

# PRA02 - BigQuery en Python

---

## Scenario

You're part of a public health organization which is tasked with identifying answers to queries related to the Covid-19 pandemic. Obtaining the right answers will help the organization in planning and focusing healthcare efforts and awareness programs appropriately.

The dataset and table that will be used for this analysis will be : `bigquery-public-data.covid19_open_data.covid19_open_data`. This repository contains country-level datasets of daily time-series data related to COVID-19 globally. It includes data relating to demographics, economy, epidemiology, geography, health, hospitalizations, mobility, government response, and weather.

---

## Query 1: Total Confirmed Cases

Build a query that will answer "What was the total count of confirmed cases on Apr 15, 2020?" The query needs to return a single row containing the sum of confirmed cases across all countries. The name of the column should be **total\_cases\_worldwide**.

---

## Query 2: Worst Affected Areas

Build a query for answering "How many states in the US had more than 100 deaths on Apr 10, 2020?" The query needs to list the output in the field **count\_of\_states**. **Hint:** Don't include NULL values.

---

## Query 3: Identifying Hotspots

Build a query that will answer "List all the states in the United States of America that had more than 1000 confirmed cases on Apr 10, 2020?" The query needs to return the State Name and the corresponding confirmed cases arranged in descending order. Name of the fields to return **state** and **total\_confirmed\_cases**.

---

## Query 4: Fatality Ratio

Build a query that will answer "What was the case-fatality ratio in Italy for the month of April 2020?" Case-fatality ratio here is defined as (total deaths / total confirmed cases) \* 100. Write a query to return the ratio for the month of April 2020 and containing the following fields in the output: **total\_confirmed\_cases**, **total\_deaths**, **case\_fatality\_ratio**.

---

### Query 5: Identifying specific day

Build a query that will answer: "On what day did the total number of deaths cross 10000 in Italy?" The query should return the date in the format **yyyy-mm-dd**.

---

### Query 6: Finding days with zero net new cases

The following query is written to identify the number of days in India between 21 Feb 2020 and 15 March 2020 when there were zero increases in the number of confirmed cases. However it is not executing properly. You need to update the query to complete it and obtain the result:

```
WITH india_cases_by_date AS (  
  SELECT  
    date,  
    SUM(cumulative_confirmed) AS cases  
  FROM  
    `bigquery-public-data.covid19_open_data.covid19_open_data`  
  WHERE  
    country_name="India"  
    AND date between '2020-02-21' and '2020-03-15'  
  GROUP BY  
    date  
  ORDER BY  
    date ASC  
)  
  
, india_previous_day_comparison AS  
(SELECT  
  date,  
  cases,  
  LAG(cases) OVER(ORDER BY date) AS previous_day,  
  cases - LAG(cases) OVER(ORDER BY date) AS net_new_cases  
FROM india_cases_by_date  
)
```

---

### Query 7: Doubling rate

Using the previous query as a template, write a query to find out the dates on which the confirmed cases increased by more than 10% compared to the previous day (indicating doubling rate of ~ 7 days) in the US between the dates March 22, 2020 and April 20, 2020. The query needs to return the list of dates, the confirmed cases on that day, the confirmed cases the

previous day, and the percentage increase in cases between the days. Use the following names for the returned fields: **Date**, **Confirmed\_Cases\_On\_Day**, **Confirmed\_Cases\_Previous\_Day** and **Percentage\_Increase\_In\_Cases**.

---

## Query 8: Recovery rate

Build a query to list the recovery rates of countries arranged in descending order (limit to 10) on the date May 10, 2020. Restrict the query to only those countries having more than 50K confirmed cases. The query needs to return the following fields: **country**, **recovered\_cases**, **confirmed\_cases**, **recovery\_rate**.

---

## Query 9: CDGR - Cumulative Daily Growth Rate

The following query is trying to calculate the CDGR on May 10, 2020 (Cumulative Daily Growth Rate) for France since the day the first case was reported. The first case was reported on Jan 24, 2020. The CDGR is calculated as:

```
((last_day_cases/first_day_cases)^1/days_diff)-1)
```

Where :

- **last\_day\_cases** is the number of confirmed cases on May 10, 2020
- **first\_day\_cases** is the number of confirmed cases on Feb 02, 2020
- **days\_diff** is the number of days between Feb 02 - May 10, 2020

The query isn't executing properly. Can you fix the error to make the query execute successfully?

```
WITH
  france_cases AS (
    SELECT
      date,
      SUM(cumulative_confirmed) AS total_cases
    FROM
      `bigquery-public-data.covid19_open_data.covid19_open_data`
    WHERE
      country_name="France"
      AND date IN ('2020-01-24',
        '2020-05-10')
    GROUP BY
      date
    ORDER BY
      date)
, summary as (
  SELECT
    total_cases AS first_day_cases,
    LEAD(total_cases) AS last_day_cases,
    DATE_DIFF(LEAD(date) OVER(ORDER BY date),date, day) AS days_diff
  FROM
```

```
france_cases
LIMIT 1
)

select first_day_cases, last_day_cases, days_diff,
SQRT((last_day_cases/first_day_cases),(1/days_diff))-1 as cdgr
from summary
```

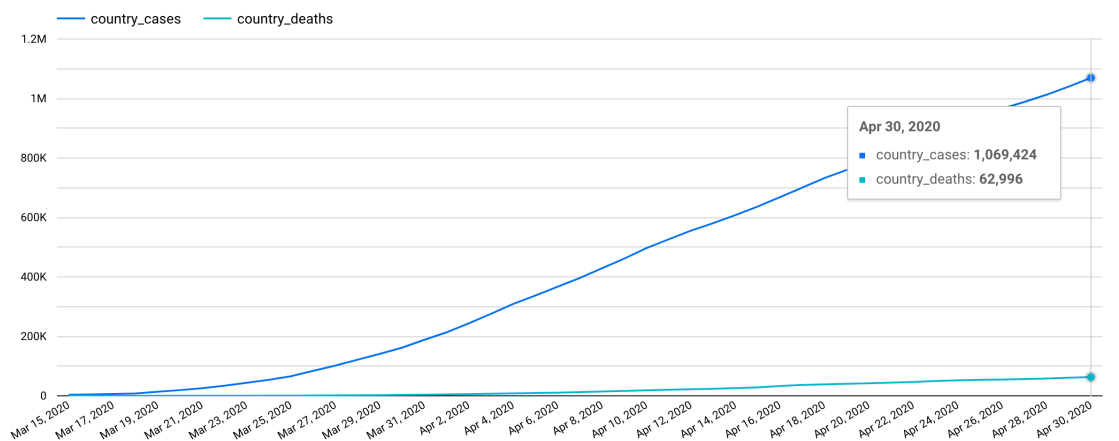
**Note:** Refer to the following [page](#) to learn more about the SQL function referenced `LEAD()`.

## Create a Datastudio report

Create a Datastudio report that plots the following for the United States:

- Number of Confirmed Cases
- Number of Deaths
- Date range : 2020-03-15 to 2020-04-30

**Hint:** Use the below image as a reference while building the report and make sure your report matches it.



## Congratulations!

In this lab you used BigQuery to query the public Covid dataset (`bigquery-public-data.covid19_open_data.covid19_open_data`) to answer important questions related to the Covid-19 pandemic. You also created a Datastudio report for one of the query outputs to visualize the cases and deaths trend.