

Context

Create SQL Server	2 – 4
Create SQL database	4 – 6
Create a Linked Services	6 – 7
Create Dataset	8 – 9
If Condition Activity	9 – 14
For Each Activity	15 – 18
Filter Activity	18 – 21
Set Variable Activity	21 – 23
Append Variable	23 – 25
Until Activity	26 – 30
Web Activity	31 – 33
Switch Activity	33 – 38
Lookup Activity	39 – 45

Create SQL Server

1. On the Home page click on SQL servers.



2. Click on Create SQL Server as shown below.

3. Under basic give the below properties as shown below.

Basics Networking Additional settings Tags Review + create

SQL database server is a logical container for managing databases and elastic pools. Complete the Basic tab, then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Resource group * [Create new](#)

Server details

Enter required settings for this server, including providing a name and location.

Server name * .database.windows.net

Location *

4. Scroll down and give the login name and password then click on the Next option.

Authentication

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Azure AD authentication [Learn more](#) using an existing Azure AD user, group, or application as Azure AD admin [Learn more](#), or select both SQL and Azure AD authentication.

Authentication method Use only Azure Active Directory (Azure AD) authentication Use both SQL and Azure AD authentication Use SQL authentication

Server admin login *

Password *

Confirm password *

[Review + create](#) | [Next : Networking >](#)

5. Set the below properties and click on Review+Create

Create SQL Database Server

Microsoft

Basics Networking Additional settings Tags Review + create

Configure networking access for your server.

Firewall rules

Allow Azure services and resources to access this server

Yes No

[Review + create](#) [< Previous](#) [Next : Additional settings >](#)

6. Next click on Create.

Basics Networking Additional settings Tags Review + create

Product details

SQL Database Server
by Microsoft
[Terms of use](#) | [Privacy policy](#)

Estimated cost per month
No additional charges

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Market frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and third-party offerings. For additional details see [Azure Marketplace Terms](#).

Basics

Subscription	Microsoft Partner Network
Resource group	ADFResource
Server name	adfsqservertrail
Authentication method	SQL authentication
Server admin login	adminsql
Location	East US

Networking

[Create](#) [< Previous](#) [Download a template for automation](#)

7. After some time you will see a screen like below.

The screenshot shows the Microsoft Azure Deployment Overview page for a deployment named "Microsoft.SQLServer.createServer_fca55494a9eb4c27a5fcf5aee35d70f". The status is "Your deployment is complete". Deployment details include: Deployment name: Microsoft.SQLServer.createServer_fca55494a9eb4c27a5fcf5aee35d70f, Start time: 7/20/2023, 1:25:21 PM, Subscription: Microsoft Partner Network, Resource group: ADFResource. Correlation ID: 5c1fa646-8fc8-4155-9ebf-9db2786e9654.

Create SQL database

1. On the Home page click on SQL database.

The screenshot shows the Microsoft Azure Home page. Under "Azure services", there is a "Create a resource" button and icons for SQL databases, Storage accounts, Data factories, App registrations, Quickstart center, Virtual machines, App Services, Azure Cosmos DB, and More services.

2. Next click on create sql database.

The screenshot shows the Microsoft Azure SQL databases blade. It displays a message: "No SQL databases to display. Try changing or clearing your filters." Below this is a "Create SQL database" button.

3. Give the Database name as sqldb and select the server that we created before then set the properties as shown below. And click on Review+Create.

The screenshot shows the "Create SQL Database" blade. It includes sections for "Subscription" (set to Microsoft Partner Network) and "Resource group" (set to ADFResource). In the "Database details" section, the "Database name" is set to "sqldb" and the "Server" is set to "adfsqservertrail (East US)". Other options include "Want to use SQL elastic pool?" (No selected), "Workload environment" (Development selected), and a note about default configurations. At the bottom are "Review + create" and "Next : Networking >" buttons.

4. Next click on Create.
5. After some time you will see the below screen as shown.

The screenshot shows the Microsoft Azure portal's 'Deployment' overview page. The title bar reads 'Microsoft.SQLDatabase.newDatabaseExistingServer_02f8fa5d79fd47e5 | Overview'. The main content area displays a green checkmark icon and the message 'Your deployment is complete'. Below this, deployment details are listed: Deployment name: Microsoft.SQLDatabase.newDatabaseExistingServer_0..., Start time: 7/20/2023, 1:31:52 PM, Subscription: Microsoft Partner Network, Correlation ID: a1e52811-272f-4ad2-af61-b2d67b4ef720, and Resource group: ADFResource.

6. Now click on the SQL Database that we created just now.
7. Go to Query editor, give login credentials that we created in the sql server click on Ok.
8. If it asks to allow access to an IP address allow it.

The screenshot shows the 'sqldatabase (sampleadmins1/sqldatabase) | Query editor (preview)' page. The left sidebar includes links for Overview, Activity log, Tags, Diagnose and solve problems, Getting started, and Query editor (preview). The main area displays a 'Welcome to SQL Database Query Editor' message and two authentication options: 'SQL server authentication' (Login: admins1, Password: [redacted]) and 'Active Directory authentication' (Continue as Lacchi.Balaji@gisul.co.in). A large 'OK' button is at the bottom right.

9. A Query editor will open so that we can create tables and export tables also.
10. I have created a table using sql query.

The screenshot shows the 'Query 1' editor window with the following SQL code:

```

CREATE TABLE TotalSale (
    id int NOT NULL,
    SalePersonName varchar(100) NULL,
    ProductName varchar(100) NULL,
    ItemsSold int NULL,
    SoldPrice int NULL,
    SoldDate Date,
    Country varchar(100) NULL,
    Region varchar(100) NULL
)
  
```

The left sidebar shows the database structure with 'Tables' selected. The bottom navigation bar includes 'Results' and 'Messages' tabs.

11. Below is the table I have created.

The screenshot shows the Azure Data Studio interface. On the left, there's a tree view of the database schema under 'sqldatabase (adminsql1)'. It includes 'Tables' (with 'dbo.TotalSale' expanded), 'Views', and 'Stored Procedures'. The main area is titled 'Query 1' and contains a table with the following data:

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

Create a Linked Services

1. Go to Azure Data Factory.
2. Go to the Manage tab and select Linked Services then click on the plus symbol to create.

The screenshot shows the 'Linked services' page in the Azure Data Factory portal. The left sidebar has 'General' selected. The main area shows a table with one item:

Name	Type
AzureBlobStorage_txt	Azure

3. Select the SQL Azure Database and click on Continue.

The screenshot shows the 'Data store' selection screen in the Azure Data Factory portal. The left sidebar has 'Compute' selected. The main area shows a grid of database options:

All	Azure	Database	File	Generic protocol	NoSQL	Services and apps
Amazon RDS for SQL Server	Azure Cosmos DB for NoSQL	Azure Database for MySQL				
Azure Database for PostgreSQL	Azure SQL Database	Azure SQL Database Managed Instance				
MySQL	Oracle	SQL Server				

At the bottom, there are 'Continue' and 'Cancel' buttons.

4. Give the Name and Runtime.

New linked service

Azure SQL Database [Learn more](#)

Name *

Description

Connect via integration runtime * ⓘ

AutoResolveIntegrationRuntime

5. And give the below properties and test the connection then click on create.

Connect via integration runtime

AutoResolveIntegrationRuntime

Connection string **Azure Key Vault**

Account selection method ⓘ

From Azure subscription Enter manually

Azure subscription

Select all

Server name *

sampleadmins1sql

Database name *

sqldatabase

Authentication type *

SQL authentication

User name *

admins1sql1

Add dynamic content [Alt+Shift+D]

Password **Azure Key Vault**

Password *

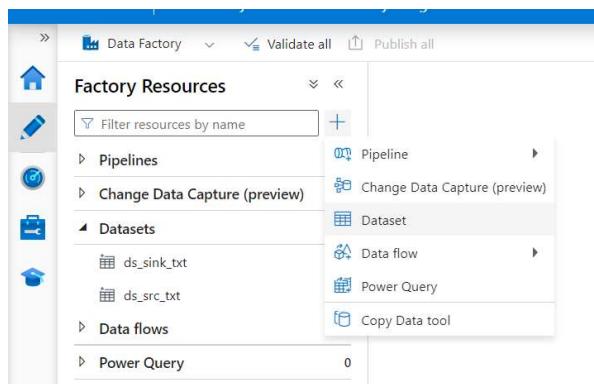
 Connection successful

 Test connection

Create Back Cancel

Create Dataset

1. Go to the Author tab and click on Plus symbol then click on Dataset.

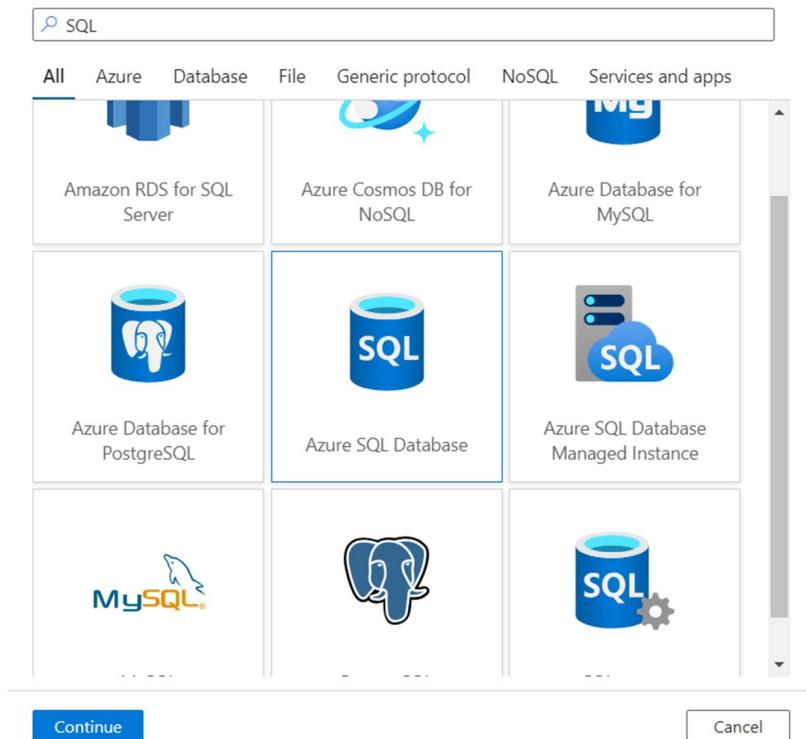


2. Select Azure SQL Database and click on Continue.

New dataset

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

Select a data store



SQL

All Azure Database File Generic protocol NoSQL Services and apps

Amazon RDS for SQL Server	Azure Cosmos DB for NoSQL	Azure Database for MySQL
Azure Database for PostgreSQL	Azure SQL Database	Azure SQL Database Managed Instance
MySQL	MongoDB	Oracle

Continue Cancel

- Give the Name and select the linked service and table that we created before. Then check the none option and click on Ok.

Set properties

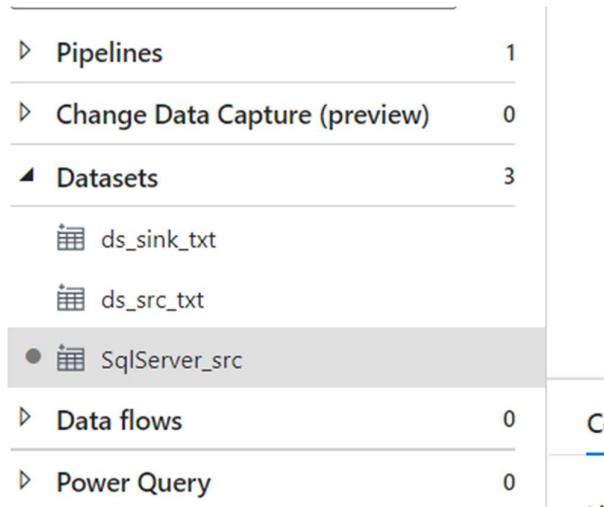
Name
SqlServer_src

Linked service *
AzureSqlDatabase

Table name
dbo.TotalSale

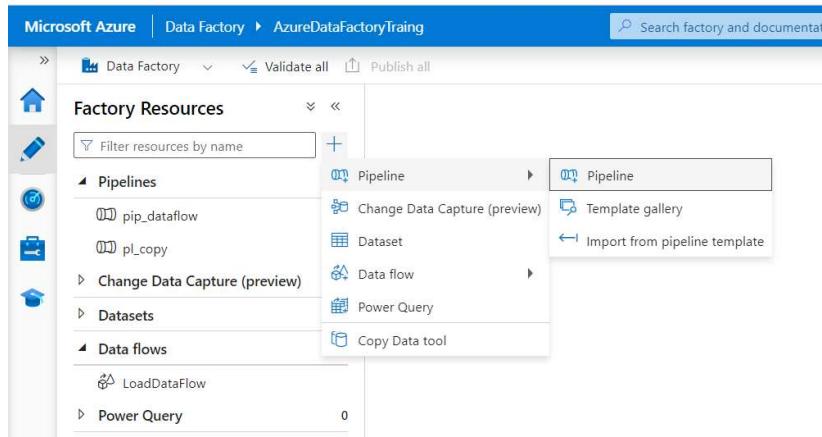
Import schema
 From connection/store None

- Under the dataset you can see our sql server dataset as shown below.



If Condition Activity

- Go to Azure Data Factory and create a Pipeline.



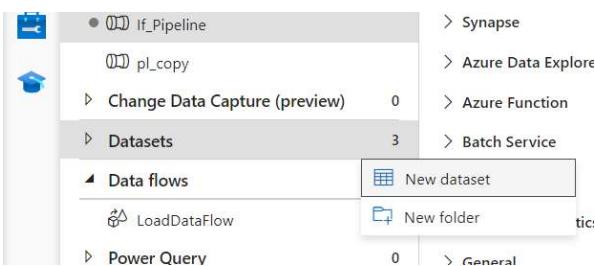
2. Under the Parameter create three parameters as shown below.

Parameters Variables Settings Output

New | **Delete**

Name	Type	Default value
copytoOutput1	String	true
Output1Folder	String	Output1
Output2Folder	String	Output2

3. Now create a Dataset for the destination.



4. Select Azure Blob Storage and click on continue. Select the Delimited text and click on continue.
 5. Give the Name, select the Linked Service, and under the file path give the container name that we created in the previous session. Then check on the None option.

Set properties

Name

Linked service *

File path

 / /

First row as header



Import schema

From connection/store From sample file None

6. Under Parameter create a below parameter.

Connection Schema Parameters

New Delete

Name	Type	Default value
outputfolder	String	Value

7. Under Connection, in the Directory give the below expression.

Exp: @dataset().outputfolder

Connection Schema Parameters

Linked service * AzureBlobStorage_txt

Test connection Edit New Learn more

File path * f-demo / @dataset().outputfolder / File name

Compression type None

Column delimiter Comma (,)

8. Now go to pipeline drag and drop the If condition activity as shown below.

If_Pipeline

If_Conditon_dataset

Activities

if

Iteration & conditionals

If Condition

If Condition1

True
No activities

False
No activities

General Activities (0) User properties

Name * If Condition1

9. Under Activity, in the expression section give the below expression.

Exp: @bool(pipeline().parameters.copytoOutput1)

General Activities (0) User properties

Expression ⓘ @bool(pipeline().parameters.copyto...)

Case	Activity
True	No activities
False	No activities

10. Next click on the True case pencil symbol.

11. Drag and drop the Copy data as shown below.

If_Pipeline If_Conditon_dataset

Activities

cop

Move & transform

Copy data

If_Pipeline > If Condition1 > True activities

Copy data1

12. Under the Source select the source dataset that we created in the previous session.

General Source Sink (1) Mapping Settings User properties

Source dataset * ds_src_txt Open New Prev

File path type

File path in dataset Prefix Wildcard file path List of files ⓘ

Start time (UTC) End time (UTC)

13. Under sink select the destination target that we created before.

14. Under the outputfolder value give the below expression.

Exp: @pipeline().parameters.Output1Folder

General Source Sink Mapping Settings User properties

Sink dataset * If_Conditon_dataset Open New Learn more ⓘ

Dataset properties ⓘ

Name	Value
outputfolder	@pipeline().parameters.Output1Folder

Copy behavior ⓘ

None

15. Next go to pipeline and click on the False pencil symbol as shown below.

General Activities (1) User properties

Expression ⓘ @bool(pipeline().parameters.copyto...)

Case	Activity
True	Copy data1 1 Activity
False	No activities

16. Next drag and drop the copy data as shown below.

If_Pipeline If_Conditon_dataset

Activities

cop

Move & transform

Copy data

If_Pipeline > If Condition1 > False activities

Copy data2

17. Under the source select the source dataset.

General Source Sink Mapping Settings User properties

Source dataset *

ds_src_txt

File path type

File path in dataset

Start time (UTC)

End time (UTC)

Filter by last modified ⓘ

18. Under sink select the destination target that we created before.

19. Under the outputfolder value give the below expression.

Exp: @pipeline().parameters.Output2Folder

General Source Sink Mapping Settings User properties

Sink dataset *

If_Conditon_dataset

Dataset properties ⓘ

Name	Type
outputfolder	string

Copy behavior ⓘ

None

20. Go to Pipeline validate and click on Debug.

21. Our pipeline execute successfully.

The screenshot shows the Azure Data Factory Pipeline validation interface. At the top, there are tabs for 'Validate' (which is checked), 'Debug' (highlighted in blue), and 'Add trigger'. Below these are sections for 'Activities' (listing 'Copy data' and 'Move & transform' options) and a detailed view of the 'If Condition' activity. The 'If Condition' activity has a 'True' path leading to a 'Copy data1' activity, which is marked as 'Succeeded'. The 'False' path is currently empty. Below this, the 'Output' tab is selected, showing a table of pipeline run details. The table includes columns for Activity name, Status, Activity type, Run start, and Duration. One row is shown: 'Copy data1' with status 'Succeeded', run start '7/20/2023 5:15:56 PM', and duration '13s'. A message at the bottom indicates that the data flow activity will start once the debug session is ready.

22. As we all know in the pipeline parameter we have given the value as true.

23. So, it means the data will be stored in the output 1 folder.

24. Go to your container you will see an Output1 folder created.

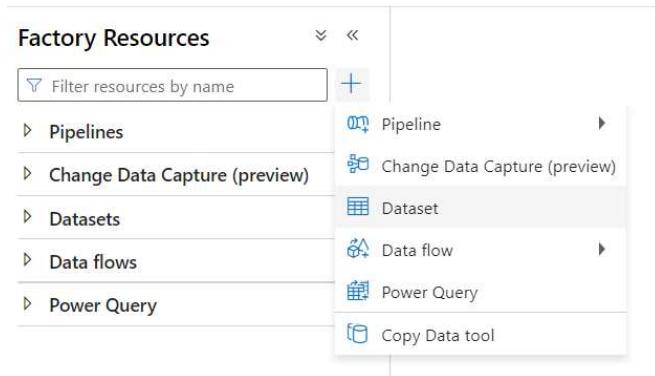
The screenshot shows the Azure Blob Storage container 'f-demo'. The left sidebar has sections for 'Overview', 'Diagnose and solve problems', and 'Access Control (IAM)'. The main area shows a search bar, upload, change access level, refresh, and delete buttons. It also displays authentication method (Access key) and location (f-demo). A search bar for blobs by prefix is present. Below is a table of blobs. The table has columns for Name and Modified. Two entries are listed: 'Output' and 'Output1', both modified on 7/20/2023 at 5:15:56 PM.

25. If you go inside the folder you will see a Source file.

The screenshot shows the contents of the 'Output1' folder. There is a 'Add filter' button. Below is a table of files. The columns are Name and Modified. One file is listed: 'Std_details.txt', modified on 7/20/2023, 5:16:07 PM.

For Each Activity

1. Create a Dataset.



2. Select Azure Blob Storage and click continue. Then select Delimited Text and click on Continue.
3. Give name, linked service, and container name then click ok.

Set properties

Name

Linked service *
 

File path
 / /  

First row as header

Import schema
 From connection/store From sample file None

4. Under parameters create a New Parameter as shown below.

Parameters		
<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	FolderName	String

5. Under connection in the directory give the below expression.

Exp: @dataset().FolderName

Connection Parameters

Linked service * 

Test connection 

File path * / /

Compression type

6. Create a Pipeline.

The screenshot shows the 'Factory Resources' pane on the left with 'Pipelines' selected. A modal dialog is open in the center, titled 'Pipeline'. It contains a list of options: 'Pipeline' (selected), 'Change Data Capture (preview)', 'Dataset', 'Data flow', 'Power Query', and 'Copy Data tool'. Below this is a link 'Import from pipeline template'.

7. Under parameter create a Parameter with the below value.

Value: ["output_1","output_2","output_3"]

The screenshot shows the 'Parameters' tab. A new parameter 'OutputFolder' is being created, set to type 'Array' with a default value of '["output_1","output_2","output_3"]'.

Name	Type	Default value
OutputFolder	Array	["output_1","output_2","output_3"]

8. Drag and drop the ForEach Activity as shown below.

The screenshot shows the 'Activities' tab. A 'ForEach' activity is selected and is being拖拽 (dragged) into the pipeline canvas. The pipeline canvas shows a 'ForEach' activity node with a child 'ForEach1' activity node.

9. Under Settings give the below Expression.

Exp: @pipeline().parameters.OutputFolder

The screenshot shows the 'Settings' tab. The 'Sequential' execution mode is selected. The 'Items' expression is set to '@pipeline().parameters.OutputFolder'.

General	Settings	Activities (0)	User properties
Sequential	<input checked="" type="checkbox"/>		
Items	@pipeline().parameters.OutputFolder		

10. Under Activity click on the pencil symbol.

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

Case	Activity
ForEach	No activities



11. Drag and drop the copy data activity.

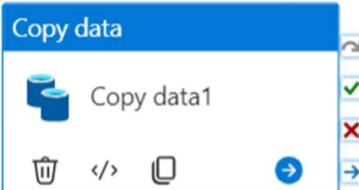
000 **ForEachAct_pipeline** ●

✓ Validate ✓ Validate copy runtime ▶ Debug ⚡ Add trigger

000 [ForEachAct_pipeline](#) >  **ForEach1**

Copy data

Copy data1



12. Under the source select the Employee Dataset.

General **Source** Sink Mapping Settings User properties

Source dataset *  ds_src_txt  Open  New  Preview data 

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

13. Under Sink select the Binary Dataset that we created before and in the FolderName give the below expression.

Exp: @item()

General Source **Sink** Mapping Settings User properties

Sink dataset *  Delimited_Dataset  Open  New 

▼ Dataset properties ⓘ

Name	Value
FolderName	@item()

14. Next go to Pipeline and validate it then click on debug.

15. Here Our Pipeline was executed successfully.

The screenshot shows the Azure Data Factory Pipeline status page. At the top, there are three buttons: 'Validate' (green checkmark), 'Debug' (blue play icon), and 'Add trigger'. Below the buttons is a pipeline diagram titled 'ForEach'. Inside the 'ForEach' container, there is an 'ForEach1' activity, which contains a single 'Copy data1' activity. The 'Copy data1' activity has a green checkmark indicating success. The pipeline run ID is listed as 'a8f3ea5a-408a-4b35-9529-816f64bb96f9'. The 'Pipeline status' is shown as 'Succeeded' with a green checkmark. The 'Output' tab is selected, displaying a table of activity results:

Activity name	Activity status	Activity type	Run start	Duration
Copy data1	Succeeded	Copy data	7/28/2023, 10:20:46 AM	8s
Copy data1	Succeeded	Copy data	7/28/2023, 10:20:36 AM	9s
Copy data1	Succeeded	Copy data	7/28/2023, 10:20:25 AM	10s
ForEach1	Succeeded	ForEach	7/28/2023, 10:20:24 AM	32s

16. Go to Destination and check whether the folders are created or not.



Filter Activity

1. Here in the folder I have 2 Txt files and 1 CSV file. So, I need to get the txt files.

Authentication method: Access key ([Switch to Azure AD User Account](#))

Location: f-demo / Switch1

Search blobs by prefix (case-sensitive)

[Add filter](#)

Name	Modified	Access tier	Archive
<input type="checkbox"/> [...]			
<input type="checkbox"/> Customer_Details.csv	7/28/2023, 10:33:14 ...	Hot (Inferred)	
<input type="checkbox"/> EmpDetails.txt	7/28/2023, 10:33:24 ...	Hot (Inferred)	
<input type="checkbox"/> Std_details.txt	7/21/2023, 10:25:48 ...	Hot (Inferred)	

2. Create a Dataset for this folder location and do not select any file.
3. Select Azure Blob Storage and click on continue. Then select Delimited Text and click on Continue.
4. Give the name, linked service, and container then folder.

Name
Filter_Dataset

Linked service *
AzureBlobStorage_txt

File path
f-demo / Switch1 / File name

First row as header

Import schema
 From connection/store From sample file None

5. Create a pipeline.

The screenshot shows the 'Factory Resources' blade. In the center, there is a search bar with 'Filter_Dataset' typed into it. To the right of the search bar, there is a dropdown menu with the following options: Pipeline (selected), Change Data Capture (preview), Dataset, Data flow, Power Query, and Copy Data tool. On the left, there is a sidebar with categories: Pipelines, Change Data Capture (preview), Datasets, Data flows, and Power Query. Below the sidebar, there is a search bar with 'Filter resources by name' and a '+' button to add new resources.

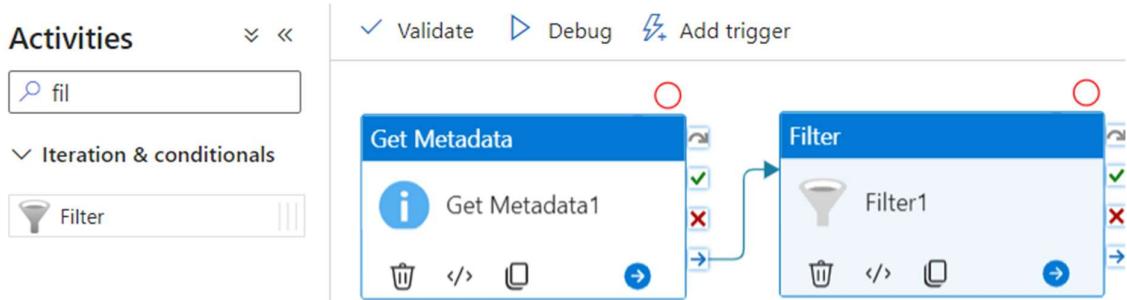
6. Drag and drop the Get Metadata.

The screenshot shows the 'Activities' blade. On the left, there is a search bar with 'me' and a tree view with 'General' expanded, showing 'Get Metadata'. On the right, there is a list of activities with 'Get Metadata' selected. The 'Get Metadata' activity has a red circle with a question mark icon above it. There are also other icons for validation, debugging, and adding triggers.

7. Under Settings select the Dataset that we created before and select the field list.

The screenshot shows the 'Settings' blade for the 'Get Metadata' activity. At the top, there are tabs: General, Settings (selected), and User properties. Below the tabs, there is a 'Dataset *' dropdown set to 'Filter_Dataset'. There are also buttons for 'Open', 'New', and 'Learn more'. The 'Field list *' section contains a 'New' button, a 'Delete' button, an 'Argument' checkbox, and a 'Child items' dropdown.

8. Drag and drop the filter activity and make the connection as shown below.



9. Under settings for the Item give the below expression.

Exp: @activity('Get Metadata1').output.childItems

10. For the Condition give the below expression.

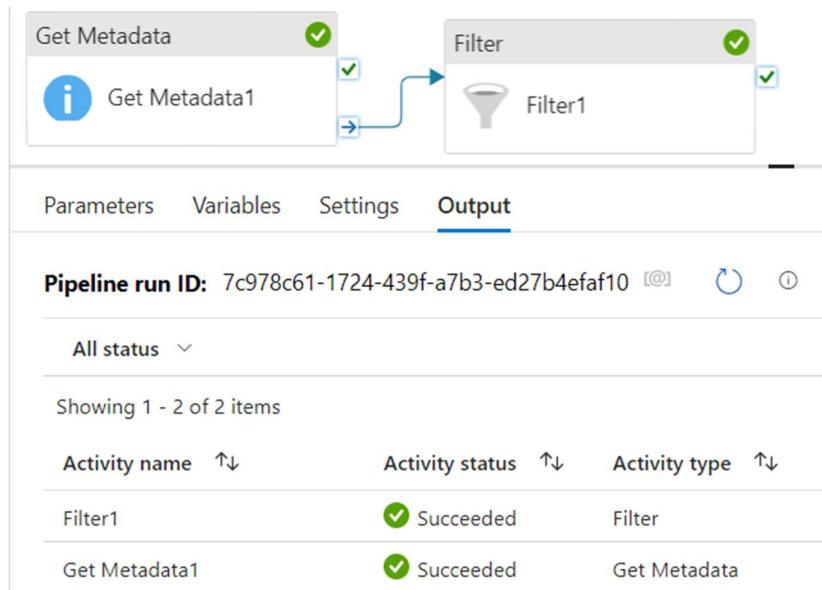
Exp: @endswith(item().name,'txt')

General **Settings** User properties

Items	@activity('Get Metadata1').output.chi...
Condition	@endswith(item().name,'txt')

11. Now validate the pipeline and click on debug.

12. Here Our Pipeline was executed successfully.



13. Now click on filter output and check the txt file names.

The screenshot shows the 'Output' section of a pipeline run. It displays the following JSON data:

```
itemsCount": 2,
"FilteredItemCount": 2,
"Value": [
  {
    "name": "EmpDetails.txt",
    "type": "File"
  },
  {
    "name": "Std_details.txt",
    "type": "File"
  }
]
```

Below the JSON, there is a table of activities:

Filter1	✓ Succeeded	Filter	7/28/2023,
Get Metadata1	✓ Succeeded	Get Metadata	7/28/2023,

Set Variable Activity

1. Create a Pipeline.

The screenshot shows the 'Factory Resources' blade. The 'Pipelines' item is selected in the left navigation pane, and its details are shown in the main area. A context menu is open over the 'Pipeline' item, with 'Pipeline' highlighted.

2. Create a parameter with the below value.

Value: Address: 221 M.G. Road Road. City: Kolkata. State: West Bengal

The screenshot shows the 'Parameters' blade. A new parameter is being created with the following details:

Name	Type	Default value
addressLine	String	Address: 221 M.G. Road R

3. Under Variables create a variable as shown below.

The screenshot shows the 'Variables' tab in a pipeline editor. At the top, there are tabs for 'Parameters', 'Variables' (which is underlined), 'Settings', and 'Output'. Below the tabs, there are buttons for 'New' and 'Delete'. A table lists variables: one row for 'cityName' which is selected, showing it's a 'String' type with a 'Value' field. There are also 'Name' and 'Type' columns.

Name	Type	Default value
cityName	String	

4. Drag and drop the set variable.

The screenshot shows the 'Activities' pane in a pipeline editor. It includes a search bar, validation and debug buttons, and an 'Add to...' button. A list of activities is shown, with '(x) Set variable' and '(x) Set variable1' both being selected, indicated by a blue border around them.

5. Under setting select the name and give the below expression.

```
Exp:@substring(pipeline().parameters.addressLine,add(indexof(pipeline().parameters.addressLine,'City:'), 6),sub(sub(indexof(pipeline().parameters.addressLine,'State:'), 2),add(indexof(pipeline().parameters.addressLine,'City:'), 6)))
```

The screenshot shows the 'Settings' tab for the 'Set variable' activity. It has tabs for 'General', 'Settings' (which is underlined), and 'User properties'. Under 'Settings', the 'Variable type' is set to 'Pipeline variable'. The 'Name' is 'cityName' and the 'Value' is '@substring(pipeline().parameters.ad...'. There is a 'New' button next to the name input.

6. Next validate the pipeline and click on debug.

7. Here our pipeline was executed successfully.

The screenshot shows the 'Output' tab for the pipeline run. It displays the pipeline run ID: 897d9403-2bc6-496b-ba81-16973473a8e1. Below it, a table shows the activity details: 'Activity name' is 'Set variable1', 'Activity status' is 'Succeeded', and 'Activity type' is 'Set variable'. The status is indicated by a green checkmark icon.

Activity name	Activity status	Activity type
Set variable1	Succeeded	Set variable

- In the “Output” tab, the output is displayed as “Kolkata”.

```

Output
Copy to clipboard

{
  "name": "cityName",
  "value": "Kolkata"
}

Set variable1 Succeeded Set variable
  
```

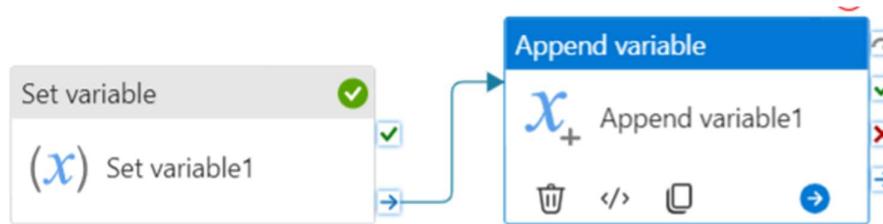
Append Variable

- In this example we are going to use the same pipeline that we created before for the set variable active.
- So open the set variable pipeline that we created just before. Under variables add two more variables.
- For the listofCities give the below value.

Value: ["New York","London","Tokyo","Singapore City"]

Name	Type	Default value
cityName	String	Value
listOfCities	Array	["New York", "London", "Tokyo",]
appendedCities	Array	Value

- Drag and drop the Append Variable Activity and make the connection as shown below.



5. Under Settings select the name and give the value.

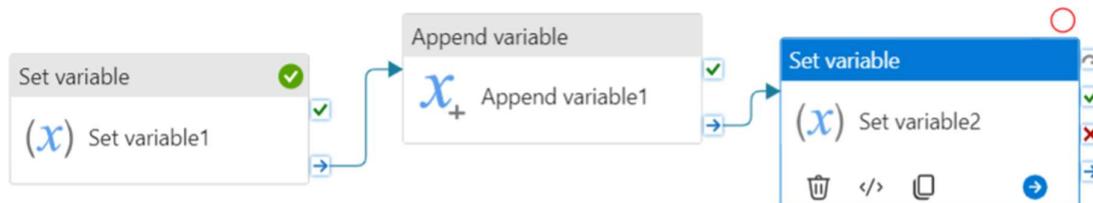
Value: @variables('cityName')

General **Settings** User properties

Name * + New

Value

6. Drag and drop the set variable activity and make connections as shown below.



7. Under Settings give the name and value.

Value: @variables('listOfCities')

General **Settings** User properties

Variable type ⓘ Pipeline variable Pipeline return value

Name * + New

Value

8. Next validate the Pipeline and click on Debug.

9. Here our Pipeline was executed successfully.

✓ Validate ▷ Debug ⚡ Add trigger

Pipeline run ID: 79f1bfa0-ac7a-40ea-93d8-bb72064e59c3 ⓘ ⟳ ⓘ Pipeline status ✓ Succeeded

All status ▼

Showing 1 - 3 of 3 items

Activity name	Activity status	Activity type	Run start	Duration
Set variable2	✓ Succeeded	Set variable	7/28/2023, 3:56:36 PM	Less than 1s
Append variable1	✓ Succeeded	Append variable	7/28/2023, 3:56:35 PM	Less than 1s
Set variable1	✓ Succeeded	Set variable	7/28/2023, 3:56:35 PM	Less than 1s

10. In the Output tab, the output is displayed as the value of the cityName Variable, appended to the Variable listOfCities of Array Type.

Output

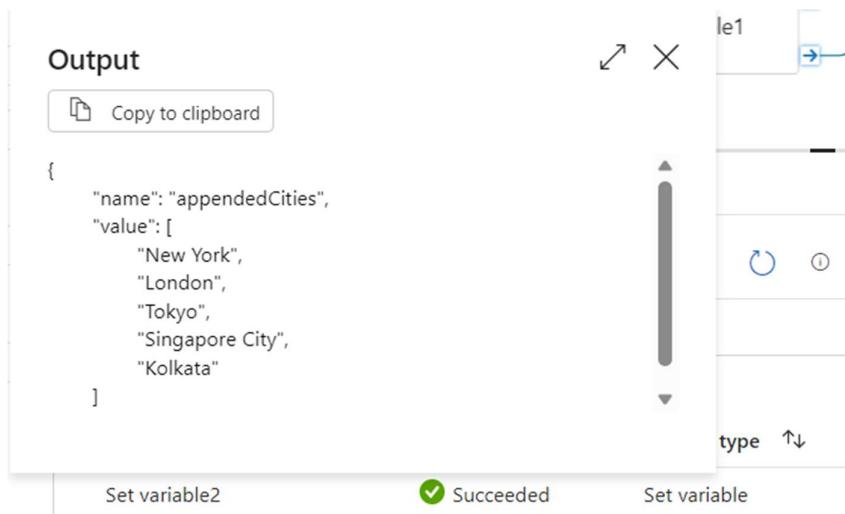
Copy to clipboard

```
{  
  "name": "appendedCities",  
  "value": [  
    "New York",  
    "London",  
    "Tokyo",  
    "Singapore City",  
    "Kolkata"  
  ]  
}
```

le1

Set variable2 Succeeded Set variable

type ↑↓



Until Activity

1. Go to Container and go to Output folder then delete the file.

The screenshot shows the Azure Storage Explorer interface. A container named 'f-demo' is selected. Inside it, there is a folder named 'Output' which contains three files: '_SUCCESS', 'part-00000-e6862a16-47ef-4712-a2dc-d150acf6f...', and 'Std_details.txt'. The 'Std_details.txt' file has a context menu open, displaying various actions such as View/edit, Download, Properties, Generate SAS, View versions, View snapshots, Create snapshot, Change tier, Acquire lease, Break lease, and Delete.

2. Create a Pipeline.

The screenshot shows the Microsoft Azure Data Factory portal. On the left, there is a sidebar titled 'Factory Resources' with options like Pipelines, Change Data Capture (preview), Datasets, Data flows, and Power Query. Under Pipelines, a tooltip is displayed, showing 'Pipeline' (selected), 'Change Data Capture (preview)', 'Dataset', 'Data flow', 'Power Query', and 'Copy Data tool'. At the top, there is a search bar and a 'Publish all' button.

3. Under variable create a variable as shown below.

The screenshot shows the 'Variables' tab in the Azure Data Factory Variables section. It displays a table with columns: Name, Type, and Default value. There is one entry: 'FileAvailable' of type String with a default value of 'false'.

Name	Type	Default value
FileAvailable	String	false

4. Drag and drop the Until activity.

The screenshot shows the Azure Data Factory Pipeline designer. A pipeline named 'Until_Pipe' is open. In the 'Activities' section, there is a search bar with 'unti' typed into it. Below it, under 'Iteration & conditionals', there is a list with 'Until' selected. The main workspace shows an 'Until' activity with a sub-activity named 'Until1' which currently has no activities.

5. Under settings, in the expression give the below expression.

Exp: @bool(variables('FileAvailable'))

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

Expression ⓘ

Timeout ⓘ

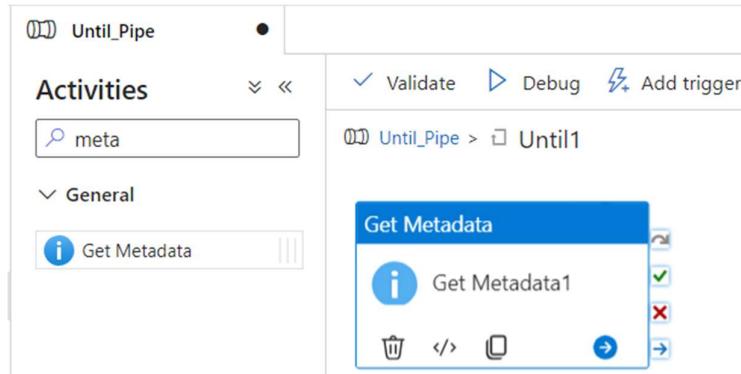
6. Under Activity click on the pencil symbol.

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

Case	Activity
Until	No activities



7. Drag and drop the get metadata activity.



The screenshot shows the 'Until' activity configuration. In the 'Activities' section, 'Get Metadata' is selected. The 'Get Metadata' activity is highlighted in a blue box. The 'Activity' pane shows 'Get Metadata1' with a green checkmark next to it. The top right of the screen has buttons for Validate, Debug, and Add trigger.

8. Under setting select the source dataset. And click on new to add an Argument.

General **Settings** User properties

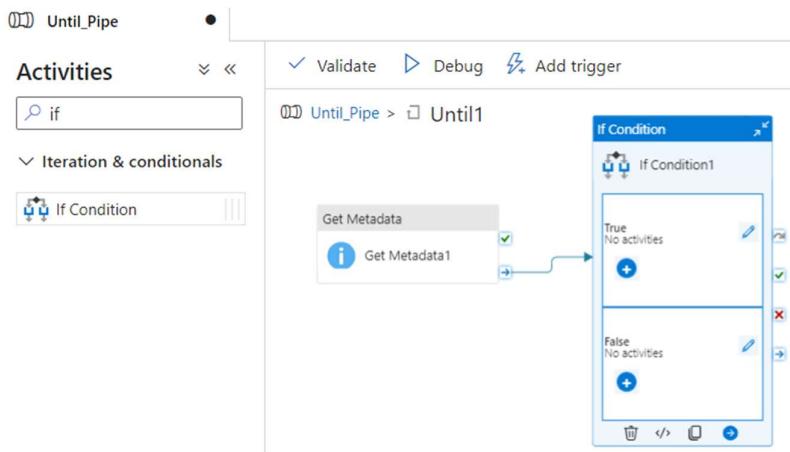
Dataset *  Open  New 

Field list *  New 

Argument

Exists

9. Drag and drop the If condition and make the connection as shown below.



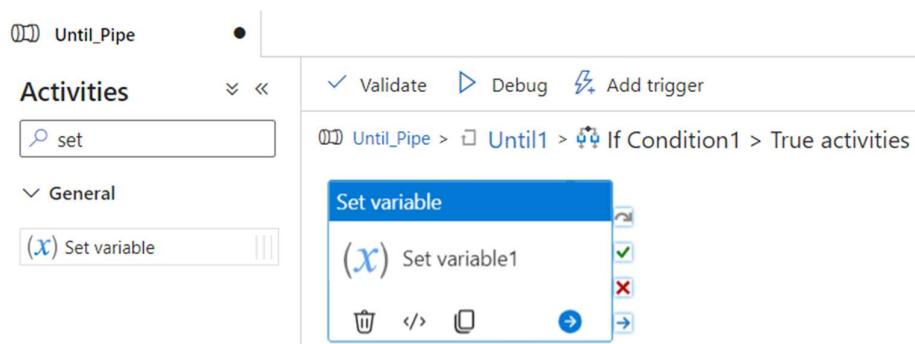
10. Under the activity, in the expression give the below expression.

Exp: @bool(activity('Get Metadata1').output.exists)

This screenshot shows the 'Activities' tab for the 'If Condition' activity. The 'Expression' field contains the expression '@bool(activity('Get Metadata1').output.exists)'. Below the expression, there are two sections: 'Case' and 'Activity'. The 'Case' section is empty. The 'Activity' section contains a single 'Activity' item, which is the 'Get Metadata' activity.

11. Click on the True condition pencil symbol.

12. Drag and drop the set variable activity as shown below.



13. Under setting give the Name and value as shown below.

This screenshot shows the 'Settings' tab for the 'Set variable' activity. Under 'Variable type', 'Pipeline variable' is selected. The 'Name' field is set to 'FileAvailable' and the 'Value' field is set to 'true'.

14. Now go back to Until and click on the False activity pencil symbol.

The screenshot shows the 'Until' activity configuration. At the top, there is an 'Expression' input field containing the expression '@bool(activity('Get Metadata1').output...'. Below it, there are two rows for 'Case' and 'Activity':

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

Each row has a blue pencil icon to its right.

15. Drag and drop the wait activity.

The screenshot shows the 'Until_Pipe' pipeline. In the 'Activities' pane, a search bar contains 'wait'. A 'Wait' activity is selected and shown in the main pane. The pipeline structure is: Until_Pipe > Until1 > If Condition1 > Wait1.

16. Under the settings give the time as 60 sec.

The screenshot shows the 'Settings' tab for the 'Wait' activity. The 'Wait time in seconds *' field is set to '60'.

17. Go back to the pipeline and drag and drop the Copy data activity and make the connection.

The screenshot shows the 'Until_Pipe' pipeline. In the 'Activities' pane, a search bar contains 'cop'. A 'Copy data' activity is selected and shown in the main pane. It is connected to the 'Wait1' activity.

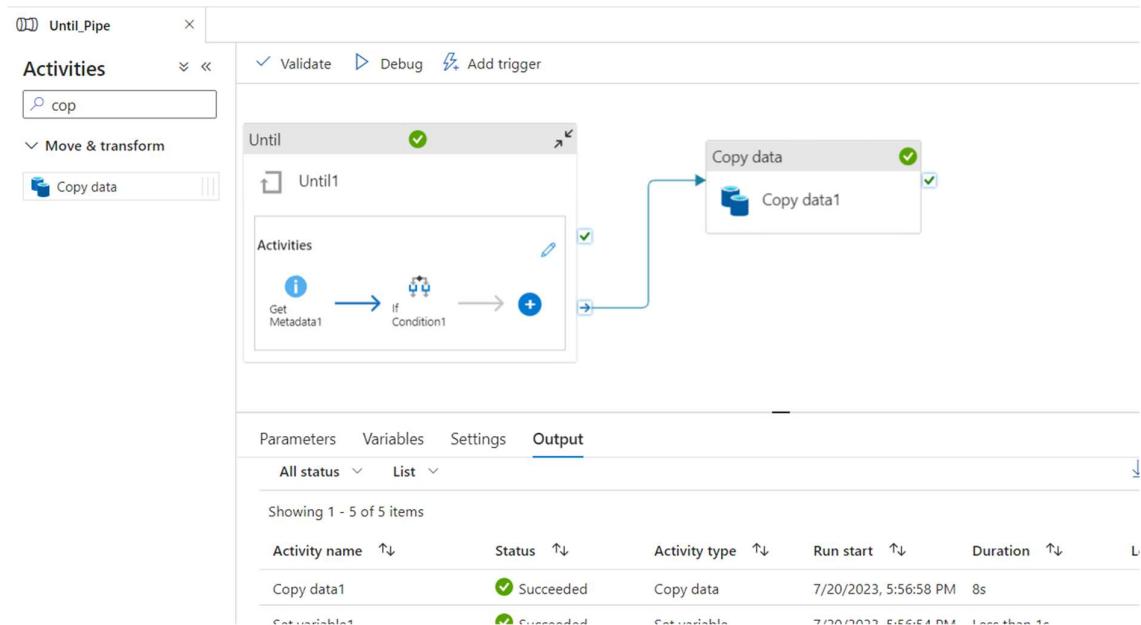
18. Under the Source select the source.

The screenshot shows the 'Source' tab for the 'Copy data1' activity. The 'Source dataset *' dropdown is set to 'ds_src_txt'. The 'File path type' dropdown is set to 'File path in dataset'. There are fields for 'Start time (UTC)' and 'End time (UTC)' with empty input boxes. A 'Filter by last modified' checkbox is also present.

19. Under sink select the destination dataset.

The screenshot shows the 'Sink' tab of a pipeline configuration. The 'Sink dataset' dropdown is set to 'ds_sink_txt'. The 'Copy behavior' dropdown is set to 'None'. There is also a 'Max concurrent connections' input field.

20. Now validate the pipeline and click on Debug. Here our pipeline runs successfully as shown below.



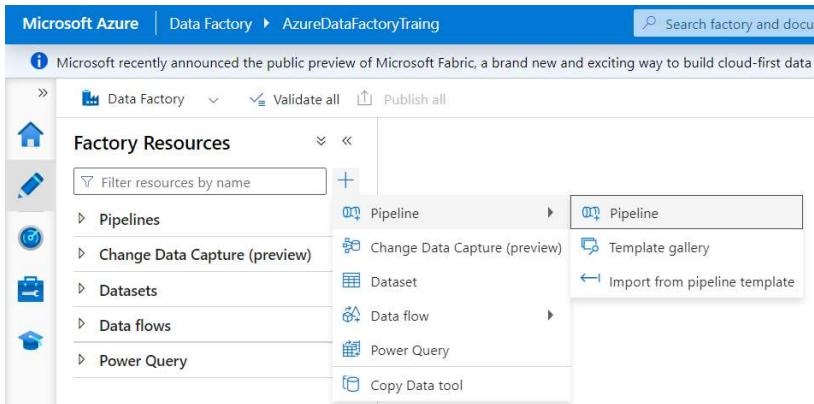
21. So in our case we already have a file in our source so it will directly copy into the target folder.

The screenshot shows the Azure Storage Explorer. It lists files in a folder. The files are: '_SUCCESS', 'part-00000-e6862a16-47ef-4712-a2dc-d150acfb6f...', and 'Std_details.txt'. The '_SUCCESS' file was modified on 7/20/2023, 4:09:54 PM and is in the 'Hot (Inferred)' access tier. The other two files were modified on 7/20/2023, 5:57:05 PM and are in the 'Hot (Inferred)' access tier.

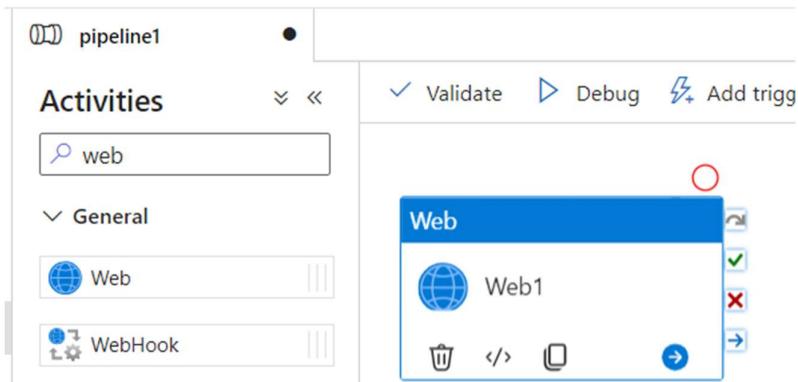
Name	Modified	Access tier
_SUCCESS	7/20/2023, 4:09:54 PM	Hot (Inferred)
part-00000-e6862a16-47ef-4712-a2dc-d150acfb6f...	7/20/2023, 4:09:54 PM	Hot (Inferred)
Std_details.txt	7/20/2023, 5:57:05 PM	Hot (Inferred)

Web Activity

1. Go to Azure Data Factory and go to Author then create a pipeline.



2. Drag and drop the web activity as shown below.



3. Under setting give the URL and set the method.

URL: <http://dummy.restapiexample.com/api/v1/employees>

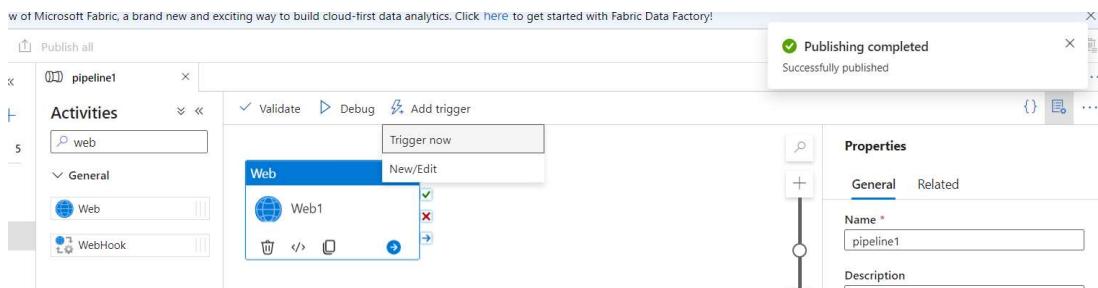
General Settings User properties

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

Method * ⓘ

Authentication ⓘ

4. Publish the pipeline and click on Add trigger then click on Trigger now.



5. Next click Ok.

6. Now go to the Monitor tab and you can see our pipeline status is succeeded.

A screenshot of the Microsoft Fabric Monitor tab. The left sidebar shows "Runs" and "Pipeline runs" selected. The main area displays "Pipeline runs" with a table header: Pipeline name, Run start, Run end, Duration, Triggered by, Status, and Run. One row is shown: "pipeline1" with a run start of "7/21/2023, 9:50:17 AM", a run end of "7/21/2023, 9:50:23 AM", a duration of "6s", triggered by "Manual trigger", and a status of "Succeeded".

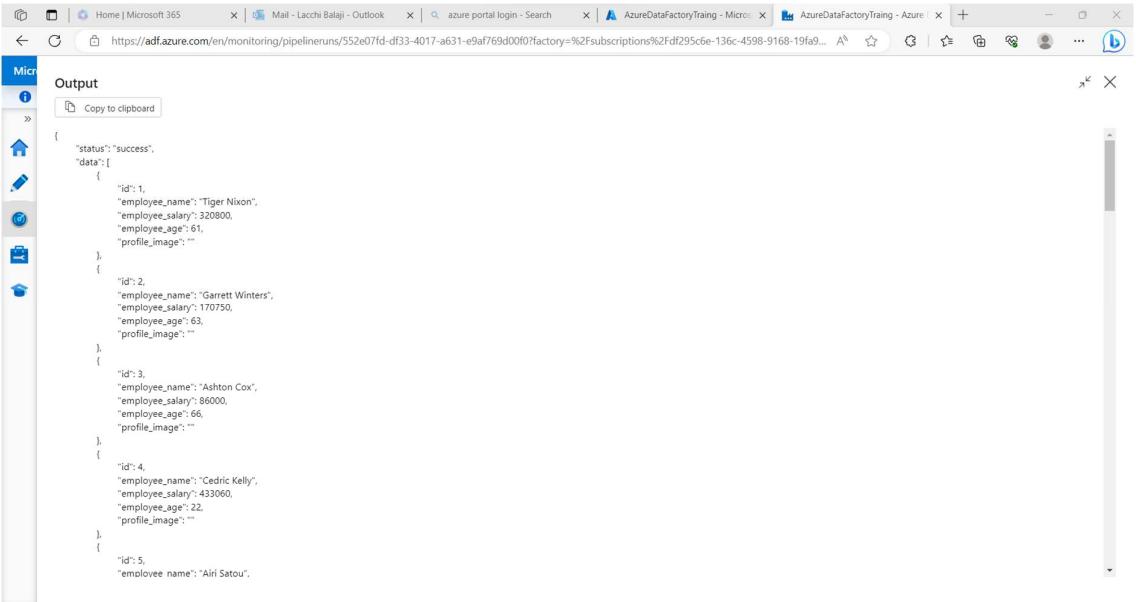
7. Next click on the Pipeline. Inside that click on Input you will see below input.

A screenshot of the Microsoft Fabric Pipeline details page for "pipeline1". The left sidebar shows "Pipeline runs" selected. The main area shows "Activity runs" for "Web1". Under "Input", there is a JSON configuration block:

```
{
  "url": "http://dummy.restapiexample.com/api/v1/employees",
  "method": "GET",
  "headers": {}
}
```

A "Copy to clipboard" button is available next to the JSON code.

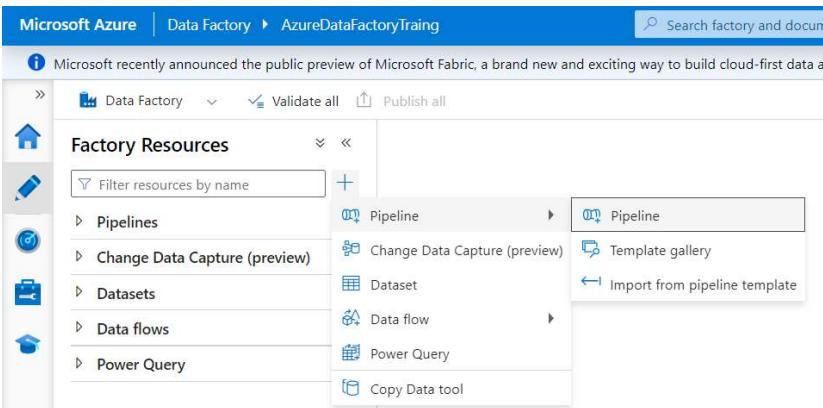
8. Now if you click on Output you will see an output as shown below.



```
{ "status": "success", "data": [ { "id": 1, "employee_name": "Tiger Nixon", "employee_salary": 320800, "employee_age": 61, "profile_image": ""}, { "id": 2, "employee_name": "Garrett Winters", "employee_salary": 170750, "employee_age": 63, "profile_image": ""}, { "id": 3, "employee_name": "Ashton Cox", "employee_salary": 86000, "employee_age": 66, "profile_image": ""}, { "id": 4, "employee_name": "Cedric Kelly", "employee_salary": 430600, "employee_age": 22, "profile_image": ""}, { "id": 5, "employee_name": "Airi Satou", "employee_salary": 112000, "employee_age": 33, "profile_image": ""} ]}
```

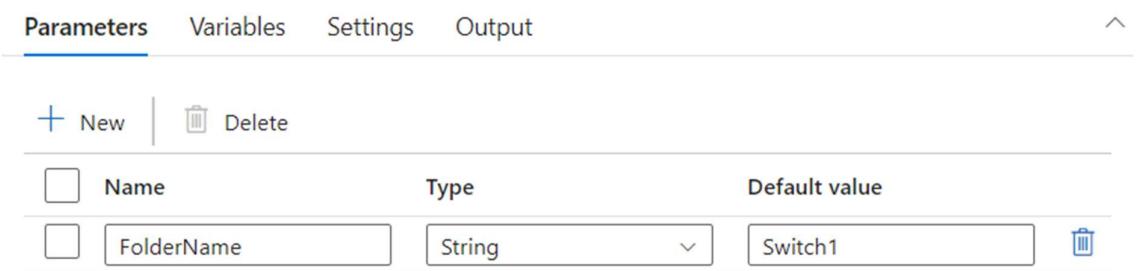
Switch Activity

1. Create a pipeline.



The screenshot shows the Microsoft Azure Data Factory interface. In the top navigation bar, it says "Microsoft Azure | Data Factory > AzureDataFactoryTraining". Below the navigation bar, there's a message about the public preview of Microsoft Fabric. On the left, there's a sidebar titled "Factory Resources" with icons for Home, Pipelines, Change Data Capture (preview), Datasets, Data flows, and Power Query. The main area shows a "Pipeline" selected in a dropdown menu. A tooltip for "Pipeline" is open, listing "Pipeline", "Change Data Capture (preview)", "Dataset", "Data flow", "Power Query", and "Copy Data tool".

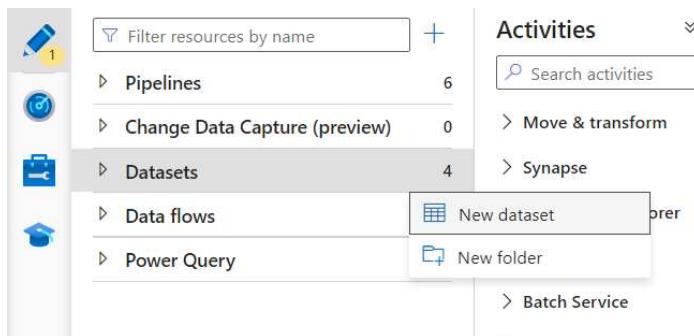
2. Under the parameter create the below parameter.



The screenshot shows the "Parameters" tab in the Azure Data Factory interface. At the top, there are tabs for "Parameters", "Variables", "Settings", and "Output". Below the tabs, there are buttons for "New" and "Delete". A table lists a single parameter: "FolderName" of type "String" with a default value of "Switch1".

Name	Type	Default value
FolderName	String	Switch1

3. Create a Dataset.
4. Select the Azure Blob Storage and click on Continue. Then select the Binary and click on continue.



5. In the properties give the name, linked service, and the container name as shown below.

Set properties

Name
SwitchDataset

Linked service *
AzureBlobStorage_txt

File path
f-demo / Directory / File name

First row as header

Import schema
 From connection/store From sample file None

6. Now under parameter create a below parameter.

Connection	Parameters
New	Delete
<input type="checkbox"/> Name	Type
<input type="checkbox"/> folder	String
	Default value
	<input type="text"/> Value

7. Under connection, in the directory give the below expression as shown below.

Exp: @dataset().folder

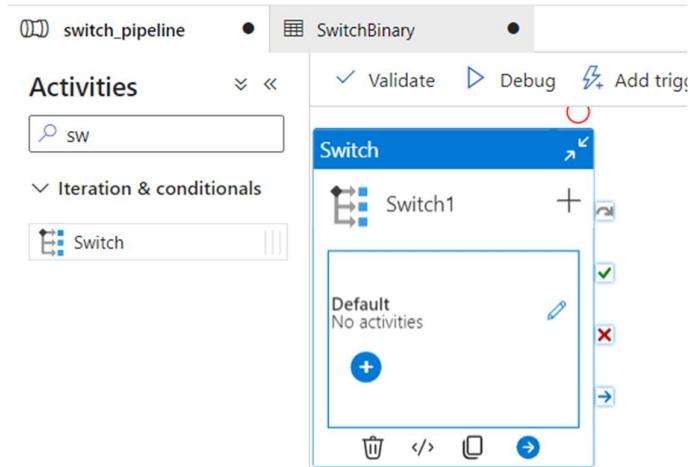
Connection Parameters

Linked service *
AzureBlobStorage_txt

File path *
f-demo / @dataset().folder / File name

Compression type
Select...

8. Go to Pipeline drag and drop the switch activity.



9. Under Activity give the below expression.

Exp: @pipeline().parameters.FolderName

A screenshot of the 'Activities' tab for the 'Switch1' activity. The 'Expression' field is set to '@pipeline().parameters.FolderName'. The 'Case' column shows 'Default' and the 'Activity' column shows 'No activities'.

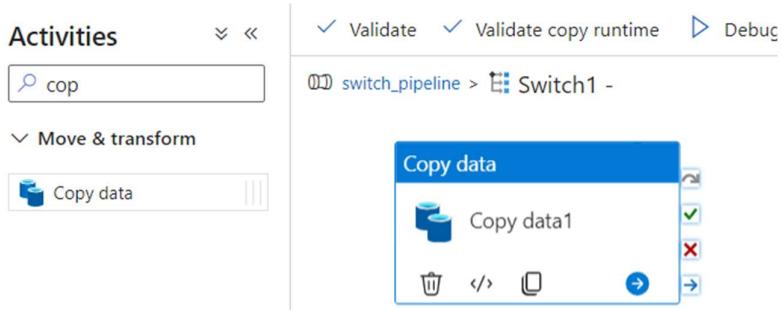
10. Now click on the plus symbol to add a case. Then click on the pencil symbol.

A screenshot of the 'Activities' tab for the 'Switch1' activity. It shows two cases: 'Default' and 'Case1'. Both cases have 'No activities' assigned. There are edit icons next to each case entry.

11. Give the case name Switch1 as shown below.

A screenshot of the 'General' tab for the 'Case1' row. The 'Case' field is filled with 'Switch1'.

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



13. Under the Source select the source dataset that we created in the day 1 session.

A screenshot of the 'Source' tab for a copy data activity. The tab has tabs for General, Source, Sink, Mapping, Settings, and User properties. The Source tab is active. A dropdown menu labeled 'Source dataset *' shows 'ds_src_txt' selected. Below the dropdown are buttons for Open, New, Preview data, and Learn more. Under 'File path type', the radio button 'File path in dataset' is selected. There are also options for Prefix, Wildcard file path, and List of files.

14. Under the sink select the dataset that we created before and under the folder name give the below expression.

Exp: @pipeline().parameters.FolderName

A screenshot of the 'Sink' tab for a copy data activity. The tab has tabs for General, Source, Sink, Mapping, Settings, and User properties. The Sink tab is active. A dropdown menu labeled 'Sink dataset *' shows 'SwitchDataset' selected. Below the dropdown are buttons for Open, New, and Learn more. Under 'Dataset properties', there is a table with a single row: 'Name' is 'folder' and 'Value' is '@pipeline().parameters.FolderName'. There is also a 'Copy behavior' section with a dropdown set to 'None'.

15. Now go to pipeline and add another case.

A screenshot of the 'Activities (1)' tab for a pipeline. The tab has tabs for General, Activities (1), and User properties. The Activities (1) tab is active. It shows two cases: 'Default' and 'Case2'. The 'Default' case has an activity named 'Copy data1'. The 'Case2' case currently has no activities.

16. Name it Switch2.

The screenshot shows the 'General' tab of a pipeline component configuration. The 'Case' field is highlighted and contains the value 'Switch2'.

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

The screenshot shows the 'Activities' pane of a pipeline editor. A 'Copy data' activity is selected and highlighted with a blue border. It is labeled 'Copy data2'.

18. Under the Source select the source dataset that we created in the day 1 session.

The screenshot shows the 'Source' tab of a pipeline component configuration. The 'Source dataset' dropdown is set to 'ds_src_txt'. Below it, under 'File path type', the radio button for 'File path in dataset' is selected.

19. Under the sink select the dataset that we created before and under the folder name give the below expression.

Exp: @pipeline().parameters.FolderName

The screenshot shows the 'Sink' tab of a pipeline component configuration. The 'Sink dataset' dropdown is set to 'SwitchDataset'. Under 'Dataset properties', there is a table:

Name	Value	Type
folder	@pipeline().parameters.FolderName	

20. Next go to Pipeline and validate it and click on debug.

21. Here our pipeline executes successfully.

The screenshot shows the Azure Data Factory Pipeline status page. At the top, there are buttons for Validate, Debug, and Add trigger. Below that is a tree view of the pipeline structure under a 'Switch' node, with 'Switch1' selected. The pipeline run ID is e34003d0-7fed-4d0d-847e-1d42a749e19a, and the Pipeline status is Succeeded. The table below lists two activities: 'Copy data1' and 'Switch1', both of which succeeded. The table has columns for Activity name, Activity status, Activity type, Run start, and Duration.

Activity name	Activity status	Activity type	Run start	Duration
Copy data1	Succeeded	Copy data	7/21/2023, 10:25:41 AM	9s
Switch1	Succeeded	Switch	7/21/2023, 10:25:41 AM	10s

22. As initially give the parameter value as switch1 so if you go to your container you will see it as a Switch1 folder.

The screenshot shows the Azure Storage Blob Container 'f-demo'. The container has an 'Overview' tab selected. It displays the authentication method as 'Access key (Switch to Azure AD User Account)' and the location as 'f-demo'. The blob list shows a folder named 'Switch1' and other files like 'Output' and 'Output1'. The table has columns for Name, Modified, and Access tier.

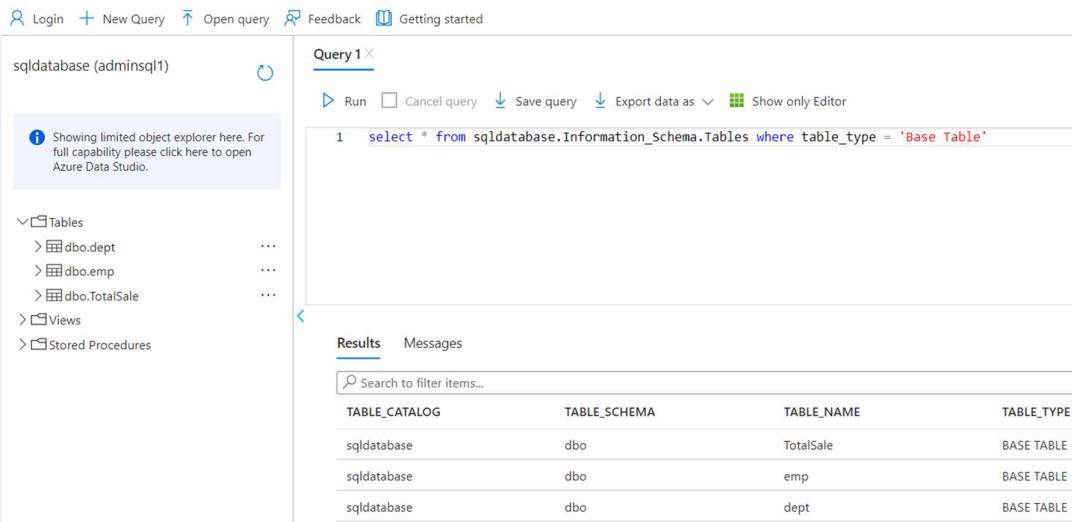
Name	Modified	Access tier
Output		
Output1		
Switch1		
Output	7/20/2023, 4:09:54 PM	Hot (Inferred)

23. Inside the folder the file is copied from source to destination.

Name	Modifi
[..]	
Std_details.txt	7/21/2

Lookup Activity

1. Open the SQL database that we created in the previous session.
2. Here we have three tables. Here dynamically we will store these tables in the Azure Blob storage.



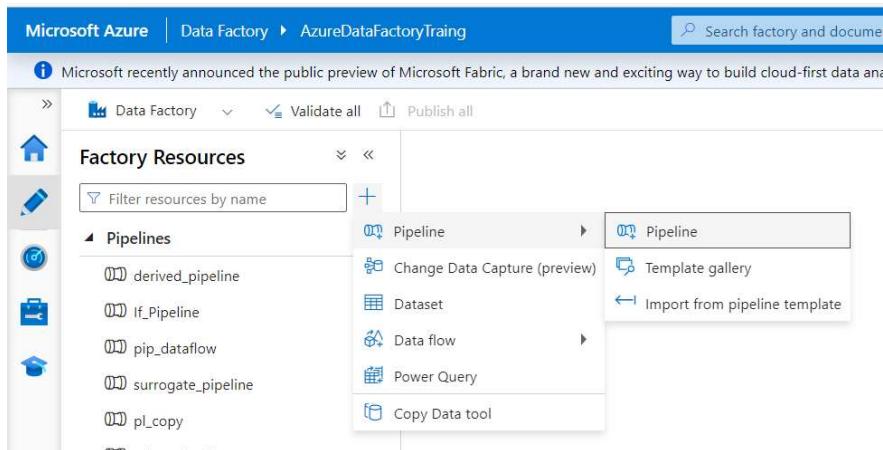
The screenshot shows the Microsoft Data Studio interface. On the left, there's a tree view of a database named 'sqldatabase' containing 'Tables', 'Views', and 'Stored Procedures'. In the center, a query editor window titled 'Query 1' displays the following SQL code:

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

The results pane below shows a table with four columns: TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, and TABLE_TYPE. The data is as follows:

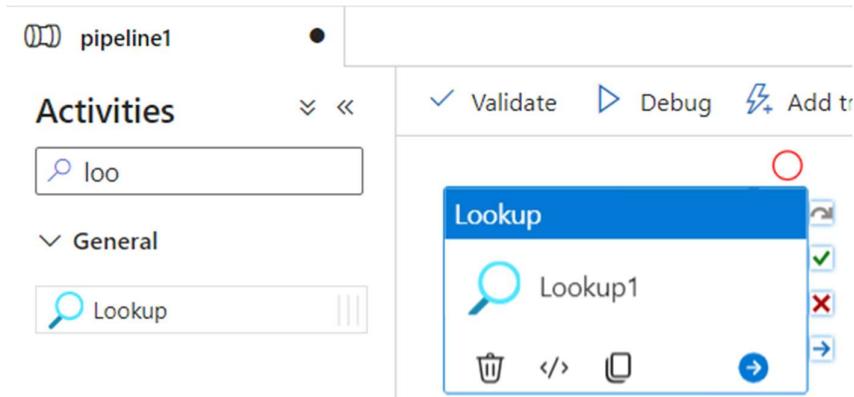
TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
sqldatabase	dbo	TotalSale	BASE TABLE
sqldatabase	dbo	emp	BASE TABLE
sqldatabase	dbo	dept	BASE TABLE

3. Create a Pipeline.



The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar lists 'Factory Resources' under 'Pipelines', showing several pipelines like 'derived_pipeline', 'If_Pipeline', etc. A context menu is open over a pipeline named 'Pipeline', with options including 'Change Data Capture (preview)', 'Dataset', 'Data flow', 'Power Query', and 'Copy Data tool'.

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



The screenshot shows the Microsoft Azure Data Factory pipeline editor for a pipeline named 'pipeline1'. In the 'Activities' section, a search bar contains 'loo'. Under the 'General' category, a 'Lookup' activity is selected. A tooltip for the 'Lookup' activity is displayed, showing its name 'Lookup1' and various icons for configuration and validation.

5. Under settings click on New.

General **Settings** ¹ User properties

Source dataset *

First row only

6. Select the Azure SQL Database and click continue.
7. Just give the name and select the linked service and click ok.

Set properties

Name

Linked service *

Table name

Import schema
 From connection/store None

> Advanced

8. Next check the query option and give the below query.

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

9. Uncheck the First row-only option.

General **Settings** User properties

Source dataset *

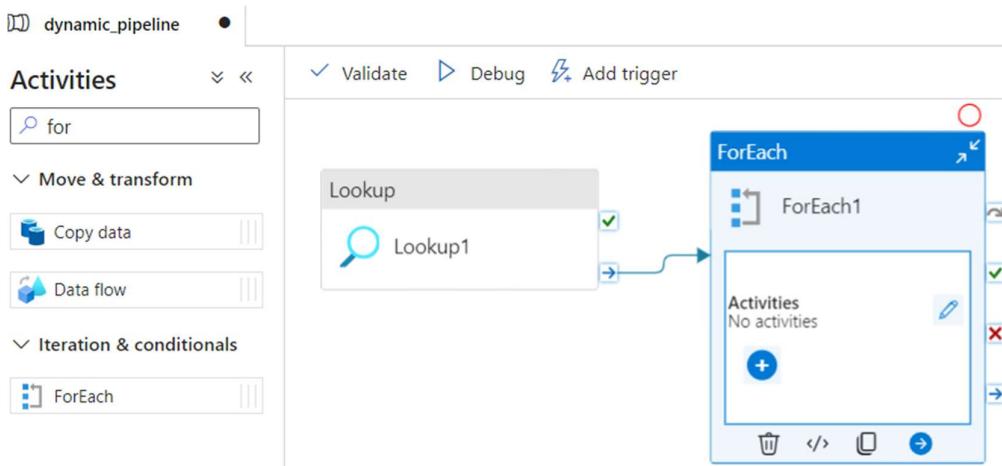
First row only

Use query
 Table Query Stored procedure

Query *

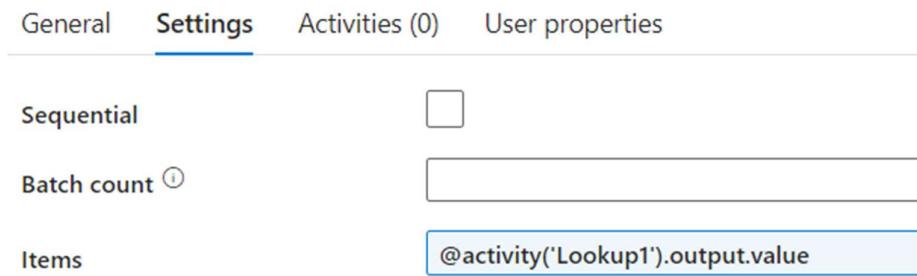
Query timeout (minutes)

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

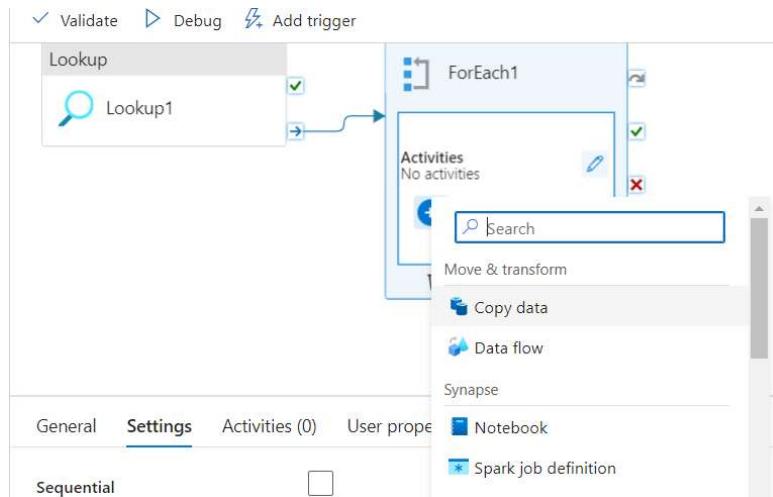


11. Under setting in the Items give the below expression as shown below.

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



12. Click on the plus symbol in the foreach loop and click on copy data option.



13. Under the source click on New.

General **Source** ¹ Sink ¹ Mapping Settings User properties

Source dataset *

14. Select the Azure SQL Database and click on continue.

15. Then give the name and linked service then click on Ok.

Set properties

Name

Dynamic_Source

Linked service *

AzureSqlDatabase

Table name



Edit

Import schema

From connection/store None

> Advanced

16. Click on Open.

General **Source** ¹ Sink ¹ Mapping Settings User properties

Source dataset *

Use auerv Table Query Stored procedure

17. Under parameters add two parameters as shown below.

Connection Schema **Parameters**

<input type="checkbox"/> Name	Type	Default value	<input type="button" value=""/>
<input type="checkbox"/> TableName	String	<input type="text" value="Value"/>	<input type="button" value=""/>
<input type="checkbox"/> SchemaName	String	<input type="text" value="Value"/>	<input type="button" value=""/>

18. Now go to connections and check the edit option and give the expression as shown below.

Exp: @dataset().SchemaName

Exp: @dataset().TableName

The screenshot shows the 'Connections' blade with the 'Connection' tab selected. A dropdown menu shows 'AzureSqlDatabase' is selected. Below it, there are buttons for 'Test connection' (with a green checkmark and 'Connection successful'), 'Edit', 'New', and 'Learn more'. Under the 'Table' section, two fields are shown: '@dataset().SchemaName' and '@dataset().TableName', both with a 'Edit' button next to them.

19. Under source for the TableName and SchemaName give the below expression.

Exp: @item().table_name

Exp: @item().table_schema

The screenshot shows the 'Source' blade for a dataset named 'Dynamic_Src'. The 'Source' tab is selected. Under 'Source dataset', 'Dynamic_Src' is selected. In the 'Dataset properties' section, there are two entries: 'TableName' with value '@item().table_name' and 'SchemaName' with value '@item().table_schema'. Below this, there are options for 'Use query' (set to 'Table') and 'Preview data'.

20. Under sink click on new.

The screenshot shows the 'Sink' blade. The 'Sink' tab is selected. Under 'Sink dataset', there is a dropdown menu with 'Select...' and a '+ New' button.

21. Select Azure Blob Storage and click on continue. Then click on delimited text and click on continue.

22. Give the Name, Linked service, and Container name.

Set properties

The screenshot shows the 'Set properties' configuration for a sink dataset. It includes fields for 'Name' (Dynamic_sink), 'Linked service' (AzureBlobStorage_txt), 'File path' (f-demo), 'First row as header' (checked), and 'Import schema' (radio buttons for 'From connection/store', 'From sample file', and 'None').

23. Click on Open.

General Source **Sink** Mapping Settings User properties

Sink dataset * Open + New Learn more

Copy behavior

24. Under parameter create below three parameters.

Connection Schema **Parameters**

+ New Delete

Name	Type	Default value
TopFolder	String	Value
FolderName	String	Value
FileName	String	Value

25. Under connection, in the directory give the below expression.

Exp: @concat(dataset().TopFolder,dataset().FolderName)

26. And in the File Name give the below expression.

Exp: @dataset().FileName

Connection Schema Parameters

Linked service * Test connection Open + New Learn more

File path * / /

Compression type

Column delimiter

Row delimiter

27. Now go to the pipeline under the sink for the top folder and give Dynamic_Pipeline.

28. For the Folder Name give this expression: @item().table_name

29. For the File Name give this expression: @concat(item().table_schema, item().table_name)

General Source **Sink** Mapping Settings User properties

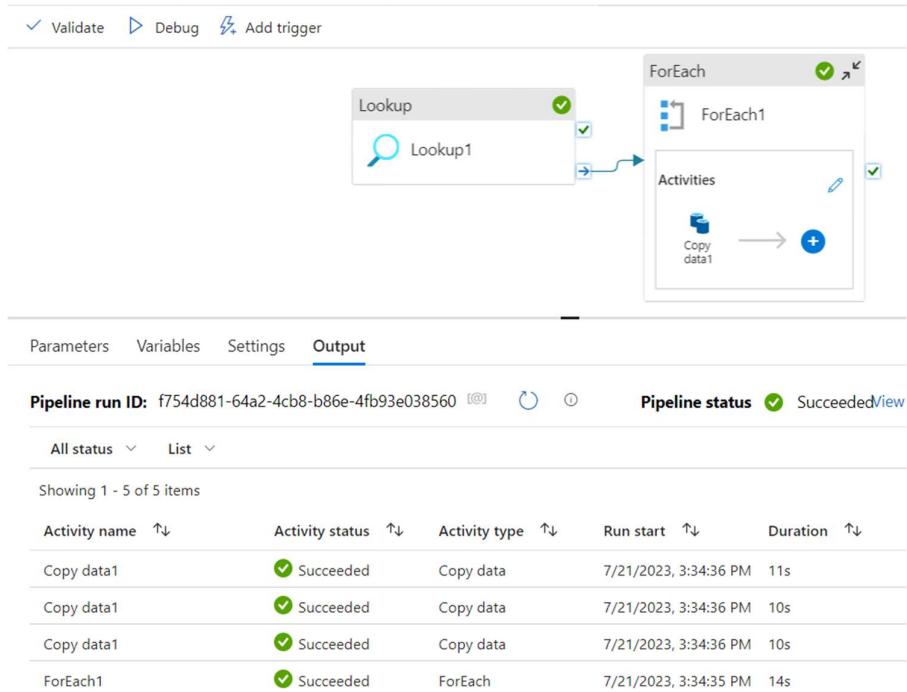
Sink dataset * Open + New Learn more

Dataset properties

Name	Value
TopFolder	Dynamic_Pipeline
FolderName	@item().table_name
FileName	@concat(item().table_schema, item()....)

30. Now validate the pipeline and click on debug.

31. Our pipeline was executed successfully.



32. Now go to container and you will see three folders as shown below. Each folder has a table.

Upload Change access level Refresh | Delete Change

Authentication method: Access key ([Switch to Azure AD User Account](#))

Location: f-demo

Search blobs by prefix (case-sensitive)

Add filter

Name	Modified
<input type="checkbox"/> Dynamic_Pipelinedept	
<input type="checkbox"/> Dynamic_Pipelineemp	
<input type="checkbox"/> Dynamic_PipelineTotalSale	