

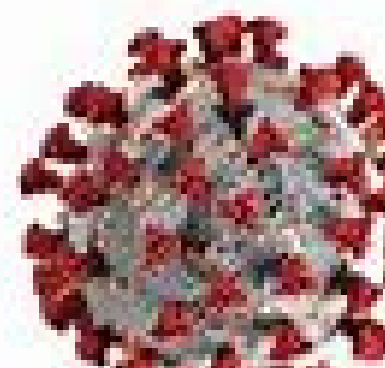
SQL PROJECT

Corona Virus Analysis

CREATED BY: MOBOLAJI
AADEGBAMIGBE

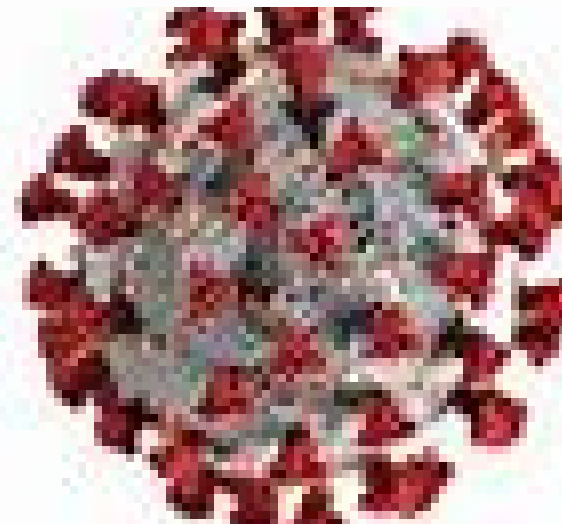
OBJECTIVE

The CORONA VIRUS pandemic has had a significant impact on public health and has created an urgent need for data-driven insights to understand the spread of the virus. The task is to derive insights using SQL from the dataset containing information such as geographic location, dates, confirmed cases, deaths, and recoveries.



Description of each column in the dataset:

- 01. PROVINCE: GEOGRAPHIC SUBDIVISION WITHIN A COUNTRY/REGION.
- 02. Country/Region: Geographic entity where data is recorded.
- 03. Latitude: North-south position on Earth's surface.
- 04. Longitude: East-west position on Earth's surface.
- 05. Date: Recorded date of CORONA VIRUS data.
- 06. Confirmed: Number of diagnosed CORONA VIRUS cases.
- 07. DEATHS: NUMBER OF CORONA VIRUS RELATED DEATHS.
- 08. RECOVERED: NUMBER OF RECOVERED CORONA VIRUS CASES.



Question 1:

WRITE A CODE TO CHECK NULL VALUES

```
1 • SELECT *
2 FROM name.`corona virus dataset`
3 WHERE Province IS NULL
4
```

Result Grid

Filter Rows:

Export:



Wrap Cell Content:

Province	Country_Region	Latitude	Longitude	Date	Confirmed	Deaths	Recovered
----------	----------------	----------	-----------	------	-----------	--------	-----------

Question 2:

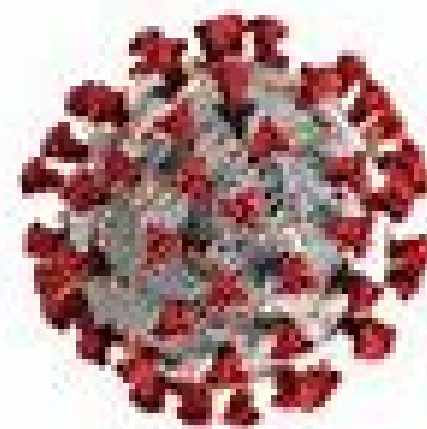
CHECK THE TOTAL NUMBER OF ROWS

```
1 • SELECT COUNT(*)
2 FROM name.`corona virus dataset`
```

Result Grid			Filter Rows: <input type="text"/>
	COUNT(*)		
	78386		

Question 3:

CHECK THE START_DATE AND
END_DATE



```
1 • SELECT *
2 FROM name.`corona virus dataset`
3 WHERE Date >= '2020'
```

Finland	Finland	61.92411	25.748151	2021-06-12	114	0	0
Finland	Finland	61.92411	25.748151	2021-06-13	47	0	0

<

Result Grid

Filter Rows:

Export:

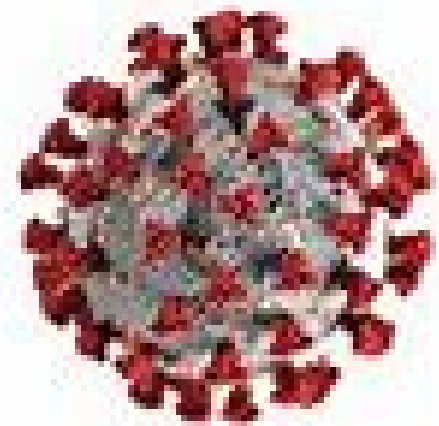
Wrap Cell Content:

Fetch rows:

	Province	Country_Region	Latitude	Longitude	Date	Confirmed	Deaths	Recovered
▶	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0

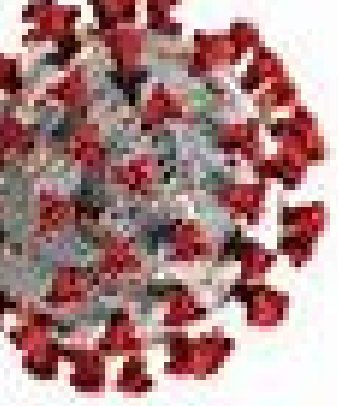
Question 4:

NUMBER OF MONTHS PRESENT IN
THE DATASET



```
1 • SELECT COUNT(DISTINCT MONTH(Date)) AS num_months
2 FROM name.`corona virus dataset`;
3
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content
	num_months			
	12			

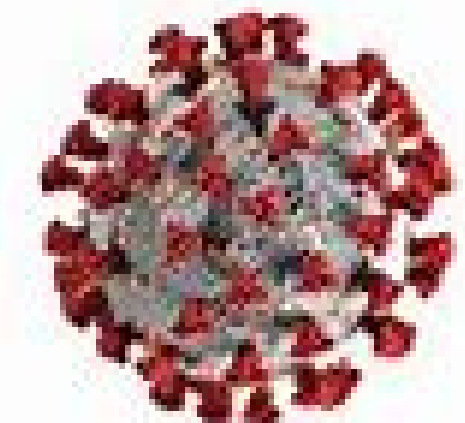


Question 5:

FIND THE MONTHLY AVERAGE FOR
CONFIRMED, DEATHS, RECOVERED

```
1 • SELECT
2     YEAR(Date) AS year,
3     MONTH(Date) AS month,
4     AVG(Confirmed) AS avg_confirmed,
5     AVG(Deaths) AS avg_deaths,
6     AVG(Recovered) AS avg_recovered
7 FROM
8     name.`corona virus dataset`
9 GROUP BY
10     YEAR(Date), MONTH(Date)
11 ORDER BY
```

	year	month	avg_confirmed	avg_deaths	avg_recovered
▶	2020	1	4.1455	0.1234	0.0929
	2020	2	15.2960	0.5936	7.0320
	2020	3	161.1303	8.6607	27.8739
	2020	4	505.8004	41.5223	171.6422
	2020	5	574.8498	30.2809	318.2964
	2020	6	859.2281	29.8175	548.7916
	2020	7	1432.3611	35.1096	983.0582
	2020	8	1611.8429	37.5367	1299.2947
	2020	9	1784.5874	34.7773	1438.9067
	2020	10	2412.1996	36.7583	1420.6431
	2020	11	3592.1944	56.7634	1985.3446
	2020	12	4050.4397	71.2183	2497.8850
	2021	1	3911.2285	84.1837	1919.6370
	2021	2	2433.3636	69.1649	1558.3917
	2021	3	2916.7972	59.1998	1652.2859



Question 6:

FIND THE MOST FREQUENT VALUE
FOR CONFIRMED, DEATHS,
RECOVERED EACH MONTH

```
1 WITH MonthlyStats AS (  
2     SELECT  
3         YEAR(Date) AS year,  
4         MONTH(Date) AS month,  
5         Confirmed,  
6         Deaths,  
7         Recovered,  
8         ROW_NUMBER() OVER (PARTITION BY YEAR(Date), MONTH(Date) ORDER BY COUNT(*) DESC) AS rn  
9     FROM  
10         name.`corona virus dataset`  
11     GROUP BY
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	year	month	Confirmed	Deaths	Recovered
2020	1	0	0	0	0
2020	2	0	0	0	0
2020	3	0	0	0	0
2020	4	0	0	0	0
2020	5	0	0	0	0
2020	6	0	0	0	0
2020	7	0	0	0	0
2020	8	0	0	0	0
2020	9	0	0	0	0
2020	10	0	0	0	0
2020	11	0	0	0	0
2020	12	0	0	0	0
2021	1	0	0	0	0
2021	2	0	0	0	0
2021	3	0	0	0	0

```
11     GROUP BY  
12         YEAR(Date), MONTH(Date), Confirmed, Deaths, Recovered  
13 )  
14     SELECT  
15         year,  
16         month,  
17         Confirmed,  
18         Deaths,  
19         Recovered  
20     FROM  
21         MonthlyStats  
22     WHERE  
23         rn = 1  
24     ORDER BY  
25         year, month;
```

Question 7:

FIND THE MINIMUM VALUES FOR
CONFIRMED, DEATHS, RECOVERED
PER YEAR

```
1 • SELECT
2     YEAR(Date) AS Year,
3     MIN(Confirmed) AS min_confirmed,
4     MIN(Deaths) AS min_deaths,
5     MIN(Recovered) AS min_recovered
6 FROM
7     name.`corona virus dataset`
8 GROUP BY
9     Year(Date)
```

<				
Result Grid				
Filter Rows: <input type="text"/>				
Export:				
	Year	min_confirmed	min_deaths	min_recovered
▶	2020	0	0	0
	2021	0	0	0

Question 8:

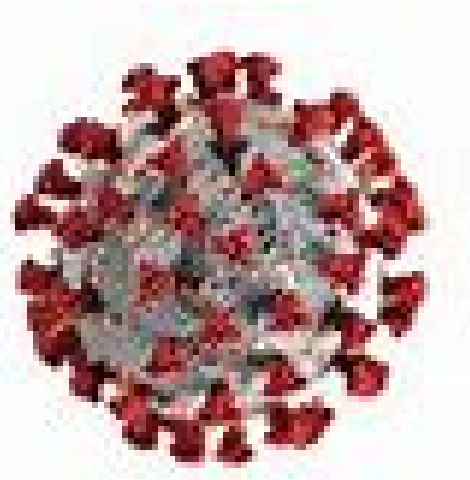
FIND THE MAXIMUM VALUES OF
CONFIRMED, DEATHS, RECOVERED
PER YEAR

```
1 • SELECT
2     YEAR(Date) AS year,
3     MAX(Confirmed) AS max_confirmed,
4     MAX(Deaths) AS max_deaths,
5     MAX(Recovered) AS max_recovered
6 FROM
7     name.`corona virus dataset`
8 GROUP BY
9     YEAR(Date)
10
11
```

Result Grid					Filter Rows:	Export:
	year	max_confirmed	max_deaths	max_recovered		
▶	2020	823225	3752	1123456		
	2021	414188	7374	422436		

Question 9:

THE TOTAL NUMBER OF CASES OF
CONFIRMED, DEATHS, RECOVERED
EACH MONTH



```
1 • SELECT
2     YEAR(Date) AS year,
3     MONTH(Date) AS month,
4     SUM(Confirmed) AS total_confirmed,
5     SUM(Deaths) AS total_deaths,
6     SUM(Recovered) AS total_recovered
7 FROM
8     name.`corona virus dataset`
9 GROUP BY
10    YEAR(Date), MONTH(Date);
11
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	year	month	total_confirmed	total_deaths	total_recovered
	2020	4	2336798	191833	792987
	2020	5	2744333	144561	1519547
	2020	6	3969634	137757	2535417
	2020	7	6838092	167613	4693120
	2020	8	7694938	179200	6202833
	2020	9	8244794	160671	6647749
	2020	10	11515841	175484	6782150
	2020	11	16595938	262247	9172292
	2020	12	19336799	339996	11924903
	2021	1	18672205	401893	9164347
	2021	2	10492664	298239	6719785
	2021	3	13924790	282620	7888013
	2021	4	21711021	362387	14205507

Question 10:

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO
CONFIRMED CASES (EG.: TOTAL CONFIRMED CASES, THEIR
AVERAGE, VARIANCE & STDEV)

I.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO TOTAL
CONFIRMED CASES

```
1 • SELECT
2     SUM(Confirmed) AS total_confirmed_case
3 FROM
4     name.`corona virus dataset`;
5
```

total_confirmed_cases
169065144

II.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO
AVERAGE
CONFIRMED CASES

```
1 • SELECT
2     AVG(Confirmed) AS average_confirmed_cases
3 FROM
4     name.`corona virus dataset`;
5
6
```

average_confirmed_cases
2156.8283

III.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO THE
VARIANCE OF
CONFIRMED CASES

```
1 • SELECT
2     VARIANCE(Confirmed) AS variance_confirmed_cases
3 FROM
4     name.`corona virus dataset`;
5
```

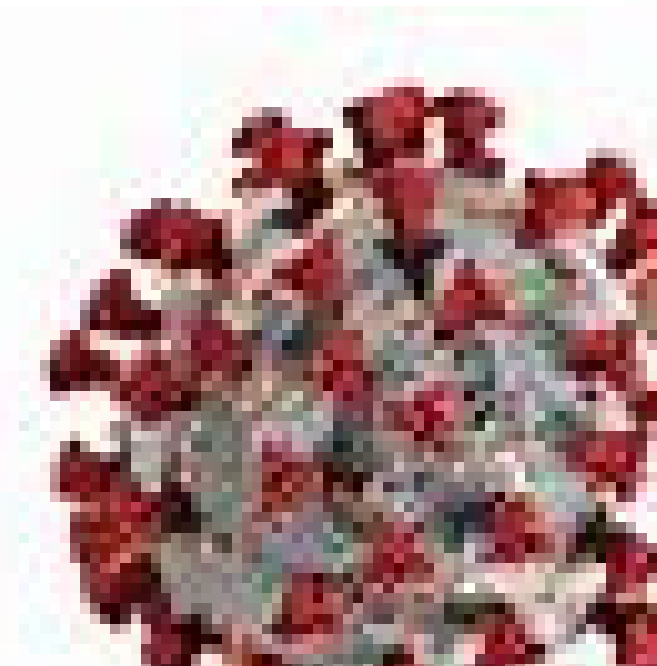
variance_confirmed_cases
157288925.07796532

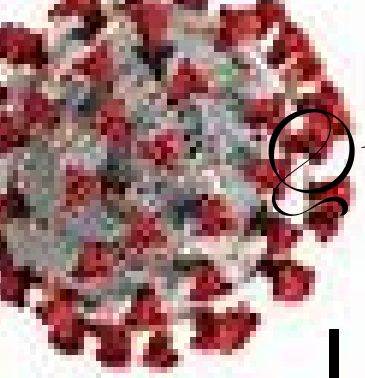
IV.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO THE
STDEV CONFIRMED
CASES

```
1 • SELECT
2     STDDEV(Confirmed) AS std_dev_confirmed_cases
3 FROM
4     name.`corona virus dataset`;
5
```

std_dev_confirmed_cases
12541.488152446875





Question 11:

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO DEATH CASES PER MONTH
-- (EG.: TOTAL CONFIRMED CASES, THEIR AVERAGE, VARIANCE & STDEV)

I.)

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO TOTAL DEATH CASES

II.)

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO AVERAGE DEATH CASES

III.)

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO THE VARIANCE OF DEATH CASES

IV.)

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO THE STDEV DEATH CASES

```
1 • SELECT
2     SUM(Deaths) AS total_death_cases
3 FROM
4     name.`corona virus dataset`
```

Result Grid	
total_death_cases	
▶	3647894

```
1 • SELECT
2     AVG(Deaths) AS average_death_cases
3 FROM
4     name.`corona virus dataset`
```

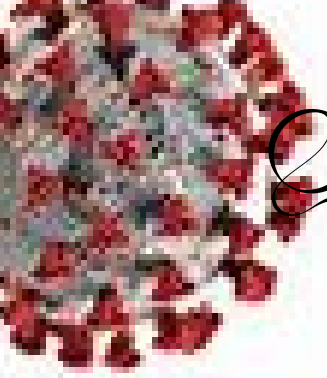
Result Grid	
average_death_cases	
▶	46.5376

```
1 • SELECT
2     VARIANCE(Deaths) AS variance_death_cases
3 FROM
4     name.`corona virus dataset`;
```

Result Grid	
variance_death_cases	
▶	45892.01885355753

```
1 • SELECT
2     STDDEV(Deaths) AS std_dev_death_cases
3 FROM
4     name.`corona virus dataset`;
5
```

Result Grid	
std_dev_death_cases	
▶	214.22422564583476



Question 12:

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO
RECOVERED CASES
-- (EG.: TOTAL CONFIRMED CASES, THEIR AVERAGE,
VARIANCE & STDEV)

I.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO TOTAL
RECOVERED CASES

II.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO
AVERAGE
RECOVERED CASES

III.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO THE
VARIANCE OF
RECOVERED CASES

IV.)

CHECK HOW
CORONA VIRUS
SPREAD OUT WITH
RESPECT TO THE
STDEV RECOVERED
CASES

```
1 • SELECT
2     SUM(Recovered) AS total_recovered_cases
3 FROM
4     name.`corona virus dataset`
```

Result Grid | Filter Rows: | Export: | Wrap Cell

total_recovered_cases
113089548

```
1 • SELECT
2     AVG(Recovered) AS average_recovered_cases
3 FROM
4     name.`corona virus dataset`
```

Result Grid | Filter Rows: | Export: | Wrap Cell

average_recovered_cases
1442.7264

```
1 • SELECT
2     VARIANCE(Recovered) AS variance_recovered_cases
3 FROM
4     name.`corona virus dataset`;
```

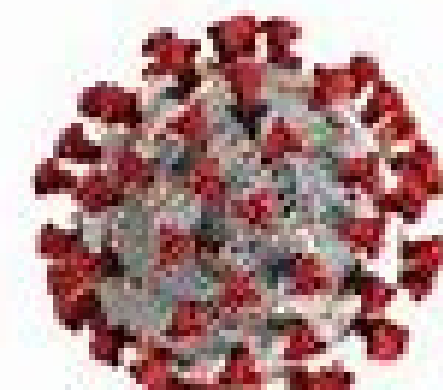
Result Grid | Filter Rows: | Export: | Wrap Cell

variance_recovered_cases
107029523.26229636

```
1 • SELECT
2     STDDEV(Recovered) AS std_dev_recovered_cases
3 FROM
4     name.`corona virus dataset`
```

Result Grid | Filter Rows: | Export: | Wrap Cell




std_dev_recovered_cases

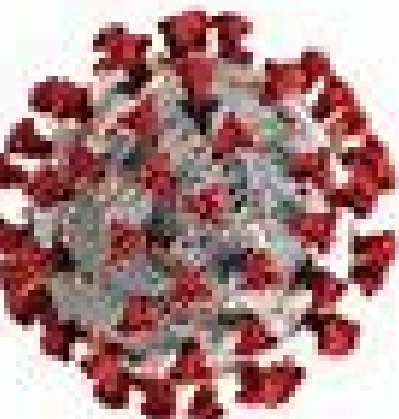


Question 13:

FIND THE COUNTRY THAT HAS THE
HIGHEST NUMBER OF CONFIRMED
CASES.

```
1 • SELECT
2     Country_Region,
3     Confirmed
4 FROM
5     name.`corona virus dataset`
6 WHERE
7     Confirmed = (
8         SELECT
9             MAX(Confirmed)
10        FROM
11        name.`corona virus dataset`
```

<		
Result Grid		 Filter Rows: <input type="text"/>
Export: 		Wrap
	Country_Region	Confirmed
▶	Turkey	823225





FIND THE COUNTRY THAT HAS THE
LOWEST NUMBER OF DEATH CASES.

[illegible]

Question 15:

FIND THE TOP 5 COUNTRIES THAT
HAS THE HIGHEST RECOVERED
CASES.

```
1 • SELECT
2     Country_Region,
3     MAX(Recovered) AS Highest_Recovered_Cases
4 FROM
5     name.`corona virus dataset`
6 GROUP BY
7     Country_Region
8 ORDER BY
9     MAX(Recovered) DESC
10 LIMIT 5;
```

<		
Result Grid  Filter Rows: <input type="text"/> Export:  Wrap		
	Country_Region	Highest_Recovered_Cases
	Brazil	388340
	Colombia	89557
	India	422436
	Turkey	1123456
▶	US	150267

Thank You!

PLEASE ZOOM IN TO
VIEW SQL QUERIES

