PUT command in Snowflake

How to load the data into the snowflake named internal stage?

We are going to learn the uploading process of the CSV file into a Snowflake internal named stage using the SnowSQL client. Once the file is uploaded into the internal named Stage, then we can perform a bulk copy operation to load the data from the file to a Snowflake table.

Before working on the problem statement, we should have knowledge of SnowSQL and Snowflake Stage.

SnowSQL is just a connector whereas a Snowflake stage is a location where we are loading our files. We can create an Internal and external stage in Snowflake.

External Stage:

Data files are stored in a location outside of Snowflake. Currently, the following cloud storage services are supported:

- Amazon S3 buckets
- Google Cloud Storage buckets
- Microsoft Azure containers

The storage location can be either private/protected or public.

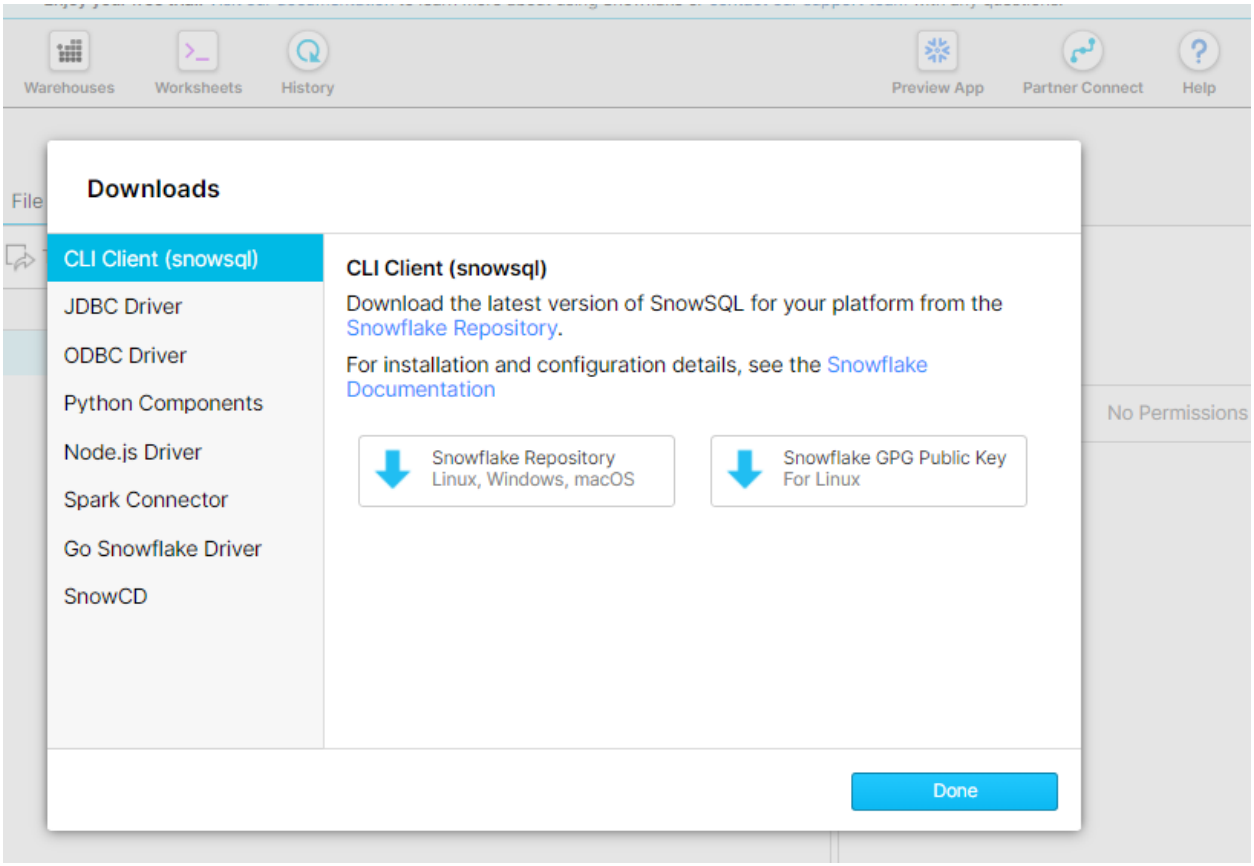
Internal Stage:

Stores data files internally within Snowflake. Internal stages can be either permanent or temporary.

- User — User stages are referenced using @~
- Table — Table stages are reference using @%
- Named (also called “Internal Named”)

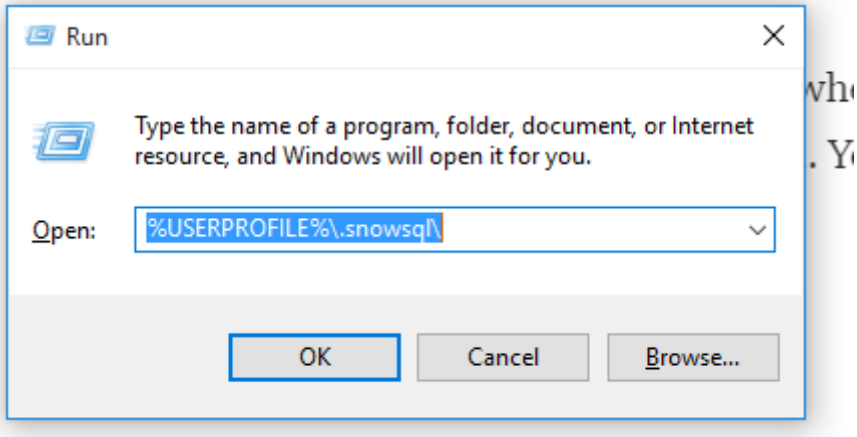
SnowSQL

It is a command-line client that is used to connect Snowflake. Instead of Snowflake Web Interface, we can use the SnowSQL tool to perform/execute SQL query, DDL, DML commands, including loading and unloading data. To install SnowSQL, please download the executable from the snowflake Help menu. PFB the screenshot,



Download the executable from the Snowflake Help menu and double-click once it is downloaded. Select respective options in the dialog box and click the finish button.

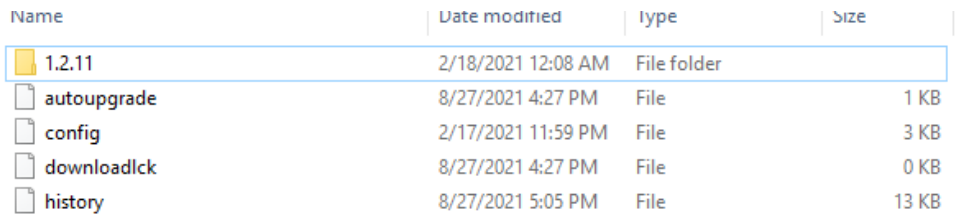
To check whether SnowSQL is installed or not press Window key + R or Run command. You will get the below screen:



Type %USERPROFILE%\.snowsql\ and then click on OK

Output:

You can see the below screen. Now open the config file in notepad or notepad++



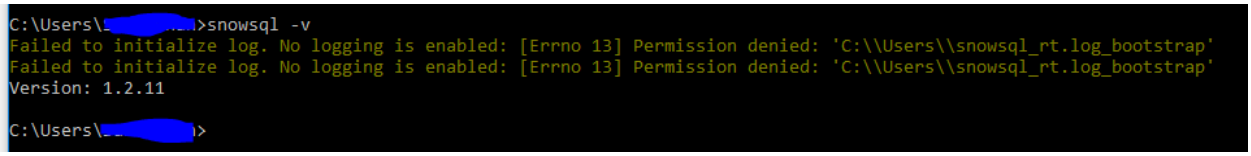
Add the Snowflake account information based on the below screen.

You need to provide an accountname, region, username, password, dbname, schemaname, warehousename, and rolename. This information you can get in your Snowflake web Interface.

```
#If a connection doesn't specify a value, it will default to these
#
#accountname = defaultaccount
#region = defaultregion
#username = defaultuser
#password = defaultpassword
#dbname = defaultdbname
#schemaname = defaultschema
#warehousename = defaultwarehouse
#rolename = defaultrolename
#proxy_host = defaultproxyhost
#proxy_port = defaultproxyport

[connections.example]
#Can be used in SnowSql as #connect example

accountname = accountname
username = username
password = password1234

[variables]
#Loads these variables on startup
#Can be used in SnowSql as select $example_variable

example variable=27
```

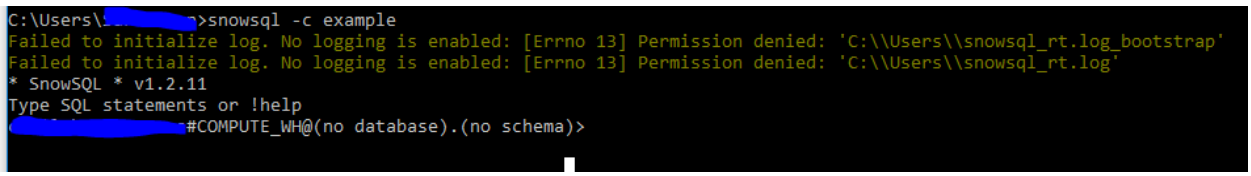
You can see [connections.example] in the above screen. Here, *example* we use to connect Snowflake in command prompt. Once you configure the setting open the command prompt,

Type ‘snowsql -v’, like shown below.



Type snowsql -c example and press the enter key

This command connects the snowsql with Snowflake as per the given configuration. In this case, it will come as ,XXXXXX*#COMPUTE_WH@(no database).(no schema)* refer below screen. Now you perform any SQL query, DML or DDL command.



Now SnowSQL is installed and configured in your system properly. Let us discuss more about loading data in to snowflake internal stage from local system by using PUT command.

To achieve the solution for a given problem, we need to create the Internal named stage, so we can upload the files into that. Apart from creating Stage in Snowflake, we can also create a stage for AWS, Azure, and GCP.

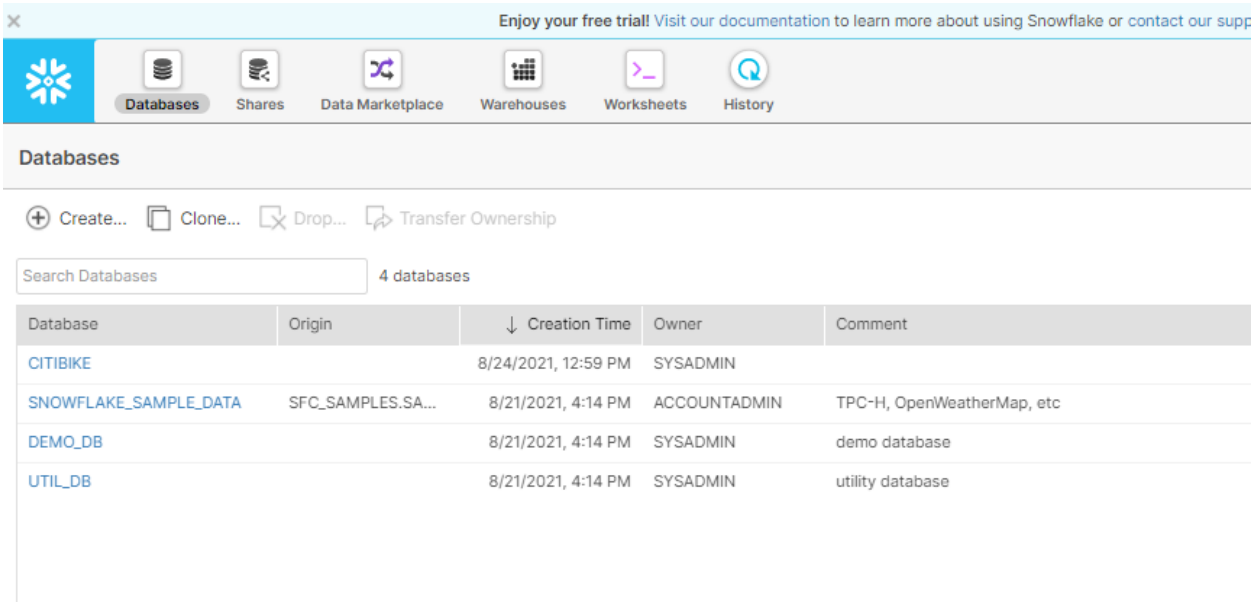
Perform the following steps to create Stage:

Step 1

Login into the Snowflake account. You will be landing on the worksheet.

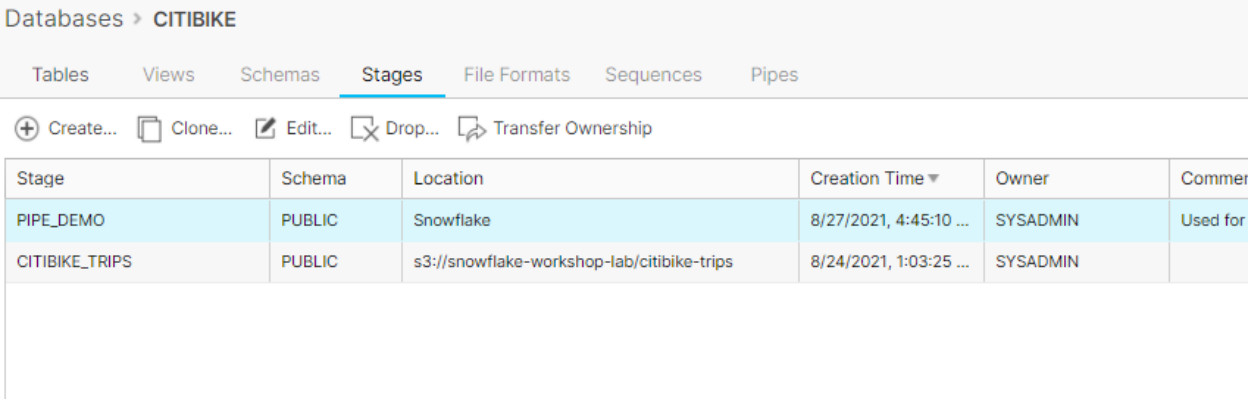
Step 2

Click on the Database from the Header (besides the Share Icon). Refer to the below screen.



Step 3

Click on the database where you want to create the stage. In this case. I have selected CITIBIKE. Once you click it, you will be getting the below screen.

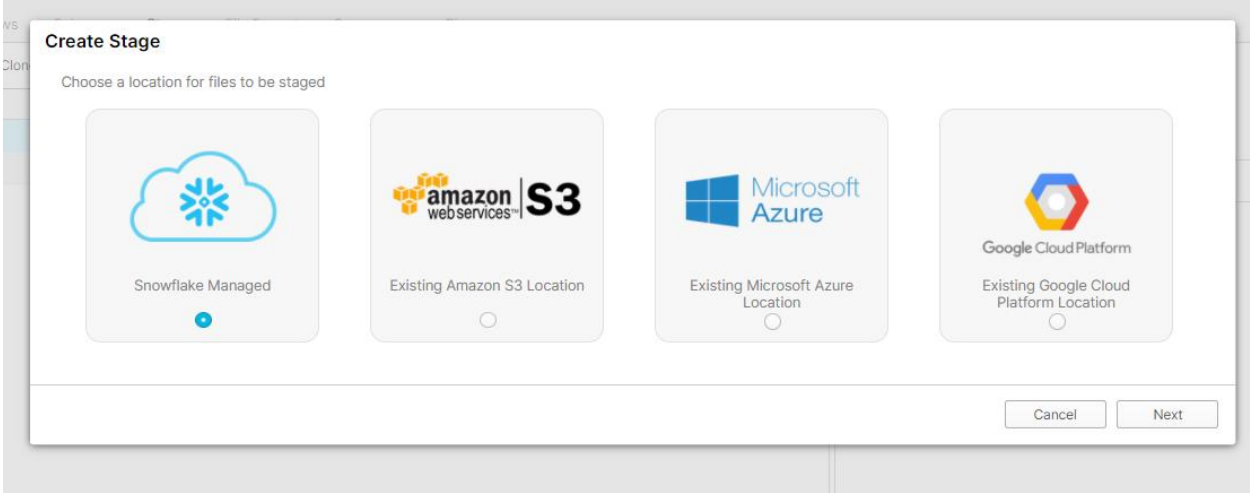


Step 4

Click on the Stages

Step 5

Click on Create. You will be getting the below screen. Now it depends on you whether you want to create stage for Snowflake or Amazon S3 or Azure or GCP.

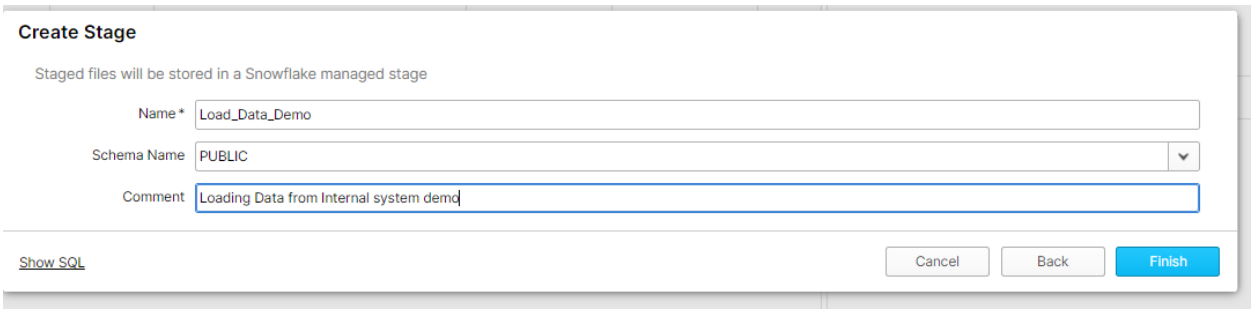


Step 6

Select Snowflake Managed in the above screen and click on Next

Step 7

Enter the stage name and select the schema. Refer to the below screen.

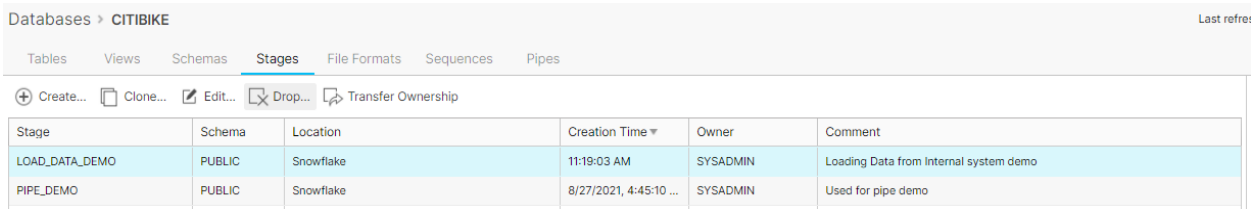


Step 8

Click on Finish.

Step 9

In the below screen, you can see the stage which we have created under MANAGEMENT schema.



Now we have configured the SnowSQL and Created the STAGE as well. Let me give you brief information on File Format as it is required while loading data from stage to table.

File Format

It is a pattern of format which we create for the type of File. It can be CSV, Json, XML, Avro, etc. It will help us while loading the different format data into the Snowflake table.

Here, I am creating the File Format for CSV and Json file and we will use these formats while loading data from stage to snowflake table.

Creating File Format inside the Database

Like a stage, we can create File Format inside the database. Perform the following steps:

Step 1

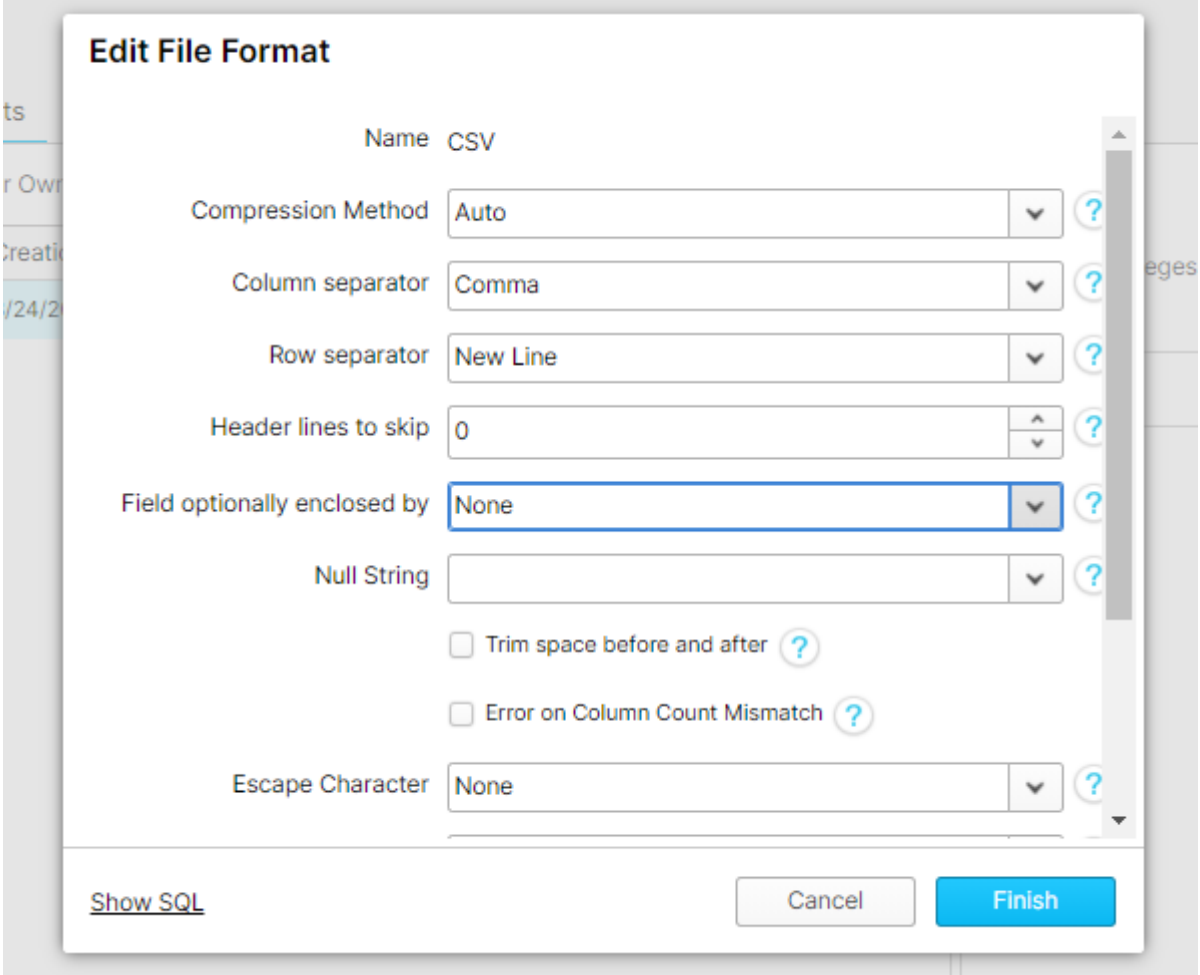
Click on the File Format from the menu

Step 2

Click on Create

Step 3

Specify the format name and schema name with the required settings. Refer to the below screen.



Step 4

Click on the Finish button.

Now we have everything ready to work on our problem statement.

Problem Statement

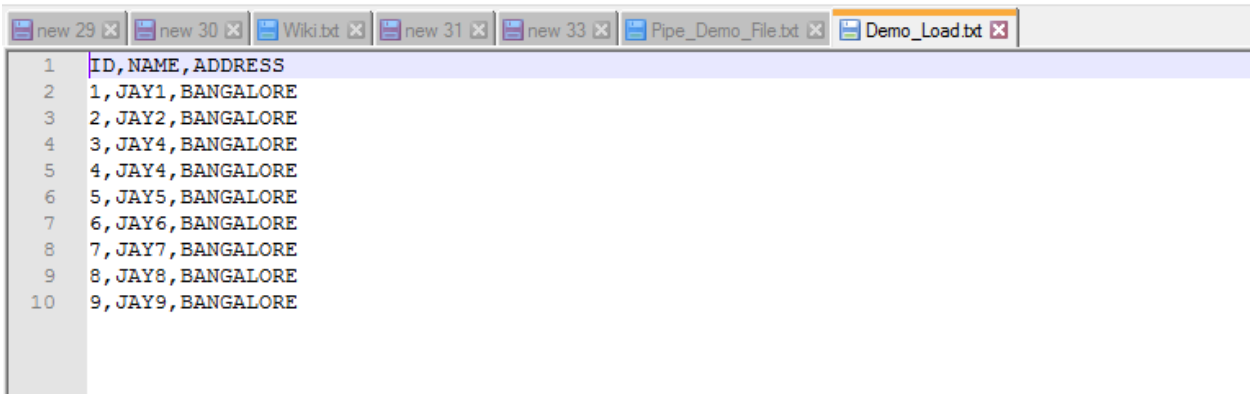
- Uploading CSV file from local system to Snowflake Stage

- Load data from stage to Snowflake table

Solution

As we have already set up and configured the SnowSQL and Snowflake Stage, now it will be very easy for us to work on this solution part. I have created a file named as employeedata.csv. These files are available in C drive inside the testdata folder.

Please have a look at the screen below CSV file data.



Now, we must perform the following steps to achieve the solution of loading file into stage and stage to Snowflake table.

Step 1

Open the command prompt.

Step 2

Type “snowsql -v “ and press enter key

It will provide you the snowsql version installed in your system.

Step 3

Type snowsql -c example and press enter key

```
C:\Users\[redacted]>snowsql -v
Failed to initialize log. No logging is enabled: [Errno 13] Permission denied: 'C:\\Users\\snowsql_rt.log_bootstrap'
Failed to initialize log. No logging is enabled: [Errno 13] Permission denied: 'C:\\Users\\snowsql_rt.log_bootstrap'
Version: 1.2.11

C:\Users\[redacted]>snowsql -c example
Failed to initialize log. No logging is enabled: [Errno 13] Permission denied: 'C:\\Users\\snowsql_rt.log_bootstrap'
Failed to initialize log. No logging is enabled: [Errno 13] Permission denied: 'C:\\Users\\snowsql_rt.log'
* SnowSQL * v1.2.11
Type SQL statements or !help
[redacted]#COMPUTE_WH@(no database).(no schema)>
```

Step 4

By using USE command check in to your database.

```
[redacted]#COMPUTE_WH@(no database).(no schema)>use database CITIBIKE;
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.735s
[redacted]#COMPUTE_WH@CITIBIKE.PUBLIC>
```

Type PUT file://c:\testdata\Demo_Load.txt @LOAD_DATA_DEMO; and press the enter key. Refer to the below screen.

```
[redacted]#COMPUTE_WH@CITIBIKE.PUBLIC>PUT file://c:\testdata\Demo_Load.txt @LOAD_DATA_DEMO;
```

Step 5

Once it is successfully loaded into Stage, you will see the below screen. The status will be uploaded, and it will be created Demo_Load.txt.gz compress file on the Snowflake stage

```
[redacted]#COMPUTE_WH@CITIBIKE.PUBLIC>PUT file://c:\testdata\Demo_Load.txt @LOAD_DATA_DEMO;
Demo_Load.txt_c.gz(0.00MB): [##########] 100.00% Done (0.236s, 0.00MB/s).
+--------------+-----------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source       | target          | source_size | target_size | source_compression | target_compression | status   | message |
|--------------+-----------------+-------------+-------------+--------------------+--------------------+----------+---------|
| Demo_Load.txt | Demo_Load.txt.gz |         177 |         113 | NONE               | GZIP               | UPLOADED |         |
+--------------+-----------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 4.520s
[redacted]#COMPUTE_WH@CITIBIKE.PUBLIC>
```

Step 6

Type *List @MYDATA to view all the files present in the stage*

It will show the list of the files available in the stage.

```
#COMPUTE_WH@CITIBIKE.PUBLIC>LIST @LOAD_DATA_DEMO;
+---------------------------------+------+----------------------------------+-------------------------------+
| name                            | size | md5                              | last_modified                 |
|---------------------------------+------+----------------------------------+-------------------------------|
| load_data_demo/Demo_Load.txt.gz |  128 | 2cba8eccfe922ceae725712e2117f493 | Sat, 28 Aug 2021 06:04:35 GMT |
+---------------------------------+------+----------------------------------+-------------------------------+
1 Row(s) produced. Time Elapsed: 0.843s
#COMPUTE_WH@CITIBIKE.PUBLIC>
```

Step 7

Perform the bulk insert operation. We have the EMPLOYEE_LOAD_DEMO table in CITIBIKE database under the PUBLIC schema.

```
COMPUTE_WH@CITIBIKE.PUBLIC>SELECT * FROM EMPLOYEE_LOAD_DEMO;
+----+------+---------+
| ID | NAME | ADDRESS |
|----+------+---------|
+----+------+---------+
0 Row(s) produced. Time Elapsed: 0.882s
#COMPUTE_WH@CITIBIKE.PUBLIC>
```

copy into EMPLOYEE_LOAD_DEMO(ID,NAME,ADDRESS)

from (select $1,$2,$3 from @load_data_demo/Demo_Load.txt.gz*)*

FILE_FORMAT = 'CSV';

Refer to the below screen.

```
#COMPUTE_WH@CITIBIKE.PUBLIC>copy into EMPLOYEE_LOAD_DEMO(ID,NAME,ADDRESS)
                            from (select $1,$2,$3 from @load_data_demo/Demo_Load.txt.gz)
                            FILE_FORMAT = 'CSV';
+---------------------------------+--------+-------------+-------------+-------------+-------------+-------------+------------------+----------------------+-------------+
| file                            | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_character | first_erro |
r_column_name |
|---------------------------------+--------+-------------+-------------+-------------+-------------+-------------+------------------+----------------------+-------------|
| load_data_demo/Demo_Load.txt.gz | LOADED |           9 |           9 |           1 |           0 | NULL        |             NULL |                 NULL | NULL
          |
+---------------------------------+--------+-------------+-------------+-------------+-------------+-------------+------------------+----------------------+-------------+
1 Row(s) produced. Time Elapsed: 3.084s
#COMPUTE_WH@CITIBIKE.PUBLIC>
```

Step 8

The data has been loaded to the EMPLOYEE_LOAD_DEMO table. To see the loaded data, run the below query

*select * from EMPLOYEE_LOAD_DEMO;*

Refer to the below screen:

```
[redacted]#COMPUTE_WH@CITIBIKE.PUBLIC>select * from EMPLOYEE_LOAD_DEMO;
+----+------+-----------+
| ID | NAME | ADDRESS   |
|----+------+-----------|
| 1  | JAY1 | BANGALORE |
| 2  | JAY2 | BANGALORE |
| 3  | JAY4 | BANGALORE |
| 4  | JAY4 | BANGALORE |
| 5  | JAY5 | BANGALORE |
| 6  | JAY6 | BANGALORE |
| 7  | JAY7 | BANGALORE |
| 8  | JAY8 | BANGALORE |
| 9  | JAY9 | BANGALORE |
+----+------+-----------+
9 Row(s) produced. Time Elapsed: 1.909s
[redacted]#COMPUTE_WH@CITIBIKE.PUBLIC>
```

Finally we have loaded the data in to CITIBIKE->EMPLOYEE_LOAD_DEMO table.