

# CONFIGURATION



# Table of Contents

Home Page .....	3
Why doesn't my Category appear on the home page? .....	4
Why don't I see any tiles for my dimensioned element on the home page, even though it's enabled and visible?.....	5
Why are all the tiles missing from the Home Page? .....	6
Data Sources .....	8
Error: You cannot save password of Data Source .....	9
User Editor .....	12
Why do I have to uncheck the LDAP User option to edit the username? .....	13
Fiscal Calendar .....	15
How do you change the fiscal calendar in Metric Insights? .....	16
Dimensions.....	22
How do you delete dimension values en masse? .....	23
SSL Certificates.....	25
SSL warning: NET::ERR_CERT_COMMON_NAME_INVALID .....	26
Sample SQL Queries .....	28
MySQL queries for various tasks.....	29
MySQL queries for Smart Folders: Usage and Engagement Stats .....	46

# Home Page

# Why doesn't my Category appear on the home page?

If your category doesn't appear on the home page but you have elements in it, most likely elements that belong to this category are not actually set to be 'visible' on the home page.

<http://prntscr.com/9lwjjx>

If a category has no visible elements, then that category is not displayed on the home page. So if we you make your elements visible, the category should appear automatically.

# Why don't I see any tiles for my dimensioned element on the home page, even though it's enabled and visible?

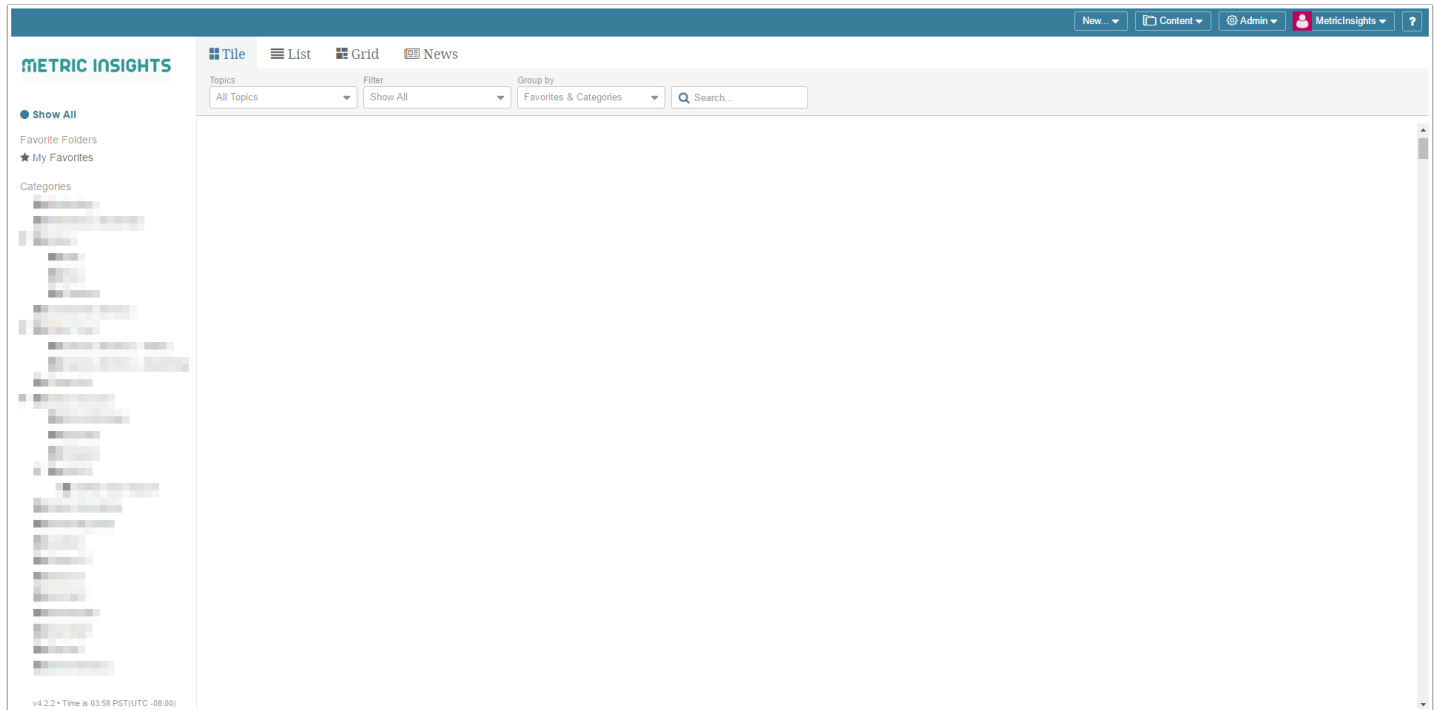
The first thing you should check is if the dimension values of the dimension assigned to the element are 'visible' (Show as Tile) <http://prntscr.com/9lwlra>

MI will only display tiles on the home page for dimension values that are set to 'visible'. To find the current settings, edit your dimension (<http://img.mtrc.in/100d0J0b0H3k>) and then check the dimension values grid at the bottom of the page (<http://img.mtrc.in/1y0s1i2C1p2A>). You can adjust the visibility settings individually (<http://img.mtrc.in/3J1o0v1H1p0H>) or en masse (<http://img.mtrc.in/251s2A1S0m39>).

# Why are all the tiles missing from the Home Page?

## Issue

Our Metric Insights home page normally has many tiles displayed, but today all the tiles are missing! This is happening for all users. What is going on?




## Resolution

The tiles can *disappear* from the home page when the number of tiles exceeds the maximum allowable limit as set by the variable **MAXIMUM\_HOME\_PAGE\_TILE\_COUNT**. This typically happens as users build out multiple dimensioned elements over time (each dimension resulting in the creation of a tile). To fix this, simply increase the maximum allowable limit by adjusting the value of the `MAXIMUM_HOME_PAGE_TILE_COUNT` variable:

1. Login as the admin user on your Metric Insights instance
2. Go to the Admin menu > Utilities > Config Variables
3. Search for the variable `MAXIMUM_HOME_PAGE_TILE_COUNT`
4. Click the gear icon to change the value of the variable
5. Enter a value greater than what's already defined
6. Save and Commit Changes


You should now see your tiles on the Home Page!

 Config Variables

New... ▾

Content ▾

Admin ▾

 MetricInsights ▾

Change advanced system settings. Changes are not applied until they are committed.


All ▾


✕


MAXIMUM\_HOME\_PAGE\_

Config Variables

● Uncommitted Changes

Variable Name	Assigned Value	Valid Values	Description
MAXIMUM_HOME_PAGE_TILE_COUNT	40000		The maximum number of tiles that will I... 

 Discard Changes

 Commit Changes

# Data Sources



# Error: You cannot save password of Data Source

## Issue

If you had moved the MI app from the /opt directory to a new location (e.g. /app) and then got an issue with Data Source's password saving (password disappeared and there is no possibility to save a new one).

The screenshot shows the 'SQL Data Sources / Dashboard DB' configuration page. The 'Data Source Password' field is highlighted with a red rectangle. The page includes fields for Name (Dashboard DB), Data Source Username (ml\_read), Data Source Password (highlighted), Host name, Database name (dashboard), JDBC driver (MySQL Connector/J), Port (3306), and JDBC string. There are also radio buttons for 'Use Remote Data Collector?' (yes/no), 'Use visual editor?' (yes/no), and 'Infer foreign keys' (yes/no). At the bottom, there are buttons for 'Permissions', 'Save & test connection', 'Refresh metadata', and 'Save'.

## Resolution

Most likely something went wrong during symlinking files to a new location. In order to fix this, please follow these steps:

1. Open terminal (ssh connection) for Instance where MI installed and check path for encryption file:
  - check content of file insight.conf (run the command: `cat /app/mi/config/insight.conf`);
  - pay attention on key\_path for encryption;

```
[root@localhost config]# cat /opt/mi/config/insight.conf
[DEFAULT]
port = 3306
host = localhost
database = dashboard
time_zone =
sql_trace = 0

[read_only]
username = mi_read
password = TIAAnJ0or

[web]
username = mi_web
password = NpPfos2qPU

[generator]
username = mi_gen
password = 98TT8hnVkJ

[insightd]
username = mi_read
password = TIAAnJ0or

[dataset_write]
username = mi_gen
password = 98TT8hnVkJ

[setup]
username = mi_setup
password = txaMerQcsP

[pyinsight]
username = mi_pyweb
password = OSjLm1YHxX

[aqb]
username = mi_aqb
password = bYhbphq49w

[dataset_read]
username = mi_read
password = TIAAnJ0or

[encryption]
key_storage = file
key_path = /opt/mi/aes_password
```

- if it is incorrect (e.g. /opt/mi/aes\_password but should be /app/mi/aes\_password), then you need to change it to correct one /app/mi/aes\_password and save.

2. Check path to mi-crypt script in config\_variable table in dashboard database and update it if it's wrong:

- check path to mi-crypt script with statement:

```
SELECT value FROM config_variable WHERE name='MI_CRYPT_ABSOLUTE_PATH';
```

- if it is incorrect (e.g. /opt/mi/lib/mi/bin/mi-crypt but should be /app/mi/lib/mi/bin/mi-crypt), then you need to update dashboard db with statement:

```
UPDATE config_variable SET value='/app/mi/lib/mi/bin/mi-crypt' WHERE
name='MI_CRYPT_ABSOLUTE_PATH';
```

- And then login into MI application in browser and perform the following steps:  
Go to Admin > Utilities > Config Variables and click "Commit changes"

Config Variables
 

New...

Content

Admin

Oleksandr

Change advanced system settings. Changes are not applied until they are committed.

All

Filter by name

Config Variables

Uncommitted Changes

Variable Name	Assigned Value	Valid Values	Description	
PORTAL_NAME	Metric Insights UAT		Name used to identify this Metric Insight...	
TOUR_AUTOSTART	Y	YN	Y = automatically starts available tours ...	
PASSWORD_RESET_LINK			IF you want a custom link for users to r...	
MAX_ANNOT_LEN	155		Maximum length of annotation text to di...	
MAX_MEAS_LIST	15		Maximum number of charting intervals ...	
MAX_RELATED_LIST	15		Maximum number of related items to s...	
MAX_SEGMENTS_LIST	15		Maximum number of dimension values...	
MAX_CUSTOM_INTERVALS_LIST	5		Maximum number of custom date rang...	
MAX_SEGMENT_VALUES_NUMBER	5000		Maximum number of dimension values...	
MAX_DATA_TABLE_DISPLAY_COLUM...	40		Maximum column width for any column...	
DIGEST_EXTERNAL_REPORT_LINK_ON	Chart_view	Chart_view,Collaborative_View	Parameter to control where user links c...	
SUPPRESS_REPORT_LINK_FOR_NO...	Y	YN	Set to Y to not show a link to the report ...	
MAX_DIGEST_ITEMS_FOR_EXCEL_D...	50		IF your digest includes excel files, this ...	
DIGEST_MAX_ATTACHMENTS_SIZE	10485760		Set the total attachment size in bytes. F...	
SHOW_ALERT_GLOSSARY_IN_EMAILS	Y	YN	Set to Y to show a glossary explaining ...	
MAX_USED_DISK_SPACE_PERCENT	80		Set the percent of disk space usage be...	
MAX_USED_INODE_PERCENT	90		Set the percent of inode usage before ...	
LDAP_SERVER	ldaps://bob.metricinsights.com		An LDAP or active directory server to c...	
LDAP_USER			Optional system user for querying LDAP	
LDAP_PASSWORD	*****		Optional system password for querying...	

Page 1 of 12

Displaying records 1 - 20 of 221

Discard Changes

Commit Changes

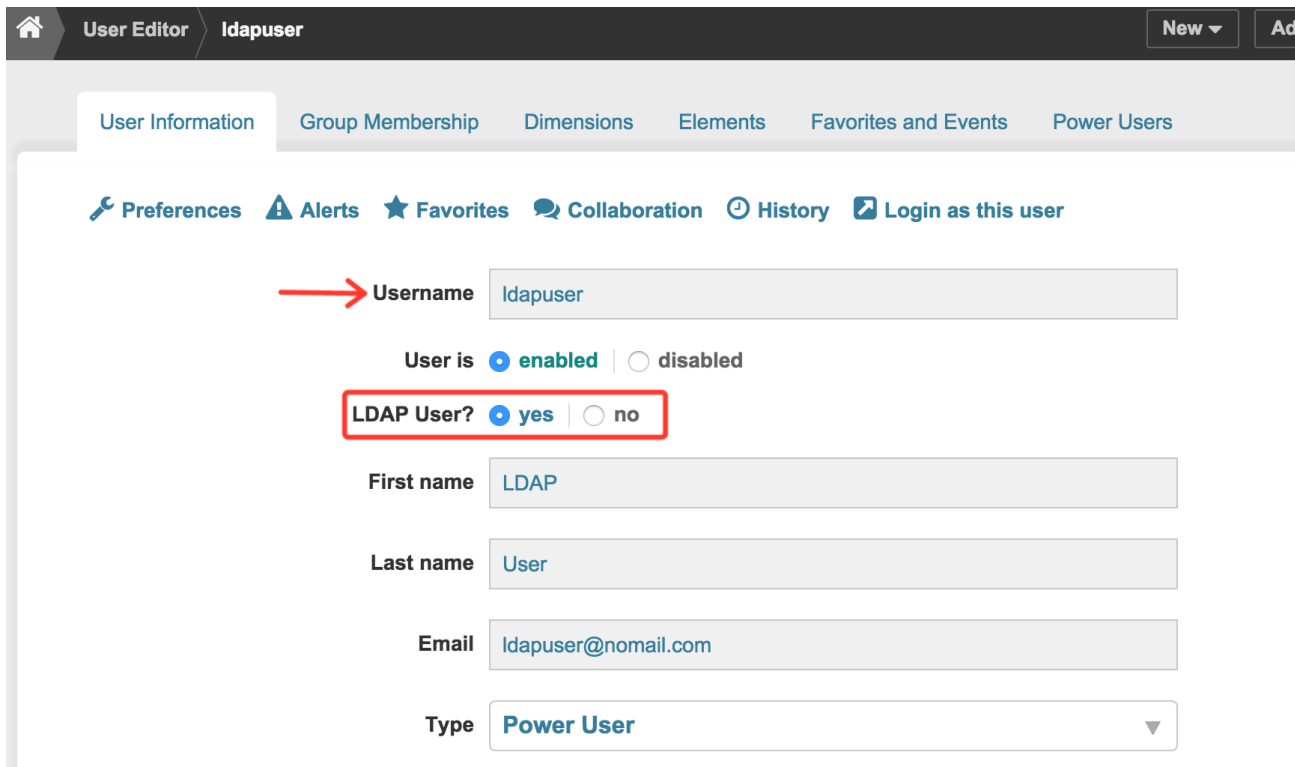
You have uncommitted changes...

# User Editor

# Why do I have to uncheck the LDAP User option to edit the username?

## ISSUE

We have an LDAP user set up in Metric Insights with the wrong username. In trying to correct that username, we found we had to first set the **LDAP user?** option to **no** in order to push through the change (then set it back to **yes** after the correction). Why is this step necessary?



The screenshot shows the 'User Editor' interface for a user named 'ldapuser'. The 'User Information' tab is selected. The 'LDAP User?' checkbox is highlighted with a red box and is currently checked (yes). A red arrow points to the 'Username' field, which contains 'ldapuser'. Other fields include 'First name' (LDAP), 'Last name' (User), 'Email' (ldapuser@nomail.com), and 'Type' (Power User).

## RESOLUTION

Having to uncheck the LDAP option is a function of how the LDAP feature works. Generally, if connecting to a company's LDAP, the user accounts will be created upon MI syncing with the LDAP server. This ensures the user information created in MI matches what's actually defined in LDAP (with the username serving as the primary key between Metric Insights and the LDAP server).

In this case, *the LDAP user was created manually in Metric Insights*. This in itself is okay to do, though not recommended. The key is ensuring the username correctly matches LDAP the first time around. Once configured, the username cannot be changed because MI defers management of the username to LDAP, therefore locking it down.

This makes sense considering how external authentication methods work (e.g., Active Directory, LDAP, or SAML Single Sign-On). A user that is mapped to an external authentication method is managed outside of Metric Insights so it should not be change-able from within Metric Insights.

Thus, if the LDAP username is wrong, you must set **LDAP user?** = **no** in order to correct it. Keep in mind, manually editing the username is only necessary if the user already has an extensive list of preferences set including favorite digests, elements, alerts, etc. If this is a newly created user profile, the alternative solution is to simply **delete** this user profile from Metric Insights, then have the user in question log in to MI with his/her LDAP credentials. Metric Insights will then confirm the credentials with the LDAP server and auto-create a new user profile with the correct information!

# Fiscal Calendar

# How do you change the fiscal calendar in Metric Insights?

## ISSUE

How do you change the fiscal calendar? Our company has a fiscal calendar that is different than what is defined in Metric Insights. I can't seem to find a Fiscal Calendar page in which to make these changes.

## SOLUTION

The fiscal calendar in Metric Insights is not yet available through the UI (to be surfaced in a future release). However, we can still make changes to the calendar directly in the database. To proceed, please follow the instructions below.

### 1.) Set up a Fiscal Calendar:

You can use the SQL code below if you obtain or create a csv file with the following columns saved as `fiscal_master.csv` (**csv file must be in the following format**):

fiscal\_month\_id:201601

period\_name:F16 P01 (MAY)

period\_number: 1

first\_day\_of\_period: 2015-04-25

last\_day\_of\_period: 2015-05-29

fiscal\_quarter\_id: 201601

quarter\_name: Q1 FY2016

quarter\_number: 1

first\_day\_of\_quarter: 2015-04-25

fiscal\_year\_id: 2016

year\_name: FY2016

first\_day\_of\_fiscal\_year: 2015-04-25

Note the sample entry for each column above. Some of the columns can be computed from other columns. This is not a minimal set, just a simple one that works.



**2.) Dump the existing fiscal calendar tables as a backup (you'll need root acces):**

```
~# mysqldump dashboard fiscal_month > /tmp/fiscal_month.sql
```

```
~# mysqldump dashboard fiscal_quarter > /tmp/fiscal_quarter.sql
```

```
~# mysqldump dashboard fiscal_year > /tmp/fiscal_year.sql
```

**3.) Now, create a fiscal master table to import the csv:**

**\*\*fiscal\_master\*\***

```
>create database temp;
```

```
>use temp;
```

```
>create table fiscal_master(
```

```
fiscal_month_id int,
```

```
period_name varchar(20),
```

```
period_number int,
```

```
first_day_of_period date,
```

```
last_day_of_period date,
```

```
fiscal_quarter_id int,
```

```
quarter_name varchar(9),
```

```
quarter_number int,
```

```
first_day_of_quarter date,
```

```
fiscal_year_id int,
```

```
year_name varchar(6),
```

```
first_day_of_fiscal_year date
```

```
);
```

**4.) Load the fiscal\_master table from your csv file to the various fiscal tables:**

**MySQL command to load the csv file into the temp.fiscal\_master table:**

```
mysql> load data local infile '/path/to/fiscal_master.csv' into table temp.fiscal_master fields
terminated by ',' ignore 1 lines;
```

**\*\*fiscal\_year\*\***

```
> use dashboard;
```

```
> truncate table fiscal_year;
```

```
> insert fiscal_year(fiscal_year_id, name, first_day_of_fiscal_year)
```

```
select distinct
```

```
fiscal_year_id,
```

```
year_name,
```

```
first_day_of_fiscal_year
```

```
from temp.fiscal_master
```

```
order by fiscal_year_id asc;
```

```
update fiscal_year y
```

```
set last_day_of_fiscal_year = (
```

```
select max(m.last_day_of_period)
```

```
from temp.fiscal_master m
```

```
where m.fiscal_year_id=y.fiscal_year_id
```

```
);
```

**\*\*fiscal\_quarter\*\***

```
> truncate table fiscal_quarter;
```

```
> insert fiscal_quarter(fiscal_quarter_id, fiscal_year_id, quarter_number, name,
first_day_of_quarter)
```

```
select distinct
```

```
fiscal_quarter_id,
```

```
fiscal_year_id,
```

```
quarter_number,
```

```
quarter_name,  
first_day_of_quarter  
from temp.fiscal_master  
order by fiscal_quarter_id asc;
```

```
> set @Row_ID=0;  
update fiscal_quarter  
set fiscal_quarter_seq = (  
select @Row_ID := @Row_ID + 1  
);
```

```
> update fiscal_quarter q  
set last_day_of_quarter = (  
select max(m.last_day_of_period)  
from temp.fiscal_master m  
where m.fiscal_quarter_id=q.fiscal_quarter_id  
);
```

### **\*\*fiscal\_month\*\***

```
> truncate table fiscal_month;
```

```
> insert fiscal_month(fiscal_month_id, name, fiscal_year_id, period_number, fiscal_quarter_id,  
first_day_of_period, last_day_of_period)  
select distinct  
fiscal_month_id,  
period_name,  
fiscal_year_id,  
period_number,  
fiscal_quarter_id,  
first_day_of_period,
```

```
last_day_of_period
```

```
from temp.fiscal_master
```

```
order by fiscal_month_id asc;
```

```
> set @Row_ID=0;
```

```
update fiscal_month
```

```
set fiscal_month_seq = (
```

```
select @Row_ID := @Row_ID + 1
```

```
);
```

```
> update fiscal_month mon
```

```
set duration_days = timestampdiff(day, first_day_of_period, last_day_of_period) + 1;
```

```
> update fiscal_month
```

```
set
```

```
last_year_fiscal_month_id = fiscal_month_id - 100,
```

```
last_month_fiscal_month_id = fiscal_month_id - 1;
```

```
> update fiscal_month
```

```
set last_month_fiscal_month_id = last_month_fiscal_month_id - 100 + 12
```

```
where last_month_fiscal_month_id/100 - floor(last_month_fiscal_month_id/100) = 0;
```

### **\*\*calendar\_day\*\***

```
> update calendar_day set fiscal_month_id = null;
```

```
> update calendar_day set fiscal_quarter_id = null;
```

```
> update calendar_day set fiscal_year_id = null;
```

```
> update calendar_day d
```

```
set d.fiscal_month_id = (
```

```
select m.fiscal_month_id  
from fiscal_month m  
where d.calendar_date between m.first_day_of_period and m.last_day_of_period  
);
```

```
> update calendar_day d  
join fiscal_month m  
on d.fiscal_month_id = m.fiscal_month_id  
set  
d.fiscal_quarter_id = m.fiscal_quarter_id,  
d.fiscal_year_id = m.fiscal_year_id;
```

**You've now updated your fiscal calendar in Metric Insights!**

# Dimensions

# How do you delete dimension values en masse?

## Question

I accidentally loaded thousands of dimensions values by accident. Trying to delete them through the UI is proving to be difficult, affecting system performance for other users. Is there a quicker way to delete the dimension values directly in the database?

## Solution

Before loading thousands of dimension values, really think about why they are needed and for what purpose they will serve. Often times, the need to load thousands of dimension values can be mitigated by utilizing an [Exception Report](#) instead (with no dimensions!).

In this scenario where you've already loaded the dimension values by mistake, you can run the following SQL statement to purge the dimension values from the database. Create mysqldumps of the following tables beforehand just in case. Note, you'll need root access to the Metric Insights server and the MySQL database to run the following statement:

```
SET @BULK_OPERATION_DISABLE_TRIGGERS=1;

DELETE FROM metric_measured_value
WHERE metric_id = <element id/s here>;

DELETE FROM last_dashboard_element_segment_value
WHERE element_id = <element id/s here>;

DELETE FROM dashboard_element
WHERE element_id = <element id/s here>;

DELETE FROM segment_value
WHERE segment_id = <dimension id/s here>;

DELETE FROM segment
WHERE segment_id = <dimension id/s here>;

CALL bulk_operation_affected_rows_process(NULL,NULL);

SET @BULK_OPERATION_DISABLE_TRIGGERS=NULL;
```

Note, the IDs can be found by going to the Element Editor and Dimension Editor in the UI and locating the ID number in the URL. For additional assistance, please contact [support@metricinsights.com](mailto:support@metricinsights.com)





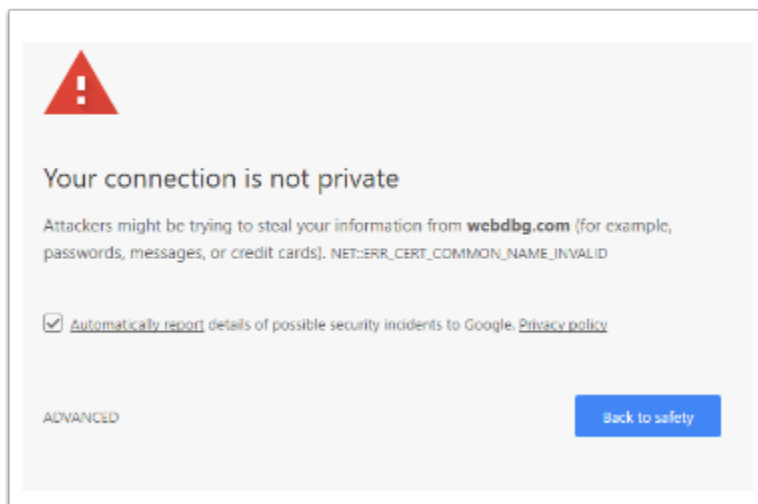
# SSL Certificates

# SSL warning: NET::ERR\_CERT\_COMMON\_NAME\_INVALID

## ISSUE

We are using self-signed certificates on the Metric Insights server provided for us by IT department.

When I try to open Metric Insights application in Chrome or Safari browser I get not private connection warning:



## RESOLUTION:

Chromium removed support for matching common name in certificates in M58:

- \* Feature bug (including motivation): [Issue 308330](#)
- \* <https://www.chromestatus.com/features/4981025180483584>

Certificates that rely on this deprecated behavior will now be rejected with:

ERR\_CERT\_COMMON\_NAME\_INVALID

The affected certificates are often locally generated ones for development purposes, or are part of a private PKI.

The solution is to re-generate the certificates to include a Subject Alternative Name extension, or to enable an option in Chrome to allow them.

Following articles should help you overcome the issue:

<https://stackoverflow.com/questions/43665243/chrome-invalid-self-signed-ssl-cert-subject-alternative-name-missing>

<https://textslashplain.com/2017/03/10/chrome-deprecates-subject-cn-matching/>

[https://alexanderzeitler.com/articles/Fixing-Chrome-missing\\_subjectAltName-selfsigned-cert-openssl/](https://alexanderzeitler.com/articles/Fixing-Chrome-missing_subjectAltName-selfsigned-cert-openssl/)

# Sample SQL Queries

# MySQL queries for various tasks

The queries below are used to retrieve data from the Dashboard database. Therefore, Dashboard DB (SQL) must be selected as a Data Source when creating Datasets/Elements from Dataset/Element Editors in Metric Insights.

## 1. Upgrade all Regular Users to the Power User role

### USE CASE:

- The provided query will **upgrade all Regular Users to Power Users in selected Groups**

```
UPDATE user
SET is_power_user_ind='Y'
WHERE user_id IN
(SELECT ugm.user_id
FROM (SELECT * FROM user) AS u
JOIN user_group_member AS ugm ON (u.user_id=ugm.user_id)
WHERE (u.is_power_user_ind='N' AND u.is_administrator_ind='N') AND ugm.user_group_id
IN(<YourTargetedGroupID>));
```

### 1.1. Get the group ID for the above query

```
SELECT ugm.user_group_id, ugm.user_id, u.username
FROM user_group_member AS ugm
JOIN user AS u ON (u.user_id=ugm.user_id)
WHERE (u.is_power_user_ind='N' AND u.is_administrator_ind='N')
ORDER BY ugm.user_group_id;
```

## 2. Elements with specific Filter Values

### USE CASE:

- The provided query will retrieve **a list of External Reports that are filtered to include ONLY specific values.**

```

SELECT it.element_id, it.name AS filter_name, it.value
FROM (
    SELECT de.element_id, ef.name, efv.value
    FROM dashboard_element de
        INNER JOIN plugin_connection_profile pcg ON pcg.plugin_connection_profile_id =
de.plugin_connection_profile_id
        INNER JOIN external_report_reference err ON err.plugin_connection_profile_id =
pcg.plugin_connection_profile_id AND err.external_report_reference_id = de.
external_report_external_id
        INNER JOIN external_filter ef ON ef.external_report_reference_id = err.
external_report_reference_id
        INNER JOIN external_filter_usage efu ON efu.external_filter_id = ef.
external_filter_id AND efu.item_type = 'element' AND efu.item_id = de.element_id
        INNER JOIN external_filter_value efv ON efv.external_filter_id = ef.
external_filter_id
    WHERE de.type = 'external report'
        AND de.data_fetch_method = 'plugin'
        AND de.external_report_auto_update_image_ind = 'Y'
        AND ef.value_source = 'manual'
        AND efu.value_set = 'all'

    UNION ALL

    SELECT de.element_id, ef.name, efv.value
    FROM dashboard_element de
        INNER JOIN plugin_connection_profile pcg ON pcg.plugin_connection_profile_id =
de.plugin_connection_profile_id
        INNER JOIN external_report_reference err ON err.plugin_connection_profile_id =
pcg.plugin_connection_profile_id AND err.external_report_reference_id = de.
external_report_external_id
        INNER JOIN external_filter ef ON ef.external_report_reference_id = err.
external_report_reference_id
        INNER JOIN external_filter_usage efu ON efu.external_filter_id = ef.
external_filter_id AND efu.item_type = 'element' AND efu.item_id = de.element_id
        INNER JOIN external_filter_usage_value efuv ON efuv.external_filter_usage_id =
efu.external_filter_usage_id
        INNER JOIN external_filter_value efv ON efv.external_filter_value_id = efuv.
external_filter_value_id
    WHERE de.type = 'external report'
        AND de.data_fetch_method = 'plugin'
        AND de.external_report_auto_update_image_ind = 'Y'
        AND ef.value_source = 'manual'

```

```

        AND efu.value_set = 'selected'

UNION ALL

SELECT de.element_id, ef.name, sv.value_display_name AS value
FROM dashboard_element de
    INNER JOIN plugin_connection_profile pcp ON pcp.plugin_connection_profile_id =
de.plugin_connection_profile_id
    INNER JOIN external_report_reference err ON err.plugin_connection_profile_id =
pcp.plugin_connection_profile_id AND err.external_report_reference_id = de.
external_report_external_id
    INNER JOIN external_filter ef ON ef.external_report_reference_id = err.
external_report_reference_id
    INNER JOIN external_filter_usage efu ON efu.external_filter_id = ef.
external_filter_id AND efu.item_type = 'element' AND efu.item_id = de.element_id
    INNER JOIN segment_value sv ON sv.segment_id = ef.source_segment_id
WHERE de.type = 'external report'
    AND de.data_fetch_method = 'plugin'
    AND de.external_report_auto_update_image_ind = 'Y'
    AND ef.value_source = 'segment'
    AND efu.value_set = 'all'

UNION ALL

SELECT de.element_id, ef.name, sv.value_display_name AS value
FROM dashboard_element de
    INNER JOIN plugin_connection_profile pcp ON pcp.plugin_connection_profile_id =
de.plugin_connection_profile_id
    INNER JOIN external_report_reference err ON err.plugin_connection_profile_id =
pcp.plugin_connection_profile_id AND err.external_report_reference_id = de.
external_report_external_id
    INNER JOIN external_filter ef ON ef.external_report_reference_id = err.
external_report_reference_id
    INNER JOIN external_filter_usage efu ON efu.external_filter_id = ef.
external_filter_id AND efu.item_type = 'element' AND efu.item_id = de.element_id
    INNER JOIN external_filter_usage_value efuv ON efuv.external_filter_usage_id =
efu.external_filter_usage_id
    INNER JOIN segment_value sv ON sv.segment_value_id = efuv.segment_value_id
WHERE de.type = 'external report'
    AND de.data_fetch_method = 'plugin'
    AND de.external_report_auto_update_image_ind = 'Y'
    AND ef.value_source = 'segment'
    AND efu.value_set = 'selected'
) it
WHERE value like '%19%';

```

## 3. User stats (general)

### 3.1. Usage activity

#### 3.1.1. Hourly usage activity

##### USE CASE:

- Retrieve a **count of elements viewed per hour**

```
SELECT DATE_FORMAT(view_time, "%Y-%m-%d %H:00:00") "Date", COUNT(distinct element_id)
"Elements Viewed"
FROM dashboard_element_view_log_detail dash
WHERE view_time > :last_measurement_time
GROUP BY 1
```

#### 3.1.2. Daily usage activity

##### USE CASE:

- Retrieve a **count of elements viewed per day**

```
SELECT DATE(view_time) "Date", COUNT(distinct element_id) "Elements Viewed"
FROM dashboard_element_view_log_detail dash
WHERE view_time > :last_measurement_time
GROUP BY 1
```



## 3.2. Login stats

### 3.2.1. Last login time per User

#### USE CASE:

- Retrieve the **last login time for User (user\_id)**
- This will show the time when a User last entered their login and password
- Recommendation: use to build a Metric

```
SELECT user_id,  
IFNULL(last_login_time, 'not available') AS Last_login_date  
FROM user  
GROUP BY 1
```

### 3.2.2. List of Users since last login time

#### USE CASE:

- Retrieve a **list of Users since last login time**
- Show User information, last login time, and the count of days since that time until now
- Recommendation: use to build a Report

```
SELECT username, first_name, last_name,  
IFNULL(last_login_time, 'Not Available') AS Last_login_date,  
DATEDIFF(CURDATE(),last_login_time) AS Days_since_last_login  
FROM user
```

## 4. User Engagement (Objects and Elements)

### 4.1. Homepage elements

#### 4.1.1. Available elements on the Homepage (per User)

##### USE CASE:

- Retrieve **available elements on the Homepage per User**

```
select d.user_id as user_id, u.username as username, count(element_id) as
number_of_available_elements_on_HP
from user_dashboard_element_instance d
join user u on u.user_id=d.user_id
where in_dashboard_ind_flag = 'Y'
group by 1;
```

#### 4.1.2. Available elements on the Homepage (per User by User types)

##### USE CASE:

- Retrieve **available elements on the Homepage per User (by User type)**

```
select case is_administrator_ind and is_power_user_ind
when is_administrator_ind = 'Y' then 'Administrator'
when is_power_user_ind ='Y' then 'Power User'
else 'Regular User'
end as usertype, d.user_id as user_id,
u.username,
count(element_id) as number_of_available_elements_on_HP
from user_dashboard_element_instance d
join user u on u.user_id=d.user_id
where in_dashboard_ind_flag = 'Y'
group by 2;
```

### 4.1.3. All available elements on the Homepage (by count of all Users)

#### USE CASE:

- Retrieve the **count of all available elements on the Homepage (by count of all Users)**

```
select (select count(element_id) from user_dashboard_element_instance where  
in_dashboard_ind_flag = 'Y')/count(user_id)  
from user_dashboard_element_instance
```

### 4.1.4. Homepage search and the number of returned Tiles (per query)

#### DESCRIPTION

- Table **homepage\_search** is used to track searches performed by Users on the Homepage
- Table **homepage\_search** contains the following: homepage search id, user id, search text, the time of search and the number of Tiles returned

#### USE CASE:

- Retrieve data on **Homepage searches and the count of Tiles that were returned by search queries**

```
SELECT *  
FROM homepage_search  
ORDER by homepage_search_id DESC  
limit 100;
```

## 4.2. Favorites

### 4.2.1. List of Favorite elements

#### USE CASE:

- Retrieve a **list of Favorite elements**
- Fetch a list of Favorites for each user
- Recommendation: build a Report

```
SELECT u.user_id AS User_ID, u.username AS User_Name, fdei.element_id AS Element_ID,
de.name AS Element_Name, f.display_name AS Fovirites_Name
FROM favorite_dashboard_element_info AS fdei
JOIN dashboard_element AS de ON(de.element_id=fdei.element_id)
JOIN user AS u ON(u.user_id=fdei.user_id)
JOIN favorite AS f ON(f.favorite_id=fdei.favorite_id)
GROUP BY fdei. favorite_dashboard_element_id
ORDER BY u.username
```

## 4.3. Viewing and usage statistics

### 4.3.1. Average number of viewed Tiles (per period)

#### USE CASE:

- Retrieve the **average number of Tiles per period**

```
SELECT user_id, ifnull(count(DISTINCT
element_id)/DATEDIFF(MAX(view_time),MIN(view_time)), 'N/A') as
AVG_number_of_tiles_per_period
FROM dashboard_element_view_log_detail
GROUP BY 1;
```

## 4.3.2. Average number of viewed Tiles (per month)

### USE CASE:

- Retrieve the **average number of Tiles per month**

```
SELECT user_id, cnt/vt as a
FROM (
SELECT user_id, CONCAT(YEAR(view_time),'-',MONTH(view_time)) AS vt, count(DISTINCT
element_id) AS cnt FROM dashboard_element_view_log_detail
GROUP BY 1
) AS t
GROUP BY 1;
```

## 4.3.3. Most viewed elements per month

### USE CASE:

- Retrieve **most viewed elements per month (Top 10)**
- Displays which elements were viewed the most within the last 30 days
- Recommendation: build a Report

```
SELECT devld.element_id AS "Element ID",
de.name AS "Element Name",
count(distinct devld.user_id) AS "Number of Distinct Portal Users",
count(devld.user_id) AS "Total Number of Views"
FROM dashboard_element_view_log_detail AS devld
JOIN dashboard_element de ON (devld.element_id = de.element_id)
WHERE DATE(devld.view_time)>(curdate()- interval 1 month)
GROUP BY 1
ORDER BY 4 desc Limit 10
```

## 4.3.4. List of last viewed elements

### USE CASE:

- Retrieve the **list of last viewed elements**
- This Report will fetch data about the last viewed elements, including User information and viewing time

```
SELECT u.user_id AS User_id, u.username AS User_name, de.name AS Viewed_element_name,
dc.category AS Element_category_name,
IFNULL(devl.last_view_time, 'Not Available') AS Last_viewed_on
FROM dashboard_element_view_log AS devl
JOIN user AS u ON(u.user_id=devl.user_id)
JOIN dashboard_element AS de ON(de.element_id=devl.element_id)
JOIN dashboard_category AS dc ON(dc.category_id=de.category_id)
ORDER BY last_view_time DESC
```

## 4.3.5. Dimension and time period changes for Metrics (at last View)

### USE CASE:

- Retrieve **a Dimension and time period changes**
- This Report shows what Dimension and time period were selected by User during the last Metric View

```
SELECT u.username AS User, u.user_id AS User_ID, de.name AS Metric_Name, de.element_id
AS Metric_ID, sv.value_display_name AS Seen_Dimension_Values, uco.last_updated_time AS
View_time, REPLACE(SUBSTRING_INDEX(SUBSTRING_INDEX(uco.
overlay_state,'"interval_unit':"",-1),'}',1),'","interval_value":',' ') AS
Last_Viewed_Time_interval
FROM user_chart_overlay AS uco
JOIN user AS u ON(u.user_id=uco.user_id)
JOIN dashboard_element AS de ON(uco.element_id=de.element_id)
JOIN segment_value AS sv ON(uco.segment_value_id=sv.segment_value_id)
ORDER BY u.username, uco.last_updated_time DESC, sv.value_display_name
```

## 4.3.6. Default View type for Metrics

### USE CASE:

- Retrieve the **default View type for Metrics**
- This Report will show what type of Metric View was selected (Standart, Stoplight, Target, Projection, etc.)

```
SELECT de.element_id AS Element_id, de.name AS Element_name, devld.view_time AS
Viewed_time, u.username AS User
, SUBSTRING(uco.overlay_state,10,POSITION('"' IN REPLACE(uco.
overlay_state,'{"view": "', ''))-1) AS View_name
FROM dashboard_element AS de
JOIN dashboard_element_view_log_detail AS devld ON(devld.element_id=de.element_id)
JOIN user_chart_overlay AS uco ON(de.element_id=uco.element_id)
JOIN user AS u ON(u.user_id=devld.user_id)
WHERE uco.overlay_state LIKE '{"view": "%'
GROUP BY u.user_id
```

## 4.3.7. Most used Datasets

### USE CASE:

- Retrieve **most used Datasets (Top 10)**
- This Report will contain the most used Datasets and the count of elements sourced from them

```
SELECT de.dataset_id As Dataset_ID, d.name AS Dataset_Name, count(distinct de.
element_id) AS Number_of_element
FROM dashboard_element AS de
JOIN dataset AS d ON(de.dataset_id=d.dataset_id)
GROUP BY de.dataset_id
ORDER BY 3 Desc LIMIT 10
```

## 5. User Engagement (Notifications)

### 5.1. Favorites and Shared Folders

#### 5.1.1. All elements in favorite Folders for a Digest with enabled Notifications

##### DESCRIPTION:

- Table **favorite\_dashboard\_element\_info** contains all elements in Favorite Folders including Shared Folders.

##### USE CASE:

- Retrieve **all elements in favorite Folders for a Digest with enabled Notifications**

```
SELECT DISTINCT fdei.user_id, fdei.element_id
FROM favorite_dashboard_element_info AS fdei
JOIN favorite AS f ON (f.favorite_id=fdei.favorite_id)
JOIN user_preference AS up ON (up.user_id=fdei.user_id)
WHERE f.include_in_favorites_digest_ind='Y'
AND up.email_notification_enabled_ind = 'Y';
```

### 5.2. Alerts

#### 5.2.1. All User Alert Subscriptions for elements with enabled Notifications

##### DESCRIPTION:

- Table **user\_alert\_rule** is used for storing information about user alert subscriptions since 4.0.
- Table **alert\_rule\_element\_info** contain elements from alert rule scope with visualizations.



## USE CASE:

- Retrieve **all User Alert Subscriptions for elements with enabled Notifications**

```
SELECT DISTINCT uar.user_id, arei.visualization_element_id
FROM user_alert_rule AS uar
JOIN alert_rule_element_info AS arei ON (uar.alert_rule_id=arei.alert_rule_id AND uar.
element_id=arei.element_id)
JOIN user_preference AS up ON (up.user_id=uar.user_id)
WHERE uar.enabled_ind='Y'
AND up.email_notification_enabled_ind = 'Y';
```

## 5.3. Bursts

### 5.3.1. Burst Recipients with enabled Notifications

## DESCRIPTION:

- Table **user\_alert\_rule** is used for storing information about user alert subscriptions since 4.0.
- Table **alert\_rule\_element\_info** contain elements from alert rule scope with visualizations.

## USE CASE:

- Retrieve **Burst Recipients with enabled Notifications**

```
SELECT DISTINCT up.user_id, IFNULL(fdei.element_id,nsdi.element_id)
FROM notification_schedule_distribution AS nsd
JOIN notification_schedule_distribution_item AS nsdi ON (nsdi.
notification_schedule_distribution_id=nsd.notification_schedule_distribution_id
AND( (nsd.content_type='favorites' AND nsdi.favorite_id>0)
OR (nsd.content_type='tiles' AND nsdi.element_id IS NOT NULL)))
LEFT JOIN favorite_dashboard_element_info AS fdei ON (fdei.favorite_id=nsdi.favorite_id)
LEFT JOIN notification_schedule_distribution_group_recipient AS nsdgr ON (nsdgr.
notification_schedule_distribution_id=nsdi.notification_schedule_distribution_id)
LEFT JOIN user_group_member AS ugm ON (ugm.user_group_id=nsdgr.group_id)
LEFT JOIN notification_schedule_distribution_user_recipient AS nsdur ON (nsdur.
notification_schedule_distribution_id=nsdi.notification_schedule_distribution_id)
```

```
JOIN user_preference AS up ON (up.user_id=IFNULL(ugm.user_id,nsdur.user_id))
WHERE nsd.enabled_ind='Y'
AND up.email_notification_enabled_ind = 'Y'
AND IFNULL(fdei.element_id,nsdi.element_id) IS NOT NULL;
```

## 6. User Engagement (Notes, Annotations, Events)

### 6.1. List of User Comments

#### USE CASE:

- Retrieve the **list of User comments**
- This query will return all comments for Notes, Annotations, and Events

```
SELECT uc.element_id As ElementID, de.name AS Element_Name, de.type AS Element_Type, sv.
value_display_name As Dimension_value, uc.scope AS First_comment_type, un.text AS
First_comment, u_un.username AS First_comment_user, un.created_time As
First_comment_time, uc.text AS Second_comment, u_uc.username AS Second_comment_user, uc.
last_updated_time AS Second_comment_time
FROM user_comment AS uc
JOIN dashboard_element AS de ON(de.element_id=uc.element_id)
LEFT JOIN segment_value AS sv ON(sv.segment_value_id=uc.segment_value_id)
JOIN user_note AS un ON(uc.user_note_id=un.user_note_id)
JOIN user AS u_un ON(un.user_id=u_un.user_id)
JOIN user AS u_uc ON(uc.user_id=u_uc.user_id)
WHERE uc.scope='note'
```

UNION

```
SELECT uc.element_id, de.name, de.type, sv.value_display_name, uc.scope, ua.
annotation_text, u_ua.username, ua.annotation_time, uc.text, u_uc.username, uc.
last_updated_time
FROM user_comment AS uc
JOIN dashboard_element AS de ON(de.element_id=uc.element_id)
LEFT JOIN segment_value AS sv ON(sv.segment_value_id=uc.segment_value_id)
JOIN user_annotation AS ua ON(uc.user_annotation_id=ua.user_annotation_id)
JOIN user AS u_ua ON(ua.user_id=u_ua.user_id)
JOIN user AS u_uc ON(uc.user_id=u_uc.user_id)
WHERE uc.scope='annotation'
```

UNION

```

SELECT uc.element_id, de.name, de.type, sv.value_display_name, uc.scope, ne.name, ne.
last_updated_by, ne.last_notable_event_activity_time, uc.text, u_uc.username, uc.
last_updated_time
FROM user_comment AS uc
JOIN dashboard_element AS de ON(de.element_id=uc.element_id)
LEFT JOIN segment_value AS sv ON(sv.segment_value_id=uc.segment_value_id)
JOIN notable_event AS ne ON(uc.notable_event_id=ne.notable_event_id)
JOIN user AS u_uc ON(uc.user_id=u_uc.user_id)
WHERE uc.scope='event'

ORDER BY 2

```

## 7. System performance

### 7.1. System load

#### 7.1.1. Count of parallel processes per day for the last month

##### USE CASE:

- Retrieve the **daily count (maximum and average) of parallel processes for the last month**

```

SELECT DATE(collection_time), MAX(total_process_count), AVG(total_process_count)
FROM mysql_processlist_log
WHERE collection_time>NOW() - INTERVAL 30 DAY
GROUP BY 1;

```

#### 7.1.2. System load average per hour

##### USE CASE:

- Retrieve the **system load average value per hour**

```

SELECT DATE_FORMAT(collection_time, "%Y-%m-%d %H:00:00"),
AVG(total_process_count)

```

```
FROM mysql_processlist_log
WHERE collection_time>NOW() - INTERVAL 30 DAY
GROUP BY 1
```

### 7.1.3. Data collection issues

### 7.1.4. List of errors upon data collection

#### USE CASE:

- Retrieve the **list of errors upon data collection**
- This basic query will return all elements with errors

```
SELECT *
FROM (
    SELECT 'trigger' AS caller, element_id, segment_value_id, start_time, success_ind,
    error_message
    FROM update_trigger_event_run_log_detail
    WHERE success_ind = 'N'

    UNION ALL

    SELECT 'editor' AS caller, element_id, segment_value_id, start_time, success_ind,
    error_message
    FROM editor_data_collection_detail
    WHERE success_ind = 'N'
) it
WHERE error_message != 'No rows are returned'
```

### 7.1.5. Elements with data collection exceeding 60 minutes

#### USE CASE:

- Retrieve the **list of errors upon data collection**
- This basic query will return all elements with long-running data collection

```
SELECT *
FROM (
```

```

SELECT 'trigger' AS caller, element_id, segment_value_id, TIMESTAMPDIFF(MINUTE,
start_time, finish_time) AS _mins
FROM update_trigger_event_run_log_detail
WHERE success_ind = 'Y'
      AND TIMESTAMPDIFF(MINUTE, start_time, finish_time) >= 60
UNION ALL
SELECT 'editor' AS caller, element_id, segment_value_id, TIMESTAMPDIFF(MINUTE,
start_time, finish_time) AS _mins
FROM editor_data_collection_detail
WHERE success_ind = 'Y'
      AND TIMESTAMPDIFF(MINUTE, start_time, finish_time) >= 60
) it
ORDER BY _mins DESC

```

## 7.1.6. List of overdue Trigger runs

### USE CASE:

- Retrieve the **list of Triggers with overdue runs**
- This basic query will return a list of overdue Trigger ids, the start time and reasons for overdue runs

```

SELECT ute.update_trigger_event_id _id, ute.name, rl.run_id, rl.run_start_time, IF (rl.
event_aborted_ind = 'Y', 'Aborted', 'Timed out') _reason
FROM update_trigger_event ute
INNER JOIN update_trigger_event_run_log rl ON rl.update_trigger_event_id = ute.
update_trigger_event_id
WHERE rl.run_timed_out_ind = 'Y' OR rl.event_aborted_ind = 'Y'
ORDER BY rl.run_start_time DESC

```

# MySQL queries for Smart Folders: Usage and Engagement Stats

The queries below are used to retrieve data from the Dashboard database. Therefore **Dashboard DB (SQL)** must be selected as a Data Source when creating Datasets from Dataset Editors in Metric Insights.

## 1. Most viewed Elements with the date condition

### USE CASE:

- The provided queries will extract the Elements (Metrics, Change Reports or External Reports) that had the biggest number of views;
- The conditions used below include the 1 month time frame changed daily (from today's day last month till today) and monthly (from 1st last month till 1st current month). For more Time Intervals available please refer to the [MySQL Reference Manual](#).

### 1.1. Top 10 most viewed Elements for the latest month interval

```
SELECT element_id, segment_value_id AS dim_value, count(*) AS views
FROM dashboard_element_view_log_detail
WHERE date(view_time) >= current_date - INTERVAL 1 MONTH
GROUP BY 1,2
ORDER BY 3 DESC
LIMIT 10
```

### 1.2. Top 10 most viewed Elements in the past month

```
SELECT element_id, segment_value_id AS dim_value, count(*) AS views
FROM dashboard_element_view_log_detail
WHERE date(view_time) >= DATE_FORMAT(current_date - INTERVAL 1 MONTH, '%Y/%m/01')
AND date(view_time) < DATE_FORMAT(current_date, '%Y/%m/01')
GROUP BY 1,2
ORDER BY 3 DESC
```

LIMIT 10

## 2. Most viewed Elements by particular User Group

### USE CASE:

- The query will extract Top 10 Elements (Metrics, Change Reports or External Reports) that had the biggest number of views by Users from a particular User Group;
- Useful to monitor the statistics by various departments.

```
SELECT element_id, segment_value_id AS dim_value, count(*) AS views
FROM dashboard_element_view_log_detail AS devd
JOIN user_group_member AS ugm USING(user_id)
WHERE ugm.user_group_id IN (
    SELECT group_id
    FROM user_group
    WHERE name = 'UserGroupName'
)
GROUP BY 1,2
ORDER BY 3 DESC
LIMIT 10
```

**i** `UserGroupName` in the query above is to be replaced with the desired User Group Name that exists in MI app to filter by. If there are several User Groups you need to include in the filter, change the condition for `WHERE name in ('UserGroupName1', 'UserGroupName2', 'UserGroupName3')` and define User Group Names in quotes.


## 3. Elements with Alerts created today or not resolved yet

### USE CASE:

- To keep track of the Metrics with Alerts created today (regardless of the status) and the Metrics with Alerts not resolved yet (regardless of the status);

- The script will extract all the Metrics that satisfy the conditions, either the user that runs the Dataset Update Data is subscribed to the Alerts or not.

```
SELECT de.element_id AS element_id,
svi.segment_value_id AS dim_value
FROM alert_event AS ae
JOIN dashboard_element AS de USING(element_id)
JOIN segment_value_info AS svi USING(segment_value_id)
JOIN alert_rule AS ar USING(alert_rule_id)
JOIN issue AS i USING(issue_id)
WHERE (ae.alert_log_time >= :measurement_time OR i.is_resolved_ind = 'N')
AND de.type = 'metric'
AND ar.assign_type = 'element'
AND ar.alert_rule_type = 'user'
AND ar.workflow_id > 0
GROUP BY de.element_id, svi.segment_value_id
```

 Attention to the **:measurement\_time** variable - the value is set in the **Set "Data For" Date** parameter on the Dataset edit page.

If the Smart Folder has to display Metrics with Alerts created the same day when the Dataset Update Data runs, it's supposed to be set for **Today**. If set for **Yesterday**, the comparison sign to be changed for **>** otherwise the query will extract the Metrics with Alerts created Yesterday and Today.

## 4. Elements with not resolved Alerts in defined statuses

### USE CASE:

- To keep track of the Metrics with Alerts in the specific statuses (to be modified directly in the query);
- The script will extract all the Metrics that satisfy the conditions, either the user that runs the Dataset Update Data is subscribed to the Alerts or not.

```
SELECT de.element_id AS element_id,
svi.segment_value_id AS dim_value
```



```
FROM alert_event AS ae
JOIN dashboard_element AS de USING(element_id)
JOIN segment_value_info AS svi USING(segment_value_id)
JOIN alert_rule AS ar USING(alert_rule_id)
JOIN issue AS i USING(issue_id)
LEFT JOIN workflow_status AS ws ON (ws.workflow_status_id = i.workflow_status_id)
WHERE ws.name in ('In Progress', 'In Review', 'Validation')
AND de.type = 'metric'
AND ar.assign_type = 'element'
AND ar.alert_rule_type = 'user'
AND ar.workflow_id > 0
GROUP BY de.element_id, svi.segment_value_id
```