

# COVID-19 Data Analytics Report

Author : Sai Chaitanya

Date : September 2025

## Abstract

This project focuses on analyzing COVID-19 data using SQL to identify country-wise and date-wise trends. The dataset contains confirmed cases, deaths, recoveries, and other related attributes. SQL queries, views, and analytics functions were applied to gain meaningful insights into the global pandemic scenario

## Objectives

- To create a relational database from COVID-19 datasets .
- To explore country-wise statistics such as total cases, deaths, and recovery rates
- To apply SQL queries, subqueries, and views for analytics.
- To demonstrate the use of window functions and aggregates in SQL.
- To build a reusable database for future analysis.

## Tools & Dataset

Database: MySQL Workbench (alternative: SQLite DB Browser).

Dataset: COVID-19 CSV files from Kaggle.

Version Control: Git + GitHub for collaboration .

## Sample Table: country\_wise\_latest

country_Region	Confirmed	Deaths	Recovered	Active	Deaths/100 Cases	Recovered/100 Cases	WHO Region
India	3,542,733	63,498	2,742,380	736,855	1.79	77.4	South-East Asia

## Methodology

- Schema Design – Defined tables for country-level COVID-19 data.
- Data Import – Loaded CSV data into MySQL using LOAD DATA INFILE
- Data Cleaning – Handled missing values and infinite ratios (NULLIF).
- Querying – Performed aggregation, filtering, and calculations.
- Views – Created reusable summaries (e.g., top countries by cases).
- Analytics – Applied window functions for running totals and weekly trends.

## Findings

- The Americas and South-East Asia regions reported the highest number of confirmed cases
- Mortality rates varied significantly across regions, with Europe having higher death ratios
- Window functions allowed cumulative analysis of confirmed cases effectively
- Recovered-to-confirmed ratio highlighted countries with strong healthcare responses

## Conclusion

This project successfully demonstrated how SQL can be used for analytics on real-world COVID-19 data. By leveraging queries, subqueries, and views, meaningful insights were extracted.

The project also highlighted the importance of data cleaning and handling edge cases such as missing or infinite values. Future

### **improvements could include:**

- Automating data updates from live APIs.
- Integrating visualizations (Tableau/Power BI).
- Applying machine learning for trend forecasting.