

COVID-19 Data Exploration: SQL Analysis Report

Introduction

This report presents an exploratory analysis of COVID-19 data using SQL queries. The dataset is sourced from PortfolioProject..CovidDeaths and PortfolioProject..CovidVaccinations, focusing on critical metrics such as total cases, deaths, population impact, and vaccination trends. The analysis employs advanced SQL techniques, including joins, common table expressions (CTEs), window functions, and aggregate operations, to derive meaningful insights.

Key Objectives

1. **Understanding COVID-19 Trends:** Analyzing total cases, deaths, and infection rates across different countries.
 2. **Assessing Mortality Rates:** Calculating the likelihood of dying after contracting COVID-19.
 3. **Evaluating Infection Spread:** Determining the percentage of the population affected by COVID-19.
 4. **Tracking Vaccination Progress:** Identifying the impact of vaccinations on cases and mortality.
 5. **Data Aggregation & Visualization:** Preparing views and summary tables for further insights.
-

SQL Techniques & Implementation

1. Data Overview & Initial Filtering

- Extracting relevant columns (location, date, total cases, new cases, total deaths, and population) from CovidDeaths.
- Filtering out irrelevant rows where the continent field is null.
- Ordering data by location and date for better trend analysis.

2. Mortality Rate Analysis

- Calculating the DeathPercentage as:
$$(\text{total_deaths} / \text{total_cases}) * 100$$
- Filtering by specific locations (e.g., United States) to understand region-specific impacts.

3. Infection Rate Analysis

- Determining the percentage of the population infected by COVID-19:
$$(\text{total_cases} / \text{population}) * 100$$
- Sorting results to identify countries with the highest infection rates.

4. Vaccination Analysis

- Joining CovidDeaths and CovidVaccinations tables to integrate case and vaccination data.

- Calculating total vaccinations administered and their impact on case numbers.

5. CTEs & Window Functions for Advanced Insights

- Using Common Table Expressions (CTEs) to break down complex calculations into readable sections.
- Applying Window Functions to compute rolling totals and moving averages for trend analysis.

6. Creating Views for Reusability

- Generating SQL views to store aggregated data for further visualization and reporting:
 - `CREATE VIEW GlobalCovidStats AS`
 - `SELECT location, date, total_cases, total_deaths, population,`
 - `(total_deaths / total_cases) * 100 AS DeathPercentage`
 - `FROM PortfolioProject..CovidDeaths;`
 - These views facilitate easy data retrieval for further analysis.
-

Key Insights

- The mortality rate varies significantly by country, with some regions experiencing higher DeathPercentage values due to healthcare disparities.
 - Infection rates relative to population highlight how COVID-19 spread differently across continents.
 - Countries with higher vaccination rates generally show lower mortality rates, reinforcing the effectiveness of vaccination programs.
 - The use of window functions enables tracking cumulative trends over time, revealing critical inflection points in case surges and vaccination rollouts.
-

Conclusion

This SQL-based exploration of COVID-19 data provides valuable insights into case trends, mortality rates, and vaccination impacts. By leveraging advanced SQL techniques such as CTEs, joins, and window functions, the analysis offers a structured and insightful approach to understanding the pandemic's effects. These findings can be further utilized for real-time monitoring and decision-making in public health.

Next Steps

- Enhance the analysis with additional datasets, such as hospital capacity and economic impact data.
- Implement stored procedures for automated reporting.

- Visualize trends using Power BI or Tableau for better interpretability.

This report showcases expertise in SQL-based data exploration and is a valuable addition to a professional portfolio, demonstrating proficiency in handling large datasets and deriving actionable insights.