

CORONA VIRUS ANALYSIS

**PRESENTING BY:
AMBIKA BAI**

CORONA VIRUS ANALYSIS

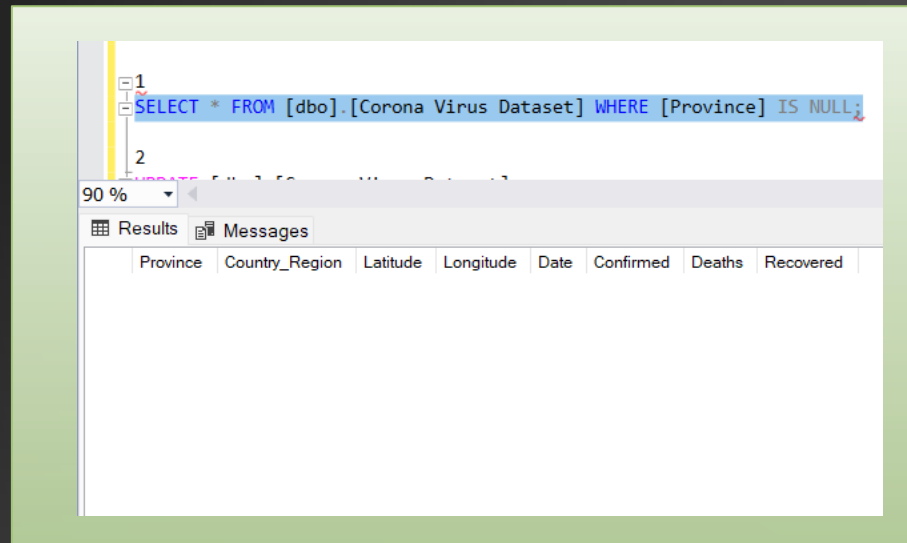
Overview:

The CORONA VIRUS pandemic has had 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.

CORONA VIRUS ANALYSIS

1. Write code to check null values.

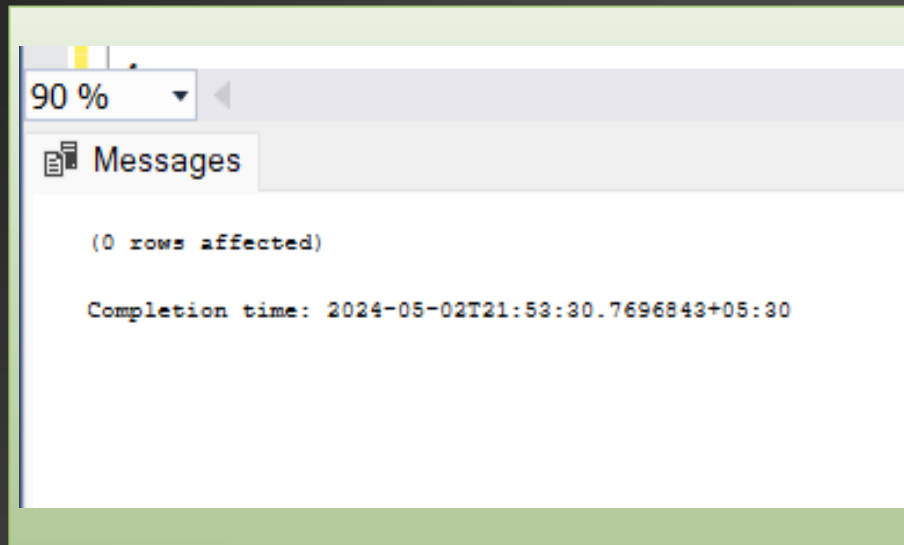
```
SELECT * FROM [dbo].[Corona Virus Dataset] WHERE [Province] IS NULL;
```



> There are no null values in given database

2. If null values are present, update them to zeros for all columns

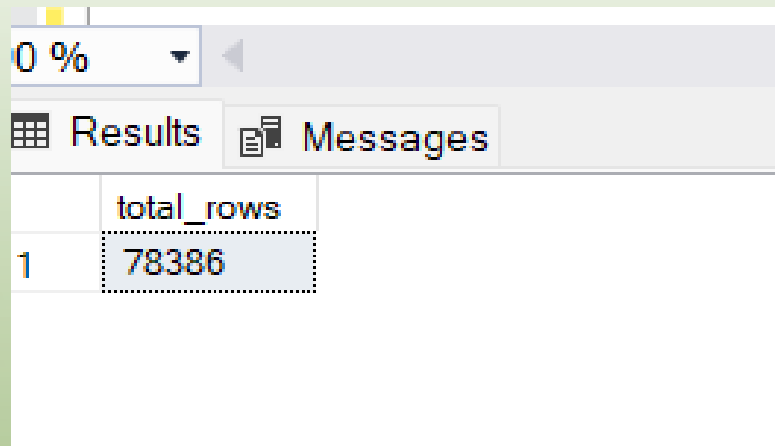
```
UPDATE [dbo].[Corona Virus Dataset]
SET Confirmed = ISNULL(Confirmed, 0),
    Deaths = ISNULL(Deaths, 0)
WHERE Confirmed IS NULL OR Deaths IS
NULL;
```



> There are no null values in given database

3. Check total number of rows.

```
SELECT COUNT(*) AS total_rows FROM [dbo].[Corona Virus Dataset];
```

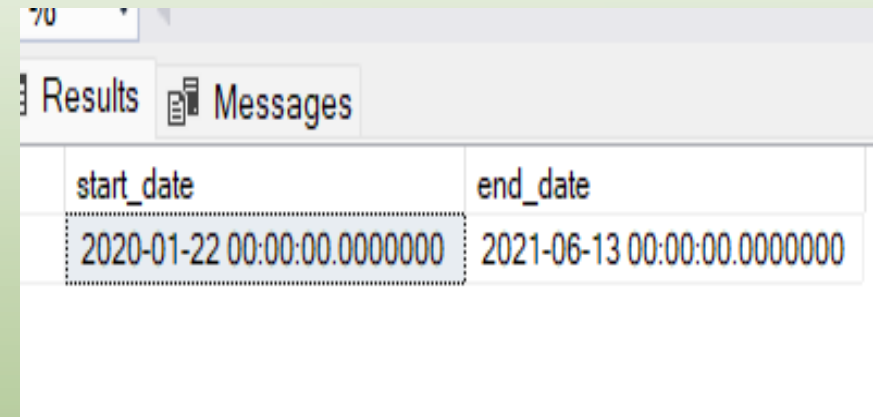


The screenshot shows a SQL Server query results window. At the top, there is a dropdown menu set to '0 %' and a navigation bar with 'Results' and 'Messages' tabs. Below the tabs, a table displays the query results. The table has two columns: 'total_rows' and an unnamed column. The first row shows the value '78386' under 'total_rows' and '1' in the unnamed column.

	total_rows
1	78386

4. Check what is start_date and end_date.

```
SELECT  
    MIN(Date) AS start_date,  
    MAX(Date) AS end_date  
FROM [dbo].[Corona Virus Dataset]
```

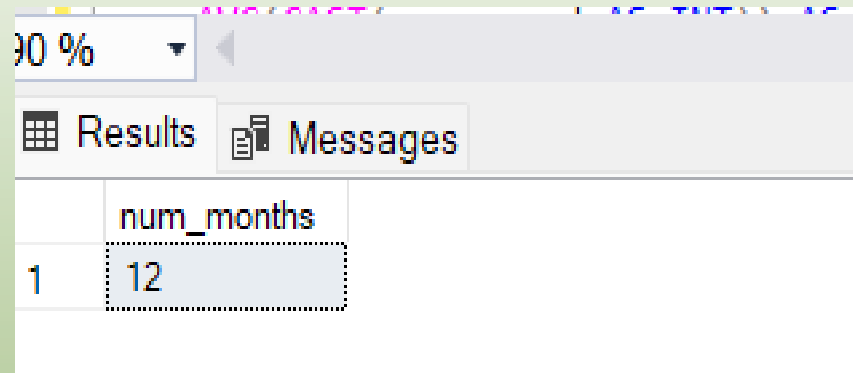


The screenshot shows a SQL Server query results window. The 'Results' tab is active, displaying a table with two columns: 'start_date' and 'end_date'. The 'start_date' column contains the value '2020-01-22 00:00:00.0000000' and the 'end_date' column contains the value '2021-06-13 00:00:00.0000000'. The 'Messages' tab is also visible but empty.

start_date	end_date
2020-01-22 00:00:00.0000000	2021-06-13 00:00:00.0000000

5.Number of month present in dataset.

```
SELECT COUNT(DISTINCT MONTH(Date)) AS num_months  
FROM [dbo].[Corona Virus Dataset];
```

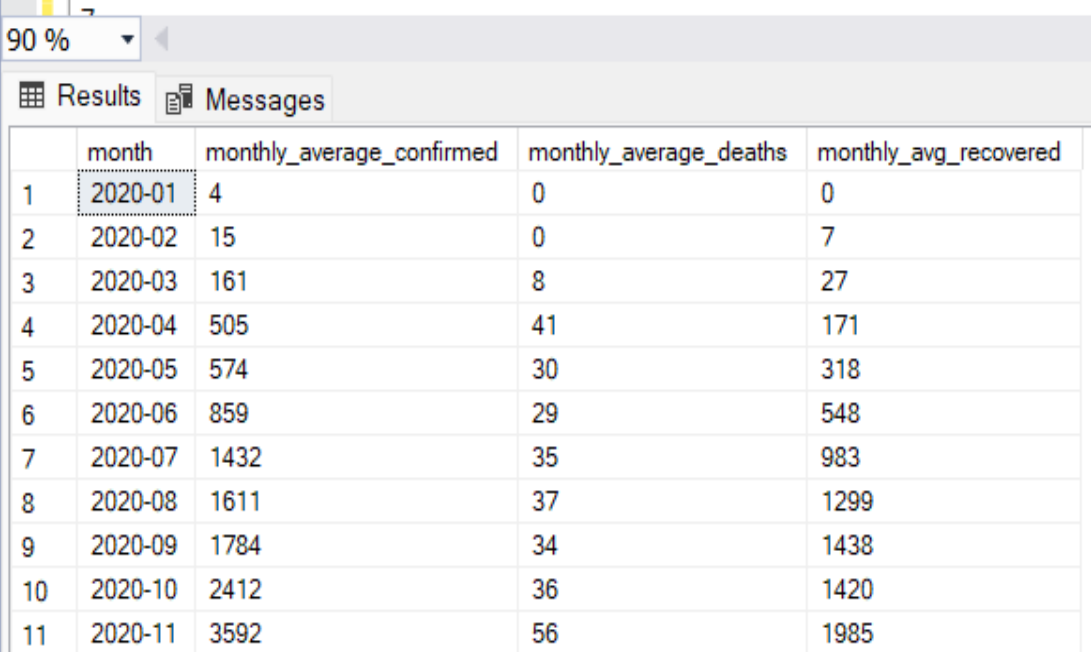


The screenshot shows a SQL query execution interface. At the top, there is a dropdown menu set to '100 %'. Below it are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with one column named 'num_months' and one row containing the value '12'.

	num_months
1	12

6. Find monthly average for confirmed,deaths,recovered.

```
SELECT
    FORMAT(Date, 'yyyy-MM') AS month,
    AVG(CAST(Confirmed AS INT)) AS
monthly_average_confirmed,
    AVG(CAST(deaths AS INT)) AS
monthly_average_deaths,
    AVG(CAST(recovered AS INT)) AS
monthly_avg_recovered
FROM
    [dbo].[Corona Virus Dataset]
GROUP BY
    FORMAT(Date, 'yyyy-MM')
ORDER BY
    month;
```



	month	monthly_average_confirmed	monthly_average_deaths	monthly_avg_recovered
1	2020-01	4	0	0
2	2020-02	15	0	7
3	2020-03	161	8	27
4	2020-04	505	41	171
5	2020-05	574	30	318
6	2020-06	859	29	548
7	2020-07	1432	35	983
8	2020-08	1611	37	1299
9	2020-09	1784	34	1438
10	2020-10	2412	36	1420
11	2020-11	3592	56	1985

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

```
WITH MonthlyCounts AS (  
  SELECT  
    YEAR(Date) AS year,  
    MONTH(Date) AS month,  
    confirmed,  
    deaths,  
    recovered,  
    ROW_NUMBER() OVER (PARTITION BY YEAR(Date), MONTH(Date), confirmed, deaths, recovered ORDER BY COUNT(*))  
  ) AS rn  
FROM [dbo].[Corona Virus Dataset]  
GROUP BY YEAR(Date), MONTH(Date), confirmed, deaths, recovered  
)  
SELECT  
  year,  
  month,  
  confirmed,  
  deaths,  
  recovered  
FROM MonthlyCounts  
WHERE rn = 1;
```

output 7

90 %

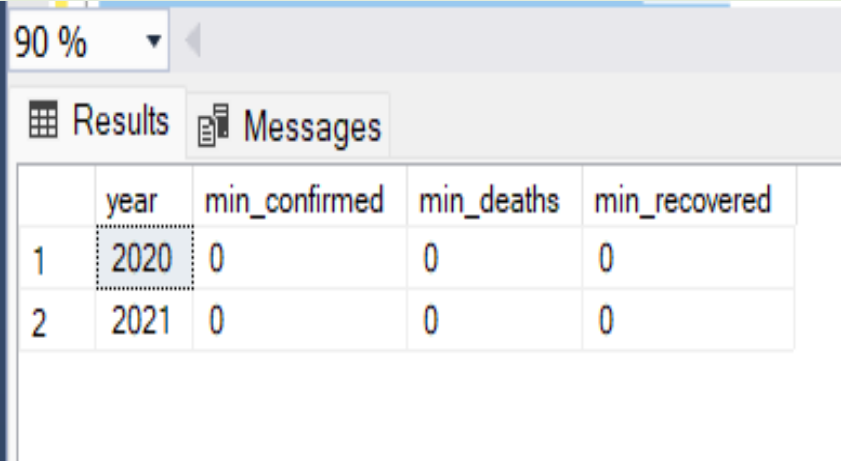
Results Messages

	year	month	confirmed	deaths	recovered
1	2020	1	0	0	0
2	2020	1	0	0	1
3	2020	1	0	0	2
4	2020	1	0	0	8
5	2020	1	0	1	0
6	2020	1	1	0	0
7	2020	1	1	0	1
8	2020	1	10	0	0
9	2020	1	105	7	3
10	2020	1	11	0	0
11	2020	1	11	0	2
12	2020	1	12	0	0
13	2020	1	12	1	0
14	2020	1	13	0	0
15	2020	1	13	0	1
16	2020	1	13	0	2
17	2020	1	1240	27	2

✓ Query executed successfully.

8. Find minimum values for confirmed, deaths, recovered per month.

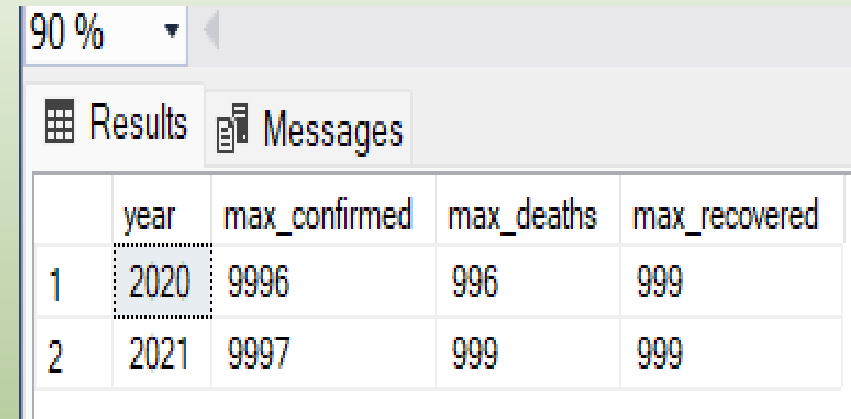
```
SELECT
    YEAR(Date) AS year,
    MIN(confirmed) AS min_confirmed,
    MIN(deaths) AS min_deaths,
    MIN(recovered) AS min_recovered
FROM
    [dbo].[Corona Virus Dataset]
GROUP BY
    YEAR(Date)
ORDER BY
    YEAR(Date) ASC;
```



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

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

```
SELECT  
    YEAR(Date) AS year,  
    MAX(confirmed) AS max_confirmed,  
    MAX(deaths) AS max_deaths,  
    MAX(recovered) AS max_recovered  
FROM  
    [dbo].[Corona Virus Dataset]  
GROUP BY  
    YEAR(Date)  
ORDER BY  
    YEAR(Date) ASC;
```



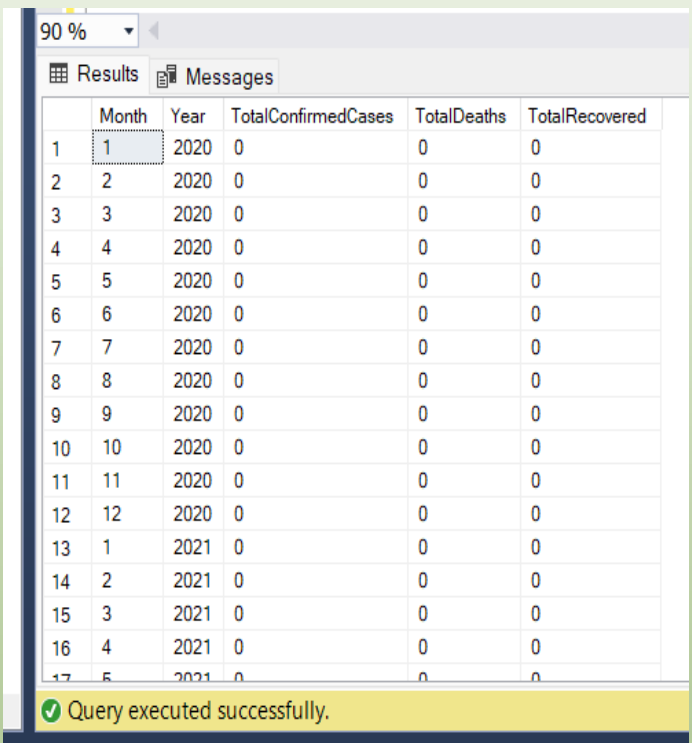
90 %

Results Messages

	year	max_confirmed	max_deaths	max_recovered
1	2020	9996	996	999
2	2021	9997	999	999

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

```
SELECT
    MONTH(Date) AS Month,
    YEAR(Date) AS Year,
    SUM(CASE WHEN Confirmed = 'Confirmed' THEN 1 ELSE 0 END) AS
TotalConfirmedCases,
    SUM(CASE WHEN Deaths = 'Deaths' THEN 1 ELSE 0 END) AS TotalDeaths,
    SUM(CASE WHEN Recovered = 'Recovered' THEN 1 ELSE 0 END) AS
TotalRecovered
FROM [dbo].[Corona Virus Dataset]
GROUP BY YEAR(Date), MONTH(Date)
ORDER BY YEAR(Date), MONTH(Date);
```

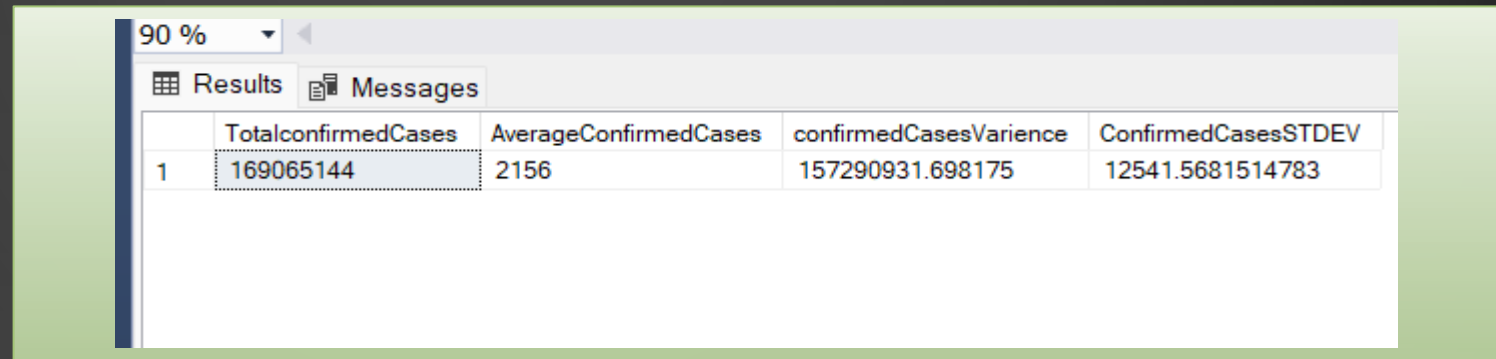


	Month	Year	TotalConfirmedCases	TotalDeaths	TotalRecovered
1	1	2020	0	0	0
2	2	2020	0	0	0
3	3	2020	0	0	0
4	4	2020	0	0	0
5	5	2020	0	0	0
6	6	2020	0	0	0
7	7	2020	0	0	0
8	8	2020	0	0	0
9	9	2020	0	0	0
10	10	2020	0	0	0
11	11	2020	0	0	0
12	12	2020	0	0	0
13	1	2021	0	0	0
14	2	2021	0	0	0
15	3	2021	0	0	0
16	4	2021	0	0	0
17	5	2021	0	0	0

Query executed successfully.

11. Check how corona virus spread out with respect to confirmed case (Eg.: total confirmed cases, their average, variance & STDEV).

```
SELECT
SUM(CAST(Confirmed AS INT)) AS TotalconfirmedCases,
AVG(CAST(Confirmed AS INT)) AS AverageConfirmedCases,
VAR(CAST(Confirmed AS INT)) AS confirmedCasesVariance,
STDEV(CAST(Confirmed AS INT)) AS ConfirmedCasesSTDEV
FROM [dbo].[Corona Virus Dataset]
```



The screenshot shows a SQL Server query results window. At the top, there is a zoom level of 90% and two tabs: 'Results' (selected) and 'Messages'. Below the tabs is a table with four columns: 'TotalconfirmedCases', 'AverageConfirmedCases', 'confirmedCasesVariance', and 'ConfirmedCasesSTDEV'. The first row of data shows the following values: 169065144, 2156, 157290931.698175, and 12541.5681514783.

	TotalconfirmedCases	AverageConfirmedCases	confirmedCasesVariance	ConfirmedCasesSTDEV
1	169065144	2156	157290931.698175	12541.5681514783

12. 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(CAST(Deaths AS INT)) AS TotalDeathCases,
    AVG(CAST(Deaths AS INT)) AS
AverageDeathCases,
    VAR( CAST(Deaths AS INT)) AS
DeathCasesVariance,
    STDEV(CAST(Deaths AS INT)) AS
DeathCasesSTDEV
FROM [dbo].[Corona Virus Dataset]
GROUP BY YEAR(Date), MONTH(Date)
ORDER BY Year, Month;
```

	Year	Month	TotalDeathCases	AverageDeathCases	DeathCasesVariance	DeathCasesSTDEV
1	2020	1	190	0	4.24857598541809	2.06120740960683
2	2020	2	2651	0	68.337150469718	8.26662872455985
3	2020	3	41346	8	3901.60952698687	62.4628651839385
4	2020	4	191833	41	40513.0371733448	201.278506486273
5	2020	5	144561	30	20689.2454049367	143.837566042174
6	2020	6	137757	29	16933.1108854449	130.127287243856
7	2020	7	167613	35	21144.5840570796	145.41177413497
8	2020	8	179200	37	23277.8724251087	152.570876726552
9	2020	9	160671	34	20107.1214145132	141.799581855918
10	2020	10	175484	36	17583.7542527085	132.60374901453

13 Check how corona virus spread out with respect to recovered case (Eg.: total confirmed cases, their average, variance & STDEV).

```
SELECT
YEAR(Date) AS Year,
MONTH(Date) AS Month,
SUM(CAST(Recovered AS INT)) AS TotalRecoveredCases,
AVG(CAST(Recovered AS INT)) AS AverageRecoveredCases,
VAR( CAST(Recovered AS INT)) AS RecoveredCasesVariance,
STDEV(CAST(Recovered AS INT)) AS RecoveredCasesSTDEV
FROM [dbo].[Corona Virus Dataset]
GROUP BY YEAR(Date), MONTH(Date)
ORDER BY Year, Month;
```

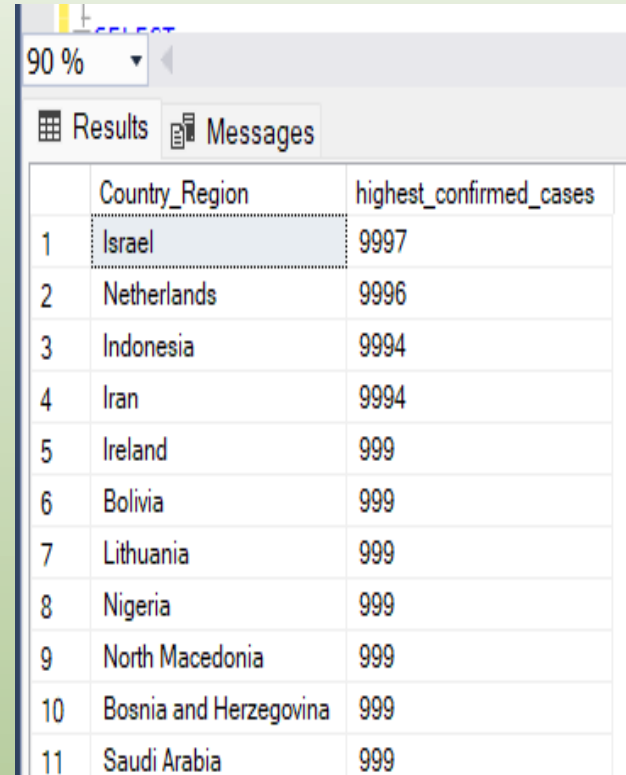
90 %

Results Messages

	Year	Month	TotalRecoveredCases	AverageRecoveredCases	RecoveredCasesVariance	RecoveredCasesSTDEV
1	2020	1	143	0	2.63529657477026	1.62335965662889
2	2020	2	31405	7	12449.4495904104	111.577101550499
3	2020	3	133070	27	40121.5939844912	200.303754294549
4	2020	4	792987	171	770059.711532687	877.530461883054
5	2020	5	1519547	318	1978620.87525624	1406.63459194499
6	2020	6	2535417	548	6531586.25639116	2555.69682403668
7	2020	7	4693120	983	24849082.9398306	4984.88544901792
8	2020	8	6202833	1299	40178838.3767708	6338.67796758684
9	2020	9	6647749	1438	57035911.8793661	7552.21238309451
10	2020	10	6782150	1420	73747150.1663075	8587.61609332342

14. Find Country having highest number of the Confirmed case.

```
SELECT
    [Country_Region],
    MAX(confirmed) AS highest_confirmed_cases
FROM
    [dbo].[Corona Virus Dataset]
GROUP BY
    [Country_Region]
ORDER BY
    highest_confirmed_cases DESC;
```



The screenshot shows a SQL query result in a table with two columns: 'Country_Region' and 'highest_confirmed_cases'. The results are ordered in descending order of confirmed cases. Israel is at the top with 9997 cases, followed by the Netherlands with 9996 cases. The table lists 11 countries in total.

	Country_Region	highest_confirmed_cases
1	Israel	9997
2	Netherlands	9996
3	Indonesia	9994
4	Iran	9994
5	Ireland	999
6	Bolivia	999
7	Lithuania	999
8	Nigeria	999
9	North Macedonia	999
10	Bosnia and Herzegovina	999
11	Saudi Arabia	999

15. Find the country having lowest number of death cases.

```
SELECT
    [Country_Region],
    MIN(deaths) AS lowest_death_cases
FROM
    [dbo].[Corona Virus Dataset]
GROUP BY
    [Country_Region]
ORDER BY
    lowest_death_cases ASC;
```

90 %

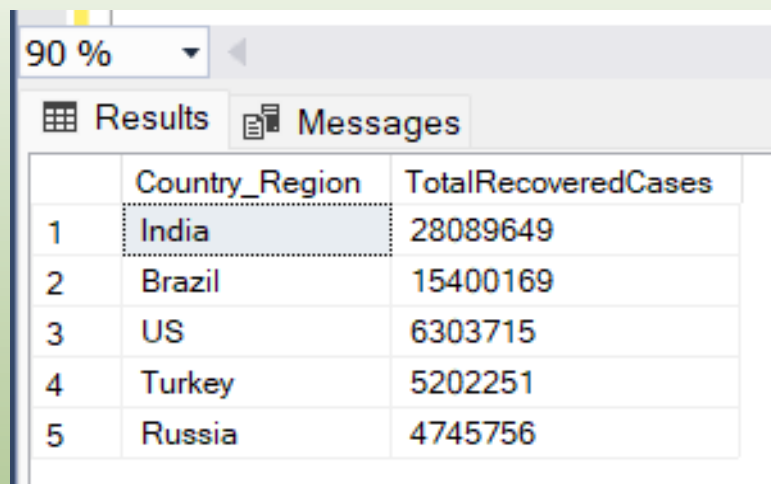
Results Messages

	Country_Region	lowest_death_cases
1	Finland	0
2	Gambia	0
3	West Bank and Gaza	0
4	Vietnam	0
5	New Zealand	0
6	Uganda	0
7	Egypt	0
8	Italy	0
9	Brazil	0
10	Netherlands	0
11	Namibia	0
12	Jordan	0
13	Korea, South	0
14	Belarus	0
15	Madagascar	0
16	Marshall Islands	0
17	Belarus	0

✓ Query executed successfully.

16. Find the top 5 countries having highest recovered case.

```
SELECT TOP 5 Country_Region, SUM(CAST(Recovered AS INT)) AS  
TotalRecoveredCases  
FROM [dbo].[Corona Virus Dataset]  
GROUP BY Country_Region  
ORDER BY TotalRecoveredCases DESC
```



A screenshot of a SQL Server query results window. The window has a title bar with a zoom level of 90%. Below the title bar are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with two columns: 'Country_Region' and 'TotalRecoveredCases'. The table contains five rows of data, ordered from highest to lowest total recovered cases. The first row is India with 2,808,964 cases, followed by Brazil (1,540,016), US (630,371), Turkey (520,225), and Russia (474,576).

	Country_Region	TotalRecoveredCases
1	India	28089649
2	Brazil	15400169
3	US	6303715
4	Turkey	5202251
5	Russia	4745756