Loading Data

Sujith Nair
Cloud Data Architect
Snowflake Snowpro Certified

Learn2Cloud Data Solutions

Do you use transformation when loading data using the copy command ?

We generally don't use transformation when loading data to the RAW tables with the copy command, the reason for this is because this tends to slow down the loads on large files. We prefer to bring the data into snowflake and then do the necessary transformation in SQL.

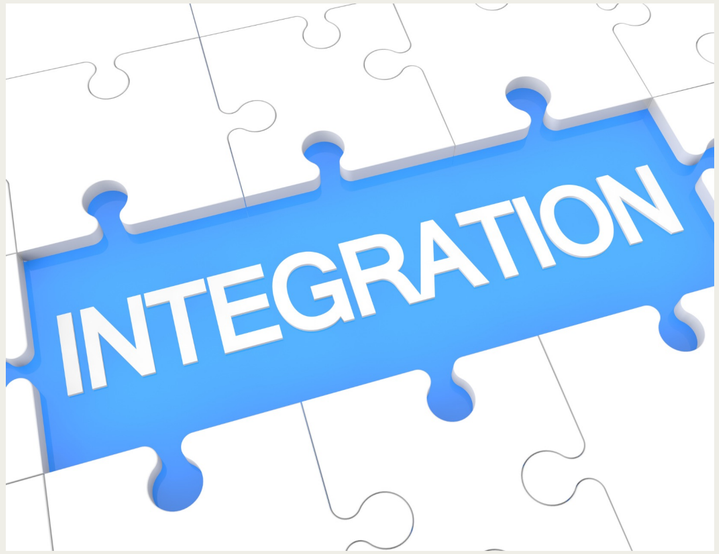
Exception:
In some cases, we may receive very large files with over 1000 columns , in this situation we may choose to load a subset of fields into the target raw table.
In this scenario we need to use select in the COPY command and may need to apply transformations.

```
COPY INTO CUSTOMER_RAW(C_NAME, C_PHONE)
FROM (SELECT TRIM(C_NAME),C_PHONE FROM @AWSSTAGE);
```

What is needed to establish connectivity between snowflake and cloud storage(s3 and Azure Blob Storage) ?

We need to create a storage integration object. The storage integration object helps us establish connectivity between cloud storage and snowflake and we can use the Copy command to copy data from s3 to snowflake tables.

```
CREATE STORAGE INTEGRATION <integration_name>
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = 'S3'
  ENABLED = TRUE
  STORAGE_AWS_ROLE_ARN = '<iam_role>'
  STORAGE_ALLOWED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/')
  [ STORAGE_BLOCKED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/') ]
```



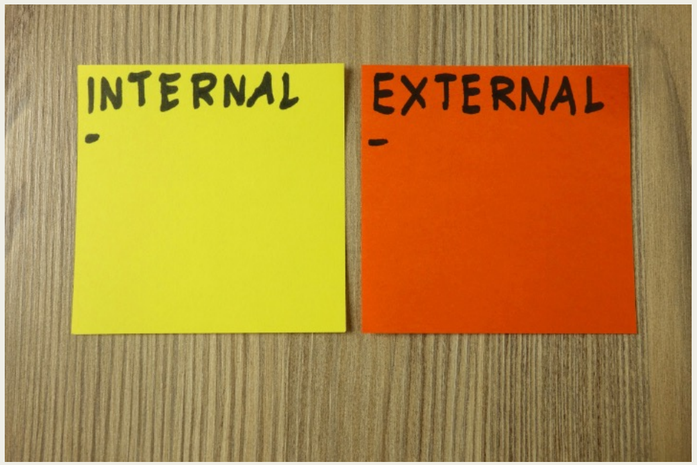
#How can we ingest data without using snowpipe, COPY or ETL tools like DBT ?

We can ingest data into snowflake by creating external tables on cloud storage folders, this gives us select access to data files and we can use the

INSERT into target_table SELECT * FROM external_table

to get the data into snowflake.
This method is very useful when you have large files of greater than 250 MB which are not suitable for loading with snowpipe.

Create streams on external tables to ingest data in near real time.



You received files from your new vendor, you are aware of the layout of the file as they shared the details in the past.
A target table was created with the file structure.
Before loading the file to the target table how will you validate if the file provided will load without any issues to the target table ?

We can use the VALIDATION_MODE=RETURN_ALL_ERRORS in the COPY command to see if there are any errors the file might encounter when loaded into the target table. When the COPY command is run in VALIDATION_MODE data is not loaded to the target table.

```
COPY INTO CUSTOMER_RAW
FROM @AWSSTAGE
VALIDATION_MODE=RETURN_ALL_ERRORS
```

How will you setup a process of automated validation and quality control of all files received from your source system before loading them to snowflake ?

When running jobs on a regular basis, we can validate files by using python to read the files from s3 and check that the important fields and values are populated and to throw an error if the validation fails.

The second option is to ingest the data to snowflake raw tables and run SQL to validate if the file is good before loading then further.

We can also use the SKIP_FILE option to not load files that have load errors when loading them.

#Once a file is loaded into a table with the COPY command, can you reload the file again ?

If you try to reload the file, snowflake will ignore the file.
Snowflake stores file load metadata which is associated with the table and is aware that the file is already loaded and will stop reload to prevent duplicates.

This metadata is re-set when the table is truncated or dropped and re-created.

#How can you reload the file, even if it is already loaded ?

I will use the FORCE option in the COPY command to force load the files.

Use Case:
When you need a large volume of data for performance testing.

```
COPY INTO CUSTOMER_RAW FROM @AWSSTAGE
FORCE = TRUE
```

What is the data loading strategy in your project?

We use Tasks/Airflow and stored procedures to ingest when data needs to be loaded in batch mode. Merge statements inside the stored procedure are used to update data in the target table.

For tables that need to be updated near real time we use task and streams. The task detects data in the stream and SQL statement associated with the task will update the target table. This SQL is generally a merge statement which uses the stream as input.

We use snowpipe for loading files that need to be ingested immediately and are less than 250 MB.IOT data is ingested in this way.

We also use external tables to ingest files that are over 1 GB in size, we use SELECT statement to get the data into snowflake.

Streams on External tables are used when files need to be ingested without any delay

When ingesting data from data lake, are most of your files in parquet format or csv and what is the reason for that?

Data in s3/Azure Blob Storage is stored in parquet format, this is our enterprise standard for storing data in s3/Azure Blob Storage and the data lake. Parquet format helps conserve space and stores data in columnar format which is more efficient when we select a subset of columns from Athena/Azure Data Lake Analytics.

While csv files tend to load faster than parquet files with the COPY command when loading snowflake, we were not interested in re-creating the parquet files for ingestion into snowflake as this would consume a lot of time and compute resources and lead to duplication of data.

Thank you!

Learn2CloudData Solutions