--Creating table covid deaths

CREATE TABLE public.covid\_deaths

(

iso\_code VARCHAR (50),

continent VARCHAR(50),

location VARCHAR(50),

Date date,

population numeric (18,0),

total\_cases numeric (18,0),

new\_cases numeric (18,5),

new\_cases\_smoothed numeric (18,5),

total\_deaths numeric (18,0),

new\_deaths numeric (18,5),

new\_deaths\_smoothed numeric (18,5),

total\_cases\_per\_million numeric (18,5),

new\_cases\_per\_million numeric (18,5),

new\_cases\_smoothed\_per\_million numeric (18,5),

total\_deaths\_per\_million numeric (18,5),

new\_deaths\_per\_million numeric (18,5),

new\_deaths\_smoothed\_per\_million numeric (18,5),

reproduction\_rate numeric (18,5),

icu\_patients numeric (18,5),

icu\_patients\_per\_million numeric (18,5),

hosp\_patients numeric (18,5),

hosp\_patients\_per\_million numeric (18,5),

weekly\_icu\_admissions numeric (18,5),

weekly\_icu\_admissions\_per\_million numeric (18,5),

weekly\_hosp\_admissions numeric (18,5),

weekly\_hosp\_admissions\_per\_million numeric (18,5))

--Creating Table covid vaccinations

CREATE TABLE public.covid\_vaccinations

(iso\_code VARCHAR(50),

continent VARCHAR(50),

location VARCHAR(50),

DATE date,

new\_tests integer,

total\_tests integer,

total\_tests\_per\_thousand numeric (18,5),

new\_tests\_per\_thousand numeric (18,5),

new\_tests\_smoothed integer,

new\_tests\_smoothed\_per\_thousand numeric (18,5),

positive\_rate numeric (18,5),

tests\_per\_case numeric (18,5),

tests\_units VARCHAR(50),

total\_vaccinations bigint,

people\_vaccinated integer,

people\_fully\_vaccinated integer,

new\_vaccinations integer,

new\_vaccinations\_smoothed integer,

total\_vaccinations\_per\_hundred numeric (18,5),

people\_vaccinated\_per\_hundred numeric (18,5),

people\_fully\_vaccinated\_per\_hundred numeric (18,5),

new\_vaccinations\_smoothed\_per\_million integer,

stringency\_index numeric (18,5),

population\_density numeric (18,5),

median\_age numeric (18,5),

aged\_65\_older numeric (18,5),

aged\_70\_older numeric (18,5),

gdp\_per\_capita numeric (18,5),

extreme\_poverty numeric (18,5),

cardiovasc\_death\_rate numeric (18,5),

diabetes\_prevalence numeric (18,5),

female\_smokers numeric (18,5),

male\_smokers numeric (18,5),

handwashing\_facilities numeric (18,5),

hospital\_beds\_per\_thousand numeric (18,5),

life\_expectancy numeric (18,5),

human\_development\_index numeric (18,5),

excess\_mortality numeric (18,5))

SELECT \* FROM covid\_deaths

ORDER BY location, date;

SELECT \* FROM covid\_vaccinations

ORDER BY location, date;

--Select data that we are going to be using

SELECT location, date, total\_cases, new\_cases, total\_deaths, population FROM covid\_deaths

ORDER BY location, date;

--Looking at Total Cases vs Total Deaths

--Shows the likelihood of dying from COVID if you contract the virus in the countries in this data set

SELECT location, date, total\_cases, total\_deaths,

ROUND((total\_deaths/total\_cases)\*100,2) AS deathrate

FROM covid\_deaths

ORDER BY location, date;

--Looking at the total cases vs population

SELECT location, date, total\_cases, population,

ROUND((total\_cases/population)\*100,2) AS infectionrate

FROM covid\_deaths

ORDER BY location, date;

--Looking at countries with highest infection rate vs population

SELECT location, population, MAX(total\_cases) AS highestinfectioncount,

MAX(ROUND((total\_cases/population)\*100,2)) AS highestinfectionrate FROM covid\_deaths

GROUP BY location, population

ORDER BY highestinfectionrate DESC NULLS LAST;

--Showing countries with the highest death count perpopulation

SELECT location, MAX(total\_deaths) AS highestdeathcount FROM covid\_deaths

WHERE continent IS NOT NULL

GROUP BY location

ORDER BY highestdeathcount DESC NULLS LAST;

--Showing the continents with the highest death count

SELECT continent, MAX(total\_deaths) AS totaldeathcount FROM covid\_deaths

WHERE continent IS NOT NULL

GROUP BY continent

ORDER BY totaldeathcount DESC;

--Global numbers by date

SELECT date, SUM(new\_cases) AS totalcases, SUM(new\_deaths) AS totaldeaths,

CASE WHEN SUM(new\_cases) = 0 THEN NULL ELSE ROUND((SUM(new\_deaths)/SUM(new\_cases))\*100,2)

END AS deathpercentage FROM covid\_deaths

WHERE continent IS NOT NULL

GROUP BY date

ORDER BY date;

--Global numbers

SELECT ROUND(SUM(new\_cases),0) AS totalcases, ROUND(SUM(new\_deaths),0) AS totaldeaths,

ROUND((SUM(new\_deaths)/SUM(new\_cases))\*100,2) AS deathpercentage FROM covid\_deaths

WHERE continent IS NOT NULL;

--Looking at total population vs vaccinated population

SELECT a.continent, a.location, a.date, a.population, b.new\_vaccinations FROM covid\_deaths AS a

INNER JOIN covid\_vaccinations AS b

ON a.location = b.location AND a.date = b.date

WHERE a.continent IS NOT NULL AND a.location = 'United Kingdom'

ORDER BY a.continent, a.location, a.date;

SELECT a.continent, a.location, a.date, a.population, b.new\_vaccinations,

SUM(b.new\_vaccinations) OVER(PARTITION BY a.location ORDER BY a.date) AS rollingpeoplevaccinated

FROM covid\_deaths AS a

INNER JOIN covid\_vaccinations AS b

ON a.location = b.location AND

a.date = b.date

WHERE a.continent IS NOT NULL

ORDER BY a.location, a.date;

CREATE VIEW percentpopulationvaccinated AS

(SELECT a.continent, a.location, a.date, a.population, b.new\_vaccinations,

SUM(b.new\_vaccinations) OVER(PARTITION BY a.location ORDER BY a.date) AS rollingpeoplevaccinated,

ROUND((SUM(b.new\_vaccinations) OVER(PARTITION BY a.location ORDER BY a.date)/a.population)\*100,2)AS rollingpeoplevaccinatedpercentage

FROM covid\_deaths AS a

INNER JOIN covid\_vaccinations AS b

ON a.location = b.location AND

a.date = b.date

WHERE a.continent IS NOT NULL

ORDER BY a.location, a.date);