

```
Create database covid19DB;
Create table CovidSurvey(
Continent Varchar(30),
Countries varchar(30),
Month Varchar(10),
Total_Cases float,
New_Cases float,
Total_Deaths float,
);

--Drop table covidSurvey;

Alter table CovidSurvey
Add Year int;

-- Select * from CovidSurvey;

Insert into CovidSurvey
Values
('Asia','India','Mar',3, 0, 0, 2020),
('Asia','India','May', 144950, 6414, 4172, 2020),
('Asia','India','Dec', 10266674, 21822, 148738, 2020),
('Asia','India','Apr', 19164969, 401993, 3523, 2021),
('Asia','Indonesia','Aug', 36928, 108, 1301,2020),
('Asia','Indonesia','Apr', 59745, 169, 2625, 2021),
('Europe','Albania','Aug', 7967, 155, 238, 2020),
('Asia','Indonesia','May', 22146, 684, 409, 2020),
('Europe','Albania','Sep', 7967, 116, 2394, 2021),
('Africa','Algeria','Mar', 7967, 3, 60, 2020),
('North America','Antigua','Nov', 139, 0, 4, 2020),
('North America','Barbuda','Dec', 1232, 0, 0, 2021),
('South America','Argentina','Mar', 4428, 143, 218, 2020),
('South America','Argentina','Dec', 2977363, 22420, 63865, 2021),
('Asia','Armenia','Dec', 216064, 536, 4108, 2021),
('Oceania','Australia','Feb', 15, 0, 0, 2020),
('Oceania','Australia','Aug', 25819 , 73, 657, 2020),
('Asia','Yemen','May', 70, 5, 10, 2020),
('North America','United States','Mar', 30395171, 61249, 551407, 2021),
('Africa','Zambia','Mar', 84, 2, 5, 2020);

TRUNCATE TABLE CovidSurvey;

Update CovidSurvey
SET countries='india', New_cases=1
Where Total_cases = '3';

Delete from covidSurvey
where Total_Deaths=0;
```

```
--IMPORTED DATA FROM DEVICE
```

```
--[CovidDeaths]
```

```
--[CovidVaccinations]
```

```
Select * From CovidDeaths;
```

```
Select Top 10 * From CovidVaccinations;
```

```
Select Distinct location from CovidDeaths --The SELECT DISTINCT statement is used ↗  
to return only distinct (different) values
```

```
Where Continent='Asia';
```

```
-----  
--SQL Aggregate Functions  
-----
```

```
Select * From CovidDeaths
```

```
Order By Total_Cases Desc; --The ORDER BY keyword is used to sort the result-set ↗  
in ascending or descending order
```

```
Select Min(Total_Deaths) as TotalDeaths -- The MIN() function returns the smallest ↗  
value of the selected column.
```

```
From CovidDeaths;
```

```
Select Max(Total_Deaths) as TotalDeaths --The MAX() function returns the largest ↗  
value of the selected column.
```

```
From CovidDeaths;
```

```
Select Count(*) as TotalRows --The COUNT() function returns the number of rows ↗  
that matches a specified criterion.
```

```
From CovidDeaths;
```

```
Select Sum(total_cases) as TotalDeaths --The SUM() function returns the total sum ↗  
of a numeric column.
```

```
From CovidDeaths
```

```
Where location='India';
```

```
Select Count(location)as count,continent as location
```

```
From CovidDeaths
```

```
Group By continent /*The GROUP BY statement is often used with aggregate functions ↗  
(COUNT(), MAX(), MIN(), SUM(), AVG())
```

```
to group the result-set by one or more columns.*/
```

```
ORDER BY continent Asc;
```

```
SELECT location, COUNT(*) AS total_count
```

```
FROM CovidDeaths
```

```
GROUP BY location;
```

```
-----
```

```
--Having clause
```

```
-----
/*
Quation:- Write a SQL Script to find Total cases and total deaths in each location
*/
SELECT location,
SUM(cast(total_cases As BIGINT)) AS total_cases, SUM(Cast(total_deaths As BIGINT)) AS total_deaths
From CovidDeaths
Group By location
Having sum(Cast(total_cases As BIGINT)) > 10000;

-----
-- Write a Script to find Minimum Cases in each location ?

Select Top 5 location,Min(total_cases) AS Min_Cases From CovidDeaths
Group By location
Having min(total_cases) < 100
Order By Min_cases Desc;

-----
-- Logical Operators --> AND, OR, NOT, IN, BETWEEN, LIKE, IS NULL
-----

Select * From CovidDeaths
Where location LIKE 'A%'; --The LIKE operator is used in a WHERE clause to search
    for a specified pattern in a column.

Select Top 10 * from CovidDeaths
where Continent = 'Asia' AND location = 'India';

Select Top 10 * From CovidDeaths
Where Continent = 'Asia' AND (Location Like 'I%' OR location Like 'O%');

Select top 50 location, total_cases
From CovidDeaths
Where total_cases Between 50 AND 100;

Select Top 100 location, total_cases
from CovidDeaths
Where NOT total_cases Between 0 AND 50
Order By total_cases Asc;

Select Top 20 * From CovidDeaths
Where location in ('India','Iran'); --The IN operator allows you to specify
    multiple values in a WHERE clause.

Select Distinct(location),Date,people_Vaccinated
from CovidVaccinations
Where people_Vaccinated IS NULL; --The IS NULL operator is used to test for empty
    values (NULL values).
```

```

-----
--Few SQL Wildcards
-----

Select Top 5 * From CovidDeaths
Where location Like 'Ind%'; -- Return all locations that starts with the letters 'Ind':

Select Top 15 * From CovidDeaths
Where location Like '[YOZ]>'; -- Return all locations starting with either "Y", "O", or "Z":

Select Top 20 * From CovidDeaths
Where location Like '_ndia'; -- Return all locations starting with any character, followed by "dia":

Select Distinct location From CovidDeaths
Where location Like '[u-z]>'; -- Return all location starting with "u", "v", "w", "x", "y" or "z":
-----

-- Total Cases vs Total Deaths in 'India'

Select Location, date, total_cases, total_deaths, (total_deaths/total_cases)*100 as DeathPercentage
From CovidDeaths
Where location like 'India%'
and continent is not null

-- Total Cases vs Population
-- Shows what percentage of population infected with Covid

Select Location, date, Population, total_cases, (total_cases/population)*100 as PercentPopulationInfected
From CovidDeaths
Where location like 'India%'

-- Countries with Highest Death Count per Population

Select Location, MAX(cast(Total_deaths as bigint)) as TotalDeathCount
From CovidDeaths
--Where location like '%states%'
Where continent is not null
Group by Location
order by TotalDeathCount desc

-----
--SQL JOINS
-----

```

```
-- For Joins I Used another Data, (Employee Dataset)
-- Fetch the Employee name and the department name they belong to

--** Inner join / Join **

--Select * From Employees;
--Select * From department;
--Select * From Manager;
--Select * From Project;

Select e.Emp_name, d.Dept_name
From Employees e
Join department d --The INNER JOIN keyword selects records that have matching
    values in both tables.
On e.dept_id = d.dept_id;

-- Fetch All the Employee name and their department name they belong to

/* @@ Left Join ----->
The LEFT JOIN keyword returns all records from the left table (table1),
and the matching records from the right table (table2), Remaining it returns Null.
*/

Select e.Emp_name, d.Dept_name
From Employees e
Left Join department d
On e.dept_id = d.dept_id;

/* @@ Right Join
The RIGHT JOIN keyword returns all records from the right table (table2),
and the matching records from the left table (table1), Remaining it returns Null
*/

Select e.Emp_name, d.Dept_name
From Employees e
Right Join department d
On e.dept_id = d.dept_id;

/* @@ Full Join
The FULL OUTER JOIN keyword returns all records when there is a match in left
(table1) or right (table2) table records.*/
Select e.Emp_name, d.Dept_name
From Employees e
Full Join department d
On e.dept_id = d.dept_id;

-----
-- Fetch details of All emp and their Manager, departement and the projects they
    work on.
```

```
Select e.Emp_name, d.dept_Name, m.Manager_Name, p.project_name
From Employees e
left Join department d On e.Dept_id = d.dept_id
join Manager m On m.manager_id = e.manager_id
left Join Project p On e.emp_id = p.team_member_id;
```

```
Select dea.continent, dea.location, dea.date, vac.new_vaccinations
From Covid19..CovidDeaths As dea
Join Covid19..CovidVaccinations As Vac
    On dea.location = vac.location
    and dea.date = vac.date
Where dea.Continent is Not Null AND vac.new_vaccinations is not null;
```

-- Total Population vs Vaccinations
-- Shows Percentage of Population that has recieved at least one Covid Vaccine

```
Select dea.continent, dea.location, dea.date, dea.population, vac.new_vaccinations
, SUM(CONVERT(int,vac.new_vaccinations)) OVER (Partition by dea.Location Order by ↗
    dea.location, dea.Date) as RollingPeopleVaccinated
--, (RollingPeopleVaccinated/population)*100
From CovidDeaths dea
Join CovidVaccinations vac
    On dea.location = vac.location
    and dea.date = vac.date
where dea.continent is not null
order by 2,3
```
