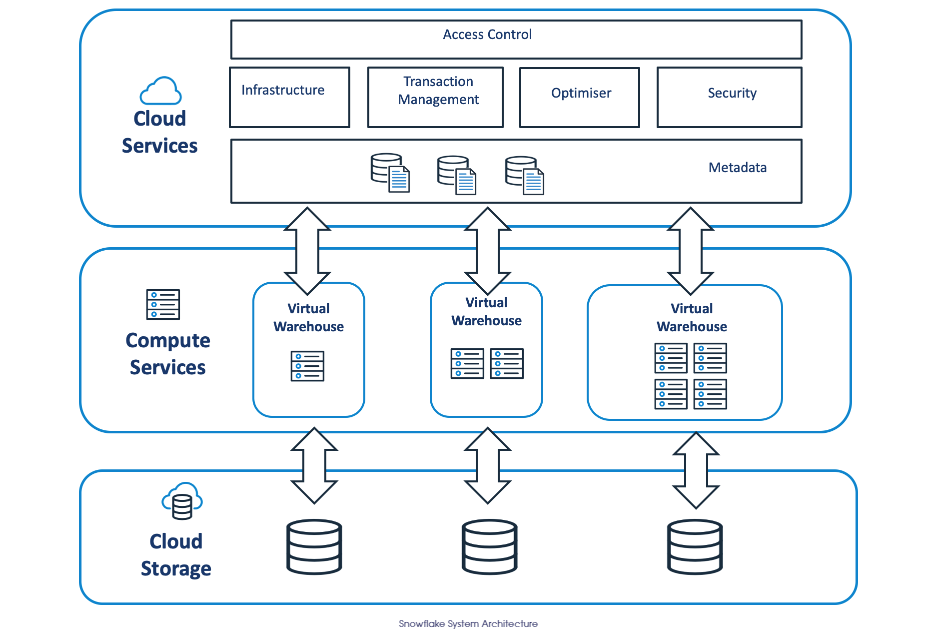
**Snowflake**

* Snowflake is a data warehouse and cloud based.
* Data model of snowflake is snowflake schema.
* All RDBMS strictly follows star schema

**Snowflake schema Vs Star schema**

|  |  |
| --- | --- |
| **Snowflake schema** | **Star schema** |
| Highly normalized | Partially normalized |
| Requires less storage – uses columnar storage with inbuilt compression | Requires more storage |
| Highly Scalable | Limited Scalability |
| Always good for complex queries | Always good for simple queries |
| Used for large volume of data | Used for limited volume of data |
|  |  |

**Architecture:**



**Layer 1 - Cloud Storage:**

* Gives the options (AWS-s3, Azure-data Lake, Google-GCS) to choose for storage.
* Acts as a default storage for virtual warehouse.

**Layer 2 - Compute Services:**

* Gives the flexibility to create virtual warehouses.
* One warehouse = one cluster.
* Queries are processed in disk.
* Snowflake connector library with spark can be used to analyze a query in memory.
* Administrative manages the user access to these virtual warehouses.

**Layer 3(Access control layer) - Cloud Services:**

**Virtual Warehouses:**

* Size of vw is denoted as XS,S,L,XL………..
* TB size starts from XL
* Default vw in snowflake compute\_WH
* Limit for creating databases in vw is dependent on vw size.
* History of all the queries executed in warehouse is stored in “result cache”.

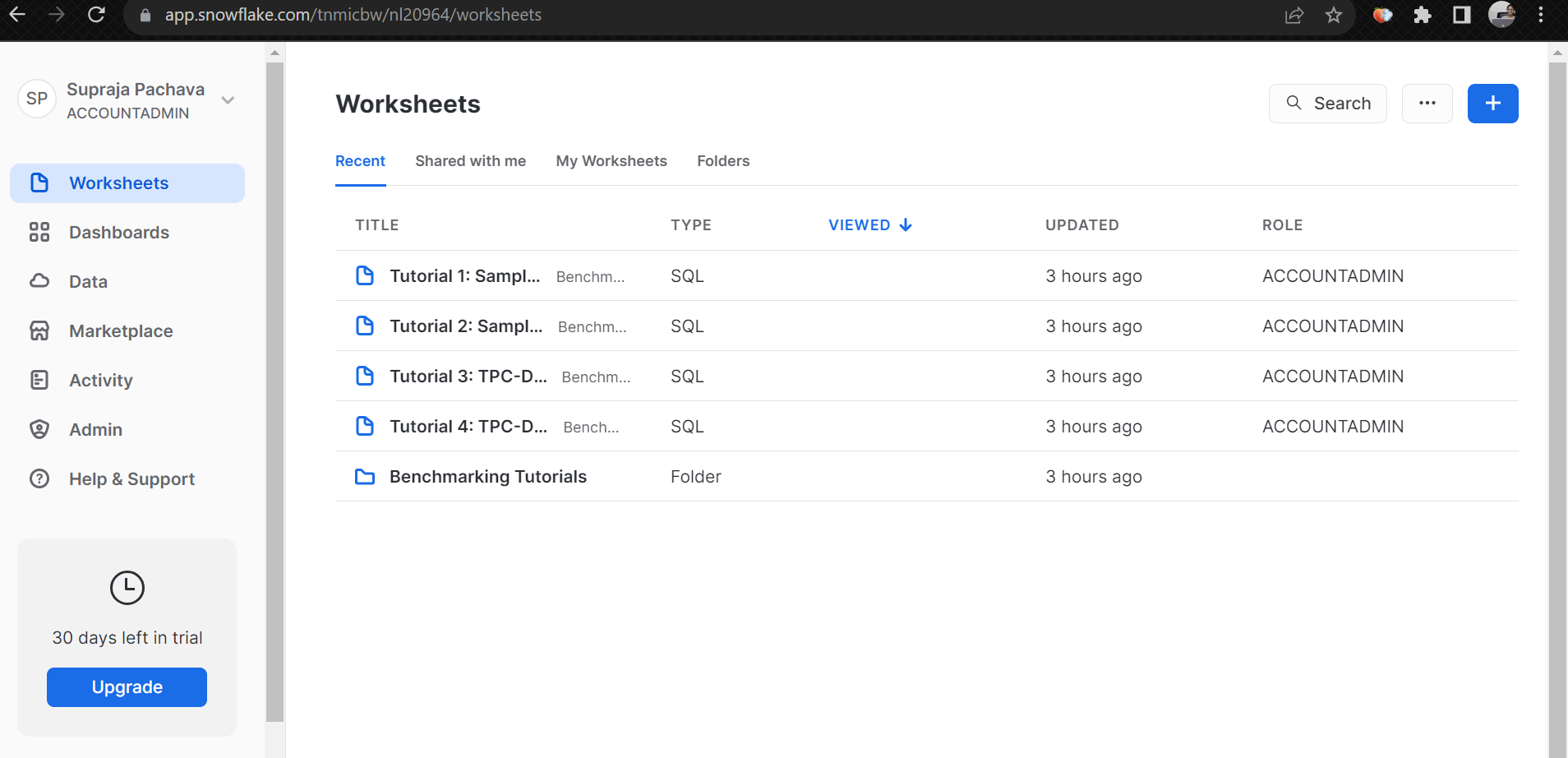
**Micro Partitions:**

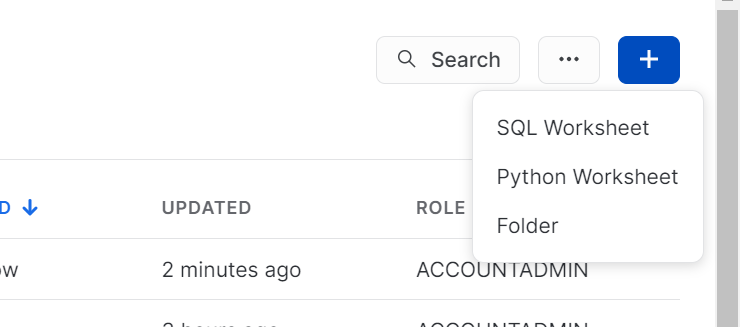
For ex, we have 500 mb of raw data when we move this to snowflake it is stored as 50mb

This 50mb data is divided into micro partitions as blocks (as column-wise) in physical structure snowflake datawarehouse.

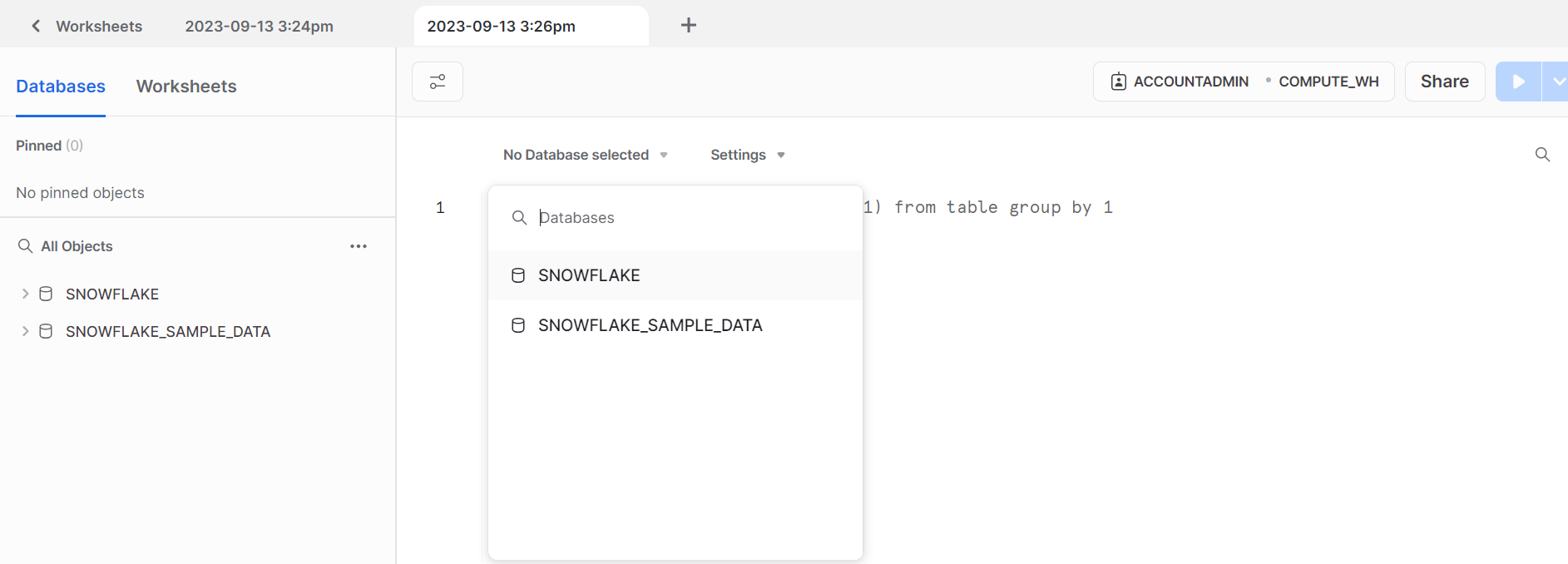
Hands-On:

Snowflake home page:

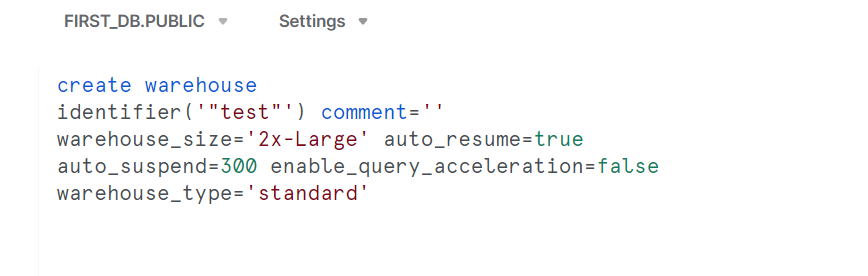




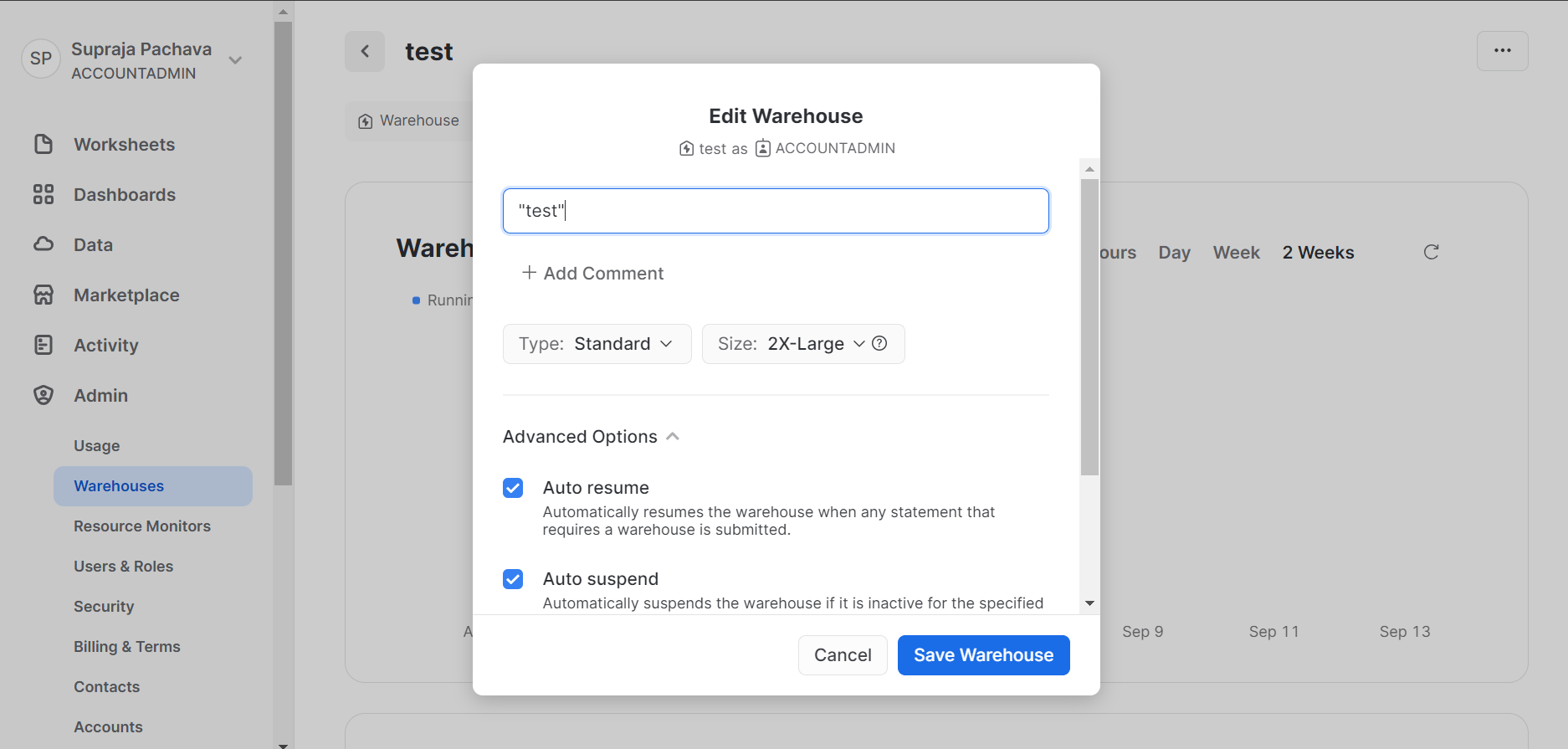
SQL worksheet:



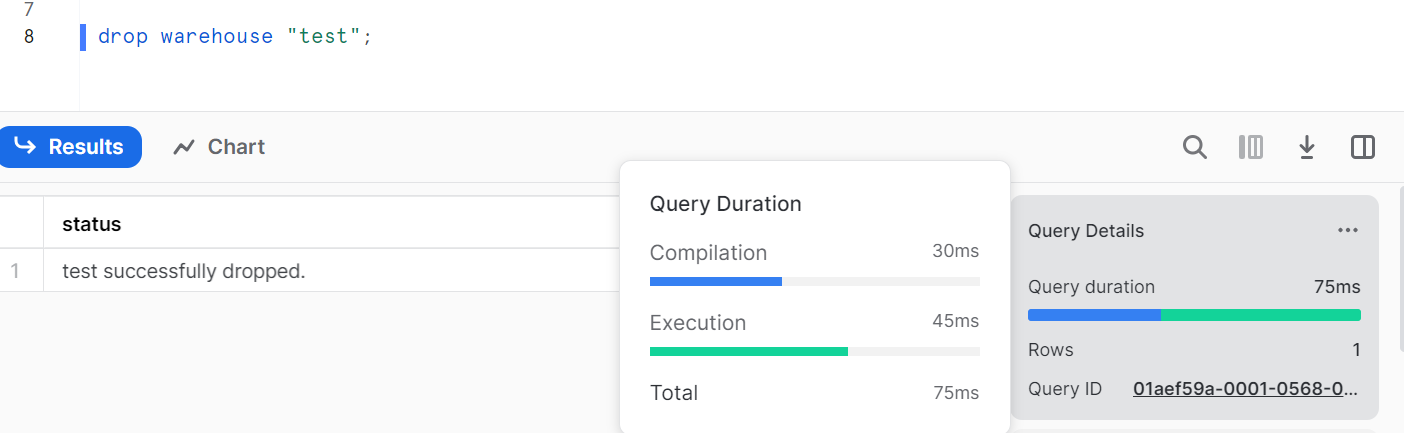
Warehouse Creation:



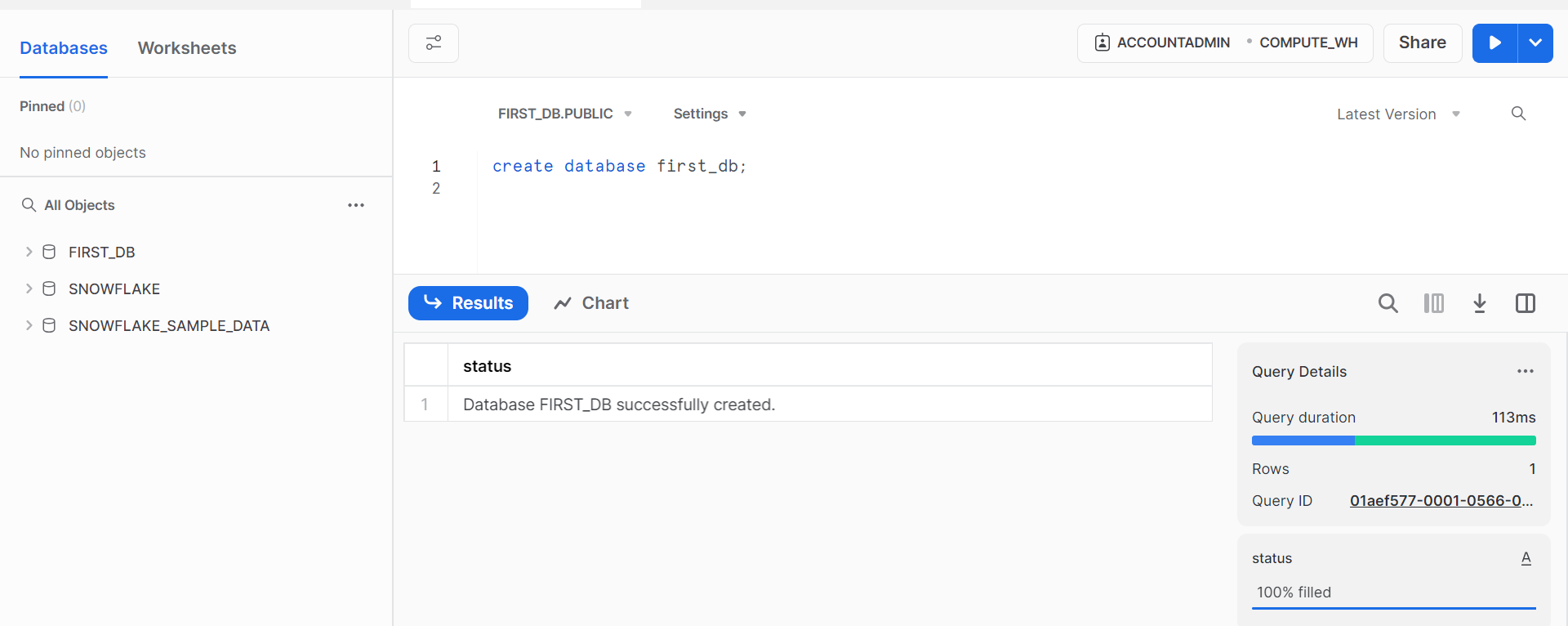
Edit warehouse after creation:



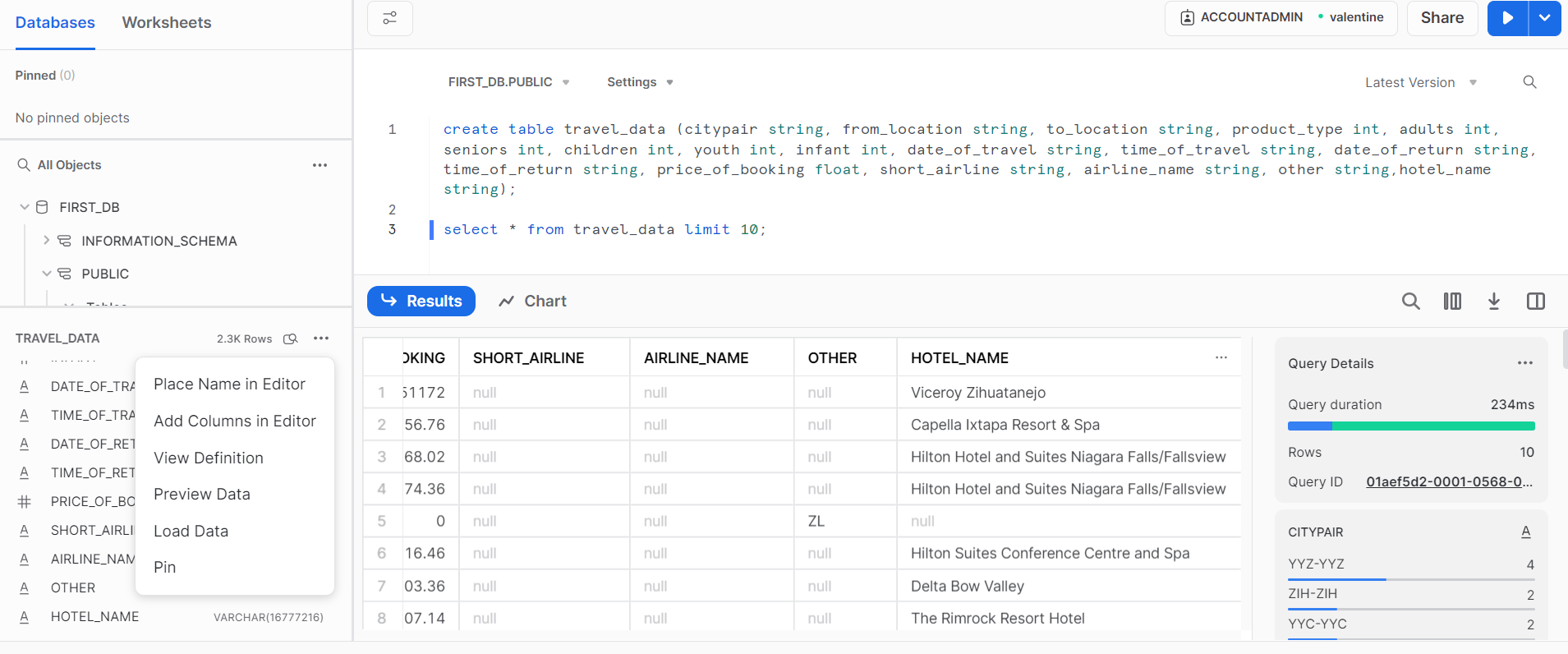
Drop warehouse:



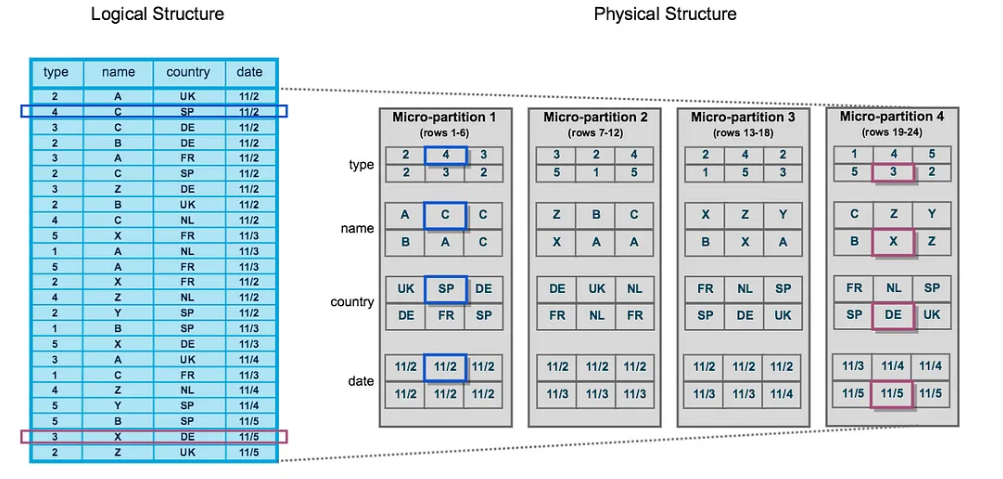
Db creation:



Load local file data into snowflake:



14/09/2023



**Stages in Snowflake:**

Snowflake Stages are locations where data files are stored (staged) for loading and unloading data. They are used to move data from one place to another, and the locations for the stages could be internal or external to the Snowflake environment.

The Snowflake stage is used to manage the data during the loading process.

For example, businesses can employ Snowflake stages to create temporary tables to store data before it is loaded into the main table. This allows businesses to ensure their data is getting appropriately loaded into Snowflake without having to bother about any potential issues that may emerge during the loading process.

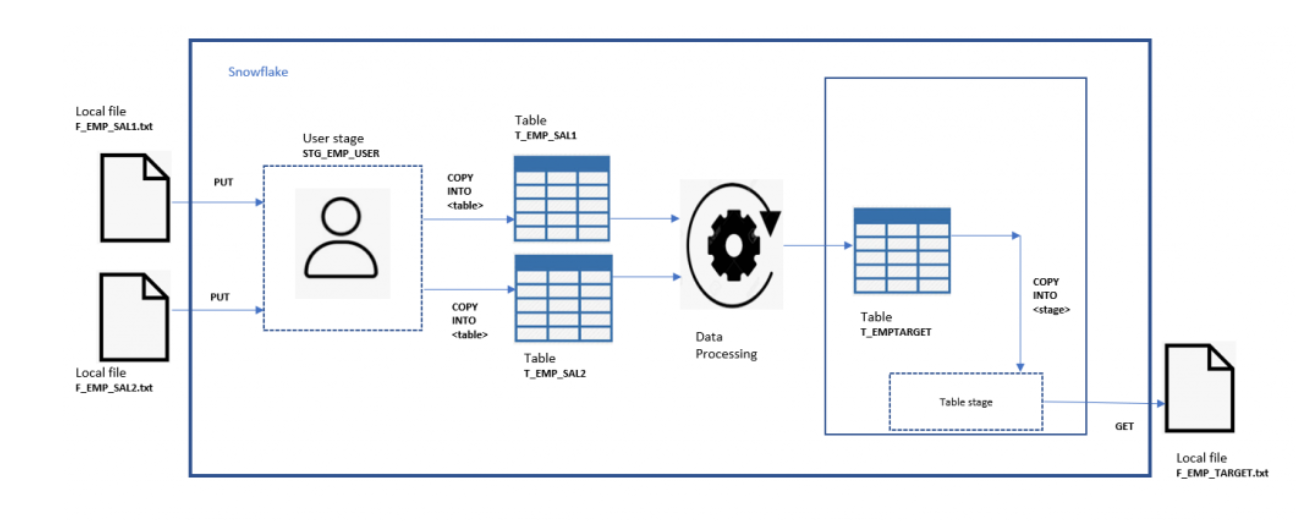
Additionally, Snowflake stages provide businesses with a secure environment to house their data. With encryption features readily available, organizations can enhance their data security and guard against unauthorized access effectively.

There are two types of stages for data transfer in snowflake. As per ETL stages acts like extraction.

1.Internal stage

Loading the local files into snowflake using put command. Data is brought to staging area

Local file-------------------------🡪Staging Area----------------------🡪Snowflake DW



2.External storage – used to bring data from external data sources like cloud

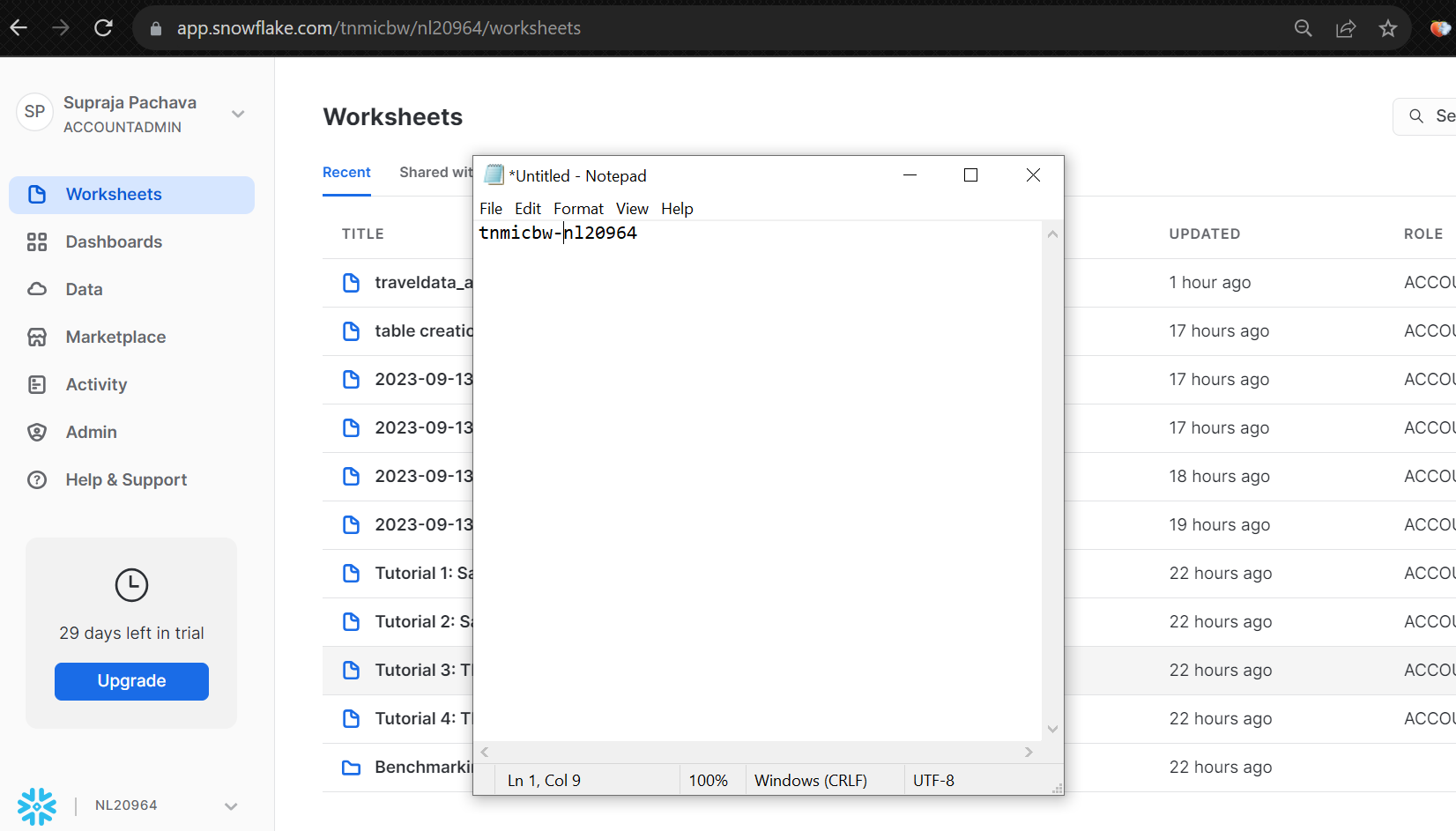
External data sources-

AWS –(storage) S3 (object storage) ---🡪buckets

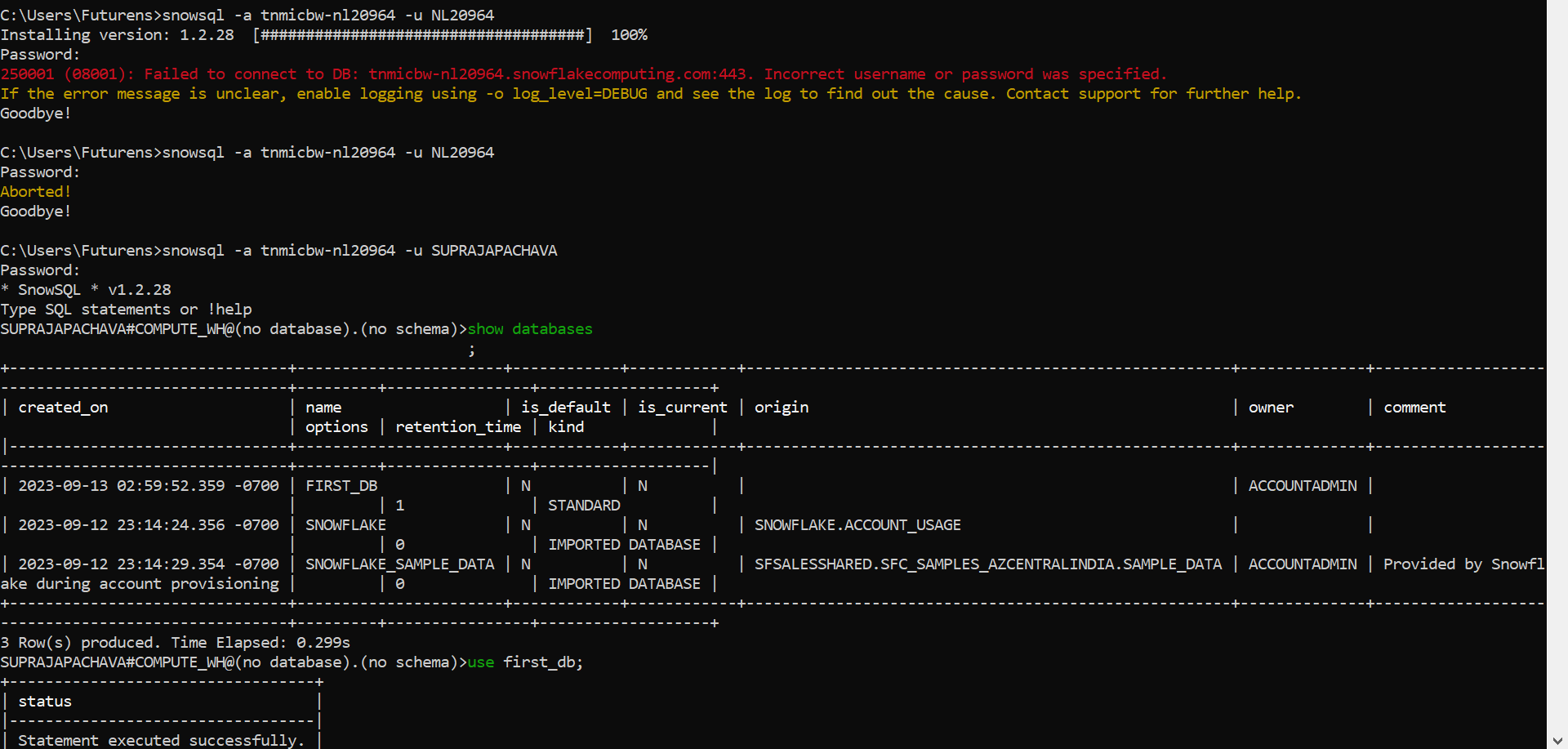
Azure – (storage) Blob (object storage)

**Snow SQL:**

CLI for snowflake to run the commands



Connecting snow SQL to snowflake:

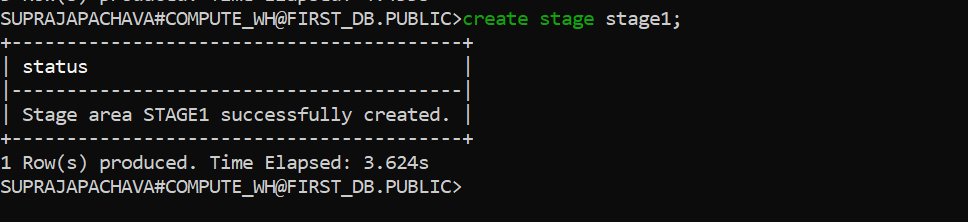


**Internal stage pipeline:**

**Steps to create stage and load data into internal stage:**

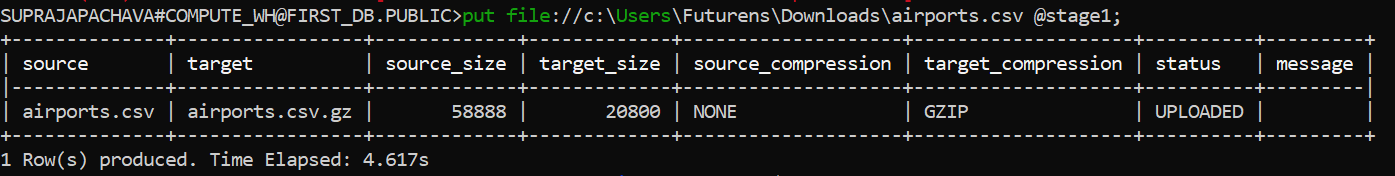
1.Creating Stage

Syntax: create stage <stage\_name>

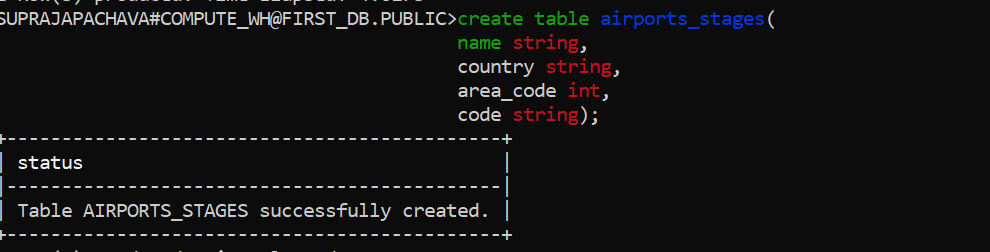


2.Put file in stage

Syntax: put file://<file-path> @<stage\_name>;



3.Create table in database



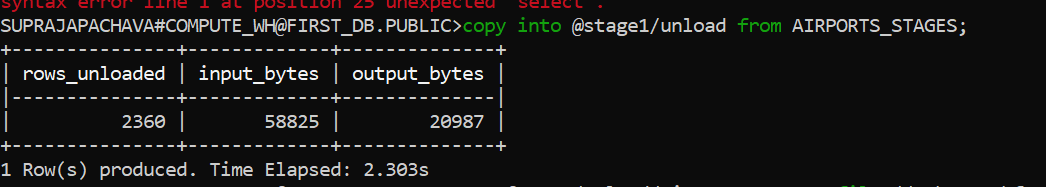
4.Copy data into table

Copy into <table\_name> from @<stage\_name>/<file\_path> file\_format=(type=’csv’,skip\_header=1);

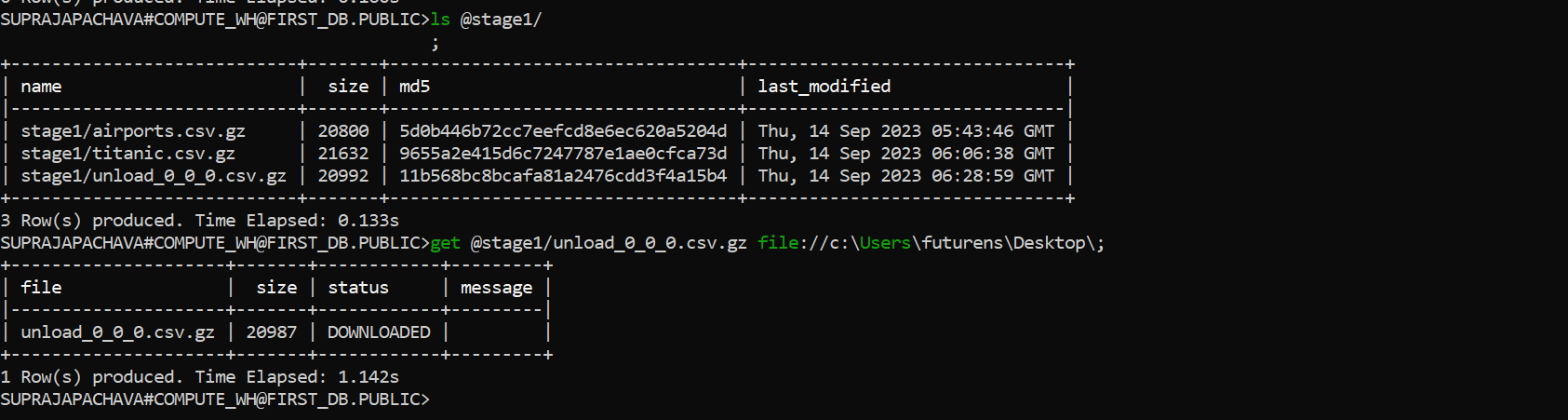


5.Processing data and storing output in a table.

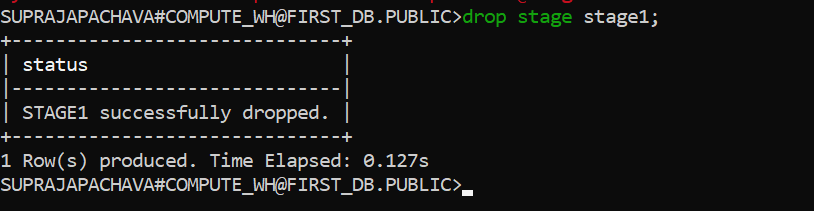
6.Copying output table into staging area



7.Unloading data from staging area to local



Drop stage:



External Stage pipeline:

1.create a bucket in s3 and upload a file in s3.

2.Create a usergroup and then add a user to the group.

3.create a role for the user.

In snowsql cli

4.create a file format

5.create an integration between snowflake and aws.

6.create a stage to load the data

7.Load the data with copy into.

15/09/2023

Copy into vs Snowpipe:

Snowpipe supports automatic data transfer and continuously ingests data from stage.

Copy into is like batch processing where only the data from staging area is transferred.

1.External stage(Azure blob)

Azure storage--🡪csv file-🡪internal stage -----🡪copy into -----🡪snowflake---🡪(MYSQL,NoSQL,Cloud db) -------🡪BI---🡪reports

2.Snow pipe

Files--🡪Amazon S3-----🡪amazon SQS sends notification--------🡪snowpipe server-less loader--------🡪snowflake db

3.Unloading

4.data sharing

5.timetravel

Used to analyze the data i.e; backup of data from key points in the past.

Using time travel you can query the data in past (updated data or deleted data)

There are 2 functions used for time travel data analysis

1.at

2.before

Ex: time travel query – used to select historical data from the tables as of the date and time represented by timestamp.

Using time travel you can retrieve the historical data from the table as 30 minutes ago

**Snowflake with Power Bi**

Snowflake ----------------🡪Power Bi --------------=reports

|

|

|-----------------------------------🡪MySQL

Steps:

1.Open powerbi

2.click on get data and search for snowflake

3.provide the details of snowflake server and warehouse name

4.give username and password

5.