**BigQuery Best Practices**

BigQuery has two pricing models for running queries:

* [On-demand pricing](https://cloud.google.com/bigquery/pricing#on_demand_pricing): You pay for the number of bytes processed by each query.
* [Flat-rate pricing](https://cloud.google.com/bigquery/pricing#flat-rate_pricing): You pay for dedicated query processing capacity, measured in [slots](https://cloud.google.com/bigquery/docs/slots).

If you are using on-demand pricing, then you can directly reduce costs by reducing the number of bytes that a query processes. With flat-rate pricing, your cost is fixed based on the number of slots that you purchase and the [slot commitment plan](https://cloud.google.com/bigquery/docs/reservations-details#commitment-plans) that you select. However, optimizing your queries can help to reduce slot usage.

**Avoid**SELECT \*

**Best practice:** Query only the columns that you need.

Using SELECT \* is the most expensive way to query data. When you use SELECT \*, BigQuery does a full scan of every column in the table.

If you are experimenting with data or exploring data, use one of the [data preview options](https://cloud.google.com/bigquery/docs/best-practices-costs#preview-data) instead of SELECT \*.

Applying a LIMIT clause to a SELECT \* query does not affect the amount of data read. You are billed for reading all bytes in the entire table, and the query counts against your free tier quota.

Instead, query only the columns you need. For example, use SELECT \* EXCEPT to exclude one or more columns from the results.

If you do require queries against every column in a table, but only against a subset of data, consider:

* Materializing results in a destination table and querying that table instead
* [Partitioning your tables by date](https://cloud.google.com/bigquery/docs/creating-partitioned-tables) and querying the relevant partition; for example, WHERE \_PARTITIONDATE="2017-01-01" only scans the January 1, 2017 partition

**Sample data using preview options**

**Best practice:** Don't run queries to explore or preview table data.

If you are experimenting with or exploring your data, you can use table preview options to view data for free and without affecting quotas.

BigQuery supports the following data preview options:

* In the Cloud Console, on the table details page, click the **Preview** tab to sample the data.
* In the bq command-line tool, use the [bq head](https://cloud.google.com/bigquery/docs/managing-table-data" \l "browse-table) command and specify the number of rows to preview.
* In the API, use [tabledata.list](https://cloud.google.com/bigquery/docs/reference/rest/v2/tabledata/list) to retrieve table data from a specified set of rows.

**Price your queries before running them**

**Best practice:** Before running queries, preview them to estimate costs.

Queries are billed according to the number of bytes read. To estimate costs before running a query:

* View the query validator in the Cloud Console
* Use the [Google Cloud Platform Pricing Calculator](https://cloud.google.com/products/calculator/)
* Perform a dry run by using the:
  + --dry\_run flag in the bq command-line tool
  + dryRun parameter when submitting a query job using the API

**Using the query validator**

When you enter a query in the Cloud Console, the query validator verifies the query syntax and provides an estimate of the number of bytes read. You can use this estimate to calculate query cost in the pricing calculator.

* If your query is not valid, then the query validator displays an error message. For example:

Not found: Table myProject:myDataset.myTable was not found in location US

* If your query is valid, then the query validator provides an estimate of the number of bytes required to process the query. For example:

This query will process 623.1 KiB when run.

**Note:** Megabyte is the smallest unit that is measured by the pricing calculator.

**Performing a dry run**

To perform a dry run:

[Console](https://cloud.google.com/bigquery/docs/best-practices-costs#console)[bq](https://cloud.google.com/bigquery/docs/best-practices-costs#bq)[API](https://cloud.google.com/bigquery/docs/best-practices-costs#api)[Go](https://cloud.google.com/bigquery/docs/best-practices-costs#go)[Java](https://cloud.google.com/bigquery/docs/best-practices-costs#java)[Node.js](https://cloud.google.com/bigquery/docs/best-practices-costs#node.js)[Python](https://cloud.google.com/bigquery/docs/best-practices-costs#python)

1. Go to the BigQuery page in the Google Cloud Console.

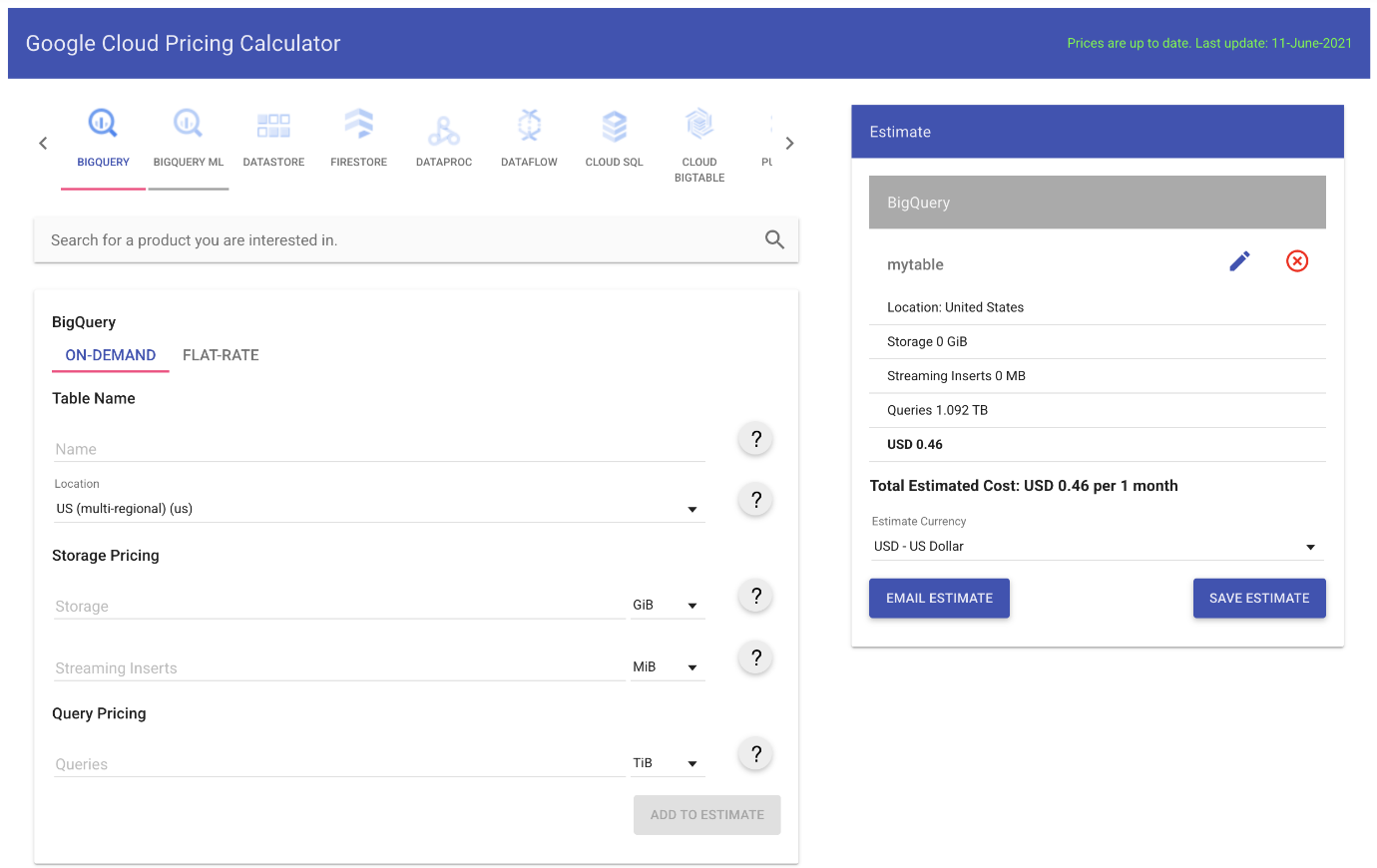
[Go to BigQuery](https://console.cloud.google.com/bigquery)

1. Enter your query in the **Query editor**.

If the query is valid, then a check mark automatically appears along with the amount of data that the query will process. If the query is invalid, then an exclamation point appears along with an error message.

**Using the pricing calculator**

To estimate query costs in the [Google Cloud Platform Pricing Calculator](https://cloud.google.com/products/calculator/), enter the number of bytes that are processed by the query as MB, GB, TB, or PB. If your query processes less than 1 TB, the estimate is $0 because BigQuery provides 1 TB of on-demand query processing free per month.



**Limit query costs by restricting the number of bytes billed**

**Best practice:** Use the maximum bytes billed setting to limit query costs.

You can limit the number of bytes billed for a query using the maximum bytes billed setting. When you set maximum bytes billed, the number of bytes that the query will read is estimated before the query execution. If the number of estimated bytes is beyond the limit, then the query fails without incurring a charge.

For clustered tables, the estimation of the number of bytes billed for a query is an upper bound, and can be higher than the actual number of bytes billed after running the query. So in some cases, if you set the maximum bytes billed, a query on a clustered table can fail, even though the actual bytes billed wouldn't exceed the maximum bytes billed setting.

If a query fails because of the maximum bytes billed setting, an error like the following is returned:

Error: Query exceeded limit for bytes billed: 1000000. 10485760 or higher required.

To set the maximum bytes billed:

[Console](https://cloud.google.com/bigquery/docs/best-practices-costs#console)[bq](https://cloud.google.com/bigquery/docs/best-practices-costs#bq)[API](https://cloud.google.com/bigquery/docs/best-practices-costs#api)

1. In the **Query editor**, click **More**, click **Query settings**, and then click **Advanced options**.
2. In the **Maximum bytes billed** field, enter an integer.
3. Click **Save**.

**Use clustered or partitioned tables**

**Best practice:** Use clustering and partitioning to reduce the amount of data scanned.

Clustering and partitioning can help to reduce the amount of data processed by queries. To limit the number of partitions scanned when querying clustered or partitioned tables, use a predicate filter.

If you run a query against a clustered table, and the query includes a filter on the clustered columns, then BigQuery uses the filter expression and the block metadata to prune the blocks scanned by the query. For more information, see [Querying clustered tables](https://cloud.google.com/bigquery/docs/querying-clustered-tables).

When querying partitioned tables, filters on the partitioning column are used to prune the partitions and therefore can reduce the query cost. For more information, see [Querying partitioned tables](https://cloud.google.com/bigquery/docs/querying-partitioned-tables#querying_partitioned_tables_2).

**Do not use LIMIT to control costs in non-clustered tables**

**Best practice:** For non-clustered tables, not use a LIMIT clause as a method of cost control.

For non-clustered tables, applying a LIMIT clause to a query does not affect the amount of data that is read. You are billed for reading all bytes in the entire table as indicated by the query, even though the query returns only a subset. With a clustered table, a LIMIT clause can reduce the number of bytes scanned.

**View costs using a dashboard and query your audit logs**

**Best practice:** Create a dashboard to view your billing data so you can make adjustments to your BigQuery usage. Also consider streaming your audit logs to BigQuery so you can analyze usage patterns.

You can [export your billing data](https://cloud.google.com/billing/docs/how-to/export-data-bigquery) to BigQuery and visualize it in a tool such as Google Data Studio. For a tutorial on creating a billing dashboard, see [Visualize GCP billing using BigQuery and Google Data Studio](https://medium.com/google-cloud/visualize-gcp-billing-using-bigquery-and-data-studio-d3e695f90c08).

You can also stream your [audit logs](https://cloud.google.com/bigquery/docs/reference/auditlogs) to BigQuery and analyze the logs for usage patterns such as query costs by user.

**Partition data by date**

**Best practice:** Partition your tables by date.

If possible, [partition](https://cloud.google.com/bigquery/docs/creating-partitioned-tables) your BigQuery tables by date. Partitioning your tables lets you query relevant subsets of data which improves performance and reduces costs.

For example, when you query partitioned tables, use the \_PARTITIONTIME pseudo column to filter for a date or a range of dates. The query processes data only in the partitions that are specified by the date or range.

**Materialize query results in stages**

**Best practice:** If possible, materialize your query results in stages.

If you create a large, multi-stage query, each time you run it, BigQuery reads all the data that is required by the query. You are billed for all the data that is read each time the query is run.

Instead, break your query into stages where each stage materializes the query results by writing them to a [destination table](https://cloud.google.com/bigquery/querying-data#permanent-table). Querying the smaller destination table reduces the amount of data that is read and lowers costs. The cost of storing the materialized results is much less than the cost of processing large amounts of data.

**Consider the cost of large result sets**

**Best practice:** If you are writing large query results to a destination table, use the default table expiration time to remove the data when it's no longer needed.

Keeping large result sets in BigQuery storage has a cost. If you don't need permanent access to the results, use the [default table expiration](https://cloud.google.com/bigquery/docs/updating-datasets#table-expiration) to automatically delete the data for you.