Case Study: Azure → Snowflake with Snowpark, then Power BI

Bavatharani S – 22/10/2025

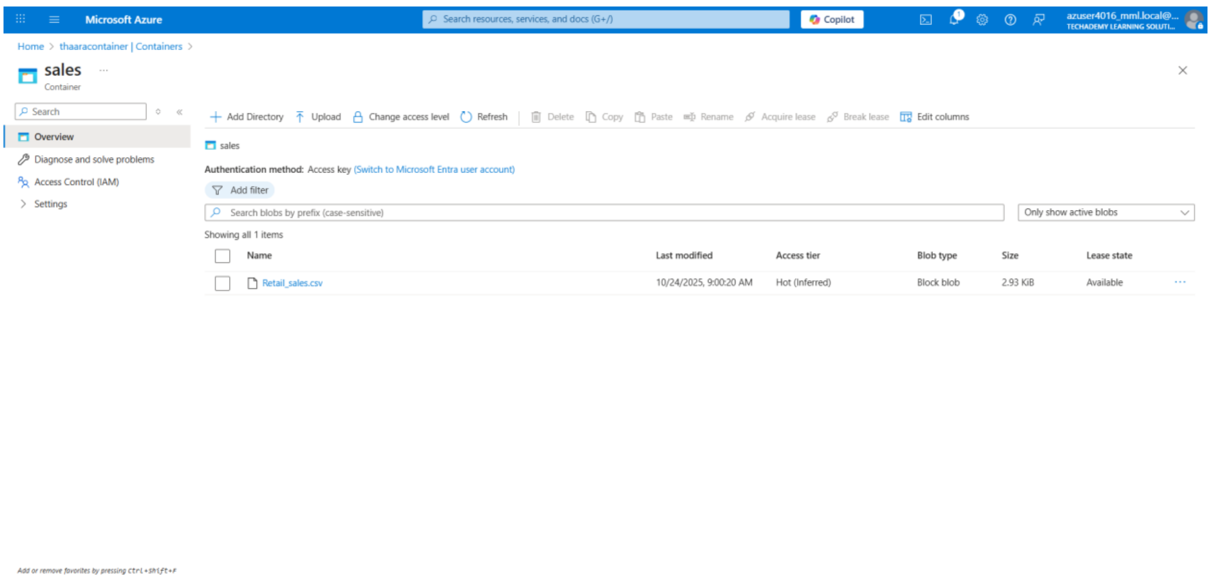
Scenario

You're the data engineer at ItTechGenie Retail. Sales teams drop monthly CSVs into an Azure Storage container. You must:

1. upload the CSV to Azure,
2. ingest it into Snowflake using Snowpark,
3. model it into proper database/schema/table, and
4. build a quick Power BI report for business users.

Steps:

Step 1: Upload CSV to Azure Blob Storage



sales
Container

sales

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.csv	10/24/2025, 9:00:20 AM	Hot (Inferred)	Block blob	2.93 KiB	Available

Step 2: Connect Azure to Snowflake (External Stage)

--Create Database

CREATE OR REPLACE DATABASE RETAIL;

USE DATABASE RETAIL;

--Create Schema

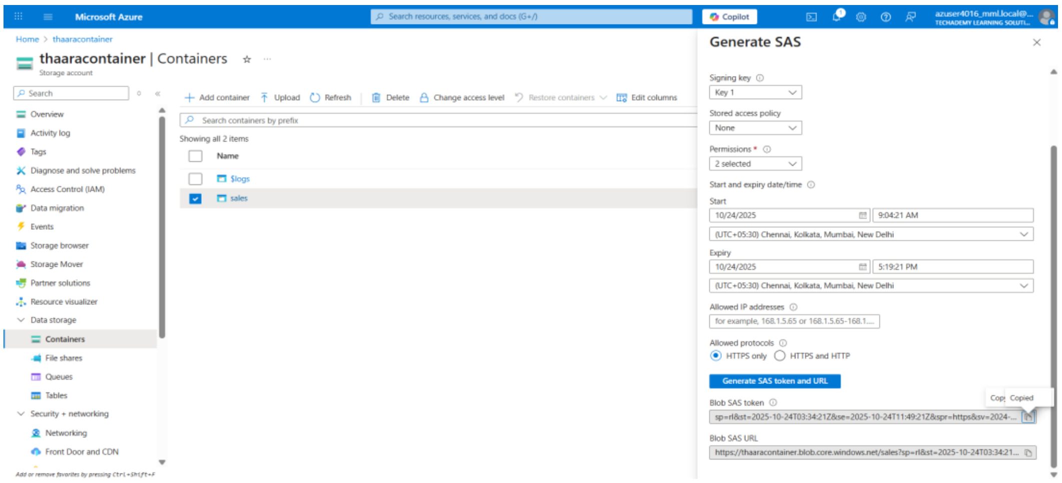
CREATE OR REPLACE SCHEMA RAW_SALES;

USE SCHEMA RAW_SALES;

--Create CSV File Format

CREATE OR REPLACE FILE FORMAT MY_CSV_FORMAT

```
TYPE = 'CSV'

FIELD_OPTIONALLY_ENCLOSED_BY = '"'

SKIP_HEADER = 1

FIELD_DELIMITER = ','

NULL_IF = ('NULL','');

--Create External Stage pointing to Azure Blob

CREATE OR REPLACE STAGE AZURE_SALES_STAGE

URL='azure://thaaracontainer.blob.core.windows.net/sales'

CREDENTIALS=(

AZURE_SAS_TOKEN='sv=2024-11-04&ss=bfqt&srt=sco&sp=rwdlacupiytfx&se=2025-10-23T17:28:58Z&st=2025-10-23T09:13:58Z&spr=https,http&sig=n0z1AuaUeygYYf7ZQf9AuM8TqVw3in8mQq%2FHdlOpC7o%3D'

)

FILE_FORMAT = MY_CSV_FORMAT;

--Verify Stage Contents

LIST @AZURE_SALES_STAGE;
```



Showing all 2 items

	Name
☐	$logs
✓	sales

Generate SAS

Signing key: Key 1

Stored access policy: None

Permissions *: 2 selected

Start and expiry date/time

Start: 10/24/2025 9:04:21 AM (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Expiry: 10/24/2025 5:19:21 PM (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses: for example, 168.1.5.65 or 168.1.5.65-168.1....

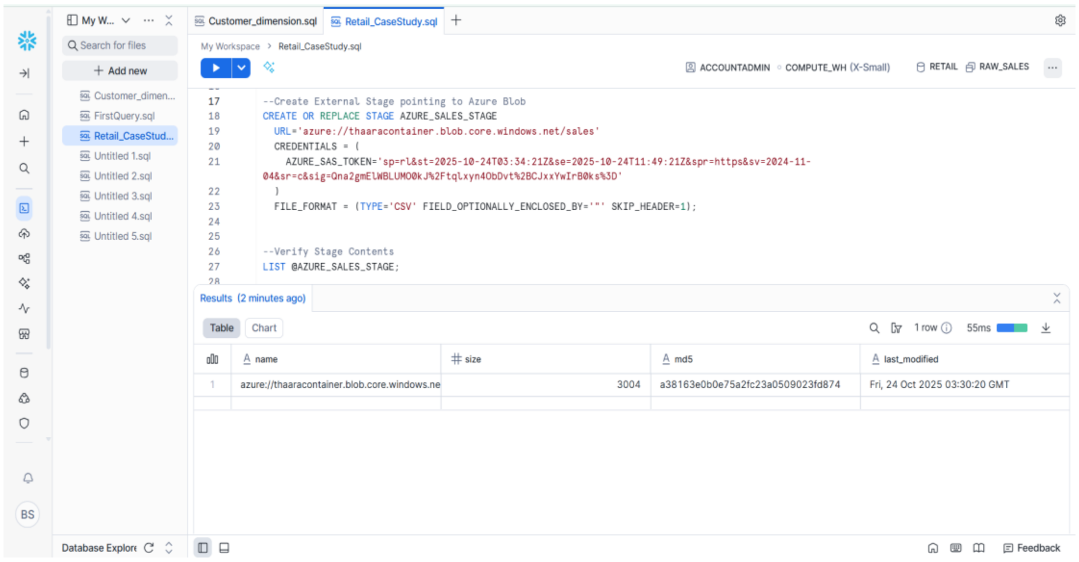
Allowed protocols: [✓ HTTPS only] HTTPS and HTTP

Generate SAS token and URL

Blob SAS token: sp=rl&st=2025-10-24T03:34:21Z&se=2025-10-24T11:49:21Z&spr=https&sv=2024-...

Copied

Blob SAS URL: https://thaaracontainer.blob.core.windows.net/sales?sp=rl&st=2025-10-24T03:34:21...



```sql
--Create External Stage pointing to Azure Blob
CREATE OR REPLACE STAGE AZURE_SALES_STAGE
  URL='azure://thaaracontainer.blob.core.windows.net/sales'
  CREDENTIALS = (
    AZURE_SAS_TOKEN='sp=rl&st=2025-10-24T03:34:21Z&se=2025-10-24T11:49:21Z&spr=https&sv=2024-11-04&sr=c&sig=Qna2gmElWBLUMO0kJ%2FtqLxyn4ObDvt%2BCJxxYwIrB0ks%3D'
  )
  FILE_FORMAT = (TYPE='CSV' FIELD_OPTIONALLY_ENCLOSED_BY='"' SKIP_HEADER=1);


--Verify Stage Contents
LIST @AZURE_SALES_STAGE;
```

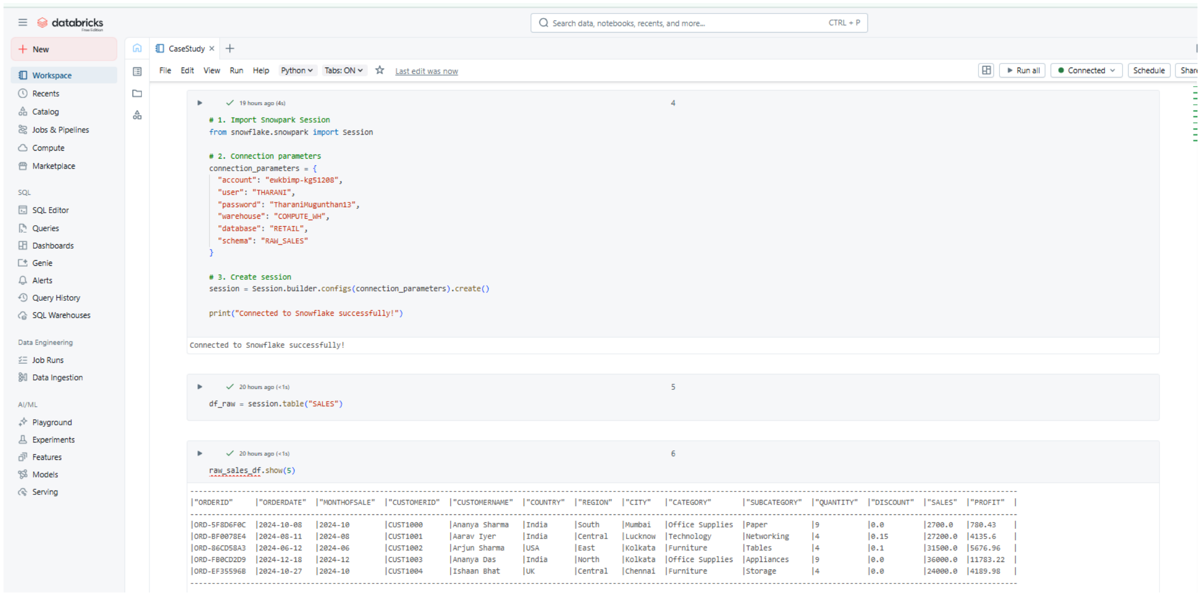
Results (2 minutes ago)

	name	size	md5	last_modified
1	azure://thaaracontainer.blob.core.windows.ne	3004	a38163e0b0e75a2fc23a0509023fd874	Fri, 24 Oct 2025 03:30:20 GMT

Step 3: Ingest Data Using Snowpark (Python)

```python
import pandas as pd

from snowflake.connector import connect

from snowflake.snowpark import Session

connection_parameters = {

  "account": "ewkbimp-kg51208",

  "user": "THARANI",

  "password": "TharaniMugunthan13",

  "warehouse": "COMPUTE_WH",

  "database": "RETAIL",

  "schema": "RAW_SALES"

}

session = Session.builder.configs(connection_parameters).create()
```

print("Connected to Snowflake successfully!")



```
# 1. Import Snowpark Session
from snowflake.snowpark import Session

# 2. Connection parameters
connection_parameters = {
  "account": "ewkbimp-kg51208",
  "user": "THARANI",
  "password": "TharaniMugunthan13",
  "warehouse": "COMPUTE_WH",
  "database": "RETAIL",
  "schema": "RAW_SALES"
}

# 3. Create session
session = Session.builder.configs(connection_parameters).create()

print("Connected to Snowflake successfully!")
```

```
Connected to Snowflake successfully!
```

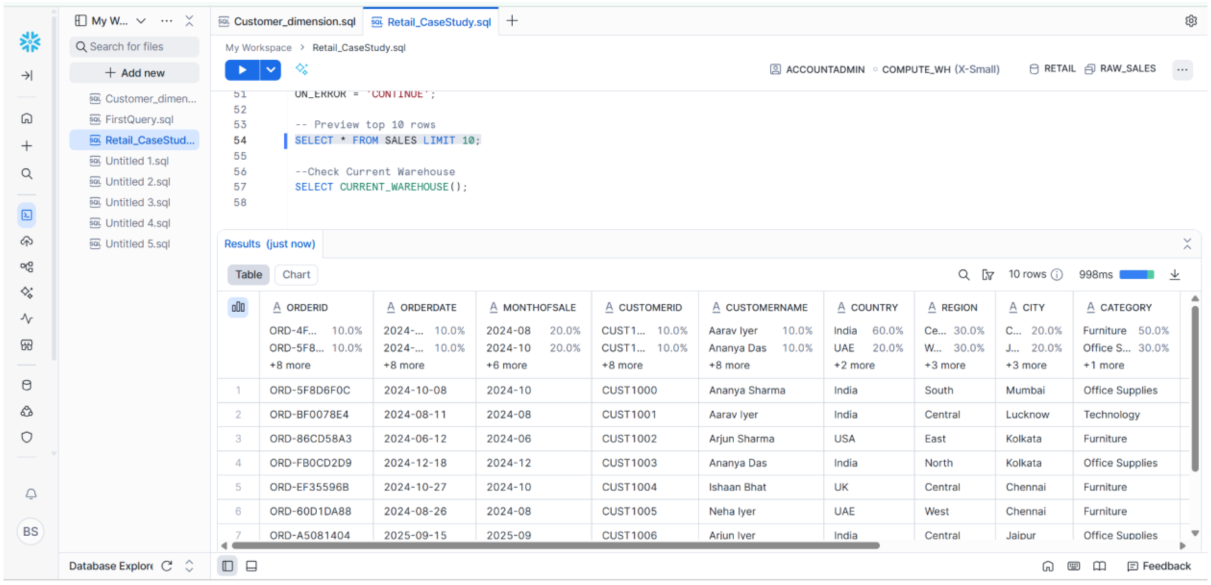
```
df_raw = session.table("SALES")
```

```
raw_sales_df.show(5)
```

```
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|"ORDERID"     |"ORDERDATE"  |"MONTHOFSALE"  |"CUSTOMERID"  |"CUSTOMERNAME"  |"COUNTRY"  |"REGION"  |"CITY"   |"CATEGORY"       |"SUBCATEGORY"  |"QUANTITY"  |"DISCOUNT"  |"SALES"  |"PROFIT"  |
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|ORD-5F8D6F0C  |2024-10-08   |2024-10        |CUST1000      |Ananya Sharma   |India      |South     |Mumbai   |Office Supplies  |Paper          |9           |0.0         |2700.0   |780.43    |
|ORD-BF0078E4  |2024-08-11   |2024-08        |CUST1001      |Aarav Iyer      |India      |Central   |Lucknow  |Technology       |Networking     |4           |0.15        |27200.0  |4135.6    |
|ORD-86CD58A3  |2024-06-12   |2024-06        |CUST1002      |Arjun Sharma    |USA        |East      |Kolkata  |Furniture        |Tables         |4           |0.1         |31500.0  |5676.96   |
|ORD-FB0CD2D9  |2024-12-18   |2024-12        |CUST1003      |Ananya Das      |India      |North     |Kolkata  |Office Supplies  |Appliances     |9           |0.0         |36000.0  |11783.22  |
|ORD-EF35596B  |2024-10-27   |2024-10        |CUST1004      |Ishaan Bhat     |UK         |Central   |Chennai  |Furniture        |Storage        |4           |0.0         |24000.0  |4189.98   |
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Step 4: Create Database, Schema & Model Tables in Snowflake

```
--Create Raw Sales Table

CREATE OR REPLACE TABLE SALES (

  OrderID STRING,

  OrderDate STRING,

  MonthOfSale STRING,

  CustomerID STRING,

  CustomerName STRING,

  Country STRING,

  Region STRING,

  City STRING,

  Category STRING,

  Subcategory STRING,

  Quantity INT,

  Discount FLOAT,

  Sales FLOAT,

  Profit FLOAT

);
```



```
ON_ERROR = 'CONTINUE';

-- Preview top 10 rows
SELECT * FROM SALES LIMIT 10;

--Check Current Warehouse
SELECT CURRENT_WAREHOUSE();
```

Results (just now) — 10 rows, 998ms

	ORDERID	ORDERDATE	MONTHOFSALE	CUSTOMERID	CUSTOMERNAME	COUNTRY	REGION	CITY	CATEGORY
	ORD-4F... 10.0% ORD-5F8... 10.0% +8 more	2024-... 10.0% 2024-... 10.0% +8 more	2024-08 20.0% 2024-10 20.0% +6 more	CUST1... 10.0% CUST1... 10.0% +8 more	Aarav Iyer 10.0% Ananya Das 10.0% +8 more	India 60.0% UAE 20.0% +2 more	Ce... 30.0% W... 30.0% +3 more	C... 20.0% J... 20.0% +3 more	Furniture 50.0% Office S... 30.0% +1 more
1	ORD-5F8D6F0C	2024-10-08	2024-10	CUST1000	Ananya Sharma	India	South	Mumbai	Office Supplies
2	ORD-BF0078E4	2024-08-11	2024-08	CUST1001	Aarav Iyer	India	Central	Lucknow	Technology
3	ORD-86CD58A3	2024-06-12	2024-06	CUST1002	Arjun Sharma	USA	East	Kolkata	Furniture
4	ORD-FB0CD2D9	2024-12-18	2024-12	CUST1003	Ananya Das	India	North	Kolkata	Office Supplies
5	ORD-EF35596B	2024-10-27	2024-10	CUST1004	Ishaan Bhat	UK	Central	Chennai	Furniture
6	ORD-60D1DA88	2024-08-26	2024-08	CUST1005	Neha Iyer	UAE	West	Chennai	Furniture
7	ORD-A5081404	2025-09-15	2025-09	CUST1006	Arjun Iyer	India	Central	Jaipur	Office Supplies

Step 5: Load the Data

--Load CSV Data from Stage to Table

COPY INTO SALES

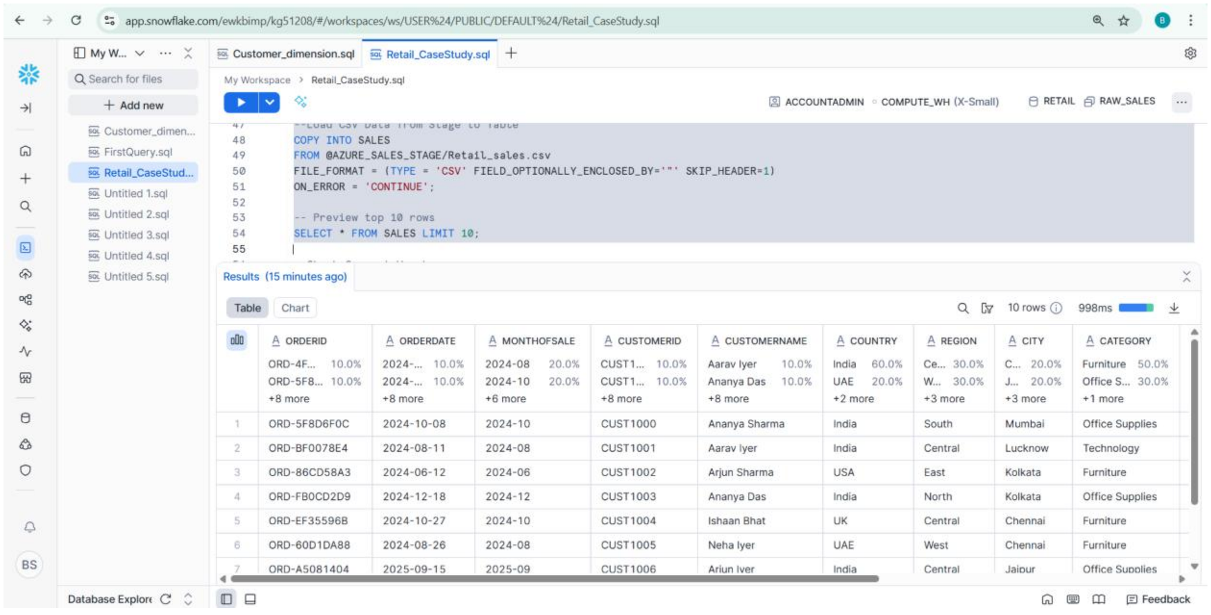
FROM @AZURE_SALES_STAGE/Retail_sales.csv

FILE_FORMAT = (TYPE = 'CSV' FIELD_OPTIONALLY_ENCLOSED_BY='"' SKIP_HEADER=1)

ON_ERROR = 'CONTINUE';

-- Preview top 10 rows

SELECT * FROM SALES LIMIT 10;



app.snowflake.com/ewkbimp/kg51208/#/workspaces/ws/USER%24/PUBLIC/DEFAULT%24/Retail_CaseStudy.sql

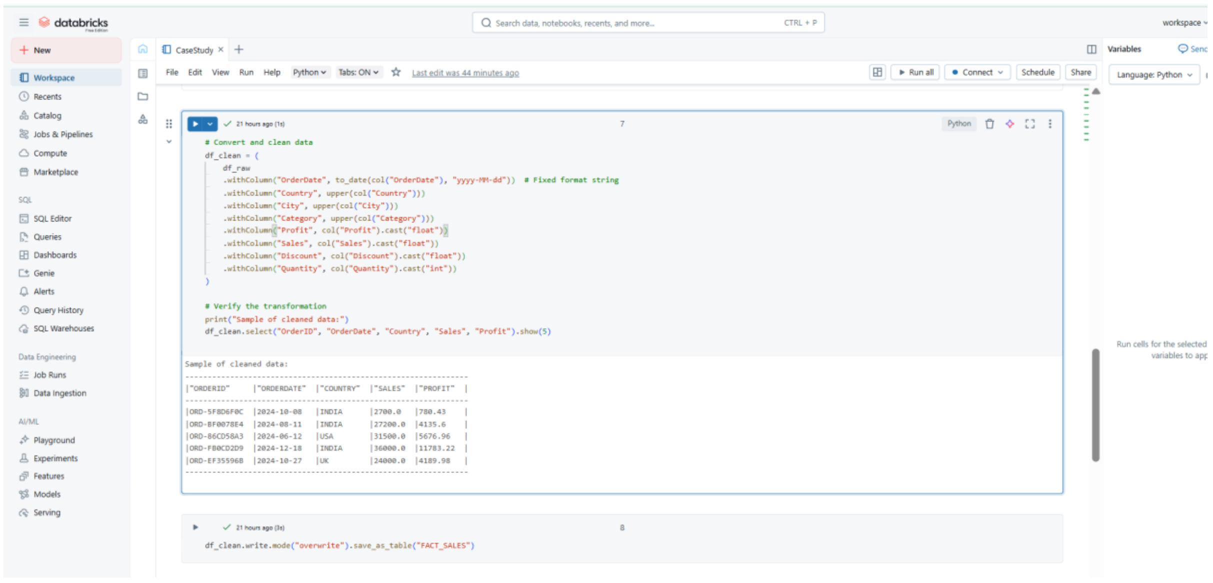
```
COPY INTO SALES
FROM @AZURE_SALES_STAGE/Retail_sales.csv
FILE_FORMAT = (TYPE = 'CSV' FIELD_OPTIONALLY_ENCLOSED_BY='"' SKIP_HEADER=1)
ON_ERROR = 'CONTINUE';

-- Preview top 10 rows
SELECT * FROM SALES LIMIT 10;
```

Results (15 minutes ago) — 10 rows, 998ms

	ORDERID	ORDERDATE	MONTHOFSALE	CUSTOMERID	CUSTOMERNAME	COUNTRY	REGION	CITY	CATEGORY
	ORD-4F... 10.0% ORD-5F8... 10.0% +8 more	2024-... 10.0% 2024-... 10.0% +8 more	2024-08 20.0% 2024-10 20.0% +6 more	CUST1... 10.0% CUST1... 10.0% +8 more	Aarav Iyer 10.0% Ananya Das 10.0% +8 more	India 60.0% UAE 20.0% +2 more	Ce... 30.0% W... 30.0% +3 more	C... 20.0% J... 20.0% +3 more	Furniture 50.0% Office S... 30.0% +1 more
1	ORD-5F8D6F0C	2024-10-08	2024-10	CUST1000	Ananya Sharma	India	South	Mumbai	Office Supplies
2	ORD-BF0078E4	2024-08-11	2024-08	CUST1001	Aarav Iyer	India	Central	Lucknow	Technology
3	ORD-86CD58A3	2024-06-12	2024-06	CUST1002	Arjun Sharma	USA	East	Kolkata	Furniture
4	ORD-FB0CD2D9	2024-12-18	2024-12	CUST1003	Ananya Das	India	North	Kolkata	Office Supplies
5	ORD-EF35596B	2024-10-27	2024-10	CUST1004	Ishaan Bhat	UK	Central	Chennai	Furniture
6	ORD-60D1DA88	2024-08-26	2024-08	CUST1005	Neha Iyer	UAE	West	Chennai	Furniture
7	ORD-A5081404	2025-09-15	2025-09	CUST1006	Arjun Iyer	India	Central	Jaipur	Office Supplies

Step 6: Data Cleaning and Transformation

```
# Convert and clean data
df_clean = (
    df_raw
    .withColumn("OrderDate", to_date(col("OrderDate"), "yyyy-MM-dd"))  # Fixed format string
    .withColumn("Country", upper(col("Country")))
    .withColumn("City", upper(col("City")))
    .withColumn("Category", upper(col("Category")))
    .withColumn("Profit", col("Profit").cast("float"))
    .withColumn("Sales", col("Sales").cast("float"))
    .withColumn("Discount", col("Discount").cast("float"))
    .withColumn("Quantity", col("Quantity").cast("int"))
)
# Verify the transformation
print("Sample of cleaned data:")
df_clean.select("OrderID", "OrderDate", "Country", "Sales", "Profit").show(5)
```



```
# Convert and clean data
df_clean = (
    df_raw
    .withColumn("OrderDate", to_date(col("OrderDate"), "yyyy-MM-dd"))  # Fixed format string
    .withColumn("Country", upper(col("Country")))
    .withColumn("City", upper(col("City")))
    .withColumn("Category", upper(col("Category")))
    .withColumn("Profit", col("Profit").cast("float"))
    .withColumn("Sales", col("Sales").cast("float"))
    .withColumn("Discount", col("Discount").cast("float"))
    .withColumn("Quantity", col("Quantity").cast("int"))
)

# Verify the transformation
print("Sample of cleaned data:")
df_clean.select("OrderID", "OrderDate", "Country", "Sales", "Profit").show(5)
```

```
Sample of cleaned data:
---------------------------------------------------------------
|"ORDERID"     |"ORDERDATE"  |"COUNTRY"   |"SALES"  |"PROFIT"  |
---------------------------------------------------------------
|ORD-5F8D6F0C  |2024-10-08   |INDIA       |2700.0   |780.43    |
|ORD-BF0078E4  |2024-08-11   |INDIA       |27200.0  |4135.6    |
|ORD-86CD58A3  |2024-06-12   |USA         |31500.0  |5676.96   |
|ORD-FB0CD2D9  |2024-12-18   |INDIA       |36000.0  |11783.22  |
|ORD-EF35596B  |2024-10-27   |UK          |24000.0  |4189.98   |
---------------------------------------------------------------
```

```
df_clean.write.mode("overwrite").save_as_table("FACT_SALES")
```

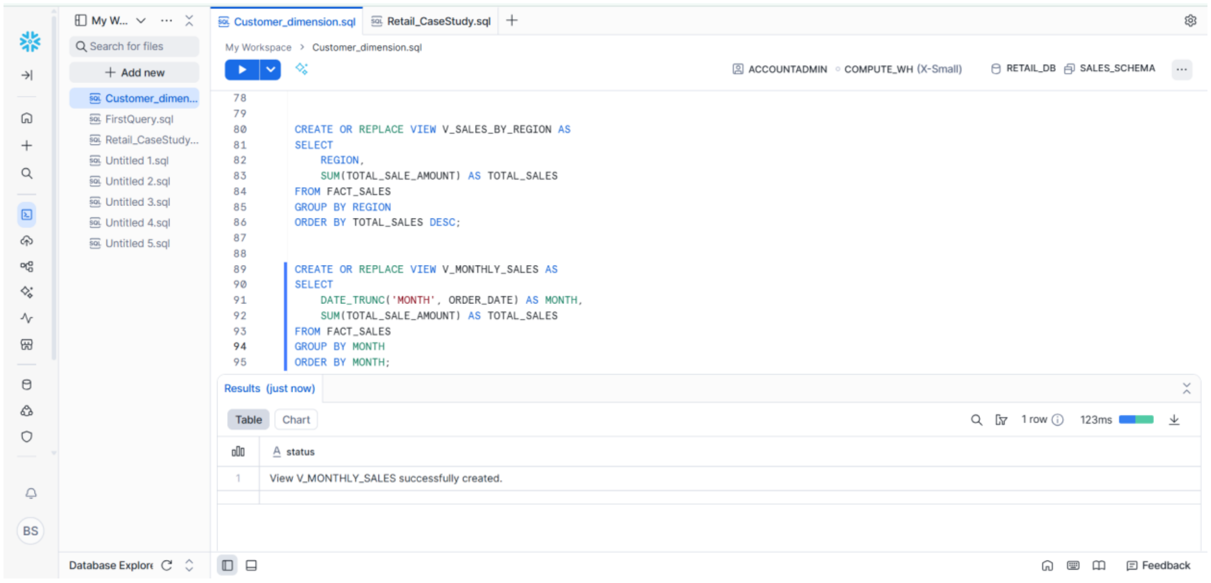
Step 7: Dimensions Model

```
CREATE OR REPLACE DATABASE RETAIL_DB;
USE DATABASE RETAIL_DB;
```

```
CREATE OR REPLACE SCHEMA SALES_SCHEMA;

USE SCHEMA SALES_SCHEMA;


CREATE OR REPLACE TABLE RETAIL_SALES_RAW (

    ORDER_ID STRING,

    DATE DATE,

    CUSTOMER_ID STRING,

    PRODUCT_CATEGORY STRING,

    PRODUCT_NAME STRING,

    QUANTITY NUMBER,

    UNIT_PRICE NUMBER,

    TOTAL_AMOUNT NUMBER,

    REGION STRING

);


CREATE OR REPLACE TABLE RETAIL_SALES_CLEANED AS

SELECT

    ORDER_ID,

    TO_DATE(DATE) AS ORDER_DATE,

    CUSTOMER_ID,

    INITCAP(PRODUCT_CATEGORY) AS PRODUCT_CATEGORY,

    INITCAP(PRODUCT_NAME) AS PRODUCT_NAME,

    QUANTITY,

    UNIT_PRICE,

    QUANTITY * UNIT_PRICE AS TOTAL_SALE_AMOUNT,

    UPPER(REGION) AS REGION

FROM RETAIL_SALES_RAW

WHERE ORDER_ID IS NOT NULL

  AND QUANTITY > 0

  AND UNIT_PRICE > 0;
```

```
CREATE OR REPLACE TABLE DIM_PRODUCT AS
SELECT DISTINCT
    PRODUCT_CATEGORY,
    PRODUCT_NAME
FROM RETAIL_SALES_CLEANED;

CREATE OR REPLACE TABLE DIM_CUSTOMER AS
SELECT DISTINCT
    CUSTOMER_ID
FROM RETAIL_SALES_CLEANED;

CREATE OR REPLACE TABLE DIM_REGION AS
SELECT DISTINCT
    REGION
FROM RETAIL_SALES_CLEANED;

CREATE OR REPLACE TABLE FACT_SALES AS
SELECT
    ORDER_ID,
    ORDER_DATE,
    CUSTOMER_ID,
    PRODUCT_NAME,
    PRODUCT_CATEGORY,
    REGION,
    QUANTITY,
    UNIT_PRICE,
    TOTAL_SALE_AMOUNT
FROM RETAIL_SALES_CLEANED;

CREATE OR REPLACE VIEW V_SALES_BY_CATEGORY AS
SELECT
```

```
    PRODUCT_CATEGORY,
    SUM(TOTAL_SALE_AMOUNT) AS TOTAL_SALES,
    SUM(QUANTITY) AS TOTAL_QUANTITY
FROM FACT_SALES
GROUP BY PRODUCT_CATEGORY
ORDER BY TOTAL_SALES DESC;


CREATE OR REPLACE VIEW V_SALES_BY_REGION AS
SELECT
    REGION,
    SUM(TOTAL_SALE_AMOUNT) AS TOTAL_SALES
FROM FACT_SALES
GROUP BY REGION
ORDER BY TOTAL_SALES DESC;

CREATE OR REPLACE VIEW V_MONTHLY_SALES AS
SELECT
    DATE_TRUNC('MONTH', ORDER_DATE) AS MONTH,
    SUM(TOTAL_SALE_AMOUNT) AS TOTAL_SALES
FROM FACT_SALES
GROUP BY MONTH
ORDER BY MONTH;
```

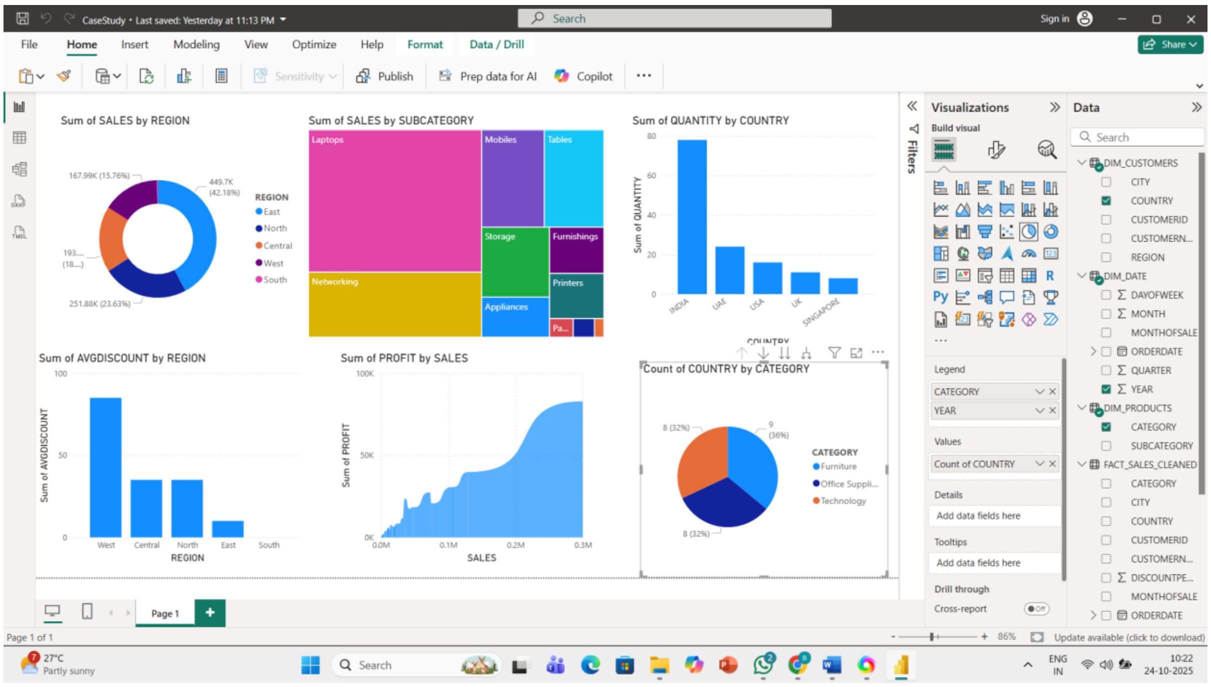


```
CREATE OR REPLACE VIEW V_SALES_BY_REGION AS
SELECT
    REGION,
    SUM(TOTAL_SALE_AMOUNT) AS TOTAL_SALES
FROM FACT_SALES
GROUP BY REGION
ORDER BY TOTAL_SALES DESC;


CREATE OR REPLACE VIEW V_MONTHLY_SALES AS
SELECT
    DATE_TRUNC('MONTH', ORDER_DATE) AS MONTH,
    SUM(TOTAL_SALE_AMOUNT) AS TOTAL_SALES
FROM FACT_SALES
GROUP BY MONTH
ORDER BY MONTH;
```

Results (just now)

	status
1	View V_MONTHLY_SALES successfully created.

Step 8: Power BI Visualization



CaseStudy • Last saved: Yesterday at 11:13 PM
Sum of SALES by REGION
Sum of SALES by SUBCATEGORY
Sum of QUANTITY by COUNTRY
Sum of AVGDISCOUNT by REGION
Sum of PROFIT by SALES
Count of COUNTRY by CATEGORY
Visualizations
Data
Page 1