

BigQuery pricing

This page is about BigQuery pricing.

For BigQuery ML, see [BigQuery ML pricing](/bigquery-ml/pricing) (/bigquery-ml/pricing).

For BigQuery Data Transfer Service, see [BigQuery Data Transfer Service pricing](/bigquery/transfer-service-pricing) (/bigquery/transfer-service-pricing).

Overview

BigQuery offers scalable, flexible pricing options to meet your technical needs and your budget.

Storage costs are based on the amount of data stored in BigQuery. Storage charges can be:

- [Active](#) (#active_storage) – A monthly charge for data stored in tables or in partitions that have been modified in the last 90 days.
- [Long-term](#) (#long-term-storage) – A lower monthly charge for data stored in tables or in partitions that have not been modified in the last 90 days.

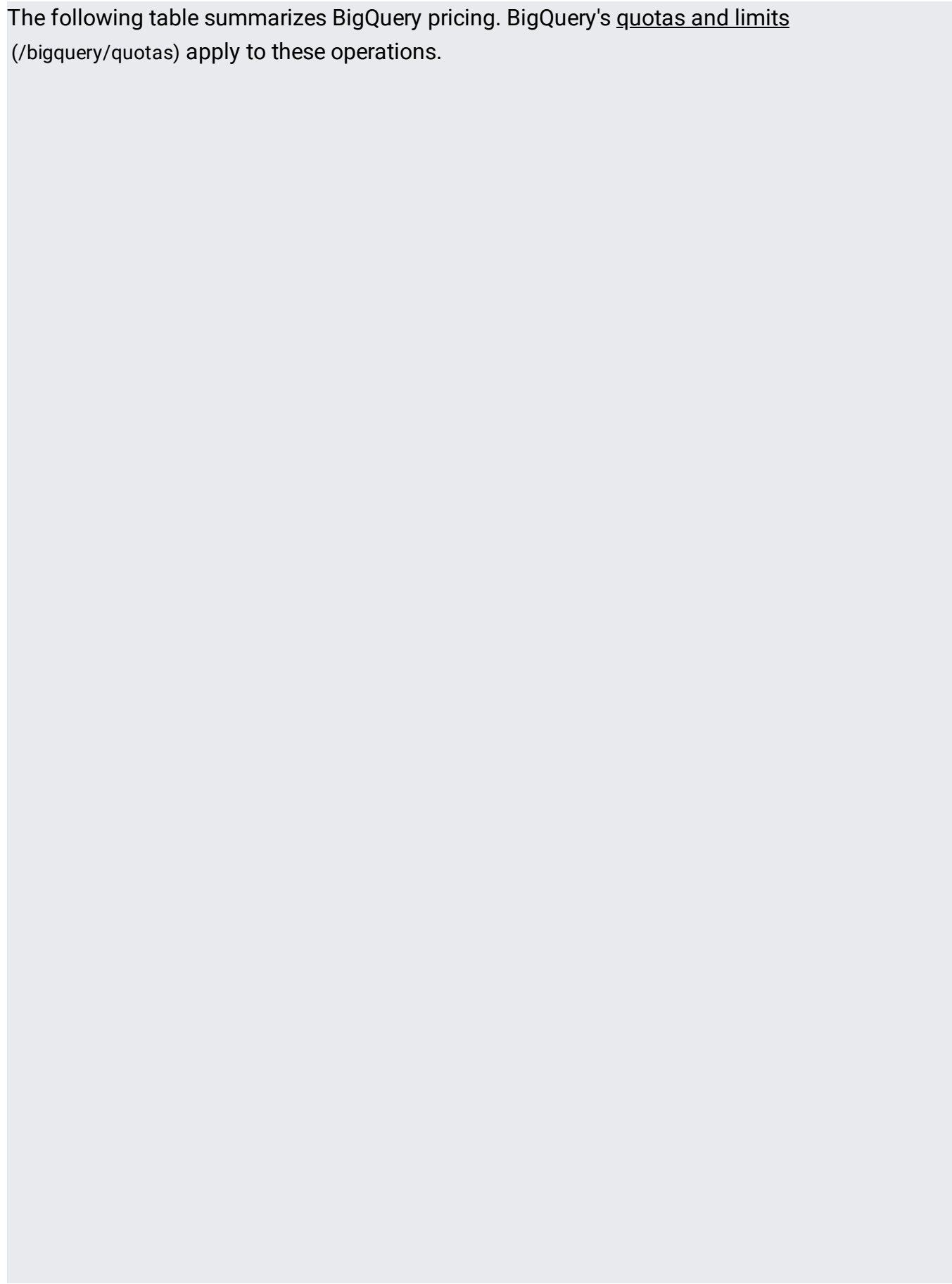
For query costs, you can choose between two pricing models:

- [On-demand](#) (#on_demand_pricing) – This is the most flexible option. On-demand pricing is based on the amount of data processed by each query you run.
- [Flat-rate](#) (#flat_rate_pricing) – This pricing option is best for customers who desire cost predictability. Flat-rate customers purchase dedicated resources for query processing and are not charged for individual queries.

For more information about storage and query pricing, see [Google Cloud SKUs](/skus/?filter=bigquery) (/skus/?filter=bigquery). Note that on-demand query pricing is referred to as *analysis pricing* on the SKUs page.

Pricing summary

The following table summarizes BigQuery pricing. BigQuery's quotas and limits (/bigquery/quotas) apply to these operations.



US (multi-region)		Monthly
Operation	Pricing	Details
Storage	\$0.020 per GB	The first 10 GB is free each month. See Storage pricing for details.
Long-term storage	\$0.010 per GB	The first 10 GB is free each month. See Storage pricing for details.
Query Storage API	\$1.10 per TB	The BigQuery Storage API is not included in the Google Cloud Free Tier .
Streaming inserts	\$0.010 per 200 MB	You are charged for rows that are successfully inserted. Individual rows are calculated using a 1 KB minimum size. See Streaming pricing for details.
Series (on-demand)	\$5.00 per TB	The first 1 TB per month is free. See On-demand pricing for details.
Series (hourly flex slots)	\$4.00 per 100 slots	You can purchase additional slots in 100-slot increments. See Flex slots: short-term commitments for details.
Series (monthly flat-rate)	\$2,000 per 100 slots	You can purchase additional slots in 100-slot increments. See Monthly flat-rate commitments for details.
Series (annual flat-rate)	\$1,700 per 100 slots	You can purchase additional slots in 100-slot increments. You are billed monthly. See Annual flat-rate commitments for details.

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

How charges are billed

Each project you create has a billing account attached to it. Any charges incurred by BigQuery jobs run in the project are billed to the attached billing account. BigQuery storage charges are also billed to the attached billing account.

How to analyze billing data

You can view BigQuery costs and trends by using the Cloud Billing reports page in the Cloud Console. For information on analyzing billing data using reports, see [View your billing reports and cost trends \(/billing/docs/how-to/reports\)](/billing/docs/how-to/reports).

For information on analyzing your billing data in BigQuery, see [Export Cloud Billing data to BigQuery \(/billing/docs/how-to/export-data-bigquery\)](/billing/docs/how-to/export-data-bigquery) in the [Cloud Billing documentation \(/billing/docs\)](/billing/docs).

Free operations

The following table shows BigQuery operations that are free of charge in every location. BigQuery's [quotas and limits \(/bigquery/quotas\)](/bigquery/quotas) apply to these operations.

Operation Details

Loading data When you load data into BigQuery from Cloud Storage, you are not charged for the load operation, but you do incur charges for storing the data in Cloud Storage. See [Data storage \(/storage/pricing#storage-pricing\)](/storage/pricing#storage-pricing) on the Cloud Storage pricing page for details. Once the data is loaded into BigQuery, the data is subject to BigQuery's [Storage pricing \(/bigquery/pricing#storage\)](/bigquery/pricing#storage). For more information, see [Loading data into BigQuery \(/bigquery/loading-data\)](/bigquery/loading-data).

When you create a dataset in BigQuery, you must choose a location for the data. If you choose **US**, you can load data into tables in the dataset from a Cloud Storage bucket in any other region. When you load data from another region into a US dataset, there is currently no charge for internet egress.

If you choose a location other than **US**, you must do one of the following:

- Load data from a Cloud Storage bucket in that region (the bucket can be either a multi-regional bucket or a regional bucket in the same region as the dataset).
- Copy the data into a bucket in that region.

When you copy data from one Cloud Storage region to another, Cloud Storage [network pricing](/storage/pricing#network-buckets) (/storage/pricing#network-buckets) applies.

Copying data You are not charged for copying a table, but you do incur charges for storing the new table and the table you copied. For more information, see [Copying an existing table](/bigquery/docs/managing-tables#copy-table) (/bigquery/docs/managing-tables#copy-table).

Exporting data When you export data from BigQuery to Cloud Storage, you are not charged for the export operation, but you do incur charges for storing the data in Cloud Storage. See [Data storage](/storage/pricing#storage-pricing) (/storage/pricing#storage-pricing) on the Cloud Storage pricing page for details. For more information, see [Exporting data from BigQuery](/bigquery/docs/exporting-data) (/bigquery/docs/exporting-data).

Deleting datasets You are not charged for deleting a dataset.

Deleting tables, views, partitions, and functions You are not charged for deleting a table, deleting a view, deleting individual table partitions, or deleting a user-defined function.

Metadata You are not charged for list, get, patch, update and delete calls. Examples include (but are not operations limited to): listing datasets, updating a dataset's access control list, updating a table's description, or listing user-defined functions in a dataset.

Reading pseudo columns You are not charged for querying the contents of the following pseudo columns:

- _TABLE_SUFFIX** – Used when querying [wildcard tables](/bigquery/docs/querying-wildcard-tables) (/bigquery/docs/querying-wildcard-tables) or to achieve [table decorator semantics in standard SQL](/bigquery/docs/reference/standard-sql/migrating-from-legacy-sql#table_decorators_and_wildcard_functions) (/bigquery/docs/reference/standard-sql/migrating-from-legacy-sql#table_decorators_and_wildcard_functions)
- _PARTITIONDATE** – Used when querying [ingestion-time partitioned tables](/bigquery/docs/querying-partitioned-tables#ingestion-time_partitioned_table_pseudo_columns) (/bigquery/docs/querying-partitioned-tables#ingestion-time_partitioned_table_pseudo_columns)
- _PARTITIONTIME** – Used when querying [ingestion-time partitioned tables](/bigquery/docs/querying-partitioned-tables#ingestion-time_partitioned_table_pseudo_columns) (/bigquery/docs/querying-partitioned-tables#ingestion-time_partitioned_table_pseudo_columns)
- _FILE_NAME** – Used when querying tables based on [external data sources](/bigquery/external-data-sources) (/bigquery/external-data-sources)

Reading meta You are not charged for querying the contents of the following meta tables:

tables `__PARTITIONS_SUMMARY__` – Used when getting metadata about partitions in a partitioned table or an ingestion-time partitioned table
`__TABLES_SUMMARY__` – Used when getting metadata about the tables and views in a dataset

Creating, replacing, or calling UDFs Currently, you are not charged for creating, replacing, or invoking persistent [user-defined functions](#) (UDFs). Persistent UDFs is currently a [beta](#) (/products#product-launch-stages) release; pricing is subject to change.

Always free usage limits

As part of the [Google Cloud Free Tier](#) (/free), BigQuery offers some resources free of charge up to a specific limit. These free usage limits are available during and after the free trial period. If you go over these usage limits and are no longer in the free trial period, you will be charged according to the pricing on this page.

Resource	Monthly free usage limits	Details
Storage	The first 10 GB per month is free.	BigQuery ML models and training data stored in BigQuery are included in the BigQuery storage free tier.
Queries (analysis)	The first 1 TB of query data processed per month is free.	Queries that use BigQuery ML prediction, inspection, and evaluation functions are included in the BigQuery analysis free tier. BigQuery ML queries that contain <code>CREATE MODEL</code> statements are not. BigQuery flat-rate pricing (#flat_rate_pricing) is also available for high-volume customers that prefer a stable, monthly cost.
BigQuery ML <code>CREATE MODEL</code> queries	The first 10 GB of data processed by queries that contain <code>CREATE MODEL</code> statements per month is free.	BigQuery ML <code>CREATE MODEL</code> queries are independent of the BigQuery analysis free tier, and only apply to BigQuery ML built-in models (models that are trained within BigQuery).

Query pricing

Query pricing refers to the cost of running your SQL commands, and user-defined functions, and qualifying [Data Manipulation Language \(DML\)](#).

(</bigquery/docs/reference/standard-sql/data-manipulation-language>) and [Data Definition Language \(DDL\)](/bigquery/docs/reference/standard-sql/data-definition-language) (</bigquery/docs/reference/standard-sql/data-definition-language>) statements.

BigQuery offers a choice of two pricing models:

- **On-demand pricing** is flexible and efficient. You pay only for the queries you run.
- **Flat-rate pricing** offers predictable and consistent month-to-month costs.

By default, you are billed according to the on-demand pricing model. You can [change your billing model](/bigquery/docs/reservations-workload-management#choose-billing-model) (</bigquery/docs/reservations-workload-management#choose-billing-model>) to flat-rate billing or mix and match the two billing models for each project and location.

On-demand pricing

Under on-demand pricing, BigQuery charges for queries by using one metric: the number of bytes processed (also referred to as bytes read). You are charged for the number of bytes processed whether the data is stored in BigQuery or in an external data source such as Cloud Storage, Drive, or Cloud Bigtable. On-demand pricing is based solely on usage.

On-demand query pricing is as follows:

US (multi-region)		Monthly
Operation	Pricing	Details

Queries (on-demand)	\$5.00 per TB	The first 1 TB per month is free. Flat-rate pricing is also available for high-volume customers that prefer a stable, monthly cost.
---------------------	---------------	---

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Note the following regarding query charges:

- BigQuery uses a [columnar data structure](https://en.wikipedia.org/wiki/Column-oriented_DBMS) (https://en.wikipedia.org/wiki/Column-oriented_DBMS). You're charged according to the total data processed in the columns you select, and the total data per column is calculated

based on the types of data in the column. For more information about how your data size is calculated, see [Data size calculation \(/bigquery/pricing#data\)](/bigquery/pricing#data).

- You aren't charged for queries that return an error or for queries that retrieve [results from the cache \(/bigquery/docs/cached-results\)](/bigquery/docs/cached-results).
- Charges are rounded to the nearest MB, with a minimum 10 MB data processed per table referenced by the query, and with a minimum 10 MB data processed per query.
- Canceling a running query job might incur charges up to the full cost for the query if you let the query run to completion.
- When you run a query, you're charged according to the data processed in the columns you select, even if you set an explicit LIMIT on the results.
- [Partitioning \(/bigquery/docs/partitioned-tables\)](/bigquery/docs/partitioned-tables) and [clustering \(/bigquery/docs/clustered-tables\)](/bigquery/docs/clustered-tables) your tables can help reduce the amount of data processed by queries. As a best practice, use partitioning and clustering whenever possible.
- On-demand query pricing is referred to as analysis pricing on the [Google Cloud SKUs \(/skus/?filter=bigquery\)](/skus/?filter=bigquery) page.

On-demand query cost controls

BigQuery provides cost control mechanisms that enable you to cap your query costs. You can set:

- User-level and project-level [custom cost controls \(/bigquery/docs/custom-quotas\)](/bigquery/docs/custom-quotas)
- The [maximum bytes billed \(/bigquery/docs/best-practices-costs#limit_query_costs_by_restricting_the_number_of_bytes_billed\)](/bigquery/docs/best-practices-costs#limit_query_costs_by_restricting_the_number_of_bytes_billed) by a query

Querying Cloud Storage data

When querying an external data source from BigQuery, you are charged for the number of bytes read by the query. For more information, see [Query pricing \(/bigquery/pricing#queries\)](/bigquery/pricing#queries). You are also charged for storing the data on Cloud Storage. For more information, see [Cloud Storage Pricing \(/storage/pricing\)](/storage/pricing).

Querying columnar formats on Cloud Storage

If your external data is stored in ORC or Parquet, the number of bytes charged is limited to the columns that BigQuery reads. Because the data types from an external data source are converted to BigQuery data types by the query, the number of bytes read is computed based on the size of BigQuery data types. For information about data type conversions, see the following pages:

- [ORC conversions](/bigquery/docs/loading-data-cloud-storage-orc#orc_conversions) (/bigquery/docs/loading-data-cloud-storage-orc#orc_conversions)
- [Parquet conversions](/bigquery/docs/loading-data-cloud-storage-parquet#parquet_conversions) (/bigquery/docs/loading-data-cloud-storage-parquet#parquet_conversions)

Flat-rate pricing

BigQuery offers flat-rate pricing for customers who prefer a stable cost for queries rather than paying the on-demand price per TB of data processed.

You can choose to use flat-rate pricing using [BigQuery Reservations](/bigquery/docs/reservations-intro) (/bigquery/docs/reservations-intro).

When you enroll in flat-rate pricing, you purchase slot [commitments](/bigquery/docs/reservations-intro#commitments) (/bigquery/docs/reservations-intro#commitments)—dedicated query processing capacity, measured in BigQuery [slots](/bigquery/docs/slots) (/bigquery/docs/slots). Your queries consume this capacity, and you are not billed for bytes processed. If your capacity demands exceed your committed capacity, BigQuery will queue up slots, and you will not be charged additional fees. For more information about how BigQuery leverages slots for query processing, see [Slots](/bigquery/docs/slots) (/bigquery/docs/slots).

Flat-rate pricing:

- Applies to query costs, including BigQuery ML, DML, and DDL statements.
- Does not apply to costs of [storage](/bigquery/pricing#storage) (/bigquery/pricing#storage), [streaming ingest](/bigquery/pricing#streaming_pricing) (/bigquery/pricing#streaming_pricing), or BI Engine.
- Is purchased as a regional resource. Slot [commitments](/bigquery/docs/reservations-intro#commitments) (/bigquery/docs/reservations-intro#commitments) purchased in one region or multi-region cannot be used in another region or multi-region and cannot be moved.
- Allows customers to raise per-project concurrency quotas by contacting [Google Cloud Support](/support) (/support).
- Is available in per-second, monthly, and annual commitments.


- Can be shared across your entire organization. There is no need to buy slot commitments (/bigquery/docs/reservations-intro#commitments) for every project.
- Has a 100-slot minimum and is purchased in increments of 100 slots.
- Is billed per second for the duration of your commitment.

If you signed an offline agreement for your flat-rate purchase, your plan may deviate from the details described by your sales representative (/contact) to take advantage of Reservations.

Customers using on-demand billing often have 2000 slots or more for query processing. Allocating 100 slots can result in worse query performance.

Monthly flat-rate commitments

The following table shows the cost of your monthly slot commitment.

US (multi-region) 		Monthly
Monthly cost	Number of slots	
000	100	

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Annual flat-rate commitments

The following table shows the cost of your annual slot commitment.

US (multi-region)		Monthly
Hourly cost	Number of slots	

700

100

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Flex slots: short-term commitments

Flex slots are a special commitment type:

- Commitment duration is only 60 seconds.
- You can cancel Flex slots any time thereafter.
- You are charged only for the seconds your commitment was deployed.

The following table shows the cost of your Flex slot commitment.

US (multi-region)		Monthly
Hourly cost	Number of slots	Monthly ¹

700

100

\$2920

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Based on an average of 730 hours per month.

Trial slots (promotion)

Trial slots are available only in the **us** and **eu** multi-regions.

On May 20, 2020, BigQuery introduced a limited promotion for new or returning BigQuery customers. Qualified customers can purchase *Trial slots*, which is a 500-slot, 6-month commitment at a deeply discounted rate.

Trial slots have the following behavior:

- You must commit to a 6-month commitment.
- You cannot cancel for 182 days from the moment of purchase.
- You can only purchase 500 slots.
- You can purchase other commitment types and combine them with Trial slots.
- Trial slots are available only in US and EU multi-regions.
- Trial slots have limited availability and are offered on a first-come, first-serve basis.
- There are no performance or availability differences between Trial slots and other types of slot commitments.

Trial slots are subject to qualification criteria and are available to the following customers:

- Brand new Google Cloud customers signing up for BigQuery
- Existing Google Cloud customers signing up for BigQuery
- Existing BigQuery customers whose spend over the past 3 months has not exceeded \$500 per month
- Customers who sign up using their company email addresses
- Offer only available for purchases directly from Google; not available through resellers or distributors

For more information about how trial slots work, see [Trial slots](#) (/bigquery/docs/reservations-details#trial_slots).

To participate in this promotion, fill out the [BigQuery Trial Slots Promotion form](#) (https://docs.google.com/forms/d/1N83fzcsslndfDV_EV5EZlkkshAX9jWfol0nmeb9X_1fQ/), and we will get back to you in no more than five business days.

US (multi-region) ▼		Monthly
Monthly cost	Number of slots	

10.00 500

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Storage pricing

Once your data is loaded into BigQuery, you are charged for storing it. Storage pricing is based on the amount of data stored in your tables when it is uncompressed.

The size of the data is calculated based on the data types of the individual columns. For a detailed explanation of how data size is calculated, see [Data size calculation \(/bigquery/pricing#data\)](#).

Active storage

Active storage charges are as follows:

US (multi-region) ▼		Monthly
Storage type	Pricing	Details

Active storage \$0.020 per GB The first 10 GB is free each month.

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Storage pricing is prorated per MB, per second. For example, if you store:

- 100 MB for half a month, you pay \$0.001 (a tenth of a cent)
- 500 GB for half a month, you pay \$5
- 1 TB for a full month, you pay \$20

Storage usage is calculated in gigabytes (GB), where 1 GB is 2^{30} bytes. This unit of measurement is also known as (GiB). (<https://en.wikipedia.org/wiki/Gibibyte>). Similarly, 1 TB is 2^{40} bytes (1,024 GB).

Long-term storage

If a table is not edited for 90 consecutive days, the price of storage for that table automatically drops by approximately 50%. There is no degradation of performance, durability, availability, or any other functionality when a table is considered long-term storage.

Each partition of a partitioned table is considered separately for long-term storage pricing. If a partition hasn't been modified in the last 90 days, the data in that partition is considered long term storage and is charged at the discounted price.

Long-term storage pricing is as follows:

US (multi-region)		Monthly
Storage type	Pricing	Details
Long-term storage	\$0.010 per GB	The first 10 GB is free each month.

If the table is edited, the price reverts back to the regular storage pricing, and the 90-day timer starts counting from zero. Anything that modifies the data in a table resets the timer, including:

Action	Details
Loading data into a table	Any load or query job that appends data to a destination table or overwrites a destination table.

Copying data into a table	Any copy job appends data to a destination table or overwrites a destination table.
Writing query results to a table	Any query job that appends data to a destination table or overwrites a destination table.
Using the Data Manipulation Language (DML)	Using a DML statement (/bigquery/docs/reference/standard-sql/data-manipulation-language) to modify table data.
Using Data Definition Language (DDL)	Using a CREATE OR REPLACE TABLE DDL statement (/bigquery/docs/reference/standard-sql/data-definition-language) to replace a table.
Streaming data into the table	Ingesting data using the tabledata.insertAll (/bigquery/docs/reference/rest/v2/tabledata/insertAll) API call.

All other actions do not reset the timer, including the following:

- Querying a table
- Creating a view that queries a table
- Exporting data from a table
- Copying a table (to another destination table)
- Patching or updating a [table resource](#) (/bigquery/docs/reference/rest/v2/tables)

For tables that reach the 90-day threshold during a billing cycle, the price is prorated accordingly.

Long-term storage pricing applies only to BigQuery storage, not to data stored in [external data sources](#) (/bigquery/external-data-sources) such as Cloud Bigtable, Cloud Storage, and Drive.

BigQuery Storage API pricing

The BigQuery Storage API has an on-demand pricing model. You are charged for the data that you read. Customers enrolled in [flat-rate pricing](#) (#flat_rate_pricing) can use the BigQuery Storage API to read up to 300 TB of data per month at no charge. Reads in excess of 300 TB per month are billed at the on-demand rate.

On-demand pricing


Under on-demand pricing, your BigQuery Storage API charges are based on the number of bytes read from BigQuery storage by calls to [ReadRows](#)

(/bigquery/docs/reference/storage/rpc/google.cloud.bigquery.storage.v1beta1#readrowsrequest).

The number of bytes read includes data used for filtering but not returned to you as output from [ReadRows](#). You are not charged for data read from [temporary tables](#)

(/bigquery/docs/cached-results#how_cached_results_are_stored).

On-demand BigQuery Storage API charges are as follows:

US (multi-region) 		Monthly
Pricing	Details	

10 per TB read The BigQuery Storage API is not included in the [Google Cloud Free Tier](#).

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Note the following regarding BigQuery Storage API charges:

- You are charged according to the total amount of data read. The total data read per column is calculated based on the type of data in the column, and the size of the data is calculated based on the column's data type. For a detailed explanation of how data size is calculated, see [Data size calculation](#) (/bigquery/pricing#data).
- You are charged for any data read in a read session even if a [ReadRows](#) call fails.
- If you cancel a [ReadRows](#) call before the end of the stream is reached, you are charged for any data read before the cancellation. Your charges can include data that was read but not returned to you before the cancellation of the [ReadRows](#) call.
- As a best practice, use partitioned and clustered tables whenever possible. You can reduce the amount of data read by using a **WHERE** clause to prune partitions. For more information, see [Querying partitioned tables](#) (/bigquery/docs/querying-partitioned-tables#querying_partitioned_tables_2).

Data size calculation

When you load data into BigQuery or query the data, you're charged according to the data size. Data size is calculated based on the size of each column's data type (</bigquery/docs/reference/standard-sql/data-types>).

The size of your stored data, and the size of the data processed by your queries is calculated in gigabytes (GB), where 1 GB is 2^{30} bytes. This unit of measurement is also known as a gibibyte (GiB). (<https://en.wikipedia.org/wiki/Gibibyte>). Similarly, 1 TB is 2^{40} bytes (1,024 GB).

The size of BigQuery's data types is as follows:

Data type	Size
INT64/INTEGER	8 bytes
FLOAT64/FLOAT	8 bytes
NUMERIC	16 bytes
BOOL/BOOLEAN	1 byte
STRING	2 bytes + the UTF-8 encoded string size
BYTES	2 bytes + the number of bytes in the value
DATE	8 bytes
DATETIME	8 bytes
TIME	8 bytes
TIMESTAMP	8 bytes
STRUCT/RECORD	0 bytes + the size of the contained fields
GEOGRAPHY	16 bytes + 24 bytes * the number of vertices in the geography type (you can verify the number of vertices using the <u>ST_NumPoints</u> (/bigquery/docs/reference/standard-sql/geography_functions#st_numpoints) function)

Null values for any data type are calculated as 0 bytes.

A repeated column is stored as an array, and the size is calculated based on the number of values. For example, an integer column (INT64) that is repeated (ARRAY<INT64>) and contains 4

entries is calculated as 32 bytes (4 entries x 8 bytes).

Streaming pricing

Loading data into BigQuery is free, with the exception of a small charge for streamed data (/bigquery/streaming-data-into-bigquery).

Pricing for streaming inserts is as follows:

US (multi-region) ▼		Monthly
Operation	Pricing	Details
Streaming inserts	\$0.010 per 200 MB	You are charged for rows that are successfully inserted. Individual rows are calculated using a 1 KB minimum size.

Data Manipulation Language pricing

BigQuery charges for DML queries based on the number of bytes processed by the query.

DML pricing for non-partitioned tables

For non-partitioned tables, the number of bytes processed is calculated as follows:

DML statement	Bytes processed
INSERT	The sum of bytes processed for all the columns referenced from the tables scanned by the query.
UPDATE	The sum of bytes in all the columns referenced from the tables scanned by the query + the sum of bytes for all columns in the updated table at the time the UPDATE starts.

DML statement	Bytes processed
DELETE	The sum of bytes in all the columns referenced from the tables scanned by the query + the sum of bytes for all columns in the modified table at the time the DELETE starts.
MERGE	If there are only INSERT clauses in the MERGE statement, you are charged for the sum of bytes processed for all the columns referenced in all tables scanned by the query. If there is an UPDATE or DELETE clause in the MERGE statement, you are charged for the sum of the bytes processed for all the columns referenced in the source tables scanned by the query + the sum of bytes for all columns in the target table (at the time the MERGE starts).

DML pricing for partitioned tables

For partitioned tables, the number of bytes processed is calculated as follows:

DML statement	Bytes processed
INSERT	The sum of bytes processed for all the columns referenced in all partitions scanned by the query.
UPDATE	The sum of bytes processed for all the columns referenced in all partitions for the tables scanned by the query + the sum of bytes for all columns in the updated or scanned partitions for the table being updated (at the time the UPDATE starts).
DELETE	The sum of bytes processed for all the columns referenced in all partitions for the tables scanned by the query + the sum of bytes for all columns in the modified or scanned partitions for the table being modified (at the time the DELETE starts).
MERGE	If there are only INSERT clauses in the MERGE statement, you are charged for the sum of bytes processed for all the columns referenced in all partitions scanned by the query. If there is an UPDATE or DELETE clause in the MERGE statement, you are charged for the sum of the bytes processed for all the columns referenced in all partitions for the source tables scanned by the query + the sum of bytes for all columns in the updated, deleted or scanned partitions for the target table (at the time the MERGE starts).

Data Definition Language pricing

BigQuery charges for DDL queries based on the number of bytes processed by the query. The number of bytes processed is calculated as follows for DDL statements:

DDL statement	Bytes processed
CREATE TABLE	None.
CREATE TABLE ... AS SELECT ...	The sum of bytes processed for all the columns referenced from the tables scanned by the query.
CREATE VIEW	None.
DROP TABLE	None.
DROP VIEW	None.

Clustered table pricing

When you create and use clustered tables in BigQuery, your charges are based on how much data is stored (#storage) in the tables and on the queries (#queries) you run against the data. Clustered tables help you to reduce query costs by pruning data so it is not processed by the query. This process is called block pruning.

Block pruning

BigQuery sorts the data in a clustered table based on the values in the clustering columns and organizes them into blocks.

When you submit a query that contains a filter on a clustered column, BigQuery uses the clustering information to efficiently determine whether a block contains any data relevant to the query. This allows BigQuery to only scan the relevant blocks—a process referred to as block pruning.

Query pricing is based on the number of bytes processed. When you run a query against a clustered table, and the query includes a filter on the clustered columns, BigQuery uses the filter expression and the block metadata to prune the blocks scanned by the query.

When a block is pruned, it is not scanned. Only the scanned blocks are used to calculate the bytes of data processed by the query. The number of bytes processed by a query against a

clustered table equals the sum of the bytes read in each column referenced by the query in the scanned blocks.

If a clustered table is referenced multiple times in a query that uses several filters, BigQuery charges for scanning the columns in the appropriate blocks in each of the respective filters.

Scripting pricing

During the BigQuery [scripting](/bigquery/docs/reference/standard-sql/scripting) beta, the BigQuery team recommends using projects with flat-rate reservations to avoid unintended query costs because the number of bytes scanned by a script is generally not known before executing it. Alternatively, you can use the BigQuery [sandbox](/bigquery/docs/sandbox) to take advantage of limited free script execution. The BigQuery team will provide more explicit control over total bytes scanned by scripts and individual statements within scripts over time. This is a beta release; for updates to pricing, see the BigQuery [release notes](/bigquery/docs/release-notes).

If a script fails, the cost of any statements up until the failure still applies. The statement that failed does not incur any costs.

For publicly released statement types such as `SELECT`, `INSERT`, and `UPDATE`, the cost of executing the statement is as described in the public pricing documentation. For scripting-specific statement types, the following pricing applies:

- **DECLARE**: the sum of bytes scanned for any tables referenced in the `DEFAULT` expression. `DECLARE` statements with no table references do not incur a cost.
- **SET**: the sum of bytes scanned for any tables referenced in the expression. `SET` statements with no table references do not incur a cost.
- **IF**: the sum of bytes scanned for any tables referenced in the condition expression. `IF` condition expressions with no table reference do not incur a cost. Any statements within the `IF` block that are not executed do not incur a cost.
- **WHILE**: the sum of bytes scanned for any tables referenced in the condition expression. `WHILE` statements with no table references in the condition expression do not incur a cost. Any statements within the `WHILE` block that are not executed do not incur a cost.
- **CONTINUE** or **ITERATE**: No associated cost.
- **BREAK** or **LEAVE**: No associated cost.

- **BEGIN** or **END**: No associated cost.

Temporary tables do not incur charges for storage while the script is running. However, regular pricing occurs for any statements that create, modify, or query them.

BigQuery pricing examples

Estimating query costs

For query pricing examples, see [Estimating query costs](/bigquery/docs/estimate-costs#estimating_query_costs) (/bigquery/docs/estimate-costs#estimating_query_costs).

Estimating storage costs

For storage pricing examples, see [Estimating storage costs](/bigquery/docs/estimate-costs#estimating_storage_costs_using_the_pricing_calculator) (/bigquery/docs/estimate-costs#estimating_storage_costs_using_the_pricing_calculator).

DML pricing examples for non-partitioned tables

The following examples demonstrate how BigQuery calculates bytes read for DML statements that modify non-partitioned tables.

Example 1: Non-partitioned table UPDATE

table1 has two columns: col1 of type INTEGER and col2 of type STRING.

```
E table1 SET col1 = 1 WHERE col2 = 2;
```

Bytes processed in this example =

- sum of the number of bytes in col1 +
- sum of the number of bytes in col2

Example 2: Non-partitioned table UPDATE

`table1` has two columns: `col1` of type `INTEGER` and `col2` of type `STRING`. `table2` has one column: `field1` of type `INTEGER`.

```
UPDATE table1 SET col1 = 1 WHERE col1 in (SELECT field1 from table2)
```

Bytes processed in this example =

- sum of the number of bytes in `table1.col1` before `UPDATE` +
- sum of the number of bytes in `table1.col2` before `UPDATE` +
- sum of the number of bytes in `table2.field1`

DML pricing examples for partitioned tables

The following examples demonstrate how BigQuery calculates bytes read for DML statements that modify ingestion-time and partitioned tables. To view the JSON schema representations for the tables used in the examples, see [Tables used in examples \(#tables_used_in_examples\)](#) on the Updating Partitioned Table Data Using DML Statements page.

Example 1: Ingestion-time partitioned table INSERT

`mytable2` has two columns: `id` of type `INTEGER` and `ts` of type `TIMESTAMP`. `mytable` has two columns: `field1` of type `INTEGER` and `field2` of type `STRING`.

```
INSERT INTO mytable (_PARTITIONTIME, field1) AS SELECT TIMESTAMP(DATE(ts)), id from mytable2
```

Bytes processed in this example =

- sum of the number of bytes in `mytable2.ts` +
- sum of the number of bytes in `mytable2.id`

The size of table into which the rows are inserted—`mytable`—does not affect the cost of the query.

Example 2: Partitioned table INSERT

`mytable2` has two columns: `id` of type `INTEGER` and `ts` of type `TIMESTAMP`. `mycolumnstable` has four columns: `field1` of type `INTEGER`, `field2` of type `STRING`, `field3` of type `BOOLEAN`, and `ts` of type `TIMESTAMP`.

```
T INTO mycolumnstable (ts, field1) AS SELECT ts, id from mytable2
```

Bytes processed in this example =

- sum of the number of bytes in `mytable2.ts` +
- sum of the number of bytes in `mytable2.id`

The size of table into which the rows are inserted—`mycolumnstable`—does not affect the cost of the query.

Example 3: Ingestion-time partitioned table UPDATE

DML statement 1: Updating a single partition

`mytable2` has two columns: `id` of type `INTEGER` and `ts` of type `TIMESTAMP`. `mytable` has two columns: `field1` of type `INTEGER` and `field2` of type `STRING`.

```
E project.mydataset.mytable T SET T.field1 = T.field1 + 100 WHERE T._PARTITIONTIME =
```

Bytes processed in this example =

- sum of the number of bytes in `mytable2.id` +
- sum of the number of bytes in `mytable.field1` in the "2017-05-01" partition +
- sum of the number of bytes in `mytable.field2` in the "2017-05-01" partition

DML statement 2: Updating a partition based on another partition in the table

```
E project.mydataset.mytable T SET T._PARTITIONTIME = TIMESTAMP("2017-06-01"), T.fiel
```

Bytes processed in this example =

- sum of the number of bytes in `mytable.field1` in the "2017-05-01" partition +
- sum of the number of bytes in `mytable.field2` in the "2017-05-01" partition +
- sum of the number of bytes in `mytable.field1` in the "2017-06-01" partition +
- sum of the number of bytes in `mytable.field2` in the "2017-06-01" partition

In this case, the cost of the UPDATE statement is the sum of sizes of all fields in the partitions corresponding to "2017-05-01" and "2017-06-01".

Example 4: Partitioned table UPDATE

DML statement 1: Updating a single partition

`mytable2` has two columns: `id` of type `INTEGER` and `ts` of type `TIMESTAMP`. `mycolumnntable` has four columns: `field1` of type `INTEGER`, `field2` of type `STRING`, `field3` of type `BOOLEAN`, and `ts` of type `TIMESTAMP`.

```
E project.mydataset.mycolumntable T SET T.field1 = T.field1 + 100 WHERE DATE(T.ts) =
```

Bytes processed in this example =

- sum of the number of bytes in `mytable2.id` +
- sum of the number of bytes in `mycolumnntable.field1` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnntable.field2` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnntable.field3` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnntable.ts` in the "2017-05-01" partition

DML statement 2: Updating a partition based on another partition in the table

```
E project.mydataset.mycolumntable T SET T.ts = TIMESTAMP("2017-06-01"), T.field1 = T
```

Bytes processed in this example =

- sum of the number of bytes in `mycolumnntable.field1` in the "2017-05-01" partition +

- sum of the number of bytes in `mycolumnstable.field2` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnstable.field3` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnstable.ts` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnstable.field1` in the "2017-06-01" partition +
- sum of the number of bytes in `mycolumnstable.field2` in the "2017-06-01" partition +
- sum of the number of bytes in `mycolumnstable.field3` in the "2017-06-01" partition +
- sum of the number of bytes in `mycolumnstable.ts` in the "2017-06-01" partition

In this case, the cost of the UPDATE statement is the sum of sizes of all fields in the partitions corresponding to "2017-05-01" and "2017-06-01".

Example 5: Ingestion-time partitioned table DELETE

`mytable2` has two columns: `id` of type `INTEGER` and `ts` of type `TIMESTAMP`. `mytable` has two columns: `field1` of type `INTEGER` and `field2` of type `STRING`.

```
E project.mydataset.mytable T WHERE T._PARTITIONTIME = TIMESTAMP("2017-05-01") AND E
```

Bytes processed in this example =

- sum of the number of bytes in `mytable2.id` +
- sum of the number of bytes in `mytable.field1` in the "2017-05-01" partition +
- sum of the number of bytes in `mytable.field2` in the "2017-05-01" partition

Example 6: Partitioned table DELETE

`mytable2` has two columns: `id` of type `INTEGER` and `ts` of type `TIMESTAMP`. `mycolumnstable` has four columns: `field1` of type `INTEGER`, `field2` of type `STRING`, `field3` of type `BOOLEAN`, and `ts` of type `TIMESTAMP`.

```
E project.mydataset.mycolumnstable T WHERE DATE(T.ts) = "2017-05-01" AND EXISTS (SELEC
```

Bytes processed in this example =

- sum of the number of bytes in `mytable2.id` +
- sum of the number of bytes in `mycolumnstable.field1` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnstable.field2` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnstable.field3` in the "2017-05-01" partition +
- sum of the number of bytes in `mycolumnstable.ts` in the "2017-05-01" partition

Clustered table pricing example

You have a clustered table named `ClusteredSalesData`. The table is partitioned by the `timestamp` column, and it is clustered by the `customer_id` column. The data is organized into the following set of blocks:

Partition identifier	Block ID	Minimum value for <code>customer_id</code> in the block	Maximum value for <code>customer_id</code> in the block
20160501	B1	10000	19999
20160501	B2	20000	24999
20160502	B3	15000	17999
20160501	B4	22000	27999

You run the following query against the table. The query contains a filter on the `customer_id` column.

```

T
(totalSale)

dataset.ClusteredSalesData`

tomer_id BETWEEN 20000
23000
DATE(timestamp) = "2016-05-01"

```

This query:

- Scans the `timestamp`, `customer_id`, and `totalSale` columns in blocks B2 and B4.
- Prunes the B3 block because of the `DATE(timestamp) = "2016-05-01"` filter predicate on the `timestamp` partitioning column.
- Prunes the B1 block because of the `customer_id BETWEEN 20000 AND 23000` filter predicate on the `customer_id` clustering column.

Scripting pricing example

The following example script contains comments above every statement that explain what cost, if any, is incurred by the following statement.

```

cost, since no tables are referenced.
RE x DATE DEFAULT CURRENT_DATE();
curs the cost of scanning string_col from dataset.table.
RE y STRING DEFAULT (SELECT MAX(string_col) FROM dataset.table);
curs the cost of copying the data from dataset.big_table. Once the
ble is created, you are not charged for storage while the rest of the
ript runs.
E TEMP TABLE t AS SELECT * FROM dataset.big_table;
curs the cost of scanning column1 from temporary table t.
T column1 FROM t;
cost, since y = 'foo' doesn't reference a table.
= 'foo' THEN
Incurs the cost of scanning all columns from dataset.other_table, if
y was equal to 'foo', or otherwise no cost since it is not executed.
ECT * FROM dataset.other_table;

Incurs the cost of scanning all columns from dataset.different_table, if
y was not equal to 'foo', or otherwise no cost since it is not executed.
ATE dataset.different_table
col = 10
RE true;
F;
curs the cost of scanning date_col from dataset.table for each
eration of the loop.
x < (SELECT MIN(date_col) FROM dataset.table) DO
No cost, since the expression does not reference any tables.
x = DATE_ADD(x, INTERVAL 1 DAY);
No cost, since the expression does not reference any tables.
true THEN
- LEAVE has no associated cost.

```

```
EAVE;  
  IF;  
Never executed, since the IF branch is always taken, so does not incur  
a cost.  
ECT * FROM dataset.big_table;  
HILE;
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-08-13 UTC.