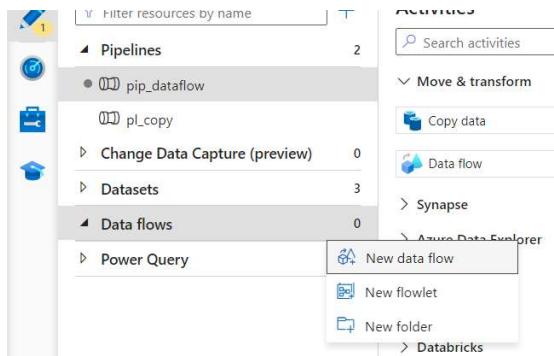


Context

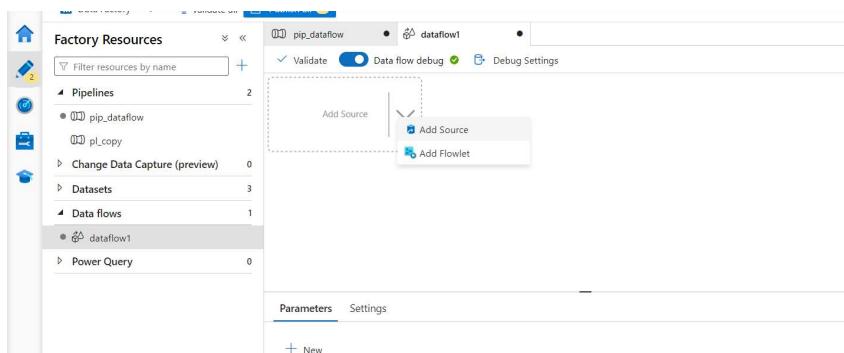
Mapping Data Flow	2 – 5
Select Transformation	6 – 10
Conditional Split Transformation	11 – 15
Derived Column Transformation	15 – 19
Union Transformation	19 – 24
Sort Transformation	25 – 27
Surrogate Key Transformation	28 – 32
Filter Transformation	33 – 36
Aggregator Transformation	36 – 39
Join Transformation	40 – 44
Lookup Transformation	45 – 48
Pipeline Using REST	49 – 55
Parameterize	55 – 60
Stringify Transformation	60 – 64
Pivot Transformation	64 – 68
Unpivot Transformation	69 – 72
Window Transformation	73 – 77
Alter Row Transformation	77 – 82
Cast Transformation	83 – 87
Web Page Usage	88 – 94
Rest Pagination	95 – 98
Rank Transformation	98 – 102
Parse Transformation	102 – 107
Assert Transformation	108 - 114

Mapping Data Flow

1. In this example we are going to create a Data Flow. To load the SQL Server table into Cloud Storage.
2. Go to Data Factory, go to Author, right-click on Data Flow, and click on New data flow.



3. Switch on the data flow debug. Click on the drop-down and click on Add Source.



4. In the Source Setting give the Name and select the SqlServer dataset that we created in the previous session and check the below properties as shown below.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * source1 Learn more

Description Import data from SqlServer_src Reset

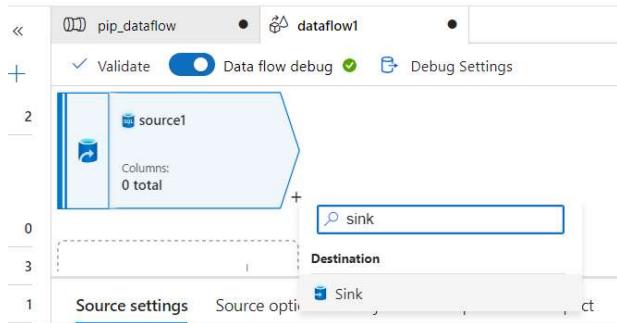
Source type * Dataset Inline

Dataset * SqlServer_src Test connection Open New

Options Allow schema drift
 Infer drifted column types
 Validate schema

Sampling * Enable Disable

5. Now click on the Plus symbol and click on Sink as shown below.

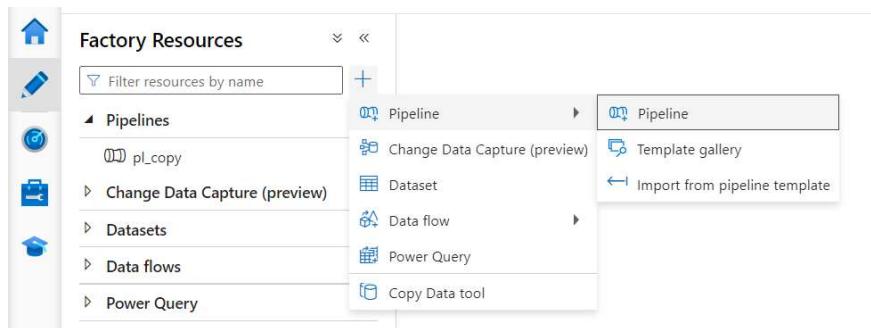


6. Now in the sink, give the name and select the sink dataset that we created in the previous session and check the below properties as shown below.

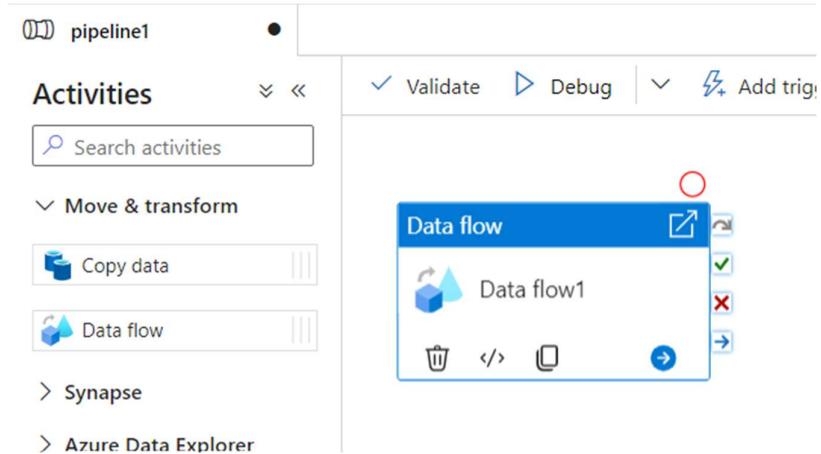
This screenshot shows the 'Sink' properties dialog. The tabs at the top are 'Sink', 'Settings', 'Errors', 'Mapping', 'Optimize', 'Inspect', and 'Data preview'. The 'Sink' tab is selected. The configuration includes:

- Output stream name ***: sink1
- Description**: Export data to ds_sink_txt
- Incoming stream ***: source1
- Sink type ***: Dataset (selected)
- Dataset ***: ds_sink_txt (selected)
- Options**:
 - Allow schema drift
 - Validate schema

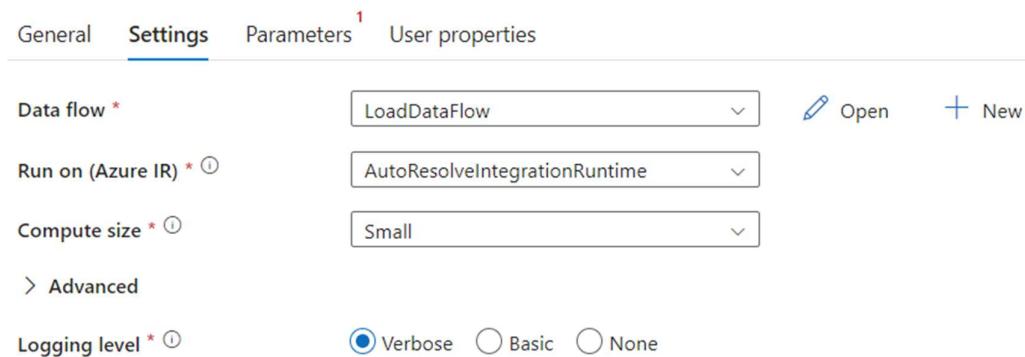
7. Create a Pipeline



8. Next Under Activity Drag and drop the Data Flow as shown below.

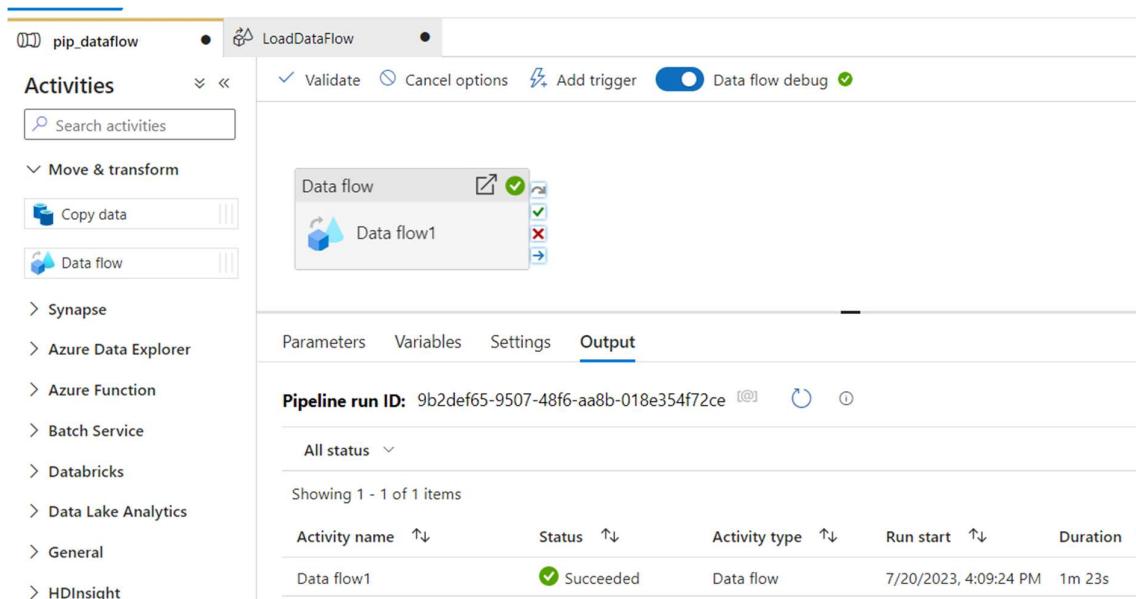


9. Under settings select the Data flow that we created.



10. Next validate the Pipeline and click on Debug.

11. Our pipeline is Successfully executed.



12. Now go to Storage account and the container folder. We have one output folder and under that, we have a new file download.

The screenshot shows the Microsoft Azure Storage Explorer interface. A blob container named 'f-demo' is selected. Inside it, there is a subfolder named 'Output' which contains several files. One file, 'part-00000-e6862a16-47ef-4712-a2dc-d150acf86f...', has a context menu open over it. The 'Download' option in the menu is highlighted.

Name	Modified	Access tier	Archive status	Blob type	Size
[...]					
SUCCESS	7/20/2023, 4:09:54 PM	Hot (Inferred)		Block blob	0 B
<input checked="" type="checkbox"/> part-00000-e6862a16-47ef-4712-a2dc-d150acf86f...	7/20/2023, 4:09:54 PM	Hot (Inferred)		Block blob	564 B
Std_details.txt	7/20/2023, 12:16:29 ...	Hot (Inferred)		Block blob	166 B

13. Open it you will see our TotalSale table.

A1	A	B	C	D	E	F	G	H
1	id	SalePerson	SalePerson	ProductNa	ItemsSold	SoldPrice	Country	Region
2	1	Aamir	Shahzad	TV	1	700	USA	North
3	2	M	Raza	Cell Phone	2	800	USA	North America
4	3	Christy	Ladson	TV	3	1600	USA	North
5	4	John	Rivers	Laptop	5	2400	USA	North
6	5	Najaf	Ali	Computer	1	300	Pakistan	Asia
7	6	Sukhjeet	Singh	TV	2	900	India	Asia
8	7	Chirag	Patel	Cell Phone	5	1500	India	Asia
9	8	Aleena	Aman	Laptop	2	800	Pakistan	Asia
10	9	Petra	Henry	TV	10	5000	France	Europe
11	10	Rita	Roger	Laptop	7	2100	France	Europe
12	11	Tamara	Tony	Cell Phone	2	1200	Germany	Europe

Select Transformation

1. In this example we are going to use the below file. I have uploaded the file for the local system to my source connection.

The screenshot shows the Azure Storage Blob view for the file 'Employee.csv'. The file contains the following data:

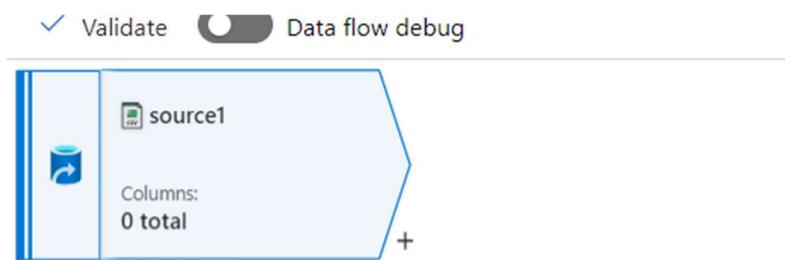
empid	name	country	department
1	maheer	india	2
2	asmin	india	2
3	wafa	india	2
4	Sarfaraj	india	2
5	Ayesha	india	1
6	Pyarijaan	india	1
7	Mahaboob	india	1
8	Arfan	india	1
9	shabbir	india	1

Edit

2. Create a Data flow.

The screenshot shows the Microsoft Azure Data Factory interface under 'Factory Resources'. The 'Data flow' item is selected and highlighted.

3. Add Data source.



4. Under source select the source connection and click on open.

Source settings Source options Projection Optimize Inspect Data preview

Description Import data from ds_src_txt Reset

Source type * Dataset Inline

Dataset * ds_src_txt Test connection Open New

Options Allow schema drift

5. Select the Employee file as shown below.

Connection Schema Parameters

Linked service * AzureBlobStorage_txt Test connection Edit New Learn more

File path * f-demo / Directory / Employee.txt Browse

Compression type Select...

Column delimiter Comma (,)

6. Now under schema click on the Import schema and click on From connection/store.

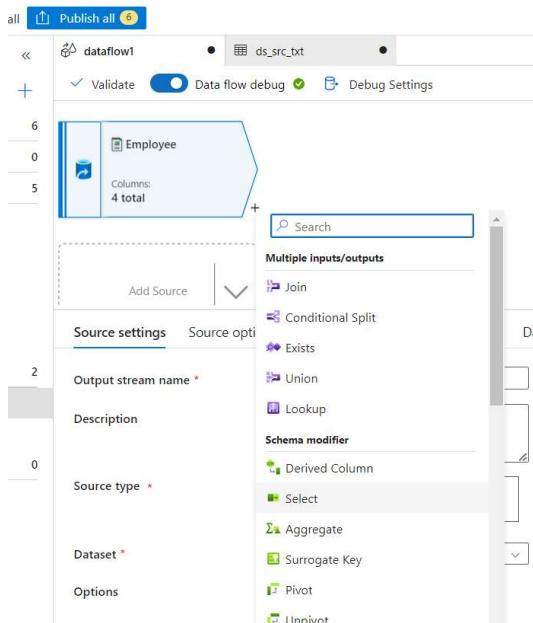
Connection Schema Parameters

Import schema Clear

From sample file

From connection/store emp1

7. Now go to Data flow click on the plus symbol and click on the select option as shown below.



- Under the setting we have 4 columns but I deleted the department column by clicking on the delete icon button.
- And I rearrange the order but click on the drop-down from the Employee's column and change the column names also as shown below.

Select settings Optimize Inspect Data preview ●

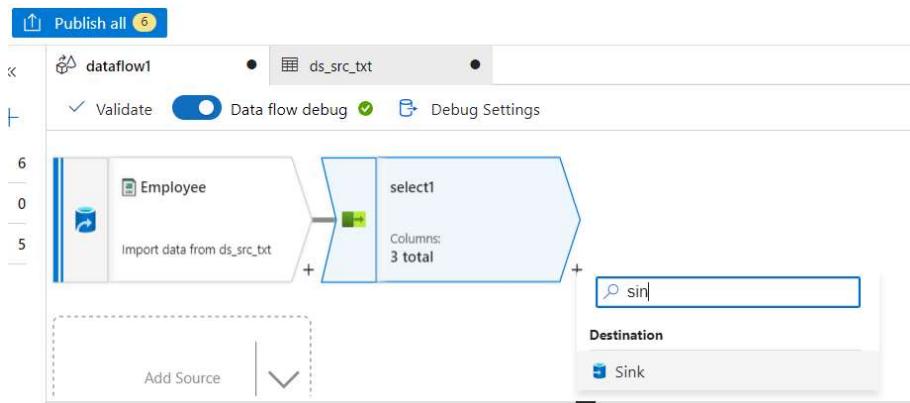
Options Skip duplicate input columns ⓘ Skip duplicate output columns ⓘ

Input columns *

Auto mapping ⓘ Reset Add mapping Delete 3 mappings: 1 column(s) from the inputs left

Employee's column	Name as
abc name	Emp_Name
abc empid	Emp_Id
abc country	Emp_Country

- Now click on the select plus symbol and click on Sink as shown below.



- Under the sink select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

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

Description Export data to ds_sink_txt

Incoming stream * select1

Sink type *

Dataset	Inline	Cache
---------	--------	-------

Dataset * ds_sink_txt Test connection Open New

12. Under settings select the File name option and give the file name 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. The re current partitioning.

Clear the folder

File name option *

Output to single file * ⓘ

13. Create a pipeline.

Microsoft Azure | Data Factory | AzureDataFactoryTraining | Search factory and doc

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data

Data Factory Validate all Publish all

Factory Resources Filter resources by name +

Pipelines

- If_Pipeline
- pip_dataflow
- switch_pipeline
- pl_copy
- Until_Pipe

Pipeline

- Change Data Capture (preview)
- Dataset
- Data flow
- Power Query
- Copy Data tool

Template gallery Import from pipeline template

14. Drag and drop the Data flow activity.

dataflow1 ds_src_txt ds_sink_txt pipeline1

Activities Data

Move & transform

Copy data

Data flow Data flow1

15. Under the settings select the dataflow.

General **Settings** Parameters User properties

Data flow * Open New

Run on (Azure IR) * ⓘ

Compute size * ⓘ

16. Next validate the pipeline and click on Debug.

17. Here our pipeline is executed successfully.

The screenshot shows the Azure Data Flow pipeline execution interface. At the top, there are buttons for Validate, Debug, Add trigger, and a toggle switch for Data flow debug. Below this, the pipeline name 'Data flow1' is shown with a green checkmark indicating success. The main area displays the 'Output' tab, showing the Pipeline run ID: b9e626f8-8837-44c5-84f1-65726d7844b0. A table below lists the activity details:

Activity name	Activity status	Activity type	Run
Data flow1	Succeeded	Data flow	7/2'

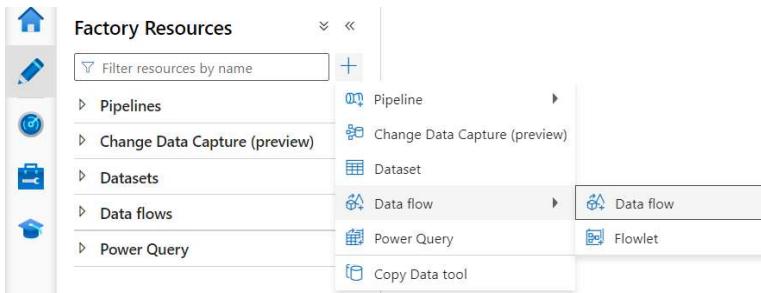
18. Now go to a destination and check the file and the data.

The screenshot shows the Azure Blob storage interface for the 'Output' container. On the left, there's a list of blobs, including '_SUCCESS', 'SelectEmployee.csv', and 'Std_details.txt'. On the right, the 'SelectEmployee.csv' blob is displayed as a table:

Emp_Name	Emp_Id	Emp_Country
maheer	1	india
asmin	2	india
wafa	3	india
Sarfaraj	4	india
Ayesha	5	india
Pyarjaan	6	india
Mahaboob	7	india
Arfan	8	india
shabbir	9	india

Conditional Split Transformation

1. For this example, also we are using the same employee table.
2. Go to Azure Data Factory, and create a data flow.



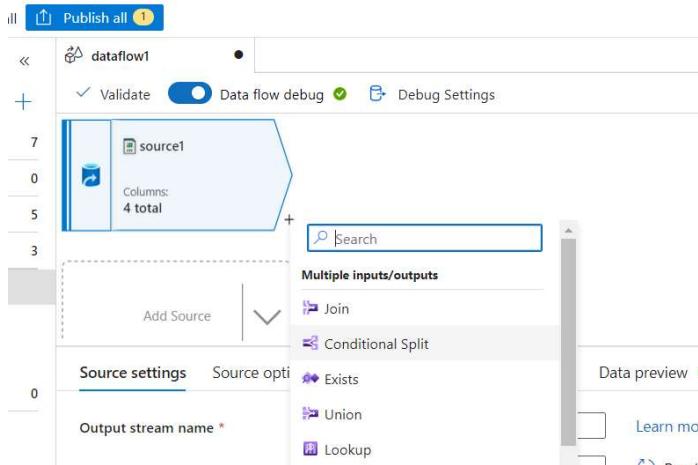
3. Select the source and employee table as shown below.

A screenshot of the 'Source settings' configuration page. At the top, there are tabs: Source settings (which is selected and highlighted in blue), Source options, Projection, Optimize, Inspect, and Data preview. Below the tabs, there are several input fields:

- Output stream name *: A text input field containing 'source1'.
- Description: A text input field containing 'Import data from ds_src_txt'.
- Source type *: A button group with two options: 'Dataset' (selected) and 'Inline'.
- Dataset *: A dropdown menu showing 'ds_src_txt'.

At the bottom right of the form, there are buttons for 'Test connection' (with a blue icon), 'Open' (with a pencil icon), and 'New' (with a plus sign icon).

4. Click on the source plus symbol and click on a conditional split as shown below.



- Under the split condition add the three conditions as shown below.
- Delete the default condition.

Conditional split settings Optimize Inspect Data preview

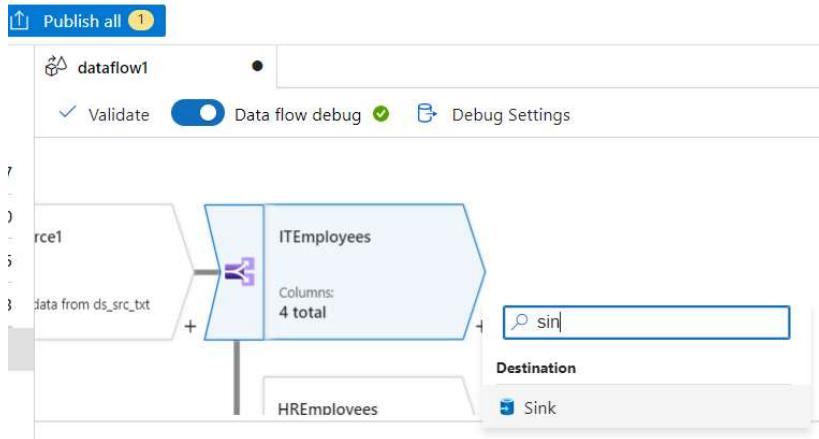
Description: Conditionally distributing the data in department groups, based on columns '[1]'

Incoming stream: source1

Split on: First matching condition

Stream names	Condition
ITEmployees	equals(department,'1')
HREmployees	equals(department,'2')
PayrollEmployees	equals(department,'3')

- Click on the IT Employee plus symbol and click on the sink.



- Under the sink select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview

Info: This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The recommended setting is to use a multi-partitioned sink.

Output stream name: sink1

Description: Export data to ds_sink_txt

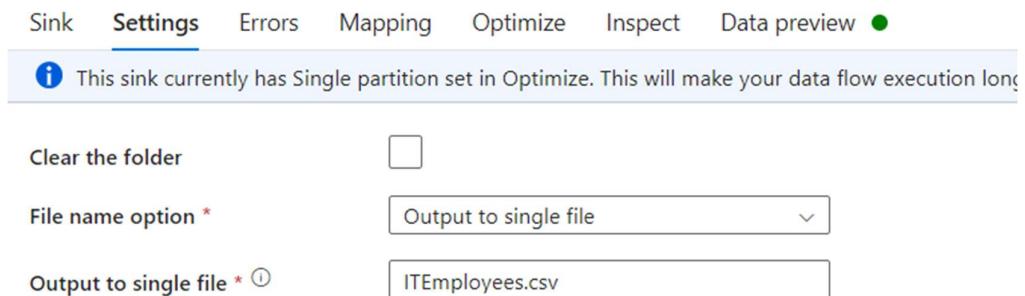
Incoming stream: split1@ITEmployees

Sink type: Dataset

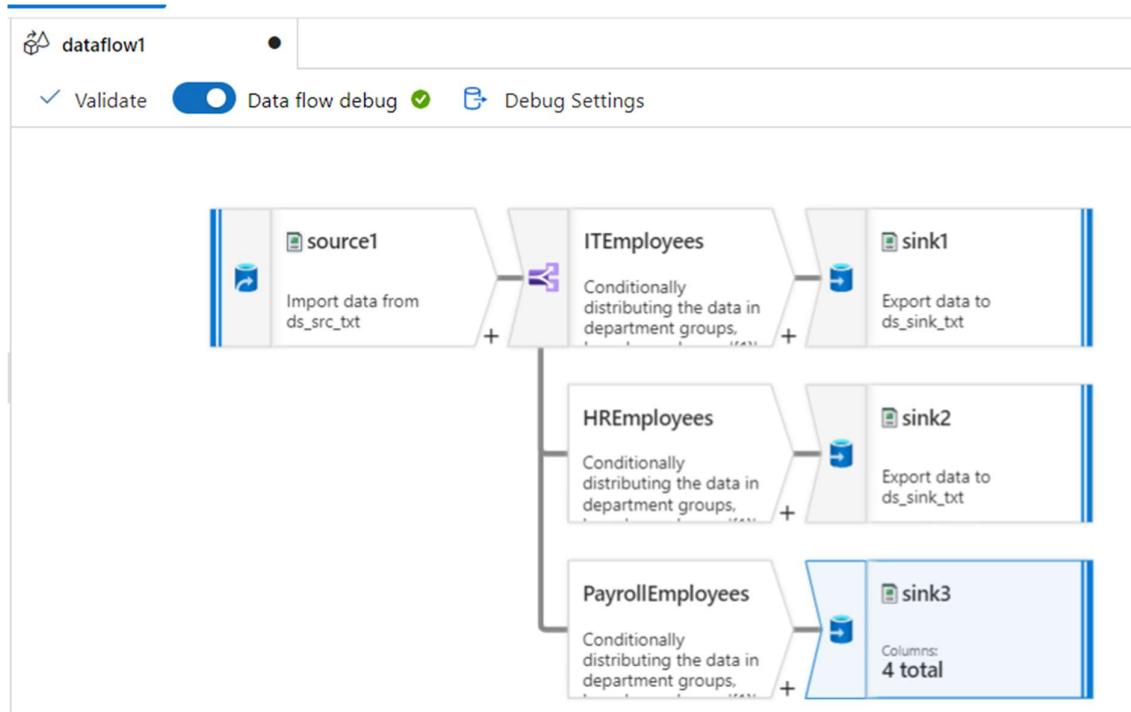
Dataset: ds_sink_txt

Test connection Open New

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



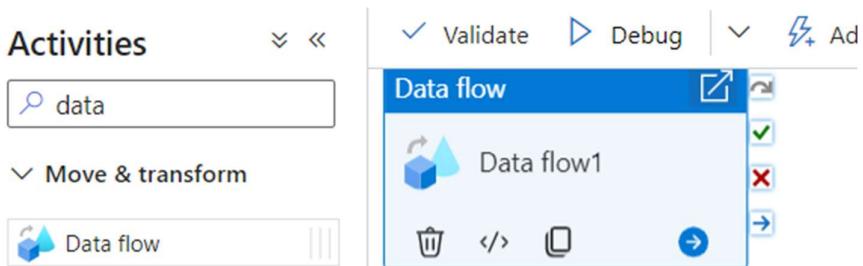
10. Similarly add the sink for other conditions also as shown below.



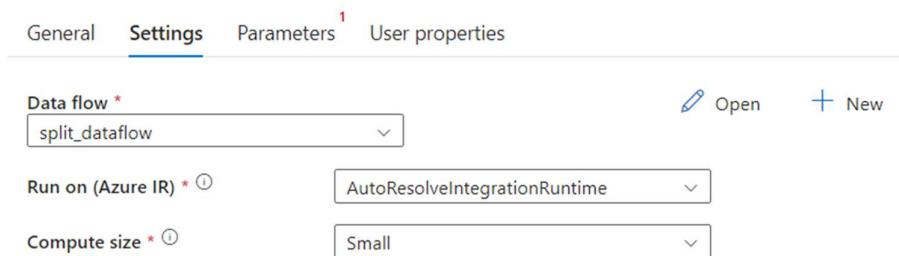
11. Create a pipeline.

The screenshot shows the Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists 'Pipelines', 'Change Data Capture (preview)', 'Datasets', and 'Data flows'. The 'Data flows' section is expanded, showing a pipeline named 'split_dataflow'. The main workspace shows the 'split_dataflow' pipeline with its components: 'source1', 'ITEmployees', 'sink1', 'HREmployees', 'sink2', 'PayrollEmployees', and 'sink3'. The pipeline is currently in 'Data flow debug' mode. At the top, there are buttons for 'Validate all' and 'Publish all'.

12. Drag and drop the Data Flow activity.



13. Under the setting select the data flow.



14. Now validate the pipeline and click on debug.

15. Our pipeline was executed successfully.

A screenshot of the 'Output' tab for a pipeline run. The 'Pipeline run ID' is listed as '93045966-d7e5-462f-b32a-b7a5e3d9592c'. Below it, a status summary shows 'All status' with 'Showing 1 - 1 of 1 items'. A table lists the activity details: 'Activity name' is 'Data flow1', 'Activity status' is 'Succeeded' (indicated by a green checkmark), 'Activity type' is 'Data flow', and 'Run start' is '7/21/202'.

Activity name	Activity status	Activity type	Run start
Data flow1	Succeeded	Data flow	7/21/202

16. Now go to a destination and check whether the files are created or not.

Name	Modified
<input type="checkbox"/> _SUCCESS	7/20/2023, 4:09:54 PM
<input checked="" type="checkbox"/> HREmployees.csv	7/21/2023, 11:50:15 AM
<input checked="" type="checkbox"/> ITEmployees.csv	7/21/2023, 11:50:18 AM
<input type="checkbox"/> part-00000-e6862a16-47ef-4712-a2dc-d150acfb6f...	7/20/2023, 4:09:54 PM
<input checked="" type="checkbox"/> PayrollEmployees.csv	7/21/2023, 11:50:17 AM
<input type="checkbox"/> SelectEmployee.csv	7/21/2023, 11:23:17 AM

Derived Column Transformation

1. In this example also we are using the same Employee file but here we are editing this file.
2. Here I have added two more columns and I didn't give the country name for the new column.
3. Change on your side also and click on save.

The screenshot shows the Azure Blob Storage interface for the 'Employee.csv' file. The file content is displayed as follows:

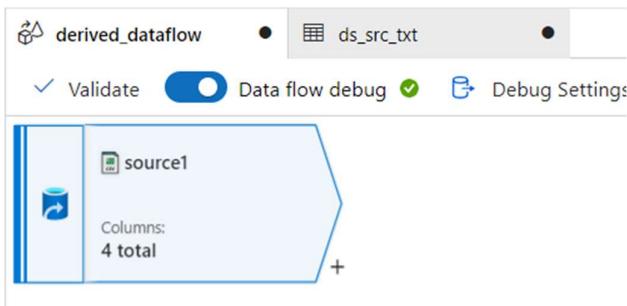
```
1 empid,name,country,department
2 1,maheer,india,2
3 2,asmin,india,2
4 3,wafa,india,2
5 4,Sarfaraj,india,3
6 5,Ayesha,india,3
7 6,Pyarijaan,india,1
8 7,Mahaboob,india,1
9 8,Arfan,india,1
10 9,shabbir,india,1
11 10,Afrin,,3
12 11,Shahin,,2
```

Below the preview, there is a 'Csv' dropdown and a 'Preview' button.

4. Create a Data flow.

The screenshot shows the 'Factory Resources' blade in the Azure Data Factory interface. The 'Data flows' item is highlighted with a red box.

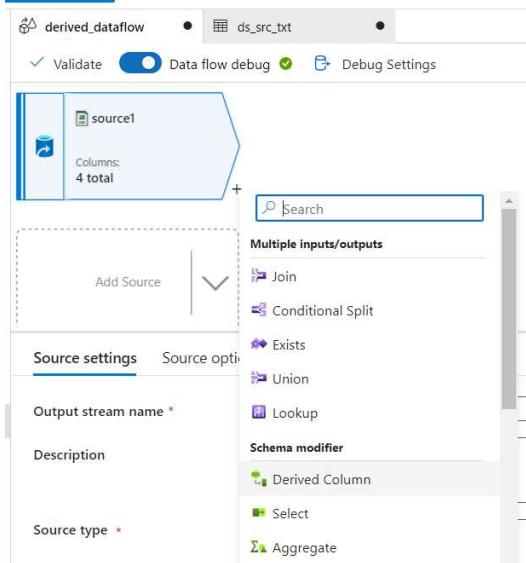
5. Get the Source.



6. Select the dataset and employee file as shown below.

The screenshot shows the 'Source settings' tab for 'source1'. It includes fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Import data from ds_src_txt'), and 'Source type' (set to 'Dataset'). Under 'Dataset', the dropdown is set to 'ds_src_txt'. There are also buttons for 'Test connection', 'Open', and 'New'. The 'Allow schema drift' option is checked.

7. Click on the Source plus symbol and click on the Derived column as shown below.



8. Under Columns add two columns and give the below expressions.
9. For the first column just click on the drop-down and select the country that gives the below expression

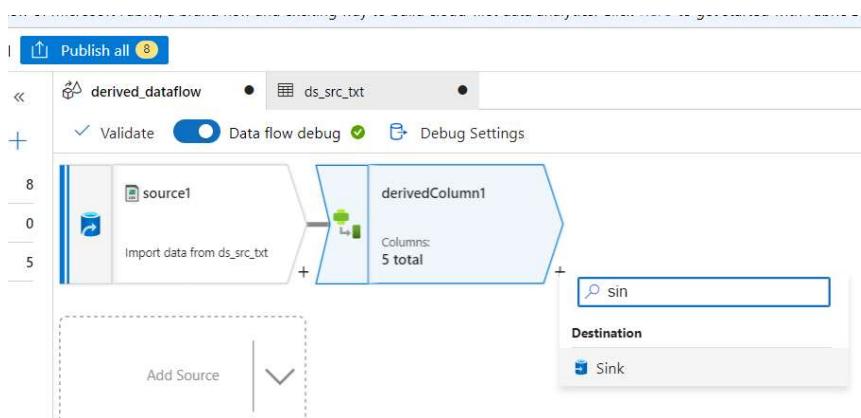
Exp: upper(country)

10. For the second column type a NewCountry and give the below expression.

Exp: iif(isNull(country), 'Unknown',upper(country))

Column	Expression
country	upper(country)
NewCountry	iif(isNull(country), 'Unknown',upper(country))

11. Now click on the plus symbol and add a sink as shown below.



12. Under the sink, select the destination dataset.

Sink	Settings	Errors	Mapping	Optimize	Inspect	Data preview
Description: Export data to ds_sink_txt						
Incoming stream: derivedColumn1						
Sink type: Dataset						
Dataset: ds_sink_txt						
Skip line count:						
Options: Allow schema drift						

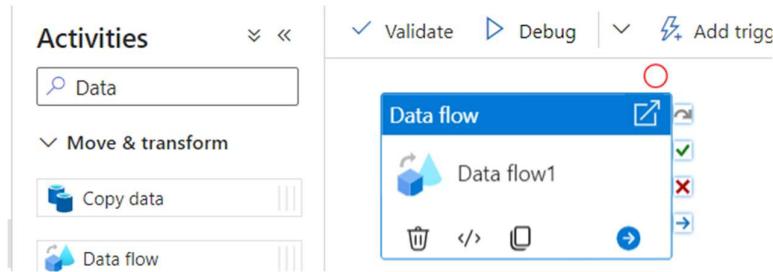
13. Under setting select the file option and file name as shown below.

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.

Clear the folder	<input type="checkbox"/>
File name option *	Output to single file
Output to single file * ⓘ	NewEmployee.csv
Quote All ⓘ	<input type="checkbox"/>
Headers ⓘ	Enter expression... ANY

14. Create a Pipeline and drag and drop the data flow activity.



15. Select the Data flow.

General **Settings** Parameters User properties

Data flow *
derived_dataflow

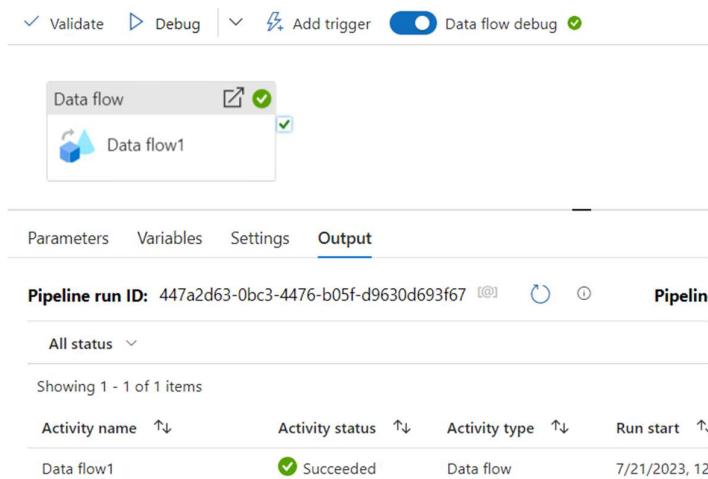
Run on (Azure IR) * ⓘ
AutoResolveIntegrationRuntime

Compute size * ⓘ
Small

Open + N

16. Now validate the data flow and click on debug.

17. Here our pipeline was executed successfully.



18. Now go to a destination and check the data.

The screenshot shows the Azure Blob storage interface for the file 'Output/NewEmployee.csv'. The left sidebar displays a list of files including '_SUCCESS', 'HREmployees.csv', 'ITEmployees.csv', 'NewEmployee.csv', 'part-00000-e6862a16-47ef-47...', 'PayrollEmployees.csv', 'SelectEmployee.csv', and 'Std_details.txt'. The main area shows the contents of 'NewEmployee.csv' in a table format:

empid	name	country	department	NewCountry
1	maheer	INDIA	2	INDIA
2	asmin	INDIA	2	INDIA
3	wafa	INDIA	2	INDIA
4	Sarfaraj	INDIA	3	INDIA
5	Ayesha	INDIA	3	INDIA
6	Pyarijan	INDIA	1	INDIA
7	Mahaboob	INDIA	1	INDIA
8	Arfan	INDIA	1	INDIA
9	shabbir	INDIA	1	INDIA
10	Afrin		3	Unknown
11	Shahin		2	Unknown

Buttons at the bottom include 'Edit' and 'Generate SAS'.

Union Transformation

1. In this example we are using the Output of conditional split transformation.
2. These outputs are from Conditional split transformation.
3. Now In our example there are the source data we are going to use a union to combine and stored in one file.

The screenshot shows a list of files in Azure Data Explorer. The files are:

<input type="checkbox"/>	_SUCCESS	7/20/2023, 4:09:54 PM	Hot (Inferred)
<input checked="" type="checkbox"/>	HREmployees.csv	7/21/2023, 11:50:15 ...	Hot (Inferred)
<input checked="" type="checkbox"/>	ITEmployees.csv	7/21/2023, 11:50:18 ...	Hot (Inferred)
<input type="checkbox"/>	NewEmployee.csv	7/21/2023, 12:16:23 ...	Hot (Inferred)
<input type="checkbox"/>	part-00000-e6862a16-47ef-4712-a2dc-d150acfb6f...	7/20/2023, 4:09:54 PM	Hot (Inferred)
<input checked="" type="checkbox"/>	PayrollEmployees.csv	7/21/2023, 11:50:17 ...	Hot (Inferred)
<input type="checkbox"/>	SelectEmployee.csv	7/21/2023, 11:23:17 ...	Hot (Inferred)

4. We need to create three datasets for these three files.

The screenshot shows the Azure Data Factory interface for creating a new dataset. The 'Datasets' folder contains five datasets. A context menu is open over 'Data flows', with options 'New dataset' and 'New folder' highlighted.

- Select the Azure Blob Storage and click on continue then select the delimited text and click on continue.
- Then give the name, select the linked service, and select the file as shown below.

Set properties

Name
ITEmployeeDataset

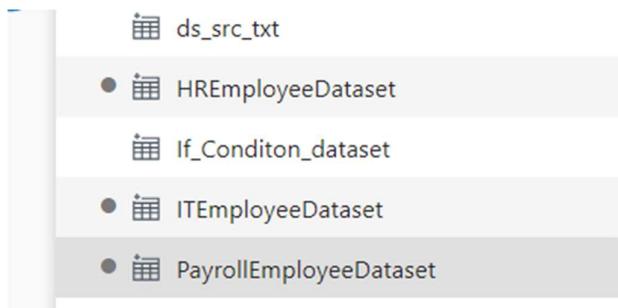
Linked service *
AzureBlobStorage_txt

File path
f-demo / Output / ITEmployees.csv

First row as header

Import schema
 From connection/store From sample file None

- Similarly create datasets for the other two files also.



- Create a Data flow.

Microsoft Azure | Data Factory | AzureDataFactoryTraining | Search factory and c

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data solutions.

Data Factory | Validate all | Publish all

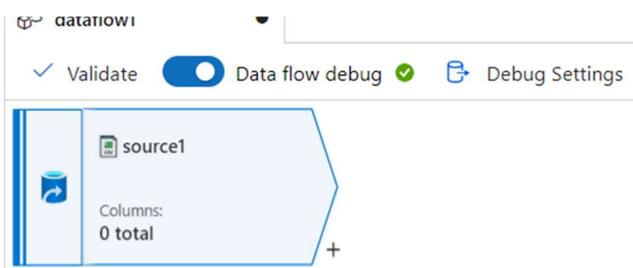
Factory Resources

Filter resources by name +

- Pipelines
- Change Data Capture (preview)
- Datasets
- Data flows**
- Power Query
- Copy Data tool

The 'Data flows' item is highlighted in the list.

9. Add a Source.



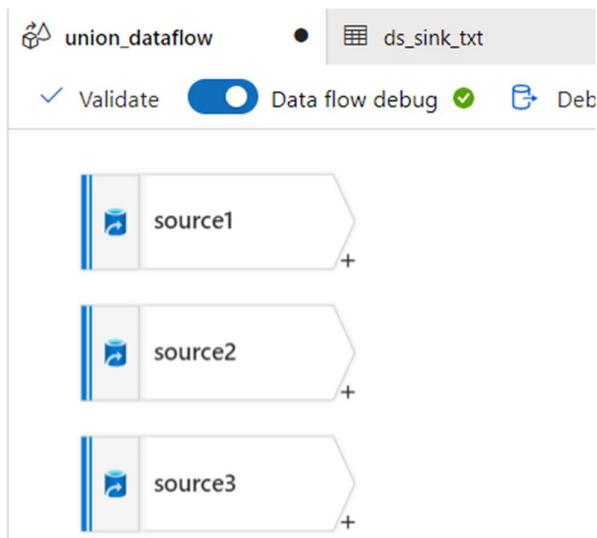
10. Under source select the destination dataset because our tables are present in destination and click on open.

This is a configuration dialog for 'source1'. It includes tabs for Source settings, Source options, Projection, Optimize, Inspect, and Data preview. The 'Source settings' tab is active. It shows the output stream name as 'source1', a description of 'Import data from ITEmployeeDataset', and two source type options: 'Dataset' (selected) and 'Inline'. Below these, a dropdown for 'Dataset' is set to 'ITEmployeeDataset', with buttons for Test connection, Open, and New. An 'Options' section contains a checked checkbox for 'Allow schema drift'.

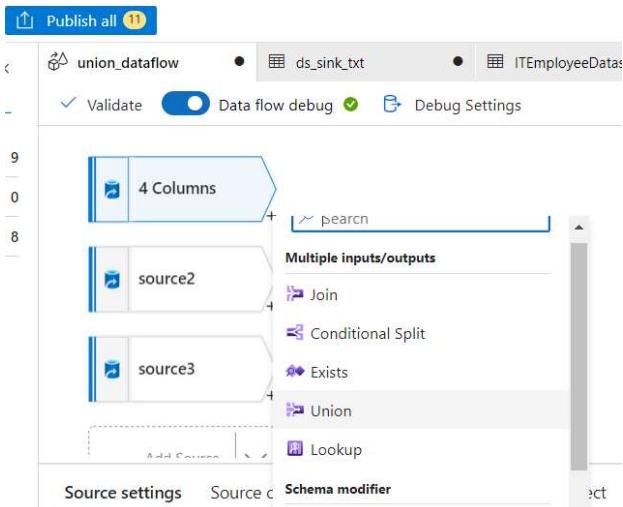
11. Under schema click on Import schema and click on From connection/store.

This screenshot shows the 'Schema' tab of the configuration dialog. It has tabs for Connection, Schema (which is selected), and Parameters. Under the Schema tab, there are buttons for 'Import schema' (which is highlighted in blue) and 'Clear'. Below these buttons is a dropdown menu with two options: 'From sample file' (selected) and 'From connection/store'.

12. Go to Data Flow and add two more data sources. Similarly configure the source file with the datasets we created for each department like the above process.



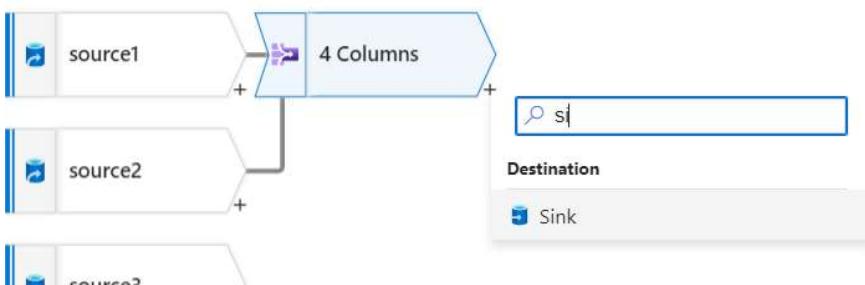
13. Now click on the plus symbol and click on the Union as shown below.



14. Under the union setting add select source 2 and source 3 as shown below.

The screenshot shows the 'Union settings' dialog. It has tabs for 'Union settings', 'Optimize', 'Inspect', and 'Data preview'. The 'Union settings' tab is active. The 'Description' field contains the text 'Combining rows from transformation 'source1, source2, source3''. The 'Incoming stream *' dropdown is set to 'source1'. The 'Union by * ⓘ' section has a radio button for 'Name' selected. The 'Union with *' section contains two entries: 'source2' and 'source3', each with a delete icon to its right. A 'Reset' button is located at the top right of the dialog.

15. Now click on the plus symbol and click on the sink.



16. Under the sink select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

Output stream name: SINK

Description: Export data to ds_sink_txt

Incoming stream: union1

Sink type: Dataset

Dataset: ds_sink_txt

Test connection Open New

17. Under setting select the file option and file name 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. The recommended current partitioning.

Clear the folder

File name option: Output to single file

Output to single file: AllEmployees.csv

18. Create a pipeline then drag and drop the data flow activity.

Activities

Data

Move & transform

Copy data

Data flow

Data flow1

Validate Debug Add trigger

19. Next select the dataflow.

General **Settings** Parameters User properties

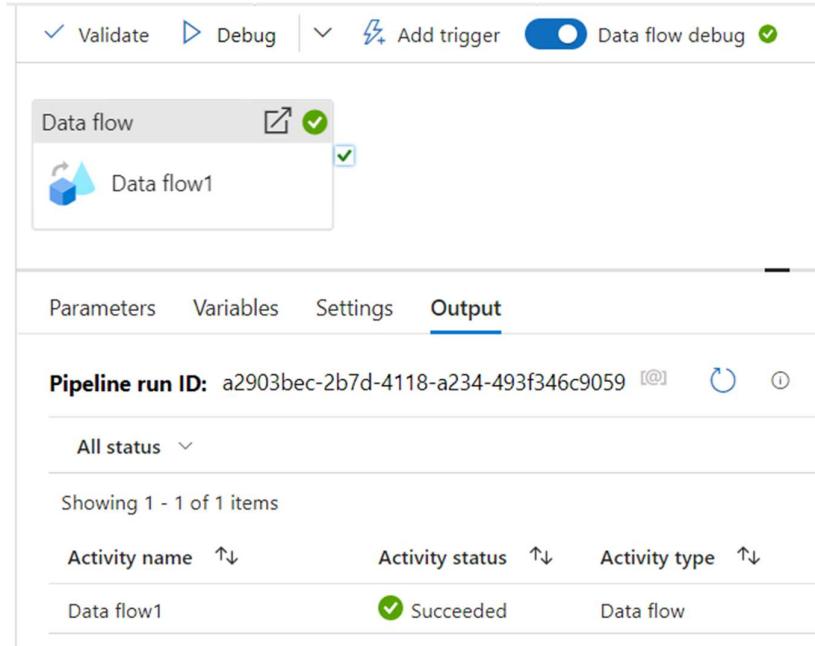
Data flow: union_dataflow

Run on (Azure IR): AutoResolveIntegrationRuntime

Compute size: Small

Open +

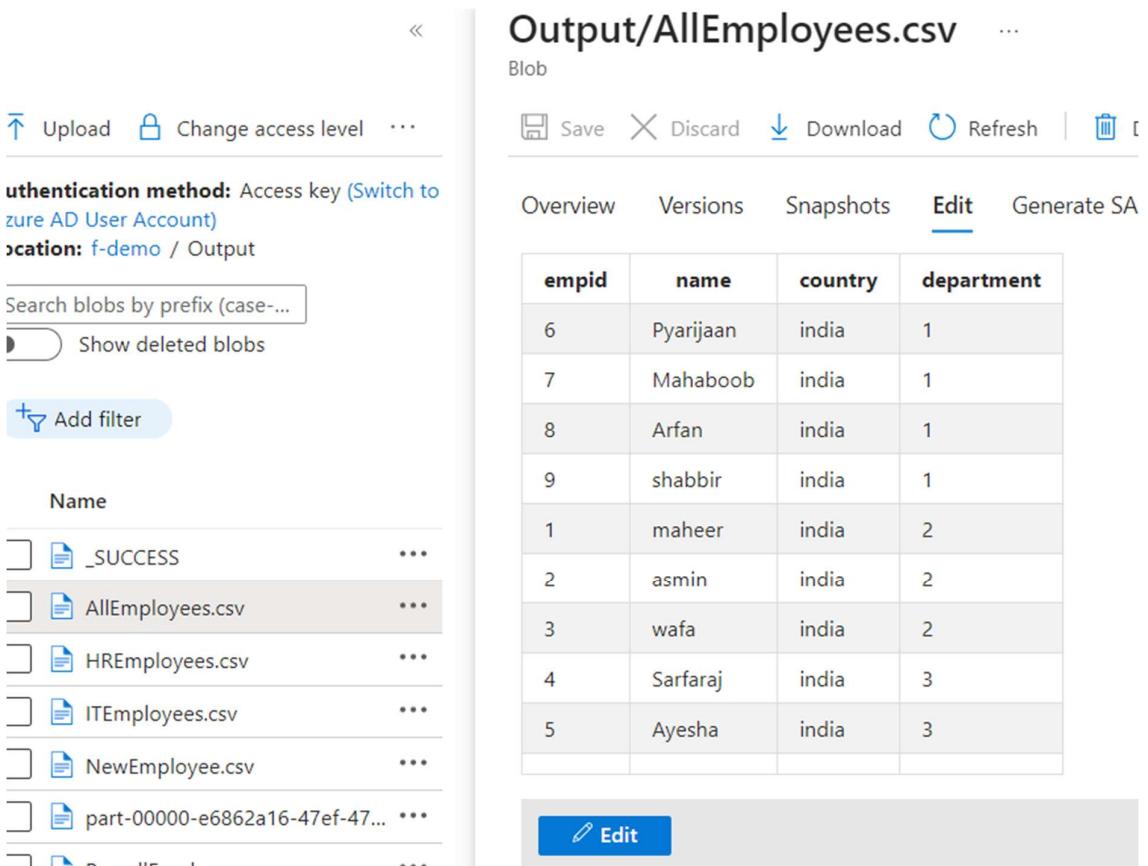
20. Validate the pipeline and click on debug.
 21. Here our pipeline was executed successfully.



The screenshot shows the 'Output' tab of the pipeline run history. The Pipeline run ID is a2903bec-2b7d-4118-a234-493f346c9059. The Data flow1 activity is listed with a status of 'Succeeded'. The table below shows the activity details:

Activity name	Activity status	Activity type
Data flow1	✓ Succeeded	Data flow

22. Next check the data in the destination.

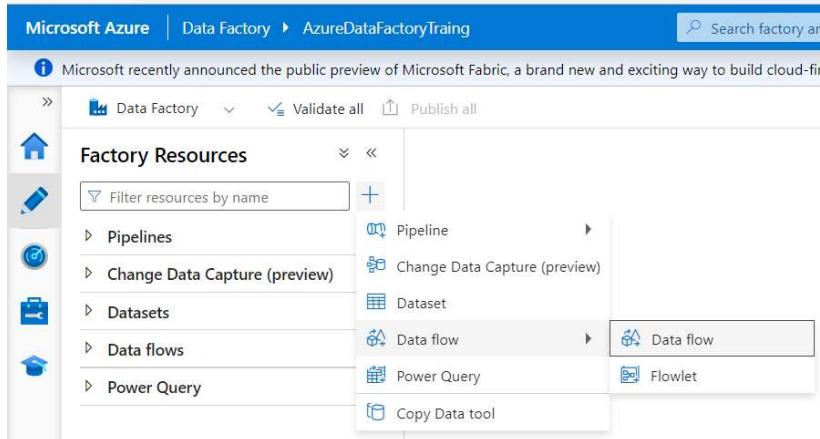


The screenshot shows the blob details for 'Output/AllEmployees.csv'. The blob contains the following data:

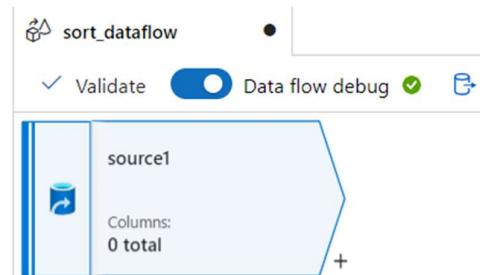
empid	name	country	department
6	Pyarijaan	india	1
7	Mahaboop	india	1
8	Arfan	india	1
9	shabbir	india	1
1	maheer	india	2
2	asmin	india	2
3	wafa	india	2
4	Sarfaraj	india	3
5	Ayesha	india	3

Sort Transformation

1. In this example we are using the Employee file. So here we are going to sort the emp name column.
2. Create a Data flow.



3. Add data source.



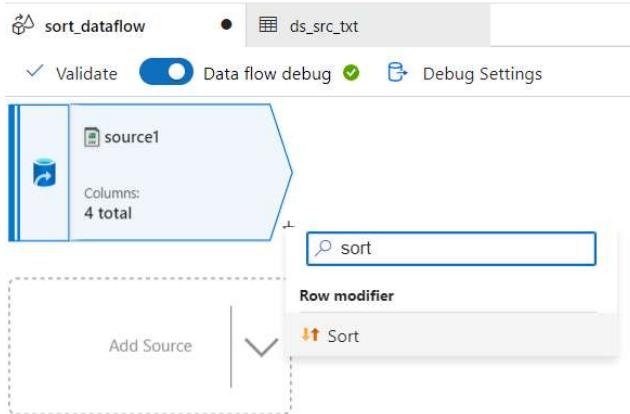
4. Under the source select the source dataset and file as shown below.

A screenshot of the 'Source settings' tab in the Azure Data Flow designer. The tab includes tabs for 'Source settings', 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview'.

- Output stream name:** source1
- Description:** Import data from ds_src_txt
- Source type:** Dataset (selected)
- Dataset:** ds_src_txt
- Options:** Allow schema drift (checked)

Other buttons include 'Test connection', 'Open', and 'New'.

5. Click on the plus symbol and click on sort as shown below.

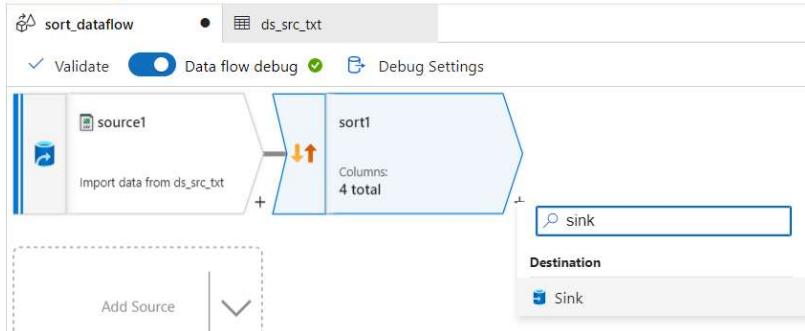


6. Under the sort setting select the name and give an Ascending order as shown below.

The 'Sort settings' configuration page shows the following details:

- Incoming stream ***: source1
- Options ***:
 - Case insensitive
 - Sort only within partition
- Sort conditions ***:
 - source1's column
 - Order:
 - abc
 - name
 - Ascending

7. Now click on the plus symbol and click on the sink.



8. Under sink select the destination dataset as shown below.

The 'Sink' configuration page shows the following details:

- Sink**: The tab is selected.
- Incoming stream ***: sort1
- Sink type ***:
 - Dataset** (selected)
 - Inline
 - Cache
- Dataset ***: ds_sink_txt
- Options**:
 - Allow schema drift

9. Under setting select the select the file option and file name 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. The current partitioning.

Clear the folder

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

Output to single file * ⓘ SortEmployee.csv

10. Create a pipeline then drag and drop the data flow activity.

Activities

Validate Debug

Data

Move & transform

Copy data

Data flow

Data flow

11. Select the Data flow.

General **Settings** Parameters 1 User properties

Data flow *

sort_dataflow

Run on (Azure IR) * ⓘ

AutoResolveIntegrationRuntime

Compute size * ⓘ

Small

Open

12. Now validate the pipeline and click on debug.

13. Here our pipeline was executed successfully.

Data flow

Data flow

Parameters Variables Settings **Output**

Pipeline run ID: f0b45b41-3d55-46cc-a38e-335e6e38443a

All status

Showing 1 - 1 of 1 items

Activity name ↑	Activity status ↑↓	Activity type ↑↓
Data flow1	✓ Succeeded	Data flow

Surrogate Key Transformation

1. In this example we are using different data files as shown below.

The screenshot shows the Azure Blob storage interface for the file 'SurrogateKeyEmp.csv'. The file details pane on the right displays the following information:

- Authentication method:** Access key (Switch to Azure AD User Account)
- Location:** f-demo
- File Content:** A table with columns Name, Country, and DepID, containing three rows: maheer (india, 2), asmin (india, 1), and wafa (india, 3).
- Actions:** Save, Discard, Download, Refresh, Generate SAS.

The left sidebar lists other files in the directory:

- Output
- Output1
- Switch1
- Department.csv
- Employee.csv
- Output
- Std_details.txt
- SurrogateKeyEmp.csv

2. In the Azure Data Factory create a Data set for this file.

Set properties

Name
SurrogatekeyDataset

Linked service *
AzureBlobStorage_txt

File path
f-demo / Directory / SurrogateKeyEmp.csv

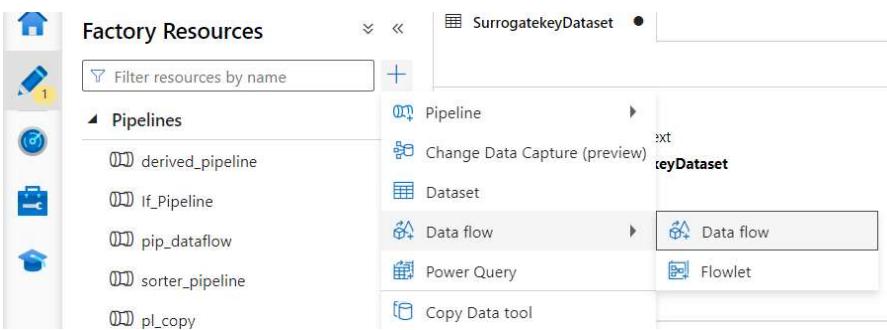
First row as header

Import schema
 From connection/store From sample file None

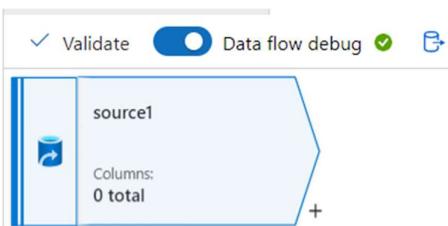
3. Next under schema. Click on Import schema and click on From connection/store.

The screenshot shows the 'Schema' tab of the dataset configuration. Under the 'Import schema' section, the 'From connection/store' option is selected. The 'Type' dropdown is visible.

4. Create a data flow.



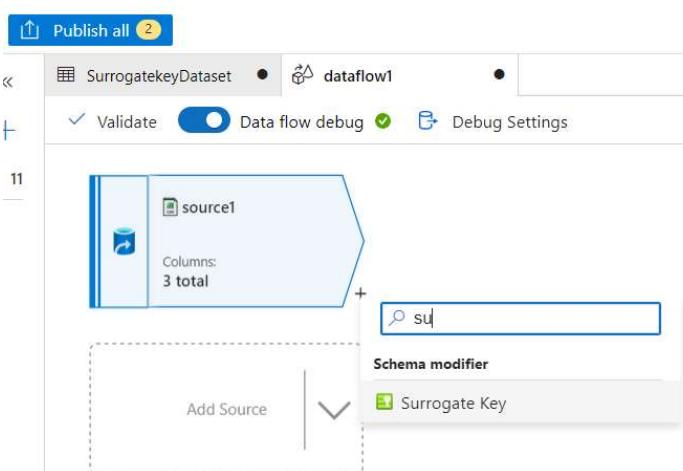
5. Add a data source.



6. Select the surrogate data set that we created in the source.

The screenshot shows the 'Source settings' tab for 'source1'. It includes fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Import data from SurrogatekeyDataset'), 'Source type' (set to 'Dataset'), and a 'Dataset' dropdown menu currently set to 'SurrogatekeyDataset'. There are also buttons for 'Test connection', 'Open', and 'New'.

7. Click on the plus symbol and click on the surrogate key.



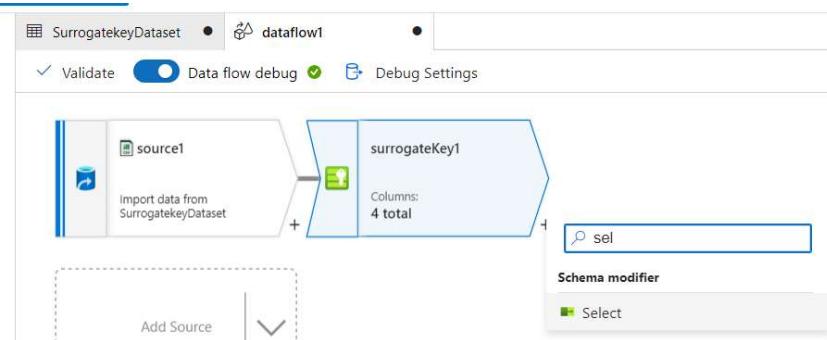
8. Set the key column as Empkey.

Surrogate key settings Optimize Inspect Data preview ●

Surrogatekey1 Learn more

Description	Adding new key Empkey starting from 1 with step 1	↻ Reset
Incoming stream *	source1	
Key column *	Empkey	
Start value *	1	
Step value	1	

9. Click on the plus symbol and click on Select.



10. Under Select set the properties as shown below.

Select settings Optimize Inspect Data preview ●

Incoming stream * surrogateKey1

Options

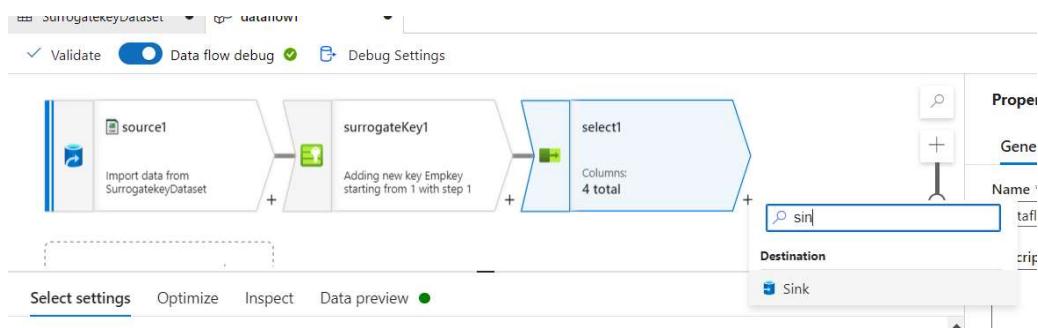
- Skip duplicate input columns ⓘ
- Skip duplicate output columns ⓘ

Input columns *

Auto mapping ⓘ ↻ Reset + Add mapping ⚡ Delete 4 mappings: All inputs mapped

Y	Name as	+	⚡
<input type="checkbox"/> surrogateKey1's column			
<input type="checkbox"/> 121 Empkey	Empkey	+	⚡
<input type="checkbox"/> abc Name	Name	+	⚡
<input type="checkbox"/> abc Country	Country	+	⚡
<input type="checkbox"/> abc DepID	DepID	+	⚡

11. Click on the plus symbol and select the sink.



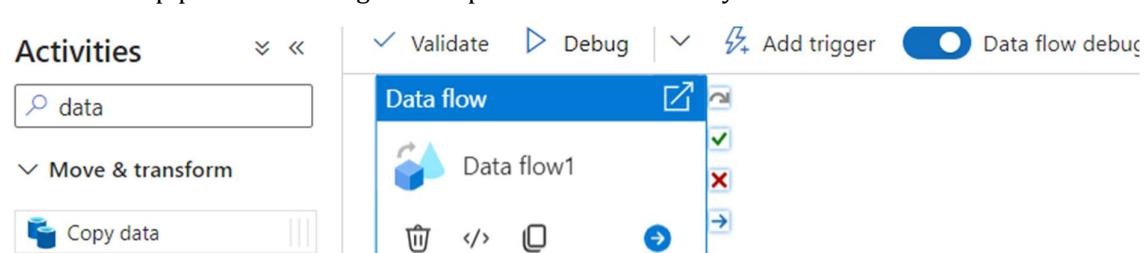
12. Select the destination dataset.

The screenshot shows the 'Sink' settings page. It includes fields for 'Output stream name' (sink1), 'Description' (Export data to ds_sink_txt), 'Incoming stream' (select1), 'Sink type' (Dataset), and 'Dataset' (ds_sink_txt). There are also buttons for 'Test connection', 'Open', and 'New'.

13. In the setting select the file option and file name as shown below.

The screenshot shows the 'Settings' tab in the Sink settings page. A message box states: 'This sink currently has Single partition set in Optimize. This will make your data flow efficient for current partitioning.' Below this, there are options for 'Clear the folder' (checkbox), 'File name option' (dropdown set to 'Output to single file'), and 'Output to single file' (text input set to 'SurrogateKeyOutEmp.csv').

14. Create a pipeline then drag and drop the data flow activity.



15. Select the data flow.

The screenshot shows the 'Settings' tab of a data flow configuration. The 'Data flow' dropdown is set to 'surrogate_dataflow'. The 'Run on (Azure IR)' dropdown is set to 'AutoResolveIntegrationRuntime'. The 'Compute size' dropdown is set to 'Small'. There are tabs for 'General', 'Settings' (selected), 'Parameters', and 'User properties'. A 'Save' button and a '+' icon are visible at the top right.

16. Validate the pipeline and click on debug.

17. Here our pipeline was executed successfully.

The screenshot shows the 'Output' tab of a pipeline run history. The 'Pipeline run ID' is 'a64259c2-b236-45e6-a43a-2316324a2341'. The status is 'Succeeded'. The table below shows one activity named 'Data flow1' which also has a status of 'Succeeded' and is of type 'Data flow'. Other tabs include 'Parameters', 'Variables', and 'Settings'.

Activity name	Activity status	Activity type
Data flow1	Succeeded	Data flow

18. Now go to a destination and check the data.

The screenshot shows the details of a blob container named 'f-demo'. It includes sections for 'Authentication method' (Access key or Azure AD User Account), 'Location' (f-demo / Output), and a 'Search blobs by prefix' input field. Below this is a table of blobs with columns: Name, Size, Type, and Last modified. The table contains three entries: '_SUCCESS', 'AllEmployees.csv', and 'maheer.csv'. An 'Edit' button is located at the bottom right of the table area.

Name	Size	Type	Last modified
_SUCCESS	0 B	File	2023-09-18T10:15:00Z
AllEmployees.csv	1.2 MB	File	2023-09-18T10:15:00Z
maheer.csv	1.2 MB	File	2023-09-18T10:15:00Z

Filter Transformation

1. In this example we are going to use Azure SQL Database TotalSale table that we created in Day 1 session.

The screenshot shows the Azure Data Studio interface. On the left, the object explorer displays the database structure: Tables (TotalSale), Views, and Stored Procedures. The TotalSale table is selected, showing its schema with columns: id, SalePersonFName, SalePersonLName, ProductName, ItemsSold, SoldPrice, Country, and Region. On the right, the results tab shows a preview of 10 rows from the TotalSale table.

id	SalePersonFName	SalePersonLName	ProductName	ItemsSold	SoldPrice	Country	Region
1	Aamir	Shahzad	TV	1	700	USA	North America
2	M	Raza	Cell Phone	2	800	USA	North America
3	Christy	Ladson	TV	3	1600	USA	North America
4	John	Rivers	Laptop	5	2400	USA	North America
5	Najaf	Ali	Computer	1	300	Pakistan	Asia
6	Sukhjeet	Singh	TV	2	900	India	Asia
7	Chirag	Patel	Cell Phone	5	1500	India	Asia
8	Aleena	Aman	Laptop	2	800	Pakistan	Asia
9	Petra	Henry	TV	10	5000	France	Europe
10	Rita	Roger	Laptop	7	2100	France	Europe

2. Create a Data flow.

The screenshot shows the 'Factory Resources' blade in the Power BI service. Under the 'Data flows' section, the 'Data flow' item is selected and highlighted. Other options like Pipeline, Change Data Capture (preview), Dataset, Power Query, and Copy Data tool are also visible.

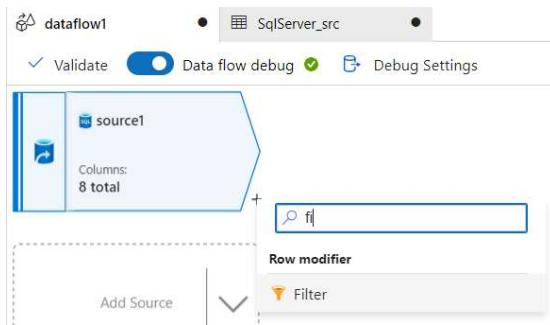
3. Add data source.

The screenshot shows the 'dataflow1' blade. A single data source named 'source1' is connected. The source type is set to 'Dataset'. The 'Data flow debug' toggle is turned on.

4. Under the source select the sql connection and table. Click on Open and go to Schema then click on Import Schema.

The screenshot shows the 'Source settings' blade for 'source1'. The 'Output stream name' is set to 'source1'. The 'Source type' is set to 'Dataset'. The 'Dataset' dropdown is set to 'SqlServer_src'. The 'Allow schema drift' option is checked.

5. Click on the Plus symbol and click on Filter.

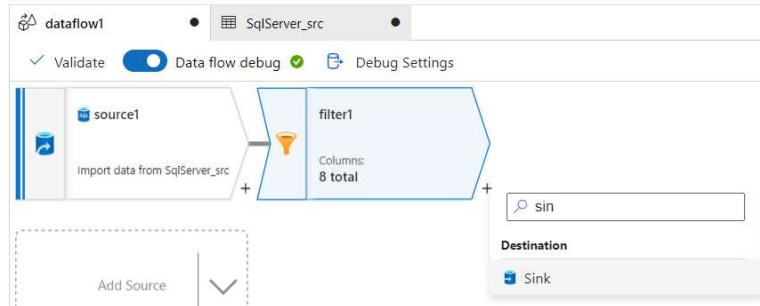


6. Under filter give the below expression.

Exp: equals(Region, 'Asia')

This screenshot shows the "Filter settings" dialog box. It includes fields for "Output stream name" (set to "filter1"), "Description" (set to "Filtering rows using expressions on columns 'Region'"), "Incoming stream" (set to "source1"), and "Filter on" (set to "equals(Region, 'Asia')"). There are tabs for "Filter settings", "Optimize", "Inspect", and "Data preview".

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



8. Under sink select the destination dataset.

This screenshot shows the "Sink" settings dialog box. It includes fields for "Incoming stream" (set to "filter1"), "Sink type" (with options "Dataset", "Inline", and "Cache" available), and "Dataset" (set to "ds_sink_txt"). There are tabs for "Sink", "Settings", "Errors", "Mapping", "Optimize", "Inspect", and "Data preview".

9. Under settings give the file option and file name as shown below.

The screenshot shows the 'Sink' tab selected in the top navigation bar. Below it, a message indicates that the sink currently has a single partition set in 'Optimize'. The 'File name option' dropdown is set to 'Output to single file', and the 'Output to single file' input field contains 'Asia_Sales.csv'. There is also a checkbox for 'Clear the folder'.

10. Create a Pipeline.

11. Drag and drop the Data flow activity.

The screenshot shows the 'Activities' pane on the left with various categories like 'Data', 'Move & transform', 'Copy data', and 'Data flow'. The 'Data flow' activity is selected and highlighted with a blue border. The main workspace shows a 'Data flow' component named 'Data flow1'.

12. Select the Data flow as shown below.

The screenshot shows the 'Settings' tab selected for the 'Data flow' activity. The 'Data flow' dropdown is set to 'filter_dataflow'. The 'Run on (Azure IR)' dropdown is set to 'AutoResolveIntegrationRuntime'. The 'Compute size' dropdown is set to 'Small'.

13. Now validate the pipeline and click on debug.

14. Here our pipeline was executed successfully.

The screenshot shows the 'Output' tab selected for the pipeline run. It displays the 'Pipeline run ID' as '0de0a2c3-f951-4e17-97b8-23994322d108'. Below it, a table lists the activities and their status:

Activity name	Activity status	Activity type
Data flow1	Succeeded	Data flow

15. Now go to a destination and check the data.

ID	SalePersonFName	SalePersonLName	ProductName	ItemsSold	SoldPrice	Country	Region
5	Najaf	Ali	Computer	1	300	Pakistan	Asia
6	Sukhjeet	Singh	TV	2	900	India	Asia
7	Chirag	Patel	Cell Phone	5	1500	India	Asia
8	Aleena	Aman	Laptop	2	800	Pakistan	Asia

Aggregator Transformation

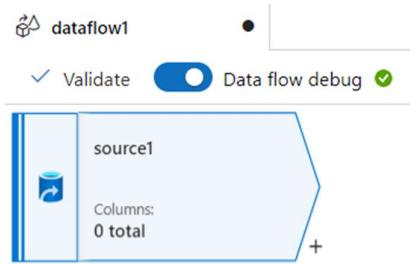
1. In this example we are going to use the Employee file.

empid	name	country	department
1	maheer	india	2
2	asmin	india	2
3	wafa	india	2
4	Sarfaraj	india	3
5	Ayesha	india	3
6	Pyarijaan	india	1
7	Mahaboob	india	1
8	Arfan	india	1
9	shabbir	india	1
10	Afrin	india	3
11	Shahin	india	2

2. Create a Dataflow.

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

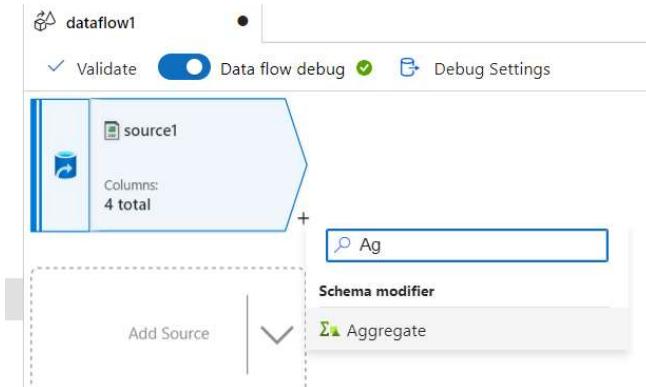
3. Add Data source.



4. Under the source select the dataset and file.

The screenshot shows the 'Source settings' tab of the Dataflow interface. It includes fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Import data from ds_src_txt'), and 'Source type' (set to 'Dataset'). A dropdown menu for 'Dataset' is open, showing 'ds_src_txt' selected. Other tabs include 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview'.

5. Click on the plus symbol and click on the Aggregate.



6. Under the Group select the Department column as shown below.

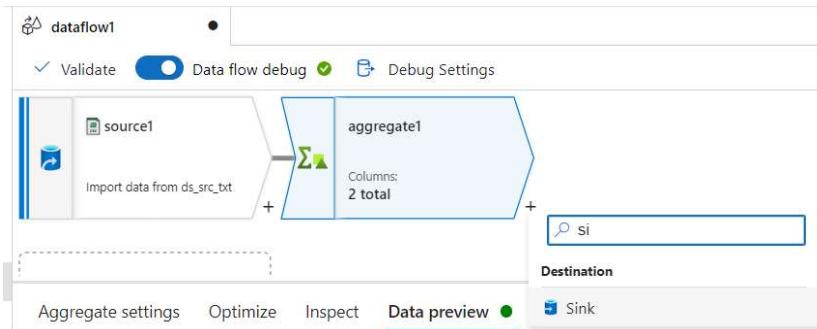
The screenshot shows the 'Aggregate settings' tab of the Dataflow interface. It includes fields for 'Output stream name' (set to 'aggregate1'), 'Description' (set to 'Add aggregate columns'), and 'Incoming stream' (set to 'source1'). The 'Group by' tab is selected. Below this, a table shows a column mapping: 'abc_department' is mapped to 'Name as' 'department'. Other tabs include 'Optimize', 'Inspect', and 'Data preview'.

7. Under Aggregates type TotalEmployees and give the below expression.

Exp: count(empid)

The screenshot shows the 'Aggregates' tab selected in the top navigation bar. The 'Incoming stream *' dropdown is set to 'source1'. Below it, there's a 'Group by' section and an 'Aggregates' section. The 'Grouped by: department' section contains a 'Count' aggregate named 'TotalEmployees' with the expression 'count(empid)'. There are also buttons for 'Add', 'Clone', 'Delete', and 'Open expression builder'.

8. Click on the Plus symbol and click on Sink.



9. Under sink select the destination dataset.

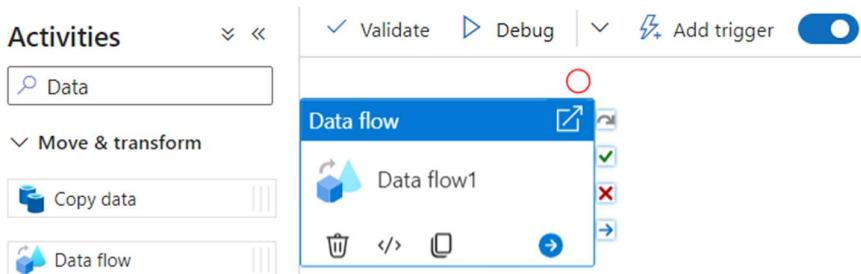
The screenshot shows the 'Sink' configuration page. The 'Sink' tab is selected. The 'Output stream name *' is 'sink1'. The 'Description' field contains 'Export data to ds_sink_txt'. The 'Incoming stream *' is 'aggregate1'. The 'Sink type *' section shows three options: 'Dataset' (selected), 'Inline', and 'Cache'. The 'Dataset *' dropdown is set to 'ds_sink_txt'. Buttons for 'Test connection', 'Open', and 'New' are available.

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

The screenshot shows the 'Settings' tab selected. A message box states: 'This sink currently has Single partition set in Optimize. This will make your data flow execution current partitioning.' Below, there are three configuration fields: 'Clear the folder' with a checkbox, 'File name option *' set to 'Output to single file', and 'Output to single file *' with the value 'Aggregate_Table.csv'.

11. Create a pipeline.

12. Drag and drop the Data flow activity.



13. Select the data flow.

A screenshot of the 'Settings' tab for a Data flow named 'Aggregate_dataflow'. It shows the 'Run on (Azure IR)' dropdown set to 'AutoResolveIntegrationRuntime' and the 'Compute size' dropdown set to 'Small'. There are tabs for General, Settings, Parameters, and User properties.

14. Validate the pipeline and click on debug.

15. Here our pipeline was executed successfully.

A screenshot of the Pipeline run details page. It shows a summary table with one item: 'Data flow1' status 'Succeeded' (green checkmark), activity type 'Data flow', and run date '7/21/'. Below this is a table showing the run history with one row: 'Activity name' 'Data flow1', 'Activity status' 'Succeeded', and 'Run' '7/21/'.

16. Now go to a destination and check the data.

A screenshot of the Azure Blob storage interface. It shows a table with three rows: 'department' and 'TotalEmployees'. The data is: department 3, TotalEmployees 3; department 1, TotalEmployees 4; department 2, TotalEmployees 4. The table has columns 'department' and 'TotalEmployees'.

Join Transformation

1. In this example, we are going to use two sources one is the Employee file, and the other one is the department file.

The screenshot shows the Azure Blob Storage interface. At the top, there are buttons for Upload, Change access level, Save, Discard, Download, Refresh, and Delete. Below that, it says 'Authentication method: Access key (Switch to Azure AD User Account)' and 'Location: f-demo'. A search bar for blobs by prefix is present, along with a 'Show deleted blobs' toggle and an 'Add filter' button. On the right, there's a table titled 'Overview' with columns 'depid' and 'depname'. The data rows are:

depid	depname
1	IT
2	HR
3	payroll

Below the table is an 'Edit' button. On the left, there's a sidebar with a list of items under 'Name': Dynamic_Pipelinedept, Dynamic_Pipelineemp, Dynamic_PipelineTotalSale, Output, Output1, Switch1, and Department.csv.

2. Create a dataset for this department file.

The screenshot shows the 'Dataset' creation dialog. It has fields for 'Name' (Dept_Dataset), 'Linked service *' (AzureBlobStorage_txt), 'File path' (f-demo / Directory / Department.csv), and 'First row as header' (checkbox checked). Under 'Import schema', the 'From connection/store' radio button is selected. The dialog also includes 'Validate' and 'Publish' buttons at the bottom.

3. Create a Data flow.

The screenshot shows the 'Factory Resources' blade with the 'Data flows' section selected. It lists several resources: Pipeline, Change Data Capture (preview), Dataset, Data flow (selected), Power Query, and Copy Data tool. The 'Dept_Dataset' dataset is also visible in the main pane.

4. Add a data source.

The screenshot shows the 'dataflow1' blade. It displays a single data source named 'source1' with the description 'Columns: 0 total'. There are buttons for Validate, Data flow debug, and Debug Settings at the top.

5. Under the source select the Employee file dataset.

Source settings Source options Projection Optimize Inspect Data preview

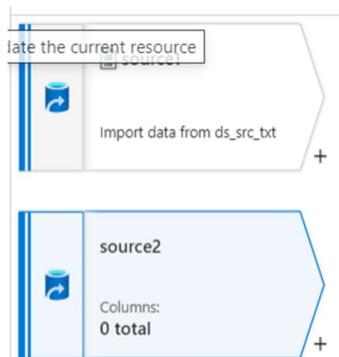
Output stream name * source1 Learn more ↗

Description Import data from ds_src_txt

Source type * Dataset Inline

Dataset * ds_src_txt Test connection Open New

6. Add another data source.



7. Select the department dataset that we created before.

Source settings Source options Projection Optimize Inspect Data preview

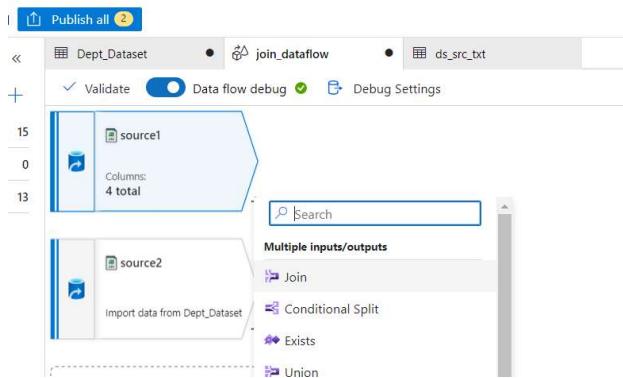
Output stream name * source2 Learn more ↗

Description Import data from Dept_Dataset

Source type * Dataset Inline

Dataset * Dept_Dataset Test connection Open New

8. Click on the plus symbol and click on join.



9. Under the join setting select the Right stream and give the below condition.

Join settings Optimize Inspect Data preview

Left stream * source1

Right stream * source2

Join type * Inner

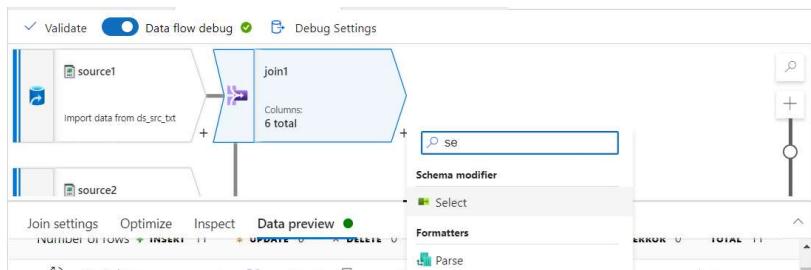
Use fuzzy matching

Join conditions

Left: source1's column Right: source2's column

abc department == abc depid

10. Click on the Plus symbol and click on Select.



11. Under settings delete the department and depid column as shown below.

Select settings Optimize Inspect Data preview

Options

Skip duplicate input columns

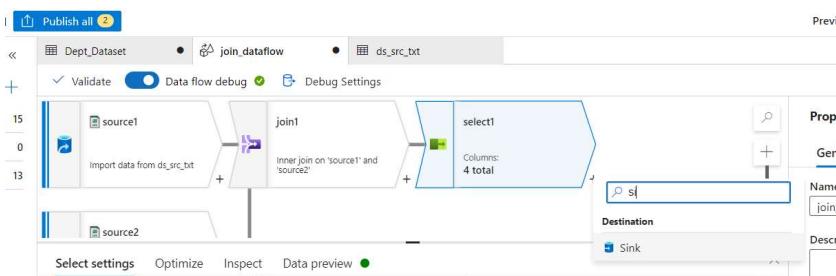
Skip duplicate output columns

Input columns *

Auto mapping Reset + Add mapping Delete 4 mappings: 2 column(s) from the inputs left unmapped

join1's column	Name as
abc empid	empid
abc name	name
abc country	country
abc depname	depname

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



13. Under sink select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview

Output stream name * sink1 Learn more

Description Export data to ds_sink_txt

Incoming stream * select1

Sink type * Dataset Inline Cache

Dataset * ds_sink_txt

Test connection Open New

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

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

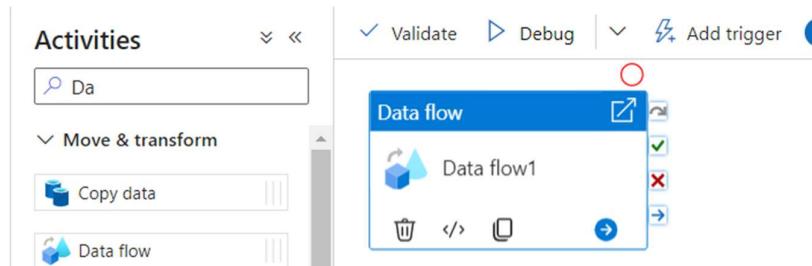
This sink currently has Single partition set in Optimize. This will make your data flow execution longer current partitioning.

Clear the folder

File name option * Output to single file

Output to single file * Join_Table.csv

15. Create a pipeline and drag and drop the data flow activity.



16. Select the data flow.

General **Settings** Parameters 1 User properties

Data flow * join_dataflow

Run on (Azure IR) * AutoResolveIntegrationRuntime

Compute size * Small

17. Validate it and click on debug.

The screenshot shows the 'Output' tab of a pipeline run. At the top, there are buttons for 'Validate', 'Debug', and 'Add trigger'. Below that is a note about triggering test runs without publishing. The main area shows a single activity named 'Data flow1' with a green checkmark indicating success. The pipeline run ID is 8c905f2b-7b70-4169-80bc-97c8e8a25ebd. The status table shows one item, 'Data flow1', with a status of 'Succeeded'.

18. Go to the destination and check the data.

The screenshot shows the 'Output/Join_Table.csv' file in a blob storage container. The file contains the following data:

empid	name	country	depname
1	maheer	india	HR
2	asmin	india	HR
3	wafa	india	HR
4	Sarfaraj	india	payroll
5	Ayesha	india	payroll
6	Pyarijaan	india	IT
7	Mahaboob	india	IT
8	Arfan	india	IT
9	shabbir	india	IT
10	Afrin	india	payroll
11	Shahin	india	HR

Lookup Transformation

1. In this example also we are using department and employee files. But in the employee file, we are going to edit a few values.
2. Here I edited the department value for the last two rows.

The screenshot shows the Azure Blob storage interface for the file 'Employee.csv'. The file contains 12 rows of data:

	empid	name	country	department
1	1	maheer	india	2
2	2	asmin	india	2
3	3	wafa	india	2
4	4	Sarfaraaj	india	3
5	5	Ayesha	india	3
6	6	Pyarijaan	india	1
7	7	Mahaboob	india	1
8	8	Arfan	india	1
9	9	shabbir	india	1
10	10	Afrin	india	4
11	11	Shahin	india	5

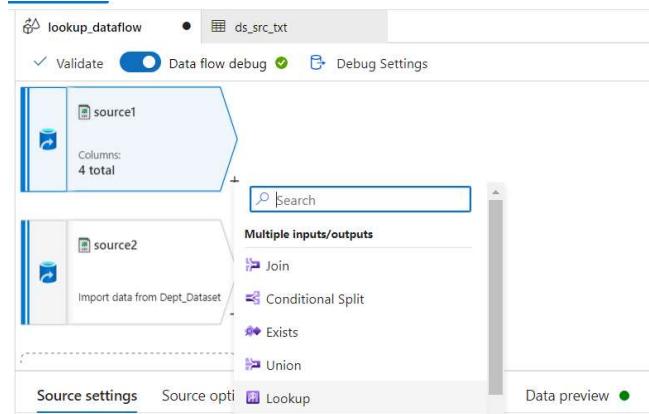
3. Create a data flow.

The screenshot shows the Azure Data Factory studio interface under the 'Data flows' section. A 'Data flow' item is selected in the list.

4. Add a source two sources and load the source file for the Employee and department.

The screenshot shows the Azure Data Flow designer. It displays two data source components: 'source1' and 'source2'. 'source1' is connected to a 'dept' dataset, which is labeled 'Import data from Dept_Dataset'.

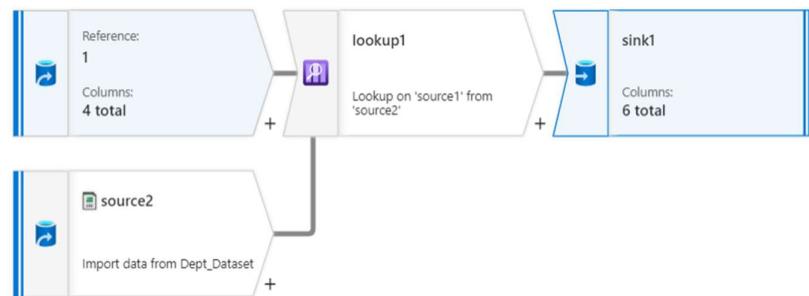
5. Click on the plus symbol and click on lookup as shown below.



6. Under lookup select the lookup stream and give the below condition.

This screenshot shows the 'Lookup settings' configuration page. It includes fields for 'Description' (Lookup on 'source1' from 'source2'), 'Primary stream' (set to 'source1'), 'Lookup stream' (set to 'source2'), 'Match multiple rows' (unchecked), 'Match on' (set to 'Any row'), and 'Lookup conditions' (Left: source1's column 'abc.department' and Right: source2's column 'abc.depid' connected by an equals sign). There are also '+' and '-' buttons for adding or removing conditions.

7. Add sink.



8. Under sink select the destination dataset.

This screenshot shows the 'Sink' configuration page. It includes tabs for 'Sink', 'Settings', 'Errors', 'Mapping', 'Optimize', 'Inspect', and 'Data preview'. Under 'Incoming stream*', 'lookup1' is selected. Under 'Sink type*', 'Dataset' is selected. In the 'Dataset*' field, 'ds_sink_txt' is entered. Other options like 'Inline' and 'Cache' are also shown. At the bottom, there are buttons for 'Test connection', 'Open', and 'New'.

9. Under setting select the file option and give the file name.

The screenshot shows the 'Sink' tab selected in the top navigation bar. Below it, a message box states: 'This sink currently has Single partition set in Optimize. This will make your data flow execution long current partitioning.' Under the 'File name option' dropdown, 'Output to single file' is selected, and the output file name is 'Lookup_table.csv'. There is also a 'Clear the folder' checkbox.

10. Create a lookup and drag and drop the data flow.

The screenshot shows the 'Activities' pane on the left with various options like Data, Move & transform, Copy data, and Data flow. The 'Data flow' activity is selected, and its details are shown on the right. A tooltip 'Data flow' is visible above the activity icon.

11. Select the data flow.

The screenshot shows the 'Settings' tab selected for the 'lookup_dataflow'. It includes fields for 'Data flow' (set to 'lookup_dataflow'), 'Run on (Azure IR)' (set to 'AutoResolveIntegrationRuntime'), and 'Compute size' (set to 'Small').

12. Validate the pipeline and click on debug.

13. Here our Pipeline was executed successfully.

The screenshot shows the 'Output' tab for the pipeline run. It displays the pipeline run ID: dc2e94c7-9ed2-4601-98df-005b0483bf97. The run status is 'Succeeded'. The table below lists the activity 'Data flow1' with status 'Succeeded' and type 'Data flow'.

Activity name	Activity status	Activity type
Data flow1	Succeeded	Data flow

14. Go to the destination and check the data.

Output/Lookup_table.csv ...

Blob

Save Discard Download Refresh Delete

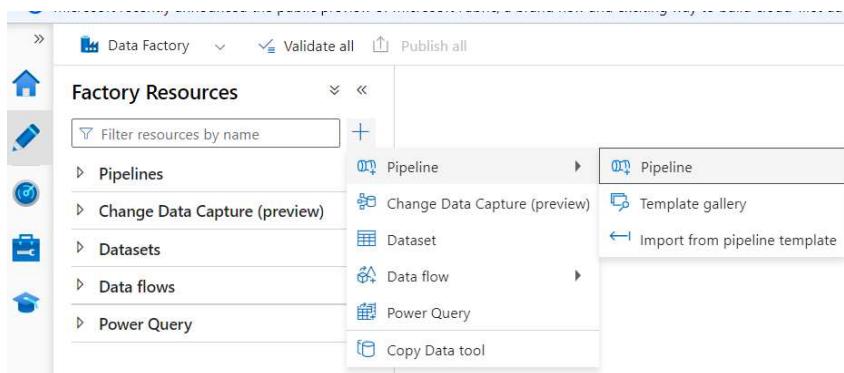
Overview Versions Snapshots Edit Generate SAS

empid	name	country	department	depid	depname
1	maheer	india	2	2	HR
2	asmin	india	2	2	HR
3	wafa	india	2	2	HR
4	Sarfaraj	india	3	3	payroll
5	Ayesha	india	3	3	payroll
6	Pyarijaan	india	1	1	IT
7	Mahaboob	india	1	1	IT
8	Arfan	india	1	1	IT
9	shabbir	india	1	1	IT
10	Afrin	india	4		
11	Shahin	india	5		

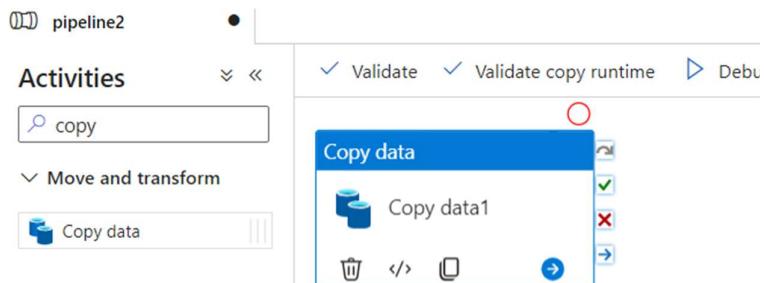
Edit

Pipeline Using REST

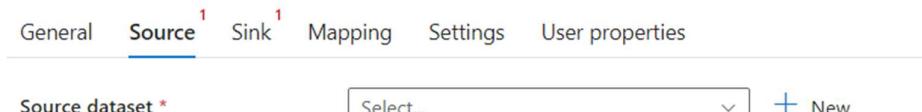
1. Open Azure Data Factory.
2. Create a Pipeline.



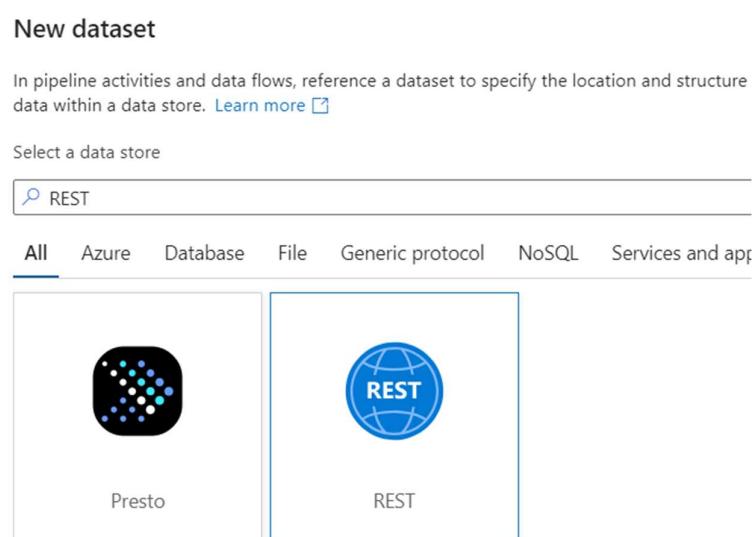
3. Drag and drop the Copy Data Activity as shown below.



4. Under source click on New.



5. Select REST and click on continue.



6. Give the Name and click on New.

Set properties

Name
RestResource

Linked service *
Select... Filter... + New

7. In the Linked Service under basic URL give the below URL.

URL: <https://swapi.dev/api/people>

Edit linked service



Name *

RestService

Description

Connect via integration runtime * ⓘ

AutoResolveIntegrationRuntime

Base URL *

https://swapi.dev/api/people

⚠ Information will be sent to the URL specified. Please ensure you trust the URL entered.

Authentication type *

Anonymous

Server Certificate Validation ⓘ

Enable Disable

Auth headers ⓘ

+ New

Annotations

+ New

✓ Connection successful

🔗 Test connection

Save

Cancel

8. After creating a linked service select it and click on Ok.

Set properties

Name Successfully created
Successfully created RestService (Linked service).

Linked service * 

> Advanced

9. We have configured our source part.

General Source Sink¹ Mapping Settings User properties

Source dataset *    [Learn more](#)

Request method 

Request timeout 

Request interval (ms) 

Additional headers 

10. Now under Sink click on New.

General Source Sink¹ Mapping Settings User properties

Sink dataset * 

11. Select Azure table Storage and click on continue.

New dataset

In pipeline activities and data flows, reference a dataset to specify the data within a data store. [Learn more](#)

Select a data store



All Azure Database File Generic protocol NoSQL


Azure Table Storage

12. Give the Name and click on New.

Set properties

Name
AzureTable

Linked service *

Select...
Filter...
+ New

+ New

13. Create a Linked Service with the below properties.

New linked service

Azure Table Storage [Learn more](#)

Name *

AzureTableStorage

Description

Connect via integration runtime * ⓘ

AutoResolveIntegrationRuntime

Authentication method

Account key

[Connection string](#)

[Azure Key Vault](#)

Account selection method ⓘ

From Azure subscription Enter manually

Azure subscription ⓘ

Microsoft Partner Network (REDACTED)

Storage account name *

adftraing

Additional connection properties

+ New

✓ Connection successful

⟳ Test connection

[Create](#) [Cancel](#)

14. Under Storage Account, go to Tables and click on plus symbol then create a Table.

The screenshot shows the 'Tables' blade in the Azure Storage account 'adftraing'. On the left, there's a sidebar with 'Data storage' options: Containers, File shares, Queues, and Tables. The 'Tables' option is selected. The main area has a search bar and a toolbar with '+ Table', 'Refresh', 'Delete', and 'Give feedback'. Below the toolbar, it says 'Authentication method: Access key (Switch to Azure AD User Account)'. A table list shows one entry: 'Characters' with a URL 'https://adftraing.table.core.windows.net/Characters'.

15. Select the Linked Service and select the table and click on Ok.

Set properties

Name
AzureTable

Linked service *
AzureTableStorage

Table name
Characters

Edit

> Advanced

16. We have configured our Sink part.

The screenshot shows the 'Sink' tab selected in a configuration pane. It includes tabs for General, Source, Sink, Mapping, Settings, and User properties. The Sink dataset is set to 'AzureTable'. There are sections for Insert type (set to 'Merge'), Partition key value selection (set to 'Specify partition value'), Default partition value, and Row key value selection (set to 'Unique identifier').

17. Under Mapping click Import schemas.

18. After a few seconds we should see the format of the API response appear.

19. Notice the response layout includes properties such as count, next, previous, and results.

ADF intuitively knows that the results property is an array.

20. Next, we'll set our Collection reference to the array named results. To do this, either select it from the drop-down list or check the box under the collection reference column.

21. Then we set up the Column names for the items we're mapping for import.

22. We'll map the fields: name, height, mass, and homeworld.

23. Uncheck the Include column for everything else.

24. Finally, type in the name we want our columns to be called (name, height, mass, homeworld.)

General Source Sink **Mapping** Settings User properties

Import schemas

+ New mapping Clear Delete Advanced editor

Collection reference ⓘ

Map complex values to string

Name	Type	Collection reference	Column name	<input type="checkbox"/> Include
results	[] Array			
name	abc String		<input type="text" value="Name"/>	
height	abc String		<input type="text" value="Height"/>	
mass	abc String		<input type="text" value="Mass"/>	
hair_color	abc String			<input type="checkbox"/>
skin_color	abc String			<input type="checkbox"/>
eye_color	abc String			<input type="checkbox"/>

25. Next validate the Pipeline and Click on Debug.

26. Here our pipeline was executed successfully.

Copy data

Copy data1

Parameters Variables Settings **Output**

Pipeline run ID: 96d2b5d4-45d4-4783-bfc9-c45238d0cd46

All status

Showing 1 - 1 of 1 items

Activity name ↑↓	Activity status ↑↓	Activity type ↑↓
Copy data1	Succeeded	Copy data

27. We built our ADF pipeline with a task to copy data from an external API and save the results into Azure Table Storage.

The screenshot shows the Azure Storage browser interface for the 'adftraing' storage account. On the left, there's a navigation sidebar with links like Home, Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Storage Mover, Containers, File shares, Queues, Tables, Security + networking, and Networking. The main area is titled 'adftraing | Storage browser' and shows a list of items under 'Tables > Characters'. The table has columns: PartitionKey, RowKey, Timestamp, Height, Homeworld, and Mass. There are 10 items listed, each with a checkbox and a preview of its values. An 'Add entity' button is at the top right.

PartitionKey	RowKey	Timestamp	Height	Homeworld	Mass
DefaultPartitionKey	06ca64a8-897d-4667-8f...	2023-08-16T05:34:48.36...	167	https://swapi.dev/api/pl...	75
DefaultPartitionKey	26380349-1c26-417c-a3...	2023-08-16T05:34:48.36...	96	https://swapi.dev/api/pl...	32
DefaultPartitionKey	38bb07db-5e0e-4fd5-9...	2023-08-16T05:34:48.37...	202	https://swapi.dev/api/pl...	136
DefaultPartitionKey	4415d037-4e65-4934-8...	2023-08-16T05:34:48.37...	182	https://swapi.dev/api/pl...	77
DefaultPartitionKey	4f30da7f-1ff9-4d11-b1...	2023-08-16T05:34:48.37...	150	https://swapi.dev/api/pl...	49
DefaultPartitionKey	51fa2e17-b106-4ca5-a4...	2023-08-16T05:34:48.37...	97	https://swapi.dev/api/pl...	32
DefaultPartitionKey	57fe8587-846b-4366-92...	2023-08-16T05:34:48.36...	172	https://swapi.dev/api/pl...	77
DefaultPartitionKey	8ee37713-93da-4f9d-9e...	2023-08-16T05:34:48.37...	178	https://swapi.dev/api/pl...	120
DefaultPartitionKey	9ed2a172-8c88-4077-a1...	2023-08-16T05:34:48.37...	183	https://swapi.dev/api/pl...	84
DefaultPartitionKey	a03a0866-a7c5-40ec-ad...	2023-08-16T05:34:48.37...	165	https://swapi.dev/api/pl...	75

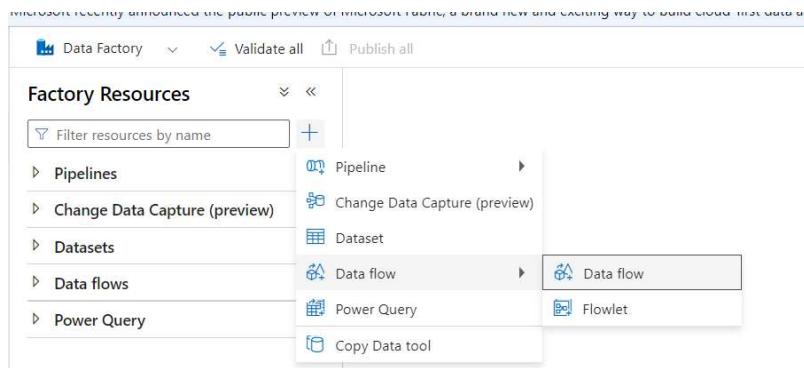
Parameterize

1. In this Example we are using Employee data.

The screenshot shows the Azure Blob storage interface for the 'Employee.csv' file. At the top, there are buttons for Save, Discard, Download, Refresh, and Delete. Below that is a navigation bar with Overview, Versions, Snapshots, Edit (which is selected), and Generate SAS. The main area displays a table with columns: empid, name, Gender, country, salary, and department. The table contains 11 rows of employee data. At the bottom is a blue 'Edit' button.

empid	name	Gender	country	salary	department
1	maheer	male	india	2000	HR
2	asmin	female	india	3000	HR
3	wafa	male	india	1000	HR
4	Sarfaraj	male	india	2000	Payroll
5	Ayesha	female	india	4000	IT
6	Pyarijaan	female	india	3000	IT
7	Mahaboob	male	india	5000	IT
8	Arfan	male	india	3000	IT
9	shabbir	male	india	4000	HR
10	Afrin	female	india	1000	Payroll
11	Shahin	female	india	5000	Payroll

2. Create a Data flow.



3. Create the below parameter.

A screenshot of the 'Parameters' blade. It shows a table with columns 'Name', 'Type', and 'Default value'. A new row is being added, with 'DepName' in the Name column, 'string' in the Type column, and 'abc' in the Default value column. The 'Enter expression...' placeholder is visible in the Default value field.

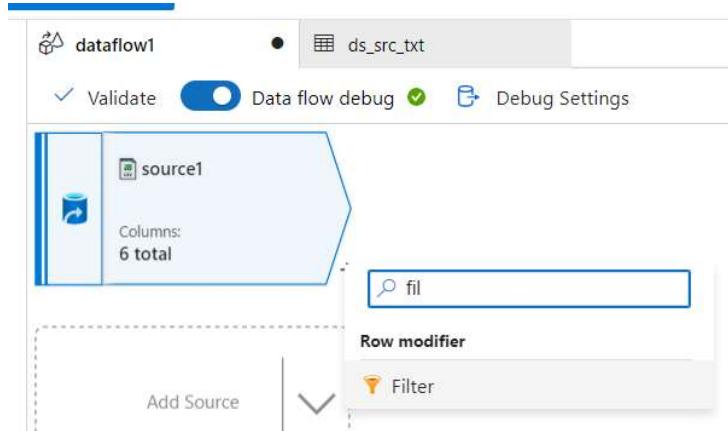
4. Add a Data Source.

A screenshot of the 'Data source' blade. It shows a single data source entry named 'source1'. The source type is listed as 'Dataset' with '0 total' columns. There is a 'Validate' button with a checkmark and a 'Data flow debug' toggle switch.

5. Under source select the source dataset.

A screenshot of the 'Source settings' blade for 'source1'. The 'Source settings' tab is active. Configuration includes: Output stream name set to 'source1'; Description set to 'Import data from ds_src_txt'; Source type set to 'Dataset'; Dataset selected as 'ds_src_txt'; and Options including 'Allow schema drift' checked. Other tabs like 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview' are also visible.

6. Click on the plus symbol and click on the filter.



7. Under filter give the below condition.

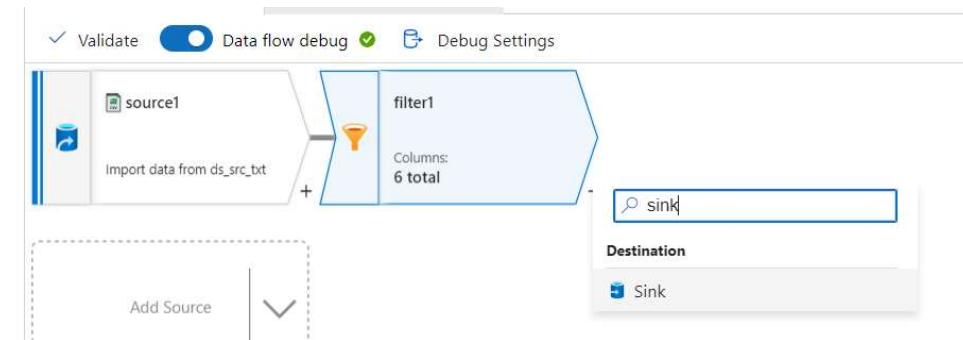
Con: department == \$DepName

This screenshot shows the 'Filter settings' dialog box. It includes tabs for 'Filter settings', 'Optimize', 'Inspect', and 'Data preview'. The 'Filter settings' tab is active. The configuration includes:

- Output stream name ***: filter1
- Description**: Filtering rows using expressions on columns 'department'
- Incoming stream ***: source1
- Filter on ***: department == \$DepName

A 'Learn more' link and a 'Reset' button are also present.

8. Click on the Plus symbol and click on Sink.



9. Under select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

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

Description Export data to ds_sink_txt [Reset](#)

Incoming stream * filter1

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

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

10. Under setting select the file option and give the filename.

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. The current partitioning.

Clear the folder

File name option * Output to single file

Output to single file * ⓘ Parameter.csv

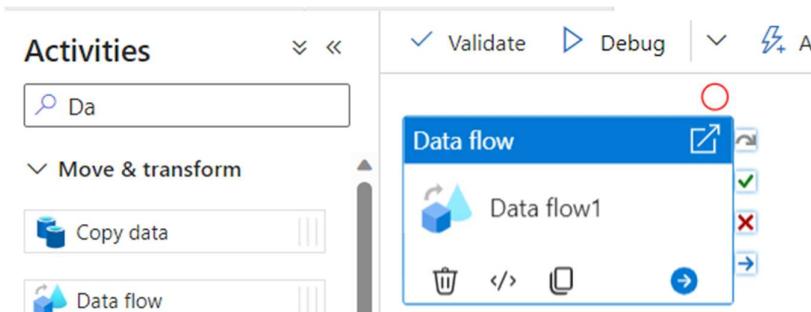
11. Create a Pipeline and create a parameter as shown below.

Parameters Variables Settings Output

+ New [Delete](#)

Name	Type	Default value
dep	String	<input type="text"/> Value Delete

12. Drag and drop the Data flow.



13. Under setting select the dataflow.

General **Settings** Parameters User properties

Data flow * Open + New

Run on (Azure IR) * ▼

Compute size * ▼

14. Under parameter click on the pipeline expression and give the below expression.

Exp: @pipeline().parameters.dep

General Settings **Parameters** ¹ User properties ^

Data flow parameters ①

Name	Value	Type	Expression ①
DepName	<input type="text" value="@pipeline().parameters.dep"/>	String	<input type="checkbox"/>

15. Now validate the pipeline and click on debug. Then it will ask for the parameter value and give it as HR.

Pipeline run

Parameters

Name	Type	Value
dep	string	<input type="text" value="HR"/>

16. Here our pipeline was executed successfully.

✓ Validate ⚡ Debug ▾ ⚡ Add trigger Data flow debug ✓

Data flow ✓
Data flow1

Parameters Variables Settings **Output**

Pipeline run ID: 5d0b0332-322a-47c1-9fa1-5c3793edb9a6 ✓ ↻ ⓘ

All status ▼

Showing 1 - 1 of 1 items

Activity name ↑↓	Activity status ↑↓	Activity type ↑↓
Data flow1	✓ Succeeded	Data flow

17. Go to Destination and check the data.

The screenshot shows the Azure Data Explorer interface. On the left, a file browser lists various CSV files. On the right, the details for 'Output/Parameter.csv' are shown. The table has the following data:

empid	name	Gender	country	salary	department
1	maheer	male	india	2000	HR
2	asmin	female	india	3000	HR
3	wafa	male	india	1000	HR
9	shabbir	male	india	4000	HR

Stringify Transformation

1. In this example we are going to use below json data.

The screenshot shows the Azure Data Explorer interface. On the left, a file browser lists various files. On the right, the details for 'demo.json' are shown. The JSON file contains the following data:

```
1 {"name": "Mahesh", "skills": ["\".net\", \"Azure\""], "contact": {"\"mobile\": \"54353\", \"landline\": \"32425\"}}\n2 {"name": "Ravi", "skills": ["\"AWS\", \"Java\""], "contact": {"\"mobile\": \"235245\", \"landline\": \"9768\"}}
```

2. Create a Dataset for the above json file.

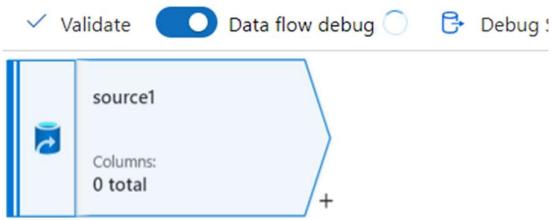
3. Create a Data flow.

The screenshot shows the Azure Data Factory 'Factory Resources' blade. The 'Data flows' section is selected. The available options are:

- Pipeline
- Change Data Capture (preview)
- Dataset
- Data flow
- Power Query
- Flowlet
- Copy Data tool

'Data flow' is highlighted.

4. Add a Data source.



5. Select the Source dataset.

The screenshot shows the 'Source settings' tab for the 'source1' component. The 'Output stream name' is set to 'source1'. The 'Description' field contains 'Import data from Demo_Json'. Under 'Source type', 'Dataset' is selected. The 'Dataset' dropdown shows 'Demo_Json'. There is a checked checkbox for 'Allow schema drift'. Other tabs like 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview' are visible at the top.

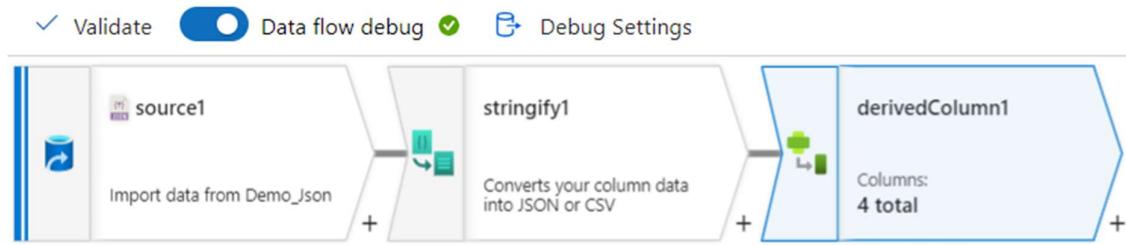
6. Click on the plus symbol and click on Stringify.

The screenshot shows the Data flow interface with a 'source1' component and a new 'Stringify' component added to the pipeline. The 'Stringify' component has a search bar containing 'strin'. The 'Source settings' tab is selected for 'source1'. Other tabs like 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview' are visible at the top.

7. Under settings give the new column name and give the below expression as contact.

The screenshot shows the 'Stringify settings' tab for the 'stringify1' component. The 'Output stream name' is 'stringify1'. The 'Description' field contains 'Converts your column data into JSON or CSV'. The 'Incoming stream' is 'source1'. The 'Format' is set to 'JSON'. Under 'Columns', there is a table with one row. The 'Column' is 'contactStringify' and the 'Expression' is 'contact'. Other tabs like 'Stringify settings', 'Optimize', 'Inspect', and 'Data preview' are visible at the top.

8. Add derived column.

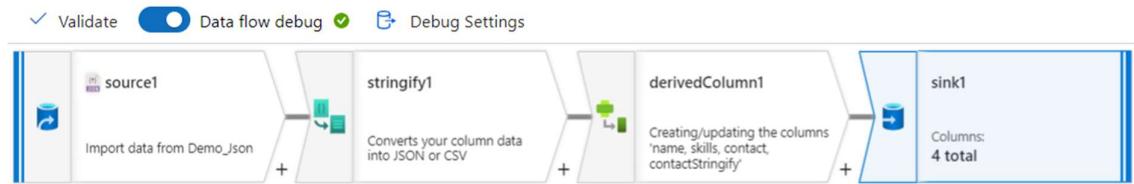


9. Under settings select the column and give the below expression.

Exp: `toString(contactStringify)`

This is a detailed view of the 'derivedColumn1' settings. It shows the output stream name as 'derivedColumn1', a description of 'Creating/updating the columns 'name, skills, contact, contactStringify'', and an incoming stream of 'stringify1'. Under 'Columns', there is one column named 'contactStringify' with the expression `toString(contactStringify)`.

10. Add Sink.



11. Add the target dataset.

This is the 'Sink' settings page for 'sink1'. It specifies the output stream name as 'sink1', a description of 'Export data to ds_sink_txt', and an incoming stream of 'derivedColumn1'. The sink type is set to 'Dataset', and the dataset is selected as 'ds_sink_txt'.

12. 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.

Clear the folder

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

Output to single file * **demo.csv**

Quote All

13. Under mapping delete the fields except below two.

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

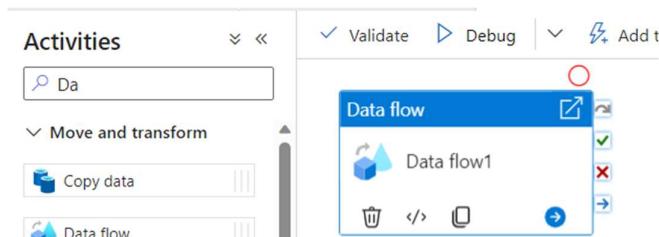
i This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The recommended setting is

Options Skip duplicate input columns ①
 Skip duplicate output columns ①

Auto mapping ① Reset Add mapping Delete **Output format** 2 mappings

Input columns	Output columns
<input type="checkbox"/> abc_name	name
<input type="checkbox"/> abc_contactStringify	contact

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



15. Under settings select the Data Flow.

General **Settings** Parameters User properties

Data flow * **Stringify_dataflow** Open New

Run on (Azure IR) * **AutoResolveIntegrationRuntime**

Compute size * **Small**

16. Validate the Pipeline and click on Debug.

17. Here our pipeline executed successfully.

The screenshot shows the Azure Data Flow pipeline execution interface. At the top, there are buttons for 'Validate' (green checkmark), 'Debug' (blue play button), 'Add trigger' (lightning bolt icon), and 'Data flow debug' (blue toggle switch with green checkmark). Below this is a 'Data flow' section with a 'Data flow1' item. The main area has tabs for 'Parameters', 'Variables', 'Settings', and 'Output'. The 'Output' tab is selected. It displays the 'Pipeline run ID' as 69d1aa54-8933-46ed-9c8e-80e1fc5d7944. Below this, it says 'Showing 1 - 1 of 1 items' and lists one activity: 'Data flow1' with status 'Succeeded' and type 'Data flow'.

18. Go to Destination and check the Data.

The screenshot shows the Azure Blob storage destination for the pipeline. It displays the file name 'FilterOutput/demo.csv'. Below it are buttons for 'Save', 'Discard', 'Download', 'Refresh', and 'Delete'. The 'Edit' tab is selected. The content of the CSV file is shown in a code editor-like interface:
1 name,contact
2 Mahesh,"{\\"mobile\\":\\"54353\\",\\"landline\\":\\"32425\\\"}"
3 Ravi,"{\\"mobile\\":\\"235245\\",\\"landline\\":\\"9768\\\"}"

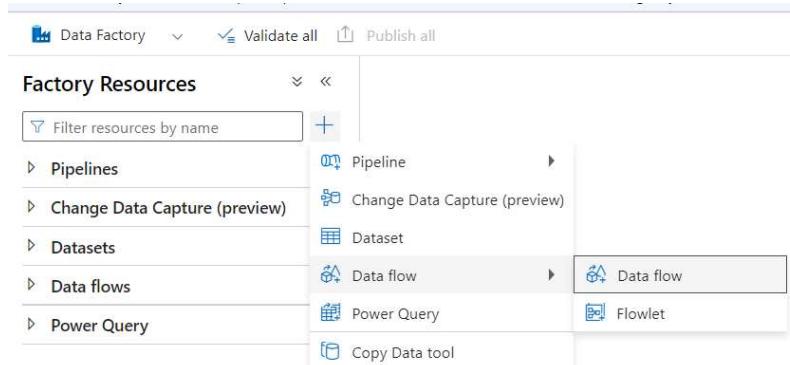
Pivot Transformation

1. In this example we are going to use below employee table as shown below.

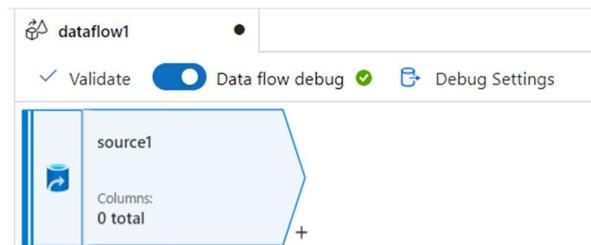
The screenshot shows the Azure Blob storage destination for the employee data. It displays the file 'Employee.csv'. The left sidebar shows a list of blobs, with 'Employee.csv' being checked. The main area shows the CSV content with 12 rows of employee data:

	empid	name	Gender	country	department
1	1	maheer	male	india	2
2	2	asmin	female	india	2
3	3	wafa	male	india	2
4	4	Sarfraz	male	india	3
5	5	Ayesha	female	india	3
6	6	Pyarijaan	female	india	1
7	7	Mahabob	male	india	1
8	8	Arfan	male	india	1
9	9	shabir	male	india	1
10	10	Afrin	female	india	4
11	11	Shahin	female	india	5

2. Go to Azure Data Factory and create a Data flow.



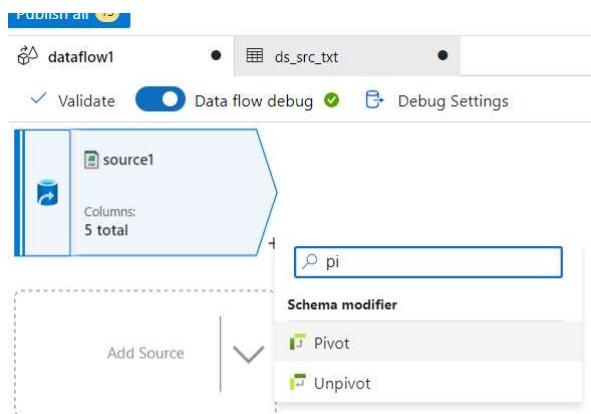
3. Add a Data Source.



4. Select the Employee data set as shown below.

A screenshot of the 'Source settings' tab in the Azure Data Factory Data flow editor. It includes fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Import data from ds_src_txt'), 'Source type' (set to 'Dataset'), and a dropdown for 'Dataset' (set to 'ds_src_txt'). There are also buttons for 'Test connection', 'Open', and 'New'. The 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview' tabs are visible above this section.

5. Now click on the plus symbol and click on Pivot as shown below.



6. Under Pivot Setting for the Group by select Department.

Pivot settings Optimize Inspect Data preview ●

Output stream name * pivot1 ? Help Learn more ⓘ

Description Pivots row values into columns, groups columns and aggregates data ⌂ Reset

Incoming stream * source1

1. Group by 2. Pivot key 3. Pivoted columns

Columns	Name as
abc_department	department

7. For the Pivot key select Gender as shown below.

1. Group by 2. Pivot key 3. Pivoted columns

Pivot key * abc_Gender

Value

Enter value (optional)...

Null value

8. Under Pivoted columns give the below expression.

Exp: count(empid)

1. Group by 2. Pivot key 3. Pivoted columns

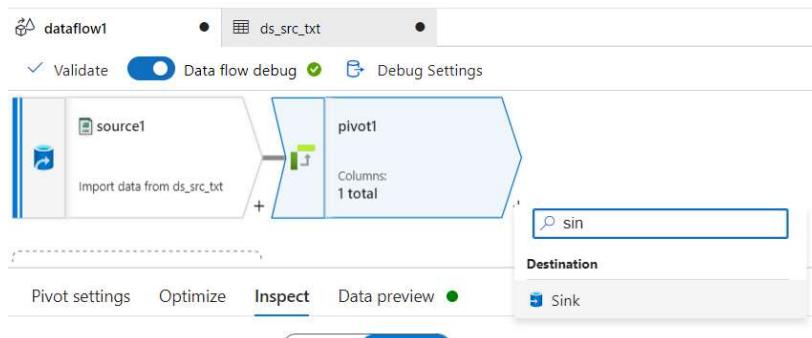
Column name pattern * prefix{expression prefix}middle{Pivot key value}suffix

Prefix Middle Suffix

Column arrangement * Normal Lateral

count(empid) 12l Enter a column prefix (optional)...

9. Now click on the Plus symbol and click on Sink.



10. Under sink select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview

Output stream name * Learn more

Description

Incoming stream *

Sink type * Dataset Inline Cache

Dataset *

Skip line count

11. Under the Setting set the file option and file name.

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. current partitioning.

Clear the folder

File name option *

Output to single file *

12. Create a Pipeline and drag and drop the Data flow as shown below.

Activities

Move & transform

Data flow Data flow1

13. Select the Data flow.

General **Settings** Parameters User properties

Data flow *

Run on (Azure IR) *

Compute size *

> Advanced

Logging level * Verbose Basic None

14. Validate and click on debug.

15. Here our pipeline was executed successfully.

The screenshot shows the Azure Data Flow pipeline execution results. At the top, there are navigation links: 'Validate' (with a checkmark), 'Debug' (with a play icon), 'Add trigger' (with a lightning bolt icon), and a toggle switch for 'Data flow debug' which is turned on (green). Below this is a 'Data flow' section with a 'Data flow1' item, indicated by a blue cube icon. The main area is titled 'Output' and displays the following information:

- Pipeline run ID:** 1f0c0b76-63fa-41cd-98c2-54c0cdb3d941
- Status:** All status (dropdown menu)
- Items:** Showing 1 - 1 of 1 items
- Activity name:** Data flow1
- Activity status:** Succeeded (green checkmark)
- Activity type:** Data flow

16. Now go to Destination and check the data.

The screenshot shows the Azure Blob storage output details for 'Output/Pivot.csv'. The page title is 'Output/Pivot.csv'. It includes standard file operations: Save, Discard, Download, Refresh, and Delete. Below these are tabs: Overview, Versions, Snapshots, Edit (which is selected and underlined), and Generate SAS. A preview table is displayed:

department	female	male
3	1	1
1	1	3
4	1	
2	1	2
5	1	

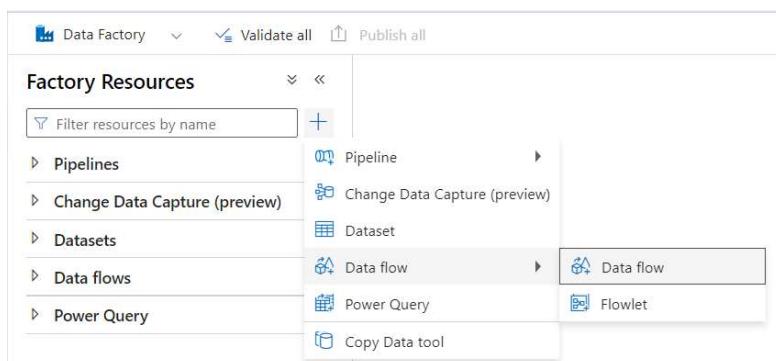
At the bottom is a large blue 'Edit' button.

Unpivot Transformation

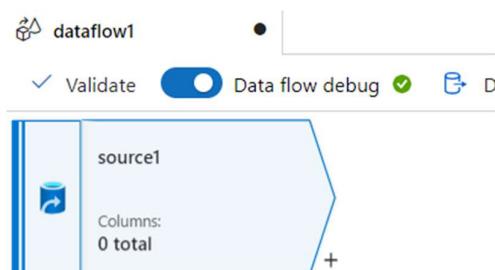
1. In this example we are going to use the below data.

```
input_text
File Edit View
PO,Vendor,Apple,Mango
1,A,2,3
2,B,1,1
```

2. Upload the file into the container and create a Dataset for this data.
3. Next create a Data flow.



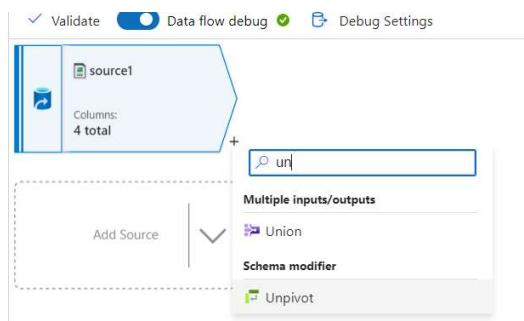
4. Add a Data Source.



5. Select the source Dataset.

A screenshot of the 'Source settings' tab in the Data flow editor. The tab has tabs for 'Source settings', 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview'. Under 'Source settings', there are fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Import data from ds_src_txt'), 'Source type' (set to 'Dataset'), and a dropdown for 'Dataset' (set to 'ds_src_txt'). There are also 'Test connection', 'Open', and 'New' buttons. At the bottom, there's an 'Options' section with a checked checkbox for 'Allow schema drift'.

6. Click on the plus symbol and click on Unpivot.



7. Under Unpivot settings, for the Ungroup add the below columns.

Unpivot settings Optimize Inspect Data preview

Output stream name * ? Help Learn more

Description ⌂ Reset

Incoming stream *

1. Ungroup by 2. Unpivot key 3. Unpivoted columns

Columns

abc PO	+	trash
abc Vendor	+	trash

8. For the Unpivot key give the column name and column type as shown below.

incoming stream

1. Ungroup by 2. Unpivot key 3. Unpivoted columns

Unpivot column name *

Unpivot column type *

Option * Pick column names as values Enter values

9. For the Unpivoted columns give the column name and column type.

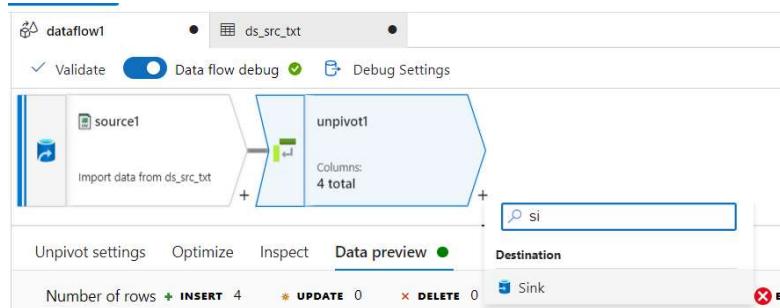
1. Ungroup by 2. Unpivot key 3. Unpivoted columns

Column arrangement * Normal Lateral

Drop rows with null

columns *	Column name	Type
	Amount	abc string

10. Click on the plus symbol and click on Sink.



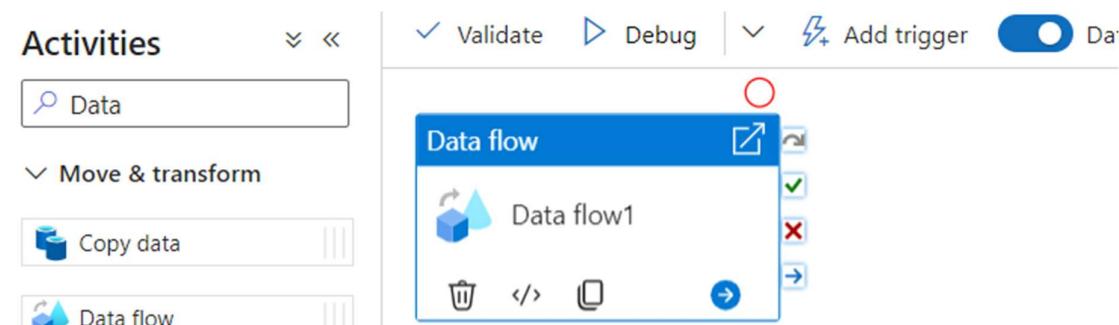
11. Select the Destination dataset.

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 'unpivot1'), 'Sink type' (set to 'Dataset'), and a 'Dataset' dropdown (set to 'ds_sink_txt'). There are also buttons for 'Test connection', 'Open', and 'New'.

12. Under setting set the file option and file name.

This screenshot shows the 'Settings' tab in the Sink configuration. A note indicates that the current partitioning is 'Single partition', which may increase execution time. Below this, there are options for 'Clear the folder' (unchecked), 'File name option' (set to 'Output to single file'), and 'Output to single file' (set to 'Unpivot.csv').

13. Create a Pipeline and drag and drop the data flow.



14. Under setting select the dataflow.

Data flow *

unpivot_dataflow

Run on (Azure IR) * ⓘ

AutoResolveIntegrationRuntime

Compute size * ⓘ

Small

✓ Validate ⚡ Debug ⚡ Add trigger ⚡ Data flow debug ✓

15. Now validate it and click on debug.

16. Here our pipeline was executed successfully.

Parameters Variables Settings Output

Pipeline run ID: 7dea48f4-39be-405d-999d-72b14f5ebc89 ⓘ

All status

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type
Data flow1	Succeeded	Data flow

17. Go to Destination and check the data.

Output/Unpivot.csv ...

Blob

Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

PO	Vendor	Fruits	Amount
1	A	Apple	2
1	A	Mango	3
2	B	Apple	1
2	B	Mango	1

Window Transformation

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

Employee.csv ...

Blob

Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

empid	name	Gender	country	salary	department
1	maheer	male	india	2000	HR
2	asmin	female	india	3000	HR
3	wafa	male	india	1000	HR
4	Sarfaraj	male	india	2000	Payroll
5	Ayesha	female	india	4000	IT
6	Pyarjaan	female	india	3000	IT
7	Mahaboop	male	india	5000	IT
8	Arfan	male	india	3000	IT
9	shabbir	male	india	4000	HR
10	Afrin	female	india	1000	Payroll
11	Shahin	female	india	5000	Payroll

Edit

2. Create a Data flow.

Data Factory Validate all Publish all

Factory Resources < <

Filter resources by name +

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

Pipeline

Change Data Capture (preview)

Dataset

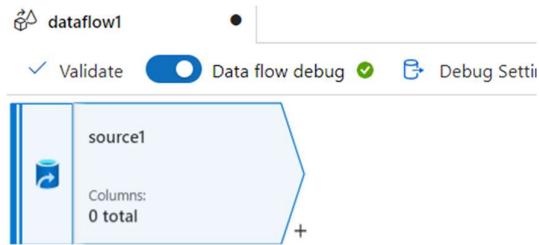
Data flow

Power Query

Flowlet

Copy Data tool

3. Add data source.



4. Select the source file dataset.

Source settings Source options Projection Optimize Inspect Data preview

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

Description

Source type * Dataset Inline

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

Allow schema drift [🔗](#)

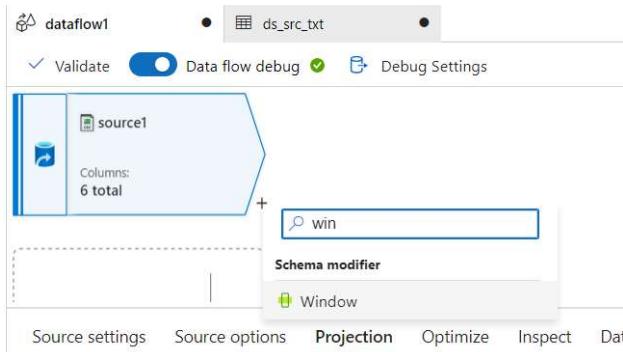
5. Under the Projection set the data types.

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	Specify format
name	abc string	Specify format
Gender	abc string	Specify format
country	abc string	Specify format
salary	123 integer	Specify format
department	abc string	Specify format

6. Click on the plus symbol and click on Window.



7. Under Setting for the Over set the department column.

Window settings Optimize Inspect Data preview

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

Description

Incoming stream *

1. Over 2. Sort 3. Range by 4. Window columns

source1's column Name as

abc department [🔗](#) department [➕](#) [✖️](#)

8. For the Sort set the salary column and order as shown below.

1. Over 2. Sort 3. Range by 4. Window columns

source1's column	Order	Nulls first
123 salary	Descending	<input checked="" type="checkbox"/>

9. Under Window columns give the column name and expression.

Exp: denseRank()

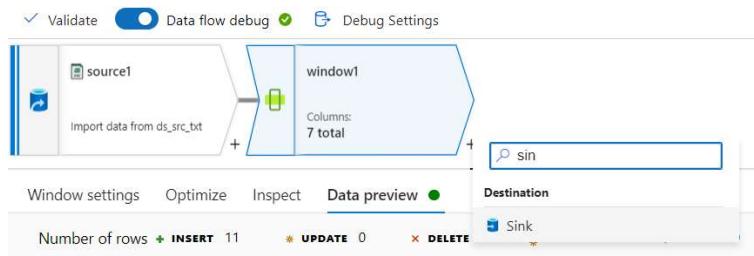
Incoming stream: source1

1. Over 2. Sort 3. Range by 4. Window columns

Add Clone Delete Open expression builder

Column	Expression
<input type="checkbox"/> DenseRank	denseRank()

10. Now click on the plus symbol and click on the sink.



11. Under sink select the destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview

Output stream name * sink1 Learn more

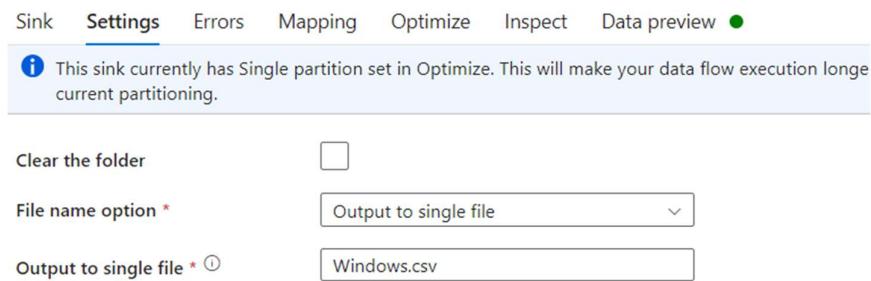
Description Export data to ds_sink_txt

Incoming stream * window1

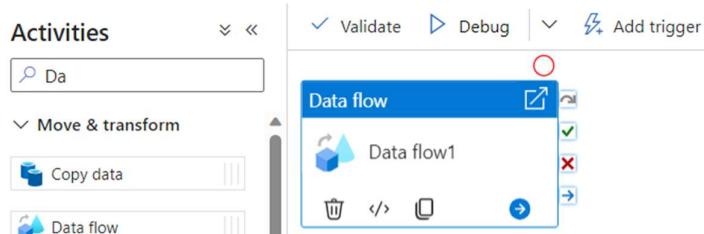
Sink type * Dataset Inline Cache

Dataset * ds_sink_txt Test connection Open

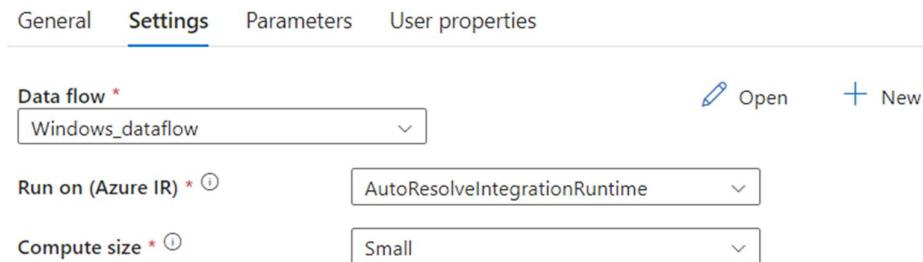
12. Under Settings set the file type and file name as shown below.



13. Create a Pipeline and drag and drop the Data flow as shown below.

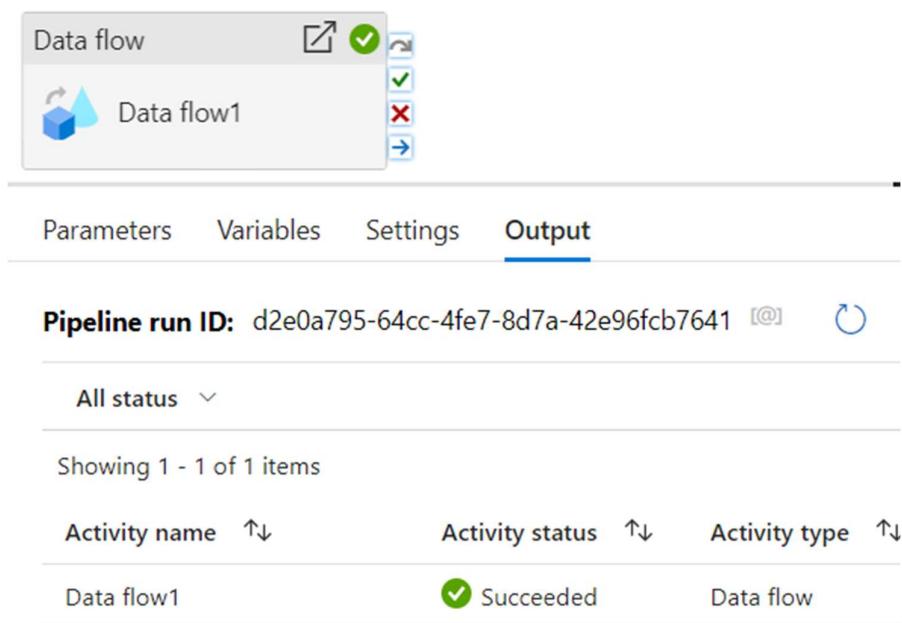


14. Under setting select the Data flow.



15. Validate the pipeline and click on debug.

16. Here our pipeline was executed successfully.



17. Go to Destination and check the data.

Output/Windows.csv ...
Blob

A screenshot of a CSV file in a browser. The file is titled "Output/Windows.csv". The interface includes standard file operations like Save, Discard, Download, Refresh, and Delete. Below these are tabs for Overview, Versions, Snapshots, Edit (which is selected), and Generate SAS. The main content is a table with the following data:

empid	name	Gender	country	salary	department	DenseRank
9	shabbir	male	india	4000	HR	1
2	asmin	female	india	3000	HR	2
1	maheer	male	india	2000	HR	3
3	wafa	male	india	1000	HR	4
11	Shahin	female	india	5000	Payroll	1
4	Sarfaraj	male	india	2000	Payroll	2
10	Afrin	female	india	1000	Payroll	3
7	Mahaboob	male	india	5000	IT	1
5	Ayesha	female	india	4000	IT	2
6	Pyarijaan	female	india	3000	IT	3
8	Arfan	male	india	3000	IT	3

Alter Row Transformation

1. In this example our is below employee details.

Employee.csv ...
Blob

A screenshot of a CSV file in a browser. The file is titled "Employee.csv". The interface includes standard file operations like Save, Discard, Download, Refresh, and Delete. Below these are tabs for Overview, Versions, Snapshots, Edit (which is selected), and Generate SAS. The main content is a table with the following data:

empid	name	Gender	country	salary	department
1	maheer	male	india	2000	HR
2	asmin	female	india	3000	HR
3	wafa	male	india	1000	HR
4	Sarfaraj	male	india	2000	Payroll
5	Ayesha	female	india	4000	IT
6	Pyarijaan	female	india	3000	IT
7	Mahaboob	male	india	5000	IT
8	Arfan	male	india	3000	IT
9	shabbir	male	india	4000	HR
10	Afrin	female	india	1000	Payroll
11	Shahin	female	india	5000	Payroll

Edit

2. In the Sql Database we have created a sample table with the sample data.
3. This is our destination table.

Code: Create table tbl_Employees(

```

EmpId int,
EmpName varchar(50),
Gender varchar(10),
Country varchar(10),
Salary int,
Department varchar(50)
)
```

Insert into tbl_Employees values(1, 'ABCD','male','India', 1000, 'XTZ')

Insert into tbl_Employees values(10, 'ABCDEF','male','India', 1000, 'XTZXYZ')

The screenshot shows the Azure Data Studio interface for a SQL database named 'sqldatabase'. The left sidebar contains navigation links like Overview, Activity log, Tags, Diagnose and solve problems, Getting started, and Query editor (preview). The main area has a 'Query 1' tab with the following code:

```

1  USE [master]
2  GO
3  IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'tbl_Employees')
4  BEGIN
5      CREATE TABLE [dbo].[tbl_Employees] (
6          EmpId int,
7          EmpName varchar(50),
8          Gender varchar(10),
9          Country varchar(10),
10         Salary int,
11         Department varchar(50)
12     )
13 END
14 GO
15 Insert into tbl_Employees values(1, 'ABCD','male','India', 1000, 'XTZ')
16 Insert into tbl_Employees values(10, 'ABCDEF','male','India', 1000, 'XTZXYZ')
17 drop table tbl_Employees
18 Select * from tbl_Employees;
19 GO

```

The results pane shows a table with two rows of data:

EmpId	EmpName	Gender	Country	Salary	Department
1	ABCD	male	India	1000	XTZ
10	ABCDEF	male	India	1000	XTZXYZ

4. Next create a dataset for this table.

5. Create a Data flow.

The screenshot shows the 'Factory Resources' blade in Power BI Data Studio. Under the 'Pipelines' section, there is a 'Data flow' item highlighted in a dropdown menu. Other options in the menu include Pipeline, Change Data Capture (preview), Dataset, Power Query, and Copy Data tool.

6. Add Data Source.

The screenshot shows the 'Data flow' configuration screen. At the top, there are validation and debug settings: 'Validate' (green checkmark), 'Data flow debug' (switch), and 'Det' (button). Below this is a component panel containing a single component labeled 'source1'. The component has a circular icon with a blue arrow and a table icon. Below the component, it says 'Columns: 0 total'.

7. Under Source 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](#)

8. Under projection set the datatypes as shown below.

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	Specify format
name	abc string	Specify format
Gender	abc string	Specify format
country	abc string	Specify format
salary	123 integer	Specify format
department	abc string	Specify format

9. Click on the plus and click on Alter Row.

✓ Validate [Data flow debug](#) [Debug Settings](#)

source1
Columns: 6 total

+ [Alter](#)

Row modifier [Alter Row](#)

10. Under alter row condition give the below conditions.

Alter row settings Optimize Inspect Data preview

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

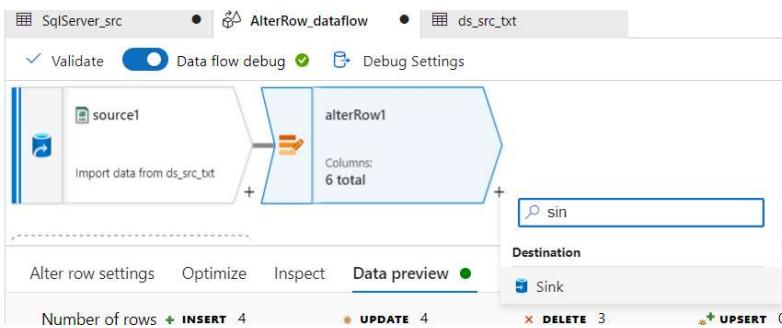
Description [Reset](#)

Incoming stream *

Alter row conditions * [Delete if department == 'Payroll'](#) [+](#)

* [Update if department == 'HR'](#) [+](#)

11. Click on the Plus symbol and click on Sink.



12. Under sink select the destination dataset.

The screenshot shows the Sink tab settings. It includes fields for Output stream name (sink1), Description (Export data to SqlServer_src), Incoming stream (alterRow1), Sink type (Dataset selected), Dataset (SqlServer_src selected), and Options (Allow schema drift checked).

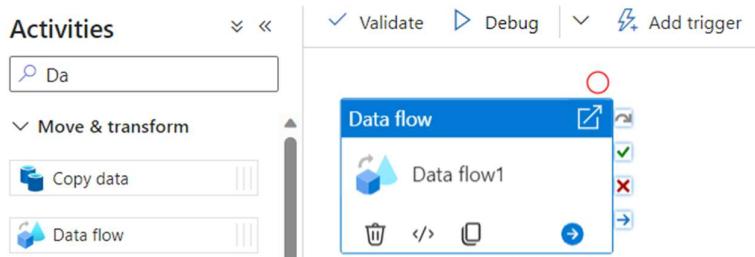
13. Under settings check the below properties.

The screenshot shows the Settings tab under Sink. It includes sections for Update method (Allow insert, Allow delete, Allow upsert, Allow update) and Key columns (List of columns selected, EmpId selected).

14. Under Mapping, Map the fields as shown below.

The screenshot shows the Mapping tab under Sink. It includes sections for Options (Skip duplicate input columns, Skip duplicate output columns checked), Auto mapping (unchecked), and a detailed mapping grid where Input columns map to Output columns.

15. Create a Pipeline and Drag and Drop the Dataflow as shown below.



16. Select Dataflow.

A screenshot of the Data flow settings page. The tab 'Settings' is selected. It shows the 'Data flow' field set to 'AlterRow_dataflow', the 'Run on (Azure IR)' dropdown set to 'AutoResolveIntegrationRuntime', and the 'Compute size' dropdown set to 'Small'. There are buttons for 'Open' and 'New'.

17. Validate the pipeline and click on debug.

18. Here our pipeline was executed successfully.

A screenshot of the pipeline run history. The top navigation bar includes 'Validate', 'Debug' (which is selected), 'Add trigger', and a toggle for 'Data flow debug' which is turned on. Below is a list of pipeline runs. The first run is shown with a green checkmark icon and the status 'Succeeded'. The 'Output' tab is selected. The pipeline run ID is 'eea24f20-f4c6-4a52-a74d-a65bbf201d95'. The table below shows one activity named 'Data flow1' with status 'Succeeded' and type 'Data flow'. The table has columns for Activity name, Activity status, and Activity type.

19. Now go to a destination and check the data.

Query 1 ×

Run Cancel query Save query Export data as Show all

Results Messages

Search to filter items...

Empld	EmpName	Gender	Country	Salary	Department
1	maheer	male	india	2000	HR
10	ABCDEF	male	India	1000	XTZXYZ
5	Ayesha	female	india	4000	IT
6	Pyarijaan	female	india	3000	IT
7	Mahaboob	male	india	5000	IT
8	Arfan	male	india	3000	IT

Cast Transformation

1. In this example we are using the below data.

The screenshot shows the Azure Data Explorer interface. On the left, there is a navigation pane with a tree view of datasets and tables. A table named 'Cast_src.csv' is selected, indicated by a checked checkbox. The main area displays the contents of this table:

id	name	gender	doj	salary
1	maheer	male	1/1/2022	2000
2	pradeep	male	1/2/2022	1000
3	wafa	male	24/5303	4000

2. Create a Dataset for this data.
3. Create a Dataflow.

The screenshot shows the Azure Data Factory portal. On the left, there is a sidebar with options like Pipelines, Datasets, and Data flows. The 'Data flows' option is selected. In the center, a new dataset is being created, with the name 'Cast_src' entered. Below the name, there is a dropdown menu with several options: Pipeline, Change Data Capture (preview), Dataset, Data flow, Power Query, and Copy Data tool. The 'Data flow' option is highlighted with a blue border.

4. Add a Source.

The screenshot shows the configuration screen for a data flow. At the top, there are buttons for 'Validate', 'Data flow debug' (which is turned on), and 'Debug Settings'. Below this, there is a section labeled 'source1' which contains a small icon of a cylinder with a circular arrow, indicating it's a data source. To the right of the source name, it says 'Columns: 0 total'. There is also a plus sign (+) button to add more sources.

5. Under source setting select the Source dataset.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * Learn more

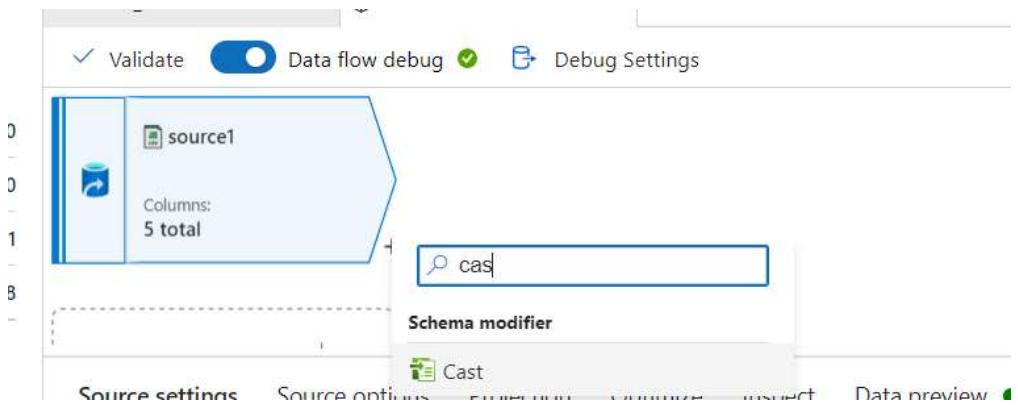
Description Reset

Source type * Dataset Inline

Dataset * Test connection Open New

Options Allow schema drift

6. Click on the plus symbol and click on the cast.



7. Under Cast Settings add the below column names and specify the Type and Format as shown below.

Cast settings Optimize Inspect Data preview

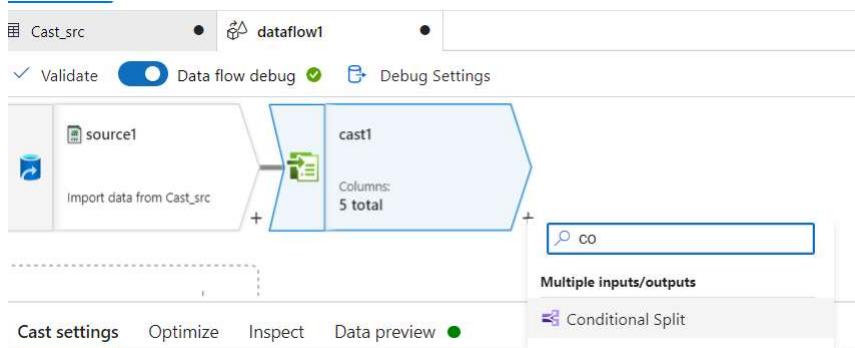
cast1

Cast columns to different types

source1

Column name	Type	Format
id	123 integer	Specify format
doj	date	dd/MM/yyyy
salary	123 integer	Specify format

8. Click on the plus symbol and click on the Conditional Split option as shown below.



9. Under the Conditional Split give the below conditions.

10. For the goodRow give this expression: !isError()

The 'Conditional split settings' dialog shows the following configuration:

- Output stream name:** split1
- Description:** Add cases to distribute the data in multiple groups
- Incoming stream:** cast1
- Split on:** First matching condition (radio button selected)
- Split condition:**

Stream names	Condition
goodRow	!isError()
badRow	Rows that do not meet any condition will use this output stream

11. Now add Two sinks as shown below.



12. For sink1 select the target dataset.

The 'Sink' configuration dialog shows the following settings:

- Sink:** (selected tab)
- Incoming stream:** split1@goodRow
- Sink type:** Dataset (selected)
- Dataset:** ds_sink_txt
- Buttons:** Test connection, Open, New

13. And give file type and name 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.

Clear the folder

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

Output to single file * **goodRow.csv**

Quote All

14. For sink2 select the target dataset.

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

Description **Export data to ds_sink_txt**

Incoming stream * **split1@badRow**

Sink type * **Dataset**

Dataset * **ds_sink_txt**

Test connection Open New

15. And give file type and name 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.

Clear the folder

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

Output to single file * **badRow.csv**

Quote All

16. Create a Pipeline, Drag and drop the Dataflow activity.

Activities Validate Debug Add trigger Data flow debug

Da

Move and transform

Copy data

Data flow

Data flow1

17. And select the Dataflow.

The screenshot shows the 'Settings' tab of a Data flow configuration. The 'Data flow' dropdown is set to 'cast_dataflow'. The 'Run on (Azure IR)' dropdown is set to 'AutoResolveIntegrationRuntime'. The 'Compute size' dropdown is set to 'Small'. There are 'Open' and 'New' buttons at the top right.

18. Now validate the pipeline and click on debug.

19. Here our pipeline was executed successfully.

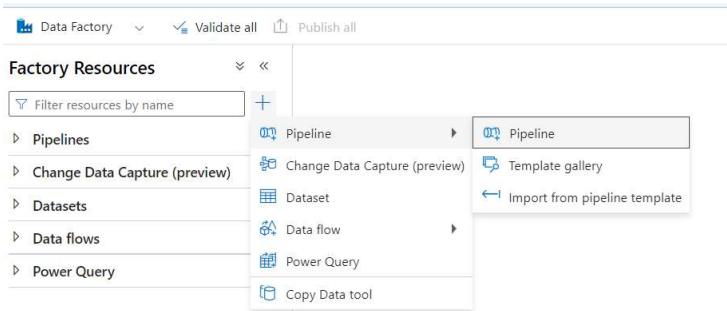
The screenshot shows the 'Output' tab of a pipeline run details page. The pipeline run ID is '6ef0cac3-fda1-4544-ad0d-caa16e86254d'. It shows one item with status 'Succeeded'. The activity name is 'Data flow1' and the type is 'Data flow'.

20. Now go to the destination and check the files and data.

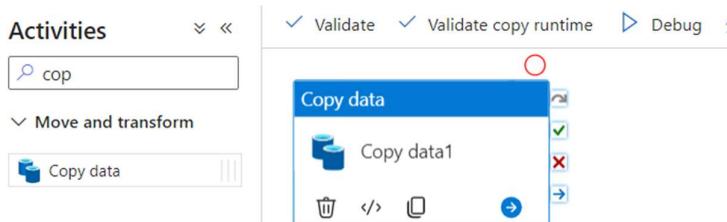
The screenshot shows a list of files in a storage account. There are two files: 'badRow.csv' and 'goodRow.csv'. Both files have a download icon next to them.

Web Page Usage

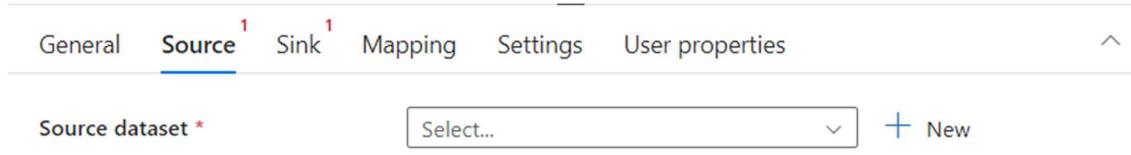
1. Create a Pipeline.



2. Drag and drop the copy data activity.



3. Under source click on the plus symbol.



4. Select the web table and click on continue.

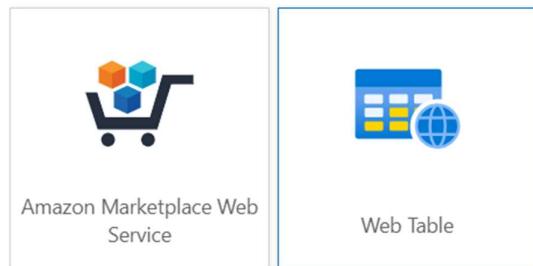
New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of data within a data store. [Learn more](#)

Select a data store



All Azure Database File Generic protocol NoSQL Services and



5. Give the name and click on New.

Set properties

Name

Linked service *

6. Under Integration Runtime click on New.

New linked service

 Web Table [Learn more](#)

Name *

Description

Connect via integration runtime * ⓘ

7. Select Selfhosted and click on continue.

Integration runtime setup

Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:

 Azure
Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.

 Self-Hosted
Use this for running activities in an on-premises / private network
[View more](#)

8. Give the Name and click on create.

Integration runtime setup

Private network support is realized by installing integration runtime to machines in the same on-premises network/VNET as the resource the integration runtime is connecting to. Follow below steps to register and install integration runtime on your self-hosted machines.

Name * ⓘ

Selfhosted

Description

Enter description here...

Type

Self-Hosted

9. Next, we need to Install Integration Runtime. In our case we are using Option 1 so click on the Option 1 link.

10. A file will be downloaded.

Edit integration runtime

[Settings](#) [Nodes](#) [Auto update](#) [Sharing](#) [Links](#)

Install integration runtime on Windows machine or add further nodes using the Authentication Key.

Name ⓘ

Selfhosted

Description

Option 1: Express setup

[Click here to launch the express setup for this computer](#)

Option 2: Manual setup

Step 1: [Download and install integration runtime](#)

Step 2: Use this key to register your integration runtime

Name

Authentication key

Key1

IR@d709a78d-0257-4137-ace8-943510d2c6aa@AzureDataFactoryTraing

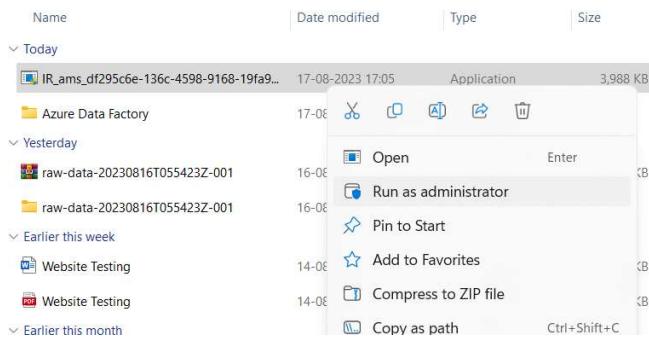


Key2

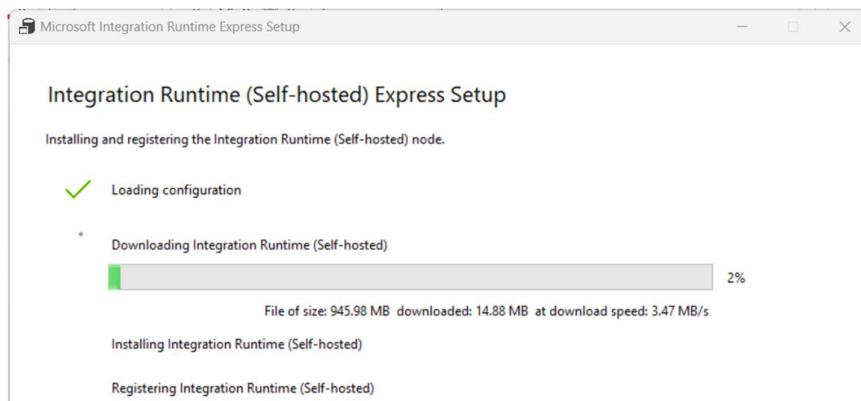
IR@d709a78d-0257-4137-ace8-943510d2c6aa@AzureDataFactoryTraing



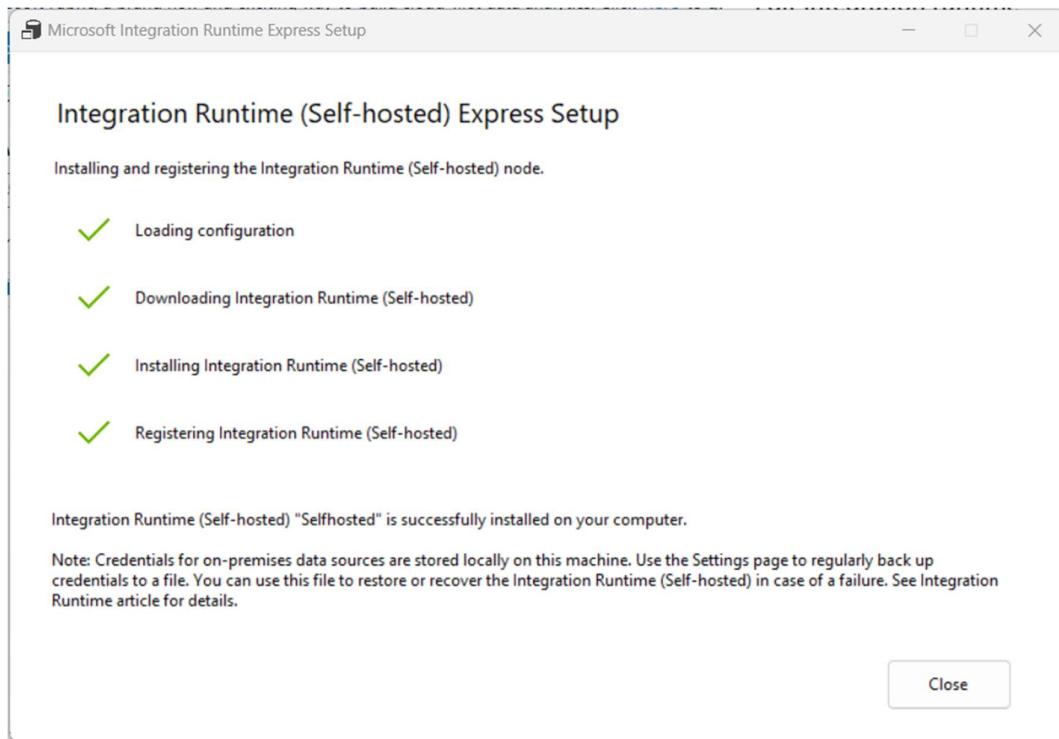
11. Right-click on the file and click on Run as administration as shown below.



12. It will start downloading and installing. It will take some time.



13. After Installing click on the close option.



14. Next in the Linked service select the Integration Runtime and give the below URL.

URL: <https://www.careerpower.in/states-and-capitals-of-india.html>

New linked service

 Web Table [Learn more](#)

Name *

WebTable

Description

Connect via integration runtime * ⓘ

 Selfhosted

URL *

<https://www.careerpower.in/states-and-capitals-of-india.html>

Authentication type *

Anonymous

Annotations

+ New

> Advanced ⓘ

 Connection successful

 Test connection

15. You will see it like this but click on Ok.

Set properties

Name

WebTable

Linked service *

WebTable

Connect via integration runtime * ⓘ

 Selfhosted

Index

Loading failed



 Failed More

Edit

> Advanced

16. Click on Open.

The screenshot shows the 'Source' tab selected in a pipeline configuration interface. The 'Source dataset' dropdown is set to 'WebTable'. Below it, there are buttons for 'Open', 'New', 'Preview data', and 'Learn more'. A section for 'Additional columns' is present with a 'New' button. The top navigation bar includes tabs for General, Source, Sink, Mapping, Settings, and User properties.

17. Under Index give the value as 0.

The screenshot shows the 'Connection' tab selected for a linked service named 'WebTable'. It includes fields for 'Test connection', 'Integration runtime' (set to 'Selfhosted'), 'Path' (empty), and 'Index' (set to 0). There are buttons for 'Edit', 'New', and 'Learn more'.

18. Now go back to the pipeline.

19. Under sink click on new.

The screenshot shows the 'Sink' tab selected in a pipeline configuration interface. The 'Sink dataset' dropdown is set to 'Select...' and has a 'New' button next to it. The top navigation bar includes tabs for General, Source, Sink, Mapping, Settings, and User properties.

20. Select the Azure Blob Storage and click on continue then select Delimited Text and click on continue.

21. Next give the Name, select the Linked service, and give the File path, Name.

Set properties

The screenshot shows the 'Set properties' dialog for a sink dataset named 'Webpage'. It includes fields for 'Name' (set to 'Webpage'), 'Linked service' (set to 'AzureBlobStorage_txt'), 'File path' (set to 'f-demo / Output / File name'), 'First row as header' (checked), and 'Import schema' (radio button selected for 'From connection/store').

22. We have configured our target part.

General Source **Sink** Mapping Settings User properties

Sink dataset * Open New Lea

Copy behavior

Max concurrent connections

✓ Validate ▶ Debug ⚡ Add trigger

Copy data Copy data1

Parameters Variables Settings **Output**

Pipeline run ID: 4b50e69d-3f8c-423f-bb68-cdec3332500b **Pipeline**

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

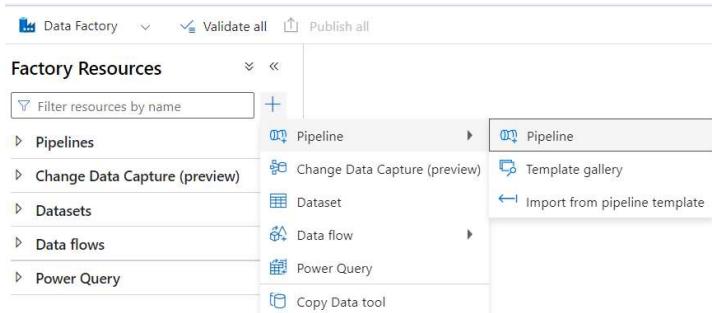
All status

Showing 1 - 1 of 1 items

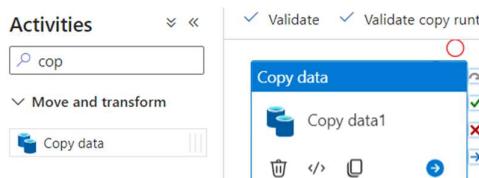
Activity name	Activity status	Activity type	Run start
Copy data1	<input checked="" type="checkbox"/> Succeeded	Copy data	8/17/2023, 10:30:30 AM

Rest Pagination

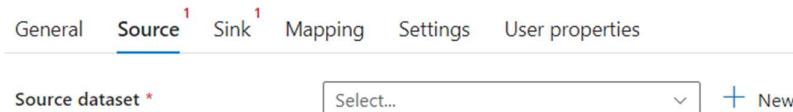
1. Create a Pipeline.



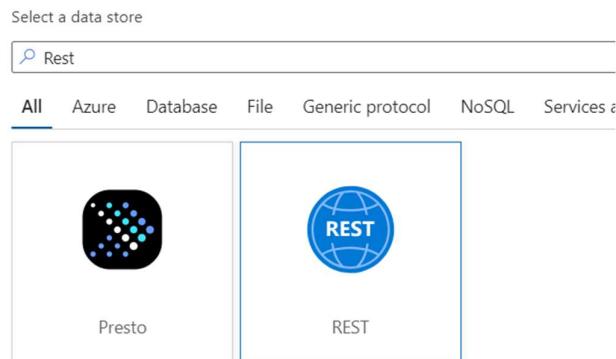
2. Drag and drop the copy data activity as shown below.



3. Under the source click on the plus symbol.



4. Select the Rest and click on Continue as shown below.



5. Give the Name and click on New.



6. Give the Below URL and click on Create.

URL: <https://pokeapi.co/api/v2/pokemon?limit=20&offset={offset}>

New linked service

 REST [Learn more](#)

Name *

RestServ

Description

Connect via integration runtime * ①

AutoResolveIntegrationRuntime

Base URL *

https://pokeapi.co/api/v2/pokemon?limit=20&offset={offset}

 Information will be sent to the URL specified. Please ensure you trust the URL entered.

Authentication type *

Anonymous

Server Certificate Validation ①

Enable Disable

Auth headers ①

 New

Annotations

 New

 Connection successful

 Test connection

 Create

 Cancel

7. Next click on Ok.

Set properties

Name

RestPagination

Linked service *

RestServ



> Advanced

8. Under Source, in the Pagination Rule set up the below Rule with the below Range.

The screenshot shows the 'Additional headers' and 'Pagination rules' sections. In the 'Additional headers' section, there is a 'New' button. In the 'Pagination rules' section, there is a 'New' button and a 'Delete' button. A table is shown with columns 'Name' and 'Value'. The 'Value' column contains a dropdown menu set to 'Range' with input fields for '0', '1281', and '20'. Below the table is a link 'Add dynamic content [Alt+Shift+D]'.

9. Under sink select the Destination.

The screenshot shows the 'Sink' tab selected. It includes fields for 'Sink dataset *' (set to 'ds_sink_txt'), 'Copy behavior' (set to 'Select...'), and 'Max concurrent connections' (set to 1).

10. Under mapping click on Import Schema.

11. Then select the reference.

12. Under Name delete every column except below two.

The screenshot shows the 'Mapping' tab selected. It includes a 'Import schemas' section with a 'New mapping' button, a 'Collection reference' dropdown set to '\${'results'}', and a 'Map complex values to string' checkbox. Below is a table for mapping schema:

Name	Type	Collection reference	Column name	Type	Include
['results'][0]['name']	ANY any	→	name	abc String	<input checked="" type="checkbox"/>
['results'][0]['url']	ANY any	→	url	abc String	<input checked="" type="checkbox"/>

13. Now validate the pipeline and click on Debug.

14. Here our pipeline was executed successfully.

The screenshot shows the pipeline run status. At the top, there are buttons for 'Validate', 'Debug', and 'Add trigger'. Below is a list of activities: 'Copy data' (green checkmark) and 'Copy data1' (red X). The 'Output' tab is selected, showing the pipeline run ID: 1187d672-7ec0-419b-b2a2-47f73467d51c. The 'Pipeline' tab is also visible. The activity table shows 'Copy data1' with status 'Succeeded', activity type 'Copy data', and run start '8/18/2023, 11:23:00 AM'.

15. Go to Destination and check the Data.

FilterOutput/data_caec6c7e-12d2-4156-bef3-e502c5

Blob

Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

```
1 name,url
2 "bulbasaur","https://pokeapi.co/api/v2/pokemon/1/"
3 "spearow","https://pokeapi.co/api/v2/pokemon/21/"
4 "zubat","https://pokeapi.co/api/v2/pokemon/41/"
5 "poliwhirl","https://pokeapi.co/api/v2/pokemon/61/"
6 "magnemite","https://pokeapi.co/api/v2/pokemon/81/"
7 "electrode","https://pokeapi.co/api/v2/pokemon/101/"
8 "starmie","https://pokeapi.co/api/v2/pokemon/121/"
9 "kabutops","https://pokeapi.co/api/v2/pokemon/141/"
10 "sentret","https://pokeapi.co/api/v2/pokemon/161/"
11 "ampharos","https://pokeapi.co/api/v2/pokemon/181/"
12 "unown","https://pokeapi.co/api/v2/pokemon/201/"
13 "piloswine","https://pokeapi.co/api/v2/pokemon/221/"
14 "miltank","https://pokeapi.co/api/v2/pokemon/241/"
15 "poochyena","https://pokeapi.co/api/v2/pokemon/261/"
16 "kirlia","https://pokeapi.co/api/v2/pokemon/281/"
17 "delcatty","https://pokeapi.co/api/v2/pokemon/301/"
18 "wailord","https://pokeapi.co/api/v2/pokemon/321/"
19 "corphish","https://pokeapi.co/api/v2/pokemon/341/"
20 "snorunt","https://pokeapi.co/api/v2/pokemon/361/"
21 "latios","https://pokeapi.co/api/v2/pokemon/381/"
```

Text

Preview

Rank Transformation

1. In this example we are using the below data.

Employee.csv ...

Blob

Upload Change access level ...

Dynamic_pipelinetotalsale

FilterOutput

Output

output_1

output_2

output_3

Output1

Switch1

Cast_src.csv

Department.csv

Employee.csv

FilterOutput

gender.csv

input_text.csv

Mobile_det.json

...

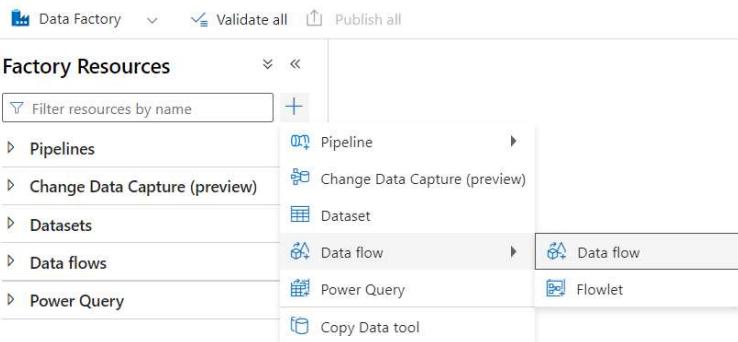
Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

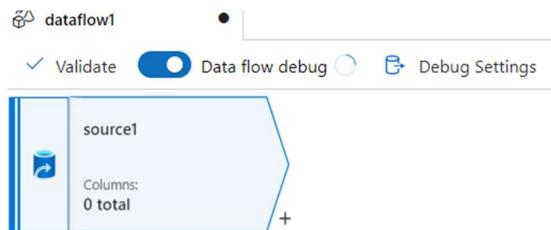
empid	name	Gender	country	salary	department
1	maheer	male	india	2000	HR
2	asmin	female	india	3000	HR
3	wafa	male	india	1000	HR
4	Sarfraz	male	india	2000	Payroll
5	Ayesha	female	india	4000	IT
6	Pyarijaan	female	india	3000	IT
7	Mahaboob	male	india	5000	IT
8	Arfan	male	india	3000	IT
9	shabbir	male	india	4000	HR
10	Afrin	female	india	1000	Payroll
11	Shahin	female	india	5000	Payroll

Edit

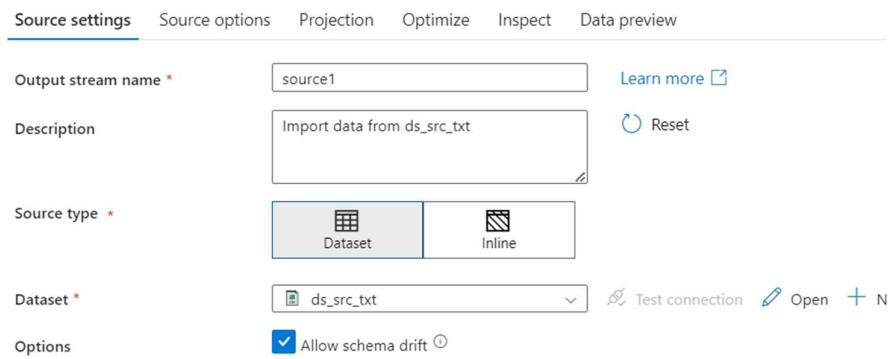
2. Create a Dataflow.



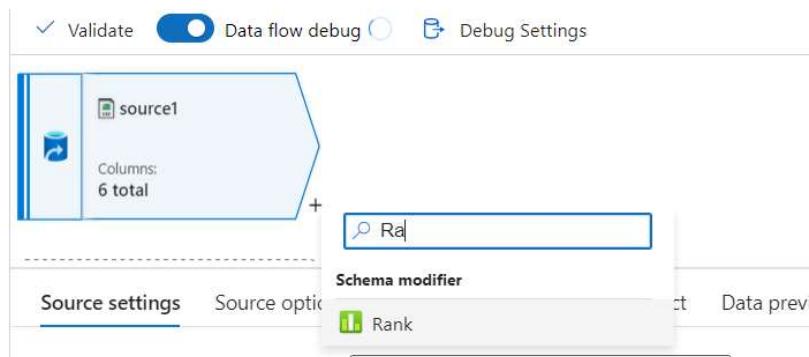
3. Add a Source.



4. Under the Source setting select the Source dataset.



5. Click on the Plus symbol and click on Rank as shown below.



6. Under Rank setting, select the Options, give the Rank column name, and set the below condition.

Rank settings Optimize Inspect Data preview

Output stream name * rank1 Learn more ↗

Description Ranking rows on columns 'salary' Reset

Incoming stream * source1

Options * Case insensitive Dense ⓘ

Rank column * Ranking

Sort conditions * source1's column Order abc salary Descending

7. Click on the plus symbol and click on Sink.



8. Select the Destination dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

Output stream name * sink1 Learn more ↗

Description Export data to ds_sink_txt Reset

Incoming stream * rank1

Sink type * Dataset Inline Cache

Dataset * ds_sink_txt Test connection Open New

This sink currently has Single partition set in Optimize. This will make your data flow execution longer than current partitioning.

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

Sink Settings Errors Mapping Optimize Inspect Data preview ●

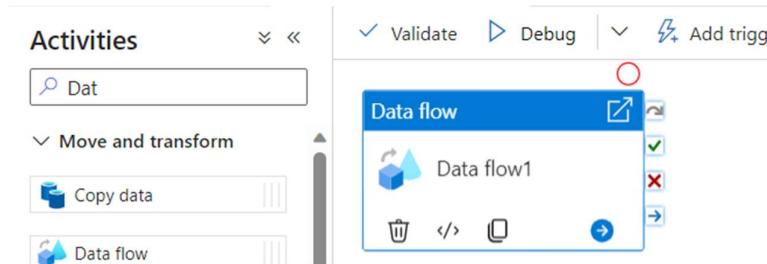
ⓘ This sink currently has Single partition set in Optimize. This will make your data flow execution longer than current partitioning.

Clear the folder

File name option * Output to single file

Output to single file * ⓘ Rank.csv

10. Create a Pipeline and drag and drop the Data Flow Activity as shown below.



11. Under settings select Dataflow.

General **Settings** Parameters User properties

Data flow * Open +

Run on (Azure IR) *

Compute size *

✓ Validate ⚡ Debug ⚡ Add trigger Data flow debug ✓

Data flow ✓

Rank_dataflow

AutoResolveIntegrationRuntime

Small

✓ Validate ⚡ Debug ⚡ Add trigger Data flow debug ✓

Parameters Variables Settings **Output**

Pipeline run ID: 2e11efd3-8bc9-462d-b7fc-6f515937df95 ↻ ⓘ

All status ▼

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run count
Data flow1	✓ Succeeded	Data flow	8/2

14. Now go to Destination and check the Data.

The screenshot shows the Azure Blob Storage interface for the file 'FilterOutput/Rank.csv'. The file is a CSV file containing the following data:

empid	name	Gender	country	salary	department	Ranking
4	Sarfaraj	male	india	2000	Payroll	4
1	maheer	male	india	2000	HR	4
10	Afrin	female	india	1000	Payroll	5
3	wafa	male	india	1000	HR	5
8	Arfan	male	india	3000	IT	3
6	Pyarijan	female	india	3000	IT	3
2	asmin	female	india	3000	HR	3
11	Shahin	female	india	5000	Payroll	1
7	Mahaboo	male	india	5000	IT	1
9	shabbir	male	india	4000	HR	2
5	Ayesha	female	india	4000	IT	2

Parse Transformation

1. In this we are using the below table from the SQL database.

Code: Create table employees (

```

empId int,
empName varchar(100),
skill varchar(100),
[address] varchar(100)
)
insert into employees
values(1,'Mahesh','.net|SQL|Azure','{"City":"Hyderabad","Country":"India"}');
insert into employees
values(2,'Hemanth','Java|SQL|AWS','{"City":"Banglore","Country":"India"}');
Select * from employees

```

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

```

1  create table employees (
2    empId int,
3    empName varchar(100),
4    skill varchar(100),
5    [address] varchar(100)
6  )
7  insert into employees values(1,'Mahesh','.net|SQL|Azure','{"City":"Hyderabad","Country":"India"}');
8  insert into employees values(2,'Hemanth','Java|SQL|AWS','{"City":"Banglore","Country":"India"}');
9  Select * from employees

```

The results pane shows the data inserted into the 'employees' table:

empId	empName	skill	address
1	Mahesh	.net SQL Azure	{"City":"Hyderabad","Country":"India"}
2	Hemanth	Java SQL AWS	{"City":"Banglore","Country":"India"}

2. Create a Dataset for the Above table.

Set properties

Name
Emp_Src

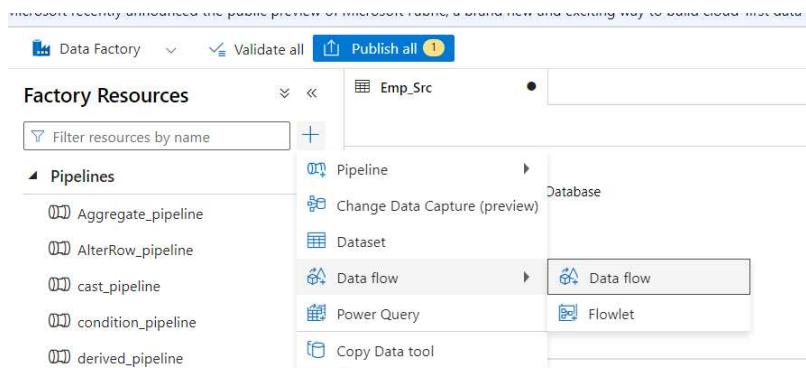
Linked service *
AzureSqlDatabase

Table name
dbo.employees

Edit

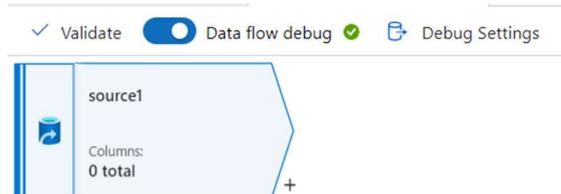
Import schema
 From connection/store None

3. Create a Dataflow.



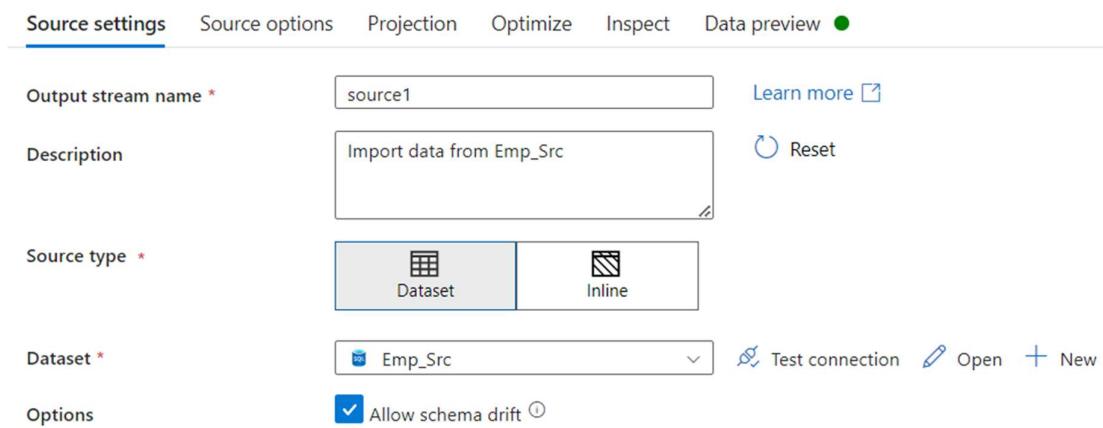
The screenshot shows the Azure Data Factory studio interface. At the top, there are buttons for 'Data Factory' (dropdown), 'Validate all', and 'Publish all' (with a '1' notification). Below this is a 'Factory Resources' section with a dropdown set to 'Emp_Src'. Under 'Pipelines', several pipelines are listed: 'Aggregate_pipeline', 'AlterRow_pipeline', 'cast_pipeline', 'condition_pipeline', and 'derived_pipeline'. To the right of these, a context menu is open over a 'Data flow' item, showing options like 'Pipeline', 'Change Data Capture (preview)', 'Dataset', 'Data flow' (which is selected and highlighted in blue), 'Power Query', and 'Flowlet'. A 'Copy Data tool' option is also visible at the bottom of the menu.

4. Add a Source.



The screenshot shows the 'source1' configuration screen for a data flow. At the top, there are buttons for 'Validate', 'Data flow debug' (which is turned on), 'Debug Settings', and a 'Source settings' tab (which is selected). Below this, the 'source1' component is shown with a preview icon and the text 'Columns: 0 total'. A '+' button is available to add more sources.

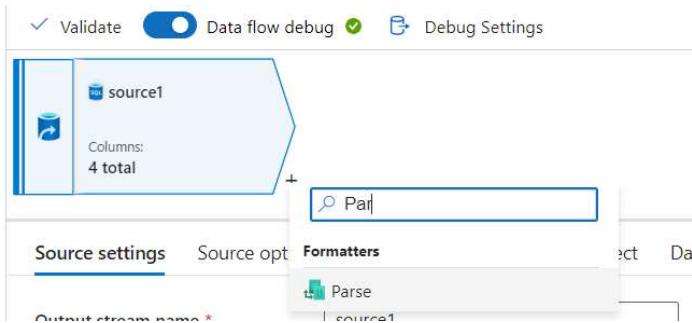
5. Select the Source Dataset.



The screenshot shows the 'Source settings' configuration screen. The tabs at the top are 'Source settings' (selected), 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview'. The 'Source settings' tab has the following fields:

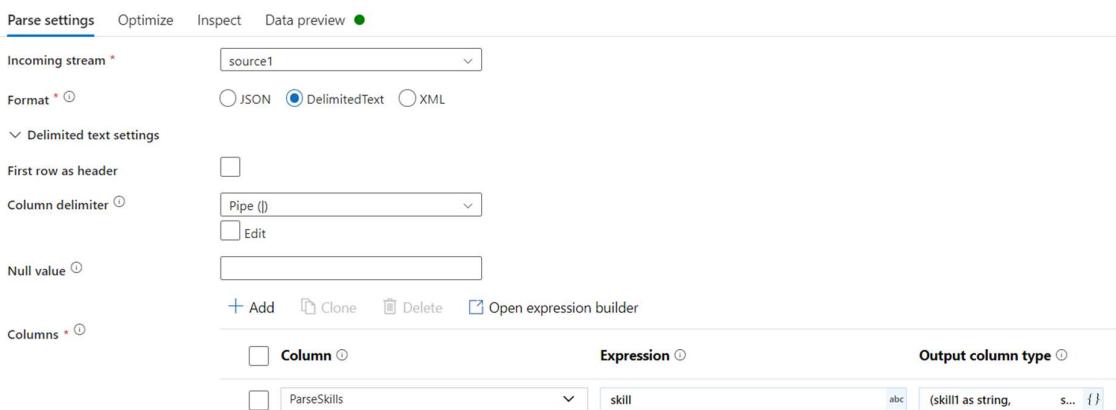
- Output stream name *: 'source1'
- Description: 'Import data from Emp_Src'
- Source type *: A button labeled 'Dataset' is selected, with 'Inline' as an alternative.
- Dataset *: A dropdown set to 'Emp_Src' with buttons for 'Test connection', 'Open', and 'New'.
- Options: A checkbox for 'Allow schema drift' is checked.

6. Click on the plus symbol and click on Parse.

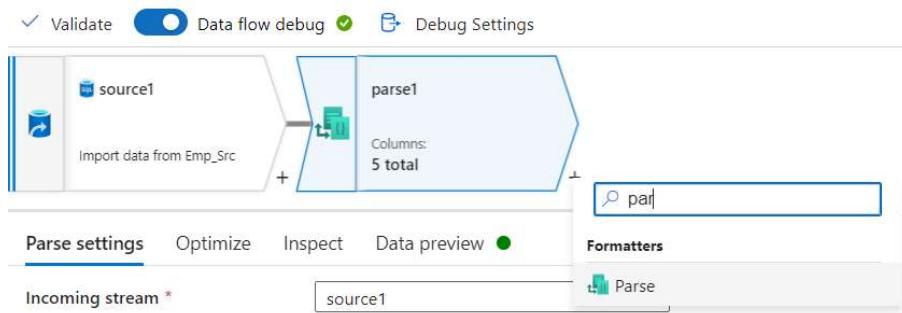


7. Under the Parse setting, set the below properties and for the Output, column type gives the below expression.

Exp: (skill1 as string, skill2 as string, skill3 as string)



8. Click on the Plus symbol and click on Parse again.



9. Under the Parse setting, set the below properties and for the Output, column type gives the below expression.

Exp: (City as string, Country as string)

Parse settings Optimize Inspect Data preview ●

Output stream name * parse2 [Learn more](#)

Description Creating/updating the columns 'empid, empName, skill, address, ParseSkills, ParseAddress'

Incoming stream * parse1

Format * JSON DelimitedText XML

JSON settings

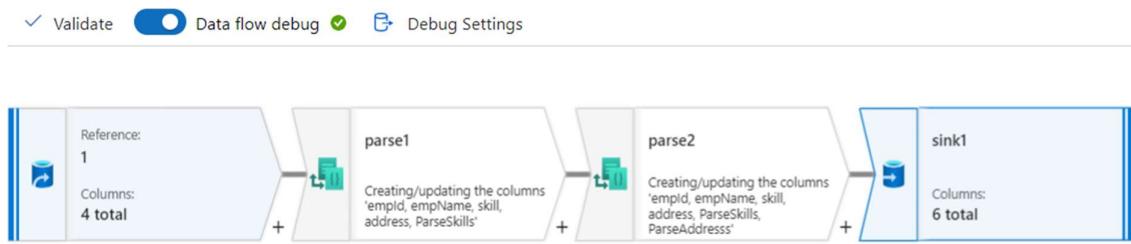
Document form Single document Document per line Array of documents

+ Add Clone Delete Open expression builder

Columns * Column Expression Output column type

<input type="checkbox"/> ParseAddress	address	abc	(City as string, Count... {})
---------------------------------------	---------	-----	-------------------------------

10. Next Add a sink.



11. Select the target dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview ●

Description Export data to ds_sink_txt [Reset](#)

Incoming stream * parse2

Sink type * Dataset Inline Cache

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

12. Under settings select the file option and file name as shown below.

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

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

Clear the folder

File name option * Output to single file

Output to single file * Parse.csv

Quote All

13. Under mapping delete the unnecessary fields and add the below fields 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. The recommended setting is to use a range partition.

Options

Skip duplicate input columns (i)

Skip duplicate output columns (i)

Auto mapping (i) Reset Add mapping Delete

Output format 7

Input columns	Output columns
123 empId	empId
abc empName	empName
abc ParseSkills.skill1	skill1
abc ParseSkills.skill2	skill2
abc ParseSkills.skill3	skill3
abc ParseAddresss.City	City
abc ParseAddresss.Country	Country

14. Create a Pipeline and drag and drop the Data flow.

Activities

Da

Move and transform

Copy data

Data flow

Validate Debug Add tr

Data flow

Data flow1

15. Select the Dataflow.

General **Settings** Parameters User properties

Data flow *

Parse_dataflow

Open +

Run on (Azure IR) * (i)

AutoResolveIntegrationRuntime

Compute size * (i)

Small

16. Validate the pipeline and click on Debug.
 17. Here our pipeline was executed successfully.

empId	empName	skill1	skill2	skill3	City	Country
1	Mahesh	.net	SQL	Azure	Hyderabad	India
2	Hemanth	Java	SQL	AWS	Banglore	India

18. Go to Destination and check the target Data.

Assert Transformation

- For this we are using the below Emp data.

The screenshot shows the Azure Blob storage interface for the file 'Emp.csv'. The left sidebar lists various files and folders: FilterOutput, Output, output_1, output_2, output_3, Output1, Switch1, Cast_src.csv, Department.csv, Emp.csv (which is selected), Employee.csv, and FilterOutput. The right panel displays the contents of 'Emp.csv' in a table format:

empid	name	Gender	doj	deptid
1	maheer	male	20210510	3
2	asmin	female	20200214	2
3	wafa	male	20210425	2
4	Sarfaraj	male	20180312	4
5	Ayesha	female	20210819	1
6	Pyarijanan	female	201uik1016	2
7	Mahaboob	male	2019gyui071	3
7	Arfan	male	20170615	1

- And also, below Department data.

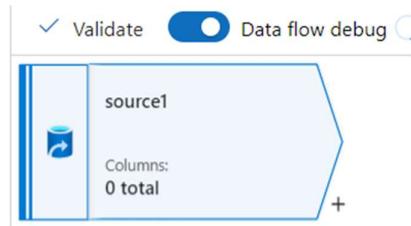
The screenshot shows the Azure Blob storage interface for the file 'Department.csv'. The left sidebar lists the same set of files as before. The right panel displays the contents of 'Department.csv' in a table format:

depid	depname
1	IT
2	HR
3	payroll

- Create a Dataset for both files.
- Create a Dataflow.

The screenshot shows the 'Factory Resources' blade in the Azure Data Factory interface. On the left, there is a sidebar with options: Pipelines, Change Data Capture (preview), Datasets, Data flows, and Power Query. The main area is titled 'Emp_Set' and contains a list of resources: Pipeline, Change Data Capture (preview), Dataset, Data flow (which is highlighted with a yellow box), Power Query, Flowlet, and Copy Data tool.

5. Add a Source.



6. Select the Emp source Dataset.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * Learn more

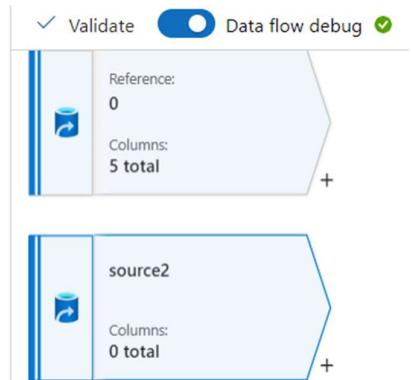
Description Reset

Source type * Dataset Inline

Dataset * Emp_Set Test connection Open New

Options Allow schema drift

7. Add another Source.



8. Select the Department source dataset.

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * Learn more

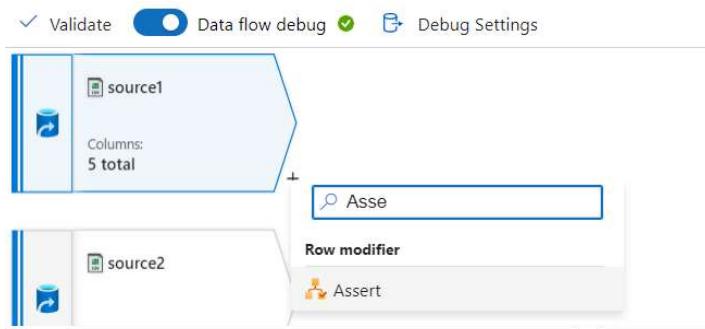
Description Reset

Source type * Dataset Inline

Dataset * Dept_Dataset Test connection Open New

Options Allow schema drift

9. Click on the plus symbol and click on Assert as shown below.



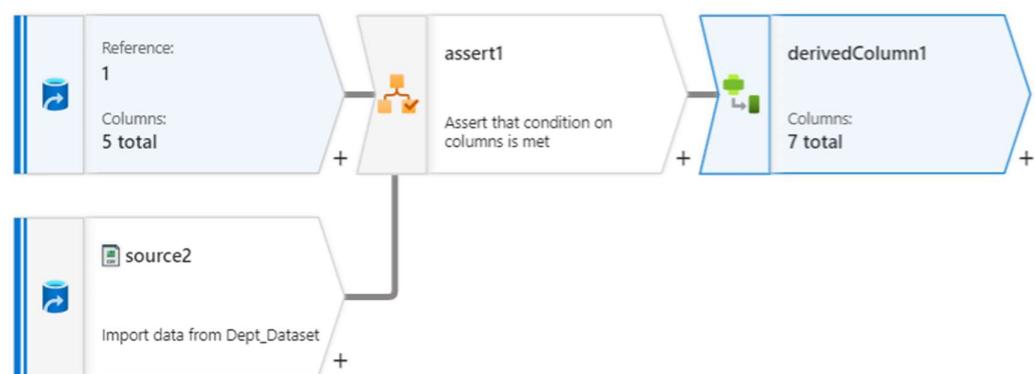
10. Under Assert settings, set the below properties then add below Asserts.
11. For the assertDoj give the below expression.

Exp: !isNull(toDate(doj, 'yyyyMMdd'))

12. For the assertDeptExists select the fields are Deptid and Depid

The screenshot shows the 'Assert settings' tab in the Data Flow Editor. It includes sections for 'Incoming stream', 'Additional streams', 'Fail data flow', and 'Asserts'. Three asserts are listed under 'Asserts': 'assertDoj' (Expect true, Expression: !isNull(toDate(doj, 'yyyyMMdd'))), 'assertempid' (Expect unique, Expression: abc.empid), and 'assertDeptExists' (Expect exists, Expression: abc.depid).

13. Add the Derived column.



14. Under the Derived column's settings add the below columns and conditions.

Derived column's settings Optimize Inspect Data preview ●

Output stream name * Learn more ↗

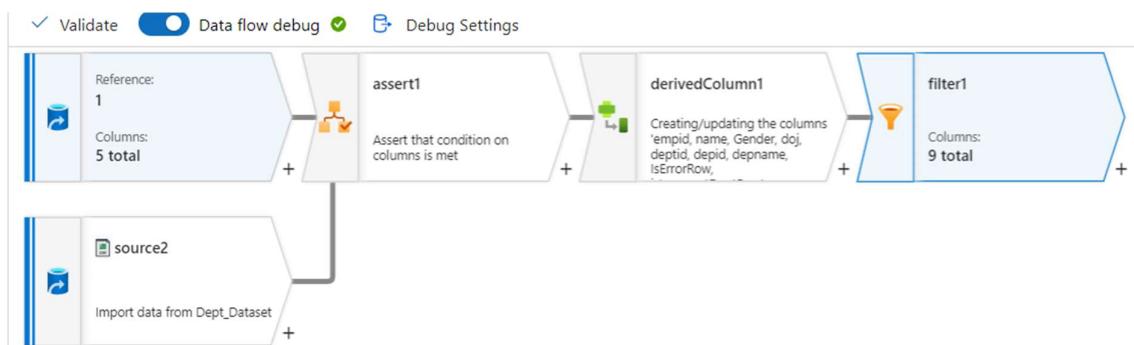
Description ⌂ Reset

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

Columns * ⓘ

Column	Expression
<input type="checkbox"/> IsErrorRow	<input type="text" value="isError()"/> ⌂ + ⌂
<input type="checkbox"/> IsIncorrectDeptRow	<input type="text" value="hasError('assertDeptExists')"/> ⌂ + ⌂

15. Add a Filter.



16. Under filter settings give the below Filter on Expression.

Exp: IsErrorRow == false()

Filter settings Optimize Inspect Data preview ●

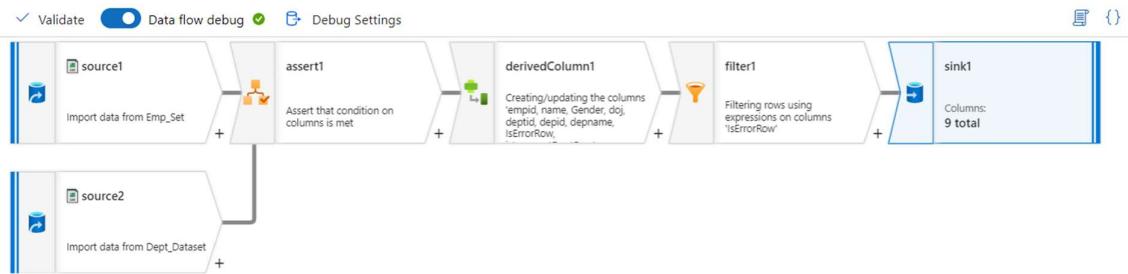
Output stream name * Learn more ↗

Description ⌂ Reset

Incoming stream *

Filter on * ⌂

17. Add Sink.



18. Select the Destination Dataset.

Sink Settings Errors Mapping Optimize Inspect Data preview

Incoming stream * filter1

Sink type * Dataset

Dataset * ds_sink_txt Test connection Open New

Skip line count

Options Allow schema drift ⓘ

19. Under settings select the File Option and File Name.

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

Clear the folder

File name option * Output to single file

Output to single file * ⓘ Assert.csv

Quote All ⓘ

20. Under Mapping keep the below columns and delete others.

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

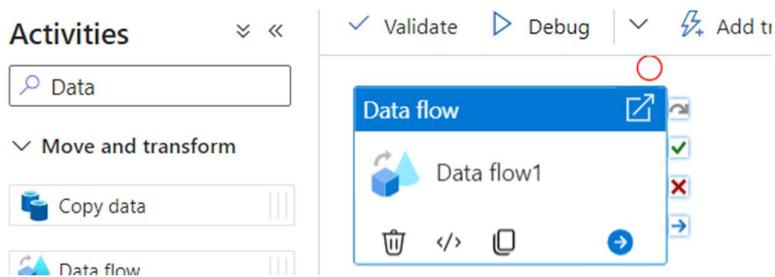
i This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The recommended setting is Use

Options	<input checked="" type="checkbox"/> Skip duplicate input columns ⓘ
<input type="checkbox"/> Auto mapping ⓘ	<input checked="" type="checkbox"/> Skip duplicate output columns ⓘ
<input type="checkbox"/> Reset	<input type="checkbox"/> Add mapping
<input type="checkbox"/> Delete	Output format
	5 mapping

Input columns	Output columns
empid	empid
name	name
Gender	Gender
doj	doj
deptid	deptid

21. Create a Pipeline

22. Drag and drop the Data flow.



23. Under setting select the Dataflow.

General **Settings** Parameters User properties

Data flow *
Assert_dataflow

Run on (Azure IR) *

Compute size *

24. Validate the pipeline and click on Debug.
25. Here our pipeline is Executed successfully.

✓ Validate ▶ Debug ↴ Add trigger Data flow debug

Data flow

Parameters Variables Settings **Output**

Pipeline run ID: cca94064-3679-4b3a-95b9-f2ad39179431

All status

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type
Data flow1	Succeeded	Data flow

26. Go to Destination and check the Data.

The screenshot shows the Azure Blob Storage interface for the file 'FilterOutput/Assert.csv'. The file is currently being edited. The table contains the following data:

empid	name	Gender	doj	deptid
5	Ayesha	female	20210819	1
2	asmin	female	20200214	2
1	maheer	male	20210510	3
3	wafa	male	20210425	2

Below the table is a blue 'Edit' button.