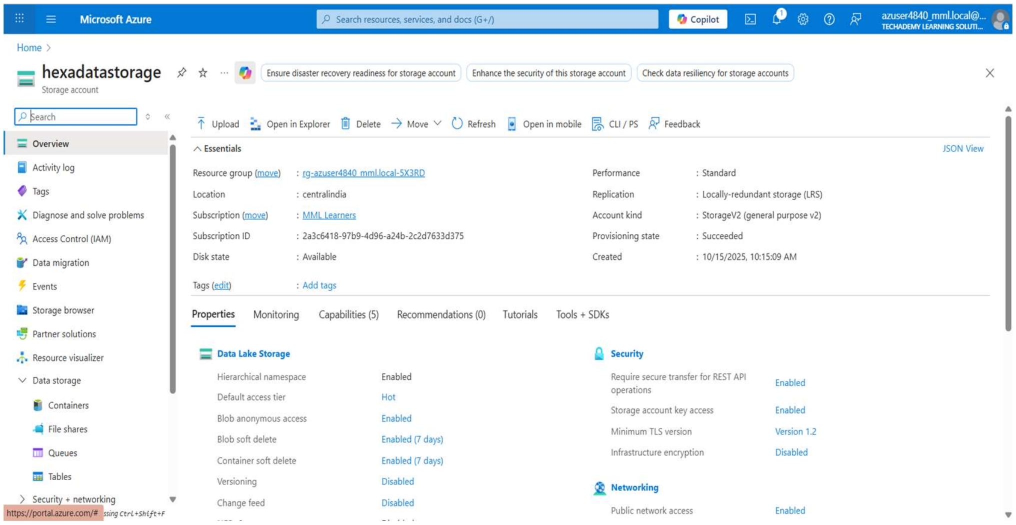
Case Study: Ingesting and Modeling Retail Sales Data from Azure to Snowflake using Snowpark, Visualized in Power BI

Step 1: Upload Data to Azure Storage

- Created an Azure Storage Account named **hexadatastorage**.
- Created a container named **retailsales** inside the storage account.
- Uploaded the file **Retail_Sales__500_rows__Preview.csv** into the container.
- Verified the file upload in the Azure Portal.



hexadatastorage
Storage account

Essentials

Resource group (move)	: rg-azuser4840_mml.local-5X3RD	Performance	: Standard
Location	: centralindia	Replication	: Locally-redundant storage (LRS)
Subscription (move)	: MML Learners	Account kind	: StorageV2 (general purpose v2)
Subscription ID	: 2a3c6418-97b9-4d96-a24b-2c2d7633d375	Provisioning state	: Succeeded
Disk state	: Available	Created	: 10/15/2025, 10:15:09 AM
Tags (edit)	: Add tags		

Data Lake Storage

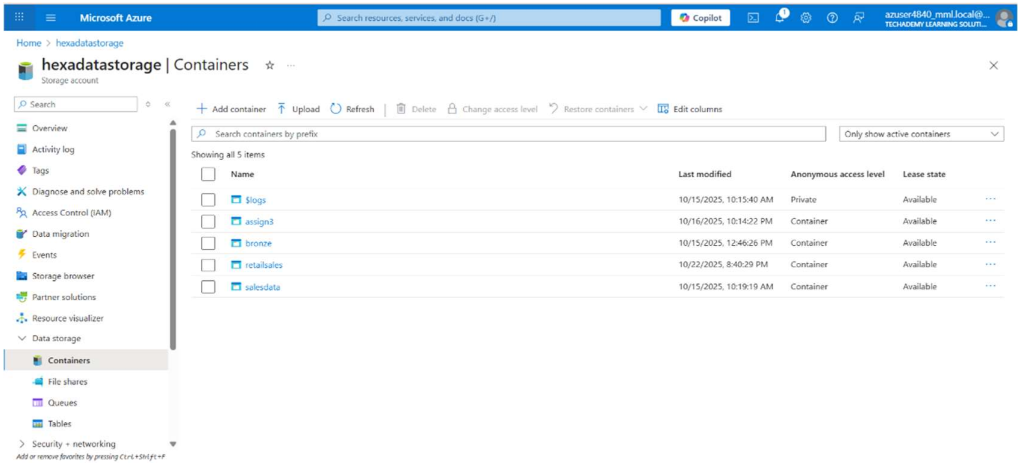
Hierarchical namespace	Enabled
Default access tier	Hot
Blob anonymous access	Enabled
Blob soft delete	Enabled (7 days)
Container soft delete	Enabled (7 days)
Versioning	Disabled
Change feed	Disabled

Security

Require secure transfer for REST API operations	Enabled
Storage account key access	Enabled
Minimum TLS version	Version 1.2
Infrastructure encryption	Disabled

Networking

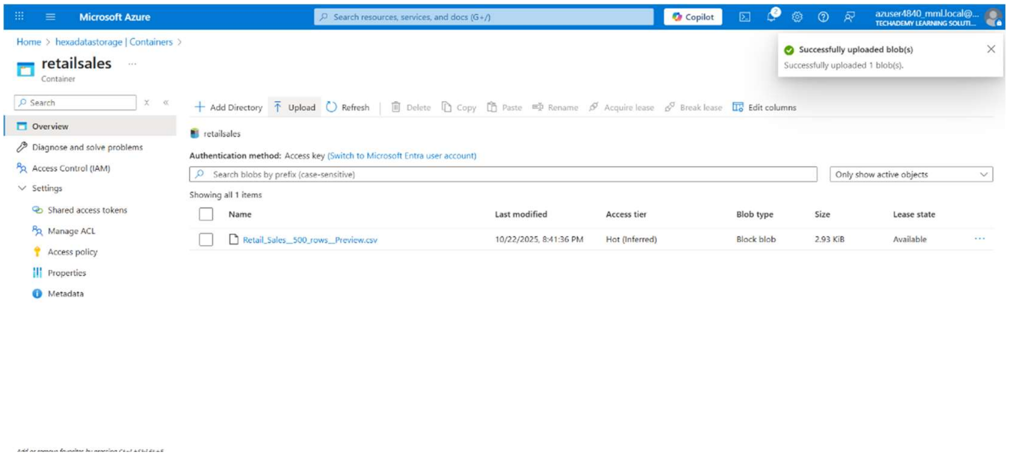
Public network access	Enabled



hexadatastorage | Containers
Storage account

Showing all 5 items

Name	Last modified	Anonymous access level	Lease state
$logs	10/15/2025, 10:15:40 AM	Private	Available
assign3	10/16/2025, 10:14:22 PM	Container	Available
bronze	10/15/2025, 12:46:26 PM	Container	Available
retailsales	10/22/2025, 8:40:29 PM	Container	Available
salesdata	10/15/2025, 10:19:19 AM	Container	Available



Successfully uploaded blob(s)
Successfully uploaded 1 blob(s).

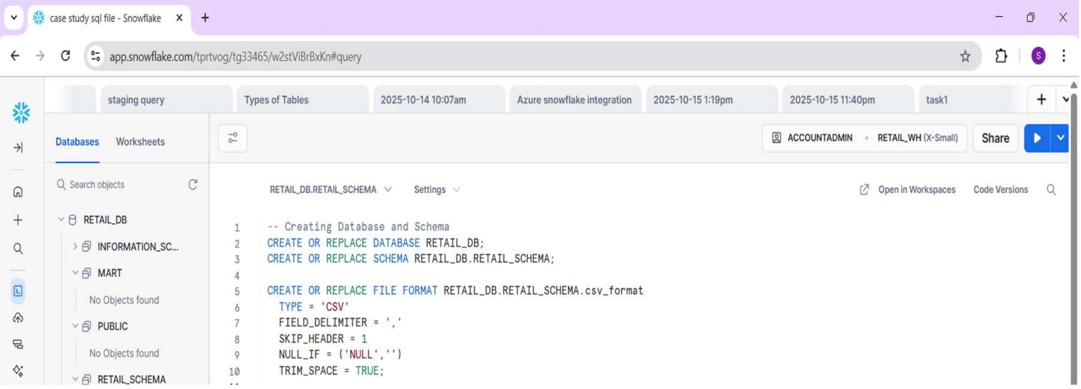
Authentication method: Access key (Switch to Microsoft Entra user account)

Showing all 1 items

Name	Last modified	Access tier	Blob type	Size	Lease state
Retail_Sales__500_rows__Preview.csv	10/22/2025, 8:41:36 PM	Hot (Inferred)	Block blob	2.93 KiB	Available

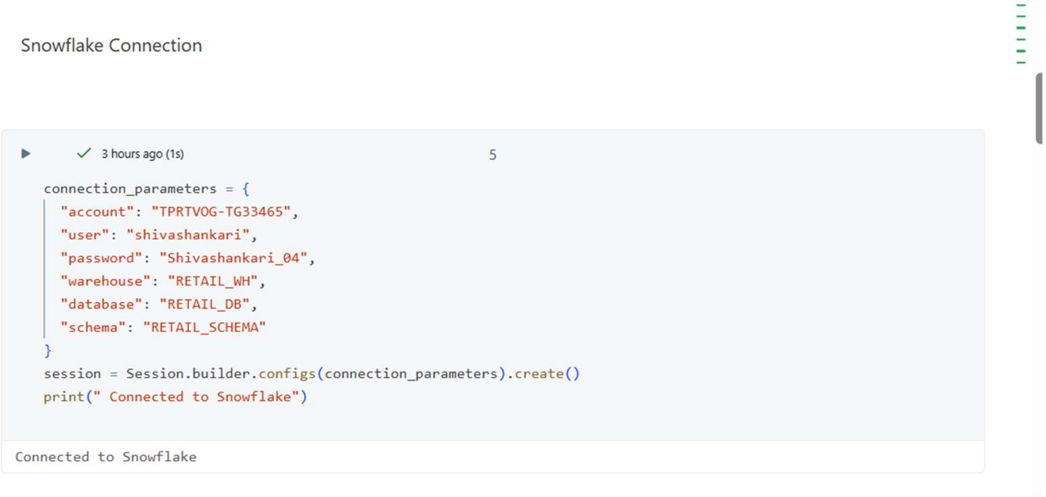
Step 2: Ingest Data into Snowflake using Snowpark

- Created database **RETAIL_DB** and schema **RETAIL_SCHEMA** in Snowflake.



```sql
-- Creating Database and Schema
CREATE OR REPLACE DATABASE RETAIL_DB;
CREATE OR REPLACE SCHEMA RETAIL_DB.RETAIL_SCHEMA;

CREATE OR REPLACE FILE FORMAT RETAIL_DB.RETAIL_SCHEMA.csv_format
  TYPE = 'CSV'
  FIELD_DELIMITER = ','
  SKIP_HEADER = 1
  NULL_IF = ('NULL','')
  TRIM_SPACE = TRUE;
```

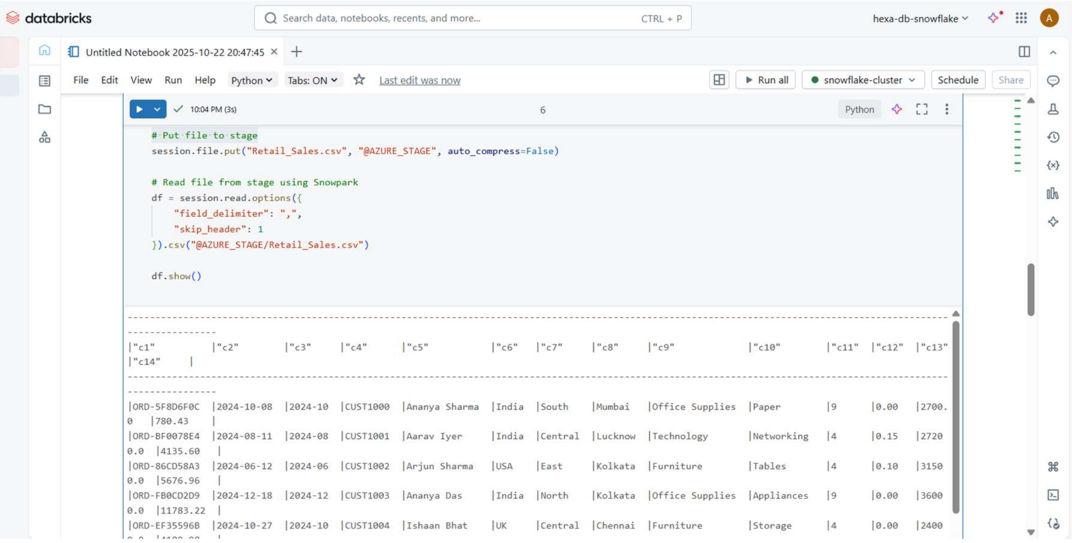
- Connected to Snowflake from Snowpark.


Snowflake Connection

```python
connection_parameters = {
  "account": "TPRTVOG-TG33465",
  "user": "shivashankari",
  "password": "Shivashankari_04",
  "warehouse": "RETAIL_WH",
  "database": "RETAIL_DB",
  "schema": "RETAIL_SCHEMA"
}
session = Session.builder.configs(connection_parameters).create()
print(" Connected to Snowflake")
```

```
Connected to Snowflake
```

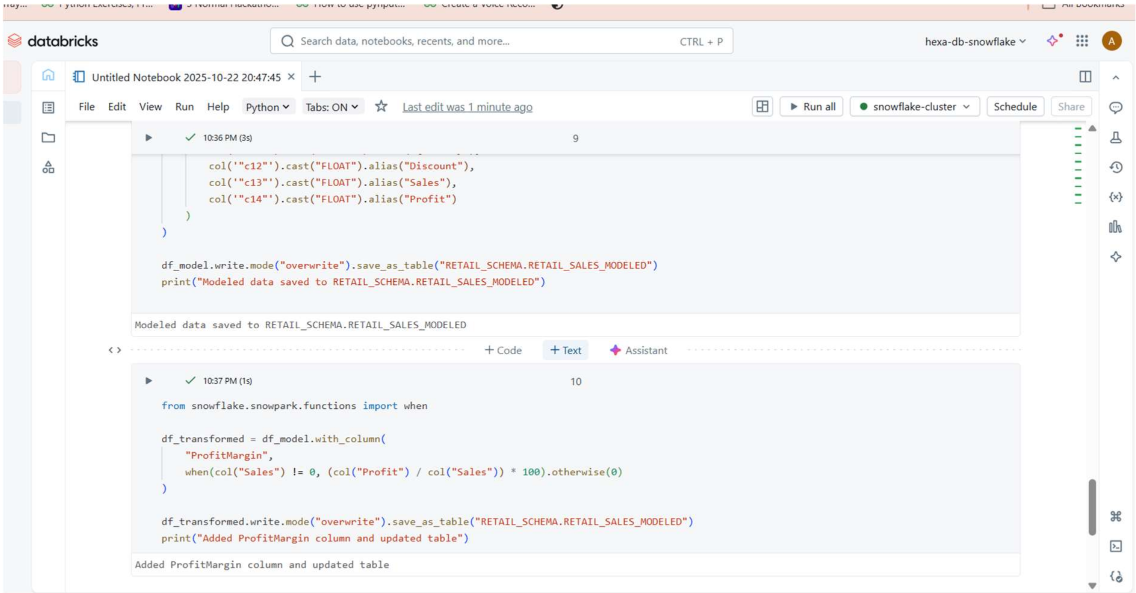
- Created an internal stage to store incoming files.
- Uploaded the CSV file from local/Azure to the Snowflake stage.
- Loaded the CSV from stage into a **Snowpark DataFrame**.
- Verified that the data loaded successfully by displaying sample records.


databricks
Search data, notebooks, recents, and more... CTRL + P
hexa-db-snowflake
Untitled Notebook 2025-10-22 20:47:45
File Edit View Run Help Python Tabs: ON Last edit was now
Run all
snowflake-cluster
Schedule
Share
10:04 PM (3s)
6
Python

```
# Put file to stage
session.file.put("Retail_Sales.csv", "@AZURE_STAGE", auto_compress=False)

# Read file from stage using Snowpark
df = session.read.options({
    "field_delimiter": ",",
    "skip_header": 1
}).csv("@AZURE_STAGE/Retail_Sales.csv")

df.show()
```

```
---------------------------------------------------------------------------------------------------------------------------------
---------------
|"c1"         |"c2"        |"c3"      |"c4"      |"c5"            |"c6"   |"c7"     |"c8"     |"c9"             |"c10"        |"c11"  |"c12"  |"c13"
|"c14"     |
---------------------------------------------------------------------------------------------------------------------------------
---------------
|ORD-5F8D6F0C  |2024-10-08  |2024-10   |CUST1000  |Ananya Sharma   |India  |South    |Mumbai   |Office Supplies  |Paper        |9      |0.00   |2700.
0   |780.43    |
|ORD-BF0078E4  |2024-08-11  |2024-08   |CUST1001  |Aarav Iyer      |India  |Central  |Lucknow  |Technology       |Networking   |4      |0.15   |2720
0.0  |4135.60   |
|ORD-86CD58A3  |2024-06-12  |2024-06   |CUST1002  |Arjun Sharma    |USA    |East     |Kolkata  |Furniture        |Tables       |4      |0.10   |3150
0.0  |5676.96   |
|ORD-FB0CD2D9  |2024-12-18  |2024-12   |CUST1003  |Ananya Das      |India  |North    |Kolkata  |Office Supplies  |Appliances   |9      |0.00   |3600
0.0  |11783.22  |
|ORD-EF35596B  |2024-10-27  |2024-10   |CUST1004  |Ishaan Bhat     |UK     |Central  |Chennai  |Furniture        |Storage      |4      |0.00   |2400
```

- Renamed and casted columns to proper data types (OrderDate, Sales, Profit, etc.).
- Added a derived column **ProfitMargin** for analysis.
- Saved the final modeled table as **RETAIL_SALES_MODELED** in RETAIL_SCHEMA.


databricks
Search data, notebooks, recents, and more... CTRL + P
hexa-db-snowflake
Untitled Notebook 2025-10-22 20:47:45
File Edit View Run Help Python Tabs: ON Last edit was 1 minute ago
Run all
snowflake-cluster
Schedule
Share
10:36 PM (3s)
9
+ Code
+ Text
Assistant
10:37 PM (1s)
10

```
        col('"c12"').cast("FLOAT").alias("Discount"),
        col('"c13"').cast("FLOAT").alias("Sales"),
        col('"c14"').cast("FLOAT").alias("Profit")
    )
)

df_model.write.mode("overwrite").save_as_table("RETAIL_SCHEMA.RETAIL_SALES_MODELED")
print("Modeled data saved to RETAIL_SCHEMA.RETAIL_SALES_MODELED")
```

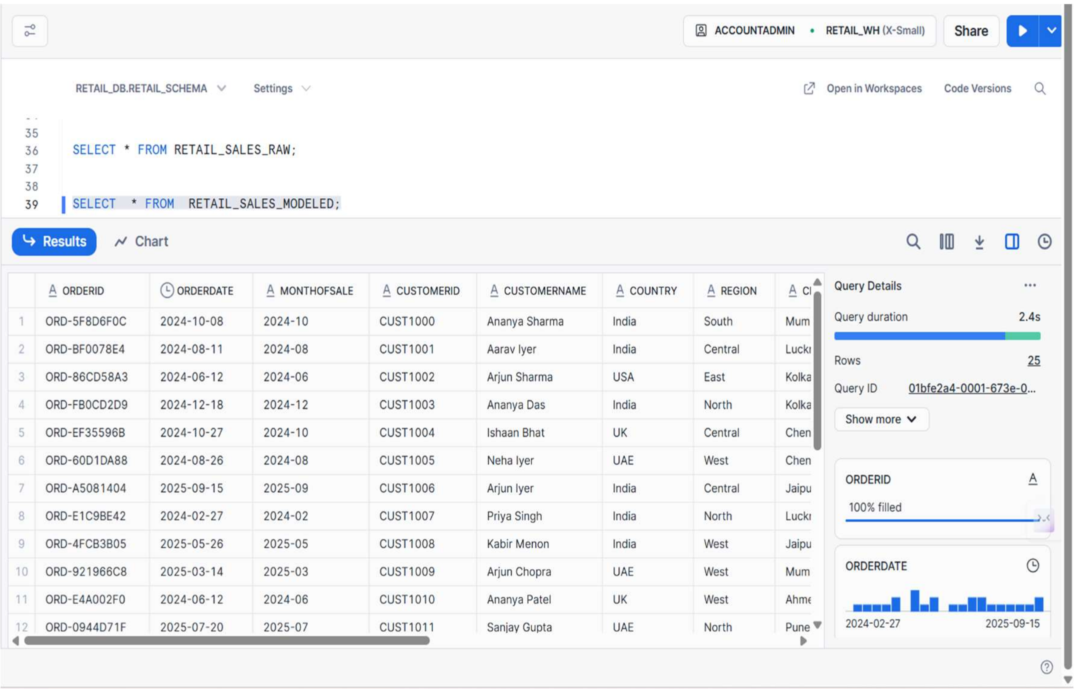
```
Modeled data saved to RETAIL_SCHEMA.RETAIL_SALES_MODELED
```

```
from snowflake.snowpark.functions import when

df_transformed = df_model.with_column(
    "ProfitMargin",
    when(col("Sales") != 0, (col("Profit") / col("Sales")) * 100).otherwise(0)
)

df_transformed.write.mode("overwrite").save_as_table("RETAIL_SCHEMA.RETAIL_SALES_MODELED")
print("Added ProfitMargin column and updated table")
```

```
Added ProfitMargin column and updated table
```

Step 3: Model and Structure Data in Snowflake

- Ensured the table **RETAIL_SALES_MODELED** contains clean, well-typed data.
- Verified column names, data types, and row counts.
- Ensured all key columns are correctly formatted (dates, floats, numbers).
- Confirmed transformations and calculations (e.g., Profit Margin).



```
SELECT * FROM RETAIL_SALES_RAW;


SELECT  * FROM  RETAIL_SALES_MODELED;
```

	ORDERID	ORDERDATE	MONTHOFSALE	CUSTOMERID	CUSTOMERNAME	COUNTRY	REGION	CI
1	ORD-5F8D6F0C	2024-10-08	2024-10	CUST1000	Ananya Sharma	India	South	Mum
2	ORD-BF0078E4	2024-08-11	2024-08	CUST1001	Aarav Iyer	India	Central	Luckı
3	ORD-86CD58A3	2024-06-12	2024-06	CUST1002	Arjun Sharma	USA	East	Kolka
4	ORD-FB0CD2D9	2024-12-18	2024-12	CUST1003	Ananya Das	India	North	Kolka
5	ORD-EF35596B	2024-10-27	2024-10	CUST1004	Ishaan Bhat	UK	Central	Chen
6	ORD-60D1DA88	2024-08-26	2024-08	CUST1005	Neha Iyer	UAE	West	Chen
7	ORD-A5081404	2025-09-15	2025-09	CUST1006	Arjun Iyer	India	Central	Jaipu
8	ORD-E1C9BE42	2024-02-27	2024-02	CUST1007	Priya Singh	India	North	Luckı
9	ORD-4FCB3B05	2025-05-26	2025-05	CUST1008	Kabir Menon	India	West	Jaipu
10	ORD-921966C8	2025-03-14	2025-03	CUST1009	Arjun Chopra	UAE	West	Mum
11	ORD-E4A002F0	2024-06-12	2024-06	CUST1010	Ananya Patel	UK	West	Ahme
12	ORD-0944D71F	2025-07-20	2025-07	CUST1011	Sanjay Gupta	UAE	North	Pune

Query Details

Query duration 2.4s

Rows 25

Query ID 01bfe2a4-0001-673e-0...

Show more

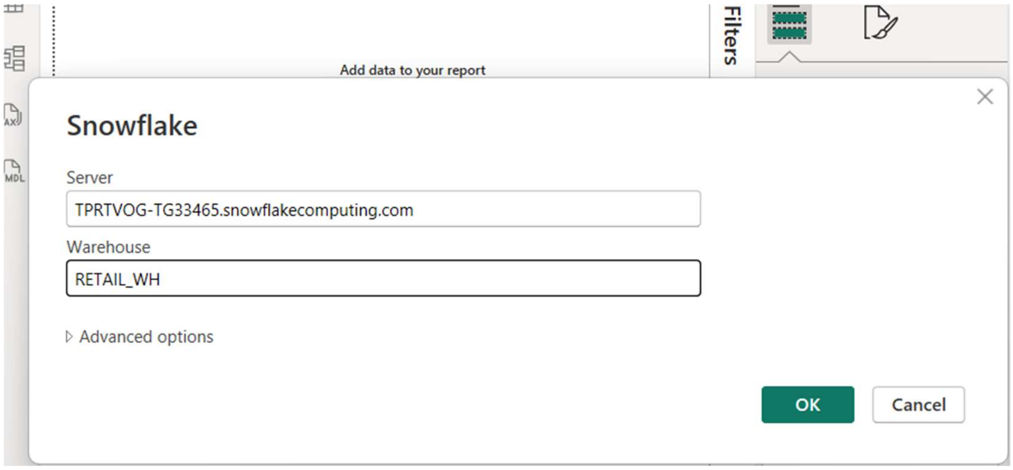
ORDERID

100% filled

ORDERDATE

Step 4: Build Power BI Report

- Connected Power BI Desktop to Snowflake using database and schema credentials.



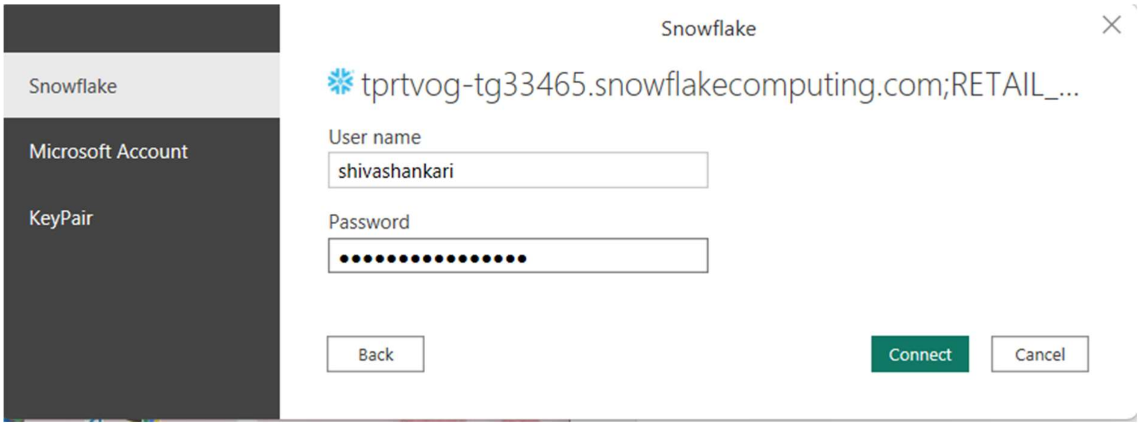
Snowflake

Server

TPRTVOG-TG33465.snowflakecomputing.com

Warehouse

RETAIL_WH



Snowflake

tprtvog-tg33465.snowflakecomputing.com;RETAIL_...

User name

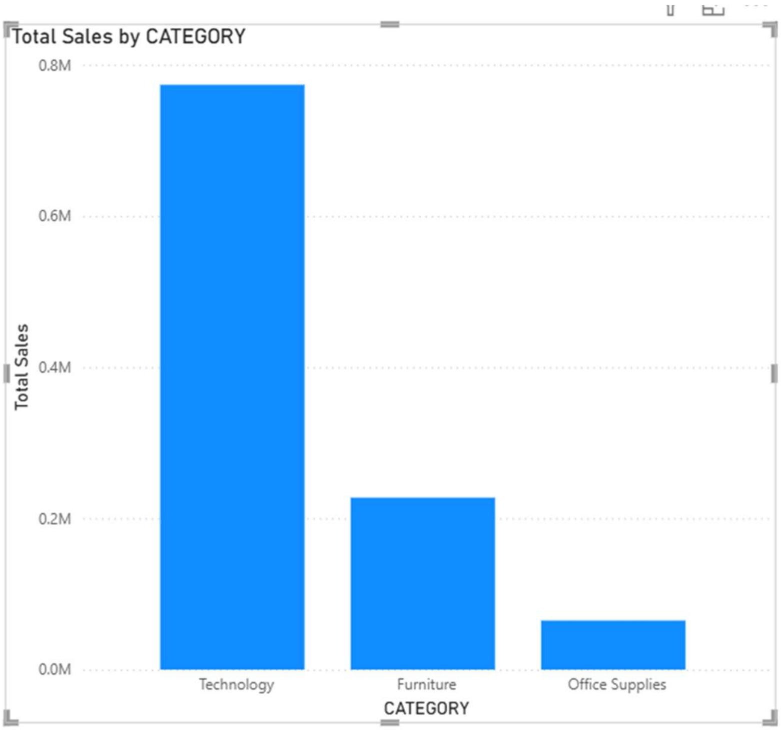
shivashankari

Password

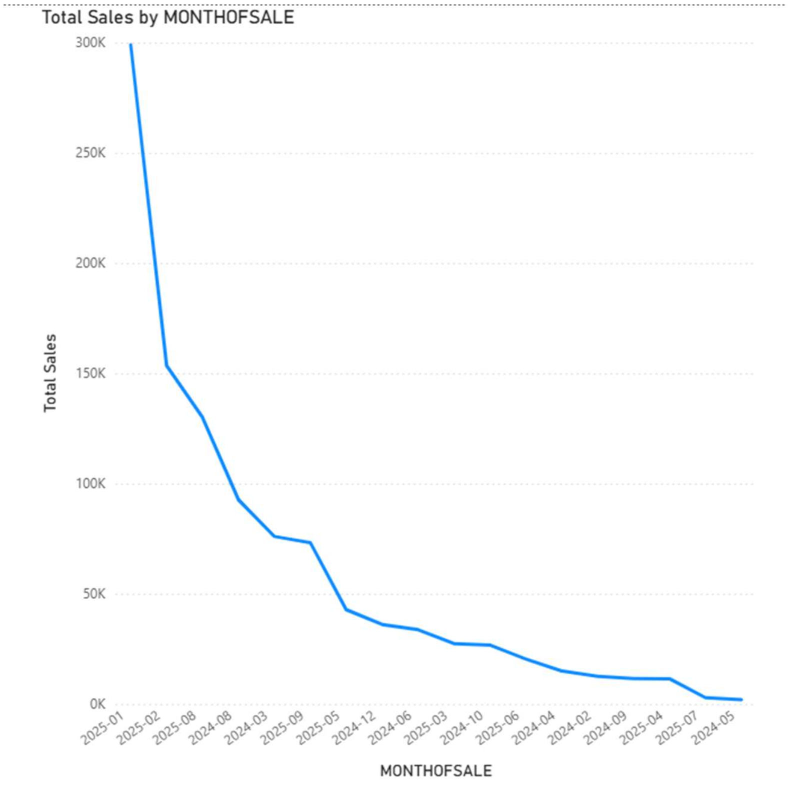
●●●●●●●●●●●●●●●●

- Selected table **RETAIL_SALES_MODELED** for data import.
- Built visuals using these measures:

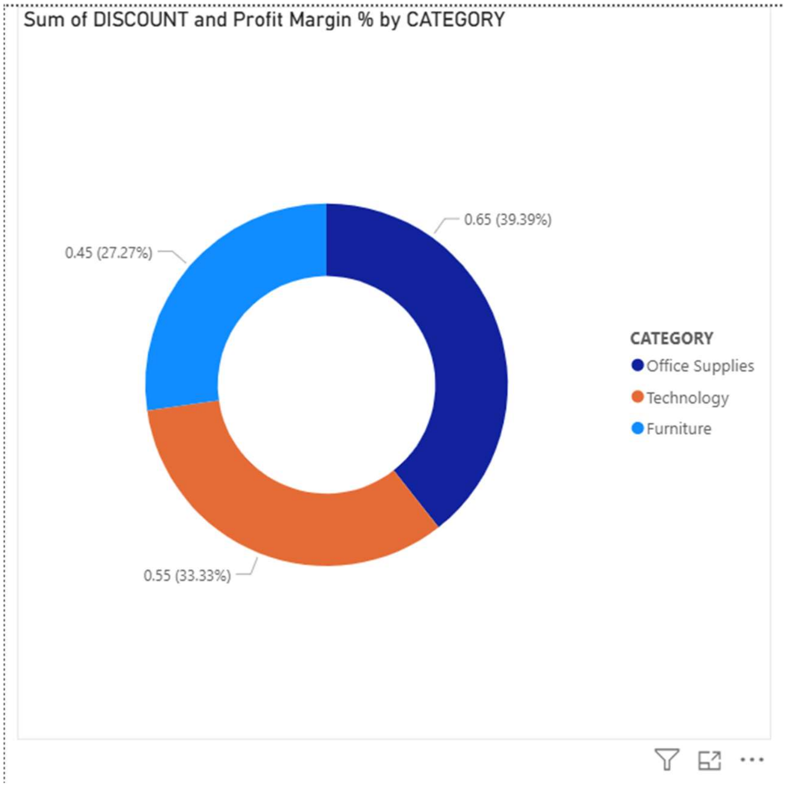
Column Chart: Sales by Category



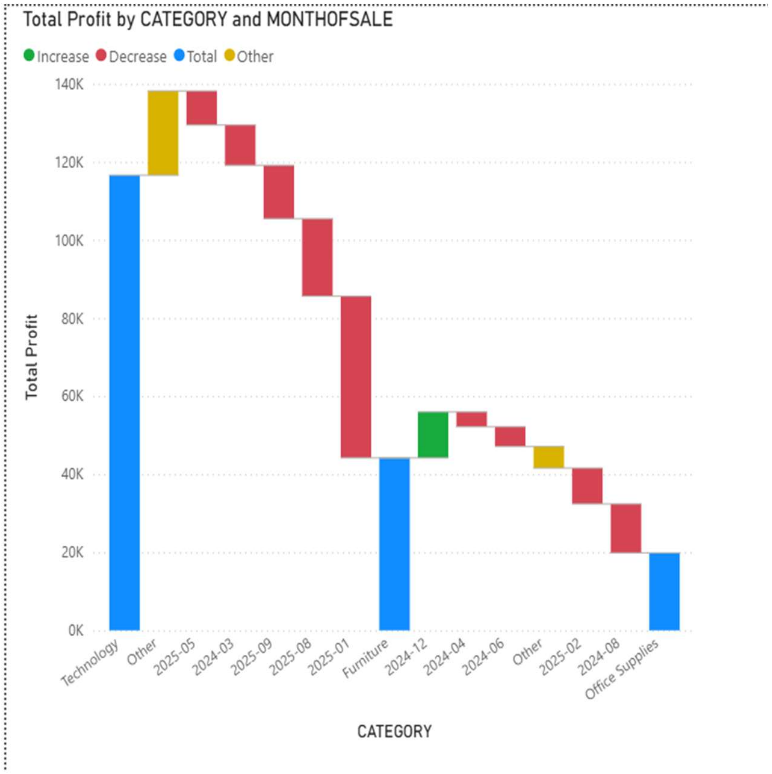
Line Chart: Monthly Sales Trend



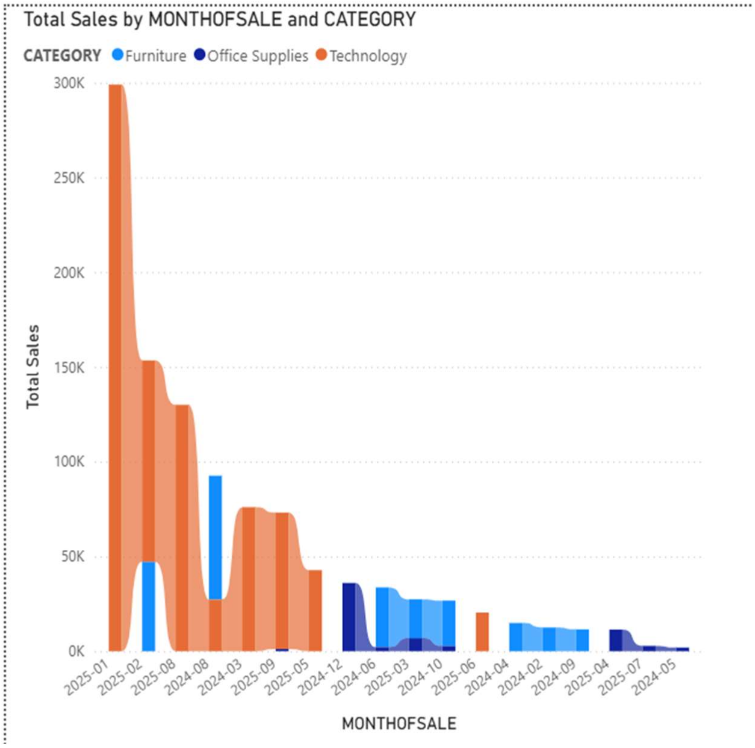
Donut Chart: Discount Impact



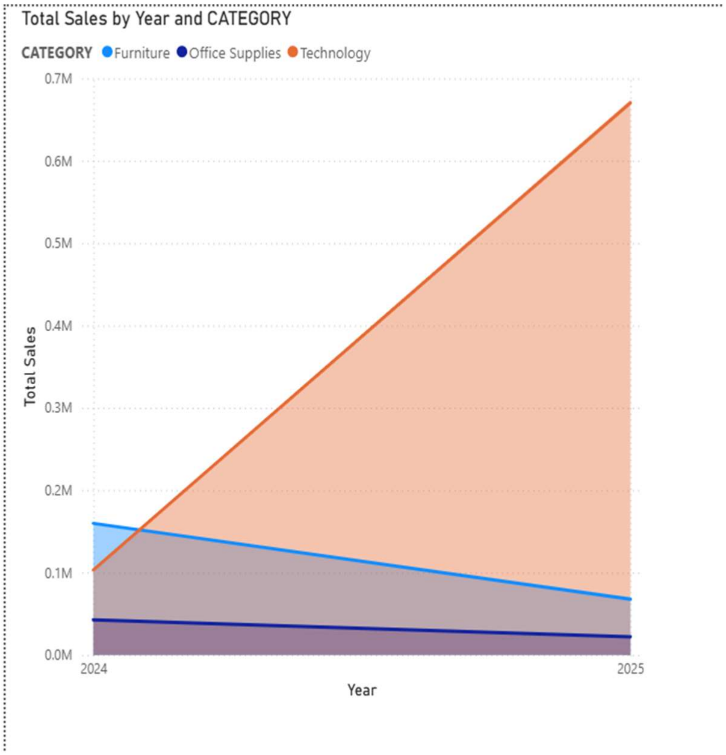
Waterfall Chart: Profit Breakdown



Ribbon Chart: Category rank changes over time



Area Chart : Show cumulative growth in sales or profit over time.


Total Sales by Year and CATEGORY
CATEGORY ● Furniture ● Office Supplies ● Technology
0.7M
0.6M
0.5M
0.4M
0.3M
0.2M
0.1M
0.0M
Total Sales
Year