

BOOTCAMP PROJECT 1

CUSTOMER ACCOUNT ANALYSIS

SUBMITTED BY – DRUSYA SURESH

PROJECT OVERVIEW

The project aims to design and implement a robust data pipeline for processing customer account data. This includes copying data from a backend team's storage account, performing necessary transformations using ADF and upserting (inserting or updating) data from a file stored in Azure Data Lake Storage into sql database table. The pipeline aims to ensure efficient, accurate, and scalable data processing to support downstream analytics and reporting needs.

PROJECT STEPS

STEP 1: Download the dataset from kaggle account to the on-premise system. Now with the help of virtual machine and self hosted integration runtime copy the csv files and load them into ADLS storage account (Bronze layer).

STEP 2: Silver layer – Now clean the data with the help of dataflows in datafactory. We have to clean the data by removing duplicates, null values, irrelevant data and any hanging data. For this we use filter, derived column, data type conversion, aggregate functions like transformations depending on the needs and load it as delta files in silver container. This completes our silver layer.

STEP 3: Gold layer – We have cleaned the data and converted to delta files. Now we perform SCD Type 1 transformation on 4 files and SCD Type 2 transformation on 5th file. We chose 5 seperate dataflows for this to run the pipeline faster.

STEP 4: Build a master pipeline and include the child pipelines inside the master and trigger the master pipeline so that it will run on a specified schedule.

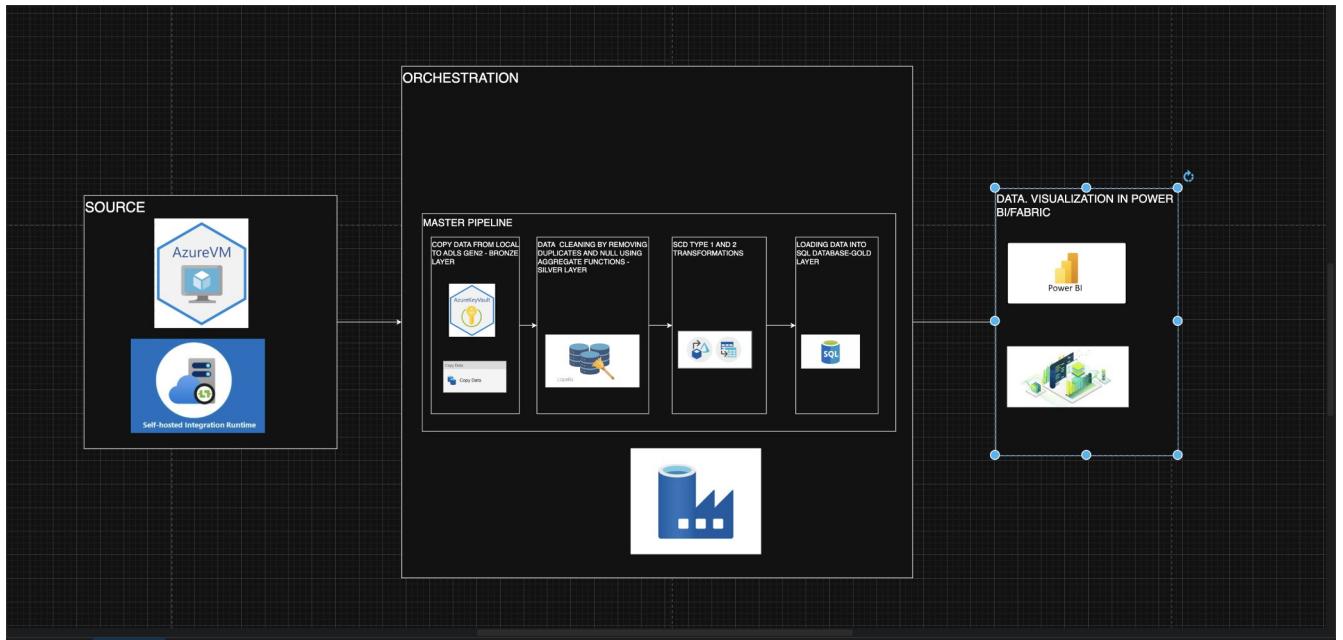
STEP 5(Optional): Visualize the data using Power BI or Fabric.

PREREQUISITES

- ▶ Download the dataset from Kaggle. <https://www.kaggle.com/datasets/varunkumari/ai-bank-dataset>. The dataset consists of 5 csv files.
 1. accounts.csv
 2. customers.csv
 3. loan_payments.csv
 4. loans.csv
 5. transactions.csv

- ▶ Create a virtual machine to move data from on-premise to cloud.
- ▶ Create a new resource datafactory in azure.
- ▶ Create linked services for accessing data in the pipelines in Azure datafactory.

ARCHITECTURE DIAGRAM



LINKED SERVICES

Created new linked services needed for building the pipelines.

1. For Azure Data Lake Storage account created a new linked service through key vault where the access key to the storage account already stored in the key vault secrets.

The screenshot shows the Microsoft Azure Data Factory interface for managing linked services. The left sidebar navigation includes General, Factory settings, Connector upgrade advisor, Connections, Linked services (selected), Integration runtimes, Microsoft Purview, Source control, Git configuration, ARM template, Author (Triggers, Global parameters, Data flow libraries), Security (Credentials, Customer managed key, Outbound rules, Managed private endpoints), and Workflow orchestration manager. The main content area is titled 'Edit linked service' for 'Azure Data Lake Storage Gen2'. It shows the 'Connect via integration runtime' set to 'AutoResolveIntegrationRuntime', 'Authentication type' as 'Account key', and 'Account selection method' set to 'Enter manually'. The 'URL' field contains 'https://adlsdrusya.dfs.core.windows.net/'. Under 'AKV linked service', 'ls_keyvaultadls' is selected. The 'Secret name' is 'adlscredentials' and 'Secret version' is '7ffce2958c034d9cb3db6b6aada04c38'. Below these fields are 'Test connection' (radio buttons for 'To linked service' and 'To file path') and 'Annotations' sections. At the bottom are 'Save' and 'Cancel' buttons, and a 'Test connection' button with a gear icon.

The screenshot shows the Microsoft Azure Key Vault interface. At the top, there's a navigation bar with 'Microsoft Azure' and an 'Upgrade' button. Below it, a search bar says 'Search resources, services, and docs (G+ /)'. On the right, there are icons for Copilot, a gear, and user information ('danidvs97@outlook.com DEFAULT DIRECTORY').

The main content area shows a secret named 'adlscredentials'. It has a 'Versions' section with a 'New Version' button, a 'Refresh' button, a 'Delete' button, and a 'Download Backup' button. A table lists the current version, which is '7ffce2958c034d9cb3db6b6aada04c38' and is marked as 'Enabled'.

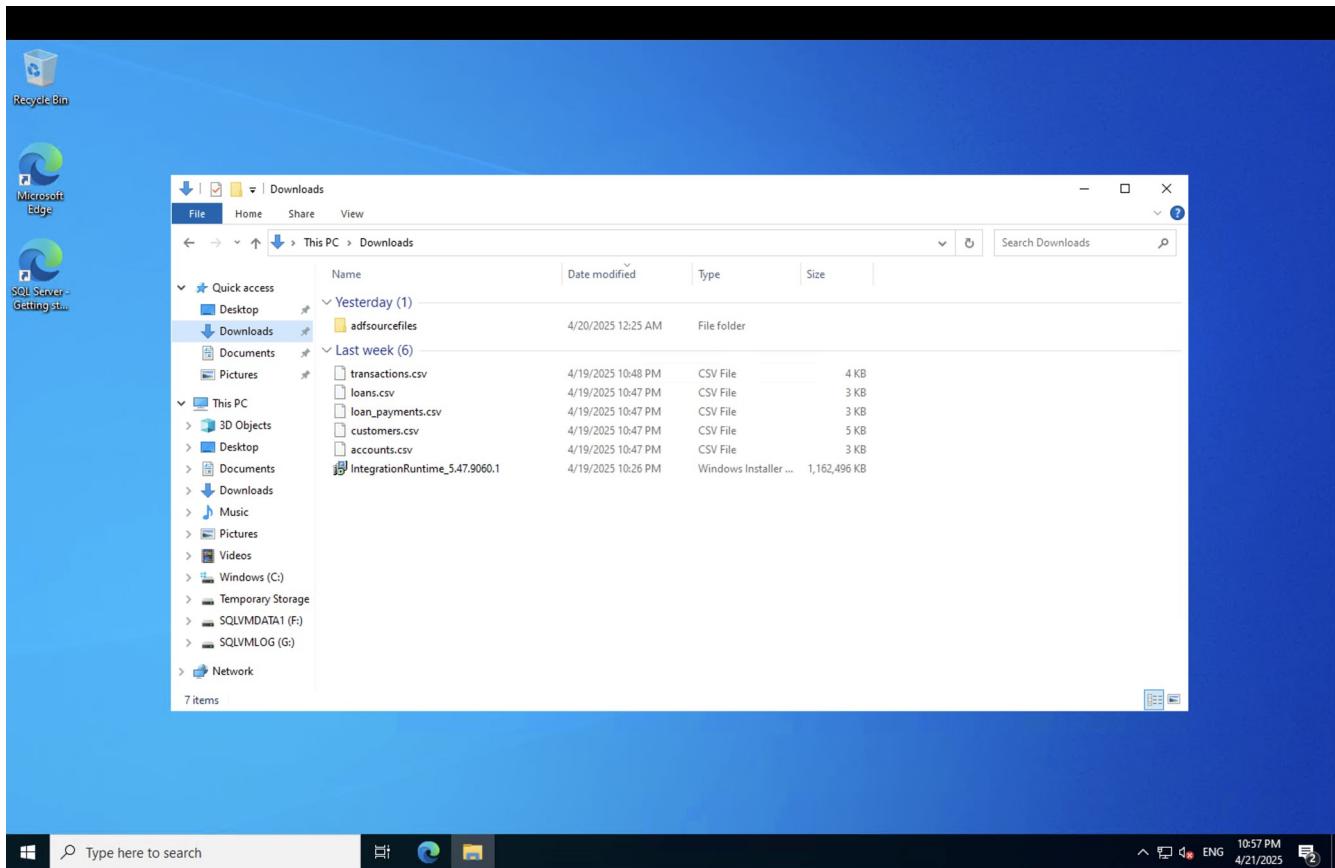
2. Created a new linked service for key vault and sql database.

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar includes options like 'General', 'Connections', 'Linked services', 'Integration runtimes', 'Microsoft Purview', 'Source control', 'Git configuration', 'ARM template', 'Author', 'Triggers', 'Global parameters', 'Data flow libraries', 'Security', 'Credentials', 'Customer managed key', 'Outbound rules', 'Managed private endpoints', and 'Workflow orchestration manager'. The 'Linked services' option is selected.

The main area shows a list of linked services: 'ls_fileSHIR' (File system), 'ls_keyvaultadls' (Azure Key Vault), 'ls_sqldatabase' (Azure SQL Database), and 'ls_adls' (Azure Data Lake Storage). On the right, a detailed view for 'ls_keyvaultadls' is shown. It's configured as an 'Azure Key Vault' linked service. The 'Name' field is 'ls_keyvaultadls'. The 'Base URL' is set to 'https://keyvaultdrusyanew.vault.azure.net/'. The 'Authentication method' is 'System-assigned managed identity'. There are sections for 'Annotations', 'Parameters', and 'Advanced'. Buttons at the bottom include 'Save', 'Cancel', and 'Test connection'.

STEP – 1 : BRONZE LAYER

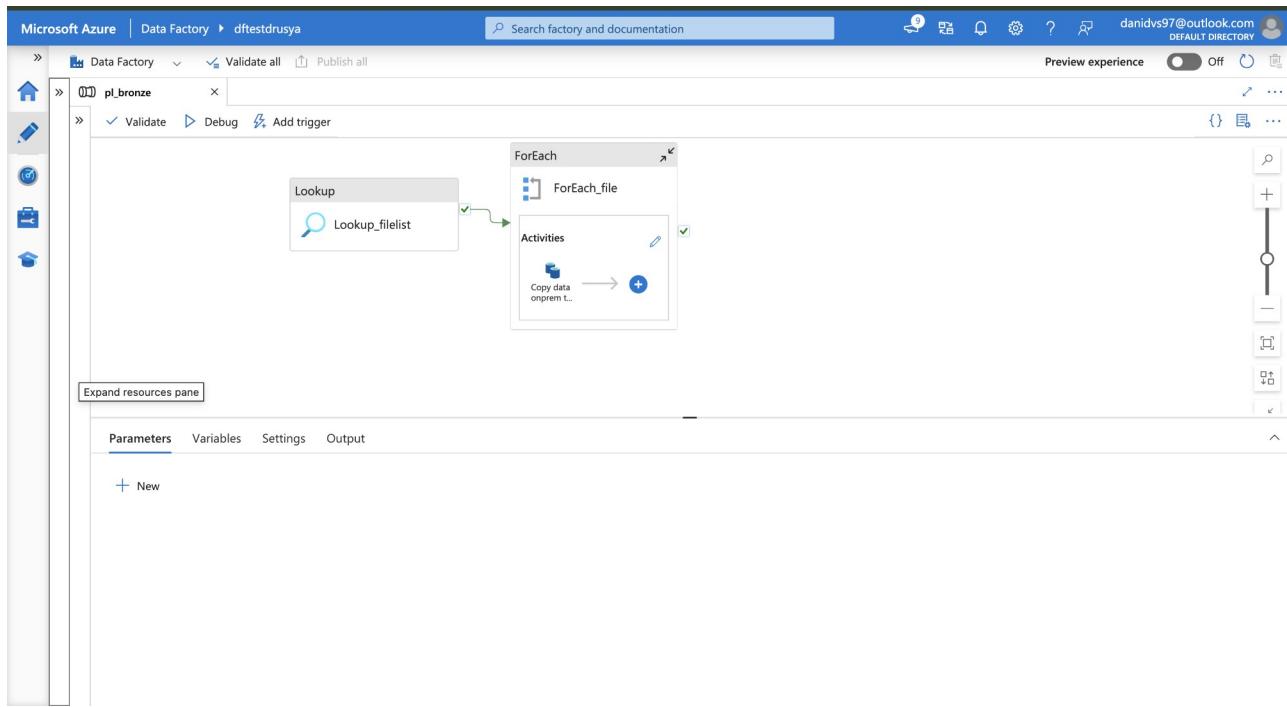
1. Created a virtual machine and downloaded the files in the virtual machine from the kaggle site. Now connected the virtual machine with cloud using the self hosted integration runtime. For this downloaded the integration runtime on virtual machine and created a new self hosted integration runtime in Azure datafactory.

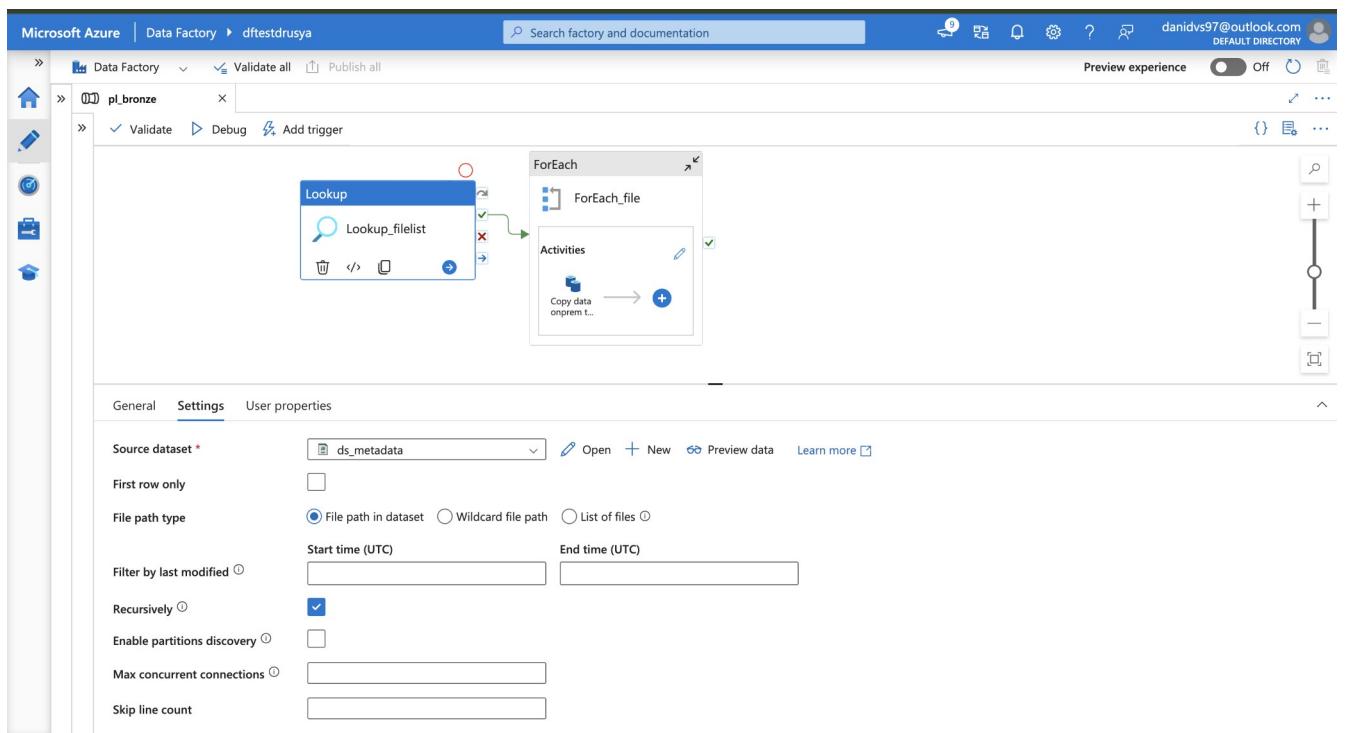


Now we have successfully created a connection with the on-premise and cloud.

2. Now we want to copy the data from on-premise to cloud with the help of a pipeline which is our bronze layer.

- Created a new pipeline in the datafactory and named it as bronze.
- Created a metadata csv file containing the names of the files to be uploaded in the storage account.
- Using for each and look up activity in the datafactory passed the file names dynamically using the metadata.
- Inside the for each added a copy activity and added the source as on-premise data source through file system linked service for self hosted runtime. Added the sink as ADLS with container name bronze.
- Performed the copy data activity and loaded the files successfully to the adls.





fileName
accounts.csv
customers.csv
loan_payments.csv
loans.csv
transactions.csv

General Settings User properties

Source dataset *: [dropdown]

First row only:

File path type:

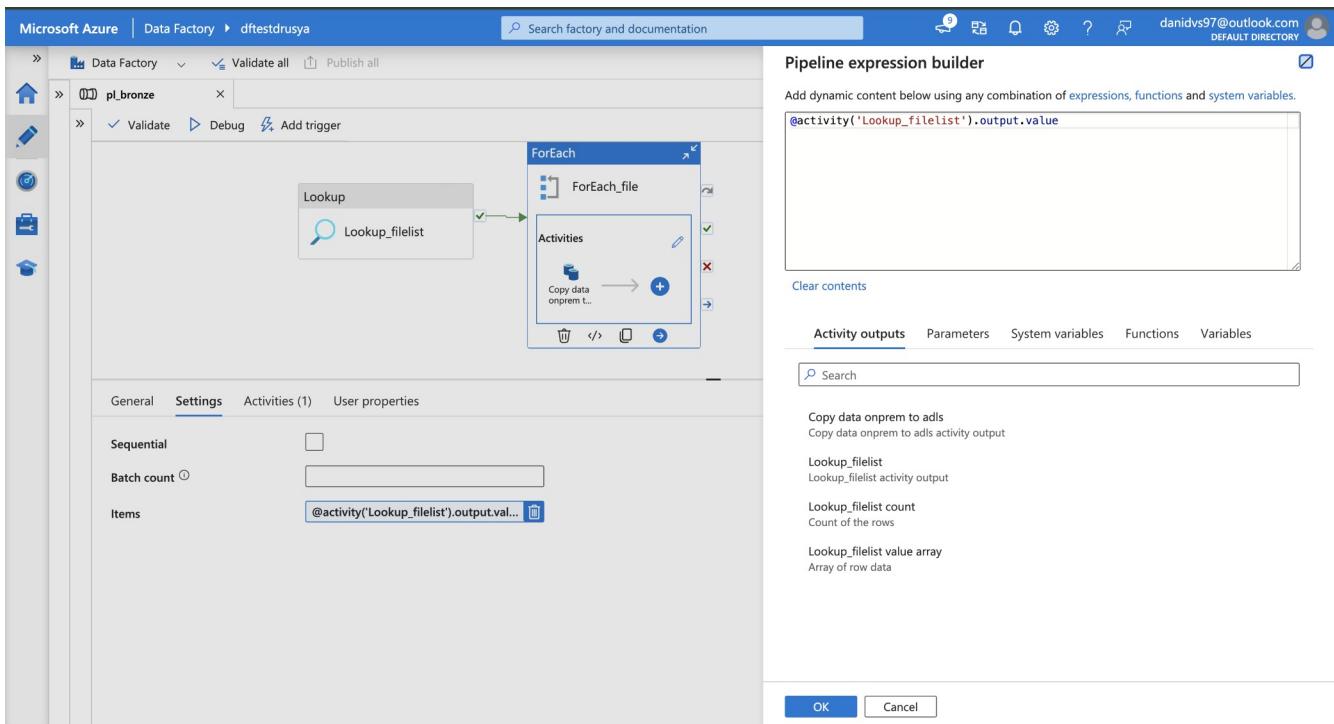
Filter by last modified:

Recursively:

Enable partitions discovery:

Max concurrent connections:

Skip line count:



General Source Sink Mapping Settings User properties

Source dataset * ds_csvlocal Open New Preview data Learn more

Dataset properties

Name	Type
Filename	string

File path type

File path in dataset File filter Wildcard file path List of files

Start time (UTC) End time (UTC)

Filter by last modified

Recursively

Enable partitions discovery

Microsoft Azure | Data Factory > dftestdrusya

Search factory and documentation

Validate all Publish all

Preview experience Off

pl_bronze

Validate Validate copy runtime Debug Add trigger

pl_bronze > ForEach_file

Copy data onprem to adls

General Source Sink Mapping Settings User properties

Sink dataset * ds_csv_adls

Dataset properties

Name	Type
Filename	string

Copy behavior Select...

Max concurrent connections

Block size (MB)

Metadata New

Quote all text

Microsoft Azure | Data Factory > dftestdrusya

Search factory and documentation

Validate all Publish all

Preview experience Off

pl_bronze

Validate Debug Add trigger

ds_csvlocal ds_csv_adls

ForEach

Lookup Lookup_filelist

Activities

Copy data onprem

Parameters Variables Settings Output

Pipeline run ID: eb6a798a-2100-488d-a604-25dce4d10627

Pipeline status Succeeded

All status List

Showing 1 - 7 of 7 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID
Copy data onprem to adls	Succeeded	Copy data	4/19/2025, 8:59:49 PM	22s	selfhostedonprem	f47cc1b1-83b9-4a9a-9d1b-35cf114cc4	
Copy data onprem to adls	Succeeded	Copy data	4/19/2025, 8:59:49 PM	26s	selfhostedonprem	155dc284-cc29-41ed-921c-2f64eca5a	
Copy data onprem to adls	Succeeded	Copy data	4/19/2025, 8:59:49 PM	32s	selfhostedonprem	63483f4e-28e6-463a-93cb-39d6fadd92	
Copy data onprem to adls	Succeeded	Copy data	4/19/2025, 8:59:49 PM	22s	selfhostedonprem	52b6f031-93f6-4a17-b0f1-3652ae1aa5	
Copy data onprem to adls	Succeeded	Copy data	4/19/2025, 8:59:49 PM	21s	selfhostedonprem	0d3efe86-6130-4da4-8f73-403e07a25e	
ForEach_file	Succeeded	ForEach	4/19/2025, 8:59:48 PM	34s		b5cdde7b-2659-4741-ab73-4f84608a3	
Lookup_filelist	Succeeded	Lookup	4/19/2025, 8:59:38 PM	10s	AutoResolveIntegrationRuntime (East US)	72d4579c-f741-495c-ab69-709ce4c40e	

Authentication method: Access key ([Switch to Microsoft Entra user account](#))
Location: bronze

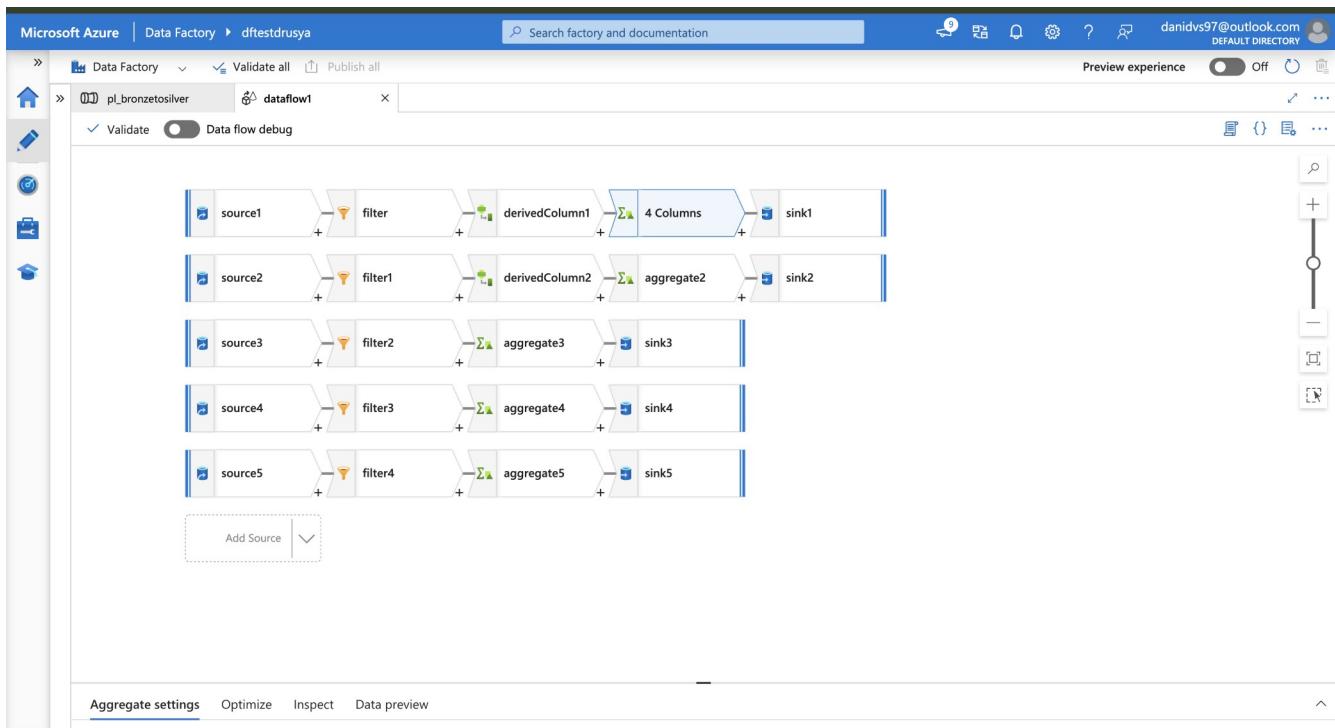
Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> accounts.csv	4/19/2025, 9:00:08 PM	Hot (Inferred)		Block blob	2.28 KiB	Available
<input type="checkbox"/> customers.csv	4/19/2025, 9:00:16 PM	Hot (Inferred)		Block blob	4.5 KiB	Available
<input type="checkbox"/> loan_payments.csv	4/19/2025, 9:00:07 PM	Hot (Inferred)		Block blob	2.55 KiB	Available
<input type="checkbox"/> loans.csv	4/19/2025, 9:00:11 PM	Hot (Inferred)		Block blob	2.29 KiB	Available
<input type="checkbox"/> transactions.csv	4/19/2025, 9:00:07 PM	Hot (Inferred)		Block blob	3.43 KiB	Available

STEP -2 : SILVER LAYER

Now we have our raw data in the bronze container. We have to clean the data and move it to Silver container. For cleaning the data I have used a separate pipeline named bronze to silver. Inside the pipeline added the dataflow activity and using data flow transformations I did the cleaning of data. The main transformations I used are as follows.

- Filter transformation – For removing null and empty values.
- Derived column transformation – For changing data types and trim white spaces.
- Aggregate transformation for removing duplicates.

I used my sink as delta file format. After cleaning the data I successfully loaded my data to the silver container.



The screenshot shows the Microsoft Azure Storage Blob container named 'silver'. The left sidebar includes 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings'. The main area displays blob details: 'Authentication method: Access key' and 'Location: silver'. A search bar at the top allows searching by prefix. Below, a table lists blobs:

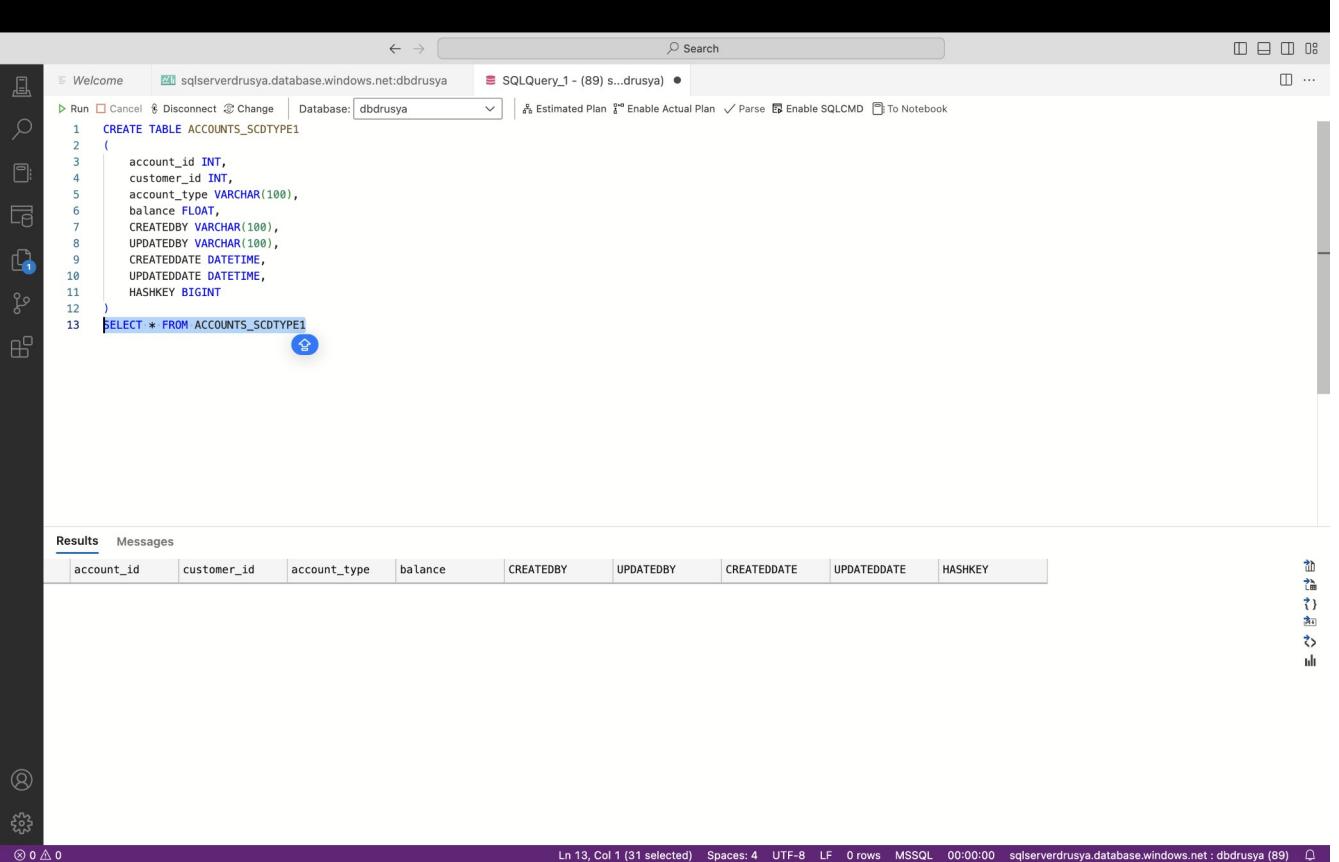
Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
accountsdelta	4/20/2025, 1:39:49 AM				-	***
customersdelta	4/20/2025, 1:39:41 AM				-	***
loan_paymentsdelta	4/20/2025, 1:39:22 AM				-	***
loansdelta	4/20/2025, 1:39:35 AM				-	***
transactionsdelta	4/20/2025, 1:39:54 AM				-	***

<https://portal.azure.com/# ping Cmd+Shift+F>

STEP – 3 : GOLD LAYER

Now I have the cleaned data as delta files in my silver container and i have to perform SCD Type 1 transformation on 4 files and SCD Type 2 transformation on the 5th file and have to load it to the SQL database.

- Created a new pipeline named silver to bronze.
- Added data flow to the workspace.
- Used 5 separate dataflow for SCD transformations of 5 different delta files.
- Created 5 different tables according to the source columns in the SQL database to load the transformed data.
- Successfully implemented the transformations and loaded the data.



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the center, there is a query window titled "SQLQuery_1 - (89) s...drusya". The window contains the following T-SQL code:

```
1 CREATE TABLE ACCOUNTS_SCDTYPE1
2 (
3     account_id INT,
4     customer_id INT,
5     account_type VARCHAR(100),
6     balance FLOAT,
7     CREATEDBY VARCHAR(100),
8     UPDATEDBY VARCHAR(100),
9     CREATEDDATE DATETIME,
10    UPDATEDDATE DATETIME,
11    HASHKEY BIGINT
12 )
13 SELECT * FROM ACCOUNTS_SCDTYPE1
```

Below the query window, the "Results" tab is selected, showing a table with the following columns: account_id, customer_id, account_type, balance, CREATEDBY, UPDATEDBY, CREATEDDATE, UPDATEDDATE, and HASHKEY. The table currently has no data rows.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the top-left corner, there's a 'Welcome' button and a 'Run' button. The top bar includes tabs for 'sqlserverdrusya.database.windows.net:dbdrusya' and 'SQLQuery_2 - (69 s...drusya)'. On the right side of the top bar are icons for 'Estimated Plan', 'Enable Actual Plan', 'Parse', 'Enable SQLCMD', and 'To Notebook'. Below the top bar, the main area contains a code editor with the following T-SQL script:

```
1 CREATE TABLE CUSTOMERS_SCDTYPE1
2 (
3     customer_id INT,
4     first_name VARCHAR(100),
5     last_name VARCHAR(100),
6     address VARCHAR(100),
7     city VARCHAR(50),
8     state VARCHAR(50),
9     zip VARCHAR(20),
10    CREATEDBY VARCHAR(100),
11    UPDATEDBY VARCHAR(100),
12    CREATEDDATE DATETIME,
13    UPDATEDDATE DATETIME,
14    HASHKEY BIGINT
15 )
16 SELECT * FROM CUSTOMERS_SCDTYPE1
```

Below the code editor, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is selected, showing a table with the following columns: customer_id, first_name, last_name, address, city, state, zip, CREATEDBY, UPDATEDBY, CREATEDDATE, UPDATEDDATE, and HASHKEY. The table currently has no data rows.

At the bottom of the screen, the status bar displays: 'Ln 16, Col 1 (32 selected) Spaces: 4 UTF-8 LF 0 rows MSSQL 00:00:00 sqlserverdrusya.database.windows.net : dbdrusya (69)'.

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. In the top-left corner, there's a 'Welcome' button and a 'Run' button. The top bar includes tabs for 'Welcome', 'sqlserverdrusya.database.windows.net:dbdrusya', 'SQLQuery_3 - (93) s...drusya', 'SQLQuery_2 - (69) s...drusya', and 'SQLQuery_1 - (89) s...drusya'. There are also icons for 'Search', 'Estimated Plan', 'Enable Actual Plan', 'Parse', 'Enable SQLCMD', and 'To Notebook'. The main area contains a code editor with the following SQL script:

```
1 CREATE TABLE LOAN_PAYMENTS_SCDTYPE1
2 (
3     payment_id INT,
4     loan_id INT,
5     payment_date DATE,
6     payment_amount FLOAT,
7     CREATEDBY VARCHAR(100),
8     UPDATEDBY VARCHAR(100),
9     CREATEDDATE DATETIME,
10    UPDATEDDATE DATETIME,
11    HASHKEY BIGINT
12 )
13 SELECT * FROM LOAN_PAYMENTS_SCDTYPE1
14
```

Below the code editor is a results grid with the following columns:

payment_id	loan_id	payment_date	payment_amount	CREATEDBY	UPDATEDBY	CREATEDDATE	UPDATEDDATE	HASHKEY

The status bar at the bottom displays: 'Ln 13, Col 1 (36 selected)' and 'Spaces: 4'. The footer of the SSMS window shows the connection details: 'sqlserverdrusya.database.windows.net : dbdrusya (93)'.

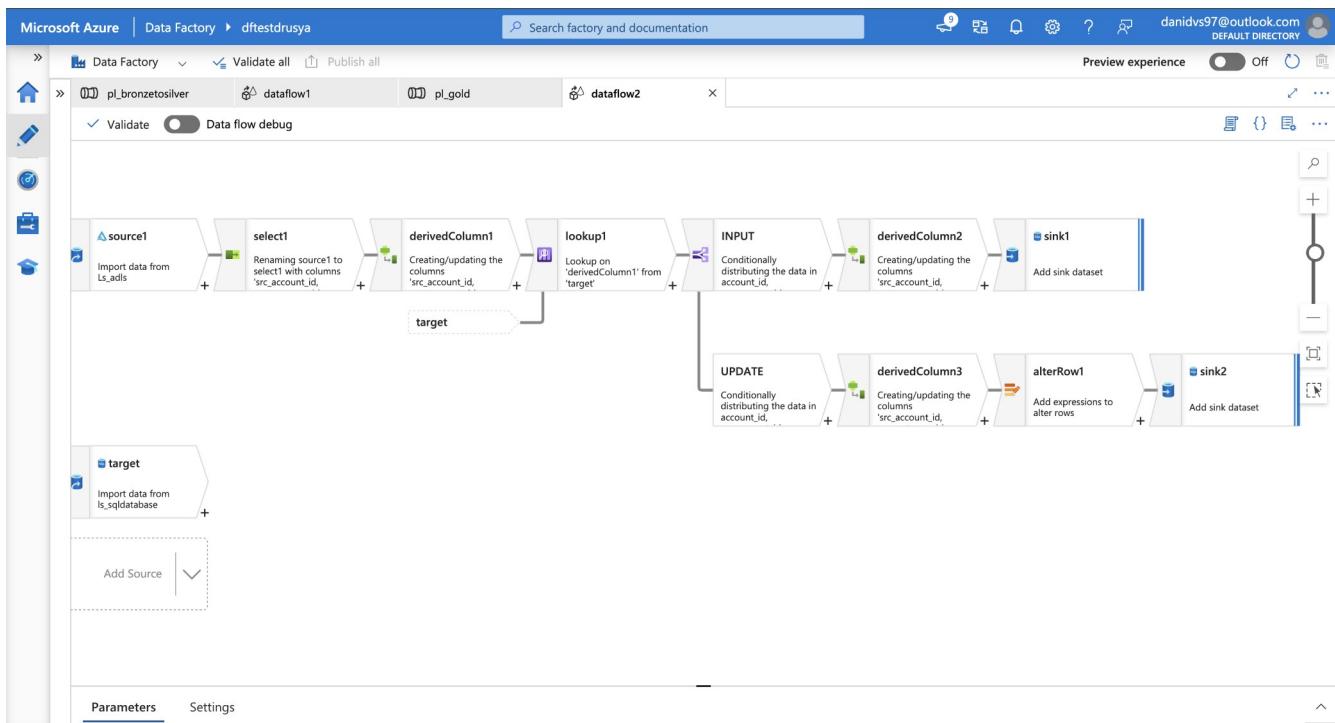
The screenshot shows the SQL Server Management Studio interface. In the top-left corner, there's a 'Welcome' button and a 'Run' button. The top bar includes tabs for 'sqlserverdrusya.database.windows.net:dbdrusya' and several other queries labeled 'SQLQuery_1' through 'SQLQuery_4'. On the left, there's a vertical toolbar with icons for search, database, file, and other management tasks. The main area contains two code snippets. The first snippet creates the 'LOANS_SCDTYPE1' table with columns: loan_id (INT), customer_id (INT), loan_amount (FLOAT), interest_rate (FLOAT), loan_term (INT), CREATEDBY (VARCHAR(100)), UPDATEDBY (VARCHAR(100)), CREATEDDATE (DATETIME), UPDATEDDATE (DATETIME), and HASHKEY (BIGINT). The second snippet is a 'SELECT * FROM LOANS_SCDTYPE1' query. Below the code, the 'Results' tab is selected, showing a table header with columns: loan_id, customer_id, loan_amount, interest_rate, loan_term, CREATEDBY, UPDATEDBY, CREATEDDATE, UPDATEDDATE, and HASHKEY. The status bar at the bottom indicates 'Ln 13, Col 1 (28 selected)' and '0 rows'.

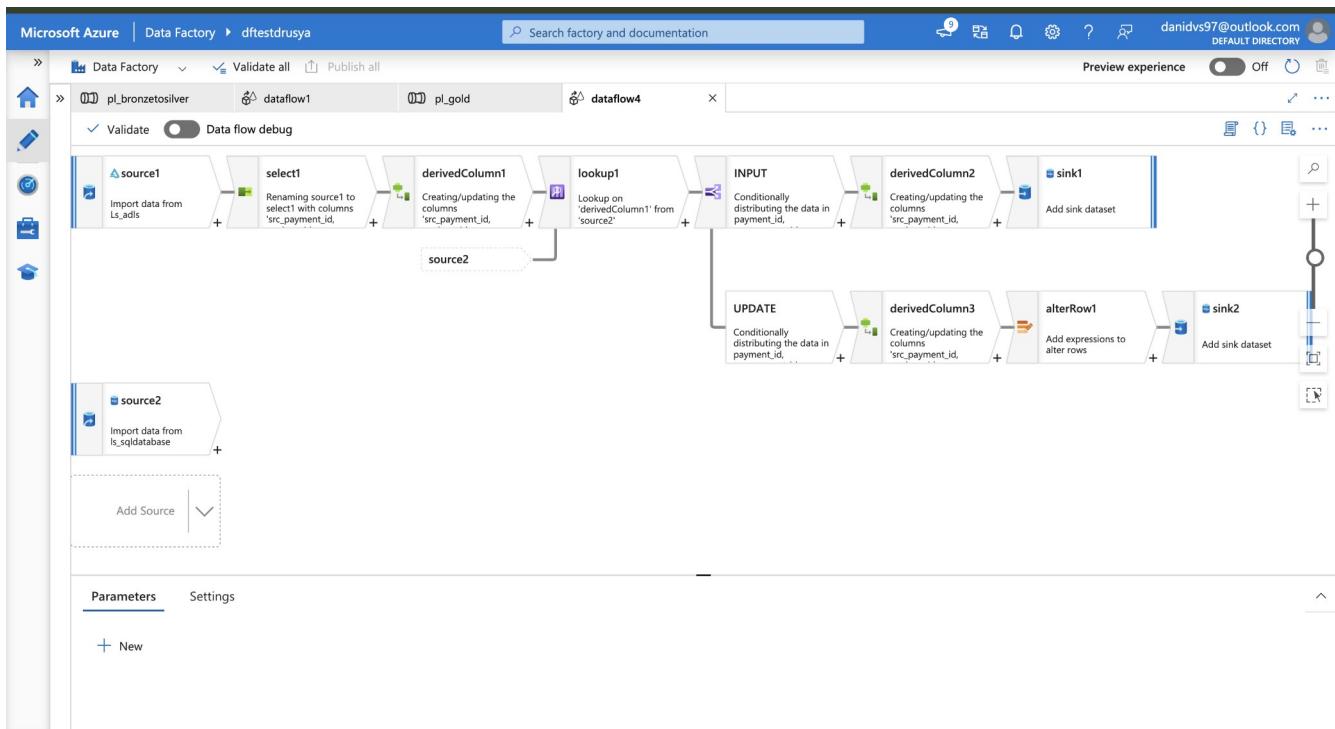
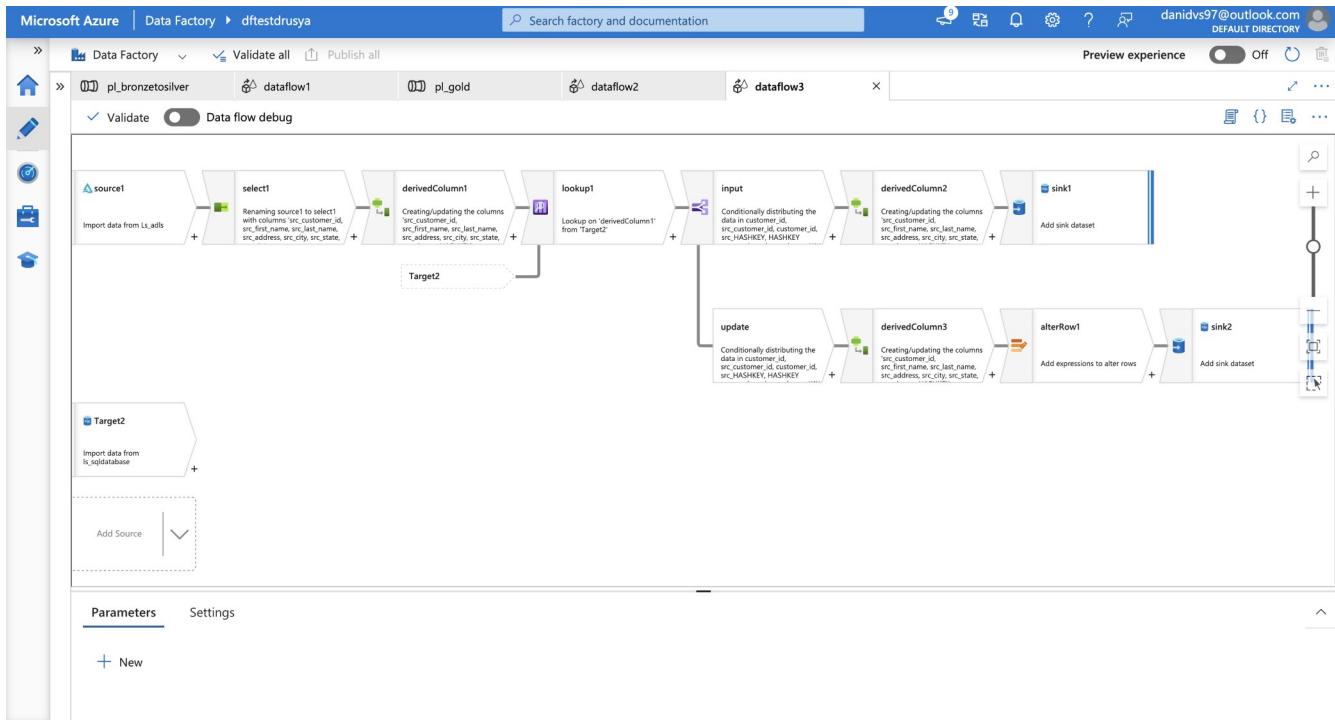
```
1 CREATE TABLE LOANS_SCDTYPE1(
2     loan_id INT,
3     customer_id INT,
4     loan_amount FLOAT,
5     interest_rate FLOAT,
6     loan_term INT,
7     CREATEDBY VARCHAR(100),
8     UPDATEDBY VARCHAR(100),
9     CREATEDDATE DATETIME,
10    UPDATEDDATE DATETIME,
11    HASHKEY BIGINT
12 )
13 SELECT * FROM LOANS_SCDTYPE1
```

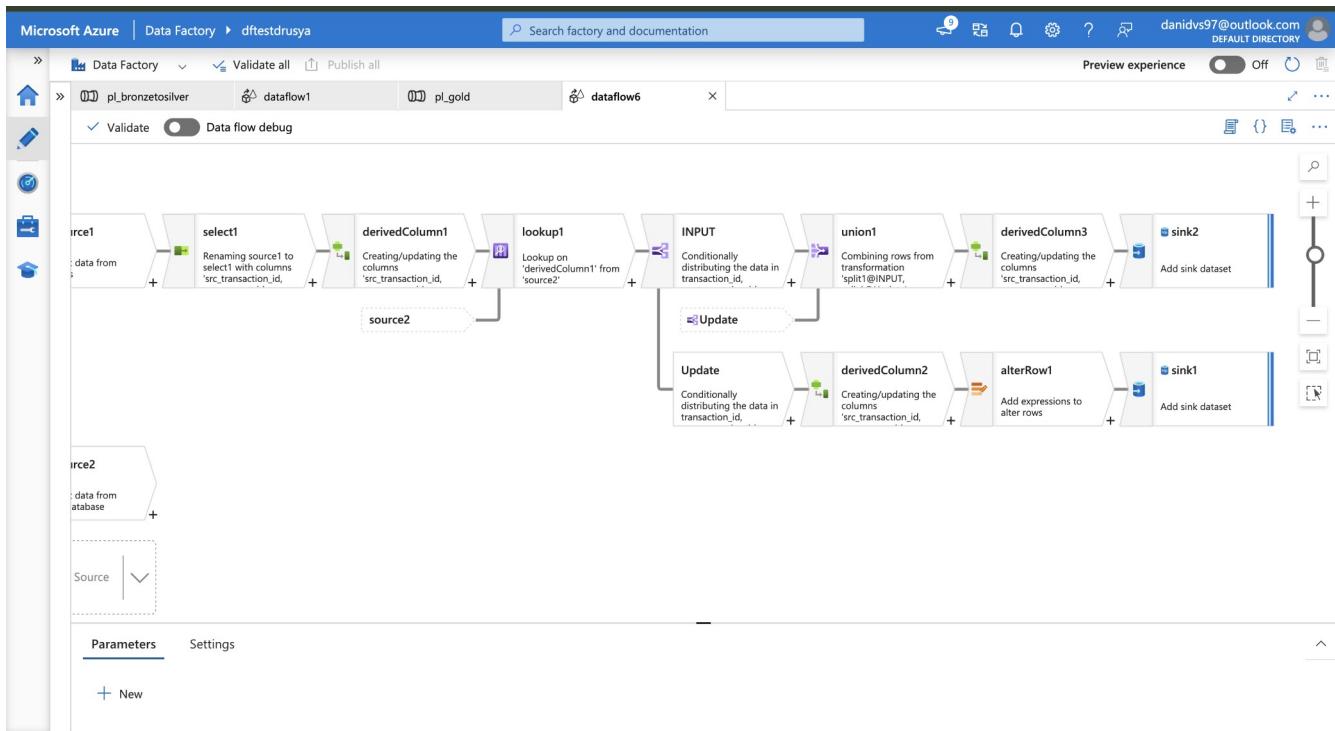
This screenshot is similar to the one above, showing the creation of another table. The top bar and toolbar are identical. The main area contains two code snippets. The first snippet creates the 'TRANSACTIONS_SCDTYPE2' table with columns: transaction_id (INT), account_id (INT), transaction_date (DATE), transaction_amount (FLOAT), transaction_type (VARCHAR(100)), CREATEDBY (VARCHAR(100)), UPDATEDBY (VARCHAR(100)), CREATEDDATE (DATETIME), UPDATEDDATE (DATETIME), ISACTIVE (INT), and HASHKEY (BIGINT). The second snippet is a 'SELECT * FROM TRANSACTIONS_SCDTYPE2' query. Below the code, the 'Results' tab is selected, showing a table header with columns: transaction_id, account_id, transaction_da..., transaction_am..., transaction_ty..., CREATEDBY, UPDATEDBY, CREATEDDATE, UPDATEDDATE, ISACTIVE, and HASHKEY. The status bar at the bottom indicates 'Ln 15, Col 1 (35 selected)' and '0 rows'.

```
1 CREATE TABLE TRANSACTIONS_SCDTYPE2(
2     transaction_id INT,
3     account_id INT,
4     transaction_date DATE,
5     transaction_amount FLOAT,
6     transaction_type VARCHAR(100),
7     CREATEDBY VARCHAR(100),
8     UPDATEDBY VARCHAR(100),
9     CREATEDDATE DATETIME,
10    UPDATEDDATE DATETIME,
11    ISACTIVE INT,
12    HASHKEY BIGINT
13 )
14
15 SELECT * FROM TRANSACTIONS_SCDTYPE2
```

The screenshot shows the Microsoft Azure Data Factory interface. On the left, there's a navigation sidebar with options: Home, Author, Monitor, Manage, and Learning Center. The main area displays a list of data flows under the project 'pl_bronzetrosilver'. The data flows listed are: Data flow1, Data flow2, Data flow3, Data flow4, and Data flow5. Each item has a small preview icon and a 'More' button. Below the list, there are tabs for Parameters, Variables, Settings, and Output, with 'Parameters' being the active tab. A 'New' button is located at the bottom of this section.

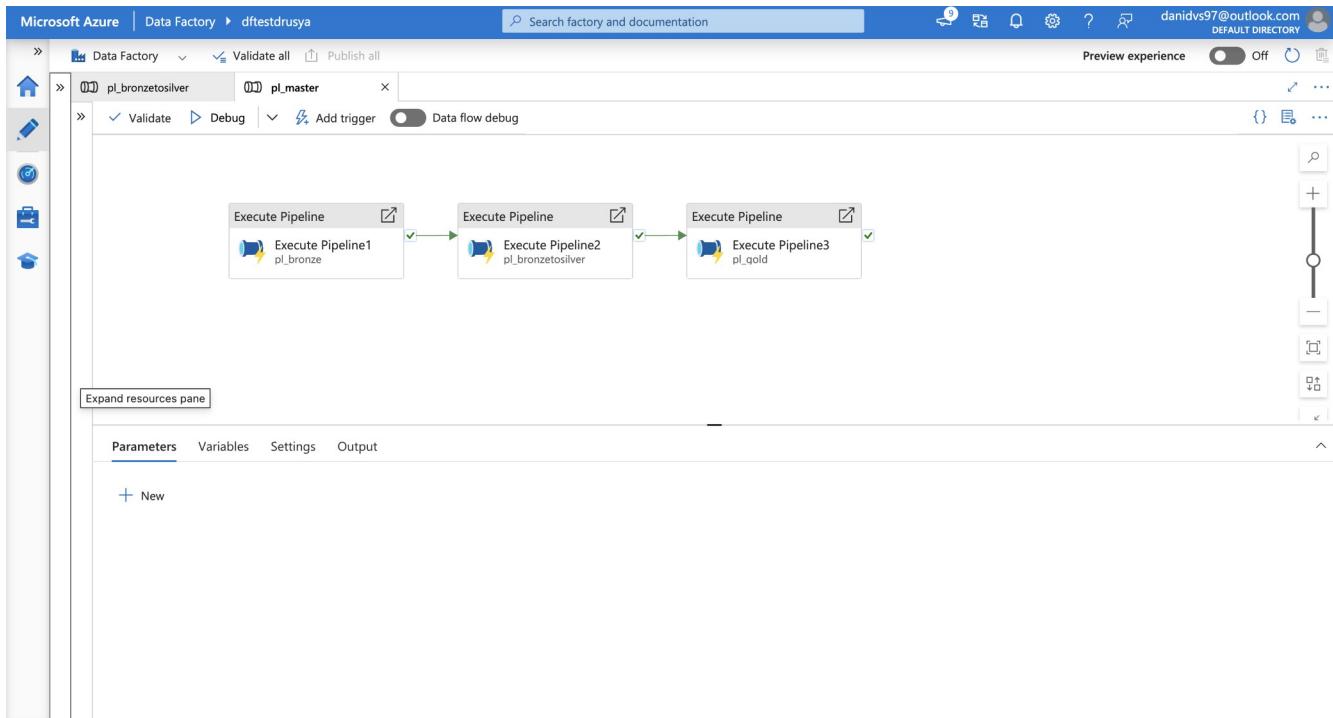






STEP – 4 : CREATED A MASTER PIPELINE

Now created a master pipeline and connected the child pipelines using the execute pipeline command and added a trigger to run the pipeline on a specified schedule.



This screenshot shows the 'New trigger' dialog box overlaid on the Data Factory interface. The dialog has a title 'New trigger' and several input fields:

- Name ***: A text input field containing 'trigger1'.
- Description**: A large empty text area.
- Type ***: A dropdown menu set to 'Schedule'.
- Start date ***: A date input field showing '4/21/2025, 11:41:09 PM'.
- Time zone ***: A dropdown menu set to 'Eastern Time (US & Canada) (UTC-5)'.
- Recurrence ***: A section with a 'Every' dropdown set to '15' and a 'Minute(s)' dropdown.
- Specify an end date**: A checkbox that is unchecked.
- Annotations**: A section with a '+ New' button.
- Start trigger**: A section with a checkbox labeled 'Start trigger on creation' which is checked.

At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Microsoft Azure | Data Factory > dfestdrusya

Search factory and documentation

Validate all Publish all Preview experience Off

Triggers

To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

+ New Refresh

Filter by name Annotations : Any

Showing 1 - 1 of 1 items

Name ↑	Type ↑↓	Status ↑↓	Related ↑↓	Annotations ↑↓
trigger1	Schedule	Started	1	

General

- Factory settings
- Connector upgrade advisor

Connections

- Linked services
- Integration runtimes
- Microsoft Purview

Source control

- Git configuration
- ARM template

Author

- Triggers
- Global parameters
- Data flow libraries

Security

- Credentials
- Customer managed key
- Outbound rules
- Managed private endpoints

Workflow orchestration manager

Microsoft Azure | Data Factory > dfestdrusya

Search factory and documentation

Validate all Publish all Preview experience Off

pl_gold pl_master

Add trigger Data flow debug

Execute Pipeline Execute Pipeline Execute Pipeline

Execute Pipeline1 pl_bronze → Execute Pipeline2 pl_bronzetrosilver → Execute Pipeline3 pl_gold

Parameters Variables Settings Output

Pipeline run ID: 0f63939c-5eb7-4d97-9d73-e67558f828a1

Pipeline status Succeeded

All status

Showing 1 - 3 of 3 items

Activity name ↑↓	Activity st... ↑↓	Activit... ↑↓	Run start ↑↓	Duration ↑↓	Integration runtime ↑↓	User prop... ↑↓	Activity run ID ↑↓
Execute Pipeline3	Succeeded	Execute Pipeline	4/20/2025, 8:01:54 PM	5m 0s		b27f569d-f23a-44af-a705-0a88fb3c9f	
Execute Pipeline2	Succeeded	Execute Pipeline	4/20/2025, 7:57:33 PM	4m 22s		6884df26-9e89-48a1-cbe-10dc13338f	
Execute Pipeline1	Succeeded	Execute Pipeline	4/20/2025, 7:56:33 PM	1m 0s		d547bb52-75fd-4156-963c-4749a2426	

View debug run consumption Monitor in Azure Metrics Export to CSV

OUTPUTS IN SQL DATABASE

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to 'sqlserverdrusya.database.windows.net' with database 'dbdrusya'. The query window contains the following SQL code:

```
SELECT * FROM ACCOUNTS_SCDTYPE1
```

The results pane displays the output of the query as a table with 21 rows. The columns are:

	account_id	customer_id	account_type	balance	CREATEDBY	UPDATEDBY	CREATEDDATE	UPDATEDDATE	HASHKEY
1	48	6	Checking	4900	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	1734299941
2	61	52	Savings	500.25	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	2662432370
3	62	35	Checking	6300.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	880230545
4	20	21	Checking	2000	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	2020209643
5	40	19	Checking	4100	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	2870805942
6	74	43	Checking	7500.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	4241197050
7	66	26	Checking	6700.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	372242038
8	72	17	Checking	7300	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	1425050024
9	87	93	Savings	825.75	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	3624526683
10	81	70	Savings	750.25	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	324394403
11	83	82	Savings	775.75	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	293251303
12	13	29	Savings	1300.25	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	3025925162
13	1	45	Savings	1000.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	2454403084
14	54	42	Checking	5500.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	221025946
15	42	36	Checking	4300.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	1993558944
16	78	4	Checking	7900.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	1790786833
17	58	16	Checking	5900.5	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	2917247935
18	3	78	Savings	1500	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	199654578
19	64	12	Checking	6500	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	1796018383
20	51	72	Savings	375.75	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	1266528029
21	56	78	Checking	5700	DATAFLOW	DATAFLOW	2025-04-20 21:19:44.657	2025-04-20 21:19:44.657	3877047592

sqlserverdrusya.database.windows.net

SQLQuery_1 - (77) s...drusya

1 SELECT * FROM CUSTOMERS_SCDTYPE1

	customer_id	first_name	last_name	address	city	state	zip	CREATEDBY	UPDATEDBY	CREATEDDATE	UPDATEDDATE
1	20	Mia	Nelson	1919 Birch Blvd	London	ON	N6A0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
2	40	Sophia	Rivera	3939 Poplar St	Milton	ON	L9T0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
3	20	Mia	Nelson	1919 Birch Blvd	London	ON	N6A0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
4	40	Sophia	Rivera	3939 Poplar St	Milton	ON	L9T0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
5	74	Harper	Graham	7373 Oak Dr	Bala	ON	P0C0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
6	74	Harper	Graham	7373 Oak Dr	Bala	ON	P0C0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
7	48	Harper	James	4747 Birch Blvd	Port Perry	ON	L9L0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
8	48	Harper	James	4747 Birch Blvd	Port Perry	ON	L9L0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
9	61	William	Butler	6060 Pine Rd	Alliston	ON	L9R0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
10	61	William	Butler	6060 Pine Rd	Alliston	ON	L9R0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
11	62	Ava	Simmons	6161 Birch Blvd	Angus	ON	L0M0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
12	62	Ava	Simmons	6161 Birch Blvd	Angus	ON	L0M0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
13	69	Joseph	Diaz	6868 Ash Blvd	Port McNicoll	ON	L0K0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
14	69	Joseph	Diaz	6868 Ash Blvd	Port McNicoll	ON	L0K0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
15	45	Christopher	Ward	4444 Maple Ave	Keswick	ON	L4P0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
16	45	Christopher	Ward	4444 Maple Ave	Keswick	ON	L4P0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
17	8	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
18	8	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
19	38	Isabella	Murphy	3737 Cypress Ave	Woodstock	ON	N4S0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267
20	38	Isabella	Murphy	3737 Cypress Ave	Woodstock	ON	N4S0A1	DATAFLOW	DATAFLOW	2025-04-21 00:05:52,267	2025-04-21 00:05:52,267

sqlserverdrusya.database.windows.net

SQLQuery_1 - (77) s...drusya

1 SELECT * FROM LOAN_PAYMENTS_SCDTYPE1

	payment_id	loan_id	payment_date	payment_amount	CREATEDBY	UPDATEDBY	CREATEDDATE	UPDATEDDATE	HASHKEY
1	61	72	2024-03-01	3100	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	3924611863
2	61	72	2024-03-01	3100	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	3924611863
3	62	83	2024-03-02	3150	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1029034042
4	62	83	2024-03-02	3150	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1029034042
5	74	15	2024-03-14	3750	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	3135475598
6	74	15	2024-03-14	3750	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	3135475598
7	20	21	2024-01-20	1050	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	161596424
8	40	41	2024-02-09	2050	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1343458395
9	20	21	2024-01-20	1050	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	161596424
10	40	41	2024-02-09	2050	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1343458395
11	69	60	2024-03-09	3500	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	491520019
12	69	60	2024-03-09	3500	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	491520019
13	45	96	2024-02-14	2300	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1621318141
14	45	96	2024-02-14	2300	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1621318141
15	66	27	2024-03-06	3350	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	994003179
16	66	27	2024-03-06	3350	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	994003179
17	72	93	2024-03-12	3650	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	3387588883
18	72	93	2024-03-12	3650	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	3387588883
19	14	55	2024-01-14	750	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	2348634532
20	14	55	2024-01-14	750	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	2348634532
21	13	44	2024-01-13	700	DATAFLOW	DATAFLOW	2025-04-21 00:05:33,240	2025-04-21 00:05:33,240	1774633368

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to `sqlserverdrusya.database.windows.net` and the database is `dbdrusya`. The query window contains the following SQL command:

```
1  SELECT * FROM LOANS_SCDTYPE1
```

The results pane displays the data from the `LOANS_SCDTYPE1` table. The table has 13 columns: `loan_id`, `customer_id`, `loan_amount`, `interest_rate`, `loan_term`, `CREATEDBY`, `UPDATEDBY`, `CREATEDDATE`, `UPDATEDDATE`, and `HASHKEY`. There are 28 rows of data, each with unique values for the columns.

	<code>loan_id</code>	<code>customer_id</code>	<code>loan_amount</code>	<code>interest_rate</code>	<code>loan_term</code>	<code>CREATEDBY</code>	<code>UPDATEDBY</code>	<code>CREATEDDATE</code>	<code>UPDATEDDATE</code>	<code>HASHKEY</code>
1	61	52	10000.25	5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2894988696
2	61	52	10000.25	5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2894988696
3	62	35	20000.5	4	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2629764668
4	62	35	20000.5	4	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2629764668
5	48	6	27500	3	24	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2806356369
6	48	6	27500	3	24	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2806356369
7	20	21	37500	3.5	24	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1388545671
8	40	19	37500	3	24	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	570153790
9	20	21	37500	3.5	24	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1388545671
10	40	19	37500	3	24	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	570153790
11	74	43	30000.5	4.5	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1215019206
12	74	43	30000.5	4.5	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1215019206
13	69	59	32500.25	5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1838491308
14	69	59	32500.25	5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1838491308
15	45	68	25000.25	5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1252029156
16	45	68	25000.25	5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	1252029156
17	66	26	17500.5	4.5	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2285969467
18	66	26	17500.5	4.5	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2285969467
19	38	15	27500.5	4	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2945825878
20	38	15	27500.5	4	48	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	2945825878
21	25	66	25000.25	5.5	36	DATAFLOW	DATAFLOW	2025-04-21 00:06:03.267	2025-04-21 00:06:03.267	5402737095

Below the results, the status bar shows: Ln 1, Col 1 (28 selected) Spaces: 4 UTF-8 LF 200 rows MSSQL 00:00:00 sqlserverdrusya.database.windows.net : dbdrusya (88)

The screenshot shows the SSMS interface with a query window titled 'SQLQuery_1 - (77) ...drusya'. The query is:

```
1 SELECT * FROM TRANSACTIONS_SCDTYPE2
```

The results grid displays the following columns: account_id, transaction_date, transaction_amount, transaction_type, CREATEDBY, UPDATEDBY, CREATEDDATE, UPDATEDDATE, ISACTIVE, and HASHKEY. The data includes various transactions such as Withdrawals and Deposits across different dates and amounts.

account_id	transaction_date	transaction_amount	transaction_type	CREATEDBY	UPDATEDBY	CREATEDDATE	UPDATEDDATE	ISACTIVE	HASHKEY
21	2024-01-20	375.25	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	199627346
19	2024-02-09	375.25	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	409636408
21	2024-01-20	375.25	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	199627346
19	2024-02-09	375.25	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	409636408
43	2024-03-14	300.25	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	201689898
43	2024-03-14	300.25	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	201689898
52	2024-03-01	100.5	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	224673097
52	2024-03-01	100.5	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	224673097
35	2024-03-02	200.75	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	332076348
35	2024-03-02	200.75	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	332076348
6	2024-02-17	275.75	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	128037333
6	2024-02-17	275.75	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	128037333
77	2024-03-31	100.5	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	913674472
77	2024-03-31	100.5	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	913674472
59	2024-03-09	325	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	266972271
59	2024-03-09	325	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	266972271
68	2024-02-14	250	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	319277867
68	2024-02-14	250	Deposit	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	319277867
26	2024-03-06	175	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	191387545
26	2024-03-06	175	Withdrawal	DATAFLOW	DATAFLOW	2025-04-21 00:02:07.307	2025-04-21 00:02:07.307	1	191387545

STEP 5: Visualization using power BI and fabric.

Downloaded power BI in the virtual machine and connected with my SQL database.

The screenshot shows the Power BI Desktop interface with the 'Navigator' pane open. The 'LOAN_PAYMENTS_SCDTYPE1' table is selected. The data grid displays columns: payment_id, loan_id, payment_date, payment_amount, CREATEDBY, and UPDATER. The data consists of multiple rows of loan payments.

payment_id	loan_id	payment_date	payment_amount	CREATEDBY	UPDATER
61	72	3/1/2024	3100	DATAFLOW	
61	72	3/1/2024	3100	DATAFLOW	
62	83	3/2/2024	3150	DATAFLOW	
62	83	3/2/2024	3150	DATAFLOW	
74	15	3/14/2024	3750	DATAFLOW	
74	15	3/14/2024	3750	DATAFLOW	
20	21	1/20/2024	1050	DATAFLOW	
40	41	2/9/2024	2050	DATAFLOW	
20	21	1/20/2024	1050	DATAFLOW	
40	41	2/9/2024	2050	DATAFLOW	
69	60	3/9/2024	3500	DATAFLOW	
69	60	3/9/2024	3500	DATAFLOW	
45	96	2/14/2024	2300	DATAFLOW	
45	96	2/14/2024	2300	DATAFLOW	
66	27	3/6/2024	3350	DATAFLOW	
66	27	3/6/2024	3350	DATAFLOW	
72	93	3/12/2024	3650	DATAFLOW	
72	93	3/12/2024	3650	DATAFLOW	
14	55	1/14/2024	750	DATAFLOW	
14	55	1/14/2024	750	DATAFLOW	
13	44	1/13/2024	700	DATAFLOW	
13	44	1/13/2024	700	DATAFLOW	
1	45	1/1/2024	100	DATAFLOW	

The screenshot shows the Power BI Desktop interface with several toolbars and panes:

- File, Home, Insert, Modeling, View, Optimize, Help, Format, Data / Drill, Table tools, Column tools**: The top navigation bar.
- Structure, Formatting, Properties**: The left-hand ribbon tabs.
- Summarization, Don't summarize, Data category, Uncategorized, Sort by column, Sort, Data groups, Manage relationships, Relationships, New column, Calculations**: The right-hand ribbon tabs.
- Visualizations pane**: Shows filters applied to the visual:
 - account_id is (All)
 - account_type is (All)
 - Sum of customer_id is (All)
 - Sum of customer_id is (All)
 - Sum of loan_id is (All)
 - Sum of loan_id is (All)
- Data pane**: Shows available columns:
 - UPDATEDBY
 - UPDATEDDATE
 - zip
 - LOAN_PAYMENTS_S...
 - CREATEDBY
 - CREATEDDATE
 - HASHKEY
 - loan_id
 - payment_amount
 - payment_id
 - payment_date
 - UPDATEDBY
 - LOANS_SCDTYPE1
 - CREATEDBY
 - CREATEDDATE
 - CUSTOMER_ID
 - HASHKEY
 - INTEREST_RATE
 - LOAN_AMOUNT
 - loan_id
 - loan_term
 - UPDATEDBY
 - UPDATEDDATE
 - TRANSACTIONS_SC...
- Table visualization**: A table with the following data:

	account_id	account_type	Sum of loan_id	Sum of customer_id	Sum of customer_id	Sum of loan_id
1	9966	Savings	7482	9874	9874	10100
2	9966	Checking	7482	9874	9874	10100
3	9966	Savings	7482	9874	9874	10100
4	9966	Checking	7482	9874	9874	10100
5	9966	Savings	7482	9874	9874	10100
6	9966	Checking	7482	9874	9874	10100
7	9966	Savings	7482	9874	9874	10100
8	9966	Checking	7482	9874	9874	10100
9	9966	Savings	7482	9874	9874	10100
10	9966	Checking	7482	9874	9874	10100
11	9966	Savings	7482	9874	9874	10100
12	9966	Checking	7482	9874	9874	10100
13	9966	Savings	7482	9874	9874	10100
14	9966	Checking	7482	9874	9874	10100
15	9966	Savings	7482	9874	9874	10100
16	9966	Checking	7482	9874	9874	10100
17	9966	Savings	7482	9874	9874	10100
Total		9966	7482	9874	9874	10100
- Page navigation**: Page 1 of 1.
- Search bar**: Type here to search.
- System status**: 6:08 PM, ENG, 4/26/2025.

The screenshot shows the Microsoft Power BI desktop application interface. A central modal dialog box is open, displaying the message "Publishing to Power BI" with a green checkmark icon and the text "Success! Open 'project1.pbix' in Power BI". Below this, there are links to "Get Quick Insights" and a "Did you know?" section with the text "You can create a portrait view of your report, tailored for mobile phones. On the View tab, select Mobile Layout. Learn more". At the bottom right of the modal is a green "Got it" button.

The main workspace shows a data grid with the following columns and data:

	account_id	account_type	Sum of loan_id	Sum of customer_id	Sum of loan_id
1	Savings	9966	7482	9874	10100
2	Savings	9966	7482	9874	10100
3	Savings	9966	7482	9874	10100
4	Checking	9966	7482	9874	10100
5	Savings	9966	7482	9874	10100
6	Checking	9966	7482	9874	10100
7	Savings	9966	7482	9874	10100
8	Checking	9966	7482	9874	10100
9	Savings	9966	7482	9874	10100
10	Checking	9966	7482	9874	10100
11	Savings	9966	7482	9874	10100
12	Checking	9966	7482	9874	10100
13	Savings	9966	7482	9874	10100
14	Checking	9966	7482	9874	10100
15	Savings	9966	7482	9874	10100
16	Checking	9966	7482	9874	10100
17	Savings	9966	7482	9874	10100
Total		9966	7482	9874	10100

The ribbon menu at the top includes File, Home, Insert, Modeling, View, Optimize, Help, Format, and Data / Drill. The Home tab is selected. The ribbon also contains various icons for data sources (Clipboard, Cut, Copy, Format painter, Paste), data management (Get data, Excel, OneLake, SQL Server, Enter data, Dataverse, Recent sources), queries (Transform data, Refresh data, New visual, Text box, More visuals), calculations (New visual calculation, New measure, Quick measure, Sensitivity, Publish, Copilot), and sharing (Copilot).

The right-hand pane displays the "Visualizations" and "Data" sections of the Power BI service. The "Visualizations" section shows a grid of visualization icons. The "Data" section lists various data sources and fields, such as "LOAN_PAYMENTS_S...", "UPDATEDBY", "UPDATEDDATE", "zip", "CREATEDBY", "CREATEDDATE", "HASHKEY", "loan_id", "payment_amount", "payment_id", "UPDATEDBY", "LOANS_SCOTYPE1", "CREATEDBY", "CREATEDDATE", "customer_id", "HASHKEY", "interest_rate", "loan_amount", "loan_id", "loan_term", "UPDATEDBY", "TRANSACTIONS_SC...", and "TRANSACTIONS_SC...".

Publish the report and open in fabric.

The screenshot shows the Power BI desktop interface. On the left, there's a navigation pane with options like Home, Workspaces, OneLake, Monitor, Real-Time, Workloads, My workspace, and project1 (which is selected). The main area displays a table with columns: account_id, account_type, Sum of loan_id, Sum of customer_id, Sum of customer_id, and Sum of loan_id. The table contains 16 rows of data, with the last row being a total. To the right of the table is a filter pane titled "Filters on this visual" containing six filter conditions: account_id is (All), account_type is (All), Sum of customer_id is (All), Sum of customer_id is (All), Sum of loan_id is (All), and Sum of loan_id is (All). The top bar shows "project1 | Data updated 4/26/25" and a search bar. The bottom right corner shows a zoom level of 94%.

Github integration

The screenshot shows the Microsoft Azure Data Factory settings page for a data factory named "dftestdrusya". The left sidebar lists various configurations: General, Factory settings, Connector upgrade advisor, Connections, Integration runtimes, Microsoft Purview, Source control (which is expanded to show Git configuration, ARM template, Author, Triggers, Global parameters, Data flow libraries, Security, Credentials, Customer managed key, Outbound rules, Managed private endpoints, and Workflow orchestration manager), and Workflow orchestration manager. The main content area is titled "Git repository" and describes it as "Git repository information associated with your data factory. CI/CD best practices". It shows the following details: Repository type is GitHub, GitHub account is Drusya-23, Repository name is Data-engineering-projects, Collaboration branch is Bootcamp_projects, Publish branch is adf_publish, Root folder is /Customer_Account_Analysis_pipeline_, Last published commit is 41155b645fe68964fbfc88508db44bf427a9959, Publish (from ADF Studio) is Enabled, and Custom comment is Enabled. The top bar shows "Microsoft Azure | Data Factory > dftestdrusya", a search bar, and user info "danidvs97@outlook.com DEFAULT DIRECTORY". The bottom right shows "Preview experience" toggle set to "Off".

CONCLUSION

Successfully completed the customer account analysis using data factory. Confirmed the data was cleaned, transformed and loaded to the SQL database from on-premise.