## Custom Email Notifications Setup Based on Thresholds Context:

To illustrate the email notification setup, we utilize two key tables: QUERY\_TIME and REFERENCE\_TIME.

The QUERY\_TIME table captures essential details such as Time Elapsed for each query and associated warehouse names. Conversely, the REFERENCE\_TIME table stores corresponding thresholds for each warehouse.

A Stored Procedure is implemented to monitor queries that surpass their designated threshold times. When such an event occurs, an alert is triggered. A designated Task invokes this Stored Procedure.

**Email Integration:**

To send an email from Snowflake we need to set email integration.

SET emailAddress = '\*\*\*\*\*\*\*\*@kipi.bi';

CREATE OR REPLACE NOTIFICATION INTEGRATION

SF\_Email\_Notifications

TYPE=EMAIL

ENABLED=TRUE

ALLOWED\_RECIPIENTS=($emailAddress)

COMMENT = 'Snowflake Email Notifications';



This SQL query sets up a notification integration named "SF\_Email\_Notifications" that sends notifications to the email address stored in the emailAddress variable. The integration is enabled and configured to send notifications when certain events occur in the Snowflake database environment.

SHOW INTEGRATIONS;



The SQL query is used to retrieve a list of all the integration objects that have been created in the database

**Stored Procedure:**

create or replace procedure email\_sender()

RETURNS STRING NOT NULL

LANGUAGE JAVASCRIPT

AS

$$

{

var query\_ids = [];

var alert\_count=snowflake.createStatement({ sqlText:`SELECT QUERY\_ID FROM QUERY\_TIME JOIN REFERENCE\_TIME ON QUERY\_TIME.WAREHOUSE\_SIZE = REFERENCE\_TIME.WH\_SIZE WHERE QUERY\_TIME.TOTAL\_ELAPSED\_TIME > REFERENCE\_TIME.TIMEOUT\_1`});

var excute\_count=alert\_count.execute();

while (excute\_count.next()) {

query\_ids.push(excute\_count.getColumnValue("QUERY\_ID"));

}

var output = "";

for (var i = 0; i < query\_ids.length; i++) {

output += query\_ids[i];

if (i < query\_ids.length - 1) {

output += "\n";

} }

var alert\_stat=snowflake.createStatement({ sqlText:`call system$send\_email(

'SF\_Email\_Notifications',

'\*\*\*\*\*\*\*\*@kipi.bi',

'[Snowflake Email Sending Test]:Test.',

'Queri\_Ids: ''` + output + `''.End Time: ' || CURRENT\_TIMESTAMP() || 'UTC \nPlease check the above:' || ` + query\_ids.length+` || 'Queries. They exceeded the Threshold limit.'

);`});

var excute\_state=alert\_stat.execute();

return 'success';

}

$$;

This stored procedure is designed to perform the following tasks:

* The email\_sender stored procedure checks the QUERY\_TIME table to identify queries that have exceeded their threshold execution time based on the information stored in the REFERENCE\_TIME table.
* It retrieves the query IDs from the queries that exceed the threshold and constructs a formatted message containing these query IDs.
* It then uses the system$send\_email procedure to send an email notification. The email is sent using the email integration named 'SF\_Email\_Notifications'.
* The email contains information about the queries that exceeded the threshold, the current timestamp, and the total number of queries that exceeded the threshold.

In this case we have used QUERY\_TIME table as the main table, REFERENCE\_TIME table as a threshold reference table, QUERY\_TIME.TOTAL\_ELAPSED\_TIME column as query run time and REFERENCE\_TIME.TIMEOUT\_1 column as threshold time but the users can customize the tables according to their needs.

**Task:**

Create or replace task task\_email\_sender

warehouse=COMPUTE\_WH

schedule='USING CRON 30 11 \* \* \* UTC'

As call email\_sender();

alter task task\_email\_sender resume;



This block of code creates a Snowflake task named “task\_email\_sender” that is scheduled to run at 11:30 AM UTC using a CRON expression. The user can update this schedule to send the notifications for a different frequency. The task uses the COMPUTE\_WH warehouse and executes the email\_sender() stored procedure. After creating the task, it is immediately resumed, meaning it is set to start executing according to the specified schedule.

**Check Emails Sent:**

 SELECT CREATED

,PROCESSED

,MESSAGE\_SOURCE

,INTEGRATION\_NAME

,MESSAGE

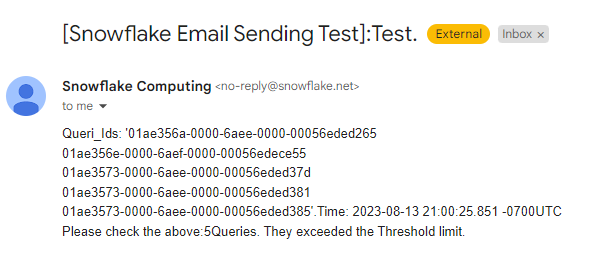
,STATUS

FROM TABLE(Snowflake.information\_schema.notification\_history(integration\_name => 'SF\_Email\_Notifications'));



The result of this query will be a result set that includes the selected columns for each notification in SF\_Email\_Notifications. This information can help you track the history of email notifications sent through this integration.

Sample Screenshot of the email received from Snowflake



Attaching the threshold limit sheet for reference ([link](https://docs.google.com/spreadsheets/d/1ASj9yhXK3HbSpxVkJDMY4nUwVvypHxy5f5S40m3Io7E/edit#gid=1407023218)).