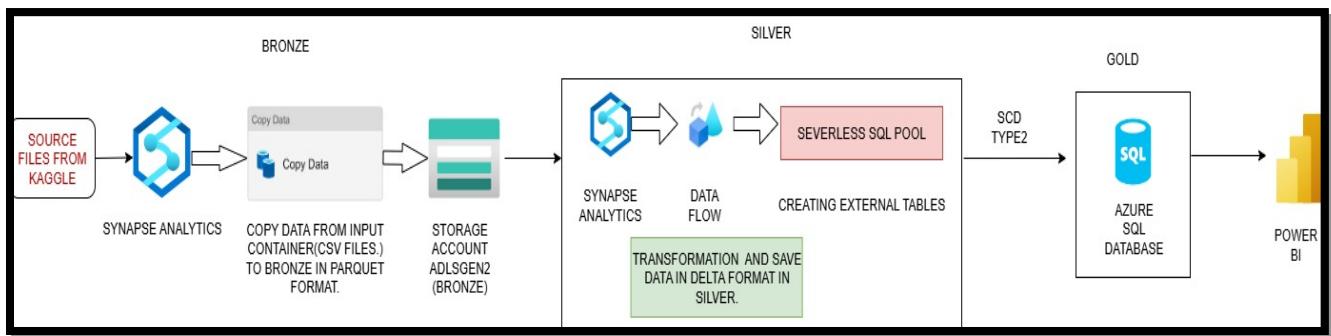


BOOTCAMP PROJECT-3

Customer 360 Data Integration

PROJECT ARCHITECTURE:-



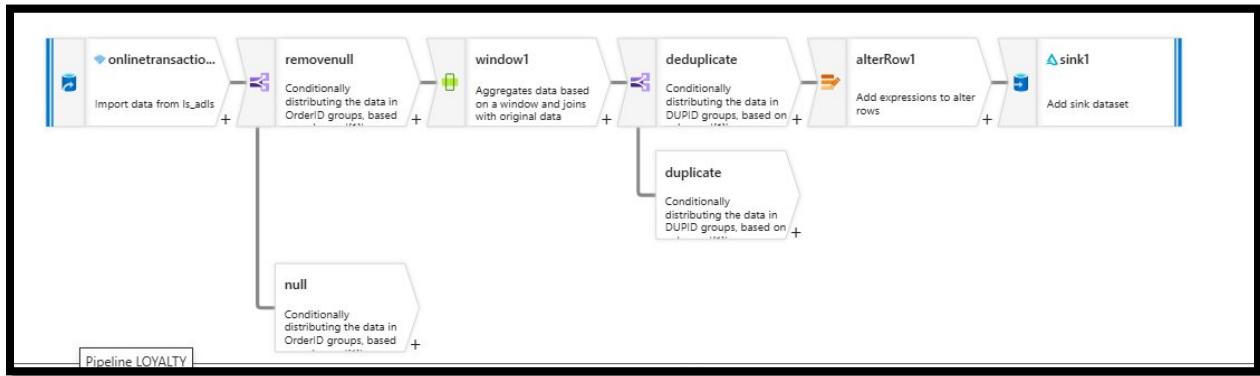
The screenshot shows the Azure Data Factory pipeline run details. On the left, there's a sidebar titled 'Activities' with a search bar and a list of activities: Synapse, Move and transform, Azure Data Explorer, Azure Function, Batch Service, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. The main area displays a pipeline run ID (0e52fe0d-3c1f-47b3-8e89-3a7872b974e8) and a Pipeline status (Succeeded). Below this, a table lists the activity details:

Activity name	Activity st...	Activit...	Run start	Duration	Integrat
Copy data1	✓ Succeeded	Copy data	8/24/2025, 6:01:48 PM	17s	AutoRes

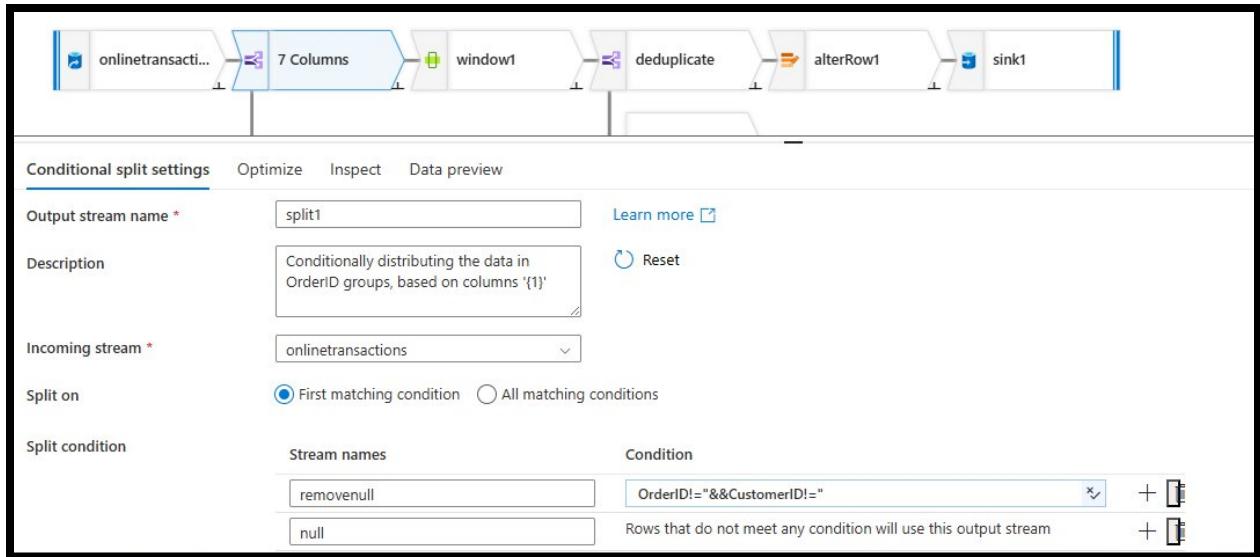
I first use copy activity to load raw csv files from input container to bronze layer in parquet format.

Then , in synapse using data flow for transformation of all the files and save them in silver layer in delta format.

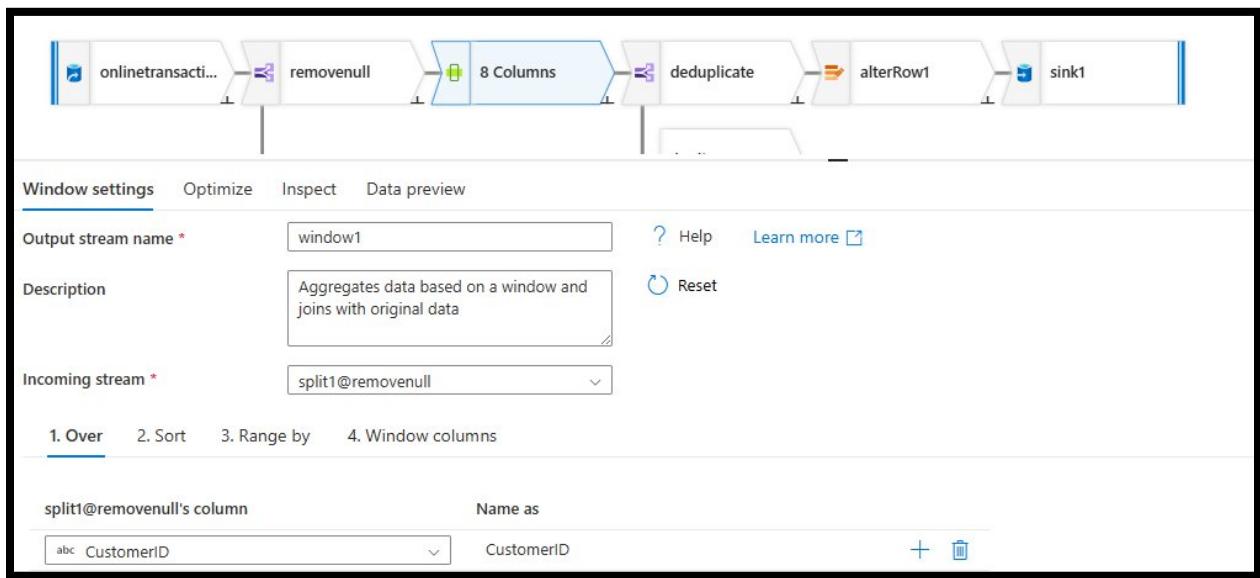
1.ONLINE TRANSACTION

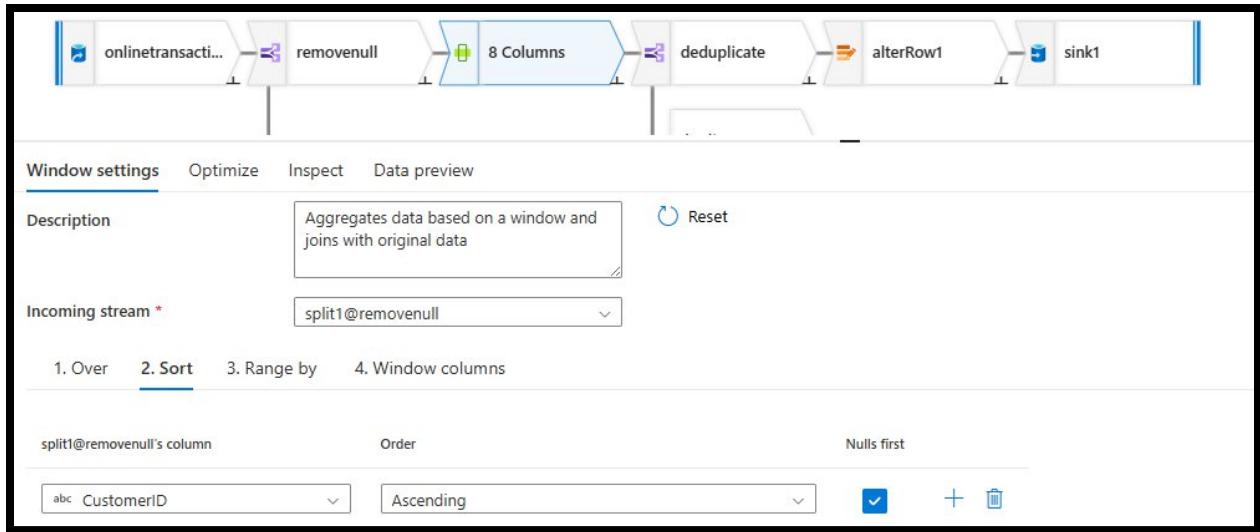


Removing null values

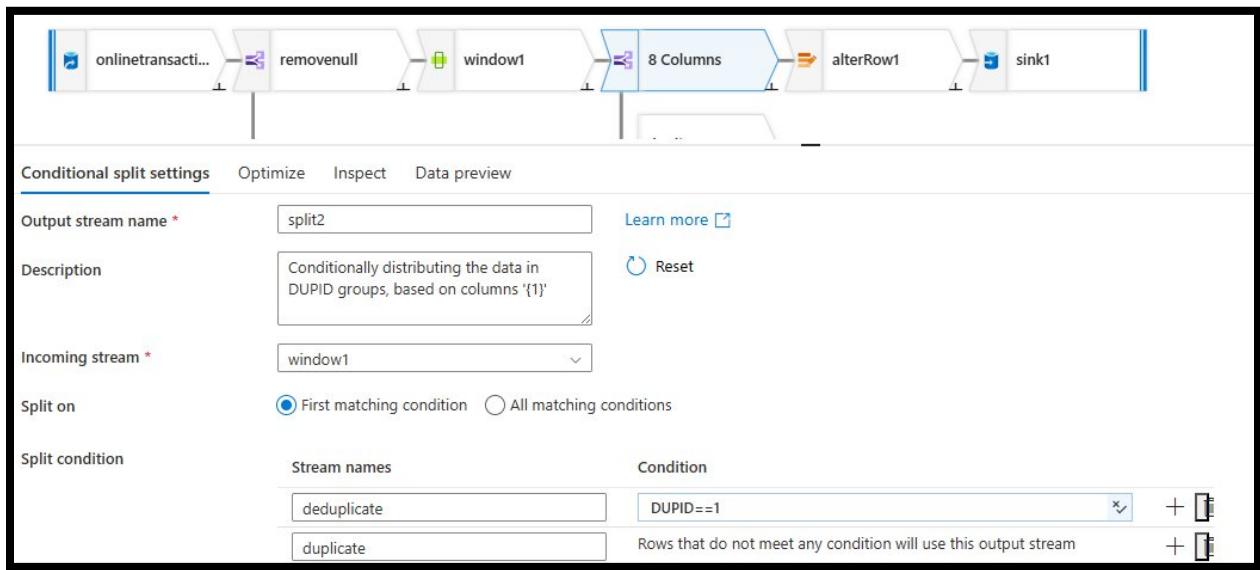


Window

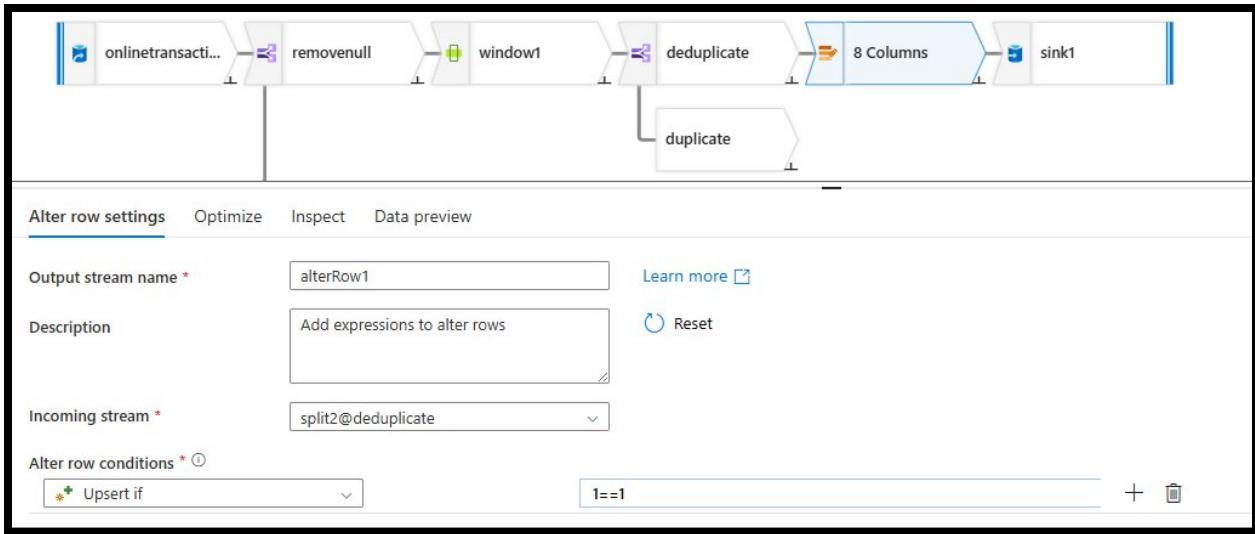




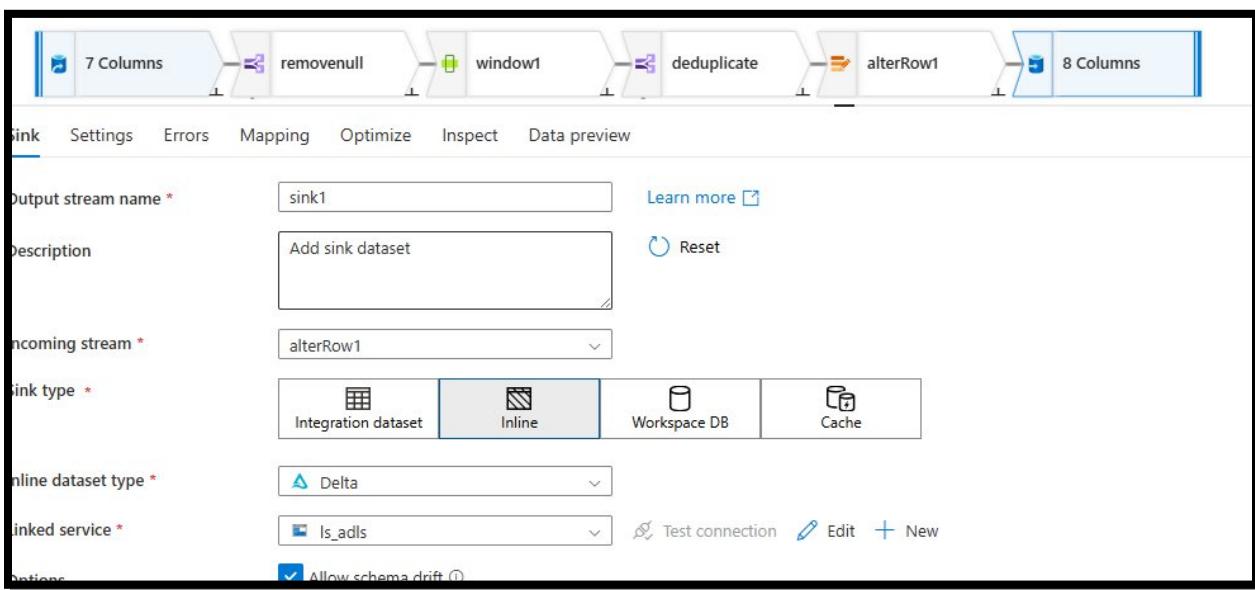
Removing duplicates

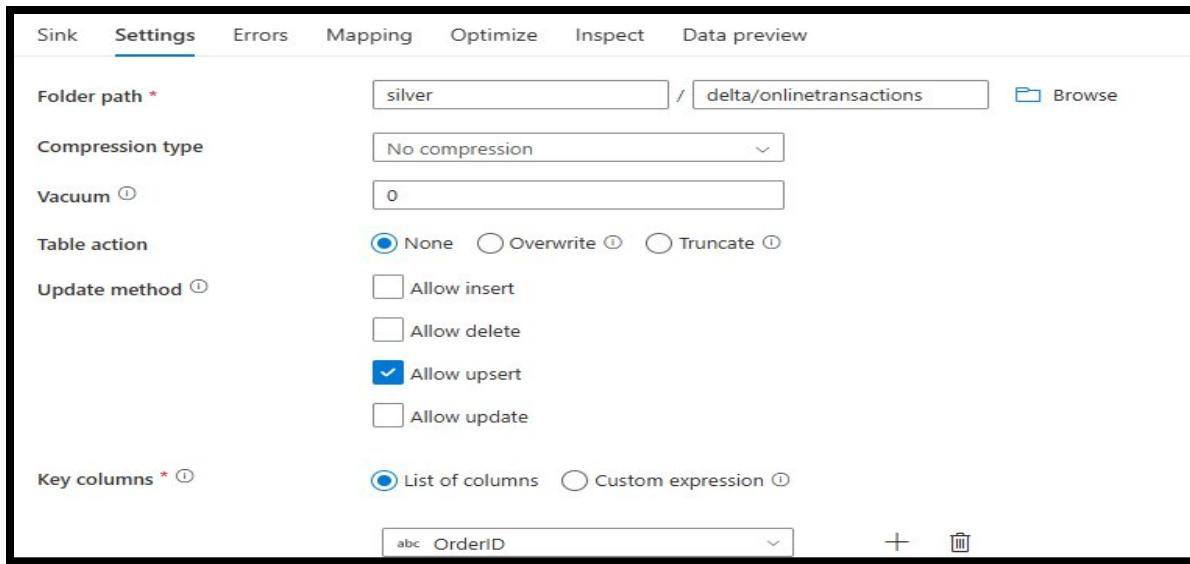


Alter row

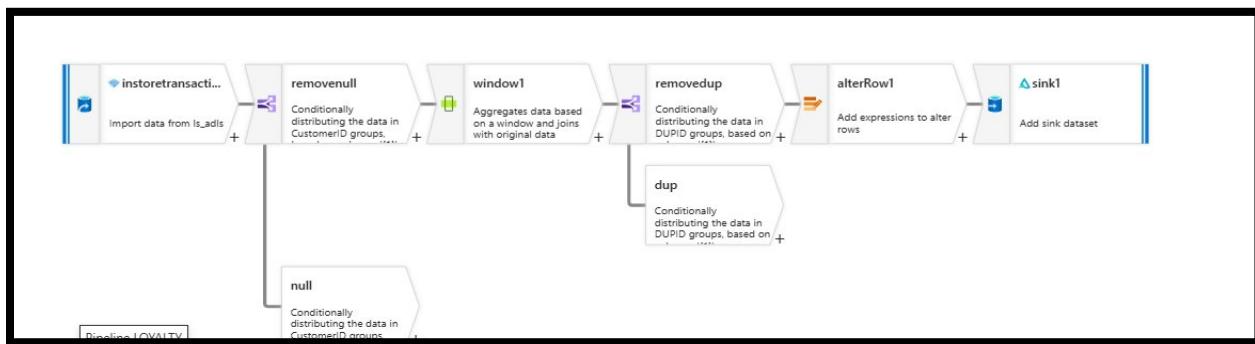


Sink





2. STORE PURCHASE



Removing null values

Conditional split settings Optimize Inspect Data preview

Description Conditionally distributing the data in CustomerID groups, based on columns '{1}' [Reset](#)

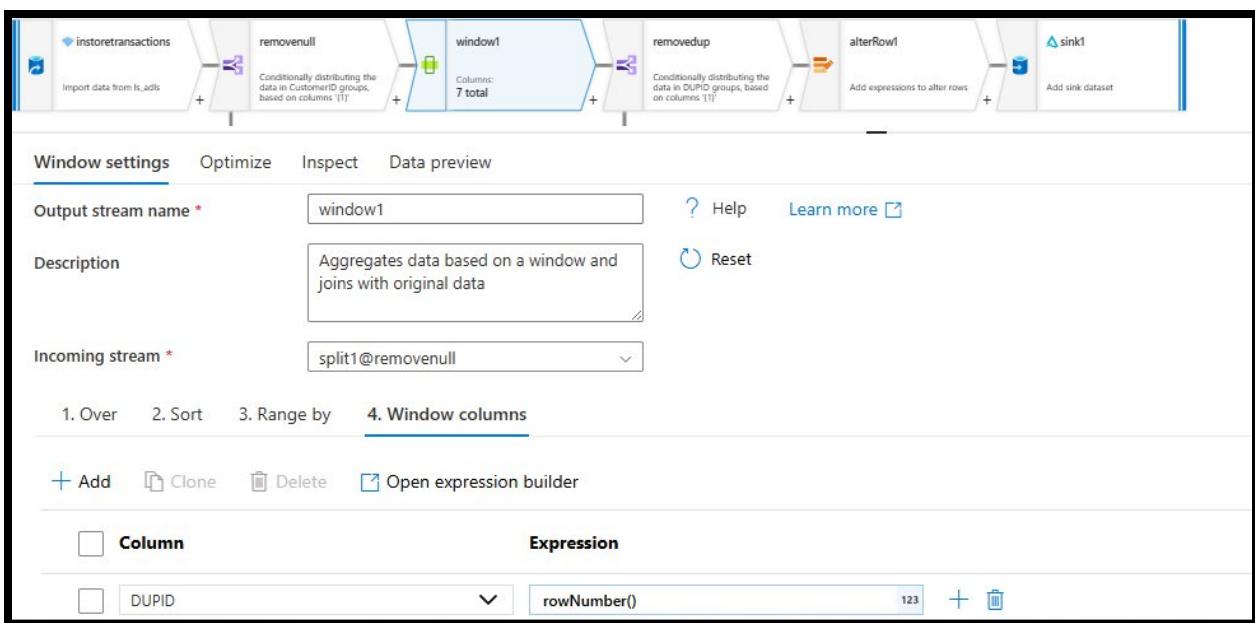
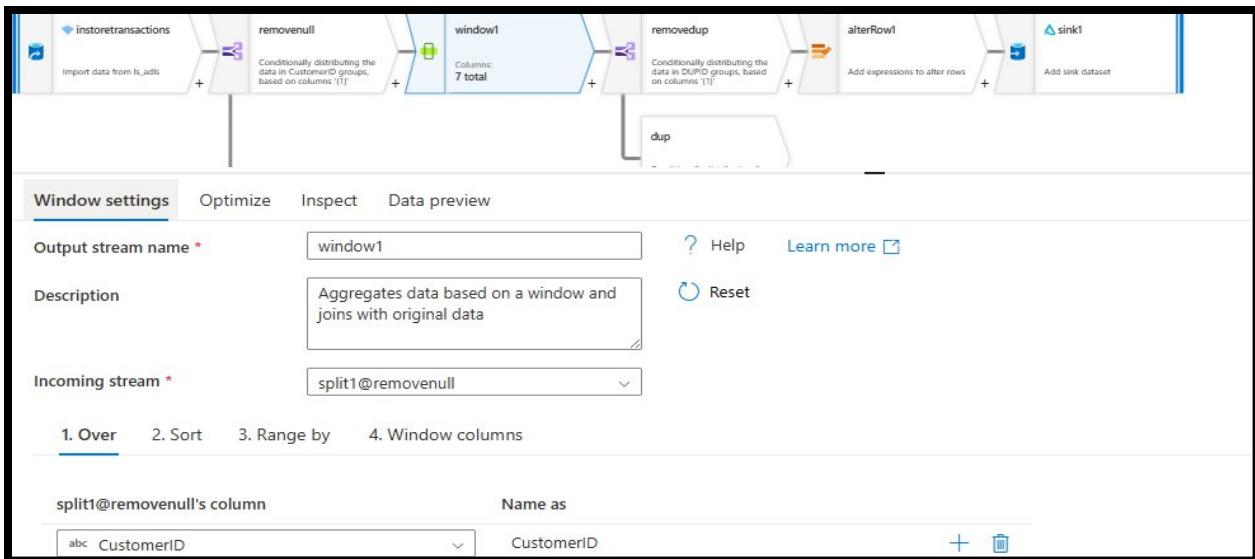
Incoming stream * instoretransactions

Split on First matching condition All matching conditions

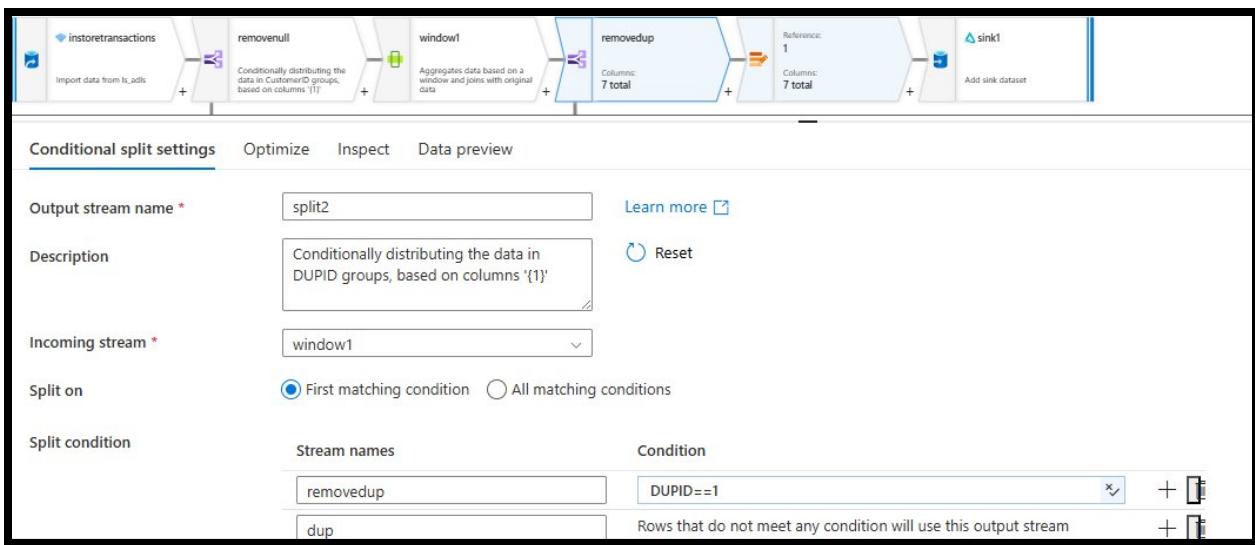
Split condition Stream names Condition

removennull	CustomerID=&&TransactionID=
null	Rows that do not meet any condition will use this output stream

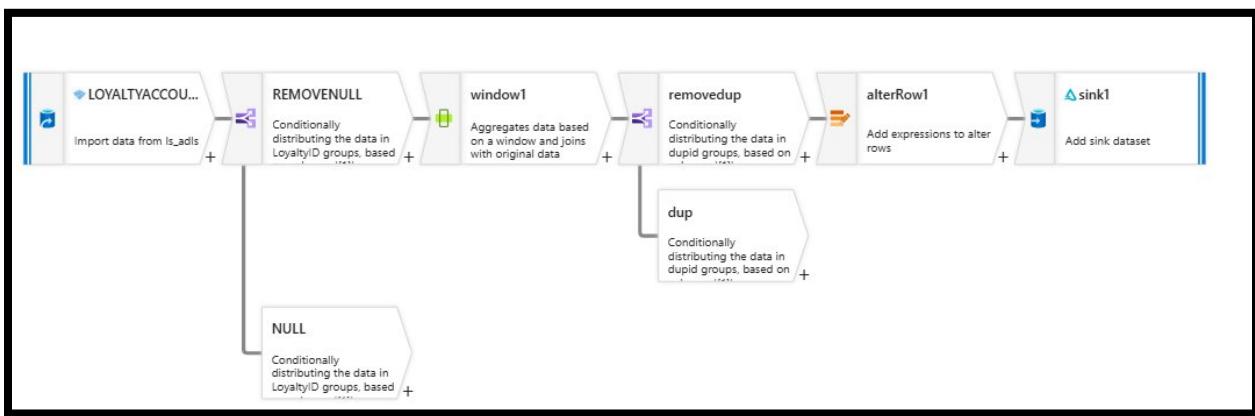
NANDINI RATHORE



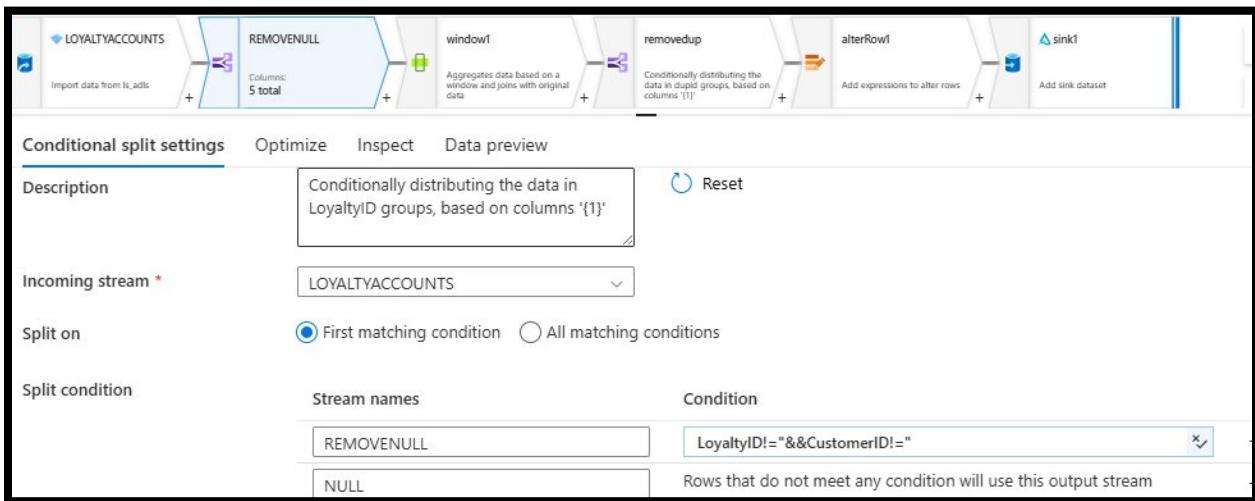
Remove duplicates



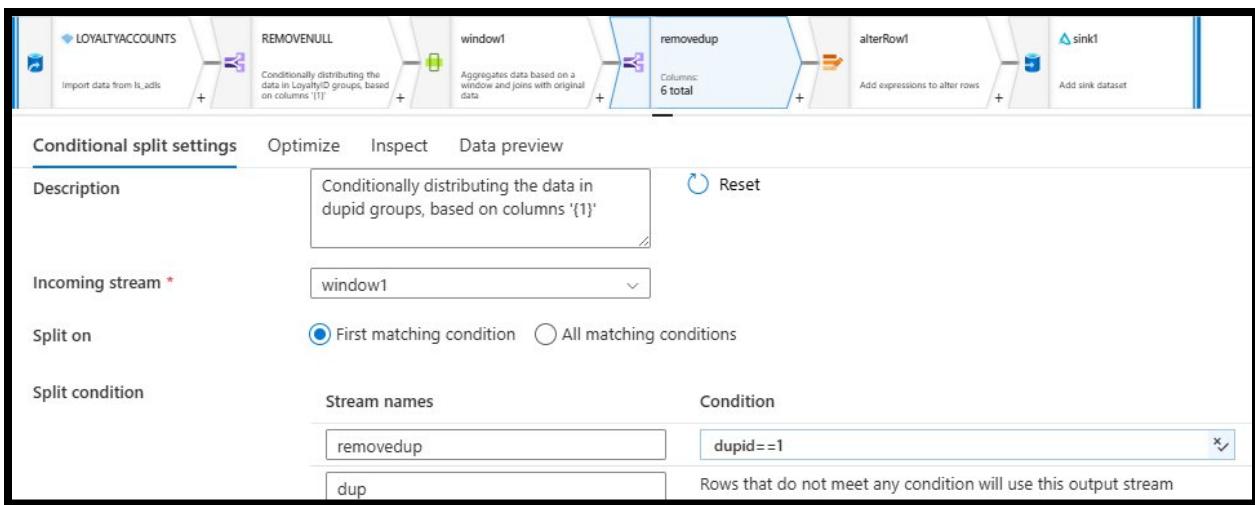
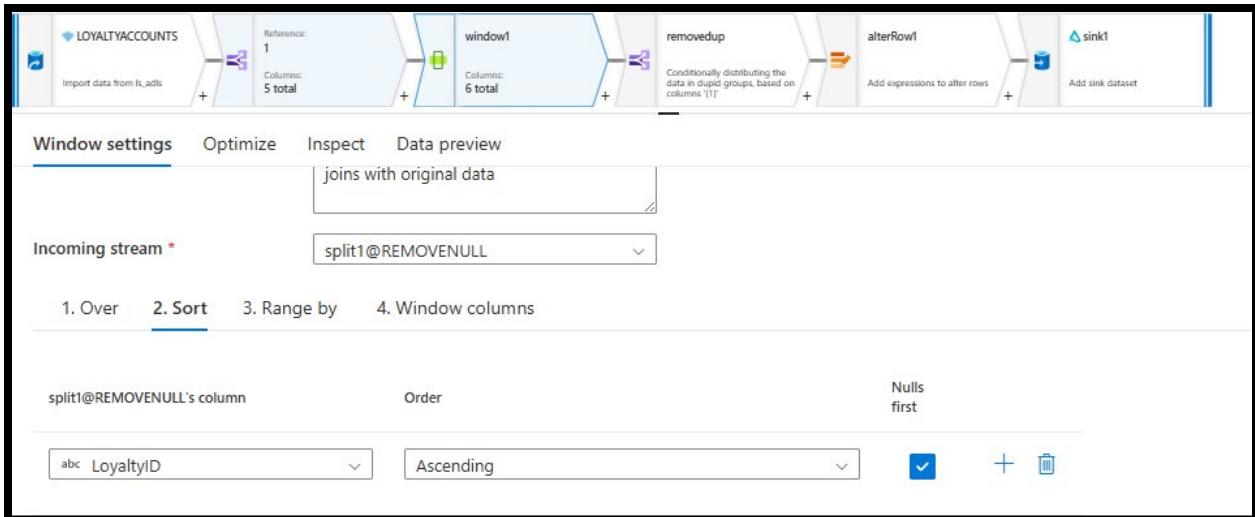
3. LOYALTY ACCOUNTS.



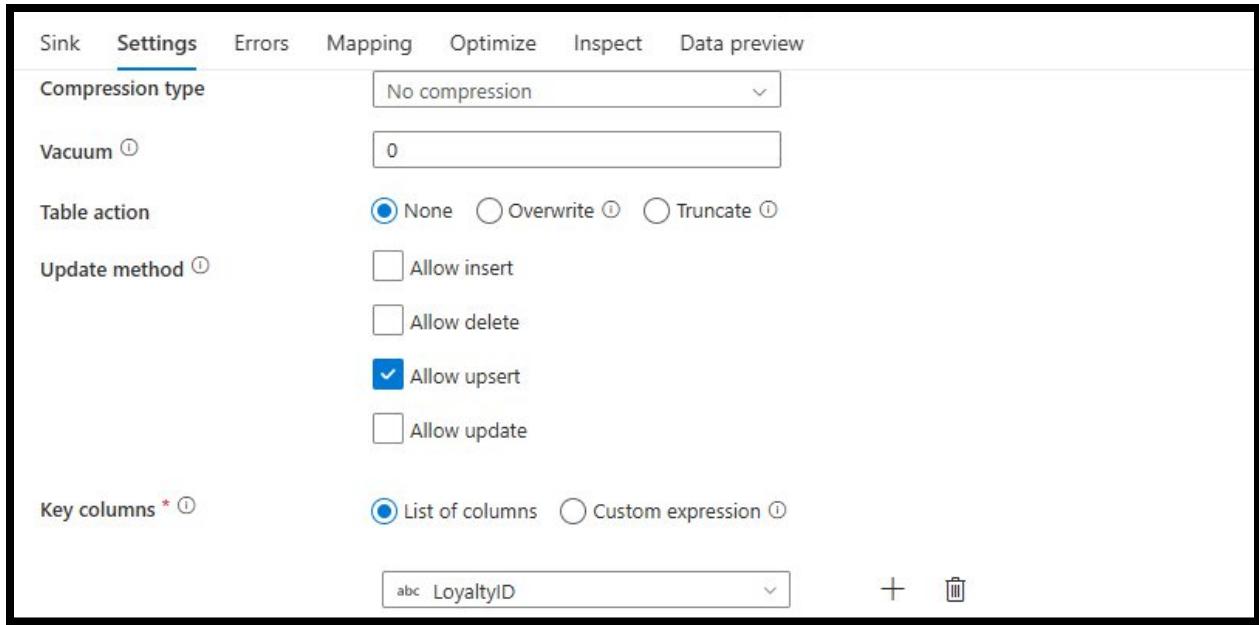
Removing null



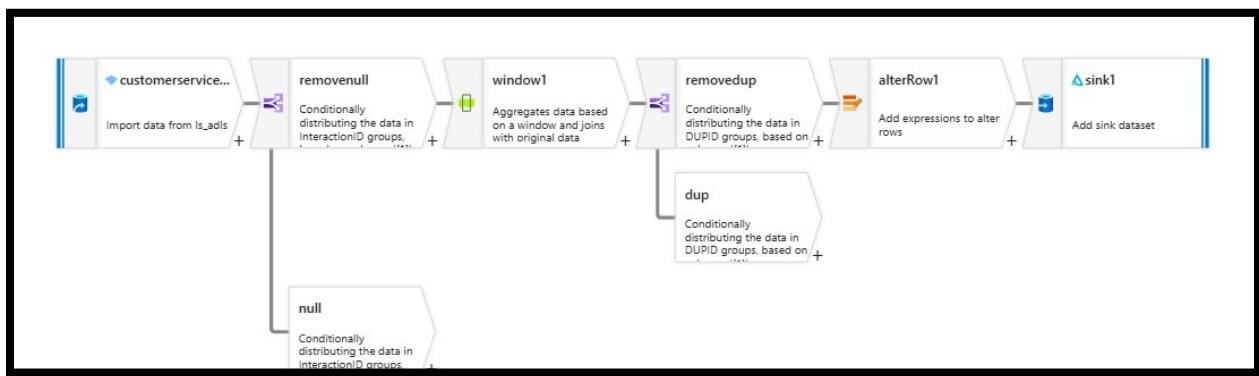
NANDINI RATHORE



Sink



4.CUSTOMER INTERACTION SERVICE.



The screenshot shows the Source settings page for a data pipeline. The pipeline consists of the following stages:

- Source: 6 Columns
- Processor: removeNull
- Processor: window1
- Processor: removedup
- Processor: alterRow1
- Sink: sink1

Source settings

Output stream name *: customerserviceinteractions [Learn more](#)

Description: Import data from ls_adls [Reset](#)

Source type *:

- Integration dataset
- Inline**
- Workspace DB

Inline dataset type *: Parquet

Linked service *: ls_adls [Test connection](#) [Edit](#) [New](#)

The screenshot shows the Conditional split settings page for a data pipeline. The pipeline consists of the following stages:

- Source: customerserviceinteractions
- Processor: 6 Columns
- Processor: window1
- Processor: removedup
- Processor: alterRow1
- Sink: sink1

Conditional split settings

Output stream name *: split1 [Learn more](#)

Description: Conditionally distributing the data in InteractionID groups, based on columns '{1}' [Reset](#)

Incoming stream *: customerserviceinteractions

Split on: First matching condition All matching conditions

Split condition

Stream names	Condition
removenull	InteractionID!=""&&CustomerID!="
null	Rows that do not meet any condition will use this output stream

The screenshot shows the Data Flow interface with the 'Window settings' tab selected. At the top, there is a flow diagram with components: 'customerservice...' → 'removennull' → '7 Columns' (with a green icon) → 'removedup' → 'alterRow1' → 'sink1'. Below the diagram, the 'Window settings' tab is active, followed by 'Optimize', 'Inspect', and 'Data preview'.
Description: Aggregates data based on a window and joins with original data.
Incoming stream *: split1@removennull
1. Over (selected), **2. Sort**, **3. Range by**, **4. Window columns**

A table for defining window columns:

split1@removennull's column	Name as
abc CustomerID	CustomerID

Buttons: +, -

The screenshot shows the same Data Flow interface with the 'Window settings' tab selected. The '2. Sort' section is highlighted. The flow diagram at the top is identical to the first screenshot.

Description: Aggregates data based on a window and joins with original data.
Incoming stream *: split1@removennull
1. Over, **2. Sort** (selected), **3. Range by**, **4. Window columns**

A table for defining sort columns:

split1@removennull's column	Order	Nulls first
abc CustomerID	Ascending	<input checked="" type="checkbox"/>

Buttons: +, -

Window settings Optimize Inspect Data preview

Description: Aggregates data based on a window and joins with original data [Reset](#)

Incoming stream *: split1@removennull

4. Window columns

+ Add [Clone](#) [Delete](#) [Open expression builder](#)

<input type="checkbox"/> Column	Expression
<input type="checkbox"/> DUPID	rowNumber()

Conditional split settings Optimize Inspect Data preview

Output stream name *: split2 [Learn more](#)

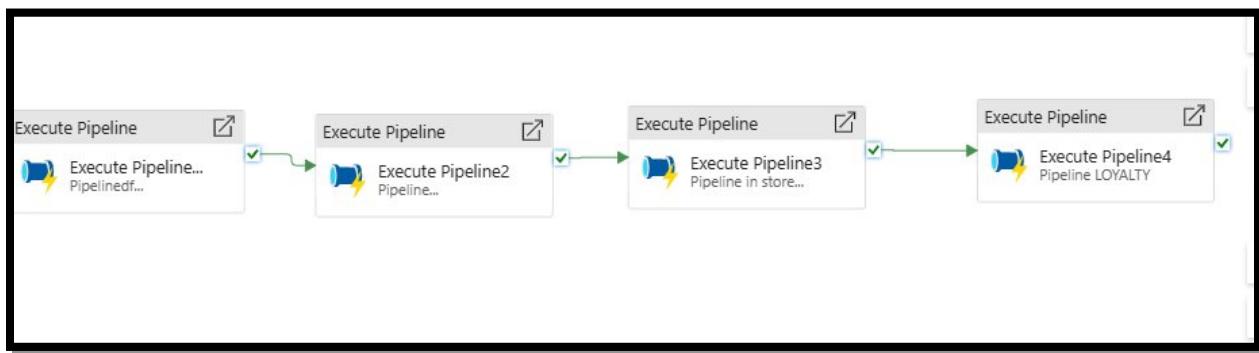
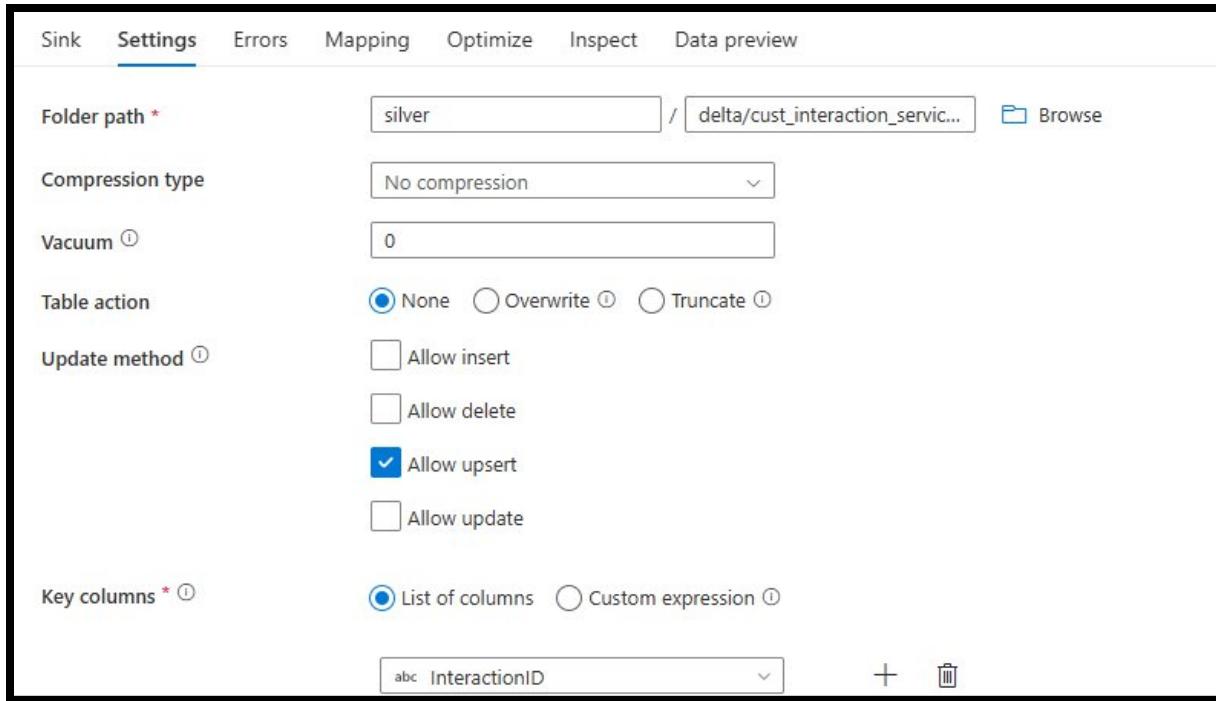
Description: Conditionally distributing the data in DUPID groups, based on columns '{1}' [Reset](#)

Incoming stream *: window1

Split on: First matching condition All matching conditions

Split condition

Stream names	Condition
removedup	DUPID==1
dup	Rows that do not meet any condition will use this output stream



EXEC PIPELINE TO RUN DATAFLOW PARALLELLY.

THEN PERFORMING SCD TYPE 2:-

1.CUSTOMER INTERACTION SERVICE.

NANDINI RATHORE

The screenshot shows the Source settings page for an integration dataset named "source1". The pipeline consists of the following steps: "7 Columns" -> "select1" -> "derivedColumn1" -> "join1" -> "insert" -> "union1" -> "derivedColumn2" -> "select3" -> "sink1". The target is labeled "target". The "Source options" tab is selected. The "Output stream name" is set to "source1". The "Description" field contains "Import data from ls_adls". The "Source type" is set to "Integration dataset". The "Inline dataset type" is set to "Delta". The "Linked service" is set to "ls_adls".

The screenshot shows the Source options page for an integration dataset. The "Source settings" tab is selected. The "Folder path" is set to "silver / delta/cust_interaction_service". The "Allow no files found" checkbox is unchecked. The "Compression type" is set to "No compression". The "Time travel" section has "Disable" selected.

The screenshot shows the Source options page for a query input. The "Source settings" tab is selected. The "Input" section shows "Query" selected. The "Query" field contains the SQL query: "select InteractionID AS tgt_Interaction_ID, hashkey as tgt_hashkey from ncpl.target".

NANDINI RATHORE

Select settings Optimize Inspect Data preview ●

Incoming stream * source1

Options

- Skip duplicate input columns ⓘ
- Skip duplicate output columns ⓘ

Input columns *

Auto mapping ⓘ Reset Add mapping Delete 1 mappings:

source1's column	Name as
1=1	concat('src_',\$\$)

Derived column's settings Optimize Inspect Data preview ●

Output stream name * derivedColumn1 [Learn more](#)

Description Creating/updating the columns 'src_InteractionID, src_CustomerID, src_DateTime, src_AgentID'

Incoming stream * select1

Add Clone Delete Open expression builder

Columns * ①

Column	Expression
src_hashkey	crc32(concat(toString(src_InteractionID,src_Custo... 121))

Join settings Optimize Inspect Data preview ●

Right stream * target

Join type *

- Full outer
- Inner
- Left outer
- Right outer
- Custom (cross)

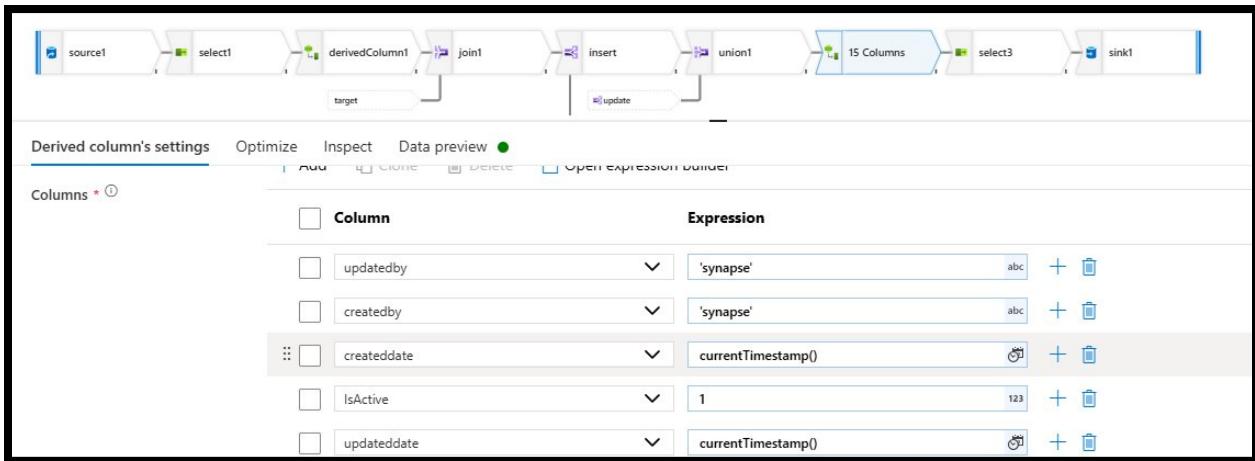
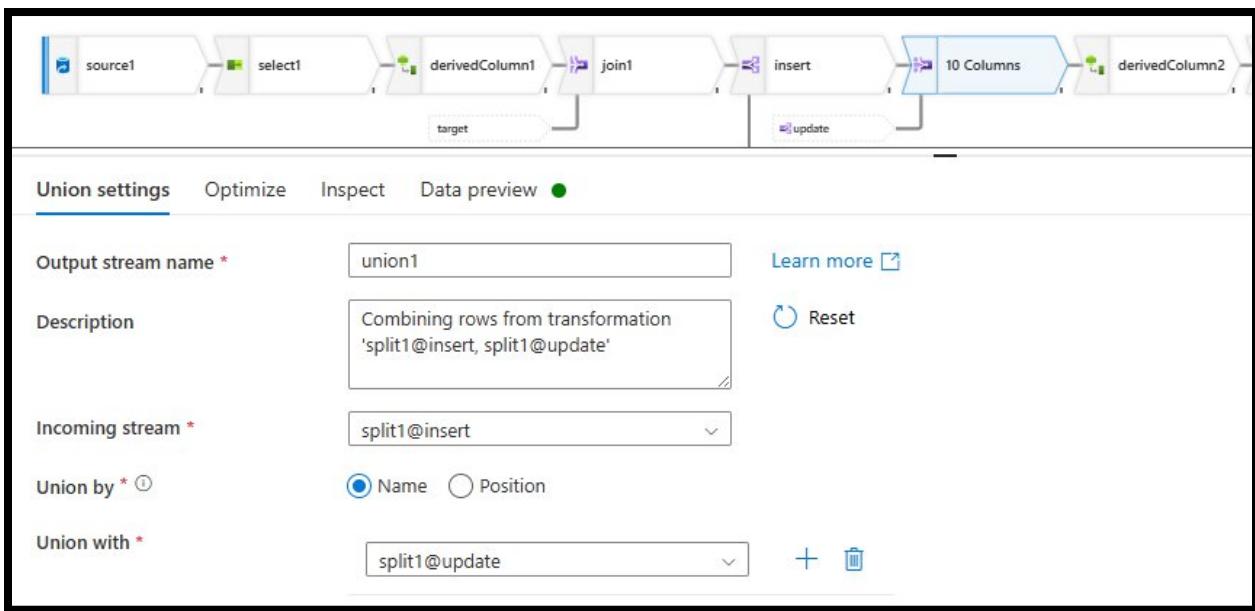
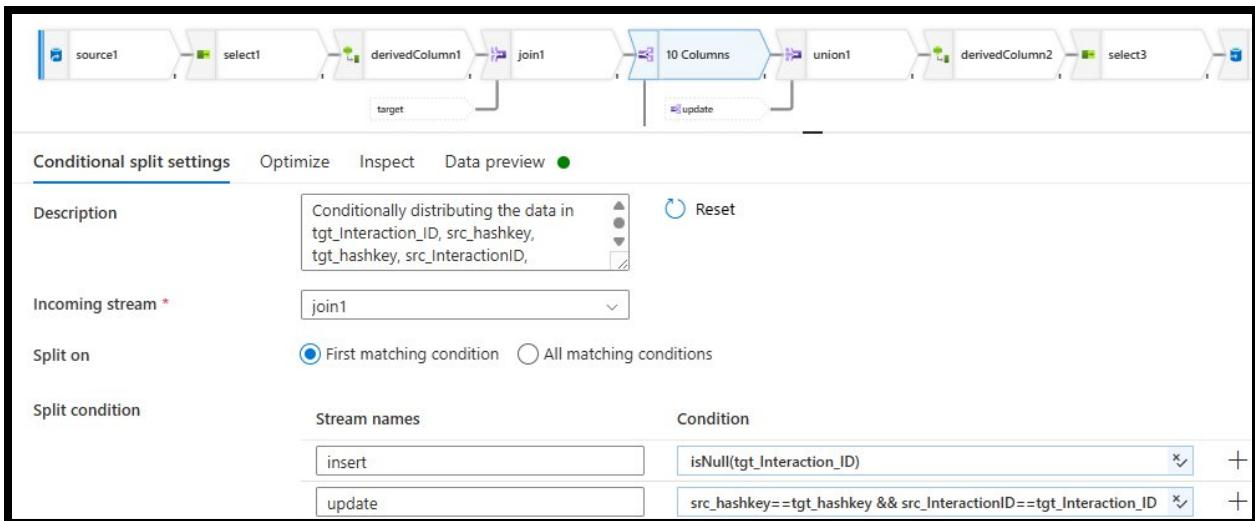
Use fuzzy matching ⓘ

Join conditions *

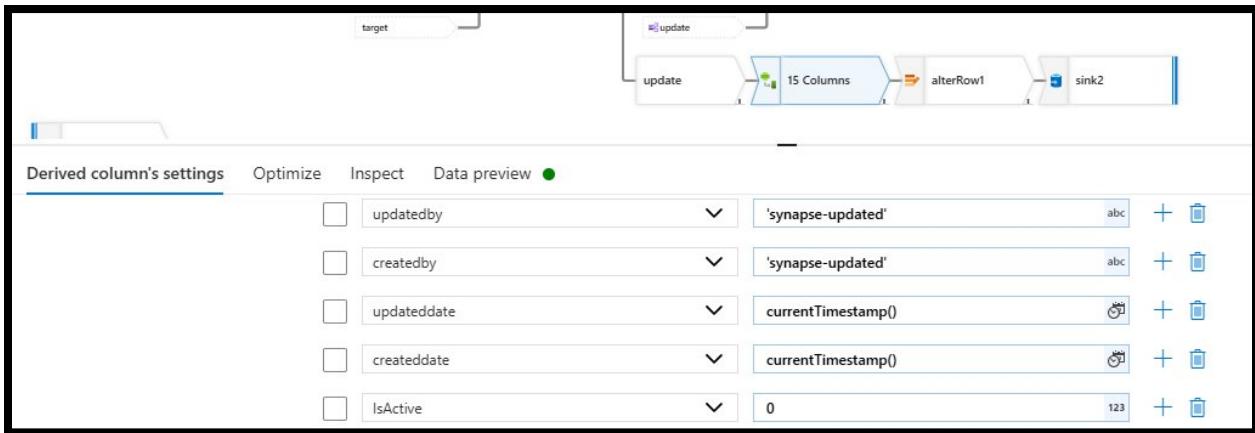
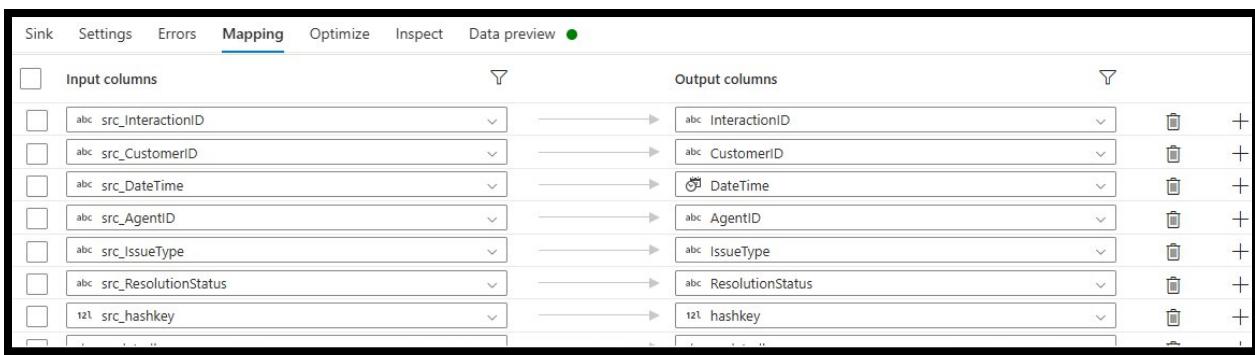
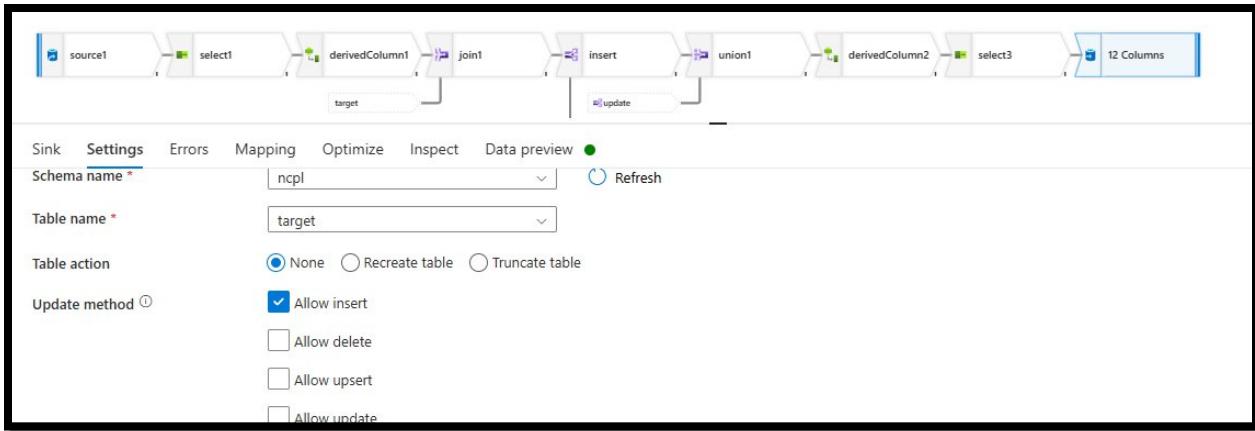
Left: derivedColumn1's column Right: target's column

abc src_InteractionID	=	abc tgt_Interaction_ID

NANDINI RATHORE



NANDINI RATHORE



The screenshot shows the Alteryx interface with the 'Alter row settings' tab selected. At the top, there is a flow diagram with components: 'update', 'derivedColumn3', '15 Columns', and 'sink2'. Below the tabs, there are sections for 'Description' (containing 'Add expressions to alter rows') and 'Incoming stream *' (set to 'derivedColumn3'). Under 'Alter row conditions *', there is a dropdown set to '* Update if' and a condition '1=1'. A 'Reset' button is also present.

The screenshot shows the Alteryx interface with the 'Settings' tab selected for a sink component. The tabs at the top are 'Sink', 'Settings', 'Errors', 'Mapping', 'Optimize', 'Inspect', and 'Data preview'. The 'Settings' tab has the following configuration:

- Schema name ***: ncpl
- Table name ***: target
- Table action**: None Recreate table Truncate table
- Update method**:
 - Allow insert
 - Allow delete
 - Allow upsert
 - Allow update
- Skip writing key columns**:
- Key columns***:
 - List of columns
 - Custom expression

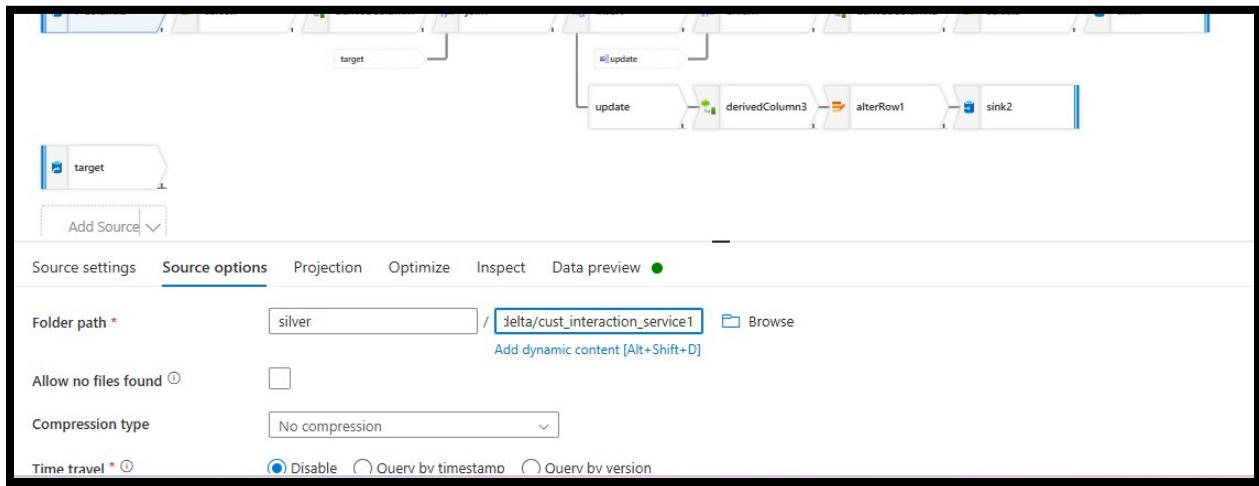
NANDINI RATHORE

```
16  select * from ncpi.target
17  |
```

Results Messages

AgentID	IssueType	ResolutionStatus	hashkey	updatedby	createdby	createddate	IsActive	updateddate
34	Billing	Resolved	4128546745	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
4	Complaint	Escalated	2790703594	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
73	Complaint	Resolved	940955941	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
88	Product Inquiry	Escalated	428630457	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
87	Technical Issue	Resolved	213253941	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
7	Other	Resolved	2036608558	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
96	Complaint	Resolved	469473356	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
72	Technical Issue	Resolved	469859290	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
66	Other	Escalated	4117180405	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
95	Technical Issue	Escalated	2629270830	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
70	Other	Pending	3890452916	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
29	Complaint	Resolved	1387019592	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
57	Product Inquiry	Pending	2093561070	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787
48	Technical Issue	Pending	2507396159	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.787

UPDATE



NANDINI RATHORE

The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Cancel, Disconnect, Change, Database (set to sqldb), Estimated Plan, Enable Actual Plan, Parse, and a dropdown menu. The code area contains the following SQL:

```
1 | CREATE TABLE ncp1.target(
2 |   InteractionID  VARCHAR(50),
3 |   CustomerID     VARCHAR(50),
4 |   DateTime        DATETIME,
5 |   AgentID         VARCHAR(50),
6 |   IssueType       VARCHAR(100),
7 |   ResolutionStatus VARCHAR(50),
8 |   hashkey         BIGINT,
9 |   updatedby       VARCHAR(50),
10 |  createdby       VARCHAR(50),
11 |  createddate     DATETIME,
12 |  IsActive        INT,
13 |  updateddate     DATETIME
14 | );
15 |
16 | select * from ncp1.target
```

Below the code, there are tabs for Results and Messages. The Results tab displays a table with the following columns: AgentID, IssueType, ResolutionStatus, hashkey, updatedby, createdby, createddate, IsActive, and updateddate. The table contains 10 rows of data.

AgentID	IssueType	ResolutionStatus	hashkey	updatedby	createdby	createddate	IsActive	updateddate
85	Technical Issue	Resolved	2189844889	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
31	Technical Issue	Pending	3307132076	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
1	Product Inquiry	Pending	94624014	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
92	Other	Escalated	4237643732	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
9	Other	Escalated	3020465112	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
13	Complaint	Pending	4174671449	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
78	Product Inquiry	Resolved	4124195403	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
50	Technical Issue	Pending	3381931800	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
66	Other	Resolved	3879407369	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
78	Product Inquiry	Escalated	1763477688	synapse	synapse	2025-08-24 22:42:13.787	1	2025-08-24 22:42:13.
101	other	escalated	609859323	synapse-updated	synapse-upd...	2025-08-24 23:14:25.580	0	2025-08-24 23:14:25.

CREATING EXTERNAL TABLE.

NANDINI RATHORE

```
1 IF NOT EXISTS (SELECT * FROM sys.external_file_formats WHERE name = 'SynapseDeltaFormat')
2     CREATE EXTERNAL FILE FORMAT [SynapseDeltaFormat]
3     WITH ( FORMAT_TYPE = DELTA)
4 GO
5
6 IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name = 'silver_datalakestorage22_dfs_core_windows_net')
7     CREATE EXTERNAL DATA SOURCE [silver_datalakestorage22_dfs_core_windows_net]
8     WITH (
9         LOCATION = 'abfss://silver@datalakestorage22.dfs.core.windows.net'
10    )
11 GO
12 create schema ncpl
13 CREATE EXTERNAL TABLE ncpl.project3 (
14     [InteractionID] nvarchar(4000),
15     [CustomerID] nvarchar(4000),
16     [DateTime] nvarchar(4000),
17     [AgentID] nvarchar(4000),
18     [IssueType] nvarchar(4000),
```

```
13 CREATE EXTERNAL TABLE ncpl.project3 (
14     [InteractionID] nvarchar(4000),
15     [CustomerID] nvarchar(4000),
16     [DateTime] nvarchar(4000),
17     [AgentID] nvarchar(4000),
18     [IssueType] nvarchar(4000),
19     [ResolutionStatus] nvarchar(4000),
20     [DUPID] int
21 )
22 WITH (
23     LOCATION = 'delta/cust_interaction_service/',
24     DATA_SOURCE = [silver_datalakestorage22_dfs_core_windows_net],
25     FILE_FORMAT = [SynapseDeltaFormat]
26 )
27 GO
28
29
30 SELECT TOP 100 * FROM ncpl.project3
31 GO
```

CREATED MASTER KEY,CREDENTIALS,EXTERNAL SOURCE,FILE FORMAT IN SQLDB ALSO. TO TAKE PERMISSIONS FROM STORAGE ACCOUNT,OTHERWISE IT WILL SHOW AN ERROR.

I CONNECTED SQLDB WITH SEVERLESS SQL SYNAPSE ENDPOINT.

```
4 CREATE MASTER KEY ENCRYPTION BY PASSWORD= 'rathore@001';
5 GO
6
7 CREATE DATABASE SCOPED CREDENTIAL projectstorage1
8 WITH
9     IDENTITY = 'SHARED ACCESS SIGNATURE',
10    SECRET = 'sv=2024-11-04&ss=bfqtb&rt=src&sp=rwdlacupyx&se=2025-08-25T01:05:38Z&st=2025-08-24T16:50:38Z&spr=https&sig=z2c%2F8T15iGAAAEw6GzRRAq6AEyoD%2F9wPk
11 GO
12 IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name = 'cred_silver_dfs_core_windows_net')
13     CREATE EXTERNAL DATA SOURCE [cred_silver_dfs_core_windows_net]
14     WITH (
15         LOCATION = 'abfss://silver@datalakestorage22.dfs.core.windows.net',
16         CREDENTIAL= projectstorage1
17     )
18 GO
```

NANDINI RATHORE

```
Run Cancel & Disconnect Change Database: DEMO Estimated Plan Enable Actual Plan Parse
19
20 CREATE EXTERNAL TABLE ncpl.project4 (
21     [InteractionID] nvarchar(4000),
22     [CustomerID] nvarchar(4000),
23     [DateTime] nvarchar(4000),
24     [AgentID] nvarchar(4000),
25     [IssueType] nvarchar(4000),
26     [ResolutionStatus] nvarchar(4000),
27     [DUPID] int
28 )
29 WITH (
30     LOCATION = 'delta/cust_interaction_service/',
31     DATA_SOURCE = [cred_silver_dfs_core_windows_net],
32     FILE_FORMAT = [SynapseDeltaFormat]
33 )
34 GO
35
36
37 SELECT TOP 100 * FROM ncpl.project4
38 GO
```

```
20 ✓ CREATE EXTERNAL TABLE ncpl.project4 (
21     [InteractionID] nvarchar(4000),
22     [CustomerID] nvarchar(4000),
23     [DateTime] nvarchar(4000),
24     [AgentID] nvarchar(4000),
25     [IssueType] nvarchar(4000),
26     [ResolutionStatus] nvarchar(4000).
```

Results Messages

	InteractionID	CustomerID	DateTime	AgentID	IssueType	ResolutionStatus	DUPID
1	44	78	2025-01-10 00:03:03	78	Product Inquiry	Resolved	1
2	42	34	2025-01-06 03:57:24	87	Technical Issue	Resolved	1
3	83	10	2025-03-06 15:32:16	31	Technical Issue	Pending	1
4	17	71	2025-01-15 03:13:48	83	Technical Issue	Resolved	1
5	72	58	2025-01-27 09:55:40	98	Other	Pending	1
6	34	17	2025-01-25 01:38:58	72	Technical Issue	Resolved	1
7	11	69	2025-01-03 21:23:28	4	Complaint	Escalated	1
8	58	15	2025-01-31 09:33:57	34	Product Inquiry	Escalated	1
9	89	33	2025-03-04 03:03:57	43	Billing	Escalated	1
10	40	27	2025-02-17 02:47:01	17	Billing	Escalated	1
11	91	61	2025-01-16 14:09:09	11	Billing	Resolved	1
12	8	50	2025-01-15 16:44:36	66	Other	Escalated	1
13	90	92	2025-01-19 13:54:03	19	Billing	Pending	1
14	94	11	2025-01-28 00:10:43	69	Complaint	Resolved	1
15	19	79	2025-02-06 17:01:07	5	Technical Issue	Pending	1

Creating a view

```
CREATE VIEW Succes AS
SELECT
    AgentID,
    COUNT(*) AS TotalInteractions,
    SUM(CASE WHEN ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) AS ResolutionSuccess,
    SUM(CASE WHEN ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS ResolutionRate
FROM ncpl.project3
GROUP BY AgentID;
SELECT * FROM Succes
```

AgentID	TotalInteractions	ResolutionSuccess	ResolutionRate
21	2	2	100.000000000000
5	1	0	0.000000000000
32	1	0	0.000000000000
95	1	0	0.000000000000

2.ONLINE TRANSACTION- SCD TYPE2

The screenshot shows the 'Source options' tab for an 'onlinetransaction' source. The 'Folder path' is set to 'silver / delta/onlinetransactions'. The 'Time travel' setting is configured to 'Disable'. Other settings include 'Allow no files found' (unchecked), 'Compression type' (No compression), and 'Time travel' (radio button selected for 'Disable').

The screenshot shows the configuration for a target table column. The 'Column name' is 'SRC_hashkey'. The 'Expression' field contains the following code:

```
crc32(concat(toString(SRC_OrderID), SRC_CustomerID, SRC_ProductID, SRC_DateTime), SRC_PaymentMethod, toString(:
```

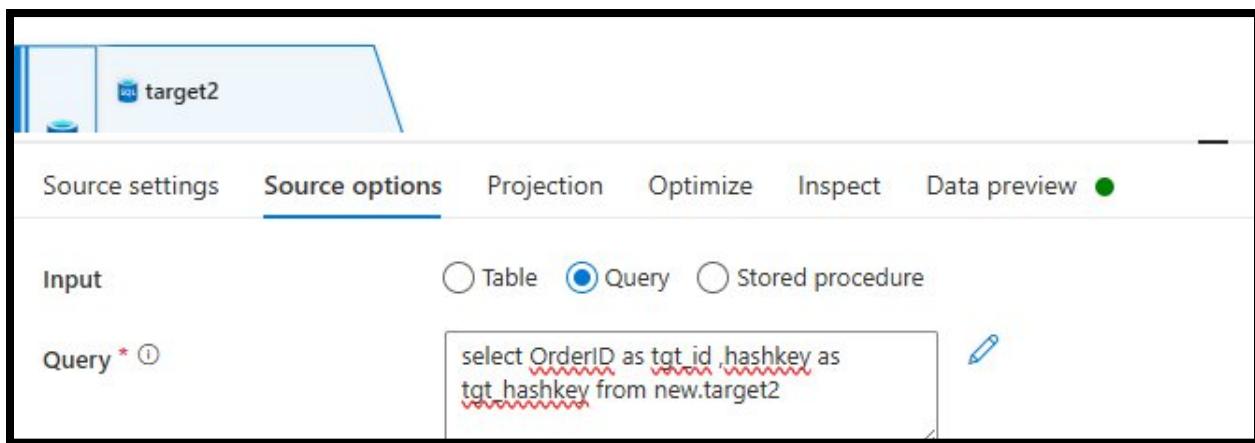
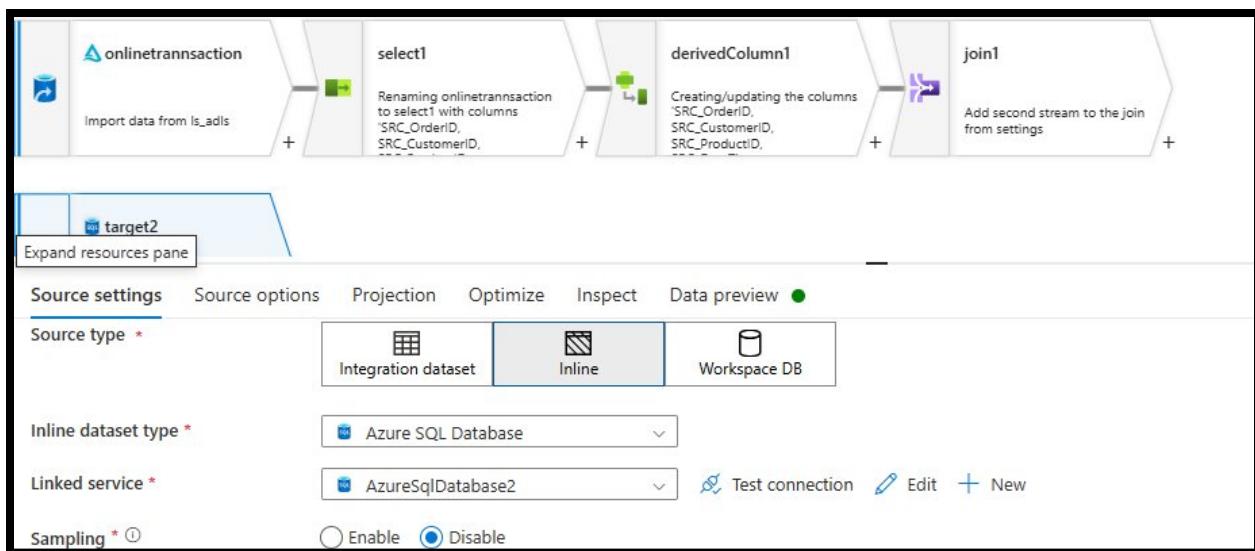
Target table

NANDINI RATHORE

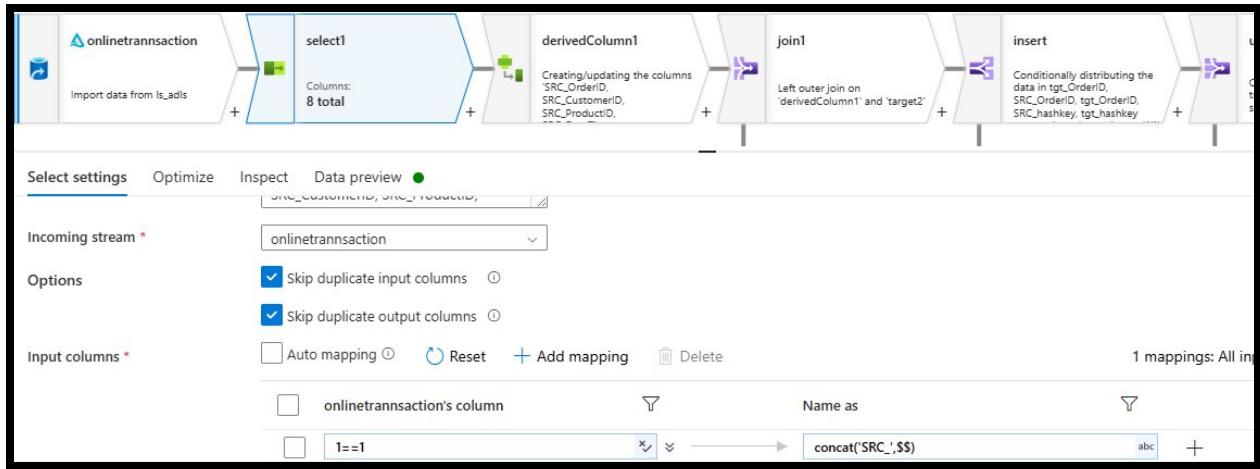
The screenshot shows a SQL query window in SSMS. The query is:

```
1 CREATE TABLE new.target2(
2     OrderID int, CustomerID int, ProductID int, DateTime DATETIME, PaymentMethod Varchar(50),
3     Amount float,
4     Status VARCHAR(50),
5     hashkey BIGINT,
6     updatedby VARCHAR(50),
7     createdby VARCHAR(50),
8     createddate DATETIME,
9     IsActive INT,
10    updateddate DATETIME
11 );
12 select * from new.target2
13
```

The results pane is empty.



NANDINI RATHORE



Derived column's settings dialog for the 'derivedColumn1' step.

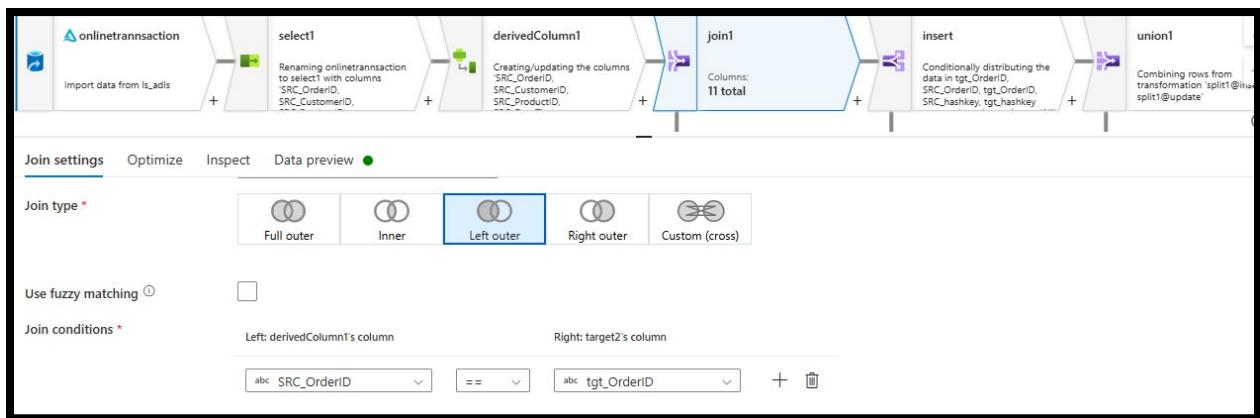
Description: Creating/updating the columns 'SRC_OrderID', 'SRC_CustomerID', 'SRC_ProductID', 'SRC_DateTime'.

Incoming stream *: select1

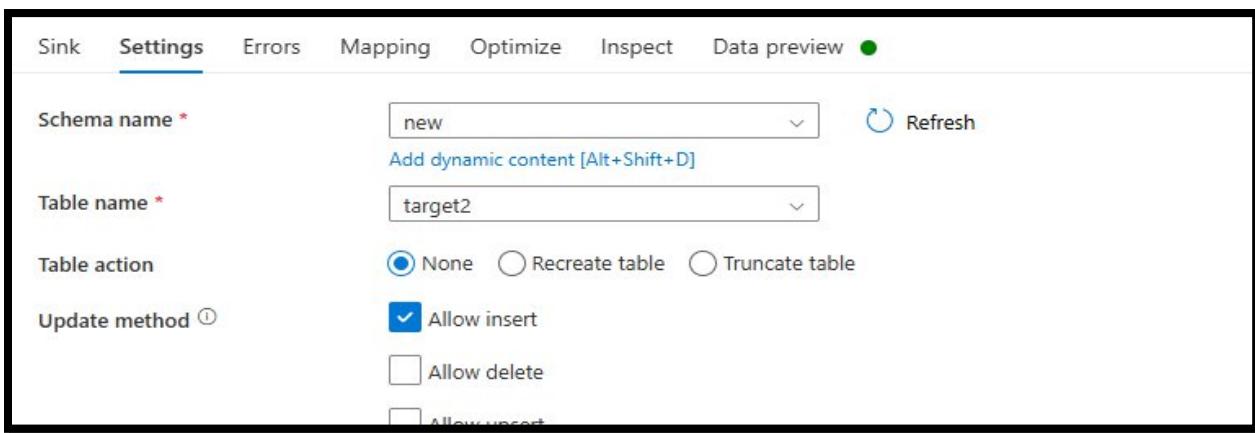
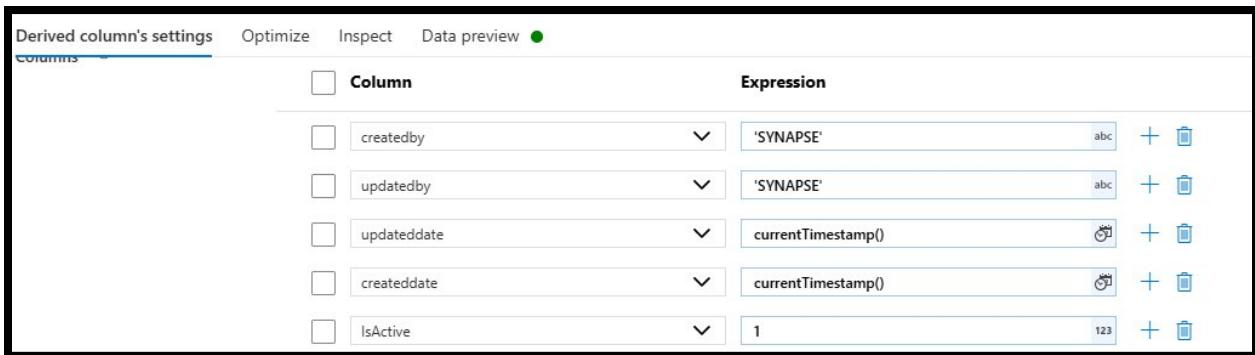
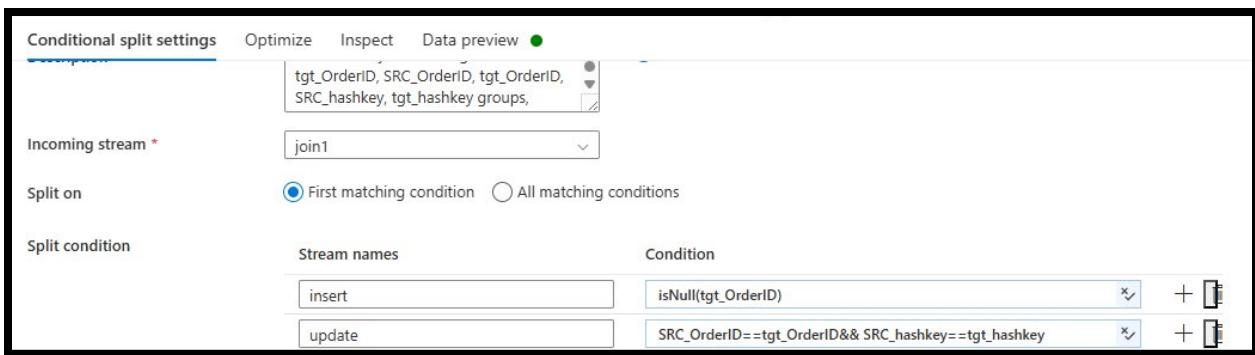
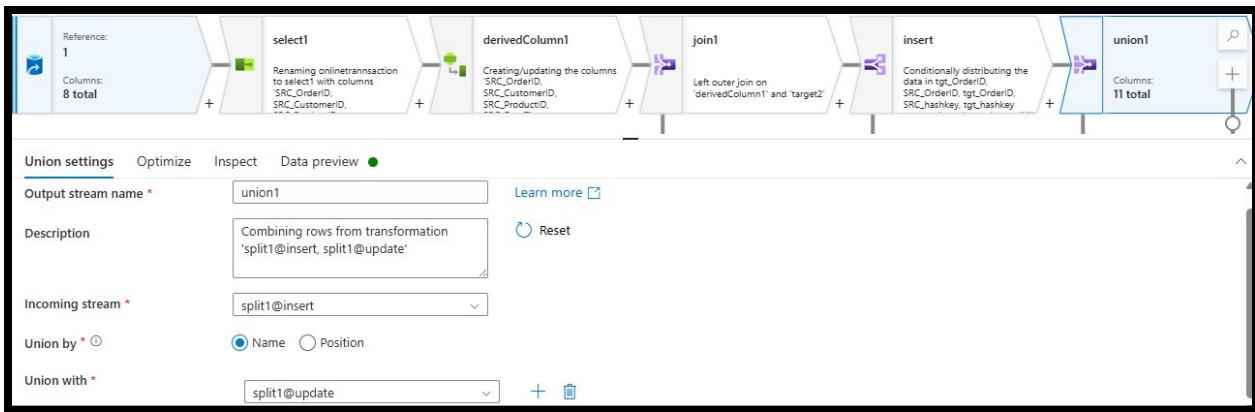
Columns * ⓘ:

Column	Expression
SRC_hashkey	crc32(concat(toString(SRC_OrderID),toString(SR... 121))

Optimize, **Inspect**, **Data preview** tabs are also visible.



NANDINI RATHORE



NANDINI RATHORE

Mapping Tab (Input columns vs Output columns)

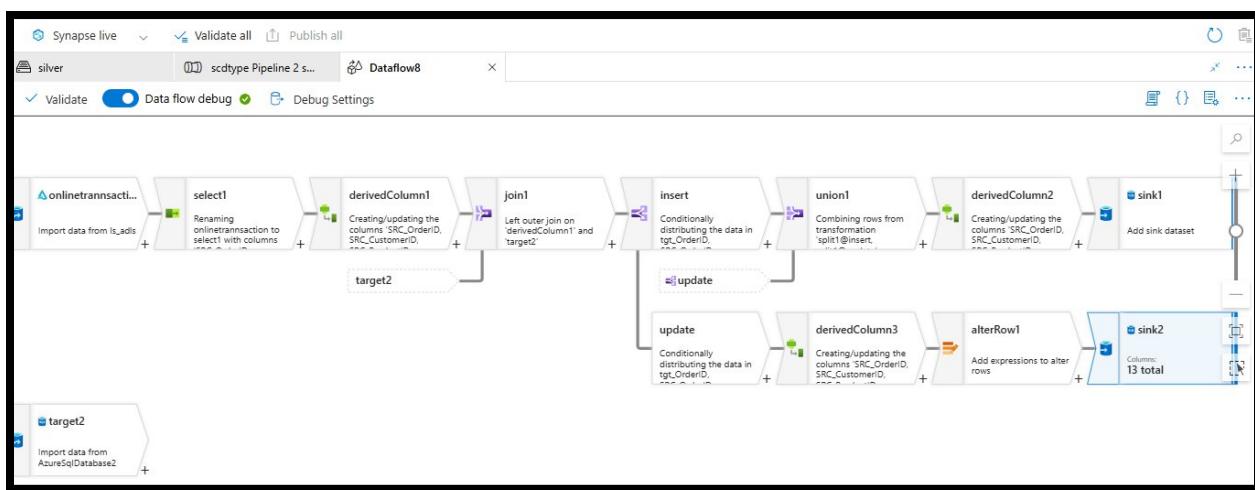
Input columns	Output columns
abc SRC_CustomerID	abc CustomerID
abc SRC_ProductID	abc ProductID
abc SRC_DateTime	abc DateTime
abc SRC_PaymentMethod	abc PaymentMethod
abc SRC_Amount	12 Amount
abc SRC_Status	abc Status
121 SRC_hashkey	121 hashkey
abc updatedby	abc updatedby
abc createdby	abc createdby
abc createddate	abc createddate
123 IsActive	123 IsActive

Derived column's settings Tab

Incoming stream: split1@update

Columns:

Column	Expression
createddate	currentTimestamp()
createdby	'synapse'
updateddate	currentTimestamp()
updatedby	'synapse'
IsActive	0



NANDINI RATHORE

```

3     OrderID VARCHAR(50),CustomerID VARCHAR(50),ProductID varchar(100),DateTime DATETIME,PaymentMethod Varchar(50),
4     Amount float,
5     Status VARCHAR(50),
6     hashkey BIGINT,
7     updatedby VARCHAR(50),
8     createdby VARCHAR(50),
9     createddate DATETIME,
10    IsActive INT,
11    updateddate DATETIME
12 );
13 select * from new.target2

```

Results Messages

	PaymentMethod	Amount	Status	hashkey	updatedby	createdby	createddate	IsActive	updateddate
:39.000	PayPal	109.86	Pending	4204031711	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:20.000	PayPal	91.74	Failed	2547056265	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:18.000	Credit Card	47.99	Pending	2045784130	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:29.000	Credit Card	94.31	Failed	774481334	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:47.000	Gift Card	71.25	Completed	2203407420	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:38.000	Credit Card	17.41	Failed	4086692138	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:39.000	Debit Card	104.86	Failed	1647668209	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:40.000	Debit Card	70.5	Pending	3535258902	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:35.000	Credit Card	57.85	Pending	779325532	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:22.000	PayPal	145.31	Pending	4191006348	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:20.000	Credit Card	90.37	Failed	2641955715	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:29.000	PayPal	78.53	Completed	2774225325	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
:13.000	Credit Card	187.64	Completed	1828626749	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207

New.target2 table in sqldb.

Pipeline run ID: 8df691fd-71a6-40da-ba55-b173969744de

Pipeline status: Succeeded

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID
Data flow1	Succeeded	Data flow	8/24/2025, 9:57:21 PM	1m 55s	AutoResolveIntegrationRuntime (Canada Central)	a399bc62-d656-4	

Results Messages

PaymentMethod	Amount	Status	hashkey	updatedby	createdby	createddate	IsActive	updateddate
Gift Card	71.25	Completed	1000806666	synapse	synapse	2025-08-25 02:08:55.257	0	2025-08-25 02:08:55
PayPal	109.86	Pending	3548010882	synapse	synapse	2025-08-25 02:08:55.257	0	2025-08-25 02:08:55
PayPal	91.74	Failed	4158057614	synapse	synapse	2025-08-25 02:08:55.257	0	2025-08-25 02:08:55
Credit Card	47.99	Pending	2752377160	synapse	synapse	2025-08-25 02:08:55.257	0	2025-08-25 02:08:55

Updated few columns.

Now creating external tables for silver delta tables.

1.customer_interaction_service—this showed previously.

2.IN STORE-

```
13  CREATE EXTERNAL TABLE PROJECT.INSTORE (
14      [TransactionID] nvarchar(4000),
15      [CustomerID] nvarchar(4000),
16      [StoreID] nvarchar(4000),
17      [DateTime] nvarchar(4000),
18      [Amount] nvarchar(4000),
19      [PaymentMethod] nvarchar(4000),
20      [DUPID] int
21  )
22  WITH (
23      LOCATION = 'delta/IN STORE/',
24      DATA_SOURCE = [silver_datalakestorage22_dfs_core_windows_net],
25      FILE_FORMAT = [SynapseDeltaFormat]
26  )
27 GO
28
29
30  SELECT TOP 100 * FROM PROJECT.INSTORE
31 GO
```

Results Messages

00:00:09 Query executed successfully.

3. LOYALTY ACCOUNT.

```
13  CREATE EXTERNAL TABLE PROJECT.LOYALTY (
14      [LoyaltyID] nvarchar(4000),
15      [CustomerID] nvarchar(4000),
16      [PointsEarned] nvarchar(4000),
17      [TierLevel] nvarchar(4000),
18      [JoinDate] nvarchar(4000),
19      [dupid] int
20  )
21  WITH (
22      LOCATION = 'delta/loyalty/',
23      DATA_SOURCE = [silver_datalakestorage22_dfs_core_windows_net],
24      FILE_FORMAT = [SynapseDeltaFormat]
25  )
26 GO
```

4.online transaction

```

13 ✓ CREATE EXTERNAL TABLE PROJECT.ONLINETRANSACTIOnS (
14     [OrderID] nvarchar(4000),
15     [CustomerID] nvarchar(4000),
16     [ProductID] nvarchar(4000),
17     [DateTime] nvarchar(4000),
18     [PaymentMethod] nvarchar(4000),
19     [Amount] nvarchar(4000),
20     [Status] nvarchar(4000),
21     [DUPID] int
22 )
23 WITH (
24     LOCATION = 'delta/onlinetransactions/',
25     DATA_SOURCE = [silver_datalakestorage22_dfs_core_windows_net],
26     FILE_FORMAT = [SynapseDeltaFormat]
27 )
28 GO
29
30

```

Results Messages

✓ 00:00:03 Query executed successfully.

CREATING VIEWS

1. FOR NUMBER OF INTERACTIONS AND RESOLUTION SUCCESS RATES PER AGENT(RESOLUTION STATUS).

```

CREATE VIEW Succes AS
SELECT
    AgentID,
    COUNT(*) AS TotalInteractions,
    SUM(CASE WHEN ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) AS ResolutionSuccess,
    SUM(CASE WHEN ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS ResolutionRate
FROM ncpl.project3
GROUP BY AgentID;
SELECT * FROM Succes

```

AgentID	TotalInteractions	ResolutionSuccess	ResolutionRate
21	2	2	100.0000000000000
5	1	0	0.0000000000000
32	1	0	0.0000000000000
95	1	0	0.0000000000000

2. FOR SEGMENT CUSTOMERS BASED ON TOTAL SPEND ,PURCHASE FREQUENCY AND LOYALTY TIER(LOYALTY ACCOUNTS.TIER LEVEL.)

```

1 CREATE OR ALTER VIEW CustomerSegmentation AS
2 SELECT
3     L.CustomerID,
4     SUM(CAST(O.Amount AS float)) AS TotalSpend,
5     COUNT(O.OrderID) AS PurchaseFrequency,
6     L.TierLevel,
7     CASE
8         WHEN SUM(CAST(O.Amount AS float)) > 5000
9             THEN 'High-Value Customer'
10        WHEN COUNT(O.OrderID) = 1
11            THEN 'One-Time Buyer'
12        WHEN COUNT(O.OrderID) = 2
13            THEN 'second-Time Buyer'
14        WHEN L.TierLevel IN ('Gold','Platinum')
15            THEN 'Loyalty Champion'
16        ELSE 'Regular Customer'
17     END AS Segment
18 FROM [project].[ONLINETRANSACTIONS] O
19 JOIN [project].[LOYALTY] L

```

FROM [project].[ONLINETRANSACTIONS] O
JOIN [project].[LOYALTY] L
ON O.CustomerID = L.CustomerID
GROUP BY L.CustomerID, L.TierLevel;
select * from CustomerSegmentation

Results

CustomerID	TotalSpend	PurchaseFrequency	TierLevel	Segment
5	147.49	1	Gold	One-Time Buyer
6	94.31	1	Gold	One-Time Buyer
18	136.61	1	Silver	One-Time Buyer

3. FOR ANALYZE DATETIME TO FIND PEAK DAYS AND TIMES IN-STORE VS ONLINE.

```

1 CREATE VIEW PeakDayONLINESALES AS
2 SELECT
3     'Online' AS Channel,
4     DATENAME(WEEKDAY, TRY_CAST (DateTime AS DATETIME)) AS DayOfWeek, -----TRY_CAST-converts valid strings, returns NULL for invalid ones.
5     DATEPART(HOUR, TRY_CAST (DateTime AS DATETIME)) AS HourOfDay,
6     COUNT(OrderID) AS TotalOrders,
7     SUM(CAST(Amount AS FLOAT)) AS TotalSales|
8 FROM [project].[ONLINETRANSACTIONS]
9 GROUP BY DATENAME(WEEKDAY, TRY CAST(DateTime AS DATETIME)), DATEPART(HOUR,TRY CAST(DateTime AS DATETIME));

```

NANDINI RATHORE

```
11 CREATE VIEW PeakDaySTORESALES AS
12 SELECT
13     'STORE' AS Channel,
14     DATENAME(WEEKDAY, TRY_CAST (DateTime AS DATETIME)) AS DayOfWeek,      -----TRY_CAST-converts valid strings, returns NULL for invalid ones.
15     DATEPART(HOUR, TRY_CAST (DateTime AS DATETIME)) AS HourOfDay,
16     COUNT(OrderID) AS TotalOrders,
17     SUM(CAST(Amount AS FLOAT)) AS TotalSales
18 FROM [project].[ONLINETRANSACTIONS]
19 GROUP BY DATENAME(WEEKDAY, TRY_CAST(DateTime AS DATETIME)), DATEPART(HOUR, TRY_CAST(DateTime AS DATETIME))
```

```
21 CREATE OR ALTER VIEW CombinedSales AS
22 SELECT * FROM PeakDayONLINESALES
23 UNION ALL
24 SELECT * FROM PeakDaySTORESALES;
25 SELECT * FROM CombinedSales
26
```

Results Messages

View Table Chart Export results ▾

Channel	DayOfWeek	HourOfDay	TotalOrders	TotalSales
Online	Wednesday	1	1	187.64
STORE	Wednesday	1	1	187.64
Online	Wednesday	12	1	57.61

```
34 CREATE OR ALTER VIEW PeakDay_MAXSales AS
35    SELECT
36        Channel, DayOfWeek, TotalSales
37    FROM CombinedSales
38
```

Results Messages

View Table Chart Export results ▾

Channel	DayOfWeek	TotalSales
Online	Tuesday	74.48
Online	Thursday	47.99
Online	Wednesday	164.04

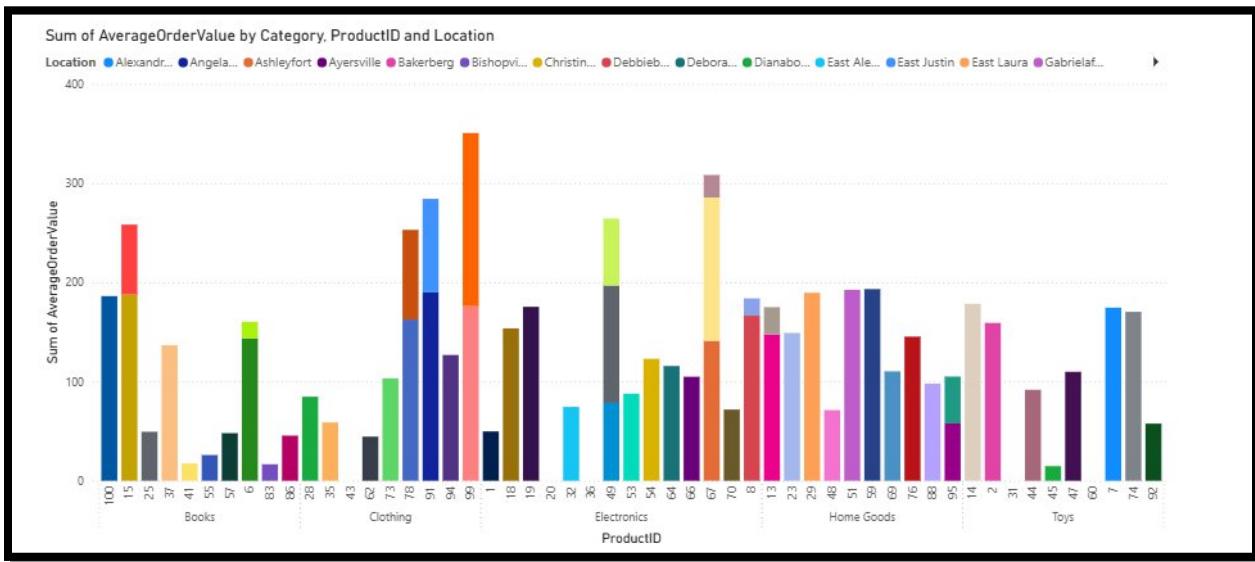
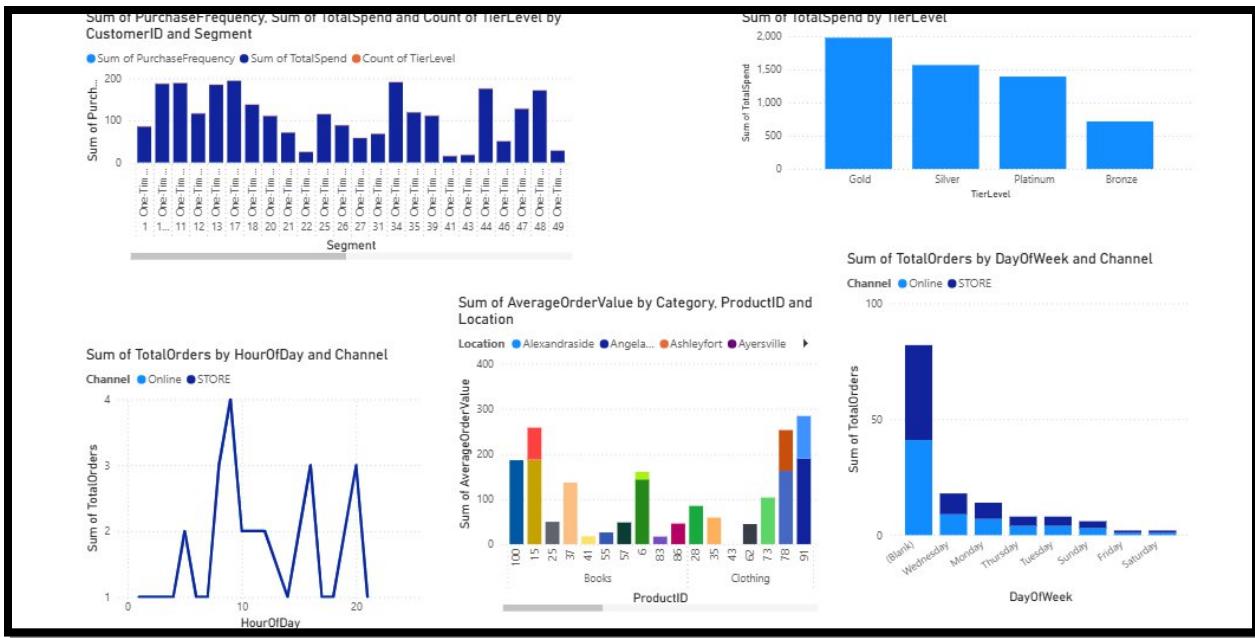
4. AVERAGE ORDER VALUE BY PRODUCTID,CATEGORY AND LOCATION.

```
1 CREATE VIEW View_AOV3 AS
2 SELECT
3     product.ProductID,
4     product.Category,
5     store.Location,
6     AVG(CAST(onlinetransaction.Amount AS FLOAT)) AS AverageOrderValue
7 FROM [project].[ONLINETRANSACtIONS] onlinetransaction
8 JOIN [project].[products2] product ON onlinetransaction.ProductID = product.ProductID
9 JOIN [project].[STORES1] store ON onlinetransaction.OrderID = store.StoreID
10 GROUP BY product.ProductID, product.Category,store.Location;
11
12 select * from View_AOV3
```

Search			
ProductID	Category	Location	AverageOrderValue
18	Electronics	Lake Jameschester	153.63
6	Books	Lake Phillip	143.57
6	Books	North Mariaburgh	16.58
100	Books	Laurenberg	186.14
100	Books	Wellsport	164.38
32	Electronics	East Alexandra	74.48

IN POWER BI

NANDINI RATHORE



NANDINI RATHORE