SNOWFLAKE CLUSTERING INFORMATION

What is total_constant_partition_count?

Definition (simplified):

> The **number of micro-partitions** in a table where the **clustering is already optimal** and **doesn't need further improvement**.

In more technical terms:

- These are **"constant" micro-partitions** — meaning, for the chosen clustering key(s), the data in these micro-partitions is already **well-organized** (i.e., all rows have the same or similar values for the clustering column).
- Because of this, **Snowflake can easily prune these partitions** during query execution, improving **performance and reducing cost**.
- These partitions **won't benefit much from reclustering**.

Why is it useful?

A high total_constant_partition_count means:

- **Efficient query performance** — Snowflake can **prune more partitions**, scanning only what's needed.
- **Lower compute cost** — since fewer partitions are scanned.
- **No need to recluster** — you're not wasting compute on partitions that are already optimized.

Conversely:

- A low value here indicates **room for optimization** via reclustering.

Use Case LinkedIn: Activity Feed Table

Problem:

LinkedIn stores billions of activity feed items in a large table: e.g., posts, likes, shares, job changes.

```
CREATE TABLE activity_feed (
    user_id STRING,
    activity_type STRING,
    activity_time TIMESTAMP,
    content STRING,
    job_id STRING,
    post_id STRING
);
```

They **cluster the table by activity_time**, so queries like:

```
SELECT * FROM activity_feed
WHERE activity_time BETWEEN '2025-07-01' AND '2025-07-05';
```

...run efficiently by **pruning partitions** based on time.

Monitoring System Clustering:

Snowflake provides system views like TABLE_STORAGE_METRICS and CLUSTERING_INFORMATION to track clustering efficiency:

```
SELECT
  table_name,
  clustering_key,
  total_partition_count,
  total_constant_partition_count
FROM information_schema.table_storage_metrics
WHERE table_name = 'ACTIVITY_FEED';
```

Scenario 1: Good Optimization

If:
total_partition_count = 10,000
total_constant_partition_count = 8,000

RESULT: 80% of the table is **already well-clustered** and **won't benefit** from more clustering.
Action: Reclustering is likely **not needed frequently**. You're saving compute while still getting strong performance.

Scenario 2: Poor Optimization

If:
total_partition_count = 10,000
total_constant_partition_count = 500

RESULT: Only 5% of micro-partitions are optimized. **Most queries will scan unnecessarily large portions of the table**.

Action:

- Consider **automated reclustering** or **manually recluster** using RECLUSTER or a TASK.
- Might also revisit the **clustering key**—maybe activity_time isn't enough. Perhaps user_id, activity_time is better for LinkedIn's feed personalization.

Summary

Term	Meaning
total_constant_partition_count	Count of **optimized micro-partitions** that won't benefit from further clustering.
Why it matters	The higher this value, the more **partition pruning** Snowflake can do — leading to faster, cheaper queries.
How to use it	Monitor to decide **if/when to recluster**, or to evaluate the effectiveness of your clustering strategy.

HAPPY LEARNING
REGARDS
SARANSH JAIN