## **WATCHKEEPER SECURITY MONITORING DASHBOARD**

### **1 TOTAL ACCOUNTADMIN USERS**

Select

Count(Distinct(GRANTEE\_NAME)) as Total\_Users

from

snowflake.account\_usage.grants\_to\_users

where

role = 'ACCOUNTADMIN'

and deleted\_on is null

and GRANTEE\_NAME in (

Select

Distinct(Name)

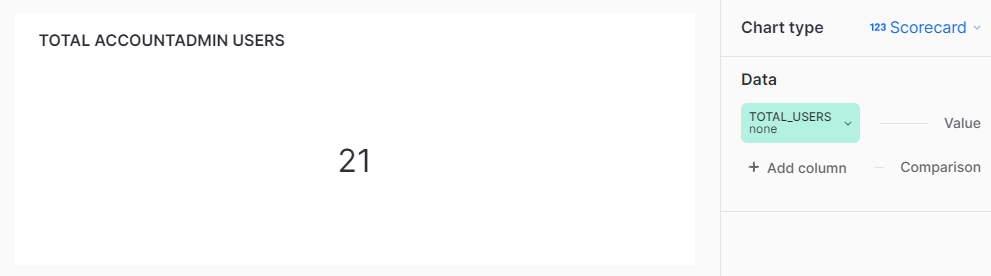
from

snowflake.account\_usage.users

where

deleted\_on is null

);



### 

### **2 USER TO ROLE RATIO (LARGER IS BETTER)**

select

round(

(

select

count(Distinct(Name))

from

"SNOWFLAKE"."ACCOUNT\_USAGE".USERS

where

deleted\_on is null

)

/(

select

count(distinct(name))

from

"SNOWFLAKE"."ACCOUNT\_USAGE".roles

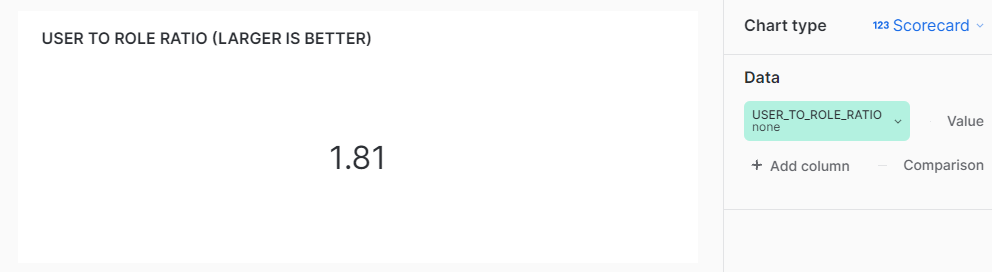
where

deleted\_on is null

)

,

2) as user\_to\_role\_ratio;



### 

### **3 TOTAL DISCONNECTED ROLES**

select

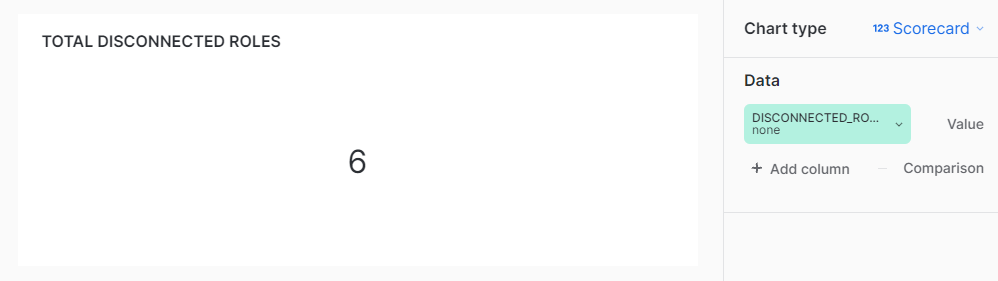
count(\*) as Disconnected\_role\_count

from

MONITOR\_DB.SECURITY\_MONITOR\_SCHEMA.ROLES\_INFO\_TB

where

ROLE\_TYPE = 'Disconnected Role';



### **4 LOGIN ERROR %**

Select

round(

(

(

Select

count(\*)

from

snowflake.account\_usage.login\_history

where

is\_success = 'NO'

) /(

Select

count(\*)

from

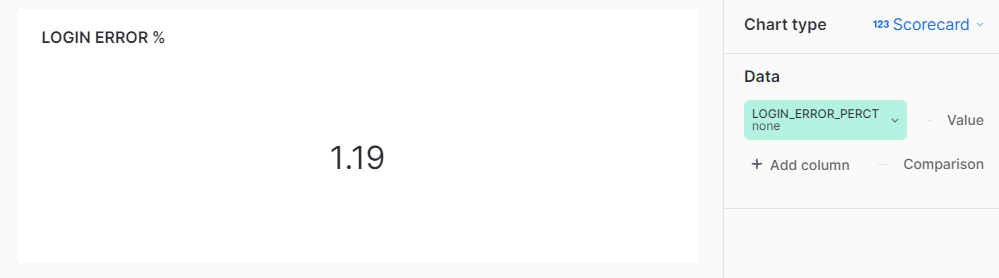
snowflake.account\_usage.login\_history

) \* 100

),

2

) as Login\_Error\_Perct;



### **5 USERS HAVING MOST PRIVILEGES**

select

privilege,

gtu.grantee\_name as USER\_NAME,

Count(DISTINCT(gtr.name)) as Privilege\_count

from

SNOWFLAKE.ACCOUNT\_USAGE.GRANTS\_TO\_ROLES as gtr,

SNOWFLAKE.ACCOUNT\_USAGE.GRANTS\_TO\_USERS as gtu

where

gtr.grantee\_name = gtu.role

and gtu.deleted\_on is null

and gtr.deleted\_on is null

and USER\_NAME in (

Select

Distinct(Name)

from

snowflake.account\_usage.users

where

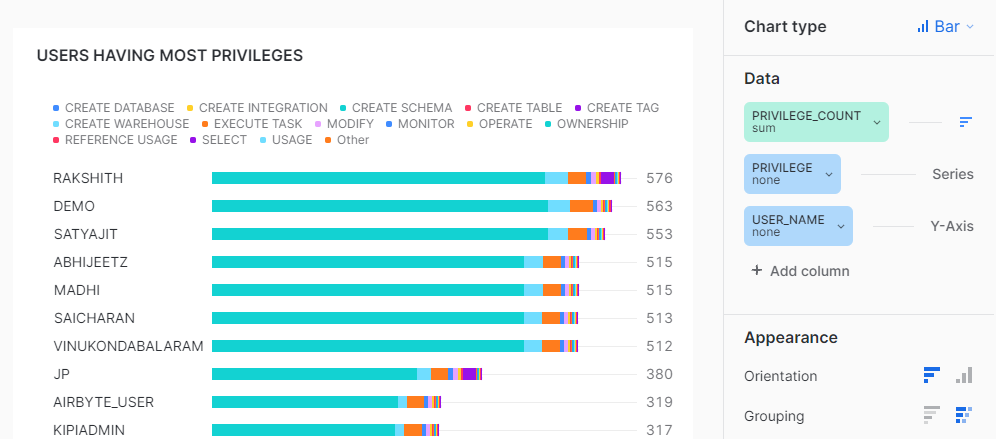
deleted\_on is null

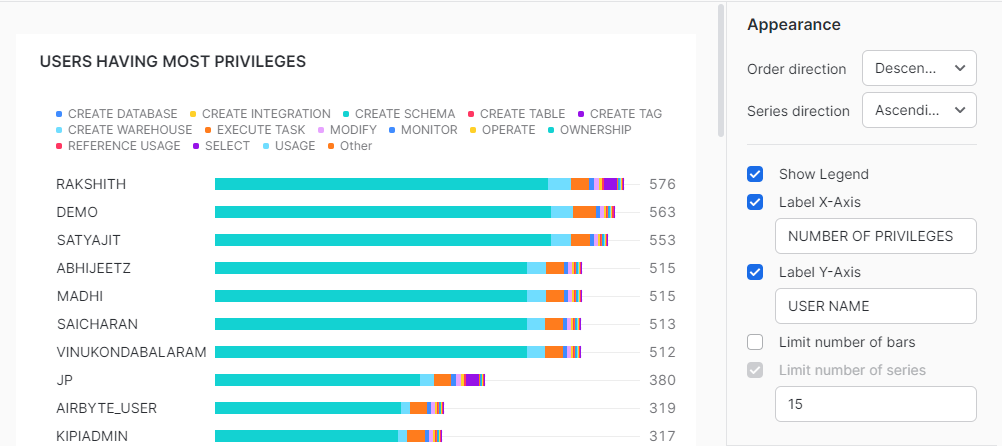
)

group by

gtu.grantee\_name,

gtr.privilege;





### **6 USERS NOT ENROLLED FOR MFA THROUGH DUO**

select

distinct(NAME) as User\_Name

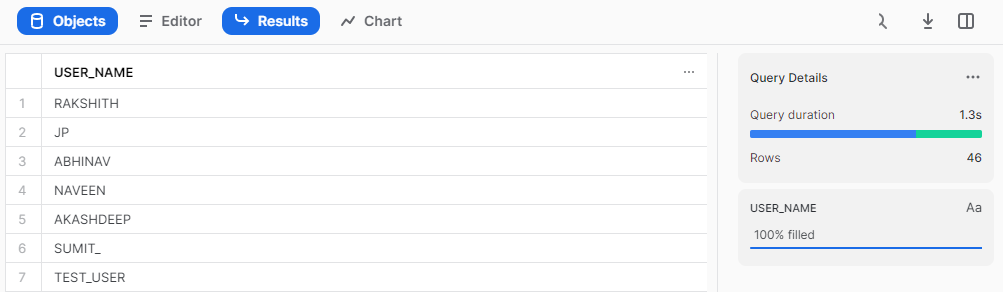
from

"SNOWFLAKE"."ACCOUNT\_USAGE"."USERS"

where

deleted\_on is null

and ext\_authn\_duo = 'false';



### **7 DISCONNECTED ROLE**

select

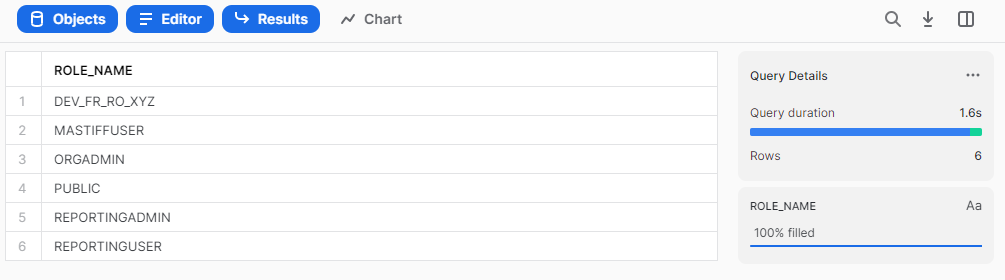
Name as Role\_Name

from

MONITOR\_DB.SECURITY\_MONITOR\_SCHEMA.ROLES\_INFO\_TB

WHERE

ROLE\_TYPE = 'Disconnected Role';



### **8 INACTIVE USERS BY QUERY EXECUTION**

select

count(query\_id) Number\_of\_Queries,

user\_name

from

"SNOWFLAKE"."ACCOUNT\_USAGE"."QUERY\_HISTORY"

where

USER\_NAME in (

Select

Distinct(Name)

from

snowflake.account\_usage.users

where

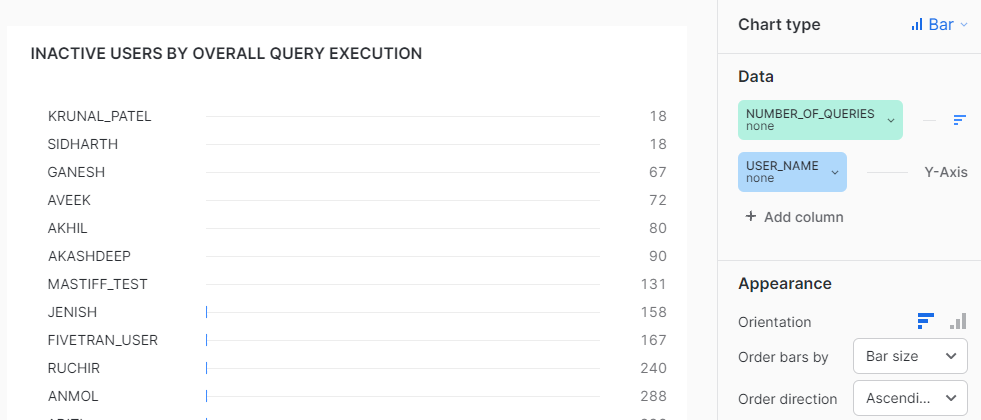
deleted\_on is null

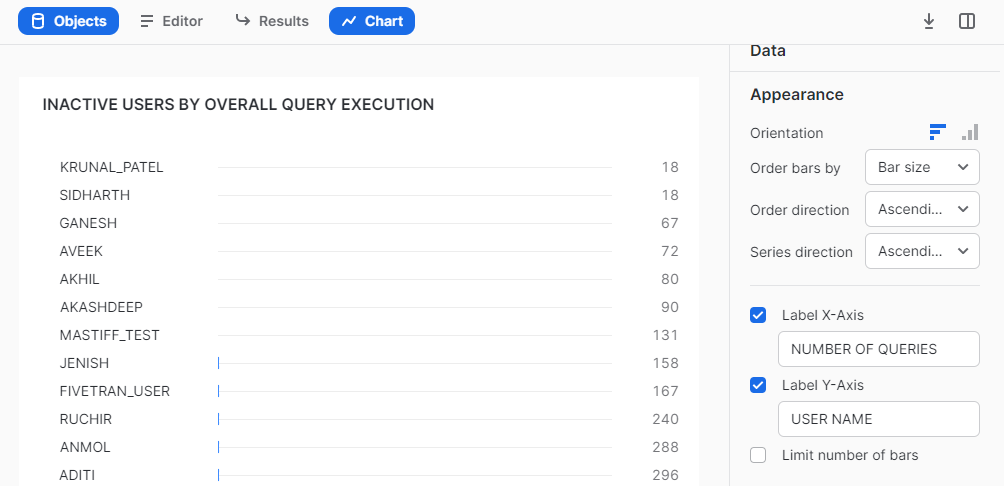
)

and start\_time = :daterange

group by

user\_name;





### **9 INACTIVE USERS BY LAST SUCCESS LOGIN**

select

NAME,

LAST\_SUCCESS\_LOGIN

from

"SNOWFLAKE"."ACCOUNT\_USAGE"."USERS"

where

deleted\_on is null minus

select

NAME,

LAST\_SUCCESS\_LOGIN

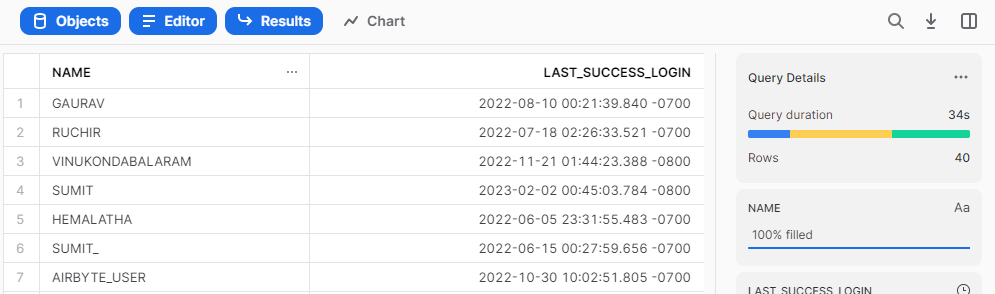
from

"SNOWFLAKE"."ACCOUNT\_USAGE"."USERS"

where

LAST\_SUCCESS\_LOGIN = :daterange

and deleted\_on is null;



### **10 INACTIVE ROLE BY QUERY EXECUTION**

select

count(query\_id) Number\_of\_Queries,

role\_name

from

"SNOWFLAKE"."ACCOUNT\_USAGE"."QUERY\_HISTORY"

where

ROLE\_NAME in (

Select

Distinct(Name)

from

snowflake.account\_usage.roles

where

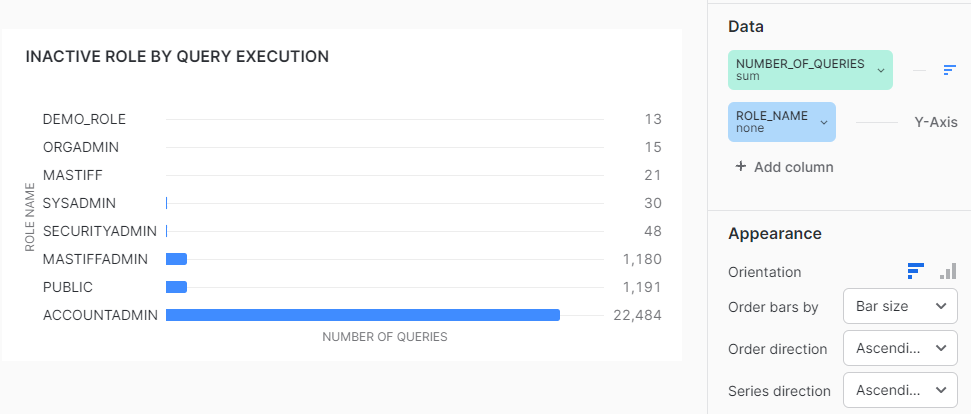
deleted\_on is null

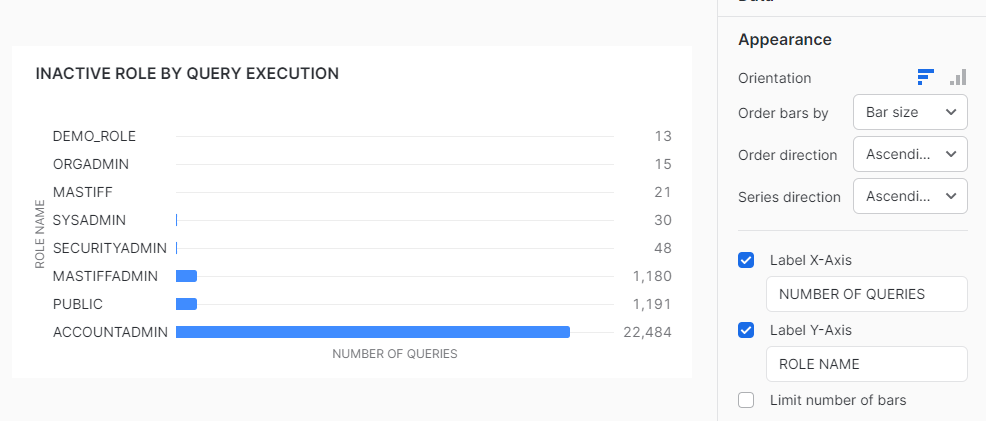
)

and start\_time = :daterange

group by

role\_name;





### **11 AUTHENTICATION BREAKDOWN**

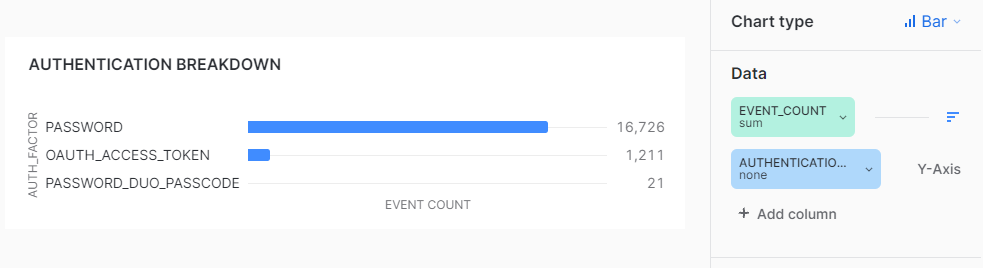
select

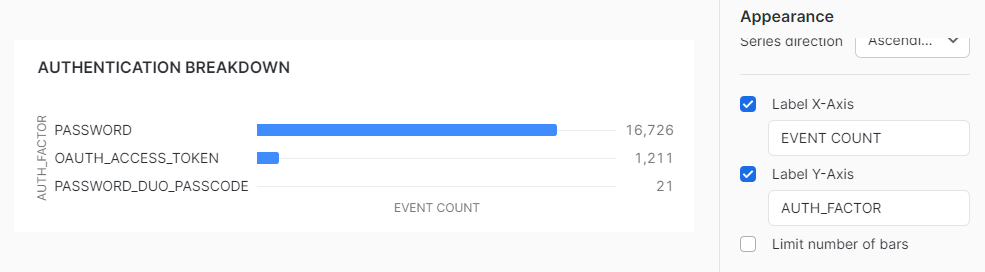
AUTHENTICATION\_FACTOR,

EVENT\_COUNT

from

MONITOR\_DB.SECURITY\_MONITOR\_SCHEMA.AUTHENTICATION\_BREAKDOWN\_TB;





### **12 TOTAL SIGN-IN ERROR BY ERROR TYPE**

select

user\_name,

event\_timestamp,

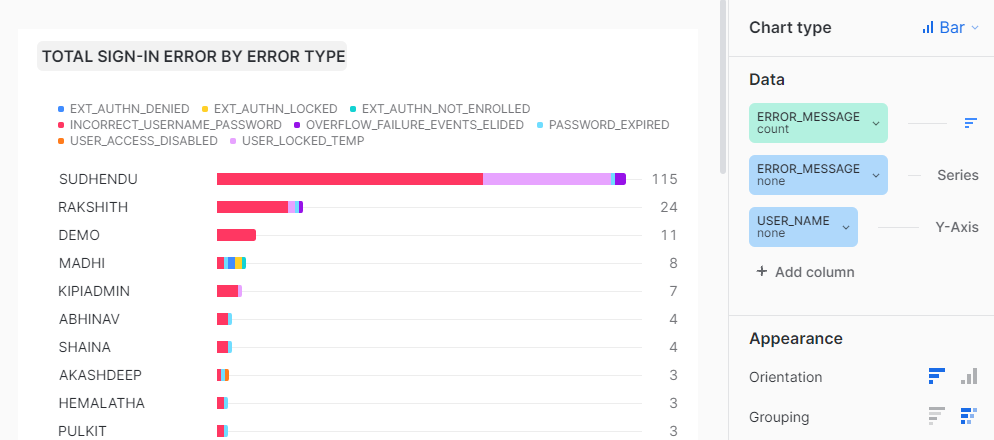
error\_message

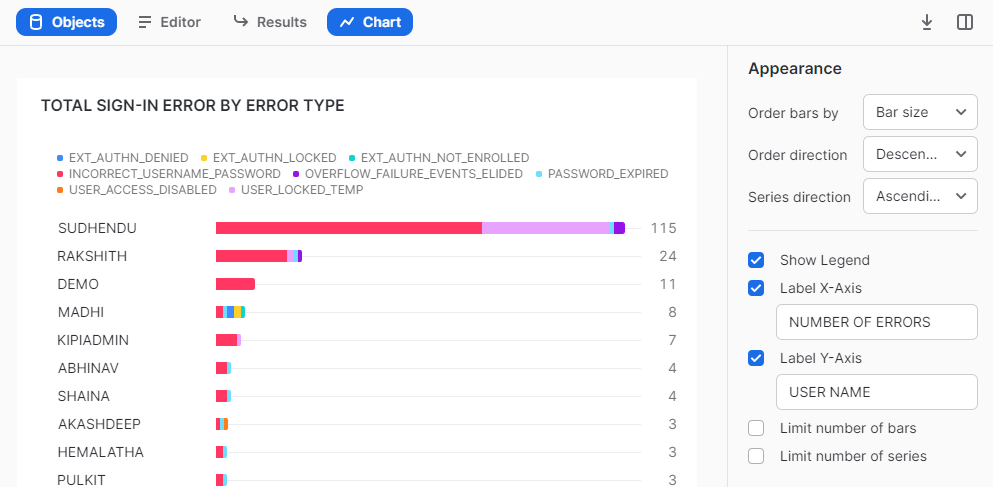
from

"SNOWFLAKE"."ACCOUNT\_USAGE"."LOGIN\_HISTORY"

where

is\_success = 'NO';





### **13 LEAST USED ROLE GRANTS FOR USERS**

select

User\_name,

Role\_name,

count(\*)

from

"SNOWFLAKE"."ACCOUNT\_USAGE"."QUERY\_HISTORY"

where

USER\_NAME in (

Select

Distinct(Name)

from

snowflake.account\_usage.users

where

deleted\_on is null

)

and role\_name in (

Select

Distinct(Name)

from

snowflake.account\_usage.roles

where

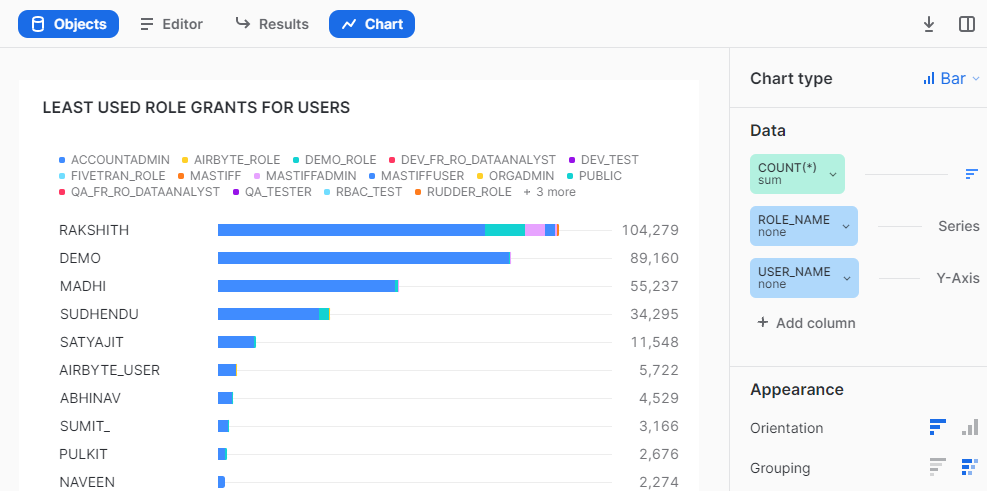
deleted\_on is null

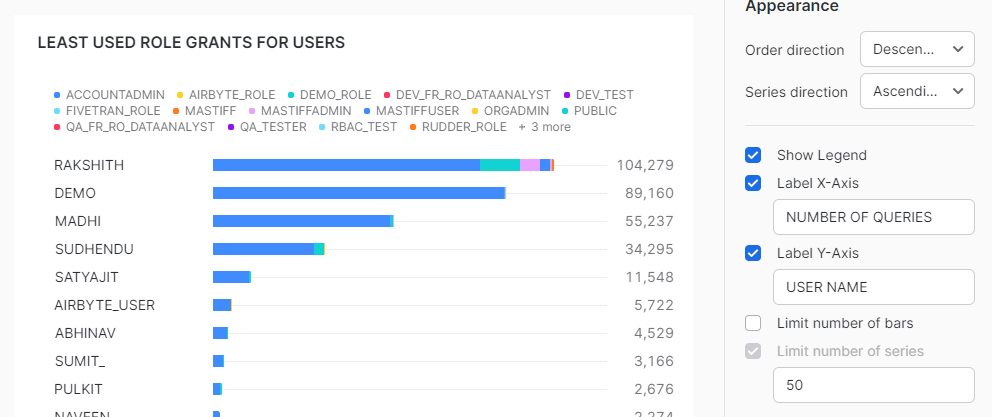
)

group by

user\_name,

Role\_name;





### **14 INDIVIDUAL USER TO ROLE RATIO**

select

1 / count(Distinct(role)) as User\_to\_Role\_Ratio,

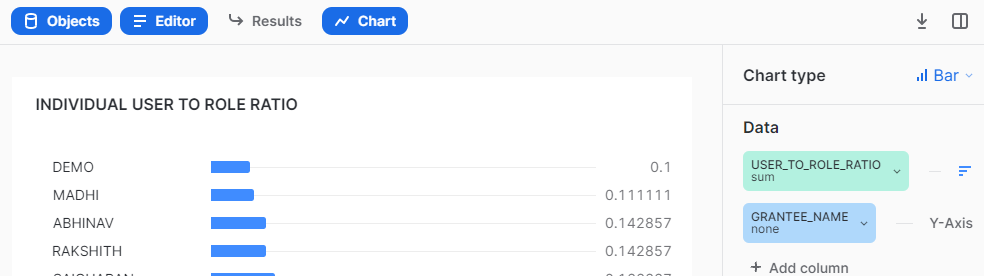
Grantee\_name

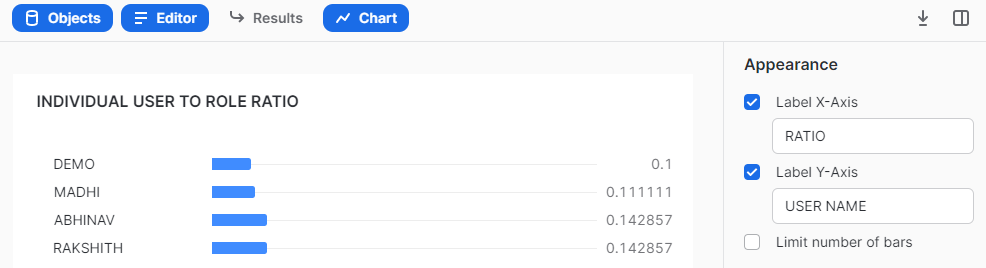
from

"SNOWFLAKE"."ACCOUNT\_USAGE"."GRANTS\_TO\_USERS"

group by

Grantee\_name;





### **15 USER GRANTING ACCOUNTADMIN PRIVILEGES**

select

count(QUERIES),

USER\_NAME,

ROLE\_NAME

from

(

Select

UPPER(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

where

query\_type in ('GRANT')

)

where

QUERIES LIKE 'GRANT ROLE ACCOUNTADMIN%'

group by

USER\_NAME,

ROLE\_NAME

union

select

count(QUERIES),

USER\_NAME,

ROLE\_NAME

from

(

Select

UPPER(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

where

query\_type in ('GRANT')

)

where

QUERIES LIKE 'GRANT ROLE "ACCOUNTADMIN%'

group by

USER\_NAME,

ROLE\_NAME

union

select

count(QUERIES),

USER\_NAME,

ROLE\_NAME

from

(

Select

UPPER(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

where

query\_type in ('GRANT')

)

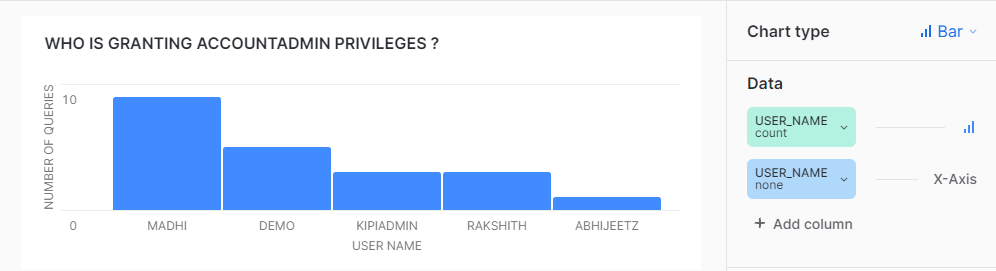
where

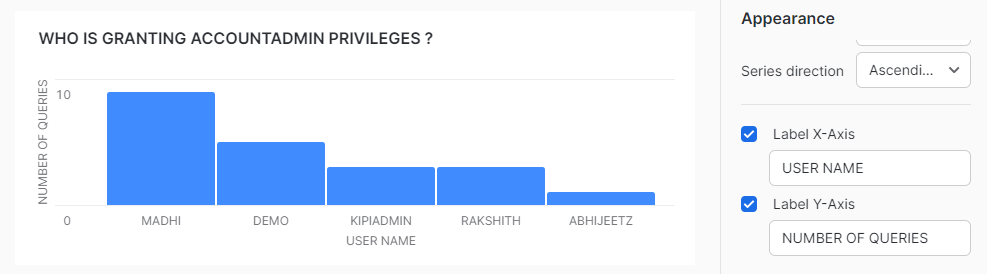
QUERIES LIKE 'GRANT ROLE IDENTIFIER%ACCOUNTADMIN%'

group by

USER\_NAME,

ROLE\_NAME;





### **16 ACCOUNTADMIN PRIVILEGE GRANTED TO WHOM**

select

\*

from

(

Select

upper(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME,

EXECUTION\_STATUS

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

where

query\_type in ('GRANT')

)

where

QUERIES LIKE 'GRANT ROLE ACCOUNTADMIN%'

union

select

\*

from

(

Select

upper(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME,

EXECUTION\_STATUS

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

where

query\_type in ('GRANT')

)

where

QUERIES LIKE 'GRANT ROLE "ACCOUNTADMIN%'

union

select

\*

from

(

Select

upper(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME,

EXECUTION\_STATUS

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

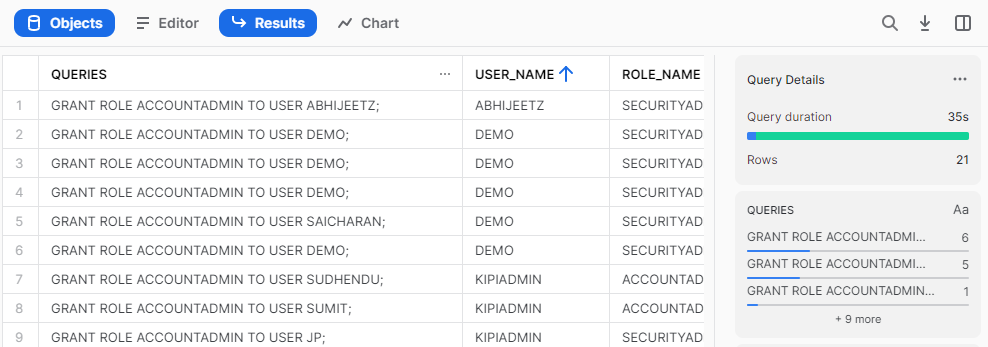
where

query\_type in ('GRANT')

)

where

QUERIES LIKE 'GRANT ROLE IDENTIFIER%ACCOUNTADMIN%';



### **17 GHOST USERS (USERS WITHOUT ROLES)**

select

name as USER\_NAME

from

SNOWFLAKE.ACCOUNT\_USAGE.USERS

where

deleted\_on is null

and USER\_NAME not in (

select

distinct(grantee\_name)

from

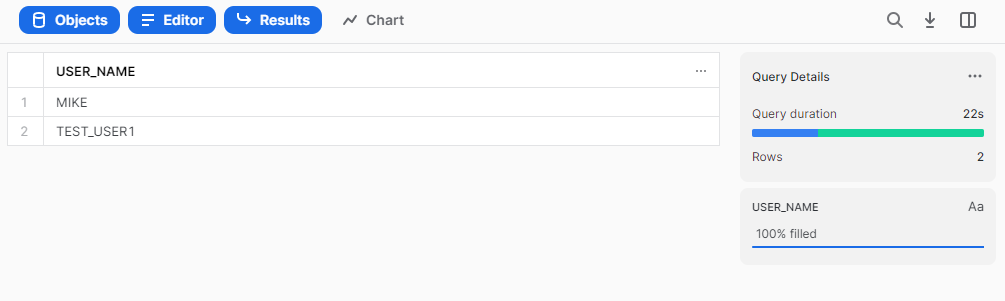
SNOWFLAKE.ACCOUNT\_USAGE.grants\_to\_users

where

deleted\_on is null

)

and name <> 'SNOWFLAKE';



### **18 GHOST ROLES (ROLES WITHOUT PRIVILEGES)**

Select

name as Role\_Name

from

SNOWFLAKE.ACCOUNT\_USAGE.ROLES

where

Role\_Name not in (

Select

distinct(grantee\_name)

from

SNOWFLAKE.ACCOUNT\_USAGE.GRANTS\_TO\_ROLES

where deleted\_on is null

)

and Role\_name in (

Select

Distinct(Name)

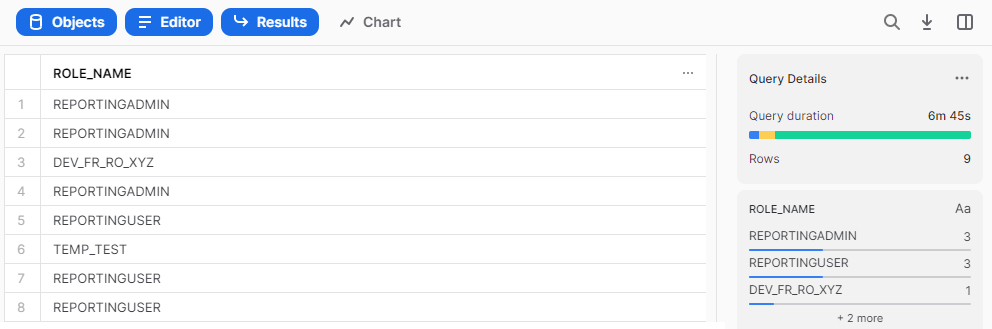
from

snowflake.account\_usage.roles

where

deleted\_on is null

);



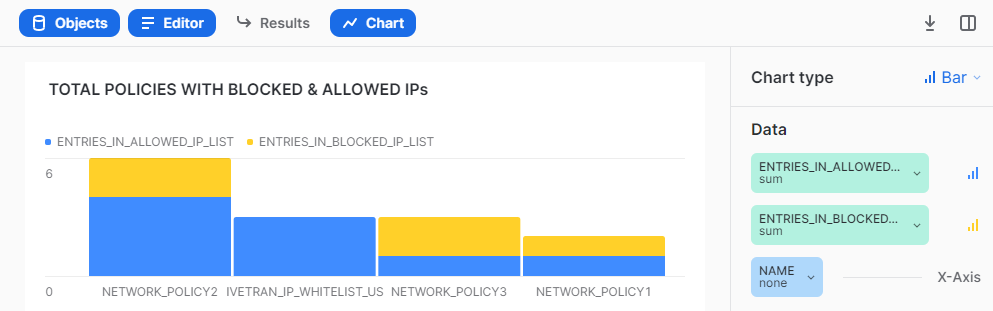
### **19 TOTAL POLICIES WITH BLOCKED & ALLOWED IPs**

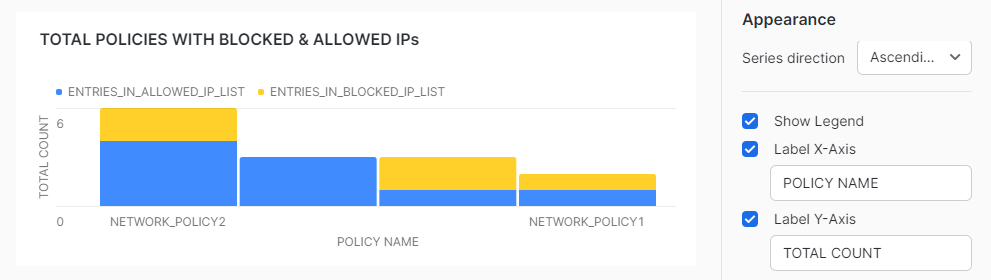
select

\*

from

MONITOR\_DB.SECURITY\_MONITOR\_SCHEMA.NETWORK\_POLICIES\_TB;





### **20 USER ALTERING NETWORK POLICIES**

select

\*

from

(

Select

upper(QUERY\_TEXT) as QUERIES,

USER\_NAME,

ROLE\_NAME,

START\_TIME

from

SNOWFLAKE.ACCOUNT\_USAGE.QUERY\_HISTORY

where

query\_type in ('ALTER\_NETWORK\_POLICY')

and execution\_status = 'SUCCESS'

and USER\_NAME in (

Select

Distinct(Name)

from

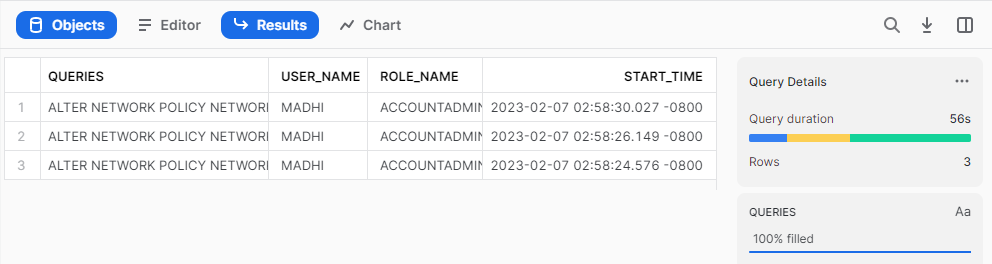
snowflake.account\_usage.users

where

deleted\_on is null

)

);



## **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.
* All the KPIs for security monitor currently include disabled users. To exclude disabled users from the KPI calculations, simply add "disabled is null" in the WHERE conditions.