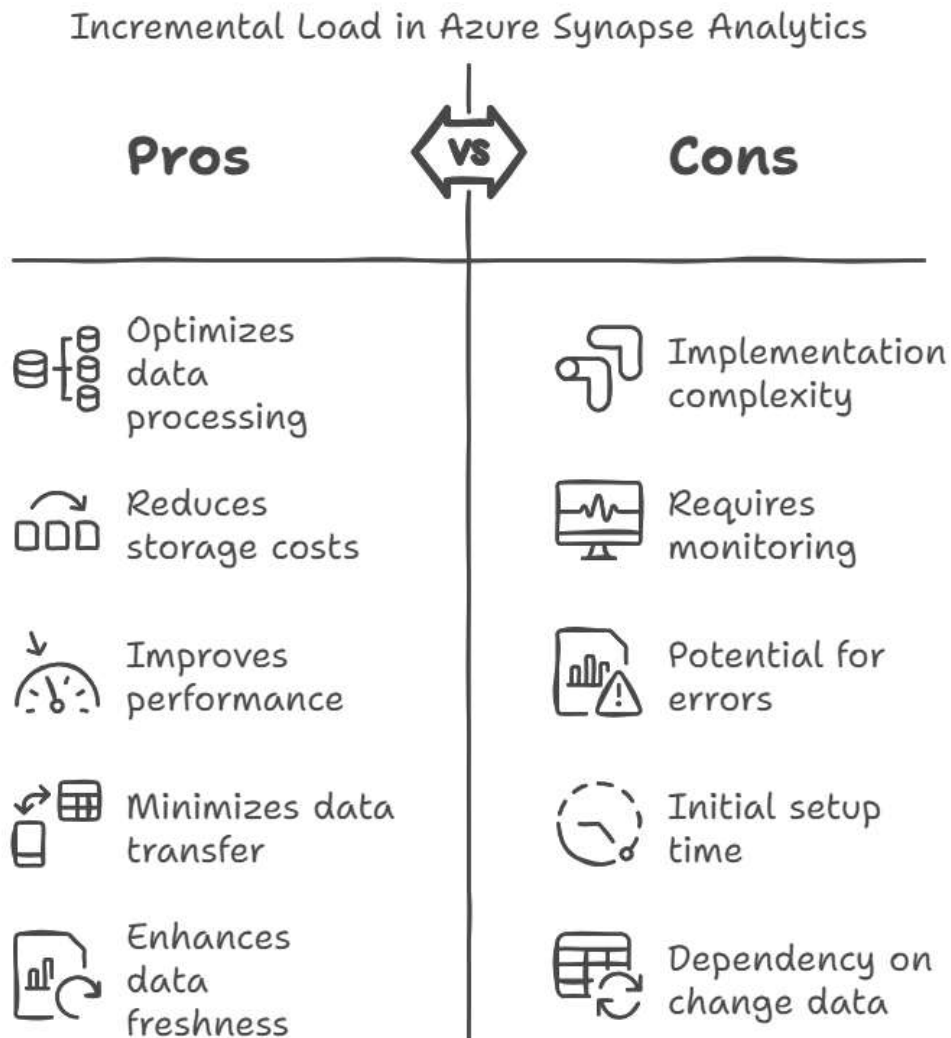




Azure
Synapse
Analytics

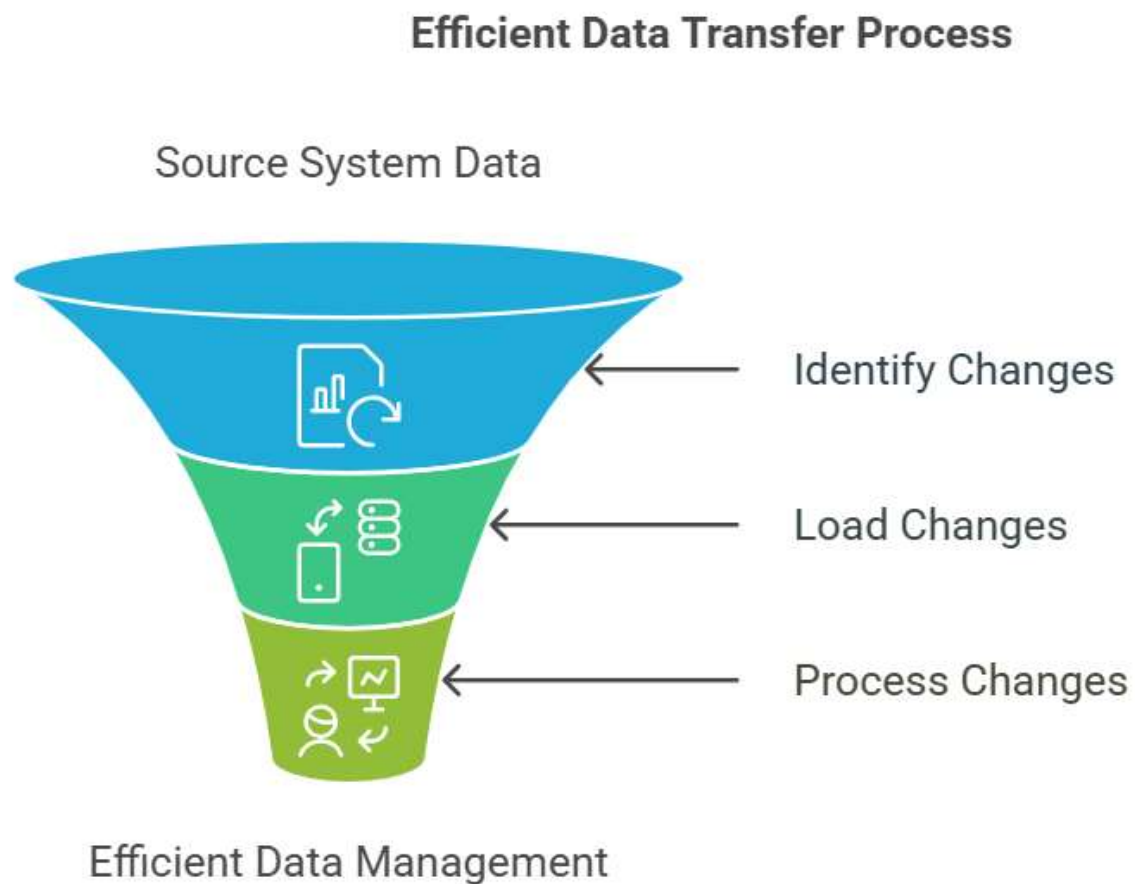
INCREMENTAL LOADING CONCEPT

Incremental load is a crucial concept in data integration and ETL processes, particularly when working with large datasets in Azure Synapse Analytics.



What is the Incremental Load?

Incremental load refers to the process of loading only the new or changed data from a source system into a target system, rather than reloading the entire dataset. This approach is particularly beneficial in scenarios where data volumes are large, and the cost of transferring and processing data can be significant. By focusing on just the changes, organizations can save time, reduce resource consumption, and improve overall efficiency.



Scenario

Assume you have an online retail store, and you have a lot of data being stored, which needs to be analyzed. There is an example case of understanding the changes in the data over period.

Step 01 – Create 5 sample tables

I have created multiple tables as follows:

- ✓ Customer data table (To get the details of the customer)
Create Customer id, Name , Phone number, Customer_datetime

SQL code-

```
--Customer data table (To get the details of the customer )
CREATE TABLE Customer (
    CustomerID INT,
    Name VARCHAR(100) NOT NULL,
    Phone VARCHAR(20),
    Customerupdateddate DATETIME      ---DeltaColumn
);
```

- ✓ Customer login table (To check the time spent online and on what products)
Create Login id, Username, password and login_datetime

SQL code-

```
--Customer login table (To check the time spent online and on what products)
CREATE TABLE Login_id (
    LoginID INT,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(255) NOT NULL,
    Updatedlogindata DATETIME      ---DeltaColumn
);
```

- ✓ Payment table (To get the list of the transactions/payments completed)
Create Transaction_id, Customer ID, Product ID and Transaction_datetime

SQL code-

```
--Payment table (To get the list of the transactions/payments completed)
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY,      ---DeltaColumn
    CustomerID INT NOT NULL,
    ProductID INT NOT NULL,
    TransactionDate DATE NOT NULL,
);
```

- ✓ Inventory table (To get the list of items in the inventory)
Create Product_id, Product_name, Price, Quantity.

SQL code-

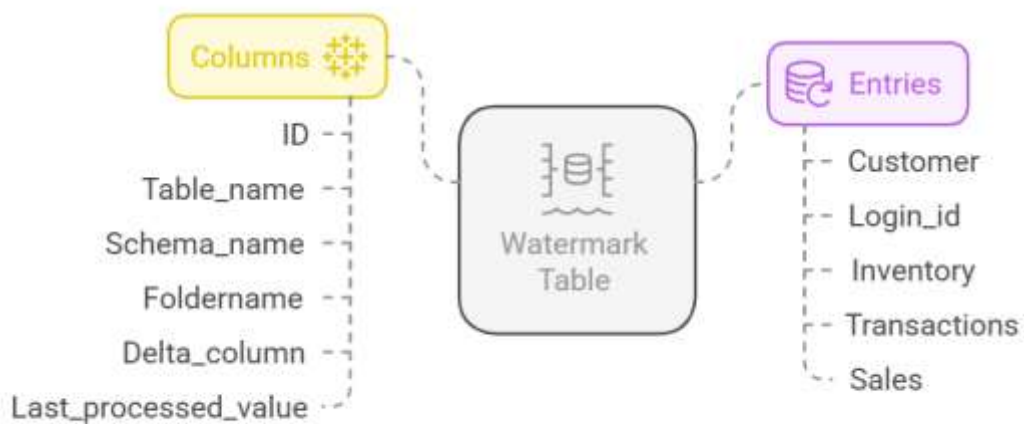
```
--Inventory table (To get the list of items in the inventory)
CREATE TABLE Inventory (
    ProductID INT PRIMARY KEY,          ---DeltaColumn
    ProductName VARCHAR(200) NOT NULL,
    Quantity INT NOT NULL CHECK (Quantity >= 0),
    Price DECIMAL(10, 2) NOT NULL
);
```

- ✓ Sales table (To check the sales of the items)
Create Sales ID, Product ID, Sales_log, Revenue

SQL code-

```
--Sales table (To check the sales of the items)
CREATE TABLE Sales (
    SalesID INT PRIMARY KEY,
    ProductID INT NOT NULL,
    Sales_log DATETIME,                ---DeltaColumn
    Revenue DECIMAL(15, 2) NOT NULL
);
```

Watermark Table Structure and Initialization



Insert values/data into the tables

```
INSERT INTO Customer (CustomerID, Name, Phone, Customerupdateddate)
VALUES
(1, 'John Doe', '123-456-7890', '2023-01-01 10:00:00'),
(2, 'Jane Smith', '987-654-3210', '2023-02-01 14:30:00'),
(3, 'Bob Johnson', NULL, '2023-03-01 09:15:00');
```

```
INSERT INTO Login_id (LoginID, Username, Password, Updatedlogindata)
VALUES
(1, 'johndoe', 'password123', '2023-01-01 10:00:00'),
(2, 'janesmith', 'securepass', '2023-02-01 14:30:00'),
(3, 'bobjohnson', 'secret123', '2023-03-01 09:15:00');
```

```
INSERT INTO Inventory (ProductID, ProductName, Quantity, Price)
VALUES
(1, 'Laptop', 50, 999.99),
(2, 'Smartphone', 100, 699.99),
(3, 'Headphones', 75, 149.99);
```

```
INSERT INTO Transactions (TransactionID, CustomerID, ProductID, TransactionDate)
VALUES
(1, 1, 1, '2023-01-05'),
(2, 2, 2, '2023-02-10'),
(3, 3, 3, '2023-03-15');
```

```
INSERT INTO Sales (SalesID, ProductID, Sales_log, Revenue)
VALUES
(1, 1, '2023-01-05 10:00:00', 999.99),
(2, 2, '2023-02-10 14:30:00', 699.99),
(3, 3, '2023-03-15 09:15:00', 299.98);
```

Step 02 – Create a watermark Table

This step helps to monitor the changes in the data i.e. it may be data entries, data modifications, etc.

SQL Code-

```
--Create a Watermark table
CREATE TABLE Watermark (
    ID INT PRIMARY KEY,                                --Can't accept similar id's or NULL -- Only
    Unique Value
    Table_name VARCHAR(100),
    Schema_name VARCHAR(100),
    Foldername VARCHAR(50),
    Delta_column VARCHAR(100),
    Last_processed_value VARCHAR(255) NOT NULL
);
```

Inserted data into the watermark table

```
-- Initialize watermark entries
INSERT INTO Watermark VALUES
(1, 'Customer', 'dbo', 'RetailDB/Customer_data', 'Customerupdateddate', '1900-01-01 00:00:00'),
(2, 'Login_id', 'dbo', 'RetailDB/Login_id_data', 'Updatedlogindata', '1900-01-01 00:00:00'),
(3, 'Inventory', 'dbo', 'RetailDB/Inventory_data', 'ProductID', '0'),
(4, 'Transactions', 'dbo', 'RetailDB/Transactions_data', 'TransactionID', '0'),
(5, 'Sales', 'dbo', 'RetailDB/Sales_data', 'Sales_log', '1900-01-01 00:00:00')
```

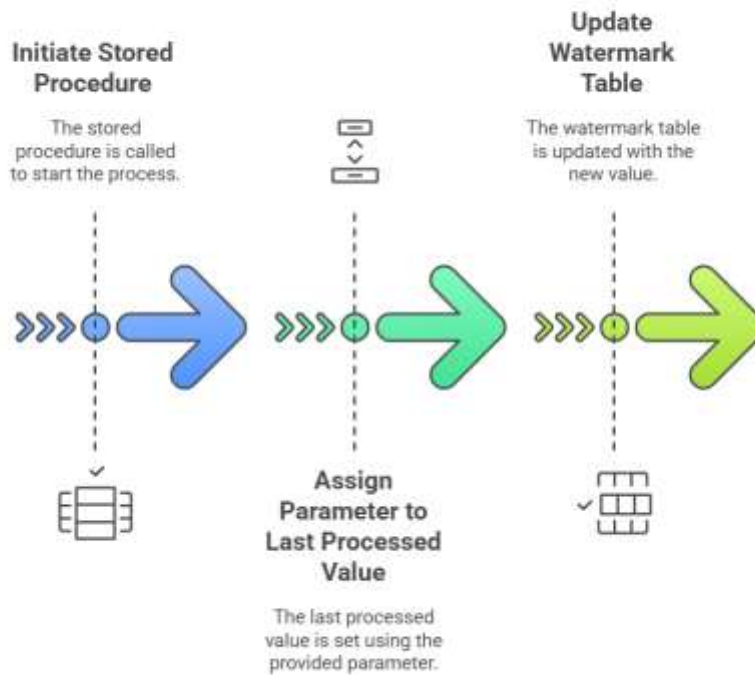
Now, the watermark table is ready to be used.

Step 03 – Create a stored procedure

This helps to assign a parameter to the changes in the last processed values corresponding to the table names.

```
--Create Stored procedure
CREATE PROC USP_Watermark_RetailDB
    @lpv VARCHAR(100),
    @TBname VARCHAR(100)
AS
BEGIN
    UPDATE Watermark
    SET Last_processed_value=@lpv WHERE Table_name=@TBname
END
```

Stored Procedure Execution Sequence



Step 04 – Creation of pipeline in Synapse

- ✓ Create a lookup activity and named it as Getwatermarkdata.
- ✓ Click on source dataset and create a new linked service with SQL server and name its as DS_SQL_Watermark.



- ✓ Open the DS_SQL_Watermark dataset.
- ✓ Create 2 parameters i.e. Schemaname and Tablename for dynamically assigning the Schemaname and Tablename.

Microsoft Azure | Synapse Analytics ▶ synapsedhrj

Synapse live Validate all Publish all

Assignment03_Incr... DS_SQL_Watermark

Data

Azure SQL Database
DS_SQL_Watermark

Connection Schema **Parameters**

+ New Delete

<input type="checkbox"/>	Name	Type	Default value	
<input type="checkbox"/>	Schemaname	String	Value	
<input type="checkbox"/>	Tablename	String	Value	

- ✓ Assigning dynamic variables to the Schema and Table name for the table

Microsoft Azure | Synapse Analytics | synapsedhrj

Synapse live | Validate all | Publish all

Assignment03_Incr... | DS_SQL_Watermark

SQL Azure SQL Database
DS_SQL_Watermark

Connection | Schema | Parameters

Linked service * AzureSqlDatabase1 | Test connection | Edit | + New | Learn more

Integration runtime * AutoResolveIntegrationRuntime | Edit

Table: @dataset().Schemaname . @dataset().Tablename
☒ Enter manually

Preview data

- ✓ Connect the source dataset to DS_SQL_Watermark (Watermark table created in SQL database)
- ✓ Enter the values of fields Schemaname and Tablename as dbo and Watermark respectively.

The screenshot displays the Microsoft Azure Synapse Analytics interface. At the top, the header shows 'Microsoft Azure | Synapse Analytics' and a search bar. Below the header, there's a navigation pane on the left with icons for Home, Databases, Data Lake, and other services. The main workspace shows a 'Lookup' task named 'Getwatermarkdata' in a pipeline. The task is highlighted with a red circle. Below the task, the 'Settings' tab is selected, showing the 'Source dataset' dropdown set to 'DS_SQL_Watermark'. A red box highlights the 'Dataset properties' section, which includes a table with the following data:

Name	Value
Schemaname	dbo
Tablename	Watermark

Below the 'Dataset properties' section, there are options for 'First row only' (unchecked), 'Use query' (radio buttons for Table, Query, Stored procedure, with 'Table' selected), 'Query timeout (minutes)' (120), and 'Isolation level' (Select...). The 'Source dataset' dropdown also has 'Open', 'New', and 'Print' buttons.

- ✓ Create a Foreach activity
- ✓ Connect the lookup activity i.e. Getwatermarkdata with the Foreach activity

The screenshot displays the Microsoft Azure Synapse Analytics interface. At the top, the header shows "Microsoft Azure" and "Synapse Analytics" with a search bar. Below the header, the left sidebar contains navigation icons for Home, Data Lake Storage, Synapse Studio, and other services. The main workspace shows a workflow diagram with a "Lookup" activity named "Getwatermarkdata" connected to a "Foreach" activity named "Foreach1". The "Foreach" activity is expanded, showing an "Activities" section with a plus sign for adding new activities. Below the diagram, the "General" tab is selected, showing the "Name" field with the value "Getmaxvalue", a "Description" field, and the "Activity state" set to "Activated".

Microsoft Azure | Synapse Analytics | synapsedhj | Search

Synapse live | Validate all | Publish all

Assignment03_Incr... X

Validate | Debug | Add trigger

Lookup: Getwatermarkdata

Foreach: ForEach1

Activities

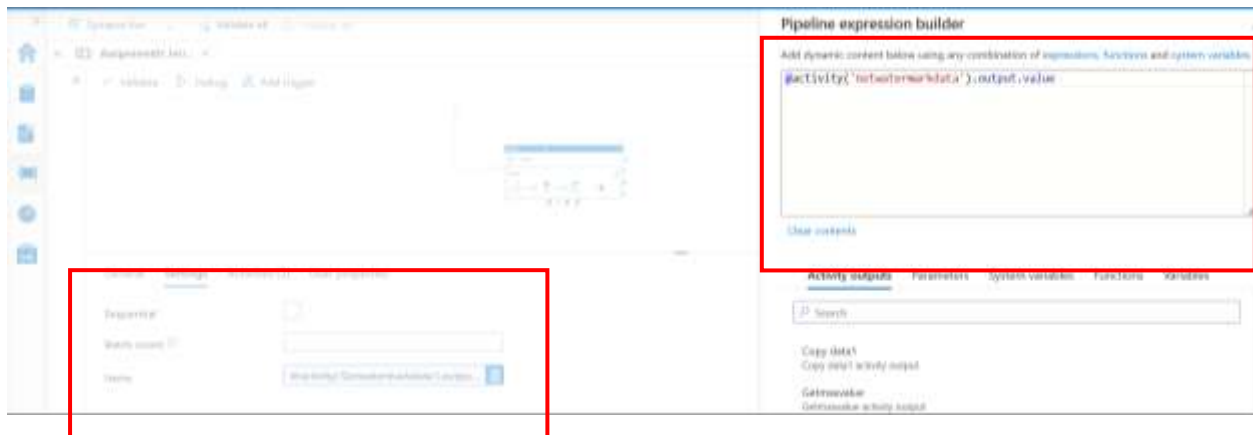
General | Settings | User properties

Name * | Getmaxvalue | Learn more

Description

Activity state | ☒ Activated | ☐ Deactivated

- ✓ Provide an expression in the items field in the Settings tab to connect with the lookup activity.
- ✓ Go to the pipeline expression builder-@activity('Lookup tablename').output.value
i.e. in current scenario @activity('Getwatermarkdata').output.value



- ✓ Created a Lookup activity inside the Foreach activity for deriving the maximum value from the watermark table.
- ✓ Name it as lookup activity as Getmaxvalue

The screenshot displays the Microsoft Azure Synapse Analytics interface. The top navigation bar shows 'Microsoft Azure | Synapse Analytics > synapsedhrj' with a search bar. Below the navigation bar, there are tabs for 'Synapse live', 'Validate all', and 'Publish all'. The main workspace shows a pipeline named 'Assignment03_Incr...' with a 'Validate' button and 'Debug' and 'Add trigger' options. The pipeline contains a 'ForEach1' loop. Inside the loop, a 'Lookup' activity is visible, named 'Getmaxvalue'. The 'Lookup' activity is shown in a preview window with a magnifying glass icon and the name 'Getmaxvalue'. Below the preview, there are icons for deleting, editing, and saving the activity. The bottom section of the interface shows the 'General' tab for the 'Getmaxvalue' activity. It includes fields for 'Name' (Getmaxvalue), 'Description', 'Activity state' (Activated), and 'Timeout' (0.12:00:00). There is also a 'Learn more' link.

Microsoft Azure | Synapse Analytics > synapsedhrj

Synapse live Validate all Publish all

Assignment03_Incr... x

Validate Debug Add trigger

Assignment03_Incremental data loading > ForEach1

Lookup

Getmaxvalue

General Settings User properties

Name * Getmaxvalue [Learn more](#)

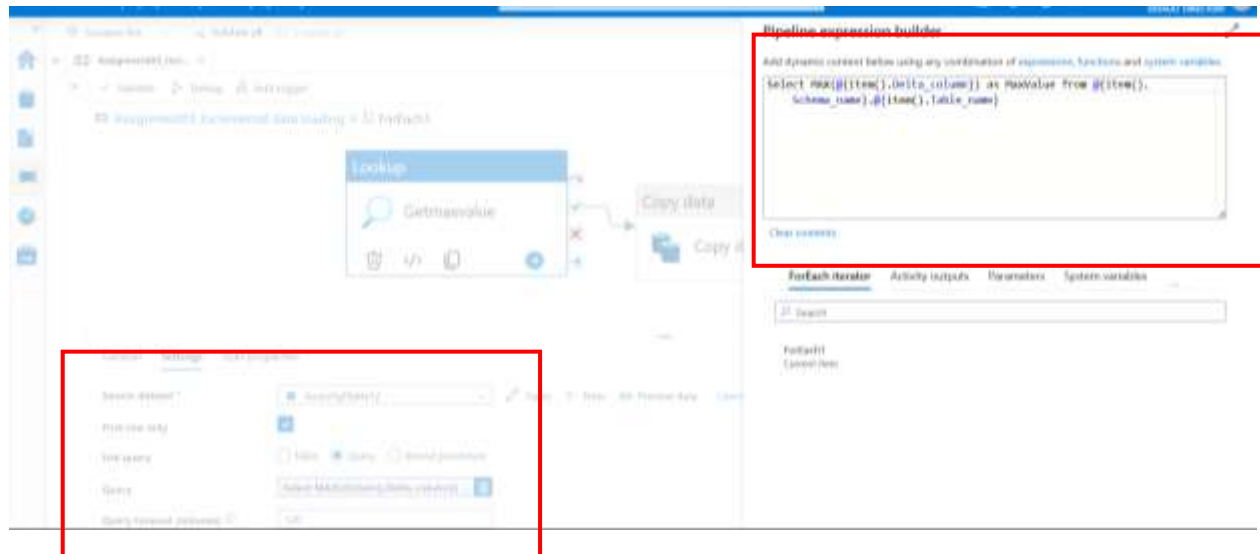
Description

Activity state ☒ Activated ☐ Deactivated

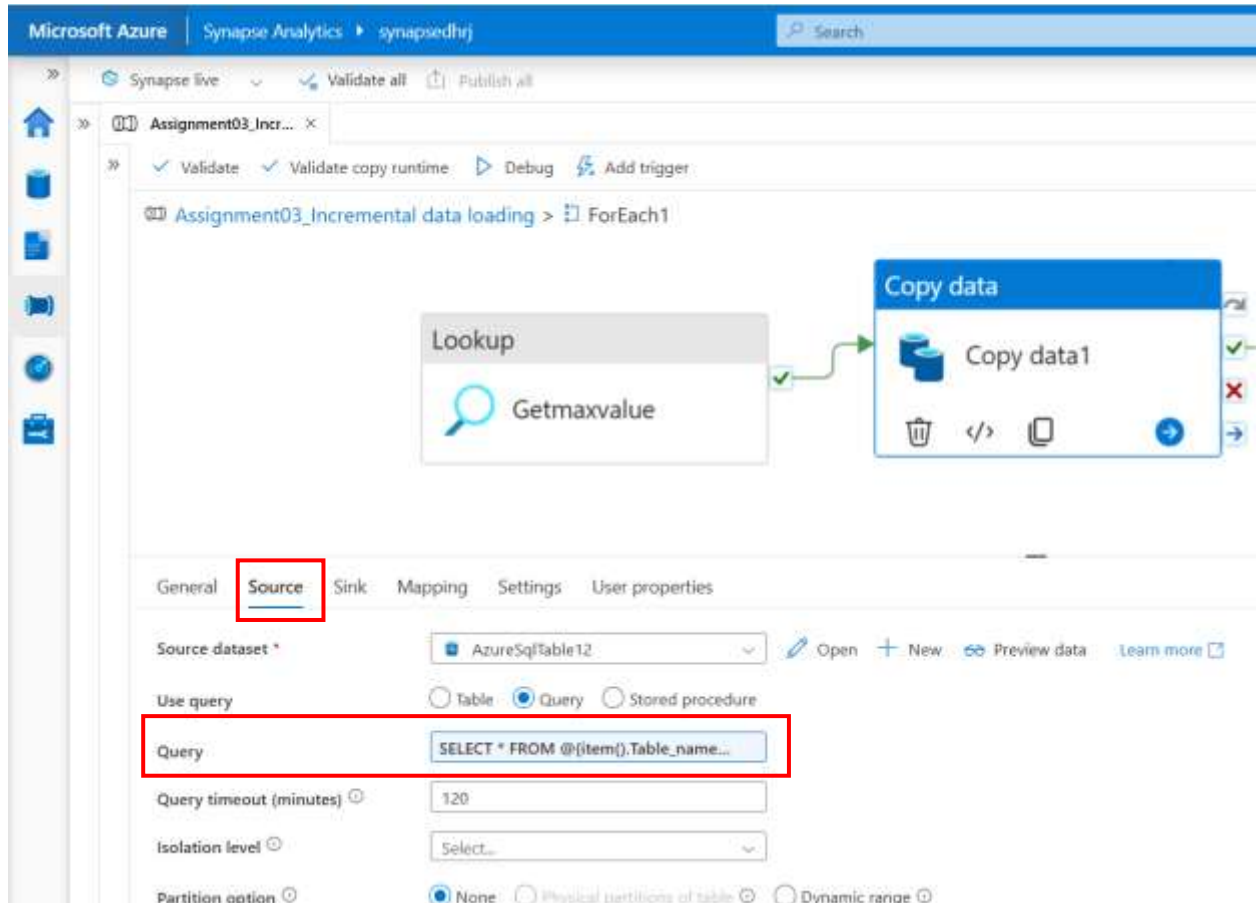
Timeout 0.12:00:00

Notes

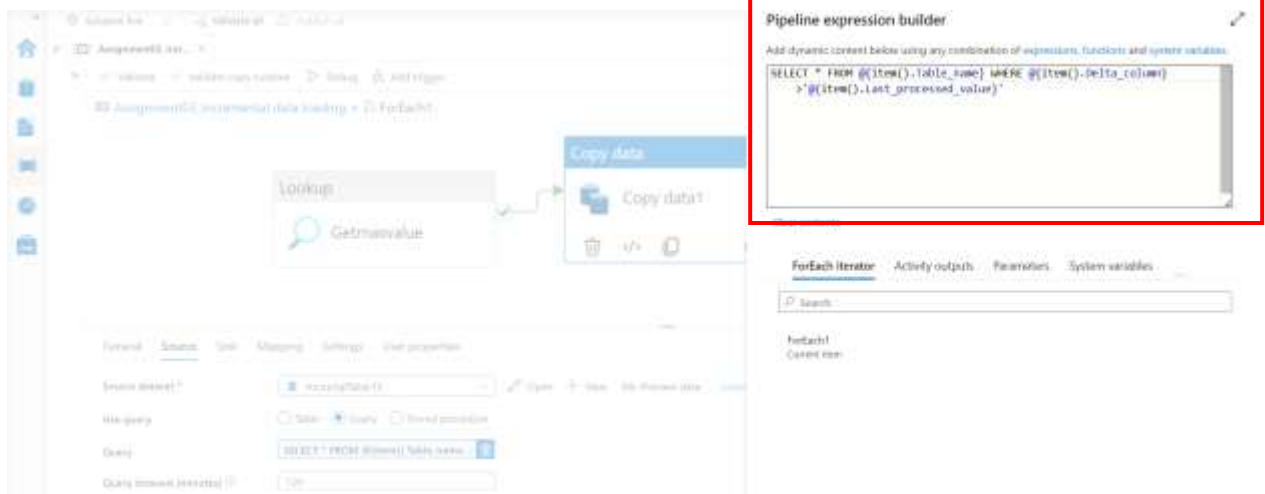
- ✓ Providing a query dynamically to get the maximum value from the watermark table
- ✓ **General** expression (For getting the maximum value from the watermark table)
 SELECT MAX('Columnname') as MaxValue FROM dbo.Watermark
- ✓ **Dynamic** expression (Converting dynamically the above expression)
 SECLECT MAX(@{item().Delta_Columnn)) as MaxValue FROM
 @({item().Schema_name}).@({item().Table_name})



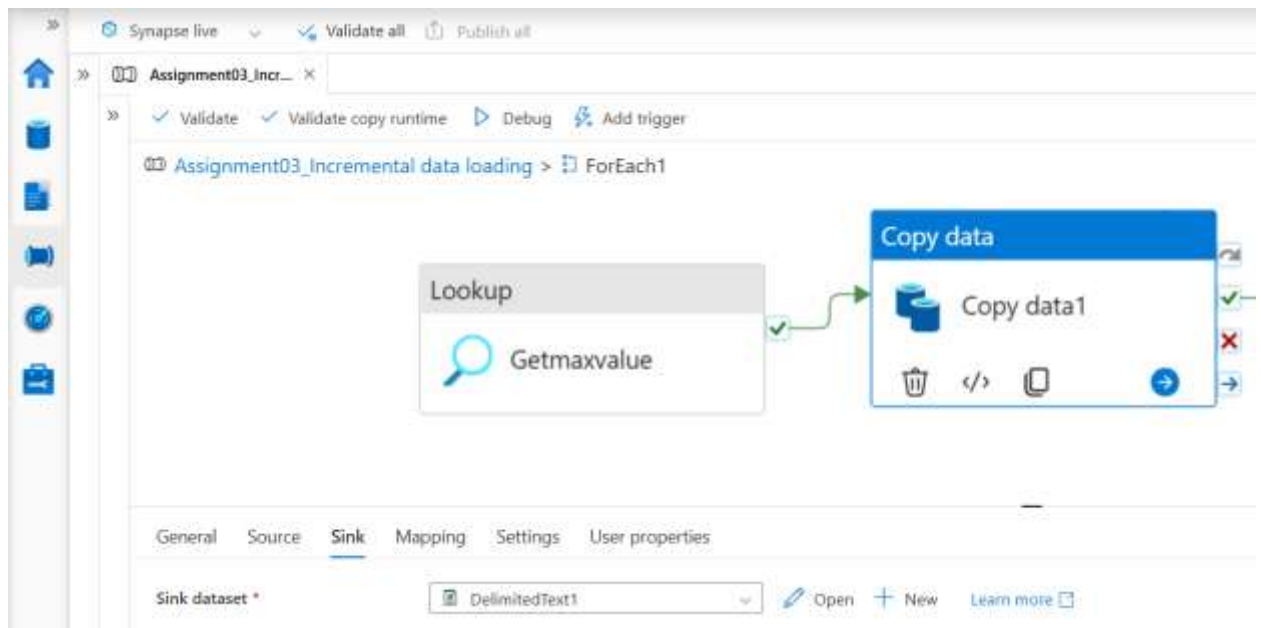
- ✓ Adding a Copy data activity
- ✓ Connect with the lookup activity i.e. Getmaxvalue activity.
- ✓ Select Source and write a dynamic query to copy only the modified values from the watermark table



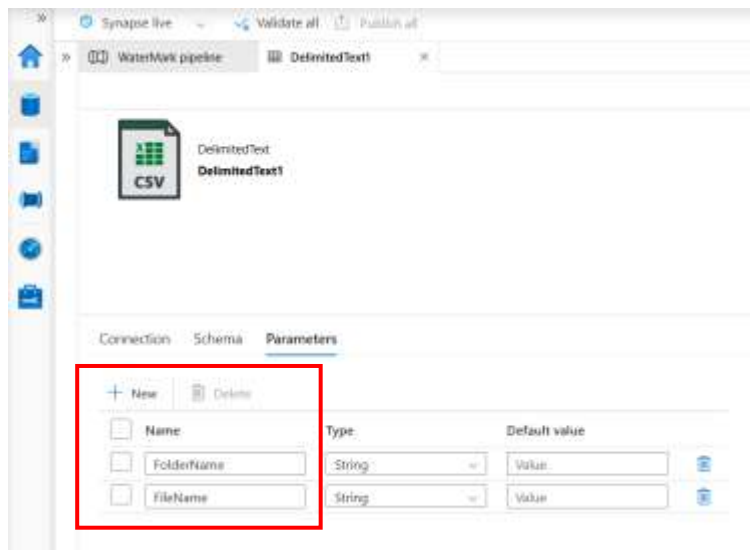
- ✓ **General** expression (To check for modified values or changes in the data entries)
SELECT * FROM TABLE WHERE Delta_Column > 'Last processed value column'
- ✓ **Dynamic** expression (Converting dynamically the above expression)
SELECT * FROM @(item().Table_name) WHERE
@(item().Delta_column)>'@(item().Last_processed_value)



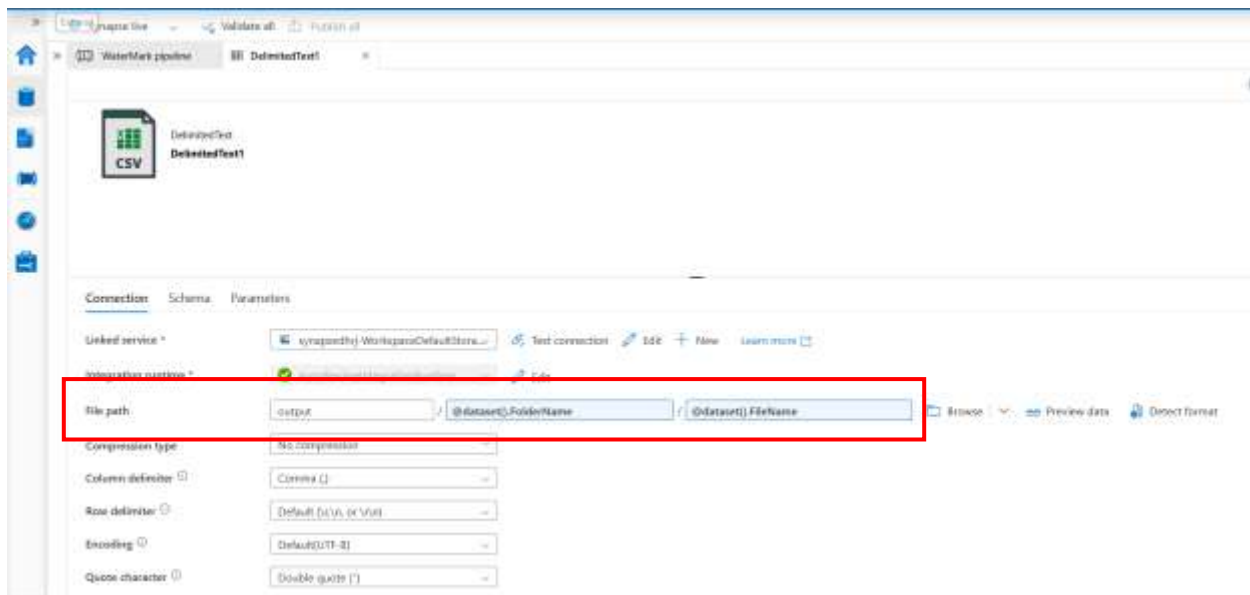
- ✓ Select Sink and create a dataset name DelimitedText1 to store the output into a Azurelakegen2 or Blob storage .



- ✓ Create 2 parameters named FolderName and FileName to store the folder name and file names respectively



- ✓ Assign the parameters dynamically to the folder name and file name in the File path.

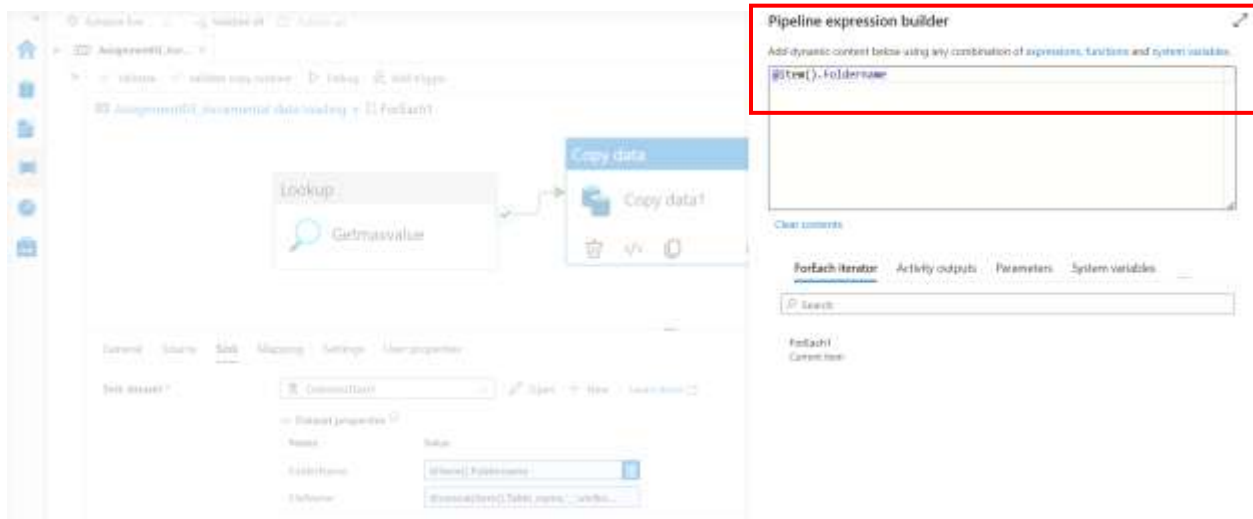


- ✓ Provide expressions as dynamic queries in the Foldername field and Filename fields.

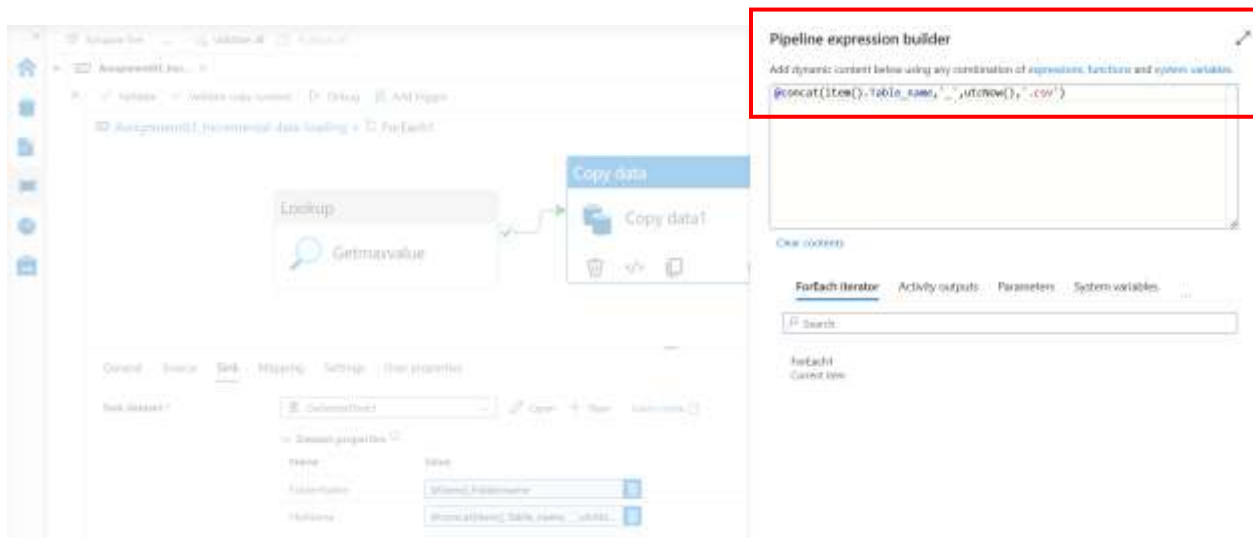
The screenshot shows the Synapse IDE interface. At the top, there are tabs for 'Assignment03_Incr...' and 'Assignment03_Incremental data loading > ForEach1'. Below the tabs, there are buttons for 'Validate', 'Validate copy runtime', 'Debug', and 'Add trigger'. The main workspace displays a data pipeline with two activities: 'Lookup' and 'Copy data'. The 'Lookup' activity has a 'Getmaxvalue' expression. The 'Copy data' activity is labeled 'Copy data1'. Below the workspace, the 'Sink' tab is selected, showing the 'Sink dataset *' as 'DelimitedText1'. The 'Dataset properties' section is expanded, showing a table with 'Name' and 'Value' columns. The 'FolderName' and 'FileName' fields are highlighted with a red box.

Name	Value
FolderName	@item().Foldername
FileName	@concat(item().Table_name,"_utcNo...

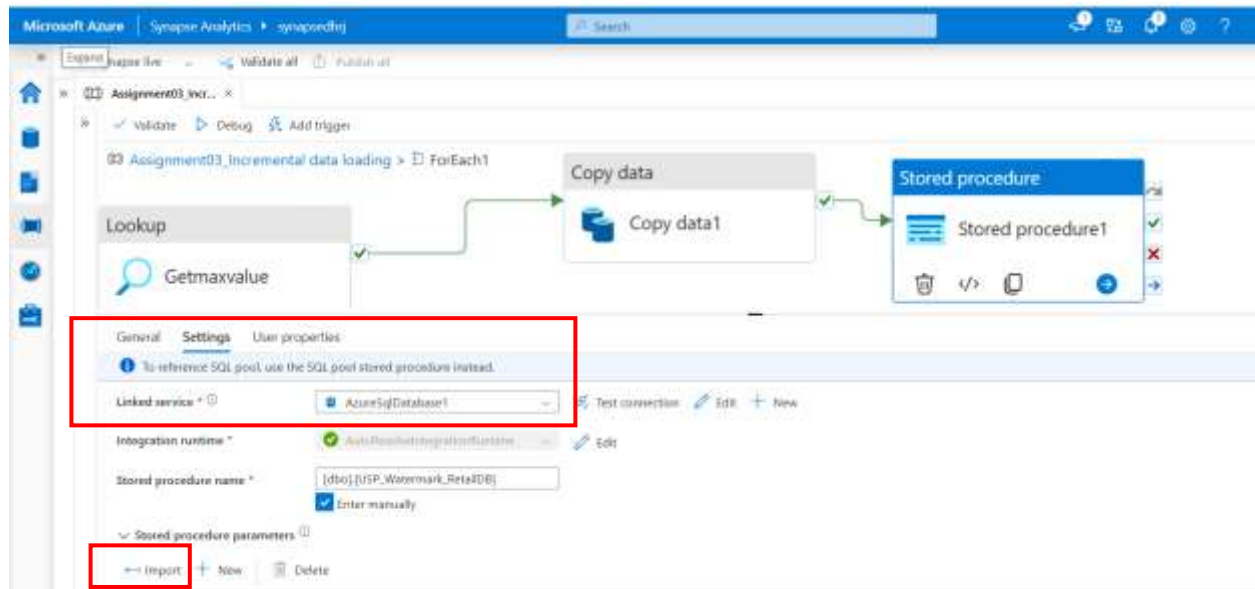
- ✓ Expression for the Foldername field would be `@(item().Foldername)`
- ✓ This expression will output the values with corresponding foldernames.



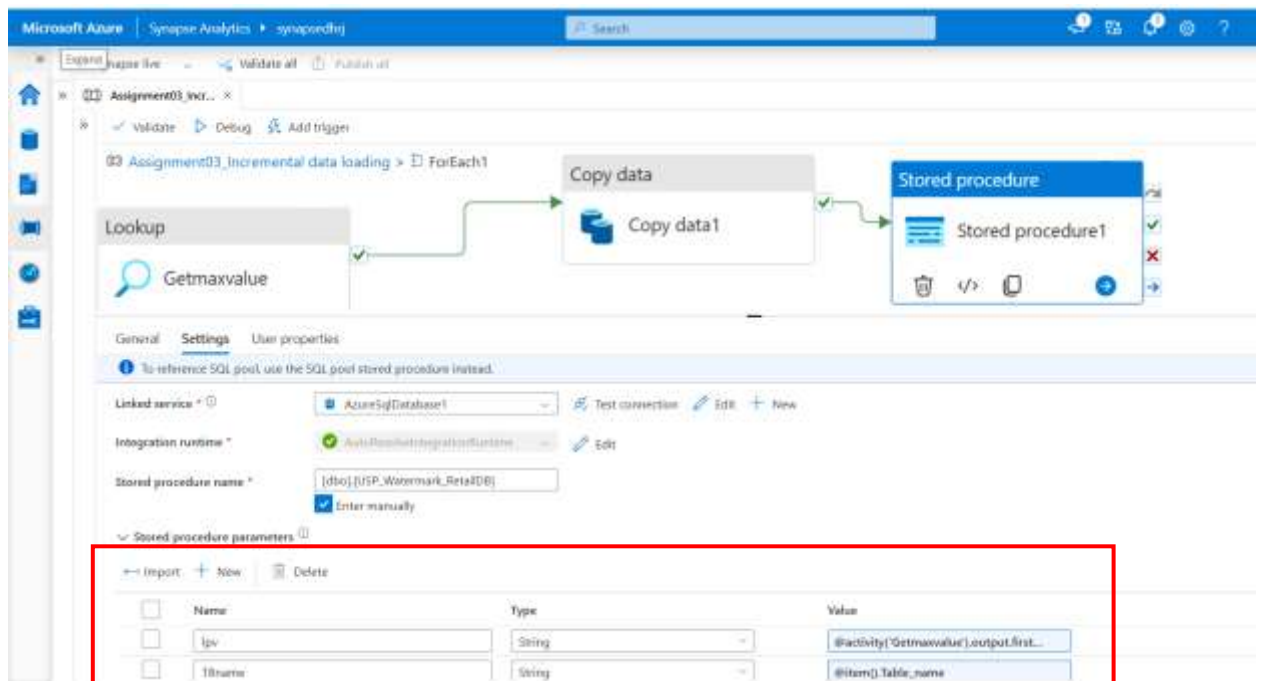
- ✓ Expression for the Foldername field would be `@(item().Foldername)` and `@concat(item().Table.name, '_', utcNow()), '.csv')`
- ✓ This expression will output the values as a .csv file format with current timestamp.



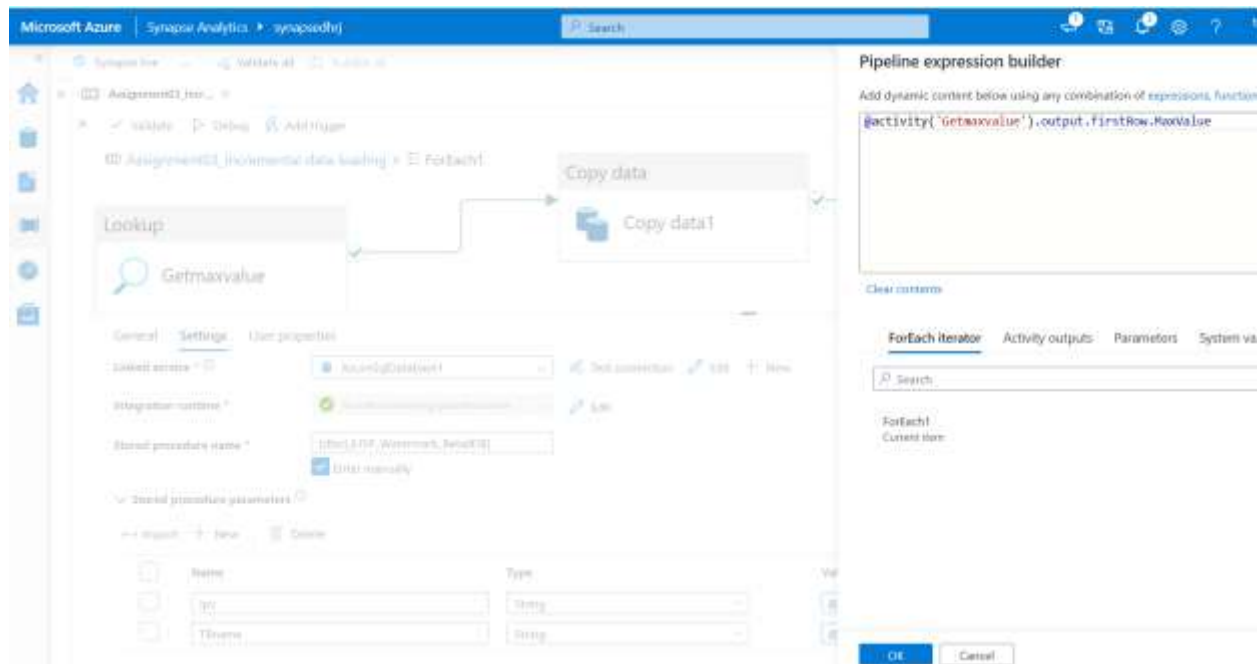
- ✓ Next Step - Adding a Stored procedure activity.
- ✓ Linking the source to the SQL database.



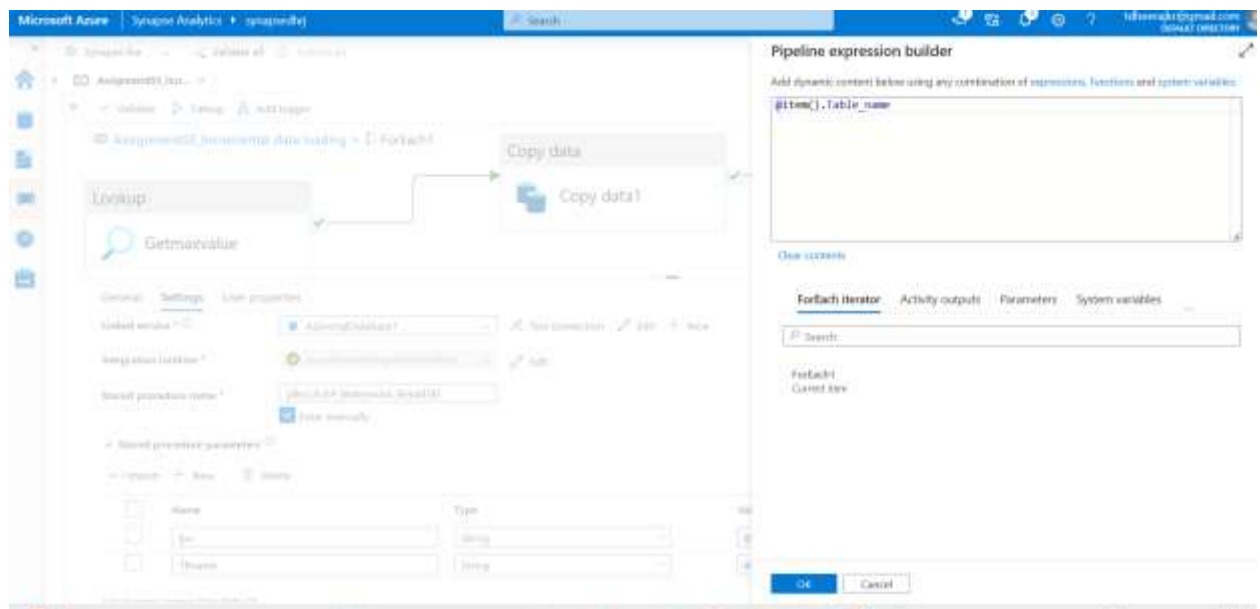
- ✓ Next click on Import parameters to import all the stored procedures created .



- ✓ Providing the dynamic expression for the lpv
@activity('Getmaxvalue').output.firstRow.Maxvalue
- ✓ This will only input the modified values from the dataset.



- ✓ Providing the dynamic expression for the TBname
@item().Table_name
- ✓ This will only input the modified values from the respective Table names.



- ✓ Publishing and checking the pipeline if its working

Microsoft Azure | Synapse Analytics | synapseedge | Search | 10/10/2023 10:00:00 AM

Synapse Edge | Validate all | /synapseedge

Assignment01_Intro... | Validate | Debug | Add trigger

Parameters | Variables | Settings | **Output**

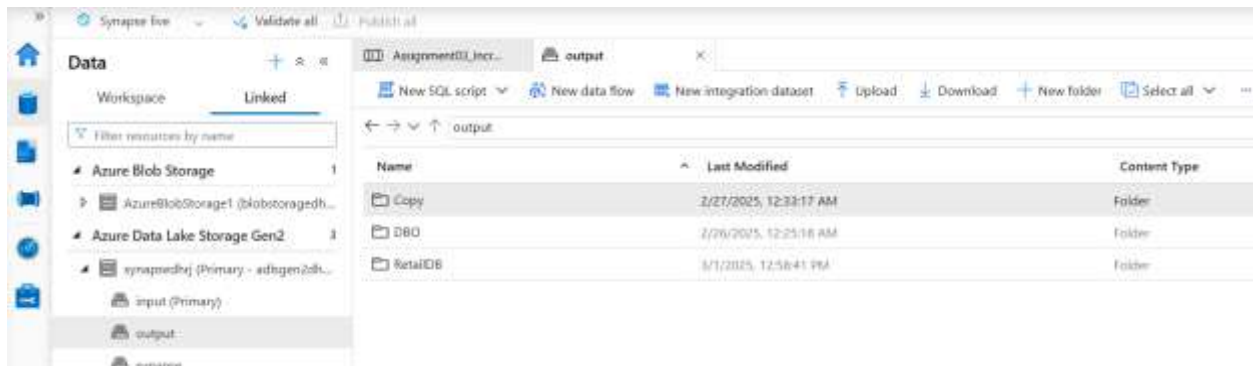
Pipeline run ID: 3b5e2666-44ac-4543-8530-3f6b798e264b | **Pipeline status:** Succeeded | View debug run consumption

All status: List | Showing 1 - 17 of 17 items

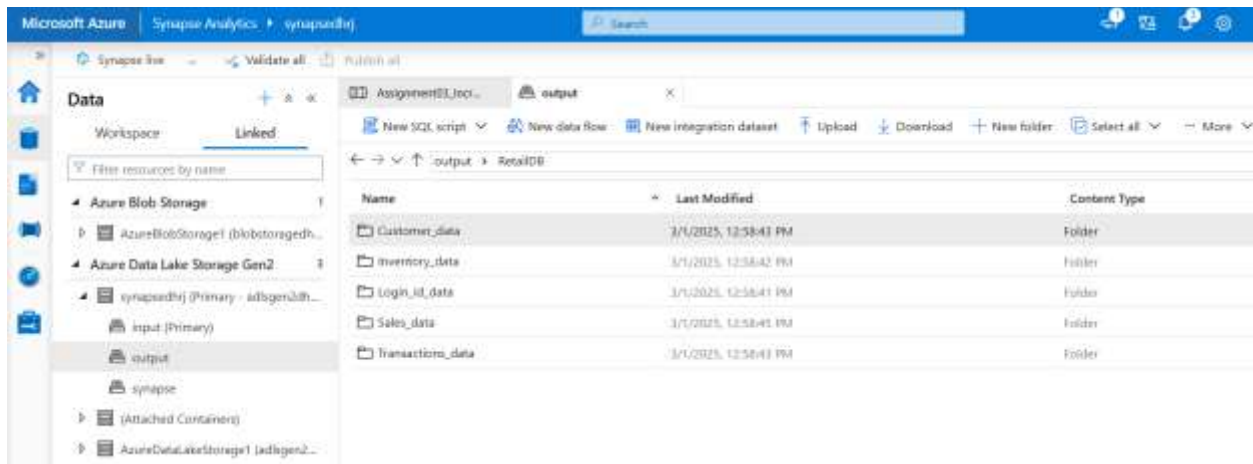
Activity name	Activity status	Activity name	Run start	Duration	Integration runtime	User prop.	Activity run ID
Stored procedure1	Succeeded	Stored procedure1	3/1/2023, 1:00:20 PM	0s	AzureDataLakeIntegrationRuntime (Australia East)		88c9473a-3461-4856-8389-c3526756c099
Stored procedure1	Succeeded	Stored procedure1	3/1/2023, 1:00:19 PM	3s	AzureDataLakeIntegrationRuntime (Australia East)		D8605164-582a-47ac-a438-ca4177175a49
Stored procedure1	Succeeded	Stored procedure1	3/1/2023, 1:00:18 PM	5s	AzureDataLakeIntegrationRuntime (Australia East)		4c2b028c-a004-4316-8125-31e52a3a3078
Stored procedure1	Succeeded	Stored procedure1	3/1/2023, 1:00:18 PM	6s	AzureDataLakeIntegrationRuntime (Australia East)		97fa4095-7c5c-4946-8d23-e479612640e0
Stored procedure1	Succeeded	Stored procedure1	3/1/2023, 1:00:17 PM	3s	AzureDataLakeIntegrationRuntime (Australia East)		ea35c152c-81a-4189-9206-2b6e41f9b3d1
Copy data1	Succeeded	Copy data	3/1/2023, 1:00:01 PM	17s	AzureDataLakeIntegrationRuntime (Australia East)		d850e02b-7800-4952-b502-82e7b5a05ea5
Copy data1	Succeeded	Copy data	3/1/2023, 1:00:00 PM	19s	AzureDataLakeIntegrationRuntime (Australia East)		3ada7229-b8c7-448b-8417-b2ac57094ada
Copy data1	Succeeded	Copy data	3/1/2023, 1:00:00 PM	19s	AzureDataLakeIntegrationRuntime (Australia East)		64d612a4-6887-4006-95c5-7375784788b0
Copy data1	Succeeded	Copy data	3/1/2023, 1:00:00 PM	19s	AzureDataLakeIntegrationRuntime (Australia East)		7e993e00-8c5e-46c5-9447-90e55c7a2a3c
Copy data1	Succeeded	Copy data	3/1/2023, 1:00:00 PM	17s	AzureDataLakeIntegrationRuntime (Australia East)		4704d0a1-0413-4e4a-b279-4231b67e1f3a
Getmaxvalue	Succeeded	Lookup	3/1/2023, 1:00:00 PM	7s	AzureDataLakeIntegrationRuntime (Australia East)		058d943-2453-4547-9420-1c5c0e9476d1
Getmaxvalue	Succeeded	Lookup	3/1/2023, 1:00:00 PM	0s	AzureDataLakeIntegrationRuntime (Australia East)		a4b43648-199a-463d-683e-44b08b0c076c

Checking the output folders...

- ✓ RetailDB folder is created



- ✓ As mentioned in the Watermark table all the corresponding folders have been created i.e. Customer_data folder, Inventory_data folder, Login_id_data folder, Sales_data folder and Transactions_data folder.



- ✓ Cross checking the folder names in the SSMS.
- ✓ Checking in SSMS whether the name are matching with the Foldername

Assignment 03.sql - sqlserver-dheeraj.database.windows.net:sqlserver-dheeraj (Server (711)) - Microsoft SQL Server Management Studio

```
85 CREATE TABLE Watermark (
86     ID INT PRIMARY KEY,
87     Table_name VARCHAR(100),
88     Schema_name VARCHAR(100),
89     Foldername VARCHAR(50),
90     Delta_column VARCHAR(100),
91     Last_processed_value VARCHAR(255) NOT NULL.
92 );
93
94
95 SELECT * FROM Watermark
96
97 INSERT INTO Watermark VALUES
98 (1, 'Customer', 'dbo', 'RetailDB/Customer_data', 'Customerupdateddate', '1900-01-01 00:00:00'),
99 (2, 'Login_id', 'dbo', 'RetailDB/Login_id_data', 'Updatedlogindata', '1900-01-01 00:00:00'),
100 (3, 'Inventory', 'dbo', 'RetailDB/Inventory_data', 'ProductID', '6'),
101 (4, 'Transactions', 'dbo', 'RetailDB/Transactions_data', 'TransactionID', '3'),
102 (5, 'Sales', 'dbo', 'RetailDB/Sales_data', 'Sales_log', '1900-01-01 00:00:00')
```

100 %

Results Messages

ID	Table_name	Schema_name	Foldername	Delta_column	Last_processed_value
1	Customer	dbo	RetailDB/Customer_data	Customerupdateddate	2023-06-01T10:00:00
2	Login_id	dbo	RetailDB/Login_id_data	Updatedlogindata	2023-03-01T09:15:00
3	Inventory	dbo	RetailDB/Inventory_data	ProductID	6
4	Transactions	dbo	RetailDB/Transactions_data	TransactionID	3
5	Sales	dbo	RetailDB/Sales_data	Sales_log	2023-06-15T10:00:00

- ✓ **Checking for the Incremental load**
- ✓ Adding more values into Customer, Inventory and Sales tables and publish in Synapse to check whether the incremental load is working or not ?

Assignment 03.sql - ...heeraj (Server (71))*

```

124 END
125
126
127 INSERT INTO Customer (CustomerID, Name, Phone, Customerupdateddate)
128 VALUES
129 (4, 'Alice Brown', '555-123-4567', '2023-04-01 12:00:00'),
130 (5, 'Charlie Davis', '555-789-0123', '2023-05-01 11:00:00'),
131 (6, 'Emily Wilson', NULL, '2023-06-01 10:00:00');
132
133 INSERT INTO Inventory (ProductID, ProductName, Quantity, Price)
134 VALUES
135 (4, 'Tablet', 30, 299.99),
136 (5, 'Smartwatch', 40, 199.99),
137 (6, 'Earbuds', 60, 89.99);
138
139
140 INSERT INTO Sales (SalesID, ProductID, Sales_log, Revenue)
141 VALUES
142 (4, 4, '2023-04-05 12:00:00', 299.99),
143 (5, 5, '2023-05-10 11:00:00', 199.99),
144 (6, 6, '2023-06-15 10:00:00', 89.99);
145

```

108 %

Results Messages

	ID	Table_name	Schema_name	Foldername	Delta_column	Last_processed_value
1	1	Customer	dbo	RetailDB/Customer_data	Customerupdateddate	2023-06-01T10:00:00
2	2	Login_id	dbo	RetailDB/Login_id_data	Updatedlogindata	2023-03-01T09:15:00
3	3	Inventory	dbo	RetailDB/Inventory_data	ProductID	6
4	4	Transactions	dbo	RetailDB/Transactions_data	TransactionID	3
5	5	Sales	dbo	RetailDB/Sales_data	Sales_log	2023-06-15T10:00:00

- ✓ Inventory table

The screenshot displays the AWS IAM console's 'Groups' page. On the left, a list of groups is shown, with 'Inventory_2025-03-21T18:09:06:3575817Z.csv' selected. The right-hand pane provides details for this selected group, including its path, size, and a table of permissions.

Inventory_2025-03-21T18:09:06:3575817Z.csv

Path: /aws-logs-2025-03-21T18:09:06:3575817Z.csv

Size: 10,000,000 bytes


Created: 2025-03-21T18:09:06:3575817Z

Modified: 2025-03-21T18:09:06:3575817Z

With column Header: ☒

PERMISSION	PERMISSION	QUANTITY	PRICE
1	Basic	10	100.00
2	Intermediate	10	100.00
3	Advanced	10	100.00

- ✓ Customer table



The screenshot shows the Microsoft Azure portal interface for a Customer Data Lake Storage (CDL) account. The left sidebar displays the navigation menu with 'Storage' selected. The main content area shows the 'Properties' tab for the account 'Customer_2025-03-01T18:08:58.361Z4982.csl'. The account is located in 'East US' and is 'Online'. A table lists the account's properties:

PROPERTY	VALUE	STATUS
NAME	Customer_2025-03-01T18:08:58.361Z4982.csl	OK
LOCATION	East US	OK
STATUS	Online	OK

- ✓ Sales table

[illegible]