Chapter 8: Data Loading

Eighth Chapter of the SnowFlake SnowPro Core Certification Complete Course.

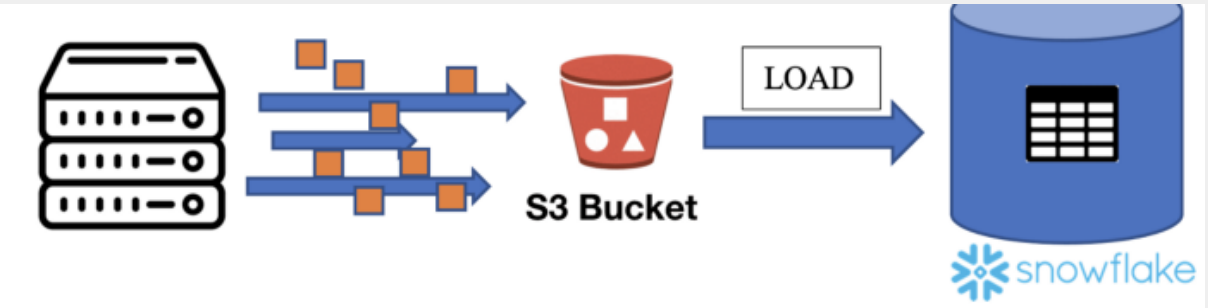
Hello everybody. In this chapter, we are going to study the different ways to COPY data into Snowflake tables, either using bulk load or continuous load with SnowPipe:

Data Loading

When we load the data into Snowflake, we usually perform the following steps:

- Source system ⟶ Snowflake Stage ⟶ Snowflake Table.

A source system (it can be, for example, your application, or not even that, it can be whatever machine that generates data, like sensors) will generate data that it will send to a stage, for example, AWS S3. Once the data is in any stage, we'll copy it into Snowflake tables. There are two different ways to do it, Bulk Load or Continuous Load; let's study the differences.



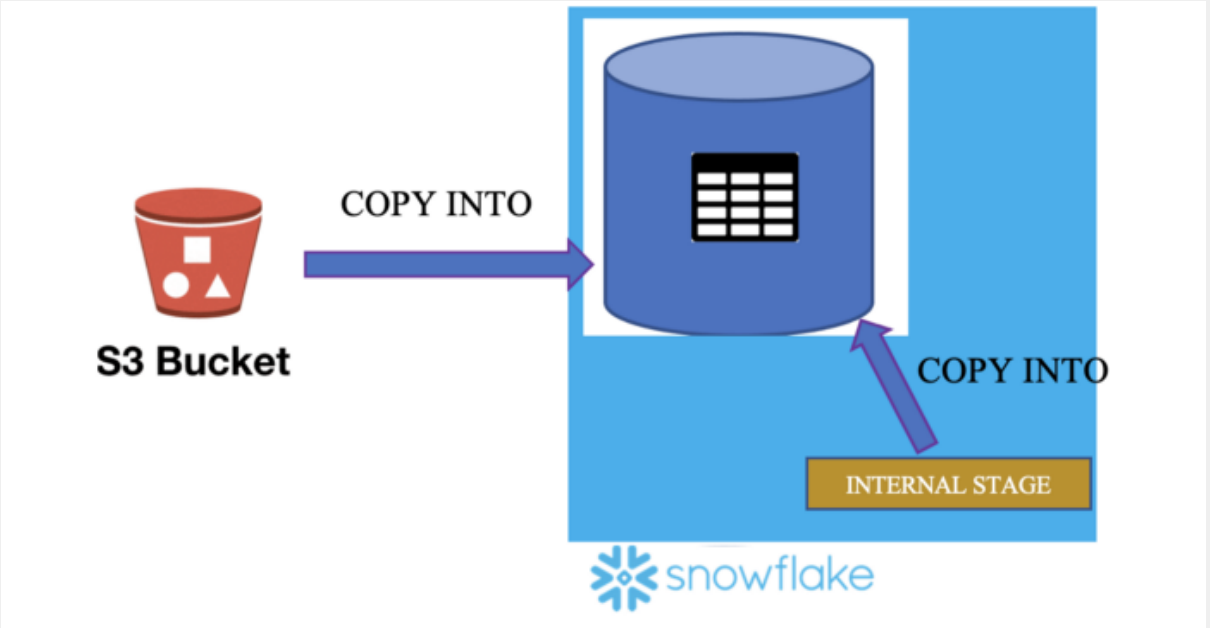
Data Loading Steps in SnowFlake.

BULK LOAD

Bulk load is the process of loading batches of data from files already available at any stage into Snowflake tables. We use the COPY command to do that (it consumes credits from the virtual warehouses). It supports data transformation while loading, using column reordering, column omission, casting... Let's see how the COPY INTO command works.

COPY INTO

You can load data from staged files to an existing table using this command. It copies the data into tables. It works for any stage (named internal, named external, table and user stages), or even from an external location like S3, Azure... Some factors affect the loading time, like the physical location of the stage, GZIP compression efficiency (files are automatically compressed using gzip unless compression is explicitly disabled), or the number and types of columns.



Copy Into command used to copy data from any stage into tables.

You can both load and unload data into tables with this command:

- LOAD → COPY data from the stages into a table. You load data into Snowflake tables.
- UNLOAD → COPY data from the table to stages. You unload data from the tables to a different location (internal stage, external stage, or external location).

Load vs. Unload Data into Snowflake

Some extra considerations:

- You need to specify the name of the table where you want to copy the data, stage where the files are, file/patterns that you want to copy, and file format.
- 64 days of metadata. The information about the loaded files is stored in Snowflake metadata. It means that you cannot COPY the same file again in the next 64 days unless you specify it ("FORCE=True" command). We can see that in the following diagram:
- You cannot Load/Unload files from your Local Drive
- Some transformations like Flatten, Join, Group by, Filters or Aggregations are not supported.

- Using the Snowflake UI, you can only Load 50MB files. You can copy bigger files using SnowSQL.
- Organizing input data by granular path can improve load performance.

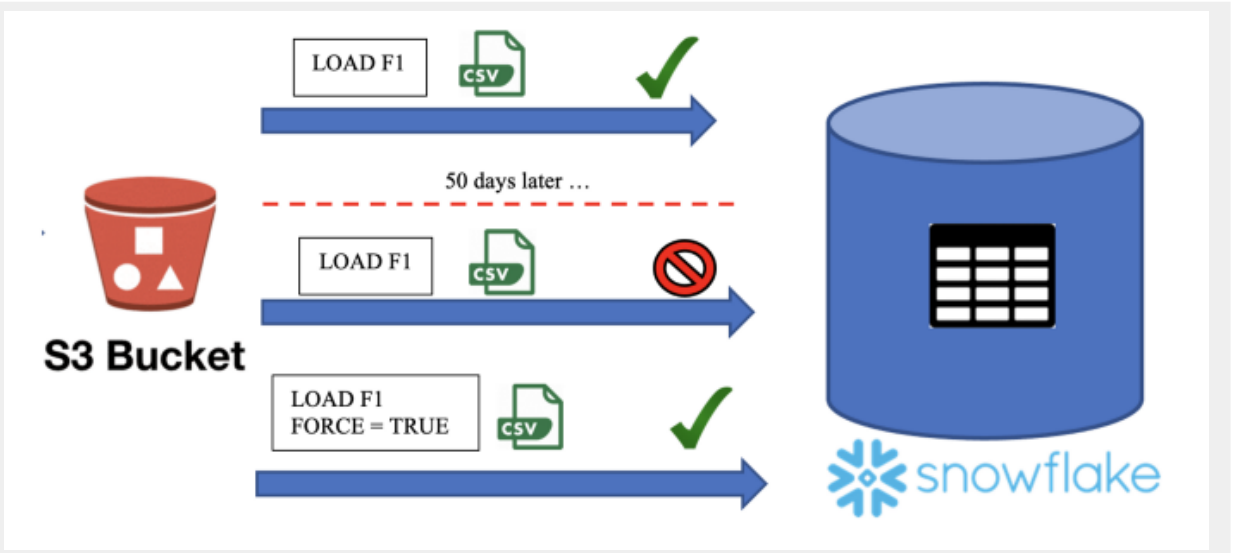
ON ERROR:

Loading some of the files might produce errors. For example, you are copying .csv files, and the data is incorrect inside the file. There are several options that you can specify in this case:

1. **ABORT_STATEMENT** → Abort the load operation if there are errors in a data file. If you don't specify any parameter in the ON_ERROR option, this will be the VALUE BY DEFAULT.
2. **CONTINUE** → Continue loading the file.
3. **SKIP_FILE** → Skip file if there are errors in the files.
4. **SKIP_FILE_num** → Skip file when the number of error rows in the file equals or exceeds the specified number.
5. **SKIP_FILE_num%** → Skip file when the percentage of error rows in the file exceeds the specified percentage.

Some other options from the COPY INTO command that normally appears in the exam:

- **pattern** = <pattern> → Load files from a stage into the table, using pattern matching.
- **FORCE** = TRUE → Once the files are copied into a table, they cannot be copied again in the next 64 days because of the files' metadata. If this option is true, it loads all files, regardless of whether they've been loaded previously and have not changed since they were loaded.



COPY INTO 64 Days Metadata

- PURGE = TRUE → It specifies whether to automatically remove the data files from the stage after the data is loaded successfully. If the purge operation fails for any reason, no error is returned. An excellent way to check whether there was an error or not would be listing the files from the stage with the "LIST stage" command.
- MAX_FILE_SIZE → You can specify the maximum size for each file when unloading the data with this option.

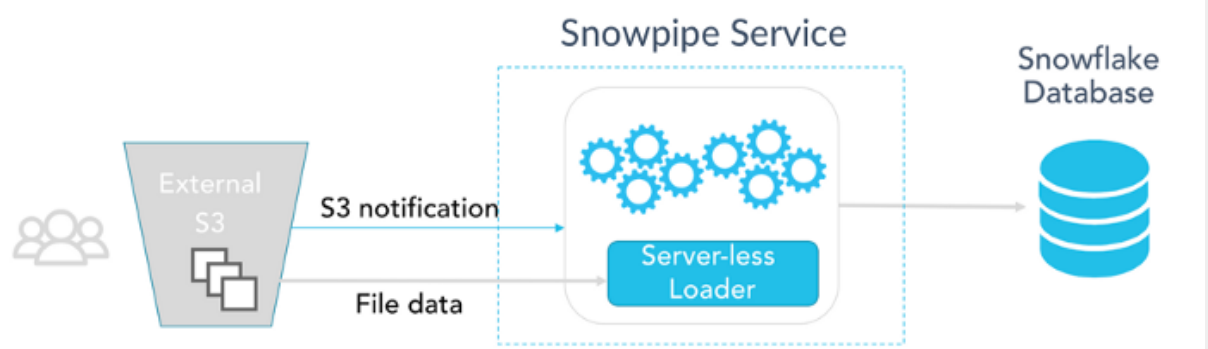
CONTINUOUS LOAD

Load small volumes of data (micro-batches) and incrementally make them available for analysis. There are different ways to do that:

1. SnowPipe → The easiest and most popular way to do it. We'll see it in this chapter.
2. Snowflake Connector for Kafka → Reads data from Apache Kafka topics and loads the data into a Snowflake table.
3. Third-Party Data Integration Tools → You can do it with other supported integration tools. You can see the list at the following link.

SNOWPIPE

SnowPipe enables loading data when the files are available in any (internal/external) stage. You use it when you have a small volume of frequent data, and you load it continuously (micro-batches). SnowPipe is serverless, which means that it doesn't use Virtual Warehouses. It is used for Streaming / Near Real-Time data. An important thing to know is that SnowPipe does not guarantee that files will be loaded in the same order as they are staged. It usually processes the oldest files first, but there is no guarantee.



How SnowPipe works (via snowflake.com)

The real question is how can SnowPipe detect new files in the stage? There are two different ways to do it:

1. Automating SnowPipe using cloud messaging → You can do a trigger when there is an event in the stage (for example, when there is a new document) using Amazon SQS (Simple Queue Service) notifications for an S3 bucket. You can follow this tutorial if you are interested in learning how to do it.
2. Calling SnowPipe REST endpoints → Your client application calls a public REST endpoint with the name of a pipe object and a list of data filenames. It requires key pair authentication with JSON Web Token (JWT), and you have the "insertFiles", "insertReport", and "loadHistoryScan" APIs to do it. You have more information at the following link.

Some extra considerations:

- 14 days of metadata (COPY INTO was 64 days). You cannot copy the same files again in these 64 days.
- By default, it does a "SKIP_FILE" when there is an error loading files. COPY INTO default ON_ERROR option was ABORT_STATEMENT.

TYPICAL SNOWPRO CORE EXAM QUESTIONS

1. What are the usual data loading steps in Snowflake?

1. Source ⟶ Snowflake Stage ⟶ Snowflake table
2. Source ⟶ Snowflake Table ⟶ Snowflake Stage
3. Snowflake Table ⟶ Source ⟶ Snowflake Stage

Solution: 1

2. What key concepts will need to be considered while loading data into Snowflake?

1. Stage
2. File Format
3. Transformation
4. File Size
5. Error validation

Solution: 1, 2, 3, 5.

3.Using COPY INTO <location> command, you can unload data from a table into which locations?

1. Named internal stage (or table/user stage).
2. Named external stage that references an external location (Amazon S3, Google Cloud Storage, or Microsoft Azure).
3. An external location like Amazon S3 or Azure.
4. Local Drive

Solution: 1, 2, 3. Once the data is in the internal stage, you can download them into your local drive using the GET command. You can also unload data into an external location.

4. After how many days does the load history of the COPY command expire?

1. 1 day
2. 14 days
3. 64 days
4. 180 days

Solution: 3

5.While loading data through the COPY command, you can transform the data. Which of the below transformations are allowed?

1. Truncate columns
2. Omit columns
3. Filters
4. Reorder columns
5. Cast

6. Aggregate

Solution: 1, 2, 4, 5

6. What option will you specify to delete the stage files after a successful load into a Snowflake table with the COPY INTO command?

1. DELETE = TRUE
2. REMOVE = TRUE
3. PURGE = TRUE
4. TRUNCATE = TRUE

Solution: 3.

7. In which of the below scenarios is SnowPipe recommended to load data?

1. We have a small volume of frequent data
2. We have a huge volume of data generated as part of a batch schedule
3. In both of the previous scenarios

Solution: 1

8. Is SnowPipe Serverless?

1. True
2. False

Solution: 1

9. After how many days does the load history of SnowPipe expire?

1. 1 day
2. 14 days
3. 90 days
4. 180 days

Solution: 2

10. Can SnowPipe load a file with the same name if it has been modified later?

1. True
2. False

Solution: 2. This is because of the SnowPipe metadata. Changing the name doesn't modify this metadata, so it won't be copied.

11. Does SnowPipe guarantee that files are loaded in the same order they are staged?

1. True
2. False

Solution: 2. It usually processes the oldest files first, but there is no guarantee.

12. Which of the below APIs are SnowPipe REST APIs?

1. insertFiles
2. insertReport
3. insertHistoryScan
4. loadFiles
5. loadHistoryScan

Solution: 1, 2, 5

13. Which data-loading method requires a user-specified warehouse to execute COPY statements?

1. Bulk Data Load
2. SnowPipe
3. Both

Solution: 1

Thanks for Reading!