



# Task 2 Corona Virus Analysis with SQL

---

BY: JENISH JAI

PROFILE: DATA ANALYST INTERN

BATCH: MIP-DA-03

# Project Overview

---

- The COVID-19 pandemic has prompted a critical need for data-driven insights to comprehend the virus's spread.
- As a data analyst, the task at hand is to analyze a COVID-19 dataset to extract valuable insights.
- Through data exploration and statistical analysis, the aim is to uncover patterns, trends, and correlations within the dataset.
- By presenting these findings, the agenda is to contribute to the collective effort in combating the COVID-19 crisis.



# Dataset Overview

Description of each column in the dataset ([Corona Virus Dataset](#))

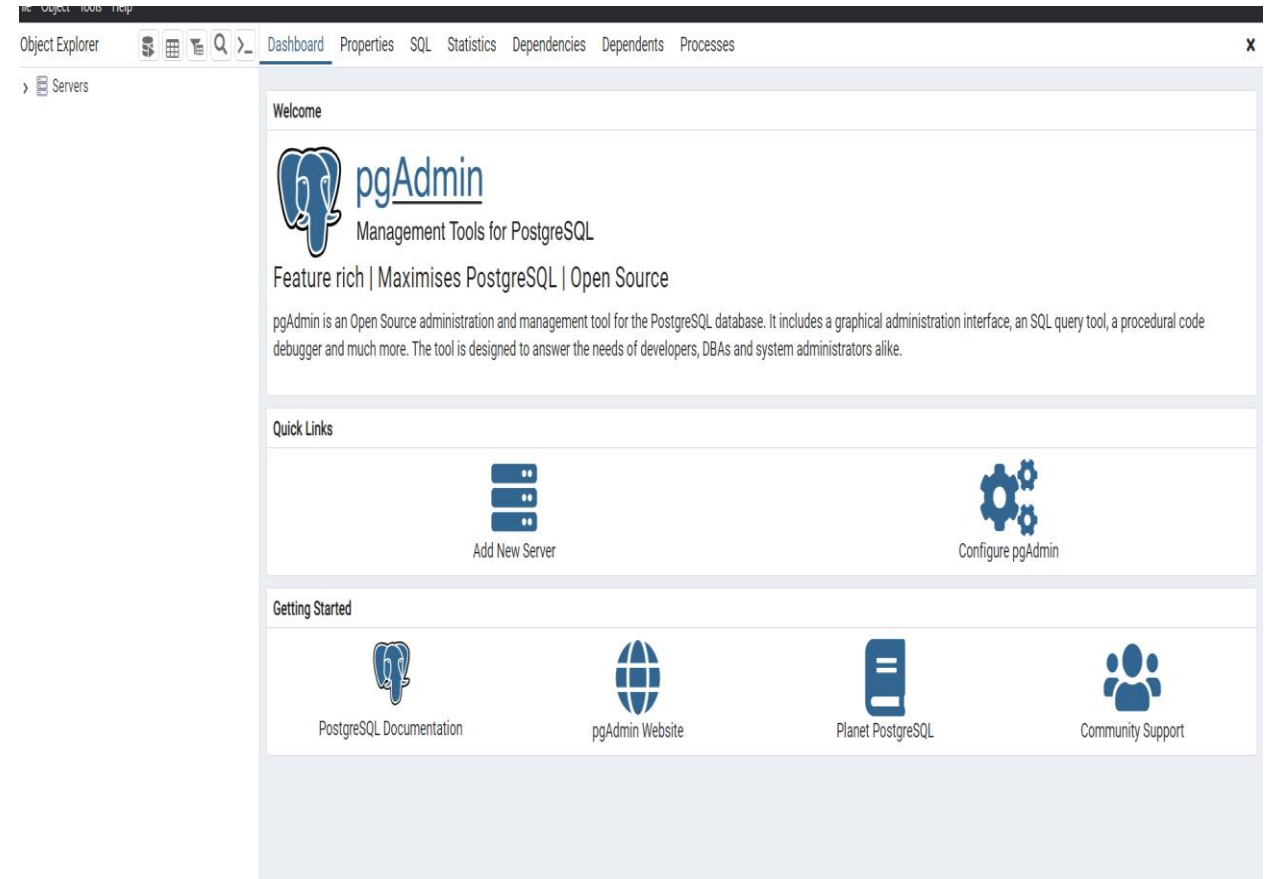
- **Province:** Geographic subdivision within a country/region.
- **Country\_Region:** Geographic entity where data is recorded.
- **Latitude:** North-south position on Earth's surface.
- **Longitude:** East-west position on Earth's surface.
- **Date:** Recorded date of CORONA VIRUS data.
- **Confirmed:** Number of diagnosed CORONA VIRUS cases.
- **Deaths:** Number of CORONA VIRUS-related deaths.
- **Recovered:** Number of recovered CORONA VIRUS cases

# Database Setup and Tool Used

## Database Used:



## Management Tool: pgAdmin 4



# Database Setup

To create “COVID ANALYSIS” database :

The image illustrates the process of creating a new database in a database management tool. It is divided into two main parts by a large black arrow pointing from left to right.

**Left Part (Database Explorer):** This section shows a tree view of the database structure. Under 'Servers (1)', there is a 'PostgreSQL 16' server. Under 'PostgreSQL 16', there is a 'Databases (5)' folder, which is currently selected. Below 'Databases (5)', there are several databases listed: 'ABC', 'Casts', 'Catalogs', and 'Event Triggers'. A context menu is open over the 'Databases (5)' folder, showing two options: 'Create' and 'Refresh'. The 'Create' option is highlighted, and a sub-menu is visible with the option 'Database...'. A large black arrow points from this 'Database...' option towards the right part of the image.

**Right Part (Create - Database Dialog):** This section shows a dialog box titled 'Create - Database'. The dialog has several tabs: 'General', 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. The 'General' tab is selected. The dialog contains the following fields:

- Database:** A text field containing the name 'COVID ANALYSIS'.
- OID:** An empty text field.
- Owner:** A dropdown menu showing 'postgres' as the selected user.
- Comment:** A large text area for entering a comment.

At the bottom of the dialog, there are three buttons: 'Close', 'Reset', and 'Save'.

# Database Setup and Table Creation

To create a table:

Query	Query History
1	<b>CREATE TABLE</b> corona_dataset
2	(
3	Province VARCHAR(50),
4	Country_Region VARCHAR(50),
5	Latitude <b>FLOAT</b> ,
6	Longitude <b>FLOAT</b> ,
7	Date DATE,
8	Confirmed INT,
9	Deaths INT,
10	Recovered INT
11	)

# Import Data into the Table

The image illustrates the steps to import data into a table. On the left, a database schema tree shows the 'corona\_dataset' table selected under the 'public' schema. A right-click context menu is open, and the 'Import/Export Data...' option is highlighted. An arrow points to the right, where a dialog box titled 'Import/Export data - table \'corona\_dataset\'' is shown. The 'General' tab is active, and the 'Import' button is selected. The 'Filename' field contains the path 'C:\Users\JENISH\Desktop\Mentoreness\Task 2\Corona Virus Dataset.csv'. The 'Format' is set to 'CSV' and the 'Encoding' is set to 'UTF8'. At the bottom of the dialog are 'Close', 'Reset', and 'OK' buttons.

Right Click the table we just created and choose  
“Import/Export Data – Table”

Navigate the File Path in the FileName section  
which leads to the desired file



# Display the Imported Data

Query

Query History

Scratch Pad ✕

1

2

3

SELECT

\*

FROM

corona\_dataset;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	<div>province</div> <div>character varying (50) 🔒</div>	<div>country_region</div> <div>character varying (50) 🔒</div>	<div>latitude</div> <div>double precision 🔒</div>	<div>longitude</div> <div>double precision 🔒</div>	<div>date</div> <div>date 🔒</div>	<div>confirmed</div> <div>integer 🔒</div>	<div>deaths</div> <div>integer 🔒</div>	<div>recovered</div> <div>integer 🔒</div>	
1	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0	
2	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-23	0	0	0	
3	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-24	0	0	0	
4	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-25	0	0	0	
5	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-26	0	0	0	
6	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-27	0	0	0	
7	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-28	0	0	0	
8	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-29	0	0	0	
9	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-30	0	0	0	
10	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-31	0	0	0	
11	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-01	0	0	0	
12	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-02	0	0	0	
13	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-03	0	0	0	
14	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-04	0	0	0	
15	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-05	0	0	0	
16	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-06	0	0	0	
17	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-07	0	0	0	
18	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-08	0	0	0	
19	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-09	0	0	0	
20	Afghanistan	Afghanistan	33.93911	67.709953	2020-02-10	0	0	0	

Total rows: 1000 of 78386

Query complete 00:00:00.155



# Data Cleaning

To avoid any errors going forward, let's check if there are any missing value / null value.

Q1. Write a code to check NULL values.

```
2  -- Q1. Write a code to check NULL values
3
4  SELECT * FROM corona_dataset
5  WHERE Province IS NULL
6         or Country_Region IS NULL
7         or Latitude IS NULL
8         or Longitude IS NULL
9         or Date IS NULL
10        or Confirmed IS NULL
11        or Deaths IS NULL
12        or Recovered IS NULL;
```

SQL Query

Data Output Messages Notifications

province	country_region	latitude	longitude	date	confirmed	deaths	recovered
character varying (50)	character varying (50)	double precision	double precision	date	integer	integer	integer

Output

**Inference:** Based on the output, it could be rest assured that there are **no null values** present in any columns within the dataset.

**Q2. If NULL values are present, update them with zeros for all columns.**


As we have seen already, there were no null values present in the dataset, but if the values were null, we could have addressed them using the below query.

```
UPDATE corona_dataset
SET
    Province = COALESCE(Province, 'NOT AVAILABLE'),
    Country_Region = COALESCE(Country_Region, 'NOT AVAILABLE'),
    Latitude = COALESCE(Latitude, 0.0),
    Longitude = COALESCE(Longitude, 0.0),
    Date = COALESCE(Date, CURRENT_DATE),
    Confirmed = COALESCE(Confirmed, 0),
    Deaths = COALESCE(Deaths, 0),
    Recovered = COALESCE(Recovered, 0);
```

Q3. Check total number of rows.

```
40 -- Q3. check total number of rows
41
42 SELECT count(*) as total_rows from corona_dataset
43
```

Data Output   Messages   Notifications

	total_rows bigint 
1	78386

**Inference:** The total number of records stored inside our dataset is **78,386**.

#### Q4. Check what is start\_date and end\_date

```
44 -- Q4. Check what is start_date and end_date
45
46 SELECT MIN(Date) AS start_date, MAX(Date) AS end_date
47 FROM corona_dataset;
48
```

Data Output Messages Notifications

	start_date date	end_date date
1	2020-01-22	2021-06-13

**Inference:** As per this data, it is evident that the records the dataset was able to collect start from January 22, 2020 (22-01-2020) and the end date noted was June 13, 2021 (13-06-2021).

## Q5. Number of month present in dataset.

```
49 -- Q5. Number of month present in dataset
50
51 SELECT EXTRACT(MONTH FROM date) AS month_order, COUNT(*) as month_occurance
52 FROM corona_dataset
53 GROUP BY month_order
54 ORDER BY month_order;
55
```

Data Output   Messages   Notifications

	month_order numeric	month_occurance bigint
1	1	6314
2	2	8778
3	3	9548
4	4	9240
5	5	9548
6	6	6622
7	7	4774
8	8	4774
9	9	4620
10	10	4774
11	11	4620
12	12	4774

- In the output, “**month\_order**” represents distinct months from JAN to DEC in numeric form, whereas the column “**month\_occurance**” is associated with number of cases or reports witnessed in that respective month.
- To cite an example, considering January as month\_order = 1, it has month\_occurance value as 6314 depicting there were 6314 occurrence of COVID-19 reported across various countries, during the month of January during the year 2020-2021.

**Inference:** The dataset includes **12** distinct months

## Q6. Find monthly average for confirmed, deaths, recovered.

-- Q6. Find monthly average for confirmed, deaths, recovered

```
SELECT
    EXTRACT(MONTH FROM Date) as month_no,
    EXTRACT(YEAR FROM Date) as Year,
    ROUND(AVG(Confirmed),2) AS avg_confirmed,
    ROUND(AVG(Deaths),2) AS avg_deaths,
    ROUND(AVG(Recovered),2) AS avg_recovered
FROM
    corona_dataset
GROUP BY
    Year, month_no
ORDER BY
    Year, month_no;
```



	month_no numeric	year numeric	avg_confirmed numeric	avg_deaths numeric	avg_recovered numeric
1	1	2020	4.15	0.12	0.09
2	2	2020	15.30	0.59	7.03
3	3	2020	161.13	8.66	27.87
4	4	2020	505.80	41.52	171.64
5	5	2020	574.85	30.28	318.30
6	6	2020	859.23	29.82	548.79
7	7	2020	1432.36	35.11	983.06
8	8	2020	1611.84	37.54	1299.29
9	9	2020	1784.59	34.78	1438.91
10	10	2020	2412.20	36.76	1420.64
11	11	2020	3592.19	56.76	1985.34
12	12	2020	4050.44	71.22	2497.89
13	1	2021	3911.23	84.18	1919.64
14	2	2021	2433.36	69.16	1558.39
15	3	2021	2916.80	59.20	1652.29
16	4	2021	4699.36	78.44	3074.79
17	5	2021	4005.25	76.78	4007.51
18	6	2021	2508.63	66.26	2769.45

**Inference:** Based on the output provided, it is apparent that the highest average values for confirmed cases, deaths, and recovered cases are as follows:

- Confirmed cases: **4699.36** in April 2021
- Deaths: **84.18** in January 2021
- Recovered cases: **4007.51** in May 2021

## Q7. Find most frequent value for confirmed, deaths, recovered each month.

-- Q7. Find most frequent value for confirmed, deaths, recovered each month

```
WITH FrequentData AS (
    SELECT
        EXTRACT(MONTH FROM Date) as month_no,
        EXTRACT(YEAR FROM Date) as year,
        Confirmed,
        Deaths,
        Recovered,
        RANK() OVER (PARTITION BY EXTRACT(MONTH FROM Date), EXTRACT(YEAR FROM Date) ORDER BY COUNT(*) DESC) as rank
    FROM
        corona_dataset
    GROUP BY
        EXTRACT(MONTH FROM Date), EXTRACT(YEAR FROM Date), Confirmed, Deaths, Recovered
)
SELECT
    month_no,
    year,
    Confirmed,
    Deaths,
    Recovered
FROM
    FrequentData
WHERE
    rank = 1
ORDER BY
    year, month_no;
```

	month_no numeric	year numeric	confirmed integer	deaths integer	recovered integer
1	1	2020	0	0	0
2	2	2020	0	0	0
3	3	2020	0	0	0
4	4	2020	0	0	0
5	5	2020	0	0	0
6	6	2020	0	0	0
7	7	2020	0	0	0
8	8	2020	0	0	0
9	9	2020	0	0	0
10	10	2020	0	0	0
11	11	2020	0	0	0
12	12	2020	0	0	0
13	1	2021	0	0	0
14	2	2021	0	0	0
15	3	2021	0	0	0
16	4	2021	0	0	0
17	5	2021	0	0	0
18	6	2021	0	0	0



Q8. Find minimum values for confirmed, deaths, recovered per year.

```
103
104 -- Q8. Find minimum values for confirmed, deaths, recovered per year
105
106 SELECT
107     EXTRACT (Year from Date) as year,
108     MIN(Confirmed) as min_confirmed_cases,
109     MIN(Deaths) as min_death_reported,
110     MIN(Recovered) as min_recover_data
111 FROM
112     corona_dataset
113 GROUP BY
114     year
115 ORDER BY
116     year;
117
```

Data Output Messages Notifications



	year numeric	min_confirmed_cases integer	min_death_reported integer	min_recover_data integer
1	2020	0	0	0
2	2021	0	0	0

Q9. Find maximum values of confirmed, deaths, recovered cases per year.

```
118 -- Q9. Find maximum values of confirmed, deaths, recovered per year.
119
120 SELECT
121     EXTRACT (Year from Date) as year,
122     MAX(Confirmed) as max_confirmed_cases,
123     MAX(Deaths) as max_death_reported,
124     MAX(Recovered) as max_recover_data
125 FROM
126     corona_dataset
127 GROUP BY
128     year
129 ORDER BY
130     year;
131
```

Data Output Messages Notifications

	year numeric	max_confirmed_cases integer	max_death_reported integer	max_recover_data integer
1	2020	823225	3752	1123456
2	2021	414188	7374	422436

### Inference:

- The year 2020 accounts for the most confirmed cases with the figures of 823,225.
- In contrary, it is surprising that despite confirmed cases being half in 2021, 2021 accounts for the most deaths which were 7374 reported during this span.
- However, the maximum recovered were witnessed in 2020, with the figures of 1,123,456 accounting for their recovery.

## Q10. The total number of case of confirmed, deaths, recovered each month.

```
-- Q10. The total number of case of confirmed, deaths, recovered each month

SELECT
    EXTRACT (MONTH from Date) as month_no,
    EXTRACT (YEAR from Date) as year,
    SUM(Confirmed) as total_confirmed_cases,
    SUM(Deaths) as total_death_reported,
    SUM(Recovered) as total_recover_data
FROM
    corona_dataset
GROUP BY
    year, month_no
ORDER BY
    year, month_no;
```

### Inference:

- With a cumulative count of 21,711,021 cases, the number of confirmed cases peaked in April 2021.
- In contrast, January 2021 saw the most number of deaths—recorded as 401,893.
- Additionally, May 2021 reported the highest number of recovered cases, which came to 19,131,842.

	month_no numeric	year numeric	total_confirmed_cases bigint	total_death_reported bigint	total_recover_data bigint
1	1	2020	6384	190	143
2	2	2020	68312	2651	31405
3	3	2020	769236	41346	133070
4	4	2020	2336798	191833	792987
5	5	2020	2744333	144561	1519547
6	6	2020	3969634	137757	2535417
7	7	2020	6838092	167613	4693120
8	8	2020	7694938	179200	6202833
9	9	2020	8244794	160671	6647749
10	10	2020	11515841	175484	6782150
11	11	2020	16595938	262247	9172292
12	12	2020	19336799	339996	11924903
13	1	2021	18672205	401893	9164347
14	2	2021	10492664	298239	6719785
15	3	2021	13924790	282620	7888013
16	4	2021	21711021	362387	14205507
17	5	2021	19121083	366549	19131842
18	6	2021	5022282	132657	5544438

**Q11. Check how corona virus spread out with respect to confirmed cases per month.  
(Eg.: total confirmed cases, their average, variance & STDEV )**

```
-- Q11. Check how corona virus spread out with respect to confirmed case per month
--      (Eg.: total confirmed cases, their average, variance & STDEV )
```

```
SELECT
    EXTRACT(MONTH from Date) as month_no,
    EXTRACT(YEAR from Date) as year,
    SUM(Confirmed) as total_confirmed_cases,
    ROUND(AVG(Confirmed),2) as avg_confirmed_cases,
    ROUND(VARIANCE(Confirmed),2) as var_confirmed_cases,
    ROUND(STDDEV(Confirmed),2) as std_confirmed_cases
FROM
    corona_dataset
GROUP BY
    year, month_no
Order by
    year, month_no;
```

	month_no numeric	year numeric	total_confirmed_cases bigint	avg_confirmed_cases numeric	var_confirmed_cases numeric	std_confirmed_cases numeric
1	1	2020	6384	4.15	4836.05	69.54
2	2	2020	68312	15.30	78507.03	280.19
3	3	2020	769236	161.13	1026629.22	1013.23
4	4	2020	2336798	505.80	7013581.36	2648.32
5	5	2020	2744333	574.85	6064850.73	2462.69
6	6	2020	3969634	859.23	13782194.73	3712.44
7	7	2020	6838092	1432.36	46923851.93	6850.10
8	8	2020	7694938	1611.84	54419982.40	7376.99
9	9	2020	8244794	1784.59	69329705.03	8326.45
10	10	2020	11515841	2412.20	69002612.88	8306.78
11	11	2020	16595938	3592.19	195858271.38	13994.94
12	12	2020	19336799	4050.44	459981798.11	21447.19
13	1	2021	18672205	3911.23	316370963.72	17786.82
14	2	2021	10492664	2433.36	79606383.04	8922.24
15	3	2021	13924790	2916.80	83742806.92	9151.11
16	4	2021	21711021	4699.36	501121674.28	22385.75
17	5	2021	19121083	4005.25	628779318.45	25075.47
18	6	2021	5022282	2508.63	110988215.34	10535.09

**Q12. Check how corona virus spread out with respect to death cases per month.**  
**(Eg.: total death cases, their average, variance & STDEV )**

```

183 -- Q12. Check how corona virus spread out with respect to death case per month
184 --      (Eg.: total death cases, their average, variance & STDEV )
185
186 SELECT
187     EXTRACT(MONTH from Date) as month_no,
188     EXTRACT(YEAR from Date) as year,
189     SUM(Deaths) as total_death_reported,
190     ROUND(AVG(Deaths),2) as avg_death_reported,
191     ROUND(VARIANCE(Deaths),2) as var_death_reported,
192     ROUND(STDDEV(Deaths),2) as std_death_reported
193 FROM
194     corona_dataset
195 GROUP BY
196     year, month_no
197 Order by
198     year, month_no;

```

	month_no numeric	year numeric	total_death_reported bigint	avg_death_reported numeric	var_death_reported numeric	std_death_reported numeric
1	1	2020	190	0.12	4.25	2.06
2	2	2020	2651	0.59	68.34	8.27
3	3	2020	41346	8.66	3901.61	62.46
4	4	2020	191833	41.52	40513.04	201.28
5	5	2020	144561	30.28	20689.25	143.84
6	6	2020	137757	29.82	16933.11	130.13
7	7	2020	167613	35.11	21144.58	145.41
8	8	2020	179200	37.54	23277.87	152.57
9	9	2020	160671	34.78	20107.12	141.80
10	10	2020	175484	36.76	17583.75	132.60
11	11	2020	262247	56.76	27779.81	166.67
12	12	2020	339996	71.22	65359.06	255.65
13	1	2021	401893	84.18	102779.96	320.59
14	2	2021	298239	69.16	68494.76	261.72
15	3	2021	282620	59.20	54397.36	233.23
16	4	2021	362387	78.44	94631.95	307.62
17	5	2021	366549	76.78	131797.08	363.04
18	6	2021	132657	66.26	113020.13	336.18



**Q13. Check how corona virus spread out with respect to recovered cases per month.**  
**(Eg.: total recovered cases, their average, variance & STDEV )**

```

001 -- Q13. Check how corona virus spread out with respect to recovered case per month
002 --      (Eg.: total recovered cases, their average, variance & STDEV )
003
004 SELECT
005     EXTRACT(MONTH from Date) as month_no,
006     EXTRACT(YEAR from Date) as year,
007     SUM(Recovered) as total_recover_data,
008     ROUND(AVG(Recovered),2) as avg_recover_data,
009     ROUND(VARIANCE(Recovered),2) as var_recover_data,
010     ROUND(STDDEV(Recovered),2) as std_recover_data
011 FROM
012     corona_dataset
013 GROUP BY
014     year, month_no
015 Order by
016     year, month_no;

```

	month_no numeric	year numeric	total_recover_data bigint	avg_recover_data numeric	var_recover_data numeric	std_recover_data numeric
1	1	2020	143	0.09	2.64	1.62
2	2	2020	31405	7.03	12449.45	111.58
3	3	2020	133070	27.87	40121.59	200.30
4	4	2020	792987	171.64	770059.71	877.53
5	5	2020	1519547	318.30	1978620.88	1406.63
6	6	2020	2535417	548.79	6531586.26	2555.70
7	7	2020	4693120	983.06	24849082.94	4984.89
8	8	2020	6202833	1299.29	40178838.38	6338.68
9	9	2020	6647749	1438.91	57035911.88	7552.21
10	10	2020	6782150	1420.64	73747150.17	8587.62
11	11	2020	9172292	1985.34	50738601.25	7123.10
12	12	2020	11924903	2497.89	326763170.52	18076.59
13	1	2021	9164347	1919.64	31500298.42	5612.51
14	2	2021	6719785	1558.39	24433077.90	4942.98
15	3	2021	7888013	1652.29	34904703.06	5908.02
16	4	2021	14205507	3074.79	224468171.33	14982.26
17	5	2021	19131842	4007.51	755333749.97	27483.34
18	6	2021	5544438	2769.45	233150866.36	15269.28

Q14. Find Country having highest number of the Confirmed case.

```
218
219 -- Q14. Find Country having highest number of the Confirmed case.
220
221 SELECT
222     country_region as Country,
223     SUM(Confirmed) as total_confirmed_cases
224 FROM corona_dataset
225 GROUP BY Country
226 ORDER BY total_confirmed_cases DESC
227 LIMIT 1;
228
```

Data Output   Messages   Notifications

	country character varying (50) 🔒	total_confirmed_cases bigint 🔒
1	US	33461982

**Inference:** US had the highest number of confirmed cases recorded which was cumulated to 33,461,982.



**Q15. Find Country having lowest number of the death cases.**

```
!47 -- Q15. Find Country having lowest number of the death cases.
!48
!49 WITH rankingCountry as (
!50     SELECT
!51         country_region as Country,
!52         SUM(Deaths) as total_death_reported,
!53         RANK() Over(ORDER by SUM(Deaths) ASC) as rank_no
!54     FROM
!55         corona_dataset
!56     GROUP BY
!57         Country
!58 )
!59 SELECT
!60     Country,
!61     total_death_reported
!62 FROM
!63     rankingCountry
!64 WHERE
!65     rank_no = 1;
!66
```

	country character varying (50) 🔒	total_death_reported bigint 🔒
1	Samoa	0
2	Kiribati	0
3	Dominica	0
4	Marshall Islands	0

**Inference:** Samoa, Kiribati, Dominica, and Marshall Islands had reported the lowest number of death cases, which each country had 0 casualties as per the datasets.

**Q16. Find top 5 countries having highest recovered cases.**

```
267 -- Q16. Find top 5 countries having highest recovered cases.
268
269 SELECT
270     country_region AS Country,
271     SUM(Recovered) AS total_recovered_data
272 FROM
273     corona_dataset
274 GROUP BY
275     Country
276 ORDER BY total_recovered_data DESC
277 limit 5;
278
```

	country character varying (50) 🔒	total_recovered_data bigint 🔒
1	India	28089649
2	Brazil	15400169
3	US	6303715
4	Turkey	5202251
5	Russia	4745756

**Inference:** **India** recorded the most recovery as per the dataset with its number summing up to 28,089,649, whereas **Brazil** and **US** are 2<sup>nd</sup> and 3<sup>rd</sup> best, followed by **Turkey** and **Russia** respectively

# Insights

---

Several revelations have been discovered after the SQL analysis of the COVID dataset.

1. According to the dataset COVID-19 Pandemic duration was from January 22, 2020, to June 13, 2021.
2. Though 2020 had the most number of cases recorded, the amount of casualties recorded was more in 2021.
3. India had the highest number of recovered cases.
4. The confirmed cases skyrocketed in April 2021.
5. Samoa, Kiribati, Dominica, and Marshall Islands had reported the lowest number of death cases.
6. Though the US was the most recorded country with confirmed cases, it comes 3rd in the case of recovery data which was accumulated in the dataset.

These datasets provide valuable insights for understanding the impact COVID pandemic caused based on the dataset.

**Thank You !!!**