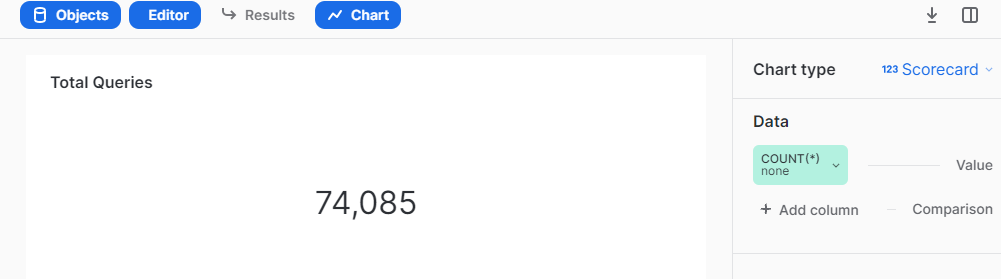
## **QUERY PERFORMANCE DASHBOARD**

# **Snowsight Dashboard Setup**

### **1 TOTAL QUERIES**

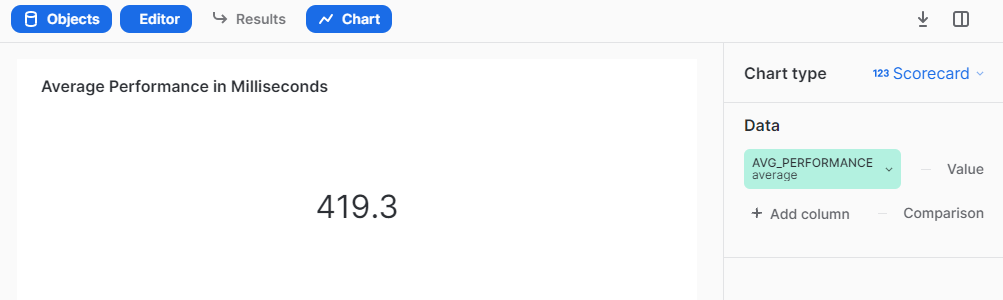
select count(\*) from snowflake.account\_usage.query\_history;



### 

### **2 AVERAGE QUERY RUNTIME IN MILLISECONDS**

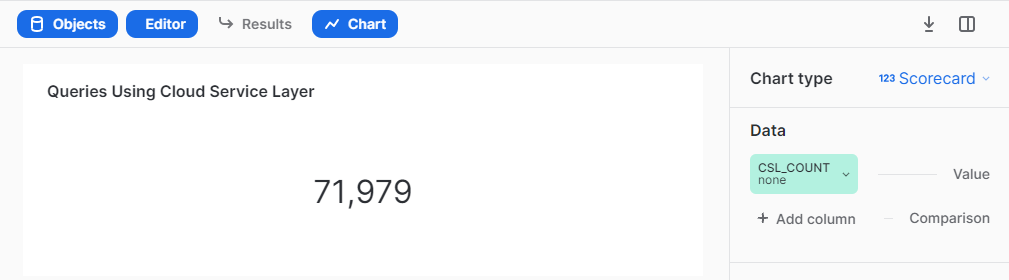
SELECT ROUND(AVG(TOTAL\_ELAPSED\_TIME),2) AS AVG\_PERFORMANCE FROM snowflake.account\_usage.query\_history ;



### 

### **3 QUERIES USING CLOUD SERVICE LAYER**

select count(warehouse\_size) as csl\_count from ( select IFNULL(WAREHOUSE\_SIZE,'NO WAREHOUSE') AS WAREHOUSE\_SIZE,warehouse\_name,query\_id,query\_text from snowflake.account\_usage.query\_history where warehouse\_size IS NULL);



### 

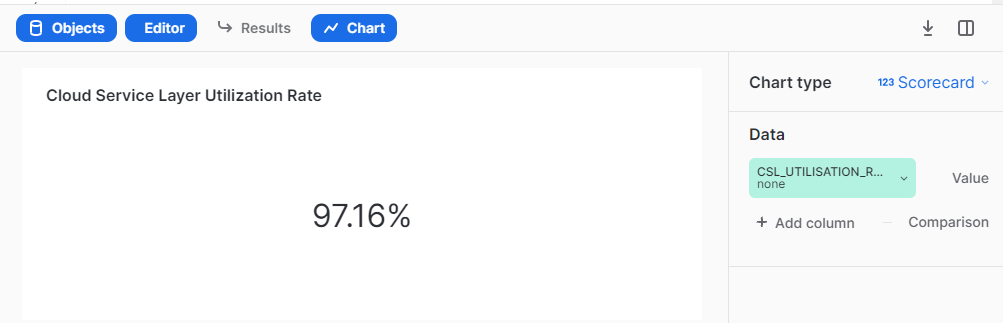
### **4 CLOUD SERVICE LAYER UTILIZATION RATE**

select CONCAT(ROUND(CSL\_UTILISATION,2), '%') AS CSL\_UTILISATION\_RATE FROM

(SELECT

(select count(warehouse\_size) as csl\_count from ( select IFNULL(WAREHOUSE\_SIZE,'NO WAREHOUSE') AS WAREHOUSE\_SIZE,warehouse\_name,query\_id,query\_text from snowflake.account\_usage.query\_history where warehouse\_size IS NULL)) /

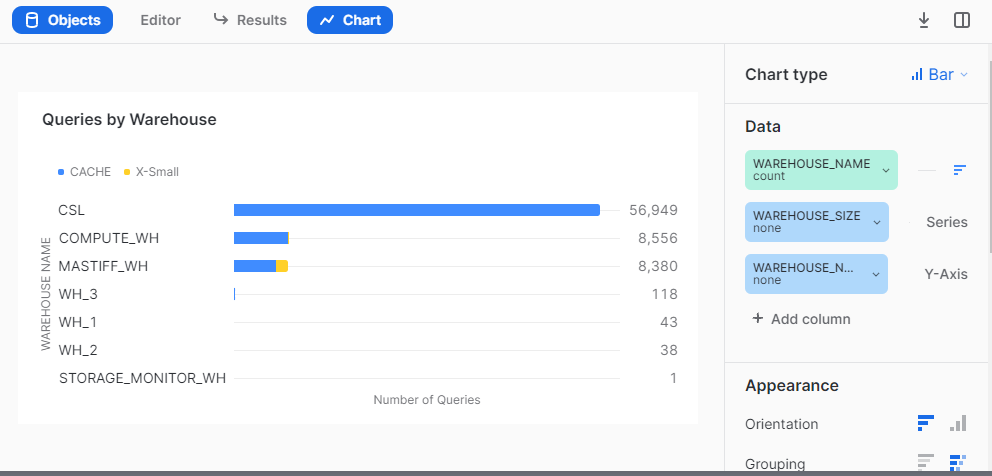
(SELECT count (IFNULL(warehouse\_size,'NO WAREHOUSE')) from snowflake.account\_usage.query\_history) \* 100 AS CSL\_UTILISATION );

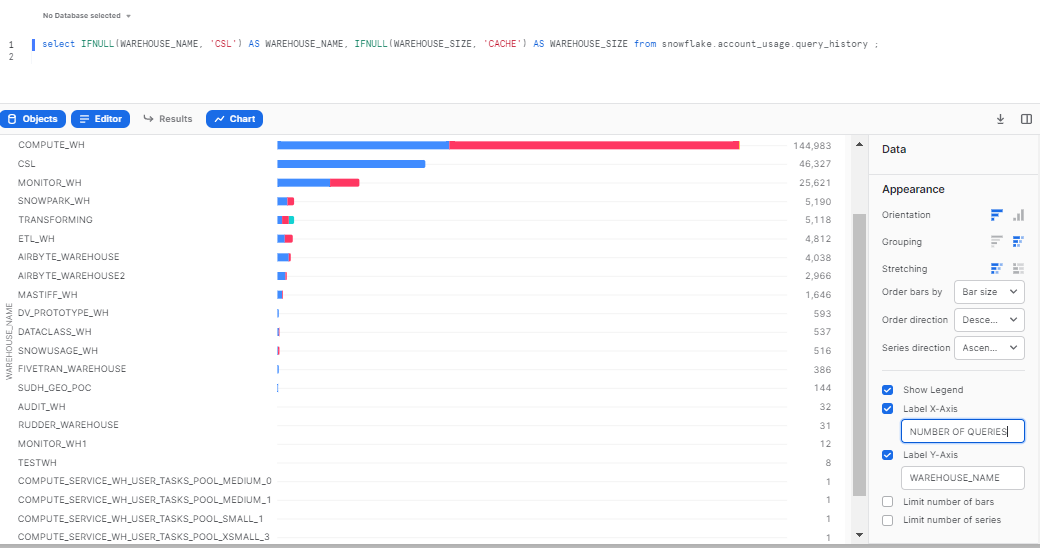


### 

### **5 QUERIES BY WAREHOUSE**

select IFNULL(WAREHOUSE\_NAME, 'CSL') AS WAREHOUSE\_NAME, IFNULL(WAREHOUSE\_SIZE, 'CACHE') AS WAREHOUSE\_SIZE from snowflake.account\_usage.query\_history;



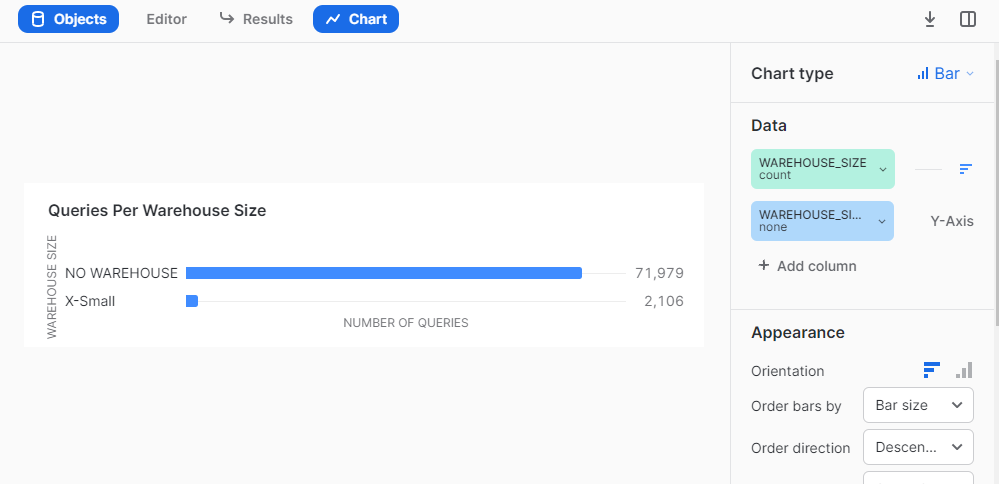


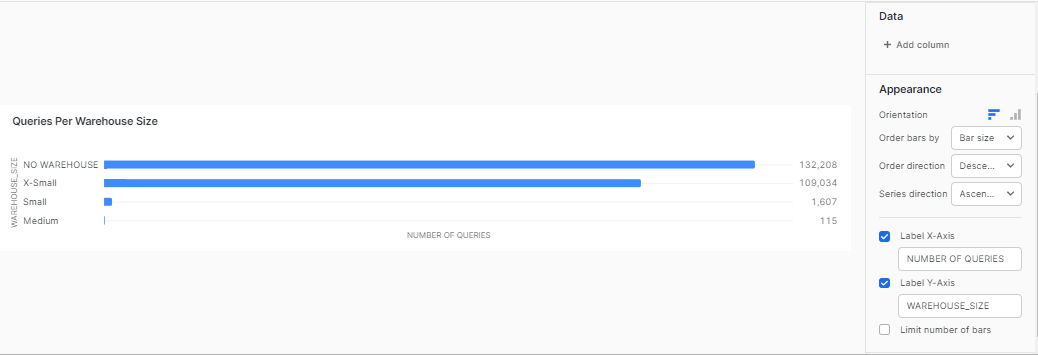
### 

### **6 QUERIES PER WAREHOUSE SIZE**

select IFNULL(WAREHOUSE\_SIZE,'NO WAREHOUSE') AS WAREHOUSE\_SIZE,\* from

Snowflake.account\_usage.query\_history;

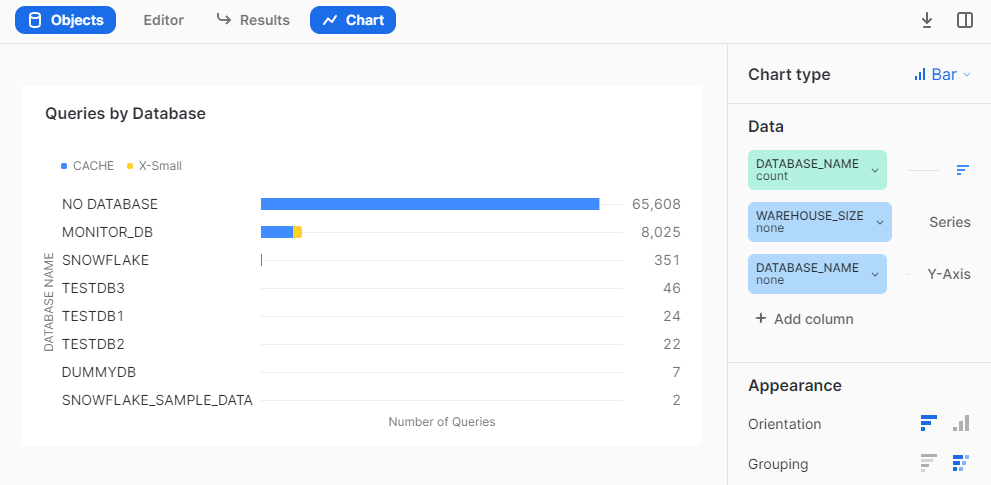


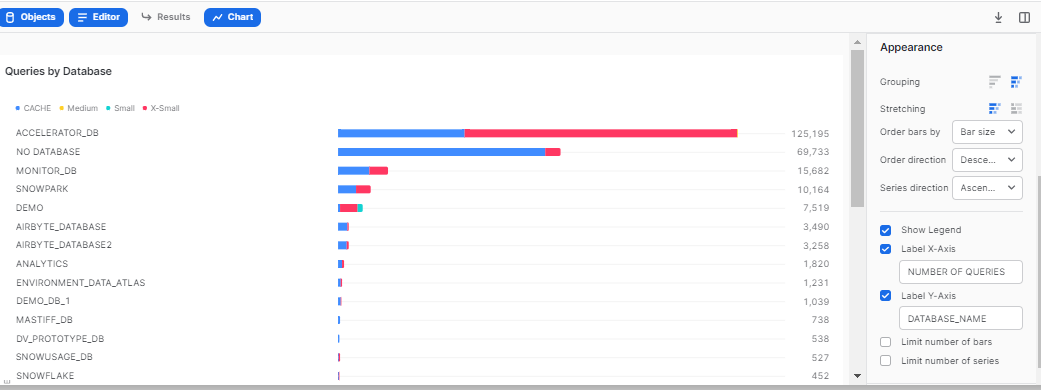


### 

### **7 QUERIES BY DATABASE**

select IFNULL(database\_name,'NO DATABASE') AS DATABASE\_NAME, IFNULL(WAREHOUSE\_SIZE, 'CACHE') AS WAREHOUSE\_SIZE, WAREHOUSE\_NAME, ROLE\_NAME from snowflake.account\_usage.query\_history ;

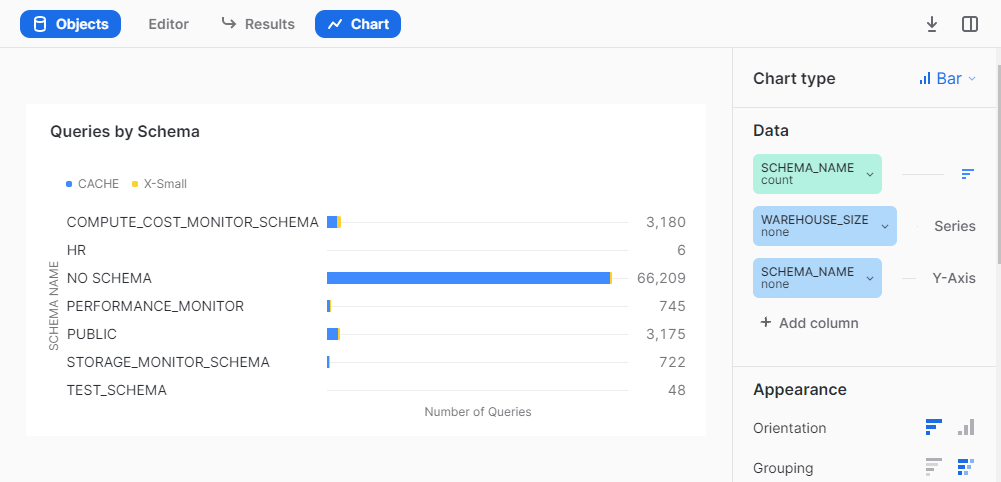


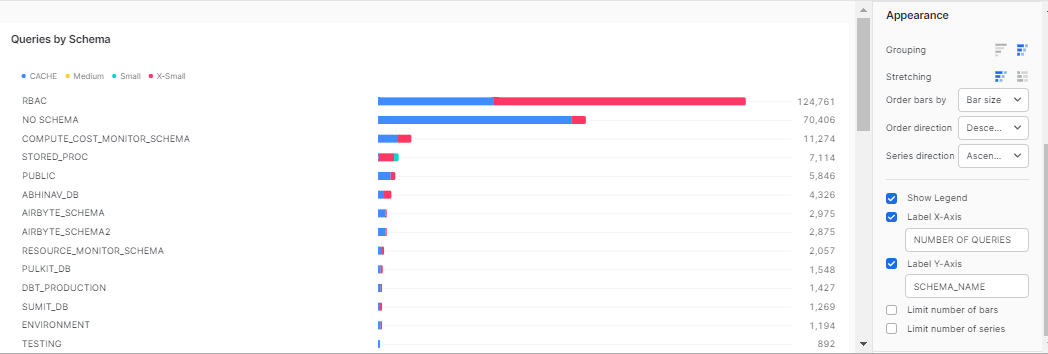


### 

### **8 QUERIES BY SCHEMA**

select IFNULL(SCHEMA\_NAME,'NO SCHEMA') AS SCHEMA\_NAME, IFNULL(WAREHOUSE\_SIZE, 'CACHE') AS WAREHOUSE\_SIZE, WAREHOUSE\_NAME, ROLE\_NAME from snowflake.account\_usage.query\_history ;

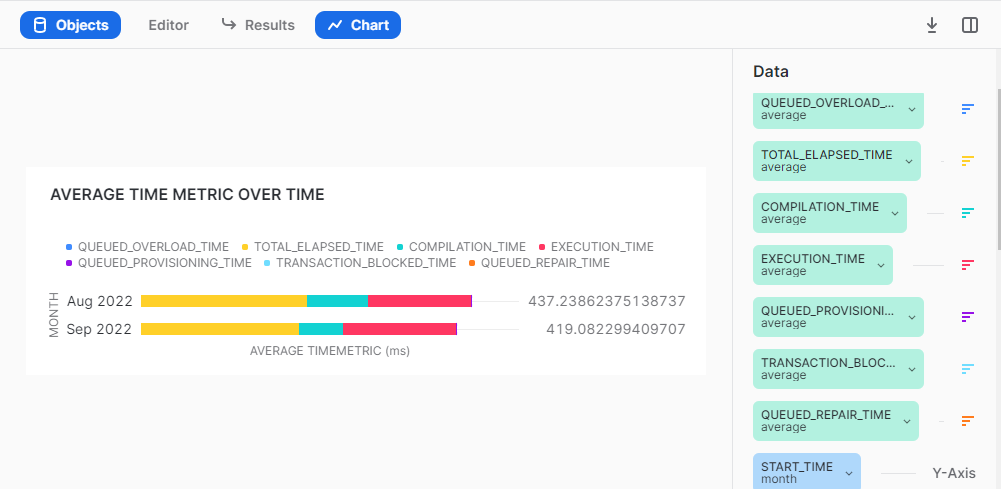


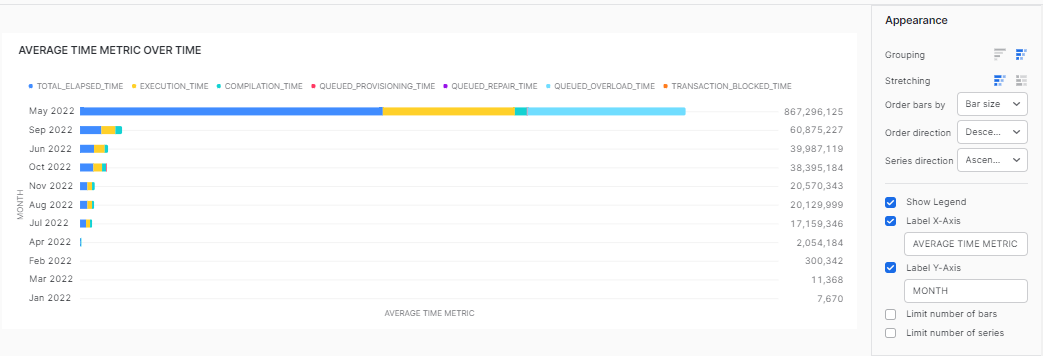


### 

### **9 AVERAGE TIME METRIC OVER TIME**

select \* from snowflake.account\_usage.query\_history;

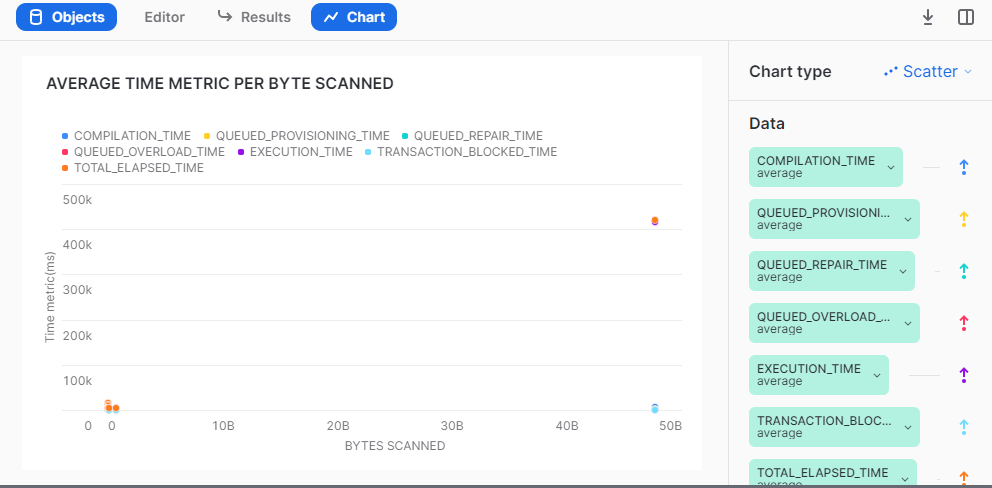




### 

### **10 AVERAGE TIME METRIC PER BYTE SCANNED**

select top 9000 compilation\_time,execution\_time,queued\_provisioning\_time,queued\_repair\_time,queued\_overload\_time,total\_elapsed\_time,transaction\_blocked\_time,bytes\_scanned from snowflake.account\_usage.query\_history ;

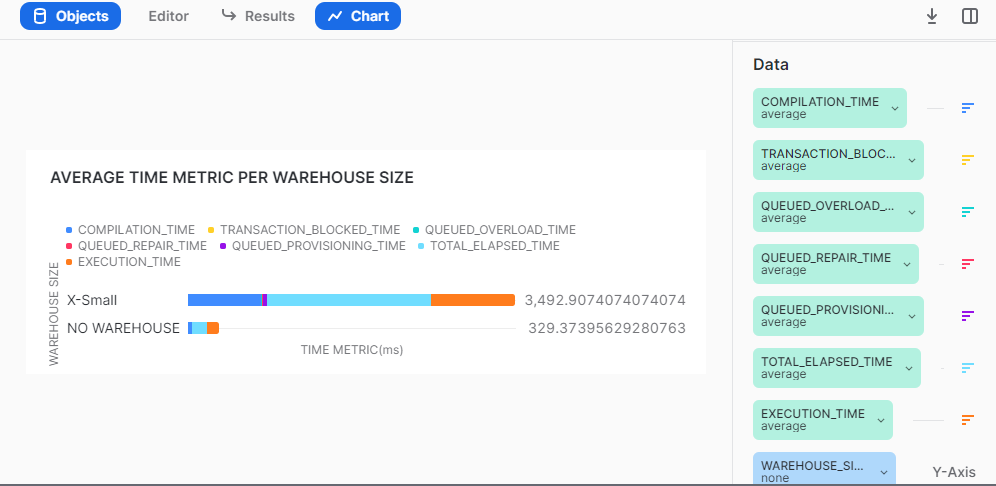


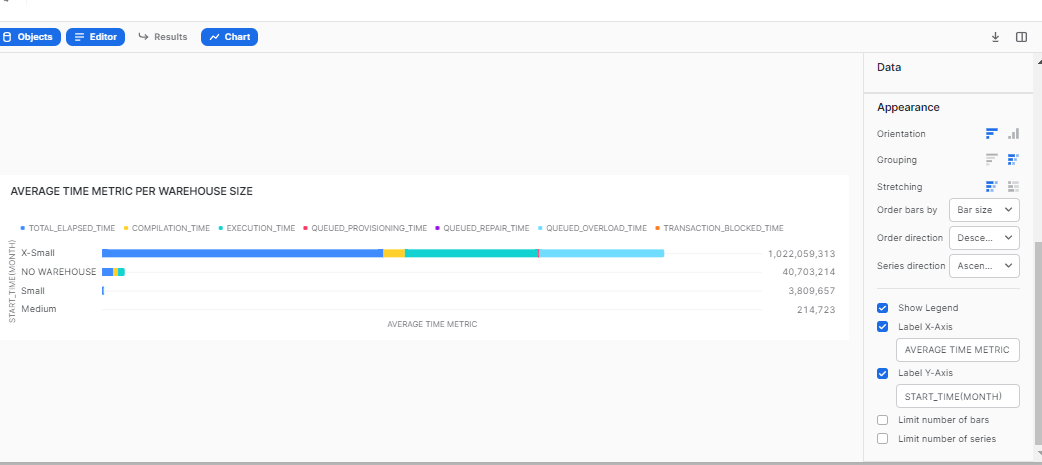
### 

### **11 AVERAGE TIME METRIC PER WAREHOUSE SIZE**

select IFNULL(WAREHOUSE\_SIZE,'NO WAREHOUSE') AS WAREHOUSE\_SIZE,\* from

snowflake.account\_usage.query\_history;



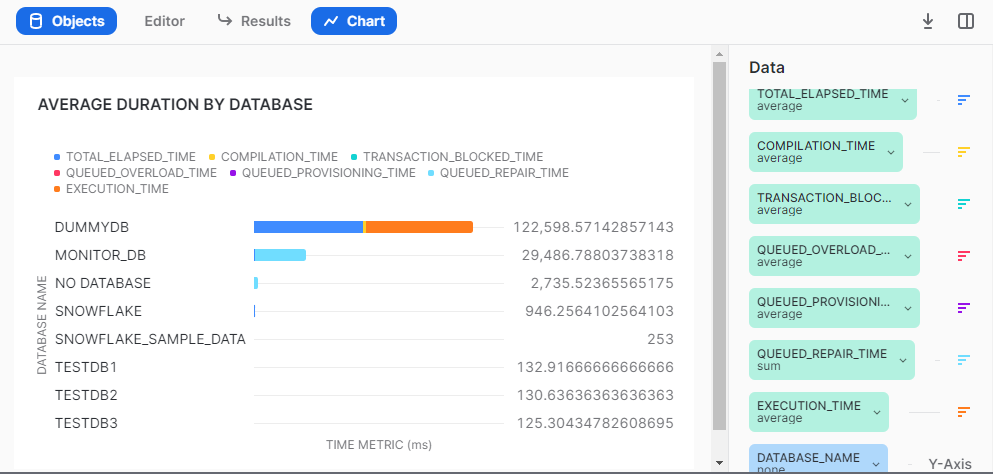


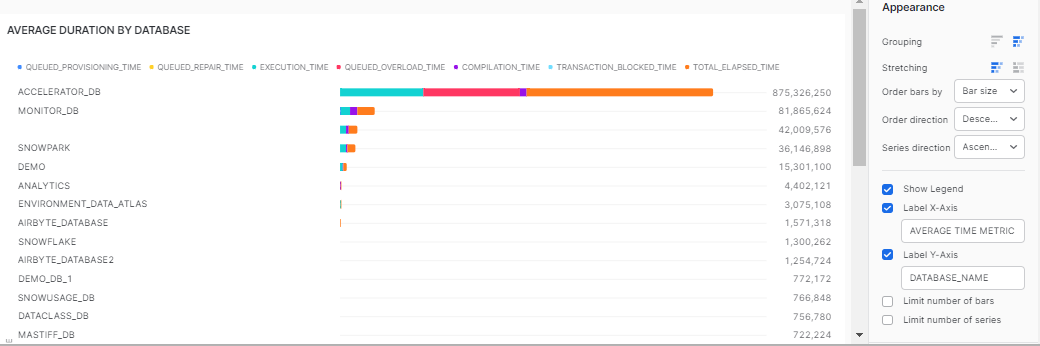
### 

### **12 AVERAGE DURATION BY DATABASE**

select IFNULL(WAREHOUSE\_SIZE,'NO WAREHOUSE') AS WAREHOUSE\_SIZE, IFNULL(DATABASE\_NAME,'NO DATABASE') AS DATABASE\_NAME,

\* from snowflake.account\_usage.query\_history;





## **Best Practices -**

* If you have records more than 10000 then select only top 9999 ( refer. link:- <https://docs.snowflake.com/en/sql-reference/constructs/top_n.html>) to create a chart as Snowsight only supports data to be visualized till 9999 rows.
* It's preferable to do aggregation in the query itself rather than to do it in Snowsight UI.
* Adjust the round off value as per your requirement ( e.g - round((<column\_name>), n) where n = natural number ).
* On hovering over any filter, visuals getting filtered would be highlighted.
* To make filter interaction more user friendly, you can even use color coding.
* If snowsight dashboard is running slow, then to optimize performance:
  + It is recommended to extract only three months (or less than three months) data from the query history using a date range filter in all the KPIs.

For example - select count(\*) from snowflake.account\_usage.query\_history where start\_time = :daterange;

* + Utilize a custom table to store the latest three months (or less) of query history data, allowing you to leverage it for the query performance dashboard.

## 