DATA LOADING Incremental and Full Load using Switch

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. Also, we have a Table name **full_laod** which we want to load at every run and overwrite the file.

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)
);
```

Full_load Table

```
Create table full_load(
id int,
name varchar(100)
)
```

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');
```

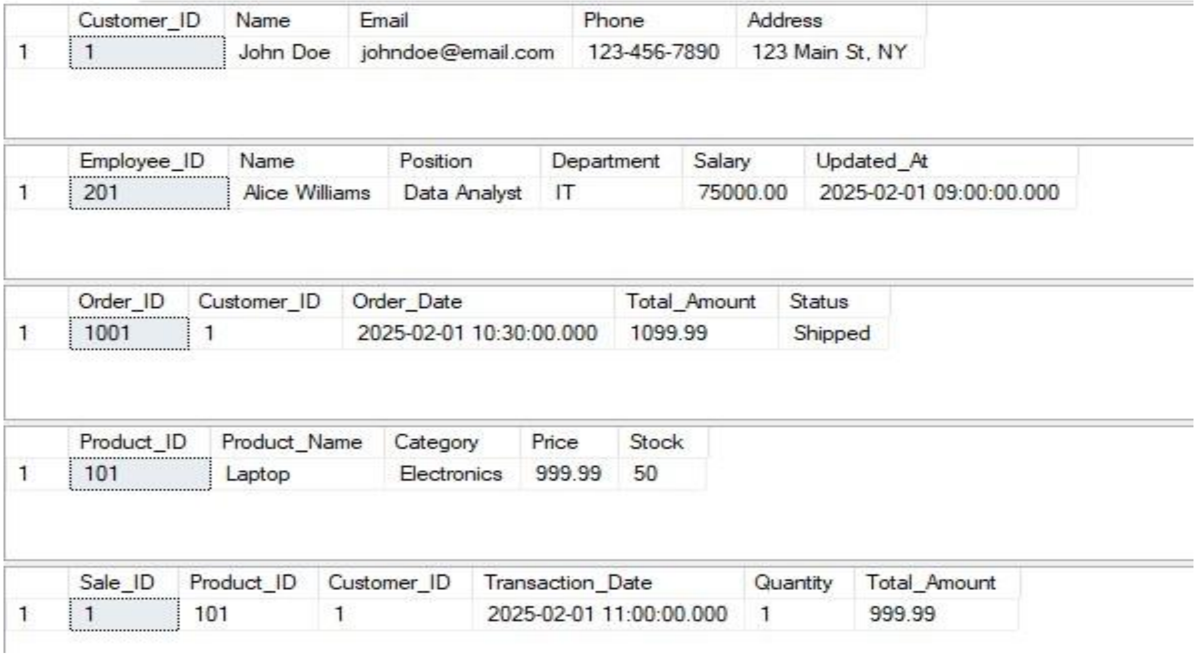
Insert Data into Sales Table (References Product & Customer)

```
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);
```

```
Insert into full_load values(1,'testcase1'),
(2,'testcase2'),
(3,'testcase3'),
(4,'testcase4'),
(5,'testcase5')
```

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;
Select top 1 * from full_load
```



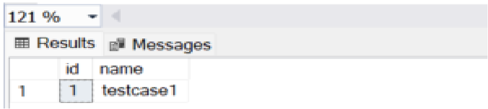
	Customer_ID	Name	Email	Phone	Address
1	1	John Doe	johndoe@email.com	123-456-7890	123 Main St, NY

	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



	id	name
1	1	testcase1

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).
- **Full Load**: It indicates whether we are going to perform a incremental load or full load. (0=incremental load and 1=full load)

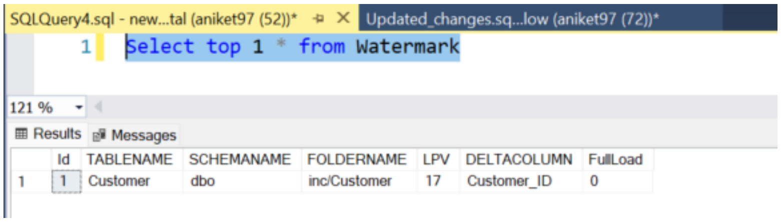
Watermark Table schema query:

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

Insert Values into Watermark Table:

```
Create table Watermark(
Id int,
TABLENAME varchar(50),
SCHEMANAME varchar(30),
FOLDERNAME varchar(100),
LPV varchar(100),
DELTACOLUMN varchar(100)
FullLoad Bit default 0
)
```

Output:


SQLQuery4.sql - new...tal (aniket97 (52))*
Updated_changes.sq...low (aniket97 (72))*
121 %
Results
Messages

```
Select top 1 * from Watermark
```

	Id	TABLENAME	SCHEMANAME	FOLDERNAME	LPV	DELTACOLUMN	FullLoad
1	1	Customer	dbo	inc/Customer	17	Customer_ID	0

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**. Also, for the **full_load** we wont update the LPV value as want it to be 0 everytime.

Stored Procedure Schema Definition:

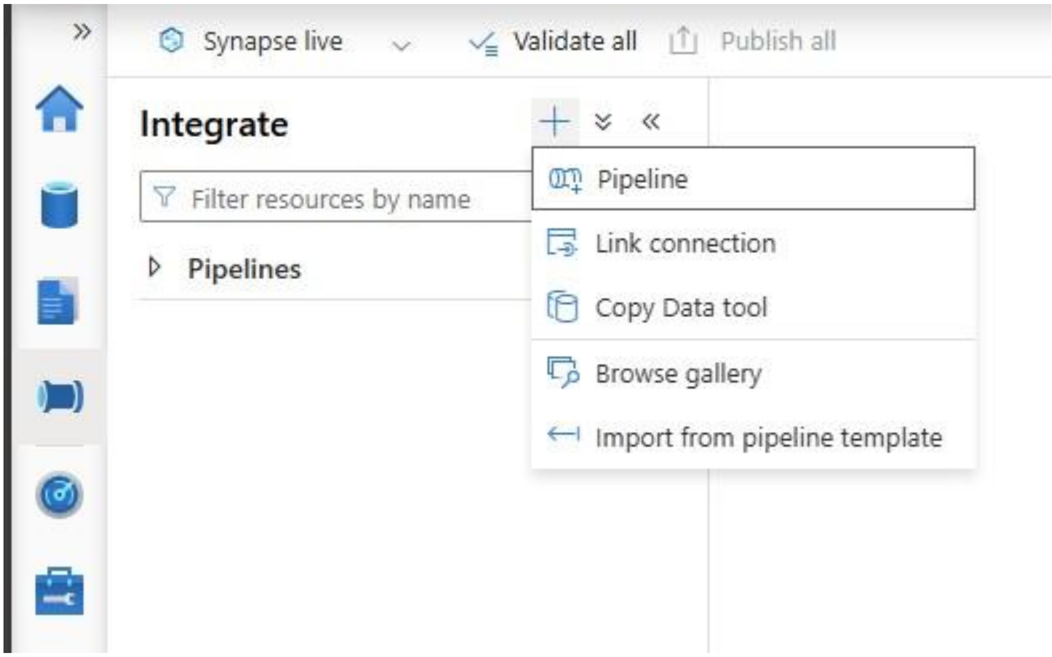
```
CREATE Proc upd_watermark
@TABLENAME varchar(100),
@LPV varchar(100)
AS

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

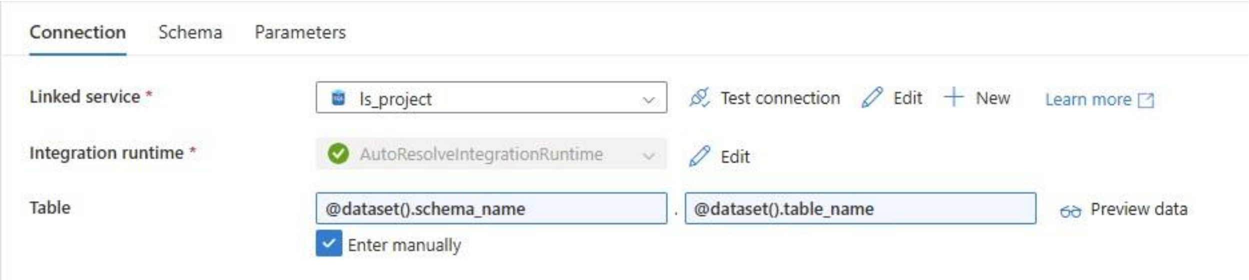
Now, that we have created the tables, watermark table and stored procedure, it’s time to create a pipeline in synapse.

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.



Connection | Schema | Parameters

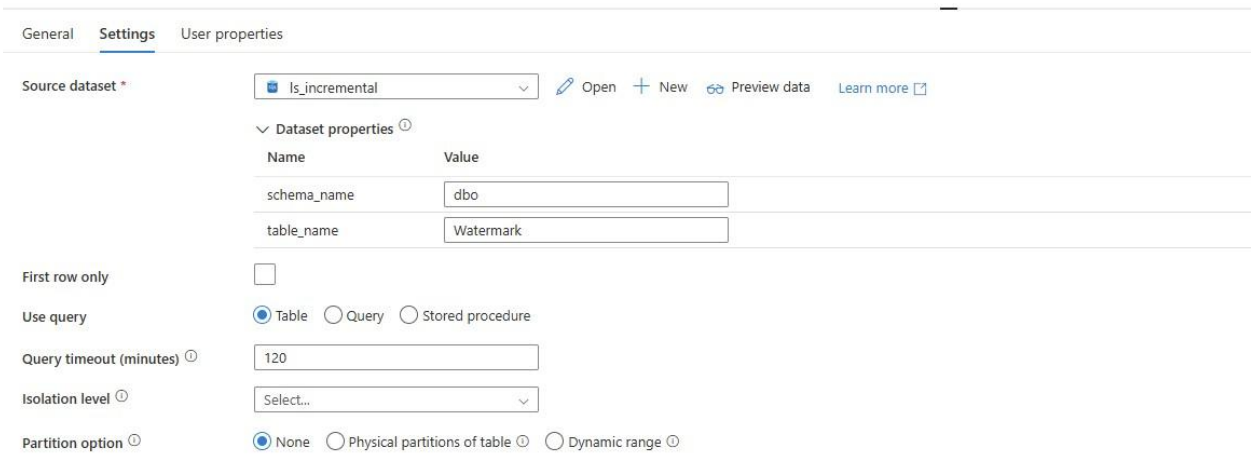
Linked service * ls_project

Integration runtime * AutoResolveIntegrationRuntime

Table @dataset().schema_name . @dataset().table_name

☑ Enter manually

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



General | **Settings** | User properties

Source dataset * : ls_incremental — Open | + New | Preview data | Learn more

Dataset properties

Name	Value
schema_name	dbo
table_name	Watermark

First row only: □

Use query: ● Table ○ Query ○ Stored procedure

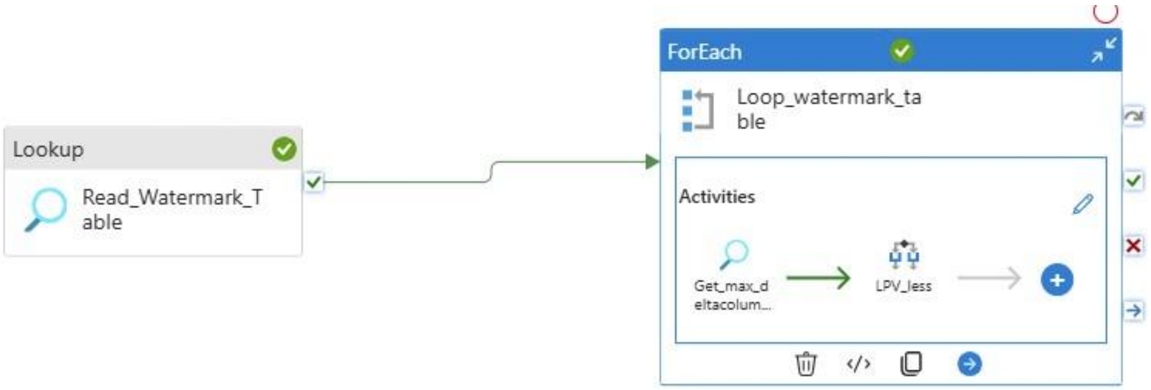
Query timeout (minutes): 120

Isolation level: Select...

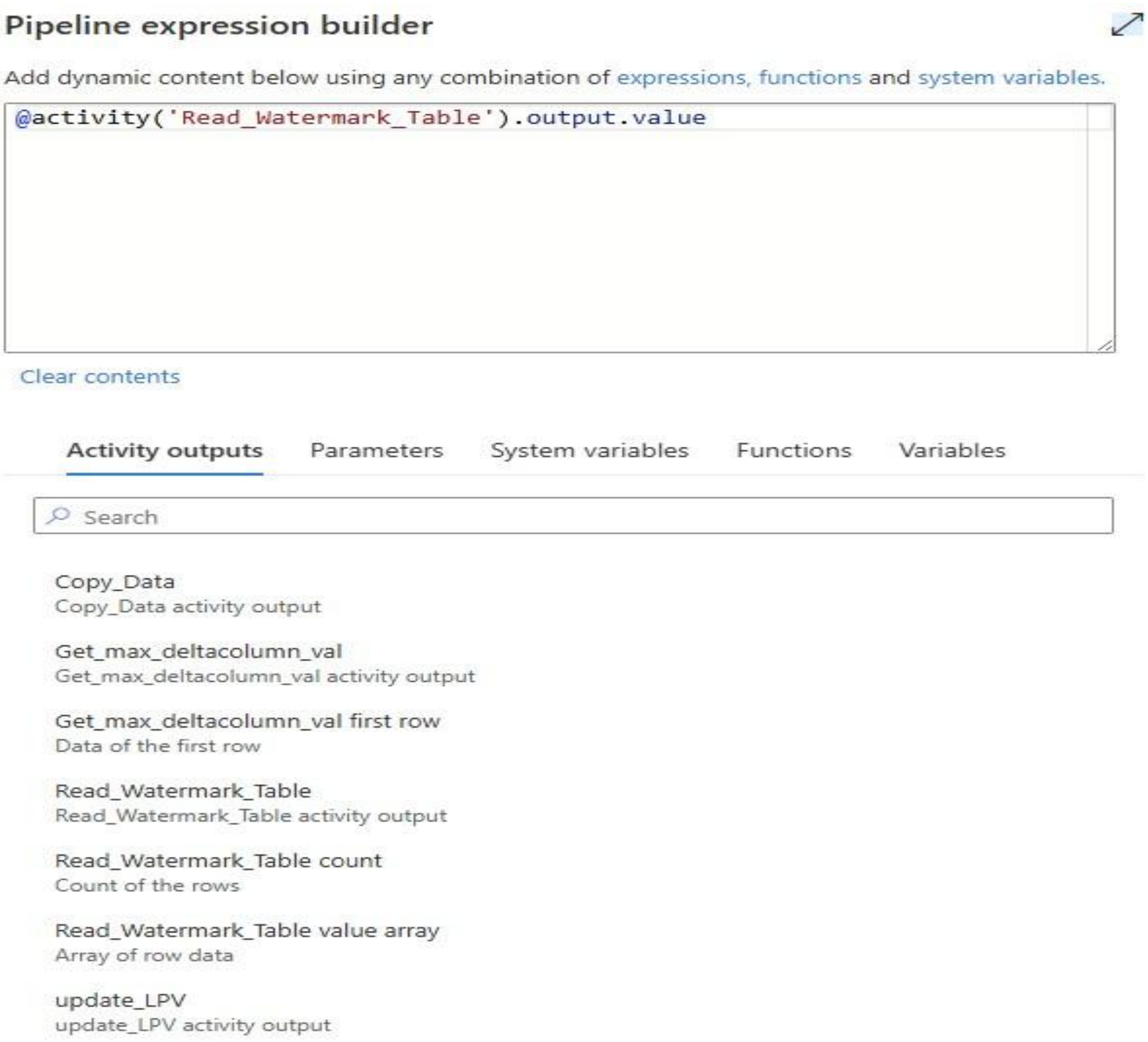
Partition option: ● None ○ Physical partitions of table ○ Dynamic range

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.



Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@activity('Read_Watermark_Table').output.value
```

Copy_Data
Copy_Data activity output

Get_max_deltacolumn_val
Get_max_deltacolumn_val activity output

Get_max_deltacolumn_val first row
Data of the first row

Read_Watermark_Table
Read_Watermark_Table activity output

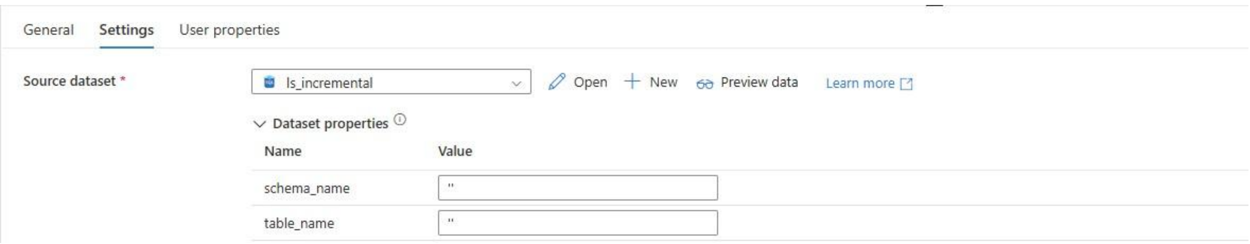
Read_Watermark_Table count
Count of the rows

Read_Watermark_Table value array
Array of row data

update_LPV
update_LPV activity output

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



Source dataset * ls_incremental

Dataset properties

Name	Value
schema_name	''
table_name	''

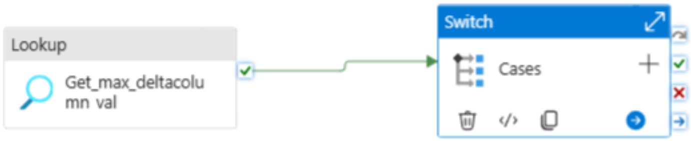
5. Click on the query, and write the below expression to get max value that we will check with LPV and based on it copy the data.

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

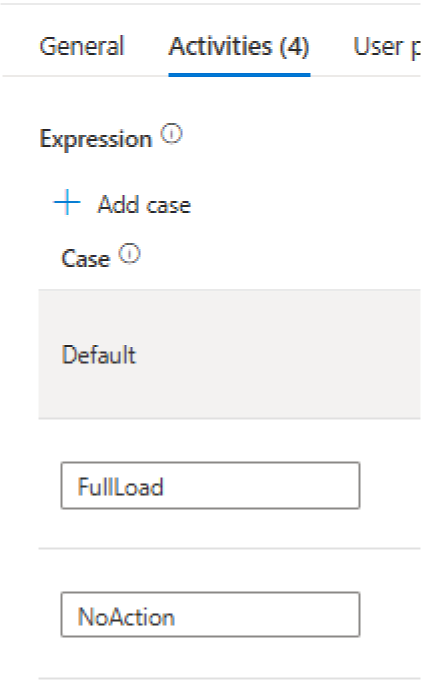
   Here, we are using the foreach activity output to get the Deltacolumn, Tablename information for each iteration using @item() function.

Step 5: Use Switch activity

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



2. Go to activities tab in switch, and create two more cases by clicking on '+ add case' and add FullLoad and NoAction case.



General Activities (4) User p

Expression ⓘ

+ Add case

Case ⓘ

Default

FullLoad

NoAction

3. Now, we write an expression that checks if the fullload column is 1, return value FullLoad. If, fulload column=1 & LPV<Maxdeltacolumn value then return value Default and otherwise return NoAction.

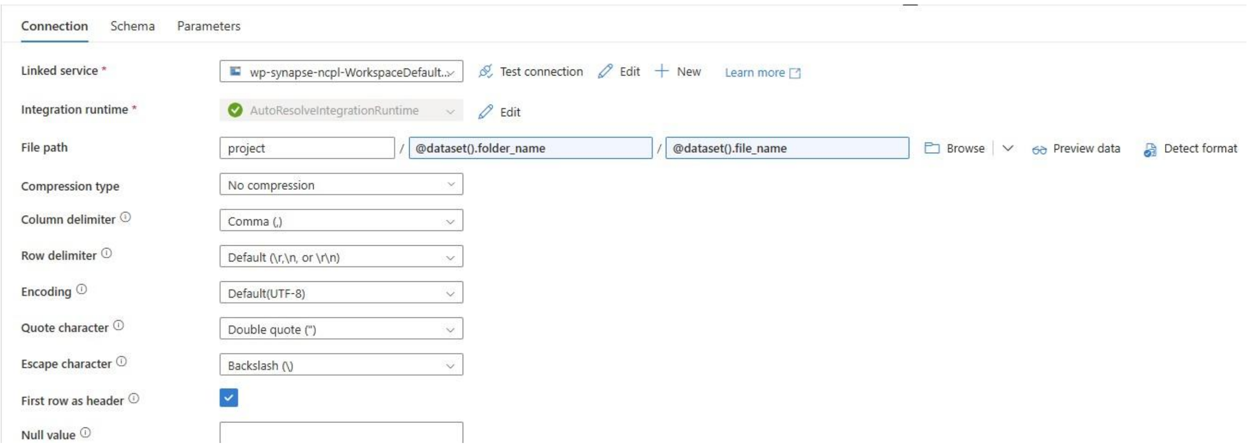
```
@if(
    equals(item().FullLoad, 1),
    'FullLoad',
    if(
        and(
            equals(item().FullLoad, 0),
            less(string(item().LPV),string(activity('Get_max_deltacolumn_val').output
.firstRow.maxvalue)
)
        ),
        'Default',
        'NoAction'
    )
)
```

4. Now click on the pencil icon in default case, it will use the return value 'Default' from the expression. Now perform below steps:

- Drag copy activity and give it a name.
- In the source settings, use the azure sql database dataset.
- Pass empty string ' ' in schemaname and tablename parameter.

- 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}'`

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

Linked service * wp-synapse-ncpl-WorkspaceDefault...

Integration runtime * AutoResolveIntegrationRuntime

File path project / @dataset().folder_name / @dataset().file_name

Compression type No compression

Column delimiter Comma (,)

Row delimiter Default (\r,\n, or \r\n)

Encoding Default(UTF-8)

Quote character Double quote (")

Escape character Backslash (\)

First row as header ☑

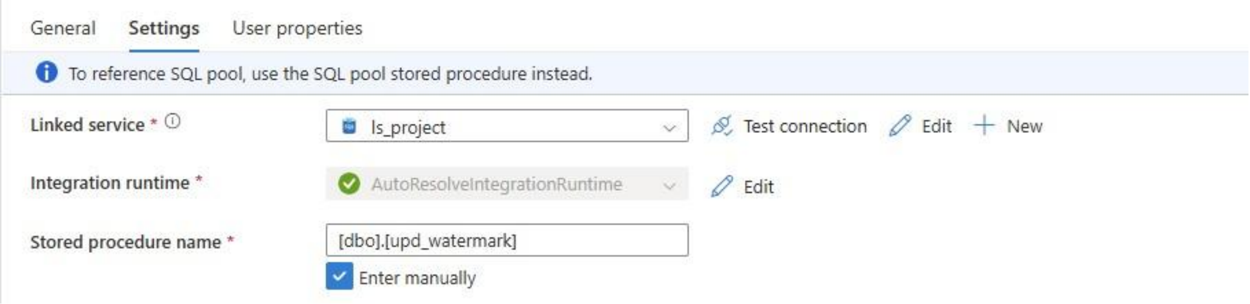
Null value

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

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

Now,

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

- 

General | **Settings** | User properties

To reference SQL pool, use the SQL pool stored procedure instead.

Linked service * : ls_project — Test connection — Edit — New

Integration runtime * : AutoResolveIntegrationRuntime — Edit

Stored procedure name * : [dbo].[upd_watermark]

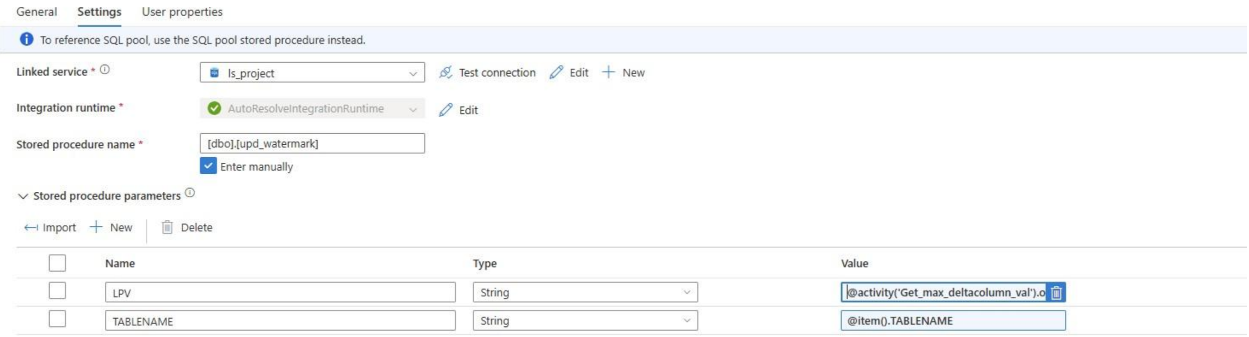
☑ Enter manually

- Click on import parameters under stored procedure parameter.
- 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`

- `In Tablename parameter, pass the tablename using foreach reference.`

- `@item().TABLENAME`



General | **Settings** | User properties

To reference SQL pool, use the SQL pool stored procedure instead.

Linked service * : ls_project — Test connection — Edit — New

Integration runtime * : AutoResolveIntegrationRuntime — Edit

Stored procedure name * : [dbo].[upd_watermark]

☑ Enter manually

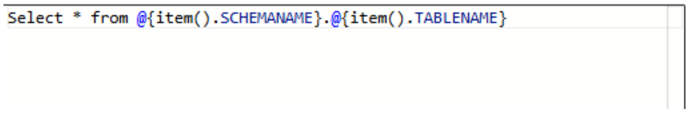
Stored procedure parameters

Import — New — Delete

□	Name	Type	Value
□	LPV	String	@activity('Get_max_deltacolumn_val').o
□	TABLENAME	String	@item().TABLENAME

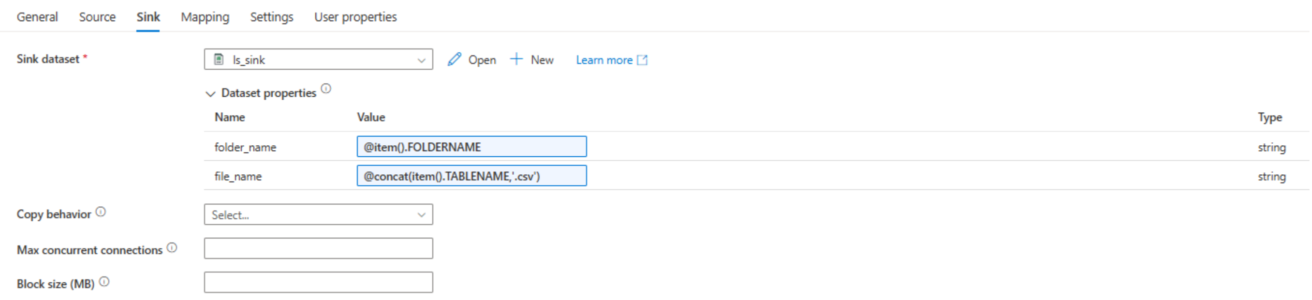
5. Now click on the pencil icon in FullLoad case, it will use the return value ‘FullLoad’ from the expression. Now perform below steps:

- Drag copy activity and connect the source with the Azure sql database. And use the expression on the source side to select all rows from the table.



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

- In the sink settings, connect the dataset and insert following expression in the folder_name and file_name paramters.



Sink dataset * ls_sink

Dataset properties

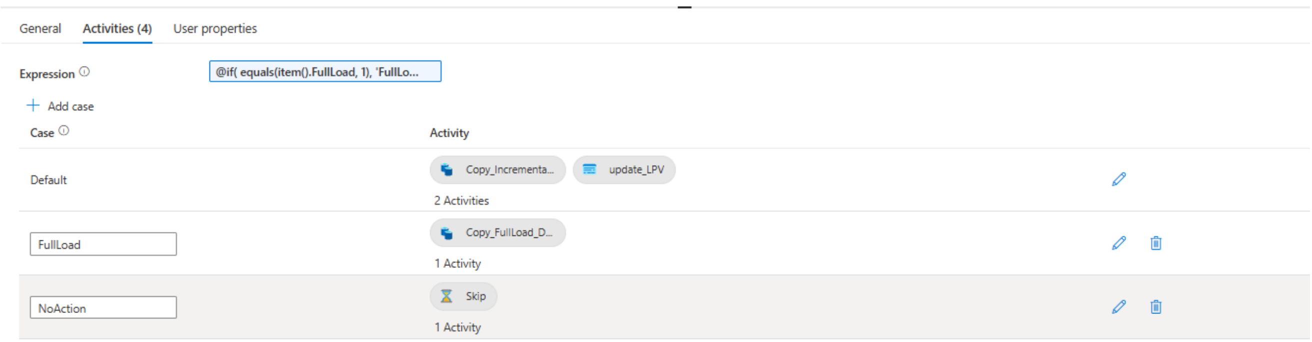
Name	Value	Type
folder_name	@item().FOLDERNAME	string
file_name	@concat(item().TABLENAME,'.csv')	string

Copy behavior: Select...

Max concurrent connections:

Block size (MB):

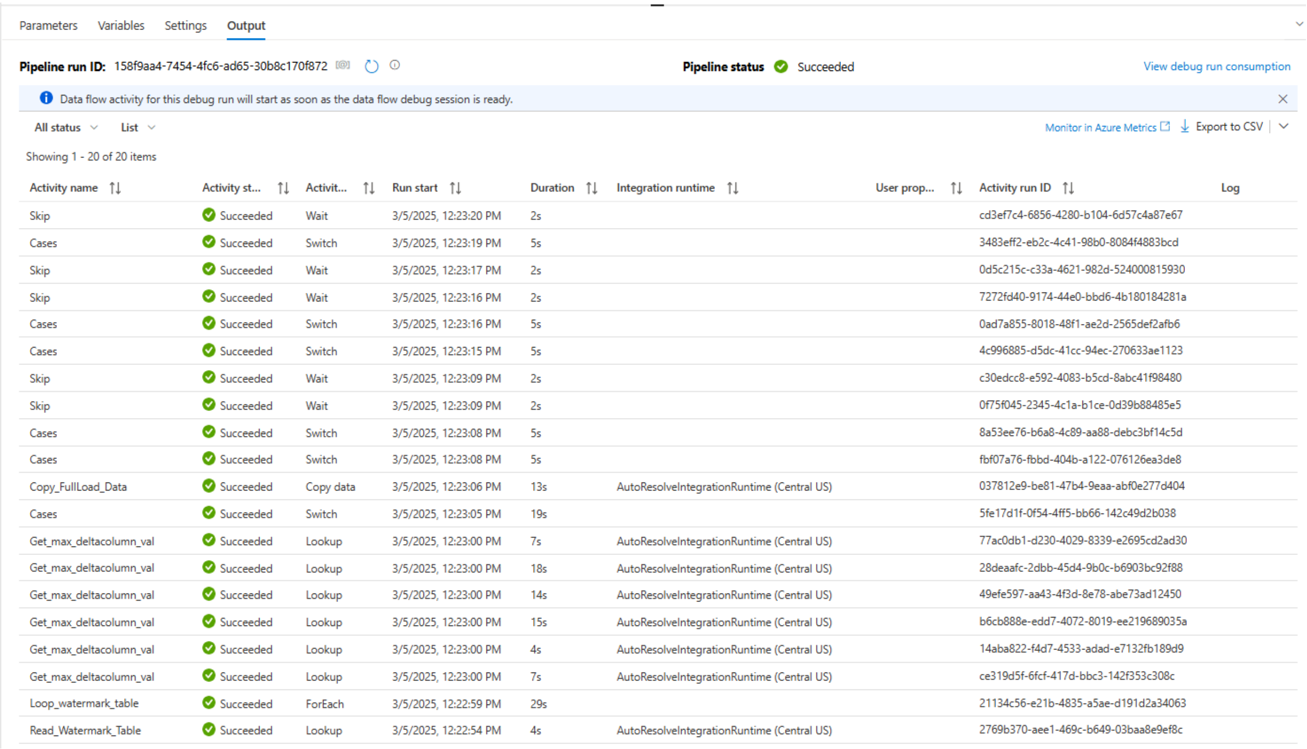
6. Now under NoAction case, click on pencil icon and insert a wait activity as case cannot be empty. We need at least one activity inside it.



Expression: @if(equals(item().FullLoad, 1), 'FullLo...

Case	Activity
Default	Copy_Incrementa... update_LPV 2 Activities
FullLoad	Copy_FullLoad_D... 1 Activity
NoAction	Skip 1 Activity

Step 6: Click on debug pipeline.



Pipeline run ID: 158f9aa4-7454-4fc6-ad65-30b8c170f872

Pipeline status ✓ Succeeded

View debug run consumption

Data flow activity for this debug run will start as soon as the data flow debug session is ready.

Showing 1 - 20 of 20 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID	Log
Skip	✓ Succeeded	Wait	3/5/2025, 12:23:20 PM	2s			cd3ef7c4-6856-4280-b104-6d57c4a87e67	
Cases	✓ Succeeded	Switch	3/5/2025, 12:23:19 PM	5s			3483eff2-eb2c-4c41-98b0-8084f4883bcd	
Skip	✓ Succeeded	Wait	3/5/2025, 12:23:17 PM	2s			0d5c215c-c33a-4621-982d-524000815930	
Skip	✓ Succeeded	Wait	3/5/2025, 12:23:16 PM	2s			7272fd40-9174-44e0-bbd6-4b180184281a	
Cases	✓ Succeeded	Switch	3/5/2025, 12:23:16 PM	5s			0ad7a855-8018-48f1-ae2d-2565def2afb6	
Cases	✓ Succeeded	Switch	3/5/2025, 12:23:15 PM	5s			4c996885-d5dc-41cc-94ec-270633ae1123	
Skip	✓ Succeeded	Wait	3/5/2025, 12:23:09 PM	2s			c30edcc8-e592-4083-b5cd-8abc41f98480	
Skip	✓ Succeeded	Wait	3/5/2025, 12:23:09 PM	2s			0f75f045-2345-4c1a-b1ce-0d39b88485e5	
Cases	✓ Succeeded	Switch	3/5/2025, 12:23:08 PM	5s			8a53ee76-b6a8-4c89-aa88-debc3bf14c5d	
Cases	✓ Succeeded	Switch	3/5/2025, 12:23:08 PM	5s			fbf07a76-fbbd-404b-a122-076126ea3de8	
Copy_FullLoad_Data	✓ Succeeded	Copy data	3/5/2025, 12:23:06 PM	13s	AutoResolveIntegrationRuntime (Central US)		037812e9-be81-47b4-9eaa-abf0e277d404	
Cases	✓ Succeeded	Switch	3/5/2025, 12:23:05 PM	19s			5fe17d1f-0f54-4ff5-bb66-142c49d2b038	
Get_max_deltacolumn_val	✓ Succeeded	Lookup	3/5/2025, 12:23:00 PM	7s	AutoResolveIntegrationRuntime (Central US)		77ac0db1-d230-4029-8339-e2695cd2ad30	
Get_max_deltacolumn_val	✓ Succeeded	Lookup	3/5/2025, 12:23:00 PM	18s	AutoResolveIntegrationRuntime (Central US)		28deaafc-2dbb-45d4-9b0c-b6903bc92f88	
Get_max_deltacolumn_val	✓ Succeeded	Lookup	3/5/2025, 12:23:00 PM	14s	AutoResolveIntegrationRuntime (Central US)		49efe597-aa43-4f3d-8e78-abe73ad12450	
Get_max_deltacolumn_val	✓ Succeeded	Lookup	3/5/2025, 12:23:00 PM	15s	AutoResolveIntegrationRuntime (Central US)		b6cb888e-edd7-4072-8019-ee219689035a	
Get_max_deltacolumn_val	✓ Succeeded	Lookup	3/5/2025, 12:23:00 PM	4s	AutoResolveIntegrationRuntime (Central US)		14aba822-f4d7-4533-adad-e7132fb189d9	
Get_max_deltacolumn_val	✓ Succeeded	Lookup	3/5/2025, 12:23:00 PM	7s	AutoResolveIntegrationRuntime (Central US)		ce319d5f-6fcf-417d-bbc3-142f353c308c	
Loop_watermark_table	✓ Succeeded	ForEach	3/5/2025, 12:22:59 PM	29s			21134c56-e21b-4835-a5ae-d191d2a34063	
Read_Watermark_Table	✓ Succeeded	Lookup	3/5/2025, 12:22:54 PM	4s	AutoResolveIntegrationRuntime (Central US)		2769b370-aee1-469c-b649-03baa8e9ef8c	