**Quarry Queries**

**Abstract:**

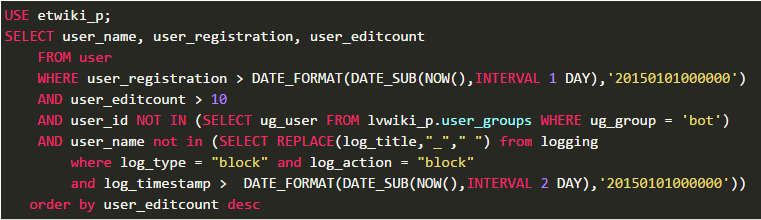
This document provides detailed descriptions of three queries which were formulated by users on the Quarry query platform Panda, Y. Quarry. Retrieved October 19, 2016, from Quarry, <https://quarry.wmflabs.org/>. The descriptions will help us better understand the working of the queries and the possible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

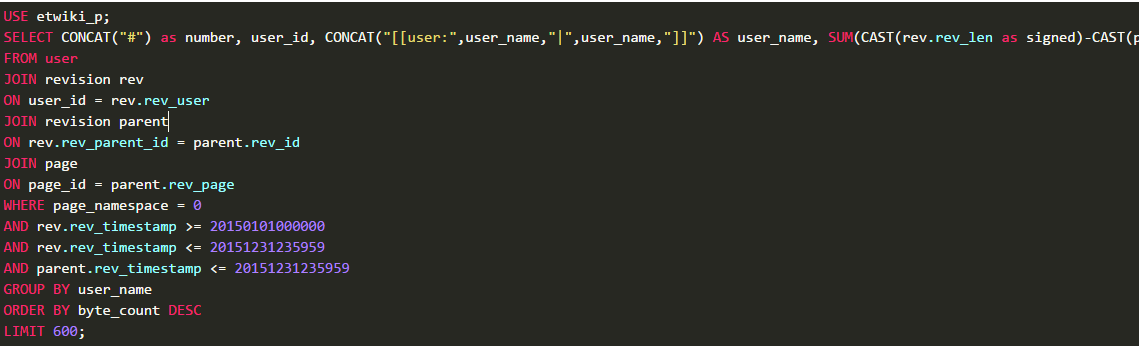
This query can help us identify as already mentioned the most active users in given wiki. With modifications we can identify the most active user in a given time range (year/month/week). We can also modify it to identify the users who have made the least edits, and we can probably consider such users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

**Query 2: Most actice users by bytes in 2015**

**Query URL:**WikedKentaur. Most active new users. Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/query/6896

**Database:**   
etwiki\_p: Estonian Wiki

**Query Author:**  
WikedKentaur

**Query Code:**

**Query Description:**

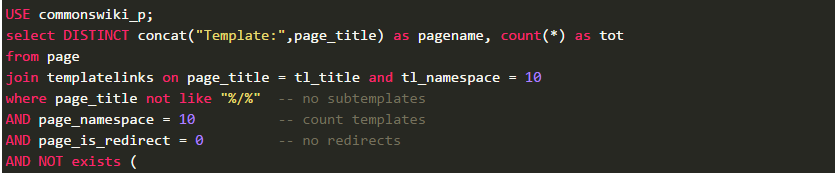
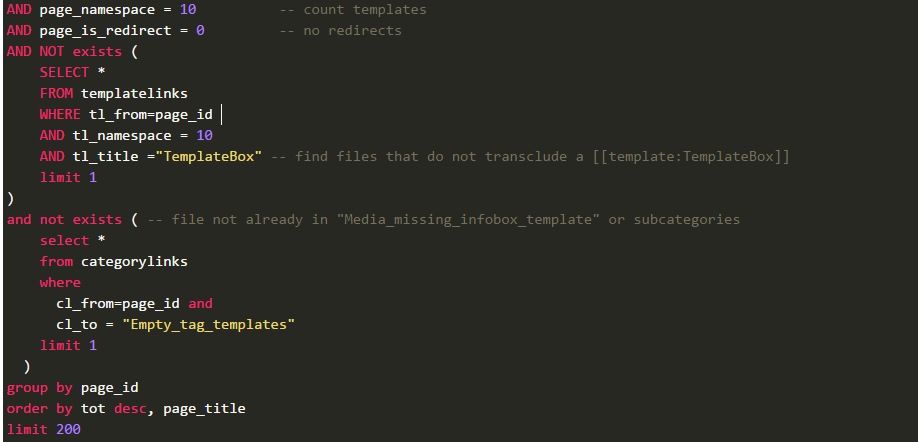
The earlier query counted the most active users on the basis of the number of edits they had made. Now we will use a different measure to calculate the number of active users. This time we consider the number of bytes to be the measure of a user activity. The user with the highest number of bytes cumulated over all the edits made since 01/01/2015 00:00:00 will be the most active user. We use an additional table known as revision which stores information about the revisions of the queries as users edit them. We do a self-join on the revision table to compare the current revision with its previous revision and find the difference in the byte length to calculate the number of bytes the user has added.

**Query 3: 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:**

**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.

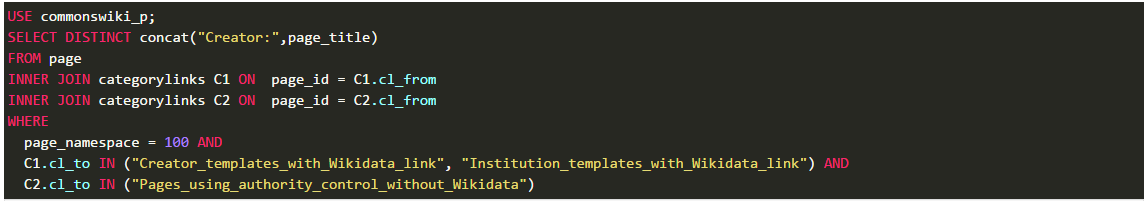
The query groups all the pages by their unique ID and then counts all the templates that are linked within the given page by comparing the template title and the page title. To do so we have an INNER JOIN between “page” and “templatelinks” on “page\_title” and “tl\_title”. We validate the result set 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 4: Creator templates with Wikidata link using authority control without Wikidata**

**Query URL:**Jarekt. Find top 200 most-used and undocumented templates on commons. Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/query/9894

**Database:**commonswiki\_p: Wikimedia Commons

**Query Author:**Jarekt

**Query Code:**

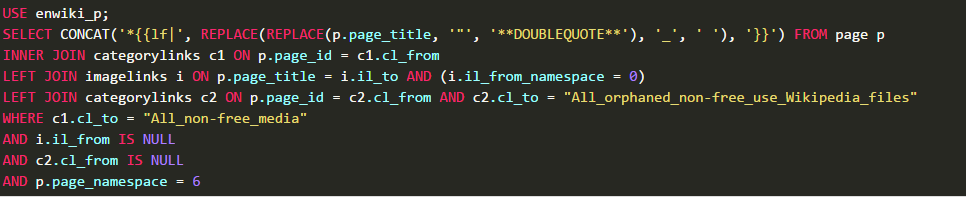
**Query Description:**This query introduces a join between page and category links and then a self-join of category link with itself based on the column category[‘cl\_from’]. The query introduces the concept of self-join and filtering using different where clauses on both the tables. We can also perform self joins on other columns within the same table.

**Query 5: 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:**enwiki\_p: English Wikimedia

**Query Author:**  
B

**Query Code:**

**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’].

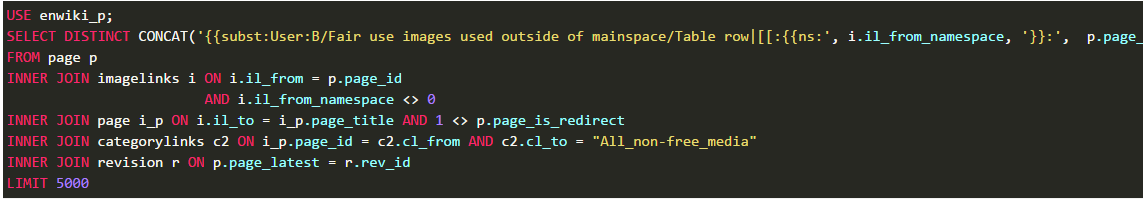
After the result set is so obtained, we need to find the images that are currently not being used and are not belonging to any category, hence where clause includes them. We see that the imagelinks[‘il\_from’] is NULL, that means the image is orphaned. We also filter on the namespace 6 i.e. {{file namespace}}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

**Query 6: Fair use images used outside of mainspace**

**Query URL:**B. Orphaned fair use images (en). Retrieved October 19, 2016, from Quarry, https://quarry.wmflabs.org/query/2900

**Database:**enwiki\_p: English Wikimedia

**Query Author:**  
B

**Query Code:**

**Query Description:**Here we are interested in obtaining images which to do not belong to the namespace 0 {{ gallery namespace }}. We concatenate the page, the namespace, the file and the image name after forming a join between the various images, pages and categories schemas as we have already done before in the above query. We filter out the namespace 0 using <> which stands for not equal to and we also make sure that these images are belonging to the category “All\_non-free\_media” which has been explained as images with the no free copyright tag. We get the images from the latest revision of the pages and also make sure that the page are not re directs.

Word Count: 1204