



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	2
2. Watermark Table	6
3. Stored Procedure for Incremental Data Load	8
4. Dynamic Pipeline Design	9
5. After Second Run Review the Output.....	57

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 identifier**:

- **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	lib_name	lib_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)
)
```

```
]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 Column 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

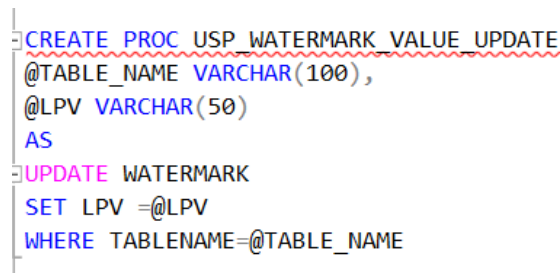
A **Stored Procedure** in **Incremental Data Load** in **ADF and Synapse** is used to **update the last processed values (watermark) to track new records**.

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
```



```
1 CREATE PROC USP_WATERMARK_VALUE_UPDATE
2 @TABLE_NAME VARCHAR(100),
3 @LPV VARCHAR(50)
4 AS
5 UPDATE WATERMARK
6 SET LPV =@LPV
7 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:

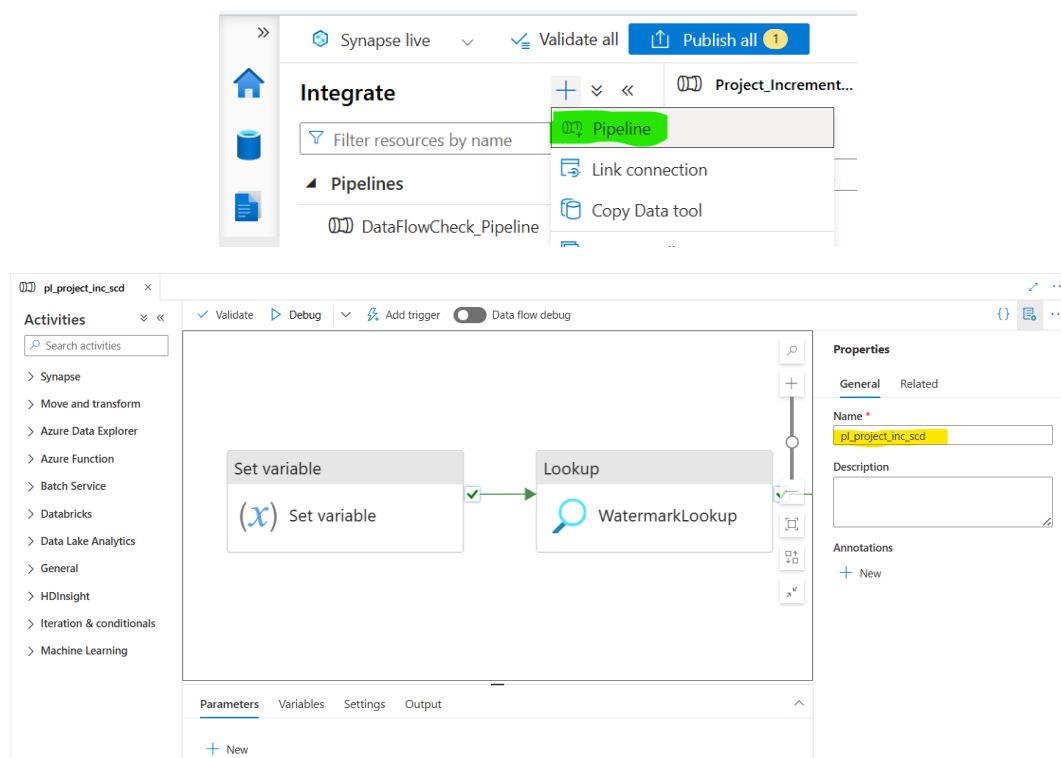
- Set Variable Activity,
- 2 Lookup Activities,
- Foreach Activity,
- 2 Copy Activities,
- IF Condition Activity,
- Store Procedure Activity

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.

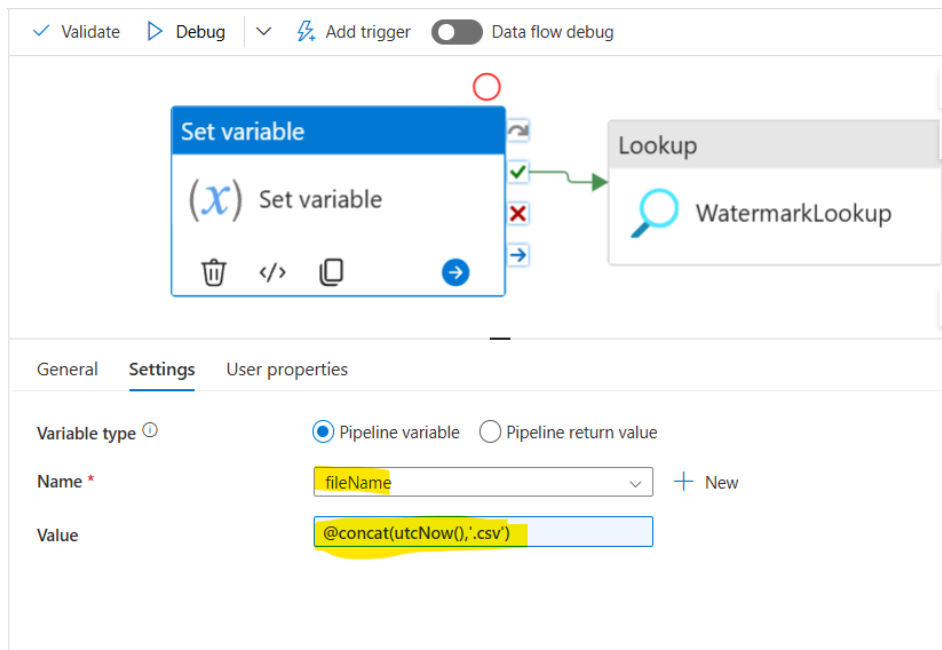
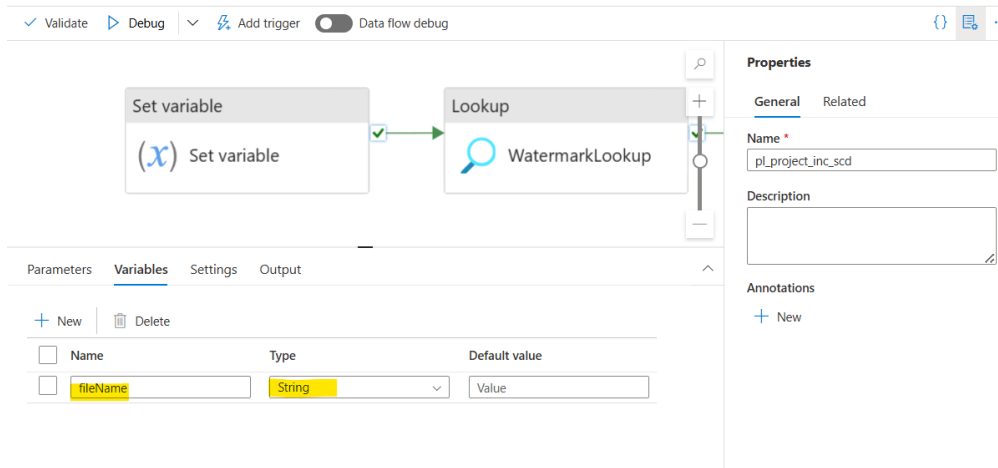
- Create a new Pipeline and Drag a **Set Variable Activity** and a **Lookup Activity**. And Connect **Set Variable Activity** with on success connection of **Lookup Activity**.
- Set up the names for both as per the project.
- Here I have named the pipeline as *pl_project_inc_scd* and Lookup activity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

Lookup WatermarkLookup

Properties

General Related

Name * pl_project_inc_scd

Description

Annotations + New

General Settings ¹ User properties

Source dataset * Select... + New

First row only ☐

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** username as **superadmin**).
- To create new linked service for on-prem SQL Server DB, follow below configurations as shown:

Edit linked service

SQL server [Learn more](#)

Name *
ls_onprem_sqldb

Description

Connect via integration runtime * ⓘ
☒ Self-HostedIntegrationRuntime [Edit](#)

⚠ The credentials are stored in the machines of self-hosted integration runtime if you don't choose to store them in Azure Key Vault.

Version
☐ Recommended ☒ Legacy

[Connection string](#) [Azure Key Vault](#)

Server name *
LAPTOP-2KFBB2GF\SQLEXPRESS

Database name *
mylocaldb

Authentication type
SQL authentication

User name *
superadmin

[Password](#) [Azure Key Vault](#)

Password *

[Apply](#) [Cancel](#)

✓ Connection successful
[Test connection](#)

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

pl_project_inc_scd ds_onprem_sqldb

SQL server
ds_onprem_sqldb

Connection Schema **Parameters**

+ New Delete

<input type="checkbox"/>	Name	Type	Default value	
<input type="checkbox"/>	Schema_Name	String	Value	
<input type="checkbox"/>	Table_Name	String	Value	

- 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***

pl_project_inc_scd ds_onprem_sqldb

SQL server
ds_onprem_sqldb

Connection Schema Parameters

Linked service * Is_onprem_sqldb Test connection Edit + New Learn more

Integration runtime * SelfHostedIntegrationRuntime Edit

Table @dataset().Schema_Name . @dataset().Table_Name Preview data

Enter manually

Pipeline expression builder

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

@dataset().Table_Name

Clear contents

Parameters Functions

Search +

Schema_Name

Table_Name

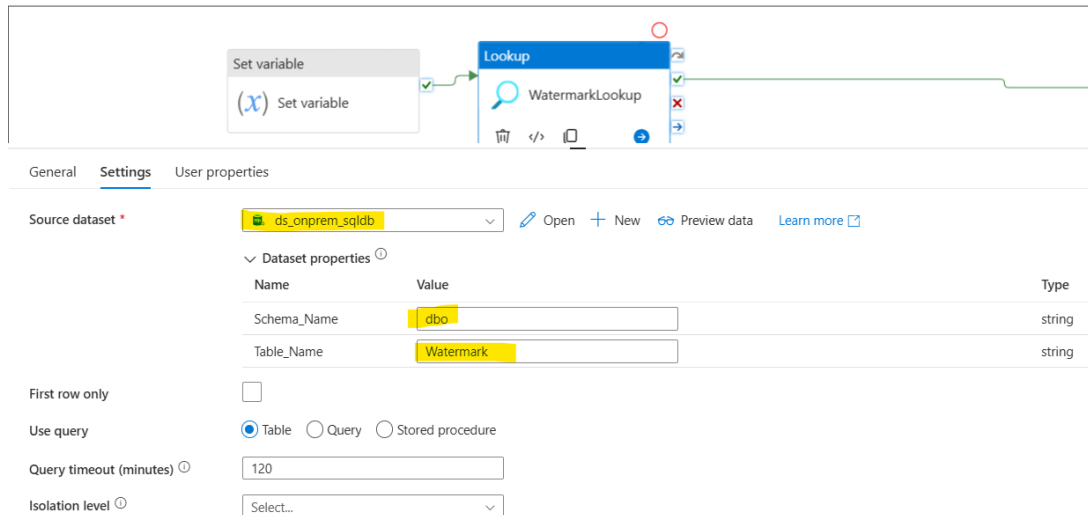
click to insert expression

OK Cancel

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

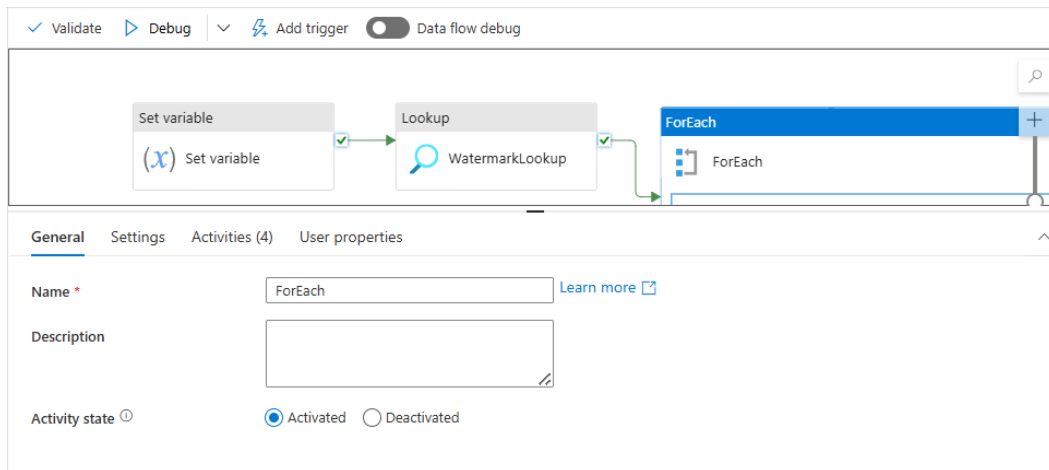
Schema_Name: **dbo**

Table_Name: **Watermark**



STEP 2: Setup For Each Activity

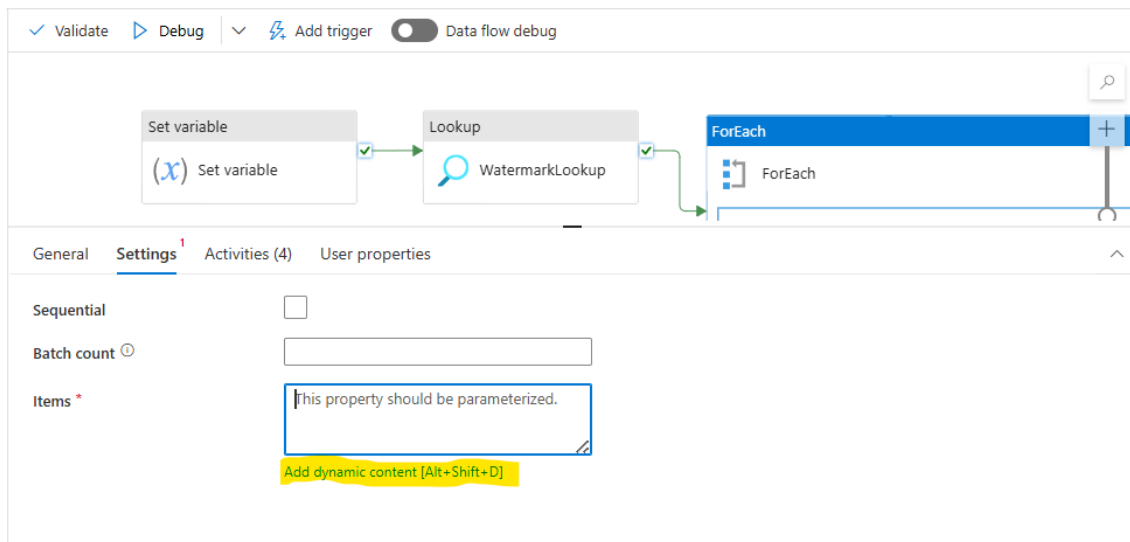
- After creating the Lookup, add the Foreach Activity and connect the Lookup with it using the on-success connection.



- 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



Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

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

[Clear contents](#)

Activity outputs

Parameters

System variables

Functions

Variables

Copy data

Copy data activity output

GetMaxValueLookup

GetMaxValueLookup activity output

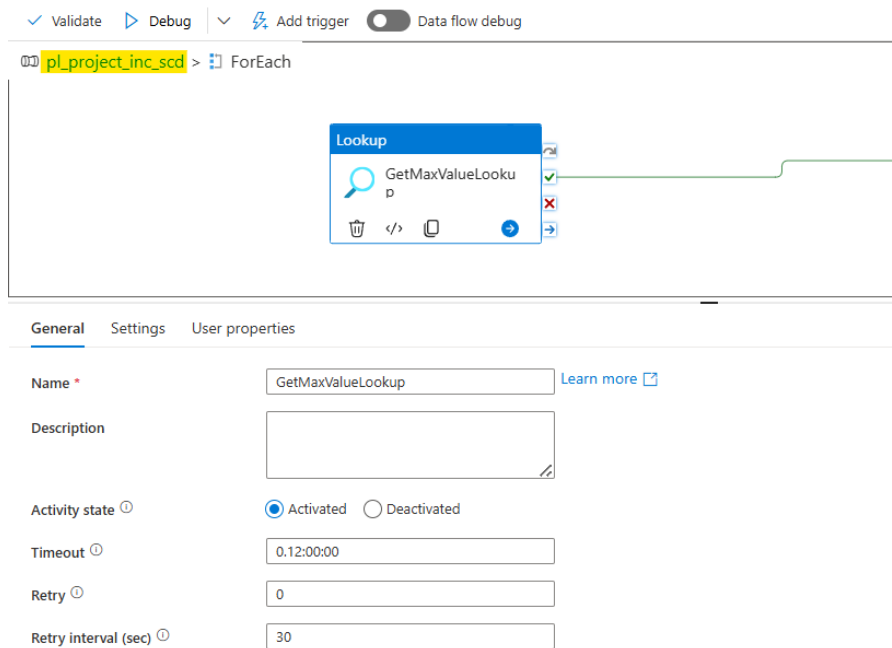
GetMaxValueLookup first row

OK

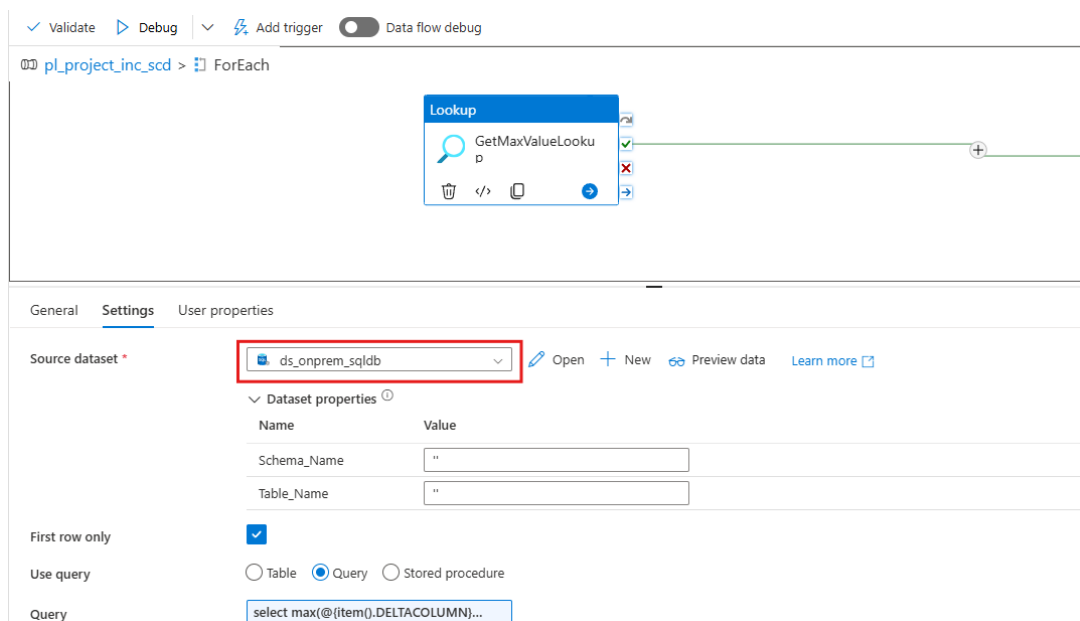
Cancel

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}}

Dataset properties ⓘ

Name	Value	Type
Schema_Name	<input type="text" value=""/>	string
Table_Name	<input type="text" value=""/>	string

First row only ☒

Use query ☐ Table ☒ Query ☐ Stored procedure

Query *

Add dynamic content [Alt+Shift+D]

Query timeout (minutes) ⓘ

Pipeline expression builder ⓘ

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

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

Clear contents

ForEach iterator

Activity outputs

Parameters

System variables

...

ForEach

Current item

OK

Cancel

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}'**

✓ Validate ✓ Validate copy runtime ▶ Debug ▼ ⚡ Add trigger ⏻ Data flow debug

pl_project_inc_scd > ⚙️ ForEach

Lookup

GetMaxValueLookup

Copy data

Copy data

General **Source** Sink Mapping Settings User properties

Source dataset * ds_onprem_sqldb Open + New 🔗 Preview data [Learn more](#)

Dataset properties ⓘ

Name	Value
Schema_Name	"
Table_Name	"

Use query ☐ Table ☒ Query ☐ Stored procedure

Query `SELECT * FROM @{item().TABLENAM...`

Query timeout (minutes) ⓘ 120

Isolation level ⓘ Select...

Partition option ⓘ ☒ None ☐ Physical partitions of table ⓘ ☐ Dynamic range ⓘ

ⓘ Please preview data to validate the partition settings.

Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

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

[Clear contents](#)

Activity outputs

Parameters

System variables

Functions

Variables

Search

GetMaxValueLookup
GetMaxValueLookup activity output

GetMaxValueLookup first row
Data of the first row

WatermarkLookup

OK

Cancel

- 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.

✓ Validate ✓ Validate copy runtime ▶ Debug ▼ ⚡ Add trigger ⏻ Data flow debug

pl_project_inc_scd > ForEach

Sink Mapping Settings User properties

Sink dataset * ds_ADLS_Sink [Open](#) [+ New](#) [Learn more](#)

Dataset properties ⓘ

Name	Value
Folder_Name	Value
File_Name	Value

Copy behavior ⓘ Select...

Max concurrent connections ⓘ

Block size (MB) ⓘ

Metadata ⓘ [+ New](#)

- 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**.

DelimitedText
ds_ADLS_Sink

Connection Schema **Parameters**

[+ New](#) [Delete](#)

<input type="checkbox"/>	Name	Type	Default value	
<input type="checkbox"/>	Folder_Name	String	Value	Delete
<input type="checkbox"/>	File_Name	String	Value	Delete

DelimitedText
ds_ADLS_Sink

Connection Schema Parameters

Linked service * wspp-ncpl-singhh-WorkspaceDefaultSw [Test connection](#) [Edit](#) [+ New](#) [Learn more](#)

Integration runtime * AutoResolveIntegrationRuntime [Edit](#)

File path mycontainer @dataset().Folder_Name / @dataset().File_Name [Browse](#) [Preview data](#)

Compression type No compression

Column delimiter Comma (,)

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

Encoding Default(UTF-8)

Quote character Double quote (")

Escape character Backslash (\)

- 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.

✓ Validate ✓ Validate copy runtime Debug Add trigger Data flow debug

pl_project_inc_scd > ForEach

Lookup GetMaxValueLooku

Copy data Copy data

General Source Sink Mapping Settings User properties

Sink dataset * ds_ADLS_Sink [Open](#) [+ New](#) [Learn more](#)

Dataset properties

Name	Value
Folder_Name	@item().FOLDERNAME
File_Name	@variables('fileName')

Copy behavior Select...

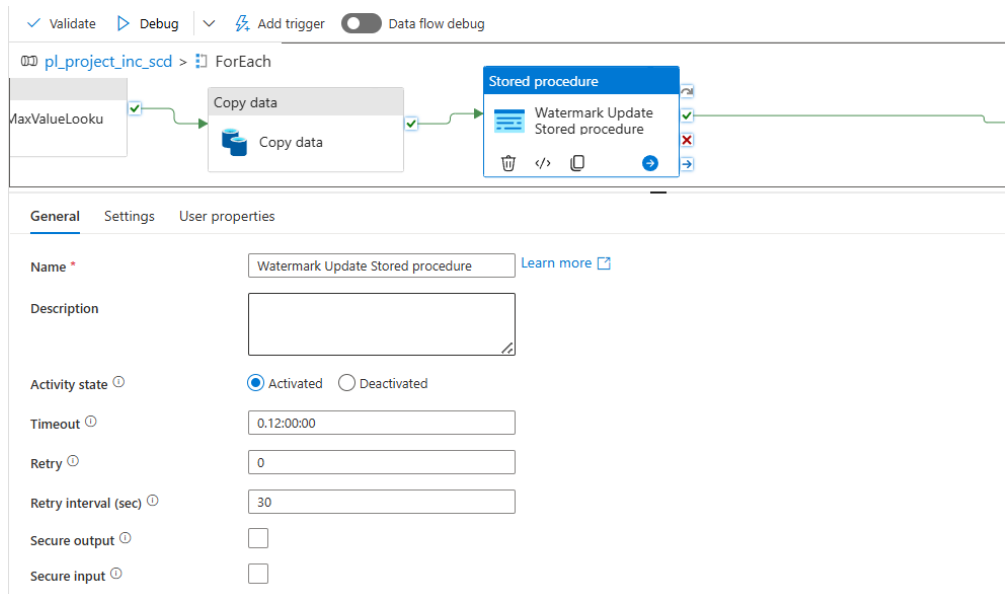
Max concurrent connections

Block size (MB)

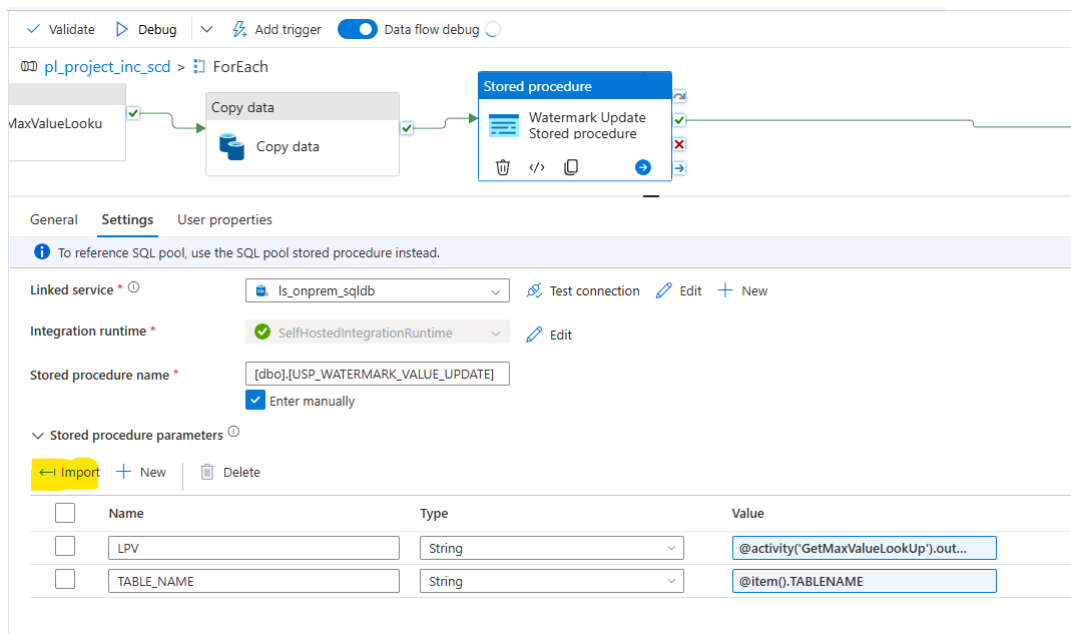
Metadata + New

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

General Settings User properties

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

Linked service * Test connection Edit + New

Integration runtime * Edit

Stored procedure name * Enter manually

Stored procedure parameters

Name	Type	Value
LPV	String	@activity('GetMaxValueLookUp').out...
TABLE_NAME	String	@item().TABLENAME

Pipeline expression builder

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

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

[Clear contents](#)

ForEach iterator Activity outputs Parameters System variables ...

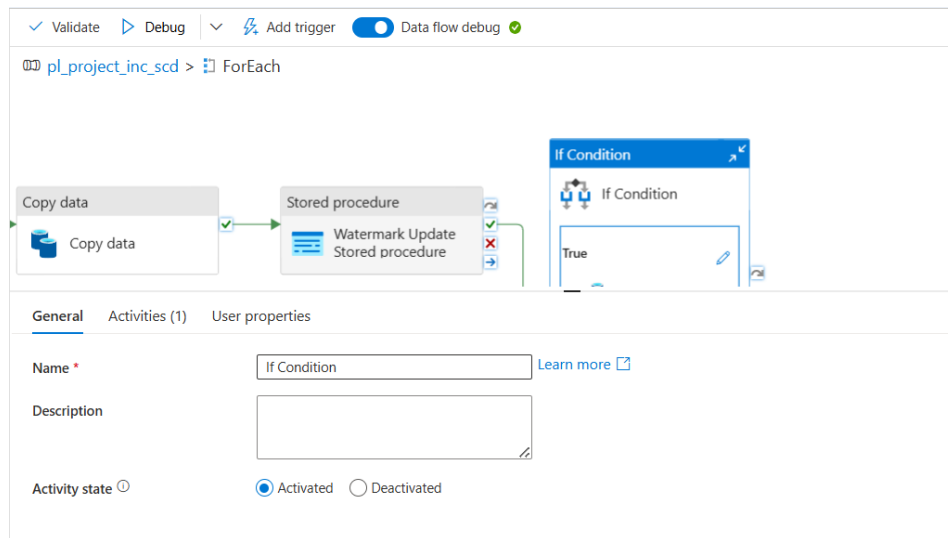
ForEach
Current item

OK

Cancel

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.

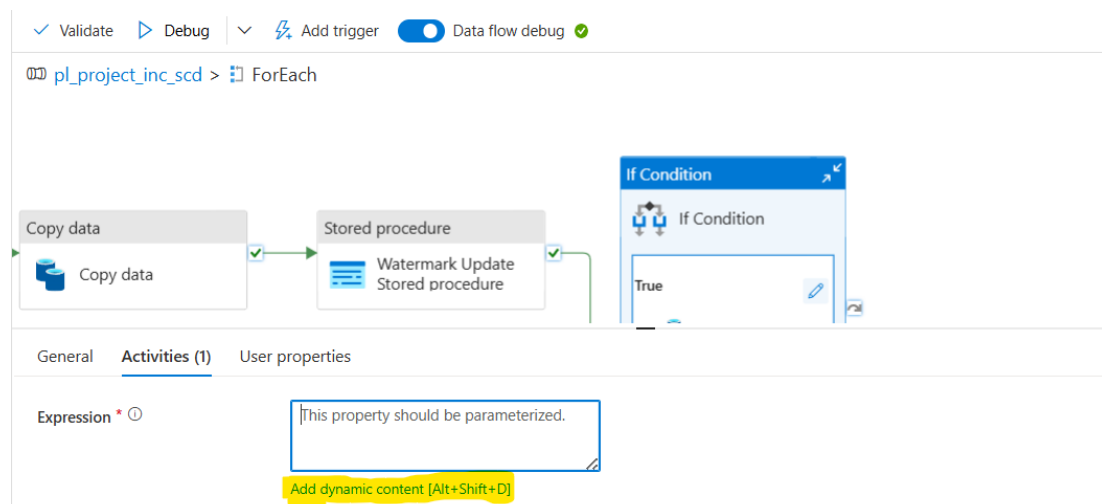


- Then, Go to Activities tab of IF Condition, set the Expression by adding it using Add dynamic content option as highlighted below.
- This IF Condition only allows the non-SCD Type tables to normally load generated csv Files data to Azure SQL Database from ADLS Gen2.

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**).



Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

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

[Clear contents](#)

ForEach iterator Activity outputs Parameters System variables ...

Search

ForEach
Current item

OK **Cancel**

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.

✓ Validate ▶ Debug ⚙ Add trigger 🔍 Data flow debug ✓

pl_project_inc_scd > ForEach

Stored procedure
Watermark Update
Stored procedure



IF Condition

True
Copy data to Azure...

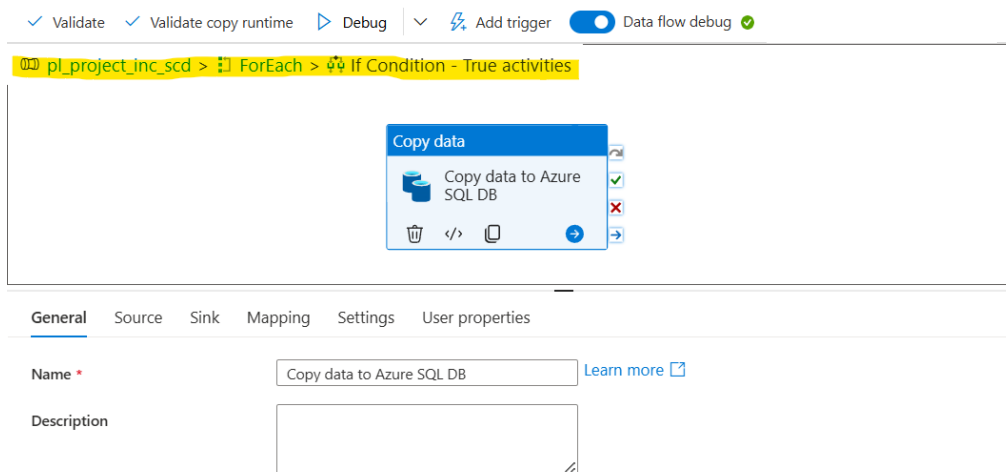
False

General **Activities (1)** User properties

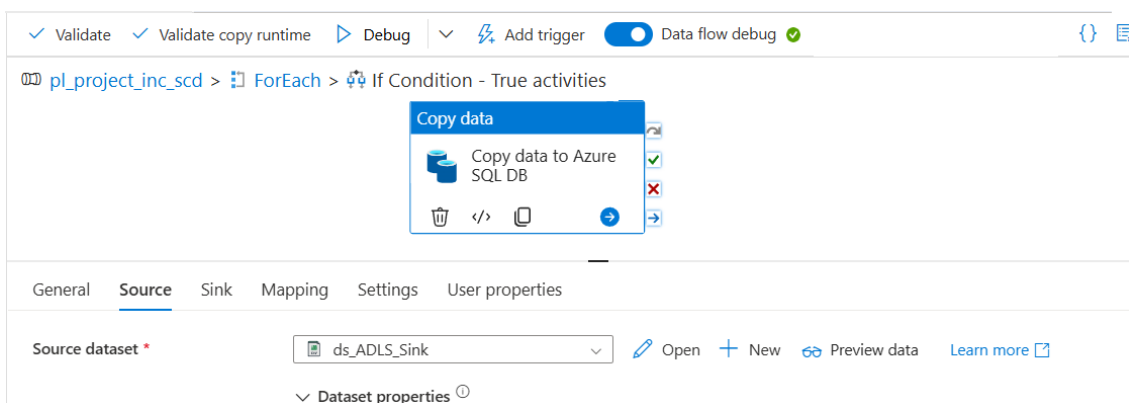
Expression ⓘ @and(not(equals(item().TableName, ...

Case	Activity	
True	Copy data to Azu...	
	1 Activity	
False	No activities	

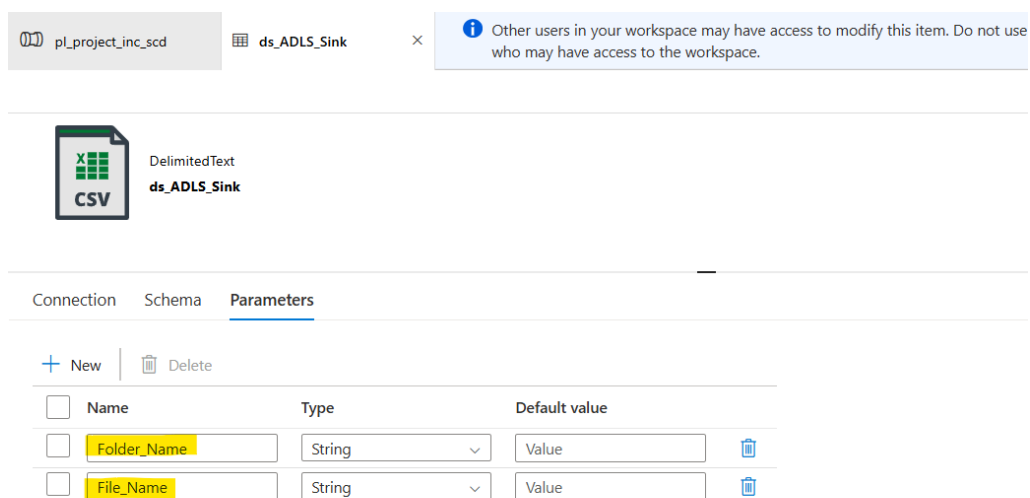
- After Going to true section, Add Copy Activity in it.



- 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***

pl_project_inc_scd ds_ADLS_Sink

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

DelimitedText
ds_ADLS_Sink

CSV

Connection Schema Parameters

Linked service * wsp-ncpl-singhh-WorkspaceDefaultSvc Test connection Edit + New Learn more

Integration runtime * AutoResolveIntegrationRuntime Edit

File path mycontainer / @dataset().Folder_Name / @dataset().File_Name Browse

Compression type No compression

Column delimiter (.)

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

- 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')***

Validate Validate copy runtime Debug Add trigger Data flow debug

pl_project_inc_scd > ForEach > If Condition - True activities

Copy data

Copy data to Azure SQL DB

General Source Sink Mapping Settings User properties

Source dataset * ds_ADLS_Sink Open + New Preview data Learn more

Dataset properties

Name	Value	Type
Folder_Name	@item().FolderName	string
File_Name	@variables('fileName')	string

File path type File path in dataset Wildcard file path List of files

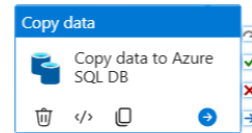
Start time (UTC) End time (UTC)

Filter by last modified

Recursively

- 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.

pl_project_inc_scd > ForEach > If Condition - True activities



General Source **Sink** Mapping Settings User properties

Sink dataset * ds_SqlDB_Source [Open](#) [+ New](#) [Learn more](#)

Dataset properties ⓘ

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



Connection Schema **Parameters**

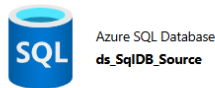
[+ New](#) [Delete](#)

<input type="checkbox"/>	Name	Type	Default value	
<input type="checkbox"/>	Schema_Name	String	Value	Delete
<input type="checkbox"/>	Table_Name	String	Value	Delete

- 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**



Connection Schema Parameters

Linked service * Is_SqlDB_connection [Test connection](#) [Edit](#) [+ New](#) [Learn more](#)

Integration runtime * AutoResolveIntegrationRuntime [Edit](#)

Table @dataset().Schema_Name @dataset().Table_Name [Preview data](#)

☒ Enter manually

- 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**

pl_project_inc_scd > ForEach > If Condition - True activities

Copy data

Copy data to Azure SQL DB

General Source Sink Mapping Settings User properties

Sink dataset * ds_SqlDB_Source [Open](#) [+ New](#) [Learn more](#)

Dataset properties

Name	Value
Schema_Name	@item().SchemaName
Table_Name	@item().TableName

Write behavior ☒ Insert ☐ Upsert ☐ Stored procedure

Bulk insert table lock ☐ Yes ☒ No

Table option ☒ Use existing ☐ Auto create table

- 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

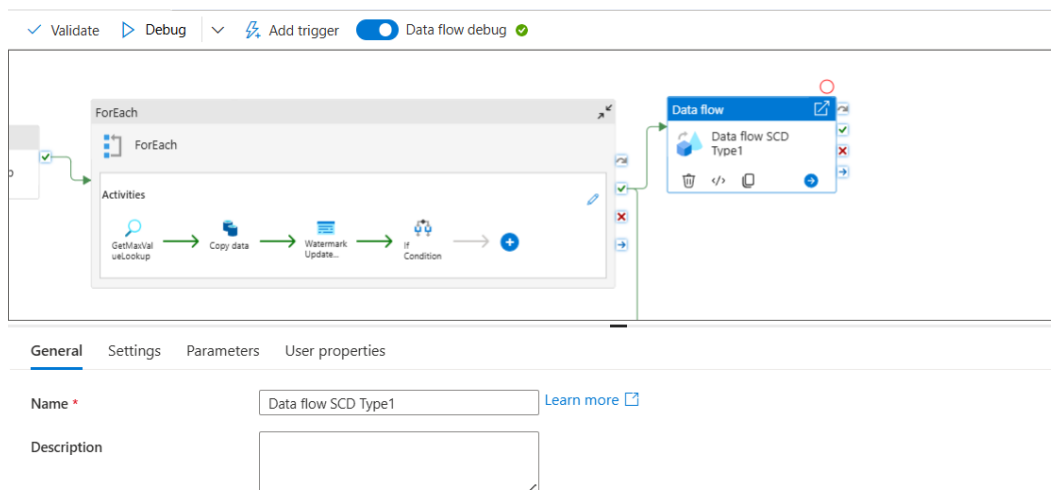
- Here, I implemented SCD Type 1 data load for table **Books** to load the data in **Azure SQL DB** from **ADLS Gen2**.
- To load the **Books** Data in Azure SQL DB, we need to create table definition in **Azure SQL DB database** as shown below:

Create Table Books_SCDTYPE1

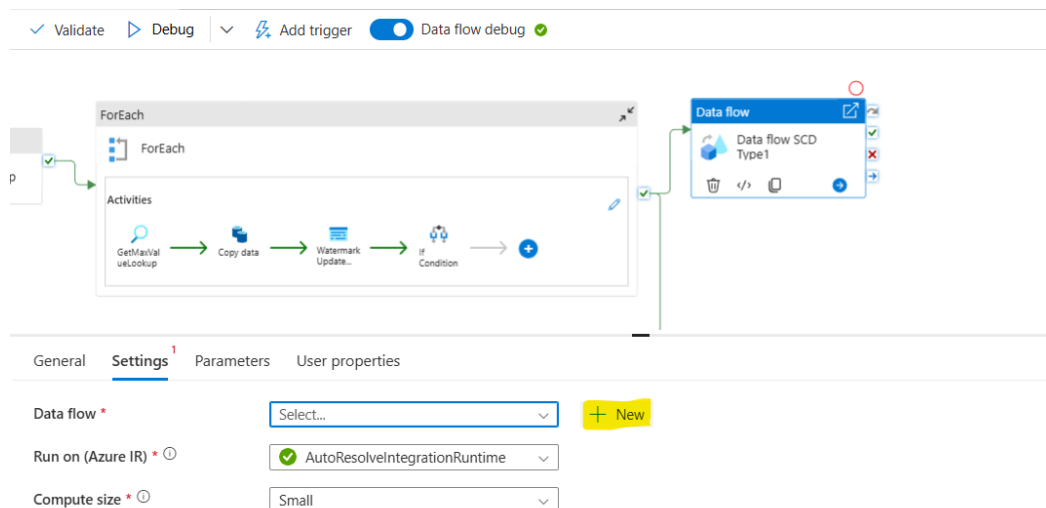
```
(
    bookId Int,
    bookTitle Varchar(255),
    bookAuthor Varchar(255),
    bookGenre Varchar(100),
    publishedYear Int,
    createdBy varchar(100),createdDate datetime,
    updatedBy varchar(100),updatedAt datetime,
    hashkey Bigint,
)
select * from Books_SCDTYPE1
```

```
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),updatedAt datetime,hashkey Bigint,
)
select * from Books_SCDTYPE1
```

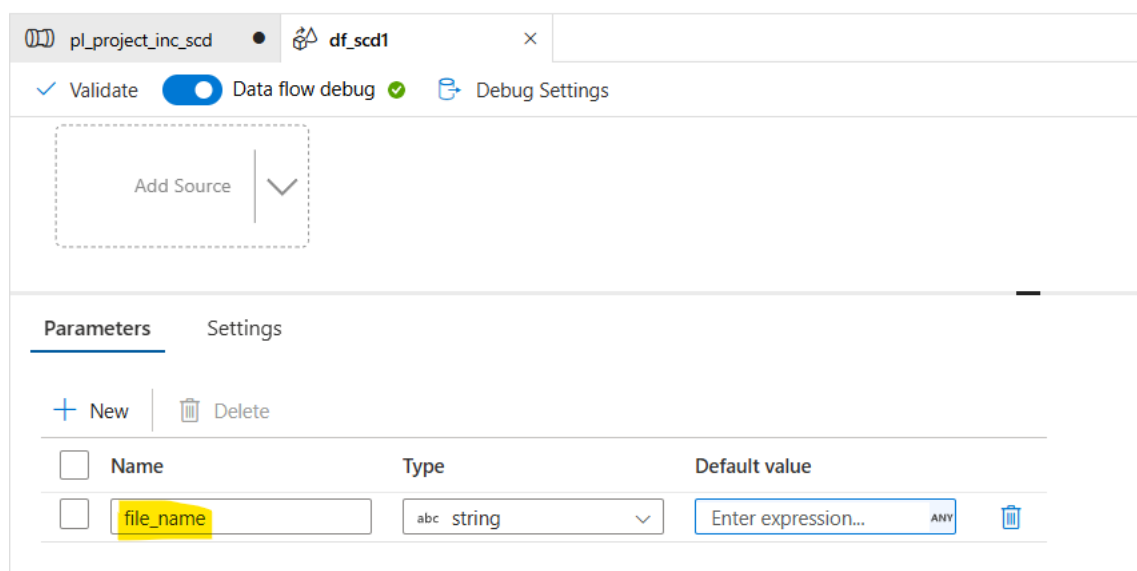
- Drag drag Data Flow Activity and turn on Data flow debug. Connect it to for each activity with success point.



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



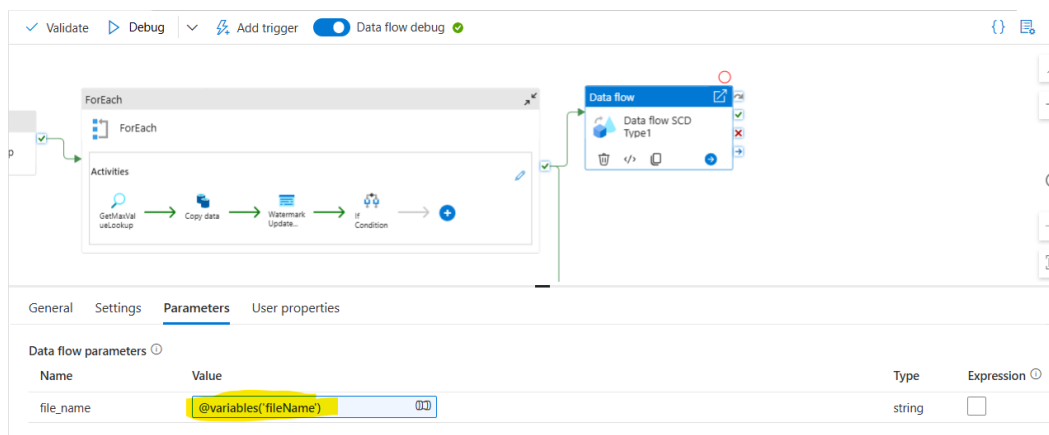
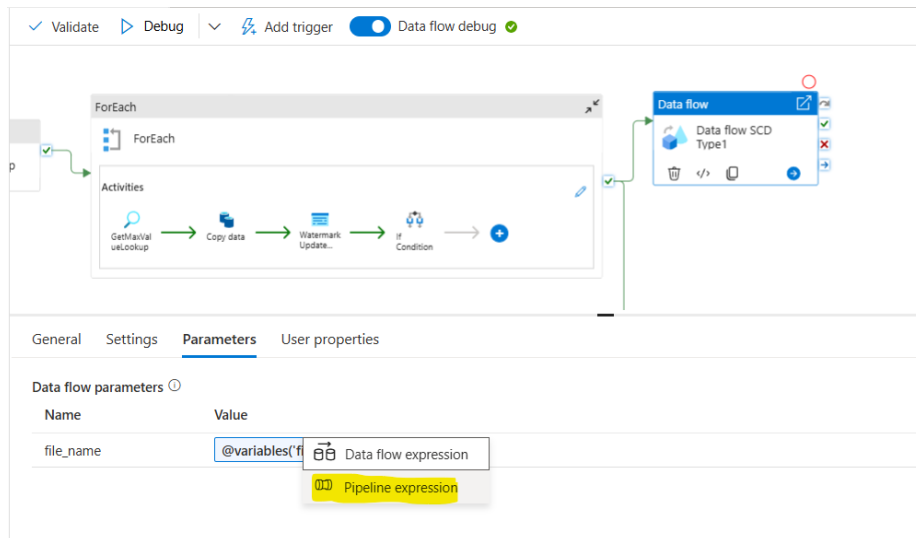
- Open the Created Data flow and create a new parameter named **file_name** inside data flow activity.



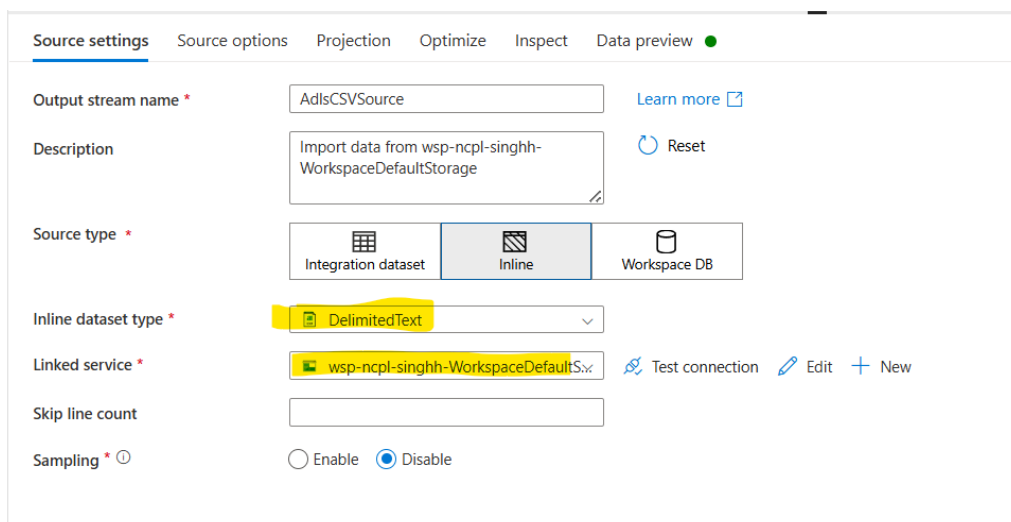
- Then, move to Parameters Tab of Data flow activity in the pipeline, and **create** a new **Parameter** name **file_name**. Use the Dynamic Expression to set the parameter's value with **previously created pipeline variable** named **filename**.

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***

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

File settings

File mode [ⓘ] ☒ File ☐ Wildcard

File path * / / [Browse](#)

Allow no files found [ⓘ] ☐

Change data capture [ⓘ] ☐

Compression type

Encoding

Column delimiter [ⓘ]

Row delimiter [ⓘ]

Quote character

Escape character

First row as header ☒

20

Dataflow expression builder

Default value

+ - * / || && ! ^ == ===

Expression elements

- All
- Functions
- Parameters
- Cached lookup

Expression values

+ Create new

123 abs(123 numeric_value)

123 acos(123 numeric_value)

ANY add(ANY first_expression , ANY second_expression)

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

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

[← Import schema](#) [✕ Clear schema](#) [🔗 Schema options](#)

Column name	↑↓ Type	↑↓ Format
book_id	123 short	Specify format
book_title	abc string	Specify format
book_author	abc string	Specify format
book_genre	abc string	Specify format
published_year	123 short	Specify format

- 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.

Derived column's settings Optimize Inspect Data preview ●

Output stream name * Hashkey [Learn more](#)

Description Creating/updating the columns 'src_book_id, src_book_title, src_book_author, src_book_genre, [Reset](#)

Incoming stream * Rename

Columns * 0

+ Add Clone Delete Open expression builder

Column	Expression
src_Hashkey	crc32(concat(toString(src_book_id), src_book_title, src_book_author, src_book_genre, toString(src_published_year)))

Dataflow expression builder

Hashkey

Derived Columns

+ Create new

src_Hashkey

Column name * src_Hashkey

Expression

```
crc32(concat(toString(src_book_id), src_book_title, src_book_author, src_book_genre, toString(src_published_year)))
```

Expression elements

All

Functions

Input schema

Parameters

Custom functions

Expression values

Filter by keyword

+ Create new

src_book_id

src_book_title

- 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.

pl_project_inc_scd x df_scd1

✓ Validate Data flow debug Debug Settings

Target

Columns: 2 total

Source settings Source options Projection Optimize Inspect Data preview ●

Output stream name * Target [Learn more](#)

Description Import data from Is_SqlDB_connection [Reset](#)

Source type *

Integration dataset **Inline** Workspace DB

Inline dataset type * **Azure SQL Database**

Linked service * **Is_SqlDB_connection** [Test connection](#) [Edit](#) + New

Sampling * 0 ☐ Enable ☒ Disable

pl_project_inc_scd • df_scd1

✓ Validate Data flow debug ✓ Debug Settings

Target
Columns: 2 total

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

Input ☐ Table ☒ Query ☐ Stored procedure

Query * ⓘ
Select bookId, hashkey from
dbo.Books_SCDTYPE1

Incremental column ⓘ ☐

Isolation level ⓘ Read uncommitted

Target
Columns: 2 total

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

← Import schema Clear schema Schema options Overwrite schema

Column name	Type
bookId	123 integer
hashkey	121 long

- 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.

pl_project_inc_scd • df_scd1

✓ Validate Data flow debug ✓ Debug Settings

AdlsCSVSource Rename Hashkey lookup
Import data from wsp-ncpl-singhh-Works paceDefaultStorage Renaming AdlsCSVSource to Rename with columns Creating/updating the columns 'src_book_id, src_book_title, ... Columns: 8 total

Lookup settings **Optimize** Inspect Data preview ●

Output stream name * lookup [Learn more](#)

Description
Lookup on 'Hashkey' from 'Target' [Reset](#)

Primary stream * Hashkey

Lookup stream * Target

Match multiple rows ☐ ⓘ

Match on * Any row

Lookup conditions *
Left: Hashkey's column Right: Target's column
125 src_book_id == 123 bookId

- 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 existing hashkey. So, means need to update record.
 - Expression used:
 Insert: ***isNull(bookId)***
 Update: ***src_book_id == bookId && hashkey != src_Hashkey***

Conditional split settings

Output stream name: [Learn more](#)

Description: Conditionally distributing the data in bookId, src_book_id, bookId, hashkey, src_Hashkey groups, based on columns [Reset](#)

Incoming stream:

Split on: ☐ First matching condition ☒ All matching conditions

Stream names	Condition
Insert	<code>isNull(bookId)</code>
Update	<code>src_book_id == bookId && hashkey != src_Hashkey</code>

- 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()***

✓ Validate ☒ Data flow debug ☐ Debug Settings

Source Rename Hashkey References: 1 Insert InsertAuditColumn...

Derived column's settings Optimize Inspect Data preview ●

Output stream name * InsertAuditColumns [Learn more](#)

Description Creating/updating the columns 'src_book_id, src_book_title, src_book_author, src_book_genre, ... [Reset](#)

Incoming stream * split@Insert

+ Add Clone Delete Open expression builder

Column	Expression
<input type="checkbox"/> src_createdby	'DataFlow' abc +
<input type="checkbox"/> src_createddate	currentTimestamp() +
<input type="checkbox"/> src_updatedby	'DataFlow' abc +
<input type="checkbox"/> src_updateddate	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.

pl_project_inc_scd df_scd1

✓ Validate ☐ Data flow debug ☒

lookup Insert InsertAuditColumn... SinkInsert

Lookup on 'Hashkey' from 'Target' Conditionally distributing the data in bookId, src_book_id, ... Creating/updating the columns 'src_book_id, src_book_title, ... Columns: 10 total

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

✓ Validate ☐ Data flow debug ●

g the
k_id.

lookup
Lookup on 'Hashkey'
from 'Target'

Insert
Conditionally
distributing the data in
bookId, src_book_id,
...

InsertAuditColum...
Creating/updating the
columns 'src_book_id,
src_book_title,

SinkInsert
Columns:
10 total

Sink Settings Errors Mapping Optimize Inspect Data preview

Schema name * Refresh

Table name *

Table action
☒ None ☐ Recreate table ☐ Truncate table

Update method ^①
☒ Allow insert
☐ Allow delete
☐ Allow upsert
☐ Allow update

Use tempdb ^① ☒

Pre SQL scripts ^①
☒ List of scripts ☐ Custom expression ^①
 +

Sink Settings Errors Mapping Optimize Inspect Data preview

Options
☒ Skip duplicate input columns ^①
☒ Skip duplicate output columns ^①

☒ Auto mapping ^① + Add mapping Delete View schema

<input type="checkbox"/> Input columns		Output columns		
<input type="checkbox"/> 12s src_book_id	→	123 bookId		+
<input type="checkbox"/> abc src_book_title	→	abc bookTitle		+
<input type="checkbox"/> abc src_book_author	→	abc bookAuthor		+
<input type="checkbox"/> abc src_book_genre	→	abc bookGenre		+
<input type="checkbox"/> 12s src_published_year	→	123 publishedYear		+
<input type="checkbox"/> abc src_createdby	→	abc createdBy		+
<input type="checkbox"/> src_createddate	→	createdDate		+
<input type="checkbox"/> abc src_updatedby	→	abc updatedBy		+
<input type="checkbox"/> src_updateddate	→	updatedDate		+
<input type="checkbox"/> 12l src_Hashkey	→	12l hashkey		+

– 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()**

Derived column's settings

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Columns *

Column	Expression
<input type="checkbox"/> src_updatedby	<input type="text" value="'Dataflow-Updated'"/> <input type="button" value="abc"/>
<input type="checkbox"/> src_updateddate	<input type="text" value="currentTimestamp()"/> <input type="button" value="🕒"/>

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

Alter row settings

Output stream name * [Learn more](#)

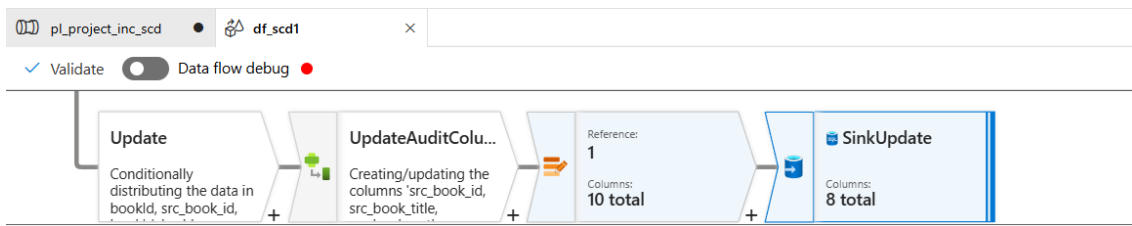
Description [Reset](#)

Incoming stream *

Alter row conditions *

- 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.



Sink Settings

Output stream name * SinkUpdate [Learn more](#)

Description Add sink dataset [Reset](#)

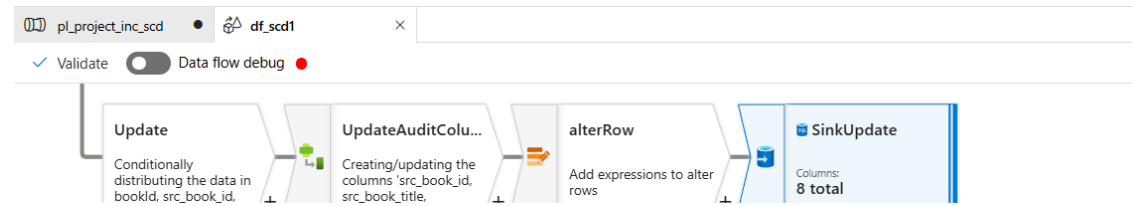
Incoming stream * alterRow

Sink type * Integration dataset **Inline** Workspace DB Cache

Inline dataset type * Azure SQL Database

Linked service * ls_SqlDB_connection [Test connection](#) [Edit](#) [New](#)

Options ☒ Allow schema drift [?](#) ☐ Validate schema [?](#)



Sink Settings

Schema name * dbo [Refresh](#)

Table name * Books_SCDTYPE1

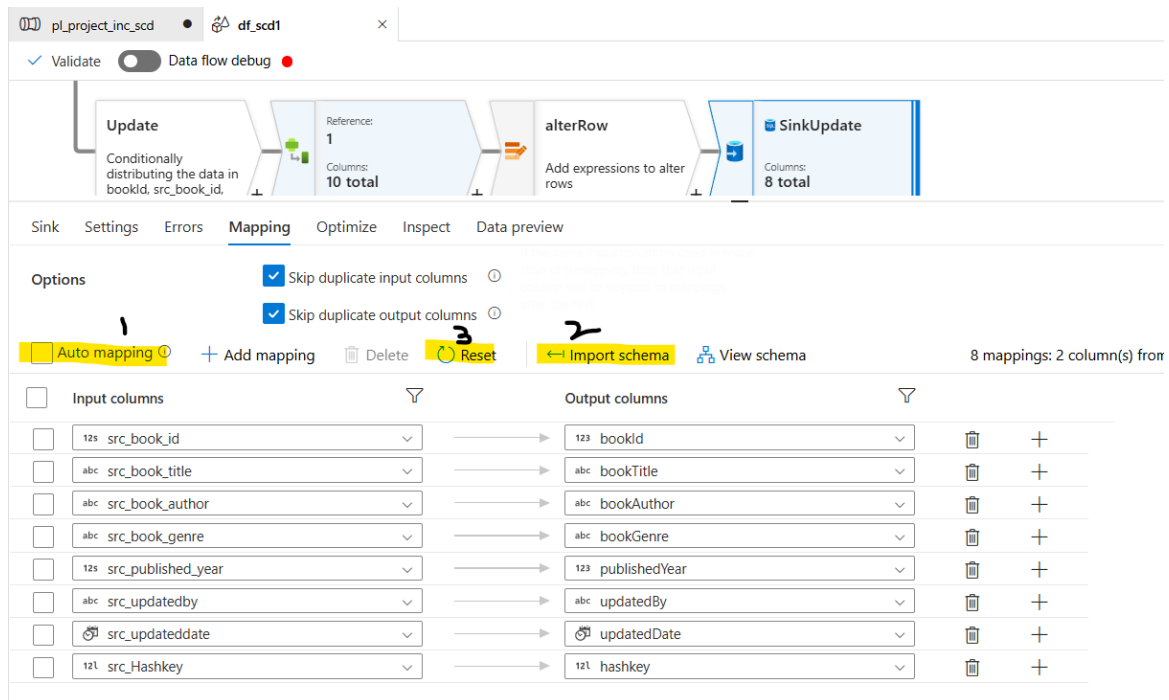
Table action ☒ None ☐ Recreate table ☐ Truncate table

Update method [?](#) ☐ Allow insert ☐ Allow delete ☐ Allow upsert ☒ Allow update

Skip writing key columns [?](#) ☐

Key columns * [?](#) ☒ List of columns ☐ Custom expression [?](#)

123 bookId [+](#) [-](#)



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

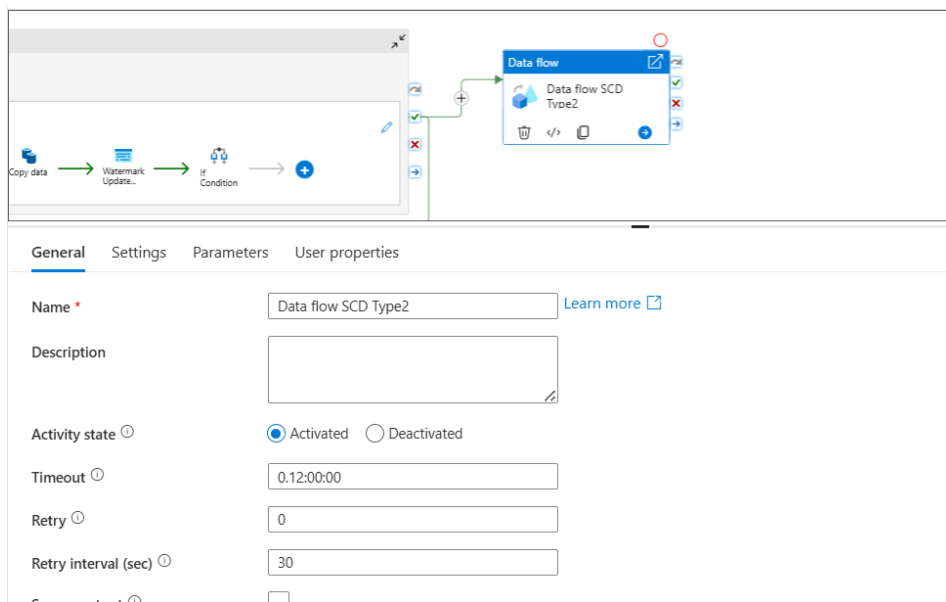
```
(
    memberId Int,
    memberName Varchar(100),
    memberEmail Varchar(255),
    memberPhone Varchar(20) ,
    createdBy varchar(100),createdDate datetime,
    updatedBy varchar(100),updatedAt datetime,hashkey Bigint,
    isActive Int
)
select * from Members_SCDTYPE2
```

```

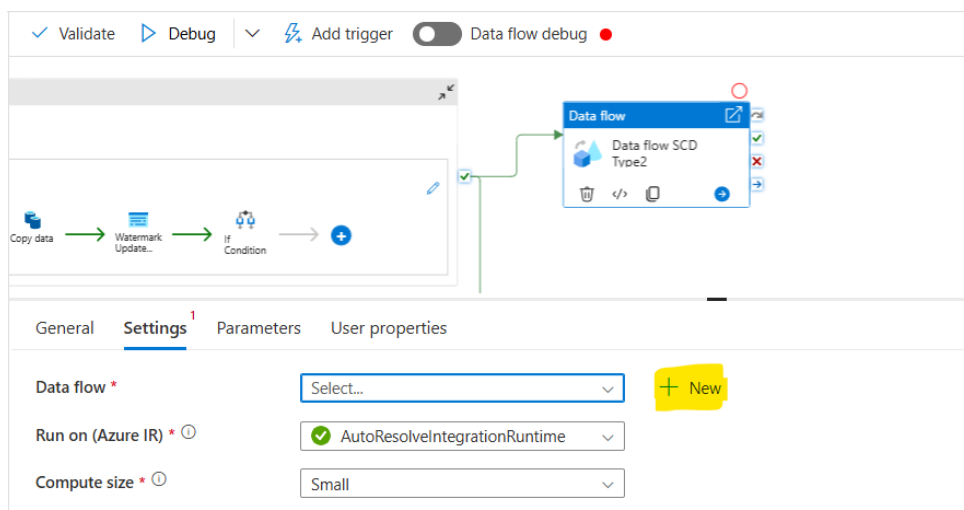
Create Table Members_SCDTYPE2
(
    memberId Int,
    memberName Varchar(100),
    memberEmail Varchar(255),
    memberPhone Varchar(20) ,
    createdBy varchar(100),createdDate datetime,
    updatedBy varchar(100),updatedDate datetime,hashkey Bigint,
    isActive Int
)
select * from Members_SCDTYPE2

```

- 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.

✓ Validate ☒ Data flow debug ☒ Debug Settings

Add Source ▼

Parameters Settings

+ New | Delete

<input type="checkbox"/>	Name	Type	Default value
<input type="checkbox"/>	file_name	abc string	Enter expression... ANY

- 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')

GetMaxVal
velLookup → Copy data → Watermark
Update... → If
Condition → +

General Settings **Parameters** User properties

Data flow parameters ⓘ

Name	Value
file_name	@variables('f' Data flow expression Pipeline expression

GetMaxVal
velLookup → Copy data → Watermark
Update... → If
Condition → +

General Settings **Parameters** User properties

Data flow parameters ⓘ

Name	Value	Type	Expression ⓘ
file_name	@variables('fileName')	string	<input type="checkbox"/>

- 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.

Source settings | Source options | Projection | Optimize | Inspect | Data preview ●

Output stream name * [Learn more](#)

Description [Reset](#)

Source type * Integration dataset **Inline** Workspace DB

Inline dataset type * **DelimitedText**

Linked service * **wsp-ncpl-singhh-WorkspaceDefaultStorage** [Test connection](#) [Edit](#) [New](#)

Skip line count

Sampling * ⓘ ☐ Enable ☒ Disable

- 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***

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

File settings

File mode ⓘ ☒ File ☐ Wildcard

File path * / / [Browse](#)

Allow no files found ⓘ ☐

Change data capture ⓘ ☐

Compression type

Encoding

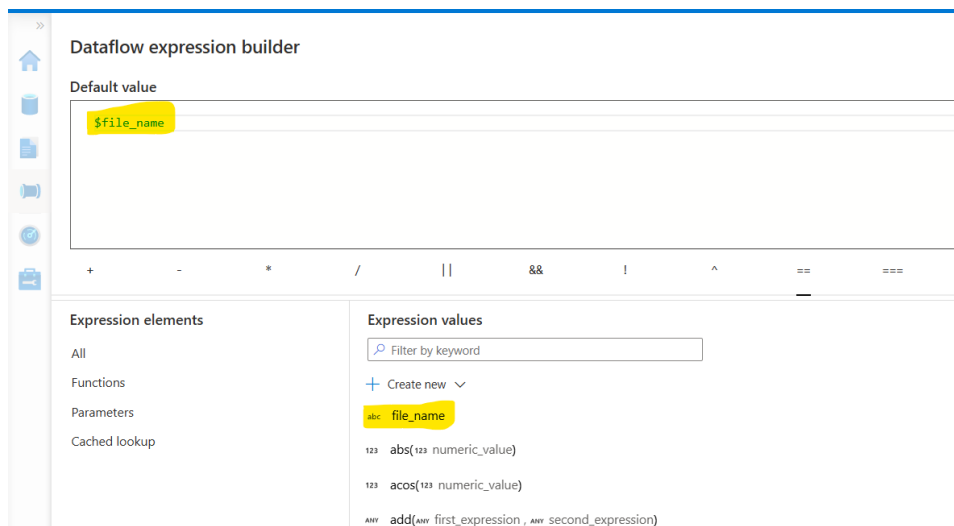
Column delimiter ⓘ

Row delimiter ⓘ

Quote character

Escape character

First row as header ☒

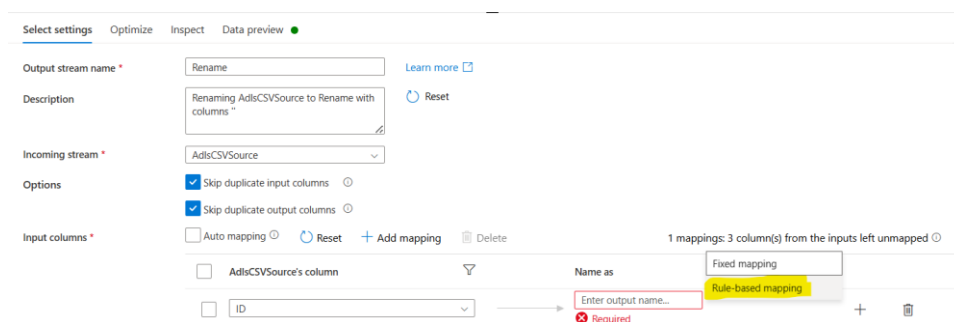


- Click on Projection tab of Source -> click on import schema -> select and review the correct type for columns.

✓ Validate

Data flow debug

- 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.



Dataflow expression builder [Expression reference documentation](#)

Matching condition ☐ Use regular expression [Save](#)

`1==1`

Output column name expression

`concat('src_', $$)`

Expression elements

- All
- Functions
- Input schema
- Parameters
- Cached lookup
- Data flow library functions

Expression values

Filter by keyword

+ Create new

- abc \$\$
- abc \$0
- 123 \$#
- 123 abs(123 numeric_value)
- 123 acos(123 numeric_value)

[Save and finish](#) [Cancel](#) [Clear contents](#)

Incoming stream * AdlsCSVSource

Options

- ☒ Skip duplicate input columns
- ☒ Skip duplicate output columns [Add dynamic content \[Alt+Shift+D\]](#)
- ☐ Auto mapping [Reset](#) [Add mapping](#) [Delete](#)

Input columns * 1 mappings: All inputs mapped

	AdlsCSVSource's column	Name as
<input type="checkbox"/>	1==1	concat('src_', \$\$)

- 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.

Derived column's settings [Optimize](#) [Inspect](#) [Data preview](#)

Output stream name * GenerateHashKey [Learn more](#)

Description Creating/updating the columns 'src_member_id, src_member_name, src_member_email, src_member_phone, [Reset](#)

Incoming stream * Rename

[+ Add](#) [Clone](#) [Delete](#) [Open expression builder](#)

Columns *

Column	Expression
<input type="checkbox"/> src_HashKey	crc32(concat(toString(src_member_id), src_member_name, src_member_email, toString(src_member_phone)))

- 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 updated records.

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

- Click on Projection -> import schema.

The screenshot displays the Azure Data Studio interface, specifically the 'Source settings' tab for a data source named 'Target'. The 'Source settings' tab is active, showing the 'Output stream name' as 'Target' and the 'Description' as 'Import data from Is_SqlDB_connection'. The 'Source type' is set to 'Integration dataset', and the 'Inline dataset type' is 'Azure SQL Database'. The 'Linked service' is 'Is_SqlDB_connection'. The 'Sampling' option is set to 'Disable'.

The 'Source options' tab is also visible, showing the 'Input' type as 'Query'. The 'Query' text area contains the SQL statement: `Select memberId, hashKey from dbo.Members_SCDTYPE2 where isActive=1`. The 'Incremental column' is not selected, and the 'Isolation level' is set to 'Read uncommitted'.

The 'Projection' tab is active, showing the 'Import schema' button. Below the button, the resulting schema is displayed in a table:

Column name	Type
memberId	123 integer
hashKey	121 long

- 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.

Lookup settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Primary stream *

Lookup stream *

Match multiple rows ☐ ☐

Match on *

Lookup conditions *

Left: GenerateHashKey's column		Right: Target's column
123 src_member_id	==	123 memberId

- 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 existing hashkey. So, means need to update record.
 - Expression used:
 Insert: ***isNull(memberId)***
 Update: ***src_member_id==memberId && src_HashKey!=hashKey***

Conditional split settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Split on ☒ First matching condition ☐ All matching conditions

Split condition

Stream names	Condition
Insert	<i>isNull(memberId)</i>
Update	<i>src_member_id==memberId && src_HashKey!=hashKey</i>

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

GenerateHashKey
Creating/updating the columns: src_member_id

lookupTarget
Lookup on 'GenerateHashKey' from 'Target'

Insert
Conditionally distributing the data in memberid, ...

union
Columns: 7 total

Union settings | Optimize | Inspect | Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Union by * ☒ Name ☐ Position

Union with * [+](#) [-](#)

- 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()**

Derived column's settings | Optimize | Inspect | Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

[+ Add](#) [Clone](#) [Delete](#) [Open expression builder](#)

Column	Expression
<input type="checkbox"/> src_createdby	<input type="text" value="'DataFlow'"/>
<input type="checkbox"/> src_createddate	<input type="text" value="currentTimestamp()"/>
<input type="checkbox"/> src_updatedby	<input type="text" value="'DataFlow'"/>
<input type="checkbox"/> src_updateddate	<input type="text" value="currentTimestamp()"/>
<input type="checkbox"/> src_isActive	<input type="text" value="1"/>

- **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 *

ts_SqlDB_connection

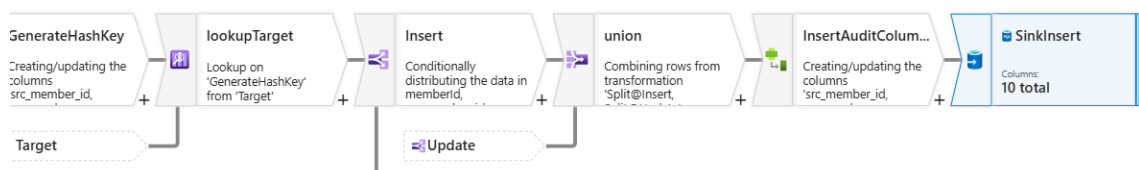
Test connection

Edit

New

Options

☒ Allow schema drift
 ☐ Validate schema



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
 10 n

Input columns	Output columns		
12s src_member_id	123 memberid		+
abc src_member_name	abc memberName		+
abc src_member_email	abc memberEmail		+
12l src_member_phone	abc memberPhone		+
abc src_createdby	abc createdBy		+
src_createddate	createdDate		+
abc src_updatedby	abc updatedBy		+
src_updateddate	updatedDate		+
12l src_HashKey	12l hashkey		+
123 src_isActive	123 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()**

Derived column's settings

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Columns * ☐

Column	Expression
<input type="checkbox"/> src_updatedby	'DataFlow-Updated'
<input type="checkbox"/> src_updateddate	currentTimestamp()
<input type="checkbox"/> src_isActive	0

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

Alter row settings

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Alter row conditions *

- 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.

The image shows two screenshots from the Databricks UI. The top screenshot displays the 'Sink' configuration page for a data flow named 'df_scd2'. The 'Output stream name' is 'SinkUpdate'. The 'Incoming stream' is 'alterRow'. The 'Sink type' is 'Inline'. The 'Inline dataset type' is 'Azure SQL Database'. The 'Linked service' is 'Is_SqlDB_connection'. The 'Options' section has 'Allow schema drift' checked and 'Validate schema' unchecked.

The bottom screenshot shows the 'Mapping' tab for the same data flow. The 'Schema name' is 'dbo' and the 'Table name' is 'Members_SCDTYPE2'. The 'Table action' is 'None'. The 'Update method' has 'Allow update' checked. The 'Key columns' section is highlighted with a red box and contains two entries: 'memberId' and 'hashkey', both with a '123' prefix in the dropdown menu.

pl_project_inc_scd df_scd2

✓ Validate Data flow debug Debug Settings

Creating/updating the columns 'src_member_id', 'Lookup on 'GenerateHashKey' from 'Target', Conditionally distributing the data in memberid, Combining rows from transformation 'Split@Insert', Creating/updating the columns 'src_member_id', Add sink dataset

Target 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 5 mappings: 5 column(s) from the output schema

Input columns	Output columns
123 memberid	123 memberid
abc src_updatedby	abc updatedBy
src_updateddate	updatedDate
121 hashKey	121 hashkey
123 src_isActive	123 isActive

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:

Parameters

Variables

Settings

Output

Pipeline run ID: 82835c48-0109-440a-9e15-c3167bdda56c

Pipeline status ✓ Succeeded

View debug run consumption

All status ▼

List ▼

Showing 1 - 28 of 28 items

Activity name ↑↓	Activity st... ↑↓	Activit... ↑↓	Run start ↑↓	Duration ↑↓	Integration runtime ↑↓	User prop... ↑↓	Activity run ID ↑↓
Data flow SCD Type1	✓ Succeeded	Data flow	3/14/2025, 7:36:34 AM	2m 59s	AutoResolveIntegrationRuntime (Australia East)		f4d99c55-e8c9-44cf-8eef-d5949f
Data flow SCD Type2	✓ Succeeded	Data flow	3/14/2025, 7:36:34 AM	1m 54s	AutoResolveIntegrationRuntime (Australia East)		87d7971e-015d-4030-bb95-16df
If Condition	✓ Succeeded	If Condition	3/14/2025, 7:36:13 AM	Less than 1s			36a1279b-c1f5-4448-a833-057ac
Copy data to Azure SQL DB	✓ Succeeded	Copy data	3/14/2025, 7:36:11 AM	17s	AutoResolveIntegrationRuntime (Australia East)		864ac2cf-d0b4-4817-b6e0-249cc
If Condition	✓ Succeeded	If Condition	3/14/2025, 7:36:11 AM	19s			4b9e3a43-3f1b-40a2-8652-2f56e
If Condition	✓ Succeeded	If Condition	3/14/2025, 7:36:09 AM	Less than 1s			df18ad24-4a27-47c0-a338-a8a9e
Copy data to ... → ↺ ↻	✓ Succeeded	Copy data	3/14/2025, 7:36:09 AM	16s	AutoResolveIntegrationRuntime (Australia East)		81696161-9736-43e8-b660-b024
If Condition	✓ Succeeded	If Condition	3/14/2025, 7:36:08 AM	18s			686f32a3-52c2-4d7b-9c3c-ea3d1
Watermark Update Stored pr...	✓ Succeeded	Stored procedu	3/14/2025, 7:36:03 AM	9s	SelfHostedIntegrationRuntime		26e605e7-b8e7-4dcd-b6a2-e39e
Copy data to Azure SQL DB	✓ Succeeded	Copy data	3/14/2025, 7:36:03 AM	18s	AutoResolveIntegrationRuntime (Australia East)		f0cedf74-78dc-40ff-ad60-a7180c
If Condition	✓ Succeeded	If Condition	3/14/2025, 7:36:03 AM	20s			c4e08cec-554a-4d20-95aa-ec011
Watermark Update Stored pr...	✓ Succeeded	Stored procedu	3/14/2025, 7:35:54 AM	16s	SelfHostedIntegrationRuntime		b2213d44-a147-4de1-9d4f-f995f
Watermark Update Stored pr...	✓ Succeeded	Stored procedu	3/14/2025, 7:35:53 AM	15s	SelfHostedIntegrationRuntime		15123069-37ff-41ec-8dbd-a6f90
Watermark Update Stored pr...	✓ Succeeded	Stored procedu	3/14/2025, 7:35:44 AM	23s	SelfHostedIntegrationRuntime		3a586e72-81c1-418e-901a-f075f
Watermark Update Stored pr...	✓ Succeeded	Stored procedu	3/14/2025, 7:35:40 AM	22s	SelfHostedIntegrationRuntime		f617164b-f194-4a88-91ac-19330

- 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 access		
← → ↑ 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):

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, 7:35:58 AM

With column header ☒ On

MEMBER_ID	MEMBER_NA...	MEMBER_EM...	MEMBER_PH
1	Rahul Sharma	rahul.sharma@g...	9876543210
2	Priya Patel	priya.patel@yah...	9988776655
3	Vikram Kumar	vikram.kumar@o...	9871234567
4	Ananya Gupta	ananya.gupta@r...	9998887770
NULL	NULL	NULL	NULL

OK

Loaded Data in the Azure SQL DB:

Books:

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	1660844415
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:

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	3277860304	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:

lib_id	lib_name	lib_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')

```

---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:

✓ Validate

▶ Debug

▼ Add trigger

☑ Data flow debug

Parameters

Variables

Settings

Output

Pipeline run ID: 16989321-69e0-4dfe-b2a3-62f89997856b

Pipeline status: ✓ Succeeded

View debug run consumption

All status

Container

Monitor in Azure Metrics

Export to CSV

Showing 1 - 28 of 28 items

Activity name	Activity status	Run start	Duration	Activity run ID
(X) 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-b922-6c54a1fde0d6
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-9f4e1fb357b
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:

pL_project_inc_scd

mycontainer

New SQL script

New notebook

New data flow

New integration dataset

Upload

Download

New folder

mycontainer > onPrem_LibraryDB > Books

Name	Last Modified
2025-03-14T15:48:52.1199898Z.csv	3/14/2025, 11:49:58 AM
2025-03-14T15:56:13.9945348Z.csv	3/14/2025, 11:57:05 AM

2025-03-14T15:56:13.9945348Z.csv

Path https://adlsgen2hsingh.dfs.core.windows.net/mycontainer/onPrem_L...
03-14T15:56:13.9945348Z.csv

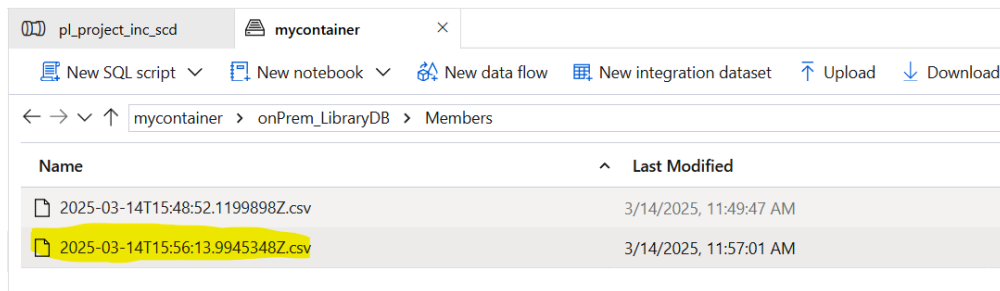
Modified 3/14/2025, 11:57:05 AM

With column header ☒ On

BOOK_ID	BOOK_TITLE	BOOK_AUTH...	BOOK_GE
101	The White Tiger	Aravind Adiga	Fiction2
102	The Guide	R.K. Narayan	Fiction2
105	The Alchemist	Paulo Coelho	Philosophic
106	Life of Pi	Yann Martel	Adventure
NULL	NULL	NULL	NULL

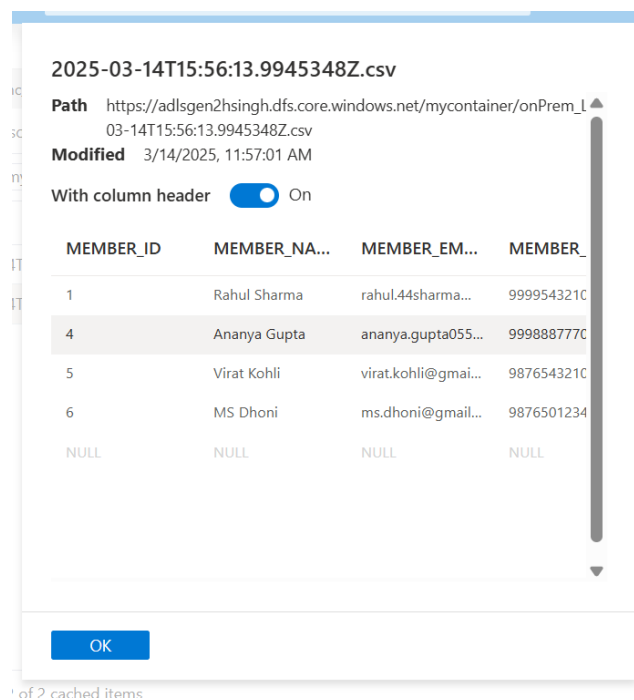
OK

Members new csv file:



The screenshot shows a web interface for a data container named 'mycontainer'. The breadcrumb path is 'mycontainer > onPrem_LibraryDB > Members'. The table has two columns: 'Name' and 'Last Modified'. Two CSV files are listed, with the second one highlighted in yellow.

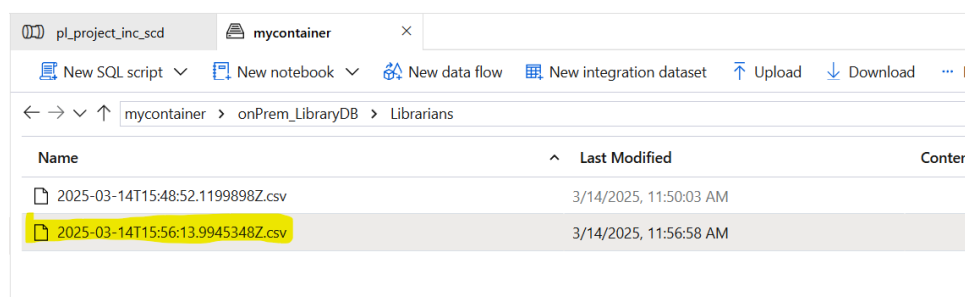
Name	Last Modified
2025-03-14T15:48:52.1199898Z.csv	3/14/2025, 11:49:47 AM
2025-03-14T15:56:13.9945348Z.csv	3/14/2025, 11:57:01 AM



The preview window shows the file path, modification date, and a toggle for 'With column header' which is turned 'On'. The CSV content is displayed in a table with four columns: MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, and MEMBER_PHONE. The data includes five rows of member information and a final row of NULL values.

MEMBER_ID	MEMBER_NAME	MEMBER_EMAIL	MEMBER_PHONE
1	Rahul Sharma	rahul.44sharma...	9999543210
4	Ananya Gupta	ananya.gupta055...	9998887770
5	Virat Kohli	virat.kohli@gmai...	9876543210
6	MS Dhoni	ms.dhoni@gmail...	9876501234
NULL	NULL	NULL	NULL

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



The screenshot shows a web interface for a data container named 'mycontainer'. The breadcrumb path is 'mycontainer > onPrem_LibraryDB > Librarians'. The table has three columns: 'Name', 'Last Modified', and 'Content'. Two CSV files are listed, with the second one highlighted in yellow.

Name	Last Modified	Content
2025-03-14T15:48:52.1199898Z.csv	3/14/2025, 11:50:03 AM	
2025-03-14T15:56:13.9945348Z.csv	3/14/2025, 11:56:58 AM	

2025-03-14T15:56:13.9945348Z.csv

Path

https://adlsgen2hsingh.dfs.core.windows.net/mycontainer/onPrem_L03-14T15:56:13.9945348Z.csv

Modified

3/14/2025, 11:56:58 AM

With column header

On

LIB_ID	LB_NAME	LB_EMAIL	HIRED_AT
5	Ravish Singh Ku...	r.kumar@library....	2021-01-01
NULL	NULL	NULL	NULL

OK

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	9996887770	DataFlow	2025-03-14T15:52:46.2530000	DataFlow-Updated	2025-03-14T16:01:50.8570000	3277860304	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	2481897584	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.sharma@gmail.com	9996543210	DataFlow	2025-03-14T16:02:09.0530000	DataFlow	2025-03-14T16:02:09.0530000	1752219969	1
4	Ananya Gupta	ananya.gupta0555@rediffmail...	9998887770	DataFlow	2025-03-14T16:02:09.0530000	DataFlow	2025-03-14T16:02:09.0530000	3582786583	1

For Librarians:

Results Messages			
🔍 Search to filter items...			
lib_id	lib_name	lib_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