Corona Virus Analysis

(SQL Project)

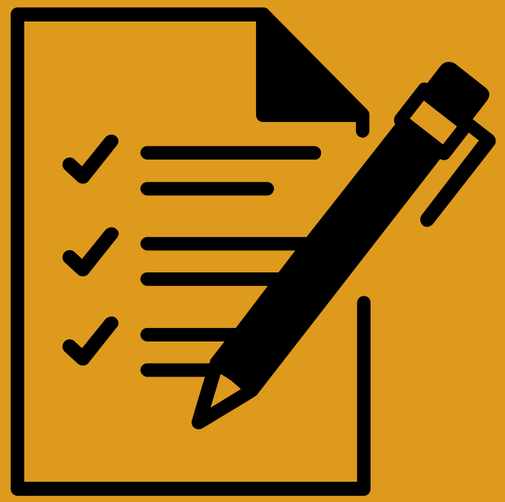
Problem statement

This project is designed to test your SQL and data analysis skills in a real-world context. You are encouraged to be a creative in your approach and to seek guidance and assistance as needed through out the internship

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.

Table Creation

```
DROP TABLE IF EXISTS Corona_virus_dataset;
CREATE TABLE Corona_virus_dataset(
        Province VARCHAR(50) NOT NULL,
        Country VARCHAR(50) NOT NULL,
        latitude FLOAT NOT NULL,
        longitude FLOAT NOT NULL,
        date date  NOT NULL,
        confirmed INT NOT NULL,
        deaths INT NOT NULL,
        recovered INT NOT NULL);

COPY "public"."corona_virus_dataset" FROM
'C:\Corona Virus Dataset.csv' DELIMITER ',' CSV HEADER;
```

Problem- 1

Write a code to check all the NULL values?

Query:

```
SELECT * FROM "public"."corona_virus_dataset"
WHERE (province  IS NULL)OR
      (country 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	country	latitude	longitude	date	confirmed	deaths	recovered

Problem- 2

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

Result:

NO NULL Values are there

Problem- 3

Check the total no of rows

Query:

```
SELECT
        count(row_number) AS total_rows
FROM
        (SELECT
                *,
                row_number()OVER()
FROM
        "public"."corona_virus_dataset")
```

Result:

	total_rows
1	78386

Problem- 4

Check what is the start date and end date

Query:

```
SELECT
        min(date)AS start_date,
        max(date) AS end_date
FROM "public"."corona_virus_dataset"
```

Result:

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

Problem- 5

Number of months present in the dataset

Query:

```
SELECT
        count(DISTINCT(month_name))AS no_of_months
FROM
        (SELECT
                *,
                TO_CHAR(date,'month')AS month_name
 FROM
        "public"."corona_virus_dataset")
```

Result:

	no_of_months
1	12

Problem- 6

Find monthly average for confirmed, deaths, recovered?

Query:

```
SELECT
        month_name,
        round(avg(confirmed))AS avg_confirmed,
        round(avg(deaths))AS avg_deaths,
        round(avg(recovered))AS avg_recovered
FROM
       (SELECT
              *,
              TO_CHAR(date,'month')AS month_name
FROM
     "public"."corona_virus_dataset")AS a
GROUP BY
       month_name
```

Result:

	month_name	avg_confirmed	avg_deaths	avg_recovered
1	april	2603	60	1623
2	june	1358	41	1220
3	december	4050	71	2498
4	february	1203	34	769
5	november	3592	57	1985
6	october	2412	37	1421
7	january	2958	64	1451
8	september	1785	35	1439
9	march	1539	34	840
10	august	1612	38	1299
11	may	2290	54	2163
12	july	1432	35	983

Problem- 7

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

Query:

```
SELECT * FROM
(SELECT
      to_char(date,'month') AS MONTH,
      confirmed,
      deaths,
      recovered,
      rank()
      OVER(PARTITION BY to_char(date,'month')
      ORDER BY count(*) DESC) AS rank
FROM "public"."corona_virus_dataset"
GROUP BY to_char(date,'month'),
            confirmed,
            deaths,
            recovered) AS a
WHERE rank = 1
ORDER BY MONTH
```

Result:

	month	confirmed	deaths	recovered	rank
1	april	0	0	0	1
2	august	0	0	0	1
3	december	0	0	0	1
4	february	0	0	0	1
5	january	0	0	0	1
6	july	0	0	0	1
7	june	0	0	0	1
8	march	0	0	0	1
9	may	0	0	0	1
10	november	0	0	0	1
11	october	0	0	0	1
12	september	0	0	0	1

Problem- 8

Find minimum values for confirmed, deaths, recovered per year?

Query:

```
SELECT
        EXTRACT(YEAR FROM date) AS YEAR,
        min(confirmed) AS min_confirmed,
        min(deaths) AS min_daeth,
        min(recovered) AS min_recovered
FROM "public"."corona_virus_dataset"
GROUP BY YEAR
```

Result:

	year	min_confirmed	min_daeth	min_recovered
1	2021	0	0	0
2	2020	0	0	0

Problem- 9

Find maximum values for confirmed, deaths, recovered per year?

Query:

```
SELECT
        EXTRACT(YEAR FROM date) AS YEAR,
        max(confirmed) AS max_confirmed,
        max(deaths) AS max_death,
        max(recovered) AS max_recovered
FROM "public"."corona_virus_dataset"
GROUP BY YEAR
```

Result:

	year	max_confirmed	max_daeth	max_recovered
1	2021	414188	7374	422436
2	2020	823225	3752	1123456

Problem- 10

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

Query:

```
SELECT
        to_char(date,'month') AS MONTH,
        sum(confirmed) AS total_confirmed,
        sum(deaths) AS total_deaths,
        sum(recovered) AS total_recovered
FROM "public"."corona_virus_dataset"
GROUP BY MONTH
```

Result:

	month	total_confirmed	total_deaths	total_recovered
1	april	24047819	554220	14998494
2	june	8991916	270414	8079855
3	december	19336799	339996	11924903
4	february	10560976	300890	6751190
5	november	16595938	262247	9172292
6	october	11515841	175484	6782150
7	january	18678589	402083	9164490
8	september	8244794	160671	6647749
9	march	14694026	323966	8021083
10	august	7694938	179200	6202833
11	may	21865416	511110	20651389
12	july	6828092	167612	4693129

Problem- 11

Check how corona virus spread out with respect to confirmed case

Query:

```
SELECT
        sum(confirmed)AS total_confirmed,
        round(avg(confirmed))AS average,
        round(variance(confirmed))AS variance,
        round(stddev(confirmed))AS standard_deviation
FROM "public"."corona_virus_dataset"
```

Result:

	total_confirmed	average	variance	standard_deviation
1	169065144	2157	157290932	12542

Problem- 12

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

Query:

```
SELECT
        to_char(date,'month')AS MONTH,
        sum(deaths)AS total_deaths,
        round(avg(deaths))AS avg_deaths,
        round(variance(deaths))AS var_death,
        round(stddev(deaths))AS std_deaths
FROM "public"."corona_virus_dataset"
GROUP BY MONTH
```

Result:

	month	total_deaths	avg_deaths	var_death	std_deaths
1	april	554220	60	67906	261
2	june	270414	41	46250	215
3	december	339996	71	65359	256
4	february	300890	34	34853	187
5	november	262247	57	27780	167
6	october	175484	37	17584	133
7	january	402083	64	79012	281
8	september	160671	35	20107	142
9	march	323966	34	29785	173
10	august	179200	38	23278	153
11	may	511110	54	76776	277
12	july	167613	35	21145	145

Problem- 13

Check how corona virus spread out with respect to recovered case

Query:

```
SELECT
      sum(recovered)AS total_recovered,
      round(avg(recovered))AS avg_recovered,
      round(variance(recovered))AS var_recovered,
      round(stddev(recovered))AS std_recovered
FROM "public"."corona_virus_dataset"
```

Result:

	total_recovered	avg_recovered	var_recovered	std_recovered
1	113089548	1443	107030889	10346

Problem- 14

Find Country having highest number of the Confirmed case?

Query:

```
SELECT
        country,
        sum(confirmed) AS total_confirmed
FROM "public"."corona_virus_dataset"
GROUP BY country
ORDER BY total_confirmed DESC
LIMIT 1
```

Result:

	country	total_confirmed
1	US	33461982

Problem- 15

Find Country having lowest number of the death case ?

Query:

```
SELECT
	country,
	sum(deaths)AS total_deaths
FROM "public"."corona_virus_dataset"
GROUP BY country
ORDER BY total_deaths
LIMIT 4
```

Result:

	country	total_deaths
1	Dominica	0
2	Marshall Islands	0
3	Kiribati	0
4	Samoa	0

Problem- 16

Find top 5 countries having highest recovered case?

Query:

```
SELECT
      country,
      sum(recovered)AS total_recovered
FROM "public"."corona_virus_dataset"
GROUP BY country
ORDER BY total_recovered DESC
LIMIT 5
```

Result:

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

Thank you!