Creating Tables with Date Partitions

A partitioned table is a special table that is divided into segments, called partitions, that make it easier to manage and query your data. By dividing a large table into smaller partitions, you can improve query performance, and you can control costs by reducing the number of bytes read by a query. Let's create a new table and bind a date or timestamp column as a partition.

View data processed from non-partitioned table

Click **Compose Query**
Copy and Paste the below query

```
#standardSQL
SELECT DISTINCT
  fullVisitorId,
  date
FROM `data-to-insights.ecommerce.all_sessions_raw`
WHERE date = '20180708'
LIMIT 5
```

Use the **Query Validator** to determine how much data this query will process
Answer: 635 MB (and we only selected two columns and returned 5 rows).

Recall that the query engine needs to scan all records in the dataset to see if they satisfy the date matching condition in the WHERE clause. Additionally, the LIMIT 5 does not reduce the total amount of data processed and is a common misconception.

Create a new Partitioned Table based on date

Let's instead bucket our existing data into individual partitions, one for each day, and compare how much data is processed.

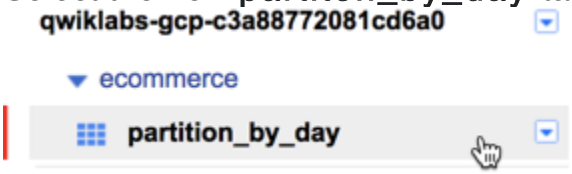
Copy and Paste the below query
Click **Run Query**

```
#standardSQL
CREATE OR REPLACE TABLE ecommerce.partition_by_day
PARTITION BY date_formatted
OPTIONS(
  description="a table partitioned by date"
) AS
SELECT DISTINCT
PARSE_DATE("%Y%m%d", date) AS date_formatted,
fullvisitorId
FROM `data-to-insights.ecommerce.all_sessions_raw`
```

Note the new option when creating the table as a result of the query to PARTITION BY a field. The two options available to partition are DATE and

TIMESTAMP. We apply a PARSE_DATE function on our date field (stored as a string) to get it into the proper DATE type for partitioning.

Select the new **partiton_by_day** table in your dataset



Select **Details**
Confirm the below:

- Partitioned by: day
- Partitioning Field: date_formatted

View data processed with a Partitioned Table

Finally, copy and paste the below query and view the total bytes processed

```
#standardSQL
SELECT *
FROM ecommerce.partition_by_day
WHERE date_formatted = '2018-07-08'
```

Answer: This query will process 0 B when run.
Why is there a difference after adding the table partitioning by day? The query engine knows which partitions already exist and knows no partition exists for 2018-07-08 (the ecommerce dataset ranges from 2016-08-01 to 2017-08-01).

Creating an auto-expiring partitioned table

Explore the available NOAA weather data tables

Click to open the NOAA Daily Weather BigQuery Public Dataset
Scroll through the tables in the noaa_gsod dataset which are manually sharded and not partitioned
Our goal is to create a table that:

- Queries on current year weather data
- Filters to only include days that have had some precipitation (rain, snow, etc.)
- Only stores each partition of data for 90 days from that partition's date (rolling window)

First, **copy and paste** this below query:

```
#standardSQL
SELECT
  DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date,
  (SELECT ANY_VALUE(name) FROM `bigquery-public-data.noaa_gsod.stations` AS stations
   WHERE stations.usaf = stn) AS station_name,  -- Stations may have multiple names
  prcp
FROM `bigquery-public-data.noaa_gsod.gsod*` AS weather
WHERE prcp < 99.9  -- Filter unknown values
  AND prcp > 0      -- Filter stations/days with no precipitation
  AND _TABLE_SUFFIX = CAST( EXTRACT(YEAR FROM CURRENT_DATE()) AS STRING)
LIMIT 100
```

Note the table wildcard * used in the FROM clause to limit the amount of tables referred to in the _TABLE_SUFFIX filter which passed an expression to find the current year and convert it to a string.

Note that although we added a LIMIT 100, this still does not reduce the total amount of data scanned (about 1.83 GB) since there are no partitions yet.

Click **Run Query**
Confirm the date is properly formatted and the precipitation field is showing non-zero values.

Your turn: Create a Partitioned Table

Modify the previous query to create a table with the below specifications:

- Table name ecommerce.days_with_rain
- Use the date field as your PARTITION BY
- For OPTIONS, specify partition_expiration_days = 90
- Add the table description = "weather stations with precipitation, partitioned by day"

Possible Solution:

```
#standardSQL
CREATE OR REPLACE TABLE ecommerce.days_with_rain
PARTITION BY date
OPTIONS (
  partition_expiration_days=90,
  description="weather stations with precipitation, partitioned by day"
) AS
SELECT
  DATE(CAST(year AS INT64), CAST(mo AS INT64), CAST(da AS INT64)) AS date,
  (SELECT ANY_VALUE(name) FROM `bigquery-public-data.noaa_gsod.stations` AS stations
   WHERE stations.usaf = stn) AS station_name,  -- Stations may have multiple names
  prcp
FROM `bigquery-public-data.noaa_gsod.gsod*` AS weather
```

```
 WHERE prcp < 99.9  -- Filter unknown values
   AND prcp > 0      -- Filter stations/days with no precipitation
   AND _TABLE_SUFFIX = CAST( EXTRACT(YEAR FROM CURRENT_DATE()) AS STRING)
```

Confirm data partition expiration is working

To confirm we are only storing data from 90 days in the past up until today we can run DATE_DIFF query to get the age of our partitions which are set to expire after 90 days.

Below is a query which tracks the average rainfall for the NOAA weather station in Wakayama, Japan which has significant precipitation.

Copy and Paste the below query

```
#standardSQL
# avg monthly precipitation
SELECT
  AVG(prcp) AS average,
  station_name,
  date,
  DATE_DIFF(CURRENT_DATE(), date, DAY) AS partition_age,
EXTRACT(MONTH FROM date) AS month
FROM ecommerce.days_with_rain
WHERE station_name = 'WAKAYAMA' #Japan
GROUP BY station_name, date, month, partition_age
ORDER BY date;
```

Click **Run Query**

Confirm the oldest partition_age is below is at or below 90 days

Note: Your results will vary if you re-run the query in the future as the weather data, and your partitions, are continuously updated.