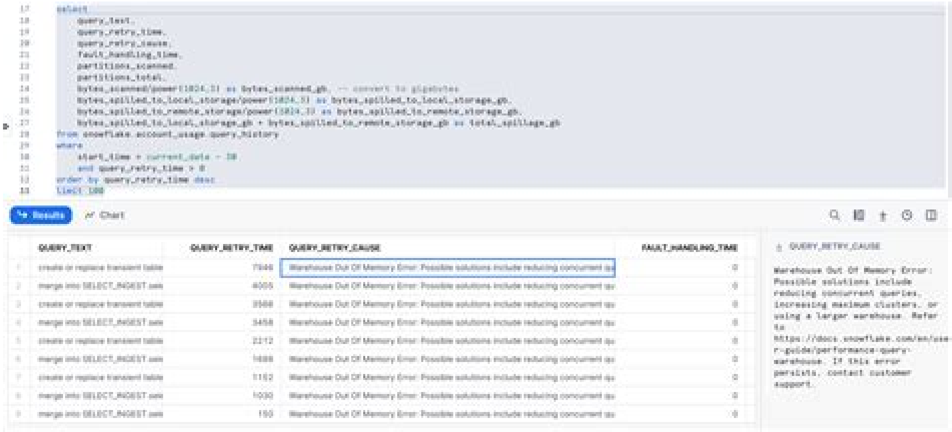
Snowflake Information Schemaquery History



```
select
    query_text,
    query_retry_time,
    query_retry_cause,
    fault_handling_time,
    partitions_scanned,
    partitions_total,
    bytes_scanned/power(1024,3) as bytes_scanned_gb, -- convert to gigabytes
    bytes_spilled_to_local_storage/power(1024,3) as bytes_spilled_to_local_storage_gb,
    bytes_spilled_to_remote_storage/power(1024,3) as bytes_spilled_to_remote_storage_gb,
    bytes_spilled_to_local_storage_gb + bytes_spilled_to_remote_storage_gb as total_spillage_gb
from snowflake.account_usage.query_history
where
    start_time > current_date - 30
    and query_retry_time > 0
order by query_retry_time desc
limit 100
```

	QUERY_TEXT	QUERY_RETRY_TIME	QUERY_RETRY_CAUSE	FAULT_HANDLING_TIME
1	create or replace transient table	7946	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
2	merge into SELECT_INGEST.sale	4005	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
3	create or replace transient table	3568	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
4	merge into SELECT_INGEST.sale	3458	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
5	create or replace transient table	2212	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
6	merge into SELECT_INGEST.sale	1688	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
7	create or replace transient table	1152	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
8	merge into SELECT_INGEST.sale	1030	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0
9	merge into SELECT_INGEST.sale	150	Warehouse Out Of Memory Error: Possible solutions include reducing concurrent qu	0

QUERY_RETRY_CAUSE

Warehouse Out Of Memory Error: Possible solutions include reducing concurrent queries, increasing maximum clusters, or using a larger warehouse. Refer to https://docs.snowflake.com/en/user-guide/performance-query-warehouse. If this error persists, contact customer support.

Snowflake information schema query history provides valuable insights for database administrators and data analysts looking to optimize their Snowflake environments. Snowflake, a cloud-based data warehousing solution, offers a robust information schema that allows users to query metadata and gain visibility into various aspects of their data operations. In this article, we will explore the components of the Snowflake information schema, delve into the query history, and provide best practices for analyzing and utilizing this data effectively.

Understanding Snowflake Information Schema

The Snowflake information schema is a set of predefined views that provide metadata about the objects in your Snowflake account. This schema serves as a vital tool for users who want to understand their database's structure, performance, and usage patterns.

Key Components of the Information Schema

The information schema contains several views that offer insights into different aspects of your Snowflake environment. Some of the key views include:

- **TABLES:** Provides information about all tables in the database, including their creation date, owner, and more.
- **COLUMNS:** Lists the columns in each table along with their data types, default values, and other attributes.
- **SCHEMATA:** Contains details about the schemas within the database, such

as schema names and owners.

- **ROLES:** Displays information about user roles and their permissions.
- **USERS:** Lists all users in the Snowflake account along with their login names and roles.

These components help administrators and analysts to manage their data more effectively and ensure that they can access the necessary information when needed.

Exploring Query History in Snowflake

One of the most useful features within the Snowflake information schema is the query history. This feature allows users to review the details of all the queries run in the Snowflake environment, providing insights into performance and usage patterns.

What is Query History?

Query history refers to a record of executed SQL statements within a Snowflake account, including information such as:

- Query ID
- User who executed the query
- Execution start and end times
- Duration of the query
- Number of rows processed
- Query status (success, failed, etc.)

This data is invaluable for performance tuning, troubleshooting issues, and understanding user behavior within the system.

Accessing Query History

Users can access the query history through the `QUERY_HISTORY` table in the Snowflake information schema. Here’s how you can query it:

```sql
SELECT
FROM TABLE(information_schema.query_history())
ORDER BY start_time DESC
LIMIT 100;
```

~~~
```
~~~

This SQL command retrieves the most recent 100 queries executed in your Snowflake account along with their metadata.

Analyzing Query History

Once you have access to the query history, you can analyze it in various ways to optimize your Snowflake environment. Here are some analysis techniques you can employ:

- **Identify Long-Running Queries:** Look for queries that take an excessive amount of time to execute. This can help you pinpoint performance bottlenecks.

- **Analyze Query Patterns:** Understand which queries are run most frequently and by whom. This can inform decisions about indexing or partitioning data.

- **Monitor Resource Usage:** Review the number of rows processed and the associated costs to optimize resource allocation.

- **Track Errors:** Investigate any failed queries to identify underlying issues, whether they are related to data quality or SQL syntax.

Best Practices for Using Query History

To make the most out of the query history in the Snowflake information schema, consider the following best practices:

1. Regularly Review Query Performance

Set a schedule to review query performance on a weekly or monthly basis. This habit will allow you to catch performance issues early and adjust accordingly.

2. Use Filters and Parameters

When querying the `QUERY_HISTORY` table, use filters such as specific users, warehouses, or time windows to narrow down the data to what’s most relevant for your analysis.

```sql
SELECT
FROM TABLE(information_schema.query_history())
WHERE user_name = 'your_user_name'
AND start_time >= '2023-01-01 00:00:00'
ORDER BY start_time DESC;
```

3. Document Insights and Changes

After analyzing your query history, document your findings and any changes made to optimize performance. This documentation can serve as a reference for future analysis and help with onboarding new team members.

4. Train Your Team

Educate your team on the importance of query performance and how to leverage query history for better practices. Conduct workshops or training sessions to share insights and promote a culture of performance optimization.

Conclusion

The **Snowflake information schema query history** is an essential tool for anyone managing a Snowflake environment. By leveraging the insights it provides, users can improve query performance, troubleshoot issues, and understand user behavior more effectively. Regular analysis of query history, combined with best practices for query optimization, will help ensure that your Snowflake environment remains efficient and cost-effective. Embrace the power of the information schema, and enhance your data management capabilities today.

Frequently Asked Questions

What is the Snowflake Information Schema query history?

The Snowflake Information Schema query history is a set of views that provide metadata about the queries executed in a Snowflake account, including details like execution time, user, and status.

How can I access the query history in Snowflake?

You can access the query history in Snowflake by querying the 'QUERY_HISTORY' view in the Information Schema, using SQL commands to filter results based on your requirements.

What kind of information can I retrieve from the query history in Snowflake?

From the query history, you can retrieve information such as query text, execution time, user who executed the query, database and schema used, and the query's current status.

Is there a limit to how far back I can view query history in Snowflake?

Yes, Snowflake retains query history for 14 days by default, but this can be extended by configuring the data retention policies for your account.

Can I filter the query history results in Snowflake?

Yes, you can filter query history results using various parameters like user name, execution time, and status by including WHERE clauses in your SQL query.

How does Snowflake ensure the security of query history data?

Snowflake provides security for query history data through role-based access controls, ensuring only authorized users can view sensitive information and logs related to query execution.

Find other PDF article:
https://soc.up.edu.ph/66-gist/pdf?dataid=lgP09-7712&title=what-is-premier-financial-alliance.pdf

Snowflake Information Schemaquery History

snowflake□□□□□□□□□□□□ - □□
Snowflake□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□MemSQL□□□□□SingleStore□□□□□□□□□□□□□□□□□ □□□share nothing ...

□□□□*snowflake*□□□□□□□□□□□□ - □□
□□□Snowflake□□□□□□□□□□□□1.5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□SaaS□□□□□□□□□□ 1.□16□ □Snowflake□□□□

Palantir Technologies □□□□□□□□□□□□ - □□
□□□□□Palantir□□□□□□□□□□□□□ □□□□Palantir□□□□□□□□□□□□□□□□□□□□Cloudera□□
□Snowflake□ □□3□□ …

□□□□□□□ Id □ snowflake □□□□ - □□
□□□□□□□□□□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□□□□□□□□
□ …

tor□□□□□□□□□□□□□□□□□□□ - □□
Snowflake □□ [5] □□ WebRTC □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□
□□□Snowflake □ …

□□□□□□□□□*ODS□DW□DWD□DWM□DWS …*
□□□□□□□□□ Hadoop□Delta Lake□□□□□□□ Databricks□Snowflake□□□□□□□ □□□□□□□□□□□□□
□□□□□□□ …

□□□□□□□□□□□□□□□□□ - □□
□□□□□ SPI□□□□JWT□□□□□ □□□□□□□□□□ (□□IP □□□□UUID □□□Snowflake □□□□□□□□)□ □□□□ □
□□/neural - □□ …

□□□Snowflake□Databricks□□□□lake house□TPC-DS …
Snowflake□IPO□□□□□DataBricks□□□TDC-DS□□□□□Snowflake□□□□□□□□□□□□□□□□□□□□□□
□□□SF□□□□□□□ …

□□□□□□□□□□□□□□□□□□□□□ *…*
□□□□□□□□□□□□□□□□□□Żaden płatek śniegu nie czuje się odpowiedzialny za lawinę.□□□□□□No
Snowflake in an avalanche ever feels …

snowflake□□□□□□□□□□□□□□□□□□□ …
□□ Snowflake □□□□□ 6 □ 27 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□
□□□ …

snowflake□□□□□□□□□ - □□
Snowflake□□□□□□□□□□□□□□□□□□□□□□□□□ □□MemSQL□□□□SingleStore□□□□□□□□□□□□□
□□□share nothing …

□□□snowflake□□□□□□□□□ - □□
□□□Snowflake□□□□□□□□□1.5□□□□□□□□□□□□□□□□□□□□□□□□□□SaaS□□□□□□□ 1.□16□
□Snowflake□□□□

Palantir Technologies □□□□□□□□□□□□ - □□
□□□□□Palantir□□□□□□□□□□□□□ □□□□Palantir□□□□□□□□□□□□□□□□□□□□Cloudera□□
□Snowflake□ □□3□□ …

□□□□□□□ Id □ snowflake □□□□ - □□
□□□□□□□□□□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□□□□□□□□
□ …

tor□□□□□□□□□□□□□□□□□□□ - □□
Snowflake □□ [5] □□ WebRTC □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□
□□□Snowflake □ …

□□□□□□□□□□□□ODS□DW□DWD□DWM□DWS …
□□□□□□□□□□□□ Hadoop□Delta Lake□□□□□□□□□ Databricks□Snowflake□□□□□□□□□ □□□□□□□□□□□□□□□□□□ □□□□□□□□□ …

□□□□□□□□□□□□□□□□□□□□□□□ - □□
□□□□□□ SPI□□□□□JWT□□□□□□ □□□□□□□□□□□□□ (□□IP □□□□□UUID □□□□Snowflake □□□□□□□□□□□)□ □□□□□ □ □□/neural - □□ …

□□□□Snowflake□Databricks□□□□□□lake house□TPC-DS …
Snowflake□IPO□□□□□□□DataBricks□□□□TDC-DS□□□□□□□Snowflake□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□SF□□□□□□□□□□ …

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ …
□□□□□□□□□□□□□□□□□□□□□□□□Żaden płatek śniegu nie czuje się odpowiedzialny za lawinę.□□□□□□□□No Snowflake in an avalanche ever feels …

snowflake□□□□□□□□□□□□□□□□□□□□□□□□□ …
□□ Snowflake □□□□□□□ 6 □ 27 □□ □□□□□□□□□□□ □□□ …

Explore the Snowflake information schema query history to optimize your database performance. Discover how to leverage query insights effectively. Learn more!

Back to Home