Date

Data loading basics

# Goals

## Launch docker container.

## Basics of loading data to snowflake.

## Checking rejected records.

## Using file format and its importance.

## Clean up staging area.

# **Preparation**

Download data from below link,

<https://drive.google.com/file/d/1g5G-d8AnaNMgSEDb6pO_QXv5N2rzW3NL/view?usp=sharing>

USE ROLE ACCOUNTADMIN;

USE DATABASE DEMO\_DB;

**Create table,**

*CREATE OR REPLACE TRANSIENT TABLE TAXI\_DRIVE*

*(*

*TRIP\_ID NUMBER,*

*CALL\_TYPE VARCHAR(2),*

*ORIGIN\_CALL NUMBER,*

*ORIGIN\_STAND NUMBER,*

*TAXI\_ID NUMBER,*

*TIMESTAMP NUMBER,*

*DAY\_TYPE VARCHAR(1),*

*MISSING\_DATA BOOLEAN,*

*POLYLINE ARRAY*

*);*

Create above table in database, DEMO\_DB in schema , PUBLIC.Graphical user interface, text, application

Description automatically generated

Create file format,

create or replace file format taxi\_csv\_format

type = csv field\_delimiter = ',' skip\_header = 1 null\_if = ('NULL', 'null') empty\_field\_as\_null = true compression = gzip;

Table

Description automatically generated

# **SNOWSQL**

Text

Description automatically generated**Login to Snowsql,**

**Upload whole file to table stage using PUT,**

Change role to ACCOUNTADMIN

Text

Description automatically generatedput file:///data-vol/train.csv @DEMO\_DB.PUBLIC.%TAXI\_DRIVE;

**Split file to smaller chunks and upload to table stage using PUT,**

If you run put on large file it will take lot of time to upload.

Split the file to smaller chunks and then upload.

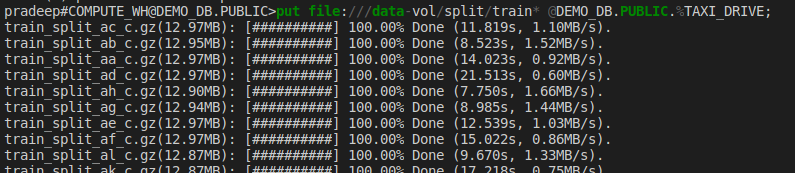
You can download already split files from below location,

<https://drive.google.com/drive/folders/1uIHYCDpyIOlaR_ABIffiSGFWJPql-BaS?usp=sharing>

Text

Description automatically generatedsplit -b 50000000 train.csv split/train\_split\_

put [file:///data-vol/split/train](../../../../data-vol/split/train)\* @DEMO\_DB.PUBLIC.%TAXI\_DRIVE;



# **COPY DATA TO SNOWFLAKE TABLE.**

Run copy command to copy data to snowflake.

copy into TAXI\_DRIVE from

'@DEMO\_DB.PUBLIC.%TAXI\_DRIVE'

file\_format = (FORMAT\_NAME='taxi\_csv\_format' field\_optionally\_enclosed\_by='"')

ON\_ERROR='CONTINUE'

If you see rejects. Check for reason for rejection using below command,

select \* from table(validate(TAXI\_DRIVE, job\_id=>'<query\_id>'));

select distinct error from table(validate(TAXI\_DRIVE, job\_id=>'01975700-04fa-fc8d-0027-a5030002804e'));

If you want to check for bad records before copying data from stage to snowflake table, you need to use below command,

copy into TAXI\_DRIVE from

'@DEMO\_DB.PUBLIC.%TAXI\_DRIVE'

file\_format = (FORMAT\_NAME='taxi\_csv\_format' field\_optionally\_enclosed\_by='"')

VALIDATION\_MODE=’RETURN\_ERRORS’

Text

Description automatically generated Understand the reason for rejection and transform the column value while executing copy command.

copy into TAXI\_DRIVE from

(select t.$1 , t.$2 , iff(t.$3='',null,t.$3) , iff(t.$4='',null,t.$4) , t.$5 , t.$6 , t.$7, t.$8, t.$9 from '@DEMO\_DB.PUBLIC.%TAXI\_DRIVE' t)

file\_format = (FORMAT\_NAME='taxi\_csv\_format' field\_optionally\_enclosed\_by='"')

ON\_ERROR='CONTINUE'

# **Importance of file format.**

Even though it’s possible to transform data while copying data using copy command.

It’s not a good practice. Why because, while doing batch processing you may need to use such copy commands across the jobs.

In future if you want to change any conditions, you may end up updating all your code.

It’s always best practice to make use of copy command file format options.

Because if you want to change something you can just do it at one place.

**Alter the file format to handle the data issue you fixed using copy command.**

alter file format taxi\_csv\_format

set null\_if = ('NULL', 'null','')

Re write copy command and execute it again.

**Note :** truncate TAXI\_DRIVE table before executing copy command.

copy into TAXI\_DRIVE from

'@DEMO\_DB.PUBLIC.%TAXI\_DRIVE'

file\_format = (FORMAT\_NAME='taxi\_csv\_format' field\_optionally\_enclosed\_by='"')

ON\_ERROR='CONTINUE'

Observe, in the above copy command you are not applying transformation using select clause.

But you have leveraged file format.

Query data in snowflake after executing copy command.

Graphical user interface, table

Description automatically generated

# **CLEAN UP**

You have uploaded data to table staging area. If you leave it there, then the storage cost will be added to snowflake bill.

Once you are done with your process, please remove these file from staging area.

list all the files in table staging area.

<https://docs.snowflake.com/en/sql-reference/sql/list.html>

A picture containing text

Description automatically generatedList @%TAXI\_DRIVE;

**Remove staged file,**

[**https://docs.snowflake.com/en/sql-reference/sql/remove.html**](https://docs.snowflake.com/en/sql-reference/sql/remove.html)

rm @%TAXI\_DRIVE;

Text

Description automatically generated

# **COMPARE PERFORMANCE**

**Q1. We used put command on two kinds of file.**

1. Directly on 1.8 GB file which we downloaded.

2. We split 1.8 GB file into smaller chunks of file and used put command.

Which option do you think is good ? and why ?

**Q2. Execute copy command on single file uploaded and Execute copy command on split file you uploaded.**

Which copy command you think will come back soon ? and why ?

Q

Q

Q3

Q3

Q3 . If you run same copy command again after doing copy to table; what will happen ?

Will data again get copied to table ? What do you think ?

Q

Q3

Q3