DATA MIGRATION

Incremental data Load from on-Prem DB to Azure SQL DB Using SCD Type0, Type1, Type2 Approach

MARCH 11, 2025

BY HARJINDER SINGH

Table of Contents

1. Table Creation on on-prem SQL DB

I have taken the same five source tables: **Books, Members, BorrowRecords, Inventory, and Librarians**.

I used this set of tables to demonstrate **Incremental Data Load** from **On-prem SQL DB** to **ADLS Gen2**. Then, further the data is moved to **Azure SQL DB** using **the Same Azure Synapse Workspace Pipeline** with three different Method:

- **Normal Datal load** (Used for three tables named **BorrowRecords, Inventory, and Librarians**).
- **SCD Type1** (Used for only a table named **Books**).
- **SCD Type2** (Used for only a table named **Members**).

➢ The tables schema definitions are as follows:

Here, the **mylocaldb** DB is created to store all the tables data.

Create Database mylocaldb

Books Table -

```
Create Table Books (
  book_id Int,
  book_title Varchar(255),
  book_author Varchar(255),
  book_genre Varchar(100),
  published_year Int,
    last_updated Datetime --- this column will be used as Delta Column
)
```

Members Tables -

```
Create Table Members (
  member_id Int,
  member_name Varchar(100),
  member_email Varchar(255),
  member_phone BigInt,
    last_updated Datetime --- this column will be used as Delta Column
)
```

BorrowRecords Table -

```
Create Table BorrowRecords (
  borrow_date Datetime, --- this column will be used as Delta Column
```

```
  book_id Int,
  member_id Int,
  return_days Int
)
```

Inventory Table -

```
Create Table Inventory (
  last_updated Datetime, --- this column will be used as Delta Column
  book_id Int,
  quantity Int
)
```

Librarians Table

```
Create Table Librarians (
  lib_id Int, --- this column will be used as Delta Column
    lb_name Varchar(100),
  lb_email Varchar(255),
    hired_at Datetime
)
```

Here, **one table** contain a column with an ***INT*** data type that serves as a **unique identifier**:

- **Librarians**: The column ***lib_id*** uniquely identifies each librarian.

Four tables contain a column with a ***DATETIME*** data type that serves as a **unique identifie**r:

- **BorrowRecords**: The column ***borrow_date*** records the date and time when a book was borrowed.
- **Inventory**: The column ***last_updated*** tracks the most recent update to the inventory.
- **Books**: The column ***last_updated*** tracks the most recent update to the Books records.
- **Members**: The column ***last_updated*** tracks the most recent update to the Members records.

➢ Here's a well-structured SQL insertion script for the mentioned tables, ensuring proper data entry while maintaining relationships:

---Books Table
Insert Into Books (book_id, book_title, book_author, book_genre, published_year, last_updated)
Values
(101,'The White Tiger', 'Aravind Adiga', 'Fiction', 2008, '2025-03-07 00:00:00'),
(102,'The Guide', 'R.K. Narayan', 'Fiction', 1958, '2025-03-07 00:00:00'),
(103,'Chetan Bhagat - One Indian Girl', 'Chetan Bhagat', 'Romance', 2016, '2025-03-07 00:00:00'),
(104,'The God of Small Things', 'Arundhati Roy', 'Fiction', 1997, '2025-03-07 00:00:00')

---Members Table
Insert Into Members (member_id, member_name, member_email, member_phone, last_updated)
Values
(1,'Rahul Sharma', 'rahul.sharma@gmail.com', 9876543210, '2025-03-07 00:00:00'),
(2,'Priya Patel', 'priya.patel@yahoo.com', 9988776655, '2025-03-07 00:00:00'),
(3,'Vikram Kumar', 'vikram.kumar@outlook.com', 9871234567, '2025-03-07 00:00:00'),
(4,'Ananya Gupta', 'ananya.gupta@rediffmail.com', 9998887770, '2025-03-07 00:00:00')

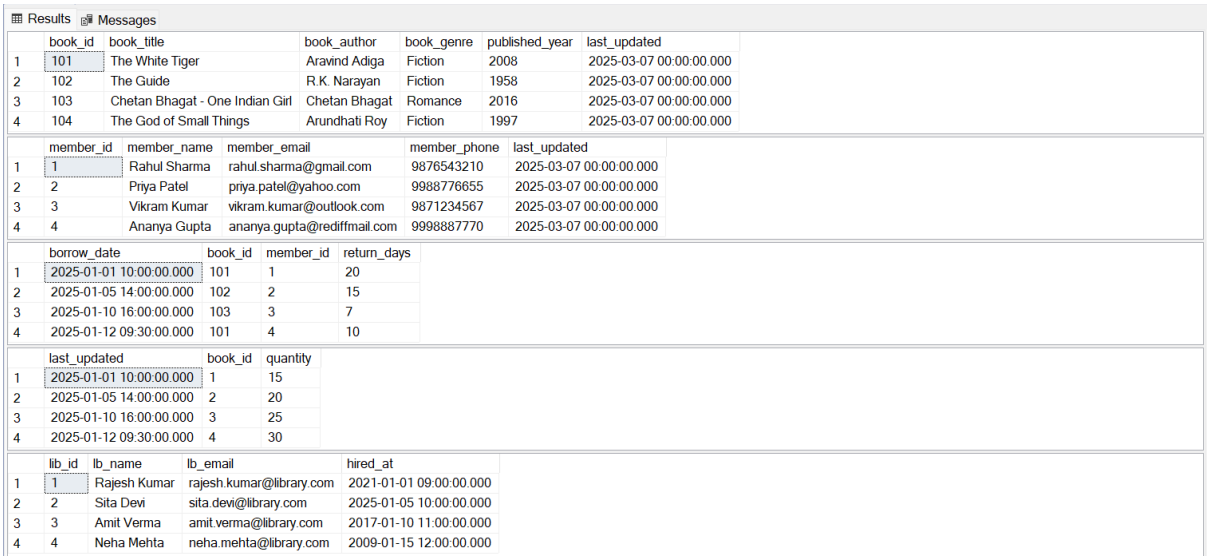
---BorrowRecords Table
Insert Into BorrowRecords (borrow_date, book_id, member_id, return_days)
Values
('2025-01-01 10:00:00', 101, 1, 20),
('2025-01-05 14:00:00', 102, 2, 15),
('2025-01-10 16:00:00', 103, 3, 7),
('2025-01-12 09:30:00', 101, 4, 10)

---Inventory Table
Insert Into Inventory (last_updated, book_id, quantity)
Values
('2025-01-01 10:00:00', 1, 15),
('2025-01-05 14:00:00', 2, 20),
('2025-01-10 16:00:00', 3, 25),
('2025-01-12 09:30:00', 4, 30)

---Librarians Table
Insert Into Librarians (lib_id, hired_at, lb_name, lb_email)
Values
(1, '2021-01-01 09:00:00', 'Rajesh Kumar', 'rajesh.kumar@library.com'),

(2, '2025-01-05 10:00:00', 'Sita Devi', 'sita.devi@library.com'),
(3, '2017-01-10 11:00:00', 'Amit Verma', 'amit.verma@library.com'),
(4, '2009-01-15 12:00:00', 'Neha Mehta', 'neha.mehta@library.com')

➢ Table Results:



Results Messages

	book_id	book_title	book_author	book_genre	published_year	last_updated
1	101	The White Tiger	Aravind Adiga	Fiction	2008	2025-03-07 00:00:00.000
2	102	The Guide	R.K. Narayan	Fiction	1958	2025-03-07 00:00:00.000
3	103	Chetan Bhagat - One Indian Girl	Chetan Bhagat	Romance	2016	2025-03-07 00:00:00.000
4	104	The God of Small Things	Arundhati Roy	Fiction	1997	2025-03-07 00:00:00.000

	member_id	member_name	member_email	member_phone	last_updated
1	1	Rahul Sharma	rahul.sharma@gmail.com	9876543210	2025-03-07 00:00:00.000
2	2	Priya Patel	priya.patel@yahoo.com	9988776655	2025-03-07 00:00:00.000
3	3	Vikram Kumar	vikram.kumar@outlook.com	9871234567	2025-03-07 00:00:00.000
4	4	Ananya Gupta	ananya.gupta@rediffmail.com	9998887770	2025-03-07 00:00:00.000

	borrow_date	book_id	member_id	return_days
1	2025-01-01 10:00:00.000	101	1	20
2	2025-01-05 14:00:00.000	102	2	15
3	2025-01-10 16:00:00.000	103	3	7
4	2025-01-12 09:30:00.000	101	4	10

	last_updated	book_id	quantity
1	2025-01-01 10:00:00.000	1	15
2	2025-01-05 14:00:00.000	2	20
3	2025-01-10 16:00:00.000	3	25
4	2025-01-12 09:30:00.000	4	30

	lib_id	lb_name	lb_email	hired_at
1	1	Rajesh Kumar	rajesh.kumar@library.com	2021-01-01 09:00:00.000
2	2	Sita Devi	sita.devi@library.com	2025-01-05 10:00:00.000
3	3	Amit Verma	amit.verma@library.com	2017-01-10 11:00:00.000
4	4	Neha Mehta	neha.mehta@library.com	2009-01-15 12:00:00.000

2. Watermark Table

A **Watermark table** is used in **Incremental Data Load** in Azure Data Factory (ADF) and Azure Synapse Analytics to track the last loaded record using a **timestamp or an ID column.**

It helps in loading only new or updated records instead of reprocessing all data, making the process efficient and faster.

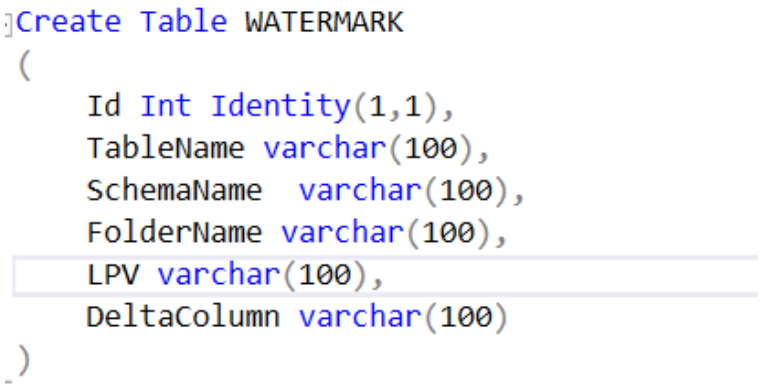
Next, Watermark Table is created with some metadata records of source tables such as **table name, schema name, destination folder, Last Processed Value, and Delta Column**.

**** LPV (Last Processed Value):** This column contains the last processed value of the Delta Column from the source table in the pipeline.

**** Delta Column**: This column contains the column name which is used to identify the new records/data from the source tables.

- The ***Watermark*** tables schema definition is as follows:

```
Create Table WATERMARK
(
        Id Int Identity(1,1),
        TableName varchar(100),
        SchemaName  varchar(100),
        FolderName varchar(100),
        LPV varchar(100),
        DeltaColumn varchar(100)
)
```



- Here's a well-structured SQL insertion script for the Watermark tables, inserting Metadata Records for the source tables:

```
Insert Into WATERMARK
Values
('Books','DBO','onPrem_LibraryDB/Books','1900-01-01 00:00:00','last_updated'),
('Members','DBO','onPrem_LibraryDB/Members','1900-01-01
00:00:00','last_updated'),
('BorrowRecords','DBO','onPrem_LibraryDB/BorrowRecords','1900-01-01
00:00:00','borrow_date'),
('Inventory','DBO','onPrem_LibraryDB/Inventory','1900-01-01
00:00:00','last_updated'),
('Librarians','DBO','onPrem_LibraryDB/Librarians','0','lib_id')
```

```
Insert Into WATERMARK
Values
('Books','DBO','onPrem_LibraryDB/Books','1900-01-01 00:00:00','last_updated'),
('Members','DBO','onPrem_LibraryDB/Members','1900-01-01 00:00:00','last_updated'),
('BorrowRecords','DBO','onPrem_LibraryDB/BorrowRecords','1900-01-01 00:00:00','borrow_date'),
('Inventory','DBO','onPrem_LibraryDB/Inventory','1900-01-01 00:00:00','last_updated'),
('Librarians','DBO','onPrem_LibraryDB/Librarians','0','lib_id')
```

- Output

Results | Messages

	Id	TableName	SchemaName	FolderName	LPV	DeltaColumn
1	1	Books	DBO	onPrem_LibraryDB/Books	2025-03-08T00:00:00	last_updated
2	2	Members	DBO	onPrem_LibraryDB/Members	2025-03-09T00:00:00	last_updated
3	3	BorrowRecords	DBO	onPrem_LibraryDB/BorrowRecords	2025-01-12T09:30:00	borrow_date
4	4	Inventory	DBO	onPrem_LibraryDB/Inventory	2025-01-12T09:30:00	last_updated
5	5	Librarians	DBO	onPrem_LibraryDB/Librarians	5	lib_id

Point to Note:

The LPV (**Last Processed Value)** for all records is initially set to basic values -

- For INT type Delta Column such as *lib_id* is set **'0'** initially.
- For DATETIME type Delta Clumn such *as borrow_date, last_updated,* and *hired_at* are set to **'1900-01-01 00:00:00'** initially.

3. Stored Procedure for Incremental Data Load

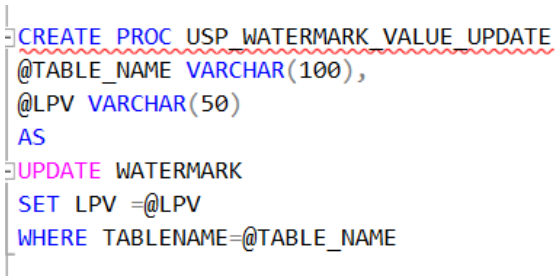
A **Stored Procedure** in **Incremental Data Load** in **ADF and Synapse** is used to **update the last processed values (watermark) to track new records**.

It ensures that only the latest data is loaded in the next cycle, improving efficiency and preventing duplicate processing.

In this, the **Stored Procedure** is created to update the **Last Processed Value (LPV)** in the **Watermark table**, allowing the system to identify new records by comparing the **delta column values** of incoming data with the updated LPV.

- The ***Stored Procedure*** definition is as follows:

```
CREATE PROC USP_WATERMARK_VALUE_UPDATE
@TABLE_NAME VARCHAR(100),
@LPV VARCHAR(50)
AS
UPDATE WATERMARK
SET LPV =@LPV
WHERE TABLENAME=@TABLE_NAME
```



Point to Note: The two Parameters is created in Stored Procedure to **update** the **LPV** value **based on Table Name**.

4. Dynamic Pipeline Design

Now, the pipeline is created to load the data from source to destination incrementally.

The Following Activities will be used in pipeline creation:

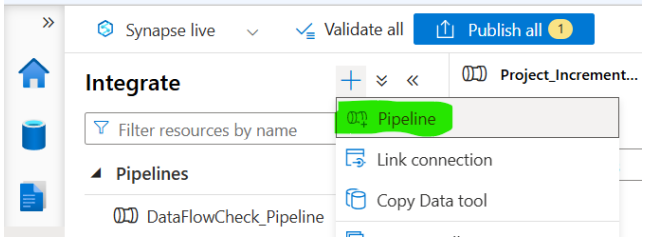
- Set Variable Activity,
- 2 Lookup Activities,
- Foreach Activity,
- 2 Copy Activities,
- IF Condition Activity,
- Store Procedure Activity

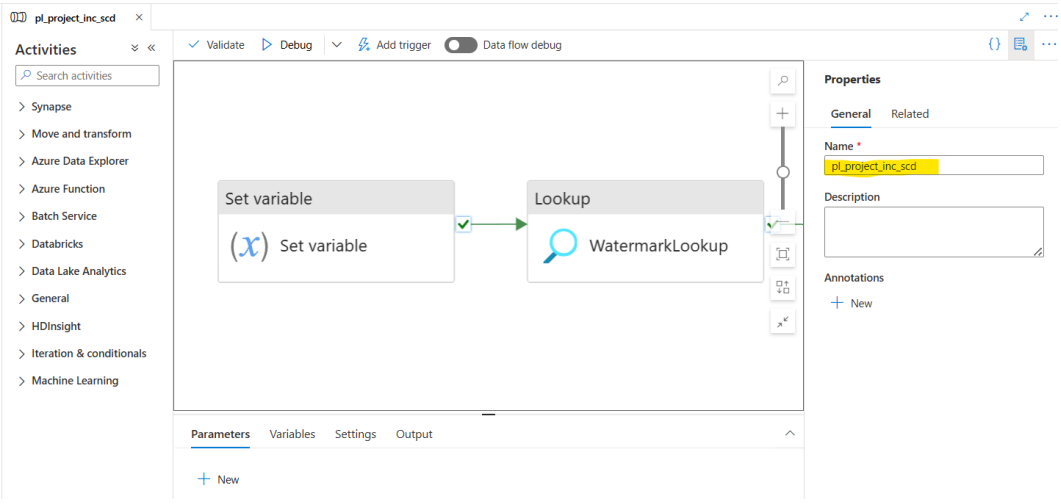
Here, the **Source is on-prem SQL Database** and **Sink is ADLS Gen2 Storage**.

➢ **Steps To Create the Dynamic Pipeline are as follows.**

STEP 1: New Pipeline Creation and Set up Lookup Activity.

- Create a new Pipeline and Drag a **Set Variable Activity** and a **Lookup Activity**. And Connect **Set Variable Activity** with on success connection of **Lookup Activity.**
- Set up the names for both as per the project.
- Here I have named the pipeline as *pl_project_inc_scd* and Lookup activity as *WatermarkLookup*.



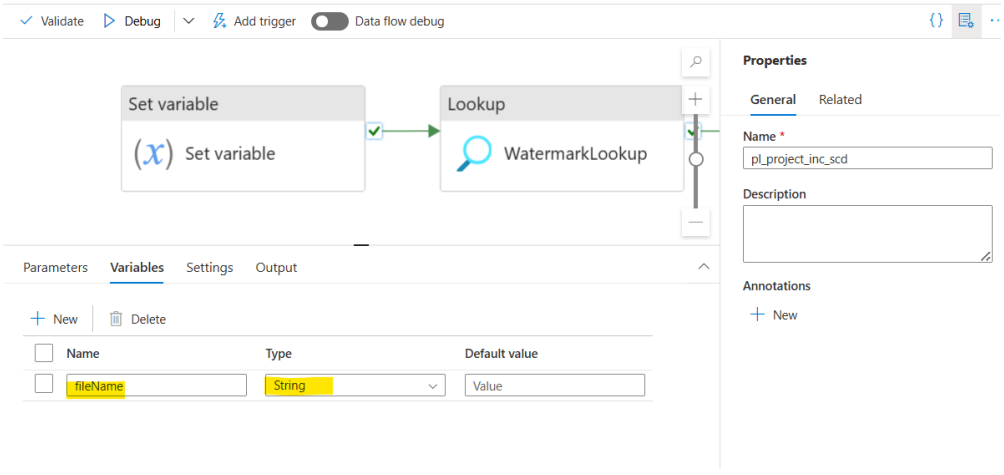


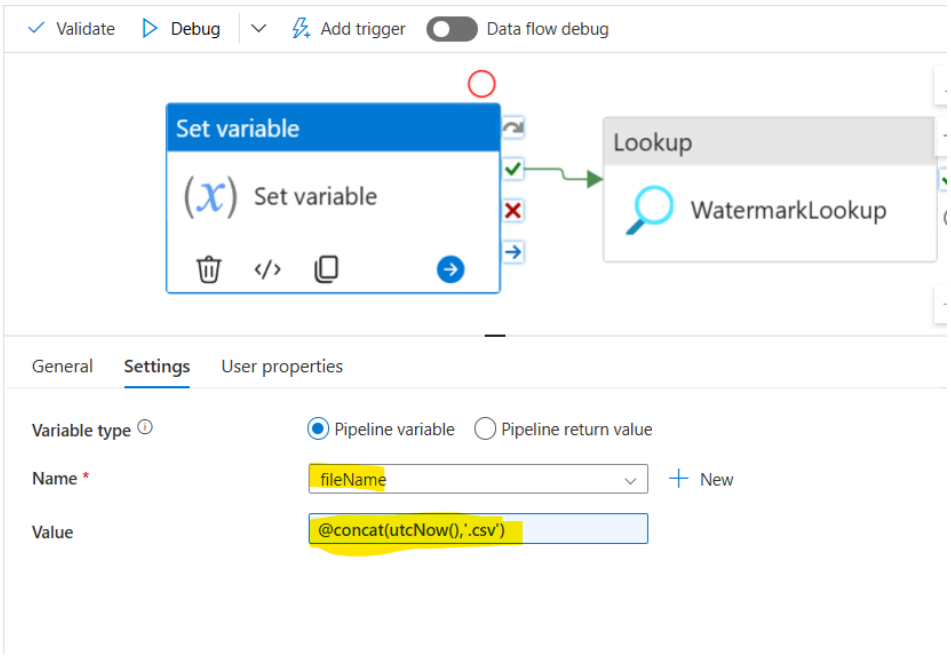
– Create a Variable named **filename** at pipeline level and set the same variable with current time stamp value using **Set Variable Activity.**

- Expression Used to set the variable name:

  @concat(utcNow(),'.csv')

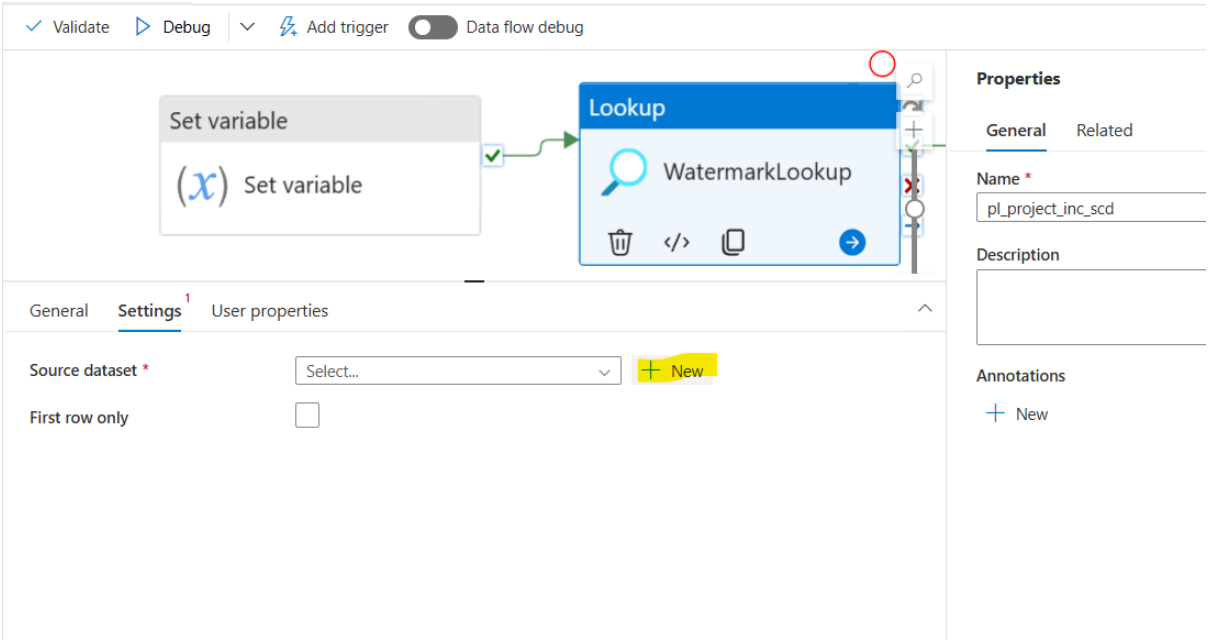
- Using the **concat function** to set the file name in ***currentDateTime.csv*** format.

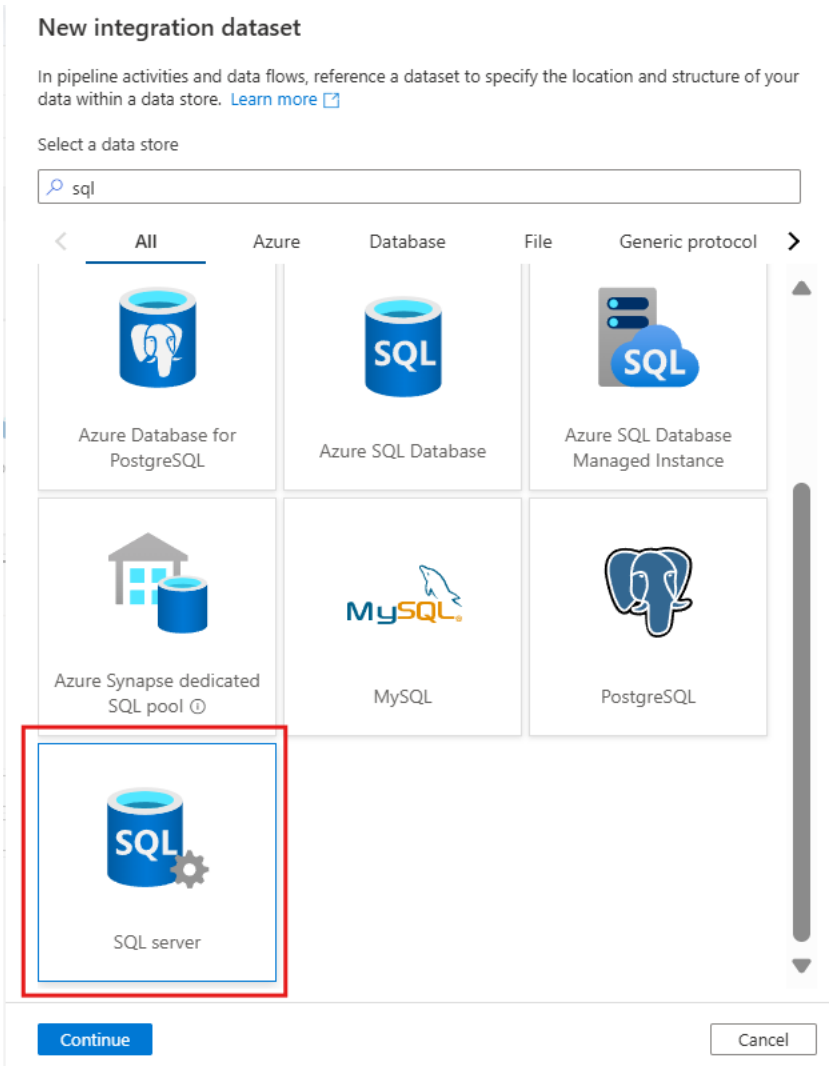




– Set the Source data set in the *WatermarkLookup* Activity by creating a new dataset or selecting the existing data set. Here, I have created a new data set i.e., SQL Server for on-prem SQL DB as the Source Dataset.

- Click on **new** -> create new dataset by selecting **SQL server** -> create new linked services.


Validate
Debug
Add trigger
Data flow debug
Set variable
Set variable
Lookup
WatermarkLookup
Properties
General
Related
Name *
pl_project_inc_scd
Description
General
Settings
1
User properties
Source dataset *
Select...
New
First row only
Annotations
New

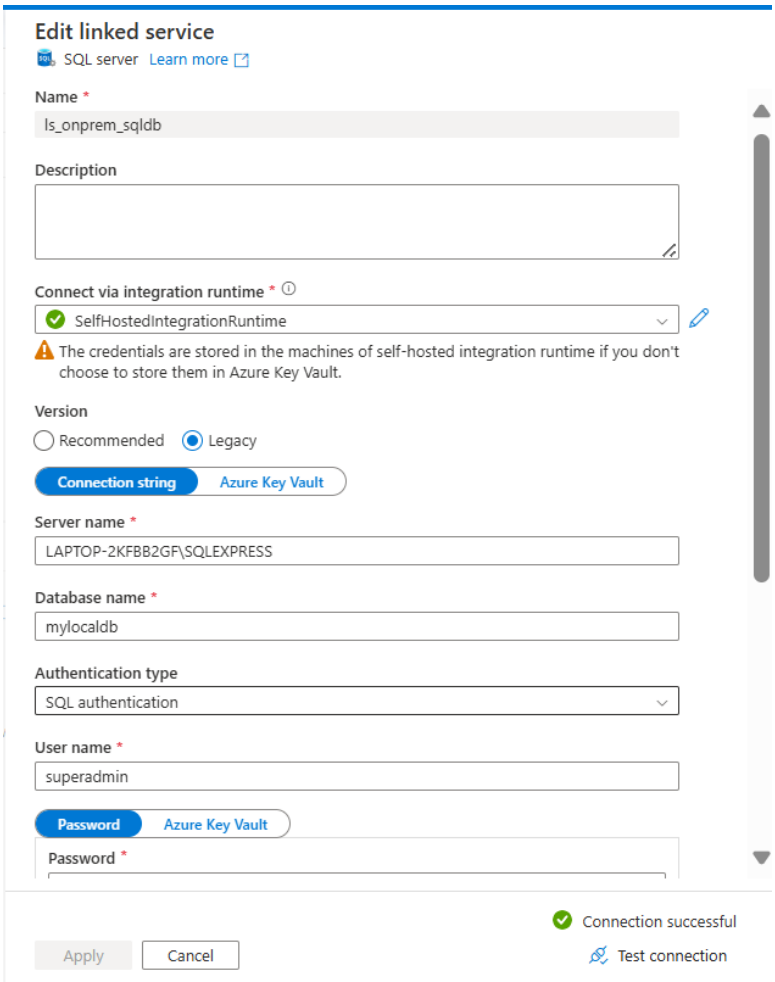

New integration dataset
In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more
Select a data store
sql
All
Azure
Database
File
Generic protocol
Azure Database for PostgreSQL
Azure SQL Database
Azure SQL Database Managed Instance
Azure Synapse dedicated SQL pool
MySQL
PostgreSQL
SQL server
Continue
Cancel

– I have connected my on-prem SQL DB and connected it using **Self Hosted Integration Runtime**.

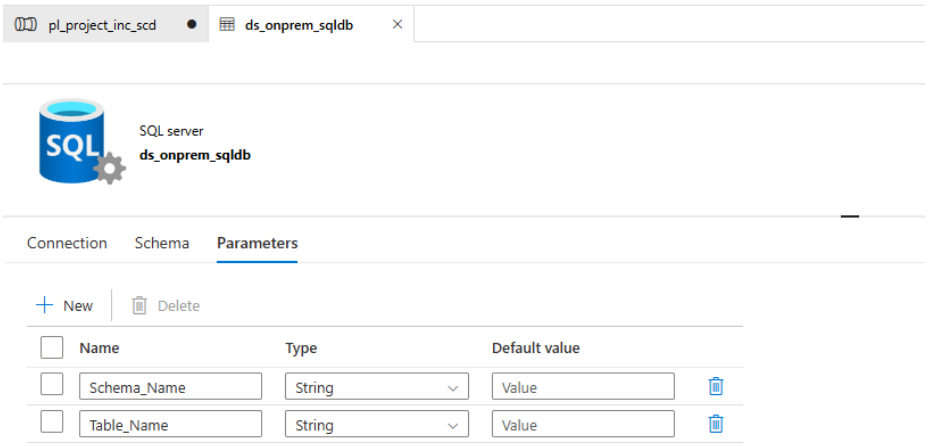
Points to Notes:

- Here, I have already created the **Self Hosted Integration Runtime** in the Azure Synapse Workspace.
- Also, Setup the SQL server in my local machine to make it act like as on prem SQL Server DB. The SQL Server DB Authentication is already created (**Login** userename as **superadmin**).

– To create new linked service for on-prem SQL Server DB, follow below configurations as shown:

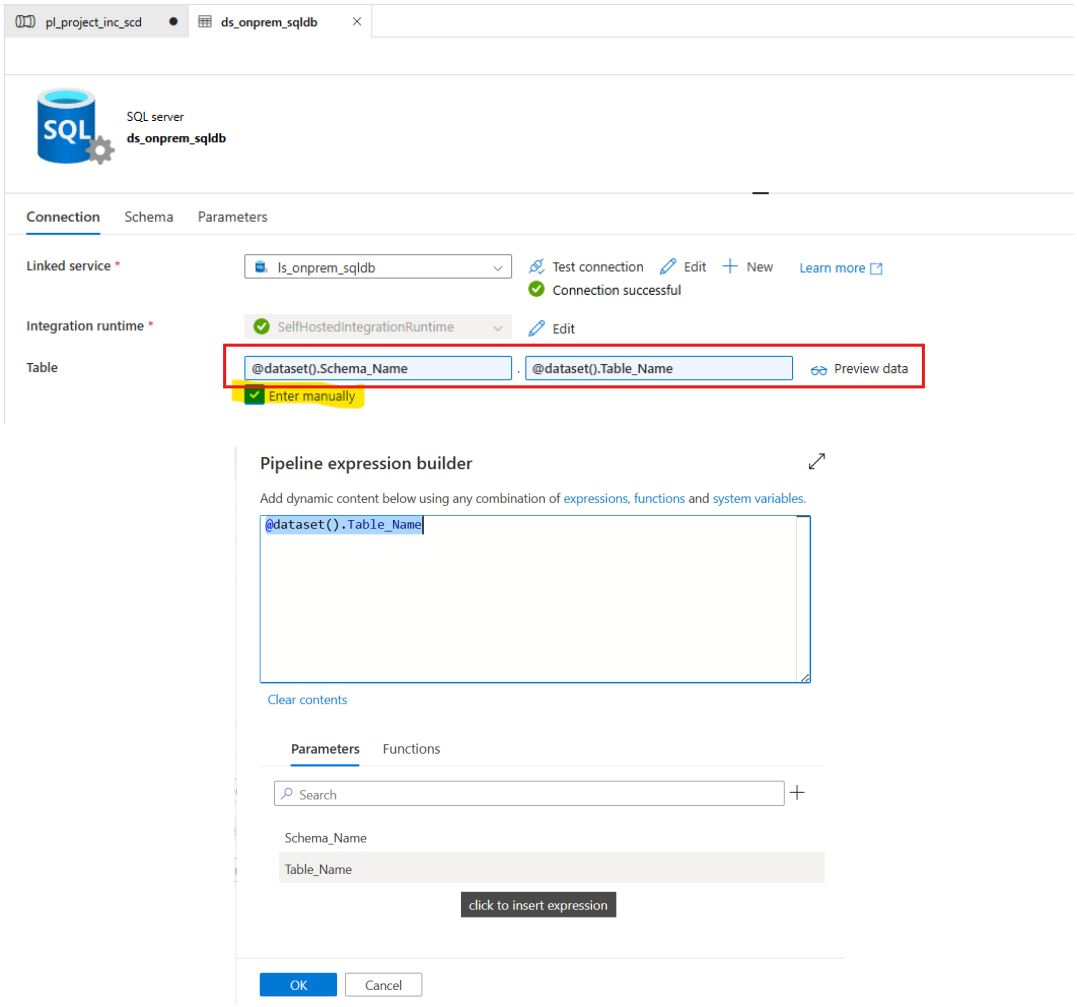


– Open the source dataset and create parameters for the schema name and table name under the parameter tab.



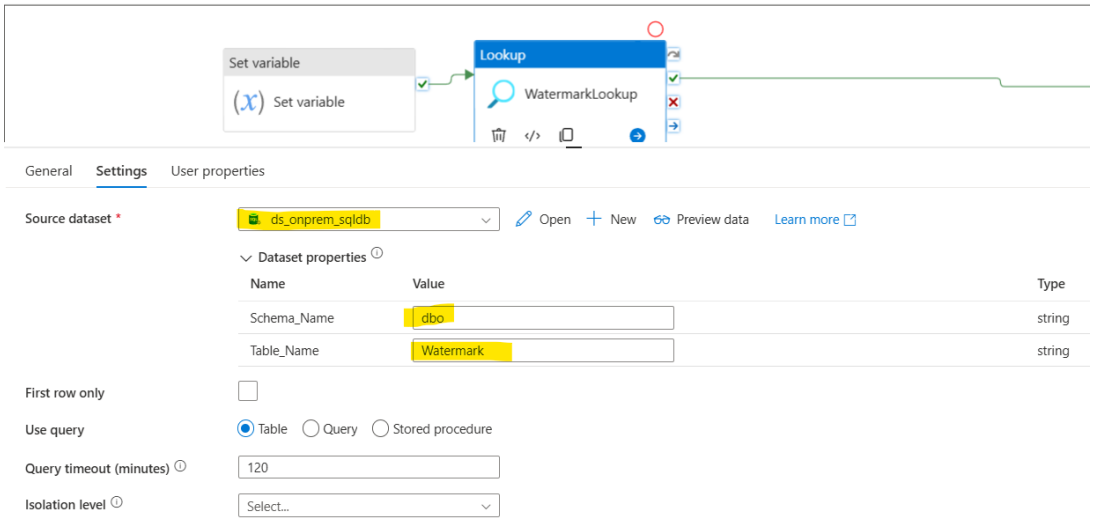
– In the Connection tab, select the Manual Option for Table and enter the mentioned Schema name and Table Name Dynamically as shown below.

Schema_Name expression: ***@dataset().Schema_Name***
Table_Name expression: ***@dataset().Table_Name***



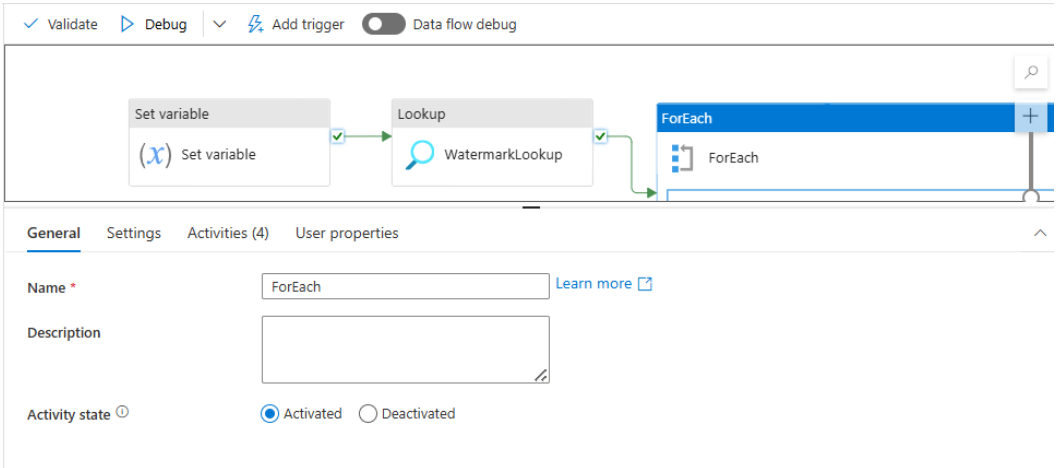
– Next, move to Setting Tab of **WatermarkLookup Activity**, enter the mentioned Schema name and Table Name for the appeared parameters as shown below.

Schema_Name: ***dbo***
Table_Name: ***Watermark***



STEP 2: Setup For Each Activity

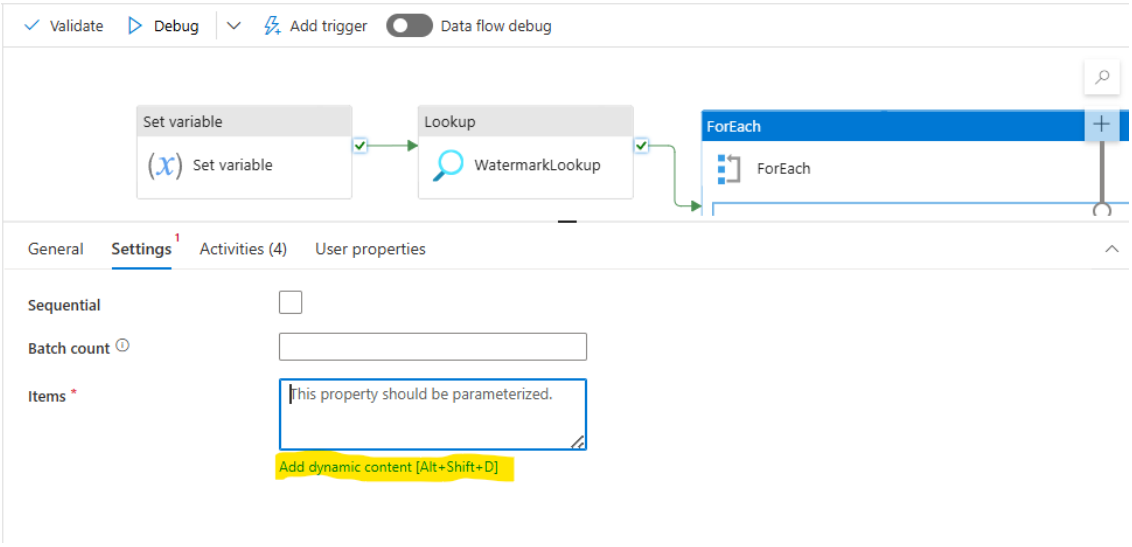
– After creating the Lookup, add the Foreach Activity and connect the Lookup with it using the on-success connection.

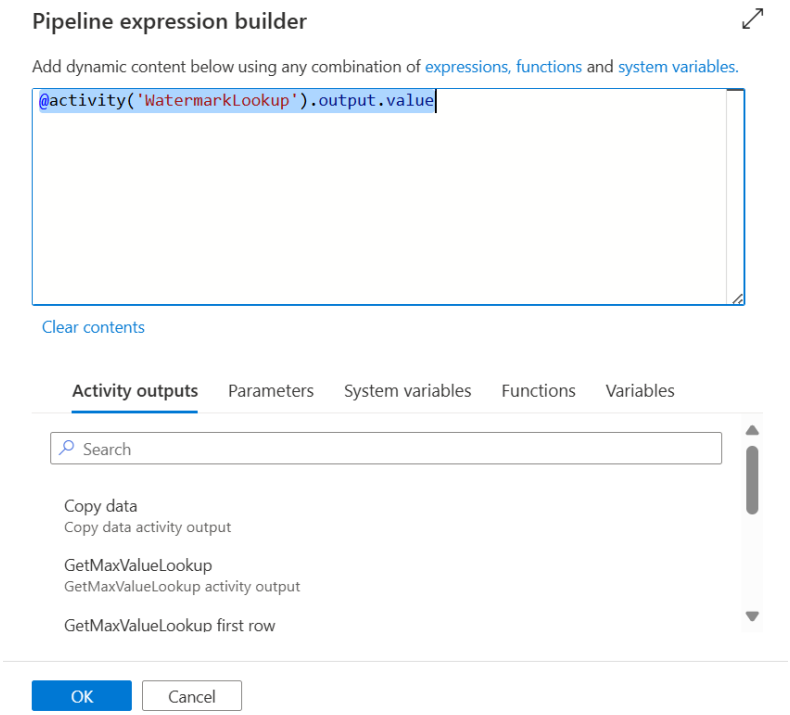


– Set the **Items** option dynamically as a **lookup activity output value** by **clicking** on **Add Dynamic content**.

lookup activity output value expression:

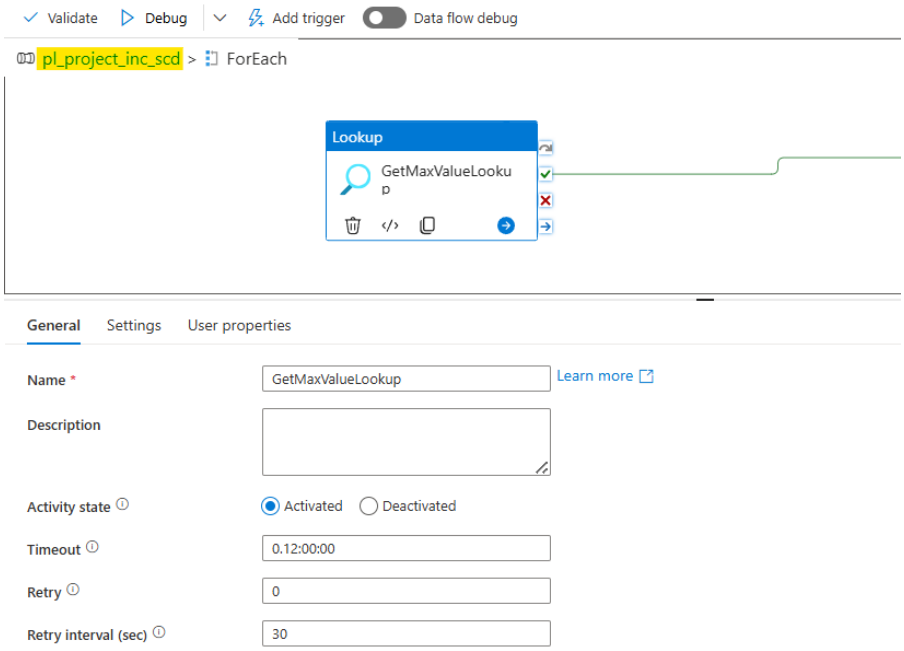
@activity('WatermarkLookup').output.value



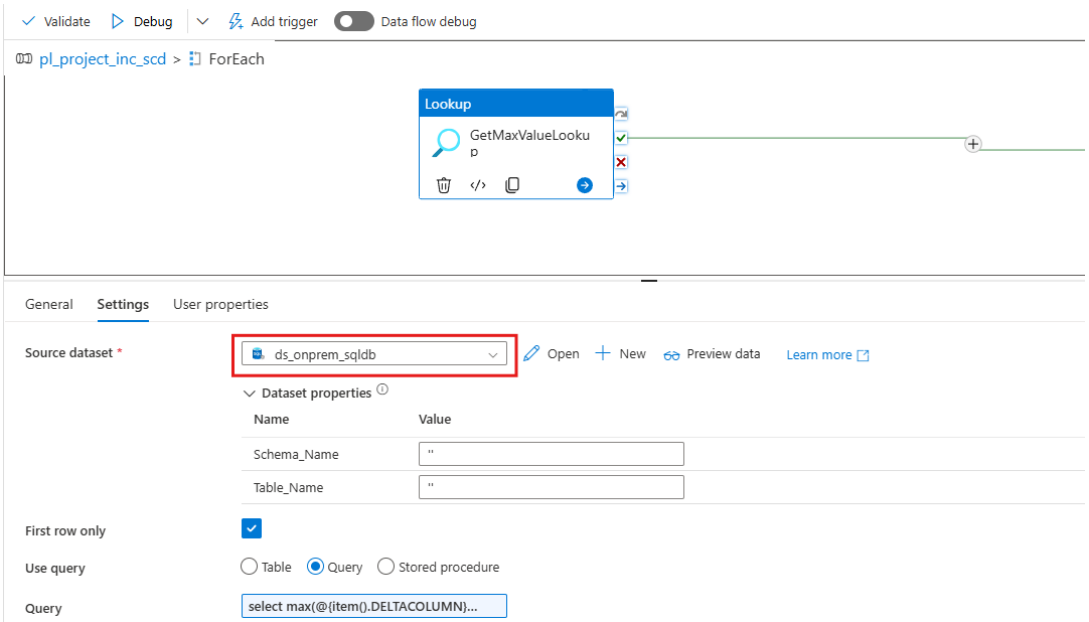


STEP 3: Set Lookup Activity in ForEach

- Move into the Foreach activity using pencil Icon, and add one more **Lookup activity inside the Foreach** to fetch the **max value** from the **source tables**.



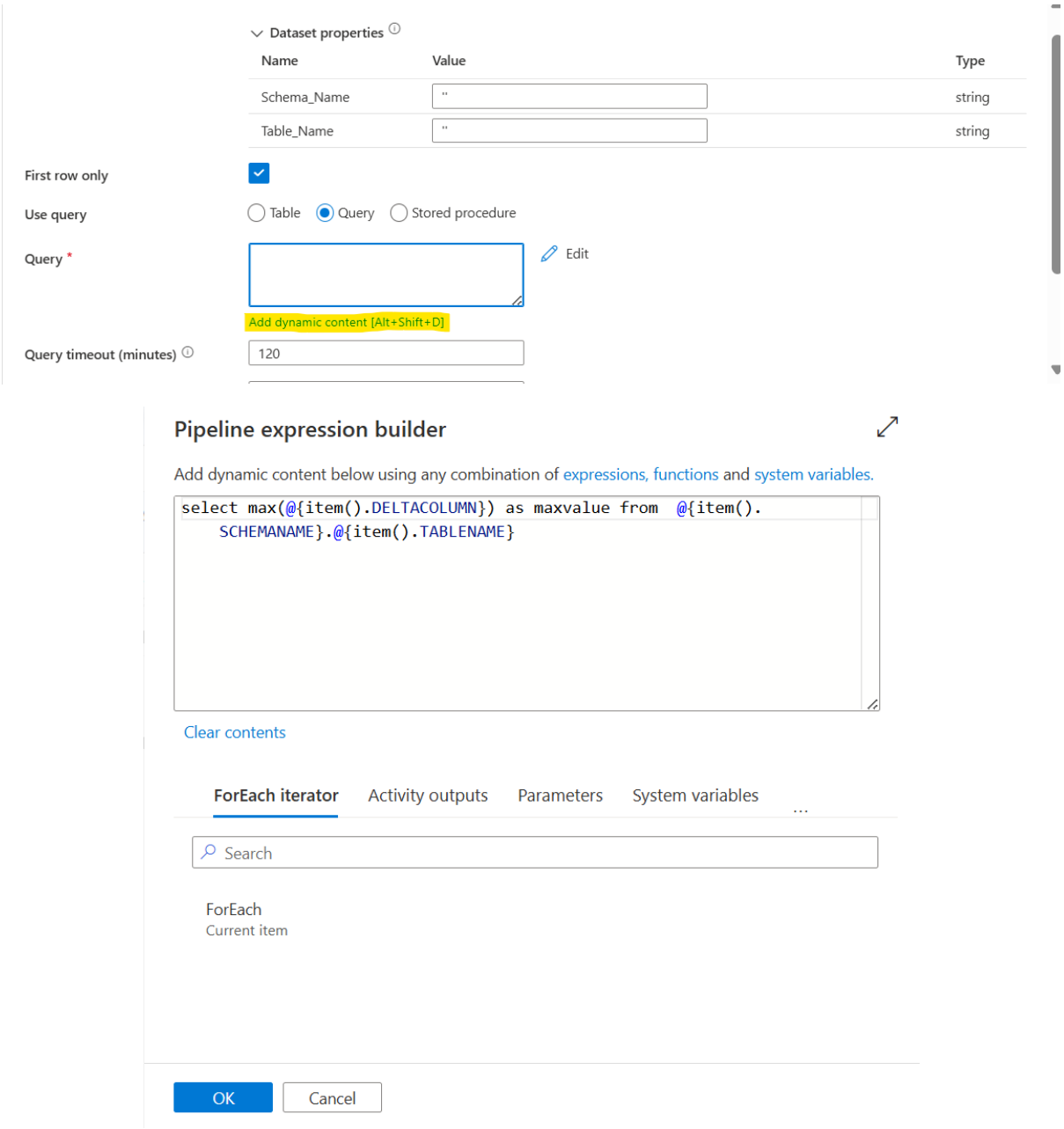
– Set the **same SQL DB dataset** again as the source dataset and give **quotes** as input for **schema and table name parameters** as we use the **dynamic query** method to **fetch the max value** from source tables.



– Use the **delta column** and **max function** in the query to fetch the max value from the source DB.

Dynamic Query Expression:

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

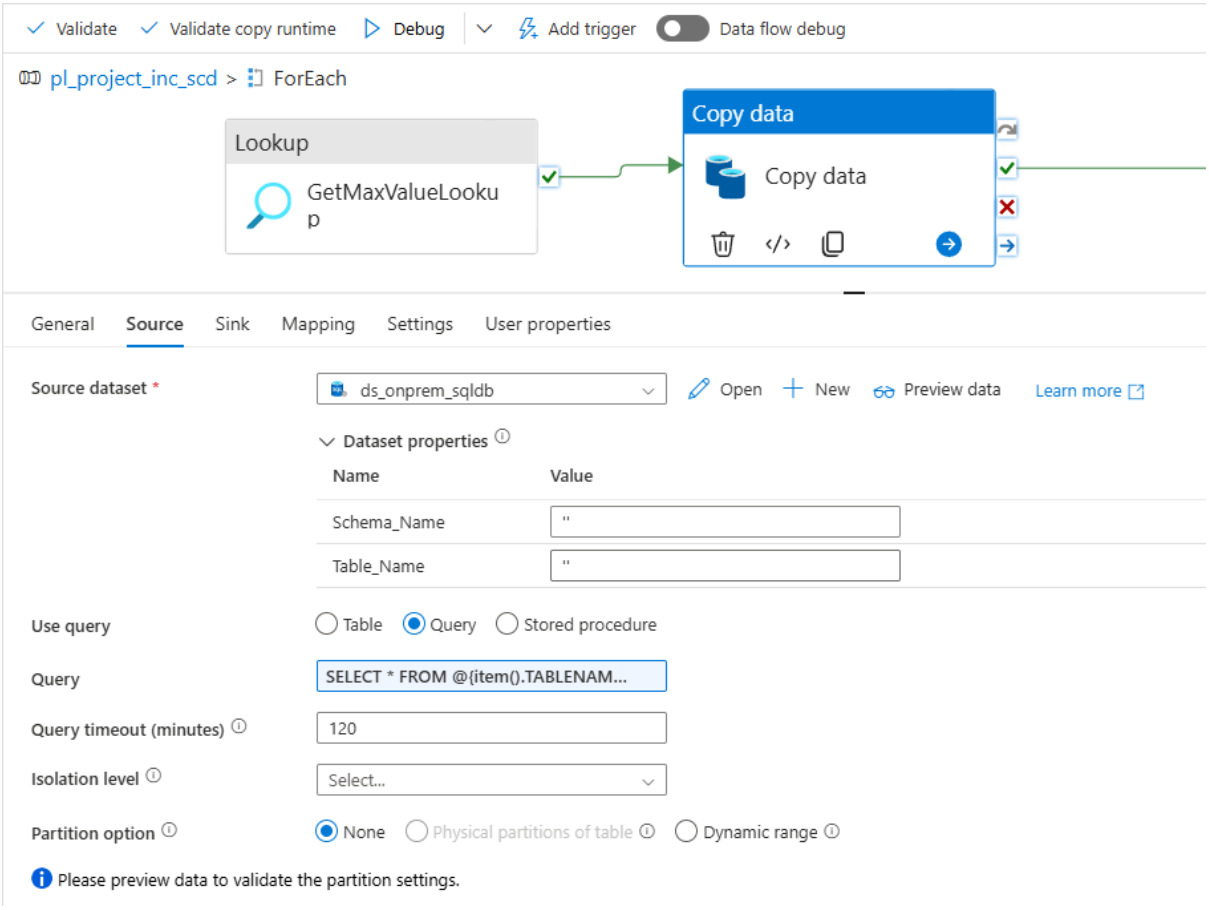


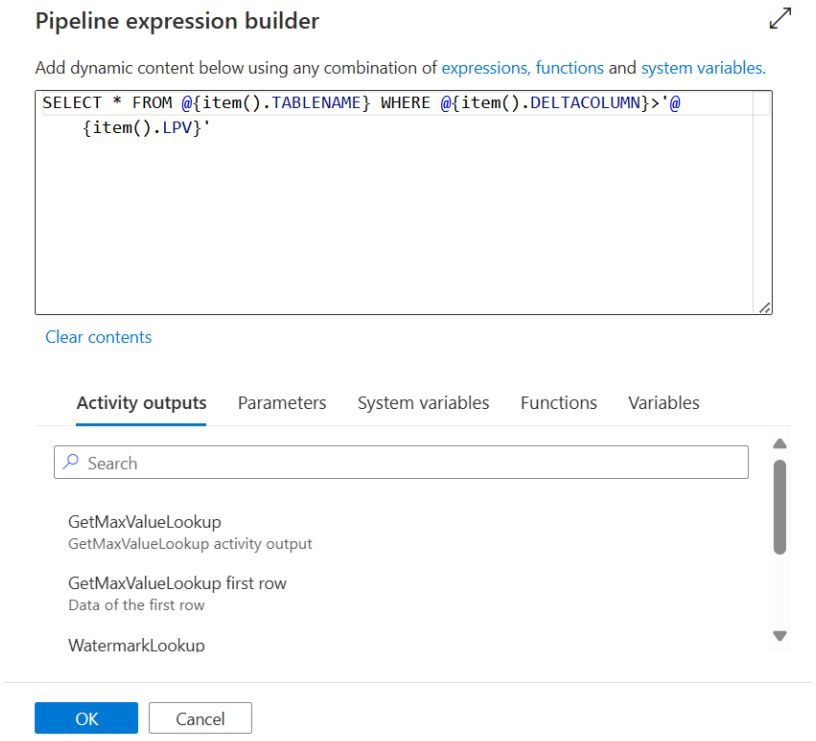
STEP 4: Set Copy Activity

- Add Copy Activity and connect it with on success node of Get max Lookup Activity.
- Move to **source tab** in copy activity, set **source dataset** for copy activity by selecting same **previously created SQL Server DB dataset**. **Give quotes** as input for **schema** and **table name parameters** as we use the dynamic query method to fetch table names whose **delta column** has a **greater value stored** in the **last processed value**.

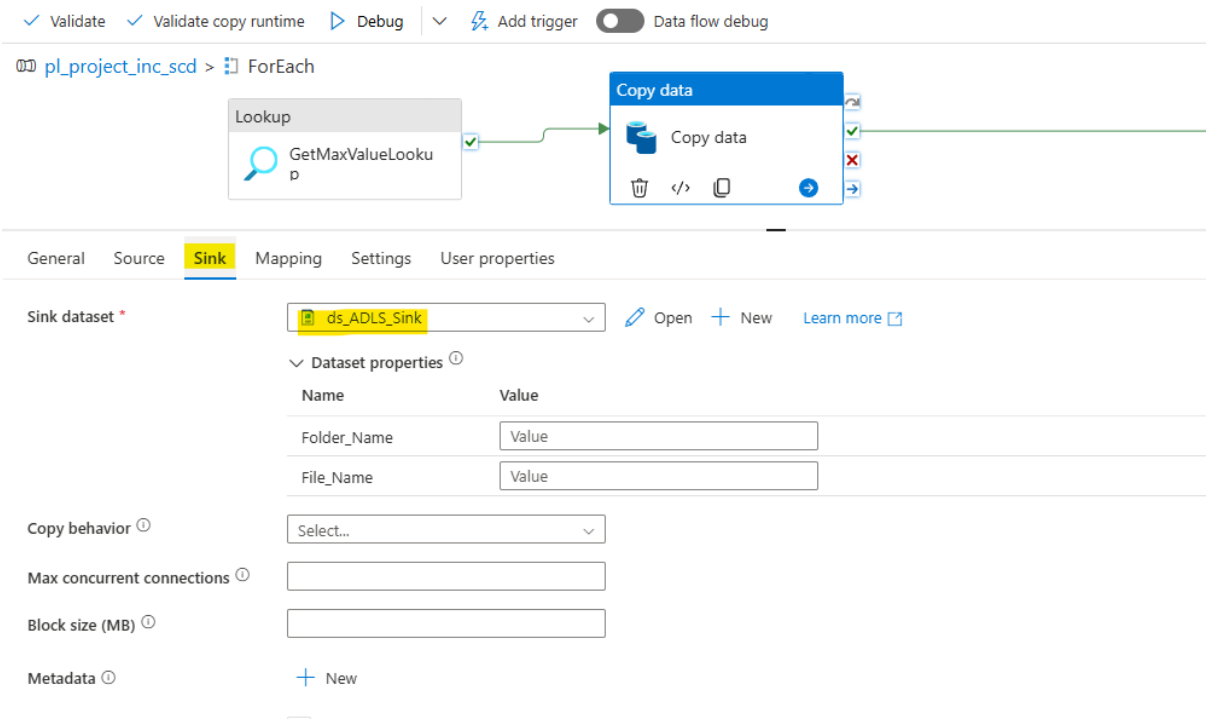
  Query Expression:

SELECT * FROM @{item().TABLENAME} WHERE @{item().DELTACOLUMN}>'@{item().LPV}'

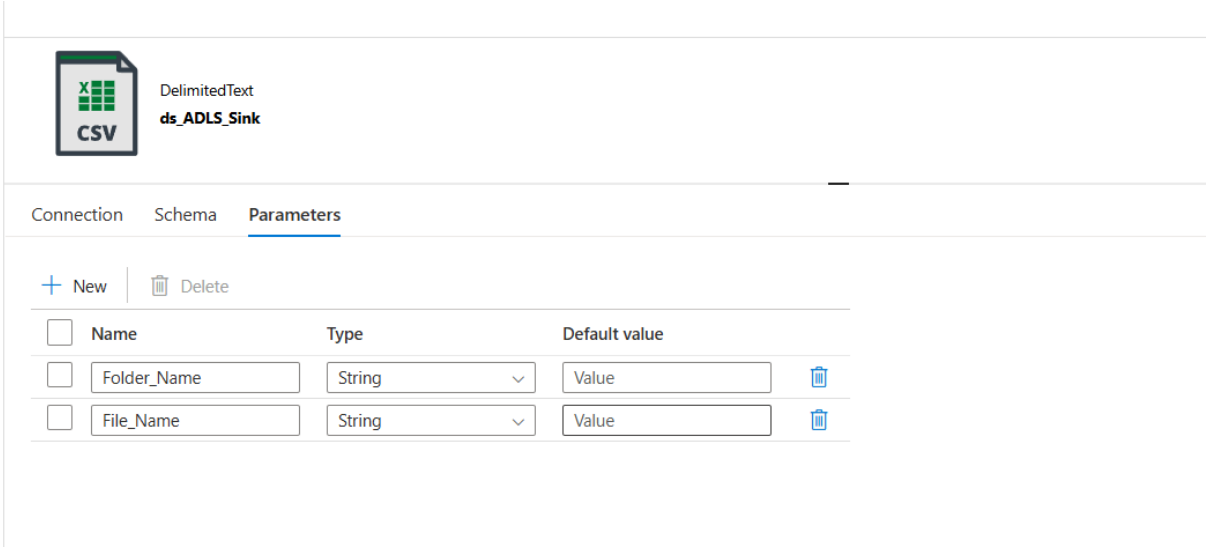


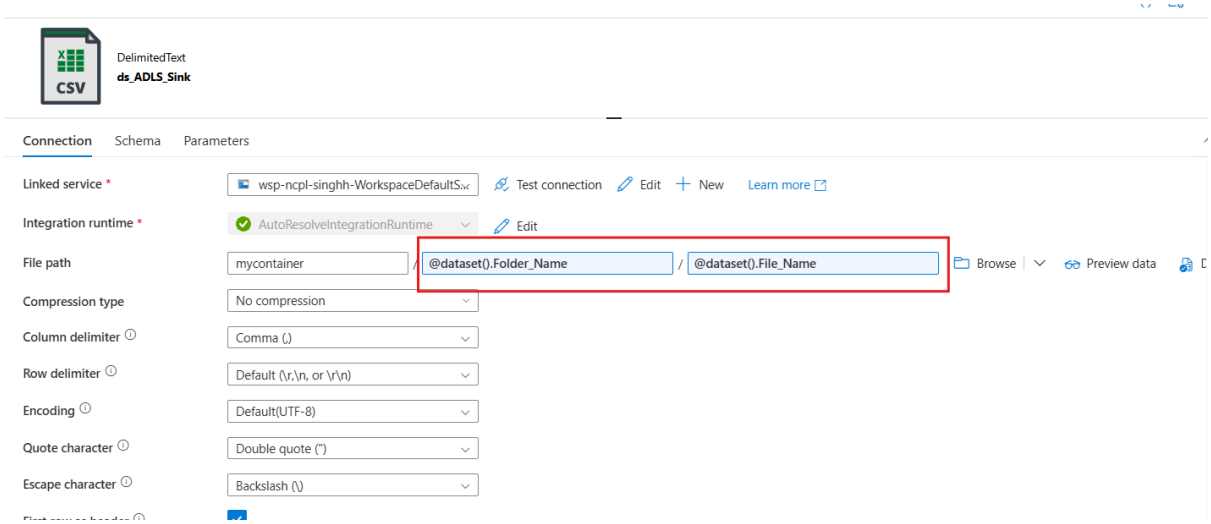


– Move to Sink tab in Copy Activity, and set the Sink dataset by creating new Data set or selecting the existing one. Here, I have selected previously created Sink Data set ADLS gen2 storage and file type as CSV.



– Create the parameters **by opening** the **dataset Activity** under the Parameters tab and setting the **folder name and file name parameters dynamically** for the file path under the **sink dataset's connection type.**



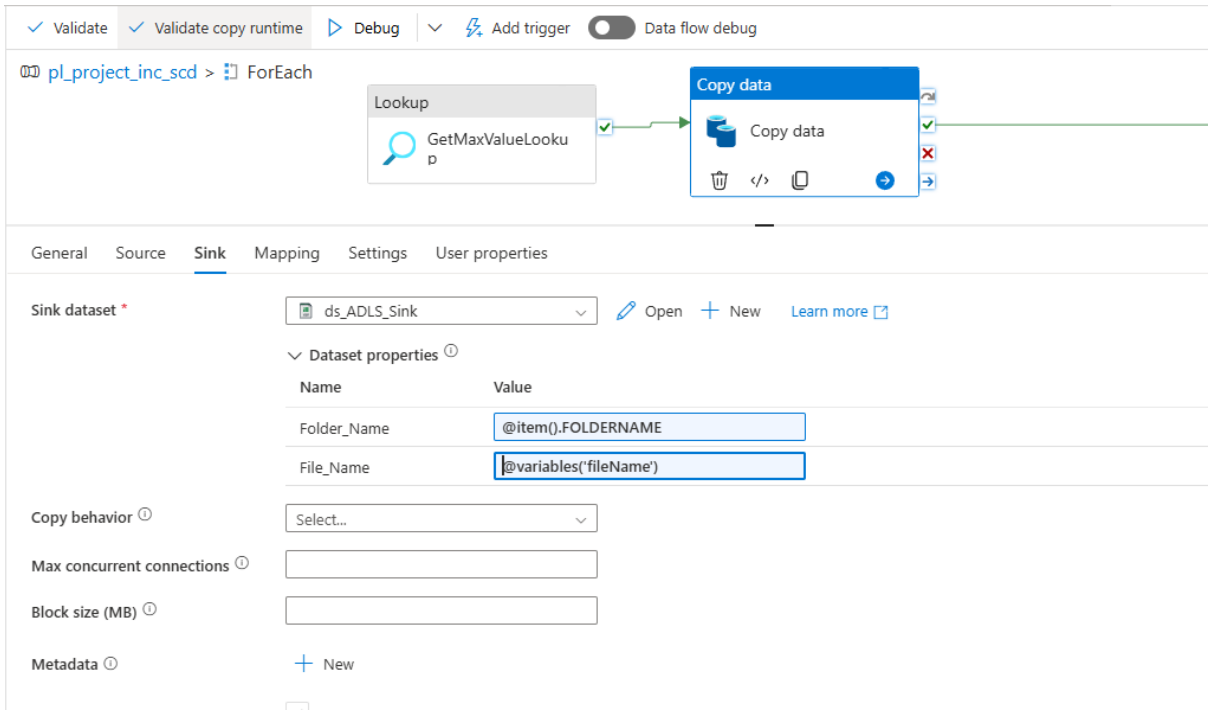


- Then, move to the sink tab of copy data activity and set the Folder Name and File Name Dynamically using foreach activity Items.
- Using already created **pipeline variable value** for **filename**.

  For Folder Name simply by: **@item().FOLDERNAME

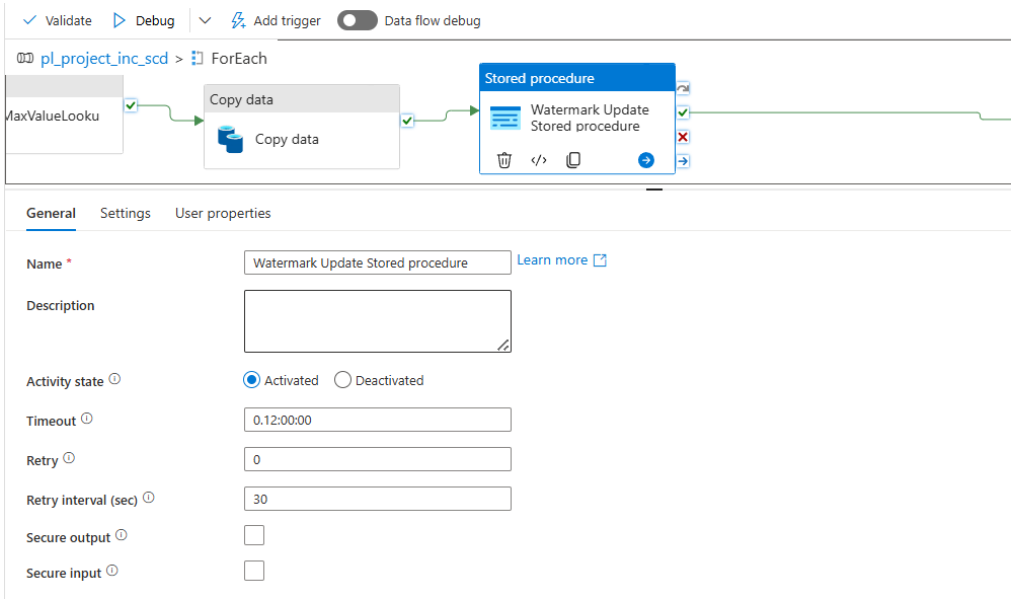
  For File Name use: **@variables('fileName')

  Here, ***utcNow()*** function is used to get current date time from the system.

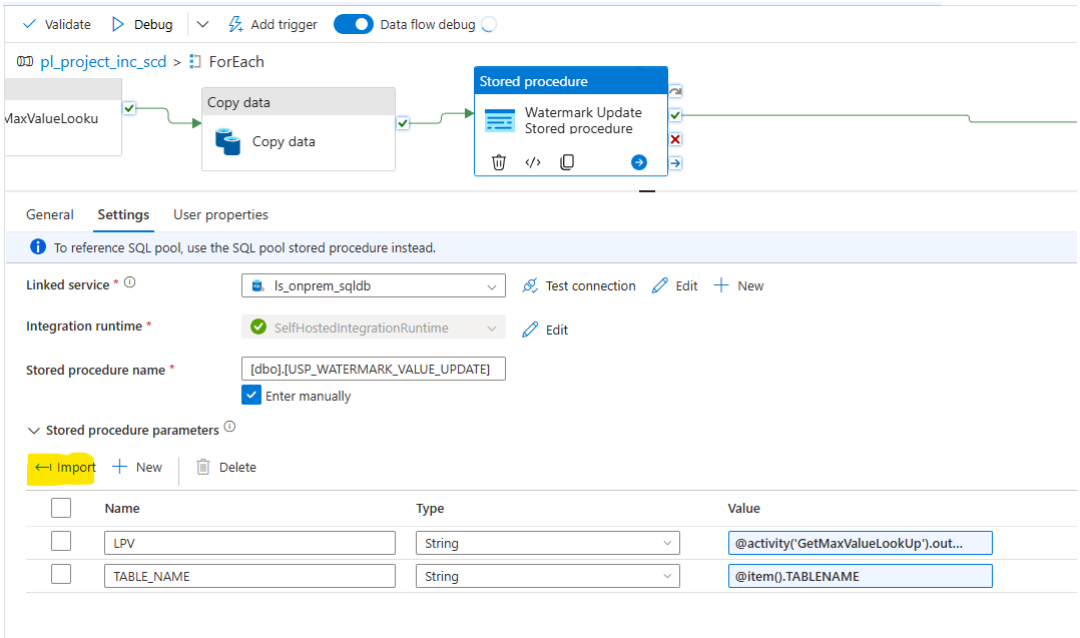


STEP 5: Set Stored Procedure Activity

- After the above step, Add the Stored Procedure Activity to use the store procedure created in DB to **update the LPV column of the watermark table with the latest max value.**



- Move to the Settings tab and set Linked Service by selecting previously created **linked service** and select the created stored procedure name. Then, click on the **Import** option to load the stored proc parameters.



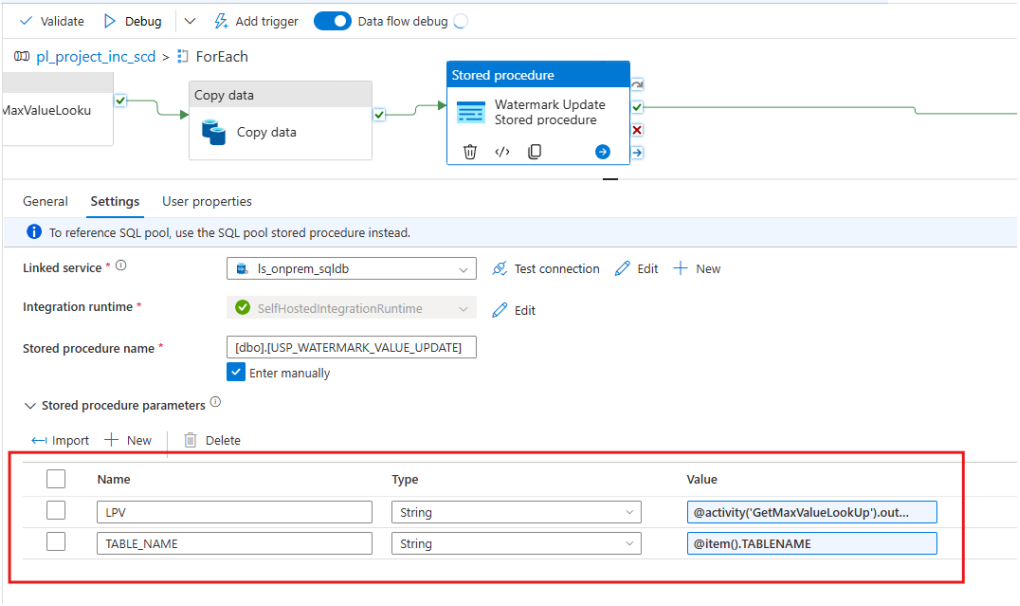
– Pass the **max value** from the getmaxvaluelookup activity to set the LPV parameter value in order to update it in the watermark table. Similarly, pass the table name value using the item table name from foreach activity.

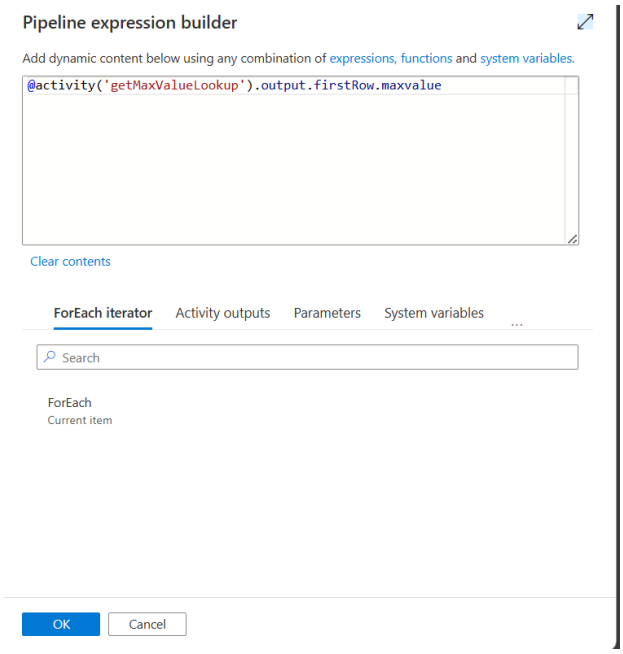
LPV Expression Values:

@activity('GetMaxValueLookUp').output.firstRow.maxvalue

TABLE_NAME Expression Values:

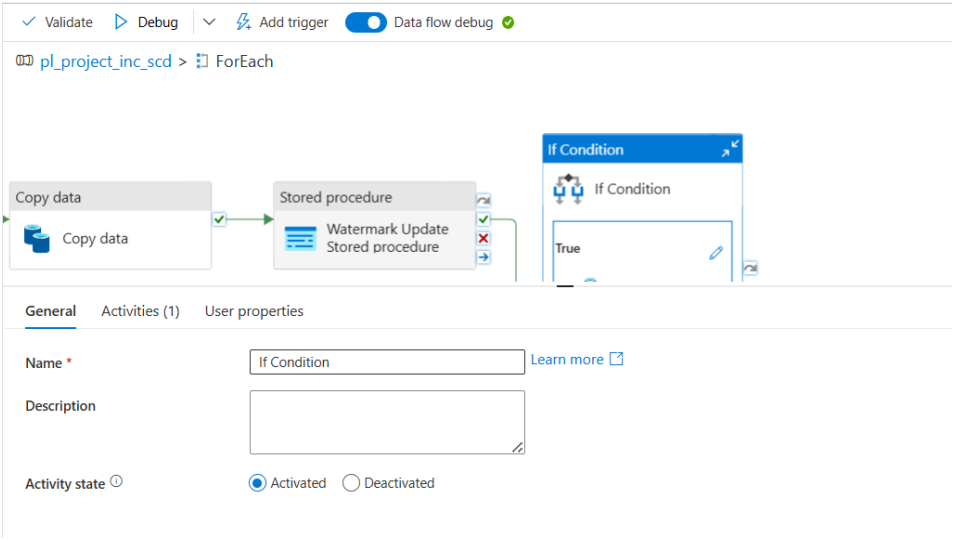
@item().TABLENAME





STEP 6: Set IF Condition Activity

- Next, Add the IF Condition activity and connect it with the store procedure activity using the on-success connection.

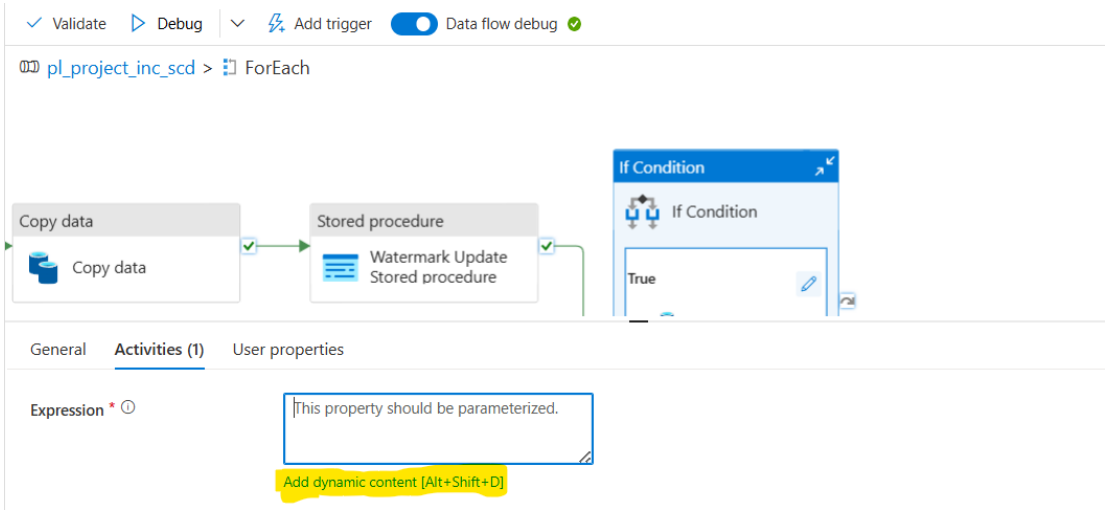


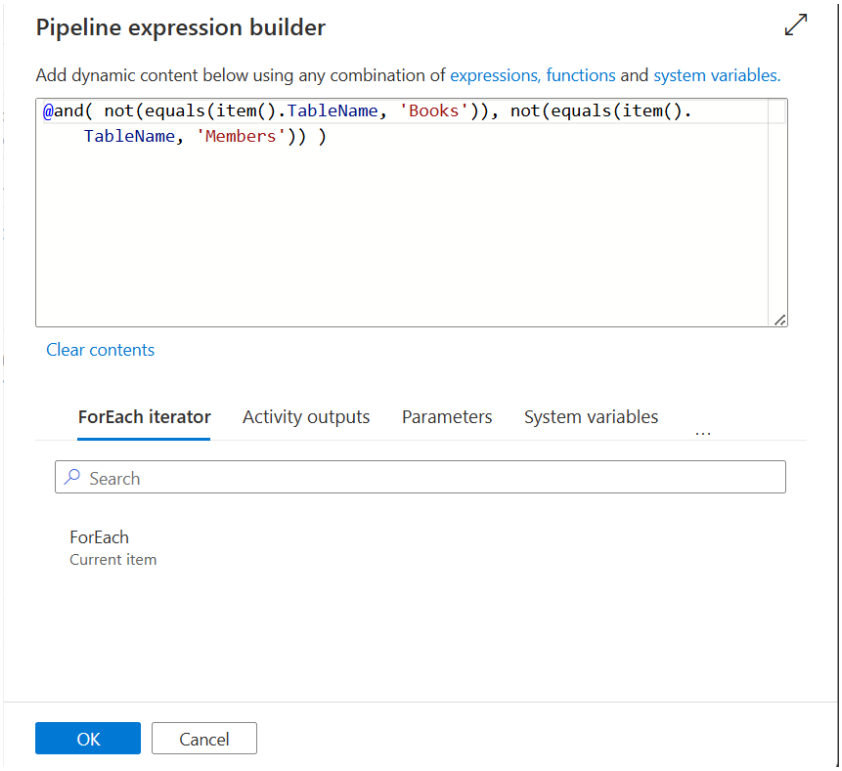
- Then, Go to Activities tab of IF Condition, set the Expression by adding it using Add dynamic content option as highlighted below.
- This IF Condition only allows the non-SCD Type tables to normally load generated csv Files data to Azure SQL Database from ADLS Gen2.

  If Condition Expression for True case:

  @and(not(equals(item().TableName, 'Books')), not(equals(item().TableName, 'Members')))

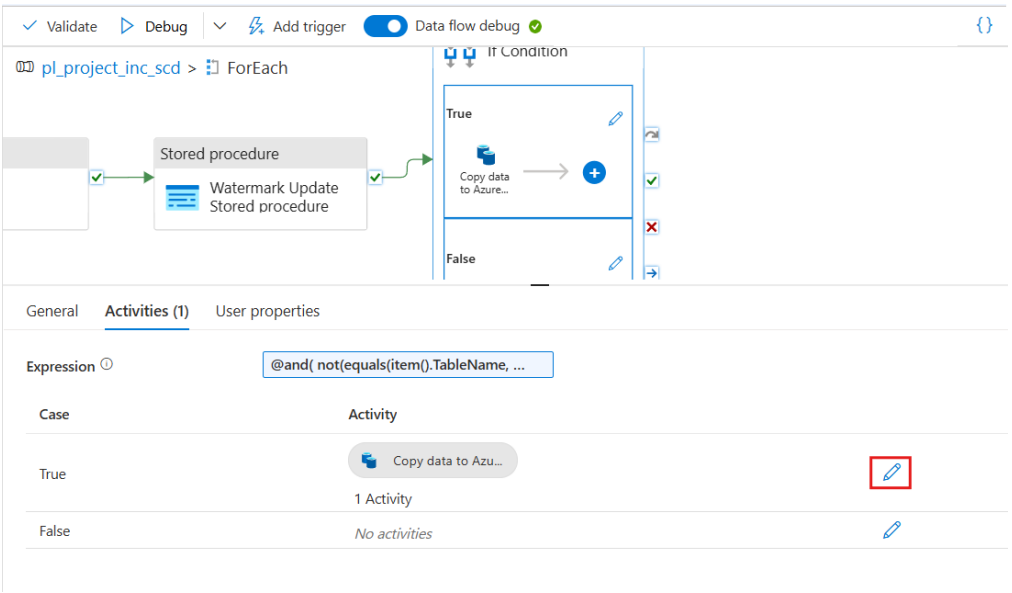
- The above condition checks if the table name is **not Books and Members** then only pass the condition and go for copy activity to copy other tables data (**BorrowRecords, Inventory, and Librarians**).



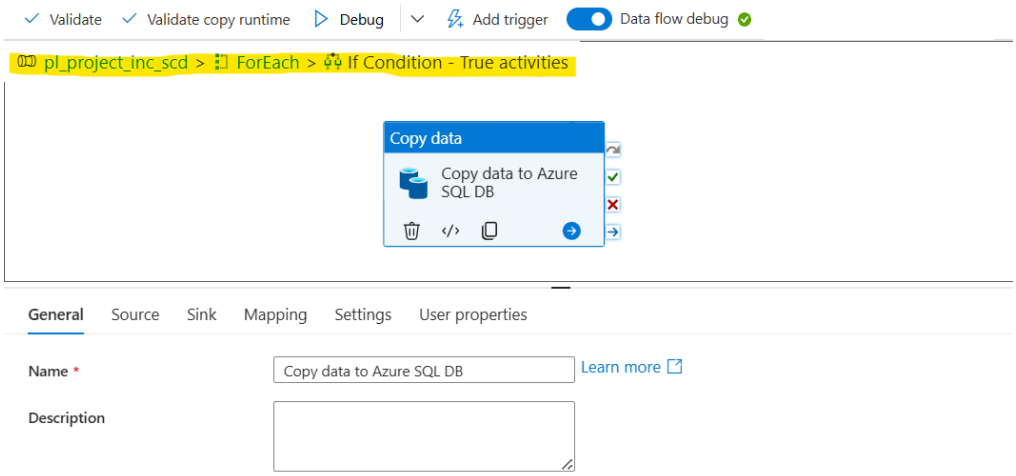


STEP 7: Set Copy Activity in IF Condition True Section

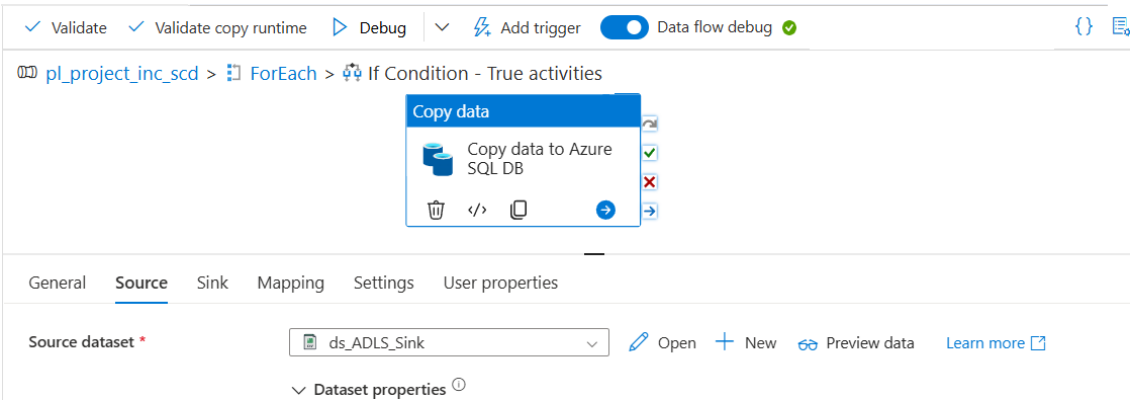
- Next, Add the Copy Data activity in IF Condition's **true section**. Go to **Activities tab** of **IF Condition activity** and click on pencil icon to add more activities in True section.



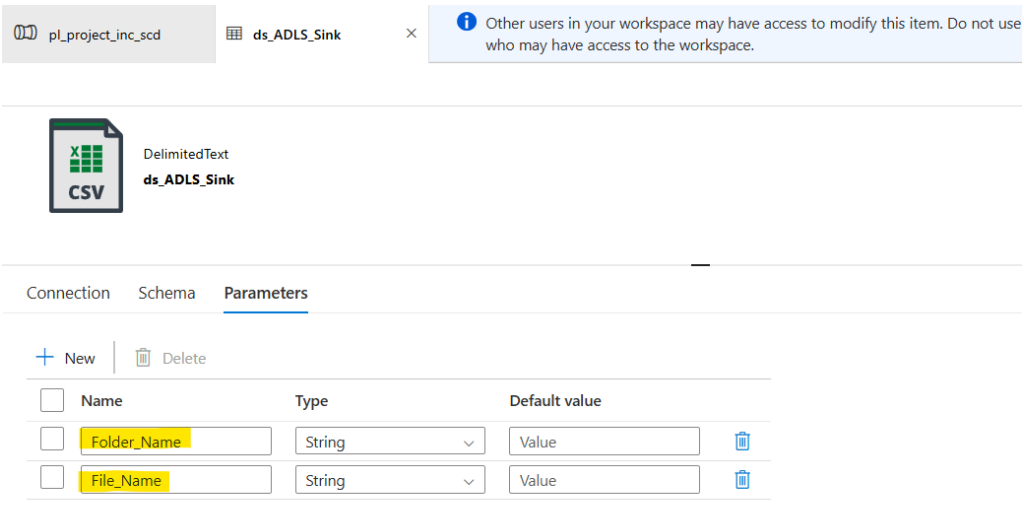
- After Going to true section, Add Copy Activity in it.



– Move to **source tab** in copy activity, set **source dataset** for copy activity by selecting or creating new dataset for **ADLS gen2** storage as CSV files. Here. I have selected the already created dataset for ADLS Gen2.



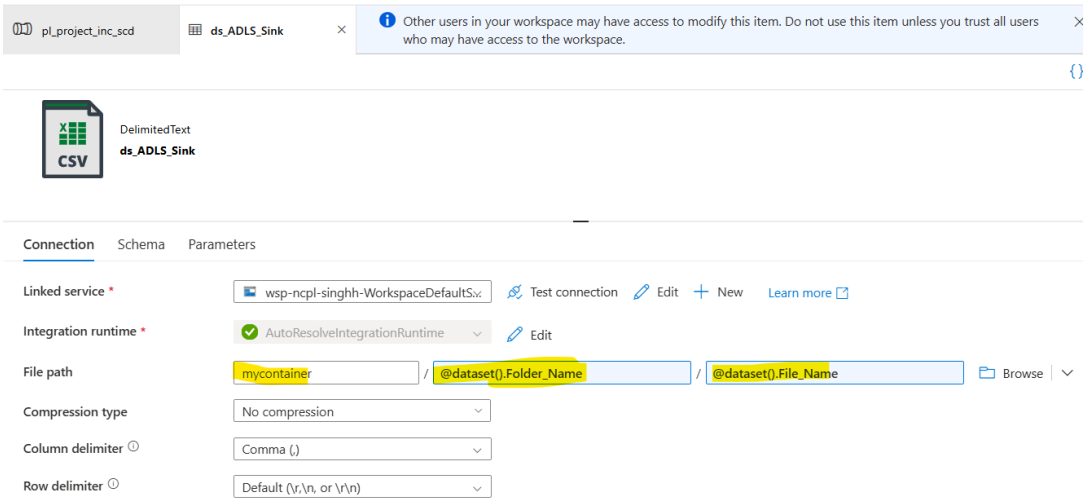
– Open the Data set, Go to Parameters tab and create 2 new parameters.



– Go to the **connection tab of source data set**, and select the container from file selection and give the Dynamic Expression for Folder and File Name Dynamically as shown below.

Folder_name expression: ***@dataset().Folder_Name***

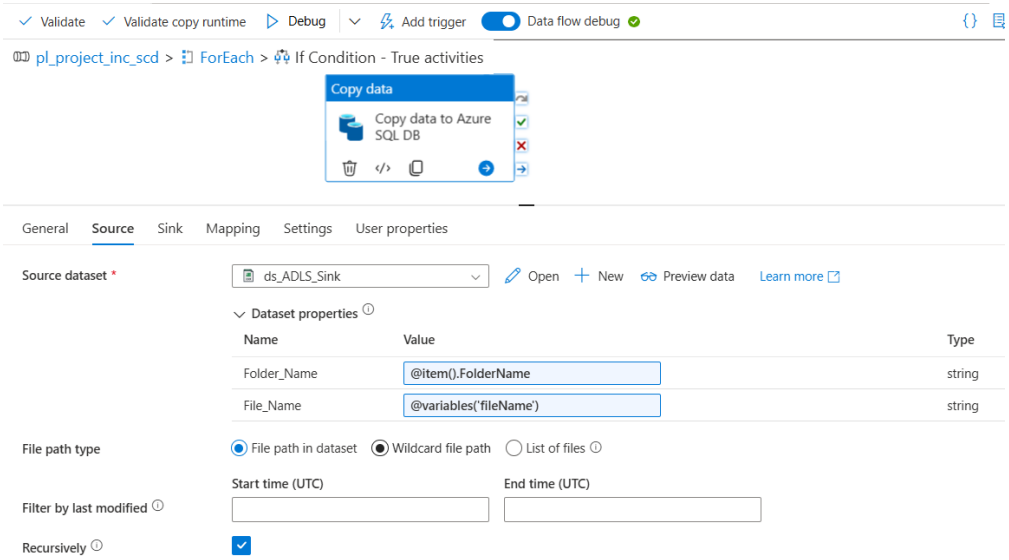
File_Name expression: ***@dataset().File_Name***



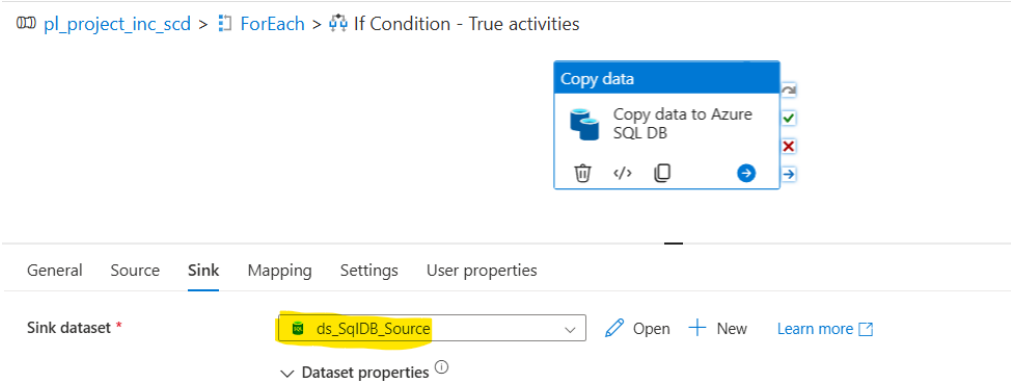
– Next, move to the **Source** tab of copy data activity, and select the container from file selection and give the Dynamic Expression for Folder and File Name Dynamically as shown below.
– Using already created **pipeline variable value** for **filename**.

Folder_name expression: ***@dataset().Folder_Name***

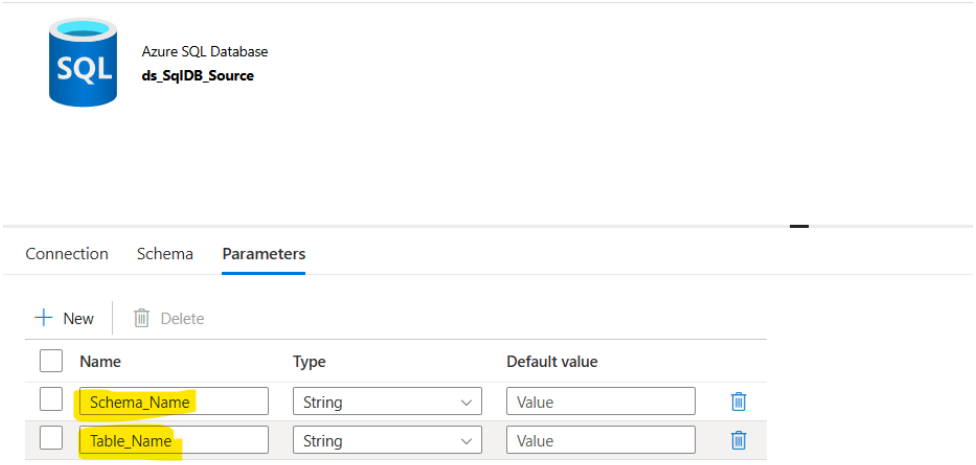
File_Name expression: ***@variables('fileName')***



– Next, Go to Sink Tab -> set **sink dataset** for copy activity by selecting or creating new dataset for Azure SQL Database. Here, I have selected the already created dataset for Azure SQL database.

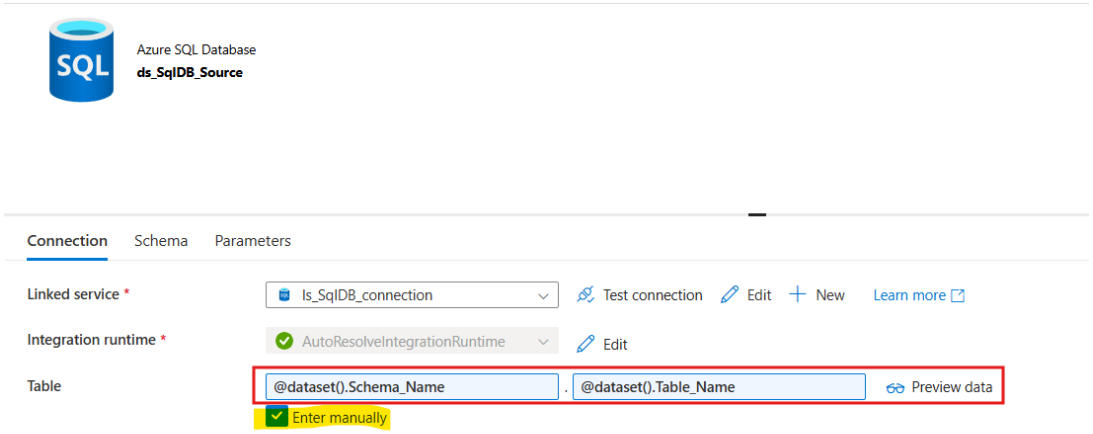


– Then, Open the sink Data set, Go to Parameters tab and create 2 new parameters.



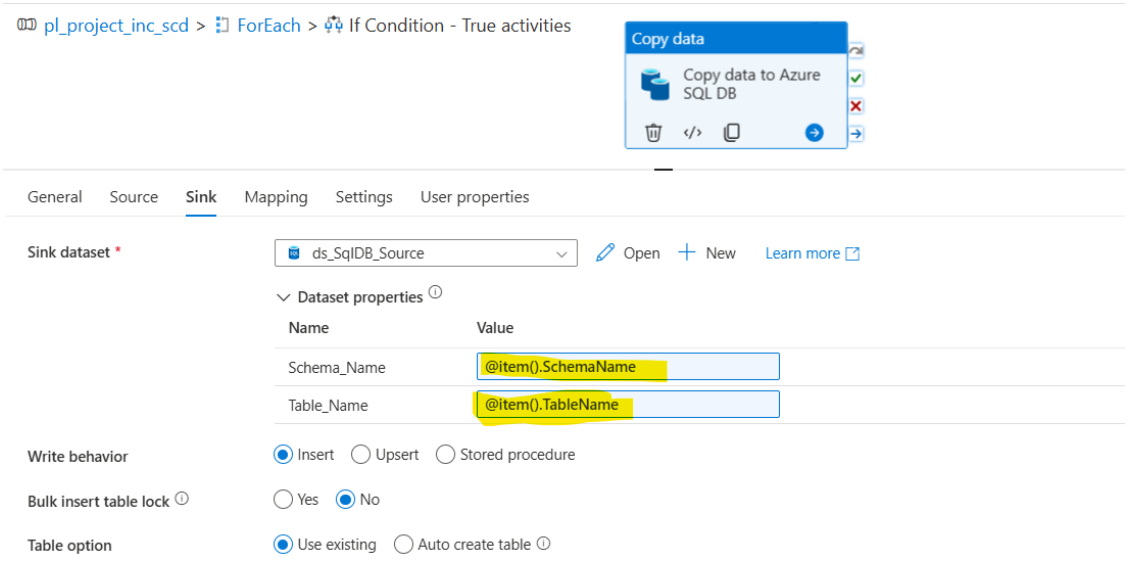
– Go to the **connection tab of sink data set**, and select the **Enter Manual** Option for Table and enter the mentioned Schema name and Table Name Dynamically as shown below.

Schema_Name expression: ***@dataset().Schema_Name***
Table_Name expression: ***@dataset().Table_Name***



– Next, move to Setting Tab of **copy Activity**, enter the mentioned Schema name and Table Name for the appeared parameters as shown below.

  Schema_Name Expression: ***@item().SchemaName***
  Table_Name Expression: ***@item().TableName***



– Finally, the Copy Activity part is done, and **move out** to the **if condition** and **for each activity.**

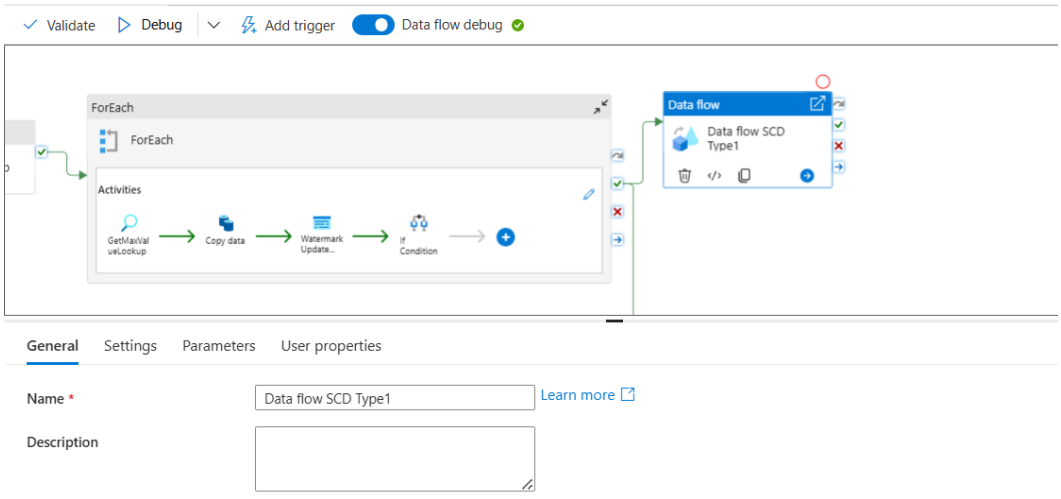
STEP 8: Set Data flow Activity for SCD Type 1

– Here, I implemented SCD Type 1 data load for table **Books** to load the data in **Azure SQL DB** from **ADLS Gen2.**
– To load the **Books** Data in Azure SQL DB, we need to create table definition in **Azure SQL DB database** as shown below:

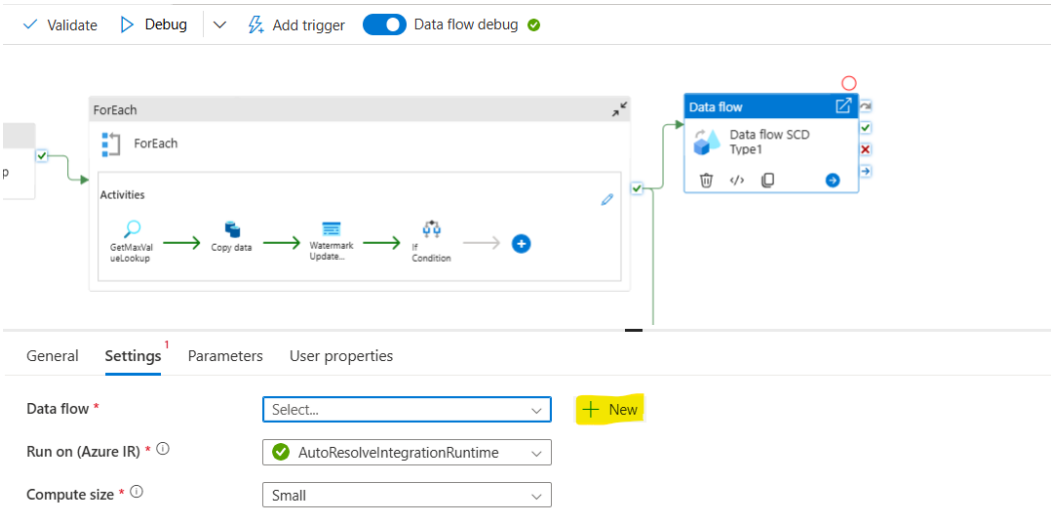
```
Create Table Books_SCDTYPE1
(
        bookId Int,
        bookTitle Varchar(255),
        bookAuthor Varchar(255),
         bookGenre Varchar(100),
        publishedYear Int,
        createdBy varchar(100),createdDate datetime,
        updatedBy varchar(100),updatedDate datetime,
        hashkey Bigint,
)
select * from Books_SCDTYPE1
```

```
Create Table Books_SCDTYPE1
(
    bookId Int,
    bookTitle Varchar(255),
    bookAuthor Varchar(255),
    bookGenre Varchar(100),
    publishedYear Int,
    createdBy varchar(100),createdDate datetime,
    updatedBy varchar(100),updatedDate datetime,hashkey Bigint,
)
select * from Books_SCDTYPE1
```

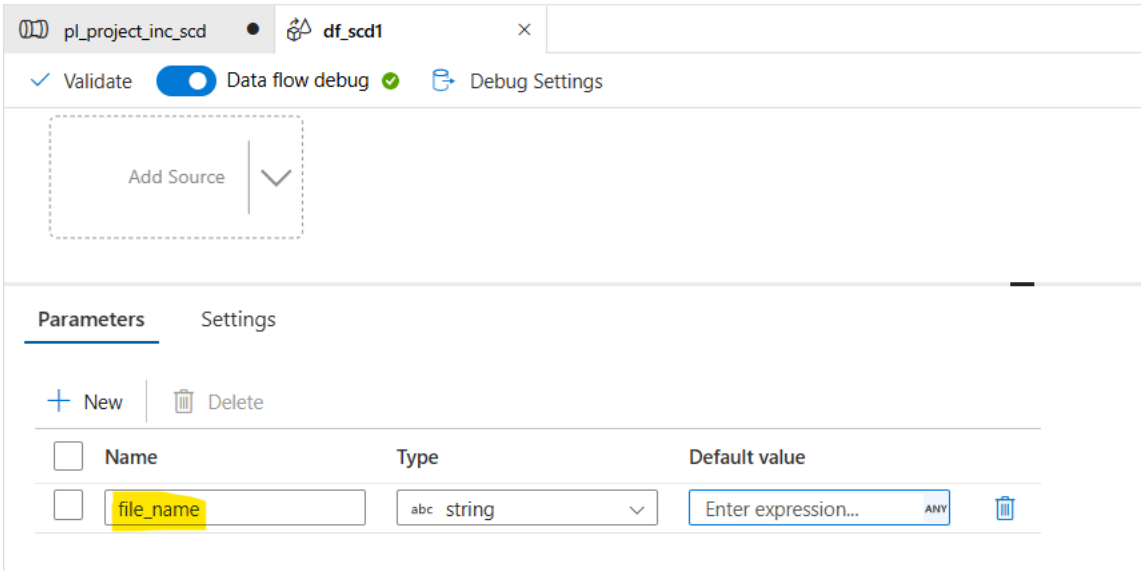
– Drag drag Data Flow Activity and turn on Data flow debug.
Connect it to for each activity with success point.



– Go to Settings Tab of Data flow activity, and create new data flow.

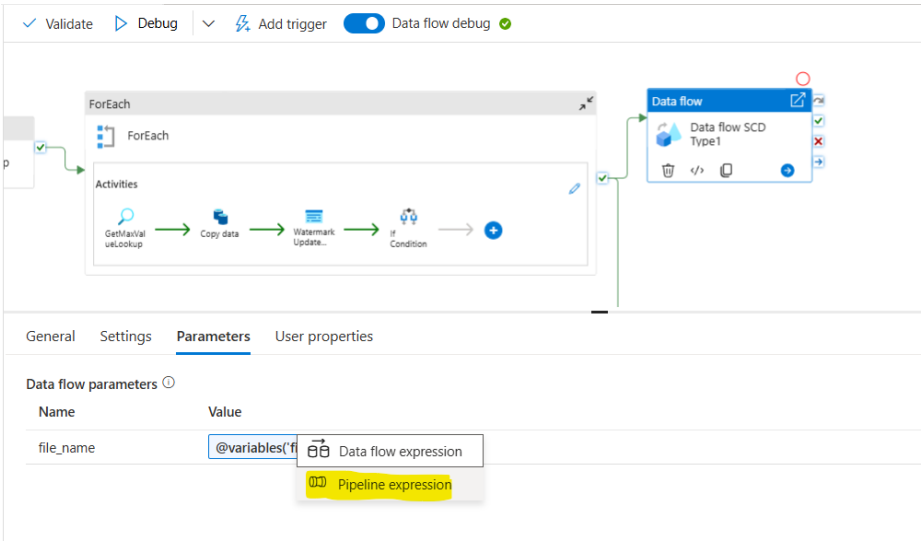


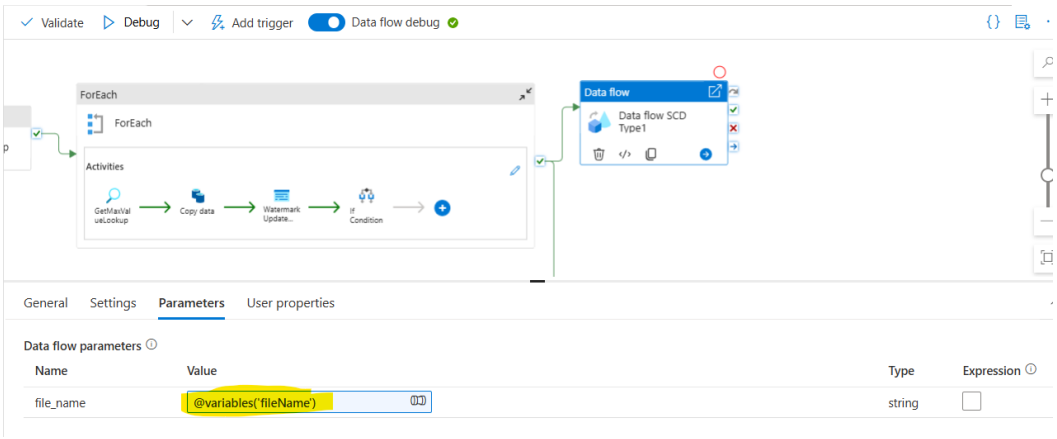
– Open the Created Data flow and create a new parameter named ***file_name*** inside data flow activity.



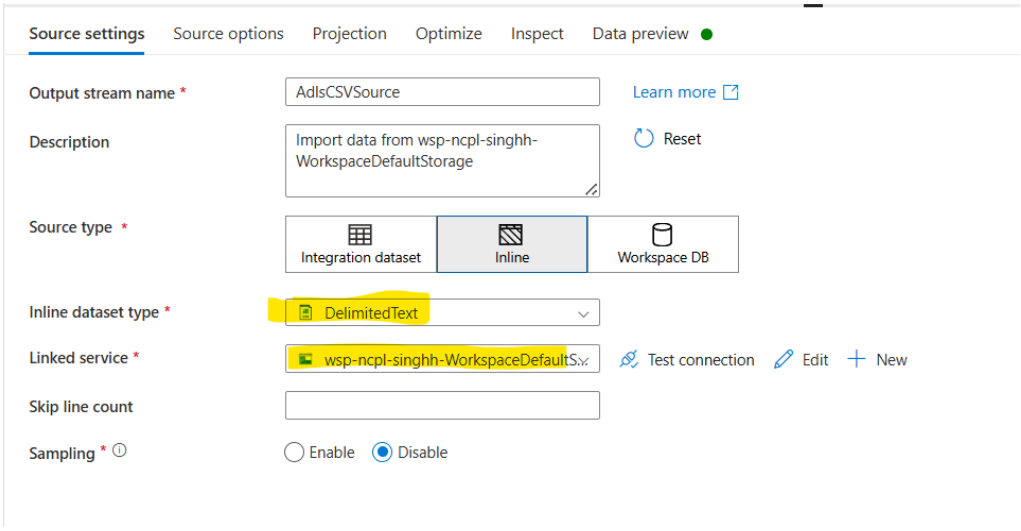
– Then, move to Parameters Tab of Data flow activity in the pipeline, and **create** a new **Parameter** name ***file_name***. Use the Dynamic Expression to set the parameter's value with **previously created pipeline variable** named ***filename.***

Expression Used:
file_name: **@variables('fileName')**

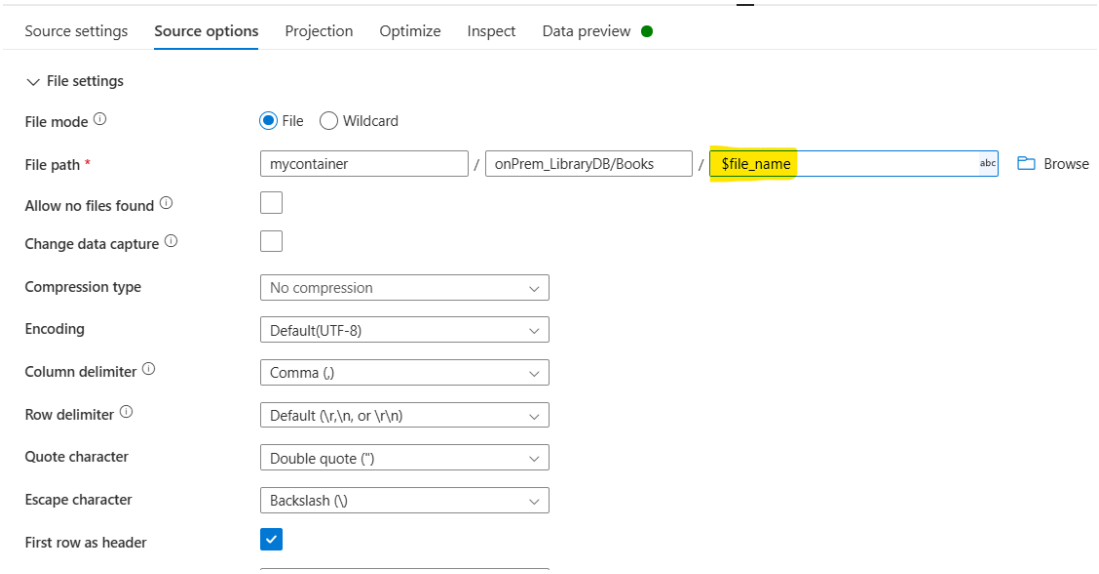


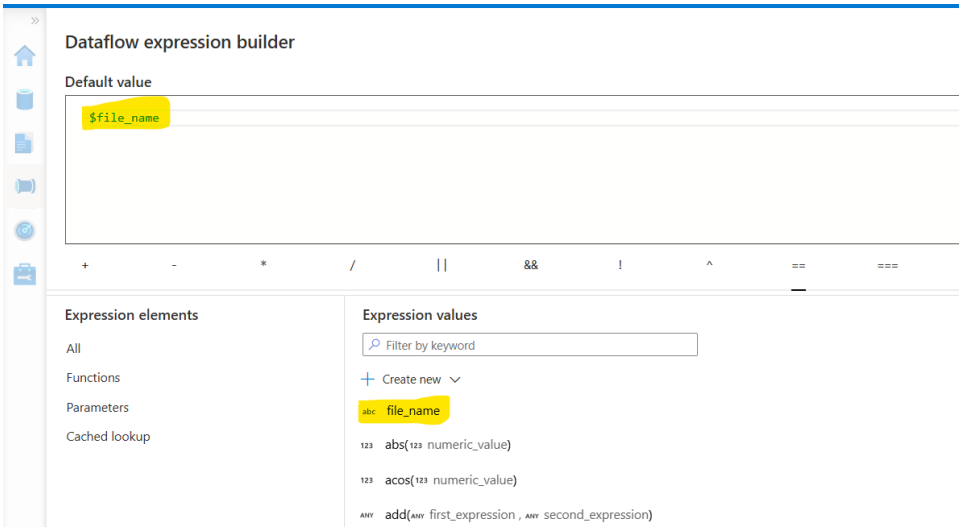


– Next Open the Data flow activity again, and click on **Add source**.
  - Then, Goto Source Settings -> Click on Source Type -> Inline and select the source.

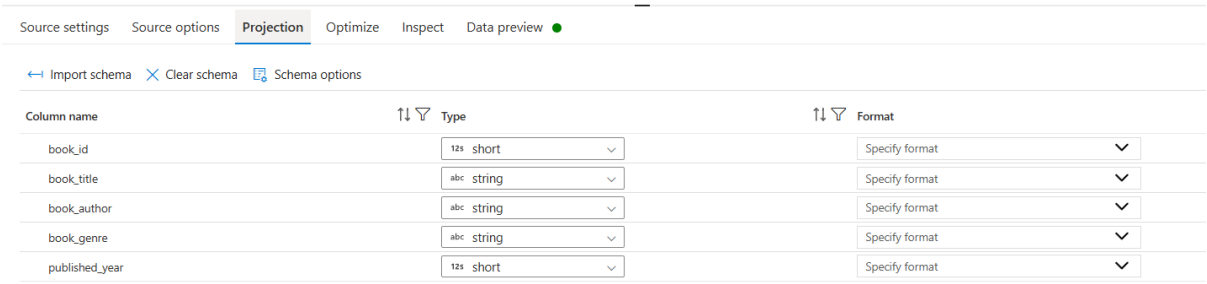


- Next, Go to Source Options -> Select the File from container.
- Here, for the file name we used the parameter value created in data flow. Dynamic Expression Used: ***$file_name***

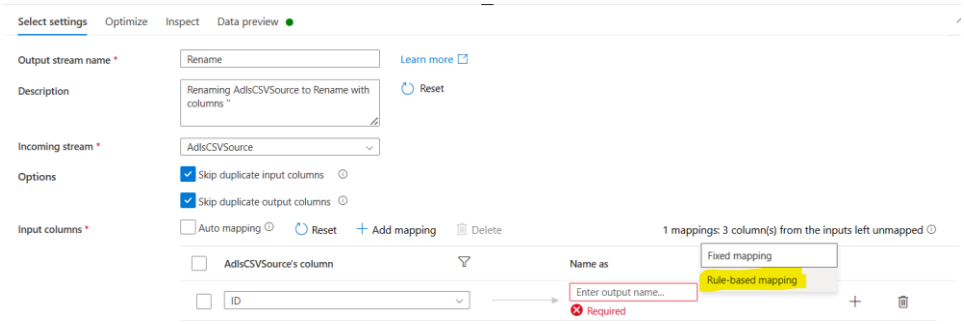


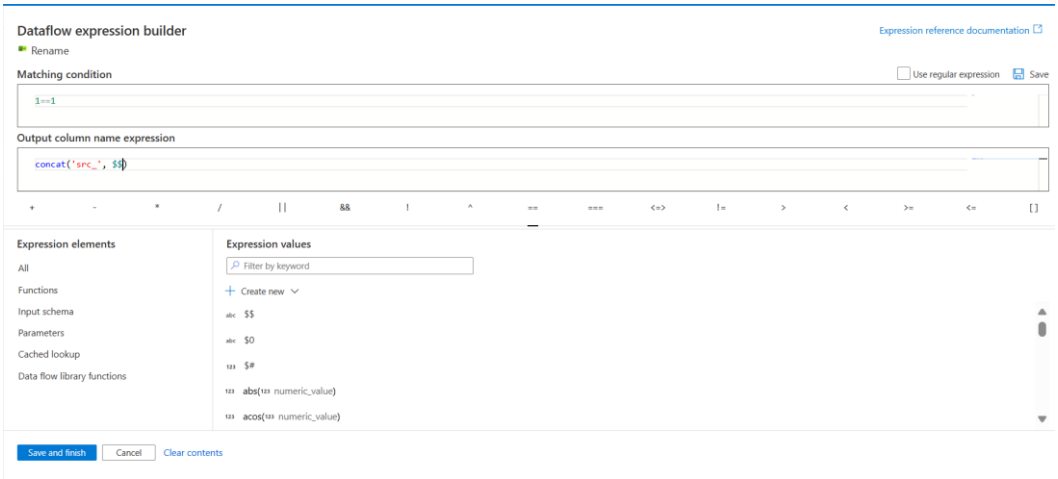


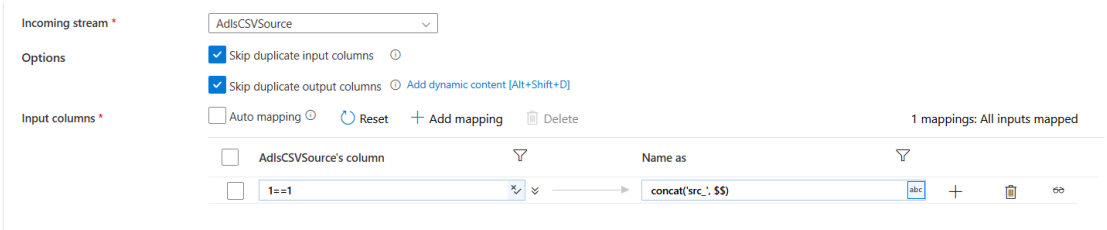
– Click on Projection tab of Source -> click on import schema -> select and review the correct type for columns.



– Add **Select column** as its needed to rename columns with src_columnnames
  - Then, under settings select all the columns and delete the mapping.
  - Next, Click on Add mapping and select rule-based mapping.
  - In the new column, give condition 1==1 to make it true, and give name as concat('src_',$$). This will make dynamic and add src in front of all the columns. Also, remove the above id one row mapping as we don't need it.

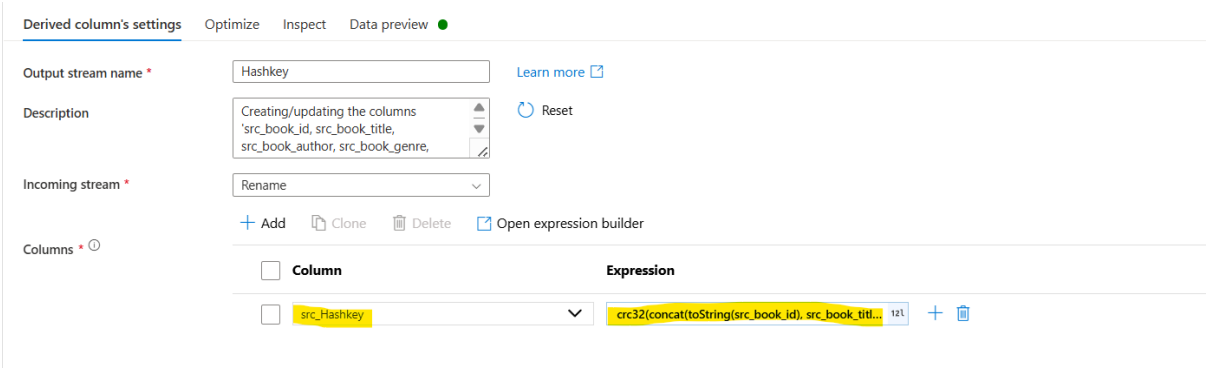


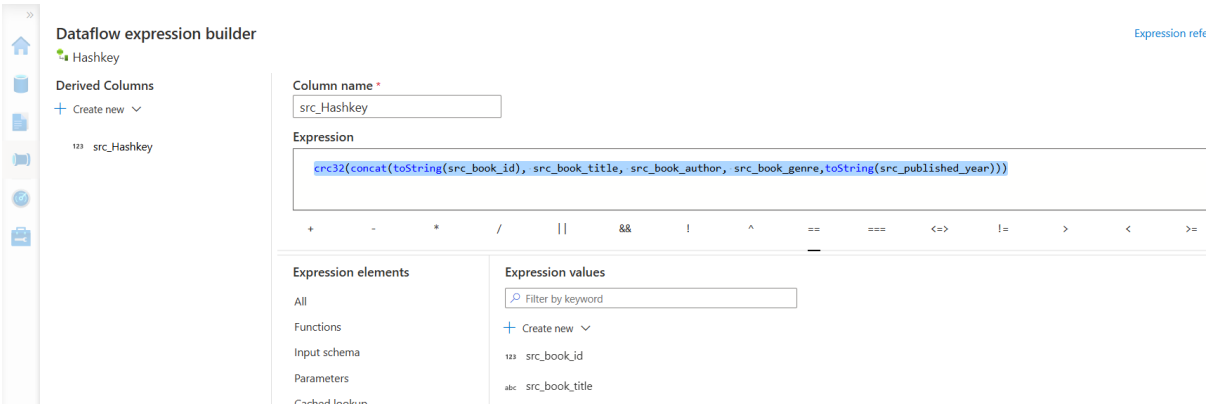




– Add **Derived column**, activity.
  - Under derived column settings, add column src_HashKey and enter this expression.
    crc32(concat(toString(src_book_id), src_book_title, src_book_author, src_book_genre,toString(src_published_year)))

    Note: Crc32 generates hashkey using the mentioned combination.



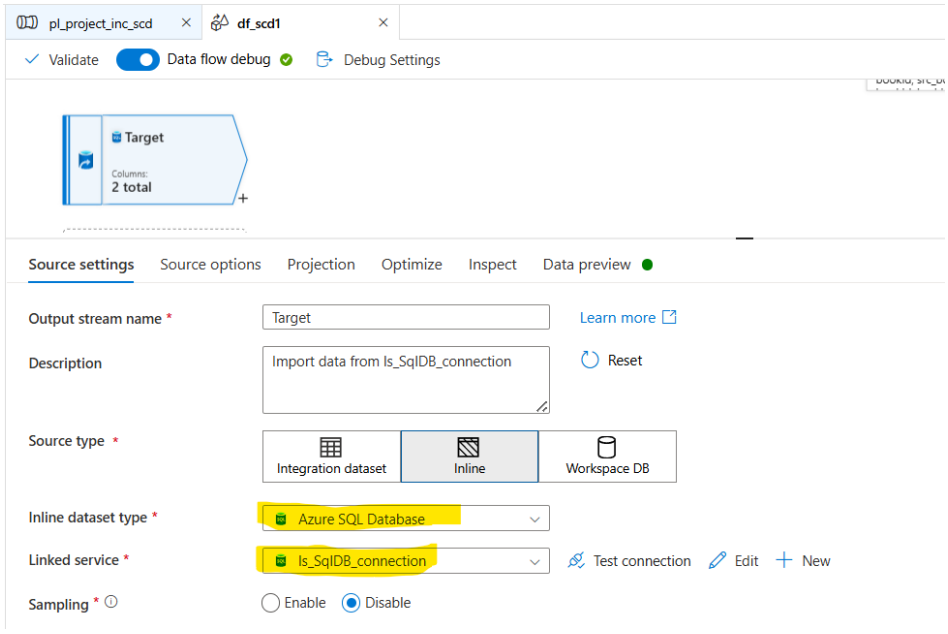


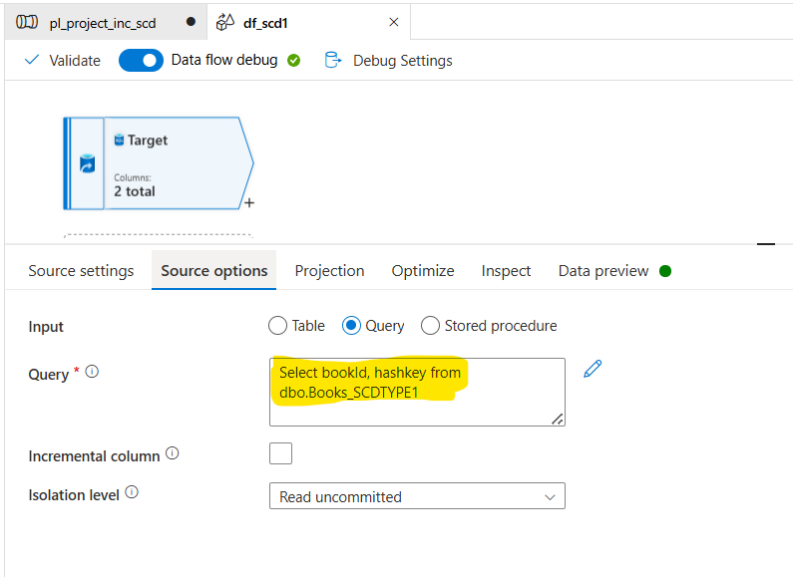
– Add **target** that is **Azure SQL DB** as another source.

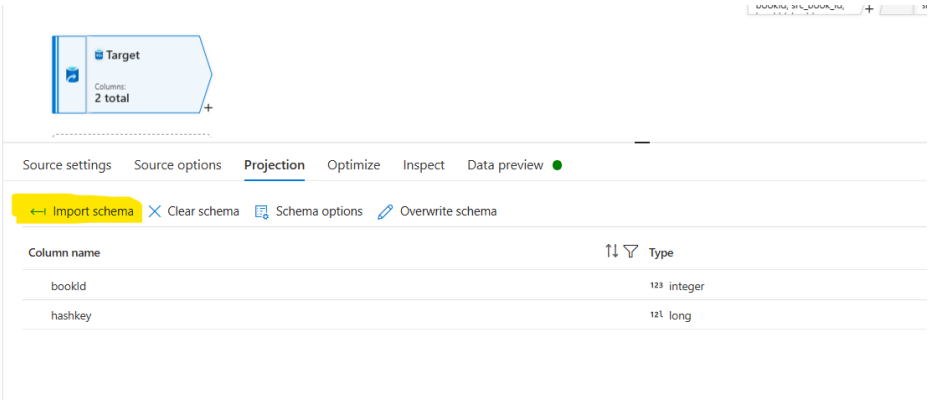
- Select two column **bookId and hashkey** from target table which we will use to check for new records or existing records.

  Query Used: **Select bookId, hashkey from dbo.Books_SCDTYPE1**

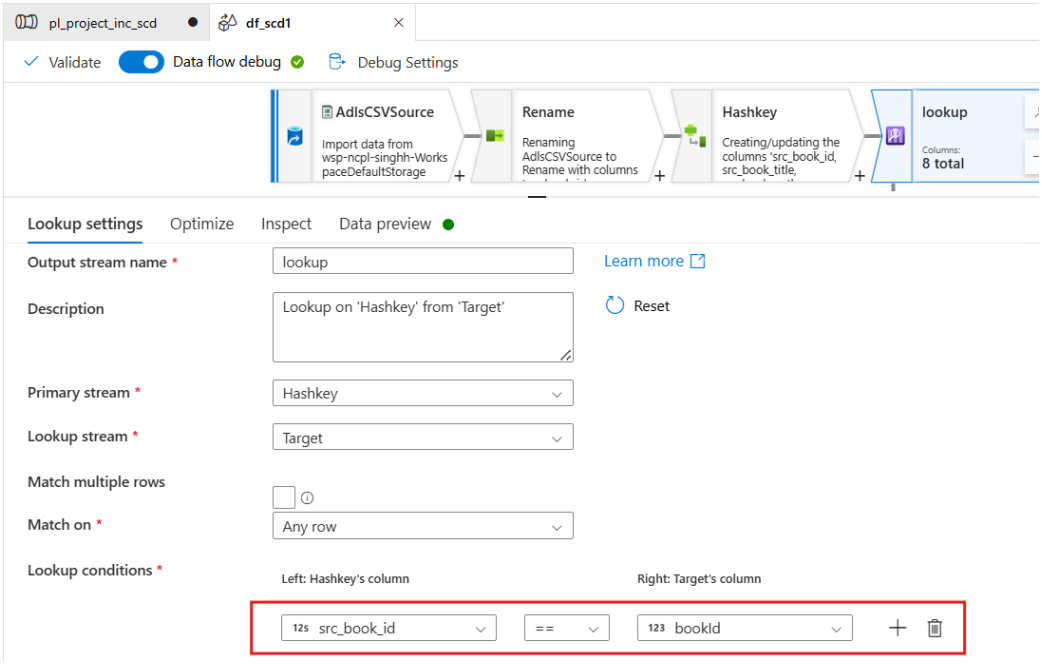
- Click on Projection -> import schema.



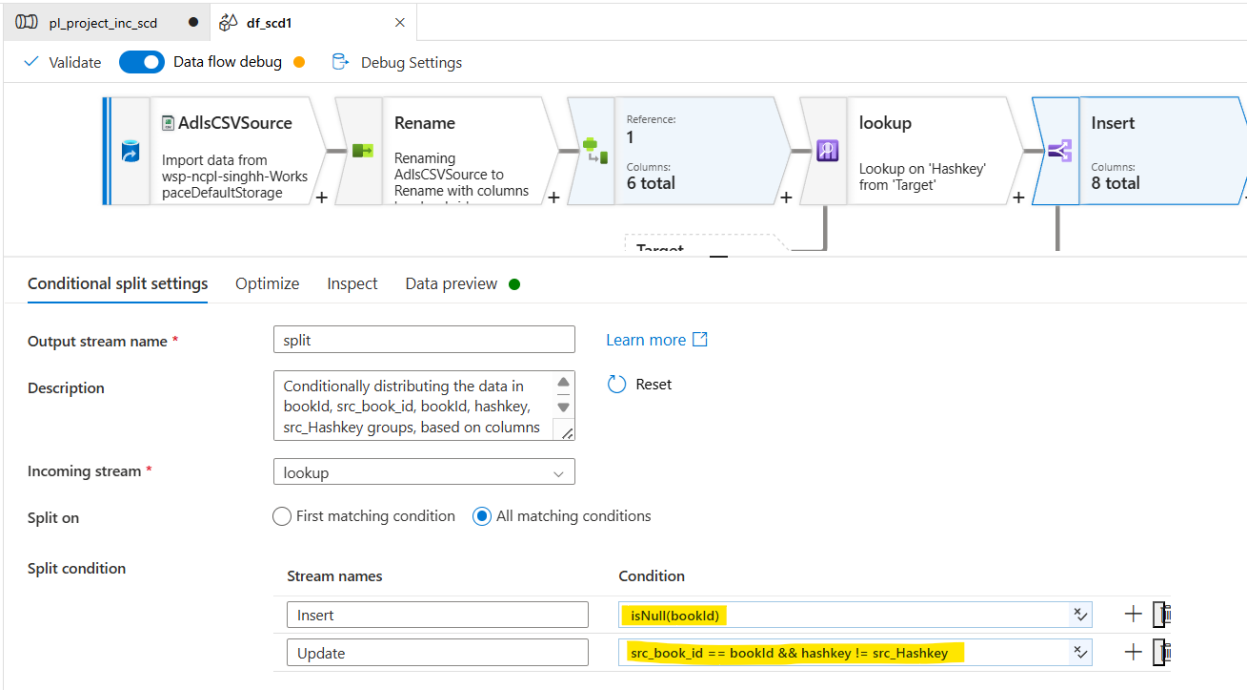




– Add **lookup activity** that will perform a left join with target as we will be checking if the record exists or not, and match on IDs.



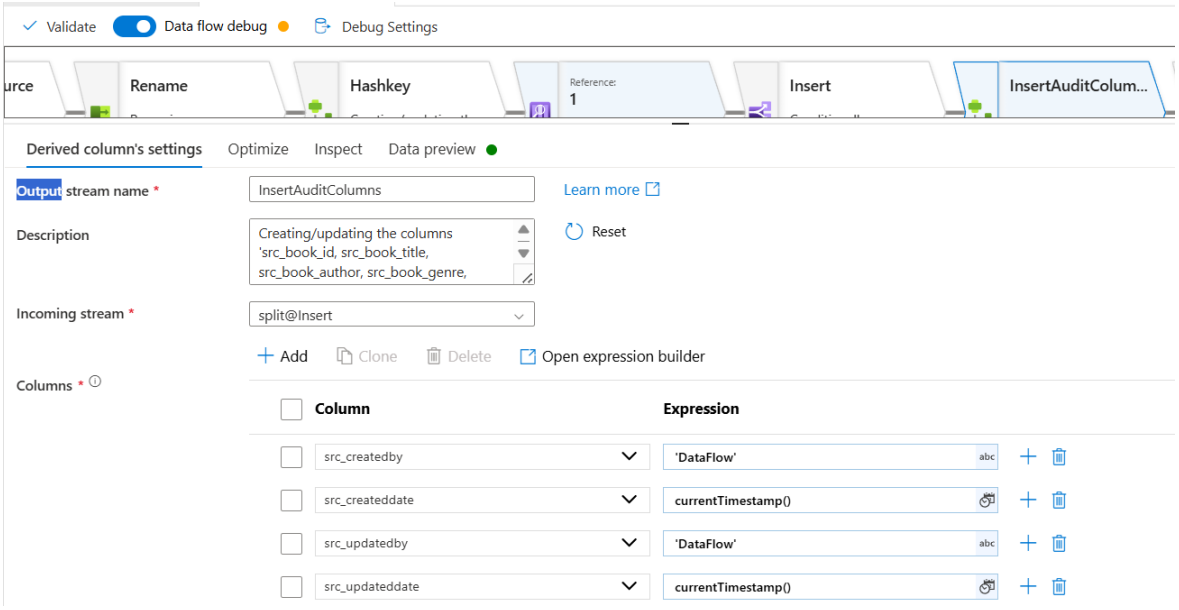
– Add **Conditional split** and add two conditions name **Insert** and **Update**.
  - Input check if Book ID is null, then it's a new record will directly insert it.
  - Update checks if our source Book ID and target Book ID matches but also hashkey shouldn't match because if there is let's say change in with ID=1, and unique Hashkey will be generated which cannot match with the already exisiting hashkey. So, means need to update record.
  - Expression used:
    Insert: ***isNull(bookId)***
    Update***: src_book_id == bookId && hashkey != src_Hashkey***



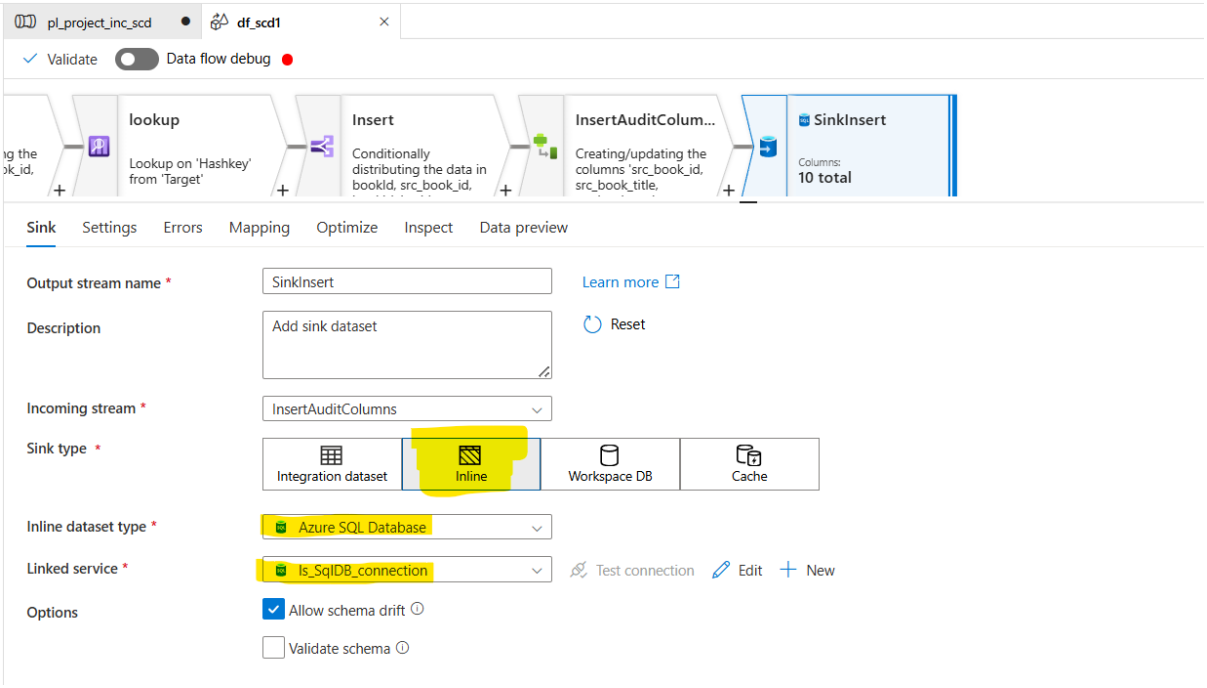
– In the Insert Flow, create a **derived column** to create the following columns,
  - src_createdby – with value as 'DataFlow'.
  - src_createddate – with value as current time stamp.
  - src_updatedby – with value as 'DataFlow'.
  - src_updateddate – with values as current time stamp.

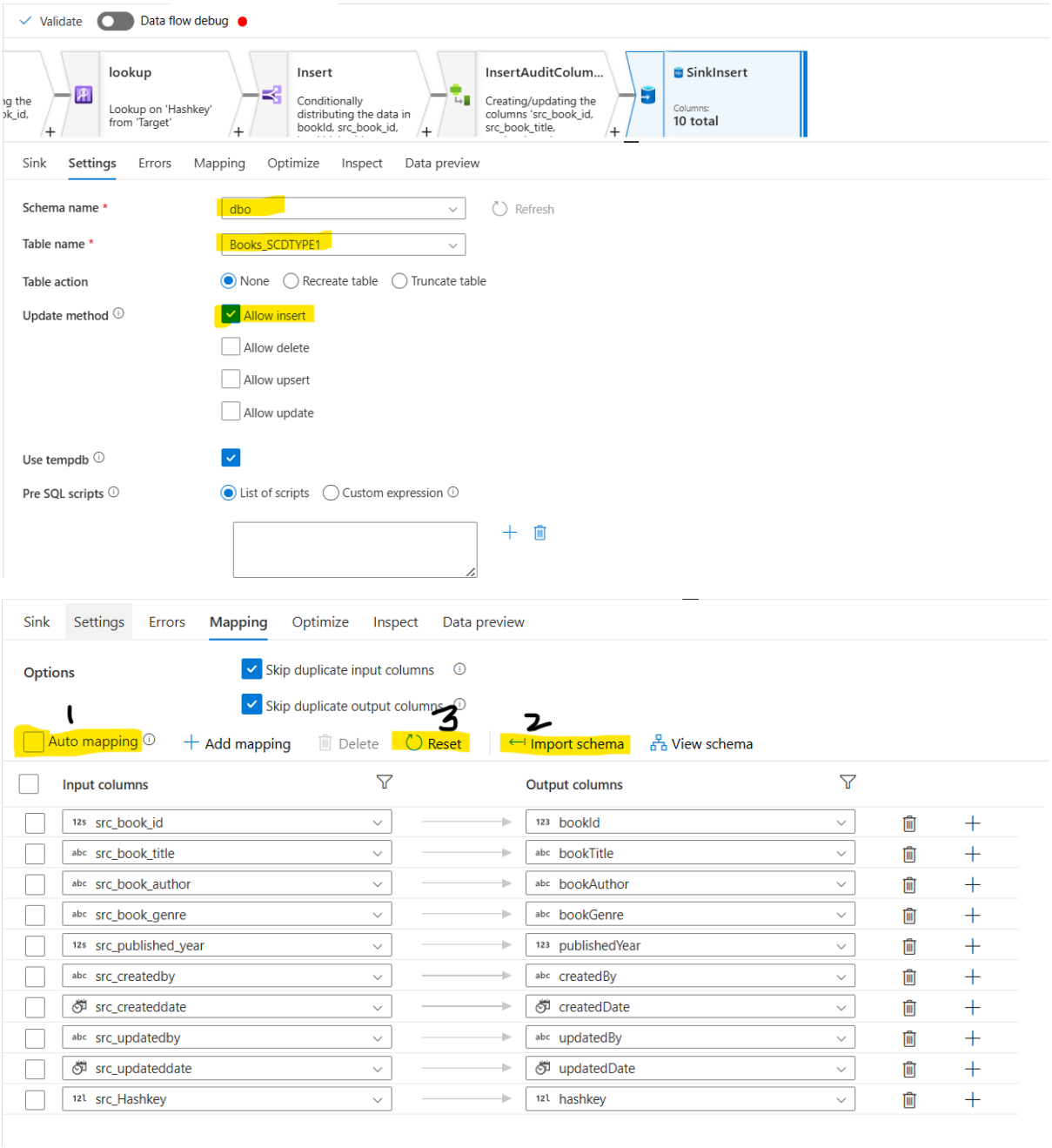
  Expression Used:

  Current timestamp: **currentTimestamp()**



– **Add sink** and select **Azure SQL DB** with the shown configuration.
  - Select only Allow **Insert** checkmark.
  - Go to mapping->Import Schema->Reset->Match input columns.

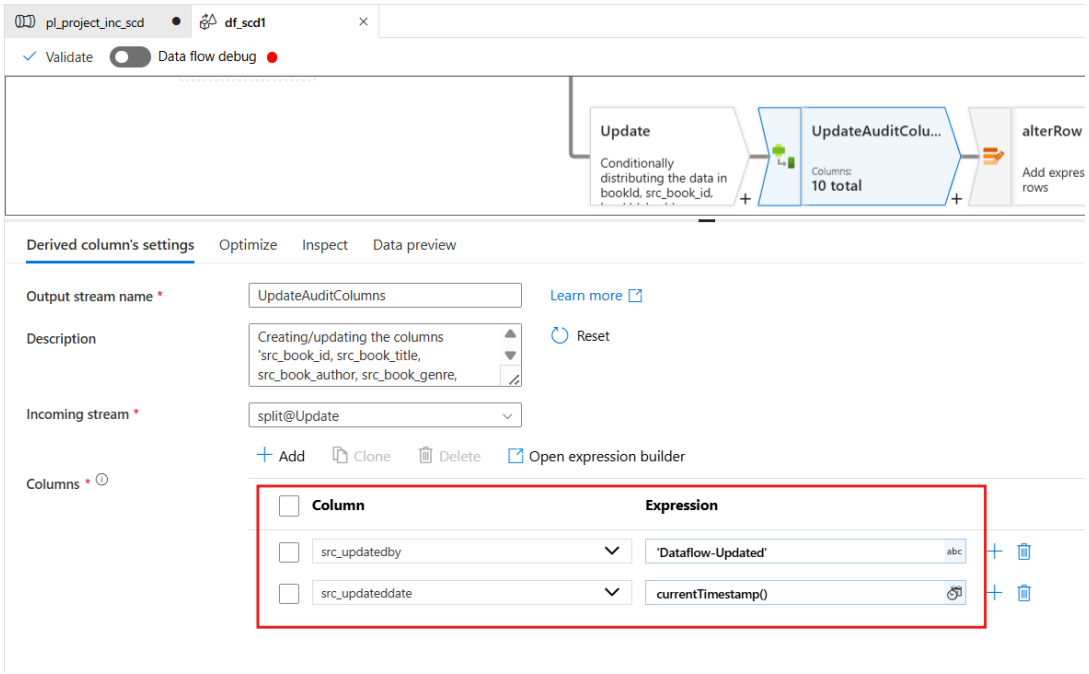




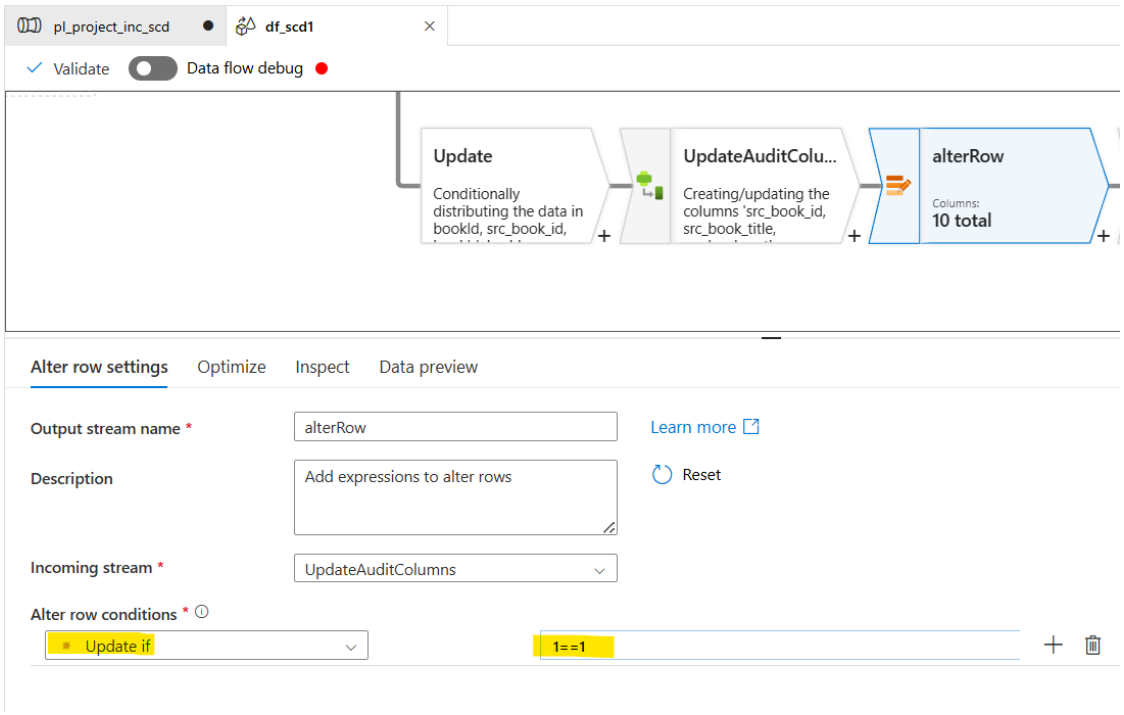
- In **update flow, add derived column**
  - Here, we will **create src_pdatedby** and **src_updateddate columns** as this update action will happen when there is any change in the existing records.
  - We have to keep createddate and createdby same, only Updatedby and updateddate will update as dataflow-Updated and current timestamp.

  Expression Used:

  Current timestamp: **currentTimestamp()**

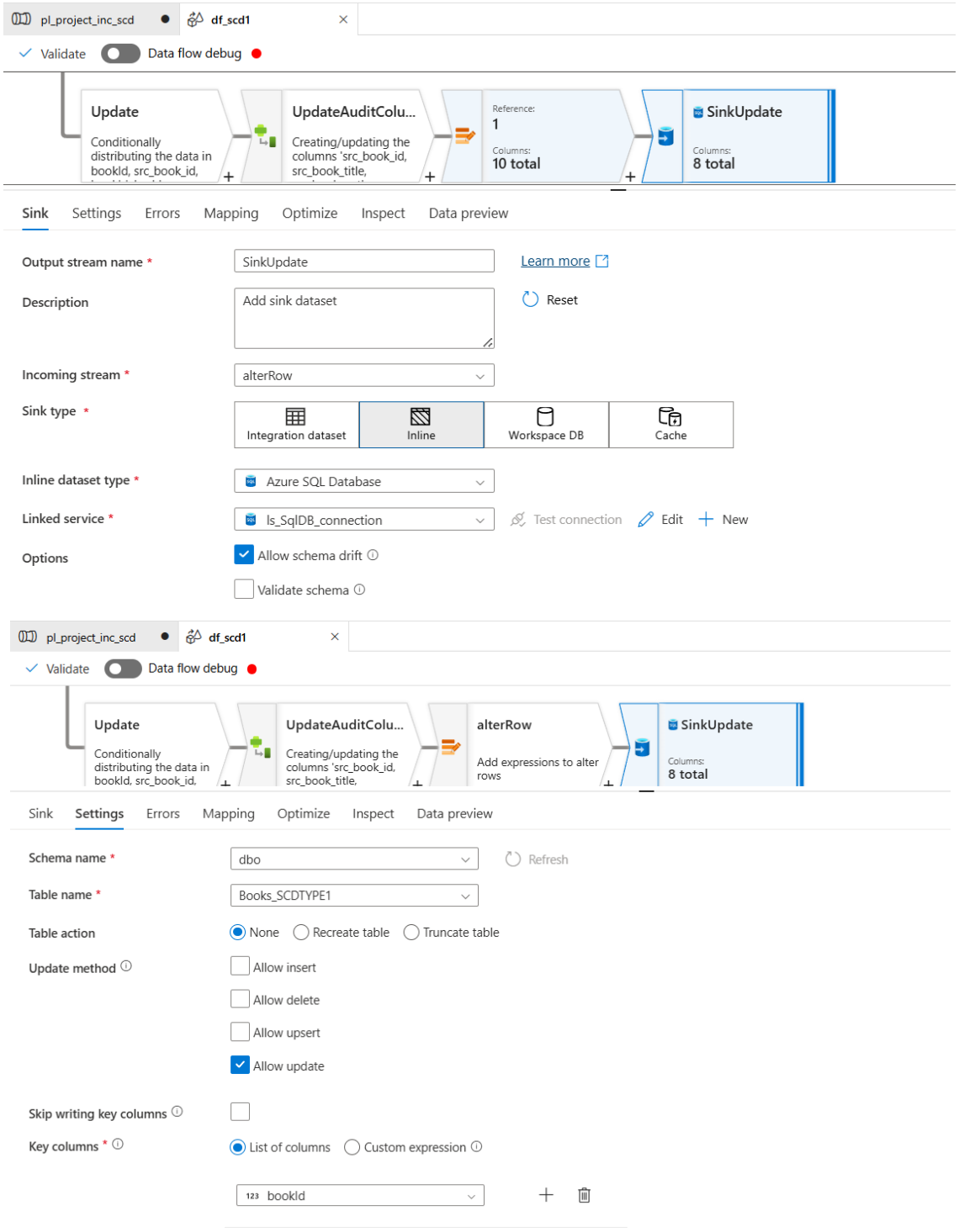


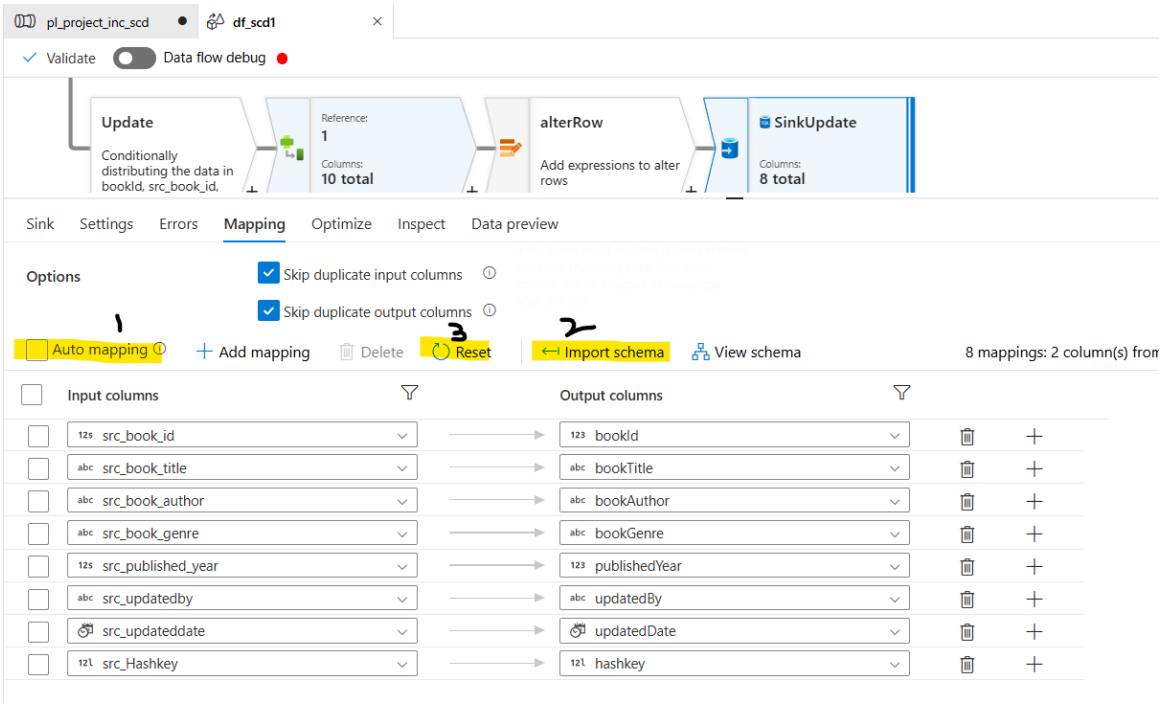
– Add **Alter row** transformation which gives permission to alter the data.
  - Give condition as **Update If** as passing expression as **1==1**



– **Add sink** and select **Azure SQL DB** with the shown configuration.
  - Select only Allow Update checkmark and give bookId in key column as this will help to identify changes.
  - Go to mapping->Import Schema->Reset->Match input columns.

- Delete createdby and createddate column as we want to keep it same with the actual and don't want to update.





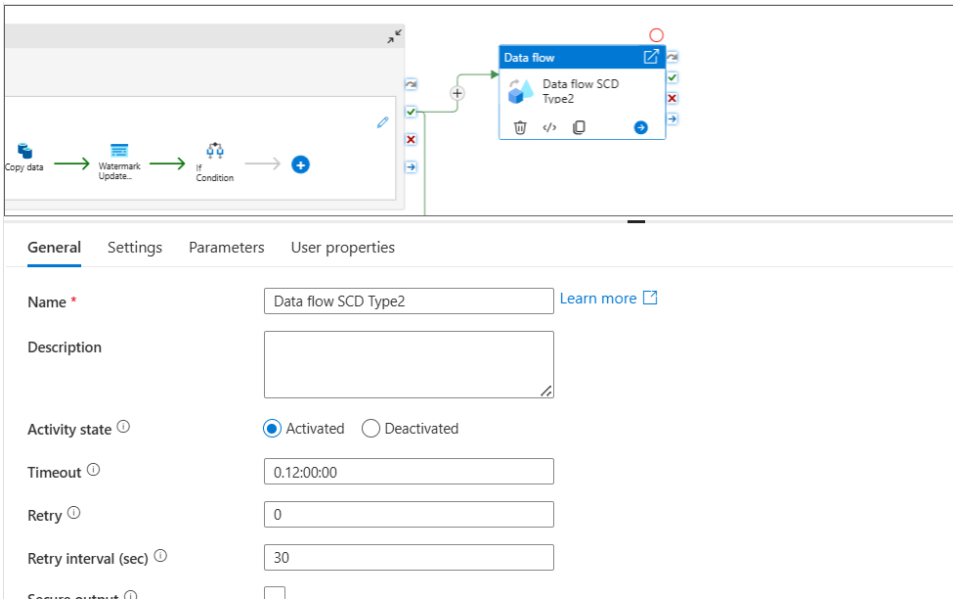
STEP 9: Set Data flow Activity for SCD Type 2

- Here, I implemented SCD Type 2 data load for table **Members** to load the data in **Azure SQL DB** from **ADLS Gen2.**
- To load the **Members** Data in Azure SQL DB, we need to create table definition in **Azure SQL DB database** as shown below:

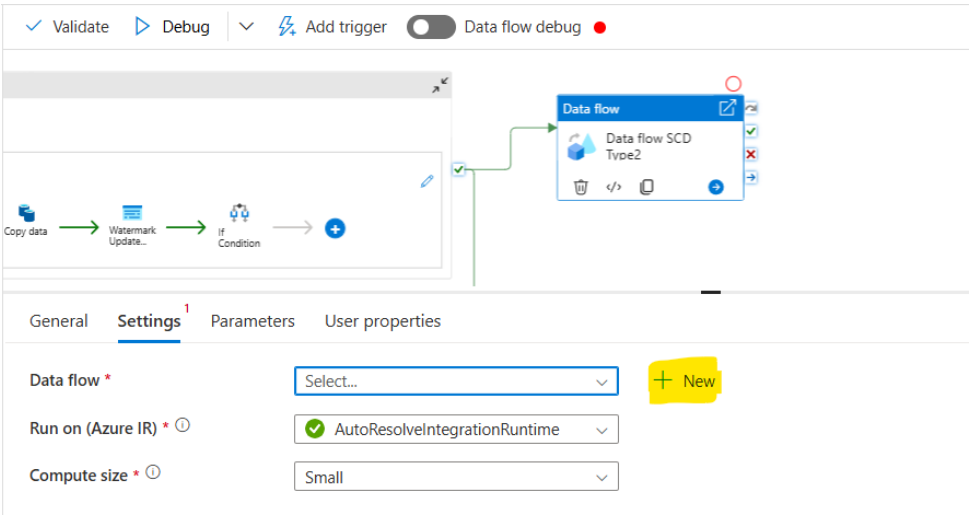
```
Create Table Members_SCDTYPE2
(
        memberId Int,
        memberName Varchar(100),
        memberEmail Varchar(255),
        memberPhone Varchar(20) ,
        createdBy varchar(100),createdDate datetime,
        updatedBy varchar(100),updatedDate datetime,hashkey Bigint,
        isActive Int
)
select * from Members_SCDTYPE2
```

```
Create Table Members_SCDTYPE2
(
    memberId Int,
    memberName Varchar(100),
    memberEmail Varchar(255),
    memberPhone Varchar(20) ,
    createdBy varchar(100),createdDate datetime,
    updatedBy varchar(100),updatedDate datetime,hashkey Bigint,
    isActive Int
)
select * from Members_SCDTYPE2
```

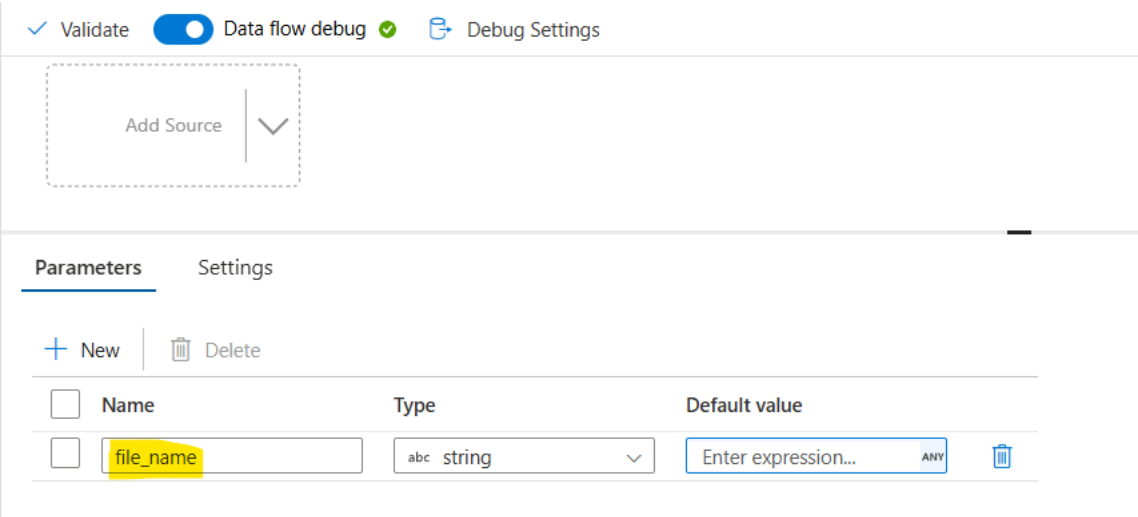
– Drag drag Data Flow Activity and turn on Data flow debug.
Connect it to for each activity with success point.



– Go to Settings Tab of Data flow activity, and create new data flow.

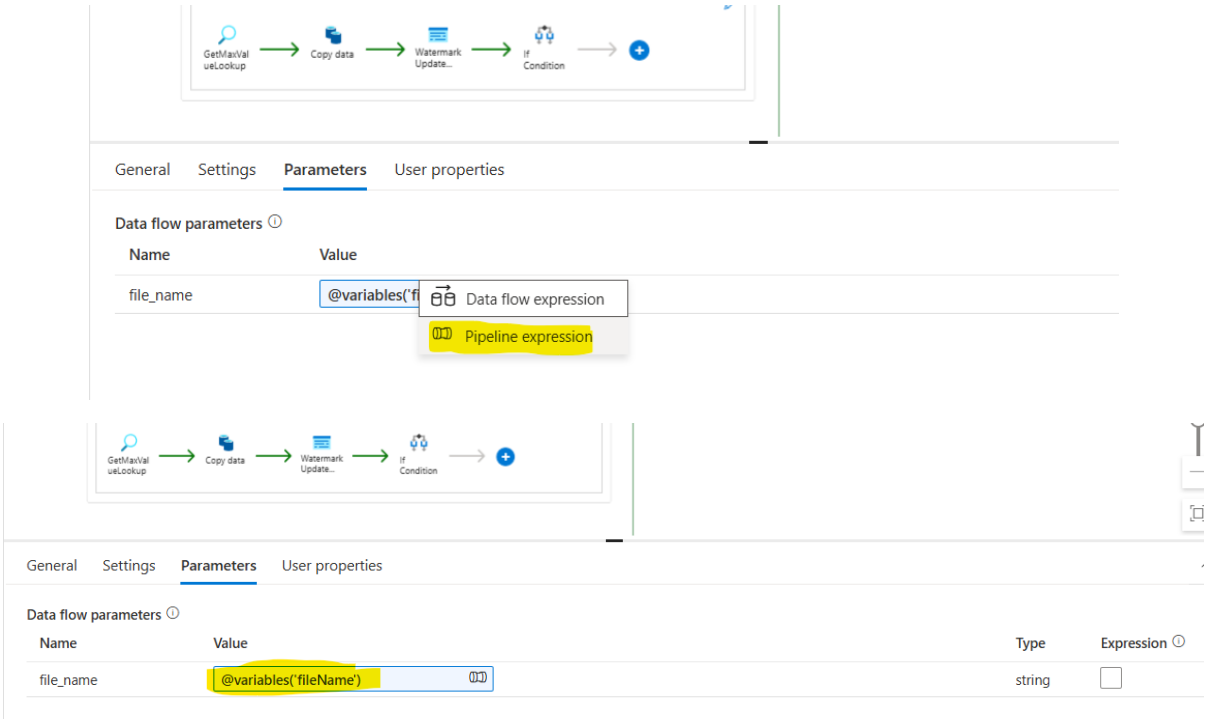


– Open the Created Data flow and create a new parameter named ***file_name*** inside data flow activity.

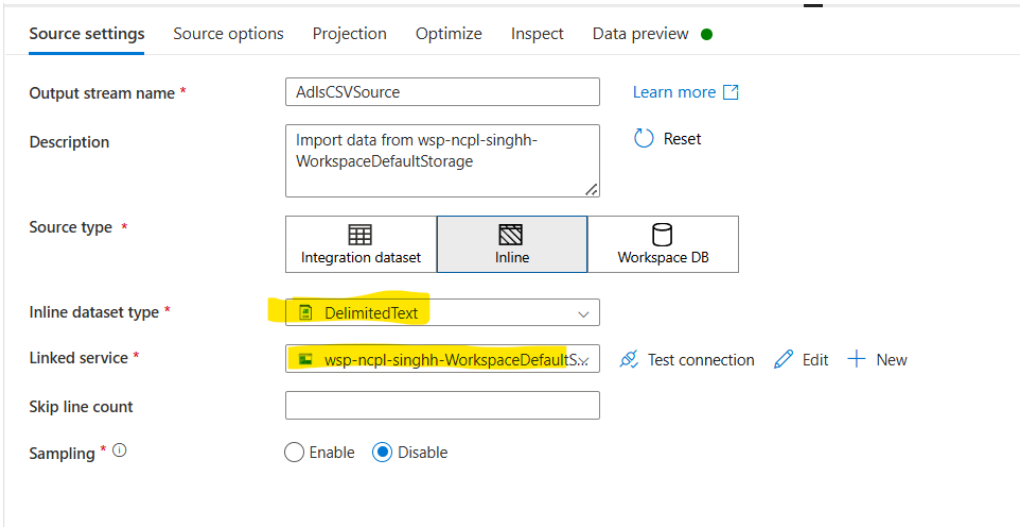


– Then, move to Parameters Tab of Data flow activity in the pipeline, and **create** a new **Parameter** name ***file_name***. Use the Dynamic Expression to set the parameter's value with **previously created pipeline variable** named ***filename.***

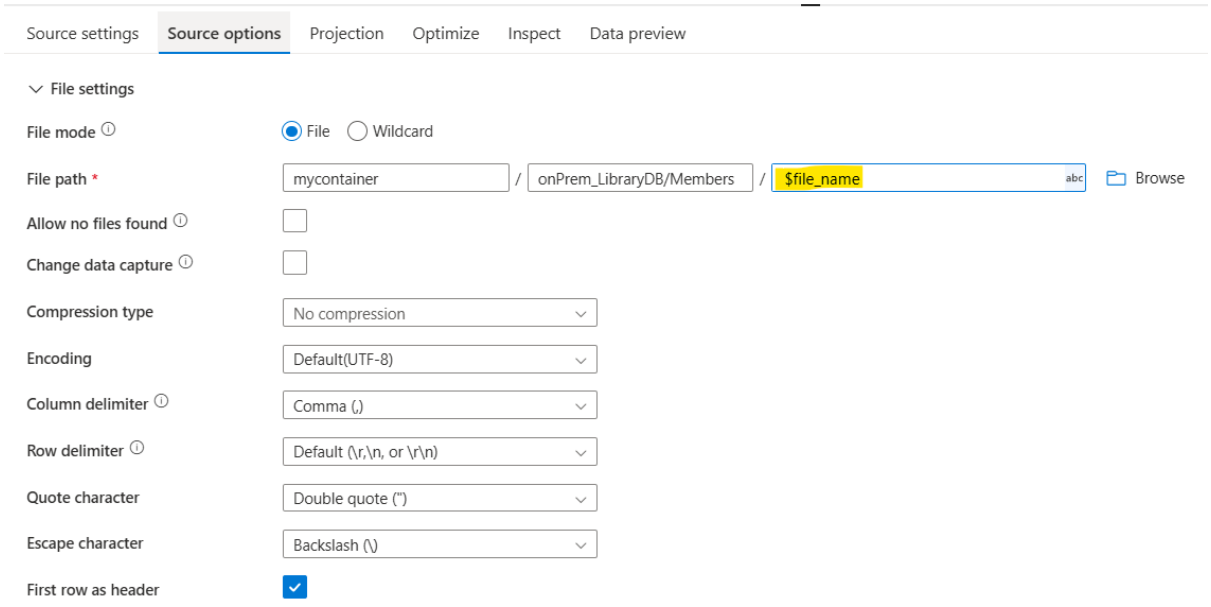
Expression Used:
file_name: **@variables('fileName')**

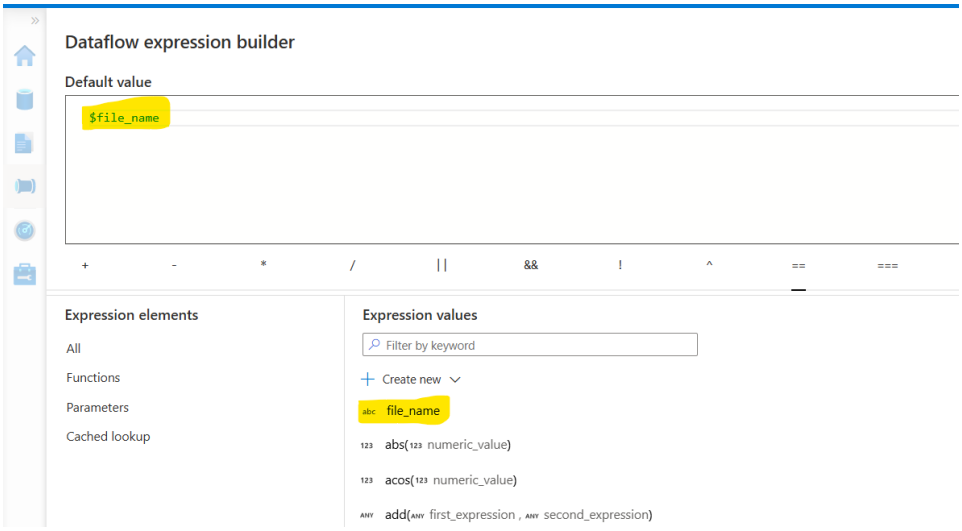


– Next Open the Data flow activity again, and click on **Add source**.
  - Then, Goto Source Settings -> Click on Source Type -> Inline and select the source.

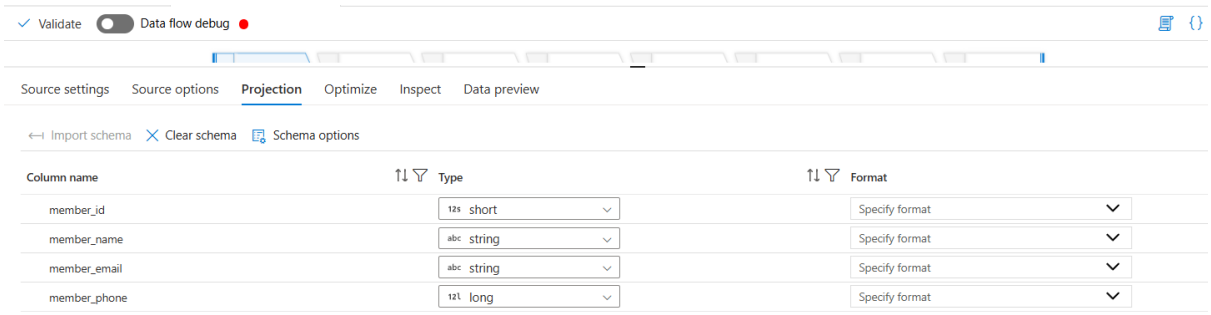


  - Next, Go to Source Options -> Select the File from container.
  - Here, for the file name we used the parameter value created in data flow. Dynamic Expression Used: ***$file_name***

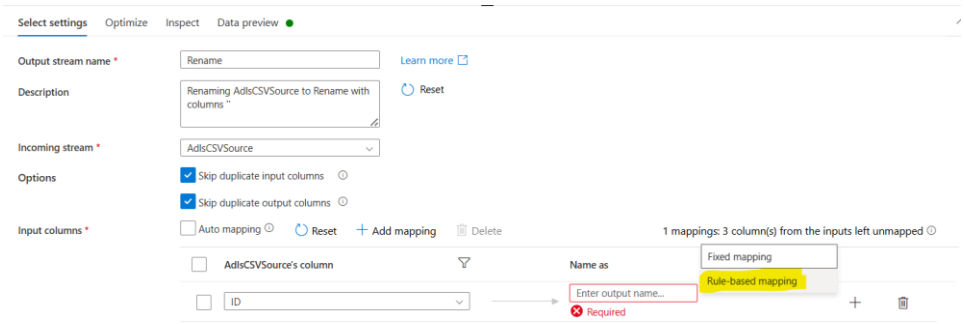


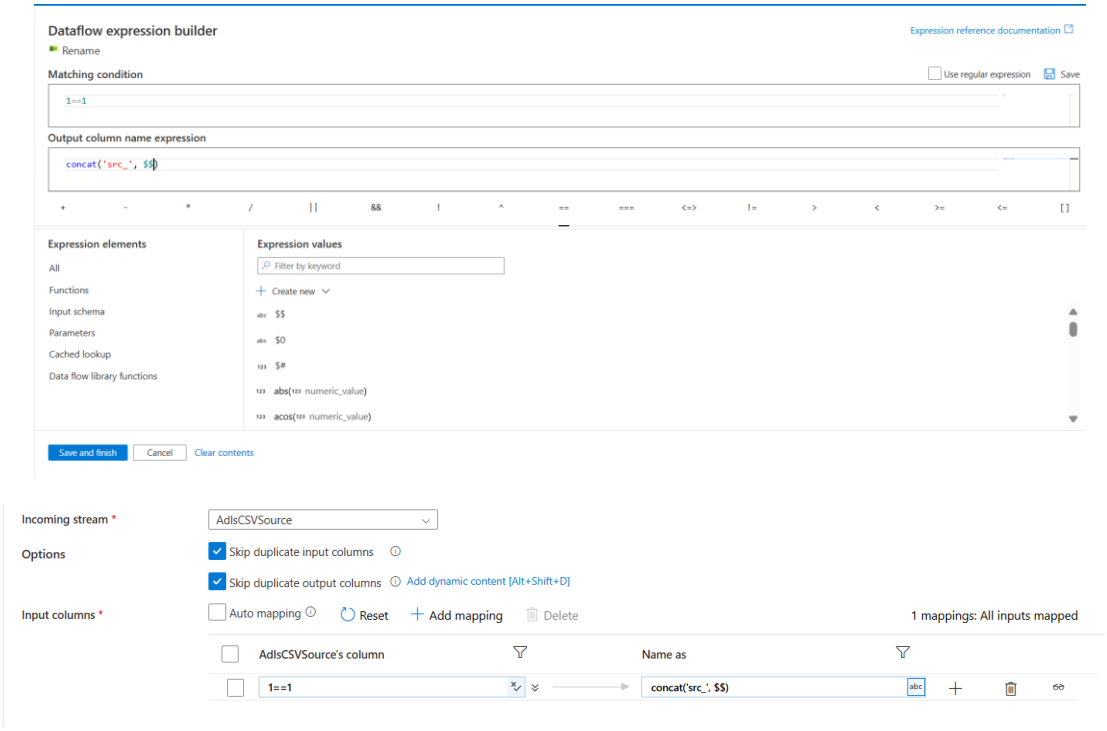


– Click on Projection tab of Source -> click on import schema -> select and review the correct type for columns.



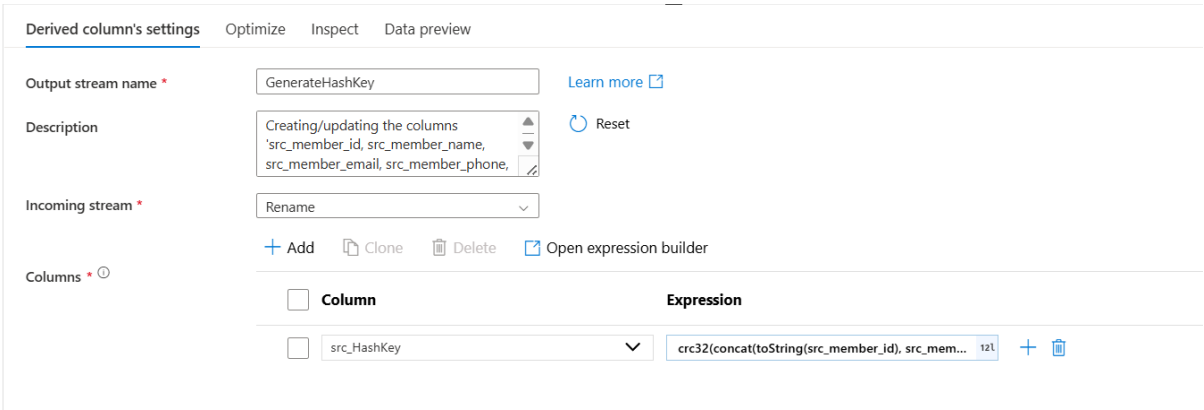
– Add **Select column** as its needed to rename columns with src_columnnames
  - Then, under settings select all the columns and delete the mapping.
  - Next, Click on Add mapping and select rule-based mapping.
  - In the new column, give condition 1==1 to make it true, and give name as concat('src_',$$). This will make dynamic and add src in front of all the columns. Also, remove the above id one row mapping as we don't need it.





– Add **Derived column**, activity.
  - Under derived column settings, add column src_HashKey and enter this expression.
    crc32(concat(toString(src_member_id), src_member_name, src_member_email, toString(src_member_phone)))

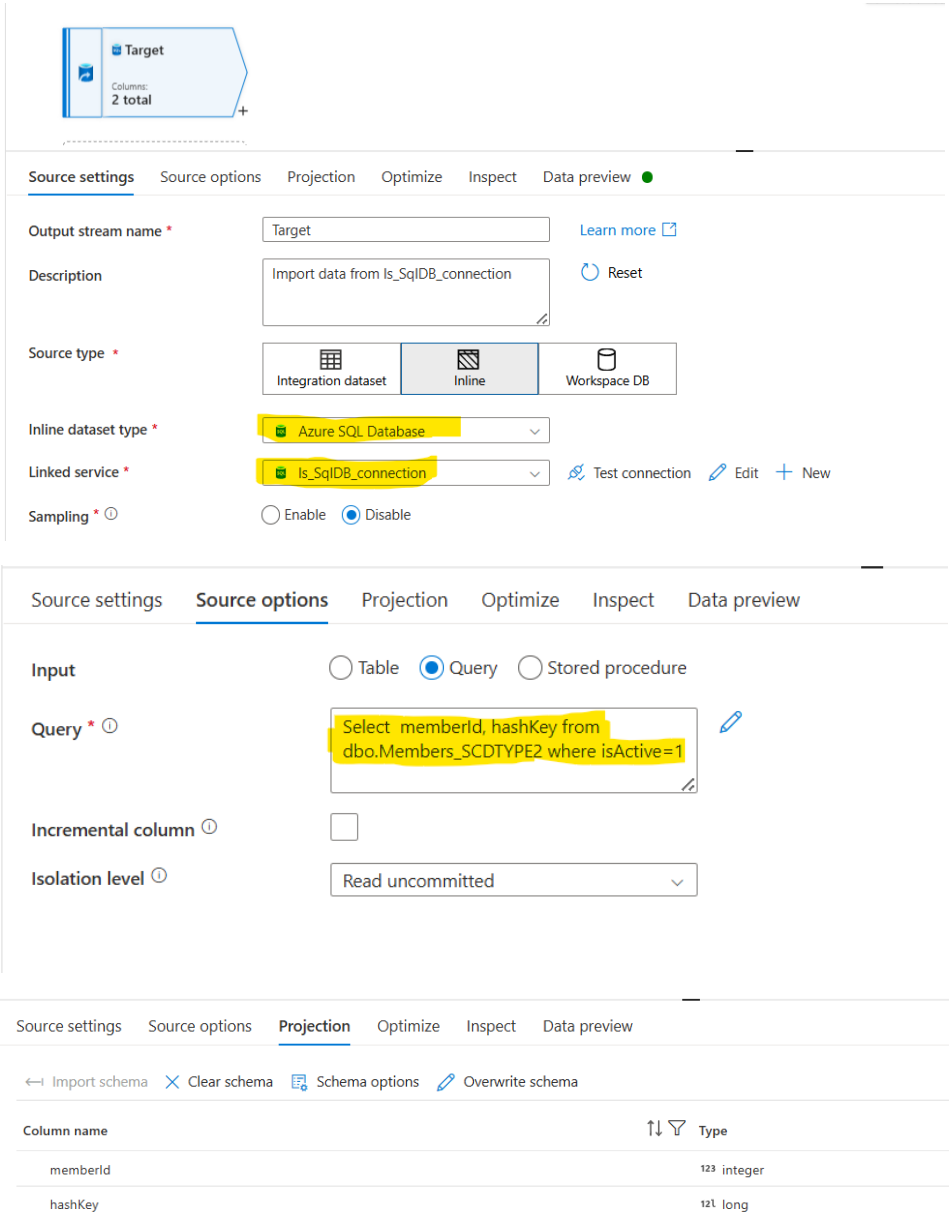
    Note: Crc32 generates hashkey using the mentioned combination.



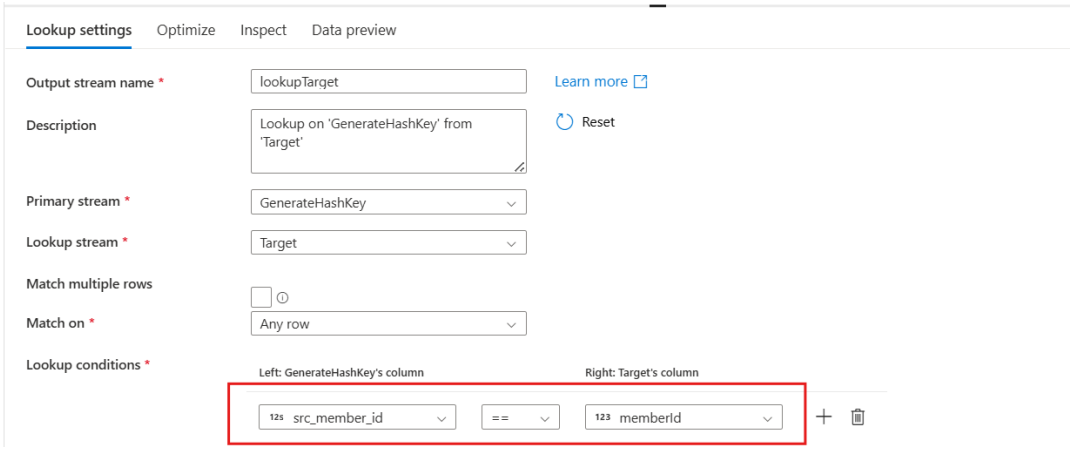
– Add **target** that is **Azure SQL DB** as another source.
  - Select two column **memberId and hashkey** from target table which we will use to check for new records or existing records. We need rows where isActive is 1 which means the most updatedrecords.

Query Used: **Select memberId, hashKey from dbo.Members_SCDTYPE2 where isActive=1**

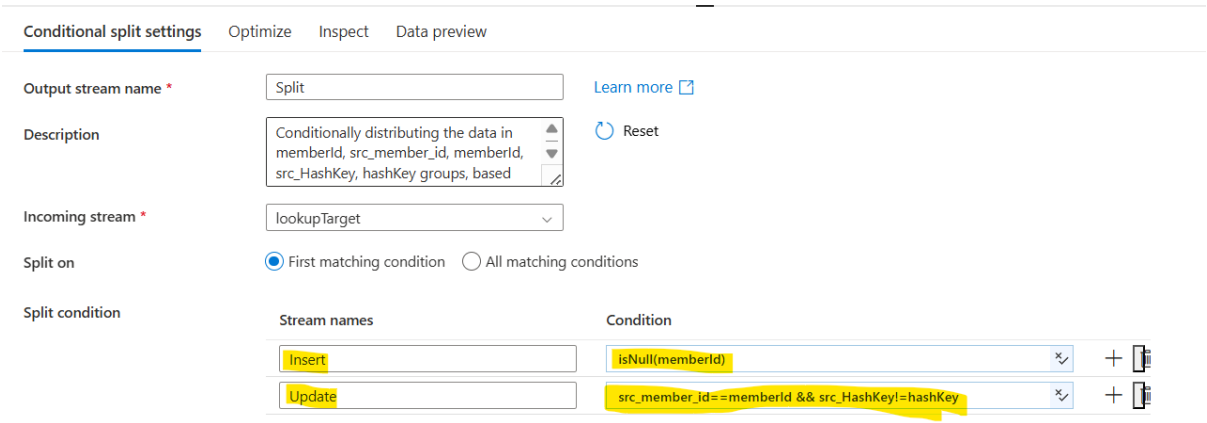
- Click on Projection -> import schema.



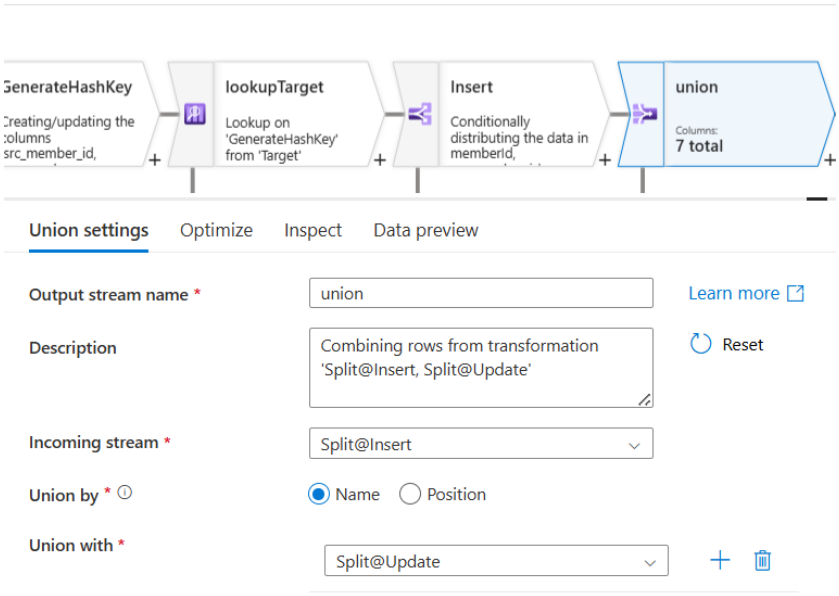
– Add **lookup activity** that will perform a left join with target as we will be checking if the record exists or not, and match on IDs.



– Add **Conditional split** and add two conditions name **Insert** and **Update**.
  - Input check if Member ID is null, then it's a new record will directly insert it.
  - Update checks if our source Member ID and target Member ID matches but also hashkey shouldn't match because if there is let's say change in with ID=1, and unique Hashkey will be generated which cannot match with the already exisiting hashkey. So, means need to update record.
  - Expression used:
    Insert: ***isNull(memberId)***
    Update***: src_member_id==memberId && src_HashKey!=hashKey***



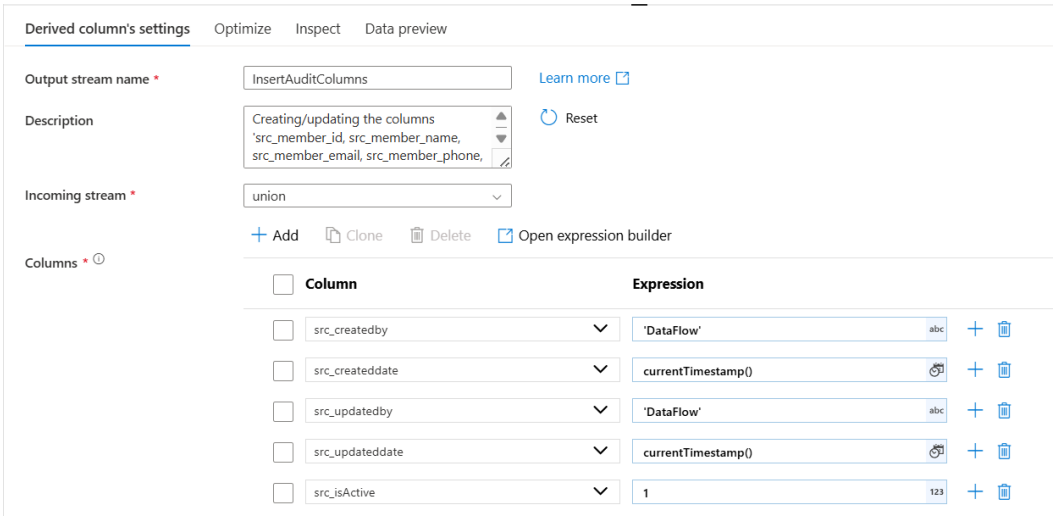
– In the Insert Flow, **add union** because we want to keep new and previous records.
  - In union with, select update branch



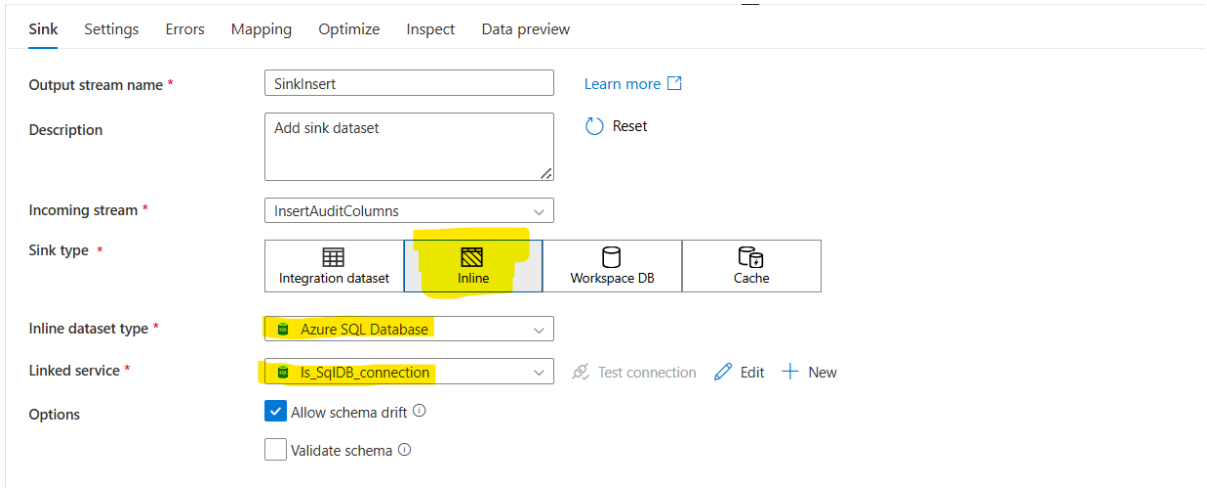
– Create a **derived column** to create the following columns,
  - src_createdby – with value as 'DataFlow'.
  - src_createddate – with value as current time stamp.
  - src_updatedby – with value as 'DataFlow'.
  - src_updateddate – with values as current time stamp.
  - src_isActive – with values as 1.

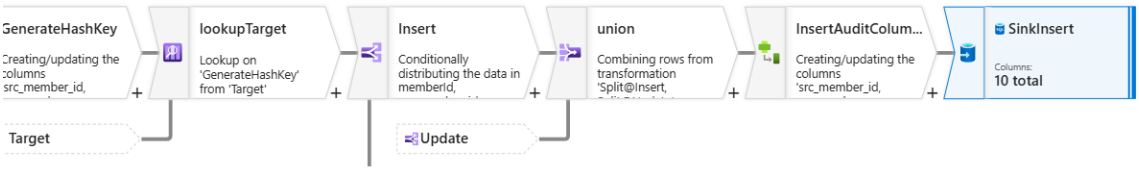
  Expression Used:

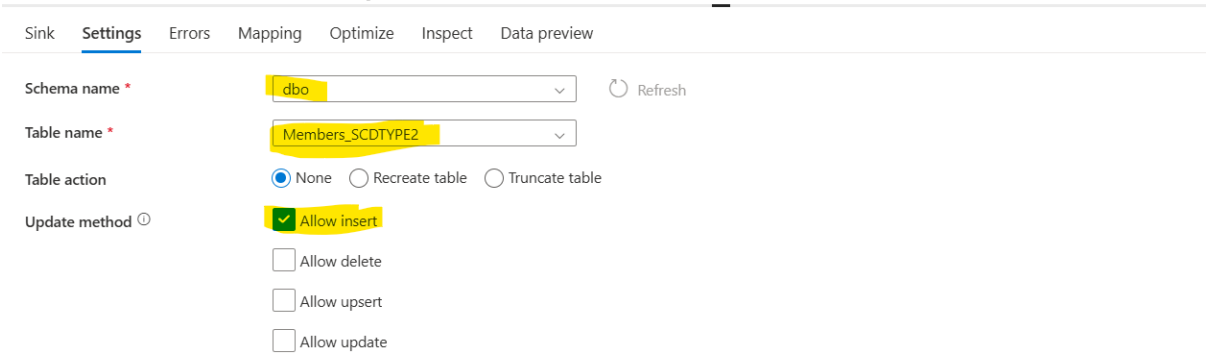
  Current timestamp: **currentTimestamp()**

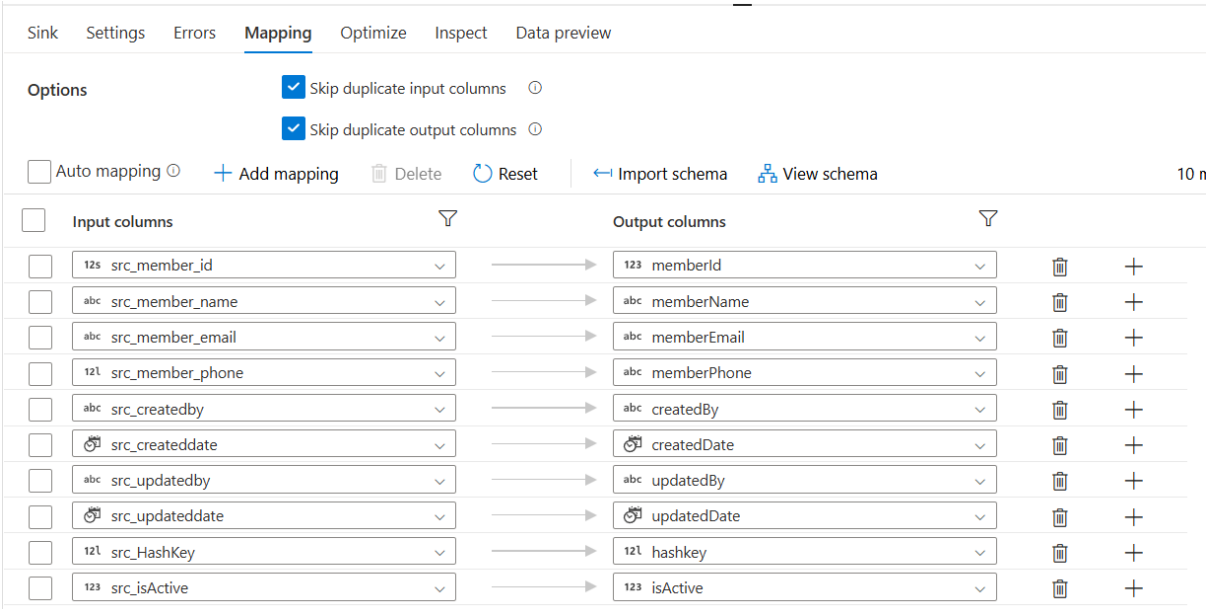


– **Add sink** and select **Azure SQL DB** with the shown configuration.
  - Select only Allow **Insert** checkmark.
  - Go to mapping->Import Schema->Reset->Match input columns.


Sink
Settings
Errors
Mapping
Optimize
Inspect
Data preview
Output stream name *
SinkInsert
Learn more
Description
Add sink dataset
Reset
Incoming stream *
InsertAuditColumns
Sink type *
Integration dataset
Inline
Workspace DB
Cache
Inline dataset type *
Azure SQL Database
Linked service *
Is_SqlDB_connection
Test connection
Edit
New
Options
Allow schema drift
Validate schema


GenerateHashKey
Creating/updating the columns src_member_id,
lookupTarget
Lookup on 'GenerateHashKey' from 'Target'
Insert
Conditionally distributing the data in memberId,
union
Combining rows from transformation 'Split@Insert,
InsertAuditColum...
Creating/updating the columns 'src_member_id,
SinkInsert
Columns:
10 total
Target
Update

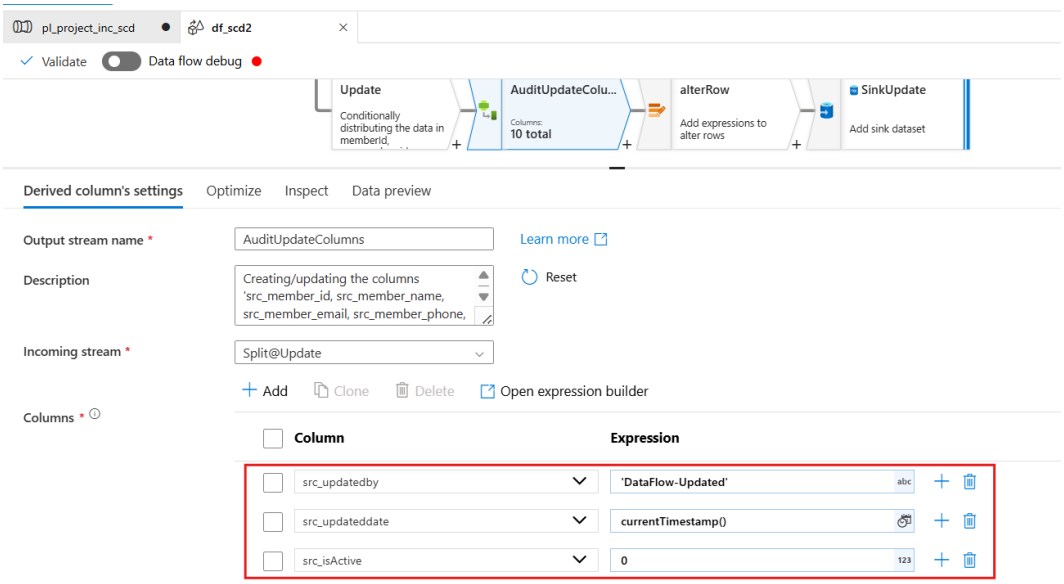

Sink
Settings
Errors
Mapping
Optimize
Inspect
Data preview
Schema name *
dbo
Refresh
Table name *
Members_SCDTYPE2
Table action
None
Recreate table
Truncate table
Update method
Allow insert
Allow delete
Allow upsert
Allow update


Sink
Settings
Errors
Mapping
Optimize
Inspect
Data preview
Options
Skip duplicate input columns
Skip duplicate output columns
Auto mapping
Add mapping
Delete
Reset
Import schema
View schema
Input columns
Output columns
src_member_id
memberId
src_member_name
memberName
src_member_email
memberEmail
src_member_phone
memberPhone
src_createdby
createdBy
src_createddate
createdDate
src_updatedby
updatedBy
src_updateddate
updatedDate
src_HashKey
hashkey
src_isActive
isActive

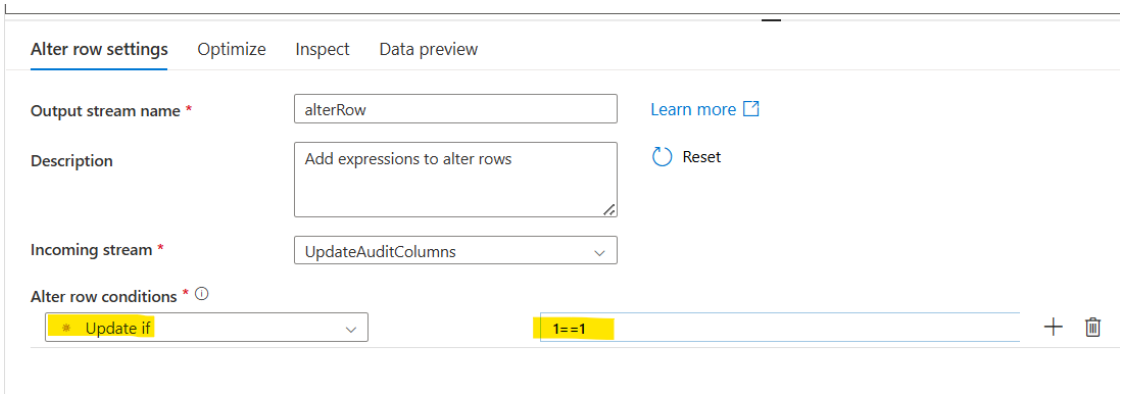
– In **update flow, add derived column**
  - Here, we will **create src_pdatedby**, **src_updateddate** and **src_isActive columns** as this update action will happen when there is any change in the existing records.
  - We have to keep createddate and createdby same, only Updatedby and updateddate will update as dataflow-Updated and current timestamp.
  - Update the isActive value as 0.

  Expression Used:

  Current timestamp: **currentTimestamp()**

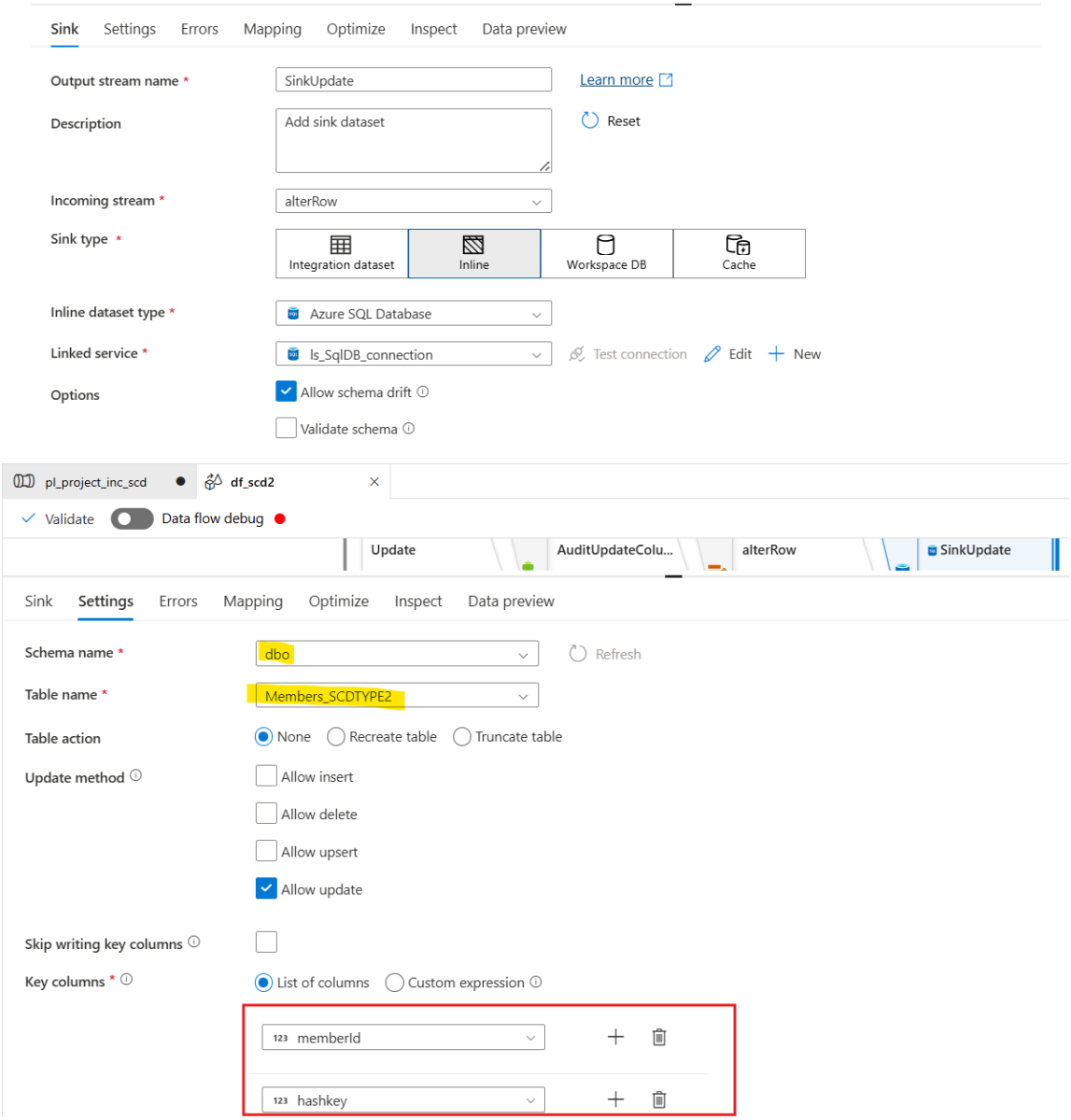


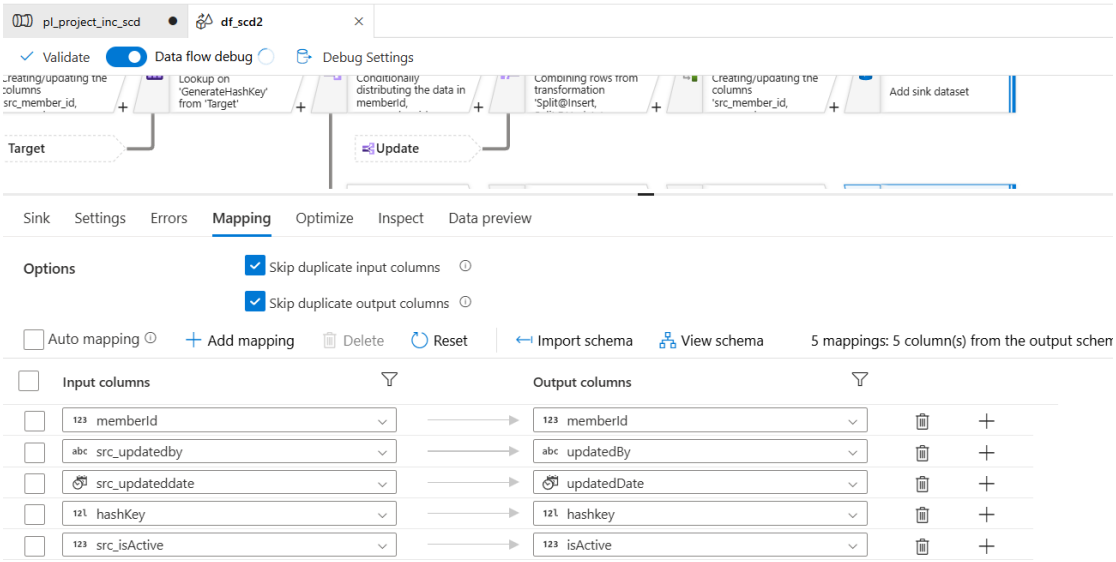
– Add **Alter row** transformation which gives permission to alter the data.
  - Give condition as **Update If** as passing expression as **1==1**



– **Add sink** and select **Azure SQL DB** with the shown configuration.

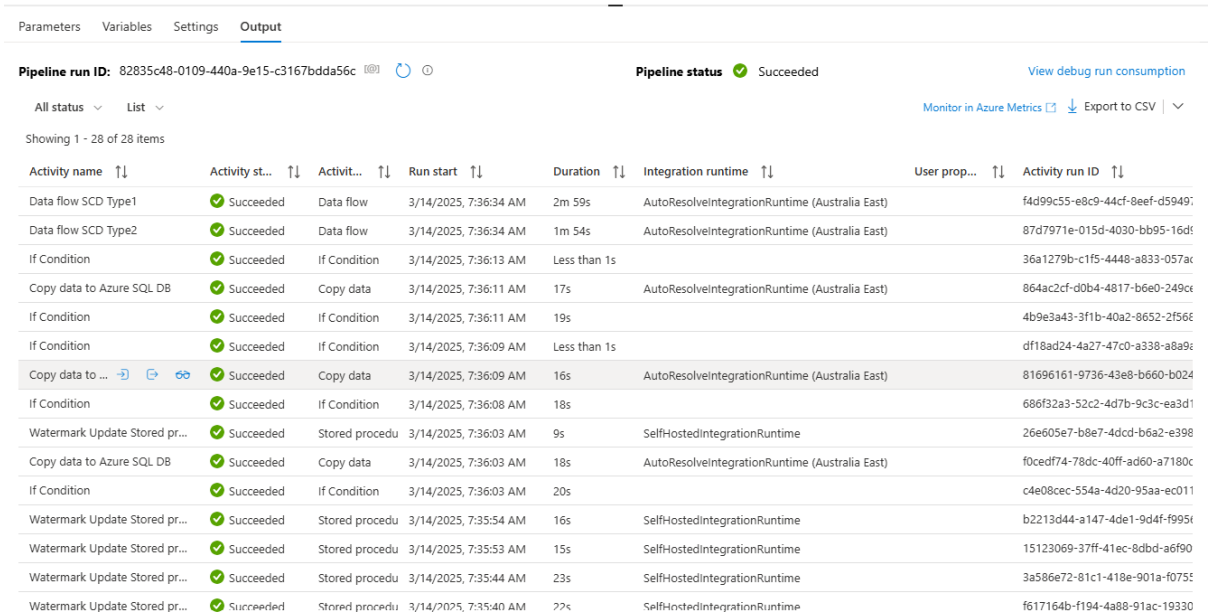
- Select only Allow Update checkmark and give memberId and hashkey in key columns, because we will have multiple ID as we are keeping previous record as well, so this combination will help to identify changes.
- Go to mapping->Import Schema->Reset->Match input columns.
- Delete createdby and createddate column as we want to keep it same with the actual and don't want to update.
- HashKey will also be kept the same by mapping it with the old one.



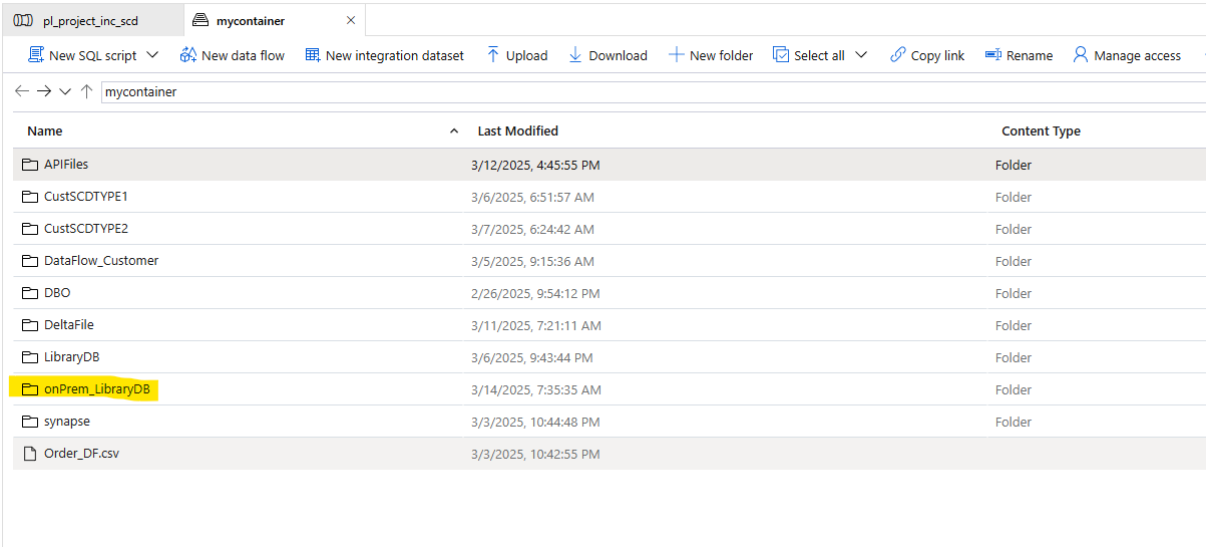


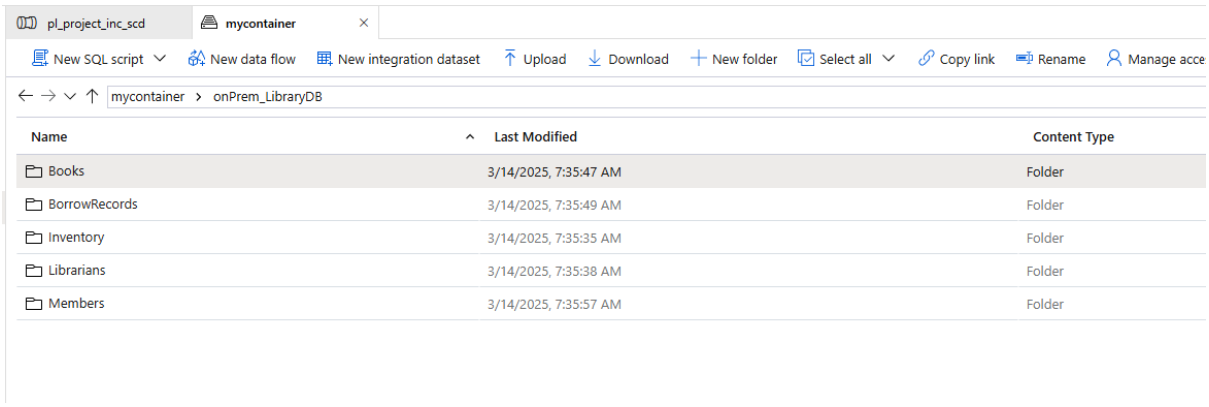
STEP 10: Debug & Publish the Activity

- Finally, Debug the Activity to execute it and publish it to save it and review the output.
- Execution Logs:

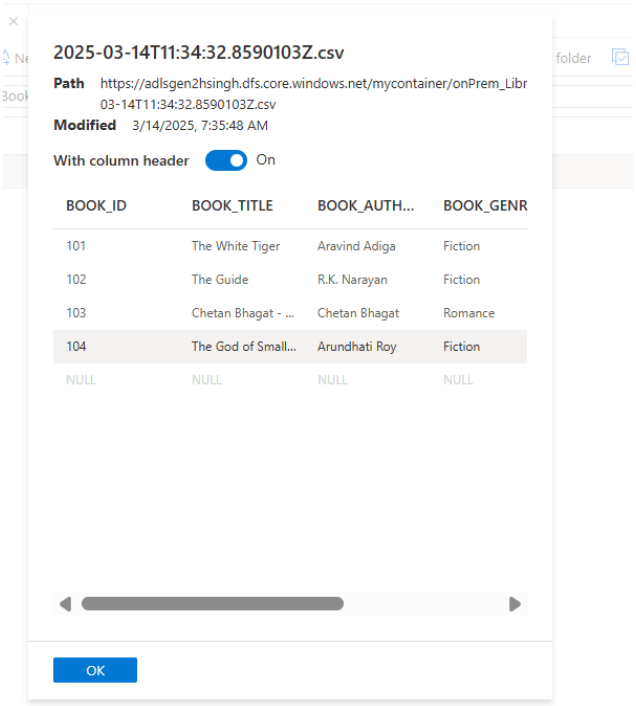


- Created File Review in **ADLS Gen2**:

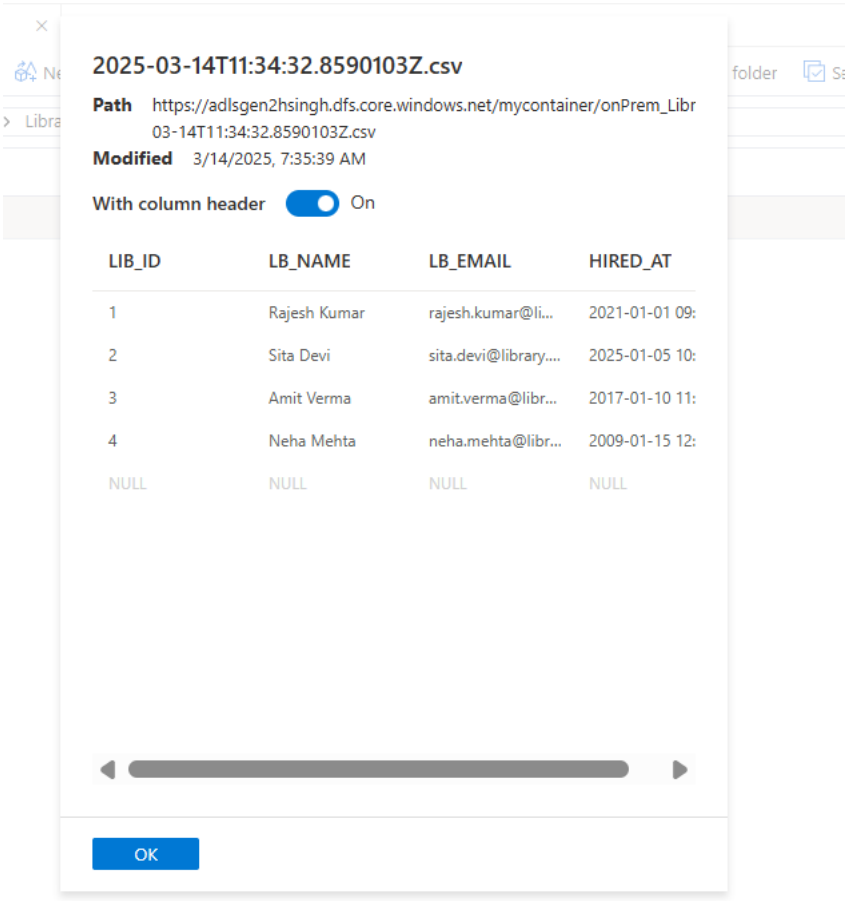

pl_project_inc_scd
mycontainer
New SQL script
New data flow
New integration dataset
Upload
Download
New folder
Select all
Copy link
Rename
Manage access
mycontainer
Name
Last Modified
Content Type
APIFiles 3/12/2025, 4:45:55 PM Folder
CustSCDTYPE1 3/6/2025, 6:51:57 AM Folder
CustSCDTYPE2 3/7/2025, 6:24:42 AM Folder
DataFlow_Customer 3/5/2025, 9:15:36 AM Folder
DBO 2/26/2025, 9:54:12 PM Folder
DeltaFile 3/11/2025, 7:21:11 AM Folder
LibraryDB 3/6/2025, 9:43:44 PM Folder
onPrem_LibraryDB 3/14/2025, 7:35:35 AM Folder
synapse 3/3/2025, 10:44:48 PM Folder
Order_DF.csv 3/3/2025, 10:42:55 PM


pl_project_inc_scd
mycontainer
New SQL script
New data flow
New integration dataset
Upload
Download
New folder
Select all
Copy link
Rename
Manage acce
mycontainer › onPrem_LibraryDB
Name
Last Modified
Content Type
Books 3/14/2025, 7:35:47 AM Folder
BorrowRecords 3/14/2025, 7:35:49 AM Folder
Inventory 3/14/2025, 7:35:35 AM Folder
Librarians 3/14/2025, 7:35:38 AM Folder
Members 3/14/2025, 7:35:57 AM Folder

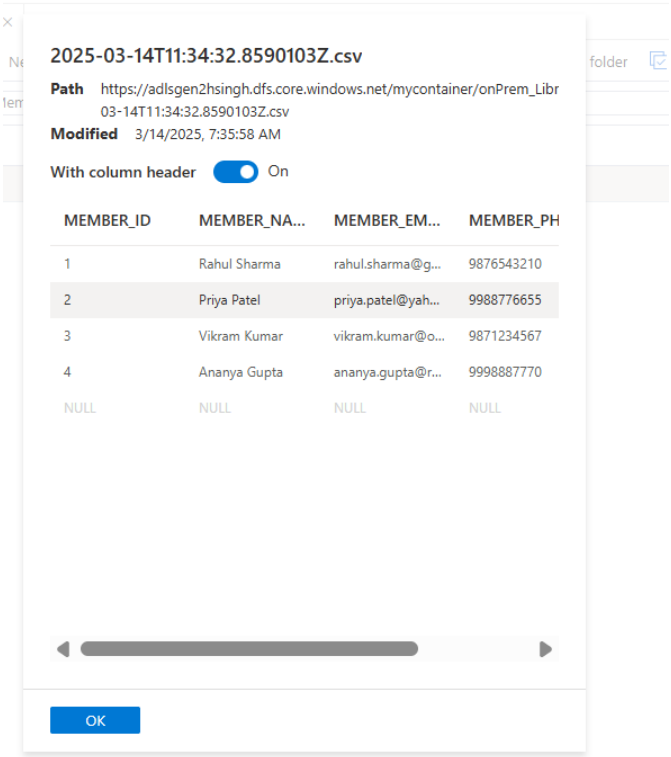
Books:


2025-03-14T11:34:32.8590103Z.csv
Path https://adlsgen2hsingh.dfs.core.windows.net/mycontainer/onPrem_Libr
03-14T11:34:32.8590103Z.csv
Modified 3/14/2025, 7:35:48 AM
With column header On
BOOK_ID BOOK_TITLE BOOK_AUTH... BOOK_GENR
101 The White Tiger Aravind Adiga Fiction
102 The Guide R.K. Narayan Fiction
103 Chetan Bhagat - ... Chetan Bhagat Romance
104 The God of Small... Arundhati Roy Fiction
NULL NULL NULL NULL
OK

Librarians:

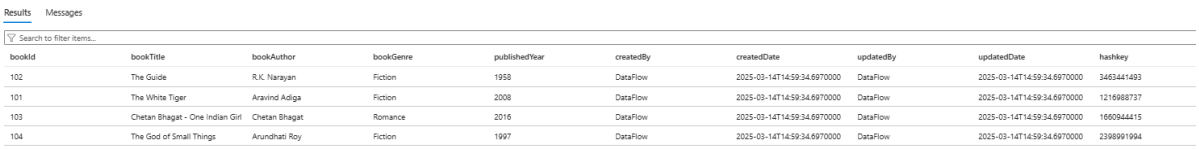

2025-03-14T11:34:32.8590103Z.csv
Path https://adlsgen2hsingh.dfs.core.windows.net/mycontainer/onPrem_Libr
03-14T11:34:32.8590103Z.csv
Modified 3/14/2025, 7:35:39 AM
With column header On
LIB_ID LB_NAME LB_EMAIL HIRED_AT
1 Rajesh Kumar rajesh.kumar@li... 2021-01-01 09:
2 Sita Devi sita.devi@library.... 2025-01-05 10:
3 Amit Verma amit.verma@libr... 2017-01-10 11:
4 Neha Mehta neha.mehta@libr... 2009-01-15 12:
NULL NULL NULL NULL
OK

Members (Same for all other tables):



Loaded Data in the Azure SQL DB:

Books:

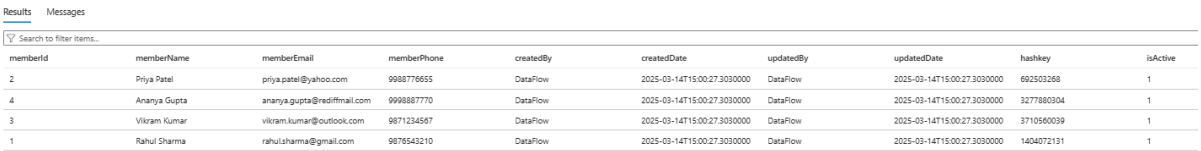


Results Messages

Search to filter items...

bookId	bookTitle	bookAuthor	bookGenre	publishedYear	createdBy	createdDate	updatedBy	updatedDate	hashkey
102	The Guide	R.K. Narayan	Fiction	1958	DataFlow	2025-03-14T14:59:34.6970000	DataFlow	2025-03-14T14:59:34.6970000	3463441493
101	The White Tiger	Aravind Adiga	Fiction	2008	DataFlow	2025-03-14T14:59:34.6970000	DataFlow	2025-03-14T14:59:34.6970000	1216988737
103	Chetan Bhagat - One Indian Girl	Chetan Bhagat	Romance	2016	DataFlow	2025-03-14T14:59:34.6970000	DataFlow	2025-03-14T14:59:34.6970000	1660944415
104	The God of Small Things	Arundhati Roy	Fiction	1997	DataFlow	2025-03-14T14:59:34.6970000	DataFlow	2025-03-14T14:59:34.6970000	2398991994

Members:

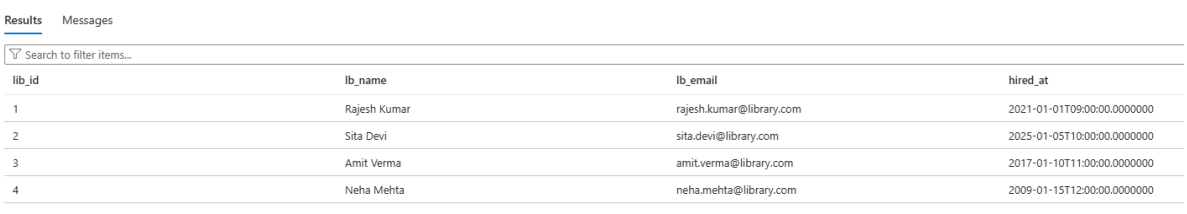


Results Messages

Search to filter items...

memberId	memberName	memberEmail	memberPhone	createdBy	createdDate	updatedBy	updatedDate	hashkey	isActive
2	Priya Patel	priya.patel@yahoo.com	9988776655	DataFlow	2025-03-14T15:00:27.3030000	DataFlow	2025-03-14T15:00:27.3030000	692503268	1
4	Ananya Gupta	ananya.gupta@rediffmail.com	9998887770	DataFlow	2025-03-14T15:00:27.3030000	DataFlow	2025-03-14T15:00:27.3030000	3277880304	1
3	Vikram Kumar	vikram.kumar@outlook.com	9871234567	DataFlow	2025-03-14T15:00:27.3030000	DataFlow	2025-03-14T15:00:27.3030000	3710560039	1
1	Rahul Sharma	rahul.sharma@gmail.com	9876543210	DataFlow	2025-03-14T15:00:27.3030000	DataFlow	2025-03-14T15:00:27.3030000	1404072131	1

Library:



Results Messages

Search to filter items...

lib_id	lb_name	lb_email	hired_at
1	Rajesh Kumar	rajesh.kumar@library.com	2021-01-01T09:00:00.0000000
2	Sita Devi	sita.devi@library.com	2025-01-05T10:00:00.0000000
3	Amit Verma	amit.verma@library.com	2017-01-10T11:00:00.0000000
4	Neha Mehta	neha.mehta@library.com	2009-01-15T12:00:00.0000000

5. After Second Run Review the Output.

Insert queries used to update/Insert in every type of tables:

---Insert/Update in Members table
Insert Into Members (member_id, member_name, member_email, member_phone, last_updated)
Values
(5,'Virat Kohli', 'virat.kohli@gmail.com', 9876543210, '2025-03-08 00:00:00'),
(6,'MS Dhoni', 'ms.dhoni@gmail.com', 9876501234, '2025-03-08 00:00:00')

Update Members Set member_phone=9999543210,
member_email='rahul.44sharma@gmail.com', last_updated='2025-03-08 00:00:00'
where member_id=1
Update Members Set member_email='ananya.gupta0555@rediffmail.com',
last_updated='2025-03-09 00:00:00'
where member_id=4

---Insert/Update in Books table
INSERT INTO Books (book_id, book_title, book_author, book_genre, published_year, last_updated)
VALUES
(105,'The Alchemist', 'Paulo Coelho', 'Philosophical Fiction', 1988, '2025-03-08 00:00:00'),
(106,'Life of Pi', 'Yann Martel', 'Adventure', 2001, '2025-03-08 00:00:00');

Update Books Set book_genre='Fiction2', published_year=2000, last_updated='2025-03-08 00:00:00'
where book_id=101
Update Books Set book_genre='Fiction2', published_year=1950, last_updated='2025-03-08 00:00:00'
where book_id=102

---Insert in Librarians table
Insert Into Librarians (lib_id, hired_at, lb_name, lb_email)
Values
(5, '2021-01-01 09:00:00', 'Ravish Singh Kumar', 'r.kumar@library.com')

```sql
-----------------------------------------------
---Insert/Update in Members table
Insert Into Members (member_id, member_name, member_email, member_phone, last_updated )
Values
(5,'Virat Kohli', 'virat.kohli@gmail.com', 9876543210, '2025-03-08 00:00:00'),
(6,'MS Dhoni', 'ms.dhoni@gmail.com', 9876501234, '2025-03-08 00:00:00')

Update Members Set member_phone=9999543210, member_email='rahul.44sharma@gmail.com', last_updated='2025-03-08 00:00:00'
where member_id=1
Update Members Set member_email='ananya.gupta0555@rediffmail.com', last_updated='2025-03-09 00:00:00'
where member_id=4


---Insert/Update in Books table
INSERT INTO Books (book_id, book_title, book_author, book_genre, published_year, last_updated)
VALUES
(105,'The Alchemist', 'Paulo Coelho', 'Philosophical Fiction', 1988, '2025-03-08 00:00:00'),
(106,'Life of Pi', 'Yann Martel', 'Adventure', 2001, '2025-03-08 00:00:00');

Update Books Set book_genre='Fiction2', published_year=2000, last_updated='2025-03-08 00:00:00'
where book_id=101
Update Books Set book_genre='Fiction2', published_year=1950, last_updated='2025-03-08 00:00:00'
where book_id=102


---Insert in Librarians table
Insert Into Librarians (lib_id, hired_at, lb_name, lb_email)
Values
(5, '2021-01-01 09:00:00', 'Ravish Singh Kumar', 'r.kumar@library.com')
```

Query Output:

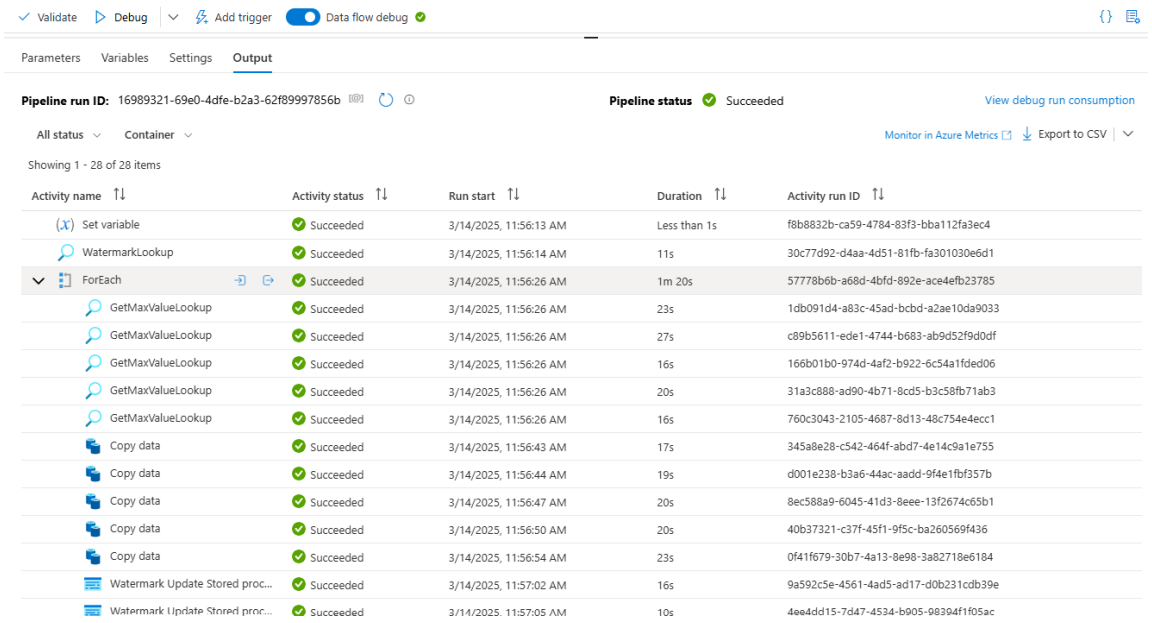
Results | Messages

	book_id	book_title	book_author	book_genre	published_year	last_updated
1	101	The White Tiger	Aravind Adiga	Fiction2	2000	2025-03-08 00:00:00.000
2	102	The Guide	R.K. Narayan	Fiction2	1950	2025-03-08 00:00:00.000
3	103	Chetan Bhag...	Chetan Bha...	Romance	2016	2025-03-07 00:00:00.000
4	104	The God of S...	Arundhati R...	Fiction	1997	2025-03-07 00:00:00.000
5	105	The Alchemist	Paulo Coelho	Philosophi...	1988	2025-03-08 00:00:00.000
6	106	Life of Pi	Yann Martel	Adventure	2001	2025-03-08 00:00:00.000

	member_id	member_name	member_email	member_phone	last_updated
1	1	Rahul Sharma	rahul.44sharma@gmail.com	9999543210	2025-03-08 00:00:00.000
2	2	Priya Patel	priya.patel@yahoo.com	9988776655	2025-03-07 00:00:00.000
3	3	Vikram Kumar	vikram.kumar@outlook.com	9871234567	2025-03-07 00:00:00.000
4	4	Ananya Gupta	ananya.gupta0555@rediff...	9998887770	2025-03-09 00:00:00.000
5	5	Virat Kohli	virat.kohli@gmail.com	9876543210	2025-03-08 00:00:00.000
6	6	MS Dhoni	ms.dhoni@gmail.com	9876501234	2025-03-08 00:00:00.000

	lib_id	lb_name	lb_email	hired_at
1	1	Rajesh Kumar	rajesh.kumar@library.com	2021-01-01 09:00:00.000
2	2	Sita Devi	sita.devi@library.com	2025-01-05 10:00:00.000
3	3	Amit Verma	amit.verma@library.com	2017-01-10 11:00:00.000
4	4	Neha Mehta	neha.mehta@library.com	2009-01-15 12:00:00.000
5	5	Ravish Sing...	r.kumar@library.com	2021-01-01 09:00:00.000

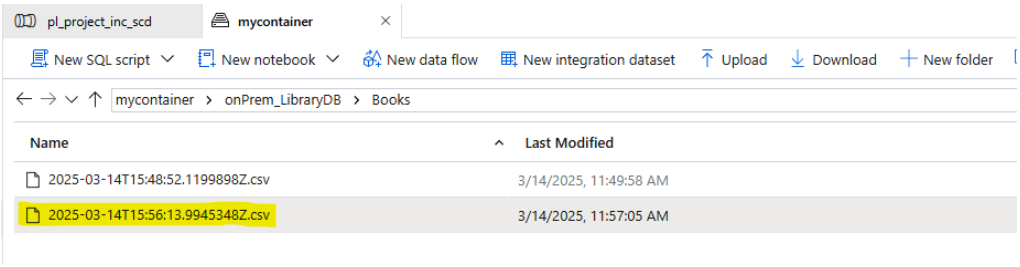
2nd Run Execution Logs:

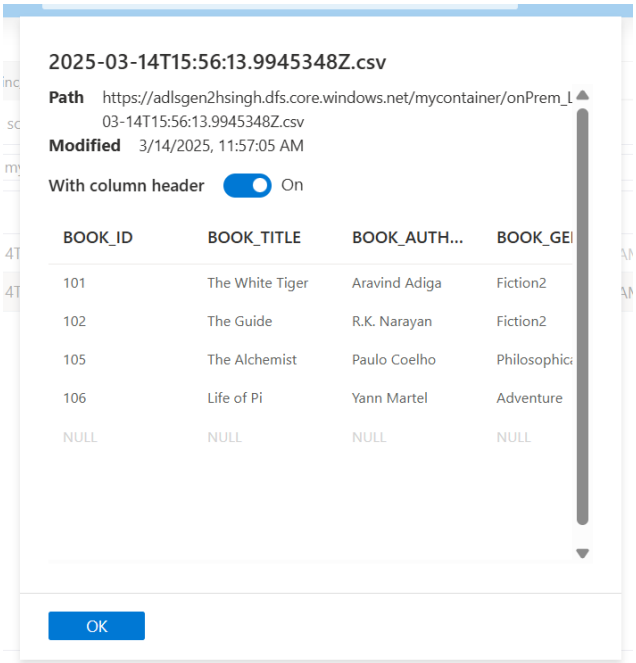


Activity name	Activity status	Run start	Duration	Activity run ID
Set variable	Succeeded	3/14/2025, 11:56:13 AM	Less than 1s	f8b8832b-ca59-4784-83f3-bba112fa3ec4
WatermarkLookup	Succeeded	3/14/2025, 11:56:14 AM	11s	30c77d92-d4aa-4d51-81fb-fa301030e6d1
ForEach	Succeeded	3/14/2025, 11:56:26 AM	1m 20s	57778b6b-a68d-4bfd-892e-ace4efb23785
GetMaxValueLookup	Succeeded	3/14/2025, 11:56:26 AM	23s	1db091d4-a83c-45ad-bcbd-a2ae10da9033
GetMaxValueLookup	Succeeded	3/14/2025, 11:56:26 AM	27s	c89b5611-ede1-4744-b683-ab9d52f9d0df
GetMaxValueLookup	Succeeded	3/14/2025, 11:56:26 AM	16s	166b01b0-974d-4af2-b922-6c54a1fded06
GetMaxValueLookup	Succeeded	3/14/2025, 11:56:26 AM	20s	31a3c888-ad90-4b71-8cd5-b3c58fb71ab3
GetMaxValueLookup	Succeeded	3/14/2025, 11:56:26 AM	16s	760c3043-2105-4687-8d13-48c754e4ecc1
Copy data	Succeeded	3/14/2025, 11:56:43 AM	17s	345a8e28-c542-464f-abd7-4e14c9a1e755
Copy data	Succeeded	3/14/2025, 11:56:44 AM	19s	d001e238-b3a6-44ac-aadd-9f4e1fbf357b
Copy data	Succeeded	3/14/2025, 11:56:47 AM	20s	8ec588a9-6045-41d3-8eee-13f2674c65b1
Copy data	Succeeded	3/14/2025, 11:56:50 AM	20s	40b37321-c37f-45f1-9f5c-ba260569f436
Copy data	Succeeded	3/14/2025, 11:56:54 AM	23s	0f41f679-30b7-4a13-8e98-3a82718e6184
Watermark Update Stored proc...	Succeeded	3/14/2025, 11:57:02 AM	16s	9a592c5e-4561-4ad5-ad17-d0b231cdb39e
Watermark Update Stored proc...	Succeeded	3/14/2025, 11:57:05 AM	10s	4ee4dd15-7d47-4534-b905-98394f1f05ac

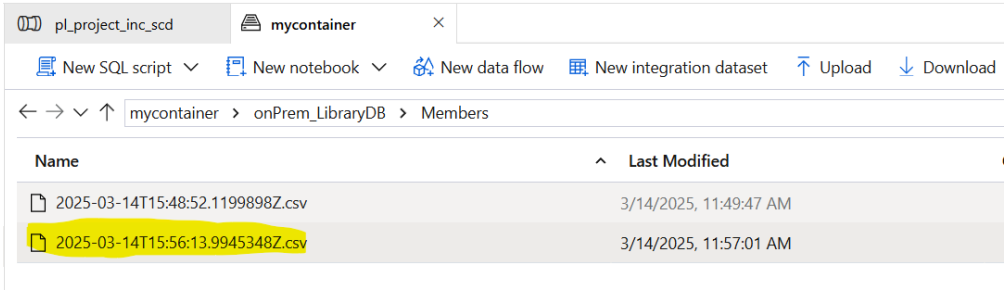
Review the new CSV Output Files in ADLS Gen2.

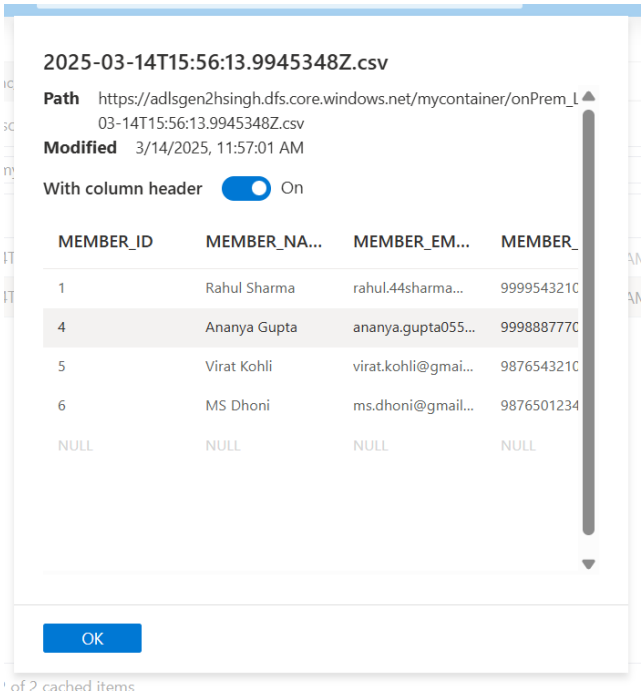
Books new csv File:



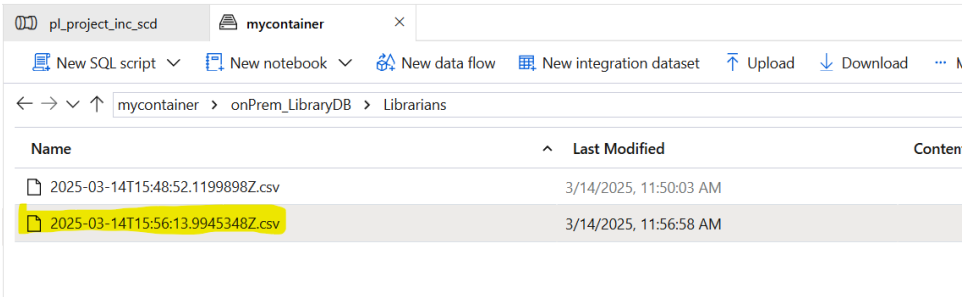


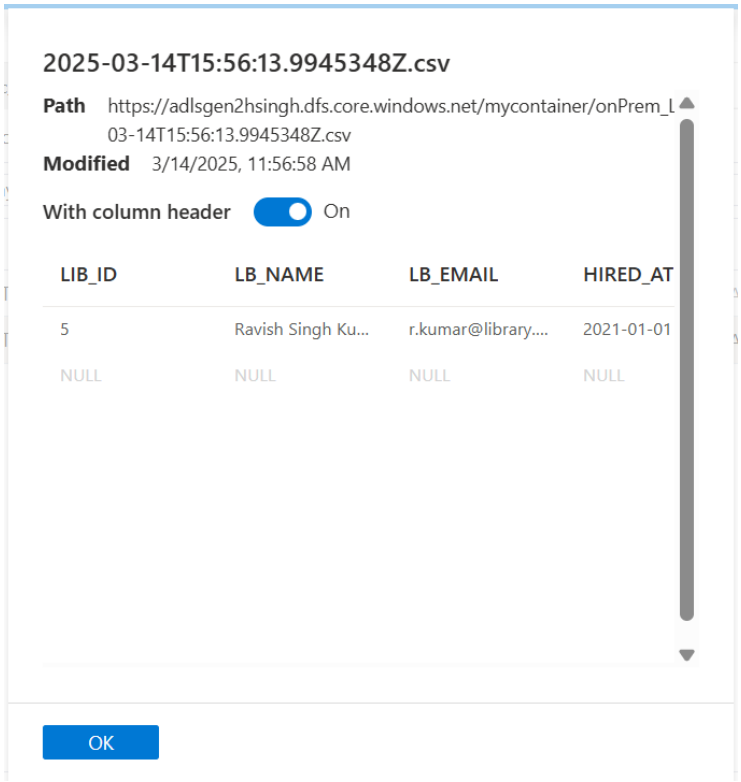
Members new csv file:





Librarians new csv file (**Note**: the empty files were generated as named as Current timestamp since there are no updates for other two table):





Review the tables in Azure SQL DB.

For Books (Books_SCDTYPE1):

Results Messages

Search to filter items...

bookId	bookTitle	bookAuthor	bookGenre	publishedYear	createdBy	createdDate	updatedBy	updatedDate	hashkey
104	The God of Small Things	Arundhati Roy	Fiction	1997	DataFlow	2025-03-14T15:51:34.0670000	DataFlow	2025-03-14T15:51:34.0670000	2398991994
103	Chetan Bhagat - One Indian Girl	Chetan Bhagat	Romance	2016	DataFlow	2025-03-14T15:51:34.0670000	DataFlow	2025-03-14T15:51:34.0670000	1660944415
102	The Guide	R.K. Narayan	Fiction2	1950	DataFlow	2025-03-14T15:51:34.0670000	Dataflow-Updated	2025-03-14T16:01:42.1730000	3518283936
101	The White Tiger	Aravind Adiga	Fiction2	2000	DataFlow	2025-03-14T15:51:34.0670000	Dataflow-Updated	2025-03-14T16:01:42.1730000	2543347042
106	Life of Pi	Yann Martel	Adventure	2001	DataFlow	2025-03-14T16:01:23.2600000	DataFlow	2025-03-14T16:01:23.2600000	4060193662
105	The Alchemist	Paulo Coelho	Philosophical Fiction	1988	DataFlow	2025-03-14T16:01:23.2600000	DataFlow	2025-03-14T16:01:23.2600000	1701707856

For Members (Members_SCDTYPE2):

Results Messages

Search to filter items...

memberId	memberName	memberEmail	memberPhone	createdBy	createdDate	updatedBy	updatedDate	hashkey	isActive
1	Rahul Sharma	rahul.sharma@gmail.com	9876543210	DataFlow	2025-03-14T15:52:46.2530000	DataFlow-Updated	2025-03-14T16:01:50.8570000	1404072131	0
3	Vikram Kumar	vikram.kumar@outlook.com	9871234567	DataFlow	2025-03-14T15:52:46.2530000	DataFlow	2025-03-14T15:52:46.2530000	3710560039	1
4	Ananya Gupta	ananya.gupta@rediffmail.com	9998887770	DataFlow	2025-03-14T15:52:46.2530000	DataFlow-Updated	2025-03-14T16:01:50.8570000	3277880304	0
2	Priya Patel	priya.patel@yahoo.com	9988776655	DataFlow	2025-03-14T15:52:46.2530000	DataFlow	2025-03-14T15:52:46.2530000	692503268	1
6	MS Dhoni	ms.dhoni@gmail.com	9876501234	DataFlow	2025-03-14T16:02:09.0530000	DataFlow	2025-03-14T16:02:09.0530000	2461897584	1
5	Virat Kohli	virat.kohli@gmail.com	9876543210	DataFlow	2025-03-14T16:02:09.0530000	DataFlow	2025-03-14T16:02:09.0530000	2791174883	1
1	Rahul Sharma	rahul.44sharma@gmail.com	9999543210	DataFlow	2025-03-14T16:02:09.0530000	DataFlow	2025-03-14T16:02:09.0530000	1752211969	1
4	Ananya Gupta	ananya.gupta0555@rediffmail...	9998887770	DataFlow	2025-03-14T16:02:09.0530000	DataFlow	2025-03-14T16:02:09.0530000	3582788583	1

For Librarians:

Results Messages

Search to filter items...

lib_id	lb_name	lb_email	hired_at
1	Rajesh Kumar	rajesh.kumar@library.com	2021-01-01T09:00:00.0000000
2	Sita Devi	sita.devi@library.com	2025-01-05T10:00:00.0000000
3	Amit Verma	amit.verma@library.com	2017-01-10T11:00:00.0000000
4	Neha Mehta	neha.mehta@library.com	2009-01-15T12:00:00.0000000
5	Ravish Singh Kumar	r.kumar@library.com	2021-01-01T09:00:00.0000000

For BorrowRecords:

Results Messages

Search to filter items...

borrow_date	book_id	member_id	return_days
2025-01-01T10:00:00.0000000	101	1	20
2025-01-05T14:00:00.0000000	102	2	15
2025-01-10T16:00:00.0000000	103	3	7
2025-01-12T09:30:00.0000000	101	4	10

For Inventory:

Results Messages

Search to filter items...

last_updated	book_id	quantity
2025-01-01T10:00:00.0000000	1	15
2025-01-05T14:00:00.0000000	2	20
2025-01-10T16:00:00.0000000	3	25
2025-01-12T09:30:00.0000000	4	30