Corona Virus Analysis

presented By

CHANDANA S

Batch : MIP-DA-11

Role : Data Analyst Intern

Overview

Introduction:

The Corona Virus Analysis project aims to derive meaningful insights from pandemic data.

Objective:

Analyze the dataset to inform public health decisions.

Dataset:

Includes data on confirmed cases, deaths, recoveries.

Tasks:

Answer 16 specific questions about the data.

Methodology:

SQL queries were used to analyze the dataset.

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 cases.

- **Deaths**: Number of related deaths.

- **Recovered**: Number of recovered cases.

Methodology

- **Approach**: SQL queries were used to extract insights from the dataset.

- **Tools Used**: - SQLite for database management.
- DB Browser for SQLite for running queries and visualizing results.

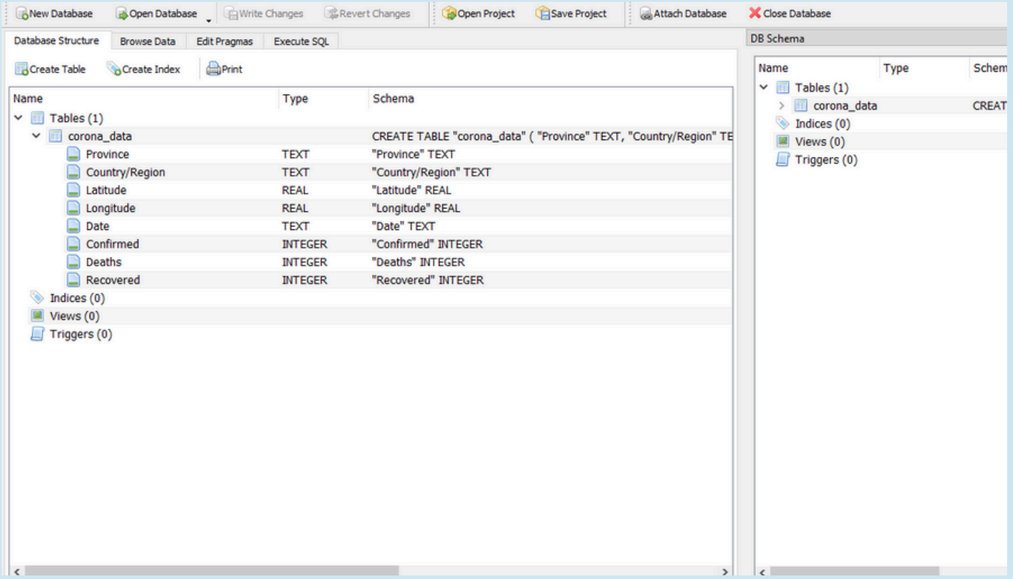
```python
import sqlite3
import pandas as pd
```

```python
csv_file_path = '/content/Corona Virus Dataset.csv'  # Update with your file path
df = pd.read_csv(csv_file_path)
df.head()  # Display the first few rows to verify the data
```

```python
# Create a connection to the SQLite database
conn = sqlite3.connect('corona_virus_analysis.db')

# Load the DataFrame into the SQLite database
df.to_sql('corona_data', conn, if_exists='replace', index=False)

# Verify that the table has been created
pd.read_sql('SELECT * FROM corona_data LIMIT 5', conn)
```



Analysis and Findings

-Q1. Write a code to check NULL values

SQL 1

```sql
SELECT
    SUM(CASE WHEN Province IS NULL THEN 1 ELSE 0 END) AS Null_Province,
    SUM(CASE WHEN "Country/Region" IS NULL THEN 1 ELSE 0 END) AS Null_Country_Region,
    SUM(CASE WHEN Latitude IS NULL THEN 1 ELSE 0 END) AS Null_Latitude,
    SUM(CASE WHEN Longitude IS NULL THEN 1 ELSE 0 END) AS Null_Longitude,
    SUM(CASE WHEN Date IS NULL THEN 1 ELSE 0 END) AS Null_Date,
    SUM(CASE WHEN Confirmed IS NULL THEN 1 ELSE 0 END) AS Null_Confirmed,
    SUM(CASE WHEN Deaths IS NULL THEN 1 ELSE 0 END) AS Null_Deaths,
    SUM(CASE WHEN Recovered IS NULL THEN 1 ELSE 0 END) AS Null_Recovered
FROM Corona_Data;
```

	ull_Province	Null_Country_Region	Null_Latitude	Null_Longitude	Null_Date	Null_Confirmed	Null_Deaths	Null_Recovere
1	0	0	0	0	0	0	0	

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

```sql
-- THERE IS NO NULL VALUES PRESENT IN THE GIVEN DATASET
```

Analysis and Findings

- Q3. check total number of rows

SQL 1 | SQL 2

```sql
SELECT COUNT(*) AS Total_Rows
FROM Corona_Data;
```

	Total_Rows
1	78386

SQL 1 | SQL 2 | SQL 3 | SQL 4

```sql
SELECT MIN(Date) AS start_date, MAX(Date) AS end_date FROM corona_data;
```

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

- Q4. Check what is start_date and end_date

Analysis and Findings

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1 | SQL 2 | SQL 3 | SQL 4 | SQL 5 | SQL 6

```sql
SELECT COUNT(DISTINCT strftime('%m', Date)) AS num_unique_months
    FROM corona_data
```

	num_unique_months
1	12

- Q5. Number of month present in dataset

- Q6. Find monthly average for confirmed, deaths, recovered

SQL 1 | SQL 2 | SQL 3 | SQL 4 | SQL 5 | SQL 6

```sql
SELECT strftime('%Y-%m', Date) AS Month,
       ROUND(AVG(Confirmed), 2) AS Avg_Confirmed,
       ROUND(AVG(Deaths), 2) AS Avg_Deaths,
       ROUND(AVG(Recovered), 2) AS Avg_Recovered
FROM corona_data
GROUP BY strftime('%Y-%m', Date);
```

	Month	Avg_Confirmed	Avg_Deaths	Avg_Recovered
1	2020-01	4.15	0.12	0.09
2	2020-02	15.3	0.59	7.03
3	2020-03	161.13	8.66	27.87
4	2020-04	505.8	41.52	171.64
5	2020-05	574.85	30.28	318.3
6	2020-06	859.23	29.82	548.79
7	2020-07	1432.36	35.11	983.06
8	2020-08	1611.84	37.54	1299.29
9	2020-09	1784.59	34.78	1438.91
10	2020-10	2412.2	36.76	1420.64
11	2020-11	3592.19	56.76	1985.34
12	2020-12	4050.44	71.22	2497.89
13	2021-01	3911.23	84.18	1919.64
14	2021-02	2433.36	69.16	1558.39

Analysis and Findings

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

```sql
SELECT SUBSTR(Date, 1, 7) AS month,
       MAX(Confirmed) AS most_freq_confirmed,
       MAX(Deaths) AS most_freq_deaths,
       MAX(Recovered) AS most_freq_recovered
FROM corona_data
GROUP BY month;
```

	month	most_freq_confirmed	most_freq_deaths	most_freq_recovered
1	2020-01	2131	49	51
2	2020-02	14840	242	3418
3	2020-03	26314	1085	4289
4	2020-04	50740	2607	33227
5	2020-05	34907	2309	51717
6	2020-06	54771	2003	94305
7	2020-07	75866	1595	140050
8	2020-08	85687	1505	95881
9	2020-09	97894	1703	101468
10	2020-10	99264	3351	388340
11	2020-11	207933	2259	139292
12	2020-12	823225	3752	1123456
13	2021-01	300462	4475	87090
14	2021-02	134975	3907	98389

```sql
SELECT SUBSTR(Date, 1, 4) AS year,
       MIN(Confirmed) AS min_confirmed,
       MIN(Deaths) AS min_deaths,
       MIN(Recovered) AS min_recovered
FROM corona_data
GROUP BY year;
```

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

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

Analysis and Findings

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

```sql
SELECT SUBSTR(Date, 1, 4) AS year,
       MAX(Confirmed) AS max_confirmed,
       MAX(Deaths) AS max_deaths,
       MAX(Recovered) AS max_recovered
FROM corona_data
GROUP BY year;
```

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

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

```sql
SELECT SUBSTR(Date, 1, 7) AS month,
       SUM(Confirmed) AS total_confirmed,
       SUM(Deaths) AS total_deaths,
       SUM(Recovered) AS total_recovered
FROM corona_data
GROUP BY month;
```

	month	total_confirmed	total_deaths	total_recovered
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

Analysis and Findings

Q11. Identify the top 10 countries/regions with the highest number of confirmed cases.

```sql
-- Top 10 Countries/Regions with the Highest Number of Confirmed Cases:
SELECT "Country/Region", SUM(Confirmed) AS total_confirmed_cases
FROM corona_data
GROUP BY "Country/Region"
ORDER BY total_confirmed_cases DESC
LIMIT 10;
```

	Country/Region	total_confirmed_cases
1	US	33461982
2	India	29460523
3	Brazil	17412766
4	France	6106009
5	Turkey	5330447
6	Russia	5148499
7	United Kingdom	4582386
8	Italy	4245020
9	Argentina	4124190
10	Spain	3818353

```sql
-- Daily Average Number of New Confirmed Cases Globally:
SELECT ROUND(AVG(daily_cases), 2) AS daily_avg_new_cases
FROM (
    SELECT Date, SUM(Confirmed) AS daily_cases
    FROM corona_data
    GROUP BY Date
) AS global_daily_cases;
```

	daily_avg_new_cases
1	332151.56

Q12. Calculate the daily average number of new confirmed cases globally.

Analysis and Findings

Q13. Determine the mortality rate (percentage) for each country/region.

SQL 1 SQL 2 SQL 3

```sql
-- Mortality Rate (Percentage) for Each Country/Region:
SELECT
    "Country/Region",
    SUM(Deaths) AS total_deaths,
    SUM(Confirmed) AS total_confirmed,
    ROUND((CAST(SUM(Deaths) AS FLOAT) / SUM(Confirmed)) * 100, 2) AS mortality_rate
FROM corona_data
GROUP BY "Country/Region";
```

	Country/Region	total_deaths	total_confirmed	mortality_rate
1	Afghanistan	3527	89861	3.92
2	Algeria	3571	133388	2.68
3	Argentina	85343	4124190	2.07
4	Australia	911	30365	3.0
5	Austria	10663	648620	1.64
6	Bangladesh	13118	826922	1.59
7	Barbados	47	4036	1.16
8	Belarus	2978	405663	0.73
9	Belgium	25205	1076338	2.34
10	Bhutan	1	1813	0.06
11	Bolivia	15542	406954	3.82
12	Bosnia and Herzegovina	9490	204597	4.64
13	Botswana	896	59480	1.51

```sql
SELECT "Country/Region",
       MAX(Confirmed) AS max_confirmed_cases
FROM corona_data;
```

	Country/Region	max_confirmed_cases
1	Turkey	823225

Q14. Find Country having highest number of the Confirmed case

Q15. Find Country having lowest number of the death case

SQL 3 | SQL 4 | SQL 5 | SQL 6 | SQL 7 | SQL 8 | SQL 9 | SQL 10

```sql
SELECT "Country/Region",
       MIN(Deaths) AS min_death_cases
FROM corona_data;
```

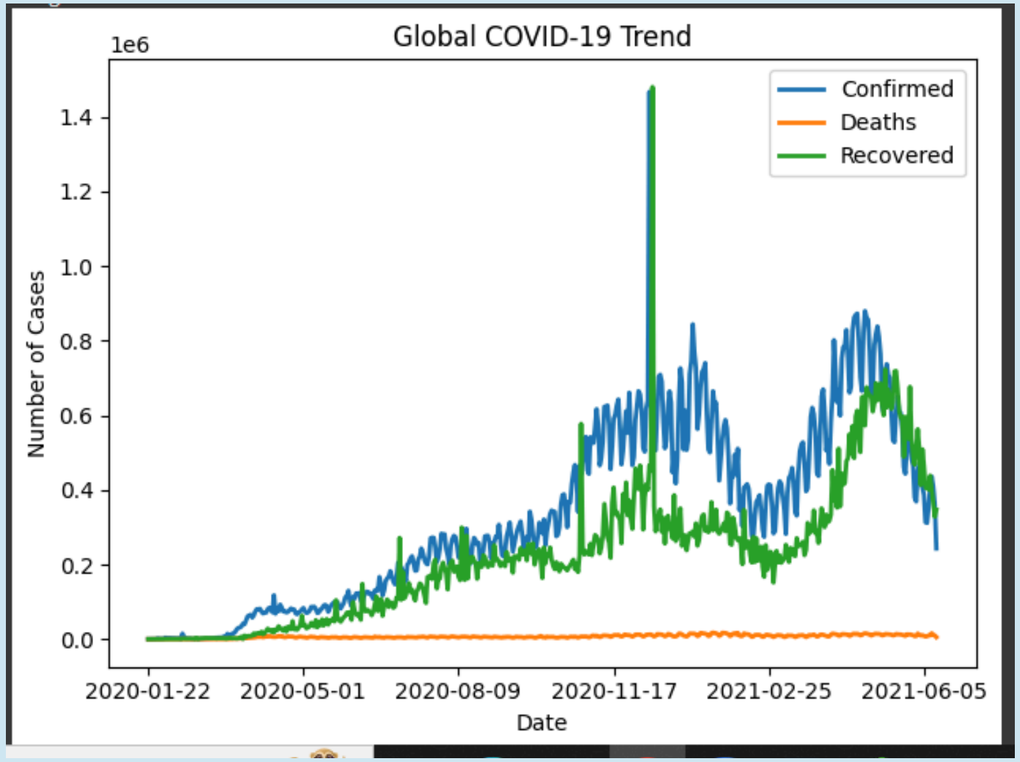
	Country/Region	min_death_cases
1	Afghanistan	0

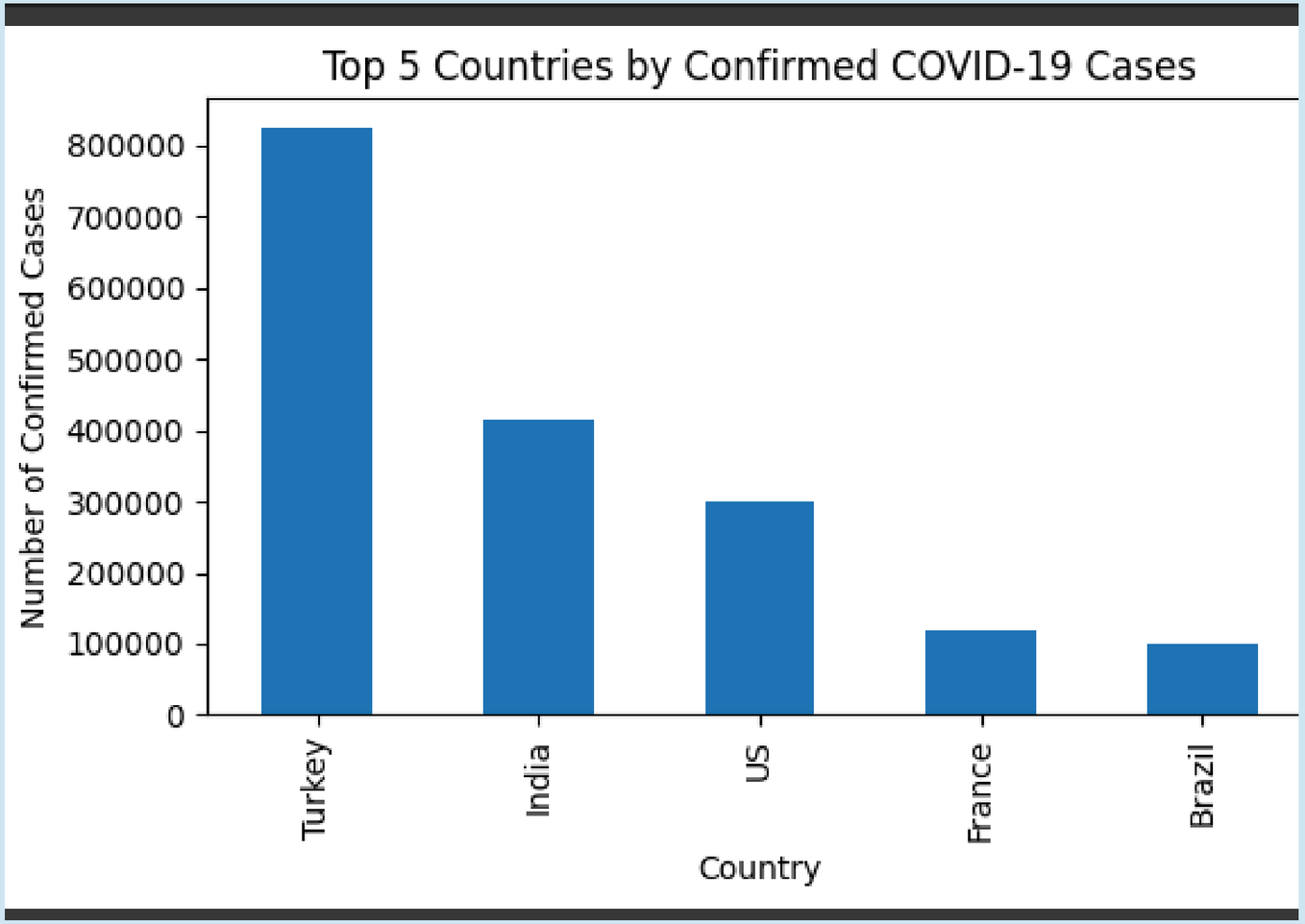
SQL 3 | SQL 4 | SQL 5 | SQL 6 | SQL 7 | SQL 8 | SQL 9

```sql
SELECT "Country/Region",
       MAX(Recovered) AS max_recovered_cases
FROM corona_data
GROUP BY "Country/Region"
ORDER BY max_recovered_cases DESC
LIMIT 5;
```

	Country/Region	max_recovered_cases
1	Turkey	1123456
2	India	422436
3	Brazil	388340
4	US	150267
5	Colombia	89557

Q16. Find top 5 countries having highest recovered case


Global COVID-19 Trend
1e6
Confirmed
Deaths
Recovered
Number of Cases
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2020-01-22
2020-05-01
2020-08-09
2020-11-17
2021-02-25
2021-06-05
Date


Top 5 Countries by Confirmed COVID-19 Cases
Number of Confirmed Cases
800000
700000
600000
500000
400000
300000
200000
100000
0
Turkey
India
US
France
Brazil
Country

Thank You