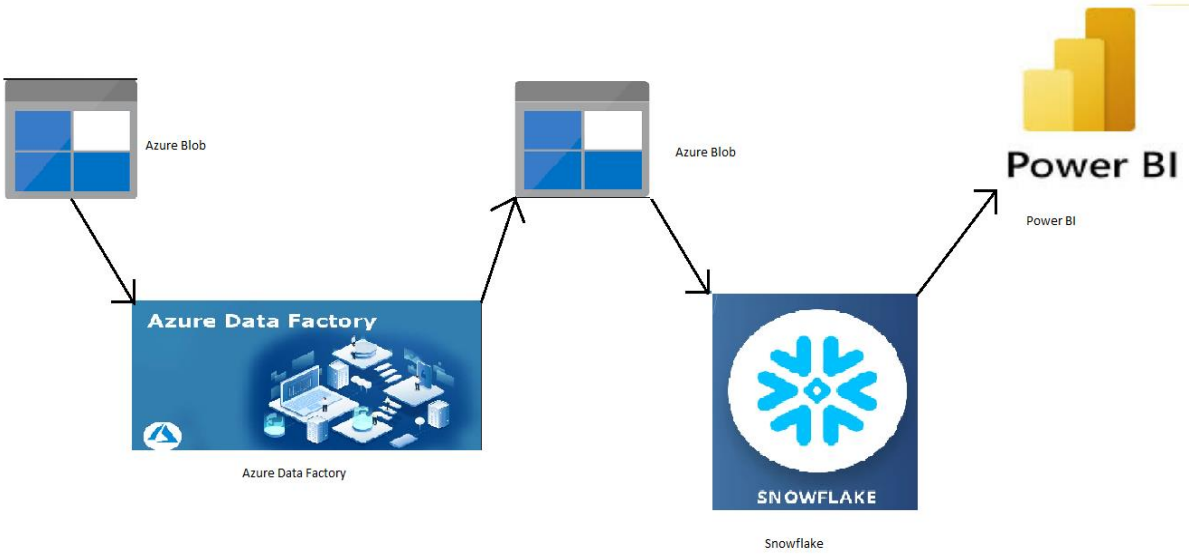
End to End Azure Data factory to Snowflake & Power BI Migration

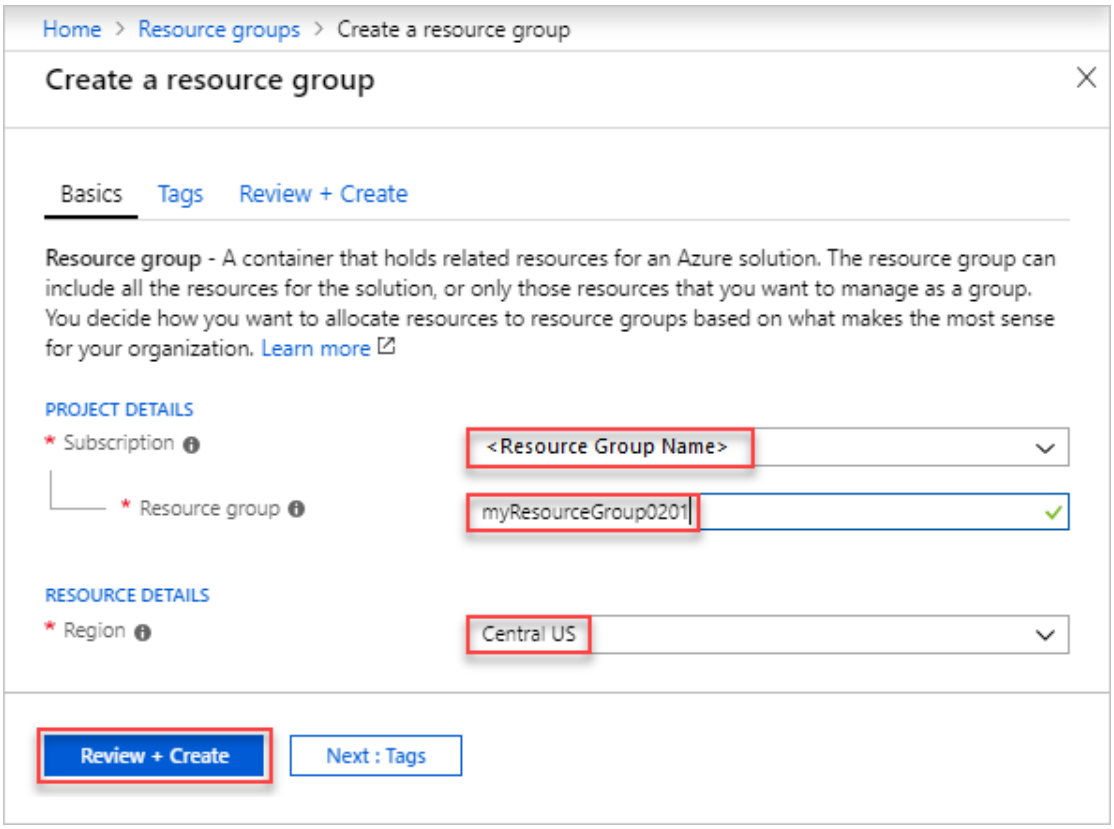
❖ Project Overview



- ✓ The goal of this project is to create a smooth data integration pipeline that collects data in JSON format from an Azure Data Blob, transforms it to CSV format, loads the data into Snowflake, and then connects Snowflake to Power BI for dashboard visualization.

- ✓ Data flow from the Azure Data Blob to Power BI is ensured through the integration process, enabling insightful data visualization and analysis.

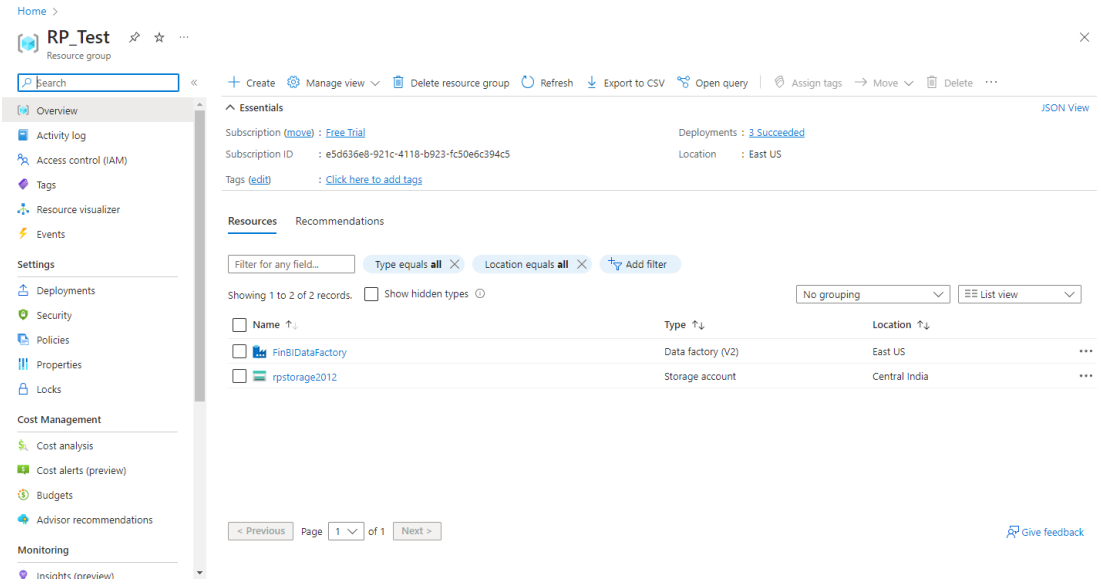
1] Create Resource Groups



- A resource group is a logical container or grouping of related resources within a cloud platform such as Microsoft Azure . It provides a way to manage and organize resources collectively, making it easier to manage, monitor, and control access to those resources.

- A resource group acts as a management unit that holds related resources for a specific application, project, or environment. It allows you to manage and apply policies, permissions, and tags to a set of resources collectively rather than managing them individually.

2] Create Azure Data Factory & Storage Account



➢ Azure Data Factory

Azure Data Factory (ADF) is a cloud-based data integration service provided by Microsoft Azure. It allows you to create, schedule, and manage data pipelines that can ingest, transform, and move data between various on-premises and cloud data sources.

➢ Azure Data Factory Perform -

✓ Data Transformation

✓ Data Movement and Copy

✓ Data Orchestration and Scheduling

✓ Monitoring and Management

✓ Integration with Ecosystem

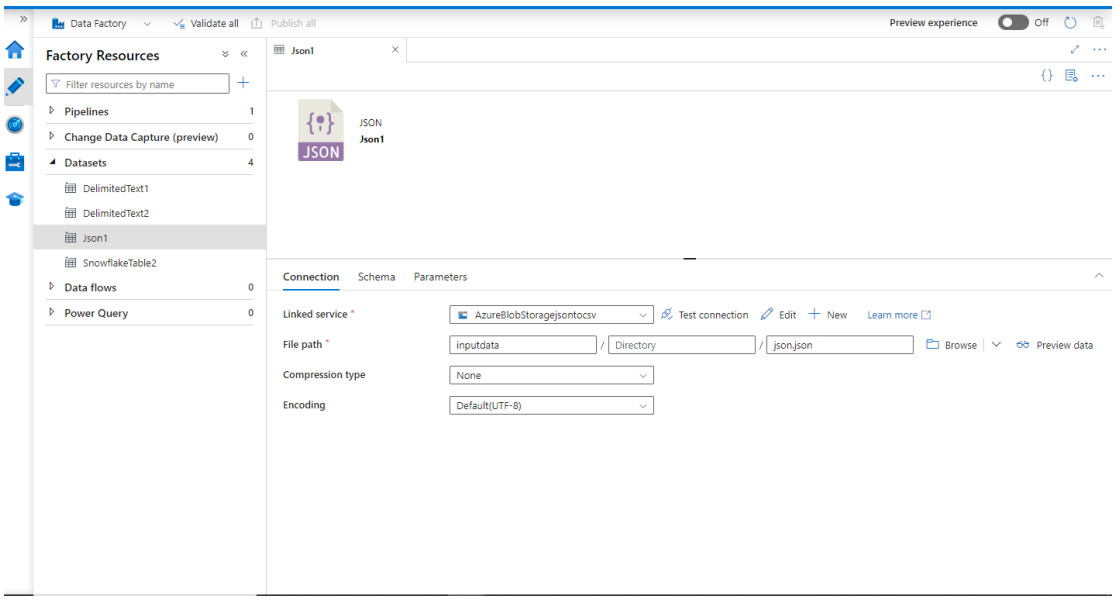
➢ Storage Account

In Azure Data Factory (ADF), a Storage Account is a key component used for storing data during data integration and movement processes. A Storage Account provides a salable and secure storage solution within the Azure ecosystem.

✧ Here's how a Storage Account is utilized in ADF:

- ✓ Source and Destination:
- ✓ Connector:
- ✓ Linked Service:
- ✓ Data Movement:
- ✓ Data Integration:
- ✓ Data Lake Integration:
- ✓ Data Partitioning:

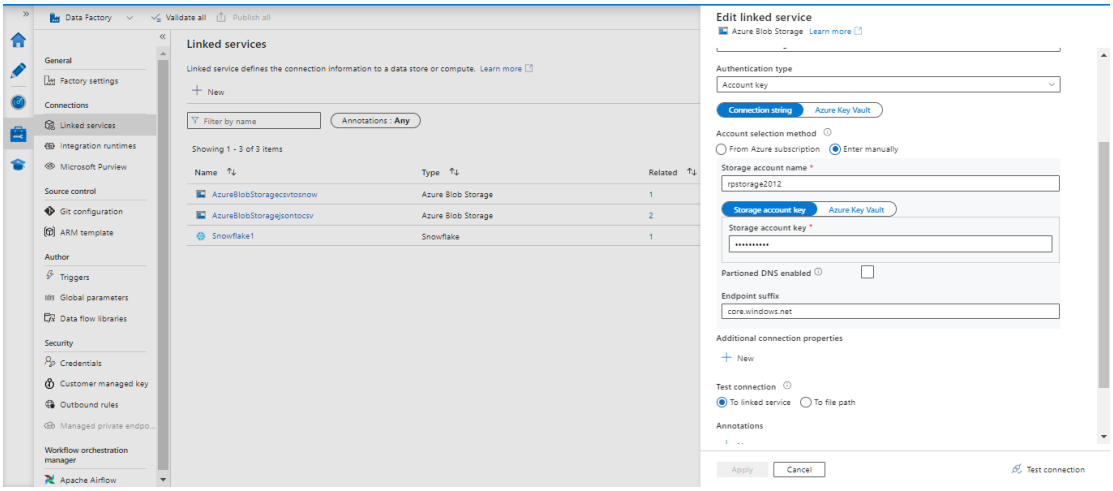
3] Create Data Set For J-son File



- ✓ Create a Blob Storage Data-Set in Azure Data Factory, this should be the data you wish to load. The connector only supports Delimited Text File loads and Parquet at this time

- ✓ You will also need to create a linked service to your storage account where your data resides.

4] Create a linked Service for J-son To CSV



- ✓ A linked service is a configuration that defines the connection information and credentials required to connect to an external data source or service.

- ✓ It acts as a bridge between Azure Data Factory and the data sources or services you want to interact with during your data integration or data transformation processes.

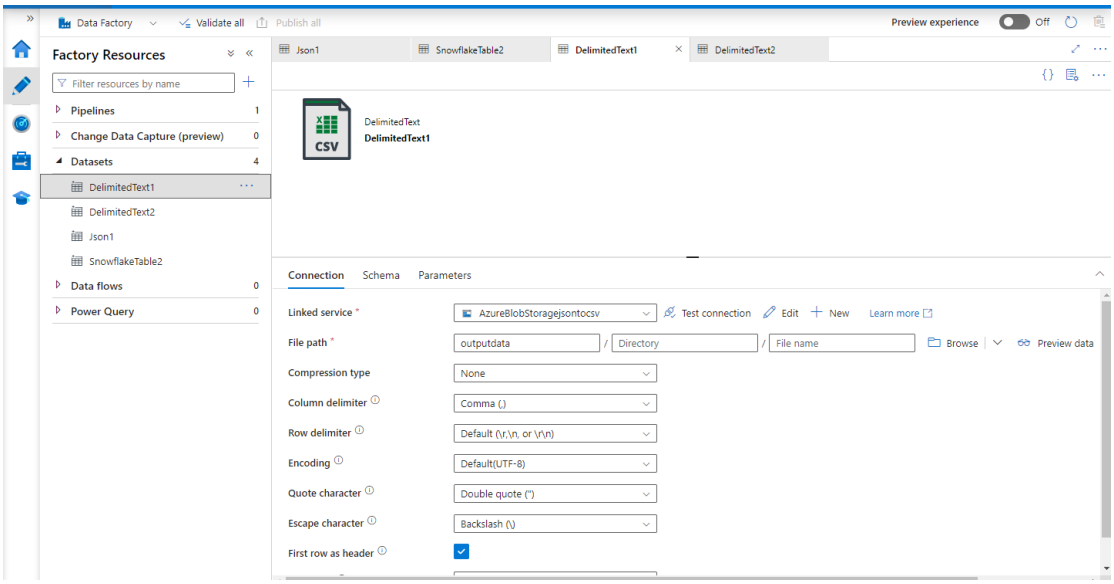
- ✓ Here we create a linked service for j-son file load into other blob and convert into CVS file format.

- ✓ Create a linked Service time give the Storage Account Name and Storage account key .

- ✓ Then after test the connection if connection is valid the linked service.

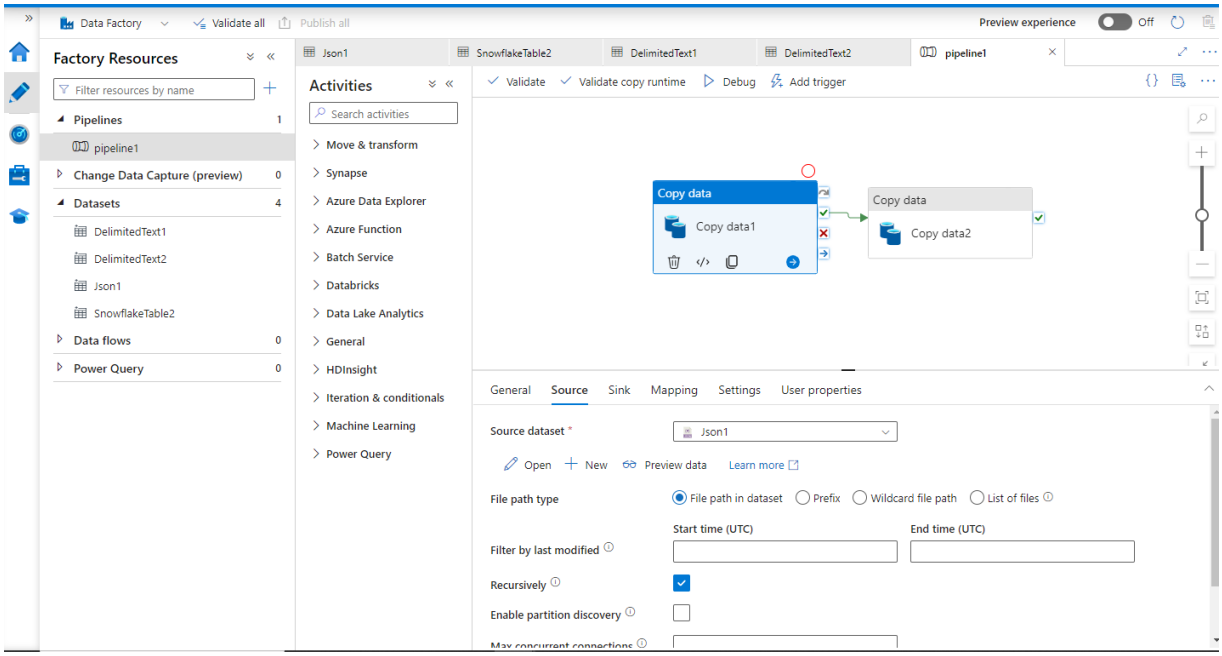
- ✓ This linked service connect the one blob to other blob storage and transform the json data to csv format

5] Create Data Set for CSV file



- ✓ Create a Blob Storage Data-Set in Azure Data Factory, this should be the data you wish to load.
- ✓ The connector only supports Delimited Text File loads and Parquet at this time.
- ✓ You will also need to create a linked service to your storage account where your data resides.

6] Create a Copy Activity For load Data One Blob to other Blob



✓ Create a new Pipeline: On the Author & Monitor hub, click on the "+ Create pipeline" button to create a new pipeline.

✓ Add an activity: Inside the created pipeline, click on the "+ Add activity" button to add a new activity.

✓ Select the Copy Activity: From the list of available activities, choose the "Copy Data" activity.

- ✓ Configure the Source dataset: In the Copy Data activity, configure the Source dataset:

  Choose the appropriate Linked Service for the Azure Blob Storage container containing the JSON files.

  Specify the JSON file format and any relevant settings like file path, folder, or wildcard patterns if required.

- ✓ Configure the Sink dataset: In the Copy Data activity, configure the Sink dataset:

  Choose the appropriate Linked Service for the Azure Blob Storage container where you want to store the CSV files.

  Specify the CSV file format and any necessary settings like file path, folder, or filename patterns.

- ✓ Mapping and Transformation: If needed, you can specify mapping and transformation within the Copy Activity to transform the data from JSON to CSV format. This step allows you to map fields, apply transformations, and define schema mapping between the source and sink datasets.

- ✓ Configure additional settings: Set any additional settings required for the Copy Activity, such as file compression options, file encoding, or other advanced settings based on your specific requirements.

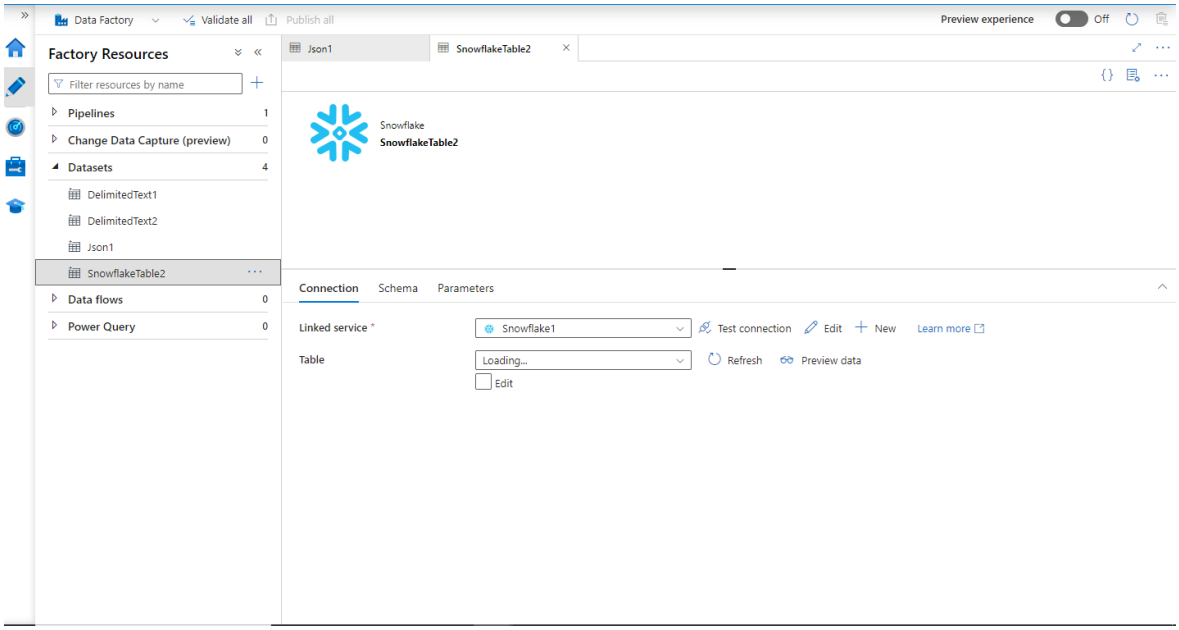
- ✓ Run and monitor the pipeline: Save the pipeline and trigger it to start the Copy Activity. Monitor the execution to ensure the data is being copied successfully from the JSON format in the source Blob Storage container to the CSV format in the destination Blob Storage container.

- ✓ With these steps, you have created a Copy Activity in Azure Data Factory to load data from one Azure Blob Storage container in

JSON format to another Azure Blob Storage container in CSV format. The Copy Activity allows you to define mappings and transformations if necessary, ensuring the data is appropriately converted during the copy process.

- ✓ Please note that the exact configuration steps and options may vary depending on your specific setup, dataset properties, and transformation requirements. Refer to the Azure Data Factory documentation for more detailed instructions and explore the available options to customize the Copy Activity based on your project's needs.

7] Create Data set for Snowflake



- ✓ Create a new dataset: In the authoring interface, click on the "New dataset" button to create a new dataset.

- ✓ Choose the appropriate dataset type: Select the appropriate dataset type based on the Snowflake data you want to connect to. This could be the Snowflake dataset type.

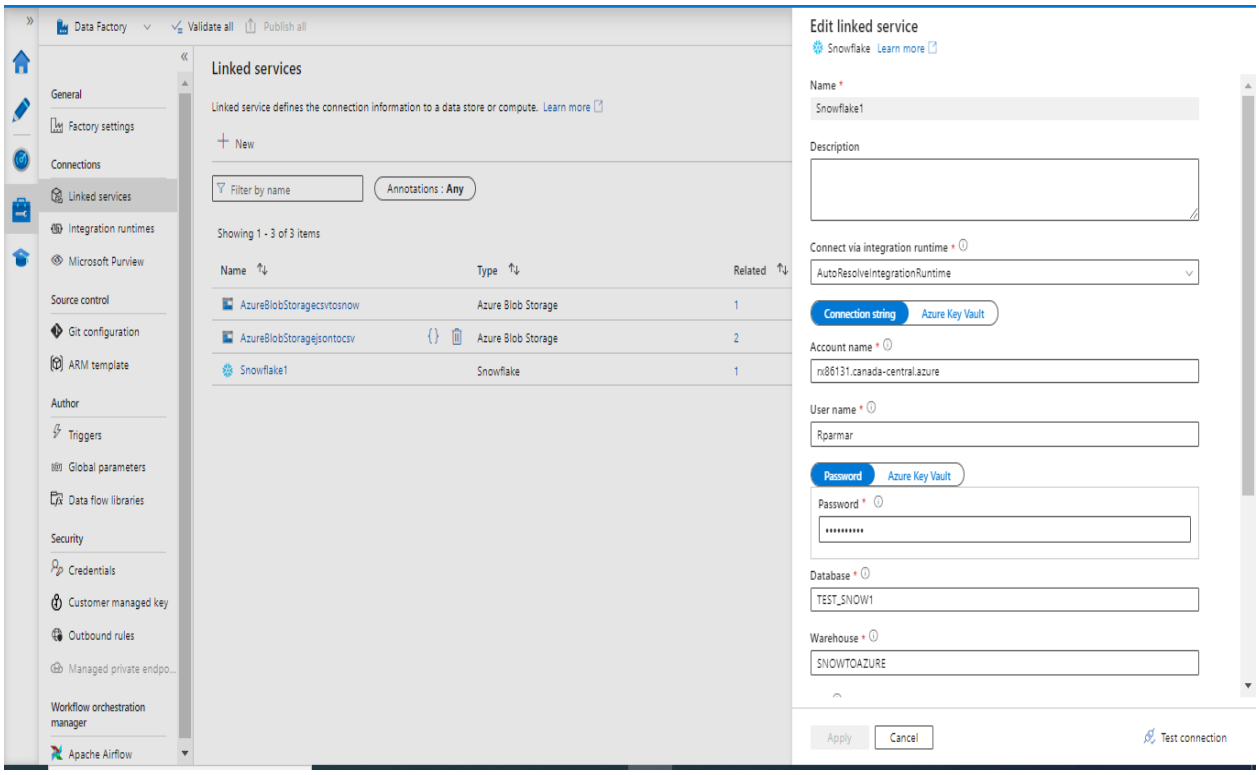
- ✓ Configure the dataset settings: Configure the dataset settings by providing the necessary information, such as the connection details to your Snowflake instance, authentication credentials, database, schema, table name, and any other relevant parameters specific to your Snowflake setup.

- ✓ Define the schema: Specify the schema of the Snowflake dataset, including the names and data types of the columns present in the Snowflake table. This helps to establish a consistent structure for processing and integration.

- ✓ Preview data: Optionally, you can preview a sample of the data from the Snowflake table to ensure the dataset configuration is accurate.

- ✓ Save the dataset: After verifying the settings and previewing the data (if applicable), save the dataset configuration.

8] Create Linked Service for ADF Blob To Snowflake



- ✓ Create a new Linked Service: On the Connections page, click on the "+ New" button to create a new Linked Service.
- ✓ Choose the appropriate Linked Service type: Select the "Azure Blob Storage" Linked Service type.
- ✓ Configure the Linked Service: Provide the necessary information to configure the Linked Service:
- ✓ Name: Give a unique name to the Linked Service.
- ✓ Connection String: Enter the connection string for the Azure Blob Storage account that contains the data you want to load into Snowflake.

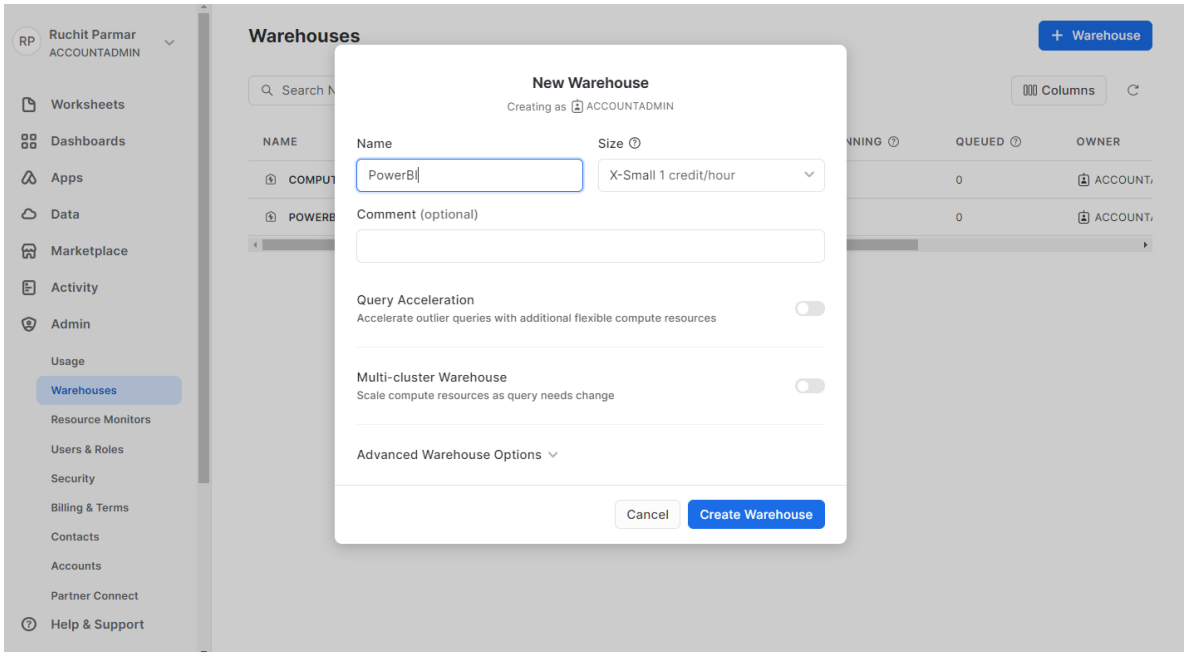
- ✓ Authentication Method: Choose the appropriate authentication method for accessing the Azure Blob Storage. This can be either using a Shared Access Signature (SAS) or an Azure Active Directory (AAD) authentication.

- ✓ Additional settings: Depending on your configuration, you may need to provide additional settings such as the container name or folder path.

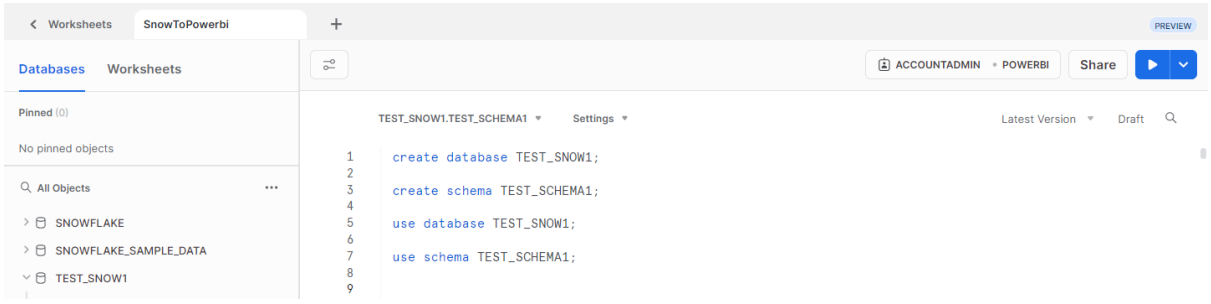
- ✓ Test the Linked Service: Once the Linked Service is configured, perform a test connection to verify the settings and ensure successful connectivity to the Azure Blob Storage account.

9] Create Warehouse & Database & schema & table in Snowflake

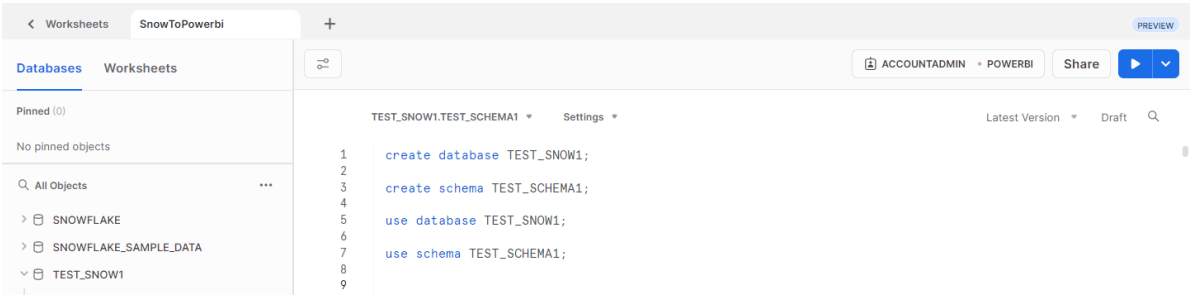
➢ Warehouse



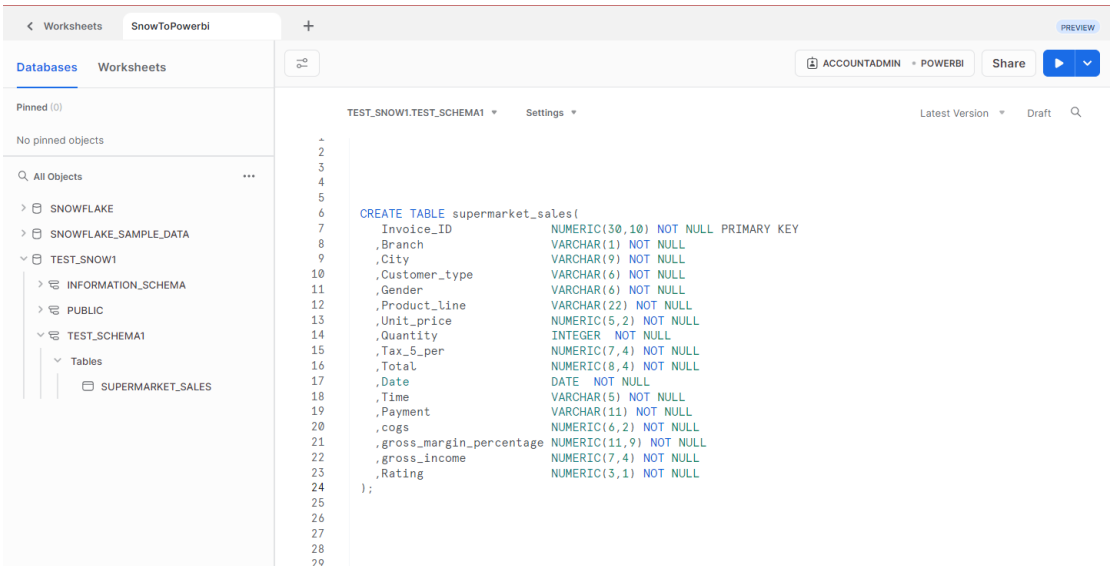
➢ Database



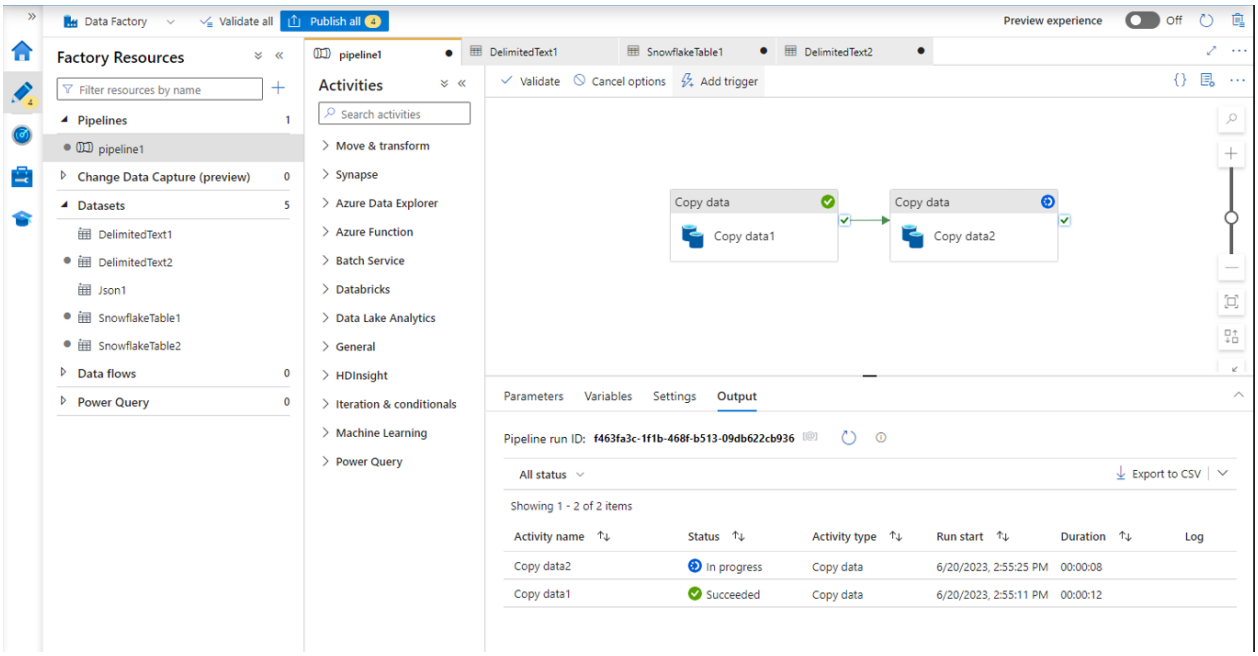
➢ schema



➢ Table



10] Create Copy Activity for ADF data Load to snowflake



✓ Create a new Pipeline: On the Author & Monitor hub, click on the "+ Create pipeline" button to create a new pipeline.

✓ Add an activity: Inside the created pipeline, click on the "+ Add activity" button to add a new activity.

✓ Select the Copy Activity: From the list of available activities, choose the "Copy Data" activity.

✓ Configure the Source dataset: In the Copy Data activity, configure the Source dataset:

- ✓ Choose the appropriate Linked Service for the data source (e.g., Azure Blob Storage, Azure Data Lake Storage).

- ✓ Specify the source dataset details such as the file path, folder, or query for retrieving the data.

- ✓ Configure the Sink dataset: In the Copy Data activity, configure the Sink dataset:

- ✓ Choose the appropriate Linked Service for Snowflake.

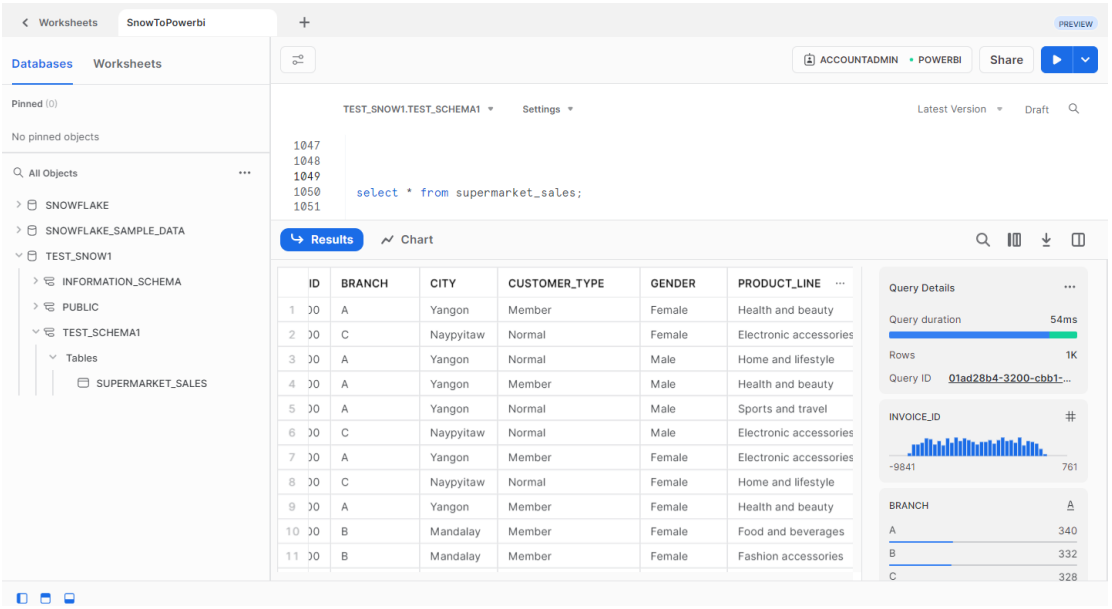
- ✓ Specify the target dataset details, including the target table or stage in Snowflake where you want to load the data.

- ✓ Mapping and Transformation: If needed, you can specify mapping and transformation within the Copy Activity to transform the data before loading it into Snowflake. This step allows you to map fields, apply data type conversions, and perform other transformations based on your data requirements.

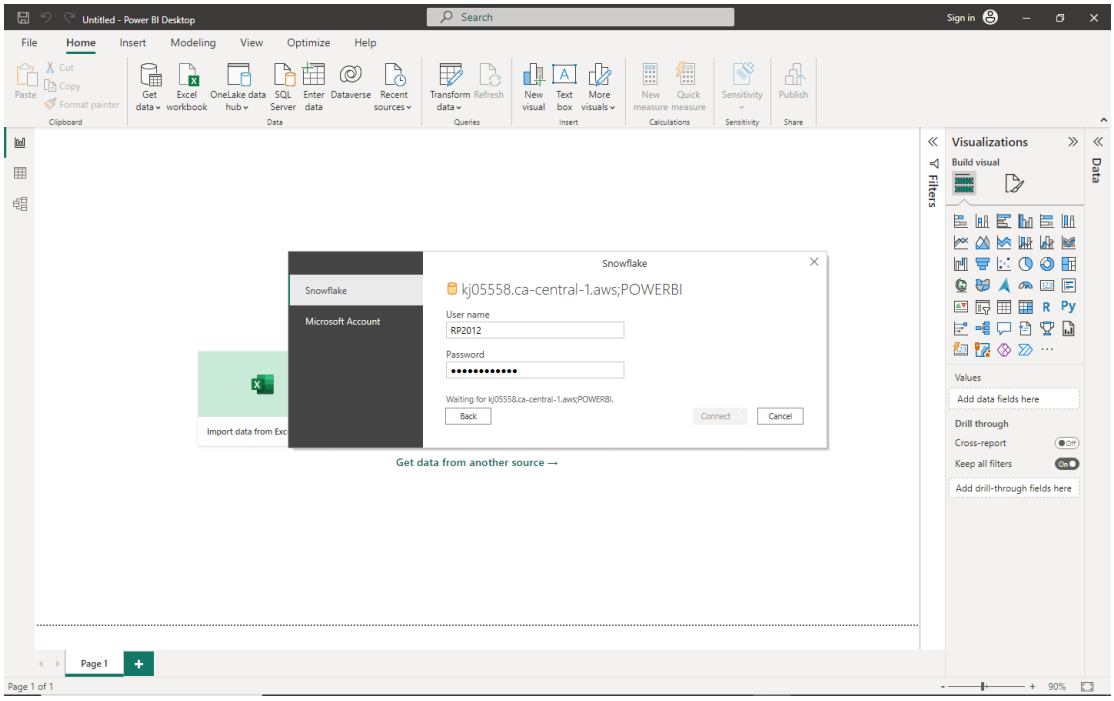
- ✓ Configure additional settings: Set any additional settings required for the Copy Activity, such as defining the write behavior (e.g., append, overwrite) and enabling options like schema drift handling or error handling.

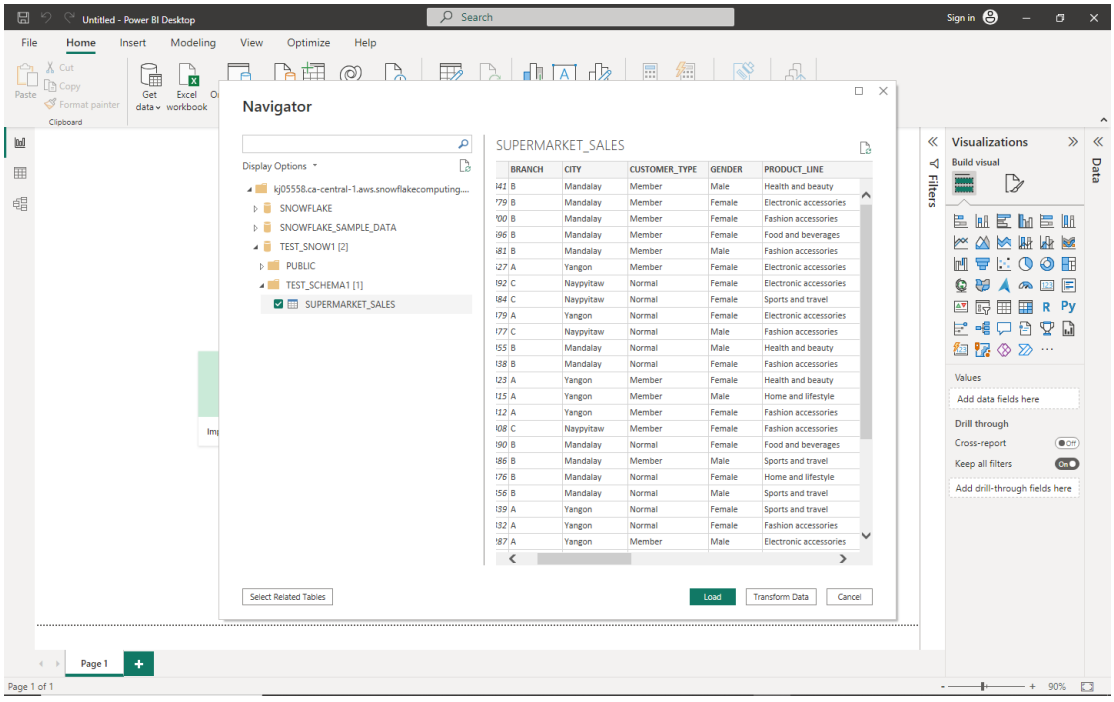
- ✓ Run and monitor the pipeline: Save the pipeline and trigger it to start the Copy Activity. Monitor the execution to ensure the data is being loaded successfully from the source to Snowflake.

11] Insert Data into warehouse using ADF Pipeline



12] Connect Snowflake To Power BI & Load the Data





- ✓ Select "Snowflake" as the data source: In the "Get Data" window, search for "Snowflake" and select the "Snowflake" connector from the available options.

- ✓ Provide Snowflake connection details: In the "Snowflake" connector window, enter the necessary information to connect to your Snowflake database:

1. Server: Enter the Snowflake server URL.

2. Database: Specify the Snowflake database name.

3. Warehouse: Select the appropriate Snowflake warehouse.

4. Authentication method: Choose the desired authentication method (e.g., Username and password, Integrated).

5. Username/Password or Integrated: Provide the necessary credentials based on the chosen authentication method.

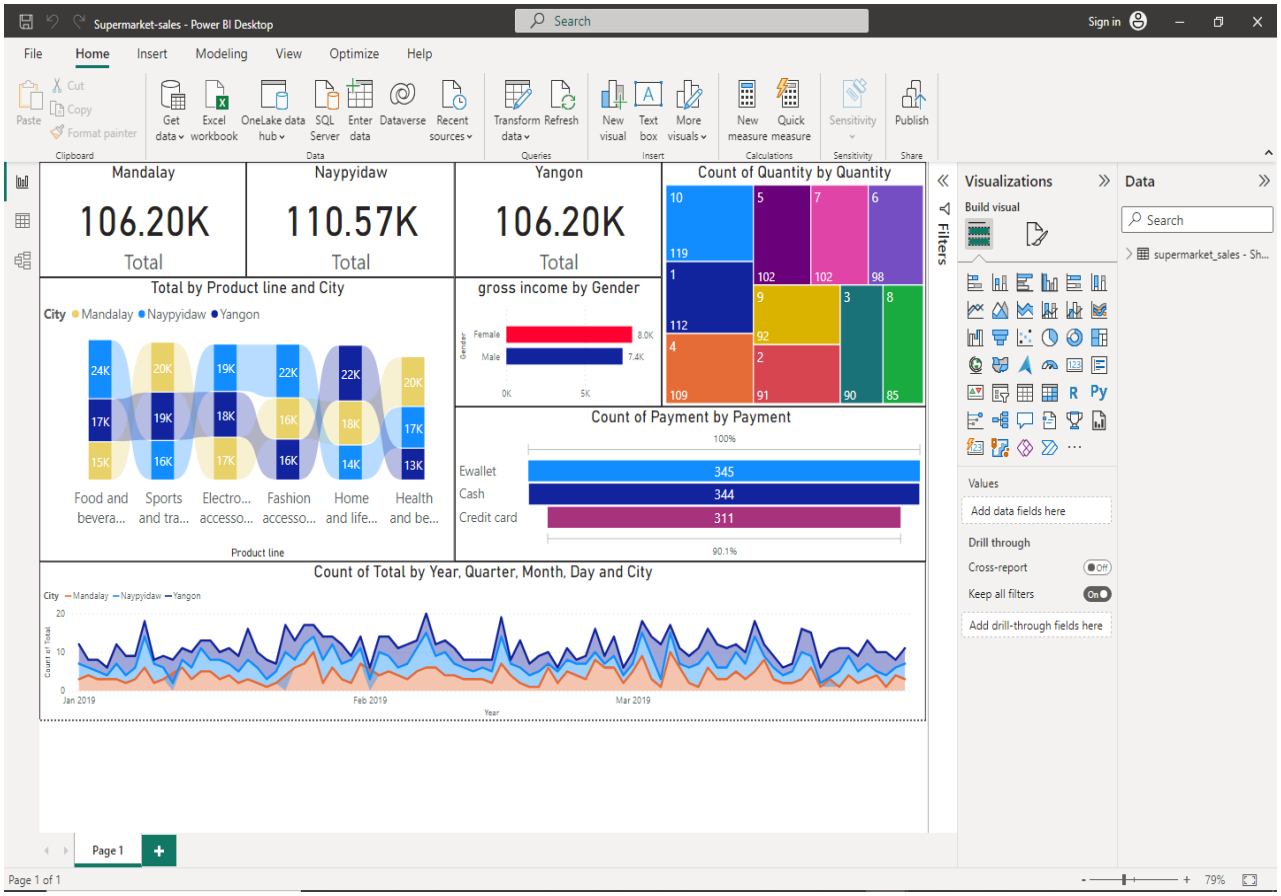
6. Connect to Snowflake: After providing the required connection details, click on the "Connect" button to establish a connection to Snowflake.

- ✓ Select tables or write SQL queries: Once connected, you can select the desired tables or views from Snowflake to import into Power BI. Alternatively, you can write custom SQL queries to retrieve specific data for visualization.

- ✓ Transform and shape data (optional): If needed, you can apply transformations to the imported data within Power BI using the Power Query Editor. This allows you to clean, filter, merge, or transform the data as per your reporting requirements.

- ✓ Create visualizations: With the imported data, you can now create interactive visualizations, reports, and dashboards in Power BI Desktop.
- ✓ Save the Power BI report: Once you have created the desired visualizations, save the Power BI report in a local file format (e.g., .pbix).

13] visualization on Power BI Dashboard



Thank You