DATA LOADING WITH INCREMENTAL LOADING

Contents

Schema Definition

For this project, we are creating five tables: **Customer**, **Employee**, **Product**, **Orders**, and **Sales**. We will select **integer** and **datetime** columns as the **DeltaColumn** for incremental loading to efficiently track and load new or updated data.

Below are the queries to create the tables:

Customer Table

```
CREATE TABLE Customer (
  Customer_ID INT PRIMARY KEY,-- Delta Column
  Name VARCHAR(255),
  Email VARCHAR(255),
  Phone VARCHAR(20),
  Address VARCHAR(500)
);
```

Employee Table

```
CREATE TABLE Employee (
  Employee_ID INT PRIMARY KEY,
  Name VARCHAR(255),
  Position VARCHAR(100),
  Department VARCHAR(100),
  Salary DECIMAL(10,2),
  Updated_At DATETIME -- Delta Column
);
```

Product Table

```
CREATE TABLE Product (
  Product_ID INT PRIMARY KEY, -- Delta Column
  Product_Name VARCHAR(255),
  Category VARCHAR(100),
  Price DECIMAL(10,2),
  Stock INT
);
```

Orders Table

```
CREATE TABLE Orders (
  Order_ID INT PRIMARY KEY,
  Customer_ID INT FOREIGN KEY REFERENCES Customer(Customer_ID),
  Order_Date DATETIME, -- Delta Column
  Total_Amount DECIMAL(10,2),
  Status VARCHAR(50)
);
```

Sales Table

```
CREATE TABLE Sales (
  Sale_ID INT IDENTITY(1,1) PRIMARY KEY,
  Product_ID INT FOREIGN KEY REFERENCES Product(Product_ID),
  Customer_ID INT FOREIGN KEY REFERENCES Customer(Customer_ID),
  Transaction_Date DATETIME, -- Delta Column
  Quantity INT,
  Total_Amount DECIMAL(10,2)
);
```

Data Insertion:

Insert Data into Customer Table

```
INSERT INTO Customer (Customer_ID, Name, Email, Phone, Address) VALUES
```

```
(1, 'John Doe', 'johndoe@email.com', '123-456-7890', '123 Main St, NY'),
(2, 'Jane Smith', 'janesmith@email.com', '987-654-3210', '456 Oak St, CA'),
(3, 'Robert Brown', 'robert@email.com', '456-789-1234', '789 Pine St, TX'),
(4, 'Emily Johnson', 'emily@email.com', '321-654-9870', '321 Maple St, FL'),
(5, 'Michael Lee', 'michael@email.com', '654-321-4567', '654 Cedar St, WA');
```

Insert Data into Product Table

```
INSERT INTO Product (Product_ID, Product_Name, Category, Price, Stock) VALUES
(101, 'Laptop', 'Electronics', 999.99, 50),
(102, 'Smartphone', 'Electronics', 599.99, 100),
(103, 'Headphones', 'Accessories', 79.99, 200),
(104, 'Desk Chair', 'Furniture', 129.99, 30),
(105, 'Running Shoes', 'Apparel', 89.99, 150);
```

Insert Data into Orders Table (References Customer)

```
INSERT INTO Orders (Order_ID, Customer_ID, Order_Date, Total_Amount, Status) VALUES
(1001, 1, '2025-02-01 10:30:00', 1099.99, 'Shipped'),
(1002, 2, '2025-02-03 15:45:00', 629.99, 'Processing'),
(1003, 3, '2025-02-05 08:20:00', 79.99, 'Delivered'),
(1004, 4, '2025-02-07 18:10:00', 249.99, 'Cancelled'),
(1005, 5, '2025-02-10 12:50:00', 129.99, 'Shipped');
```

Insert Data into Employee Table

```
INSERT INTO Employee (Employee_ID, Name, Position, Department, Salary, Updated_At)
VALUES
(201, 'Alice Williams', 'Data Analyst', 'IT', 75000, '2025-02-01 09:00:00'),
(202, 'Bob Harris', 'Software Engineer', 'IT', 90000, '2025-02-02 10:15:00'),
(203, 'Charlie Young', 'HR Manager', 'HR', 85000, '2025-02-04 14:30:00'),
(204, 'Diana Scott', 'Marketing Lead', 'Marketing', 78000, '2025-02-06 16:45:00'),
(205, 'Ethan Clark', 'Finance Manager', 'Finance', 95000, '2025-02-08 12:00:00');
```

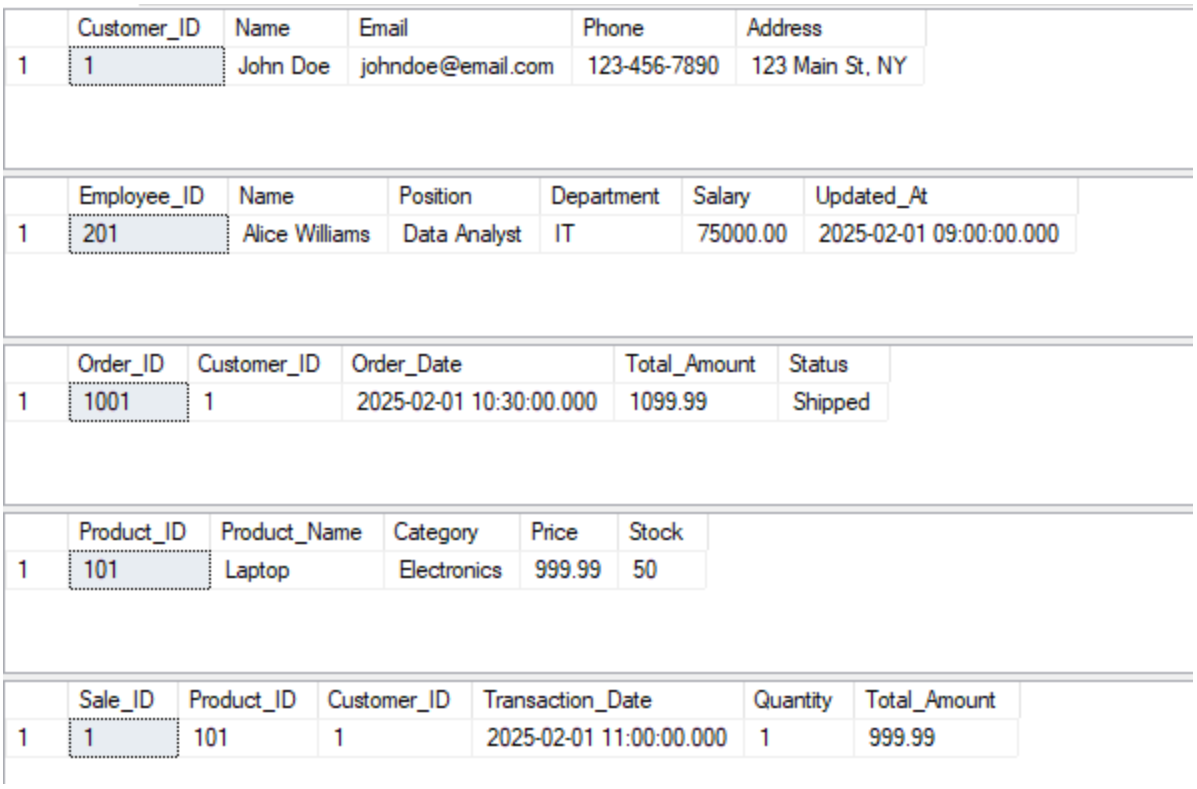
Insert Data into Sales Table (References Product & Customer)

```
INSERT INTO Sales (Product_ID, Customer_ID, Transaction_Date, Quantity, Total_Amount)
VALUES
(101, 1, '2025-02-01 11:00:00', 1, 999.99),
```

```
(102, 2, '2025-02-03 16:00:00', 1, 599.99),
(103, 3, '2025-02-05 08:30:00', 2, 159.98),
(104, 4, '2025-02-07 18:20:00', 1, 129.99),
(105, 5, '2025-02-10 13:00:00', 2, 179.98);
```

Output

```
Select top 1 * from Customer;
Select top 1 * from Employee;
Select top 1 * from Orders;
Select top 1 * from Product;
Select top 1 * from Sales;
```



	Customer_ID	Name	Email	Phone	Address
1	1	John Doe	johndoe@email.com	123-456-7890	123 Main St, NY

	Employee_ID	Name	Position	Department	Salary	Updated_At
1	201	Alice Williams	Data Analyst	IT	75000.00	2025-02-01 09:00:00.000

	Order_ID	Customer_ID	Order_Date	Total_Amount	Status
1	1001	1	2025-02-01 10:30:00.000	1099.99	Shipped

	Product_ID	Product_Name	Category	Price	Stock
1	101	Laptop	Electronics	999.99	50

	Sale_ID	Product_ID	Customer_ID	Transaction_Date	Quantity	Total_Amount
1	1	101	1	2025-02-01 11:00:00.000	1	999.99

Watermark Table

The **Watermark Table** will serve as a metadata table to manage information related to incremental data loads:

- **SchemaName**: Stores the schema name for the tables (e.g., "dbo" for all tables in our case).

- **TableName**: Stores the names of the tables (e.g., "Customer", "Sales").
- **FolderName**: Defines the file path where the data will be stored during the sink process, helping to direct the data to the appropriate location.
- **LPV (Last Processed Value)**: Stores the maximum value from the last pipeline run, which helps track which records have been processed so the pipeline can continue from the right point.
- **DeltaColumn**: Indicates which column in the table should be used for incremental loading (e.g., the column that tracks changes or new records).

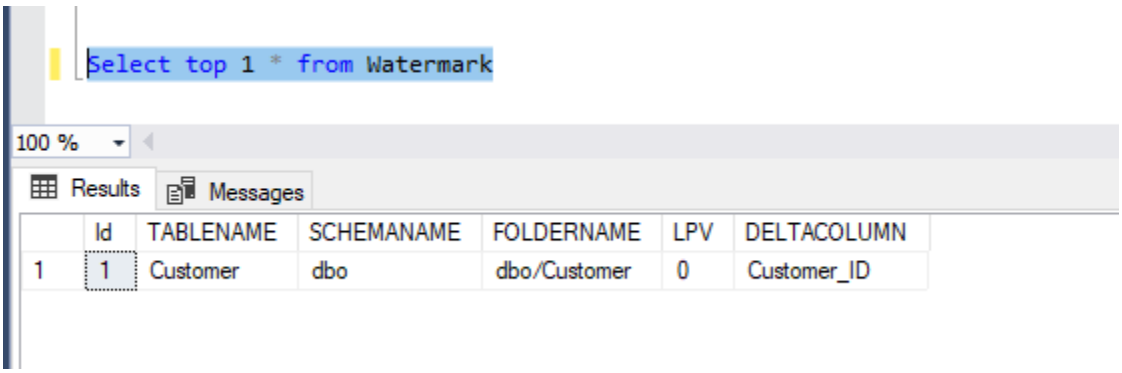
Watermark Table schema query:

```
Create table Watermark(
Id int,
TABLENAME varchar(50),
SCHEMANAME varchar(30),
FOLDERNAME varchar(100),
LPV varchar(100),
DELTACOLUMN varchar(100)
)
```

Insert Values into Watermark Table:

```
Insert into Watermark Values
(1,'Customer','dbo','dbo/Customer','0','Customer_ID'),
(2,'Product','dbo','dbo/Product','0','Product_ID'),
(3,'Employee','dbo','dbo/Employee','2000-01-01 00:00:00','Updated_At'),
(4,'Orders','dbo','dbo/Orders','2000-01-01 00:00:00','Order_Date'),
(5,'Sales','dbo','dbo/Sales','2000-01-01 00:00:00','Transaction_Date')
```

Output:



Update Watermark (LPV)

After creating the Watermark table, we need a stored procedure to update the **LPV (Last Processed Value)** with the latest processed max value from the tables. This procedure will be executed at the **end of the pipeline**.

Stored Procedure Schema Definition:

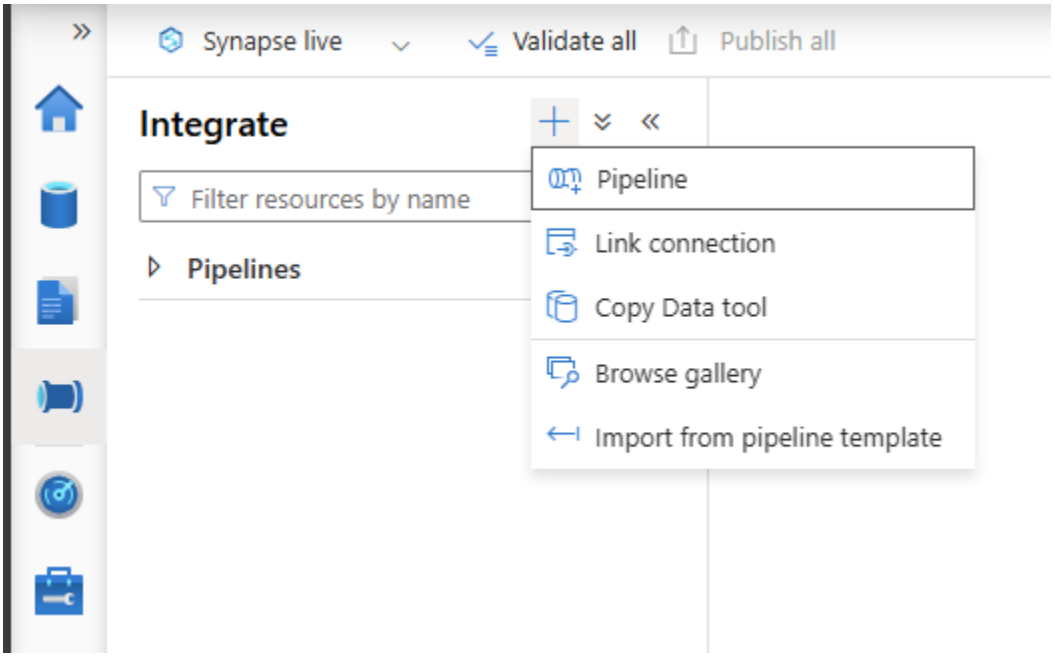
```
CREATE Proc upd_watermark
@TABLENAME varchar(100),
@LPV varchar(100)
AS

UPDATE Watermark
Set LPV=@LPV
Where TABLENAME=@TABLENAME
GO
```

This stored procedure will update the **LPV** column in the **Watermark** table with the latest value based on the **TABLENAME** provided.

Pipeline

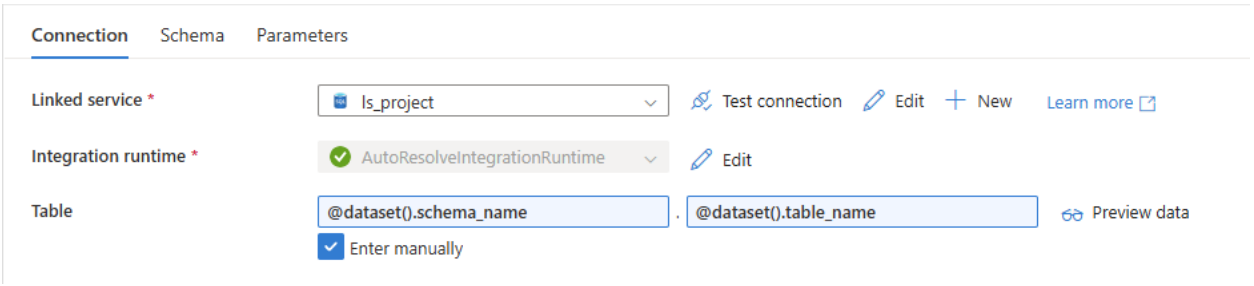
Now, that we have created the tables, watermark table and stored procedure, it's time to create a pipeline in synapse.

Step 1: Open synapse workspace and click on create a new pipeline

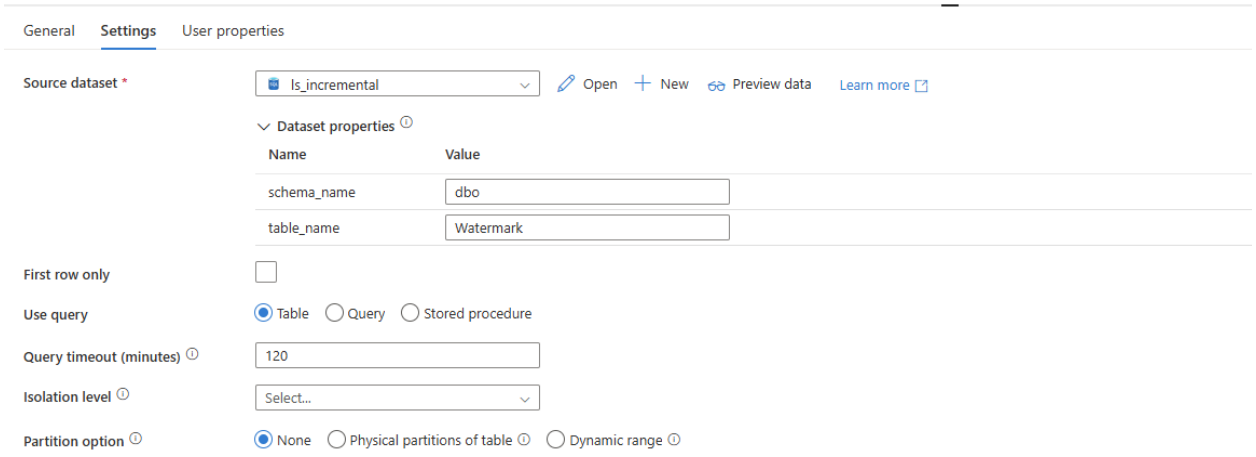


Step 2: Drag Lookup Activity from activities

1. Give name to your lookup activity
2. Create a Azure sql database dataset and use linked service to connect to the database that have the tables and watermark.
3. It will store the information in form of array.
4. Open Dataset and create two parameters schema_name and table_name. Pass the parameter in the connection tab.

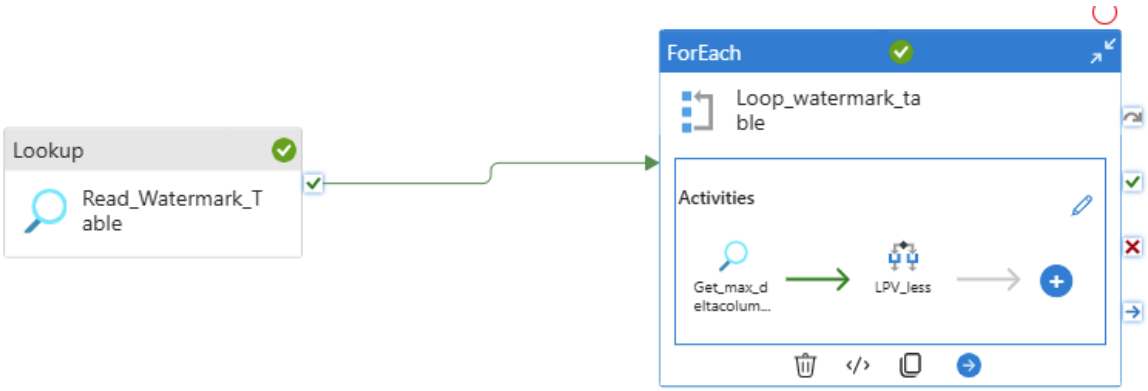


5. Under settings, enter the schem_name and table_name to read Watermark Table.

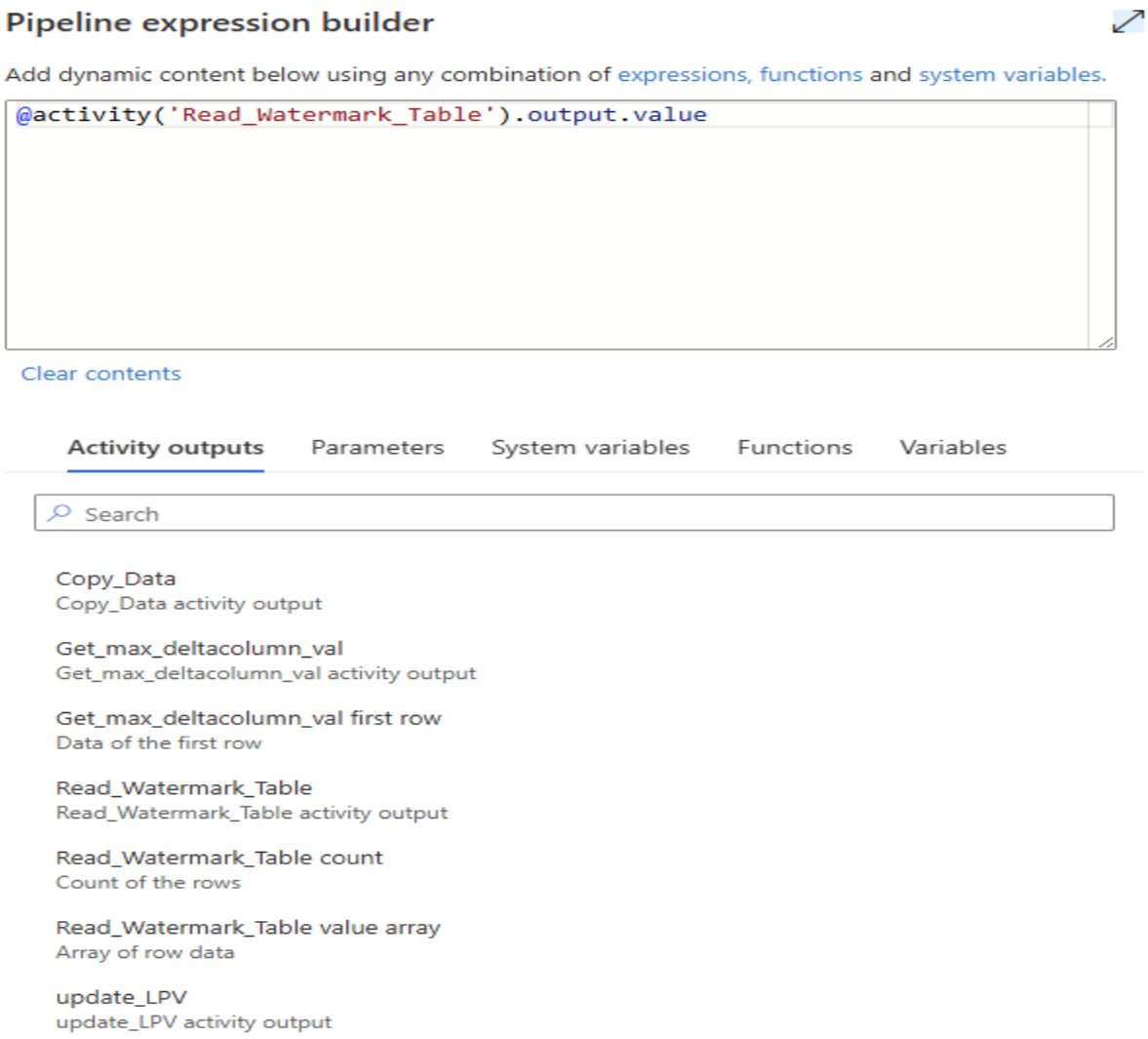


Step 3: Drag Foreach activity

1. Drag the foreach activity and give it a name.
2. We will iterate through the array one by one.
3. Connect the lookup success node with it.

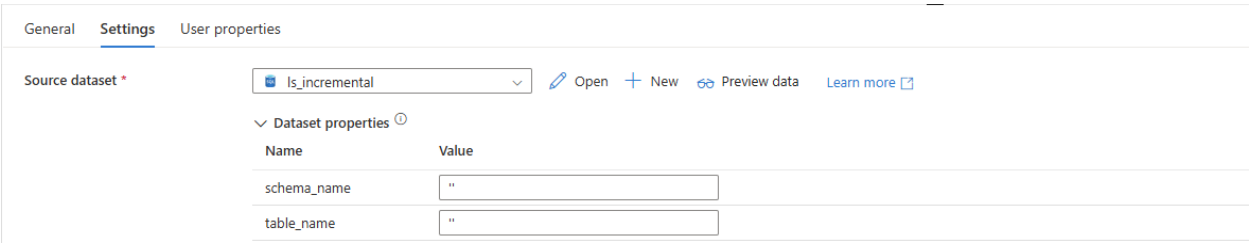


4. Under settings, in items click on dynamic content. Under Activity outputs, click on Read watermark table array value.



Step 4: Drag new lookup under Foreach activity

1. In the foreach activity, drag lookup and give name like max_value.
2. Click on same Azure sql database dataset that we created during the first lookup.
3. In schemaname and tablename parameter, enter empty string ‘ ‘ because we want to use query to get the max value from our deltacolumns.
4. Click on First row only



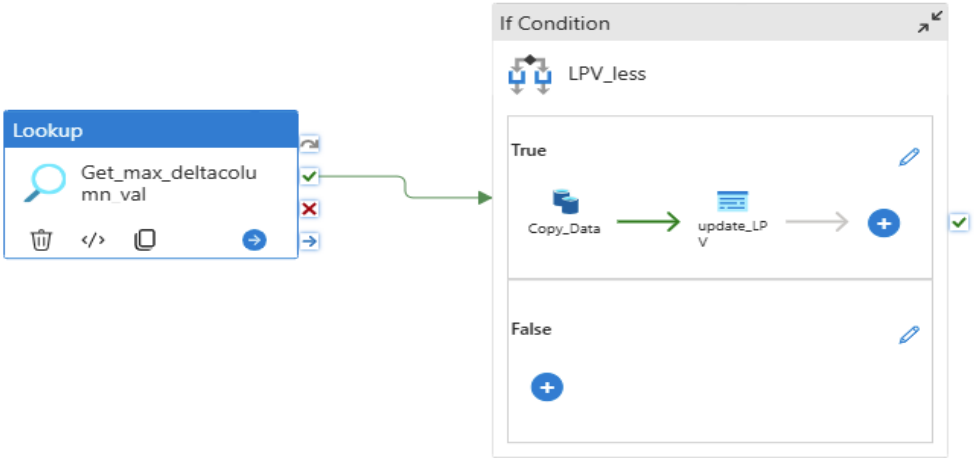
5. Click on the query, and write the below expression to get max value that we will check with LPV and based on it copy the data.

   Select max(@{item().DELTACOLUMN}) as maxvalue
   from @{item().SCHEMANAME}.@{item().TABLENAME}

   Here, we are using the foreach activity output to get the Deltacolumn, Tablename information for each iteration using @item() function.

Step 5: Use IF Condition

1. Drag a if condition activity and connect lookup success node with it.



2. Pass empty string ‘ ‘ in schemaname and tablename.
3. Now, We will check if LPV is less than max value Deltacolumn, then only copy data else skip. This will also prevent creation of empty files.
4. Click on query and use less function that takes two values and check if val 1 is less than val 2 , then it returns true.

   ```
   Select max(@{item().DELTACOLUMN}) as maxvalue
   ```

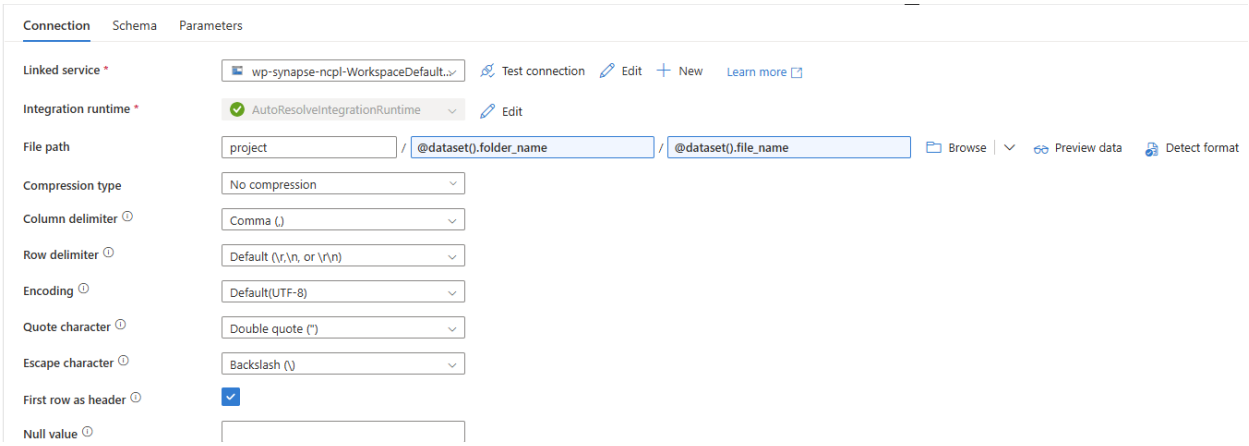
```
from @{item().SCHEMANAME}.@{item().TABLENAME}
```

Step 6: In the true activity of If condition, drag copy activity

1. Drag copy activity and give it a name.
2. In the source settings, use the azure sql database dataset.
3. Pass empty string ‘ ‘ in schemaname and tablename parameter.
4. Under query, write dynamic expression to copy the data where Deltacolumn values is greater than the LPV.

```
Select * from @{item().TABLENAME}
WHERE @{item().DELTACOLUMN}>'@{item().LPV}'
```

5. In the sink settings, create a new ADLS dataset and choose the container where you want to store the csv files.
6. Click on open dataset and create two parameters filename and foldername, and pass the parameter in the filepath in connection tab.

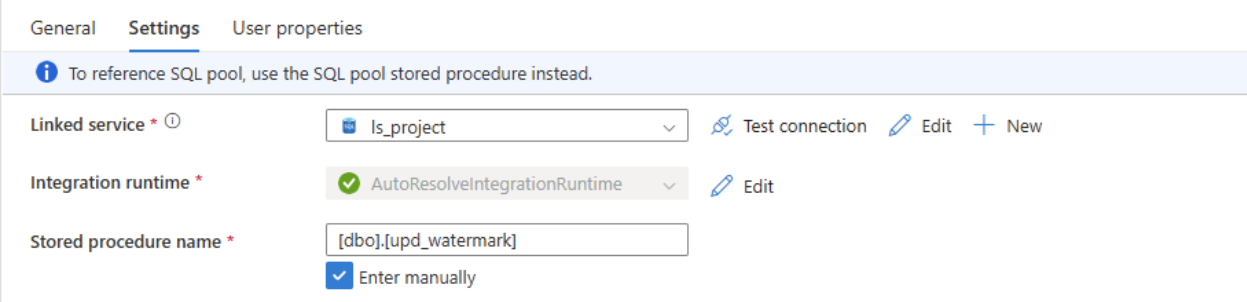


7. In the sink settings, under parameters pass the value from foreach activity in folderpath.
8. In filename parameter, we are using concat function to merge tablename,the datetime at which the file was processed.

```
@concat(item().TABLENAME,'_',utcNow(),'.csv')
```

Step 7: Drag stored procedure activity

1. Drag store procedure. This will update the LPV after each pipeline run.
2. Use the Azure sql database dataset to connect to the stored procedure.
3. Select the stored procedure from dropdown.

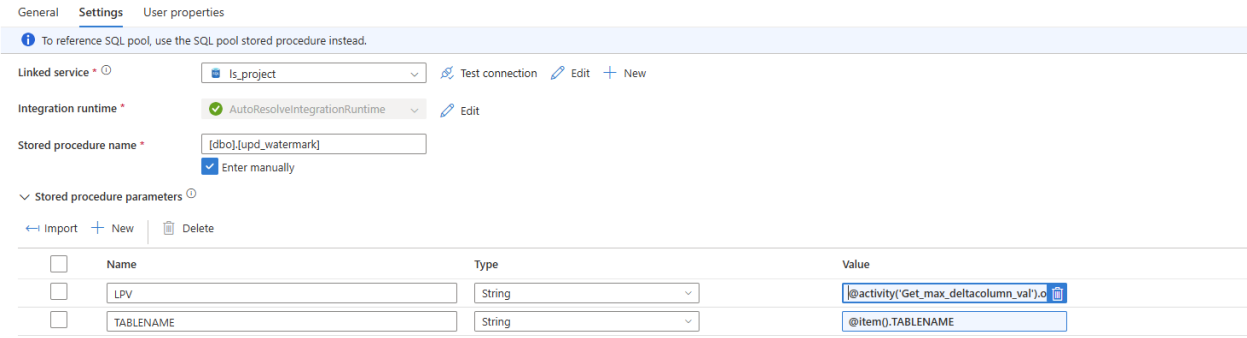


4. Click on import parameters under stored procedure parameter.
5. In LPV parameter, go to activity output and click on max delta Value first row which will fetch the maxvalue from the lookup we created.

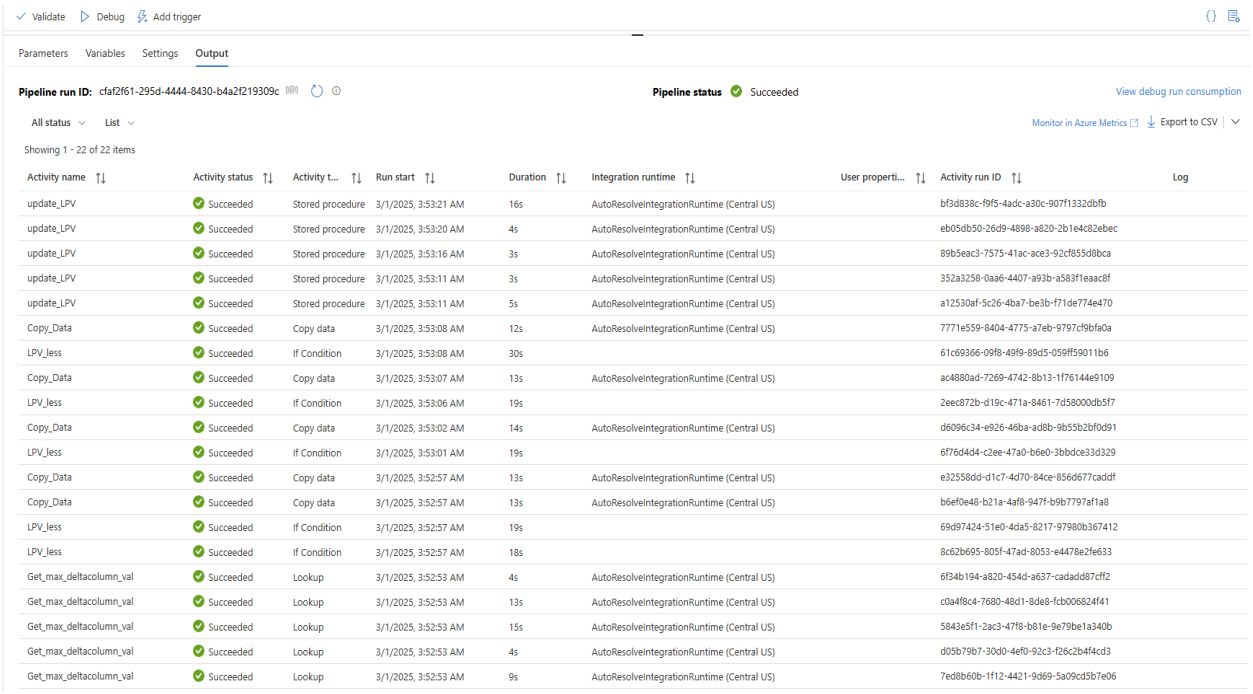
```
@activity('Get_max_deltacolumn_val').output.firstRow.maxvalue
```

6. In Tablename parameter, pass the tablename using foreach reference.

```
@item().TABLENAME
```

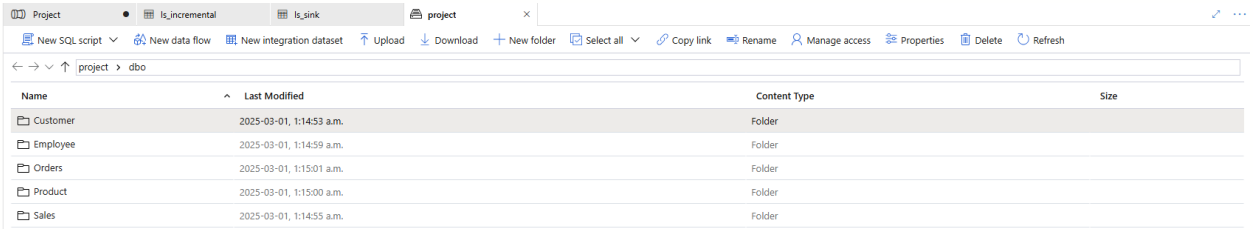


Step 8: Click on debug pipeline.

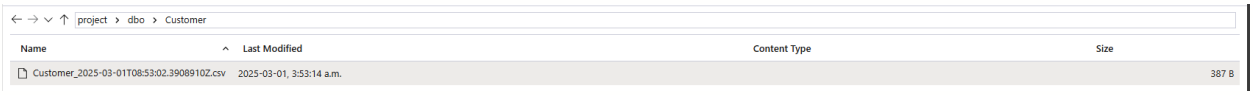


Activity name	Activity status	Activity t...	Run start	Duration	Integration runtime	User properti...	Activity run ID	Log
update_LPV	Succeeded	Stored procedure	3/1/2025, 3:53:21 AM	16s	AutoResolveIntegrationRuntime (Central US)		bf3d838c-f9f5-4adc-a30c-907f1332dbfb	
update_LPV	Succeeded	Stored procedure	3/1/2025, 3:53:20 AM	4s	AutoResolveIntegrationRuntime (Central US)		eb05db50-26d9-4898-a820-2b1e4c82ebec	
update_LPV	Succeeded	Stored procedure	3/1/2025, 3:53:16 AM	3s	AutoResolveIntegrationRuntime (Central US)		89b5eac3-7575-41ac-ace3-92cf855d8bca	
update_LPV	Succeeded	Stored procedure	3/1/2025, 3:53:11 AM	3s	AutoResolveIntegrationRuntime (Central US)		352a3258-0aa6-4407-a93b-a583f1eaac8f	
update_LPV	Succeeded	Stored procedure	3/1/2025, 3:53:11 AM	5s	AutoResolveIntegrationRuntime (Central US)		a12530af-5c26-4ba7-be3b-f71de774e470	
Copy_Data	Succeeded	Copy data	3/1/2025, 3:53:08 AM	12s	AutoResolveIntegrationRuntime (Central US)		7771e559-8404-4775-a7eb-9797cf9bfa0a	
LPV_less	Succeeded	If Condition	3/1/2025, 3:53:08 AM	30s			61c69366-09f8-49f9-89d5-059ff59011b6	
Copy_Data	Succeeded	Copy data	3/1/2025, 3:53:07 AM	13s	AutoResolveIntegrationRuntime (Central US)		ac4880ad-7269-4742-8b13-1f76144e9109	
LPV_less	Succeeded	If Condition	3/1/2025, 3:53:06 AM	19s			2eec872b-d19c-471a-8461-7d58000db5f7	
Copy_Data	Succeeded	Copy data	3/1/2025, 3:53:02 AM	14s	AutoResolveIntegrationRuntime (Central US)		d6096c34-e926-46ba-ad8b-9b55b2bf0d91	
LPV_less	Succeeded	If Condition	3/1/2025, 3:53:01 AM	19s			6f76d4d4-c2ee-47a0-b6e0-3bbdce33d329	
Copy_Data	Succeeded	Copy data	3/1/2025, 3:52:57 AM	13s	AutoResolveIntegrationRuntime (Central US)		e32558dd-d1c7-4d70-84ce-856d677caddf	
Copy_Data	Succeeded	Copy data	3/1/2025, 3:52:57 AM	13s	AutoResolveIntegrationRuntime (Central US)		b6ef0e48-b21a-4af8-947f-b9b7797af1a8	
LPV_less	Succeeded	If Condition	3/1/2025, 3:52:57 AM	19s			69d97424-51e0-4da5-8217-97980b367412	
LPV_less	Succeeded	If Condition	3/1/2025, 3:52:57 AM	18s			8c62b695-805f-47ad-8053-e4478e2fe633	
Get_max_deltacolumn_val	Succeeded	Lookup	3/1/2025, 3:52:53 AM	4s	AutoResolveIntegrationRuntime (Central US)		6f34b194-a820-454d-a637-cadadd87cff2	
Get_max_deltacolumn_val	Succeeded	Lookup	3/1/2025, 3:52:53 AM	13s	AutoResolveIntegrationRuntime (Central US)		c0a4f8c4-7680-48d1-8de8-fcb006824f41	
Get_max_deltacolumn_val	Succeeded	Lookup	3/1/2025, 3:52:53 AM	15s	AutoResolveIntegrationRuntime (Central US)		5843e5f1-2ac3-47f8-b81e-9e79be1a340b	
Get_max_deltacolumn_val	Succeeded	Lookup	3/1/2025, 3:52:53 AM	4s	AutoResolveIntegrationRuntime (Central US)		d05b79b7-30d0-4ef0-92c3-f26c2b4f4cd3	
Get_max_deltacolumn_val	Succeeded	Lookup	3/1/2025, 3:52:53 AM	9s	AutoResolveIntegrationRuntime (Central US)		7ed8b60b-1f12-4421-9d69-5a09cd5b7e06	

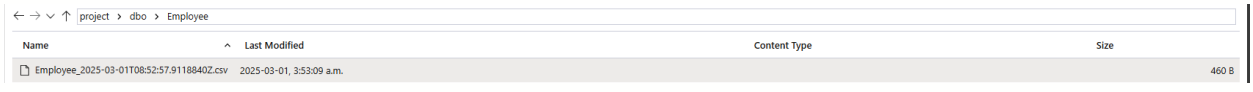
Step 9: Go the path and check for folder and files.



Name	Last Modified	Content Type	Size
Customer	2025-03-01, 1:14:53 a.m.	Folder	
Employee	2025-03-01, 1:14:59 a.m.	Folder	
Orders	2025-03-01, 1:15:01 a.m.	Folder	
Product	2025-03-01, 1:15:00 a.m.	Folder	
Sales	2025-03-01, 1:14:55 a.m.	Folder	



Name	Last Modified	Content Type	Size
Customer_2025-03-01T08:53:02.3908910Z.csv	2025-03-01, 3:53:14 a.m.		387 B



Name	Last Modified	Content Type	Size
Employee_2025-03-01T08:52:57.9118840Z.csv	2025-03-01, 3:53:09 a.m.		460 B

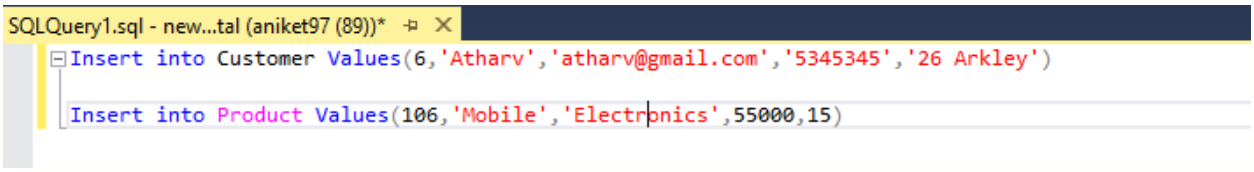
Name	Last Modified	Content Type	Size
Orders_2025-03-01T08:53:08.5397127Z.csv	2025-03-01, 3:53:19 a.m.		325 B

Name	Last Modified	Content Type	Size
Product_2025-03-01T08:52:57.7741093Z.csv	2025-03-01, 3:53:09 a.m.		250 B

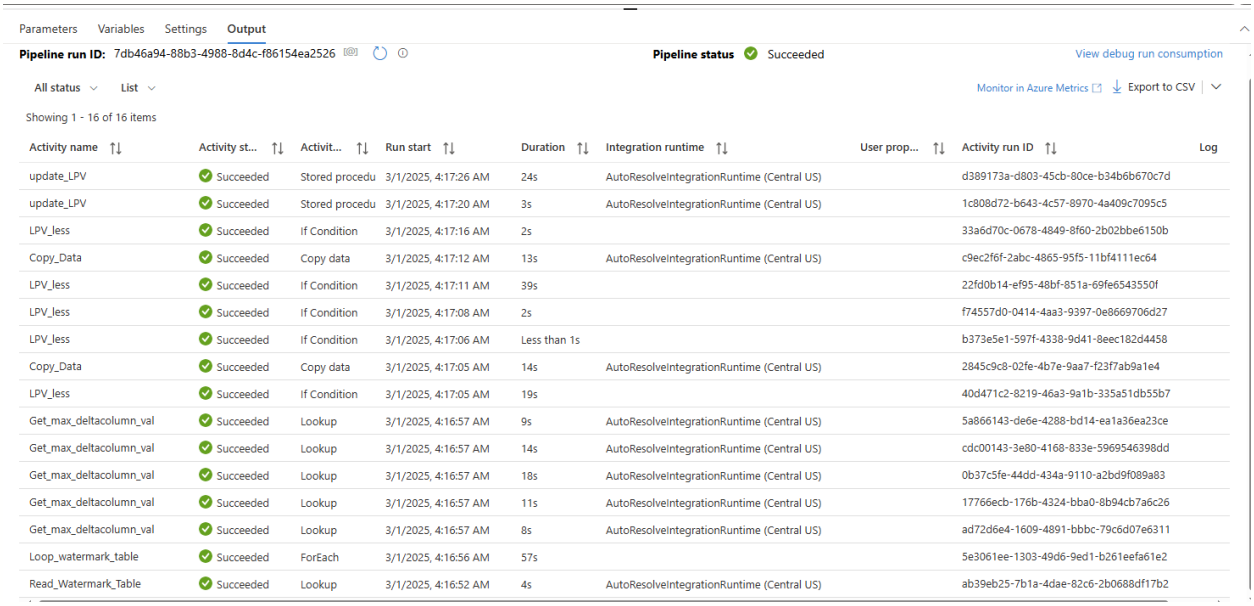
Name	Last Modified	Content Type	Size
Sales_2025-03-01T08:53:07.1933083Z.csv	2025-03-01, 3:53:18 a.m.		301 B

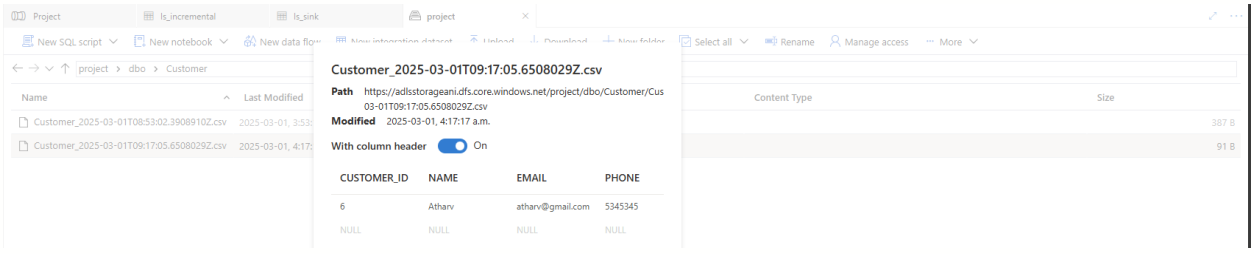
Step 10: Debug pipeline with testcases

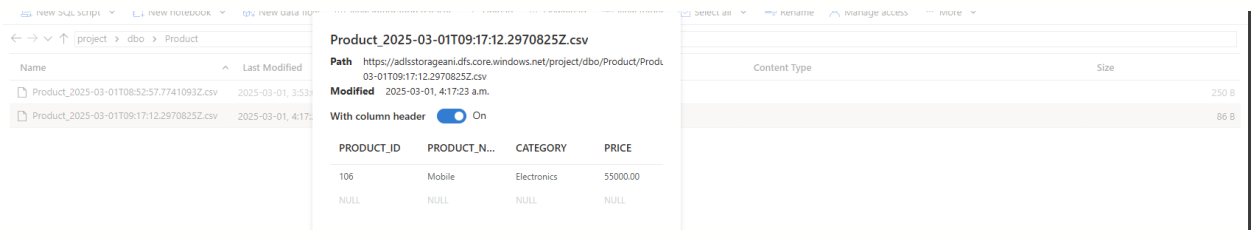
1. We will insert values into two tables.



2. Debug the pipeline again, and customer and product should have a new csv file. Also, check the watermark table if the max value got updated in LPV.







```
Select * from Watermark
```

100 %

Results | Messages

	Id	TABLENAME	SCHEMANAME	FOLDERNAME	LPV	DELTACOLUMN
1	1	Customer	dbo	dbo/Customer	6	Customer_ID
2	2	Product	dbo	dbo/Product	106	Product_ID
3	3	Employee	dbo	dbo/Employee	2025-02-08T12:00:00	Updated_At
4	4	Orders	dbo	dbo/Orders	2025-02-10T12:50:00	Order_Date
5	5	Sales	dbo	dbo/Sales	2025-02-10T13:00:00	Transaction_Date

As we can see, new files were created as well as Watermark table got updated with max deltacolumn values.