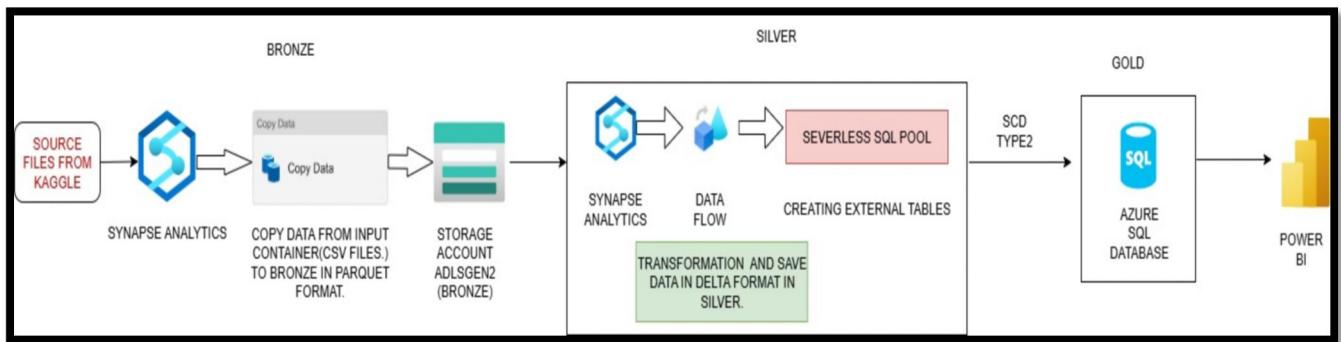


BOOTCAMP PROJECT-3

Customer 360 Data Integration

PROJECT ARCHITECTURE:-



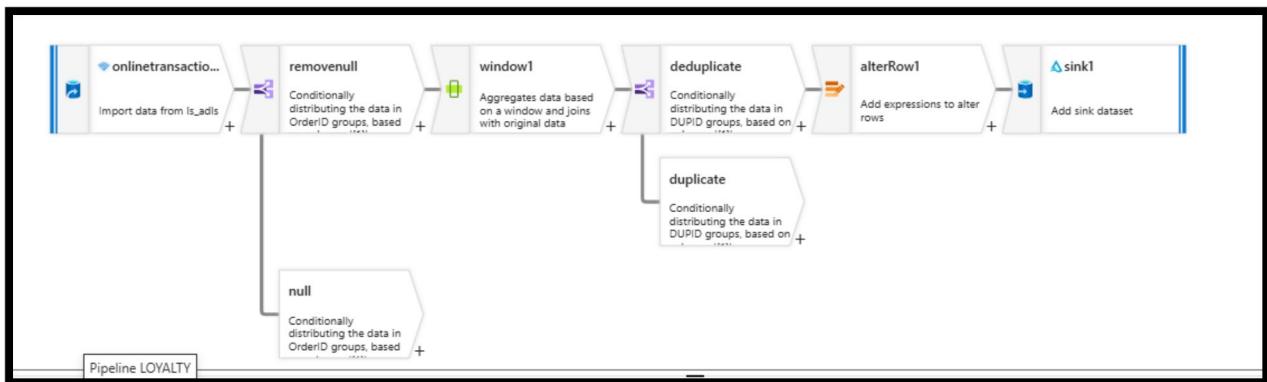
The screenshot shows the Azure Data Factory pipeline editor interface. On the left, there's a sidebar titled "Activities" with a search bar and a list of activity types: Synapse, Move and transform, Azure Data Explorer, Azure Function, Batch Service, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. The main area displays a single activity named "Copy data1" under the "Copy data" category. Below the activity, the "Output" tab is selected, showing details about the pipeline run. The pipeline run ID is 0e52fe0d-3c1f-47b3-8e89-3a7872b974e8, and the status is "Succeeded". The run started on 8/24/2025, 6:01:48 PM, and took 17s. The table below 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

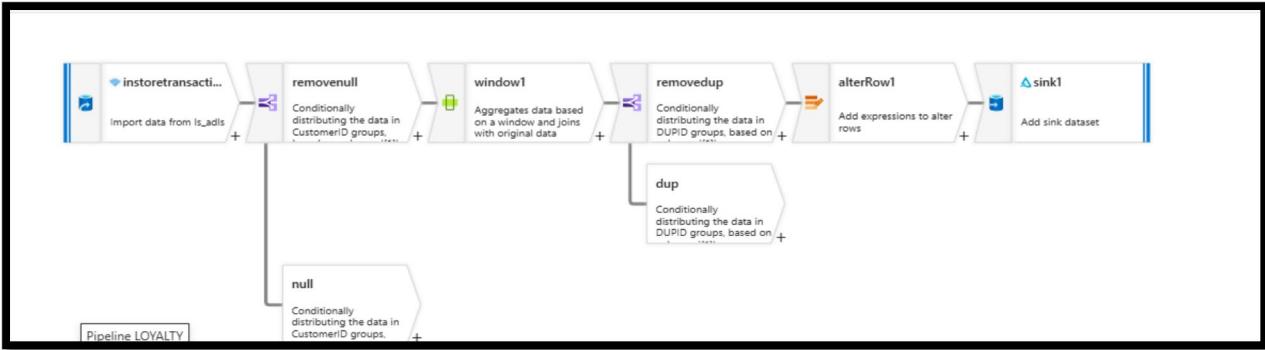
I first use copy activity to load raw csv files from input container to bronze layer in parquet format.

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

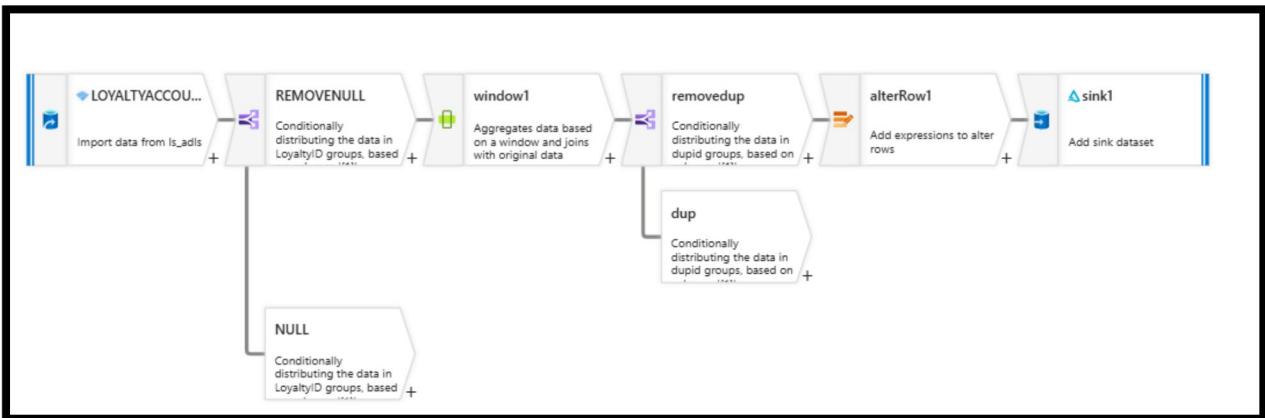
1.ONLINE TRANSACTION



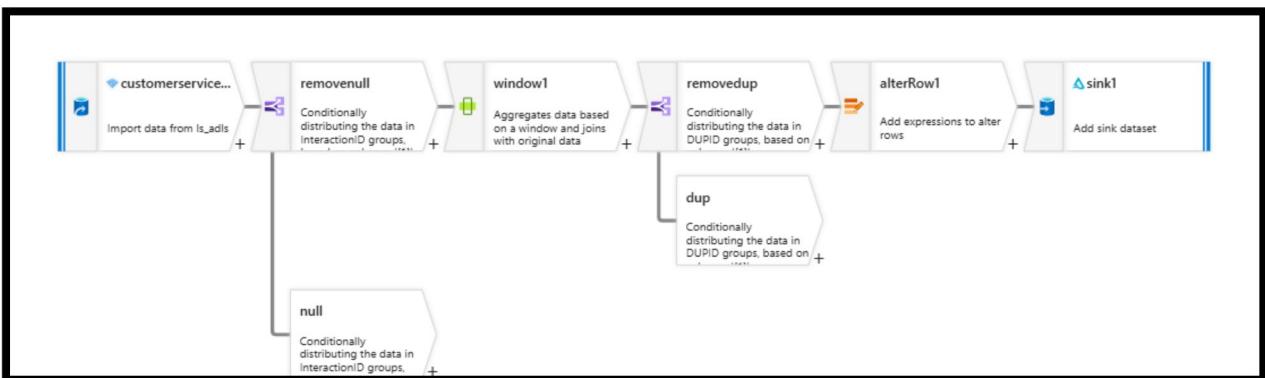
2.STORE PURCHASE

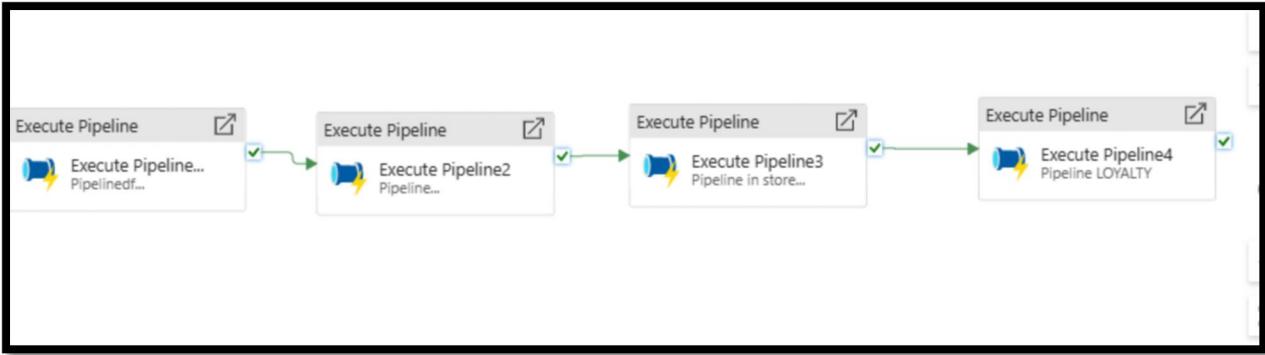


3. LOYALTY ACCOUNTS.



4. CUSTOMER INTERACTION SERVICE.





EXEC PIPELINE TO RUN DATAFLOW PARALLELLY.

THEN PERFORMING SCD TYPE 2:-

1.CUSTOMER INTERACTION SERVICE.

Source settings

Output stream name: source1

Description: Import data from ls_adls

Source type: **Inline**

Inline dataset type: Delta

Linked service: ls_adls

Source settings

Source options

Folder path: silver / delta/cust_interaction_service

Allow no files found:

Compression type: No compression

Time travel: Disable Query by timestamp Query by version

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

Input

Table Query Stored procedure

Query * ⓘ

```
select InteractionID AS tgt_Interaction_ID,
       hashkey as tgt_hashkey from ncpl.target
```

Select settings Optimize Inspect Data preview

columns 'src_InteractionID,
src_CustomerID, src_DateTime,'

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

Description

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

Incoming stream * select1

+ Add Clone Delete Open expression builder

Columns * ⓘ

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

The screenshot shows a Stream Analytics job configuration. At the top, there is a visual representation of the data flow: source1 → select1 → derivedColumn1 → 10 Columns → insert → union1 → derivedColumn2. Below this, the "Join settings" tab is selected. The "Right stream" dropdown is set to "target". The "Join type" section shows five options: Full outer, Inner, Left outer (selected), Right outer, and Custom (cross). The "Use fuzzy matching" checkbox is unchecked. The "Join conditions" section shows two columns: "Left: derivedColumn1's column" and "Right: target's column", with dropdowns for "abc src_InteractionID" and "abc tgt_Interaction_ID" and an equals sign operator (=). There are also "+" and delete icons for adding or removing conditions.

The screenshot shows a Stream Analytics job configuration. The data flow is: source1 → select1 → derivedColumn1 → join1 → 10 Columns → union1 → derivedColumn2 → select3. Below this, the "Conditional split settings" tab is selected. The "Description" field contains the text "Conditionally distributing the data in tgt_Interaction_ID, src_hashkey, tgt_hashkey, src_InteractionID," with a "Reset" button. The "Incoming stream" dropdown is set to "join1". The "Split on" section has a radio button for "First matching condition" (selected) and "All matching conditions". The "Split condition" section lists two entries: "insert" with the condition "isNull(tgt_Interaction_ID)" and "update" with the condition "src_hashkey==tgt_hashkey && src_InteractionID==tgt_Interaction_ID". Each entry has a delete icon (+) to its right.

The screenshot shows the 'Union settings' configuration page. At the top, there is a visual flow diagram of the data pipeline. Below it, the configuration fields are as follows:

- Output stream name ***: union1
- Description**: Combining rows from transformation 'split1@insert, split1@update'
- Incoming stream ***: split1@insert
- Union by ***: Name Position
- Union with ***: split1@update

The screenshot shows the 'Derived column's settings' configuration page. At the top, there is a visual flow diagram of the data pipeline. Below it, the configuration fields are as follows:

- Columns ***

Column	Expression
updatedby	'synapse'
createdby	'synapse'
createddate	currentTimestamp()
IsActive	1
updateddate	currentTimestamp()

The screenshot shows the 'Sink' configuration page. At the top, there is a visual flow diagram of the data pipeline. Below it, the configuration fields are as follows:

- Sink Settings**
- Schema name ***: ncpl
- Table name ***: target
- Table action**: None Recreate table Truncate table
- Update method**:
 - Allow insert
 - Allow delete
 - Allow upsert
 - Allow update

Mapping

Input columns	Output columns
abc_src_InteractionID	abc_InteractionID
abc_src_CustomerID	abc_CustomerID
abc_src_DateTime	abc_DateTime
abc_src_AgentID	abc_AgentID
abc_src_IssueType	abc_IssueType
abc_src_ResolutionStatus	abc_ResolutionStatus
12l_src_hashkey	12l_hashkey

Derived column's settings

updatedby	'synapse-updated'
createdby	'synapse-updated'
updateddate	currentTimestamp()
createddate	currentTimestamp()
IsActive	0

```

graph LR
    target((target)) --> update[update]
    update --> derivedColumn[derivedColumn3]
    derivedColumn --> alterRow[alterRow1]
    alterRow --> sink2[sink2]
  
```

Alter row settings

Description: Add expressions to alter rows

Incoming stream *: derivedColumn3

Alter row conditions *: Update if 1==1

```

graph LR
    target((target)) --> update[update]
    update --> derivedColumn[derivedColumn3]
    derivedColumn --> alterRow[alterRow1]
    alterRow --> sink2[sink2]
  
```

NANDINI RATHORE

Sink Settings Errors Mapping Optimize Inspect Data preview ●

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 ⓘ

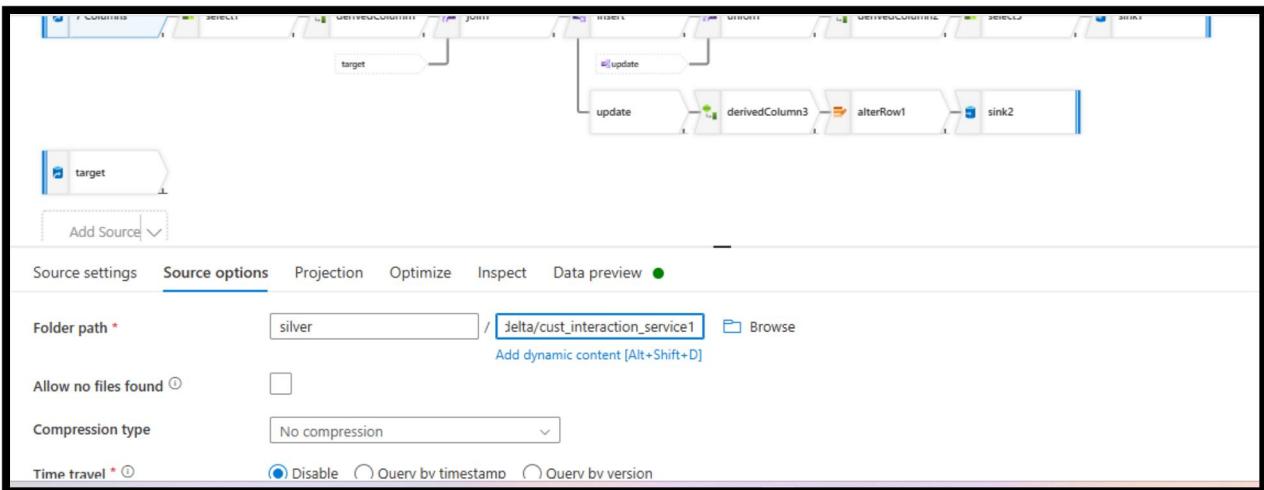
```
16 select * from ncpl.target
17
```

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 being run in a database tool. The query creates a table named "target" with various columns and then selects all rows from it. The results section is currently empty.

```
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
```

16 select * from ncpl.target
 17

Results											Messages	
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-upd...	synapse-upd...	2025-08-24 23:14:25.580	0	2025-08-24 23:14:25.				

CREATING EXTERNAL TABLE.

```

 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_datalakeservice22_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=bfqt&srt=sc0&sp=rwdlacupyx&se=2025-08-25T01:05:38Z&st=2025-08-24T16:50:38Z&spr=https&sig=z2c%2F8Ti5iGAAAEw6GzRRAg6AEyoD%2F9wPkl'
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://silverdatalakeservice22.dfs.core.windows.net',
16         CREDENTIAL= projectstorage1
17     )
18 GO

```

```

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

```

	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
16	51	51	2025-02-16 05:10:26	75	Product Inquiry	Resolved	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	100.0000000000000
5	1	0	0.0000000000000
32	1	0	0.0000000000000
95	1	0	0.0000000000000

2.ONLINE TRANSACTION- SCD TYPE2

The screenshot shows the 'Source options' tab for the 'onlinetransaction' dataset. The 'Folder path' is set to 'silver / delta/onlinetransactions'. The 'Compression type' is 'No compression'. The 'Time travel' setting is 'Disable'.

The screenshot shows the expression configuration for generating a hash key. The column name is 'SRC_hashkey' and the expression is 'crc32(concat(toString(SRC_OrderID), SRC_CustomerID, SRC_ProductID, SRC_DateTime), SRC_PaymentMethod, toString(:'))'.

Target table

The screenshot shows the SQL query editor with the following code:

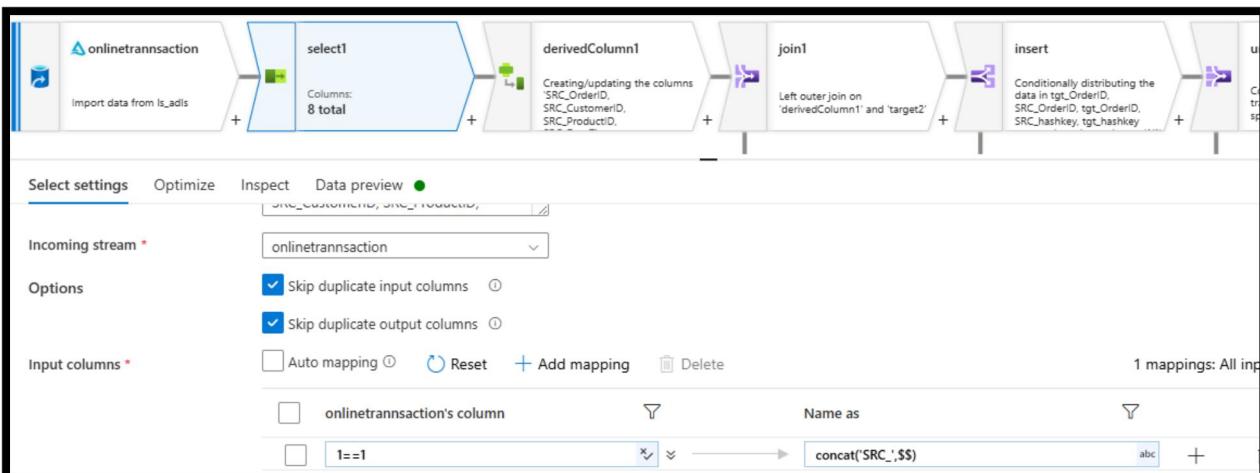
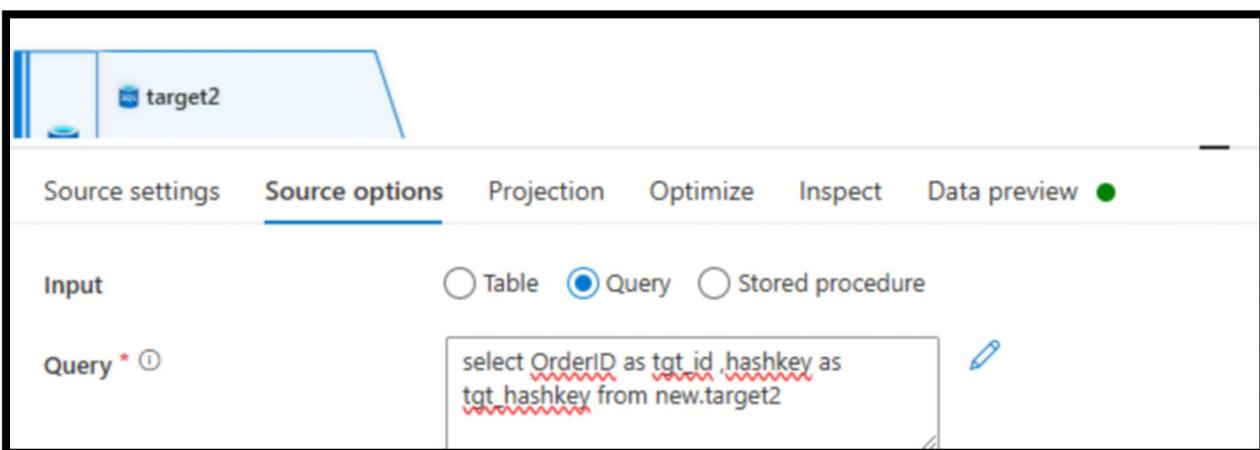
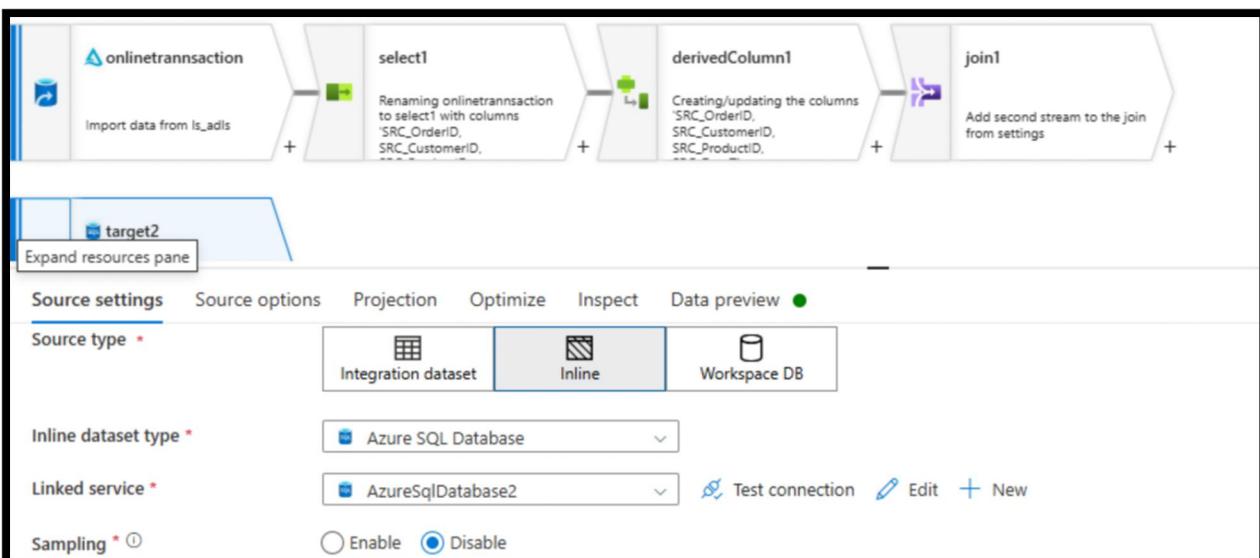
```

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
14

```

The results pane shows the newly created table structure with columns: OrderID, CustomerID, ProductID, DateTime, PaymentMethod, Amount, Status, hashkey, and updatedby.

NANDINI RATHORE



NANDINI RATHORE

Derived column's settings

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... 128))

Flow diagram:

```

graph LR
    A[onlinetransaction] --> B[select1]
    B --> C[derivedColumn1]
    C --> D[join1]
    D --> E[insert]
    E --> F[union1]
    
```

Join settings:

- Join type: Left outer (selected)
- Use fuzzy matching:
- Join conditions:

 - Left: derivedColumn1's column
 - Right: target2's column
 - Condition: abc SRC_OrderID == abc tgt_OrderID

Flow diagram:

```

graph LR
    A[Reference: 1] --> B[select1]
    B --> C[derivedColumn1]
    C --> D[join1]
    D --> E[insert]
    E --> F[union1]
    
```

Union settings:

- Output stream name: union1
- Description: Combining rows from transformation 'split1@insert, split1@update'
- Incoming stream: split1@insert
- Union by: Name (selected)
- Union with: split1@update

Conditional split settings:

Incoming stream: join1

Split on: First matching condition (selected)

Split condition:

Stream names	Condition
insert	isNull(tgt_OrderID)
update	SRC_OrderID == tgt_OrderID & SRC_hashkey == tgt_hashkey

NANDINI RATHORE

Derived column's settings Optimize Inspect Data preview ●

Columns

Column	Expression
createdby	'SYNAPSE'
updatedby	'SYNAPSE'
updateddate	currentTimestamp()
createddate	currentTimestamp()
IsActive	1

Sink Settings Errors Mapping Optimize Inspect Data preview ●

Schema name * new Refresh

Add dynamic content [Alt+Shift+D]

Table name * target2

Table action None Recreate table Truncate table

Update method ⓘ Allow insert
 Allow delete
 Allow upsert

Sink Settings Errors **Mapping** Optimize Inspect Data preview ●

Input columns 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	121 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

NANDINI RATHORE

Derived column's settings

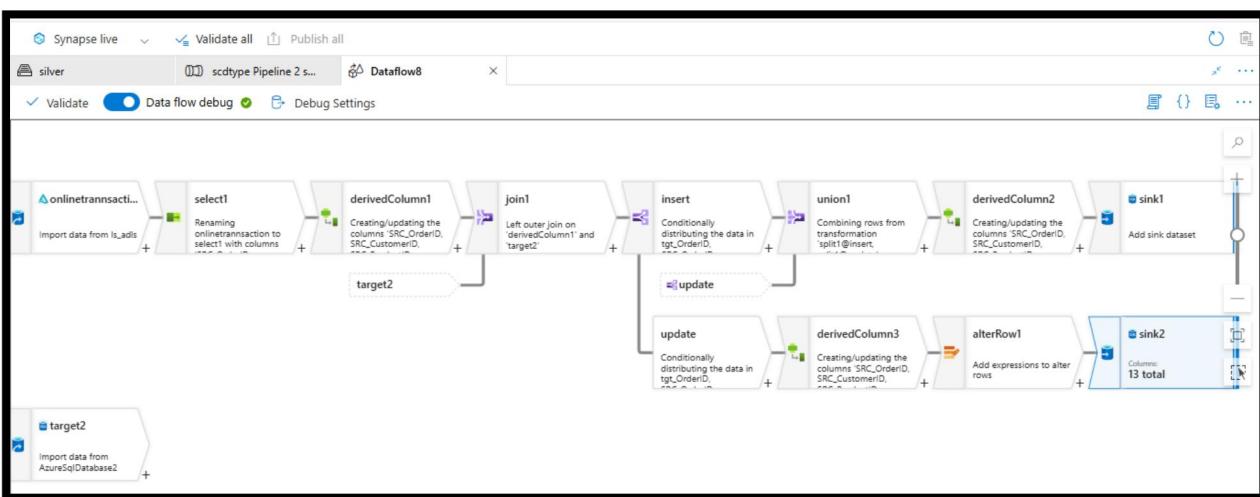
Optimize Inspect Data preview ●

Incoming stream * split1@update

Columns * ①

+ Add Clone Delete Open expression builder

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
4:39.000	PayPal	109.86	Pending	4204031711	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
8:20.000	PayPal	91.74	Failed	2547056265	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
3:18.000	Credit Card	47.99	Pending	2045784130	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
3:29.000	Credit Card	94.31	Failed	774481334	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
7:47.000	Gift Card	71.25	Completed	2203407420	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
5:38.000	Credit Card	17.41	Failed	4086692138	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
3:39.000	Debit Card	104.86	Failed	1647668209	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
6:40.000	Debit Card	70.5	Pending	3535258902	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
2:35.000	Credit Card	57.85	Pending	779325532	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
5:22.000	PayPal	145.31	Pending	4191006348	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
4:20.000	Credit Card	90.37	Failed	2641955715	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
6:29.000	PayPal	78.53	Completed	2774225325	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
1:13.000	Credit Card	187.64	Completed	1828626749	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207
7:00.000	Credit Card	102.33	Completed	3270256100	SYNAPSE	SYNAPSE	2025-08-25 01:39:01.207	1	2025-08-25 01:39:01.207

New.target2 table in sqldb.

Validate | Debug | Add trigger | Data flow debug | Pipeline run ID: 8df691fd-71a6-40da-ba55-b173969744de | Pipeline status: Succeeded | View debug run consumption | Monitor in Azure Metrics | Export to CSV

Parameters | Variables | Settings | Output

Pipeline run ID: 8df691fd-71a6-40da-ba55-b173969744de | Pipeline status: Succeeded | View debug run consumption

All status | Monitor in Azure Metrics | Export to CSV

Show 1 - 1 of 1 items

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
Gift Card	102.33	Completed	3270256100	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
27
```

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].[ONLINETRANSACTI0NS] O
19   JOIN [project].[LOYALTY] L
20  SPR00

```

```

18    FROM [project].[ONLINETRANSACTI0NS] O
19   JOIN [project].[LOYALTY] L
20  |  ON O.CustomerID = L.CustomerID
21  GROUP BY L.CustomerID, L.TierLevel;
22  select * from CustomerSegmentation
23

```

Results Messages

View Table Chart Export results ▾

Search

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].[ONLINETRANSACTI0NS]
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.

```

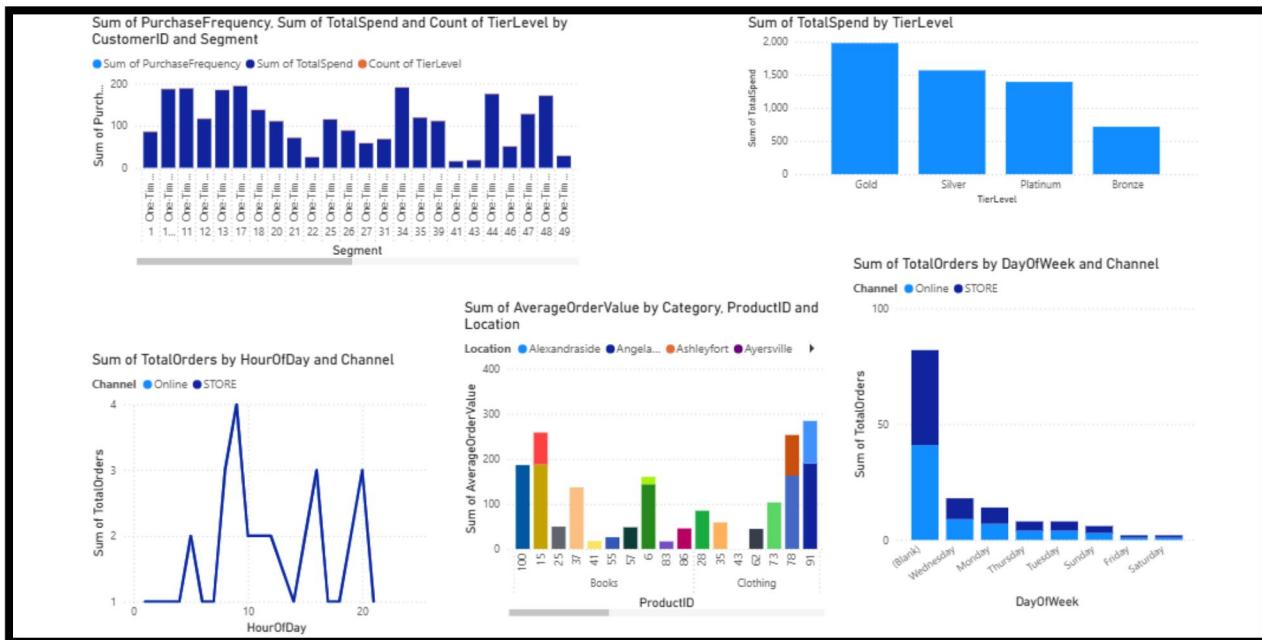
1  CREATE VIEW View_AOV3 AS
2  SELECT
3      product.ProductID,
4      product.Category,
5      store.Location,
6      AVG(CAST(onlinetransaction.Amount AS FLOAT)) AS AverageOrderValue
7  FROM [project].[ONLINETRANSACtions] onlinetransaction
8  JOIN [project].[products2] product ON onlinetransaction.ProductID = 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

