Contents

[TO DO CHECKLIST 1](#_Toc202968421)

[1. Check properties in the GetFile 1](#_Toc202968422)

[2. Review the answers below by ChatGPT 1](#_Toc202968423)

[**Hive Table Creation** 1](#_Toc202968424)

[Q. 1](#_Toc202968425)

[A. 2](#_Toc202968426)

[Execute Analytical Queries 4](#_Toc202968427)

[--1. Retrieve the top 5 users by followers’ count. (1 pt) 4](#_Toc202968428)

[--2. Identify the top 25 most used languages. (1 pt) 4](#_Toc202968429)

[--3. Count the number of unique screen\_name values. (1 pt) 4](#_Toc202968430)

[--4. Extract the top 25 hashtags and their counts. # hashtags is an array of values 4](#_Toc202968431)

[Q. 5](#_Toc202968432)

[A. 5](#_Toc202968433)

## TO DO CHECKLIST

### Check properties in the GetFile

* Also check “Run Schedule” in ***the Scheduling Tab*** for GetFile configure processor
* In the properties tab check the below:
  + FileFilter to pick only files of interest (gzip files)
  + ***Optional:*** “**Setting Up GetFile Processor**: The GetFile processor is configured to read files from a specified directory in the local file system and delete them after reading to avoid duplication“

### Review the answers below by ChatGPT

## **Hive Table Creation**

### Q.

Ok - thanks for clarifying. That was helpful. Now please provide steps and code for the below related to Hive Table Creation:

\* Create a Hive User Managed table to read the parquet data.

• Compute and report:

o Table statistics.

o Total number of rows

o Total storage size.

### A.

You're very welcome! Here's a full guide to help you **create a Hive user-managed table** from existing **Parquet data**, and then compute and report table statistics including **row count** and **storage size**.

**✅ Step-by-Step: Hive Table Creation & Statistics**

#### 🔹 Step 1: Prepare the Parquet Data Location

Let’s say your Parquet files are stored in:

/user/hive/warehouse/parquet\_data/tweets\_data.parquet

(You can also have a folder with multiple parquet files.)

#### 🔹 Step 2: Create Hive User-Managed Table (External Table)

Use EXTERNAL so Hive treats this as user-managed (i.e., data isn't deleted when the table is dropped).

##### 🧠 Option 1: Let Hive read the schema from the Parquet files (Schema-on-read)

CREATE EXTERNAL TABLE tweets\_data

STORED AS PARQUET

LOCATION '/user/hive/warehouse/parquet\_data/';

✅ This works **if the Parquet files contain embedded schema metadata.**

##### 🧠 Option 2: Explicitly define the schema (Recommended)

CREATE EXTERNAL TABLE tweets\_data (

lang STRING,

retweet\_count INT,

userid STRING,

location STRING,

followers\_count INT,

screen\_name STRING,

friends\_count INT,

hashtags ARRAY<STRING>

)

STORED AS PARQUET

LOCATION '/user/hive/warehouse/parquet\_data/';

#### 🔹 Step 3: Compute Table Statistics

Use the following command to gather statistics (row count, data size, etc.):

ANALYZE TABLE tweets\_data COMPUTE STATISTICS;

✅ This collects **basic table statistics**, such as:

* Number of rows
* File size
* Number of files
* etc.

#### 🔹 Step 4: Get Total Number of Rows

SELECT COUNT(\*) FROM tweets\_data;

This gives you an exact row count. (The value may also be available after ANALYZE depending on Hive version.)

#### 🔹 Step 5: Get Storage Size (Table Size on Disk)

Use this command in Hive CLI or Beeline:

SHOW TABLE EXTENDED LIKE 'tweets\_data';

Look in the output for fields like:

location: hdfs://.../parquet\_data

inputFormat: org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat

totalSize: 12345678

numRows: 9876

You’ll get:

* totalSize → size in **bytes**
* numRows → row count (if collected)

You can also check from HDFS directly:

hdfs dfs -du -h /user/hive/warehouse/parquet\_data/

**✅ Summary of Commands**

-- Step 1: Create Hive external table

CREATE EXTERNAL TABLE tweets\_data (

lang STRING,

retweet\_count INT,

userid STRING,

location STRING,

followers\_count INT,

screen\_name STRING,

friends\_count INT,

hashtags ARRAY<STRING>

)

STORED AS PARQUET

LOCATION '/user/hive/warehouse/parquet\_data/';

-- Step 2: Compute statistics

ANALYZE TABLE tweets\_data COMPUTE STATISTICS;

-- Step 3: Count rows

SELECT COUNT(\*) FROM tweets\_data;

-- Step 4: Show storage size and metadata

SHOW TABLE EXTENDED LIKE 'tweets\_data';

Let me know if you're working with **Hive-on-Tez**, **Hive-on-Spark**, or want to **store this table in ORC** instead for better performance and compression.

## Execute Analytical Queries

--lang, retweet\_count, userid, location, followers\_count, screen\_name, friends\_count, hashtags.

### --1. Retrieve the top 5 users by followers’ count. (1 pt)

Select userid, followers\_count

from tweets\_data

ORDER BY followers\_count DESC

LIMIT 5

### --2. Identify the top 25 most used languages. (1 pt)

Select lang, count(\*) as used\_lang

from tweets\_data

GROUP BY lang

ORDER BY used\_lang DESC

LIMIT 25

### --3. Count the number of unique screen\_name values. (1 pt)

Select count(distinct(screen\_name)) from tweets\_data

### --4. Extract the top 25 hashtags and their counts. # hashtags is an array of values

SELECT single\_hashtag, COUNT(\*) AS hashtag\_usage\_count

FROM tweets\_data,

UNNEST(hashtags) AS t(single\_hashtag)

--UNNEST(SPLIT(hashtags, ',')) AS t(single\_hashtag)

GROUP BY single\_hashtag

ORDER BY hashtag\_usage\_count DESC

LIMIT 25;

### Q.

Assume the role of a seasoned data engineer who has experience working with Apache Superset, please provide step-by-step on how to use Apache Superset on to work on and achieve below task:

Task 3: Dashboard Creation in Superset

Build a dashboard in Apache Superset that visualizes the insights obtained above. Include at least the following three charts:

Chart 1: Distribution of tweets by screen\_name.

Chart 2a: Top 5 users by followers count.

Chart 2b: Top 25 most used languages.

Chart 3: Top 25 hashtags.

### A.

Certainly! Here's a detailed, step-by-step guide — from the perspective of a **seasoned data engineer** — on how to use **Apache Superset** to create a dashboard that includes the charts you listed.

#### ✅ 🧱 Prerequisites

Before you start, make sure:

* Apache Superset is installed and running (locally or on a server).
* You have **access to the Hive table** or **Trino connection** in Superset that contains the tweets\_data.
* The data source (Hive/Trino) is already added in **Data > Databases**.

#### 🚀 Step-by-Step Guide: Dashboard Creation in Superset

##### 🔹 Step 1: Connect to the Dataset

1. Go to **Data > Datasets**.
2. Click **+ Dataset**.
3. Select the database (Hive or Trino).
4. Choose the schema and select the tweets\_data table.
5. Click **Add**.

Now your dataset is ready for visual exploration.

##### 🔹 Step 2: Create Charts

**📊 Chart 1: Distribution of Tweets by screen\_name**

1. Go to **Charts > + Chart**.
2. Select your dataset: tweets\_data.
3. Choose chart type: **Bar Chart** (or Pie Chart if preferred).
4. Click **Create New Chart**.
5. In the chart builder:
   * **X Axis / Dimension**: screen\_name
   * **Metric / Y Axis**: COUNT(\*)
   * **Sorting**: Order descending by count
   * **Limit**: Top 10 or Top 20 (optional, for readability)
6. Click **Run**.
7. Customize the chart title: e.g., "Tweets per User (screen\_name)".
8. Click **Save**, and name it something like: Tweets by screen\_name.

###### 📊 Chart 2a: Top 5 Users by Followers Count

1. Again, **+ Chart > tweets\_data dataset**.
2. Select chart type: **Bar Chart** or **Table**.
3. Chart settings:
   * **Dimension**: screen\_name
   * **Metric**: MAX(followers\_count) (or just followers\_count if pre-aggregated)
   * **Sorting**: Order by followers descending
   * **Limit**: 5
4. Run the query and preview.
5. Title the chart: "Top 5 Users by Followers Count".
6. Save the chart.

###### 📊 Chart 2b: Top 25 Most Used Languages

1. Create another chart: **Bar Chart** or **Pie Chart**.
2. Dimension: lang
3. Metric: COUNT(\*)
4. Sorting: by count, descending
5. Limit: 25
6. Run, title, and save: "Top 25 Languages Used in Tweets".

###### 📊 Chart 3: Top 25 Hashtags

**Assuming hashtags column is an array and already exploded (flattened) into a separate table or virtual dataset via SQL Lab.**

**Option 1: Prepare virtual dataset using SQL Lab**:

1. Go to **SQL Lab > SQL Editor**.
2. Write:

SELECT hashtag, COUNT(\*) AS count

FROM (

SELECT UNNEST(hashtags) AS hashtag

FROM tweets\_data

) AS flat\_hashtags

GROUP BY hashtag

ORDER BY count DESC

LIMIT 25;

1. Run the query.
2. Click **Explore** (top-right) to create a chart from this result.
3. Choose chart type: **Bar Chart**
4. Dimension: hashtag
5. Metric: count
6. Save as "Top 25 Hashtags".

**Alternative**: You can also **create a View in Hive/Trino** that does the unnesting and register it as a dataset.

##### 🔹 Step 3: Create a Dashboard

1. Go to **Dashboards > + Dashboard**.
2. Enter dashboard name: Tweets Insights Dashboard.
3. Save it.

##### 🔹 Step 4: Add Charts to the Dashboard

1. Open the dashboard.
2. Click **Edit Dashboard**.
3. Use the drag-and-drop layout builder:
   * Add the charts you saved earlier:
     + Tweets by screen\_name
     + Top 5 Users by Followers Count
     + Top 25 Languages
     + Top 25 Hashtags
4. Rearrange charts as needed for clarity.
5. Optionally:
   * Add **filters** (e.g., by language, location).
   * Add **titles, markdown, or text boxes** to explain insights.
6. Click **Save**.

**📌 Final Dashboard Should Include:**

| **Chart** | **Description** |
| --- | --- |
| **Chart 1** | Bar chart showing number of tweets per screen\_name |
| **Chart 2a** | Top 5 users by highest followers\_count |
| **Chart 2b** | Top 25 languages used in tweets |
| **Chart 3** | Top 25 most frequent hashtags (unnested) |

**💡 Pro Tips**

* Use **filters** in the dashboard to allow interaction (e.g., by lang or screen\_name).
* Add **tooltips** and labels for readability.
* Use **horizontal bar charts** for better label alignment on long category names.

Let me know if you'd like help with:

* Automating chart updates
* Connecting Superset to a live stream (e.g., Kafka or Flink backend)
* Creating scheduled email reports or alerts from dashboards