Bootcamp Project – 1

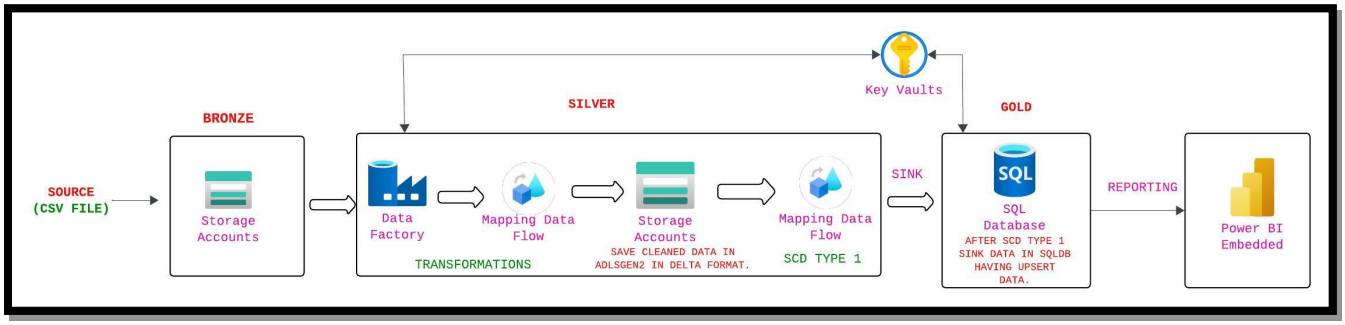
Data Pipeline for Customer Account Analysis

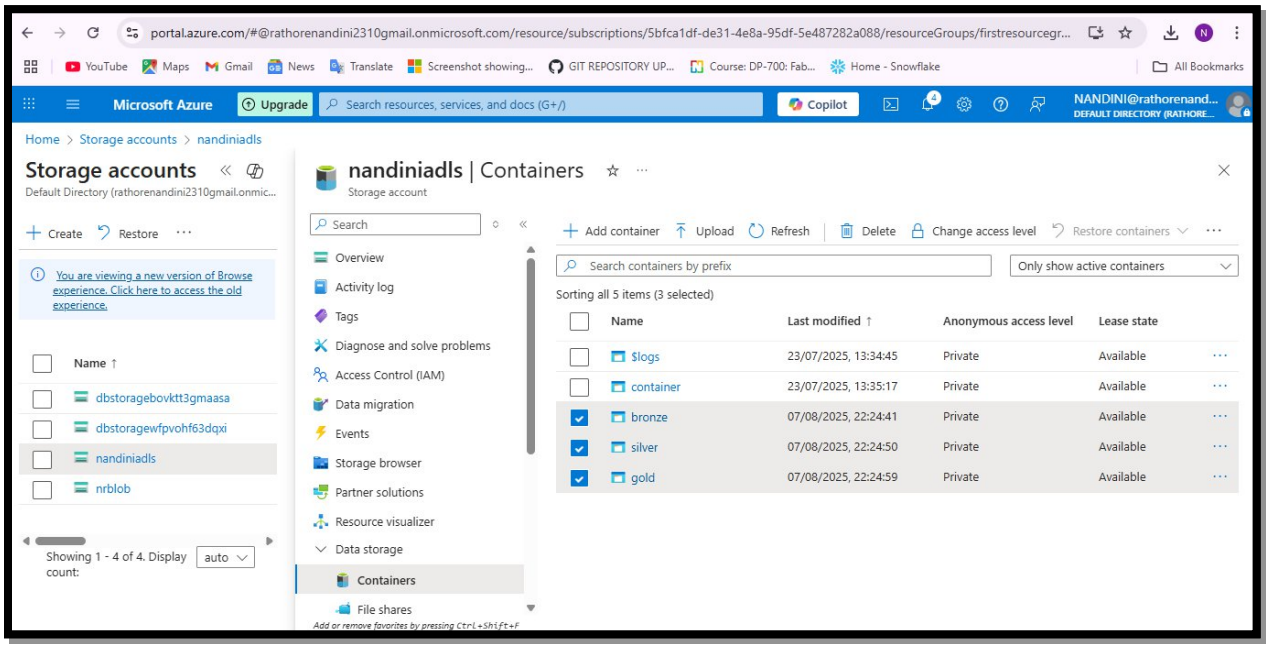
Objective

The project aims to design and implement a robust data pipeline for processing customer account data. This includes copying data from a backend team's storage account, performing necessary transformations using datafactory, and upserting (inserting or updating) (SCD-1) data from a file stored in Azure Data Lake Storage (ADLS) GOLD Storage into a SQL DB. The pipeline aims to ensure efficient, accurate, and scalable data processing to support downstream analytics and reporting needs.

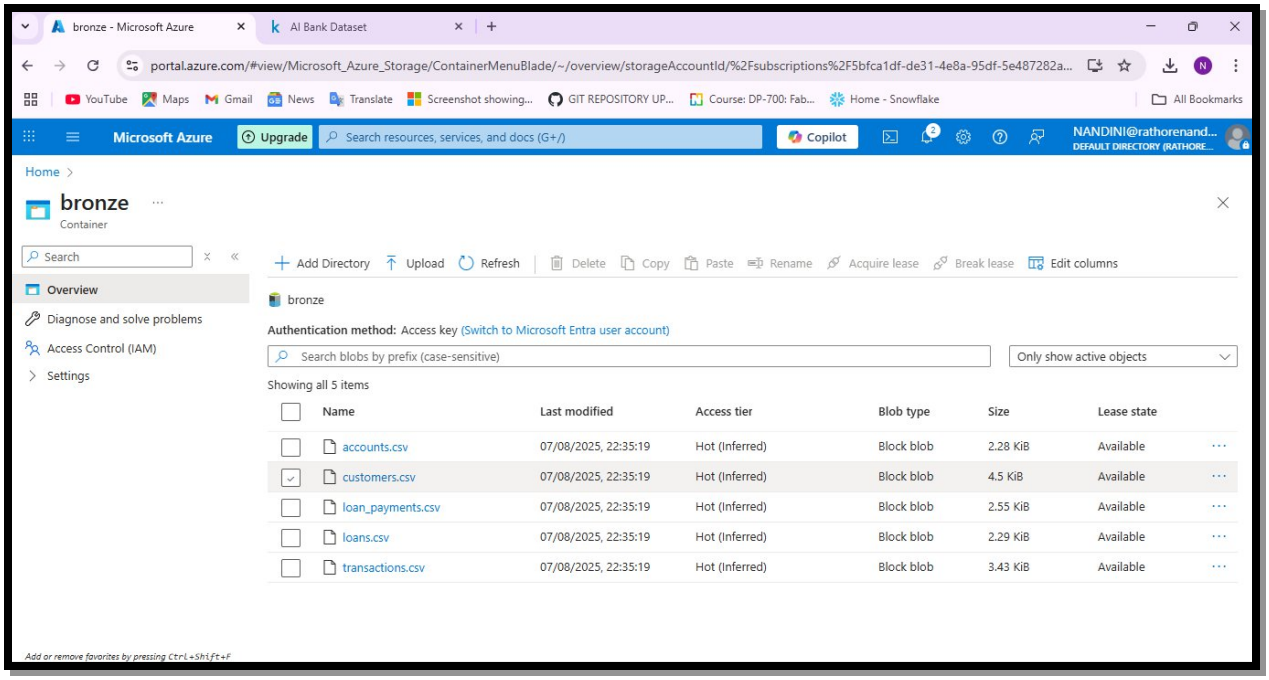
NANDINI RATHORE

ARCHITECTURE OF PROJECTà

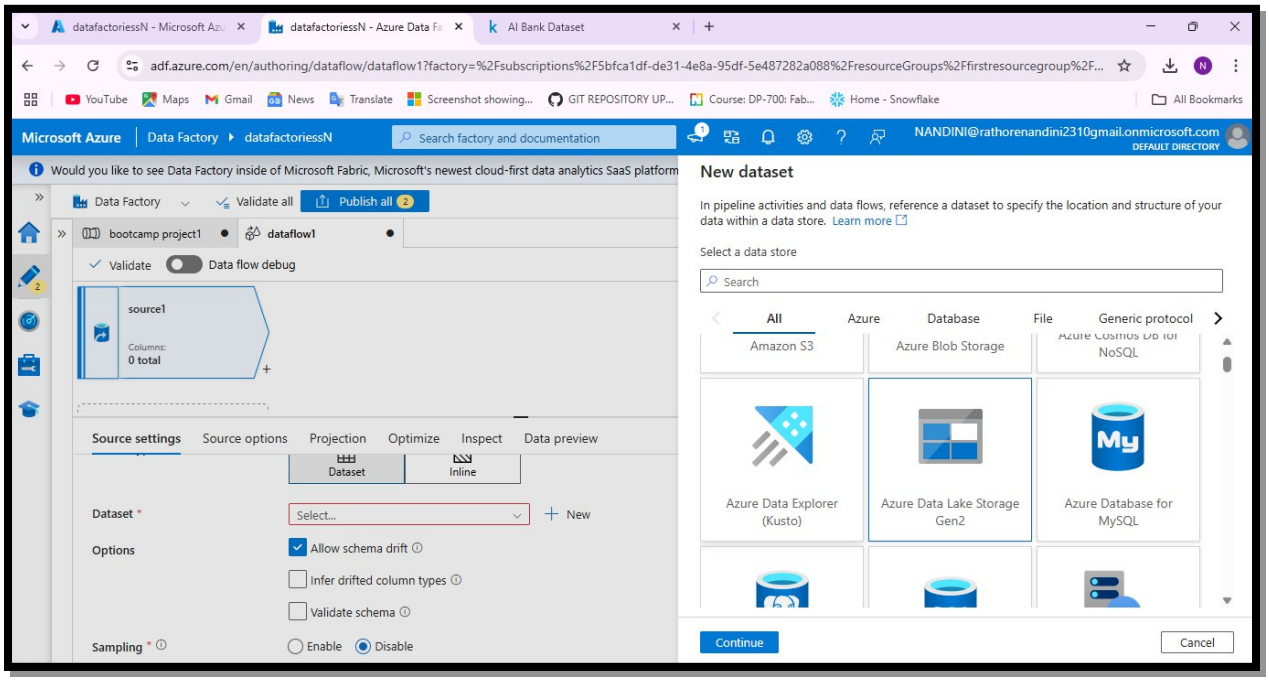




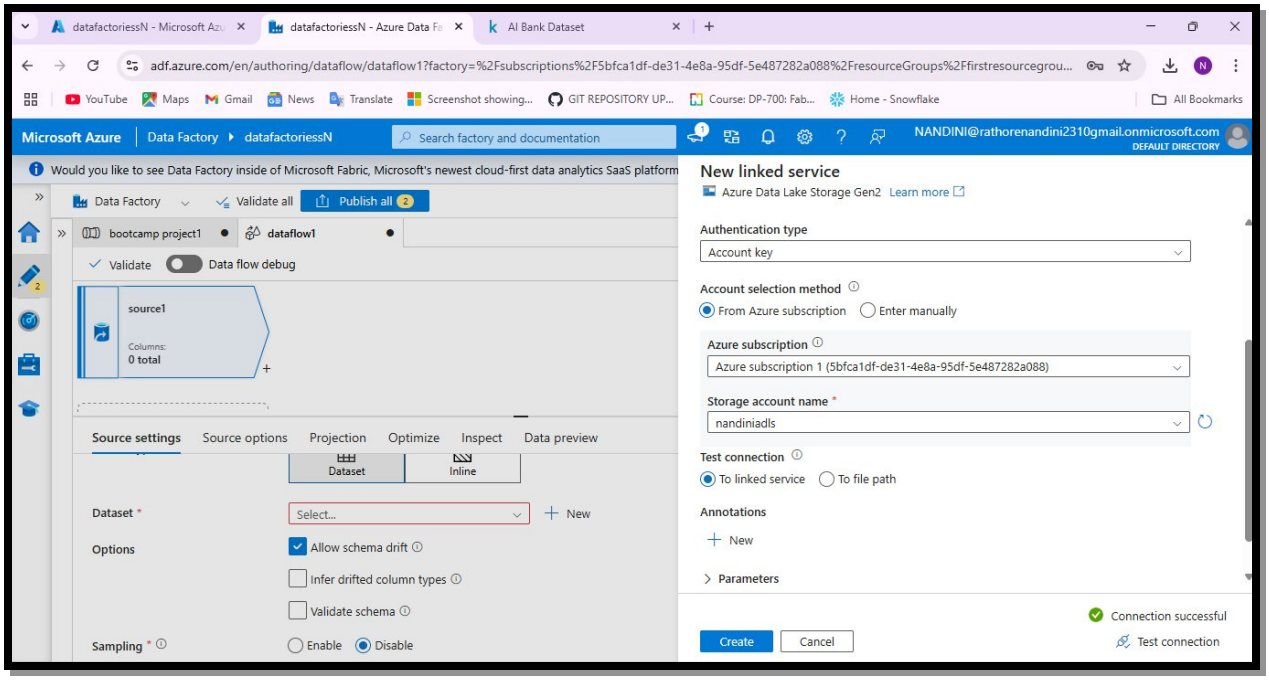
Creating 3 container in adlsgen2.-->bronze,silver,gold.

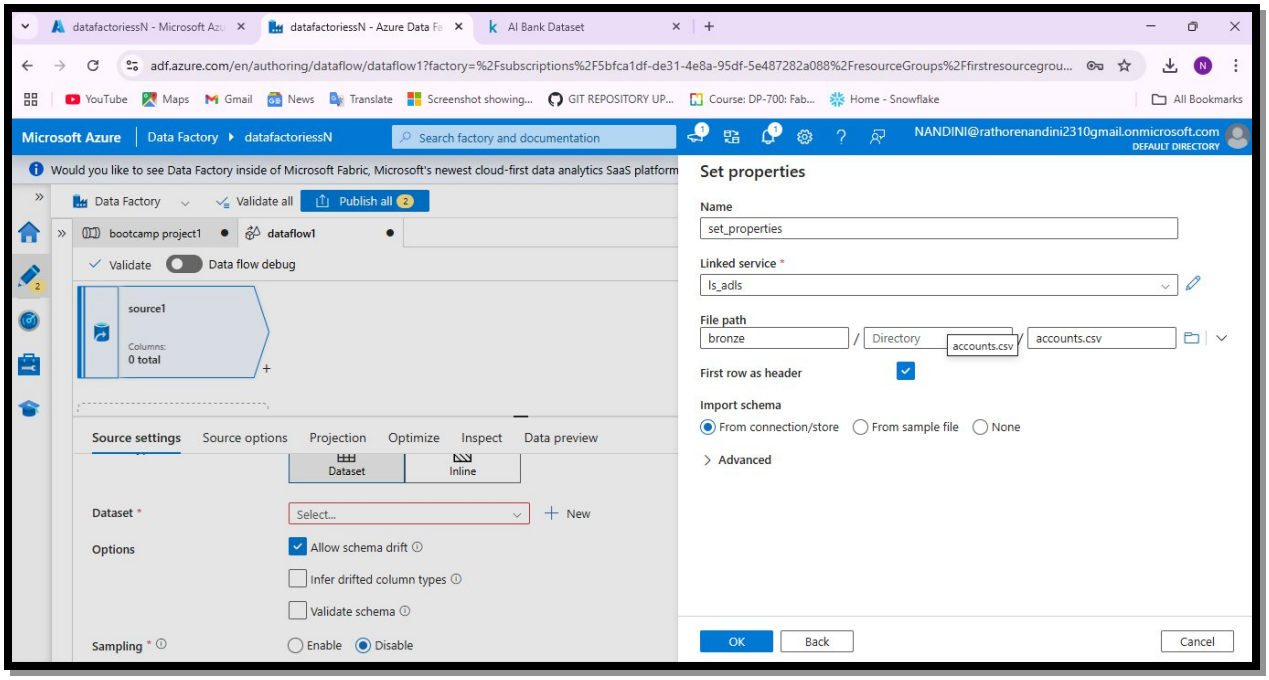


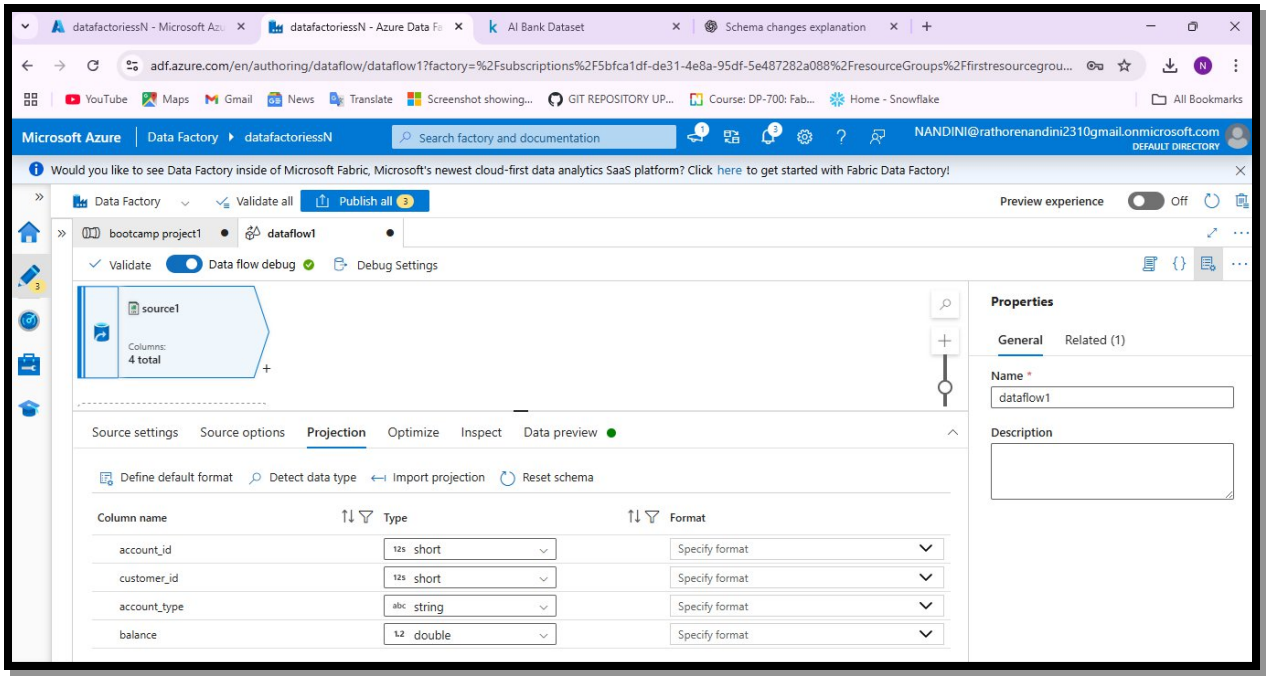
Ingesting raw data in bronze layer.

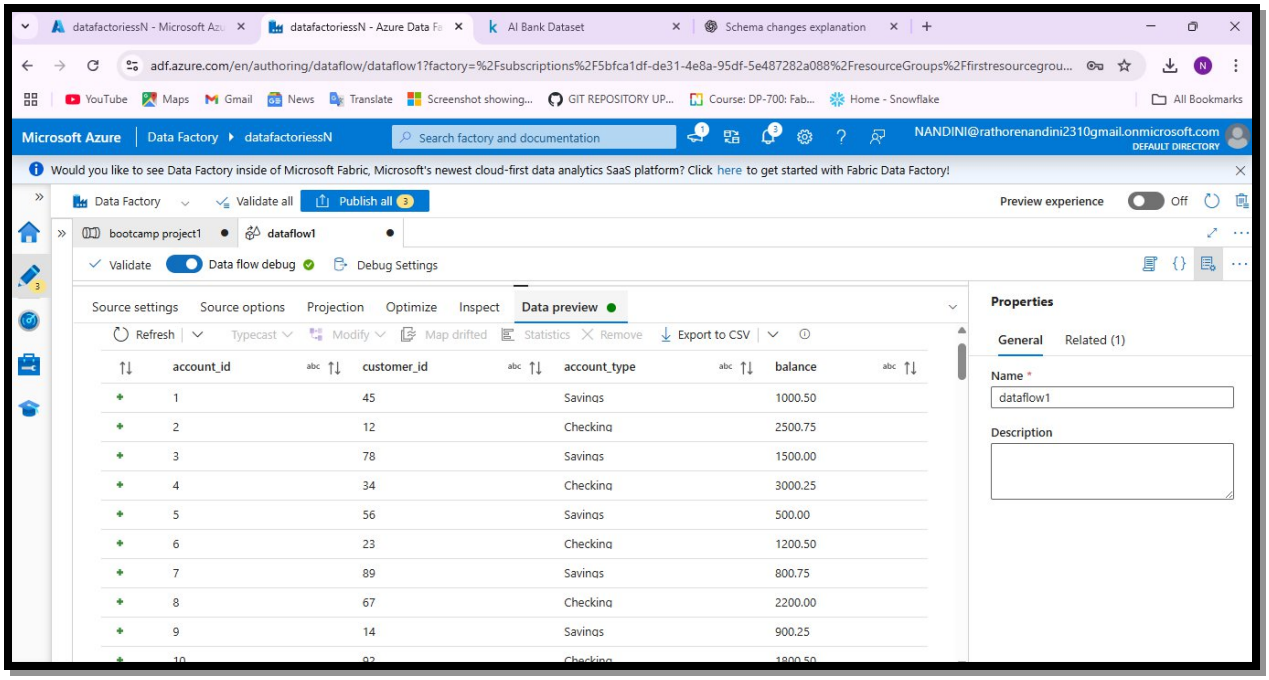


Select source dataset.

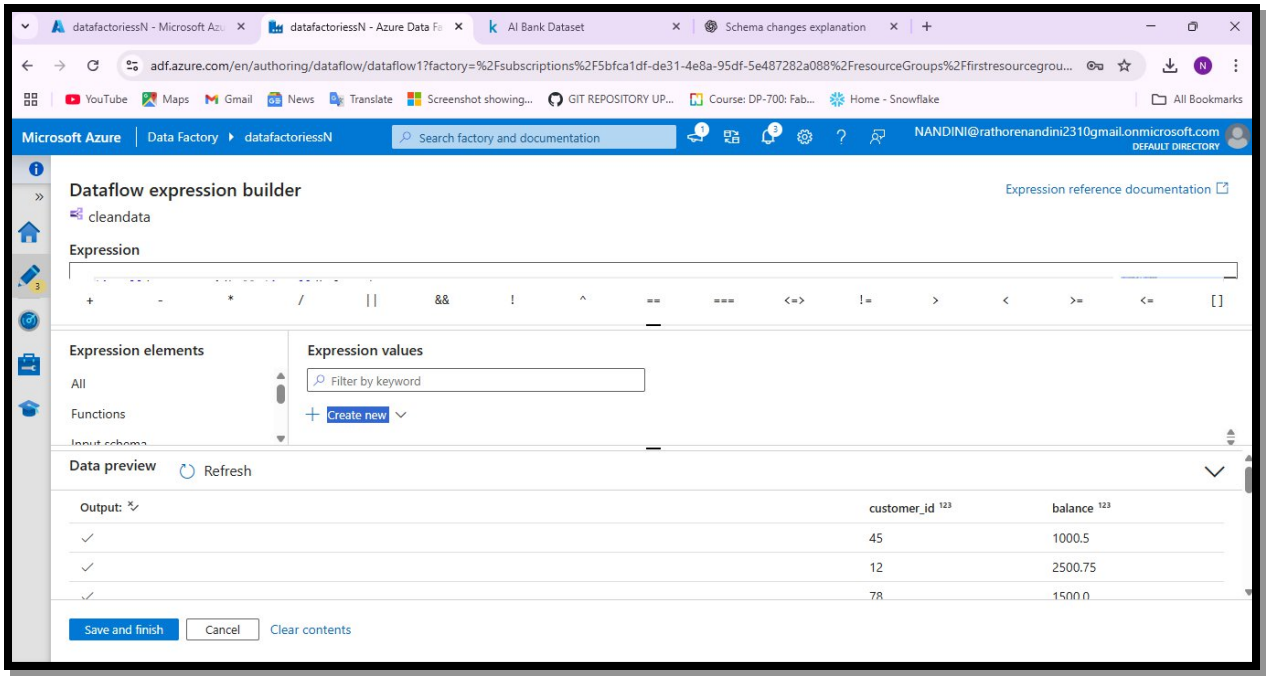

datafactoriessN - Microsoft Azu
datafactoriessN - Azure Data Fa
AI Bank Dataset
adf.azure.com/en/authoring/dataflow/dataflow1?factory=%2Fsubscriptions%2F5bfca1df-de31-4e8a-95df-5e487282a088%2FresourceGroups%2Ffirstresourcegrou...
YouTube
Maps
Gmail
News
Translate
Screenshot showing...
GIT REPOSITORY UP...
Course: DP-700: Fab...
Home - Snowflake
All Bookmarks
Microsoft Azure
Data Factory
datafactoriessN
Search factory and documentation
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform
Data Factory
Validate all
Publish all 2
bootcamp project1
dataflow1
Validate
Data flow debug
source1
Columns:
0 total
Source settings
Source options
Projection
Optimize
Inspect
Data preview
Dataset
Inline
Dataset *
Select...
New
Options
Allow schema drift
Infer drifted column types
Validate schema
Sampling *
Enable
Disable
New linked service
Azure Data Lake Storage Gen2
Learn more
Authentication type
Account key
Account selection method
From Azure subscription
Enter manually
Azure subscription
Azure subscription 1 (5bfca1df-de31-4e8a-95df-5e487282a088)
Storage account name *
nandiniadls
Test connection
To linked service
To file path
Annotations
New
Parameters
Connection successful
Create
Cancel
Test connection

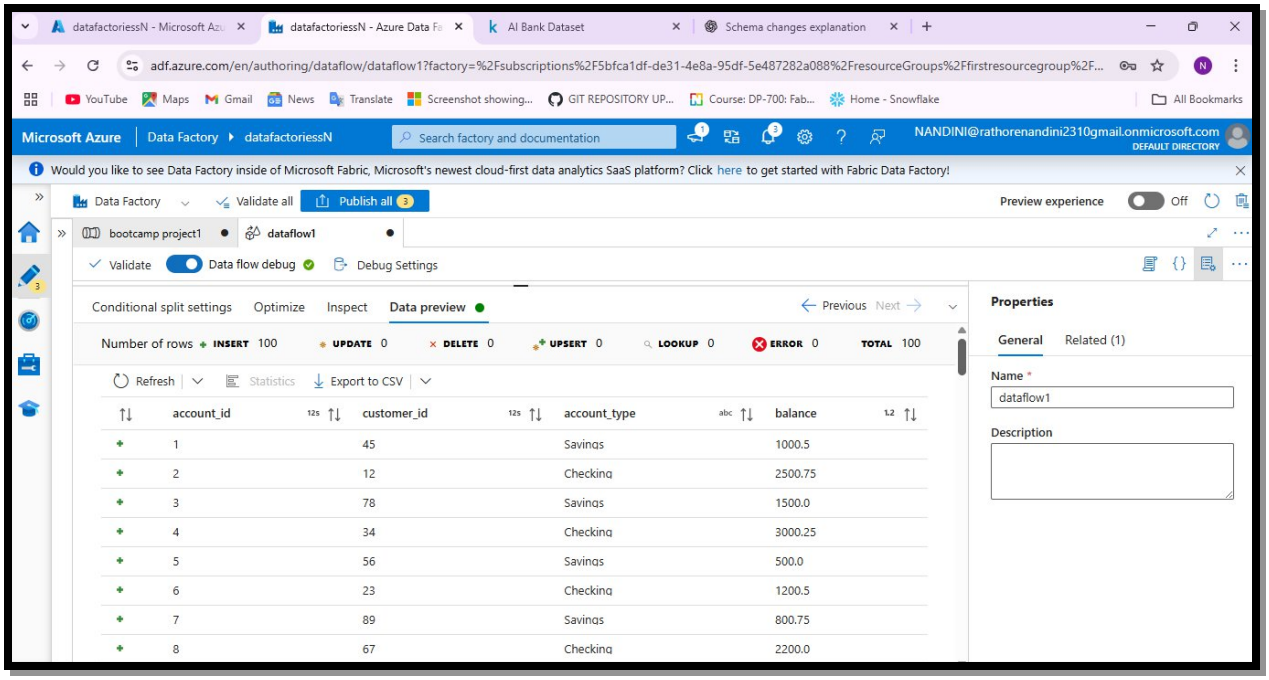

datafactoriessN - Microsoft Azu
datafactoriessN - Azure Data Fa
AI Bank Dataset
adf.azure.com/en/authoring/dataflow/dataflow1?factory=%2Fsubscriptions%2F5bfca1df-de31-4e8a-95df-5e487282a088%2FresourceGroups%2Ffirstresourcegrou...
YouTube
Maps
Gmail
News
Translate
Screenshot showing...
GIT REPOSITORY UP...
Course: DP-700: Fab...
Home - Snowflake
All Bookmarks
Microsoft Azure
Data Factory
datafactoriessN
Search factory and documentation
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform
Data Factory
Validate all
Publish all 2
bootcamp project1
dataflow1
Validate
Data flow debug
source1
Columns:
0 total
Source settings
Source options
Projection
Optimize
Inspect
Data preview
Dataset
Inline
Dataset *
Select...
New
Options
Allow schema drift
Infer drifted column types
Validate schema
Sampling *
Enable
Disable
Set properties
Name
set_properties
Linked service *
ls_adls
File path
bronze
Directory
accounts.csv
First row as header
Import schema
From connection/store
From sample file
None
Advanced
OK
Back
Cancel

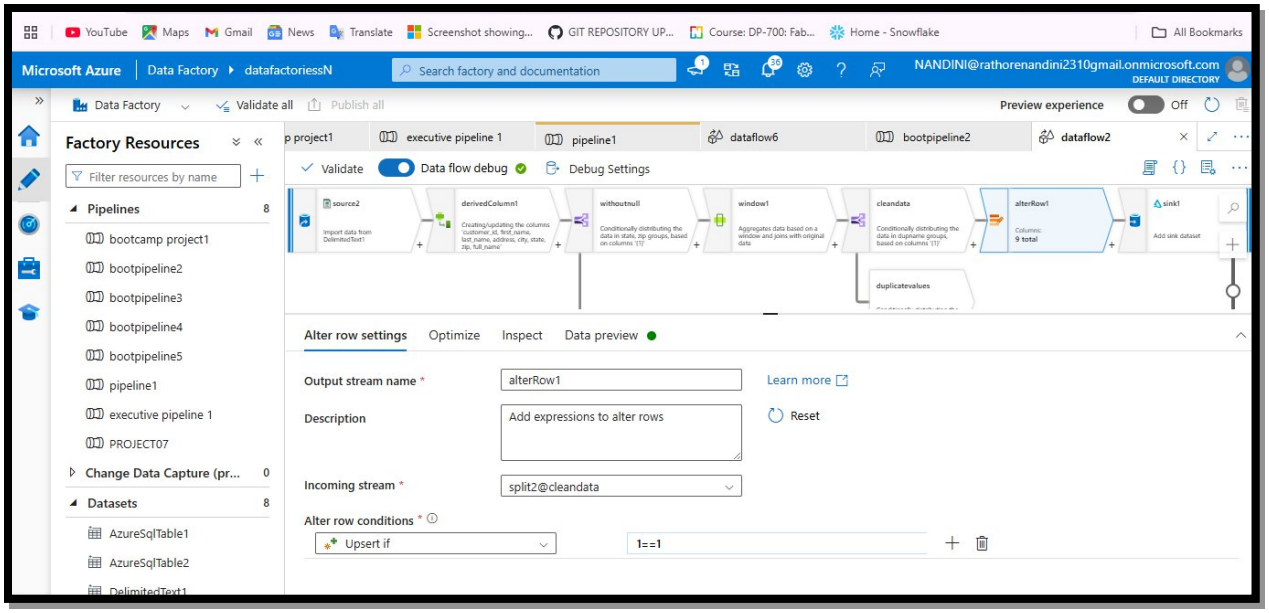




Add conditional splitting,split into two streams clean data andnull values.

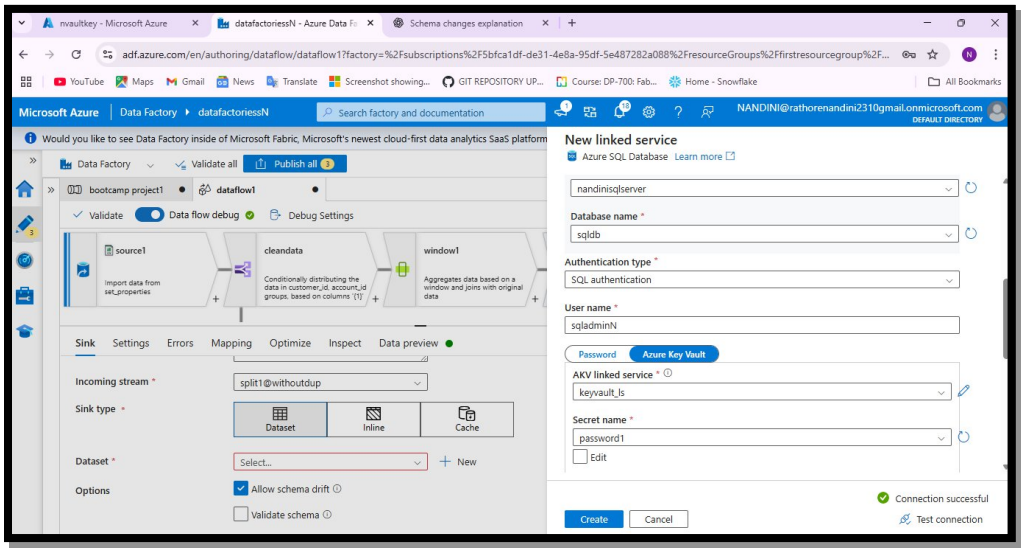

Microsoft Azure
Data Factory
datafactoriessN
Search factory and documentation
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Dataflow expression builder
cleandata
Expression reference documentation
Expression
Expression elements
All
Functions
Expression values
Filter by keyword
Create new
Data preview
Refresh
Output:
customer_id
balance
45
1000.5
12
2500.75
78
1500.0
Save and finish
Cancel
Clear contents

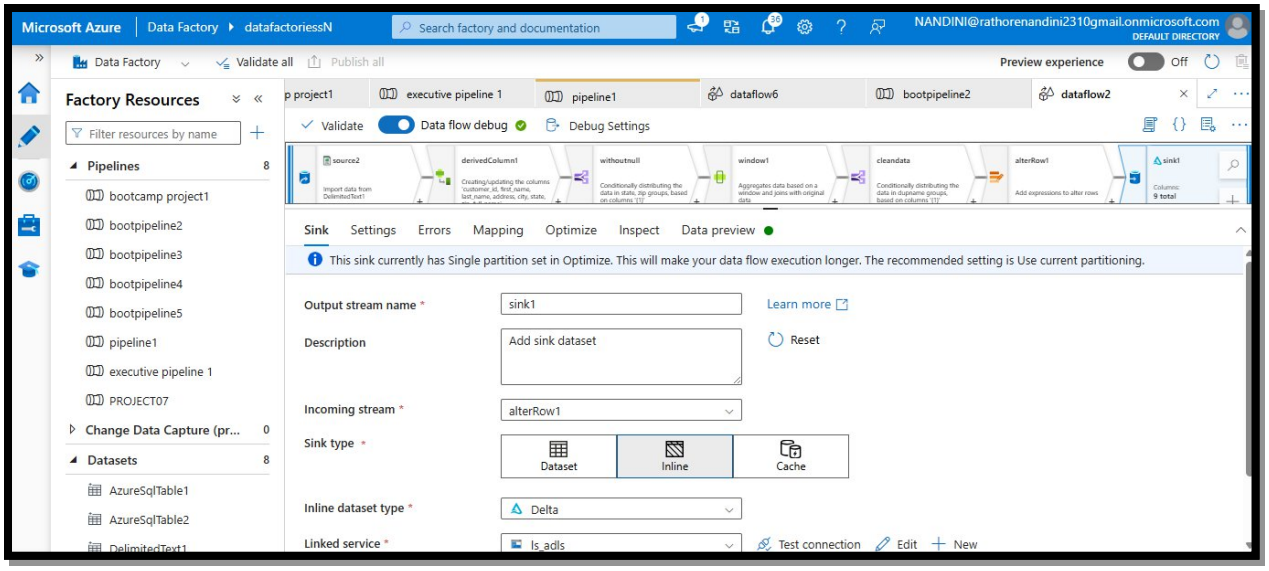

Microsoft Azure
Data Factory
datafactoriessN
Search factory and documentation
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click here to get started with Fabric Data Factory!
Data Factory
Validate all
Publish all 3
Preview experience
Off
bootcamp project1
dataflow1
Validate
Data flow debug
Debug Settings
Conditional split settings
Optimize
Inspect
Data preview
Previous
Next
Number of rows
INSERT 100
UPDATE 0
DELETE 0
UPSERT 0
LOOKUP 0
ERROR 0
TOTAL 100
Refresh
Statistics
Export to CSV
account_id
customer_id
account_type
balance
1 45 Savings 1000.5
2 12 Checking 2500.75
3 78 Savings 1500.0
4 34 Checking 3000.25
5 56 Savings 500.0
6 23 Checking 1200.5
7 89 Savings 800.75
8 67 Checking 2200.0
Properties
General
Related (1)
Name *
dataflow1
Description



alter row.

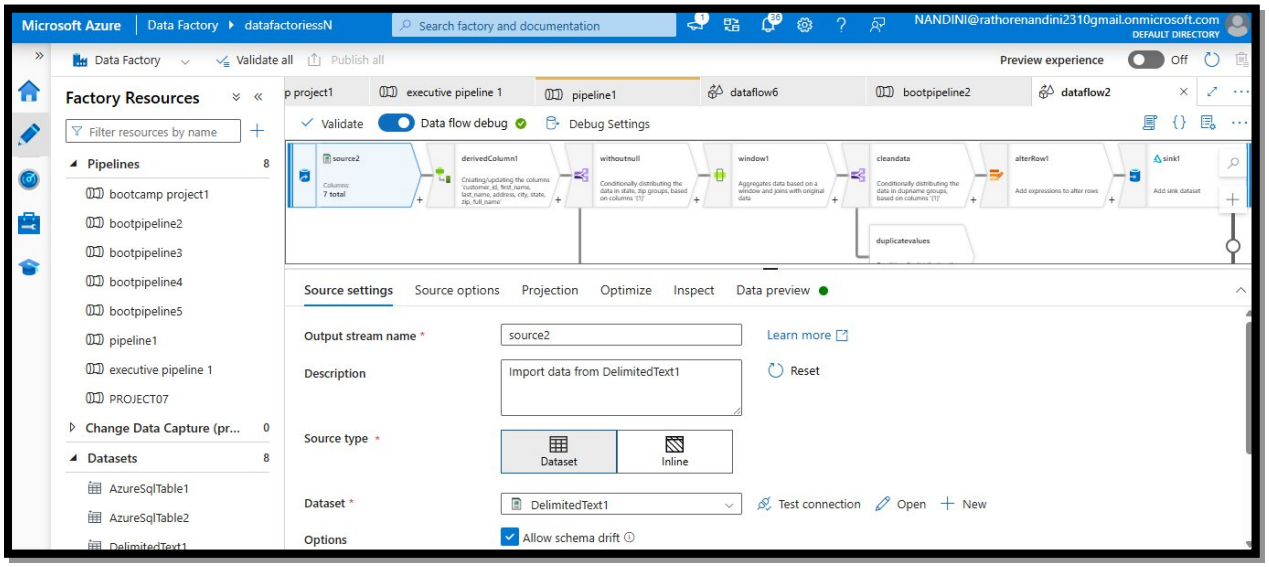
Using key vault to save secrets.

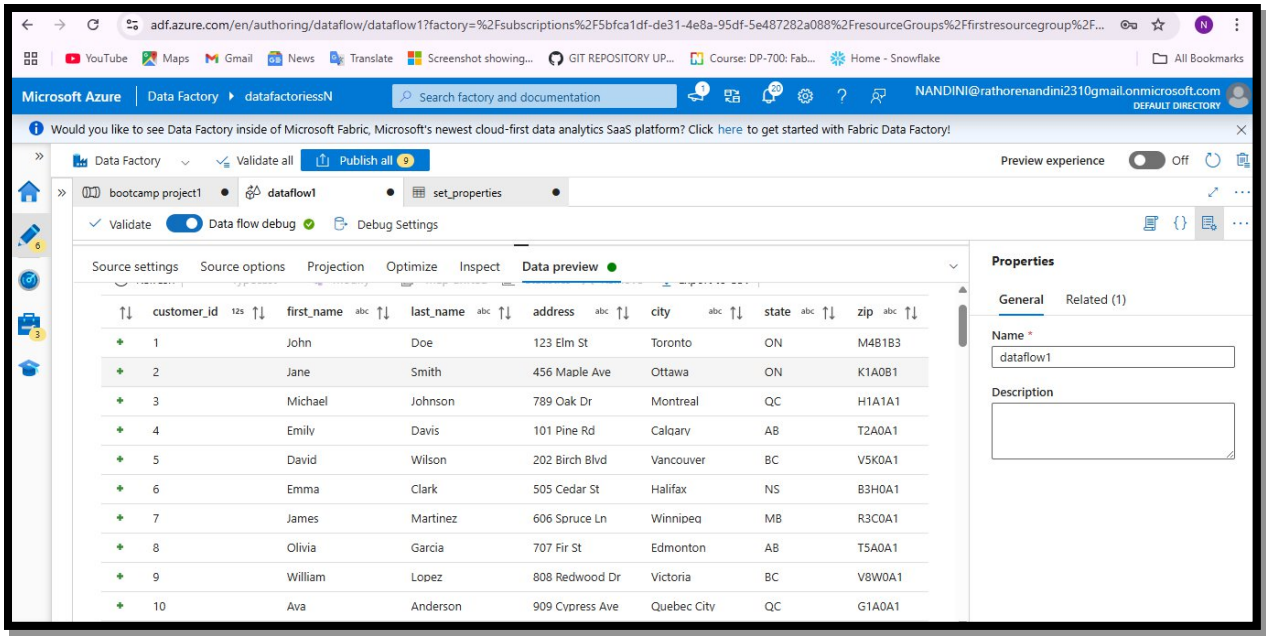


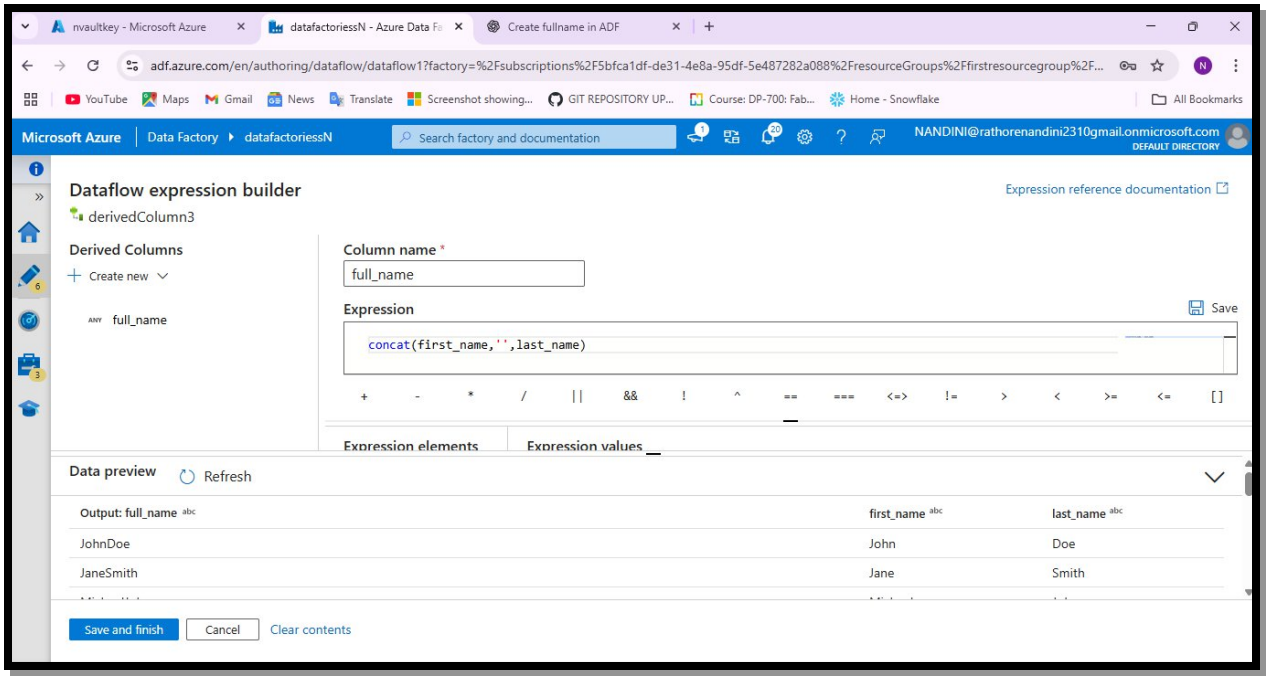


Sink csv file into delta format in silver layer.

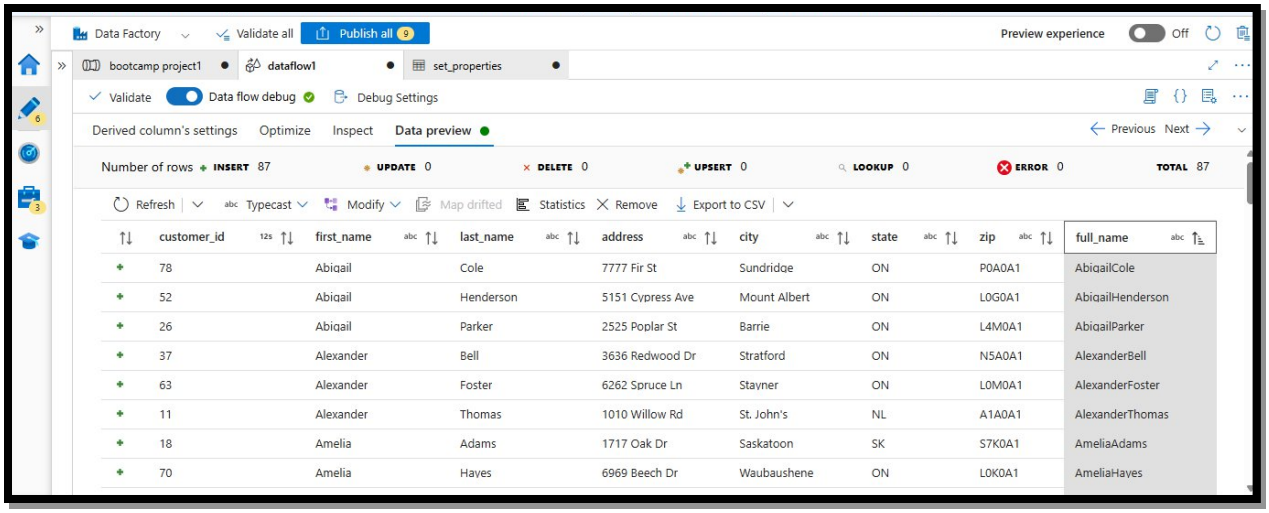
SOURCE 2—CUSTOMER.

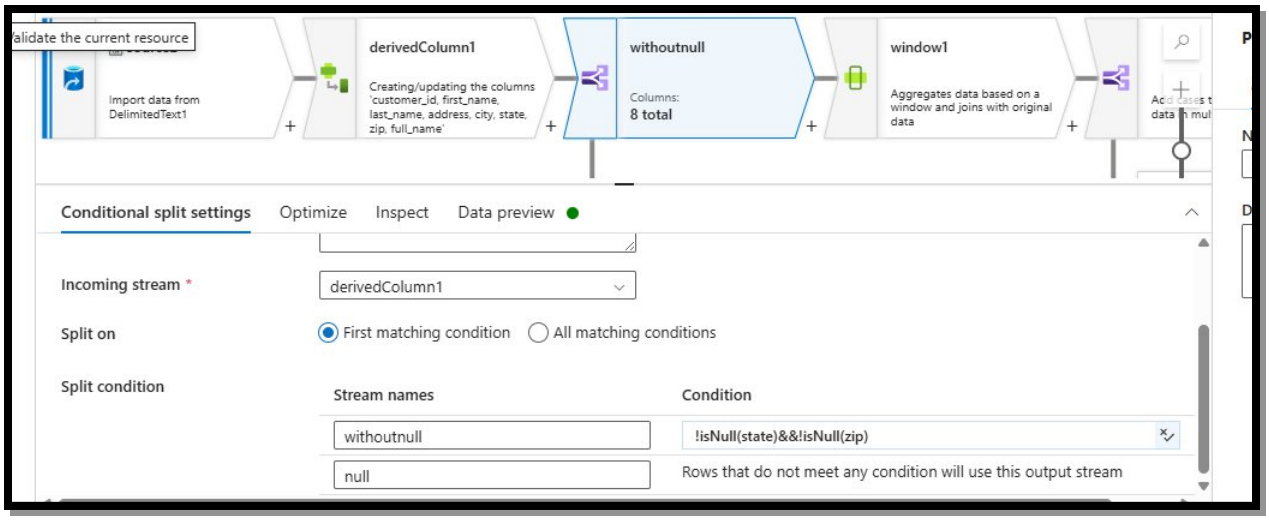




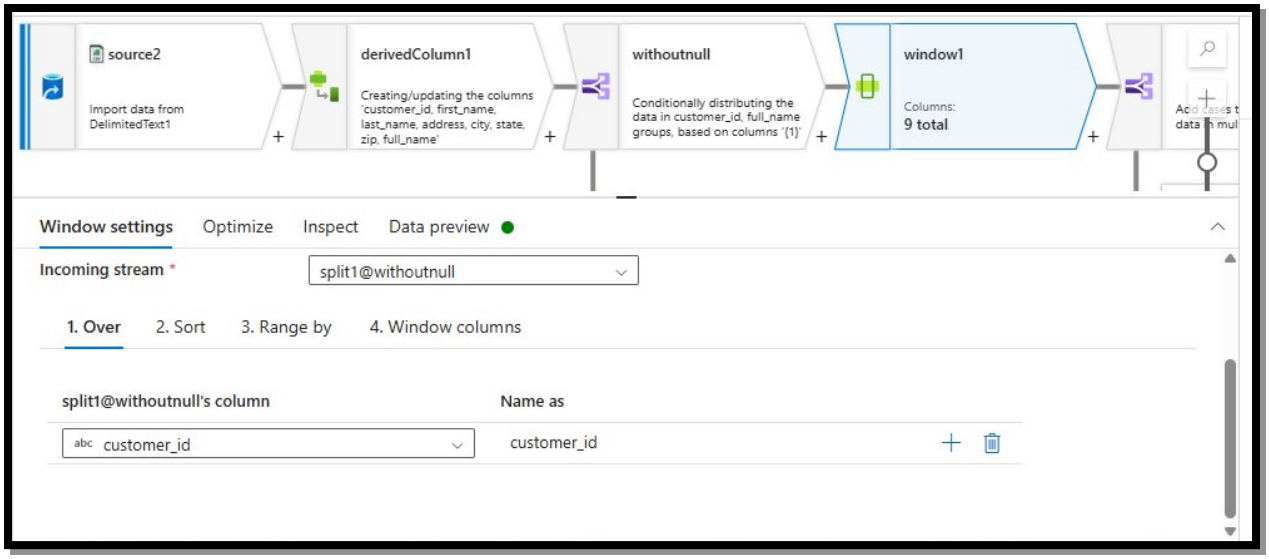


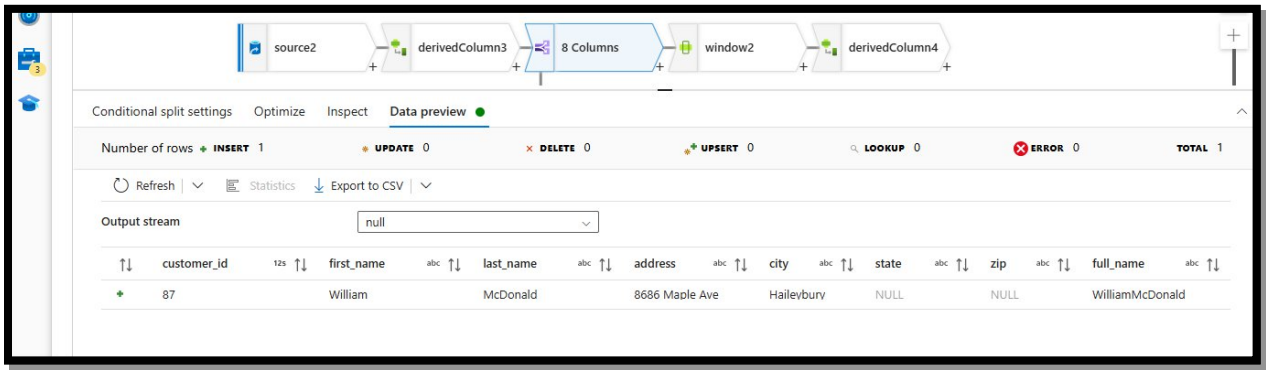
Using derived column to generate a new column-full name.



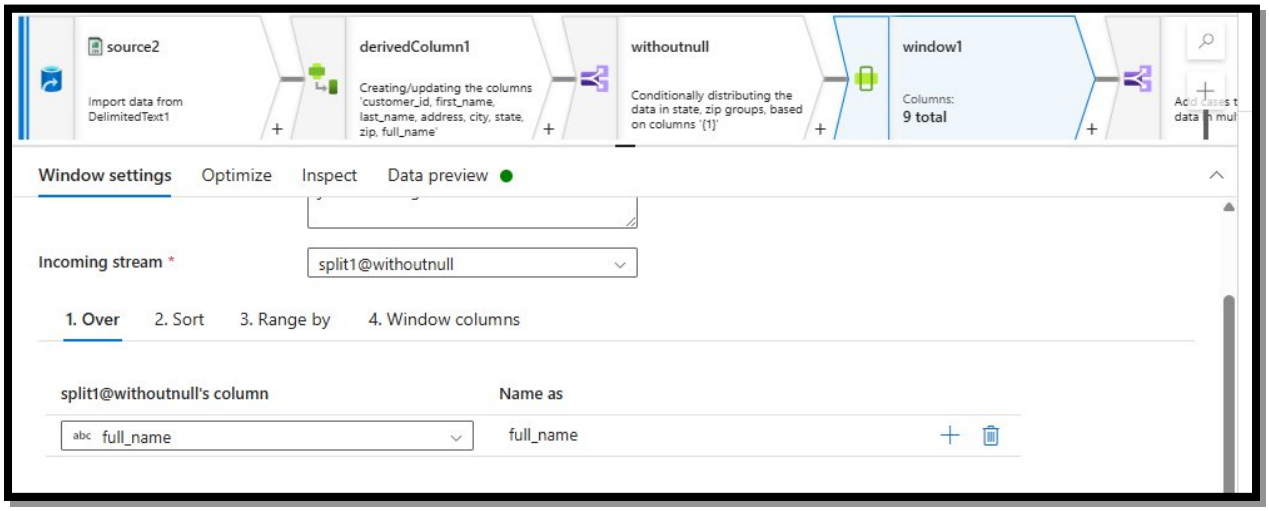


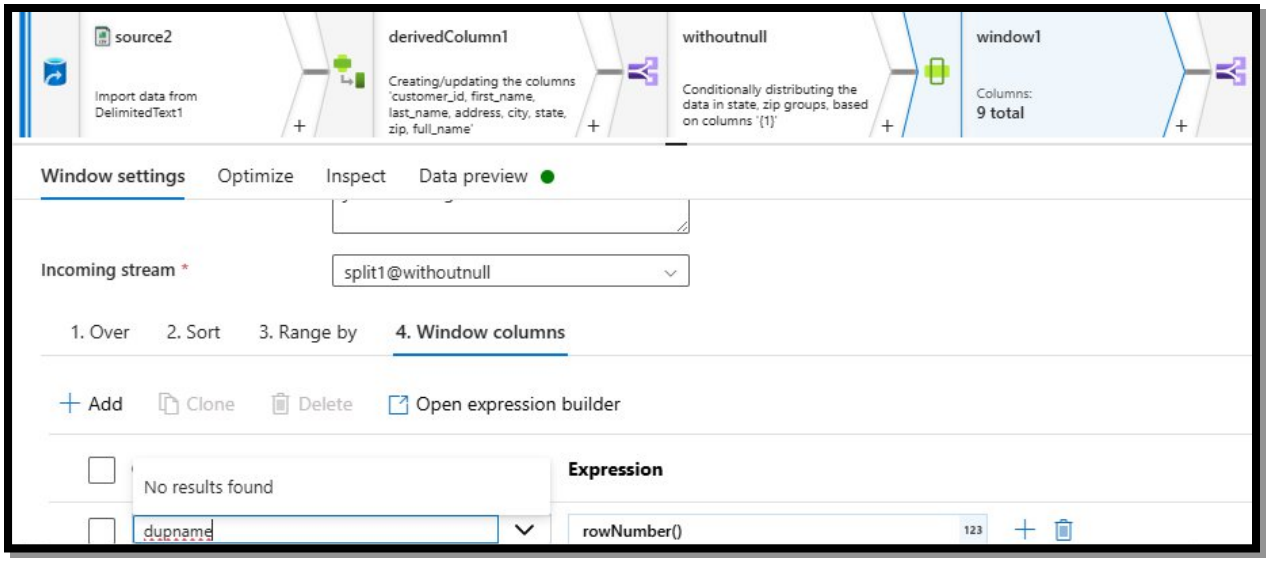
Removing null values.

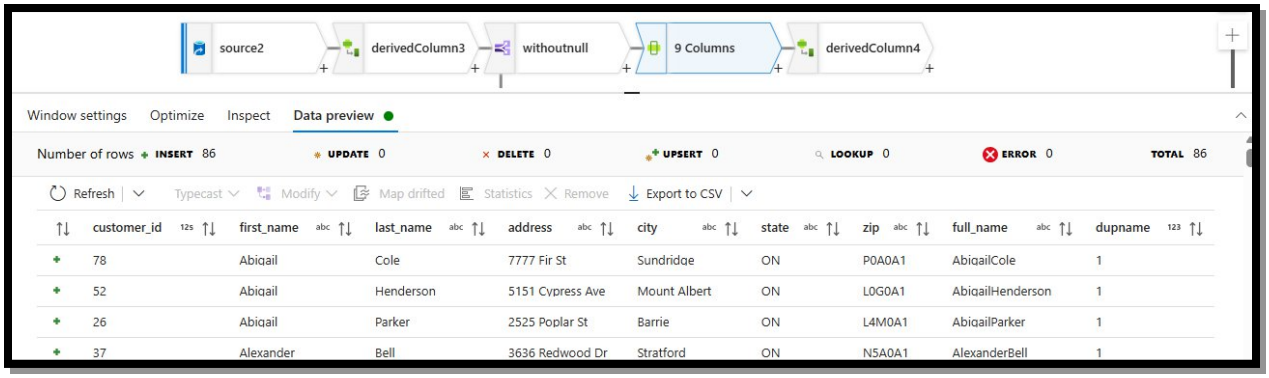


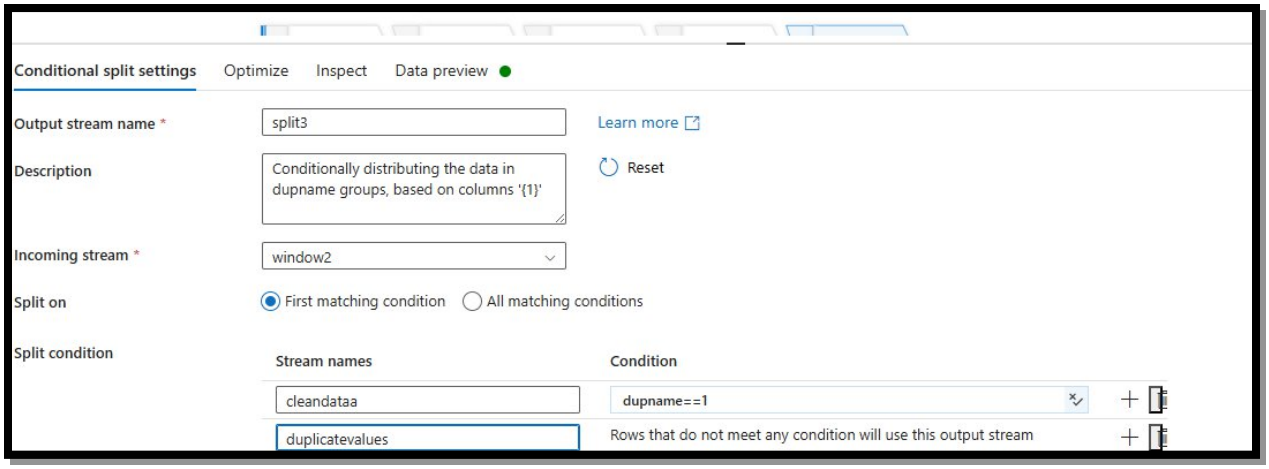


1 row have null values.

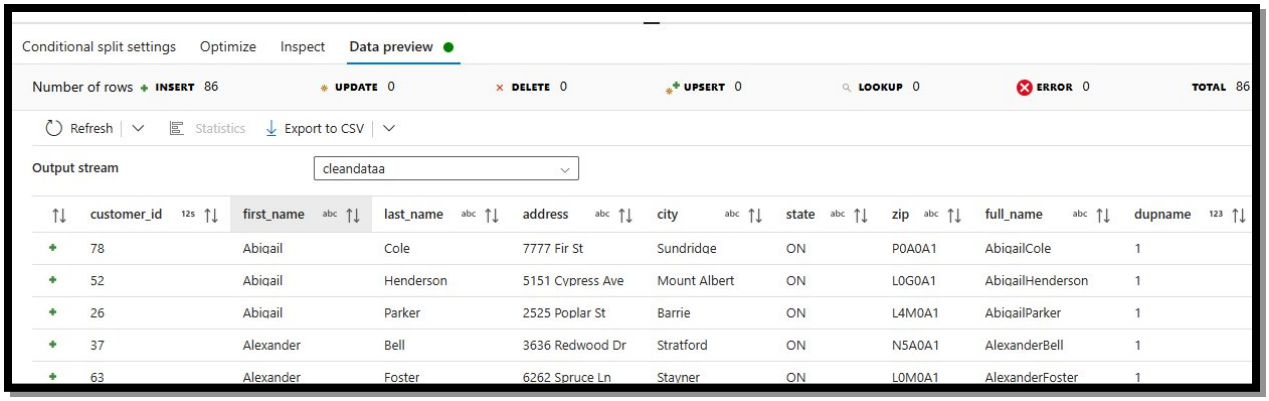


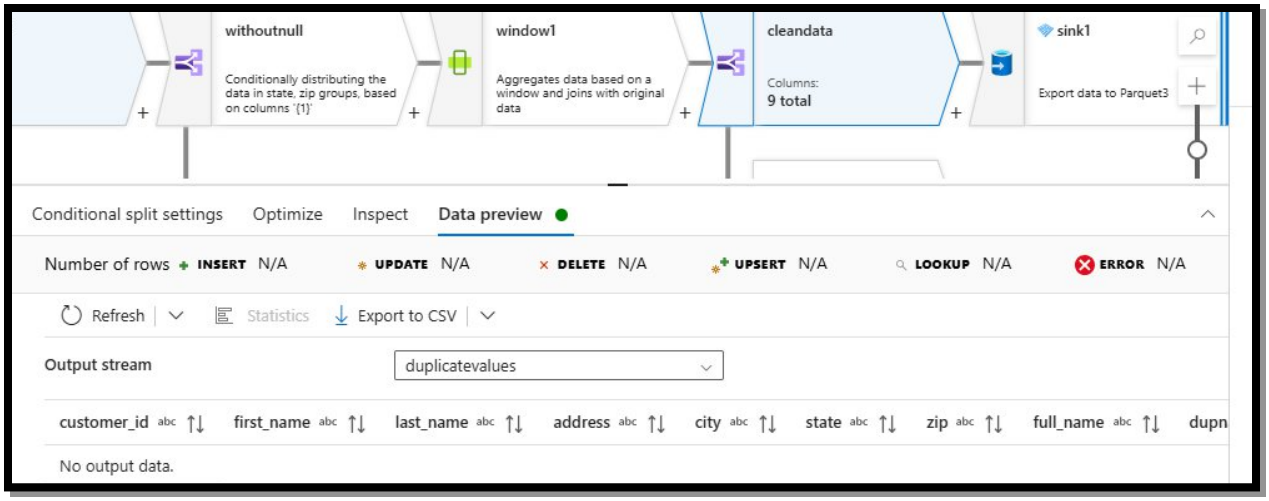




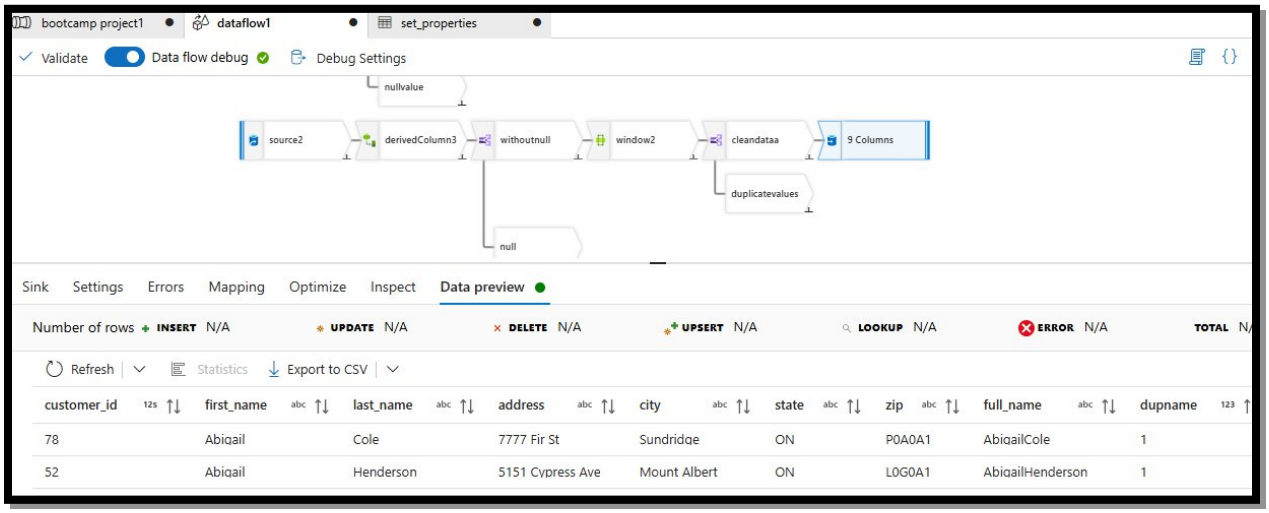


Removing duplicate values.





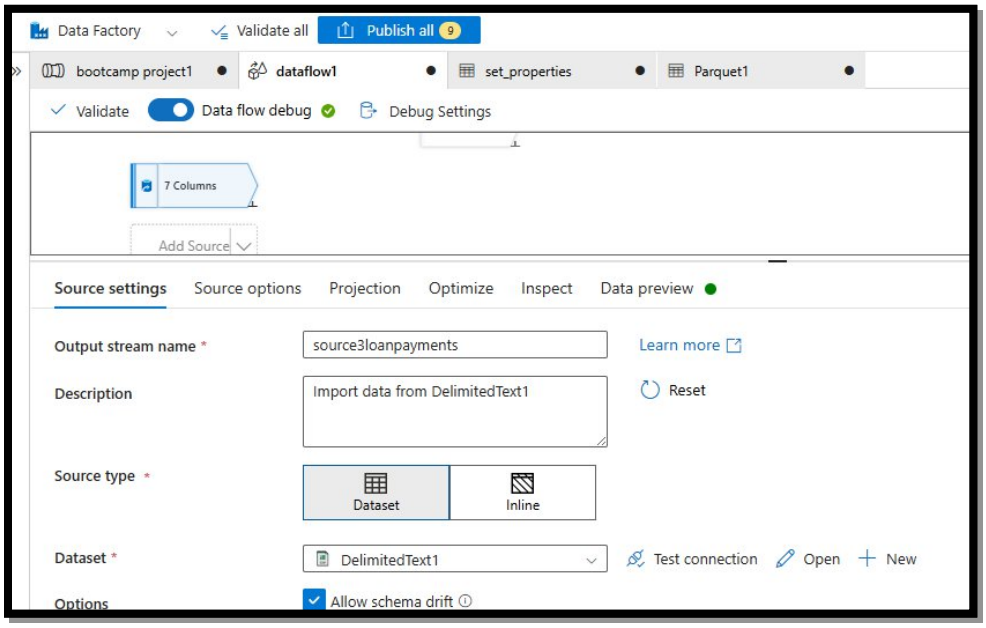
No duplicate values are there.

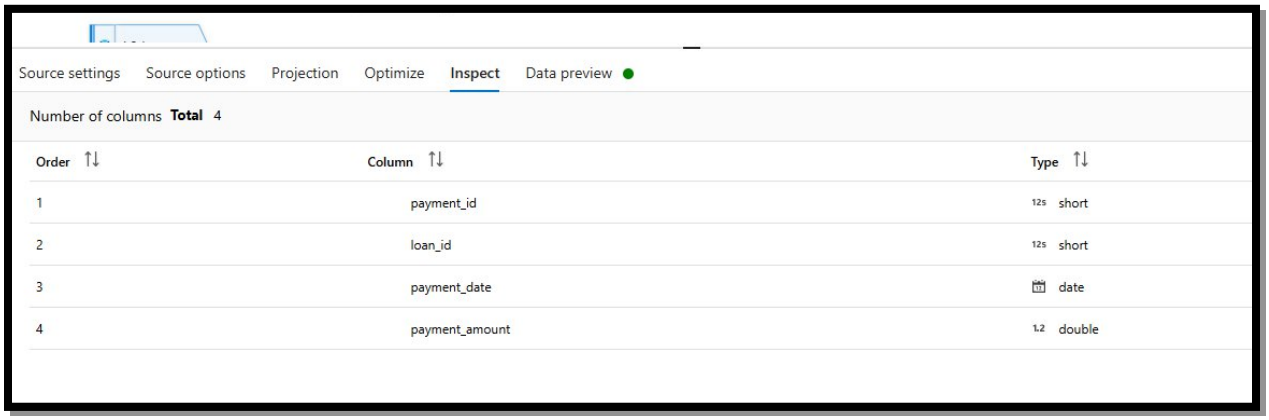


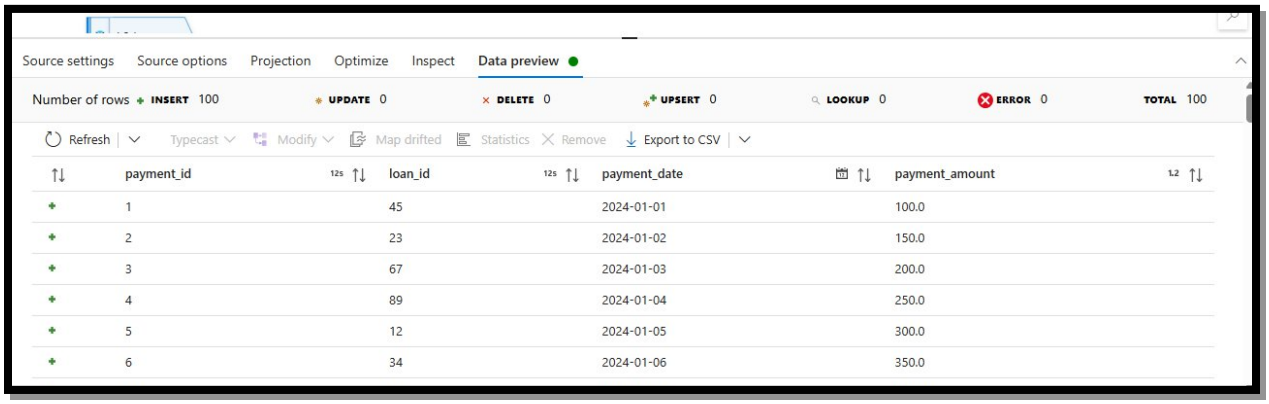
3 source

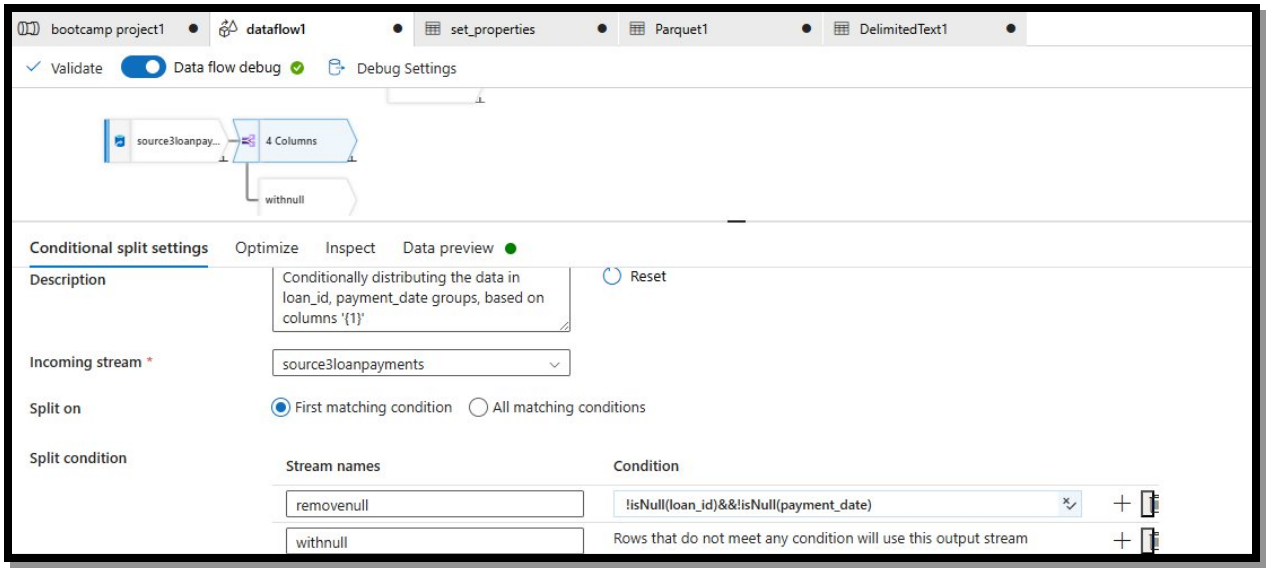
NANDINI RATHORE

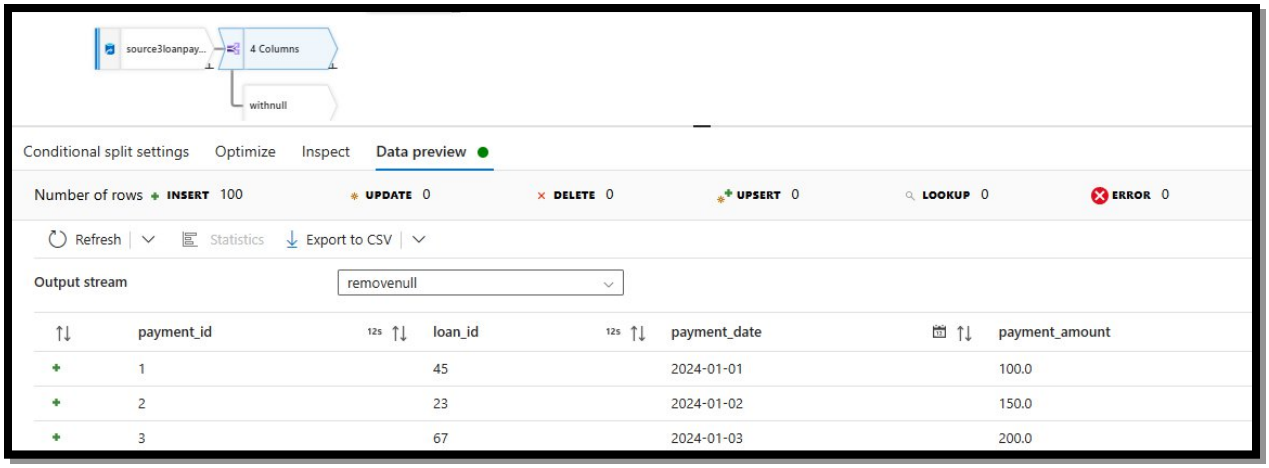
3 SOURCE—LOAN_PAYMENTS.



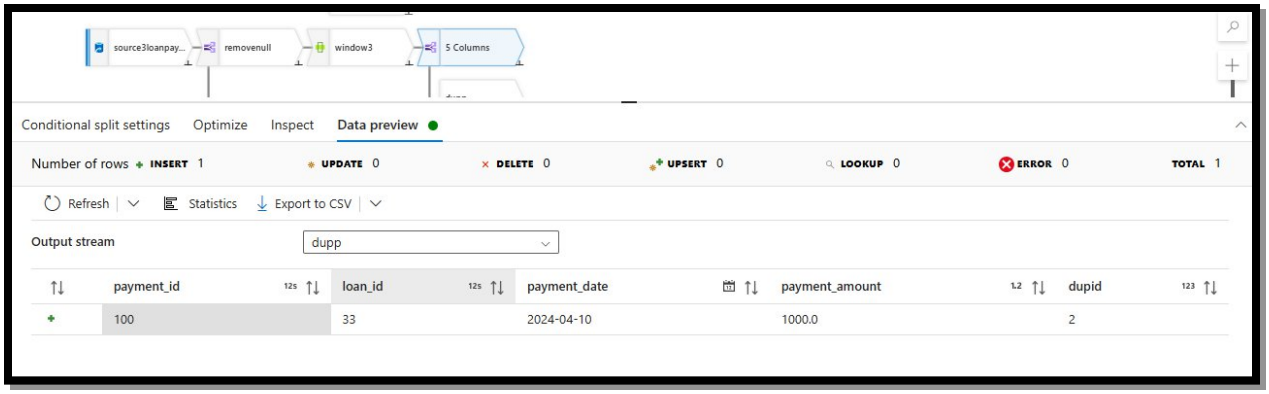




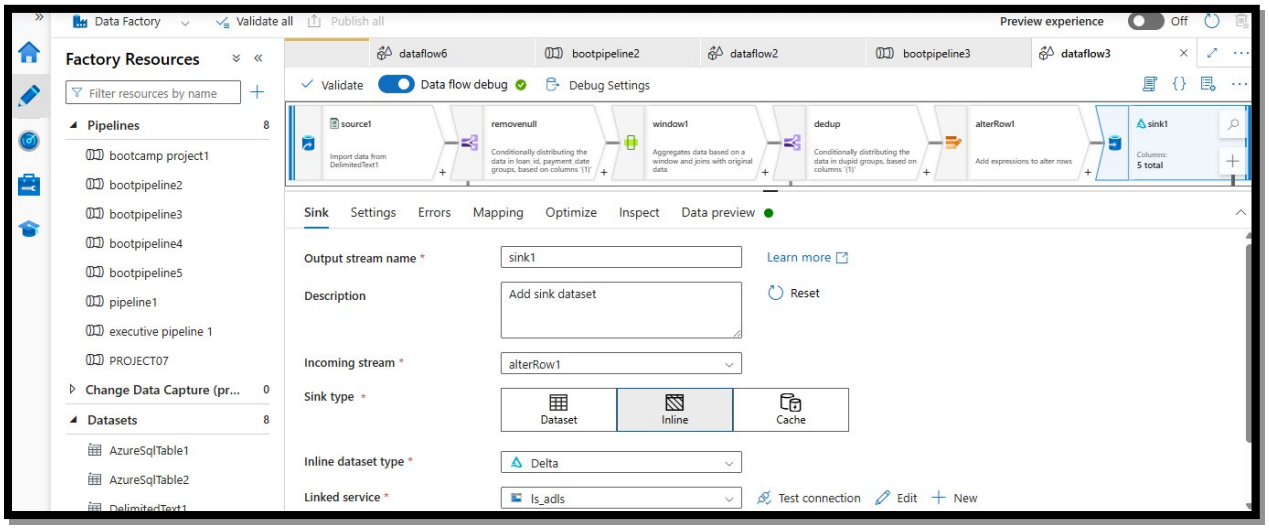

bootcamp project1
dataflow1
set_properties
Parquet1
DelimitedText1
Validate
Data flow debug
Debug Settings
source3loanpay...
4 Columns
withnull
Conditional split settings
Optimize
Inspect
Data preview
Description
Conditionally distributing the data in loan_id, payment_date groups, based on columns '{1}'
Reset
Incoming stream *
source3loanpayments
Split on
First matching condition
All matching conditions
Split condition
Stream names
Condition
removenull
!isNull(loan_id)&&!isNull(payment_date)
withnull
Rows that do not meet any condition will use this output stream


source3loanpay...
4 Columns
withnull
Conditional split settings
Optimize
Inspect
Data preview
Number of rows
INSERT 100
UPDATE 0
DELETE 0
UPSERT 0
LOOKUP 0
ERROR 0
Refresh
Statistics
Export to CSV
Output stream
removenull
payment_id
loan_id
payment_date
payment_amount
1
45
2024-01-01
100.0
2
23
2024-01-02
150.0
3
67
2024-01-03
200.0

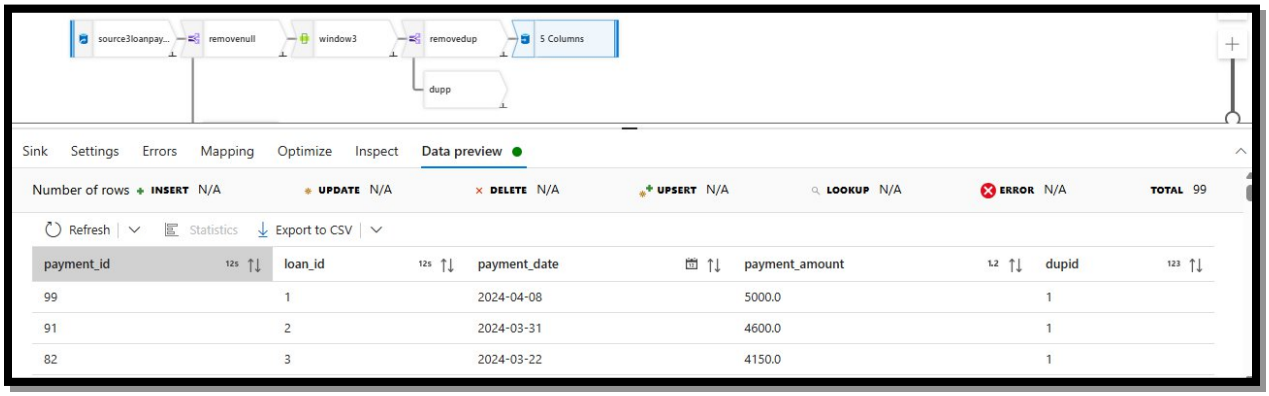

Column name *
dupid
Expression
rowNumber()
+ - * / || && ! ^ == === <=> != >



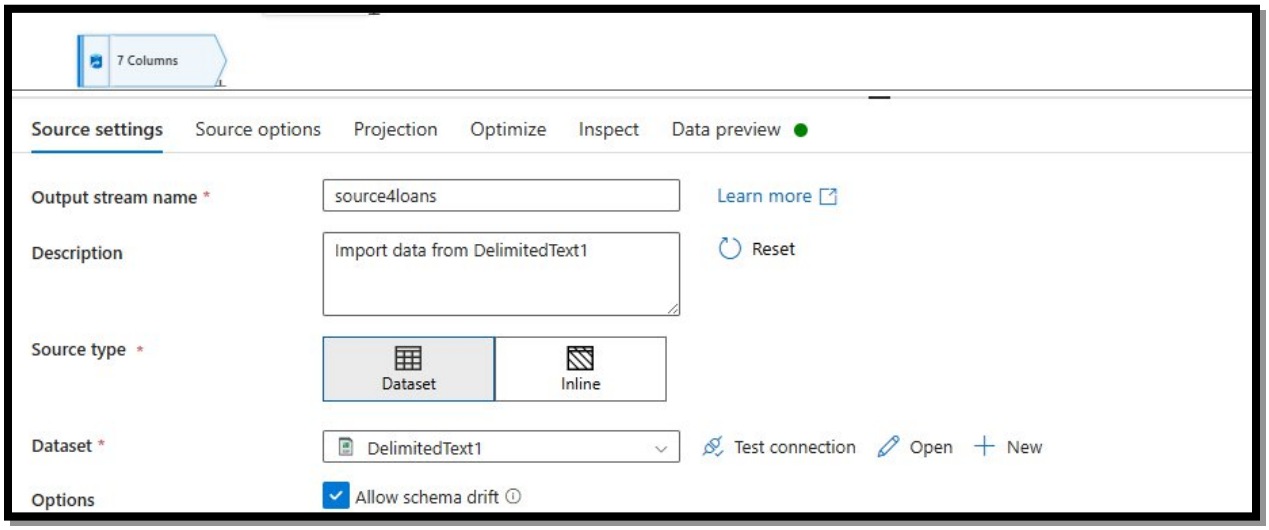
DUPLICATE VALUE.

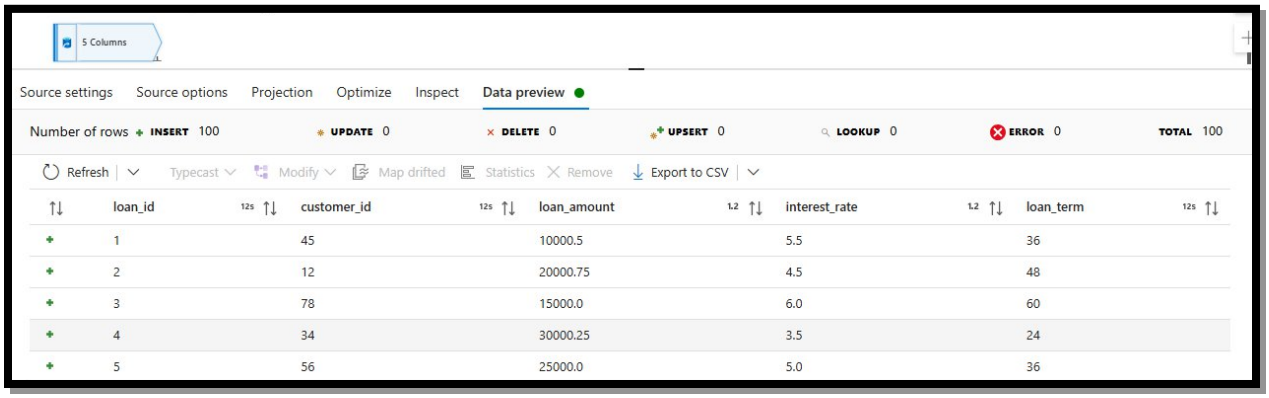


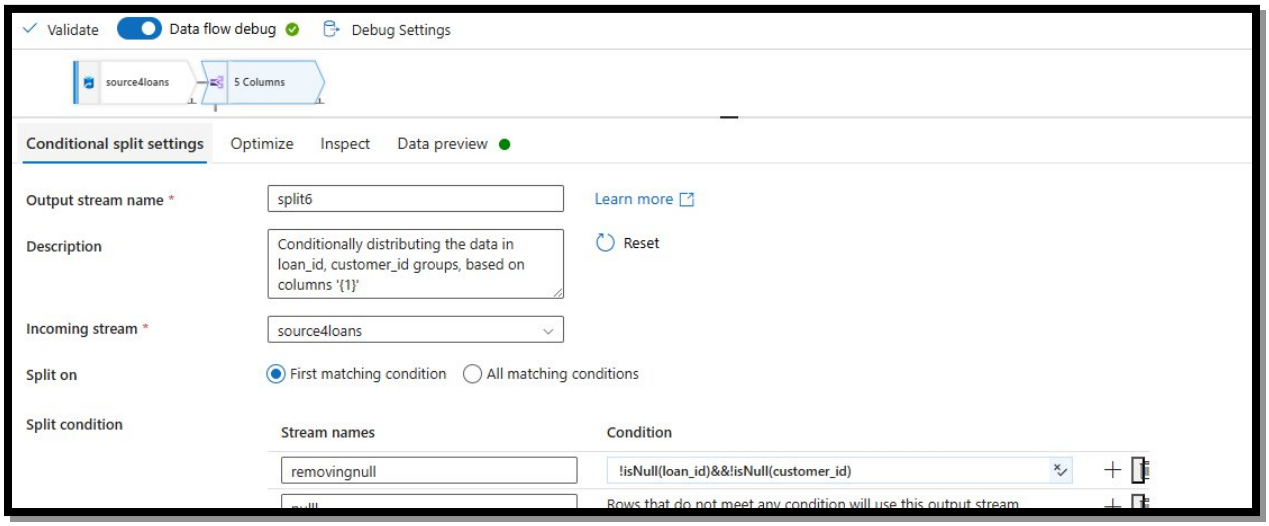
SINK TO SQLDB.

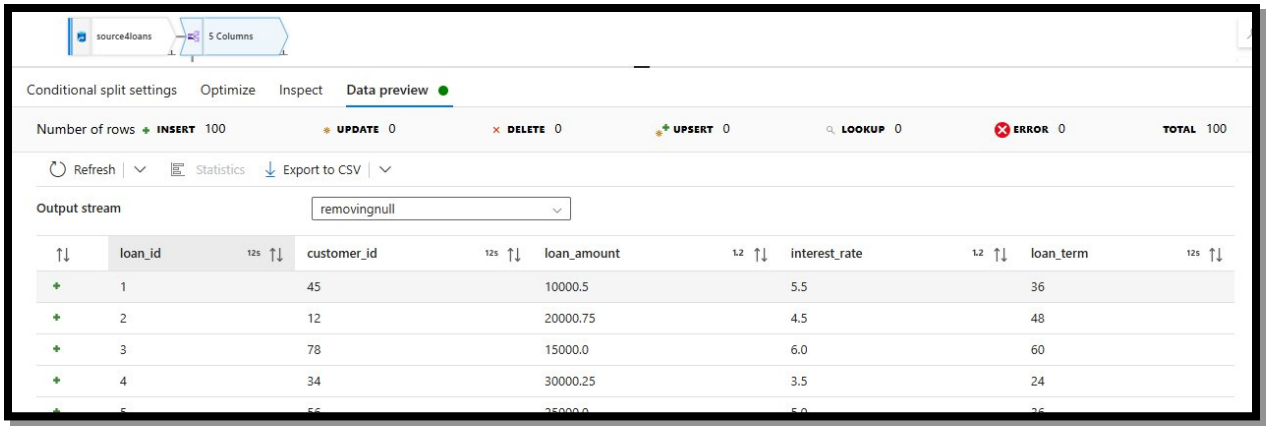


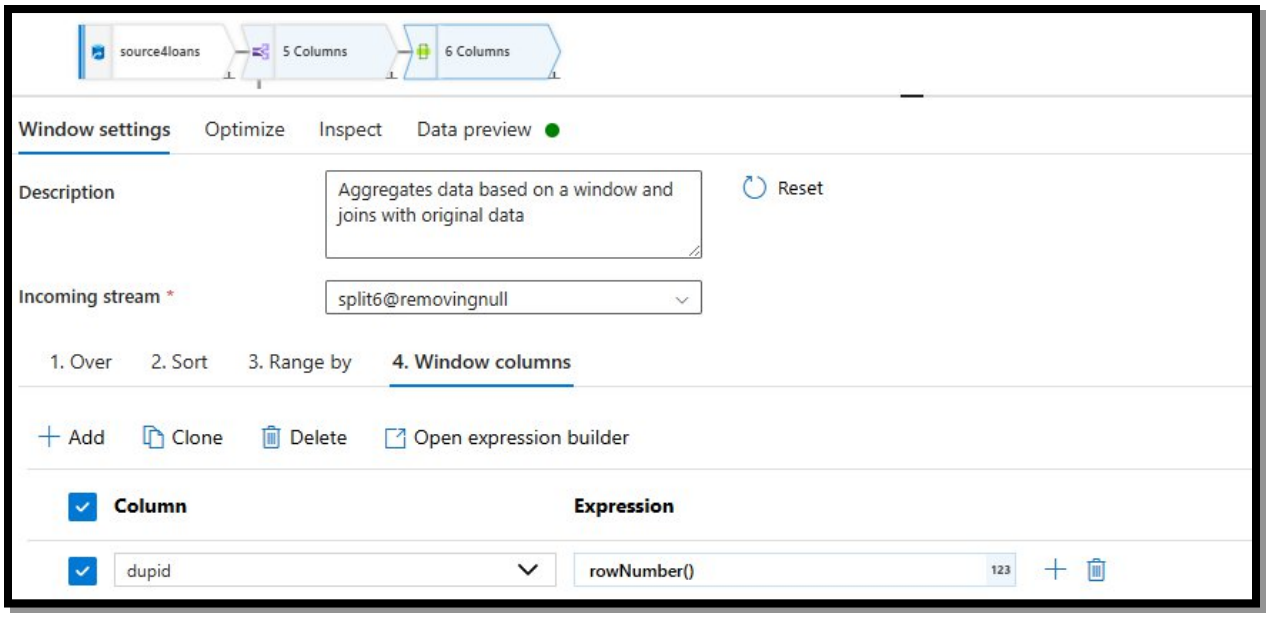
4 SOURCEàLOANS.

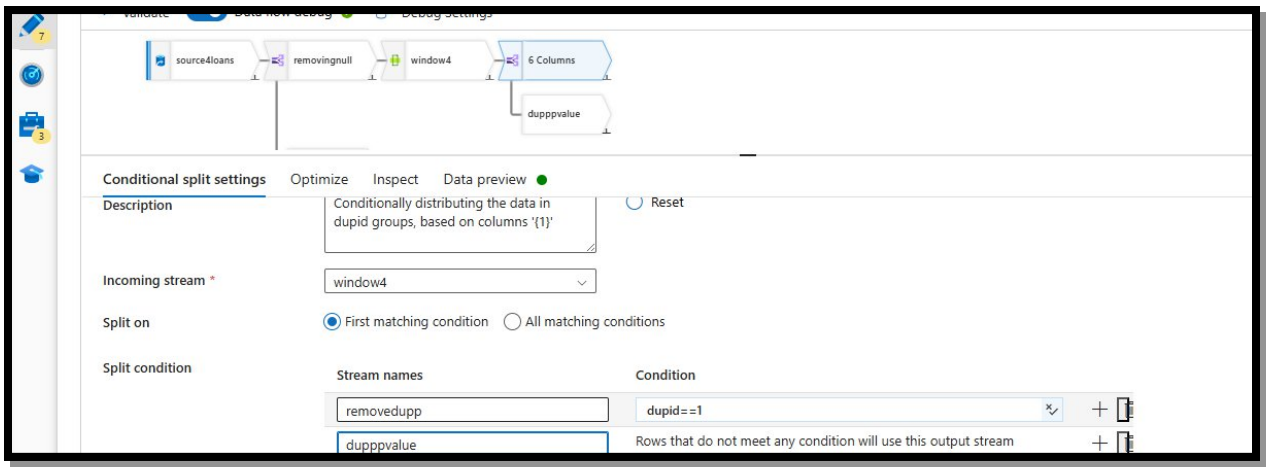


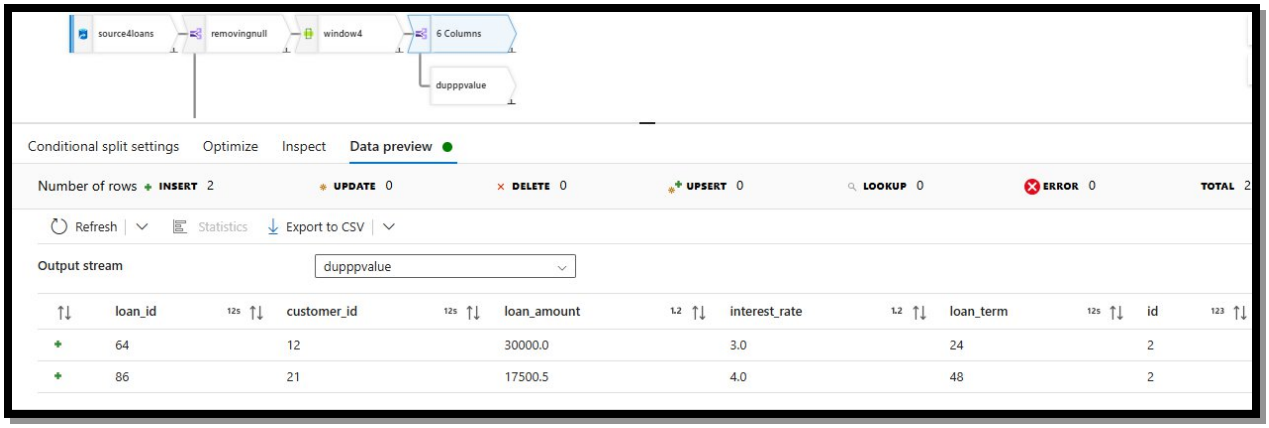


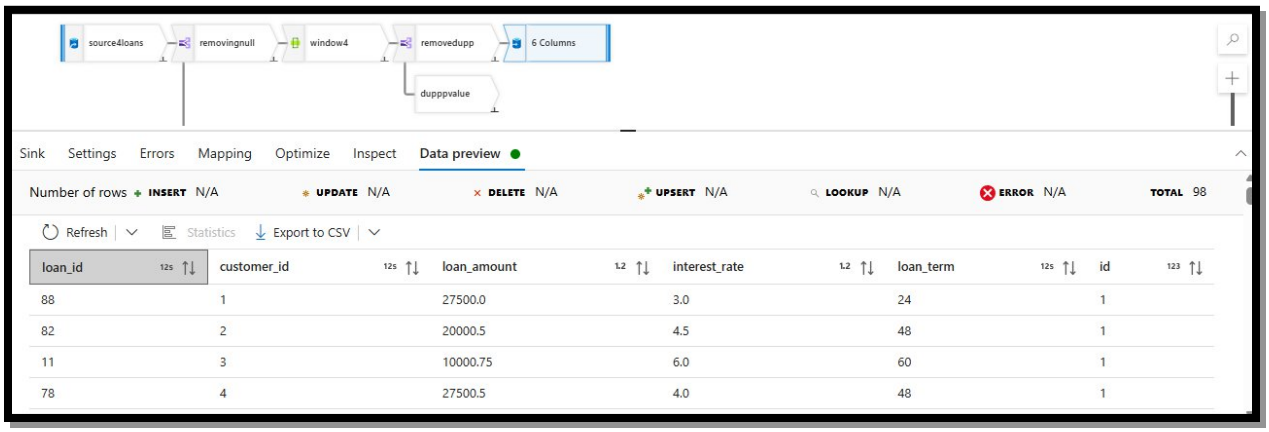



source4loans
5 Columns
Conditional split settings
Optimize
Inspect
Data preview
Number of rows
INSERT 100
UPDATE 0
DELETE 0
UPSERT 0
LOOKUP 0
ERROR 0
TOTAL 100
Refresh
Statistics
Export to CSV
Output stream
removingnull
loan_id
customer_id
loan_amount
interest_rate
loan_term
1 45 10000.5 5.5 36
2 12 20000.75 4.5 48
3 78 15000.0 6.0 60
4 34 30000.25 3.5 24

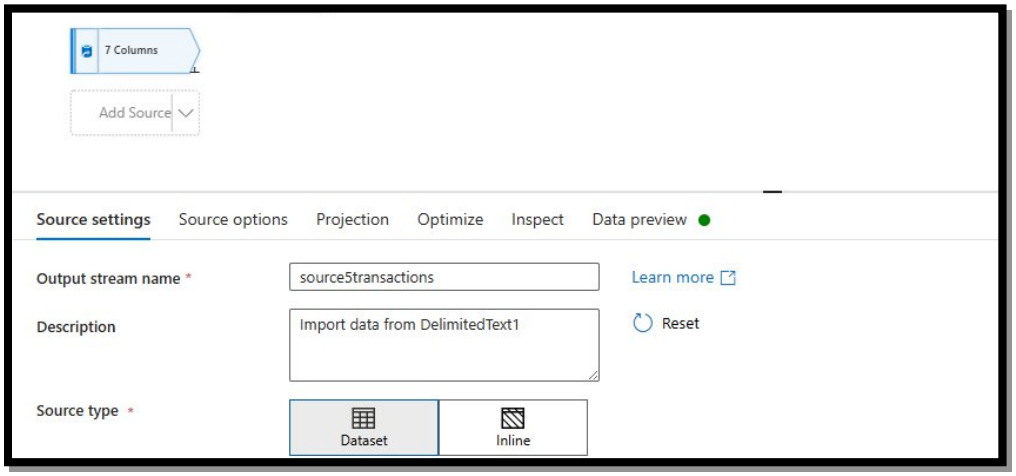

source4loans
5 Columns
6 Columns
Window settings
Optimize
Inspect
Data preview
Description
Aggregates data based on a window and joins with original data
Reset
Incoming stream *
split6@removingnull
1. Over
2. Sort
3. Range by
4. Window columns
Add
Clone
Delete
Open expression builder
Column
Expression
dupid
rowNumber()

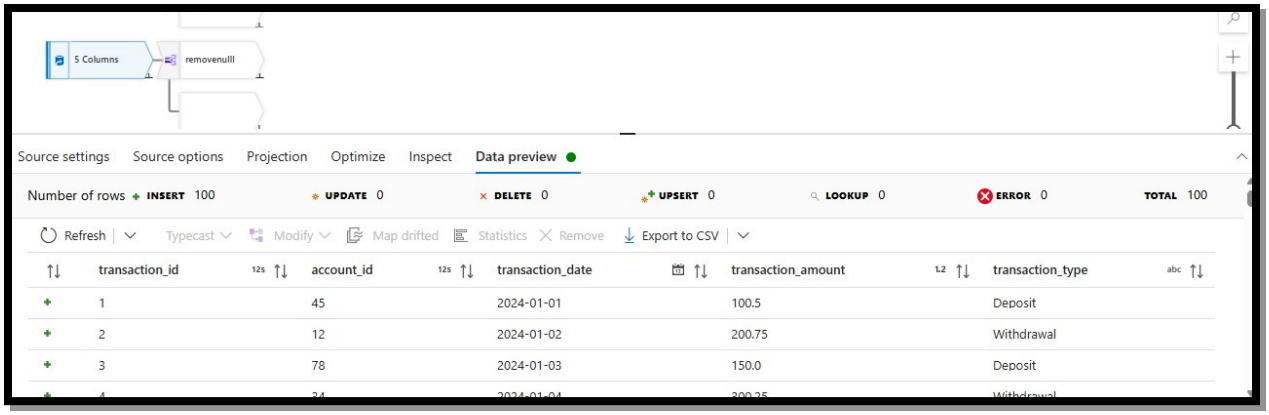

source4loans
removingnull
window4
6 Columns
dupppvalue
Conditional split settings
Optimize
Inspect
Data preview
Description
Conditionally distributing the data in dupid groups, based on columns '{1}'
Reset
Incoming stream *
window4
Split on
First matching condition
All matching conditions
Split condition
Stream names
Condition
removedupp
dupid==1
dupppvalue
Rows that do not meet any condition will use this output stream



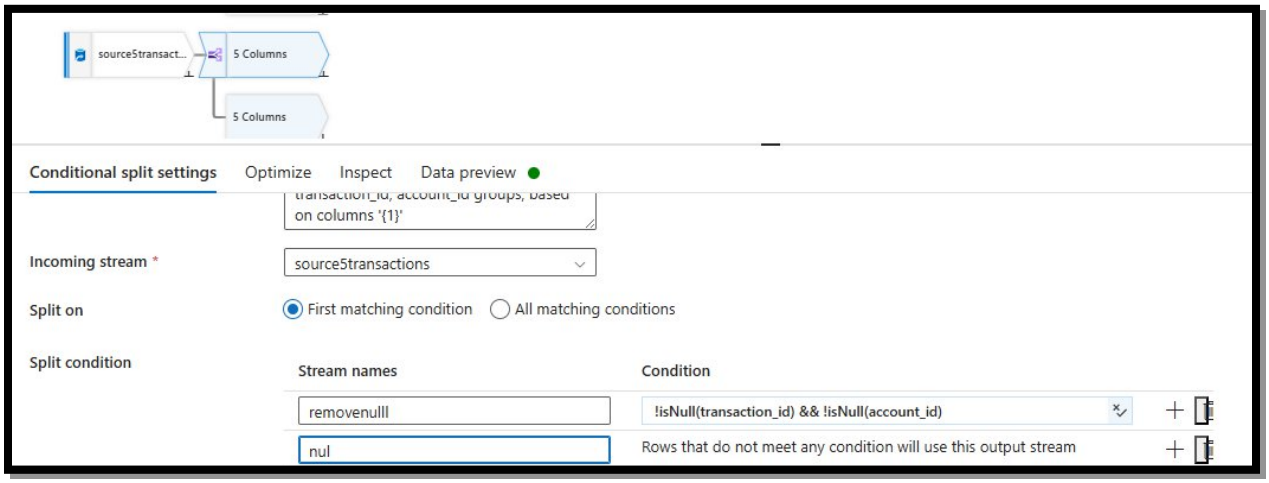


5. SOURCEàTRANSACTIONS

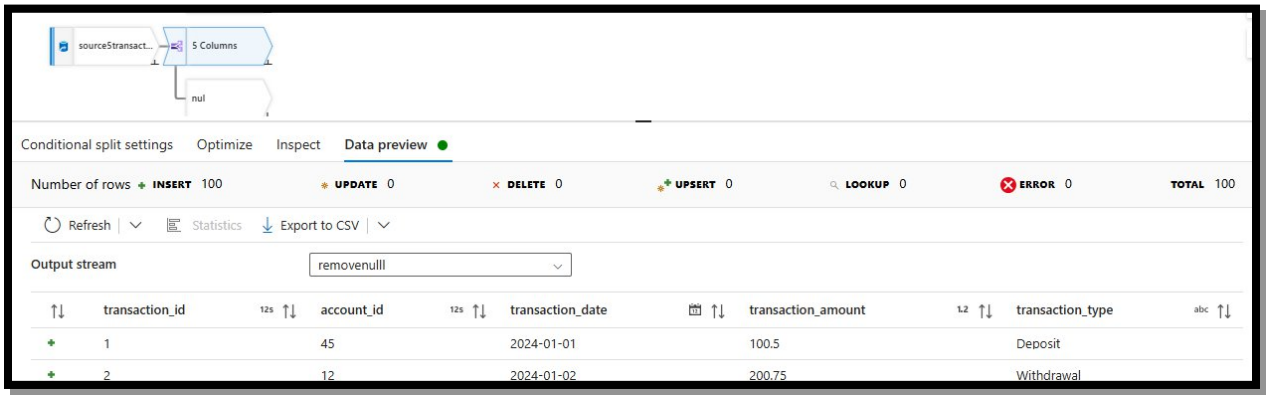


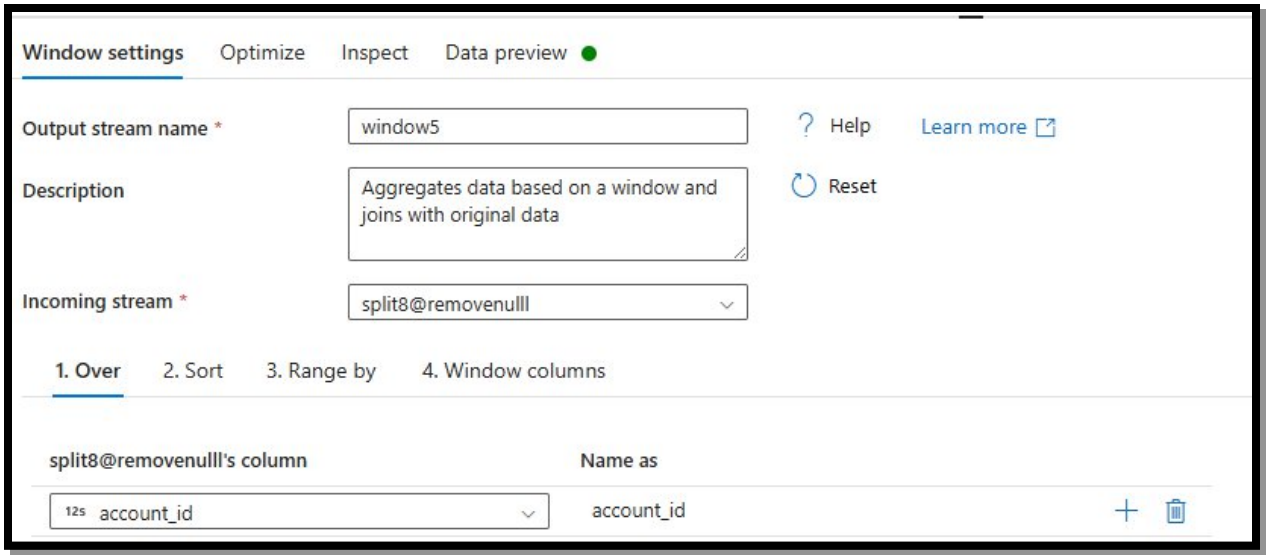


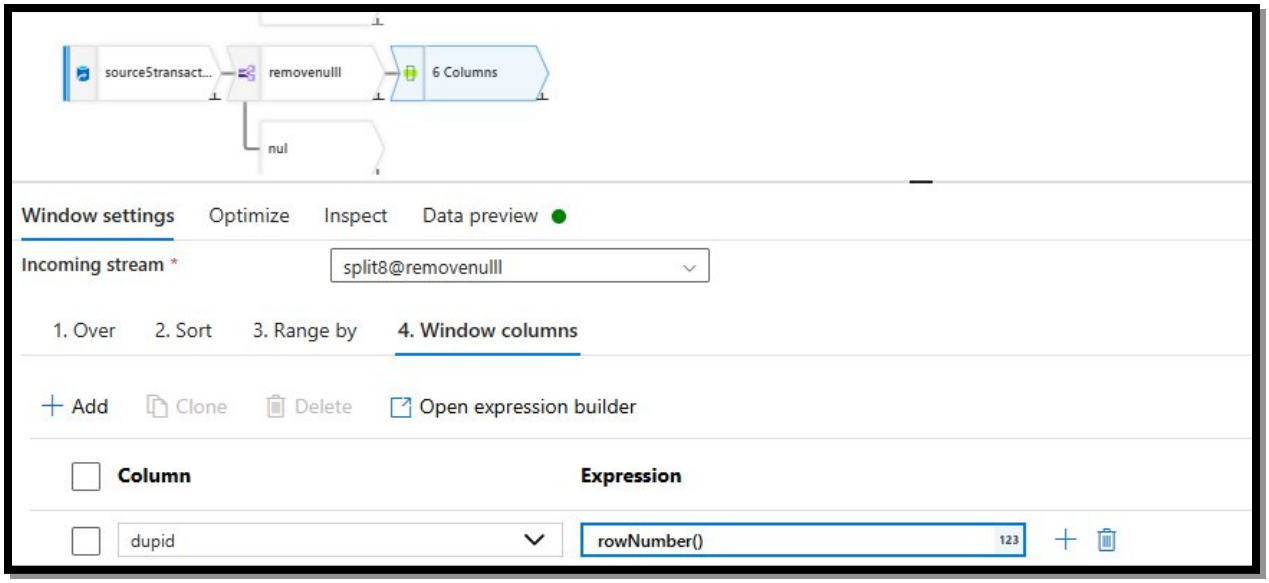
DATA PREVIEW.



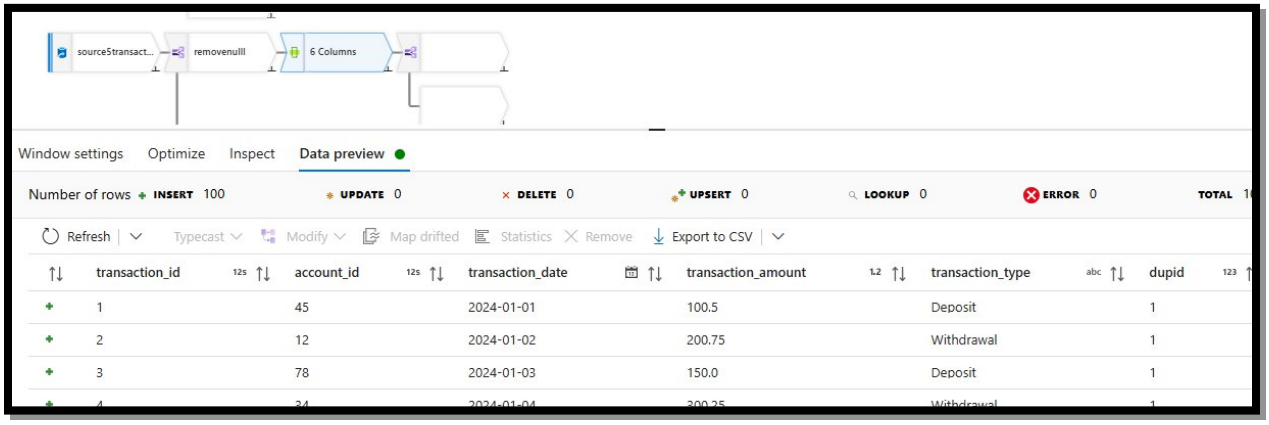
REMOVING NULL VALUES.

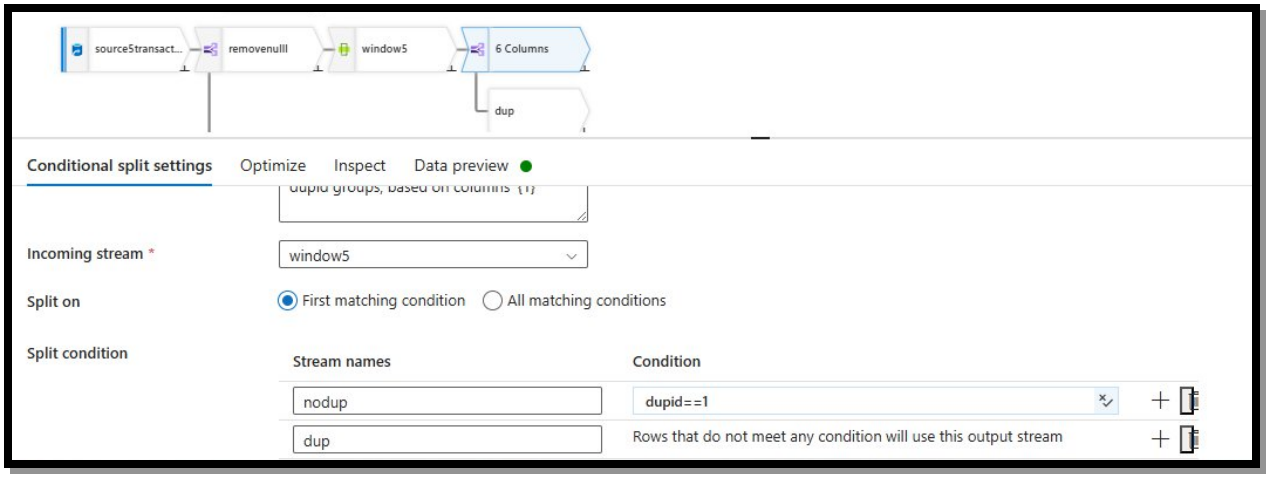


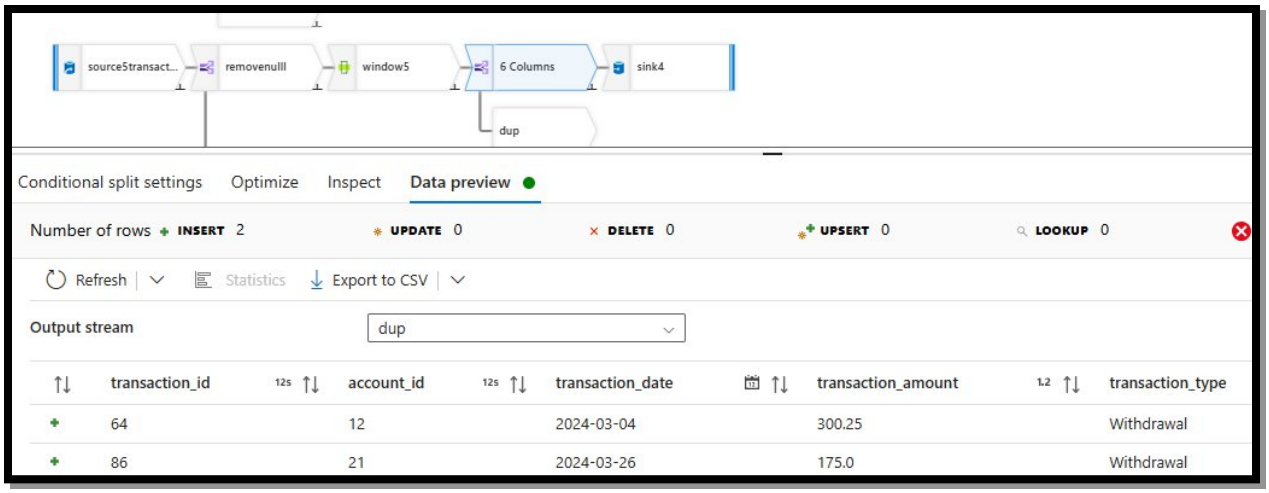


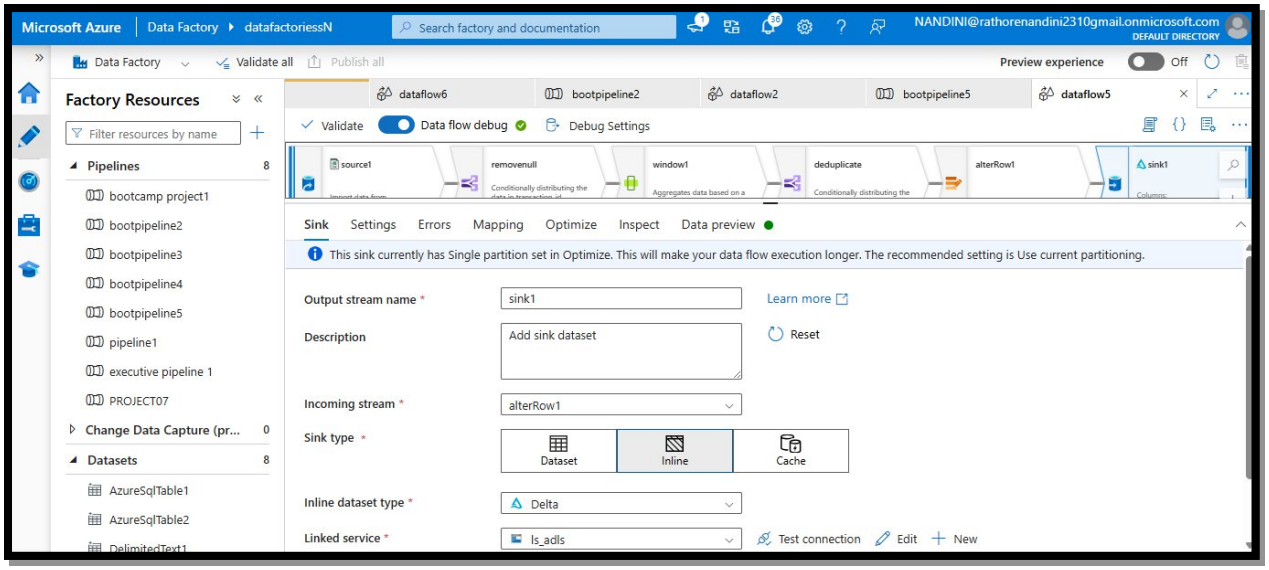


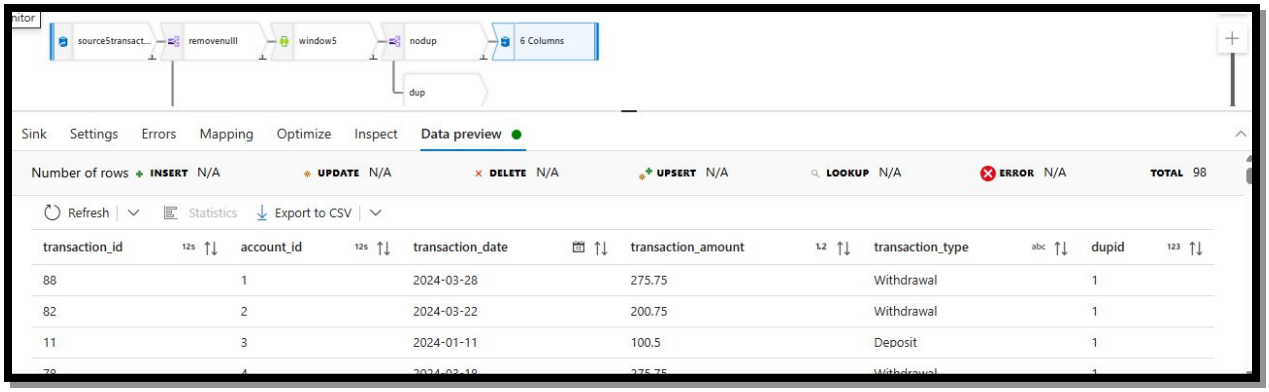
CREATING WINDOW COLUMN. TO REMOVE DUPLICATE VALUES.

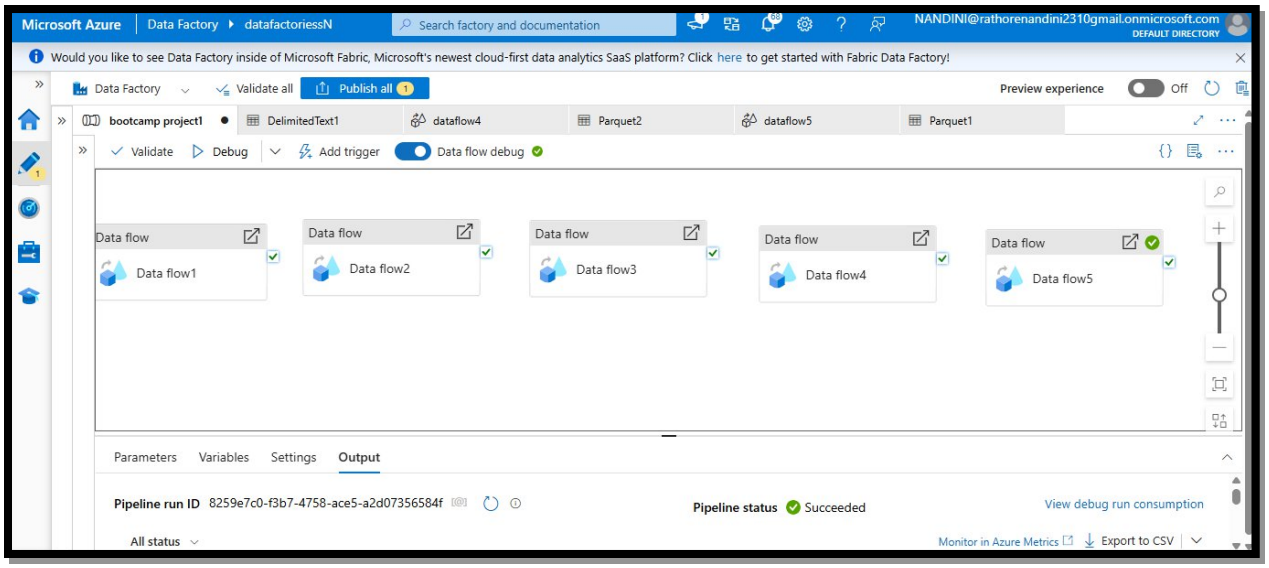



source5transact...
removenulll
window5
6 Columns
dup
Conditional split settings
Optimize
Inspect
Data preview
Incoming stream *
window5
Split on
First matching condition
All matching conditions
Split condition
Stream names
Condition
nodup
dupid==1
dup
Rows that do not meet any condition will use this output stream

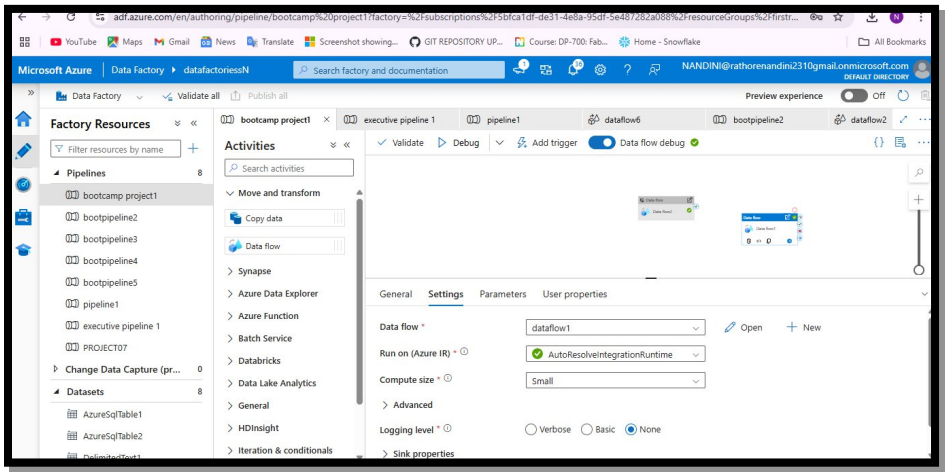

source5transact...
removenulll
window5
6 Columns
sink4
dup
Conditional split settings
Optimize
Inspect
Data preview
Number of rows INSERT 2
UPDATE 0
DELETE 0
UPSERT 0
LOOKUP 0
Refresh
Statistics
Export to CSV
Output stream
dup
transaction_id
account_id
transaction_date
transaction_amount
transaction_type
64
12
2024-03-04
300.25
Withdrawal
86
21
2024-03-26
175.0
Withdrawal


Microsoft Azure
Data Factory
datafactoriessN
Search factory and documentation
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
bootcamp project1
bootpipeline2
bootpipeline3
bootpipeline4
bootpipeline5
pipeline1
executive pipeline 1
PROJECT07
Change Data Capture (pr...
Datasets
AzureSqlTable1
AzureSqlTable2
dataflow6
bootpipeline2
dataflow2
bootpipeline5
dataflow5
Validate
Data flow debug
Debug Settings
source1
removenull
window1
deduplicate
alterRow1
sink1
Sink
Settings
Errors
Mapping
Optimize
Inspect
Data preview
This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The recommended setting is Use current partitioning.
Output stream name *
sink1
Learn more
Description
Add sink dataset
Reset
Incoming stream *
alterRow1
Sink type *
Dataset
Inline
Cache
Inline dataset type *
Delta
Linked service *
ls_adls
Test connection
Edit
New

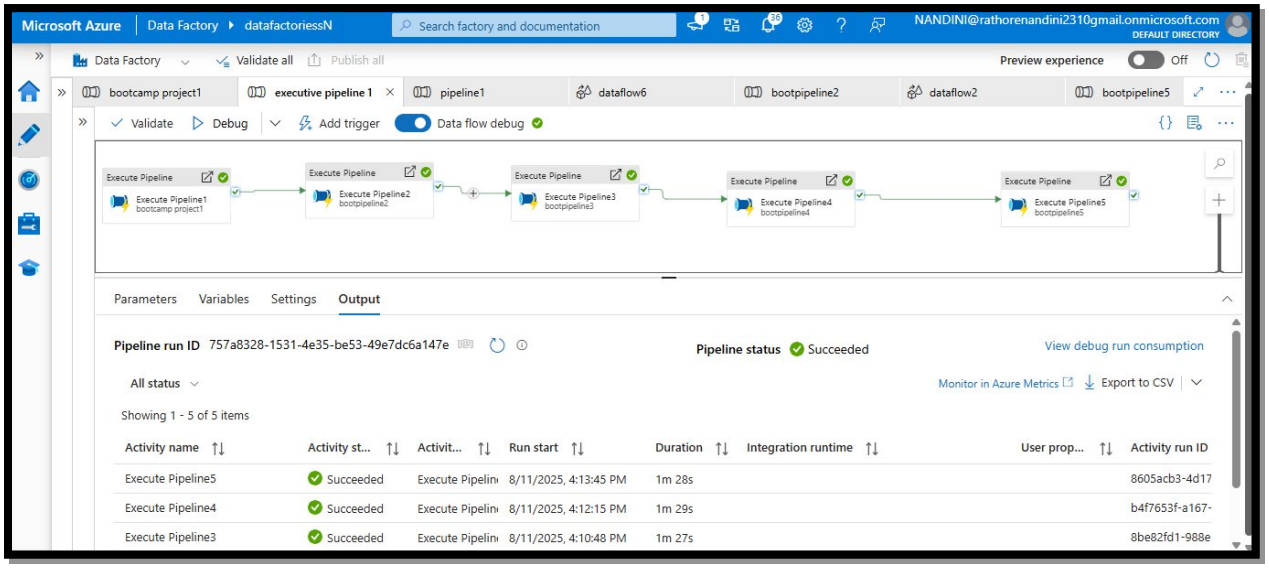


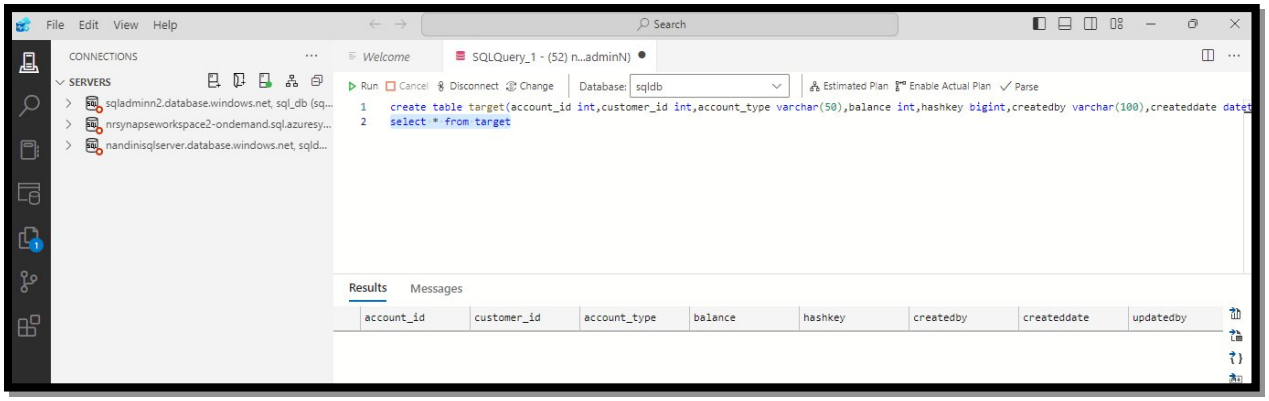


First I used this method for dataflow of 5 different sources.



I used 5 diffent pipelines having different dataflows and run them in a master pipeline to reduce the execution time and connect each pipeline with other and set behavior true . to trigger pipeline means other pipeline executes when previously executed successfully.





Target table creating in sql db.

```sql
create table logging_table
(id int identity(1,1),
tablename varchar(100),
schemaname varchar(100),
error_message varchar(500),
pipeline_status varchar(20),
pipeline_name varchar(100),
pipeline_runid varchar(100),
execution_details varchar(100))
drop  table logging_table
```

Logging table.

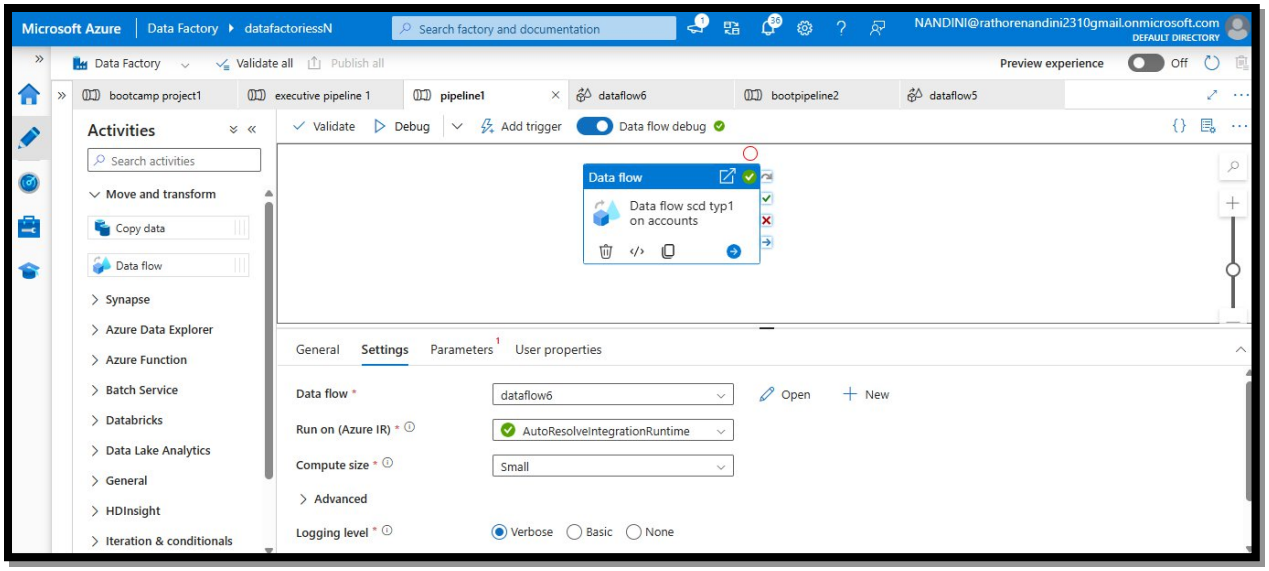
```sql
create procedure log_proc
@tablename varchar(100),
@schemaname varchar(100),
@error_message varchar(1000),
@pipeline_status varchar(20),
@pipeline_name varchar(100),
@pipeline_runid varchar(100),
@execution_details varchar(100)
AS
BEGIN
    SET NOCOUNT ON
insert into logging_table VALUES
(@tablename,
@schemaname,
@error_message,
@pipeline_status,
@pipeline_name,
@pipeline_runid,
@execution_details
)
END

select * from logging_table

drop  procedure log_proc
```

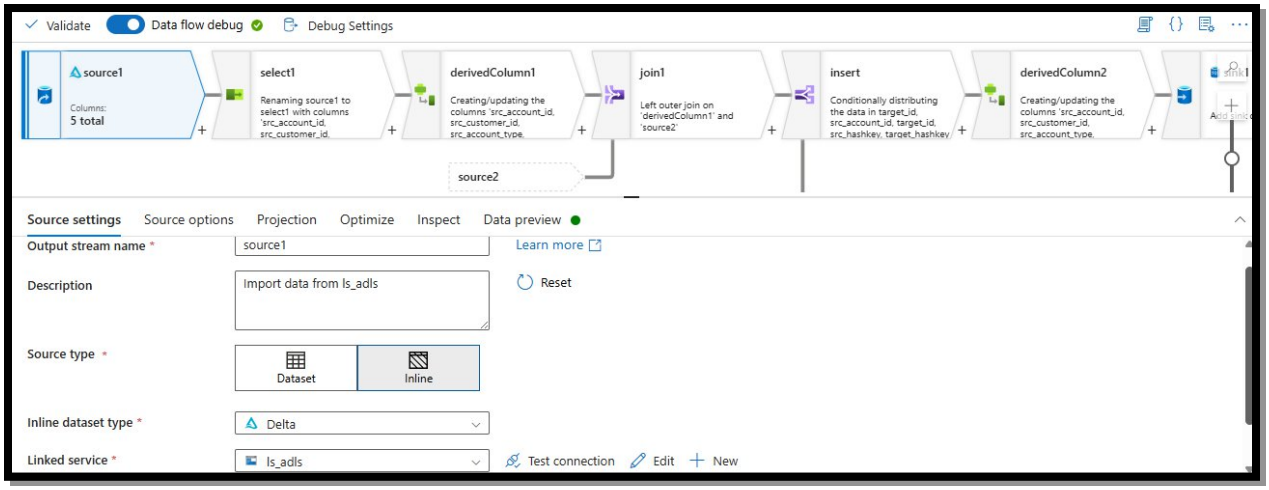
Results Messages

id	tablename	schemaname	error_message	pipeline_status	pipeline_name	pipeline_runid	execution_detai…

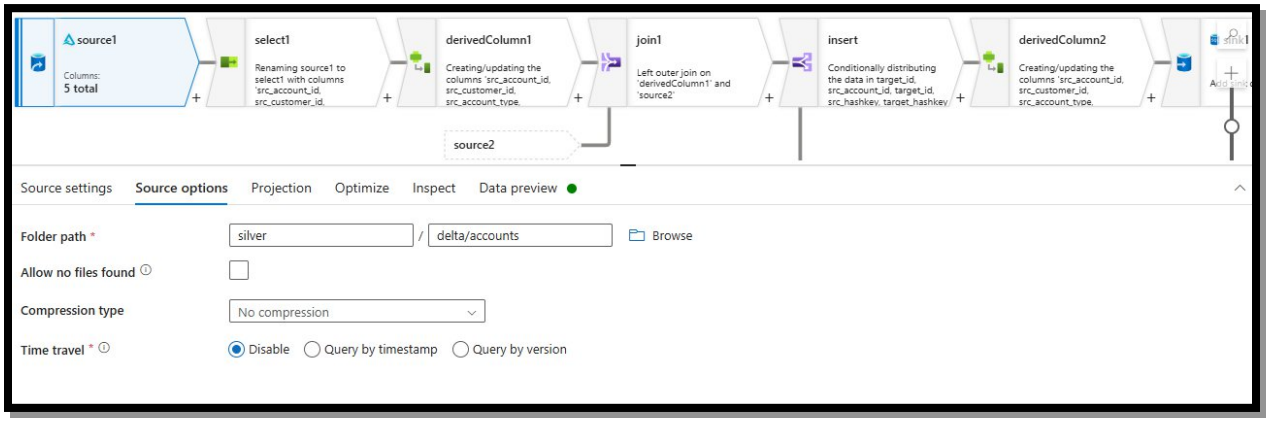
Creating stored procedure.

Scdtype1 :-ACCOUNTS



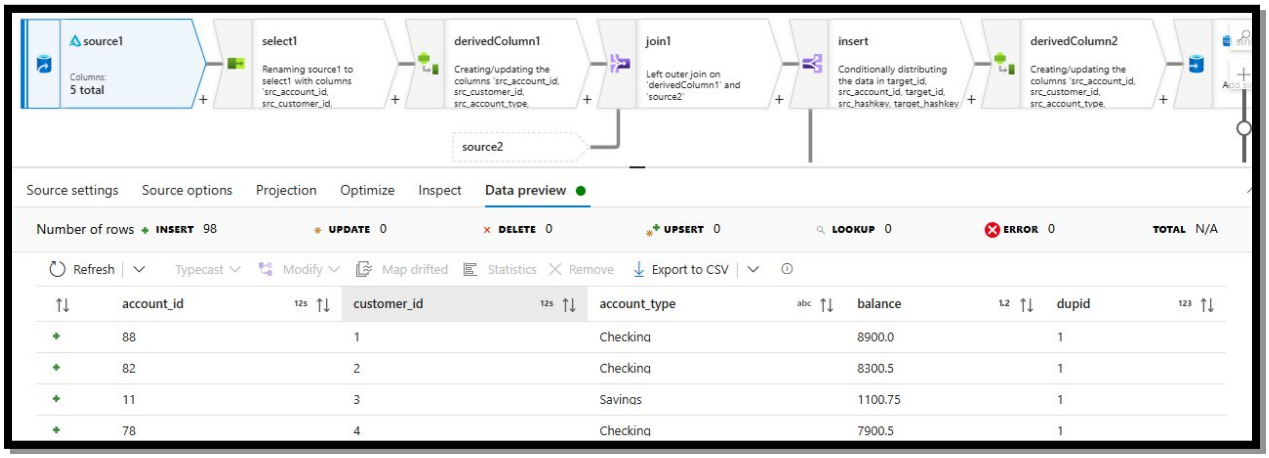


Source:> ACCOUNTS.

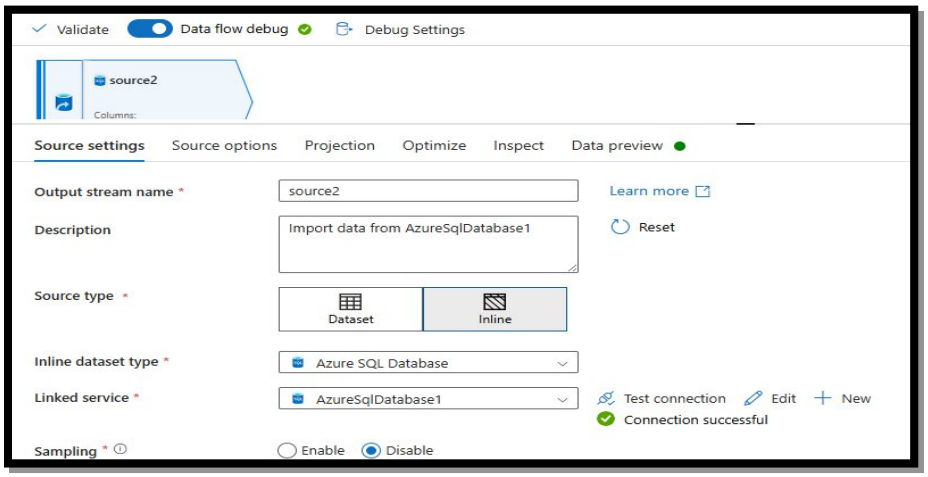


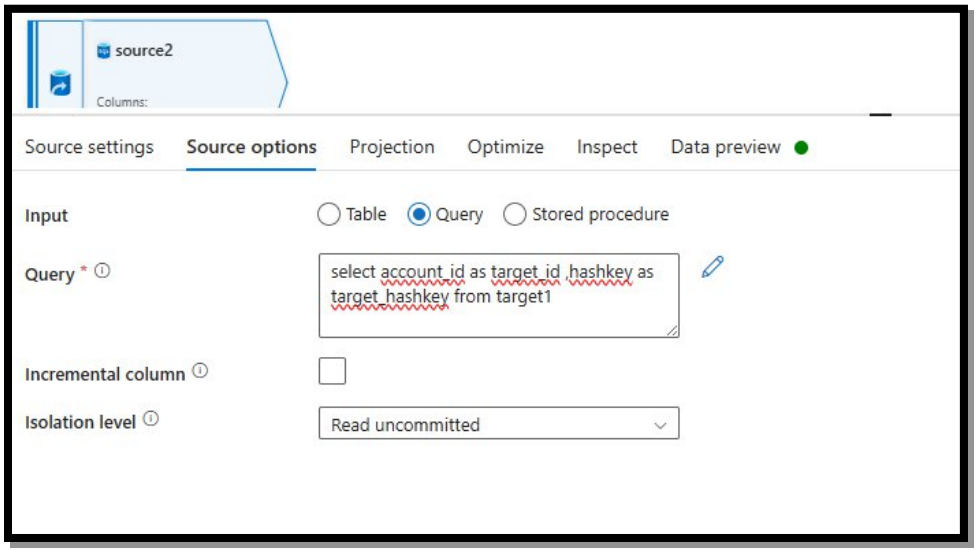
Browsing the accounts file in delta format in silver container in adlsgen2.

Import schema.



Target table creating in sqldb.





Using query

"**select account_id as target_id,hashkey as target_hashkey from target1**"

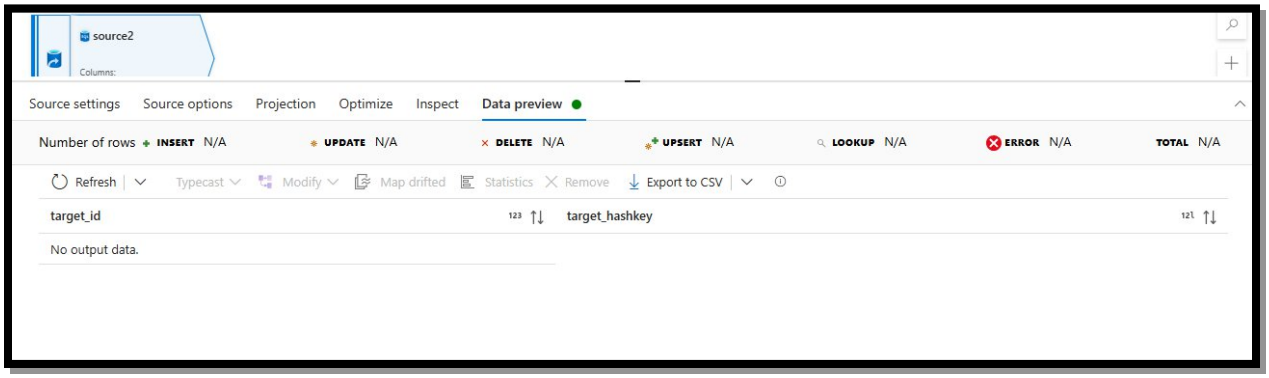
SELECT-→Specifies which columns you want to retrieve from the table.

Account_id AS target_id-→The column id will be renamed (aliased) as target_id in the output.

hashkey AS target_hashkey-→The column hashkey will be renamed as target_hashkey in the output.

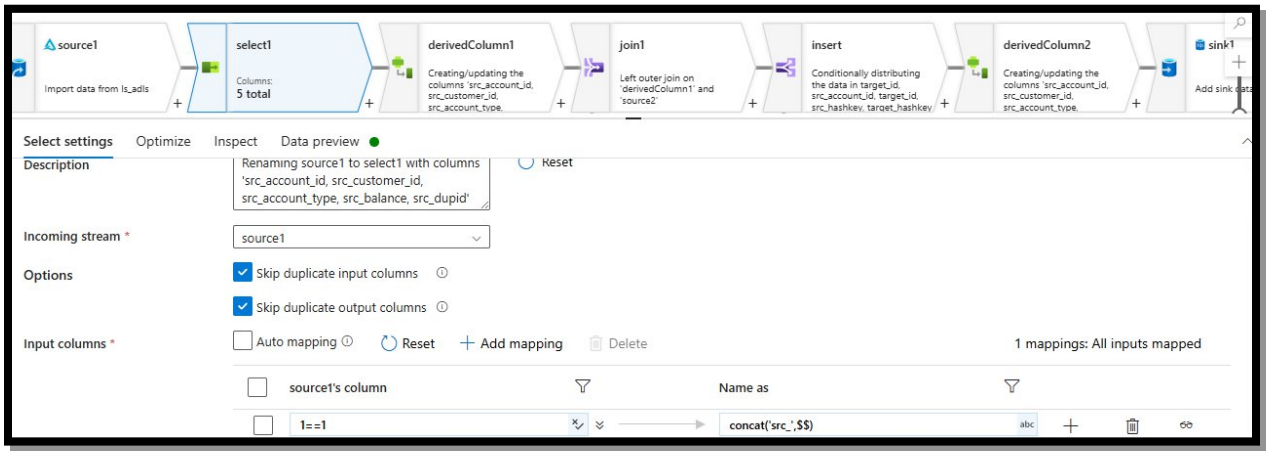
FROM target1-→The data is being fetched from the table named target1 .

THEN IMPORT SCHEMA,OPTIMIZE-USE CURRENT PARTITIONING,INSPECT AND DATA PREVIEW.



To clearly differentiate source vs target fields — especially useful in ETL pipelines, SCD Data Flows, or comparison queries.

Makes output easier to understand — you know these are "target" values.



USE OF 1==1:> Always True Condition.

Used in joins/lookups to match all rows when no specific key is needed.

Helps create Cross Joins or dummy conditions.

Example:

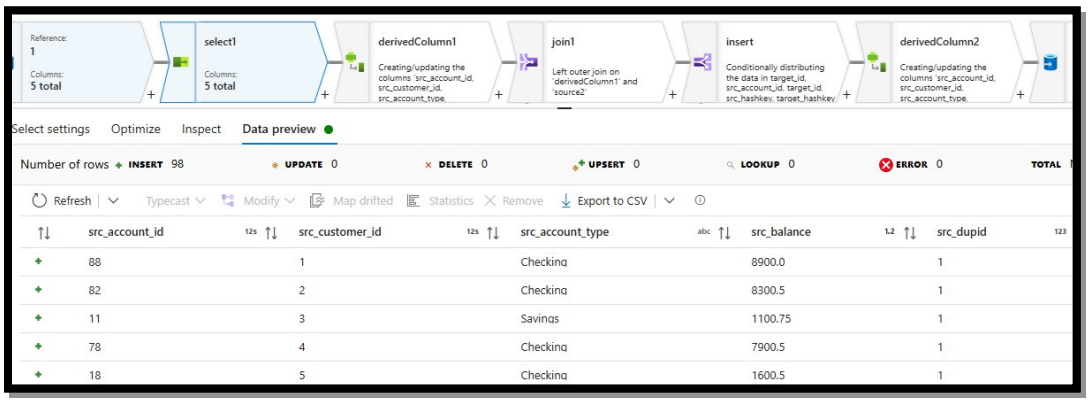
1==1 ensures every row matches every other row. Means every row of source matches to target row.

Use of concat('src_',$$):> To dynamically build strings in expressions.

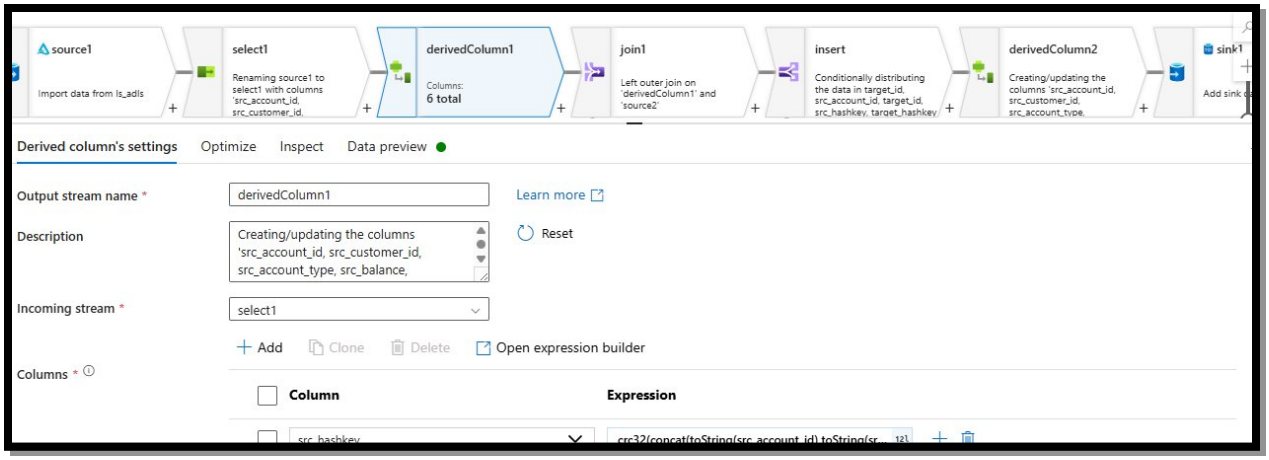
$$ is a parameter placeholder — its value comes at runtime.Useful for making column names or strings

dynamic.Example:

If $$ = 'id' then: concat('src_', $$) ➡ 'src_id'.



Generating hashkey:-

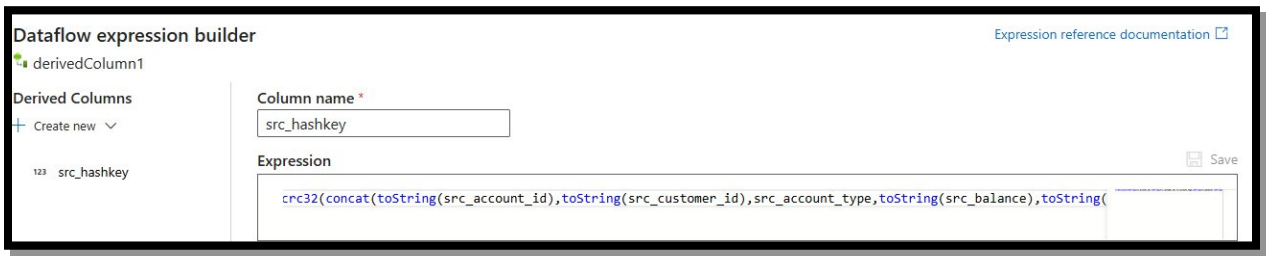


Hashkey is the unique number that represents the combines value of coloumns.

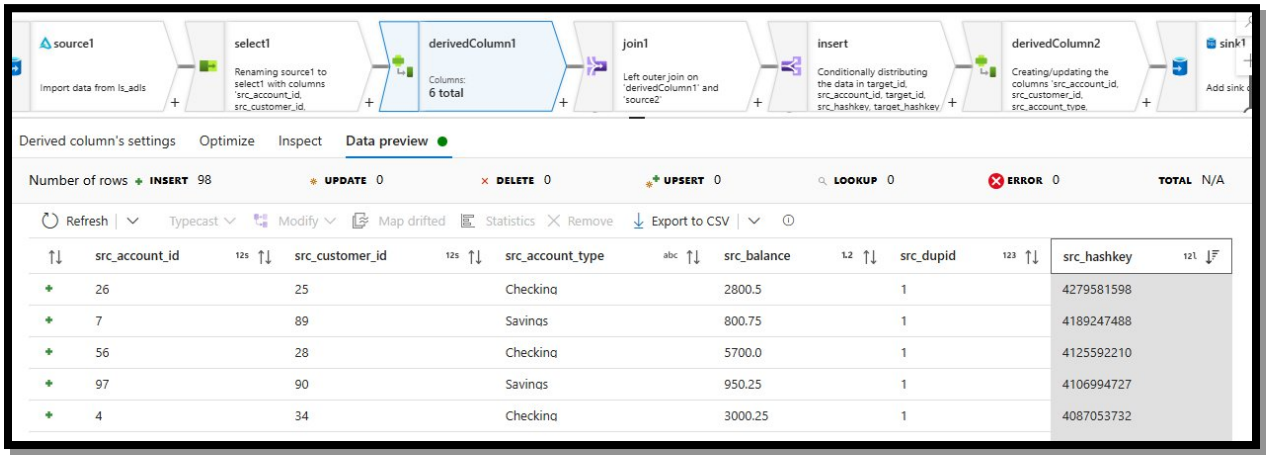
It detects if any value in those number is change if changes,data changed.

Concat is used to join all the coloumns, toString is used to convert numbers into string.

Crc32 hash function which helps to create a unique numeric value (hashkey).

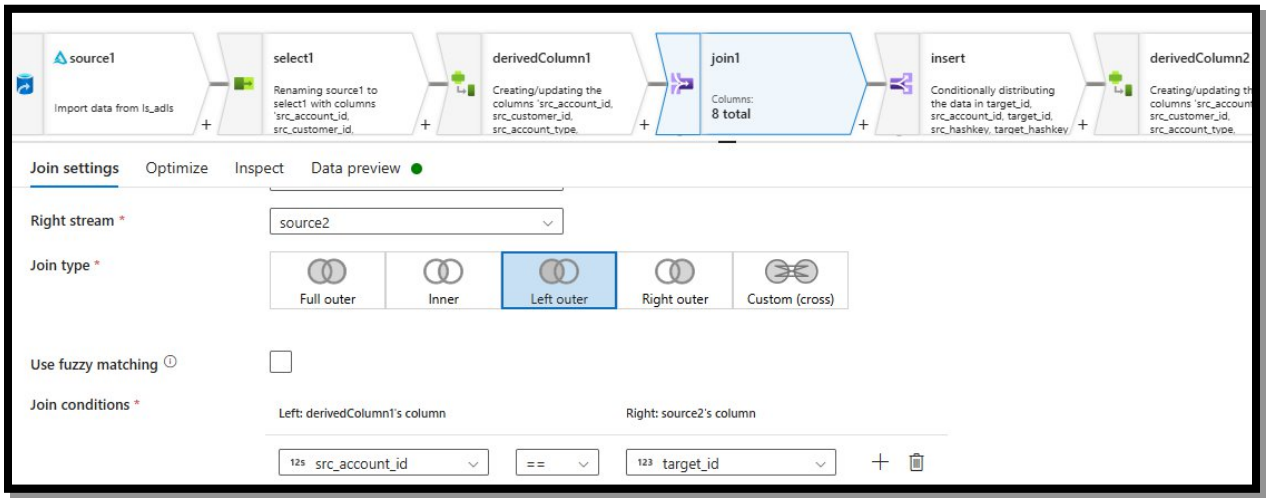


Use current partitioning to optimize value.

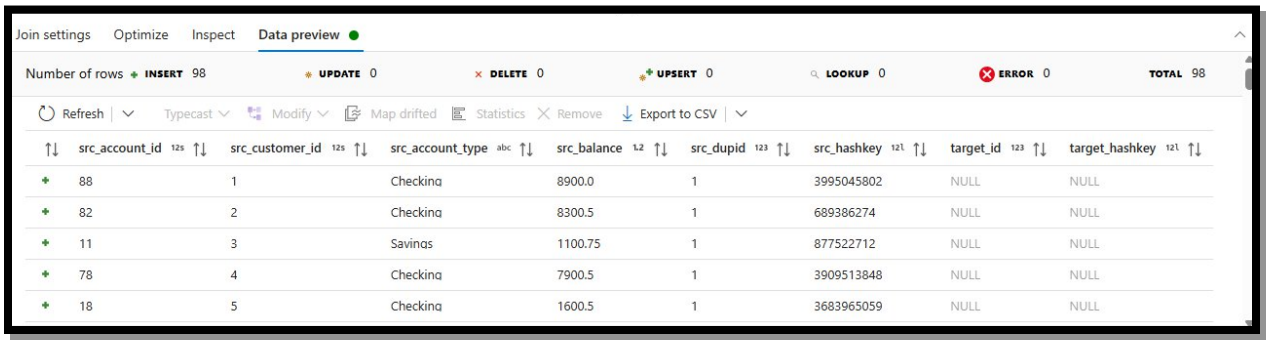


Data preview.

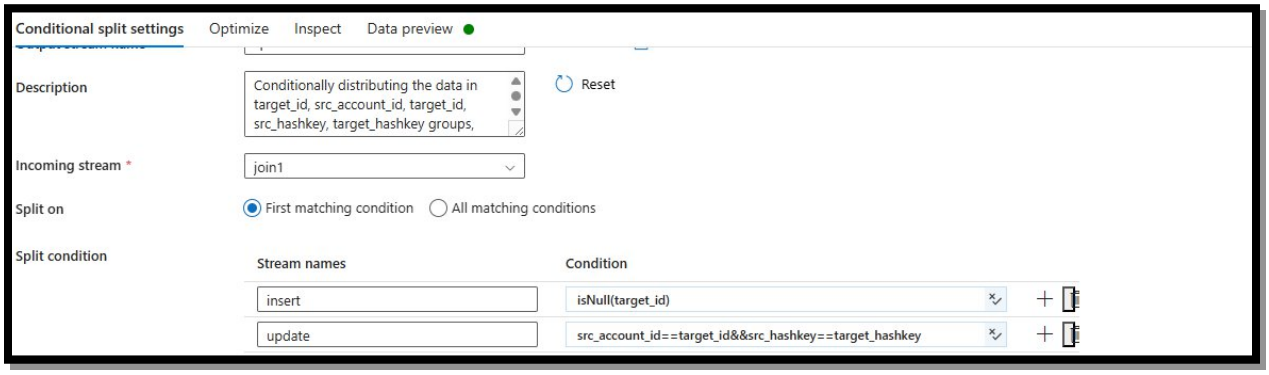
JOIN:- left outer join of delta files with target table.

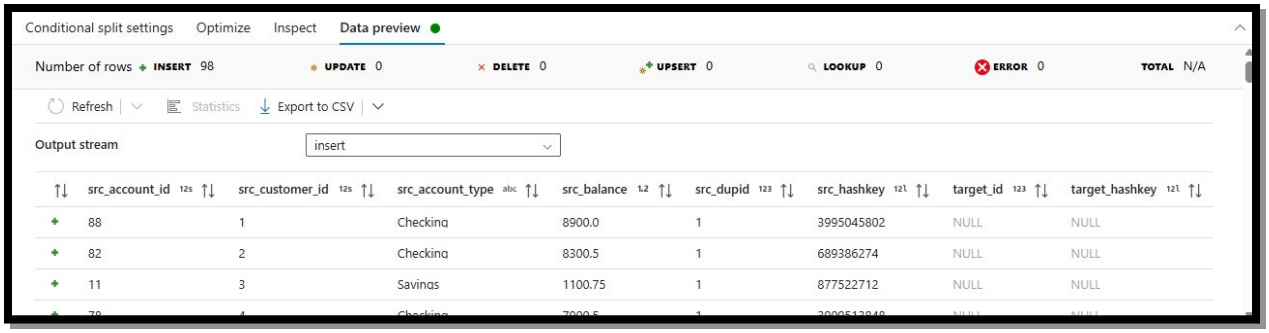


Src_customer_id==target_id mean source equals to target on the bases of id.



src_account_id	src_customer_id	src_account_type	src_balance	src_dupid	src_hashkey	target_id	target_hashkey
88	1	Checking	8900.0	1	3995045802	NULL	NULL
82	2	Checking	8300.5	1	689386274	NULL	NULL
11	3	Savings	1100.75	1	877522712	NULL	NULL
78	4	Checking	7900.5	1	3909513848	NULL	NULL
18	5	Checking	1600.5	1	3683965059	NULL	NULL





Insert:> isNull(target_id)

Update:> src_account_id==target_id&&src_hashkey==target_hashkey

For the first run: Insert Condition is TRUE — because target_id is null (record does not exist in target).

Update Condition is FALSE — because nothing matches (no old record yet).

First time run = Only Insert works (because target is empty).

✔Update is empty because no matching records yet.

✔Next runs = Update will work when changes are detected.

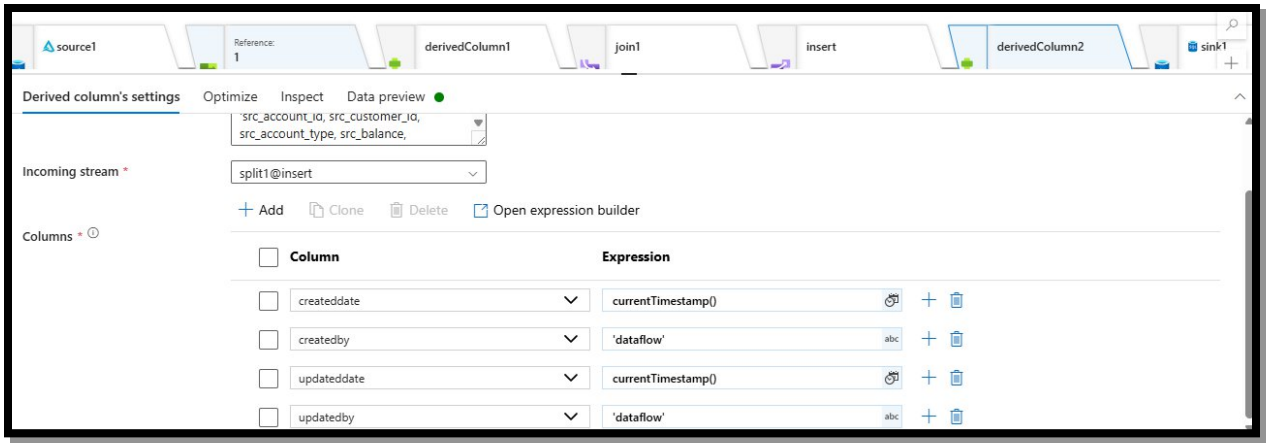
isNull(target_id) checks if the target record does not exist (means no matching id found in the target table).

used to identify new records that should be inserted into the target.

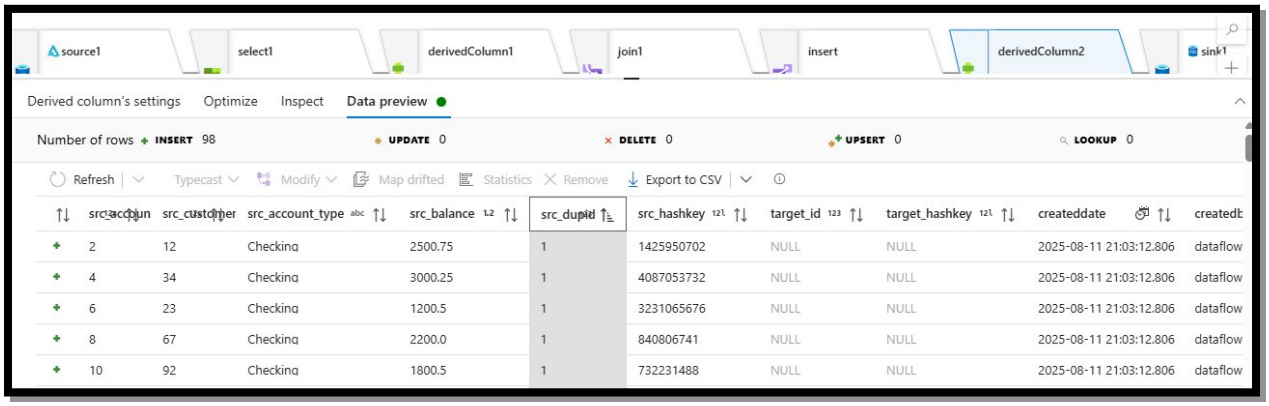
src_account_id == target_id && src_hashkey == target_hashkey checks if the record exists in the target and the data

has not changed.

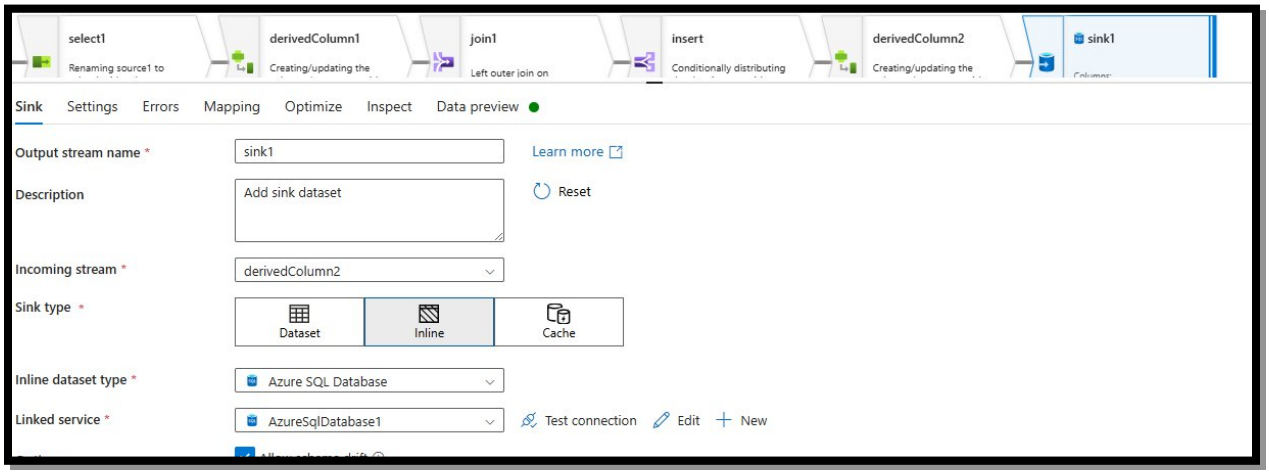
this condition usually means **"No Update Needed"** — because both ID and data (hashkey) are the same.

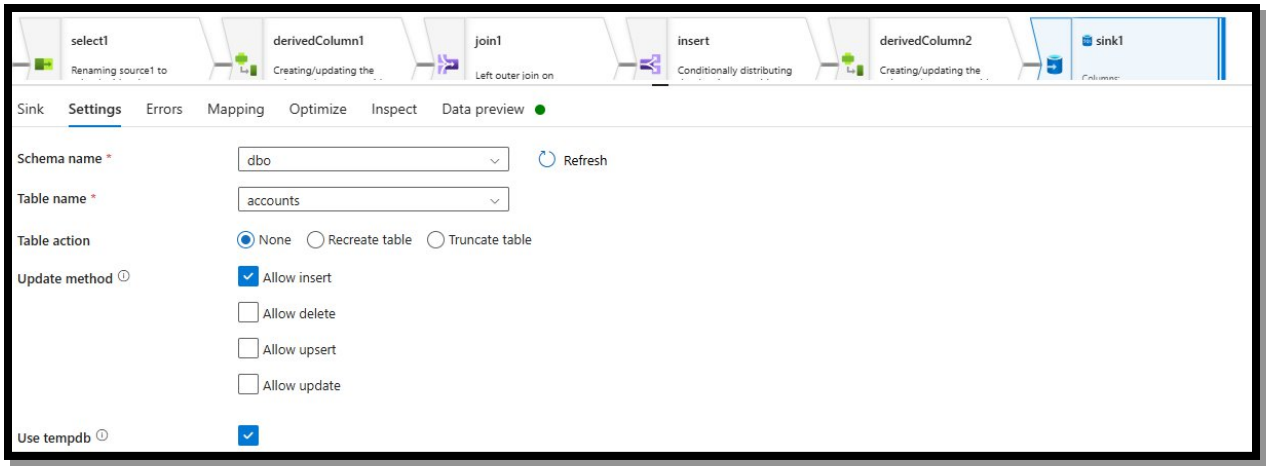


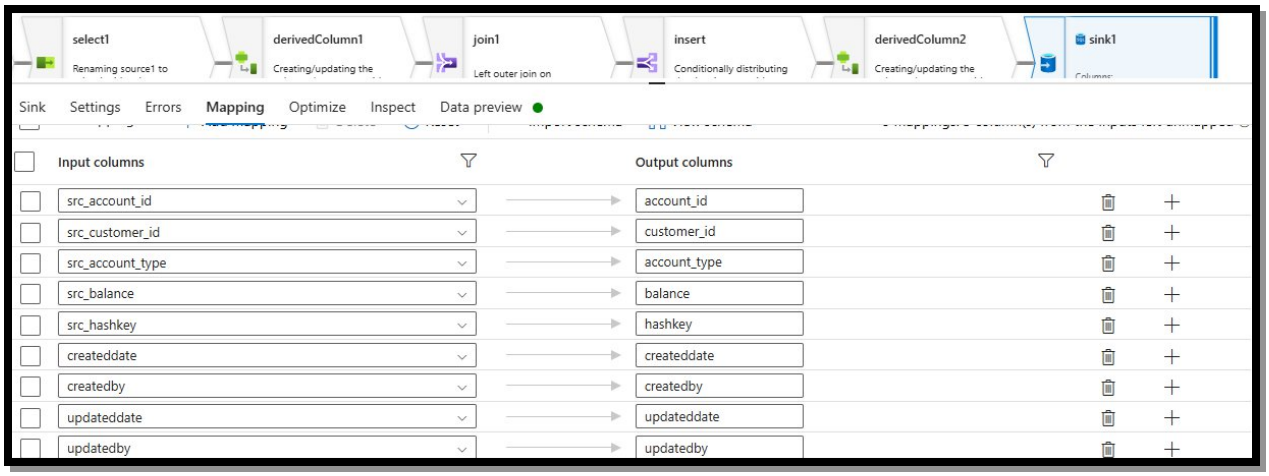
Adding new columns.

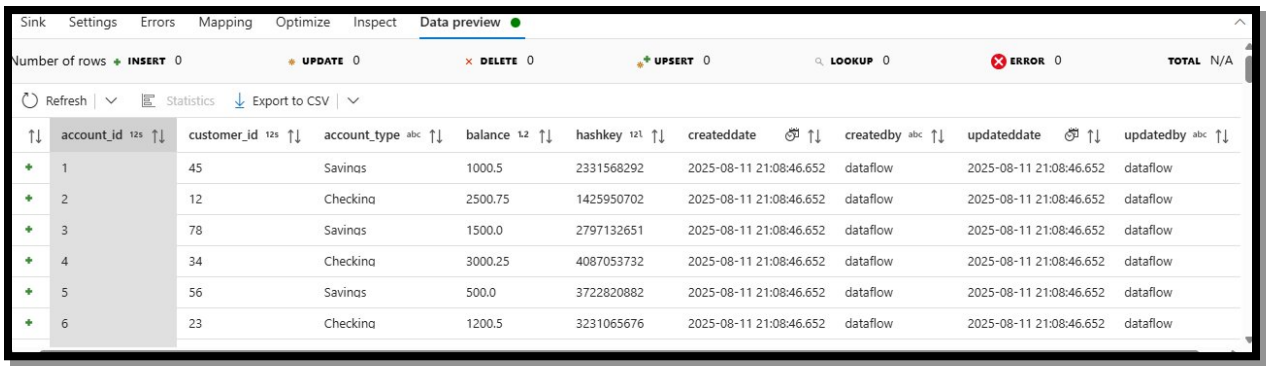


Sink:

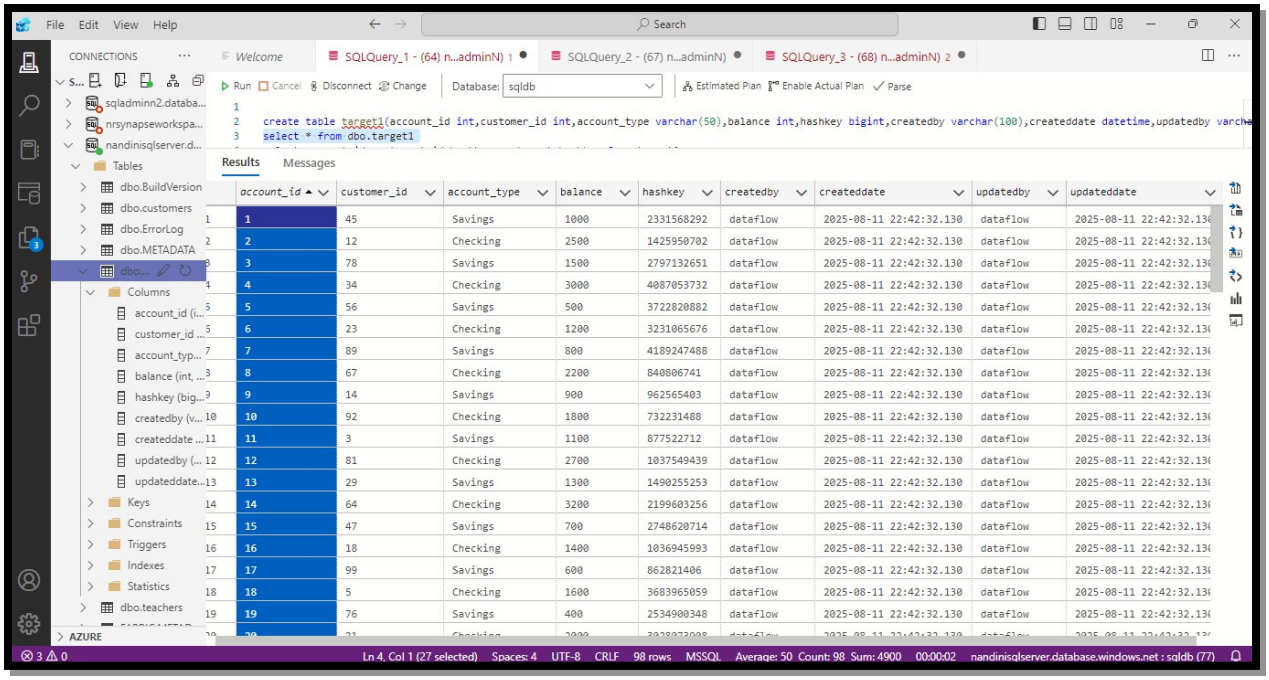


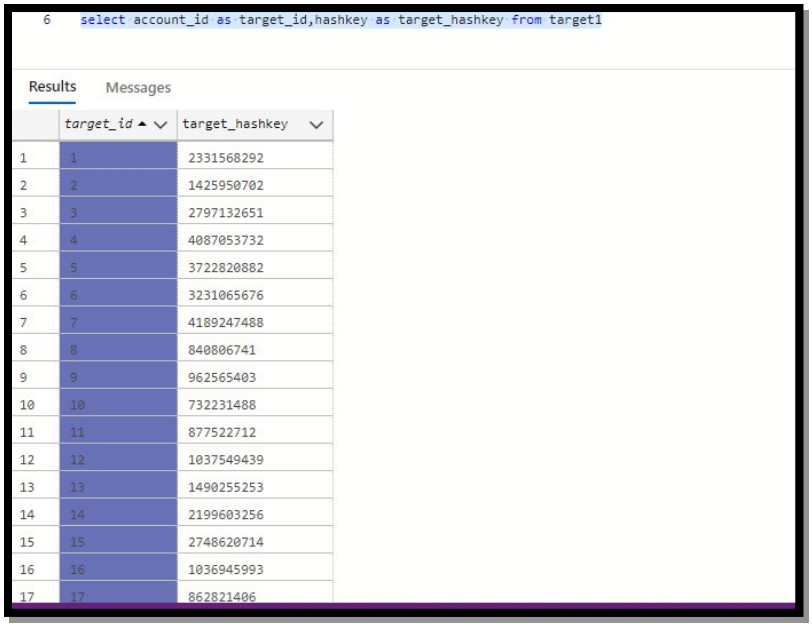






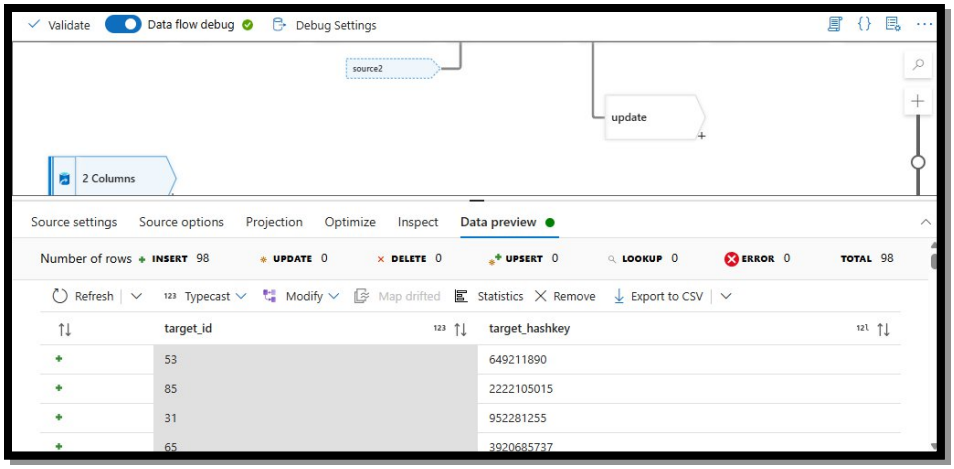
account_id	customer_id	account_type	balance	hashkey	createddate	createdby	updateddate	updatedby
1	45	Savings	1000.5	2331568292	2025-08-11 21:08:46.652	dataflow	2025-08-11 21:08:46.652	dataflow
2	12	Checking	2500.75	1425950702	2025-08-11 21:08:46.652	dataflow	2025-08-11 21:08:46.652	dataflow
3	78	Savings	1500.0	2797132651	2025-08-11 21:08:46.652	dataflow	2025-08-11 21:08:46.652	dataflow
4	34	Checking	3000.25	4087053732	2025-08-11 21:08:46.652	dataflow	2025-08-11 21:08:46.652	dataflow
5	56	Savings	500.0	3722820882	2025-08-11 21:08:46.652	dataflow	2025-08-11 21:08:46.652	dataflow
6	23	Checking	1200.5	3231065676	2025-08-11 21:08:46.652	dataflow	2025-08-11 21:08:46.652	dataflow

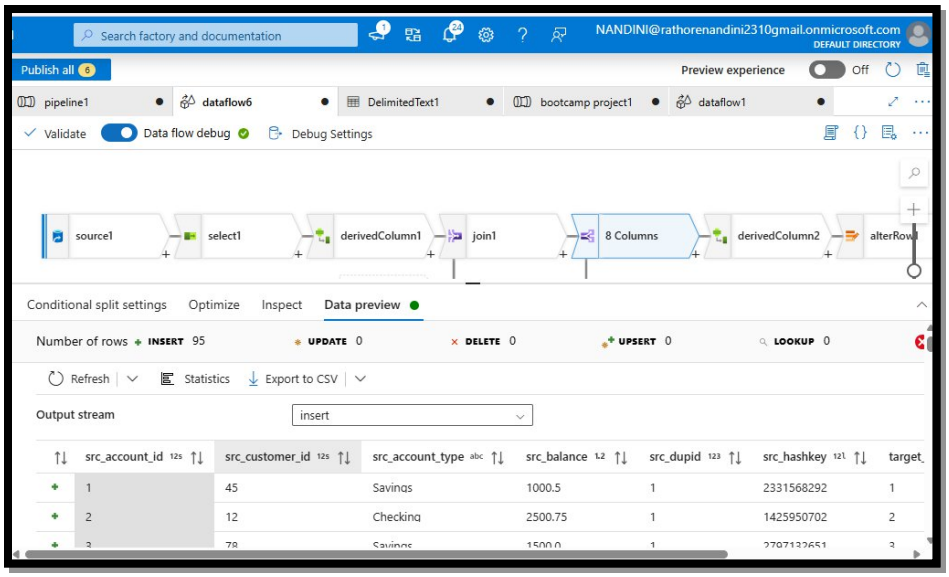


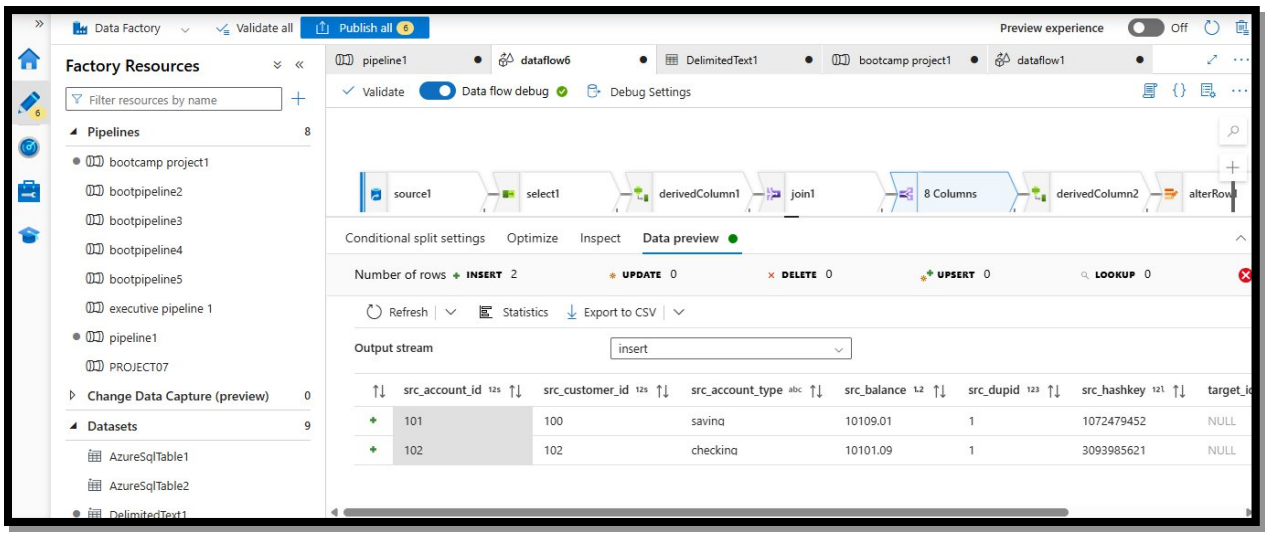


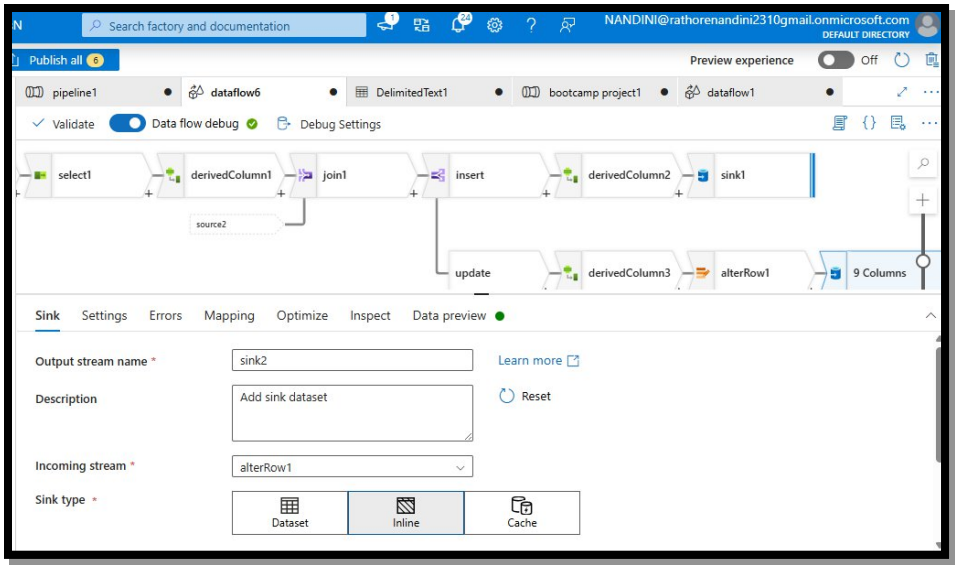
Logging table;-

Update table:>

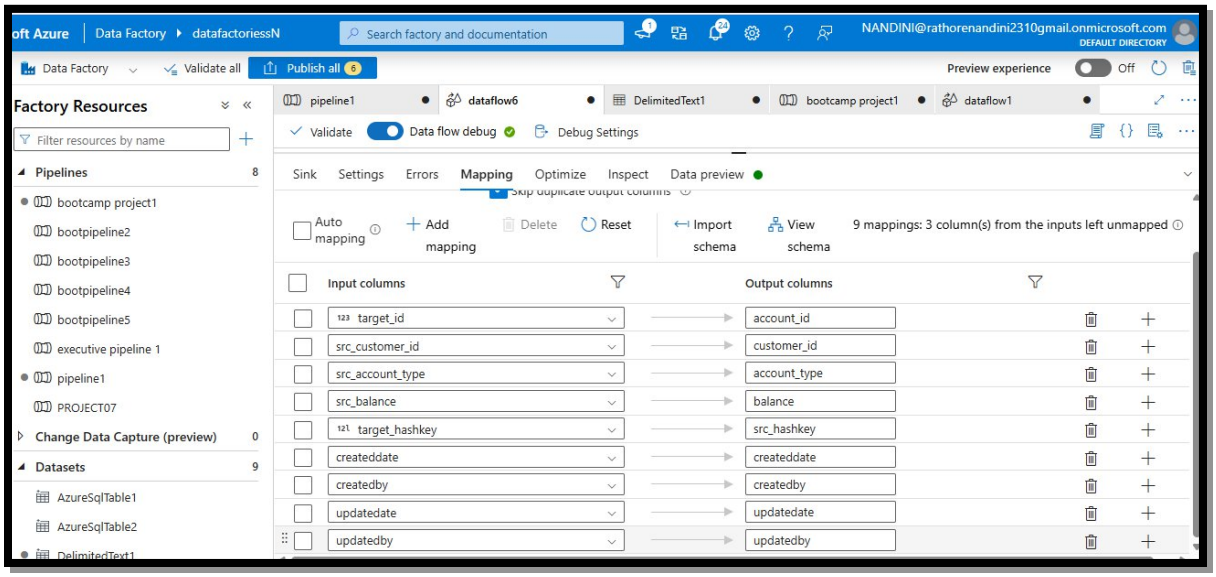

Validate
Data flow debug
Debug Settings
source2
update
2 Columns
Source settings
Source options
Projection
Optimize
Inspect
Data preview
Number of rows INSERT 98 UPDATE 0 DELETE 0 UPSERT 0 LOOKUP 0 ERROR 0 TOTAL 98
Refresh
Typecast
Modify
Map drifted
Statistics
Remove
Export to CSV
target_id
target_hashkey
53 649211890
85 2222105015
31 952281255
65 3920685737


Search factory and documentation
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Publish all
Preview experience
Off
pipeline1
dataflow6
DelimitedText1
bootcamp project1
dataflow1
Validate
Data flow debug
Debug Settings
source1
select1
derivedColumn1
join1
8 Columns
derivedColumn2
alterRow1
Conditional split settings
Optimize
Inspect
Data preview
Number of rows INSERT 95 UPDATE 0 DELETE 0 UPSERT 0 LOOKUP 0
Refresh
Statistics
Export to CSV
Output stream
insert
src_account_id
src_customer_id
src_account_type
src_balance
src_dupid
src_hashkey
target_
1 45 Savings 1000.5 1 2331568292 1
2 12 Checking 2500.75 1 1425950702 2

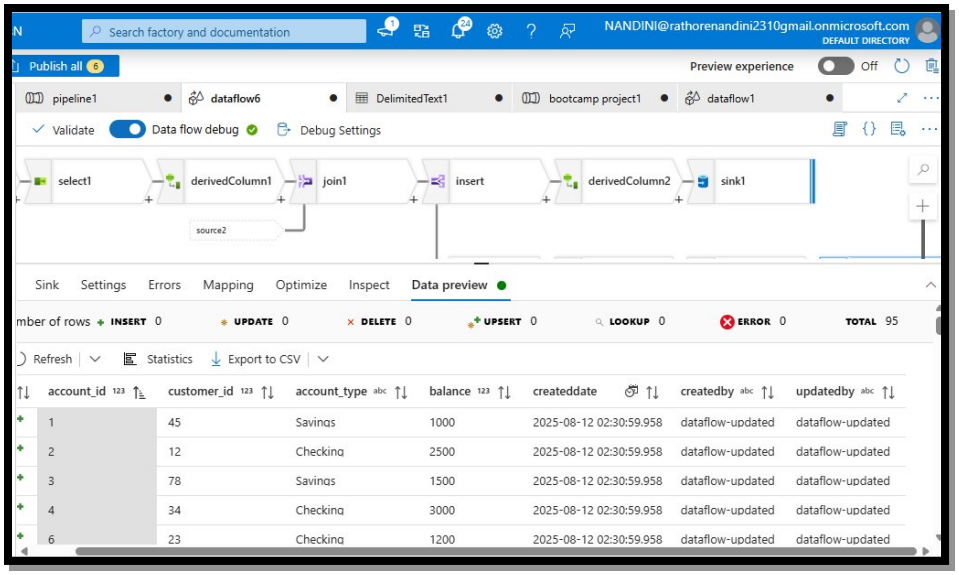

Data Factory
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 8
bootcamp project1
bootpipeline2
bootpipeline3
bootpipeline4
bootpipeline5
executive pipeline 1
pipeline1
PROJECT07
Change Data Capture (preview) 0
Datasets 9
AzureSqlTable1
AzureSqlTable2
DelimitedText1
pipeline1
dataflow6
DelimitedText1
bootcamp project1
dataflow1
Validate
Data flow debug
Debug Settings
source1
select1
derivedColumn1
join1
8 Columns
derivedColumn2
alterRow1
Conditional split settings
Optimize
Inspect
Data preview
Number of rows INSERT 2 UPDATE 0 DELETE 0 UPSERT 0 LOOKUP 0
Refresh
Statistics
Export to CSV
Output stream
insert
src_account_id
src_customer_id
src_account_type
src_balance
src_dupid
src_hashkey
target_id
101 100 saving 10109.01 1 1072479452 NULL
102 102 checking 10101.09 1 3093985621 NULL



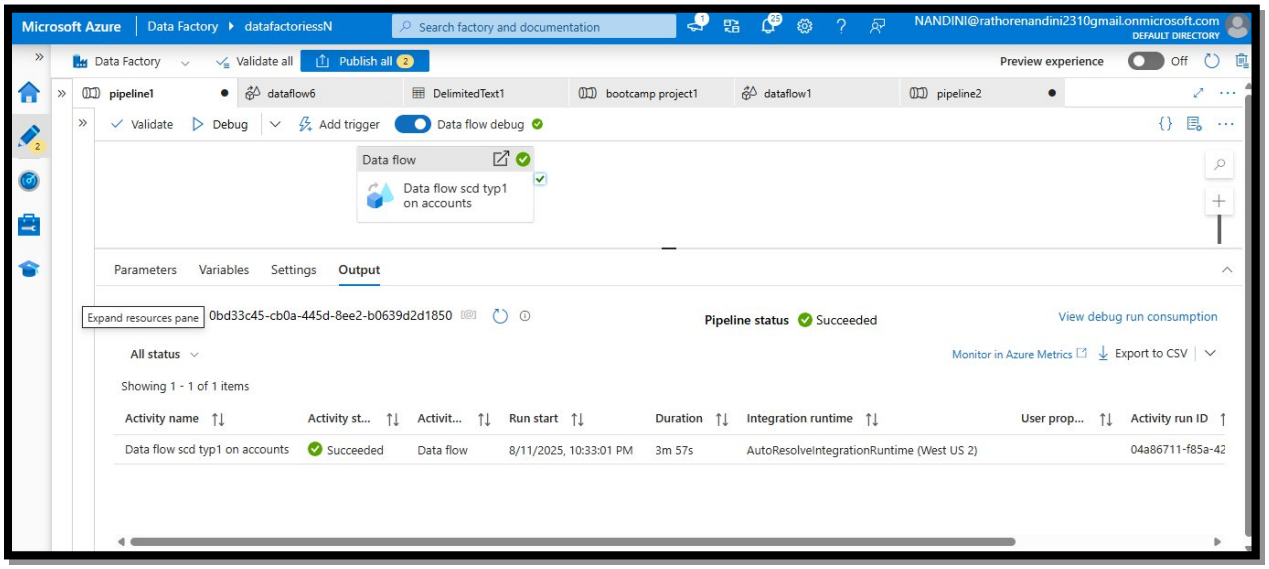
Data preview->

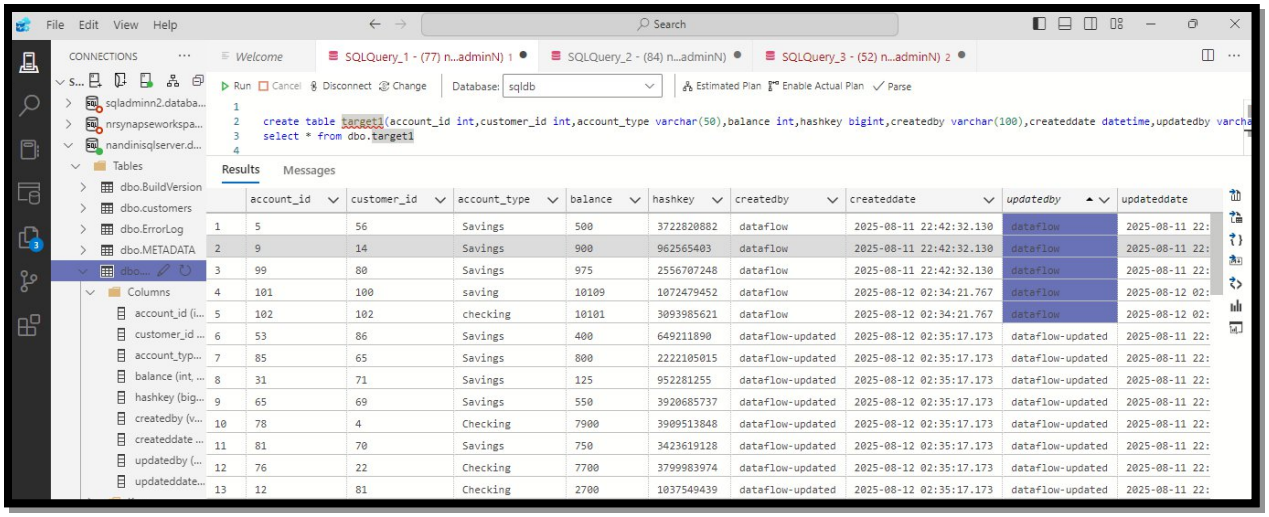


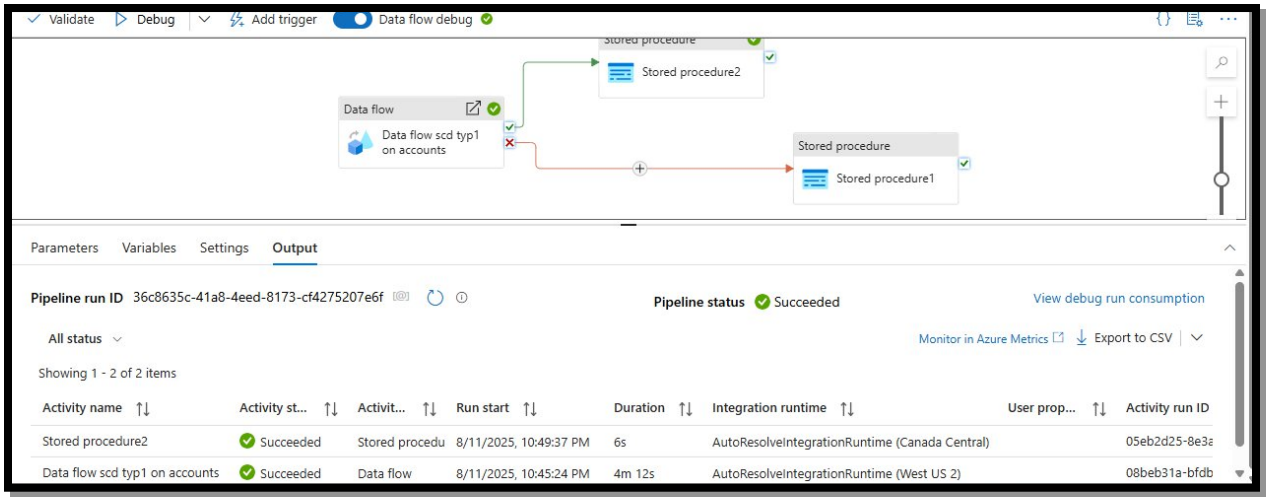
Mapping of update coloumns.

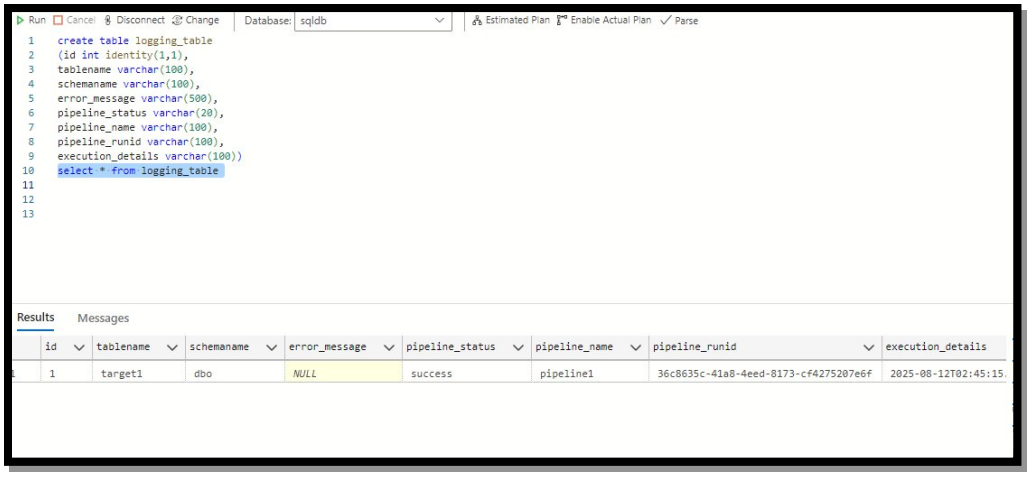


Now run the pipeline and check table in sqldb.



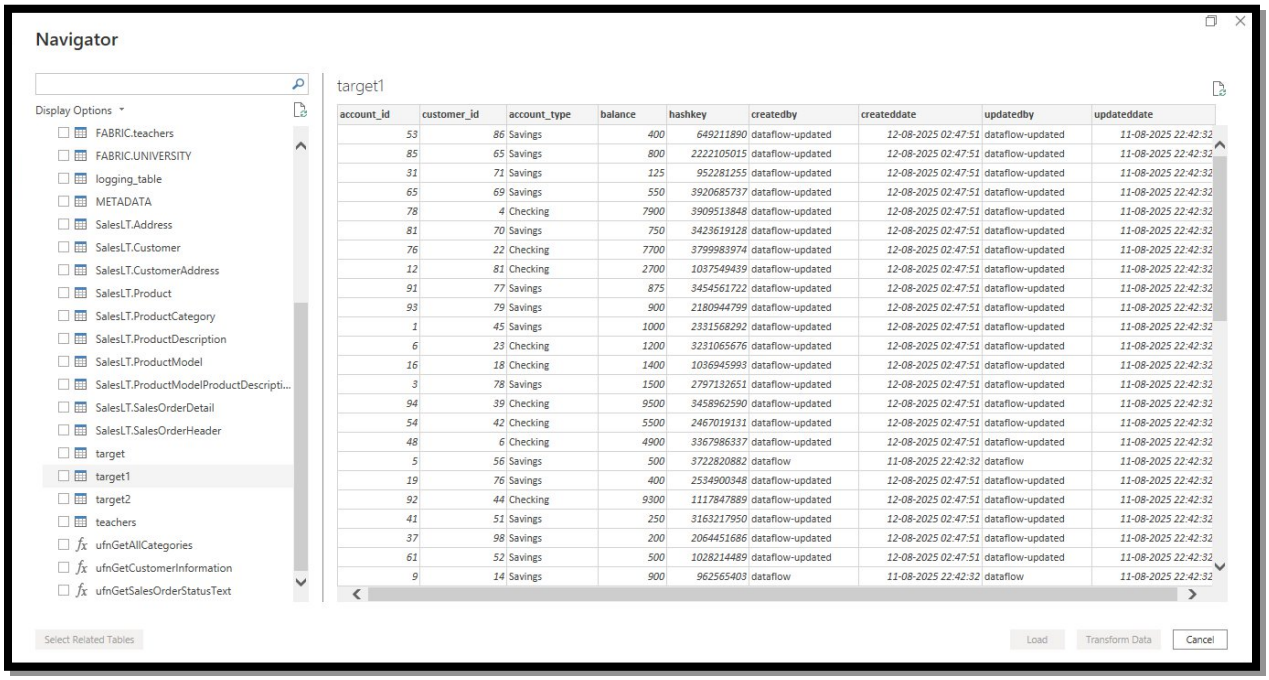

File Edit View Help
Search
CONNECTIONS
Welcome
SQLQuery_1 - (77) n...adminN) 1
SQLQuery_2 - (84) n...adminN)
SQLQuery_3 - (52) n...adminN) 2
Run Cancel Disconnect Change
Database: sqldb
Estimated Plan Enable Actual Plan Parse
create table target1(account_id int,customer_id int,account_type varchar(50),balance int,hashkey bigint,createdby varchar(100),createddate datetime,updatedby varcha
select * from dbo.target1
Results
Messages
Tables
dbo.BuildVersion
dbo.customers
dbo.ErrorLog
dbo.METADATA
Columns
account_id (i...
customer_id ...
account_typ...
balance (int, ...
hashkey (big...
createdby (v...
createddate ...
updatedby (...
updateddate...
account_id customer_id account_type balance hashkey createdby createddate updatedby updateddate
1 5 56 Savings 500 3722820882 dataflow 2025-08-11 22:42:32.130 dataflow 2025-08-11 22:
2 9 14 Savings 900 962565403 dataflow 2025-08-11 22:42:32.130 dataflow 2025-08-11 22:
3 99 80 Savings 975 2556707248 dataflow 2025-08-11 22:42:32.130 dataflow 2025-08-11 22:
4 101 100 saving 10109 1072479452 dataflow 2025-08-12 02:34:21.767 dataflow 2025-08-12 02:
5 102 102 checking 10101 3093985621 dataflow 2025-08-12 02:34:21.767 dataflow 2025-08-12 02:
6 53 86 Savings 400 649211890 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
7 85 65 Savings 800 2222105015 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
8 31 71 Savings 125 952281255 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
9 65 69 Savings 550 3920685737 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
10 78 4 Checking 7900 3909513848 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
11 81 70 Savings 750 3423619128 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
12 76 22 Checking 7700 3799983974 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:
13 12 81 Checking 2700 1037549439 dataflow-updated 2025-08-12 02:35:17.173 dataflow-updated 2025-08-11 22:


Validate Debug Add trigger Data flow debug
Stored procedure
Stored procedure2
Data flow
Data flow scd typ1 on accounts
Stored procedure
Stored procedure1
Parameters Variables Settings Output
Pipeline run ID 36c8635c-41a8-4eed-8173-cf4275207e6f
Pipeline status Succeeded
View debug run consumption
All status
Monitor in Azure Metrics
Export to CSV
Showing 1 - 2 of 2 items
Activity name Activity st... Activit... Run start Duration Integration runtime User prop... Activity run ID
Stored procedure2 Succeeded Stored procedu 8/11/2025, 10:49:37 PM 6s AutoResolveIntegrationRuntime (Canada Central) 05eb2d25-8e3a
Data flow scd typ1 on accounts Succeeded Data flow 8/11/2025, 10:45:24 PM 4m 12s AutoResolveIntegrationRuntime (West US 2) 08beb31a-bfdb

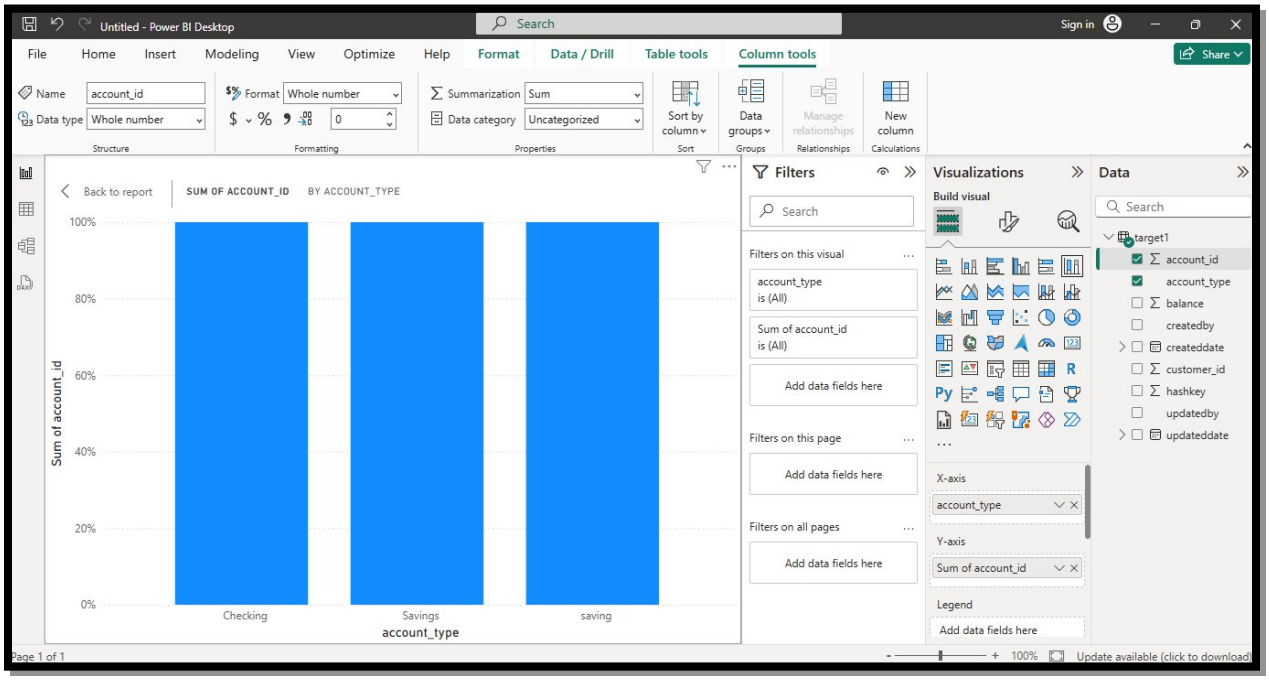

Run Cancel Disconnect Change
Database: sqldb
Estimated Plan Enable Actual Plan Parse
create table logging_table
(id int identity(1,1),
tablename varchar(100),
schemaname varchar(100),
error_message varchar(500),
pipeline_status varchar(20),
pipeline_name varchar(100),
pipeline_runid varchar(100),
execution_details varchar(100))
select * from logging_table
Results
Messages
id tablename schemaname error_message pipeline_status pipeline_name pipeline_runid execution_details
1 target1 dbo NULL success pipeline1 36c8635c-41a8-4eed-8173-cf4275207e6f 2025-08-12T02:45:15.

NANDINI RATHORE

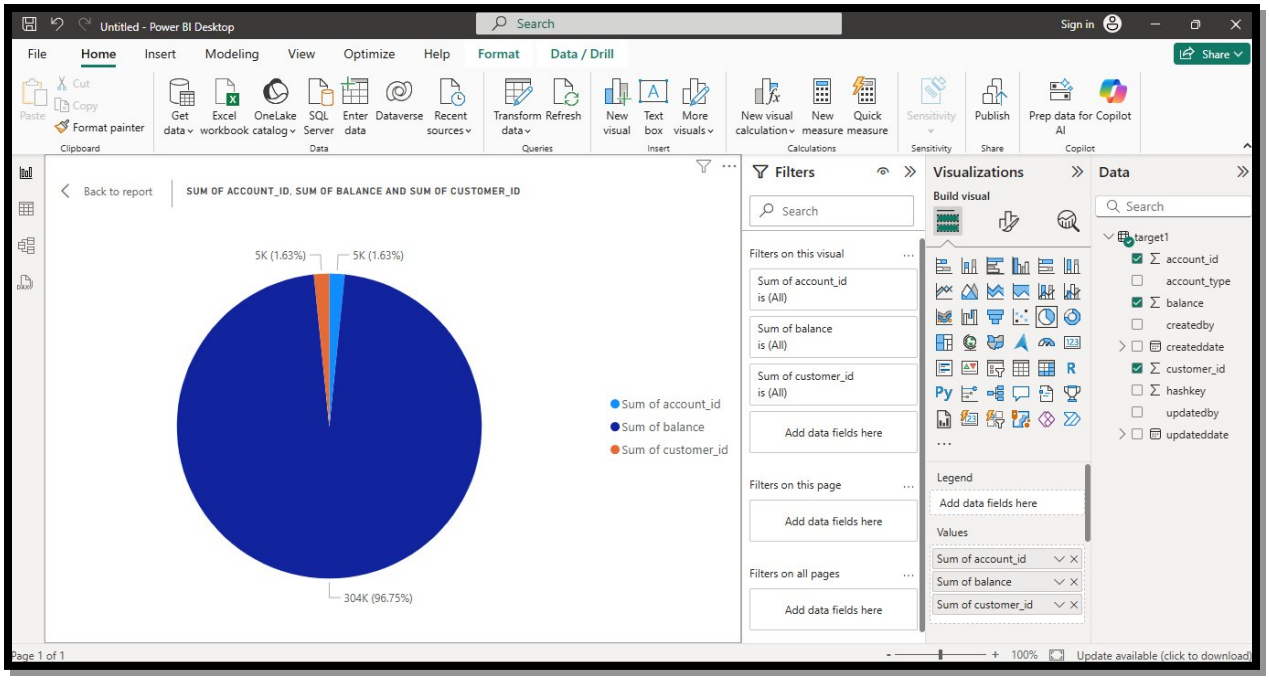
POWER BI:-



SELECTED TARGET1 TABLE.

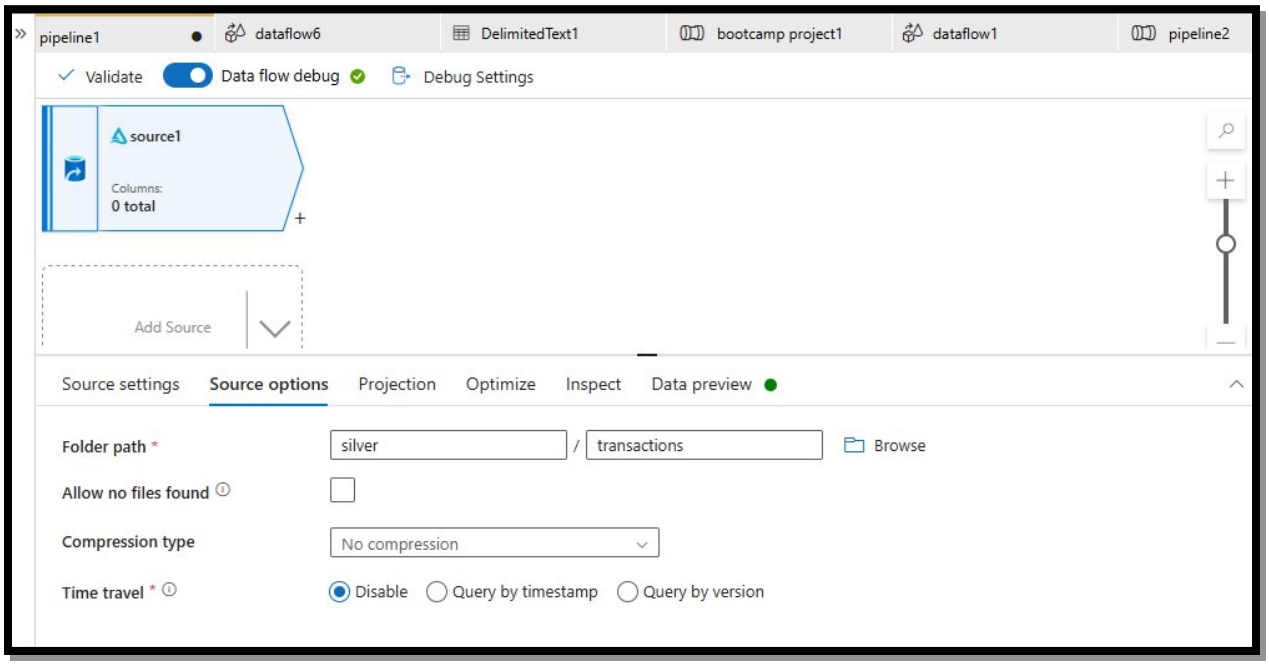


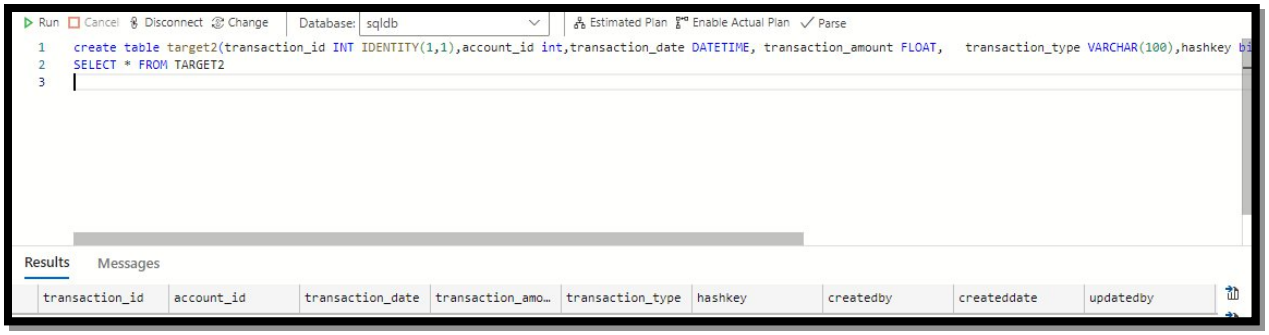
COLUMNAR CHART OF TABLE TARGET1

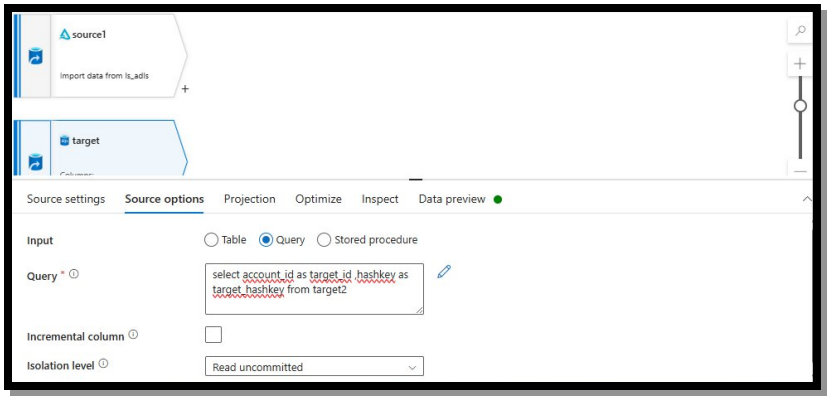


PIE CHART VISUAL REPRESENTATION OF TARGET1.

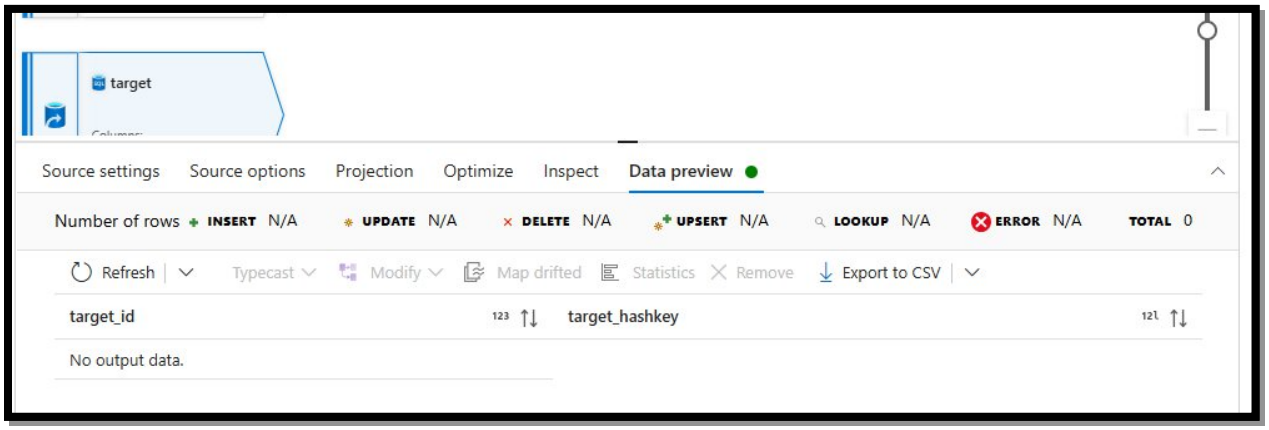
2. SCDTYPE1-> TRANSACTIONS.





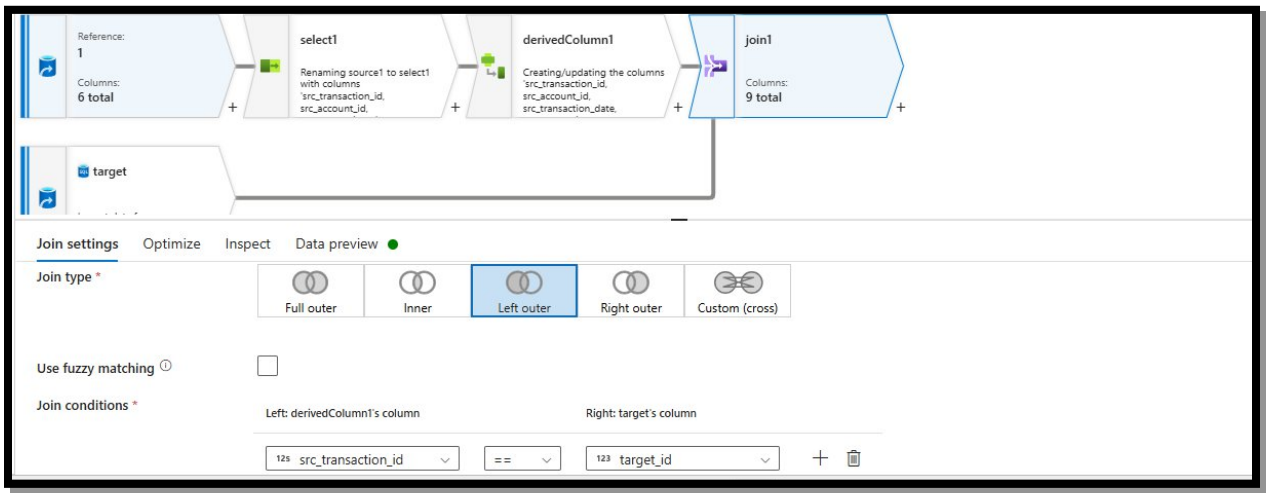


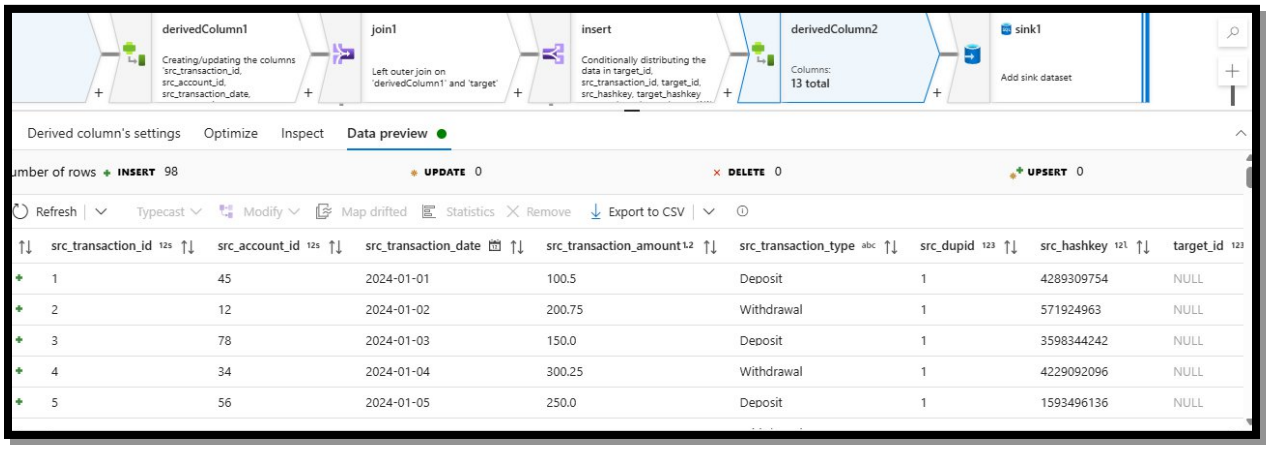
Target table



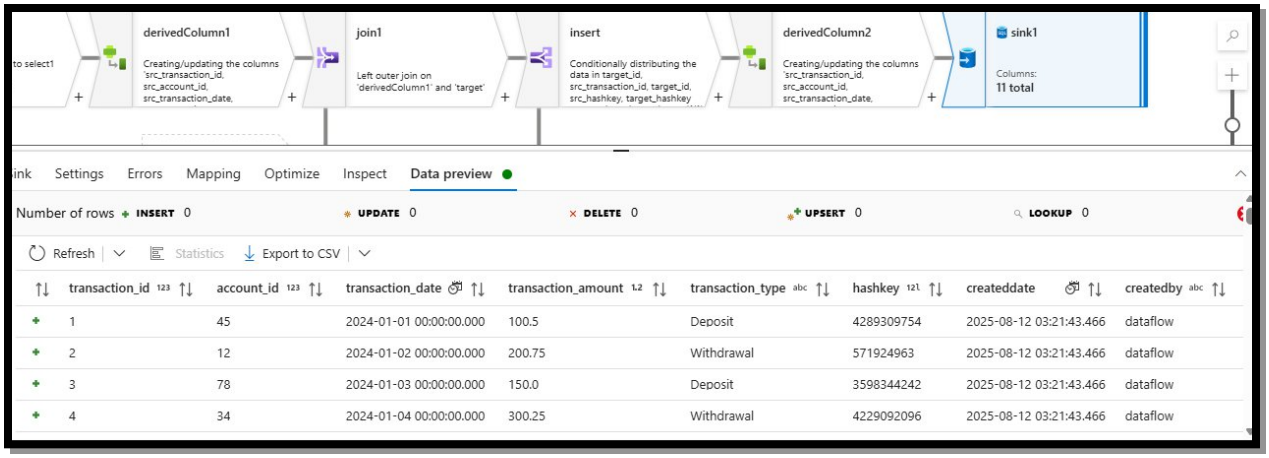
Generate hashkey using crc32 by selecting derived column->

Join

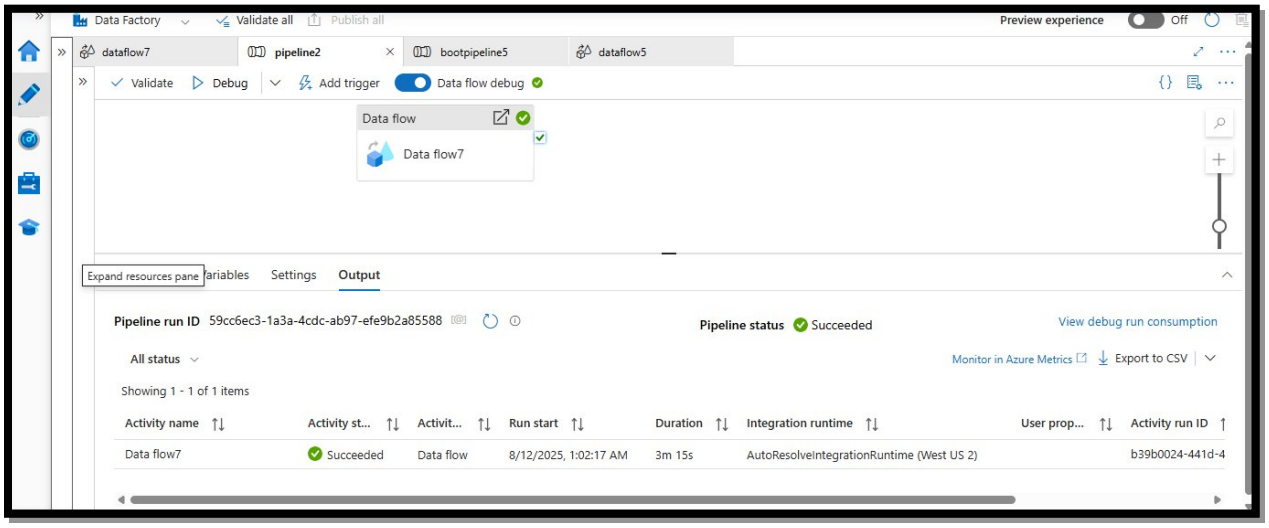


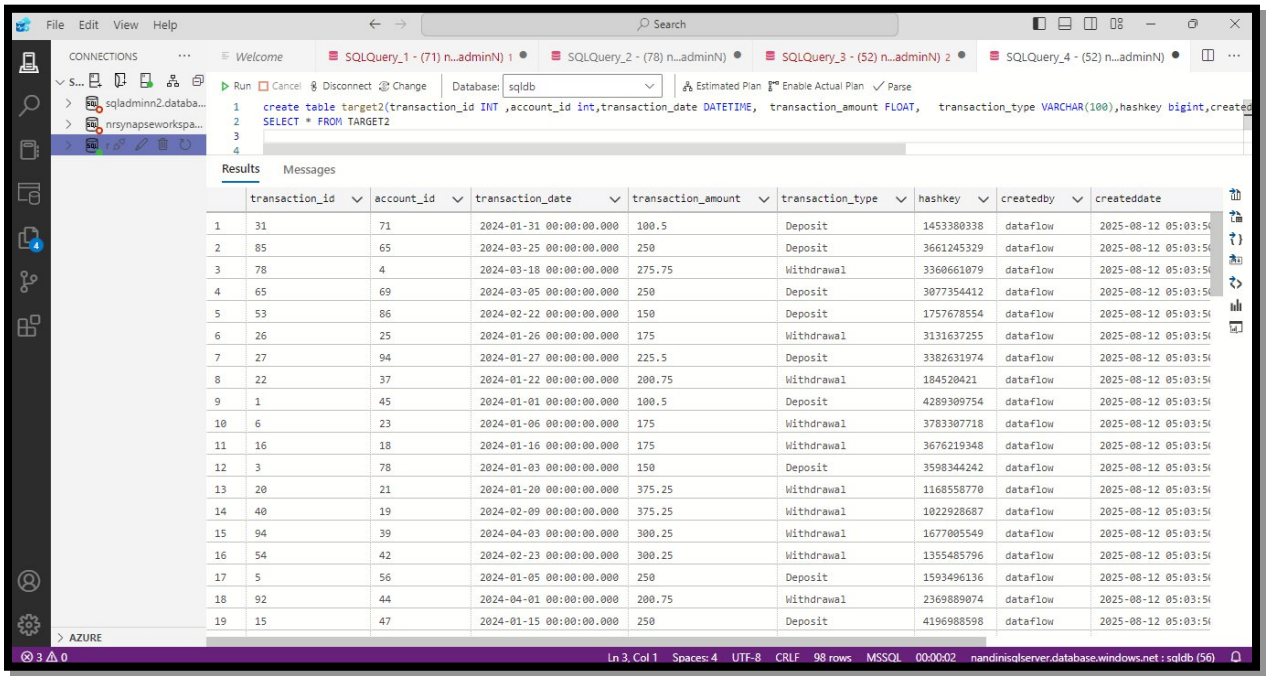


	src_transaction_id	src_account_id	src_transaction_date	src_transaction_amount	src_transaction_type	src_dupid	src_hashkey	target_id
+	1	45	2024-01-01	100.5	Deposit	1	4289309754	NULL
+	2	12	2024-01-02	200.75	Withdrawal	1	571924963	NULL
+	3	78	2024-01-03	150.0	Deposit	1	3598344242	NULL
+	4	34	2024-01-04	300.25	Withdrawal	1	4229092096	NULL
+	5	56	2024-01-05	250.0	Deposit	1	1593496136	NULL

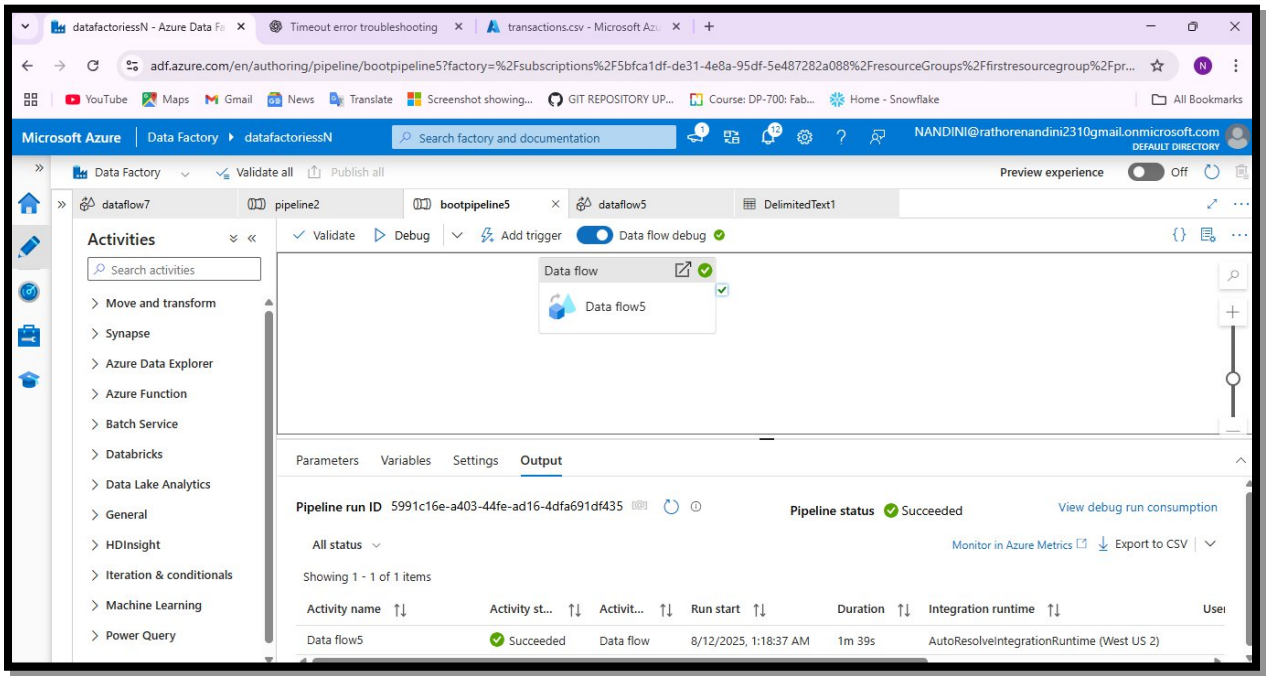


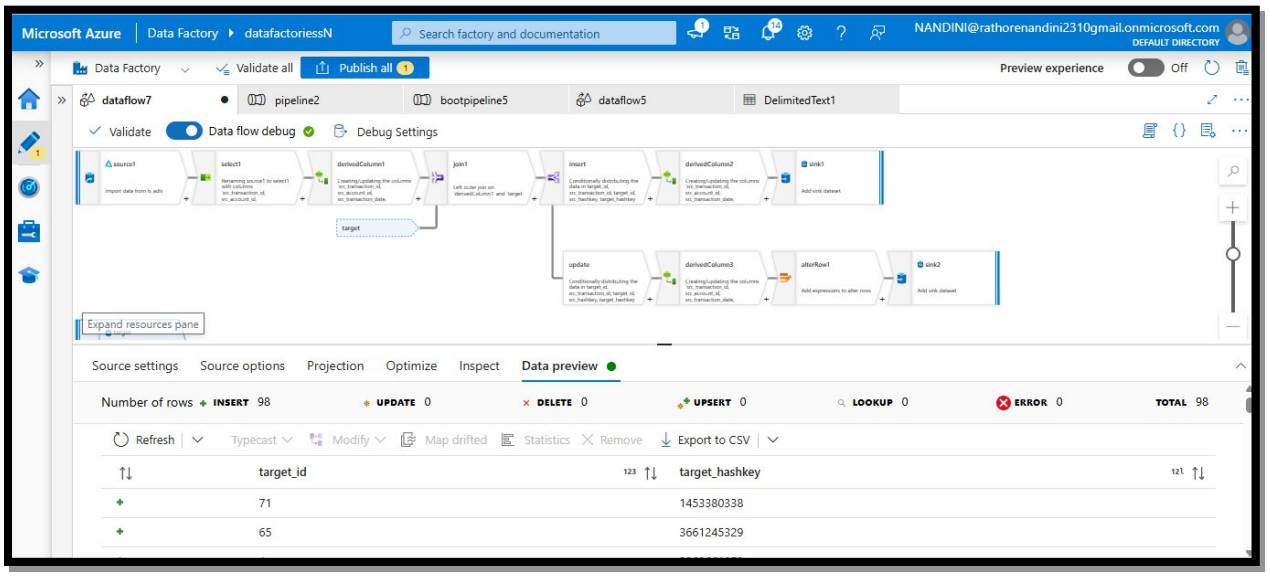
Sink data preview of inserted data.

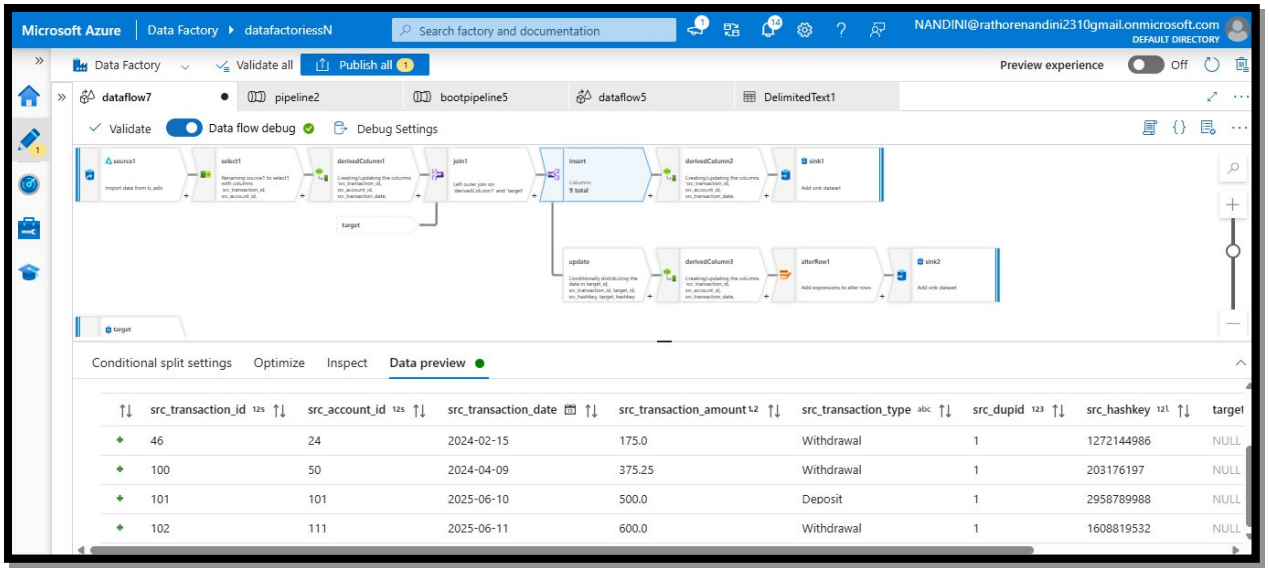


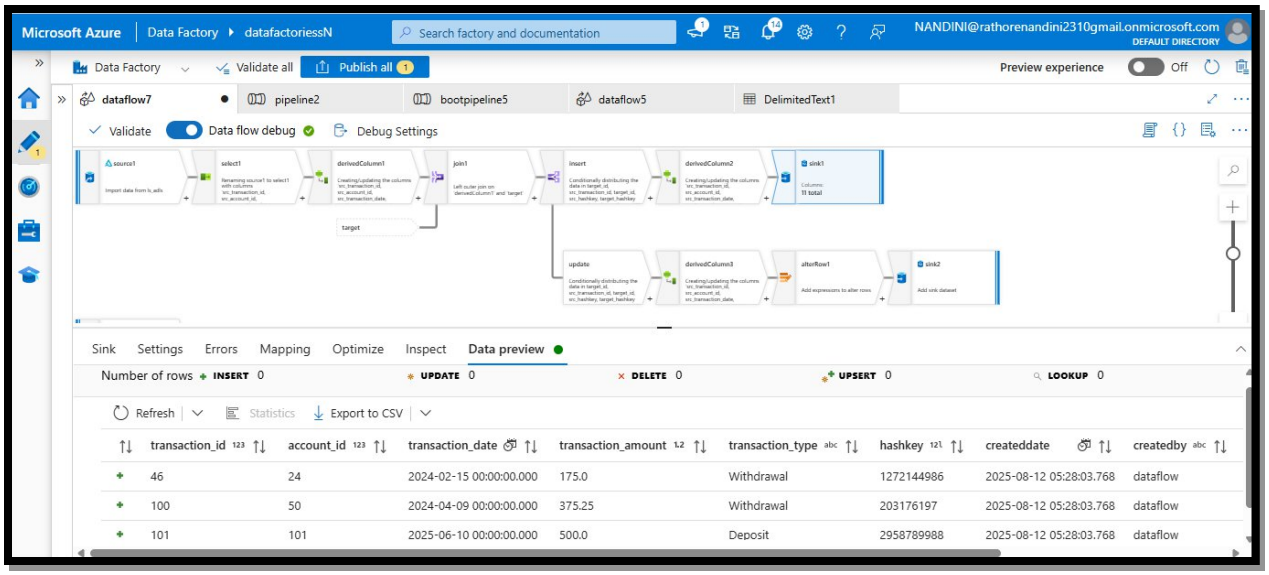


Now load the updated data as a source.

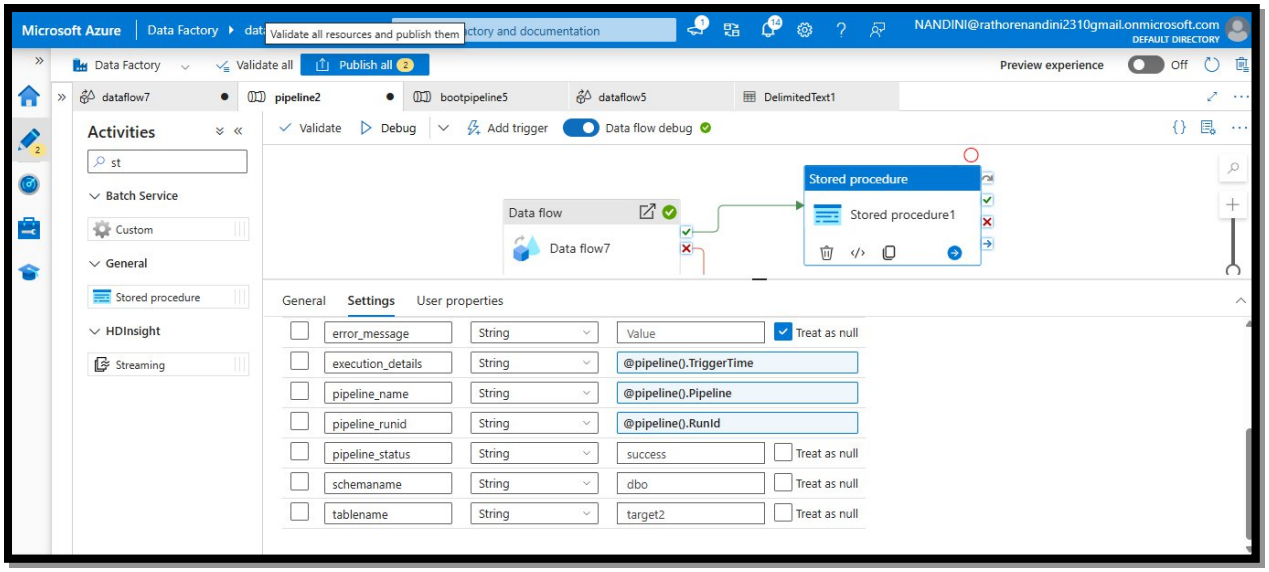


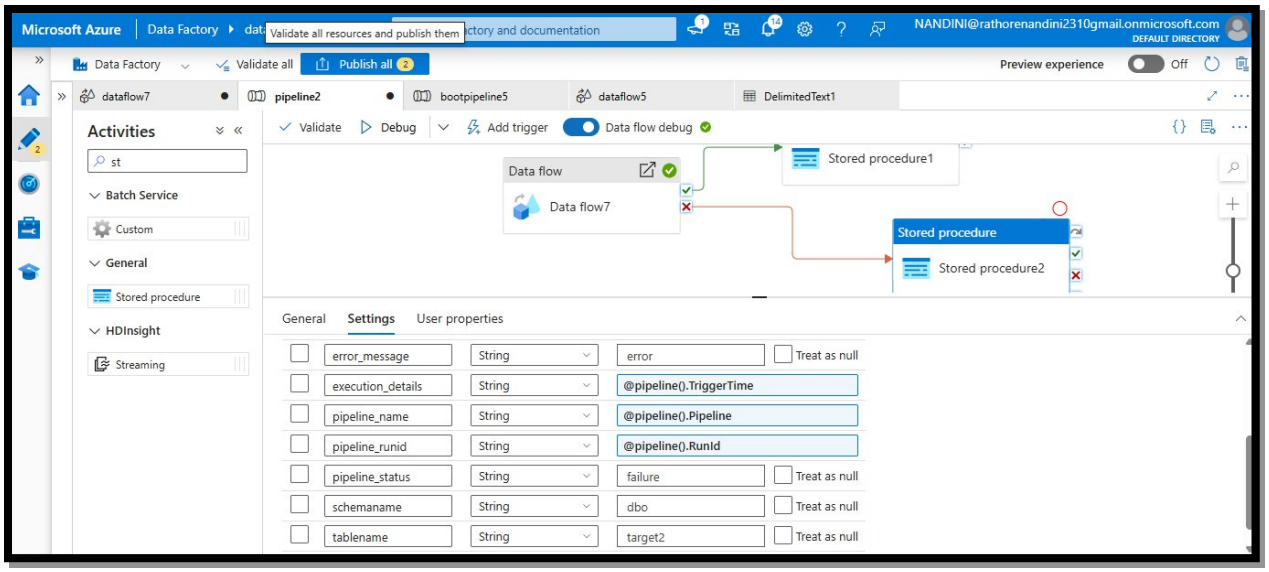


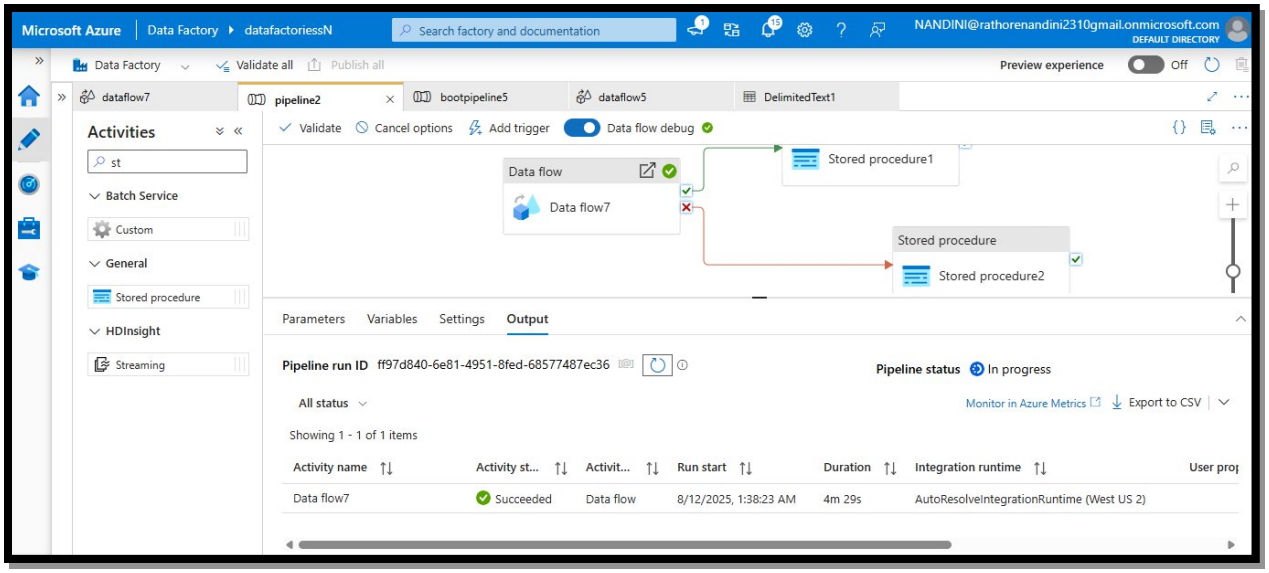


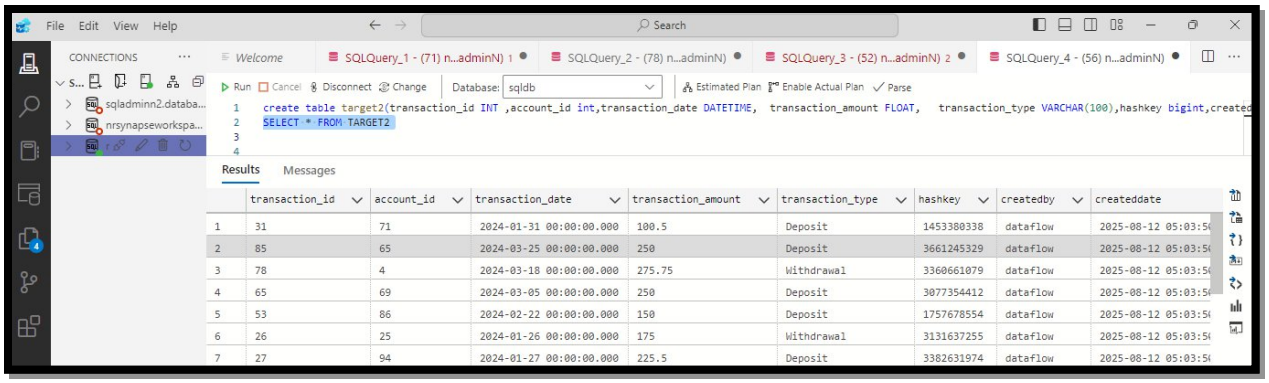


Sink of inserted columns .

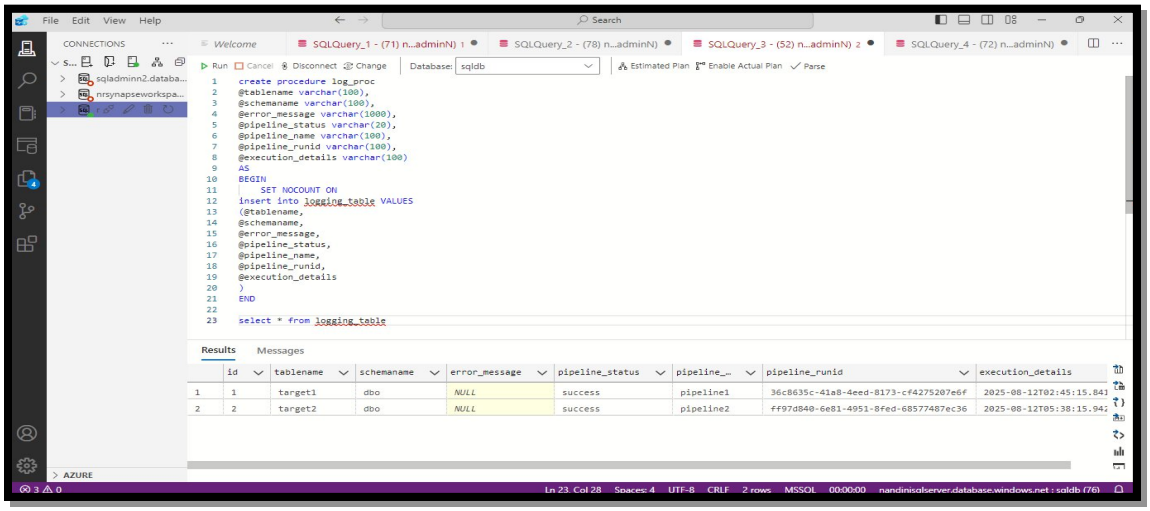

Microsoft Azure
Data Factory
Validate all resources and publish them
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all
Preview experience
Off
dataflow7
pipeline2
bootpipeline5
dataflow5
DelimitedText1
Activities
st
Batch Service
Custom
General
Stored procedure
HDInsight
Streaming
Validate
Debug
Add trigger
Data flow debug
Data flow
Data flow7
Stored procedure
Stored procedure1
General
Settings
User properties
error_message
String
Value
Treat as null
execution_details
@pipeline().TriggerTime
pipeline_name
@pipeline().Pipeline
pipeline_runid
@pipeline().RunId
pipeline_status
success
schemaname
dbo
tablename
target2


Microsoft Azure
Data Factory
Validate all resources and publish them
NANDINI@rathorenandini2310gmail.onmicrosoft.com
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all
Preview experience
Off
dataflow7
pipeline2
bootpipeline5
dataflow5
DelimitedText1
Activities
st
Batch Service
Custom
General
Stored procedure
HDInsight
Streaming
Validate
Debug
Add trigger
Data flow debug
Data flow
Data flow7
Stored procedure1
Stored procedure
Stored procedure2
General
Settings
User properties
error_message
String
error
Treat as null
execution_details
@pipeline().TriggerTime
pipeline_name
@pipeline().Pipeline
pipeline_runid
@pipeline().RunId
pipeline_status
failure
schemaname
dbo
tablename
target2



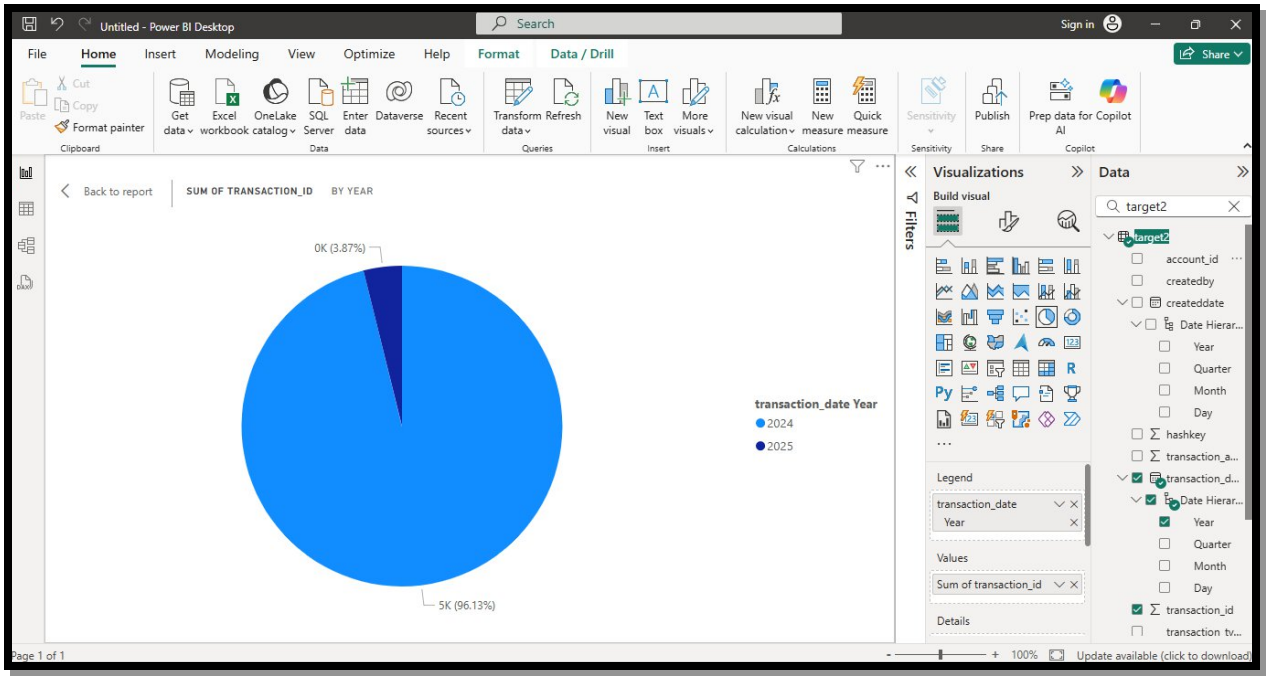


Target2 table in sql db after upsert in dataflow in azure data factory.

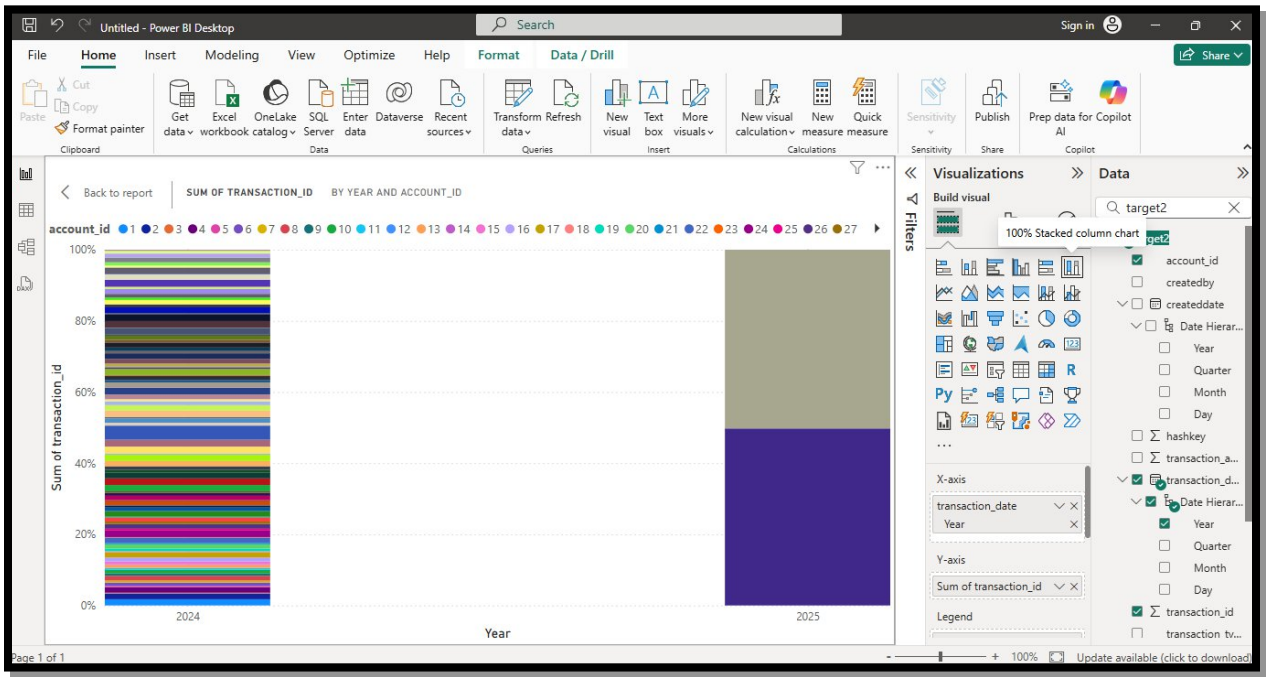


Logging table showing the pipeline status.

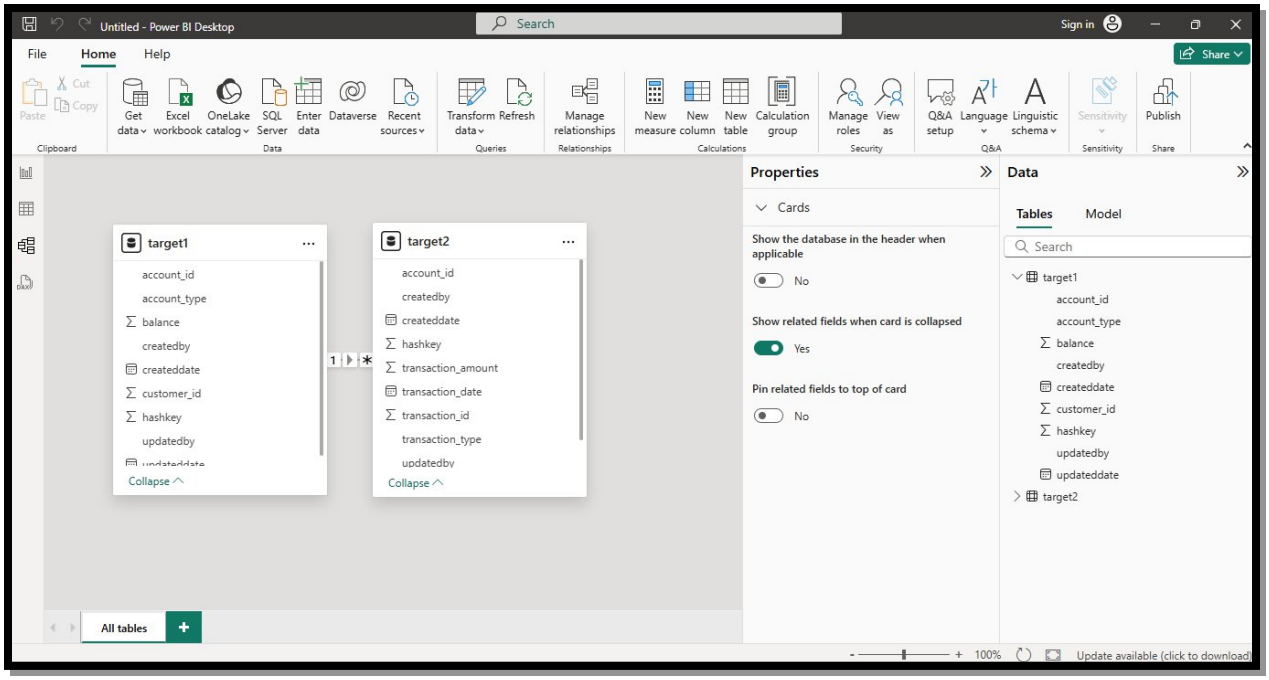
POWER BI REPORT:- PIE CHART



COLUMNAR CHART



MODEL VIEW TWO TABLES ARE CONNECTED ON THE BASIS OF ACCOUNT ID



Conclusion

The "Data Pipeline for Customer Account Analysis" project successfully demonstrates the design and implementation of a secure, efficient, and scalable data processing workflow using Azure Data Factory, Azure Data Lake Storage, and Azure SQL Database. By integrating Azure Key Vault for credential management, the solution ensures data security while adhering to best practices.

Through the staged architecture (Bronze → Silver → Gold), raw customer account data from the backend storage was ingested, cleansed, transformed, and loaded into the target SQL Database using SCD-1 upsert logic. This enabled both new data insertion and existing data updates without duplication.

The project not only streamlined the ETL process but also laid a foundation for advanced analytics by integrating the processed data with Power BI, enabling the generation of meaningful business insights. Overall, the pipeline meets its objectives of reliability, maintainability, and scalability, ensuring that downstream analytics and reporting can be performed on accurate and up-to-date data.

THANKYOU!