CORONA VIRUS ANALYSIS

SQL AND DATA ANALYSIS SKILLS IN REAL-WORLD CONTEXT

MENTORNESS INTERNSHIP PROJECT
BY
BODA JAMPAIAH

CONTENTS

MENTOR NESS

PROJECT OVERVIEW

01

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.

02

As a data analyst, you have been tasked with analyzing a CORONA VIRUS dataset to derive meaningful insights and present your findings.

MENTOR NESS

DATASET

Description of each column in 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.

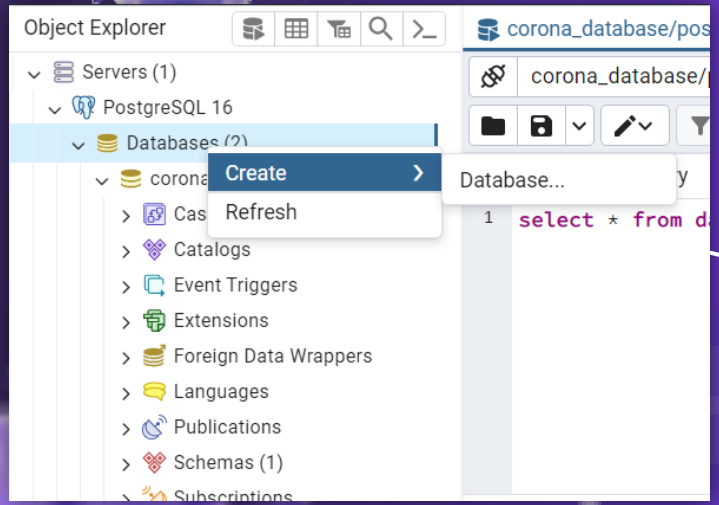
Recovered: Number of recovered CORONA VIRUS cases.

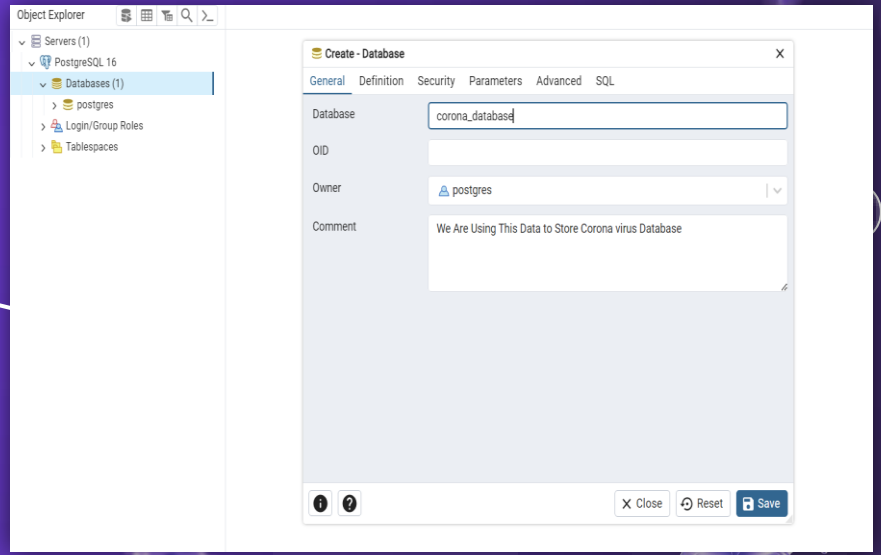
MENTOR NESS

Introduction

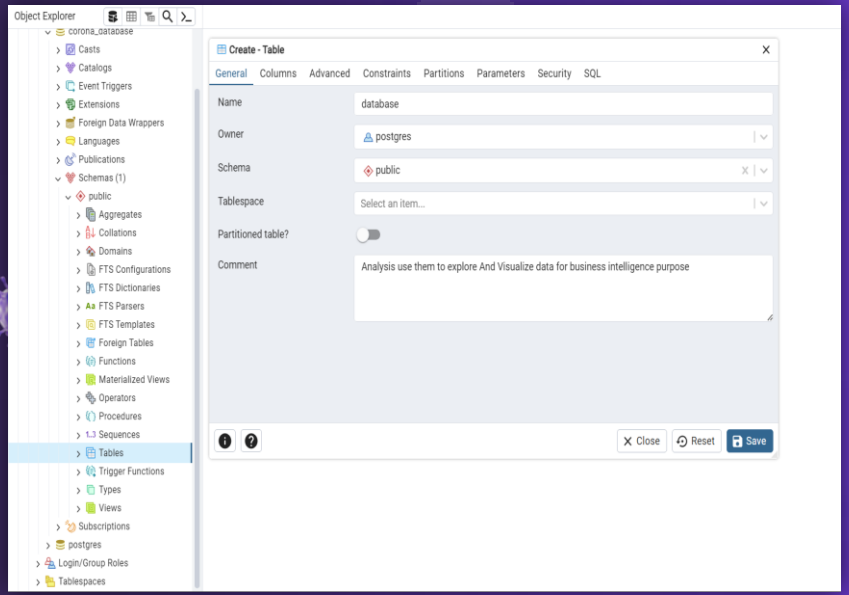
Welcome to an internship project crafted to evaluate your SQL proficiency and data analysis skills within a practical context. Throughout this experience, you're encouraged to approach tasks creatively and seek guidance whenever needed.

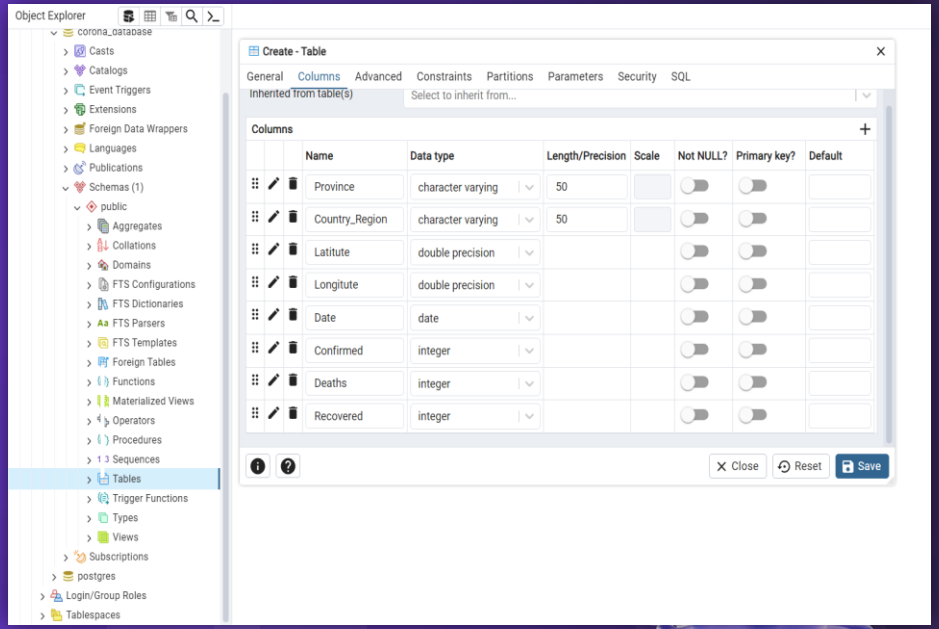
Creating Database



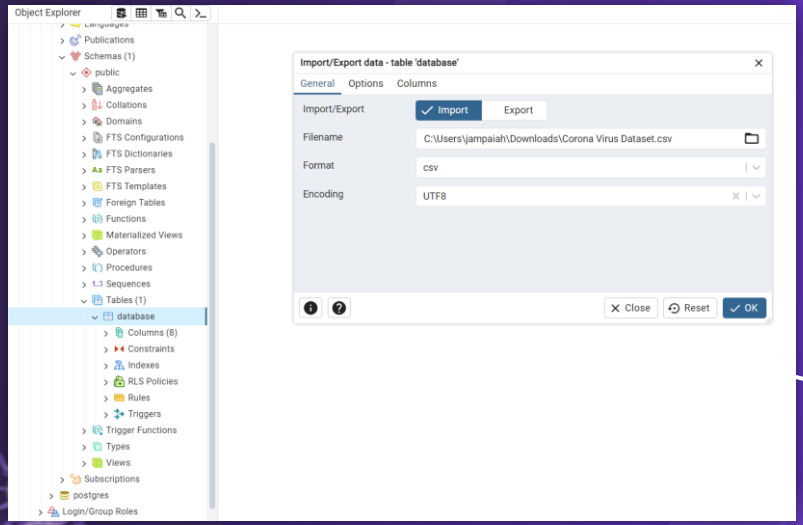


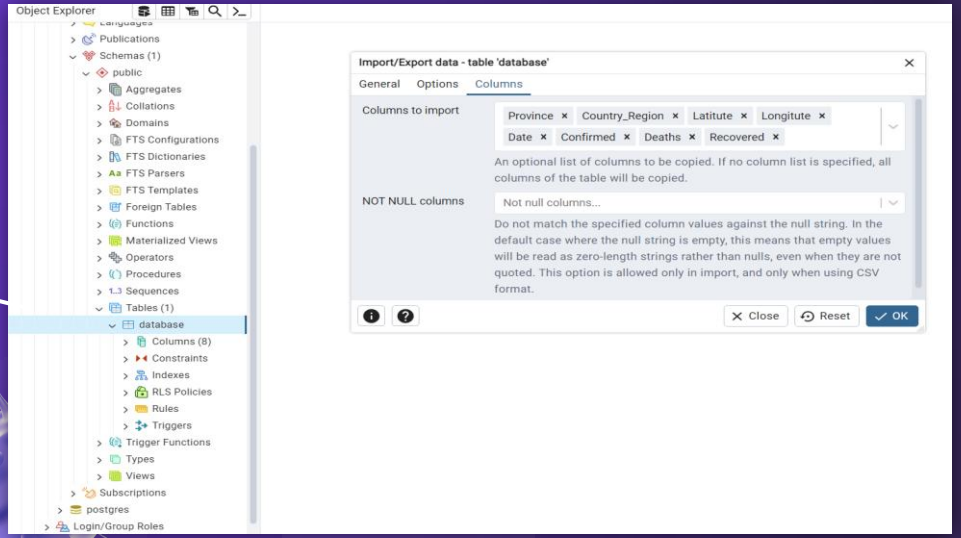
Creating Table In SQL



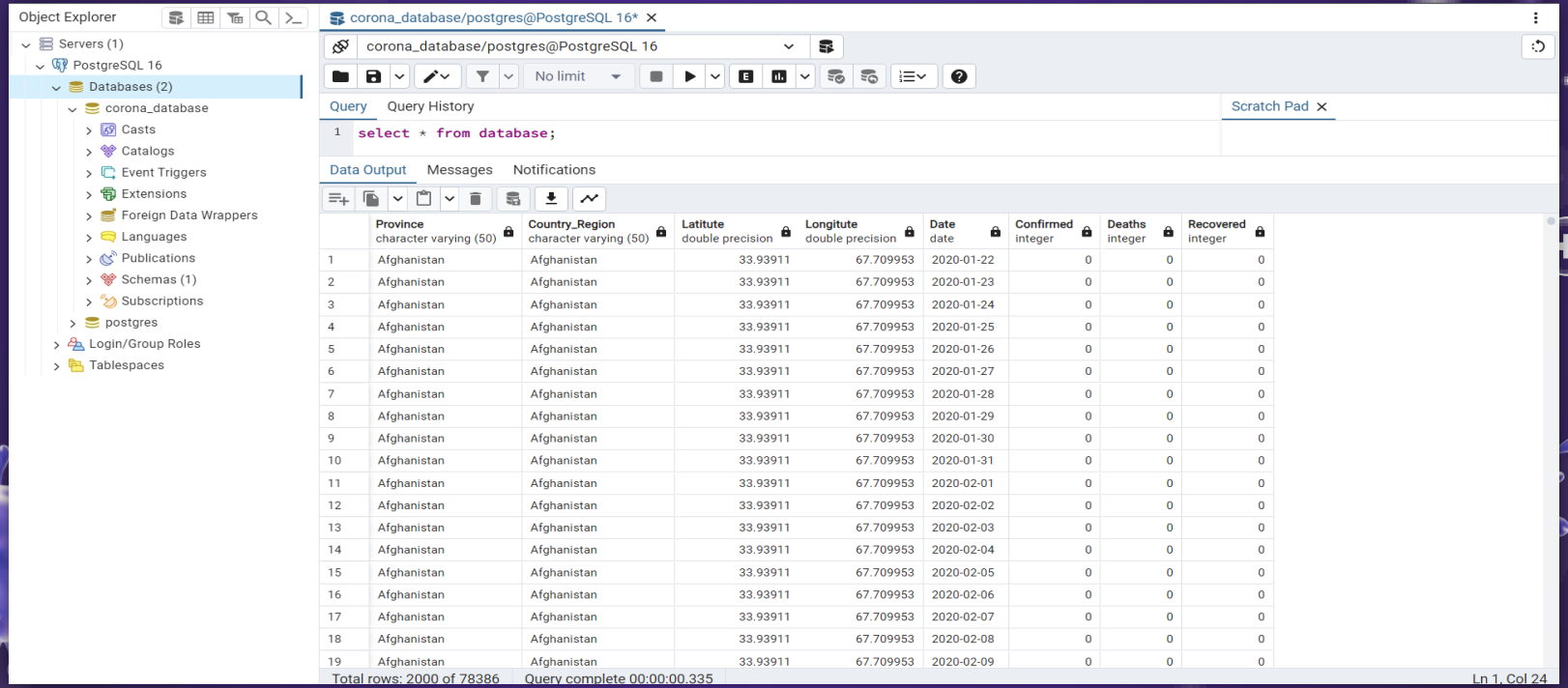


Import Data Into Table



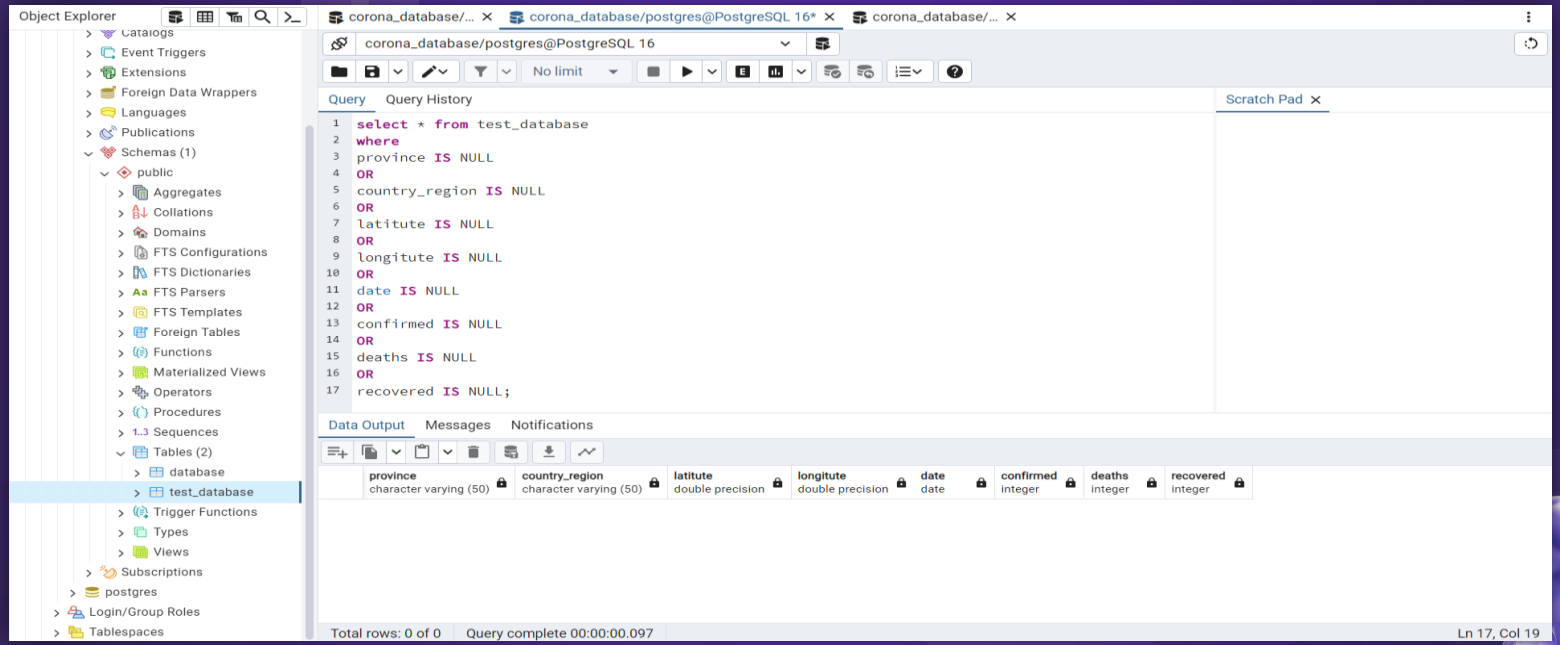


Import Data Into Table



	Province character varying (50)	Country_Region character varying (50)	Latitute double precision	Longitute double precision	Date date	Confirmed integer	Deaths integer	Recovered integer
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

1. Write a code to check NULL values



MENTOR NESS

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

Query | Query History

```
UPDATE test_database
SET
     province = COALESCE(province,'Not Applicable'),
     country_region = COALESCE(country_region, 'Not Applicable'),
     latitute = COALESCE(latitute, 0.0),
     longitute = COALESCE(longitute, 0.0),
     date = COALESCE(Date, '1970-01-01'::DATE),
     confirmed = COALESCE(confirmed,0),
     deaths = COALESCE(deaths,0),
     recovered = COALESCE(recovered,0);
```

Data Output | Messages | Notifications

```
UPDATE 78386

Query returned successfully in 345 msec.
```

3. check total number of rows

Query | Query History

```
select count(*) as total_rows
From test_database;
```

Data Output | Messages | Notifications

	total_rows bigint
1	78386

4. Check what is start date and end date

Query | Query History

```
select MIN(Date) AS start_date,MAX(Date) AS end_date
From test_database;
```

Data Output | Messages | Notifications

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

5. Number of month present in dataset

Query | Query History

```
select EXTRACT(MONTH FROM date) AS month_number, Count(*) as month_count
From test_database
Group By month_number
Order By month_number;
```

Data Output | Messages | Notifications

	month_number numeric	month_count 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

MENTOR NESS

6. Find monthly average for confirmed, deaths, recovered

Query | Query History

```sql
SELECT
    EXTRACT(YEAR FROM Date) As year_num,
    EXTRACT(MONTH FROM Date) As month_num,
    ROUND (AVG (Confirmed), 2) AS confirmed_avg,
    ROUND (AVG (Deaths), 2) AS deaths_avg,
    ROUND (AVG (Recovered), 2) AS recovered_avg
FROM test_database
GROUP BY year_num, month_num
ORDER BY year_num, month_num ASC;
```

Data Output | Messages | Notifications

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

Total rows: 18 of 18 | Query complete 00:00:00.329

MENTOR NESS

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

Query | Query History

```sql
WITH FrequentValues AS (
    SELECT
        EXTRACT (MONTH FROM Date) as month_num,
        EXTRACT (YEAR FROM Date) as year_num,
        Confirmed,
        Deaths,
        Recovered,
        RANK() OVER (PARTITION BY EXTRACT (MONTH FROM Date),
                     EXTRACT (YEAR FROM Date)
                     ORDER BY COUNT(*) DESC) as rank
    FROM
        test_database
    GROUP BY
        EXTRACT (MONTH FROM Date), EXTRACT (YEAR FROM Date), Confirmed, Deaths, Recovered
)
SELECT
    month_num,
    year_num,
    Confirmed,
    Deaths,
    recovered
FROM
    FrequentValues
WHERE
    rank = 1
ORDER BY
    year_num, month_num ASC;
```

Data Output | Messages | Notifications

	month_num numeric	year_num 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

Total rows: 18 of 18 | Query complete 00:00:00.209

MENTOR NESS

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

Query | Query History

```sql
SELECT
     EXTRACT (YEAR FROM Date) As Year_num,
     MIN(confirmed) AS min_confirmed,
     MIN(deaths) AS min_deaths,
     MIN(recovered) AS min_recovered
FROM
     test_database
GROUP BY
     year_num
ORDER BY
     year_num ASC;
```

Data Output | Messages | Notifications

	year_num numeric	min_confirmed integer	min_deaths integer	min_recovered integer
1	2020	0	0	0
2	2021	0	0	0

9. Find maximum values of confirmed, deaths, recovered per year

Query | Query History

```sql
SELECT
     EXTRACT (YEAR FROM Date) As Year_num,
     MAX(confirmed) AS min_confirmed,
     MAX(deaths) AS min_deaths,
     MAX(recovered) AS min_recovered
FROM
     test_database
GROUP BY
     year_num
ORDER BY
     year_num ASC;
```

Data Output | Messages | Notifications

	year_num numeric	min_confirmed integer	min_deaths integer	min_recovered integer
1	2020	823225	3752	1123456
2	2021	414188	7374	422436

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

Query | Query History

```sql
SELECT
    EXTRACT(YEAR FROM Date) AS Year_num,
    EXTRACT(MONTH FROM Date) AS Month_num,
    SUM(confirmed) AS total_confirmed,
    SUM(deaths) AS total_deaths,
    SUM(recovered) AS total_recovered
FROM
    test_database
GROUP BY
    Year_num, Month_num
ORDER BY
    Year_num ASC, Month_num ASC;
```

Data Output | Messages | Notifications

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

MENTOR NESS

11. Check how corona virus spread out with respect to confirmed case (total confirmed cases, their average, variance & STDEV)

Query | Query History

```sql
SELECT
    SUM(confirmed) AS total_confirmed_cases,
    AVG(confirmed) AS average_confirmed_cases,
    VARIANCE(confirmed) AS variance_confirmed_cases,
    STDDEV(confirmed) AS stddev_confirmed_cases
FROM
    test_database;
```

Data Output | Messages | Notifications

	total_confirmed_cases bigint	average_confirmed_cases numeric	variance_confirmed_cases numeric	stddev_confirmed_cases numeric
1	169065144	2156.8283111780164825	157290931.69817455	12541.56815148

MENTOR NESS

12. Check how corona virus spread out with respect to death case per month (total confirmed cases, their average, variance & STDEV)

Query | Query History

```sql
select
    Extract(year From Date) As year_num,
    Extract(month From Date) As month_num,
    SUM(Deaths) AS total_deaths,
    ROUND(AVG(Deaths),2) AS avg_deaths,
    ROUND(VARIANCE(Deaths),2) AS variance_deaths,
    ROUND(STDDEV(Deaths),2) AS standard_dev_deaths
FROM test_database
GROUP BY year_num,month_num
ORDER BY year_num,month_num ASC;
```

Data Output | Messages | Notifications

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

MENTOR NESS

13. Check how corona virus spread out with respect to recovered case (total confirmed cases, their average, variance & STDEV)

Query | Query History

```sql
select
    Extract(year From Date) As year_num,
    Extract(month From Date) As month_num,
    SUM(Recovered) AS total_Recovered,
    ROUND(AVG(Recovered),2) AS avg_recovered,
    ROUND(VARIANCE(Deaths),2) AS variance_deaths,
    ROUND(STDDEV(Deaths),2) AS standard_dev_deaths
FROM test_database
GROUP BY year_num,month_num
ORDER BY year_num,month_num ASC;
```

Data Output | Messages | Notifications

	year_num numeric	month_num numeric	total_recovered bigint	avg_recovered numeric	variance_deaths numeric	standard_dev_deaths numeric
1	2020	1	143	0.09	4.25	2.06
2	2020	2	31405	7.03	68.34	8.27
3	2020	3	133070	27.87	3901.61	62.46
4	2020	4	792987	171.64	40513.04	201.28
5	2020	5	1519547	318.30	20689.25	143.84
6	2020	6	2535417	548.79	16933.11	130.13
7	2020	7	4693120	983.06	21144.58	145.41
8	2020	8	6202833	1299.29	23277.87	152.57
9	2020	9	6647749	1438.91	20107.12	141.80
10	2020	10	6782150	1420.64	17583.75	132.60
11	2020	11	9172292	1985.34	27779.81	166.67
12	2020	12	11924903	2497.89	65359.06	255.65
13	2021	1	9164347	1919.64	102779.96	320.59
14	2021	2	6719785	1558.39	68494.76	261.72
15	2021	3	7888013	1652.29	54397.36	233.23
16	2021	4	14205507	3074.79	94631.95	307.62
17	2021	5	19131842	4007.51	131797.08	363.04
18	2021	6	5544438	2769.45	113020.13	336.18

Total rows: 18 of 18 | Query complete 00:00:00.352

MENTOR NESS

14. Find Country having highest number of the Confirmed case

Query | Query History

```sql
select
    Country_Region,
    SUM(Confirmed) AS total_confirmed_cases
From test_database
GROUP BY Country_Region
ORDER BY total_confirmed_cases DESC
LIMIT 1;
```

Data Output | Messages | Notifications

	country_region character varying (50)	total_confirmed_cases bigint
1	US	33461982

MENTOR NESS

15. Find Country having lowest number of the death case

Query | Query History

```sql
WITH rankingCountry AS (
    Select
        Country_region AS Country,
        SUM(Deaths) AS total_death_reported,
        RANK() OVER(ORDER by SUM(Deaths) ASC) AS rank_no
    FROM
        test_database
    GROUP BY
        Country
)
SELECT
    Country
    total_death_reported
FROM
    rankingCountry
WHERE
    rank_no = 1;
```

Data Output | Messages | Notifications

	total_death_reported character varying (50)
1	Samoa
2	Kiribati
3	Dominica
4	Marshall Islands

MENTOR NESS

16. Find top 5 countries having highest recovered case

Query | Query History

```
SELECT
    Country_Region,
    SUM(Recovered) AS total_recovered_cas
FROM test_database
GROUP BY Country_Region
ORDER BY total_recovered_cases DESC
LIMIT 5;
```

Data Output | Messages | Notifications

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

MENTOR NESS

INSIGHTS

1. COVID-19 Pandemic duration: June 22, 2020, to January 13, 2021.
2. India has the highest number of recovered cases.
3. Samoa, Kiribati, Dominica, and the Marshall Islands have the lowest death counts.
4. The US leads in confirmed COVID-19 cases.
5. Peak confirmed cases occurred in April 2021.
6. Peak death rate in January 2021.

MENTOR NESS

SUMMARY

Data Gathering: First, we collect data from various sources like hospitals, health departments, and research institutes. This includes information on confirmed cases, deaths, recoveries, and demographic details.

Data Cleaning: Next, we clean the data to remove any inconsistencies, errors, or missing values. This ensures that our analysis is based on accurate information.

Exploratory Analysis: We use SQL queries to explore the data, looking for patterns, trends, and correlations. This helps us understand how the virus is spreading, which regions are most affected, and how different factors like age and gender influence outcomes.

Aggregation: We aggregate the data to summarize key metrics such as total cases, deaths, and recovery rates for different countries, regions, and time periods. This allows us to compare the impact of the virus across different areas and track its progression over time.

THANKS!

Does anyone have any questions?

jampaiahboda@gmail.com
+91 6303440165