

CORONA VIRUS ANALYSIS

BY
EMMANUEL NNANNA
ORJI

INTRODUCTION

I AM PLEASED TO WELCOME YOU TO THIS PRESENTATION FOCUSING ON MY COMPREHENSIVE ANALYSIS OF THE COVID-19 PANDEMIC. OVER THE RECENT WEEKS,

I HAVE DILIGENTLY COLLECTED AND ANALYZED DATA TO DERIVE MEANINGFUL INSIGHTS INTO THE IMPACT, TRENDS, AND RESPONSES RELATED TO THIS GLOBAL HEALTH CRISIS.

TODAY WE WILL DELVE INTO KEY ASPECTS SUCH AS TOTAL CASES, RECOVERIES, DEATHS, ACTIVE CASES AND MORE, AIMING TO PROVIDE A CLEAR PICTURE OF THE PANDEMIC'S TRAJECTORY AND THE EFFORTS UNDERTAKEN TO MITIGATE IT'S EFFECTS.

WITHOUT ANY ADO, LET'S EMBARK ON THIS INSIGHTFUL JOURNEY TOGETHER.
MANY THANKS FOR JOINING ME!

➡ SQL(STRUCTURED QUERY LANGUAGE) WAS USED FOR ANALYSIS.

ABOUT THE DATASET

IN THIS PROJECT, I AM ANALYZING A COMPREHENSIVE COVID-19 DATASET USING SQL AND ADVANCED DATA ANALYSIS TECHNIQUES. THE DATASET INCLUDES CRUCIAL INFORMATION SUCH AS GEOGRAPHIC LOCATIONS, DATES, CONFIRMED CASES, DEATHS AND RECOVERIES, PROVIDING A RICH SOURCE FOR MY ANALYSIS.

MY TASK INVOLVES DERIVING MEANINGFUL INSIGHTS FROM THIS DATASET, FOCUSING ON KEY ASPECTS SUCH AS TRENDS IN INFECTION RATES, MORTALITY RATES ACROSS DIFFERENT REGIONS, THE EFFECTIVENESS OF INTERVENTIONS OVERTIME AND CORRELATIONS BETWEEN VARIOUS DATA POINTS.

TO GUIDE MY ANALYSIS, I HAVE A REFERENCE FILE CONTAINING 16 DETAILED QUESTIONS THAT SERVE AS MY ROADMAP, COVERING A WIDE RANGE OF ANALYTICAL ASPECTS. THESE QUESTIONS WILL HELP ME UNCOVER PATTERNS, ANOMALIES AND VALUABLE INSIGHTS THAT CAN AID IN UNDERSTANDING THE PANDEMIC'S DYNAMICS AND INFORMING DECISION-MAKING PROCESSES.

PROBLEM STATEMENT

THE COVID-19 PANDEMIC HAS POSED AN UNPRECEDENTED CHALLENGES GLOBALLY, IMPACTING PUBLIC HEALTH, ECONOMIES, AND SOCIETAL NORMS. AS A DATA ANALYST, MY OBJECTIVE IS TO COMPREHENSIVELY ANALYZE THE PANDEMIC'S DATA TO UNDERSTAND IT'S SPREAD, IDENTIFY TRENDS, ASSESS THE EFFECTIVENESS OF INTERVENTIONS AND DERIVE INSIGHTS THAT CAN INFORM DECISION-MAKING PROCESSES. THROUGH THIS ANALYSIS, I AIM TO CONTRIBUTE TO THE COLLECTIVE EFFORT IN COMBATING THE PANDEMIC AND MITIGATING IT'S FAR-REACHING CONSEQUENCES.

PROCESS WORKFLOW



EXPORTED THE DATASET FROM MICROSOFT EXCEL WORKSHEET TO MySQL WORKBENCH.



IMPORTED DATASET INTO MySQL WORKBENCH AS A CSV FILE.



USED SQL TO DELVE INTO THE DATASET BY WRITING OUT THE CODES/QUERIES TO DERIVE MEANINGFUL INSIGHTS AND PROVIDE ACCURATE ANALYSIS.

CHALLENGE I

WRITE A CODE TO
CHECK NULL VALUES.

QUERY

```
SELECT * FROM covid.`corona virus dataset ii`
```

```
WHERE Province IS NULL;
```

```
SELECT * FROM covid. `corona virus dataset ii`
```

```
WHERE Latitude IS NULL;
```

QUERY OUTPUT

Result Grid



Filter Rows:



Export:



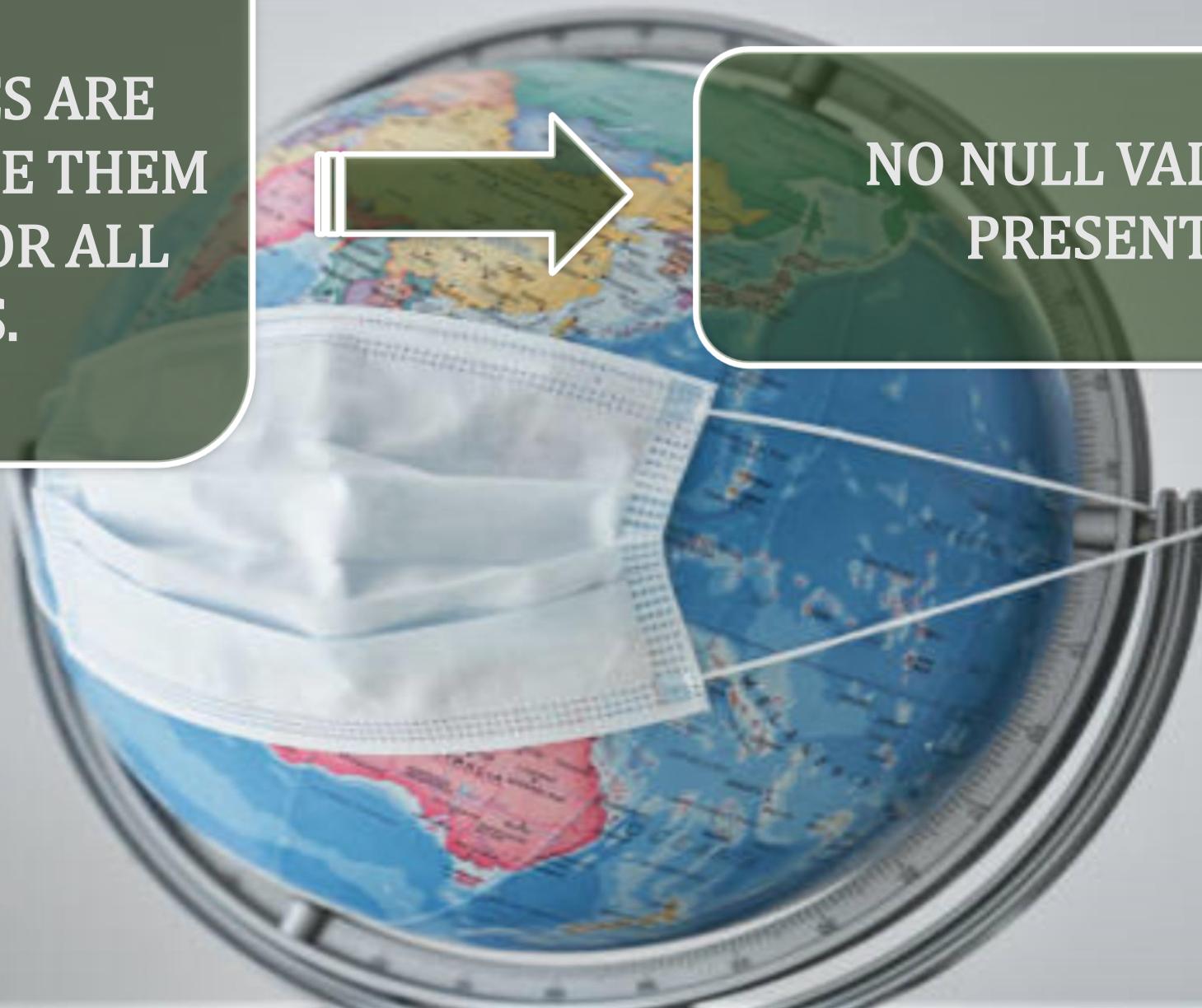
Wrap Cell Content:

Province	Country/Region	Latitude	Longitude	Date	Confirmed	Deaths	Recovered
----------	----------------	----------	-----------	------	-----------	--------	-----------

CHALLENGE II

IF NULL VALUES ARE
PRESENT, UPDATE THEM
WITH ZEROS FOR ALL
COLUMNS.

NO NULL VALUES
PRESENT.



CHALLENGE III

CHECK TOTAL NUMBER
OF ROWS.

QUERY

```
SELECT  
COUNT(*) AS total_rows  
FROM  
'corona virus dataset ii';
```

QUERY OUTPUT

	total_rows
▶	21371

CHALLENGE III

CHECK WHAT IS THE
START_DATE AND
END_DATE.

QUERY

```
SELECT  
MIN(Date) AS start_date, MAX(Date) AS end_date  
FROM  
'corona virus dataset ii'
```

QUERY OUTPUT

	start_date	end_date
▶	1/1/2021	9/9/2020

CHALLENGE V

NUMBER OF MONTHS
PRESENT IN THE
DATASET.

QUERY

```
SELECT  
COUNT(DISTINCT EXTRACT(MONTH FROM STR_TO_DATE(Date, '%d-%m-%Y'))) AS num_months  
FROM  
`corona virus dataset ii`
```

QUERY OUTPUT

Result Grid	
	num_months
▶	12

CHALLENGE VI

FIND MONTHLY
AVERAGE FOR
CONFIRMED, DEATH,
RECOVERED.

QUERY

```
SELECT  
EXTRACT(MONTH FROM Date) AS month,  
AVG(Confirmed) AS avg_confirmed,  
AVG(Recovered) AS avg_recovered,  
AVG(Deaths) AS avg_deaths  
FROM  
`corona virus dataset ii`  
GROUP BY EXTRACT(MONTH FROM Date);
```

QUERY OUTPUT

	Result Grid		Filter Rows:	<input type="text"/>	Export:
	month	avg_confirmed	avg_recovered	avg_deaths	
▶	NULL	1317.2126	1104.4938	33.8395	

CHALLENGE VII

FIND MOST FREQUENT
VALUE FOR
CONFIRMED, DEATHS,
RECOVERED EACH
MONTH.

QUERY

```
SELECT
    EXTRACT(MONTH FROM Date) AS month,
    max(Confirmed) AS most_frequent_confirmed,
    max(Recovered) AS most_frequent_recovered,
    max(Deaths) AS most_frequent_deaths
FROM
    `corona virus dataset ii`
GROUP BY
    EXTRACT(MONTH FROM Date);
```

QUERY OUTPUT

	month	most_frequent_confirmed	most_frequent_recovered	most_frequent_deaths
▶	NULL	100158	388340	4249

CHALLENGE VIII

FIND MINIMUM VALUES
FOR CONFIRMED,
DEATHS, RECOVERED
PER YEAR.

QUERY

```
SELECT
    EXTRACT(YEAR FROM Date) AS year,
    MIN(Confirmed) AS minimum_confirmed,
    MIN(Recovered) AS minimum_recovered,
    MIN(Deaths) AS minimum_deaths
FROM
    `corona virus dataset ii`
GROUP BY
    EXTRACT(YEAR FROM Date);
```

QUERY OUTPUT

	year	minimum_confirmed	minimum_recovered	minimum_deaths
▶	NULL	0	0	0

CHALLENGE IX

FIND MAXIMUM VALUES
OF CONFIRMED,
DEATHS, RECOVERED
PER YEAR.

QUERY

```
SELECT
    EXTRACT(YEAR FROM Date) AS year,
    MAX(Confirmed) AS maximum_confirmed,
    MAXRecovered) AS maximum_recovered,
    MAX(Deaths) AS maximum_deaths
FROM
    `corona virus dataset ii`
GROUP BY
    EXTRACT(YEAR FROM Date);
```

QUERY OUTPUT

Result Grid		Filter Rows:	Export:	Wrap
	year	maximum_confirmed	maximum_recovered	maximum_deaths
▶	NULL	100158	388340	4249

CHALLENGE X

THE TOTAL NUMBER
OF CONFIRMED,
DEATHS, RECOVERED
EACH MONTH.

QUERY

```
SELECT
    EXTRACT(MONTH FROM Date) AS month,
    SUM(Confirmed) AS total_confirmed,
    SUM(Recovered) AS total_recovered,
    SUM(Deaths) AS total_deaths
FROM
    `corona virus dataset ii`
GROUP BY
    EXTRACT(MONTH FROM Date)
ORDER BY
    EXTRACT(MONTH FROM Date);
```

QUERY OUTPUT

	month	total_confirmed	total_recovered	total_deaths
▶	NULL	28150150	23604138	723184

CHALLENGE XI

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO CONFIRMED CASES.

-- EG. : TOTAL CONFIRMED CASES, THEIR AVERAGE, VARIANCE & STDEV.

QUERY

```
-- Calculate total confirmed cases  
SELECT SUM(Confirmed) AS total_confirmed_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate average confirmed cases  
SELECT AVG(Confirmed) AS average_confirmed_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate variance of confirmed cases  
SELECT VARIANCE(Confirmed) AS variance_confirmed_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate standard deviation of confirmed cases  
SELECT STDDEV(Confirmed) AS stdev_confirmed_cases  
FROM `corona virus dataset ii`;
```

QUERY OUTPUT

Result Grid	Filter Row
stdev_confirmed_cases	6865.9956

CHALLENGE XII

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO DEATH CASES PER MONTH.

-- EG. : (TOTAL CONFIRMED CASES, THEIR AVERAGE, VARIANCE & STDEV).

QUERY

```
-- Calculate total deaths cases  
SELECT SUM(Deaths) AS total_deaths_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate average deaths cases  
SELECT AVG(Deaths) AS average_deaths_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate variance of deaths cases  
SELECT VARIANCE(Deaths) AS variance_deaths_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate standard deviation of deaths cases  
SELECT STDDEV(Deaths) AS stdev_deaths_cases  
FROM `corona virus dataset ii`;
```

QUERY OUTPUT

	Result Grid	Filter
►	stdev_deaths_cases 202.6720	

CHALLENGE XIII

CHECK HOW CORONA VIRUS SPREAD OUT WITH RESPECT TO RECOVERED CASES.

-- EG. : TOTAL RECOVERED CASES, THEIR AVERAGE, VARIANCE & STDEV.

QUERY

```
-- Calculate total recovered cases  
SELECT SUM(Recovered) AS total_recovered_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate average recovered cases  
SELECT AVG(Recovered) AS average_recovered_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate variance of recovered cases  
SELECT VARIANCE(Recovered) AS variance_recovered_cases  
FROM `corona virus dataset ii`;  
  
-- Calculate standard deviation of recovered cases  
SELECT STDDEV(Recovered) AS stdev_recovered_cases  
FROM `corona virus dataset ii`;
```

QUERY OUTPUT

Result Grid	Filter Rows:
stdev_recovered_cases	7142.8776

CHALLENGE XIV

FIND COUNTRY HAVING
THE HIGHEST NUMBER
OF CONFIRMED CASE.

QUERY

```
SELECT
`Country/Region`, Confirmed
FROM
`corona virus dataset ii`
WHERE
Confirmed = (SELECT
MAX(Confirmed)
FROM
`corona virus dataset ii`);
```

QUERY OUTPUT

	Result Grid			Filter Rows:
	Country/Region	Confirmed		
▶	Brazil	100158		

CHALLENGE XV

FIND COUNTRY HAVING
LOWEST NUMBER OF
DEATH CASE.

QUERY

```
SELECT
`Country/Region`, Confirmed
FROM
`corona virus dataset ii`
WHERE
Confirmed = (SELECT
MIN(Confirmed)
FROM
`corona virus dataset ii`);
```

QUERY OUTPUT

	Result Grid	Filter Rows:
	Country/Region	Confirmed
▶	Afghanistan	0
	Algeria	0

CHALLENGE XVI

FIND TOP 5 COUNTRIES HAVING HIGHEST RECOVERED CASE.

QUERY

```
SELECT
`Country/Region`, SUM(Recovered) AS total_recovered_cases
FROM
`corona virus dataset ii`
GROUP BY
`Country/Region`
ORDER BY
total_recovered_cases DESC
LIMIT
5;
```

QUERY OUTPUT

Result Grid			Filter Rows:
	Country/Region	total_recovered_cases	
▶	Brazil	15400169	
	Argentina	3721350	
	Chile	1399187	
	Bangladesh	766266	
	Austria	634624	



THANK YOU!!