

Project Name: Data Pipeline for Customer Account Analysis.

Sudhanshu Kharbanda

Objective

The objective of this project is to design and implement a robust and scalable **data pipeline** for processing **customer account data**. The pipeline performs the following functions:

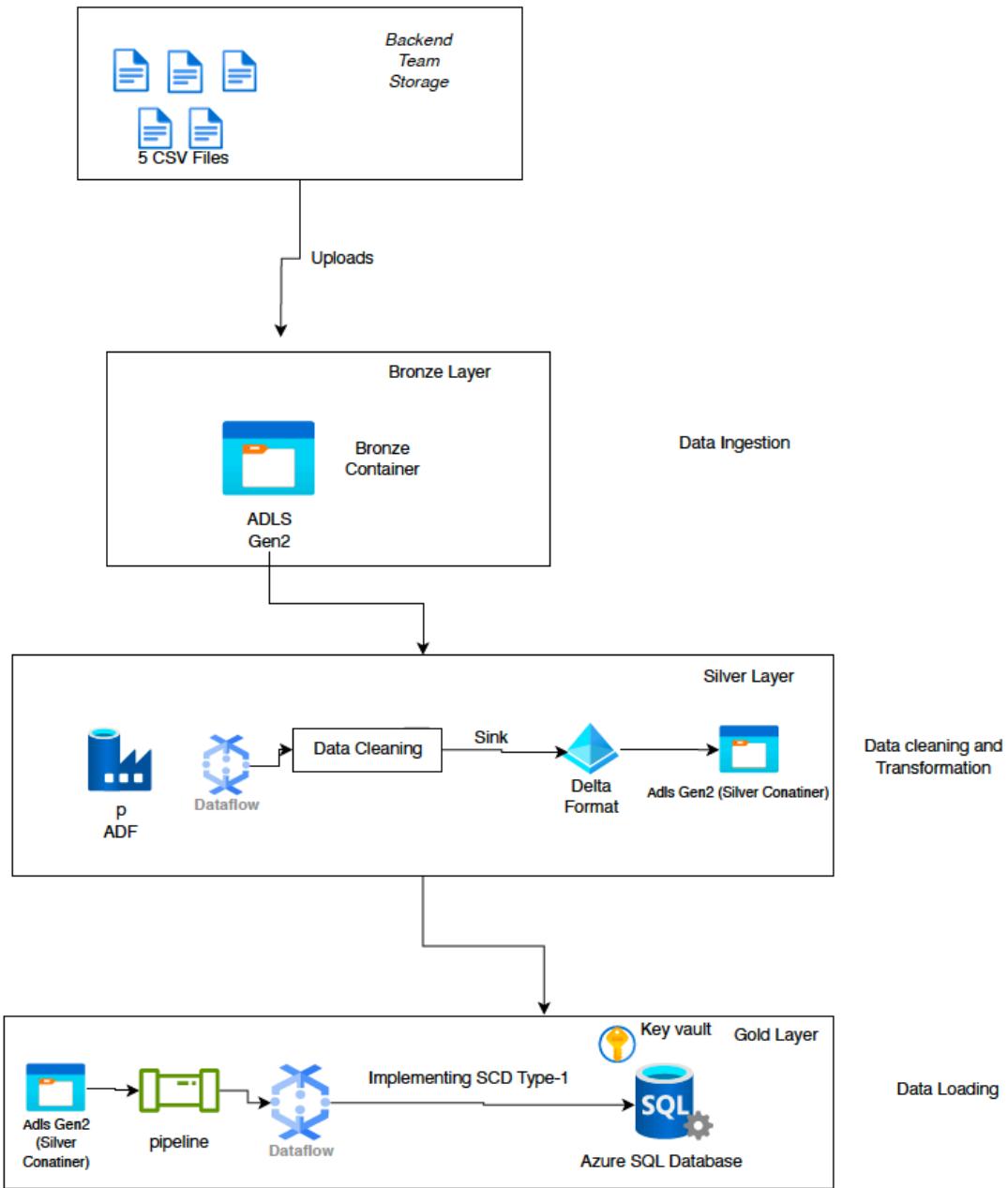
- **Data Ingestion:** Copy raw data from a backend team's Azure Storage Account to a **Data Lake Raw (Bronze) container**.
- **Data Cleaning & Transformation:** Use **Azure Data Factory (ADF)** Data Flows to clean and transform the ingested data.
- **ETL Processing:** Prepare curated (Silver) datasets and upsert (Insert or Update) them into a **SQL Database** using **Slowly Changing Dimension Type 1 (SCD-1)** logic.
- **Security:** Secure credentials and connection strings using **Azure Key Vault**.

This solution ensures **accuracy, efficiency, and reliability** in data handling, enabling downstream analytics and reporting.

Tools & Services Used

1. **Azure Data Factory (ADF)** – For data ingestion, cleaning, transformation, and loading.
2. **Azure Data Lake Storage (ADLS)** – Organized in a **Bronze-Silver-Gold** architecture.
3. **Azure SQL Database** – Final destination for cleaned and transformed data.
4. **Azure Key Vault** – Secure storage for credentials and connection strings.
5. **Delta/Parquet File Format** – Optimized storage format for transformations and analytics.
6. **Draw.io** – For creating an Architecture diagram.

Specifications Project Workflow and Architecture diagram



Step 1 - Data Ingestion (Backend Storage → Bronze Layer)

- 
- **Purpose:** Copy raw data from the backend team's Azure Storage account to the Bronze container in ADLS.
 - **Process:**
 - Used **ADF Copy Activity** to transfer files from the backend team's container to the Bronze layer.
 - Source Files:
 - `accounts.csv`
 - `customers.csv`
 - `loan_payments.csv`
 - `loans.csv`
 - `transactions.csv`
 - Sink: ADLS Raw (Bronze) Container.

Step 2 – Data Cleaning & Transformation (Bronze → Silver Layer)

- **Purpose:** Ensure data quality before further processing.
- **Actions Taken:**
 - Read data from Bronze container.
 - Applied transformations using **ADF Data Flow**:
 - Removed duplicate rows.
 - Null checks for primary keys
 - Stored cleaned datasets in the **Curated (Silver) container** in **Delta format**.



Step 3 – Loading into SQL DB with SCD-1 Logic (Silver → SQL DB(Gold))

- **Purpose:** Maintain updated records by replacing old data with new updates.
- **Process:**
 - Created an **ADF Pipeline** to read Delta files from the **Silver** layer.
 - Implemented **SCD Type 1**:
 - **Insert** new records if they do not exist in the target table.
 - **Update** existing records if primary key matches.
 - Executed pipeline to load initial dataset into SQL DB.
 - Modified source file (Bronze) to:
 - Add a **new record**.
 - Update an **existing record**.
 - Reran the pipeline to verify **upsert functionality** worked as expected.

Step 4 - Security Implementation

- All connection strings, SQL credentials, and storage keys were securely stored in **Azure Key Vault**.
- ADF pipelines referenced secrets dynamically from Key Vault, preventing hardcoding of sensitive information.

Implementation screenshots

Creating Silver and Bronze containers in the ADLS Gen2

The screenshot shows the Azure Storage Explorer interface for the 'adlsprodabrics' storage account. The 'Containers' section is selected. Four containers have been created: 'logs', 'bronze', 'input', and 'silver'. Each container has a timestamp indicating its creation or last modification.

Name	Last modified	Anonymous access level	Lease state
logs	7/28/2025, 5:18:41 PM	Private	Available
bronze	8/11/2025, 2:14:10 PM	Private	Available
input	7/28/2025, 5:22:09 PM	Private	Available
silver	8/11/2025, 1:54:32 PM	Private	Available

Inside the bronze containers 5 directories has been created with specific directories for each csv file.

The screenshot shows the 'bronze' container details. Inside the 'bronze' container, five sub-directories have been created: 'Accounts', 'customers', 'loan_payments', 'loans', and 'transactions'. Each directory was created on August 11, 2025, at different times between 2:14:32 PM and 2:15:29 PM.

Name	Last modified	Access tier	Blob type	Size	Lease state
Accounts	8/11/2025, 2:14:32 PM				...
customers	8/11/2025, 2:14:42 PM				...
loan_payments	8/11/2025, 2:14:52 PM				...
loans	8/11/2025, 2:14:59 PM				...
transactions	8/11/2025, 2:15:29 PM				...

Here each csv files has been uploaded to their specific folders.

The screenshot shows the 'bronze' container details again. Inside the 'Accounts' folder, a single CSV file named 'accounts.csv' has been uploaded. The file was uploaded on August 11, 2025, at 9:55:38 PM.

Name	Last modified	Access tier	Blob type	Size	Lease state
accounts.csv	8/11/2025, 9:55:38 PM	Hot (Inferred)	Block blob	2.32 KB	Available

Creating a silver container and creating directories for each of the files.

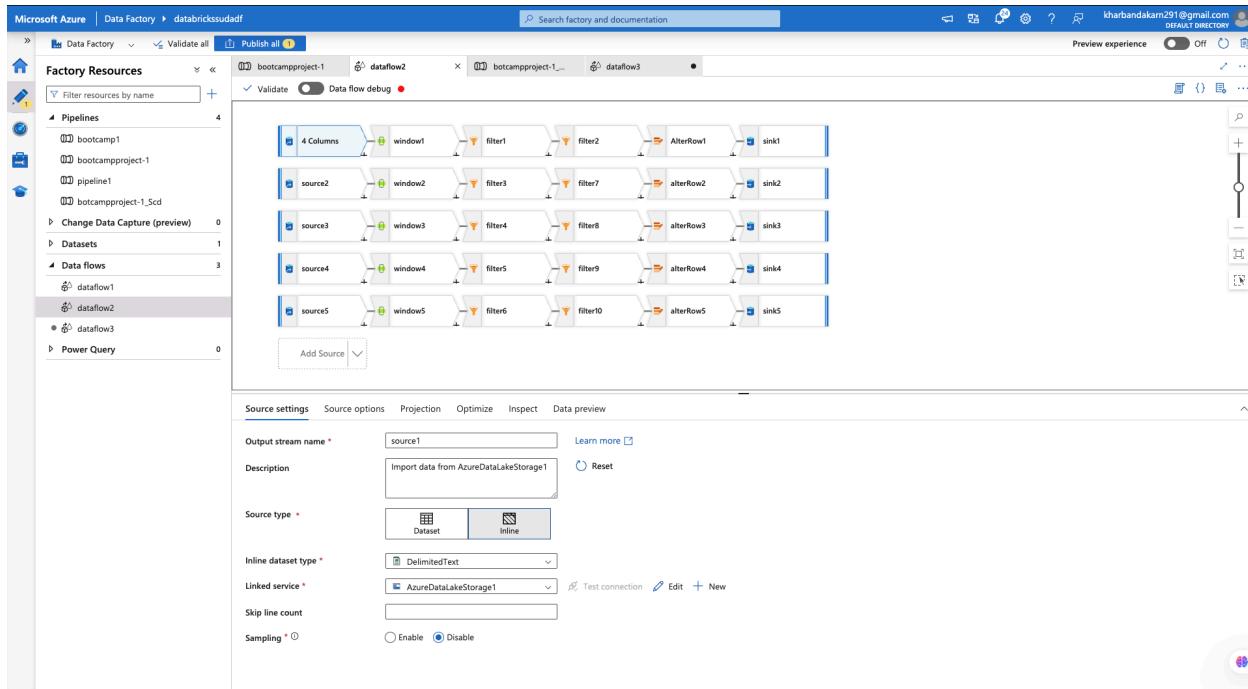
Name	Last modified	Access tier	Blob type	Size	Lease state
Accounts	8/11/2025, 2:09:08 PM				...
Customers	8/11/2025, 2:10:12 PM				...
loan.payments	8/11/2025, 2:10:34 PM				...
loans	8/11/2025, 2:10:41 PM				...
transactions	8/11/2025, 2:10:52 PM				...

After data cleaning the file will be saved in delta format in the silver container.

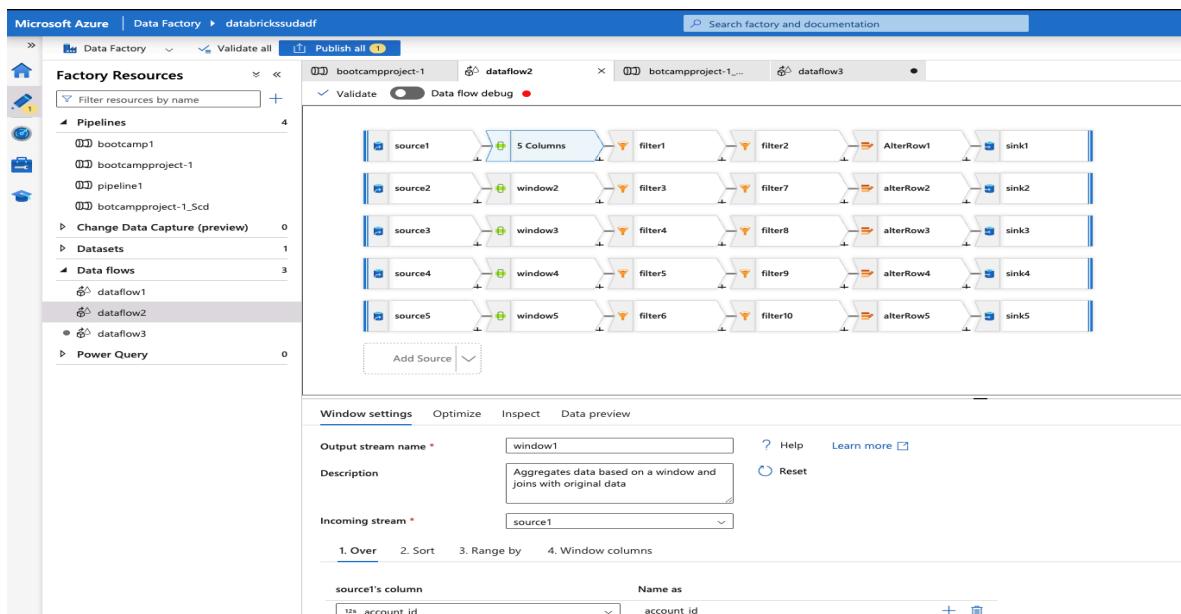
Name	Last modified	Access tier	Blob type	Size	Lease state	
... (many rows)	8/11/2025, 2:52:20 PM				...	
... (many rows)	8/11/2025, 3:32:49 PM				...	
... (many rows)	8/12/2025, 3:48:25 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 10:01:13 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 3:32:52 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 10:01:13 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 2:52:23 PM	Hot (Inferred)	Block blob	2.65 KIB	Available	...
... (many rows)	8/11/2025, 3:32:54 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 8:32:45 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 7:44:50 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 9:39:43 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 9:39:41 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/12/2025, 3:48:27 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 7:44:50 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 8:32:47 PM	Hot (Inferred)	Block blob	688 B	Available	...
... (many rows)	8/11/2025, 9:39:43 PM	Hot (Inferred)	Block blob	1.48 KIB	Available	...

Here are the five pipelines which is created in a single data flow where we are removing duplicates and null values for the primary column of each csv files.

For the source select source dataset type as delimited text and select azure datalake storage as linked service.



After loading the data window activity is used for removing the duplicates where account id column is choosed where window function is applied.



All the partitions are storted according to the accountid in ascending order.

The screenshot shows the 'Window settings' configuration for a window named 'window1'. The 'Incoming stream' is 'source1'. The 'Sort' tab is selected, showing a single column 'account_id' sorted in ascending order. The 'Nulls first' option is checked. There are tabs for 'Over', 'Sort', 'Range by', and 'Window columns'.

source1's column	Order	Nulls first
12s account_id	Ascending	<input checked="" type="checkbox"/>

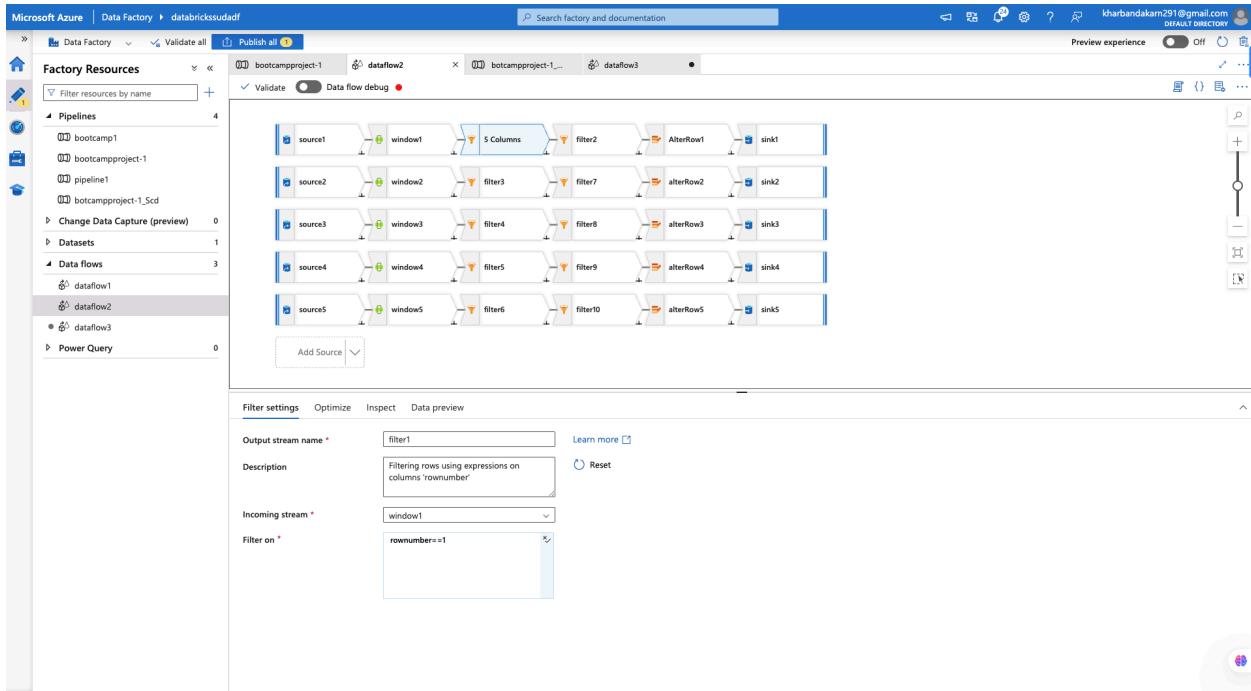
Creating a column which will give rank to each element in partition.

The screenshot shows the 'Window settings' configuration for a window named 'window1'. The 'Incoming stream' is 'source1'. The 'Window columns' tab is selected, showing a new column 'rownumber' with the expression 'rowNumber()'. There are tabs for 'Over', 'Sort', 'Range by', and 'Window columns'.

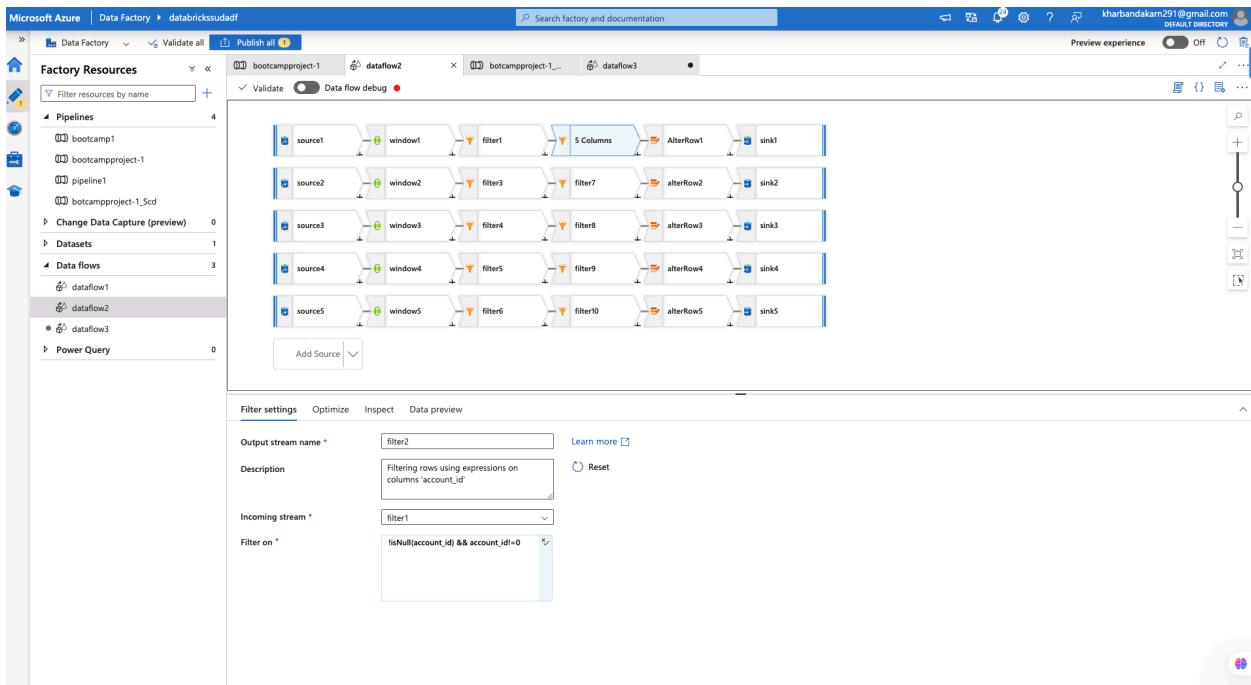
Column	Expression
rownumber	rowNumber()

Buttons at the bottom include '+ Add', 'Clone', 'Delete', and 'Open expression builder'.

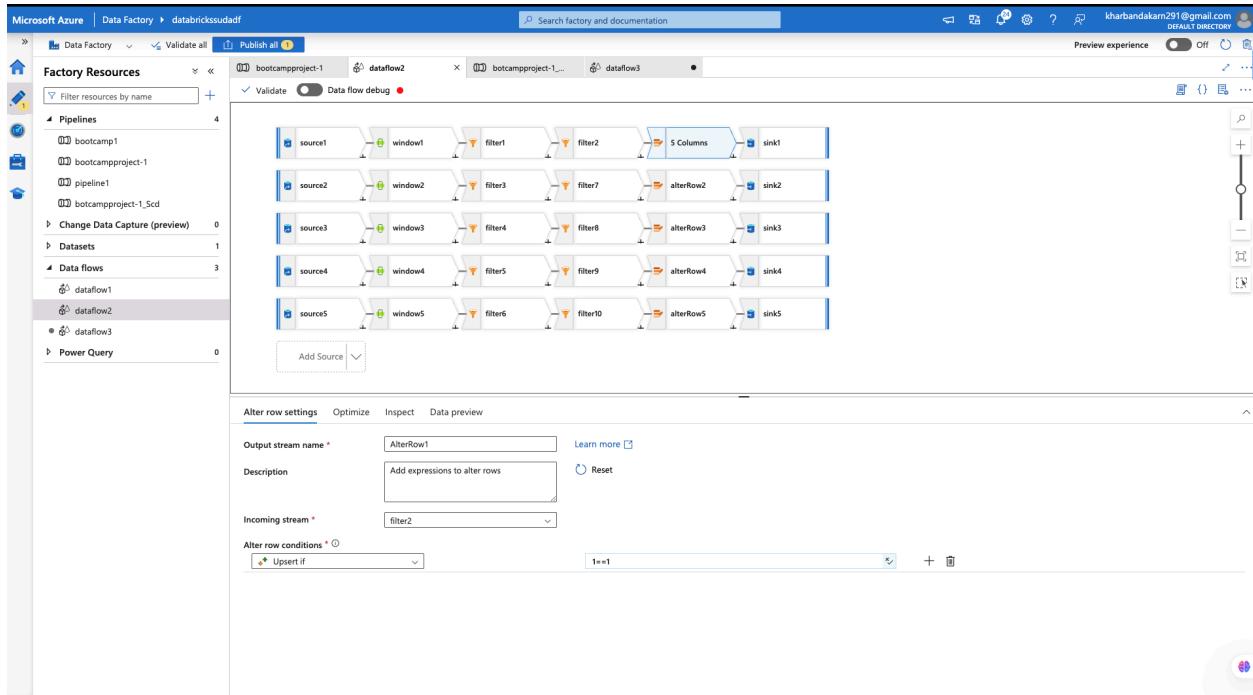
Now filter activity is added where we are filtering rows that is having rownumber=1, basically here we are filtering duplicates rows.



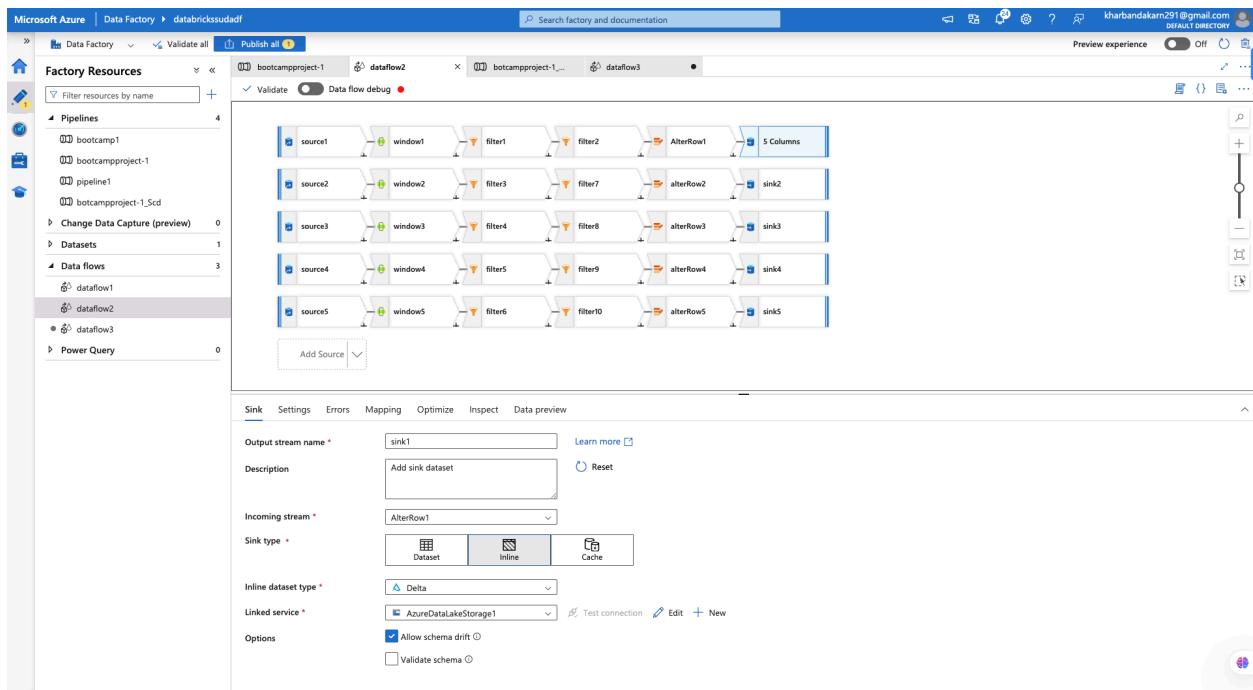
In this filter activity we are removing all the account id with null values and having value = 0.



Since we are implementing upsert in sink activity alter row activity is added where we alter row conditions as upsert.



Now we are adding sink activity where specifying the format in which data has been saved.



Here we are providing the path of the silver container where all the data has been stored. Here upsert method will be used since there will be updated and new rows will be added into the system.

The screenshot shows the Microsoft Azure Data Factory Data Flow blade. On the left, the navigation pane lists Pipelines, Datasets, Data flows, and Power Query. In the center, a complex data flow is displayed with five parallel source components (source1 to source5) followed by various processing steps like windows, filters, and alterRow operations, leading to five sink components (sink1 to sink5). Below the data flow, the 'Sink' tab is selected in the ribbon. The 'Settings' section is expanded, showing the 'Folder path' set to 'silver / Accounts'. Under the 'Table action' section, the 'Allow upsert' checkbox is checked, indicating that the data flow will handle both updates and inserts. Other options like 'Allow insert' and 'Allow delete' are also present but unchecked.

Here structure of the data that has been saved in the delta format.

The screenshot shows the Microsoft Azure Data Factory Data Flow blade with the 'Inspect' tab selected in the ribbon. The 'Schema' section is active, displaying the structure of the data. It shows 5 columns: 'account_id', 'customer_id', 'account_type', 'balance', and 'rownumber'. The 'Input' tab is selected, showing the schema details. The table below provides a detailed view of the schema:

Order	Column	Type	Updated	Total
1	account_id	short		1
2	customer_id	short		1
3	account_type	string		1
4	balance	double		1
5	rownumber	integer		1

Now we are adding source activity for loading second file.

The screenshot shows the Microsoft Azure Data Factory Data Flow blade. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, and Power Query. Under 'Data flows', 'dataflow2' is selected. The main workspace displays a data flow diagram with five parallel source activities (source1 to source5) followed by various processing steps like windowing, filtering, and altering rows, leading to five sinks (sink1 to sink5). A modal dialog is open at the bottom, titled 'Source settings', with the following configuration:

- Output stream name:** source2
- Description:** Import data from AzureDataLakeStorage1
- Source type:** Dataset (selected)
- Inline dataset type:** DelimitedText
- Linked service:** AzureDataLakeStorage1 (selected)
- Skip line count:** (empty input field)
- Sampling:** (radio button set to) Disable

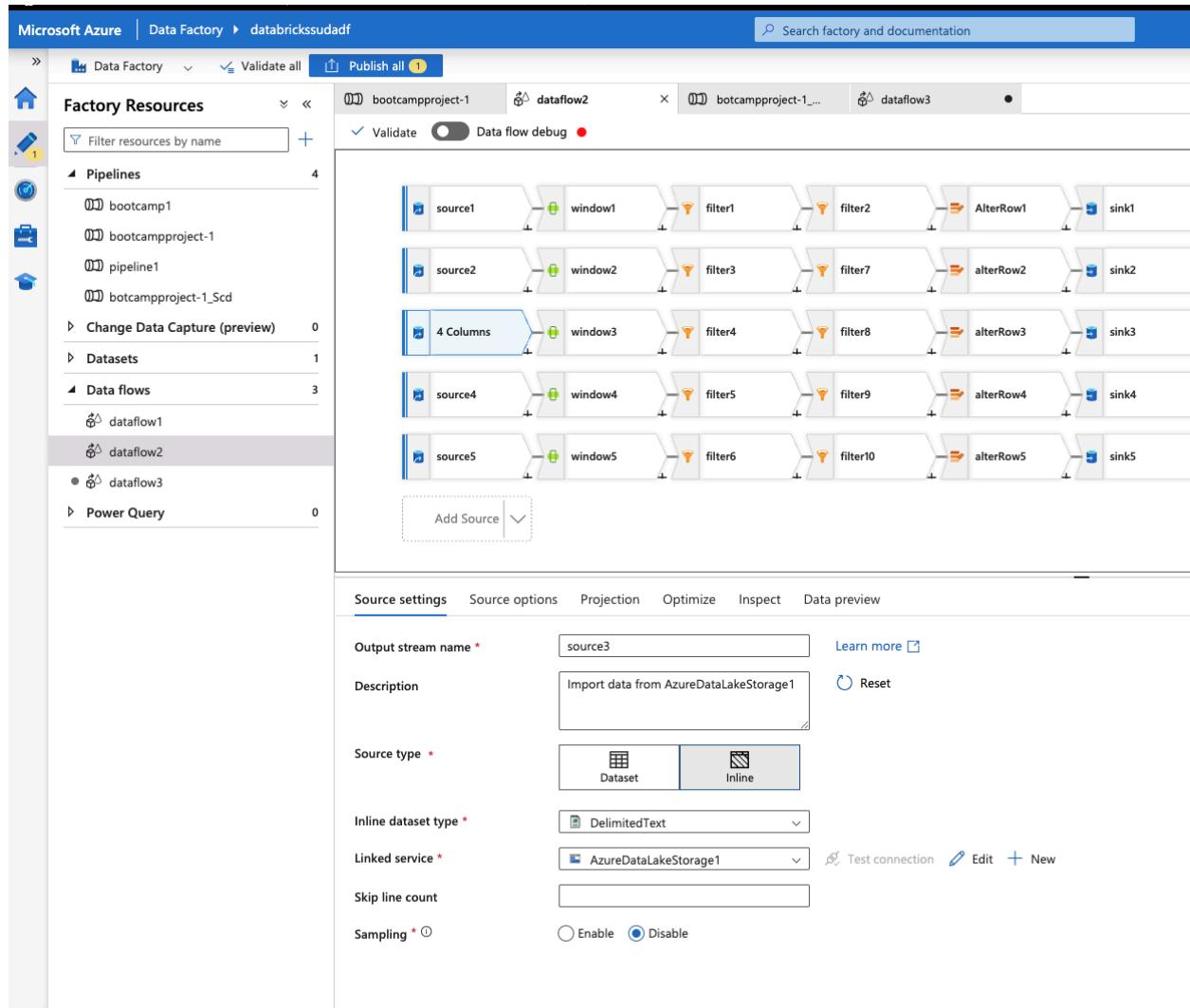
Here we are providing location of the file from where it has to be loaded from bronze layer.

The screenshot shows the Microsoft Azure Data Factory Data Flow blade with the same layout and context as the previous screenshot. The 'Source settings' dialog is now open, showing the 'Source options' tab. Under 'File settings', the following parameters are configured:

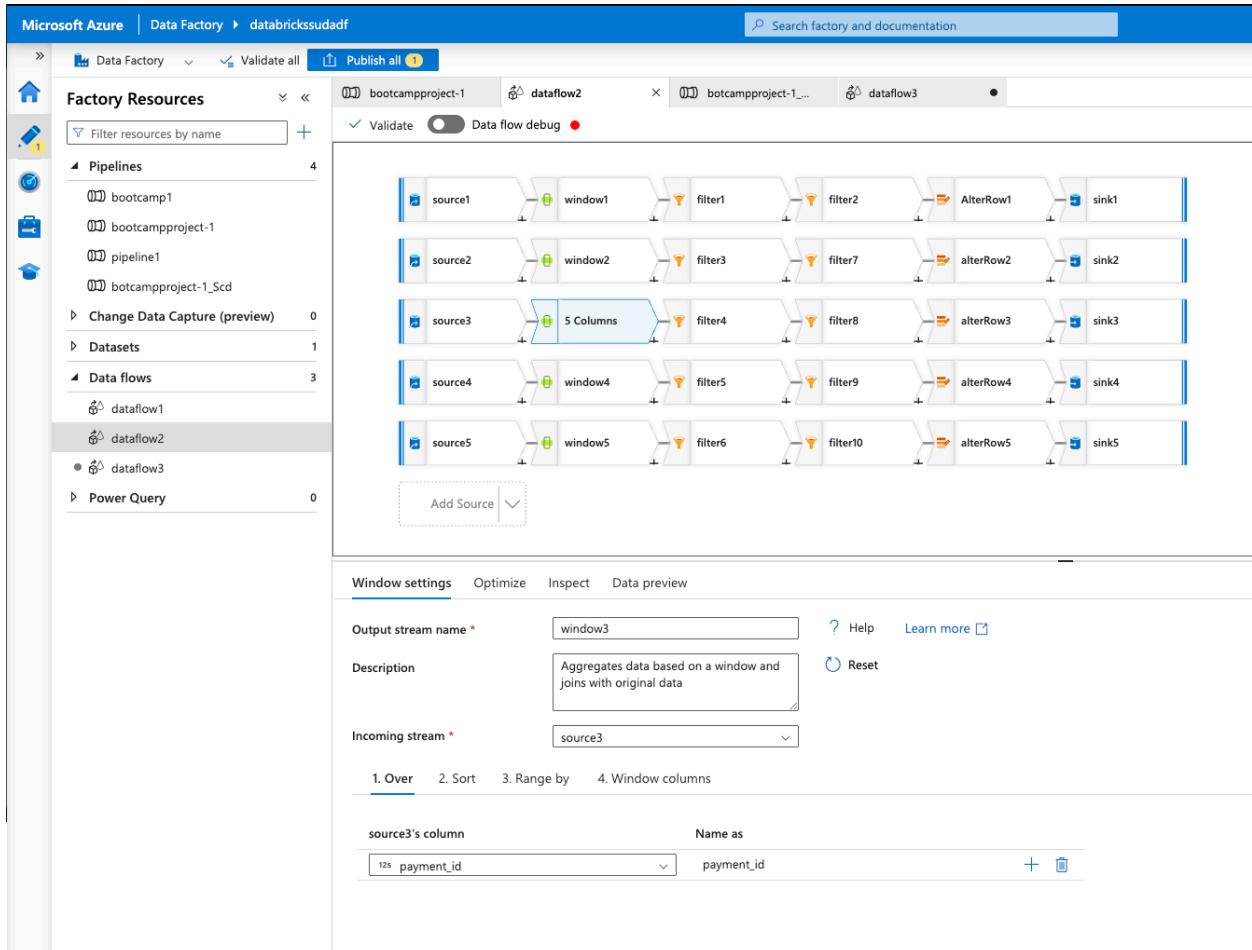
- File mode:** File (radio button selected)
- File path:** bronze / customers / customers.csv
- Allow no files found:** (checkbox unchecked)
- Change data capture:** (checkbox unchecked)
- Compression type:** No compression
- Encoding:** Default(UTF-8)
- Column delimiter:** Comma (,)
- Row delimiter:** Default (\r\n or \n\r)
- Quote character:** Double quote (")
- Escape character:** Backslash (\)

We will be doing filtering of data and saving it to the silver layer in the same way that we have implemented in previous steps. So only source activities for the other pipelines will shown below.

Here is the activity that will be loading the payments.csv file.



Here windows activity is checking the duplicates rows.



Here is the path from where loan_payments.csv file is loaded.

The screenshot shows the Microsoft Azure Data Factory Data Flow blade. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, and Power Query. Under 'Data flows', 'dataflow2' is selected. The main area displays five parallel data flow components, each consisting of a source, a window, one or more filters, and an alterRow component followed by a sink. The sources are labeled source1 through source5. The windows are labeled window1 through window5. The filters are labeled filter1 through filter10. The sinks are labeled sink1 through sink5. Below the data flows, the 'Source options' tab is selected, showing settings for 'File mode' (File selected), 'File path' (bronze / loan_payments / loan_payments.csv), and other encoding and delimiter options. A 'Browse' button is also present.

Here we are defining source for Fourth csv file.

The screenshot shows the Microsoft Azure Data Factory Data Flow blade with 'dataflow2' selected. The 'Source options' tab is active, showing the configuration for the fourth data flow, labeled 'source4'. The 'Output stream name' is set to 'source4'. The 'Description' field contains the text 'Import data from AzureDataLakeStorage1'. Under 'Source type', the 'Dataset' option is selected. The 'Inline dataset type' is set to 'DelimitedText'. The 'Linked service' is set to 'AzureDataLakeStorage1'. The 'Sampling' section has 'Disable' selected. Other fields like 'Description', 'Source type', 'Inline dataset type', 'Linked service', 'Skip line count', and 'Sampling' are also visible.

Here we are providing the path for loans.csv file from where it has been loaded.

The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists several pipelines, datasets, and data flows. The 'Data flows' section is expanded, showing three data flows: dataflow1, dataflow2, and dataflow3. The 'dataflow2' card is selected. The main workspace displays five parallel data flow activities. Each activity consists of a source (source1 to source5), followed by a sequence of windows, filters, and alterRow steps, leading to a sink (sink1 to sink5). The 'Source options' tab is active for the first activity, showing the following configuration:

- File settings**
- File mode**: File (radio button selected)
- File path**: bronze / loans / loans.csv
- Allow no files found**: Unchecked
- Change data capture**: Unchecked
- Compression type**: No compression
- Encoding**: Default(UTF-8)
- Column delimiter**: Comma (,)
- Row delimiter**: Default (\r,\n, or \n)
- Quote character**: Double quote ("")
- Escape character**: \

This source activity is used for loading the 5th csv file.

The screenshot shows the Microsoft Azure Data Factory pipeline editor with the same interface as the previous screenshot. The 'dataflow2' card is selected. The main workspace shows the five parallel data flow activities. The 'Source settings' tab is now selected for the fifth activity (source5), displaying the following configuration:

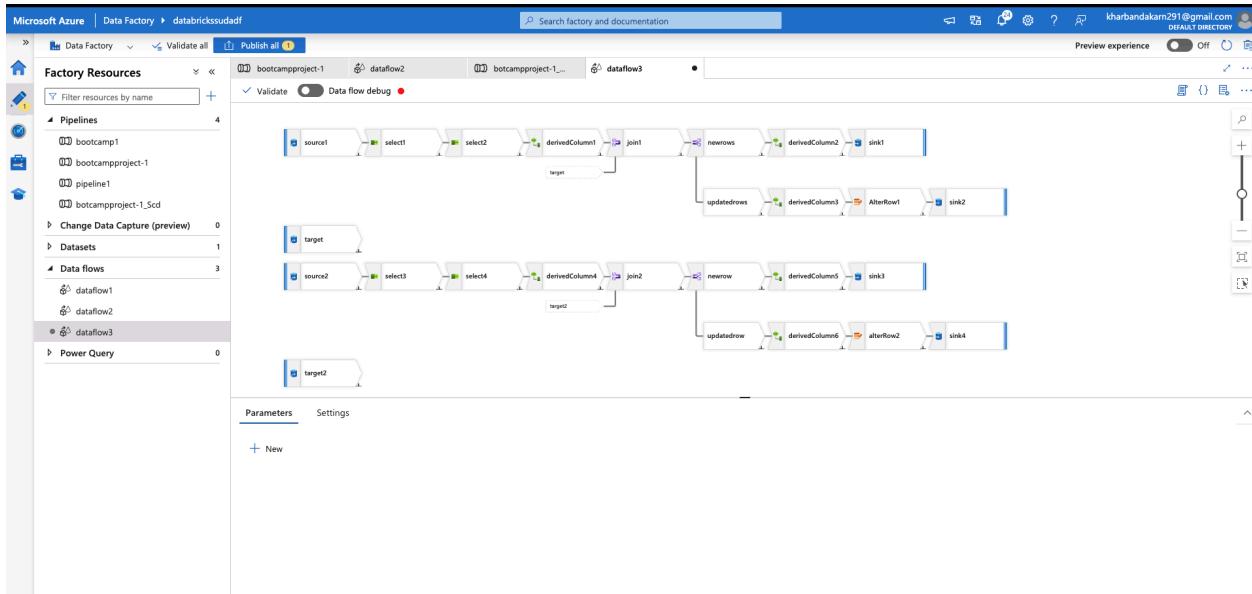
- Output stream name**: source5
- Description**: Import data from AzureDataLakeStorage1
- Source type**: Dataset
- Inline dataset type**: DelimitedText
- Linked service**: AzureDataLakeStorage1
- Skip line count**: (empty input field)
- Sampling**: (radio buttons for Enable and Disable)

Now we are providing the path for the loans.csv file from where it has to loaded.

We are creating a new data flow in the new pipeline form where SCD type-1 will be implemented.

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID
Data flow1	Succeeded	Data flow	8/12/2025, 3:59:01 PM	9m 43s	AutoResolveIntegrationRuntime (North Europe)		77106503-2212-4da3-8a29-effd2d67

We are going to implement SCD type -1 in 2 of the 5 csv files so here is how the pipeline will look like inside the dataflow.



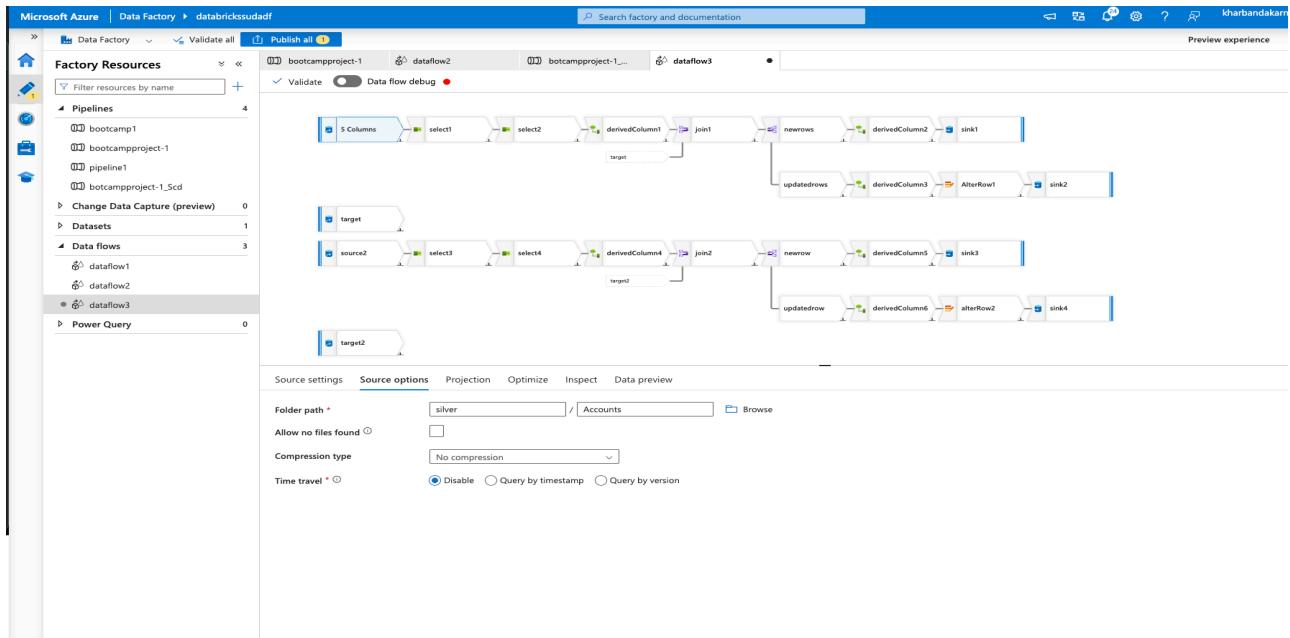
Source activity has been added that will loading data from delta format present in the silver container.

This screenshot shows the configuration pane for the 'source1' activity within the 'dataflow2' data flow. The configuration tabs at the top include 'Source settings', 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview'. The 'Source settings' tab is selected. The configuration fields are as follows:

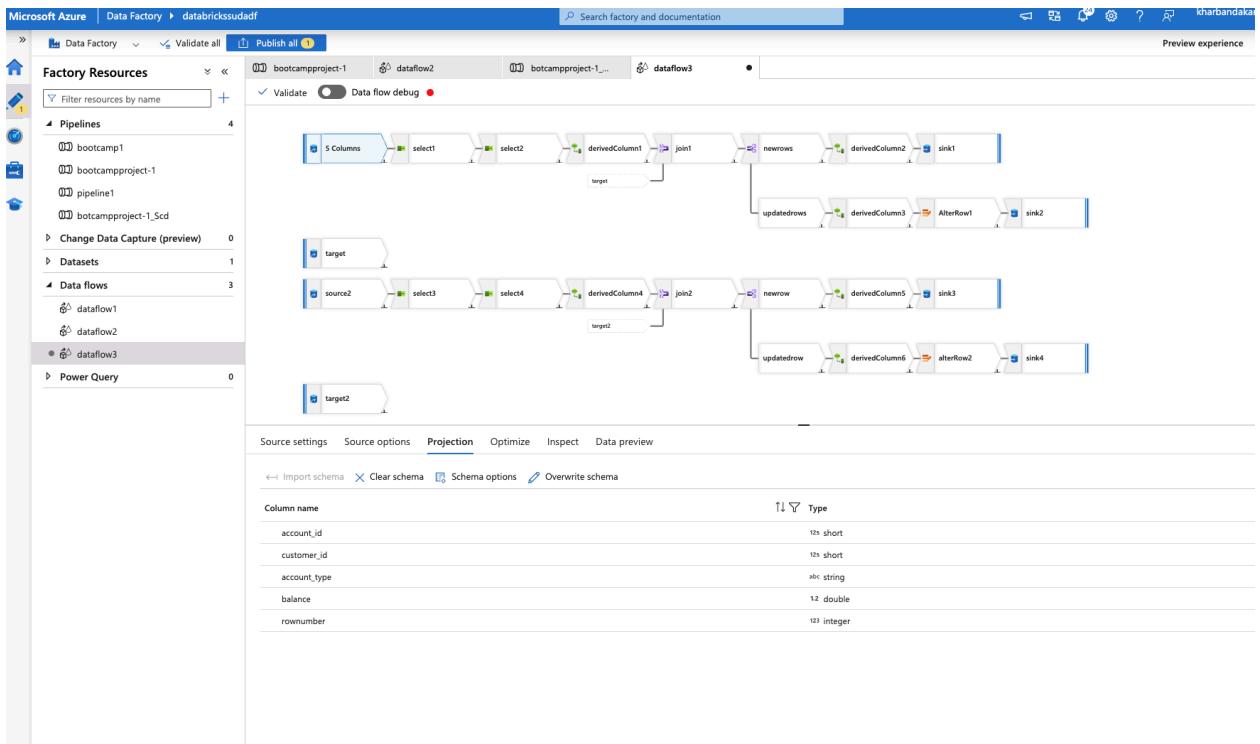
- Output stream name:** source1
- Description:** Import data from AzureDataLakeStorage1
- Source type:** Dataset (selected)
- Inline dataset type:** Delta
- Linked service:** AzureDataLakeStorage1
- Sampling:** Disable (selected)

The main data flow diagram is visible in the background, showing the parallel source paths and their connection to the target areas.

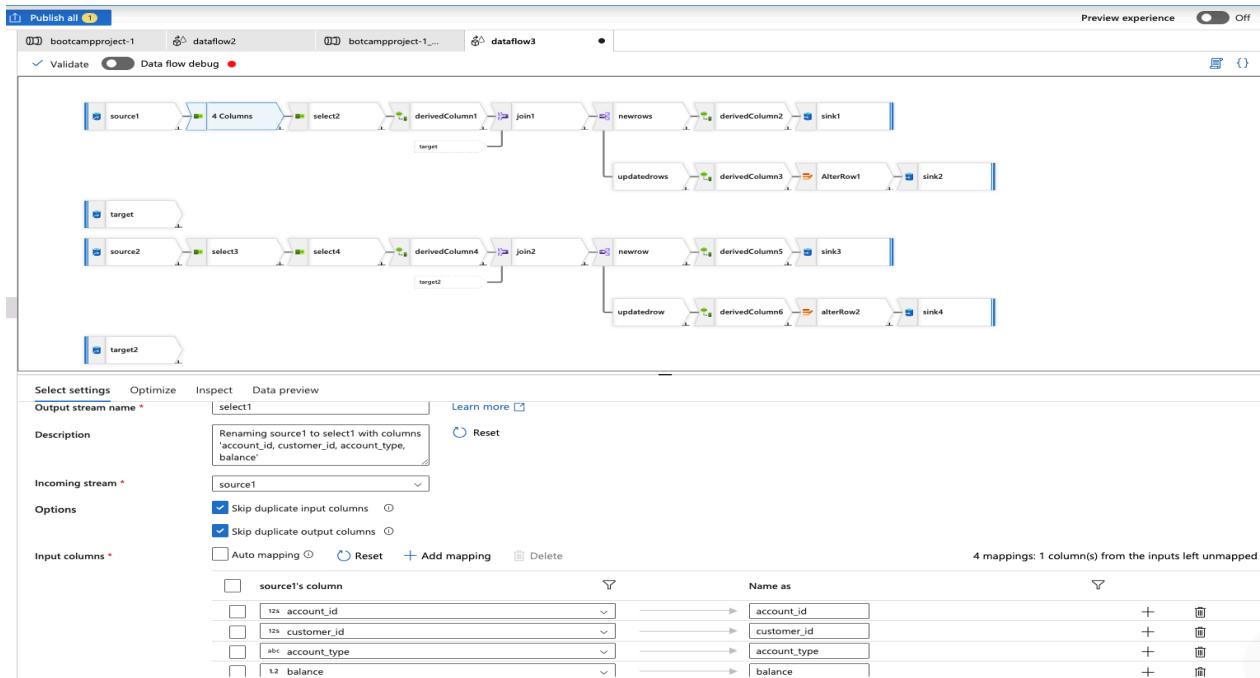
Here we are providing the path of file where it is stored.



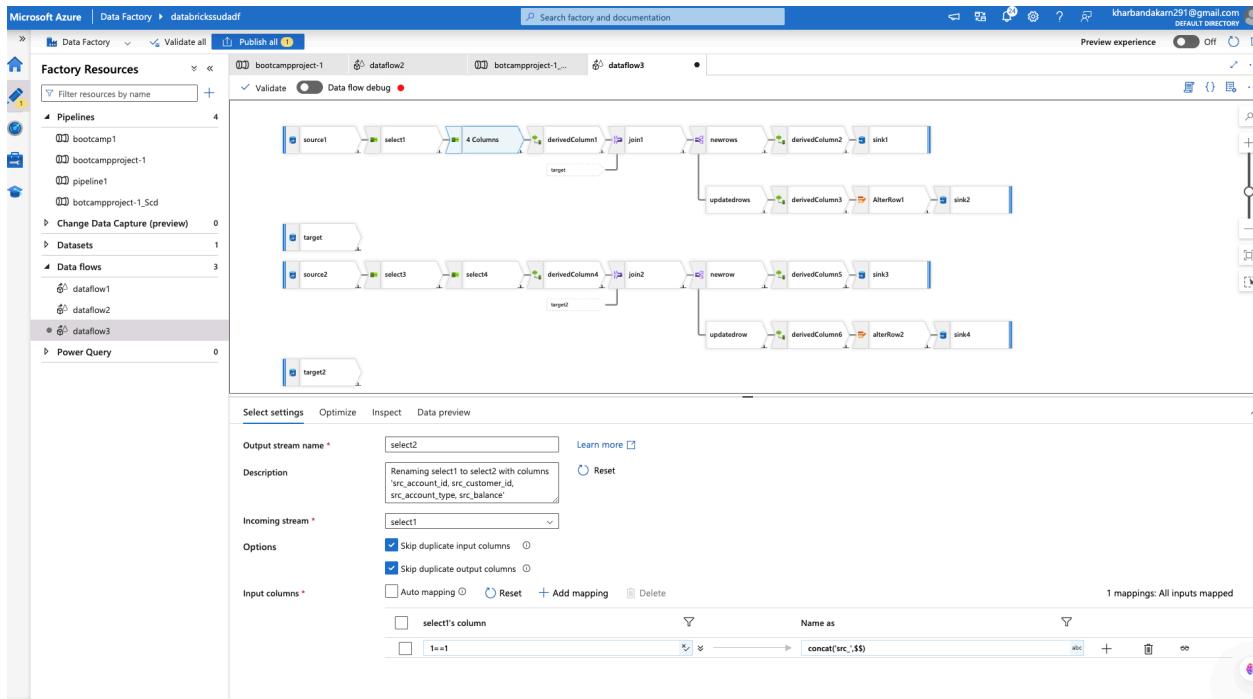
Here is how data will be coming in to SCD type dataflow from the silver container.



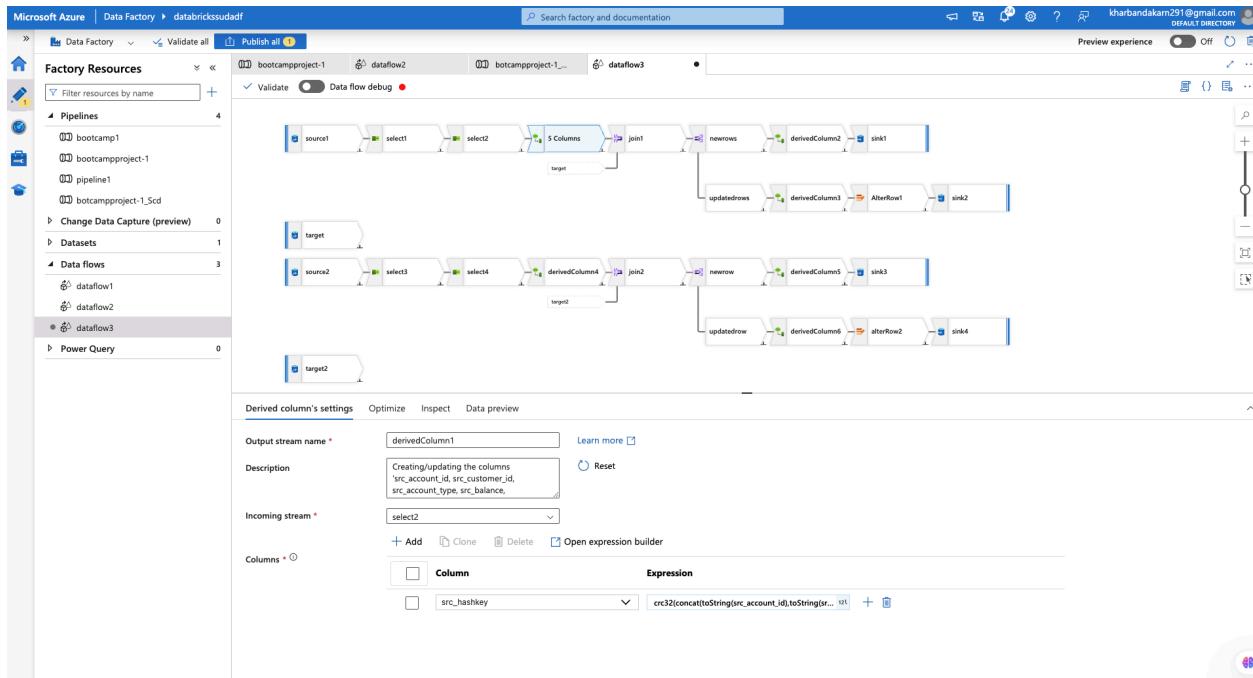
Now the select activity has been used for removing the unwanted column rownumber from the file.



Now we will add "src_" in all the columns of the table.



Here, using derived column activity, we are making a new column which will contain haskey. The haskey will made using all the column using crc32.



Here is the expression that we are using for making hash key.

Microsoft Azure | Data Factory > databrickssudaf

Dataflow expression builder

Column name: src_hashkey

Expression:

```
crc32(concat(toString(src_account_id),toString(src_customer_id),src_account_type,toString(src_balance)))
```

Expression elements

- All
- Functions
- Input schema
- Parameters
- Cached lookup
- Data flow library functions
- Locals

Expression values

- src_account_id
- src_customer_id
- src_account_type
- src_balance
- abs($=>$ numeric_value)
- acos($=>$ numeric_value)
- add($=>$ first_expression, $=>$ second_expression)
- addDays($=>$ date/timestamp, $=>$ days to add)
- addMonths($=>$ date/timestamp, $=>$ months to add)

Data preview

Save and finish Cancel Clear contents

Here we are inspecting how the data will look like.

Microsoft Azure | Data Factory > databrickssudaf

Factory Resources

Pipelines

- bootcamp1
- bootcampproject-1
- pipeline1
- bootcampproject-1_Scd

Change Data Capture (preview) 0

Datasets 1

Data flows 3

- dataflow1
- dataflow2
- dataflow3

Power Query 0

Validate Data flow debug

dataflow1

dataflow2

dataflow3

target

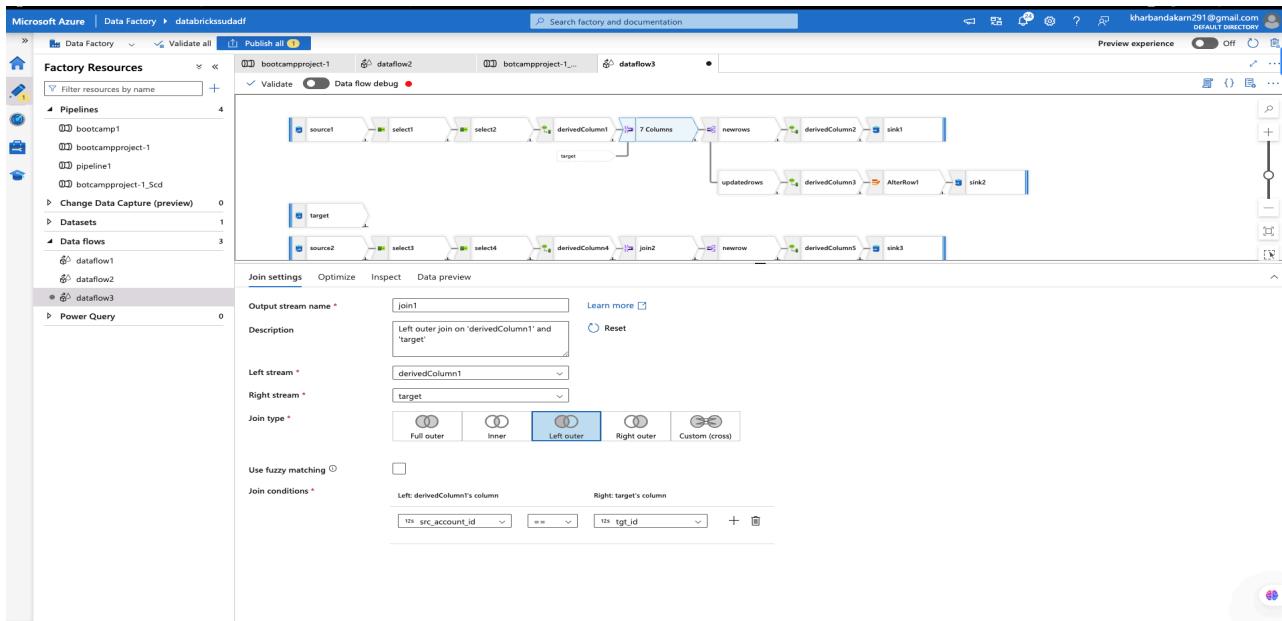
target2

Derived column's settings Optimize Inspect Data preview

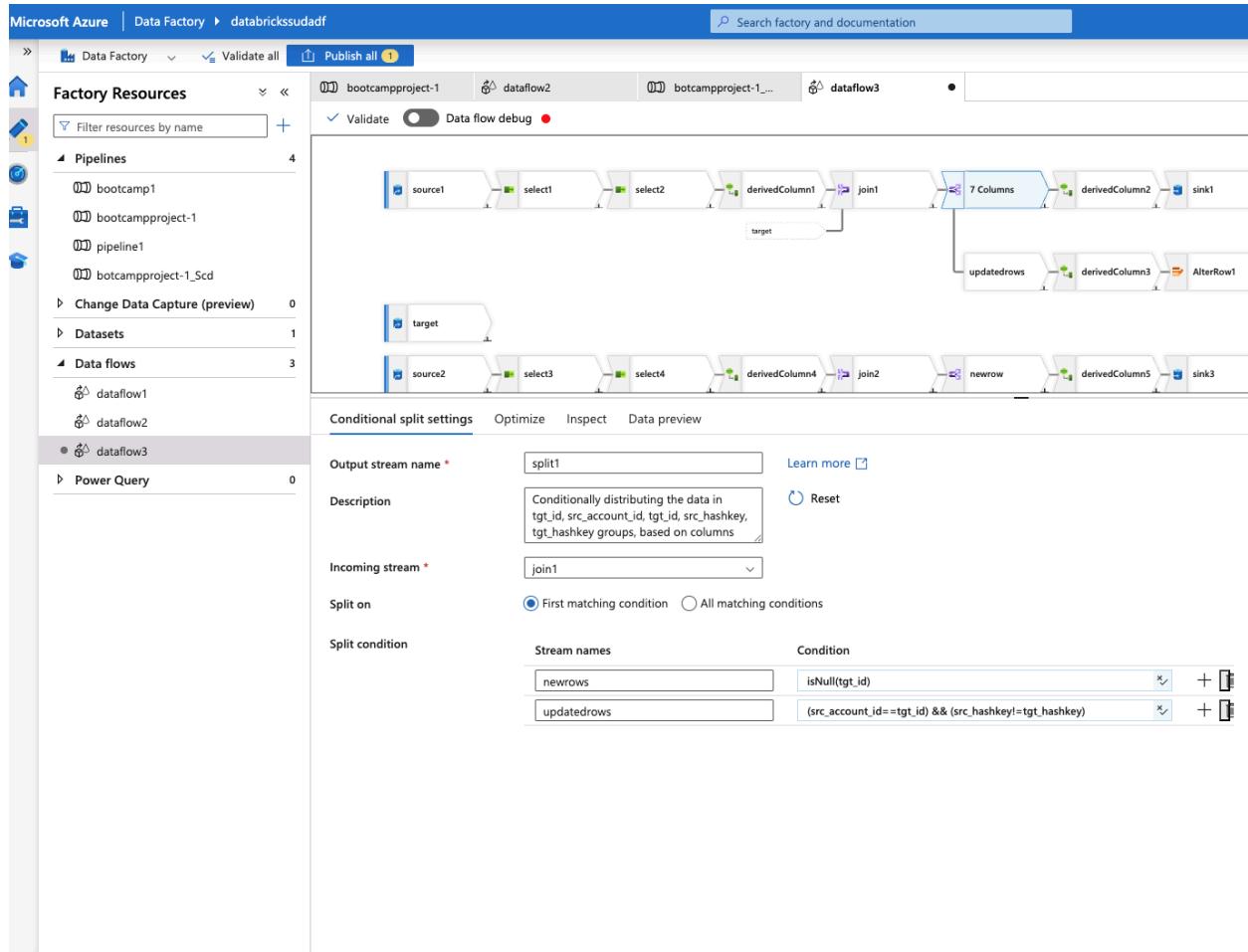
Schema Input Output

Number of columns New 1		Updated 0		Unchanged 4		Total 5
Order ↑↓	Column ↑↓	Type ↑↓	Updated ↑↓	Based on ↑↓		
1	src_account_id	121 short				
2	src_customer_id	121 short				
3	src_account_type	abc string				
4	src_balance	12 double				
5	src_hashkey	121 long	*	src_account_id,src_customer_id,src_account_type,src_balance		

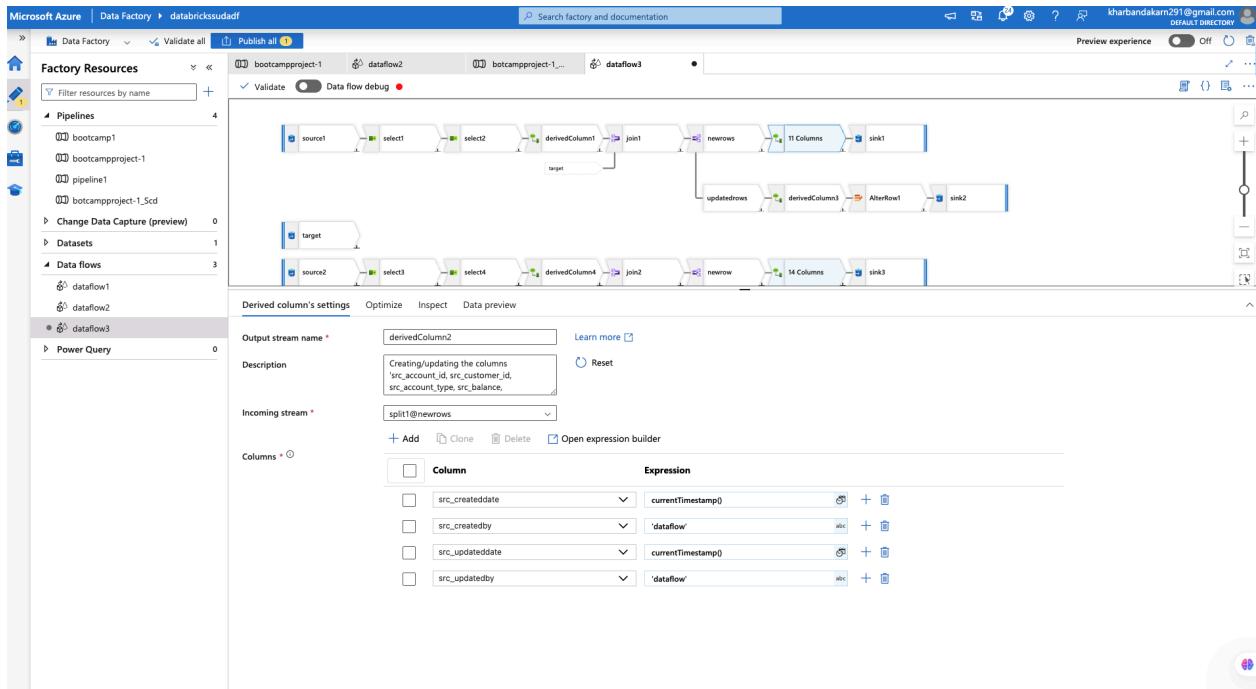
Now we are adding join activity where we will be joining two different tables. I.e the source and target(SCD type-1) table using left outer join.



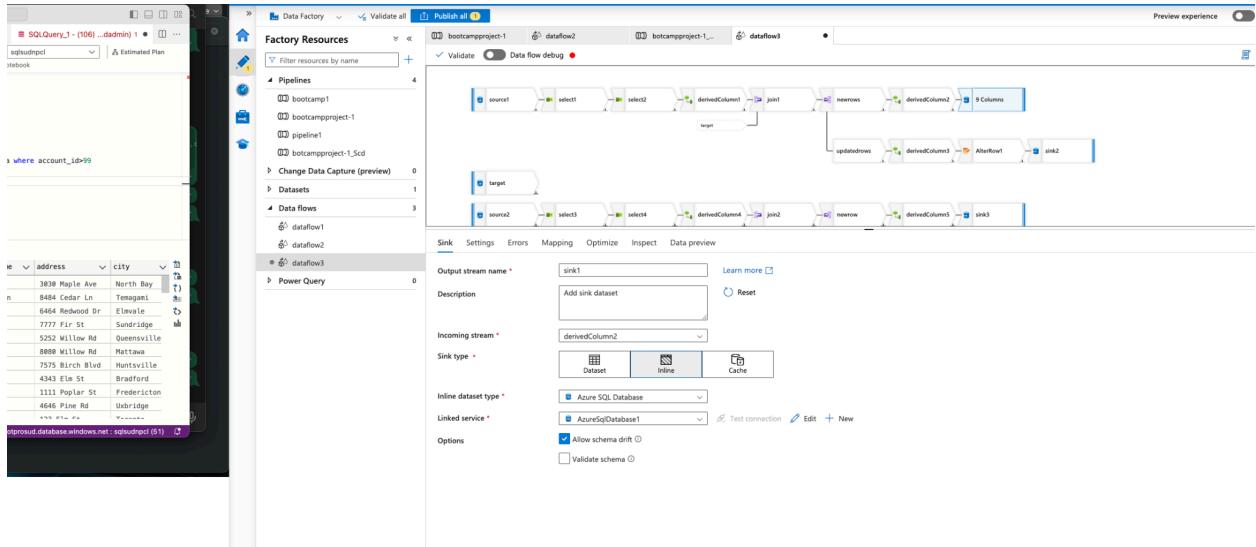
Here by using a conditional split we are splitting the data into 2 streams. One stream will contain the rows that are newly inserted, while the other stream will contain the rows that are updated.



Now using the derived column activity we are adding new column so we can map the data into the SCD type table.



Now we will be using a sink activity for saving the data in SQL database. Here we will do the following configuration for implementing sink.



In the setting tab select insert because here we will be inserting new data only, and select the schema name and table name.

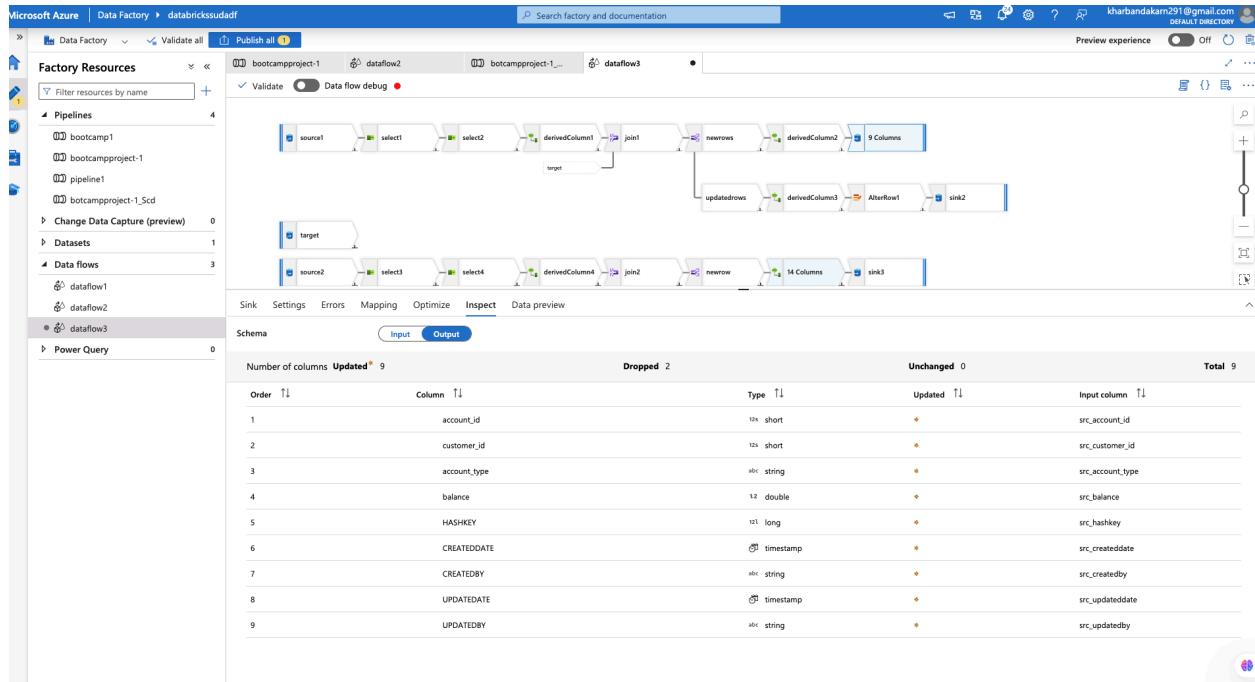
Screenshot of Microsoft Azure Data Factory Data Flow settings tab for dataflow3. The 'Sink' tab is selected. Schema name is set to 'BootPro1' and Table name is set to 'AccData'. Under 'Table action', 'Allow insert' is checked. Under 'Update method', 'Allow insert' is checked. Under 'Pre SQL scripts', 'List of scripts' is selected. Under 'Post SQL scripts', 'List of scripts' is selected.

Do the following mapping by aligning the source columns with the target scd type columns.

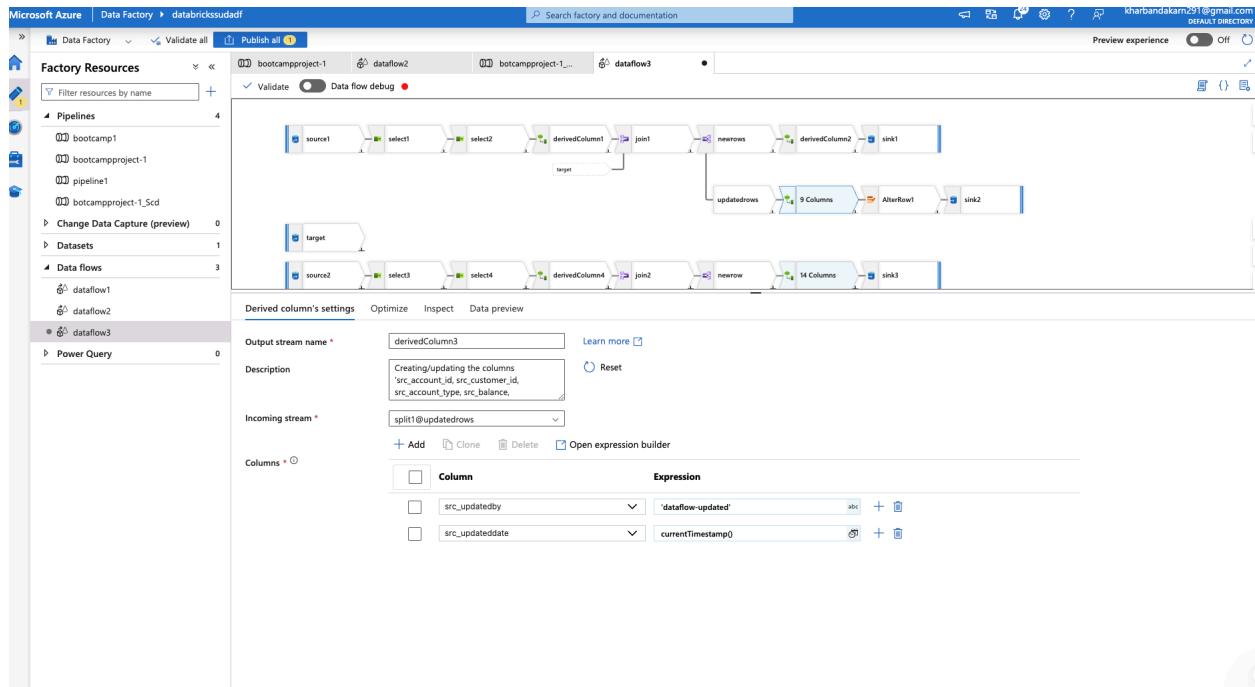
Input columns		Output columns	
src_account_id	account_id	src_customer_id	customer_id
src_createdate	CREATEDATE	src_createdby	CREATEDBY
src_updatedate	UPDATEDATE	src_updatedby	UPDATEDBY
src_hashkey	HASHKEY	src_account_type	account_type
src_balance	balance	src_updateddate	UPDATEDDATE

9 mappings: All outputs mapped

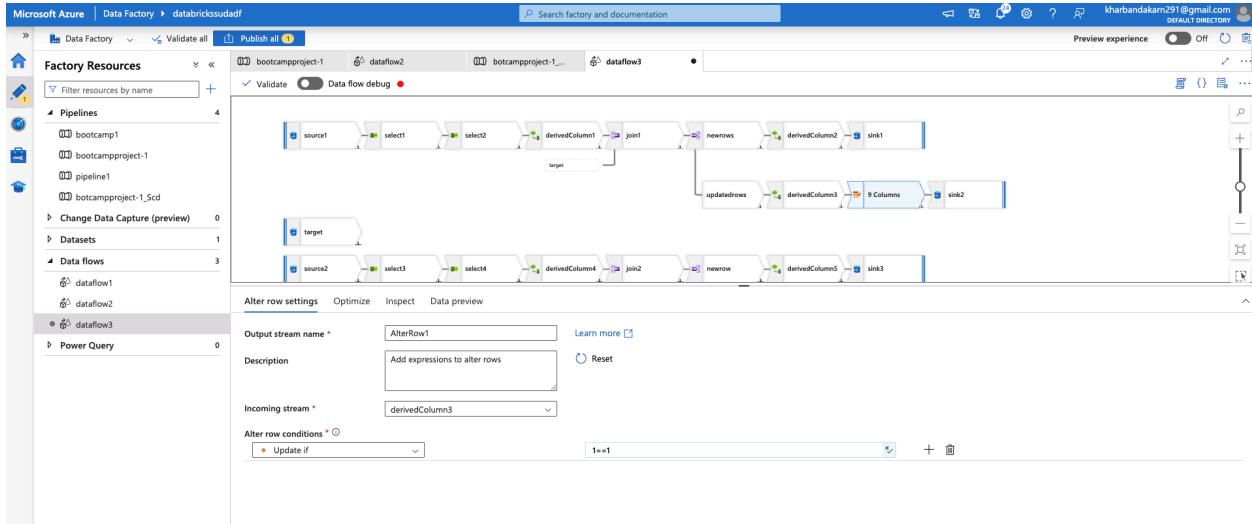
Here we are inspecting the data before it is saved into the SQL database(Gold Layer).



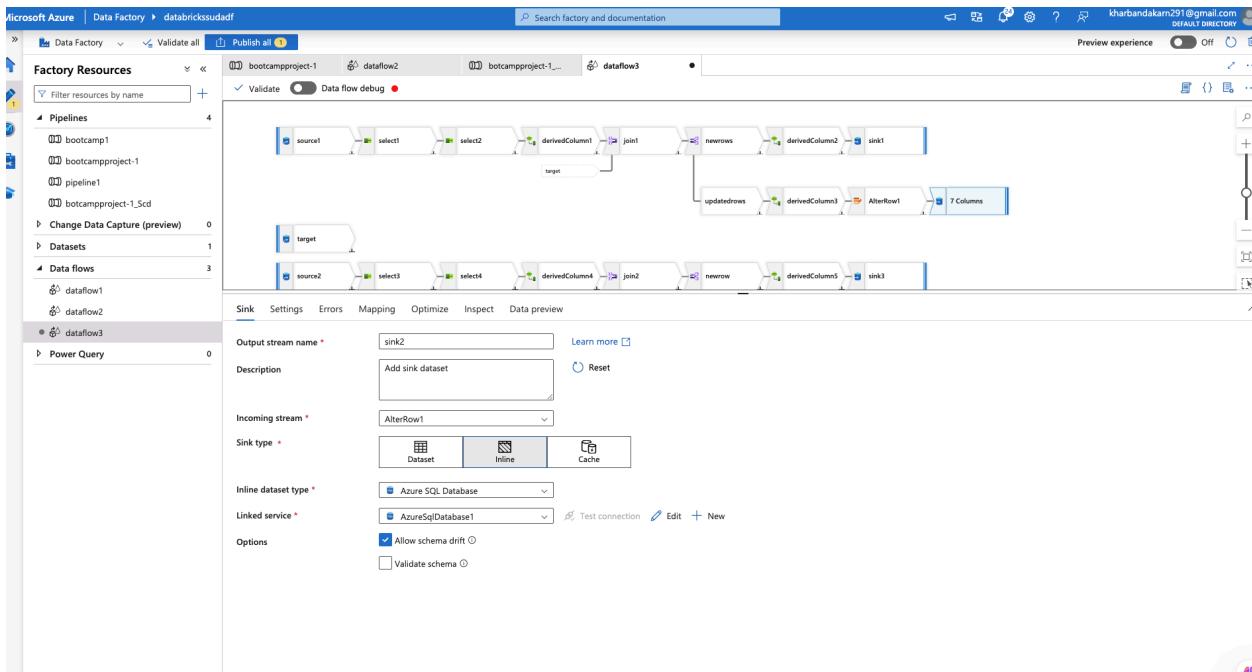
Now we add derived column activity after the updaterows stream. Here we will adding only updated by and updateddate column since we want add the recent updated time and person.



Here we will be alter row activity since update will be used in the sink activity.



Here is the sink activity that will be sinking the updated row in the table.



Add the schema name and the table and in the update method, select allow update.

Screenshot of Microsoft Azure Data Factory Data Flow blade showing the configuration for dataflow3. The 'Sink' tab is selected, showing:

- Schema name:** BootPro1
- Table name:** AccData
- Table action:** Recreate table (radio button selected)
- Update method:** Allow update (checkbox checked)
- Key columns:** account_id (List of columns selected)
- Use tempdb:** checked
- Pre SQL scripts:** List of scripts (radio button selected)

Do the following mapping for the updating the SCD type table. Note- Here we will remove createdby and createddate while mapping and update the old hashkey with the new hashkey.

Screenshot of Microsoft Azure Data Factory Data Flow blade showing the 'Mapping' tab for dataflow3. Under 'Options':

- ✓ Skip duplicate input columns
- ✓ Skip duplicate output columns

Input columns:

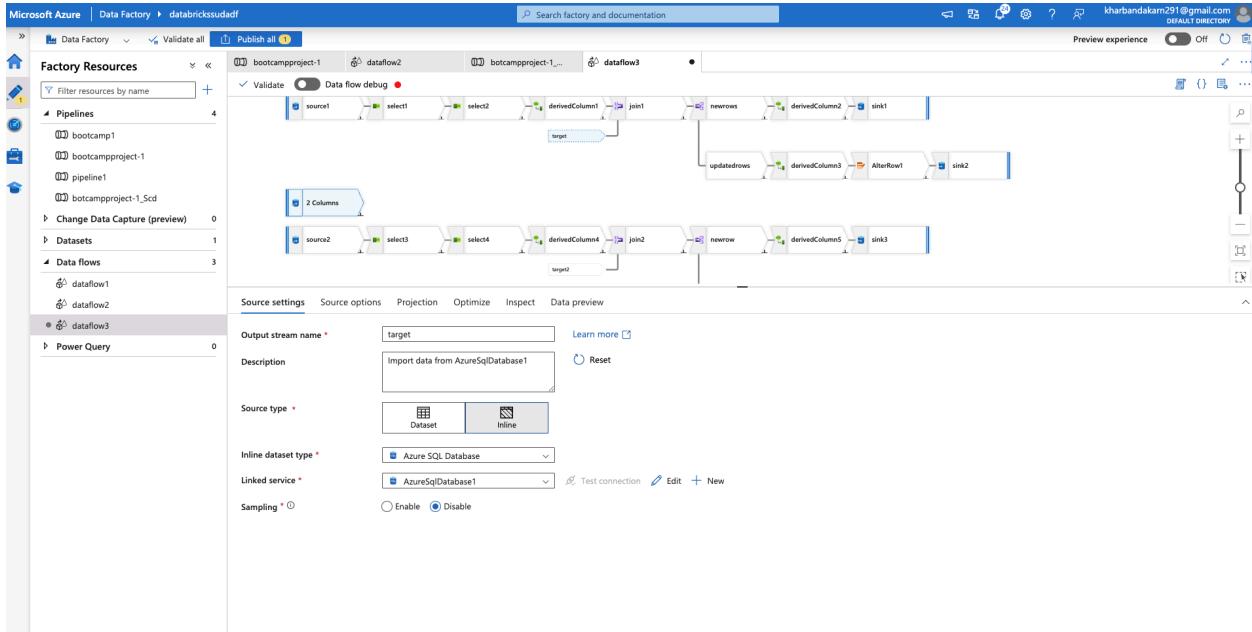
Input columns		Output columns	
src_customer_id	→	customer_id	+
src_account_type	→	account_type	+
src_balance	→	balance	+
src_hashkey	→	HASHKEY	+
tgt_id	→	account_id	+
src_updatedby	→	UPDATEDBY	+
src_updateddate	→	UPDATEDATE	+

Output columns:

Input columns		Output columns	
src_customer_id	→	customer_id	+
src_account_type	→	account_type	+
src_balance	→	balance	+
src_hashkey	→	HASHKEY	+
tgt_id	→	account_id	+
src_updatedby	→	UPDATEDBY	+
src_updateddate	→	UPDATEDATE	+

7 mappings: 2 column(s) from the output schema left unmapped

Here is the target activity for where the SCD type data will be loaded into the pipeline. Do the following configurations.



Creating a key vault for storing the password of the database.

Creating the link service for connecting the key vault.

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, and Power Query. In the center, a data flow named 'dataflow3' is being edited. The data flow consists of three main stages: 'select1', 'select2', and 'select3'. Stage 'select1' has an 'Incoming stream' of 'bootcamp1'. Stage 'select2' has a 'Sink' of 'derivedColumn1' and a 'target' of 'join1'. Stage 'select3' has an 'Incoming stream' of 'bootcamp1', a 'Sink' of 'derivedColumn2', and a 'target' of 'join2'. Stage 'select3' also includes a 'Mapping' tab where 'derivedColumn4' is mapped to 'join2'. On the right, a modal window titled 'New linked service' is open, specifically for 'Azure Key Vault'. The 'Name' field is set to 'AzureKeyVault1'. Under 'Azure key vault selection method', the radio button 'From Azure subscription' is selected. The 'Azure subscription' dropdown shows 'Azure subscription 1 (e7f63f14-30ce-4a77-9500-d226649402ad)'. The 'Azure key vault name' dropdown shows 'bootsecret'. The 'Authentication method' dropdown shows 'System-assigned managed identity'. Below these fields, it says 'Managed identity name: databrickssudafad' and 'Managed identity object ID: e61e9700-0bda-4b6d-be15-c7c9e464f9b1'. A note states 'Grant Data Factory service managed identity access to your Azure Key Vault.' with a 'Learn more' link. At the bottom of the modal, there are 'Create' and 'Cancel' buttons, and a status message 'Connection successful' with a green checkmark and a 'Test connection' button.

Now we have to select the linked service and secret name for the key vault.

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar shows the same resource list as before. In the center, the 'dataflow3' data flow is being edited. The 'Sink' tab is selected, showing 'derivedColumn2' as the sink type. The 'Linked service' dropdown is set to 'AzureSqlDatabase1'. On the right, a modal window titled 'Edit linked service' is open for 'Azure SQL Database'. It shows the 'Fully qualified domain name' as 'bootprosud.database.windows.net', 'Database name' as 'sqlsudnpl', 'Authentication type' as 'SQL authentication', 'User name' as 'sqlsudadmin', and 'Password' as 'sqlpassword'. Under 'AKV linked service', the dropdown is set to 'AzureKeyVault2'. The 'Secret name' is 'sqlpassword' with the 'Edit' checkbox checked. The 'Secret version' is 'd4b3db537c3d40ecae047c20add28197' with the 'Edit' checkbox checked. There are also 'Always encrypted' and 'Encrypt' options at the bottom. At the very bottom of the modal, there are 'Save' and 'Cancel' buttons, and a 'Test connection' button with a blue checkmark icon.

Use the following query to get the data into the table. Here, we will be fetching only the primary key and the hashkey column.

The screenshot shows the Microsoft Azure Data Factory Data Flow interface. On the left, the 'Factory Resources' sidebar lists Pipelines (bootcamp1, bootcampproject-1, pipeline1, bootcampproject-1_Scd), Datasets (1), and Data flows (3). The current view is on 'dataflow3'. The main area displays a complex data flow diagram with three parallel source paths (source1, source2, source3) followed by various transformations (select1-4, derivedColumn1-5, join1-2, newrow, updatedrows, AlterRow) and sinks (sink1, sink2, sink3). The 'Source options' tab is selected, showing the input query:

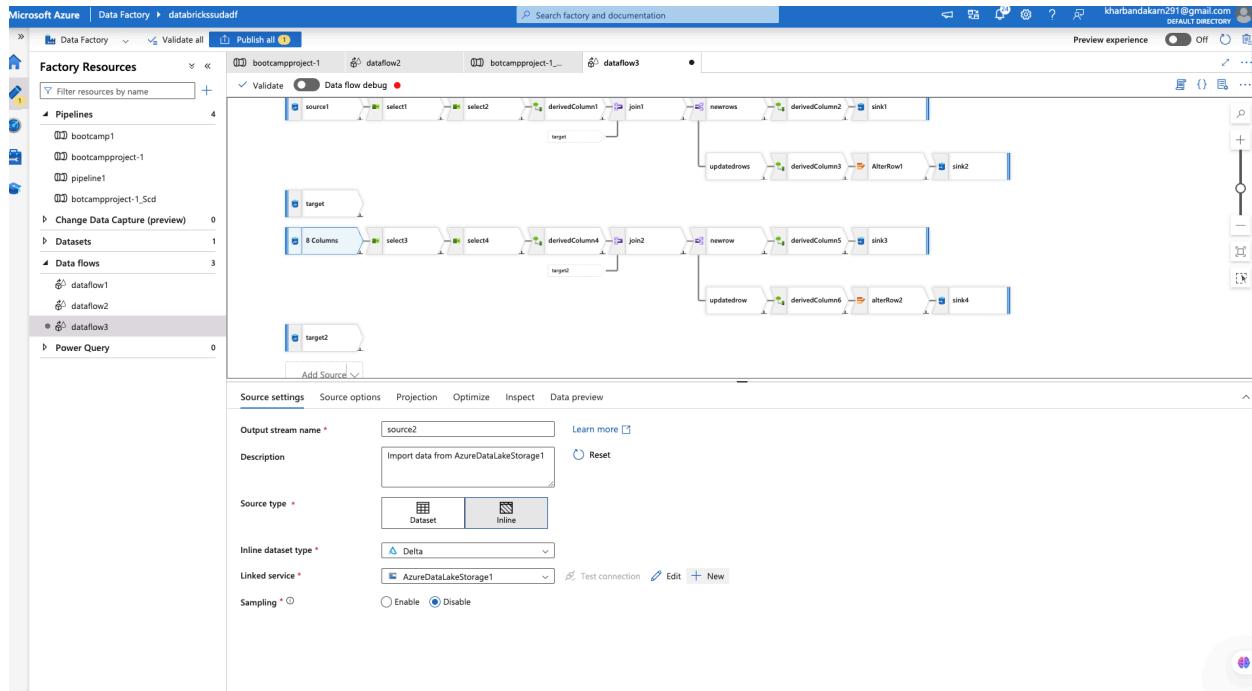
```
Select account_id as tgt_id, HASHKEY as tgt_hashkey from BootPro1AccData
```

The 'Inspect' tab is also visible at the bottom.

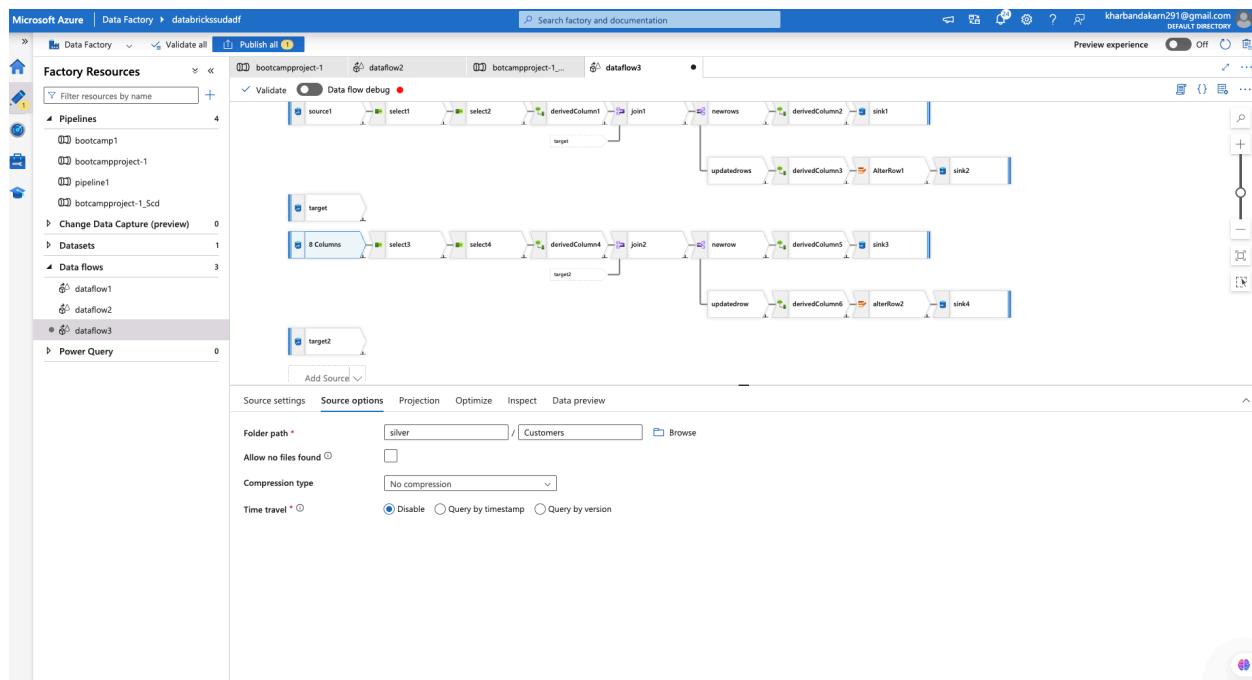
The screenshot shows the Microsoft Azure Data Factory Data Flow interface with the 'Inspect' tab selected. The schema is defined as follows:

Order	Column	Type
1	tgt_id	short
2	tgt_hashkey	long

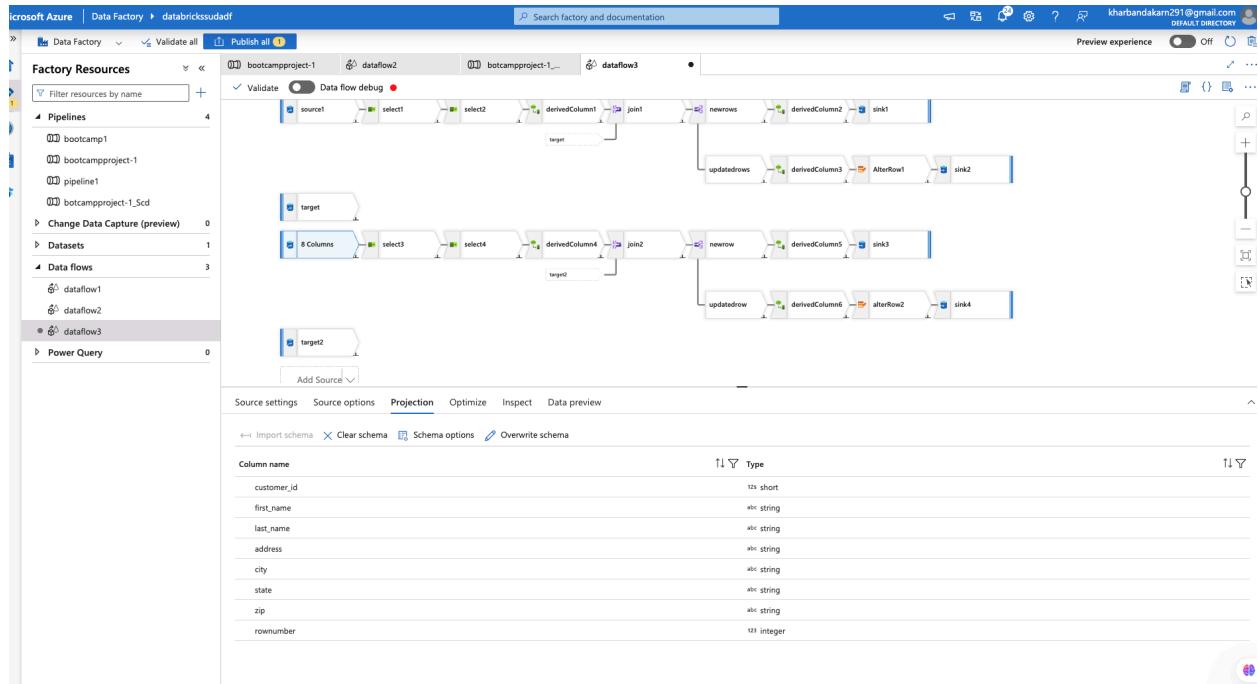
Now the same will be implementing scd type-1 for another csv file named customers. Do the following configuration for loading file into the pipeline from silver layer.



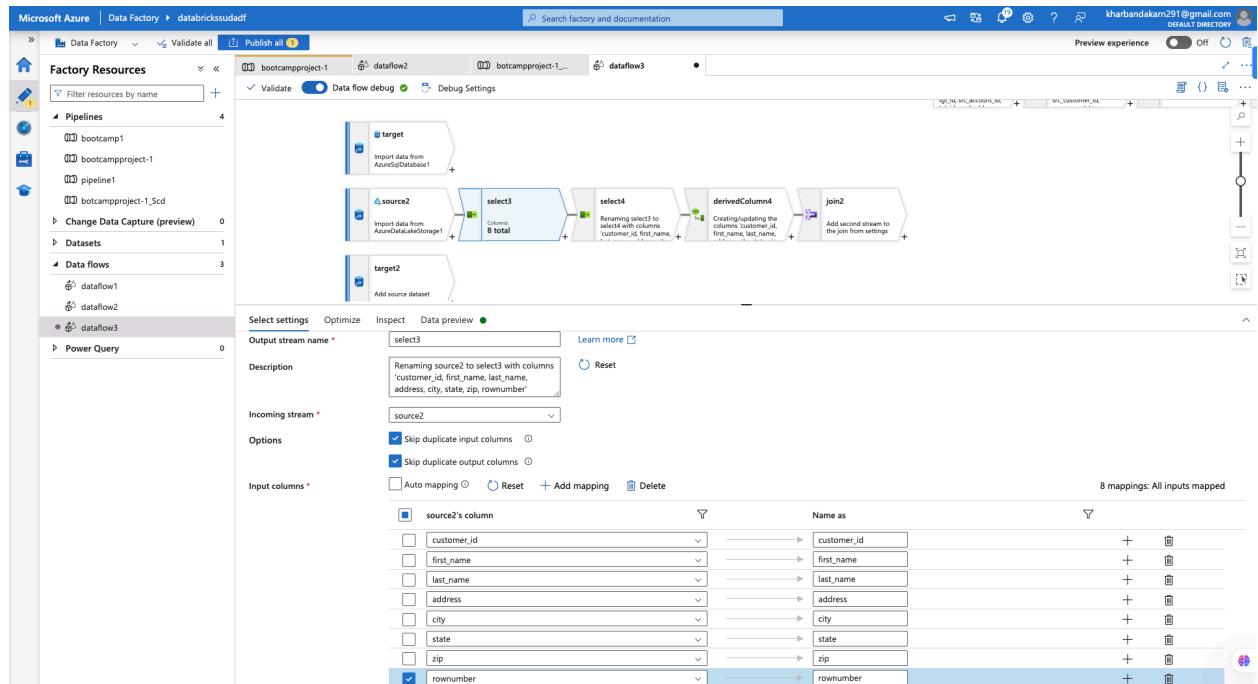
Here provide the path from where data has been loaded into the pipeline.



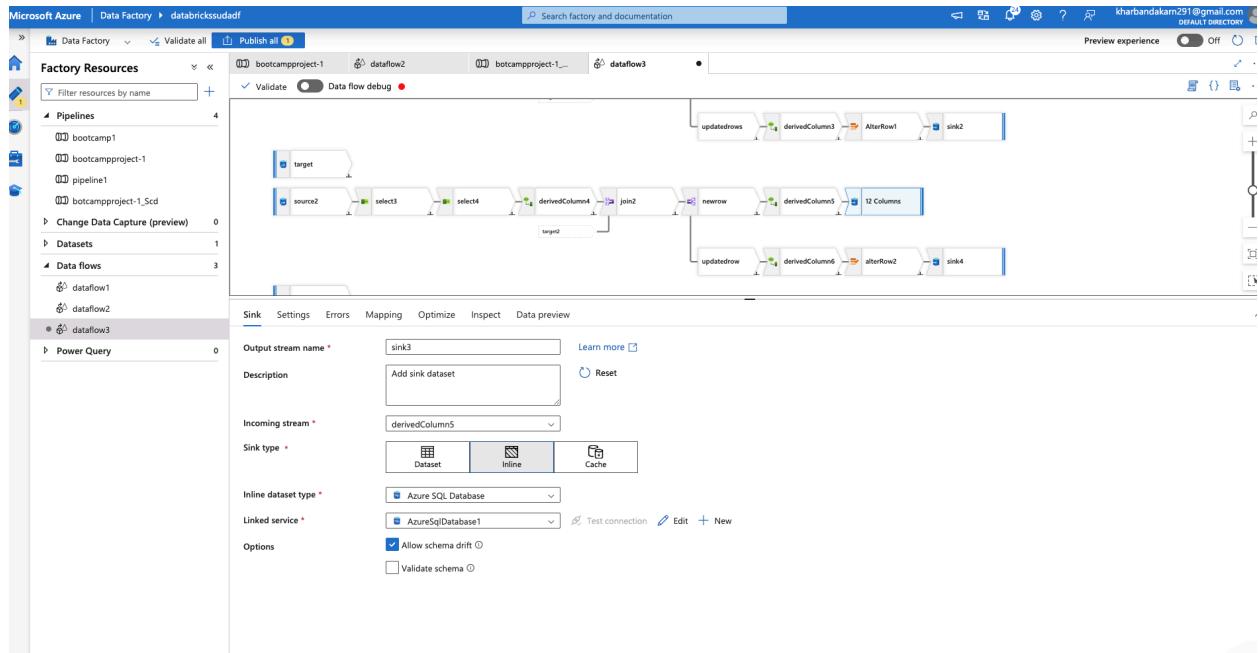
Now we will import the schema to check the columns.



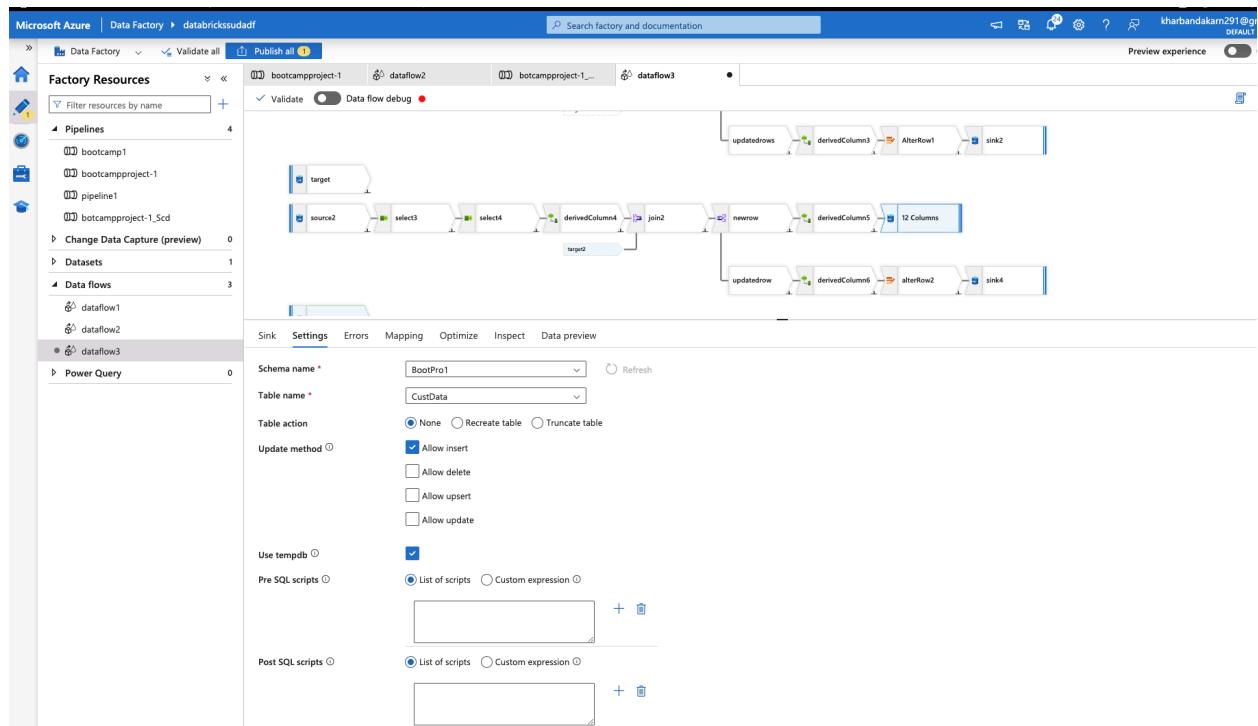
We are adding select activity to remove the unwanted column rownumber



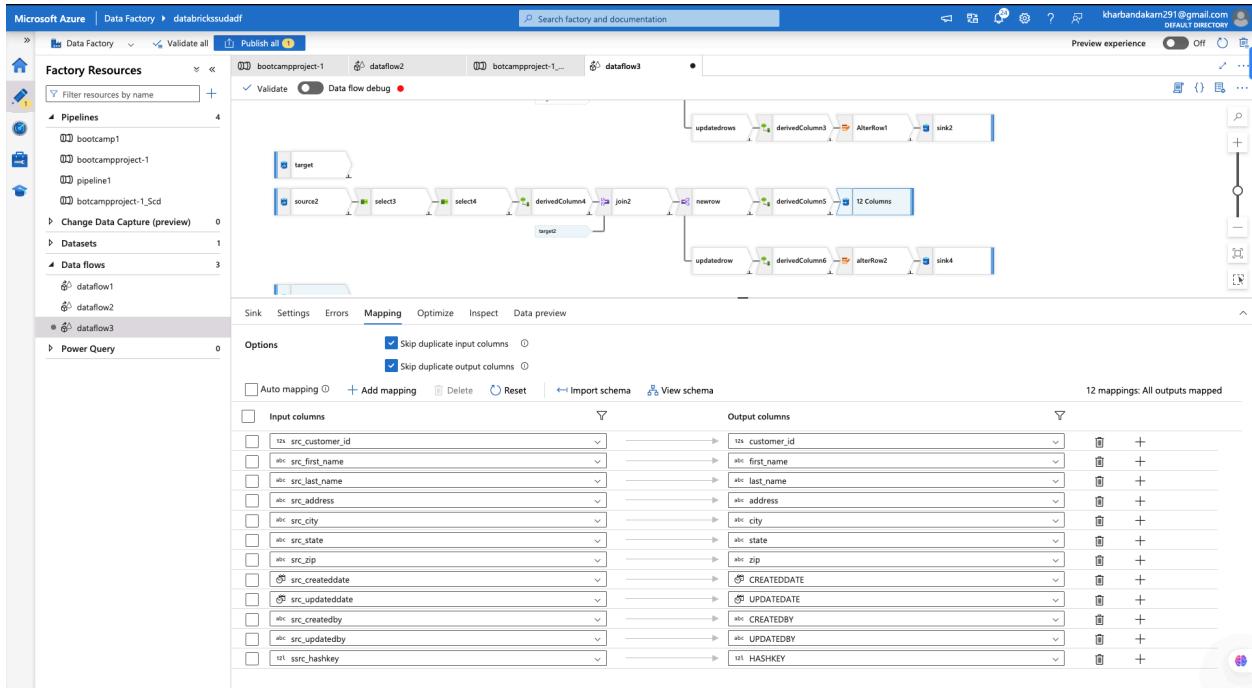
We will be implementing scd type-1 in the same way as we have implemented for the above CSV file there will be slight change in the mapping part of the sink.



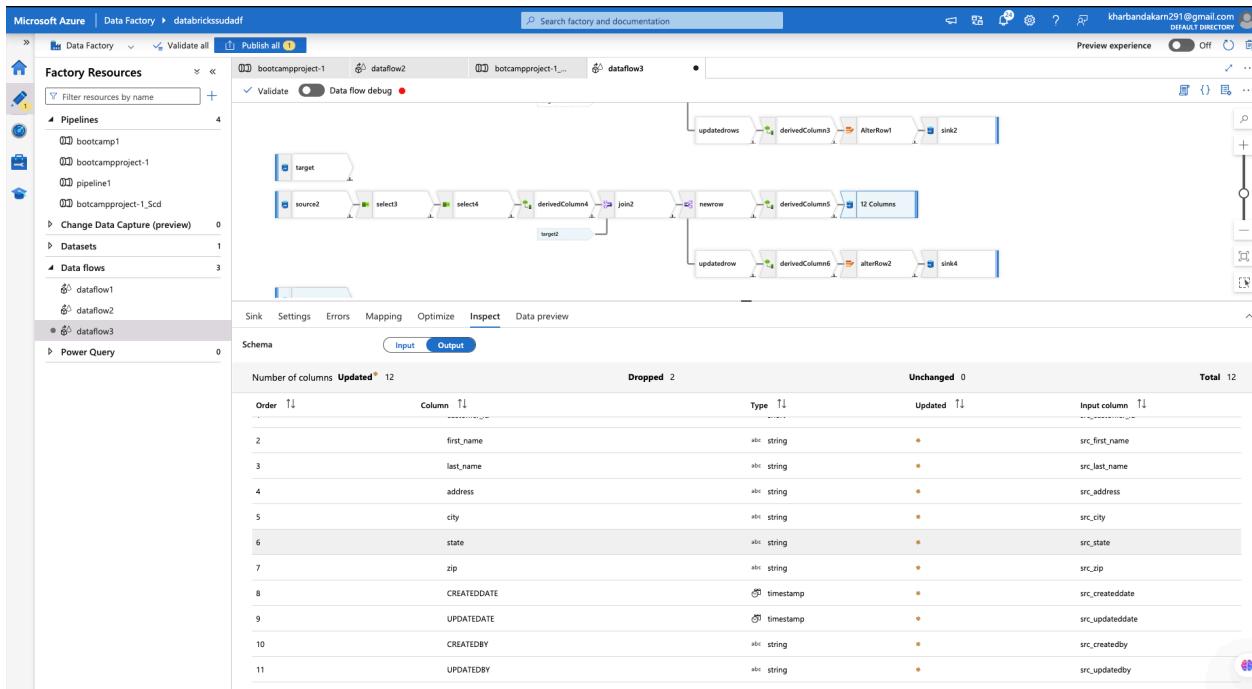
Here will check insert as it update method since this activity will sink new added rows in the SQL database.



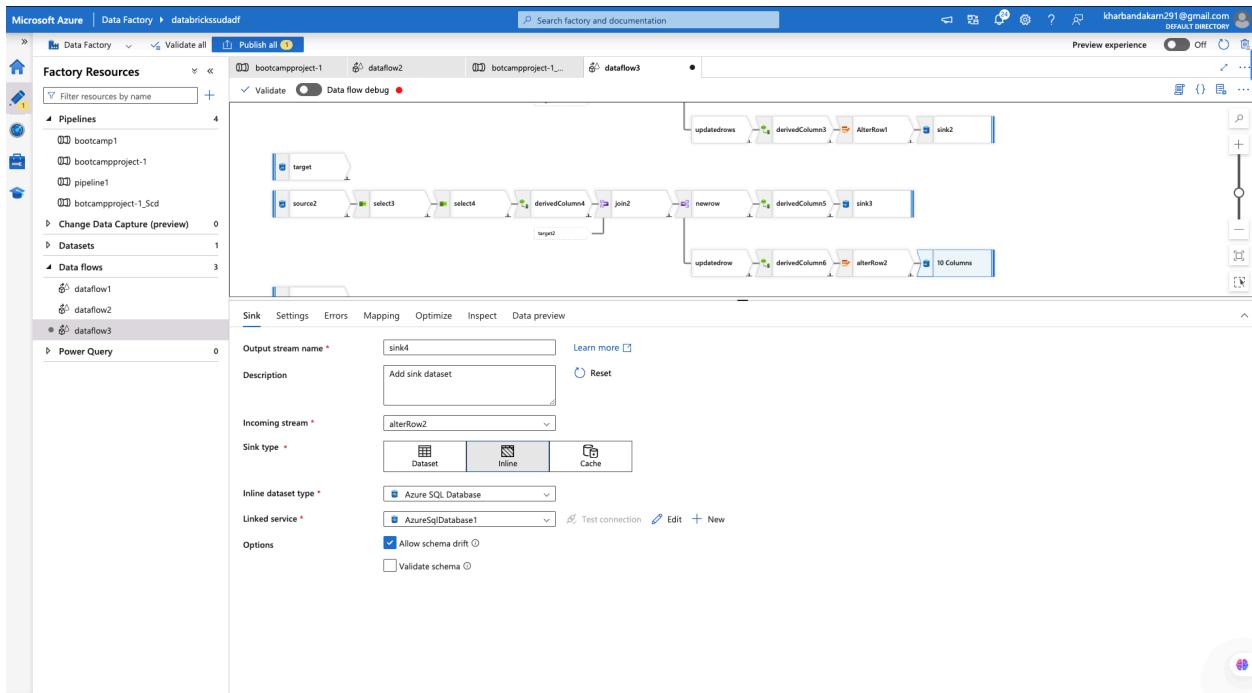
Here is the mapping that we have to new rows stream.



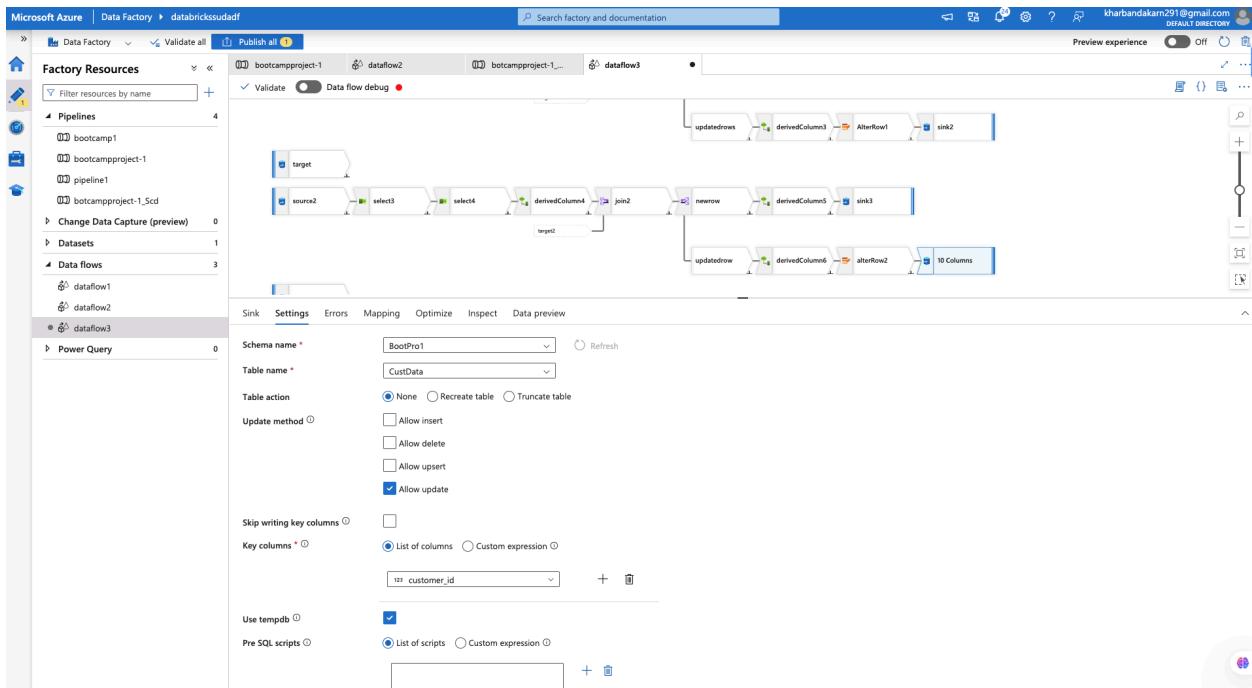
Here is the data review.



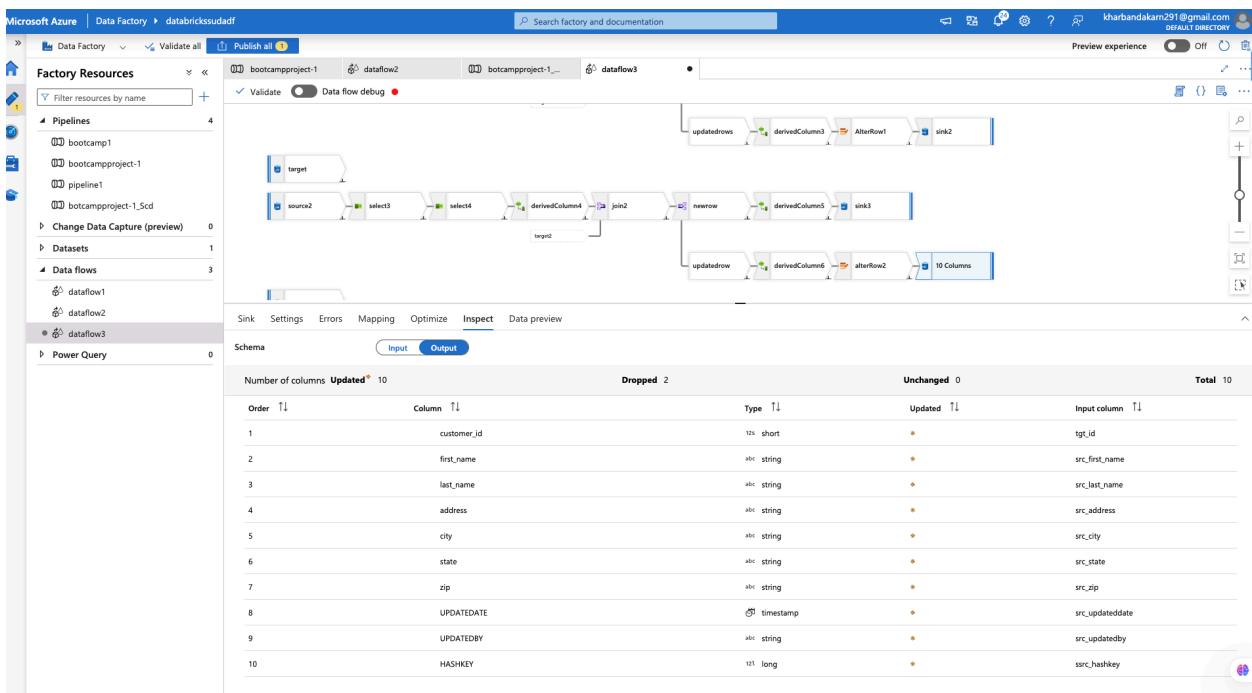
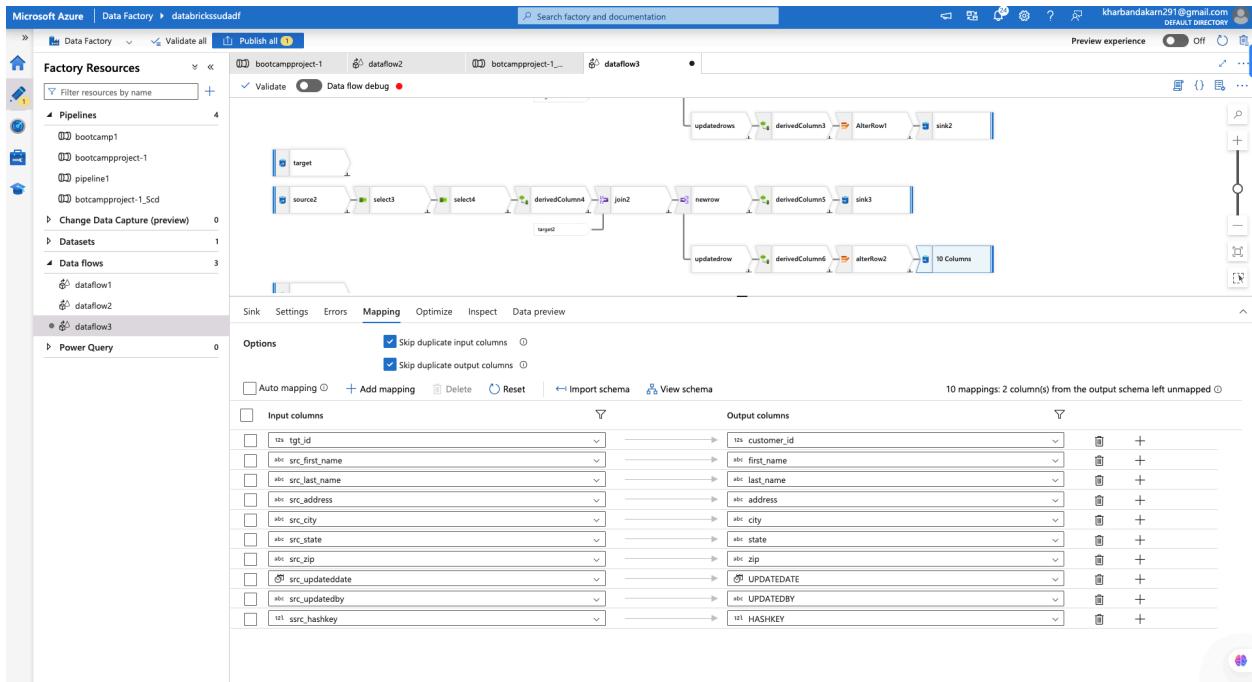
Now we will be adding sink activity where we will be adding new updated row in the sql database.



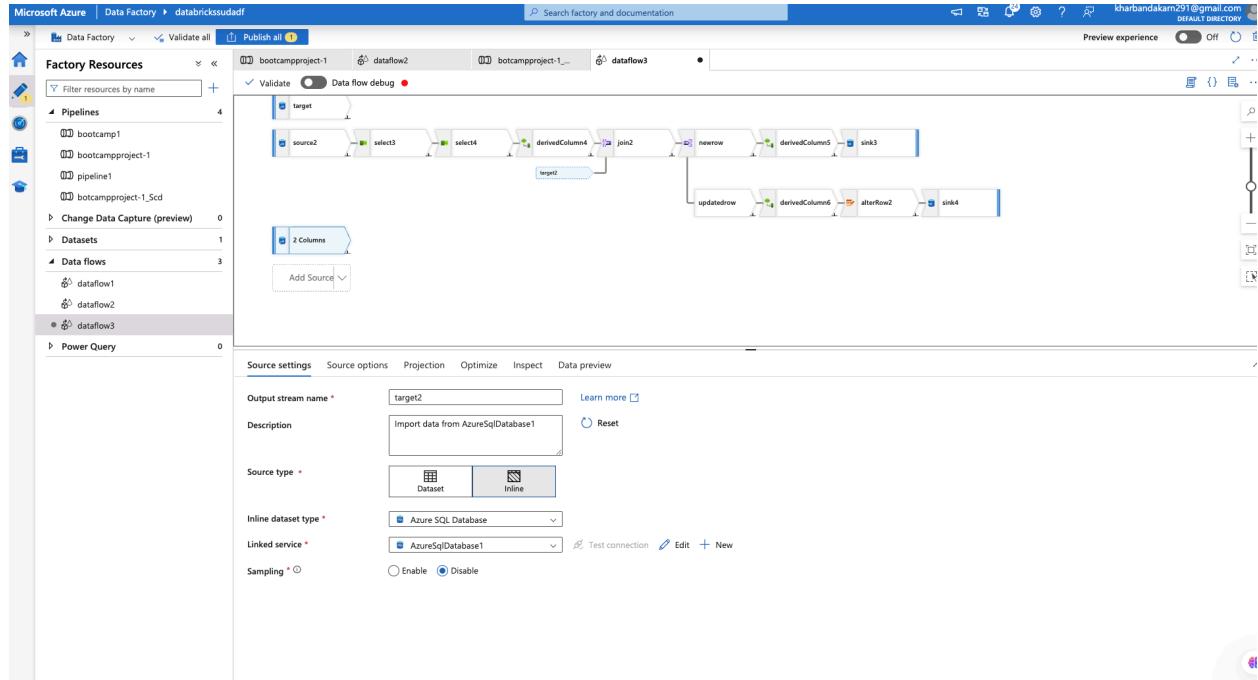
Here we will check update method since we adding updated rows in the database.



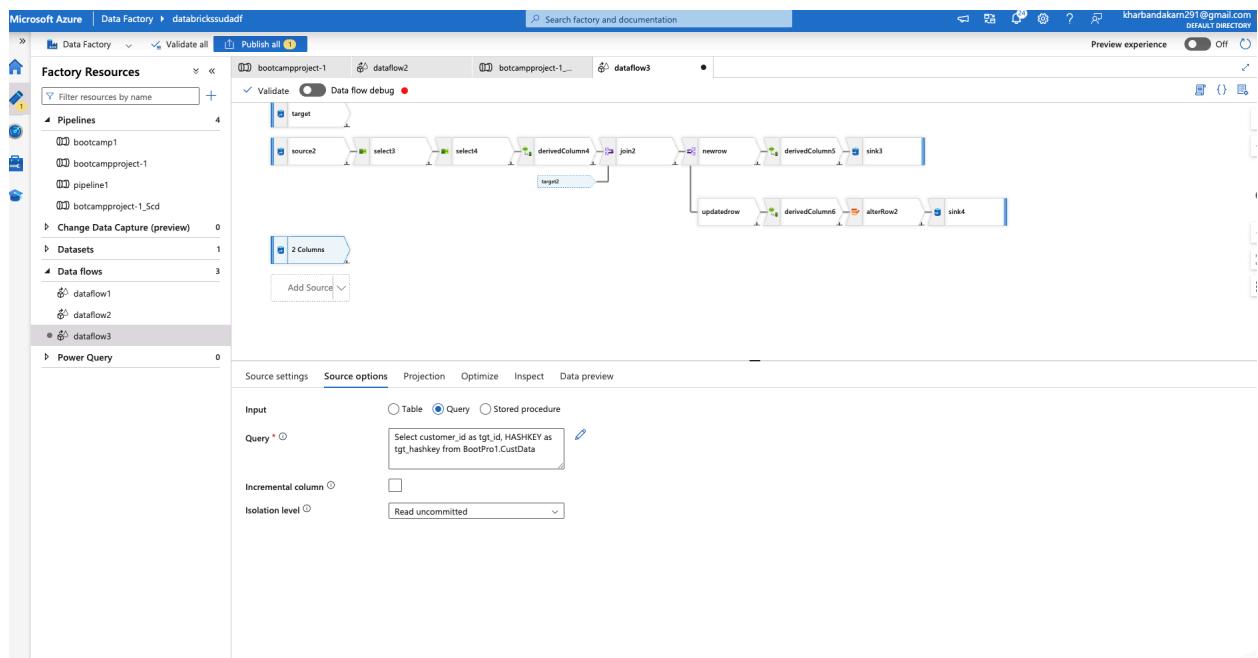
Here is the mapping that we have to done for update row stream.



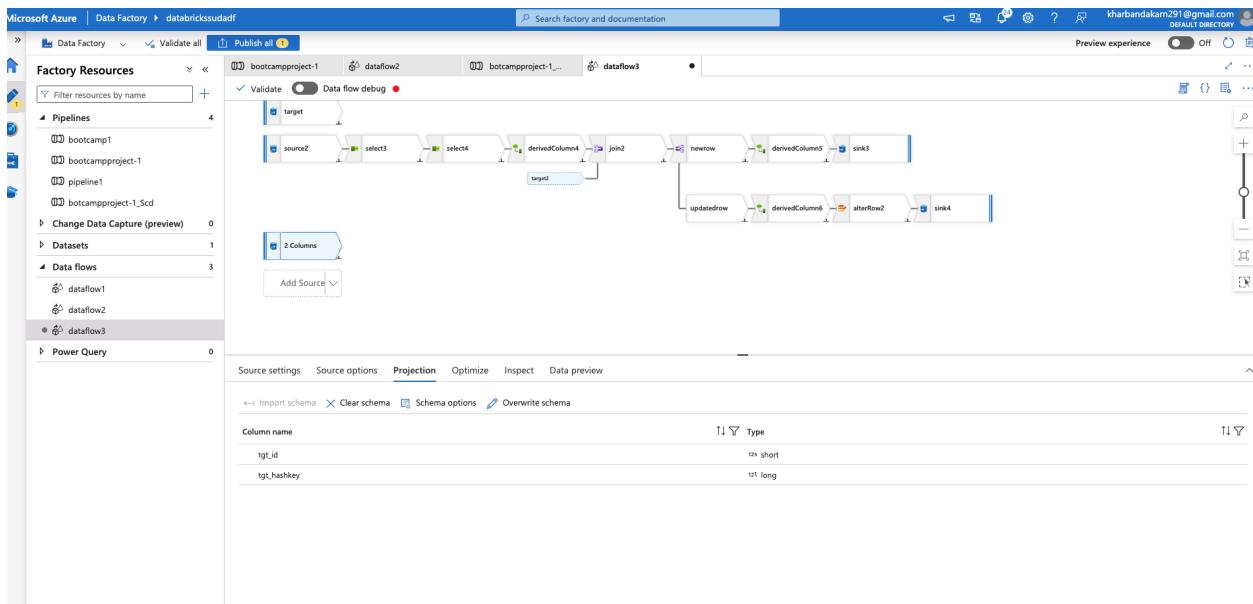
Here is target activity that will loading data from SQL database.



We will using query to load only the primary key and hash key data.



Here is a description of the columns loaded.



Here is SQL script that we have used for creating scd type -1 table in SQL database.

```

SQLQuery_2.sql - disconnected 3 bootprosud.database.windows.net SQLQuery_1 - (106) ...dadmin) 1
Run Cancel & Disconnect & Change Database: sqlsudnpcl Estimated Plan & Enable Actual Plan & Parse & Enable SQLCMD & To Notebook
1
2 CREATE SCHEMA BootPro1;
3
4 CREATE TABLE BootPro1.AccData(
5     account_id SMALLINT,
6     customer_id SMALLINT,
7     account_type VARCHAR(50),
8     balance DECIMAL(18, 2),
9     CREATEDATE DATETIME,
10    UPDATEDATE DATETIME,
11    CREATEDBY VARCHAR (100),
12    UPDATEDBY VARCHAR (100),
13    HASHKEY BIGINT
14 );
15
16 CREATE TABLE BootPro1.CustData(
17     customer_id SMALLINT,
18     first_name VARCHAR(50),
19     last_name VARCHAR(50),
20     address VARCHAR(100),
21     city VARCHAR(50),
22     state VARCHAR(50),
23     zip VARCHAR(10),
24     CREATEDATE DATETIME,
25     UPDATEDATE DATETIME,
26     CREATEDBY VARCHAR (100),
27     UPDATEDBY VARCHAR (100),
28     HASHKEY BIGINT
29 );
30
31 SELECT * FROM BootPro1.AccData;
32 SELECT account_id FROM BootPro1.AccData where account_id>99
33 SELECT * FROM BootPro1.CustData;
34

```

So after running the pipeline for the first time, when all the data has been loaded in the scd type table. We are making changes in the bronze layer where we are updating rows and inserting rows.

The screenshot shows a cloud storage interface with a blue header bar containing a search bar ('Search resources, services, and docs (G+/)') and a 'Copilot' button. Below the header, the file name 'customers/customers.csv' is displayed in bold black text. A 'Blob' label is visible above the file preview area. Below the file name, there are several action buttons: 'Save', 'Discard', 'Download', 'Refresh', and 'Delete'. A horizontal line separates this from the file content. Underneath the file name, there are tabs for 'Overview', 'Versions', 'Edit' (which is underlined), and 'Generate SAS'. The main content area displays a list of 19 customer records, each numbered from 1 to 19. The columns are 'customer_id', 'first_name', 'last_name', 'address', 'city', 'state', and 'zip'. The data includes various names and addresses across Canada, such as John Doe at 123 Elm St, Toronto, ON, M4B1B3, and Olivia Adams at 1717 Oak Dr, Saskatoon, SK, S7K0A1.

```

1 customer_id,first_name,last_name,address,city,state,zip
2 1,John,Doe,123 Elm St,Toronto,ON,M4B1B3
3 2,Jane,Smith,456 Maple Ave,Ottawa,ON,K1A0B1
4 3,Michael,Johnson,721 Oak Dr,Montreal,QC,H1A1A1
5 4,Emily,Davis,101 Pine Rd,Calgary,AB,T2A0A1
6 5,David,Wilson,202 Birch Blvd,Vancouver,BC,V5K0A1
7 6,Emma,Clark,505 Cedar St,Halifax,NS,B3H0A1
8 7,James,Martinez,606 Spruce Ln,Winnipeg,MB,R3C0A1
9 8,Olivia,Garcia,707 Fir St,Edmonton,AB,T5A0A1
10 9,William,Lopez,808 Redwood Dr,Victoria,BC,V8W0A1
11 10,Ava,Anderson,909 Cypress Ave,Quebec City,QC,G1A0A1
12 11,Alexander,Thomas,1010 Willow Rd,St. John's,NL,A1A0A1
13 12,Isabella,Lee,1111 Poplar St,Fredericton,NB,E3B0A1
14 13,Daniel,Harris,1212 Ash Blvd,Charlottetown,PE,C1A0A1
15 14,Sophia,Young,1313 Beech Dr,Yellowknife,NT,X1A0A1
16 15,Matthew,King,1414 Cedar Ln,Whitehorse,YT,Y1A0A1
17 16,Charlotte,Scott,1515 Elm St,Iqaluit,NU,X0A0A1
18 17,Joseph,Green,1616 Maple Ave,Regina,SK,S4P0A1
19 18,Amelia,Adams,1717 Oak Dr,Saskatoon,SK,S7K0A1

```

Here we are inserting a new row.

The screenshot shows a cloud storage interface with a blue header bar containing a search bar ('Search resources, services, and docs (G+/)') and a 'Copilot' button. Below the header, the file name 'customers/customers.csv' is displayed in bold black text. A 'Blob' label is visible above the file preview area. Below the file name, there are several action buttons: 'Save', 'Discard', 'Download', 'Refresh', and 'Delete'. A horizontal line separates this from the file content. Underneath the file name, there are tabs for 'Overview', 'Versions', 'Edit' (which is underlined), and 'Generate SAS'. The main content area displays a list of 89 customer records, each numbered from 1 to 89. The columns are 'customer_id', 'first_name', 'last_name', 'address', 'city', 'state', and 'zip'. The data includes various names and addresses across Canada, such as Christopher Myers at 7070 Cedar Ln, Coldwater, ON, L0K0A1, and Olivia Gibbs at 8586 Elm St, New Lisk, ON, P0J2A1. The last record shown is number 89.

```

72 71,Christopher,Myers,7070 Cedar Ln,Coldwater,ON,L0K0A1
73 72,Mia,Ford,7171 Elm St,Orillia,ON,L3V0A1
74 73,Andrew,Hamilton,7272 Maple Ave,Gravenhurst,ON,P1P0A1
75 74,Harper,Graham,7373 Oak Dr,Bala,ON,P0C0A1
76 75,Joshua,Sullivan,7474 Pine Rd,Bracebridge,ON,P1L0A1
77 76,Evelyn,Wallace,7575 Birch Blvd,Huntsville,ON,P1H0A1
78 77,Daniel,Woods,7676 Spruce Ln,Burks Falls,ON,P0A0A1
79 78,Abigail,Cole,7777 Fir St,Sundridge,ON,P0A0A1
80 79,James,West,7878 Redwood Dr,South River,ON,P0A0A1
81 80,Emily,Jordan,7979 Cypress Ave,North Bay,ON,P1B0A1
82 81,Michael,Owens,8080 Willow Rd,Mattawa,ON,P0H0A1
83 82,Elizabeth,Reynolds,8181 Poplar St,Sturgeon Falls,ON,P2B0A1
84 83,David,Fisher,8282 Ash Blvd,Verner,ON,P0H0A1
85 84,Sophia,Ellis,8383 Beech Dr,Field,ON,P0H0A1
86 85,John,Harrison,8484 Cedar Ln,Temagami,ON,P0H0A1
87 86,Olivia,Gibson,8585 Elm St,New Liskeard,ON,P0J0A1
88 87,William,McDonald,8686 Maple Ave,Haileybury
89 88,Oliv,Gibbs,8586 Elm St>New Lisk,ON,P0J2A1

```

At the bottom of the screen, there are two buttons: 'Csv' and 'Preview'.

Here are the updated rows in the second CSV file.

The screenshot shows a cloud storage interface for a CSV file named "Accounts/accounts.csv". The file is categorized as a "Blob". The interface includes standard actions like Save, Discard, Download, Refresh, and Delete. Below these are tabs for Overview, Versions, Edit (which is selected), and Generate SAS. The main content area displays 41 rows of data, each consisting of a row number and a comma-separated value (CSV) string. The data represents account numbers, names, and balances. At the bottom, there are buttons for Csv and Preview.

Row	Value
57	56,28,Checking,5700.00
58	57,97,Savings,450.25
59	58,16,Checking,5900.50
60	59,75,Savings,475.75
61	60,20,Checking,6100.00
62	61,52,Savings,500.25
63	62,35,Checking,6300.50
64	63,84,Savings,525.75
65	64,12,Checking,6500.00
66	65,69,Savings,750.25
67	66,26,Checking,6700.50
68	67,96,Savings,575.75
69	68,8,Checking,6900.00
70	69,59,Savings,600.25
71	70,33,Checking,7100.50
72	71,73,Savings,625.75
73	72,17,Checking,7300.00
74	73,87,Savings,650.25
75	74,43,Checking,7500.50
76	75,61,Savings,675.75
77	76,22,Checking,7700.00
78	77,91,Savings,700.25
79	78,4,Checking,7900.50
80	79,55,Savings,725.75
81	80,30,Checking,8100.00
82	81,70,Savings,750.25
83	82,2,Checking,8300.50
84	83,82,Savings,775.75
85	84,40,Checking,8500.00
86	85,65,Savings,800.25
87	86,21,Checking,8700.50
88	87,93,Savings,825.75
89	88,1,Checking,8900.00
90	89,54,Savings,850.25
91	90,38,Checking,9100.50
92	91,77,Savings,875.75
93	92,44,Checking,9300.00
94	93,79,Savings,900.25
95	94,39,Checking,9500.50
96	95,60,Savings,925.75
97	96,48,Checking,9700.00

Here we are inserting new rows in the CSV file.

Accounts/accounts.csv

Blob

Save Discard Download Refresh | Delete

Overview Versions Edit Generate SAS

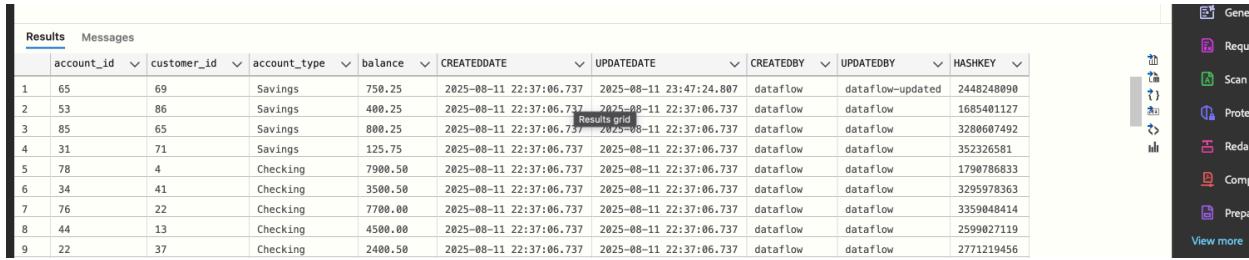
63	62,35,Checking,6300.50
64	63,84,Savings,525.75
65	64,12,Checking,6500.00
66	65,69,Savings,750.25
67	66,26,Checking,6700.50
68	67,96,Savings,575.75
69	68,8,Checking,6900.00
70	69,59,Savings,600.25
71	70,33,Checking,7100.50
72	71,73,Savings,625.75
73	72,17,Checking,7300.00
74	73,87,Savings,650.25
75	74,43,Checking,7500.50
76	75,61,Savings,675.75
77	76,22,Checking,7700.00
78	77,91,Savings,700.25
79	78,4,Checking,7900.50
80	79,55,Savings,725.75
81	80,30,Checking,8100.00
82	81,70,Savings,750.25
83	82,2,Checking,8300.50
84	83,82,Savings,775.75
85	84,40,Checking,8500.00
86	85,65,Savings,800.25
87	86,21,Checking,8700.50
88	87,93,Savings,825.75
89	88,1,Checking,8900.00
90	89,54,Savings,850.25
91	90,38,Checking,9100.50
92	91,77,Savings,875.75
93	92,44,Checking,9300.00
94	93,79,Savings,900.25
95	94,39,Checking,9500.50
96	95,60,Savings,925.75
97	96,48,Checking,9700.00
98	97,90,Savings,950.25
99	98,49,Checking,9900.50
100	99,80,Savings,975.75
101	100,50,Checking,10100.00
102	101,100,Savings,523.12
103	

Csv

Preview

After running the silver layer and gold layer(SCD Type-1) data flow. We are getting the updated rows and newly inserted rows in the SCD-type table.

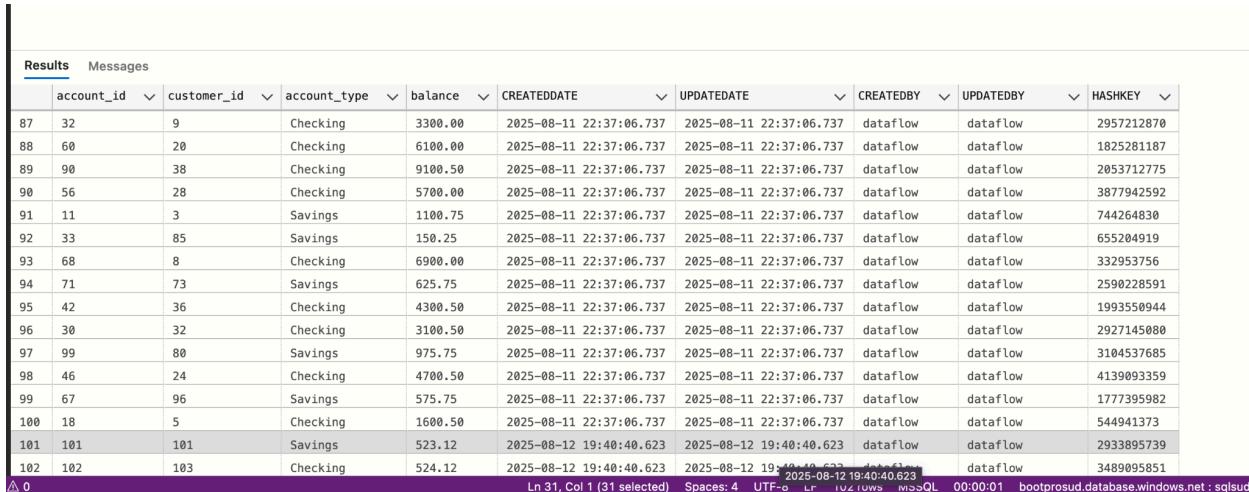
Updated row



A screenshot of a database interface showing a table named 'Results' with 9 rows. The columns are: account_id, customer_id, account_type, balance, CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, and HASHKEY. The last row (row 9) has its 'customer_id' value changed from 37 to 37, indicating an update. The interface includes a toolbar with various icons and a sidebar with navigation links.

	Results Messages								
	account_id	customer_id	account_type	balance	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
1	65	69	Savings	750.25	2025-08-11 22:37:06.737	2025-08-11 23:47:24.807	dataflow	dataflow-updated	2448248890
2	53	86	Savings	400.25	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	1685401127
3	85	65	Savings	800.25	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	3280607492
4	31	71	Savings	125.75	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	352326581
5	78	4	Checking	7900.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	1790786833
6	34	41	Checking	3500.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	3295978363
7	76	22	Checking	7700.00	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	3359048414
8	44	13	Checking	4500.00	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	2599027119
9	22	37	Checking	2400.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	2771219456

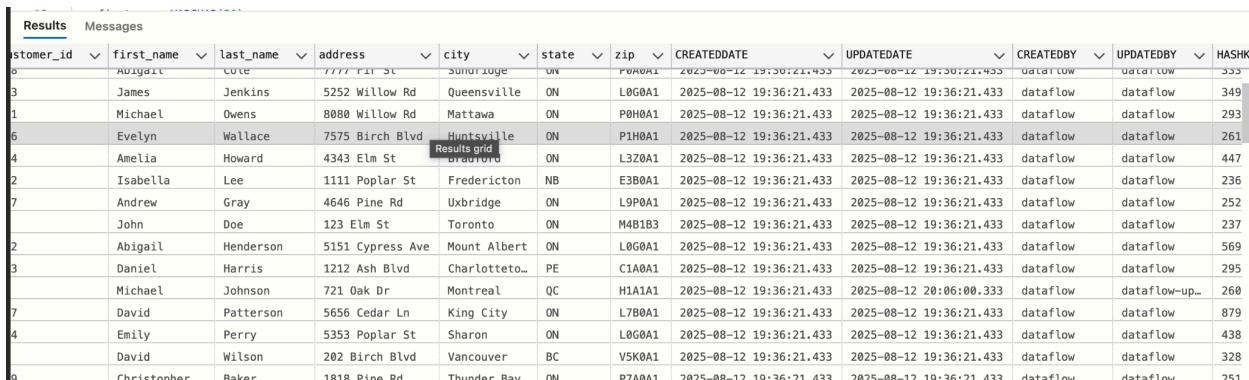
Inserted row



A screenshot of a database interface showing a table named 'Results' with 102 rows. The columns are: account_id, customer_id, account_type, balance, CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, and HASHKEY. The last row (row 102) has a new entry with customer_id 103, account_type 'Checking', and balance 524.12, indicating an insert. The interface includes a toolbar with various icons and a sidebar with navigation links.

	Results Messages								
	account_id	customer_id	account_type	balance	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
87	32	9	Checking	3300.00	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	2957212870
88	60	20	Checking	6100.00	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	1825281187
89	90	38	Checking	9100.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	2053712775
90	56	28	Checking	5700.00	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	3877942592
91	11	3	Savings	1100.75	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	744264830
92	33	85	Savings	150.25	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	655204919
93	68	8	Checking	6900.00	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	332953756
94	71	73	Savings	625.75	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	2590228591
95	42	36	Checking	4300.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	1993550944
96	30	32	Checking	3100.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	2927145080
97	99	80	Savings	975.75	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	3104537685
98	46	24	Checking	4700.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	4139093359
99	67	96	Savings	575.75	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	1777395982
100	18	5	Checking	1600.50	2025-08-11 22:37:06.737	2025-08-11 22:37:06.737	dataflow	dataflow	544941373
101	101	101	Savings	523.12	2025-08-12 19:40:40.623	2025-08-12 19:40:40.623	dataflow	dataflow	2933895739
102	102	103	Checking	524.12	2025-08-12 19:40:40.623	2025-08-12 19:40:40.623	dataflow	dataflow	3489095851

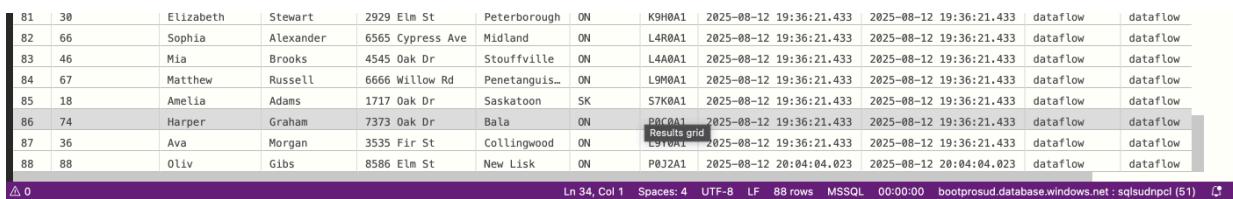
Inserted row



A screenshot of a database interface showing a table named 'Results' with 88 rows. The columns are: customer_id, first_name, last_name, address, city, state, zip, CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, and HASHKEY. The last row (row 88) has a new entry with customer_id 103, first_name 'Baker', last_name 'Christopher', address '1818 Pine Rd', city 'Thunder Bay', state 'ON', zip 'P7A0A1', indicating an insert. The interface includes a toolbar with various icons and a sidebar with navigation links.

	Results Messages											HASHKEY
	customer_id	first_name	last_name	address	city	state	zip	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	
0	1	Augie	Carl	1111 Elm St	Summerville	ON	P0H0A1	2025-08-12 19:30:21.433	2025-08-12 19:30:21.433	dataflow	dataflow	555
1	3	James	Jenkins	5252 Willow Rd	Queensville	ON	L0G0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	349
2	1	Michael	Owens	8080 Willow Rd	Mattawa	ON	P0H0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	293
3	6	Evelyn	Wallace	7575 Birch Blvd	Huntsville	ON	P1H0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	261
4	2	Amelia	Howard	4343 Elm St	Broadford	ON	L3Z0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	447
5	7	Isabella	Lee	1111 Poplar St	Fredericton	NB	E3B0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	236
6	3	Andrew	Gray	4646 Pine Rd	Uxbridge	ON	L9P0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	252
7	2	John	Doe	123 Elm St	Toronto	ON	M4B1B3	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	237
8	4	Abigail	Henderson	5151 Cypress Ave	Mount Albert	ON	L0G0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	569
9	3	Daniel	Harris	1212 Ash Blvd	Charlottetown	PE	C1A0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	295
10	1	Michael	Johnson	721 Oak Dr	Montreal	QC	H1A1A1	2025-08-12 19:36:21.433	2025-08-12 20:06:00.333	dataflow	dataflow-updated	268
11	7	David	Patterson	5656 Cedar Ln	King City	ON	L7B0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	879
12	4	Emily	Perry	5353 Poplar St	Sharon	ON	L0G0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	438
13	5	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	328
14	6	Christopher	Baker	1818 Pine Rd	Thunder Bay	ON	P7A0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	251

Updated row



A screenshot of a database interface showing a table named 'Results' with 88 rows. The columns are: customer_id, first_name, last_name, address, city, state, zip, CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, and HASHKEY. The last row (row 88) has its 'first_name' value changed from 'Oliv' to 'Gib', indicating an update. The interface includes a toolbar with various icons and a sidebar with navigation links.

	customer_id	first_name	last_name	address	city	state	zip	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
81	30	Elizabeth	Stewart	2929 Elm St	Peterborough	ON	K9H0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
82	66	Sophia	Alexander	6565 Cypress Ave	Midland	ON	L4R0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
83	46	Mia	Brooks	4545 Oak Dr	Stouffville	ON	L4A0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
84	67	Matthew	Russell	6666 Willow Rd	Penetanguishene	ON	L9M0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
85	18	Amelia	Adams	1717 Oak Dr	Saskatoon	SK	S7K0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
86	74	Harper	Graham	7373 Oak Dr	Bala	ON	P0C0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
87	36	Ava	Morgan	3535 Fir St	Collingwood	ON	L9T0A1	2025-08-12 19:36:21.433	2025-08-12 19:36:21.433	dataflow	dataflow	
88	88	Oliv	Gibs	8586 Elm St	New Lisk	ON	P0J2A1	2025-08-12 20:04:04.023	2025-08-12 20:04:04.023	dataflow	dataflow	

Ln 34, Col 1 Spaces: 4 UTF-8 LF 88 rows MSSQL 00:00:00 bootprod.database.windows.net : sqlspncl (51)