Introduction to clustered tables

1. When you create a clustered table in BigQuery, the table data is automatically organized based on the contents of one or more columns in the table's schema.
2. When you cluster a table using multiple columns, the order of columns you specify is important. The order of the specified columns determines the sort order of the data.
3. Clustering can improve the performance of certain types of queries such as queries that use filter clauses and queries that aggregate data.

When to use clustering

Both partitioning and clustering can improve performance and reduce query cost.

Use clustering under the following circumstances:

- You don't need strict cost guarantees before running the query.
- You need more granularity than partitioning alone allows. To get clustering benefits in addition to partitioning benefits, you can use the same column for both partitioning and clustering.
- Your queries commonly use filters or aggregation against multiple particular columns.
- The cardinality of the number of values in a column or group of columns is large.

Use partitioning under the following circumstances:

- You want to know query costs before a query runs. Partition pruning is done before the query runs, so you can get the query cost after partitioning pruning through a dry run. Cluster pruning is done when the query runs, so the cost is known only after the query finishes.
- You need partition-level management. For example, you want to set a partition expiration time, load data to a specific partition, or delete partitions.
- You want to specify how the data is partitioned and what data is in each partition. For example, you want to define time granularity or define the ranges used to partition the table for integer range partitioning.

Prefer clustering over partitioning under the following circumstances:

- Partitioning results in a small amount of data per partition (approximately less than 1 GB).
- Partitioning results in a large number of partitions beyond the limits on partitioned tables.
- Partitioning results in your mutation operations modifying most partitions in the table frequently (for example, every few minutes).

You can also combine partitioning with clustering. Data is first partitioned and then data in each partition is clustered by the clustering columns.

Creating and using clustered tables

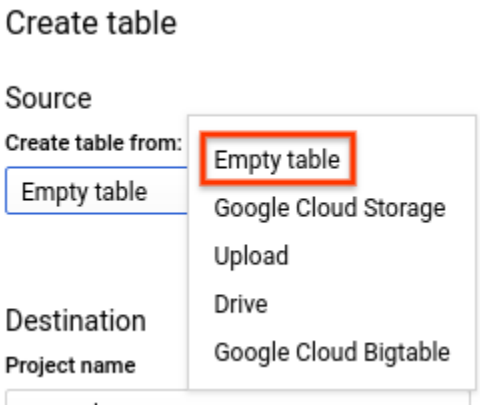
Limitations

Clustered tables in BigQuery are subject to the following limitations:

- Only standard SQL is supported for querying clustered tables and for writing query results to clustered tables.
- Clustering columns must be top-level, non-repeated columns of one of the following types:
    - DATE
    - BOOL
    - GEOGRAPHY
    - INT64
    - NUMERIC
    - BIGNUMERIC
    - STRING
    - TIMESTAMP
    - DATETIME
- For more information about data types, see Standard SQL data types.
- You can specify up to four clustering columns.
- When using STRING type columns for clustering, BigQuery uses only the first 1,024 characters to cluster the data. The values in the columns can themselves be longer than 1,024.

To create an empty clustered table with a schema definition:

1. In the Google Cloud Console, go to the BigQuery page.
   Go to the BigQuery page
2. In the **Explorer** panel, expand your project and select a dataset.
3. Expand the more_vert **Actions** option and click **Open**.
4. In the details panel, click **Create table** add_box.
5. On the **Create table** page, under **Source**, for **Create table from**, select **Empty table**.



6. Under **Destination**:
   - For **Dataset name**, choose the appropriate dataset, and in the **Table name** field, enter the name of the table you're creating.
   - Verify that **Table type** is set to **Native table**.
7. Under **Schema**, enter the schema definition.
   - Enter schema information manually by:
     - Enabling **Edit as text** and entering the table schema as a JSON array.
       Note: You can view the schema of an existing table in JSON format by entering the following command in the **bq** command-line tool: **bq show --format=prettyjson dataset.table**.
     - Using **Add field** to manually input the schema.
8. For **Clustering order**, enter between one and four comma-separated column names.
9. (Optional) Click **Advanced options** and for **Encryption**, click **Customer-managed key** to use a Cloud Key Management Service key. If you leave the **Google-managed key** setting, BigQuery encrypts the data at rest.
10. Click **Create table**.

```
CREATE TABLE
  mydataset.myclusteredtable
PARTITION BY
  DATE(timestamp)
CLUSTER BY
  customer_id AS
SELECT
  *
FROM
  `mydataset.mytable`
```

Clustering improves efficiency, but there are some limitations:

- Clustering is only supported for partitioned tables.
- We can specify the clustering column only while creating a table. We can't modify it later.
- We can specify a maximum of four non-repeated columns for clustering.
- Clustering can only be used with standard SQL.