



Corona Virus Analysis with SQL

Analyzing the impact and spread of the virus

Presented By
Ravi Patel

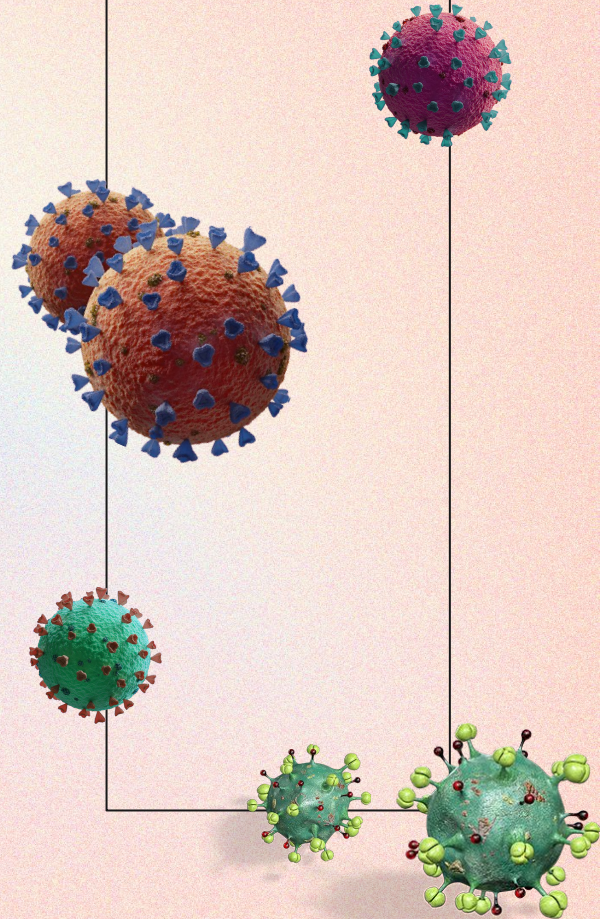


TABLE OF CONTENTS

01

Overview

You can describe the topic of the section here

02

Dataset Description

Details of each column in the dataset.

03

Queries

Some key Queries to get insights

04

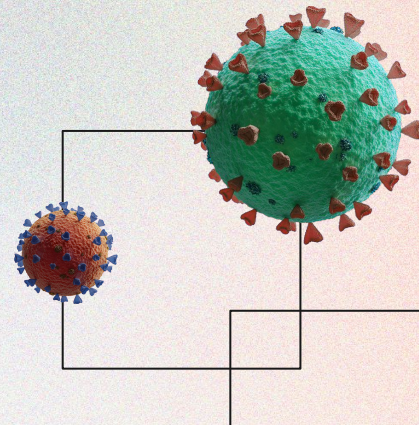
Analysis Summary

Summary of key insights from the data analysis.

05

Conclusion

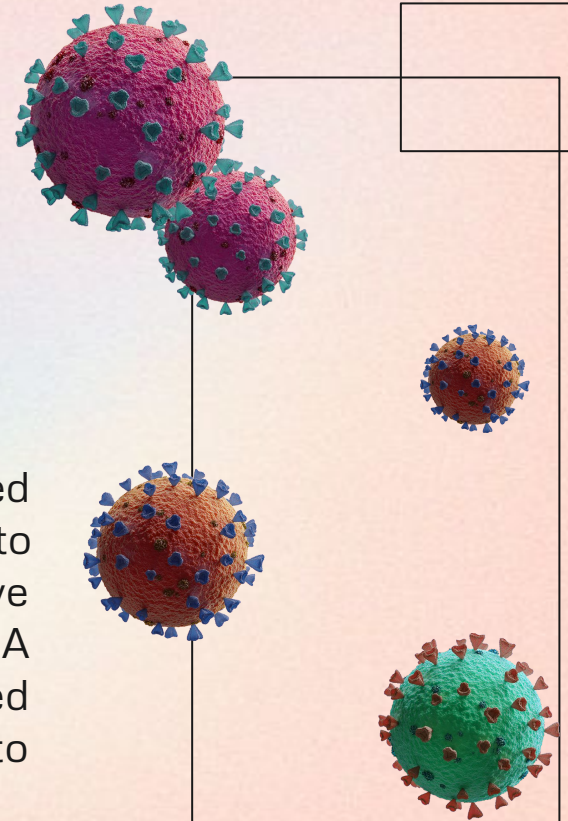
Importance of data-driven decision-making





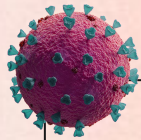
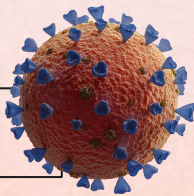
Overview

The CORONA VIRUS pandemic has significantly impacted public health, necessitating data-driven insights to understand the virus's spread. This analysis aims to derive meaningful insights from a comprehensive CORONA VIRUS dataset, focusing on key metrics such as confirmed cases, deaths, and recoveries over time. The goal is to support data-driven decision-making in public health.



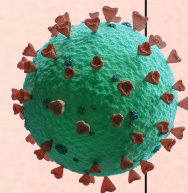
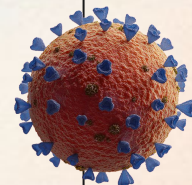
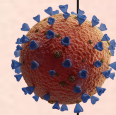
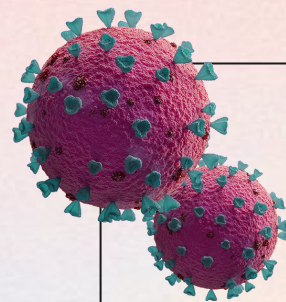
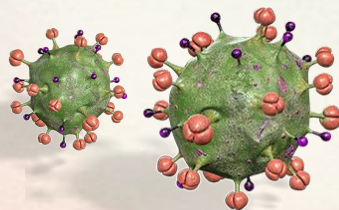
Dataset Description

- **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.





Queries

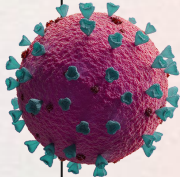






Write a code to check NULL values





SELECT

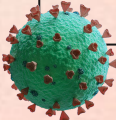
```
COUNT (*) - COUNT (province) AS Province,  
COUNT (*) - COUNT (country_region) AS Country_Region,  
COUNT (*) - COUNT (latitude) AS Latitude,  
COUNT (*) - COUNT (longitude) AS Longitude,  
COUNT (*) - COUNT (date) AS Date,  
COUNT (*) - COUNT (confirmed) AS Confirmed,  
COUNT (*) - COUNT (deaths) AS Deaths,  
COUNT (*) - COUNT (recovered) AS Recovered
```

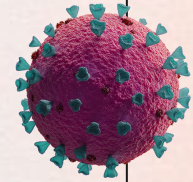
FROM corona_virus_data;



	province bigint 	country_region bigint 	latitude bigint 	longitude bigint 
1	0	0	0	0

date bigint 	confirmed bigint 	deaths bigint 	recovered bigint 
0	0	0	0





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

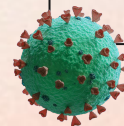
```
UPDATE corona_virus_data  
SET
```

```
province = COALESCE (province, ''),  
country_region = COALESCE (country_region, ''),  
latitude = COALESCE (latitude, 0),  
longitude = COALESCE (longitude, 0),  
date = COALESCE (date, '1970-01-01'),  
confirmed = COALESCE (confirmed, 0),  
deaths = COALESCE (deaths, 0),  
recovered = COALESCE (recovered, 0);
```

```
UPDATE 78386
```


Query returned successfully in 919 msec.

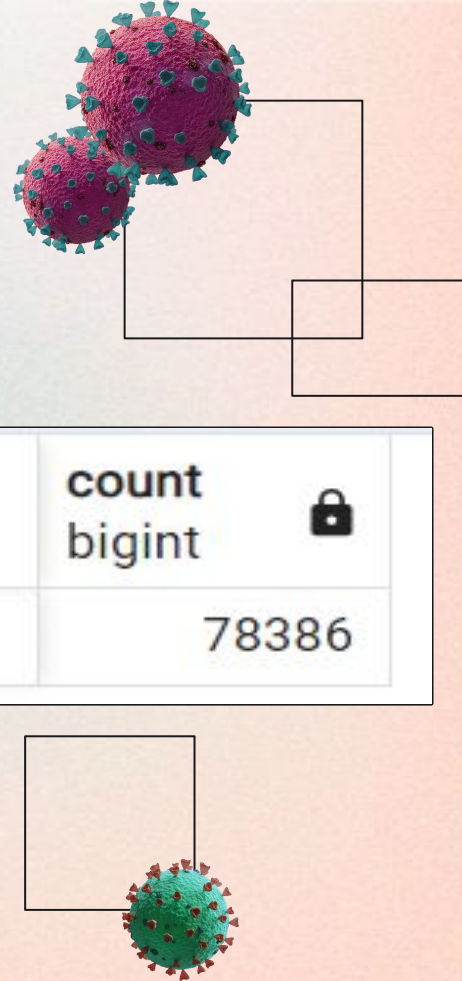
Here in table any place null value present then updated with 0

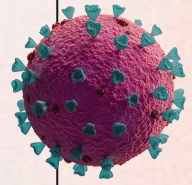


Check total number of rows

```
SELECT COUNT (*) FROM corona_virus_data;
```

	count bigint 
1	78386

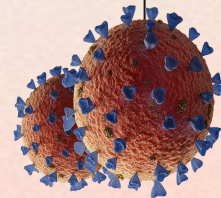


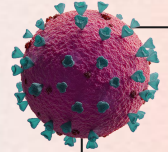


Check what is start_date and end_date

```
SELECT MIN (date) AS start_date, MAX(date) AS end_date  
FROM corona_virus_data;
```


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

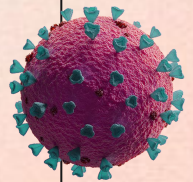




Number of month present in dataset

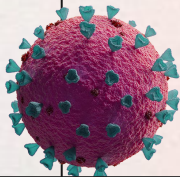
```
SELECT COUNT (DISTINCT TO_CHAR(date, 'YYYY-MM'))  
AS months_count FROM corona_virus_data;
```

	months_count 
1	18

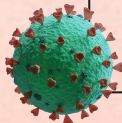


Find monthly average for confirmed, deaths, recovered

```
SELECT
    TO_CHAR(date, 'YYYY-MM') AS month,
    ROUND(CAST(AVG(confirmed)AS numeric), 1) AS
avg_confirmed,
    ROUND(CAST(AVG(deaths)AS numeric), 1) AS
avg_deaths,
    ROUND(CAST(AVG(recovered)AS numeric), 1) AS
avg_recovered
FROM corona_virus_data
GROUP BY month
ORDER BY month;
```



	month text	avg_confirmed numeric	avg_deaths numeric	avg_recovered numeric
1	2020-01	4.1	0.1	0.1
2	2020-02	15.3	0.6	7.0
3	2020-03	161.1	8.7	27.9
4	2020-04	505.8	41.5	171.6
5	2020-05	574.8	30.3	318.3
6	2020-06	859.2	29.8	548.8
7	2020-07	1432.4	35.1	983.1
8	2020-08	1611.8	37.5	1299.3
9	2020-09	1784.6	34.8	1438.9
10	2020-10	2412.2	36.8	1420.6
11	2020-11	3592.2	56.8	1985.3
12	2020-12	4050.4	71.2	2497.9
13	2021-01	3911.2	84.2	1919.6
14	2021-02	2433.4	69.2	1558.4
15	2021-03	2916.8	59.2	1652.3
16	2021-04	4699.4	78.4	3074.8
17	2021-05	4005.3	76.8	4007.5
18	2021-06	2508.6	66.3	2769.4



Find most frequent value for confirmed, deaths, recovered each month

```
WITH freq_values AS (  
  SELECT  
    TO_CHAR(date, 'YYYY-MM') AS month ,  
    confirmed, deaths, recovered,  
    ROW_NUMBER () OVER (PARTITION BY  
TO_CHAR(date, 'YYYY-MM') ORDER BY COUNT (*)  
DESC) AS rn  
  FROM corona_virus_data  
  GROUP BY month, confirmed, deaths, recovered  
)  
SELECT month, confirmed, deaths, recovered  
FROM freq_values  
WHERE rn = 1;
```



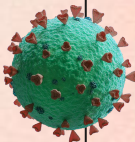
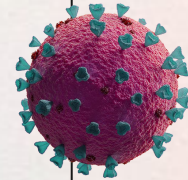
	month text	confirmed integer	deaths integer	recovered integer
1	2020-01	0	0	0
2	2020-02	0	0	0
3	2020-03	0	0	0
4	2020-04	0	0	0
5	2020-05	0	0	0
6	2020-06	0	0	0
7	2020-07	0	0	0
8	2020-08	0	0	0
9	2020-09	0	0	0
10	2020-10	0	0	0
11	2020-11	0	0	0
12	2020-12	0	0	0
13	2021-01	0	0	0
14	2021-02	0	0	0
15	2021-03	0	0	0
16	2021-04	0	0	0
17	2021-05	0	0	0
18	2021-06	0	0	0

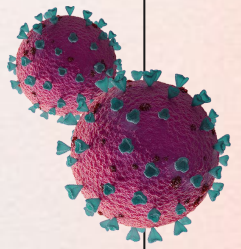


Find minimum values for confirmed, deaths, recovered per year

```
SELECT
  EXTRACT (YEAR FROM date) AS year ,
  MIN (confirmed) AS min_confirmed,
  MIN (deaths) AS min_deaths,
  MIN (recovered) AS min_recovered
FROM corona_virus_data
GROUP BY year
ORDER BY year;
```





	year numeric 🔒	min_confirmed integer 🔒	min_deaths integer 🔒	min_recovered integer 🔒
1	2020	0	0	0
2	2021	0	0	0

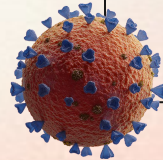




Find maximum values of confirmed, deaths, recovered per year

```
SELECT  
  EXTRACT (YEAR FROM date) AS year,  
  MAX(confirmed) AS max_confirmed,  
  MAX(deaths) AS max_deaths,  
  MAX(recovered) AS max_recovered  
FROM corona_virus_data  
GROUP BY year  
ORDER BY year;
```

	year numeric 	max_confirmed integer 	max_deaths integer 	max_recovered integer 
1	2020	823225	3752	1123456
2	2021	414188	7374	422436



The total number of case of confirmed, deaths, recovered each month

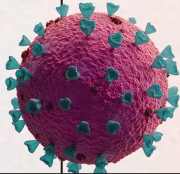
SELECT

TO_CHAR(date, 'YYYY-MM') **AS** month,
SUM(confirmed) **AS** total_confirmed,
SUM(deaths) **AS** total_deaths,
SUM(recovered) **AS** total_recovered

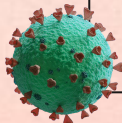
FROM corona_virus_data

GROUP BY month

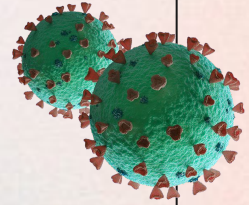
ORDER BY month;



	month text	total_confirmed bigint	total_deaths bigint	total_recovered bigint
1	2020-01	6384	190	143
2	2020-02	68312	2651	31405
3	2020-03	769236	41346	133070
4	2020-04	2336798	191833	792987
5	2020-05	2744333	144561	1519547
6	2020-06	3969634	137757	2535417
7	2020-07	6838092	167613	4693120
8	2020-08	7694938	179200	6202833
9	2020-09	8244794	160671	6647749
10	2020-10	11515841	175484	6782150
11	2020-11	16595938	262247	9172292
12	2020-12	19336799	339996	11924903
13	2021-01	18672205	401893	9164347
14	2021-02	10492664	298239	6719785
15	2021-03	13924790	282620	7888013
16	2021-04	21711021	362387	14205507
17	2021-05	19121083	366549	19131842
18	2021-06	5022282	132657	5544438

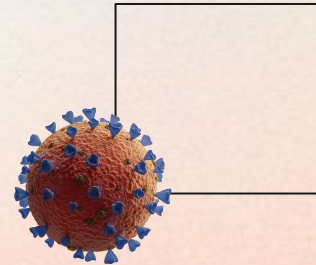


Check how corona virus spread out with respect to confirmed case



```
SELECT
    SUM (confirmed) AS total_confirmed,
    ROUND (CAST (AVG (confirmed) AS numeric), 1) AS
avg_confirmed,
    ROUND (CAST (VARIANCE (confirmed) AS numeric), 1)
AS var_confirmed,
    ROUND (CAST (STDDEV (confirmed) AS numeric), 1) AS
stdev_confirmed
FROM corona_virus_data;
```

	total_confirmed bigint	avg_confirmed numeric	var_confirmed numeric	stdev_confirmed numeric
1	169065144	2156.8	157290931.7	12541.6



Check how corona virus spread out with respect to death case per month

SELECT

TO_CHAR(date, 'YYYY-MM') **AS** month,

SUM(deaths) **AS** total_deaths,

ROUND(**CAST**(**AVG**(deaths)**AS** numeric), 1)**AS**

avg_deaths,

ROUND(**CAST**(**VARIANCE**(deaths)**AS** numeric), 1)**AS**

var_deaths,

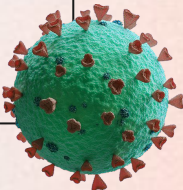
ROUND(**CAST**(**STDDEV**(deaths)**AS** numeric), 1)**AS**

stdev_deaths

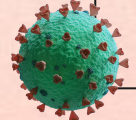
FROM corona_virus_data

GROUP BY month

ORDER BY month;



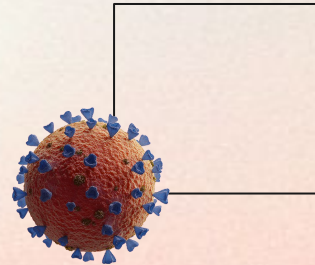
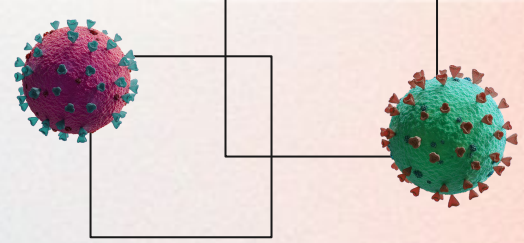
	month text	total_deaths bigint	avg_deaths numeric	var_deaths numeric	stdev_deaths numeric
1	2020-01	190	0.1	4.2	2.1
2	2020-02	2651	0.6	68.3	8.3
3	2020-03	41346	8.7	3901.6	62.5
4	2020-04	191833	41.5	40513.0	201.3
5	2020-05	144561	30.3	20689.2	143.8
6	2020-06	137757	29.8	16933.1	130.1
7	2020-07	167613	35.1	21144.6	145.4
8	2020-08	179200	37.5	23277.9	152.6
9	2020-09	160671	34.8	20107.1	141.8
10	2020-10	175484	36.8	17583.8	132.6
11	2020-11	262247	56.8	27779.8	166.7
12	2020-12	339996	71.2	65359.1	255.7
13	2021-01	401893	84.2	102780.0	320.6
14	2021-02	298239	69.2	68494.8	261.7
15	2021-03	282620	59.2	54397.4	233.2
16	2021-04	362387	78.4	94632.0	307.6
17	2021-05	366549	76.8	131797.1	363.0
18	2021-06	132657	66.3	113020.1	336.2

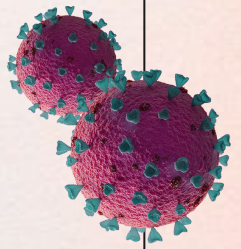


Check how corona virus spread out with respect to recovered case

```
SELECT
    SUM(recovered) AS total_recovered,
    ROUND(CAST(AVG(recovered)AS numeric), 1)AS
avg_recovered,
    ROUND(CAST(VARIANCE(recovered)AS numeric),
1)AS var_recovered,
    ROUND(CAST(STDDEV(recovered)AS numeric), 1)AS
stdev_recovered
FROM corona_virus_data;
```



	total_confirmed bigint	avg_confirmed numeric	var_confirmed numeric	stdev_confirmed numeric
1	169065144	2156.8	157290931.7	12541.6

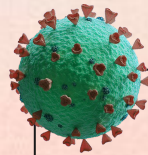




Find Country having highest number of the Confirmed case

```
SELECT
    country_region,
    SUM(confirmed) AS total_confirmed
FROM corona_virus_data
GROUP BY country_region
ORDER BY total_confirmed DESC
LIMIT 1;
```

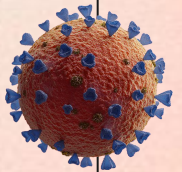
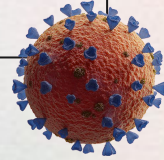
	country_region 	total_confirmed 
	text	bigint
1	US	33461982



Find Country having lowest number of the death case

```
SELECT
    country_region,
    SUM(deaths) AS total_deaths
FROM corona_virus_data
GROUP BY country_region
ORDER BY total_deaths ASC
LIMIT 1;
```

	country_region text	total_deaths bigint
1	Kiribati	0

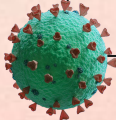


Find top 5 countries having highest recovered case

```
SELECT
    country_region,
    SUM(recovered) AS total_recovered
FROM corona_virus_data
GROUP BY country_region
ORDER BY total_recovered DESC
LIMIT 5;
```

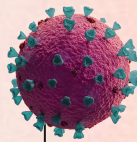
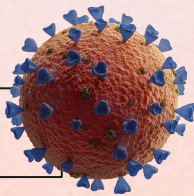


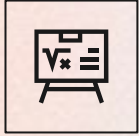
	country_region text	total_recovered bigint
1	India	28089649
2	Brazil	15400169
3	US	6303715
4	Turkey	5202251
5	Russia	4745756



Analysis Summary

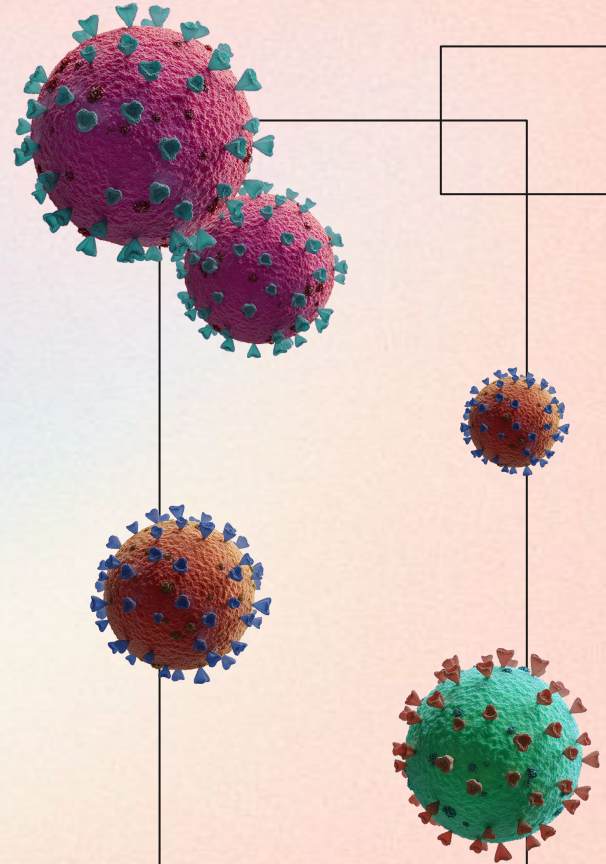
- Handled NULL values and ensured data integrity.
- Dataset contains 78386 records from 2020-01-22 to 2021-06-13.
- Data covers 18 distinct months.
- Calculated monthly averages and most frequent values for confirmed cases, deaths, and recoveries.
- Identified yearly minimum and maximum values, and analyzed the spread of cases.





Conclusion

The dataset provided valuable insights into the spread and impact of the CORONA VIRUS, revealing trends in confirmed cases, deaths, and recoveries over time. This analysis underscores the importance of data-driven decision-making in public health.



THANKS!

Do you have any questions?

Please feel free to send me a message on
LinkedIn.

