**Case Study**

**Case Study: Azure to Snowflake with Snowpark, then Power BI**

**Scenario**

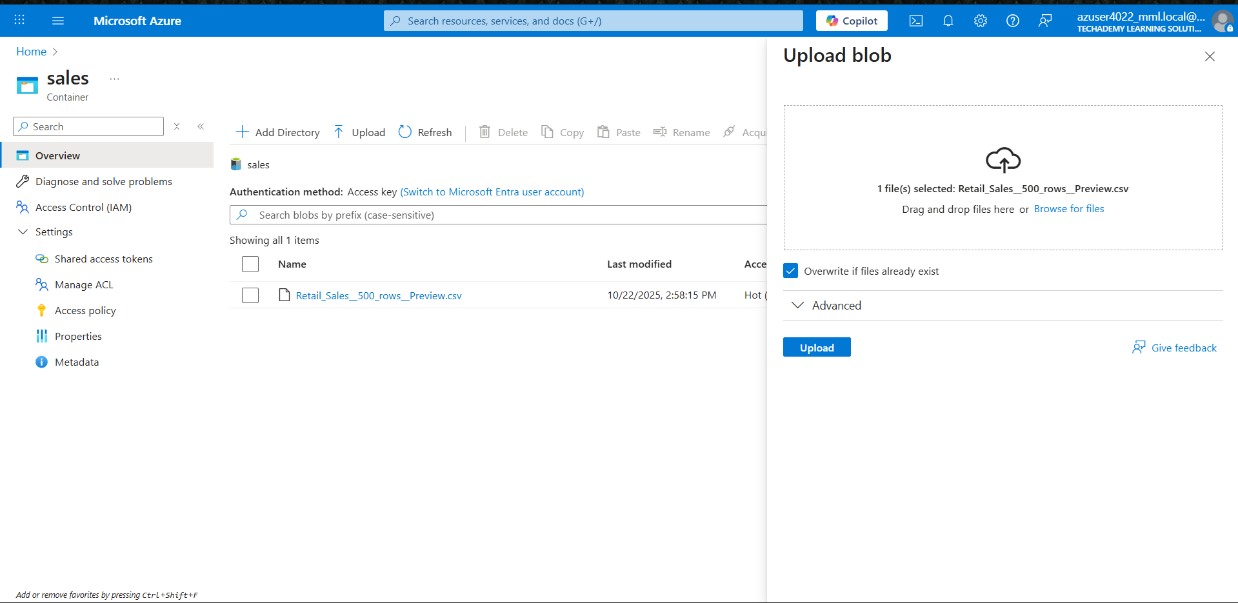
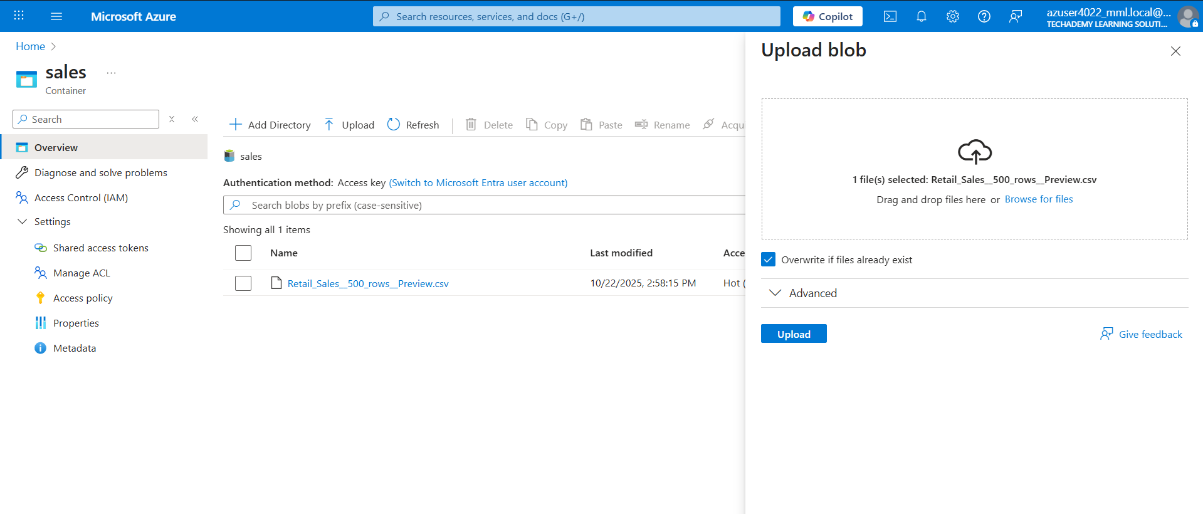
You’re the data engineer at ItTechGenie Retail.  
The sales team uploads monthly transaction CSVs to an Azure Storage container, which must be ingested into Snowflake for reporting and analytics.  
Your goal is to automate the entire data flow — from Azure upload to Power BI dashboard — ensuring accuracy, scalability, and efficiency.

**1. Upload the CSV to Azure Blob Container**

The monthly CSV files are stored securely in **Azure Blob Storage**, which acts as a centralized data lake for incoming sales data.

projectsamp.blob.core.windows.net/sales/

**File:**  
Retail\_Sales\_\_500\_rows\_\_Preview.csv



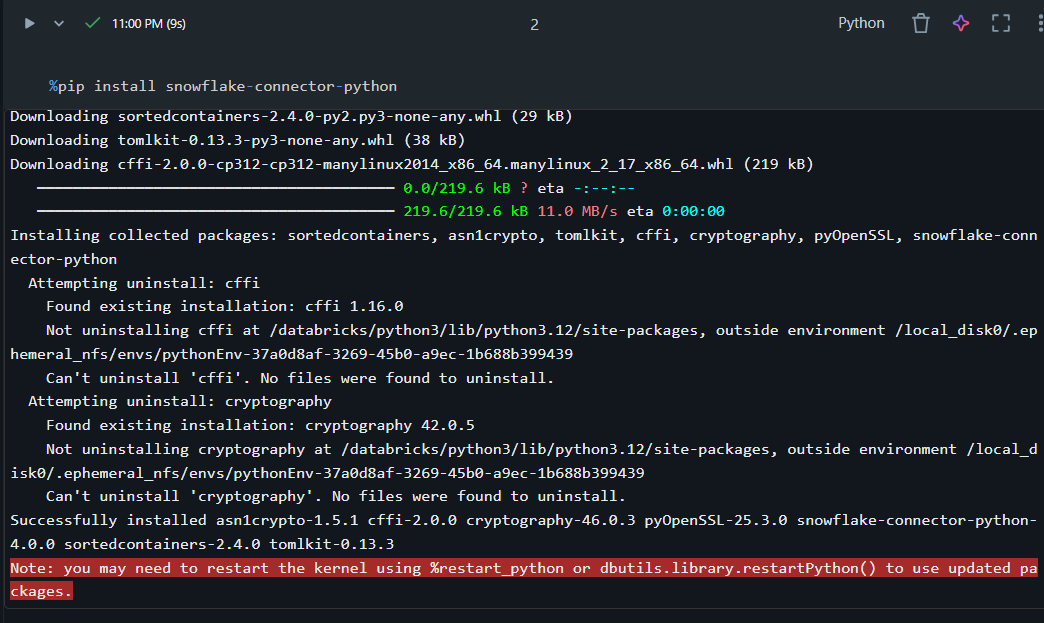
**2. Ingest into Snowflake as a Table Using Snowpark (via Databricks)**

Once the CSV is uploaded, it is read into Databricks using Pandas and then written to Snowflake using the Snowpark connector.  
This approach allows transformation flexibility within Spark while leveraging Snowflake’s scalability.

**Installing Required Libraries**

**%pip install snowflake-connector-python**

**%restart\_python**

This installs the Snowflake Python connector required for data transfer and restarts the Python runtime.

**Reading SAS URL Using Pandas**

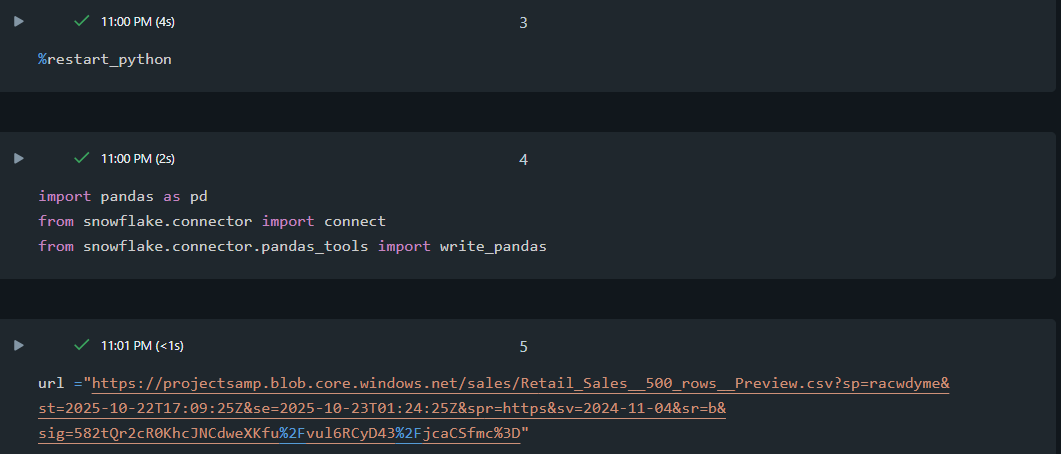
The SAS URL (Shared Access Signature) allows secure, time-bound access to the Azure Blob file without exposing credentials.

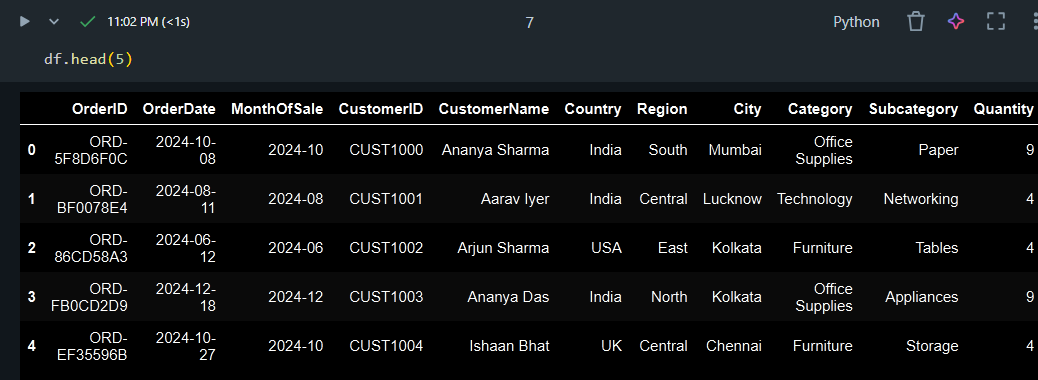
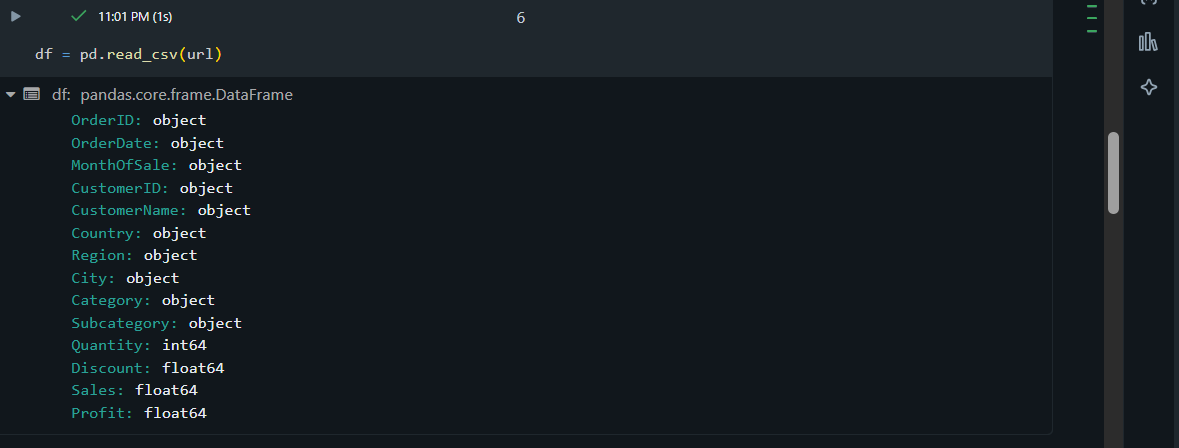
**import pandas as pd**

**url = "https://projectsamp.blob.core.windows.net/sales/Retail\_Sales\_\_500\_rows\_\_Preview.csv?sp=racwdyme&..."**

**df = pd.read\_csv(url)**

**df.head(5)**





**Converting Pandas DataFrame → Spark DataFrame**

Converting to Spark DataFrame enables distributed data processing and seamless integration with Snowflake using Snowpark.

**from pyspark.sql import SparkSession**

**spark = SparkSession.builder.getOrCreate()**

**df\_spark = spark.createDataFrame(df)**

**Snowflake Connection Configuration**

Connection parameters define the target Snowflake account, warehouse, and schema where the table will reside.

**sfOptions = {**

**"sfURL": "XAPNZGA-KTA45095.snowflakecomputing.com",**

**"sfDatabase": "PRACTICE",**

**"sfSchema": "PUBLIC",**

**"sfWarehouse": "COMPUTE\_WH",**

**"sfRole": "ACCOUNTADMIN",**

**"sfUser": "Priyesh",**

**"sfPassword": "Priyeshwar2664"**

**}**



**Writing to Snowflake Table**

This command writes the Spark DataFrame into Snowflake as the table RETAIL\_SALES.  
Using .mode("overwrite") ensures that the table refreshes with each new upload cycle.

**(df\_spark.write**

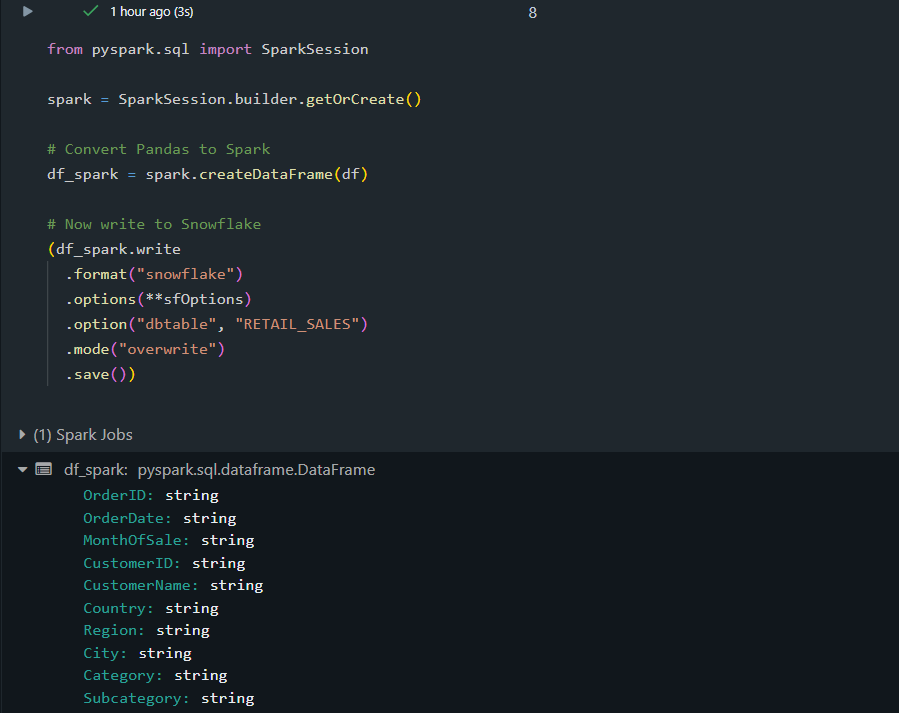
**.format("snowflake")**

**.options(\*\*sfOptions)**

**.option("dbtable", "RETAIL\_SALES")**

**.mode("overwrite")**

**.save())**

****

**3. Modeling into Proper Database / Schema / Table**

Data modeling defines how raw data is structured for analysis.  
Here, we ensure columns, data types, and relationships match analytical requirements.

Performed from **VS Code**, connected to Snowflake using the official Snowflake SQL extension.

**Table Definition Example**

CREATE OR REPLACE TABLE RETAIL\_SALES (

ORDER\_ID STRING,

ORDER\_DATE DATE,

MONTH\_OF\_SALE STRING,

CUSTOMER\_ID STRING,

CUSTOMER\_NAME STRING,

COUNTRY STRING,

REGION STRING,

CITY STRING,

CATEGORY STRING,

SUBCATEGORY STRING,

QUANTITY INT,

DISCOUNT FLOAT,

SALES FLOAT,

PROFIT FLOAT

);

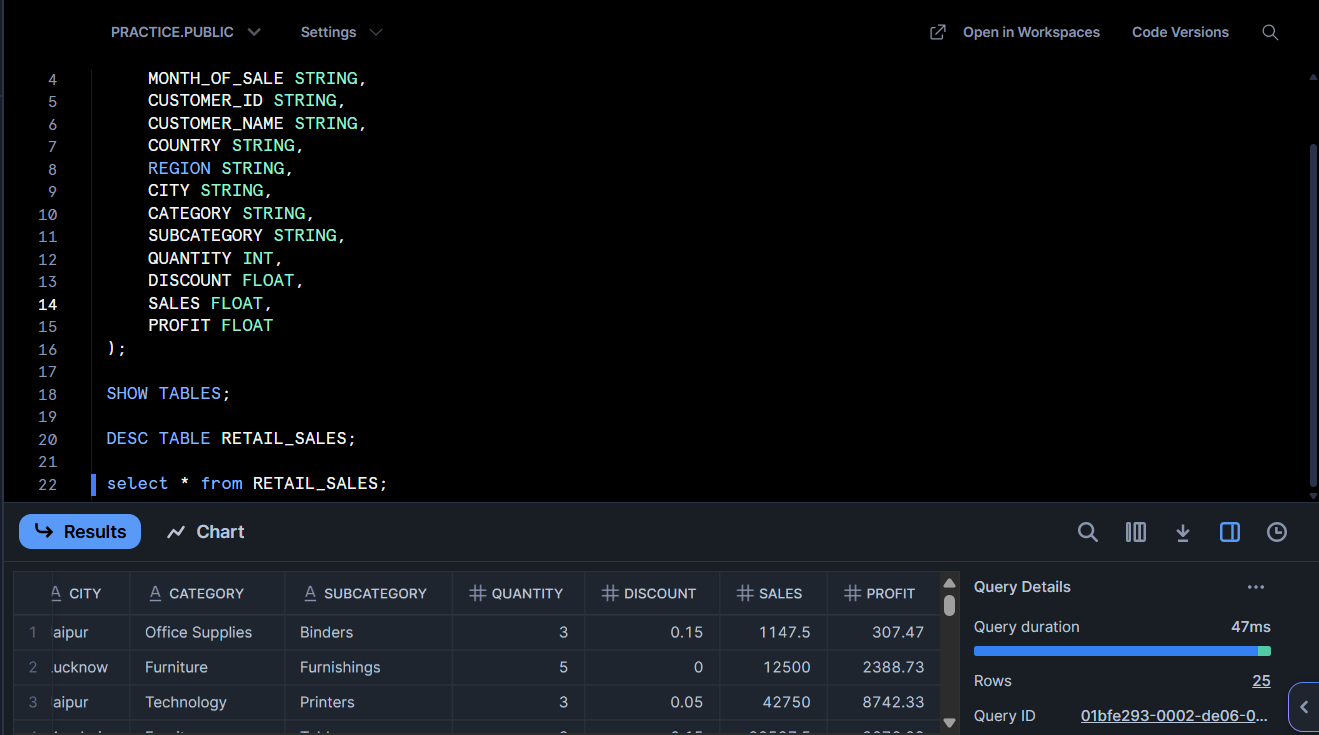
**Code:**

SHOW TABLES;

**Output:**  
Lists all available tables, including the newly created RETAIL\_SALES table.

**Code:**

DESC TABLE RETAIL\_SALES;



**Output:**  
Displays column definitions, data types, and constraints for verification.

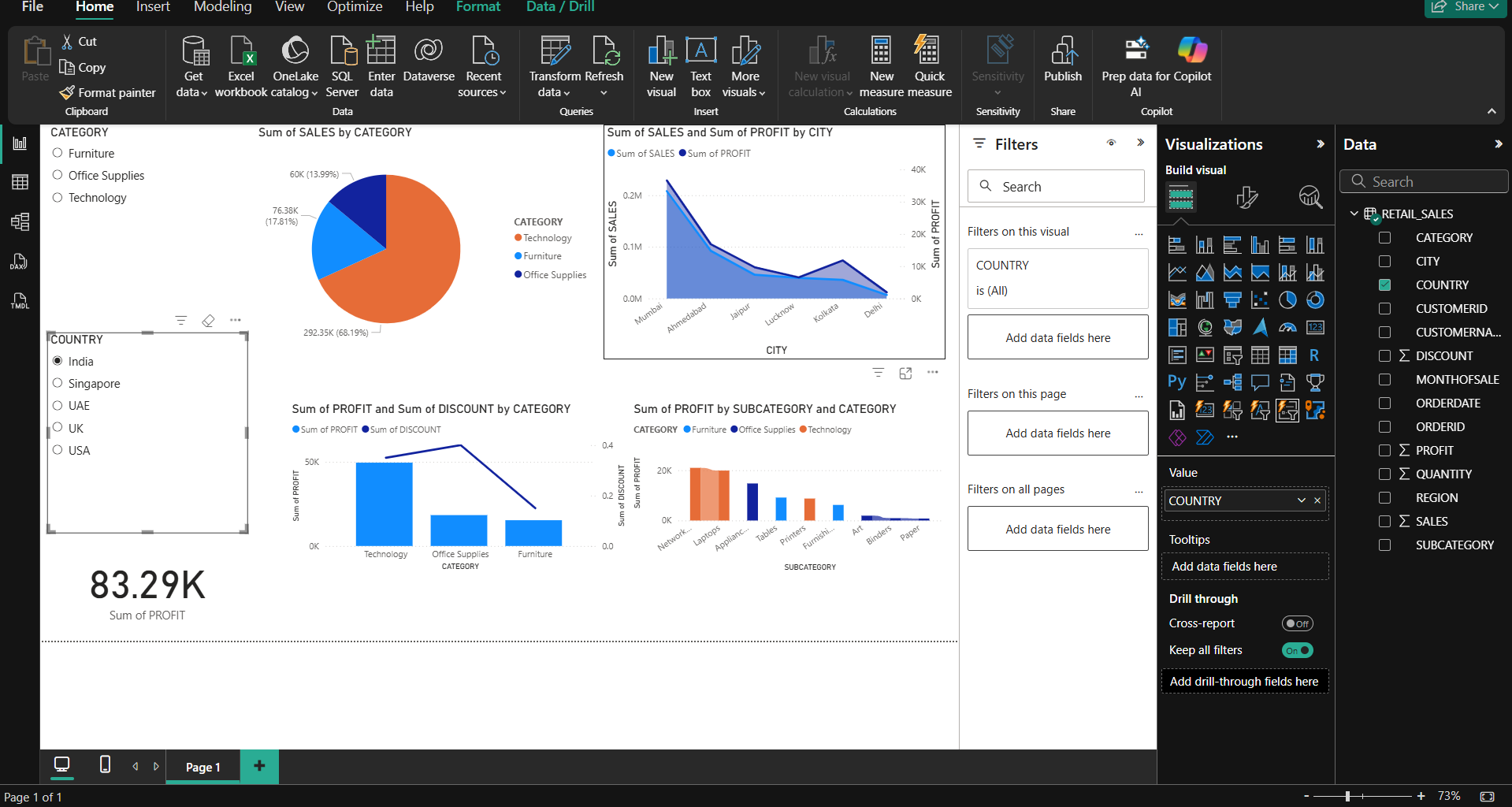
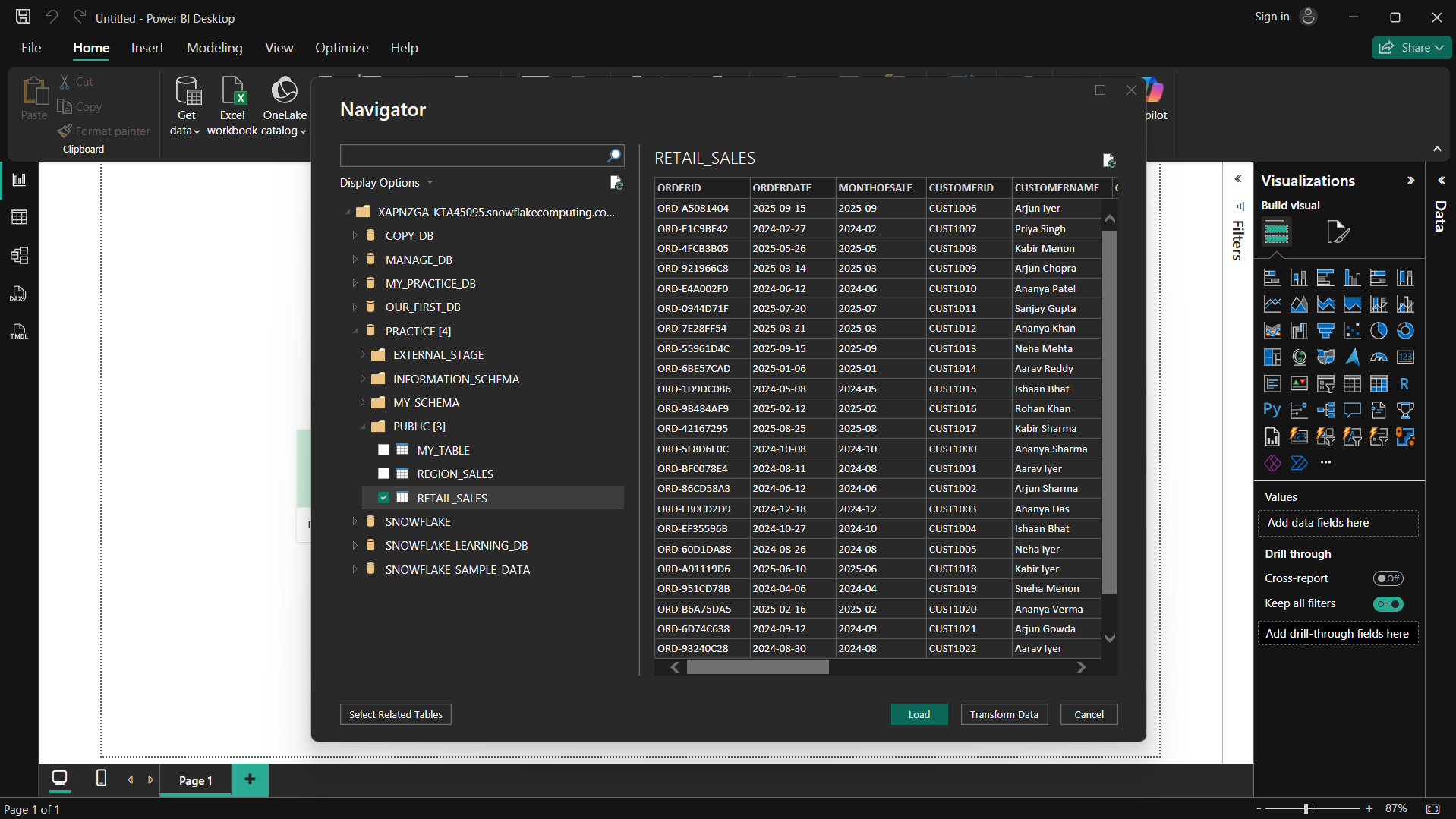
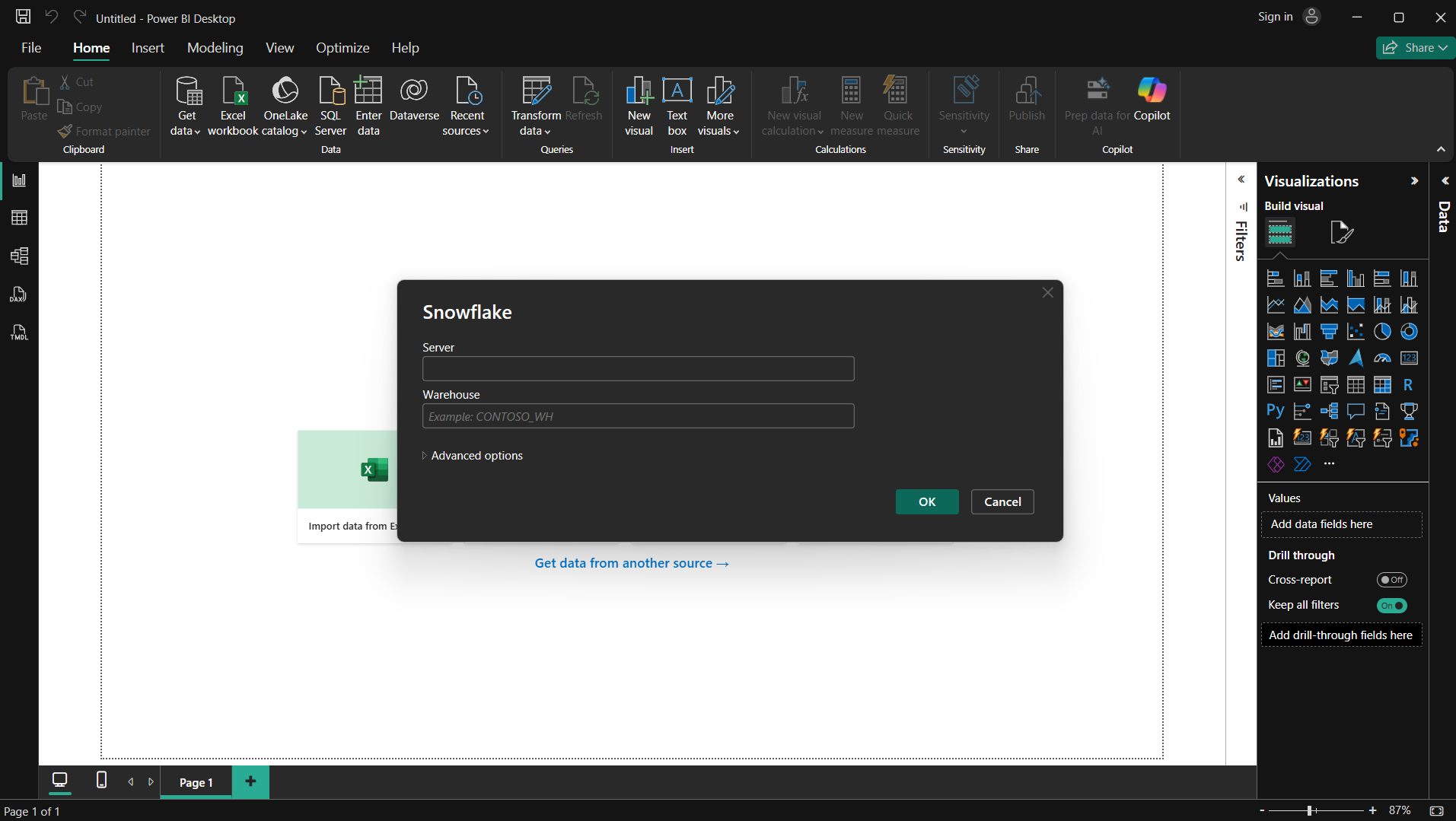
**4. Build a Quick Power BI Report**

The final step connects **Power BI** directly to Snowflake to visualize key sales metrics.  
This enables business users to explore data interactively without depending on IT.

**Loading the Table from Snowflake Schema**

1. Open **Power BI Desktop → Get Data → Snowflake**.
2. Enter connection details:
   * Server: XAPNZGA-KTA45095.snowflakecomputing.com
   * Warehouse: COMPUTE\_WH

Select the **RETAIL\_SALES** table and load it into Power BI.



The Power BI dashboard provides an interactive view of sales, profit, and discount trends across categories, cities, and countries.  
It enables quick insights