

Date

# UNLOAD DATA FROM SNOWFLAKE

## 1. GOALS

In this assignment we will try to cover below scenarios,

- A. Creating named stage and unload data.
- B. Unload selected fields
- C. Best practice after doing unload.
- D. Download data to local system.
- E. Calculate overall cost for the activity we performed.

## 2. 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,

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;
```

```
pradeep#COMPUTE_WH@DEMO_DB.PUBLIC>list @taxi_unload;
```

name	size	md5	last_modified
taxi_unload/data_0_0_0.csv.gz	16819520	799bcfec1d78ae3ca44e26d78fbd3f0	Sun, 4 Oct 2020 02:22:45 GMT
taxi_unload/data_0_0_1.csv.gz	16779520	3b398638858612e11446f05bebc885d9	Sun, 4 Oct 2020 02:22:50 GMT
taxi_unload/data_0_0_2.csv.gz	16826320	d2ddae0475022b06bf8e5d274710b2e5	Sun, 4 Oct 2020 02:22:54 GMT
taxi_unload/data_0_0_3.csv.gz	13776144	128e6defbf7e19fbfcd4a260927824c7	Sun, 4 Oct 2020 02:23:00 GMT
taxi_unload/data_0_1_0.csv.gz	16789760	bcc683832b657a7563592e30b78f962f	Sun, 4 Oct 2020 02:22:45 GMT
taxi_unload/data_0_1_1.csv.gz	16801008	8afe034c9398ccb390ca7d15d9c321d8	Sun, 4 Oct 2020 02:22:50 GMT
taxi_unload/data_0_1_2.csv.gz	16815456	6ca5e2c66032dc24dc31ee8f61cfed0b	Sun, 4 Oct 2020 02:22:55 GMT
taxi_unload/data_0_1_3.csv.gz	12717872	1a319009ada73eb7e938770a3955a021	Sun, 4 Oct 2020 02:23:00 GMT
taxi_unload/data_0_2_0.csv.gz	16821744	410e8334cbe3be6eb70599d4f425cb1f	Sun, 4 Oct 2020 02:22:45 GMT
taxi_unload/data_0_2_1.csv.gz	16827760	12a68b0f98c6f68c3646b9184ea5ec52	Sun, 4 Oct 2020 02:22:50 GMT
taxi_unload/data_0_2_2.csv.gz	16814352	dcff21372c0a1e02f25b99627faf0d4c	Sun, 4 Oct 2020 02:22:54 GMT
taxi_unload/data_0_2_3.csv.gz	14116640	42390829e5b9e0eddb54916951e16ee0	Sun, 4 Oct 2020 02:23:00 GMT
taxi_unload/data_0_3_0.csv.gz	16827360	7d312b50ed0b7c8430d9f311d00a0de5	Sun, 4 Oct 2020 02:22:45 GMT
taxi_unload/data_0_3_1.csv.gz	16821728	de16a8e31e010efc52c26b343bafc8e3	Sun, 4 Oct 2020 02:22:50 GMT
taxi_unload/data_0_3_2.csv.gz	16820600	eddbee1b81dc5c1525c44635d5171727	Sun, 4 Oct 2020 02:22:55 GMT

### 3. 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,

```
pradeep#COMPUTE_WH@DEMO_DB.PUBLIC>list @taxi_unload/select_;
```

name	size	md5	last_modified
taxi_unload/select_0_0_0.csv.gz	1106736	cc47458f8efdbd0447391f3503661f5	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_1_0.csv.gz	1280896	746f74aab373d5c099f252c318cdfb1e	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_2_0.csv.gz	1119200	937ac297971c6e52b2399135b559028c	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_3_0.csv.gz	1139168	5e260674955c5584a19e30aed03b24a8	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_4_0.csv.gz	1105360	17207222406196336bfea58607b5c150	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_5_0.csv.gz	1116992	95d26be9a032835c652ab27e16777b43	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_6_0.csv.gz	1120864	61a812479359383581f6d52daaa96cc0	Sun, 4 Oct 2020 02:35:37 GMT
taxi_unload/select_0_7_0.csv.gz	1091040	efe29386bc908a1ebb5a5dbbc92fd781	Sun, 4 Oct 2020 02:35:37 GMT

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.*

#### 4. 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/
```

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_`

5. 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 'DEL_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,

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](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,

--