Find Archived Pages

Archive note: all links and images have been disabled as their destinations and/or source locations no longer exist. Content marked with "[...]" have been removed or simplified.

Created by Brian Immel on June 4, 2022

- Overview
- Find archived pages
- Filters
- Variables

Overview

This document shows entry-level programmers how to set up a SQL query using the [table] table to generate a list of wiki pages that have the status of "archived" and sort them by their date stamp and page title. This tutorial also shows you how to exclude other document types as listed in our Wiki Page Types. Additionally, this tutorial shows you how to apply a variable in place of hard-coded values in the query so that you can re-use this query in other places. This query is part of a Learning SQL series designed to help you get started with SQL and create your own dashboard filled with queries to report on your wiki's health.

This tutorial assumes you are familiar with using the [Query UI] to write and test the query used in this document and are using the [wiki_database] database set in your [Query UI]. See our documentation for how to use [Query UI] to learn the basics of this interface.

Find archived pages

This query is designed to help you find wiki documents that have been archived.

Copy and paste this code snippet into your [Query UI] and click the **Submit** button.

```
SELECT
     ds, page_title
FROM [table]
WHERE
     page_title LIKE '%[wiki_root]%'
     AND archive_time > 1
GROUP BY
     ds, page_title
```

What is this query doing? First off, it selects the date stamp (`ds`) and page title (`page_title`) from the [table] so that the query can find all pages that matches the wiki root. Then, it filters for any document who's page archive status is greater than zero (0). The [database] marks the date a wiki page has been archived and this is helpful for us to find any wiki document that has an archive status of one (1) or greater. Finally, the query groups the results by their date stamp and page title to make it easier to sort and organize wiki pages accordingly.

Filters

You can modify this query by appending the following filters in the `WHERE` clause:

- `redirected = 0` (binary) filters for pages that have not been redirected.
- `deletion_time = 0` (date) filters for pages that have not been deleted.
- `page_is_hidden = 0` (binary) filters pages that do not have the hidden status. If this value is set to `1`, the page is not visible to normal users (but still visible to wiki administrators.)
- `ds = '2022-06-04'` to limit the results to pages only archived on a specific date.

If you choose to use all these filters, your query should look like this:

```
SELECT
     ds, page_title
FROM [table]
WHERE
     page_title LIKE '%[wiki_root]%'
     AND archive_time > 1
     AND redirected = 0
     AND deletion_time = 0
     AND page_is_hidden = 0
     AND ds = '2022-06-04'
GROUP BY
     ds, page_title
```

Now your query will only find wiki root pages that have been archived on June 4th, 2022 and exclude any document that has been archived, deleted, or has the status of hidden.

Variables

If you read the document called Using Query Dashboard, you should be familiar with the notion of using variables. Using variables with queries allows you to reuse a query in multiple places (other dashboards, program queries, and documentation automation widgets) by passing values to the query instead of having to copy and rewrite a query to just change a few values.

To create a variable in [Query UI], follow these steps:

1. Open the **Variable panel** in the right-hand panel.
2. Click the **Add New Variable** button and select the appropriate variable type (string, date, number, other). In this case, you want to select the **string** type.
3. Provide a name in the **Variable Name** and the variable's **default value**. In this case, you want to enter `wiki_root` for both the variable name and variable value.
4. Click the **Save** button.
5. For this tutorial, replace the hard-coded value of [wiki_root] with `$wiki_root$`. When this query is used with a dashboard and connected with an input field, the value of the input field will pass to the query and return metrics for that user selected wiki root. If the user doesn't supply a wiki root in the input field, the default value will be displayed and the query uses that default value.

Using this variable: replace `page_title LIKE '%[wiki_root]%'` with `page_title LIKE '%$wiki_root$%'` as shown in this code snippet:

```
SELECT
     ds, page_title
FROM [table]
WHERE
     page_title LIKE '%$wiki_root$%'
     AND archive_time > 1
GROUP BY
     ds, page_title
```

It should be noted that if you use a string instead of a date for a date value, the query will read the "date string" as a string and not a number. The results from this query would yield bad data as the date won't be valid.