AZURE DATA FACTORY
CAPSTONE - COVID USE CASE

Name : Induja D (2319849)

Cohort Code : CSDAIA24AZ003

Introduction:

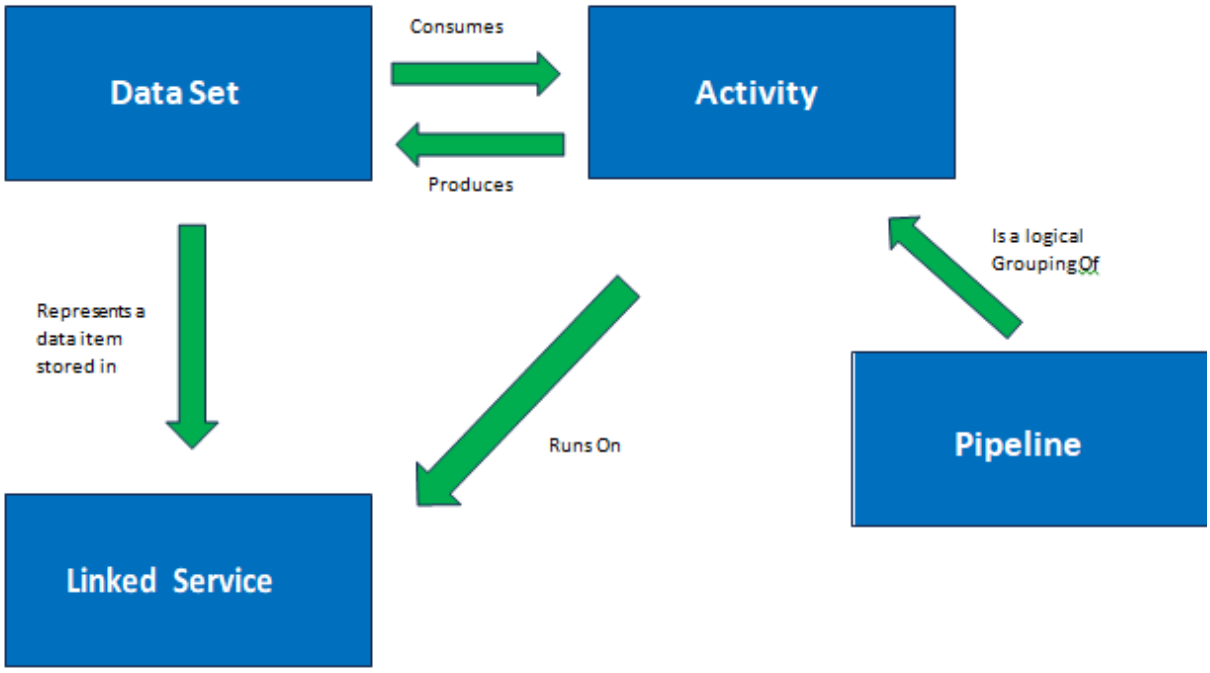
The "Azure Data Factory Capstone - Covid Use Case" project was completed in compliance with business requirements using a variety of tools, including Synapse, Azure Data Factory, and Azure Data Lake Storage. These Azure components facilitate the process of removing data from its source (Data Lake), transforming it to meet business requirements, and then loading it into a destination data warehouse. The "Azure Data Factory Capstone - Covid Use Case" system was developed to teach users how to use Azure cloud data services to build a practical data pipeline in Azure Data Factory (ADF) for the purpose of analyzing the covid trend across the regions. This case study will teach us anything.

1. How to use Azure Data Factory (ADF) to ingest data from flat filesinto Azure Data Lake Gen2 and Azure Synapse.
2. How to transform data using Data Flows in Azure Data Factory(ADF).

Objective:

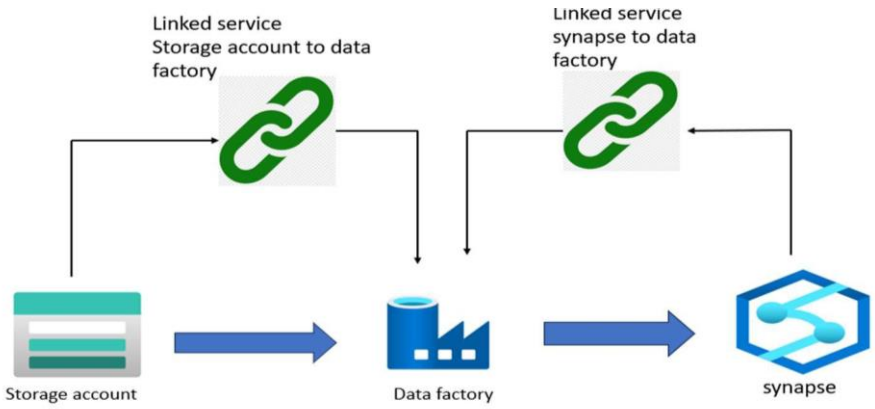
This project's main goal is to give us hands-on experience with storage, ADFpipelines, mapping data flows, and Azure Synapse. It also aims to teach us how to use Azure Data Factory (ADF) to ingest data from flat files into Azure Data Lake Gen2 and Azure Synapse, as well as how to transform data using Data Flows in Azure Data Factory (ADF) and load it into Azure data lakes. This report provides an overview of the entire project, enabling us to comprehend and analyze the Azure use case scenario and its applications.

Overview of Data Factory Flow:

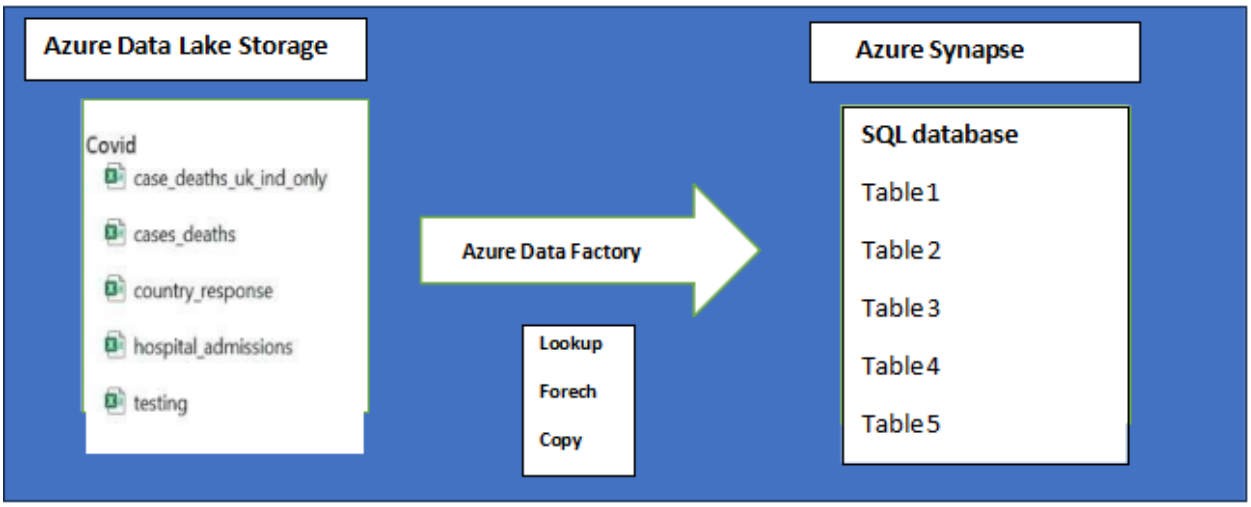


Project Requirements:

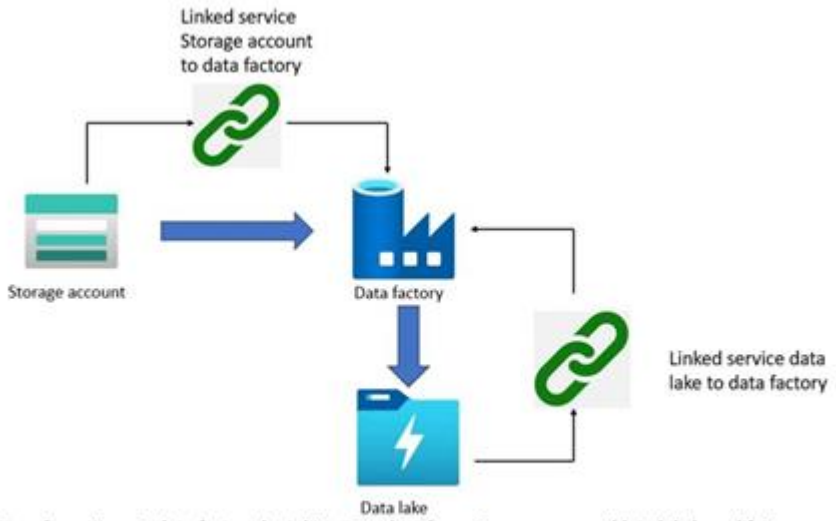
Requirement 1: Ingest raw flat files from Data Lake tosynapse (data warehouse)



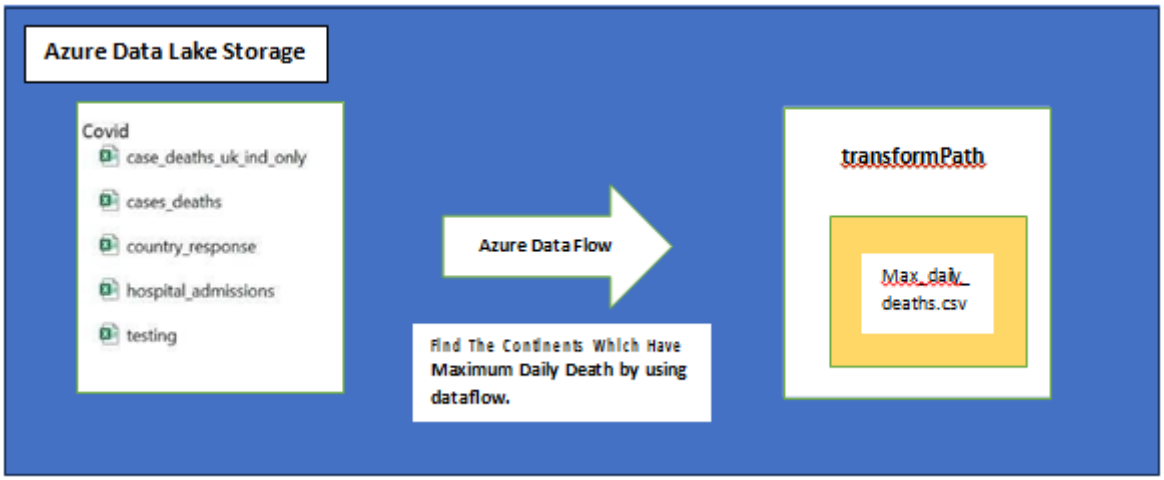
Ingesting process taken place in data factory by taking raw data from storage account(data lake) to relation database by creating tables in synapse(data warehouse)

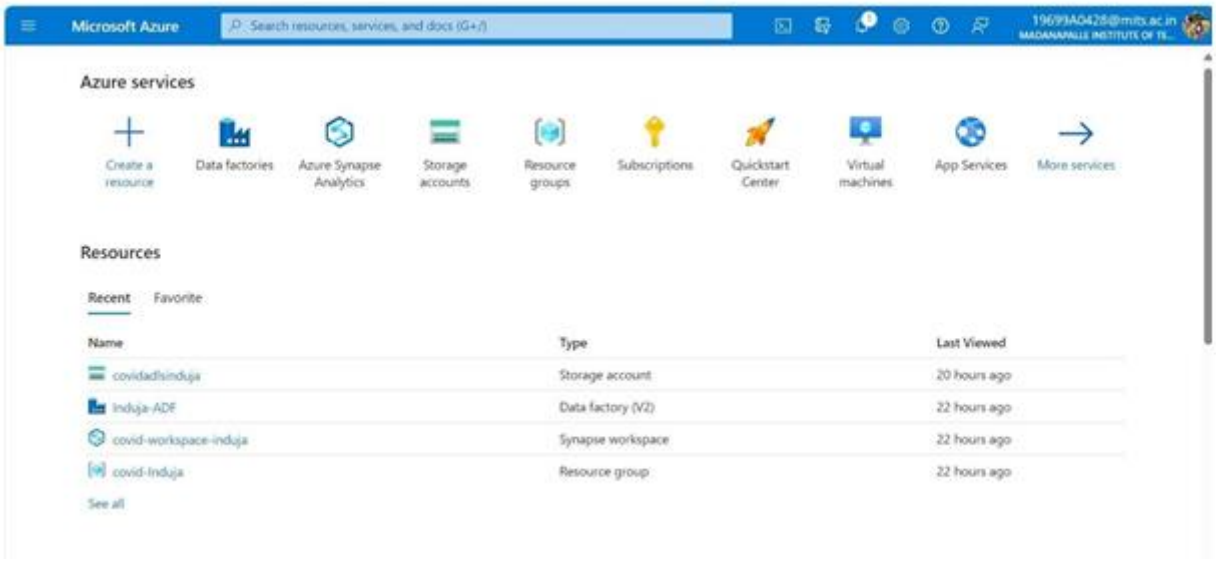


Requirement 2: Transform data using Data Flows in AzureData Factory (ADF)

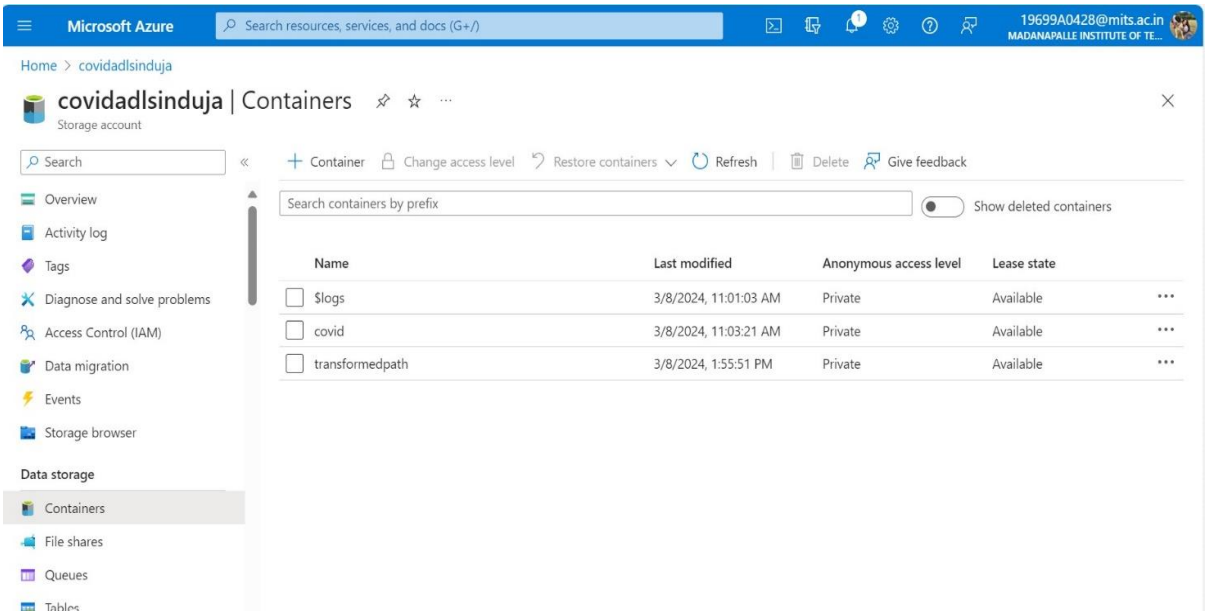


Data Transformation in data factory by taking Raw data from Storage account(data lake) modifying as per over requirements given in project and storing that file in storage account(data lake)

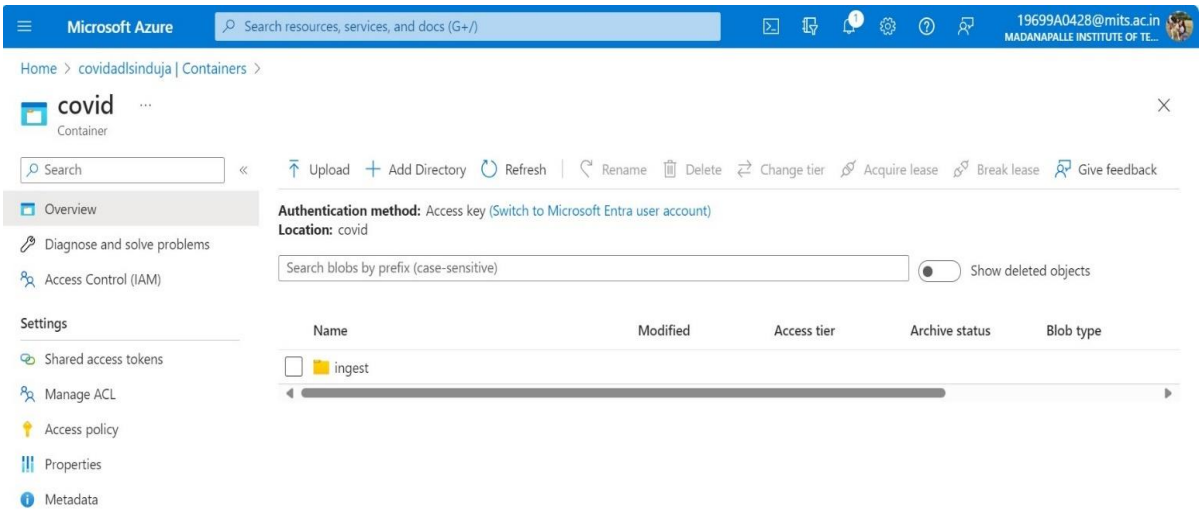


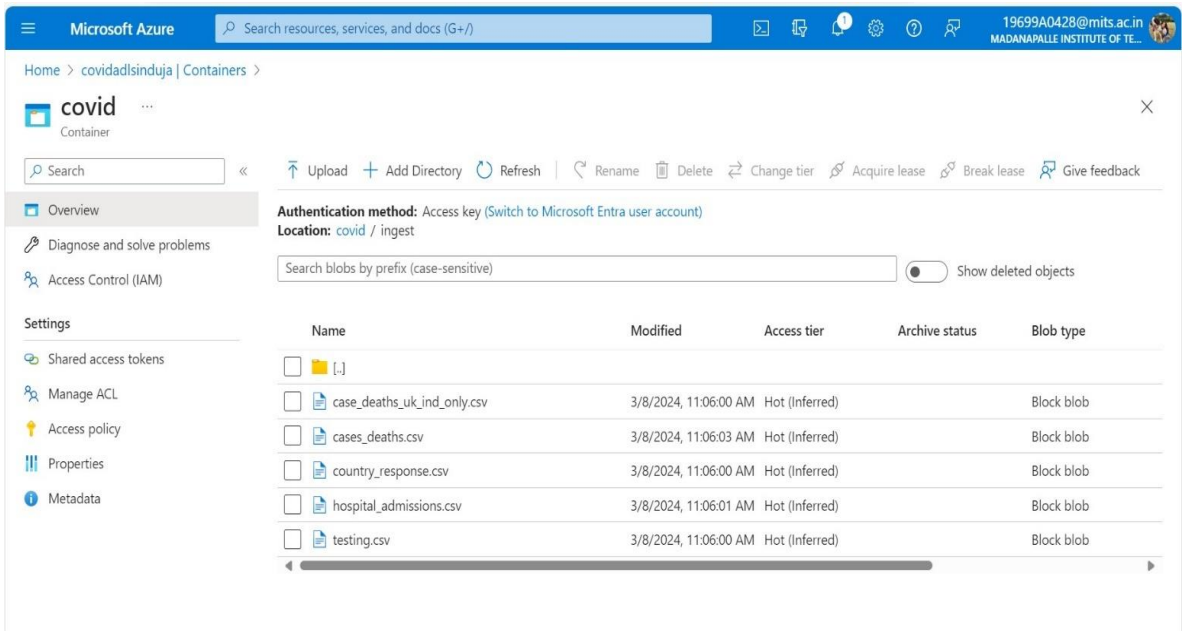


Step 1: Created one Resource group(**covid-Induja**) and required resources for project like Storage account(**covidadlsinduja**), Synapse workspace (data warehouse) (**covid-workspace-induja**), Azure Datafactory(**Induja-ADF**).

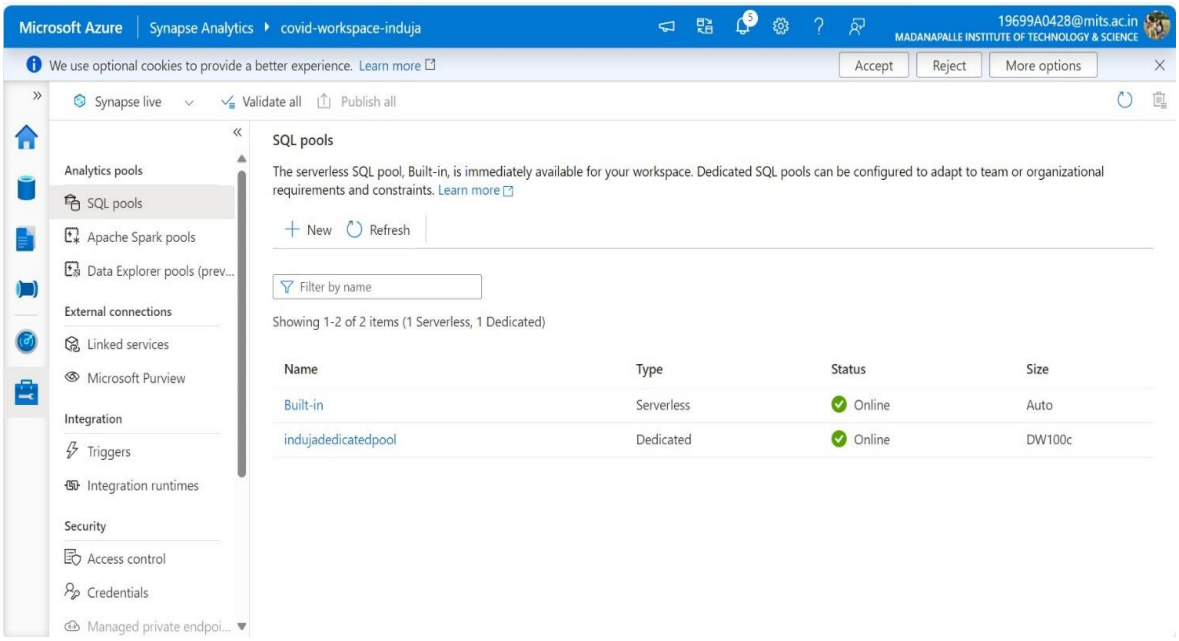


Step 2: Created Container with name "**covid**" in Storage account (Data Lake) for holding folder which contains flat files.





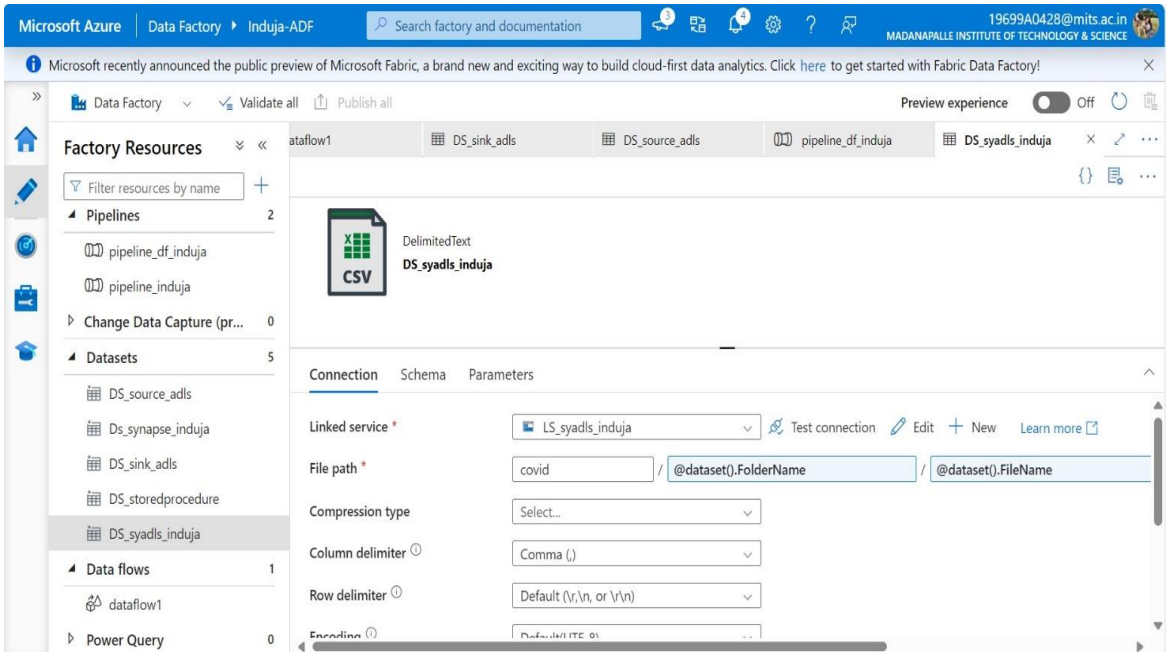
Step 3: Create Folder with name "**ingest**" inside the Container "covid" and uploaded csv files (data sets) form personal computer which are there in zip file given in project document.



Step 4: Create **Azure synapse resource** and one **dedicated pool** inside the azure synapse for data warehouse creation and it should be turn on.

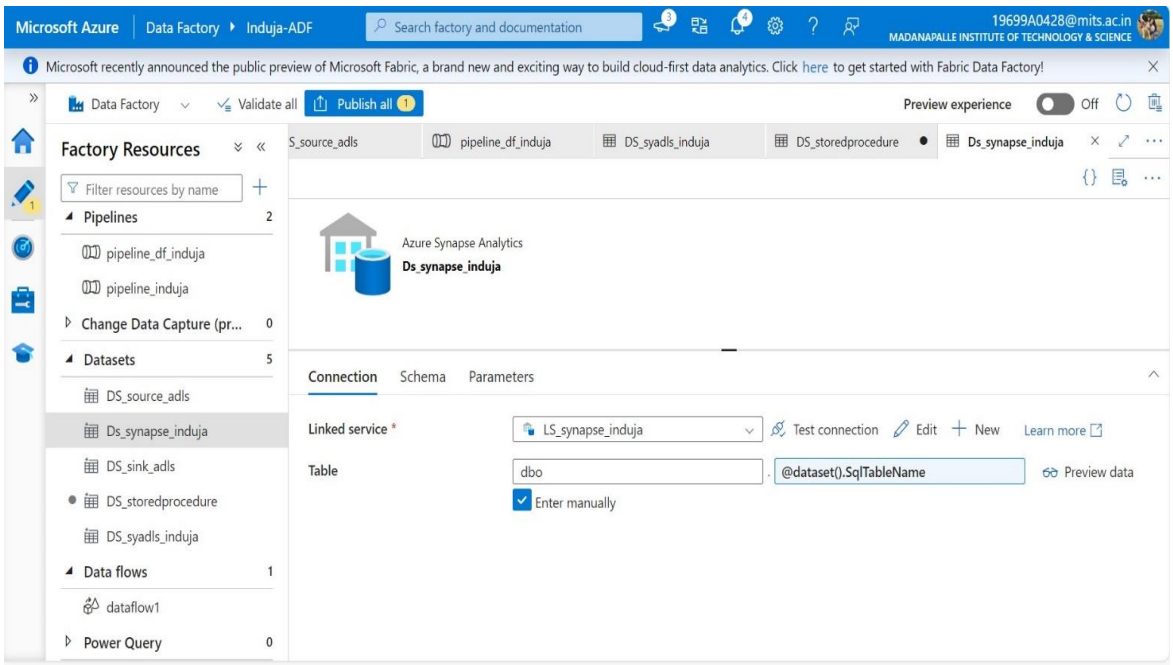
Step 5: Creating **two linked services** as per the project requirement in Azure data factory.

- **Storage account (Data Lake) to Azure data factory.**
- **Azure synapse workspace to Azure Data factory**.



Step 6: Created **dataset (DS_syadls_induja)** for fetching flat files from storage account (data lake) which are present in ingest folder inside covid container.

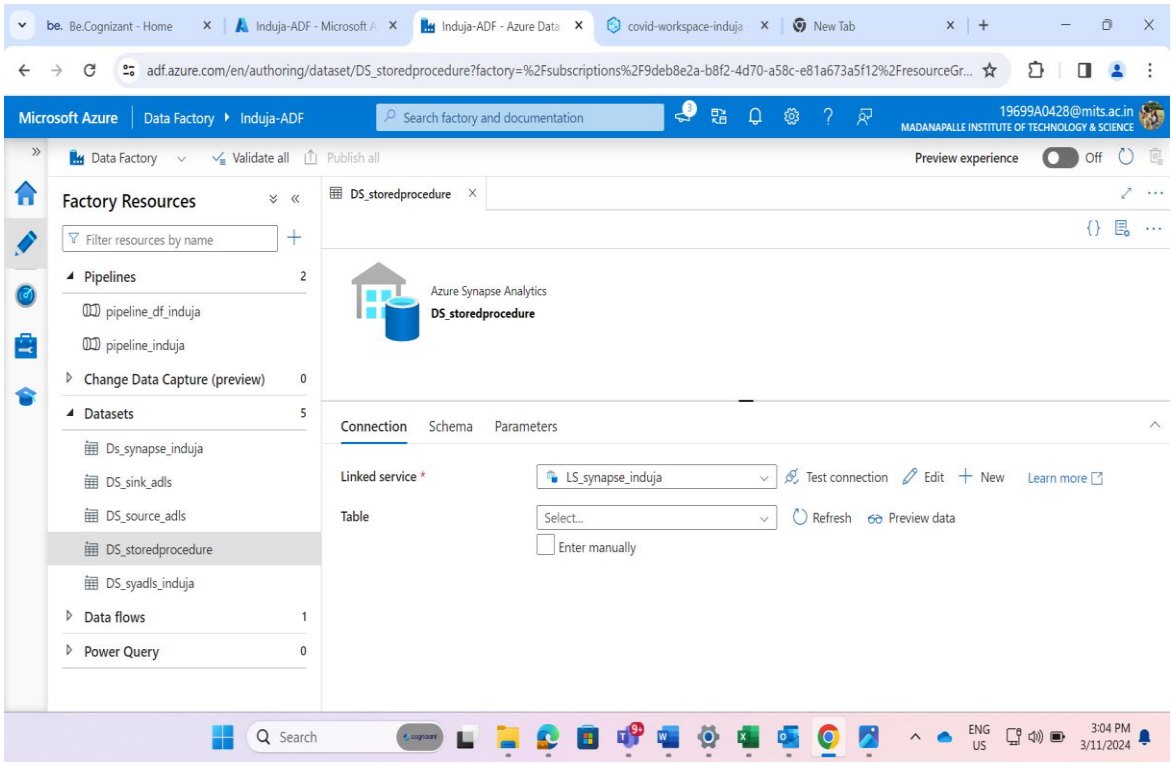
Step 7: Created requried **SQL Tables** in synapse SQL database by writing create table queries in Synapse workspace (SQL Script).

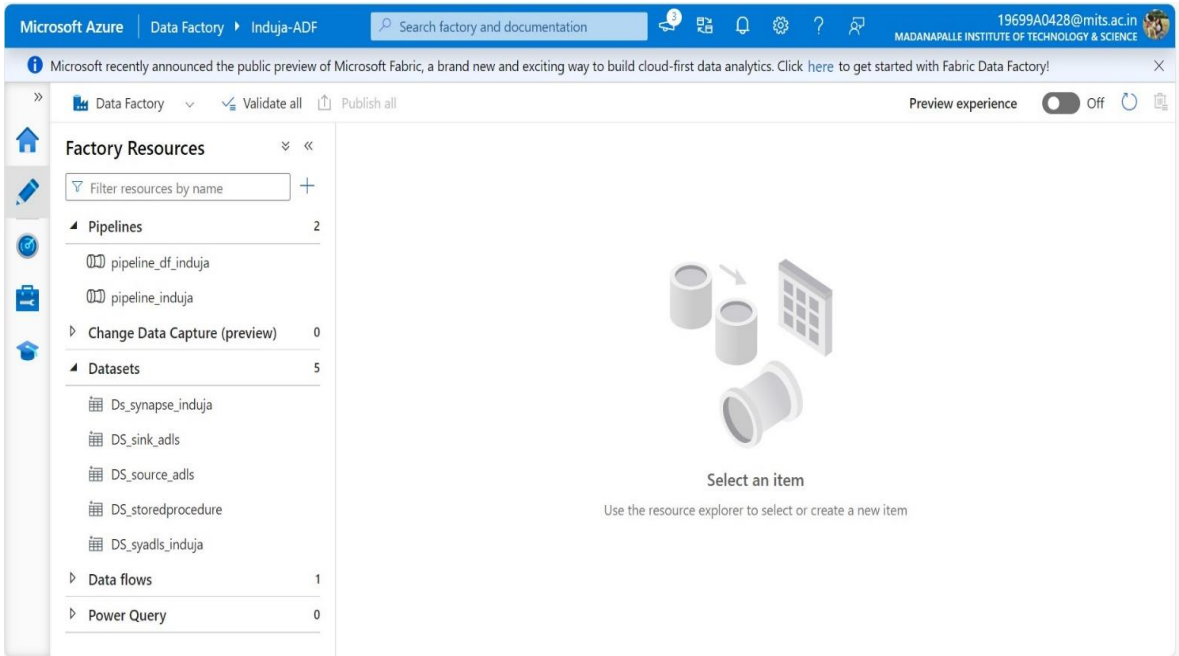


Step 8: Created **dataset (Ds_synapse_induja)** for inserting into SQL Tables created in synapse (data warehouse).

Step 9: Created **stored procedure(sp_GetConfigSrcFileNames)** for fetching records from parameters table.

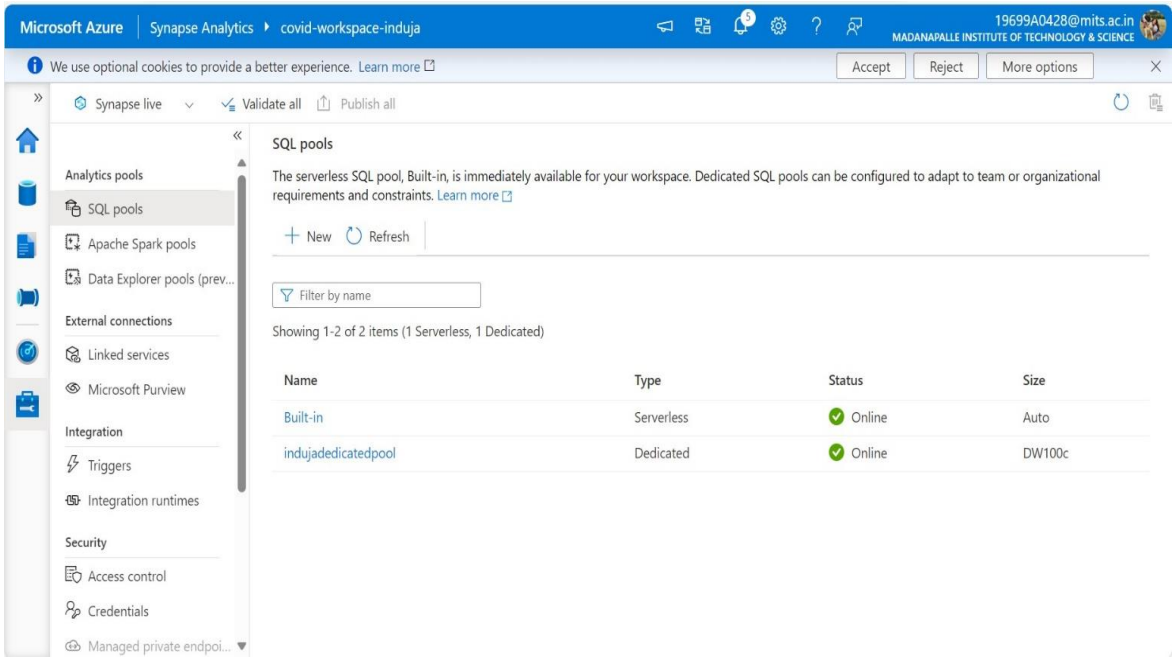
Step 10: Created **dataset(DS_storedprocedure)** for fetching parameters table form synapse(data warehouse).


Microsoft Azure
Data Factory
Induja-ADF
Search factory and documentation
19699A0428@mits.ac.in
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 2
pipeline_df_induja
pipeline_induja
Change Data Capture (preview) 0
Datasets 5
Ds_synapse_induja
DS_sink_adls
DS_source_adls
DS_storedprocedure
DS_syadls_induja
Data flows 1
Power Query 0
DS_storedprocedure
Azure Synapse Analytics
DS_storedprocedure
Connection
Schema
Parameters
Linked service *
LS_synapse_induja
Test connection
Edit
New
Learn more
Table
Select...
Refresh
Preview data
Enter manually
Search
ENG
US
3:04 PM
3/11/2024

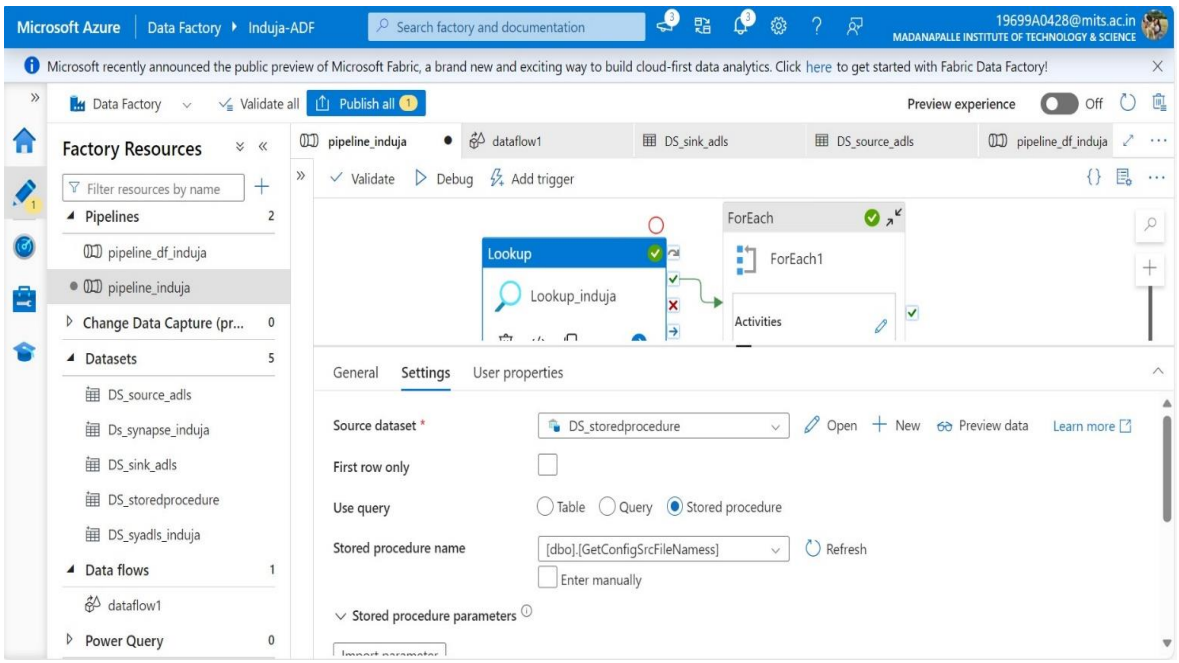

Microsoft Azure
Data Factory
Induja-ADF
Search factory and documentation
19699A0428@mits.ac.in
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 2
pipeline_df_induja
pipeline_induja
Change Data Capture (preview) 0
Datasets 5
Ds_synapse_induja
DS_sink_adls
DS_source_adls
DS_storedprocedure
DS_syadls_induja
Data flows 1
Power Query 0
Select an item
Use the resource explorer to select or create a new item

Step 11: Creating required datasets in Azure data factory.

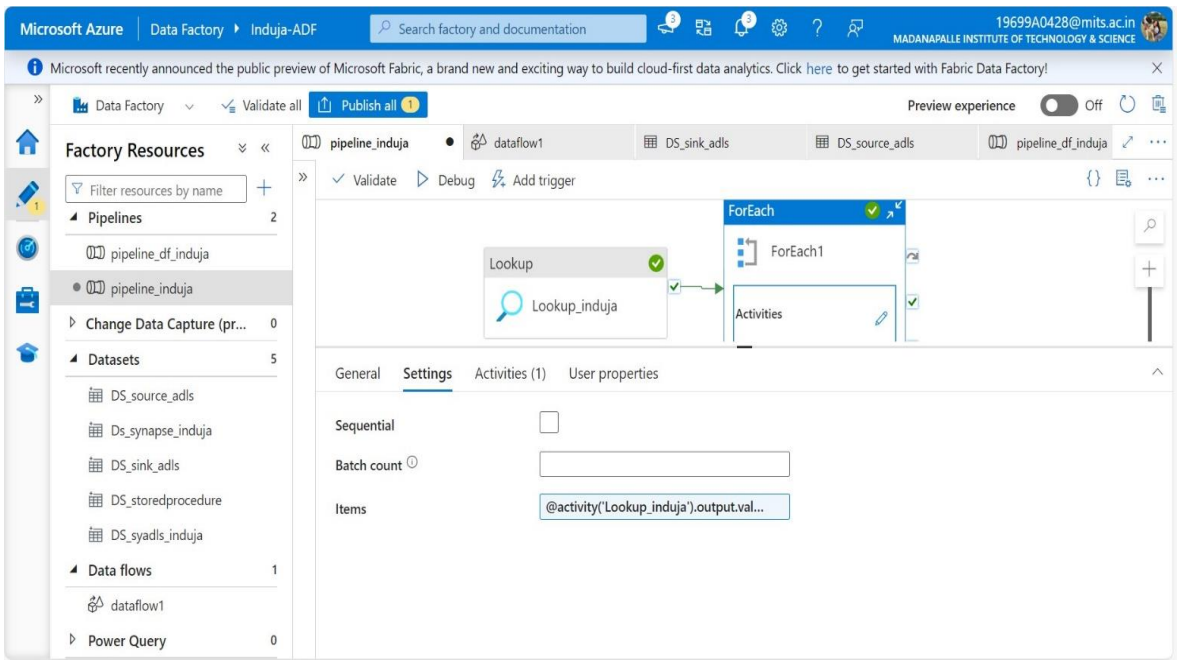
- To fetch covid flat files from Storage account (data lake) **(DS_syadlsinduja).**
- To fetch parameters table from Synapse (data warehouse) **(DS_storedprocedure).**
- To insert data into SQL tables in Synapse (data warehouse) **(DS_synapse_induja).**
- To fetch cases_deaths.csv file from storage account (data lake) **(DS_source_adls).**
- To insert data transformation done file by using dataflow into storage account (data lake) **(DS_sink_adls)**



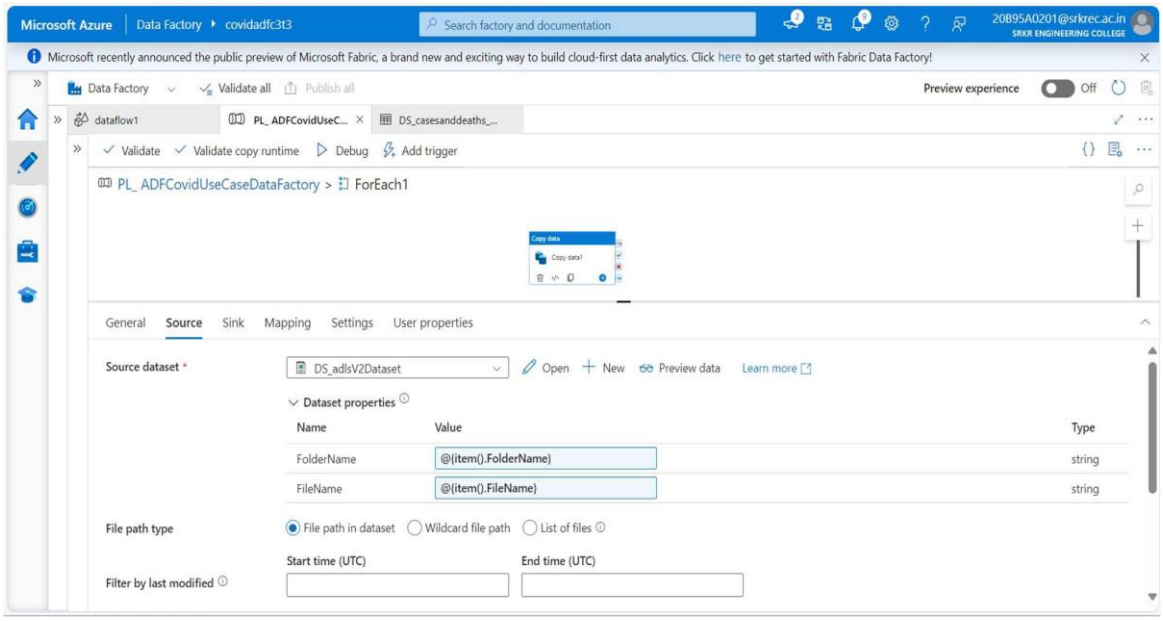
Step 12: Before creating pipeline in data factory, we need **to turn on the dedicated pool**. we need to check these two to three times while moving on to creation of pipeline.

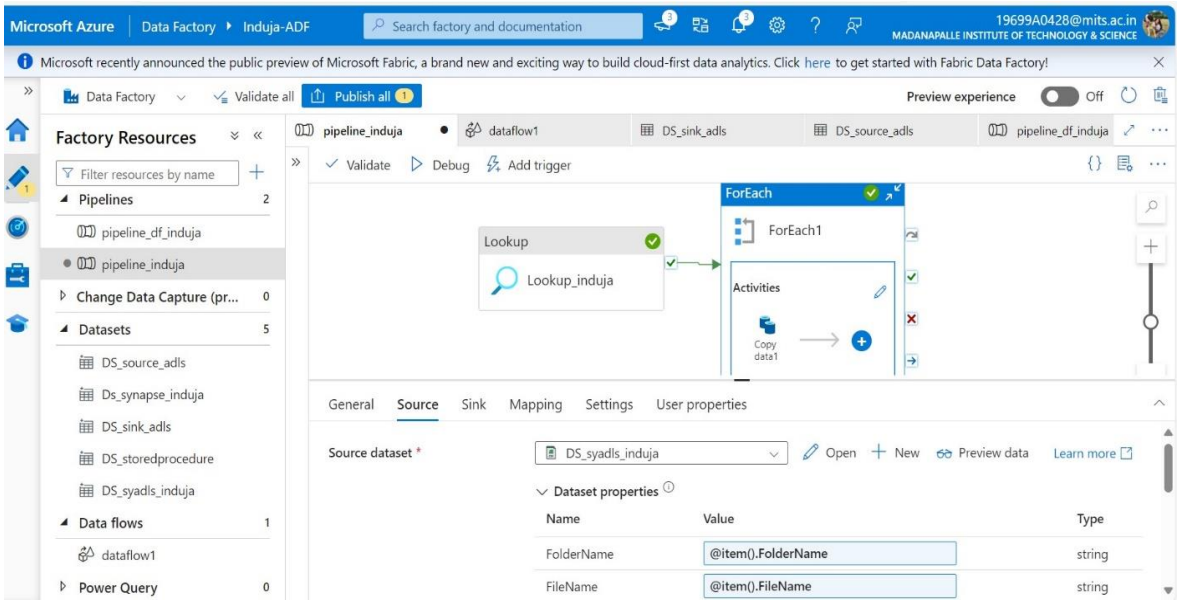


Step 13: Create a **pipeline** (**Pipeline_induja**) and Drag and Drop the **Look up Activity** into pipeline workspace and set the **source dataset (DS_storedprocedure)** for Lookup and choose the option **stored procedure** and given the stored procedure name created in synapse (data warehouse).



Step 14: Drag and Drop **For each activity** in pipeline workspace and configure the for each activity settings like **Items** with output of look up activity **(@activity('Lookup1'). Output. Value).**

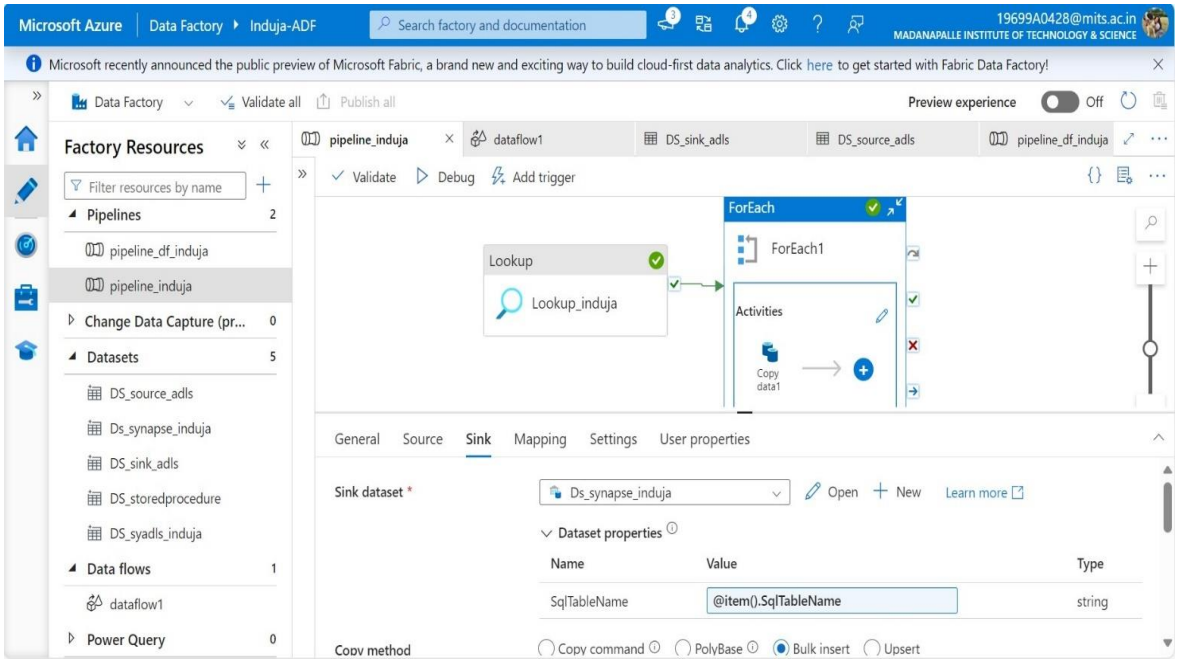




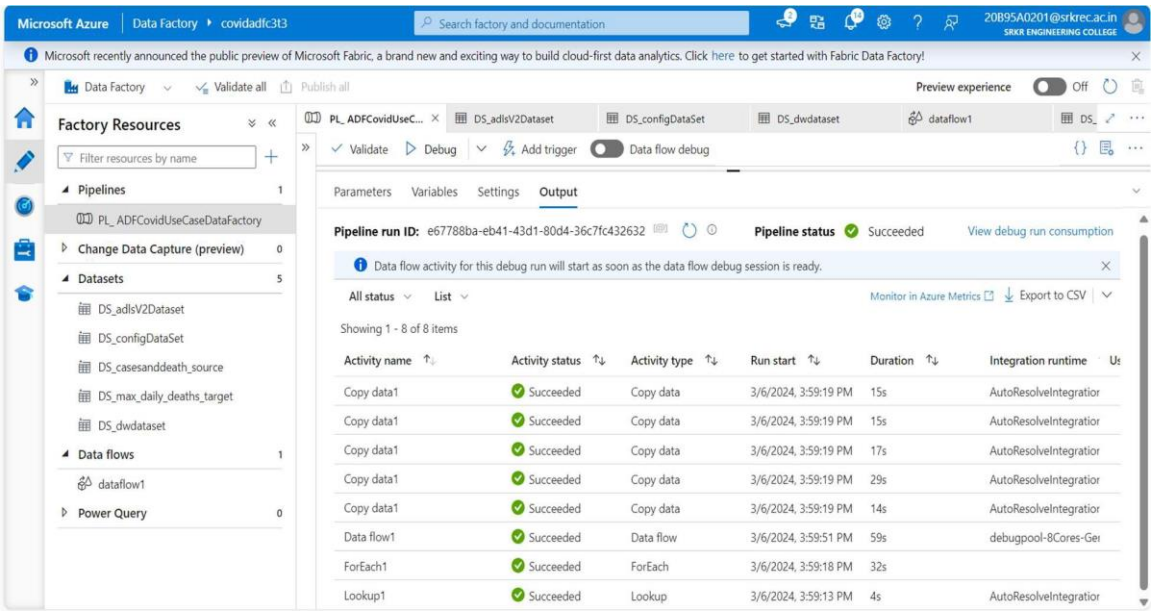
Step 15: Click on add activity symbol present on foreach activity and inside foreach activity add a **Copy activity** for copy data from CVS file into SQL table.

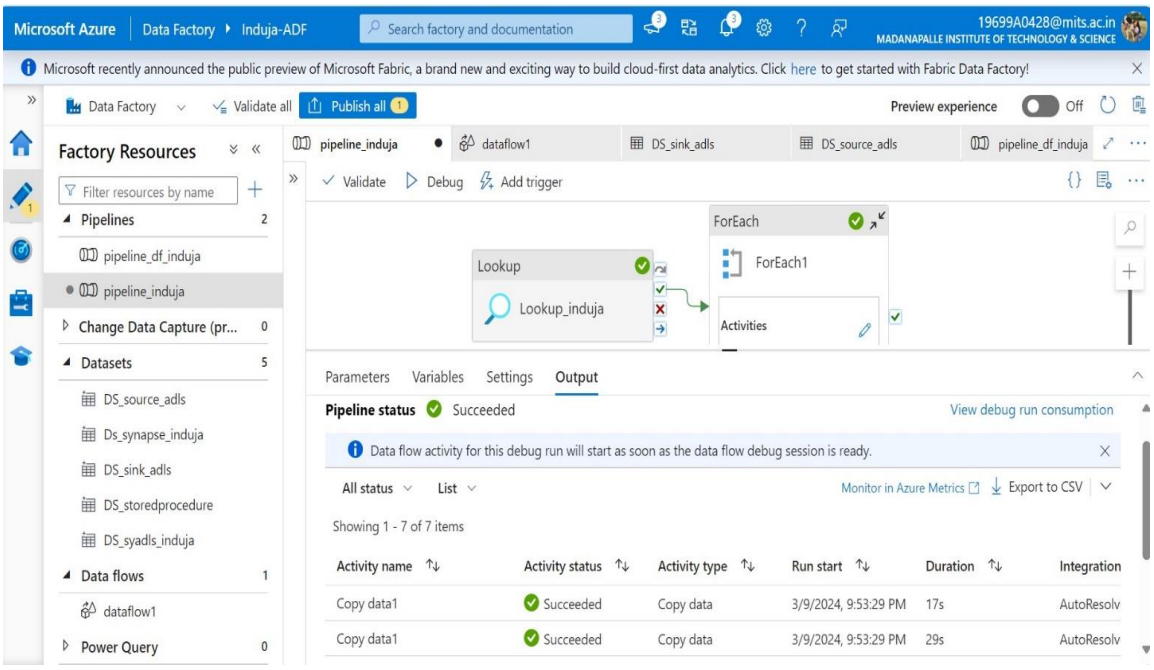
Configure settings at source side in copy activity by giving dataset **(DS_syadls_induja)** and giving folder name and file name dynamic by taking from foreach activity by **item.**

Folder Name **(@ {item (). FolderName}),** File Name **(@ {item (). FileName})**

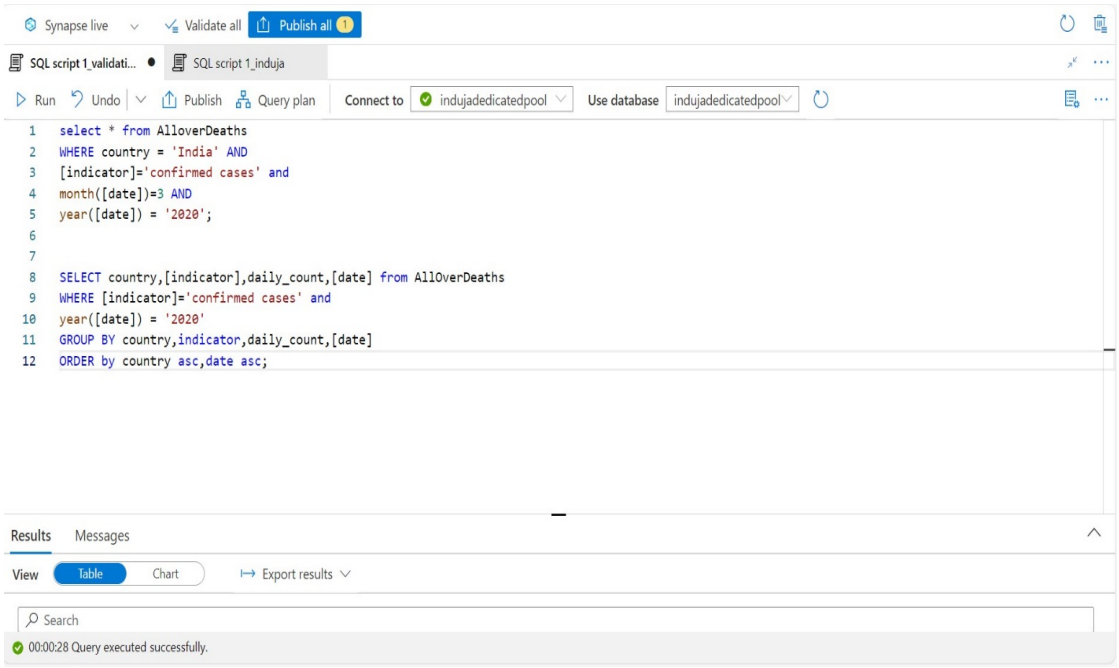


Step 16: Configure setting in copy activity at sink side by giving dataset **(DS_synapse_induja)** and giving sqltableName dynamically by taking from foreach activity by item sqltableName **(@ {item (). sqltableName})**

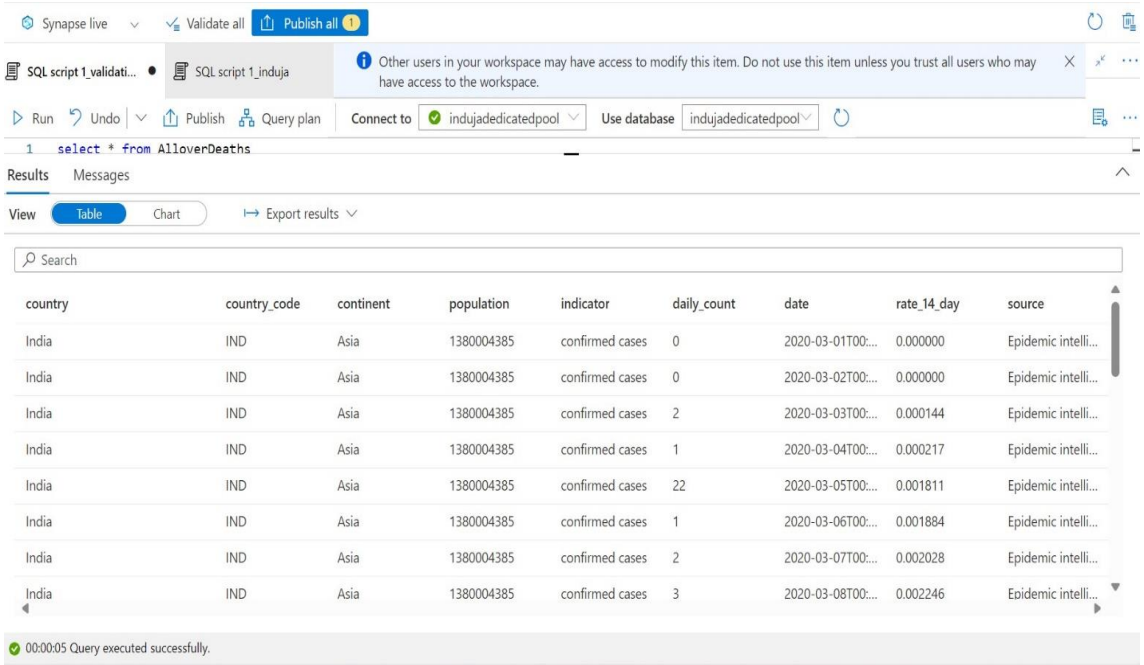




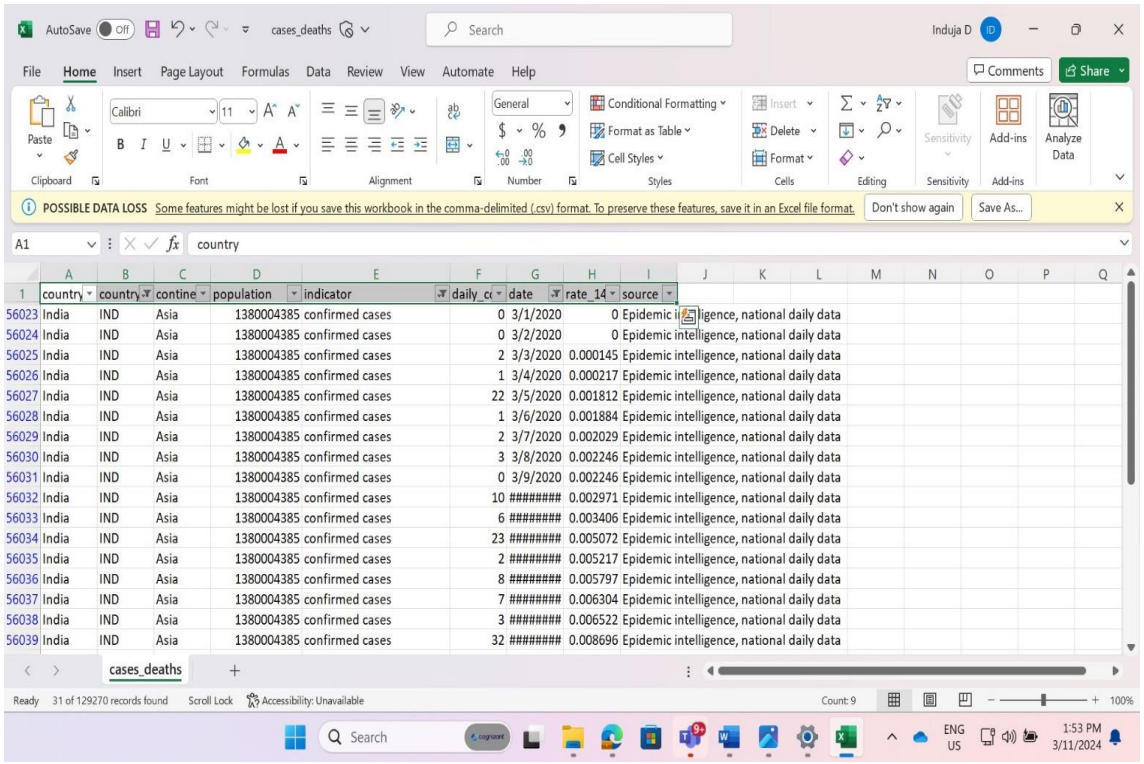
Step 17: After setting whole pipeline by using Lookup and Foreach activity recheck all parameters given in each configuration setting, check the dedicated pool is turn on and then turn on the debug option in pipeline. Finally, all the activities are successfully completed.



Step 18: After successfully run of pipeline now we need check the data inserted into tables in data warehouse by preforming two SQL queries operation given in project documentation.

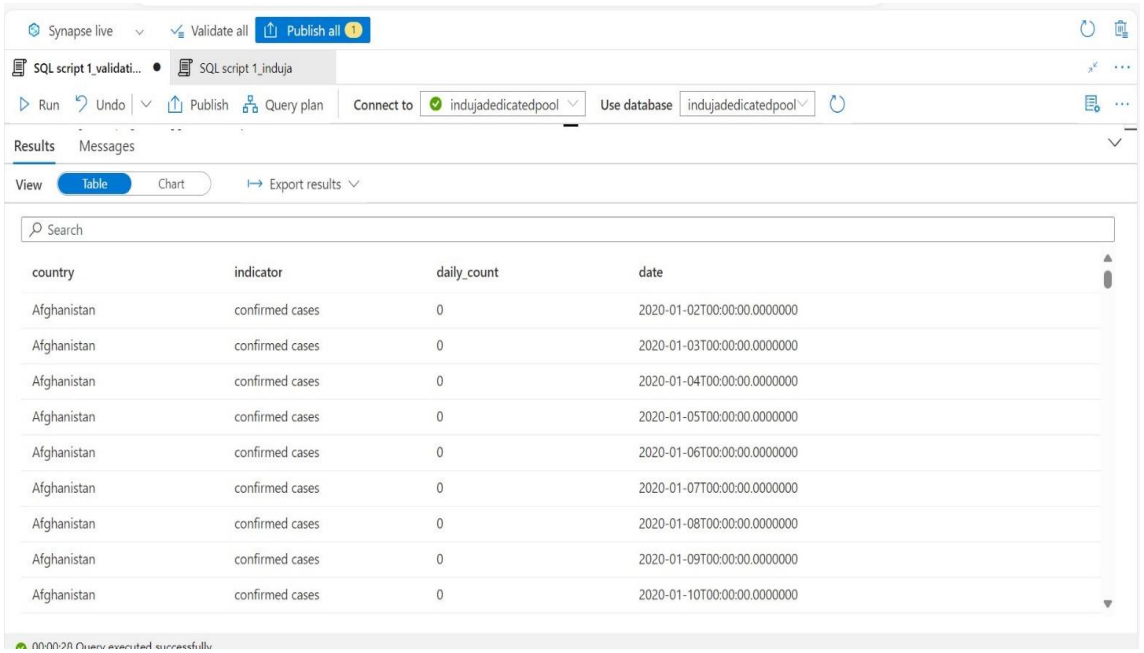


Step 19: After running first and second SQL Query written in SQL script and it is successfully completed and given output as per the query.



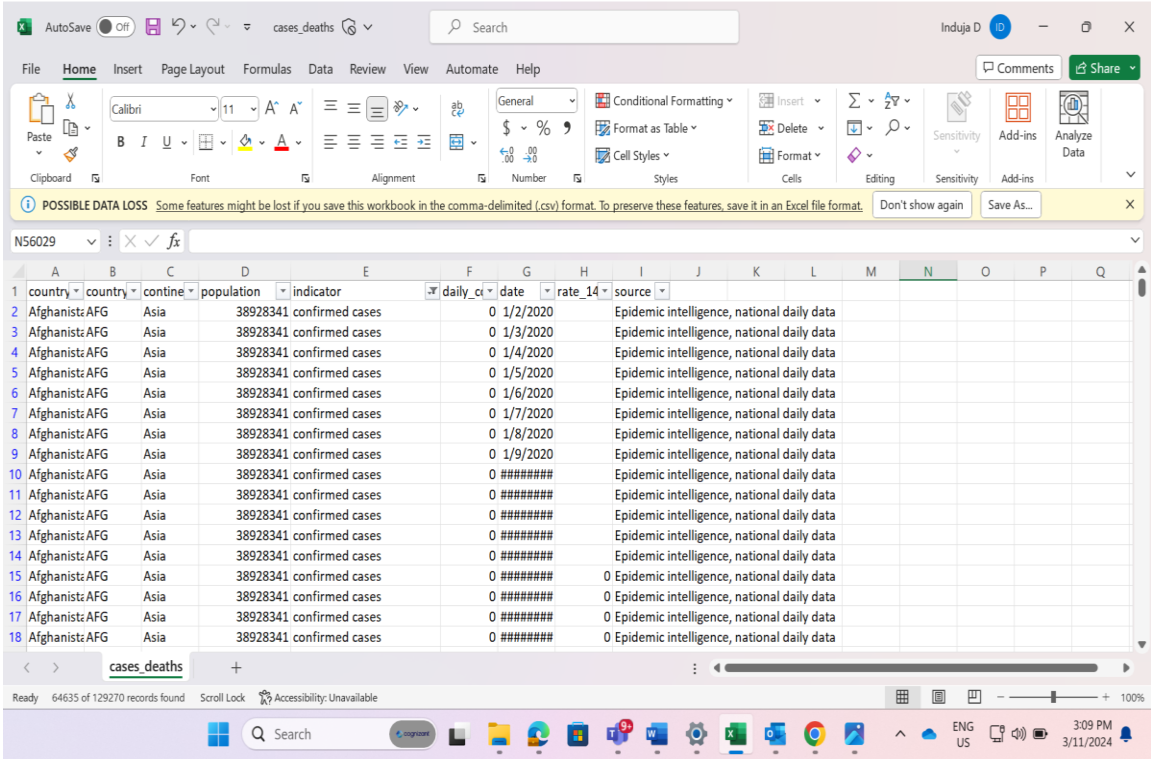
	country	country	contine	population	indicator	daily_cc	date	rate_14	source
56023	India	IND	Asia	1380004385	confirmed cases	0	3/1/2020	0	Epidemic intelligence, national daily data
56024	India	IND	Asia	1380004385	confirmed cases	0	3/2/2020	0	Epidemic intelligence, national daily data
56025	India	IND	Asia	1380004385	confirmed cases	2	3/3/2020	0.000145	Epidemic intelligence, national daily data
56026	India	IND	Asia	1380004385	confirmed cases	1	3/4/2020	0.000217	Epidemic intelligence, national daily data
56027	India	IND	Asia	1380004385	confirmed cases	22	3/5/2020	0.001812	Epidemic intelligence, national daily data
56028	India	IND	Asia	1380004385	confirmed cases	1	3/6/2020	0.001884	Epidemic intelligence, national daily data
56029	India	IND	Asia	1380004385	confirmed cases	2	3/7/2020	0.002029	Epidemic intelligence, national daily data
56030	India	IND	Asia	1380004385	confirmed cases	3	3/8/2020	0.002246	Epidemic intelligence, national daily data
56031	India	IND	Asia	1380004385	confirmed cases	0	3/9/2020	0.002246	Epidemic intelligence, national daily data
56032	India	IND	Asia	1380004385	confirmed cases	10	########	0.002971	Epidemic intelligence, national daily data
56033	India	IND	Asia	1380004385	confirmed cases	6	########	0.003406	Epidemic intelligence, national daily data
56034	India	IND	Asia	1380004385	confirmed cases	23	########	0.005072	Epidemic intelligence, national daily data
56035	India	IND	Asia	1380004385	confirmed cases	2	########	0.005217	Epidemic intelligence, national daily data
56036	India	IND	Asia	1380004385	confirmed cases	8	########	0.005797	Epidemic intelligence, national daily data
56037	India	IND	Asia	1380004385	confirmed cases	7	########	0.006304	Epidemic intelligence, national daily data
56038	India	IND	Asia	1380004385	confirmed cases	3	########	0.006522	Epidemic intelligence, national daily data
56039	India	IND	Asia	1380004385	confirmed cases	32	########	0.008696	Epidemic intelligence, national daily data

Step 20: compare the output appear for previous SQL query written in SQL script with original CSV file in excelsheet.

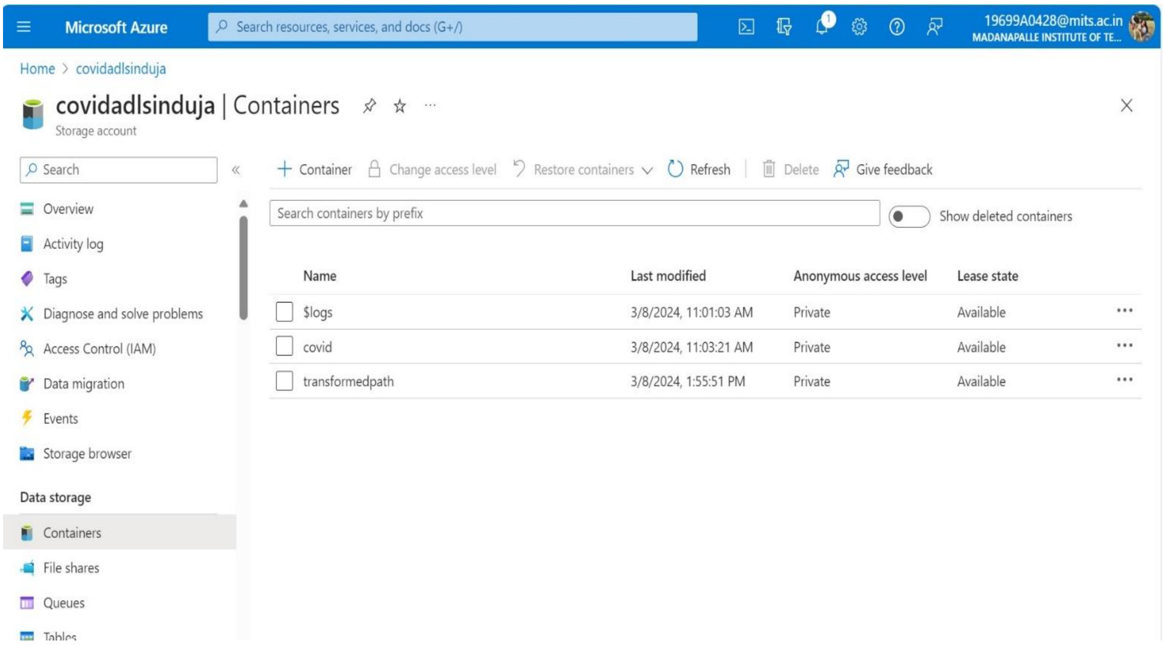


country	indicator	daily_count	date
Afghanistan	confirmed cases	0	2020-01-02T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-03T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-04T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-05T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-06T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-07T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-08T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-09T00:00:00.0000000
Afghanistan	confirmed cases	0	2020-01-10T00:00:00.0000000

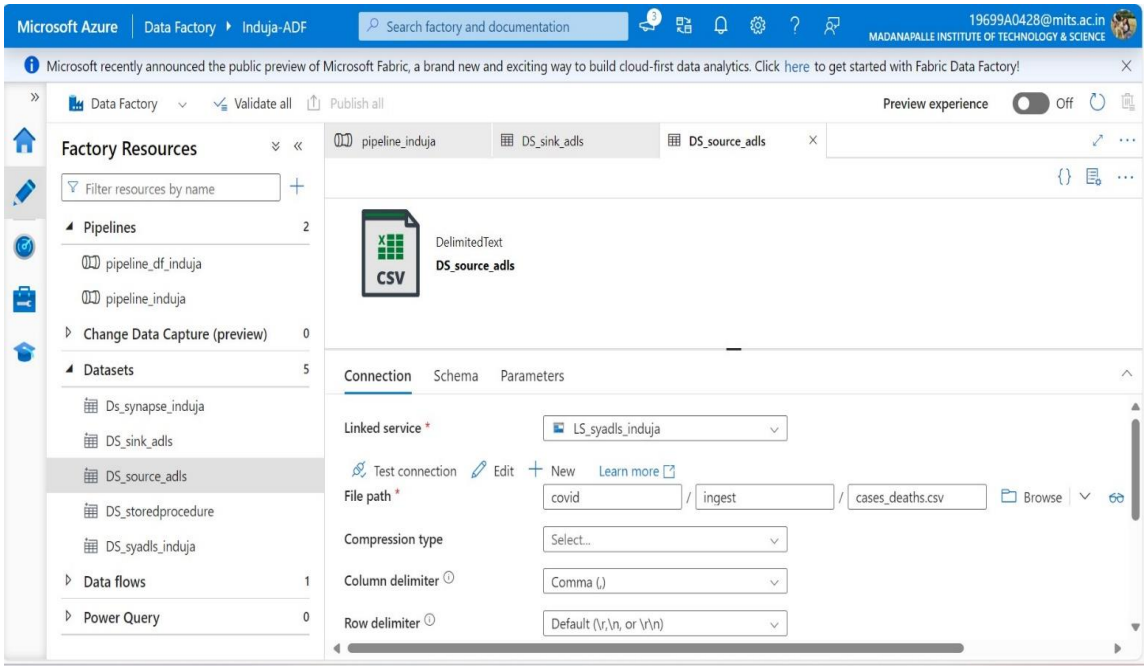
Step 21: Successfully query is done and given output as per the query.

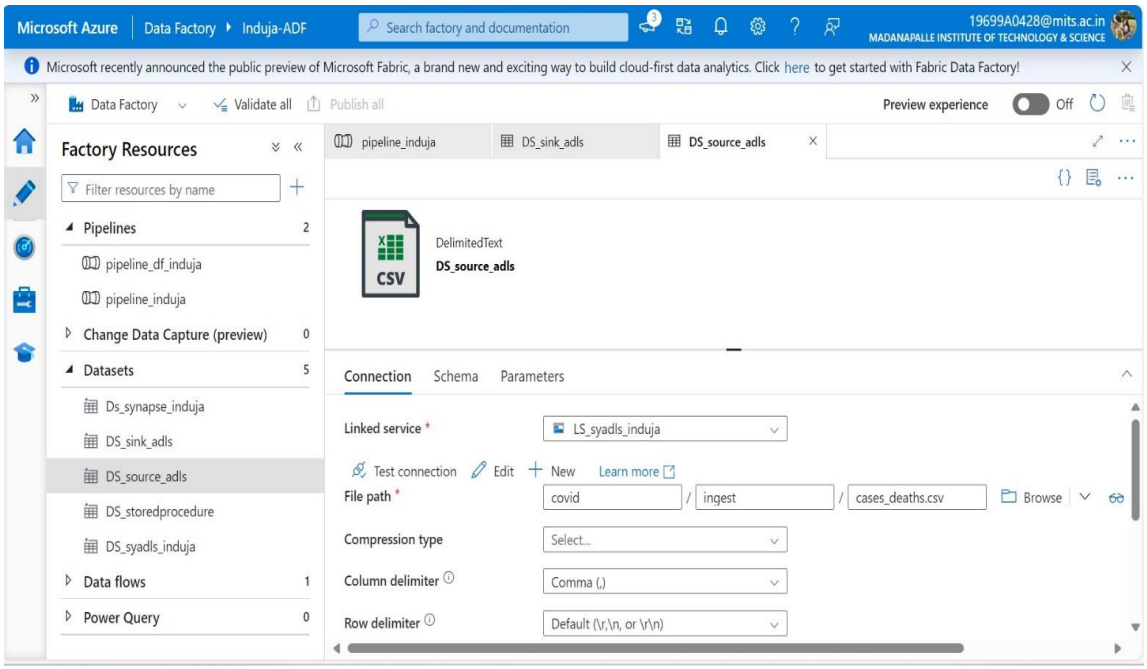


Step 22: Compare the above SQL query output with data which is appear from CSV file which is open in excel for reference check.

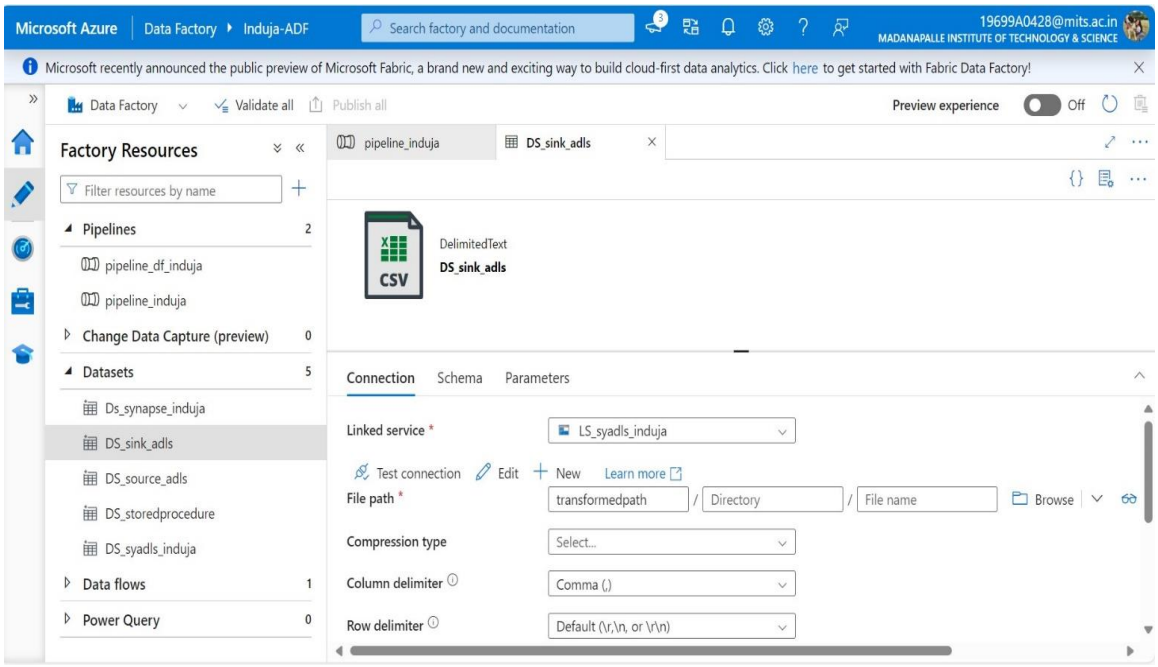


Step 23: Create another container with name **“transformedpath”** for second requirement given in project for storing transformed data file by using data flow.

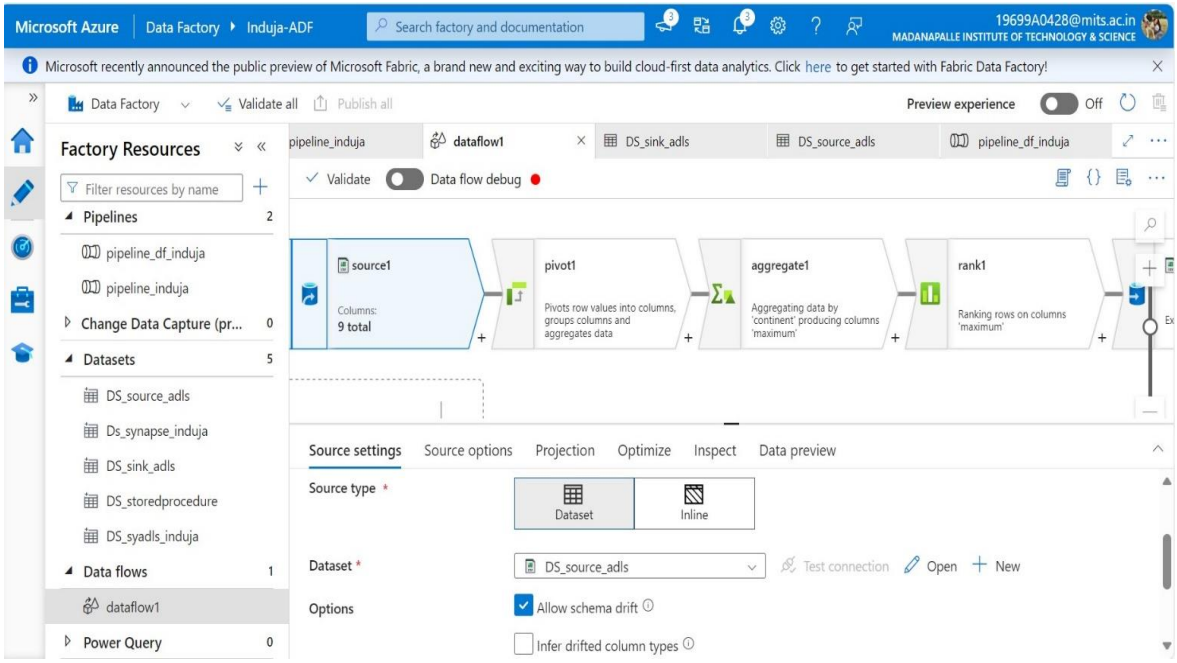




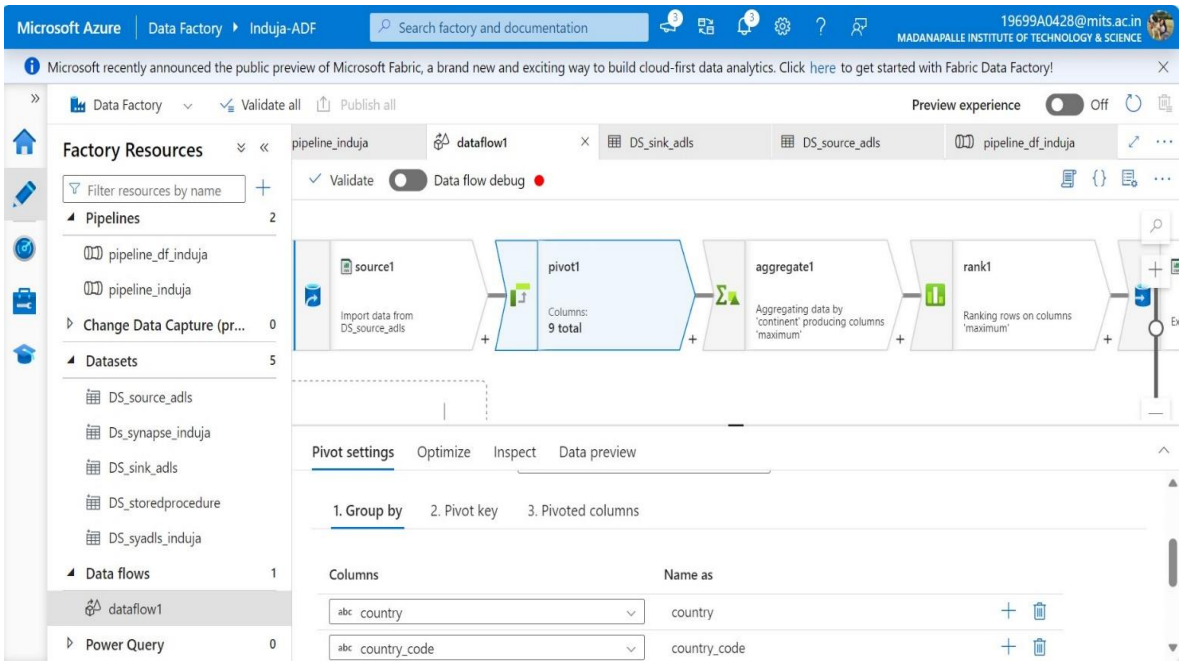
Step 24: Create source dataset (**DS_source_adls**) for dataflow by giving a file specific filename on which data transformation need to be taken place as per project requirement.



Step 25: Create target dataset (**DS_sink_adls**) for dataflow to keep that data transformpath file in specific place for further use.

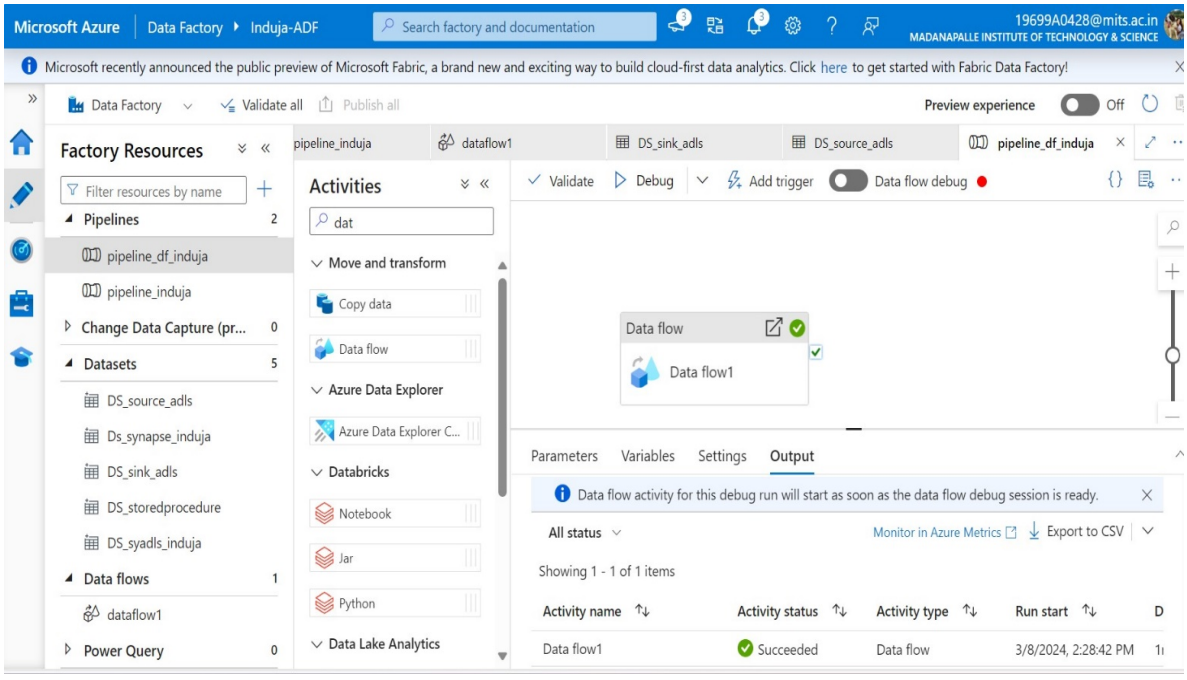


Step 26: Develop dataflow by using some transformations like source, filter, pivot, aggregate, rank and sink as per the question given in project documentation.

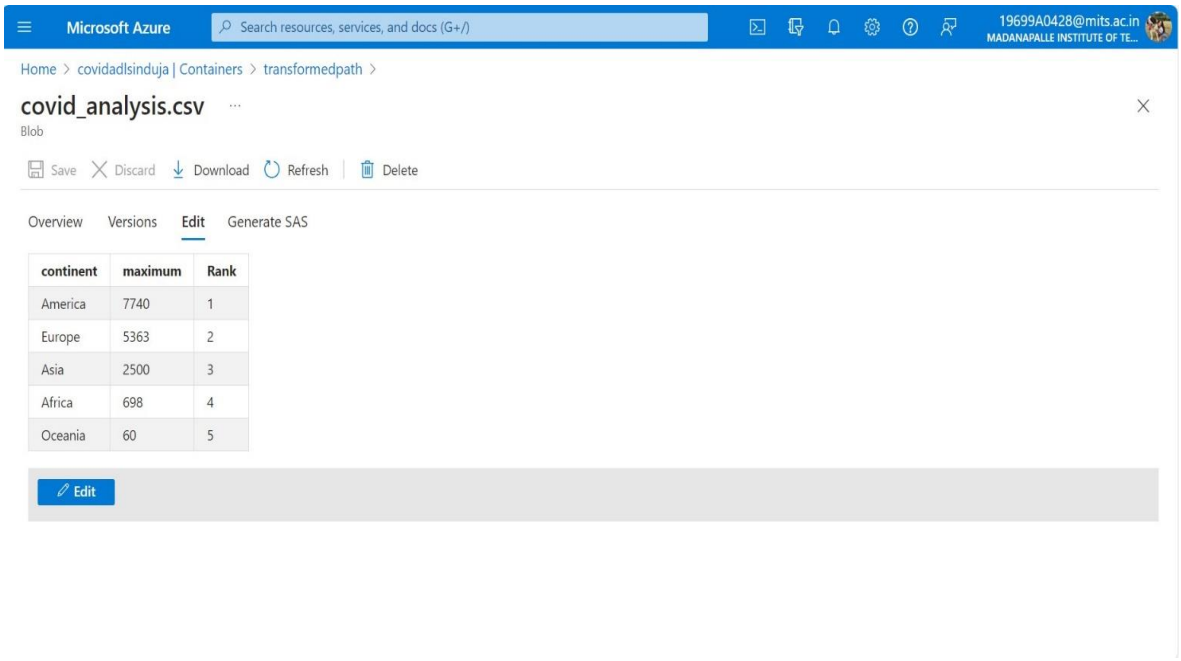


Step 27: After successfully creating dataflow, I click on dataflow debug output appear as per the question.

Step 28: Drag and drop the data flow into the pipeline workspace and connect with before activity to run after successfully complication of before activity.



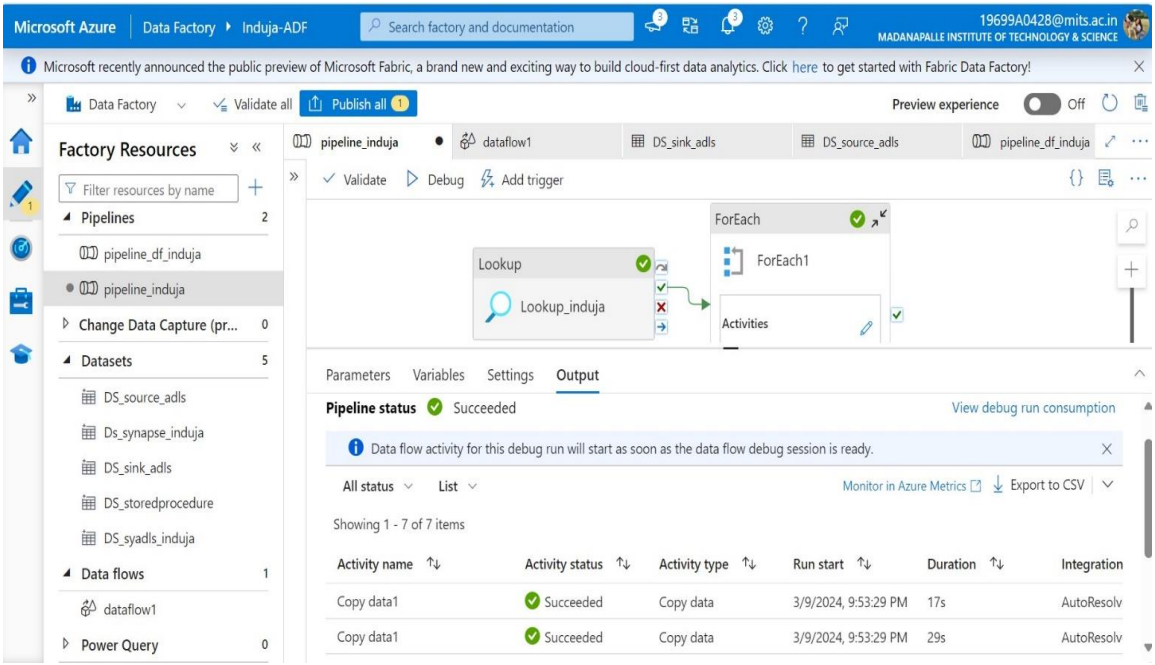
Step 29: Dataflow is completed successfully after click on debug as how in above picture

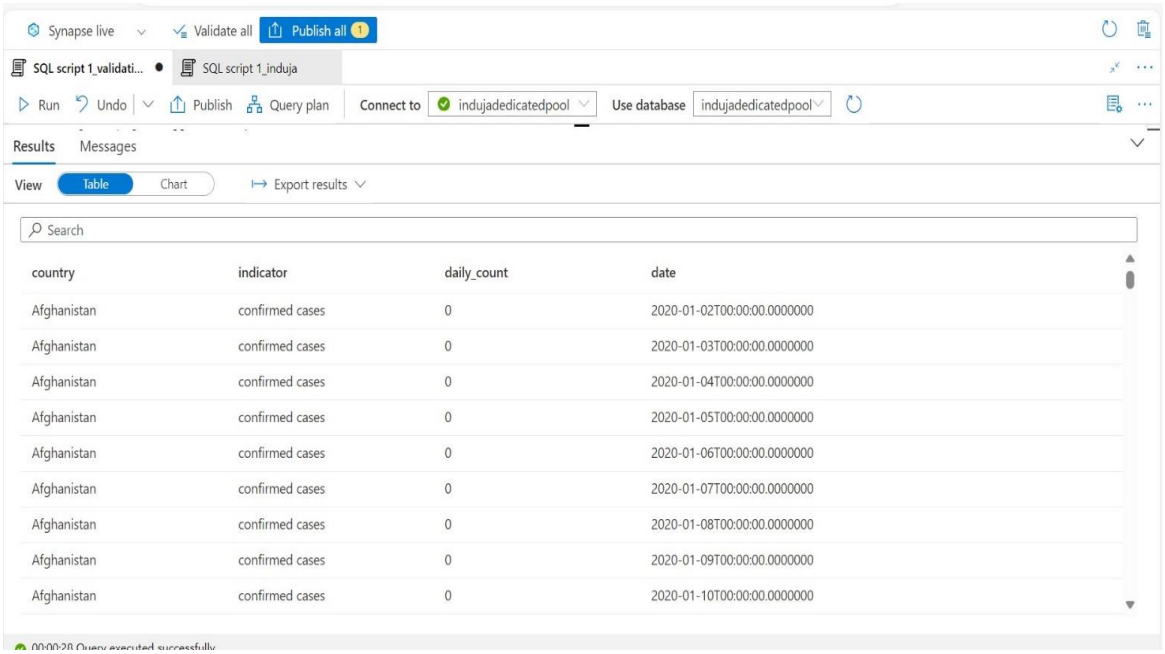


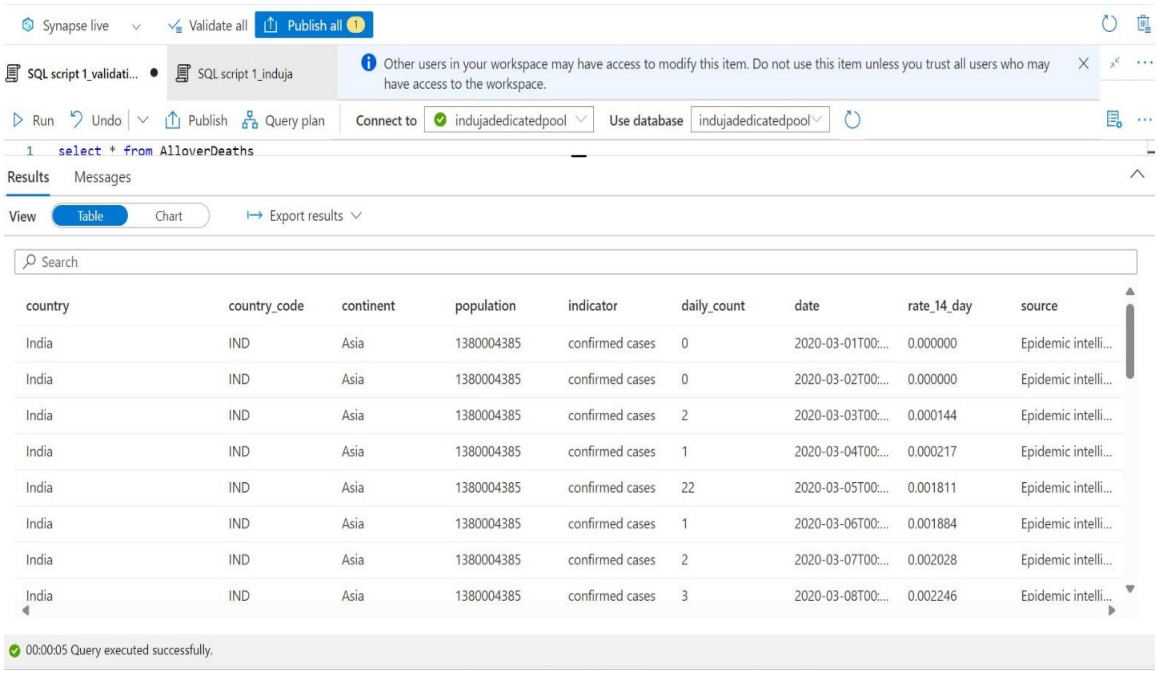
Step 30: After that we need to check file is appear in the transformation container in storage account (data lake). I successfully got that file in my container as per the question given in the project documentation.

Conclusion:

Requirement 1 Output:







Requirement 2 Output: