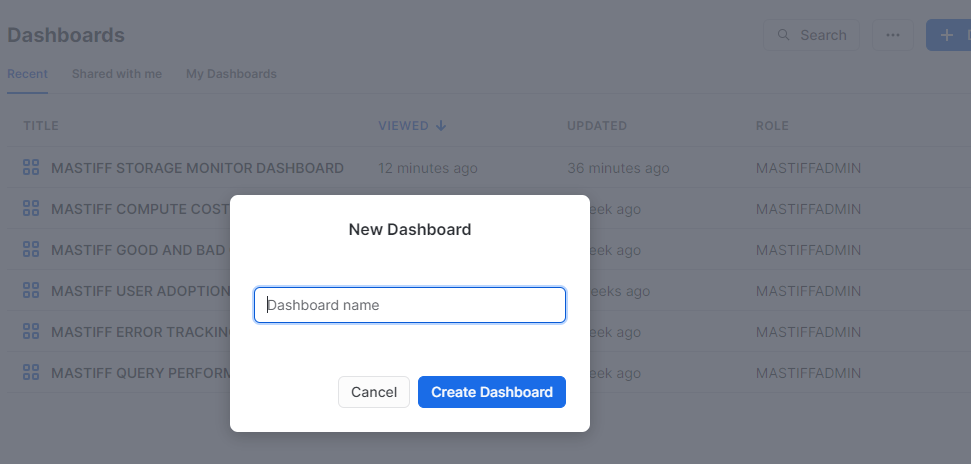
## **ERROR TRACKING DASHBOARD**

# **Snowsight Dashboard Setup**

## 1. Create a Dashboard

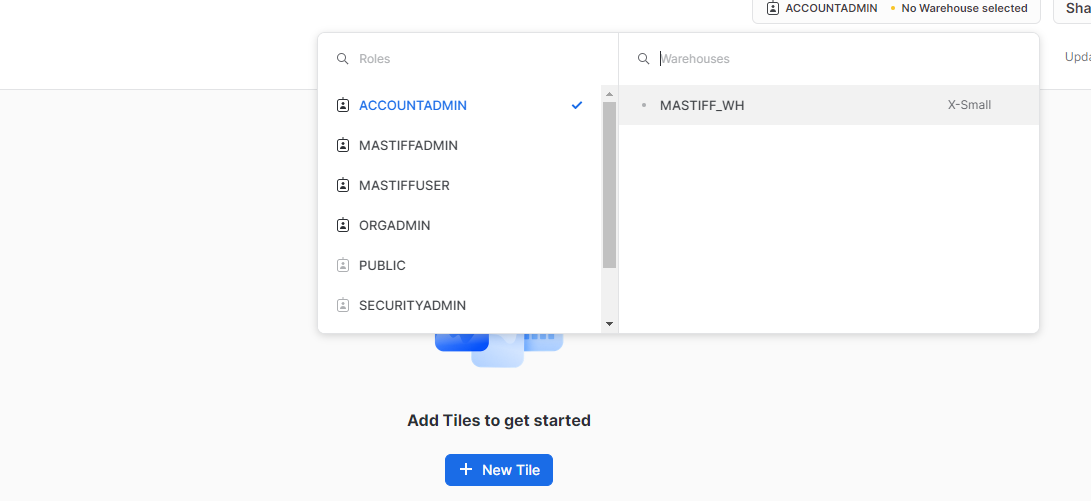
Login to the account and navigate to the **tab dashboard** in the left side options ,then click on the **+dashboard** option highlighted in blue to create a new dashboard.

## 2. Name the dashboard

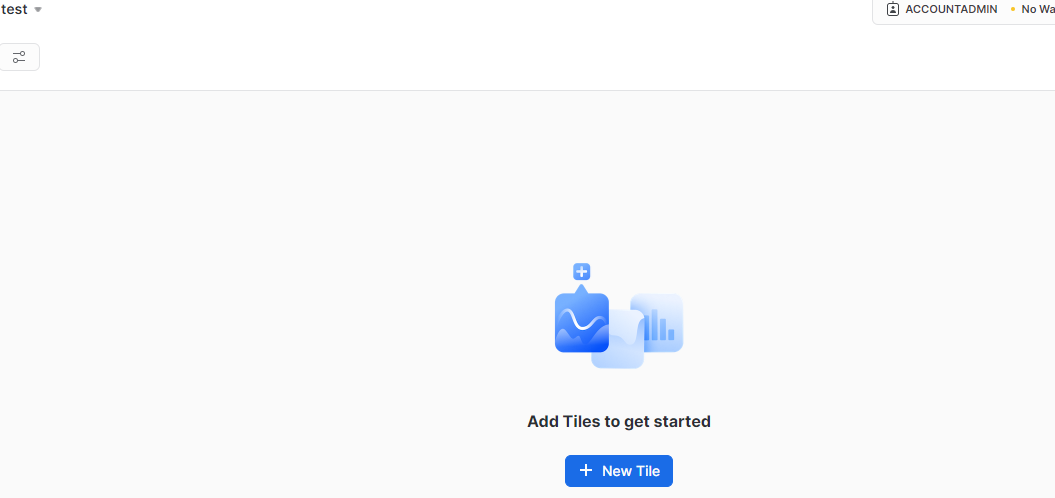


Next step is to give an appropriate name to the dashboard and click **Create Dashboard**.

## 3. Setup the Role and Warehouse

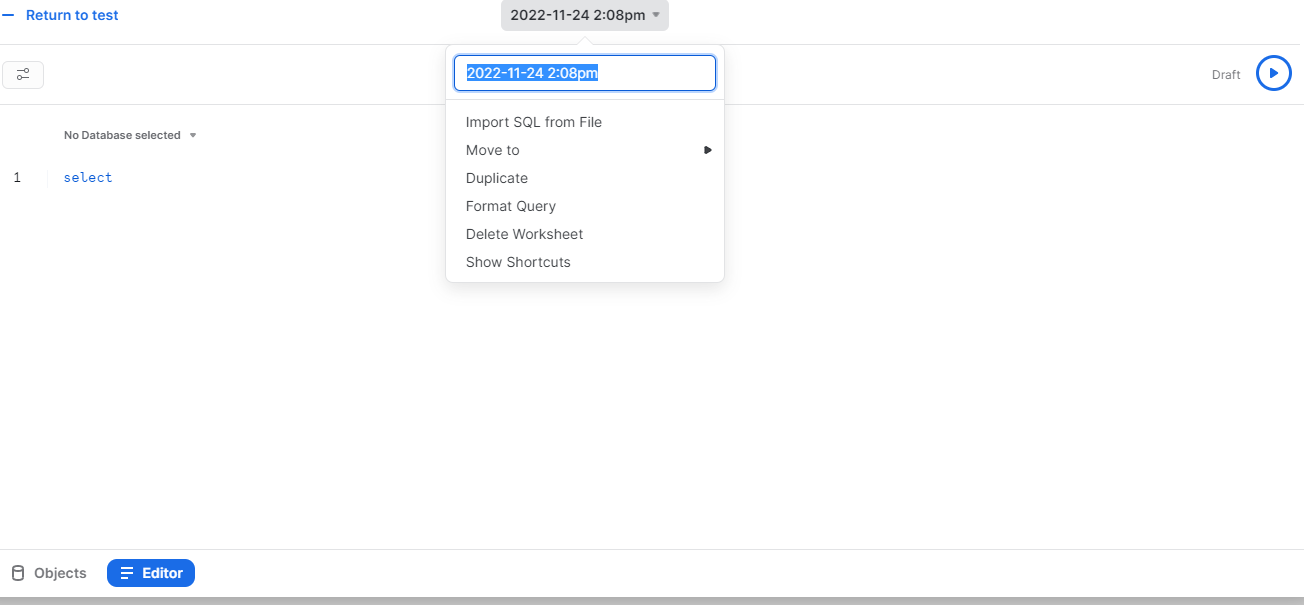


## 4. Creating KPIs and Charts



To create KPIs and charts, start by adding a new tile by clicking **+New Tile .**

## 5. Adding names to the KPIs and Executing the queries.



We can name the KPI by clicking the date followed by the time and changing it to the required name. To execute a query, type in the editor and click the run button to get the desired results.

**Note :** Before you create all the dashboards in Snowsight, Use the role **MONITOR\_ADMIN** and warehouse **MONITOR\_WH** to create dashboards in SNOWSIGHT for Watchkeeper.

## 

## **ERROR TRACKING DASHBOARD SETUP**

## **Filters**

**Customize the filter query's refresh frequency to align with your requirements. If you regularly analyze dashboards, consider refreshing it daily for up-to-date data. However, if dashboard analysis is infrequent, opt for "never refresh" to reduce unnecessary costs. Keep in mind that choosing "never refresh" means running filter query manually when utilizing the dashboard, which can be done from the filter section.**

Create the filters which will be used in this dashboard.

### **Role Filter**

**Display Name** :- ROLE

**SQL Keyword** :- ROLENAME

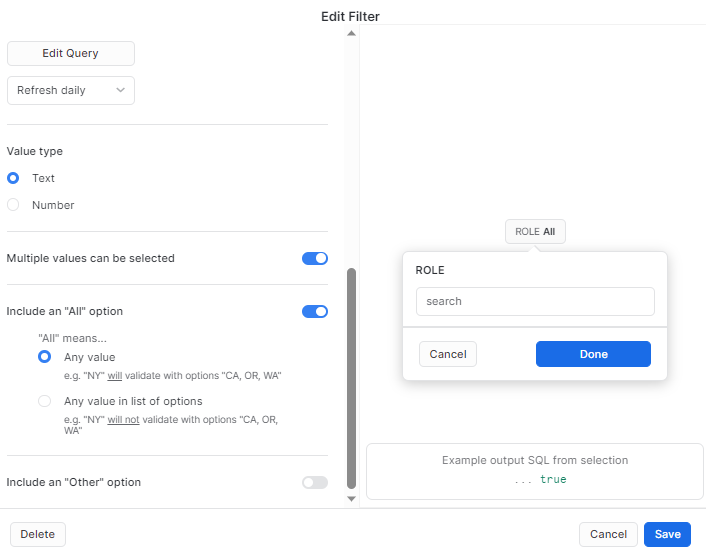
**Role** :- MONITOR\_ADMIN

**Warehouse** :- MONITOR\_WH

**Options Via** :- Query

**Write Query** :- SELECT DISTINCT(ROLE\_NAME) FROM SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY;

Go for the below selections :



### **Query Type Filter**

**Display Name** :- QUERY TYPE

**SQL Keyword** :- querytype

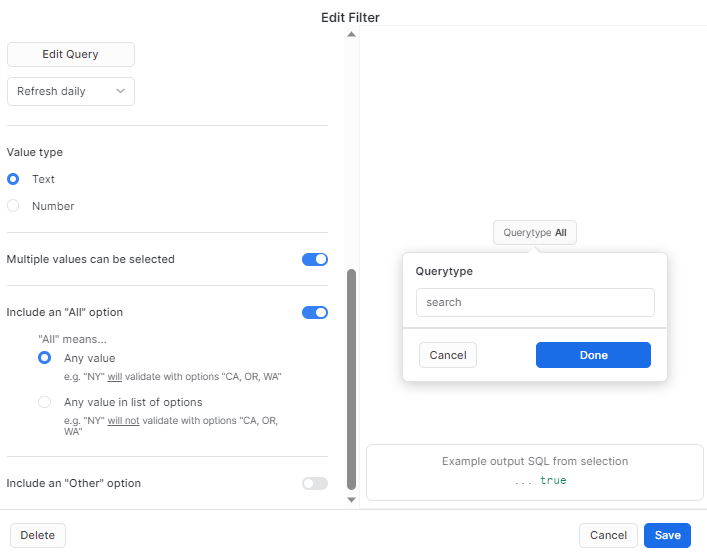
**Role** :- MONITOR\_ADMIN

**Warehouse** :- MONITOR\_WH

**Options Via** :- Query

**Write Query** :- select distinct(query\_type) from SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY;

Go for the below selections :



### **User Filter**

**Display Name** :- USER

**SQL Keyword** :- usernamefilter

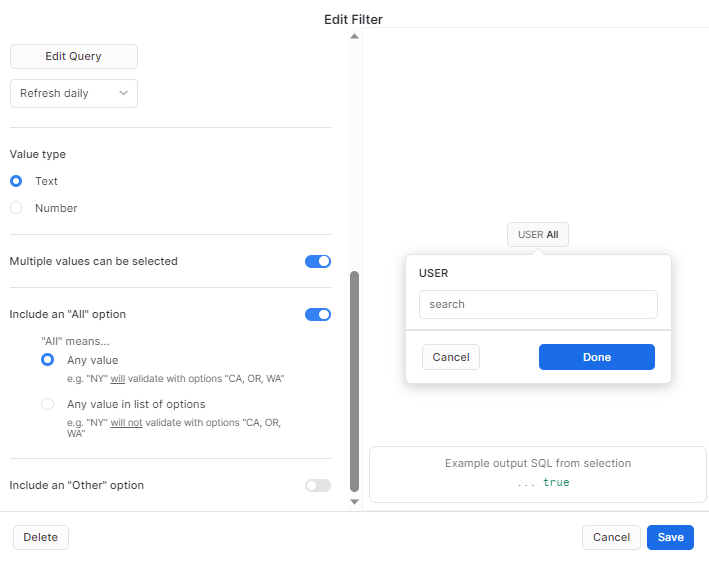
**Role** :- MONITOR\_ADMIN

**Warehouse** :- MONITOR\_WH

**Options Via** :- Query

**Write Query** :- select distinct(user\_name) from SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY;

Go for the below selections :



### **Query Tag Filter**

**Display Name** :- Querytag

**SQL Keyword** :- querytag

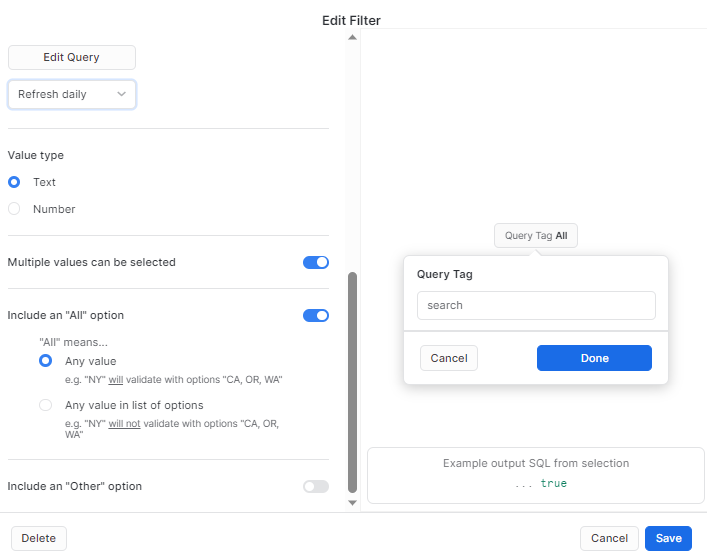
**Role** :- MONITOR\_ADMIN

**Warehouse** :- MONITOR\_WH

**Options Via** :- Query

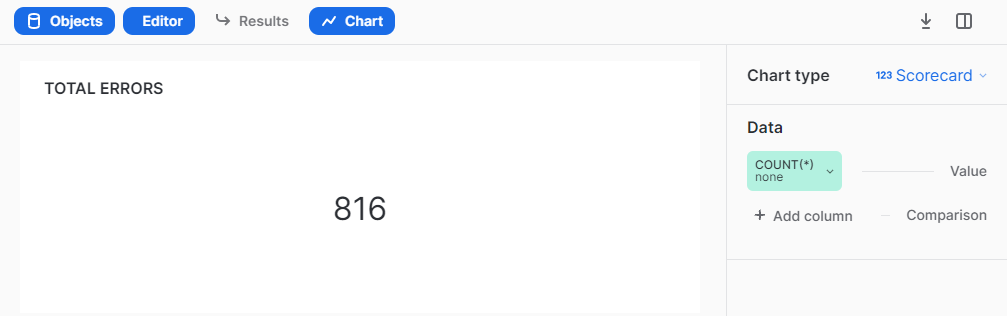
**Write Query** :- select distinct(query\_tag) from SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY;

Go for the below selections :



### **1 TOTAL ERRORS**

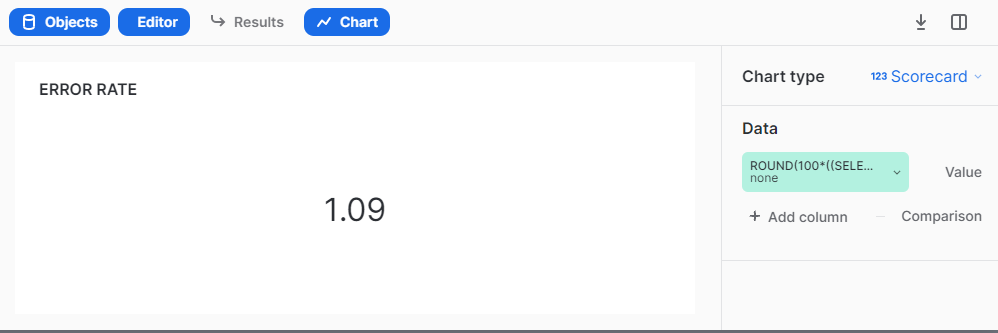
select count(\*) from snowflake.account\_usage.query\_history where error\_code is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag;



### 

### **2 ERROR RATE**

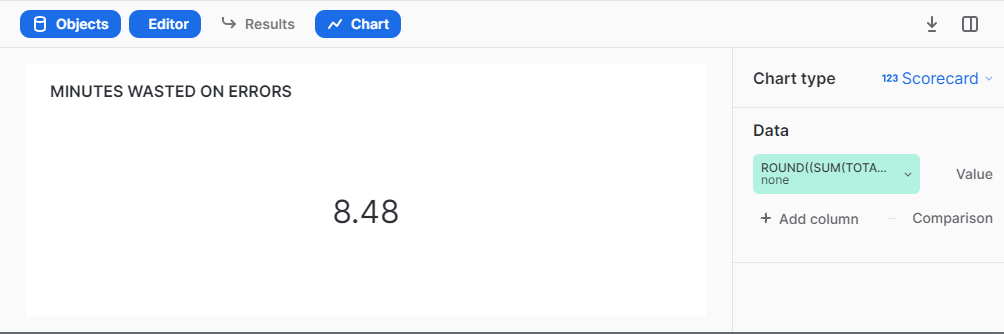
select round(100\*((select count(\*) from snowflake.account\_usage.query\_history where error\_code is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag)/(select count(\*) from snowflake.account\_usage.query\_history where role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag)),2);



### 

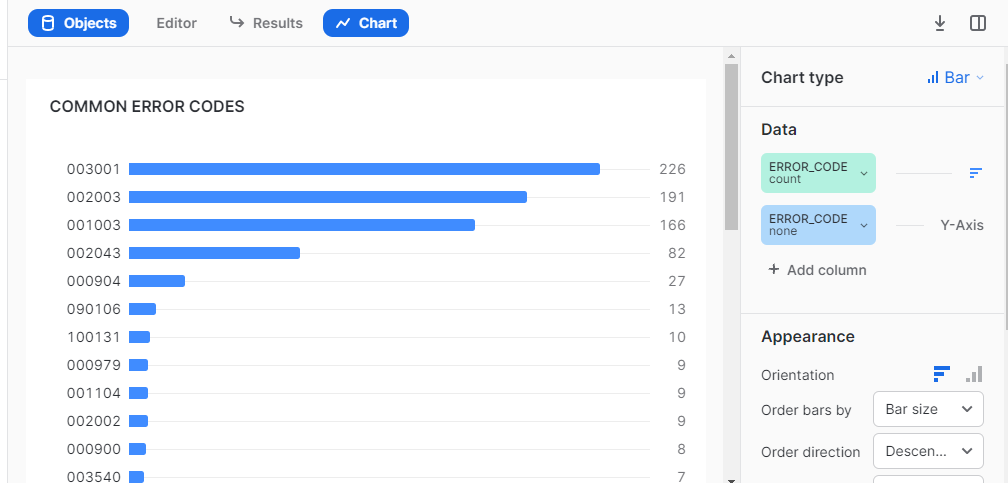
### **3 MINUTES WASTED ON ERRORS**

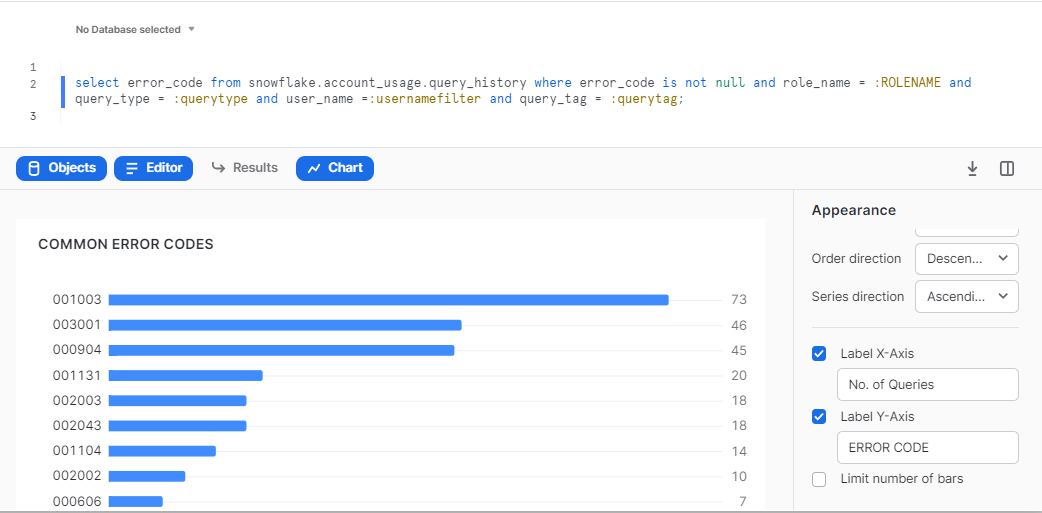
select round((sum(total\_elapsed\_time))/60000,2) from snowflake.account\_usage.query\_history where error\_code is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag;



### **4 COMMON ERROR CODES**

select error\_code from snowflake.account\_usage.query\_history where error\_code is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag;

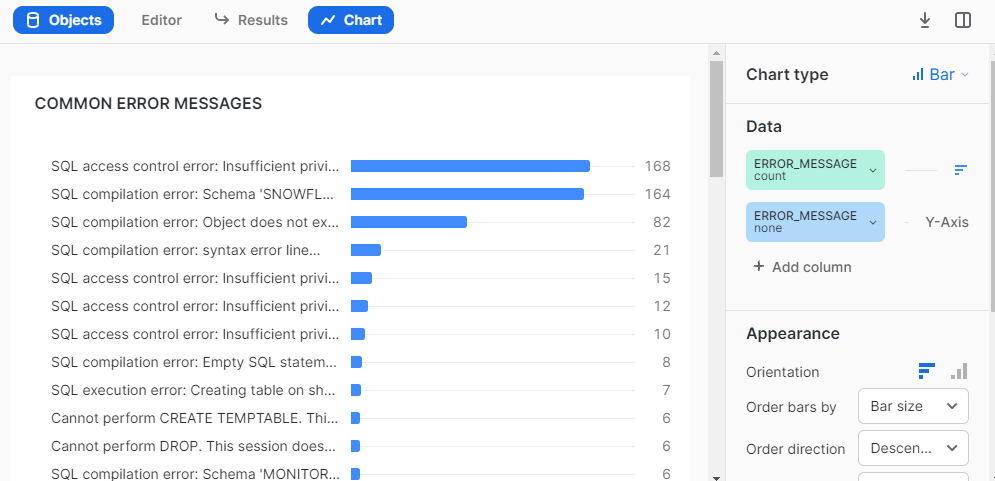


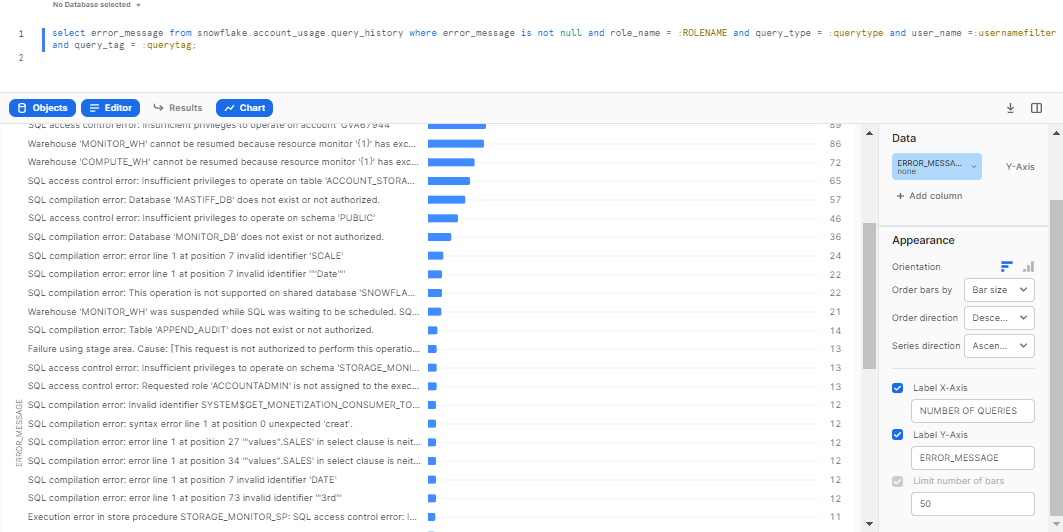


### 

### **5 COMMON ERROR MESSAGES**

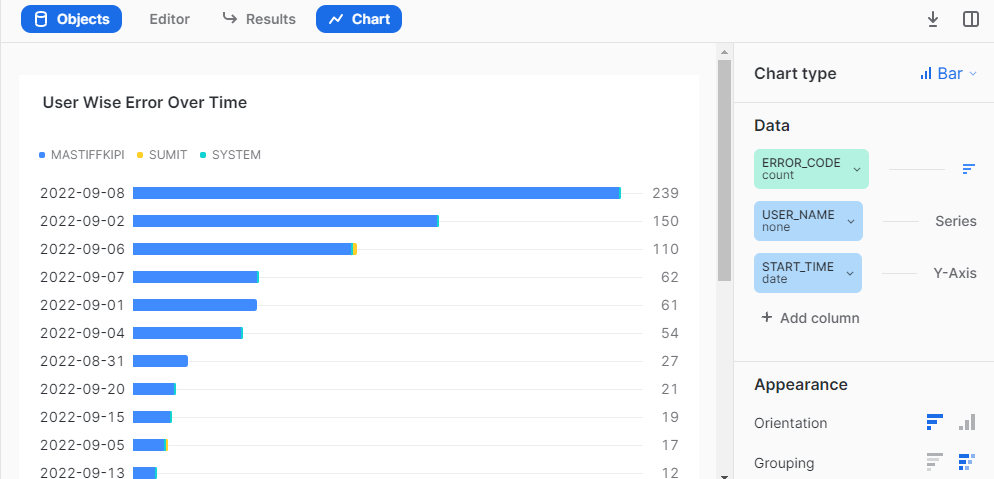
select error\_message from snowflake.account\_usage.query\_history where error\_message is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag;

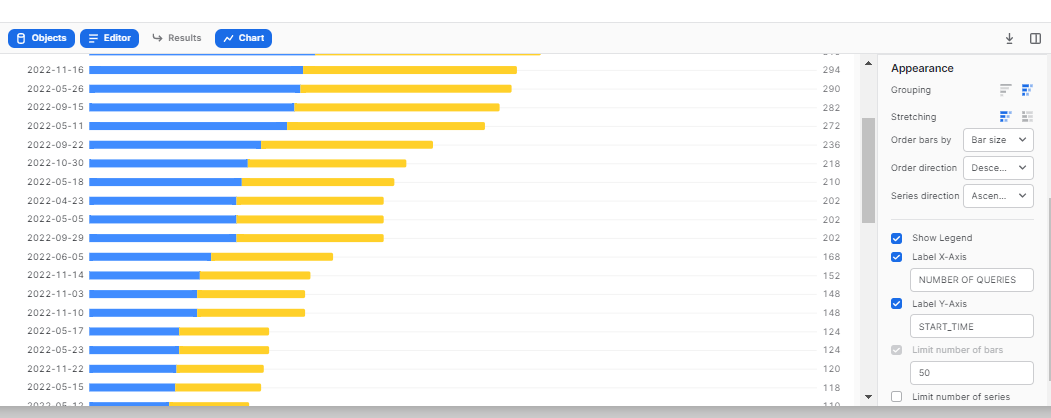




### **6 User Wise Error Over Time**

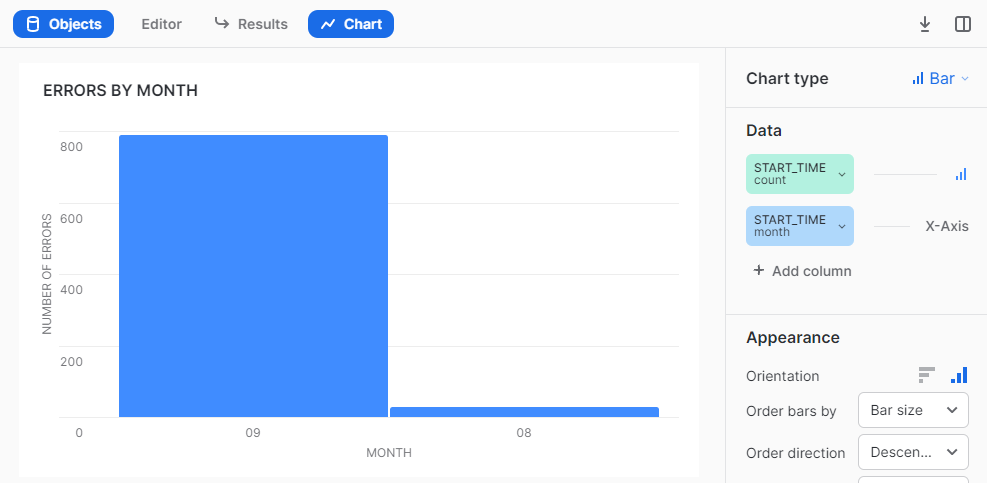
select \* from snowflake.account\_usage.query\_history where error\_code is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag;

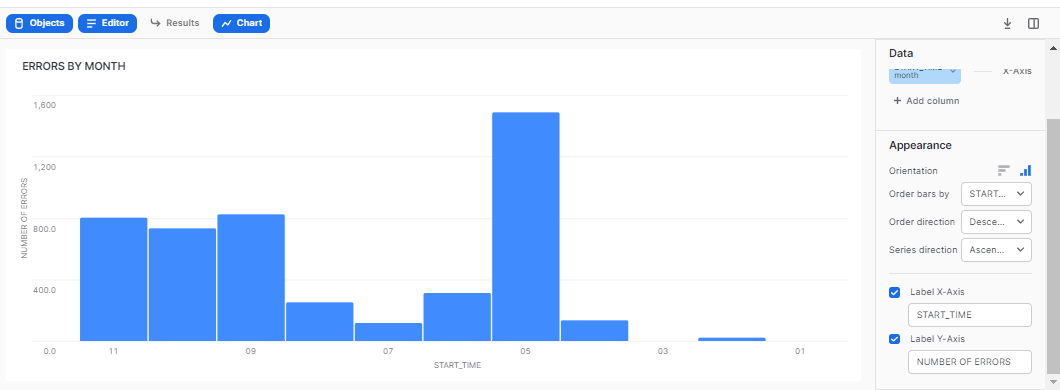




### **7 ERRORS BY MONTH**

select QUERY\_ID, error\_code, start\_time from snowflake.account\_usage.query\_history where error\_code is not null and role\_name = :ROLENAME and query\_type = :querytype and user\_name =:usernamefilter and query\_tag = :querytag order by start\_time;





## **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.
* Customize the refreshment time of the filter query according to your specific needs.
* 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 error tracking dashboard.