



Incremental Data Loading and Automated Notifications using Microsoft Fabric

—Sudhanshu Kharbanda

Introduction

Modern organizations rely on timely and accurate data to drive decision-making, yet integrating information from diverse on-premises sources into centralized analytics platforms remains a challenge. Issues such as data quality, transformation consistency, and the need for automation highlight the importance of efficient end-to-end data pipelines. Microsoft Fabric provides a unified environment that addresses these challenges by combining ingestion, storage, transformation, and monitoring capabilities within a single platform.

This project demonstrates the development of an automated data pipeline on Microsoft Fabric using the AI Bank Dataset. The pipeline securely ingests structured data through the On-Prem Gateway, applies cleansing and transformations with Dataflow Gen1, and loads the results into a Fabric Warehouse. Slowly Changing Dimension (SCD) Type 1 logic is implemented via Fabric Notebooks to maintain data consistency, while automated scheduling and email notifications ensure reliable monitoring and stakeholder updates. The solution highlights Fabric's ability to streamline data workflows and deliver scalable, analytics-ready datasets.

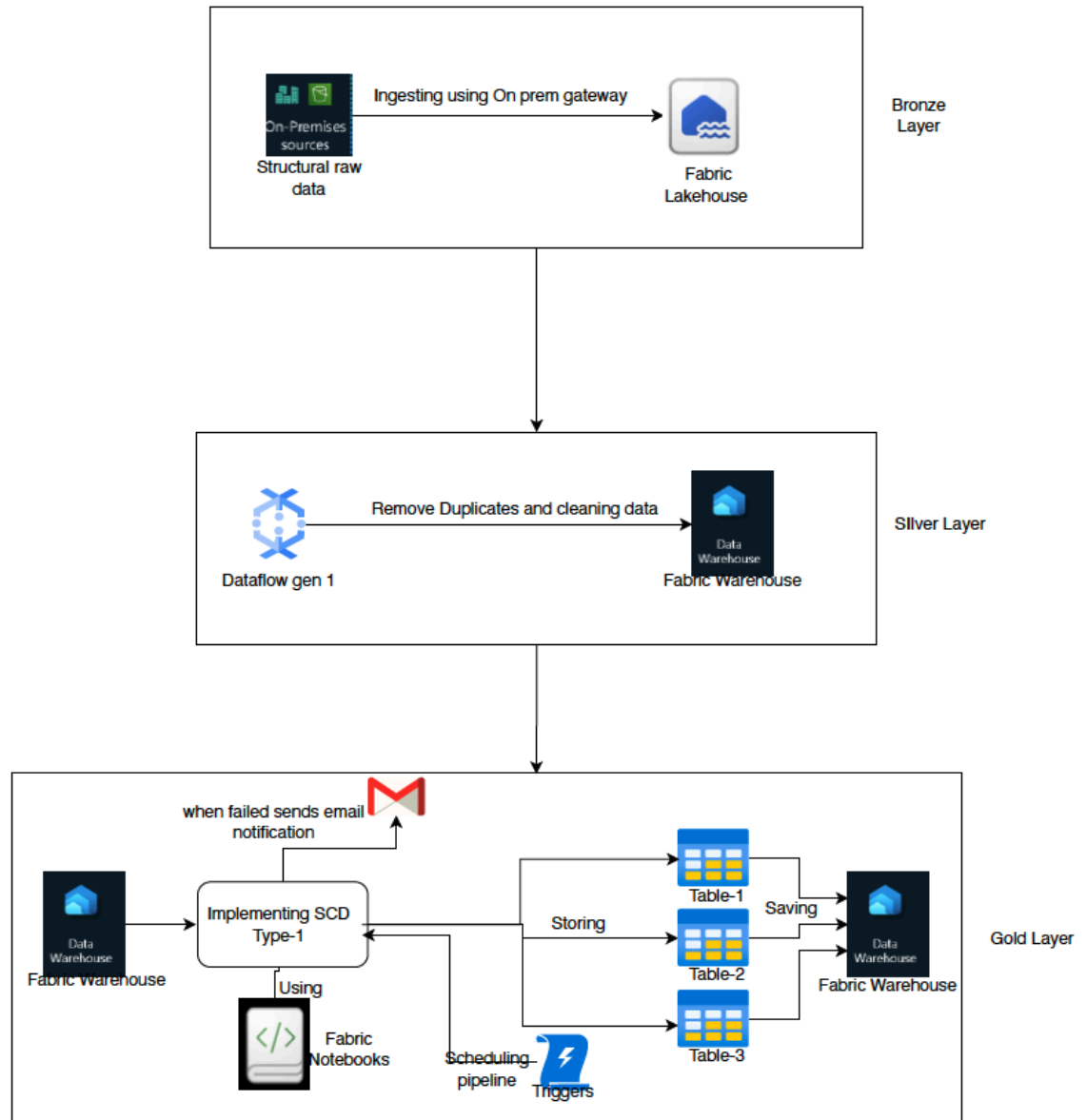
Specifications

The primary objective of this project is to design and implement an end-to-end automated data pipeline on Microsoft Fabric that demonstrates secure ingestion, transformation, and monitoring of structured data. Specifically, the project aims to:

1. Ingest structured data from on-premises environments into a Fabric Lakehouse using the On-Prem Gateway.
2. Utilize the AI Bank Dataset as the source for pipeline implementation.
3. Apply data cleansing operations—such as joins, duplicate removal, and formatting—using Dataflow Gen1.
4. Load the cleansed data into a Fabric Warehouse for downstream analytics.
5. Implement Slowly Changing Dimension (SCD) Type 1 logic with Fabric Notebooks to maintain data consistency in dimension tables.

- Schedule and monitor the pipeline while enabling automated email notifications to fail execution of the pipeline.

Architecture Diagram



Click on the entraID for provide the permission for enabling fabric for the account.

Home > Default Directory | Overview >

Users

Default Directory

+ New user Edit Delete Download users (Preview) Bulk operations Refresh Manage view Per-user MFA Got feedback?

Azure Active Directory is now Microsoft Entra ID

Search Add filter

3 users found (1 user selected)

| <input type="checkbox"/> | Display name | User principal name | User type | On-premises sy... | Identities | Company name | Creation type |
|-------------------------------------|--------------|---------------------|-----------|-------------------|---------------------------|--------------|---------------|
| <input type="checkbox"/> | Roman Syed | roman7816@jyoti... | Member | No | jyotikharbanda233gmail... | | |
| <input checked="" type="checkbox"/> | Sudhanshu | sudkhar@jyotikha... | Member | No | jyotikharbanda233gmail... | | |
| <input type="checkbox"/> | Syed Hasan | jyotikharbanda23... | Member | No | MicrosoftAccount | | |

All users Audit logs Sign-in logs Diagnose and solve problems Deleted users Password reset User settings Bulk operation results Bulk operation results (Preview) New support request

Click on the assigned roles and add the fabric administrator role to the user

Home > Default Directory | Overview > Users >

Sudhanshu


User


Search Edit properties Delete Refresh Reset password Revoke sessions Manage view Got feedback?

Overview Audit logs Sign-in logs Diagnose and solve problems Custom security attributes Assigned roles Administrative units Groups Applications Licenses Devices Azure role assignments Authentication methods New support request

Overview Monitoring Properties


Basic info


 **Sudhanshu**
sudkhar@jyotikharbanda233gmail.onmicrosoft.com
Member



| | | | |
|---------------------|--|-------------------|---|
| User principal name | sudkhar@jyotikharbanda233gmail.onmicrosoft.com | Group memberships | 0 |
| Object ID | 888da4b5-1075-45cb-a9a4-ac258ba027f6 | Applications | 0 |
| Created date time | Aug 23, 2025, 10:35 PM | Assigned roles | 1 |
| User type | Member | Assigned licenses | 0 |
| Identities | jyotikharbanda233gmail.onmicrosoft.com | | |
| Agent ID | | | |

My Feed

 **Account status**
Enabled
[Edit](#)

 **B2B invitation**
[Convert to external user](#)

Quick actions

Go to Azure and search for the fabric capacity. And create the fabric capacity.

[Home](#) > [sudhanshuboot](#) > [Marketplace](#) > [Microsoft Fabric](#) >

Create Fabric capacity

Welcome to Microsoft Fabric

Fabric delivers an end-to-end analytics platform from the data lake to the business user.

[Find out more](#)



*** Basics** Tags Review + create

Create Fabric capacity that you can use with your Fabric workspaces.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

| | |
|------------------|---|
| Subscription * | <input type="text" value="Azure subscription 1"/> |
| Resource group * | <input type="text" value="sudhanshuboot"/> |

[Create new](#)

Capacity details

Name your Capacity and select a location.

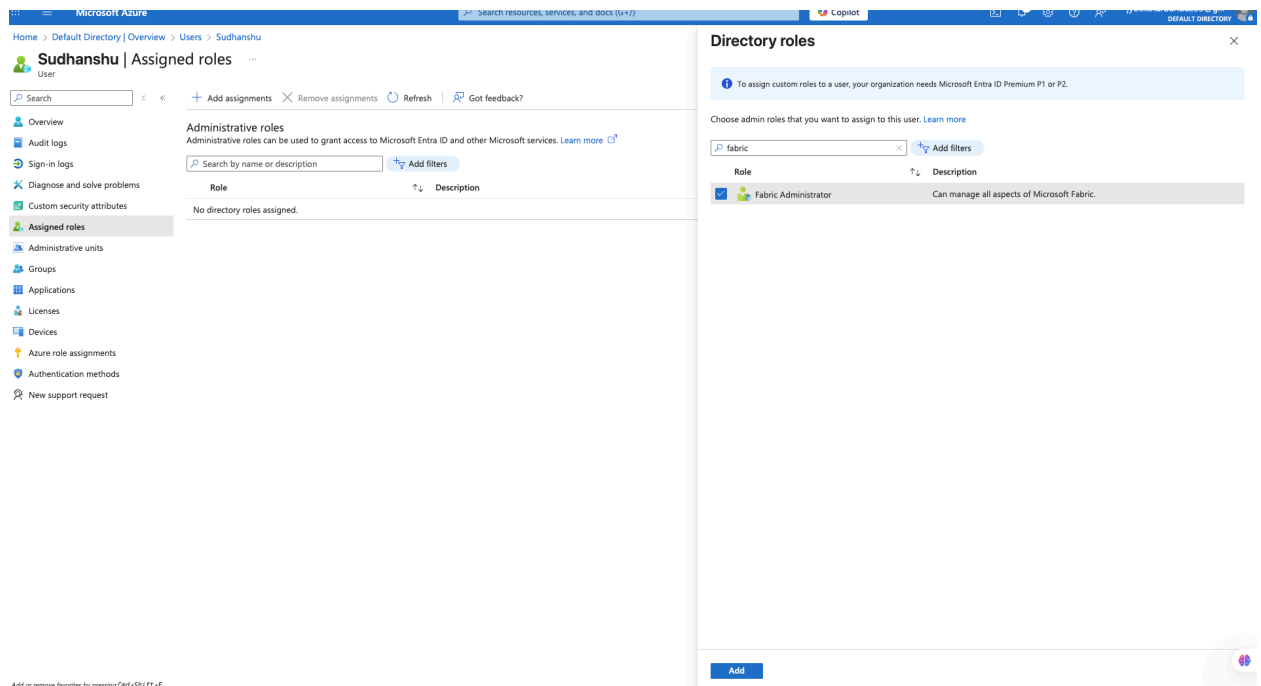
| | |
|---------------------------------|---|
| Capacity name * | <input type="text" value="sudfabric"/> |
| Region * | <input type="text" value="(Canada) Canada Central"/> |
| Size | <div>F2 2 Capacity units Change size</div> |
| Fabric capacity administrator * | <input type="text" value="sudkhar@jyotikharbanda233gmail.onmicrosoft.com"/> |

[Select](#)

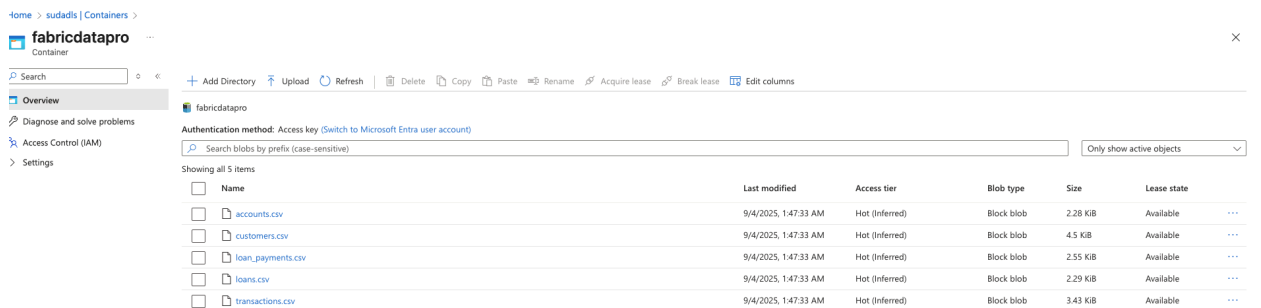
[Review + create](#)

[< Previous](#)

[Next: Tags >](#)



Create a container in the adls gen2 and upload the following csv files into it.



Code for creating a watermark table for incremental loading

```
Create table watermarkDemo
```

```
(
  id INT,
  tablename varchar(100),
```

```

schemaname varchar(50),

filename varchar(20),

foldername varchar(30),

lastprocessedvalue varchar(20),

incrementalcolumn varchar(20)

);

insert into watermarkDemo values
(1,'accounts','dbo','accounts','bootcamp','0','accounts')

insert into watermarkDemo values
(2,'customers','dbo','customers','bootcamp','0','customers')

insert into watermarkDemo values
(3,'loan_payments','dbo','loan_payments','bootcamp','0','loan_payments')

insert into watermarkDemo values
(4,'loans','dbo','loans','bootcamp','0','loans')

insert into watermarkDemo values
(5,'transactions','dbo','transactions','bootcamp','0','transactions')

Select * from watermarkDemo

DROP TABLE watermarkDemo

```

Watermark table for incremental loading

Messages

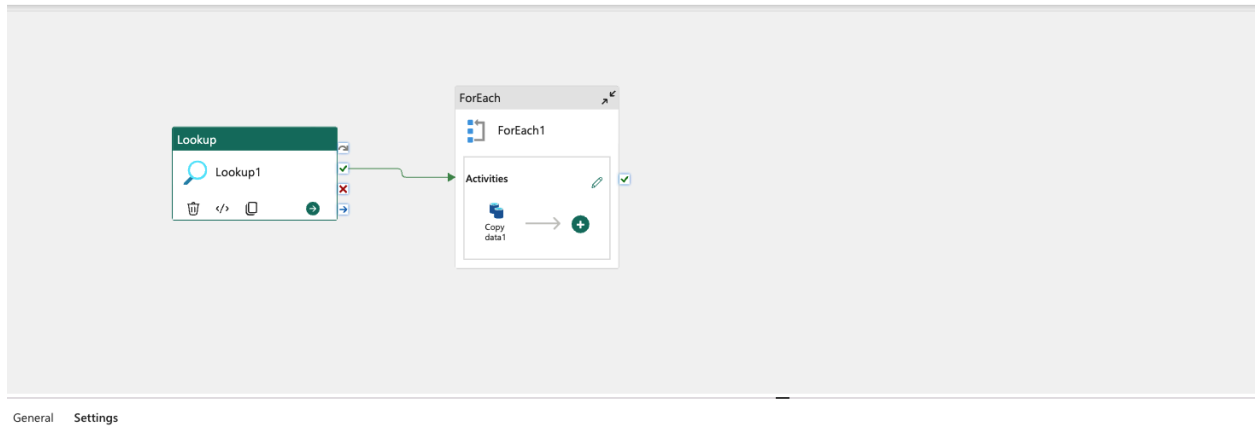
Results

Copy

Search

| | 123 id | ABC tablename | ABC schemaname | ABC filename | ABC foldername | ABC lastprocessedvalue | ABC incrementalcolumn |
|---|--------|---------------|----------------|---------------|----------------|------------------------|-----------------------|
| 1 | 5 | transactions | dbo | transactions | bootcamp | 0 | transactions |
| 2 | 4 | loans | dbo | loans | bootcamp | 0 | loans |
| 3 | 3 | loan_payments | dbo | loan_payments | bootcamp | 0 | loan_payments |
| 4 | 1 | accounts | dbo | accounts | bootcamp | 0 | accounts |
| 5 | 2 | customers | dbo | customers | bootcamp | 0 | customers |

We are not doing incremental loading since we have to implement scd type-2 and updated rows will not come in the incremental loading, hence we are loading multiples files using lookup. This is how pipeline for loading data in the bronze layer will look like.



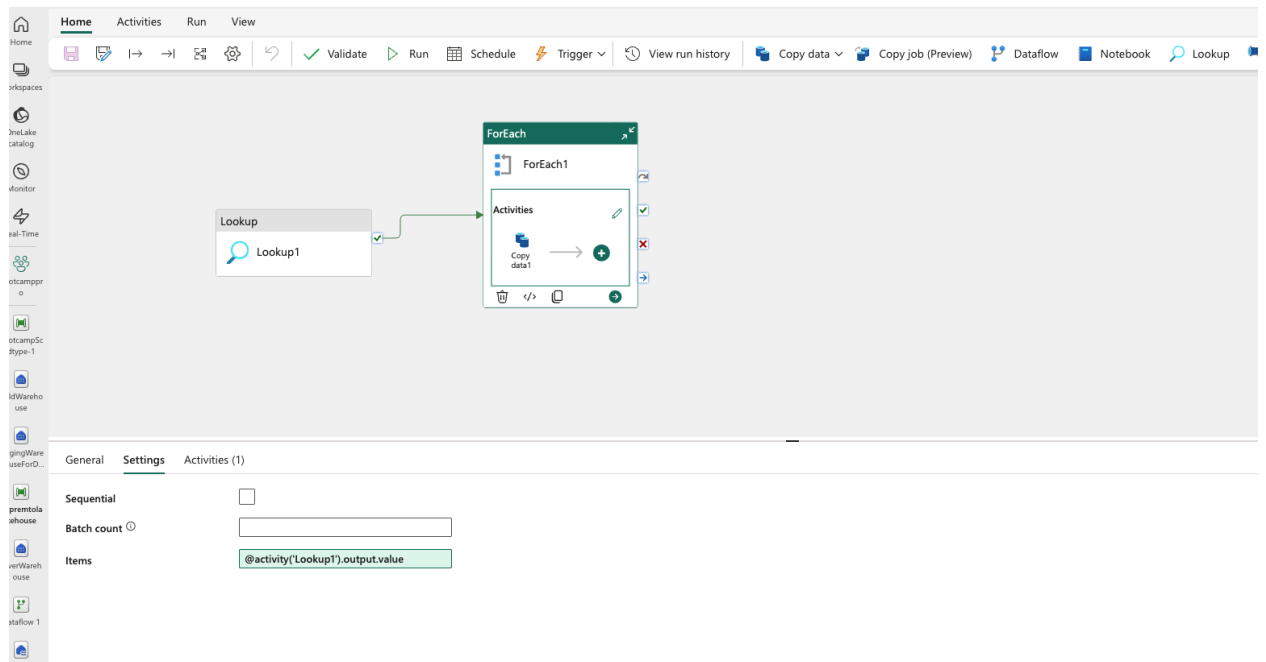
Adding for lookup activity, select the place where we have created the watermark table and selecting the name of the table.

The screenshot shows the Azure Data Factory pipeline editor with the 'Settings' tab selected for the 'Lookup' activity. The 'Settings' tab has two sections: 'General' and 'Settings'. The 'Settings' section is expanded, showing the following configuration:

- Use query:** ☒ Table ☐ Query ☐ Stored procedure
- Table:**
- First row only:** ☐
- Advanced:** ☐

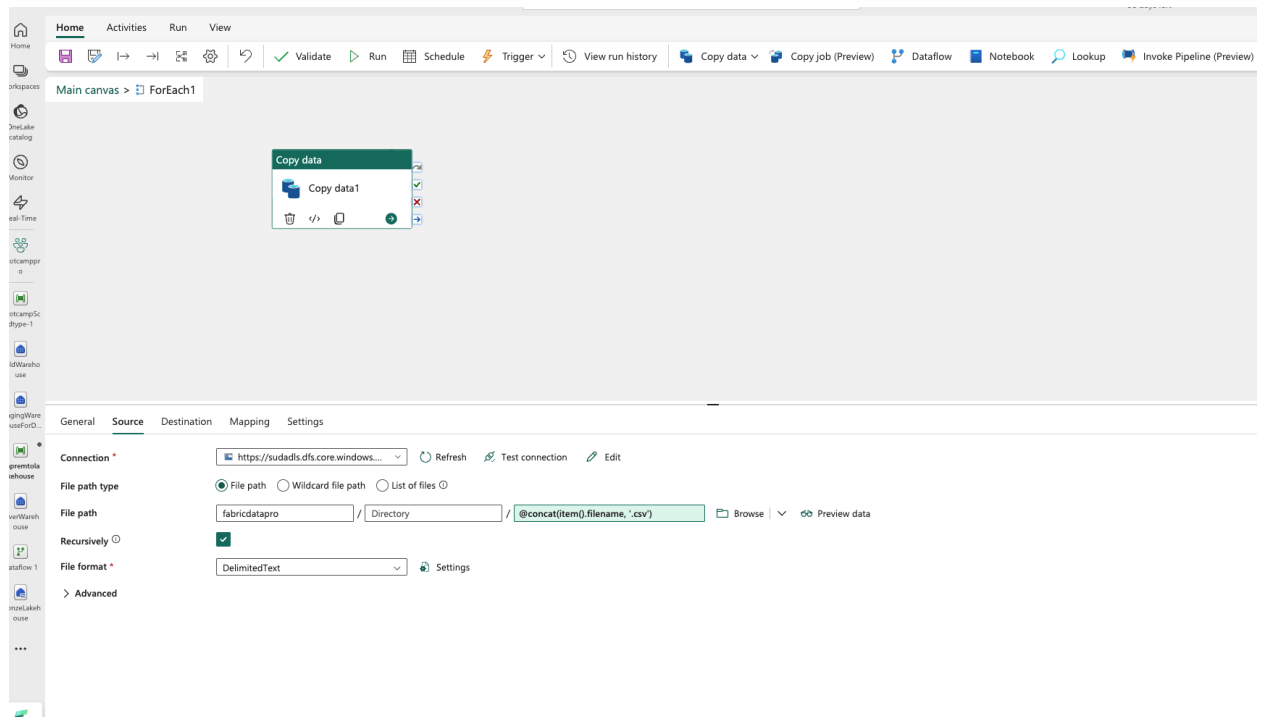
The 'General' tab is also visible, showing the 'Lookup' activity name and a dropdown menu for the 'StagingWarehouseForDataflows...'.

Adding the for each activity and adding the following expression.



```
@activity('Lookup1').output.value
```


Going inside the for each activity adding the copy data activity. In the copy data add the source path and the destination path.



Creating the connection in the copy data activity to connect ADLS Gen2 storage.

Get data

Connect data source

 **Azure Data Lake Storage Gen2**
Azure
[Learn more](#)

Connection settings

URL * ⓘ
Example: https://contosoadlscdm.dfs.core.windows.net...

Connection credentials


Connection
Create new connection ▼ ↺

Connection name
Connection

Data gateway
(none) ▼ ↺

Authentication kind
Organizational account ▼

You are currently signed in as:

 **Sudhanshu**
sudkhar@jyotikharbanda233gmail...
[Switch account](#)


Privacy Level
None ▼

☐ This connection can be used with on-premises data gateways and VNet data gateways.

Providing the URL, the structure can be found on the internet.

Get data

Connect data source

 **Azure Data Lake Storage Gen2**
Azure
[Learn more](#)

Connection settings

URL * ⓘ
https://sudadls.dfs.core.windows.net/

Connection credentials


Connection
Create new connection ▼ ↺

Connection name
https://sudadls.dfs.core.windows.net/

Data gateway
(none) ▼ ↺

Authentication kind
Organizational account ▼

You are currently signed in as:

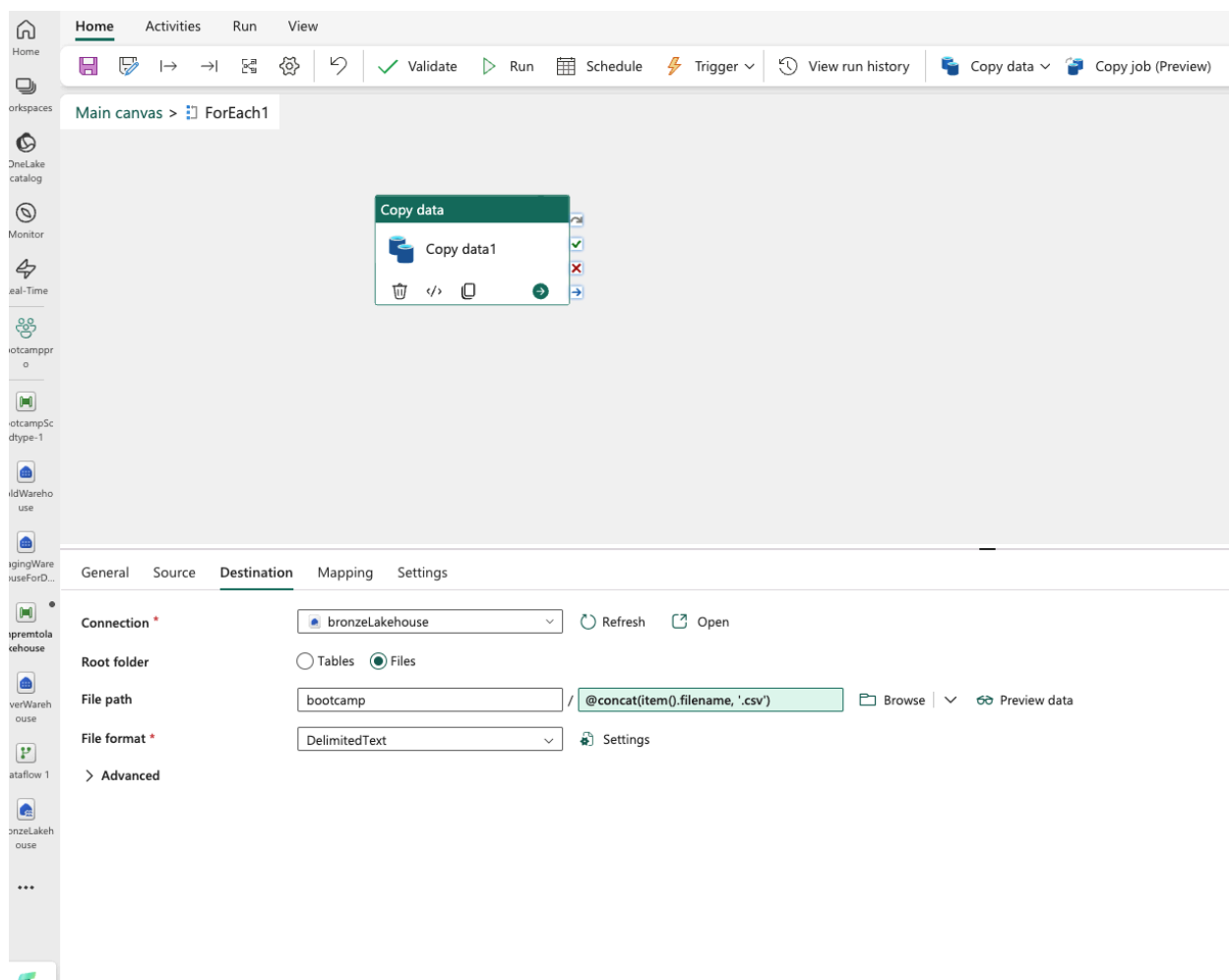
 **Sudhanshu**
sudkhar@jyotikharbanda233gmail...
[Switch account](#)

Privacy Level
None ▼

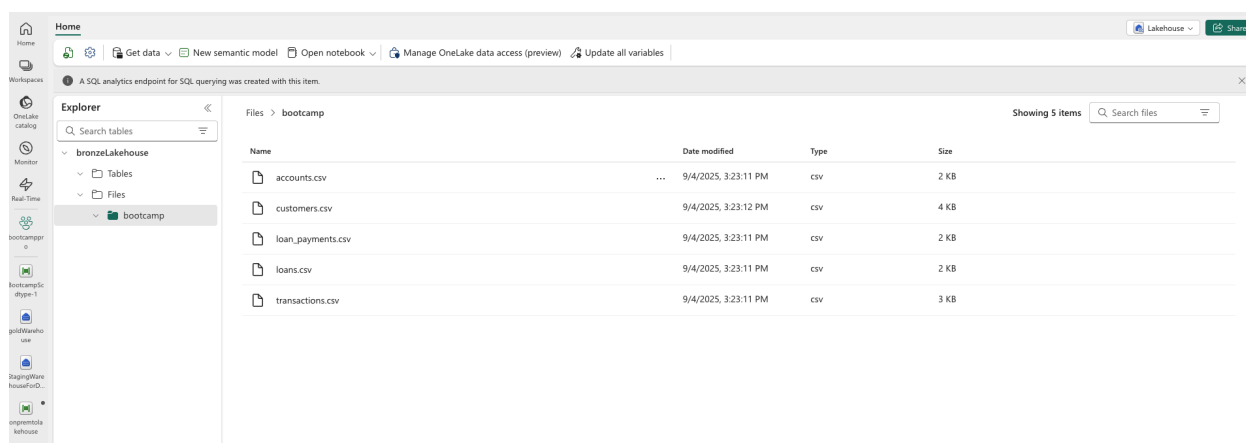
☐ This connection can be used with on-premises data gateways and VNet data gateways.

Back Connect

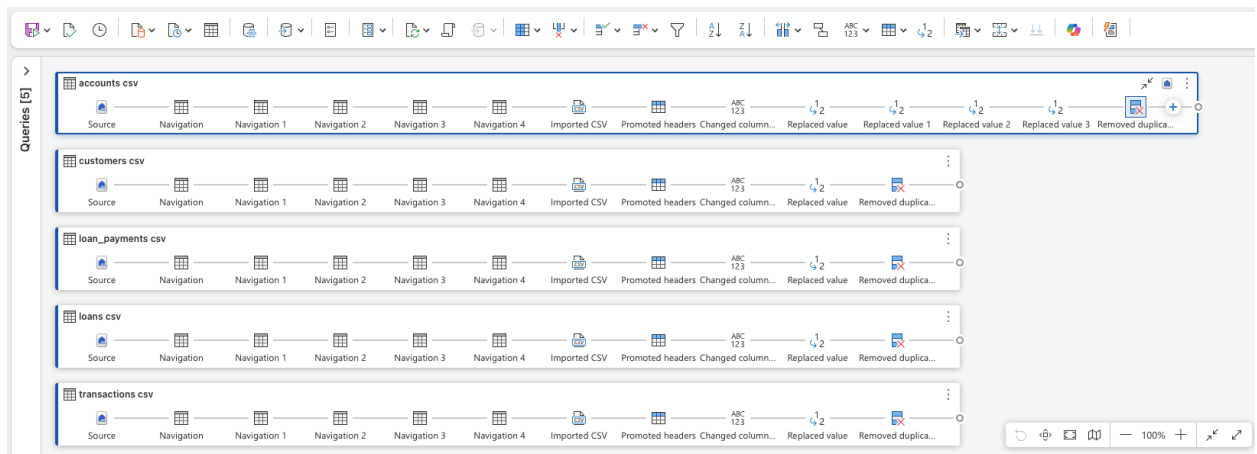
Providing the destination URL.



RUN THE pipeline, you can the data is copied from ADLS Gen2 to Fabric Lake House.



Going to the dataflow gen2 and creating the dataflow for cleaning the CSV file.



Importing the csv file.

The screenshot shows the 'Imported CSV' step being configured for the 'accounts csv' dataflow. The configuration window displays the file path and a preview of the CSV data. The configuration text is: `Csv.Document("#Navigation 4", [Delimiter = ",", Columns = 4, QuoteStyle = QuoteStyle.None])`. The preview table shows the following data:

| | Column1 | Column2 | Column3 | Column4 |
|----|------------|-------------|--------------|---------|
| 1 | account_id | customer_id | account_type | balance |
| 2 | 1 | 45 | Savings | 1000.50 |
| 3 | 2 | 12 | Checking | 2500.75 |
| 4 | 3 | 78 | Savings | 1500.00 |
| 5 | 4 | 34 | Checking | 3000.25 |
| 6 | 5 | 56 | Savings | 500.00 |
| 7 | 6 | 23 | Checking | 1200.50 |
| 8 | 7 | 89 | Savings | 800.75 |
| 9 | 8 | 67 | Checking | 2200.00 |
| 10 | 9 | 14 | Savings | 900.25 |

Adding the first row as columns.

Power Query Editor interface showing the 'Promoted headers' step for the 'accounts csv' table. The formula bar displays the M code: `Table.PromoteHeaders(#"Imported CSV", [PromoteAllScalars = true])`. The data table is visible below, showing columns: `account_id`, `customer_id`, `account_type`, and `balance`. The status bar indicates 'Columns: 4 Rows: 99+'.

| | account_id | customer_id | account_type | balance |
|----|------------|-------------|--------------|---------|
| 1 | 1 | 45 | Savings | 1000.50 |
| 2 | 2 | 12 | Checking | 2500.75 |
| 3 | 3 | 78 | Savings | 1500.00 |
| 4 | 4 | 34 | Checking | 3000.25 |
| 5 | 5 | 56 | Savings | 500.00 |
| 6 | 6 | 23 | Checking | 1200.50 |
| 7 | 7 | 89 | Savings | 800.75 |
| 8 | 8 | 67 | Checking | 2200.00 |
| 9 | 9 | 14 | Savings | 900.25 |
| 10 | 10 | 92 | Checking | 1800.50 |

Changing the datatype of columns

Power Query Editor interface showing the 'TransformColumnTypes' step for the 'accounts csv' table. The formula bar displays the M code: `Table.TransformColumnTypes(#"Promoted headers", {("account_id", Int64.Type), ("customer_id", Int64.Type), ("account_type", type text), ("balance", type text)})`. The data table is visible below, showing columns: `account_id`, `customer_id`, `account_type`, and `balance`. The status bar indicates 'Completed (1.00 s) Columns: 4 Rows: 99+'.

| | account_id | customer_id | account_type | 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 |

Changing the values where there is null value to 0.

Power Query

Home Transform Add column View Help

Queries [5]

- accounts csv
- customers csv
- loan_payments csv
- loans csv
- transactions csv

Table.ReplaceValue(#"Changed column type", null, 0, Replacer.ReplaceValue, {"account_id"})

| 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 |
| 9 | 9 | 14 Savings | 900.25 |
| 10 | 10 | 92 Checking | 1800.5 |

Completed (1.00 s) Columns: 4 Rows: 99+

Power Query

Home Transform Add column View Help

Search (Alt + Q)

Queries [5]

- accounts csv
- customers csv
- loan_payments csv
- loans csv
- transactions csv

Table.ReplaceValue(#"Replaced value", null, 0, Replacer.ReplaceValue, {"customer_id"})

| 1,2 account_id | 1,2 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 |
| 9 | 9 | 14 Savings | 900.25 |
| 10 | 10 | 92 Checking | 1800.5 |

Completed (0.85 s) Columns: 4 Rows: 99+

Changing the values where there is null value to 0.

The screenshot shows a data pipeline in Microsoft Fabric. The pipeline consists of several steps: Source, Navigation, Navigation 1, Navigation 2, Navigation 3, Navigation 4, Imported CSV, Promoted headers, Changed column..., Replaced value, Replaced value 1, Replaced value 2, Replaced value 3, and Removed duplicate. The query editor shows the following SQL query:

```
Table.ReplaceValue("#Replaced value 1", null, "Unknown", Replacer.ReplaceValue, {"account_type"})
```

The query results show a table with 10 rows and 4 columns: 1.2 account_id, 1.2 customer_id, 1.2 account_type, and 1.2 balance. The data is as follows:

| 1.2 account_id | 1.2 customer_id | 1.2 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 |
| 9 | 9 | 14 Savings | 900.25 |
| 10 | 10 | 92 Checking | 1800.5 |

The pipeline is completed in 0.93 seconds, with 4 columns and 99+ rows.

The screenshot shows a data pipeline in Microsoft Fabric. The pipeline consists of several steps: Source, Navigation, Navigation 1, Navigation 2, Navigation 3, Navigation 4, Imported CSV, Promoted headers, Changed column..., Replaced value, Replaced value 1, Replaced value 2, Replaced value 3, and Removed duplicate. The query editor shows the following SQL query:

```
Table.ReplaceValue("#Replaced value 2", null, 0, Replacer.ReplaceValue, {"balance"})
```

The query results show a table with 10 rows and 4 columns: 1.2 account_id, 1.2 customer_id, 1.2 account_type, and 1.2 balance. The data is as follows:

| 1.2 account_id | 1.2 customer_id | 1.2 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 |
| 9 | 9 | 14 Savings | 900.25 |
| 10 | 10 | 92 Checking | 1800.5 |

The pipeline is completed in 0.93 seconds, with 4 columns and 99+ rows.

Removing duplicates in the dataflow and adding the sink as data warehouse in each dataflow. Run the dataflow.

The screenshot displays the Azure Data Factory 'Queries' pane with five dataflows. Each dataflow follows a consistent pipeline structure: Source, Navigation, Navigation 1, Navigation 2, Navigation 3, Navigation 4, Imported CSV, Promoted headers, Changed column..., Replaced value, Replaced value 1, Replaced value 2, Replaced value 3, and Removed duplicates. The 'accounts csv' dataflow is selected and expanded, showing the query: `Table.Distinct(#'Replaced value 3', {'account_id'})`. Below the query, a data preview is shown with columns: 1.2 account_id, 1.2 customer_id, 1.2 account_type, and 1.2 balance. The preview contains 10 rows of data.

| 1.2 account_id | 1.2 customer_id | 1.2 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 |
| 9 | 9 | 14 Savings | 900.25 |
| 10 | 10 | 92 Checking | 1800.5 |

Here we can see all the tables have been created for the respective csv file where the cleaned data is stored.

The screenshot shows the Microsoft Fabric interface. The Explorer pane on the left displays the 'silverWarehouse' database with a list of tables. The 'accounts' table is selected, and the 'Data preview - accounts' table is shown. The preview displays 1000 rows of data with columns: 1.2 account_id, 1.2 customer_id, 1.2 account_type, and 1.2 balance. The preview shows a list of accounts with their respective customer IDs, account types, and balances.

| 1.2 account_id | 1.2 customer_id | 1.2 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 |
| 9 | 9 | 14 Savings | 900.25 |
| 10 | 10 | 92 Checking | 1800.5 |

Now we are creating an empty target table, which will be used for storing SCD type-1 data.


```

CREATE TABLE dbo.TarAccounts
(
    account_id    FLOAT           NOT NULL,
    customer_id   FLOAT           NOT NULL,
    account_type  VARCHAR(50)     NULL,
    balance       FLOAT           NULL,
    createdBy     VARCHAR(50)     NULL,
    createdOn     DATETIME2(3)    NULL,
    updatedBy     VARCHAR(50)     NULL,
    updatedOn     DATETIME2(3)    NULL
);

SELECT * FROM dbo.TarAccounts

SELECT * FROM silverWarehouse.dbo.accounts as source_accounts

SELECT account_id from source_accounts;

```

Here is the source table.

| 12F account_id | 12F customer_id | 40C account_type | 12F balance |
|----------------|-----------------|------------------|-------------|
| 1 | 45 | Savings | 1000.5 |
| 2 | 12 | Checking | 2500.75 |
| 3 | 78 | Savings | 1500 |
| 4 | 34 | Checking | 3000.25 |
| 5 | 56 | Savings | 500 |
| 6 | 23 | Checking | 1200.5 |
| 7 | 89 | Savings | 800.75 |
| 8 | 67 | Checking | 2200 |
| 9 | 14 | Savings | 900.25 |
| 10 | 92 | Checking | 1800.5 |
| 11 | 3 | Savings | 1100.75 |
| 12 | 81 | Checking | 2700 |
| 13 | 29 | Savings | 1300.25 |
| 14 | 64 | Checking | 3200.5 |
| 15 | 47 | Savings | 700.75 |

Creating a procedure for implementing SCD type-1

```

CREATE OR ALTER PROCEDURE dbo.sp_SyncAccounts

AS

BEGIN

    SET NOCOUNT ON;

```

```

-- Update existing records

UPDATE TARGETS SET

    TARGETS.[account_id] = SOURCE.[account_id],

    TARGETS.[customer_id] = SOURCE.[customer_id],

    TARGETS.[account_type] = SOURCE.[account_type],

    TARGETS.[Balance] = SOURCE.[balance],

    TARGETS.updatedBy = 'DataFlow_Updates',

    TARGETS.updatedOn = SYSDATETIME()

FROM goldWarehouse.dbo.TarAccounts AS TARGETS

JOIN silverWarehouse.dbo.accounts AS SOURCE

    ON TARGETS.[account_id]=SOURCE.[account_id]

WHERE TARGETS.[account_id] <> SOURCE.[account_id] OR

    TARGETS.[customer_id] <> SOURCE.[customer_id] OR

    TARGETS.[account_type] <> SOURCE.[account_type] OR

    TARGETS.[Balance] <> SOURCE.[balance];

-- Insert new records

INSERT INTO goldWarehouse.dbo.TarAccounts(

    [account_id], [customer_id], [account_type], [Balance],

    createdBy, createdOn, updatedBy, updatedOn

)

SELECT

    SOURCE.[account_id], SOURCE.[customer_id], SOURCE.[account_type],

    SOURCE.[balance],

    'Dataflow', SYSDATETIME(),

    'Dataflow', SYSDATETIME()

FROM silverWarehouse.dbo.accounts AS SOURCE

LEFT JOIN goldWarehouse.dbo.TarAccounts AS TARGETS

    ON TARGETS.account_id = SOURCE.account_id

```

```

WHERE TARGETS.account_id IS NULL;

END;

GO

EXEC dbo.sp_SyncAccounts;

```

After executing scd type-1, here is the output.

TarAccounts SQL query 1

Run Save as view Explain query Fix query errors

Preview Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. [Review terms](#)

```

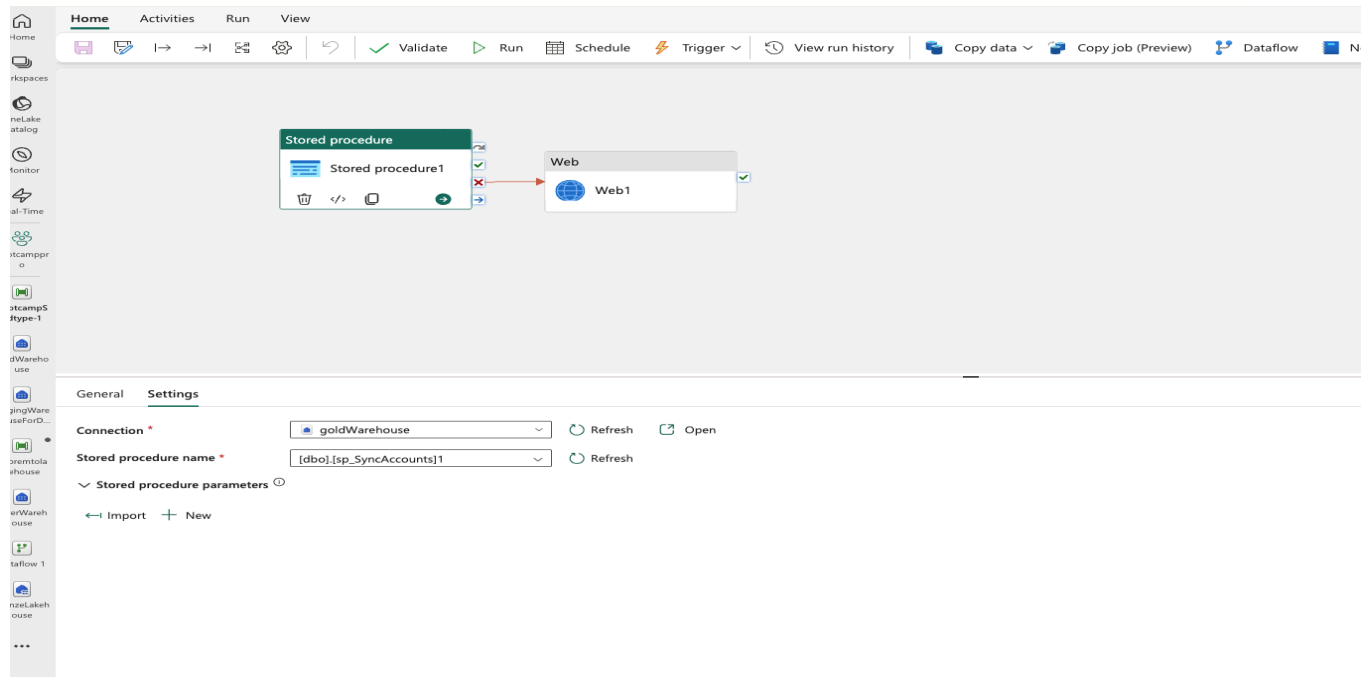
6 balance FLOAT NULL,
7 createdBy VARCHAR(50) NULL,
8 createdOn DATETIME2(3) NULL,
9 updatedBy VARCHAR(50) NULL,
10 updatedOn DATETIME2(3) NULL
11 );
12
13 SELECT * FROM dbo.TarAccounts
14
15
16 SELECT * FROM silverWarehouse.dbo.accounts as source_accounts
17 SELECT account_id from source_accounts;
18
19

```

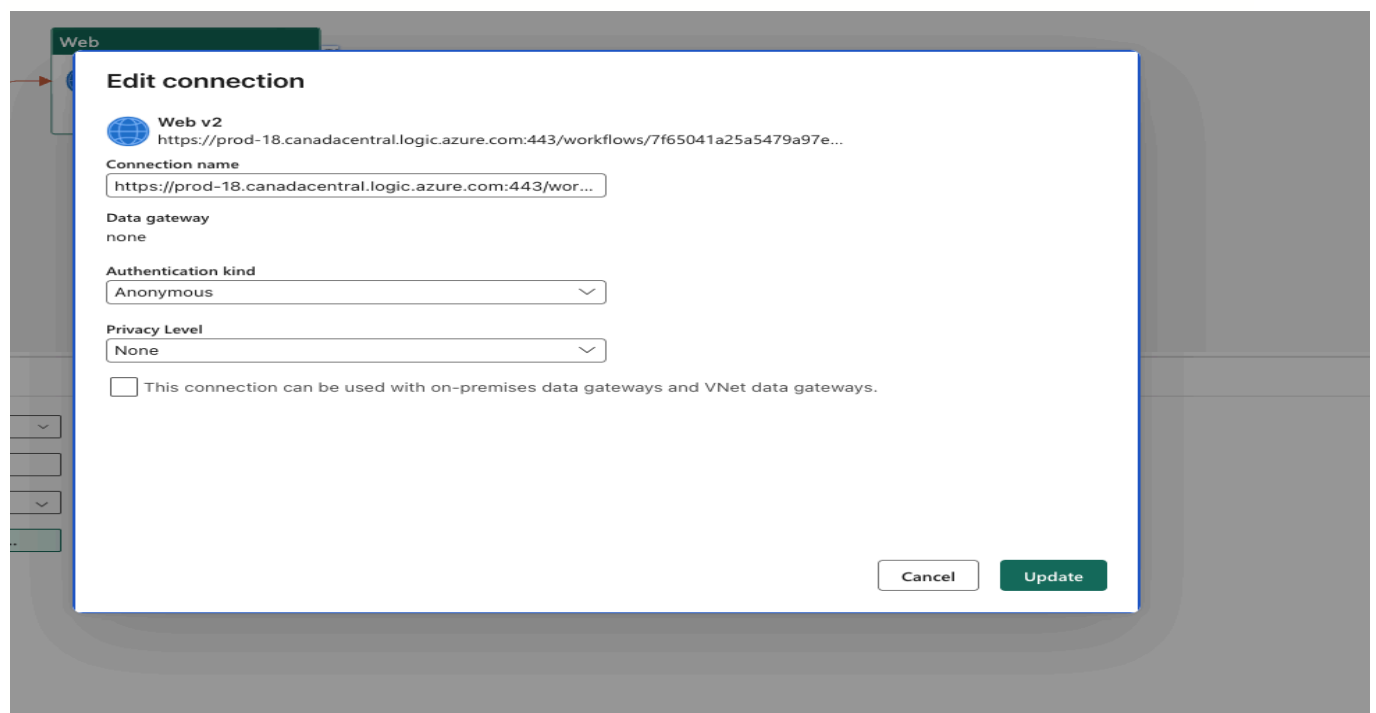
Messages Results Save as table Open in Excel Explore this data (preview) Copy Search

| | 12f account_id | 12f customer_id | ABC account_type | 12f Balance | ABC createdBy | createdOn | ABC updatedBy | updatedOn |
|----|----------------|-----------------|------------------|-------------|---------------|-------------------------|---------------|-------------------------|
| 1 | 1 | 45 | Savings | 1000.5 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 2 | 2 | 12 | Checking | 2500.75 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 3 | 3 | 78 | Savings | 1500 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 4 | 4 | 34 | Checking | 3000.25 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 5 | 5 | 56 | Savings | 500 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 6 | 6 | 23 | Checking | 1200.5 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 7 | 7 | 89 | Savings | 800.75 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 8 | 8 | 67 | Checking | 2200 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 9 | 9 | 14 | Savings | 900.25 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 10 | 10 | 92 | Checking | 1800.5 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 11 | 11 | 3 | Savings | 1100.75 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 12 | 12 | 81 | Checking | 2700 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 13 | 13 | 29 | Savings | 1300.25 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 14 | 14 | 64 | Checking | 3200.5 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 15 | 15 | 47 | Savings | 700.75 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 16 | 16 | 18 | Checking | 1400 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 17 | 17 | 99 | Savings | 600.25 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |
| 18 | 18 | 5 | Checking | 1600.5 | Dataflow | 2025-09-05 00:19:41.774 | Dataflow | 2025-09-05 00:19:41.774 |

Now we are creating a new pipeline that will trigger a mail when scdtype-1 implementation fails. Add a stored procedure activity in the pipeline and do the following configuration.



In the web activity, creating a connection with logic app that we will create in the azure.



Provide the following expression in the expression builder.

The screenshot shows the Microsoft Fabric interface. In the center, a pipeline diagram displays a 'Stored procedure' activity connected to a 'Web' activity. The 'Web' activity settings are open, showing a POST request to 'https://prod-18.canadacentral.io...' with a body containing a pipeline expression. The 'Pipeline expression builder' pane on the right shows the expression: {"PipelineName": "abc", "ErrorMessage": {"type": "object", "properties": {"PipelineName": {"type": "string"}, "ErrorMessage": {"type": "string"}, "workspacename": {"type": "integer"}}}}, "workspacename": 1}. The 'Activity outputs' pane shows the output of 'Stored procedure1'.

```
{
  "type": "object",
  "properties": {
    "PipeLineName": {
      "type": "string"
    },
    "ErrorMessage": {
      "type": "string"
    },
    "workspacename": {
      "type": "integer"
    }
  }
}
```

Create a logic app, add a when an http received and send email v2 activity. Do the following configuration. Copy the HTTP URL and paste into the web activity.

The screenshot shows the Azure Logic App Designer interface for a logic app named 'bootpro'. The workflow consists of two activities: 'When an HTTP request is received' and 'Send email (V2)'. The right-hand pane displays the configuration for the 'When an HTTP request is received' trigger. The HTTP URL is set to 'https://prod-18.canadacentral.logic.azure.com:443/workflows/7f65041a25a5479a97a2c9f9397a0644/...', the method is 'POST', and the request body JSON schema is defined as follows:

```
{
  "type": "object",
  "properties": {
    "PipelineName": {
      "type": "string"
    },
    "ErrorMessage": {
      "type": "string"
    },
    "workspaceName": {
      "type": "integer"
    }
  }
}
```

Provide your email ID and provide the following configuration.

The screenshot shows the configuration for the 'Send email (V2)' activity. The 'To' field is set to 'dude.sk44@gmail.com'. The 'Advanced parameters' section shows 'Showing 3 of 6'. The 'Importance' is set to 'High'. The 'Subject' is 'Pipeline Failed'. The 'Body' is configured with a rich text editor containing the following text:

Hi team
Name of Pipeline - PipelineName
Workspace - workspaceName
Error Message - ErrorMessage

Run the pipeline when the stored procedure fails, you will receive the mail.

