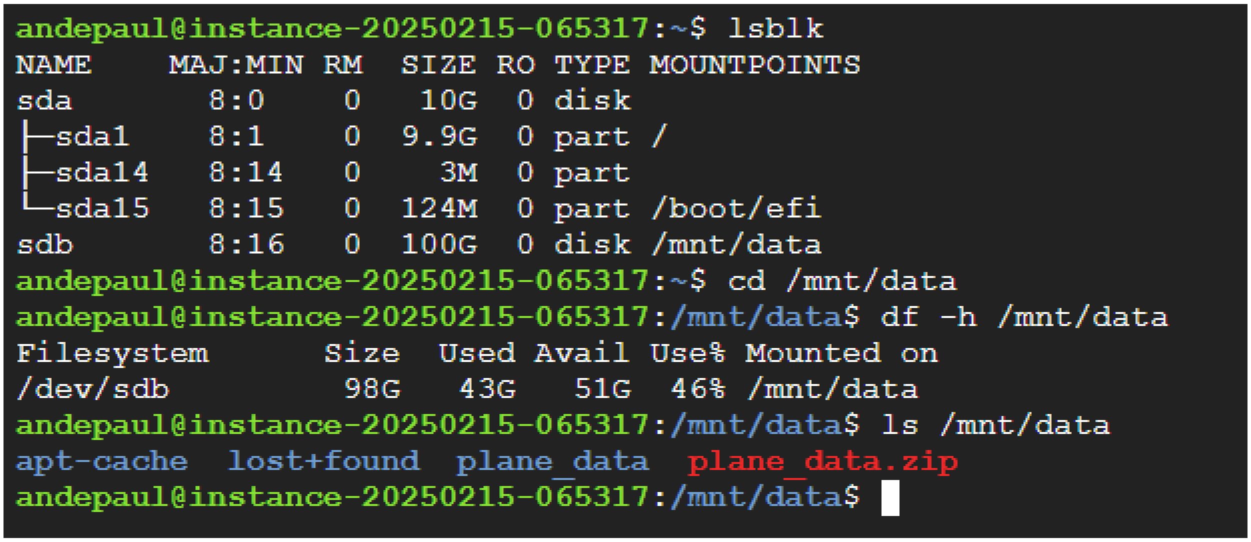
Oregon State University
CS512, Winter 2025
Group Homework: BigQuery - Bigish Data
Team Nexus
Paul J Anderson
Rachel Hughes

Date 2.15.2025

This is a Big Data assignment that served to give us experience with setting up and running Linux-based VMs from a Unix SSH terminal, loading large datasets, scrubbing our data using cloud tools, then doing initial investigation into trends and patterns and answering simple questions. For this assignment, we used Google Cloud Services (Compute Engine, Cloud Dataprep, and BigQuery) and a 10GB datafile of worldwide ICAO transmissions for a given period of time.

Question: "How many unique airplanes identified by their ICAO were in the area defined by Lat 44.497222 +/- 0.2 and Lon -123.289444 +/- 0.2. This is a rectangle around the Corvallis, OR airport?"
ANSWER: 85

1. Get Zip file onto Compute Engine Instance

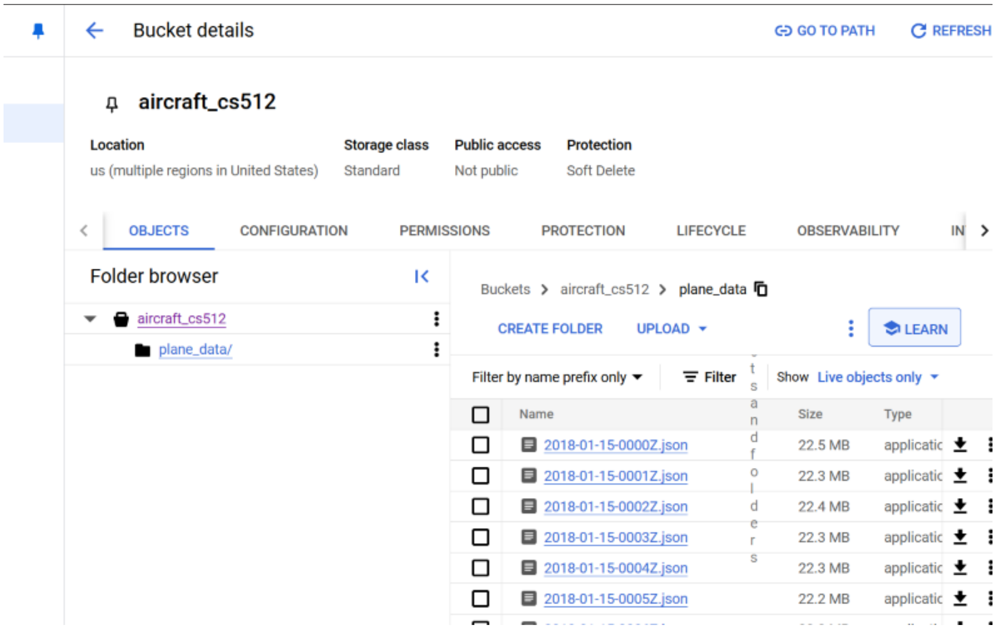


```
andepaul@instance-20250215-065317:~$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda       8:0    0   10G  0 disk
├─sda1    8:1    0  9.9G  0 part /
├─sda14   8:14   0    3M  0 part
└─sda15   8:15   0  124M  0 part /boot/efi
sdb       8:16   0  100G  0 disk /mnt/data
andepaul@instance-20250215-065317:~$ cd /mnt/data
andepaul@instance-20250215-065317:/mnt/data$ df -h /mnt/data
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb         98G   43G   51G  46% /mnt/data
andepaul@instance-20250215-065317:/mnt/data$ ls /mnt/data
apt-cache  lost+found  plane_data  plane_data.zip
andepaul@instance-20250215-065317:/mnt/data$
```

We created a VM using Google Compute Engine. Initially, we used a simple machine with only 4 GB of RAM and 10 GB of boot disk space. We purchased an external disk with 100 GB as an expansion. The initial download of the zip file took about 50 minutes and stopped near completion due to a full hard drive. We stopped the machine and remade it with an e2-standard-4 instance, 16 GB RAM, and a 100 GB SCSI disk, costing $98/month. Before downloading the zip file, we ensured we were using the 100 GB disk. We followed the documentation on file to use the necessary commands. This new machine downloaded the zip file in 17 minutes. The external disk space needed mapping, so we followed the directions provided in the documentation to mount the disk and alter our commands to include the mount drive location (in our case, /mnt/data/). We were able to successfully unzip the JSON files and isolate them by creating another folder and moving all non-JSON files there.

2. Load JSON files into Google Cloud Storage.

We then created a bucket on Google Cloud Storage named cs512_aircraft and used cloud-init in the SSH terminal to initiate the cloud. We moved some files around and reviewed what files we had in each folder directory to ensure our JSONs were ready for upload. We uploaded the plane_data JSONs to our bucket, gs://aircraft, using the gsutil cp command. We did run into a permission issue that prevented 75 objects from being transferred. Otherwise, the JSON data is confirmed in Google Cloud Storage.



aircraft_cs512

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	Soft Delete

Folder browser

- aircraft_cs512
  - plane_data/

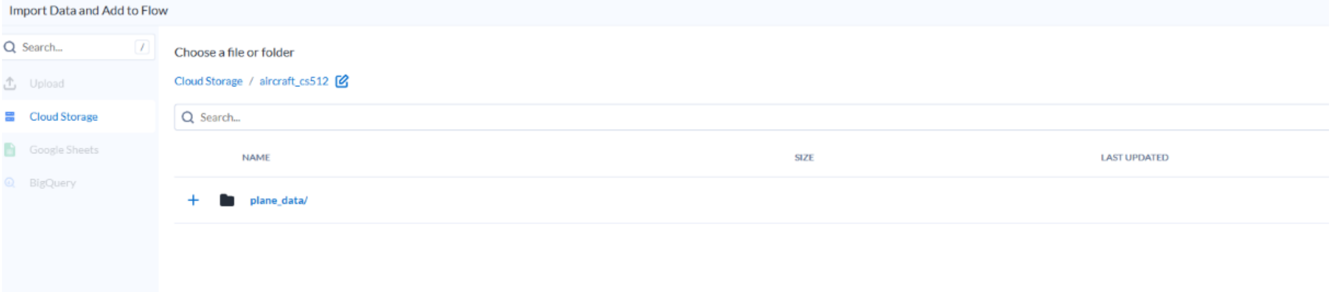
Buckets > aircraft_cs512 > plane_data

Show Live objects only

Name	Size	Type
2018-01-15-0000Z.json	22.5 MB	applicatic
2018-01-15-0001Z.json	22.3 MB	applicatic
2018-01-15-0002Z.json	22.4 MB	applicatic
2018-01-15-0003Z.json	22.3 MB	applicatic
2018-01-15-0004Z.json	22.3 MB	applicatic
2018-01-15-0005Z.json	22.2 MB	applicatic

3. Load JSON files as a data set into Google Cloud Dataprep.

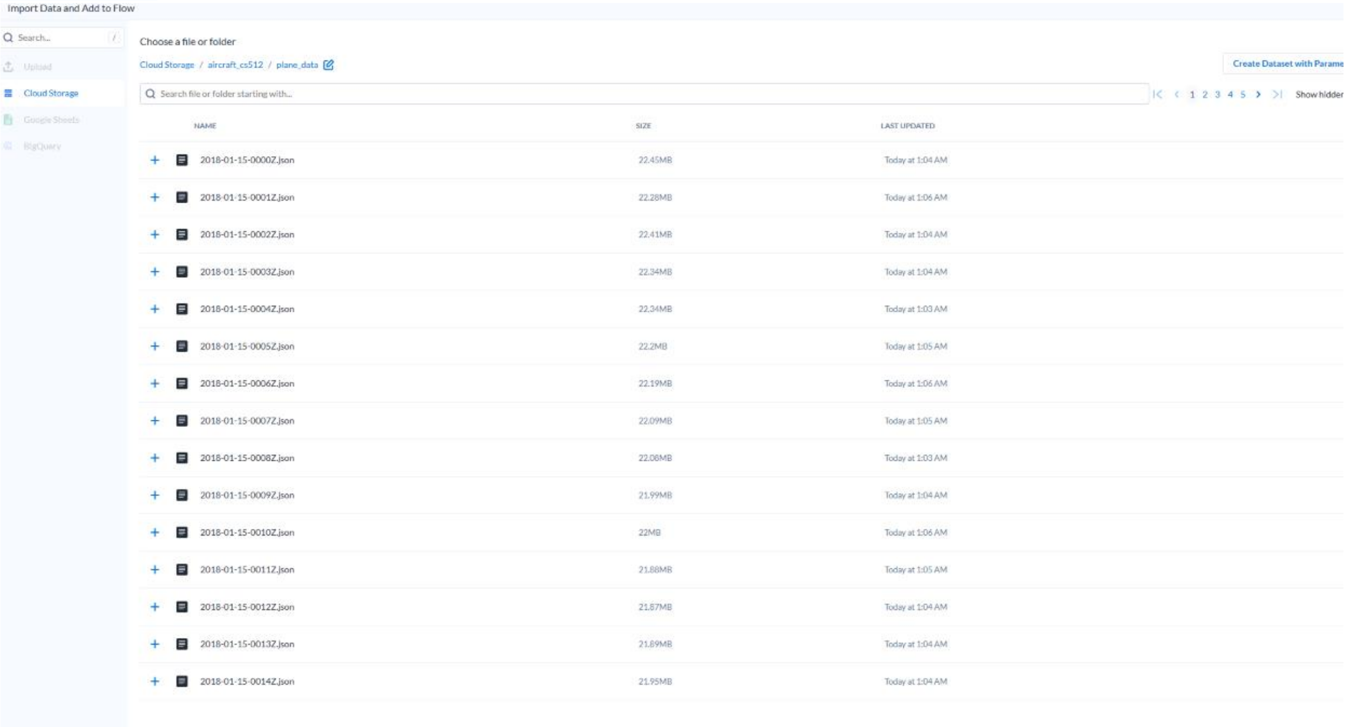
Next, we attempted to load the JSON files as a dataset into Dataprep. We tried this in several ways, as illustrated in our screenshots below. Although plane_data is clearly a file full of JSONs, previews of these data were not available. We created datasets in SSH gs:// and verified they were present in BigQuery, but the imported JSONs were not. Even a single JSON showed no records. We attempted to remove formatting from the JSON files and upload them again, but this returned another error. We consulted Ed, spoke with other students, messaged the instructor, and attended office hours. We were encouraged to complete the assignment without these steps. Therefore, this portion of the assignment was not completed. Significant effort was made to try and process a JSON through these steps. Assistance from team members, the professor, and the TA was sought, but a solution was not forthcoming.



Choose a file or folder

Cloud Storage / aircraft_cs512

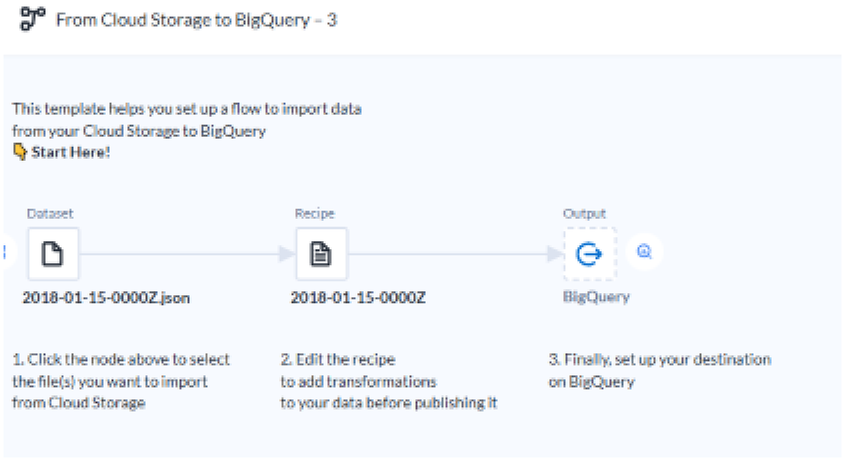
NAME	SIZE	LAST UPDATED
plane_data/		


Import Data and Add to Flow
Search...
Upload
Cloud Storage
Google Sheets
BigQuery
Create Dataset with Parame
Search file or folder starting with...
1 2 3 4 5
Show hidder

Choose a file or folder

Cloud Storage / aircraft_cs512 / plane_data

NAME	SIZE	LAST UPDATED
2018-01-15-0000Z.json	22.45MB	Today at 1:04 AM
2018-01-15-0001Z.json	22.28MB	Today at 1:06 AM
2018-01-15-0002Z.json	22.41MB	Today at 1:04 AM
2018-01-15-0003Z.json	22.34MB	Today at 1:04 AM
2018-01-15-0004Z.json	22.34MB	Today at 1:03 AM
2018-01-15-0005Z.json	22.2MB	Today at 1:05 AM
2018-01-15-0006Z.json	22.19MB	Today at 1:06 AM
2018-01-15-0007Z.json	22.09MB	Today at 1:05 AM
2018-01-15-0008Z.json	22.08MB	Today at 1:03 AM
2018-01-15-0009Z.json	21.99MB	Today at 1:04 AM
2018-01-15-0010Z.json	22MB	Today at 1:06 AM
2018-01-15-0011Z.json	21.88MB	Today at 1:05 AM
2018-01-15-0012Z.json	21.87MB	Today at 1:04 AM
2018-01-15-0013Z.json	21.89MB	Today at 1:04 AM
2018-01-15-0014Z.json	21.95MB	Today at 1:04 AM



From Cloud Storage to BigQuery - 3

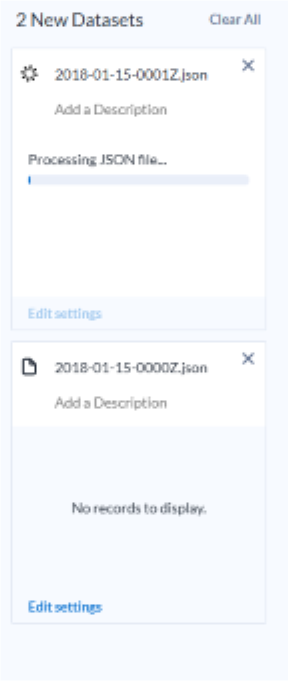
This template helps you set up a flow to import data from your Cloud Storage to BigQuery

👇 Start Here!

1. Click the node above to select the file(s) you want to import from Cloud Storage

2. Edit the recipe to add transformations to your data before publishing it

3. Finally, set up your destination on BigQuery


2 New Datasets

2018-01-15-0001Z.json

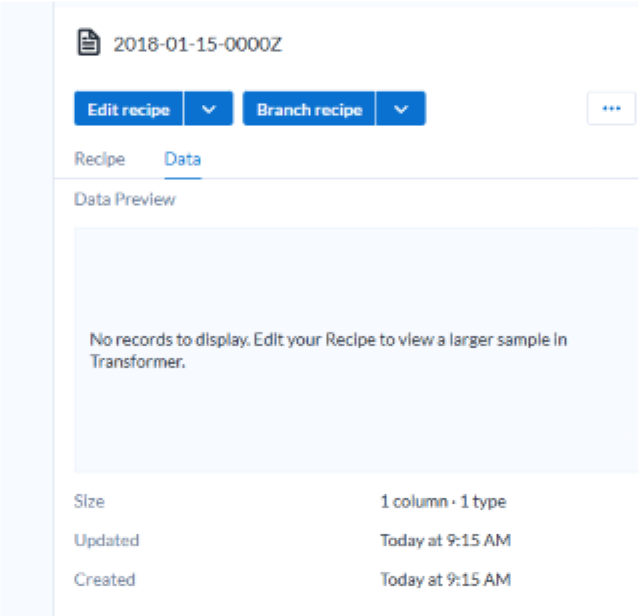
Add a Description

Processing JSON file...

2018-01-15-0000Z.json

Add a Description

No records to display.


2018-01-15-0000Z

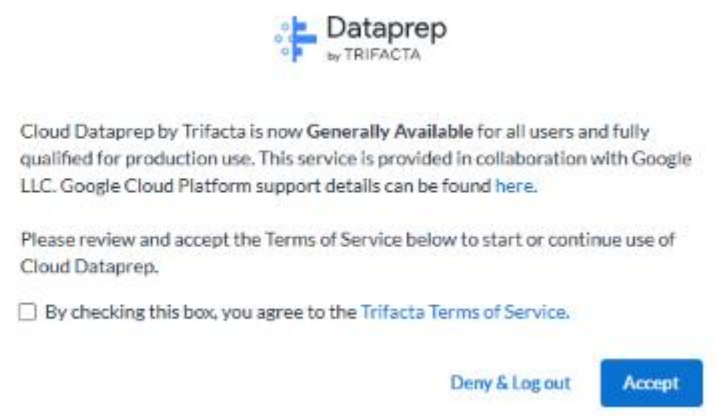
Data Preview

No records to display. Edit your Recipe to view a larger sample in Transformer.

Size 1 column · 1 type

Updated Today at 9:15 AM

Created Today at 9:15 AM

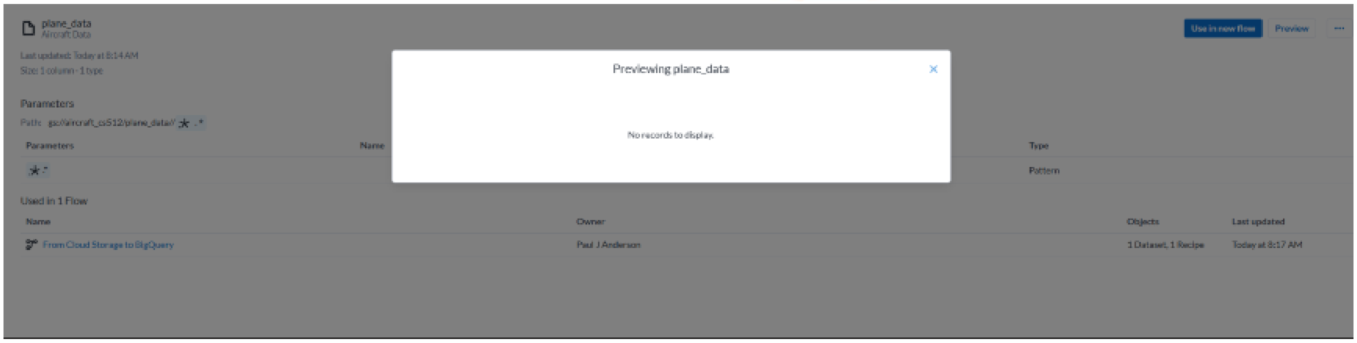


Dataprep by TRIFACTA

Cloud Dataprep by Trifacta is now **Generally Available** for all users and fully qualified for production use. This service is provided in collaboration with Google LLC. Google Cloud Platform support details can be found here.

Please review and accept the Terms of Service below to start or continue use of Cloud Dataprep.

□ By checking this box, you agree to the Trifacta Terms of Service.



plane_data
Aircraft Data

Last updated: Today at 8:14 AM
Size: 1 column · 1 type

Parameters

Path: gs://aircraft_cs512/plane_data/ ⁎ .*

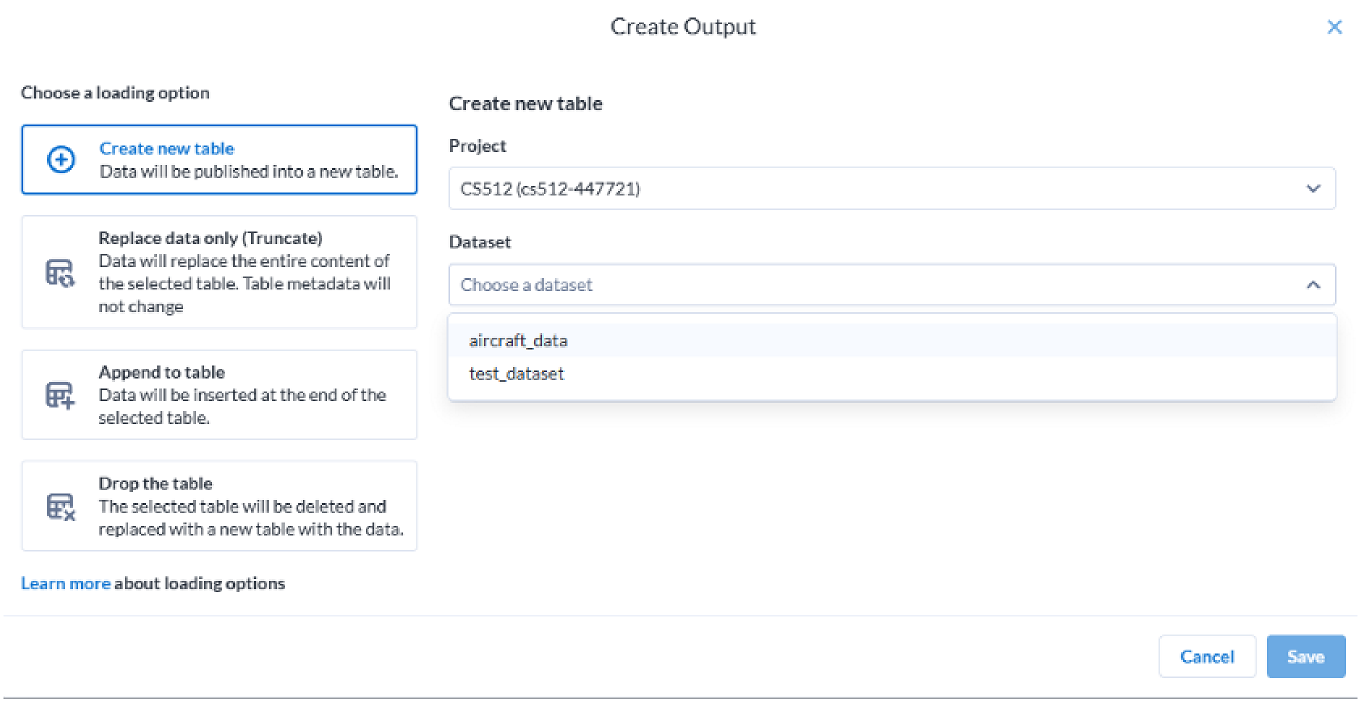
Parameters	Name	Type
⁎ .*		Pattern

Used in 1 Flow

Name	Owner	Objects	Last updated
From Cloud Storage to BigQuery	Paul J Anderson	1 Dataset, 1 Recipe	Today at 8:17 AM

Previewing plane_data

No records to display.



Create Output

Choose a loading option

- ⊕ **Create new table** Data will be published into a new table.
- **Replace data only (Truncate)** Data will replace the entire content of the selected table. Table metadata will not change
- **Append to table** Data will be inserted at the end of the selected table.
- **Drop the table** The selected table will be deleted and replaced with a new table with the data.

Learn more about loading options

Create new table

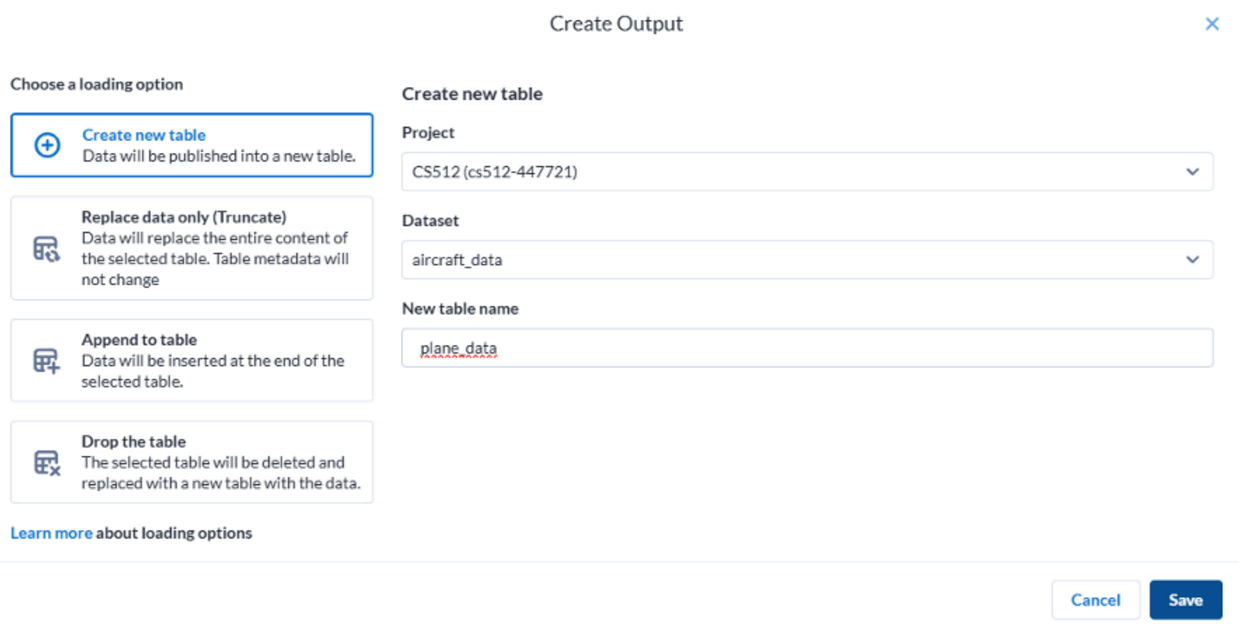
Project

CS512 (cs512-447721)

Dataset

Choose a dataset

- aircraft_data
- test_dataset



Choose a loading option

- **Create new table** Data will be published into a new table.
- **Replace data only (Truncate)** Data will replace the entire content of the selected table. Table metadata will not change
- **Append to table** Data will be inserted at the end of the selected table.
- **Drop the table** The selected table will be deleted and replaced with a new table with the data.

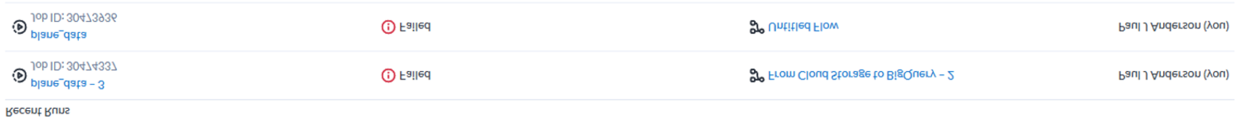
Learn more about loading options

Create new table

Project: CS512 (cs512-447721)

Dataset: aircraft_data

New table name: plane_data



Recent Runs

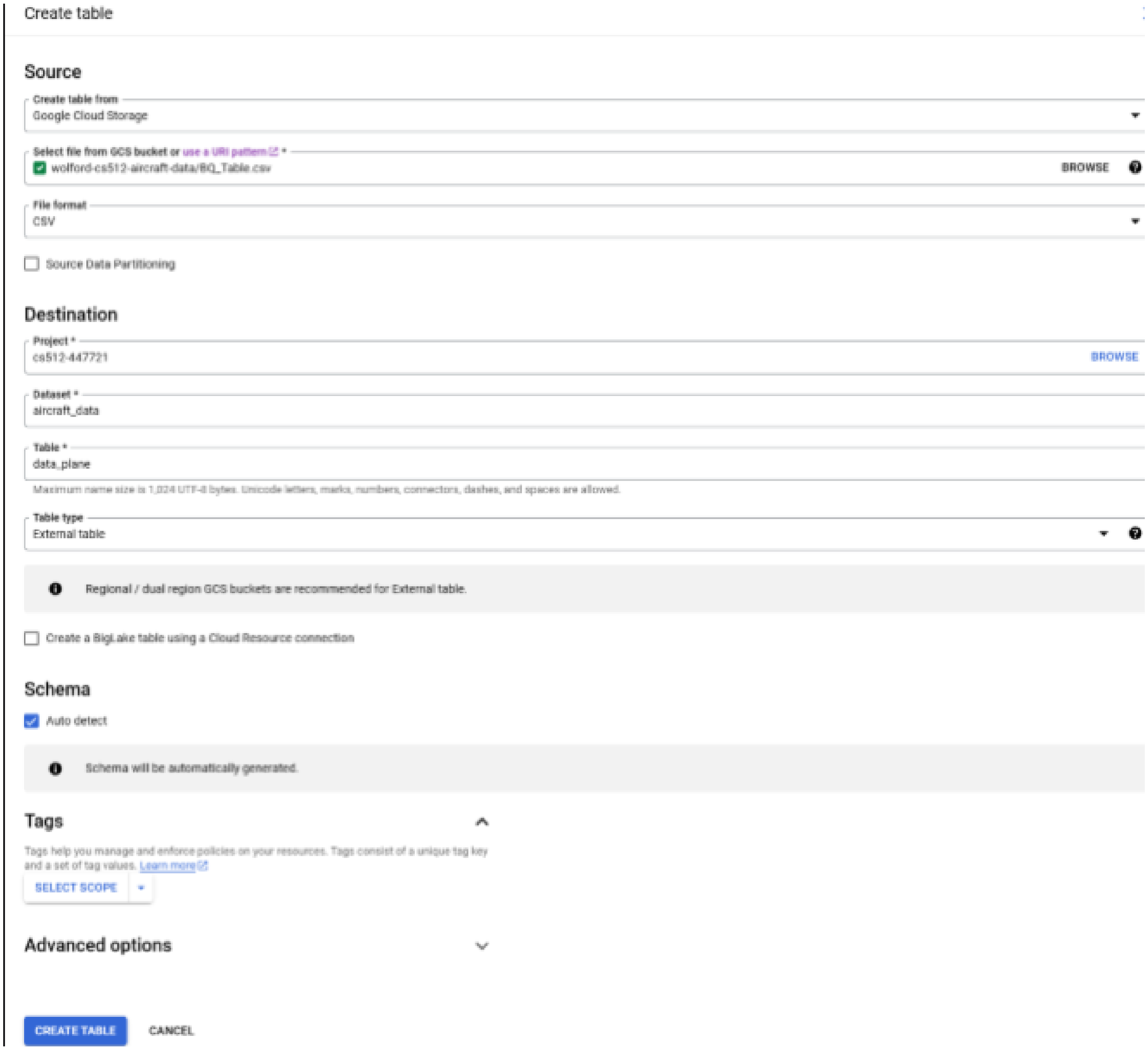
plane_data – 3 Job ID: 30474332	Failed	From Cloud Storage to BigQuery – 2	Paul J Anderson (you)
plane_data Job ID: 30472939	Failed	Untitled Flow	Paul J Anderson (you)

4. Parse JSON into appropriate columns in Dataprep

Like the previous step, this portion of the assignment was not completed. Significant effort was made to try and process a JSON through these steps. Assistance from team members, the professor, and the TA was sought, but a solution was not forthcoming.

5. Export Dataprep job into BigQuery

We were able to upload a pre-wrangled .CSV into BigQuery via the following steps. Create table from a GCS bucket.



Source

Create table from: Google Cloud Storage

Select file from GCS bucket or use a URI pattern *: ☑ wolford-cs512-aircraft-data/BQ_Table.csv

File format: CSV

□ Source Data Partitioning

Destination

Project *: cs512-447721

Dataset *: aircraft_data

Table *: data_plane

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.

Table type: External table

Regional / dual region GCS buckets are recommended for External table.

□ Create a BigLake table using a Cloud Resource connection

Schema

☑ Auto detect

Schema will be automatically generated.

Tags

Tags help you manage and enforce policies on your resources. Tags consist of a unique tag key and a set of tag values. Learn more

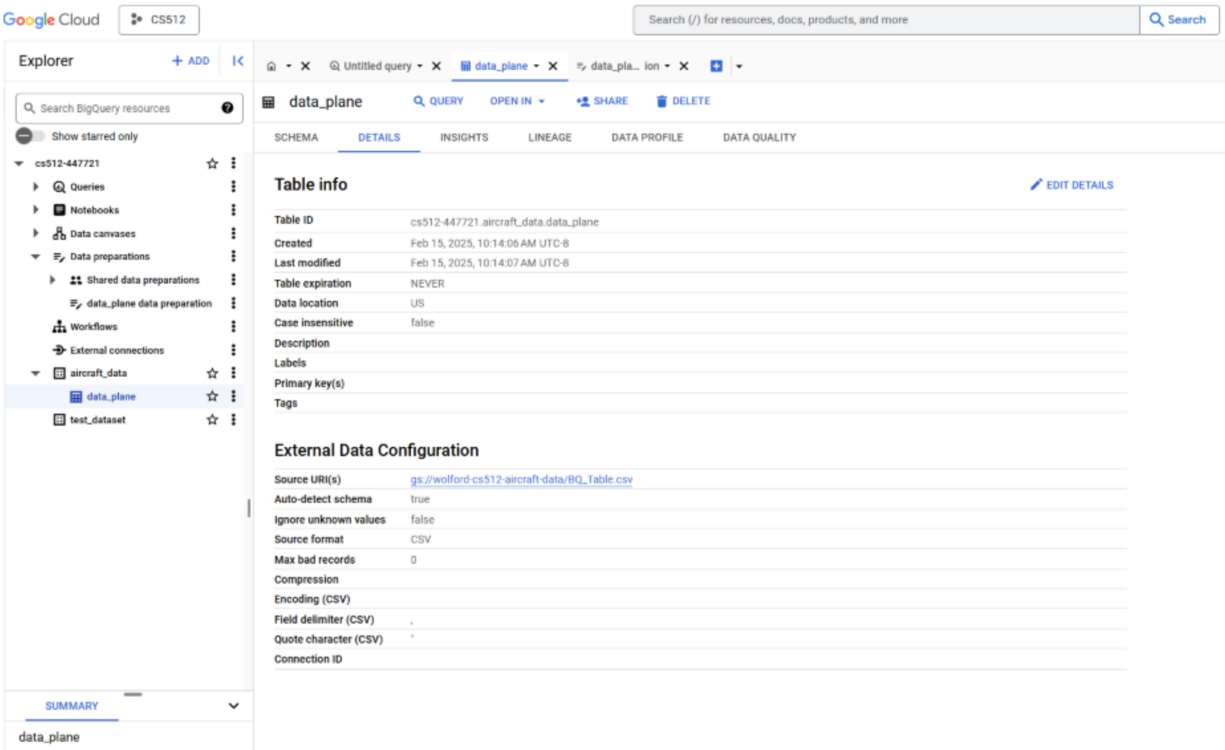


Table info

Table ID	cs512-447721.aircraft_data.data_plane
Created	Feb 15, 2025, 10:14:06 AM UTC-8
Last modified	Feb 15, 2025, 10:14:07 AM UTC-8
Table expiration	NEVER
Data location	US
Case insensitive	false
Description	
Labels	
Primary key(s)	
Tags	

External Data Configuration

Source URI(s)	gs://wolford-cs512-aircraft-data/BQ_Table.csv
Auto-detect schema	true
Ignore unknown values	false
Source format	CSV
Max bad records	0
Compression	
Encoding (CSV)	
Field delimiter (CSV)	,
Quote character (CSV)	"
Connection ID	

Open in data prep.

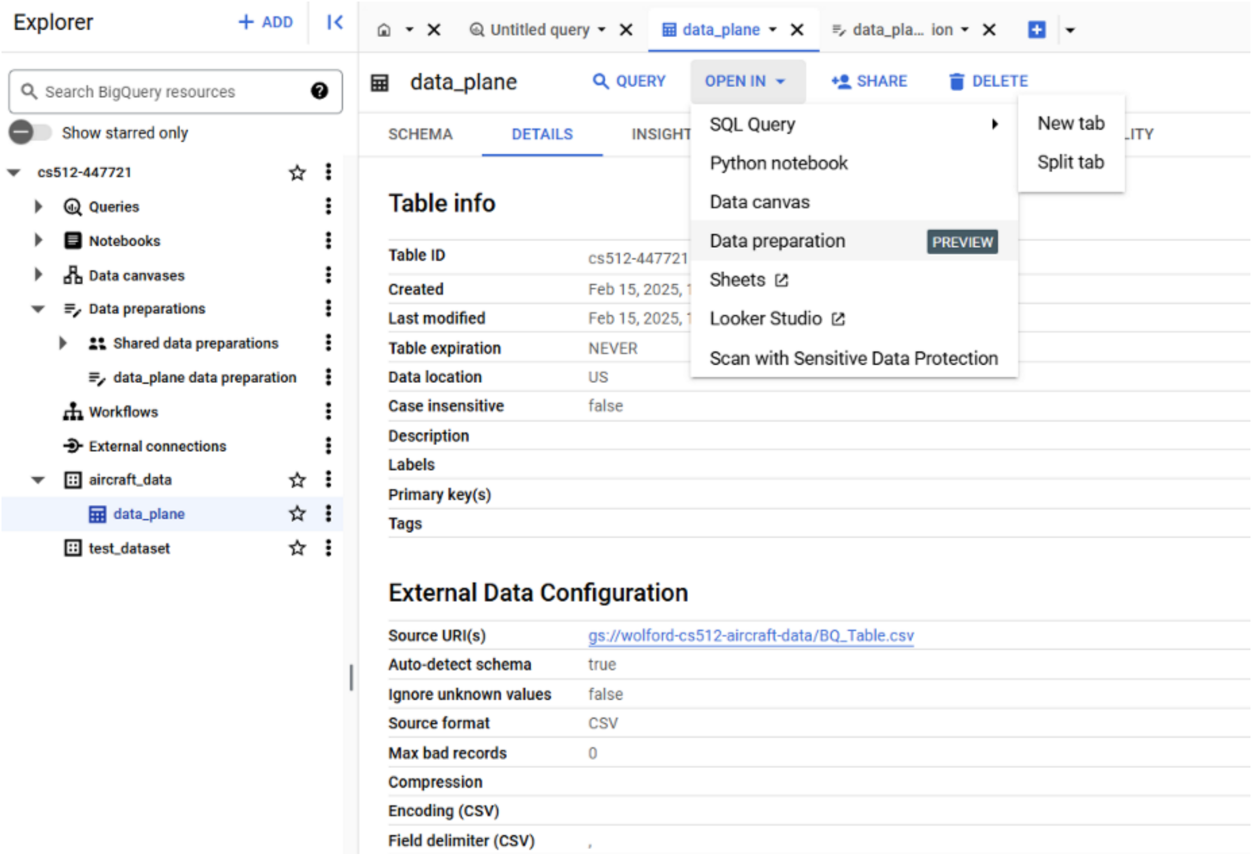


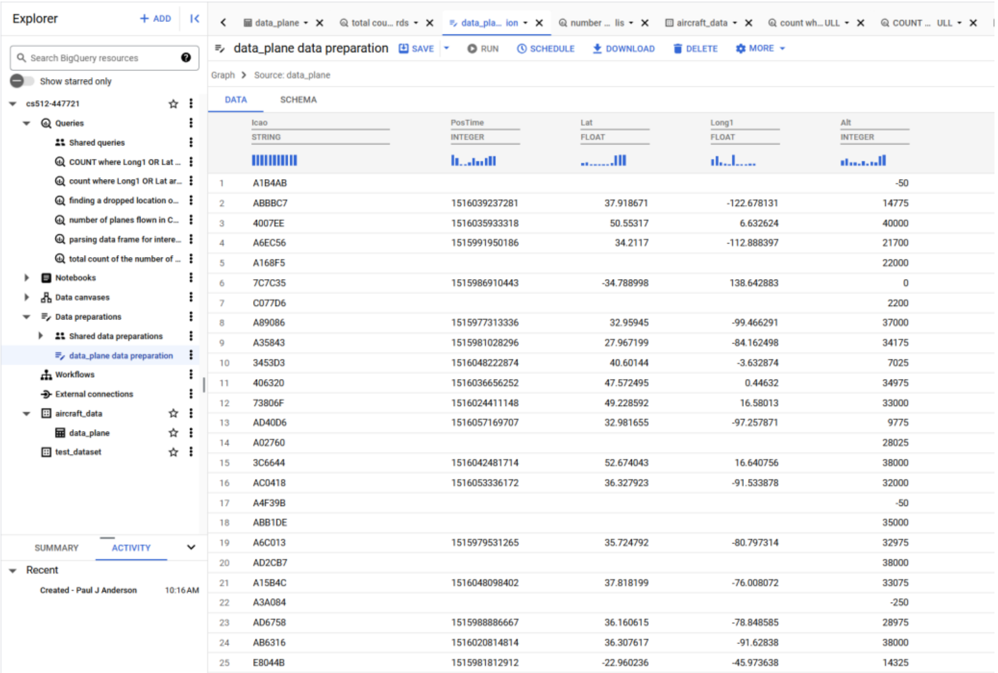
Table info

Table ID	cs512-447721
Created	Feb 15, 2025, 1
Last modified	Feb 15, 2025, 1
Table expiration	NEVER
Data location	US
Case insensitive	false
Description	
Labels	
Primary key(s)	
Tags	

External Data Configuration

Source URI(s)	gs://wolford-cs512-aircraft-data/BQ_Table.csv
Auto-detect schema	true
Ignore unknown values	false
Source format	CSV
Max bad records	0
Compression	
Encoding (CSV)	
Field delimiter (CSV)	,

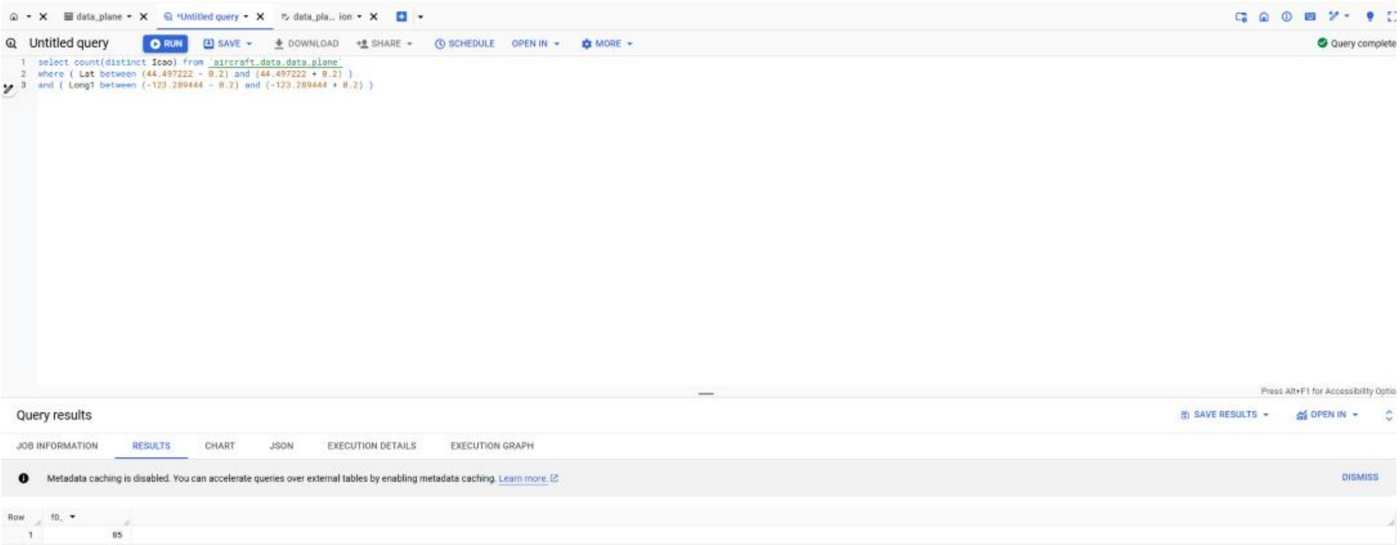
And visualize all 10,000 rows.



	Icao STRING	PosTime INTEGER	Lat FLOAT	Long1 FLOAT	Alt INTEGER
1	A1B4AB				-50
2	ABBBC7	1516039237281	37.918671	-122.678131	14775
3	4007EE	1516035933318	50.55317	6.632624	40000
4	A6EC56	1515991950186	34.2117	-112.888397	21700
5	A168F5				22000
6	7C7C35	1515986910443	-34.788998	138.642883	0
7	C077D6				2200
8	A89086	1515977313336	32.95945	-99.466291	37000
9	A35843	1515981028296	27.967199	-84.162498	34175
10	3453D3	1516048222874	40.60144	-3.632874	7025
11	406320	1516036656252	47.572495	0.44632	34975
12	73806F	1516024411148	49.228592	16.58013	33000
13	AD40D6	1516057169707	32.981655	-97.257871	9775
14	A02760				28025
15	3C6644	1516042481714	52.674043	16.640756	38000
16	AC0418	1516053336172	36.327923	-91.533878	32000
17	A4F39B				-50
18	ABB1DE				35000
19	A6C013	1515979531265	35.724792	-80.797314	32975
20	AD2CB7				38000
21	A15B4C	1516048098402	37.818199	-76.008072	33075
22	A3A084				-250
23	AD6758	1515988886667	36.160615	-78.848585	28975
24	AB6316	1516020814814	36.307617	-91.62838	38000
25	E8044B	1515981812912	-22.960236	-45.973638	14325

6. Write BigQuery SQL to compute the answer

We then opened a query in Google Big Query and used the given query to answer the question about the data and hit Run.



```
select count(distinct Icao) from `aircraft_data.data_plane`
where ( Lat between (44.497222 - 0.2) and (44.497222 + 0.2) )
and ( Long1 between (-123.289444 - 0.2) and (-123.289444 + 0.2) )
```

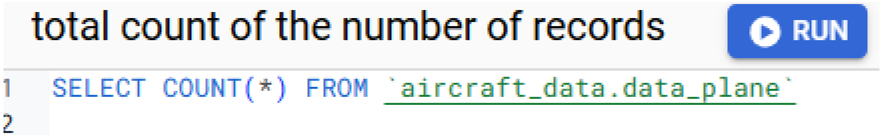
Metadata caching is disabled. You can accelerate queries over external tables by enabling metadata caching. Learn more

Row	f0_
1	85

The query returned 85 entries.

I then took some time to explore this large dataset, which seems to be worldwide flights for a set time period. I ran the following queries to explore the data:

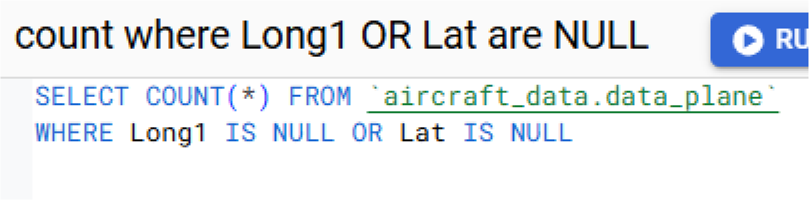
How many total rows



```
SELECT COUNT(*) FROM `aircraft_data.data_plane`
```

Returns: 14446218

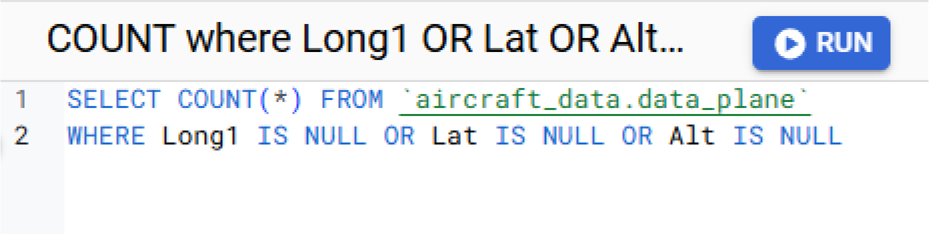
How many cells with empty spaces in Long1 or Lat



```
SELECT COUNT(*) FROM `aircraft_data.data_plane`
WHERE Long1 IS NULL OR Lat IS NULL
```

Returns: 3953109

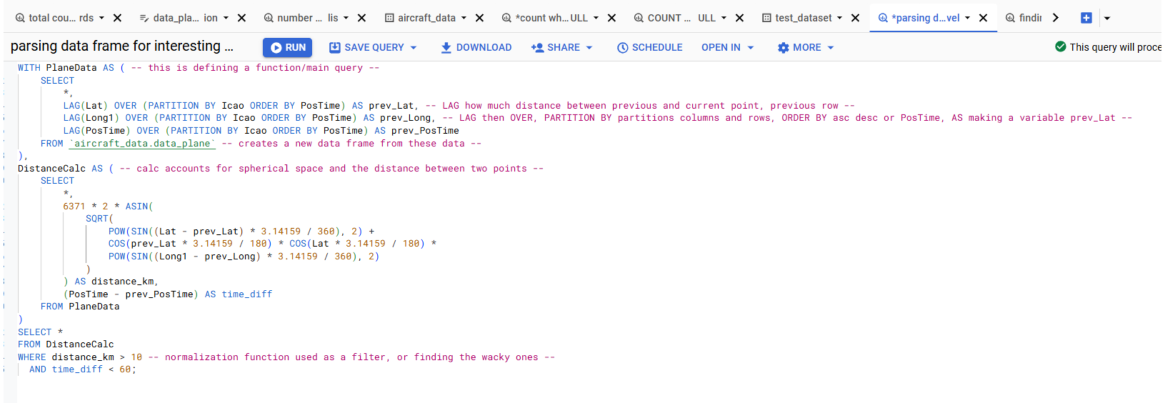
How many cells with empty spaces in Long1 or Lat or Alt



```
SELECT COUNT(*) FROM `aircraft_data.data_plane`
WHERE Long1 IS NULL OR Lat IS NULL OR Alt IS NULL
```

Returns: 3991106

The TA then had us run the following Query



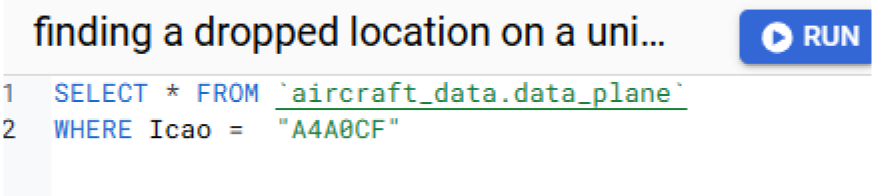
```
WITH PlaneData AS ( -- this is defining a function/main query --
    SELECT
        *,
        LAG(Lat) OVER (PARTITION BY Icao ORDER BY PosTime) AS prev_Lat, -- LAG how much distance between previous and current point, previous row --
        LAG(Long1) OVER (PARTITION BY Icao ORDER BY PosTime) AS prev_Long, -- LAG then OVER, PARTITION BY partitions columns and rows, ORDER BY asc desc or PosTime, AS making a variable prev_Lat --
        LAG(PosTime) OVER (PARTITION BY Icao ORDER BY PosTime) AS prev_PosTime
    FROM `aircraft_data.data_plane` -- creates a new data frame from these data --
),
DistanceCalc AS ( -- calc accounts for spherical space and the distance between two points --
    SELECT
        *,
        6371 * 2 * ASIN(
            SQRT(
                POW(SIN((Lat - prev_Lat) * 3.14159 / 360), 2) +
                COS(prev_Lat * 3.14159 / 180) * COS(Lat * 3.14159 / 180) *
                POW(SIN((Long1 - prev_Long) * 3.14159 / 360), 2)
            )
        ) AS distance_km,
        (PosTime - prev_PosTime) AS time_diff
    FROM PlaneData
)
SELECT *
FROM DistanceCalc
WHERE distance_km > 10 -- normalization function used as a filter, or finding the wacky ones --
  AND time_diff < 60;
```

This parsed the colums and rows to create a new data frame with the fields for previous position so that the distance between those two positions can be calculated using the geometry of a sphere. It then sets a filter level for distance (km) and time (seconds).

After running this, we found two interesting patterns.

1. There were many null fields for PosTime, Lat, and Long1
2. There Icao ID A4A0CF seemed to jump in position irregularly.

We ran the following script on the original dataframe to try and find out more about this plane.



```
SELECT * FROM `aircraft_data.data_plane`
WHERE Icao =  "A4A0CF"
```

After running this, we found one interesting pattern.

1. Out of the 873 records for this unique Icao, 587 had null records for PosTime, Lat, and Long1
2. 12 of these had null altitude

In considering why this might be the case, we hypothesized that the plane could have had its transmitter turned off or it could have simply malfunctioned midair. With more insight into the instrumentation of aircraft, and transmission of these four variables, with the planes Icao signature, there may be a simple conclusion or more interesting questions to be investigated.

Note: There is some concern that the JSON scrubbing in data prep was not completed. Based on the rubric, this is a significant issue. However, we feel we have documented and communicated about the issue well. Please let our team know if we can learn about this function of Google Cloud Services in another way and we would be happy to review it.

We were both able to complete the assignment in a similar fashion. Paul wrote the report.

Code References

1. Debian. (n.d.). Unix text tools. In *Debian Reference Manual*. Retrieved February 15, 2025, from https://www.debian.org/doc/manuals/debian-reference/ch01.en.html#_unix_text_tools
2. Debian. (n.d.). *FileSystem*. Retrieved February 15, 2025, from https://wiki.debian.org/FileSystem
3. Linux Kernel Labs. (n.d.). *Filesystem*. Retrieved February 15, 2025, from https://linux-kernel-labs.github.io/refs/heads/master/lectures/fs.html
4. Debian. (n.d.). *Debian Reference Manual*. Retrieved February 15, 2025, from https://www.debian.org/doc/manuals/debian-reference/ch01.en.html
5. Debian. (n.d.). *Shell Commands*. Retrieved February 15, 2025, from https://wiki.debian.org/ShellCommands
6. Debian. (n.d.). *Package basics*. In *Debian FAQ*. Retrieved February 15, 2025, from https://www.debian.org/doc/manuals/debian-faq/pkg-basics.en.html
7. Linux.com. (n.d.). *How to use the Linux command line: Basics of CLI*. Retrieved February 15, 2025, from https://www.linux.com/training-tutorials/how-use-linux-command-line-basics-cli/
8. Oregon State University. (n.d.). *Exploration: Data storage walkthrough 2* [Course module]. In CS512: Data Storage. Retrieved February 15, 2025, from https://canvas.oregonstate.edu/courses/1988535/pages/exploration-data-storage-walkthrough-2?module_item_id=25112246