Corona Virus Analysis

by Jaloliddin Sultonov

Overview

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

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.

Q1. Write a code to check NULL values

```
SELECT * FROM corona
WHERE province IS NULL OR
'Country/Region' IS NULL OR
latitude IS NULL OR
longitude IS NULL OR
date IS NULL OR
confirmed IS NULL OR
deaths IS NULL OR
recovered IS NULL
```

Result

	province character varying (50)	Country/Region character varying (50)	latitude numeric	longitude numeric	date character varying (50)	confirmed integer	deaths integer	recovered integer

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

```sql
UPDATE corona
SET
    province = COALESCE(province, ''),
    "Country/Region" = COALESCE('Country/Region', ''),
    latitude = COALESCE(latitude, 0),
    longitude = COALESCE(longitude, 0),
    date = COALESCE(date, ''),
    confirmed = COALESCE(confirmed, 0),
    deaths = COALESCE(deaths, 0),
    recovered = COALESCE(recovered, 0)
WHERE
    province IS NULL OR
    "Country/Region" IS NULL OR
    latitude IS NULL OR
    longitude IS NULL OR
    date IS NULL OR
    confirmed IS NULL OR
    deaths IS NULL OR
    recovered IS NULL;
```

Result

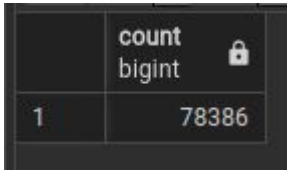
```
UPDATE 0

Query returned successfully in 49 msec.
```

Q3. Check total number of rows

```
SELECT COUNT(*) FROM corona
```

Result



	count bigint
1	78386

Q4. Check what is start_date and end_date

```
SELECT MIN(CAST(date AS DATE)) AS start_date,
MAX(CAST(date AS DATE)) AS end_date
FROM corona;
```

Result

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

Q5. Number of month present in dataset

```sql
SELECT
    COUNT(DISTINCT EXTRACT(YEAR FROM CAST(date AS DATE)) || '-' || EXTRACT(MONTH FROM CAST(date AS DATE)))
    AS num_months
FROM corona;
```

Data Output | Messages | Notifications

	num_months bigint
1	18

Q6. Find monthly average for confirmed, deaths, recovered

```sql
SELECT EXTRACT(YEAR FROM CAST(date AS DATE)) AS year,
EXTRACT(MONTH FROM CAST(date AS DATE)) AS month,
AVG(confirmed) AS avg_confirmed,
AVG(deaths) AS avg_deaths,
AVG(recovered) AS avg_recovered
FROM corona
GROUP BY EXTRACT(YEAR FROM CAST(date AS DATE)), EXTRACT(MONTH FROM CAST(date AS DATE))
ORDER BY year, month;
```

Data Output | Messages | Notifications

	year numeric	month numeric	avg_confirmed numeric	avg_deaths numeric	avg_recovered numeric
1	2020	1	4.1454545454545455	0.12337662337662337662	0.09285714285714285714
2	2020	2	15.2960143304970891	0.59359605911330049261	7.0320197044334975
3	2020	3	161.1302890657729367	8.6606619187264349	27.8739002932551320
4	2020	4	505.8004329004329004	41.5222943722943723	171.6422077922077922
5	2020	5	574.8498114788437369	30.2808965228320067	318.2963971512358609
6	2020	6	859.2281385281385281	29.8175324675324675	548.7915584415584416
7	2020	7	1432.3611227482195224	35.1095517385839966	983.0582320904901550
8	2020	8	1611.8428990364474235	37.5366568914956012	1299.2947214076246334
9	2020	9	1784.5874458874458874	34.7772727272727273	1438.9067099567099567
10	2020	10	2412.1996229576874738	36.7582739840804357	1420.6430666108085463
11	2020	11	3592.1943722943722944	56.7634199134199134	1985.3445887445887446
12	2020	12	4050.4396732299958106	71.2182656053623796	2497.8850020946795140
13	2021	1	3911.2285295349811479	84.1837033933808127	1919.6369920402178467
14	2021	2	2433.3636363636363636	69.1648886827458256	1558.3916975881261596
15	2021	3	2916.7972350230414747	59.1998324256388773	1652.2859237536656891
16	2021	4	4699.3551948051948052	78.4387445887445887	3074.7850649350649351
17	2021	5	4005.2540846250523670	76.7802681189777964	4007.5077503142019271
18	2021	6	2508.6323676323676324	66.2622377622377622	2769.4495504495504496

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

```sql
SELECT
    EXTRACT(YEAR FROM CAST(date AS DATE)) AS year,
    EXTRACT(MONTH FROM CAST(date AS DATE)) AS month,
    (SELECT confirmed
     FROM corona
     WHERE EXTRACT(YEAR FROM CAST(date AS DATE)) = EXTRACT(YEAR FROM CAST(corona.date AS DATE))
       AND EXTRACT(MONTH FROM CAST(date AS DATE)) = EXTRACT(MONTH FROM CAST(corona.date AS DATE))
     GROUP BY confirmed
     ORDER BY COUNT(*) DESC
     LIMIT 1) AS most_frequent_confirmed,
    (SELECT deaths
     FROM corona
     WHERE EXTRACT(YEAR FROM CAST(date AS DATE)) = EXTRACT(YEAR FROM CAST(corona.date AS DATE))
       AND EXTRACT(MONTH FROM CAST(date AS DATE)) = EXTRACT(MONTH FROM CAST(corona.date AS DATE))
     GROUP BY deaths
     ORDER BY COUNT(*) DESC
     LIMIT 1) AS most_frequent_deaths,
    (SELECT recovered
     FROM corona
     WHERE EXTRACT(YEAR FROM CAST(date AS DATE)) = EXTRACT(YEAR FROM CAST(corona.date AS DATE))
       AND EXTRACT(MONTH FROM CAST(date AS DATE)) = EXTRACT(MONTH FROM CAST(corona.date AS DATE))
     GROUP BY recovered
     ORDER BY COUNT(*) DESC
     LIMIT 1) AS most_frequent_recovered
FROM
    corona
GROUP BY
    EXTRACT(YEAR FROM CAST(date AS DATE)), EXTRACT(MONTH FROM CAST(date AS DATE))
ORDER BY
    year, month;
```

Result

	year numeric	month numeric	most_frequent_confirmed integer	most_frequent_deaths integer	most_frequent_recovered integer
1	2020	1	0	0	0
2	2020	2	0	0	0
3	2020	3	0	0	0
4	2020	4	0	0	0
5	2020	5	0	0	0
6	2020	6	0	0	0
7	2020	7	0	0	0
8	2020	8	0	0	0
9	2020	9	0	0	0
10	2020	10	0	0	0
11	2020	11	0	0	0
12	2020	12	0	0	0
13	2021	1	0	0	0
14	2021	2	0	0	0
15	2021	3	0	0	0
16	2021	4	0	0	0
17	2021	5	0	0	0
18	2021	6	0	0	0

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

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

Data Output | Messages | Notifications

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

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

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

Data Output | Messages | Notifications

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

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

```sql
SELECT
    EXTRACT(YEAR FROM CAST(date AS DATE)) AS year,
    EXTRACT(MONTH FROM CAST(date AS DATE)) AS month,
    SUM(confirmed) AS total_confirmed,
    SUM(deaths) AS total_deaths,
    SUM(recovered) AS total_recovered
FROM
    corona
GROUP BY
    EXTRACT(YEAR FROM CAST(date AS DATE)),
    EXTRACT(MONTH FROM CAST(date AS DATE))
ORDER BY
    year, month;
```

Result

Data Output | Messages | Notifications

	year numeric	month 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

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

```sql
SELECT
    total_confirmed_cases,
    average_confirmed_cases,
    variance_confirmed_cases,
    SQRT(variance_confirmed_cases) AS stdev_confirmed_cases
FROM (
    SELECT
        (SELECT SUM(confirmed) FROM corona) AS total_confirmed_cases,
        (SELECT AVG(confirmed) FROM corona) AS average_confirmed_cases,
        (SELECT VARIANCE(confirmed) FROM corona) AS variance_confirmed_cases
) AS results;
```

Data Output | Messages | Notifications

	total_confirmed_cases bigint	average_confirmed_cases numeric	variance_confirmed_cases numeric	stdev_confirmed_cases numeric
1	169065144	2156.8283111780164825	157290931.69817455	12541.56815147829

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

```sql
SELECT
    total_deaths_cases,
    average_deaths_cases,
    variance_deaths_cases,
    SQRT(variance_deaths_cases) AS stdev_deaths_cases
FROM (
    SELECT
        (SELECT SUM(deaths) FROM corona) AS total_deaths_cases,
        (SELECT AVG(deaths) FROM corona) AS average_deaths_cases,
        (SELECT VARIANCE(deaths) FROM corona) AS variance_deaths_cases
) AS results;
```

Data Output | Messages | Notifications

	total_deaths_cases bigint	average_deaths_cases numeric	variance_deaths_cases numeric	stdev_deaths_cases numeric
1	3647894	46.5375704845252979	45892.604322956217	214.2255921288496

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

```sql
SELECT
    total_recovered_cases,
    average_recovered_cases,
    variance_recovered_cases,
    SQRT(variance_recovered_cases) AS stdev_recovered_cases
FROM (
    SELECT
        (SELECT SUM(recovered) FROM corona) AS total_recovered_cases,
        (SELECT AVG(recovered) FROM corona) AS average_recovered_cases,
        (SELECT VARIANCE(recovered) FROM corona) AS variance_recovered_cases
) AS results;
```

Data Output | Messages | Notifications

	total_recovered_cases bigint	average_recovered_cases numeric	variance_recovered_cases numeric	stdev_recovered_cases numeric
1	113089548	1442.7263541959023295	107030888.69602982	10345.57338652768

Q14. Find Country having highest number of the Confirmed case

```sql
SELECT "Country/Region" AS country, SUM(confirmed) AS highest_confirmed_cases
FROM corona
GROUP BY "Country/Region"
ORDER BY highest_confirmed_cases DESC
LIMIT 1;
```

Data Output | Messages | Notifications

	country character varying (50)	highest_confirmed_cases bigint
1	US	33461982

Q15. Find Country having lowest number of the death case

```sql
SELECT "Country/Region" AS country, SUM(deaths) AS lowest_deaths_cases
FROM corona
GROUP BY "Country/Region"
ORDER BY lowest_deaths_cases
LIMIT 4;
```

Data Output | Messages | Notifications

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

Q16. Find top 5 countries having highest recovered case

```sql
SELECT "Country/Region" AS country, SUM(recovered) AS highest_recovered_cases
FROM corona
GROUP BY "Country/Region"
ORDER BY highest_recovered_cases DESC
LIMIT 5;
```

Data Output | Messages | Notifications

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

Thank you!