Data Analytics
Case Study

(using BigQuery SQL)

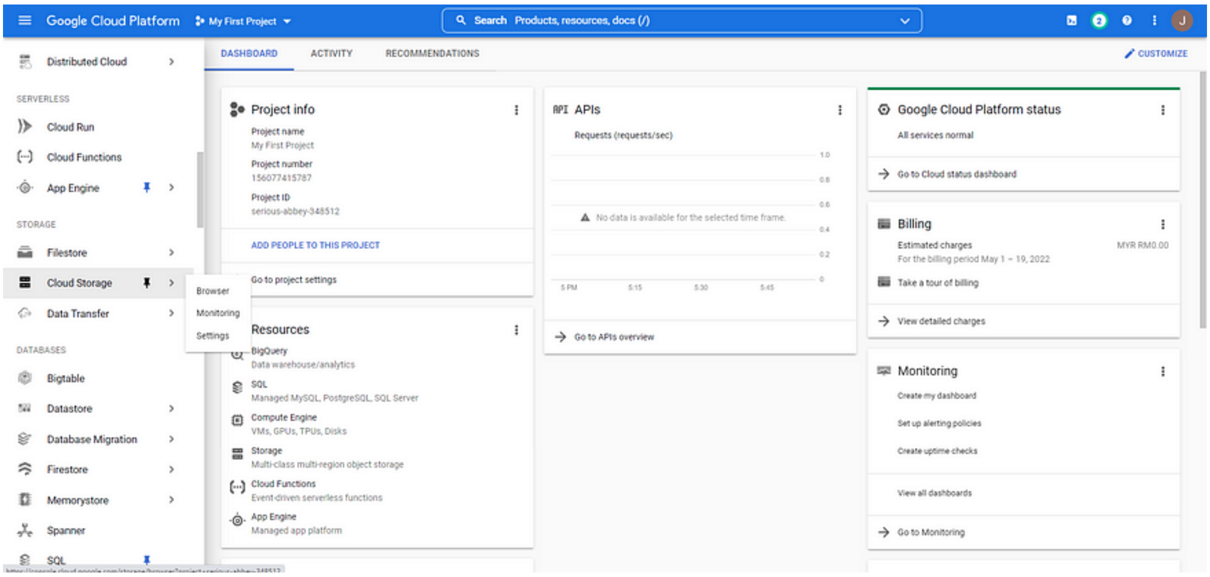
This separate article would document how I used BigQuery SQL instead of RStudio for the analysis process of Prepare & Process.

As a side note, I will not be re-explaining the context of the scenario again, so if you haven’t read my original post already, I highly recommend doing so over here!

The first section of the article would be dedicated to how I uploaded the datasets for use in BigQuery’s system, as so far, the majority of the other students that enrolled in this course have had trouble doing so. Hopefully, this will serve as a guide as well!

The rest of the article would be documenting how I used BigQuery SQL to process the data.

Prepare - Uploading datasets



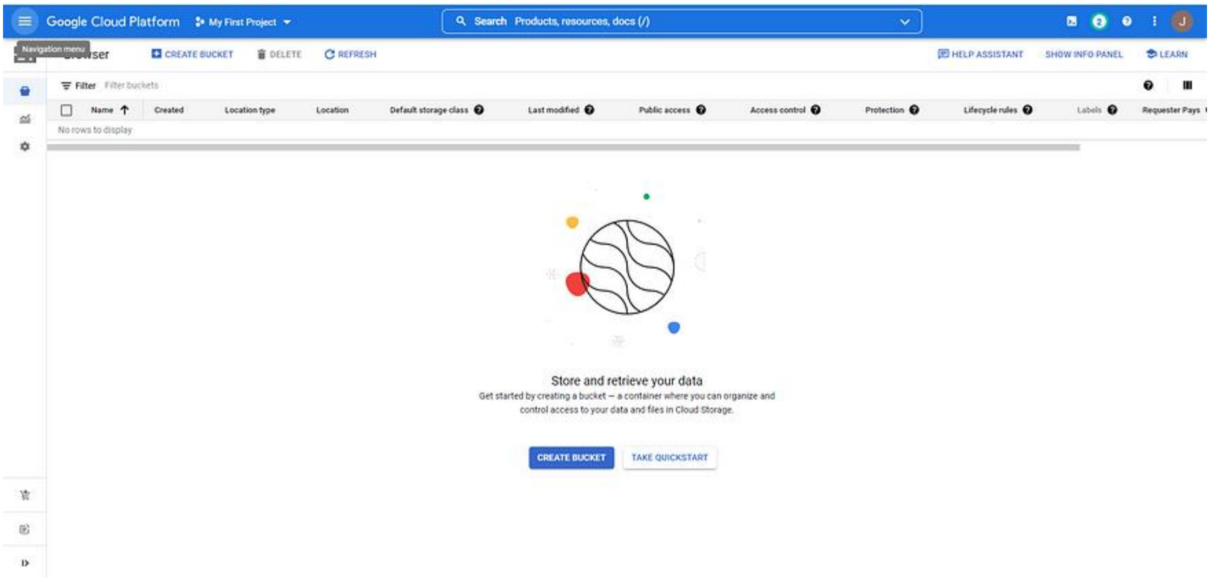
Project name
My First Project
Project number
156077415787
Project ID
serious-abbey-348512

No data is available for the selected time frame.

All services normal

Estimated charges MYR RM0.00
For the billing period May 1 – 19, 2022

First, log in to Google Cloud Platform and click the navigation menu on the top left, scroll down till you see the ‘Storage’ section, and press on ‘Cloud Storage’.



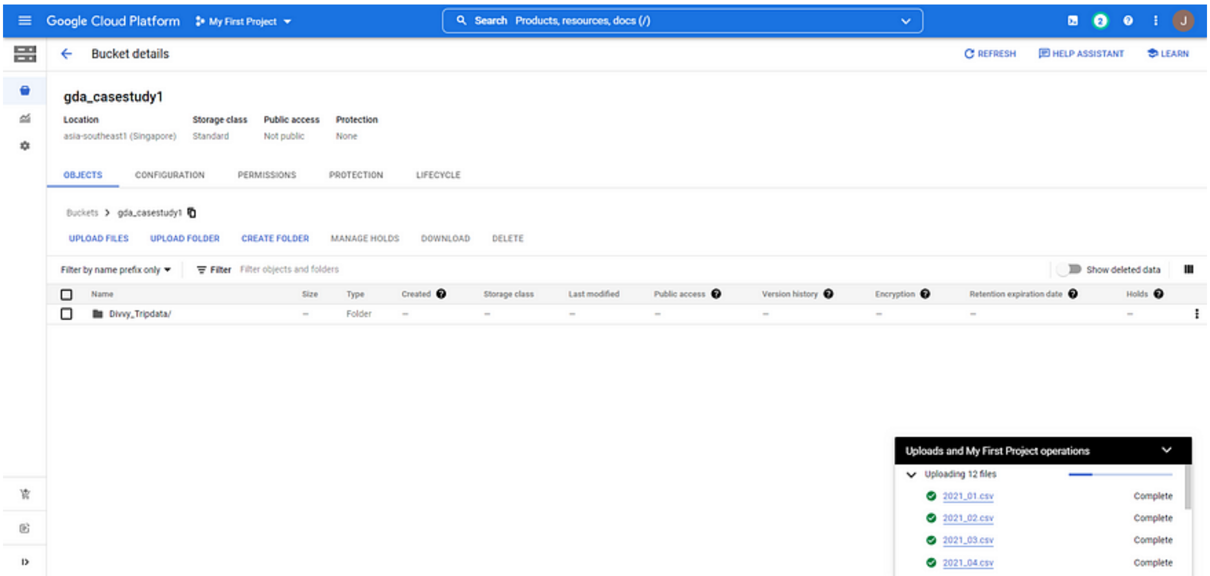
No rows to display

Store and retrieve your data
Get started by creating a bucket — a container where you can organize and control access to your data and files in Cloud Storage.

Next, create your bucket and give it a name as well.

Now we need to choose where to store our data.

Select 'Region — lowest latency within a single region' and select the region you're the closest to. Proceed with the basic settings for the rest and create your bucket.



gda_casestudy1

Location	Storage class	Public access	Protection
asia-southeast1 (Singapore)	Standard	Not public	None

□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration date	Holds
□	Divvy_Tripdata/	–	Folder	–	–	–	–	–	–	–	–

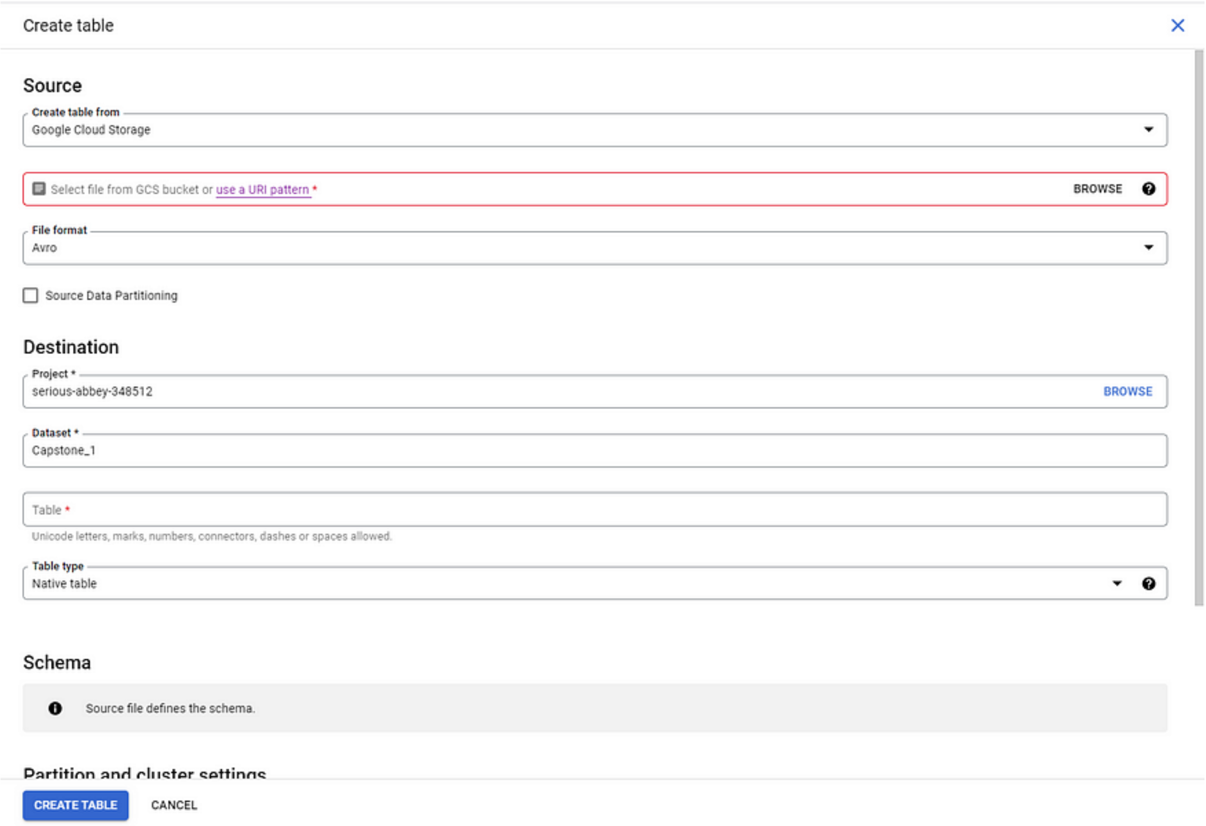
Uploads and My First Project operations

Uploading 12 files

2021_01.csv	Complete
2021_02.csv	Complete
2021_03.csv	Complete
2021_04.csv	Complete

Now, upload the files/datasets that you've obtained from the pdf . I would recommend uploading/creating a folder first so that you can organize your data.

Accessing datasets



Create table

Source

Create table from: Google Cloud Storage

Select file from GCS bucket or use a URI pattern *

File format: Avro

□ Source Data Partitioning

Destination

Project *: serious-abbey-348512

Dataset *: Capstone_1

Table *

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

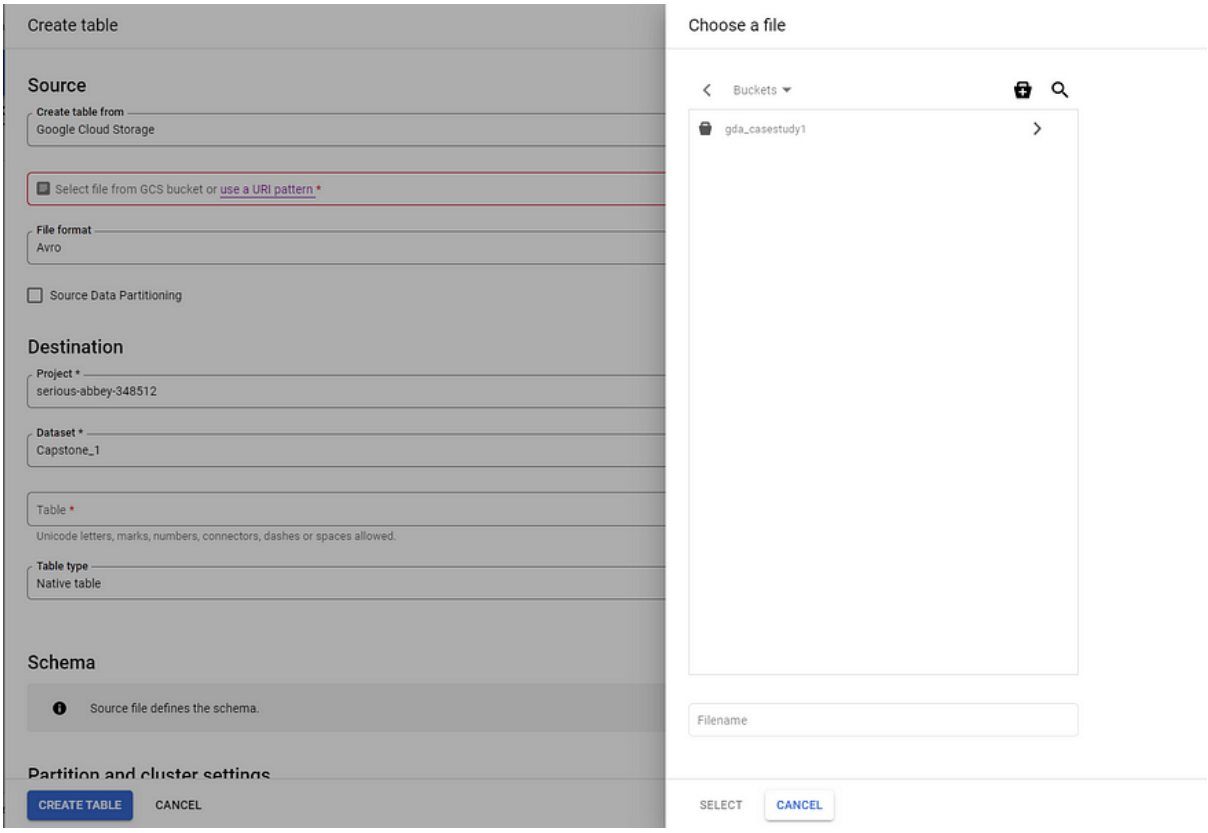
Table type: Native table

Schema

Source file defines the schema.

Partition and cluster settings

Now, head over to your BigQuery dashboard/homepage and create a table. Select the Google Cloud Storage option when you press the dropdown menu from ‘Create table from’


Create table

Source

Create table from: Google Cloud Storage

□ Select file from GCS bucket or use a URI pattern *

File format: Avro

□ Source Data Partitioning

Destination

Project *: serious-abbey-348512

Dataset *: Capstone_1

Table *

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

Table type: Native table

Schema

Source file defines the schema.

Partition and cluster settings

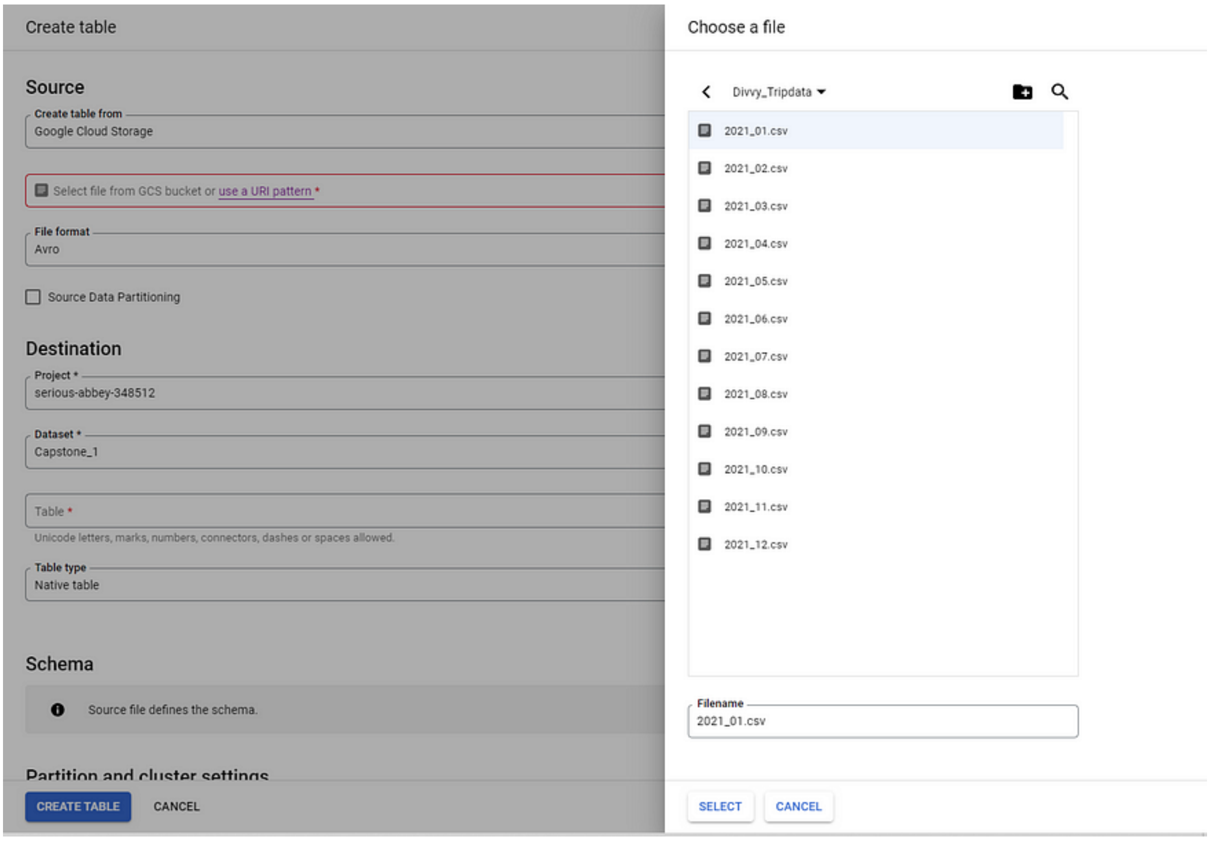
Choose a file

Buckets

gda_casestudy1

Filename

gda_casestudy1 is my folder name and the file are contained inside



Create table

Source

Create table from: Google Cloud Storage

☐ Select file from GCS bucket or use a URI pattern *

File format: Avro

☐ Source Data Partitioning

Destination

Project *: serious-abbey-348512

Dataset *: Capstone_1

Table *

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

Table type: Native table

Schema

Source file defines the schema.

Partition and cluster settings

Choose a file

Divvy_Tripdata

- 2021_01.csv
- 2021_02.csv
- 2021_03.csv
- 2021_04.csv
- 2021_05.csv
- 2021_06.csv
- 2021_07.csv
- 2021_08.csv
- 2021_09.csv
- 2021_10.csv
- 2021_11.csv
- 2021_12.csv

Filename: 2021_01.csv

Import the files one by one into your table (remember to name them as well) and make sure to enable ‘Auto Detect Schema’

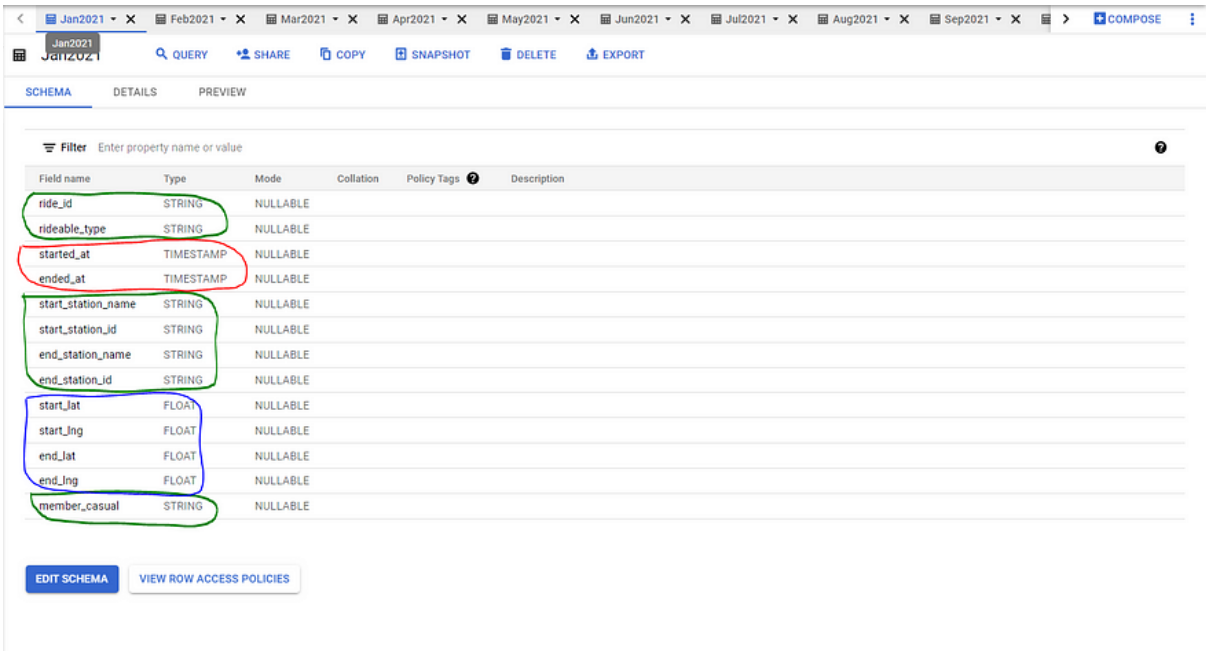
This whole process will take about 10 minutes.

Now that we’ve imported our datasets, it's time to move on to the Prepare phase!

Process

First, let's check the schema of all the tables that we’ve imported. We should check if:

- The format of each field is identical
- The naming of each field is identical



Field name	Type	Mode	Collation	Policy Tags	Description
ride_id	STRING	NULLABLE			
rideable_type	STRING	NULLABLE			
started_at	TIMESTAMP	NULLABLE			
ended_at	TIMESTAMP	NULLABLE			
start_station_name	STRING	NULLABLE			
start_station_id	STRING	NULLABLE			
end_station_name	STRING	NULLABLE			
end_station_id	STRING	NULLABLE			
start_lat	FLOAT	NULLABLE			
start_lng	FLOAT	NULLABLE			
end_lat	FLOAT	NULLABLE			
end_lng	FLOAT	NULLABLE			
member_casual	STRING	NULLABLE			

After confirming its identical, we will merge all the tables together into one dataset(which we will be calling dataframe from now on) by using UNION ALL. As to why we’re using this instead of JOIN, joins will combine data into new columns, which means in our final dataset, we would have ride_id,ride_id2,ride_id3, and so on.

Unions, on the other hand, will combine new data into new rows while staying in the same column (given that the column names are identical).

more information regarding unions can be found here.

The syntax for UNION is shown as the following:

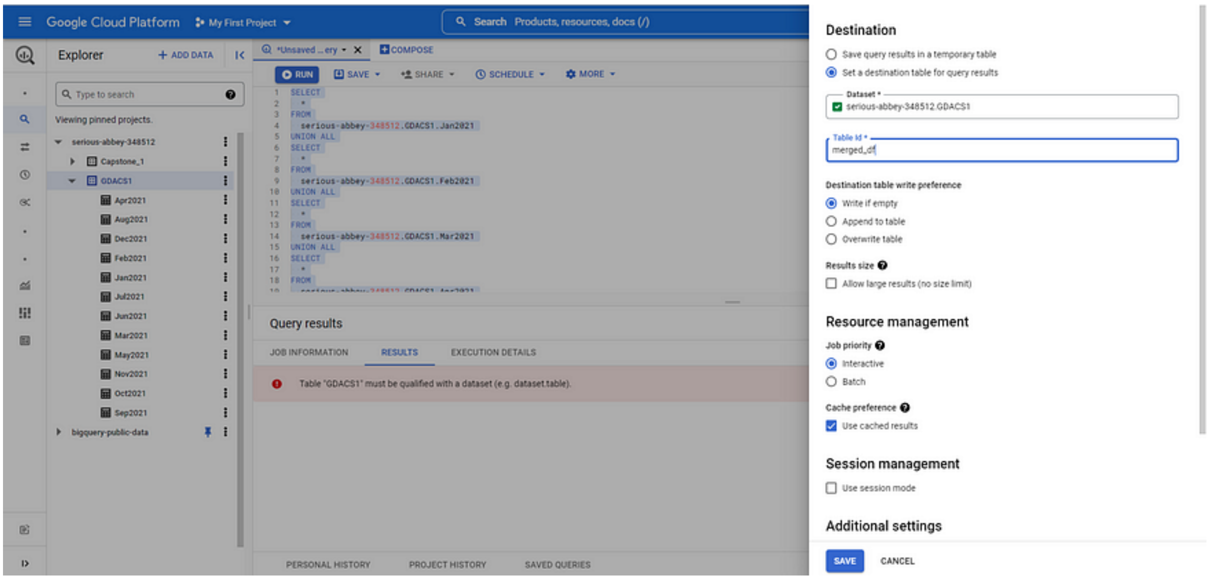
```
SELECT column_name(s) FROM table1

UNION ALL

SELECT column_name(s) FROM table2;
```

This is how my query looked like.

Now let's save our query as a new table called ‘merged_df’, which we do by pressing more > query settings: set a destination table and name the table.



```
SELECT
  *
FROM
  serious-abbey-348512.GDACS1.Jan2021
UNION ALL
SELECT
  *
FROM
  serious-abbey-348512.GDACS1.Feb2021
UNION ALL
SELECT
  *
FROM
  serious-abbey-348512.GDACS1.Mar2021
UNION ALL
SELECT
  *
FROM
```

Table "GDACS1" must be qualified with a dataset (e.g. dataset.table).

Destination

○ Save query results in a temporary table
● Set a destination table for query results

Dataset * serious-abbey-348512.GDACS1

Table Id * merged_df

Destination table write preference

● Write if empty
○ Append to table
○ Overwrite table

Results size

☐ Allow large results (no size limit)

Resource management

Job priority

● Interactive
○ Batch

Cache preference

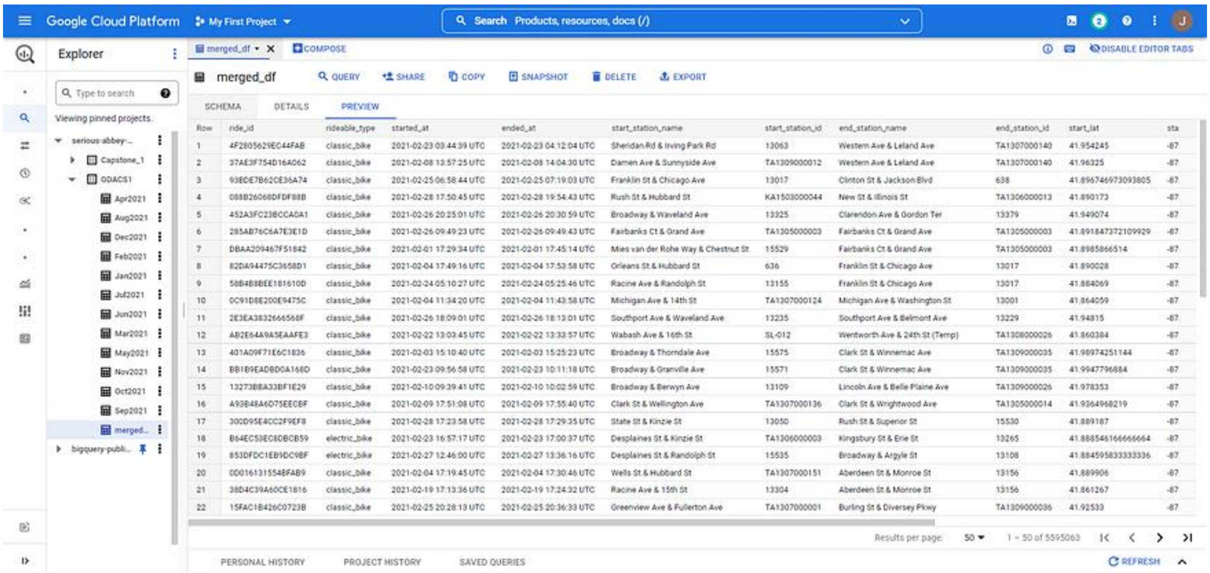
☑ Use cached results

Session management

☐ Use session mode

Additional settings

SAVE CANCEL



merged_df

Row	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	sta
1	4F2805629EC44FAB	classic_bike	2021-02-23 03:44:39 UTC	2021-02-23 04:12:04 UTC	Sheridan Rd & Irving Park Rd	13063	Western Ave & Leland Ave	TA1307000140	41.954245	-87
2	37AE3F754D16A062	classic_bike	2021-02-08 13:57:25 UTC	2021-02-08 14:04:30 UTC	Damen Ave & Sunnyside Ave	TA1309000012	Western Ave & Leland Ave	TA1307000140	41.96325	-87
3	93BDE7B62CE36A74	classic_bike	2021-02-25 06:58:44 UTC	2021-02-25 07:19:03 UTC	Franklin St & Chicago Ave	13017	Clinton St & Jackson Blvd	638	41.896746973093805	-87
4	088B26068DFDF88B	classic_bike	2021-02-28 17:50:45 UTC	2021-02-28 19:54:43 UTC	Rush St & Hubbard St	KA1503000044	New St & Illinois St	TA1306000013	41.890173	-87
5	452A3FC23BCCA0A1	classic_bike	2021-02-26 20:25:01 UTC	2021-02-26 20:30:59 UTC	Broadway & Waveland Ave	13325	Clarendon Ave & Gordon Ter	13379	41.949074	-87
6	285AB76C6A7E3E1D	classic_bike	2021-02-26 09:49:23 UTC	2021-02-26 09:49:43 UTC	Fairbanks Ct & Grand Ave	TA1305000003	Fairbanks Ct & Grand Ave	TA1305000003	41.891847372109929	-87
7	DBAA209467F51842	classic_bike	2021-02-01 17:29:34 UTC	2021-02-01 17:45:14 UTC	Mies van der Rohe Way & Chestnut St	15529	Fairbanks Ct & Grand Ave	TA1305000003	41.8985866514	-87
8	82DA94475C3658D1	classic_bike	2021-02-04 17:49:16 UTC	2021-02-04 17:53:58 UTC	Orleans St & Hubbard St	636	Franklin St & Chicago Ave	13017	41.890028	-87
9	58B4B8BEE181610D	classic_bike	2021-02-24 05:10:27 UTC	2021-02-24 05:25:46 UTC	Racine Ave & Randolph St	13155	Franklin St & Chicago Ave	13017	41.884069	-87
10	0C91D8E200E9475C	classic_bike	2021-02-04 11:34:20 UTC	2021-02-04 11:43:58 UTC	Michigan Ave & 14th St	TA1307000124	Michigan Ave & Washington St	13001	41.864059	-87
11	2E3EA3832666568F	classic_bike	2021-02-26 18:09:01 UTC	2021-02-26 18:13:01 UTC	Southport Ave & Waveland Ave	13235	Southport Ave & Belmont Ave	13229	41.94815	-87
12	AB2E64A9A5EAAFE3	classic_bike	2021-02-22 13:03:45 UTC	2021-02-22 13:33:57 UTC	Wabash Ave & 16th St	SL-012	Wentworth Ave & 24th St (Temp)	TA1308000026	41.860384	-87
13	401A09F71E6C1836	classic_bike	2021-02-03 15:10:40 UTC	2021-02-03 15:25:23 UTC	Broadway & Thorndale Ave	15575	Clark St & Winnemac Ave	TA1309000035	41.98974251144	-87
14	BB1B9EADBD0A168D	classic_bike	2021-02-23 09:56:58 UTC	2021-02-23 10:11:18 UTC	Broadway & Granville Ave	15571	Clark St & Winnemac Ave	TA1309000035	41.9947796884	-87
15	13273BBA33BF1E29	classic_bike	2021-02-10 09:39:41 UTC	2021-02-10 10:02:59 UTC	Broadway & Berwyn Ave	13109	Lincoln Ave & Belle Plaine Ave	TA1309000026	41.978353	-87
16	A93B48A6D75EECBF	classic_bike	2021-02-09 17:51:08 UTC	2021-02-09 17:55:40 UTC	Clark St & Wellington Ave	TA1307000136	Clark St & Wrightwood Ave	TA1305000014	41.9364968219	-87
17	300D95E4CC2F9EF8	classic_bike	2021-02-28 17:23:58 UTC	2021-02-28 17:29:35 UTC	State St & Kinzie St	13050	Rush St & Superior St	15530	41.889187	-87
18	B64EC53EC8DBCB59	electric_bike	2021-02-23 16:57:17 UTC	2021-02-23 17:00:37 UTC	Desplaines St & Kinzie St	TA1306000003	Kingsbury St & Erie St	13265	41.88854616666664	-87
19	853DFDC1EB9DC9BF	electric_bike	2021-02-27 12:46:00 UTC	2021-02-27 13:36:16 UTC	Desplaines St & Randolph St	15535	Broadway & Argyle St	13108	41.884595833333336	-87
20	0D0161315548FAB9	classic_bike	2021-02-04 17:19:45 UTC	2021-02-04 17:30:46 UTC	Wells St & Hubbard St	TA1307000151	Aberdeen St & Monroe St	13156	41.889906	-87
21	38D4C39A60CE1816	classic_bike	2021-02-19 17:13:36 UTC	2021-02-19 17:24:32 UTC	Racine Ave & 15th St	13304	Aberdeen St & Monroe St	13156	41.861267	-87
22	15FAC1B426C0723B	classic_bike	2021-02-25 20:28:13 UTC	2021-02-25 20:36:33 UTC	Greenview Ave & Fullerton Ave	TA1307000001	Burling St & Diversey Pkwy	TA1309000036	41.92533	-87

Results per page: 50 1 – 50 of 5595063

The new dataset, ‘merged_df’

Based on the current information available, it is just simply not enough to perform more intricate analysis, therefore we need to create more columns with the following:

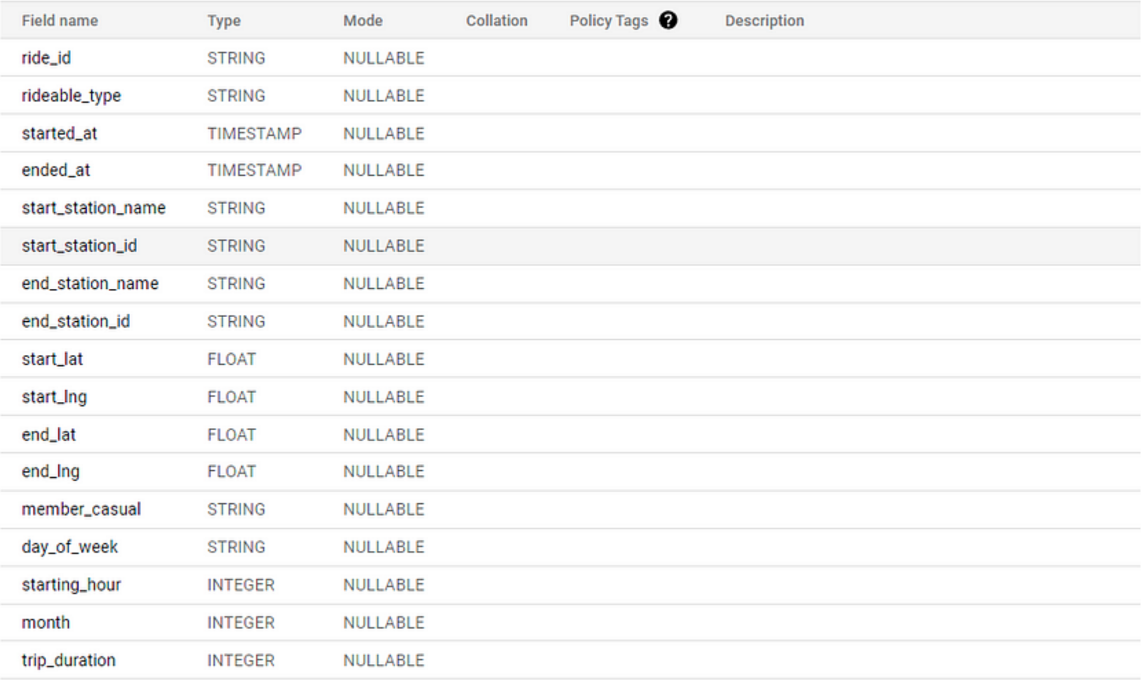
- Day of week — By using EXTRACT() & CASE (explained shortly)

for more information, click on EXTRACT & DAYOFWEEK

- Starting hour & Month — By using EXTRACT()
- Trip duration — By using TIMESTAMP_DIFF

for more information, click on TIMESTAMP_FUNCTIONS

I will be saving our new query by overwriting the old one. We also need to check if our new columns have incorrect formatting as well.



Field name	Type	Mode	Collation	Policy Tags ?	Description
ride_id	STRING	NULLABLE			
rideable_type	STRING	NULLABLE			
started_at	TIMESTAMP	NULLABLE			
ended_at	TIMESTAMP	NULLABLE			
start_station_name	STRING	NULLABLE			
start_station_id	STRING	NULLABLE			
end_station_name	STRING	NULLABLE			
end_station_id	STRING	NULLABLE			
start_lat	FLOAT	NULLABLE			
start_lng	FLOAT	NULLABLE			
end_lat	FLOAT	NULLABLE			
end_lng	FLOAT	NULLABLE			
member_casual	STRING	NULLABLE			
day_of_week	STRING	NULLABLE			
starting_hour	INTEGER	NULLABLE			
month	INTEGER	NULLABLE			
trip_duration	INTEGER	NULLABLE			

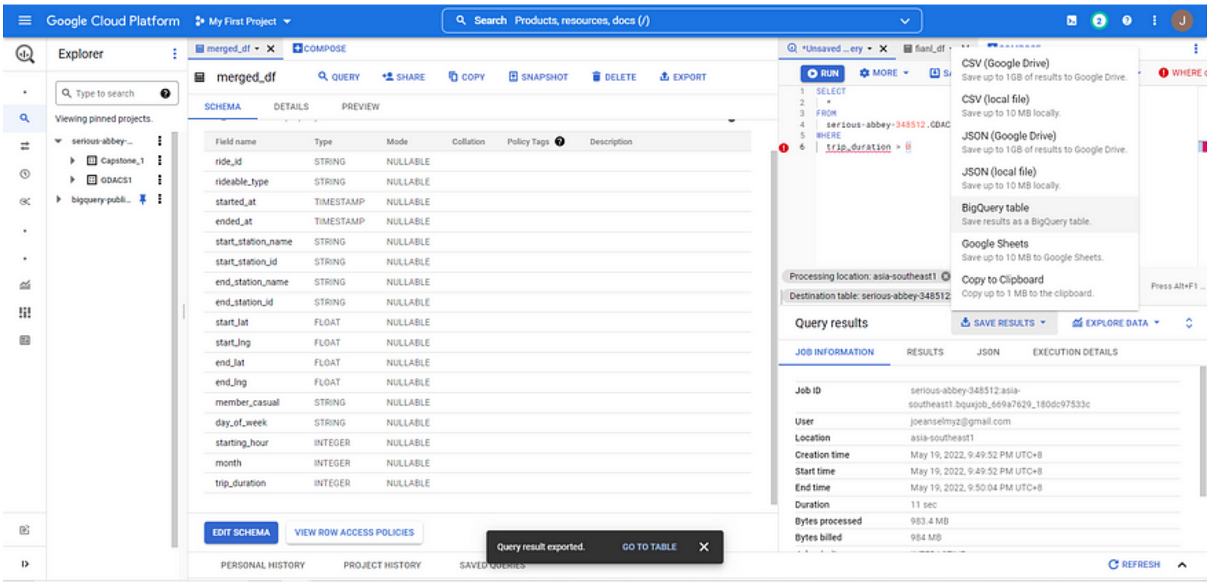
All seems good. We will be filtering all trip_durations which are 0 seconds and less by using the following query:

```
SELECT  * FROM  [table_name] WHERE  trip_duration > 0
```

Share

Now that we’re done, it's time to export the file for visualization using Tableau.

First, click on your latest query, and save the results to a BigQuery table. There may be compatibility issues, where you might need to create a new dataset (i named my exports) to save the new dataframe into.



merged_df

Field name	Type	Mode	Collation	Policy Tags	Description
ride_id	STRING	NULLABLE			
rideable_type	STRING	NULLABLE			
started_at	TIMESTAMP	NULLABLE			
ended_at	TIMESTAMP	NULLABLE			
start_station_name	STRING	NULLABLE			
start_station_id	STRING	NULLABLE			
end_station_name	STRING	NULLABLE			
end_station_id	STRING	NULLABLE			
start_lat	FLOAT	NULLABLE			
start_lng	FLOAT	NULLABLE			
end_lat	FLOAT	NULLABLE			
end_lng	FLOAT	NULLABLE			
member_casual	STRING	NULLABLE			
day_of_week	STRING	NULLABLE			
starting_hour	INTEGER	NULLABLE			
month	INTEGER	NULLABLE			
trip_duration	INTEGER	NULLABLE			

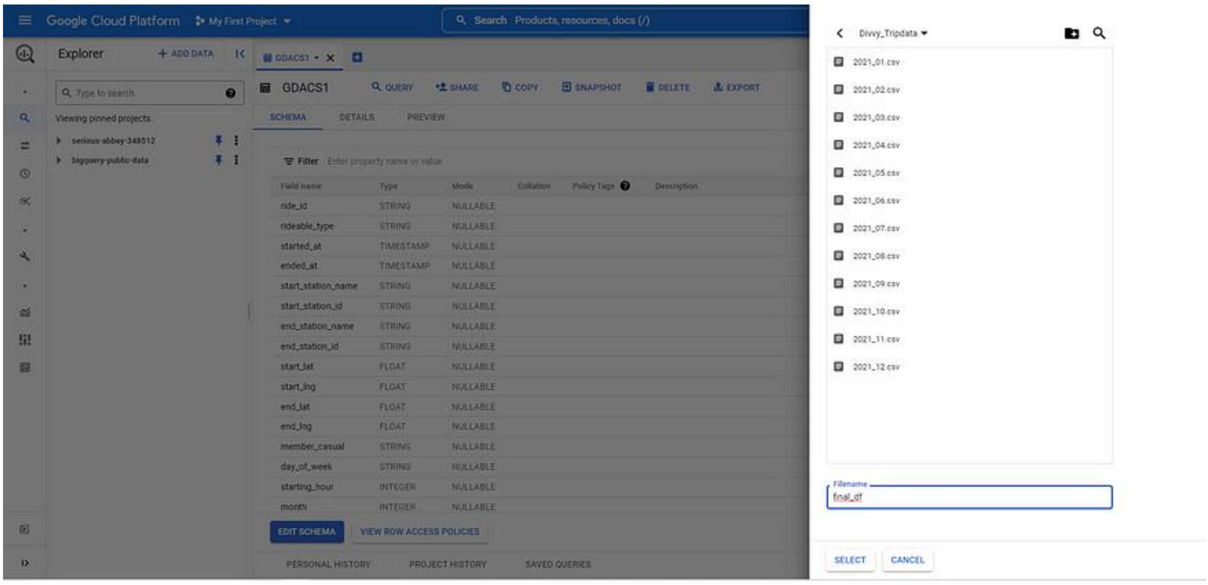
```
SELECT
  *
FROM
  serious-abbey-348512.GDAC
WHERE
  trip_duration > 0
```

Processing location: asia-southeast1

Destination table: serious-abbey-348512

Job ID	serious-abbey-348512:asia-southeast1.bquxjob_669a7629_180dc97533c
User	joeanselmyz@gmail.com
Location	asia-southeast1
Creation time	May 19, 2022, 9:49:52 PM UTC+8
Start time	May 19, 2022, 9:49:52 PM UTC+8
End time	May 19, 2022, 9:50:04 PM UTC+8
Duration	11 sec
Bytes processed	983.4 MB
Bytes billed	984 MB

Next, open your newly created table(the latest one), and press export to GCS as a CSV with or without GZIP compression (compress to save bandwidth but exports will take a lot longer about 20–30 minutes).



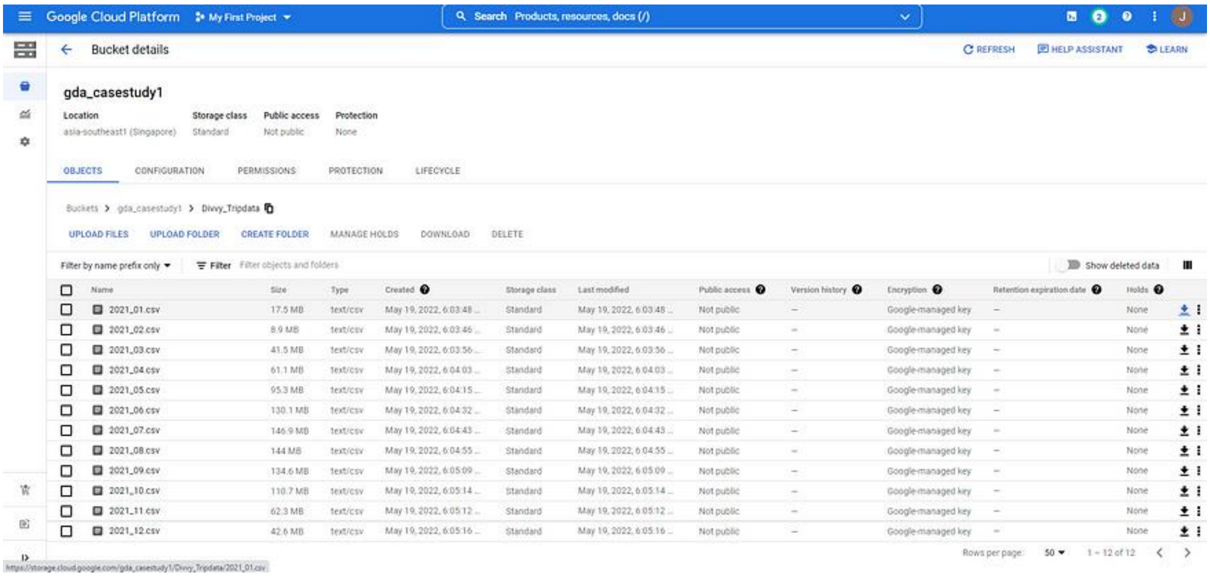
Field name	Type	Mode	Collation	Policy Tags	Description
ride_id	STRING	NULLABLE			
rideable_type	STRING	NULLABLE			
started_at	TIMESTAMP	NULLABLE			
ended_at	TIMESTAMP	NULLABLE			
start_station_name	STRING	NULLABLE			
start_station_id	STRING	NULLABLE			
end_station_name	STRING	NULLABLE			
end_station_id	STRING	NULLABLE			
start_lat	FLOAT	NULLABLE			
start_lng	FLOAT	NULLABLE			
end_lat	FLOAT	NULLABLE			
end_lng	FLOAT	NULLABLE			
member_casual	STRING	NULLABLE			
day_of_week	STRING	NULLABLE			
starting_hour	INTEGER	NULLABLE			
month	INTEGER	NULLABLE			

2021_01.csv
2021_02.csv
2021_03.csv
2021_04.csv
2021_05.csv
2021_06.csv
2021_07.csv
2021_08.csv
2021_09.csv
2021_10.csv
2021_11.csv
2021_12.csv

Filename: final_df

Select your output destination, and again, give your file a name (ENDING IN .CSV). The exporting process can take up to 30minutes. have a quick bite or get a cup of coffee while you’re at it.

After it's done, press the navigation menu, open your Google Cloud Storage, navigate to your file destination, and you can download it to share, or to use for visualization.



gda_casestudy1

Location	Storage class	Public access	Protection
asia-southeast1 (Singapore)	Standard	Not public	None

Buckets > gda_casestudy1 > Divvy_Tripdata

□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration date	Holds
□	2021_01.csv	17.5 MB	text/csv	May 19, 2022, 6:03:48 ...	Standard	May 19, 2022, 6:03:48 ...	Not public	–	Google-managed key	–	None
□	2021_02.csv	8.9 MB	text/csv	May 19, 2022, 6:03:46 ...	Standard	May 19, 2022, 6:03:46 ...	Not public	–	Google-managed key	–	None
□	2021_03.csv	41.5 MB	text/csv	May 19, 2022, 6:03:56 ...	Standard	May 19, 2022, 6:03:56 ...	Not public	–	Google-managed key	–	None
□	2021_04.csv	61.1 MB	text/csv	May 19, 2022, 6:04:03 ...	Standard	May 19, 2022, 6:04:03 ...	Not public	–	Google-managed key	–	None
□	2021_05.csv	95.3 MB	text/csv	May 19, 2022, 6:04:15 ...	Standard	May 19, 2022, 6:04:15 ...	Not public	–	Google-managed key	–	None
□	2021_06.csv	130.1 MB	text/csv	May 19, 2022, 6:04:32 ...	Standard	May 19, 2022, 6:04:32 ...	Not public	–	Google-managed key	–	None
□	2021_07.csv	146.9 MB	text/csv	May 19, 2022, 6:04:43 ...	Standard	May 19, 2022, 6:04:43 ...	Not public	–	Google-managed key	–	None
□	2021_08.csv	144 MB	text/csv	May 19, 2022, 6:04:55 ...	Standard	May 19, 2022, 6:04:55 ...	Not public	–	Google-managed key	–	None
□	2021_09.csv	134.6 MB	text/csv	May 19, 2022, 6:05:09 ...	Standard	May 19, 2022, 6:05:09 ...	Not public	–	Google-managed key	–	None
□	2021_10.csv	110.7 MB	text/csv	May 19, 2022, 6:05:14 ...	Standard	May 19, 2022, 6:05:14 ...	Not public	–	Google-managed key	–	None
□	2021_11.csv	62.3 MB	text/csv	May 19, 2022, 6:05:12 ...	Standard	May 19, 2022, 6:05:12 ...	Not public	–	Google-managed key	–	None
□	2021_12.csv	42.6 MB	text/csv	May 19, 2022, 6:05:16 ...	Standard	May 19, 2022, 6:05:16 ...	Not public	–	Google-managed key	–	None

Rows per page: 50 1 – 12 of 12

https://storage.cloud.google.com/gda_casestudy1/Divvy_Tripdata/2021_01.csv

To follow up with the visualization, do head over to my main article where i use Tableau to create striking visuals from this dataset over here!