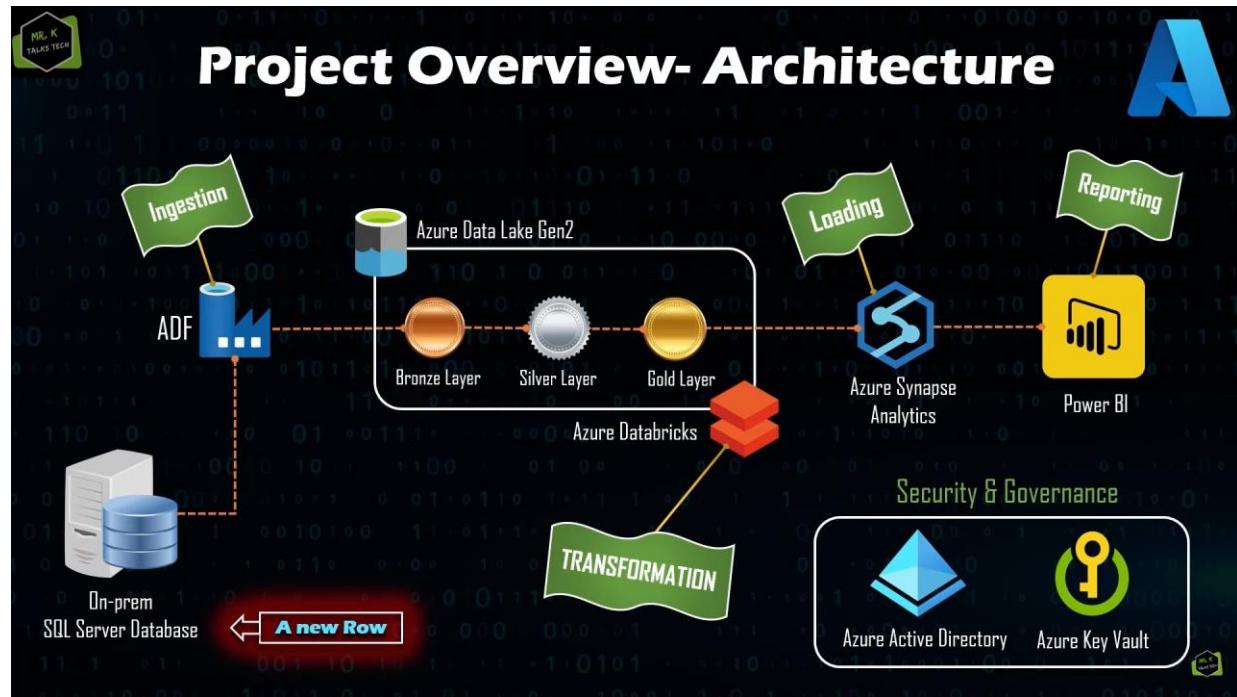


The tools that are used in this project are

1. Azure Data Factory
2. Azure Data Lake Storage Gen2
3. Azure Databricks
4. Azure Synapse Analytics
5. Azure Key vault
6. Azure Active Directory (AAD) and
7. Microsoft Power BI

The use case for this project is building an end to end solution by ingesting the tables from on-premise SQL Server database using Azure Data Factory and then store the data in Azure Data Lake. Then Azure databricks is used to transform the RAW data to the most cleanest form of data and then we are using Azure Synapse Analytics to load the clean data and finally using Microsoft Power BI to integrate with Azure synapse analytics to build an interactive dashboard. Also, we are using Azure Active Directory (AAD) and Azure Key Vault for the monitoring and governance purpose.



# Section I-Environment Setup

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes tabs for 'Azure for College Students—Offer' and 'rg-data-engineer-project - Microsoft Azure'. The main title is 'rg-data-engineer-project' under 'Resource groups'. The left sidebar lists several resource groups: 'databricks-rg-adb-vcg-project-b3c', 'NetworkWatcherRG', 'rg', 'rg-data-engineer-project' (which is selected), and 'synapsworkspace-managedrg-70'. The right pane displays the 'Overview' section for 'rg-data-engineer-project'. It shows basic information like 'Subscription (move) Azure for Students', 'Subscription ID 0e8fa421-bf62-4ff4-ab63-bb983feaaa4c', and 'Location Central India'. Below this is a 'Resources' section with a table listing five resources: 'adb-vcg-project' (Azure Databricks Service, Central India), 'df-vcg-project' (Data factory (V2), Central India), 'ky-vcg-project' (Key vault, Central India), 'newstoragegevcfgforlake' (Storage account, Central India), and 'sgvcgproject' (Storage account, Central India). A 'Give feedback' link is at the bottom right.

So there is no connection bw azure and this database which is stored in my local machine. And we are interested only in transporting only the SalesLT files not the dbo files.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads 'Microsoft SQL Server Management Studio'. The 'File' menu is open. The 'Object Explorer' pane on the left shows a tree view of the 'AdventureworksLT2017' database, including 'Tables', 'Views', 'External Resources', 'Synonyms', 'Programmability', 'Service Broker', 'Storage', and 'Security' nodes. The main central area is currently empty, indicating no results for the current query or task.

Making credentials for us to access the DB in further use.and giving the user vcg the db reader aces from security in DB and by clicking o vcg right click properties ->memberships->dbreader->ok

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, the database 'AdventureWorksLT2017' is selected. In the center Query Editor window, the following T-SQL commands are run:

```
create login vcg with PASSWORD = 'VishalChavan0711';
create user vcg for login vcg;
```

The results pane shows the execution completed successfully with the message "Commands completed successfully." and a completion time of "2024-10-08T16:12:15.4480604+05:30". At the bottom status bar, it says "Query executed successfully." and "0 rows".

In key vaults assign urself Key Vault Administrator Role/Permission to Genereate Secrets from keyvault.(problem resolved here self google)

The screenshot shows the Azure Key Vault interface for the project 'ky1-vcg-project'. On the left, a sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Access policies, Events, Objects, Keys, Certificates, Settings, Monitoring, Automation, and Help. The 'Secrets' option is currently selected. The main pane displays a table of secrets:

Name	Type	Status	Expiration date
password		Enabled	
username		Enabled	

A message at the top of the main pane states: 'The secret 'password' has been successfully created.'

## Section II – Data Ingestion

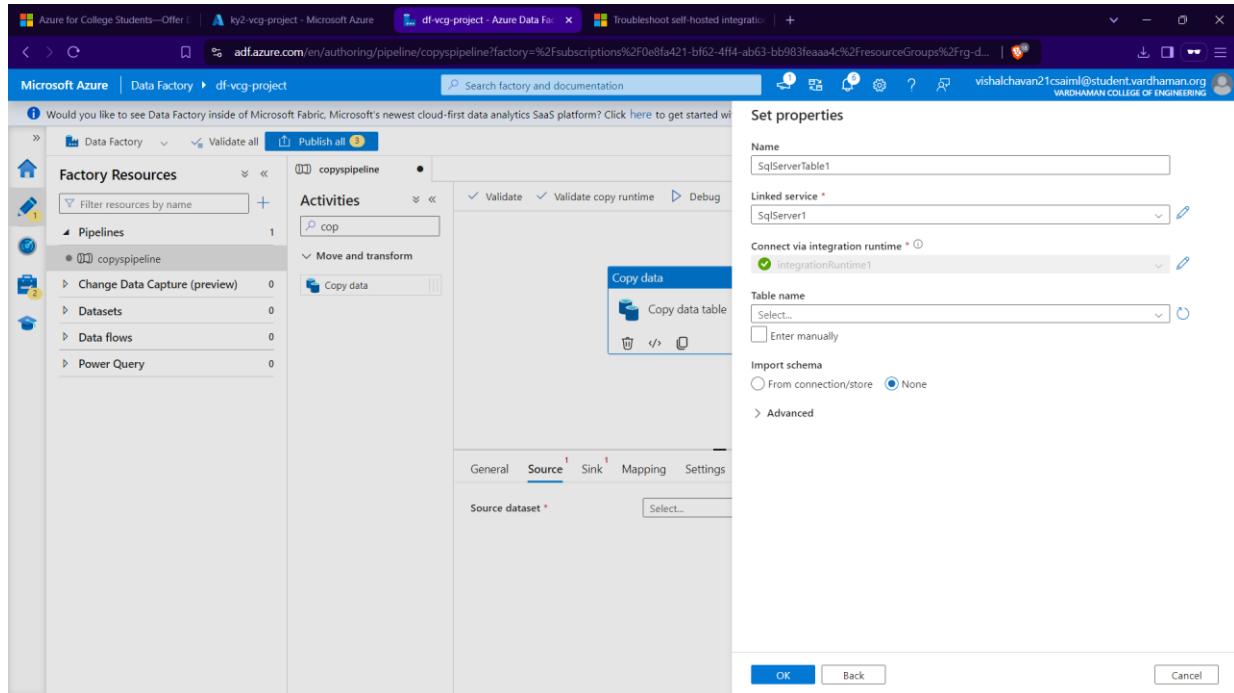
IN ADF go to Manage Integration Runtime Create Self hosted 2 times

The screenshot shows the Microsoft Integration Runtime Configuration Manager for the Data Factory 'df-vcg-project'. The left sidebar includes Home, Author, Monitor, Manage (which is selected), and Learning Center. The main pane displays the 'Home' tab of the configuration manager. It shows the following details:

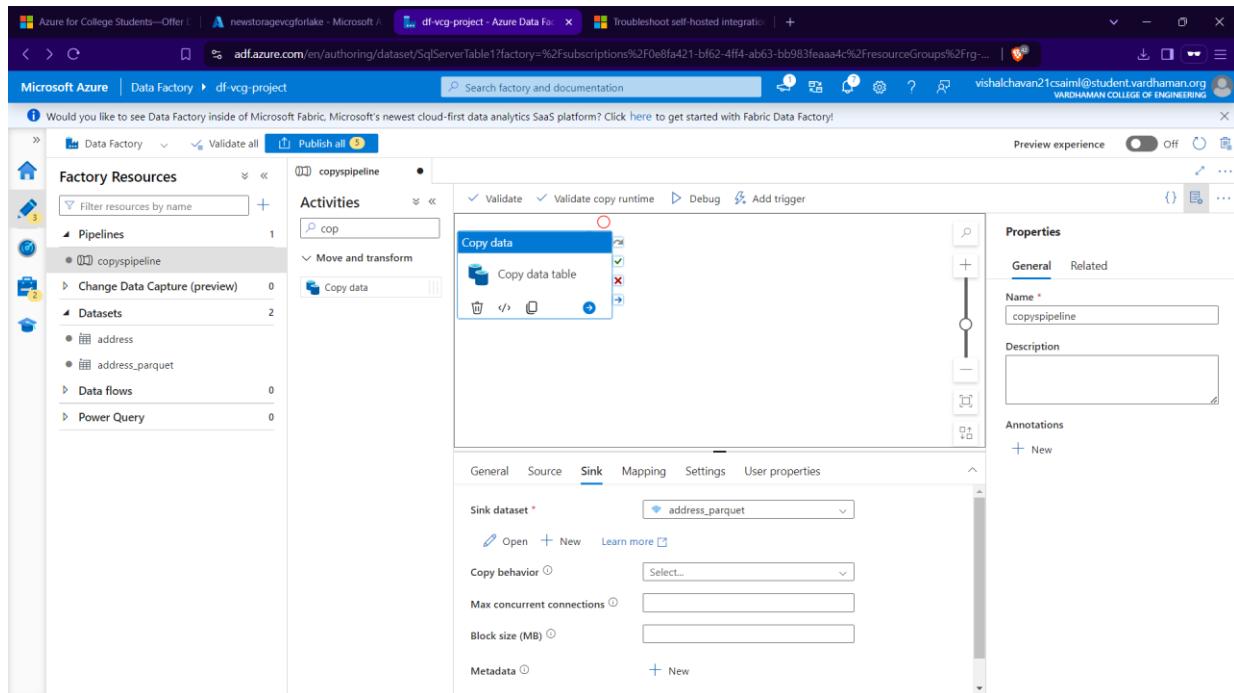
- Data Factory: df-vcg-project
- Integration Runtime: SHIR
- Node: LAPTOP-91NR70N7

A prominent green checkmark indicates that a self-hosted node is connected to the cloud service. Below this, there's a section for 'Data Source Credential' with fields for Credential store (On-premises), Credential status (In sync), and Last backup time (N/A). Buttons for 'Generate Backup' and 'Import Backup' are present. A status bar at the bottom right shows 'Connected to the cloud service (Data Factory V2)'.

Go to author create a pipe line go to copy data in activites and then sourse and add linked service using keyvault and self hosted integration runtime and select table name



Go to sink for azure datalake and parquet format of data with Auto IR unlike for source ie Self hosted



Now to test the pipe line we can use Debug and trigger option here we use debug

The screenshot shows the Azure Data Factory pipeline run details. The pipeline run ID is 34a5e69e-2930-4a7e-8196-d5b8b4808ca7. The Pipeline status is Succeeded. The output table shows one item: Copy data table, Activity status: Succeeded, Activity type: Copy data, Run start: 10/8/2024, 8:12:05 PM, Duration: 47s, Integration runtime: integrationRuntime1.

After debug we go to the container specified fot datalake and hit refresh to get the address table in container

The screenshot shows the Azure Storage Container Overview page for userfilesysvcg. The container contains one blob named SalesLT.Address.parquet, which was modified on 10/8/2024, 8:12:50 PM, has a hot access tier, is a block blob, and is available.

Publish all the changes in ADF->pipeline

Till above we saw that one pipeline was used to copy one file from onpremise to the Cloud now we will make the whole dataset to be copied in one pipeline itself.

Drag the loop up activity in new pipeline and drag and drop it.

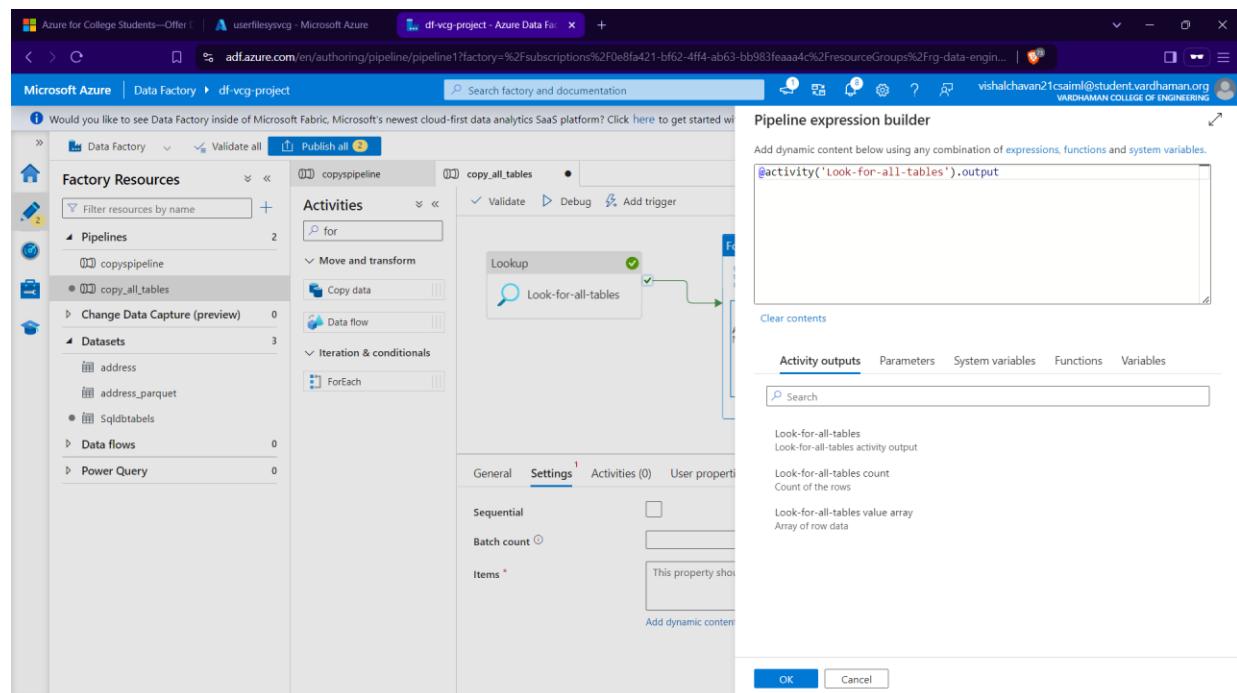
After managing setting don't select any tables and create then in settings select query as give this:

select

```
s.name as SchemaName,  
t.name as TableName  
from sys.tables t  
inner join sys.schemas s  
on t.schema_id=s.schema_id  
where s.name='SalesLT'
```

then click on preview data to make sure if its working fine(listing all tables) and debug it

now D&D from activities :"For each"



Then in foreach activity go again d&d copy data and perform the source with query as:

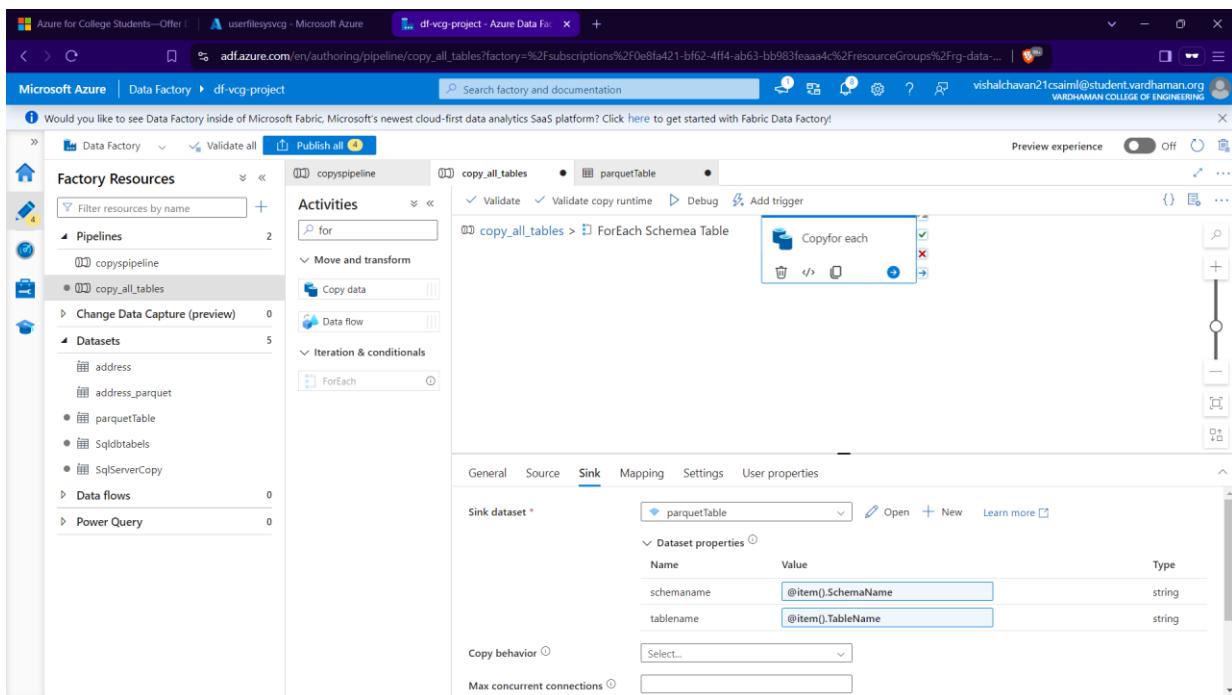
```
@{concat('SELECT * FROM ',item().SchemaName,'.',item().TableName)}
```

And sink as normal by datalake.

We want the structure as:

Container/Schema/tableName/TableName.parquet

After adding 2 parameters in parquetTable



In File name in parquetTable->Connection

For dir name :

```
@{concat(dataset().schemaname,'/',$dataset().tablename)}
```

For file name:

```
@}concat(dataset().tablename,'.parquet')}
```

The screenshot shows the Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, and other components. The 'copypipeline' pipeline is selected. Inside the pipeline editor, the 'copy\_all\_tables' activity is being configured. The 'Connection' tab is active, showing a linked service 'AzureDataLakeStorage1' and a file path template. The 'Schema' and 'Parameters' tabs are also visible.

## Now Publish

For this lets explore Add Trigger Option->TriggerNow->ok

The screenshot shows the 'All pipeline runs' page for the 'copy\_all\_tables' pipeline. The pipeline run status is 'Succeeded'. The table below details the individual activities and their execution results.

Activity	Status	Duration	Integration Runtime	Run ID
Look-for-all-tables	Succeeded	23s	integrationRuntime1	b767c1c6-278c-4c1c-4...
ForEach Schema T...	Succeeded	2m 26s		ab58cb64-a54c-4911-4...
Copyfor each	Succeeded	2m 23s	integrationRuntime1	42dc2b58-0157-4918-4...
Copyfor each	Succeeded	2m 0s	integrationRuntime1	4c878cae-388b-4907-4...
Copyfor each	Succeeded	39s	integrationRuntime1	bc285cb7-bce8-49e4-4...
Copyfor each	Succeeded	19s	integrationRuntime1	8d975cf2-8076-450e-4...
Copyfor each	Succeeded	21s	integrationRuntime1	4cdd91ca-4d79-4e6a-4...
Copyfor each	Succeeded	19s	integrationRuntime1	6204b1a7-84dc-49aa-4...
Copyfor each	Succeeded	48s	integrationRuntime1	96347531-e9cc-4d5f-4...

The screenshot shows the Azure Storage Explorer interface for the 'userfilesysvcg' container. The 'Overview' tab is selected. The container has 12 blobs listed:

- [-]
- Address
- Customer
- CustomerAddress
- Product
- ProductCategory
- ProductDescription
- ProductModel
- ProductModelProductDescription
- SalesOrderDetail
- SalesOrderHeader

Till here we did the Data Ingestion Process and have Completed till Bronze Layer.

## Section III: Data Transformation we use DataBricks

The screenshot shows the Databricks workspace. The left sidebar lists various workspace components:

- 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
- Playground
- Experiments
- Features
- Models

The main area is titled "Welcome to Databricks" and features three prominent cards:

- Set up your workspace**: Follow this step-by-step guide that walks you through setting up the workspace for your new Databricks account. [Get started](#)
- Extract, transform and load (ETL) data for analytics**: Start with a raw sample data set, clean and transform it for analysis, automate your data processing then build and share a dashboard that refreshes automatically. [Start transforming](#)
- Build and train your first LLM chatbot**: Create an LLM-based chatbot, then improve its responses by providing it some sample text. [Start building](#)

At the bottom, there's a "Start your journey" section with a note: "Try the 'New' menu, where you can upload or connect to data and then explore it in a notebook or dashboard." A blue "+ New" button is located at the bottom right.

We create a cluster in databricks with enabling credentials in advance option

The screenshot shows the 'Cluster Details - Databricks' page in a browser. The cluster name is 'data\_transformation'. The 'Configuration' tab is selected. Key settings include:

- Policy:** Unrestricted
- Access mode:** Single user access (Single user: 21881A66C7)
- Databricks Runtime Version:** 15.4 LTS (includes Apache Spark 3.5.0, Scala 2.12)
- Node type:** Standard\_D4ds\_v5 (16 GB Memory, 4 Cores)
- Tags:** No custom tags
- Advanced options:** Use Photon Acceleration (checked), Terminate after 15 minutes of inactivity (checked)

A summary panel on the right indicates 1 Driver, 16 GB Memory, 4 Cores, Runtime 15.4x-scala2.12, and Photon Standard\_D4ds\_v5 2 DBU/h.

Created a notebook In workspace and update souce and mount\_point

The screenshot shows a Databricks workspace notebook titled 'storage\_mount' in Python mode. The code cell contains the following Python code:

```
configs = {
    "fs.azure.account.auth.type": "CustomAccessToken",
    "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")
}

# Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
    source = "abfs://userfilesysvcg@newstoragevcgforlake.dfs.core.windows.net/",
    mount_point = "/mnt/userfilesysvcg",
    extra_configs = configs)
```

A tooltip at the bottom of the code cell provides keyboard shortcuts: [Shift+Enter] to run and move to next cell, [Ctrl+Shift+P] to open the command palette, and [Esc H] to see all keyboard shortcuts.

A 'Try out the Command Palette!' modal is visible in the bottom right corner, explaining how to quickly access actions without leaving the keyboard.

To see the files in the given path

The screenshot shows a Databricks notebook interface. The left sidebar contains a navigation menu with sections like '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', 'Playground', 'Experiments', 'Features', 'Models', and 'Samples'. The main area has tabs for 'storage\_mount' (selected), 'Python', and 'Star'. A search bar at the top right says 'Search data, notebooks, recents, and more...'. Below the search bar are buttons for 'Run all', 'data\_transformatio...', 'Schedule', and 'Share'. The notebook content shows two code cells:

```
1
configs = {
    "fs.azure.account.auth.type": "CustomAccessToken",
    "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")
}

# Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
    source = "abfs://userfilesysvcg@mnewstoragevcgforlake.dfs.core.windows.net/",
    mount_point = "/mnt/userfilesysvcg",
    extra_configs = configs)

Out[1]: True

2
dbutils.fs.ls("/mnt/userfilesysvcg/SalesLT")

Out[4]: [FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/Address/', name='Address', size=0, modificationTime=1728402277000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/Customer/', name='Customer', size=0, modificationTime=1728402245000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/CustomerAddress/', name='CustomerAddress', size=0, modificationTime=1728402227000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/Product', name='Product', size=0, modificationTime=1728402349000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/ProductCategory', name='ProductCategory', size=0, modificationTime=1728402245000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/ProductDescription', name='ProductDescription', size=0, modificationTime=1728402325000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/ProductModel', name='ProductModel', size=0, modificationTime=1728402324000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/ProductModelProductDescription', name='ProductModelProductDescription', size=0, modificationTime=1728402225000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/SalesOrderDetail', name='SalesOrderDetail', size=0, modificationTime=1728402224000),
 FileInfo(path='dbfs:/mnt/userfilesysvcg/SalesLT/SalesOrderHeader', name='SalesOrderHeader', size=0, modificationTime=1728402253000)]
```

As for Data Transformation we can change try changing the columns data type in the data base in SSMS from datetime to date.

In these notebooks we can also use magic commands by %sql that notebook will be of py but with this it will also be able to run sql commands in the notebook as well

```

PRODUCTSDF=spark.read.parquet('mnt/userfilesysvcg/SalesLT/Products')
df=spark.read.parquet('mnt/userfilesysvcg/SalesLT/SalesOrderDetail')
df=df.withColumn("orderdate",df.orderdate.cast(TimestampType()).cast(DateType()))
df.write.parquet('mnt/userfilesysvcg2/SalesLT/Products')
df.show()

```

With this code we have successfully copied the entire files from ..vcg to vcg2 container/datalake using databrick

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
Address					-	---
Customer					-	---
CustomerAddress					-	---
Product					-	---
ProductCategory					-	---
ProductDescription					-	---
ProductModel					-	---
ProductModelProductDescription					-	---
SalesOrderDetail					-	---
SalesOrderHeader					-	---

This all was Level 1 Transformation ie Bronze -> Silver

Now we do from Silver->Gold here we join tables changes naming conf etc..

```

from 2 - 3 Python
Last edit was now
File Edit View Run Help 11
from 2 - 3 - Databricks from 1 - 2 - Databricks + 
adb-138752369830465.azuredatabricks.net/editor/notebooks/228545234577330?o=138752369830465#command/228545234577342 | 
Search data, notebooks, recents, and more... CTRL + P
Run all data_transformation Schedule Share
+ 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 Playground Experiments Features Models
from 2 - 3 (Python)
from 2 - 3 (Python) Last edit was now
11
for name in table_name:
    path='/mnt/userfilesysvcg2/SalesLT/*name
    print(path)
    df=spark.read.format("delta").load(path)
    #get list of col names
    column_names=df.columns

    for old_col_name in column_names:
        #convert col name from colname to col_name format
        new_col_name="".join(["_" if char.isupper() and not old_col_name[i-1].isupper() else char for i,char in enumerate(old_col_name)])
        lstrip("_")
        #change col name using withColumnRenamed and regexp_replace
        df=df.withColumnRenamed(old_col_name,new_col_name)

    output_path='/mnt/userfilesysvcg3/SalesLT/*name'
    df.write.format("delta").mode("overwrite").save(output_path)

(30) Spark jobs
dt: pyspark.sql.dataframe.DataFrame = [Sales_Order_ID: integer, Revision_Number: integer ... 20 more fields]
/mnt/userfilesysvcg2/SalesLT/Address
/mnt/userfilesysvcg2/SalesLT/Customer
/mnt/userfilesysvcg2/SalesLT/CustomerAddress
/mnt/userfilesysvcg2/SalesLT/Product
/mnt/userfilesysvcg2/SalesLT/ProductCategory
/mnt/userfilesysvcg2/SalesLT/ProductDescription
/mnt/userfilesysvcg2/SalesLT/ProductModel
/mnt/userfilesysvcg2/SalesLT/ProductModelProductDescription
/mnt/userfilesysvcg2/SalesLT/SalesOrderDetail
/mnt/userfilesysvcg2/SalesLT/SalesOrderHeader

```

With this we have changed the col names from vcg2 to vcg3 all names having "\_" in b/w them.

	Sales_Order_ID	Revision_Number	Order_Date	Due_Date	Ship_Date	Status	Online_Order_Flag
9	71815	2	2008-06-01	2008-06-13	2008-06-08	5	false
10	71816	2	2008-06-01	2008-06-13	2008-06-08	5	false
11	71831	2	2008-06-01	2008-06-13	2008-06-08	5	false
12	71832	2	2008-06-01	2008-06-13	2008-06-08	5	false
13	71845	2	2008-06-01	2008-06-13	2008-06-08	5	false
14	71846	2	2008-06-01	2008-06-13	2008-06-08	5	false
15	71856	2	2008-06-01	2008-06-13	2008-06-08	5	false
16	71858	2	2008-06-01	2008-06-13	2008-06-08	5	false
17	71863	2	2008-06-01	2008-06-13	2008-06-08	5	false
18	71867	2	2008-06-01	2008-06-13	2008-06-08	5	false
19	71885	2	2008-06-01	2008-06-13	2008-06-08	5	false
20	71895	2	2008-06-01	2008-06-13	2008-06-08	5	false
21	71897	2	2008-06-01	2008-06-13	2008-06-08	5	false
22	71898	2	2008-06-01	2008-06-13	2008-06-08	5	false

32 rows | 0.26 seconds runtime      Refreshed 1 minute ago

Now we go to ADF and make pipeline for level 1 – 2 and 2 – 3 transformation

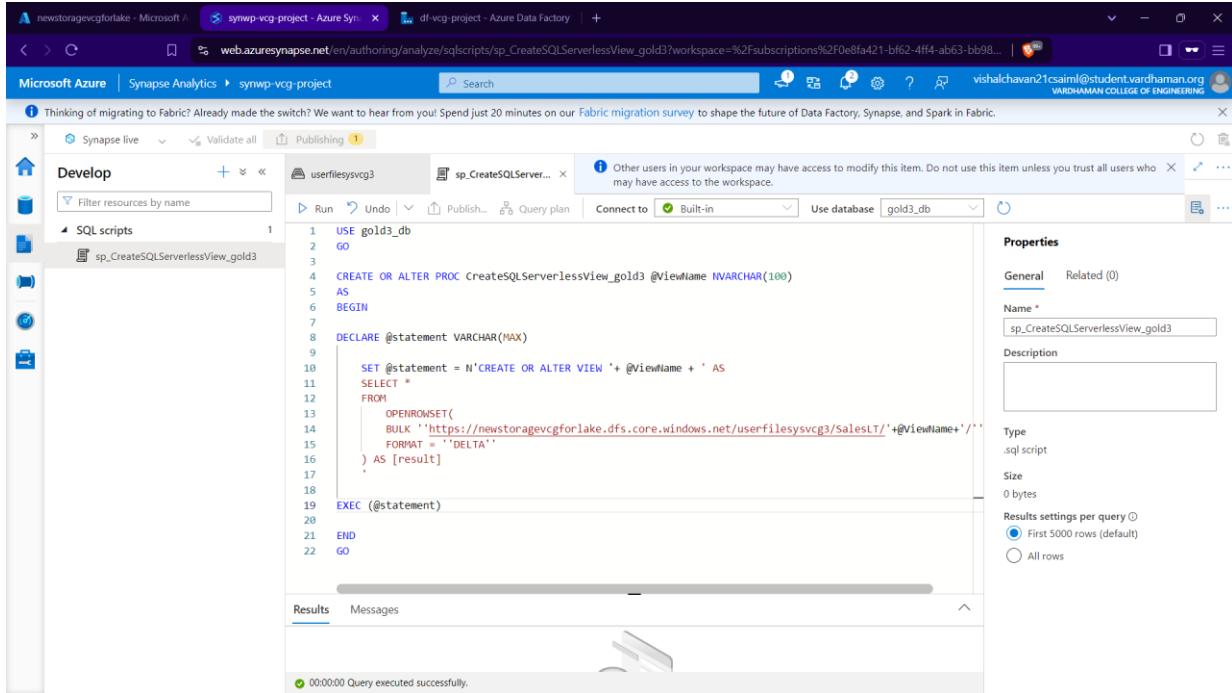
Create a linked service in ADF and go to compute->azuredatabricks

Now we create 2 notebook with path given to specific notebooks from 1 – 2 and 2 – 3 and publishing the changes

# Section IV – Data Loading

Now we have vlean/transformed data now we need to load in synapse analytics to load

Created a Database named gold3\_db and made a sql query to show the rows in the dataset in our device

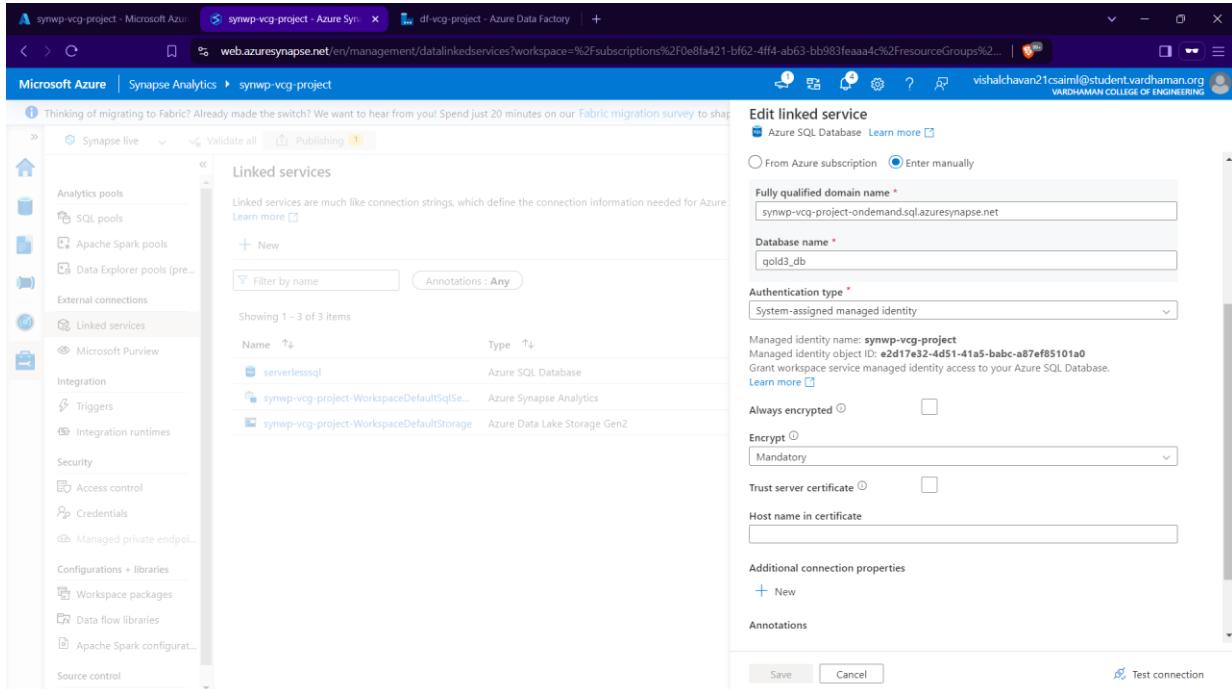


The screenshot shows the Microsoft Azure Synapse Analytics studio interface. The left sidebar has a 'Develop' tab selected, showing a list of 'SQL scripts'. One script is open, titled 'sp\_CreateSQLServerlessView\_gold3'. The code is as follows:

```
1 USE gold3_db
2 GO
3
4 CREATE OR ALTER PROC CreateSQLServerlessView_gold3 @viewName NVARCHAR(100)
5 AS
6 BEGIN
7
8    DECLARE @statement VARCHAR(MAX)
9
10   SET @statement = N'CREATE OR ALTER VIEW ' + @viewName + ' AS
11      SELECT *
12      FROM
13        OPENROWSET(
14          BULK ''https://newstoragevcgforlake.dfs.core.windows.net/userfilesysvcg3/SalesLT/' + @viewName + ''
15          FORMAT = ''DELTA'',
16        ) AS [result]
17
18   EXEC (@statement)
19
20
21 END
22 GO
```

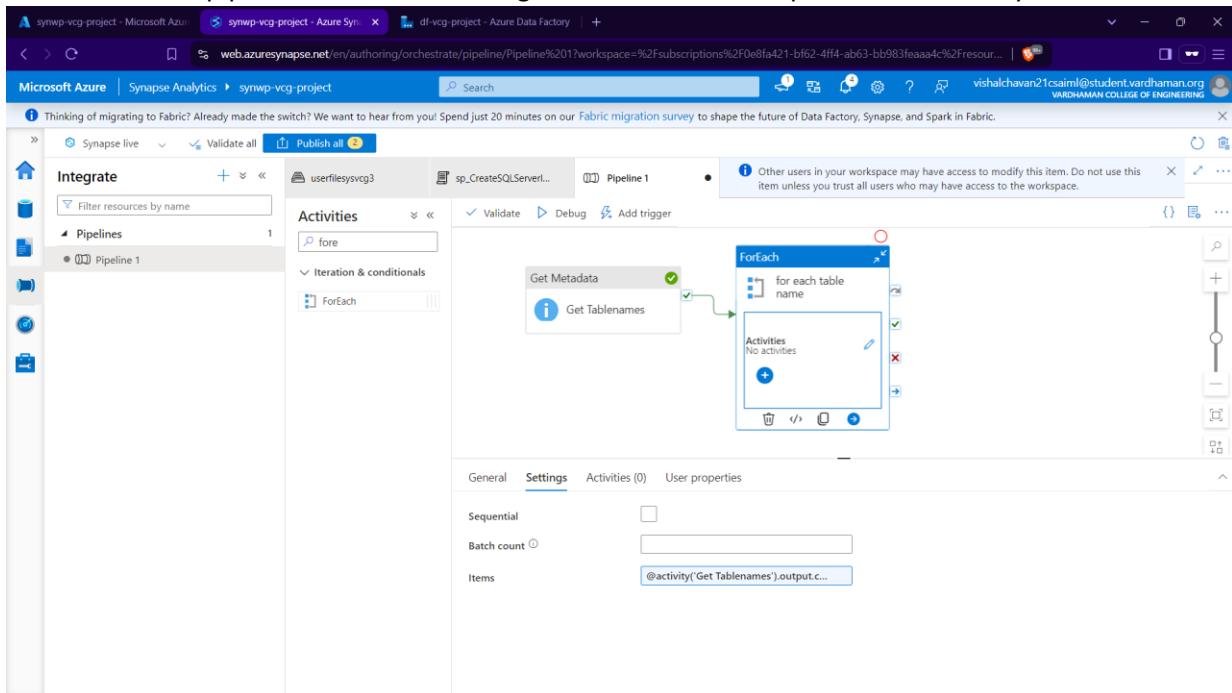
The right panel shows the 'Properties' for the stored procedure, with 'Name' set to 'sp\_CreateSQLServerlessView\_gold3'. The 'Results' tab at the bottom indicates the query was executed successfully.

then we made a linked service

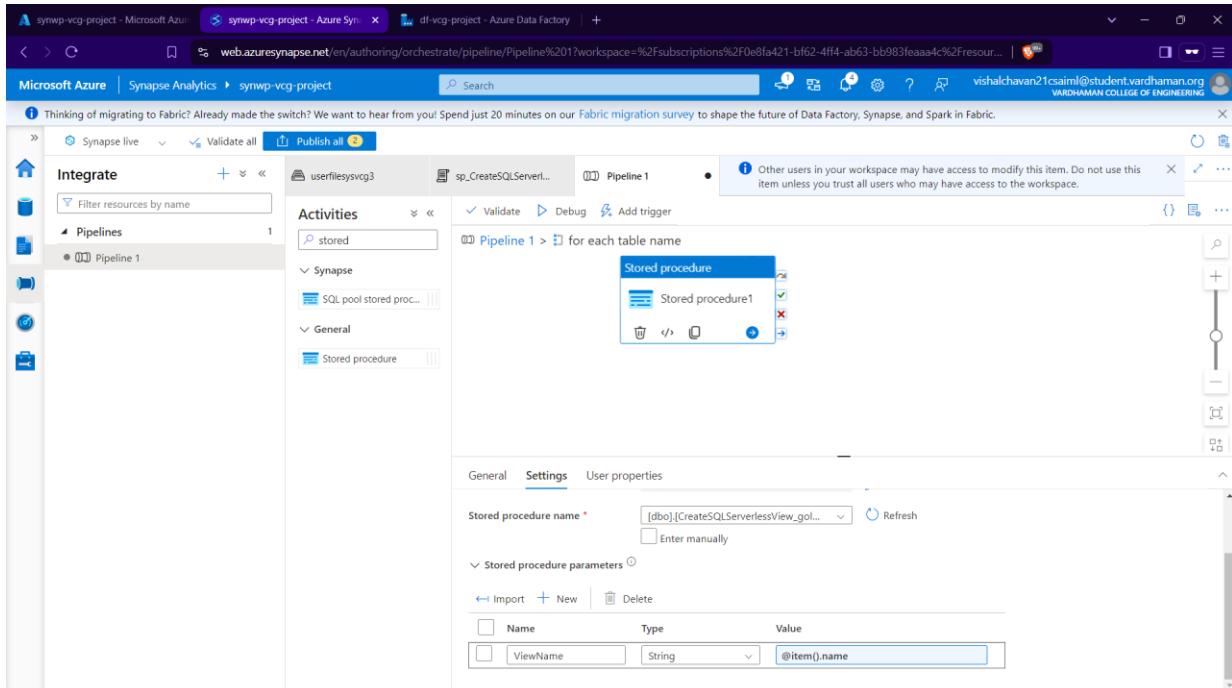


The screenshot shows the 'Linked services' blade in the Microsoft Azure Synapse Analytics studio. The left sidebar has a 'Linked services' item selected. The main area shows a table of existing linked services and a 'New' button. A modal window titled 'Edit linked service' is open, showing the configuration for a new Azure SQL Database linked service. The 'Name' is 'serverlessql', 'Type' is 'Azure SQL Database', 'Fully qualified domain name' is 'synwp-vcg-project-ondemand.sql.azuresynapse.net', 'Database name' is 'gold3\_db', and 'Authentication type' is 'System-assigned managed identity'. Other options like 'Always encrypted' and 'Encrypt' are also visible.

Then we make a pipe line where metadata gives tablename as input for each activity



Inside activity for 'for each' it gets the tablenames Views



Then after running the pipeline when we refresh the database in view section the view of the top rows are now available for us to view. for which we can create a new sql script to view.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, the 'Data' section displays a tree view of a SQL database named 'gold3\_db' containing various tables like 'dbo.address', 'dbo.Customer', etc. In the center, a pipeline named 'sp\_CreateSQLServer...' is being edited. The pipeline consists of two main activities: 'Get Metadata' and 'ForEach'. The 'ForEach' activity is configured to iterate over 'for each table name' and contains an 'Activities' step with a 'Stored procedure...' option. Below the pipeline editor, the 'Pipeline status' is shown as 'Succeeded'. A table at the bottom provides details for the single run item, including activity names, status, type, run start time, duration, and integration runtime.

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime
Get Tablenames	Succeeded	Get Metadata	10/9/2024, 9:32:49 PM	21s	AutoResolveIntegratio

## Section V – Data Reporting

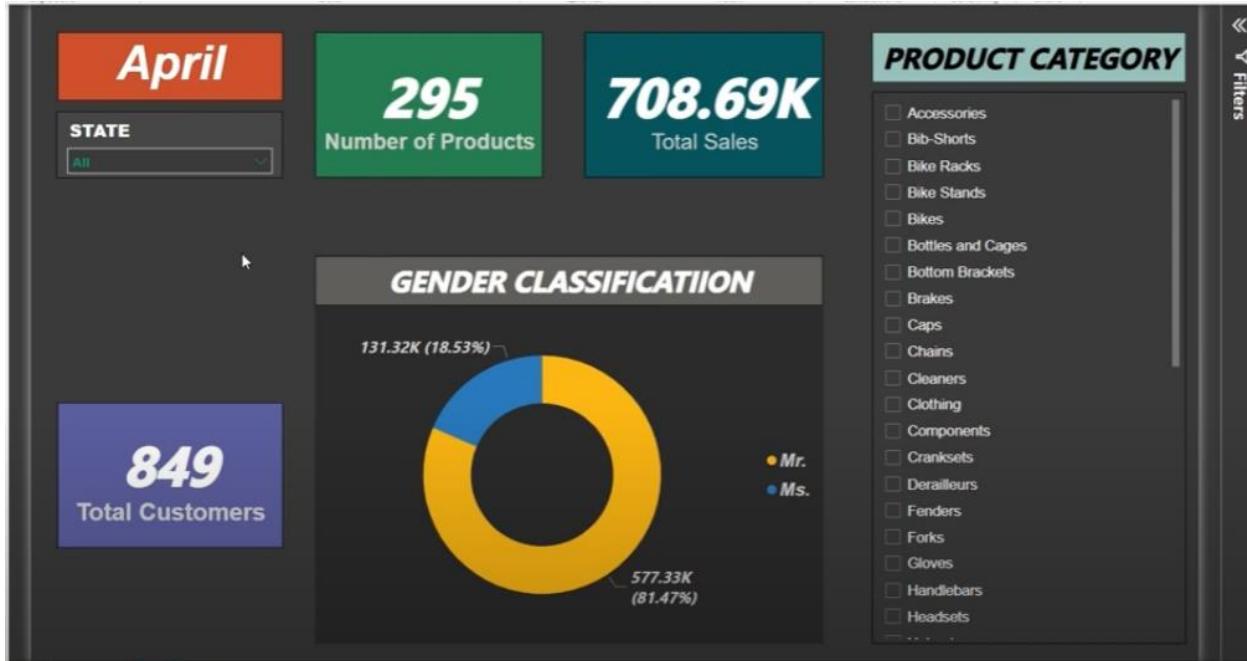
Now we use Power BI to fetch the views of the database.

Go to GetData->More->Azure->synaptic analytics->connect

Give the serverless endpoint URL to the pop up window and click ok then go to microsoft account as security and connect.

Select all tables and load

In Model View Section by Manage Relationship we can mangae them accordingly. and proceed with the reporting



## Section VI - Security and Governance (AAD)

As I am the sole owner of this only I as a user can access it but to add someone else is really annoying like going to resource grp and giving access control etc we can make security grp and anyone on security grp can use the resource.

Go to Microsoft Entra ID and in groups

The screenshot shows the Microsoft Azure Active Directory Overview page for 'Vardhaman College of Engineering'. The left sidebar contains navigation links for Overview, Preview features, Diagnose and solve problems, Manage (Users, Groups, External identities, Roles and administrators, Administrative units, Delegated admin partners, Enterprise applications, Devices, App registrations, Identity Governance, Application proxy, Custom security attributes), and Licenses. The main content area displays basic information about the tenant, including Name (Vardhaman College of Engineering), Tenant ID (084a029e-1435-40bc-8201-87ec1b251fb3), Primary domain (vardhaman.org), License (Microsoft Entra ID Basic for EDU), and user statistics (17,516 users, 3,954 groups, 220 applications, 4,683 devices). A warning message about service changes to Microsoft Entra Connect is present. The URL in the address bar is [https://portal.azure.com/#view/Microsoft\\_AAD\\_IAM/ActiveDirectoryMenuBlade/~/Groups](https://portal.azure.com/#view/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/~/Groups).

Give the grp owner and members I have only given owner as myself.

The screenshot shows the 'New Group' creation page in the Microsoft Azure Active Directory. The form fields include: Group type (Security selected), Group name (data-engineer-grp), Group description (Enter a description for the group), Membership type (Assigned), Owners (1 owner selected), and Members (No members selected). At the bottom is a 'Create' button. The URL in the address bar is [https://portal.azure.com/#view/Microsoft\\_AAD\\_IAM/AddGroupBlade](https://portal.azure.com/#view/Microsoft_AAD_IAM/AddGroupBlade).

Go to role assignments in resource grp- privilegd roles->selct members and add grp name

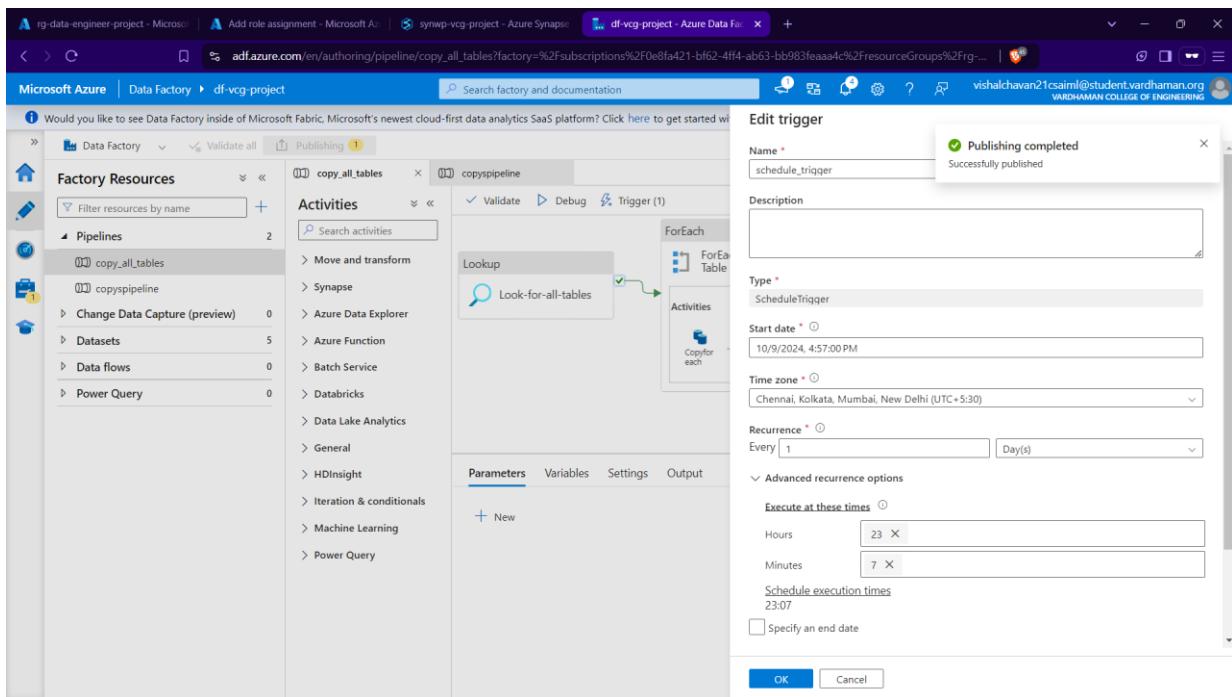
The screenshot shows the 'Add role assignment' interface in the Microsoft Azure portal. The 'Members' tab is active. A 'Contributor' role is selected. The 'Assign access to' section has 'User, group, or service principal' selected. A single member, 'data-engineer-grp', is listed under 'Members'. The 'Description' field is empty. At the bottom, the 'Review + assign' button is highlighted in blue, while 'Previous' and 'Next' buttons are in grey.

When we give access to any other account and in that account we can go to resource grp there and when we refresh the page they have been granted the resource grp which we used.

## Section VII – Pipeline Testing

Here when we add a row in our premise database it should be reflected in the report.

We can do this by creating triggers from ADF



Now before the ‘trigger’ triggers we update the Customer Table by inserting 2 new rows in to it.

```

SQLQuery2.sql - LAPTOP-91NR70N7\SQLPRESS.AdventureWorksLT2017 (LAPTOP-91NR70N7\VishalChavan (63)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
AdventureWorksLT2017 | Execute | New Query | New Table | New Script | New File | New Database | New Log | New Diagram | New View | New System Table | New Filegroup | New Full Text Catalog | New Index | New Constraint | New Trigger | New Column | New Statistics | New Table | New System Table | New File | New External Table | New Graph Table | New Build Version | New Error Log | New Address | New Customer | New Customer Address | New Product | New Product Category | New Product Description | New Product Model | New Product Model | New Sales Order | New Sales Order Header | New View | New External Resource | New Synonym | New Programmability | New Service Broker | New Storage | New Security | New SQL Proj
Object Explorer | LAPTOP-91NR70N7\SalesLT.Customer | SQLQuery3.sql - LA_\VishalChavan (68) | SQLQuery2.sql - LA_\VishalChavan (63)* | SQLQuery1.sql - LA_\VishalChavan (61)
Connect | 
LAPTOP-91NR70N7\SQLPRESS
Database | AdventureWorksLT2017
Tables | SalesLT.Customer
System Tables | 
FileTables | 
External Tables | 
Graph Tables | 
Build Version | 
Error Log | 
Address | 
Customer | 
Customer Address | 
Product | 
Product Category | 
Product Description | 
Product Model | 
Product Model | 
Sales Order | 
Sales Order Header | 
Views | 
External Resources | 
Synonyms | 
Programmability | 
Service Broker | 
Storage | 
Security | 
SQL Proj | 

[LAPTOP-91NR70N7\SalesLT.Customer]
    , [FirstName]
    , [MiddleName]
    , [LastName]
    , [Suffix]
    , [CompanyName]
    , [SalesPerson]
    , [EmailAddress]
    , [Phone]
    , [PasswordHash]
    , [PasswordSalt]
    , [Rowguid]
    , [ModifiedDate]
VALUES
    (595959, 0, 'Mr.', 'Vishal', 'NA', 'Chavan', 'Jr.', 'abc', 'adventure-works/jon', 'xx@gmail.com', '999-999-9999', 'uRlpvZDNNJX91+hTlRK+liT4UKRgWhApJgcMC2d4',
    (696969, 0, 'Mr.', 'Next', 'NA', 'Chavan', 'Jr.', 'als', 'adventure-works/sen', 'xy@gmail.com', '999-666-9669', '26rwIeP9yofo+P6lpdeASp/MAC2/tdKFbmFVC4UBRmA=')

SET IDENTITY_INSERT [AdventureworksLT2017].[SalesLT].[Customer] OFF;

(2 rows affected)
Completion time: 2024-10-09T23:03:01.787560+05:30
Messages
Query executed successfully.

Ln 5 Col 1 Ch 1 INS

```

Then after sucessfully running sql we then monitor the pipeline as it gets triggered at the specified time.

All pipeline runs > copy\_all\_tables - Activity runs

Activity runs

Pipeline run ID: 3fb2248c-7cf0-4d25-862a-1ee89acfdf2e

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID
Look-for-all-tables	Succeeded	Lookup	10/9/2024, 11:07:01 PM	16s	integrationRuntime1		9fb58b31-9ca1-4f3e-a
ForEach Schema Table	Succeeded	ForEach	10/9/2024, 11:07:17 PM	1m 6s			7e602696-6caf-4713-f
Copy for each	Succeeded	Copy data	10/9/2024, 11:07:18 PM	38s	integrationRuntime1		9b5cdc11-4529-4fa3-f

Now we can see that the no of customers have increased by two in the Power BI dashboard.

