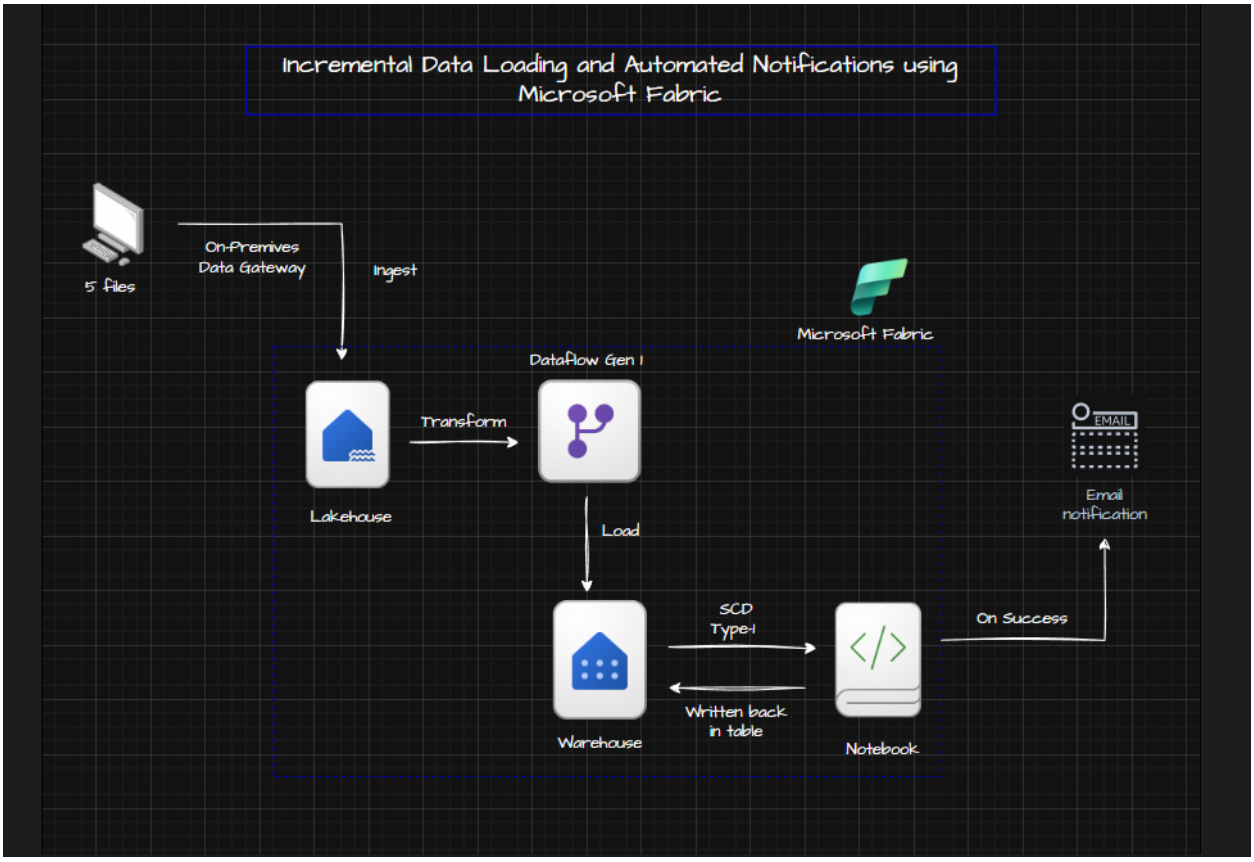
Project 4 – Dynamic Data Ingestion from On-Premises to Microsoft Fabric Lakehouse

Objective

To dynamically ingest multiple structured data files located on a local file system into Microsoft Fabric Lakehouse tables using **Dataflow Gen2** and **On-Premises Data Gateway**, ensuring only the **latest version** of each file is processed.

Data Sources

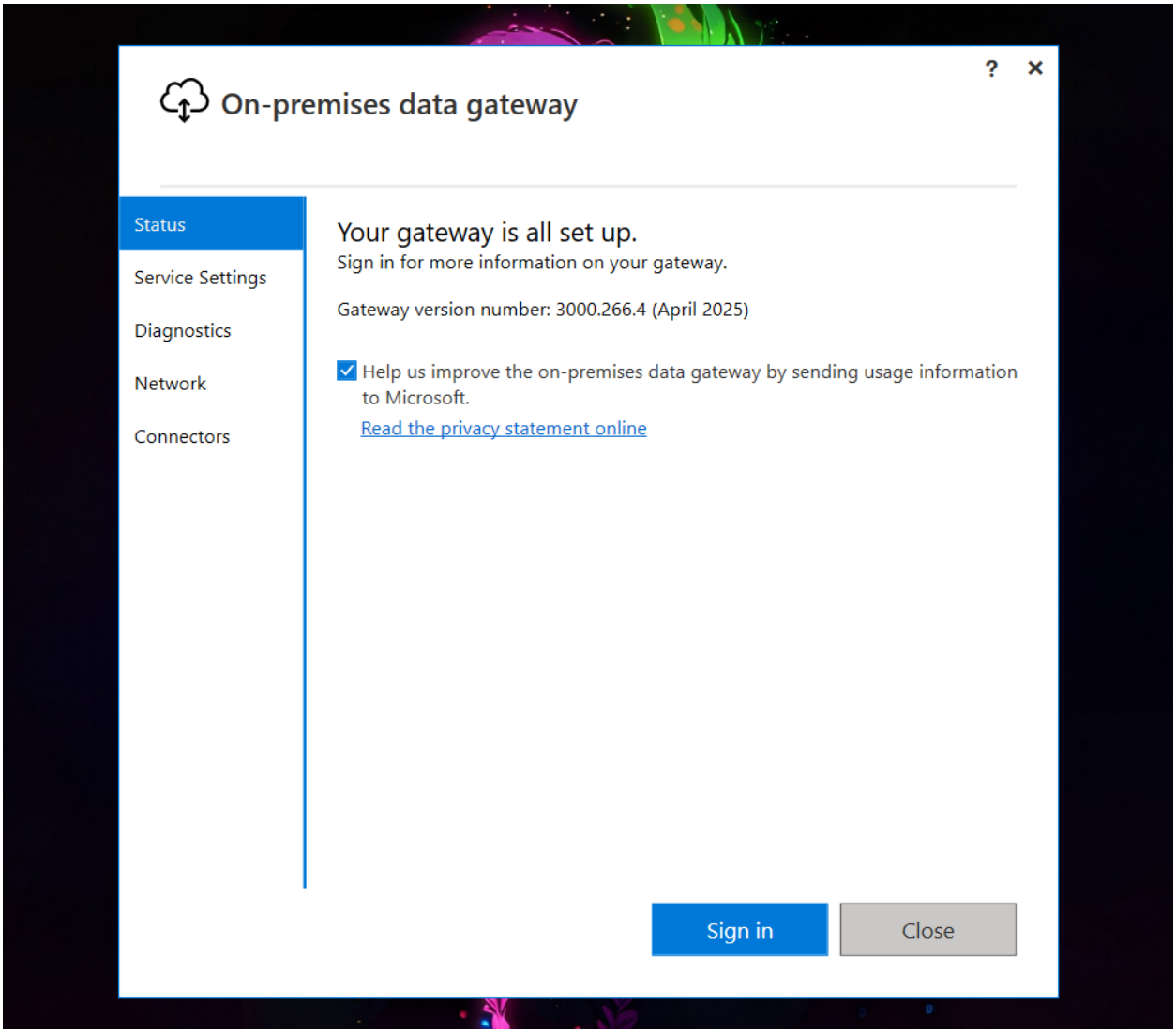
- **Location**: AI Bank Dataset
- **Files**:
    - accounts.csv
    - customers.csv
    - loans.csv
    - loan_payments.csv
    - transactions.csv
- Architecture Diagram

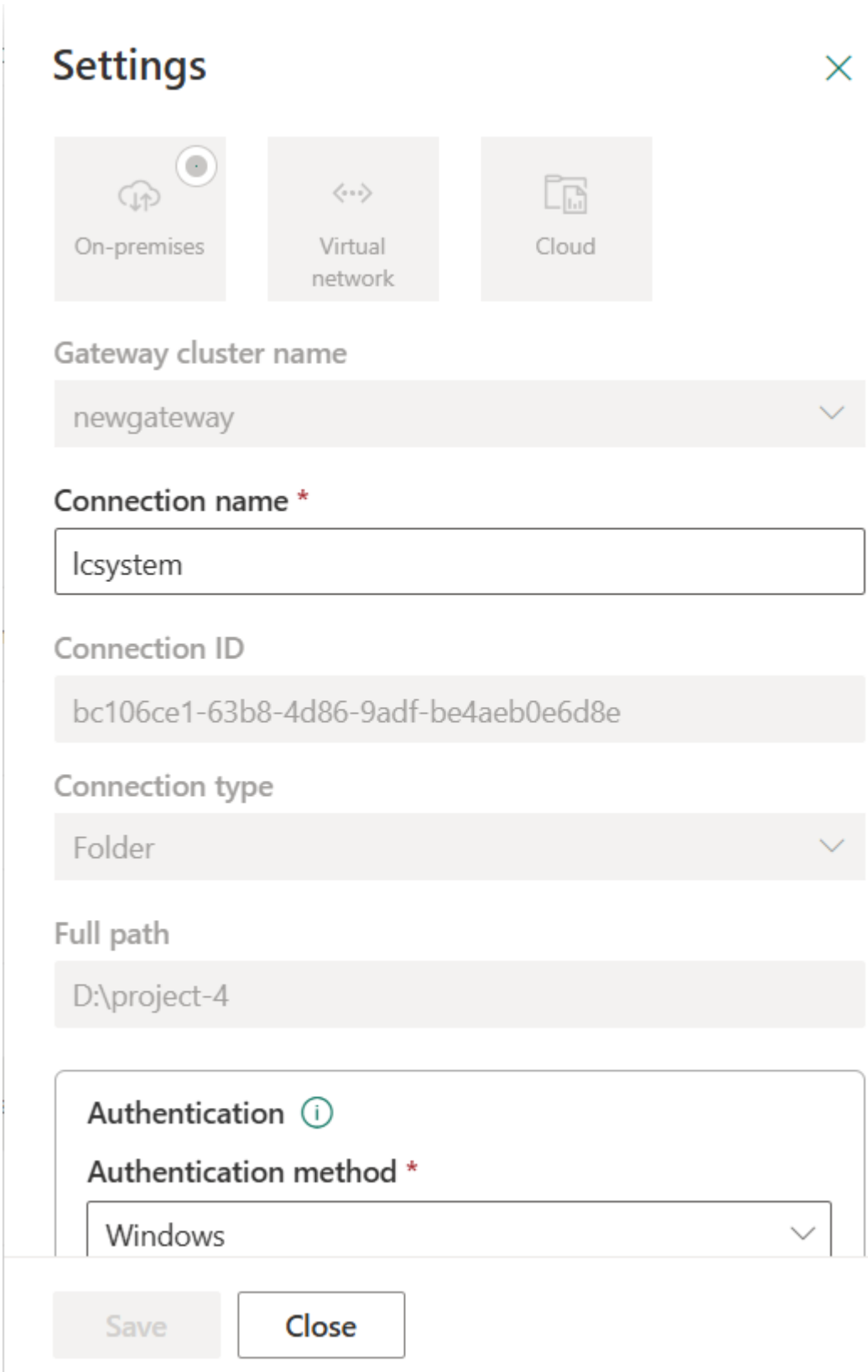


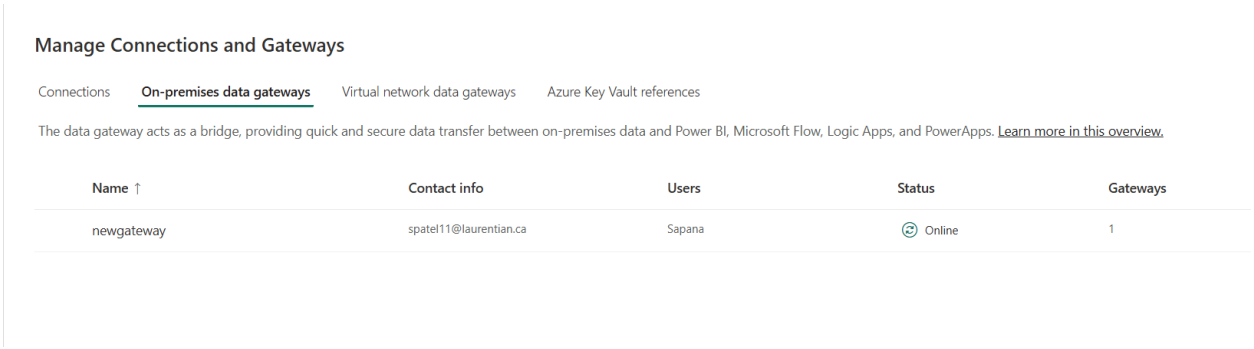
1. Bronze Layer :- load data from local system to fabric Lakehouse

1. Gateway Installation & Connection

- Installed **On-Premises Data Gateway** on the VM.
- Configured Gateway in **Fabric** to enable folder access.



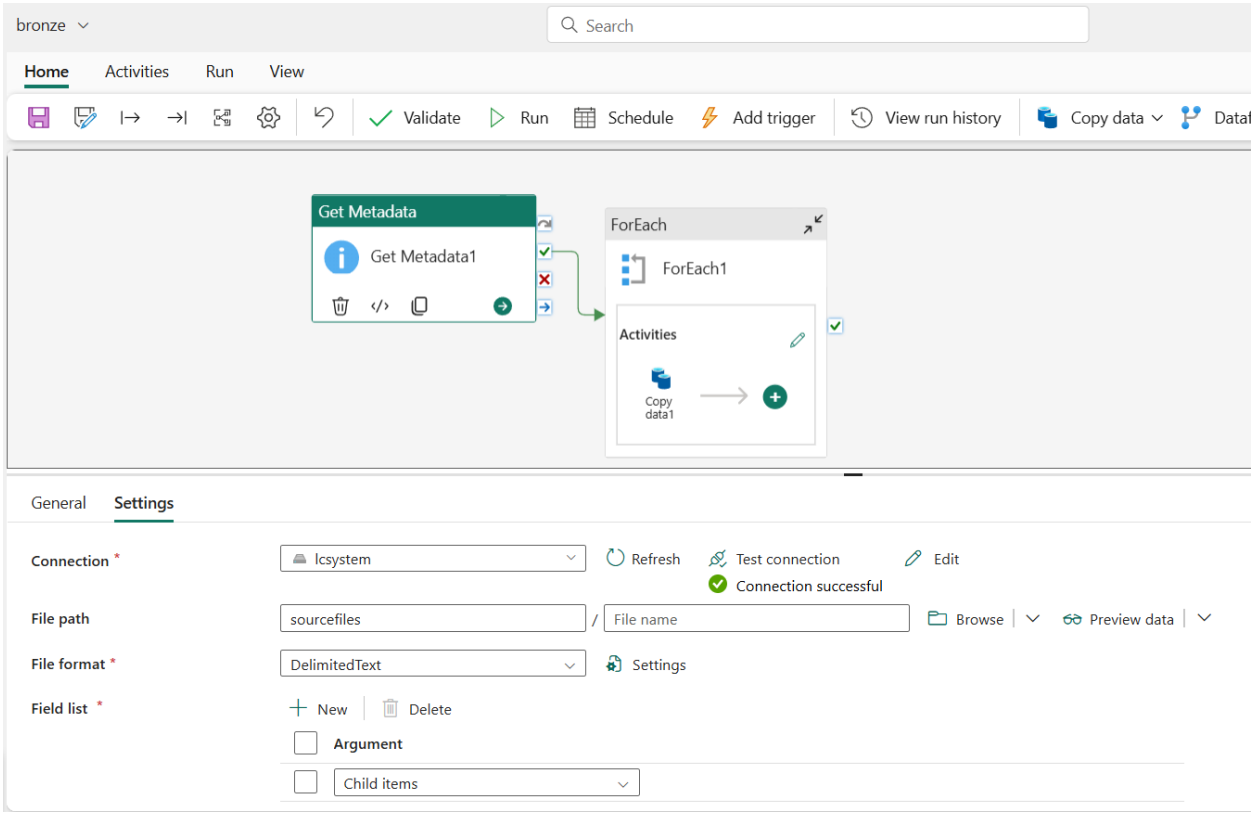

Settings
On-premises
Virtual network
Cloud
Gateway cluster name
newgateway
Connection name *
lcsystem
Connection ID
bc106ce1-63b8-4d86-9adf-be4aeb0e6d8e
Connection type
Folder
Full path
D:\project-4
Authentication
Authentication method *
Windows
Save
Close



Step 1: Get Metadata Activity (List of Files)

Configuration:

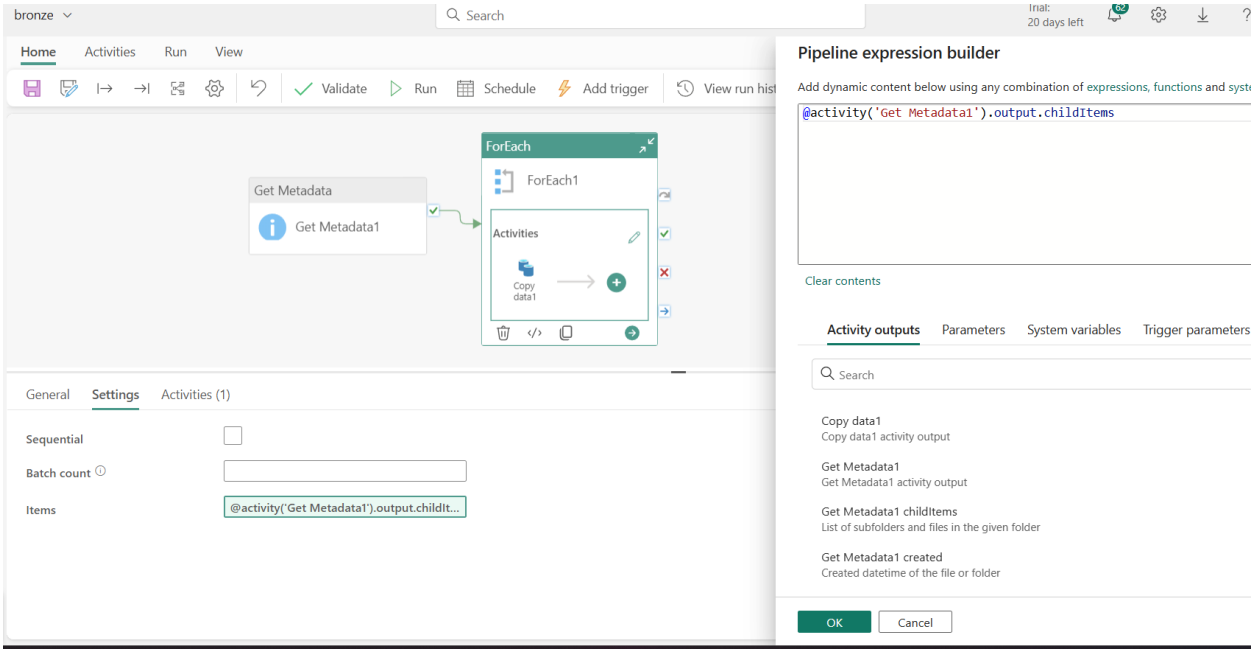
- Dataset points to **Folder Path** on your VM, e.g., C:\Project-4.
- **Field List**: Child Items.
- Returns an **array of file names** in that folder.



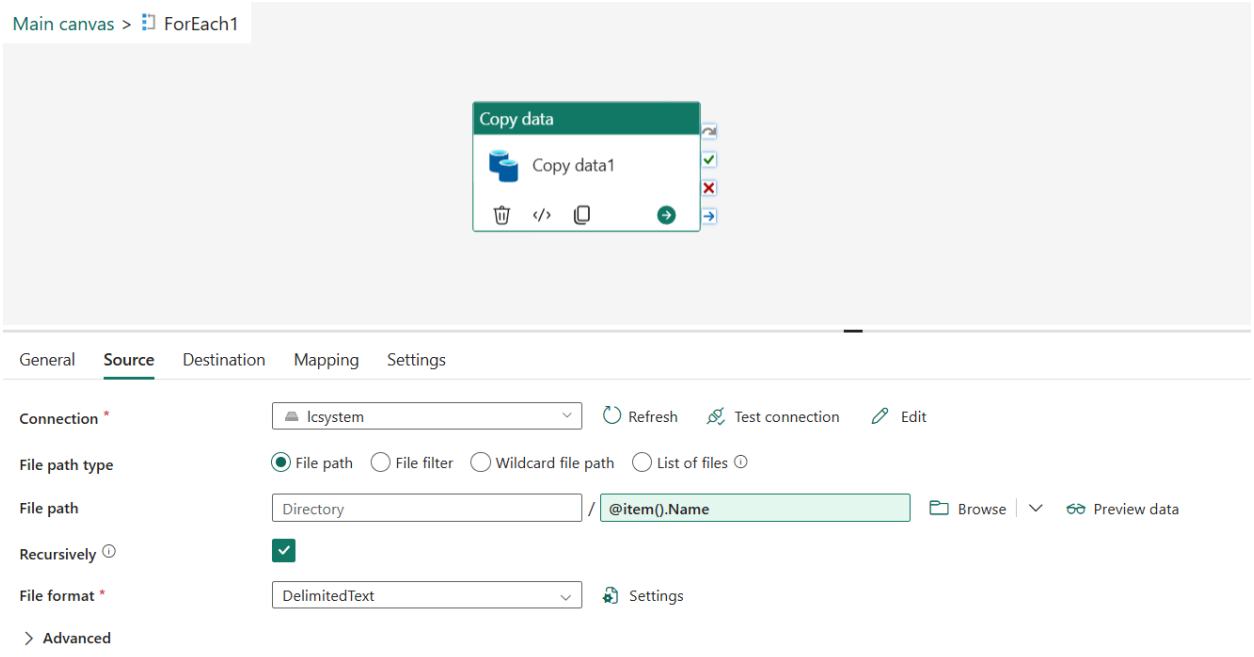
Step 2: ForEach Activity (Loop Over Files)

Configuration:

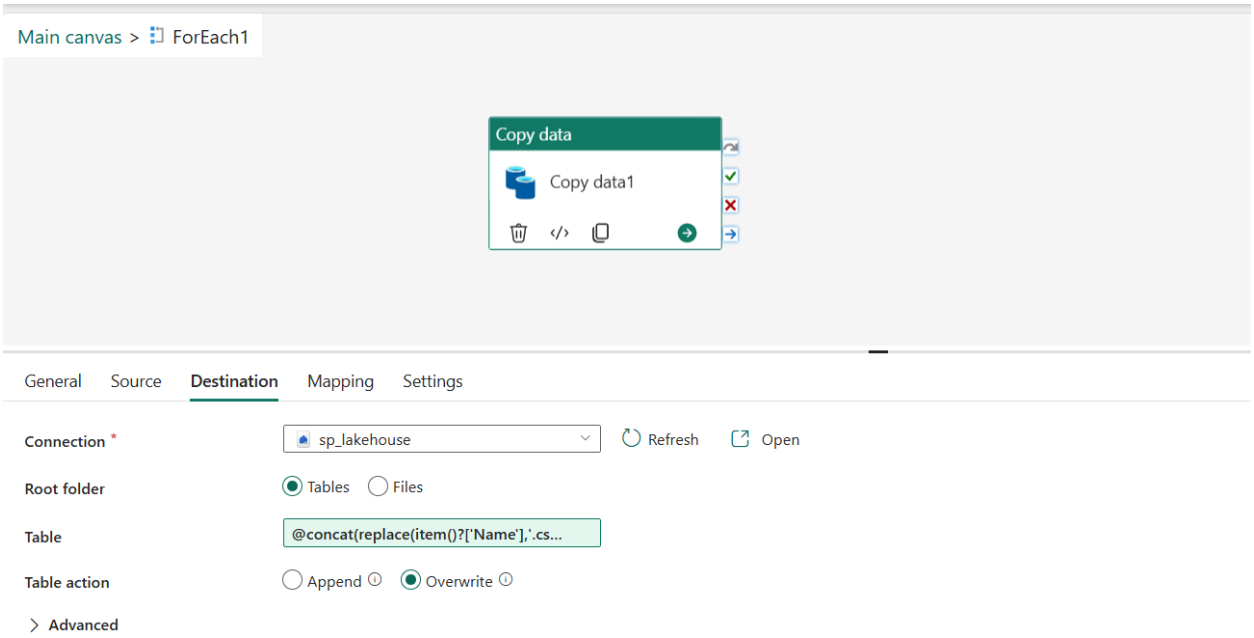
- **Items**: @activity('Get Metadata1').output.childItems
- **Batch Count**: Set to 1 for sequential processing (or higher for parallel).



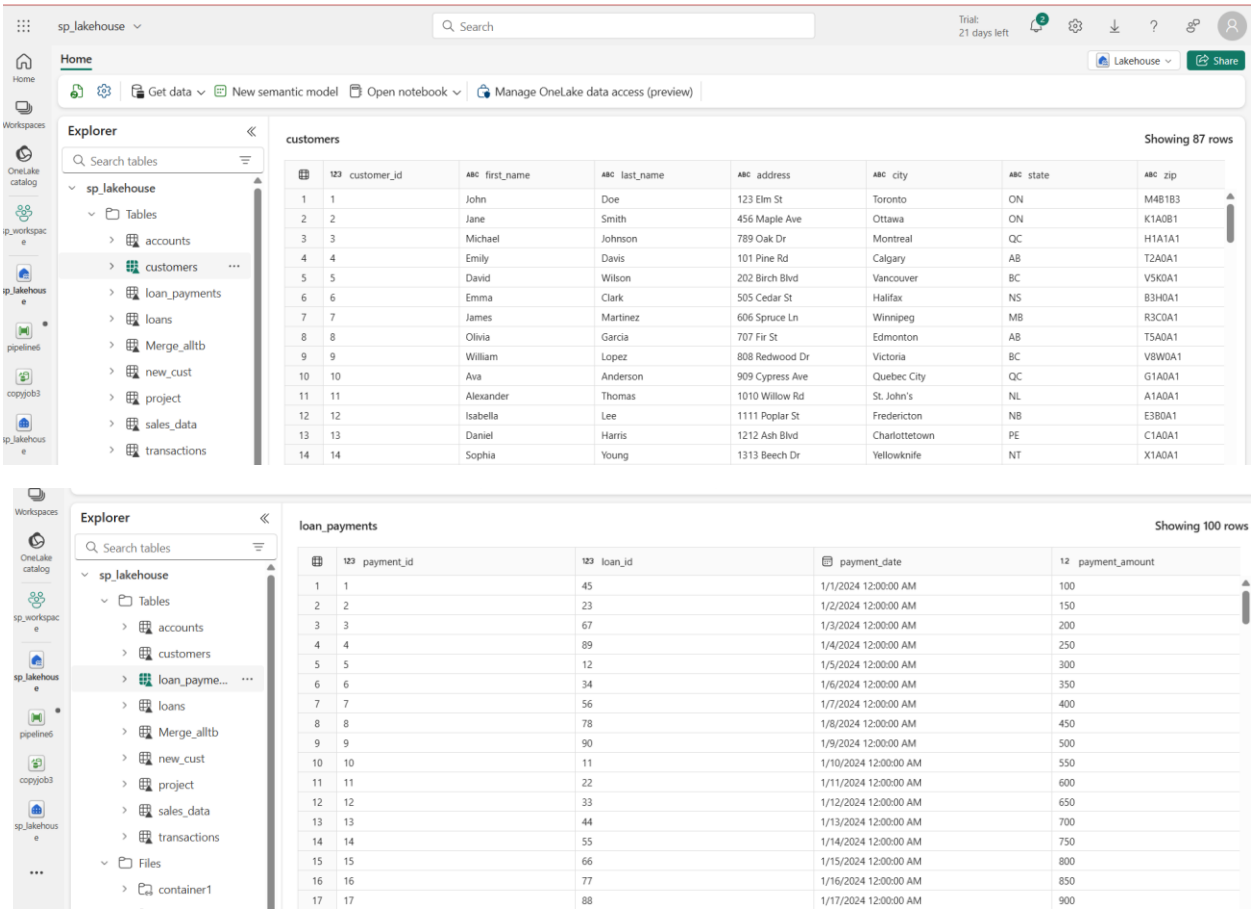
Step 3: Copy Data Activity (Load Each File Dynamically)

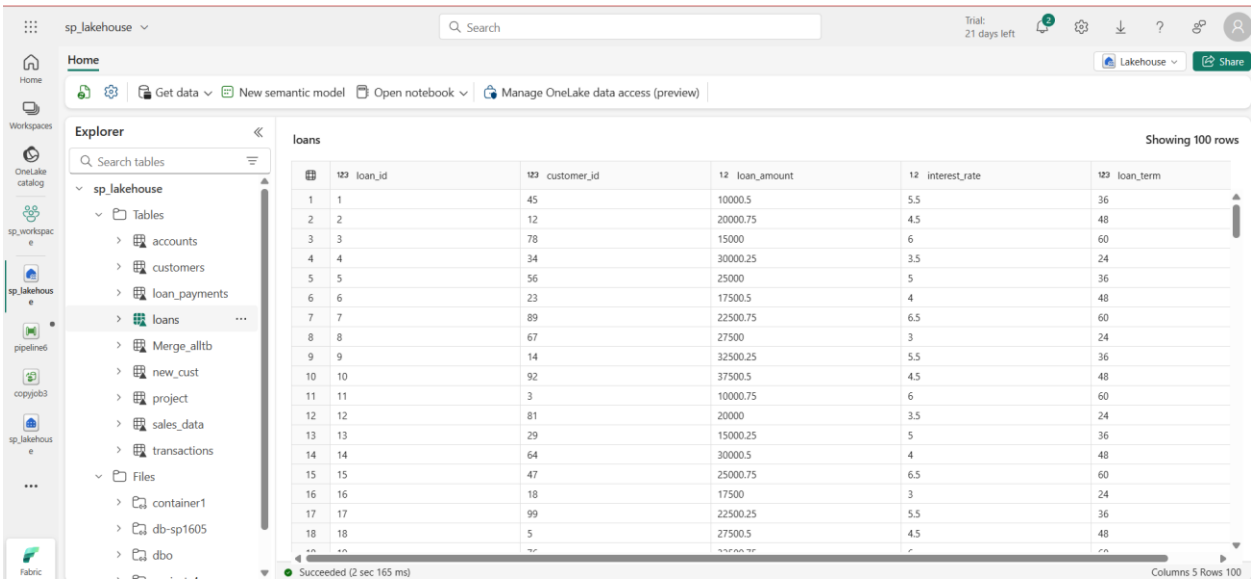


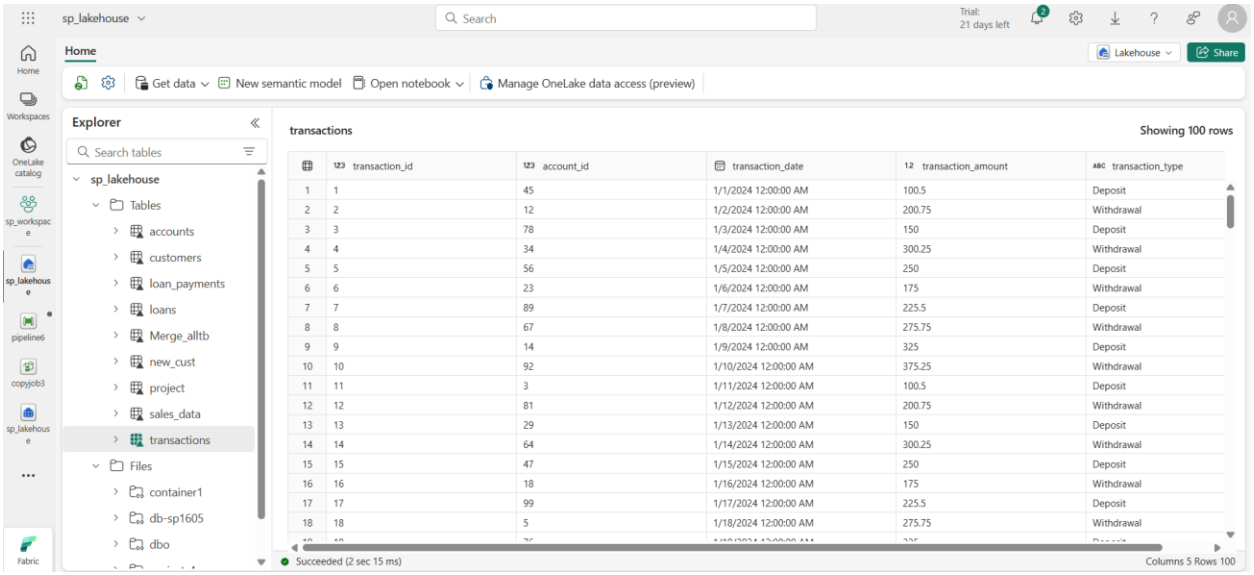
Parameterized dataset for **File Path** and **File Name**.



Each file loaded to **Lakehouse**.

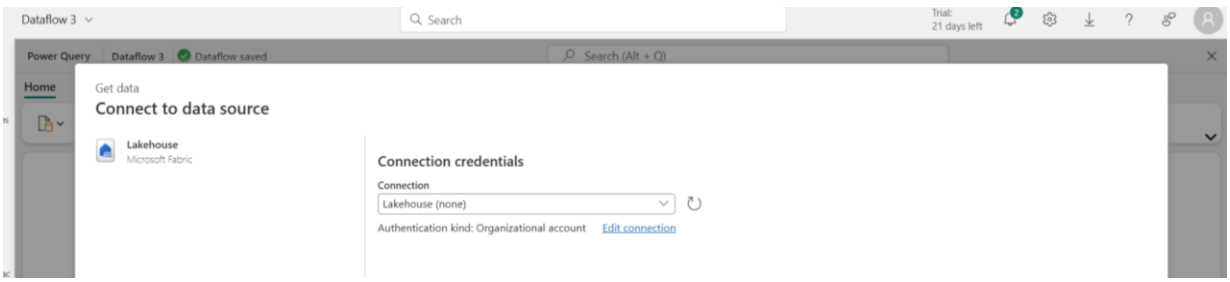


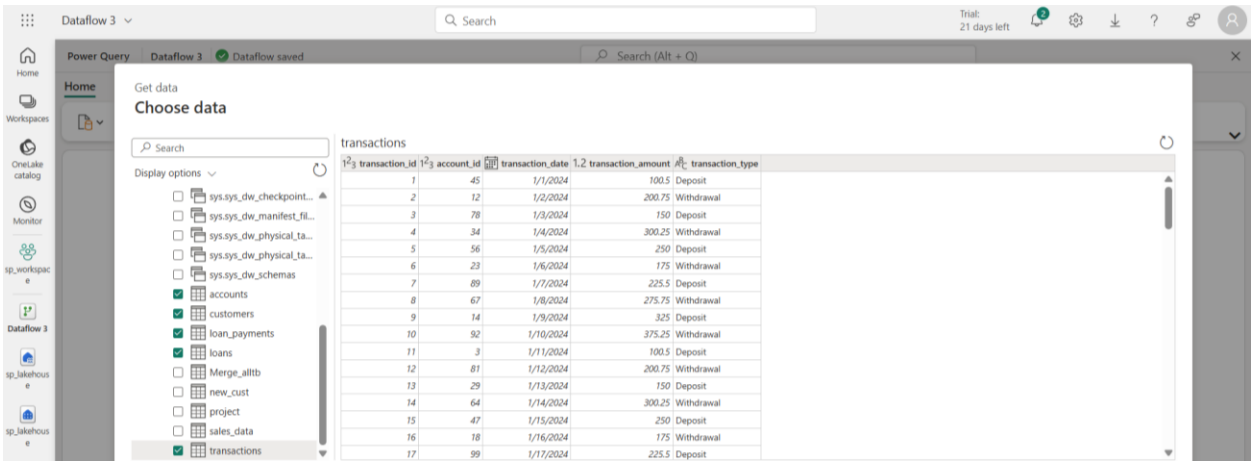


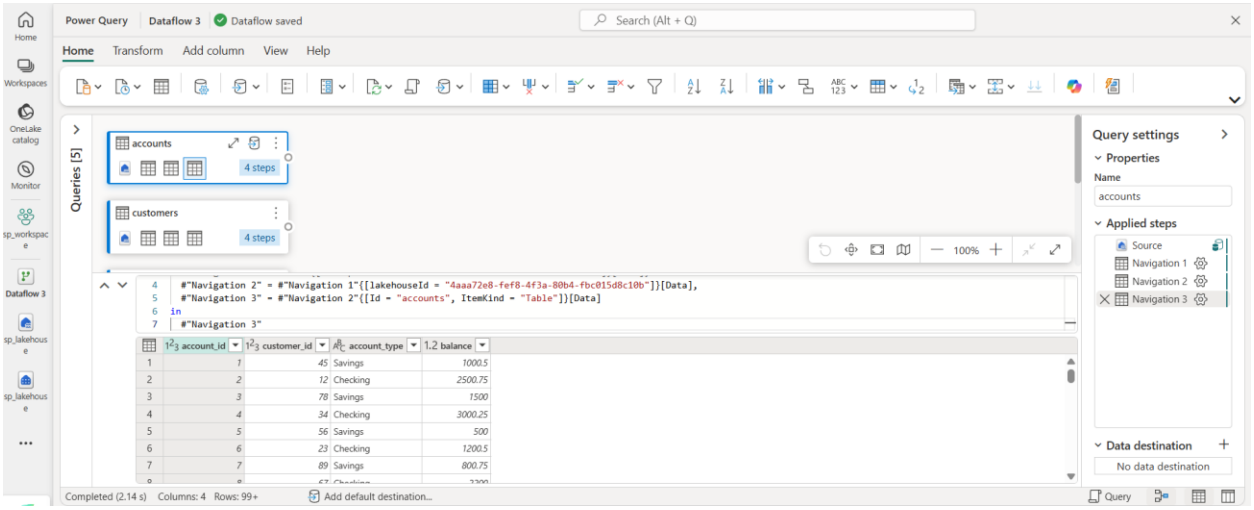


2. Bronze to Silver:- Clean data using dataflow gen2 in fabric

Select Dataflow gen2 and got to source ----Lakehouse ----select 5 all files for data cleaning.

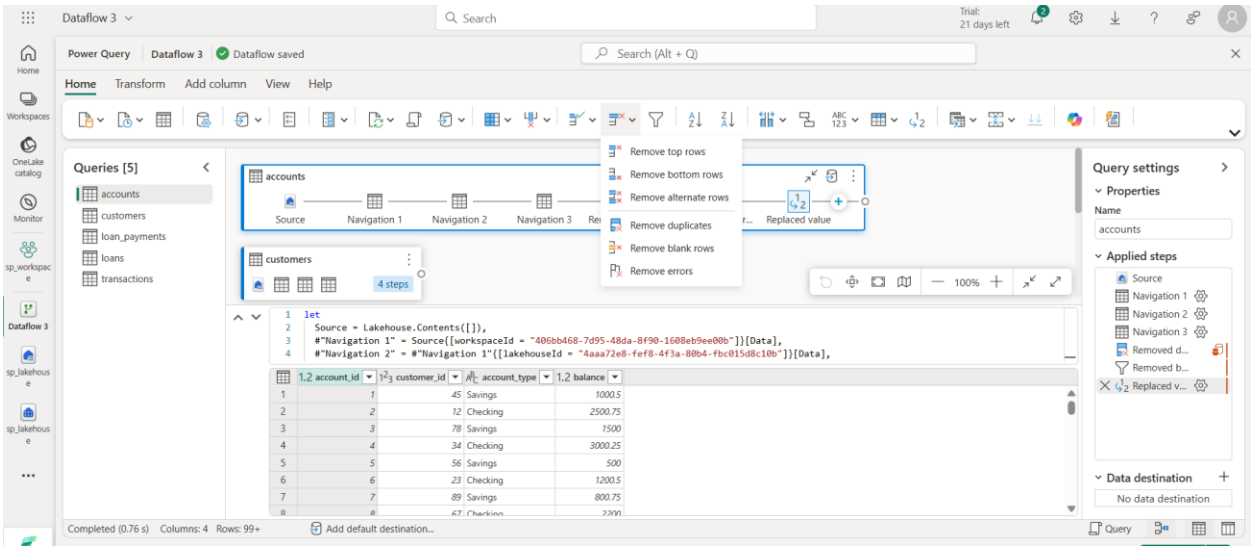




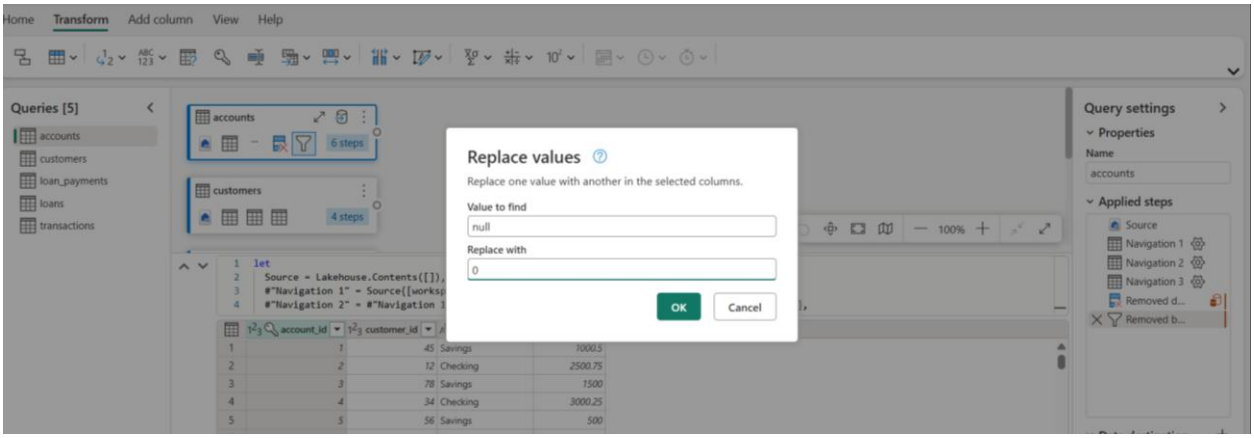


Explanation of data filtering process for Account table data :-

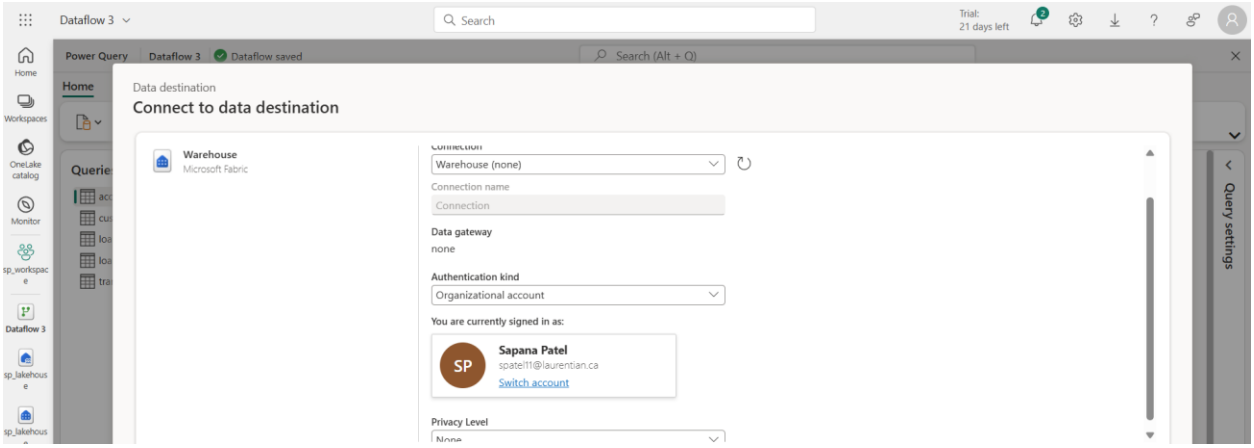
Select remove duplicates and remove blank rows to filter data and all into dataflow.



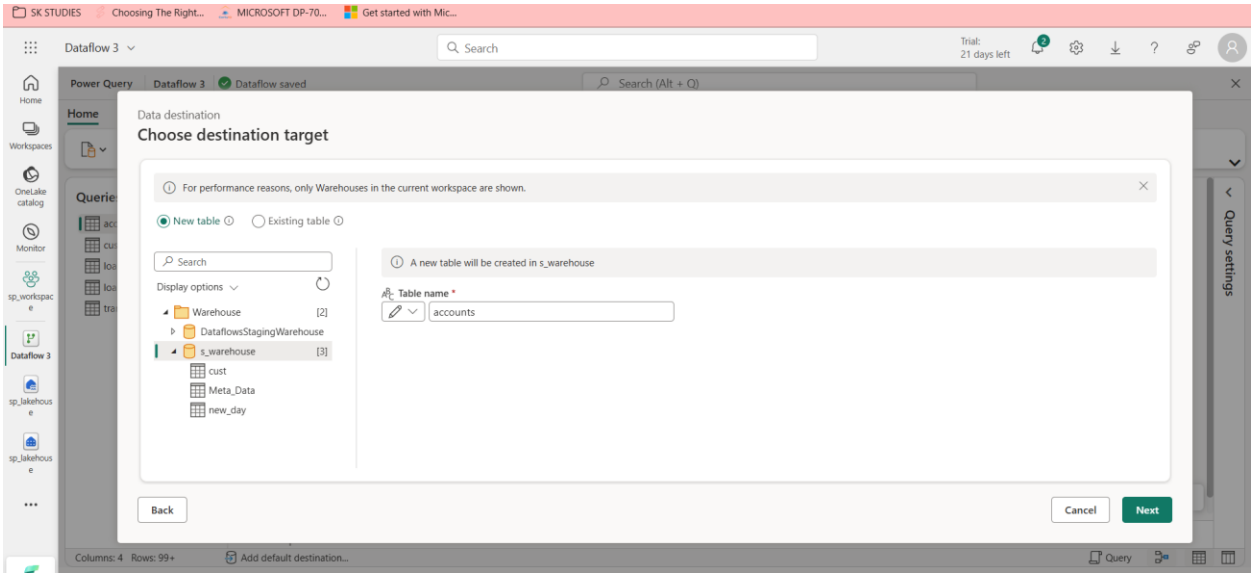
Select replace values to remove null values from data. Select null == 0 .

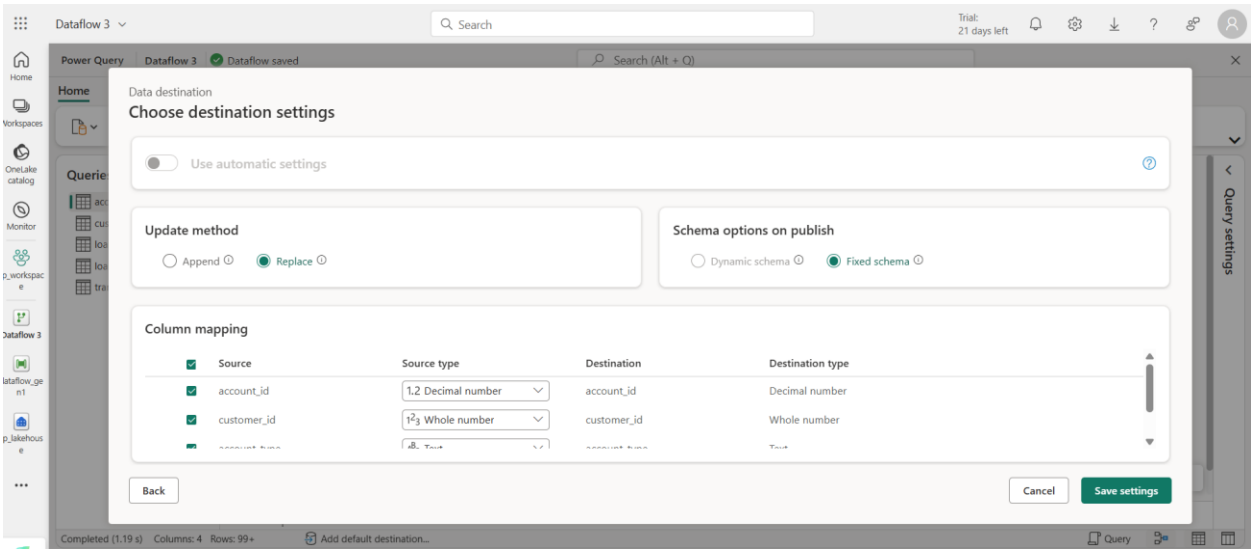


Select destination as a warehouse.

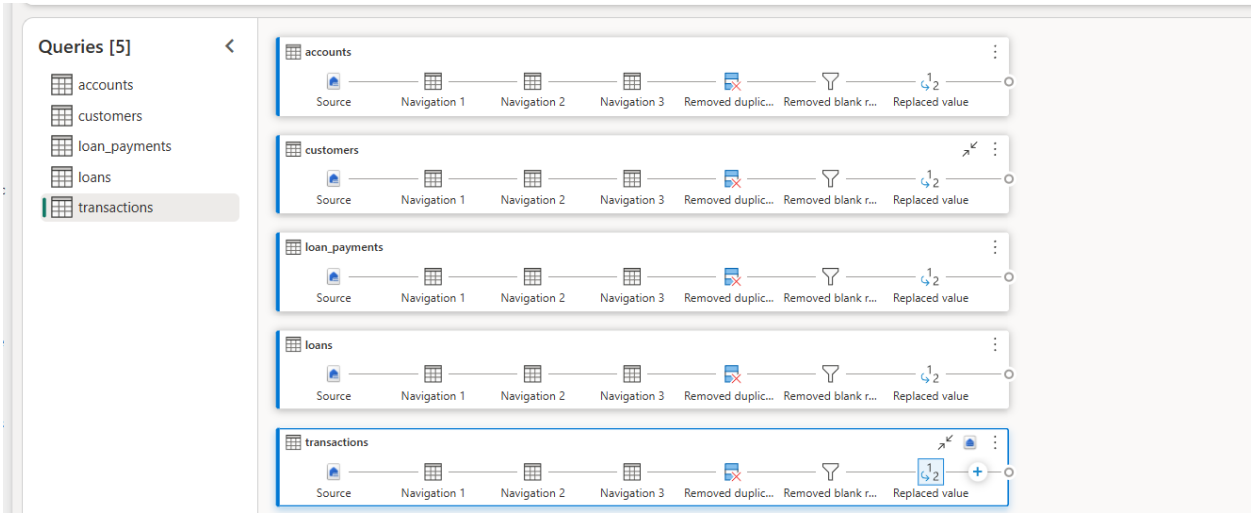


Select the new table in warehouse.

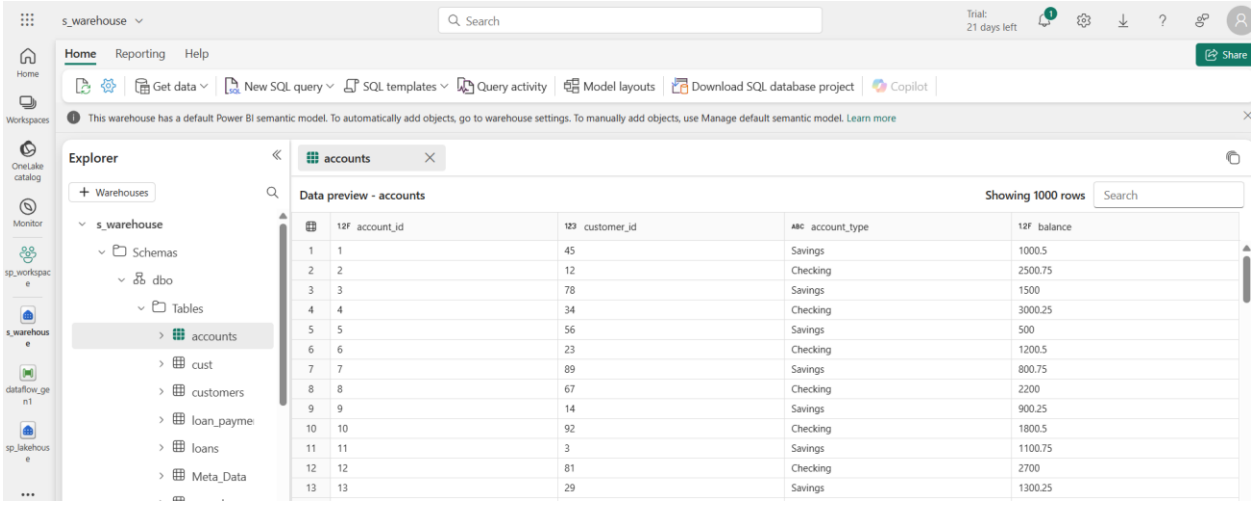




Repeat same steps for all other 4 files in data flow.



We got all tables in warehouse.



3. Silver to Gold :- SCD TYPE-1 Perform in Fabric Notebook by using Pyspark.

1. Import Required Libraries

```
import com.microsoft.spark.fabric
from com.microsoft.spark.fabric.Constants import Constants
import pandas as pd
```

2. Load Source Data from Warehouse

```
df = spark.read.synapsesql("s_warehouse.dbo.accounts")
df.show()
```

- Command executed in 3 sec 482 ms by Sapana Patel on 3:05:53 PM, 5/11/25

```
+-----------+-------+------------+----------+
|customer_id|balance|account_type|account_id|
+-----------+-------+------------+----------+
|         45| 1000.5|     Savings|       1.0|
|         12|2500.75|    Checking|       2.0|
|         78| 1500.0|     Savings|       3.0|
|         34|3000.25|    Checking|       4.0|
|         56|  500.0|     Savings|       5.0|
```

3.Created Target table for accounts

```
# Define Delta table creation query
create_table_query = """
CREATE TABLE IF NOT EXISTS Accounts_Gold (
    account_id INT,
    customer_id INT,
    account_type STRING,
    balance FLOAT,
    hash_key BIGINT,
    created_by STRING,
    created_date TIMESTAMP,
    updated_by STRING,
    updated_date TIMESTAMP
)
USING DELTA
LOCATION 'Files/Gold_layer/Accounts_Gold'
"""

# Execute the table creation
spark.sql(create_table_query)

#spark.sql("DROP TABLE IF EXISTS Customers_Gold")
```

- Command executed in 2 sec 433 ms by Sapana Patel on 1:57:52 PM, 5/11/25

DataFrame[]

4. Add Metadata and Hash Key

```
from pyspark.sql.functions import col, lit, current_timestamp, crc32, concat_ws
from delta.tables import DeltaTable

# Load source data from Warehouse Silver Layer
source_df = spark.read.synapsesql("s_warehouse.dbo.accounts")

# Add metadata and Hash Key
source_df = source_df \
    .withColumn("hash_key", crc32(concat_ws("||", *source_df.columns))) \
    .withColumn("created_date", current_timestamp()) \
    .withColumn("updated_date", current_timestamp()) \
    .withColumn("created_by", lit("fabric")) \
    .withColumn("updated_by", lit("fabric"))
```

5. **Load Target Delta Table**

```
# Load target Delta Table in Lakehouse (Gold Layer)
target_table_path = "Files/Gold_layer/Accounts_Gold"
target_delta = DeltaTable.forPath(spark, target_table_path)
target_df = target_delta.toDF()
```

6. Identify New or Updated Records

```
# Find records that are new or changed by comparing account_id and hash_key
df_diff = source_df.alias("src").join(
    target_df.alias("tgt"),
    (col("src.account_id") == col("tgt.account_id")) & (col("src.hash_key") == col("tgt.hash_key")),
    "anti"
).select("src.*")
```

7. Apply SCD-1 Merge Logic

```
# Apply SCD-1 Merge Logic to Gold Layer Delta Table
target_delta.alias("tgt").merge(
    df_diff.alias("src"),
    "tgt.account_id = src.account_id"
) \
.whenMatchedUpdate(set={
    "account_id": "src.account_id",
    "customer_id": "src.customer_id",
    "account_type": "src.account_type",
    "balance": "src.balance",
    "hash_key": "src.hash_key",
    "updated_date": "current_timestamp()",
    "updated_by": lit("fabric-updated")
}) \
.whenNotMatchedInsert(values={
    "account_id": "src.account_id",
    "customer_id": "src.customer_id",
    "account_type": "src.account_type",
    "balance": "src.balance",
    "hash_key": "src.hash_key",
    "created_date": "current_timestamp()",
    "created_by": lit("fabric"),
    "updated_date": "current_timestamp()",
    "updated_by": lit("fabric")
}).execute()

df = spark.read.format("delta").load("Files/Gold_layer/Accounts_Gold")
```

8. Verify Target Table Content

```
28
29  # Convert to Pandas and display as table
30  pd_df = df.toPandas()
31  display(pd_df)
32
```

[87] ✓ - Command executed in 4 sec 800 ms by Sapana Patel on 3:07:32 PM, 5/11/25

Table | New chart

Table view — Download

	account_id	customer_id	account_type	balance	hash_key	created_by	created_date	updated_by	updated_date
1	1	45	Savings	1000.5	3361178304	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
2	2	12	Checking	2500.75	907240447	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
3	3	78	Savings	1500.0	89572566	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
4	4	34	Checking	3000.25	581580781	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
5	5	56	Savings	500.0	628982445	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
6	6	23	Checking	1200.5	3905468925	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
7	7	89	Savings	800.75	3802202745	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
8	8	67	Checking	2200.0	1906974630	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
9	9	14	Savings	900.25	1069521115	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
10	10	92	Checking	1800.5	2882685028	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
11	11	3	Savings	1100.75	1845849907	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
12	12	81	Checking	2700.0	4217717292	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...
13	13	29	Savings	1300.25	3077108573	fabric	2025-05-11 17:3...	fabric	2025-05-11 17:37...

Similar Steps are repeated for other 4 files and same line of code for SCD type-1 logic.

Here is 4 table data that loaded in target table.

Table | New chart — 12 columns, 87 rows

Table view — Download — Search

	customer_id	first_name	last_name	address	city	state	zip	hash_key	created_by	created_date	updated_by	update
1	1	John	Doe	123 Elm St	Toronto	ON	M4B1B3	3786732380	fabric	2025-05-11 18:4...	fabric	2025-0
2	2	Jane	Smith	456 Maple A...	Ottawa	ON	K1A0B1	1390150273	fabric	2025-05-11 18:4...	fabric	2025-0
3	3	Michael	Johnson	789 Oak Dr	Montreal	QC	H1A1A1	2899704259	fabric	2025-05-11 18:4...	fabric	2025-0
4	4	Emily	Davis	101 Pine Rd	Calgary	AB	T2A0A1	3870622128	fabric	2025-05-11 18:4...	fabric	2025-0
5	5	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1	3487818084	fabric	2025-05-11 18:4...	fabric	2025-0
6	6	Emma	Clark	505 Cedar St	Halifax	NS	B3H0A1	947878324	fabric	2025-05-11 18:4...	fabric	2025-0
7	7	James	Martinez	606 Spruce Ln	Winnipeg	MB	R3C0A1	3905582380	fabric	2025-05-11 18:4...	fabric	2025-0
8	8	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1	4020006276	fabric	2025-05-11 18:4...	fabric	2025-0
9	9	William	Lopez	808 Redwoo...	Victoria	BC	V8W0A1	34926619	fabric	2025-05-11 18:4...	fabric	2025-0
10	10	Ava	Anderson	909 Cypress ...	Quebec City	QC	G1A0A1	3673199934	fabric	2025-05-11 18:4...	fabric	2025-0
11	11	Alexander	Thomas	1010 Willow...	St. John's	NL	A1A0A1	418981251	fabric	2025-05-11 18:4...	fabric	2025-0
12	12	Isabella	Lee	1111 Poplar St	Fredericton	NB	E3B0A1	2357521938	fabric	2025-05-11 18:4...	fabric	2025-0
13	13	Daniel	Harris	1212 Ash Blvd	Charlotteto...	PE	C1A0A1	941740525	fabric	2025-05-11 18:4...	fabric	2025-0

Table | New chart — 10 columns, 100

Table view — Download — Search

	loan_id	customer_id	loan_term	loan_amount	interest_rate	hash_key	created_by	created_date	updated_by	updated_date
1	1	45	36	10000.5	5.5	321323532	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
2	2	12	48	20000.75	4.5	1152749887	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
3	3	78	60	15000.0	6.0	1999392758	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
4	4	34	24	30000.25	3.5	452958275	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
5	5	56	36	25000.0	5.0	3042352453	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
6	6	23	48	17500.5	4.0	3510396815	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
7	7	89	60	22500.75	6.5	277152135	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
8	8	67	24	27500.0	3.0	1770314337	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
9	9	14	36	32500.25	5.5	2527454859	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
10	10	92	48	37500.5	4.5	3518443399	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
11	11	3	60	10000.75	6.0	3128066097	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
12	12	81	24	20000.0	3.5	63114391	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...
13	13	29	36	15000.25	5.0	464270404	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:51...

Table | New chart

Table view

Download

	payment_id	loan_id	payment_date	payment_amount	hash_key	created_by	created_date	updated_by	updated_date
1	1	45	2024-01-01	100.0	2737019449	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
2	2	23	2024-01-02	150.0	3509823974	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
3	3	67	2024-01-03	200.0	3219615906	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
4	4	89	2024-01-04	250.0	577236198	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
5	5	12	2024-01-05	300.0	1196719864	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
6	6	34	2024-01-06	350.0	1361610787	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
7	7	56	2024-01-07	400.0	3108358108	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
8	8	78	2024-01-08	450.0	3679801302	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
9	9	90	2024-01-09	500.0	1747851957	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
10	10	11	2024-01-10	550.0	1419562063	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...
11	11	22	2024-01-11	600.0	3949794927	fabric	2025-05-11 18:5...	fabric	2025-05-11 18:56...

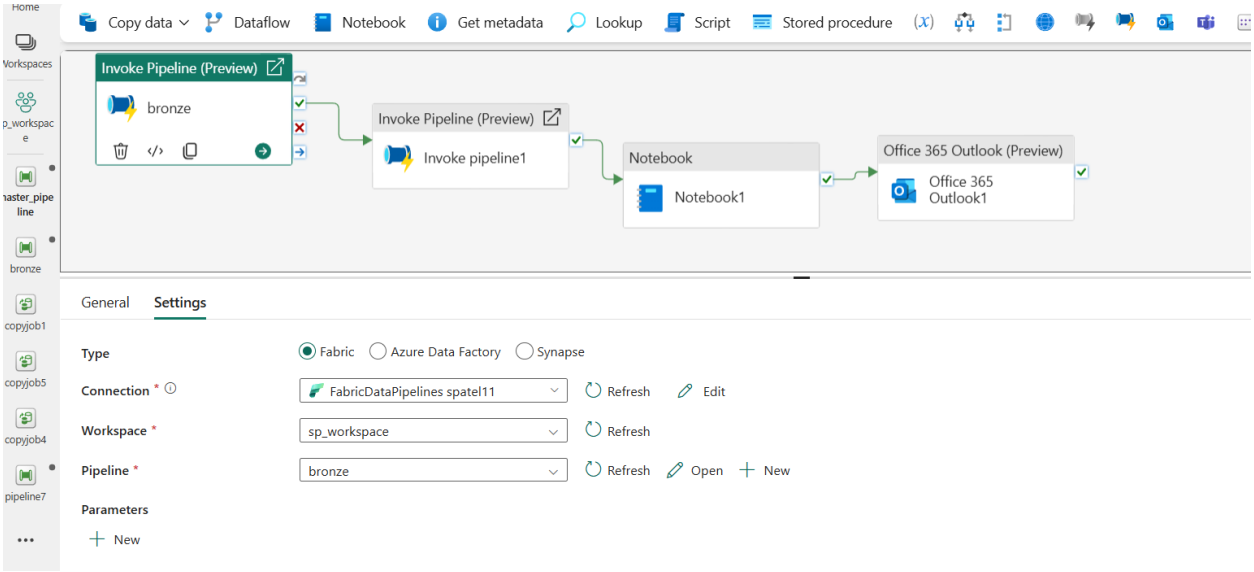
Table view

Download | Search

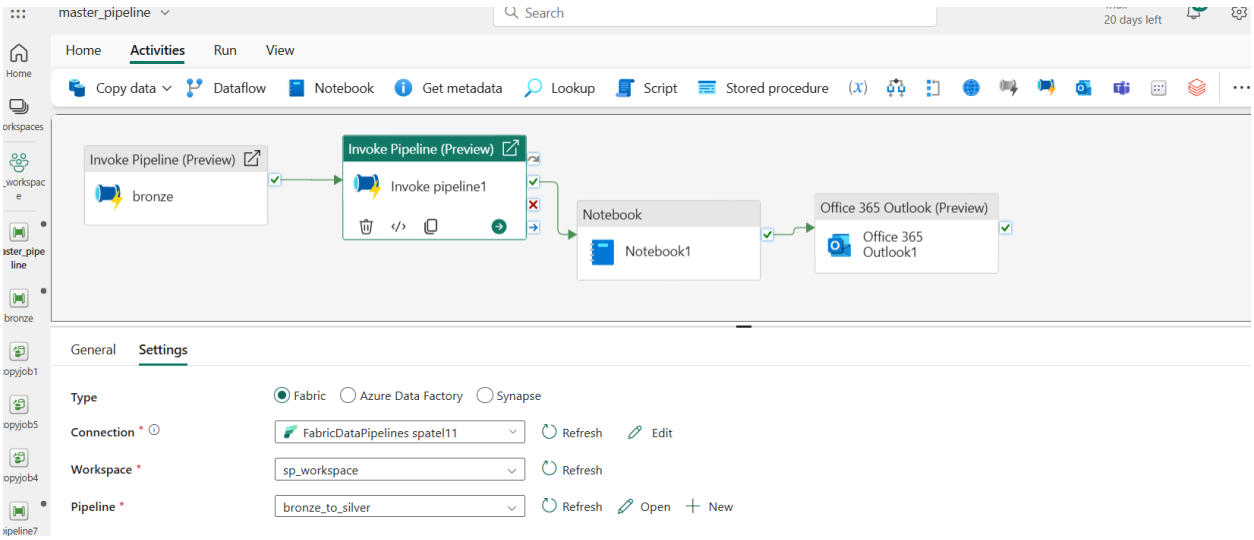
	transaction_id	account_id	transaction_date	transaction_type	transaction_amount	hash_key	created_by	created_date	updated_by	updated
1	1	45	2024-01-01	Deposit	100.5	3754156198	fabric	2025-05-11 18:5...	fabric	2025-05-
2	2	12	2024-01-02	Withdrawal	200.75	4058931298	fabric	2025-05-11 18:5...	fabric	2025-05-
3	3	78	2024-01-03	Deposit	150.0	3749805095	fabric	2025-05-11 18:5...	fabric	2025-05-
4	4	34	2024-01-04	Withdrawal	300.25	4695379	fabric	2025-05-11 18:5...	fabric	2025-05-
5	5	56	2024-01-05	Deposit	250.0	2060751710	fabric	2025-05-11 18:5...	fabric	2025-05-
6	6	23	2024-01-06	Withdrawal	175.0	1019709028	fabric	2025-05-11 18:5...	fabric	2025-05-
7	7	89	2024-01-07	Deposit	225.5	495552080	fabric	2025-05-11 18:5...	fabric	2025-05-
8	8	67	2024-01-08	Withdrawal	275.75	3746532401	fabric	2025-05-11 18:5...	fabric	2025-05-
9	9	14	2024-01-09	Deposit	325.0	2562028586	fabric	2025-05-11 18:5...	fabric	2025-05-
10	10	92	2024-01-10	Withdrawal	375.25	83465714	fabric	2025-05-11 18:5...	fabric	2025-05-
11	11	3	2024-01-11	Deposit	100.5	2345856361	fabric	2025-05-11 18:5...	fabric	2025-05-
12	12	81	2024-01-12	Withdrawal	200.75	1292421933	fabric	2025-05-11 18:5...	fabric	2025-05-

Master Pipeline

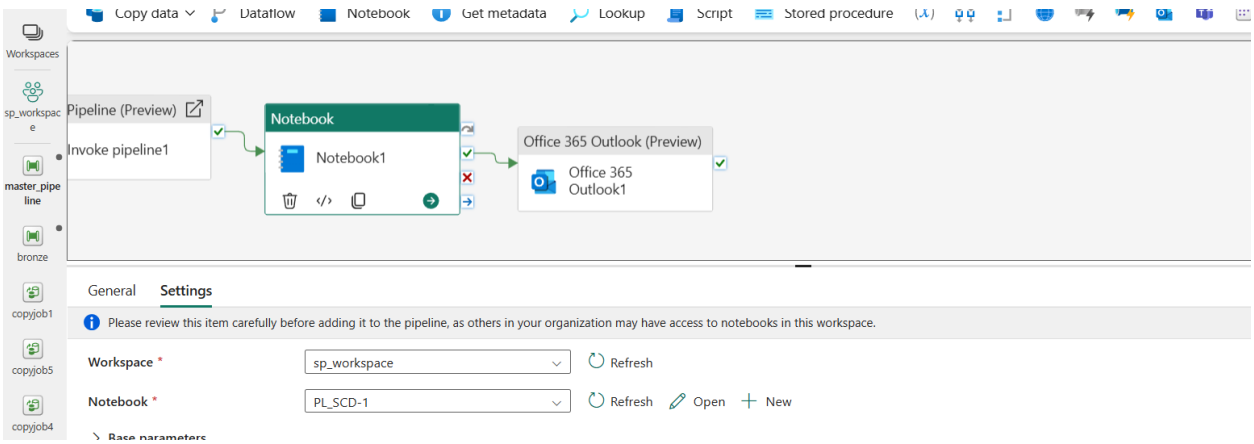
1.Bronze:- select bronze layer pipeline



2. Select Bronze to silver layer dataflow pipeline.



3.Notebook: select SCD type-1 logic code notebook



4.Office 365 outlook :- use to get mail if pipeline is successful.