

Project Title : A. Create a pipeline to copy customer data from db to adls only if record count is more than 500. Once a data get copy it should call a child pipeline (which will copy the product data from table if customer record count is > 600).
 B. Design the pipeline in such a manner that it will pass the Customer pipeline pass the customer count to the child product pipeline via Pipeline parameter.

Step 1 : setting up Self-hosted IR to enable secure on-prem-to-cloud transfer.

Step 2: Configure Microsoft Integration Runtime with Azure Data Factory .

Step 2: Add Linked Services of Mysql Databases and Add Mysql Table Databases into Datasets.

Microsoft Azure | Data Factory > CsfFetch

Linked services

Showing 1 - 3 of 3 items

Name	Type
AzureDataLakeStorage1	Azure Data Lake Storage
HttpServer1	HTTP
MySqlConnect	MySQL

Edit linked service

Name *: MySqlConnect

Description:

Connect via integration runtime *: IntegrationRuntimeSystem

Server name *: 127.0.0.1

Port: 3306

Database name *: customers

User name *: root

Password: (Azure Key Vault)

SSL mode: Preferred

Use system trust store Not use system trust store

Save **Cancel** **Test connection**

Microsoft Azure | Data Factory > CsfFetch

Factory Resources

Datasets

- DelimitedText1
- Json1
- Json2
- Json3
- MySqlTable1
- MySqlTable2

Connection Parameters

Linked service *: MySqlConnect

Integration runtime *: IntegrationRuntimeSystem

Table: 'customers'

Preview experience: Off

Step 3: Create a Pipeline and add look up activity set dataset as Mysql table and perform Query to count the Row on tables.

Microsoft Azure | Data Factory > CsfFetch

Activities

CustomerTable

Activities

- Move and transform
- Synapse
- Azure Data Explorer
- Azure Function
- Batch Service
- DataBricks
- Data Lake Analytics
- General
- HDIInsight
- Iteration & conditionals
- Machine Learning
- Power Query

Properties

General

Name *: CustomerTable

Description:

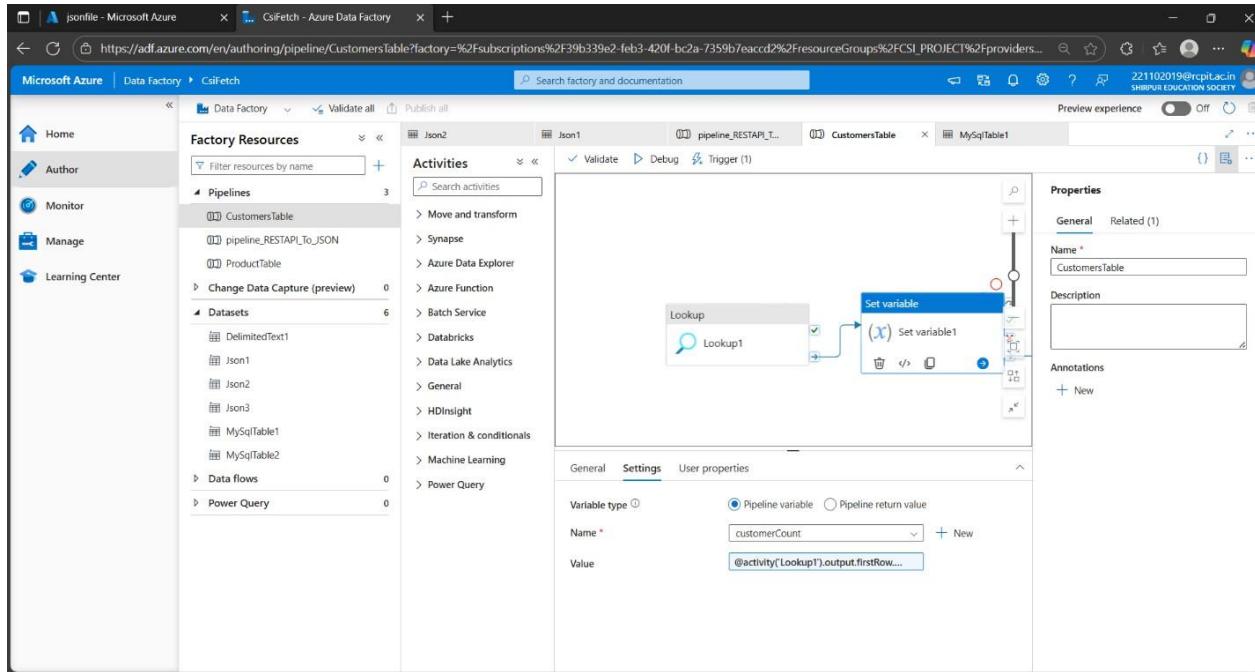
Source dataset *: MySqlTable1

First row only:

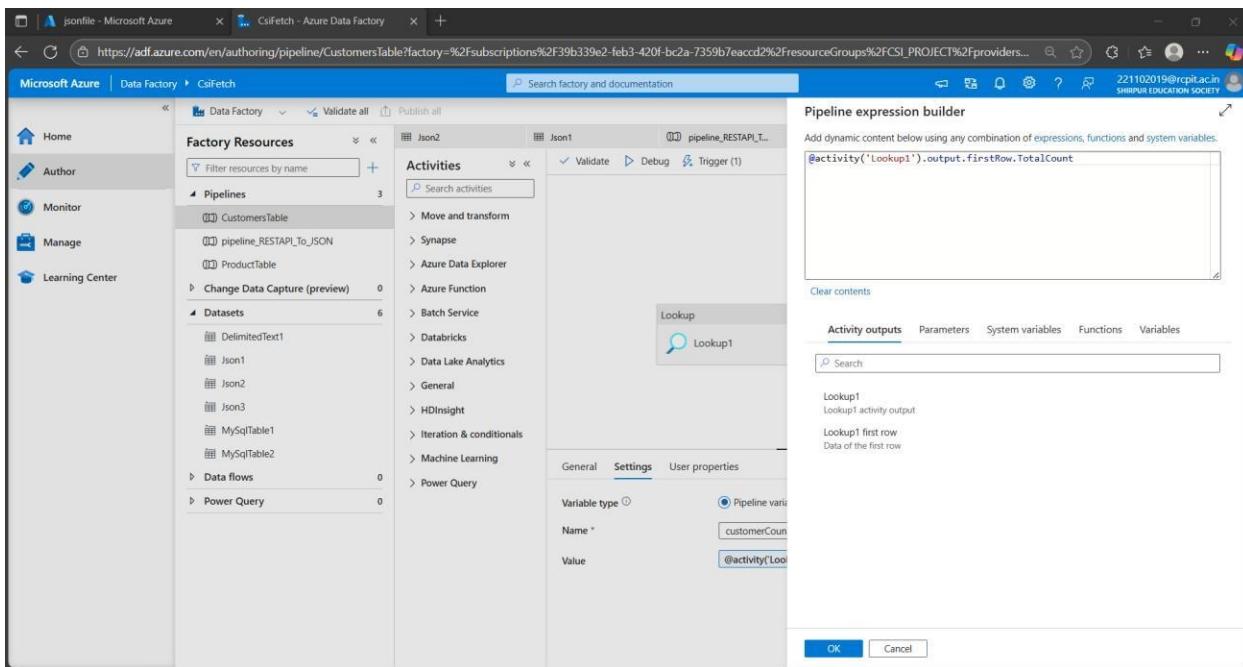
Use query: Table Query

Query: `SELECT COUNT(*) AS TotalCount FROM `customers``

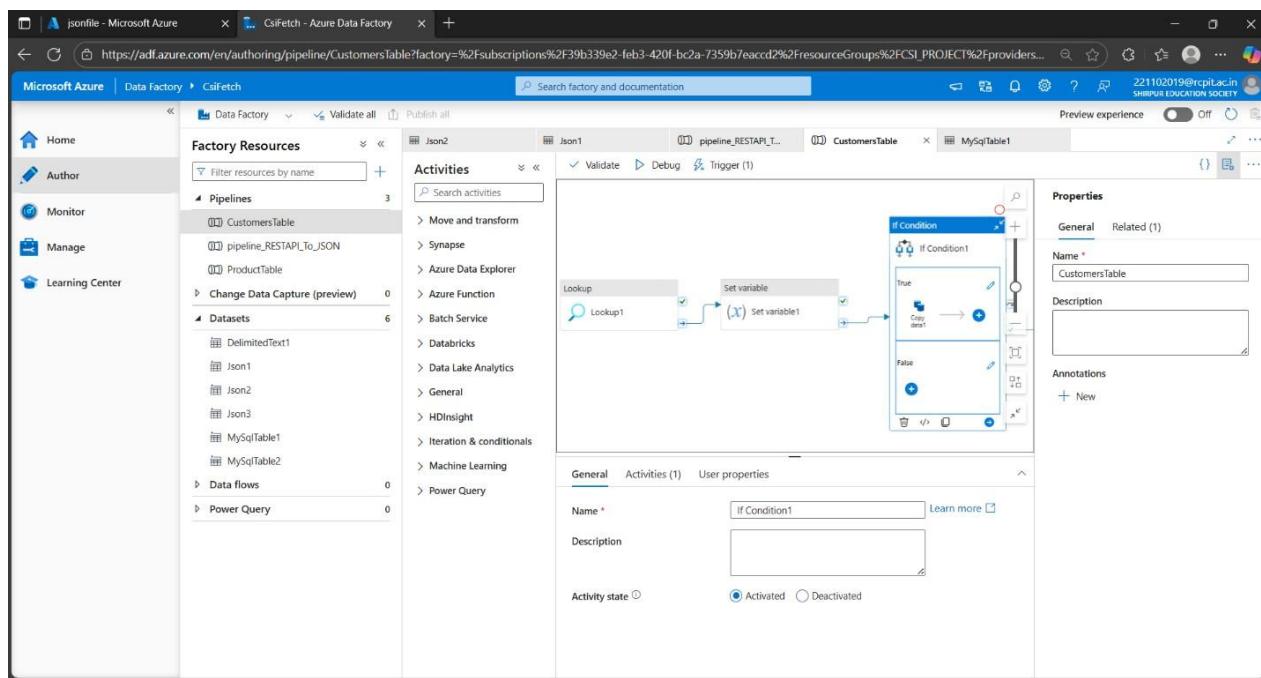
Step 4: Use Set Variable activity to Store Row Count as variable in Set variable activity.



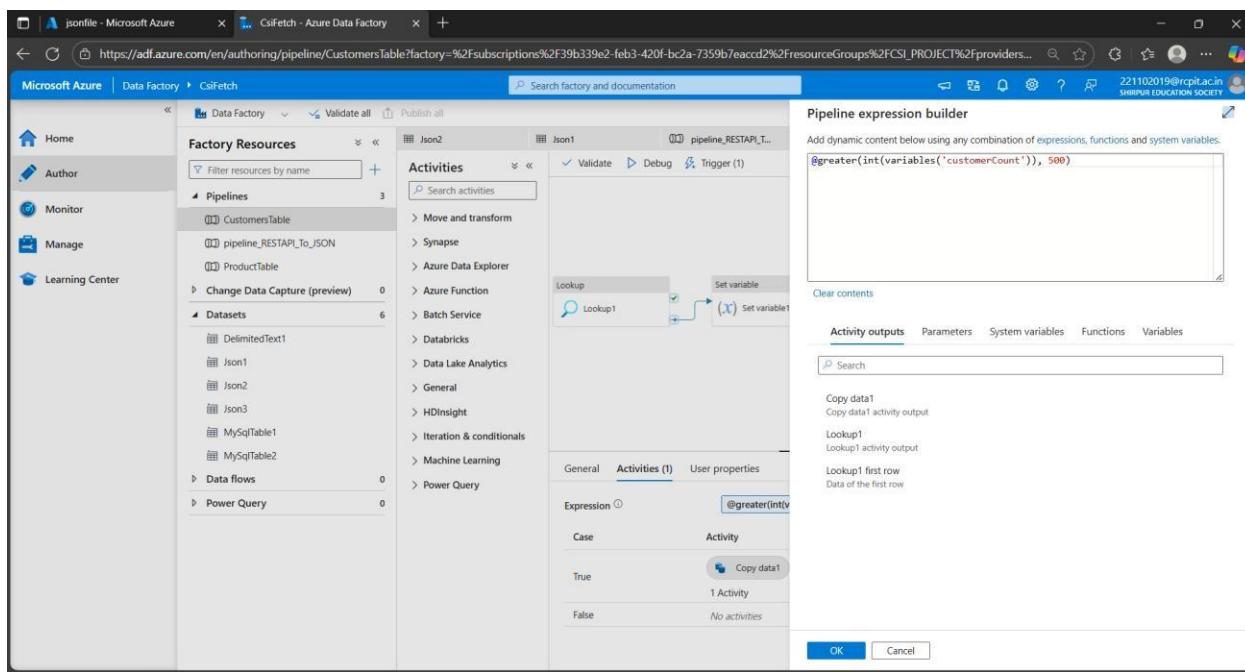
Add Pipeline Expression at Set variable activity



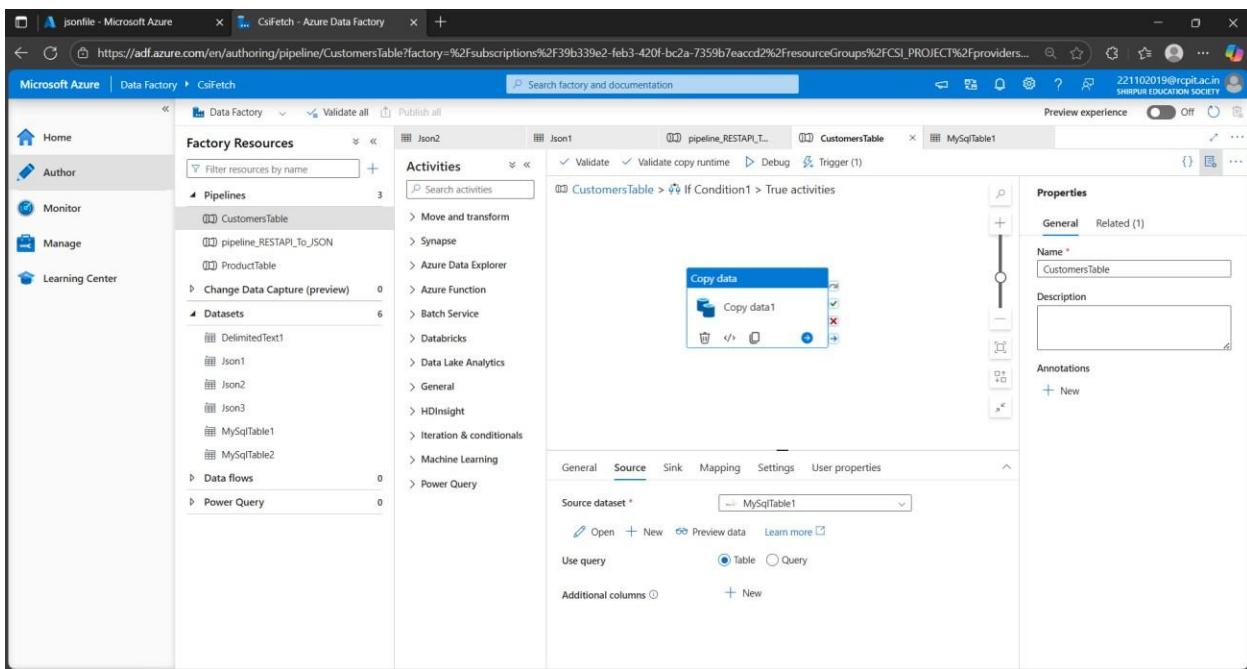
Step 5 : Implement IF Condition activity to check Condition Row Count >500.



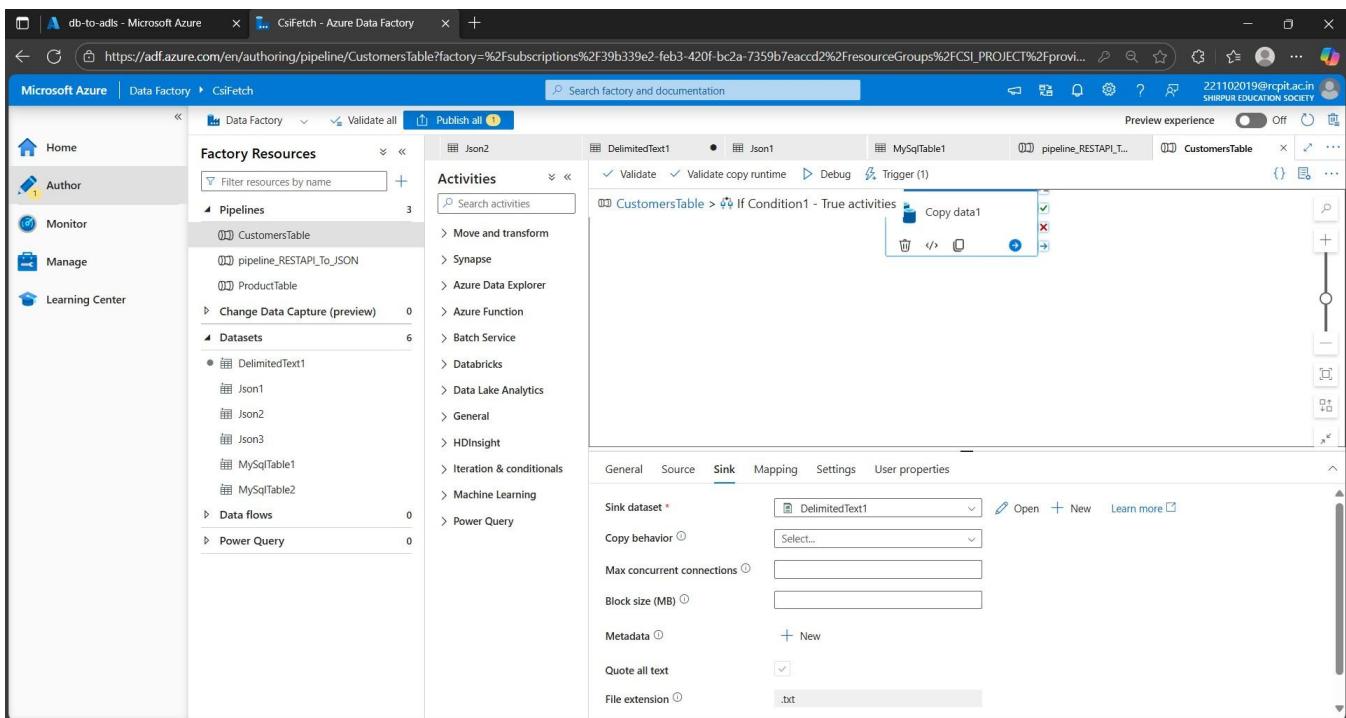
Step 6: Set If Activity Condition as Expression Builder



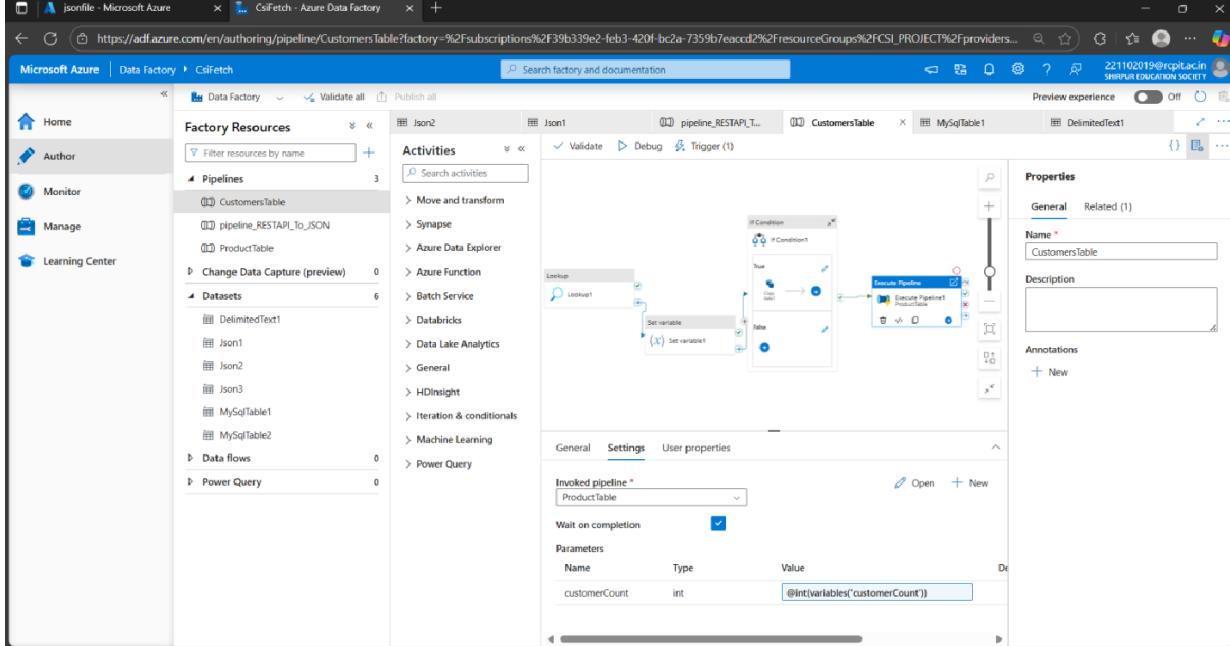
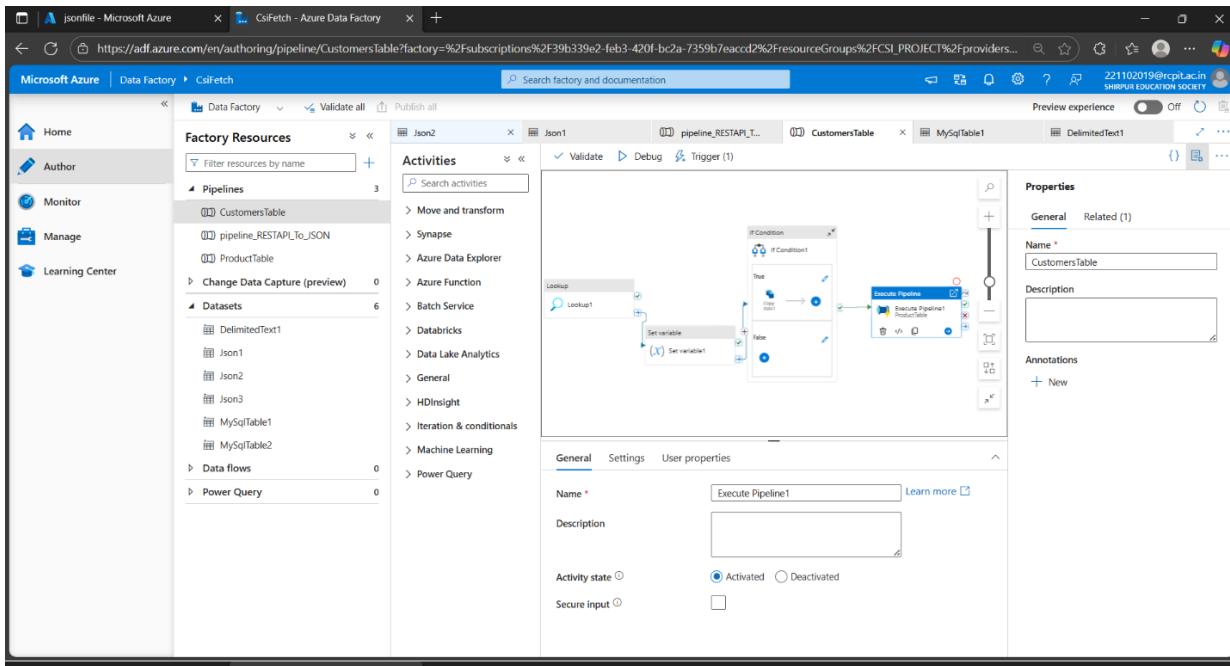
Step 7: Inside If Activity implement Copy Activity and Source Input set MySqlTable



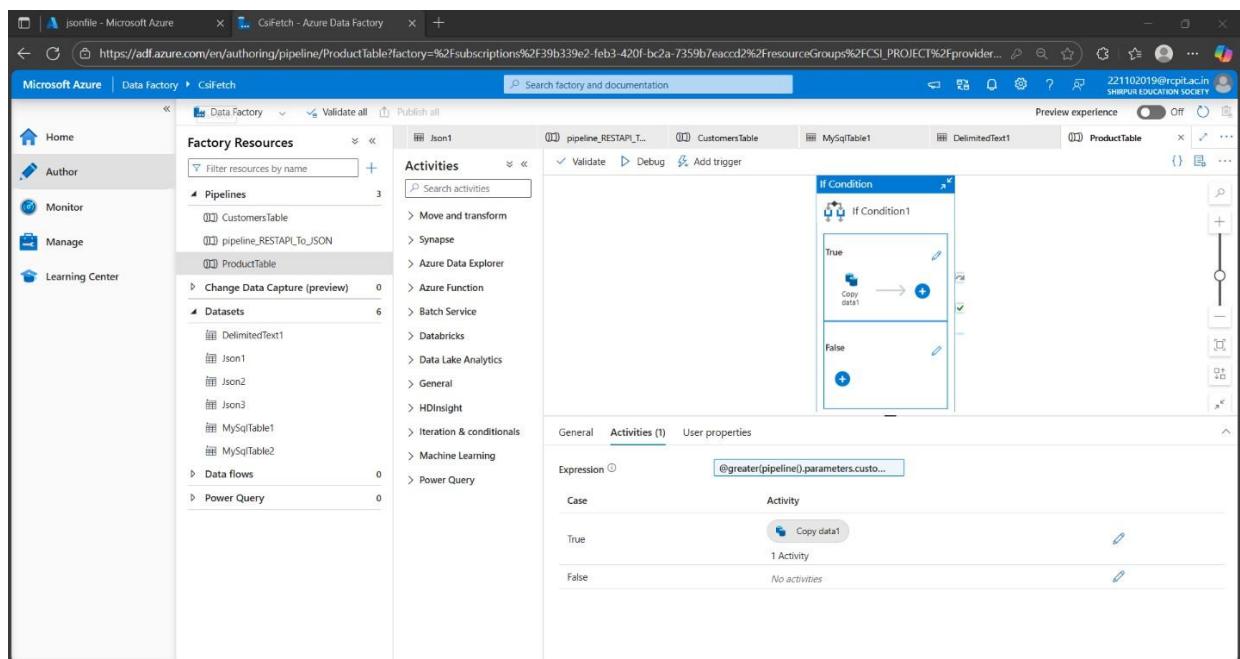
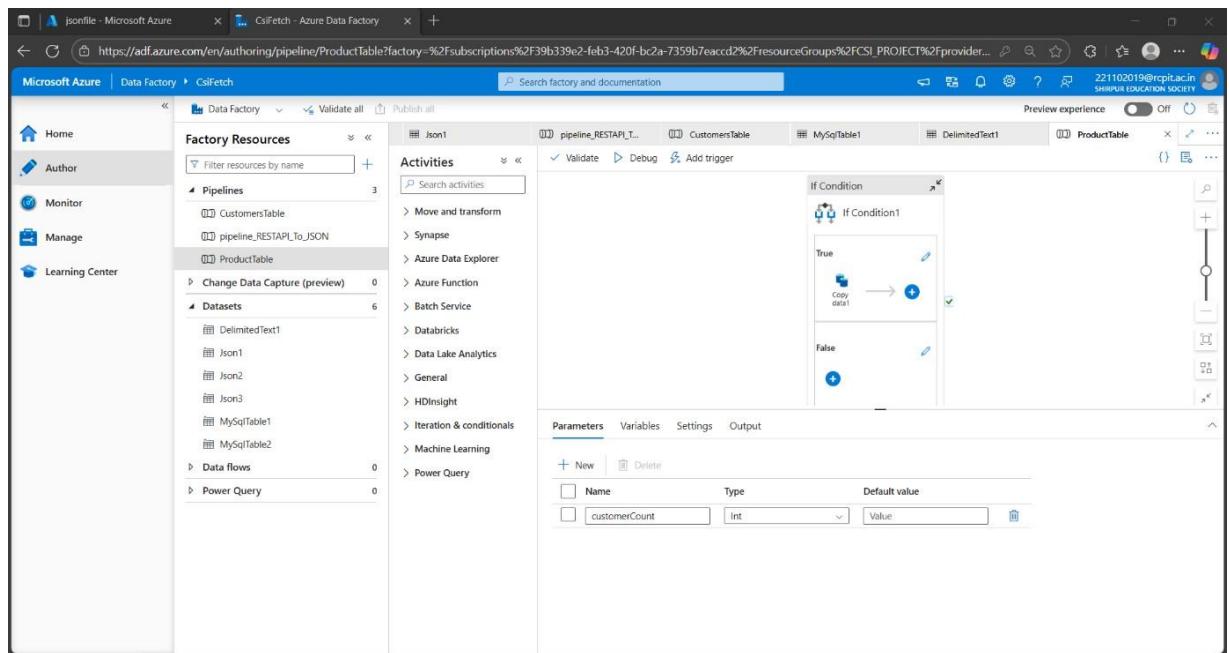
Step 8: Copy Activity Sink Set CSV File as Storage File on Azure Data Lake Gen2.



Step 9 :After IF Condition Activity Success Connect This into Execute Pipeline Activity to Invoke the ProductTable Pipeline Activity to pass Customer Count into Another pipeline that pipeline is ProductTable Pipeline and Run this pipeline .



Create ProductTable Pipeline and Set Parameter customerCount this same in CustomerTable Pipeline Execute Activity. And if condition check condition Row Count >600.



The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, and Data flows. In the main workspace, a pipeline named 'Json1' is selected. Under 'Activities', a 'Copy data' activity is highlighted. The 'Source' tab is active, showing 'Source dataset' set to 'MySqlTable2' and 'Use query' set to 'Table'. The 'Sink' tab is also visible.

This screenshot shows the same Azure Data Factory pipeline editor interface as the first one, but with different configuration for the 'Copy data' activity. The 'Sink' tab is now active, showing 'Sink dataset' set to 'Json3', 'Copy behavior' set to 'Select...', and other parameters like 'Max concurrent connections' and 'Block size (MB)'.

Step 10 : Execute Pipeline CustomerTable Pipeline are Execute as well Execute Activity to Execute Child Pipeline as passing parameter.

The screenshot shows the Azure Data Factory pipeline editor. The left sidebar lists 'Factory Resources' under 'Pipelines' with 'CustomersTable' selected. The main workspace displays the 'Activities' tab for the 'CustomersTable' pipeline. The pipeline consists of several activities: a 'Lookup' activity, a 'Set variable' activity, an 'If Condition' activity, a 'Copy data' activity, and another 'Execute Pipeline' activity. The 'Output' tab is open, showing pipeline run details. The pipeline run ID is 4026aaec-9f7b-4139-931f-7c6ac78eb2d, and the status is Succeeded. The run started at 7/15/2025, 7:25:50 PM, took 27s, and used the integrationRuntimeSystem.

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime
Execute Pipeline1	Succeeded	Execute Pipeline	7/15/2025, 7:25:50 PM	27s	
Copy data1	Succeeded	Copy data	7/15/2025, 7:25:21 PM	27s	integrationRuntimeSystem
If Condition1	Succeeded	If Condition	7/15/2025, 7:25:20 PM	29s	
Set variable1	Succeeded	Set variable	7/15/2025, 7:25:19 PM	Less than 1s	

This screenshot is identical to the one above, showing the execution of the 'CustomersTable' pipeline. The pipeline structure, activities, and execution results are the same, with the pipeline run ID 4026aaec-9f7b-4139-931f-7c6ac78eb2d and a successful outcome.

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime
Execute Pipeline1	Succeeded	Execute Pipeline	7/15/2025, 7:25:50 PM	27s	
Copy data1	Succeeded	Copy data	7/15/2025, 7:25:21 PM	27s	integrationRuntimeSystem
If Condition1	Succeeded	If Condition	7/15/2025, 7:25:20 PM	29s	
Set variable1	Succeeded	Set variable	7/15/2025, 7:25:19 PM	Less than 1s	

ProductTable (Child Pipeline) Pipeline Execute as Auto Passing Parameter into CustomerTable pipeline (Parent Pipeline)and Trigger

The screenshot shows the Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists 'Pipelines' containing 'CustomersTable', 'pipeline_RESTAPI_To_JSON', and 'ProductTable'. Under 'Activities', 'If Condition' is selected. The main workspace displays an 'If Condition' activity with two branches: 'True' leading to a 'Copy data' activity, and 'False' leading to another 'If Condition1' activity. Below the workspace, a table shows the pipeline run history:

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Copy data1	Succeeded	Copy data	7/15/2025, 7:25:52 PM	22s	integrationRuntimeSystem
If Condition1	Succeeded	If Condition	7/15/2025, 7:25:52 PM	23s	

Container of this all file are Store :

The screenshot shows the Azure Storage Explorer interface. A container named 'db-to-adls' is selected under 'db-to-adls'. The 'Overview' tab is active, showing blob details like 'CustomerTable', 'ProductPipeline', and 'concat('customer'. The table below lists the blobs:

Name	Last modified	Access tier	Blob type	Size	Lease state
CustomerTable	7/15/2025, 7:37:00 PM	Hot (Inferred)	Block blob	0	Available
ProductPipeline	7/15/2025, 11:57:39 AM	Hot (Inferred)	Block blob	0	Available
concat('customer	7/14/2025, 10:14:36 PM	Hot (Inferred)	Block blob	0	Available

CustomerTable(Parent Pipeline) Pipeline Output As CSV:

The screenshot shows the Microsoft Azure Storage Container interface for the 'CustomerTable/customer' blob. The container has one item named 'customer'. The preview pane displays the CSV data for 702 rows, starting from row 681. Each row contains three columns: CustomerID, Name, and Email.

CustomerID	Name	Email
680	"Customer_93"	"customer93@example.com"
681	"Customer_193"	"customer193@example.com"
682	"Customer_293"	"customer293@example.com"
683	"Customer_393"	"customer393@example.com"
684	"Customer_493"	"customer493@example.com"
685	"Customer_593"	"customer593@example.com"
686	"Customer_693"	"customer693@example.com"
687	"Customer_793"	"customer793@example.com"
688	"Customer_192"	"customer192@example.com"
689	"Customer_292"	"customer292@example.com"
690	"Customer_392"	"customer392@example.com"
691	"Customer_492"	"customer492@example.com"
692	"Customer_592"	"customer592@example.com"
693	"Customer_692"	"customer692@example.com"
694	"Customer_792"	"customer792@example.com"
695	"Customer_91"	"customer91@example.com"
696	"Customer_191"	"customer191@example.com"
697	"Customer_291"	"customer291@example.com"
698	"Customer_391"	"customer391@example.com"
699	"Customer_491"	"customer491@example.com"
700	"Customer_591"	"customer591@example.com"
701	"Customer_691"	"customer691@example.com"
702		

ProductTable(Child Pipeline) Pipeline Output as JSON:

The screenshot shows the Microsoft Azure Storage Container interface for the 'ProductPipeline/customers.json' blob. The container has one item named 'customers.json'. The preview pane displays the JSON data for 702 rows, starting from row 680. Each row is a JSON object with fields: CustomerID, Name, and Email.

CustomerID	Name	Email
680	"Customer_93"	"customer93@example.com"
681	"Customer_193"	"customer193@example.com"
682	"Customer_293"	"customer293@example.com"
683	"Customer_393"	"customer393@example.com"
684	"Customer_493"	"customer493@example.com"
685	"Customer_593"	"customer593@example.com"
686	"Customer_693"	"customer693@example.com"
687	"Customer_793"	"customer793@example.com"
688	"Customer_192"	"customer192@example.com"
689	"Customer_292"	"customer292@example.com"
690	"Customer_392"	"customer392@example.com"
691	"Customer_492"	"customer492@example.com"
692	"Customer_592"	"customer592@example.com"
693	"Customer_692"	"customer692@example.com"
694	"Customer_792"	"customer792@example.com"
695	"Customer_91"	"customer91@example.com"
696	"Customer_191"	"customer191@example.com"
697	"Customer_291"	"customer291@example.com"
698	"Customer_391"	"customer391@example.com"
699	"Customer_491"	"customer491@example.com"
700	"Customer_591"	"customer591@example.com"
701	"Customer_691"	"customer691@example.com"
702		