Continue

Looking to extract the MONTH NAME from a DATE in BigQuery, where the type is DATE (i.e., 2019-09-19). Tried using PARSE_DATE and other functions but didn't get the desired result. Noted that some SQL languages have MONTHNAME() function, however it's not available in BQ. Is there a way to achieve this without relying on CASE function? Given article text here SELECT DATE(created_at) AS date, EXTRACT(MONTH FROM created_at) AS month_num, DATE_TRUNC(created_at, MONTH) AS month_start, FORMAT_DATE(‘%Y-%m‘, created_at) AS month_str FROM `bigquery-public-data.github_repos.sample_commits` LIMIT 5 Query Results: Row date month_num month_start month_str 1 2017-03-30 3 2017-03-01 2017-03 2 2017-03-30 3 2017-03-01 2017-03 3 2017-03-29 3 2017-03-01 2017-03 4 2017-03-29 3 2017-03-01 2017-03 5 2017-03-29 3 2017-03-01 2017-03 To calculate month-over-month changes in commits, we first aggregate by repository and month using the following query: WHERE created_at BETWEEN ‘2017-01-01‘ AND ‘2017-12-31‘ GROUP BY 1, 2 ORDER BY 1, 3 DESC This breakdown of top repositories by commit count for each month can be further analyzed. We then use window functions like LAG() to calculate the absolute and percentage changes between months. The query calculates the total commits per month in a common table expression (CTE) and uses LAG(commits) OVER (ORDER BY month) to access data from previous rows, allowing us to calculate the absolute change (commits - LAG(commits)) and percentage change ((commits - LAG(commits)) / LAG(commits)). This produces a result set showing how commits changed month-over-month, both in absolute and percentage terms. To optimize monthly aggregation queries in BigQuery, filtering is key. Use the WHERE clause to limit data scanned and improve performance. For example, using 'WHERE created_at BETWEEN ‘2017-01-01‘ AND ‘2017-12-31‘' limits a query to a specific year. Partitioned tables can also be beneficial for large datasets by dividing them into smaller segments based on date. Queries that filter on the partitioning column only scan relevant partitions, reducing data scanned. Consider materializing complex queries into separate views or tables if running the same aggregations repeatedly. BigQuery allows easy creation of views from SELECT queries and provides a query validator to check for syntax errors and estimated cost. Review the query explanation after running to understand which stages were most expensive. Cost controls should also be set up on your BigQuery project to avoid unexpected charges, such as quotas at the project or user level to limit data queried per day or month. Extracting Month from Timestamp in BigQuery == BigQuery provides an efficient way to extract information from timestamps using the `get_month_from_timestamp()` function. This article will demonstrate how to use this function to retrieve the month from a given timestamp. Timestamp Formats Supported by BigQuery --------------------------------------- Before diving into the tutorial, it is essential to understand the different timestamp formats supported by BigQuery: * `YYYY-MM-DD` * `YYYYMMDD` * `YYYY-MM-DDTHH:MM:SS.SSSZ` Using the `DATE_TRUNC` Function ---------------------------------- To extract the month from a timestamp in BigQuery, we can utilize the `DATE_TRUNC` function. Here's an example: ```sql SELECT get_month_from_timestamp(TIMESTAMP '2023-03-08') AS month ``` This query will return the output as `3`, representing March. Examples of Using the `get_month_from_timestamp()` Function --- Here are some additional examples demonstrating how to use this function with different timestamp formats: ### Example 1: YYYY-MM-DD ```sql SELECT get_month_from_timestamp(TIMESTAMP '2023-03-08') AS month ``` Output: `3` ### Example 2: YYYYMMDD ```sql SELECT get_month_from_timestamp(TIMESTAMP '20230308') AS month ``` Output: `3` ### Example 3: YYYY-MM-DDTHH:MM:SS.SSSZ ```sql SELECT get_month_from_timestamp(TIMESTAMP '2023-03-08T12:00:00.000Z') AS month ``` Output: `3` Conclusion --------- The BigQuery `get_month_from_timestamp()` function is a valuable tool for extracting the month from a timestamp. By utilizing this function, you can efficiently filter data by month, group data by month, and calculate monthly statistics. Whether working with dates in the format `YYYY-MM-DD`, `YYYYMMDD`, or `YYYY-MM-DDTHH:MM:SS.SSSZ`, this function provides an easy-to-use solution for extracting meaningful insights from your data. BigQuery's get month from timestamp function extracts the month from a given timestamp, returning an integer value representing the month (1-12). The function only works with UTC timestamps. It also doesn't support timestamps with fractional seconds; these must be rounded to the nearest second. The following query demonstrates how to use this function: SELECT get_month(TIMESTAMP '2023-01-01') This query returns 1, representing January. The function can filter data by month (e.g., selecting all rows from a table where the timestamp is in January) or aggregate data by month (e.g., calculating total rows per month). Usage notes include converting non-UTC timestamps to UTC and rounding timestamps with fractional seconds. Limitations of the function include only supporting UTC timestamps and not supporting timestamps with fractional seconds. Given article text here Looking at how to extract various components from a timestamp in BigQuery, let's consider the day of the month. To get this information, one can utilize the following SQL query: sql SELECT EXTRACT(DAY FROM timestamp_column) AS day FROM table_name This results in a new column called `day` that contains the day of the month for each row in the `timestamp_column` column. For obtaining the hour from a timestamp, the SQL query is as follows: sql SELECT EXTRACT(HOUR FROM timestamp_column) AS hour FROM table_name This yields a new column called `hour` that contains the hour of the day for each row in the `timestamp_column` column. Similarly, to obtain the minute and second from a timestamp, one can use the following SQL queries: sql SELECT EXTRACT(MINUTE FROM timestamp_column) AS minute FROM table_name sql SELECT EXTRACT(SECOND FROM timestamp_column) AS second FROM table_name In this blog post, we previously discussed how to extract the month from a timestamp in BigQuery. We first introduced the concept of timestamps and their different formats in BigQuery. Then, we demonstrated how to use the `EXTRACT` function to extract the month from a timestamp. Moving on, Hatch has undergone significant changes since its establishment in 2011 by Marcus Greenwood. Initially designed to seamlessly merge content management with social networking, Hatch now focuses on unraveling mysteries and answering a myriad of questions. This shift is part of an effort to expand the scope of topics covered by Hatch, delving into the unknown and the unexplored. Recently, I started working on a time series forecasting project this morning. The first step in this project involved gathering the data I had to run a straightforward SQL query in Google BigQuery. While looking up how to extract the year from a timestamp, I thought it would be useful to document this for future reference. This might also benefit others who encounter similar issues. Here's a list of available date parts: DAYOFWEEK (1-7), DAY (0-365), WEEK (0-53, starts on Sunday), WEEK(), ISOWEEK (ISO 8601 week number, Monday start), MONTH, QUARTER, YEAR (ISO 8601 year). You can extract a date part using EXTRACT(part FROM date_expression). For example: EXTRACT(YEAR FROM 2019-04-01) returns 2019. To construct a date from integers, use DATE(year, month, day). Adding and subtracting dates can be done with DATE_ADD(date_expression, INTERVAL INT64_expr date_part) or DATE_SUB(date_expression, INTERVAL INT64_expr date_part). A practical application of this is filtering dynamic dates: where my_date between DATE_SUB(current_date, INTERVAL 7 DAY) and DATE_SUB(current_date, INTERVAL 1 DAY). The difference between two dates can be calculated.

Month bigquery. Bigquery get month from date. Bigquery extract month from date. Month gbq. Bigquery extract month name from date. Bigquery month from datetime. Bigquery get first day of month from date. Bigquery subtract 1 month from date. Bigquery extract month and year from date.