MentorNess Task 2

Corona Virus Analysis With SQL

Presented By - Harish R(Data Analyst Intern, Mentorness)

OVERVIEW

This project involves analyzing a COVID-19 dataset using SQL and data analysis skills. The task is to derive insights from the dataset containing information such as geographic location, dates, confirmed cases, deaths, and recoveries. Also i used SQL to querry and analyse the data as well as answer for specific questions about the dataset.

Dataset Details

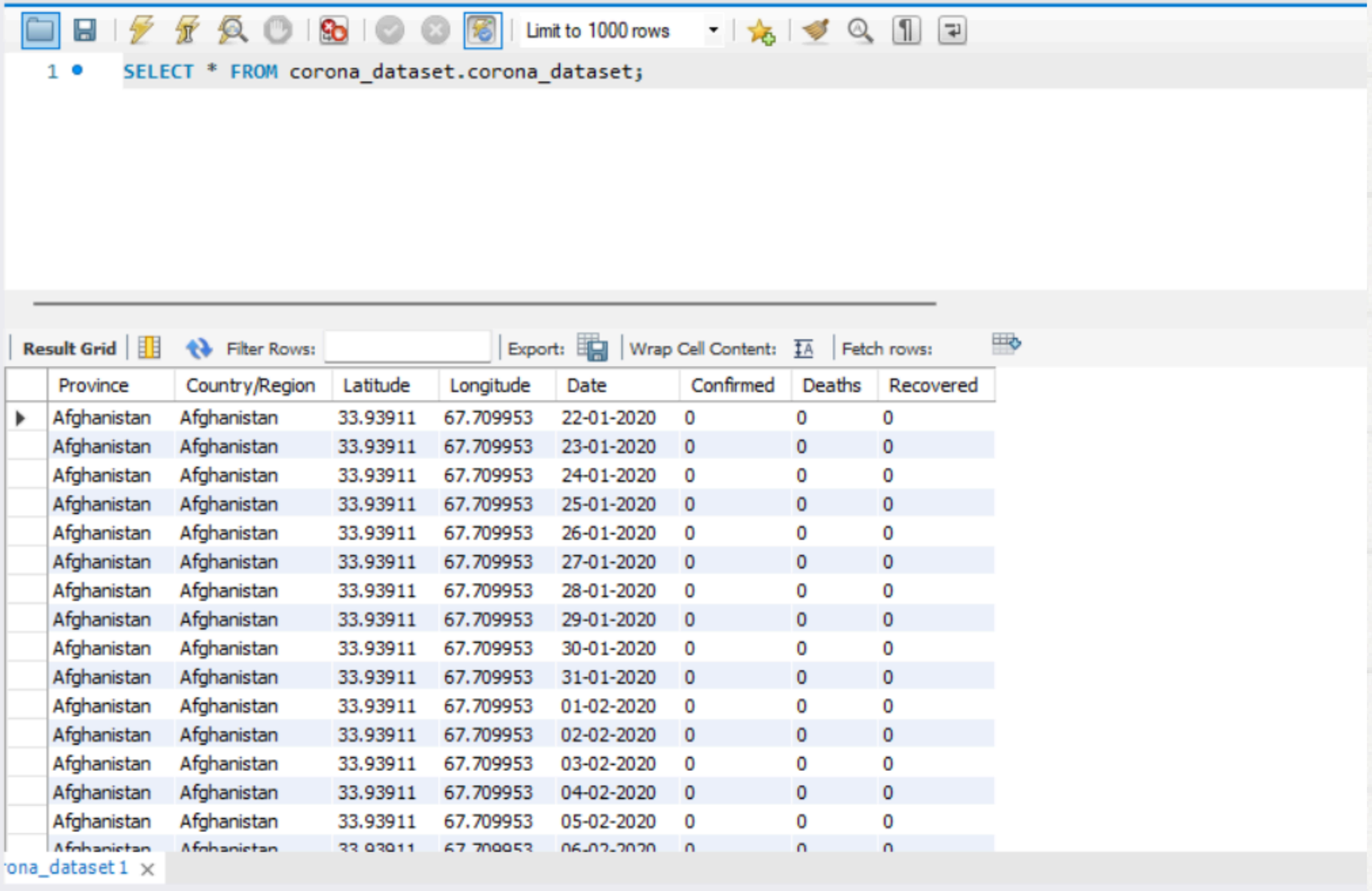
The dataset includes the following columns

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.

Problem Statement:

1. Write a code to check NULL values
2. If NULL values are present, update them with zeros for all columns.
3. check total number of rows
4. Check what is start_date and end_date
5. Number of month present in dataset
6. Find monthly average for confirmed, deaths, recovered
7. Find most frequent value for confirmed, deaths, recovered each month
8. Find minimum values for confirmed, deaths, recovered per year
9. Find maximum values of confirmed, deaths, recovered per year
10. The total number of case of confirmed, deaths, recovered each month
11. Check how corona virus spread out with respect to confirmed case
(Eg.: total confirmed cases, their average, variance & STDEV)
12. Check how corona virus spread out with respect to death case per month
(Eg.: total confirmed cases, their average, variance & STDEV)
13. Check how corona virus spread out with respect to recovered case
(Eg.: total confirmed cases, their average, variance & STDEV)
14. Find Country having highest number of the Confirmed case
15. Find Country having lowest number of the death case
16. Find top 5 countries having highest recovered case

✦ CSV File Imported into SQL workbench



Province	Country/Region	Latitude	Longitude	Date	Confirmed	Deaths	Recovered
Afghanistan	Afghanistan	33.93911	67.709953	22-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	23-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	24-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	25-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	26-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	27-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	28-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	29-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	30-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	31-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	01-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	02-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	03-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	04-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	05-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	06-02-2020	0	0	0

1. Write a code to check NULL values

Limit to 1000 rows

```
SELECT * FROM corona_dataset.corona_dataset;
SELECT *
FROM corona virus_dataset
WHERE (Confirmed,Deaths,Recovered IS NULL)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Province	Country/Region	Latitude	Longitude	Date	Confirmed	Deaths	Recovered
Afghanistan	Afghanistan	33.93911	67.709953	22-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	23-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	24-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	25-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	26-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	27-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	28-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	29-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	30-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	31-01-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	01-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	02-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	03-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	04-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	05-02-2020	0	0	0
Afghanistan	Afghanistan	33.93911	67.709953	06-02-2020	0	0	0

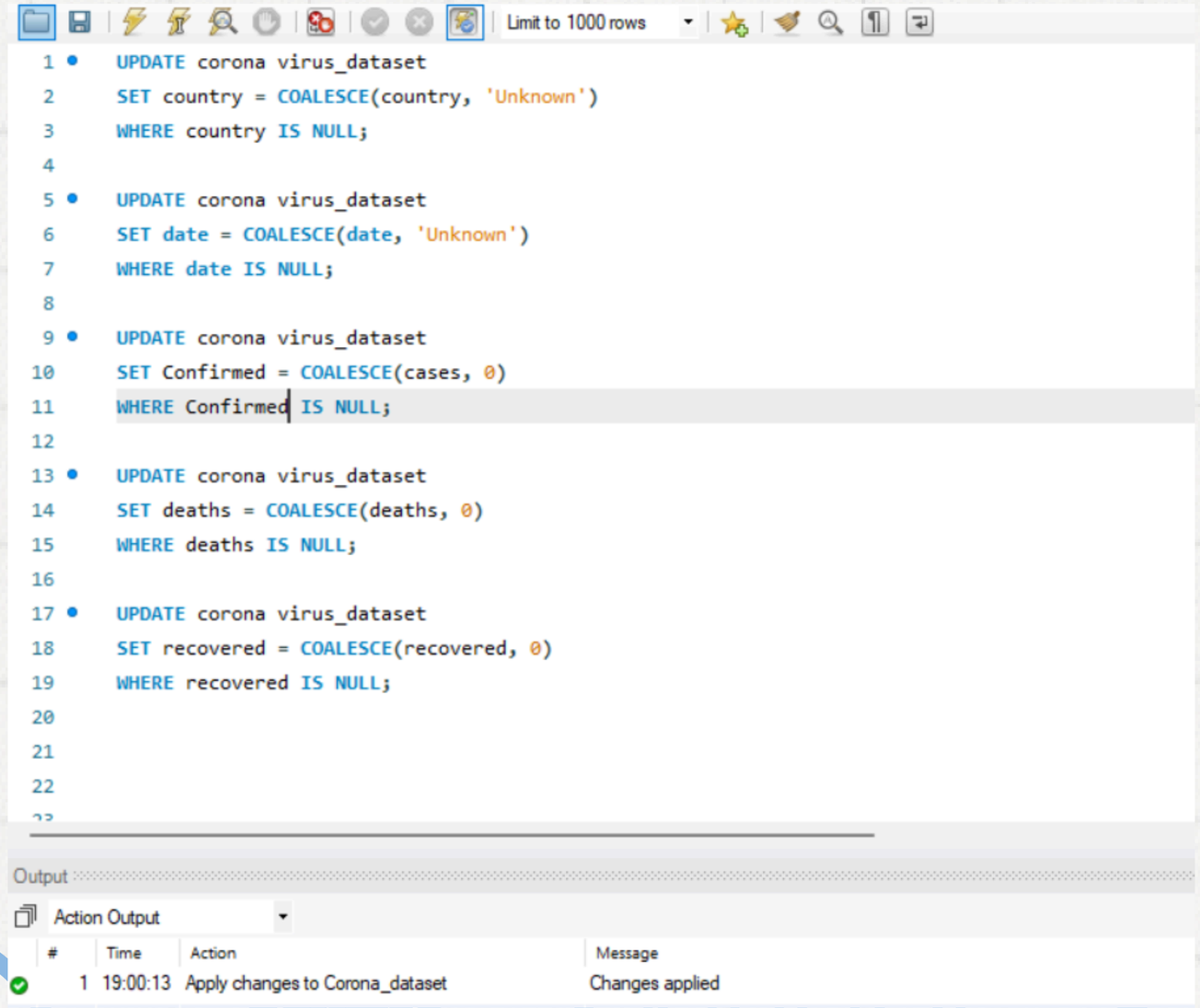
corona_dataset 2

Output

Action Output

#	Time	Action	Message
40	17:26:02	SELECT * FROM corona_dataset.corona_dataset LIMIT 0, 1000	1000 row(s) returned

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



```
UPDATE corona virus_dataset
SET country = COALESCE(country, 'Unknown')
WHERE country IS NULL;

UPDATE corona virus_dataset
SET date = COALESCE(date, 'Unknown')
WHERE date IS NULL;

UPDATE corona virus_dataset
SET Confirmed = COALESCE(cases, 0)
WHERE Confirmed IS NULL;

UPDATE corona virus_dataset
SET deaths = COALESCE(deaths, 0)
WHERE deaths IS NULL;

UPDATE corona virus_dataset
SET recovered = COALESCE(recovered, 0)
WHERE recovered IS NULL;
```

Output

Action Output

#	Time	Action	Message
1	19:00:13	Apply changes to Corona_dataset	Changes applied

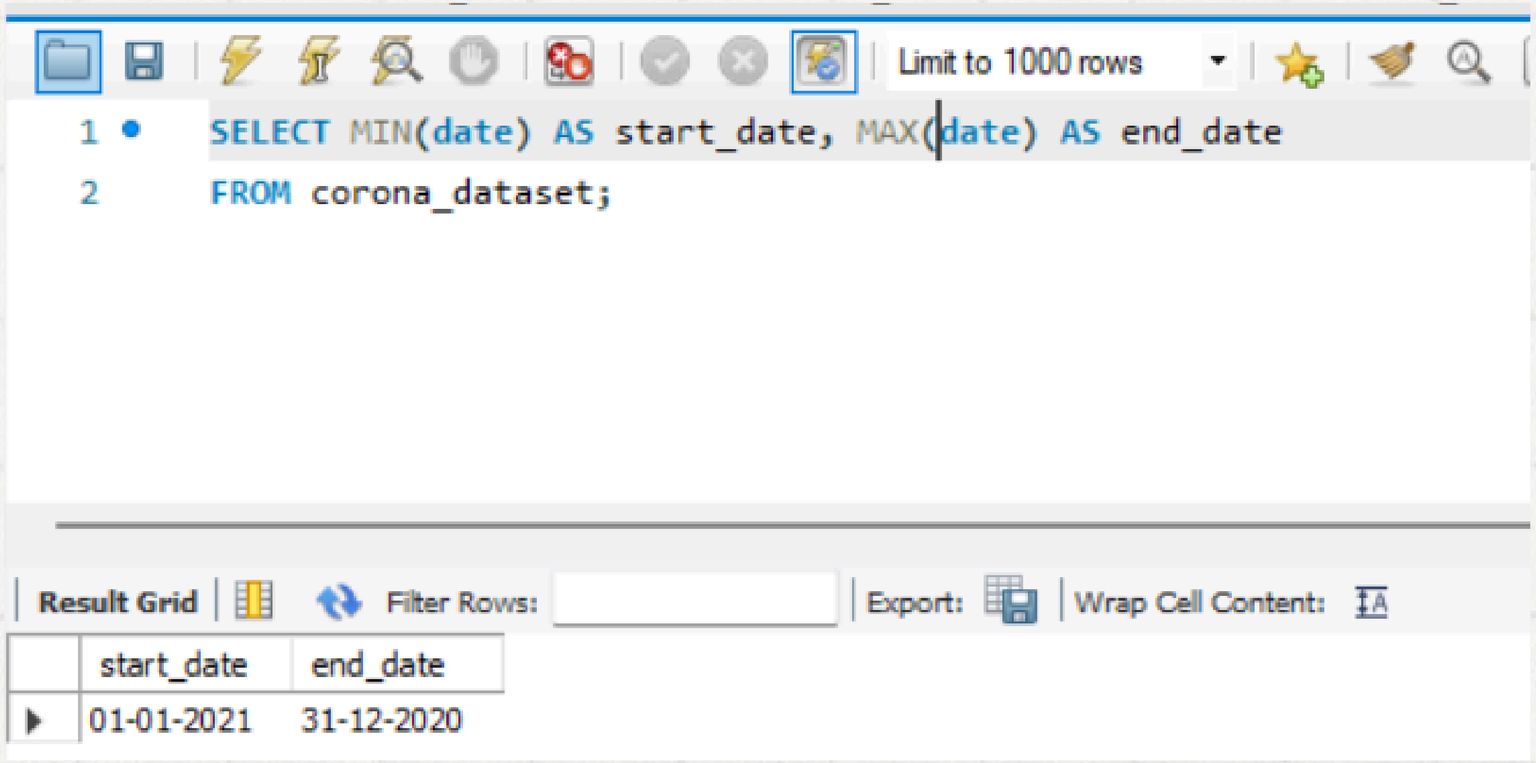
3.) Check total number of rows

```
1 •  Select count(*) As TotalRows
2    from corona_dataset
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	TotalRows
▶	78386

4. Check what is start_date and end_date



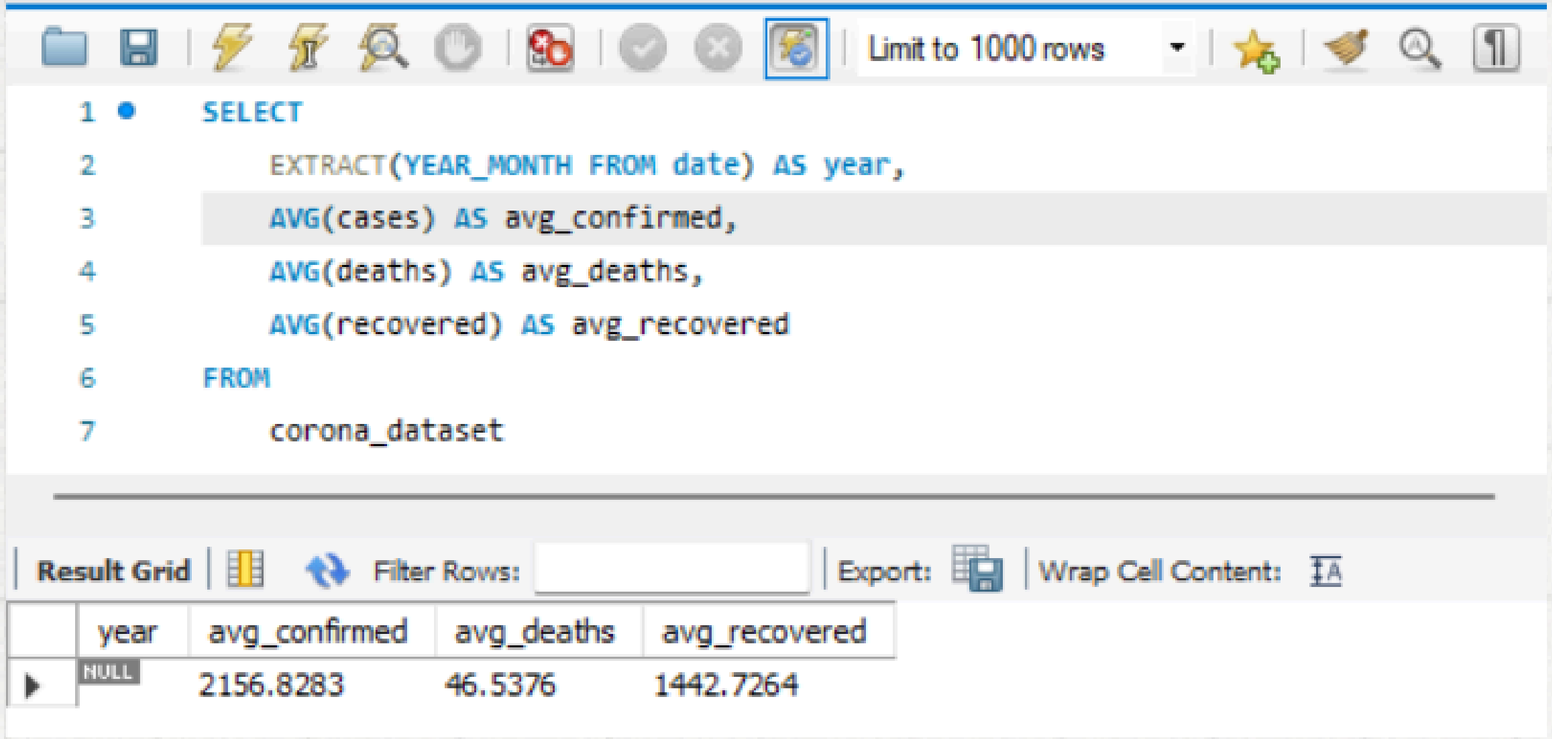
5. Number of month present in dataset

```
SELECT COUNT(DISTINCT CONCAT(YEAR(TRY_CONVERT(date,Date,105)), '-', MONTH(TRY_CONVERT(date,Date,105)))) AS num_of_months
FROM corona_dataset;
```

Results | Messages

	NumberOfMonths
1	18

6. Find monthly average for confirmed, deaths, recovered



7.) Monthly avg fo nfirmed dead recovey

SQLQuery1.sql - D...O9O2GQM\msi (55))*

```sql
SELECT
    DATEPART(YEAR, TRY_CONVERT(date, Date, 105)) AS Year,
    DATEPART(MONTH, TRY_CONVERT(date, Date, 105)) AS Month,
    AVG(CONVERT(float, TRY_CONVERT(int, Confirmed))) AS MonthlyAverageConfirmed,
    AVG(CONVERT(float, TRY_CONVERT(int, Deaths))) AS MonthlyAverageDeaths,
    AVG(CONVERT(float, TRY_CONVERT(int, Recovered))) AS MonthlyAverageRecovered
FROM
    Dataset
WHERE
    DATEPART(YEAR, TRY_CONVERT(date, Date, 105)) IN (2020, 2021)
GROUP BY
    DATEPART(YEAR, TRY_CONVERT(date, Date, 105)),
    DATEPART(MONTH, TRY_CONVERT(date, Date, 105))
ORDER BY
    Year, Month;
```

100 %

Results | Messages

	Year	Month	MonthlyAverageConfirmed	MonthlyAverageDeaths	MonthlyAverageRecovered
1	2020	1	4,16535947712418	0,124183006535948	0,0934640522875817
2	2020	2	14,6885282848772	0,593869731800766	7,0718954248366
3	2020	3	160,78431372549	8,6864853468269	26,9179843980603
4	2020	4	508,891067538126	41,7749455337691	171,965795206972
5	2020	5	578,45329959941	30,4739616276618	320,092135779043
6	2020	6	864,550544662309	30,0100217864924	552,118954248366
7	2020	7	1441,40965633565	35,3350200295172	989,141893316466
8	2020	8	1621,14526670883	37,7771452667088	1307,37254901961
9	2020	9	1795,44379084967	34,9847494553377	1446,90217864924
10	2020	10	2427,38667510015	36,9877714526671	1429,36095298334
11	2020	11	3613,92614379085	57,1213507625272	1997,55206971678
12	2020	12	4071,19586759435	71,6013071895425	2511,03415559772
13	2021	1	3933,26291376766	84,6268184693232	1926,8376554923
14	2021	2	2446,57819794585	69,5749299719888	1565,59850606909
15	2021	3	2932,99219903015	59,5593506219692	1660,05566097407
16	2021	4	4725,93159041394	78,9305010893246	3091,20021786492
17	2021	5	4027,60236137466	77,2542694497154	4029,66814252583
18	2021	6	2521,27099044746	66,6827551533434	2783,7501256913

8.) Minimum values for confirmed, deaths, recovered per Year

SQLQuery1.sql - D...O9O2GQM\msi (54))*

```
SELECT
    YEAR(TRY_CONVERT(date, Date, 105)) AS Year,
    MIN(TRY_CONVERT(int, Confirmed)) AS MinConfirmed,
    MIN(TRY_CONVERT(int, Deaths)) AS MinDeaths,
    MIN(TRY_CONVERT(int, Recovered)) AS MinRecovered
FROM
    Dataset
GROUP BY
    YEAR(TRY_CONVERT(date, Date, 105));
```

100 %

Results | Messages | Live Query Statistics

	Year	MinConfirmed	MinDeaths	MinRecovered
1	2021	0	0	0
2	2020	0	0	0

9.) Maximum values for confirmed, deaths, recovered per Year

SQLQuery1.sql - D...O9O2GQM\msi (54))*

```
SELECT
    YEAR(TRY_CONVERT(date, Date, 105)) AS Year,
    MAX(TRY_CONVERT(int, Confirmed)) AS MaxConfirmed,
    MAX(TRY_CONVERT(int, Deaths)) AS MaxDeaths,
    MAX(TRY_CONVERT(int, Recovered)) AS MaxRecovered
FROM
    Dataset
GROUP BY
    YEAR(TRY_CONVERT(date, Date, 105));
```

100 %

Results | Messages | Live Query Statistics

	Year	MaxConfirmed	MaxDeaths	MaxRecovered
1	2021	414188	7374	422436
2	2020	823225	3752	1123456

10.) The total number of cae of confirmed, death recovered each month

SQLQuery1.sql - D...O9O2GQM\msi (55))*

```sql
SELECT
    YEAR(TRY_CONVERT(date, Date, 105)) AS Year,
    MONTH(TRY_CONVERT(date, Date, 105)) AS Month,
    SUM(TRY_CONVERT(int, Confirmed)) AS TotalConfirmed,
    SUM(TRY_CONVERT(int, Deaths)) AS TotalDeaths,
    SUM(TRY_CONVERT(int, Recovered)) AS TotalRecovered
FROM
    Dataset
GROUP BY
    YEAR(TRY_CONVERT(date, Date, 105)),
    MONTH(TRY_CONVERT(date, Date, 105))
ORDER BY
    Year, Month;
```

100 %

Results | Messages

	Year	Month	TotalConfirmed	TotalDeaths	TotalRecovered
1	2020	1	6373	190	143
2	2020	2	65173	2635	31378
3	2020	3	762600	41200	127672
4	2020	4	2335810	191747	789323
5	2020	5	2743604	144538	1518197
6	2020	6	3968287	137746	2534226
7	2020	7	6836606	167594	4691500
8	2020	8	7689092	179177	6200868
9	2020	9	8241087	160580	6641281
10	2020	10	11513095	175433	6779459
11	2020	11	16587921	262187	9168764
12	2020	12	19309682	339605	11909835
13	2021	1	18655466	401385	9138991
14	2021	2	10481141	298059	6707024
15	2021	3	13911182	282490	7873644
16	2021	4	21692026	362291	14188609
17	2021	5	19102918	366417	19112716
18	2021	6	5014808	132632	5536879

11.) How corona virus spread out with respect to confirmed case

```sql
--how corona virus spread out with respect to confirmed case
SELECT
    COUNT(Confirmed) AS TotalConfirmedCases,
    AVG(TRY_CONVERT(BIGINT, Confirmed)) AS AverageConfirmedCases,
    SUM(POWER(TRY_CONVERT(BIGINT, Confirmed) - AVG_Cases.AverageConfirmedCases, 2)) / COUNT(Confirmed) AS VarianceConfirmedCases,
    SQRT(SUM(POWER(TRY_CONVERT(BIGINT, Confirmed) - AVG_Cases.AverageConfirmedCases, 2)) / COUNT(Confirmed)) AS StdDevConfirmedCases
FROM
    Dataset,
    (SELECT AVG(TRY_CONVERT(BIGINT, Confirmed)) AS AverageConfirmedCases FROM Dataset) AS AVG_Cases;
```

built-in function SQRT(expression float) RETURNS float

150 %

Results | Messages

	TotalConfirmedCases	AverageConfirmedCases	VarianceConfirmedCases	StdDevConfirmedCases
1	78386	2169	157265668	12540.5609124951

12.) How corona virus spread out with respect to death case per month

SQLQuery1.sql - D...O9O2GQM\msi (55))*

```sql
--how corona virus spread out with respect to death case per month

SELECT
    DATEPART(YEAR, TRY_CONVERT(date, Date, 105)) AS Year,
    DATEPART(MONTH, TRY_CONVERT(date, Date, 105)) AS Month,
    COUNT(Deaths) AS TotalDeathCases,
    AVG(TRY_CONVERT(BIGINT, Deaths)) AS AverageDeathCases,
    SUM(POWER(TRY_CONVERT(BIGINT, Deaths) - AVG_Cases.AverageDeathCases, 2)) / COUNT(Deaths) AS VarianceDeathCases,
    SQRT(SUM(POWER(TRY_CONVERT(BIGINT, Deaths) - AVG_Cases.AverageDeathCases, 2)) / COUNT(Deaths)) AS StdDevDeathCases
FROM
    Dataset,
    (SELECT AVG(TRY_CONVERT(BIGINT, Deaths)) AS AverageDeathCases FROM Dataset) AS AVG_Cases
GROUP BY
    DATEPART(YEAR, TRY_CONVERT(date, Date, 105)),
    DATEPART(MONTH, TRY_CONVERT(date, Date, 105))
ORDER BY
    Year, Month;
```

150 %

Results | Messages

	Year	Month	TotalDeathCases	AverageDeathCases	VarianceDeathCases	StdDevDeathCases
2	2020	1	1530	0	2296	47,9165942028438
3	2020	2	4437	0	2316	48,1248376620639
4	2020	3	4743	8	5471	73,9662085009094
5	2020	4	4590	41	40797	201,982672524155
6	2020	5	4743	30	21121	145,330657467721
7	2020	6	4590	30	17358	131,749762808136
8	2020	7	4743	35	21430	146,389890361322
9	2020	8	4743	37	23520	153,362316101447
10	2020	9	4590	34	20396	142,814565083538
11	2020	10	4743	36	17808	133,446618540898
12	2020	11	4590	57	28018	167,385781952948
13	2020	12	4743	71	66306	257,499514562649
14	2021	1	4743	84	104741	323,637142491402
15	2021	2	4284	69	69365	263,372359977276
16	2021	3	4743	59	54855	234,211442931382
17	2021	4	4590	78	96149	310,079022186281
18	2021	5	4743	77	133451	365,309457857308

13.) Country having highest number of the confirmed case.

SQLQuery1.sql - D...O9O2GQM\msi (55))*

```
SELECT TOP 1
    [Country Region] AS Country,
    SUM(CAST(Confirmed AS INT)) AS TotalConfirmedCases
FROM
    Dataset
WHERE
    ISNUMERIC(Confirmed) = 1
GROUP BY
    [Country Region]
ORDER BY
    SUM(CAST(Confirmed AS INT)) DESC;
```

150 %

Results | Messages

	Country	TotalConfirmedCases
1	US	33461982

14.) Countries having owest number of the death case

SQLQuery1.sql - D...O9O2GQM\msi (55))*

```
SELECT
    [Country Region] AS Country,
    SUM(CAST(Deaths AS INT)) AS TotalDeathCases
FROM
    Dataset
WHERE
    ISNUMERIC(Deaths) = 1
GROUP BY
    [Country Region]
HAVING
    SUM(CAST(Deaths AS INT)) = 0
ORDER BY
    TotalDeathCases ASC;
```

Results | Messages

	Country	TotalDeathCases
1	Marshall Islands	0
2	Samoa	0
3	Kiribati	0
4	Dominica	0

15.) Top 5 countries having highest recoveried case

```sql
SELECT TOP 5
    [Country Region] AS Country,
    SUM(CAST(Recovered AS INT)) AS TotalRecoveredCases
FROM
    Dataset
WHERE
    ISNUMERIC(Recovered) = 1
GROUP BY
    [Country Region]
ORDER BY
    TotalRecoveredCases DESC;
```

Results | Messages

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