

QUERY-1: BASE DATA SELECTION

Selects core data fields (location, date, cases, deaths, population) from the CovidDeaths table.

Final Output Fields: location, date, total_cases, new_cases, total_deaths, population.

```
SELECT
    location,
    date,
    total_cases,
    new_cases,
    total_deaths,
    population
FROM
    project..CovidDeaths
ORDER BY
    location, date; -- Order by location first, then date
```

QUERY-2: COVID DEATH LIKELIHOOD (PAKISTAN)

Calculates the likelihood of dying if infected with COVID-19 specifically in Pakistan.

Final Output Fields: location, date, total_cases, total_deaths, Death_Percentage

```
SELECT
    location,
    date,
    total_cases,
    total_deaths,
    (total_deaths / total_cases) * 100 AS Death_Percentage -- Calculate death percentage
```

FROM

project..CovidDeaths

WHERE

location = 'Pakistan' -- Filter for Pakistan

AND continent IS NOT NULL -- Ensure rows are countries, not continents/world aggregates

ORDER BY

location, date;

QUERY-3: COVID CONTAMINATION PERCENTAGE

Shows the percentage of the population that contracted COVID-19 over time for each location.

Final Output Fields: location, date, total_cases, population, Contamination_percentage.

SELECT

location,

date,

total_cases,

population,

(total_cases / population) * 100 AS Contamination_percentage -- Calculate contamination percentage

FROM

project..CovidDeaths

WHERE

continent IS NOT NULL -- Exclude continent/world aggregates

ORDER BY

location, date;

QUERY-4: COUNTRIES WITH HIGHEST CONTAMINATION RATE

Identifies countries with the highest COVID-19 contamination rate relative to their population.

Final Output Fields: location, max_cases, population, max_Contamination_percentage.

SELECT

location,

MAX(total_cases) AS max_cases, -- Find the peak total cases for each country

population,

MAX((total_cases / population) * 100) AS max_Contamination_percentage -- Find the peak contamination percentage

FROM

project..CovidDeaths

WHERE

continent IS NOT NULL -- Exclude continent/world aggregates

GROUP BY

location, population -- Group by country and its population

ORDER BY

max_Contamination_percentage DESC; -- Order by highest percentage first

QUERY-5: COUNTRIES WITH HIGHEST DEATH COUNT

Ranks countries based on the highest total number of COVID-19 deaths recorded.

Final Output Fields: location, max_deaths.

SELECT

location,

MAX(CAST(total_deaths AS INT)) AS max_deaths -- Find the peak total deaths (casting to INT for aggregation)

FROM

project..CovidDeaths

WHERE

continent IS NOT NULL -- Exclude continent/world aggregates

GROUP BY

location -- Group by country

ORDER BY

max_deaths DESC; -- Order by highest death count first

QUERY-6: CONTINENTS WITH HIGHEST DEATH COUNT

Ranks continents based on the highest total number of COVID-19 deaths recorded.

Final Output Fields: continent, max_deaths.

SELECT

continent,

MAX(CAST(total_deaths AS INT)) AS max_deaths -- Find the peak total deaths per continent

FROM

project..CovidDeaths

WHERE

continent IS NOT NULL -- Focus only on rows where continent is specified

GROUP BY

continent -- Group by continent

ORDER BY

max_deaths DESC; -- Order by highest death count first

QUERY-7: GLOBAL DAILY COVID NUMBERS

Calculates the total new cases, total new deaths, and the global death percentage for each date.

Final Output Fields: date, total_cases, total_deaths, death_percentage

SELECT

date,

SUM(new_cases) AS total_cases, -- Sum of new cases across all countries for the date

SUM(CAST(new_deaths AS INT)) AS total_deaths, -- Sum of new deaths (cast to INT) across all countries

(SUM(CAST(new_deaths AS INT)) / SUM(new_cases)) 100 AS death_percentage -- Calculate daily global death percentage

FROM

project..CovidDeaths

WHERE

continent IS NOT NULL -- Exclude continent/world aggregates from the sum

GROUP BY

date -- Group by date to get daily totals

ORDER BY

date, total_cases; -- Order by date

QUERY-8: POPULATION VS VACCINATIONS (USING CTE)

Calculates the rolling total of people vaccinated and the percentage of the population vaccinated over time for each country, using a Common Table Expression (CTE).

Final Output Fields: continent, location, date, population, new_vaccinations, rolling_people_vaccinated, vaccination_percentage (implicit from final SELECT).

```
WITH popsvvac (continent, location, date, population, new_vaccinations,  
rolling_people_vaccinated)
```

```
AS
```

```
(
```

```
-- CTE to calculate rolling vaccination count per country
```

```
SELECT
```

```
    d.continent,
```

```
    d.location,
```

```
    d.date,
```

```
    d.population,
```

```
    c.new_vaccinations,
```

```
    SUM(CONVERT(INT, c.new_vaccinations)) OVER (PARTITION BY d.location ORDER  
BY d.location, d.date) AS rolling_people_vaccinated -- Calculate rolling sum of  
vaccinations partitioned by location
```

```
FROM
```

```
    project.dbo.CovidDeaths d
```

```
JOIN
```

```
    project.dbo.CovidVaccinations c ON d.location = c.location AND d.date = c.date --  
Join deaths and vaccinations tables
```

```
WHERE
```

```
    d.continent IS NOT NULL -- Exclude continent/world aggregates
```

```
)
```

```
-- Final selection from the CTE, calculating vaccination percentage
```

```
SELECT
```

```
    ,
```

```
    (rolling_people_vaccinated / population) 100 AS vaccination_percentage -- Calculate  
percentage of population vaccinated
```

```
FROM
```

popsvvac

ORDER BY

location, date;

QUERY-9: POPULATION VS VACCINATIONS (USING TEMP TABLE)

Calculates the rolling total of people vaccinated and the percentage of the population vaccinated over time for each country, using a Temporary Table.

Final Output Fields: continent, location, date, population, new_vaccinations, rolling_people_vaccinated, vaccination_percentage (implicit from final SELECT).

-- Step 1: Drop the temporary table if it already exists

DROP TABLE IF EXISTS #PERCENTPOPULATIONVACCINATED;

-- Step 2: Create the temporary table structure

CREATE TABLE #PERCENTPOPULATIONVACCINATED

(

continent NVARCHAR(255),

location NVARCHAR(255),

date DATETIME,

population NUMERIC,

new_vaccinations NUMERIC,

rolling_people_vaccinated NUMERIC

); -- Step 3: Insert data into the temporary table, calculating the rolling vaccination count

INSERT INTO #PERCENTPOPULATIONVACCINATED

SELECT

d.continent,

d.location,

```

d.date,
d.population,
c.new_vaccinations,
SUM(CONVERT(INT, c.new_vaccinations)) OVER (PARTITION BY d.location ORDER BY
d.location, d.date) AS rolling_people_vaccinated -- Calculate rolling sum
FROM
    project.dbo.CovidDeaths d
JOIN
    project.dbo.CovidVaccinations c ON d.location = c.location AND d.date = c.date --
Join tables
WHERE
    d.continent IS NOT NULL; -- Exclude continent/world aggregates

-- Step 4: Select data from the temporary table and calculate the vaccination
percentage

SELECT
    ,
    (rolling_people_vaccinated / population) 100 AS vaccination_percentage -- Calculate
percentage
FROM
    #PERCENTPOPULATIONVACCINATED
ORDER BY
    location, date;

```


QUERY-10: VIEW CREATION FOR VACCINATION DATA

Creates a SQL View named 'PERCENTPOPULATIONVACCINATED' to store the calculation of rolling vaccinations per population for later use (e.g., in visualizations).

Final Output Fields (of the View): continent, location, date, population, new_vaccinations, rolling_people_vaccinated.

```
CREATE VIEW PERCENTPOPULATIONVACCINATED AS
```

```
SELECT
```

```
    d.continent,
```

```
    d.location,
```

```
    d.date,
```

```
    d.population,
```

```
    c.new_vaccinations,
```

```
    SUM(CONVERT(INT, c.new_vaccinations)) OVER (PARTITION BY d.location ORDER BY  
d.location, d.date) AS rolling_people_vaccinated -- Calculate rolling sum
```

```
FROM
```

```
    project.dbo.CovidDeaths d
```

```
JOIN
```

```
    project.dbo.CovidVaccinations c ON d.location = c.location AND d.date = c.date --  
Join tables
```

```
WHERE
```

```
    d.continent IS NOT NULL; -- Exclude continent/world aggregates
```