To implement **incremental loading** using **Snowpipe** triggered by a **Snowflake Task**, the goal is to load only new or updated data (i.e., incremental data) into Snowflake. Typically, Snowpipe loads new files placed in a cloud storage location, such as S3 or Azure Blob Storage, and a Snowflake Task can automate and monitor this process.

Here’s an example that demonstrates how to achieve incremental data loading by combining **Snowpipe** and **Snowflake Task**:

**Assumptions**

* New files are added to an external stage (e.g., an S3 bucket) periodically.
* The filenames or data contain a timestamp or unique identifier that allows Snowpipe to load only the new data.
* The incremental load depends on these new files being loaded into Snowflake.

**Workflow Overview**

1. **Data lands in S3 bucket.**
2. **Snowpipe** is configured to load new files into a staging table in Snowflake.
3. A **Snowflake Task** runs periodically to:
   * **Identify and load incremental data** from the staging table into the target table.
   * **Clear the staging table** after the incremental load.

Step-by-Step Example

**1. Create Target Table**

This is the table where your final, incrementally loaded data will be stored.

CREATE OR REPLACE TABLE sales (

id INT,

product STRING,

amount DECIMAL(10,2),

sale\_date DATE

);

**2. Create Staging Table**

This table temporarily holds the data loaded by Snowpipe. The task will move new data from here to the target table.

CREATE OR REPLACE TABLE sales\_staging (

id INT,

product STRING,

amount DECIMAL(10,2),

sale\_date DATE,

load\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP()

);

The *load\_timestamp* column helps you identify when data was loaded.

**3. Create an External Stage (e.g., S3)**

You need a stage that points to your S3 bucket where the new data files are stored.

CREATE OR REPLACE STAGE my\_s3\_stage

URL = 's3://my-bucket/path/'

CREDENTIALS = (AWS\_KEY\_ID = 'your\_aws\_key\_id' AWS\_SECRET\_KEY = 'your\_aws\_secret\_key')

FILE\_FORMAT = (TYPE = 'CSV' SKIP\_HEADER = 1 FIELD\_OPTIONALLY\_ENCLOSED\_BY = '"');

This Snowpipe will *continuously load data into the staging table whenever new files are added to the S3 stage*.

**5. Create a Snowflake Task for Incremental Load**

The Snowflake Task will periodically run to:

* Move the new data from the staging table to the final target table.
* Optionally remove the data from the staging table.

Here's how you can set up a Snowflake Task to do this:

CREATE OR REPLACE TASK incremental\_load\_task

WAREHOUSE = my\_wh

SCHEDULE = 'USING CRON 0 \* \* \* \* UTC' -- Runs every hour

COMMENT = 'Task to incrementally load new data from staging table to target table'

AS

-- Insert new data from sales\_staging to sales, filtering by the load\_timestamp

INSERT INTO sales (id, product, amount, sale\_date)

SELECT id, product, amount, sale\_date

FROM sales\_staging

WHERE load\_timestamp > (

SELECT COALESCE(MAX(load\_timestamp), '1970-01-01'::TIMESTAMP) FROM sales

);

-- Optionally, clear the staging table after loading new data

DELETE FROM sales\_staging

WHERE load\_timestamp <= (

SELECT COALESCE(MAX(load\_timestamp), '1970-01-01'::TIMESTAMP) FROM sales

);

 **SCHEDULE = 'USING CRON 0 \* \* \* \* UTC'**: This task runs every hour, but you can modify the cron schedule as per your requirements.

 **INSERT INTO sales**: This query selects data from sales\_staging where the load\_timestamp is greater than the last load, ensuring only new data is loaded into sales.

 **DELETE FROM sales\_staging**: This is optional. It clears the staging table after the data is processed, ensuring that the table only contains new, unprocessed data for the next task execution.

**6. Enable the Task**

Enable the task so that it begins running according to the schedule.

ALTER TASK incremental\_load\_task RESUME;

**7. Monitor the Task and Snowpipe**

You can monitor the status of the task and Snowpipe execution:

* **Check task execution history**:

SELECT \* FROM table(information\_schema.task\_history(task\_name => 'incremental\_load\_task'));

* **Check Snowpipe load history:**

SELECT \* FROM table(information\_schema.copy\_history(table\_name => 'sales\_staging', start\_time => dateadd('hour', -1, current\_timestamp())));

**Summary of the Process:**

1. **Snowpipe automatically loads new files into the staging table (sales\_staging) from an S3 bucket or external stage.**
2. **Snowflake Task runs on a schedule to:**
   * **Load new and incremental data from the staging table to the target table (sales).**
   * **Optionally clean up the staging table after the data has been moved.**
3. **The pipeline is fully automated to continuously handle incremental data loads into Snowflake.**