**BigQuery - Partition and Cluster**

BigQuery organizes data tables into units called datasets. These datasets are scoped to GCP project. These multiple scopes—project, dataset, and table—helps you structure your information logically.

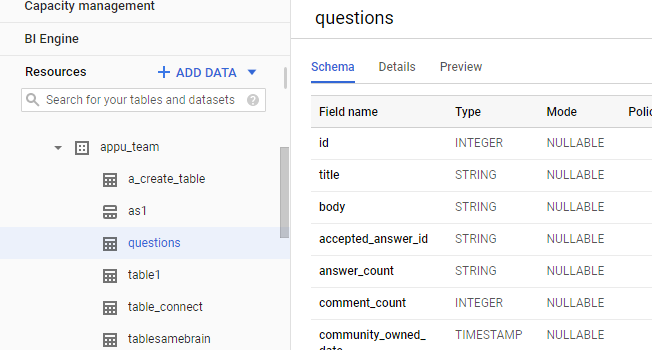
Optimizing cost: Keeping *data in BigQuery* instead of creating a separate storage place in Cloud storage and to use BigQuery’s table *partitioning and clustering* features is a best practice if we are looking to optimize both cost and performance.

**Partitioning**

**Why is Partitioning necessary in BigQuery?**

**BigQuery** is a tool to analyse huge amount amounts of data. But there are some cases where we want to analyze a subset of data. For example, if you want to know data over a particular period of time or over a particular column

* **Create a table from an existing table(** Copy table from Public dataset: stackoverflow>posts\_answers)

****

SELECT

  \*

FROM

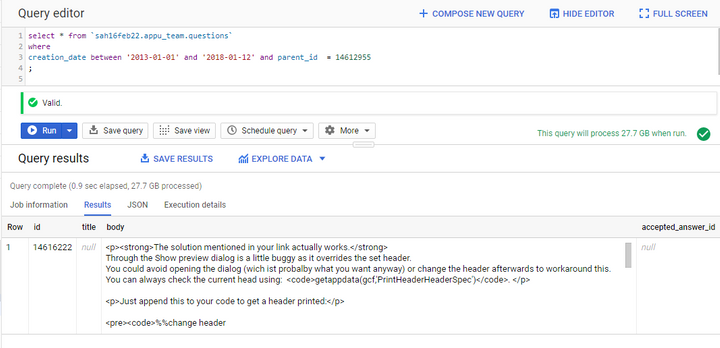
 `sah16feb22.appu\_team.questions`

WHERE

 creation\_date BETWEEN '2013-01-01'

 AND '2018-01-12'

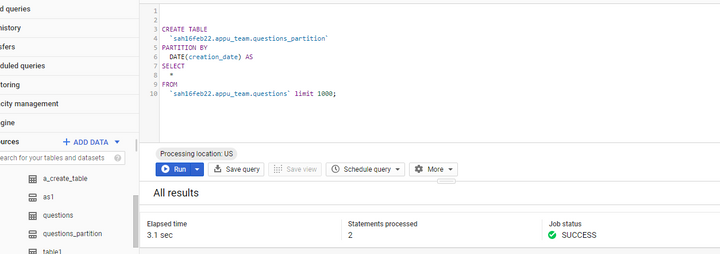
 AND parent\_id = 14612955 ;

****

Query takes 0.9 seconds to process 27.7 GB of data.

**Create a partition table** under the dataset.

CREATE TABLE  
sah16feb22.appu\_team.questions\_partition  
PARTITION BY  
DATE(creation\_date) AS  
SELECT  
\*  
FROM  
sah16feb22.appu\_team.questions limit 1000;



Now run the query against the partition table.

Select \* from ` sah16feb22.appu\_team.questions\_partition`

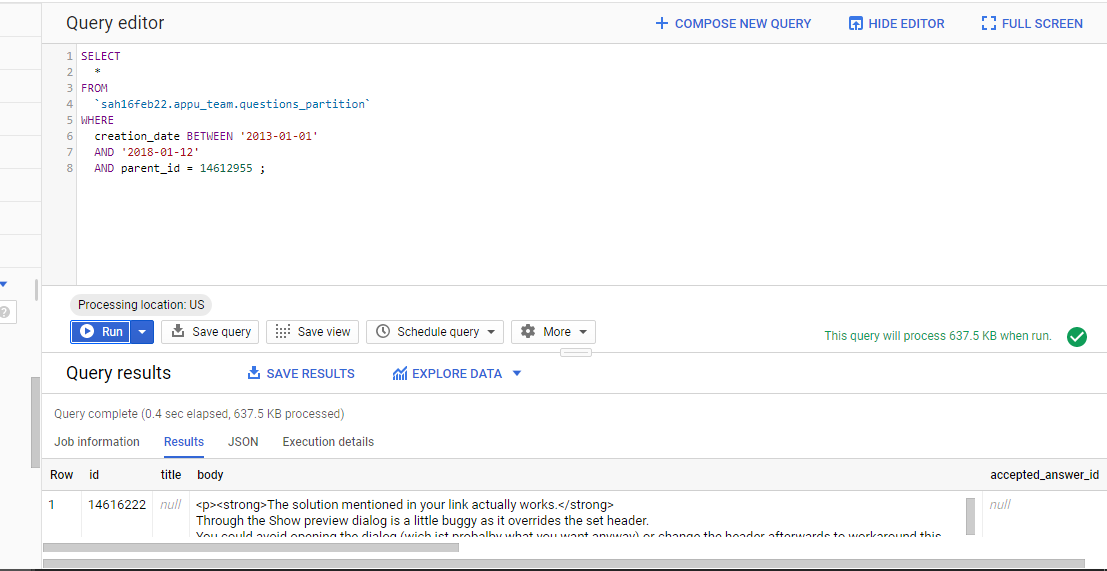
creation\_date BETWEEN '2013-01-01'

 AND '2018-01-12'

 AND parent\_id = 14612955

Limit 10;

tags = ‘android’



Here the query takes 0.4 sec to run 637.5KB of data.

**Note**:

* just give what date field you want to partition by …then BigQuery does…what partition ranges to use to maximize efficiency.
* BigQuerysupports single-column partitioning. If you have a composite primary key, you must choose a column from the primary key list as the partition key.

**Disadvantage**

While running the query against my standard table, it has to scan all the data and all of the ‘homeTeamName’ and the ‘startTime’ column. It takes few seconds to run 1.76 GB of data.

**CLUSTER**

Is a method where BigQuery optimizes the placement of our data as it to store data closer together to make queries more efficient. We can apply clustering on multiple columns.

CREATE TABLE

  `projectddl.stackoverflow.questions\_clustering\_new`

PARTITION BY

  DATE (date)

CLUSTER BY

  name AS

SELECT

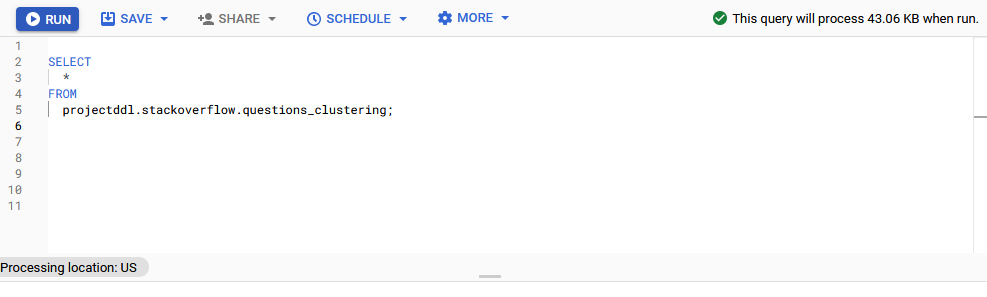
  \*

FROM

  `bigquery-public-data.stackoverflow.badges`

LIMIT

  1000;

****

SELECT

  \*

FROM

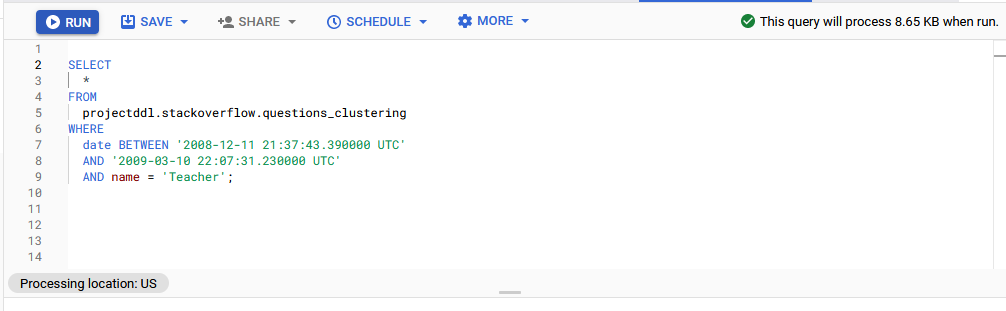
  projectddl.stackoverflow.questions\_clustering

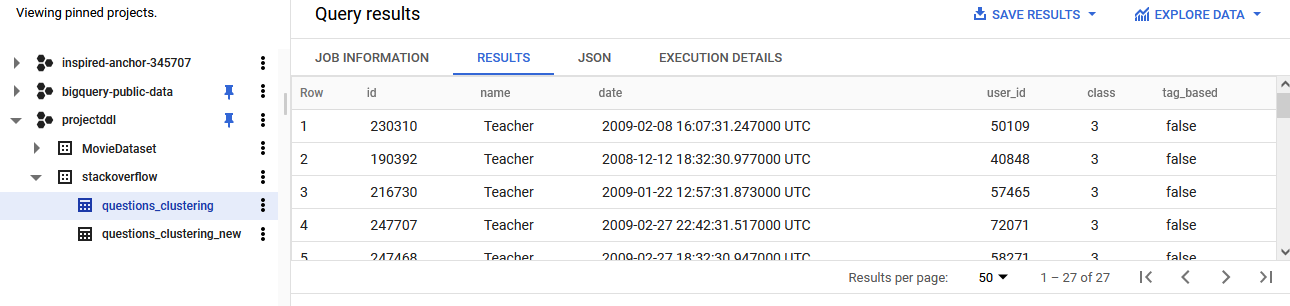
WHERE

  date BETWEEN '2008-12-11 21:37:43.390000 UTC'

  AND '2009-03-10 22:07:31.230000 UTC'

  AND name = 'Teacher';

****

****

Query runs 8.65 KB of data.

==================================

In partitioning a table is divided to sections by partitions. Dividing a large table into smaller partitions allows to run queries faster while spreading less. Partitions boundaries are based on UTC time. We can partition BigQuery tables by:

* **Time-unit column**: Tables are partitioned based on a [TIMESTAMP](https://cloud.google.com/bigquery/docs/reference/standard-sql/data-types#timestamp_type), [DATE](https://cloud.google.com/bigquery/docs/reference/standard-sql/data-types#date_type), or [DATETIME](https://cloud.google.com/bigquery/docs/reference/standard-sql/data-types#datetime_type) column in the table.
* **Ingestion time**: Tables are partitioned based on the timestamp when BigQuery ingests the data.
* **Integer range**: Tables are partitioned based on an integer column.

## **Time-unit column partitioning**

* 1. **TIMESTAMP and DATETIME**
* hourly
* daily
* Monthly
* Yearly
  1. **DATE**
* daily
* monthly
* yearly

## **Ingestion time partitioning**

An ingestion-time partitioned table has a pseudo column named \_PARTITIONTIME. The value of this column is the ingestion time for each row, truncated to the partition boundary (such as hourly or daily).

* Hourly
* Daily
* Monthly
* Yearly

## **Integer range partitioning**

To create an integer-range partitioned table, it needs

* The partitioning column.
* The starting value for range partitioning (inclusive).
* The ending value for range partitioning (exclusive).
* The interval of each range within the partition.

**Choosing daily, hourly, monthly, or yearly partitioning**

**Daily partitioning**

* Daily partitioning is useful when the given data is spread out over a wide range of dates, or if data is continuously added over time.

**Hourly partitioning**

* Tables have a high volume of data that spans a short date range — typically less than six months of timestamp values.

**Monthly or yearly partitioning**

* Is used when data requires frequent update or to add rows that span a wide date range(more than 500 dates)
* Have relatively small amount of data.

Grant Identity and Access Management (IAM) roles that give users the necessary permissions to perform each task in this document.

# **Creating partitioned tables**

## **Create an empty partitioned table**

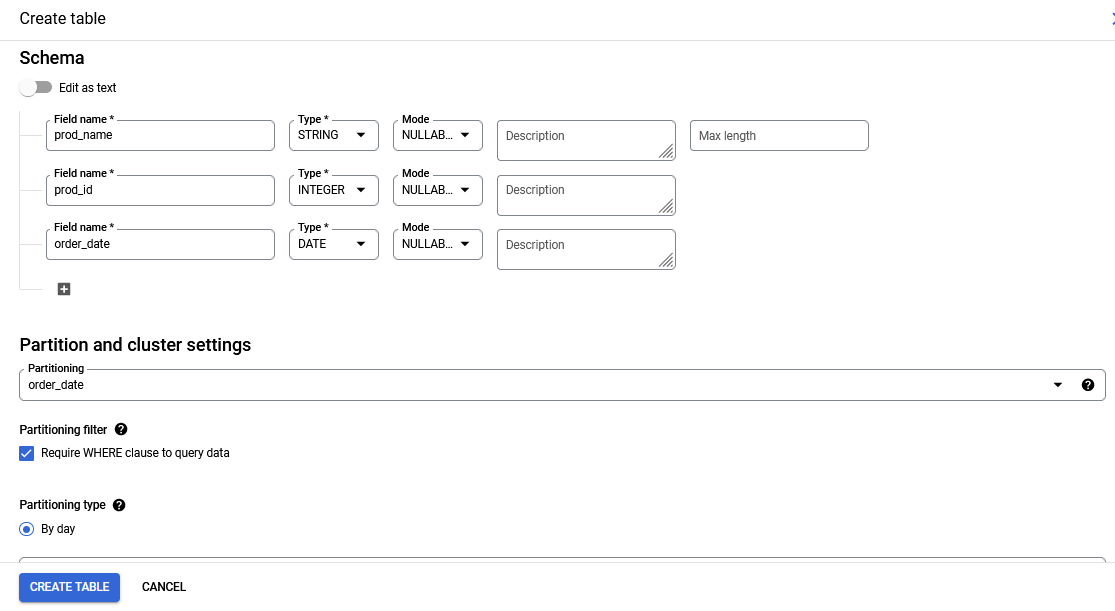
The steps to create a partitioned table in BigQuery are similar to creating a [standard table](https://cloud.google.com/bigquery/docs/tables), except that you specify the partitioning options, along with other table options.

*Create a time-unit column-partitioned table*

* Open the BigQuery page in the Cloud Console.

[Go to the BigQuery page](https://console.cloud.google.com/bigquery)

* In the **Explorer** panel, expand your project and select a dataset.
* Create table under a **dataset**
* In the **Schema** section, enter the [schema](https://cloud.google.com/bigquery/docs/schemas) definition.
  + Option 1: Use **Add field** and specify each field's **Name**, [**Type**](https://cloud.google.com/bigquery/docs/schemas#standard_sql_data_types), and [**Mode**](https://cloud.google.com/bigquery/docs/schemas#field_modes)
  + Option 2: Click **Edit as text** and paste the schema in the form of a JSON array.
* In the Schema section, enter the schema definition. Make sure the schema includes a DATE, TIMESTAMP, or DATETIME column for the partitioning column.
* In the Partition and cluster settings section, in the Partitioning drop-down list
  + Select Partition by field and choose the partitioning column. This option is only available if the schema contains a DATE, TIMESTAMP, or DATETIME column.



**Create an ingestion-time partitioned table**

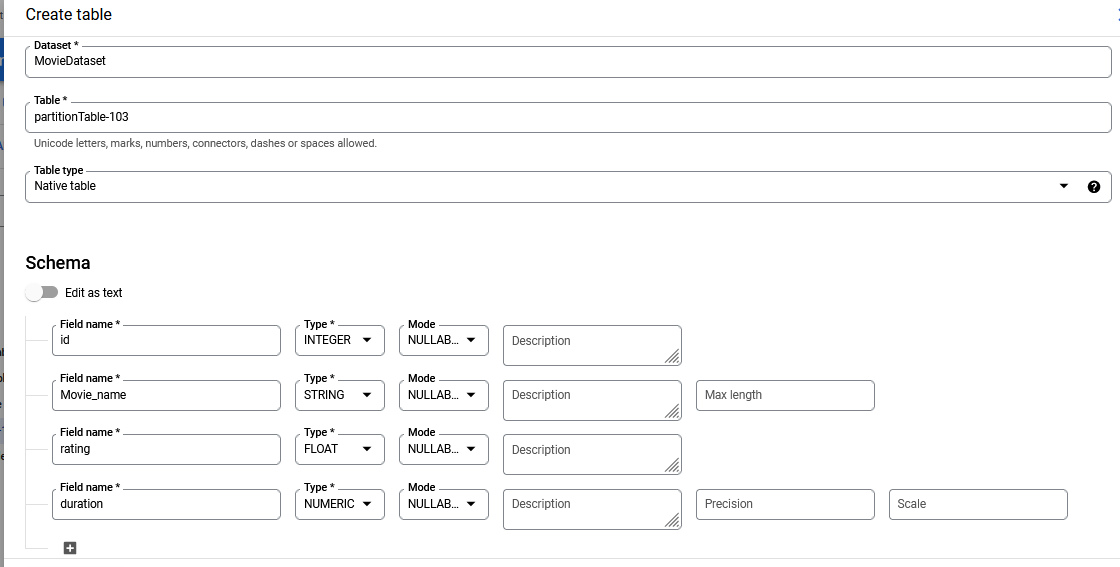
By selecting each different type of partitioning, we can see the various options that that go along with it:

* If we want to partition by ingestion time we must also provide the partitioning ingestion time granularity (hour, day, month, year).

### 

### Create an integer-range partitioned table

If we choose to partition the table based on an integer-range and column (e.g., customer\_id), we must also provide the start value, end value and the interval. We have used *end value*: 100





Click on **CREATE TABLE**.

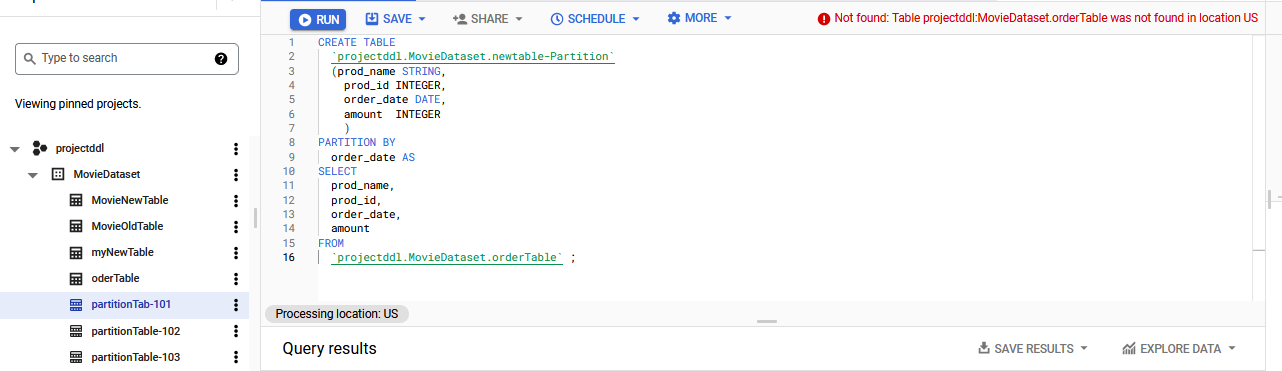
## **How to partition an existing BigQuery table**

In order to partition an existing table, we must

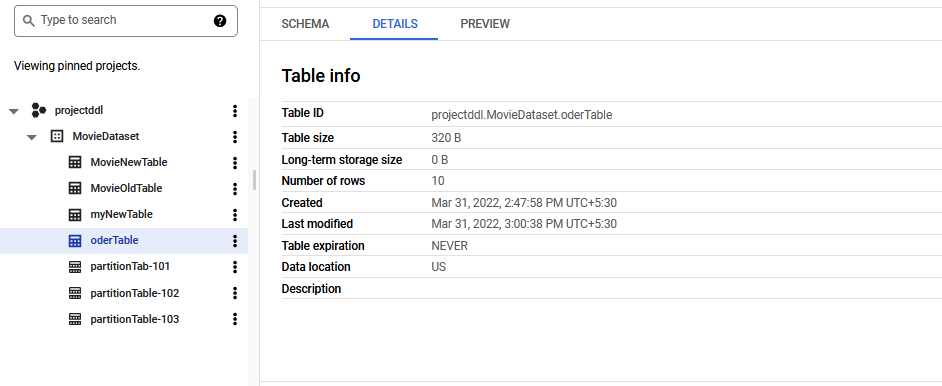
* Use a query to create a new table and provide the necessary options for the partitioning.
* **Note**, when using a query, you can only copy an existing BigQuery table and partition it either by an integer-range column or by a time-unit column, as it’s not possible to use ingestion-time partitioning.

**Basic Type Specific Type**

**Numeric INTEGER , SHORTINTEGER , LONGINTEGER , DECIMAL , SHORTDECIMAL , NUMBER**



Details:::



Check (Details, Schema, Preview) options…After the table is created.

ALTER TABLE `projectddl.MovieDataset.MovieOldTable`

 SET OPTIONS (

   partition\_expiration\_days=5

 ) ;

CREATE OR REPLACE TABLE `projectddl.MovieDataset.orderTableNew\_11` AS

SELECT \*

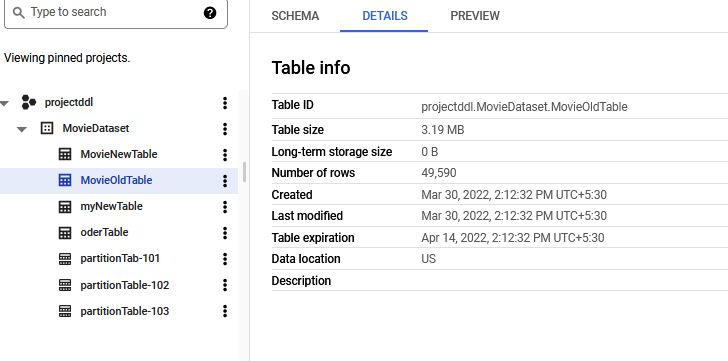
FROM `projectddl.MovieDataset.oder\_new`

WHERE order\_date BETWEEN '01-04-2020' AND '01-07-2020';SELECT \* FROM `projectddl.MovieDataset. orderNewTable` LIMIT 10;

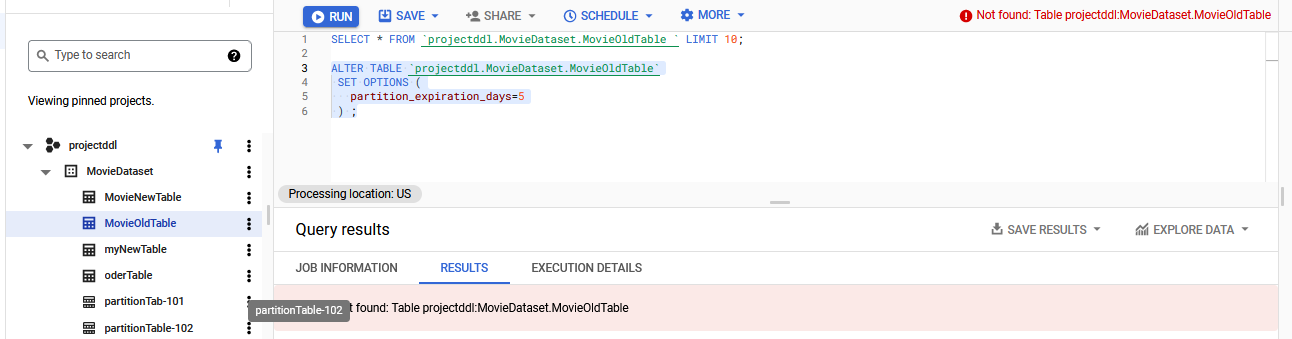
## **BigQuery partitioned table expiration**

Partitioning a table using a **time-unit** or **ingestion-time** as the partition type, we can specify a partition **expiration time**. This configuration option specifies how long BigQuery will keep the data in a partition. After the expiration date of a partition is passed, BigQuery deletes all data in this particular partition.

In order to alter an existing table partition expiration date, we can just use the ALTER\_TABLE command and adjust the expiration date option



**Error shows**

****

**Before partitioning**

## **How to query a BigQuery partitioned table**

<https://blog.coupler.io/bigquery-partition/#How_to_create_a_new_BigQuery_partitioned_table>

<https://cloud.google.com/bigquery/docs/managing-partitioned-tables>

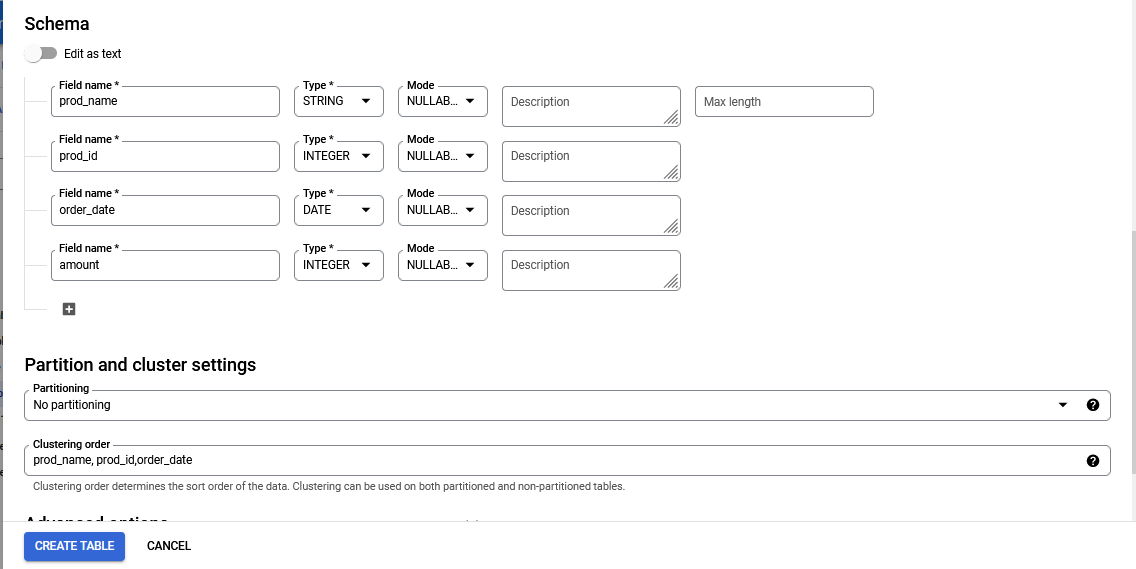
**Numeric** and **bignumeric** **types** can contain fractional numbers and are defined with precision (number of digits) as well as scale (number of digits after the decimal point). The specs are as follows:

|  |  |  |
| --- | --- | --- |
|  | NUMERIC | BIGNUMERIC |
| Precision | 38 | 76.76 (77th digit is partial) |
| Scale | 8 | 38 |

**Clustering**

When a table is clustered in BigQuery, the table data is automatically organized based on the contents of one or more columns in the tables’s schema. When data is written to a clustered table, BigQuery sorts the data using the values in the clustering columns.

**To create an empty clustered table with a schema definition:**



CLICK on CREATE TABLE.

### Create a clustered table from a query result

There are two ways to create a clustered table from a query result:

* Write the results to a new destination table and specify the clustering columns. We can create a clustered table by specifying clustering columns when you load data into a new table. It does not need to create an empty table before loading data into it.
* By using a DDL CREATE TABLE AS SELECT statement.

**Limitation**

* You can create a clustered table by querying either a partitioned table or a non-partitioned table. You cannot change an existing table to a clustered table by using query results.
* When you create a clustered table from a query result, you must use standard SQL. Currently, legacy SQL is not supported for querying clustered tables or for writing query results to clustered tables.

SELECT \* FROM `projectddl.MovieDataset.orderNewTable` LIMIT 10;

CREATE OR REPLACE TABLE `projectddl.MovieDataset.clustered`

PARTITION BY

  DATE(order\_date)

CLUSTER BY

  tags AS

SELECT

  \*

FROM

  `projectddl.MovieDataset.orderNewTable`

WHERE

  order\_date BETWEEN '2020-5-01' AND '2020-10-01';