

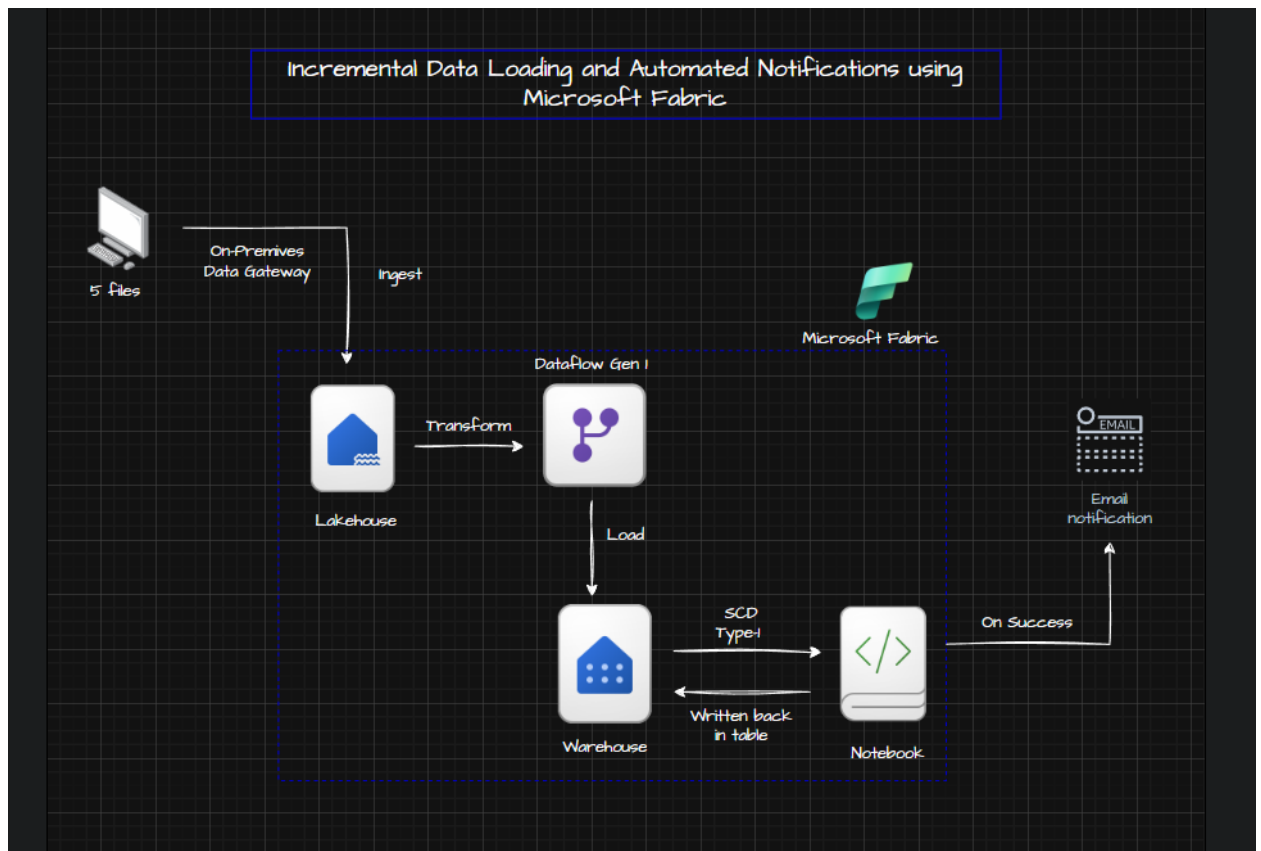
Project 4 – Dynamic Data Ingestion from On-Premises to Microsoft Fabric Lakehouse

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

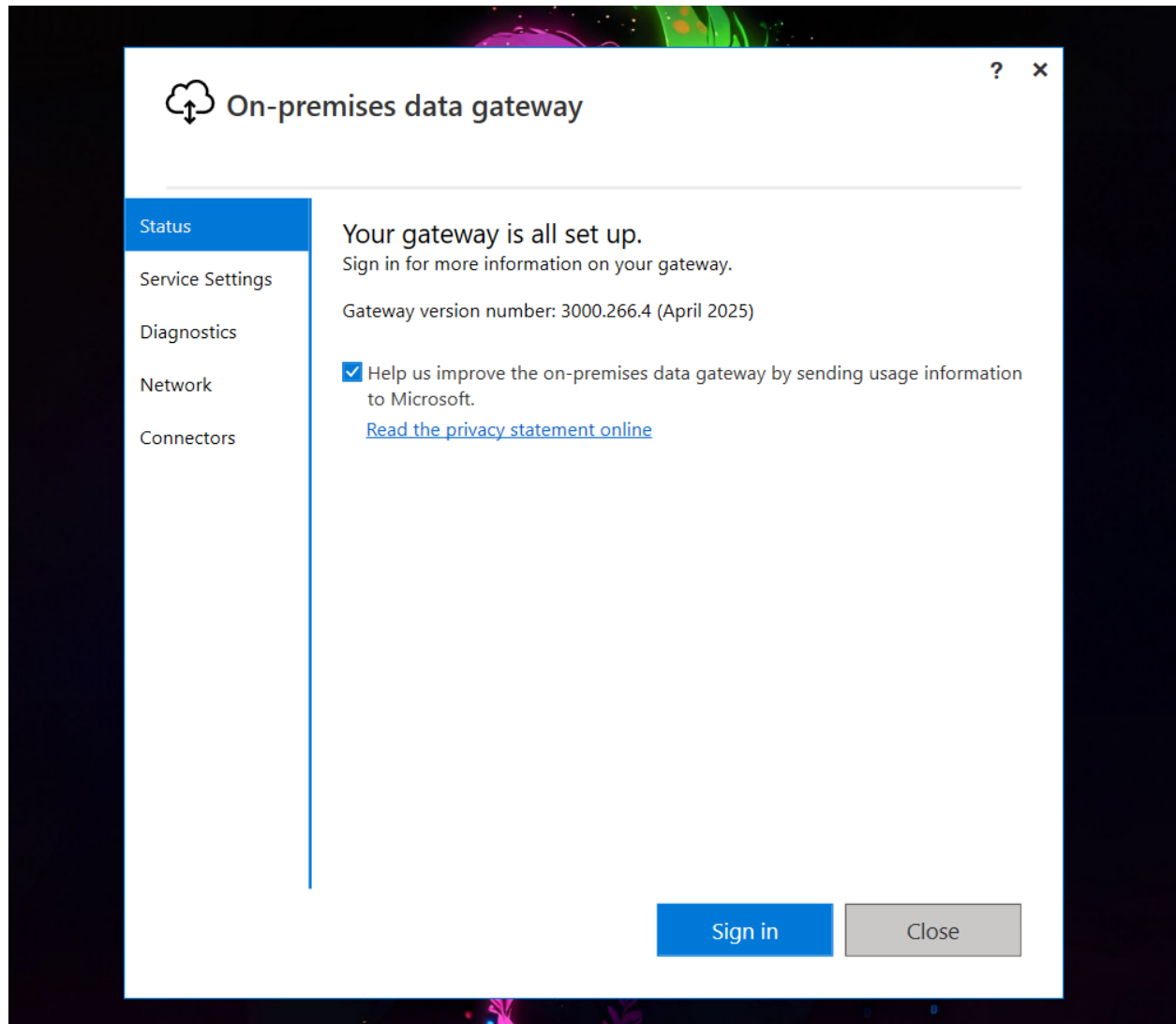
- **Location:** [AI Bank Dataset](#)
- **Files:**
 - accounts.csv
 - customers.csv
 - loans.csv
 - loan_payments.csv
 - transactions.csv
- Architecture Diagram



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

Manage Connections and Gateways

Connections **On-premises data gateways** Virtual network data gateways Azure Key Vault references

The data gateway acts as a bridge, providing quick and secure data transfer between on-premises data and Power BI, Microsoft Flow, Logic Apps, and PowerApps. [Learn more in this overview.](#)

Name ↑	Contact info	Users	Status	Gateways
newgateway	spatel11@laurentian.ca	Sapana	Online	1

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.

bronze ▾ Search

Home Activities Run View

Validate Run Schedule Add trigger View run history Copy data Data

Get Metadata

Get Metadata1

ForEach

ForEach1

Activities

Copy data1

General Settings

Connection * Icssystem Refresh Test connection Edit

File path sourcefiles / File name Browse Preview data

File format * DelimitedText Settings

Field list * + New | Delete

☐ Argument

☐ Child items

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).

The screenshot displays the Azure Data Factory (ADF) interface. The main canvas shows a workflow with a 'Get Metadata' activity followed by a 'ForEach' activity. The 'ForEach' activity is expanded, showing its configuration. The 'Items' field is set to '@activity('Get Metadata1').output.childItems'. The 'Batch count' is set to 1. The 'Settings' tab is active, showing the 'Sequential' checkbox and the 'Batch count' field. The 'Items' field is highlighted with a green border. On the right, the 'Pipeline expression builder' is open, showing the expression '@activity('Get Metadata1').output.childItems'. Below the expression builder, the 'Activity outputs' tab is active, listing the outputs of the 'Copy data1' activity: 'Copy data1 activity output', 'Get Metadata1 activity output', 'Get Metadata1 childItems', and 'Get Metadata1 created'.

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

Main canvas > ForEach1

The screenshot displays the Azure Data Factory (ADF) interface. The main canvas shows a workflow with a 'Copy data' activity. The 'Copy data' activity is expanded, showing its configuration. The 'Source' tab is active, showing the 'Connection' dropdown set to 'lcsystem', the 'File path type' set to 'File path', the 'File path' set to 'Directory / @item().Name', the 'Recursively' checkbox checked, and the 'File format' set to 'DelimitedText'. The 'Settings' tab is also visible, showing the 'Advanced' section.

Parameterized dataset for File Path and File Name.

Main canvas > ForEach1

Copy data

Copy data1

General

Source

Destination

Mapping

Settings

Connection *

sp_lakehouse

Refresh

Open

Root folder

☒ Tables ☐ Files

Table

@concat(replace(item)?['Name'],'.cs...

Table action

☐ Append ☒ Overwrite

> Advanced

Each file loaded to Lakehouse.

sp_lakehouse

Search

Trial: 21 days left

Home

Get data

New semantic model

Open notebook

Manage OneLake data access (preview)

Home

Workspaces

OneLake catalog

sp_workspace

sp_lakehouse

pipeline6

copyjob3

sp_lakehouse

Explorer

Search tables

sp_lakehouse

Tables

accounts

customers

loan_payments

loans

Merge_alltb

new_cust

project

sales_data

transactions

customers

Showing 87 rows

	customer_id	first_name	last_name	address	city	state	zip
1	1	John	Doe	123 Elm St	Toronto	ON	M4B1B3
2	2	Jane	Smith	456 Maple Ave	Ottawa	ON	K1A0B1
3	3	Michael	Johnson	789 Oak Dr	Montreal	QC	H1A1A1
4	4	Emily	Davis	101 Pine Rd	Calgary	AB	T2A0A1
5	5	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1
6	6	Emma	Clark	505 Cedar St	Halifax	NS	B3H0A1
7	7	James	Martinez	606 Spruce Ln	Winnipeg	MB	R3C0A1
8	8	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1
9	9	William	Lopez	808 Redwood Dr	Victoria	BC	V8W0A1
10	10	Ava	Anderson	909 Cypress Ave	Quebec City	QC	G1A0A1
11	11	Alexander	Thomas	1010 Willow Rd	St. John's	NL	A1A0A1
12	12	Isabella	Lee	1111 Poplar St	Fredericton	NB	E3B0A1
13	13	Daniel	Harris	1212 Ash Blvd	Charlottetown	PE	C1A0A1
14	14	Sophia	Young	1313 Beech Dr	Yellowknife	NT	X1A0A1

loan_payments

Showing 100 rows

	payment_id	loan_id	payment_date	payment_amount
1	1	45	1/1/2024 12:00:00 AM	100
2	2	23	1/2/2024 12:00:00 AM	150
3	3	67	1/3/2024 12:00:00 AM	200
4	4	89	1/4/2024 12:00:00 AM	250
5	5	12	1/5/2024 12:00:00 AM	300
6	6	34	1/6/2024 12:00:00 AM	350
7	7	56	1/7/2024 12:00:00 AM	400
8	8	78	1/8/2024 12:00:00 AM	450
9	9	90	1/9/2024 12:00:00 AM	500
10	10	11	1/10/2024 12:00:00 AM	550
11	11	22	1/11/2024 12:00:00 AM	600
12	12	33	1/12/2024 12:00:00 AM	650
13	13	44	1/13/2024 12:00:00 AM	700
14	14	55	1/14/2024 12:00:00 AM	750
15	15	66	1/15/2024 12:00:00 AM	800
16	16	77	1/16/2024 12:00:00 AM	850
17	17	88	1/17/2024 12:00:00 AM	900

sp_lakehouse

Home

Get data New semantic model Open notebook Manage OneLake data access (preview)

Explorer

Search tables

sp_lakehouse

- Tables
 - accounts
 - customers
 - loan_payments
 - loans
 - Merge_alltb
 - new_cust
 - project
 - sales_data
 - transactions
- Files
 - container1
 - db-sp1605
 - dbo

loans

Showing 100 rows

	loan_id	customer_id	loan_amount	interest_rate	loan_term
1	1	45	10000.5	5.5	36
2	2	12	20000.75	4.5	48
3	3	78	15000	6	60
4	4	34	30000.25	3.5	24
5	5	56	25000	5	36
6	6	23	17500.5	4	48
7	7	89	22500.75	6.5	60
8	8	67	27500	3	24
9	9	14	32500.25	5.5	36
10	10	92	37500.5	4.5	48
11	11	3	10000.75	6	60
12	12	81	20000	3.5	24
13	13	29	15000.25	5	36
14	14	64	30000.5	4	48
15	15	47	25000.75	6.5	60
16	16	18	17500	3	24
17	17	99	22500.25	5.5	36
18	18	5	27500.5	4.5	48

Succeeded (2 sec 165 ms)

Columns 5 Rows 100

sp_lakehouse

Home

Get data New semantic model Open notebook Manage OneLake data access (preview)

Explorer

Search tables

sp_lakehouse

- Tables
 - accounts
 - customers
 - loan_payments
 - loans
 - Merge_alltb
 - new_cust
 - project
 - sales_data
 - transactions
- Files
 - container1
 - db-sp1605
 - dbo

transactions

Showing 100 rows

	transaction_id	account_id	transaction_date	transaction_amount	transaction_type
1	1	45	1/1/2024 12:00:00 AM	100.5	Deposit
2	2	12	1/2/2024 12:00:00 AM	200.75	Withdrawal
3	3	78	1/3/2024 12:00:00 AM	150	Deposit
4	4	34	1/4/2024 12:00:00 AM	300.25	Withdrawal
5	5	56	1/5/2024 12:00:00 AM	250	Deposit
6	6	23	1/6/2024 12:00:00 AM	175	Withdrawal
7	7	89	1/7/2024 12:00:00 AM	225.5	Deposit
8	8	67	1/8/2024 12:00:00 AM	275.75	Withdrawal
9	9	14	1/9/2024 12:00:00 AM	325	Deposit
10	10	92	1/10/2024 12:00:00 AM	375.25	Withdrawal
11	11	3	1/11/2024 12:00:00 AM	100.5	Deposit
12	12	81	1/12/2024 12:00:00 AM	200.75	Withdrawal
13	13	29	1/13/2024 12:00:00 AM	150	Deposit
14	14	64	1/14/2024 12:00:00 AM	300.25	Withdrawal
15	15	47	1/15/2024 12:00:00 AM	250	Deposit
16	16	18	1/16/2024 12:00:00 AM	175	Withdrawal
17	17	99	1/17/2024 12:00:00 AM	225.5	Deposit
18	18	5	1/18/2024 12:00:00 AM	275.75	Withdrawal

Succeeded (2 sec 15 ms)

Columns 5 Rows 100

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.

Dataflow 3

Power Query Dataflow 3 Dataflow saved

Get data

Connect to data source

Lakehouse
Microsoft Fabric

Connection credentials

Connection
Lakehouse (none)

Authentication kind: Organizational account [Edit connection](#)

Power Query - Dataflow 3 - Dataflow saved

Get data - Choose data

Search

Display options

- ☐ sys.sys_dw_checkpoint...
- ☐ sys.sys_dw_manifest_fil...
- ☐ sys.sys_dw_physical_ta...
- ☐ sys.sys_dw_physical_ta...
- ☐ sys.sys_dw_schemas
- ☒ accounts
- ☒ customers
- ☒ loan_payments
- ☒ loans
- ☐ Merge_allb
- ☐ new_cust
- ☐ project
- ☐ sales_data
- ☒ transactions

1:2 transaction_id	1:3 account_id	1:4 transaction_date	1:2 transaction_amount	1:6 transaction_type
1	45	1/1/2024	100.5	Deposit
2	12	1/2/2024	200.75	Withdrawal
3	78	1/3/2024	150	Deposit
4	34	1/4/2024	300.25	Withdrawal
5	56	1/5/2024	250	Deposit
6	23	1/6/2024	175	Withdrawal
7	89	1/7/2024	225.5	Deposit
8	67	1/8/2024	275.75	Withdrawal
9	14	1/9/2024	325	Deposit
10	92	1/10/2024	375.25	Withdrawal
11	3	1/11/2024	100.5	Deposit
12	81	1/12/2024	200.75	Withdrawal
13	29	1/13/2024	150	Deposit
14	64	1/14/2024	300.25	Withdrawal
15	47	1/15/2024	250	Deposit
16	18	1/16/2024	175	Withdrawal
17	99	1/17/2024	225.5	Deposit

Power Query - Dataflow 3 - Dataflow saved

Home Transform Add column View Help

Queries [5]

- accounts (4 steps)
- customers (4 steps)

```

4 #"Navigation 2" = #"Navigation 1"[[lakehouseId = "4aaa72e8-fef8-4f3a-80b4-fbc015d8c10b"]][Data],
5 #"Navigation 3" = #"Navigation 2"[[Id = "accounts", ItemKind = "Table"]][Data]
6 In
7 #"Navigation 3"

```

1:2 account_id	1:3 customer_id	1:4 account_type	1:2 balance
1	1	45 Savings	1000.5
2	2	12 Checking	2500.75
3	3	78 Savings	1500
4	4	34 Checking	3000.25
5	5	56 Savings	500
6	6	23 Checking	1200.5
7	7	89 Savings	800.75
8	8	67 Checking	2200

Completed (2.14 s) Columns: 4 Rows: 99+

Query settings

Properties

Name: accounts

Applied steps

- Source
- Navigation 1
- Navigation 2
- Navigation 3

Data destination: No data destination

Explanation of data filtering process for Account table data :-

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

Power Query - Dataflow 3 - Dataflow saved

Home Transform Add column View Help

Queries [5]

- accounts (4 steps)
- customers (4 steps)
- loan_payments
- loans
- transactions

Remove top rows

Remove bottom rows

Remove alternate rows

Remove duplicates

Remove blank rows

Remove errors

```

1 let
2 Source = Lakehouse.Contents([],
3 #"Navigation 1" = Source[[workspaceId = "406bb468-7d95-48da-8f90-1608eb9ee00b"]][Data],
4 #"Navigation 2" = #"Navigation 1"[[lakehouseId = "4aaa72e8-fef8-4f3a-80b4-fbc015d8c10b"]][Data],

```

1:2 account_id	1:3 customer_id	1:4 account_type	1:2 balance
1	1	45 Savings	1000.5
2	2	12 Checking	2500.75
3	3	78 Savings	1500
4	4	34 Checking	3000.25
5	5	56 Savings	500
6	6	23 Checking	1200.5
7	7	89 Savings	800.75
8	8	67 Checking	2200

Completed (0.76 s) Columns: 4 Rows: 99+

Query settings

Properties

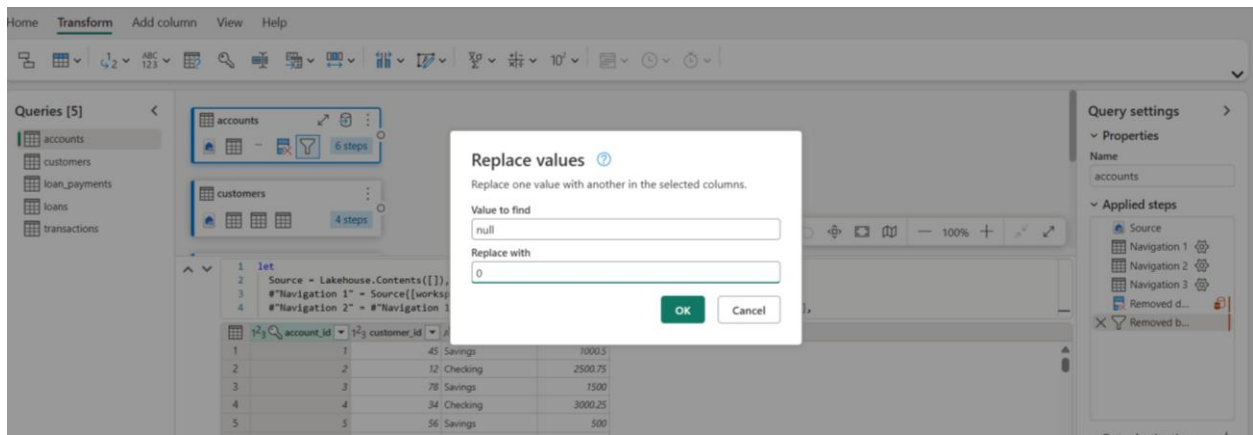
Name: accounts

Applied steps

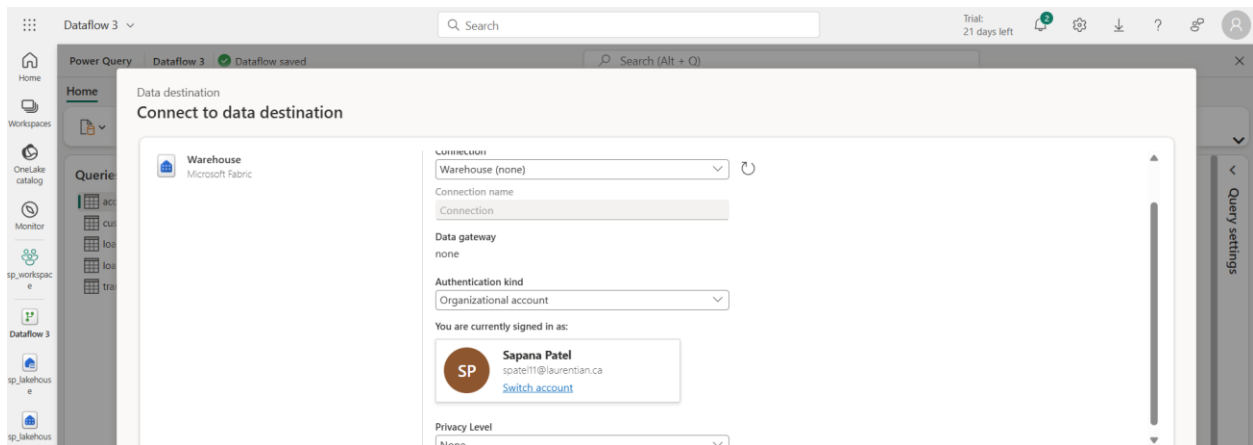
- Source
- Navigation 1
- Navigation 2
- Navigation 3
- Removed d...
- Removed b...
- Replaced v...

Data destination: No data destination

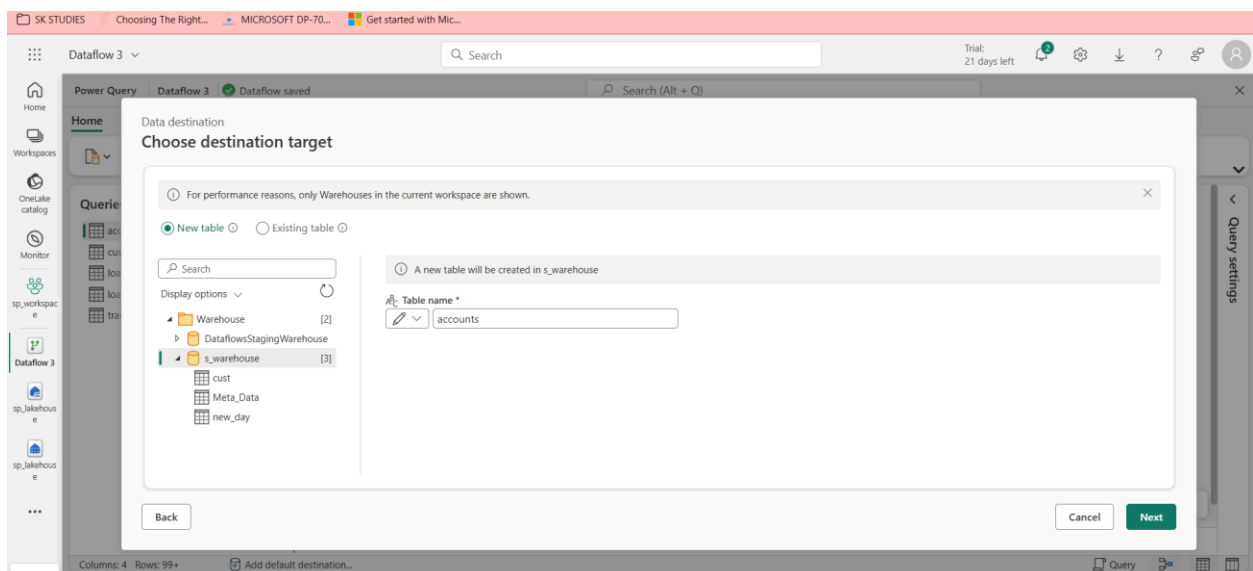
Select replace values to remove null values from data. Select null == 0 .

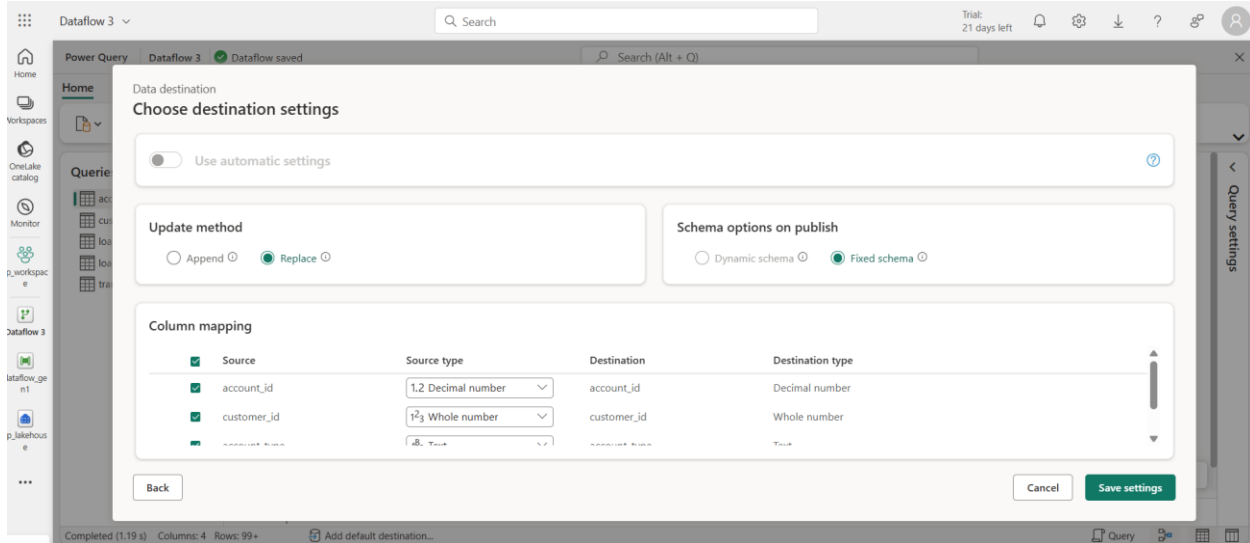


Select destination as a warehouse.

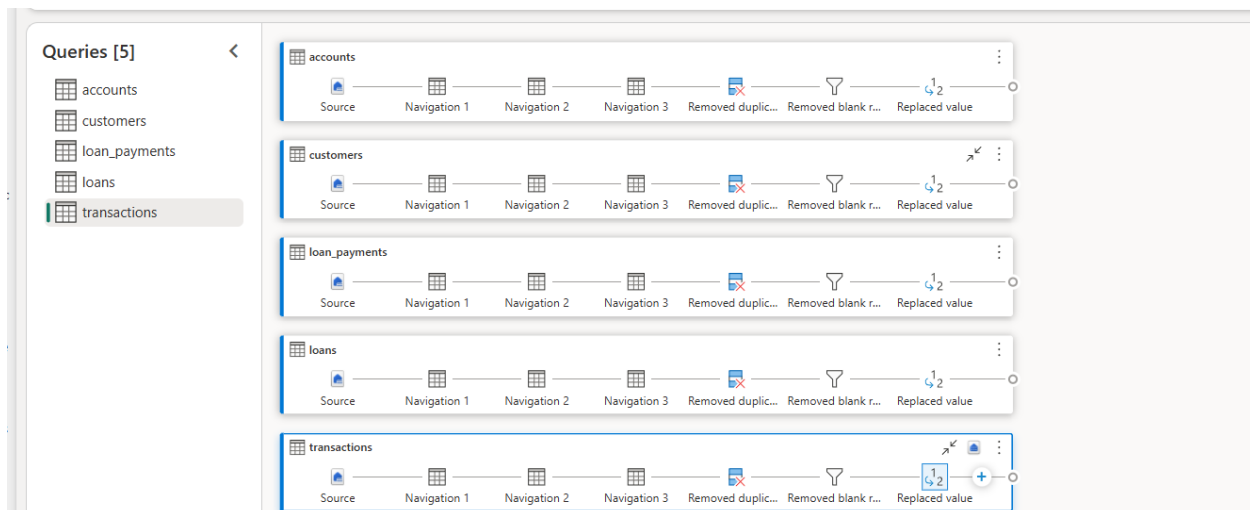


Select the new table in warehouse.





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



We got all tables in warehouse.

s_warehouse

Home Reporting Help

Get data New SQL query SQL templates Query activity Model layouts Download SQL database project Copilot

This warehouse has a default Power BI semantic model. To automatically add objects, go to warehouse settings. To manually add objects, use Manage default semantic model. Learn more

Explorer

- Warehouses
 - s_warehouse
 - Schemas
 - dbo
 - Tables
 - accounts
 - cust
 - customers
 - loan_payme
 - loans
 - Meta_Data

Data preview - accounts

Showing 1000 rows

	12F account_id	123 customer_id	ABC account_type	12F balance
1	1	45	Savings	1000.5
2	2	12	Checking	2500.75
3	3	78	Savings	1500
4	4	34	Checking	3000.25
5	5	56	Savings	500
6	6	23	Checking	1200.5
7	7	89	Savings	800.75
8	8	67	Checking	2200
9	9	14	Savings	900.25
10	10	92	Checking	1800.5
11	11	3	Savings	1100.75
12	12	81	Checking	2700
13	13	29	Savings	1300.25

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

1. Import Required Libraries

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

2. Load Source Data from Warehouse

```
1 df = spark.read.synapsesql("s_warehouse.dbo.accounts")
2 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

```
1 # Define Delta table creation query
2 create_table_query = """
3 CREATE TABLE IF NOT EXISTS Accounts_Gold (
4     account_id INT,
5     customer_id INT,
6     account_type STRING,
7     balance FLOAT,
8     hash_key BIGINT,
9     created_by STRING,
10    created_date TIMESTAMP,
11    updated_by STRING,
12    updated_date TIMESTAMP
13 )
14 USING DELTA
15 LOCATION 'Files/Gold_layer/Accounts_Gold'
16 """
17
18 # Execute the table creation
19 spark.sql(create_table_query)
20
21 #spark.sql("DROP TABLE IF EXISTS Customers_Gold")
22
23
24
```

- 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

```
1  from pyspark.sql.functions import col, lit, current_timestamp, crc32, concat_ws
2  from delta.tables import DeltaTable
3
4  # Load source data from Warehouse Silver Layer
5  source_df = spark.read.synapsesql("s_warehouse.dbo.accounts")
6
7  # Add metadata and Hash Key
8  source_df = source_df \
9      .withColumn("hash_key", crc32(concat_ws("||", *source_df.columns))) \
10     .withColumn("created_date", current_timestamp()) \
11     .withColumn("updated_date", current_timestamp()) \
12     .withColumn("created_by", lit("fabric")) \
13     .withColumn("updated_by", lit("fabric"))
```

5. Load Target Delta Table

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

6. Identify New or Updated Records

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

7. Apply SCD-1 Merge Logic

```
1  # Apply SCD-1 Merge Logic to Gold Layer Delta Table
2  target_delta.alias("tgt").merge(
3      df_diff.alias("src"),
4      "tgt.account_id = src.account_id"
5  ) \
6  .whenMatchedUpdate(set={
7      "account_id": "src.account_id",
8      "customer_id": "src.customer_id",
9      "account_type": "src.account_type",
10     "balance": "src.balance",
11     "hash_key": "src.hash_key",
12     "updated_date": "current_timestamp()",
13     "updated_by": lit("fabric-updated")
14 }) \
15 .whenNotMatchedInsert(values={
16     "account_id": "src.account_id",
17     "customer_id": "src.customer_id",
18     "account_type": "src.account_type",
19     "balance": "src.balance",
20     "hash_key": "src.hash_key",
21     "created_date": "current_timestamp()",
22     "created_by": lit("fabric"),
23     "updated_date": "current_timestamp()",
24     "updated_by": lit("fabric")
25 }).execute()
26
27 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 view

	123 account_id	123 customer_id	ABC account_type	12F balance	12L hash_key	ABC created_by	ABC created_date	ABC updated_by	ABC 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 view

	123 customer_id	ABC first_name	ABC last_name	ABC address	ABC city	ABC state	ABC zip	12L hash_key	ABC created_by	ABC created_date	ABC updated_by	ABC 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 view

	123 loan_id	123 customer_id	ABC loan_term	12F loan_amount	12F interest_rate	12L hash_key	ABC created_by	ABC created_date	ABC updated_by	ABC 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

9 colour

Table view

Download

Se

	123 payment_id	123 loan_id	ABC payment_date	12F payment_amount	12L hash_key	ABC created_by	ABC created_date	ABC updated_by	ABC updated_date	
1	1	45	2024-01-01	100.0	2737019449	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
2	2	23	2024-01-02	150.0	3509823974	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
3	3	67	2024-01-03	200.0	3219615906	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
4	4	89	2024-01-04	250.0	577236198	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
5	5	12	2024-01-05	300.0	1196719864	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
6	6	34	2024-01-06	350.0	1361610787	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
7	7	56	2024-01-07	400.0	3108358108	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
8	8	78	2024-01-08	450.0	3679801302	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
9	9	90	2024-01-09	500.0	1747851957	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
10	10	11	2024-01-10	550.0	1419562063	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	
11	11	22	2024-01-11	600.0	3949794927	abric	2025-05-11 18:5...	abric	2025-05-11 18:56...	

Table view

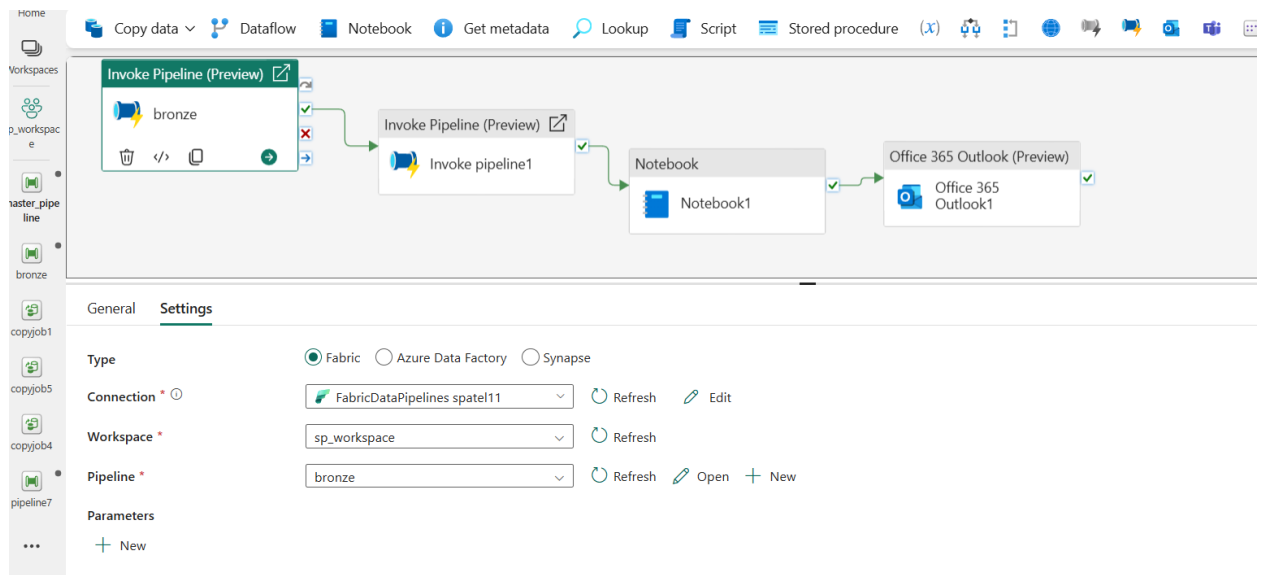
Download

Search

	123 transaction_id	123 account_id	ABC transaction_date	ABC transaction_type	12F transaction_amount	12L hash_key	ABC created_by	ABC created_date	ABC updated_by	ABC updated
1	1	45	2024-01-01	Deposit	100.5	3754156198	abric	2025-05-11 18:5...	abric	2025-05-
2	2	12	2024-01-02	Withdrawal	200.75	4058931298	abric	2025-05-11 18:5...	abric	2025-05-
3	3	78	2024-01-03	Deposit	150.0	3749805095	abric	2025-05-11 18:5...	abric	2025-05-
4	4	34	2024-01-04	Withdrawal	300.25	4695379	abric	2025-05-11 18:5...	abric	2025-05-
5	5	56	2024-01-05	Deposit	250.0	2060751710	abric	2025-05-11 18:5...	abric	2025-05-
6	6	23	2024-01-06	Withdrawal	175.0	1019709028	abric	2025-05-11 18:5...	abric	2025-05-
7	7	89	2024-01-07	Deposit	225.5	495552080	abric	2025-05-11 18:5...	abric	2025-05-
8	8	67	2024-01-08	Withdrawal	275.75	3746532401	abric	2025-05-11 18:5...	abric	2025-05-
9	9	14	2024-01-09	Deposit	325.0	2562028586	abric	2025-05-11 18:5...	abric	2025-05-
10	10	92	2024-01-10	Withdrawal	375.25	83465714	abric	2025-05-11 18:5...	abric	2025-05-
11	11	3	2024-01-11	Deposit	100.5	2345856361	abric	2025-05-11 18:5...	abric	2025-05-
12	12	81	2024-01-12	Withdrawal	200.75	1292421933	abric	2025-05-11 18:5...	abric	2025-05-

Master Pipeline

1.Bronze:- select bronze layer pipeline



2. Select Bronze to silver layer dataflow pipeline.

The screenshot shows the 'Invoke Pipeline (Preview)' settings for the 'bronze' pipeline. The 'General' tab is active, displaying the following configuration:

- Type:** Fabric (selected), Azure Data Factory, Synapse
- Connection:** FabricDataPipelines spatel11 (with Refresh and Edit buttons)
- Workspace:** sp_workspace (with Refresh button)
- Pipeline:** bronze_to_silver (with Refresh, Open, and New buttons)

The pipeline diagram above shows the flow: Invoke Pipeline (Preview) -> Invoke pipeline1 -> Notebook1 -> Office 365 Outlook (Preview).

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

The screenshot shows the 'Notebook' settings for the 'Notebook1' notebook. The 'General' tab is active, displaying the following configuration:

- Workspace:** sp_workspace (with Refresh button)
- Notebook:** PL_SCD-1 (with Refresh, Open, and New buttons)

The pipeline diagram above shows the flow: Invoke pipeline1 -> Notebook1 -> Office 365 Outlook (Preview).

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

The screenshot shows the 'Office 365 Outlook (Preview)' settings for the 'Office 365 Outlook1' action. The 'General' tab is active, displaying the following configuration:

- Signed in as:** spatel11@laurentian.ca (with Change account button)
- To:** patelsk9709@gmail.com
- Subject:** Pipeline status
- Body:** Hey Team ,
Pipeline was successful.

The pipeline diagram above shows the flow: Invoke pipeline1 -> Notebook1 -> Office 365 Outlook (Preview).