

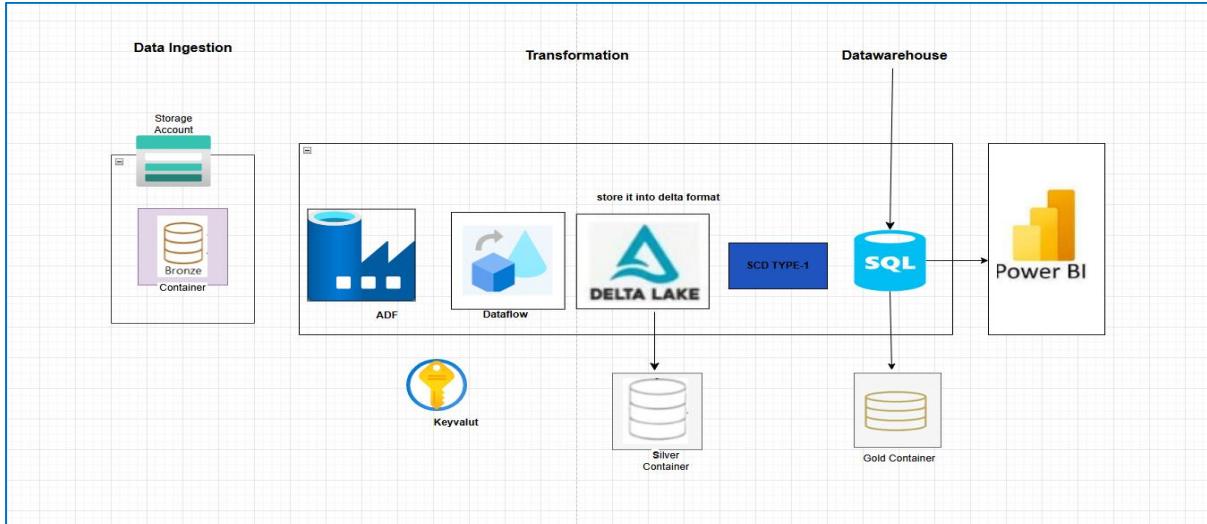
Bootcamp project1-ADF, Dataflows and SCD Type1

Table of Contents

Data Ingestion (Backend Storage to Raw(Bronze) Container)	3
Remove duplicates by using window function	8
What rowNumber() Does:	8
Conditional Split to sepearte duplicate primary keys.	11
AlterRow to mark the rows to deside to be inserted or updated when writing data to the sink	12
Select Coloumn to select the required coloumns to send to silver layer	13
Add sink and save data into delta format to silver layer of storage account	14
Accounts csv file cleaning	20
Join to combine customers and accounts data to select matched records which have relationship	22
Window function to remove duplicate keys in accounts	24
Conditional Split to sepearte duplicate records and keys without duplicates....	26
AlterRow to deside the rows to be inserted or updated.....	27
Select function to remove unneccesary coloumns.....	28
Sink to add the good records.....	29
Silver to Gold	36
Step3 : Perform SCD Type1 on Customers delta file which is stored to silvercontainer storage account.....	36
Source as Customers delta file data	37
Target as sql, so open ssms and create a table.....	38
Create Target DIM_Customers table in Azure SQL database(SSMS)	38
Select select to rename source file coloumns using (1==1 concat('src_',\$\$)) with src	43
Add Derrive coloumn to generate hashkey	46
Add join to combine source and target coloumns	48
Add Conditional split to seperate insert an dupdate data rows	50
Now add 1more derrived coloumn to add Audit coloumns(CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY)	52

Now add Sink to load the final data(insert new records)	53
Derived column for update	55
Add sink for update data.....	57

Architecture Diagram

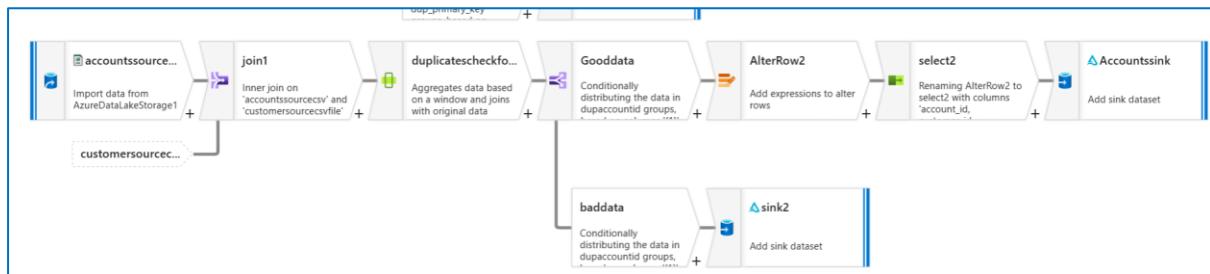
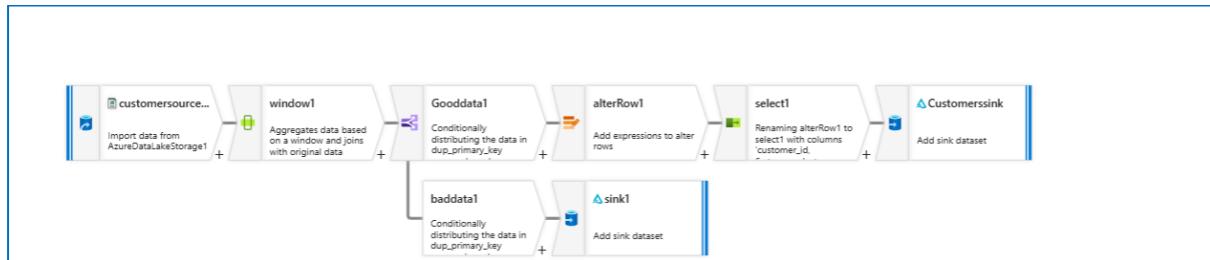


- Next is, to store the data in a folder called “Bronze” and he will place the files in PARQUET format in sub folders.
- Next step is to combine the extracted data which also includes to perform cleanup activity in the data or to check if there are any null values are present.
- Once data is cleaned/combined, perform SCD type transformation and keep the data in a dedicated SQLPOOL called as Datawarehouse.
- To get this all done Data Engineer has to develop a pipeline and once the pipeline is developed. He will perform ETL activities as mentioned in the above points.
- If there are any failures happening on them, Data Engineer has to fix those things.

Step 1:

Data Ingestion (Backend Storage to Raw(Bronze) Container)

- Extract data from from kaggle dataset link is provided below csv format and then load the extracted data into cloud ADLS GEN2 Raw(Bronze) Container.
- Link: <https://www.kaggle.com/datasets/varunkumari/ai-bank-dataset>
- Download csv files from kaggle website and upload it into storage account bronze container which contains raw daw data, so place our raw data into bronze container
- Create container in your storage account to add bronze directory then upload raw data(csv files)



Name	Last modified	Access tier	Block type	Size	Lease state
accounts.csv	8/9/2023, 1:49:01 PM	Hot (Inferred)	Block blob	2.28 Kib	Available
customers.csv	8/9/2023, 1:49:01 PM	Hot (Inferred)	Block blob	4.5 Kib	Available
loan_payments.csv	8/9/2023, 1:49:01 PM	Hot (Inferred)	Block blob	2.55 Kib	Available
loans.csv	8/9/2023, 1:49:01 PM	Hot (Inferred)	Block blob	2.29 Kib	Available
transactions.csv	8/9/2023, 1:49:01 PM	Hot (Inferred)	Block blob	3.43 Kib	Available

- Select dataflow activity and click on new at dataflow where you will be able to select your source
- You need the dataset as delimitedText(csv) and linkedservice which makes a connection to your storage account to load the csv files.

Microsoft Azure | Data Factory | adfcustomerproject1

Would you like to see Data Factory inside of Microsoft Fabric? Microsoft's newest cloud-first data analytics SaaS platform? Click [here](#) to get started with Fabric Data Factory!

Validate all | Publish all | Search factory and documentation

Factory Resources

- Pipelines
- Change Data Capture (preview)
- Datasets
- Data flows
- Power Query

Validate | Data flow debug

pl_bootcamp1adf | dataflow1

accountsfilresource

Column: 0 total

Add Source | +

Source settings | Source options | Projection | Optimize | Inspect | Data preview

Output stream name: accountsfilresource | Learn more

Description: Add source dataset | Reset

Source type: Dataset | Inline

Inline dataset type: DelimitedText

Linked service: AzureDataLakeStorage1 | Test connection | Edit | New

Skip line count:

Sampling: Enable | Disable

Edit linked service

Name: Azure Data Lake Storage Gen2 | Learn more

Description:

Connect via integration runtime: AutoResolveIntegrationRuntime

Authentication type: Account key (highlighted)

Account selection method: Enter manually

URL: https://adlsgen2storagebehav.dfs.core.windows.net/

Storage account key: Azure Key Vault

Storage account key:

Test connection: To linked service (highlighted)

Annotations: New | Parameters | Advanced

Apply | Cancel | Connection successful (highlighted)

Microsoft Azure | Upgrade

Home > Storage accounts > adlsgen2storagebehav

Storage accounts | adlsgen2storagebehav | Access keys

You are viewing a new version of Browse experience. Click here to access the old experience.

+ Create | Restore | ...

Name: adlsgen2storagebehav | blobstrsnowflake

Overview | Activity log | Tags | Diagnose and solve problems | Access Control (IAM) | Data migration | Events | Storage browser | Partner solutions | Resource visualizer | Data storage | Security + networking | Networking | Access keys (highlighted)

key1 | Rotate key | Last rotated: 7/27/2025 (13 days ago)

Key: [REDACTED] | Show

Connection string: [REDACTED] | Show

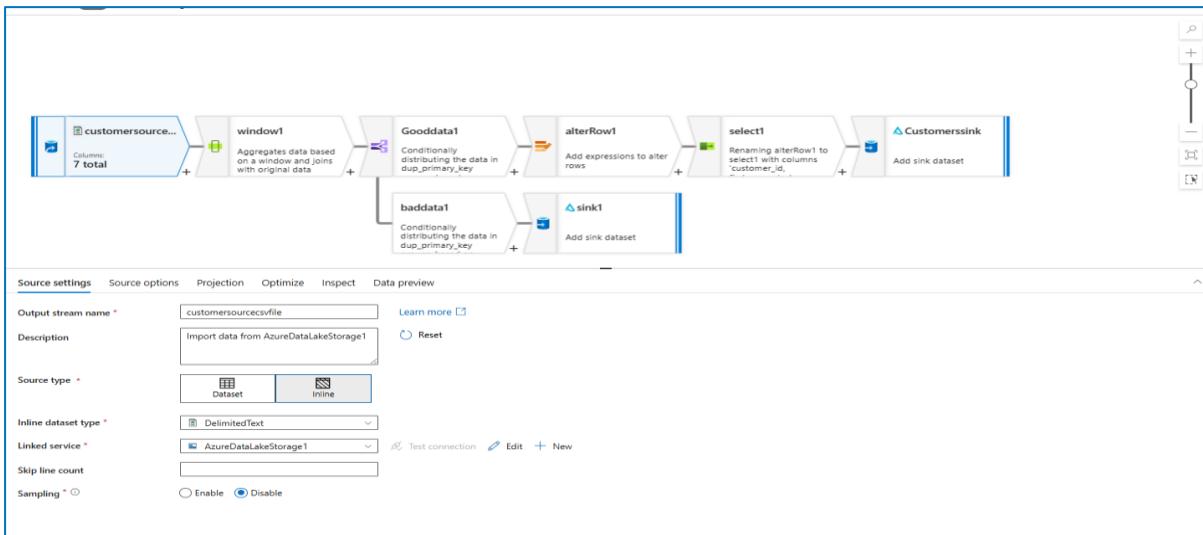
key2 | Rotate key | Last rotated: 7/27/2025 (13 days ago)

Key: [REDACTED] | Show

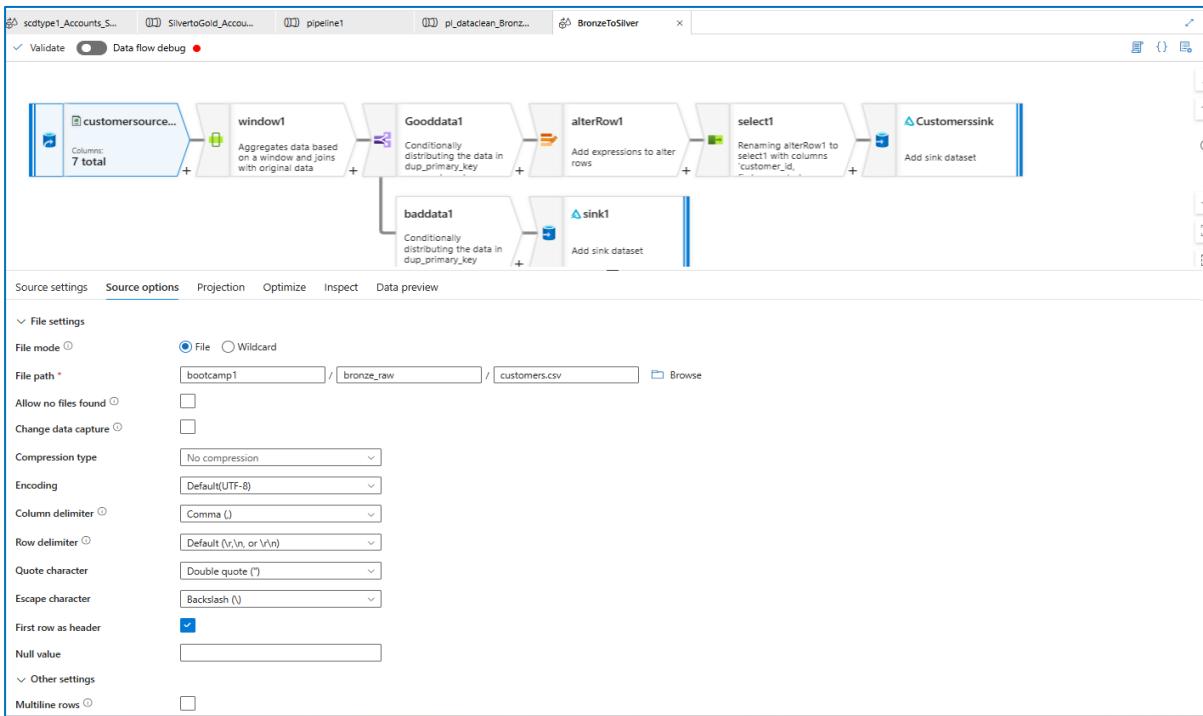
Connection string: [REDACTED] | Show

Show 1 - 2 of 2. Display count: auto

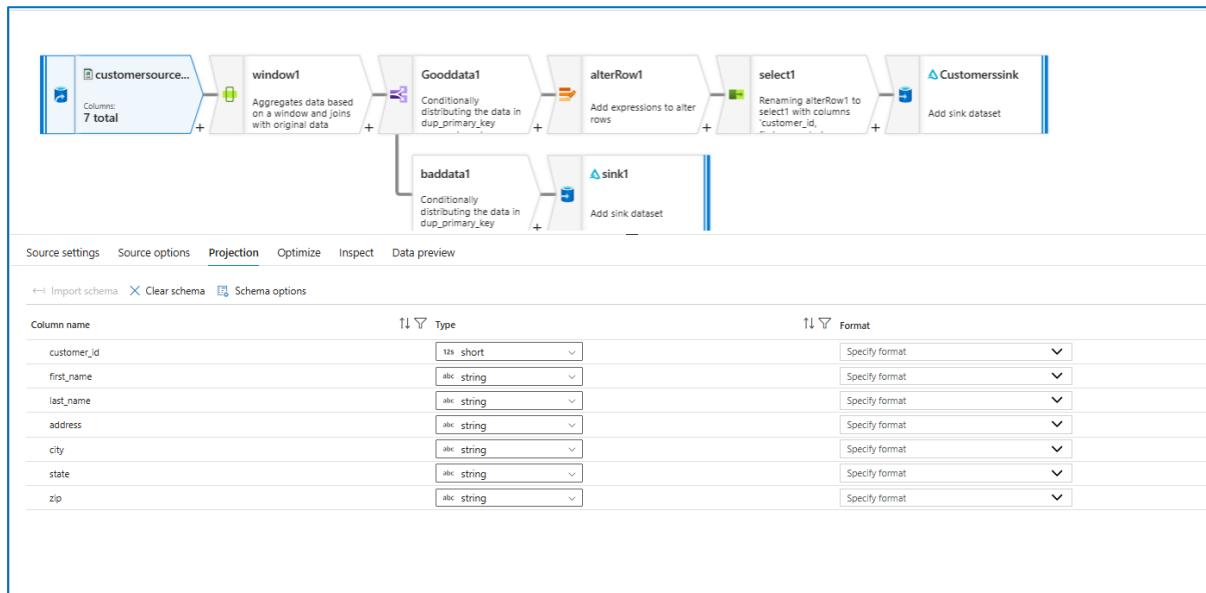
Customers file Cleaning(Bronze to Silver)



Select the file from container from source options and Make sure you have to select first row as header



Do projection and import schema then only you will be able to see the csv columns
data in the preview tab



Do preview

customer_id	first_name	last_name	address	city	state	zip
1	John	Doe	123 Elm St	Toronto	ON	M4B1B3
2	Jane	Smith	456 Maple Ave	Ottawa	ON	K1A0B1
3	Michael	Johnson	789 Oak Dr	Montreal	QC	H1A1A1
4	Emily	Davis	101 Pine Rd	Calgary	AB	T2A0A1
5	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1
6	Emma	Clark	505 Cedar St	Halifax	NS	B3H0A1
7	James	Martinez	606 Spruce Ln	Winnipeg	MB	R3C0A1

Remove duplicates by using window function

Note: We have to consider primary key as to check the duplicate records.

To find which column we have primary key in given data, if it is csv file you have to select each column and check the duplicates in excel if any column which does not have any duplicate records then we can consider it as a Primary key for that particular file data.

In customer file i have checked each column and every column has duplicates other than customer_id so I'm making customer_id as primary key for customer csv file to remove duplicates

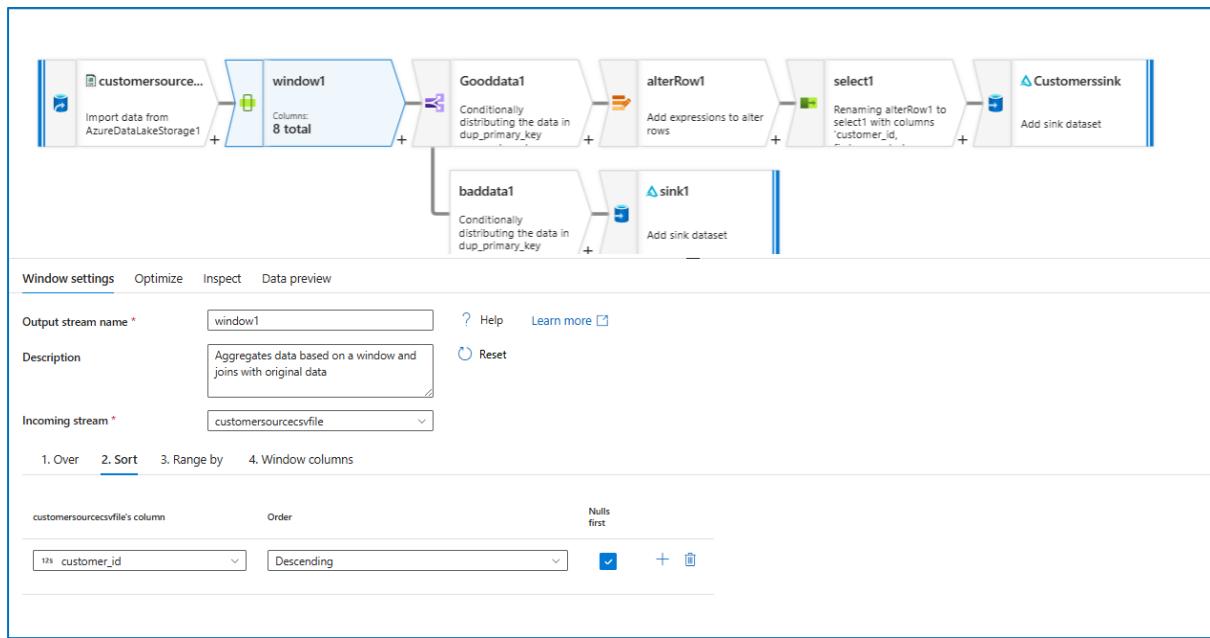
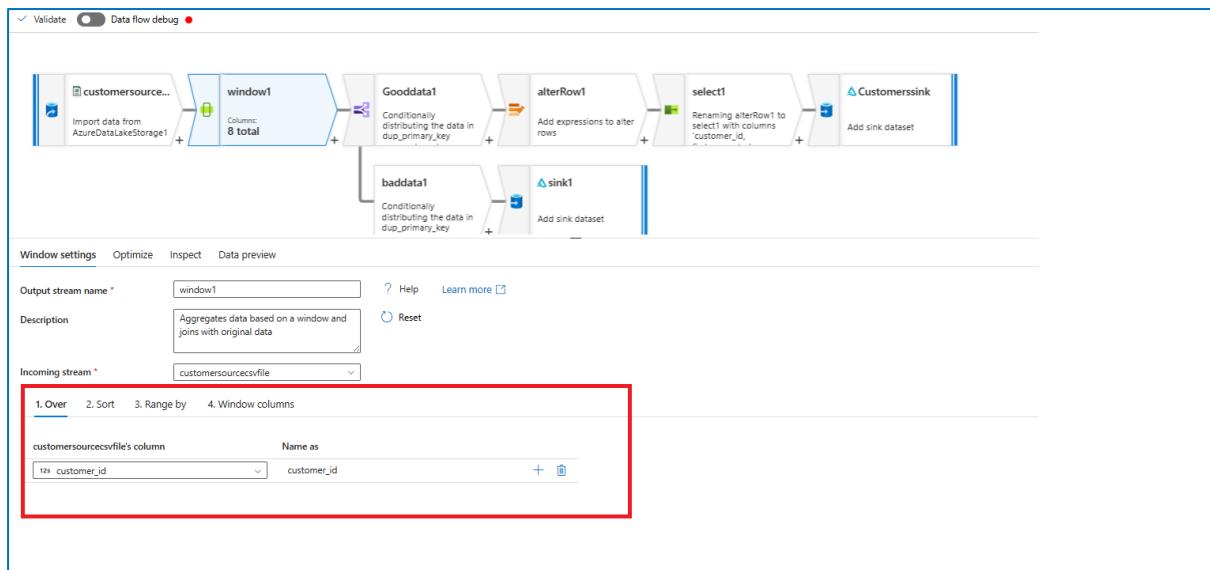
Make sure your dataflow debug should be on to see the loaded csv data.

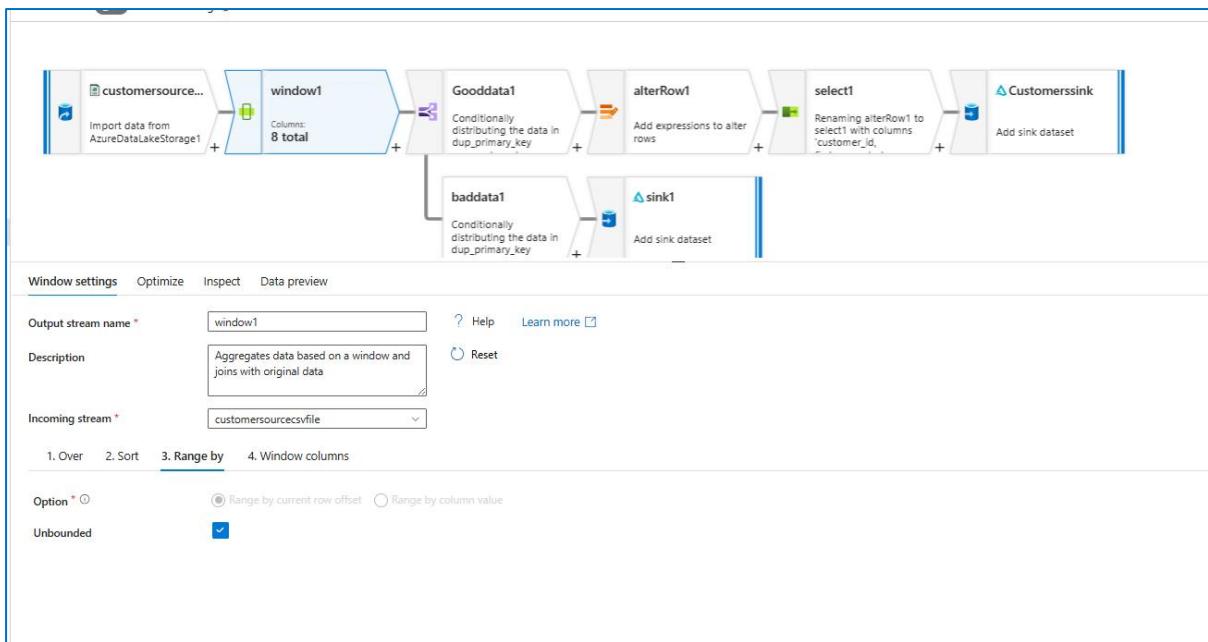
A	B	C	D	E	F	G	H	I	J	K	L	M
28	James	Evans	2626 Beech Guelph	ON	N1H0A1							
29	Emily	Edwards	2727 Beech Brantford	ON	N3T0A1							
30	Michael	Collins	2828 Cedar Thunder Bay	ON	P7B0A1							
31	Elizabeth	Stewart	2929 Elm Street Peterborough	ON	K9H0A1							
32	David	Sanchez	3030 Maple North Bay	ON	P1B0A1							
33	Sophia	Morris	3131 Oak Street Belleville	ON	K8N0A1							
34	John	Rogers	3232 Pine Street Timmins	ON	P4N0A1							
35	Olivia	Reed	3333 Birch Street Orillia	ON	L3V0A1							
36	William	Cook	3434 Spruce Midland	ON	L4R0A1							
37	Ava	Morgan	3535 Fir Street Collingwood	ON	L9Y0A1							
38	Alexander	Bell	3636 Redwood Stratford	ON	N5A0A1							
39	Isabella	Murphy	3737 Cypress Woodstock	ON	N4S0A1							
40	Daniel	Bailey	3838 Willow Orangeville	ON	L9W0A1							
41	Sophia	Rivera	3939 Poplar Milton	ON	L9T0A1							
42	Matthew	Cooper	4040 Ash Street Georgetown	ON	L7G0A1							
43	Charlotte	Richardson	4141 Beech Newmarket	ON	L3Y0A1							
44	Joseph	Cox	4242 Cedar Aurora	ON	L4G0A1							
45	Amelia	Howard	4343 Elm Street Bradford	ON	L3Z0A1							
46	Christopher	Ward	4444 Maple Keswick	ON	L4P0A1							
47	Mia	Brooks	4545 Oak Street Stouffville	ON	L4A0A1							
48	Andrew	Gray	4646 Pine Street Uxbridge	ON	L9P0A1							
49	Harper	James	4747 Birch Street Port Perry	ON	L9L0A1							
50	Joshua	Bennett	4848 Spruce Beaverton	ON	L0K0A1							
51	Evelyn	Barnes	4949 Fir Street Sutton	ON	L0E0A1							
52	Daniel	Ross	5050 Redwood Pefferlaw	ON	L0E0A1							
53	Abigail	Henderson	5151 Cypress Mount Albion	ON	L0G0A1							
54	James	Jenkins	5252 Willow Queen'sville	ON	L0G0A1							
55	Emily	Perry	5353 Poplar Sharon	ON	L0G0A1							
56	Michael	Butler	5454 Ash Street Holland Landing	ON	L9N0A1							
57	Elizabeth	Long	5555 Beech East Gwillimbury	ON	L9N0A1							
58	David	Patterson	5656 Cedar King City	ON	L7B0A1							
59	Sophia	Hughes	5757 Elm Street Nobleton	ON	L0G0A1							
60	John	Flores	5858 Maple Schomber	ON	L0G0A1							
61	Olivia	Washington	5959 Oak Street Tottenham	ON	L0G0A1							
62	William	Butler	6060 Pine Street Alliston	ON	L9R0A1							

Window condition: Over by Customer_id sory by customer_id dessecnding order window columns rowNumber()

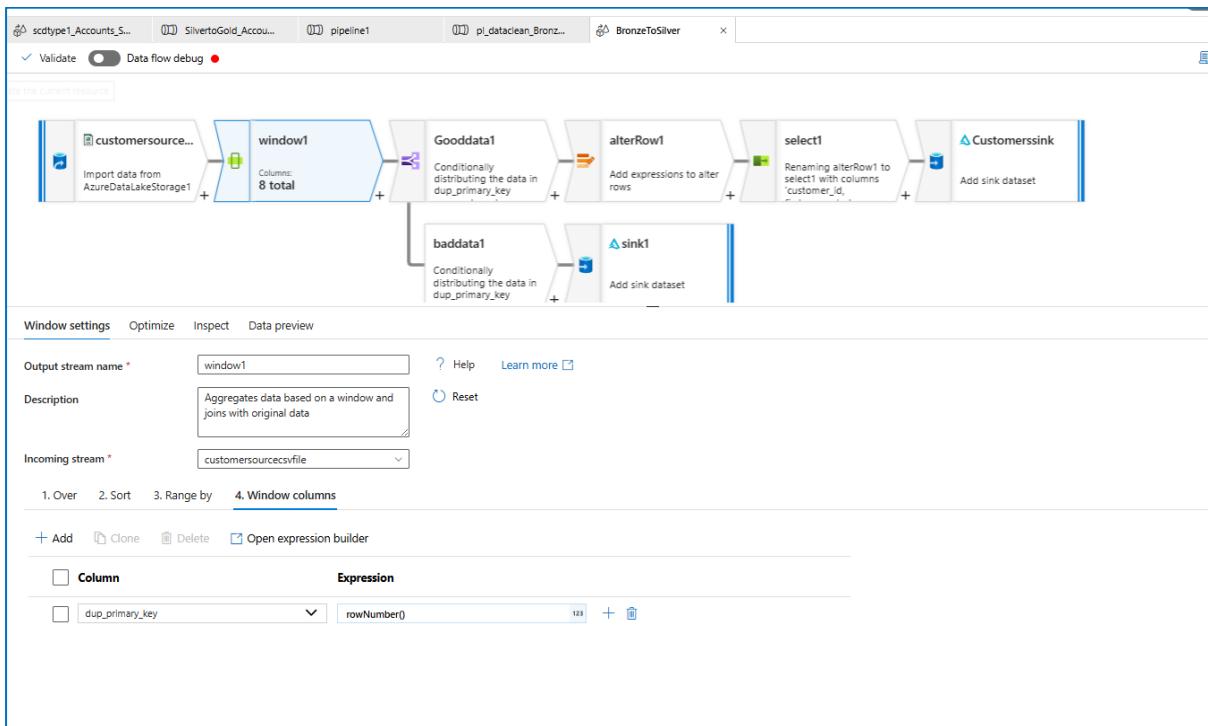
What rowNumber() Does:

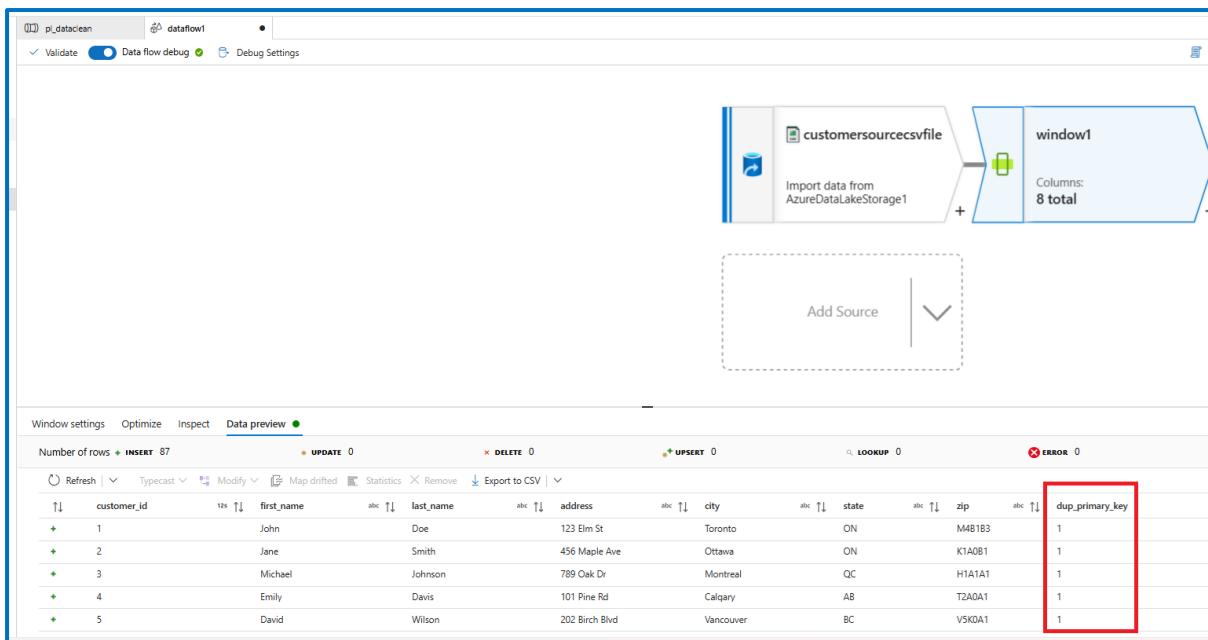
- Assigns a sequential number to rows.
- The numbering starts at 1 for each partition.
- The order of rows is controlled by the "order by" clause in the window transformation.
- The scope (i.e. partitioning) is controlled by the "partition by" clause.





There are no duplicates found for primary key values

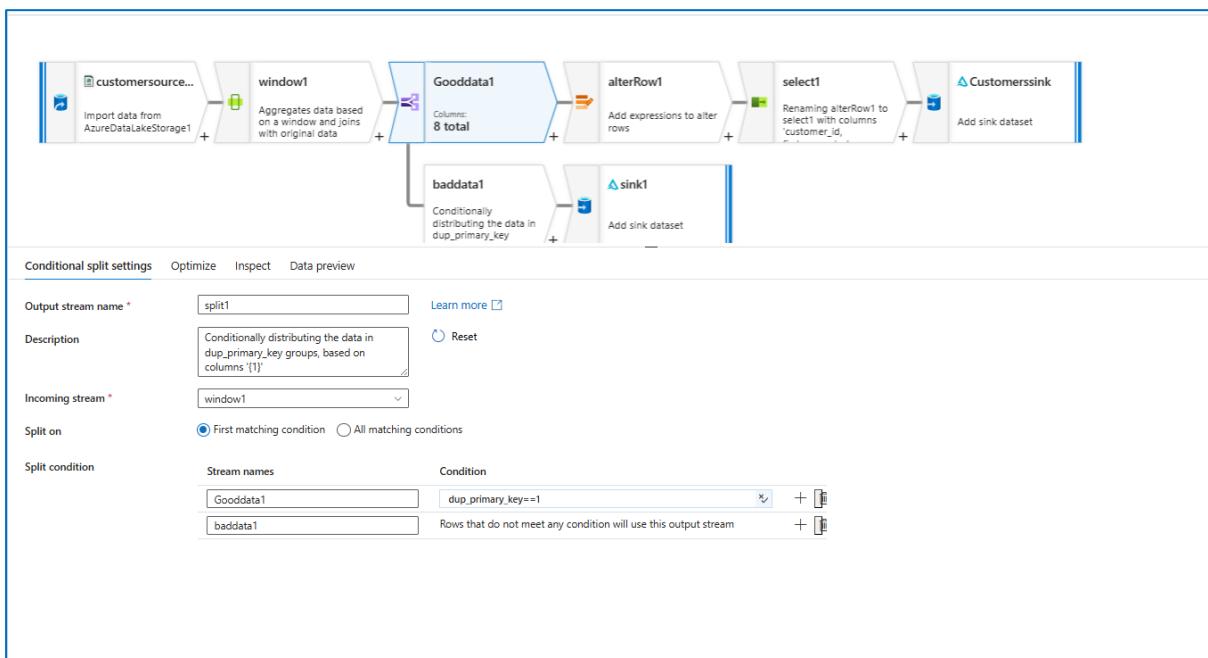


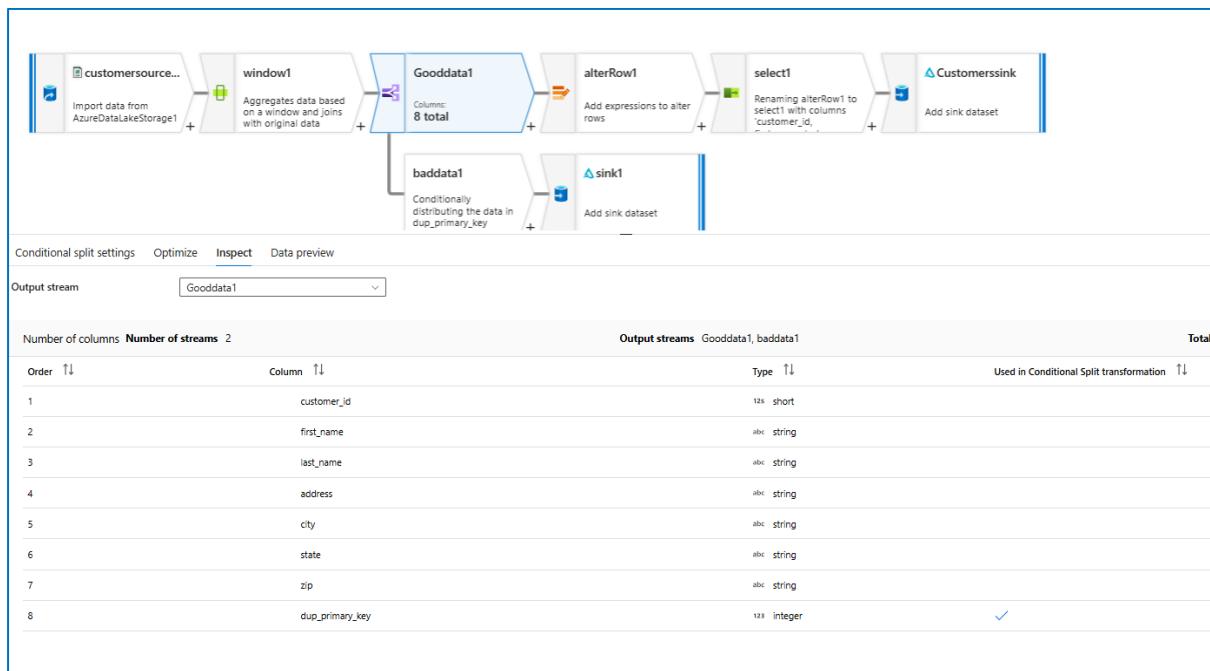


Conditional Split to separate duplicate primary keys.

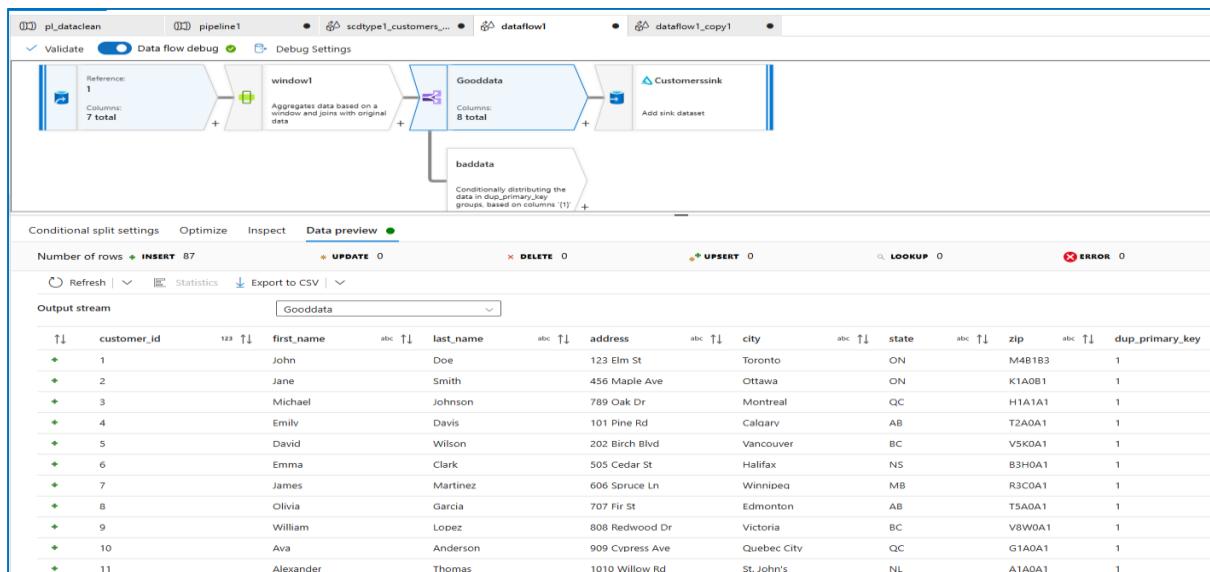
We go with conditional split if we want to separate duplicates keys into bad records file and good records into good file cleaned data.

I will go with dupidprimarykey==1 condition, so that it will carryforward only records which dont have duplicate primary keys(customer_id)



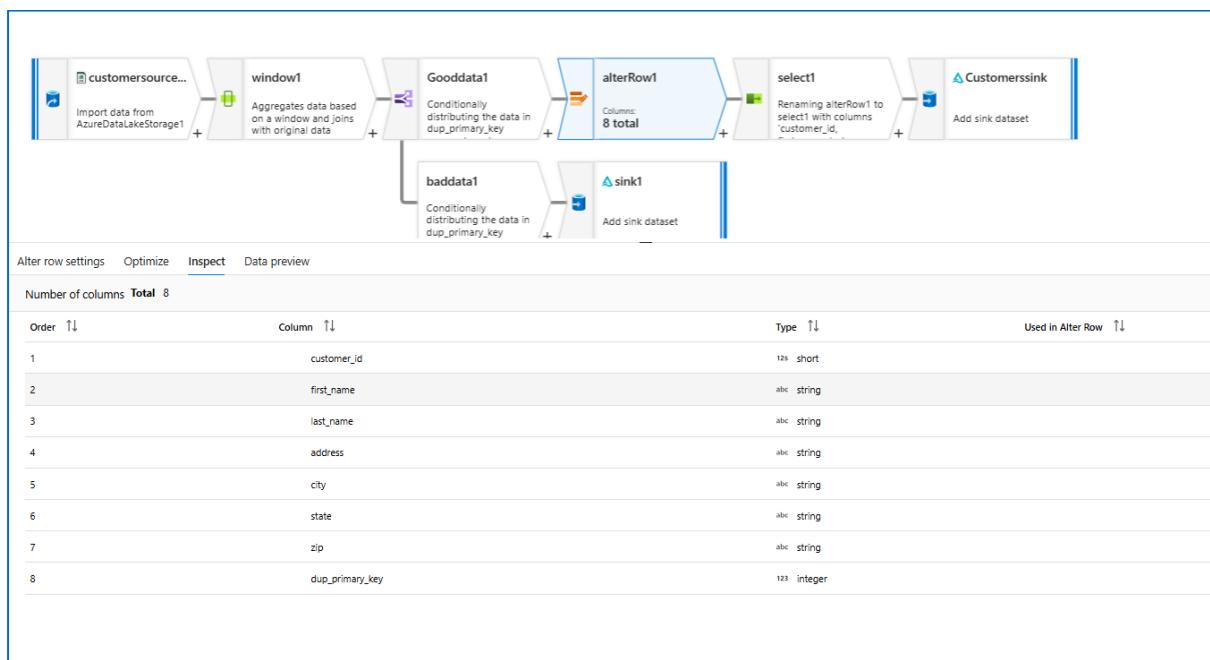
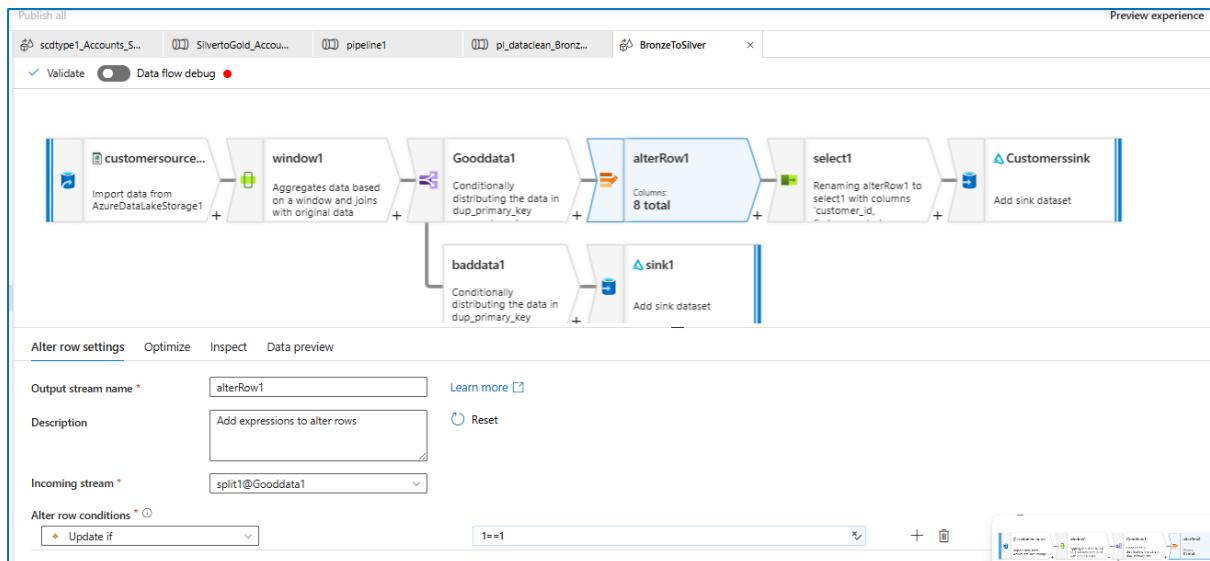


Do datapreview and see the data



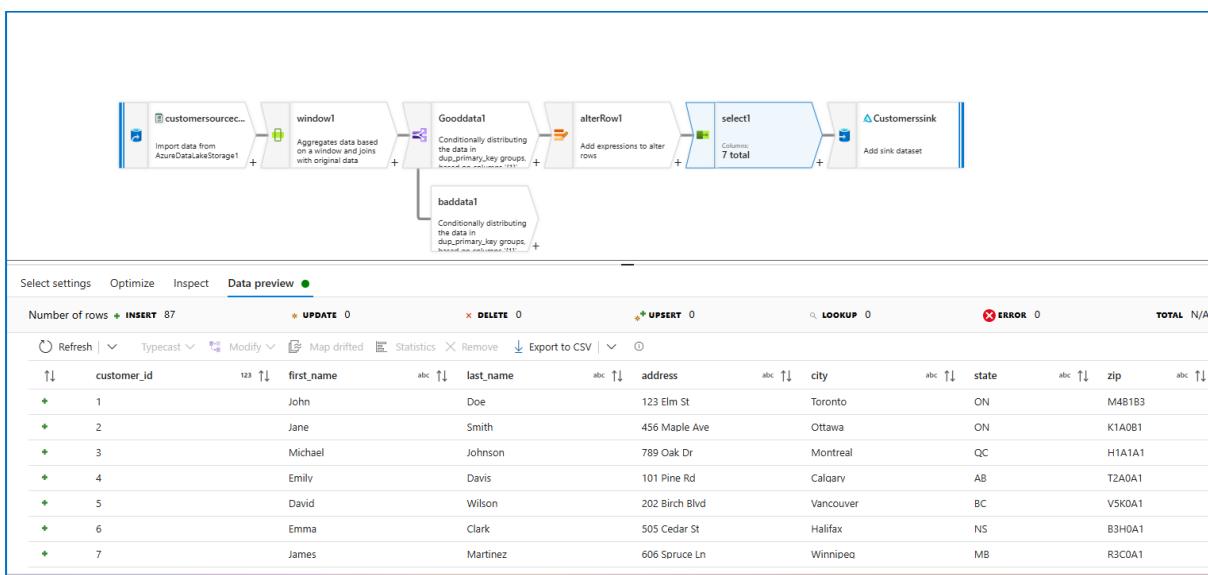
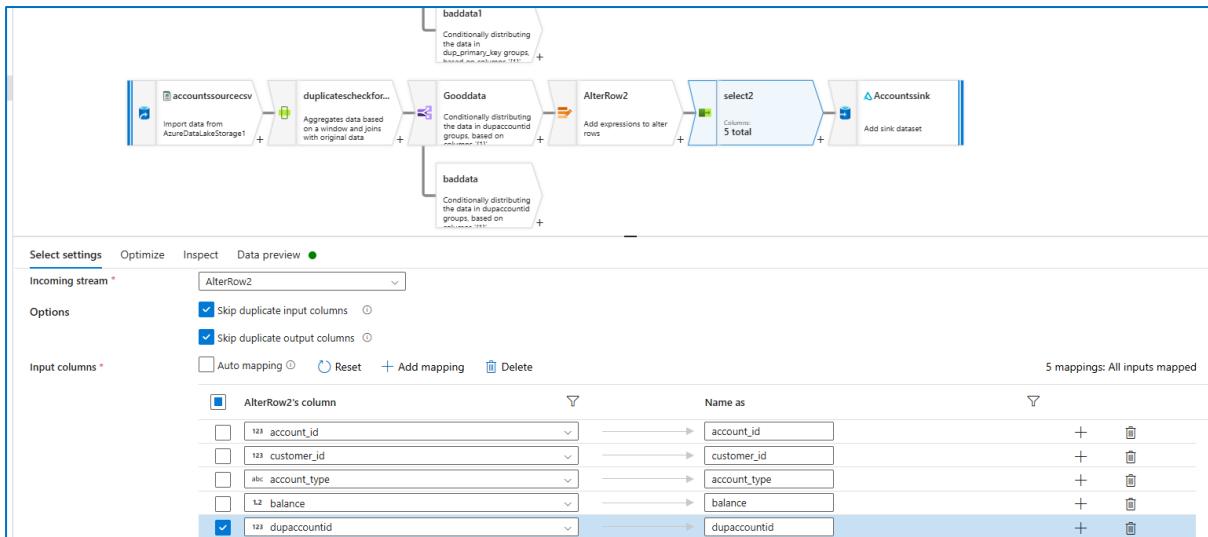
AlterRow to mark the rows to deside to be inserted or updated when writing data to the sink

For upsert operation sink needs alterrow before sink



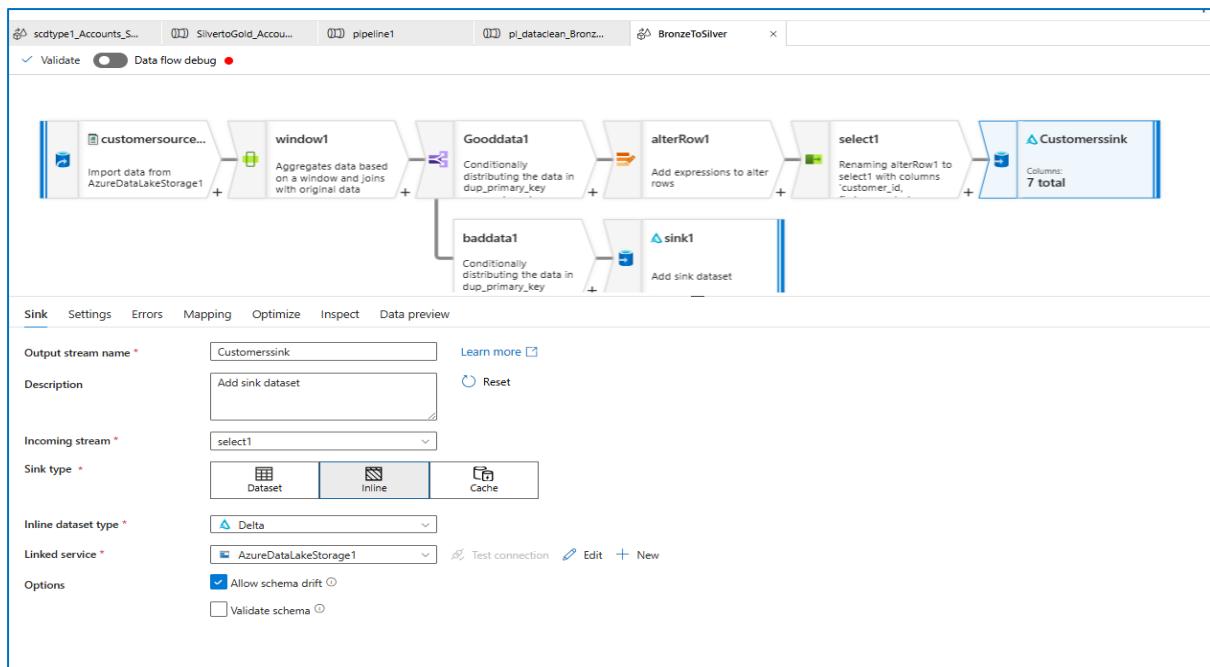
Select Column to select the required columns to send to silver layer

Use select column to rename and remove duplicate id column before we send it to sink(silver layer)



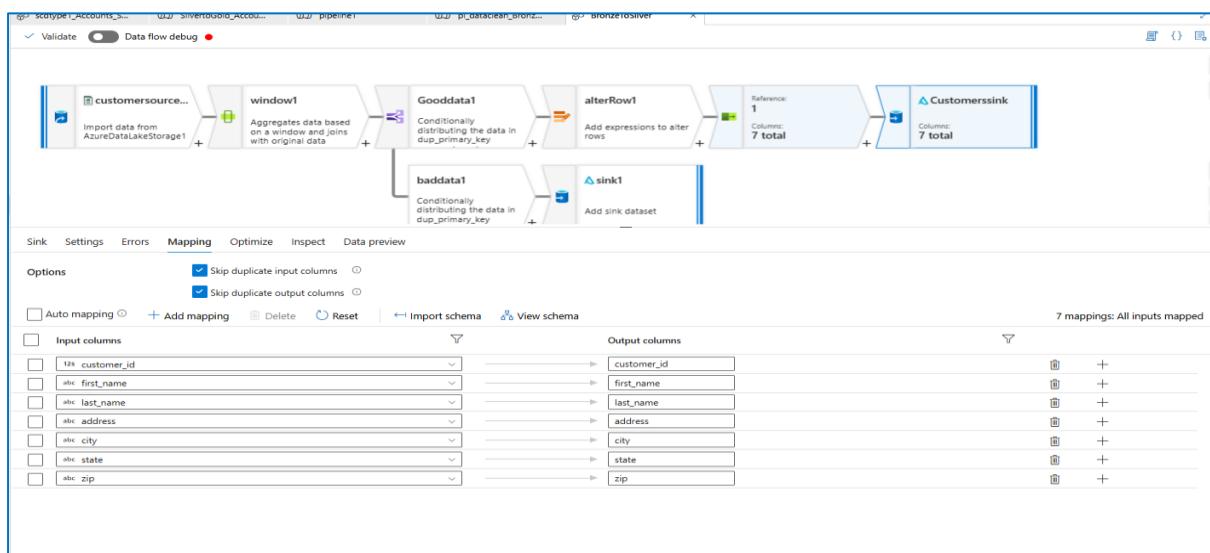
Add sink and save data into delta format to silver layer of storage account

Note: in the mapping tab uncheck automapping and remove dup_primarykey as we dont need it

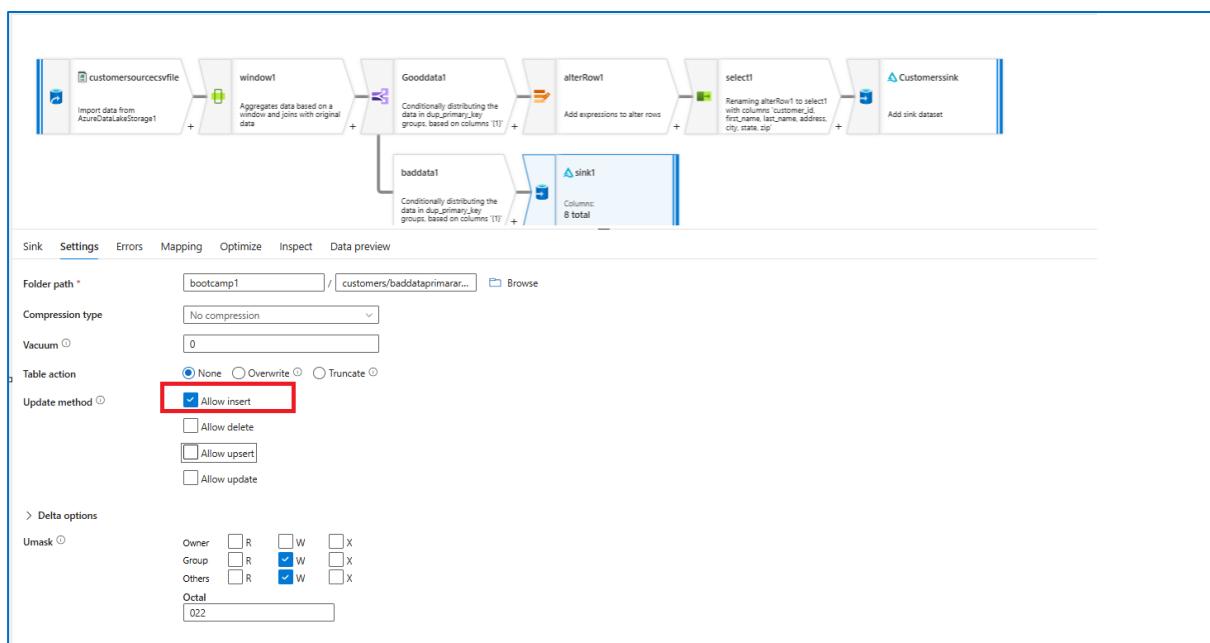
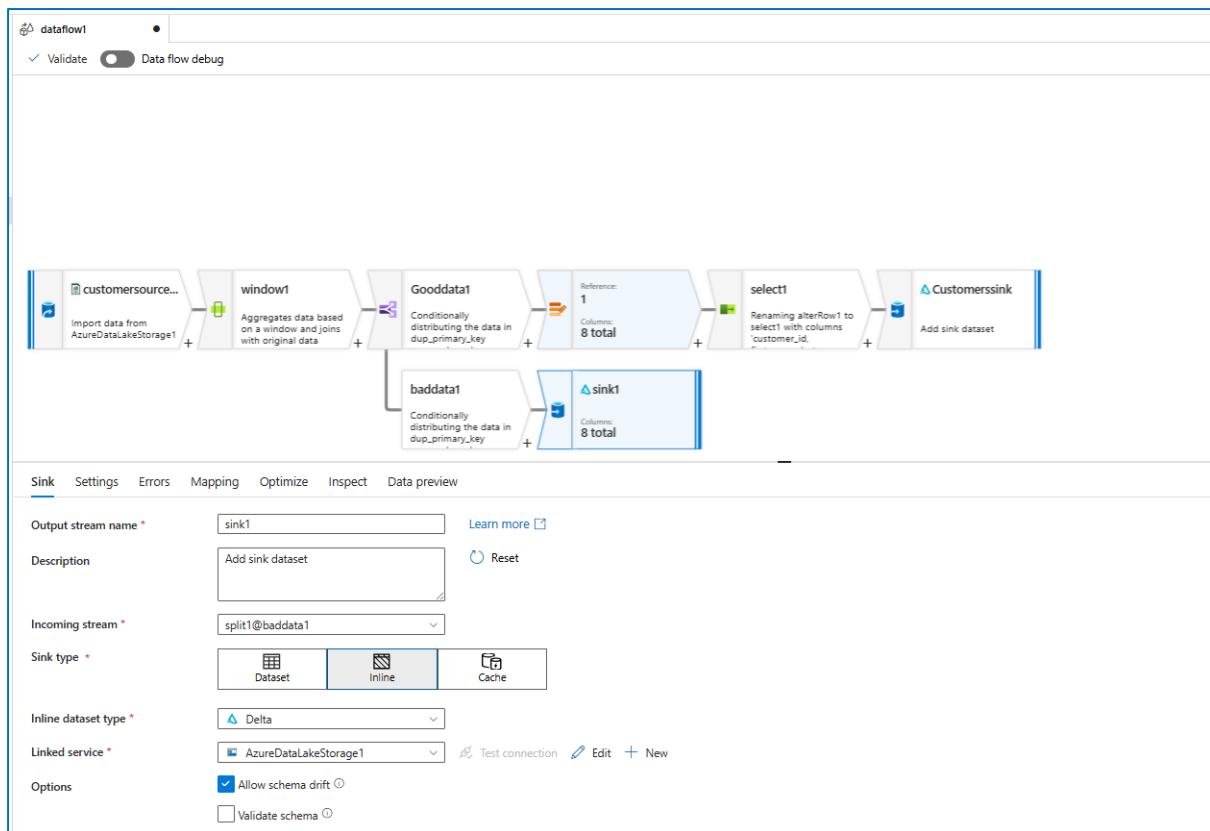


In settings select **upsert** which means either of the conditions of insert and update it will handle

Based on **customer_id** it will decide whether it has to insert or update it will decide before adding data into sink and do mapping.



Bad data into sink



Input

```

Input
Copy to clipboard

{
  "dataflow": {
    "referenceName": "dataflow1",
    "type": "DataFlowReference",
    "parameters": {},
    "datasetParameters": {
      "customersourcecsvfile": {},
      "sink1": {}
    }
  },
  "staging": 0,
  "compute": {
    "coreCount": 8,
    "computeType": "General"
  },
  "traceLevel": "Fine",
  "dataFlowDebugSessionId": "6024589f-18f0-429f-92e9-9ba286c3c267",
  "continuationSettings": {
    "customizedCheckpointKey": "pl_dataclean-Data flow1-70453637-3bbc-4777-8a57-f8e65df2b94c"
  }
}

```

Output

```

{ "runStatus": { "ClusterId": "adfcustomerproject1.AutoResolveIntegrationRuntime.5", "sparkVersion": "3.4", "computeAcquisitionDuration": 1934, "version": "20250711.2", "profile": { "customersourcecsvfile": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 7, "total": 7, "updated": 0 }, "window1": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 1, "total": 8, "updated": 0 }, "filtercustomerprimarykeynullscheck": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 8, "updated": 0 }, "sink1": { "computed": [ { "source": "customersourcecsvfile", "columns": [ "customer_id" ] } ], "lineage": { "first_name": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "first_name" ] } ] }, "city": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "city" ] } ] }, "customer_id": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "customer_id" ] } ] }, "zip": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "zip" ] } ] }, "state": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "state" ] } ] }, "dup_primary_key": { "mapped": false, "from": [ { "source": "window1", "columns": [ "dup_primary_key" ] } ] }, "last_name": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "last_name" ] } ] }, "address": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "address" ] } ] }, "dropped": 0, "drifted": 0, "newer": 0, "total": 8, "updated": 8 } }, "metrics": { "sink1": { "format": "", "stages": [ { "stage": 1, "partitionTimes": [ 546 ], "recordsWritten": 87, "lastUpdateTime": "2025-08-11 17:40:53.975", "bytesWritten": 0, "recordsRead": 87, "bytesRead": 4603, "partitionStatus": "Success", "streams": { "customersourcecsvfile": { "count": 87, "cached": false, "totalPartitions": 1, "partitionStatus": "Success", "partitionCounts": [ 87 ], "type": "source" } }, "target": "sink1", "time": 918, "progressState": "Completed" }, { "stage": 3, "partitionTimes": [ 546 ], "recordsWritten": 87, "lastUpdateTime": "2025-08-11 17:40:53.975", "bytesWritten": 0, "recordsRead": 87, "bytesRead": 4603, "partitionStatus": "Success", "streams": { "customersourcecsvfile": { "count": 87, "cached": false, "totalPartitions": 1, "partitionStatus": "Success", "partitionCounts": [ 87 ], "type": "source" } }, "target": "sink1", "time": 918, "progressState": "Completed" } ] } }

```

```
[ 4938 ], "recordsWritten": 0, "lastUpdateTime": "2025-08-11 17:41:00.807",  

"bytesWritten": 944, "recordsRead": 87, "bytesRead": 0, "partitionStatus": "Success",  

"streams": { "window1": { "count": 87, "cached": false, "totalPartitions": 1,  

"partitionStatus": "Success", "partitionCounts": [ 87 ], "type": "window" } }, "target":  

"sink1", "time": 6425, "progressState": "Completed" } ], "sinkPostProcessingTime": 0,  

"store": "", "rowsWritten": 0, "details": {}, "progressState": "Completed", "sources":  

{ "customersourcecsvfile": { "rowsRead": 87, "store": "adlsgen2", "details":  

{ "pathResolutionDuration": [ 144 ], "fileReadCount": [ 1 ], "fileLoadDuration": [ 21403 ],  

"fileSystemInitDuration": [ 21853 ], "format": "delimited" } }, "sinkProcessingTime":  

16317 } }, "clusterComputeId": "9a65c141-10d8-42af-85ee-37a409942f8c", "dsl":  

"\nsource() ~> customersourcecsvfile\n\ncustomersourcecsvfile window() ~>  

window1\n\nwindow1 filter() ~>  

filtercustomerprimarykeynullscheck\n\nfiltercustomerprimarykeynullscheck sink() ~>  

sink1" }, "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (Canada  

Central)", "billingReference": { "activityType": "executedataflow", "billableDuration":  

[ { "meterType": "Data Flow", "duration": 0.16293214866666667, "unit": "coreHour",  

"sessionType": "JobCluster" } ] }, "reportLineageToPurview": { "status": "NotReported" } }
```

Output

[Copy to clipboard](#)

```
{
  "runStatus": [
    {
      "ClusterId": "adfcustomerproject1.AutoResolveIntegrationRuntime.5",
      "sparkVersion": "3.4",
      "computeAcquisitionDuration": 1934,
      "version": "20250711.2",
      "profile": {
        "customersourcecsvfile": {
          "computed": [],
          "lineage": {},
          "dropped": 0,
          "drifted": 0,
          "newer": 7,
          "total": 7,
          "updated": 0
        },
        "window1": {
          "computed": [],
          "lineage": {},
          "dropped": 0,
          "drifted": 0,
          "newer": 1,
          "total": 8,
          "updated": 0
        },
        "filtercustomerprimarykeynullscheck": {
          "computed": [],
          "lineage": {},
          "dropped": 0,
          "drifted": 0,
          "newer": 0,
          "total": 8,
          "updated": 0
        },
        "sink1": {
          "computed": [
            {
              "source": "customersourcecsvfile",
              "columns": [
                "customer_id"
              ]
            }
          ],
          "lineage": {
            "first_name": {

```

Microsoft Azure | Data Factory > adfcustomerproject1

All pipeline runs > **pl_dataclean - Activity runs**

Refresh Update pipeline List Gantt

This is a recent debug run. The local pipeline configuration is shown.

Data flow Data flow1

Activity runs

Pipeline run ID: eca90ad9-40be-4822-8dfa-ffba5afaf8bd

All status Showing 1-1 of 1 items

Activity name	Activity status	Activity ...	Run start	Duration	Integration runtime	User proper...	Activity run ID	Log
Data flow1	Succeeded	Data flow	8/11/2025, 1:39:48 PM	2m 9s	AutoResolveIntegrationRuntime (Canada Central)	9a65c141-10d8-42af-85ee-37a409942fbc		

Monitor in Azure Metrics Export to CSV

Microsoft Azure | Upgrade

Home > Storage accounts > adfsigen2storagephar | Containers > silvercontainercleanneddata

Container

Add Directory Upload Refresh Delete Copy Paste Rename Acquire lease Break lease Edit columns

Search blobs by prefix (case-sensitive) Only show active objects

Overview

Diagnose and solve problems Access Control (IAM) Settings

Showing the first 100 items

Name	Last modified	Access tier	Block type	Size	Lease state
..	8/11/2025, 5:45:49 PM			838 B	Available
..	8/11/2025, 5:45:53 PM	Hot (Inferred)	Block blob	838 B	Available
..	8/13/2025, 7:14:08 AM	Hot (Inferred)	Block blob	838 B	Available
..	8/13/2025, 7:14:08 AM	Hot (Inferred)	Block blob	838 B	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	2.01 kB	Available
..	8/13/2025, 7:14:08 AM	Hot (Inferred)	Block blob	2.01 kB	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	1.99 kB	Available
..	8/13/2025, 7:14:08 AM	Hot (Inferred)	Block blob	1.99 kB	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	2 kB	Available
..	8/13/2025, 7:14:08 AM	Hot (Inferred)	Block blob	2 kB	Available
..	8/13/2025, 7:14:08 AM	Hot (Inferred)	Block blob	2.02 kB	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	2.02 kB	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	1.98 kB	Available
..	8/13/2025, 7:14:09 AM	Hot (Inferred)	Block blob	1.98 kB	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	1.99 kB	Available
..	8/13/2025, 7:14:09 AM	Hot (Inferred)	Block blob	1.99 kB	Available
..	8/11/2025, 5:45:50 PM	Hot (Inferred)	Block blob	1.99 kB	Available

Accounts csv file cleaning

For Accounts addition to checking nulls on primary key of account_id, also checking orphan records and separating from them.

To find the primary key in account csv file open it in excel and check for duplicates for each column and see which column don't have duplicates then consider that column as primarykey and do null check, etc based primary key.

And also we need customer file available as source along with Accounts file as accounts are depending on the customers data

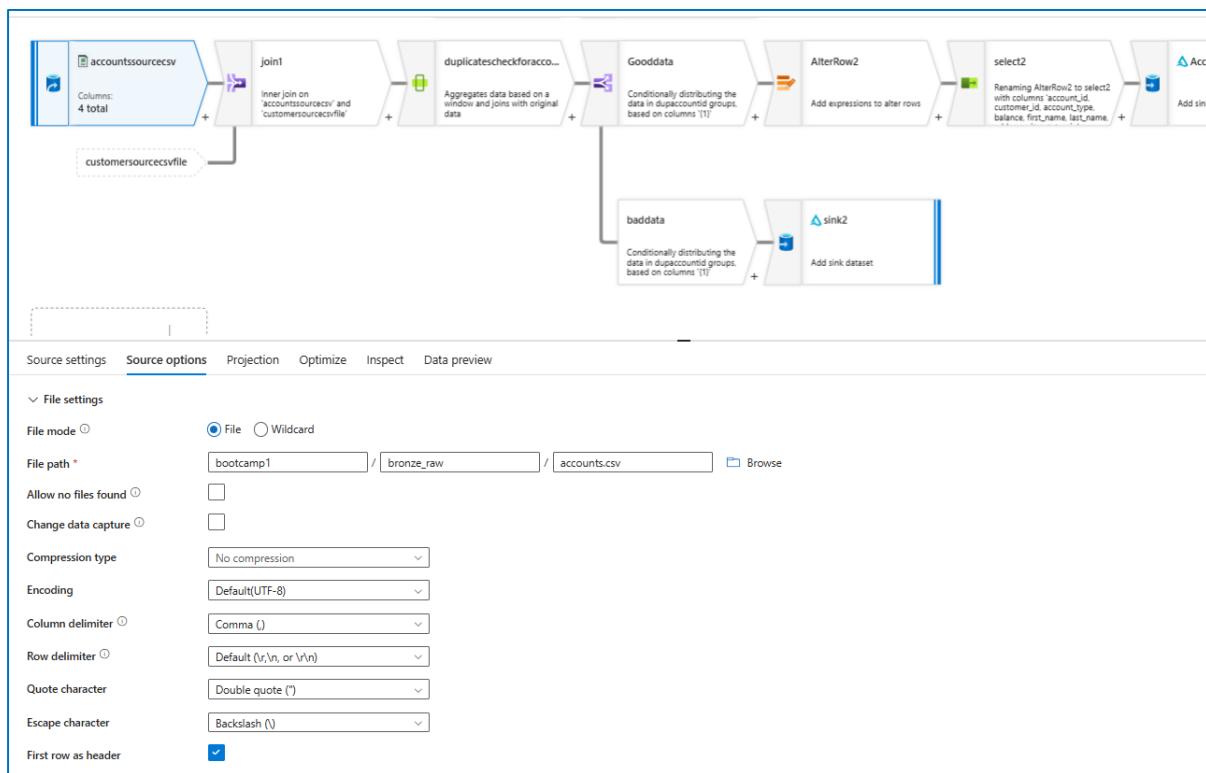
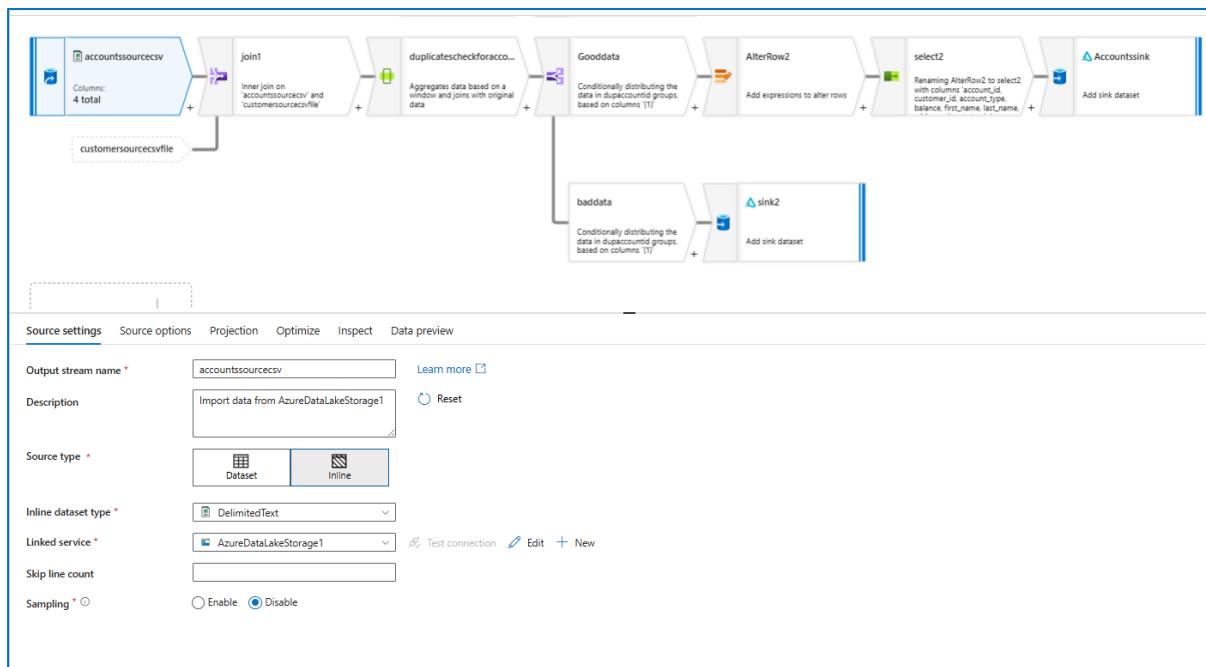
Here I found account_id has no duplicates so I'm considering it as my primary_key

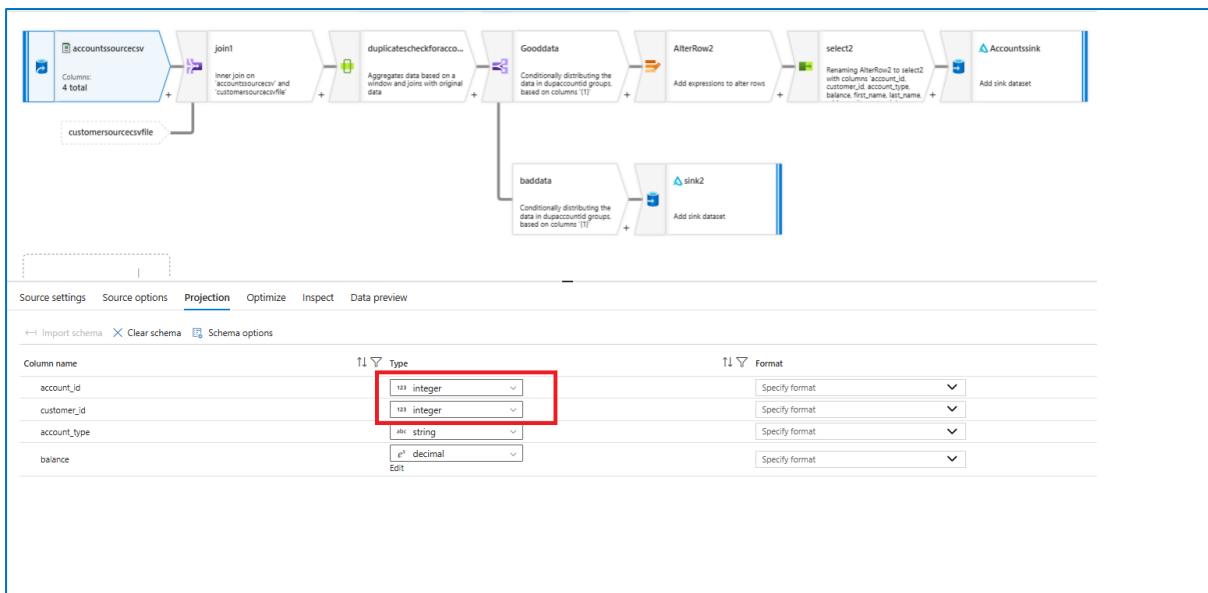
Note: Also I'm changing accountid and customer id datatypes to integer

The screenshot shows a Microsoft Excel spreadsheet titled "Book 2". The table has columns labeled A through N. The first few rows of data are:

account_id	customer_id	account_type	balance
1	1	45 Savings	1000.5
2	2	12 Checking	2500.75
3	3	78 Savings	1500
4	4	34 Checking	3000.25
5	5	56 Savings	500
6	6	23 Checking	1200.5
7	7	89 Savings	800.75
8	8	67 Checking	2200
9	9	14 Savings	900.25
10	10	92 Checking	1800.5
11	11	3 Savings	1100.75
12	12	81 Checking	2700
13	13	29 Savings	1300.25
14	14	64 Checking	3200.5
15	15	47 Savings	700.75
16	16	18 Checking	1400
17	17	99 Savings	600.25
18	18	5 Checking	1600.5
19	19	76 Savings	400.75
20	20	21 Checking	2000
21	21	53 Savings	300.25
22	22	37 Checking	2400.5
23	23	88 Savings	200.75
24	24	11 Checking	2600
25	25	66 Savings	100.25
26	26	25 Checking	2800.5
27	27	94 Savings	50.75
28	28	7 Checking	2900
29	29	58 Savings	75.25
30	30	32 Checking	3100.5
31	31	71 Savings	125.75
32	32	9 Checking	3300
33	33	85 Savings	150.25
34	34	41 Checking	3500.5

Conditional formatting is applied to the "account_type" column (Column C). The dropdown menu for Conditional Formatting is open, showing options like "Greater Than...", "Less Than...", "Between...", "Equal To...", "Text That Contains...", "A Date Occurring...", and "Duplicate Values...".

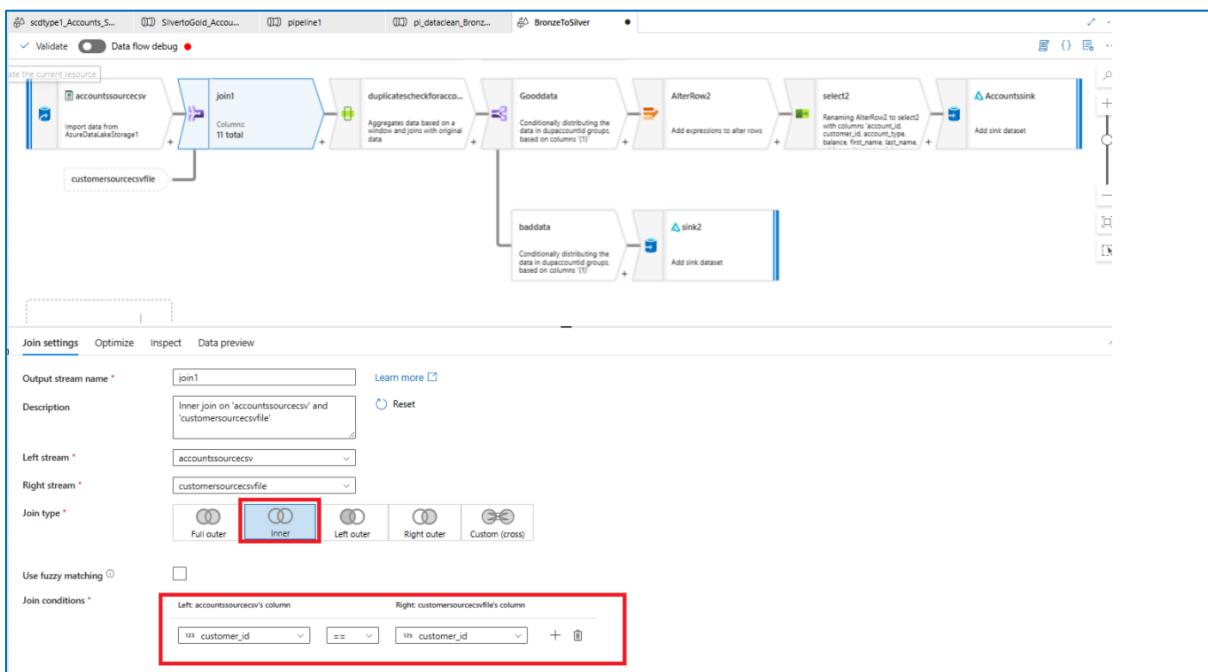




Join to combine customers and accounts data to select matched records which have relationship

Take a join to compare the customer id in both tables and the [inner join](#) will give only the accountid's of customers with parent records.

Now take join and give right stream as customersourcefile and left is accounts table and do inner join so that whatever account id containing customer id matching those records will be carry forwarded and remaining records which dont have any relation they will be send to another file.



Here i got only only 87 records as 87 customers exists

Join settings Optimize Inspect **Data preview** ● Previous

Number of rows: **INSERT 88** **UPDATE 0** **DELETE 0** **UPSERT 0** **LOOKUP 0** **ERROR 0**

Refresh Typecast Modify Map drifted Statistics Remove Export to CSV

account_id	customer_id	customer_id	account_type	balance	first_name	last_name	address	city	state	zip
85	65	65	Savings	800.25	Daniel	Bryant	6464 Redwood Dr	Elmvale	ON	L0L
25	66	66	Savings	100.25	Sophia	Alexander	6565 Cypress Ave	Midland	ON	L4R
8	67	67	Checking	2200.0	Matthew	Russell	6666 Willow Rd	Penetanguishene	ON	L9M
45	68	68	Savings	300.25	Charlotte	GriFFIN	6767 Poplar St	Victoria Harbour	ON	LOK
65	69	69	Savings	550.25	Joseph	Diaz	6868 Ash Blvd	Port McNicoll	ON	LOK
81	70	70	Savings	750.25	Amelia	Hayes	6969 Beech Dr	Wubaushene	ON	LOK
31	71	71	Savings	125.75	Christopher	Myers	7070 Cedar Ln	Coldwater	ON	L0K
51	72	72	Savings	375.75	Mia	Ford	7171 Elm St	Orillia	ON	L3V
71	73	73	Savings	625.75	Andrew	Hamilton	7272 Maple Ave	Gravenhurst	ON	P1P
39	74	74	Savings	225.75	Harper	Graham	7373 Oak Dr	Bala	ON	P0C
59	75	75	Savings	475.75	Joshua	Sullivan	7474 Pine Rd	Bracebridge	ON	P1L
19	76	76	Savings	400.75	Evelyn	Wallace	7575 Birch Blvd	Huntsville	ON	P1H
91	77	77	Savings	875.75	Daniel	Woods	7676 Spruce Ln	Burks Falls	ON	P0A
3	78	78	Savings	1500.0	Abigail	Cole	7777 Fir St	Sundridge	ON	P0A
93	79	79	Savings	900.25	James	West	7878 Redwood Dr	South River	ON	P0A
99	80	80	Savings	975.75	Emily	Jordan	7979 Cypress Ave	North Bay	ON	P1B
12	81	81	Checking	2700.0	Michael	Owens	8080 Willow Rd	Mattawa	ON	P0H
83	82	82	Savings	775.75	Elizabeth	Reynolds	8181 Polar St	Sturgeon Falls	ON	P2B
43	83	83	Savings	275.75	David	Fisher	8282 Ash Blvd	Verner	ON	P0H
63	84	84	Savings	525.75	Sophia	Ellis	8383 Beech Dr	Field	ON	P0H
33	85	85	Savings	150.25	John	Harrison	8484 Cedar Ln	Temagami	ON	P0H
53	86	86	Savings	400.25	Olivia	Gibson	8585 Elm St	New Liskeard	ON	P0H
73	87	87	Savings	650.25	William	McDonald	8686 Maple Ave	Haileybury	ON	N0L

Preview experience: Off

dataflow1

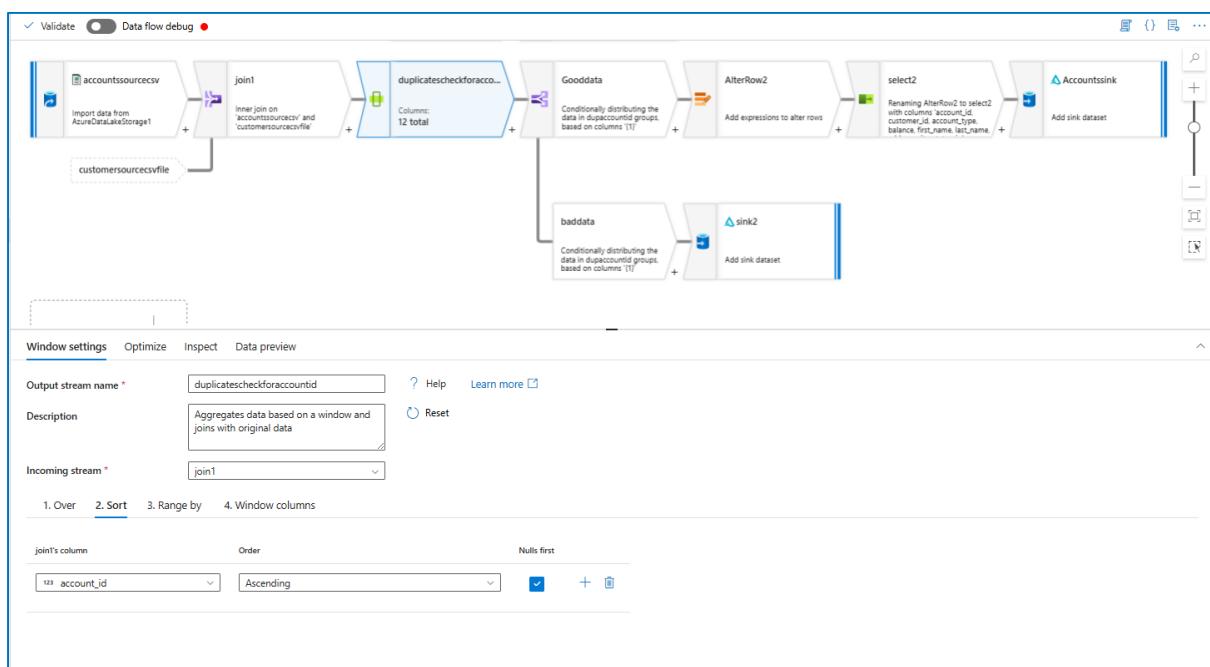
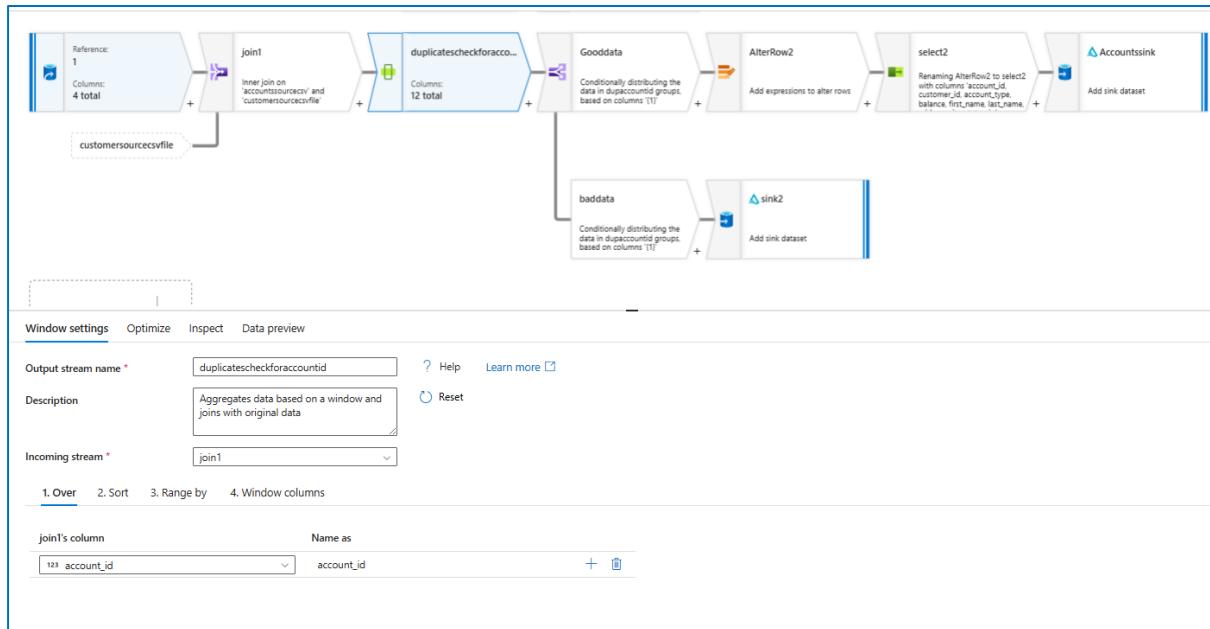
Validate Data flow debug Debug Settings

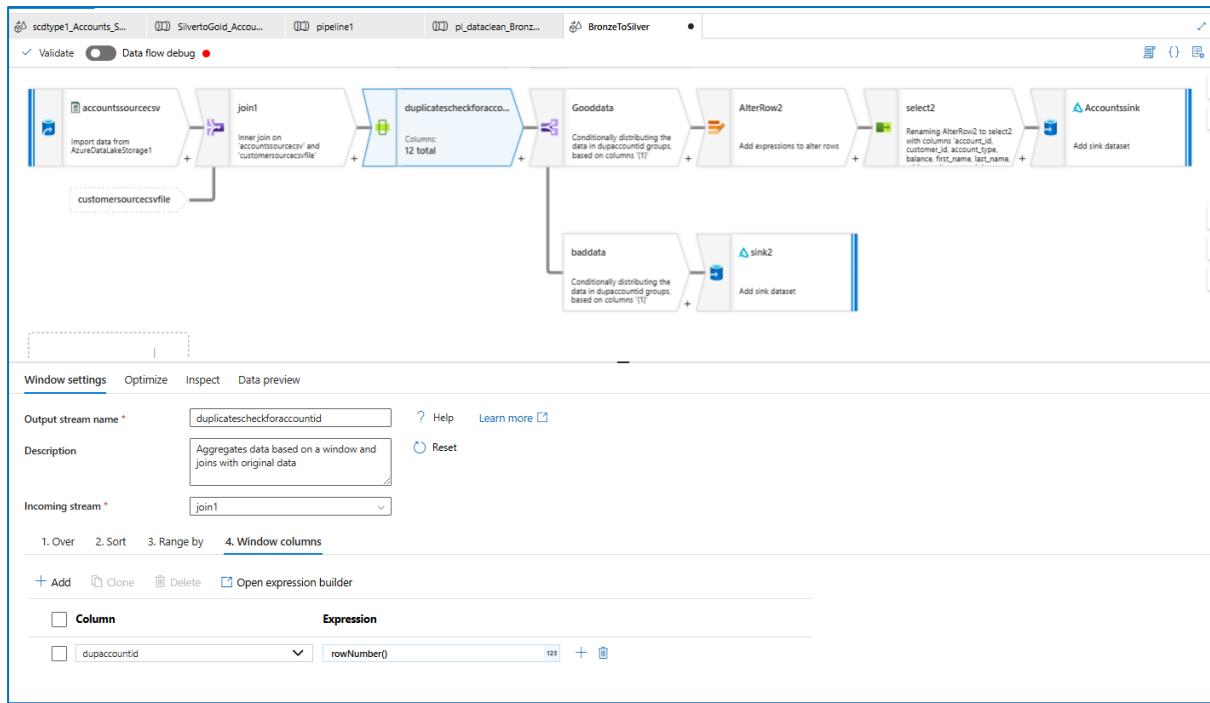
Join settings Optimize Inspect **Data preview** ● Previous Next TOTAL 88

Refresh Typecast Modify Map drifted Statistics Remove Export to CSV

account_id	customer_id	customer_id	account_type	balance	first_name	last_name	address	city	state	zip
88	1	1	Checking	8900.0	John	Doe	123 Elm St	Toronto	ON	M4B1B3
82	2	2	Checking	8300.5	Jane	Smith	456 Maple Ave	Ottawa	ON	K1A0B1
11	3	3	Savings	1100.75	Michael	Johnson	789 Oak Dr	Montreal	QC	H1A1A1
78	4	4	Checking	7900.5	Emily	Davis	101 Pine Rd	Gatineau	AB	T2A0A1
18	5	5	Checking	1600.5	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1
48	6	6	Checking	4900.0	Emma	Clerk	505 Cedar St	Halifax	NS	B3H0A1
28	7	7	Checking	2900.0	James	Martinez	606 Spruce Ln	Winnipeg	MB	R3C0A1
68	8	8	Checking	6900.0	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1
32	9	9	Checking	3300.0	William	Lopez	808 Redwood Dr	Victoria	BC	V8W0A1
52	10	10	Checking	5300.0	Ava	Anderson	909 Cypress Ave	Quebec City	QC	G1A0A1
24	11	11	Checking	2600.0	Alexander	Thomas	1010 Willow Rd	St. John's	NL	A1A0A1
64	12	12	Checking	6500.0	Isabella	Lee	1111 Poplar St	Fredericton	NB	E3B0A1
2	12	12	Checking	2500.75	Isabella	Lee	1111 Poplar St	Fredericton	NB	E3B0A1
44	13	13	Checking	4500.0	Daniel	Harris	1212 Ash Blvd	Charlottetown	PE	C1A0A1
9	14	14	Savings	900.25	Sophia	Young	1313 Beech Dr	Yellowknife	NT	X1A0A1
38	15	15	Checking	3900.5	Matthew	King	1414 Cedar Ln	Whitehorse	YT	Y1A0A1
58	16	16	Checking	5900.5	Charlotte	Scott	1515 Elm St	Iqaluit	NU	X0A0A1
72	17	17	Checking	7300.0	Joseph	Green	1616 Maple Ave	Regina	SK	S4P0A1
16	18	18	Checking	1400.0	Amelia	Adams	1717 Oak Dr	Saskatoon	SK	S7K0A1
40	19	19	Checking	4100.0	Christopher	Baker	1818 Pine Rd	Thunder Bay	ON	P7A0A1
60	20	20	Checking	6100.0	Mia	Nelson	1919 Birch Blvd	London	ON	N6A0A1
86	21	21	Checking	8700.5	Andrew	Mitchell	2020 Spruce Ln	Hamilton	ON	L8P0A1
20	21	21	Checking	2000.0	Andrew	Mitchell	2020 Spruce Ln	Hamilton	ON	L8P0A1

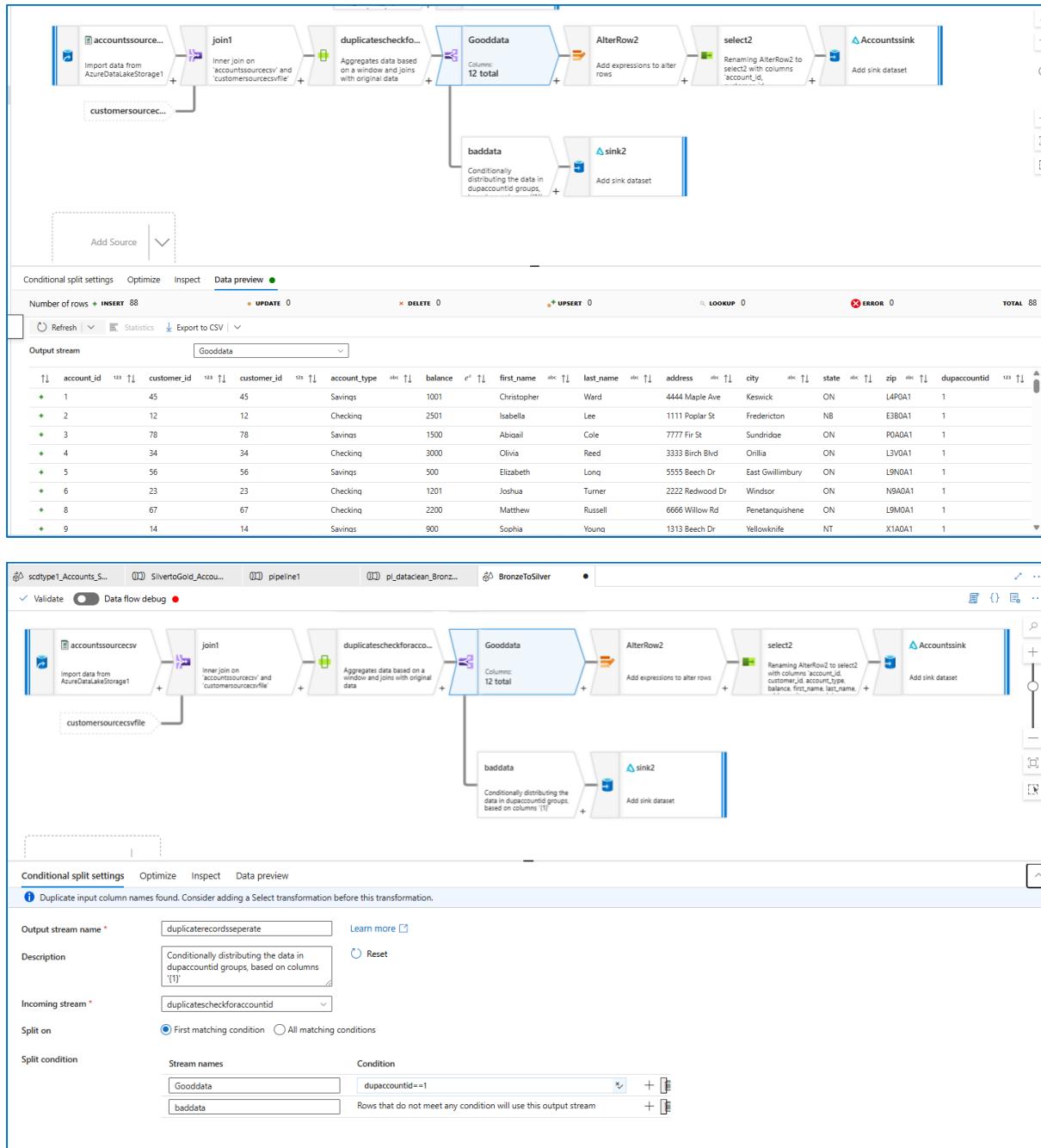
Window function to remove duplicate keys in accounts



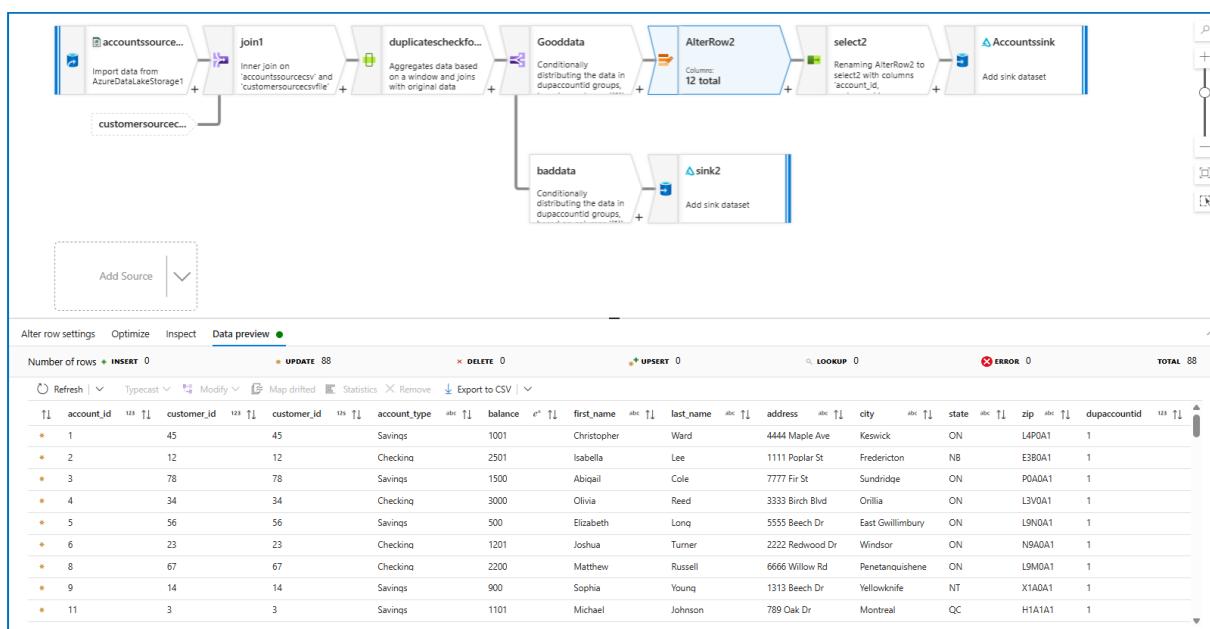
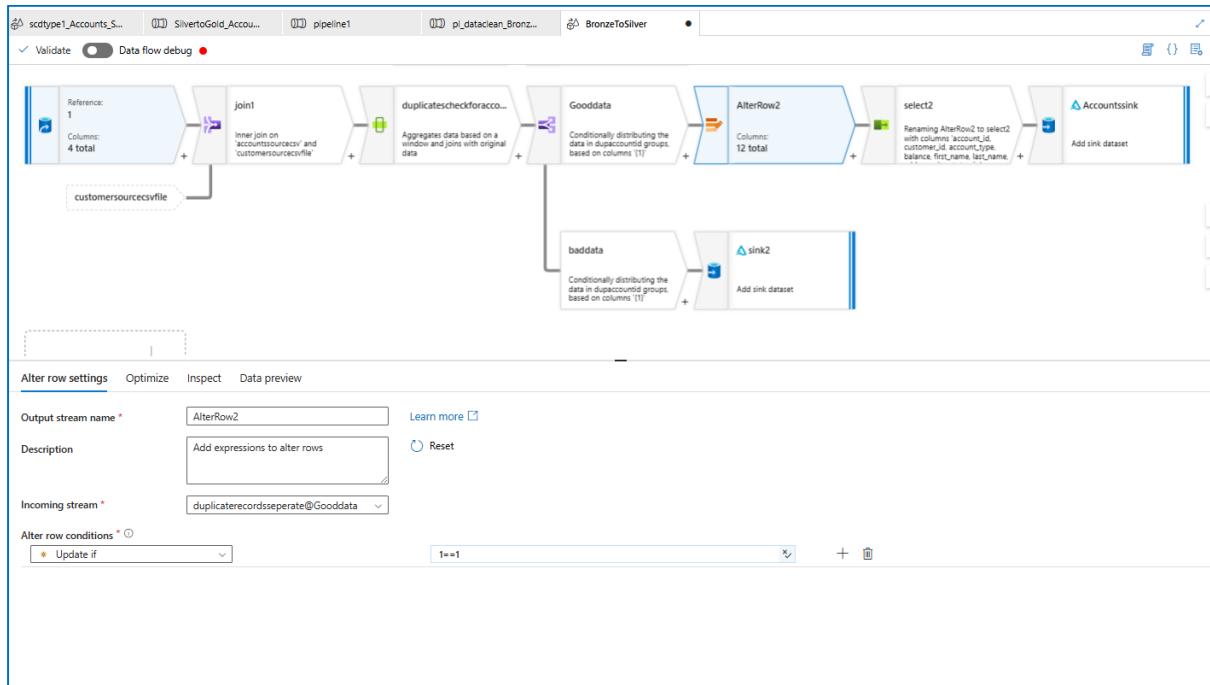


account_id	customer_id	customer_id	customer_id	account_type	balance	first_name	last_name	address	city	state	zip
88	1	1		Checking	8900	John	Doe	123 Elm St	Toronto	ON	M4B1B3
82	2	2		Checking	8301	Jane	Smith	456 Maple Ave	Ottawa	ON	K1A0B1
11	3	3		Savings	1101	Michael	Johnson	789 Oak Dr	Montreal	QC	H1A1A1
78	4	4		Checking	7901	Emily	Davis	101 Pine Rd	Calgary	AB	T2A0A1
18	5	5		Checking	1601	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1
48	6	6		Checking	4900	Emma	Clark	505 Cedar St	Halifax	NS	B3H0A1
28	7	7		Checking	2900	James	Martinez	606 Spruce Ln	Winnipeg	MB	R3C0A1
68	8	8		Checking	6900	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1
32	9	9		Checking	3300	William	Lopez	808 Redwood Dr	Victoria	BC	V8W0A1

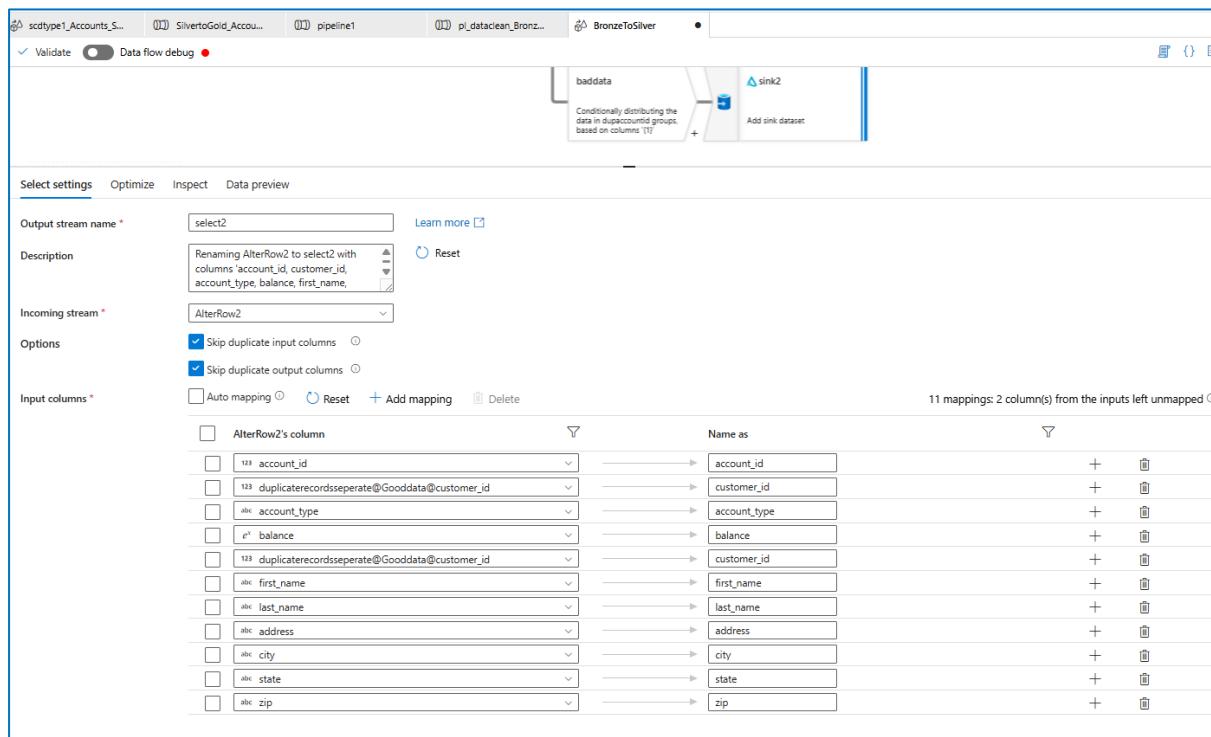
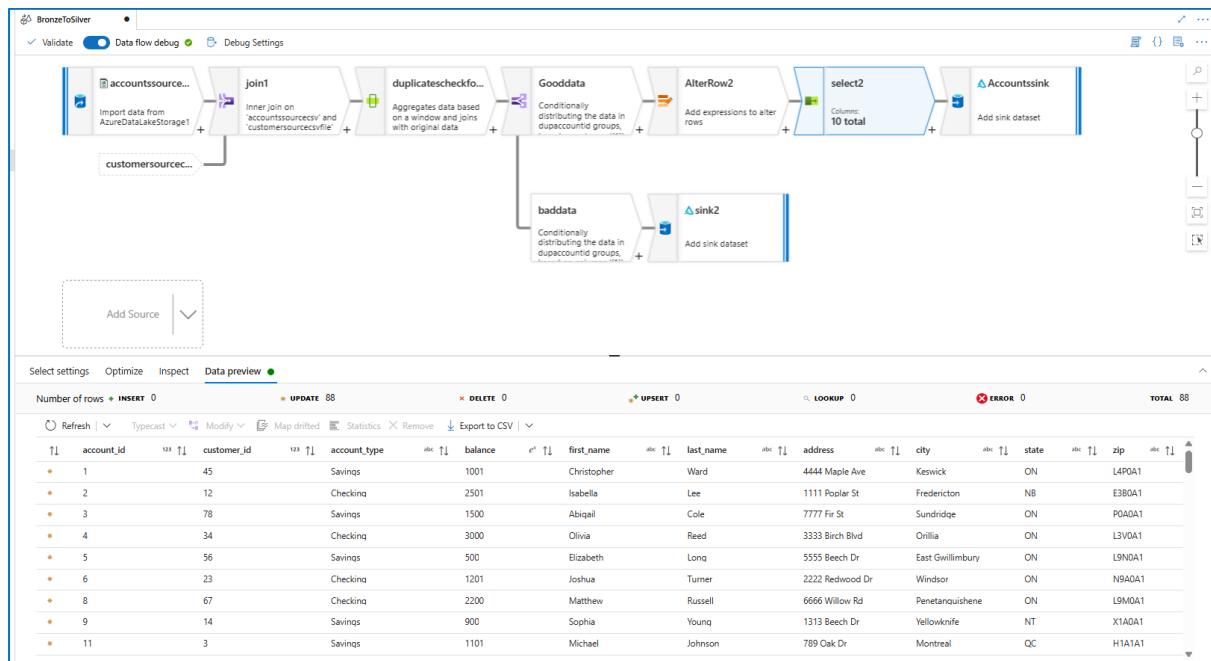
Conditional Split to separate duplicate records and keys without duplicates

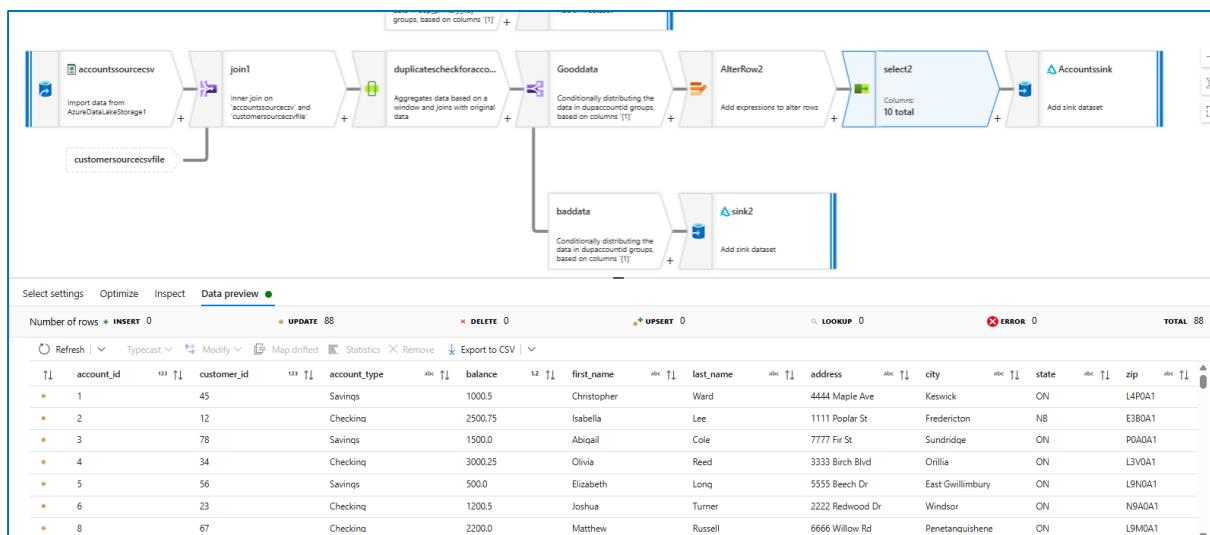


AlterRow to deside the rows to be inserted or updated

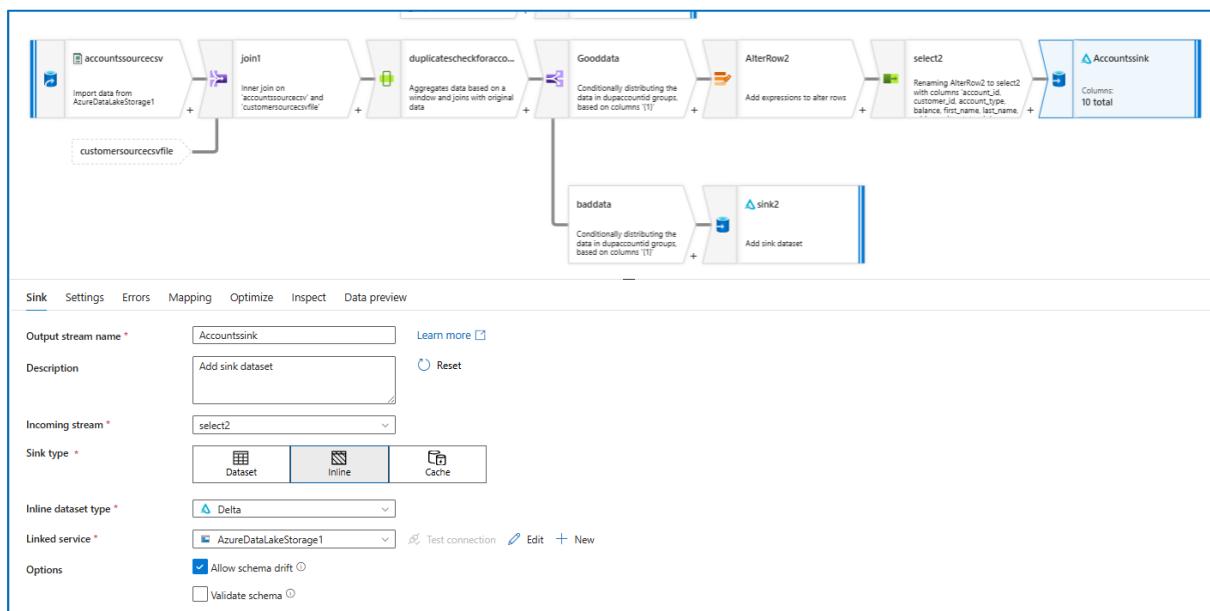


Select function to remove unnecessary columns





Sink to add the good records



Run the pipeline

The screenshot shows the Azure Data Flow pipeline run details. At the top, there's a toolbar with icons for search, add, delete, copy, and others. Below it is a summary card for 'Data flow' named 'Data flow1'. The main area has tabs for 'Parameters', 'Variables', 'Settings', and 'Output', with 'Output' being the active tab. It displays the pipeline run ID (88f4b019-f813-49dc-bd58-db0cf1c6052e), pipeline status (Succeeded), and a table of activity runs. The table includes columns for Activity name, Activity st..., Activit..., Run start, Duration, Integration runtime, User prop..., and Activity run ID. One row is shown: 'Data flow1' succeeded at 8/11/2025, 3:55:08 PM, with a duration of 1m 37s and an integration runtime of AutoResolveIntegrationRuntime (Canada Central). A link to 'View debug run consumption' is also present.

Input

{

```
"dataflow":{  
    "referenceName": "dataflow1",  
    "type": "DataFlowReference",  
    "parameters": {},  
    "datasetParameters": {  
        "customersourcecsvfile": {},  
        "accountssourcecsv": {},  
        "Customerssink": {},  
        "Accountssink": {}  
    },  
    "staging": {},  
    "compute": {  
        "coreCount": 8,  
        "computeType": "General"  
    },  
    "traceLevel": "Fine",  
    "dataFlowDebugSessionId": "c3f85ac3-bd35-4ee7-840e-02cf70161e32",  
    "continuationSettings": {  
        "customizedCheckpointKey": "pl_dataclean-Data flow1-fa7f7fcf-9548-4893-  
8d0c-8a70202f5582"  
    }  
}
```

Output:

```
{ "runStatus": { "ClusterId": "adfcustomerproject1.AutoResolveIntegrationRuntime.8", "sparkVersion": "3.4", "computeAcquisitionDuration": 941, "version": "20250711.2", "profile": { "window1": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 1, "total": 8, "updated": 0 }, "split1@Gooddata1": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 8, "updated": 0 }, "duplicatescheckforaccountid": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 1, "total": 5, "updated": 0 }, "accountssourcecsv": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 4, "total": 4, "updated": 0 }, "select1": { "computed": [], "lineage": {}, "dropped": 1, "drifted": 0, "newer": 0, "total": 7, "updated": 7 }, "alterRow1": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 8, "updated": 0 }, "split2@Gooddata": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 5, "updated": 0 }, "split1@baddata1": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 8, "updated": 0 }, "Customerssink": { "computed": [ { "source": "window1", "columns": [ "dup_primary_key" ] } ], "lineage": { "first_name": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "first_name" ] } ] }, "city": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "city" ] } ] }, "customer_id": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "customer_id" ] } ] }, "zip": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "zip" ] } ] }, "state": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "state" ] } ] }, "last_name": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "last_name" ] } ] }, "address": { "mapped": true, "from": [ { "source": "customersourcecsvfile", "columns": [ "address" ] } ] }, "dropped": 0, "drifted": 0, "newer": 0, "total": 7, "updated": 7 }, "split2@baddata": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 5, "updated": 0 }, "AlterRow2": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 5, "updated": 0 }, "select2": { "computed": [], "lineage": {}, "dropped": 1, "drifted": 0, "newer": 0, "total": 4, "updated": 4 }, "Accountssink": { "computed": [ { "source": "duplicatescheckforaccountid", "columns": [ "dupaccountid" ] }, { "source": "window1", "columns": [ "dup_primary_key" ] } ], "lineage": { "account_id": { "mapped": true, "from": [ { "source": "accountssourcecsv", "columns": [ "account_id" ] } ] }, "customer_id": { "mapped": true, "from": [ { "source": "accountssourcecsv", "columns": [ "customer_id" ] } ] }, "account_type": { "mapped": true, "from": [ { "source": "accountssourcecsv", "columns": [ "account_type" ] } ] }, "balance": { "mapped": true, "from": [ { "source": "accountssourcecsv", "columns": [ "balance" ] } ] }, "dropped": 0, "drifted": 0, "newer": 0, "total": 4, "updated": 4 }, "customersourcecsvfile": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 7, "total": 7, "updated": 0 }, "metrics": { "Accountssink": { "format": "delta", "stages": [ { "stage": 106, "partitionTimes": [ 66 ], "recordsWritten": 100, "lastUpdateTime": "2025-08-11 21:45:22.762", "bytesWritten": 0, "recordsRead": 100, "bytesRead": 2331, "partitionStatus": "Success", "streams": { "accountssourcecsv": { "count": 100, "cached": false, "totalPartitions": 1, "partitionStatus": "Success", "partitionCounts": [ 100 ], "type": "source" } }, "target": "Accountssink", "time": 85 } ] } }
```



```

1, 0, 2, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 1, 0, 2, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 2,
0, 1, 0, 1, 0, 1, 1, 0, 1, 2, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 1, 2, 1, 2, 0, 0, 1, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 1, 0, 1, 2, 0, 0, 0, 0, 0, 1, 1, 0,
0, 0, 0, 1 ], "type": "alterRow" }, "Customerssink": { "count": 87, "cached": false,
"totalPartitions": 199, "partitionStatus": "Success", "partitionCounts": [ 0, 0, 1, 1, 1, 1,
0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 2, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 1, 1, 0, 0, 0, 1, 3, 0, 1, 0, 2, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 2,
0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 1, 0, 2, 1, 1, 0, 1, 0, 1, 0, 1,
0, 1, 0, 0, 0, 1, 0, 0, 2, 0, 1, 0, 1, 1, 0, 1, 2, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 2, 1, 2, 0, 0, 1, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 1, 0, 1, 2, 0,
0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1 ], "type": "sink" } }, "target": "Customerssink", "time": 6039, "progressState": "Completed" } ], "sinkPostProcessingTime": 0, "store": "adlsgen2", "rowsWritten": 87, "details": {}, "progressState": "Completed", "sources": { "customersourcecsvfile": { "rowsRead": 87, "store": "adlsgen2", "details": { "pathResolutionDuration": [ 23 ], "fileReadCount": [ 1 ], "fileLoadDuration": [ 12197 ], "fileSystemInitDuration": [ 12236 ] }, "format": "delimited" } }, "sinkProcessingTime": 14777 }, "clusterComputelid": "23f66f2e-6cd2-4132-ad55-e1f49cbb5c6a", "dsl": "\nsource() ~> accountssourcecsv\n\naccountssourcecsv window() ~>\nduplicatescheckforaccountid\n\nDuplicatesCheckForAccountID split() ~>\nsplit2@(Gooddata, baddata)\n\nsplit2@Gooddata alterRow() ~>\nAlterRow2\n\nAlterRow2 select() ~> select2\n\nselect2 sink() ~>\nAccountssink\n\nsource() ~> customersourcecsvfile\n\ncustomersourcecsvfile\nwindow() ~> window1\n\nwindow1 split() ~> split1@(Gooddata1,\nbaddata1)\n\nsplit1@Gooddata1 alterRow() ~> alterRow1\n\nalterRow1 select() ~>\nselect1\n\nselect1 sink() ~> Customerssink" }, "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (Canada Central)", "billingReference": { "activityType": "executedataflow", "billableDuration": [ { "meterType": "Data Flow", "duration": 0.1372946117777777, "unit": "coreHour", "sessionType": "JobCluster" } ] }, "reportLineageToPurview": { "status": "NotReported" } }

```

Storage account saved delta format data of accounts

Microsoft Azure Upgrade

Home > silvercontainercleaneddata Container

Overview

Search Add Directory Upload Refresh Delete Copy Paste Rename Acquire lease Break lease Edit columns

silvercontainercleaneddata > Delta > Accounts

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

Search blobs by prefix (case-sensitive) Only show active objects

Show all 83 items

Name	Last modified	Access tier	Blob type	Size	Lease state
l4	8/11/2023, 5:45:22 PM	Hot (Inferred)	Block blob	587 B	Available
delta_log	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00002-7a79d0b-1c7f-4169-b775-a0f044ae6528-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00003-02288797-fc6f-4c6c-9dc9-f5c1dedf3fb08-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00004-fe113905-79c5-4c12-b55a-cbd0e01c109c-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00005-2a4d80ac-66ee-486b-86c2-23dd98c0296a-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00007-a53b3b3-16c7-4f68-9045-36d4145defa7-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00011-a8080fb3-ffa3-400b-b49c-e6f5146c778a-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00013-86187ba9-7444-4230-8ec0-e41963b4c715-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00014-d76ee099-056e-4880-8774-acf136984a9-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00018-1df4bca-93f8-4cc0-af73-42992c659db8-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00019-2c0d37ca-4096-4dfe-666e-e99cd0d113ba-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00021-329b2c5a-c4d9-4a21-9280-9b781f493b89-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00024-32b8cdcb-4c80-4235-baa5-1f1a3b41476b-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00027-d572f748-1b5f-4893-991a-cf54e4096e51-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available
part-00030-c085d2c6-544c-4b8d-aeef-2711150808a4a-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.24 KB	Available
part-00032-d8b3d414e-5e21-40fe-b37e-3a58d76f19d9-c000.unappy.parquet	8/11/2023, 5:45:23 PM	Hot (Inferred)	Block blob	1.23 KB	Available

Add or remove resources by pressing **Ctrl + Shift + F**

Microsoft Azure Upgrade

Home > silvercontainercleaneddata Container

Overview

Search Add Directory Upload Refresh Delete Copy Paste Rename Acquire lease Break lease Edit columns

silvercontainercleaneddata > Delta > Customers > _delta_log

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

Search blobs by prefix (case-sensitive) Only show active objects

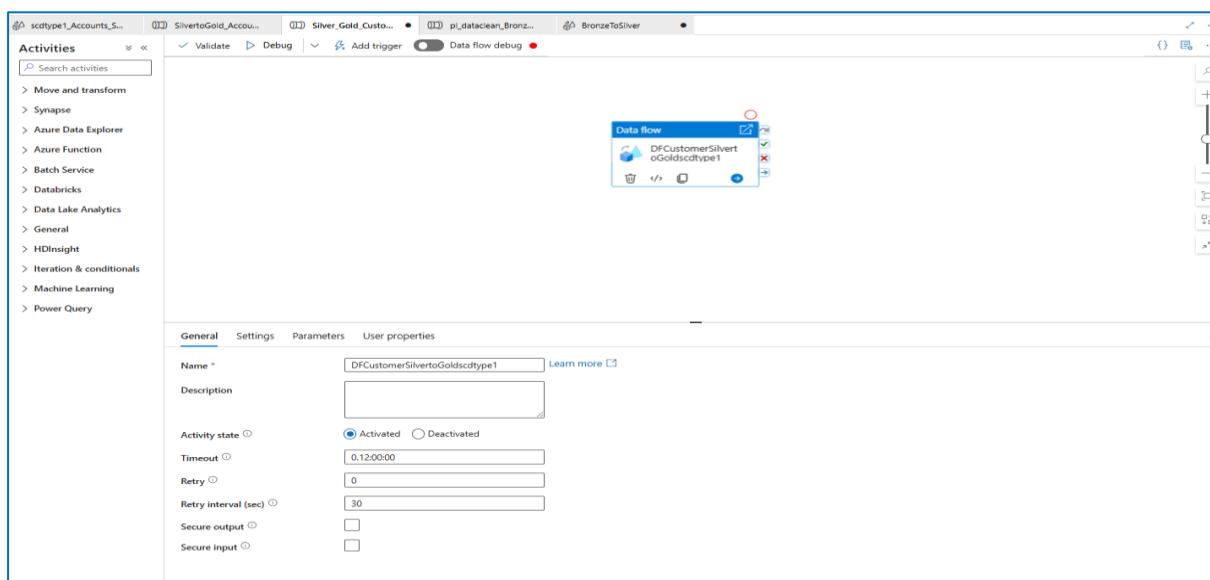
Show all 2 items

Name	Last modified	Access tier	Blob type	Size	Lease state
l4	8/11/2023, 5:45:56 PM	Hot (Inferred)	Block blob	47.89 kB	Available
_temporary	8/11/2023, 5:45:56 PM	Hot (Inferred)	Block blob	47.89 kB	Available

I have done same procedure for remaining 3 files

Silver to Gold

Step3 : Perform SCD Type1 on Customers delta file which is stored to silvercontainer storage account



Go to settings and add dataflow then fill the source details

Source settings

Output stream name *: deltacustomerfile

Description: Add source dataset

Source type *: Dataset

Inline dataset type *: Delta

Linked service *: AzureDataLakeStorage1

Sampling *: Disable

Source as Customers delta file data

Go to source options and select the path to select the delta files

Note: you have to select up to customers, don't select delta log path, you have to select before delta_log.

Note: if you have any temporary folders created please remove it.

Factory Resources

- Pipelines: pl_deltaclean, pipeline1, p_dataclean
- Datasets: dataflow1, dataflow1_copy1, dataflow1_copy2, scotttype1_customers_bootcamp1ipe3
- Data flows: dataflow1, dataflow1_copy1, dataflow1_copy2
- Power Query: 0

pl_deltaclean

Source settings

Folder path: silvercontainercleandata / Delta/Customers

Allow no files found:

Compression type: No compression

Time travel: Disable, Query by timestamp, Query by version

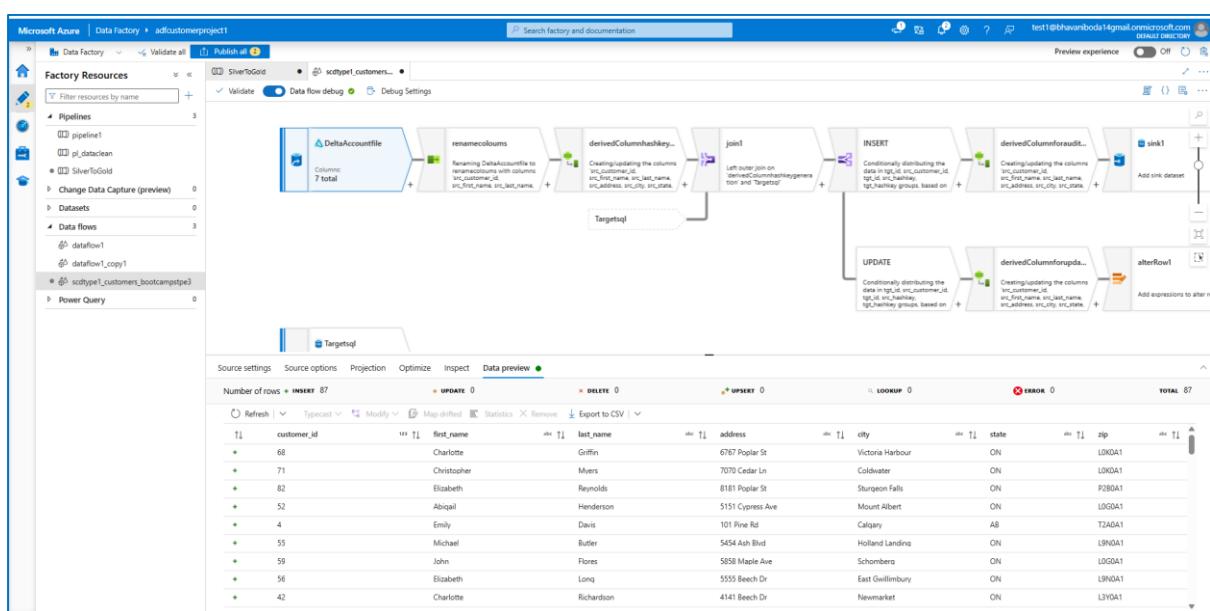
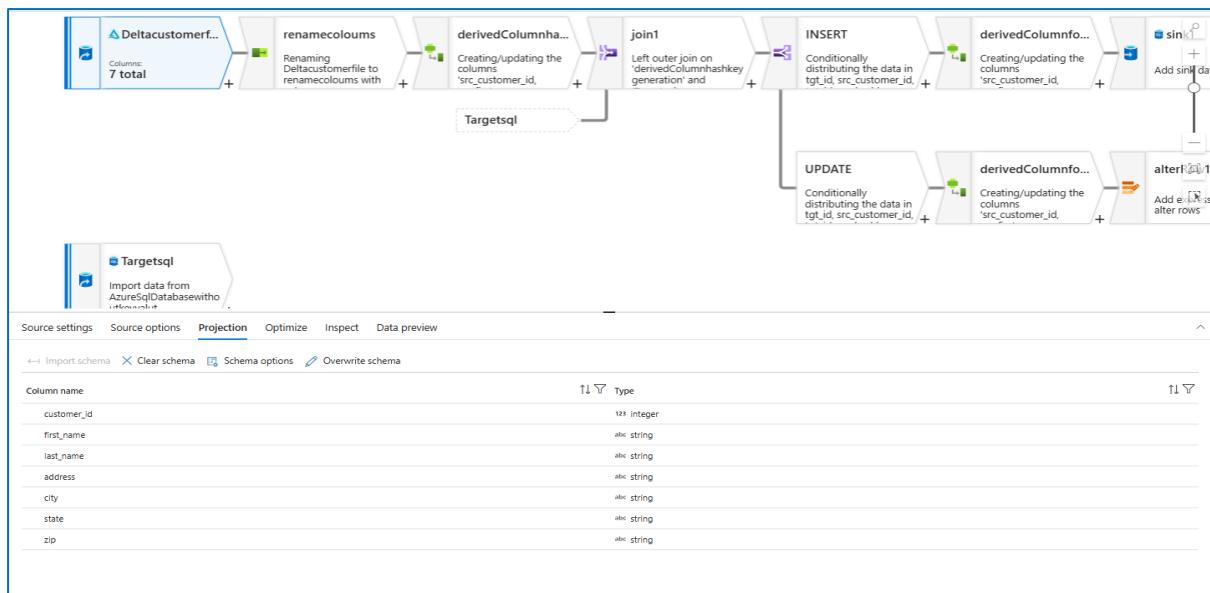
Choose a folder

Root folder > silvercontainercleandata > Delta > Customers

_delta_log

Showing 1 item

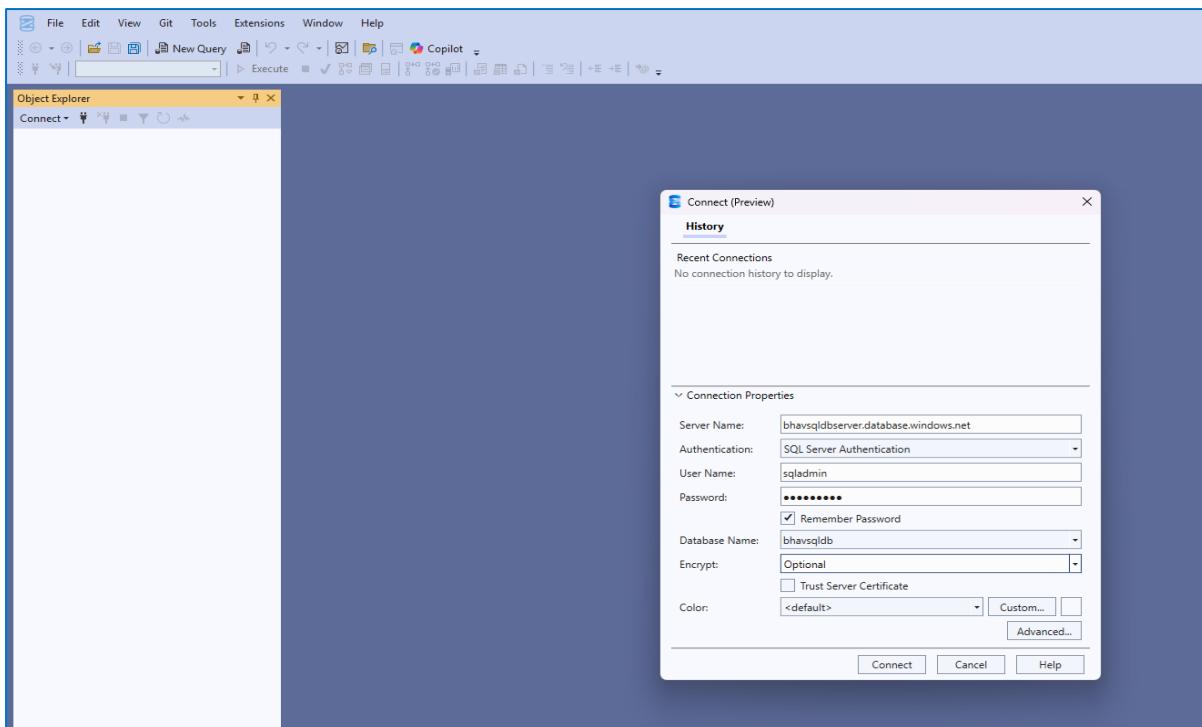
OK Cancel



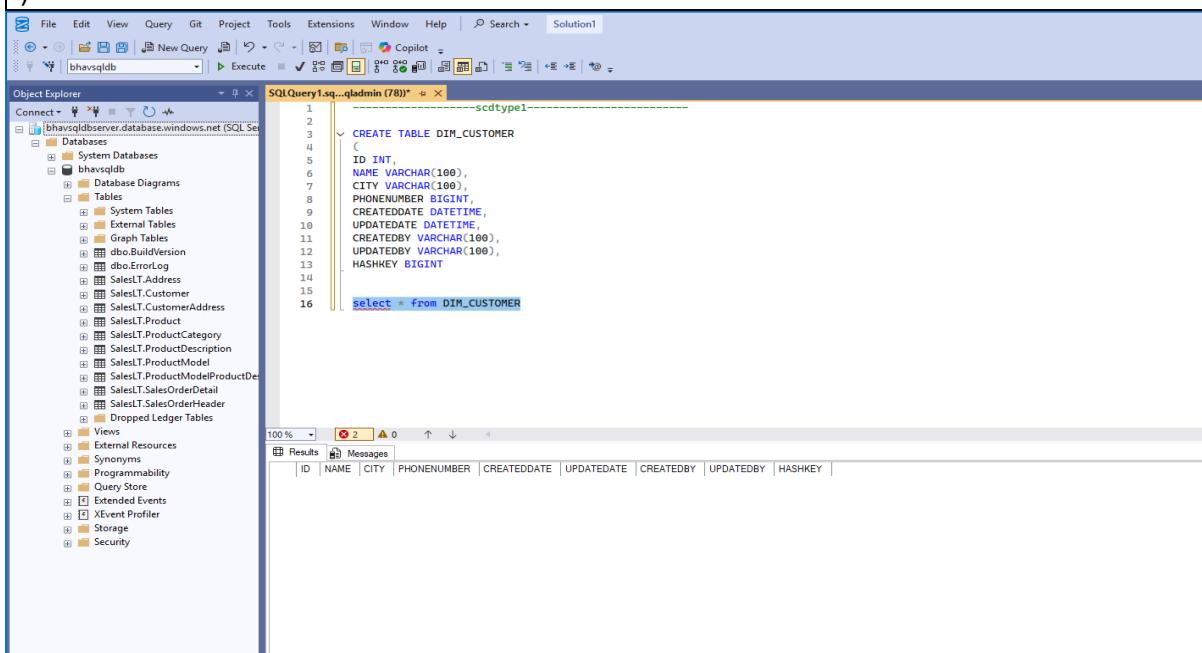
Target as sql, so open ssms and create a table.

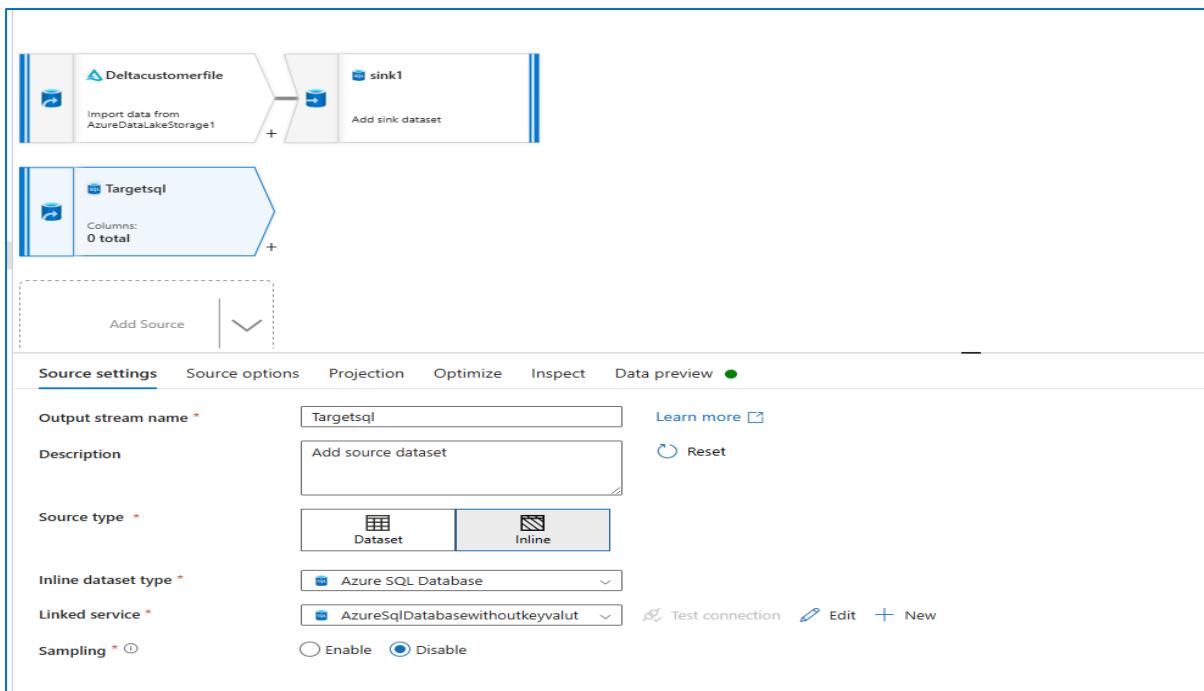
Create Target DIM_Customers table in Azure SQL database(SSMS)

Note: Create columns which should match the customers delta file columns.



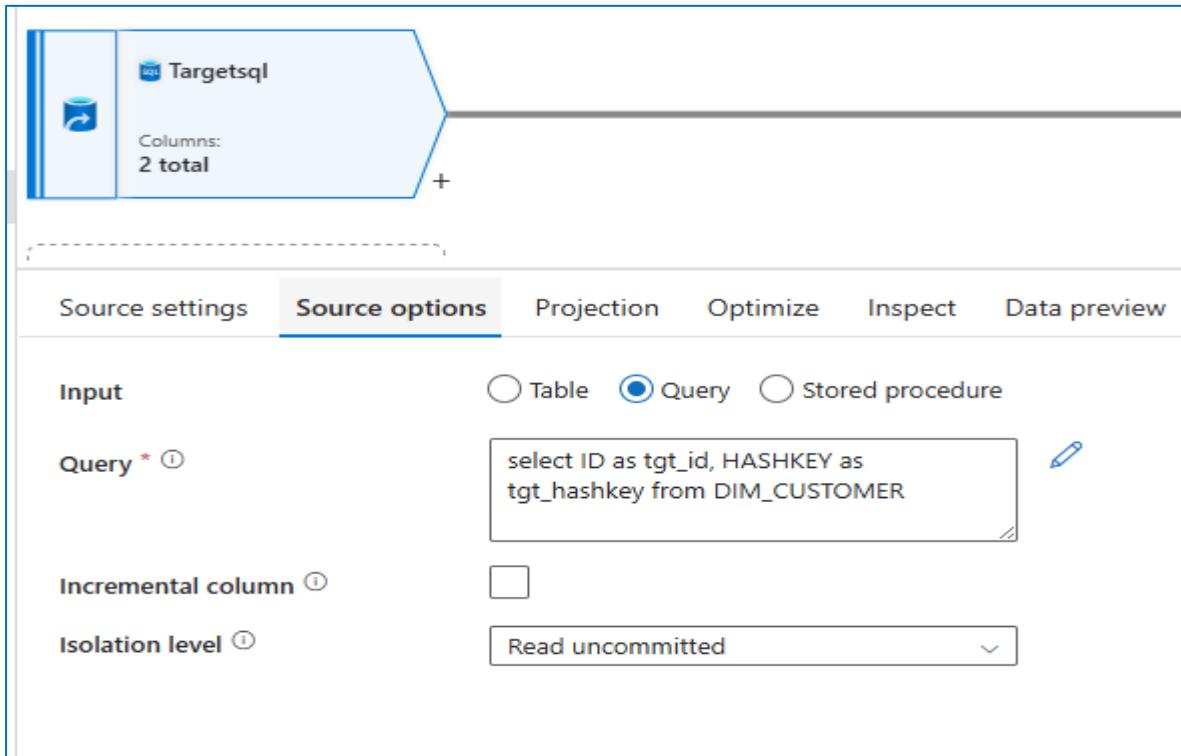
```
CREATE TABLE DIM_CUSTOMER
(
ID INT,
NAME VARCHAR(100),
CITY VARCHAR(100),
CREATEDDATE DATETIME,
UPDATEDATE DATETIME,
CREATEDBY VARCHAR(100),
UPDATEDBY VARCHAR(100),
HASHKEY BIGINT
)
```





Use query to generate ID as tgt_id and HASHKEY as tgt_hashkey

select ID as tgt_id, HASHKEY as tgt_hashkey from DIM_CUSTOMER



The screenshot shows the 'Projection' tab of a data pipeline configuration. At the top, there is a summary box for 'Targetsqli' with 'Columns: 2 total'. Below the tabs, there are buttons for 'Import schema', 'Clear schema', 'Schema options', and 'Overwrite schema'. The main area displays two columns: 'tgt_id' (integer) and 'tgt_hashkey' (long). The 'Type' column is aligned to the right.

Column name	Type
tgt_id	integer
tgt_hashkey	long

The screenshot shows the 'Inspect' tab of a data pipeline configuration. At the top, there is a summary box for 'Targetsqli' with 'Columns: 2 total'. Below the tabs, it shows 'Number of columns: Total 2'. The table lists columns by order (1 and 2), name (tgt_id and tgt_hashkey), and type (integer and long).

Order	Column	Type
1	tgt_id	integer
2	tgt_hashkey	long

Data preview as we dont have any data from target table

The screenshot shows the 'Data preview' tab of a data pipeline configuration. At the top, there is a summary box for 'Targetsqli' with 'Columns: 2 total'. Below the tabs, it shows 'Number of rows: N/A' with buttons for INSERT, UPDATE, DELETE, and UPSERT. The table lists columns by name (tgt_id and tgt_hashkey) and type (integer and long). A message at the bottom states 'No output data.'

tgt_id	tgt_hashkey
--------	-------------

Targetsqli
Columns: 9 total

Add Source

Source settings Source options **Projection** Optimize Inspect Data preview

Import schema Clear schema Schema options Overwrite schema

Column name	Type
ID	123 integer
NAME	abc string
CITY	abc string
PHONENUMBER	12l long
CREATEDDATE	⌚ timestamp
UPDATEDATE	⌚ timestamp
CREATEDBY	abc string
UPDATEDBY	abc string
HASHKEY	12l long

Inspect tab

Targetsqli
Columns: 9 total

Add Source

Source settings Source options Projection Optimize **Inspect** Data preview

Number of columns **Total 9**

Order ↑↓	Column ↑↓	Type ↑↓
1	ID	123 integer
2	NAME	abc string
3	CITY	abc string
4	PHONENUMBER	12l long
5	CREATEDDATE	⌚ timestamp
6	UPDATEDATE	⌚ timestamp
7	CREATEDBY	abc string

It should not have any data as we didnt added anything

The screenshot shows the Data Preview tab in Azure Data Studio. At the top, there's a summary of the target source: "Targetsqli" with "9 total" columns. Below this is a table with the following columns:

ID	NAME	CITY	PHONENUMBER	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
123	abc	abc	abc	123	123	abc	abc	123

No output data.

Select select to rename source file coloumns using (1==1 concat('src_',\$\$)) with src
It will add src to each source column, here \$\$ will select all the colomns to apply src_

The screenshot shows the Data Flow tab in Azure Data Studio. The pipeline consists of several stages:

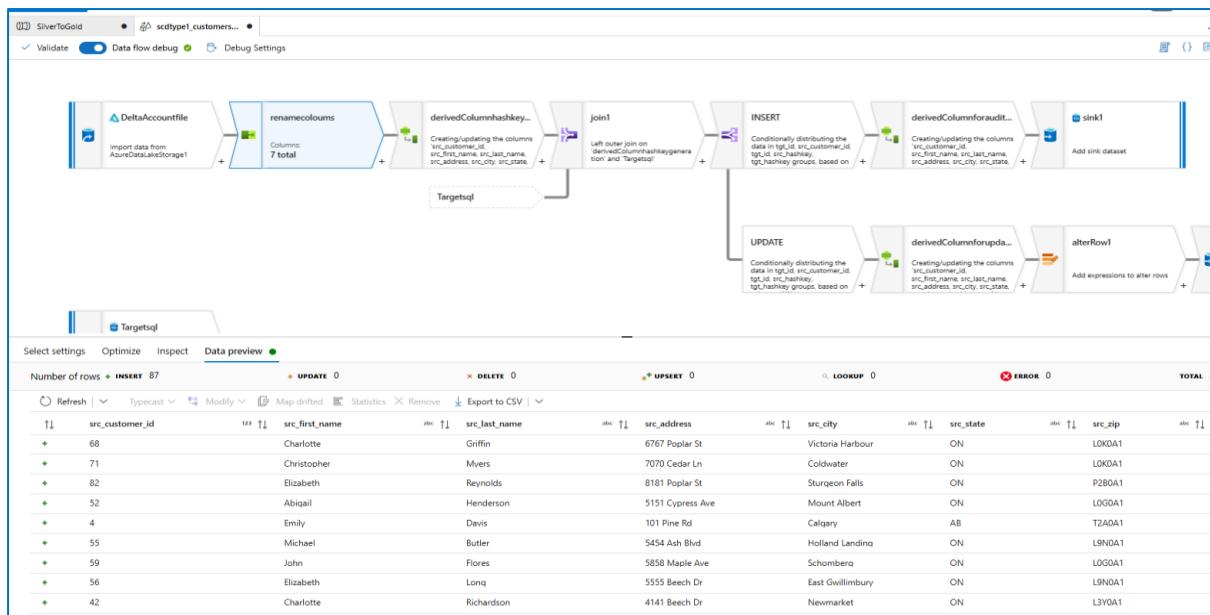
- Import data from AzureDataLakeStorage1**: Imports data from a source.
- renamecolumns**: Renames columns.
- derivedColumnhashkey**: Creates/Updates the column 'src_customer_id'.
- join1**: Left outer join on 'derivedColumnhashkey'.
- INSERT**: Conditionally distributing the data in 'tgt_id', 'src_customer_id'.
- Reference**: A reference stream with 10 total columns.
- derivedColumninfo**: Creating/updating the column 'src_customer_id'.
- sin1**: Adds single data.
- alter1**: Adds alter rows.

The pipeline ends at the **Targetsqli** target.

The screenshot shows the Data Flow tab in Azure Data Studio, specifically the Schema mapping section. It displays a table of updated columns:

Order	Column	Type	Updated	Total
1	src_customer_id	int	0	7
2	src_first_name	string	0	
3	src_last_name	string	0	
4	src_address	string	0	
5	src_city	string	0	
6	src_state	string	0	
7	src_zip	string	0	

Data Preview



Create Table in database as DIM_Accounts

```
16
17
18 CREATE TABLE DIM_Accounts
19 (
20     account_id INT PRIMARY KEY,
21     ID INT,
22     account_type VARCHAR(50),
23     balance DECIMAL(18,2),
24     CREATEDDATE DATETIME,
25     UPDATEDATE DATETIME,
26     CREATEDBY VARCHAR(100),
27     UPDATEDBY VARCHAR(100),
28     HASHKEY BIGINT
29 )
30
31
32
33
34
35
36
37
38
39
40
```

00 % No issues found

Messages

Commands completed successfully.

Completion time: 2025-08-13T07:32:38.5627042-04:00

```
CREATE TABLE DIM_Accounts

(
    account_id INT PRIMARY KEY,
    ID INT,
    account_type VARCHAR(50),
    balance DECIMAL(18,2),
```

```

CREATEDDATE DATETIME,
UPDATEDATE DATETIME,
CREATEDBY VARCHAR(100),
UPDATEDBY VARCHAR(100),
HASHKEY BIGINT
)

```

Add Derive column to generate hashkey

select ID as tgt_id, HASHKEY as tgt_hashkey from DIM_Customers

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure under "bhavsqlserver.database.windows.net (SQL Server)".
- SQL Query1.sql... (qldmin (86)):** The current query window contains the following code:

```

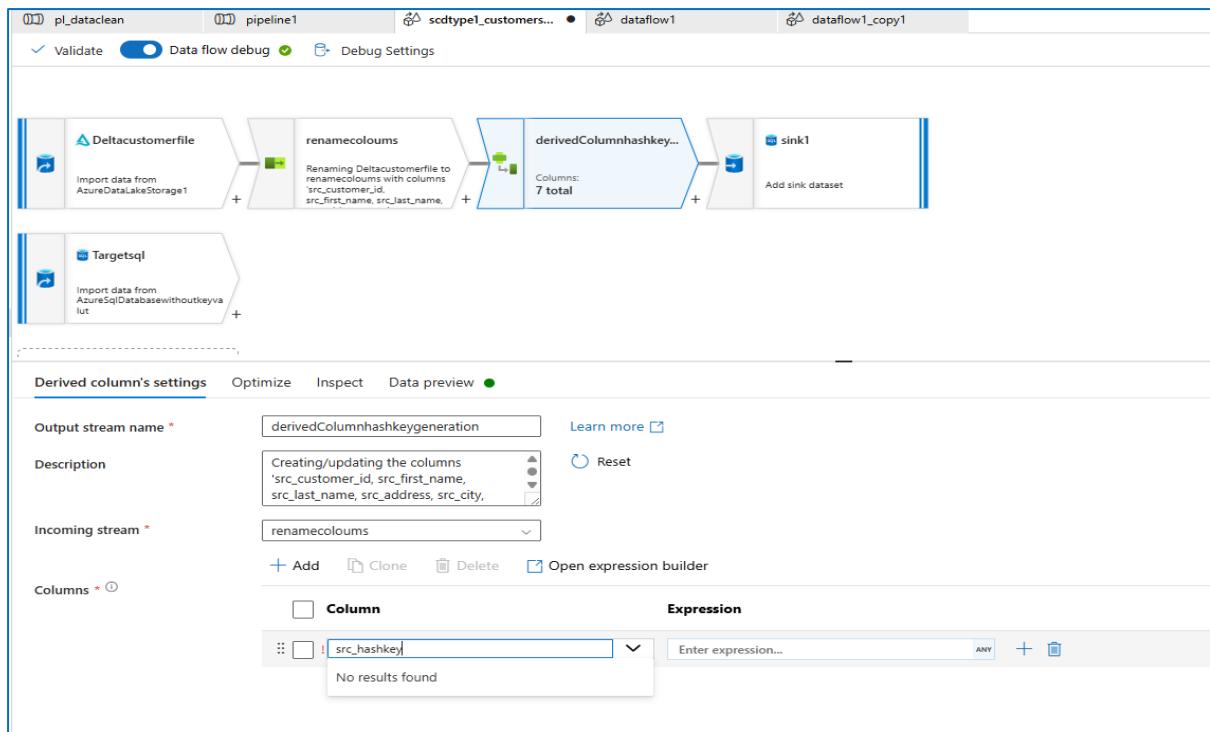
1  CREATE TABLE DIM_CUSTOMER
2  (
3      ID INT,
4      NAME VARCHAR(100),
5      CITY VARCHAR(100),
6      PHONENUMBER BIGINT,
7      CREATEDDATE DATETIME,
8      UPDATEDATE DATETIME,
9      CREATEDBY VARCHAR(100),
10     UPDATEDBY VARCHAR(100),
11     HASHKEY BIGINT
12 )
13
14 select * from DIM_CUSTOMER
15
16
17
18
19 select ID as tgt_id, HASHKEY as tgt_hashkey from DIM_CUSTOMER

```

- Results:** A results grid is shown with two columns: "tgt_id" and "tgt_hashkey".
- Status Bar:** Shows "No issues found".

Create src_hashkey as we will get few columns from customer using them we have to generate hashkey

`crc32(concat(toString(src_customer_id),src_first_name,src_last_name,src_addresses,src_city,src_state,src_zip))`



Screenshot of the Azure Data Factory Dataflow expression builder for the derived column 'src_hashkey'.

Derived Columns: src_hashkey

Column name: src_hashkey

Expression: concat(toString(src_customer_id),src_first_name,src_last_name,src_address,src_city,src_state,src_zip)

Expression elements:

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

Expression values:

- src_address
- src_city
- src_state
- src_zip
- abs(x) numeric_value

Data preview:

	src_customer_id	src_first_name	src_last_name	src_address	src_city	src_state	src_zip
68CharlotteGriffin	68	Charlotte	Griffin	6767 Poplar St	Victoria Harbour	ON	LOK0A1
71ChristopherMyers	71	Christopher	Myers	7070 Cedar Ln	Coldwater	ON	LOKOA1
82ElizabethReynolds	82	Elizabeth	Reynolds	8181 Poplar St	Sturgeon Falls	ON	P2B0A1
52AhhinailHanderson	52	Ahhinail	Handerson	5151 Curacao Ave.	Mount Albert	ON	LNVGN1

Microsoft Azure | Data Factory > adfcustomerproject1

Dataflow expression builder

derivedColumnhashkeygeneration

Derived Columns

+ Create new

abc src_hashkey

Column name * src_hashkey

Expression

```
crc32(concat(toString(src_customer_id),src_first_name,src_last_name,src_address,src_city,src_state,src_zip))
```

Expression elements

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

Expression values

src_customer_id src_first_name src_last_name src_address src_city src_state src_zip

Data preview Refresh

Save and finish Cancel Clear contents

Preview the data and see generated hashkey for each row

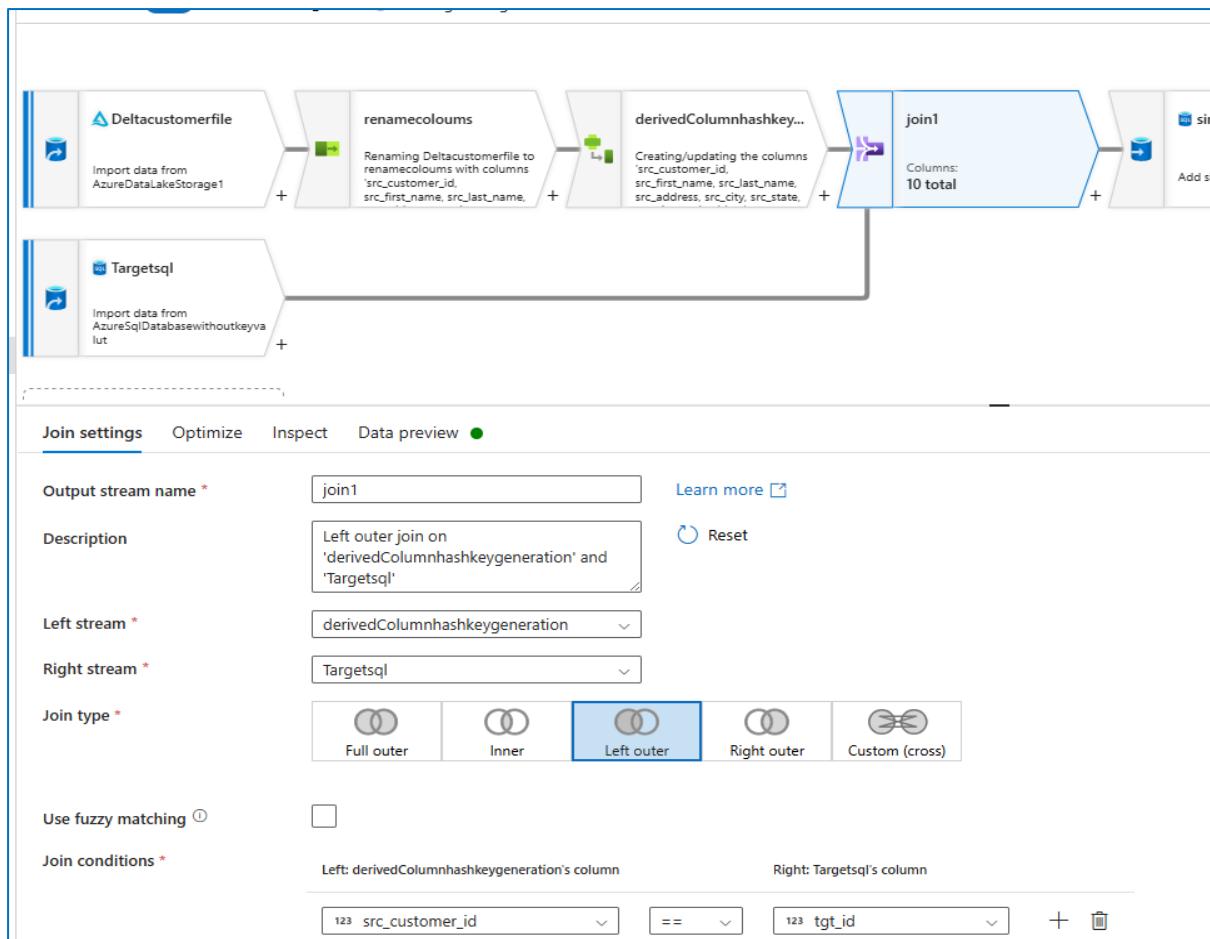
Derived column's settings		Optimize	Inspect	Data preview
Number of rows	INSERT 87	UPDATE 0	DELETE 0	UPSERT 0
Refresh	Typecast	Modify	Map drifted	Statistics Remove Export to CSV
↑↓	src_customer_id	↑↓↑↓	src_first_name	abc ↑↓ src_last_name abc ↑↓ src_address abc ↑↓ src_city abc ↑↓ src_state abc ↑↓ src_zip abc ↑↓ src_hashkey
+	68		Charlotte	Griffin 6767 Poplar St Victoria Harbour ON L0K0A1 3360600281
+	71		Christopher	Myers 7070 Cedar Ln Coldwater ON L0K0A1 2774108813
+	82		Elizabeth	Reynolds 8181 Poplar St Sturgeon Falls ON P2B0A1 3672179661
+	52		Abigail	Henderson 5151 Cypress Ave Mount Albert ON L0G0A1 569556118
+	4		Emily	Davis 101 Pine Rd Calgary AB T2A0A1 165372536
+	55		Michael	Butler 5454 Ash Blvd Holland Landing ON L9N0A1 2036728289
+	59		John	Flores 5858 Maple Ave Schomberg ON L0G0A1 935727508
+	56		Elizabeth	Long 5555 Beech Dr East Gwillimbury ON L9N0A1 2859890804
+	42		Charlotte	Richardson 4141 Beech Dr Newmarket ON L3Y0A1 2954704707
+	25		Daniel	Campbell 2424 Willow Rd St. Catharines ON L2R0A1 3303870564

Add join to combine source and target columns

to combine left columns, left join and if there are no data it should replace with null values

Everything from the left column will be taken and if no match blank columns or null values it has to return.

Joined id==id as we dont have any values from target table it is returning null values just see in data preview tab, because there is no matching source data in target data.



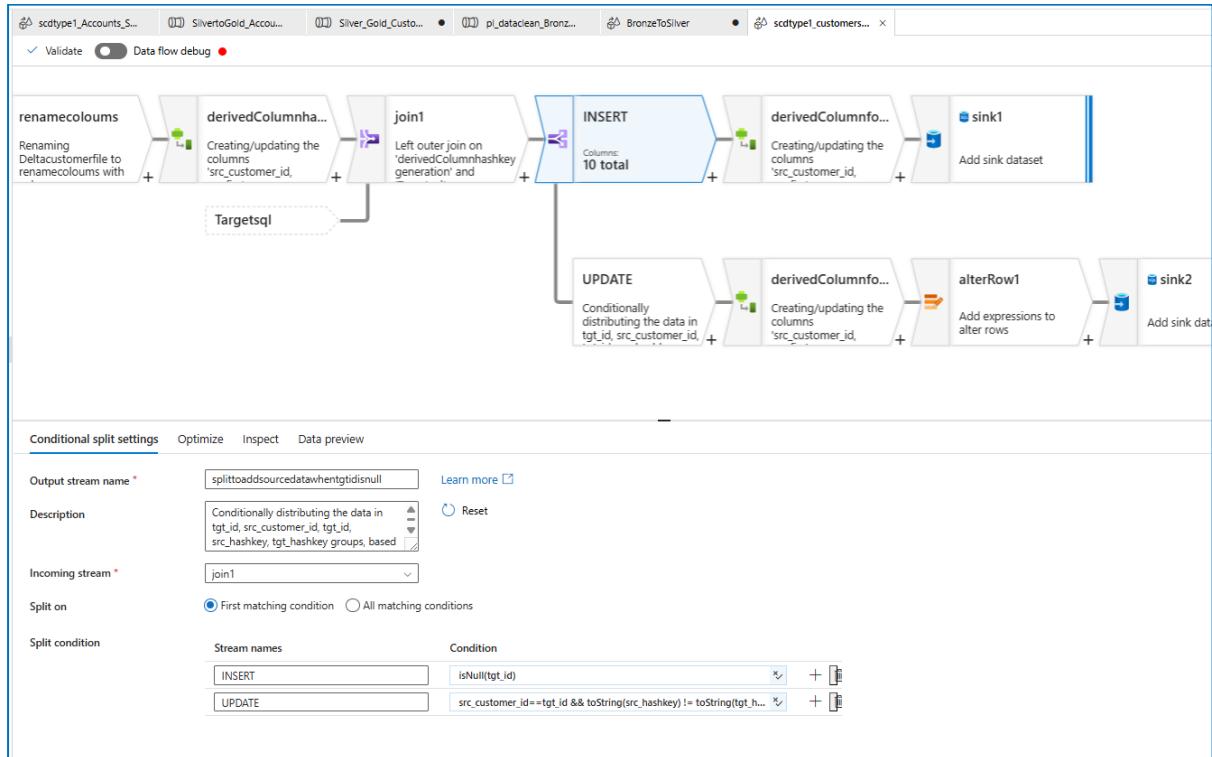
Data preview

	src_customer_id	src_first_name	src_last_name	src_address	src_city	src_state	src_zip	src_hashkey	tgt_id	tgt_hashkey
+	68	Charlotte	Griffin	6767 Poplar St	Victoria Harbour	ON	L0K0A1	68CharlotteGriffin6767 Poplar S...	NULL	NULL
+	71	Christopher	Myers	7070 Cedar Ln	Coldwater	ON	L0K0A1	71ChristopherMyers7070 Ceda...	NULL	NULL
+	82	Elizabeth	Reynolds	8181 Poplar St	Sturgeon Falls	ON	P2B0A1	82ElizabethReynolds8181 Popla...	NULL	NULL
+	52	Abigail	Henderson	5151 Cypress Ave	Mount Albert	ON	LOG0A1	52AbigailHenderson5151 Cyp...	NULL	NULL
+	4	Emily	Davis	101 Pine Rd	Calgary	AB	T2A0A1	4EmilyDavis101 Pine RdCalgary...	NULL	NULL
+	55	Michael	Butler	5454 Ash Blvd	Holland Landing	ON	L9N0A1	55MichaelButler5454 Ash Blvd...	NULL	NULL
+	59	John	Flores	5858 Maple Ave	Schomberg	ON	L0G0A1	59JohnFlores5858 Maple AveSc...	NULL	NULL
+	56	Elizabeth	Long	5555 Beech Dr	East Gwillimbury	ON	L9N0A1	56ElizabethLong5555 Beech Dr...	NULL	NULL
+	42	Charlotte	Richardson	4141 Beech Dr	Newmarket	ON	L3Y0A1	42CharlotteRichardson4141 Be...	NULL	NULL
+	25	Daniel	Campbell	2424 Willow Rd	St. Catharines	ON	L2R0A1	25DanielCampbell2424 Willow ...	NULL	NULL

Add Conditional split to separate insert and update data rows

So whenever target id is null which means no source is matching to target table data then it has to go with insert operation, we have to use conditional split(based on the conditions we can segregate the data into multiple sources).

Now we have to write the condition whenever we see target id is null we have to insert the new data i.e thumb rule



IsNull(tgt_id)

The screenshot shows the Microsoft Azure Data Factory Dataflow expression builder. The main area displays the expression `IsNull(tgt_id)`. The left sidebar lists Expression elements: All, Functions, Input schema, Parameters, Cached lookup, and Data flow library functions. The bottom section includes buttons for Save and Publish, Cancel, and Clear contents.

Now do update when we have to go with update is when there are records exists target side which means `source_id==tgt_id`

`src_customer_id==tgt_id` --->for insert

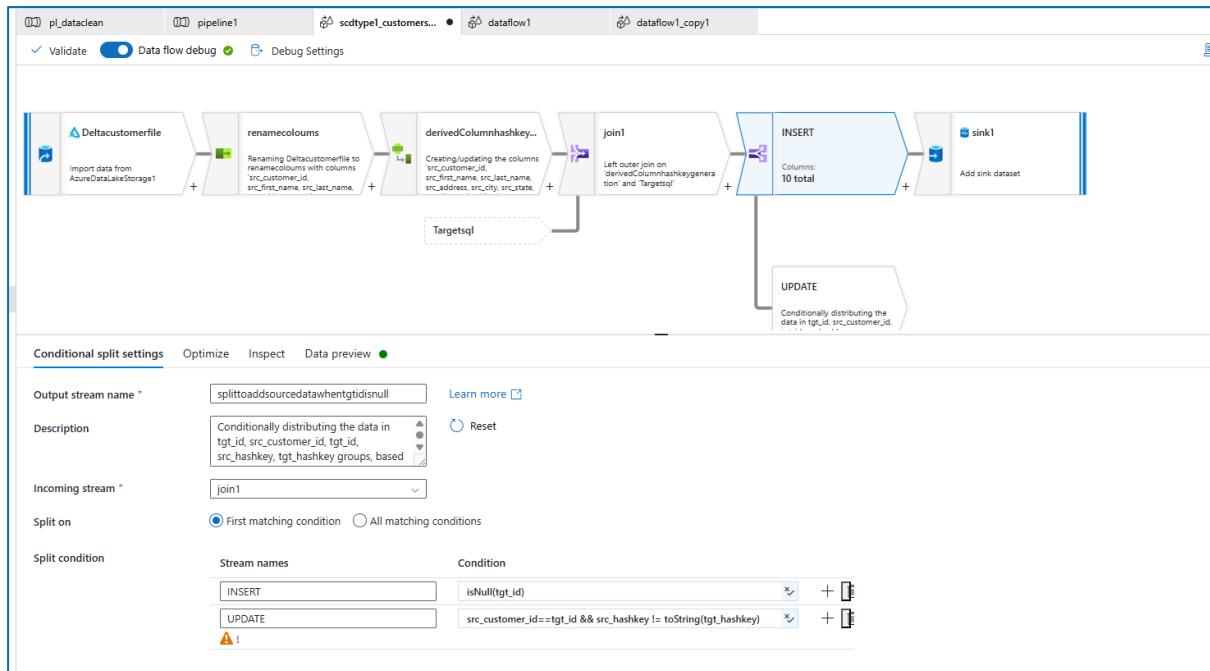
Now to update the records when the ids are matching and hashkey not matched it will go with update operation. As there are changes in other columns then we can recognize using hashkey not matched then it will go with update operation.

`src_hashkey != tgt_hashkey`

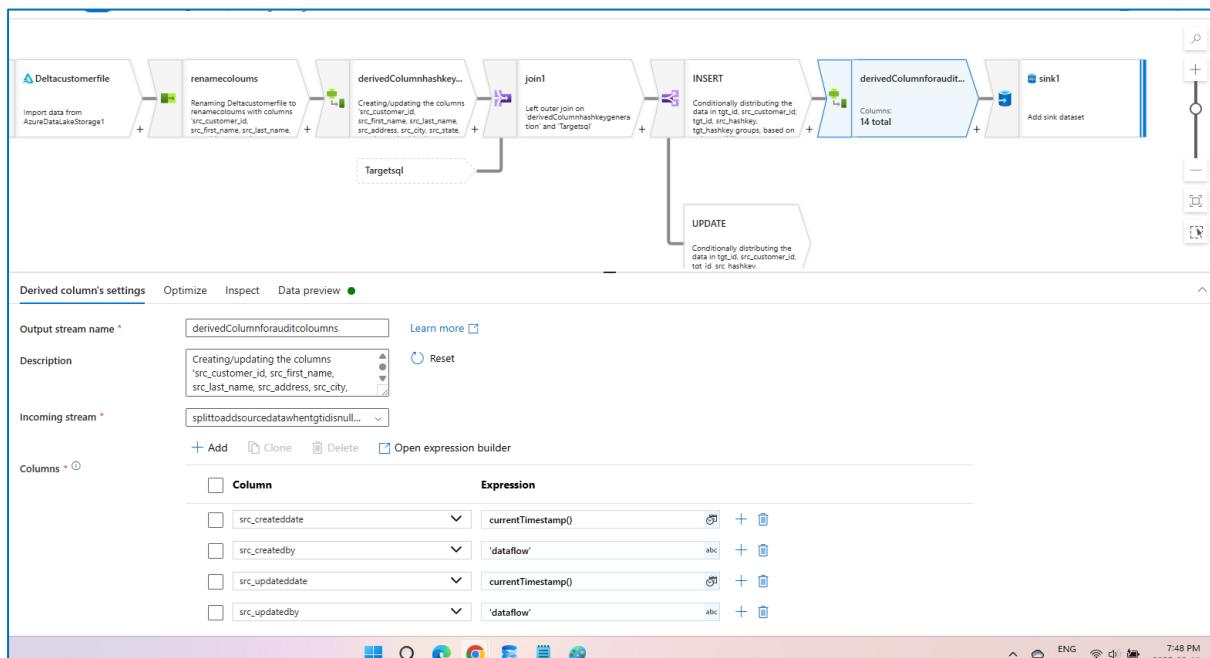
The screenshot shows the Microsoft Azure Data Factory Dataflow expression builder. The main area displays the expression `src_customer_id==tgt_id`. Below the expression, the Expression elements sidebar lists various functions and parameters. The Data preview section shows a single row of data with columns: src_customer_id, tgt_id, src_hashkey, and tgt_hashkey. The src_customer_id and tgt_id values are 68 and 71 respectively, while the hashkeys are 3360600281 and 2774108813, which are different.

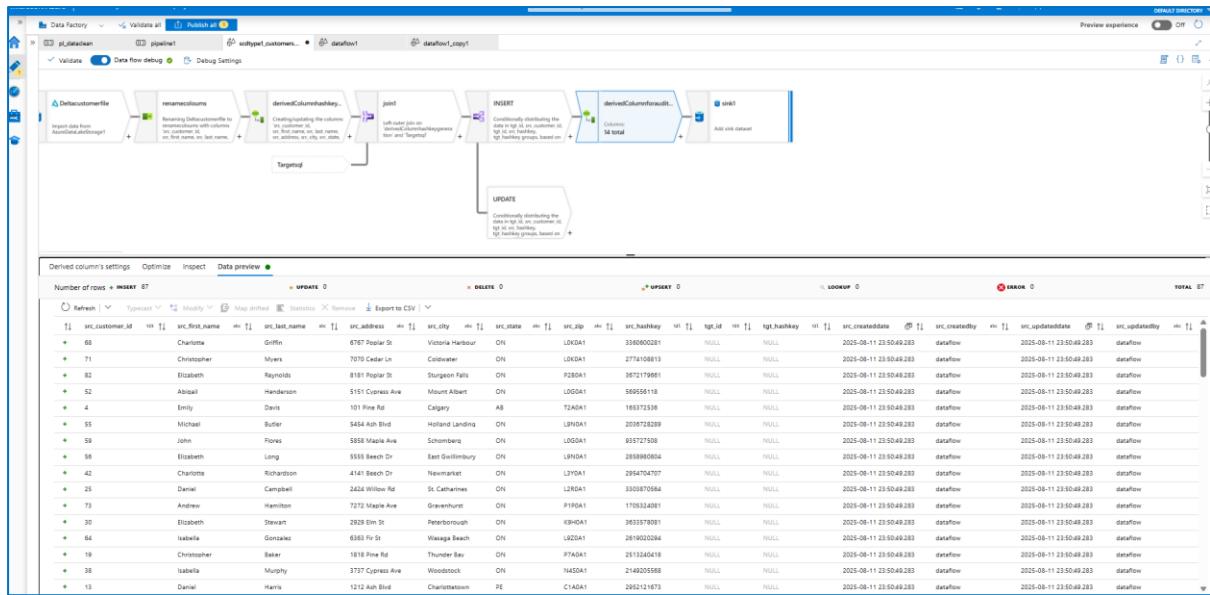
`src_customer_id==tgt_id && toString(src_hashkey) != toString(tgt_hashkey)`

This screenshot shows the same Dataflow expression builder interface as the previous one, but with a more complex expression: `src_customer_id==tgt_id && toString(src_hashkey) != toString(tgt_hashkey)`. The Expression elements sidebar now includes additional columns from the source data: src_first_name, src_last_name, src_address, and src_city. The Data preview section shows four rows of data, each with different values for these additional columns, further illustrating how the expression handles multiple columns during a comparison.

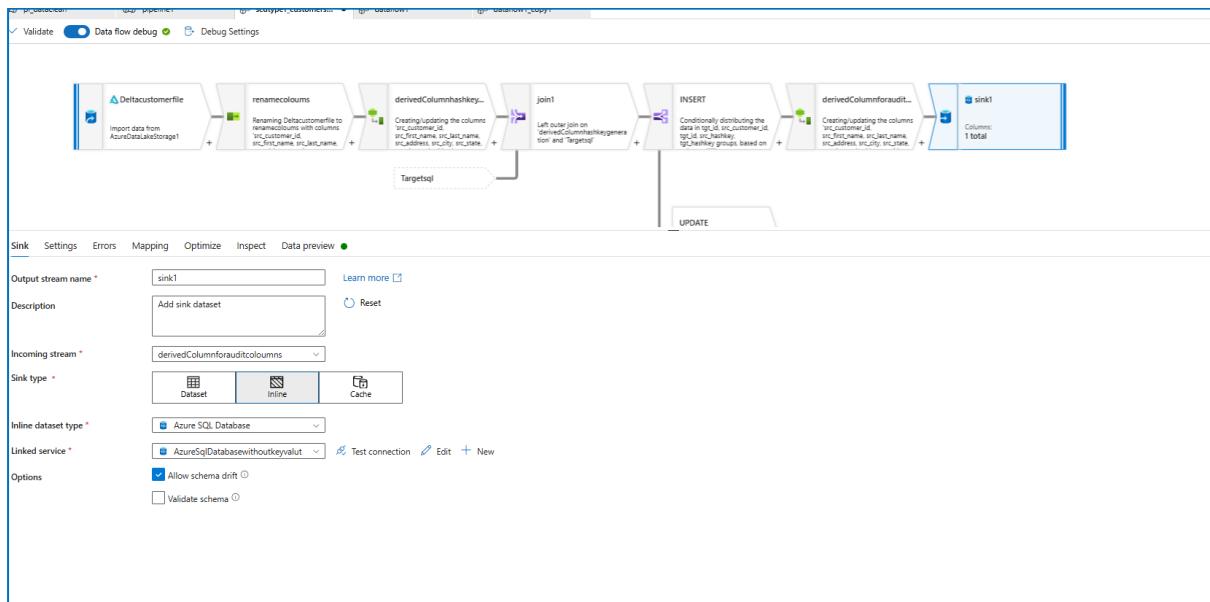


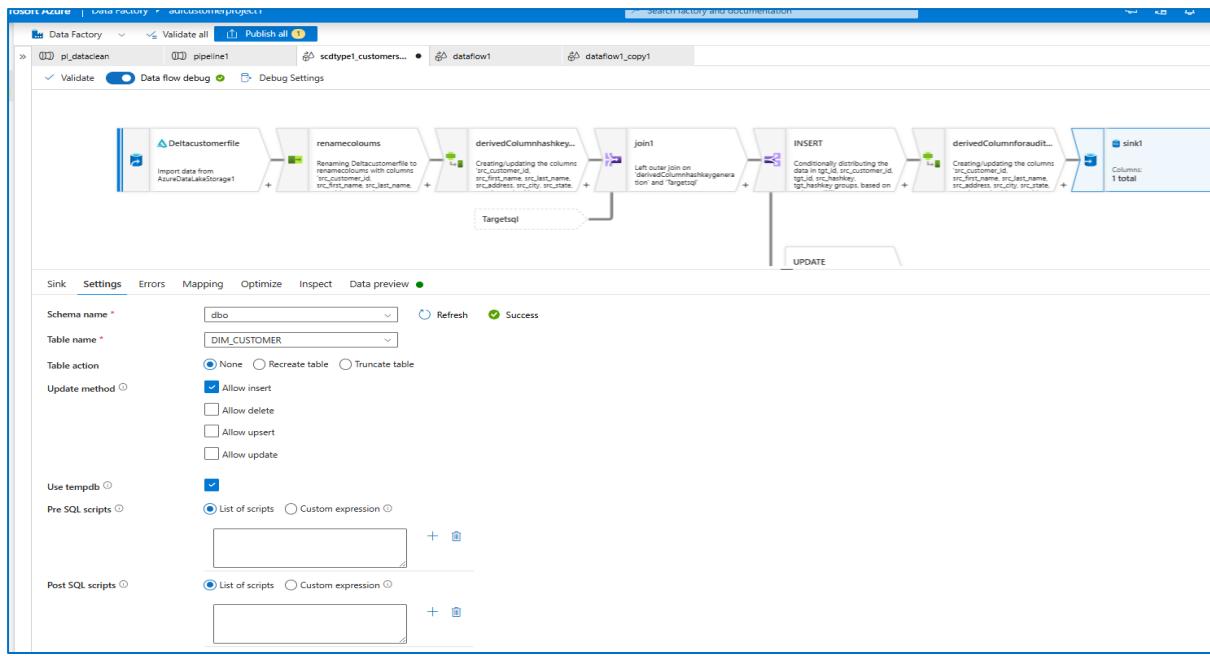
Now add 1 more derived column to add Audit columns(CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY)



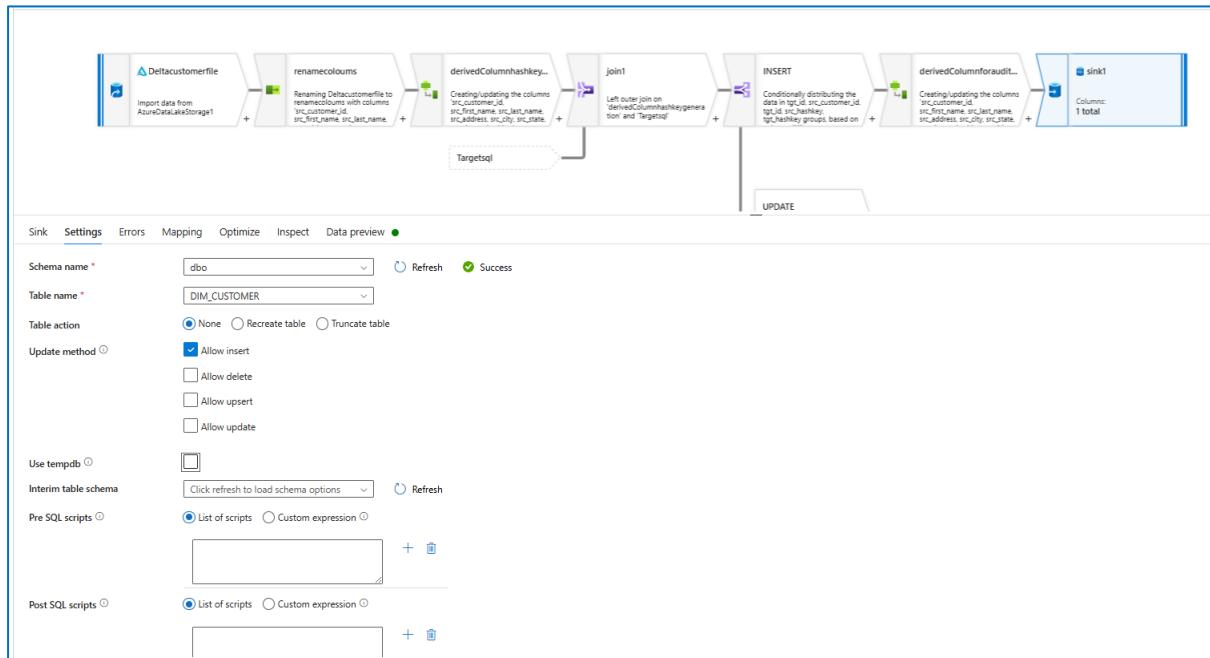


Now add Sink to load the final data(insert new records)





Remove use tempdb uncheck it



Remove automapping and generate import schema then do mapping

The screenshot shows the Mapping Designer interface with the following configuration:

- Sink:** Schema name: dbo, Table name: DIM_CUSTOMER, Table action: None, Update method: Allow insert.
- Interim table schema:** Click refresh to load schema options.
- Pre SQL scripts:** List of scripts.
- Post SQL scripts:** List of scripts.

Mapping Details:

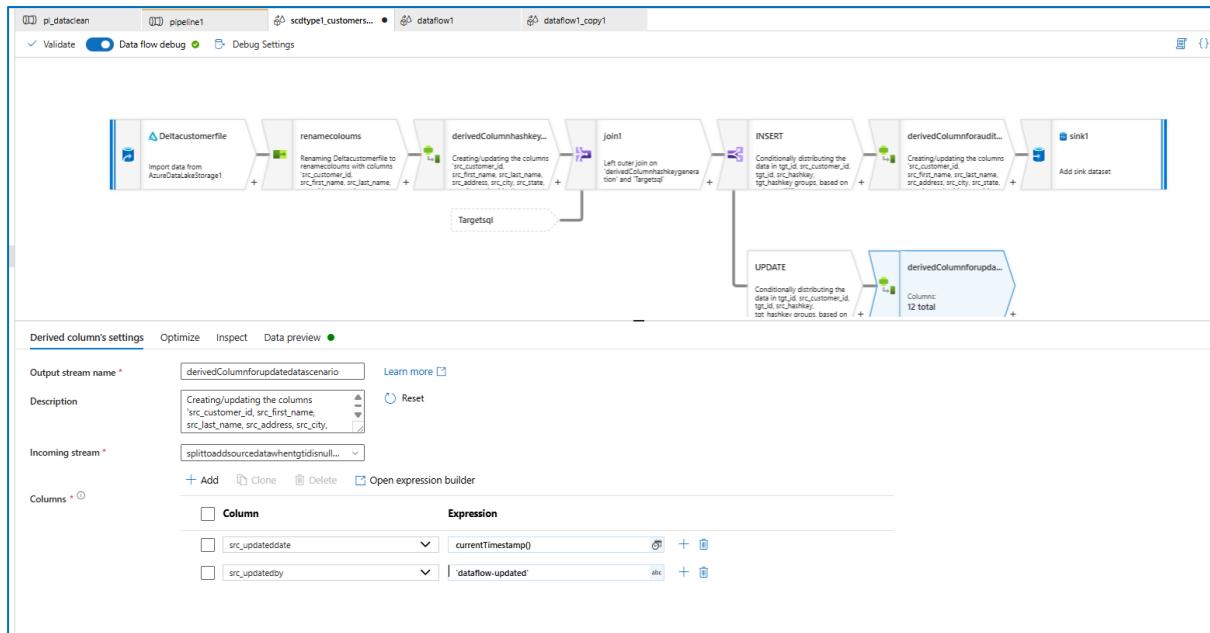
- Input columns:** src_customer_id, src_first_name, src_city, src_zip, src_createdate, src_updatedate, src_createdby, src_updatedby, src_hashkey.
- Output columns:** ID, NAME, CITY, PHONENUMBER, CREATEDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, HASHKEY.
- Mapping Options:** Skip duplicate input columns, Skip duplicate output columns.

Data Preview:

ID	NAME	CITY	PHONENUMBER	CREATEDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
68	Charlotte	Victoria Harbour	NULL	2025-08-12 00:05:38.833	2025-08-12 00:05:38.833	dataflow	dataflow	3360600281
71	Christopher	Goldwater	NULL	2025-08-12 00:05:38.833	dataflow	dataflow	dataflow	1774108813
82	Elizabeth	Sturgeon Falls	NULL	2025-08-12 00:05:38.833	2025-08-12 00:05:38.833	dataflow	dataflow	3672179661
52	Ariyal	Mount Albert	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	569556118
4	Emily	Calgary	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	165372536
53	Michael	Holland Landing	NULL	2025-08-12 00:05:38.833	2025-08-12 00:05:38.833	dataflow	dataflow	2036728289
59	John	Schomberg	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	935727508
56	Elizabeth	East Gwillimbury	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	285898004
42	Charlotte	Newmarket	NULL	2025-08-12 00:05:38.833	2025-08-12 00:05:38.833	dataflow	dataflow	2954704707
25	Daniel	St. Catharines	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	3303870564
73	Andrew	Gravenhurst	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	1705524081
30	Elizabeth	Peterborough	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	3633578081
64	Isabella	Wasaga Beach	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	2619020294
19	Christophe	Thunder Bay	NULL	2025-08-12 00:05:38.833	2025-09-12 00:05:38.833	dataflow	dataflow	2513240418

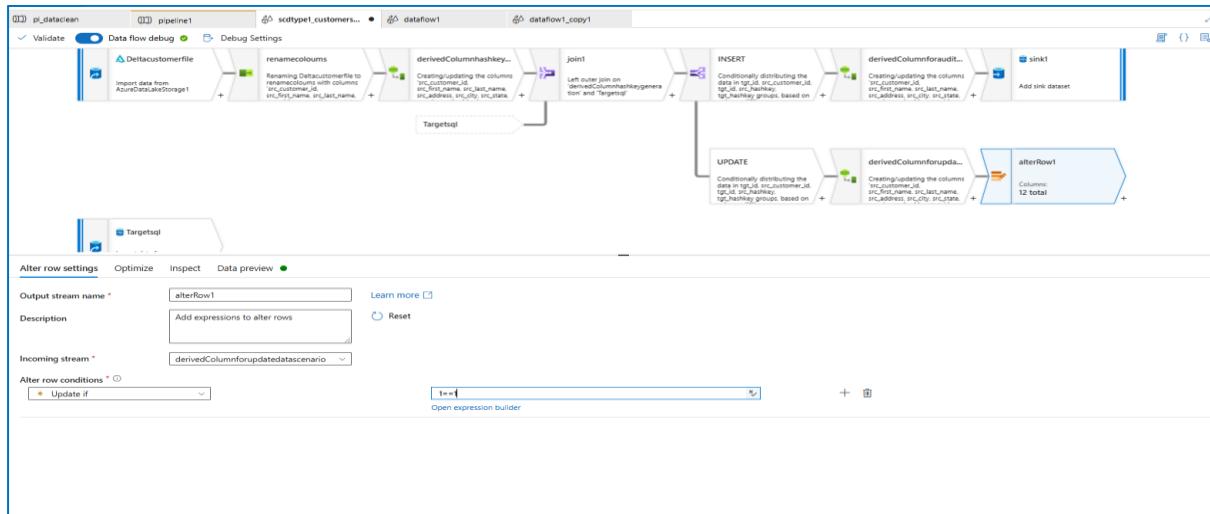
Derived column for update

Now go with update scenario if there is something change in data go with update for that take derived column

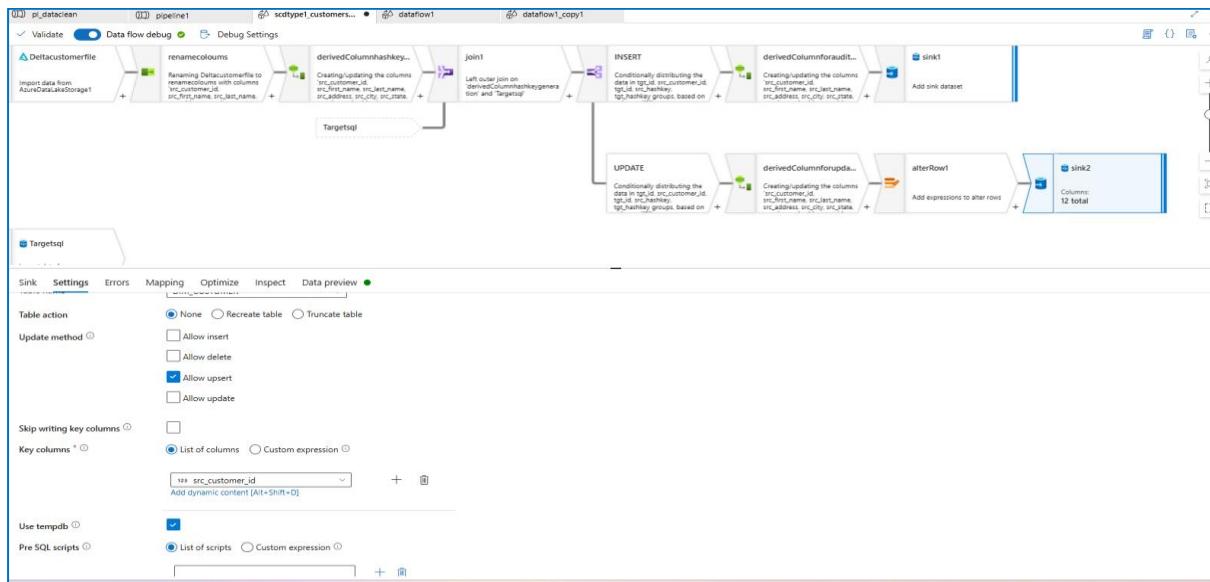


Dont preview data as we dont have updated data

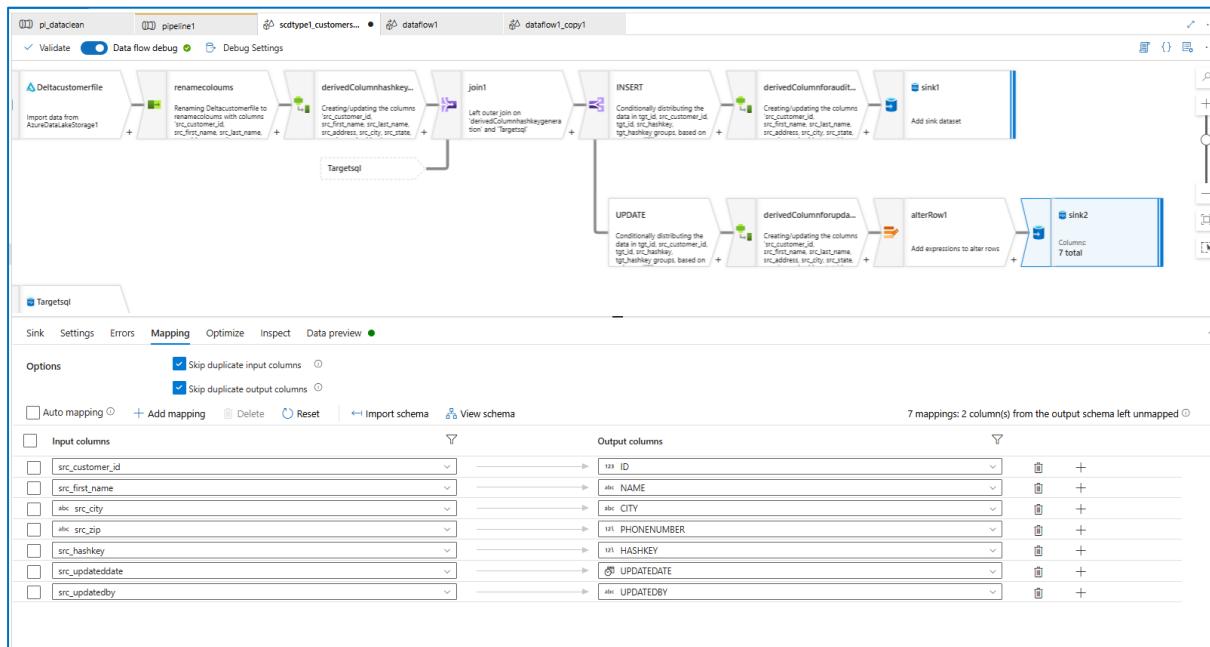
Now we ahve to take alterrow to give some permision to update when it matches the condition 1==1



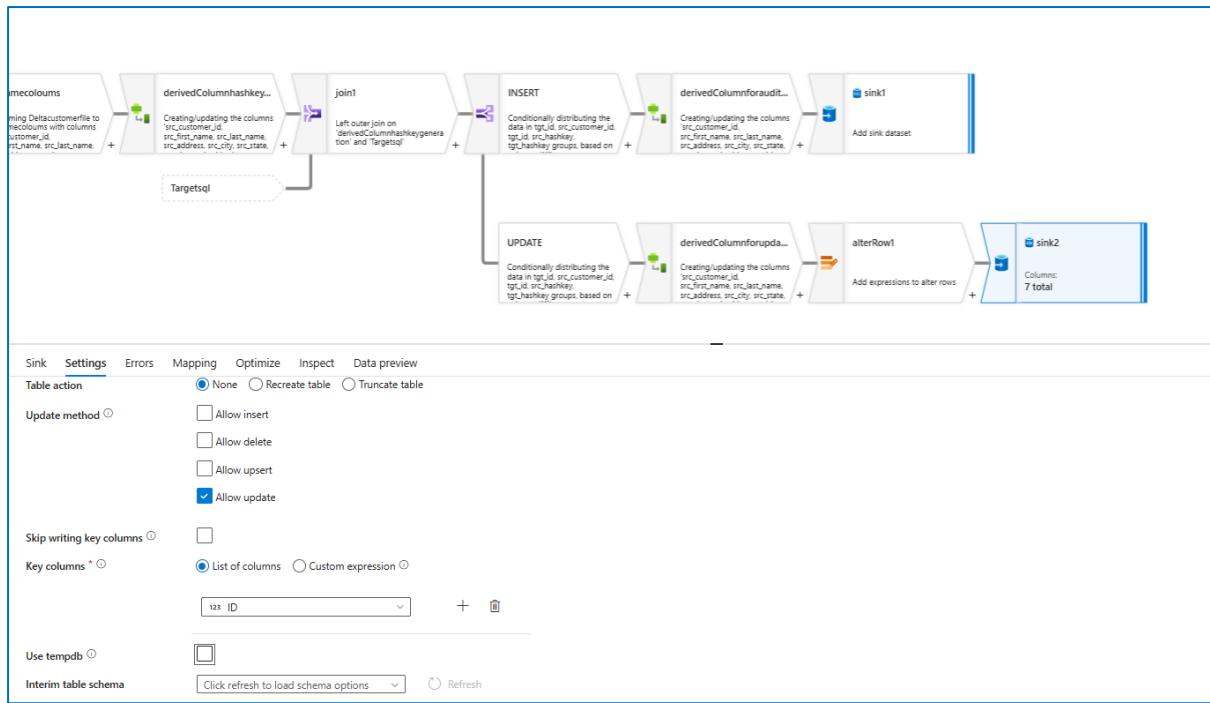
Add sink for update data



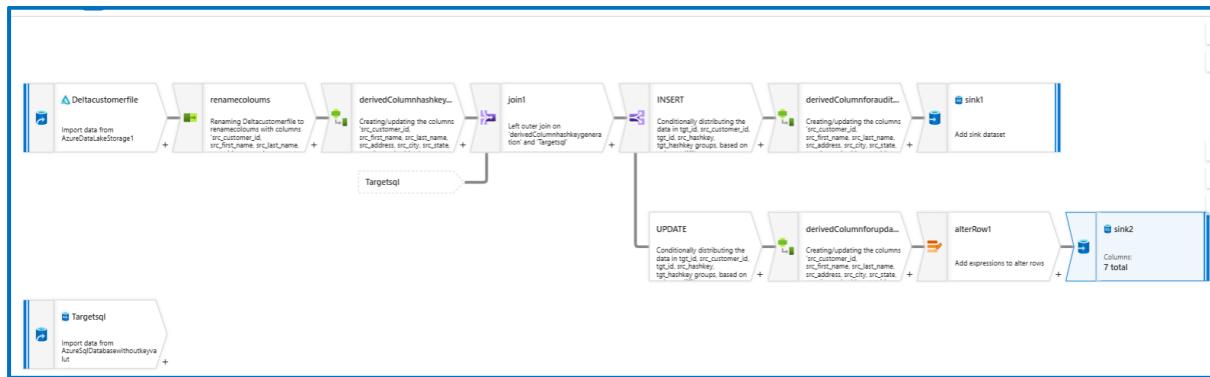
Do mapping



Note: After you import the coloumns and map the coloumns then you come to settings tab and select your ID as key



Final pipeline scdtype1



For now I'm just inserting data as i dont have updated data

Run the pipeline and see the output

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure for 'bhavsqlserver.database.windows.net (SQL Server)'.
- SQL Query2.sql...gladmin (800)***: The current query window contains the following T-SQL code:


```

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
      select * from DIM_CUSTOMER
    
```
- Results Grid:** Displays the output of the query. The 'Messages' table has columns: ID, FIRSTNAME, CITY, CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, and HASHKEY. The data includes 19 rows of customer information such as David, North Bay, John, Temagami, Daniel, Elmvale, James, Queensville, Abigail, Sundridge, Olivia, Orilla, Emily, Bradford, Barbara, Sudbury, James, Guelph, Annelia, Bradford, Andrew, Uxbridge, John, Toronto, Abigail, Mount Albert, Michael, Montreal, Ma, London, Sophia, Milton, Paul, Penetanguishene, and Matthew, Whitehorse.

INPUT:

```
{
  "dataflow": {
    "referenceName": "scdtype1_customers_bootcampstpe3",
    "type": "DataFlowReference",
    "parameters": {},
    "datasetParameters": {
      "Deltacustomerfile": {}
    },
    "Targetsqli": {},
    "sink1": {}
  },
  "staging": {},
  "compute": {
    "coreCount": 8,
    "computeType": "General"
  },
  "traceLevel": "Fine",
  "dataFlowDebugSessionId": "622c5e8b-2bd4-4206-8fe4-ea2842799623",
  "continuationSettings": {
    "customizedCheckpointKey": "pipeline1-DFscdtype1 customersfile-fa7f7fcf-9548-4893-8d0c-8a70202f5582"
  }
}
```

Input

```
{
  "dataflow": {
    "referenceName": "scdtype1_customers_bootcampstpe3",
    "type": "DataFlowReference",
    "parameters": {},
    "datasetParameters": {
      "Deltacustomerfile": {}
    },
    "Targetsqli": {},
    "sink1": {}
  },
  "staging": {},
  "compute": {
    "coreCount": 8,
    "computeType": "General"
  },
  "traceLevel": "Fine",
  "dataFlowDebugSessionId": "622c5e8b-2bd4-4206-8fe4-ea2842799623",
  "continuationSettings": {
    "customizedCheckpointKey": "pipeline1-DFscdtype1 customersfile-fa7f7fcf-9548-4893-8d0c-8a70202f5582"
  }
}
```

OUTPUT:

```
{ "runStatus": { "ClusterId": "adfcustomerproject1.AutoResolveIntegrationRuntime.9", "sparkVersion": "3.4", "computeAcquisitionDuration": 1863, "version": "20250711.2", "profile": { "derivedColumnforauditcoloumns": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 4, "total": 14, "updated": 0 }, "Targetsqli": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 2, "total": 2, "updated": 0 }, "renamocoloums": { "computed": [], "lineage": {}, "dropped": 7, "drifted": 0, "newer": 7, "total": 7, "updated": 0 }, "join1": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 10, "updated": 0 }, "derivedColumnhashkeygeneration": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 1, "total": 8, "updated": 0 }, "Deltacustomerfile": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 7, "total": 7, "updated": 0 }, "splittoaddsourcedatawhentgtidisNull@UPDATE": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 10, "updated": 0 }, "sink1": { "computed": [ { "source": "Deltacustomerfile", "columns": [ "zip", "customer_id", "address", "state", "first_name", "last_name", "city" ] }, { "source": "Targetsqli", "columns": [ "tgt_id", "tgt_hashkey" ] } ], "lineage": { "ID": { "mapped": true, "from": [ { "source": "Deltacustomerfile", "columns": [ "customer_id" ] } ] }, "NAME": { "mapped": true, "from": [ { "source": "Deltacustomerfile", "columns": [ "first_name" ] } ] }, "CITY": { "mapped": true, "from": [ { "source": "Deltacustomerfile", "columns": [ "city" ] } ] }, "PHONENUMBER": { "mapped": true, "from": [ { "source": "Deltacustomerfile", "columns": [ "zip" ] } ] }, "UPDATEDBY": { "mapped": false, "from": [ { "source": "derivedColumnforauditcoloumns", "columns": [ "src_updatedby" ] } ] }, "CREATEDDATE": { "mapped": false, "from": [ { "source": "derivedColumnforauditcoloumns", "columns": [ "src_createddate" ] } ] }, "UPDATEDATE": { "mapped": false, "from": [ { "source": "derivedColumnforauditcoloumns", "columns": [ "src_updateddate" ] } ] }, "CREATEDBY": { "mapped": false, "from": [ { "source": "derivedColumnforauditcoloumns", "columns": [ "src_createdby" ] } ] }, "HASHKEY": { "mapped": false, "from": [ { "source": "Deltacustomerfile", "columns": [ "zip", "customer_id", "address", "state", "first_name", "last_name", "city" ] } ] }, "dropped": 14, "drifted": 0, "newer": 9, "total": 9, "updated": 0 }, "splittoaddsourcedatawhentgtidisNull@INSERT": { "computed": [], "lineage": {}, "dropped": 0, "drifted": 0, "newer": 0, "total": 10, "updated": 0 }, "metrics": { "sink1": { "format": "table", "stages": [ { "stage": 89, "partitionTimes": [ 997, 1045, 994, 991 ] }, "recordsWritten": 87, "lastUpdateTime": "2025-08-12 00:08:18.868", "bytesWritten": 0, "recordsRead": 87, "bytesRead": 125812, "partitionStatus": "Success", "streams": { "renamocoloums": { "count": 87, "cached": false, "totalPartitions": 4, "partitionStatus": "Success", "partitionCounts": [ 22, 22, 22, 21 ], "type": "select" }, "Deltacustomerfile": { "count": 87, "cached": false, "totalPartitions": 4, "partitionStatus": "Success", "partitionCounts": [ 22, 22, 22, 21 ], "type": "source" }, "derivedColumnhashkeygeneration": { "count": 87, "cached": false, "totalPartitions": 4, "partitionStatus": "Success", "partitionCounts": [ 22, 22, 22, 21 ], "type": "derive" } }, "target": "sink1", "time": 1099, "progressState": "Completed" }, { "stage": 91 }
```



```

12820 } }, "clusterComputeId": "17f6ec3f-f38d-4133-a3aa-ecb670bcac21", "dsl":
"\nsource() ~> Deltacustomerfile\n\nDeltacustomerfile select() ~>
renamocoloums\n\nrenamocoloums derive() ~>
derivedColumnhashkeygeneration\n\nnsource() ~>
Targetsql\nnderivedColumnhashkeygeneration, Targetsql join() ~> join1\n\njoin1
split() ~> splittoaddsourcedatawhentgtidisnull@(INSERT,
UPDATE)\n\nsplittoaddsourcedatawhentgtidisnull@INSERT derive() ~>
derivedColumnforauditcoloumns\nnderivedColumnforauditcoloumns sink() ~>
sink1" }, "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (Canada
Central)", "billingReference": { "activityType": "executedataflow", "billableDuration": [
{ "meterType": "Data Flow", "duration": 0.09050433022222222, "unit": "coreHour",
"sessionType": "JobCluster" } ] }, "reportLineageToPurview": { "status": "NotReported" } }

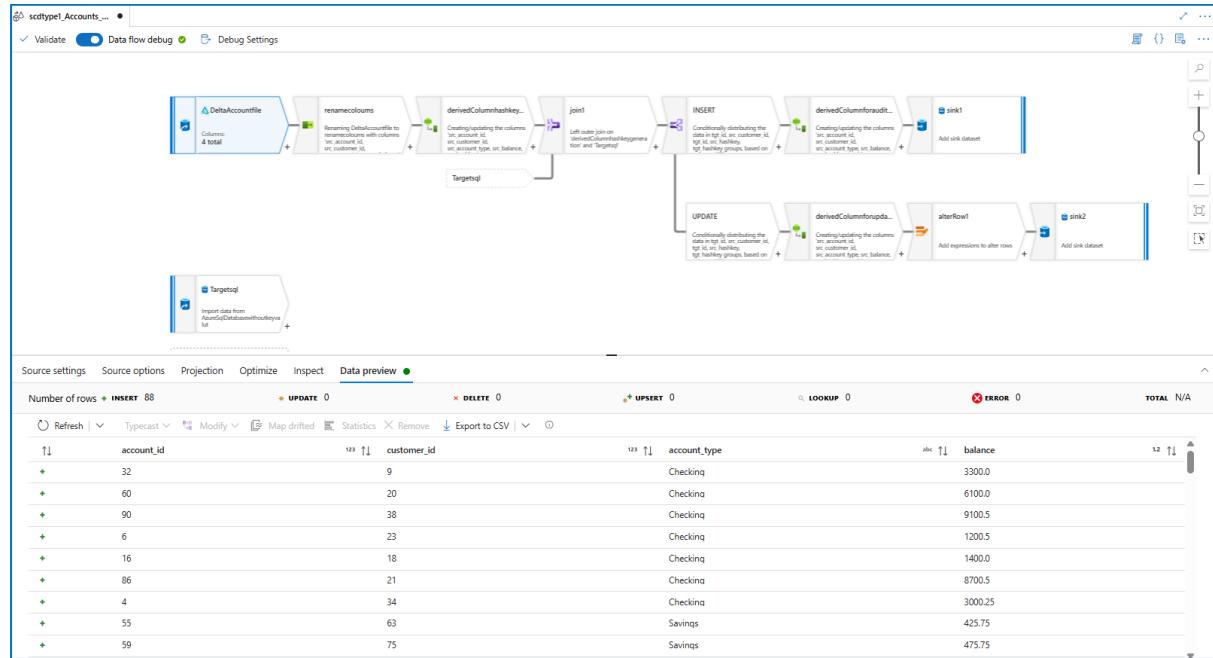
```

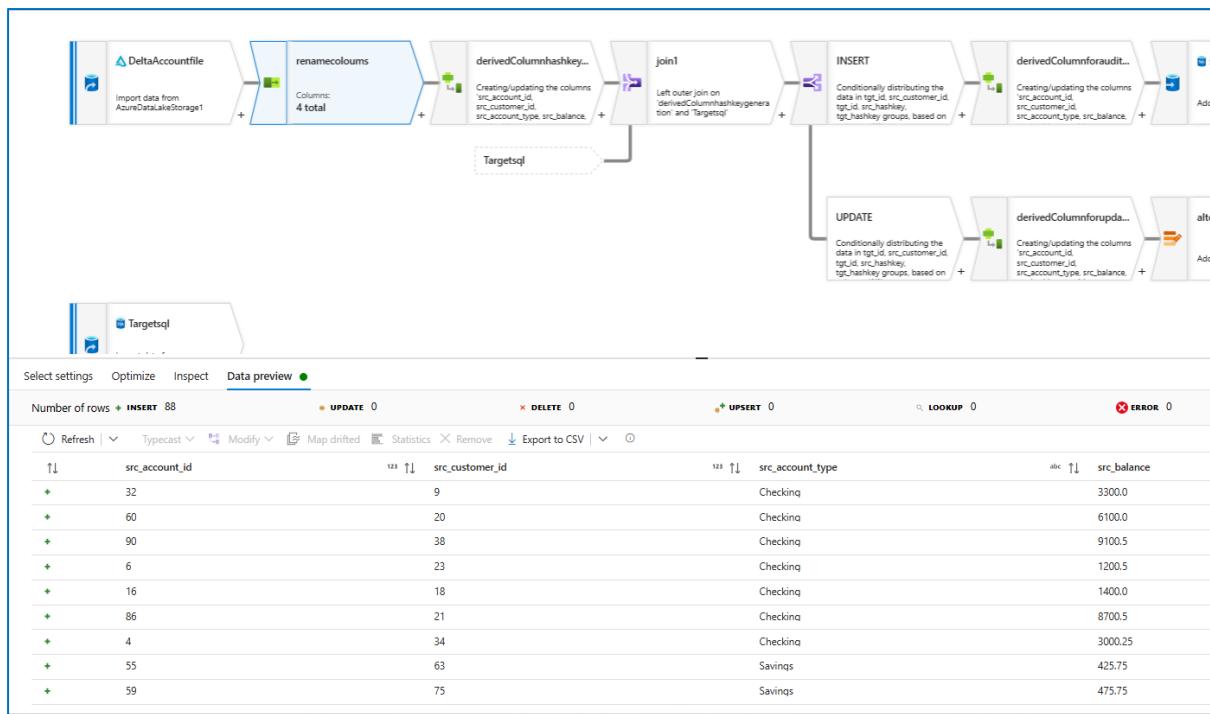
Output

Copy to clipboard

```
{
  "runStatus": {
    "ClusterId": "adfcustomerproject1.AutoResolveIntegrationRuntime.9",
    "sparkVersion": "3.4",
    "computeAcquisitionDuration": 1863,
    "version": "20250711.2",
    "profile": {
      "derivedColumnforauditcoloumns": {
        "computed": [],
        "lineage": {},
        "dropped": 0,
        "drifted": 0,
        "newer": 4,
        "total": 14,
        "updated": 0
      },
      "Targetsqli": {
        "computed": [],
        "lineage": {},
        "dropped": 0,
        "drifted": 0,
        "newer": 2,
        "total": 2,
        "updated": 0
      },
      "renamocoloums": {
        "computed": [],
        "lineage": {},
        "dropped": 7,
        "drifted": 0,
        "newer": 7,
        "total": 7,
        "updated": 0
      },
      "join1": {
        "computed": [],
        "lineage": {},
        "dropped": 0,
        "drifted": 0,
        "newer": 0,
        "total": 10,
        "updated": 0
      },
      "derivedColumnhashkeygeneration": {
        "computed": [],
        "lineage": {},
        "dropped": 0,
        "drifted": 0,
        "newer": 1,
        "total": 8,
        "updated": 0
      }
    }
  }
}
```

SCD TYPE-2 For Accounts





Do crc32 to generate hashkey

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

The screenshot shows the Azure Data Factory Data Flow expression builder. The 'derivedColumnhashkeygeneration' derived column is selected. The 'Expression' field contains the following code:

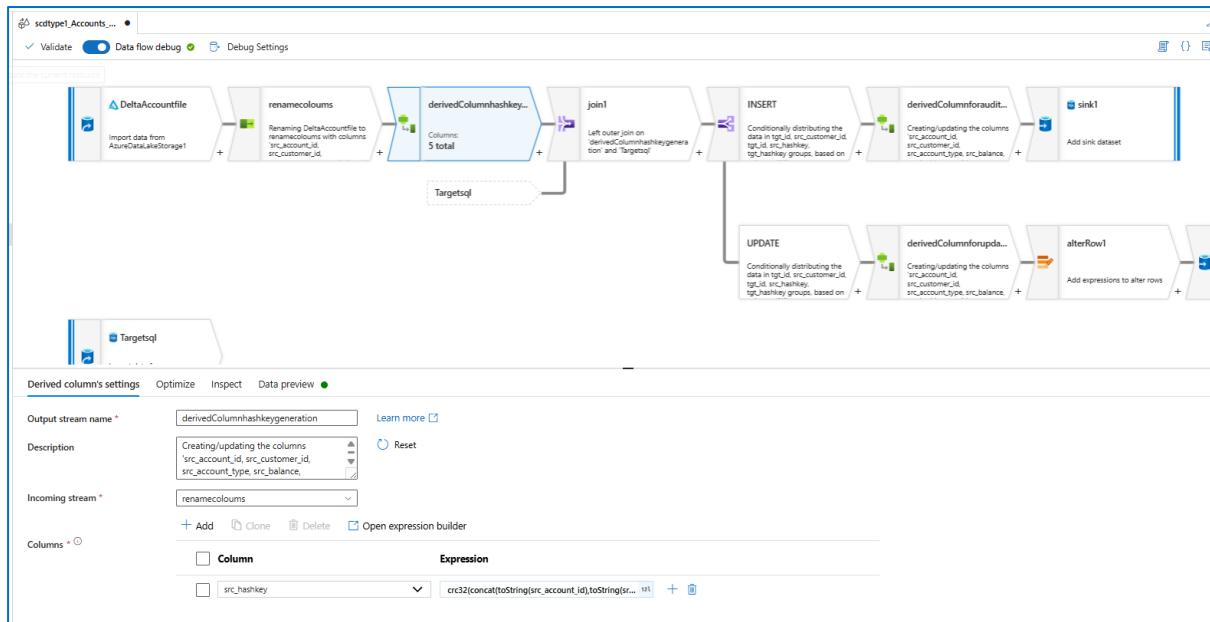
```
crc32(concat(toString(src_account_id),toString(src_customer_id),src_account_type,toString(src_balance)))
```

The 'Expression elements' sidebar lists the variables used in the expression:

- All: src_account_id, src_customer_id, src_account_type, src_balance
- Functions: abs, acos, add, first, second
- Input schema: src_account_id, src_customer_id, src_account_type, src_balance
- Parameters: None
- Cached lookup: None
- Data flow library functions: None
- Locals: None

The 'Data preview' section shows the generated hashkeys for the same set of source data:

src_account_id	src_customer_id	src_account_type	src_balance	src_hashkey
32	9	Checking	3300.0	2957212870
60	20	Checking	6100.0	1825281187
90	38	Checking	9100.5	2053712775
6	23	Checking	1200.5	6037340794
16	18	Checking	1400.0	10000000000000000000
86	21	Checking	8700.5	10000000000000000000
4	34	Checking	3000.25	10000000000000000000
55	63	Savings	425.75	10000000000000000000
59	75	Savings	475.75	10000000000000000000



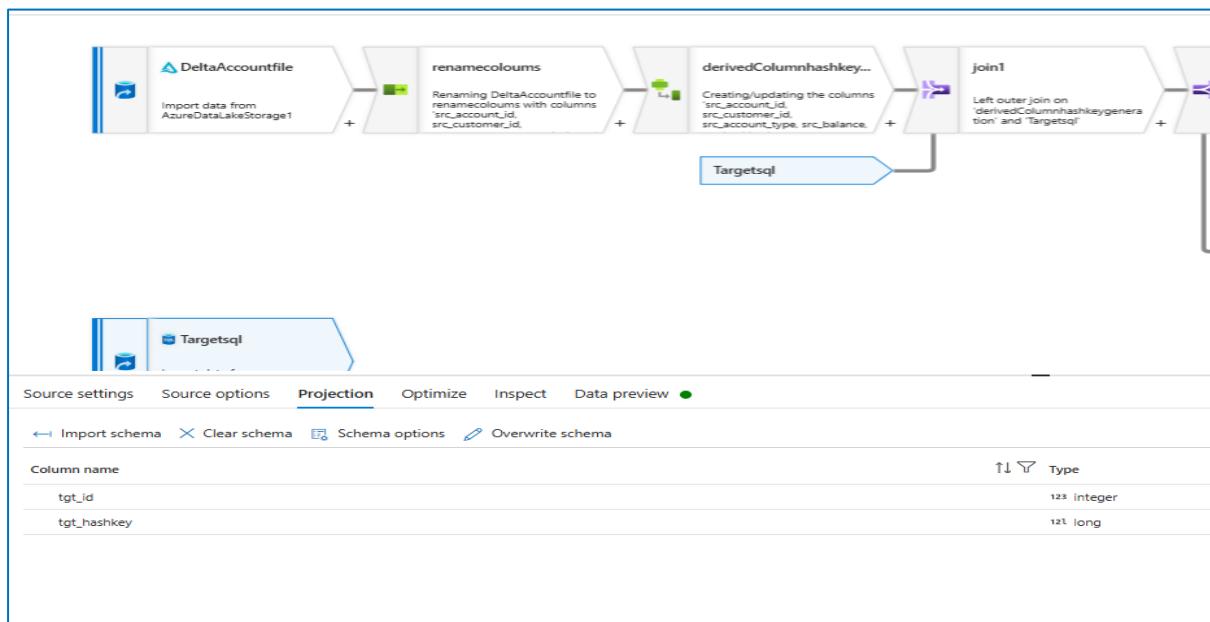
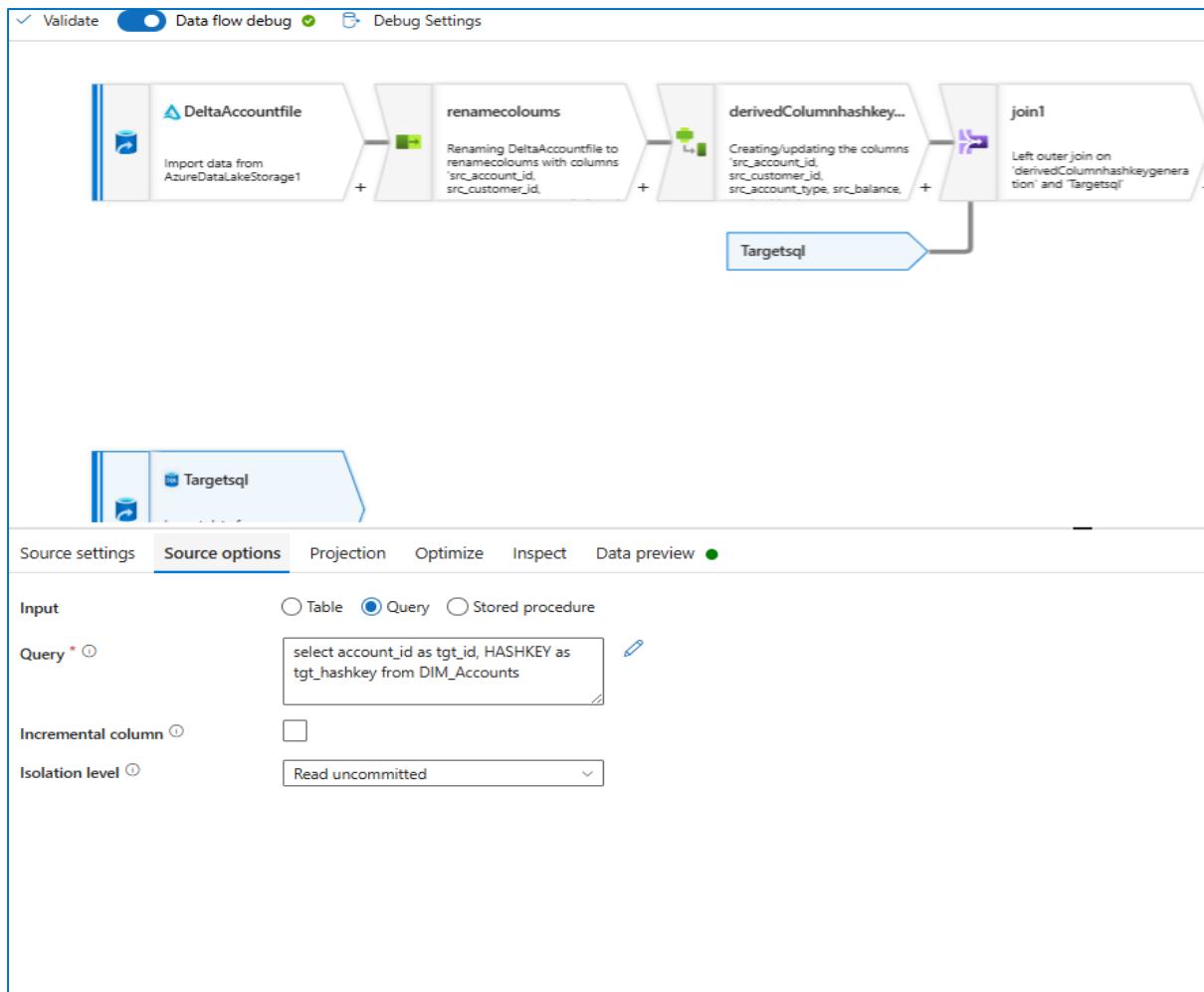
```

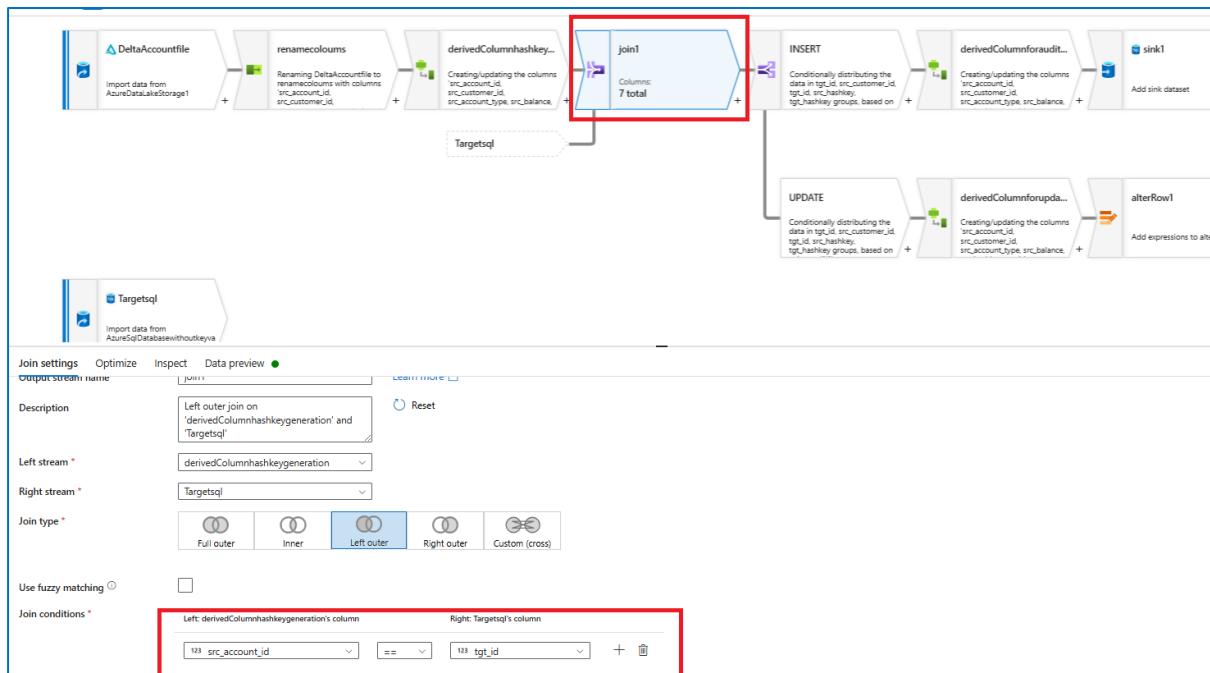
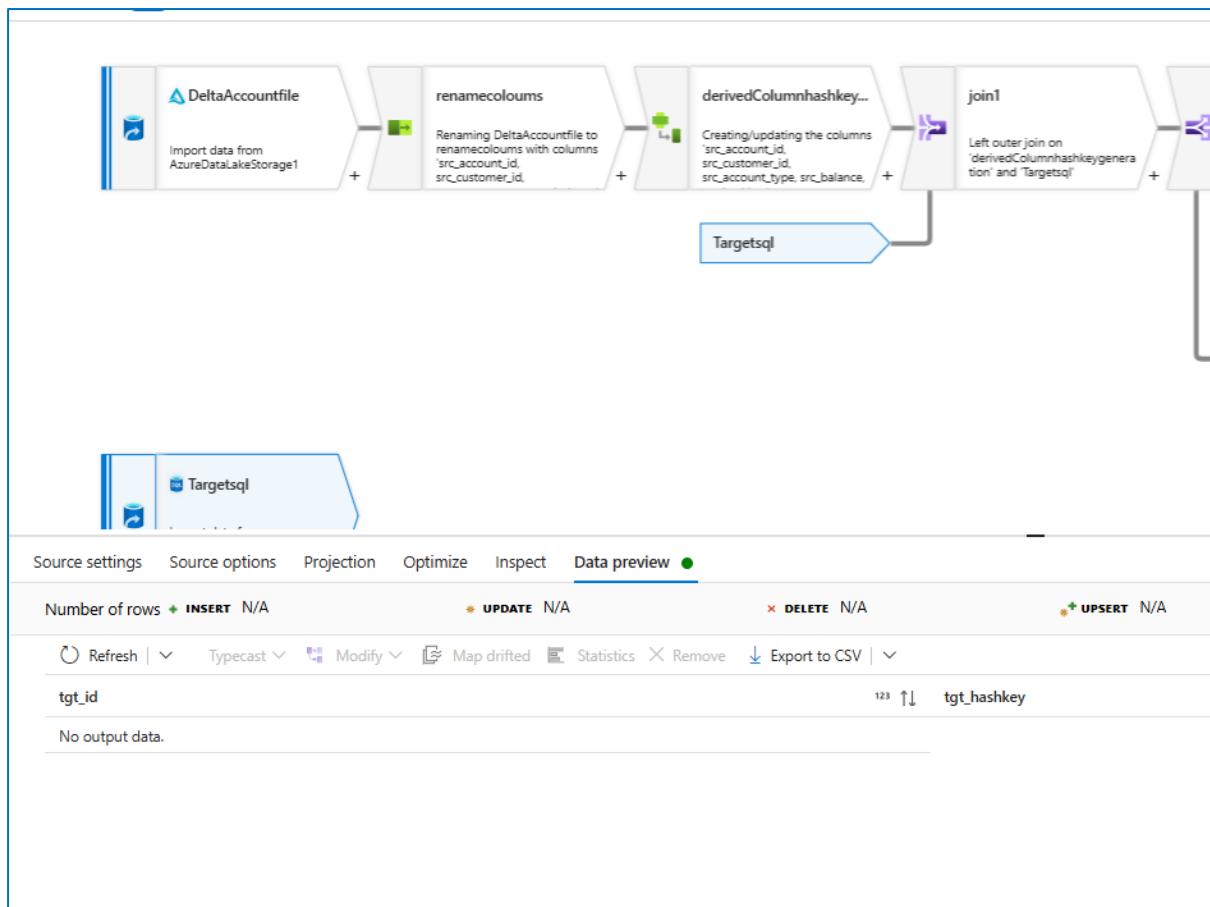
42
43
44 select account_id as tgt_id, HASHKEY as tgt_hashkey from DIM_Accounts
45
46
47
48
49
50

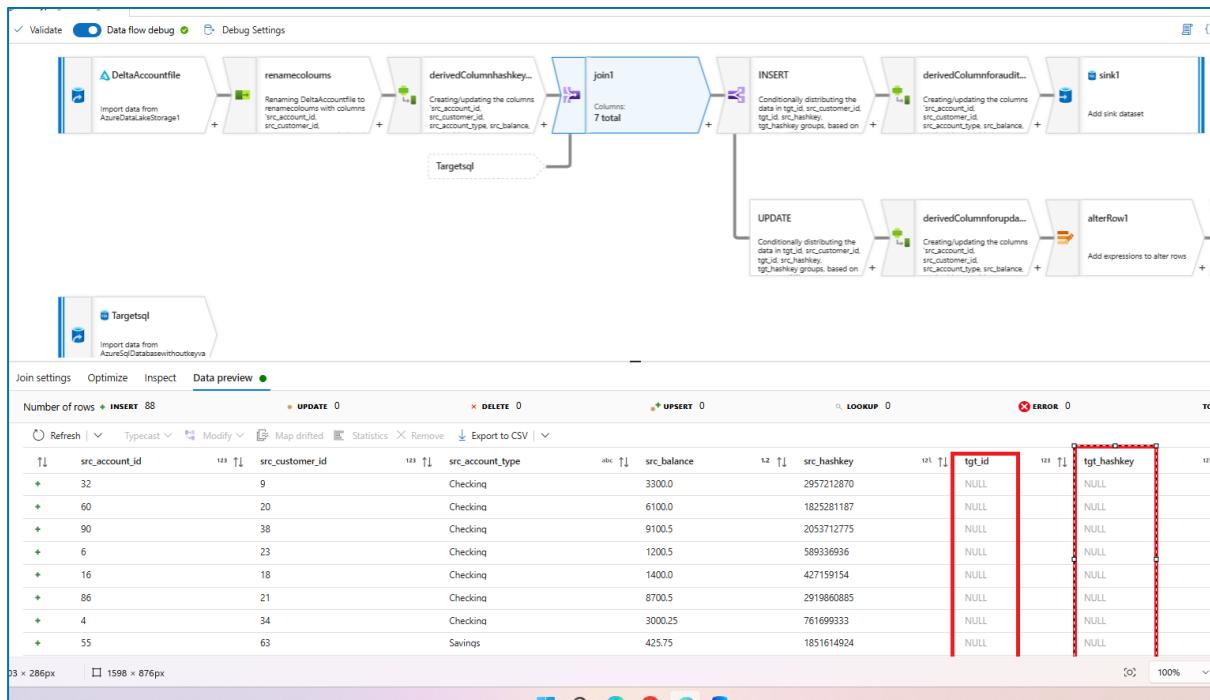
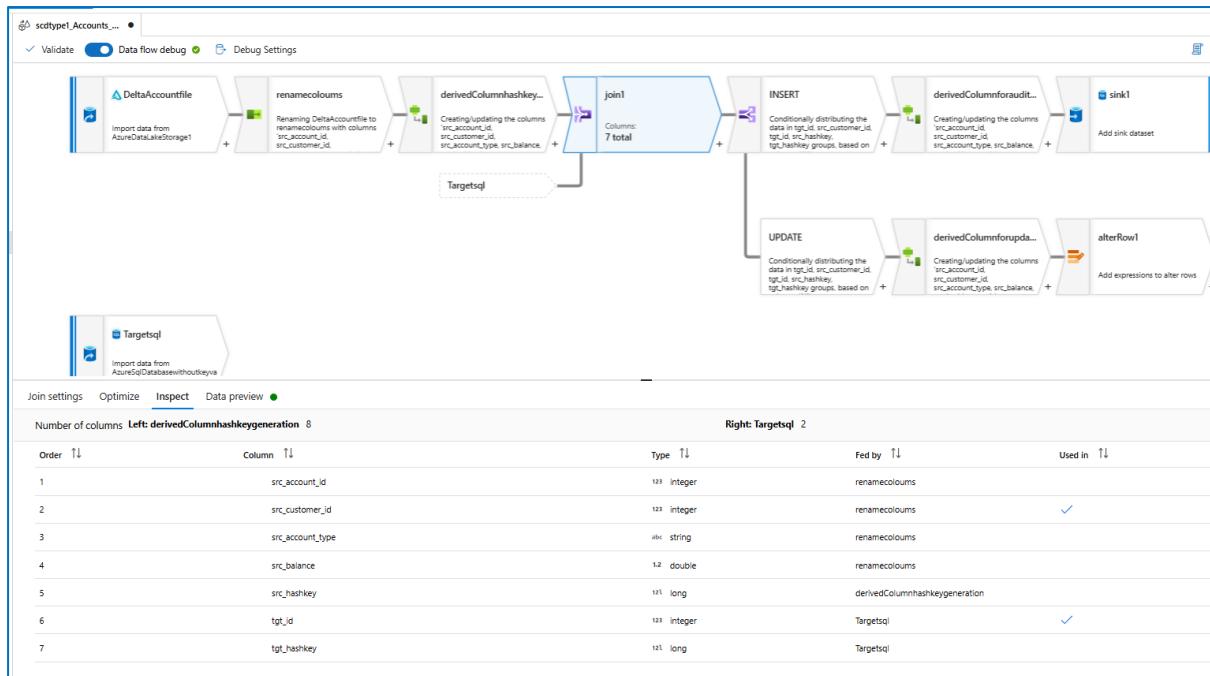
```

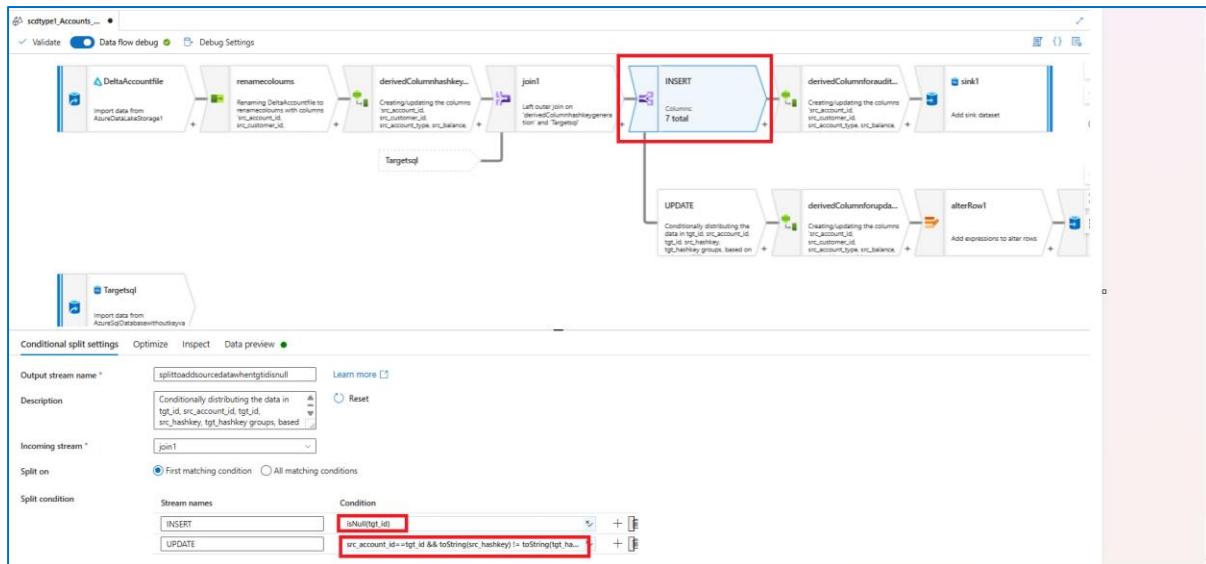
0% ✓ No issues found

Results	Messages
tgt_id tgt_hashkey	



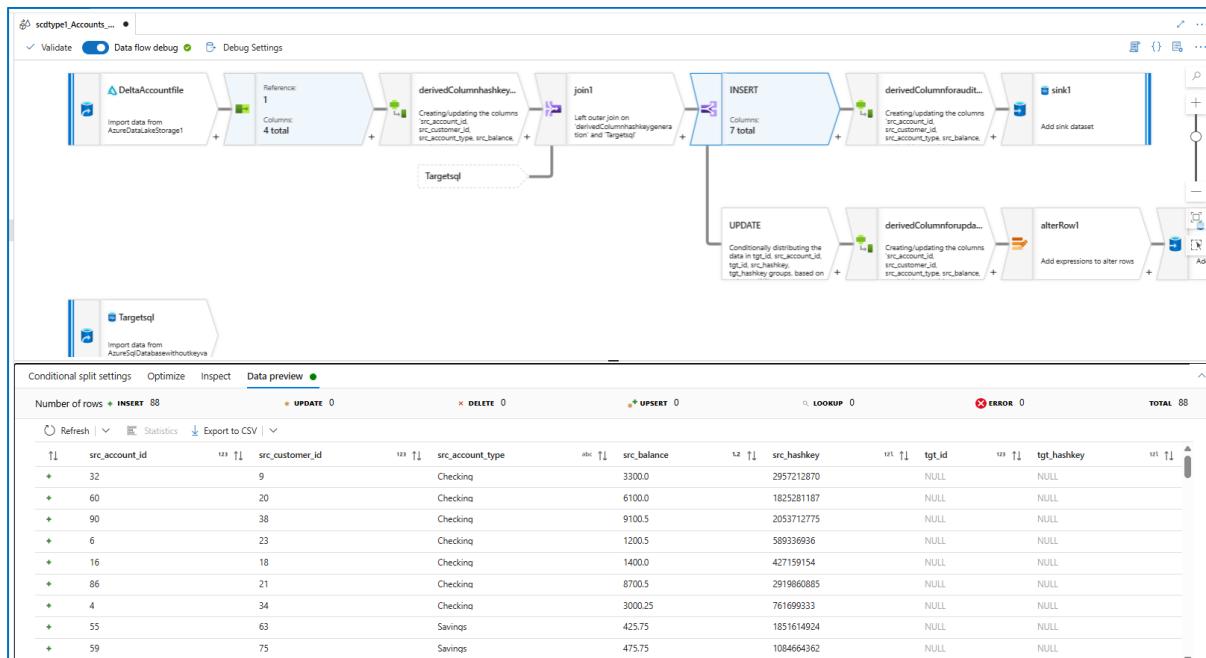




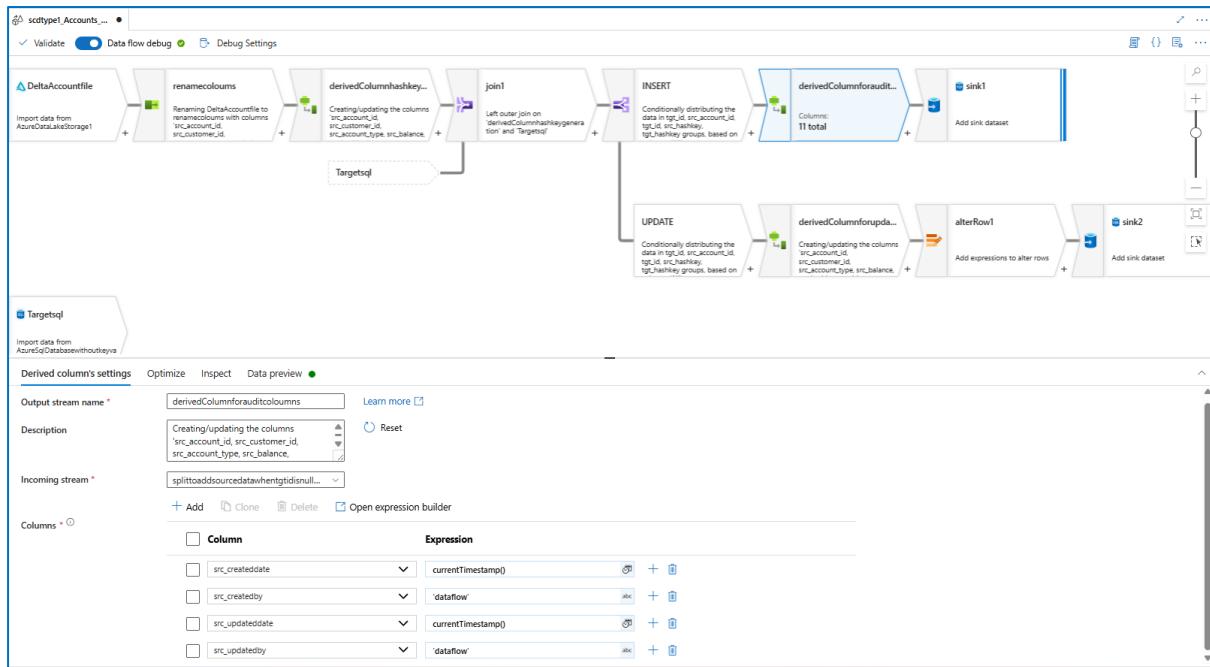


`src_account_id==tgt_id && toString(src_hashkey) != toString(tgt_hashkey)`

`isNull(tgt_id)`

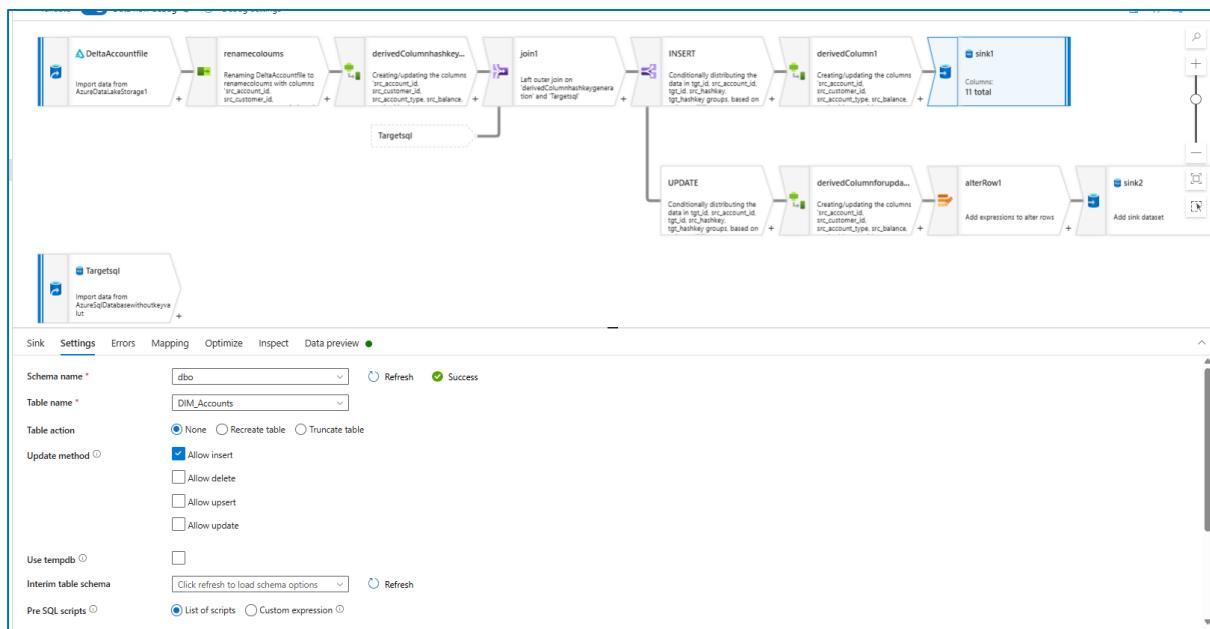
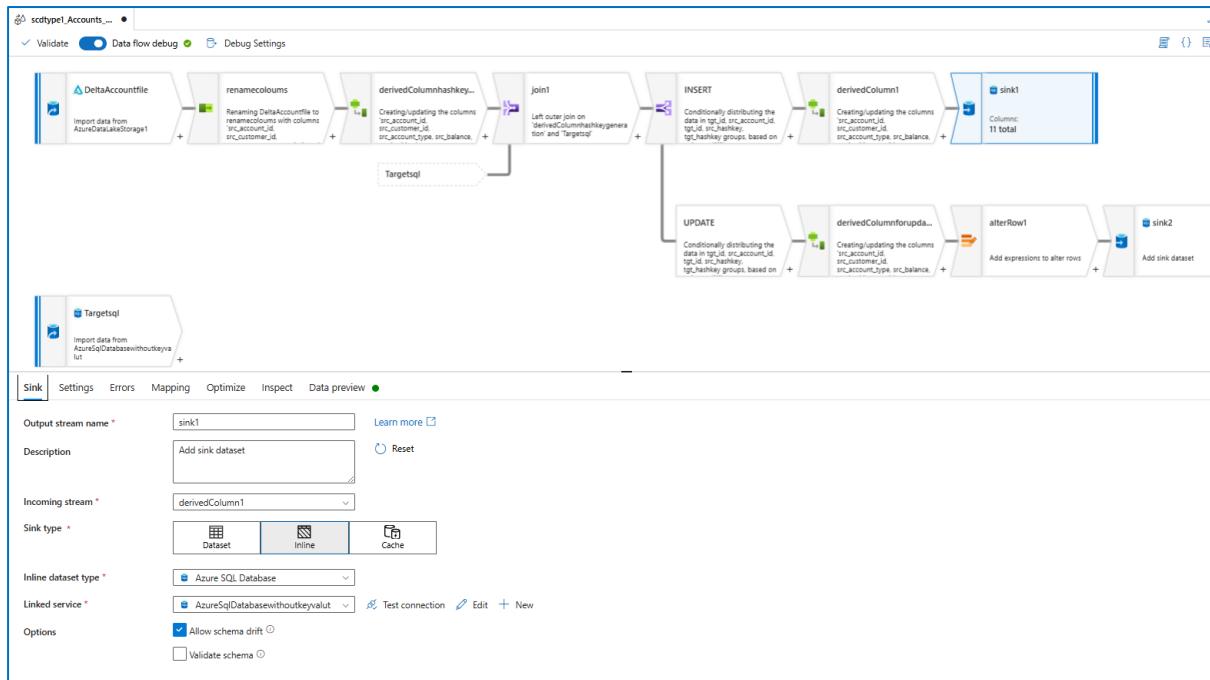


Adding Audit Columns for insert

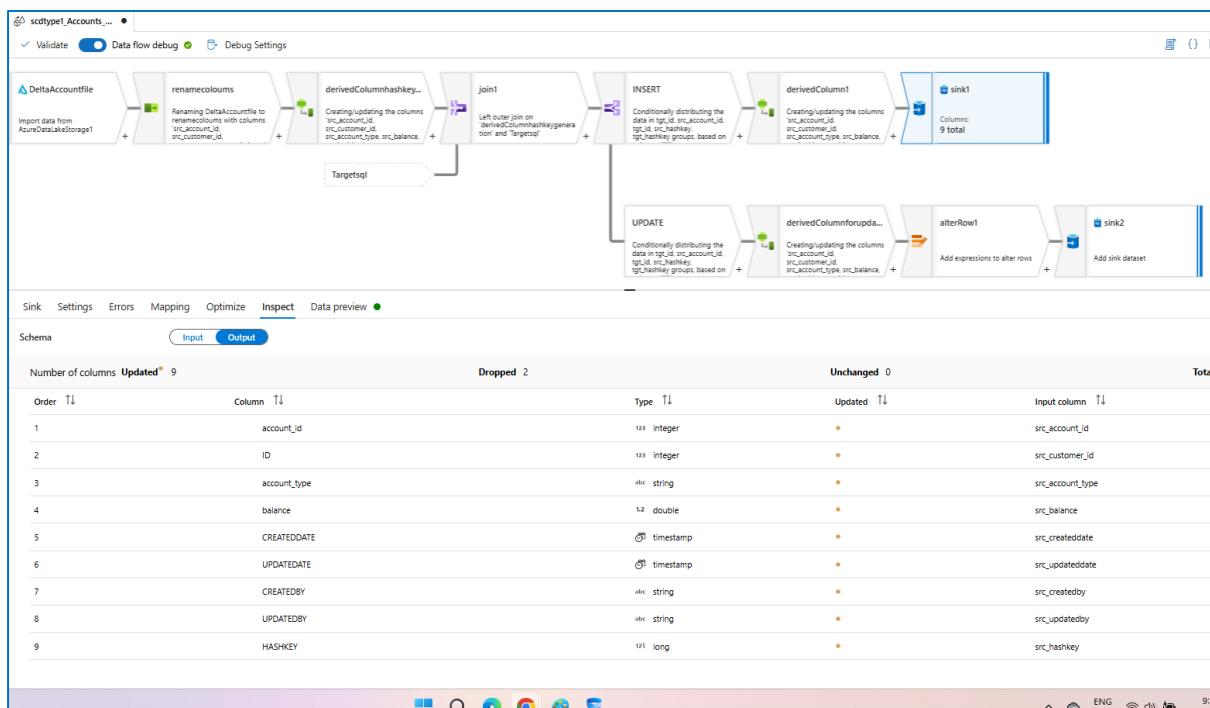
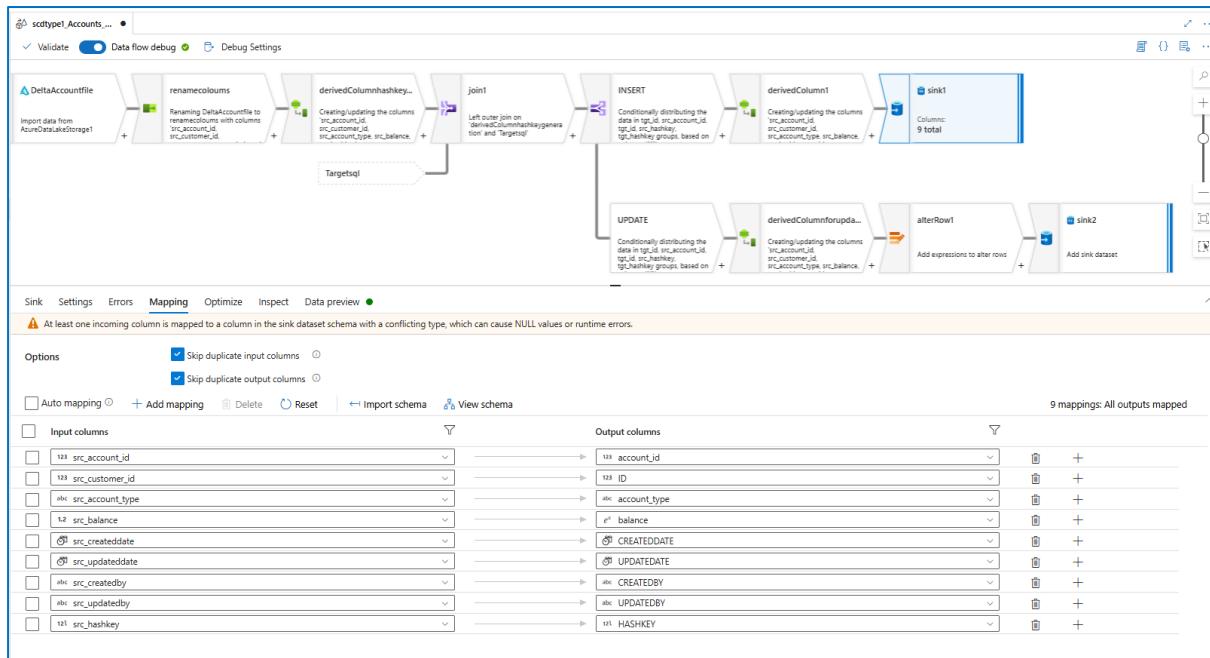


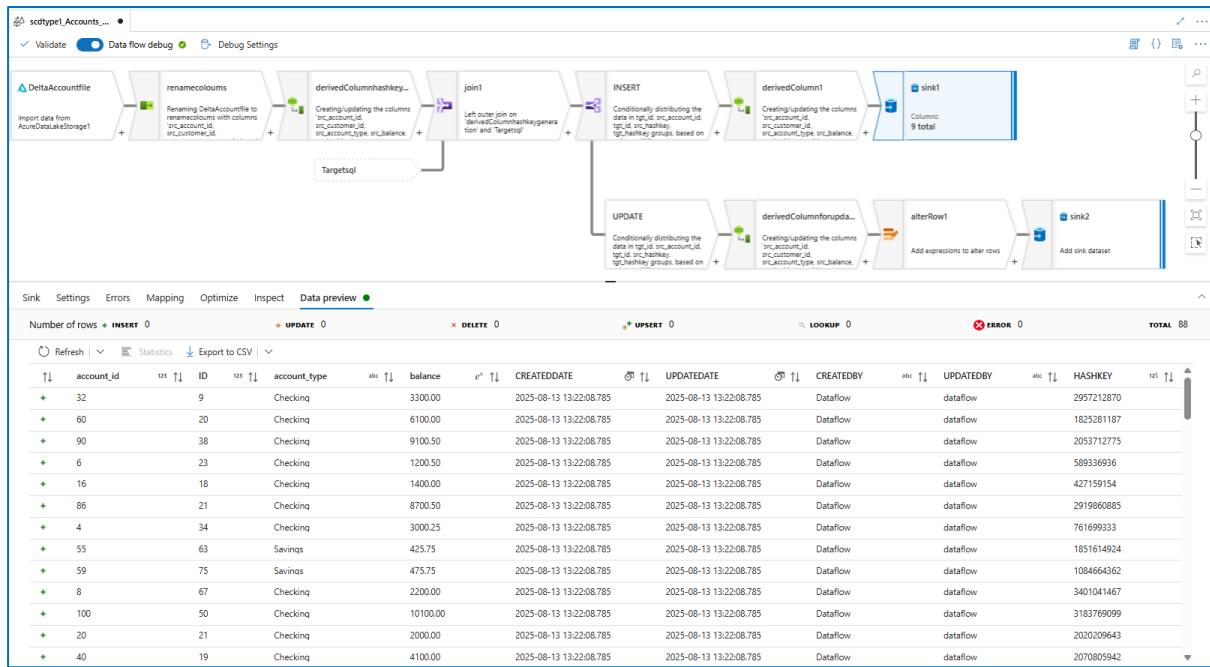
The screenshot shows the Azure Data Factory Data Flow blade with the 'Data preview' tab selected. The pipeline structure is identical to the first screenshot. The 'Data preview' tab displays 88 rows of sample data with the following schema:

	src_account_id	src_customer_id	src_account_type	src_balance	src_hashkey	tgt_id	tgt_hashkey	src_createddate	src_createdby	src_updateddate	src_updatedby
+	32	9	Checking	3300.0	2957212870	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	60	20	Checking	6100.0	1825261187	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	90	38	Checking	9100.5	2053712775	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	6	23	Checking	1200.5	589336936	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	16	18	Checking	1400.0	427159154	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	86	21	Checking	8700.5	2919860885	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	4	34	Checking	3000.25	761699333	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	55	63	Savings	425.75	1851614924	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	59	75	Savings	475.75	1054664362	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	8	67	Checking	2200.0	3401041467	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	100	50	Checking	10100.0	3183769099	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	20	21	Checking	2000.0	2020209643	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	40	19	Checking	4100.0	2070805942	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	25	66	Savings	100.25	2865279566	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	73	87	Savings	650.25	2825810413	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	15	47	Savings	700.75	1744154229	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	43	83	Savings	275.75	1235267699	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow
+	11	3	Savings	1100.75	744264830	NULL	NULL	2025-08-13 13:15:02.013	Dataflow	2025-08-13 13:15:02.013	Dataflow

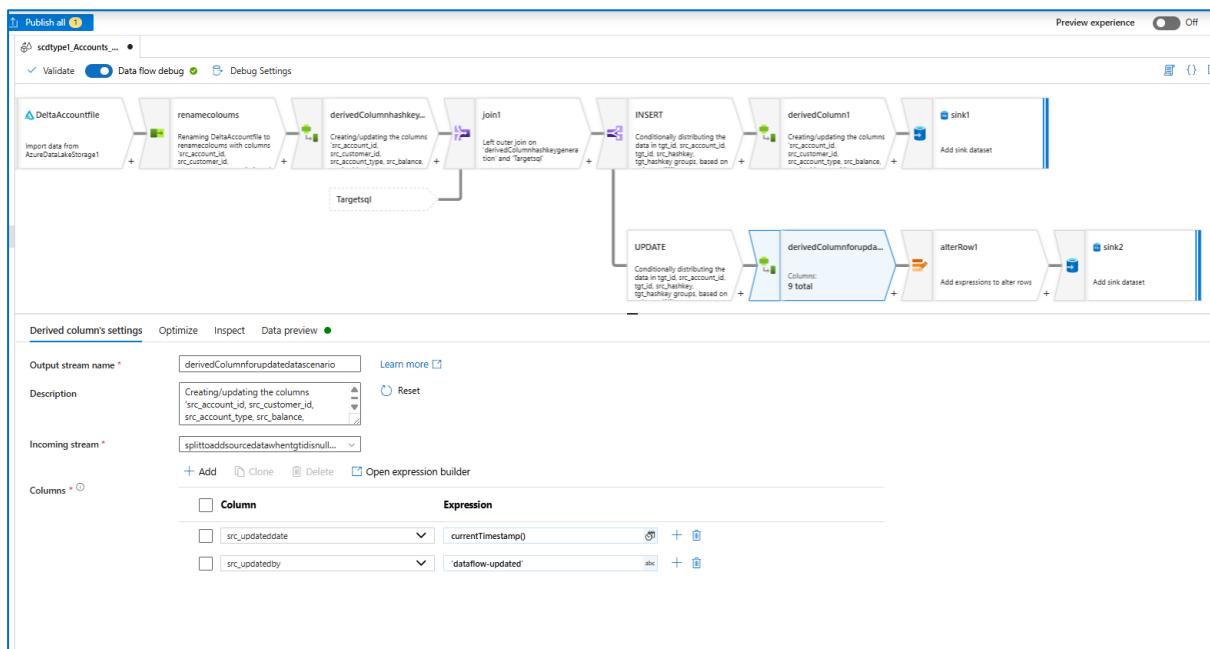


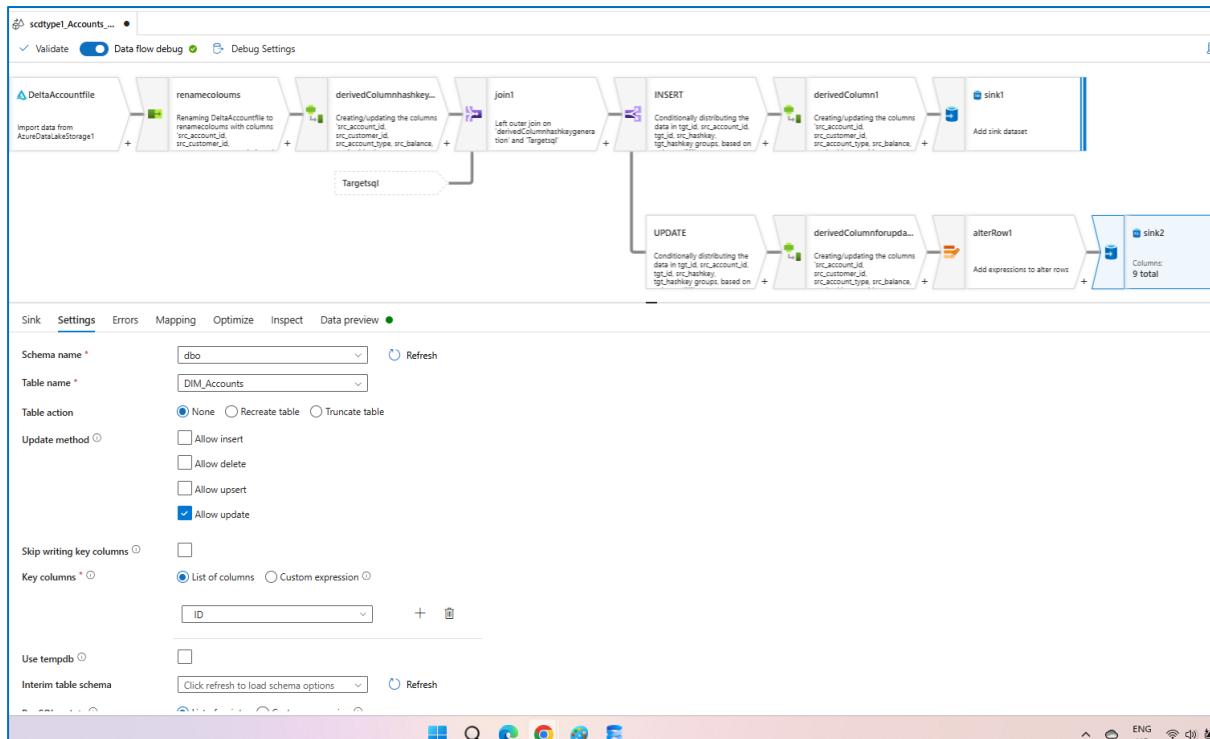
Remove automapping and reset then import schema



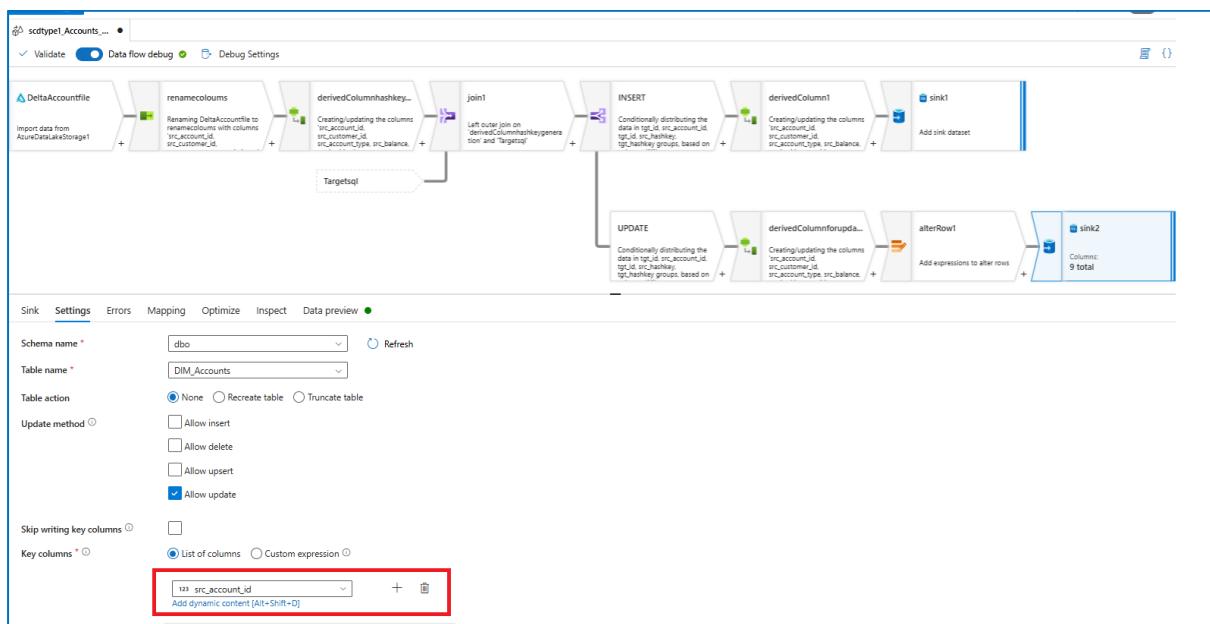


For update data sink

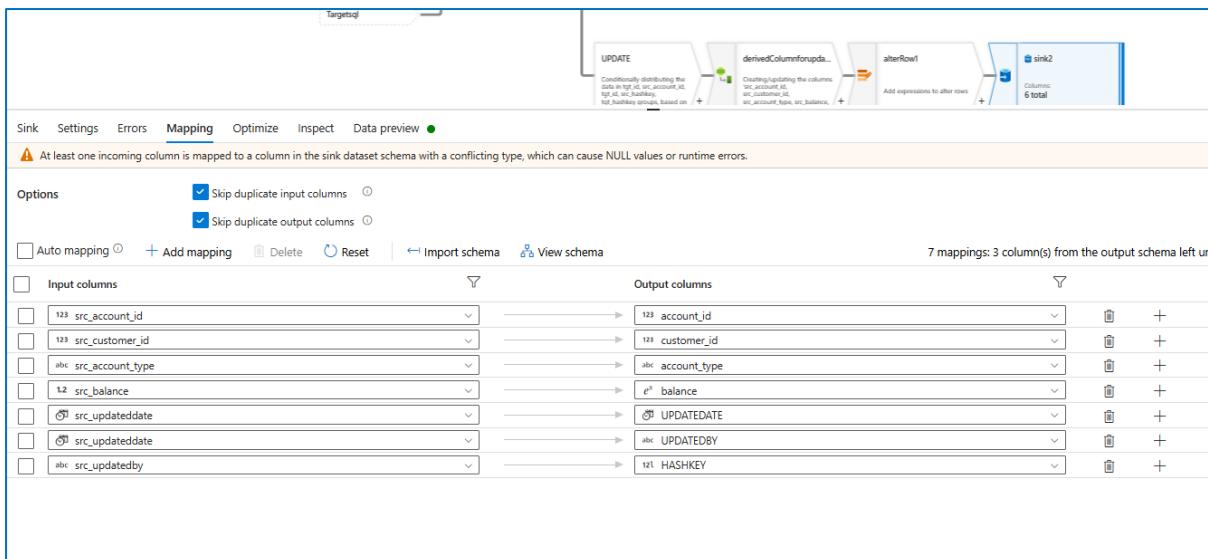




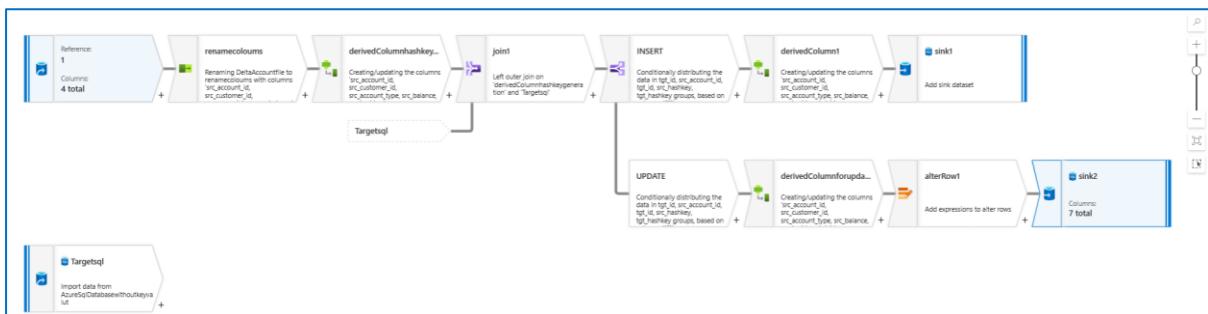
Based on account id we are updating



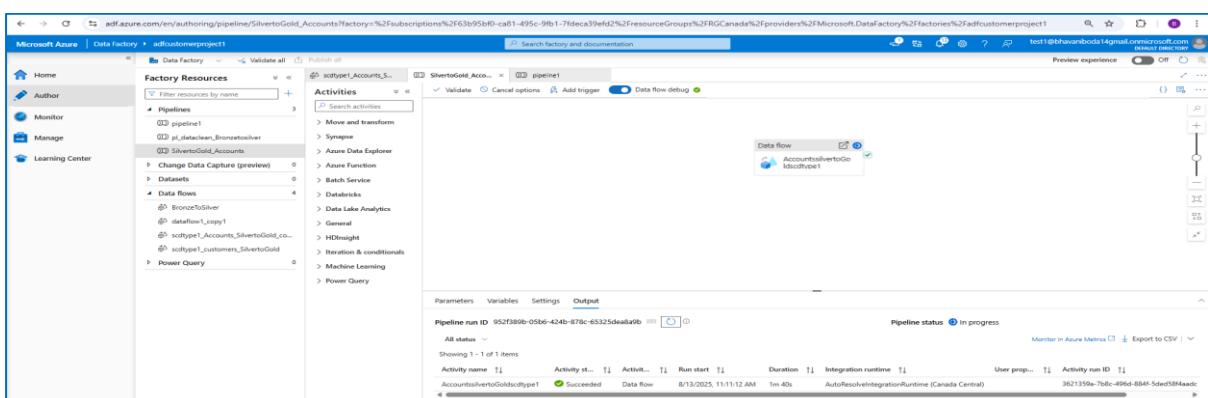
Mapping for updated coloumns



SCD Type1 accounts dataflow



Run the pipeline



```

S1 | select * from DIM_Accounts
S2 |
S3 |
S4 |
S5 |
S6 |
S7 |
S8 |

```

No issues found

	account_id	customer_id	account_type	balance	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
1	45	Savings	1000.50	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	2454403084	
2	12	Checking	2500.75	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	796571617	
3	78	Savings	1500.00	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	199654578	
4	34	Checking	3000.25	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	761699333	
5	56	Savings	500.00	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	3886787448	
6	23	Checking	1200.50	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	589336936	
8	67	Checking	2200.00	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	3401041467	
9	14	Savings	900.25	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	1397533172	
11	3	Savings	1100.75	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	744264830	
0	12	Checking	2700.00	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	4249714	
1	13	29	1300.25	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	3025925162	
2	14	64	3200.50	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	3349647104	
3	15	47	700.75	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	1744154229	
4	16	18	1400.00	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	427159154	
5	18	5	1600.50	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	54941373	
6	19	76	400.75	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	3392242716	
7	20	21	2000.00	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	2020209643	
8	21	53	300.25	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	Dataflow	2800704470	
9	22	37	Checking	2400.50	2025-08-13 13:41:49.867	2025-08-13 13:41:49.867	Dataflow	2771219456	

Query executed successfully.

bhavsqlserver.database.wi...

Perform update by editing some records in customer csv file

Before update record highlighted

bronze_raw/customers.csv	
Blob	
	Save Discard Download Refresh Delete
Overview	Versions
Edit	Generate SAS
56,Michael,Butler,5454 Ash Blvd,Holland Landing,ON,L9N0A1	
57,Elizabeth,Long,5555 Beech Dr,East Gwillimbury,ON,L9N0A1	
58,David,Patterson,5656 Cedar Ln,King City,ON,L7B0A1	
59,Sophia,Hughes,5757 Elm St,Nobleton,ON,L0G0A1	
60,John,Flores,5858 Maple Ave,Schomberg,ON,L0G0A1	
61,Olivia,Washington,5959 Oak Dr,Tottenham,ON,L0G0A1	
62,William,Butler,6060 Pine Rd,Alliston,ON,L9R0A1	
63,Ava,Simmons,6161 Birch Blvd,Angus,ON,L0M0A1	
64,Alexander,Foster,6262 Spruce Ln,Stayner,ON,L0M0A1	
65,Isabella,Gonzalez,6363 Fir St,Wasaga Beach,ON,L9Z0A1	
66,Daniel,Bryant,6464 Redwood Dr,Elmvale,ON,L0L0A1	
67,Sophia,Alexander,6565 Cypress Ave,Midland,ON,L4R0A1	
68,Matthew,Russell,6666 Willow Rd,Penetanguishene,ON,L9M0A1	
69,Charlotte,Griffin,6767 Poplar St,Victoria Harbour,ON,L0K0A1	
70,Joseph,Diaz,6868 Ash Blvd,Port McNicoll,ON,L0K0A1	
71,Amelia,Hayes,6969 Beech Dr,Waubaushene,ON,L0K0A1	
72,Christopher,Myers,7070 Cedar Ln,Coldwater,ON,L0K0A1	
73,Mia,Ford,7171 Elm St,Orillia,ON,L3V0A1	
74,Andrew,Hamilton,7272 Maple Ave,Gravenhurst,ON,P1P0A1	
75,Harper,Graham,7373 Oak Dr,Bala,ON,P0C0A1	
76,Joshua,Sullivan,7474 Pine Rd,Bracebridge,ON,P1L0A1	
77,Evelyn,Wallace,7575 Birch Blvd,Huntsville,ON,P1H0A1	
78,Daniel,Woods,7676 Spruce Ln,Burks Falls,ON,P0A0A1	
79,Abigail,Cole,7777 Fir St,Sundridge,ON,P0A0A1	
80,James,West,7878 Redwood Dr,South River,ON,P0A0A1	
81,Emily,Jordan,7979 Cypress Ave,North Bay,ON,P1B0A1	
82,Michael,Owens,8080 Willow Rd,Mattawa,ON,P0H0A1	
83,Elizabeth,Reynolds,8181 Poplar St,Sturgeon Falls,ON,P2B0A1	
84,David,Fisher,8282 Ash Blvd,Verner,ON,P0H0A1	
84,Sophia,Ellis,8383 Beech Dr,Field,ON,P0H0A1	
85,John,Harrison,8484 Cedar Ln,Temagami,ON,P0H0A1	
86,Olivia,Stil,Gibson,8585 Elm St,New Liskeard,ON,P0J0A1	
87,William,McDonald,8686 Maple Ave,Haileybury,ON,P0J0A1	

Changed the 86, 87 record and see if it is updating or not

Note: For inserting new record you have to change customer sink settings update method to insert then only insertion of new record possible

silvercontainercleandata						
Container		Overview				
Add Directory		Upload	Refresh	Delete	Copy	Paste
Search		Search resources, services, and docs (G+)	Copilot
Diagnose and solve problems		Add Directory	Upload	Refresh	Delete	Copy
Access Control (IAM)		Search blobs by prefix (case-sensitive)	Acquire lease	Break lease	Edit columns	Only show active objects
Settings		Sorting all 150 items (3 selected)				
<input type="checkbox"/> Name		Last modified ↓	Access tier	Blob type	Size	Lease state
<input type="checkbox"/> H						
<input checked="" type="checkbox"/> part-00000-8e5b0bc2-ab53-490a-a53f-0428d232d6af-c000.snappy.parquet		8/14/2025, 7:18:20 PM	Hot (inferred)	Block blob	5.58 KIB	Available
<input checked="" type="checkbox"/> part-00000-fbcb4c44-8c14-471e-85ca-8754ef1f01cf-c000.snappy.parquet		8/14/2025, 7:18:20 PM	Hot (inferred)	Block blob	838 B	Available
<input checked="" type="checkbox"/> part-00000-5ab8de37-380c-4538-a568-89792c80ac47-c000.snappy.parquet		8/14/2025, 7:10:16 PM	Hot (inferred)	Block blob	5.58 KIB	Available
<input type="checkbox"/> part-00000-905bb87e-b830-4b16-965d-a977fc135b-c000.snappy.parquet		8/14/2025, 7:10:14 PM	Hot (inferred)	Block blob	838 B	Available
<input type="checkbox"/> part-00168-1569fa2e-9d51-40b1-af2d-1ad13b198130-c000.snappy.parquet		8/13/2025, 7:14:15 AM	Hot (inferred)	Block blob	2 KIB	Available
<input type="checkbox"/> part-00173-fb8bede-ad95-4a9f-9ac4-e9c69c0ebe09-c000.snappy.parquet		8/13/2025, 7:14:15 AM	Hot (inferred)	Block blob	2.05 KIB	Available

Next run scd type1 for customers and see the data is updated or not

Name is updated for 86 record as modified

```

53
54
55
56
57      Select * from DIM_CUSTOMER where ID=86
58
59
60
61
62

```

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) window. A red box highlights the results grid, which displays a single row of data from the DIM_CUSTOMER table. The columns are ID, NAME, CITY, CREATEDDATE, UPDATEDATE, CREATEDBY, UPDATEDBY, and HASHKEY. The data for the first row is: ID=86, NAME=OliviaST, CITY>New Liskeard, CREATEDDATE=2025-08-12 00:08:17.177, UPDATEDATE=2025-08-14 23:21:50.513, CREATEDBY=dataflow, UPDATEDBY=dataflow-updated, HASHKEY=321571388.

The screenshot shows the Azure Storage Explorer interface. A red box highlights the 'customers.csv' file under the 'bronze_raw/customers' folder. The file contains a large list of customer records in CSV format. The first few lines of the data are:

```

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,789 Oak Dr,Montréal,QC,H1A1A1
5 4,Emily,Davis,101 Pine Rd,Calgary,AB,T2B0A1
6 5,David,Wilson,202 Birch Blvd,Vancouver,BC,V5K0A1
7 6,Emma,Parkinson,345 Cedar Ln,Vancouver,BC,V5K0A1
8 7,Olivia,Marquez,123 Spruce Ln,Edmonton,AB,T5K0A1
9 8,Olivia,Garcia,707 Fir St,Edmonton,AB,T5A0A1
10 9,William,Lopez,888 Redwood Dr,Victoria,BC,V8W0A1
11 10,Ava,Anderson,999 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,Matthew,Johnson,1234 Elm St,Markham,ON,L3R0A1
15 14,Sophia,Young,1233 Beech Dr,Yellowknife,NT,X1A0A1
16 15,Matthew,Xing,1214 Cedar Ln,Whitehorse,YT,Y1A0A1
17 16,Charlotte,Scott,1515 Elm St,Calgary,AB,T2B0A1
18 17,Joseph,Green,1016 Maple Ave,Regina,SK,S4P0A1
19 18,Amelia,Adams,1717 Oak Dr,Saskatoon,SK,S7K0A1
20 19,Christopher,Miller,1818 Pine St,Victoria Bay,ON,P7A0A1
21 20,James,Anderson,1999 Elm St,St. John's,NL,A1A0A1
22 21,Andrew,Mitchell,2000 Spruce Ln,Hamilton,ON,L8P0A1
23 22,Hanper,Roberts,2121 Fir St,Kitchener,ON,N2B0A1
24 23,Joshua,Turner,2222 Redwood Dr,Windsor,ON,N9A0A1
25 24,Evelyn,Phillips,2323 Cypress Ave,Kingston,ON,K7L0A1
26 25,Daniel,Campbell,2424 Willow Rd,St. Catharines,ON,L2R0A1
27 26,Abigail,Johnson,2525 Elm St,Brampton,ON,L6R0A1
28 27,George,Evans,2626 Ash Blvd,Waterloo,ON,N2L0A1
29 28,Emily,Edwards,2727 Beech Dr,Brantford,ON,N3T0A1
30 29,Michael,Collins,2828 Cedar Ln,Thunder Bay,ON,P7B0A1
31 30,Elizabeth,Stewart,2929 Elm St,Peterborough,ON,K9H0A1
32 31,David,Sanchez,3030 Maple Ave,North Bay,ON,P1B0A1
33 32,Sophia,Morris,3111 Oak Dr,Belleville,ON,K9H0A1

```

bronze_raw/customers.csv	
Blob	
	Save Discard Download Refresh Delete
Overview	Versions
	Edit Generate SAS
1	customer_id,first_name,last_name,address,city,state,zip
2	1,John,Doe,123 Elm St,Milton,ON,M4B1B3
3	2,Jane,Smith,456 Maple Ave,Ottawa,ON,K1A0B1
4	3,Michael,Johnson,789 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
20	19,Christopher,Baker,1818 Pine Rd,Thunder Bay,ON,P7A0A1
21	20,Mia,Nelson,1919 Birch Blvd,London,ON,N6A0A1
22	21,Andrew,Mitchell,2020 Spruce Ln,Hamilton,ON,L8P0A1
23	22,Harper,Roberts,2121 Fir St,Kitchener,ON,N2G0A1
24	23,Joshua,Turner,2222 Redwood Dr,Windsor,ON,N9A0A1
25	24,Evelyn,Phillips,2323 Cypress Ave,Kingston,ON,K7L0A1
26	25,Daniel,Campbell,2424 Willow Rd,St. Catharines,ON,L2R0A1
27	26,Abigail,Parker,2525 Poplar St,Barrie,ON,L4M0A1
28	27,James,Evans,2626 Ash Blvd,Guelph,ON,N1H0A1
29	28,Emily,Edwards,2727 Beech Dr,Brantford,ON,N3T0A1
30	29,Michael,Collins,2828 Cedar Ln,Thunder Bay,ON,P7B0A1
31	30,Elizabeth,Stewart,2929 Elm St,Peterborough,ON,K9H0A1
32	31,David,Sanchez,3030 Maple Ave,North Bay,ON,P1B0A1
33	32,Sophia,Morris,3131 Oak Dr,Belleville,ON,K8N0A1

Csv

Preview

Data is updated in ssms

50							
51							
52							
53							
54							
55							
56							
57	select * from DIM_CUSTOMER						
58							
59							
60							
61							
62							
63							
	x 3 ▲ 0 ↑ ↓						
Results	Messages						
ID	NAME	CITY	CREATEDDATE	UPDATEDATE	CREATEDBY	UPDATEDBY	HASHKEY
31	David	North Bay	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	1611955277
85	John	Temagami	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	1599281831
65	Daniel	Elmvale	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2240443872
53	James	Queensville	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	3491966471
78	Abigail	Sundridge	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	3332988847
34	Olivia	Orrilla	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2844006823
28	Emily	Brantford	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	314608446
26	Abigail	Barn	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2907621863
27	James	Guelph	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2772220524
44	Amelia	Bradford	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	447506045
47	Andrew	Uxbridge	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2520758040
1	John	Milton	2025-08-12 00:08:17.177	2025-08-14 23:31:47.033	dataflow	dataflow-updated	3002656611
52	Abigail	Mount Albert	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	569556118
3	Michael	Montreal	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	4114944968
20	Mia	London	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	1253444537
40	Sophia	Milton	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2044068006
48	Harper	Port Perry	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	1193783485
5	David	Vancouver	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	3284019540
15	Matthew	Whitehorse	2025-08-12 00:08:17.177	2025-08-12 00:08:17.177	dataflow	dataflow	2616855931

Eros

I havent executed select ID as tgt_id, HASHKEY as tgt_hashkey from DIM_CUSTOMER

This has to execute in the target using query

Source settings **Source options** Projection Optimize Inspect Data preview ●

Input Table Query Stored procedure

Query * ⓘ

```
select ID as tgt_id, HASHKEY as  
tgt_hashkey from DIM_CUSTOMER
```

Add dynamic content [Alt+Shift+D]

Incremental column ⓘ

Isolation level ⓘ Read uncommitted

Targetsqli

Columns: 2 total

+

Source settings Source options Projection Optimize Inspect **Data preview** ●

Number of rows **INSERT** N/A **UPDATE** N/A **DELETE** N/A **UPSERT** N/A

Refresh | Typecast | Modify | Map drifted | Statistics | Remove | Export to CSV |

tgt_id	tgt_hashkey
123	↑↓

No output data.

So i couldnt able to see the tgt_id at the join coloumn join condition

The screenshot shows the 'Join settings' page in the Azure Data Factory interface. The 'Join type' is set to 'Left outer'. The 'Left stream' is 'derivedColumnhashkeygeneration' and the 'Right stream' is 'Targetsqli'. The 'Join conditions' section shows a comparison between 'src_customer_id' (Left) and 'tgt_id' (Right) using the '=' operator.

Error2:

I got type mismatch issue in conditional split to perform insert and upsert operation at src hashkey as it is string and tgt hashkey is long

The screenshot shows the 'Dataflow expression builder' interface. The expression being built is 'src_customer_id==tgt_id && src_hashkey != tgt_hashkey'. A tooltip indicates that the expression's type ('long') does not match the required type ('string'). The 'Expression elements' sidebar lists various variables and functions, and the 'Data preview' section shows a sample of data with columns 'src_customer_id' and 'tgt_id'.

So i converted tgt hashkey to toString then it is resolved

```
src_customer_id==tgt_id && src_hashkey != toString(tgt_hashkey)
```

Dataflow expression builder

Expression

```
src_customer_id == tgt_id || src_hashkey == toString(tgt_hashkey)
```

Expression elements

Expression values

	src_customer_id	tgt_id	src_hashkey	tgt_hashkey
NULL	68	NULL	68CharlotteGriffin6767 Pop...	NULL
NULL	71	NULL	71ChristopherMyers7070 C...	NULL
NULL	82	NULL	82ElizabethReynolds8181 P...	NULL
NULL	62	NULL	62AinslieHenderson6151 C...	NULL

Data preview

Output:

src_customer_id	tgt_id	src_hashkey	tgt_hashkey
NULL	68	NULL	68CharlotteGriffin6767 Pop...
NULL	71	NULL	71ChristopherMyers7070 C...
NULL	82	NULL	82ElizabethReynolds8181 P...
NULL	62	NULL	62AinslieHenderson6151 C...

Save and finish Cancel Clear contents

Error 3: i just want to use only numeric values for primary key but this expression getting error

```
isNull(customer_id) || trim(toString(customer_id)) == '' || 
(toString(customer_id) =~ ^\d+$)
```

Dataflow expression builder

Expression

```
isNull(customer_id) || trim(toString(customer_id)) == '' || !(toString(customer_id) =~ ^\d+$)
```

Expression elements

Expression values

	customer_id	first_name	last_name	address	city
x					
x					
x					
x					

Data preview

Output:

customer_id	first_name	last_name	address	city
x				
x				
x				
x				

Save and finish Cancel Clear contents

Microsoft Azure | Data Factory > adfcustomerproject1

Dataflow expression builder

derivedColumn1

Derived Columns

+ Create new <

customer_id

Column name *

customer_id

Expression

`isNotNull(customer_id) || trim(toString(customer_id)) = '' || (customer_id == '^^(0-9)+$')`

✖ 28 1 of 2 problems

Unable to parse the expression. Please make sure it is valid.

Expression elements

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

Expression values

Filter by keyword

+ Create new <

customer_id

first_name

last_name

address

city

Data preview

Refresh

Error:

Cannot read properties of undefined (reading 'length')

Save and finish Cancel Clear contents

Erroe4:At scdtype source loading delta file i got error and after removing temporary folder in storage account delta folder it was resolved

factory > adfcustomerproject1

Validate all Publish all

pl_dataclean pipeline1 scdtyped_customers...

DeltaCustomerFile

Columns: 0 total

Add Source

Notifications

Dismiss all

DeltaCustomerFile

Import schema failed.
Run ID: 642cf0b-2ae5-4fd1-b8c6-24ebada471.
View details
8 minutes ago

DeltaCustomerFile

Import schema failed.
Run ID: cc2ec001-6869-41be-90dd-2940eb5d742.
View details
9 minutes ago

Data flow debug

Failed to fetch data preview.
Run ID: b1da260d-3b07-4d84-bd42-1d8d29a366be.
View details
10 minutes ago

ad.azure.com/en/authoring/dataflow?factory=%2Fsubscriptions%2F63b95bf0-ca81-495c-9fb1-7fdeca39ef2%2FresourceGroups%2FRGCanada%2Fproviders%2FMicrosoft.DataFactory%2Ffactories%2Fadfcustomerproject1

Microsoft Azure | Data Factory > adfcustomerproject1

Data Factory Validate all Publish all

pl_dataclean pipeline1 scdtyped_customers...

DeltaCustomerFile

Columns: 0 total

Add Source

Error details

Error code

DF-Executor-InvalidPath Troubleshooting guide

Activity ID

642cf0b-2ae5-4fd1-b8c6-24ebada471

Details

at source DeltaCustomerFile Path
abfss://silvercontainercleaneddata@adigen2storagebsu.dfs.core.windows.net/customers/Delta/_delta does not resolve to any files. Please make sure the file/folder exists and is not hidden. All the same if special character is not included in file/folder name, for example, name starting with ..