# 1. Creating the database and importing the data

-- Creating the database in MySQL

CREATE DATABASE covid19\_data;

USE covid19\_data;

-- Creating a table for the covid deaths csv

CREATE TABLE covid\_deaths (

    iso\_code CHAR(3), continent TINYTEXT,

    location TEXT, date\_ DATE, population INT, total\_cases INT,

    new\_cases INT, new\_cases\_smoothed FLOAT(3), total\_deaths INT,

    new\_deaths INT, new\_deaths\_smoothed FLOAT(3), total\_cases\_per\_million FLOAT(3),

    new\_cases\_per\_million FLOAT(3), new\_cases\_smoothed\_per\_million FLOAT(3),

    total\_deaths\_per\_million FLOAT(3), new\_deaths\_per\_million FLOAT(3),

    new\_deaths\_smoothed\_per\_million FLOAT(3), reproduction\_rate FLOAT(3),

    icu\_patients INT, icu\_patients\_per\_million FLOAT(3), hosp\_patients INT,

    hosp\_patients\_per\_million FLOAT(3), weekly\_icu\_admissions INT,

    weekly\_icu\_admissions\_per\_million FLOAT(3), weekly\_hosp\_admissions INT,

    weekly\_hosp\_admissions\_per\_million FLOAT(3)

    );

CREATE TABLE covid\_vaccinations (

    iso\_code CHAR(3), continent TINYTEXT, location TEXT, date\_ DATE,

    new\_tests INT, total\_tests INT,

    total\_tests\_per\_thousand FLOAT(3),

    new\_tests\_per\_thousand FLOAT(3),

    new\_tests\_smoothed INT,

    new\_tests\_smoothed\_per\_thousand FLOAT(3),

    positive\_rate FLOAT(3), tests\_per\_case FLOAT(3),

    tests\_units TEXT, total\_vaccinations INT,

    people\_vaccinated INT, people\_fully\_vaccinated INT,

    total\_boosters INT, new\_vaccinations INT,

    new\_vaccinations\_smoothed INT,

    total\_vaccinations\_per\_hundred FLOAT(3),

    people\_vaccinated\_per\_hundred FLOAT(3),

    people\_fully\_vaccinated\_per\_hundred FLOAT(3),

    total\_boosters\_per\_hundred FLOAT(3),

    new\_vaccinations\_smoothed\_per\_million INT,

    new\_people\_vaccinated\_smoothed INT,

    new\_people\_vaccinated\_smoothed\_per\_hundred FLOAT(3),

    stringency\_index FLOAT(3), population\_density FLOAT(3),

    median\_age INT, aged\_65\_older FLOAT(3), aged\_70\_older FLOAT(3),

    gdp\_per\_capita FLOAT(3), extreme\_poverty FLOAT(3),

    cardiovasc\_death\_rate FLOAT(3), diabetes\_prevalence FLOAT(3),

    female\_smokers FLOAT(3), male\_smokers FLOAT(3),

    handwashing\_facilities FLOAT(3),

    hospital\_beds\_per\_thousand FLOAT(3),

    life\_expectancy FLOAT(3), human\_development\_index FLOAT(3),

    excess\_mortality\_cumulative\_absolute FLOAT(3),

    excess\_mortality\_cumulative FLOAT(3),

    excess\_mortality FLOAT(3),

    excess\_mortality\_cumulative\_per\_million FLOAT(3)

    );

-- Loading data from my CSVs in the repository into my MySQL DB

LOAD DATA LOCAL INFILE 'C:/Users/Dillon/Documents/Coding Learning/Projects/covid19/CovidDeaths.csv'

INTO TABLE covid\_deaths

COLUMNS TERMINATED BY ','

IGNORE 1 LINES;

LOAD DATA LOCAL INFILE 'C:/Users/Dillon/Documents/Coding Learning/Projects/covid19/CovidVaccinations.csv'

INTO TABLE covid\_vaccinations

COLUMNS TERMINATED BY ','

IGNORE 1 LINES;

# 2. Checks

-- Checking the table has populated

SELECT \*

FROM covid19\_data.covid\_deaths

LIMIT 100;

-- Checking the table has populated

SELECT \*

FROM covid19\_data.covid\_vaccinations

LIMIT 100

# 3. Exploratory Analysis

## 3.1 Calculating Death Rate

-- Total Cases vs Total Deaths

-- Shows death rate (Likelihood of death after contraction)

SELECT location, date\_, total\_cases, total\_deaths, (total\_deaths/total\_cases)\*100 AS prop\_death

FROM covid19\_data.covid\_deaths

WHERE  date\_ = '2022-02-26'

order by 5 DESC;

## 3.2 Percentage of the Population that have tested positive

-- Total Cases vs Population

--Shows percentage of population that have tested positive for the virus (ordered from low to high)

SELECT location, date\_, total\_cases, population, (total\_cases/population)\*100 AS perc\_pop\_pos

FROM covid19\_data.covid\_deaths

WHERE  date\_ = '2022-02-26'

order by 5 DESC;

## 3.3 Countries with highest death counts

-- Finding countries with highest death counts

-- Needed to add the where clause due to data in table for continents and the world. Clause excludes the data to show only countries

-- Change the <> to = to filter by continents & the world instead of countries

SELECT location, MAX(total\_deaths) as totaldeathcount

FROM covid19\_data.covid\_deaths

WHERE continent <> ""

GROUP BY location

ORDER BY totaldeathcount DESC;

## 3.4 Checking manual calculation of total world deaths vs the value in the data

-- Running a check on the "World" count, comparing the datas value vs my calculated value

-- Pulling the value from the data

SELECT location, MAX(total\_deaths) as totaldeathcount

FROM covid19\_data.covid\_deaths

WHERE location = "world"

GROUP BY location

ORDER BY totaldeathcount DESC;

-- Calculating it myself

SELECT SUM(totals)

FROM

    (SELECT MAX(total\_deaths) AS totals

    FROM covid\_deaths

    WHERE continent <> ""

    GROUP BY location)

AS totalworlddeaths;

The calculated value was 5,944,420 vs the real value of 5,944,313. A difference of 107. It might be possible that a country was double counted or the WHERE continent <> "" clause didn’t filter out all non-countries.

After a check in excel, it seems that all locations from the manual calc are real countries so there might be an issue with the dataset’s calculation of the world value.