

The background features three vertical stripes on the left side in shades of pink, blue, and beige. On the right side, there are two rectangular areas filled with a grid of small, light pink dots.

# **CORONA ANALYSIS**

**Presented By : Ankita singh vats**

```
CREATE TABLE covid (  
    Province VARCHAR(255),  
    `Country/Region` VARCHAR(255),  
    Latitude FLOAT,  
    Longitude FLOAT,  
    Date VARCHAR(10), -- Temporary storage for the date as string  
    Confirmed INT,  
    Deaths INT,  
    Recovered INT  
);  
  
ALTER TABLE corona  
ADD COLUMN year INT,  
ADD COLUMN month INT;
```

# CREATED A TABLE

```
-- Step 3: Update the new columns with extracted 'year' and 'month' values  
UPDATE corona  
SET year = EXTRACT(YEAR FROM Date_converted),  
    month = EXTRACT(MONTH FROM Date_converted);  
select * from corona
```

-- 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_converted IS NULL
      OR Confirmed IS NULL
      OR Deaths IS NULL
      OR Recovered IS NULL;
```

	Province	Country/Region	Latitude	Longitude	Confirmed	Deaths	Recovered	Date_converted	year	month

```
/*Q2. If NULL values are present, update them with zeros for all columns. */
UPDATE corona
SET Province = COALESCE(Province, 'Unknown'),
    `Country/Region` = COALESCE(`Country/Region`, 'Unknown'),
    Latitude = COALESCE(Latitude, 0),
    Longitude = COALESCE(Longitude, 0),
    Date_converted = COALESCE(Date_converted, '1900-01-01'),
    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_converted IS NULL
      OR Confirmed IS NULL
      OR Deaths IS NULL
      OR Recovered IS NULL;
```

**THERE IS NO NULL VALUE IN THE DATASET**

```
-- Q3. check total number of rows  
SELECT COUNT(*) FROM corona;
```

	COUNT(*)
▶	78386

```
-- Q4. Check what is start_date and end_date  
SELECT MIN(date_converted) AS start_date, MAX(date_converted) AS end_date FROM corona;
```

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

```
-- Q5. Number of month present in dataset  
SELECT COUNT(DISTINCT(month)) AS num_months FROM corona;
```

	num_months
▶	12

```
-- Q6. Find monthly average for confirmed, deaths, recovered
SELECT month,
       AVG(Confirmed) AS avg_confirmed,
       AVG(Deaths) AS avg_deaths,
       AVG(Recovered) AS avg_recovered
FROM (
    SELECT DATE_FORMAT(Date_converted, '%Y-%m') AS month,
           Confirmed,
           Deaths,
           Recovered
    FROM corona
) AS subquery
GROUP BY month;
```

	month	avg_confirmed	avg_deaths	avg_recovered
►	2020-01	4.1455	0.1234	0.0929
	2020-02	15.2960	0.5936	7.0320
	2020-03	161.1303	8.6607	27.8739
	2020-04	505.8004	41.5223	171.6422
	2020-05	574.8498	30.2809	318.2964
	2020-06	859.2281	29.8175	548.7916
	2020-07	1432.3611	35.1096	983.0582
	2020-08	1611.8429	37.5367	1299.2947
	2020-09	1784.5874	34.7773	1438.9067
	2020-10	2412.1996	36.7583	1420.6431
	2020-11	3592.1944	56.7634	1985.3446
	2020-12	4050.4397	71.2183	2497.8850
	2021-01	3911.2285	84.1837	1919.6370
	2021-02	2433.3636	69.1649	1558.3917
	2021-03	2916.7972	59.1998	1652.2859
	2021-04	4699.3552	78.4387	3074.7851
	2021-05	4005.2541	76.7803	4007.5078

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

```
SELECT month,  
       year,  
       Confirmed,  
       Deaths,  
       Recovered  
FROM (  
    SELECT month,  
           year,  
           Confirmed,  
           Deaths,  
           Recovered,  
           ROW_NUMBER() OVER (PARTITION BY month, year ORDER BY COUNT(*) DESC) AS rn  
    FROM corona  
    GROUP BY month, year, Confirmed, Deaths, Recovered  
) AS subquery  
WHERE rn = 1;
```

	month	year	Confirmed	Deaths	Recovered
►	1	2020	0	0	0
	1	2021	0	0	0
	2	2020	0	0	0
	2	2021	0	0	0
	3	2020	0	0	0
	3	2021	0	0	0
	4	2020	0	0	0
	4	2021	0	0	0
	5	2020	0	0	0
	5	2021	0	0	0
	6	2020	0	0	0
	6	2021	0	0	0
	7	2020	0	0	0
	8	2020	0	0	0
	9	2020	0	0	0
	10	2020	0	0	0
	11	2020	0	0	0
	12	2020	0	0	0

```
-- Q8. Find minimum values for confirmed, deaths, recovered per year
SELECT  year,
        MIN(Confirmed) AS min_confirmed,
        MIN(Deaths) AS min_deaths,
        MIN(Recovered) AS min_recovered
FROM corona
GROUP BY year;
```

	year	min_confirmed	min_deaths	min_recovered
▶	2020	0	0	0
	2021	0	0	0

```
-- Q9. Find maximum values of confirmed, deaths, recovered per year
SELECT  year,
        Max(Confirmed) AS max_confirmed,
        Max(Deaths) AS max_deaths,
        Max(Recovered) AS max_recovered
FROM corona
GROUP BY year;
```

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



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

```
SELECT  month,  
        sum(Confirmed) AS total_confirmed,  
        sum(Deaths) AS total_deaths,  
        sum(Recovered) AS total_recovered  
FROM corona  
GROUP BY month;
```

	month	total_confirmed	total_deaths	total_recovered
►	1	18678589	402083	9164490
	2	10560976	300890	6751190
	3	14694026	323966	8021083
	4	24047819	554220	14998494
	5	21865416	511110	20651389
	6	8991916	270414	8079855
	7	6838092	167613	4693120
	8	7694938	179200	6202833
	9	8244794	160671	6647749
	10	11515841	175484	6782150
	11	16595938	262247	9172292
	12	19336799	339996	11924903

```
-- 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_confirmed,
       AVG(Confirmed) AS avg_confirmed,
       VARIANCE(Confirmed) AS var_confirmed,
       STDDEV(Confirmed) AS stdev_confirmed
FROM corona;
```

	total_confirmed	avg_confirmed	var_confirmed	stdev_confirmed
▶	169065144	2156.8283	157288925.07796532	12541.488152446875

```
-- Q12. Check how corona virus spread out with respect to death case per month
--      (Eg.: total confirmed cases, their average, variance & STDEV )
SELECT month,
        SUM(Deaths) AS total_deaths,
        AVG(Deaths) AS avg_deaths,
        VARIANCE(Deaths) AS var_deaths,
        STDDEV(Deaths) AS stdev_deaths
FROM corona
GROUP BY month;
```

	month	total_deaths	avg_deaths	var_deaths	stdev_deaths
►	1	402083	63.6812	78999.5307609659	281.0685517110833
	2	300890	34.2777	34848.64785490521	186.67792546229245
	3	323966	33.9302	29781.93292256146	172.57442719754704
	4	554220	59.9805	67898.57559453539	260.5735512183372
	5	511110	53.5306	76767.73838185583	277.06991605343194
	6	270414	40.8357	46243.20314719306	215.04232873365433
	7	167613	35.1096	21140.154944373826	145.39654378414167
	8	179200	37.5367	23272.99645685882	152.55489653517785
	9	160671	34.7773	20102.7692237308	141.78423475030925
	10	175484	36.7583	17580.07101972725	132.589860169348
	11	262247	56.7634	27773.793596962234	166.6547136955995
	12	339996	71.2183	65345.36920134891	255.6274030720277

```
-- Q13. Check how corona virus spread out with respect to recovered case
--      (Eg.: total confirmed cases, their average, variance & STDEV )
SELECT SUM(Recovered) AS total_recovered,
       AVG(Recovered) AS avg_recovered,
       VARIANCE(Recovered) AS var_recovered,
       STDDEV(Recovered) AS stdev_recovered
FROM corona;
```

	total_recovered	avg_recovered	var_recovered	stdev_recovered
►	113089548	1442.7264	107029523.26229636	10345.507395110999

```
-- Q14. Find Country having highest number of the Confirmed case
SELECT `Country/Region`, SUM(Confirmed) AS total_confirmed
FROM corona
GROUP BY `Country/Region`
ORDER BY total_confirmed DESC
LIMIT 1;
```

	Country/Region	total_confirmed
▶	US	33461982

```
-- Q15. Find Country having lowest number of the death case
SELECT `Country/Region`, SUM(Deaths) AS total_deaths
FROM corona
GROUP BY `Country/Region`
ORDER BY total_deaths ASC
LIMIT 1;
```

	Country/Region	total_deaths
▶	Dominica	0

```
-- Q16. Find top 5 countries having highest recovered case
SELECT `Country/Region`, SUM(Recovered) AS total_recovered
FROM corona
GROUP BY `Country/Region`
ORDER BY total_recovered DESC
LIMIT 5;
```

	Country/Region	total_recovered
▶	India	28089649
	Brazil	15400169
	US	6303715
	Turkey	5202251
	Russia	4745756

