**Query 1:**

1. <https://quarry.wmflabs.org/query/22130>
2. <https://quarry.wmflabs.org/query/22129>
3. <https://quarry.wmflabs.org/query/22125>

In the series above the user ‘Mxn’ is querying daily English article moves, redirects, and. He is using CONCAT to string the date of each move together. He is using the ‘logging’ schema to find the log type and log action. Then he combines the values into the same column called moves which is arranged in descending order. This query is quite simple but he goes on to use the same basic structure to find redirects created and templates tagged with {{coord}}. The namespace value was confusing to me at first but once I realized the values were defined the string made more sense.

**Query 2:**

1. <https://quarry.wmflabs.org/query/15830>
2. <https://quarry.wmflabs.org/query/20378>

In the first query the user ‘Zhuyifei1999’ searches for a specific log\_id. This query returns information related to the next query and a user who is listed in the second query. I was able to see which values he was using to find information about users who are blocked from WP0. This query is complicated and I only have a basic understanding of SQL. With that, I had spent a lot of time drilling down the individual statements and still only have only a small understanding of what is going on. But, I will record what I found.

Zhuyifei1999 is looking for users who have been blocked from Wikipedia Zero. Under the logging schema he is searching for blocked users and returning a regular expression value for log\_comment in the table below. I cannot see exactly what the regular expression is looking for but it returns the type of abuse in a value. He also goes on to exclude information that is listed in two subqueries.

**Query 3:**

1. <https://quarry.wmflabs.org/query/21386>
2. <https://quarry.wmflabs.org/query/21397>
3. <https://quarry.wmflabs.org/query/21201>

In this query user ‘SoWhy’ is looking to find non-EC users commenting on a certain page. He uses select distinct to only return different values. He then renames each column header he is interested in; each is taken from the user schema. He combines similar fields by using LEFT JOIN. SoWhy includes the page title he initially started the query for (https://goo.gl/ms8FCv) and returns the results in ascending order of when the users registered. The important piece of the first query is that he searches for users whose ug\_group IS NULL. He is looking to list the users who are not assigned to a particular group.

In his next query he is searching for all users commenting on the same page as above. But, this time he is trying to find all user\_groups with all different types of permissions. The user Jclemens stood out to him for some reason. In his last query he looks for specific user info to obtain the user\_id.