# **DBT JOBS - ERROR TRACKING**

Tracking DBT job errors through Snowsight dashboards allows you to identify and resolve issues in your data transformation workflows effectively. Snowsight provides real-time visibility into errors encountered during the execution of DBT jobs. You can monitor error rates, identify patterns, and track the types of errors occurring across different jobs or time periods. Snowsight's data visualization capabilities enable you to create custom metrics and visualizations specific to error tracking. This helps you quickly identify common error messages, their frequency, and their impact on job performance. Snowsight dashboards can be shared with relevant stakeholders, promoting collaboration and enabling teams to work together to address errors and improve the reliability of data transformation processes.

With below KPIs on Error Tracking of DBT jobs, you can proactively identify and address issues, ensuring the accuracy and quality of your data transformations.

## 

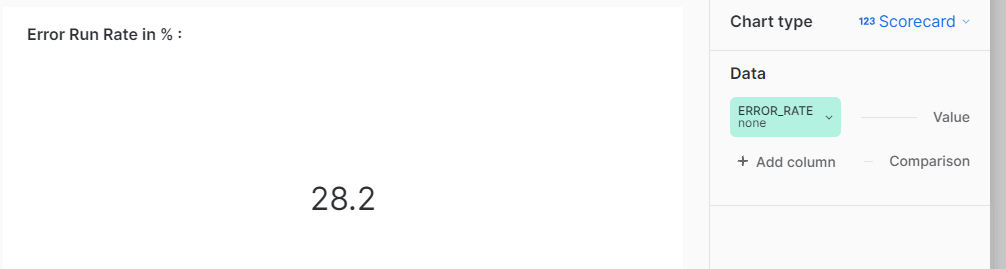
## **Error Run Rate in % :**

select (select count(\*) from runs\_list\_atm where status='Error') as error\_count,

(select count(\*) from runs\_list\_atm) as total\_count,

round((error\_count/total\_count)\*100,2) as error\_rate;





## 

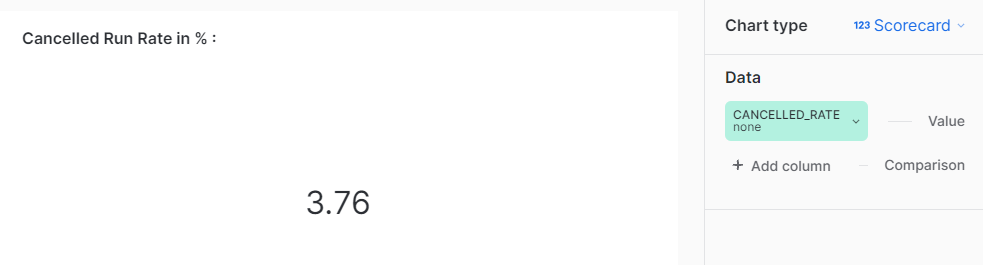
## Canceled Run Rate in % :

select (select count(\*) from runs\_list\_atm where status='Cancelled') as cancel\_count,

(select count(\*) from runs\_list\_atm) as total\_count,

round((cancel\_count/total\_count)\*100,2) as cancelled\_rate;



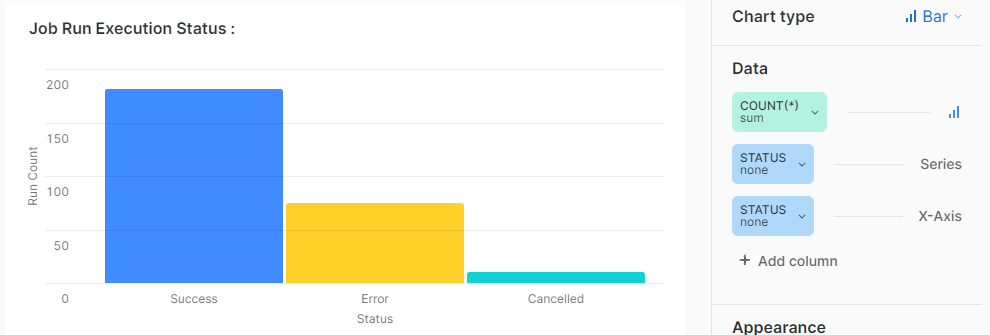


## 

## Job Run Execution Status :

select count(\*), status from runs\_list\_atm group by status





## 

## Error Rates on Each Job

SELECT

j.job\_name,

COUNT(CASE WHEN status = 'Error' THEN 1 END) AS error\_count,

COUNT(\*) AS total\_count,

ROUND((COUNT(CASE WHEN status = 'Error' THEN 1 END) / COUNT(\*)) \* 100, 2) AS error\_rate

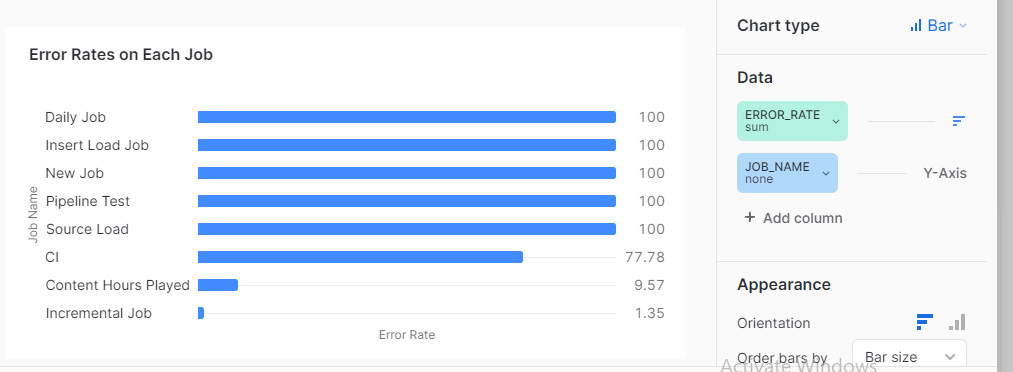
FROM runs\_list\_atm r

join

jobs\_list\_atm j on r.job\_id=j.job\_id

GROUP BY j.job\_name





## 

## Canceled Jobs Rate :

SELECT

j.job\_name,

COUNT(CASE WHEN status = 'Canceled' THEN 1 END) AS cancel\_count,

COUNT(\*) AS total\_count,

ROUND((COUNT(CASE WHEN status = 'Canceled' THEN 1 END) / COUNT(\*)) \* 100, 2) AS cancel\_rate

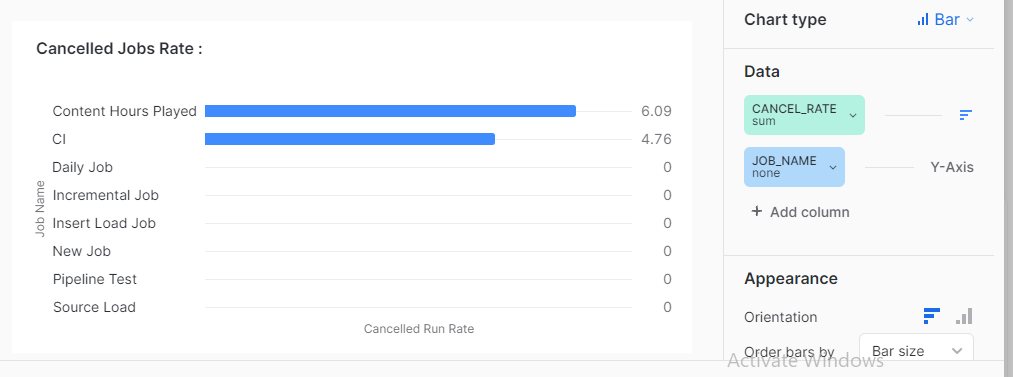
FROM runs\_list\_atm r

join

jobs\_list\_atm j on r.job\_id=j.job\_id

GROUP BY j.job\_name





## 

## Unsuccessful Runs trend

SELECT

TRUNC(created\_at, 'DAY') AS run\_date,

status,

COUNT(\*) AS run\_count

FROM

runs\_list\_Atm

WHERE status in ('Error','Canceled')

GROUP BY

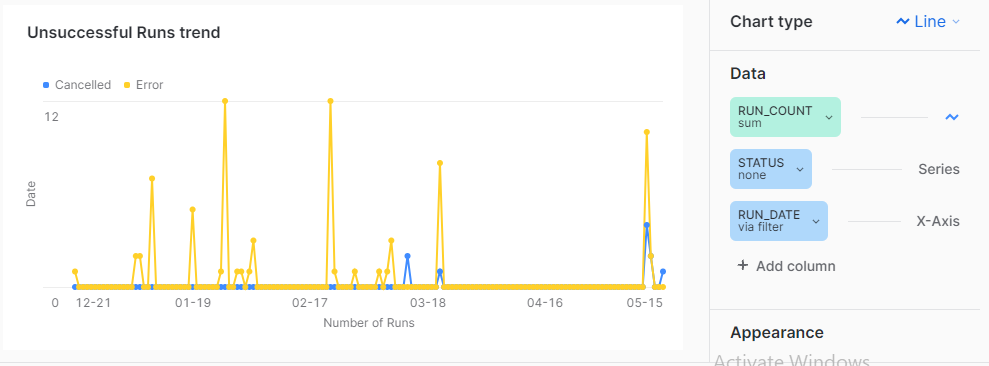
run\_date,

status

ORDER BY

status desc;





## 

## Jobs terminated due to Snowflake errors :

SELECT ERROR\_MESSAGE , count(\*) ,

CASE

WHEN ERROR\_CODE =002003 THEN 'Compilation Error'

WHEN ERROR\_CODE =000630 THEN 'Statement or Warehouse Timeout'

WHEN ERROR\_CODE =000604 THEN 'Object does not exist'

ELSE 'Others'

END AS category

FROM DBT\_DASHBOARD\_DB.ACCOUNT\_USAGE.QUERY\_HISTORY

WHERE ERROR\_CODE IS NOT NULL

group by category,error\_message;





## 

## Job runs canceled by DBT :

(select (split(status\_message,'.')[0]) as run\_terminated, status,count(\*), job\_name from dbt\_dashboard\_db.dbt\_coe.runs\_list\_atm

join dbt\_dashboard\_db.dbt\_coe.jobs\_list\_atm using (job\_id)

where (status\_message ILIKE '%canceled%' ) and (status\_message NOT ILIKE '%User%') and (status\_message NOT LIKE '%user%')

group by status, status\_message, job\_name)

union all

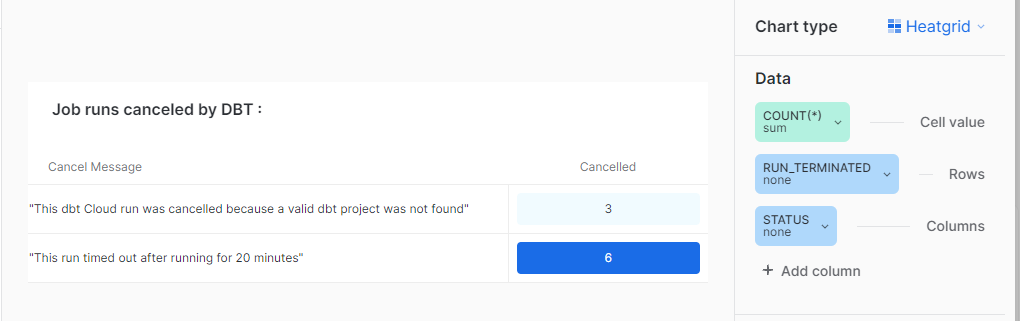
(select (split(status\_message,',')[0]) as message, status,count(\*),job\_name from dbt\_dashboard\_db.dbt\_coe.runs\_list\_atm

join dbt\_dashboard\_db.dbt\_coe.jobs\_list\_atm using (job\_id)

where status\_message NOT ILIKE '%canceled%'

group by message, status, job\_name)





## Job runs canceled by User :

(select (split(status\_message,'.')[0]) as user\_terminated, status,count(\*), job\_name from runs\_list\_atm

join jobs\_list\_atm using (job\_id)

where lower(status\_message) LIKE '%user%'

group by status, status\_message, job\_name)



