School of Computing, Engineering and Built Environment

Big Data Landscape

# **Table of contents**

[**Task 1 3**](#_7su9jczfrgzv)

[1. A detailed overview, in your own words, of the dataset. 3](#_z4vvwb1rvv56)

[2. An appraisal of the quality of the dataset in terms of the integrity rules. 7](#_kn14xpnyntca)

[a. Data Completeness 7](#_mqu0n61okioa)

[b. Data Consistency 9](#_lslwhvghoz9r)

[c. Data Validation 9](#_89ac42x2fawe)

[d. Data Duplication 10](#_pbea2zon2au5)

[e. Data Timelines 11](#_fztysxkrr32q)

[**Task 2 13**](#_ms94ssvxer5)

[1. Use the Google BigQuery Console to create all the appropriate SQL code required to enable you to answer your question and to also deal with any quality issues identified in task 1. 13](#_z90638pia8bp)

[2. Create a Google Colab notebook that includes your SQL code as well as Python code that allows your SQL code to be executed using BigQuery. Your Python code must also store the information obtained that answers your question in Google Cloud Storage. 22](#_wz9bj550shkx)

[**Task 3 26**](#_6tdv09hsryqx)

[Visualize the information, now present in Google Cloud Storage, in a manner of your choosing, that you consider clearly communicates your findings and the value obtained. 26](#_gltprg37u3kn)

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

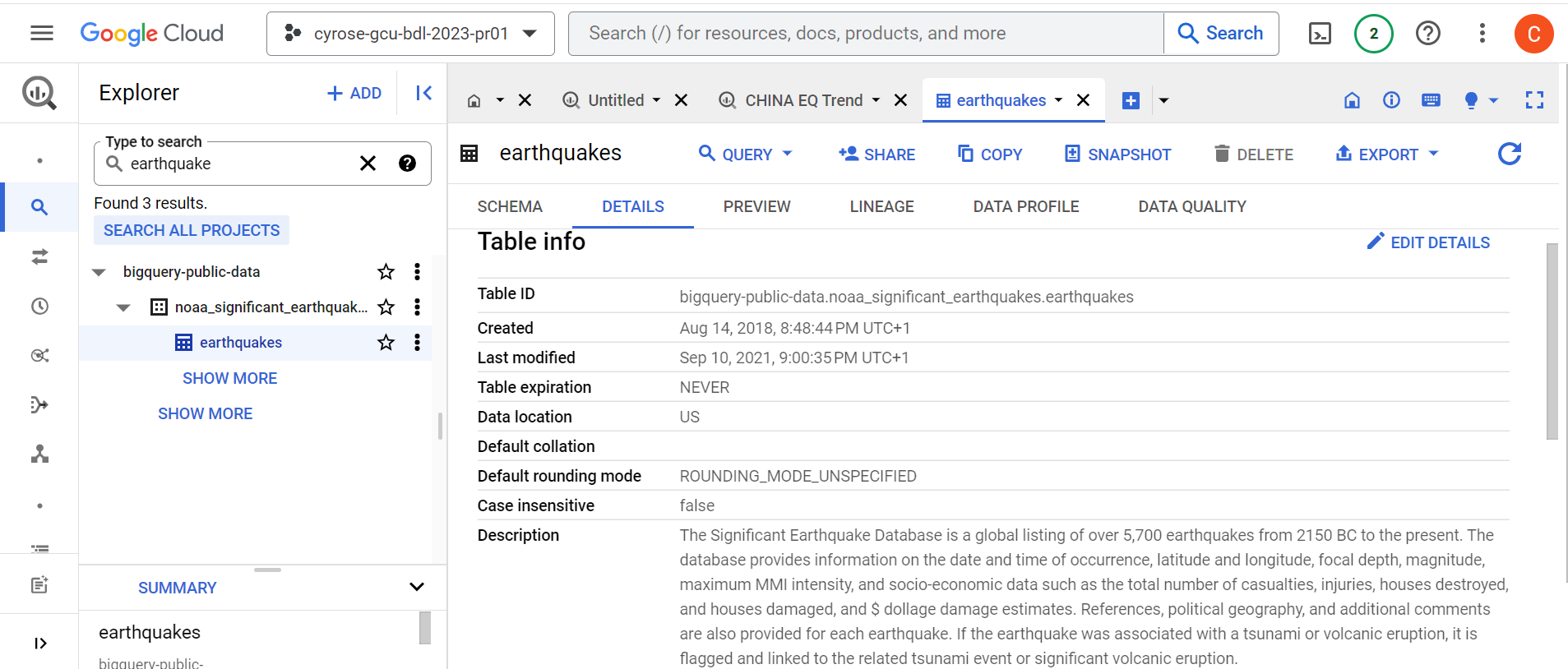
## 

## **Task 1**

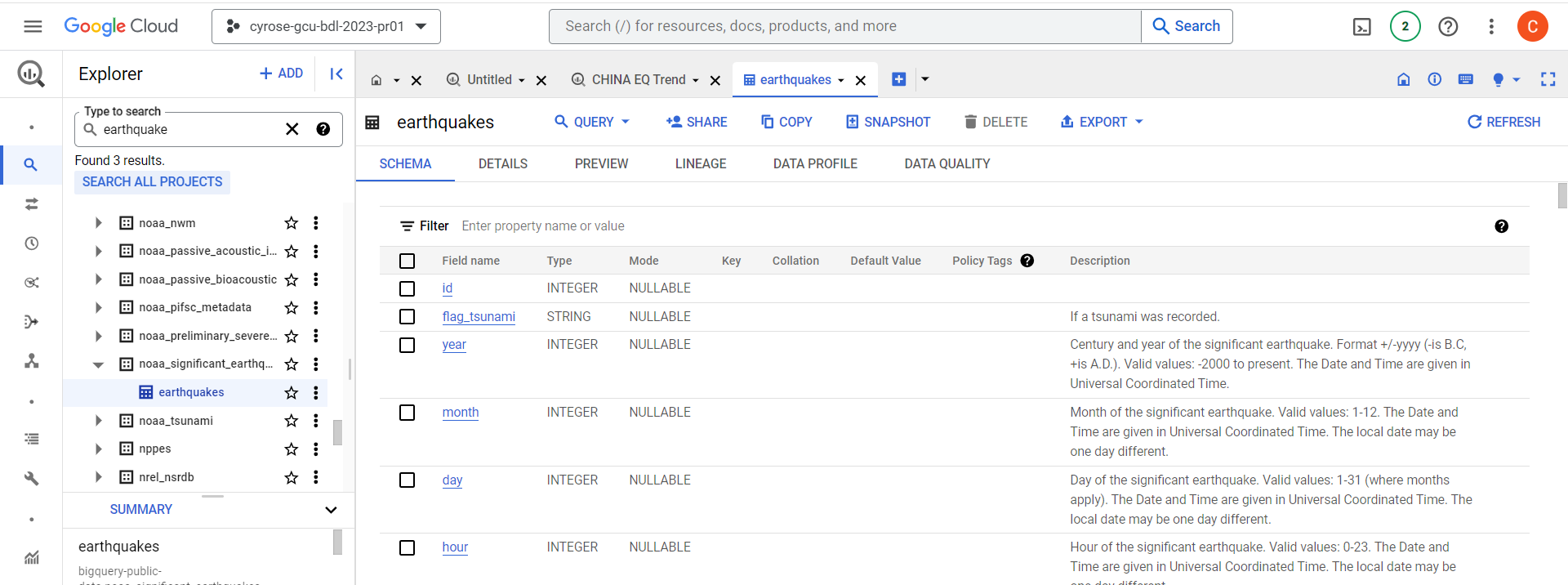
Explore and study your allocated public dataset and provide:

### **A detailed overview, in your own words, of the dataset.**

The earthquake data is a detailed record of historical earthquake occurrences between the 2150 B.C to 2021 A.D in all the continents in the world including various attributes and details about each earthquake event. It offers insights into earthquake occurrences worldwide, including critical data such as the year, month, day, and time of each event. Additionally, it encompasses key seismic measurements, such as primary and moment magnitude values, as well as descriptors for the event's intensity. See below the table information for the data.



The screenshot below shows the schema of the earthquake dataset.



Furthermore, the dataset delves into the geographical context of these earthquakes, specifying the country, state or region where they occurred, along with precise latitude and longitude coordinates. It also provides additional regional insights through region codes, enabling a more comprehensive understanding of the affected areas.

The dataset goes beyond seismic details, incorporating vital societal impact statistics, including the number of deaths, missing persons, injuries, estimated economic damage in millions of dollars, and counts of houses destroyed or damaged. This holistic dataset is a valuable resource for seismic and disaster researchers, facilitating in-depth analysis of earthquake events and their broader consequences.

The following are the respective columns and their descriptions:

|  |  |  |
| --- | --- | --- |
| Column name | Data type | description(the purpose of the column) |
| id | Integer | This is a unique identifier for each earthquake event, presumably for internal tracking or referencing purposes. |
| flag\_tsunami | String | Indicates whether a tsunami was associated with the earthquake. "Tsu" suggests that a tsunami occurred in connection with the earthquake, while blank entries indicate no tsunami. |
| year | Integer | The year in which the earthquake occurred. |
| month | Integer | The month in which the earthquake occurred. |
| day | Integer | The day of the month when the earthquake took place. |
| hour | Integer | The hour of the day when the earthquake occurred. |
| minute | Integer | The minute of the hour when the earthquake took place. |
| second | Integer | The second of the minute when the earthquake happened. |
| focal\_depth | Float | The focal depth of the earthquake, which is the depth within the Earth where the earthquake's energy was released. |
| eq\_primary | Float | The primary magnitude of the earthquake. It appears to be measured in the moment magnitude scale (Mw). |
| eq\_mag\_mw | Float | The earthquake magnitude measured in the moment magnitude scale (Mw). |
| eq\_mag\_ms | Float | The earthquake magnitude measured in the surface wave magnitude scale (Ms). |
| eq\_mag\_mb | Float | The earthquake magnitude measured in the body wave magnitude scale (Mb). |
| eq\_mag\_ml | Float | The earthquake magnitude measured in the local magnitude scale (Ml). |
| eq\_mag\_mfa | Float | The earthquake magnitude measured using the mfa scale. |
| eq\_mag\_unk | Float | The earthquake magnitude measured in an unspecified or unknown scale. |
| intensity | Integer | An intensity value associated with the earthquake, indicating its impact. |
| country | String | The country where the earthquake occurred. |
| state | String | The state or region within the country where the earthquake took place |
| location\_name | String | A descriptive name or location reference for the earthquake. |
| latitude | Float | The latitude coordinates of the earthquake's epicenter. |
| longitude | Float | The longitude coordinates of the earthquake's epicenter. |
| region\_code | Integer | A code or identifier for the specific region where the earthquake occurred. |
| deaths | Integer | The number of deaths resulting from the earthquake |
| deaths\_description | Integer | A description of the severity or impact of the deaths caused by the earthquake. |
| missing | Integer | The number of missing persons as a result of the earthquake. |
| missing\_description | Integer | A description of the severity or impact of the missing persons caused by the earthquake. |
| injuries | Integer | The number of injuries resulting from the earthquake. |
| injuries\_description | Integer | A description of the severity or impact of the injuries caused by the earthquake. |
| Damage\_millions\_  dollars | Float | The estimated damage caused by the earthquake in millions of dollars. |
| damage\_description | Integer | A description of the severity or impact of the damage caused by the earthquake. |
| houses\_destroyed | Integer | The number of houses destroyed due to the earthquake. |
| Houses\_destroyed\_  description | Integer | A description of the severity or impact of the houses destroyed by the earthquake. |
| houses\_damaged | Integer | The number of houses damaged but not completely destroyed by the earthquake. |
| Houses\_damaged\_  description | Integer | A description of the severity or impact of the houses damaged by the earthquake. |
| total\_deaths | Integer | The total number of deaths caused by the earthquake, considering all affected regions. |
| total\_deaths\_description | Integer | A description of the severity or impact of the total deaths caused by the earthquake. |
| total\_missing | Integer | The total number of missing persons caused by the earthquake, considering all affected regions. |
| total\_missing\_description | Integer | A description of the severity or impact of the total missing persons caused by the earthquake. |
| total\_injuries | Integer | The total number of injuries caused by the earthquake, considering all affected regions. |
| total\_injuries\_description | Integer | A description of the severity or impact of the total injuries caused by the earthquake. |
| total\_damage\_millions\_dollars | Float | The total estimated damage caused by the earthquake in millions of dollars, considering all affected regions. |
| total\_damage\_description | Integer | A description of the severity or impact of the total damage caused by the earthquake. |
| total\_houses\_destroyed | Integer | The total number of houses destroyed by the earthquake, considering all affected regions. |
| total\_houses\_destroyed\_description | Integer | A description of the severity or impact of the total houses destroyed by the earthquake. |
| total\_houses\_damaged | Integer | The total number of houses damaged by the earthquake, considering all affected regions. |
| total\_houses\_damaged\_description | Integer | A description of the severity or impact of the total houses damaged by the earthquake. |

### **An appraisal of the quality of the dataset in terms of the integrity rules.**

To assess the quality of the dataset and evaluate its adherence to the integrity rules, I examined various aspects of the dataset, including data completeness, accuracy, consistency, validity, timeliness, integrity, and documentation as follows:

### **Data Completeness**

Completeness assesses whether the dataset has missing values that could lead to biases in an analysis. To check the completeness of the earthquake data I used the ***Completeness*** query:

|  |
| --- |
| SELECT  (SELECT COUNT(\*) FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`) AS total\_rows,  (SELECT COUNT(\*) FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes` WHERE flag\_tsunami IS NULL) AS missing\_flag\_tsunami  ... |

The provided SQL query is using subqueries within a ***SELECT*** statement to retrieve two specific pieces of information from the ***bigquery-public-data.noaa\_significant\_earthquakes.earthquakes*** table. Let's break down the query:

* **Total Rows Count**

***(SELECT COUNT(\*) FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes):*** This subquery calculates the total number of rows in the *earthquakes* table using the ***COUNT(\*)*** aggregation function. It counts all rows in the specified table.

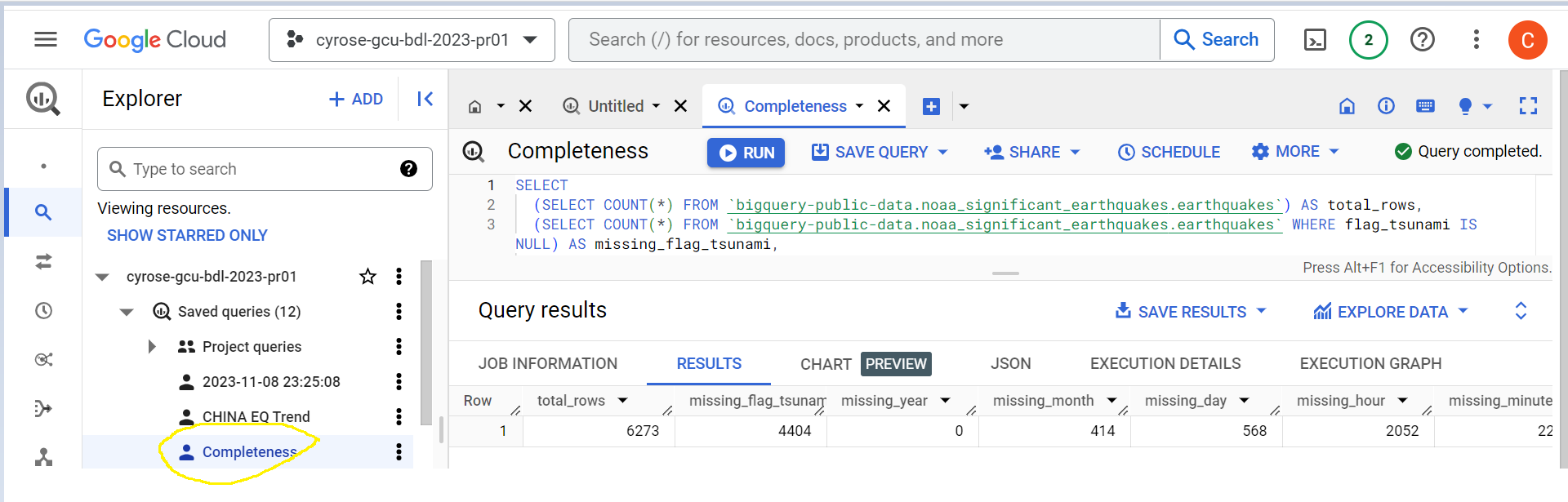
* ***AS total\_rows:*** The result of this subquery is aliased as t***otal\_rows***, providing a label for the column in the output.

|  |
| --- |
| SELECT  (SELECT COUNT(\*) FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`) AS total\_rows, |

* **Count Missing Values for a Specific Column (Example: flag\_tsunami):** This subquery counts the number of rows where the ***flag\_tsunami*** column is *NULL* (it's missing) in the earthquakes table. It aliases this count as ***missing\_flag\_tsunami***. This line of code provides the count of missing values for the flag\_tsunami column.

|  |
| --- |
| (SELECT COUNT(\*) FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes` WHERE flag\_tsunami IS NULL) AS missing\_flag\_tsunami… |

The query continues with similar subqueries for all other columns you want to check for missing values. Each subquery follows the same structure, with a ***COUNT(\*)*** that counts the NULL values in the respective column, and each result is aliased with a name that describes the column being checked.  
  
**Output:**

**

The data is notably incomplete with numerous missing values. Out of 6,273 earthquake occurrences, the variables ***eq\_mag\_mfa*** *(6,259)****,******missing people*** *(6,249)* ***& their descriptions*** *(6,249)* ***,******total\_missing*** *(6,246)****& its descriptions*** *(6,244)**and* ***eq\_mag\_ml*** *(6,088)**recorded the highest number of missing values.*

### **Data Consistency**

For data to be consistent there needs to be very small to no gaps in time when the data was recorded. To check this we check the DISTINCT years in the data.

|  |
| --- |
| SELECT year, COUNT(DISTINCT(month)),  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  GROUP BY year  ORDER BY year DESC; |

The query ***COUNTS*** the number of distinct months ***(month)*** in each unique ***year*** when an earthquake occurrence was recorded. ***FROM*** the***earthquakes*** table, it SELECTS the year and DISTINCT months groups ***(GROUP BY)*** year and orders the results in descending order ***(ORDER BY year DESC). See below the years are consistent.***

******

### **Data Validation**

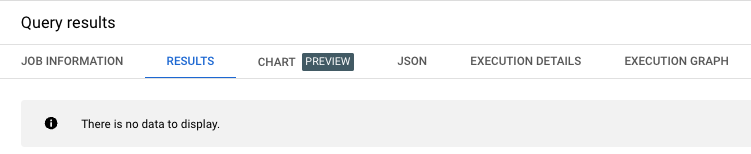
The data contains NULL values that breaks the consistency of the data based on the description given the valid values rules for each column. As a sample case,using the month, day and hour column, the query below identifies and removes the inconsistencies if the values are uniform:

|  |
| --- |
| SELECT \*  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE  -- Check for valid values in the "month" column  (month >= 1 AND year <= 12) AND  -- Check for valid values in the "day" column  (day >= 1 AND year <= 31) AND  -- Check for valid values in the "hour" column  (hour >= 0 AND year <= 23); |

The query SELECTS ***all columns*** FROM the ***earthquake*** table in the ***bigquery-public-data.noaa\_significant\_earthquakes*** database and returns rows that whose values meet all the below conditions:

1. month values between 1 & 12 inclusive.
2. day values between 1 & 31 inclusive.
3. hour values between 0 & 23 inclusive.

The conditions are according to the descriptions given for each variable/column.



**Output: *There will be no data displayed since no entry satisfies all the condition (because of the AND operator)***

### **Data Duplication**

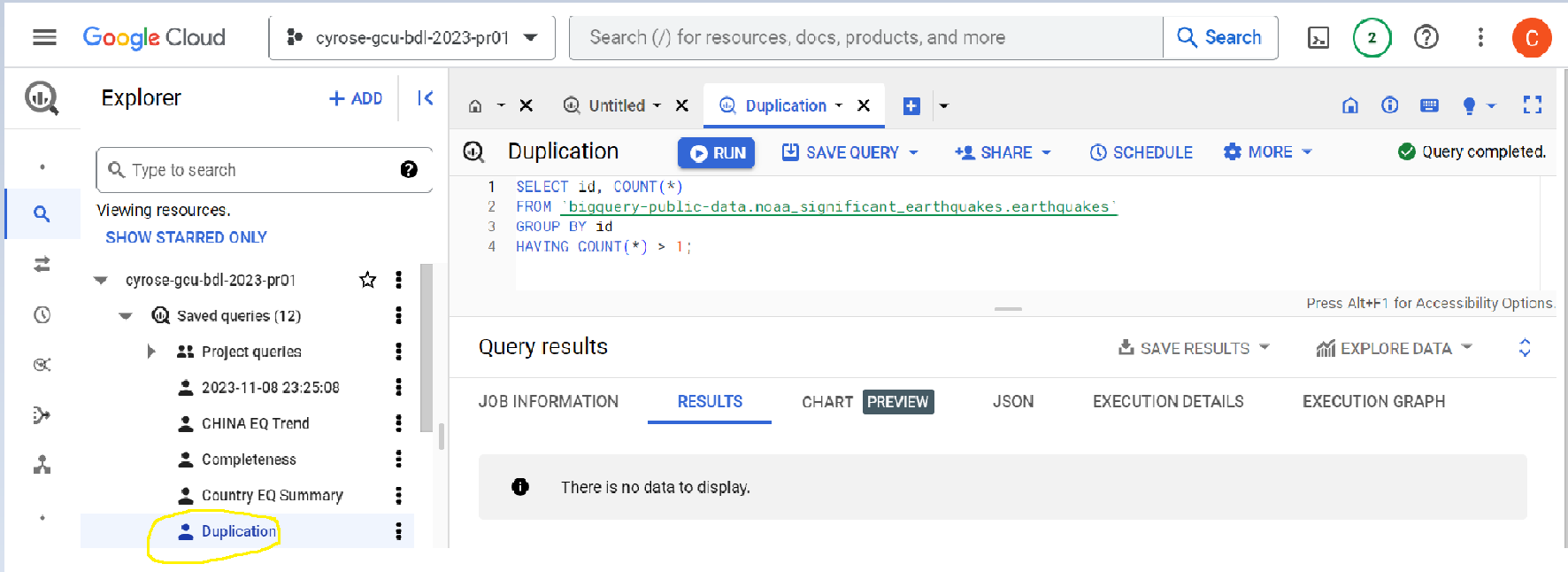
For accurate analysis, I had to check for double entries of earthquake events using the unique identifier for each event using the query:

|  |
| --- |
| SELECT id, COUNT(\*)  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  GROUP BY id  HAVING COUNT(\*) > 1; |

The SQL query is used to identify and count duplicate records in the ***bigquery-public-data.noaa\_significant\_earthquakes.earthquakes*** table. Here's a step-by-step explanation of the query:

* ***SELECT id, COUNT(\*):*** In the SELECT clause, we specify two columns to be included in the result set:
* **id**: This column will be included in the output, allowing us to identify the id values associated with duplicate records.
* **COUNT(\*)**: This is an aggregation function that counts the number of times each id appears in the dataset. It will count the occurrences of each unique id value.
* ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`***: This specifies the table from which the data is being queried.
* **GROUP BY id**: The **GROUP BY** clause is used to group the rows based on the **id** column. This means that all rows with the same id value will be grouped together.
* ***HAVING COUNT(\*) > 1***: The HAVING clause is used to filter the grouped results. In this case, it filters the groups to only include those where the count of occurrences of a particular id (i.e., COUNT(\*)) is greater than 1. This means we are interested in id values that appear more than once, indicating duplicates.

**Output**: ***The data was clear of any duplicates. See below***

******

### **Data Timelines**

From the data documentation, events are only considered relevant if they fall within the period 2000 B.C to present, (i.e -2000 to present or latest date/year recorded). To identify how many valid data are there that fall within the timeline, I used the query:

|  |
| --- |
| SELECT  (SELECT COUNT(\*) FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes` WHERE year >= -2000) AS valid,  (SELECT COUNT(\*) FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes` WHERE year < -2000) AS invalid |

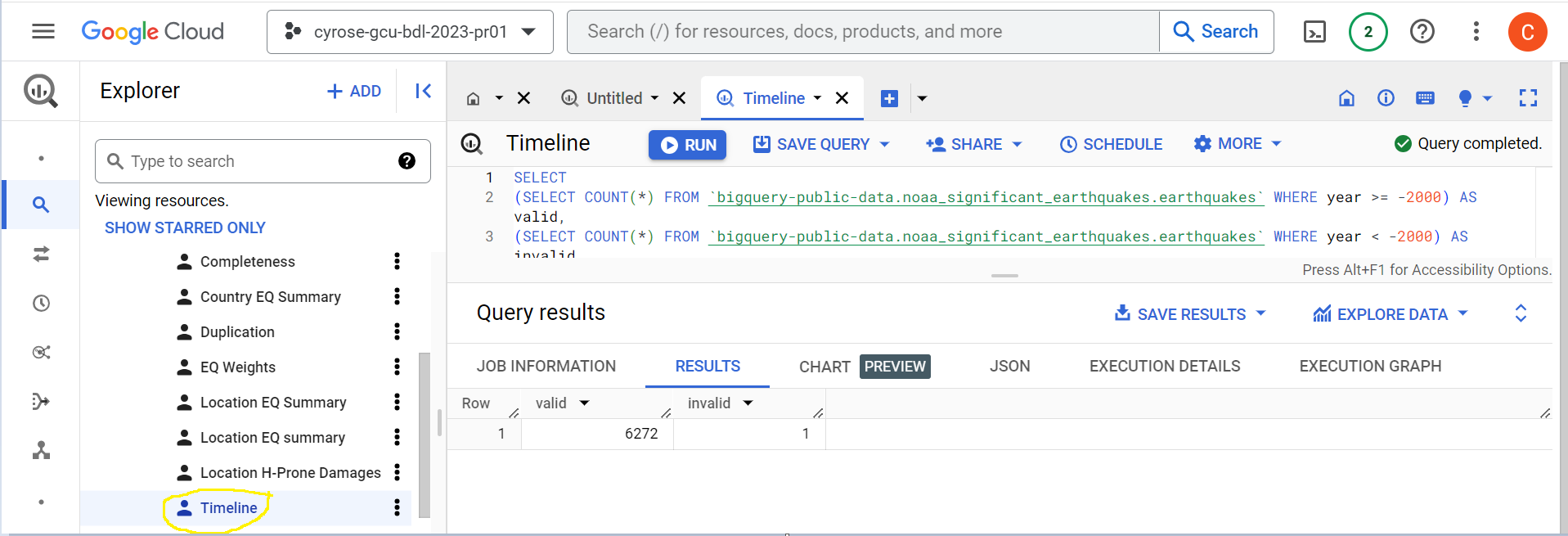
**Query breakdown:**

* ***SELECT Clause***: The SELECT clause is used to specify the columns or expressions that will be included in the result set.
* ***(SELECT COUNT(\*) FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes WHERE year >= -2000):***

This is a subquery that calculates the count of records in the earthquakes table from the noaa\_significant\_earthquakes dataset that meet the specified condition. The subquery consists of the following components:

* + ***SELECT COUNT(\*)***: This part of the subquery counts the number of records (rows) that meet the condition specified in the ***WHERE*** clause.
  + ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`***`: This part indicates the table from which the data is being selected. In this case, it's the *earthquakes table* in the *noaa\_significant\_earthquakes* dataset within the *bigquery-public-data project*.
  + ***WHERE year >= -2000***: This condition filters the records, ensuring that only records with a year value greater than or equal to -2000 are included in the count.
* ***AS valid***: The AS keyword is used to alias the result of the subquery with the name "valid." This means that the count of valid records will be labeled as "valid" in the output. The results of the second subquery will be labeled as “invalid”.

**Output: *All the records are valid except for one; event with id 1 whose year is -2150 (2150 B.C) See below.***

******

**Conclusion:**

***In conclusion, the earthquake data shows strength in validity, accuracy, timelines, duplication and consistency but exhibits lack of completeness. Implementing methods to deal with missing values will enhance data quality and reliability and this will be discussed in task 2.***

1. A single question that you consider, if answered, will provide reasonable value from your dataset. You should include an explanation in your narrative concerning who you consider would benefit from this value and why they would benefit.

***Question***

Which area/location is highly prone to earthquake occurrences?

***Beneficiaries***

Governments

Citizens

Business investors

***Value***

* Given the data's comprehensiveness in terms of geographic variables, encompassing details such as country, state or region, and precise latitude and longitude coordinates, a thorough geographical analysis can offer valuable insights into regions prone to high-intensity earthquakes. By pinpointing these vulnerable areas and gaining a deeper understanding of earthquake occurrence patterns, not only can governments better plan settlements and allocate resources for unexpected damages, but it can also provide essential information for individuals looking to select secure locations for settling. With access to relevant insights, individuals can make informed decisions such as knowing the frequency of an earthquake occurring in a certain location, particularly concerning mortgage considerations, as they seek areas with a lower earthquake risk. Moreover, these insights can guide business investors in identifying strategic locations for their investments.
* The analysis of the earthquake occurrence patterns can assist with providing warnings and alerts to individuals who live near the locations and this can help them decide on the locations to move to.

## **Task 2**

Using your question, undertake the following work on your allocated dataset:

### **Use the Google BigQuery Console to create *all the appropriate SQL code required* to enable you to answer your question and to also deal with any quality issues identified in task 1.**

Geographical analysis of the data:  
In an attempt to answer the question, which of the areas that have earthquake occurrence is less prone to such and better for settlement?, we will answer the following questions:

1. What is the overall view of earthquake occurrences?

Analyze this by filtering out the longitude and latitude and creating a weight column which counts the number of times a location experienced an earthquake using the query:

|  |
| --- |
| SELECT a.latitude, a.longitude, eq\_weight  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes` a  LEFT OUTER JOIN (  SELECT location\_name, COUNT(\*) AS eq\_weight  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE location\_name = location\_name  GROUP BY location\_name) Y  ON a.location\_name = Y.location\_name  WHERE latitude IS NOT NULL AND longitude IS NOT NULL |

*Explanation of query*:

1. ***Main Query (SELECT a.latitude, a.longitude, eq\_weight):***

The main query selects three columns: latitude and longitude from the earthquakes table (***a alias***) and eq\_weight from the subquery result (***Y alias***). This implies that the result will include earthquake locations' coordinates and the calculated earthquake weight (***eq\_weight***) for each unique location.

1. ***Left Outer Join (LEFT OUTER JOIN ... ON a.location\_name = Y.location\_name):***

The query performs a left outer join between the main table (***a***) and a subquery result (***Y***) based on the common column location\_name.

The subquery calculates the count of earthquakes (***eq\_weight***) for each unique location in the original table.

1. ***Subquery (SELECT location\_name, COUNT(\*) AS eq\_weight ... GROUP BY location\_name):***

The subquery selects location\_name and counts the occurrences for each unique location, aliasing the count as eq\_weight.

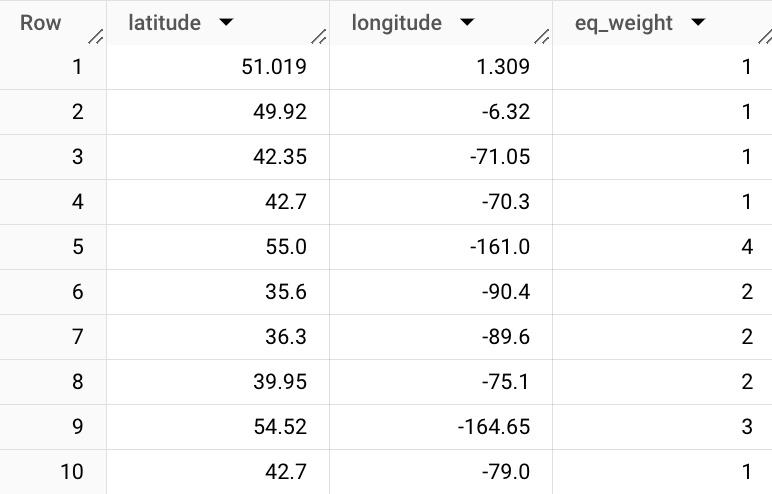
The GROUP BY location\_name ensures that the count is aggregated for each unique location.

***In dealing with the issue of completeness of data there being only two values, entries with empty latitude and corresponding latitude were filtered by the subquery below.***

1. ***WHERE Clause (WHERE latitude IS NOT NULL AND longitude IS NOT NULL):***

The ***WHERE*** clause filters out rows where the ***latitude or longitude is NUL***L, ensuring that only records with valid coordinates are included in the result.

**Output:**

****

1. Which countries have the highest earthquake occurences:

|  |
| --- |
| SELECT country, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE year >= -2000  GROUP BY country  ORDER BY COUNT(\*) DESC  LIMIT 10; |

*Explanation of query*:

1. ***SELECT country, COUNT(\*) AS eq\_count:***

The SELECT statement specifies the columns to be included in the result set.

country: Represents the country column from the earthquakes table.

COUNT(\*) AS eq\_count: Calculates the count of rows for each unique country and aliases it as eq\_count.

1. ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes``:***

Specifies the source table for the data, which is bigquery-public-data.noaa\_significant\_earthquakes.earthquakes.

***In working with the correct timeline the below condition drops records that are before the -2000 (2000 B.C) filtering the correct data required.***

1. ***WHERE year >= -2000:***

Implements a filter condition to include only earthquake records where the year is greater than or equal to -2000. This condition restricts the query to earthquakes occurring after the year 2000 BCE.

1. ***GROUP BY country:***

Groups the results by the country column, ensuring that the count of earthquakes is calculated for each unique country.

1. ***ORDER BY COUNT(\*) DESC:***

Orders the result set in descending order based on the count of earthquakes (eq\_count). This means that countries with a higher count of earthquakes will appear first in the result set.

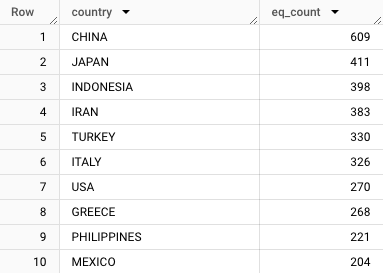
1. ***LIMIT 10:***

Limits the result set to only the top 10 rows. This ensures that the output includes information about the ten countries with the highest counts of significant earthquakes.

***The data had complete records of country data hence no issue in data quality to deal with in this query.***

**Output:**

*The output shows the top ten countries with the highest earthquake occurrences between 2000 B.C and 2021 A.D with* ***China and Japan****, both in the Asia continent, emerging as the most prone countries to earthquakes.*

****

1. What is the general trend of earthquakes in China in the last 15 years?

|  |
| --- |
| SELECT year, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE year >= 2007 AND year <= 2021 AND country = 'CHINA'  GROUP BY year  ORDER BY year; |

***Explanation of query*:**

1. ***SELECT year, COUNT(\*) AS eq\_count***:

This part selects two columns from the dataset: ***year*** and the ***count of rows for each year*** (total earthquakes in that year), aliased as eq\_count.

1. ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes``***:

It specifies the dataset and table (noaa\_significant\_earthquakes.earthquakes) from which the data is being retrieved. This table likely contains information about significant earthquakes, including their year.

1. ***WHERE year >= 2007 AND year <= 2021 AND country = 'CHINA':***

This filters the data to include earthquakes only from the years 2007 to 2021 that occurred in China. It restricts the results to earthquakes within this time frame and in the specified country.

1. ***GROUP BY year***:
2. This groups the results by the year column, which means that the count of earthquakes (eq\_count) will be aggregated for each unique year.
3. ***ORDER BY year***:This arranges the output rows in ascending order based on the year column.

***According to the description given, only one record had to be dropped because it was recorded in the year 2150 B.C exceeding the 2000 B.C limit expected in the data.***

**Output:**

China has registered inconsistent earthquake occurrences. A gradual decrease was however from 2019 to 2021.



1. What regions of China experience high earthquake occurences?

|  |
| --- |
| SELECT location\_name, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE country = 'CHINA'  GROUP BY location\_name  ORDER BY COUNT(\*) DESC  LIMIT 10; |

*Explanation of query*:

1. ***SELECT year, COUNT(\*) AS eq\_count:***

* The SELECT statement specifies the columns to be included in the result set.
* year: Represents the year column from the earthquakes table.
* COUNT(\*) AS eq\_count: Calculates the count of rows for each unique year and aliases it as eq\_count.

1. ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes``:***

Specifies the source table for the data, which is bigquery-public-data.noaa\_significant\_earthquakes.earthquakes.

1. ***WHERE year >= 2007 AND year <= 2021 AND country = 'CHINA':***

Implements filter conditions to include only earthquake records where:

* The year is greater than or equal to 2007.
* The year is less than or equal to 2021.
* The country is 'CHINA'. These conditions narrow down the scope to significant earthquakes that occurred in China within the specified year range.

1. ***GROUP BY year:***

Groups the results by the year column, ensuring that the count of earthquakes is

calculated for each unique year.

1. ***ORDER BY year:***

Orders the result set in ascending order based on the year. This ensures that the output is sorted chronologically.

***The data had complete records of location data hence no issue in data quality to deal with in this query.***

**Output:**

***Yunnan province*** is the most prone location to earthquakes with 68 occurrences followed by Sichuan province with 44 occurrences.

****

**Output:**

1. Monthly frequency of earthquakes occurrences in Yunnan.

|  |
| --- |
| SELECT month, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE location\_name LIKE '%YUNNAN%'  GROUP BY month  ORDER BY month; |

Explanation:

1. ***SELECT month, COUNT(\*) AS eq\_count:***

* The SELECT statement specifies the columns to be included in the result set.
* month: Represents the month column from the earthquakes table.
* COUNT(\*) AS eq\_count: Calculates the count of rows for each unique month and aliases it as eq\_count.

1. ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes``:***

* Specifies the source table for the data, which is ***bigquery-public-data.noaa\_significant\_earthquakes.earthquakes.***

1. ***WHERE location\_name LIKE '%YUNNAN%':***

Implements a filter condition to include only earthquake records where the location\_name column contains the substring 'YUNNAN'. The % signs are wildcards, allowing for any characters before and after 'YUNNAN' in the location name.

1. ***GROUP BY month:***

Groups the results by the month column, ensuring that the count of earthquakes is calculated for each unique month.

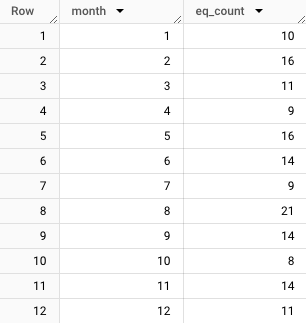
1. ***ORDER BY month:***

Orders the result set in ascending order based on the month. This ensures that the output is sorted chronologically.

***The data had complete records of month data hence no issue in data quality to deal with in this query.***

**Output:**

Yunnan tends to have high earthquake occurrences in the months of February (2), June (5), July (6) and September (9).



1. **Magnitude** and **depth** are two basic features of an earthquake that are important for understanding plate tectonics as well as **earthquake hazard**. Typically, the *shallower the earthquake and larger the magnitude, the more potential for destruction.*

Which particular occurrences in China likely faced more destruction based on **Magnitude** and **depth?**

|  |
| --- |
| SELECT id, location\_name, focal\_depth, eq\_primary, total\_deaths, total\_deaths\_description  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE year >= -2000 AND country = 'CHINA' AND focal\_depth IS NOT NULL  ORDER BY focal\_depth ASC, eq\_primary DESC; |

*Explanation of query*:

1. ***SELECT id, location\_name, focal\_depth, eq\_primary, total\_deaths, total\_deaths\_description:***

* The **SELECT** statement specifies the columns to be included in the result set.
* ***id, location\_name, focal\_depth, eq\_primary, total\_deaths, and total\_deaths\_description*** are selected columns.

1. ***FROM bigquery-public-data.noaa\_significant\_earthquakes.earthquakes``:***

* Specifies the source table for the data, which is *bigquery-public-data.noaa\_significant\_earthquakes.earthquakes*.

In dealing with completeness and timeliness of records the below subquery selected records above 2000 B.C and dropped the missing values selecting only focal\_depth with filled values.

1. ***WHERE year >= -2000 AND country = 'CHINA' AND focal\_depth IS NOT NULL:***

Implements filter conditions to include only earthquake records where:

* The year is greater than or equal to -2000.
* The country is 'CHINA'.
* The focal\_depth is not NULL.

These conditions narrow down the scope to significant earthquakes in China with available focal depth information.

1. ***ORDER BY focal\_depth ASC, eq\_primary DESC:***

* Orders the result set based on the specified columns and sorting directions.
* ***focal\_depth ASC***: Sorts the result set in ascending order based on the focal\_depth column.
* ***eq\_primary DESC***: Within the ascending order of focal\_depth, further sorts the result set in descending order based on the eq\_primary column.

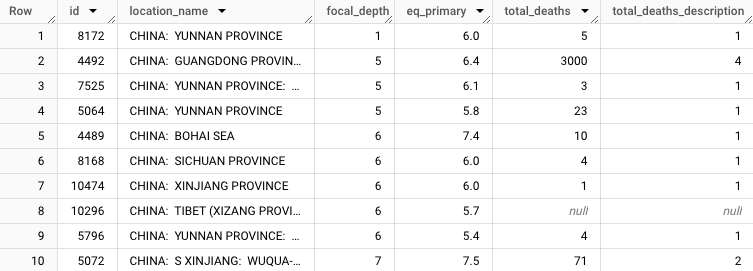
**Output:**

By:

i. Elimination the null values in the focal depth &

ii. Arranging the focal depth in Ascending order and primary magnitude in Descending

…Yunnan province and Guangdong province occurrences with id 8172, 4492, 7525 and 5064 were identified with likelihood of more potential for destruction. The flagged events however did not record high deaths as expected except for the Guangdong event of id 4492. Yunnan province, however, registered multiple events with magnitudes greater than 5.0.



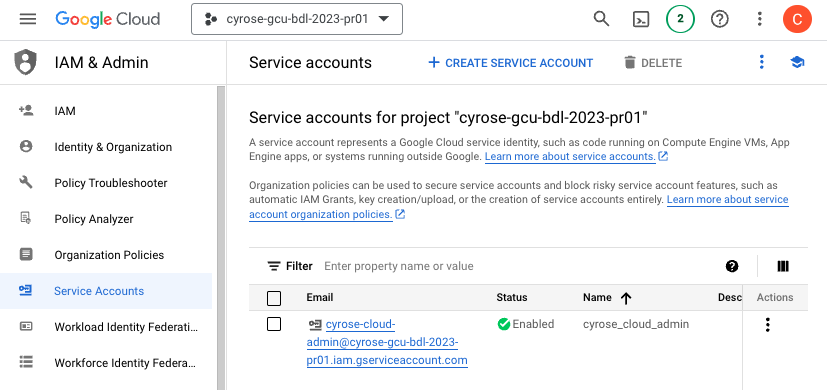
**Conclusion**

**China's emergency funds should be greater than any nation seeing the surge of the earthquake events. More particularly, locations in Yunnan province should be the main focus for residents and the government especially in the months of February, June, July and September where earthquakes with low focal depths and high magnitudes are likely to cause more damage.**

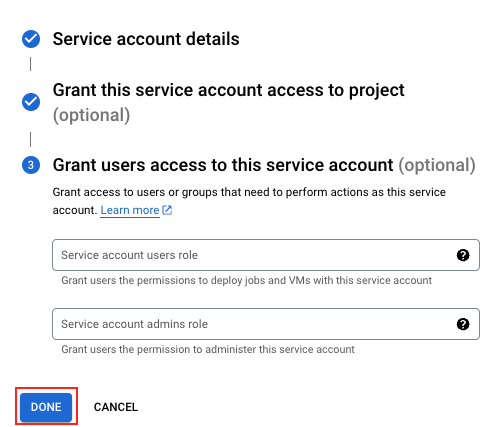
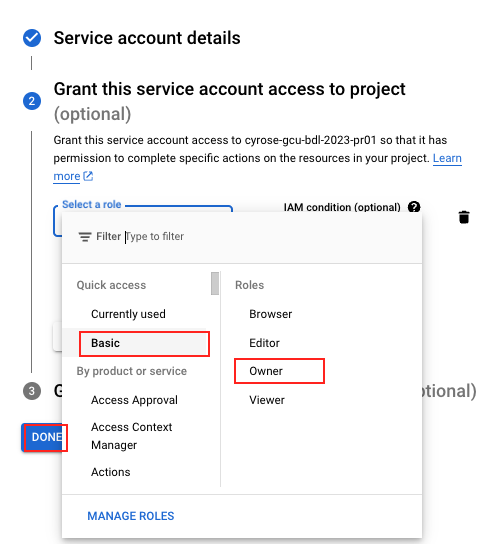
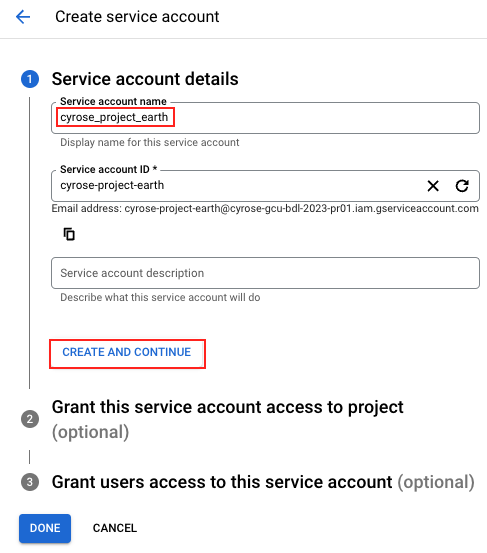
### **Create a Google Colab notebook that includes your SQL code as well as Python code that allows your SQL code to be executed using BigQuery. Your Python code must also store the information obtained that answers your question in Google Cloud Storage.**

Before creating the Python code:  
**A**. we must follow the ***acquire a Google Cloud Platform (GCP) service account key*** to access GCP services programmatically This is generated through the following steps:

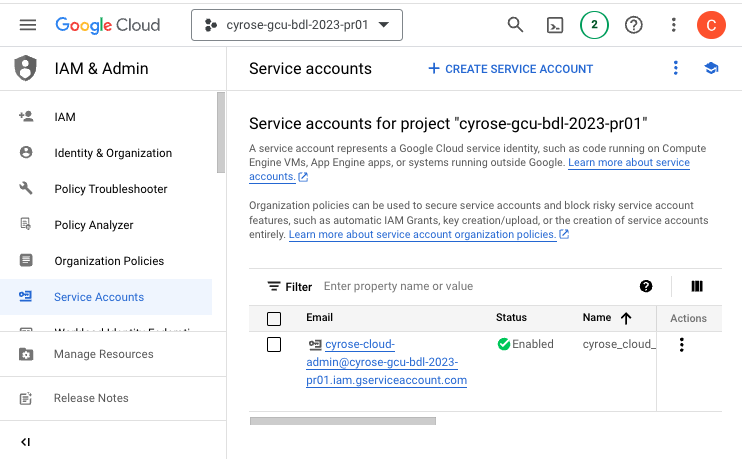
On the left GCP menu accessed through the hamburger menu Go to **IAM & Admin**  **> Service Accounts.**On the Service accounts page click **+CREATE SERVICE ACCOUNT**

****

A “create service account” page will open and there ***input the name of the service account*** then click ***CREATE AND CONTINUE.*** On the select role dropdown menu select ***Basic t****hen* ***owner.* This will give the Python Colab code all access to the GCP service.** Proceed and click ***DONE.***

****

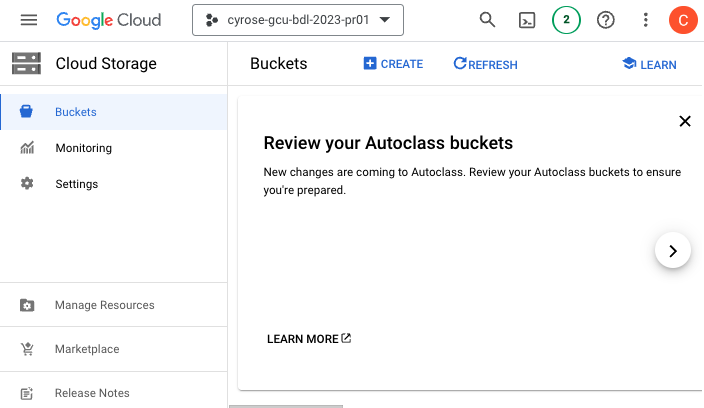
Now we have a service account with an activated API key which is downloaded automatically as a .JSON file. This is the file we will use in Authenticating our Python-based app that will execute the BigQuery SQL queries.

****

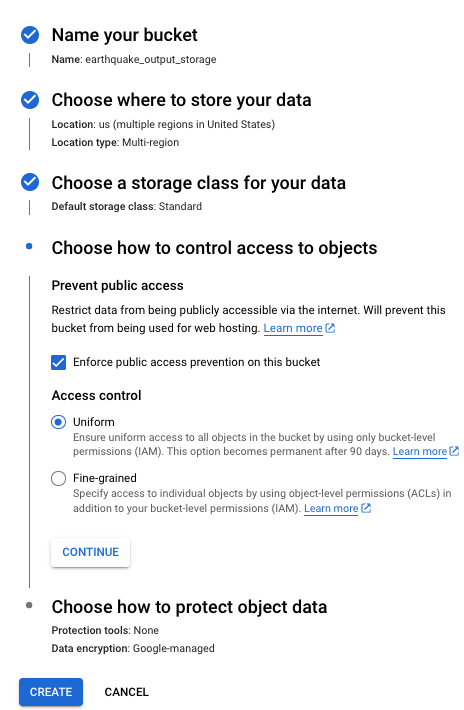
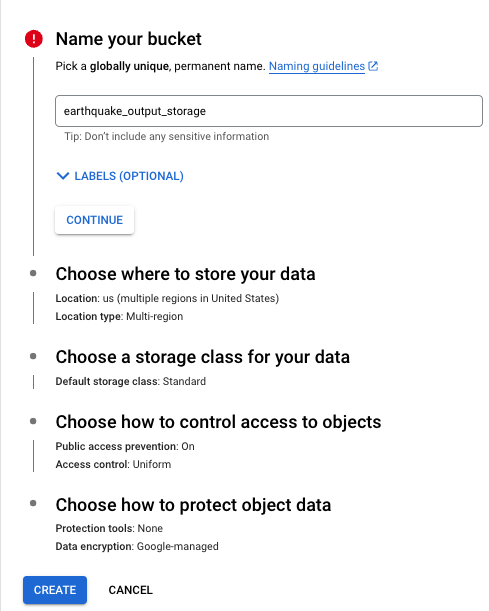
**B. Create a *bucket in the Google Cloud Storage*** where Python-Based App will store the data from the queries. The following steps are to be followed:

On the left GCP menu accessed through the hamburger menu Go to **Cloud Storage**  **> Bucket.**

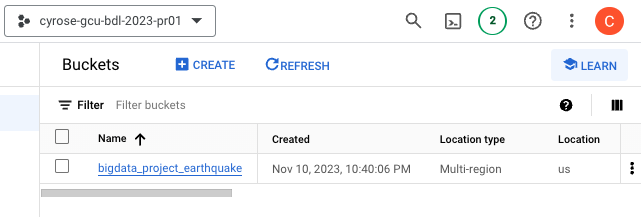
On the Bucket page, initiate the creation of a new bucket by clicking **+CREATE**

****

Give a name to the bucket storage *“earthquake\_storage\_output”* then click ***continue***. The other variables should be remain as they are then click ***“create”***



Now we have a storage platform for the App.



With the ***service account API key*** *and the* ***bucket storage*** we can now create the Python App.

Procedures for creating the two have been guided from Google [Create service key](https://cloud.google.com/iam/docs/service-accounts-create) and [Create bucket](https://cloud.google.com/storage/docs/creating-buckets) documentation.

[Google Colab\_Python Code](https://colab.research.google.com/drive/1x_T11ddf677UF2psTiRT_r-HcbNxQqmZ?usp=sharing).

In setting up the Python Environment the following modules will be used in the app.

|  |
| --- |
| import os  import json  import pandas as pd  from google.cloud import storage  from google.cloud import bigquery  import plotly.express as px  from ipyleaflet import Map, basemaps, Marker, MarkerCluster |

The below code mounts both the google storage and the create a python instance that allows the python app to run the Big Queries from the python app.

|  |
| --- |
| # Import Google service account API Key for Authentication  os.environ['GOOGLE\_APPLICATION\_CREDENTIALS'] = 'cyrose-gcu-bdl-acct.json'  # Accessing the bucket in the storage  storage\_client = storage.Client()  bucket = storage\_client.get\_bucket('bigdata\_project\_earthquake')  # For running queries  client = bigquery.Client() |

## **Task 3**

### **Visualize the information, now present in Google Cloud Storage, in a manner of your choosing, that you consider clearly communicates your findings and the value obtained.**

The codes included in this task produce visuals for each question in Task 2, Question 1 using the specific SQL query.

The syntax in each code is as follows:

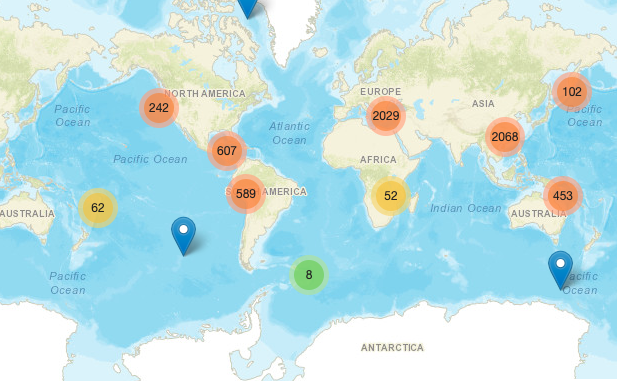
|  |
| --- |
| **name\_of\_query** = client.query('''  **{insert specific query in Task 1}**  ''').result().to\_dataframe()  # storage the data in Google Cloud Storage:  bucket.blob('**name\_of\_data.csv**').upload\_from\_string(**name\_of\_query**.to\_csv(), 'text/csv')  # show sample results  **name\_of\_query**.head()  **#Type of Graph Example:**  *fig = px.bar(eq\_country\_query, x='country', y='eq\_count', text\_auto=True)*  *fig.show()* |

The following are the visuals for each question?

1. What is the overall view of earthquake occurrences?

**Code:**

|  |
| --- |
| map\_query = client.query('''  SELECT a.latitude, a.longitude, eq\_weight  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes` a  LEFT OUTER JOIN (  SELECT location\_name, COUNT(\*) AS eq\_weight  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE location\_name = location\_name  GROUP BY location\_name) Y  ON a.location\_name = Y.location\_name  WHERE latitude IS NOT NULL AND longitude IS NOT NULL  ''').result().to\_dataframe()  # Write dataframe to google storage:  bucket.blob('geographical\_eq\_count.csv').upload\_from\_string(map\_query.to\_csv(), 'text/csv')  map\_query.head()  #Visual Map  eq\_map = Map(basemap = basemaps.Esri.WorldStreetMap)  marker = []  for i in range(len(map\_query)):  marker.append([map\_query.iloc[i]['latitude'],  map\_query.iloc[i]['longitude']])  list\_eq = [Marker(location = x, draggable = False) for x in marker]  eq\_map.add\_layer(MarkerCluster(markers = list\_eq));  display(eq\_map) |



***As observed, Asia has the highest number of earthquake occurrences.***

1. Which countries have the highest earthquake occurrences?`

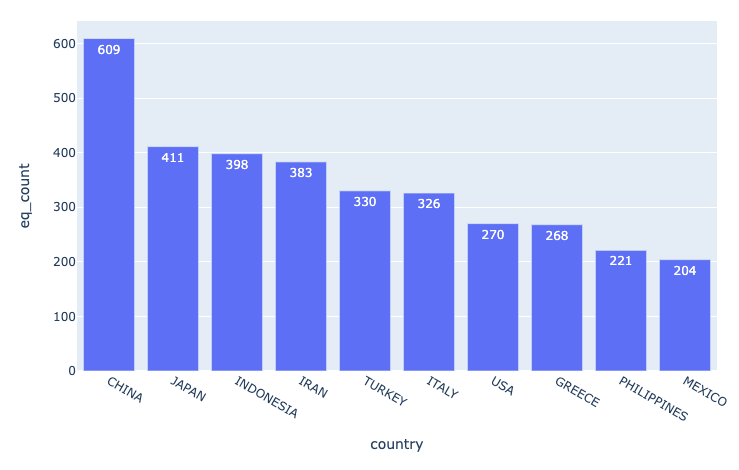
Accessed by counting the number of occurrences for each country?

The query gives results of the top 10 countries with the highest earthquake occurrences.

**Code:**

|  |
| --- |
| eq\_country\_query = client.query('''  SELECT country, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE year >= -2000  GROUP BY country  ORDER BY COUNT(\*) DESC  LIMIT 10;  ''').result().to\_dataframe()  # Write dataframe to google storage:  bucket.blob('country\_eq\_count.csv').upload\_from\_string(eq\_country\_query.to\_csv(), 'text/csv')  eq\_country\_query.head()  #Bar graph  fig = px.bar(eq\_country\_query, x='country', y='eq\_count', text\_auto=True)  fig.show() |

**Output:**



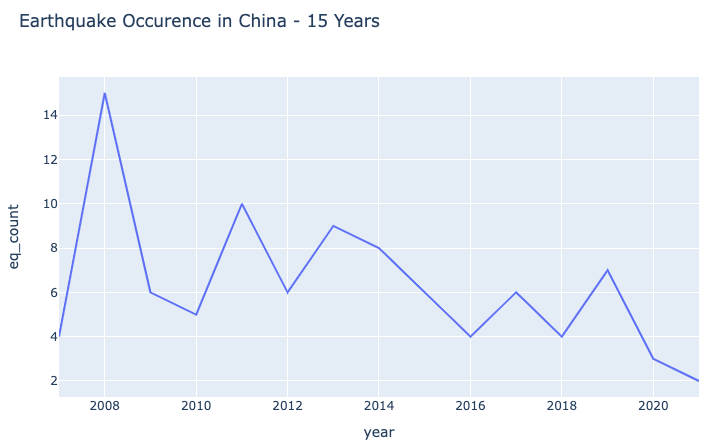
***The continent of Asia tops with the highest number of earthquake occurrences. This is because China and Japan have the highest number of earthquake occurrences with 609 and 411 occurrences respectively.***

1. What is the general trend of earthquakes in China in the last 15 years?

**Code:**

|  |
| --- |
| eq\_china\_query = client.query('''  SELECT year, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE year >= 2007 AND year <= 2021 AND country = 'CHINA'  GROUP BY year  ORDER BY year;  ''').result().to\_dataframe()  # Write dataframe to google storage:  bucket.blob('china\_eq\_count\_by\_year.csv').upload\_from\_string(eq\_china\_query.to\_csv(), 'text/csv')  eq\_china\_query.head()  #Line Graph  fig = px.line(eq\_china\_query,  x="year", y="eq\_count",  title='Earthquake Occurence in China - 15 Years',  width=800, height=500)  fig.show() |

**Output:**



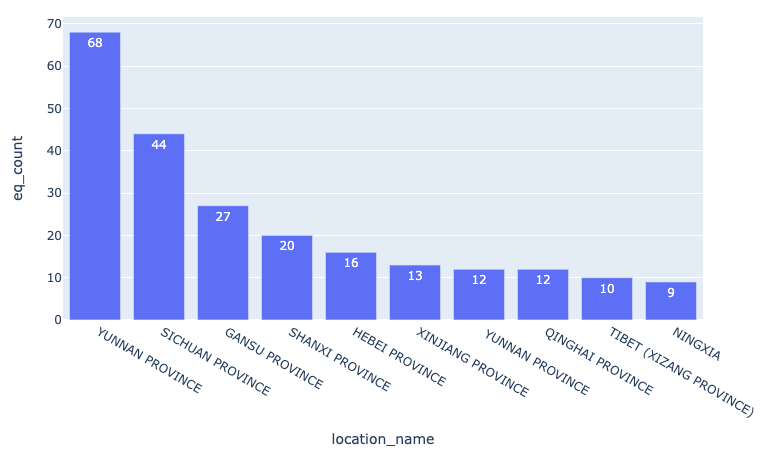
***China recorded the highest number of earthquakes in 2008. There has been a gradual decrease in earthquake activity since 2008 to 2021(latest year in the data.)***

1. What regions of China experience high earthquake occurrences?

**Code:**

|  |
| --- |
| eq\_china\_location\_query = client.query('''  SELECT location\_name, COUNT(\*) AS eq\_count  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE country = 'CHINA'  GROUP BY location\_name  ORDER BY COUNT(\*) DESC  LIMIT 10;  ''').result().to\_dataframe()  eq\_china\_location\_query['location\_name'] = eq\_china\_location\_query['location\_name'].str.replace('CHINA: ', '')  # Write dataframe to google storage:  bucket.blob('china\_location\_eq\_count.csv').upload\_from\_string(eq\_china\_location\_query.to\_csv(), 'text/csv')  eq\_china\_location\_query.head()  # Line Graph  fig = px.bar(eq\_china\_location\_query, x="location\_name", y="eq\_count", text\_auto=True, width=800, height=500)  fig.show() |

**Output:**



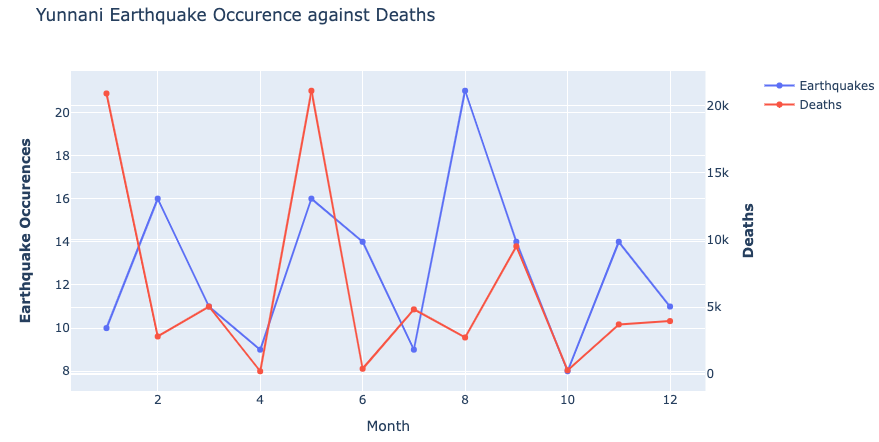
***Yunnan province is more prone to earthquakes than any province in China.***

1. What is the frequency of earthquakes occurrences compared to deaths per month in Yunnan Province?

**Code:**

|  |
| --- |
| eq\_yunnani\_month\_query = client.query('''  SELECT month, COUNT(\*) AS eq\_count,SUM(total\_deaths) AS deaths  FROM `bigquery-public-data.noaa\_significant\_earthquakes.earthquakes`  WHERE location\_name LIKE '%YUNNAN%' AND month IS NOT NULL  GROUP BY month  ORDER BY month;  ''').result().to\_dataframe()  # Write dataframe to google storage:  bucket.blob('yunnan\_deaths\_eq\_count\_by\_month.csv').upload\_from\_string(eq\_yunnani\_month\_query.to\_csv(), 'text/csv')  eq\_yunnani\_month\_query.head()  #Line Graph  # Create figure with secondary y-axis  fig = make\_subplots(specs=[[{"secondary\_y": True}]])  fig.add\_trace(  go.Scatter(x= eq\_yunnani\_month\_query['month'], y= eq\_yunnani\_month\_query['eq\_count'], name="Earthquakes"),  secondary\_y=False, )  fig.add\_trace(  go.Scatter(x = eq\_yunnani\_month\_query['month'], y= eq\_yunnani\_month\_query['deaths'], name="Deaths"),  secondary\_y=True, )  fig.update\_layout(  title\_text="Yunnani Earthquake Occurrence against Deaths"  )  # Set x-axis title  fig.update\_xaxes(title\_text="Month")  # Set y-axes titles  fig.update\_yaxes(title\_text="<b>Earthquake Occurrences</b>", secondary\_y=False)  fig.update\_yaxes(title\_text="<b>Deaths</b>", secondary\_y=True)  fig.show() |

**Output:**



Generally, **Yunnan Province** experiences a high number of earthquakes in August. In the month of July, there was an increment in both earthquakes and deaths.