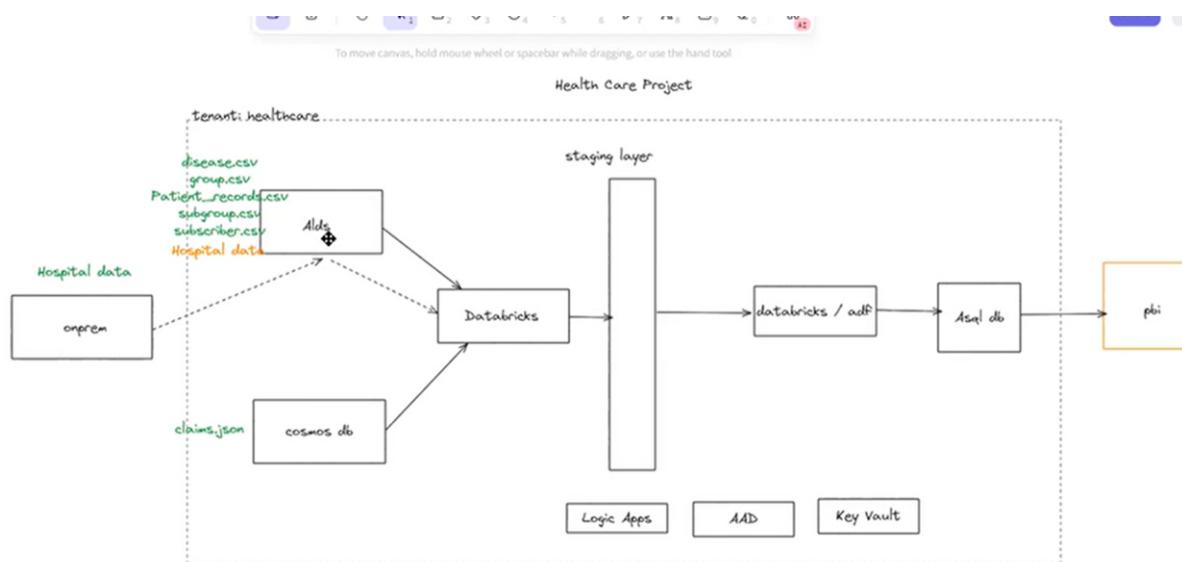


## HEALTH CARE PROJECT USING AZURE DATA BRICKS

### NEED SERVICES

1. ADLS STORAGE
2. ADF
3. DATABRICKS
4. AZURE SQL SERVER
5. COSMOS DB
6. ON MYSQL WITH INTEGRATION RUN TIME
7. KEY VAULT AND IF NEEDED LOGIC APPS ALSO

WE NEED TO CREATE THESE SERVICES SO ALREADY WE KNOW THE HOW TO CREATE SOME SERVICES BUT NOW I AM CREATING DATA BRICKS SERVICES.



This screenshot shows the Azure portal interface for the 'optum-data-pipeline' resource group. The left sidebar includes navigation links for Home, Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Deployments, Security, Deployment stacks, Policies, Properties, Locks, Cost analysis, and Give feedback. The main content area displays the 'Essentials' section with a table of resources. The table has columns for Name, Type, and Location. The resources listed are:

Name	Type	Location
ksr_sql (optumserver/ksr_sql)	SQL database	East US
optumdatafactory	Data factory (V2)	East US
optumdataworkspace	Azure Databricks Service	West US 3
optumkeyvault	Key vault	East US
optumnosql	Azure Cosmos DB for MongoDB account (RU)	West US 3
optumsads	Storage account	West US 3
optumserver	SQL server	East US

Create an Azure Databricks workspace | Cost Management: MANIDEEP K. | KSR DATA VIZON PRIVATE LIMITED | +

https://portal.azure.com/#create/Microsoft.Databricks

Microsoft Azure Search resources, services, and docs (G+/-) Copilot Home > Azure Databricks Create an Azure Databricks workspace ...

Basics Networking Encryption Security & compliance Tags Review + create

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 \*

Instance Details

Workspace name \*  Region \*  Pricing Tier \*  Managed Resource Group name

Review + create < Previous Next : Networking >

8 USD/INR +0.47% 17:00:15 07-07-2025

Create an Azure Databricks workspace | Cost Management: MANIDEEP K. | KSR DATA VIZON PRIVATE LIMITED | +

https://portal.azure.com/#create/Microsoft.Databricks

Microsoft Azure Search resources, services, and docs (G+/-) Copilot Home > Azure Databricks Create an Azure Databricks workspace ...

Basics **Networking** Encryption Security & compliance Tags Review + create

Deploy Azure Databricks workspace with Secure Cluster Connectivity (No Public IP)  Yes  No

Deploy Azure Databricks workspace in your own Virtual Network (VNet)  Yes  No

Review + create < Previous Next : Encryption >

8 USD/INR +0.47% 17:00:28 07-07-2025

Create an Azure Databricks workspace | Cost Management: MANIDEEP K. | KSR DATA VIZON PRIVATE LIMITED | +

https://portal.azure.com/#create/Microsoft.Databricks

Microsoft Azure Search resources, services, and docs (G+)

Copilot

manideep4942@gmail.com DEFAULT DIRECTORY

Home > Azure Databricks >

## Create an Azure Databricks workspace

Encryption

Basics Networking Encryption Security & compliance Tags Review + create

### Data Encryption

For additional control of your data, you can add your own key to protect and control access to some types of data. Enabling customer-managed key encryption for Managed Services or Managed Disks is an irreversible action. The key, key vault, and key version may be updated but the features cannot be disabled after being enabled.

#### Managed Disks

Use your own key  The current pricing tier does not support customer-managed key encryption.

#### Managed Services

Use your own key  The current pricing tier does not support customer-managed key encryption.

#### Double encryption for DBFS root

In addition to your choice of the default encryption or your own managed key encryption, Azure Databricks DBFS root can also be encrypted with a second layer of encryption called infrastructure encryption using platform-managed key to achieve Double Encryption for DBFS root.

Enable Infrastructure Encryption

Review + create < Previous Next : Security & compliance >

USD/INR +0.47% 17:00:42 07-07-2025

Create an Azure Databricks workspace | Cost Management: MANIDEEP K. | KSR DATA VIZON PRIVATE LIMITED | +

https://portal.azure.com/#create/Microsoft.Databricks

Microsoft Azure Search resources, services, and docs (G+)

Copilot

manideep4942@gmail.com DEFAULT DIRECTORY

Home > Azure Databricks >

## Create an Azure Databricks workspace

Security & compliance

Basics Networking Encryption Security & compliance Tags Review + create

### Enhanced Security & Compliance

Enhanced Security and Compliance Add-On helps simplify the complexity of meeting security and regulatory requirements.

Enable compliance security profile  The current pricing tier does not support the Enhanced Security and Compliance add-on.

Enable enhanced security monitoring  The current pricing tier does not support the Enhanced Security and Compliance add-on.

Enable automatic cluster update  The current pricing tier does not support the Enhanced Security and Compliance add-on.

Review + create < Previous Next : Tags >

USD/INR +0.47% 17:00:55 07-07-2025

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#create/Microsoft.Databricks>. The page is titled "Create an Azure Databricks workspace". The "Tags" tab is selected. A table is present with columns "Name" and "Value". There are two rows in the table, both labeled "2 selected". At the bottom, there are buttons for "Review + create", "< Previous", and "Next : Review + create >". The status bar at the bottom shows "USD/INR +0.47%", system icons, and the date "07-07-2025".

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#create/Microsoft.Databricks>. The page is titled "Create an Azure Databricks workspace". The "Review + create" tab is selected. The page displays validation messages: "Validating..." and "Deploying...". It includes sections for "Summary", "Networking", and "Encryption". Under "Networking", it shows settings for "Deploy Azure Databricks workspace with Secure Cluster Connectivity (No Public IP)" (Yes) and "Deploy Azure Databricks workspace in your own Virtual Network (VNet)" (No). Under "Encryption", it shows "Enable Infrastructure Encryption" (No). At the bottom, there are buttons for "Create" and "< Previous". The status bar at the bottom shows "USD/INR +0.47%", system icons, and the date "07-07-2025".

Here it is now we created

The screenshot shows the Microsoft Azure portal interface. At the top, there are three tabs: 'optumworkspace - Microsoft Azure', 'Create Cluster - Databricks', and 'Resource groups - Microsoft Azure'. The main content area is titled 'optumworkspace' and shows the 'Overview' tab selected. The 'Essentials' section displays the following information:

		Value
Status	:	Active
Resource group	:	optum-rg
Location	:	South India
Subscription	:	Azure for Students
Subscription ID	:	fc99c321-a8e4-4c30-8ab9-ab6da2ed958b
Tags	:	Add tags

Below this, there is a large red icon of three stacked cubes. A blue button labeled 'Launch Workspace' is positioned below the icon. To the right of the 'Essentials' section, a 'JSON View' link is visible. At the bottom of the page, there is a toolbar with various icons and a status bar showing the date and time.

Now create a compute engine to run the queries .

Select the personal computer

The screenshot shows the Databricks Compute blade. The left sidebar includes options like 'Compute', 'Marketplace', 'SQL', 'Data Engineering', 'Job Runs', 'Data Ingestion', 'AI/ML', 'Playground', 'Experiments', 'Features', 'Models', and 'Serving'. The main area is titled 'Compute > New compute > Simple form: OFF'. A dropdown menu for 'Policy' is open, showing 'Personal Compute' selected. The 'Summary' section on the right provides details about the cluster configuration:

Setting	Value
1 Driver	14 GB Memory, 4 Cores
Runtime	17.0.x-cpu-ml-scala2.13
Unity Catalog	Standard_DS3_v2
0.75 DBU/h	

At the bottom, there are 'Create compute' and 'Cancel' buttons.

And create the cluster for the notebook

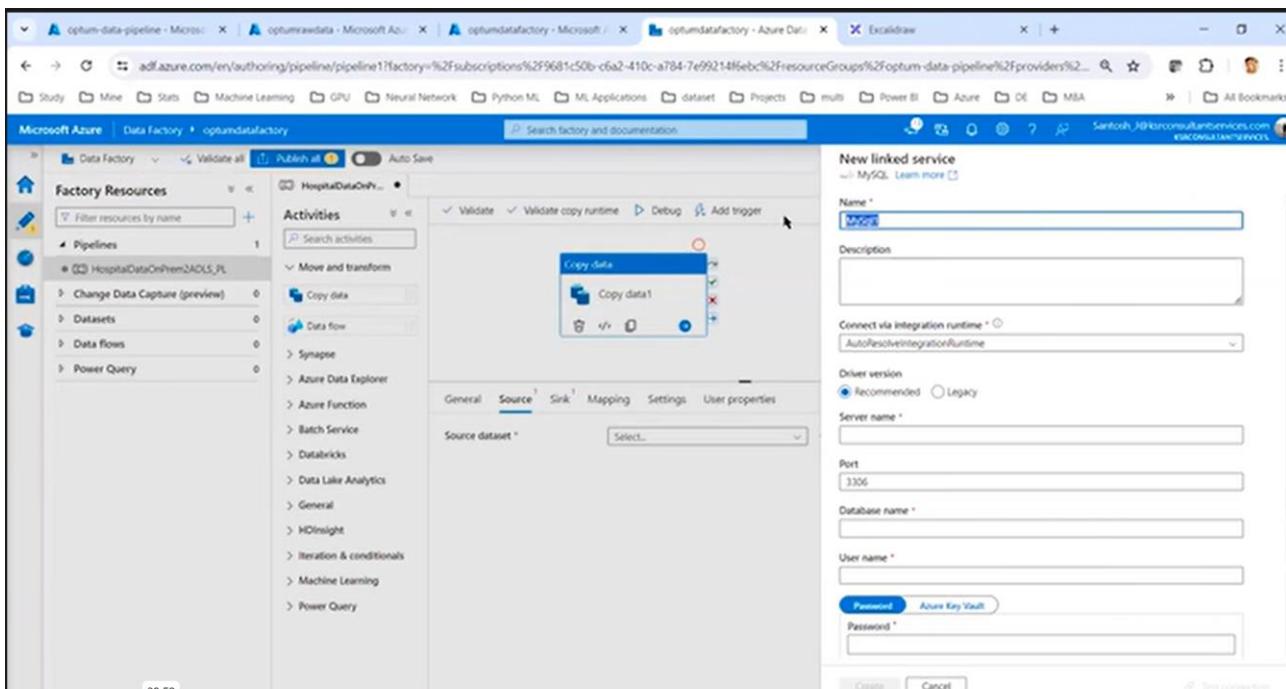
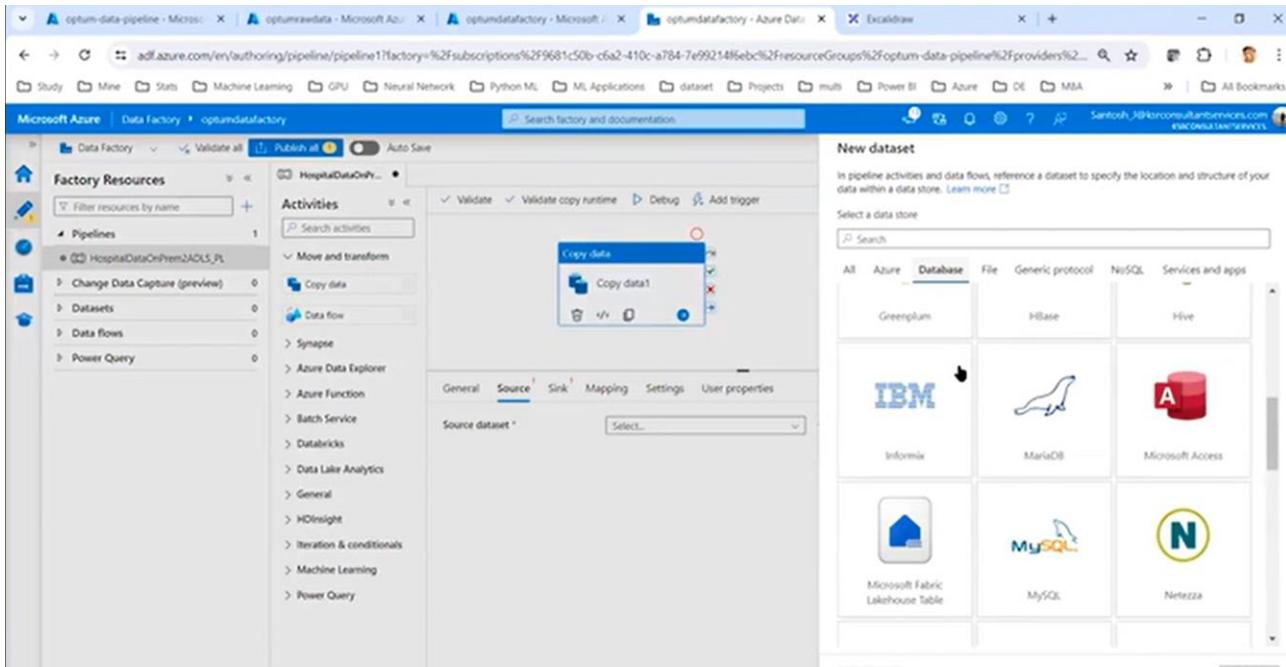
And not create the adls storage and upload these files into this

The screenshot shows the Microsoft Azure Storage Explorer interface. The left sidebar displays the 'Overview' tab for the 'optumrawdata' container. The main area lists five CSV files: 'Patient\_records.csv', 'disease.csv', 'group.csv', 'subgroup.csv', and 'subscriber.csv'. Each file entry includes columns for Name, Last modified, Access tier, Blob type, Size, and Lease state. The 'Access tier' column shows 'Cool (Inferred)' for all files. The 'Size' column shows sizes ranging from 4.99 KiB to 11.78 KiB. The 'Lease state' column shows 'Available' for all files. The top navigation bar shows multiple tabs including 'optumworkspace - Microsoft Azure', 'optumrawdata - Microsoft Azure', 'hospital data dev - Databricks', and 'Cluster Details - Databricks'. The address bar indicates the URL is https://portal.azure.com/#view/Microsoft\_Azure\_Storage/ContainerMenuBlade/-/overview/storageAccountName%2Fsubscriptions%2Fcfc99c321-a8e4-4c30-8ab9-ab6da2ed95... . The bottom taskbar shows various icons and system status.

Name	Last modified	Access tier	Blob type	Size	Lease state
Patient_records.csv	7/7/2025, 5:47:33 pm	Cool (Inferred)	Block blob	4.99 KiB	Available
disease.csv	7/7/2025, 5:47:33 pm	Cool (Inferred)	Block blob	1.45 KiB	Available
group.csv	7/7/2025, 5:47:33 pm	Cool (Inferred)	Block blob	4.29 KiB	Available
subgroup.csv	7/7/2025, 5:47:33 pm	Cool (Inferred)	Block blob	561 B	Available
subscriber.csv	7/7/2025, 5:47:33 pm	Cool (Inferred)	Block blob	11.78 KiB	Available

IN THIS WE USE THE DATA BRICKS BECAUSE FOR COPY ACTIVITY , WE CAN ALSO DO FROM DATABRICKS ALSO BUT IT WILL CHARGE LIKE ANY THING .

## THE FIRST PIPELINE WE NEED TO CREATE (ON-PREM TO ADLS)(HOSPITAL DATASET)

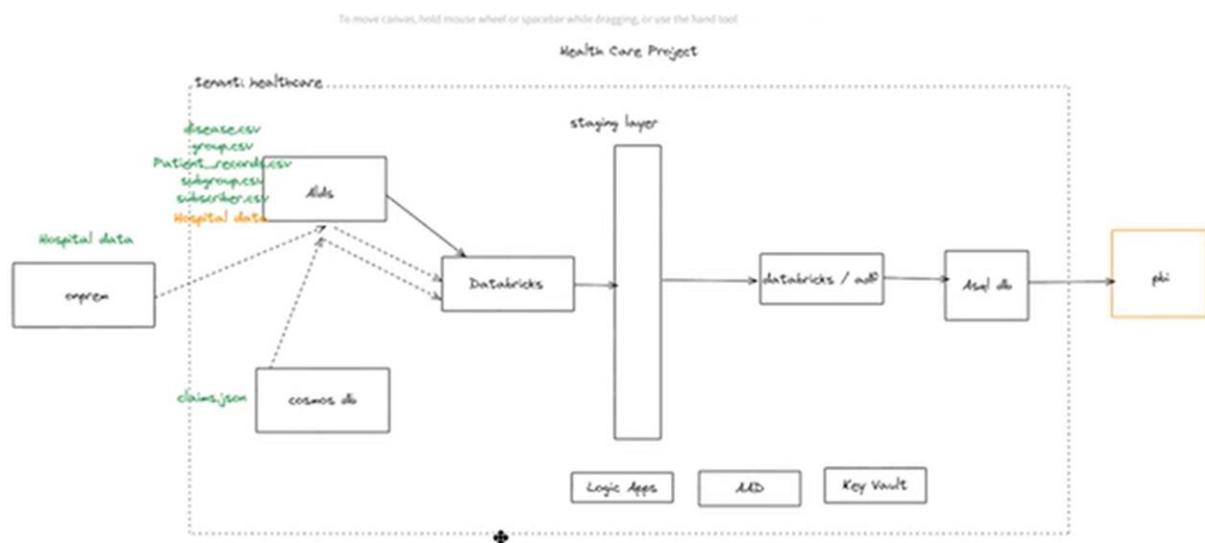


## CREATE THE INPUT AND OUTPUT DATA SET TO GET DATA FROM ONPREM TO BLOB.

## CREATE ONE MAIN PIPELINE

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' pane lists 'Pipelines' (including 'HospitalDataOnPrem2ADLS\_PL' and 'Optum\_PL'), 'Datasets' (including 'OnPremHospitalData' and 'RawHospitalData'), and other resources like 'Data flows' and 'Power Query'. In the center, an 'Execute Pipeline' activity is selected. The 'Properties' pane on the right shows the pipeline name 'Optum\_PL' and a description. The 'Settings' tab is active, with the 'Invoked pipeline' dropdown set to 'HospitalDataOnPrem2ADLS\_PL' and the 'Wait on completion' checkbox checked.

## SIMILAR JSON ALSO I LL NEED TO BRING TO THE ADLS



## CREATE ANOTHER PIPELINE FOR JSON(CLIMS DATA)

The screenshot shows the Microsoft Azure Data Factory interface. The 'Factory Resources' pane lists 'Pipelines' (including 'HospitalDataOnPrem2ADLS\_PL', 'Optum\_PL', and 'ClaimDataCosmos2ADLS\_PL'), 'Datasets' (including 'OnPremHospitalData' and 'RawHospitalData'), and other resources. In the center, a 'Copy data' activity is selected. The 'Properties' pane on the right shows the pipeline name 'ClaimDataCosmos2ADLS\_PL'. The 'Source' tab is active in the 'Copy data1' activity settings. A 'New dataset' dialog is open on the right, listing various data store options under the 'Azure' category, such as Azure Blob Storage, Azure Cosmos DB for MongoDB, Azure Cosmos DB for NoSQL, Azure Data Explorer (Kusto), Azure Data Lake Storage Gen2, Azure Database for MariaDB, and Azure MySQL.

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 'Copy data' activity is selected. The 'Set properties' panel on the right shows the following configuration:

- Name:** RawClaimData
- Linked service:** adl2adls\_ls
- File path:** File system / Directory / File name
- First row as header:** checked
- Import schema:** None

## SINK THE DATA IN CSV FORMAT.

The screenshot shows the Microsoft Azure Data Factory pipeline editor. The 'Activities' pane on the left lists various options like 'Execute Pipeline', 'Append variable', 'Delete', 'Execute Pipeline', etc. Two parallel 'Execute Pipeline' activities are present in the workspace, each pointing to a different pipeline: 'Execute Pipeline1' and 'Execute Pipeline2'. The 'Properties' panel on the right shows the following for the first pipeline:

- Name:** Optum\_Pl
- Description:** (empty)

AGAIN ADD TO THE ONPREM\_PL TO EXECUTE PARALLEL ACTIVITIES

NOW GO AND TRIGGER IT

Home > optumadsis | Containers >

**optumrawdata** Container

Search Overview Diagnose and solve problems Access Control (IAM)

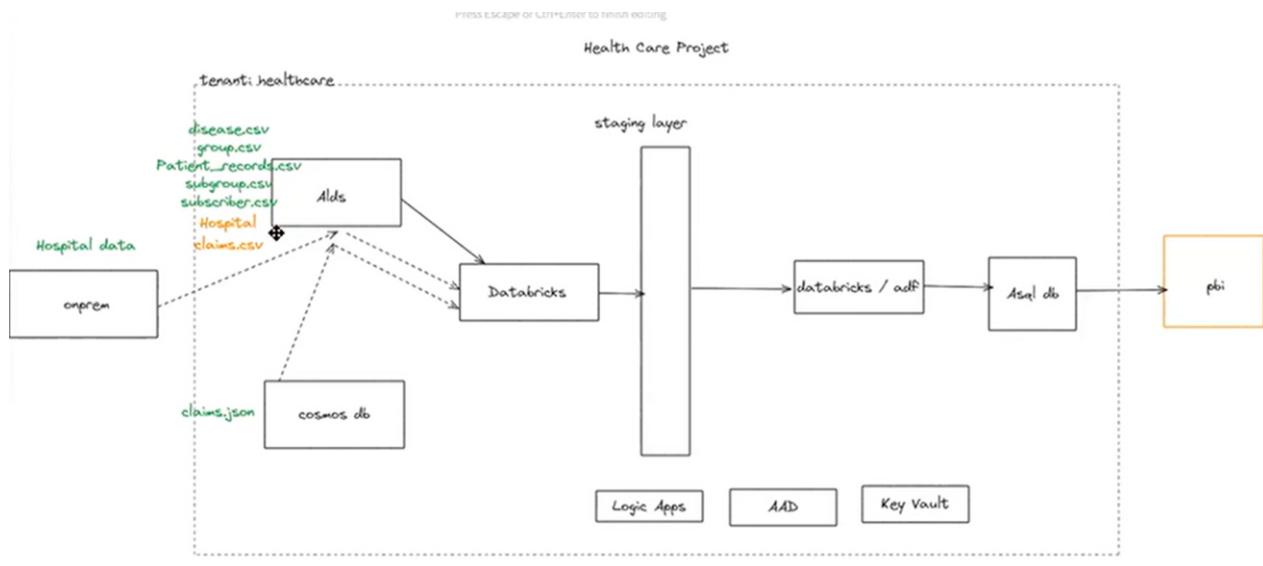
Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: optumrawdata

Search blobs by prefix (case-sensitive) Show deleted objects

Settings Shared access tokens Manage ACL Access policy Properties Metadata

Name	Modified	Access tier	Archive status	Block type	Size	Lease state
claims.csv	5/5/2024, 8:50:43 AM	Hot (Inferred)		Block blob	5.63 KB	Available
disease.csv	5/4/2024, 9:51:48 AM	Hot (Inferred)		Block blob	1.45 KB	Available
group.csv	5/4/2024, 9:51:48 AM	Hot (Inferred)		Block blob	4.29 KB	Available
hospital.csv	5/5/2024, 8:58:47 AM	Hot (Inferred)		Block blob	1.49 KB	Available
Patient_records.csv	5/4/2024, 9:51:48 AM	Hot (Inferred)		Block blob	4.99 KB	Available
subgroup.csv	5/4/2024, 9:51:48 AM	Hot (Inferred)		Block blob	561 B	Available
subscriber.csv	5/4/2024, 9:51:48 AM	Hot (Inferred)		Block blob	11.78 KB	Available

## NOW ALL THE DATA HAS BEEN COME INTO ADLS



NOW LETS GO THE DATABRICKS →

New

Santosh J's Cluster

Compute > New compute >

Policy: Unrestricted

Access mode: Single user access

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

Summary

1 Driver 16 GB Memory, 4 Cores

Runtime 13.3.x-scala2.12

Unity Catalogs Photon Standard D4ds v3 2 DBUs/h

## CREATE CLUSTER

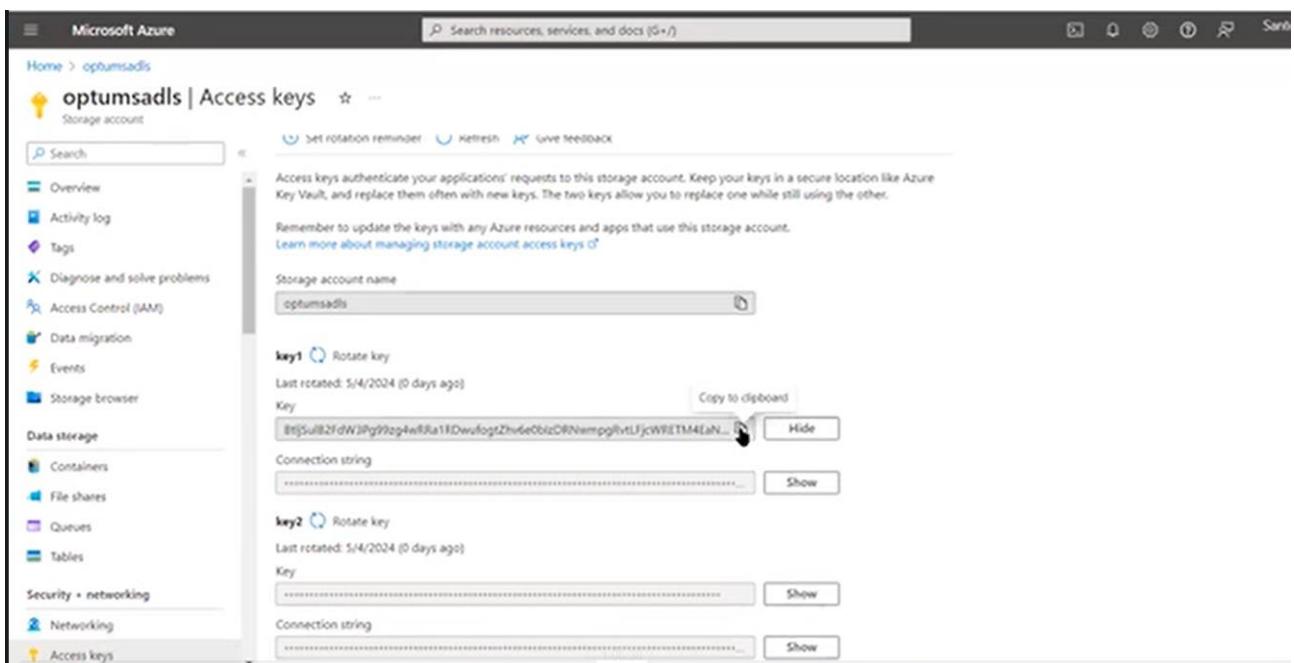
File Edit View Run Help Last edit was 1 minute ago New cell UI ON Run all Santosh J's Cluster Schedule Share

subgroup\_transformation Python

```
Just now(+)
Import pyspark
from pyspark.sql import *
```

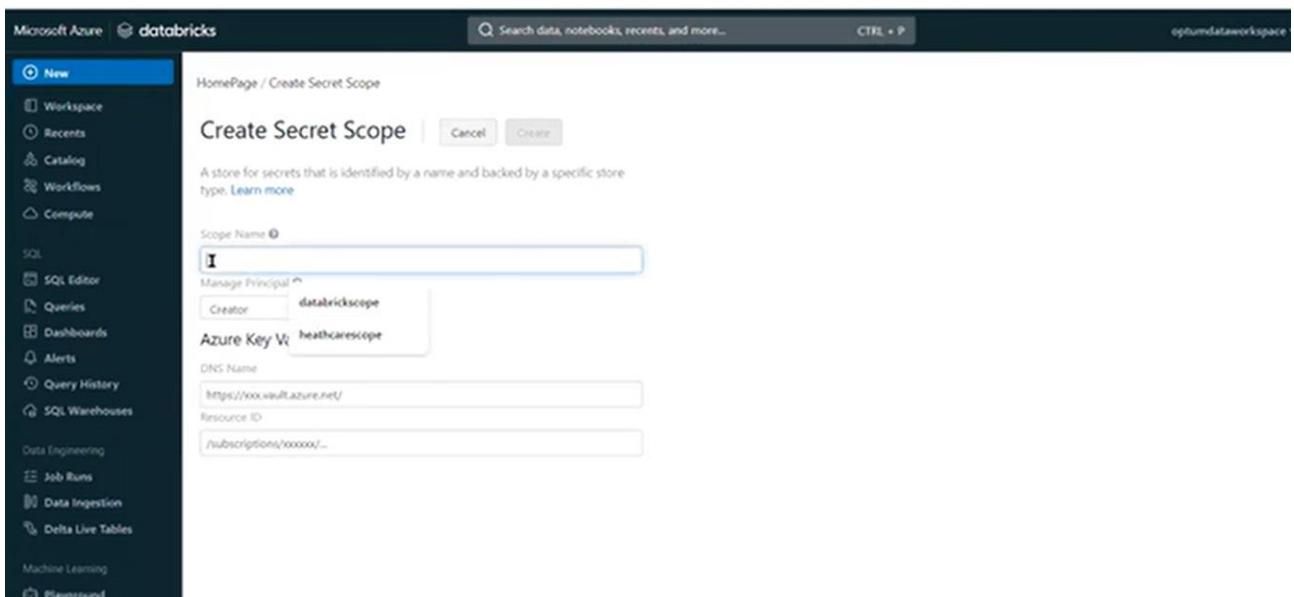
Start typing or generate with AI (Ctrl + I)...

NEXT WE NEED TO DO CREATE A KEY VALUATE FOR DB



The screenshot shows the Microsoft Azure Storage account access keys page for the storage account 'optumsadls'. The left sidebar lists various management options like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Data storage (Containers, File shares, Queues, Tables), Security + networking (Networking, Access keys), and a general Home link. The main content area displays two sets of access keys: 'key1' and 'key2'. Each key includes a 'Rotate key' button, a timestamp (Last rotated: 5/4/2024 (0 days ago)), a 'Key' field containing a long hex string, a 'Copy to clipboard' button, and a 'Hide' button. Below each key is a 'Connection string' section with a 'Show' button.

## NOW CREATE THE SCOPE IN DATA BRICKS



The screenshot shows the Databricks interface with the 'databricks' workspace selected. The left sidebar contains navigation links for New, Workspace, Recents, Catalog, Workflows, Compute, SQL (SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses), Data Engineering (Job Runs, Data Ingestion, Delta Live Tables), Machine Learning, and Playground. The main content area is titled 'Create Secret Scope' and includes fields for 'Scope Name' (set to 'databrickscope'), 'Manage Principal' (set to 'Creator'), 'Azure Key V' (set to 'heathcarescope'), 'DNS Name' (set to 'https://xxx.vault.azure.net/'), and 'Resource ID' (set to '/subscriptions/xxxxxx/...'). There are 'Cancel' and 'Create' buttons at the bottom.

Microsoft Azure

Home > optum-data-pipeline > optumkeyvault

## optumkeyvault | Properties

Key vault

Search Save Discard changes Refresh

**Properties**

Name: optumkeyvault  
 Sku (Pricing tier): Standard  
 Location: eastus  
 Vault URL: https://optumkeyvault.vault.azure.net/  
 Resource ID: /subscriptions/9681c50b-cfa2-410c-a784-7e99214f6ebc/resourceGroups/optum-data-pipeline/providers/Microsoft.KeyVault/vaults/optumkeyvault  
 Subscription ID: 9681c50b-cfa2-410c-a784-7e99214f6ebc  
 Subscription Name: KSR DataVizion Paid Subscription  
 Directory ID: 88944d0-1278-45f9-88d8-dc81b5f777b1  
 Directory Name: karconsultantservices.  
**Soft-delete:** Soft delete has been enabled on this key vault  
 Days to retain deleted vaults: 90  
 Disable purge protection (allow key vault and objects to be purged during retention period)  
 Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

## NOW LIST SCOPE

subgroup\_transformation Python

File Edit View Run Help Last edit was now New cell Up: ON

Run all Santosh J's Cluster Schedule Share

```
1 ✓ 2 minutes ago (10)
import pyspark
from pyspark.sql import *

2 ✓ Just now (10)
dbutils.secrets.listScopes()
[SecretScope(name="databricksscope")]

3
Start typing <generate with AI (Ctrl + S)...>

[Shift+Enter] to run and move to next cell
[Esc H] to see all keyboard shortcuts
```

Python

```
dbutils.secrets.list(scope = "databricksscope")
[SecretMetadata(key='adlskey')]
```

Just now (2s)

4

```
spark.conf.set("fs.azure.account.key.optumadls.dfs.core.windows.net",dbutils.secrets.get(scope="databricksscope", key="adlskey"))
```

## NOW WE ARE GOING TO USE SUB GROUP DATASET

Just now (8s) 5  
display(dbutils.fs.ls("abfss://optumrawdata@optumsadls.dfs.core.windows.net"))  
(2) Spark Jobs

	path	name	size	modificationTime
1	abfss://optumrawdata@optumsadls.dfs.core.windows.net/Patient_records.csv	Patient_records.csv	5110	1714796508000
2	abfss://optumrawdata@optumsadls.dfs.core.windows.net/claims.csv	claims.csv	5766	1714879723000
3	abfss://optumrawdata@optumsadls.dfs.core.windows.net/disease.csv	disease.csv	1489	1714796508000
4	abfss://optumrawdata@optumsadls.dfs.core.windows.net/group.csv	group.csv	4390	1714796508000
5	abfss://optumrawdata@optumsadls.dfs.core.windows.net/hospital.csv	hospital.csv	1528	1714879727000

Just now (3s) 6  
data = spark.read.csv("abfss://optumrawdata@optumsadls.dfs.core.windows.net/subgroup.csv")  
(1) Spark Jobs  
data: pyspark.sql.dataframe.DataFrame = [c0: string, \_c1: string ... 2 more fields]

Just now (1s) 8  
data.count()  
(2) Spark Jobs  
10

Just now (<1s) 9  
#drop duplicates  
data = data.dropDuplicates()  
data: pyspark.sql.dataframe.DataFrame = [subgrp\_sk: string, subgrp\_name: string ... 2 more fields]

Just now (1s) 10  
data.count()  
(3) Spark Jobs  
10

Interrupt 00:01 10 Python ⚡ ⌂ ⌂ ⌂

```
data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in data.columns]).show()
```

[All](#) [Images](#) [Videos](#) [News](#) [Shopping](#) | More

Tools

About 34,900 results (0.29 seconds)

In PySpark, the explode function is used to transform each element of an array in a DataFrame column into a separate row. However, this function requires the column to be an array. If your data is in string format, you'll need to convert it to an array before using explode . 10 JUL 2023

 Waiting

```
import pyspark
from pyspark.sql import *
from pyspark.sql.functions import *
```

▶ ✓ Just now (&lt;1s)

11

```
data = data.withColumn("subgrp_id", split(data["subgrp_id"], ","))
```

```
▶ data: pyspark.sql.dataframe.DataFrame = [subgrp_sk: string, subgrp_name: string ... 2 more fields]
```

Just now (1s) 12 Python

```
data.show(5, False)

▶ (2) Spark Jobs
```

subgrp_sk	subgrp_name	monthly_premium	subgrp_id
S104	Therapy	1500	[GRP103, GRP113, GRP123, GRP133, GRP143]
S105	Allergies	2300	[[GRP153, GRP104, GRP114, GRP124]]
S103	Physiology	2000	[[GRP122, GRP108, GRP138, GRP148]]
S102	Accident	1000	[[GRP110, GRP150, GRP136]]
S101	Deficiency Diseases	3000	[[GRP101, GRP105]]

only showing top 5 rows

## NOW IT AS BEEN CHANGED TO ARRAY

Just now (<1s) 13 Python

```
data = data.withColumn("subgrp_id", explode(data["subgrp_id"]))

▶ data: pyspark.sql.dataframe.DataFrame = [subgrp_sk: string, subgrp_name: string ... 2 more fields]
```

## Here the output of the data

	subgrp_sk	subgrp_name	monthly_premium	subgrp_id
	S101	Deficiency Diseases	3000	GRP101
	S101	Deficiency Diseases	3000	GRP105
	S102	Accident	1000	GRP110
	S102	Accident	1000	GRP150
	S102	Accident	1000	GRP136
	S103	Physiology	2000	GRP122
	S103	Physiology	2000	GRP108
	S103	Physiology	2000	GRP138
	S103	Physiology	2000	GRP148
	S104	Therapy	1500	GRP103
	S104	Therapy	1500	GRP113
	S104	Therapy	1500	GRP123
	S104	Therapy	1500	GRP133
	S104	Therapy	1500	GRP143
	S105	Allergies	2300	GRP153
	S105	Allergies	2300	GRP104
	S105	Allergies	2300	GRP114
	S105	Allergies	2300	GRP124

## Create an another folder in ADLS TO STAGGING THE DATA.

The screenshot shows the Microsoft Azure Storage account interface for 'optumsadls'. On the left, there's a sidebar with 'Containers', 'File shares', 'Queues', 'Tables', and 'Networking'. The main area shows a table of containers with one entry: 'optumrawdata' created on 5/4/2024 at 9:51:10 AM, marked as 'Private'. To the right, a 'New container' dialog is open, prompting for a 'Name' (set to 'I') and 'Anonymous access' (set to 'Private (no anonymous access)'). Below these settings is an 'Advanced' section with a note about access control and a list of users: 'ksrdatavision', 'storage', 'ksr\_admin', 'data', 'datavizonksradmin', and 'ppdsession'. At the bottom of the dialog are 'Create' and 'Close feedback' buttons.

```
output_container_path = "abfss://optumrawdata@optumsadls.dfs.core.windows.net"
output_blob_folder = "stagingoptumdata"  I
subgroup_data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save
(output_blob_folder)
files = dbutils.fs.ls(output_blob_folder)
output_file = [x for x in files if x.name.startswith("part-")]
dbutils.fs.mv(output_file[0].path, "%s/subgroupdatastagging.csv" % output_container_path)
```

THIS IS BOILER PLATE FOR TO PUSH THE DATA INTO STAGING DATAINTO STAGING FOLDER.

IN 1<sup>ST</sup> LINE WE NEED TO GIVE PATHA AND GIVE THE CONTAINER NAME IN WHICH FOLDER IT SHOULD BE STORE

2<sup>ND</sup> LINE WE GIVE THE FOLDER NAME

INSTEAD OF SUBGROUP\_DATA WE WRITE THE **DATA**

BECAUSE WE HAVE SOTRED THE DATEFRAME INTO DATA VARIABLE

```
1
2 output_container_path = "abfss://optumrawdata@optumsadls.dfs.core.windows.net"
3 output_blob_folder = "stagingoptumdata/"  I
4 data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save
(output_blob_folder)
5 files = dbutils.fs.ls(output_blob_folder)
6 output_file = [x for x in files if x.name.startswith("part-")]
7 dbutils.fs.mv(output_file[0].path, "%sIsubgroupdatastagging.csv" % output_container_path)
```

Execution failed with the following error:

```
I > NameError: name 'subgroup_data' is not defined
```

IN THE 7<sup>TH</sup> LINE WE NEED TO GIVE OUR OWN NAME WHICH FILE NAME YOU NEED

**WE GET AS TRUE**

```
▶ ✎ ✓ Just now (2d) 17 Python 🗑 +  
  
output_container_path = "abfss://optumrawdata@optumsadls.dfs.core.windows.net"  
output_blob_folder = "stagingoptumdata/"  
data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save(  
(output_blob_folder)  
files = dbutils.fs.ls(output_blob_folder)  
output_file = [x for x in files if x.name.startswith("part-")]  
dbutils.fs.mv(output_file[0].path, "%s/subgroupdatatagging.csv" % output_container_path)  
▶ (1) Spark Jobs
```

**SO THIS FILE IS ALL IN DEVELOPING FACE SO WE NEED TO CREATE AN ANOTHER FILE AS PRODUCTION DATA FILE.**

## SO NOW DO THE HOSPITAL DATA

The screenshot shows a Jupyter Notebook interface with a code cell at the top containing `key="adlskey")`. Below it is another cell with the following code:

```
display(dbutils.fs.ls("abfss://optumrawdata@optumsadls.dfs.core.windows.net"))
```

The output is a DataFrames with the following data:

	path	name	size	modificationTime
1	abfss://optumrawdata@optumsadls.dfs.core.windows.net/Patient_records.csv	Patient_records.csv	5110	17147965080
2	abfss://optumrawdata@optumsadls.dfs.core.windows.net/claims.csv	claims.csv	5766	17148797230
3	abfss://optumrawdata@optumsadls.dfs.core.windows.net/disease.csv	disease.csv	1489	17147965080
4	abfss://optumrawdata@optumsadls.dfs.core.windows.net/group.csv	group.csv	4390	17147965080
5	abfss://optumrawdata@optumsadls.dfs.core.windows.net/hospital.csv	hospital.csv	1528	17148797270
6	abfss://optumrawdata@optumsadls.dfs.core.windows.net/subgroup.csv	subgroup.csv	561	17147965080
7	abfss://optumrawdata@optumsadls.dfs.core.windows.net/other.csv	other.csv	4704	17147965080

```
hospital_data = spark.read.csv("abfss://optumrawdata@optumsadls.dfs.core.windows.net/hospital.csv", header = True, inferSchema=True)
```

## Drop null values

```
J Interrupt 00:01
8
hospital_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in hospital_data.columns]).show()
```

## Check the duplicates

	0	0	0	4	0

```
9
Python ⚡ ⚡ ⚡ ⚡
hospital_data.groupby(hospital_data.columns).count().where("count > 1").show() #check the duplicates| I
```

[Shift+Enter] to run and move to next cell  
[Esc H] to see all keyboard shortcuts

## We have null in state column

```
7
Python ⚡ ⚡ ⚡ ⚡
3 minutes ago (<1s)
hospital_data.show(5, False)
▶ (1) Spark Jobs
+-----+-----+-----+
|Hospital_id|Hospital_name|city|state|country|
+-----+-----+-----+
|H1000|All India Institute of Medical Sciences|New Delhi|NIN|India|
|H1001|Medanta The Medicity|Gurgaon|Haryana|India|
|H1002|The Christian Medical College|Vellore|Tamil Nadu|India|
|H1003|PGIMER - Postgraduate Institute of Medical Education and Research|Chandigarh|Haryana|India|
|H1004|Apollo Hospital - Chennai|Chennai|Tamil Nadu|India|
+-----+-----+-----+
only showing top 5 rows| I
```

```
13
Python ⚡ ⚡ ⚡ ⚡
hospital_data = hospital_data.fillna({"state": "UT"})| I
```

[Shift+Enter] to run and move to next cell  
[Esc H] to see all keyboard shortcuts

```
11
hospital_data = hospital_data.replace('NaN', None)
▶ hospital_data: pyspark.sql.dataframe.DataFrame = [Hospital_id: string, Hospital_name: string ... 3 more fields]
```

```

> (2) Spark Jobs
+-----+-----+-----+
|Hospital_id|Hospital_name          |city    |state   |country|
+-----+-----+-----+
|H1004      |Apollo Hospital - Chennai      |Chennai |Tamil Nadu|India |
|H1002      |The Christian Medical College    |Vellore |Tamil Nadu|India |
|H1003      |PGIMER - Postgraduate Institute of Medical Education and Research|Chandigarh|Haryana |India |
|H1000      |All India Institute of Medical Sciences    |New Delhi|UT     |India |
|H1001      |Medanta The Medicity           |Gurgaon |Haryana |India |
+-----+-----+-----+
only showing top 5 rows

```

▶ ⏪ ⏴ Just now (1s) 15  
`hospital_data.select("city").distinct().show(10)`

> (2) Spark Jobs

```

+-----+
|      city|
+-----+
| Chennai|
| Mumbai|
| Gurgaon|
| Vellore|
| Delhi|
| Chandigarh|
| Bengaluru|
| New Delhi|
| Hyderabad|
+-----+

```

### replace column values in pyspark

All Videos Shopping Images News More

About 1,310,000 results (0.25 seconds)

 Apache Spark  
<https://spark.apache.org/api/python/reference/api.html>

#### [pyspark.sql.DataFrame.replace](#)

`DataFrame.replace()` ... Returns a new DataFrame replacing a value with another value  
`DataFrame.replace()` and `DataFrameNaFunctions.replace()` are aliases of each other

 Stack Overflow  
<https://stackoverflow.com/questions/how-to-conditionally-replace-value-in-a-column-based-on>

How to conditionally replace value in a column based on ...

▶ ✓ Just now (<1s)

16

```
hospital_data = hospital_data.replace(['New Delhi'], ['Delhi'], 'city')
```

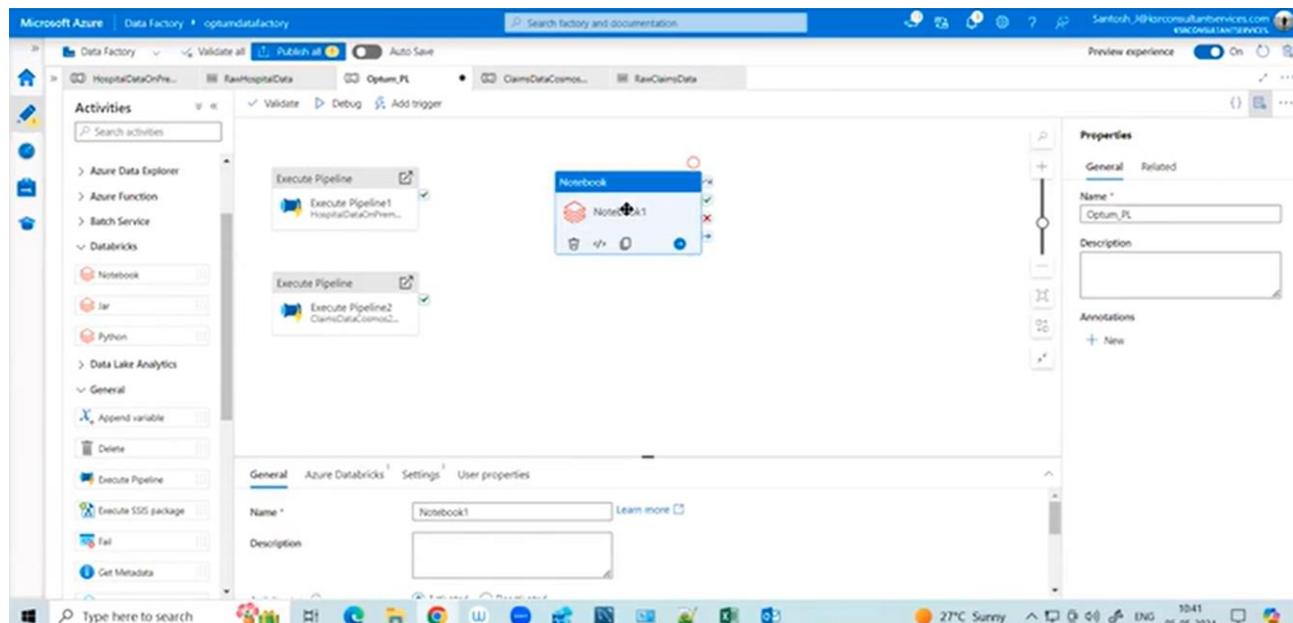
▶ hospital\_data: pyspark.sql.dataframe.DataFrame = [Hospital\_id: string, Hospital\_name: string — 3 more fields]

## Send it to staging folder

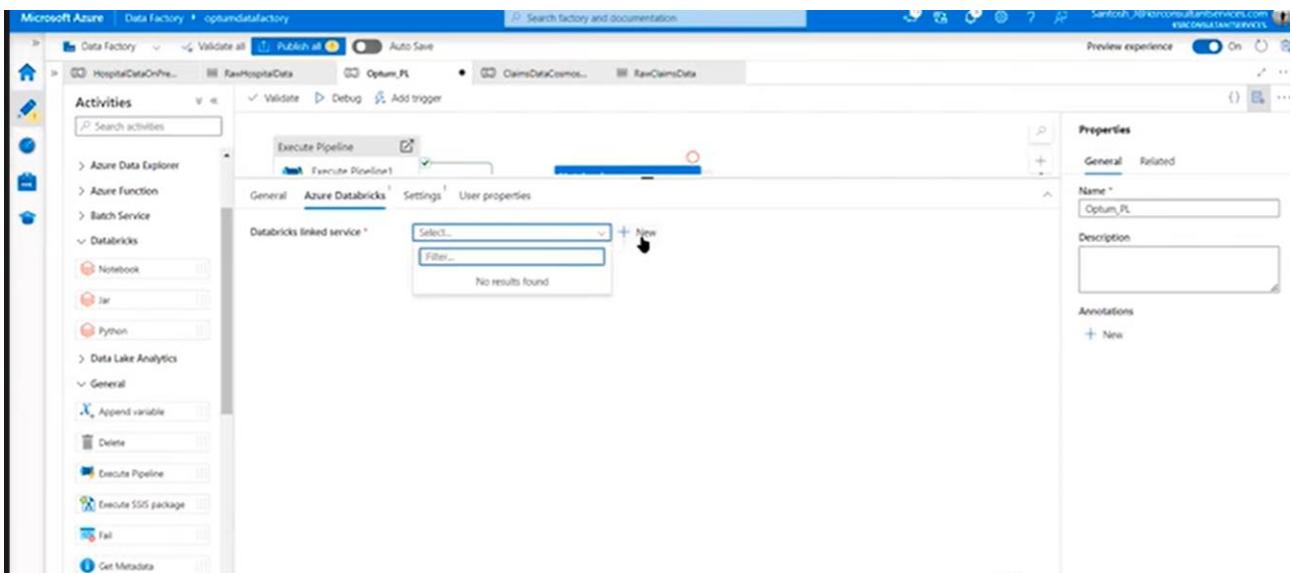
```
▶ v ✓ Just now (2s) 19 Python 🗑 + [ ] :  
output_container_path = "abfss://optumstagingdata@optumsadls.dfs.core.windows.net"  
output_blob_folder = "stagingoptumdata/"  
hospital_data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save(  
(output_blob_folder)  
files = dbutils.fs.ls(output_blob_folder)  
output_file = [x for x in files if x.name.startswith("part-")]  
dbutils.fs.mv(output_file[0].path, "%s/hospitalstagging.csv" % output_container_path)
```

## Now do it in dev to production

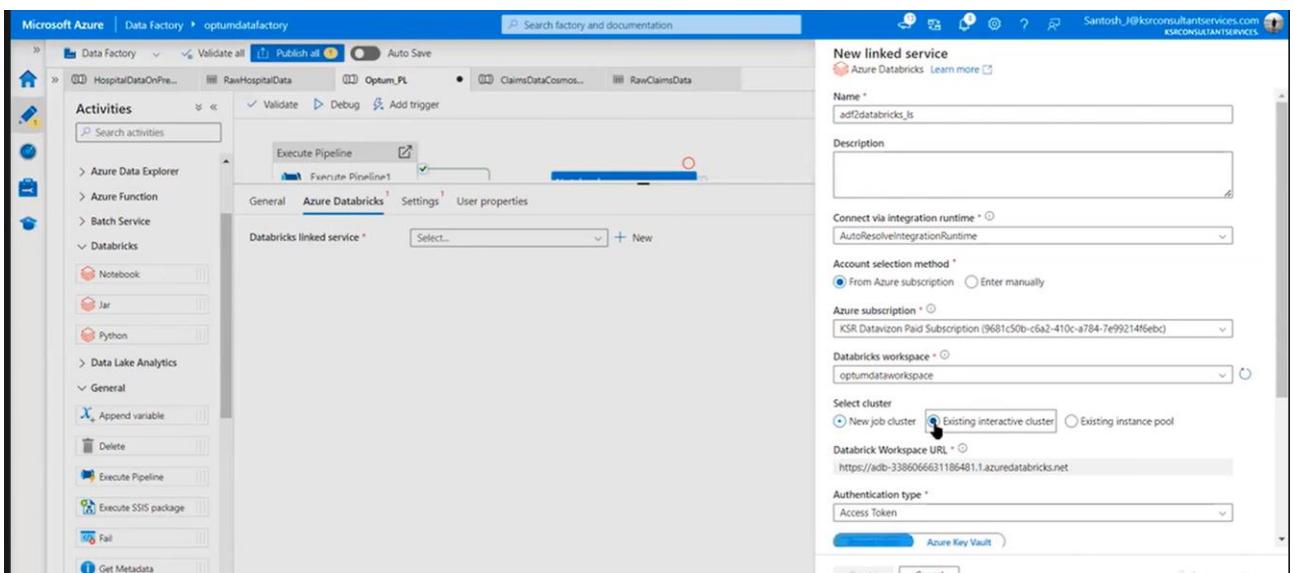
**NOW GO TO DATA FACTORY U NEED TO RUN THE PRODUCTION NOTE BOOK FILES IN  
DATAFACTORY**



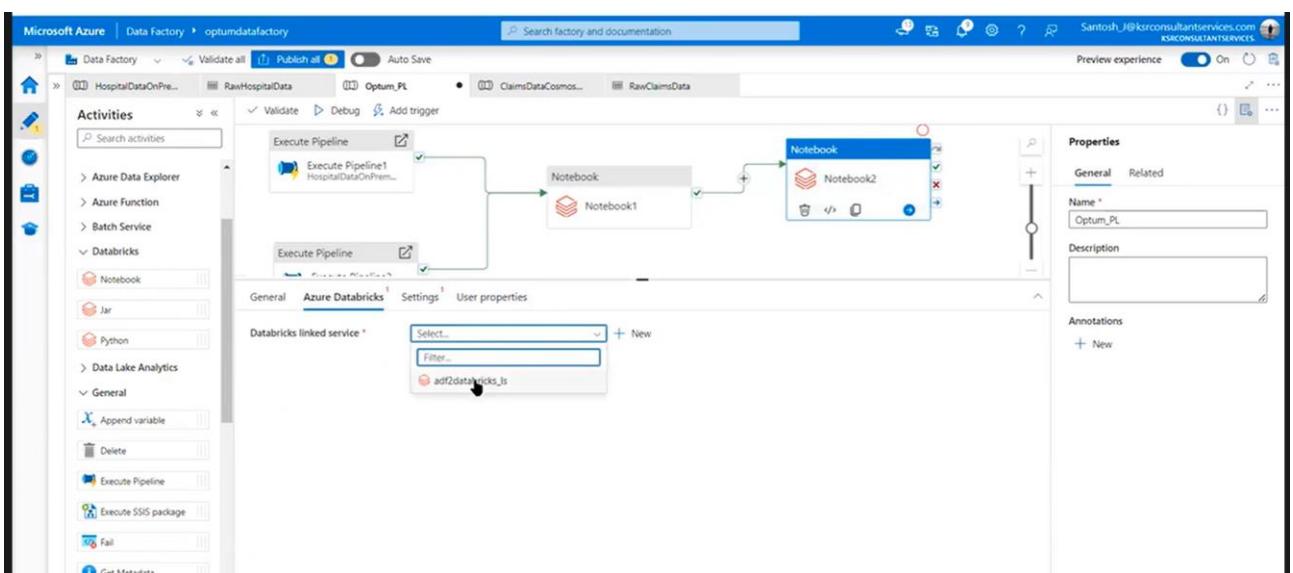
**WE CAN CALL THE NOTE BOOK HERE ALSO**



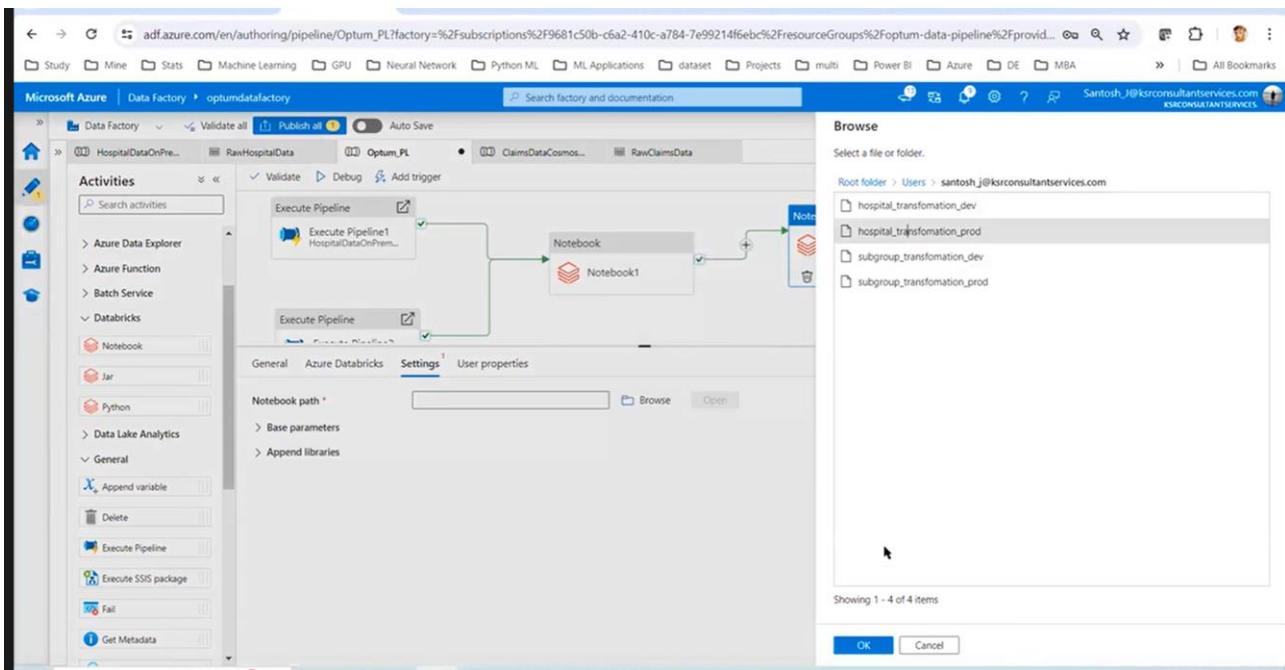
## CREATE THE LINK SERVICE FOR THE ADF TO DATA BRICKS LIKE THIS



IN CLUSTER WE NEED TO SLECT CLUSTER AS EXISTING NOW A NEW CLUSTER.



NEXT HOSPITAL DATA NOTEBOOK WE NEED TO SELCT IN ADF



## JUST DO THE PUBLISH

**NOW DELET ALL THE STAGING DATA RUN THIS PIPELINE**

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
hospitaltagging.csv	5/5/2024, 10:48:49 AM	Hot (Inferred)	Block blob	1.26 kB	Available	***
subgroupdatastagging.csv	5/5/2024, 10:48:16 AM	Hot (Inferred)	Block blob	1.09 kB	Available	***

## NOW LETS DO THE SUBGROUP DATA

The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (ClaimsDataCosmos2ADLS\_Pl, HospitalDataOnPrem2ADLS\_Pl, Optum\_Pl), 'Datasets' (CosmoClaimsData, OnPremHospitalData, RawClaimsData, RawHospitalData), and 'Data flows'. The 'Optum\_Pl' pipeline is selected. The main workspace displays a flow diagram with two 'Notebook' activities: 'Notebook1' and 'Notebook2'. A green arrow points from 'Notebook1' to 'Notebook2'. The 'Activities' pane on the right lists options like 'Move and transform', 'Synapse', 'Azure Data Explorer', etc. The 'Settings' tab is selected for the 'Notebook' activity. A 'Browse' window on the right shows a file tree under 'Root Folder > Users > santosh\_j@karconsultantservices.com' with items like 'group\_transformation\_dev', 'group\_transformation\_prod', 'hospital\_transformation\_dev', 'hospital\_transformation\_prod', 'subgroup\_transformation\_dev', and 'subgroup\_transformation\_prod'. The 'OK' button is visible at the bottom right of the dialog.

## NOW DO THE DISEAS TRANSFORMATION

The screenshot shows a Jupyter Notebook cell. The code cell contains:

```
disease_data = spark.read.csv("abfss://optumrawdata@optumsadls.dfs.core.windows.net/disease.csv", header = True, inferSchema=True)
```

The output cell shows the result of the spark.read.csv command:

```
[{"subgrp_id": "S101", "disease_id": "110001", "disease_name": "Beriberi"}, {"subgrp_id": "S101", "disease_id": "110002", "disease_name": "Scurvy"}, {"subgrp_id": "S101", "disease_id": "110003", "disease_name": "Goitre"}, {"subgrp_id": "S101", "disease_id": "110004", "disease_name": "Osteoporosis"}, {"subgrp_id": "S101", "disease_id": "110005", "disease_name": "Rickets"}]
```

Below the output, it says "only showing top 5 rows".

## CHECK THE NULL VALUES

```

Just now (1s) Python
disease_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in disease_data.columns]).show()
(2) Spark Jobs
+-----+-----+-----+
|subgrp_id|disease_id|disease_name|
+-----+-----+-----+
|      0|       0|        0|
+-----+-----+-----+

```

## CHECK THE DUPLICATES

```

Waiting Python
disease_data.groupby(disease_data.columns).count().where("count > 1").show() #check the duplicates
+-----+-----+-----+-----+
|Hospital_id|Hospital_name|city|state|country|count|
+-----+-----+-----+-----+
+-----+-----+-----+

```

```

2 days ago (20) Python
output_container_path = "abfss://optumstagingdata@optumsadls.dfs.core.windows.net"
output_blob_folder = "stagingoptumdata/"
hospital_data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save(
    output_blob_folder)
files = dbutils.fs.ls(output_blob_folder)
output_file = [x for x in files if x.name.startswith("part-")]
dbutils.fs.mv(output_file[0].path, "%s/staging.csv" % output_container_path)
True

```

12

ad.azure.com/en/authoring/pipeline/Optum\_PL?factory=%2fsubscriptions%2f9681c50b-c6a2-410c-a784-7e99214f6ebc%2fresourceGroups%2foptum-data-pipeline%2fproviders%...

Microsoft Azure | Data Factory | optumdatafactory | Search factory and documentation

Factory Resources Pipelines Optum\_PL Activities

- Pipelines
  - ClaimsDataCosmos2ADLS\_PL
  - HospitalDataOnPrem2ADLS\_PL
  - Optum\_PL**
  - Change Data Capture (preview)
- Datasets
  - CosmosClaimsData
  - OnPremHospitalData
  - RawClaimsData
  - RawHospitalData
- Data Flows
- Power Query

Validate all Publish all Auto Save

Optum\_PL

Activities

Validate Debug Add trigger

Notebook → Notebook2 → Notebook3

General Azure Databricks Settings User properties

Notebook path

Browse Select a file or folder.

Root folder > Users > sandesh\_j@karconsultantservices.com

- disease\_transformation\_dev
- disease\_transformation\_prod
- group\_transformation\_dev
- group\_transformation\_prod
- hospital\_transformation\_dev
- hospital\_transformation\_prod
- subgroup\_transformation\_dev
- subgroup\_transformation\_prod

Showing 1 - 8 of 8 items

NOW CONNECT ANOTHER NOTEBOOK INTO DATABRICKS

## LET'S TAKE THE PATIENT DATA

```
6 Python
pat_data = spark.read.csv("abfss://optumrawdata@optumsadls.dfs.core.windows.net/Patient_records.csv", header = True, inferSchema=True) I
▶ [ ] hospital_data: pyspark.sql.DataFrame = [Hospital_id: string, Hospital_name: string ... 3 more fields]
```

```
Just now (1s) 8 Python
pat_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in pat_data.columns]).show()
▶ (2) Spark Jobs
+-----+-----+-----+-----+-----+
|Patient_id|Patient_name|patient_gender|patient_birth_date|patient_phone|disease_name|city|hospital_id|
+-----+-----+-----+-----+-----+-----+
|      0|          T|            0|           0|         2|        0|   0|       0|
+-----+-----+-----+-----+-----+
```

## CHECK THE DUPLICATES

```
9 Python
pat_data.groupby(pat_data.columns).count().where("count > 1").show() #check the duplicates
▶ (2) Spark Jobs
+-----+-----+-----+-----+-----+
|Patient_id|Patient_name|patient_gender|patient_birth_date|patient_phone|disease_name|city|hospital_id|count|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

```
2 days ago (<1s) 10
patient_data = patient_data.dropDuplicates()
▶ [ ] hospital_data: pyspark.sql.DataFrame = [Hospital_id: string,
```

```
2 days ago (<1s) 11
```

## FILL THE NULL

```
pat_data = pat_data.fillna({"Patient_name": "Guest/NA"})
```

## WE DO SOME ONE CHANGE

patient\_transformation\_dev Python

File Edit View Run Help Last edit was 4 min ago New cell UI: ON Run all Santosh J's Personal Co... Schedule Share

```
pat_data.show(5, False)
```

(1) Spark Jobs

Patient_id	Patient_name	patient_gender	patient_birth_date	patient_phone	disease_name	city	hospital_id
187158	Harbir	Female	1924-06-30	+91 0112009318	Galactosemia	Rourkela	H1001
112766	Brahmdev	Female	1948-12-20	+91 1727749552	Bladder cancer	Tiruvottiyur	H1016
199252	Ujjawal	Male	1980-04-16	+91 8547451606	Kidney cancer	Berhampur	H1009
133424	Ballari	Female	1969-09-25	+91 0106026841	Suicide	Bihar Sharif	H1017
172579	Devnath	Female	1946-05-01	+91 1868774631	Food allergy	Bidhannagar	H1019

only showing top 5 rows

```
pat_data.select({count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in pat_data.columns}).show()
```

## WE DON'T NEED THE PATIENT DOB WE NEED A AGE OF A PATIENT

12

```
pat_data = pat_data.withColumn("Patient_age", (months_between(current_date(), col('patient_birth_date'))/12).cast("int"))
```

14

```
pat_data = pat_data.drop(col("patient_birth_date"))
```

2 days ago (2s) 18 Python

```

output_container_path = "abfss://optumstagingdata@optumsadls.dfs.core.windows.net"
output_blob_folder = "stagingoptumdata/"
hospital_data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save(output_blob_folder)
files = dbutils.fs.ls(output_blob_folder)
output_file = [x for x in files if x.name.startswith("part-")]
dbutils.fs.mv(output_file[0].path, "%s/patienttagging.csv" % output_container_path)

True

```

## NOW JUST TAKE A CLONE AND DO CREATE A PRODUCTION FILE FOR DATA FACTORY

The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, and Power Query. The 'Optum\_PL' pipeline is selected. The main workspace shows a sequence of activities: 'Notebook3' followed by a connector icon, and then 'Notebook4'. A 'Browse' dialog is open on the right, showing a list of items under 'Root Folder > Users > sandesh\_j@ksconsultantservices.com'. The items listed are: disease\_transformation\_dev, disease\_transformation\_prod, group\_transformation\_dev, group\_transformation\_prod, hospital\_transformation\_dev, hospital\_transformation\_prod, patient\_transformation\_dev, patient\_transformation\_prod, subgroup\_transformation\_dev, and subgroup\_transformation\_prod. The 'OK' button at the bottom of the dialog is highlighted.

## NOW TAKE A CLAIM DATASET

08:20 AM (2s)

```

claim_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in claim_data.columns]).show()
+-----+
|country|premium_written|zip_code|grp_id|grp_name|grp_type|city|
+-----+
|    0|           0|      0|     0|      0|      0|    0|

```

Waiting

```
claim_data = claim_data.dropDuplicates()

▶ [ ] group_data: pyspark.sql.dataframe.DataFrame = [cou
```

## NOW WE NEED TO REMOVE THIS

Edit View Run Help Last edit was 2 mi... New cell UI: ON ▾

Python ⚡ ⌂ ⌂ ⌂

```
✓ 2 minutes ago (<1s) 7
claim_data.show(5, False)

▶ (1) Spark Jobs
+-----+-----+-----+-----+
|claim_id|patient_id|disease_name|SUB_ID|Claim_Or_Rejected|claim_type|claim_amount|claim_date|
+-----+-----+-----+-----+
|0      |187158    |Galactosemia |SUBID1000 |N           |claims of value |79874   |1949-03-14|
|1      |112766    |Bladder cancer|SUBID10001|NaN        |claims of policy|151142  |1970-03-16|
|2      |199252    |Kidney cancer |SUBID10002|NaN        |claims of value |59924   |2008-02-03|
|3      |133424    |Suicide       |SUBID10003|NaN        |claims of fact  |143120  |1995-02-08|
|4      |172579    |Food allergy  |SUBID10004|Y           |claims of value |168634   |1967-05-23|
+-----+-----+-----+-----+
only showing top 5 rows
```

✓ Just now (<1s)

```
claim_data = claim_data.replace('NaN', None)
claim_data = claim_data.fillna({"Claim_Or_Rejected": "N"})

▶ [ ] claim_data: pyspark.sql.dataframe.DataFrame = [claim_id: integer, patient_id: integer ... 6 more fields]
```

```
✓ Just now (<1s) 9
#claim_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in claim_data.columns]).show()
```

Waiting 10 Python ⚡ ⌂ ⌂ ⌂

```
claim_data.groupby(claim_data.columns).count().where("count > 1").show() #check the duplicates
```

```
+-----+-----+-----+-----+-----+-----+-----+
|claim_id|patient_id|disease_name|SUB_ID|Claim_Or_Rejected|claim_type|claim_amount|claim_date|count|
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+
```

```

output_container_path = "abfss://optumstagingdata@optumsadls.dfs.core.windows.net"
output_blob_folder = "stagingoptumdata/"
hospital_data.coalesce(1).write.mode("overwrite").option("header", "true").format("com.databricks.spark.csv").save(output_blob_folder)
files = dbutils.fs.ls(output_blob_folder)
output_file = [x for x in files if x.name.startswith("part-")]
dbutils.fs.mv([output_file[0].path, "%s/claimstagging.csv" % output_container_path])

```

True

## MAKE A PRODUCTIN FILE AND ADD INTO THE ADF

The screenshot shows the Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, and Optum\_P1. The main workspace displays a pipeline named 'Optum\_P1' with two activities: 'ebook4' (Move and transform) and 'Notebook5' (Notebook). A 'Browse' dialog is open on the right, showing a list of Databricks notebooks under the path 'Root folder > Users > santosh\_j@ksconsultantservices.com'. The list includes items like 'claim\_transformation\_dev', 'claim\_transformation\_prod', 'disease\_transformation\_dev', etc.

This screenshot shows the same Azure Data Factory pipeline editor interface as the previous one. The pipeline 'Optum\_P1' is visible with its two activities. In the 'Notebook' activity settings, the 'Notebook path' dropdown has been updated to 'AUsersSantosh\_j@ksconsultantservices.c...', indicating a specific notebook has been selected for execution.

## SUBSCRIBER DATA SET

```
✓ 08:19 AM (5s) I 6 Python ⚡ ⚡ ⚡ ...  
subscriber_data = spark.read.csv("abfss://optumrawdata@optumsadls.dfs.core.windows.net/subscriber.csv", header = True, inferSchema=True)  
group_data: pyspark.sql.dataframe.DataFrame = [country: string, premium_written: integer ... 5 more fields]
```

```
subsciber_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in subsciber_data.columns]).show()  
()
```

	country	premium_written	zip_code	grp_id	grp_name	grp_type	city
+	0	0	0	0	0	0	0
+							

```
▶ ✓ Just now (<1s) 10
  subscriber_data = subscriber_data.dropDuplicates()

▶ [1] subscriber_data: pyspark.sql.dataframe.DataFrame = [sub_id: string, first_name: string ... 12]
```

```
subsciber_data = subsciber_data.fillna({"Elig_ind": "N", "first_name": "Guest/NA"})
```

Just now (<1s) 12 Python

```
subscriber_data = subscriber_data.fillna({"Elig_Ind": "N", "First_Name": "Guest/NA"})
subscriber_data = subscriber_data.drop('Phone')
subscriber_data = subscriber_data.withColumn("Subscriber_Age", (months_between(current_date(), col('Birth_Date'))/12).cast("int"))
subscriber_data = subscriber_data.drop(col("Birth_Date"))

> subscriber_data: pyspark.sql.DataFrame = [sub_id: string, first_name: string ... 12 more fields]
```

Waiting 6 Python

```
subscriber_data = spark.read.csv("abfss://optumrawdata@optumsadls.dfs.core.windows.net/subscriber.csv", header = True, inferSchema=True)

> subscriber_data: pyspark.sql.DataFrame = [sub_id: string, first_name: string ... 12 more fields]
```

## ILL CHECK THE MISSING VALUE COUNT

(2) Spark Jobs

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|sub_id|first_name|last_name|Street|Birth_date|Gender|Phone|Country|City|Zip_Code|Subgrp_id|Elig_Ind|eff_date|term_date|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|    0|       127|        0|      0|       0|      0|     3|      0|     0|      0|      2|      4|      0|
0|
```

Just now (1s) 13 Python

```
subscriber_data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in subscriber_data.columns]).show()
```

(2) Spark Jobs

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|sub_id|first_name|last_name|Street|Gender|Country|City|Zip_Code|Subgrp_id|Elig_Ind|eff_date|term_date|subscriber_age|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|    0|        0|        0|      0|      0|      0|     0|      0|      0|      2|      0|      0|      0|
0|
```

SO THE SUBSCRIBERS ARE SHOWING NULL

Just now (<1s)

14

Python

sub_id	first_name	last_name	Street	Gender	Country	City	Zip Code	Subgrp_id	Elig_ind	eff_date	term_date	subsciber_age
SUBID10000 NULL	Harbir	Vishwakarma	Baria Marg	Female	India	Rourkela	767058	S107	Y	30-06-1944 14-01-1954 NULL	NULL	NULL
SUBID10001 NULL	Brahmdev	Sonkar	Lala Marg	Female	India	Tiruvottiyur	34639	S105	Y	20-12-1968 16-05-1970 NULL	NULL	NULL
SUBID10002 NULL	Ujjawal	Devi	Mammen Zila	Male	India	Berhampur	914455	S106	N	16-04-2000 04-05-2008 NULL	NULL	NULL
SUBID10003 NULL	Ballari	Mishra	Sahni Zila	Female	India	Bihar Sharif	91481	S104	N	25-09-1989 05-06-1995 NULL	NULL	NULL
SUBID10004 NULL	Devnath	Srivastav	Magar Zila	Female	India	Bidhannagar	531742	S110	N	01-05-1966 09-12-1970 NULL	NULL	NULL

```
▶ v ✓ Just now (5) 13 Py  
claim_data.select("*").filter(col('SUB_ID')=='SUBID10022').show(5)  
▶ (2) Spark Jobs  
  
+-----+-----+-----+-----+-----+  
|claim_id|patient_id|disease_name|SUB_ID|Claim_Or_Rejected|claim_type|claim_amount|claim_date|  
+-----+-----+-----+-----+-----+  
|    12|    134184|        Flu|SUBID10022|          Y|claims of value|      34771|1948-05-23|  
+-----+-----+-----+-----+-----+
```

## **DO IT FOR ANTOHR PATIENT**

```

Waiting
claim_data.select("*").filter(col('SUB_ID')=='SUBID10049').show(5)

```

## NOW WE NEED DO THE REVERSE ENGINEER

```

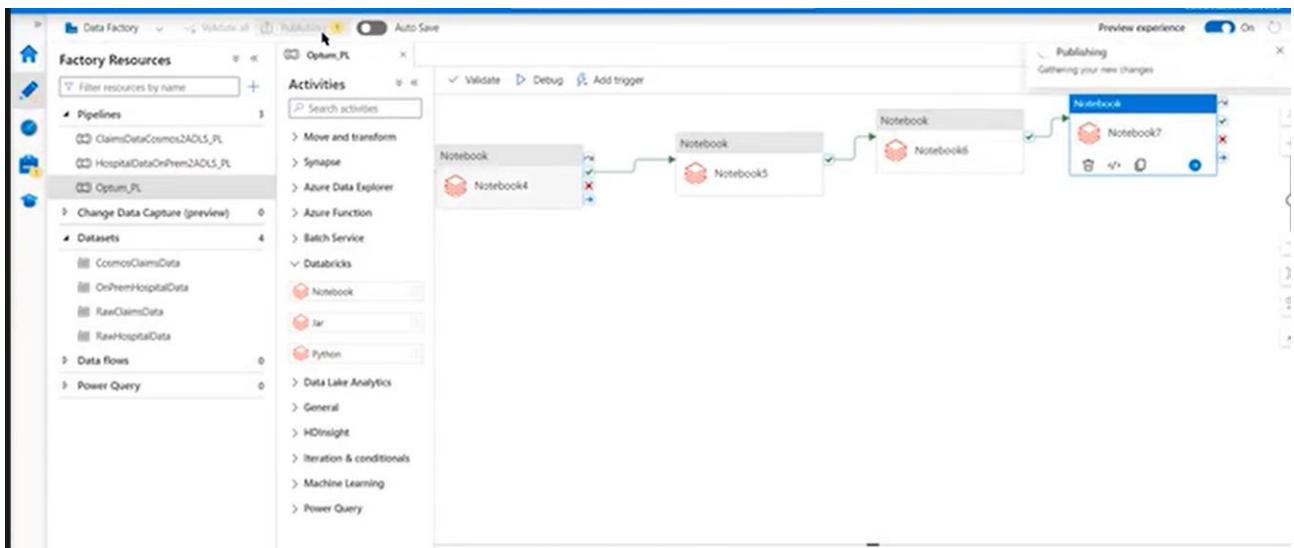
16
the reason why we are hardcoding because we dont have Subgrp_id for 2 subscriptions
subscriber_data = subscriber_data.withColumn("Subgrp_id",when(col("sub_id") =='SUBID10022', 'S110').otherwise(col("Subgrp_id")))
subscriber_data = subscriber_data.withColumn("Subgrp_id",when(col("sub_id") =='SUBID10049', 'S107').otherwise(col("Subgrp_id")))

17
✓ 2 days ago (2d) Python
output_container_path = "abfss://optumstaindata@optumdatafile.dfs.core.windows.net"
output_blob_folder = "stagingoptumdata/"
subscriber_data.coalesce(1).write.mode("Overwrite")
(output_blob_folder)
files = dbutils.fs.ls(output_blob_folder)
output_file = [x for x in files if x.name]
dbutils.fs.mv(output_file[0].path, "dbfs:/subscriber/tagging.csv" % output_container_path)
spark.csv").save
True

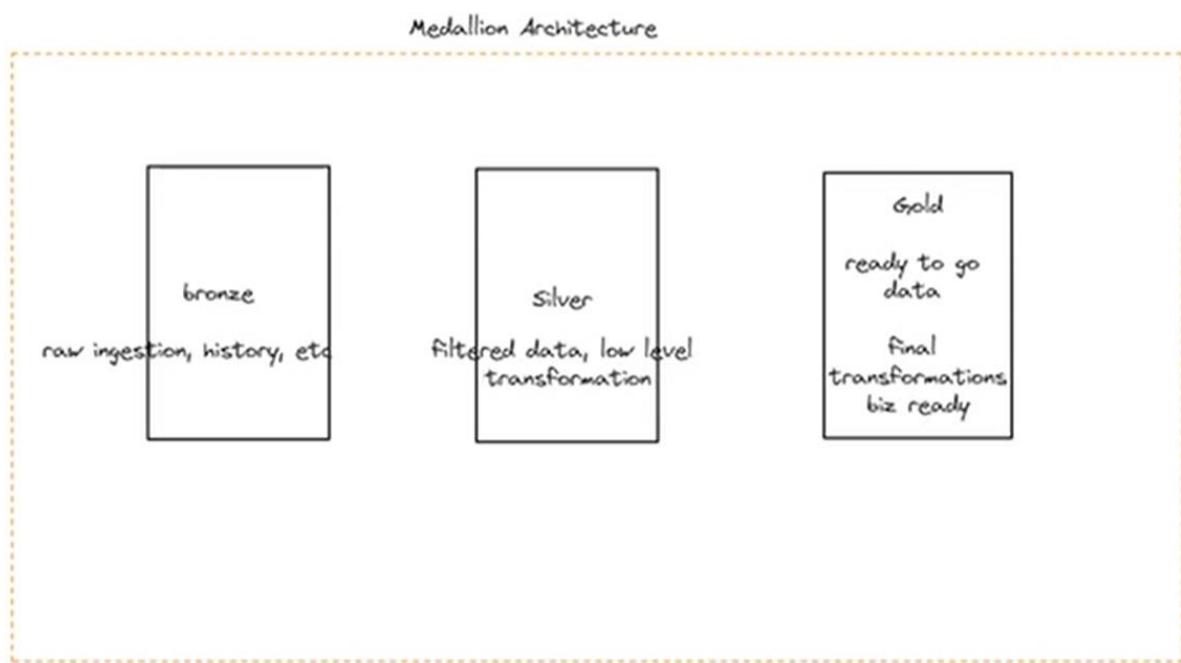
```

## NOW DO THE PRODUCTION NOTEBOOK INTO ADF

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists 'Pipelines', 'Datasets', and 'Data Flows'. In the main area, a pipeline named 'Optum\_PL' is selected. A 'Notebook' activity is being added to the pipeline. The 'Activities' pane shows 'Move and transform', 'Synapse', 'Azure Data Explorer', 'Batch Service', and 'Databricks' sections. Under 'Databricks', there are 'Notebook', 'Jar', and 'Python' options. The 'Notebook' option is highlighted. The 'Settings' tab is selected for the 'Notebook' activity. A 'Browse' dialog box is open, showing a list of notebooks in a folder structure under 'Root folder > Users > sandesh.j@krisconsultantservices.com'. The list includes 'claim\_transformation\_dev', 'claim\_transformation\_prod', 'disease\_transformation\_dev', 'disease\_transformation\_prod', 'group\_transformation\_dev', 'group\_transformation\_prod', 'hospital\_transformation\_dev', 'hospital\_transformation\_prod', 'patient\_transformation\_dev', 'patient\_transformation\_prod', 'subgroup\_transformation\_dev', 'subgroup\_transformation\_prod', 'subscriber\_transformation\_dev', and 'subscriber\_transformation\_prod'. The 'OK' button is visible at the bottom of the dialog.



**JUST PUBLISH IT**



**OUR STAGING DATA IS READY**

**NOW ITS YOU WISH TO DO THE FINAL TRANSFOMATION EITEHR USE ADF ARE NOTEBOOK**

**I WOULD PREFER ADF BECAUSE IT IS EASY TO JOIN**

**NOW THE DATA IS COMIN GFROM ADLS**

Preview data								
		CSV						
		Linked service: adls2adfl_ls						
		Object: grouptagging.csv						
		#	country	premium_written	zip_code	grp_id	grp_name	grp_type city
Group Data Capture (preview)	1	India	60000	482012	GRP142	Liberty General Insurance	Private	Mumbai
	2	India	93000	482046	GRP133	DHFL General Insurance	Private	Mumbai
	3	India	39000	482024	GRP135	Edelweiss General Insurance	Private	Mumbai
	4	India	86000	482004	GRP153	Shriram General Insurance	Private	Jaipur
	5	India	95000	482003	GRP154	Star Health and Allied Insurance	Private	Chennai
	6	India	71000	482009	GRP139	ICICI Lombard	Private	Mumbai
	7	India	70000	482006	GRP107	TATA AIG Life Insurance Co. Ltd.	Private	Mumbai
	8	India	31000	482044	GRP151	Royal Sundaram General Insurance	Private	Chennai
	9	India	60000	482006	GRP125	Acko General Insurance	Private	Mumbai
	10	India	84000	482038	GRP150	Religare Health Insurance Company Limited	Private	Gurgaon

**TO JOIN WE NEE A DATA SET**

Microsoft Azure | Data Factory | optumdatafactory

Study Mine Stats Machine Learning GPU Neural Network Python ML ML Applications database Projects multi Power BI Azure DE MBA All Bookmarks

Santosh\_J@krconsultantservices.com  
KRCONSULTANTSERVICES

Set properties

Name: StagingSubgroupData

Linked service: Select...

First row as header:

Connection	Schema	Parameters
Linked service: adls2adft_ls	Test connection	<input type="button" value="Edit"/>
File path: optumstagingdata / Directory / groupstagga		
Compression type: Select...		
Column delimiter: Comma (,)		
Row delimiter: Default (\r\n, or \n\r)		
Encoding: Default(UTF-8)		
Quote character: Double quote (")		
Escape character: Backslash (\)		

## Set properties

Name

StaggingSubgroupData

Linked service \*

adls2adf\_ls



File path

optumstagingdata

/ Directory

/ subgroupdatastagging...



First row as header



Import schema

From connection/store    From sample file    None

LIKE WE NEED TO CREATE DATA SET FOR ALL STAGINGDATASETS

### Datasets

11

CosmosClaimsData

OnPremHospitalData

RawClaimsData

RawHospitalData

StaggingClaimsData

StaggingDiseaseData

StaggingGroupData

StaggingHospitalData

StaggingPatData

StaggingSubgroupData

StaggingsubscriberData

HERE IT IS DATASETS

## LETS GO TO THE DATAFLOW

The screenshot shows the Azure Data Factory Data Flow blade. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, and Data flows. The 'Data flows' section shows one item named 'Optum\_DF'. The main workspace displays a data flow diagram with a 'source1' component. Below the diagram, the 'Properties' pane shows the data flow is named 'Optum\_DF'. The 'Source settings' tab is selected, showing the source dataset is 'StaggeringGroupData'. The 'Projection' tab is active, displaying a mapping table:

source1's column	Name as
abc_country	country
abc_premium_written	premium_written
abc_zip_code	zip_code
abc_grp_id	grp_id
abc_grp_name	grp_name
abc_grp_type	grp_type
abc_city	city

The 'Inspect' tab is also visible at the top of the blade.

## MAKE THE GROUP ID FIRST

The screenshot shows the Azure Data Factory Data Flow blade with a modified mapping configuration. The 'Projection' tab is selected, and the mapping table now includes 'grp\_id' as the first column:

source1's column	Name as
abc_groupdata	grp_id
abc_premium_written	premium_written
abc_zip_code	zip_code
abc_grp_id	grp_id
abc_grp_name	grp_name
abc_grp_type	grp_type
abc_city	city

The 'Output stream name' field is set to 'subgroupdata'. The 'Source settings' tab shows the source dataset is 'StaggeringSubgroupData'.

## NEED TO EDIT THE COLUMN NAME

Microsoft Azure | Data Factory > optimdatafactory

Search factory and documentation

Validate all | Publish all | Auto Save

Factory Resources | Filter resources by name

- Pipelines: 3
- Change Data Capture (preview): 0
- Datasets: 11
- Data flows: 1
- Power Query: 0

✓ Validate | Data flow debug | Debug Settings

groupdata → selectedgroupdata  
Import data from StagingGroupData

Renaming groupdata to selectedgroupdata with columns 'grp\_id, grp\_name, country, premium\_written'.

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

Number of rows: 38 | INSERT 38 | UPDATE 0 | DELETE 0 | UPSERT 0 | LOOKUP 0 | ERROR 0 | TOTAL 38

subgrp_sk	subgrp_name	monthly_premium	subgrp_id
S108	Infectious disease	1500	GRP130
S108	Infectious disease	1500	GRP104
S108	Infectious disease	1500	GRP109
S104	Therapy	1500	GRP103
S104	Therapy	1500	GRP113
S104	Therapy	1500	GRP123
S104	Therapy	1500	GRP133
S104	Therapy	1500	GRP143
S105	Allergies	2300	GRP153
S105	Allergies	2300	GRP104

groupdata → selectedgroupdata  
Import data from StagingGroupData

Renaming subgroupdata to select1 with columns 'subgrp\_id, subgrp\_name, monthly\_premium, grp\_id'.

Select settings | Optimize | Inspect | Data preview

Output stream name: select1 | Learn more

Description: Renaming subgroupdata to select1 with columns 'subgrp\_id, subgrp\_name, monthly\_premium, grp\_id'

Incoming stream: subgroupdata

Options:
  Skip duplicate input columns
  Skip duplicate output columns

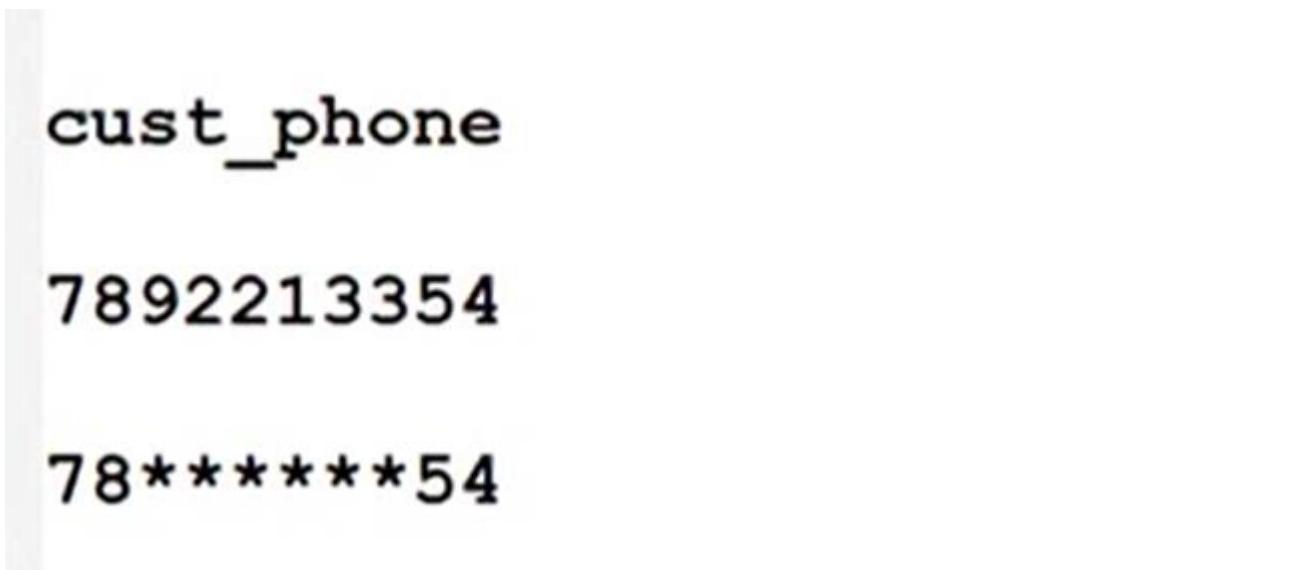
Input columns: 4 mappings: All inputs mapped

subgroupdata's column	Name as
Subgrp_sk	subgrp_id
Subgrp_name	subgrp_name
monthly_premium	monthly_premium
Subgrp_id	grp_id

NOW DO ANOTHER HOSPITAL DATA

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists Pipelines (3), Change Data Capture (preview) (0), Datasets (11), Data flows (1), and Power Query (0). The main workspace displays a data flow step named 'Optum\_DF'. The data flow consists of two stages: 'Import data from StagingSubgroupData' followed by a transformation step 'Renaming subgroupdata to subgroupdata with column mapping: subgroup\_id, subgroup\_name, premium, group\_id'. The 'Source settings' tab is selected, showing the output stream name 'source1', dataset 'StagingHospitalData', and options like 'Allow schema drift' checked. The 'Properties' panel on the right shows the name 'Optum\_DF'.

THIS IS AND ASSIGNMENT FOR U



NOW AGAIN CLAIMS DATA SOURCE

The screenshot shows the Microsoft Azure Data Factory interface, similar to the previous one but for a different pipeline step. The 'Factory Resources' sidebar shows the same structure. The main workspace displays a data flow step named 'Optum\_DF'. The data flow consists of two stages: 'Import data from StagingClaimsData'. The 'Source settings' tab is selected, showing the output stream name 'claimsdata', dataset 'StagingClaimsData', and options like 'Allow schema drift' checked. The 'Properties' panel on the right shows the name 'Optum\_DF'.

Microsoft Azure | Data Factory > optimudatafactory

Validate all | Publish all | Auto Save | Preview experience | On

**Factory Resources**

- Pipelines 3
- Change Data Capture (preview) 0
- Datasets 11
- Data flows 1
- Power Query 0

**Optum\_DF**

**Source settings**

Output stream name: diseasedata | Learn more | Reset

Description: Import data from StaggingDiseaseData

Source type: Dataset | Inline

Dataset: StaggingDiseaseData | Test connection | Open | New

Options: Allow schema drift (checked), Infer drifted column types, Validate schema

Skip line count: [ ]

Sampling: Enable (radio button)

**Properties**

Name: Optum\_DF

Description:

Microsoft Azure | Data Factory > optimudatafactory

Validate all | Publish all | Auto Save | Preview experience | On

**Factory Resources**

- Pipelines 3
- Change Data Capture (preview) 0
- Datasets 11
- Data flows 1
- Optum\_DF 0
- Power Query 0

**Optum\_DF**

**Source settings**

Output stream name: subscriberdata | Learn more | Reset

Description: Import data from StaggingsubscriberData

Source type: Dataset | Inline

Dataset: StaggingsubscriberData | Test connection | Open | New

Options: Allow schema drift (checked), Infer drifted column types, Validate schema

**Properties**

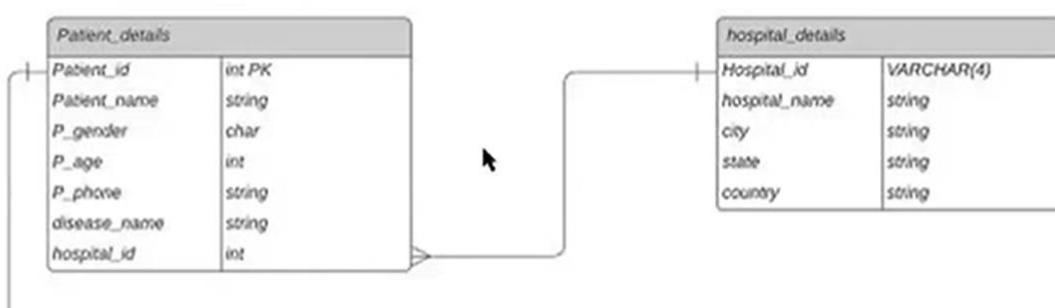
Name: Optum\_DF

Description:

## OBSERVE THIS

Table Schema

Yogesh Bantai | May 23, 2022



IN ONE HSPTL HAVE MANY PATIENTS

Microsoft Azure | Data Factory > optumdatafactory

Search factory and documentation

Validate all Auto Save

Preview experience On

Santosh.J@ksconsultantservices.com  
KSCONSULTANTSERVICES

Factory Resources

- Pipelines 3
- Change Data Capture (preview) 0
- Datasets 11
- Data flows 1

Optum\_DF

Power Query 0

StagingSubgroupD... StagingHospitalData StagingClaimsData StagingDiseaseData StagingPatData StagingsubscriberD...

Validate Data flow debug Debug Settings

hospitaldata

join1

Counting 5 total

patdata

Join settings Optimize Inspect Data preview

Output stream name \* pathodata Learn more

Description Add second stream to the join from settings Reset

Left stream \* hospitaldata

Right stream \* patdata

Join type \* Full outer Inner Left outer Right outer Custom (cross)

Use fuzzy matching

Join conditions \* Left: hospitaldata's column Right: patdata's column

Select column... == Select column... +

Properties General Related

Name \* Optum\_DF

Description

Join settings Optimise Inspect Data preview

Output stream name \* pathodata Learn more

Description Inner join on 'hospitaldata' and 'patdata' Reset

Left stream \* hospitaldata

Right stream \* patdata

Join type \* Full outer Inner Left outer Right outer Custom (cross)

Use fuzzy matching

Join conditions \* Left: hospitaldata's column Right: patdata's column

Hospital\_Id == hospital\_id +

← Previous Next →

WWE CITY IN 2 TIMES

Microsoft Azure | Data Factory > optimdatafactory

Factory Resources

Data preview

id	Patient_id	city	country	disease	Hospital	city
68	80	Chandigarh	Haryana	Measles	P...	Berhampore
32	47	Gurgaon	Haryana	Choking	M...	Ghaziabad
13	65	Hyderabad	Telangana	Bladder ...	A...	Jabalpur
88	77	Mumbai	Maharashtra	Colorect...	F...	Baranagar
09	88	Bengaluru	Karnataka	Flu	A...	Morbi
51	41	Mumbai	Maharashtra	Anthrax	L...	Karimnagar
14	69	Mumbai	Maharashtra	Breast c...	F...	Ghaziabad
31	32	Hyderabad	Telangana	Cystic fi...	Y...	Ambala
97	93	Bengaluru	Karnataka	Choking	M...	Shivpuri
69	81	Mumbai	Maharashtra	Anæmia	J...	Muzaffarpur
86	78	Bengaluru	Karnataka	Food all...	A...	Bidhan Nagar
90	99	Bengaluru	Karnataka	Glaucoma	M...	Jabalpur
68	91	Mumbai	Maharashtra	Diabetes	J...	Satna
17	98	Delhi	UT	Asthma	S...	Morbi

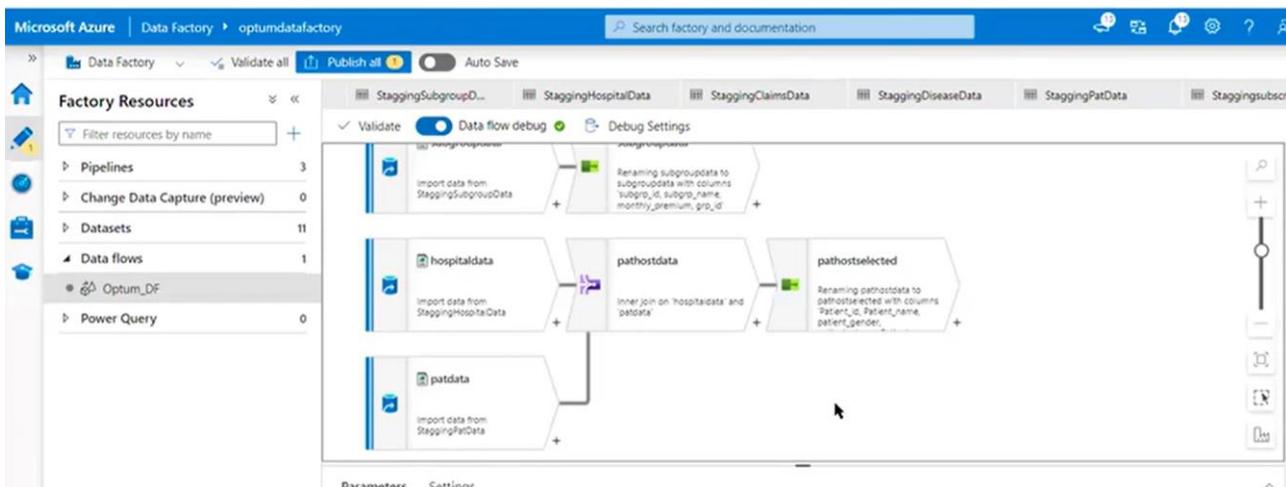
Properties

Name: Optum\_DF

Description:

NAME IT CLEAR ONE IS HSPTL CITY ,ONE AS PATIENT CITY

IF BOTH AS SAME VALUE I WILL REMOVE



Microsoft Azure | Data Factory > optimdatafactory

Factory Resources

Data flows

Join settings

Output stream name: join1

Description: Inner join on 'pathosselected' and 'claimsdata'

Left stream: pathosselected

Right stream: claimsdata

Join type: Inner

Use fuzzy matching:

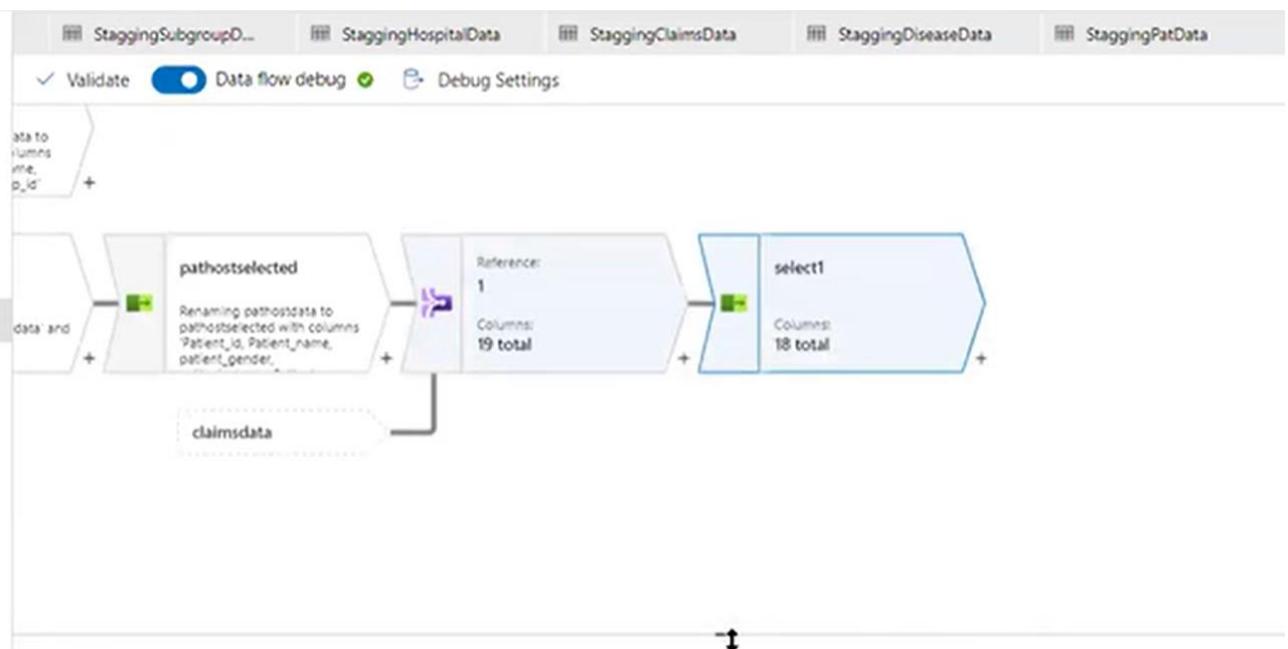
Join conditions:

Left: pathosselected's column	Right: claimsdata's column
Patient_id	patient_id

Properties

Name: Optum\_DF

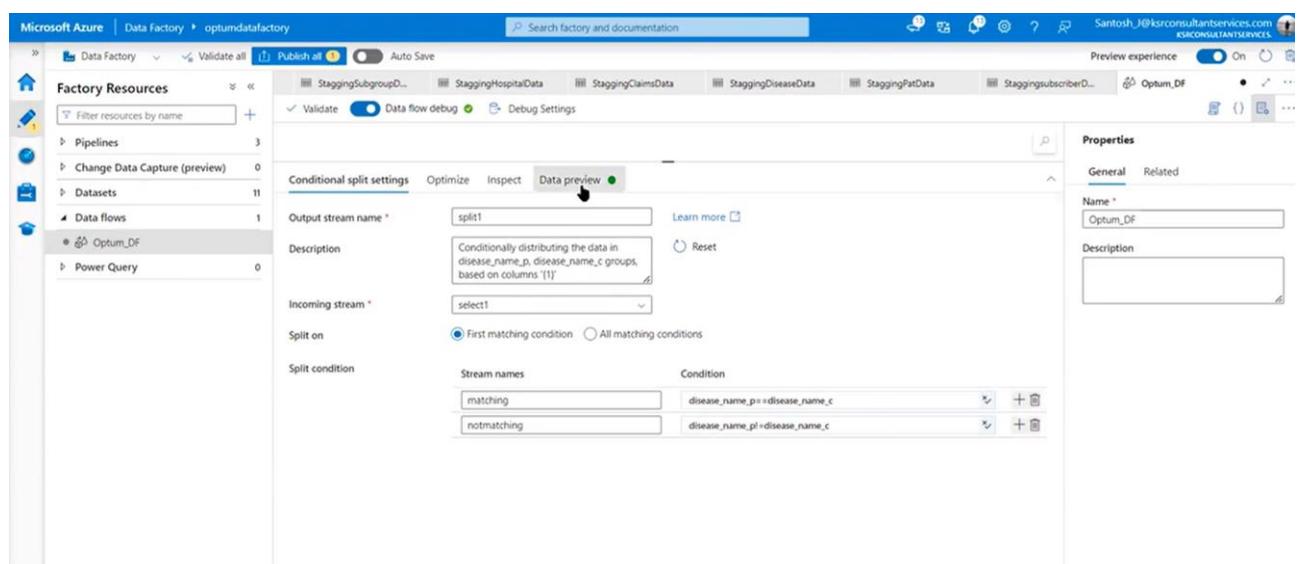
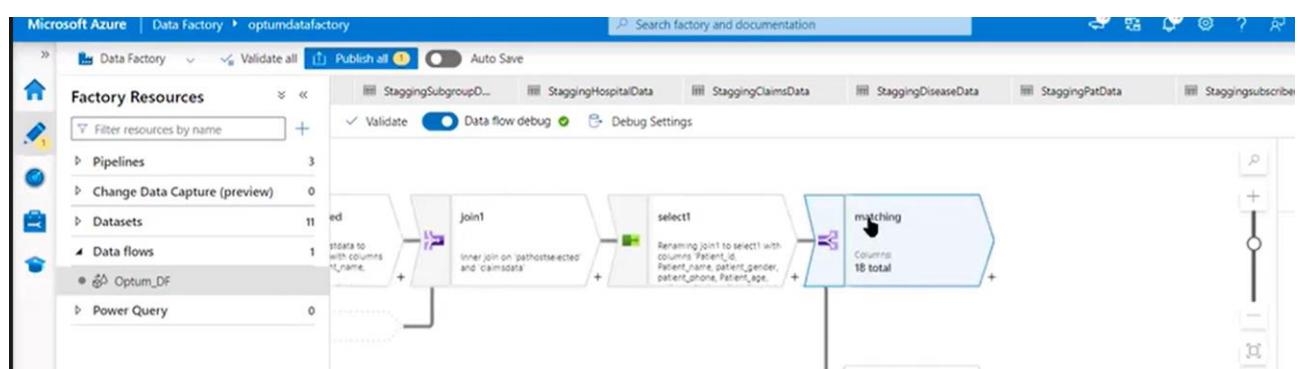
Description:

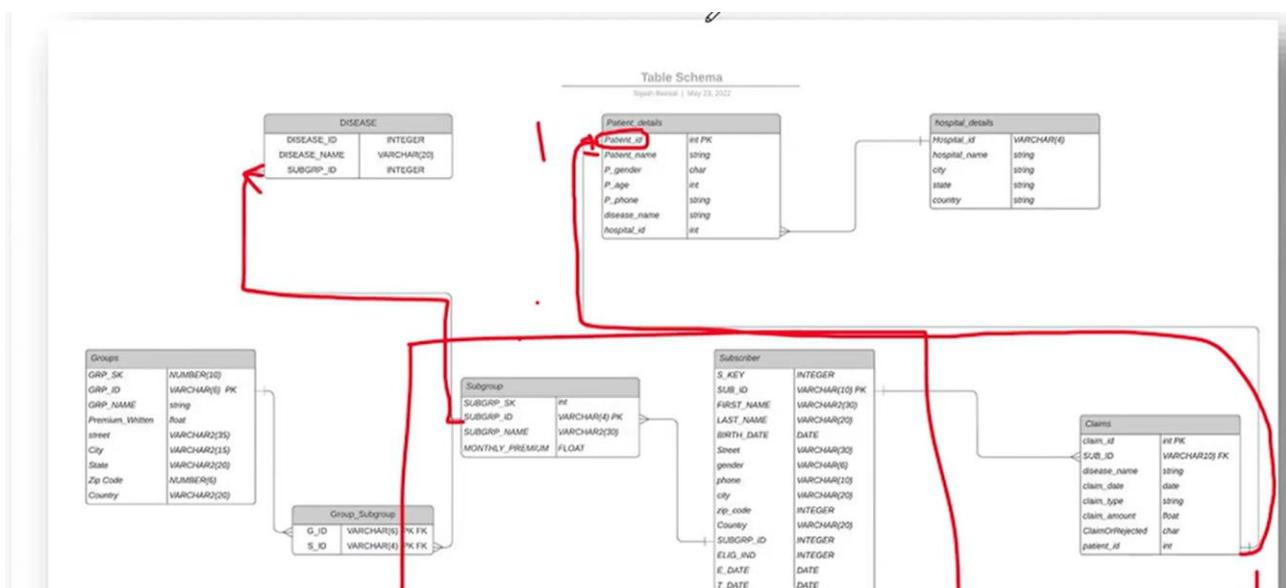
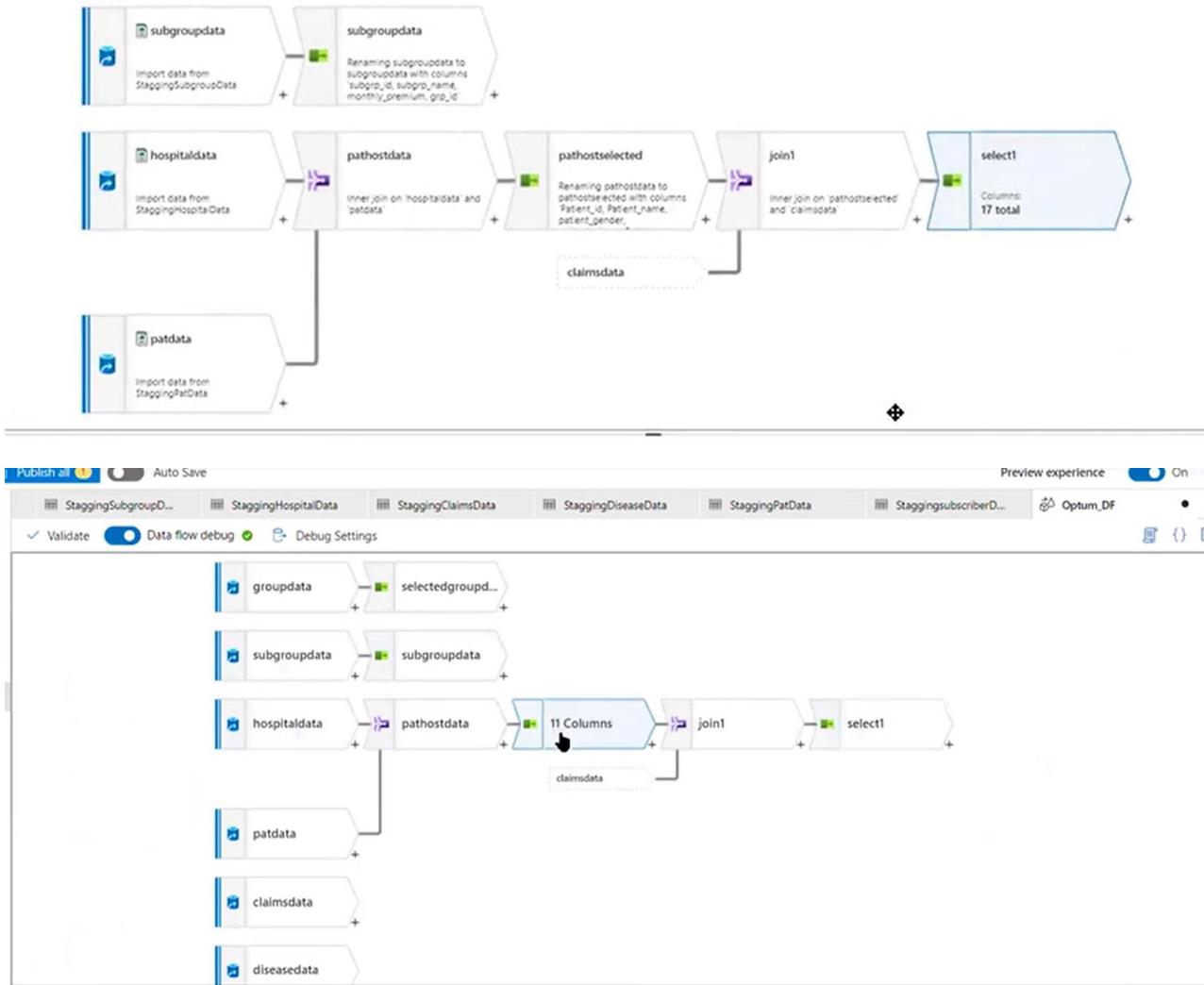


**CHECK ALL THE COLUMNS**

**CHECK WHILE IT AS SAME**

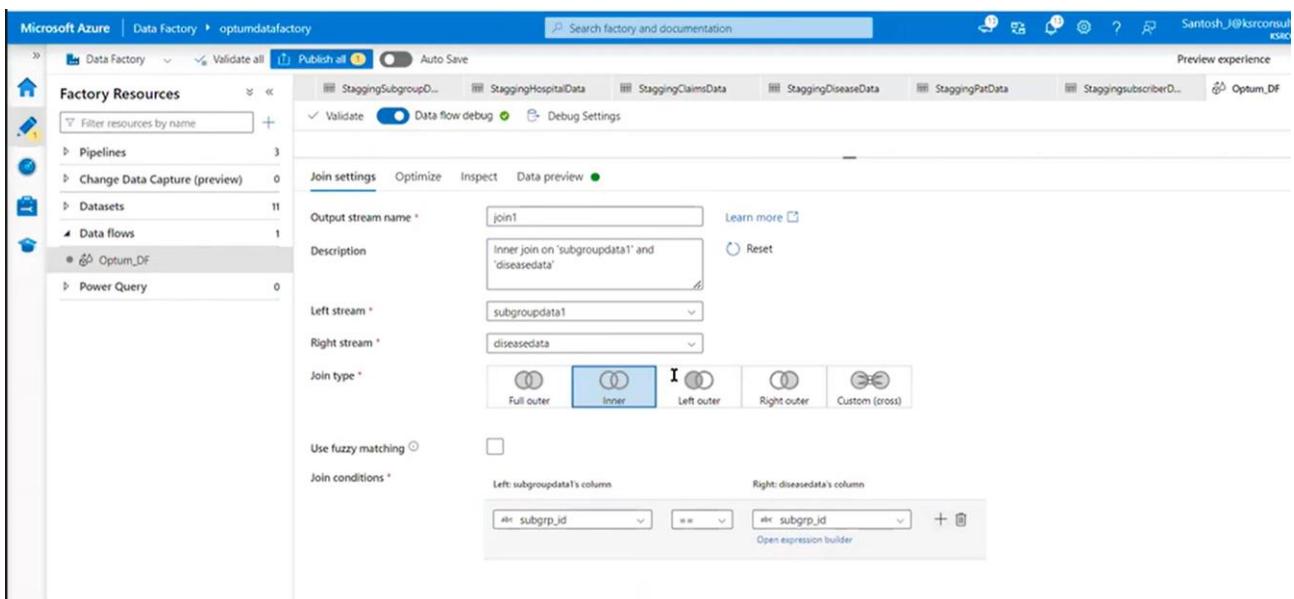
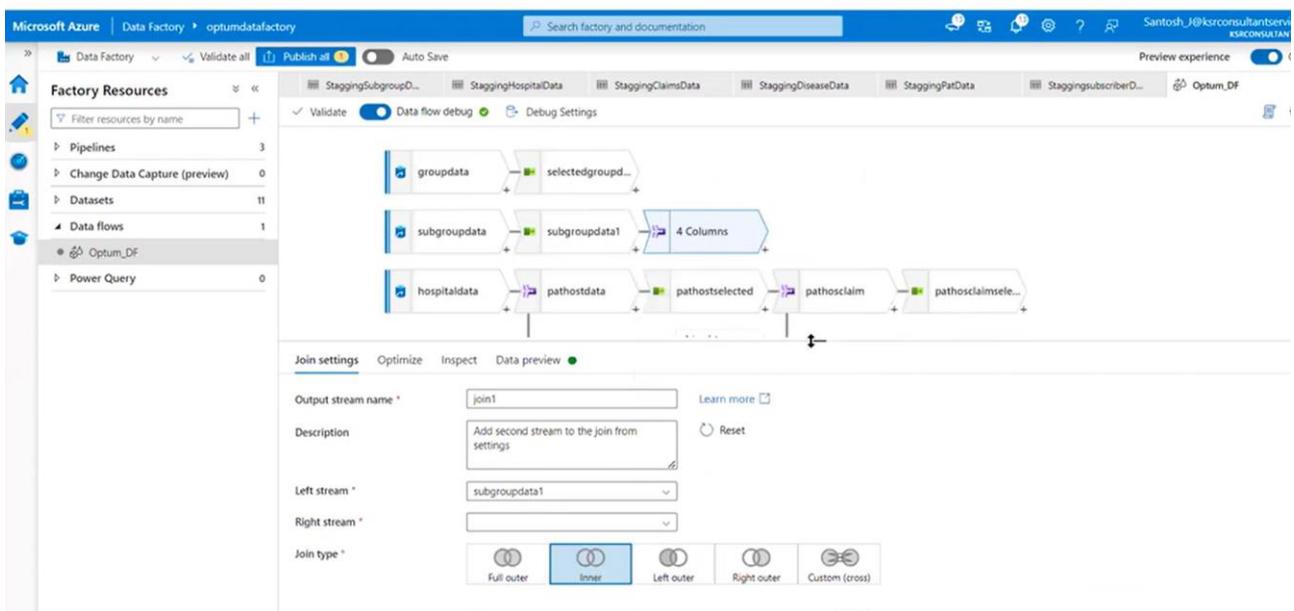
**WITHOUT DOING MANUALLY WE CAN USE THE CONDITIONAL ACTIVITY**



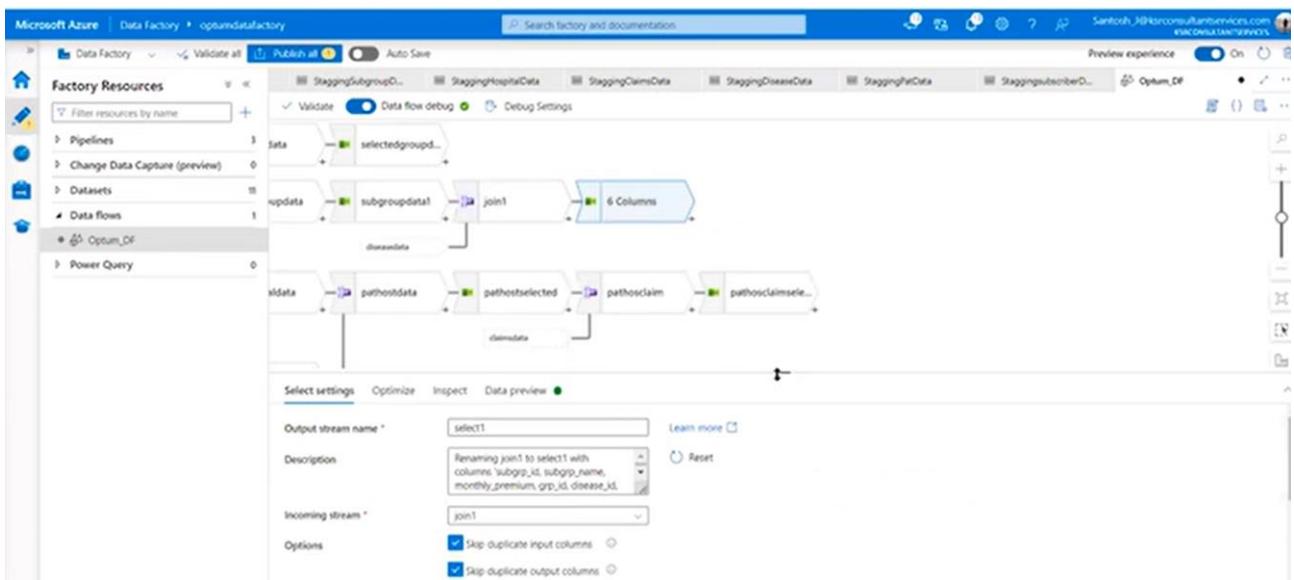


NOW LETS GO TO DISEAS DATA SOURCE

LETS JOIN THIS WITH SUBGROUP DATA



We get the datapreview and we got the groupid in both tables so we need to remove this group id



## We use the selected activity

This screenshot shows the 'Input columns' mapping section for the 'select1' activity. It maps 7 columns from the 'join1' incoming stream to new names:
 

Join1's column	Name as
subgrp_id	subgrp_id
subgrp_name	subgrp_name
monthly_premium	monthly_premium
grp_id	grp_id
diseasedata@subgrp_id	subgrp_id
disease_id	disease_id
disease_name	disease_name

 The 'Add mapping' button is visible at the bottom right of the mapping grid.

Arrenge like this

Output stream name: select1

Description: Renaming join1 to select1 with columns 'grp\_id', 'disease\_name', 'disease\_id', 'subgrp\_name'.

Incoming stream: join1

Options: Skip duplicate input columns, Skip duplicate output columns

Input columns:

join1's column	Name as
grp_id	grp_id
disease_name	disease_name
disease_id	disease_id
subgrp_name	subgrp_name
monthly_premium	monthly_premium

Join settings: Use fuzzy matching

Join conditions:

- Right: Select column...
- selectedgroupdata
- subgroupdata
- pathostdata
- pathosselected

## JOIN IT WITH GROUP ID

Output stream name: join2

Description: Inner join on 'dissubdata' and 'selectedgroupdata'

Left stream: dissubdata

Right stream: selectedgroupdata

Join type: Inner

Use fuzzy matching

Join conditions:

Left: dissubdata's column	=	Right: selectedgroupdata's column
grp_id	=	grp_id

## NOW USE THE SELECTED COLUMNS AND REMOVE SOME COLUMNS

The screenshot shows the Microsoft Azure Data Factory Data Flow interface. A pipeline named 'Optum\_DF' is displayed. The pipeline consists of several stages: 'subgroupdata', 'join1', 'dissubdata', '13 Columns', 'pathosdata', 'pathosselected', 'pathosclaim', and 'pathosclaimselected'. The '13 Columns' stage is currently selected. In the 'Data preview' tab, a table is shown with the following columns:

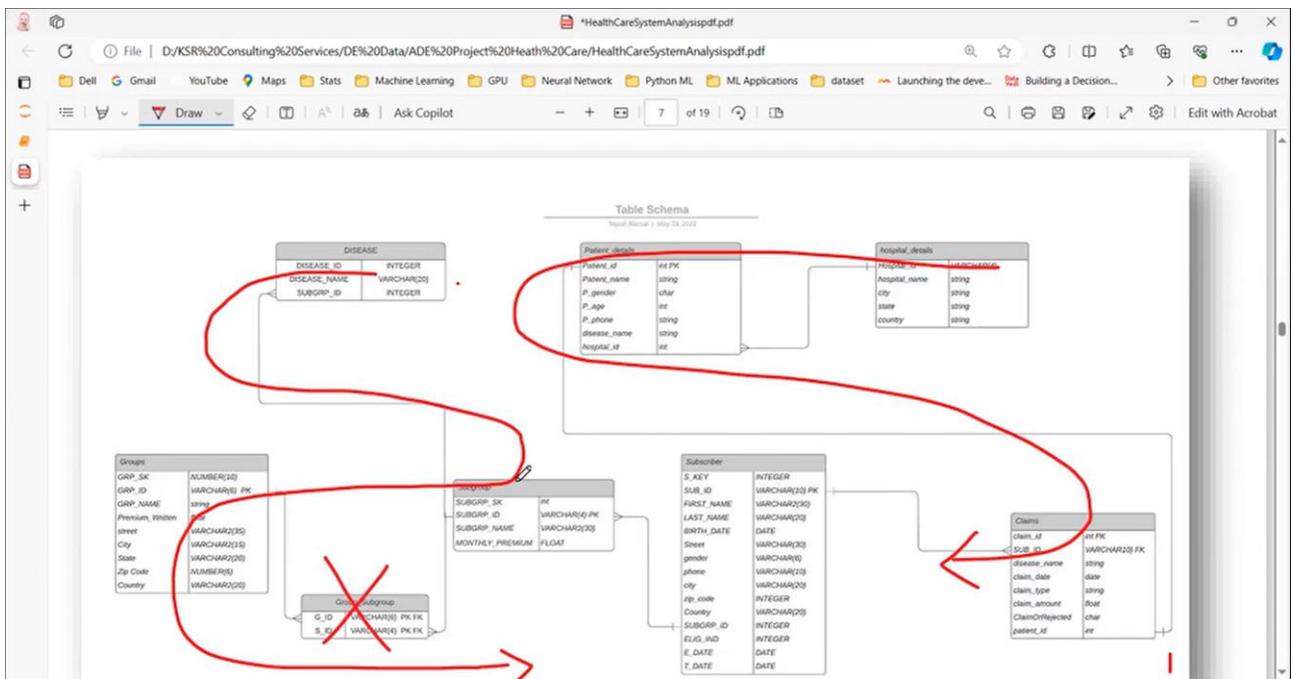
subgrp_name	monthly_premium	grp_id	grp_type	city	disease_id	disease_name	subgrp_id	subgrp_name	grp_name	country	premium_written	zip_code	grp_type	
Infectious di...	1500	GRP130	GRP130	Bhan	Uniprot								Private	Mumbai
Infectious di...	1500	GRP130	GRP130	Bharti AXA Gen...	XXXXXX								Private	Mumbai
Infectious di...	1500	GRP130	GRP130	Bharti AXA Gen...	42000								Private	Mumbai
Infectious di...	1500	GRP130	GRP130	Bharti AXA Gen...	42000								Private	Mumbai

The 'Selected' schema modifier is applied to the '13 Columns' stage.

THIS WE WILL KEEP LIKE THIS

The screenshot shows the Microsoft Azure Data Factory Data Flow interface. The 'Select settings' tab is selected for the 'join2' stage. The 'Incoming stream' dropdown is set to 'join2'. Under 'Options', the checkboxes for 'Skip duplicate input columns' and 'Skip duplicate output columns' are checked. The 'Input columns' section shows a mapping table:

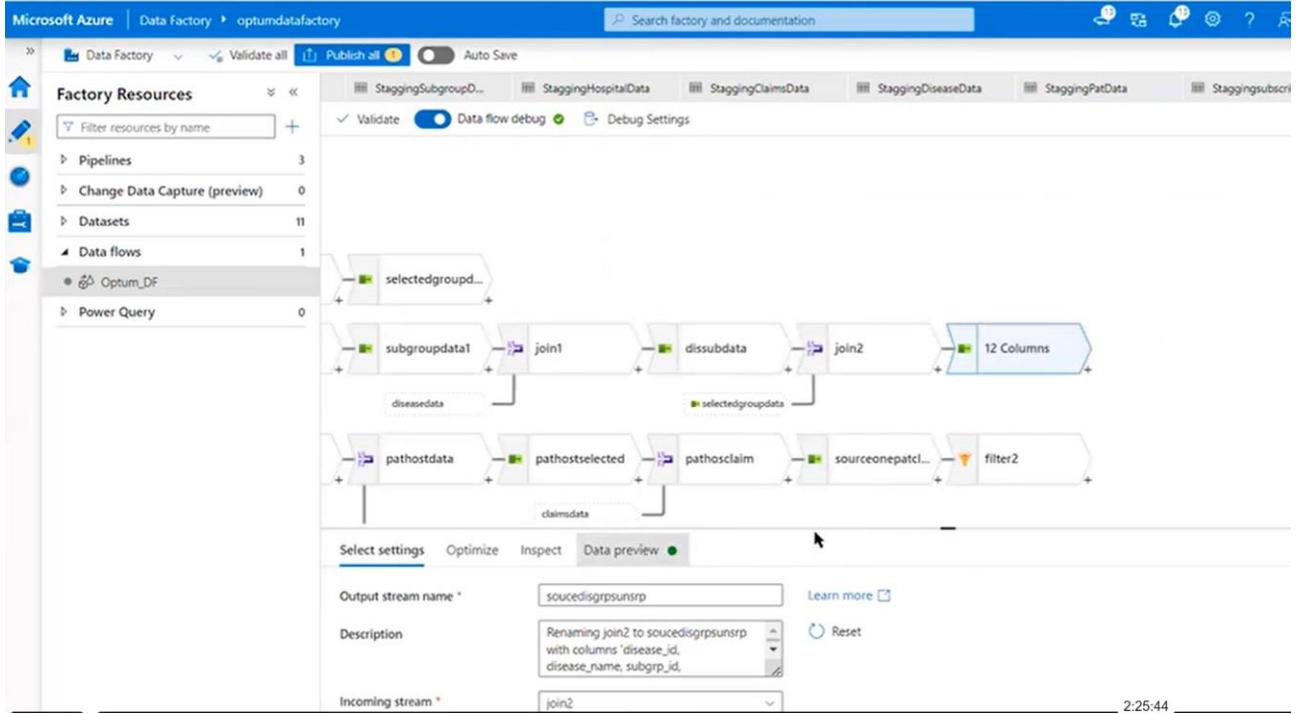
join2's column	Name as
disease_id	disease_id
disease_name	disease_name
subgrp_id	subgrp_id
subgrp_name	subgrp_name
monthly_premium	monthly_premium
selectedgroupdata[grp_id]	grp_id
grp_name	grp_name
country	country
premium_written	premium_written
zip_code	zip_code
grp_type	grp_type



**WE HAVE DONE LIKE THIS**

**NOW LETS GO TO THE SUBSCRIBER DATA.**

**IN SUBSCRIBER DATA WE GET THE SUBGROUP DATA FROM SUBGROUPDATASET**



**12 COL IS A SUBGROUP ID**

**GET THE SUGRGROUP ID INFORMATION TO SUBSCRIBER DATA**

**USE JOIN**

Santosh.J@krsconsultantservices.com  
KRS CONSULTANT SERVICES

Microsoft Azure | Data Factory > optimumdatafactory

Search factory and documentation

Validate all Auto Save

StagingSubgroupD... StagingHospitalData StagingClaimsData StagingDiseaseData StagingPatData StagingsubscriberD... Optum\_DF

Preview experience On

**Factory Resources**

- Pipelines 3
- Change Data Capture (preview) 0
- Datasets 11
- Data flows 1
- Optum\_DF
- Power Query 0

**Join settings**

Output stream name: join3

Description: Inner join on 'subscriberdata' and 'subgroupdata1'.

Left stream: subscriberdata

Right stream: subgroupdata1

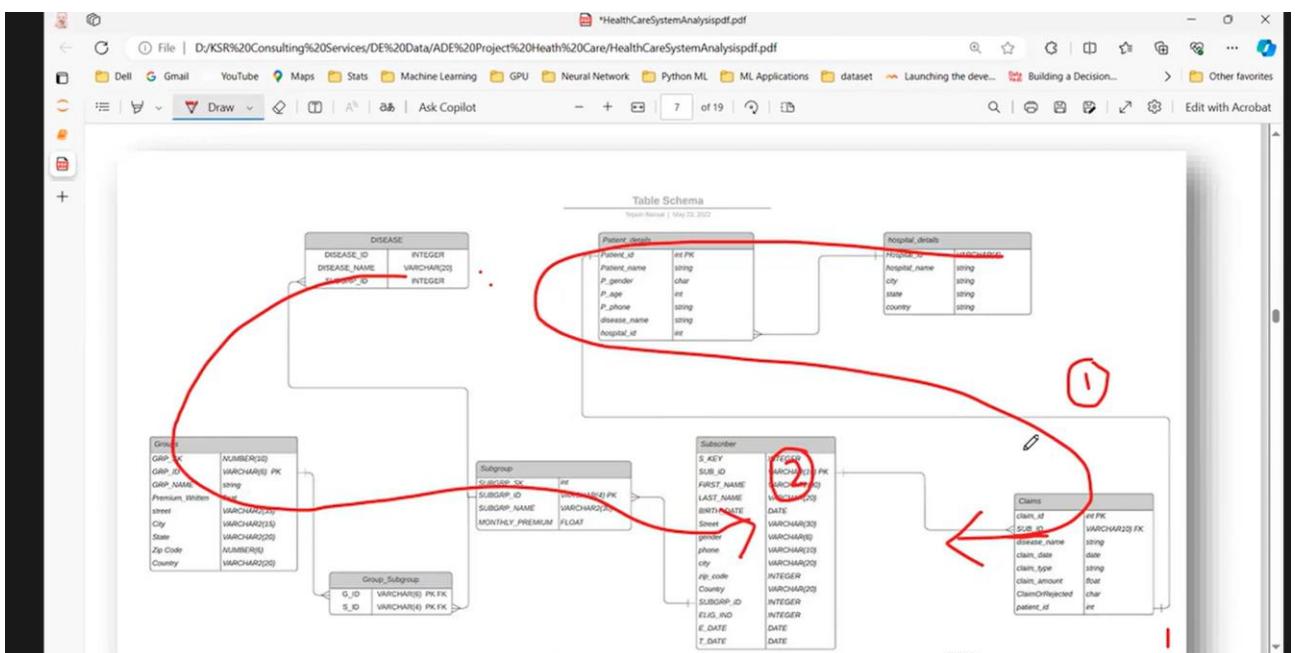
Join type: Inner (selected)

Use fuzzy matching:

Join conditions:

- Left: subscriberdata's column: Subgrp\_id
- Right: subgroupdata1's column: subgrp\_id

## WE HAVE DONE THE 2 TABLES



Santosh.J@krsconsultantservices.com  
KRS CONSULTANT SERVICES

Microsoft Azure | Data Factory > optimumdatafactory

Search factory and documentation

Validate all Auto Save

StagingSubgroupD... StagingHospitalData StagingClaimsData StagingDiseaseData StagingPatData StagingsubscriberD... Optum\_DF

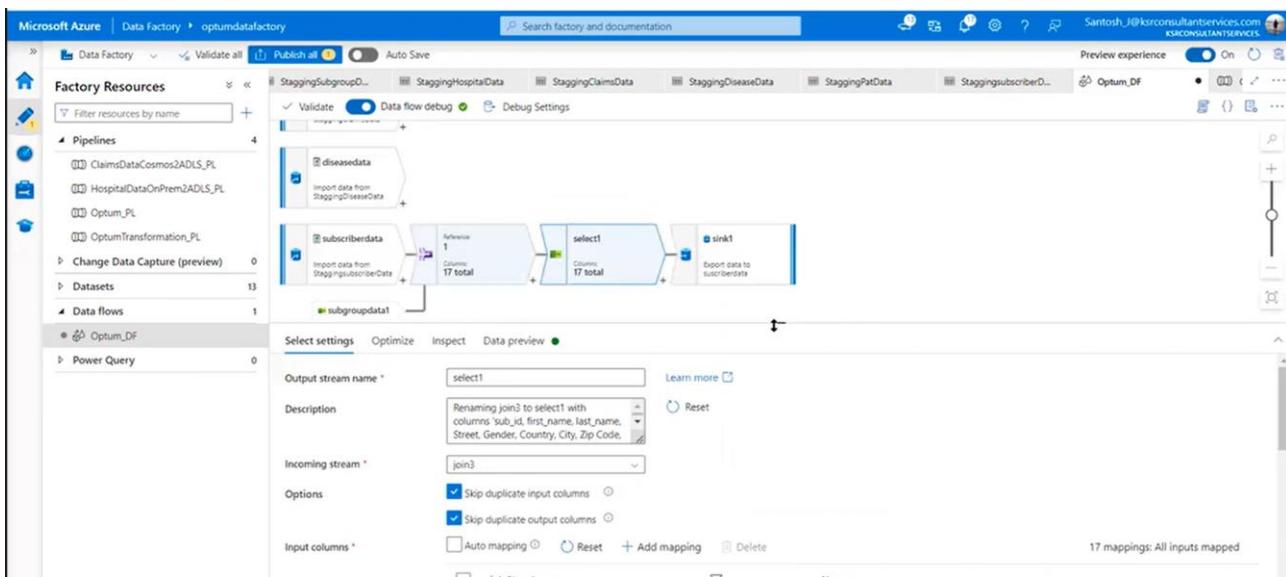
Preview experience On

**Factory Resources**

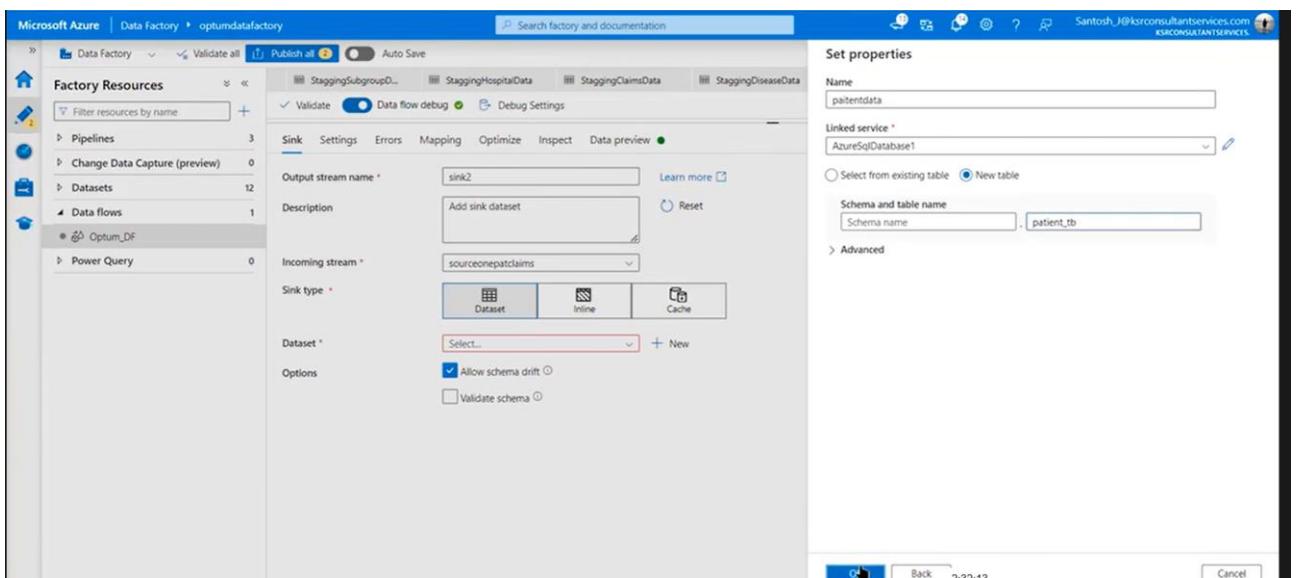
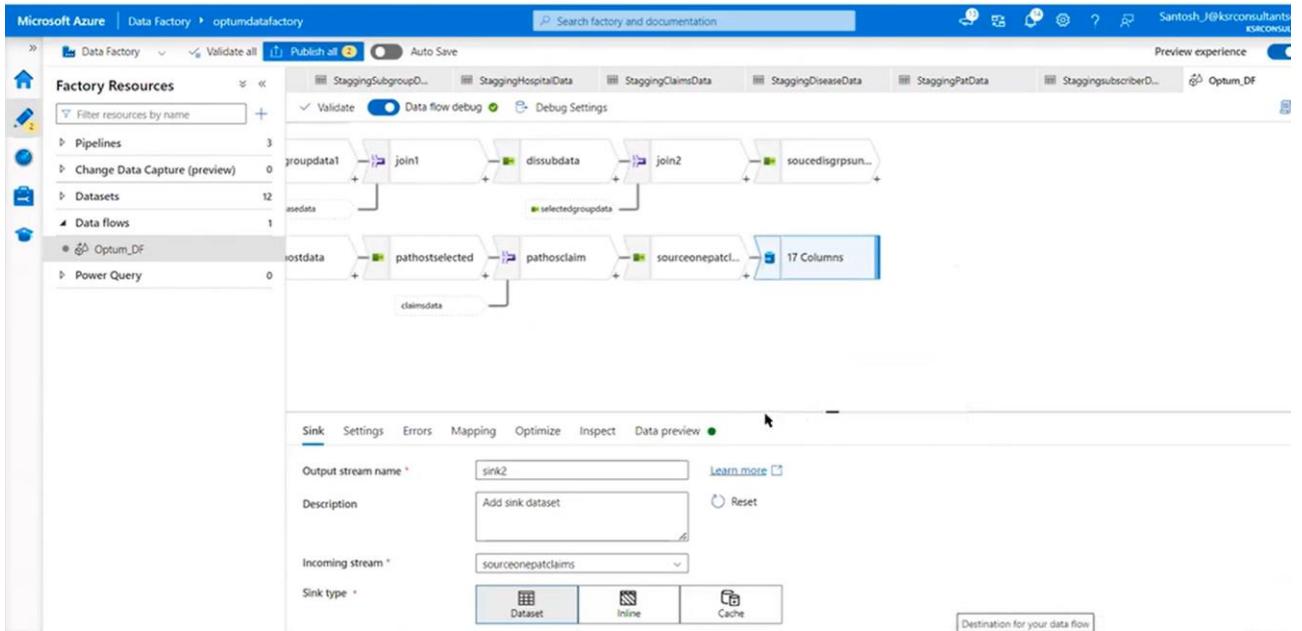
- Pipelines 4
- ClaimsDataCosmos2ADLS\_PL
- HospitalDataOnPrem2ADLS\_PL
- Optum\_PL
- OptumTransformation\_PL
- Change Data Capture (preview) 0
- Datasets 13
- Data flows 1
- Optum\_DF
- Power Query 0

**Select settings**

join3's column	Name as
patient_name	first_name
patient_name	last_name
Street	Street
Gender	Gender
Country	Country
City	City
Zip_Code	Zip_Code
subscribersubgrp_id	Subgrp_id
Elig_IND	Elig_IND
eff_date	eff_date
term_date	term_date
subscriber_age	subscriber_age
subgroupdata1@subgrp_id	subgrp_id
subgrp_name	subgrp_name
monthly_premium	monthly_premium

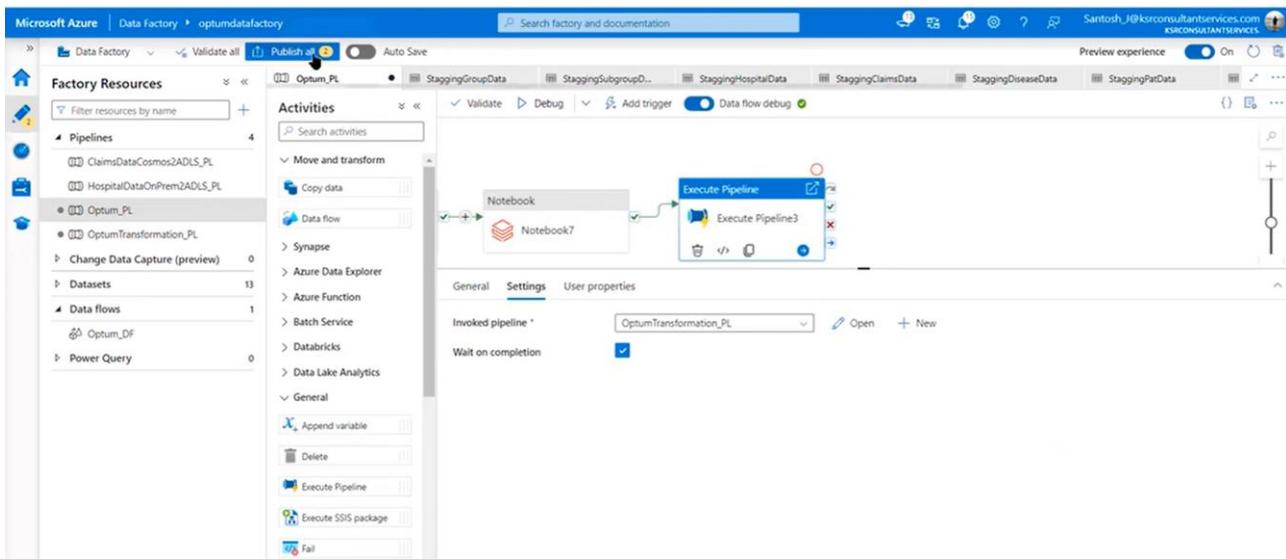


**NOW ADD THE ANOTEHR SINK**

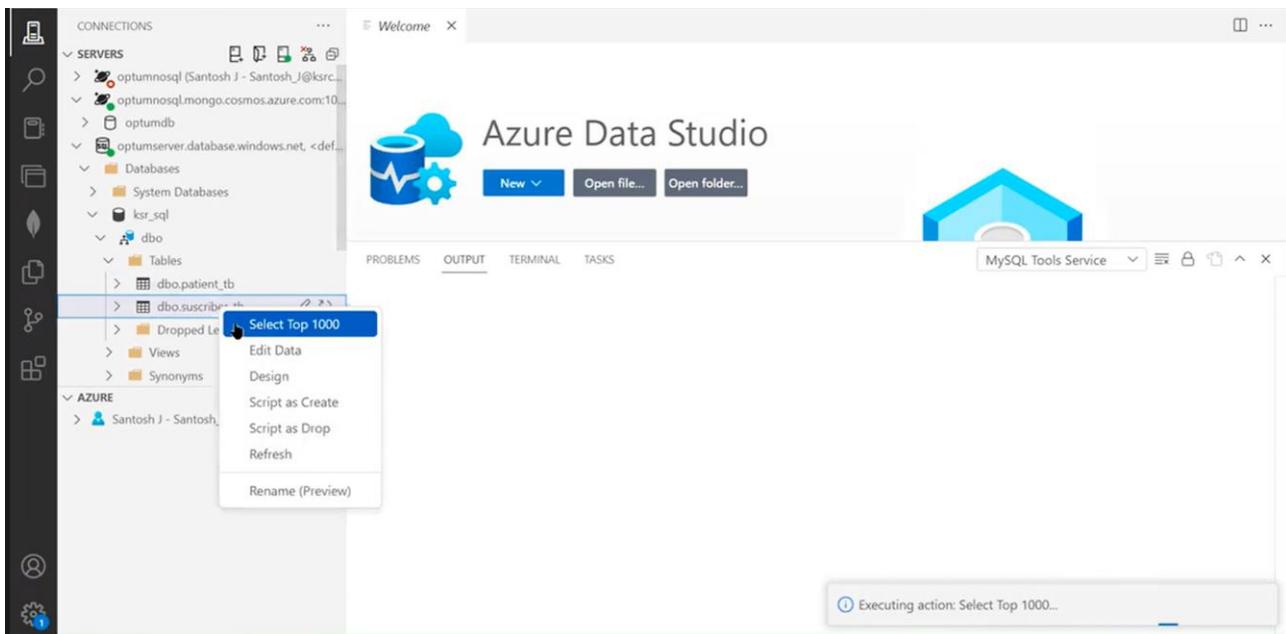


## CREATE AN ANOTHER PIPELINE AS OPTUM\_TRANSFORMATION\_PL

## GO TO MAINPIPELINE AND USE EXECUTE PIPELINE



## LETS GO AND EXECUTE



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer displays a tree view of servers, databases, and tables. The central pane shows a query window with the following SQL code:

```

1  SELECT TOP (1000) [sub_id]
2      ,[first_name]
3      ,[last_name]

```

The results grid shows data from a table with columns: Zip Code, Subgrp\_id, Elig\_Ind, eff\_date, term\_date, subscriber\_age, subgrp\_name, and monthly\_premium. The data includes rows for various zip codes, subgroup IDs, and disease categories like Infectious disease and Allergies.

## IN THIE WE NEED TO SOLVE THIS QUESITON IN THD SQL DATA BASE

### Use Cases?

- # which disease having maximum number of claims.
- # Find those Subscribers having age less than 30 and they subscribe any subgroup
- # Find out which group has maximum subgroups.
- # Find out hospital which serve most number of patients
- # Find out which subgroups subscribe most number of times
- # Find out total number of claims which were rejected
- # From where most claims are coming (city)
- # Which groups of policies subscriber subscribe mostly Government or private
- # Average monthly premium subscriber pay to insurance company.
- # Find out Which group is most profitable
- # List all the patients below age of 18 who admit for cancer
- # List patients who have cashless insurance and have total charges greater than or equal for Rs. 50,000.
- # List female patients over the age of 40 that have undergone knee surgery in the past year