

Getting streaming metadata using INFORMATION_SCHEMA

ature is covered by the [Pre-GA Offerings Terms \(/terms/service-terms#1\)](/terms/service-terms#1) of the Google Cloud Platform Terms & Conditions. Pre-GA features may have limited support, and changes to pre-GA features may not be compatible with other products. For more information, see the [launch stage descriptions \(/products#product-launch-stages\)](/products#product-launch-stages).

INFORMATION_SCHEMA is a series of views that provide access to metadata about datasets, routines, tables, views, jobs, reservations, and streaming data.

You can query the **INFORMATION_SCHEMA** streaming views to retrieve historical and real-time information about streaming data into BigQuery. These views contain per minute aggregated statistics for each table that have data streamed into them. For more information about streaming data into BigQuery, see [Streaming data into BigQuery](/bigquery/streaming-data-into-bigquery).

(/bigquery/streaming-data-into-bigquery).

Required permissions

Retrieving streaming metadata by using **INFORMATION_SCHEMA** tables requires appropriately scoped permissions:

- **STREAMING_TIMELINE_BY_PROJECT** requires `bigquery.tables.list` for the project and is available to the `BigQuery User`, `BigQuery Data Viewer`, `BigQuery Data Editor`, `BigQuery Data Owner`, `BigQuery Metadata Viewer`, `BigQuery Resource Admin`, and `BigQuery Admin` roles.
- **STREAMING_TIMELINE_BY_ORGANIZATION** requires `bigquery.tables.list` for the organization and is available to the `BigQuery User`, `BigQuery Data Viewer`, `BigQuery Data Editor`, `BigQuery Data Owner`, `BigQuery Metadata Viewer`, `BigQuery Resource Admin`, and `BigQuery Admin` roles.

Schemas

When you query the `INFORMATION_SCHEMA` streaming views, the query results contain historical and real-time information about streaming data into BigQuery. Each row in the following views represents statistics for streaming into a specific table, aggregated over a one minute interval starting at `start_timestamp`. Statistics are grouped by error code, so there will be one row for each error code encountered during the one minute interval for each timestamp and table combination. Successful requests have the error code set to `NULL`. If no data was streamed into a table during a certain time period, then no rows are present for the corresponding timestamps for that table.

- `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT` contains per-minute aggregated streaming statistics for the current project.
- `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_ORGANIZATION` contains per-minute aggregated streaming statistics for the whole organization associated with the current project.

The `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT` and `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_ORGANIZATION` views have the following schema:

Column name	Data type	Value
<code>start_timestamp</code>	<code>TIMESTAMP</code>	<i>(Partitioning column)</i> Start timestamp of the 1 minute interval for the aggregated statistics.
<code>project_id</code>	<code>STRING</code>	<i>(Clustering column)</i> ID of the project.
<code>project_number</code>	<code>INTEGER</code>	Number of the project.
<code>dataset_id</code>	<code>STRING</code>	<i>(Clustering column)</i> ID of the dataset.
<code>table_id</code>	<code>STRING</code>	<i>(Clustering column)</i> ID of the table.
<code>error_code</code>	<code>STRING</code>	Error code returned for the requests specified by this row. <code>NULL</code> for successful requests.
<code>total_requests</code>	<code>INTEGER</code>	Total number of requests within the 1 minute interval.
<code>total_rows</code>	<code>INTEGER</code>	Total number of rows from all requests within the 1 minute interval.
<code>total_input_bytes</code>	<code>INTEGER</code>	Total number of bytes from all rows within the 1 minute interval.

Data retention

Currently, only the last 180 days of streaming history is retained in the `INFORMATION_SCHEMA` streaming views.

Regionality

BigQuery's `INFORMATION_SCHEMA` streaming views are regionalized. To query these views, you must prefix a supported region name in the format ``region-region-name``. `INFORMATION_SCHEMA.view`.

For example:

- To query data in the US multi-region, use ``region-us``. `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT`
- To query data in the EU multi-region, use ``region-eu``. `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT`
- To query data in the asia-northeast1 region, use ``region-asia-northeast1``. `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT`

For a list of available regions, see [Dataset locations](/bigquery/docs/locations) (/bigquery/docs/locations).

Examples

Example 1: Recent streaming failures

The following example calculates the per minute breakdown of total failed requests for all tables in the project in the last 30 minutes, split by error code.

To run the query against a project other than your default project, add the project ID in the following format: ``project_id`.`region-region-name``. `INFORMATION_SCHEMA.view`, for example, ``myproject`.`region-us``. `INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT`.

To run the query:

1. In the Cloud Console, open the BigQuery web UI.

Go to the Cloud Console (<https://console.cloud.google.com/bigquery>)

2. In the **Query editor** box, enter the following standard SQL query. **INFORMATION_SCHEMA** requires standard SQL syntax. Standard SQL is the default syntax in the Cloud Console.

```
SELECT
  start_timestamp,
  error_code,
  SUM(total_requests) AS num_failed_requests
FROM
  `region-us`.INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT
WHERE
  error_code IS NOT NULL
  AND start_timestamp > TIMESTAMP_SUB(CURRENT_TIMESTAMP, INTERVAL 30 MINUT
GROUP BY
  start_timestamp,
  error_code
ORDER BY
  1 DESC
```

★ **Note:** **INFORMATION_SCHEMA** view names are case-sensitive.

3. Click **Run**.

The results should look like the following:

start_timestamp	error_code	num_failed_requests
020-04-15 20:55:00	INTERNAL_ERROR	41
020-04-15 20:41:00	CONNECTION_ERROR	5
020-04-15 20:30:00	INTERNAL_ERROR	115

Example 2: Per minute breakdown for all requests with error codes

The following example calculates a per minute breakdown of successful and failed streaming requests, split into error code categories. This query could be used to populate a dashboard.

To run the query against a project other than your default project, add the project ID in the following format: ``project_id`.`region-region_name`.INFORMATION_SCHEMA.view`, for example, ``myproject`.`region-us`.INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT`.

To run the query:

ConsoleCLI (#cli)

1. In the Cloud Console, open the BigQuery web UI.

[Go to the Cloud Console \(https://console.cloud.google.com/bigquery\)](https://console.cloud.google.com/bigquery)

2. Enter the following standard SQL query in the **Query editor** box. `INFORMATION_SCHEMA` requires standard SQL syntax. Standard SQL is the default syntax in the Cloud Console.

```
SELECT
  start_timestamp,
  SUM(total_requests) AS total_requests,
  SUM(total_rows) AS total_rows,
  SUM(total_input_bytes) AS total_input_bytes,
  SUM(IF(error_code IN ('QUOTA_EXCEEDED', 'RATE_LIMIT_EXCEEDED'),
    total_requests, 0)) AS quota_error,
  SUM(IF(error_code IN ('INVALID_VALUE', 'NOT_FOUND', 'SCHEMA_INCOMPATIBLE',
    'BILLING_NOT_ENABLED', 'ACCESS_DENIED', 'UNAUTHENT
    total_requests, 0)) AS user_error,
  SUM(IF(error_code IN ('CONNECTION_ERROR', 'INTERNAL_ERROR'),
    total_requests, 0)) AS server_error,
  SUM(IF(error_code IS NULL, 0, total_requests)) AS total_error,
FROM
  `region-us`.INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT
GROUP BY
  start_timestamp
ORDER BY
  1 DESC
```

★ **Note:** `INFORMATION_SCHEMA` view names are case-sensitive.

3. Click **Run**.

The results should look like the following:

start_timestamp	total_requests	total_rows	total_input_bytes	quota_error	u
0-04-15 22:00:00	441854	441854	23784853118	0	
0-04-15 21:59:00	355627	355627	26101982742	0	
0-04-15 21:58:00	354603	354603	26160565341	0	
0-04-15 21:57:00	298823	298823	23877821442	0	

Example 3: Tables with the most incoming traffic

The following example returns the streaming statistics for the 10 tables with the most incoming traffic.

To run the query against a project other than your default project, add the project ID in the following format: ``project_id`.`region-region_name`.INFORMATION_SCHEMA.view`, for example, ``myproject`.`region-us`.INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT`.

To run the query:

ConsoleCLI (#cli)

1. In the Cloud Console, open the BigQuery web UI.

[Go to the Cloud Console \(https://console.cloud.google.com/bigquery\)](https://console.cloud.google.com/bigquery)

2. In the **Query editor** box, enter the following standard SQL query. `INFORMATION_SCHEMA` requires standard SQL syntax. Standard SQL is the default syntax in the Cloud Console.

```

SELECT
  project_id,
  dataset_id,
  table_id,
  SUM(total_rows) AS num_rows,
  SUM(total_input_bytes) AS num_bytes,
  SUM(total_requests) AS num_requests
FROM
  `region-us`.INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT
GROUP BY 1, 2, 3
ORDER BY num_bytes DESC
LIMIT 10

```

★ **Note:** INFORMATION_SCHEMA view names are case-sensitive.

3. Click **Run**.

The results should look like the following:

project_id	dataset_id	table_id	num_rows	num_bytes
y-project	dataset1	table1	8016725532	7378
y-project	dataset1	table2	26319580	3419
y-project	dataset2	table1	38355294	2287
y-project	dataset1	table3	270126906	1759
y-project	dataset2	table2	95511309	1737
y-project	dataset2	table3	46500443	1283
y-project	dataset2	table4	25846270	748
y-project	dataset1	table4	18318404	566
y-project	dataset1	table5	42829431	534
y-project	dataset1	table6	8771021	511

Example 4: Streaming error ratio for a table

The following example calculates a per-day breakdown of errors for a specific table, split by error code.

To run the query against a project other than your default project, add the project ID in the following format: ``project_id`.`region-region_name`.INFORMATION_SCHEMA.view`, for example, ``myproject`.`region-us`.INFORMATION_SCHEMA.JOBS_BY_PROJECT`.

To run the query:

ConsoleCLI (#cli)

1. In the Cloud Console, open the BigQuery web UI.

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

2. In the **Query editor** box, enter the following standard SQL query. `INFORMATION_SCHEMA` requires standard SQL syntax. Standard SQL is the default syntax in the Cloud Console.

```
SELECT
  TIMESTAMP_TRUNC(start_timestamp, DAY) as day,
  project_id,
  dataset_id,
  table_id,
  error_code,
  SUM(total_rows) AS num_rows,
  SUM(total_input_bytes) AS num_bytes,
  SUM(total_requests) AS num_requests
FROM
  `region-us`.INFORMATION_SCHEMA.STREAMING_TIMELINE_BY_PROJECT
WHERE table_id LIKE "my_table"
GROUP BY project_id, dataset_id, table_id, error_code, day
ORDER BY day, project_id, dataset_id DESC
```

★ **Note:** `INFORMATION_SCHEMA` view names are case-sensitive.

3. Click **Run**.

The results should look like the following:

day	project_id	dataset_id	table_id	error_code	num_rows
0-04-21 00:00:00	my_project	my_dataset	my_table	NULL	41
0-04-20 00:00:00	my_project	my_dataset	my_table	NULL	2798
0-04-19 00:00:00	my_project	my_dataset	my_table	NULL	2005
0-04-18 00:00:00	my_project	my_dataset	my_table	NULL	2054
0-04-17 00:00:00	my_project	my_dataset	my_table	NULL	2056
0-04-17 00:00:00	my_project	my_dataset	my_table	INTERNAL_ERROR	4

Next steps

- For an overview of `INFORMATION_SCHEMA`, see [Introduction to BigQuery INFORMATION_SCHEMA](#) (/bigquery/docs/information-schema-intro).
- Learn how to use `INFORMATION_SCHEMA` to get [job metadata](#) (/bigquery/docs/information-schema-jobs).
- Learn how to use `INFORMATION_SCHEMA` to get [reservations metadata](#) (/bigquery/docs/information-schema-reservations).
- Learn how to use `INFORMATION_SCHEMA` to get [dataset metadata](#) (/bigquery/docs/information-schema-datasets).
- Learn how to use `INFORMATION_SCHEMA` to get [table metadata](#) (/bigquery/docs/information-schema-tables).
- Learn how to use `INFORMATION_SCHEMA` to get [view metadata](#) (/bigquery/docs/information-schema-views).

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/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](#) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-07-31 UTC.