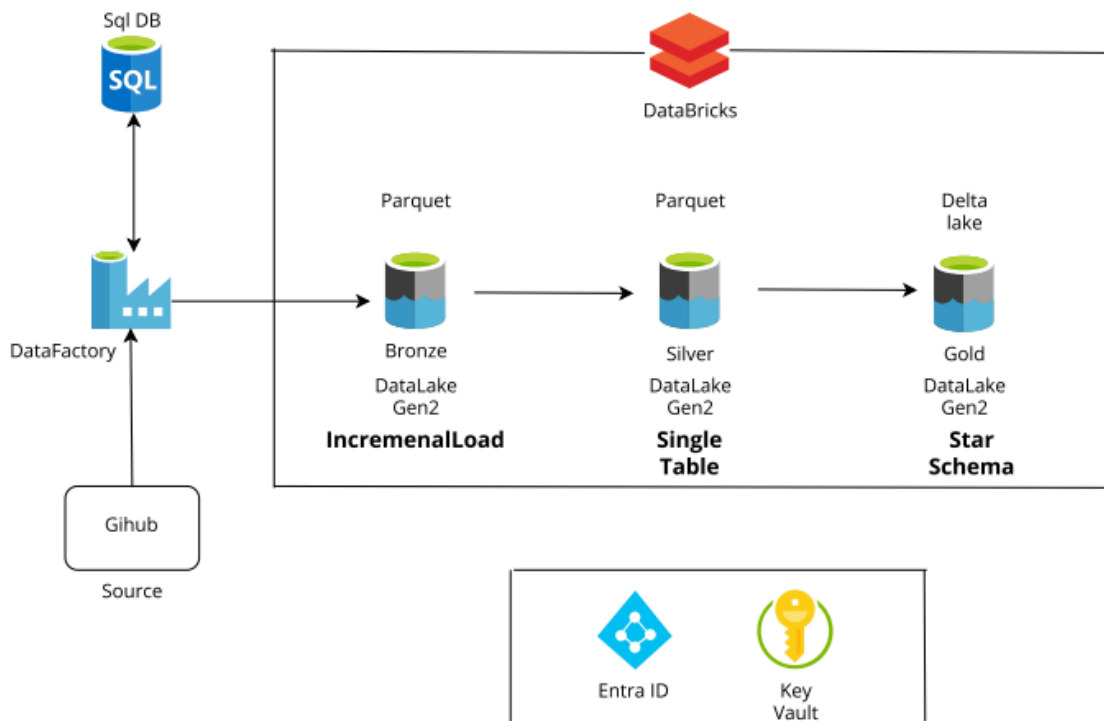


Architecture:



Environment Setup:

1. Create a Resource Group for the Project.
2. Create a Azure Data Factory.
3. Create a Azure Data Lake gen 2 storage account.

Under storage account created bronze, silver and gold container.

Bronze → to store the raw data

Silver → to store the processed data

Gold → to store the cleaned data that connect to power bi for end users

4. Create Azure SQL Database.
5. Create Azure Databricks.
6. Create Azure Key Vault

Dataset required:

Project dataset

PHASE 1:

1. Created a source table in Azure SQL database.

```
create table source_cars_data
(
    Branch_Id varchar(200),
    Dealer_Id varchar(200),
    Model_Id varchar(200),
    Revenue bigint,
    Units_Sold bigint,
    Date_Id varchar(200),
    Day int,
    Month int,
    Year int,
    Branch_Name varchar(200),
    Dealer_Name varchar(200)
)
```

In phase 1, we have created a pipeline to copy the data from [Project dataset](#) to Azure Sql database using Azure Data Factory.

Base URL: <https://raw.githubusercontent.com>

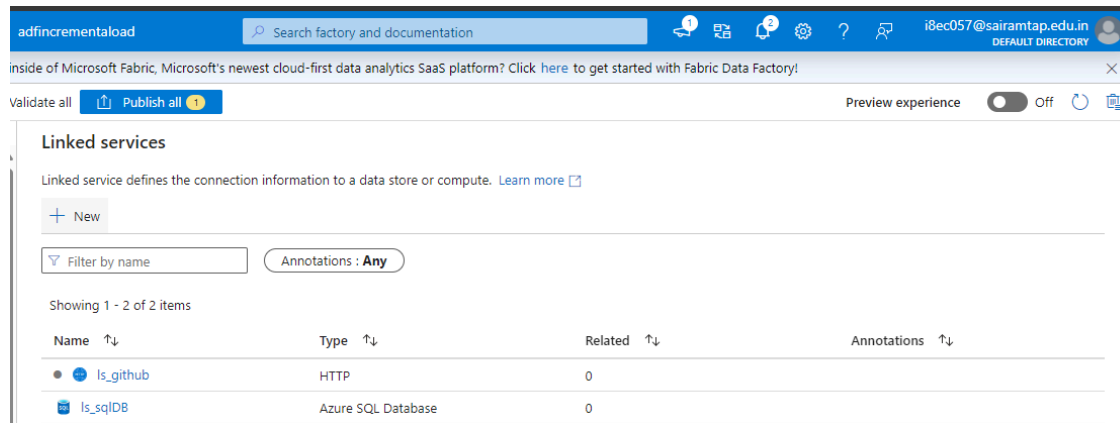
Relative URL: (Initial full load)

MohamedBashid/Azure_Project_IncrementalLoad/refs/heads/main/Dataset/SalesData.csv

Relative URL: (incremental load)

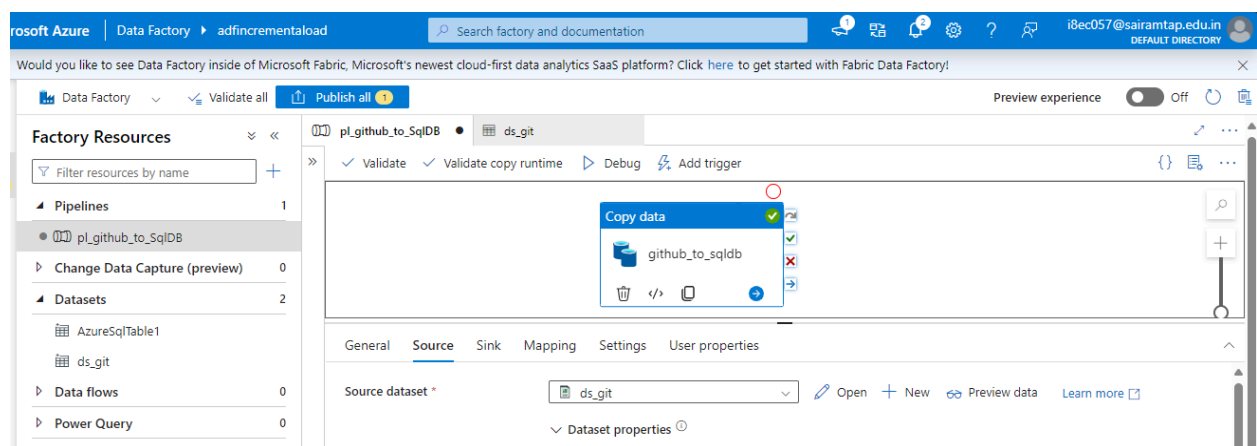
MohamedBashid/Azure_Project_IncrementalLoad/refs/heads/main/Dataset/IncrementalSales.csv

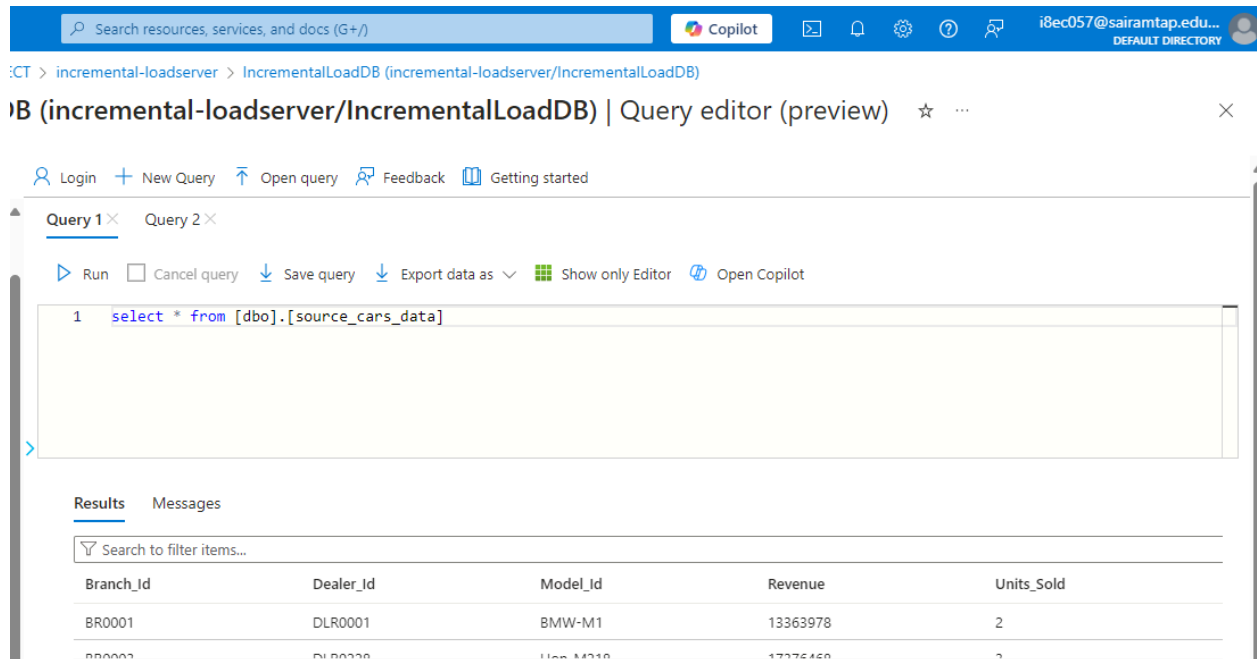
2. Created a Linked service for http(source) and Azure Sql DB(destination).



Configured the base url in http(source) linked service.

I have created a pipeline to copy the initial full load of the data from github to Azure SQL DB.



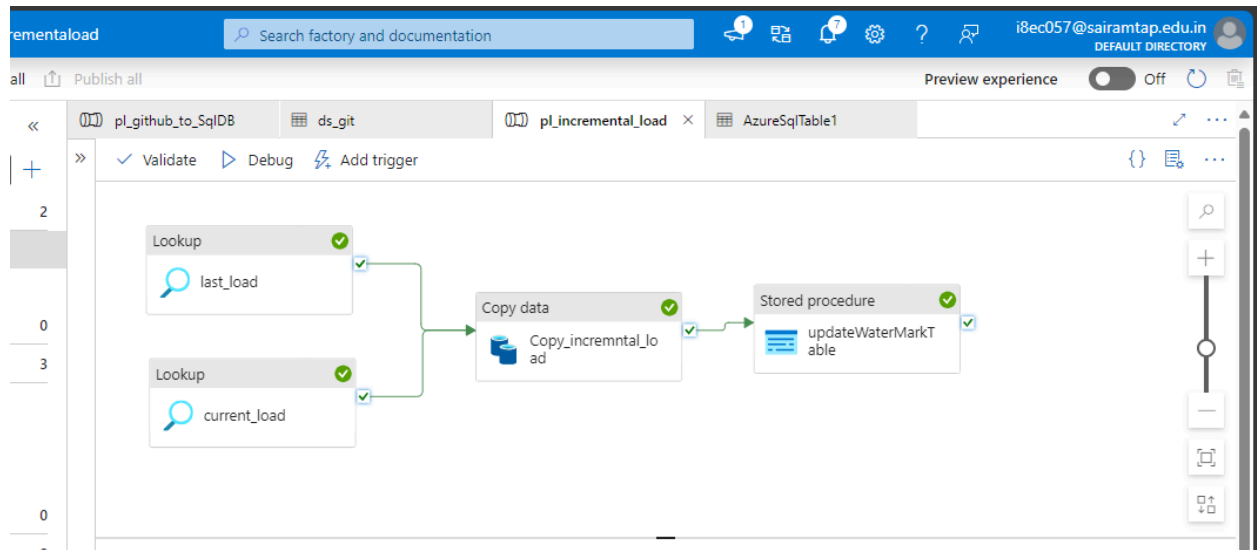


Now, for incremental load i have created a water_mark table and stored procedure in Azure SQL DB.

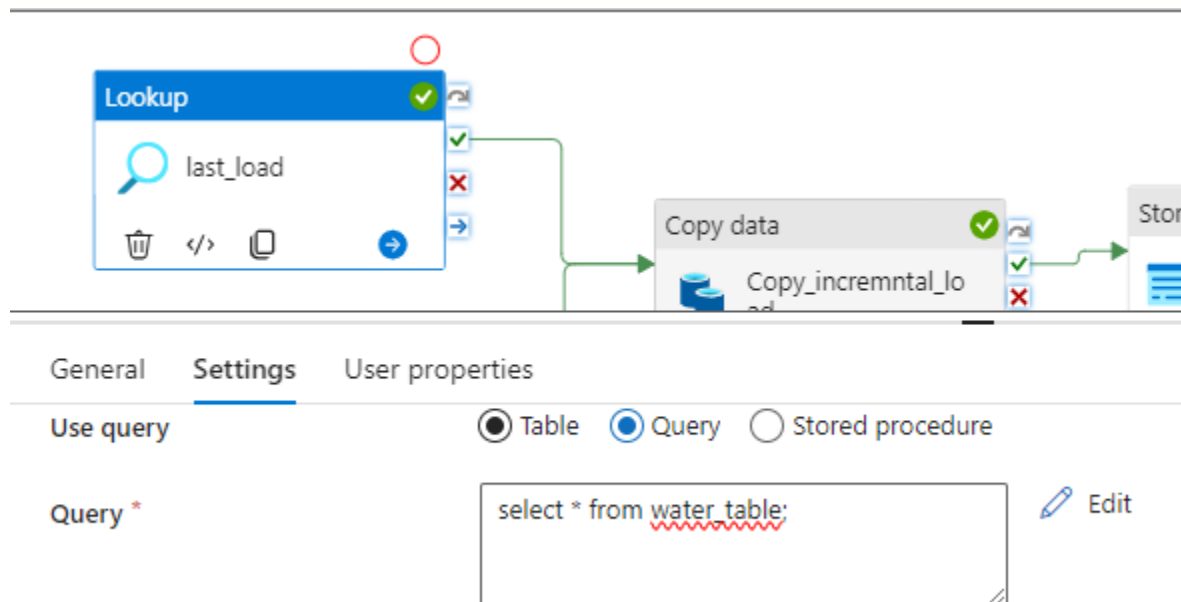
```
create table water_table
(
    last_load varchar(200)
)
insert into water_table (last_load) values ('DT00000')
```

```
create procedure updateWaterMarkTable @lastload varchar(200)
as
begin
    begin transaction;
    update water_table set last_load = @lastload
    commit transaction;
end;
```

I have created a end to end incremental load pipeline to copy data from Azure SQL DB to Azure ADLS gen2.



last_load Lookup Activity:



current_load Lookup Activity:

✓ Validate ▶ Debug ⚡ Add trigger

Lookup
current_load

Source dataset * AzureSqlTable1 [Open](#) [New](#) [Preview data](#) [Learn more](#)

Dataset properties ⓘ

Name	Value
table_name	source_cars_data

First row only ☐

Use query ☐ Table ☒ Query ☐ Stored procedure

Query * `select max(Date_Id) as max_date from source_cars_data` [Edit](#)

Copy activity:

Pipeline expression builder ⓘ

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

```
select * from source_cars_data where Date_Id > '{@activity('last_load').output.value[0].last_load}' AND Date_Id <= '{@activity('current_load').output.value[0].max_date}'
```

[Clear contents](#)

Activity outputs Parameters System variables Functions Variables

Search

- current_load
- current_load activity output
- current_load count
- Count of the rows

Stored procedure Activity:

The screenshot displays the Azure Data Factory (ADF) interface. On the left, a pipeline diagram shows a 'last_load' lookup activity followed by a 'Copy' activity. The 'Stored procedure' activity settings are visible, showing the linked service 'ls_sqldb' and the stored procedure name '[dbo].[updateWater]'. The 'Pipeline expression builder' is open, showing the expression '@activity('current_load').output.value[0].max_date'.

In the above pipeline, last_load lookup activity will contain the start date_id value with the help of water_mark_table and the current_load lookup activity will contain the maximum date_id value with the help of source_cars_data table.

In the copy activity the data from start to maximum date_id value will be copied from DB to Data Lake. In the stored procedure activity the last_load value will be assigned with the current_load which is the maximum value.

PHASE 2:

To Set up Unity Catalog:

Create a metastore in Databricks.

I have created a separate container for the unity catalog metastore.

+ Container 🔒 Change access level 🔄 Restore containers 🔄 Refresh 🗑️ Delete 🗣️ Give feedback		
<input type="text" value="Search containers by prefix"/>		
Name	Last modified	Anonymous acc
<input type="checkbox"/> \$logs	1/2/2025, 8:23:29 AM	Private
<input type="checkbox"/> bronze	1/2/2025, 9:12:48 AM	Private
<input type="checkbox"/> gold	1/2/2025, 9:13:01 AM	Private
<input type="checkbox"/> silver	1/2/2025, 9:12:55 AM	Private
<input type="checkbox"/> unitycatalogmetastore	1/2/2025, 3:11:35 PM	Private

I have created a access connector to connect the data lake with the databricks.

Create metastore

- 1 Create metastore
- 2 Assign to workspaces

*** Name**

*** Region**

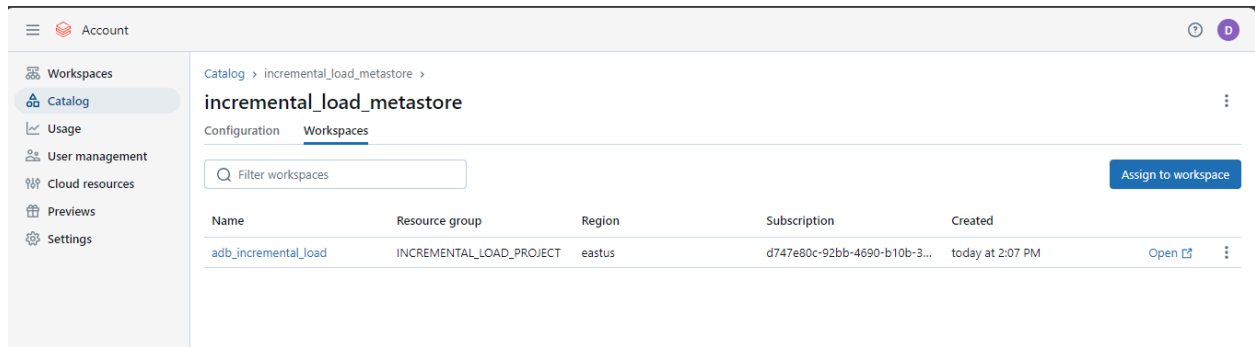
Select the region for your metastore. You will only be able to assign workspaces in this region to this metastore.

ADLS Gen 2 path (optional) ?

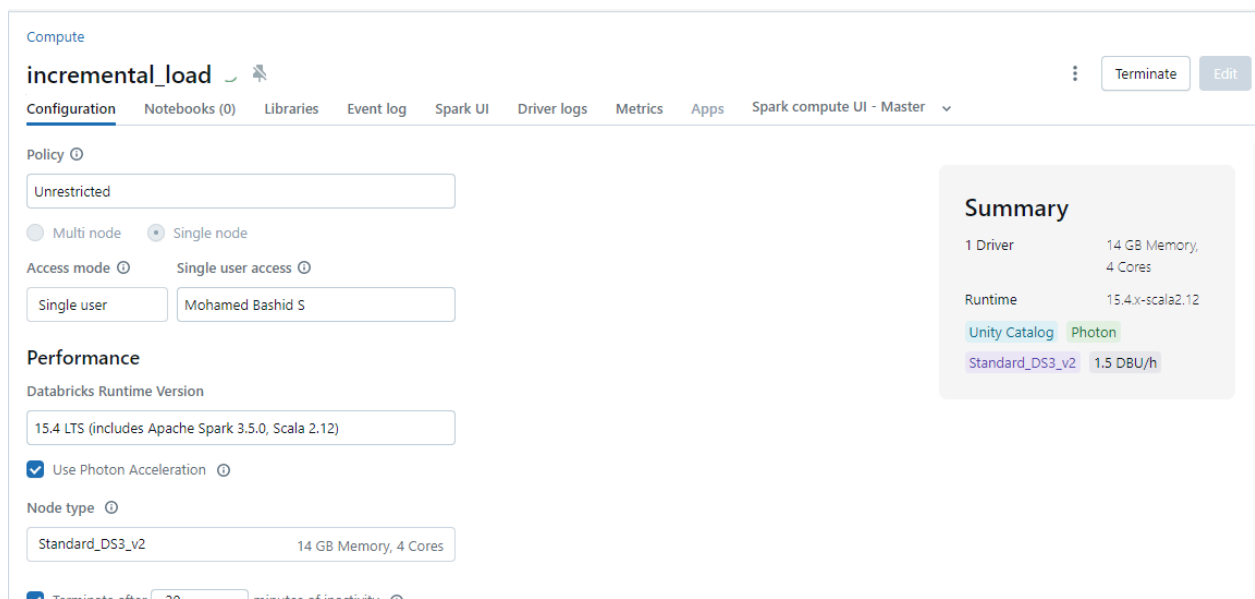
Optional location for storing managed tables data across all catalogs in the metastore. [Learn more](#)

Access Connector Id ?

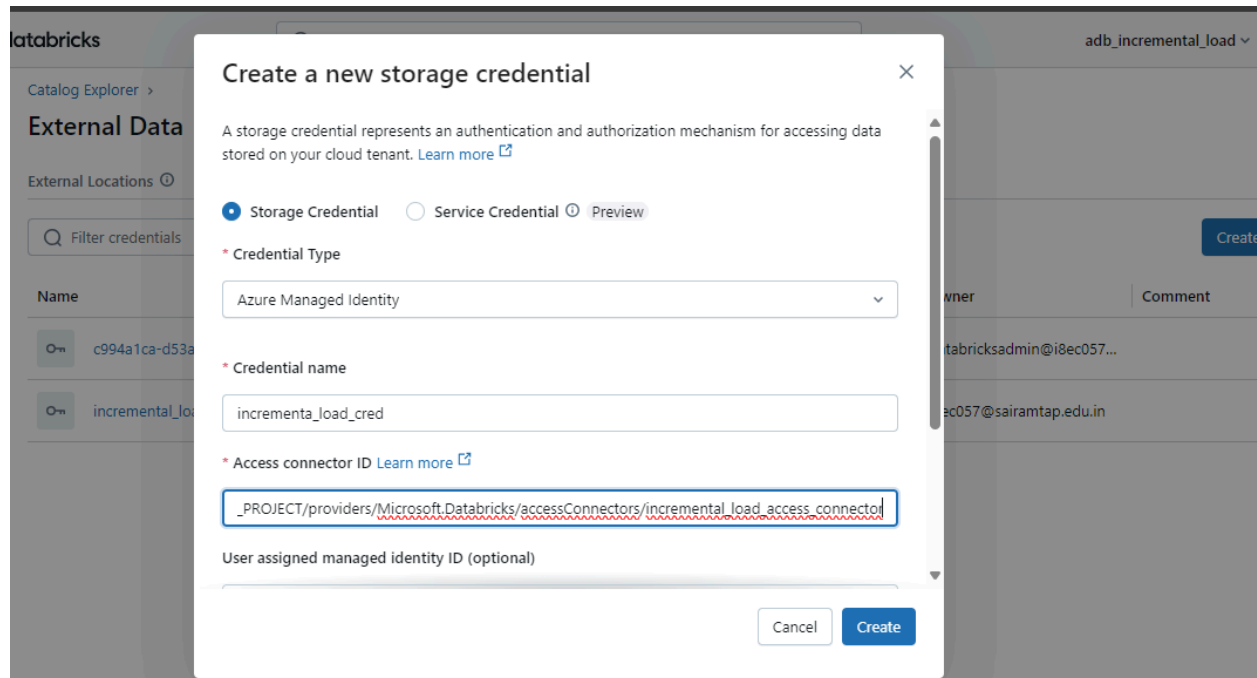
Assign the databricks workspace to the created unity catalog metastore.



I have created a cluster.



Now, i have created a external location for bronze, silver and gold container.
To create a external location, storage credential is required.
For storage credential we can use the created access connector.



External Data

External Locations 3 locations [Create external location](#)

Name	Credential	URL	Owner	Comment
bronze_ext_location	incremental_load_...	abfss://bronze@saincrementalload.dfs.core.windows.net/	i8ec057@sairamtap.edu.in	
gold_ext_location	incremental_load_...	abfss://gold@saincrementalload.dfs.core.windows.net/	i8ec057@sairamtap.edu.in	
silver_ext_location	incremental_load_...	abfss://silver@saincrementalload.dfs.core.windows.net/	i8ec057@sairamtap.edu.in	

Refer catalog_notebook

Refer silver_notebook

PHASE 3:

Creating Star Schema

Creating a Dimension Tables:

Refer gold_notebook_dim_model

Refer gold_notebook_dim_branch

Refer gold_notebook_dim_dealer

Refer gold_notebook_dim_date

Creating a Fact Table:

Refer gold_notebook_fact_sales

