



Jeevarathnam R T

Corona virus Analysis

with SQL





Objective

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.



```
create database corona;  
  
select * From virus;  
  
ALTER TABLE virus  
CHANGE `Country/Region` Region varchar(50);
```

- 
- Create a database to store the dataset in MYSQL
 - Check the tables
 - Change the table name to easily callable and useful format

```
-- To avoid any errors, check missing value / null value
-- Q1. Write a code to check NULL values
select *
FROM Virus
where Province IS NULL OR
       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
;
```

	Province	Region	Latitude	Longitude	Confirmed	Deaths	Recovered	Date

- Check for the null values and show the values.
- IS NULL - used in where clause to check the null values
- This table has no null values.

```
-- Q2. If NULL values are present, update them with zeros for all columns.  
-- No Null Values
```

- There no null values so we cannot impute zeros.
- If we wanted to impute any null values by zero we need to use
- Syntax:
UPDATE [table] SET [column]=0 WHERE [column] IS NULL;

```
-- Q3. check total number of rows
SELECT count(*) AS Count_of_rows
From virus;
```

	Count_of_rows
▶	78386

- Count function is used to count the values
- We can count by column name or for total count of row we can use count(*)
- The Total count of rows is 78,386.

```

-- Q4. Check what is start_date and end_date
-- Step 1: Add a new date column
ALTER TABLE Virus ADD COLUMN Date_ DATE;

-- Step 2: Update the new column with converted date values
UPDATE Virus SET Date_ = STR_TO_DATE(date, '%d-%m-%Y');

-- Step 3: Drop the old text column
ALTER TABLE Virus DROP COLUMN date;

-- Step 4: Rename the new column to the original column name
ALTER TABLE Virus CHANGE COLUMN Date_ Date DATE;

Select min(date) AS Start_date,
       max(date) AS End_date
from virus;

```

- First we need to change the date column to date format
- Here we assigned date to new column and updated the column name to previous column name
- Drop the date column with text format
- Start date = 22-01-2020
- End date = 13-06-2021

Start_date	End_date
2020-01-22	2021-06-13


```
-- Q5. Number of month present in dataset
```

```
SELECT TIMESTAMPDIFF(MONTH, '2020-01-22', '2021-06-13') AS MonthDifference;
```

MonthDifference

16

- Here we use date difference to find the difference between two dates
- Calculated by month
- The number of months present is $16+1=17$ Months total.

```
-- Q6. Find monthly average for confirmed, deaths, recovered
SELECT  year(date) AS YEAR, monthname(date) AS MONTH,
        avg(confirmed) AS AVG_CONFIRMED_CASES,
        avg(deaths) AS AVG_DEATHS,
        avg(recovered) AS AVG_RECOVERED
FROM virus
GROUP BY monthname(date), year(date);
```

YEAR	MONTH	AVG_CONFIRMED_CASES	AVG_DEATHS	AVG_RECOVERED
2020	January	4.1455	0.1234	0.0929
2020	February	15.2960	0.5936	7.0320
2020	March	161.1303	8.6607	27.8739
2020	April	505.8004	41.5223	171.6422
2020	May	574.8498	30.2809	318.2964
2020	June	859.2281	29.8175	548.7916
2020	July	1432.3611	35.1096	983.0582
2020	August	1611.8429	37.5367	1299.2947
2020	September	1784.5874	34.7773	1438.9067
2020	October	2412.1996	36.7583	1420.6431
2020	November	3592.1944	56.7634	1985.3446
2020	December	4050.4397	71.2183	2497.8850
2021	January	3911.2285	84.1837	1919.6370
2021	February	2433.3636	69.1649	1558.3917
2021	March	2916.7972	59.1998	1652.2859
2021	April	4699.3552	78.4387	3074.7851
2021	May	4005.2541	76.7803	4007.5078
2021	June	2508.6324	66.2622	2769.4496

- calculated the average confirmed cases, deaths and recoveries
- For each month
- we used year to avoid months of different years been sum up into single month

```
-- Q7. Find most frequent value for confirmed, deaths, recovered each month
SELECT
    MONTH,
    confirmed, deaths, recovered
FROM (SELECT MONTHNAME(date) AS MONTH,
             confirmed, deaths, recovered,
             ROW_NUMBER() OVER (PARTITION BY MONTHNAME(date) ORDER BY COUNT(confirmed) DESC) AS rn_c,
             ROW_NUMBER() OVER (PARTITION BY MONTHNAME(date) ORDER BY COUNT(deaths) DESC) AS rn_d,
             ROW_NUMBER() OVER (PARTITION BY MONTHNAME(date) ORDER BY COUNT(recovered) DESC) AS rn_r
    FROM
        virus
    GROUP BY
        MONTHNAME(date), confirmed, deaths, recovered
) AS subquery
WHERE
    rn_c = 1 AND rn_d = 1 AND rn_r = 1;
```

MONTH	confirmed	deaths	recovered
April	0	0	0
August	0	0	0
December	0	0	0
February	0	0	0
January	0	0	0
July	0	0	0
June	0	0	0
March	0	0	0
May	0	0	0
November	0	0	0
October	0	0	0
September	0	0	0

- To find the most frequent value for each month
- Using subquery within FROM clause
- For those three columns we assign row number to by highest occurred value.
- In WHERE clause we put condition to get only values with row number 1

```
-- Q8. Find minimum values for confirmed, deaths, recovered per year
SELECT YEAR(DATE) AS YEAR,
       MIN(CONFIRMED) AS MIN_cases,
       MIN(DEATHS) AS MIN_deaths,
       MIN(RECOVERED) AS MIN_recovered
FROM VIRUS
GROUP BY YEAR;
```

YEAR	MIN_cases	MIN_deaths	MIN_recovered
2020	0	0	0
2021	0	0	0

- Used MIN() to find the min value for each year

```
-- Q9. Find maximum values of confirmed, deaths, recovered per year
SELECT YEAR(DATE) AS YEAR,
       MAX(CONFIRMED) AS MAX_cases,
       MAX(DEATHS) AS MAX_deaths,
       MAX(RECOVERED) AS MAX_recovered
FROM Virus
GROUP BY YEAR;
```

YEAR	MAX_cases	MAX_deaths	MAX_recovered
2020	823225	3752	1123456
2021	414188	7374	422436

- Used MAX() to find the maximum value from dataset for each year
- Remember this value is from a single row not aggregated
- The values shown have been recorded on single day

```
-- Q10. The total number of case of confirmed, deaths, recovered each month
SELECT YEAR(Date) AS YEAR,
       MONTH(Date) AS MONTH,
       SUM(CONFIRMED) AS Total_cases,
       SUM(DEATHS) AS Total_deaths,
       SUM(RECOVERED) AS Total_recovered
FROM VIRUS
GROUP BY YEAR, MONTH;
```

YEAR	MONTH	Total_cases	Total_deaths	Total_recovered
2020	1	6384	190	143
2020	2	68312	2651	31405
2020	3	769236	41346	133070
2020	4	2336798	191833	792987
2020	5	2744333	144561	1519547
2020	6	3969634	137757	2535417
2020	7	6838092	167613	4693120
2020	8	7694938	179200	6202833
2020	9	8244794	160671	6647749
2020	10	11515841	175484	6782150
2020	11	16595938	262247	9172292
2020	12	19336799	339996	11924903
2021	1	18672205	401893	9164347
2021	2	10492664	298239	6719785
2021	3	13924790	282620	7888013
2021	4	21711021	362387	14205507
2021	5	19121083	366549	19131842
2021	6	5022282	132657	5544438

- The values shown are aggregated sum of values per month
- so we can see till 2020 Sept no of cases been increasing
- Normalized at mid and increases at 2021 MAR.

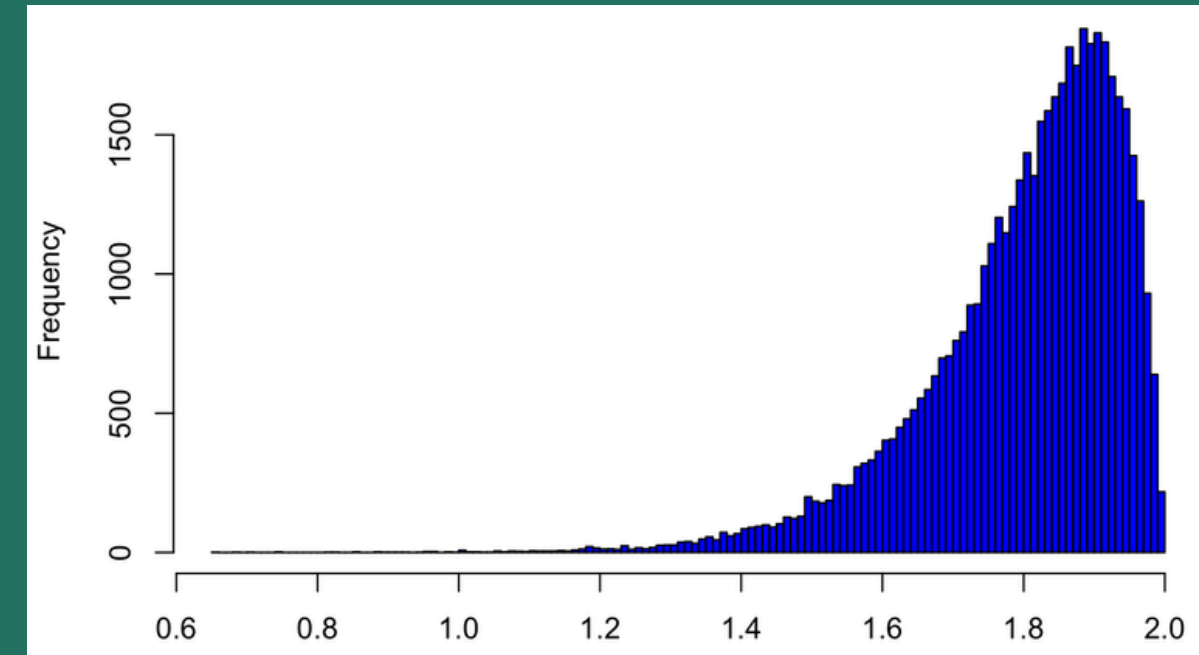
```
-- Q11. Check how corona virus spread out with respect to confirmed case
--      (Eg.: total confirmed cases, their average, variance & STDEV )
SELECT SUM(CONFIRMED) AS Total_cases,
       CAST(avg(CONFIRMED) AS DECIMAL(10, 2)) AS Avg_cases,
       CAST(variance(CONFIRMED) AS DECIMAL(10, 2)) AS Variance,
       CAST(stddev(CONFIRMED) AS DECIMAL(10, 2)) AS standard_deviation
FROM VIRUS;
```

Total_cases	Avg_cases	Variance	standard_deviation
169065144	2156.83	999999999.99	12541.49

- Average cases shows the no of cases confirmed per day
- standard deviation shows this values has outliers
- This data left skewed

```
-- Q12. Check how corona virus spread out with respect to death case per month
--      (Eg.: total confirmed cases, their average, variance & STDEV )
SELECT YEAR(Date) AS YEAR, MONTH(Date) AS MONTH,
       SUM(DEATHS) AS Total_cases,
       CAST(avg(DEATHS) AS DECIMAL(10, 2)) AS Avg_cases,
       CAST(variance(DEATHS) AS DECIMAL(10, 2)) AS Variance,
       CAST(stddev(DEATHS) AS DECIMAL(10, 2)) AS standard_deviation
FROM VIRUS
GROUP BY YEAR, MONTH;
```

YEAR	MONTH	Total_cases	Avg_cases	Variance	standard_deviation
2020	1	190	0.12	4.25	2.06
2020	2	2651	0.59	68.32	8.27
2020	3	41346	8.66	3900.79	62.46
2020	4	191833	41.52	40504.27	201.26
2020	5	144561	30.28	20684.91	143.82
2020	6	137757	29.82	16929.45	130.11
2020	7	167613	35.11	21140.15	145.40
2020	8	179200	37.54	23273.00	152.55
2020	9	160671	34.78	20102.77	141.78
2020	10	175484	36.76	17580.07	132.59
2020	11	262247	56.76	27773.79	166.65
2020	12	339996	71.22	65345.37	255.63
2021	1	401893	84.18	102758.43	320.56
2021	2	298239	69.16	68478.87	261.68
2021	3	282620	59.20	54385.97	233.21
2021	4	362387	78.44	94611.47	307.59
2021	5	366549	76.78	131769.47	363.00
2021	6	132657	66.26	112963.67	336.10



- This chart represents Left skewness
- There are more outliers


```
-- Q13. Check how corona virus spread out with respect to recovered case
--      (Eg.: total confirmed cases, their average, variance & STDEV )
SELECT SUM(RECOVERED) AS Recovered_cases,
       CAST(avg(RECOVERED) AS DECIMAL(10, 2)) AS Avg_cases,
       CAST(variance(RECOVERED) AS DECIMAL(10, 2))AS Variance,
       CAST(stddev(RECOVERED) AS DECIMAL(10, 2)) AS standard_deviation
FROM VIRUS;
```

Recovered_cases	Avg_cases	Variance	standard_deviation
113089548	1442.73	99999999.99	10345.51

```
-- Q14. Find Country having highest number of the Confirmed case
SELECT REGION,
       SUM(CONFIRMED) AS Total_cases
FROM VIRUS
GROUP BY REGION
order by Total_cases desc
LIMIT 1;
```

REGION	Total_cases
US	33461982

- In USA holds 1st place in highest no. of cases recorded
- Total sum of 33,461,982 peoples confirmed covid positive.

```
-- Q15. Find Country having lowest number of the death case
SELECT REGION,
       SUM(CONFIRMED) AS Total_cases
FROM VIRUS
GROUP BY REGION
order by Total_cases
LIMIT 1;
```

REGION	Total_cases
Kiribati	2

- The lowest no. of confirmed cases are from **Kiribati**
- Hold only 2 confirmed cases.

```
-- Q16. Find top 5 countries having highest recovered case
SELECT Region,
       SUM(RECOVERED) AS Recovered_cases
FROM VIRUS
GROUP BY REGION
order by Recovered_cases desc
LIMIT 5;
```

Region	Recovered_cases
India	28089649
Brazil	15400169
US	6303715
Turkey	5202251
Russia	4745756

The Top 5 countries with recovered records:

- 1.India
- 2.Brazil
- 3.US
- 4.Turkey
- 5.Russia

 Jeevarathinam969@gmail.com

 www.linkedin.com/in/jeeva46/

Thank
You