**1. Download raw Twitter data using the Tweepy API class:**

The Tweepy API class is a Python library that can be used to access the Twitter API. To download raw Twitter data, you will need to create an instance of the Tweepy API class and then use the search method to search for tweets that match your criteria.

**import tweepy**

**# Create an instance of the Tweepy API class**

**auth = tweepy.OAuthHandler("CONSUMER\_KEY", "CONSUMER\_SECRET")**

**auth.set\_access\_token("ACCESS\_TOKEN", "ACCESS\_TOKEN\_SECRET")**

**api = tweepy.API(auth)**

**# Search for tweets that contain the word "Snowflake"**

**tweets = api.search(q="Snowflake")**

**# Print the tweets**

**for tweet in tweets:**

**print(tweet.text)**

**2.Create database objects:**

Once we have downloaded the raw Twitter data, we need to create initial database objects in Snowflake. This includes creating a database, a schema, and a table.

**CREATE DATABASE twitter\_data;**

**USE twitter\_data;**

**CREATE SCHEMA raw\_data;**

**CREATE TABLE raw\_data.tweets (**

**id INT,**

**text VARCHAR(255),**

**created\_at TIMESTAMP**

**);**

**3. Load raw data into those database:**

Once we have created the database objects, we can load the raw Twitter data into them using the **copy into** command.

**COPY INTO raw\_data.tweets**

**FROM '/path/to/twitter\_data.json'**

**FILE\_FORMAT = JSON;**

**4.Institute change tracking on your staging table:**

Change tracking is a feature in Snowflake that allows we to track changes to data in a table. This can be useful for ensuring that downstream tables are always up-to-date.

To institute change tracking on a table, we can use the **ALTER TABLE** command.

**ALTER TABLE raw\_data.tweets**

**ADD CHANGE TRACKING;**

**5.Create and load downstream tables:**

Once we have loaded the raw Twitter data into the staging table, we can create and load downstream tables. Downstream tables are tables that are created from the data in the staging table.

**CREATE TABLE processed\_data.tweets (**

**id INT,**

**text VARCHAR(255),**

**created\_at TIMESTAMP,**

**sentiment VARCHAR(255)**

**);**

We can then load the data from the staging table into the downstream table using the COPY INTO command.

**6.** **Create task dependencies to ensure downstream data is loaded:**

Task dependencies are used to ensure that downstream tasks are not executed until the upstream tasks have completed successfully. This can be useful for ensuring that downstream data is always up-to-date.

To create task dependencies, we can use the CREATE TASK command.

**CREATE TASK load\_processed\_data**

**DEPENDS ON load\_raw\_data;**