# Quarry Queries

**Abstract:**

This document provides detailed descriptions of three queries which were formulated by users on the Quarry query platform Panda, Y. Quarry. Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/. The descriptions will help us better understand the working of the queries and the possible intent behind formulating them.

## Query 1: Most Active New Users

**Query URL:**

WikedKentaur. Most active new users. Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/query/6894

**Database:**

etwiki_p: Estonian Wiki

**Query Author:**

WikedKentaur

**Query Code:**

```
USE etwiki_p;
SELECT user_name, user_registration, user_editcount
    FROM user
    WHERE user_registration > DATE_FORMAT(DATE_SUB(NOW(),INTERVAL 1 DAY),'20150101000000')
    AND user_editcount > 10
    AND user_id NOT IN (SELECT ug_user FROM lvwiki_p.user_groups WHERE ug_group = 'bot')
    AND user_name not in (SELECT REPLACE(log_title,"_"," ") from logging
        where log_type = "block" and log_action = "block"
        and log_timestamp >  DATE_FORMAT(DATE_SUB(NOW(),INTERVAL 2 DAY),'20150101000000'))
   order by user_editcount desc
```

**Query Output:**

| user_name | user_registration | user_editcount |
|---|---|---|
| Valga raamatukogu | 20150715072613 | 412 |
| Dewdrop12 | 20150505083359 | 383 |
| Astromaailm | 20150318172228 | 358 |
| Jüri Eintalu | 20150601220340 | 279 |

**Query Description:**

As the name describes, the query aims to find the most active new users. The query involves a single table 'user'. We consider 'user_editcount' as a measure of activity of the users.

The query obtains three attributes 'user_name','user_regidtration','user_editcount' from the table which satisfy the following where clauses:

i) User registration date is greater than 01/01/2015 00:00:00 and less than current day i.e. Yesterday

ii) The user_id is not of that of a bot

iii) The user is not blocked for the given time period

iv) The user has made a minimum of 10 edits

The expected result set is order in descending order of 'user_editcount' so that we can get the most active user at the top of the result. The result set consists of 304 tuples and each tuple has above mentioned 3 attributes.

This query can help us identify as already mentioned the most active users in given wiki. With modifications we can identify the most active user in a given time range (year/month/week). We can also modify it to identify the users who have made the least edits, and we can probably consider such users to marked inactive.

## Query 2: Find top 200 most-used and undocumented templates on Commons

**Query URL:**

Jarekt. Find top 200 most-used and undocumented templates on commons. Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/query/6313

**Database:**

commonswiki_p: Wikimedia Commons

**Query Author:**

Jarekt

**Query Code:**

```
USE commonswiki_p;
select DISTINCT concat("Template:",page_title) as pagename, count(*) as tot
from page
join templatelinks on page_title = tl_title and tl_namespace = 10
where page_title not like "%/%"  -- no subtemplates
```

```
AND page_namespace = 10          -- count templates
AND page_is_redirect = 0         -- no redirects
AND NOT exists (
    SELECT *
    FROM templatelinks
    WHERE tl_from=page_id
    AND tl_namespace = 10
    AND tl_title ="TemplateBox" -- find files that do not transclude a [[template:TemplateBox]]
    limit 1
)
and not exists ( -- file not already in "Media_missing_infobox_template" or subcategories
    select *
    from categorylinks
    where
      cl_from=page_id and
      cl_to = "Empty_tag_templates"
    limit 1
  )
group by page_id
order by tot desc, page_title
limit 200
```

**Query Output:**

| pagename | tot |
|---|---|
| Template:Parse_source | 27598404 |
| Template:Cc-by-sa-layout | 16904232 |
| Template:Edit | 11970116 |
| Template:IsNum | 7234455 |
| Template:Tlp | 5083499 |
| Template:License_migration_is_redundant_multiple | 3874868 |
| Template:Str_left | 3746090 |

**Query Description:**

The query uses the Commons Wikimedia, which is a media library multilingual content (images, sounds and videos) for educational purposes in the public domain or released under a free license. We use three tables here, ‘page’; which is the core of the wiki, and is identified by an id and by the title, ‘templatelinks’; each page contains various links and these are identified by using the id of the host page, also each template is classified into a namespace and has an associated title and finally ‘categorylinks’; where each page is defined as a category member.

The query groups all the pages by their unique ID and then counts all the templates that are linked within the given page by comparing the template title and the page title. To do so we have an INNER JOIN between “page” and “templatelinks” on “page_title” and “tl_title”. We validate the result set by making sure that the page is not a sub template, belongs to the namespace 10 i.e. {{ template namespace }} and that it is not a redirect. We also check that the specific template ‘TemplateBox’ is not included in the result set and also that the page doesn’t belong to category/subcategory of empty tag templates. We order the result set by the columns alias ‘tot’ which is a count of all the repeating templates and we restrict the result set to the top 200.

## Query 3: Orphaned fair use images

**Query URL:**

B. Orphaned fair use images (en). Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/query/3268

**Database:**

enwii_p: English Wikimedia

**Query Author:**

B

**Query Code:**

```
USE enwiki_p;
SELECT CONCAT('*{{lf|', REPLACE(REPLACE(p.page_title, '"', '**DOUBLEQUOTE**'), '_', ' '), '}}') FROM page p
INNER JOIN categorylinks c1 ON p.page_id = c1.cl_from
LEFT JOIN imagelinks i ON p.page_title = i.il_to AND (i.il_from_namespace = 0)
LEFT JOIN categorylinks c2 ON p.page_id = c2.cl_from AND c2.cl_to = "All_orphaned_non-free_use_Wikipedia_files"
WHERE c1.cl_to = "All_non-free_media"
AND i.il_from IS NULL
AND c2.cl_from IS NULL
AND p.page_namespace = 6
```

**Query Output:**

| CONCAT('*{{lf\|', REPLACE(REPLACE(p.page_title, '"', '**DOUBLEQUOTE**'), '_', ' '), '}}') |
|---|
| *{{lf\|~ZONK~.jpg}} |
| *{{lf\|1. FC Trogen.jpg}} |
| *{{lf\|1904 Illinois Fighting Illini football team.jpg}} |
| *{{lf\|2005NCAAVBLOGO.jpg}} |
| *{{lf\|A recent group picture of Crayon Pop (2016).jpg}} |

**Query Description:**

The query aims to identify the images that have been orphaned i.e. no pages actively link to them. This is extremely useful because at periodic intervals once can identify such orphaned images and clear them from the system. This helps in saving memory and in optimizing storage management. Also all these images belong to the category of 'non_free_media', which are basically images which have the non_free_media copyright tags.

The query progresses in a complex manner, with joins between 'page','categorylinks' and 'imagelinks'. We first comclude a join INNER JOIN between 'page' and 'categorylinks' on page['page_id'] and category['cl_from']. Then we perform a LEFT OUTER JOIN of the obtained result set with 'imagelinks' on the basis of page['page_id'] and imagelinks['il_to'] and filtering all images such that they belong to the namespace 0 i.e. {{ gallery namespace}}. Then we check that the page belongs to the category 'All_orphaned_non-free_use_Wikipedia_files' with one more LEFT OUTER JOIN on page['page_id'] and categorylinks['cl_from'].

After the result set is so obtained, we need to find the images that are currently not being used and are not belonging to any category, hence where clause includes them. We see that the imagelinks['il_from'] is NULL, that means the image is orphaned. We also filter on the namespace 6 i.e. {{file namespace}} were all the images and media is stored.

Word Count: 801