	2015-08-09	Second Class	CA-1231045	Christine Abe	Corporate		Menton	Provence	France	Western	Europe	TEC-PH-5	Technolo	Phones
6	14392	ES-2015-	2015-08-12	2015-08-16	Standard Cla	ZC-2191045	Zuschuss Car	Consumer		Le Pontet	Provence	France	Western	Europe	OFF-ST-5	Office Sup	Storage
7	12588	ES-2015-	2015-08-06	2015-08-10	Standard Cla	BD-1150045	Bradley Druc	Consumer		Marseille	Provence	France	Western	Europe	OFF-ST-6	Office Sup	Storage
8	14391	ES-2015-	2015-08-12	2015-08-16	Standard Cla	ZC-2191045	Zuschuss Car	Consumer		Le Pontet	Provence	France	Western	Europe	FUR-BO-5	Furniture	Bookcase
9	11640	IT-2015-0	2015-08-07	2015-08-09	Second Class	CA-1231045	Christine Abe	Corporate		Menton	Provence	France	Western	Europe	FUR-CH-5	Furniture	Chairs
10	12067	ES-2015-	2015-08-21	2015-08-27	Standard Cla	AH-1021045	Alan Hwang	Consumer		La Seyne-sur	Provence	France	Western	Europe	OFF-AP-4	Office Sup	Appliance
11	10810	ES-2015-	2015-08-06	2015-08-10	Standard Cla	JF-1541545	Jennifer Ferg	Consumer		Maubeuge	Nord-Pas	France	Western	Europe	OFF-ST-4	Office Sup	Storage
12	17504	ES-2015-	2015-08-02	2015-08-07	Standard Cla	AC-1066045	Anna Chung	Consumer		Brive-la-Gaill	Aquitaine	France	Western	Europe	OFF-ST-6	Office Sup	Storage
13	15782	ES-2015-	2015-08-11	2015-08-17	Standard Cla	HJ-1487545	Heather Jas	Home Office		Rillieux-la-Pa	Auvergne	France	Western	Europe	OFF-ST-4	Office Sup	Storage
14	13801	ES-2015-	2015-08-05	2015-08-11	Standard Cla	NM-1852045	Neoma Murr	Consumer		Chartres	Centre-V	France	Western	Europe	OFF-ST-6	Office Sup	Storage
15	13265	ES-2015-	2015-08-09	2015-08-15	Standard Cla	EN-1378045	Edward Nazz	Consumer		Albi	Languedo	France	Western	Europe	FUR-CH-5	Furniture	Chairs
16	18344	ES-2015-	2015-08-08	2015-08-14	Standard Cla	CS-1249045	Cindy Schnell	Corporate		Lille	Nord-Pas	France	Western	Europe	FUR-CH-4	Furniture	Chairs
17	12158	IT-2015-T	2015-08-02	2015-08-06	Standard Cla	TH-2110045	Thea Hendric	Consumer		Tarbes	Languedo	France	Western	Europe	FUR-BO-5	Furniture	Bookcase
18	15783	ES-2015-	2015-08-11	2015-08-17	Standard Cla	HJ-1487545	Heather Jas	Home Office		Rillieux-la-Pa	Auvergne	France	Western	Europe	OFF-AP-4	Office Sup	Appliance
19	15162	ES-2015-	2015-08-02	2015-08-03	First Class	RA-1994545	Ryan Akin	Consumer		Le Petit-Quev	Normand	France	Western	Europe	FUR-BO-3	Furniture	Bookcase
20	48899	NI-2015-9	2015-08-12	2015-08-17	Standard Cla	SF-1020095	Sarah Foster	Consumer		Lagos	Lagos	Nigeria	Western	Africa	OFF-AR-3	Office Sup	Art
21	48900	NI-2015-9	2015-08-12	2015-08-17	Standard Cla	SF-1020095	Sarah Foster	Consumer		Lagos	Lagos	Nigeria	Western	Africa	TEC-MA-9	Technolo	Machines

Step 1: Data Ingestion

Creating New Workspace:



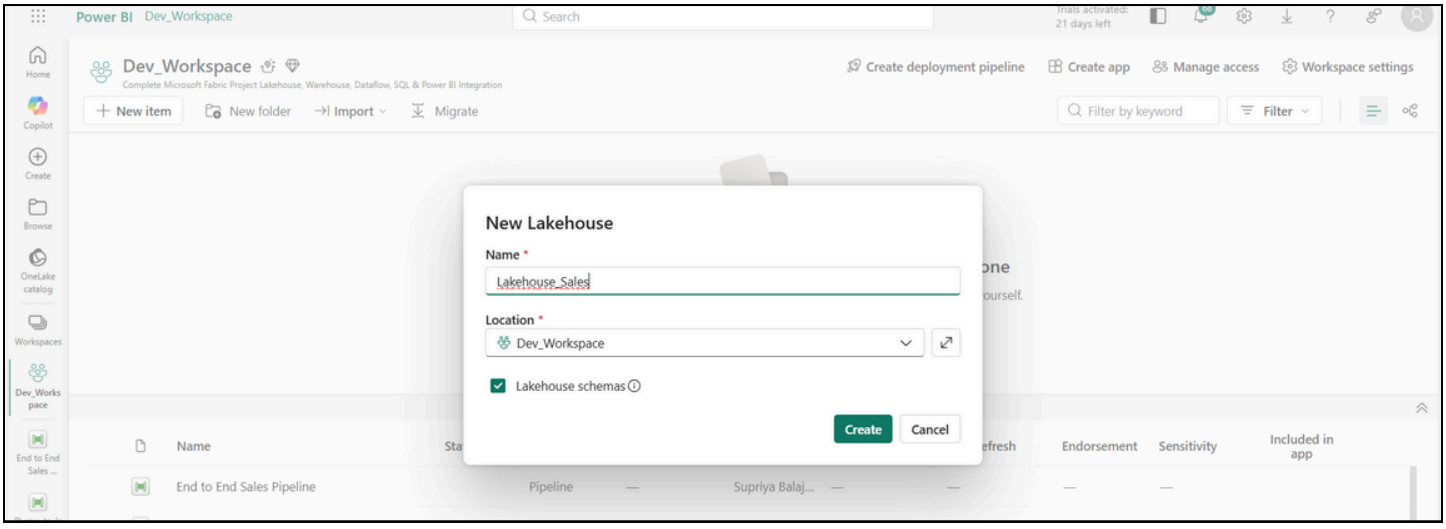
Step 2: Medallion Architecture Implementation



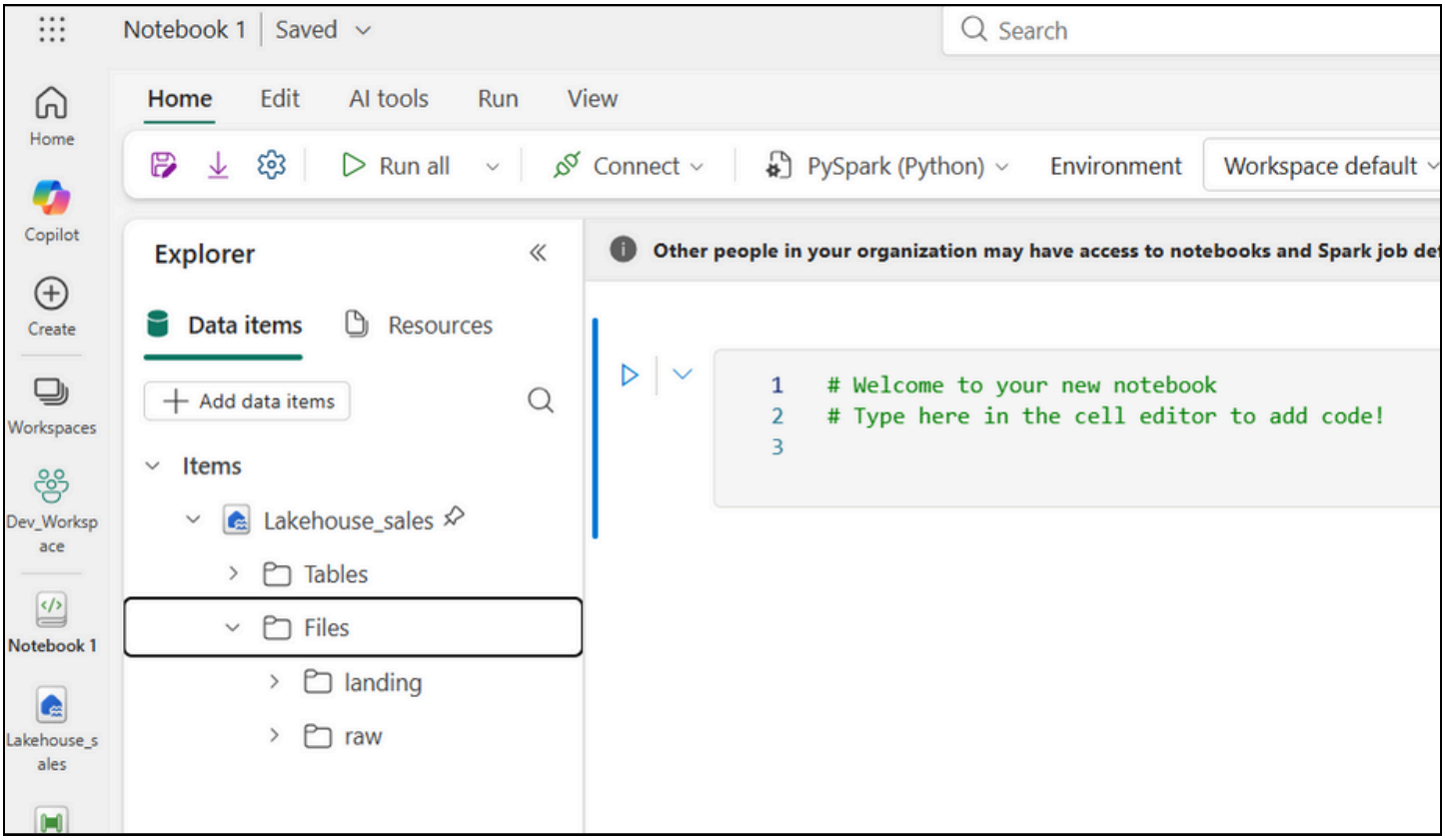
Monthly only one file will be come from sales Department, and we will save it to raw folder.

Pipeline will pick up latest file from raw folder based on last modified date and will be copied to landing folder and Processing date will be added.

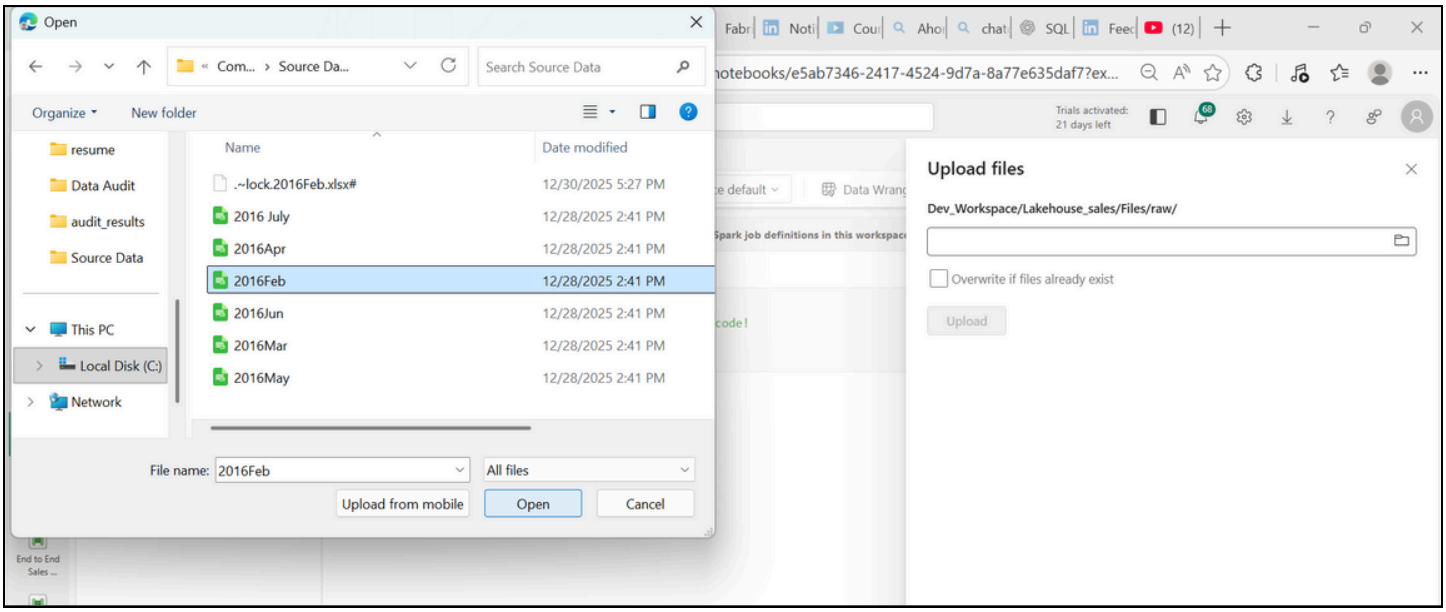
Creating Lakehouse in Fabric:

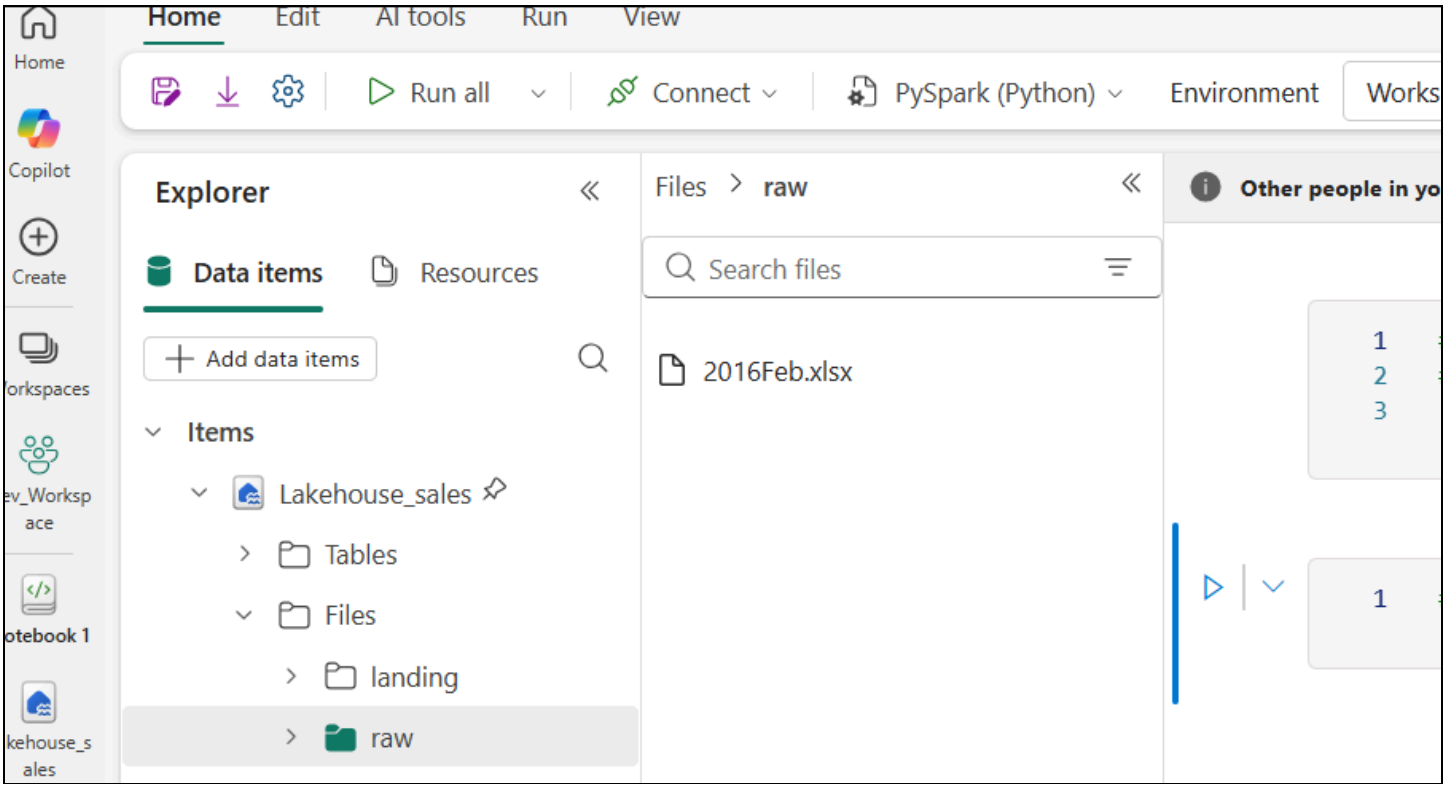


Create 2 folders under files section of lakehouse:



Upload file to raw folder:





Step 3: Creating Notebook to copy data from raw to landing folder

In notebook first take two parameters, later these parameters will get data from pipeline. For timebeing we are hardcoding it:

```
1  file_name = '2016Feb.csv'
2  processed_date = '2025-12-28'
```

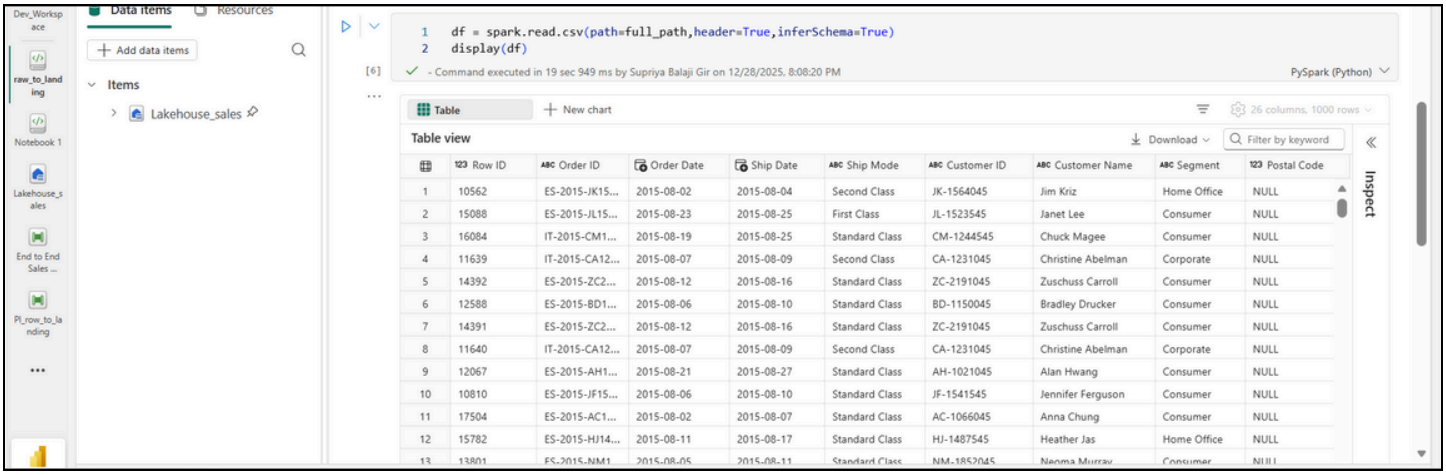
[4] ✓ - Command executed in 332 ms by Supriya Balaji Gir on 12/28/2025, 8:07:56 PM

Taking path of raw folder:

```
1  adfs_path = 'abfss://Dev_Workspace@onelake.dfs.fabric.microsoft.com/Lakehouse_sales.Lakehouse/Files/raw'
2
3  full_path = f"{adfs_path}/{file_name}"
4  print(full_path)
```

✓ - Command executed in 268 ms by Supriya Balaji Gir on 12/28/2025, 8:07:58 PM PySpark (F

Read and display data to pandas Dataframe:



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Postal Code
1	10562	ES-2015-JK15...	2015-08-02	2015-08-04	Second Class	JK-1564045	Jim Kriz	Home Office	NULL
2	15088	ES-2015-JL15...	2015-08-23	2015-08-25	First Class	JL-1523545	Janet Lee	Consumer	NULL
3	16084	IT-2015-CM1...	2015-08-19	2015-08-25	Standard Class	CM-1244545	Chuck Magee	Consumer	NULL
4	11639	IT-2015-CA12...	2015-08-07	2015-08-09	Second Class	CA-1231045	Christine Abelman	Corporate	NULL
5	14392	ES-2015-ZC2...	2015-08-12	2015-08-16	Standard Class	ZC-2191045	Zuschuss Carroll	Consumer	NULL
6	12588	ES-2015-BD1...	2015-08-06	2015-08-10	Standard Class	BD-1150045	Bradley Drucker	Consumer	NULL
7	14391	ES-2015-ZC2...	2015-08-12	2015-08-16	Standard Class	ZC-2191045	Zuschuss Carroll	Consumer	NULL
8	11640	IT-2015-CA12...	2015-08-07	2015-08-09	Second Class	CA-1231045	Christine Abelman	Corporate	NULL
9	12067	ES-2015-AH1...	2015-08-21	2015-08-27	Standard Class	AH-1021045	Alan Hwang	Consumer	NULL
10	10810	ES-2015-JF15...	2015-08-06	2015-08-10	Standard Class	JF-1541545	Jennifer Ferguson	Consumer	NULL
11	17504	ES-2015-AC1...	2015-08-02	2015-08-07	Standard Class	AC-1066045	Anna Chung	Consumer	NULL
12	15782	ES-2015-HJ14...	2015-08-11	2015-08-17	Standard Class	HJ-1487545	Heather Jas	Home Office	NULL

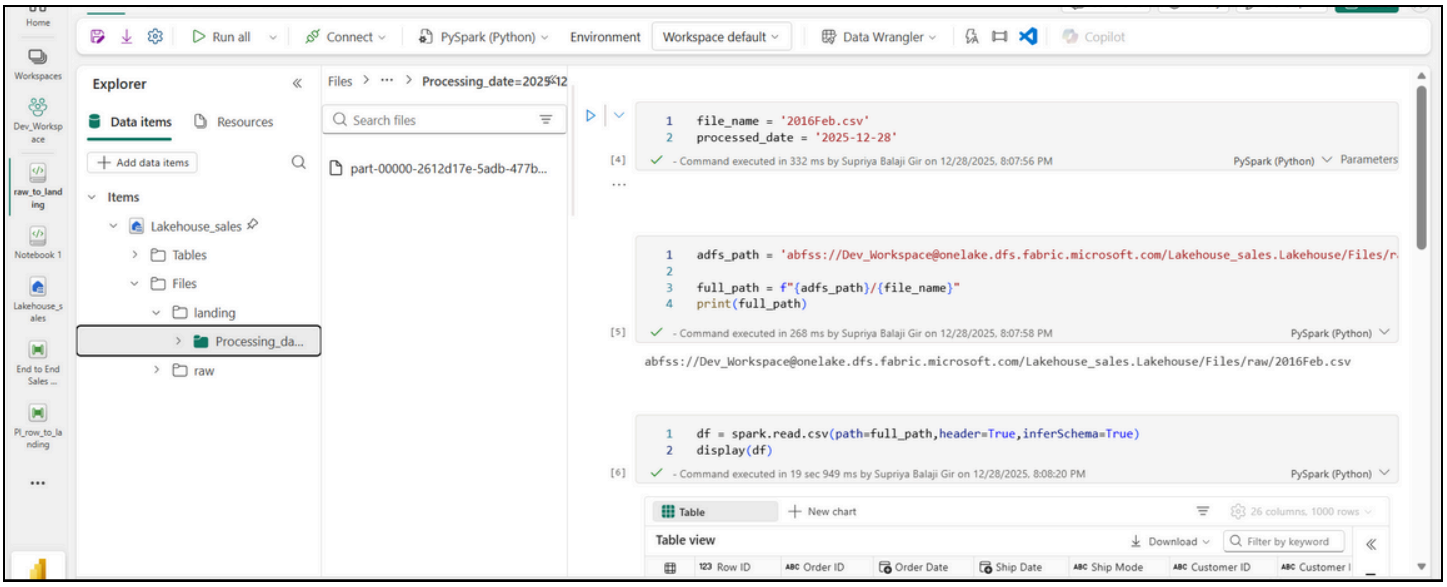
Now load this Dataframe data to landing folder with todays' processing date:

```
1  from pyspark.sql.functions import lit
2
3  if df.count() > 1:
4      df_new = df.withColumn("Processing_date",lit(processed_date))
5
6      #display(df_new)
7      df_new.write.format('csv').option("header","True").mode("append").partitionBy("Processing_date").save("abfss://Dev_Workspace@onelake.
8      print("Data loaded to Landing zone successfully")
9  else:
10     print("File contain no Data")
```

[7] ✓ - Command executed in 3 sec 246 ms by Supriya Balaji Gir on 12/28/2025, 8:08:31 PM PySpark (Python)

Diagnostics 1

Data loaded to Landing zone successfully



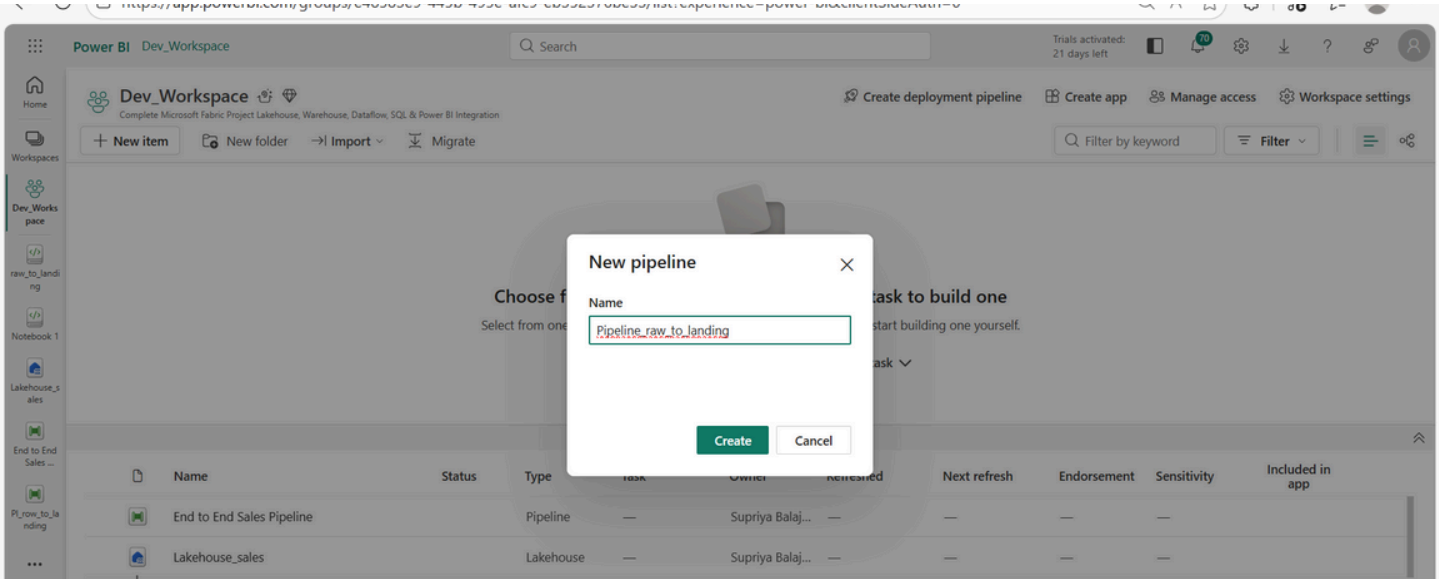
Now for incremental data upload to landing folder for file of every month we have to make the processing_date filed and file name filed parameterized.

Notebook should select current file of month and that days date as processing date:

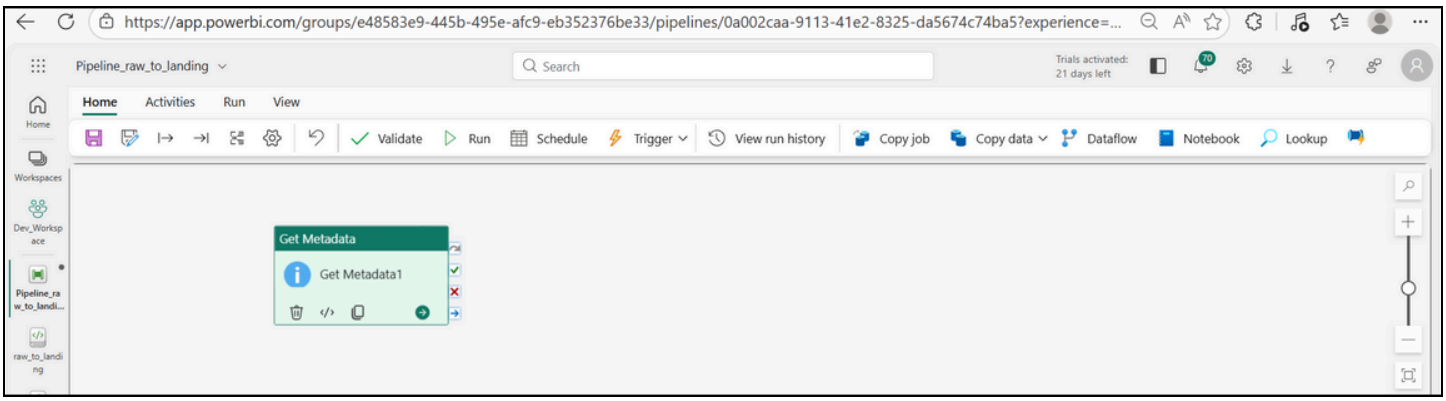
Step 4: Transfer data from raw to landing folder using pipeline

Creating Data Pipeline for the same:

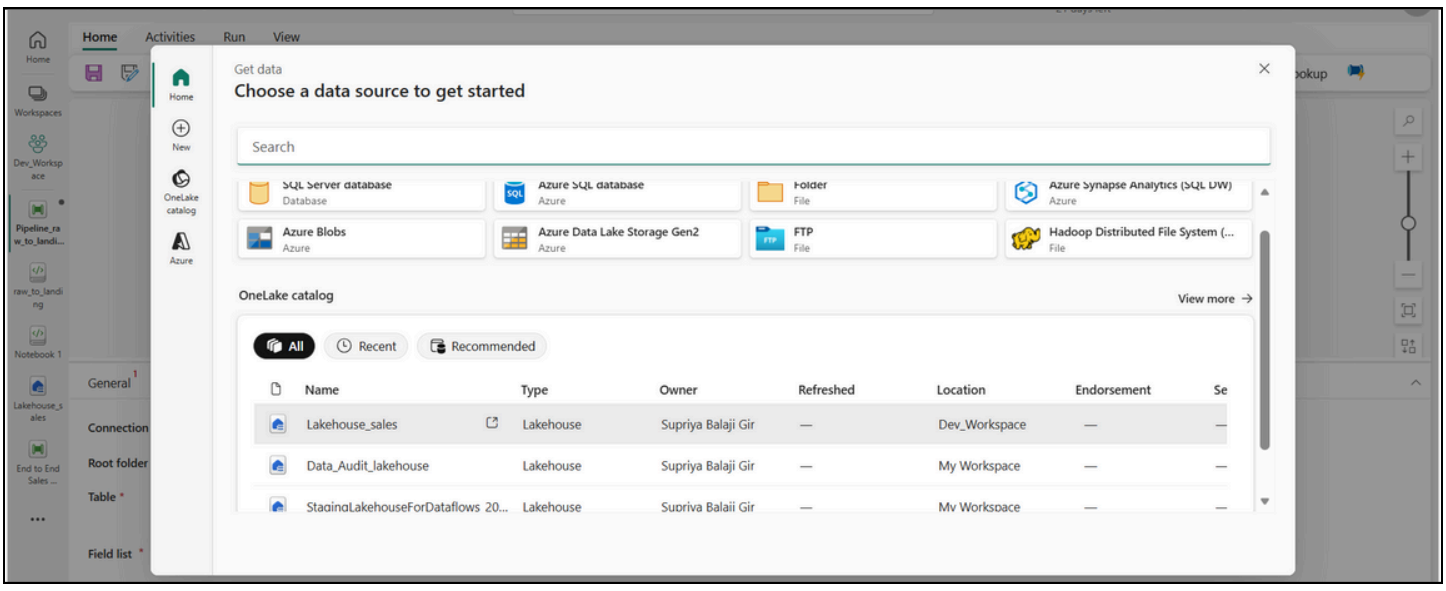
delete manually added file and we will add this file using pipeline.



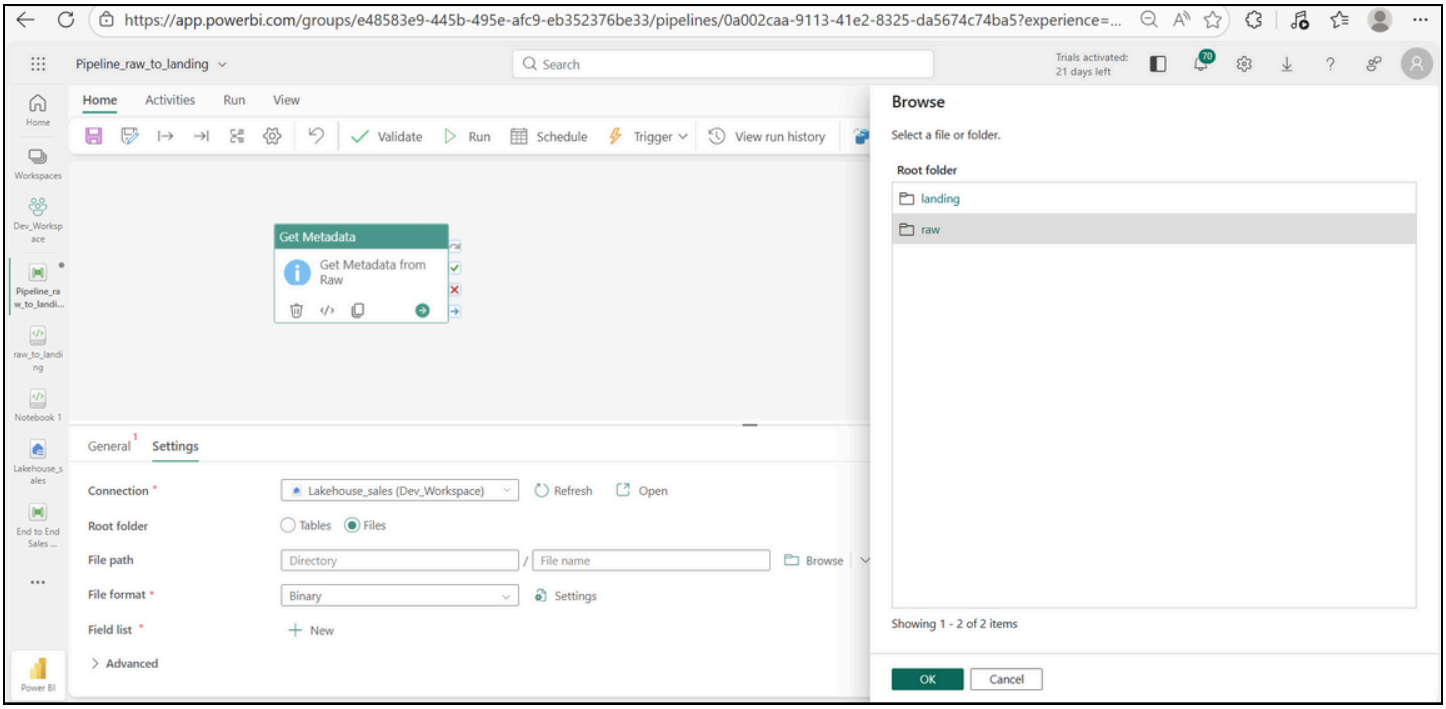
First add GET_Metadata activity to pipeline:



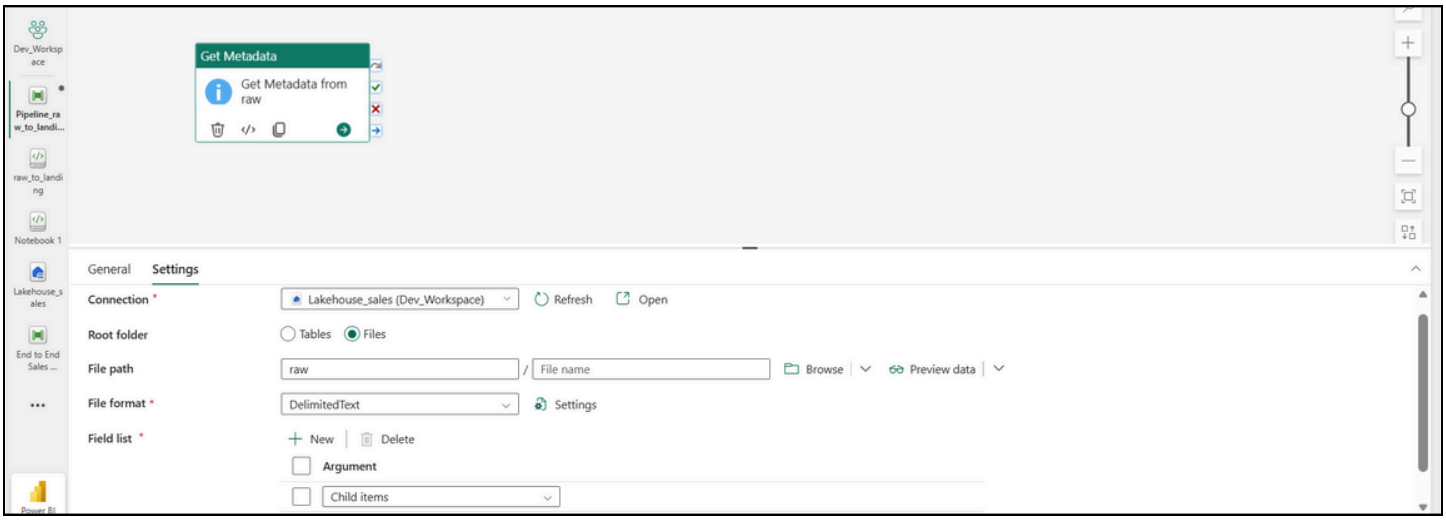
Select your lakehouse in setting tab of activity:



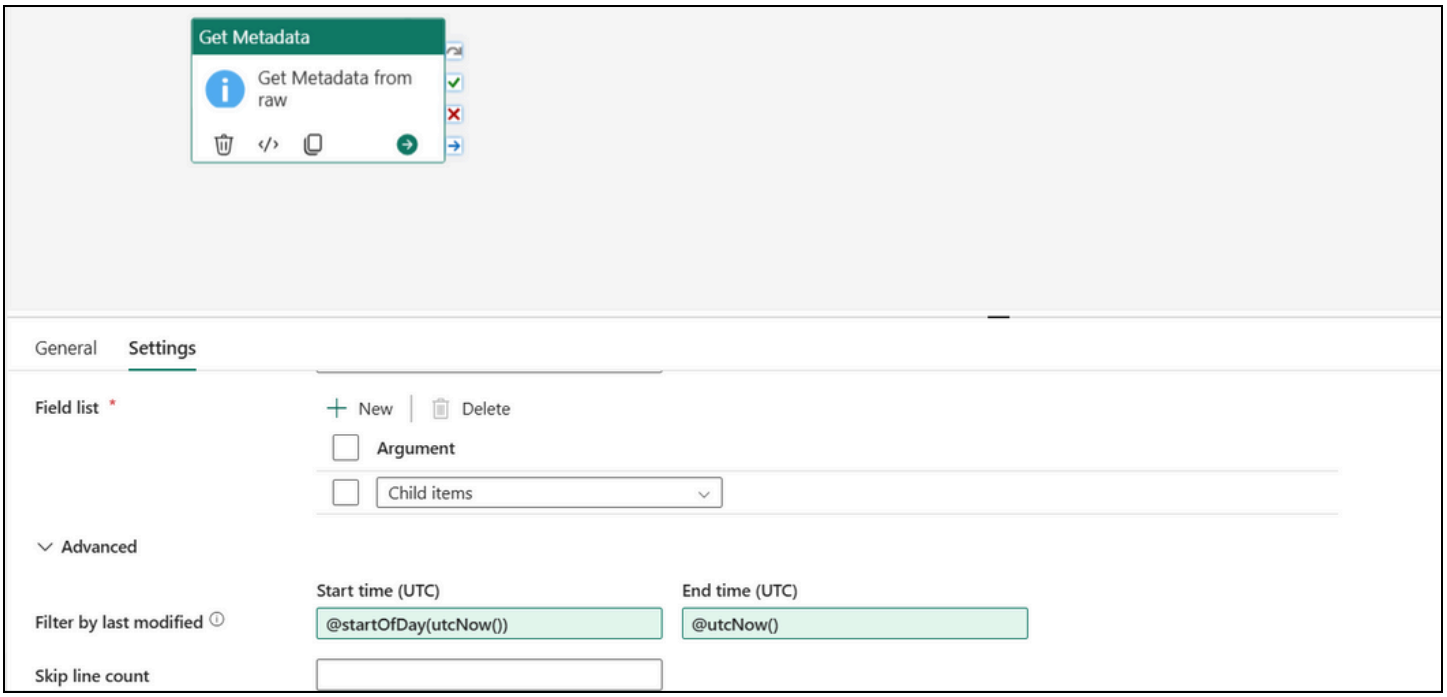
Select root folder as Files. Browse and select raw folder.



under filed list select the child item, so that it will get the filename:

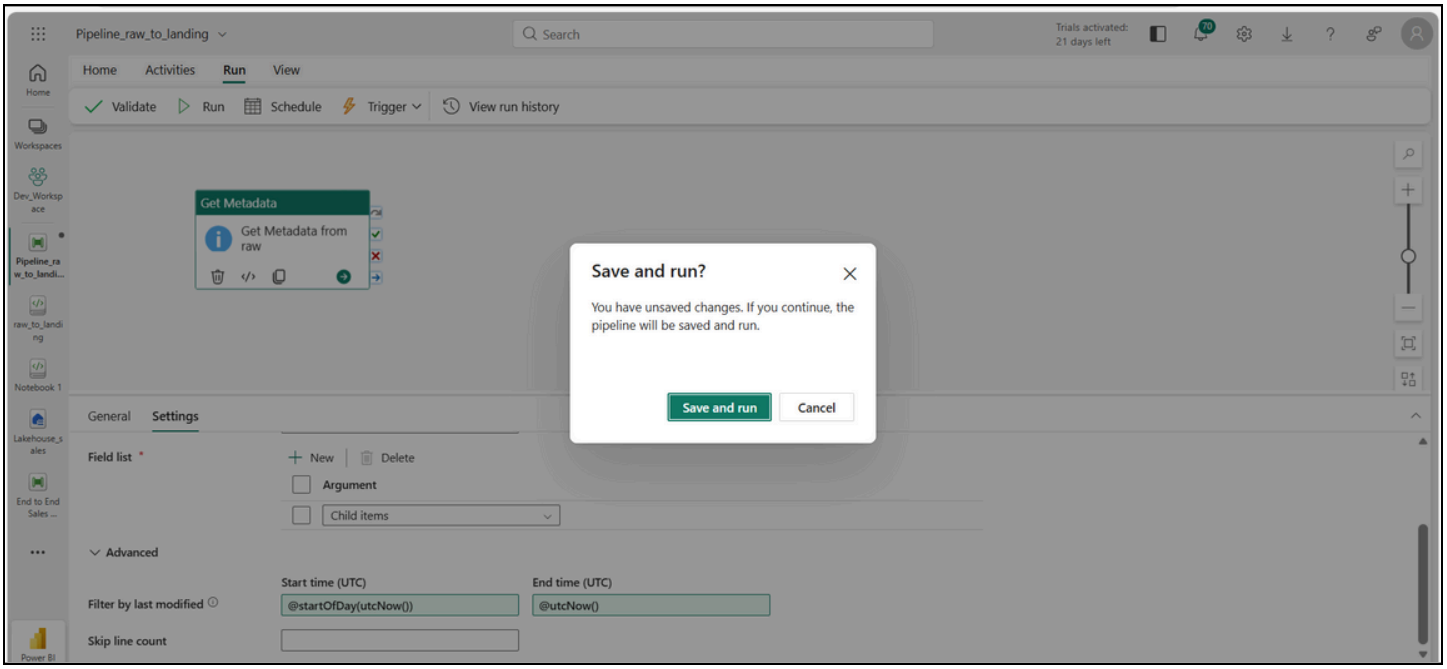


Select start and end time:

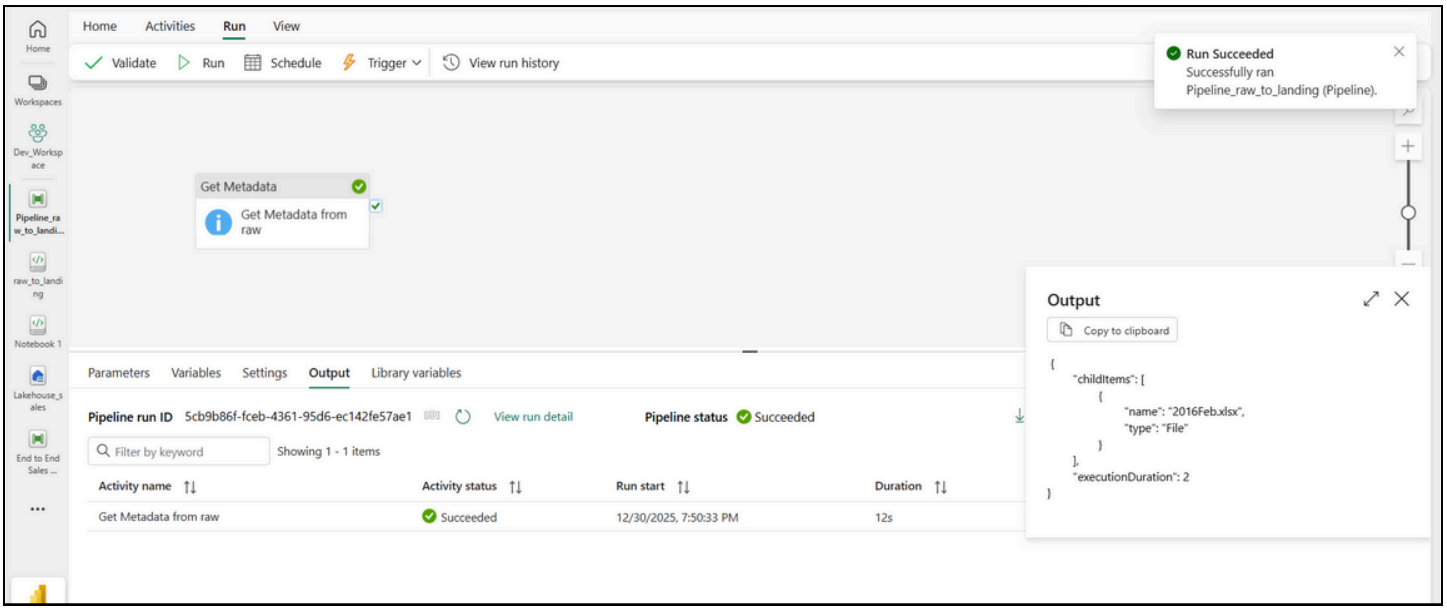


Now add file to raw folder and this file name will get in Get metadata activity of pipeline:

Save and run Activity:

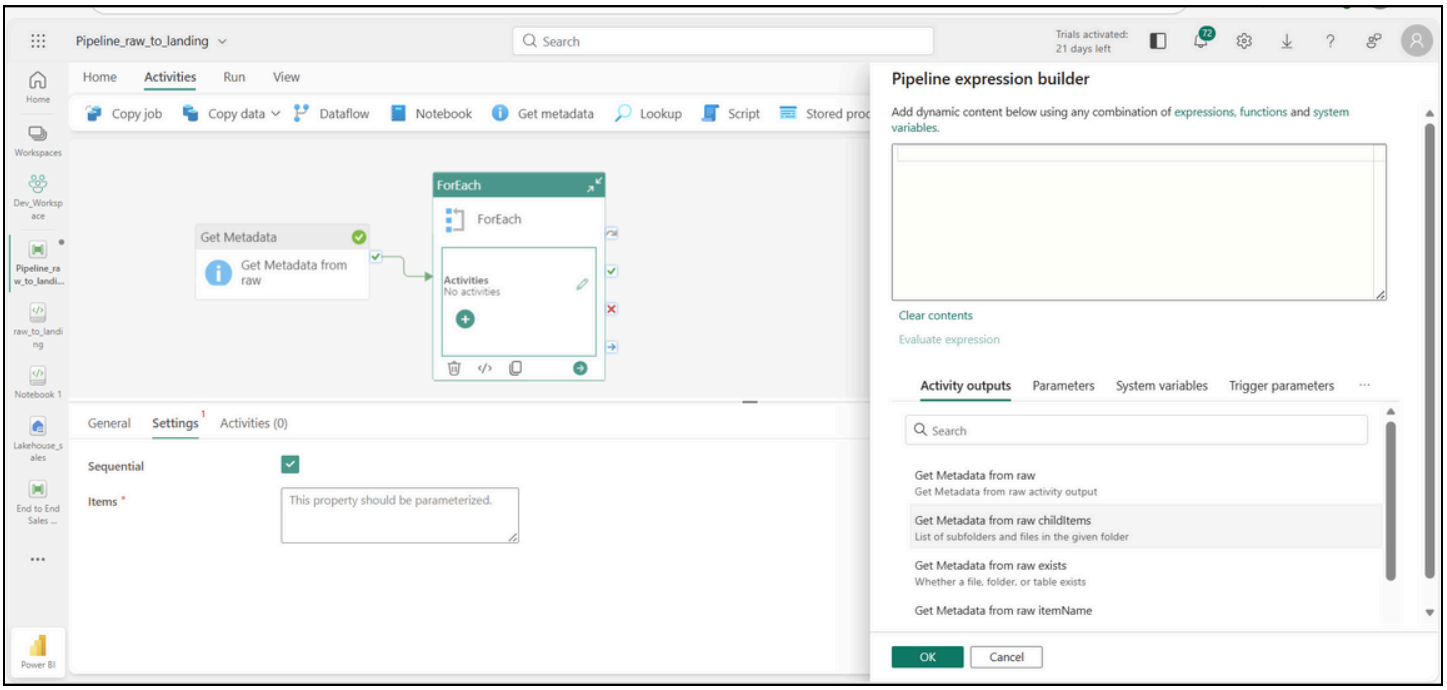


You get the data in output under childitems:

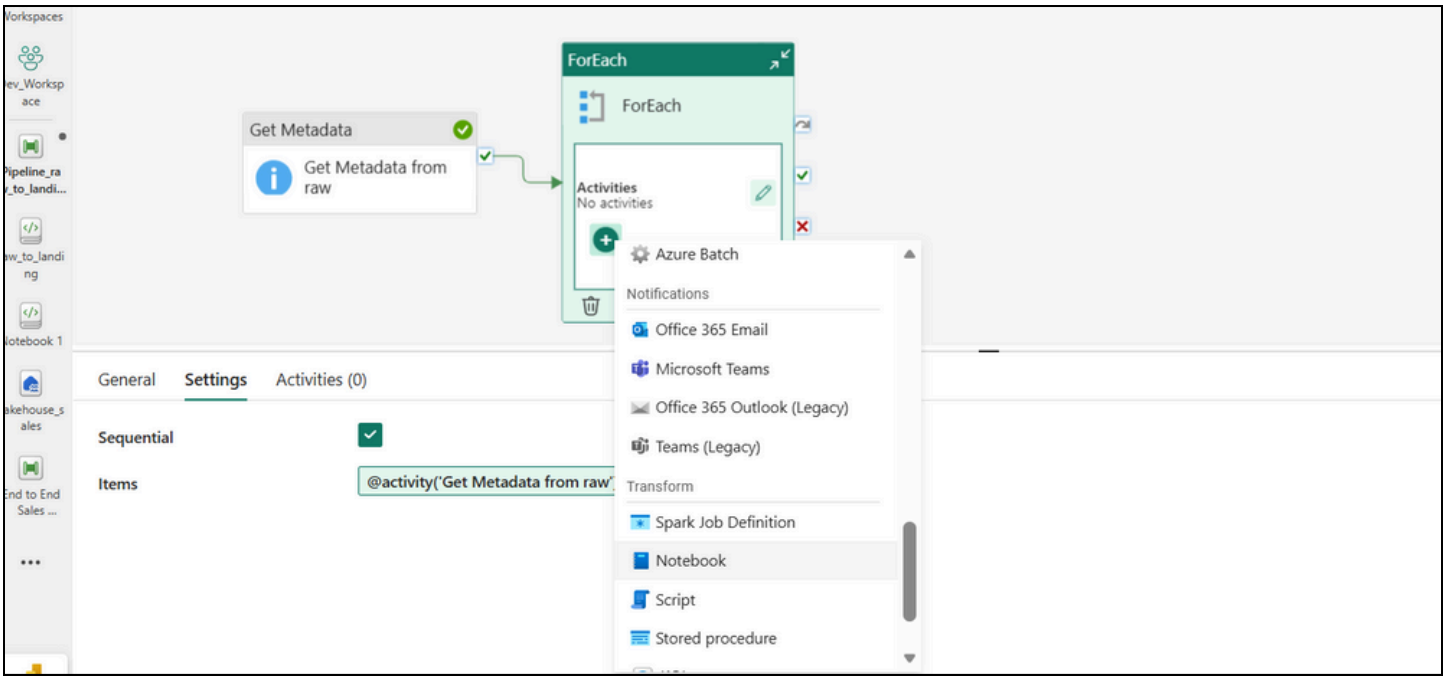


Add Foreach Activity now to process this file through notebook previously created and load data to landing folder:

Get Metadata from childitems of Metadata Activity:

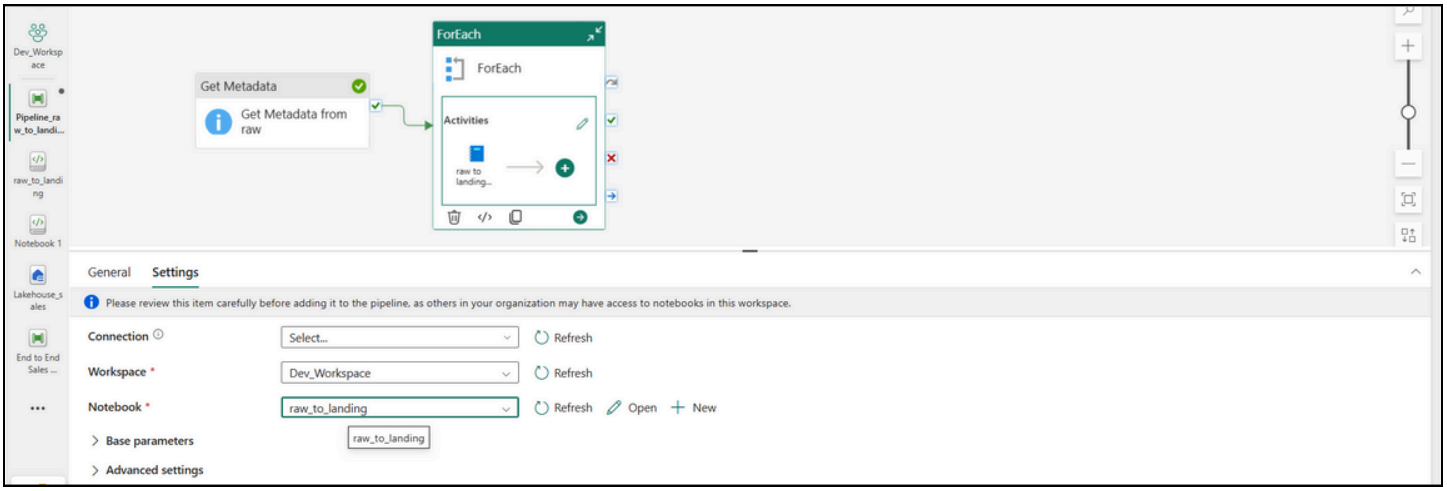


Add Notebook inside this Foreach activity:

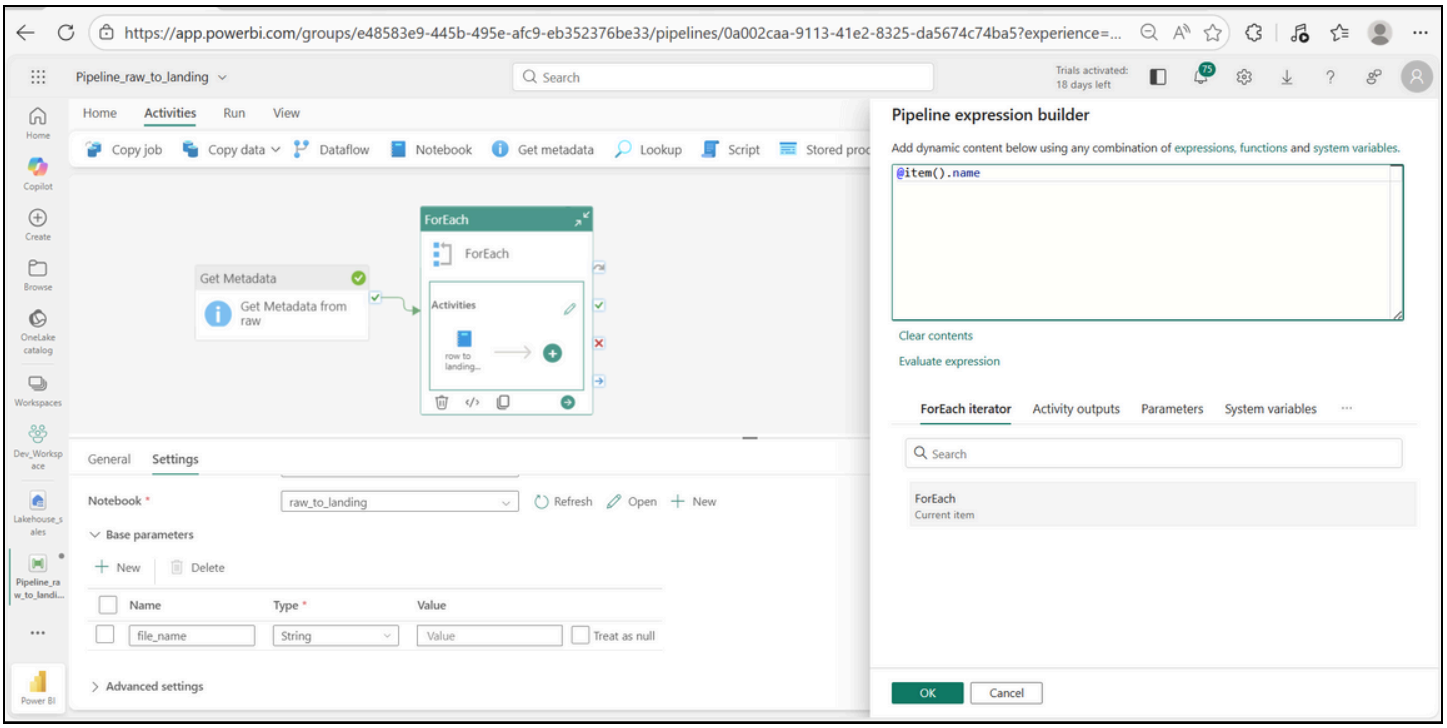


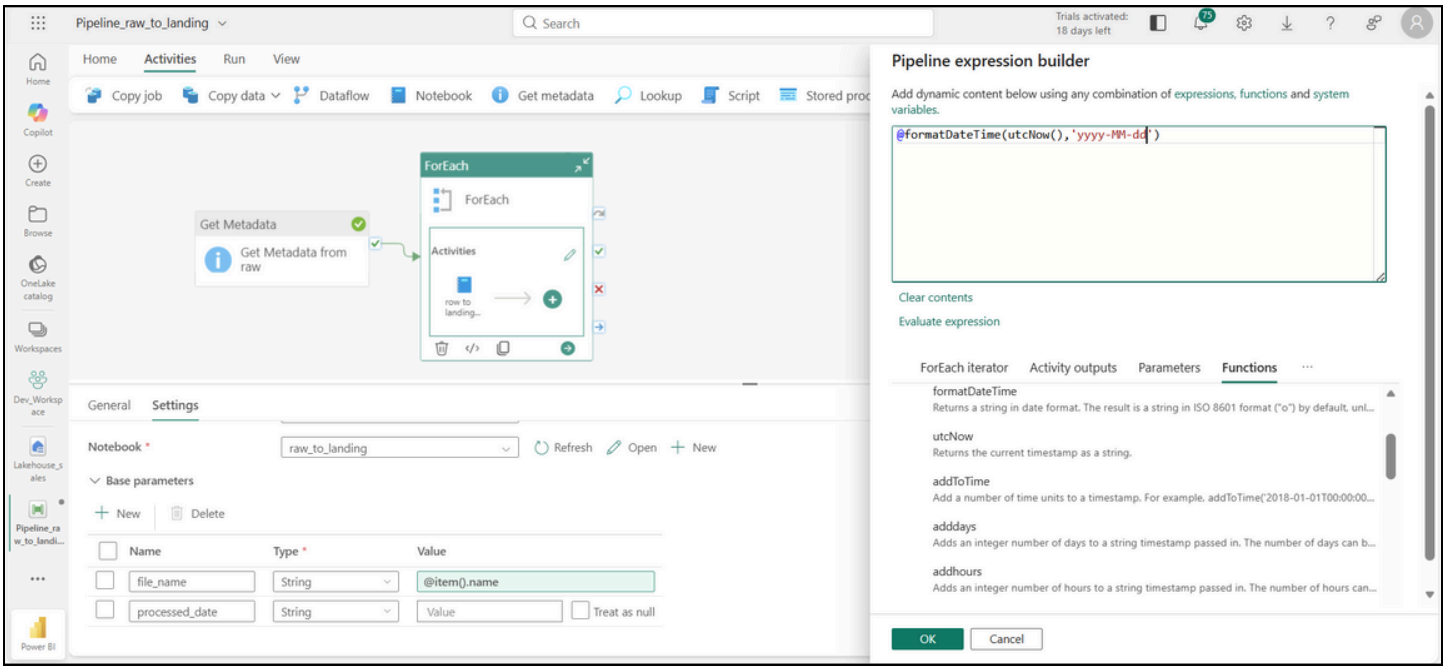
And this notebook will be linked to our raw_to_landing notebook which we have previously created:

Select Workspace and our Notebook:

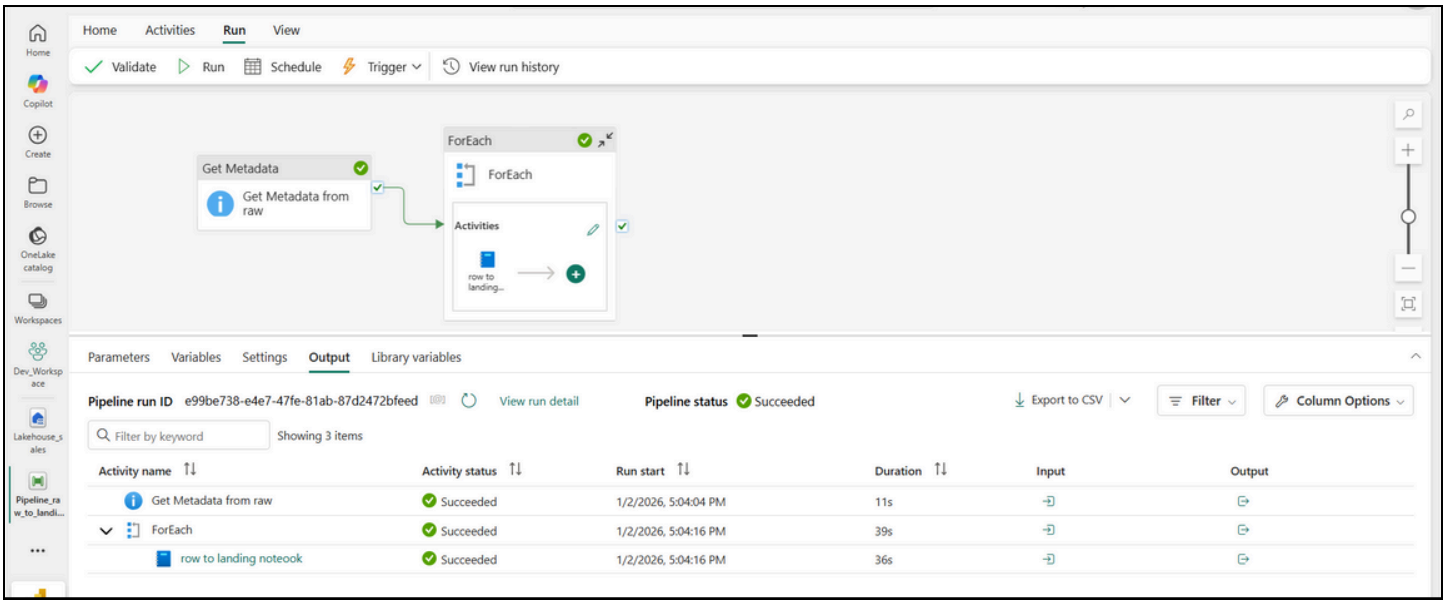


Now for this notebook we have to send parameters date and filename, so under base parameters add these parameters:

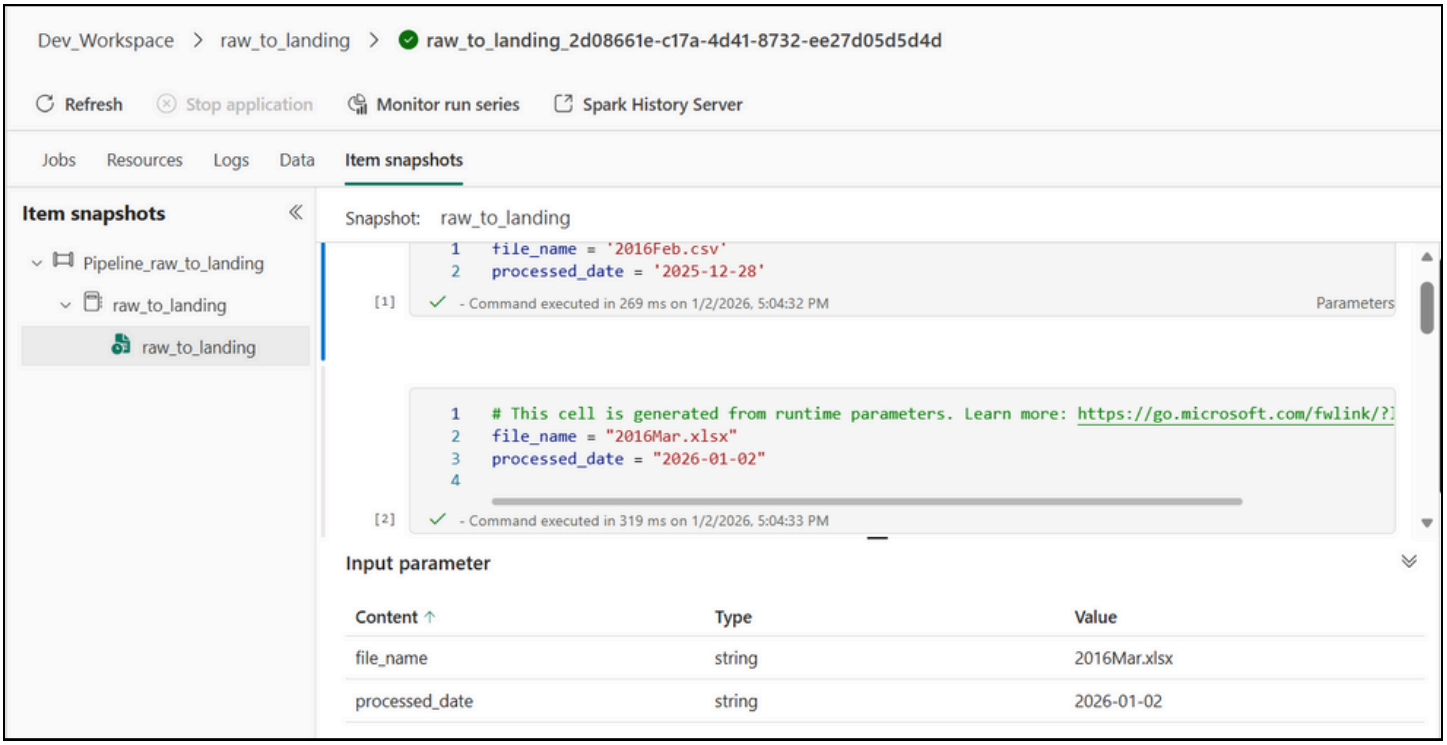


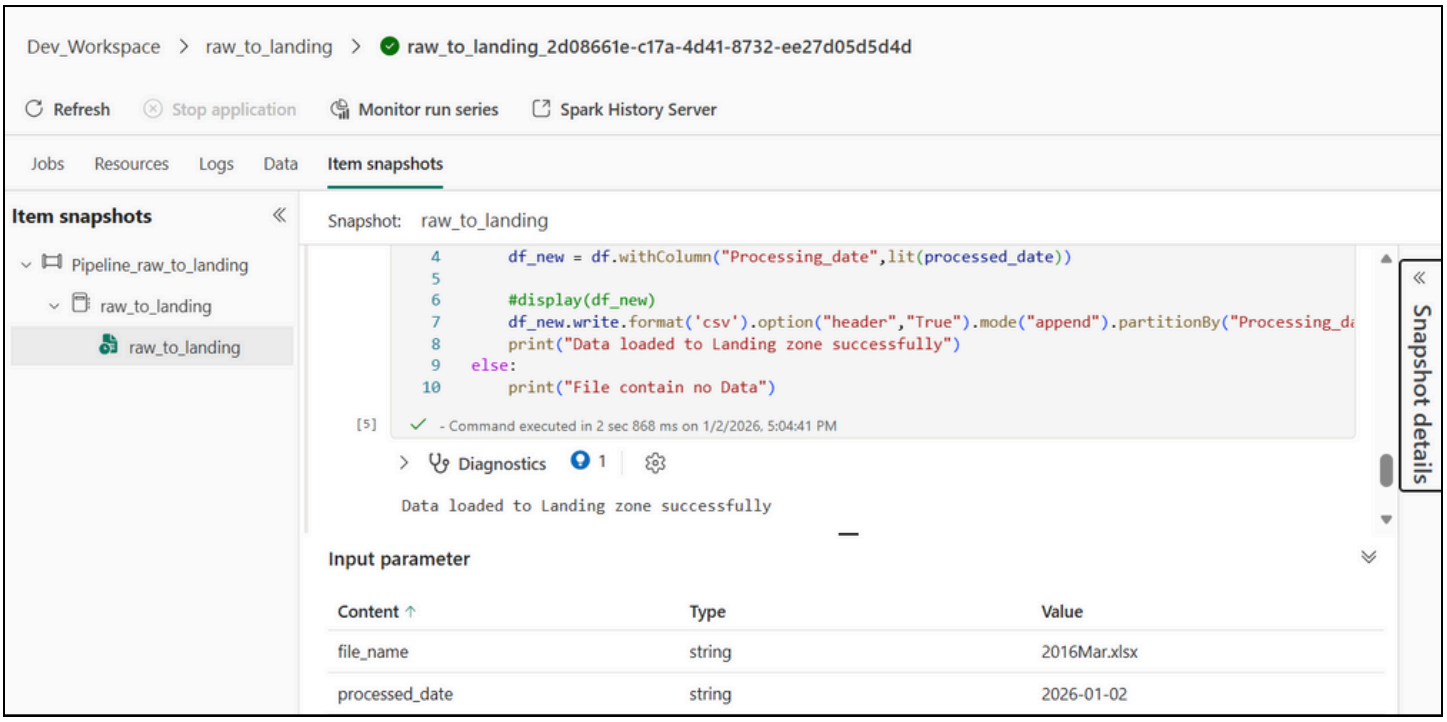


Save and run the pipeline:

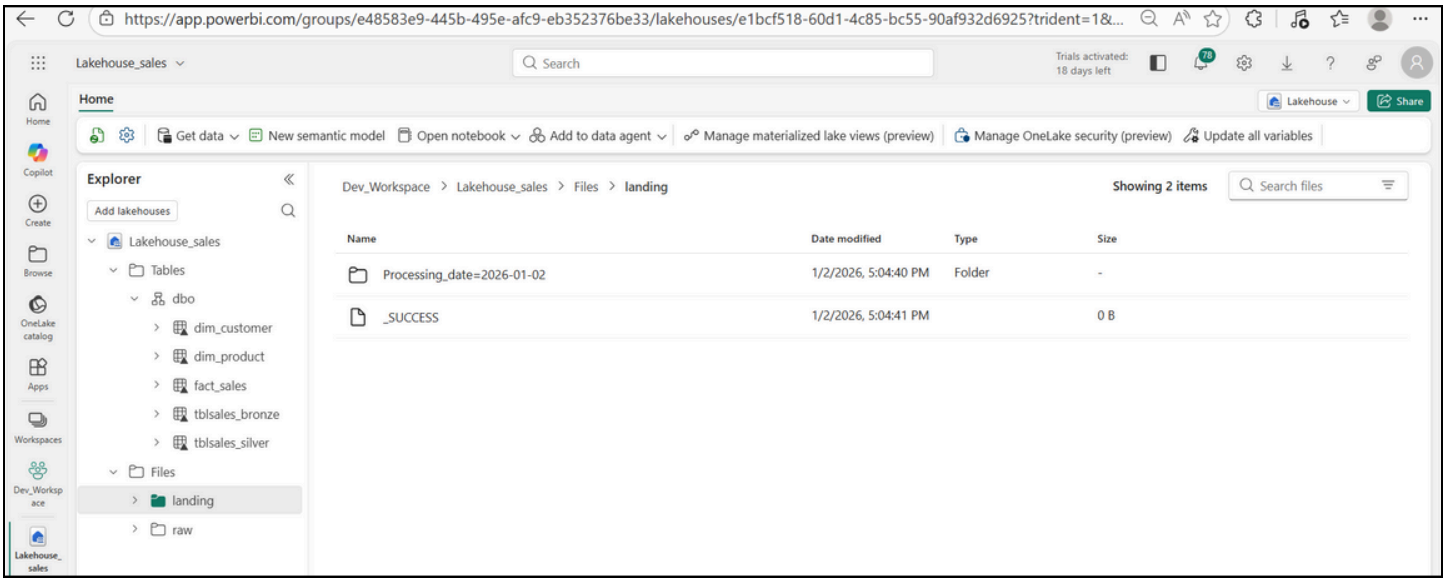


In notebooks snapshot you can see current date:



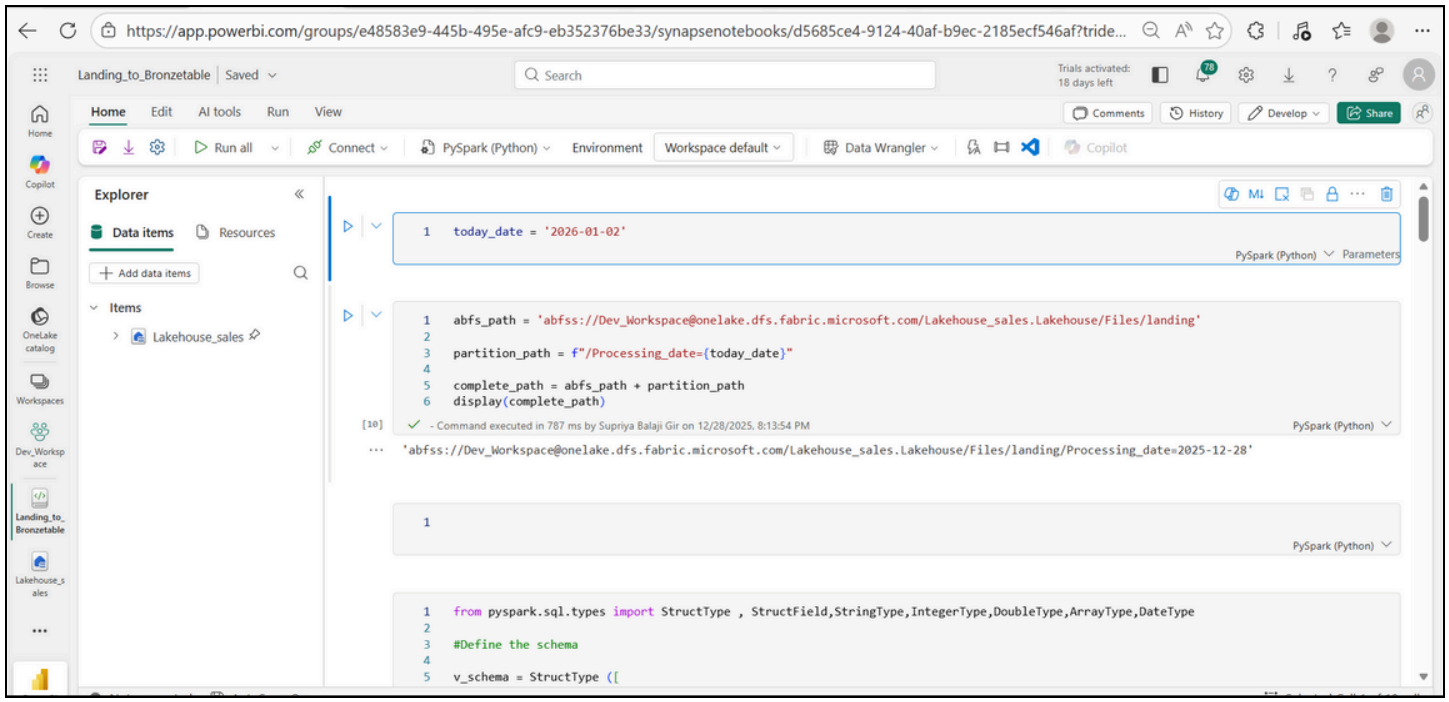


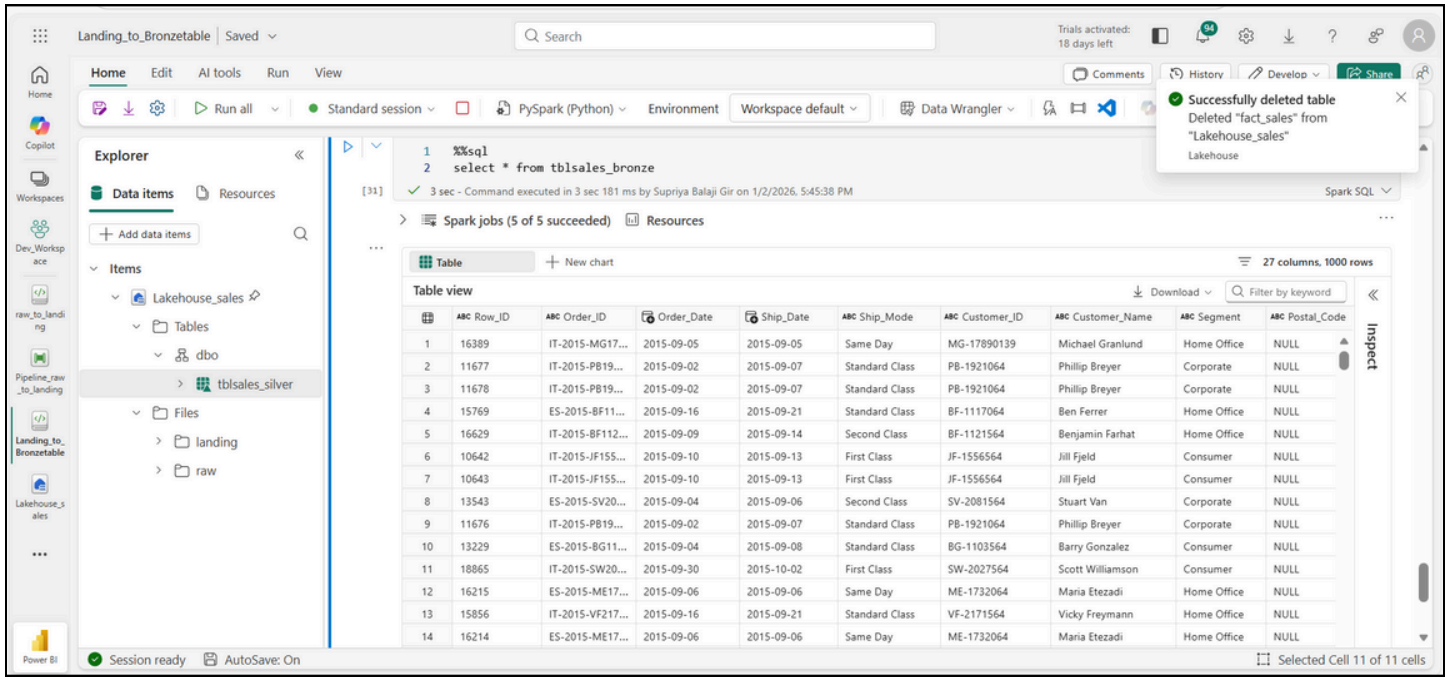
Data got loaded to landing folder:



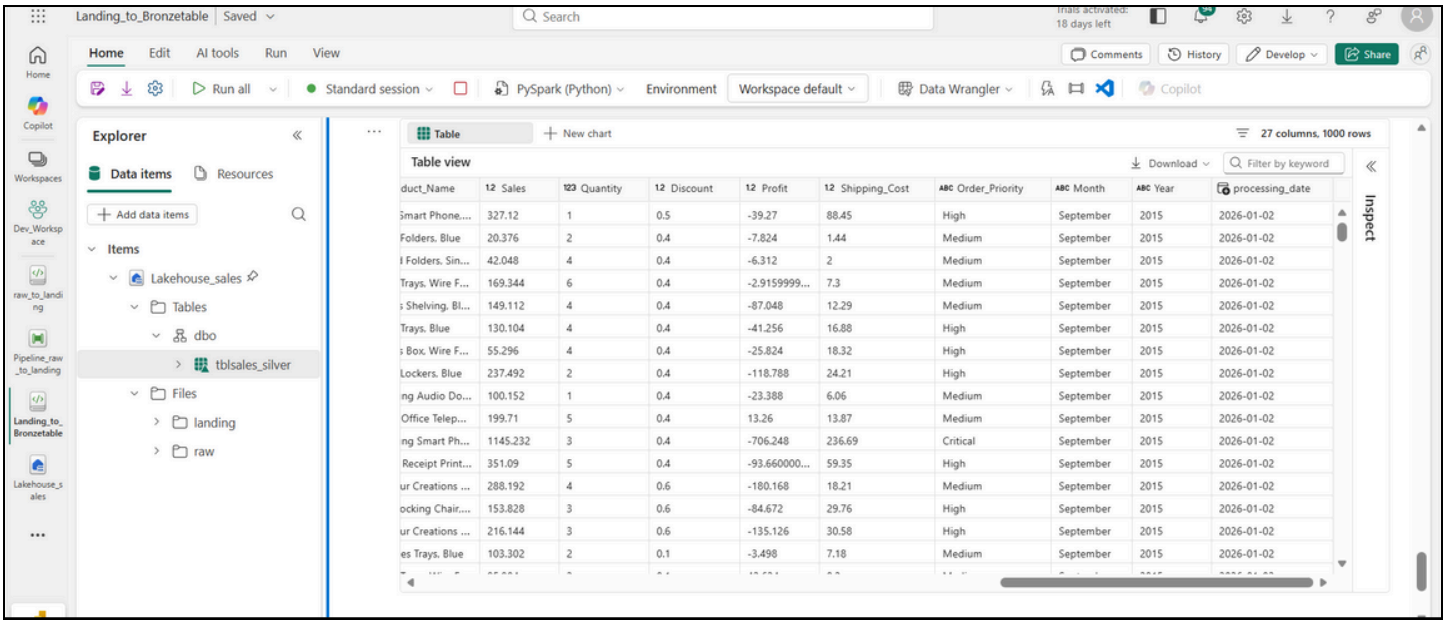
Step 5: Transfer data from landing folder to Bronze layer(table)

Create a new Notebook: Landing to Bronze table notebook:





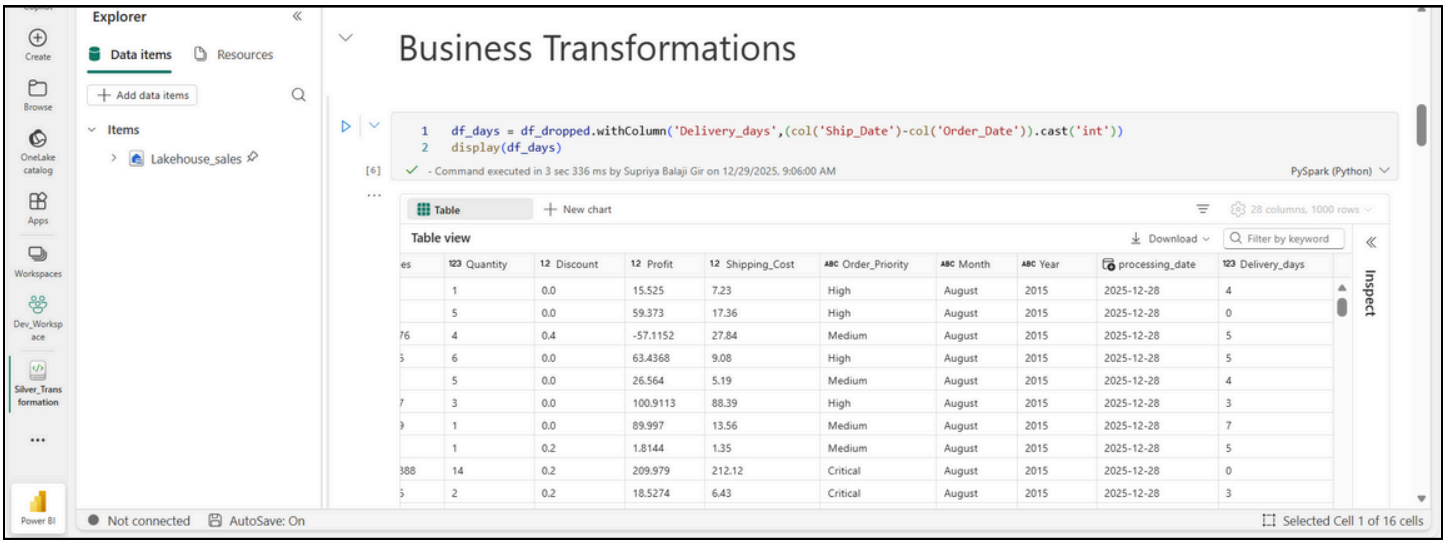
New table got created in bronze layer with additional processing_date column:


Landing_to_Bronzetable
Saved
Search
Trials activated:
18 days left
Home
Edit
AI tools
Run
View
Comments
History
Develop
Share
Run all
Standard session
PySpark (Python)
Environment
Workspace default
Data Wrangler
Copilot
Explorer
Data items
Resources
Add data items
Items
Lakehouse_sales
Tables
dbo
tblsales_silver
Files
landing
raw
Table
New chart
27 columns, 1000 rows
Table view
Download
Filter by keyword
Inspect
duct_Name | Sales | Quantity | Discount | Profit | Shipping_Cost | Order_Priority | Month | Year | processing_date
Smart Phone.... | 327.12 | 1 | 0.5 | -39.27 | 88.45 | High | September | 2015 | 2026-01-02
Folders. Blue | 20.376 | 2 | 0.4 | -7.824 | 1.44 | Medium | September | 2015 | 2026-01-02
Folders. Sin... | 42.048 | 4 | 0.4 | -6.312 | 2 | Medium | September | 2015 | 2026-01-02
Trays. Wire F... | 169.344 | 6 | 0.4 | -2.9159999... | 7.3 | Medium | September | 2015 | 2026-01-02
Shelving, Bl... | 149.112 | 4 | 0.4 | -87.048 | 12.29 | Medium | September | 2015 | 2026-01-02
Trays. Blue | 130.104 | 4 | 0.4 | -41.256 | 16.88 | High | September | 2015 | 2026-01-02
Box. Wire F... | 55.296 | 4 | 0.4 | -25.824 | 18.32 | High | September | 2015 | 2026-01-02
Lockers. Blue | 237.492 | 2 | 0.4 | -118.788 | 24.21 | High | September | 2015 | 2026-01-02
ng Audio Do... | 100.152 | 1 | 0.4 | -23.388 | 6.06 | Medium | September | 2015 | 2026-01-02
Office Telep... | 199.71 | 5 | 0.4 | 13.26 | 13.87 | Medium | September | 2015 | 2026-01-02
ng Smart Ph... | 1145.232 | 3 | 0.4 | -706.248 | 236.69 | Critical | September | 2015 | 2026-01-02
Receipt Print... | 351.09 | 5 | 0.4 | -93.660000... | 59.35 | High | September | 2015 | 2026-01-02
ur Creations ... | 288.192 | 4 | 0.6 | -180.168 | 18.21 | Medium | September | 2015 | 2026-01-02
ocking Chair.... | 153.828 | 3 | 0.6 | -84.672 | 29.76 | High | September | 2015 | 2026-01-02
ur Creations ... | 216.144 | 3 | 0.6 | -135.126 | 30.58 | High | September | 2015 | 2026-01-02
es Trays. Blue | 103.302 | 2 | 0.1 | -3.498 | 7.18 | Medium | September | 2015 | 2026-01-02

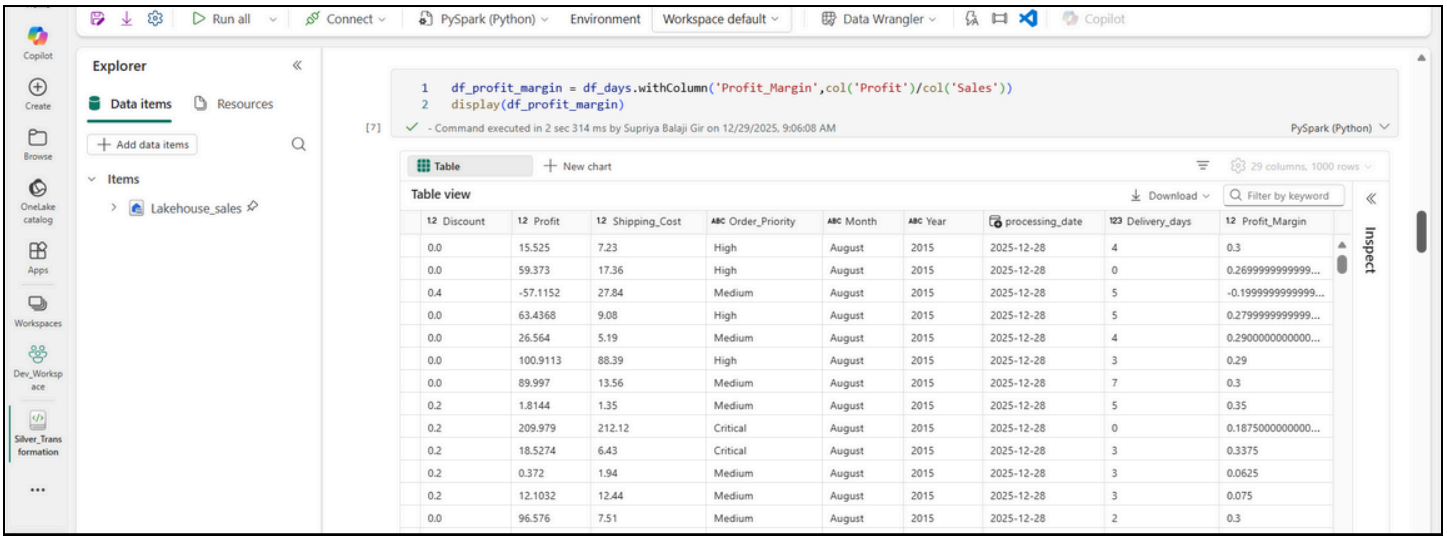
Step 6: Transfer data from Bronze layer to Silver layer(table)

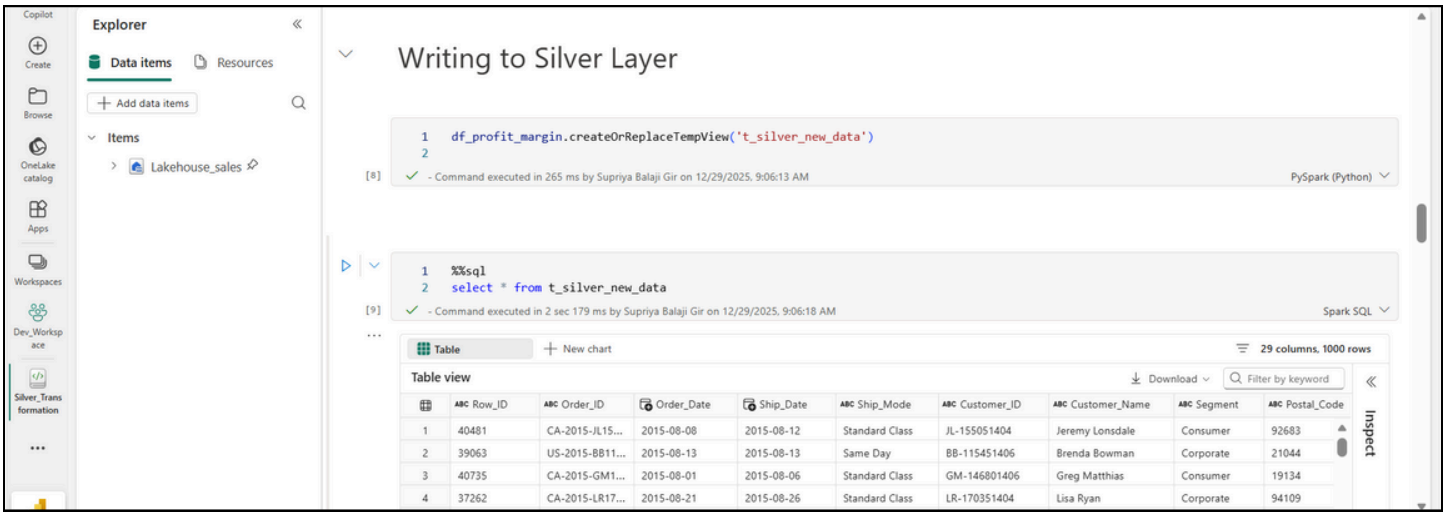
In this transformation the cleaning part is done, some business-related transformations are done:

added Delivery Date column with the help of ship_date and order_Date:



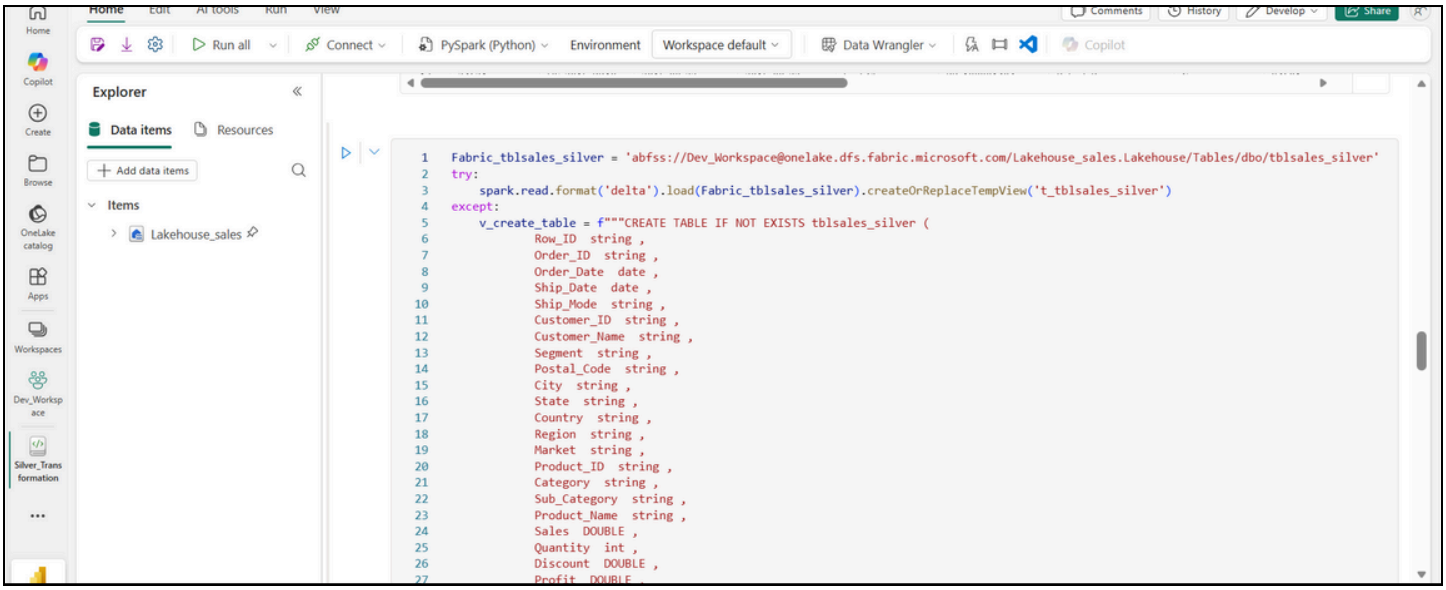
add Profit margin column:





Create table in Silver layer:

Load data:

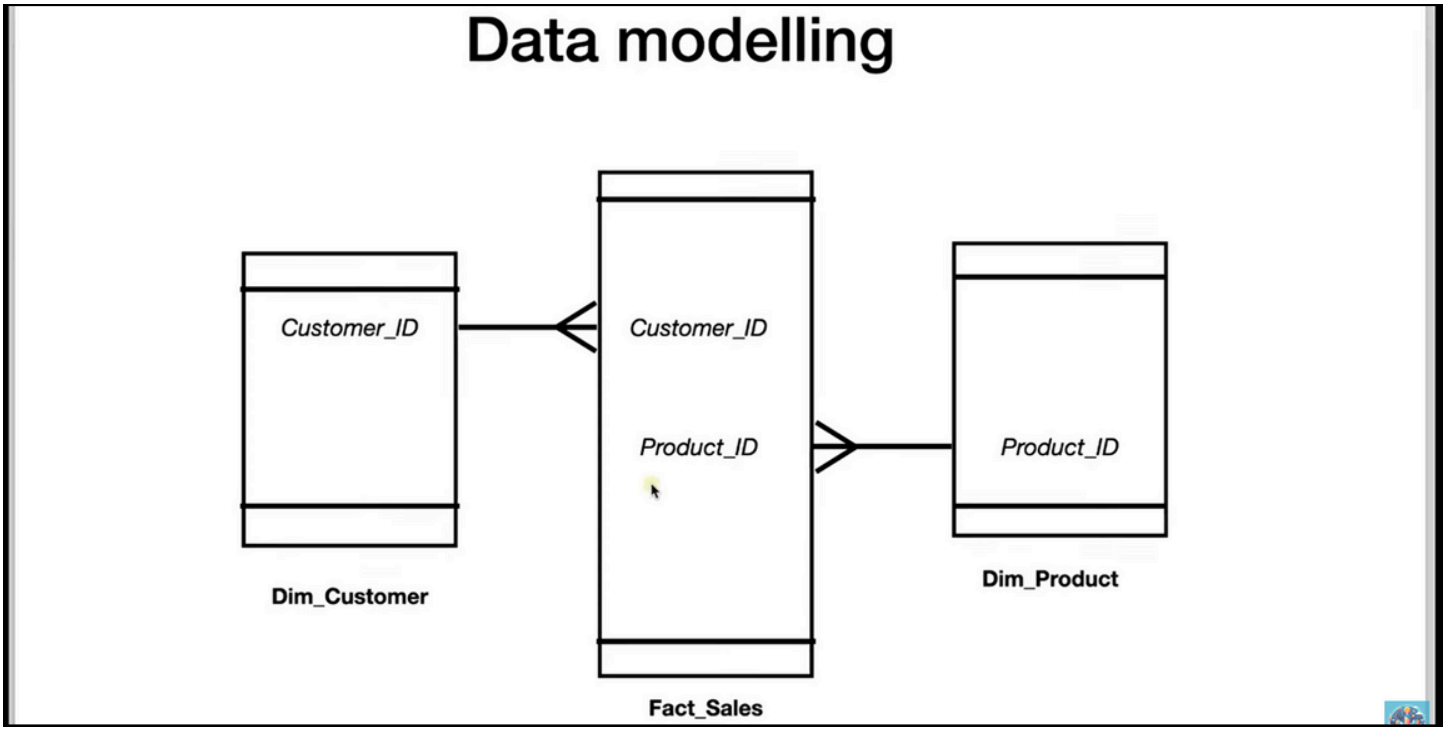


Silver table is seen now in lakehouse:

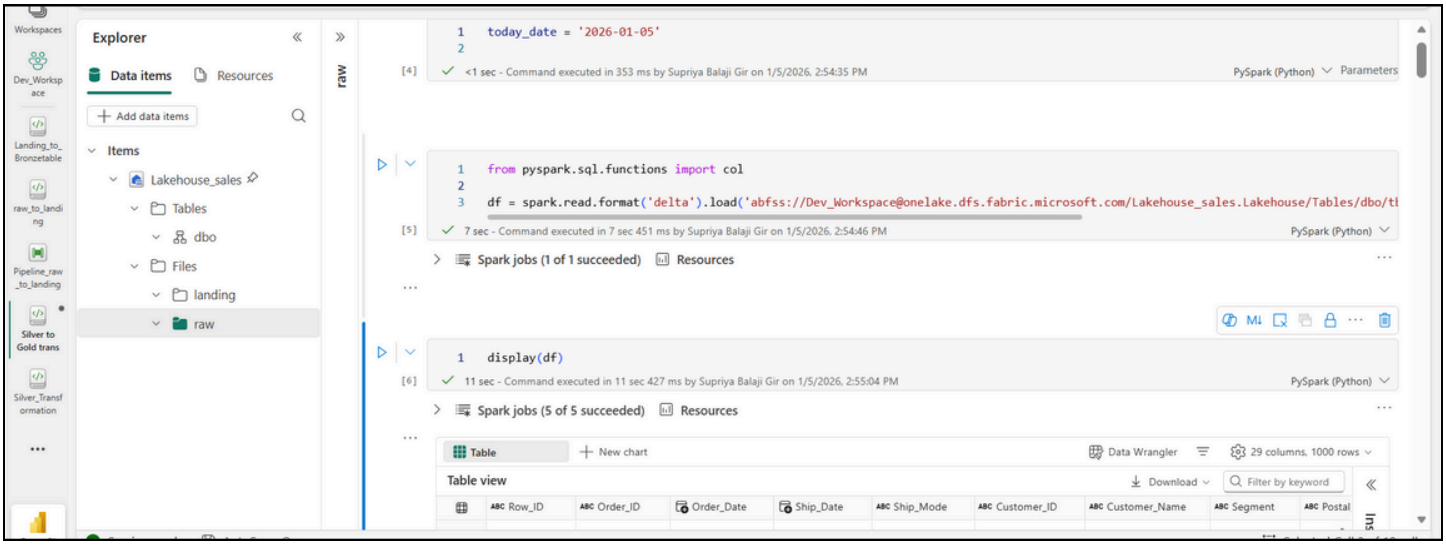
At the end of Silver transformation data is cleaned and transformed. But the data is yet not ready for business consumption.

Step 7: Transfer data from Silver layer to Gold layer(table)

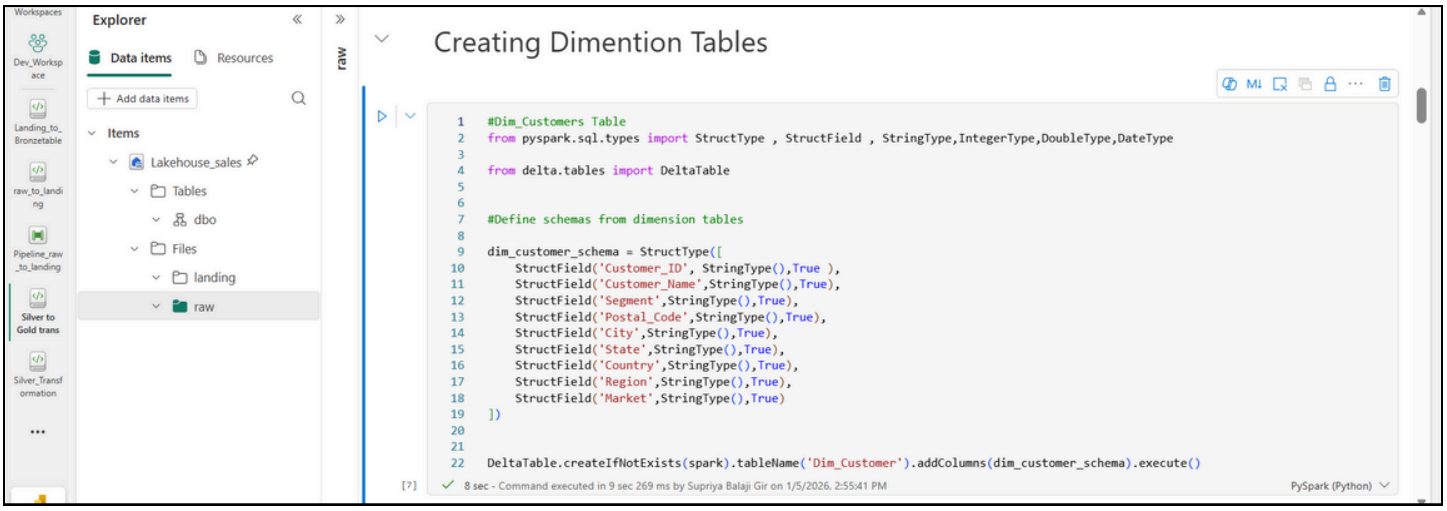
Make data usable for Business by doing data modeling:



Creating new Notebook for Silver to gold transformation and attach lakehouse to it:



Creating dimention Tables:

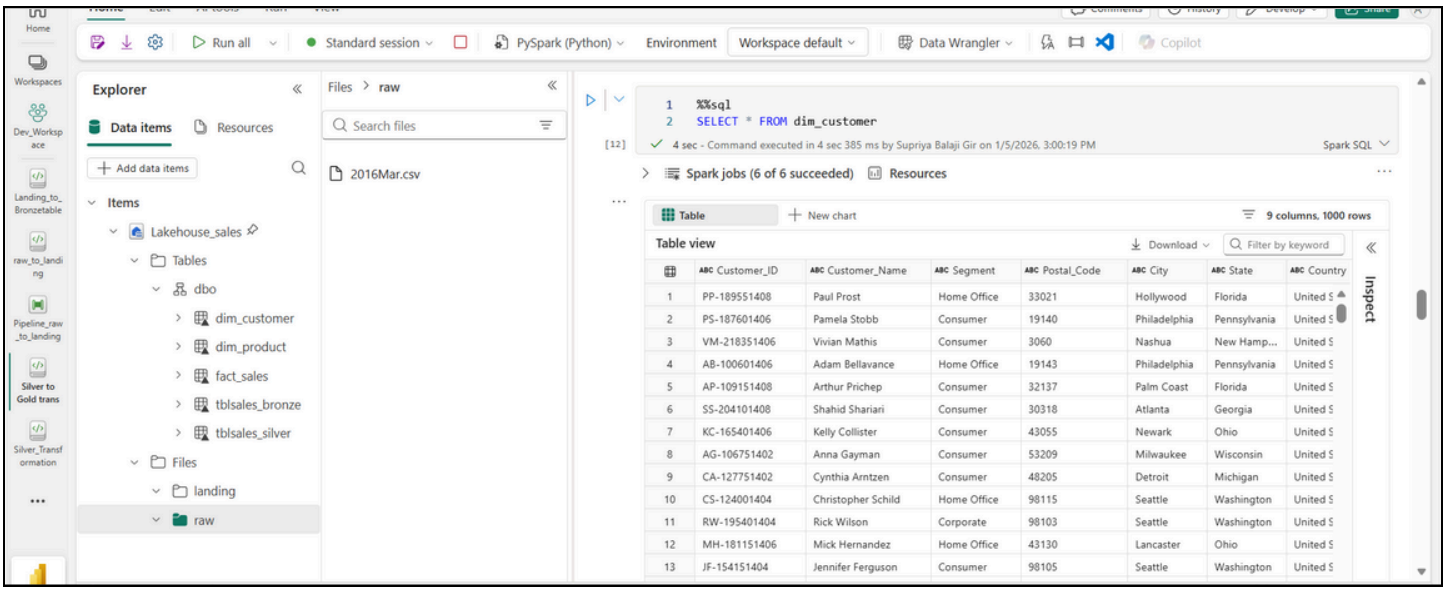


Creation of fact Table:

insert data into newly created tables from table in silver transformation:

```
from delta.tables import *

dim_customer_table_path = 'abfss://Dev_Workspace@onelake.dfs.fabric.microsoft.com/Lakehouse_sales.Lakehouse/Tables/dbo/dim_customer
dim_deltacustomer = DeltaTable.forPath(spark,dim_customer_table_path)

## Perform the MERGE (UPSERT)

dim_deltacustomer.alias('target').merge(
df_selected_dim_customer.alias('source'),'target.Customer_ID=source.Customer_ID'
).whenMatchedUpdate( set = {
            'Customer_Name':'source.Customer_Name',
            'Segment':'source.Segment',
            'Postal_Code':'source.Postal_Code',
            'City':'source.City',
            'State':'source.State',
            'Country':'source.Country',
            'Region':'source.Region',
            'Market':'source.Market'

}).whenNotMatchedInsert(values={
            'Customer_ID':'source.Customer_ID',
            'Customer_Name':'source.Customer_Name',
            'Segment':'source.Segment',
            'Postal_Code':'source.Postal_Code',
            'City':'source.City',
            'State':'source.State',
            'Country':'source.Country',
            'Region':'source.Region',
            'Market':'source.Market'
```

Customer Table:



```
%%sql
SELECT * FROM dim_customer
```

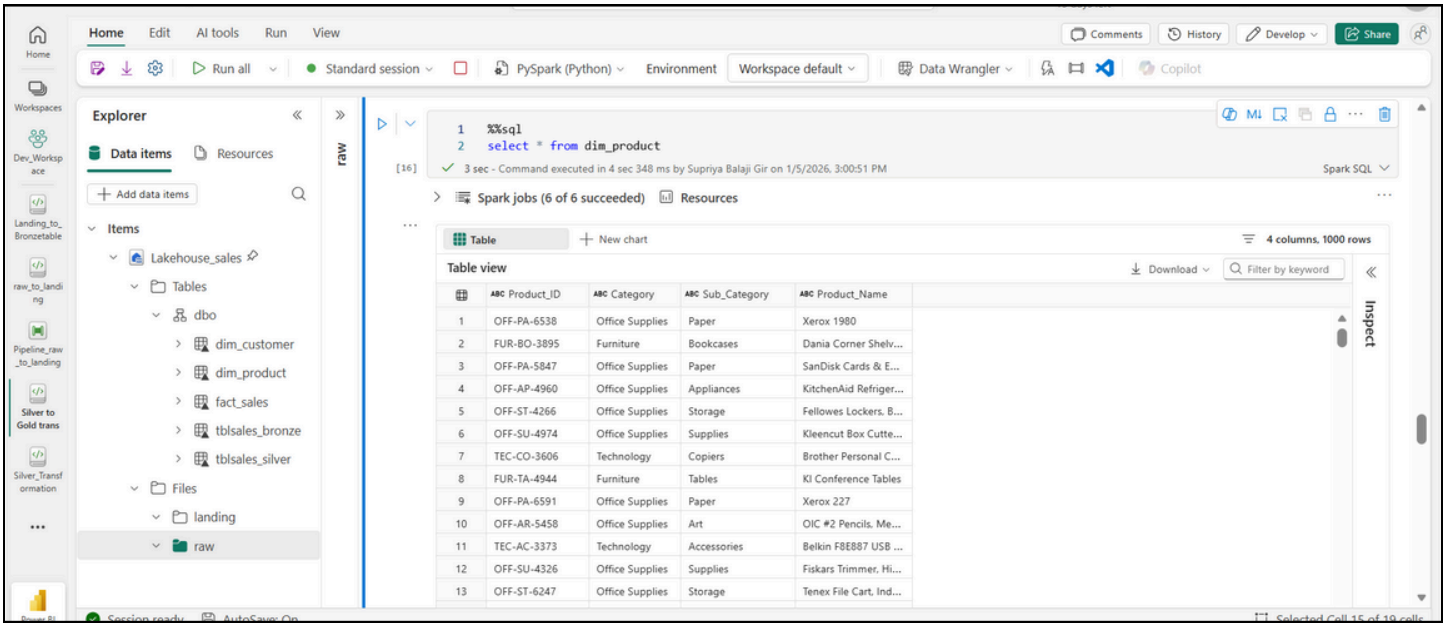
4 sec - Command executed in 4 sec 385 ms by Supriya Balaji Gir on 1/5/2026, 3:00:19 PM

Spark jobs (6 of 6 succeeded) Resources

Table view — 9 columns, 1000 rows

	Customer_ID	Customer_Name	Segment	Postal_Code	City	State	Country
1	PP-189551408	Paul Prost	Home Office	33021	Hollywood	Florida	United S
2	PS-187601406	Pamela Stobb	Consumer	19140	Philadelphia	Pennsylvania	United S
3	VM-218351406	Vivian Mathis	Consumer	3060	Nashua	New Hamp...	United S
4	AB-100601406	Adam Bellavance	Home Office	19143	Philadelphia	Pennsylvania	United S
5	AP-109151408	Arthur Prichep	Consumer	32137	Palm Coast	Florida	United S
6	SS-204101408	Shahid Shariari	Consumer	30318	Atlanta	Georgia	United S
7	KC-165401406	Kelly Collister	Consumer	43055	Newark	Ohio	United S
8	AG-106751402	Anna Gayman	Consumer	53209	Milwaukee	Wisconsin	United S
9	CA-127751402	Cynthia Arntzen	Consumer	48205	Detroit	Michigan	United S
10	CS-124001404	Christopher Schild	Home Office	98115	Seattle	Washington	United S
11	RW-195401404	Rick Wilson	Corporate	98103	Seattle	Washington	United S
12	MH-181151406	Mick Hernandez	Home Office	43130	Lancaster	Ohio	United S
13	JF-154151404	Jennifer Ferguson	Consumer	98105	Seattle	Washington	United S

Product table:



```
%%sql
select * from dim_product
```

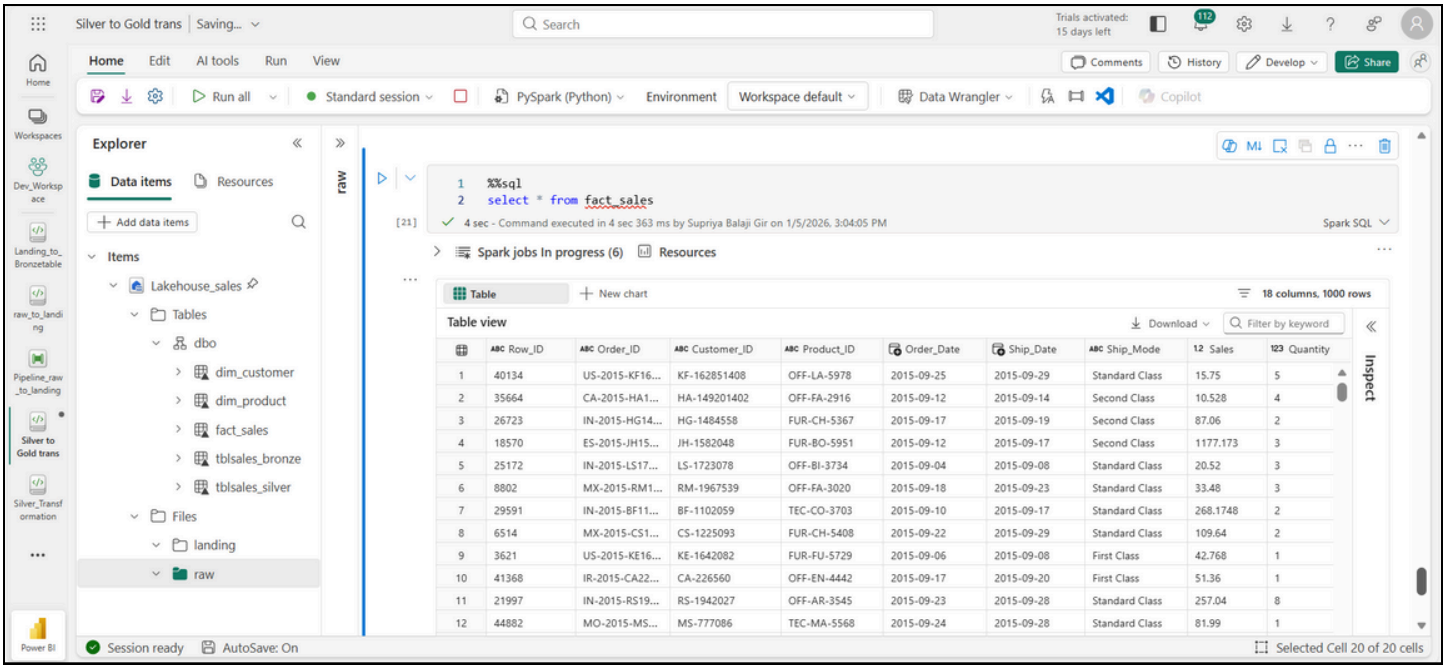
3 sec - Command executed in 4 sec 348 ms by Supriya Balaji Gir on 1/5/2026, 3:00:51 PM

Spark jobs (6 of 6 succeeded) Resources

Table view — 4 columns, 1000 rows

	Product_ID	Category	Sub_Category	Product_Name
1	OFF-PA-6538	Office Supplies	Paper	Xerox 1980
2	FUR-BO-3895	Furniture	Bookcases	Dania Corner Shelv...
3	OFF-PA-5847	Office Supplies	Paper	SanDisk Cards & E...
4	OFF-AP-4960	Office Supplies	Appliances	KitchenAid Refriger...
5	OFF-ST-4266	Office Supplies	Storage	Fellowes Lockers, B...
6	OFF-SU-4974	Office Supplies	Supplies	Kleencut Box Cutte...
7	TEC-CO-3606	Technology	Copiers	Brother Personal C...
8	FUR-TA-4944	Furniture	Tables	KI Conference Tables
9	OFF-PA-6591	Office Supplies	Paper	Xerox 227
10	OFF-AR-5458	Office Supplies	Art	OIC #2 Pencils, Me...
11	TEC-AC-3373	Technology	Accessories	Belkin F8E887 USB ...
12	OFF-SU-4326	Office Supplies	Supplies	Fiskars Trimmer, Hi...
13	OFF-ST-6247	Office Supplies	Storage	Tenex File Cart, Ind...

Fact_Sales table:

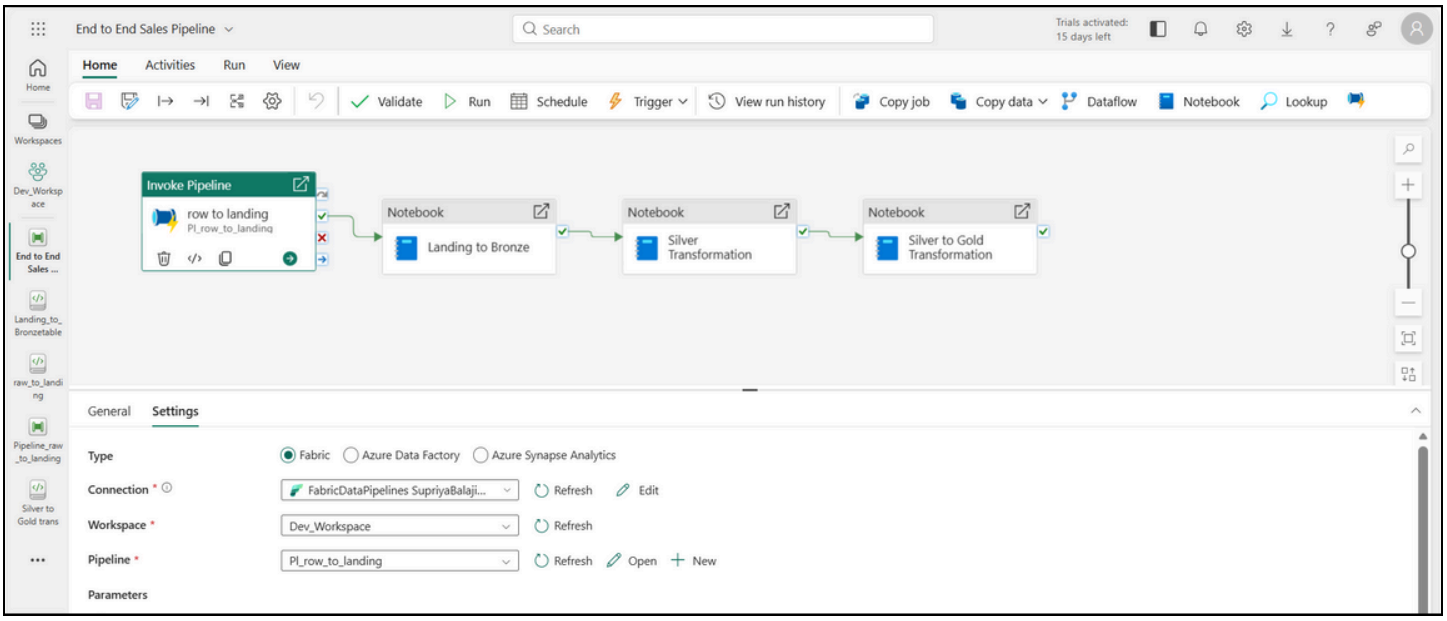


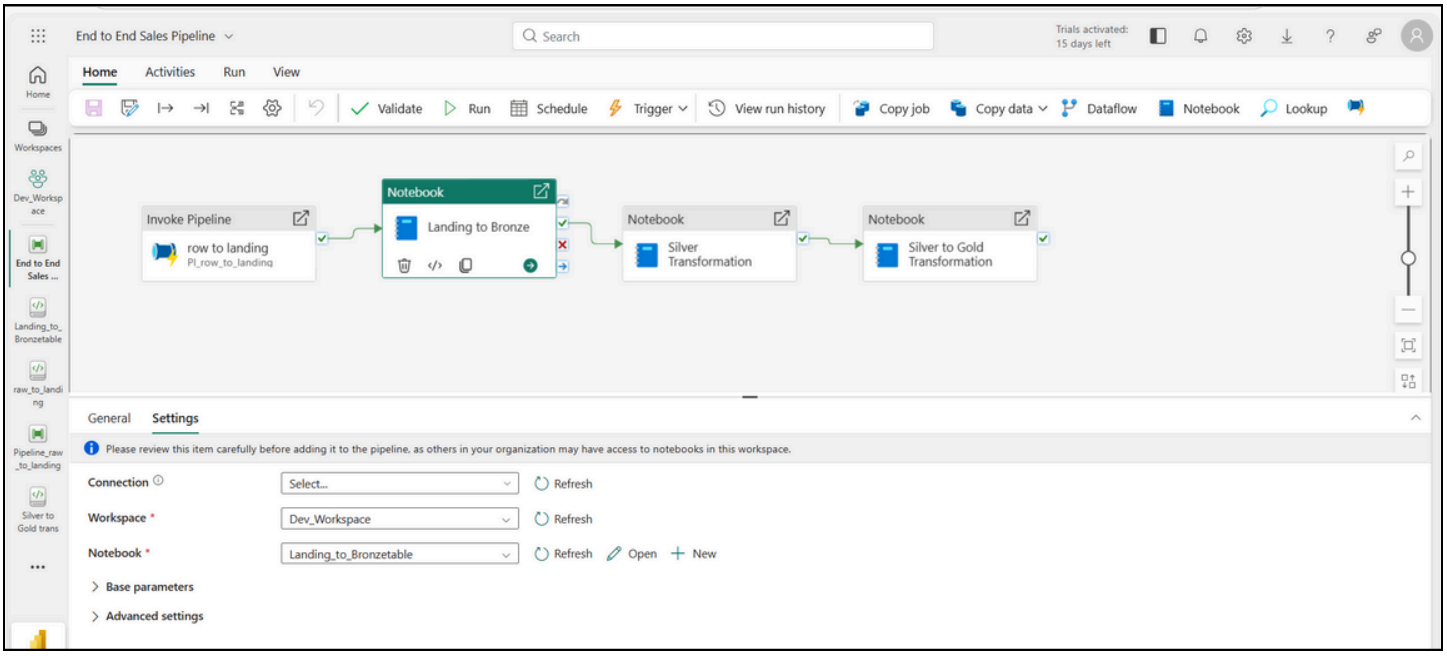
Step 8: Creating End to End Pipeline to transfer data from raw folder to Golden layer

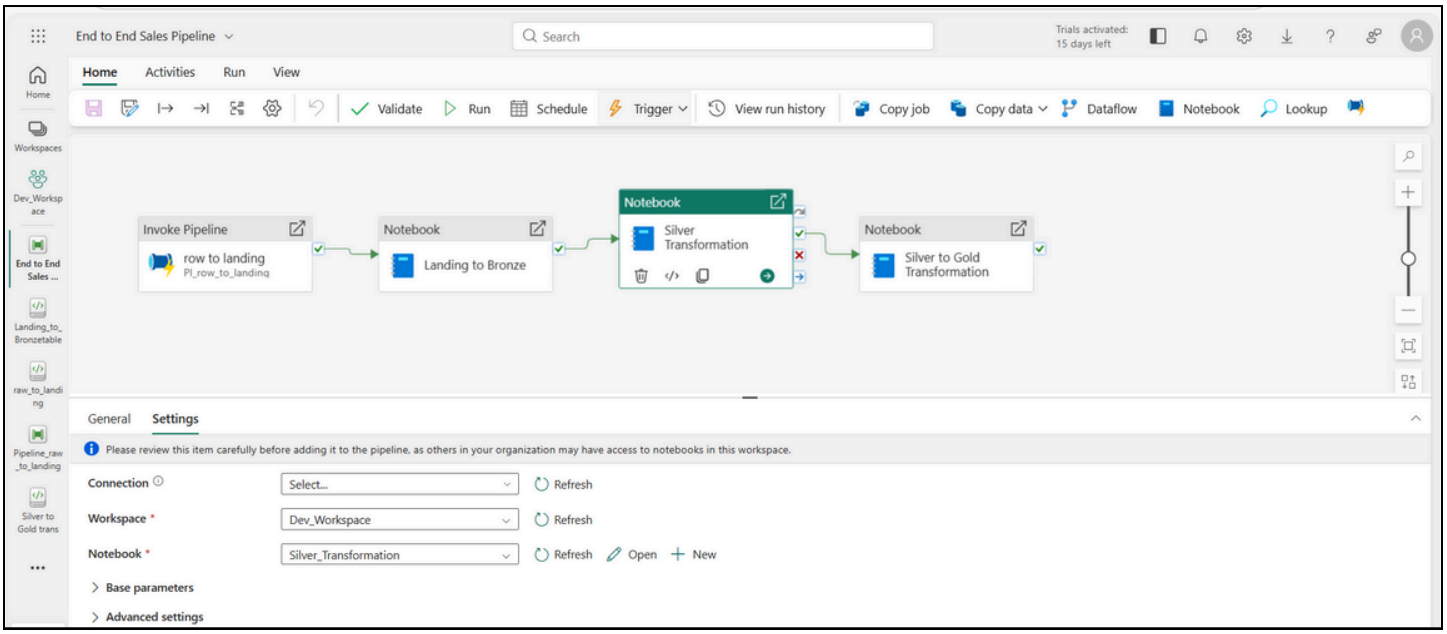
Now creating again a full end-to-end pipeline for data movement from raw folder till Gold level:

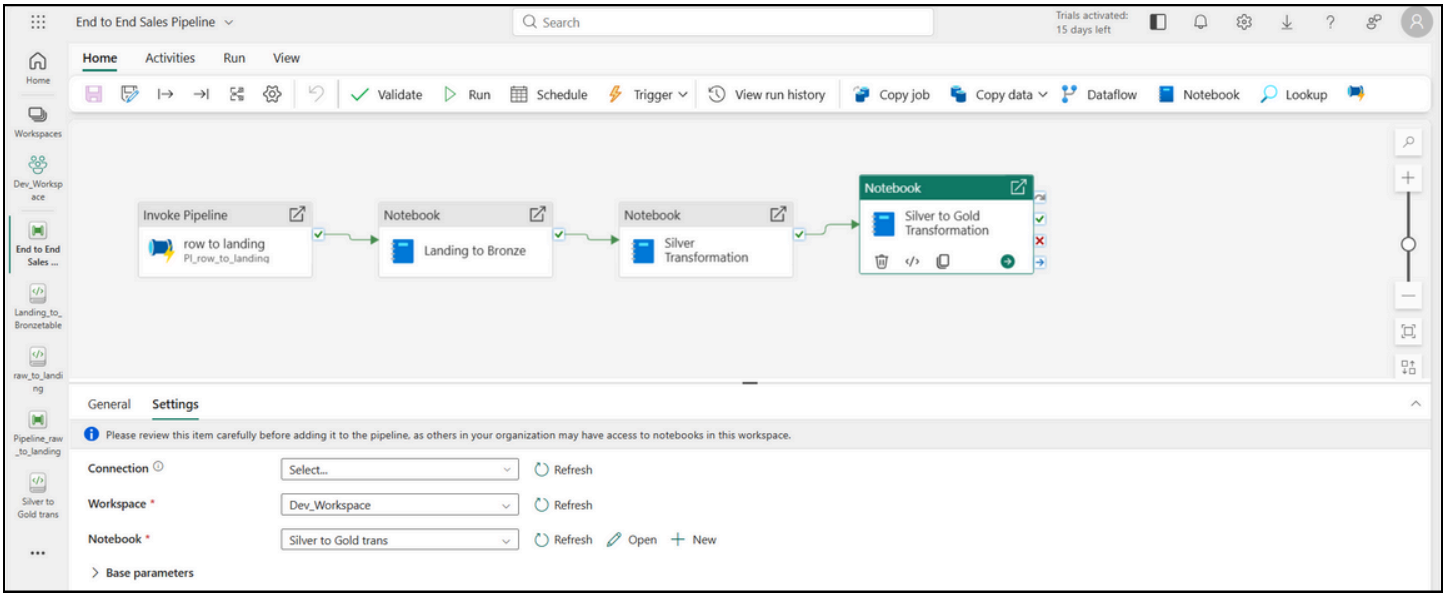
Already the raw to landing pipeline is created so first in new pipeline we will invoke that pipeline as first activity of end-to-end pipeline:

And then attach 3 further notebooks as next activities:









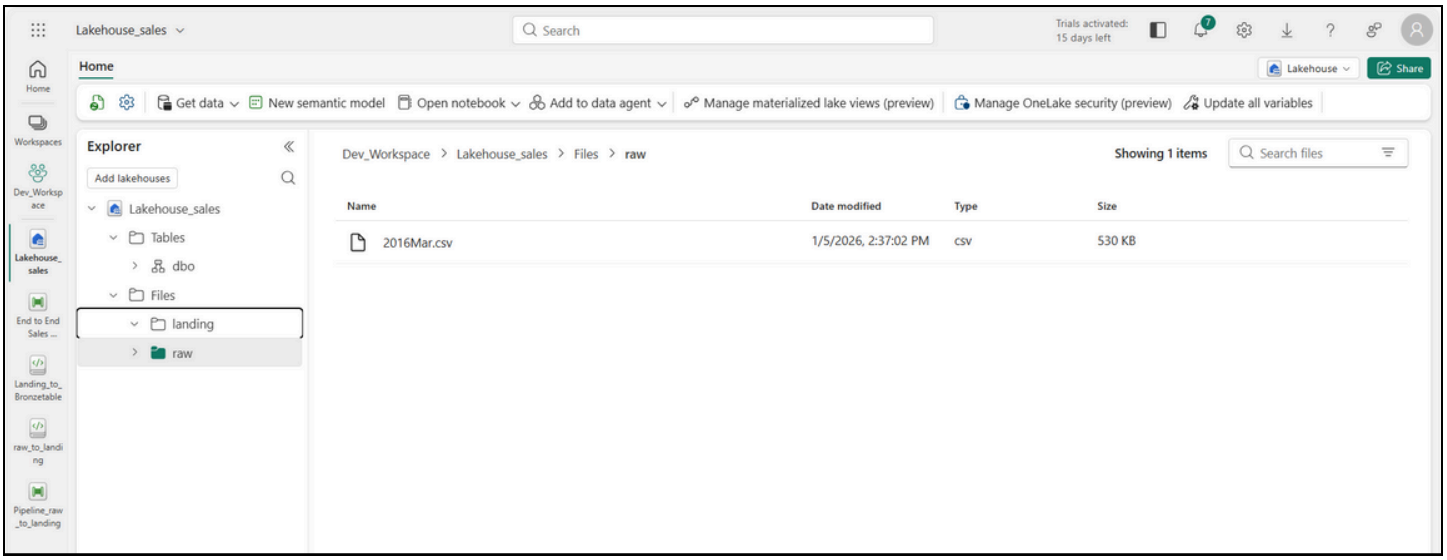
Testing Final Pipeline:

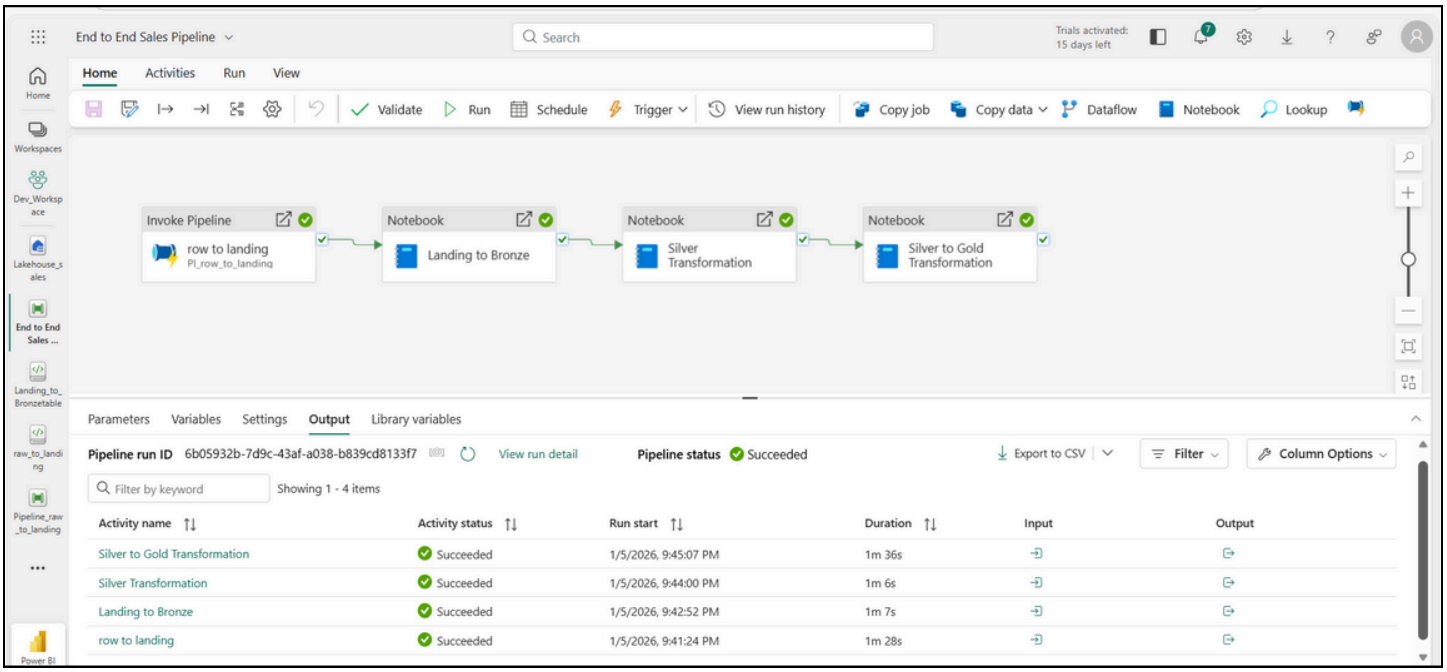
In lakehouse file 2016Mar csv is added. All other folders and tables are not created.

Landing folder is empty.

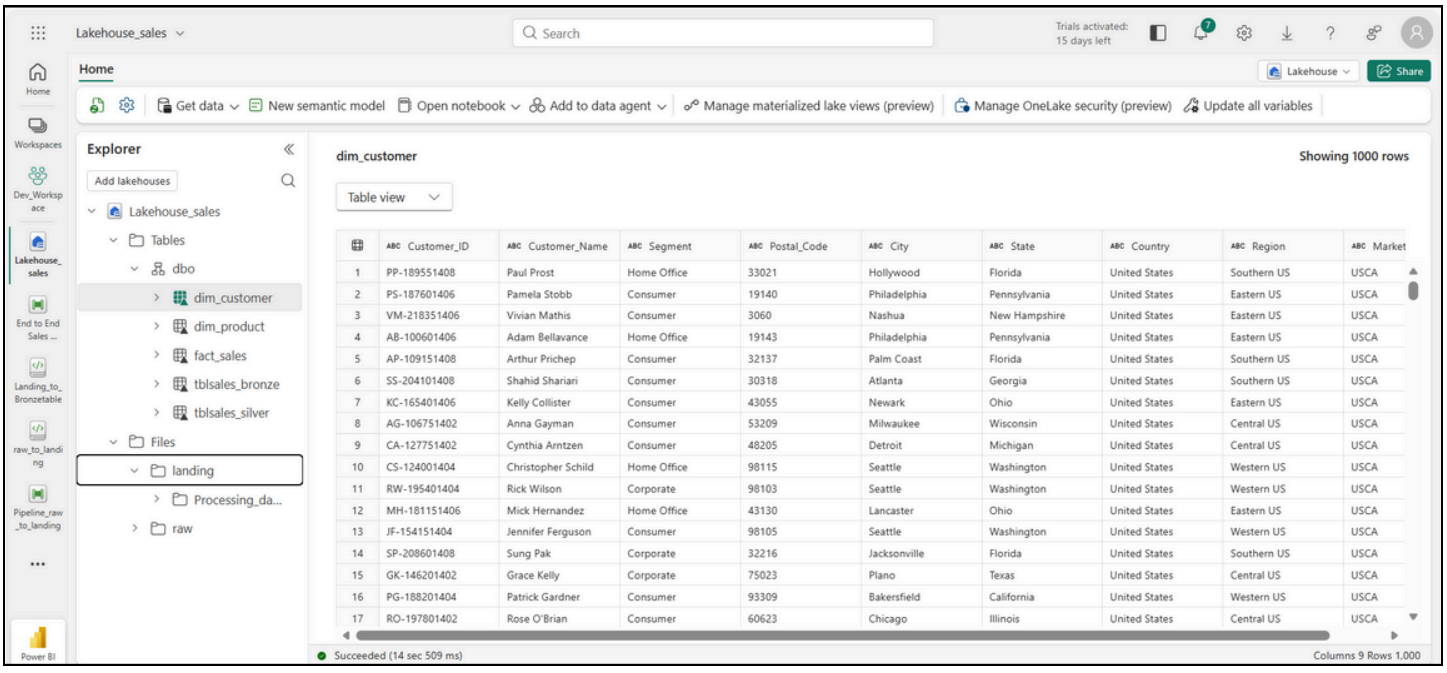
After running end-to-end pipeline, data file should move from:

1. raw to landing folder
2. from landing folder to bronze table
3. from bronze table to silver table
4. at last from silver table to gold table with creation of fact and dimention tables

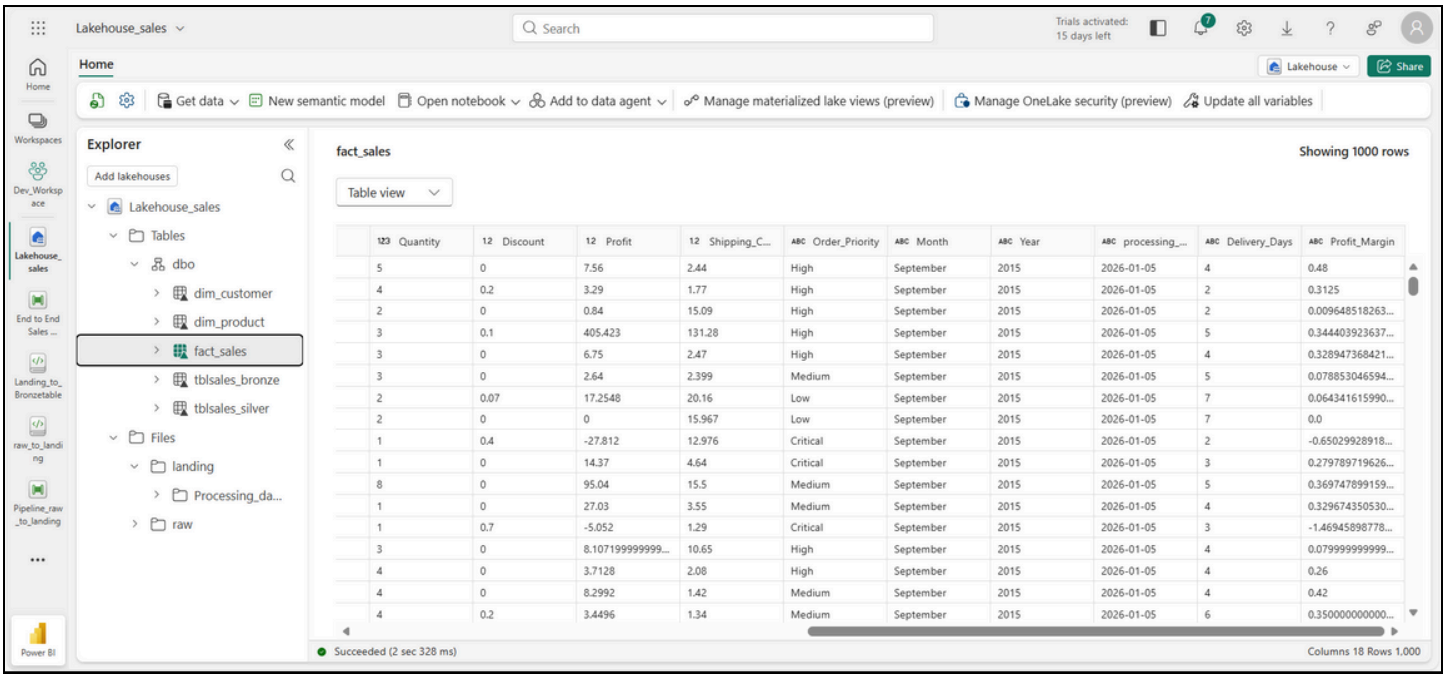




result:



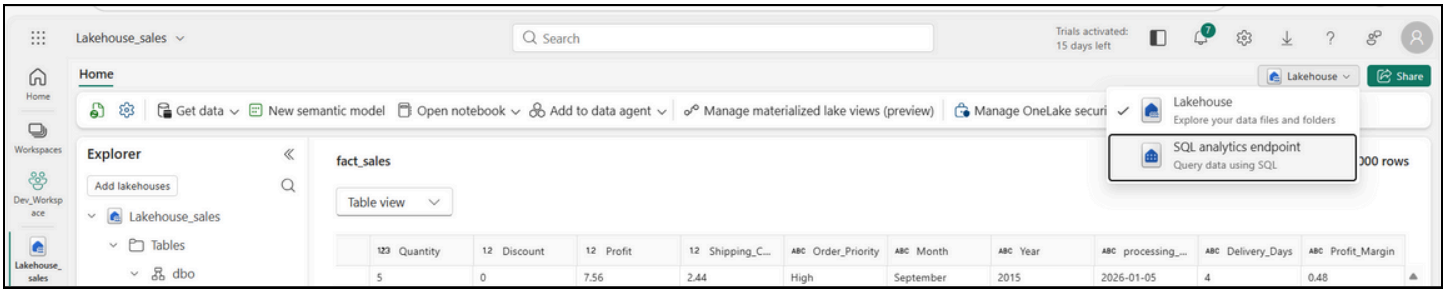
sales table with all new columns:



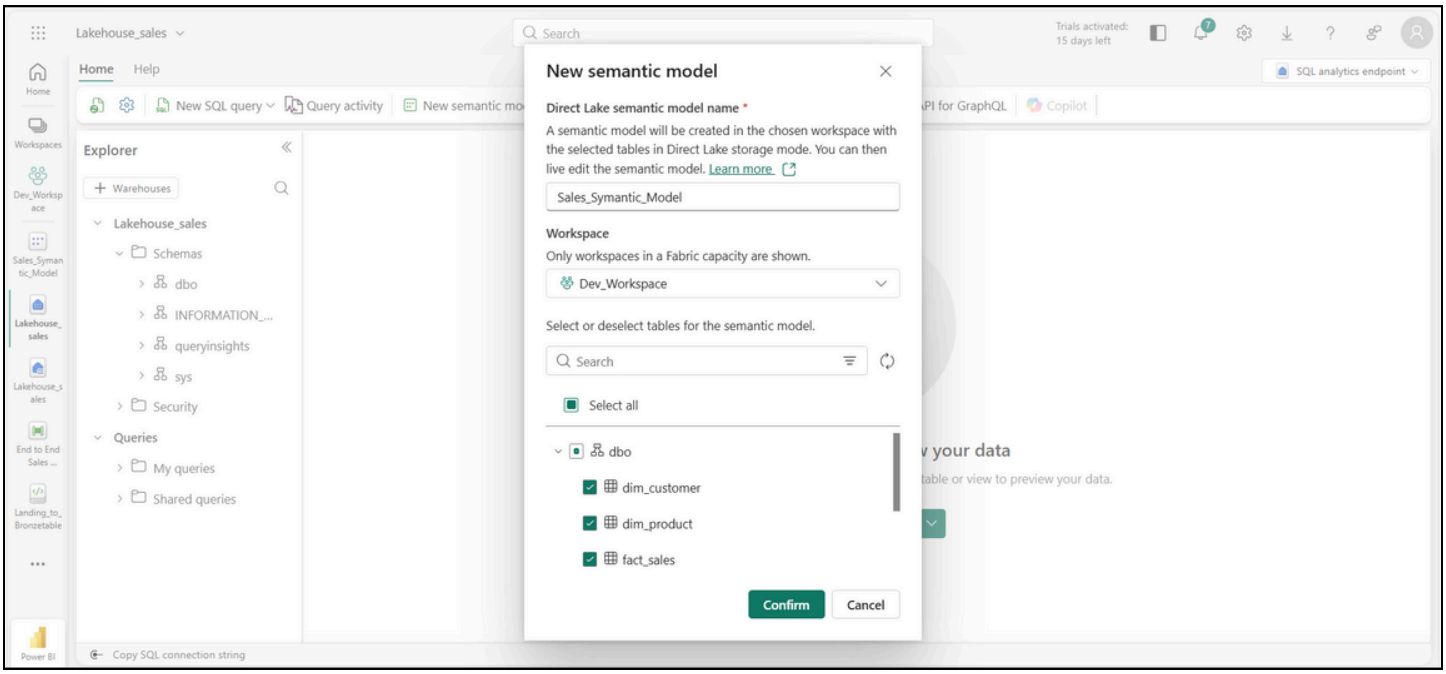
Quantity	Discount	Profit	Shipping_C...	Order_Priority	Month	Year	processing_...	Delivery_Days	Profit_Margin
5	0	7.56	2.44	High	September	2015	2026-01-05	4	0.48
4	0.2	3.29	1.77	High	September	2015	2026-01-05	2	0.3125
2	0	0.84	15.09	High	September	2015	2026-01-05	2	0.009648518263...
3	0.1	405.423	131.28	High	September	2015	2026-01-05	5	0.344403923637...
3	0	6.75	2.47	High	September	2015	2026-01-05	4	0.328947368421...
3	0	2.64	2.399	Medium	September	2015	2026-01-05	5	0.078853046594...
2	0.07	17.2548	20.16	Low	September	2015	2026-01-05	7	0.064341615990...
2	0	0	15.967	Low	September	2015	2026-01-05	7	0.0
1	0.4	-27.812	12.976	Critical	September	2015	2026-01-05	2	-0.65029928918...
1	0	14.37	4.64	Critical	September	2015	2026-01-05	3	0.279789719626...
8	0	95.04	15.5	Medium	September	2015	2026-01-05	5	0.369747899159...
1	0	27.03	3.55	Medium	September	2015	2026-01-05	4	0.329674350530...
1	0.7	-5.052	1.29	Critical	September	2015	2026-01-05	3	-1.46945898778...
3	0	8.107199999999...	10.65	High	September	2015	2026-01-05	4	0.079999999999...
4	0	3.7128	2.08	High	September	2015	2026-01-05	4	0.26
4	0	8.2992	1.42	Medium	September	2015	2026-01-05	4	0.42
4	0.2	3.4496	1.34	Medium	September	2015	2026-01-05	6	0.350000000000...

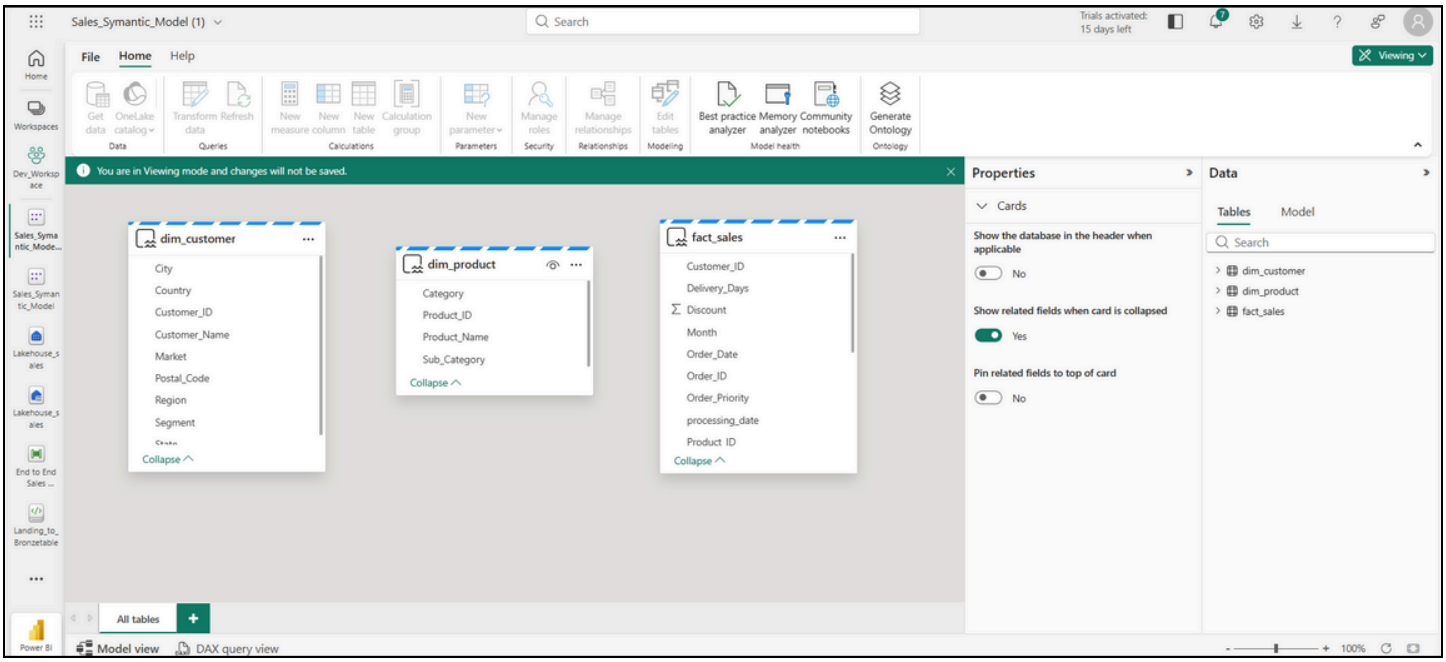
Step 9: Creating relation between tables(Modelling)

- Go to SQL Analytics Endpoint option of lakehaouse:

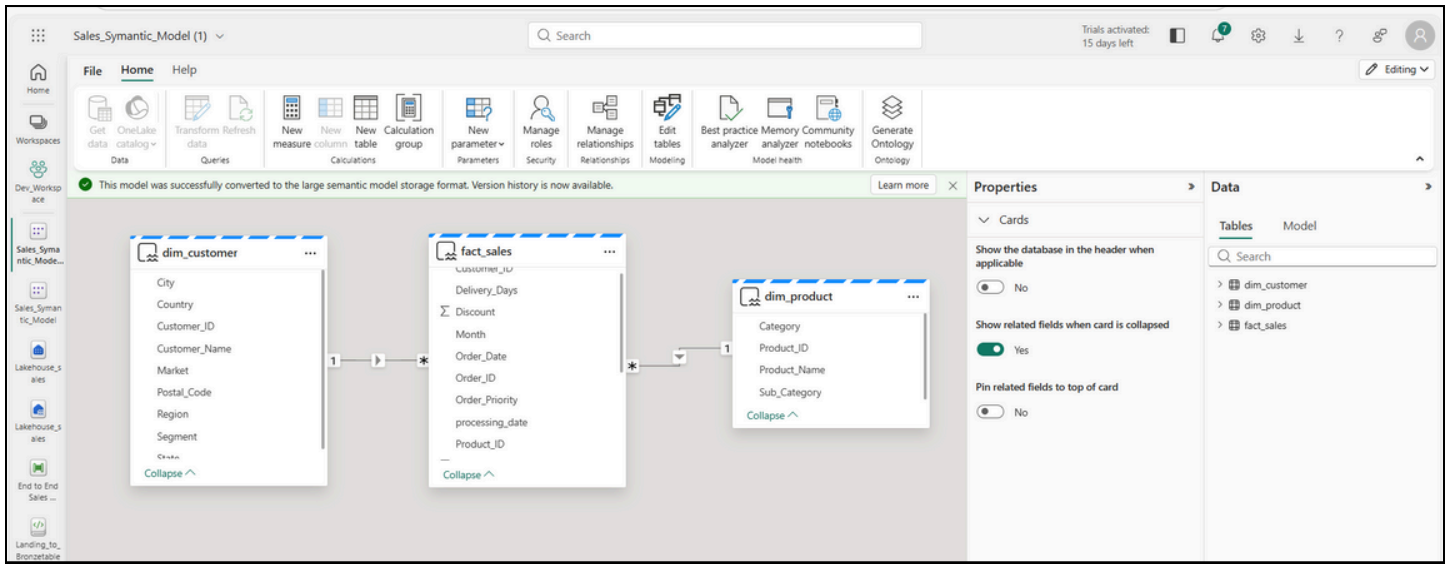


- Go to Reporting tab and Select new Symantic Model and choose tables



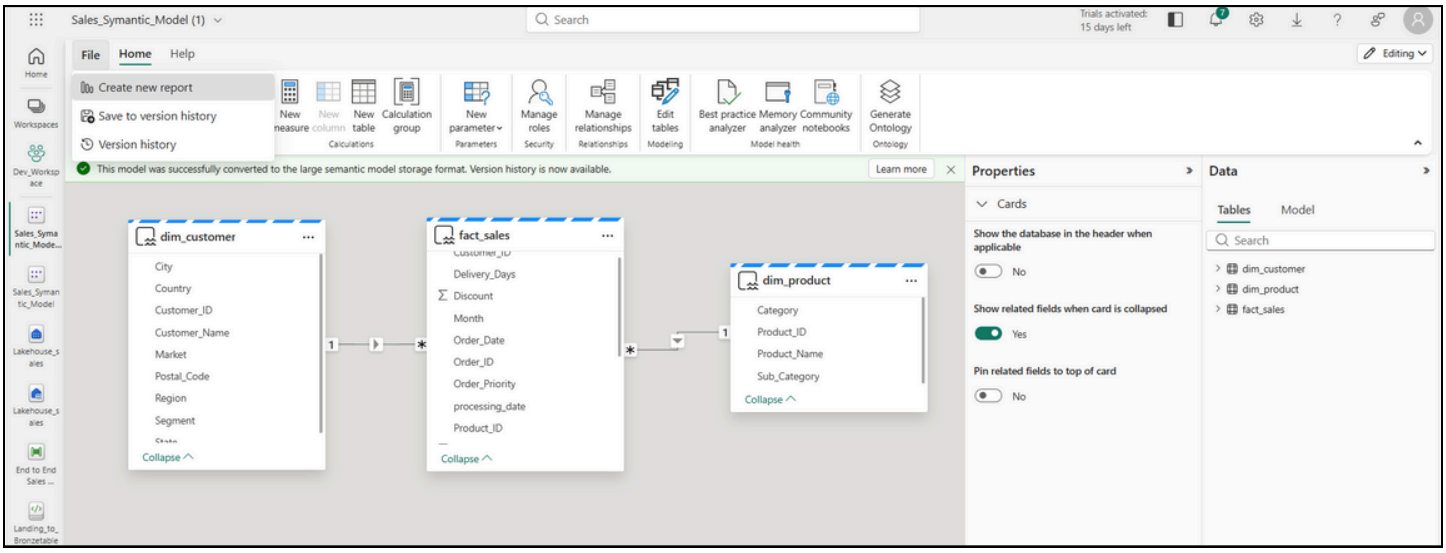


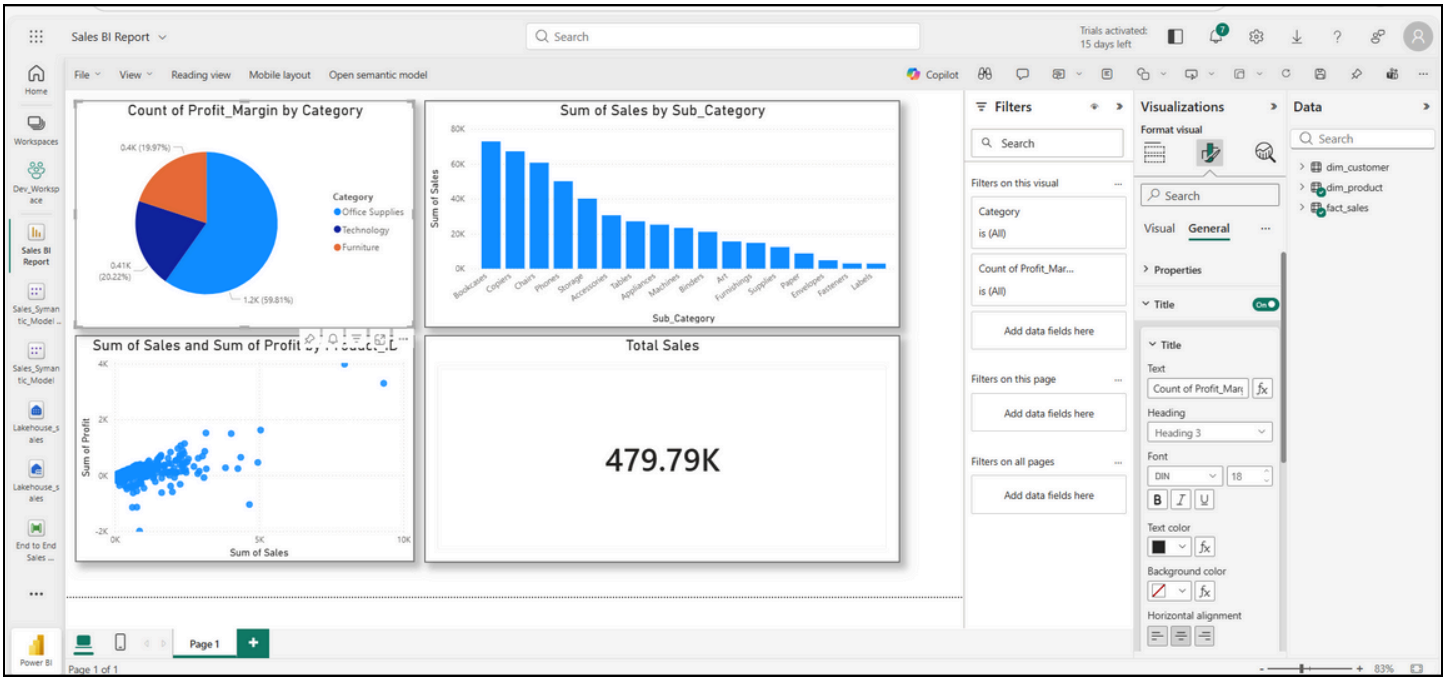
- join tables with primary and foreign keys



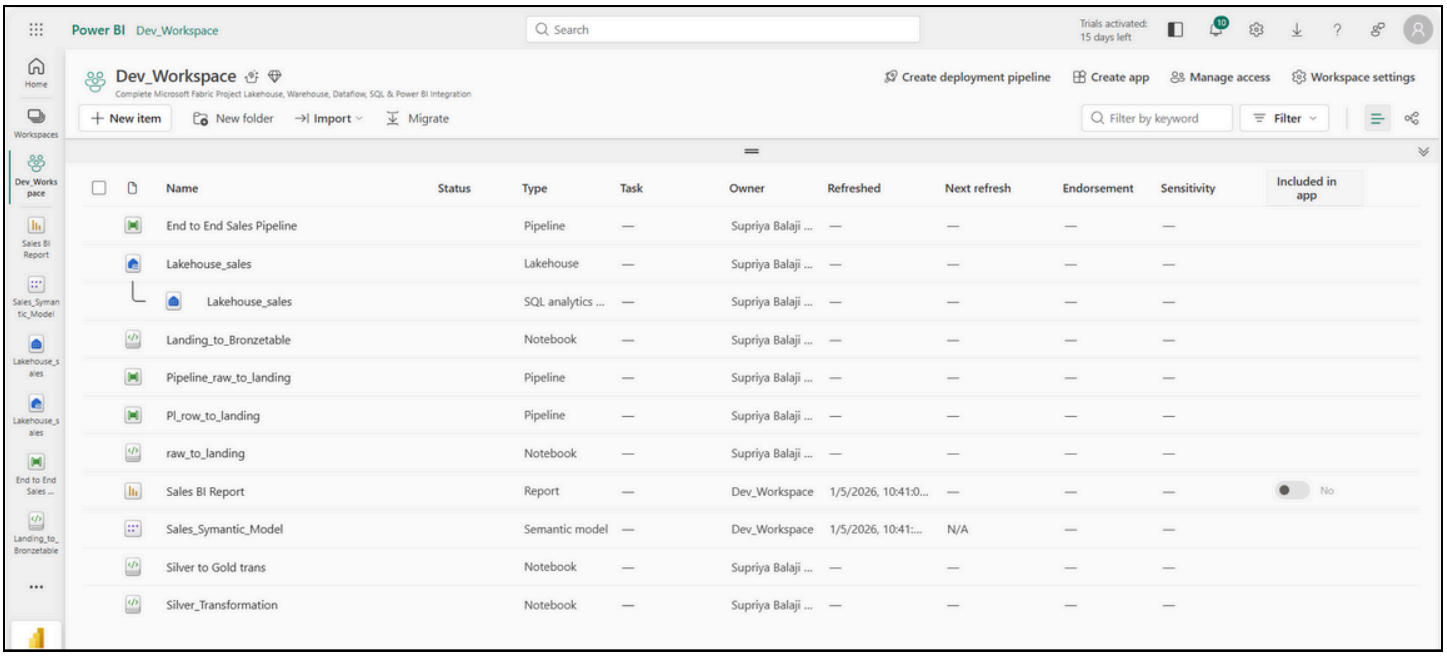
Step 10: Creation of Power BI Report

- now create Power BI report using this model:





Finally, the workspace contains following:



Conclusion

This project demonstrates the successful design and implementation of a fully functional, end-to-end data engineering and analytics solution using Microsoft Fabric. By employing a medallion architecture (Bronze → Silver → Gold), the solution ensures that raw data is progressively refined, cleansed, and transformed into business-ready insights. The automated data pipelines, combined with PySpark-based transformation notebooks, enable scalable and reliable processing of structured and unstructured data.

The implementation provides the following benefits:

- **Improved Data Quality:** Data is validated, cleansed, and enriched at multiple stages.
- **Scalability:** Pipelines and PySpark transformations can handle growing volumes of data efficiently.
- **Business-Ready Insights:** Gold-layer tables support actionable reporting and analytics via Power BI.
- **Governance & Monitoring:** Data lineage, quality checks, and monitoring ensure reliability and compliance.