

## Context

CDC(change data capture)	2 – 10
CDC Real-Time	11 – 18
Dynamically Pass Expression	19 – 25
Condition Logic	25 – 29
User Defined Functions	29 – 34
Error Handling	35 – 36
Create Alert	37 – 42
Create Azure Automation	42 – 44
Create Schedule	45 – 48
Flowlet	48 – 57
Slowly Changing Dimension Type 1	58 – 62
Slowly Changing Dimension Type 2	63 – 72

## CDC (change data capture)

1. Here in this example, we are using an SQL Server table. So, create a table in Azure SQL using the below query.

Query: Create table citizens (  
id int Primary Key,  
name varchar(50))

```
insert into citizens values(1, 'Mahesh');  
insert into citizens values(2, 'Ravi');  
insert into citizens values(3, 'Mike');
```

Query 1

Run  Cancel query Save query Export data as Show only Editor

```
1  create table citizens (  
2    id int Primary Key,  
3    name varchar(50))  
4  
5  insert into citizens values(1, 'Mahesh');  
6  insert into citizens values(2, 'Ravi');  
7  insert into citizens values(3, 'Mike');  
8  
9  Select * from citizens
```

Results Messages

Search to filter items...

id	name
1	Mahesh
2	Ravi
3	Mike

2. Create a Target table with the below query.

Query: Create table citizensDW (  
id int,  
name varchar(50))

Query 1

Run  Cancel query Save query Export data as

```
1  create table citizensDW (  
2    id int,  
3    name varchar(50))
```

3. Run the below Query.

Query: EXEC sys.sp\_cdc\_enable\_db  
GO

Query 1 ×

Run Cancel query Save query Export data

```
1 EXEC sys.sp_cdc_enable_db
2 GO
```

4. Also run the below query.

Query: EXEC sys.sp\_cdc\_enable\_table  
@source\_schema='dbo',  
@source\_name='citizens',  
@role\_name=NULL,  
@supports\_net\_changes=1

Query 1 ×

Run Cancel query Save query Export data

```
1 EXEC sys.sp_cdc_enable_table
2 @source_schema='dbo',
3 @source_name='citizens',
4 @role_name=NULL,
5 @supports_net_changes=1
```

Results Messages

Query succeeded: Affected rows: 0

5. Create a Dataset for Source Data.

#### Set properties

Name

Linked service \*  
 

Table name  
   
 Edit

Import schema  
 From connection/store  None

6. Create a Dataset for the target table.

#### Set properties

Name  
CitizenDW

Linked service \*  
AzureSqlDatabase

Table name  
dbo.citizensDW

Edit

Import schema  
 From connection/store  None

7. Create a data flow.

Factory Resources

CitizenDS CitizenDW

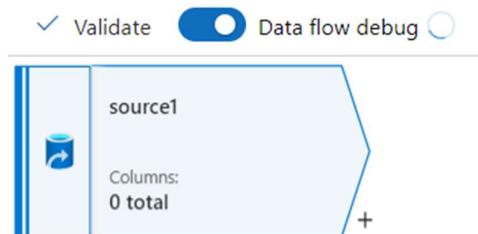
+ Pipeline Change Data Capture (preview) Database

+ Datasets Data flow Data flow

+ Data flows Power Query Flowlet

+ Power Query Copy Data tool

8. Add a Data source.



9. Add a Source dataset.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name \* source1 Learn more

Description Import data from CitizenDS

Source type \* Dataset Inline

Dataset \* CitizenDS Test connection Open New

Options  Allow schema drift

10. Under Source options set the below properties.

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

---

Input       Table    Query    Stored procedure

Batch size

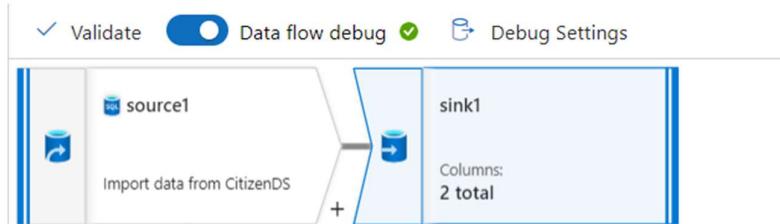
Change data capture

Type       Incremental column    SQL Server CDC

Run mode      Full on the first run, then incremental

Isolation level  Read uncommitted

11. Next add Sink.



12. Select the target dataset.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

---

Output stream name \*  sink1   [Learn more](#)

Description  Export data to CitizenDW   [Reset](#)

Incoming stream \*  source1

Sink type \*   
 Dataset    Inline    Cache

Dataset \*  CitizenDW   [Test connection](#)   [Open](#)   [N](#)

Allow schema drift

13. Check all the methods and select the key column.

Sink    **Settings**    Errors    Mapping    Optimize    Inspect    Data preview

Update method ⓘ

Allow insert

Allow delete

Allow upsert

Allow update

Key columns \* ⓘ

List of columns     Custom expression ⓘ

123 id

Add dynamic content [Alt+Shift+D]

+

trash bin

14. Create a New Pipeline. Drag and drop the Dataflow.

Activities

Validate    Debug    Ad

Data flow

Data flow1

15. Now select the Dataflow.

General    **Settings**    Parameters 1    User properties

Data flow \*

CDC\_dataflow

Run on (Azure IR) \* ⓘ

AutoResolveIntegrationRuntime

Compute size \* ⓘ

Small

Open

16. Validate the pipeline and click on Debug.

17. Here our pipeline executed successfully.

Data flow

Data flow1

Parameters    Variables    Settings    **Output**

Pipeline run ID: 49ac6536-e1a5-4b48-94f4-91b6b6e7c96f

Pipeline status: Succeeded

6 | Page

18. Go and check the target table. Here the source data is loaded into the target table.

```
3   select * from citizensDW;
```

Results    Messages

Search to filter items...	
id	name
1	Mahesh
2	Ravi
3	Mike

19. Using the below query, insert a row in the source table.

Query: `insert into citizens values(4, 'Dean');`

20. No in the source table we have four rows.

```
1   insert into citizens values(4, 'Dean');
2
3   select * from citizens;
4
5   select * from citizensDW;
```

Results    Messages

Search to filter items...	
id	name
1	Mahesh
2	Ravi
3	Mike
4	Dean

21. Go to Data Factory and click on debug again.

22. Here our pipeline executed successfully.

The screenshot shows the 'Data flow' section of the Azure Data Factory pipeline status page. At the top, there's a green checkmark icon indicating success. Below it, the pipeline name 'Data flow1' is shown with its own green checkmark. A horizontal navigation bar below the pipeline list includes tabs for 'Parameters', 'Variables', 'Settings', and 'Output', with 'Output' being the active tab. Further down, the 'Pipeline run ID' is listed as 'eb26a912-9515-48c0-bd40-dfb66d6cbe71' with a copy icon. To the right of the run ID is a circular refresh button and a gear icon. The 'Pipeline status' is displayed as 'Succeeded' with a green checkmark icon.

23. Go to the Target table and check the data. It inserts only the fourth row.

```
4  
5  select * from citizensDW;
```

The screenshot shows the 'Results' tab of the Azure Data Factory output page. It displays a table with two columns: 'id' and 'name'. The data rows are:

id	name
1	Mahesh
2	Ravi
3	Mike
4	Dean

24. Now use the below query and update the source data row value.

Query: update citizens set name = 'Mahi' where id = 1

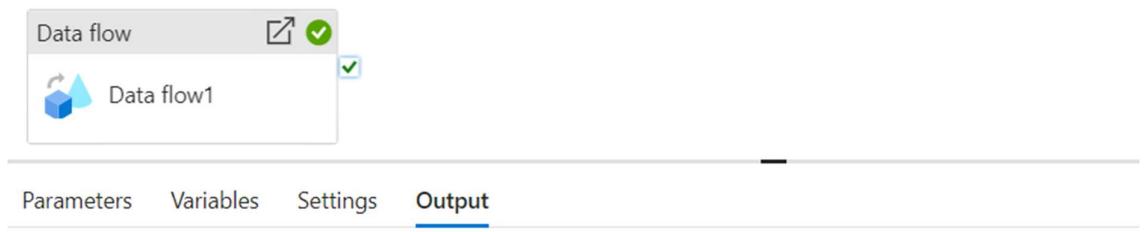
```
▷ Run  □ Cancel query  ⌂ Save query  ⌂ Export data as  └ Show only Editor  
1  insert into citizens values(4, 'Dean');  
2  
3  update citizens set name = 'Mahi' where id = 1  
4  
5  select * from citizens;  
6  
7  select * from citizensDW;
```

The screenshot shows the 'Results' tab of the Azure Data Factory output page. It displays a table with two columns: 'id' and 'name'. The data rows are:

id	name
1	Mahi
2	Ravi
3	Mike
4	Dean

25. Go to Data Factory and click on debug again.

26. Here our pipeline executed successfully.



A screenshot of the Azure Data Factory pipeline status page. At the top, there's a header with 'Data flow' and a green checkmark icon. Below it, a card shows 'Data flow1' with its own green checkmark. A horizontal navigation bar below the card includes tabs for 'Parameters', 'Variables', 'Settings', and 'Output', with 'Output' being the active tab. To the right of the tabs, there's a 'Pipeline run ID' (b5503fa3-2b58-4e71-945c-8d2d3abcd468) and a 'Pipeline status' indicator showing a green checkmark and the word 'Success'.

27. Go to the target table and check the data.

```
6  
7   select * from citizensDW;
```

Results    Messages

Search to filter items...

id	name
1	Mahi
2	Ravi
3	Mike
4	Dean

28. Now use the below query and update the source data row value.

Query: Delete citizens where id = 1

Run    Cancel query    Save query    Export data as    Show only Editor

```
1  insert into citizens values(4, 'Dean');  
2  
3  update citizens set name = 'Mahi' where id = 1  
4  
5  Delete citizens where id = 1  
6  
7  select * from citizens;  
8  
9  select * from citizensDW;
```

Results    Messages

Search to filter items...

id	name
2	Ravi
3	Mike
4	Dean

29. Go to Data Factory and click on debug again.

30. Here our pipeline executed successfully.

The screenshot shows the Azure Data Factory Pipeline run status page. At the top, there's a navigation bar with tabs: Data flow, Parameters, Variables, Settings, and Output. The Output tab is selected. Below the tabs, the pipeline run ID is 57773a35-730a-4b75-96d3-221b6eca7a2c. The Pipeline status is shown as Succeeded with a green checkmark icon. There are also icons for Refresh and View. The main content area displays a query result table. The table has two columns: id and name. The data rows are:

id	name
2	Ravi
3	Mike
4	Dean

# CDC Real-Time

1. In this example we are going to use the below Customer Table in Azure SQL Database.

Query: CREATE TABLE Customer (

```
CustomerID INT PRIMARY KEY,
FirstName VARCHAR(50),
LastName VARCHAR(50),
Email VARCHAR(100)
);
```

Query 1 ×

Run Cancel query Save query Export data as Show only Editor

```
1 CREATE TABLE Customer (
2     CustomerID INT PRIMARY KEY,
3     FirstName VARCHAR(50),
4     LastName VARCHAR(50),
5     Email VARCHAR(100)
6 );
```

Results Messages

Query succeeded: Affected rows: 0

2. Run the below Query.

Query: EXEC sys.sp\_cdc\_enable\_db  
GO

Query 1 ×

Run Cancel query Save query Export data

```
1 EXEC sys.sp_cdc_enable_db
2 GO
```

3. Also run the below query.

Query: EXEC sys.sp\_cdc\_enable\_table  
@source\_schema='dbo',  
@source\_name='Customer',  
@role\_name=NULL,  
@supports\_net\_changes=1

Run Cancel query Save query

```
1 EXEC sys.sp_cdc_enable_table
2 @source_schema='dbo',
3 @source_name='Customer',
4 @role_name=NULL,
5 @supports_net_changes=1
```

Results Messages

Query succeeded: Affected rows: 0

4. Next create a Target table.

```
Query: CREATE TABLE CustomerDW (
    CustomerID INT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100)
);
```

Query 1 X

Run  Cancel query    Show only Edit

```
1  CREATE TABLE CustomerDW (
2      CustomerID INT,
3      FirstName VARCHAR(50),
4      LastName VARCHAR(50),
5      Email VARCHAR(100)
6 );
```

Results Messages

Query succeeded: Affected rows: 0

5. Go to Azure Data Factory, click on the plus symbol, and click on Change Data Capture.

Microsoft Azure | Data Factory ▶ AzureDataFactoryTraing

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and

»

**Factory Resources** +<

- ▷ Pipelines
- ▷ Change Data Capture (preview)
- ▷ Datasets
- ▷ Data flows
- ▷ Power Query
-

6. Give the Name, select the Source type, Linked service, and table.
7. Then click on Continue.

#### Choose Your Sources

Change Data Capture (CDC) automatically detects data changes at the source and then sends the updated data to the destination.

CDC name *	<input type="text" value="CustomerCDC"/>		
Source type *	<input type="button" value="Azure SQL Database"/>		
Source linked service *	<input type="button" value="AzureSqlDatabase1"/> 		
Source tables	<input type="text" value="Filter by table"/>		
Showing 1 - 1 of 1 tables (1 selected)			
	Name	Incremental column	Preview
<input checked="" type="checkbox"/>	dbo.Customer	SQL Server CDC	
			

8. Select the source type, linked service, and Target table then click on continue.

#### Choose Your Targets

Change Data Capture (CDC) automatically detects data changes at the source and then sends the updated data to the destination.

Target type *	<input type="button" value="Azure SQL Database"/>	
Target linked service *	<input type="button" value="AzureSqlDatabase1"/> 	
Target tables	<input type="text" value="Filter by table"/>	
<input type="radio"/> New entities <input checked="" type="radio"/> Existing entities		
Showing 1 - 11 of 11 tables (existing:1, total:1 tables selected)		
	Name	Preview
<input type="checkbox"/>	dbo.Customer	
<input type="checkbox"/>	cdc.cdc_jobs	
<input type="checkbox"/>	cdc.dbo_Customer_CT	
<input checked="" type="checkbox"/>	dbo.CustomerDW	
<input type="checkbox"/>	dbo.TotalSale	
<input type="checkbox"/>	dbo.sytranschemas	
<input type="checkbox"/>	cdc.change_tables	
<input type="button" value="Continue"/> <input type="button" value="Back"/> <input type="button" value="Cancel"/>		

9. Next Uncheck the Auto map and click on Column Mapping.

The screenshot shows the Azure Data Factory Data Flow blade. A mapping is being configured from 'Source Table' `dbo.Customer` to 'Target Table' `dbo.CustomerDW`. The 'Auto map' checkbox is unchecked, and the 'Column mappings' tab is selected. Below the mapping table, a table shows the source linked service and target type: 'AzureSQLDatabase1' (Azure SQL Database) and '1' source table.

10. Set the Key column and give Upper case for both the First and Last Name.

11. Click on dbo.CustomerDW to go back.

The screenshot shows the mapping configuration for the `dbo.CustomerDW` table. It lists four columns: `CustomerID`, `Email`, `FirstName`, and `LastName`. Each column is mapped to its corresponding target column with the 'Upper' mapping method selected. The 'Keys' column indicates that `CustomerID` is the primary key.

12. Next click on the Set Latency option.

The screenshot shows the `CustomerCDC` blade. The 'Set Latency' button is highlighted. A message at the bottom states: 'This preview feature is licensed to you as part of your Azure'.

13. Select the Real-Time and click on Apply.

### Set Latency

Real-time

15 minute

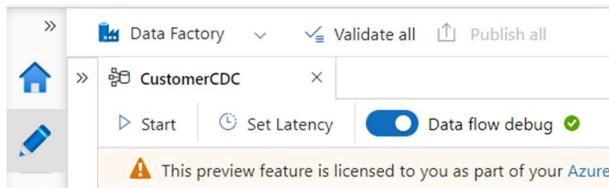
30 minute

1 hour

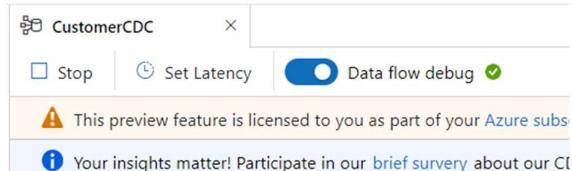
2 hours

14. Next publish our CDC.

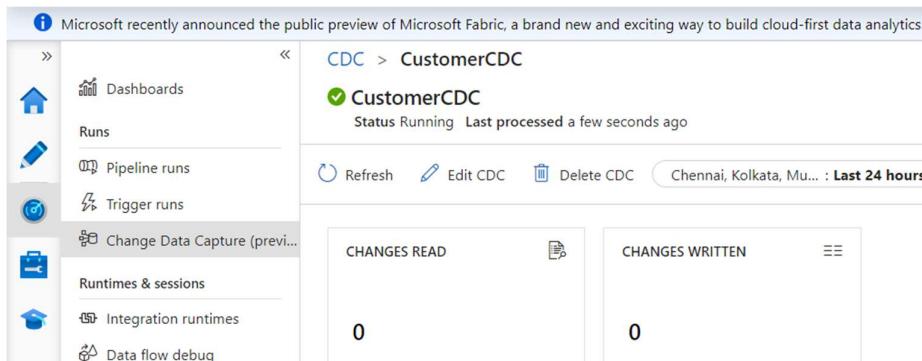
15. Then click on the Start option.



16. After some time you will see a stop option. Which means our CDC is running.



17. Under Monitor, you can see the status of our CDC.



18. Now go to Azure Sql and Insert Data into our Customer table.

Query: `INSERT INTO Customer (CustomerID, FirstName, LastName, Email)  
VALUES`

```
(1, 'John', 'Doe', 'john.doe@example.com'),  
(2, 'Jane', 'Smith', 'jane.smith@example.com'),  
(3, 'Alice', 'Johnson', 'alice.johnson@example.com'),  
(4, 'Bob', 'Williams', 'bob.williams@example.com');
```

Query 1 ×

Run Cancel query Save query Export data as Show only Editor

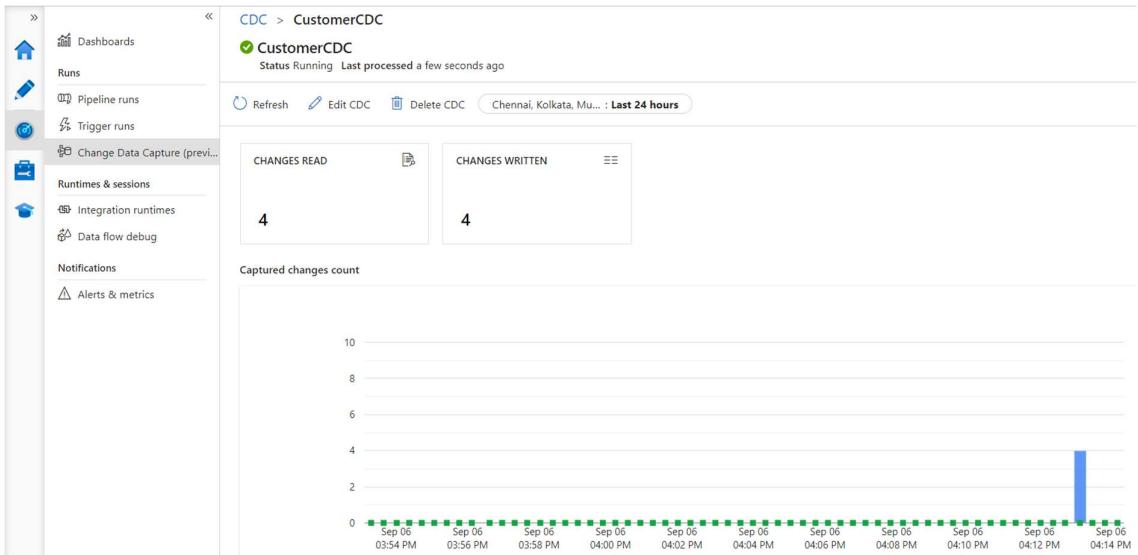
```
1 INSERT INTO Customer (CustomerID, FirstName, LastName, Email)
2 VALUES
3 (1, 'John', 'Doe', 'john.doe@example.com'),
4 (2, 'Jane', 'Smith', 'jane.smith@example.com'),
5 (3, 'Alice', 'Johnson', 'alice.johnson@example.com'),
6 (4, 'Bob', 'Williams', 'bob.williams@example.com');
7
8 Select * from Customer;
```

Results Messages

Search to filter items...

CustomerID	FirstName	LastName	Email
1	John	Doe	john.doe@example.com
2	Jane	Smith	jane.smith@example.com
3	Alice	Johnson	alice.johnson@example.com
4	Bob	Williams	bob.williams@example.com

19. Now go to CDC Monitor and check the changes.



20. Now go back to Azure SQL and check the Target table.

```
8 | select * from CustomerDW;
```

Results    Messages

Search to filter items...

CustomerID	FirstName	LastName	Email
1	JOHN	DOE	john.doe@example.com
2	JANE	SMITH	jane.smith@example.com
3	ALICE	JOHNSON	alice.johnson@example.com
4	BOB	WILLIAMS	bob.williams@example.com

21. Now update a row value in our source table.

Query: UPDATE Customer SET Email = 'jane.smith.updated@example.com'  
WHERE CustomerID = 2;

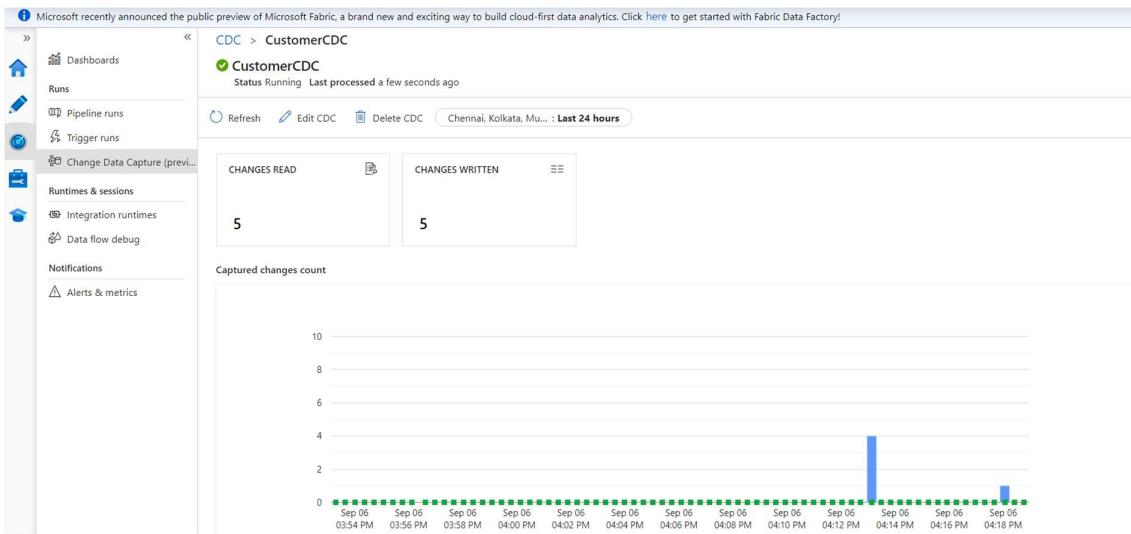
```
▷ Run     Cancel query    ⌂ Save query    ⌂ Export data as     Show only Editor  
1 | UPDATE Customer SET Email = 'jane.smith.updated@example.com'  
2 | WHERE CustomerID = 2;  
3 |  
4 | Select * from Customer;
```

Results    Messages

Search to filter items...

CustomerID	FirstName	LastName	Email
1	John	Doe	john.doe@example.com
2	Jane	Smith	jane.smith.updated@example.com
3	Alice	Johnson	alice.johnson@example.com
4	Bob	Williams	bob.williams@example.com

## 22. Go to CDC Monitor and changes of Reaad and Written.



## 23. Go back to Azure SQL and check the Target data.

```
7  Select * from CustomerDW;
```

Results    Messages

Search to filter items...

CustomerID	FirstName	LastName	Email
1	JOHN	DOE	john.doe@example.com
2	JANE	SMITH	jane.smith.updated@example.com
3	ALICE	JOHNSON	alice.johnson@example.com
4	BOB	WILLIAMS	bob.williams@example.com

## 24. Next delete a Row in our source table.

Query: DELETE FROM Customer WHERE CustomerID = 4;

Run    Cancel query    Save query    Export data as    Show only Editor

```
1  DELETE FROM Customer WHERE CustomerID = 4;
2
3  Select * from Customer;
```

Results    Messages

Search to filter items...

CustomerID	FirstName	LastName	Email
1	John	Doe	john.doe@example.com
2	Jane	Smith	jane.smith.updated@example.com
3	Alice	Johnson	alice.johnson@example.com

## 25. Go to CDC Monitor and changes of Reaad and Written.

The screenshot shows the Microsoft Azure Data Factory interface. In the top navigation bar, it says "Microsoft Azure | Data Factory > AzureDataFactoryTraining". The main content area is titled "CDC > CustomerCDC" with a status of "Running" and "Last processed a few seconds ago". Below this, there are two boxes: "CHANGES READ" and "CHANGES WRITTEN", both showing a count of "6". A chart titled "Captured changes count" displays a single blue bar at the value of 6, spanning from approximately 04:13 PM to 04:14 PM on September 6th. The left sidebar contains links for Dashboards, Runs, Pipeline runs, Trigger runs, Change Data Capture (previews), Runtimes & sessions, Integration runtimes, Data flow debug, Notifications, and Alerts & metrics.

## 26. Go back to Azure SQL and check the Target data.

```
7   Select * from CustomerDW;
```

---

Results    Messages

Search to filter items...

CustomerID	FirstName	LastName	Email
1	JOHN	DOE	john.doe@example.com
2	JANE	SMITH	jane.smith.updated@example.com
3	ALICE	JOHNSON	alice.johnson@example.com

---

## Dynamically Pass Expression

1. In this example we are going to use the below CSV file data.

```
Std_details
File Edit View
SID,NAME,SUBJECT,MARKS
1,Mike,Maths,80
1,Mike,Science,75
1,Mike,English,76
1,Mike,Hindi,68
2,Dean,Maths,78
2,Dean,Science,45
2,Dean,English,65
2,Dean,Hindi,69
```

2. We also need to prepare the CSV file with the file and filter expression as shown below.

```
rulesFile
Std_details.txt,byName('NAME') == 'Mike'
```

3. I have uploaded both files to my source folder.



4. Go to Azure Data Factory and create a Dataset for the rulesFile.
5. Select Azure Blob Storage and click on continue. Then select Delimited and click on Continue.
6. Next give the name, Linked Services, and select the file as shown below.

Name  
RuleDataset

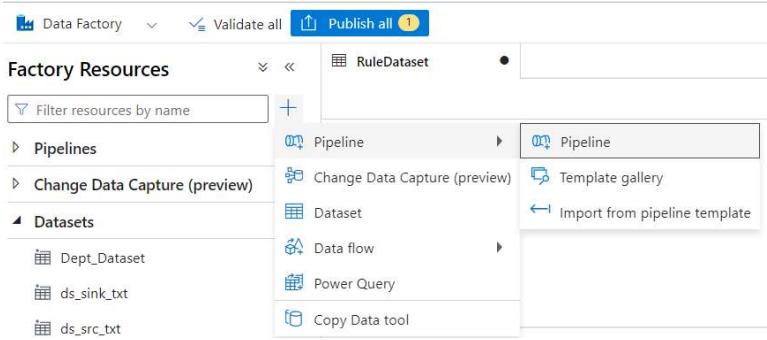
Linked service \*  
AzureBlobStorage\_txt

File path  
f-demo / Directory / rulesFile.csv

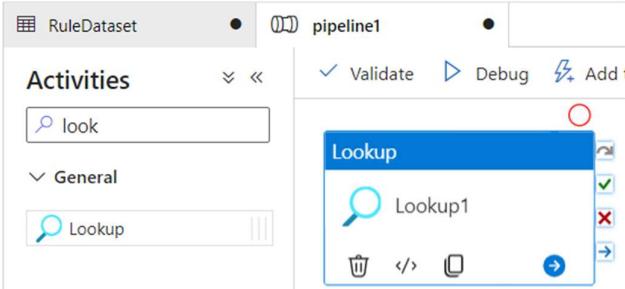
First row as header

Import schema  
 From connection/store  From sample file  None

7. Create a Pipeline.



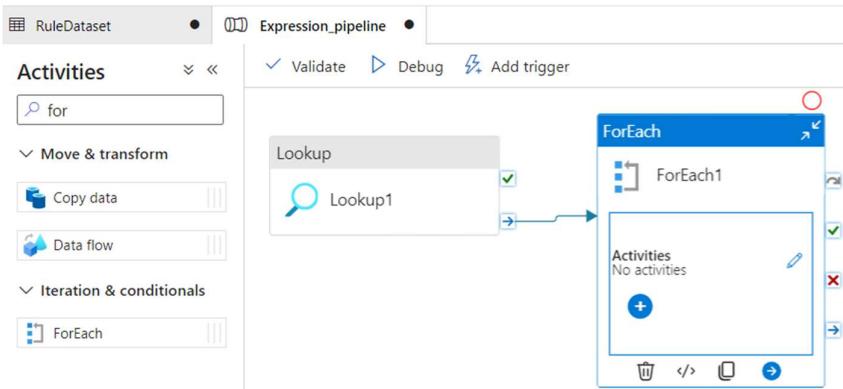
8. Drag and drop the lookup activity as shown below.



9. Under the setting select the Rule data set that we created before. And uncheck the First row-only option.



10. Drag and drop the Foreach activity and make the connection as shown below.



11. Under setting check the sequential option and give the below expression.

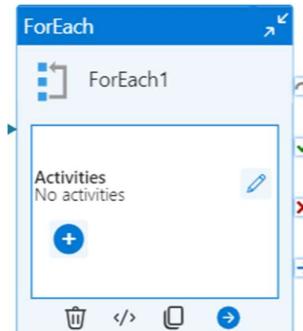
Exp: @activity('Lookup1').output.value

General    **Settings**    Activities (0)    User properties

Sequential   

Items    `@activity('Lookup1').output.value`

12. Now click on the pencil symbol in the foreach loop.



13. Now drag and drop the Data flow activity as shown below.

RuleDataset    Expression\_pipeline

Activities

da

Move & transform

Copy data

Data flow

Azure Data Explorer

Expression\_pipeline > ForEach1

Data flow

Data flow1

14. Under the setting click on the plus symbol to create.

General    **Settings**    Parameters    User properties

Data flow \*

Select...

+ New

Run on (Azure IR) \*

AutoResolveIntegrationRuntime

15. Add a Data source.

Validate     Data flow debug    Debug Settings

source1

Columns:  
0 total

+

16. Under the Source Setting, in the dataset click on the plus symbol.

Source settings    Source options    Projection    Optimize    Inspect    Data preview

Output stream name \*  Learn more

Description  Reset

Source type \* Dataset Inline

Dataset \*  New

17. Select Azure Blob Storage and click on continue. Then select Delimited Text and click on Continue.

18. Give the name, linked service, and container name as shown below.

Name

Linked service \*

File path  /  /

First row as header

Import schema  From connection/store  From sample file  None  
 Advanced

19. Click on Open.

Source type \* Dataset Inline

Dataset \*  Test connection Open New

20. Under parameter create a parameter as shown below.

Connection   Schema   **Parameters**

New Delete

<input type="checkbox"/> Name	Type	Default value
<input type="checkbox"/> FileName	String	<input type="text" value="Value"/>

21. Under the connection in the filename give the below expression.

Exp: @dataset().FileName

Connection   Schema   Parameters

Linked service \*

File path \*  /  /

Compression type

22. Now go to Foreach loop and under settings in the FileName give the below expression.

Exp: @item().FileName

The screenshot shows the 'Settings' tab for a Foreach loop named 'expression\_dataflow'. Under 'Data flow \*', it is set to 'expression\_dataflow'. A parameter 'source1 parameters' is defined with a single entry: 'FileName' with a value of '@item().FileName'. The 'Run on (Azure IR) \*' dropdown is set to 'AutoResolveIntegrationRuntime'.

23. Go to Data flow and create the below parameter.

The screenshot shows the 'Parameters' tab for a Data flow. It lists a single parameter 'FilterRule' of type 'string' with a default value of 'abc'. An 'Enter expression...' placeholder is present.

24. Next again go to the Foreach loop, under the parameter for the FilterRule give the below pipeline expression.

Exp: @item().rule

The screenshot shows the 'Parameters' tab for a Foreach loop. It lists a parameter 'FilterRule' with a value of '@item().rule'.

25. Now go to dataflow click on the plus symbol and click on the filter as shown below.

The screenshot shows the Data Flow designer. A 'source1' component is connected to a 'Row modifier' step. The 'Row modifier' step has a 'Filter' option selected. The 'Add Source' button is visible at the bottom left of the interface.

26. Under filter give the below expression as shown below.

Exp: `toBoolean(expr($FilterRule))`

Filter settings    Optimize    Inspect    Data preview ●

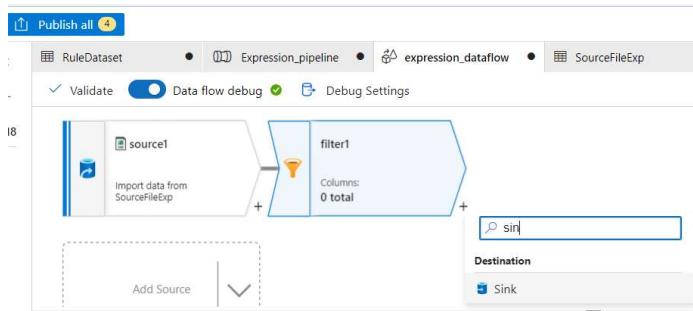
Output stream name \*  Learn more ↗

Description  ⌂ Reset

Incoming stream \*  ▾

Filter on \*  ✎

27. Click on the plus symbol and click on the sink.



28. Under the sink select the destination dataset.

Sink    Settings    Errors    Mapping    Optimize    Inspect    Data preview ●

Description  ⌂ Reset

Incoming stream \*  ▾

Sink type \*  Dataset  Inline  Cache

Dataset \*  ▾ Test connection Open New

29. Under the settings select the file option and give the filename as shown below.

Sink    **Settings**    Errors    Mapping    Optimize    Inspect    Data preview ●

ⓘ This sink currently has Single partition set in Optimize. This will make your data flow execution longer. Th

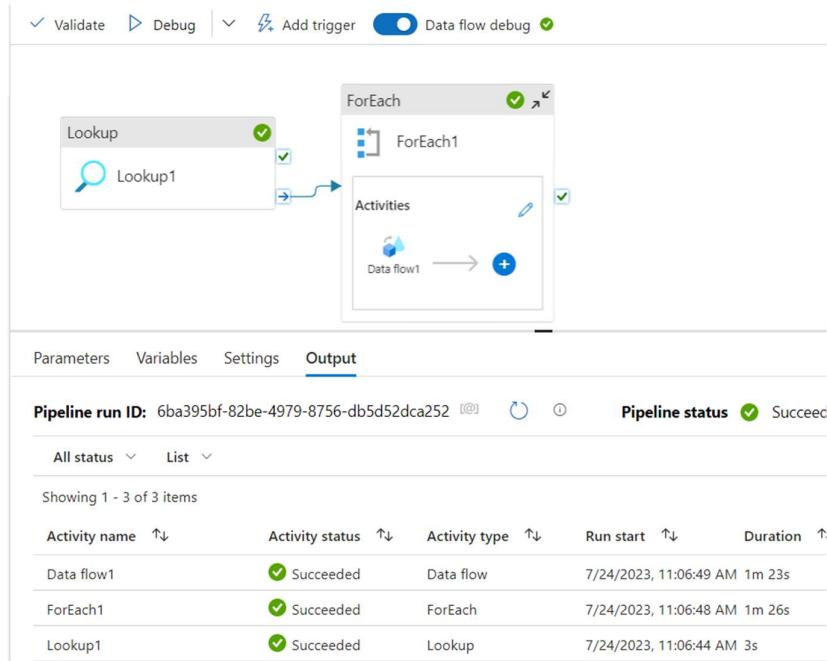
Clear the folder

File name option \*  ▾

Output to single file \* ⓘ

30. Now go to Pipeline, validate it, and click on debug.

31. Here our pipeline runs successfully.



32. Go to the destination and check the output.

Name	Modified	Access tier
[...]		
_SUCCESS	7/21/2023, 5:21:41 PM	Hot (Inferred)
Aggregate_Table.csv	7/21/2023, 6:02:16 PM	Hot (Inferred)
AllEmployees.csv	7/21/2023, 12:55:16 ...	Hot (Inferred)
Asia_Sales.csv	7/21/2023, 5:26:07 PM	Hot (Inferred)
ExpressionFile.csv	7/24/2023, 11:07:23 ...	Hot (Inferred)
UDEmployees.csv	7/21/2023, 11:50:15	Hot (Inferred)

## Condition Logic

1. Go to Azure Data Factory and create a pipeline.

- Factory Resources
- Filter resources by name
- Change Data Capture (preview)
- Datasets
- Data flows
- Power Query
- Pipeline
- Change Data Capture (preview)
- Dataset
- Data flow
- Power Query
- Copy Data tool

2. Go to the “Parameters” tab of the Pipeline. Click on the “+ New” link to add the following two Parameters -
3. “listOfCities” of “Array” Type, and, provide [“New York”, “London”, “Tokyo”, “Singapore City”, “Lisbon”] as the Default value, which is a comma-separated value.
4. “cityToSearch” of “String” Type, and, provide “Kolkata” as the Default value.

**Parameters <sup>1</sup>** Variables Settings Output ^

+ New | Delete

Name	Type	Default value
listOfCities	Array	["New York", "London", "Tokyo", "Singapore City", "Lisbon"]
cityToSearch	String	Kolkata

5. Now, go to the “Variables” tab of the Pipeline. Click on the “+ New” link to add a Variable “ifCityExists” of “String” Type.
6. Do not provide any Default value to this created Variable “ifCityExists”.

**Parameters <sup>1</sup> Variables <sup>1</sup> Settings Output** ^

+ New | Delete

Name	Type	Default value
ifCityExists	String	Value

7. Drag and drop the if condition as shown below.

The screenshot shows the Azure Data Factory pipeline editor interface. On the left, there's a tree view labeled "Activities" with a search bar containing "if". Below it is a section for "Iteration & conditionals" with an "If Condition" activity selected. On the right, the main workspace displays an "If Condition" activity. This activity has two branches: "True" and "False", both of which currently show "No activities". There are edit icons (+, pencil) for each branch. At the top of the workspace, there are buttons for "Validate", "Debug", and "Add trigger".

8. Under the Activity tab give the below expression as shown below.

Exp: @contains(pipeline().parameters.listOfCities,pipeline().parameters.cityToSearch)

General Activities (0)<sup>1</sup> User properties

Expression ⓘ @contains(pipeline().parameters.list...)

9. Now, to add the Activity to perform if the logical expression provided in the Textbox “Expression” becomes True, click on the pencil icon for “True” displayed under the “Case” section.

General Activities (0)<sup>1</sup> User properties

Case	Activity
True	No activities
False	No activities

10. Inside the “True activities” design pane, search for the “Set variable” Activity in the Activity Explorer. Once found, drag and drop the Activity to the design pane.



11. Under settings select the Name and give the below value.

General Settings User properties

Variable type ⓘ	<input checked="" type="radio"/> Pipeline variable <input type="radio"/> Pipeline return value
Name *	ifCityExists
Value *	Yes. It Exists

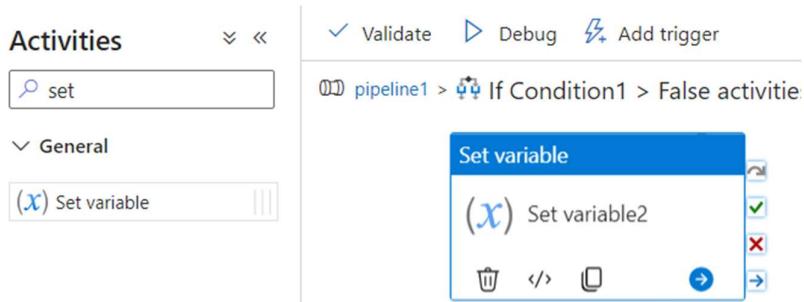
12. Now go to pipeline and click on the False pencil symbol.

General Activities (1)<sup>1</sup> User properties

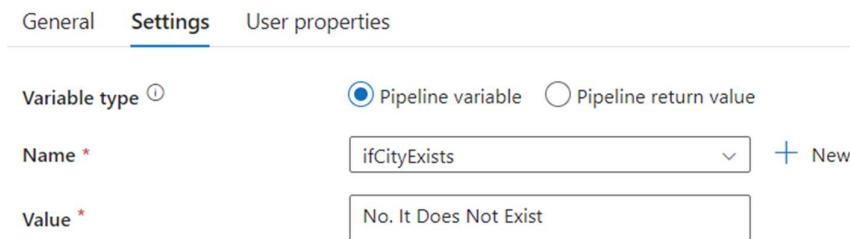
Expression ⓘ @contains(pipeline().parameters.list...)

Case	Activity
True	(x) Set variable1 1 Activity
False	No activities

13. Drag and drop the Set variable as shown below.

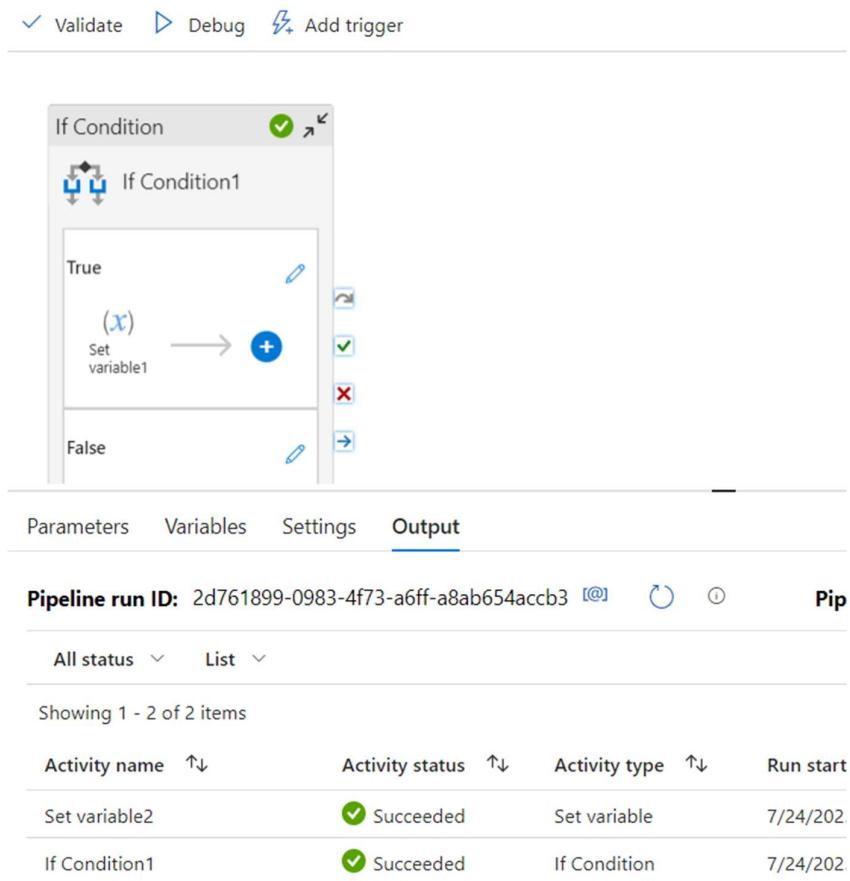


14. Under the settings give the name and value as shown below.

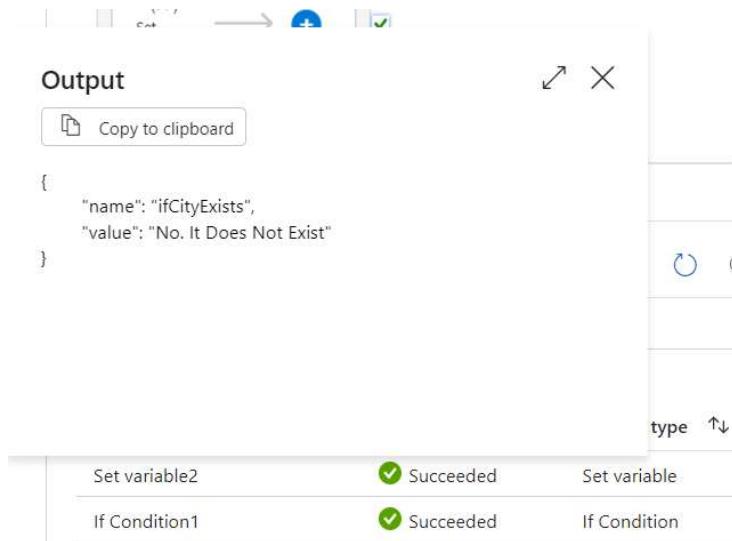


15. Now go back to the pipeline and validate it then click on Debug.

16. Here our pipeline runs successfully/



17. In the “Output” tab, of the “Set variable” Activity attached to the “False” Condition in the “If Condition” Activity, the output is displayed as “No. It Does Not Exist”.



## User Defined Functions

1. In this example we are going to below CSV file data.

gender		
File	Edit	View
<pre>id,name,gender 1,maheer,male 2,pradeep,male 3,swetha,female 4,annu,female 5,himanshu,male 6,shakti, 7,swati,female</pre>		

2. Upload this in the Azure Blob Storage source container folder.

gender.csv	7/24/2023, 12:03:54 ...	Hot (Inferred)

3. So, in this example we are going to create a column that indicates a numeric number based on gender.
4. If the gender is male the value will be 1, if the gender is female the value will be 0, or if there is no value it will be -1.

5. Go to Azure Data Factory and create a dataset for this CSV file.

#### Set properties

Name  
GenderDataset

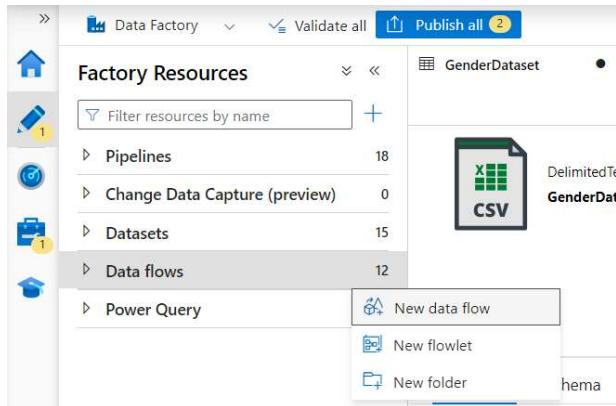
Linked service \*  
AzureBlobStorage\_txt

File path  
f-demo / Directory / gender.csv

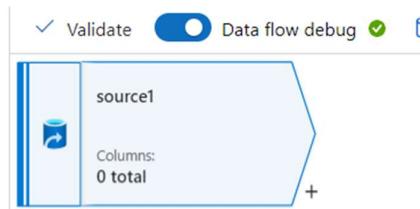
First row as header

Import schema  
 From connection/store  From sample file  None

6. Create a Dataflow.



7. Add a Data source.



8. Select the gender dataset that we created before.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name \* source1 Learn more

Description Import data from GenderDataset Reset

Source type \* Dataset

Dataset \* GenderDataset Test connection Open New

Options Allow schema drift

9. Under projection set the Id type as shown below.

Column name	Type	Format
id	integer	Specify format
name	string	Specify format
gender	string	Specify format

10. Go to Manage and click on Data flow libraries then click on New.

11. Give the name and click on the plus symbol.

Name	Return Type	Body

12. Give the name and add an argument then give the below expression.

Exp: `iif(i1=='male',1,iif(i1=='female',0,-1))`

New data flow function

Name	Type
i1	string

13. Click on save.

Name \*

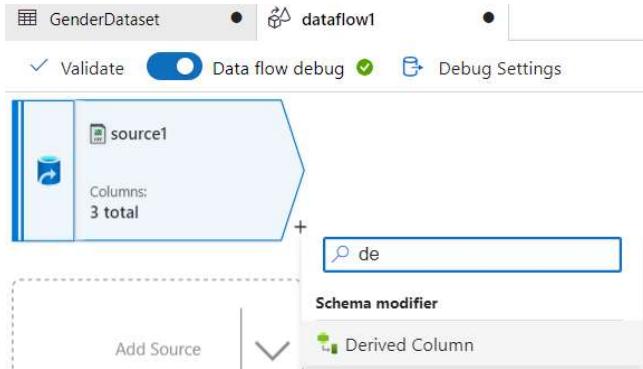
Description

Functions

+ New

Name	Return Type	Body
gendervalue	integer	iif(i1=='male',1,iif(i1=='female',0,-1))

14. Go to Dataflow click on the plus symbol and click on the derived column as shown below.



15. Under the column select the gender and give the below expression as shown below.

Exp: gendervalue(gender)

Derived column's settings    Optimize    Inspect    Data preview

Description

Creating/updating the columns 'id, name, gender'

Incoming stream \*

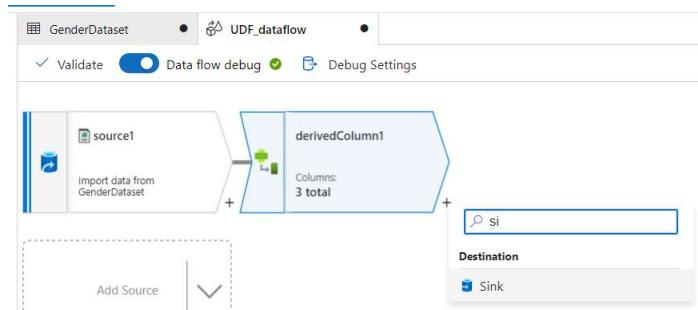
source1

+ Add    Clone    Delete    Open expression builder

Columns \* (1)

Column	Expression
<input type="checkbox"/> gender	gendervalue(gender)

16. Click on Plus and click on Sink as shown below.



17. Under sink select the destination dataset.

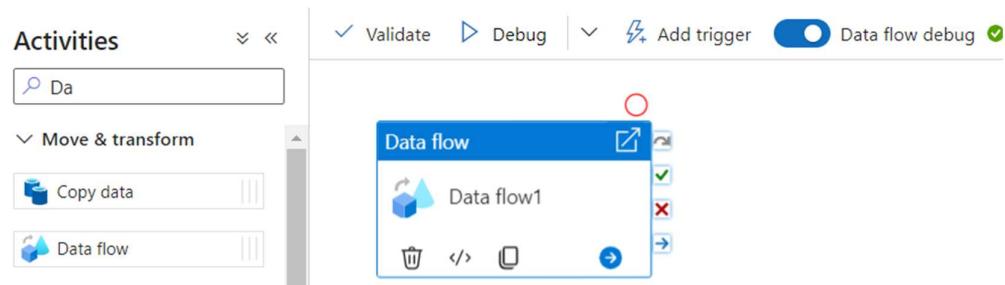
This screenshot shows the 'Sink' configuration page. It includes fields for 'Output stream name' (set to 'sink1'), 'Description' (set to 'Export data to ds\_sink\_txt'), 'Incoming stream' (set to 'derivedColumn1'), and 'Sink type' (set to 'Dataset'). Below these, there's a 'Dataset' dropdown set to 'ds\_sink\_txt', and buttons for 'Test connection', 'Open', and 'New'. The top navigation bar has tabs for 'Sink', 'Settings', 'Errors', 'Mapping', 'Optimize', 'Inspect', and 'Data preview'.

18. Under the setting select the file option and file name as shown below.

This screenshot shows the 'Settings' tab for the sink. A message box at the top states: 'This sink currently has Single partition set in Optimize. This will make your data flow execution longer.' Below this, there are settings for 'Clear the folder' (checkbox), 'File name option' (dropdown set to 'Output to single file'), and 'Output to single file' (text input field containing 'UDF\_table.csv').

19. Create a pipeline.

20. Drag and drop the Data flow activity as shown below.



21. Under setting select the data flow.

The screenshot shows the 'Settings' tab of the Azure Data Flow configuration interface. It includes fields for 'Data flow' (set to 'UDF\_dataflow'), 'Run on (Azure IR)' (set to 'AutoResolveIntegrationRuntime'), and 'Compute size' (set to 'Small'). Below the configuration is a toolbar with buttons for Validate, Debug, Add trigger, and Data flow debug, with the 'Data flow debug' button being active.

22. Validate the pipeline and click on debug.

23. Here our pipeline was executed successfully.

The screenshot shows the pipeline run history for 'Data flow1'. The pipeline run ID is 43764c16-9842-4b87-b0a1-e8cd671e0789. The run status is 'Succeeded' and it is a 'Data flow' run started on 7/24/2022. The table below lists the activity details.

Activity name	Activity status	Activity type	Run start
Data flow1	Succeeded	Data flow	7/24/2022

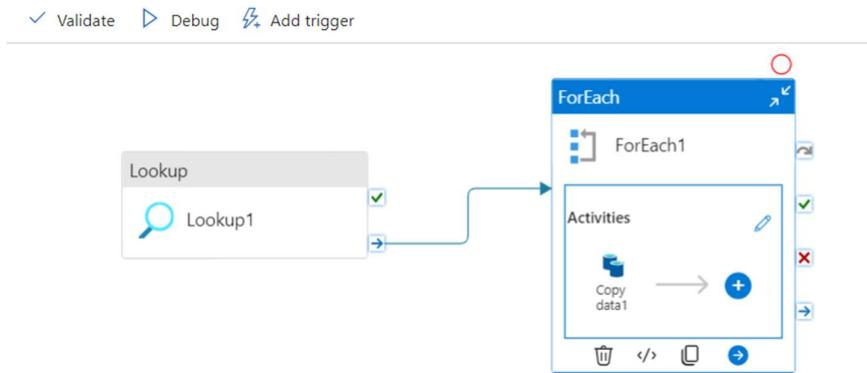
24. Now go to the destination and check the file as shown below.

The screenshot shows the 'Output/UDF\_table.csv' file in Azure Blob storage. The file contains the following data:

id	name	gender
1	maheer	1
2	pradeep	1
3	swetha	0
4	annu	0
5	himanshu	1
6	shakti	-1
7	swati	0

# Error Handling

1. In this Example we are using the Lookup activity that we created in the previous session.



2. In the Lookup activity, check the First row-only option as shown below.

General    **Settings**    User properties

Source dataset \*

First row only

Use query

Query \*

```
select * from  
sqldatabase.Information_Schema.Tables  
where table_type = 'Base Table'
```

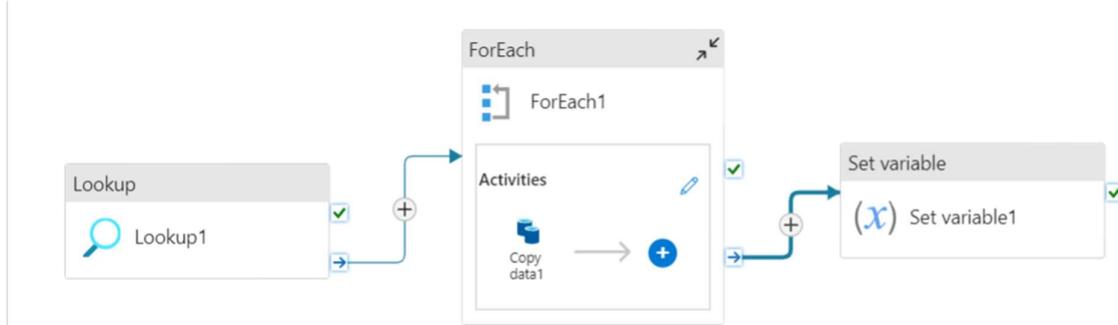
3. For the pipeline create a below Variable.

Parameters    **Variables**    Settings    Output

+ New | - Delete

Name	Type	Default value
Error	String	Value

4. Drag and drop the Set variable and make the connection as shown below.



5. Under Settings, select the Name and give the below Expression.

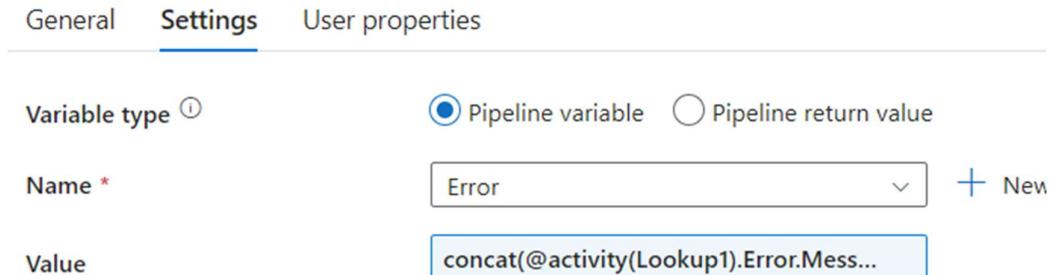
Exp: concat(@activity(Lookup1).Error.Message,'|',@activity(ForEach1).Error.Message)

General    **Settings**    User properties

Variable type ⓘ  Pipeline variable  Pipeline return value

Name \* Error [New](#)

Value concat(@activity(Lookup1).Error.Mess...)



6. Now validate the pipeline and click on Debug.

7. Here you can see that the ForEach loop failed because of Lookup. You will see the Error message in the Set variable output.

Activities << [Validate](#) [Debug](#) [Add trigger](#)

Synapse  
Notebook  
Spark job definition

Batch Service

Custom

General  
[Set variable](#)

Output [Copy to clipboard](#)

```
{
  "name": "Error",
  "value": "|The expression 'length(activity('Lookup1').output.value)' cannot be evaluated because property 'value' doesn't exist, available properties are 'firstRow', 'effectiveIntegrationRuntime', 'billingReference', 'durationInQueue'."
}
```

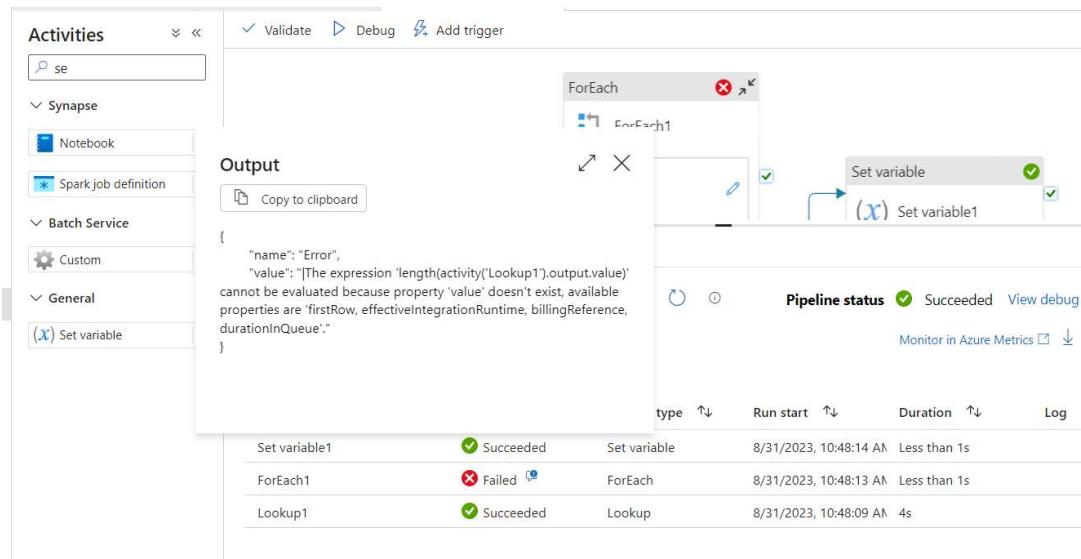
ForEach [Edit](#) [Delete](#)

Set variable [Edit](#) [Delete](#)

Pipeline status [Succeeded](#) [View debug](#)

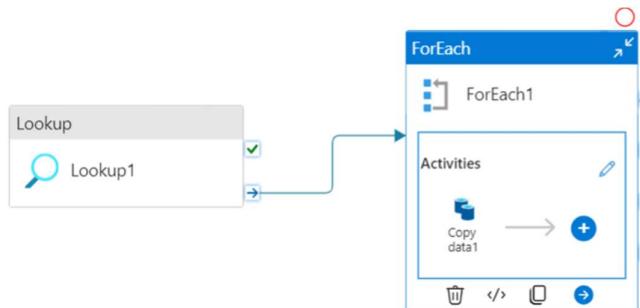
Monitor in Azure Metrics

type	Run start	Duration	Log
Set variable1	<span style="color: green;">✓</span> Succeeded	Set variable	8/31/2023, 10:48:14 AM Less than 1s
ForEach1	<span style="color: red;">✗</span> Failed <a href="#">View</a>	ForEach	8/31/2023, 10:48:13 AM Less than 1s
Lookup1	<span style="color: green;">✓</span> Succeeded	Lookup	8/31/2023, 10:48:09 AM 4s



# Create Alert

1. In this example also we are using the same Lookup Activity.
2. Just remove the Set Variable Activity that we used in the previous example.



3. In the Monitor, under Alerts and Metrics click on New alert rule.

4. Give the below properties and click on the Target criteria plus symbol.

## New alert rule

Alert rule name \*

Description

Severity \*

Target criteria \* ⓘ

+ Add criteria

5. Select the Failed Pipeline runs metrics and click on continue.

#### Add criteria

Select one metric to set up the alert condition.

Metrics ↑↓

- External capacity utilization of MVNet integration runtime
- External waiting queue length of MVNet integration runtime
- Failed activity runs metrics
- Failed pipeline runs metrics**
- Failed SSIS integration runtime start metrics
- Failed SSIS package execution metrics
- Failed trigger runs metrics
- Integration runtime available memory
- Integration runtime available node count
- Integration runtime CPU utilization
- Integration runtime queue duration
- Integration runtime queue length
- Maximum allowed entities count
- Maximum allowed factory size (GB unit)
- Pipeline available capacity percentage of MVNet integration runtime
- Pipeline capacity utilization of MVNet integration runtime

**Continue**

**Cancel**

6. Here select all options for Name and Failure Type. Then give the Alert Logic and Evaluate based on Properties as shown below.

#### Configure alert logic

Selecting the dimension values will help you filter to the right time series.

Dimension	Values
Name	41 selected
FailureType	3 selected

#### Alert logic

Condition * ⓘ	<input type="text" value="Greater than"/>
Time aggregation * ⓘ	<input type="text" value="Total"/>
Threshold count * ⓘ	<input type="text" value="0"/>

#### Evaluate based on

Period * ⓘ	<input type="text" value="Over the last 5 minutes"/>
Frequency * ⓘ	<input type="text" value="Every 1 minute"/>

**Add criteria**

**Back**

**Cancel**

7. Here we created our Target criteria and now click on the Plus symbol of the Configure notification.

Target criteria	Actions
Whenever Pipeline Failed Runs metric is Greater	

[+ Add criteria](#)

**i** There will be a monthly rate for the configured criteria. [Learn more](#)

Configure Email/SMS/Push/Voice notification \*

[+ Configure notification](#)

Enable rule upon creation

On

8. Give the names and click on the Notification plus symbol.

## Configure notification

Notify your team via email and text messages or automate actions functions logic apps or integrating with external ITSM solutions.

Create new  Use existing

Action group name \*

emailnotify

Short name \*

en

Notifications \*

[+ Add notification](#)

9. Give the name and set the below properties then click on Add Notification.

#### Add notification

Learn more about [Pricing](#) and [Privacy statement](#).

Action name \*

DemoAction

#### Select which notifications you'd like to receive

Email

lacchi.balaji@qisul.co.in

SMS

Country code

1

Phone number \*

1234567890

Carrier charges may apply.

Azure app push notifications

Enter your email used to log into your Azure account. [Learn about connecting to your Azure resources using the Azure app](#).

email@example.com

Voice

Country code

1

Phone number \*

1234567890

[Add notification](#)

[Cancel](#)

10. Now click on

#### Configure notification

Notify your team via email and text messages or automate actions using webhooks, runbooks, functions logic apps or integrating with external ITSM solutions.

Create new  Use existing

Action group name \*

emailnotify

Short name \*

en

Notifications	Action type	Actions
DemoAction	Email/SMS/Push/Voice	 

[+ Add notification](#)

[Add action group](#)

[Cancel](#)

11. Next click on Create Alert Rule.

New alert rule

Alert rule name \*  
SampleAlert

Description

Severity \*  
Sey1

Target criteria  
Actions  
Whenever Pipeline Failed Runs metric is Greater Edit

+ Add criteria

There will be a monthly rate for the configured criteria. Learn more about Pricing

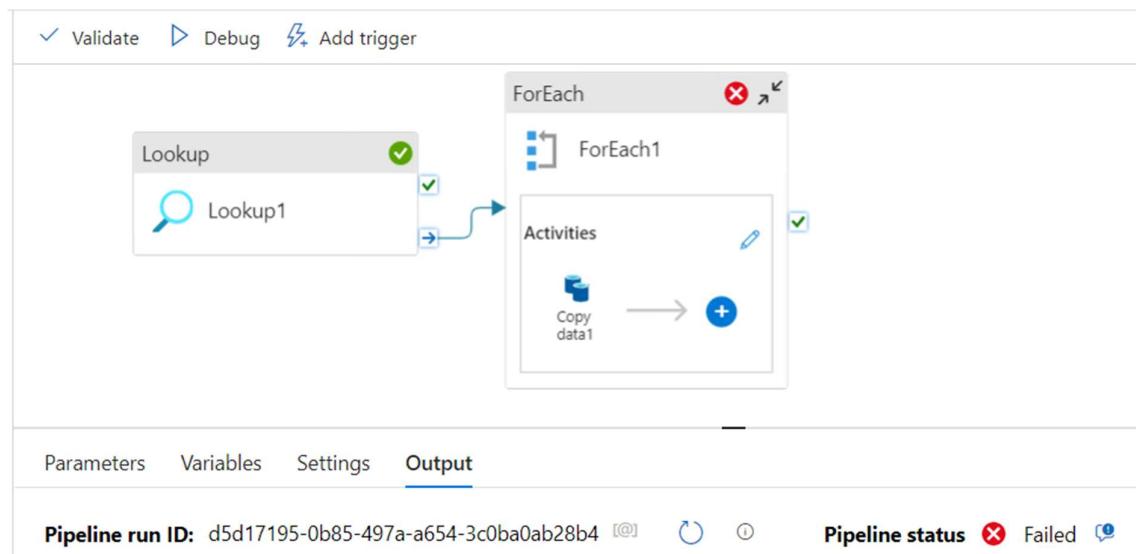
Notifications Action group type Actions  
emailNotify 1 Email Edit

+ Configure notification

Create alert rule Cancel

12. Now go to our pipeline and click on debug.

13. Here our pipeline failed.



14. Go to the Home page, under the monitor, in the Alert click on Alert Rule.

Home > Monitor

**Monitor | Alerts** Microsoft

Search View as timeline (preview) + Create Alert rules Action groups Alert processing rules

Overview Activity log Alerts Matrices

New: View alerts visualized on a timeline for a clearer picture of your events. You can switch between views anytime.

Subscription : df295c6e-136c-4598-9168-19fa9dba7fe1 Time range

15. You will see the alert that we created with 1 error as shown below.

## Alert rules ...

Subscription : Microsoft Partner Network							No grou
Name ↑↓	Condition	Severity ↑↓	Target scope	Target resource type	Signal type ↑↓	Status ↑↓	
Failure Anomalies - DemoMaheen	Failure Anomalies detected	3 - Informational	demomaheen	Application Insights	Smart detector	Enabled	
SampleAlert	PipelineFailedRuns > 0	1 - Error	AzureDataFactoryTraing	Data factory (V2)	Metrics	Enabled	

## Create Azure Automation

1. In the Home click on Create Resource and go to All services then search for Automation account and click on it.

The screenshot shows the Microsoft Azure 'All services' search interface. A search bar at the top contains the query 'Automation'. Below the search bar, there are three main search results:

- Automation Accounts**: Keywords: automate, process automation, automate; Resource type: Micro... (with a 'PREVIEW' link)
- Automanage**: Keywords: automanage, azure automanage
- Power Platform**: Keywords: process automation, robotic process automation

On the left side, there is a sidebar with sections for Favorites, Recents, Recommended, Categories (AI + machine learning, Analytics), and a 'Give feedback' button.

2. Next click on Create.

The screenshot shows the 'Automation Accounts' blade in the Azure portal. The title bar says 'Automation Accounts' and includes a 'Create' button. The main content area displays a message: 'No automation accounts to display' with a gear icon. Below this, it says 'Try changing or clearing your filters.' and has a 'Create automation account' button. At the bottom right, there is a 'Give feedback' link.

### 3. Create a New Resource for the Automation Account.

All services > Automation Accounts >

#### Create an Automation Account ...

Basics Advanced Networking Tags Review + Create

Create an Automation Account to hold the Automation runbooks & configuration used for automating operations and management tasks around Azure and non-Azure resources. You could execute cloud jobs in a serverless environment or use hybrid jobs on your compute via Azure Virtual machines, Arc-enabled servers or Arc-enabled VMWare VM (preview). [Learn more](#)

Subscription \* ⓘ Microsoft Partner Network

Resource group \* ⓘ Select a resource group Create new

Instance Details

Automation account name \* ⓘ

Region \* ⓘ Automation

A resource group is a container that holds related resources for an Azure solution.

Name \* Automation

OK Cancel

### 4. Give the name and click on Review+Create.

#### Create an Automation Account ...

Basics Advanced Networking Tags Review + Create

Create an Automation Account to hold the Automation runbooks & configuration used for automating operations and management tasks around Azure and non-Azure resources. You could execute cloud jobs in a serverless environment or use hybrid jobs on your compute via Azure Virtual machines, Arc-enabled servers or Arc-enabled VMWare VM (preview). [Learn more](#)

Subscription \* ⓘ Microsoft Partner Network

Resource group \* ⓘ (New) Automation Create new

Instance Details

Automation account name \* ⓘ AutomationAccount1

Region \* ⓘ East US

Review + Create

Previous

Next

5. Next click on Create.

## Create an Automation Account ...

Validation passed

Basics   Advanced   Networking   Tags   Review + Create

**Basics**

Name	AutomationAccount1
Subscription	Microsoft Partner Network
Resource group	Automation
Region	East US

**Advanced**

System assigned identity	Yes
User assigned identity	None

**Networking**

Network connectivity	Public access
----------------------	---------------

**Tags**

**Create**   **Previous**   **Next**

6. We have created our Automation Account now click on Go To Resource.

**Microsoft.AutomationAccount | Overview** ⚡ ...

Deployment

Search   «   Delete   Cancel   Redeploy   Download   Refresh

**Overview**   **Inputs**   **Outputs**   **Template**

**Your deployment is complete**

Deployment name : Microsoft.AutomationAccount  
Subscription : Microsoft Partner Network  
Resource group : Automation

Start time : 8/31/2023, 2:47:39 PM  
Correlation ID : 01f6399d-6557-4ffd-9505-35616573c2ef

> Deployment details  
▽ Next steps

**Go to resource**

# Create Schedule

7. Under Schedules click on Add a Schedule option.

The screenshot shows the 'AutomationAccount1 | Schedules' section of the Microsoft Automation Account Overview. The left sidebar includes links for Runbooks, Jobs, and Hybrid worker groups under Process Automation; Inventory, Change tracking, and State configuration (DSC) under Configuration Management; and Update management. The right main area has a search bar, a 'Add a schedule' button, and a 'Feedback' link. It displays a table with columns 'Name' and 'Next run'. A message says 'No schedules found.'

8. Give the Name and the below properties.

### New Schedule

X

Name \*

 ✓

Description

Starts \* ⓘ

01/09/2023 Calendar icon 8:00 AM

Time zone

 ▼

Recurrence

Once Recurring

9. Next set the below remaining properties and click on Create.

The screenshot shows the 'Recurrence' section of the Azure Automation schedule creation interface. It includes fields for 'Recur every \*' (set to '1 Week'), 'On these days' (checkboxes for Monday through Friday), and 'Set expiration' (radio buttons for 'Yes' or 'No'). Below these, there are 'Expires' and 'Never' options. A large blue 'Create' button is at the bottom.

10. Here we have created our schedule.

Name	Next run	Time zone	Status
Weekdays at 8AM	9/1/2023, 8:00 AM	India Standard Time	✓ On

11. Under Runbook by default we have 2 RunBooks open the first one.

The screenshot shows the 'Runbooks' section of the Azure Automation portal. It includes a search bar, filter buttons for 'Runbook type: All' and 'Authoring status: All', and a table listing two runbooks. The table columns are Name, Authoring status, Runbook type, and Runtime version. Both runbooks listed are 'Published' and 'PowerShell' type.

Name	Authoring status	Runbook type	Runtime version
AzureAutomationTutorialWithI...	Published	PowerShell	5.1
AzureAutomationTutorialWithI...	Published	Graphical PowerShell	5.1

12. Under the Schedule click on Add a Schedule.

The screenshot shows the Azure Automation Runbook interface. On the left, there's a sidebar with icons for Overview, Activity log, Tags, and Diagnose and solve problems. Below that is a 'Resources' section with icons for Jobs, Schedules (which is highlighted), and Webhooks. At the top right, there's a search bar, an 'Add a schedule' button, and a 'Refresh' button. The main area has a heading 'AzureAutomationTutorialWithIdentity (AutomationAcc)' and a 'Runbook' label. It shows a 'Name' field with the placeholder 'No schedules found.'

13. Next click on Link a schedule to your runbook.

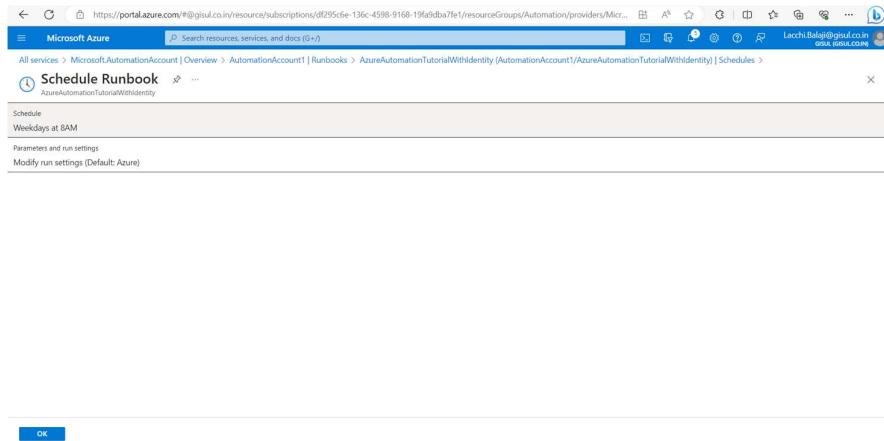
The screenshot shows the 'Schedule Runbook' page for 'AzureAutomationTutorialWithIdentity'. It features a clock icon and the title 'Schedule Runbook'. Below it is a link to 'AutomationAccount1 | Runbooks'. The main content area is titled 'Schedule' and contains the instruction 'Link a schedule to your runbook'. There's also a 'Parameters and run settings' section with a 'Modify run settings (Default: Azure)' link.

14. Click on a schedule that we created.

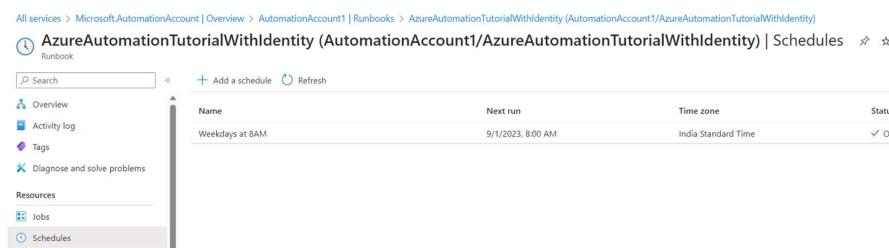
The screenshot shows the 'Schedules' page for 'AutomationAccount1/AzureAutomationTutorialWithIdentity'. It has a 'Add a schedule' button. A single schedule is listed with the name 'Weekdays at 8AM' and a 'Next run' time of '9/1/2023, 8:00 AM'.

Name	Next run
Weekdays at 8AM	9/1/2023, 8:00 AM

15. Now click on OK.



16. Now here you can see we added a schedule to the Runbook.



## Flowlets

1. In this example we are using the below Duplicate\_Emp data.

Duplicate_Emp.csv		
Blob		
<a href="#">Upload</a>	<a href="#">Change access level</a>	...
<a href="#">output_2</a>		...
<a href="#">output_3</a>		...
<a href="#">Output1</a>		...
<a href="#">Switch1</a>		...
<a href="#">Cast_src.csv</a>		...
<a href="#">demo.json</a>		...
<a href="#">Department.csv</a>		...
<a href="#">Duplicate_Dept.csv</a>		...
<a href="#">Duplicate_Emp.csv</a>		...

Overview Versions Snapshots Edit Generate SAS

id	name	gender
1	Mahesh	male
2	Ravi	male
1	Mahesh	male

[Edit](#)

2. And below Duplicate\_Dept data also.

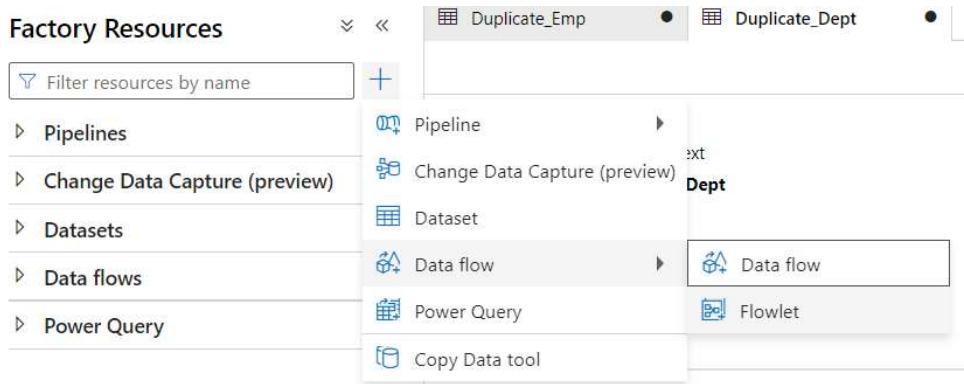
Duplicate_Dept.csv		
Blob		
<a href="#">Upload</a>	<a href="#">Change access level</a>	...
<a href="#">output_2</a>		...
<a href="#">output_3</a>		...
<a href="#">Output1</a>		...
<a href="#">Switch1</a>		...
<a href="#">Cast_src.csv</a>		...
<a href="#">demo.json</a>		...
<a href="#">Department.csv</a>		...
<a href="#">Duplicate_Dept.csv</a>		...
<a href="#">Duplicate_Emp.csv</a>		...

Overview Versions Snapshots Edit Generate SAS

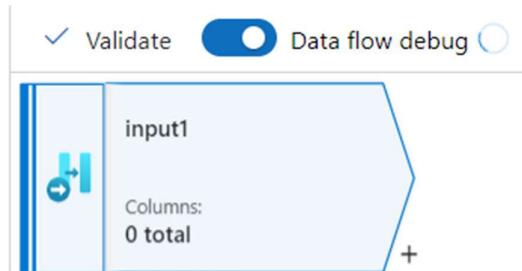
id	name
1	IT
2	HR
3	Payroll
2	HR

[Edit](#)

3. Create a Datasets for the above csv files.
4. Click on the plus symbol and go to dataflow then click on Flowlet.



5. Add an input.



6. Under settings add below three columns.

Input settings	Optimize	Inspect
Output stream name *	input1	<a href="#">Learn more</a>
Allow schema drift	<input checked="" type="checkbox"/>	
Columns *	Column name	Type
	id	abc string
	name	abc string
	gender	abc string

7. Add aggregate transformation.



8. In the Settings, add the below columns under Group By.

Aggregate settings    Optimize    Inspect

Output stream name \*  Learn more [🔗](#)

Description

Incoming stream \*

[Group by](#) [Aggregates](#)

Columns	Name as
abc id	id
abc name	name
abc gender	gender

9. Under Aggregate give the column and expression.

Incoming stream \*

[Group by](#) [Aggregates](#)

Grouped by: id, name, gender

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

Column	Expression
Count	count()

10. Add Select Transformation.



11. Under Settings delete the count column.

Select settings    Optimize    Inspect

columns: id, name, gender

Incoming stream \*

Options

- Skip duplicate input columns  ⓘ
- Skip duplicate output columns  ⓘ

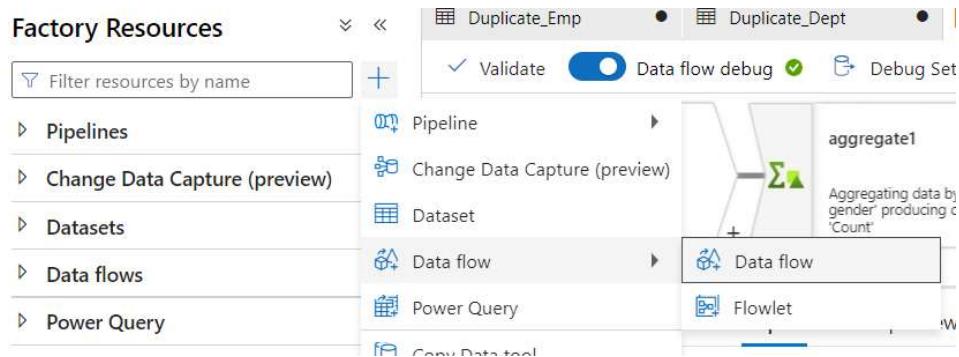
Input columns \*  Auto mapping  ⓘ [Reset](#)   3 mappings: 1 column(s) from the inputs left unmapped

aggregate1's column	Name as
abc id	id
abc name	name
abc gender	gender

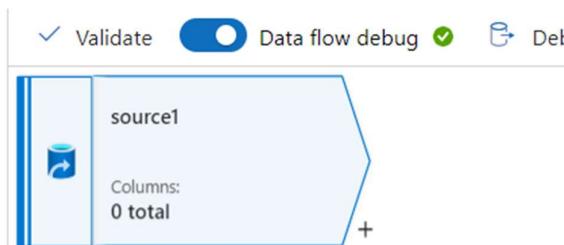
## 12. Add Output.



## 13. Next create a Dataflow.

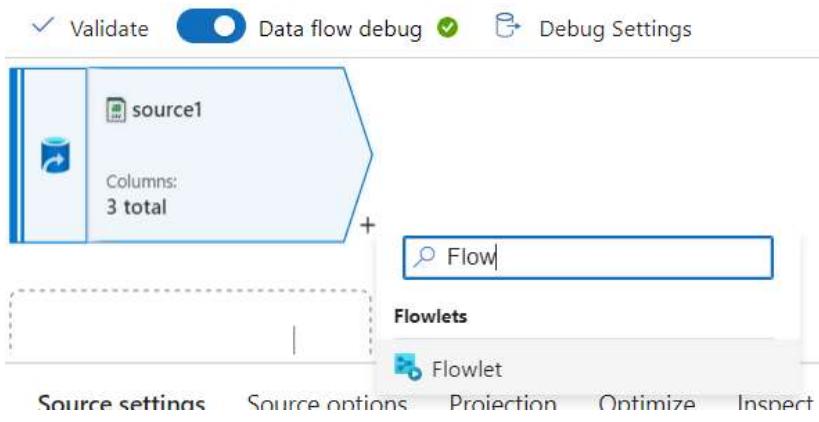


## 14. Add a Dataflow.



## 15. Add an Emp Source Dataset.

16. Click on the plus symbol and click on Flowlet.



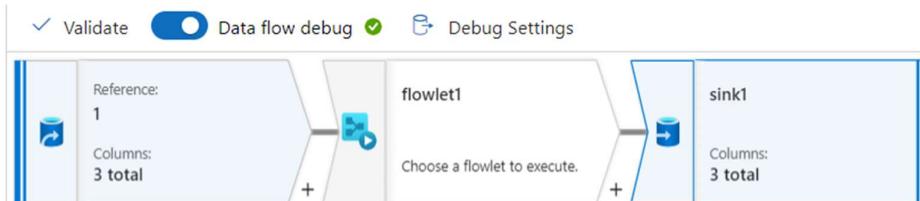
17. Under flowlet select the flowlet that we created before.

The screenshot shows the 'Flowlet settings' tab selected. It contains the following fields:

- Output stream name \*: flowlet1
- Description: Choose a flowlet to execute.
- Incoming stream \*: source1
- Flowlet \*: flowlet

Below the form are buttons for 'Open' and 'New'.

18. Add Sink.



19. Select the Target dataset.

The screenshot shows the 'Sink' tab selected. It contains the following fields:

- Sink
- Settings
- Errors
- Mapping
- Optimize
- Inspect
- Data preview

Fields specific to the Sink tab:

- Output stream name \*: sink1
- Description: Export data to ds\_sink\_txt
- Incoming stream \*: flowlet1@output1
- Sink type \*: Dataset (selected)
- Dataset \*: ds\_sink\_txt

Below the form are buttons for 'Test connection', 'Open', and 'New'.

20. Under setting select the File option and give the file name.

Sink    **Settings**    Errors    Mapping    Optimize    Inspect    Data preview

**File name option \*** Output to single file

**Output to single file \*** Emp\_Flowlet.csv

This sink currently has Single partition set in Optimize. This will make your data flow execution faster.

21. Create a Pipeline. Drag and drop the Data flow.

Activities

Data

Move and transform

Copy data

Data flow

Data flow

22. Select the Dataflow.

General    **Settings**    Parameters    User properties

Data flow \* Flowlet\_Emp\_dataflow

Run on (Azure IR) \* AutoResolveIntegrationRuntime

Compute size \* Small

Open    New

23. Validate the pipeline and click on Debug.

24. Here our pipeline was executed successfully.

Validate    Debug    Add trigger    Data flow debug ✓

Data flow

Parameters    Variables    Settings    **Output**

Pipeline run ID: 280d5a28-42c4-4477-95ff-64f97a57e5be

All status

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start
Data flow1	Succeeded	Data flow	8/25/2023,

25. Go to Destination and check the data.

**FilterOutput/Emp\_Flowlet.csv** ...

Blob

Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

id	name	gender
1	Mahesh	male
2	Ravi	male

**Edit**



26. Now go back to the Data Factory and create a Dataflow.

27. Add a Data source.

✓ Validate  Data flow debug  Delete

source1  
Columns: 0 total



28. Select the Dept source dataset.

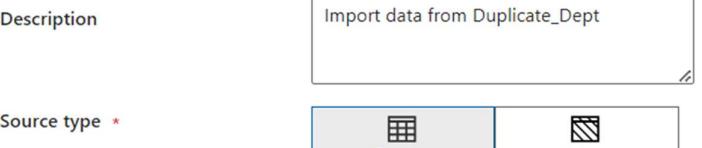
Source settings Source options Projection Optimize Inspect Data preview

Description Import data from Duplicate\_Dept  Reset

Source type \*  Dataset  Inline

Dataset \*  Duplicate\_Dept  Test connection  Open  New

Options  Allow schema drift



29. Add Derived column.

Reference: 1  
Columns: 2 total

derivedColumn1  
Columns: 2 total



30. Under settings give the column name and a dummy value under expression as shown below.

Derived column's settings    Optimize    Inspect    Data preview ●

Output stream name \*  Learn more ↗

Description  ⌂ Reset

Incoming stream \*  + Add ⌂ Clone ⌂ Delete ⌂ Open expression builder

Columns \* ⓘ

Column	Expression
<input type="checkbox"/> gender	'abcd'

31. Add flowlet.



32. Under Settings select the flowlet that we created before.

Flowlet settings    Mapping    Parameters    Inspect    Data preview ●

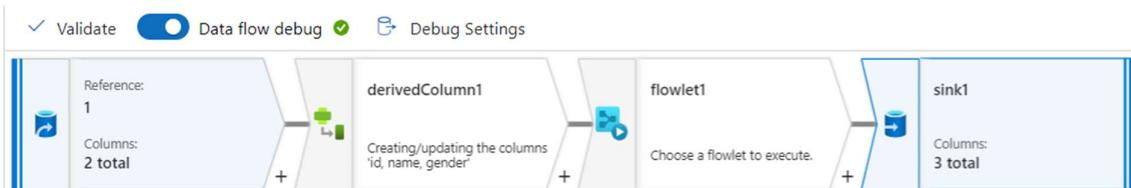
Output stream name \*  Learn more ↗

Description  ⌂ Reset

Incoming stream \*  +

Flowlet \*  ⌂ Open ⌂ New

33. Add Sink.



34. Select the Destination dataset.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

Description: Export data to ds\_sink\_txt

Incoming stream: flowlet1@output1

Sink type: Dataset

Dataset: ds\_sink\_txt

Skip line count:

Options: Allow schema drift

35. Under settings select the File option and give the file name as shown below.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

i This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The recommended set

Clear the folder:

File name option: Output to single file

Output to single file: Dept\_Flowlet.csv

36. Under mapping delete the gender column.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

i This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The recommended set

Options: Skip duplicate input columns, Skip duplicate output columns

Auto mapping:

Reset:

Add mapping:

Delete:

Output format:

Input columns: id → id, name → name

Output columns: id, name

37. Create a Pipeline. Drag and drop the Dataflow.

Activities: Data flow, Copy data, Data flow

Validate:

Debug:

Data flow: Data flow1

38. Under settings select Dataflow.

General    **Settings**    Parameters    User properties

---

Data flow \*    flowlet\_dept\_dataflow    Open   

Run on (Azure IR) \* ⓘ    AutoResolveIntegrationRuntime   

Compute size \* ⓘ    Small   

39. Validate the pipeline and click on debug.

40. Here our pipeline was executed successfully.

Data flow   

Data flow1   

---

Parameters    Variables    Settings    **Output**

Pipeline run ID: fa662c4a-2051-4d8b-803b-239b98a36738             **Pipe**

All status   

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start
Data flow1	Succeeded	Data flow	8/25/2023

41. Go to Destination and check the data.

**FilterOutput/Dept\_Flowlet.csv**    ...

Blob

Save    Discard    Download    Refresh    Delete

---

Overview    Versions    Snapshots    **Edit**    Generate SAS

id	name
1	IT
3	Payroll
2	HR

Edit

# Slowly Changing Dimension Type 1

1. In this example we are using the below source csv file. I uploaded the file into my Storage account.

The screenshot shows the Azure Storage Blob view for the file `SCD1/SCD1_src.csv`. The blob contains the following data:

empid	name	country	department
1	maheer	india	2
2	asmin	india	1
3	mahesh	india	3
4	aman		2
5	anil	india	4

2. And I have a target table with some dummy data.

```
Query: create table tbl_scd1(
empid int,
name varchar(30),
country varchar(30),
department int)
insert into tbl_scd1 values(1,'ABCD','ADF',1),(2,'XYZ','ASF',1)
```

The screenshot shows the Azure Data Studio query editor with the following code:

```
1  create table tbl_scd1(
2    empid int,
3    name varchar(30),
4    country varchar(30),
5    department int)
6
7  insert into tbl_scd1 values(1,'ABCD','ADF',1),(2,'XYZ','ASF',1)
8
9  Select * From tbl_scd1
```

The Results tab displays the data from the target table:

empid	name	country	department
1	ABCD	ADF	1
2	XYZ	ASF	1

3. Go to Azure Data Factory and create a Dataset for the source file.
4. Select the Azure Blob Storage and click on continue. Select the Delimited Text and click on continue.
5. Give the name, select the Linked service and file then set the below properties.

#### Set properties

Name

Linked service \*  
 

File path  
 /  /   | 

First row as header

Import schema  
 From connection/store  From sample file  None

6. Create a dataset for the target table.
7. Select the Azure SQL Database and click on continue.
8. Give the name, select the linked service and select the table then set the below properties.

#### Set properties

Name

Linked service \*  
 

Table name  
    
 Edit

Import schema  
 From connection/store  None

9. Create a Dataflow and add source.



10. Under source settings select the source dataset.

Source settings   Source options   Projection   Optimize   Inspect   Data preview

Output stream name \*  Learn more [🔗](#)

Description  [↻](#) Reset

Source type \*  Dataset  Inline

Dataset \*  [🔗](#) Test connection [🔗](#) Open [+](#) New

Options  Allow schema drift  [ⓘ](#)

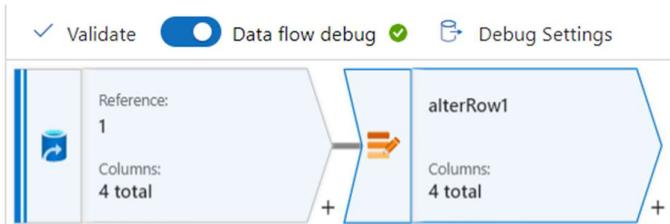
11. Under projection set the datatypes.

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

[Define default format](#) [Detect data type](#) [Import projection](#) [↻](#) Reset schema

Column name	Type	Format
empid	123 integer	<a href="#">Specify</a>
name	abc string	<a href="#">Specify</a>
county	abc string	<a href="#">Specify</a>
department	123 integer	<a href="#">Specify</a>

12. Add alter row transformation.



13. Under settings, in the Alter Row condition set it as Upsert if and give the below expression.

Exp:  $1 == 1$

Alter row settings   Optimize   Inspect   Data preview

Output stream name \*  Learn more [🔗](#)

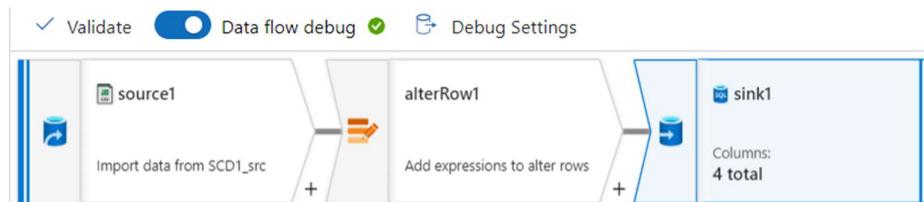
Description  [↻](#) Reset

Incoming stream \*

Alter row conditions \*  [ⓘ](#)

*	Upsert if	<input type="text" value="1 == 1"/>	<a href="#">✖</a>	<a href="#">+</a>	<a href="#">trash</a>
---	-----------	-------------------------------------	-------------------	-------------------	-----------------------

14. Add sink transformation.



15. Under sink select the target dataset.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

**Output stream name \*** sink1   [Learn more](#)

**Description** Export data to SCD1\_tgt   [Reset](#)

**Incoming stream \*** alterRow1

**Sink type \*** [Dataset](#) [Inline](#) [Cache](#)

**Dataset \*** SCD1\_tgt   [Test connection](#) [Open](#) [New](#)

Allow schema drift

16. Under settings select the Update method and key column.

Sink   **Settings**   Errors   Mapping   Optimize   Inspect   Data preview

**Update method**  Allow insert  
 Allow delete  
 Allow upsert  
 Allow update

**Key columns \***  List of columns  Custom expression

123 empid [Add dynamic content \[Alt+Shift+D\]](#) [+](#) [Delete](#)

17. Under mapping, map the fields as shown below.

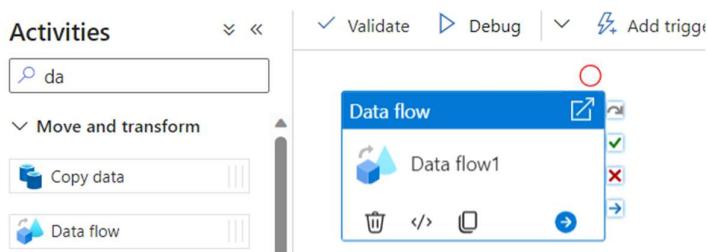
Sink   Settings   Errors   **Mapping**   Optimize   Inspect   Data preview

**Options**  Skip duplicate input columns  
 Skip duplicate output columns

Auto mapping   [Reset](#) [Add](#) [Delete](#) **Output format** 4 mappings: All outputs mapped

Input columns	Output columns
123 empid	123 empid
abc name	abc name
abc county	abc country
123 department	123 department

18. Create a pipeline. Drag and drop the data flow activity.



19. Under settings select the dataflow.

A screenshot of the Pipeline Settings tab. It includes fields for "Data flow" (set to "SCD1\_dataflow"), "Run on (Azure IR)" (set to "AutoResolveIntegrationRuntime"), and "Compute size" (set to "Small"). There is also an "Open" button next to the data flow field.

20. Validate the pipeline and click on debug.

21. Here our pipeline executed successfully.

A screenshot of the Pipeline Overview page for the pipeline "SCD1\_dataflow". It shows the "Pipeline run ID" as "c81c529b-588c-48c2-8ce1-dc024f900a2a" and the "Pipeline status" as "Succeeded". The status is indicated by a green checkmark icon. Other tabs visible include "Parameters", "Variables", "Settings", and "Output". At the bottom, there is a "Monitor in Azure Metrics" button and a "View details" link.

22. Go to the target table and check the data.

```
8  
9  Select * From tbl_scd1
```

A screenshot of the Query Results page. It shows a table with four columns: "empid", "name", "country", and "department". The data is as follows:

empid	name	country	department
1	maheer	india	2
2	asmin	india	1
3	mahesh	india	3
4	aman		2
5	anil	india	4

## Slowly Changing Dimension Type 2

1. In this example we are using the below Source file.

SCD2/SCD2\_src.csv ...  
Blob  
Save Discard Download Refresh Delete  
Overview Versions Snapshots Edit Generate SAS  
empid empname gender country  
1001 maheer male2 Canada  
1002 Ankitha female India  
1003 Mike male USA  
Edit

2. I have created our target table and inserted a Row in it.

Query: create table tbl\_SCD2(  
surrKey int identity,  
empid int,  
empname varchar(30),  
gender varchar(30),  
country varchar(30),  
isActive int)  
insert into tbl\_SCD2 values(1001,'maheer','male','India',1)

Run Cancel query Save query Export data as Show only Editor  
1 create table tbl\_SCD2(  
2 surrKey int identity,  
3 empid int,  
4 empname varchar(30),  
5 gender varchar(30),  
6 country varchar(30),  
7 isActive int)  
8  
9 insert into tbl\_SCD2 values(1001,'maheer','male','India',1)  
10

Results Messages  
Search to filter items...  
surrKey empid empname gender country isActive  
1 1001 maheer male India 1

23. Go to Azure Data Factory and create a Dataset for the source file.
24. Select the Azure Blob Storage and click on continue. Select the Delimited Text and click on continue.
25. Give the name, select the Linked service and file then set the below properties.

#### Set properties

Name

Linked service \*  
 

File path  
 /  /   | 

First row as header

Import schema  
 From connection/store  From sample file  None

26. Create a dataset for the target table.
27. Select the Azure SQL Database and click on continue.
28. Give the name, select the linked service and select the table then set the below properties.

#### Set properties

Name

Linked service \*  
 

Table name  
  

Edit

Import schema  
 From connection/store  None

29. Create a Dataflow and add a source.



30. Under source settings select the source dataset.

Source settings   Source options   Projection   Optimize   Inspect   Data preview

Output stream name \*  Learn more

Description  Reset

Source type \* Dataset Inline

Dataset \* SCD2\_src Test connection Open New

Options  Allow schema drift

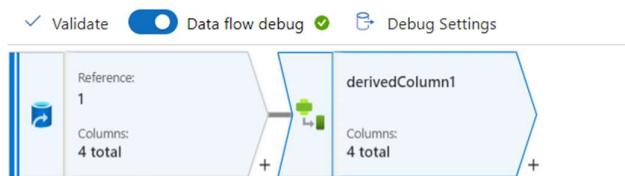
31. Under projection set the datatypes.

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

Define default format Detect data type Import projection Reset schema

Column name	Type	Format
empid	123 integer	
empname	abc string	
gender	abc string	
country	abc string	

32. Add the Derived column.



33. Under settings give the column name and expression as 1 as shown below.

Derived column's settings   Optimize   Inspect   Data preview

Output stream name \*  Learn more

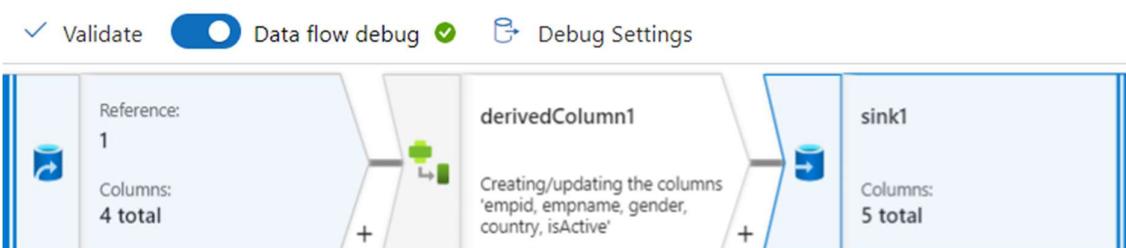
Description  Reset

Incoming stream \*  Add Clone Delete Open expression builder

Columns \*

Column	Expression
isActive	1

34. Add sink.



35. Under sink select the Target dataset as shown below.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

Output stream name \* sink1   Learn more

Description Export data to SCD2\_tgt   Reset

Incoming stream \* derivedColumn1

Sink type \* Dataset   Inline   Cache

Dataset \* SCD2\_tgt   Test connection   Open   New

Options  Allow schema drift

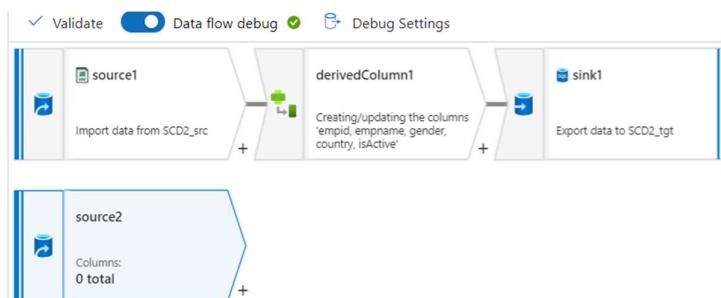
36. Under mapping, map the fields as shown below and delete the SurrKey.

Sink   Settings   Errors   **Mapping**   Optimize   Inspect   Data preview

Options  Skip duplicate input columns    Skip duplicate output columns    Auto mapping   Reset   + Add mapping   - Delete   Output format   5 mappings: 1 colu

Input columns	Output columns
123 empid	123 empid
abc empname	abc empname
abc gender	abc gender
abc country	abc country
123 isActive	123 isActive

37. Add another data source.



38. Select the target dataset.

Source settings   Source options   Projection   Optimize   Inspect   Data preview

Output stream name \* source2   Learn more

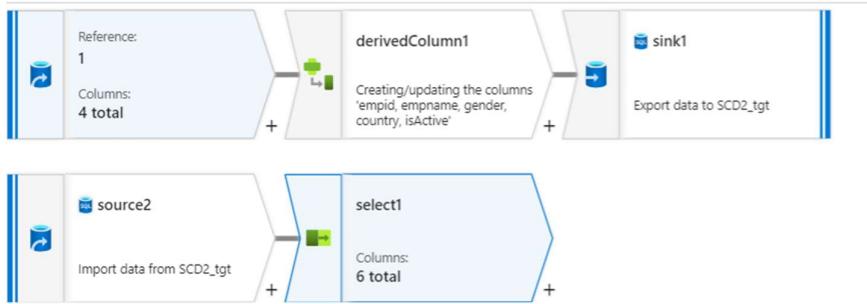
Description Import data from SCD2\_tgt   Reset

Source type \* Dataset   Inline

Dataset \* SCD2\_tgt   Test connection   Open   New

Options  Allow schema drift

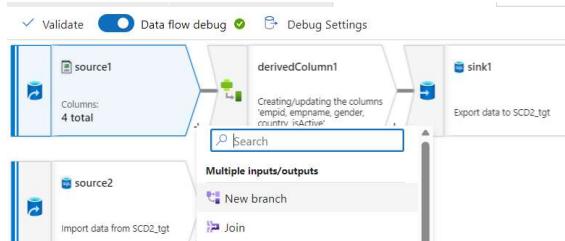
39. Add select transformation.



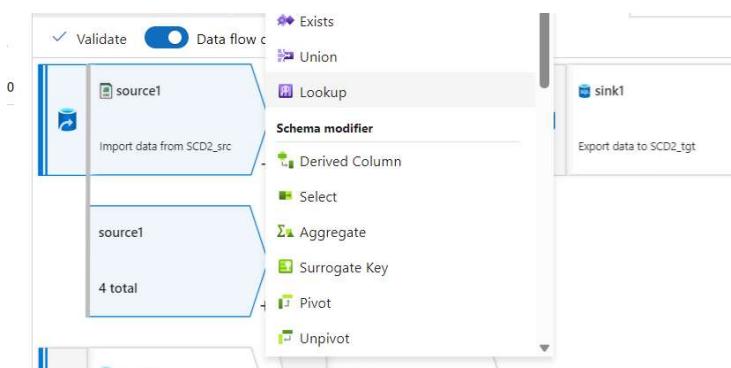
40. Under settings, name the fields as shown below.

source2's column	Name as
surrKey	SQL_surrKey
empid	SQL_empid
empname	SQL_empname
gender	SQL_gender
country	SQL_country
isActive	SQL_isActive

41. Click on the plus symbol of source 1 and click on New Branch.



42. Click on the plus symbol and click on Lookup.



43. Under Lookup settings, select the Lookup stream and give the below condition.

Lookup settings   Optimize   Inspect   Data preview ●

Output stream name \*  Learn more [🔗](#)

Description  [Reset](#)

Primary stream \*

Lookup stream \*

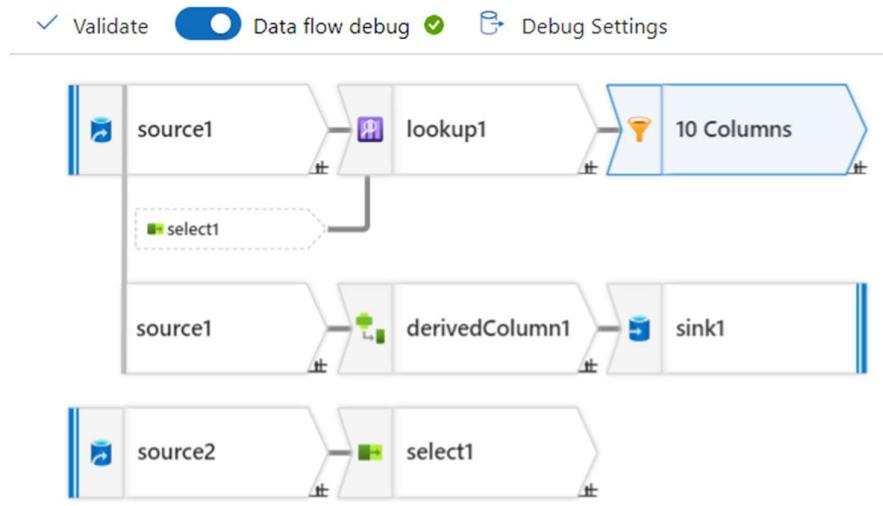
Match multiple rows  ⓘ

Lookup conditions \*

Left: source1's column	Right: select1's column	
123 empid	= =	123 SQL_empid

+ [Add condition](#)

44. Add filter transformation.



45. Under settings, in the filter give the below expression.

Exp: !isNull(SQL\_empid)

Filter settings   Optimize   Inspect   Data preview ●

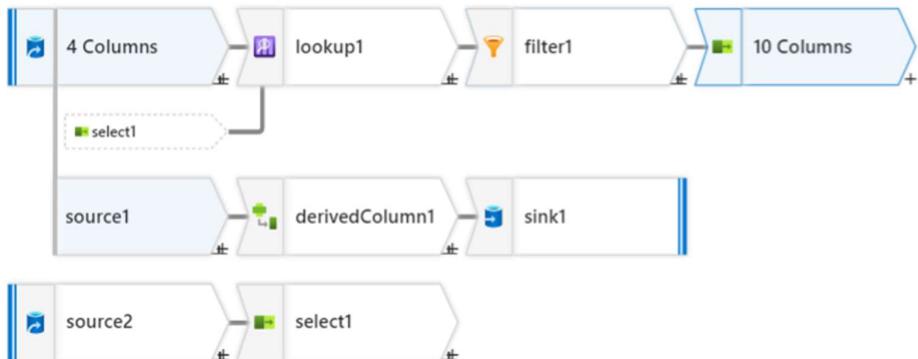
Output stream name \*  Learn more [🔗](#)

Description  [Reset](#)

Incoming stream \*

Filter on \*  [X](#)

46. Add select transformation.



47. Under settings, remove all the source rows and keep only the below rows.

Select settings    Optimize    Inspect    Data preview

Incoming stream \* filter1

Options

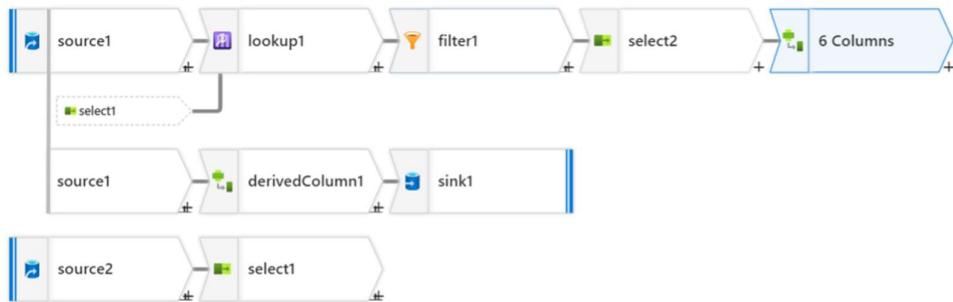
- Skip duplicate input columns
- Skip duplicate output columns

Input columns \*

Auto mapping     Reset    + Add mapping     Delete    6 mappings: 4 column(s) from the inputs le

filter1's column	Name as
123 SQL_surrKey	SQL_surrKey
123 SQL.empid	SQL.empid
abc SQL_empname	SQL_empname
abc SQL_gender	SQL_gender
abc SQL_country	SQL_country
123 SQL_isActive	SQL_isActive

48. Add derived column transformation.



49. Under settings, select the below column and give the expression as 0.

Derived column's settings    Optimize    Inspect    Data preview

Output stream name \* derivedColumn2

Description

Creating/updating the columns  
'SQL\_surrKey, SQL.empid, SQL\_empname,  
SQL\_gender, SQL\_country, SQL\_isActive'

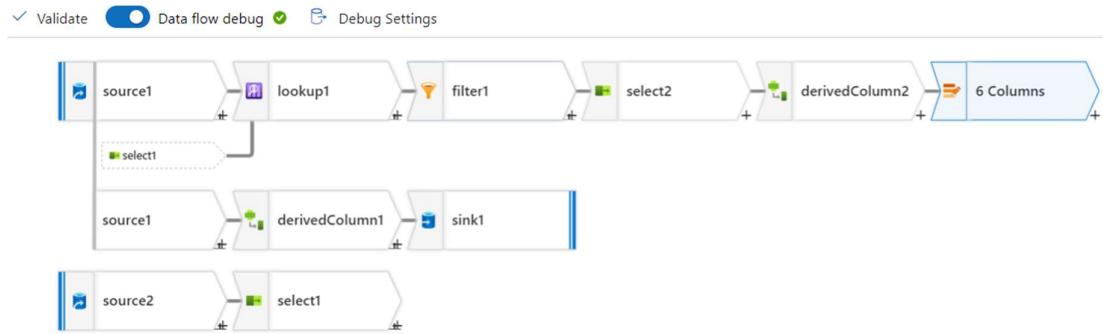
Incoming stream \* select2

+ Add     Clone     Delete     Open expression builder

Columns \*

Column	Expression
SQL_isActive	0

## 50. Add Alter Row transformation.



## 51. Under settings, give the below alter row condition.

Exp:  $1 == 1$

Alter row settings   Optimize   Inspect   Data preview ●

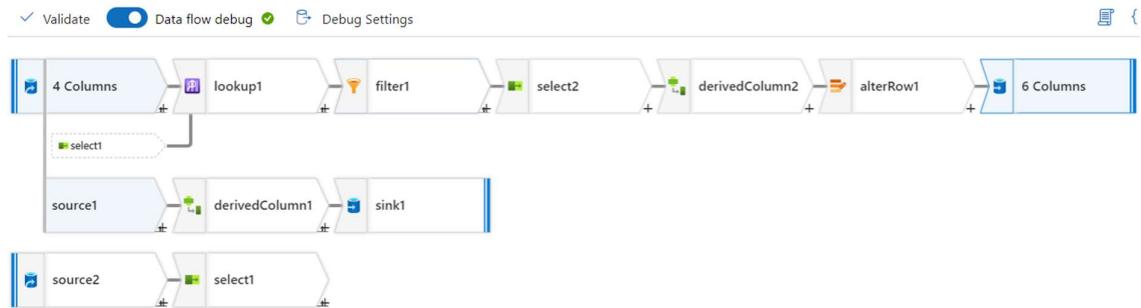
Output stream name \*  [Learn more](#)

Description  [Reset](#)

Incoming stream \*

Alter row conditions \* [Update if](#)  [x](#) [+](#) [Delete](#)

## 52. Add Sink transformation.



## 53. Under sink select the target dataset.

Sink   [Settings](#)   [Errors](#)   [Mapping](#)   [Optimize](#)   [Inspect](#)   [Data preview](#) ●

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

Description  [Reset](#)

Incoming stream \*

Sink type \*  Dataset  Inline  Cache

Dataset \*  [Test connection](#) [Open](#) [New](#)

Options  Allow schema drift [?](#)

54. Under settings, select the Update method and Key column.

Sink   Settings   Errors   Mapping   Optimize   Inspect   Data preview

Update method ⓘ

Allow insert  
 Allow delete  
 Allow upsert  
 Allow update

Key columns \* ⓘ

List of columns    Custom expression ⓘ

123 SQL\_surrKey

Add dynamic content [Alt+Shift+D]

55. Under mapping, map the fields as shown below.

Sink   Settings   Errors   **Mapping**   Optimize   Inspect   Data preview

Options

Skip duplicate input columns ⓘ  
 Skip duplicate output columns ⓘ  
 Auto mapping ⓘ    Reset        

Input columns	Output columns
123 SQL_surrKey	123 surrKey
123 SQL_emplid	123 empid
abc SQL_empname	abc empname
abc SQL_gender	abc gender
abc SQL_country	abc country
123 SQL_isActive	123 isActive

56. Just click on Empty space and in the settings set the below properties.

57. This means sink2 run 1<sup>st</sup> and sink 2 run 2<sup>nd</sup>.

Parameters   **Settings**

Custom sink ordering ⓘ

Sink name   Write order

Sink Group

sink1   2

sink2   1

58. Create a Pipeline. Drag and drop the Data flow.

Activities

Da

Move and transform

Copy data

Data flow

✓ Validate   ▶ Debug   ⚡ Add trigger    D...

Data flow

Data flow1

59. Select the Dataflow.

General    **Settings**    Parameters    User properties

Data flow \* SCD2\_dataflow

Run on (Azure IR) \* AutoResolveIntegrationRuntime

Compute size \* Small



60. Validate the pipeline and click on debug.

61. Here our pipeline was executed successfully.



Parameters    Variables    Settings    **Output**

Pipeline run ID: 471e81a5-a0b5-48d4-ad52-c14b4cc79fc9    ⏪    ⏴    Pipeline status  Succeeded [View debug run logs](#)

62. Go to the Target table and check the data.

63. Active 0 means it is old data. Active 1 means it is new data.

```
10  
11  select * from tbl_SCD2
```

Results    Messages

Search to filter items...

surrKey	empid	empname	gender	country	isActive
1	1001	maheer	male	India	0
2	1001	maheer	male2	Canada	1
3	1002	Ankitha	female	India	1
4	1003	Mike	male	USA	1