SQL queries used

Timeseries – Table

SELECT \* FROM vedant.timeseriesud;

select monthname(date), sum(confirmed) as cases from timeseriesud

group by monthname(date)

order by cases desc;

select month(date) as mnthnumber, monthname(date) as mnth, week(date) as weekdays, sum(confirmed) as confirmed , sum(recovered) as recovered, sum(deceased) as deaths, sum(tested) as tests

from timeseriesud

group by mnthnumber, mnth, weekdays

order by mnthnumber, weekdays

select week(date), month(date) from timeseriesud;

select state\_code, sum(deceased)\*100/sum(confirmed) as "Case Fatility Rate" from timeseriesud

group by state\_code

order by "Case Fatility Rate" desc;

State\_wise – Table

SELECT \* FROM vedant.state\_wise;

select State, total.tested from state\_wise

select State, TestingRatio , case when TestingRatio >= 0.05 or TestingRatio <= 0.1 then "Category A"

when TestingRatio >= 0.1 or TestingRatio <= 0.3 then "Category B"

when TestingRatio >= 0.3 or TestingRatio <= 0.5 then "Category C"

when TestingRatio >= 0.5 or TestingRatio <= 0.75 then "Category D"

when TestingRatio >= 0.75 or TestingRatio <= 1.0 then "Category E" else "Unknown"end as "Category" from cte

select state, (totalrecovered/totalconfirmed)\*100 as RecoveryRate from state\_wise

order by RecoveryRate desc;

District\_wise – Table

SELECT \* FROM vedant.districtud;

select \* , totaltested/population as TestingRatio, case when totaltested/population >= 0.05 and totaltested/population <= 0.1 then "Category A"

when totaltested/population >= 0.1 and totaltested/population <= 0.3 then "Category B"

when totaltested/population >= 0.3 and totaltested/population <= 0.5 then "Category C"

when totaltested/population >= 0.5 and totaltested/population <= 0.75 then "Category D"

when totaltested/population >= 0.75 and totaltested/population <= 1.0 then "Category E" else "Very Low" end as "TestingRatio Category" from districtud;

select state, district, (totalrecovered/totalconfirmed)\*100 as RecoveryRate from districtud;