Date

Unload data from snowflake

# goals

In this assignment we will try to cover below scenarios,

## Creating named stage and unload data.

## Unload selected fields

## Best practice after doing unload.

## Download data to local system.

## Calculate overall cost for the activity we performed.

# Creating named stage.

Assuming we have TAXI\_DRIVE table already loaded from previous assignment, we will try to unload data to named stage.

Create named stage executing below command,

create stage data\_unload;

Copy table data to stage,

copy into @data\_unload/taxi\_unload

from

taxi\_drive

Question: in the above steps, we have not mentioned or created **file format object**. Even without file format object copy command seems to work. How it’s possible ?

Unload as parquet file,

copy into @data\_unload/taxi\_unload/PARQUET\_

from

taxi\_drive

file\_format=(type=parquet)

Unload as XML file,

copy into @data\_unload/taxi\_unload/XML\_

from

taxi\_drive

file\_format=(type=XML)

Unload as AVRO file,

copy into @data\_unload/taxi\_unload/AVRO\_

from

taxi\_drive

file\_format=(type=AVRO)

Unload as Json file

copy into @data\_unload/taxi\_unload/JSON\_

from

taxi\_drive

file\_format=(type=JSON)

Mention your observation below,

# Best practice.

Once you unload data to staging area from snowflake, it’s always a best practice to count

the records in snowflake table and records in snowflake staging area.

Count records in table and stage :

select count(\*) from taxi\_drive

Mention records count :

select count(\*) from @data\_unload/taxi\_unload

Mention records count :

List files in staging area,

list @data\_unload/taxi\_unload;

# 

# Unload only selected columns.

You can always unload only few selected column values.

You can apply filter condition and join conditions while doing unload.

Execute below copy command,

copy into @data\_unload/taxi\_unload/select\_

from

(

select

trip\_id,

call\_type

from

taxi\_drive

)

List unloaded files,

Text

Description automatically generated

Count records in table and stage :

select count(\*) from taxi\_drive

Mention records count :

select count(\*) from @data\_unload/taxi\_unload/select\_

Mention records count :

You should also note that, it’s possible to filter data, aggregate data and join data before doing upload.

You can try and check below copy command,

copy into @data\_unload/taxi\_unload/select\_

from

(

select

a.trip\_id,

b.call\_type

from

taxi\_drive a,

taxi\_drive b

where a.origin\_call = b.origin\_call

and a.call\_type =’Z’

limit 1000

)

OVERWRITE=’TRUE’

*Make a note of OVERWRITE option I used here. Try executing command without overwrite.*

# Download DATA to local system.

From web console you can download data up to ~100MB. If you want to download your table data more than this, then you have to use get command with SNOWSQL.

get @taxi\_unload/select\_ [file:///data-vol/unload/](../../../../data-vol/unload/)

Once downloaded, it’s always a best practice to remove files from the staged area.

Otherwise, storage cost will be added to snowflake bill.

Remove the files from staged location,

rm @data\_unload/taxi\_unload/select\_

# Calculate cost for this experiment.

WITH WAREHOUSE\_COST AS

(

select start\_time::date as usage\_date,

warehouse\_name,

sum(credits\_used) as total\_credits\_used,

sum(credits\_used) \* 3600 total\_active\_time,

(sum(credits\_used))\*1.94 COST\_IN\_DOLLAR

from snowflake.account\_usage.warehouse\_metering\_history

--where start\_time >= date\_trunc(day, current\_date)

group by 1,2

),

QUERY\_COST AS

(

select

QUERY\_TYPE,

SUM((TOTAL\_ELAPSED\_TIME/1000)) ACTIVE\_TIME,

SUM((TOTAL\_ELAPSED\_TIME/1000)\*0.0003+CREDITS\_USED\_CLOUD\_SERVICES) ACTUAL\_COST,

SUM((TOTAL\_ELAPSED\_TIME/1000)\*0.0003+CREDITS\_USED\_CLOUD\_SERVICES)\*1.94 COST\_IN\_DOLLAR

from table(information\_schema.QUERY\_HISTORY\_BY\_WAREHOUSE())

where TOTAL\_ELAPSED\_TIME>0

group by QUERY\_TYPE

),

ACTUAL\_GIVEN AS (

SELECT CRITERIA,ACTIVE\_TIME,COST,COST\_IN\_DOLLAR

FROM

(

SELECT 'ACTUAL' CRITERIA , SUM(ACTIVE\_TIME) ACTIVE\_TIME, SUM(ACTUAL\_COST) COST,SUM(COST\_IN\_DOLLAR) COST\_IN\_DOLLAR FROM QUERY\_COST

UNION

SELECT 'GIVEN' CRITERIA , SUM(total\_active\_time) ACTIVE\_TIME, SUM(total\_credits\_used) COST,SUM(COST\_IN\_DOLLAR) COST\_IN\_DOLLAR FROM WAREHOUSE\_COST

)

)

SELECT CRITERIA,ACTIVE\_TIME,COST,COST\_IN\_DOLLAR FROM ACTUAL\_GIVEN

UNION

SELECT 'IDEL\_TIME\_COST' CRITERIA , MAX(ACTIVE\_TIME)-MIN(ACTIVE\_TIME) ACTIVE\_TIME ,MAX(COST)-MIN(COST) COST,(MAX(COST)-MIN(COST))\*1.94 COST\_IN\_DOLLAR FROM ACTUAL\_GIVEN

Paste the screen shot of the output below,

What’s your observation please mention…….

**Quick note on warehouse.**

When you are using warehouse, always think of this analogy,

Spinning up warehouse and using it is always like , taking a ride in taxi. How you feel if your taxi meter is running when your taxi is waiting on a signal. You ask the driver and he says , It’s not the distance you travel cost you, if **engine** is on **meter** is on!!!

Listening his answer you instruct the driver, See, if you are waiting on a signal for more than 2 mins please turn off your engine.

Indirectly you are saying, if you are not driving or travelling turn your engine off.

Same way , if you are not querying or analyzing you should ask snowflake to turn of warehouse cluster.

Set your **Auto suspend time wisely** while creating warehouse.!!!!!!!!!!!!!

If warehouse is **ON** cost meter is **ON**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**QUESTIONS.**

In the above data upload scenario, you might have faced error while upload JSON data.

But it’s possible to upload table data in snowflake as JSON file.

Go to doc link,

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

Read through it and write down the copy command to upload table data as json file below,