COVID-19 Worldwide Data Analysis Using SQL

Nazish Khalid

October 31, 2025

Contents

Abstract

The COVID-19 pandemic has profoundly impacted lives across the globe, creating a pressing need to analyze and understand trends in infection, mortality, and vaccination. This project uses SQL to extract, analyze, and interpret COVID-19 data obtained from global datasets. The analysis focuses on infection rates, death percentages, and vaccination progress across different countries and continents. By leveraging SQL queries, this study provides insights into how the pandemic evolved and how nations responded through vaccination efforts.

1. Introduction

The COVID-19 pandemic presented an unprecedented global challenge that affected public health, economies, and societies worldwide. Data analysis plays a crucial role in understanding the spread and impact of the virus. This project, titled *COVID-19 Worldwide Data Analysis Using SQL*, was designed to explore patterns and relationships within the COVID-19 dataset through structured SQL queries.

1.1 Objectives

The primary objectives of this analysis are:

- To determine mortality and infection rates by country and continent.
- To identify countries with the highest infection and death counts.
- To compare vaccination progress across different regions.
- To compute global death percentages and create visual-ready SQL views.

2. Tools and Technologies

- **SQL Server / MySQL**: Used for data analysis and querying.
- **COVID-19 Dataset**: Data source containing information about cases, deaths, and vaccinations.
- **CTEs and Views**: For intermediate data processing and visualization.

3. Data Exploration and Analysis

This section contains the SQL queries used to analyze the COVID-19 dataset. Each query focuses on extracting meaningful insights about the pandemic.

3.1 Preview the Data

```sql
-- Preview COVID Deaths dataset
SELECT *
FROM CovidDeaths
ORDER BY 3, 4;

-- Preview COVID Vaccinations dataset
SELECT * FROM CovidVaccinations ORDER BY 3, 4;
```

3.2 Basic Data Selection

```sql
SELECT
    location,
    date,
    total_cases,
    new_cases,
    total_deaths,
    population
FROM CovidDeaths
ORDER BY 1, 2;
```

3.3 Total Cases vs Total Deaths

```sql
SELECT
    location,
    date,
    total_cases,
    total_deaths,
    (total_deaths * 100.0 / total_cases) AS DeathPercentage
FROM CovidDeaths
WHERE location LIKE '%Pakistan%'
ORDER BY 1, 2;
```

3.4 Infection Percentage by Population

```sql
SELECT
    location,
    date,
    population,
    total_cases,
    (total_cases * 100.0 / population) AS CovidPercentage
FROM CovidDeaths
ORDER BY 1, 2;
```

3.5 Highest Infection Rate by Country

```sql
SELECT
    location,
    population,
    MAX(total_cases) AS HighestInfectionCount,
    MAX((total_cases * 100.0 / population)) AS
       InfectionPercentage
FROM CovidDeaths
GROUP BY location, population
ORDER BY InfectionPercentage DESC;
```

3.6 Countries with Highest Death Count

```sql
SELECT
    location,
    MAX(total_deaths) AS TotalDeathCount
FROM CovidDeaths
WHERE continent IS NOT NULL
GROUP BY location
ORDER BY TotalDeathCount DESC;
```

3.7 Continents with Highest Death Count

```sql
SELECT
    continent,
    MAX(total_deaths) AS TotalDeathCount
FROM CovidDeaths
WHERE continent IS NOT NULL
GROUP BY continent
ORDER BY TotalDeathCount DESC;
```

3.8 Global Daily Death Percentage

```sql
SELECT
    date,
    SUM(new_cases) AS Total_New_Cases,
    SUM(new_deaths) AS Total_New_Deaths,
    (SUM(new_deaths) * 100.0 / SUM(new_cases)) AS
       Death_Percentage
FROM CovidDeaths
WHERE continent IS NOT NULL
GROUP BY date
ORDER BY date;
```

3.9 Worldwide Death Percentage (All Time)

```sql
SELECT
```

```
    SUM(new_cases) AS Total_New_Cases,
    SUM(new_deaths) AS Total_New_Deaths,
    (SUM(new_deaths) * 100.0 / SUM(new_cases)) AS
       Death_Percentage
FROM CovidDeaths
WHERE continent IS NOT NULL;
```

3.10 Population vs Vaccination using CTE

```
WITH PopvsVac AS (
  SELECT
    d.continent,
    d.location,
    d.date,
    d.population,
    v.new_vaccinations,
    SUM(CAST(v.new_vaccinations AS BIGINT)) OVER (
      PARTITION BY d.location
      ORDER BY d.location, d.date
    ) AS Cumulative_Vaccinations
  FROM CovidDeaths d
  JOIN CovidVaccinations v
    ON d.location = v.location
    AND d.date = v.date
  WHERE d.continent IS NOT NULL
)
SELECT *,
  (Cumulative_Vaccinations * 100.0 / population) AS
     Vaccination_Percentage
FROM PopvsVac;
```

3.11 Creating a View for Visualization

```
CREATE VIEW PercentPopulationVaccination AS
SELECT
    d.continent,
    d.location,
```

```
    d.date,
    d.population,
    v.new_vaccinations,
    SUM(CAST(v.new_vaccinations AS BIGINT)) OVER (
      PARTITION BY d.location
      ORDER BY d.location, d.date
    ) AS Cumulative_Vaccinations
FROM CovidDeaths d
JOIN CovidVaccinations v
  ON d.location = v.location
  AND d.date = v.date
WHERE d.continent IS NOT NULL;
```

4. Results and Insights

From the SQL analysis, the following insights were derived:

- Death and infection rates vary significantly between countries.
- Countries with larger populations often show higher case counts but lower death percentages.
- Global death percentages offer a macro-level understanding of pandemic severity.
- Vaccination progress data reflects the global response to COVID-19.

5. Conclusion

This project demonstrates how SQL can be used to analyze real-world pandemic data. Through queries involving joins, aggregation, and window functions, we extracted key metrics such as infection rates, death percentages, and vaccination coverage. These insights can support further analysis and visualization in tools like Power BI or Tableau.