Project Title - Unveiling COVID-19 Dynamics in India Through Excel Dashboard Analytics

BATCH - GP02 / DA04

DataSet - COVID-19

Section A

Project Overview — COVID-19 Data Analytics Project

- This project analyzes **India's COVID-19 pandemic data** to understand how the virus spread across different states and time periods.
- It combines multiple datasets covering **confirmed cases, recoveries, deaths, vaccination progress, and testing statistics**
- Perform **ETL (Extract, Transform, Load)** processes using SQL to clean, structure, and integrate COVID-19 data.
- The transformed data will be analyzed in Excel to identify **pandemic trends, vaccination impact, and testing effectiveness**.
- The final deliverable is an **interactive Excel dashboard** that visually explains COVID-19 patterns and supports data-driven insights.
- The project emphasizes real-world analytical skills such as data cleaning, trend analysis, KPI creation, and visual storytelling using pandemic data.

Objective — COVID-19 Data Analytics Project

- To understand and implement the **end-to-end data analytics process** using real COVID-19 datasets from India.
- To perform **data extraction, cleaning, and transformation** using SQL for accurate and reliable analysis.
- To integrate multiple datasets (cases, vaccinations, and testing) into a single structured format.
- To analyze pandemic trends such as **confirmed cases, recovery rates, death rates, testing activity, and vaccination progress** across states and time periods.
- To develop skills in **data visualization and dashboard creation** using Excel.
- To present insights through an **interactive dashboard** that supports clear storytelling.

Section 2: ETL Phase (SQL)

This section explains the complete ETL (Extract, Transform, Load) process performed using PostgreSQL to prepare the COVID-19 dataset for analysis. The main objective of this phase was to build a clean, reliable, and structured dataset by importing raw CSV files, cleaning the data, performing transformations, and generating a final summary table.

The entire process was done using PostgreSQL SQL queries.

Link:https://drive.google.com/file/d/127l6IVL5hDEpqm_mXuCM2p6ib0atlTEO/view?usp=sharing

The following three datasets were used:

- covid_19_india – Contains daily COVID cases
- covid_vaccine_statewise – Contains vaccination data
- StatewiseTestingDetails – Contains testing data

2.1 Extract Phase – Creating Tables and Loading Data

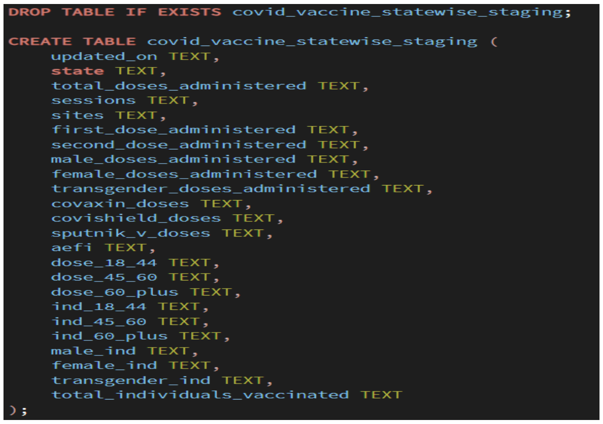
Creating Staging Tables

First, staging tables were created to load the raw CSV data exactly as it is. All columns were defined as TEXT because raw data may contain missing values, symbols, or inconsistent formats.

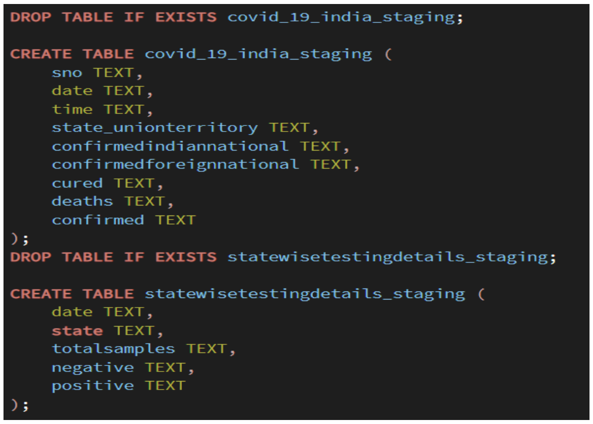
Three staging tables were created:

- covid_19_india_staging
- covid_vaccine_statewise_staging
- statewisetestingdetails_staging

These staging tables act as temporary storage for raw data.



```sql
DROP TABLE IF EXISTS covid_vaccine_statewise_staging;

CREATE TABLE covid_vaccine_statewise_staging (
    updated_on TEXT,
    state TEXT,
    total_doses_administered TEXT,
    sessions TEXT,
    sites TEXT,
    first_dose_administered TEXT,
    second_dose_administered TEXT,
    male_doses_administered TEXT,
    female_doses_administered TEXT,
    transgender_doses_administered TEXT,
    covaxin_doses TEXT,
    covishield_doses TEXT,
    sputnik_v_doses TEXT,
    aefi TEXT,
    dose_18_44 TEXT,
    dose_45_60 TEXT,
    dose_60_plus TEXT,
    ind_18_44 TEXT,
    ind_45_60 TEXT,
    ind_60_plus TEXT,
    male_ind TEXT,
    female_ind TEXT,
    transgender_ind TEXT,
    total_individuals_vaccinated TEXT
);
```



```sql
DROP TABLE IF EXISTS covid_19_india_staging;

CREATE TABLE covid_19_india_staging (
    sno TEXT,
    date TEXT,
    time TEXT,
    state_unionterritory TEXT,
    confirmedindiannational TEXT,
    confirmedforeignnational TEXT,
    cured TEXT,
    deaths TEXT,
    confirmed TEXT
);
DROP TABLE IF EXISTS statewisetestingdetails_staging;

CREATE TABLE statewisetestingdetails_staging (
    date TEXT,
    state TEXT,
    totalsamples TEXT,
    negative TEXT,
    positive TEXT
);
```

Creating Final Production Tables

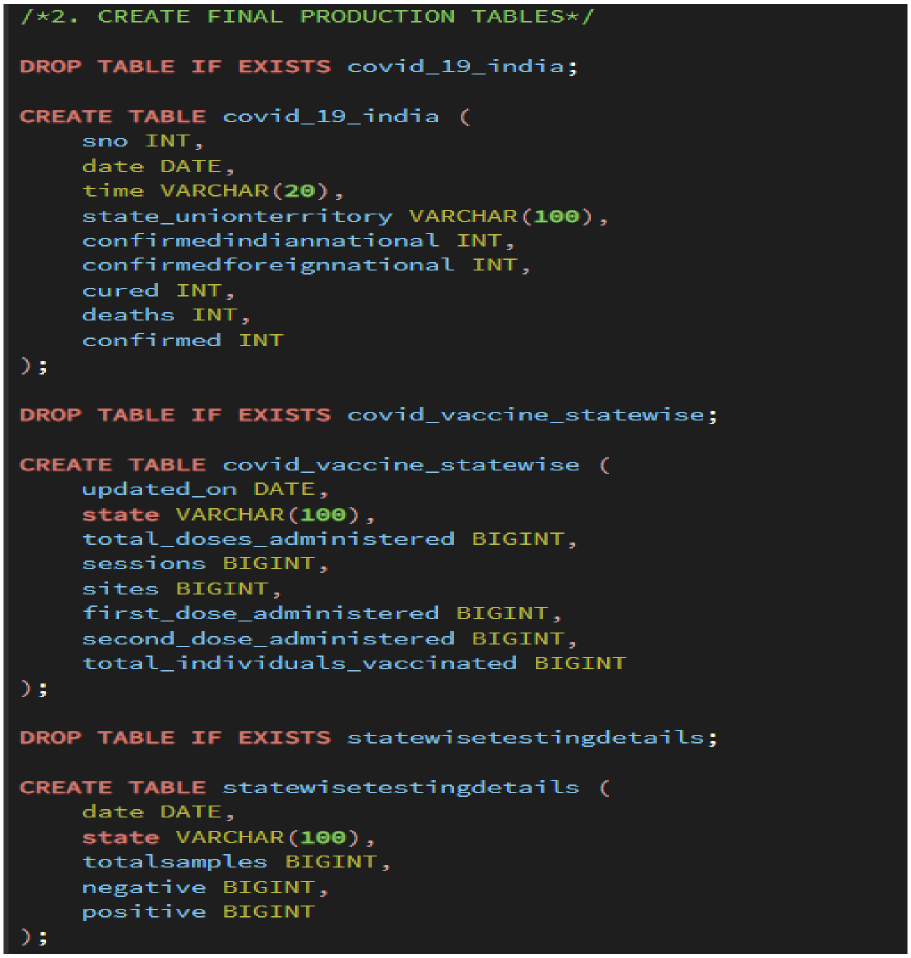
After staging tables, final production tables were created with proper data types such as:

- INT for numeric values
- DATE for date columns
- BIGINT for large numeric values
- VARCHAR for text fields

Production tables created:

- Covid_19_india
- Covid_vaccine_statewise
- statewisetestingdetails

These tables store cleaned and structured data.



```
/*2. CREATE FINAL PRODUCTION TABLES*/

DROP TABLE IF EXISTS covid_19_india;

CREATE TABLE covid_19_india (
    sno INT,
    date DATE,
    time VARCHAR(20),
    state_unionterritory VARCHAR(100),
    confirmedindiannational INT,
    confirmedforeignnational INT,
    cured INT,
    deaths INT,
    confirmed INT
);

DROP TABLE IF EXISTS covid_vaccine_statewise;

CREATE TABLE covid_vaccine_statewise (
    updated_on DATE,
    state VARCHAR(100),
    total_doses_administered BIGINT,
    sessions BIGINT,
    sites BIGINT,
    first_dose_administered BIGINT,
    second_dose_administered BIGINT,
    total_individuals_vaccinated BIGINT
);

DROP TABLE IF EXISTS statewisetestingdetails;

CREATE TABLE statewisetestingdetails (
    date DATE,
    state VARCHAR(100),
    totalsamples BIGINT,
    negative BIGINT,
    positive BIGINT
);
```

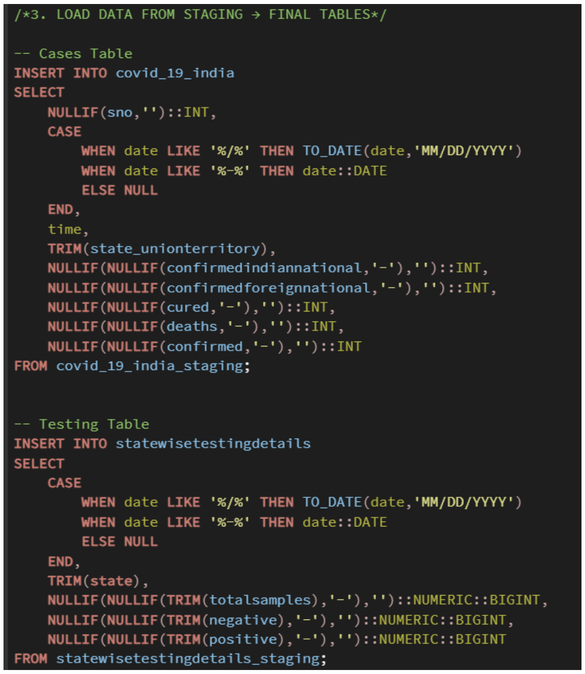
Loading Data into Final Tables

Data was loaded from staging tables into production tables using INSERT INTO SELECT statements.

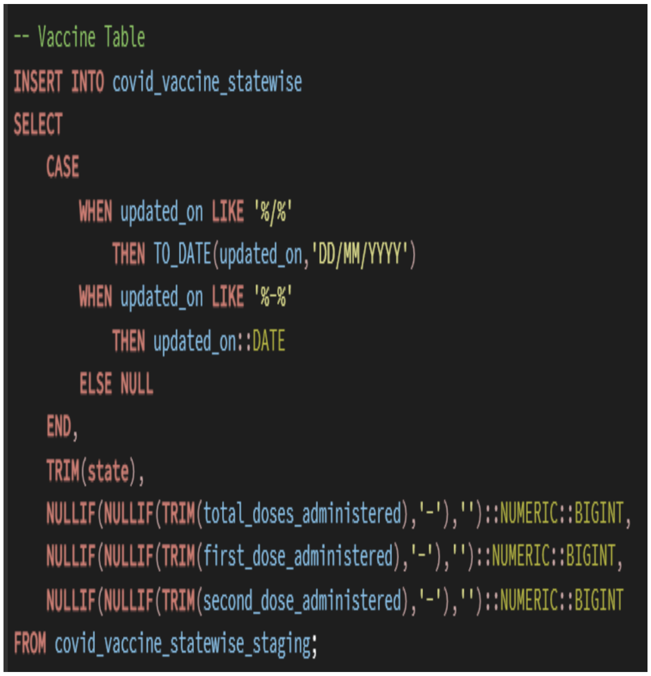
During this step:

- TEXT values were converted into correct numeric types
- Dates were converted into DATE format
- NULL and missing values were handled using NULLIF
- State names were trimmed using TRIM

This step ensures clean and structured data.



```
/*3. LOAD DATA FROM STAGING → FINAL TABLES*/

-- Cases Table
INSERT INTO covid_19_india
SELECT
    NULLIF(sno,'')::INT,
    CASE
        WHEN date LIKE '%/%' THEN TO_DATE(date,'MM/DD/YYYY')
        WHEN date LIKE '%-%' THEN date::DATE
        ELSE NULL
    END,
    time,
    TRIM(state_unionterritory),
    NULLIF(NULLIF(confirmedindiannational,'-'),'')::INT,
    NULLIF(NULLIF(confirmedforeignnational,'-'),'')::INT,
    NULLIF(NULLIF(cured,'-'),'')::INT,
    NULLIF(NULLIF(deaths,'-'),'')::INT,
    NULLIF(NULLIF(confirmed,'-'),'')::INT
FROM covid_19_india_staging;


-- Testing Table
INSERT INTO statewisetestingdetails
SELECT
    CASE
        WHEN date LIKE '%/%' THEN TO_DATE(date,'MM/DD/YYYY')
        WHEN date LIKE '%-%' THEN date::DATE
        ELSE NULL
    END,
    TRIM(state),
    NULLIF(NULLIF(TRIM(totalsamples),'-'),'')::NUMERIC::BIGINT,
    NULLIF(NULLIF(TRIM(negative),'-'),'')::NUMERIC::BIGINT,
    NULLIF(NULLIF(TRIM(positive),'-'),'')::NUMERIC::BIGINT
FROM statewisetestingdetails_staging;
```

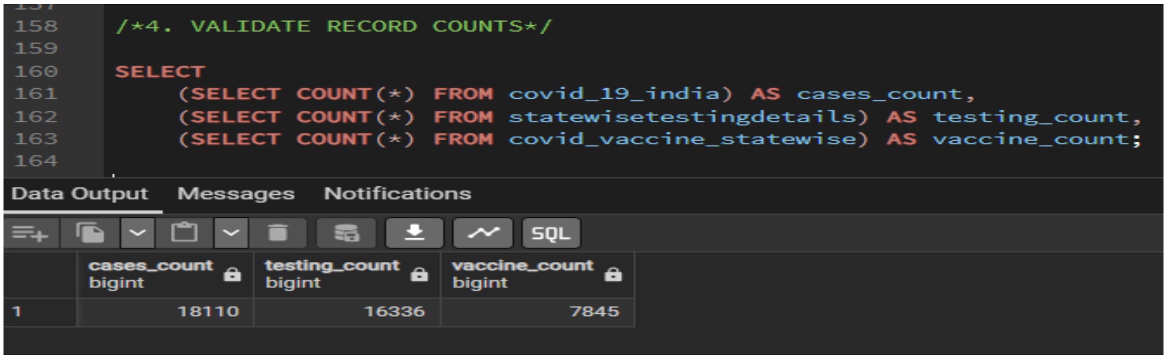


```
-- Vaccine Table
INSERT INTO covid_vaccine_statewise
SELECT
    CASE
        WHEN updated_on LIKE '%/%'
            THEN TO_DATE(updated_on,'DD/MM/YYYY')
        WHEN updated_on LIKE '%-%'
            THEN updated_on::DATE
        ELSE NULL
    END,
    TRIM(state),
    NULLIF(NULLIF(TRIM(total_doses_administered),'-'),'')::NUMERIC::BIGINT,
    NULLIF(NULLIF(TRIM(first_dose_administered),'-'),'')::NUMERIC::BIGINT,
    NULLIF(NULLIF(TRIM(second_dose_administered),'-'),'')::NUMERIC::BIGINT
FROM covid_vaccine_statewise_staging;
```

Data Validation

After loading the data, record counts were verified using COUNT(*) to ensure data was properly loaded. This confirms:

- No data loss
- Successful loading



```
/*4. VALIDATE RECORD COUNTS*/

SELECT
    (SELECT COUNT(*) FROM covid_19_india) AS cases_count,
    (SELECT COUNT(*) FROM statewisetestingdetails) AS testing_count,
    (SELECT COUNT(*) FROM covid_vaccine_statewise) AS vaccine_count;
```

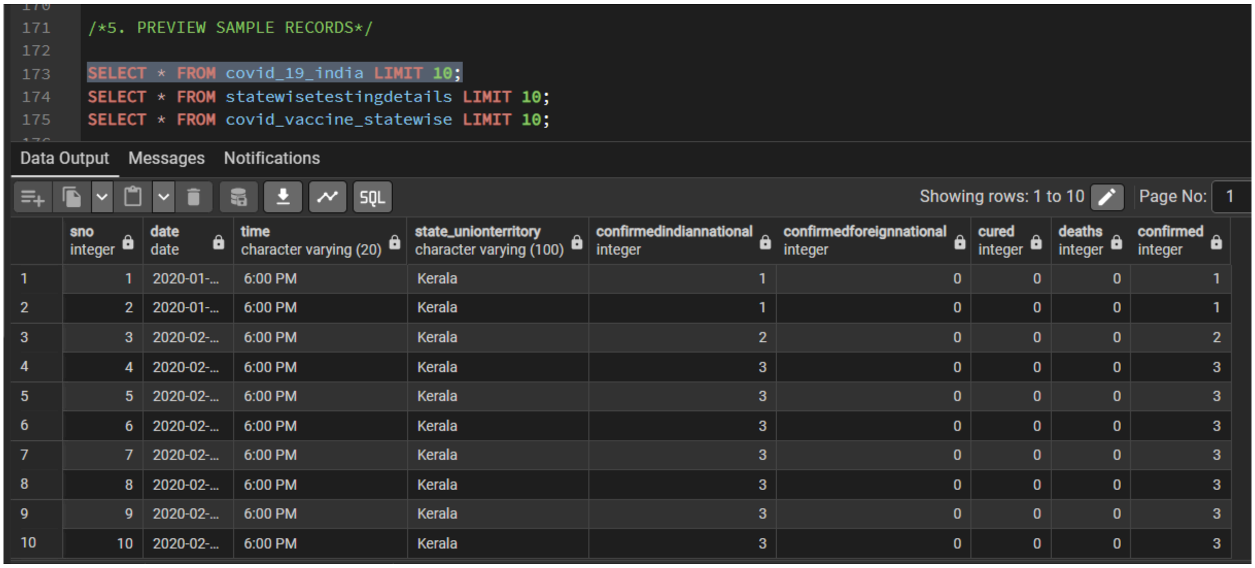
```
     cases_count   testing_count   vaccine_count
     bigint        bigint          bigint
1          18110           16336            7845
```

Data Preview

The first 10 rows of each table were displayed using SELECT * LIMIT 10.

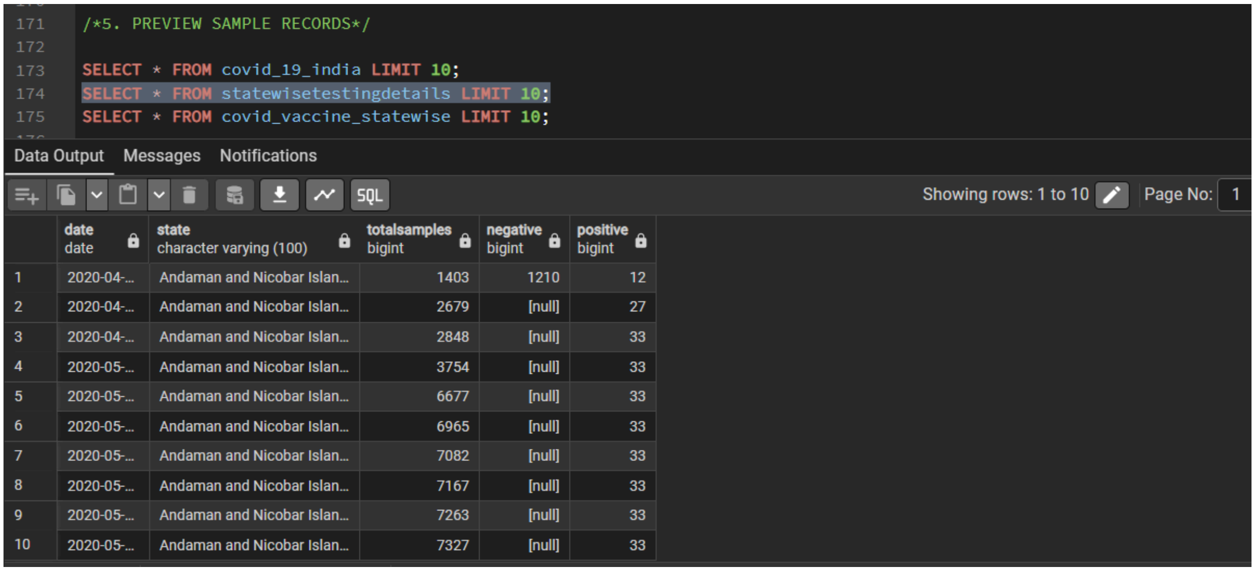
This was done to verify:

- Correct data loading
- Correct column values



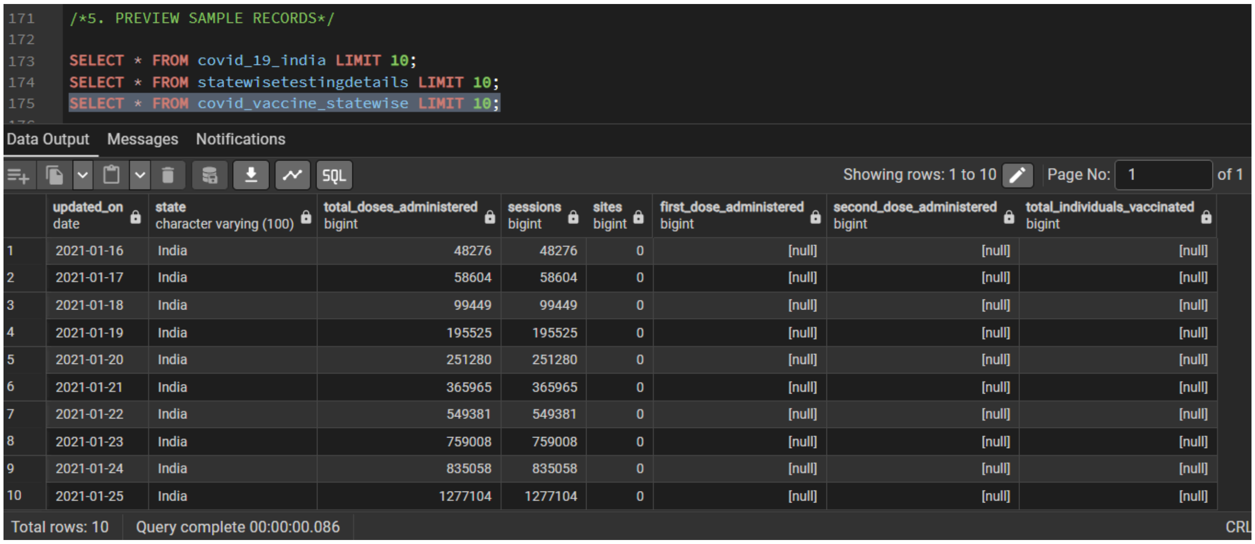
```sql
/*5. PREVIEW SAMPLE RECORDS*/

SELECT * FROM covid_19_india LIMIT 10;
SELECT * FROM statewisetestingdetails LIMIT 10;
SELECT * FROM covid_vaccine_statewise LIMIT 10;
```

	sno integer	date date	time character varying (20)	state_unionterritory character varying (100)	confirmedindiannational integer	confirmedforeignnational integer	cured integer	deaths integer	confirmed integer
1	1	2020-01-...	6:00 PM	Kerala	1	0	0	0	1
2	2	2020-01-...	6:00 PM	Kerala	1	0	0	0	1
3	3	2020-02-...	6:00 PM	Kerala	2	0	0	0	2
4	4	2020-02-...	6:00 PM	Kerala	3	0	0	0	3
5	5	2020-02-...	6:00 PM	Kerala	3	0	0	0	3
6	6	2020-02-...	6:00 PM	Kerala	3	0	0	0	3
7	7	2020-02-...	6:00 PM	Kerala	3	0	0	0	3
8	8	2020-02-...	6:00 PM	Kerala	3	0	0	0	3
9	9	2020-02-...	6:00 PM	Kerala	3	0	0	0	3
10	10	2020-02-...	6:00 PM	Kerala	3	0	0	0	3



```sql
/*5. PREVIEW SAMPLE RECORDS*/

SELECT * FROM covid_19_india LIMIT 10;
SELECT * FROM statewisetestingdetails LIMIT 10;
SELECT * FROM covid_vaccine_statewise LIMIT 10;
```

	date date	state character varying (100)	totalsamples bigint	negative bigint	positive bigint
1	2020-04-...	Andaman and Nicobar Islan...	1403	1210	12
2	2020-04-...	Andaman and Nicobar Islan...	2679	[null]	27
3	2020-04-...	Andaman and Nicobar Islan...	2848	[null]	33
4	2020-05-...	Andaman and Nicobar Islan...	3754	[null]	33
5	2020-05-...	Andaman and Nicobar Islan...	6677	[null]	33
6	2020-05-...	Andaman and Nicobar Islan...	6965	[null]	33
7	2020-05-...	Andaman and Nicobar Islan...	7082	[null]	33
8	2020-05-...	Andaman and Nicobar Islan...	7167	[null]	33
9	2020-05-...	Andaman and Nicobar Islan...	7263	[null]	33
10	2020-05-...	Andaman and Nicobar Islan...	7327	[null]	33



```
/*5. PREVIEW SAMPLE RECORDS*/

SELECT * FROM covid_19_india LIMIT 10;
SELECT * FROM statewisetestingdetails LIMIT 10;
SELECT * FROM covid_vaccine_statewise LIMIT 10;
```

	updated_on date	state character varying (100)	total_doses_administered bigint	sessions bigint	sites bigint	first_dose_administered bigint	second_dose_administered bigint	total_individuals_vaccinated bigint
1	2021-01-16	India	48276	48276	0	[null]	[null]	[null]
2	2021-01-17	India	58604	58604	0	[null]	[null]	[null]
3	2021-01-18	India	99449	99449	0	[null]	[null]	[null]
4	2021-01-19	India	195525	195525	0	[null]	[null]	[null]
5	2021-01-20	India	251280	251280	0	[null]	[null]	[null]
6	2021-01-21	India	365965	365965	0	[null]	[null]	[null]
7	2021-01-22	India	549381	549381	0	[null]	[null]	[null]
8	2021-01-23	India	759008	759008	0	[null]	[null]	[null]
9	2021-01-24	India	835058	835058	0	[null]	[null]	[null]
10	2021-01-25	India	1277104	1277104	0	[null]	[null]	[null]

Total rows: 10 Query complete 00:00:00.086

2.2 Data Cleaning Phase

State Name Standardization

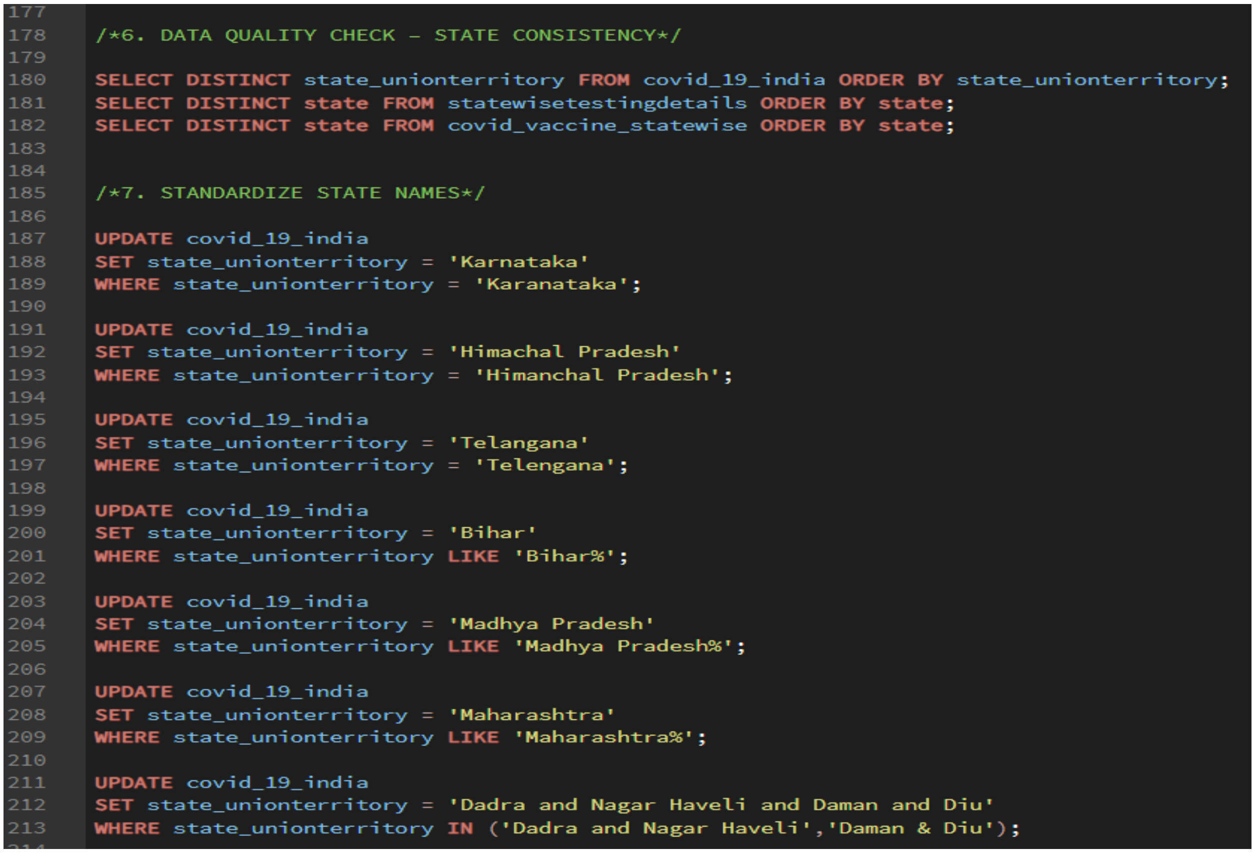
Some state names were inconsistent due to spelling differences.

Example issues:

- Karanataka → Karnataka
- Telengana → Telangana
- Himanchal Pradesh → Himachal Pradesh

These were corrected using UPDATE queries.

This ensures proper joining between tables.



```sql
/*6. DATA QUALITY CHECK - STATE CONSISTENCY*/

SELECT DISTINCT state_unionterritory FROM covid_19_india ORDER BY state_unionterritory;
SELECT DISTINCT state FROM statewisetestingdetails ORDER BY state;
SELECT DISTINCT state FROM covid_vaccine_statewise ORDER BY state;


/*7. STANDARDIZE STATE NAMES*/

UPDATE covid_19_india
SET state_unionterritory = 'Karnataka'
WHERE state_unionterritory = 'Karanataka';

UPDATE covid_19_india
SET state_unionterritory = 'Himachal Pradesh'
WHERE state_unionterritory = 'Himanchal Pradesh';

UPDATE covid_19_india
SET state_unionterritory = 'Telangana'
WHERE state_unionterritory = 'Telengana';

UPDATE covid_19_india
SET state_unionterritory = 'Bihar'
WHERE state_unionterritory LIKE 'Bihar%';

UPDATE covid_19_india
SET state_unionterritory = 'Madhya Pradesh'
WHERE state_unionterritory LIKE 'Madhya Pradesh%';

UPDATE covid_19_india
SET state_unionterritory = 'Maharashtra'
WHERE state_unionterritory LIKE 'Maharashtra%';

UPDATE covid_19_india
SET state_unionterritory = 'Dadra and Nagar Haveli and Daman and Diu'
WHERE state_unionterritory IN ('Dadra and Nagar Haveli','Daman & Diu');
```

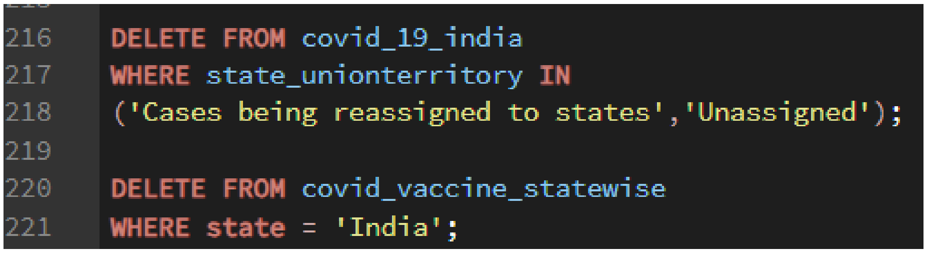
Removing Unnecessary Records

Some unwanted records like:

- Unassigned
- Cases being reassigned
- India (from vaccine dataset)

were removed using DELETE queries.

This ensures only valid state-level data is used.

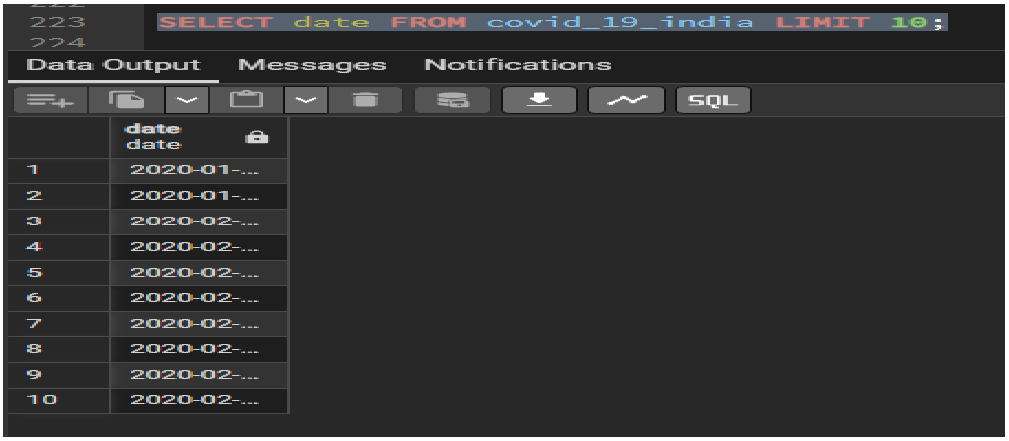


```sql
DELETE FROM covid_19_india
WHERE state_unionterritory IN
('Cases being reassigned to states','Unassigned');

DELETE FROM covid_vaccine_statewise
WHERE state = 'India';
```

Date Standardization

Dates were converted into standard format (YYYY-MM-DD) using the **TO_DATE function during table creation**.

This ensured that the date column was stored in the proper DATE data type, allowing accurate



```sql
SELECT date FROM covid_19_india LIMIT 10;
```

```
     date
     date
1    2020-01-...
2    2020-01-...
3    2020-02-...
4    2020-02-...
5    2020-02-...
6    2020-02-...
7    2020-02-...
8    2020-02-...
9    2020-02-...
10   2020-02-...
```

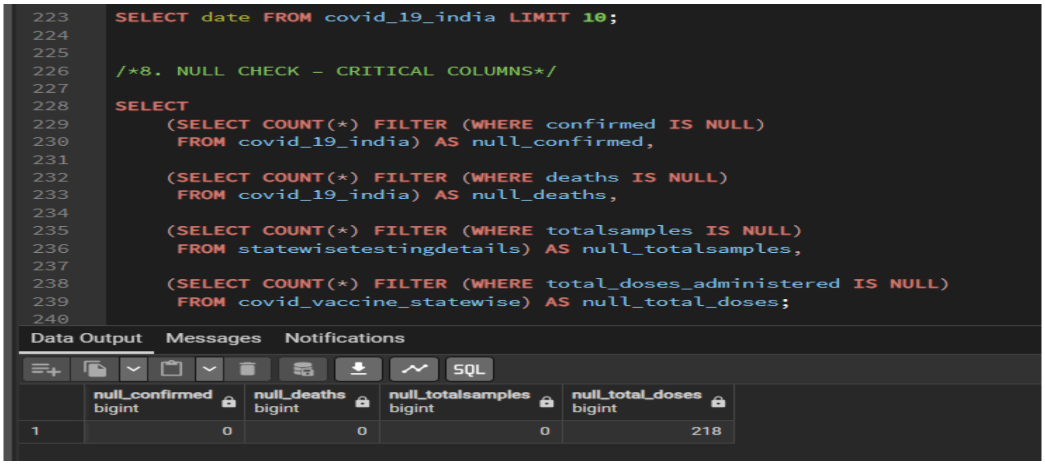
Missing Data Analysis

NULL values were checked in important columns such as:

- Confirmed
- Deaths
- TotalSamples
- Total Doses Administered

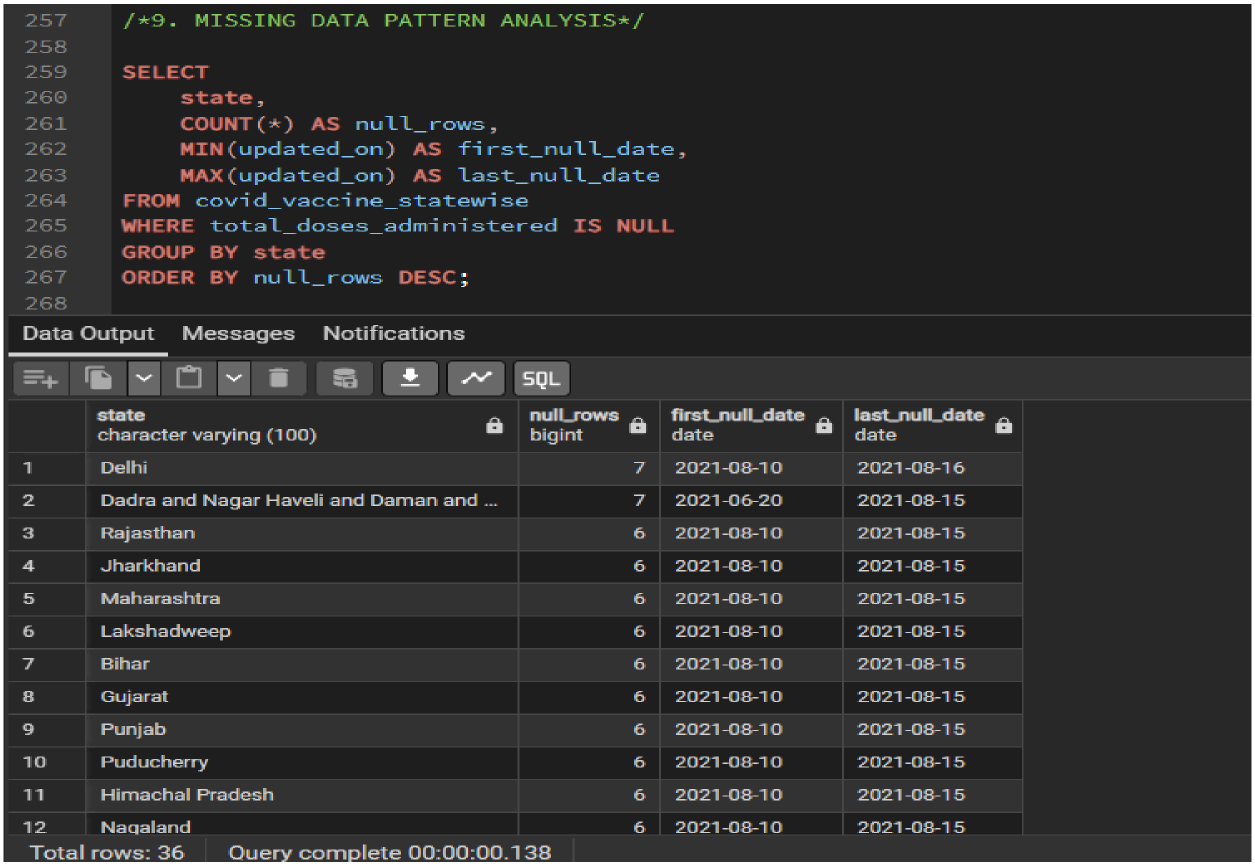
Missing values were found mainly in early pandemic stages where data was not recorded.

This was expected and handled during transformation.



```sql
SELECT date FROM covid_19_india LIMIT 10;


/*8. NULL CHECK - CRITICAL COLUMNS*/

SELECT
    (SELECT COUNT(*) FILTER (WHERE confirmed IS NULL)
     FROM covid_19_india) AS null_confirmed,

    (SELECT COUNT(*) FILTER (WHERE deaths IS NULL)
     FROM covid_19_india) AS null_deaths,

    (SELECT COUNT(*) FILTER (WHERE totalsamples IS NULL)
     FROM statewisetestingdetails) AS null_totalsamples,

    (SELECT COUNT(*) FILTER (WHERE total_doses_administered IS NULL)
     FROM covid_vaccine_statewise) AS null_total_doses;
```

```
    null_confirmed   null_deaths   null_totalsamples   null_total_doses
    bigint           bigint        bigint              bigint
1                0             0                   0                218
```



```sql
/*9. MISSING DATA PATTERN ANALYSIS*/

SELECT
    state,
    COUNT(*) AS null_rows,
    MIN(updated_on) AS first_null_date,
    MAX(updated_on) AS last_null_date
FROM covid_vaccine_statewise
WHERE total_doses_administered IS NULL
GROUP BY state
ORDER BY null_rows DESC;
```

	state character varying (100)	null_rows bigint	first_null_date date	last_null_date date
1	Delhi	7	2021-08-10	2021-08-16
2	Dadra and Nagar Haveli and Daman and ...	7	2021-06-20	2021-08-15
3	Rajasthan	6	2021-08-10	2021-08-15
4	Jharkhand	6	2021-08-10	2021-08-15
5	Maharashtra	6	2021-08-10	2021-08-15
6	Lakshadweep	6	2021-08-10	2021-08-15
7	Bihar	6	2021-08-10	2021-08-15
8	Gujarat	6	2021-08-10	2021-08-15
9	Punjab	6	2021-08-10	2021-08-15
10	Puducherry	6	2021-08-10	2021-08-15
11	Himachal Pradesh	6	2021-08-10	2021-08-15
12	Nagaland	6	2021-08-10	2021-08-15

Total rows: 36 Query complete 00:00:00.138

2.3 Transform Phase – Feature Engineering

This phase involved creating calculated metrics required for analysis.

Intermediate tables were created to perform transformations step-by-step.

Intermediate tables created:

- Covid_summary_base
- Covid_summary_with_testing
- covid_summary_with_vaccine

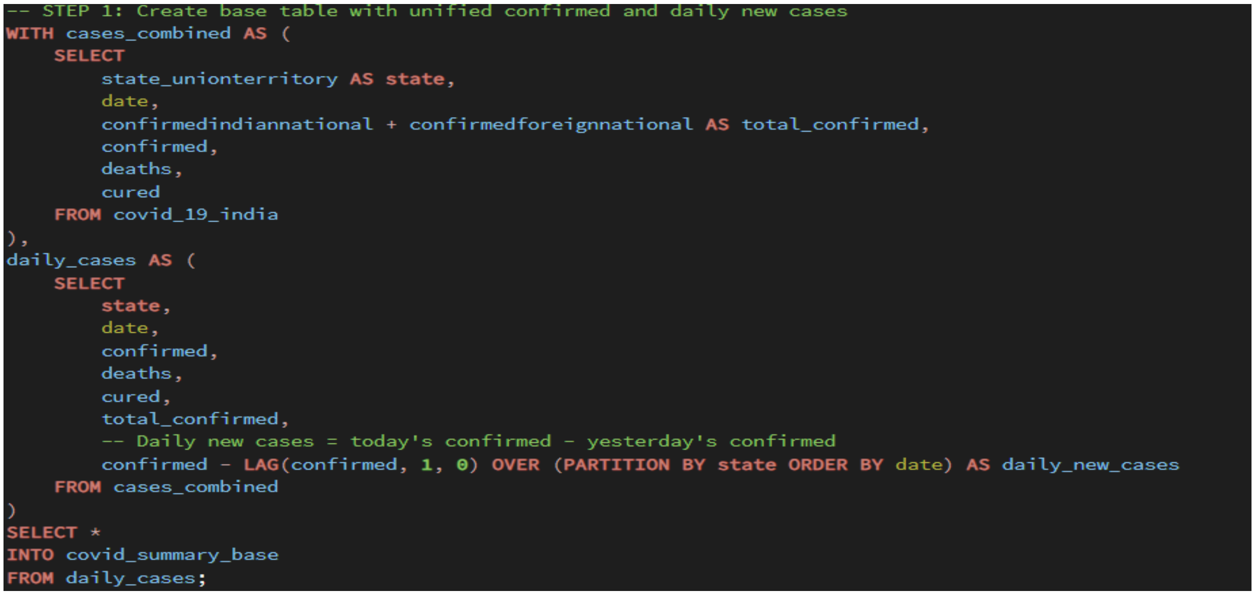
These tables helped in building the final dataset step-by-step and made the process organized and easier to debug.

Unified Confirmed Cases

ConfirmedIndianNational and ConfirmedForeignNational columns were combined into one column:

total_confirmed

This gives total confirmed cases.



```
-- STEP 1: Create base table with unified confirmed and daily new cases
WITH cases_combined AS (
    SELECT
        state_unionterritory AS state,
        date,
        confirmedindiannational + confirmedforeignnational AS total_confirmed,
        confirmed,
        deaths,
        cured
    FROM covid_19_india
),
daily_cases AS (
    SELECT
        state,
        date,
        confirmed,
        deaths,
        cured,
        total_confirmed,
        -- Daily new cases = today's confirmed - yesterday's confirmed
        confirmed - LAG(confirmed, 1, 0) OVER (PARTITION BY state ORDER BY date) AS daily_new_cases
    FROM cases_combined
)
SELECT *
INTO covid_summary_base
FROM daily_cases;
```

Daily New Cases Calculation

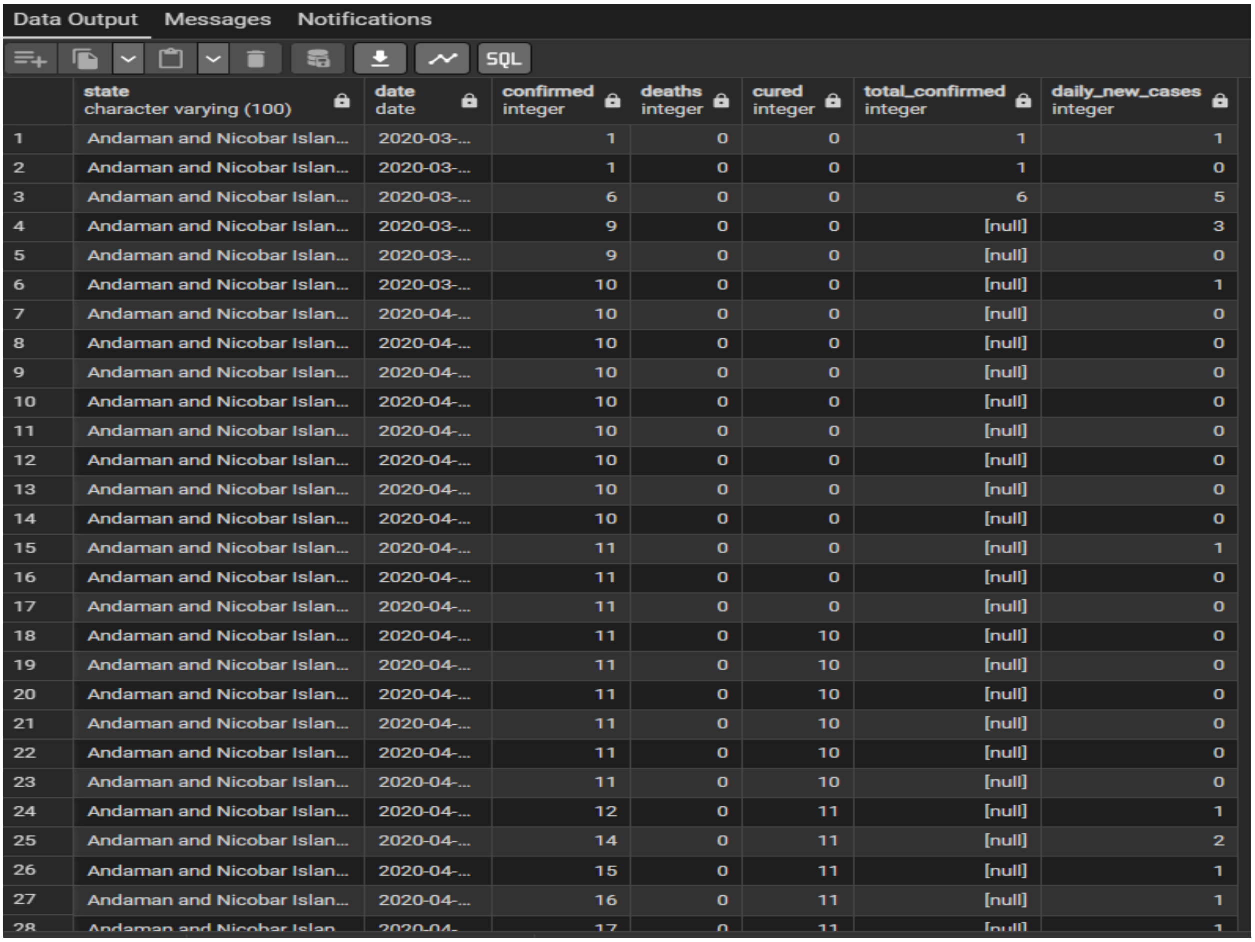
Daily new cases were calculated using LAG window function.

Formula:

Daily New Cases = Today's Confirmed − Previous Day Confirmed

Partition was done by the State.

This helps in analyzing daily growth.



	state character varying (100)	date date	confirmed integer	deaths integer	cured integer	total_confirmed integer	daily_new_cases integer
1	Andaman and Nicobar Islan...	2020-03-...	1	0	0	1	1
2	Andaman and Nicobar Islan...	2020-03-...	1	0	0	1	0
3	Andaman and Nicobar Islan...	2020-03-...	6	0	0	6	5
4	Andaman and Nicobar Islan...	2020-03-...	9	0	0	[null]	3
5	Andaman and Nicobar Islan...	2020-03-...	9	0	0	[null]	0
6	Andaman and Nicobar Islan...	2020-03-...	10	0	0	[null]	1
7	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
8	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
9	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
10	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
11	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
12	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
13	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
14	Andaman and Nicobar Islan...	2020-04-...	10	0	0	[null]	0
15	Andaman and Nicobar Islan...	2020-04-...	11	0	0	[null]	1
16	Andaman and Nicobar Islan...	2020-04-...	11	0	0	[null]	0
17	Andaman and Nicobar Islan...	2020-04-...	11	0	0	[null]	0
18	Andaman and Nicobar Islan...	2020-04-...	11	0	10	[null]	0
19	Andaman and Nicobar Islan...	2020-04-...	11	0	10	[null]	0
20	Andaman and Nicobar Islan...	2020-04-...	11	0	10	[null]	0
21	Andaman and Nicobar Islan...	2020-04-...	11	0	10	[null]	0
22	Andaman and Nicobar Islan...	2020-04-...	11	0	10	[null]	0
23	Andaman and Nicobar Islan...	2020-04-...	11	0	10	[null]	0
24	Andaman and Nicobar Islan...	2020-04-...	12	0	11	[null]	1
25	Andaman and Nicobar Islan...	2020-04-...	14	0	11	[null]	2
26	Andaman and Nicobar Islan...	2020-04-...	15	0	11	[null]	1
27	Andaman and Nicobar Islan...	2020-04-...	16	0	11	[null]	1
28	Andaman and Nicobar Islan...	2020-04-...	17	0	11	[null]	1

Big Join – Combining Testing Data

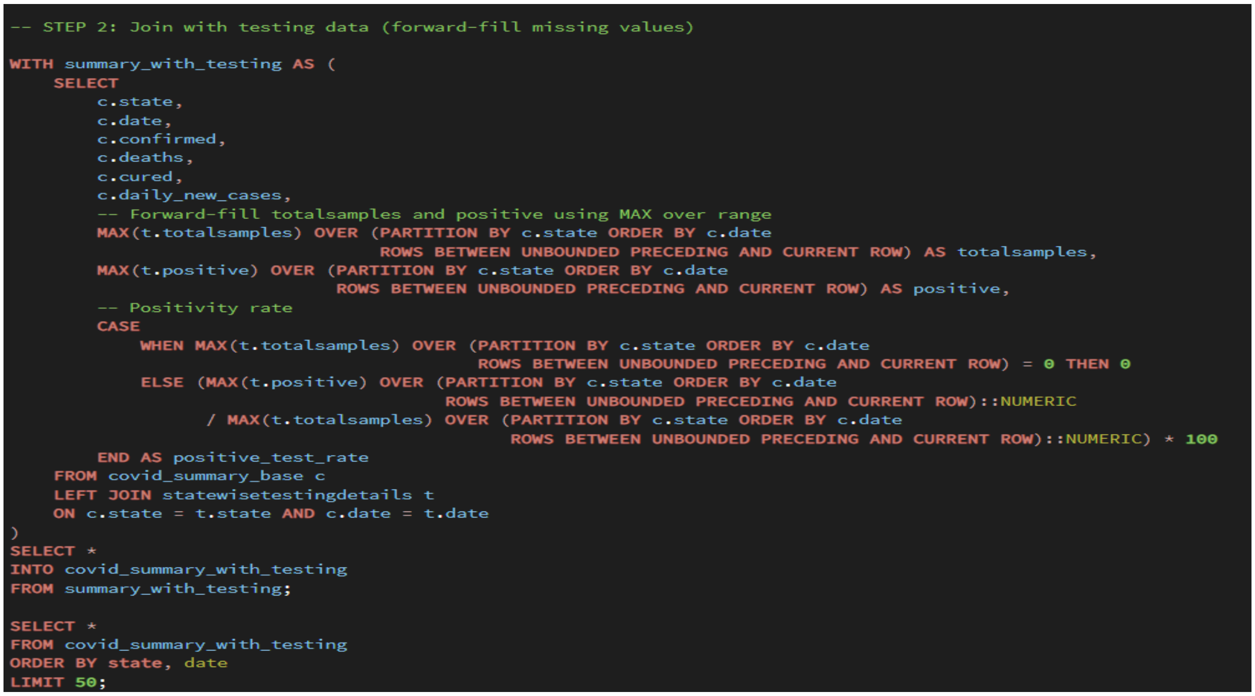
Testing data was joined using:

State and Date

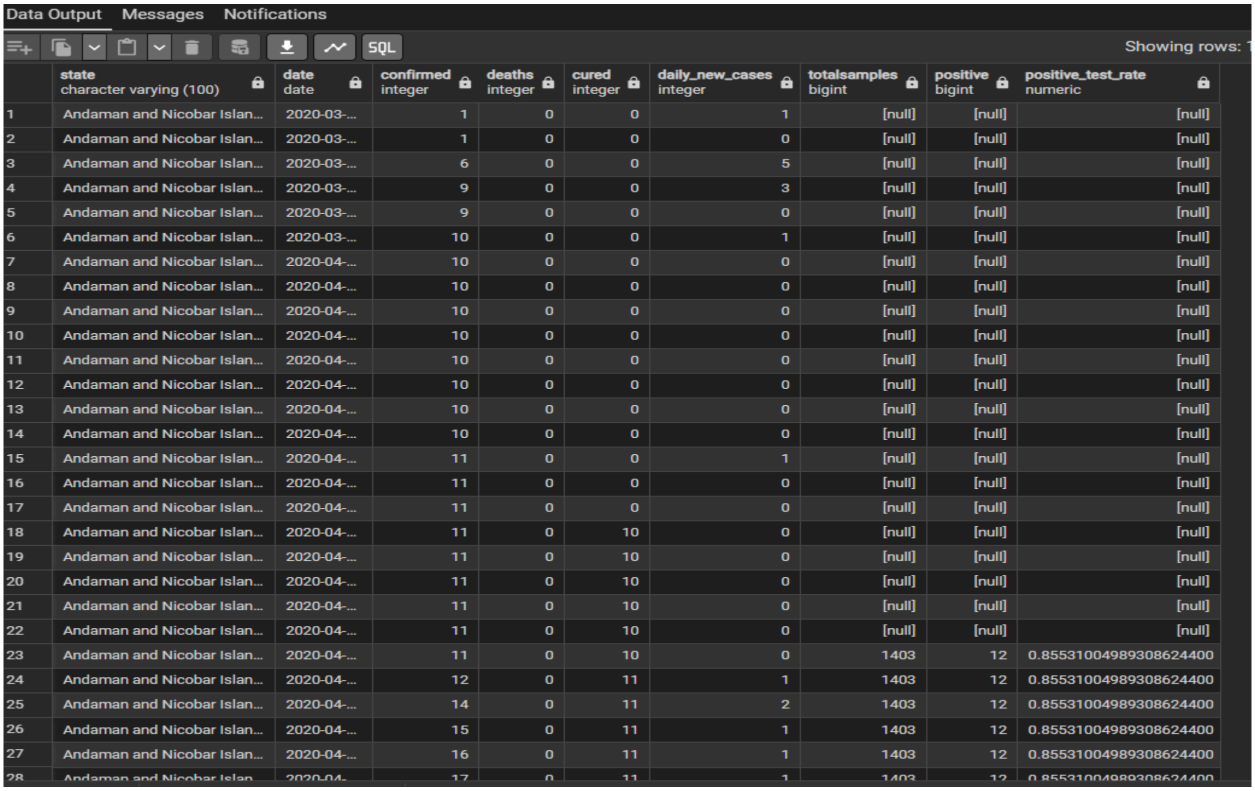
This added:

- TotalSamples
- positive

to the dataset.



```sql
-- STEP 2: Join with testing data (forward-fill missing values)

WITH summary_with_testing AS (
    SELECT
        c.state,
        c.date,
        c.confirmed,
        c.deaths,
        c.cured,
        c.daily_new_cases,
        -- Forward-fill totalsamples and positive using MAX over range
        MAX(t.totalsamples) OVER (PARTITION BY c.state ORDER BY c.date
                                  ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS totalsamples,
        MAX(t.positive) OVER (PARTITION BY c.state ORDER BY c.date
                              ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS positive,
        -- Positivity rate
        CASE
            WHEN MAX(t.totalsamples) OVER (PARTITION BY c.state ORDER BY c.date
                                           ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) = 0 THEN 0
            ELSE (MAX(t.positive) OVER (PARTITION BY c.state ORDER BY c.date
                                        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)::NUMERIC
                  / MAX(t.totalsamples) OVER (PARTITION BY c.state ORDER BY c.date
                                              ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)::NUMERIC) * 100
        END AS positive_test_rate
    FROM covid_summary_base c
    LEFT JOIN statewisetestingdetails t
    ON c.state = t.state AND c.date = t.date
)
SELECT *
INTO covid_summary_with_testing
FROM summary_with_testing;

SELECT *
FROM covid_summary_with_testing
ORDER BY state, date
LIMIT 50;
```



	state character varying (100)	date date	confirmed integer	deaths integer	cured integer	daily_new_cases integer	totalsamples bigint	positive bigint	positive_test_rate numeric
1	Andaman and Nicobar Islan...	2020-03-...	1	0	0	1	[null]	[null]	[null]
2	Andaman and Nicobar Islan...	2020-03-...	1	0	0	0	[null]	[null]	[null]
3	Andaman and Nicobar Islan...	2020-03-...	6	0	0	5	[null]	[null]	[null]
4	Andaman and Nicobar Islan...	2020-03-...	9	0	0	3	[null]	[null]	[null]
5	Andaman and Nicobar Islan...	2020-03-...	9	0	0	0	[null]	[null]	[null]
6	Andaman and Nicobar Islan...	2020-03-...	10	0	0	1	[null]	[null]	[null]
7	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
8	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
9	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
10	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
11	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
12	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
13	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
14	Andaman and Nicobar Islan...	2020-04-...	10	0	0	0	[null]	[null]	[null]
15	Andaman and Nicobar Islan...	2020-04-...	11	0	0	1	[null]	[null]	[null]
16	Andaman and Nicobar Islan...	2020-04-...	11	0	0	0	[null]	[null]	[null]
17	Andaman and Nicobar Islan...	2020-04-...	11	0	0	0	[null]	[null]	[null]
18	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
19	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
20	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
21	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
22	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
23	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	1403	12	0.85531004989308624400
24	Andaman and Nicobar Islan...	2020-04-...	12	0	11	1	1403	12	0.85531004989308624400
25	Andaman and Nicobar Islan...	2020-04-...	14	0	11	2	1403	12	0.85531004989308624400
26	Andaman and Nicobar Islan...	2020-04-...	15	0	11	1	1403	12	0.85531004989308624400
27	Andaman and Nicobar Islan...	2020-04-...	16	0	11	1	1403	12	0.85531004989308624400
28	Andaman and Nicobar Islan...	2020-04-...	17	0	11	1	1403	12	0.85531004989308624400

Forward Fill Handling

Testing and vaccination data were not available for every date.

So the forward fill method was used.

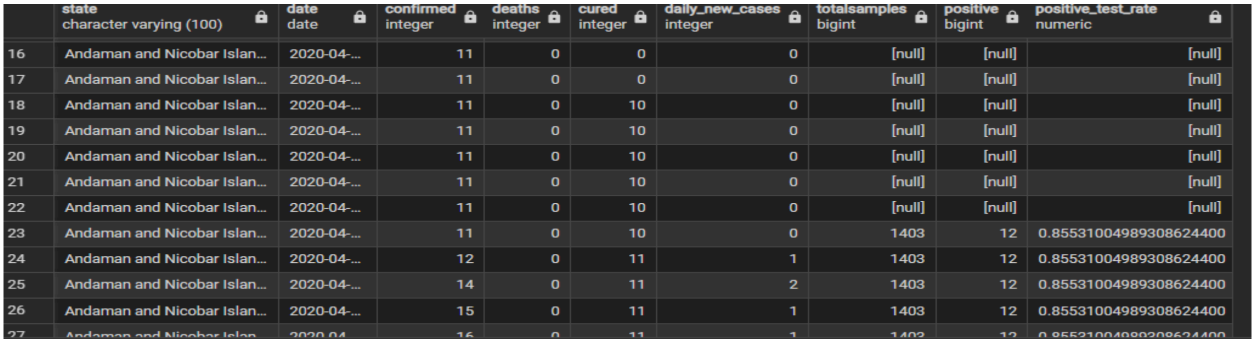
Forward fill means:

The previous available value is carried forward until a new value appears.

This was done using:

MAX() OVER (PARTITION BY state ORDER BY date)

This ensures continuous and complete data.

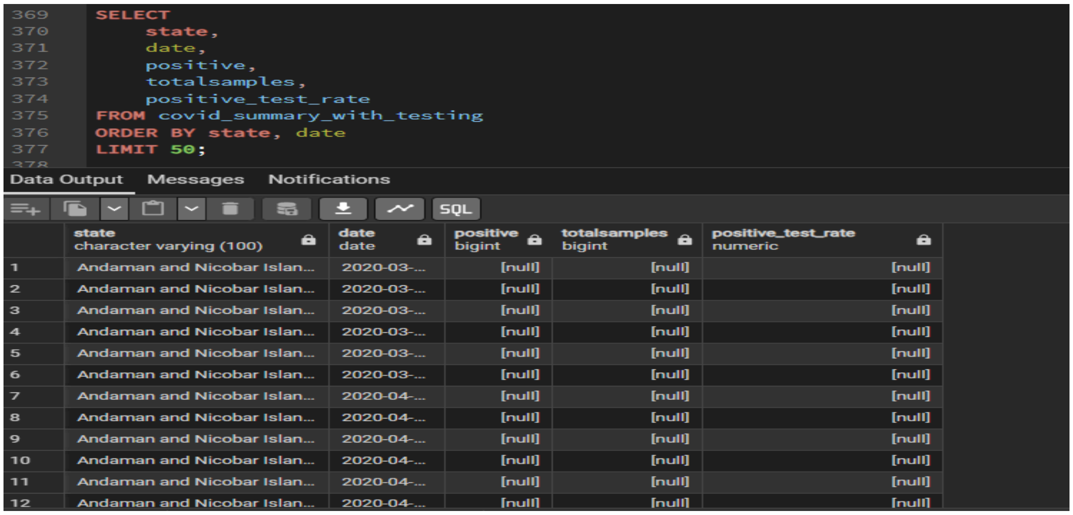


	state character varying (100)	date date	confirmed integer	deaths integer	cured integer	daily_new_cases integer	totalsamples bigint	positive bigint	positive_test_rate numeric
16	Andaman and Nicobar Islan...	2020-04-...	11	0	0	0	[null]	[null]	[null]
17	Andaman and Nicobar Islan...	2020-04-...	11	0	0	0	[null]	[null]	[null]
18	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
19	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
20	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
21	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
22	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	[null]	[null]	[null]
23	Andaman and Nicobar Islan...	2020-04-...	11	0	10	0	1403	12	0.85531004989308624400
24	Andaman and Nicobar Islan...	2020-04-...	12	0	11	1	1403	12	0.85531004989308624400
25	Andaman and Nicobar Islan...	2020-04-...	14	0	11	2	1403	12	0.85531004989308624400
26	Andaman and Nicobar Islan...	2020-04-...	15	0	11	1	1403	12	0.85531004989308624400
27	Andaman and Nicobar Islan...	2020-04-...	16	0	11	1	1403	12	0.85531004989308624400

Positive Test Rate Calculation

Formula: Positive Test Rate = Positive / TotalSamples × 100

This shows infection spread level.



```
SELECT
    state,
    date,
    positive,
    totalsamples,
    positive_test_rate
FROM covid_summary_with_testing
ORDER BY state, date
LIMIT 50;
```

	state character varying (100)	date date	positive bigint	totalsamples bigint	positive_test_rate numeric
1	Andaman and Nicobar Islan...	2020-03-...	[null]	[null]	[null]
2	Andaman and Nicobar Islan...	2020-03-...	[null]	[null]	[null]
3	Andaman and Nicobar Islan...	2020-03-...	[null]	[null]	[null]
4	Andaman and Nicobar Islan...	2020-03-...	[null]	[null]	[null]
5	Andaman and Nicobar Islan...	2020-03-...	[null]	[null]	[null]
6	Andaman and Nicobar Islan...	2020-03-...	[null]	[null]	[null]
7	Andaman and Nicobar Islan...	2020-04-...	[null]	[null]	[null]
8	Andaman and Nicobar Islan...	2020-04-...	[null]	[null]	[null]
9	Andaman and Nicobar Islan...	2020-04-...	[null]	[null]	[null]
10	Andaman and Nicobar Islan...	2020-04-...	[null]	[null]	[null]
11	Andaman and Nicobar Islan...	2020-04-...	[null]	[null]	[null]
12	Andaman and Nicobar Islan...	2020-04-...	[null]	[null]	[null]

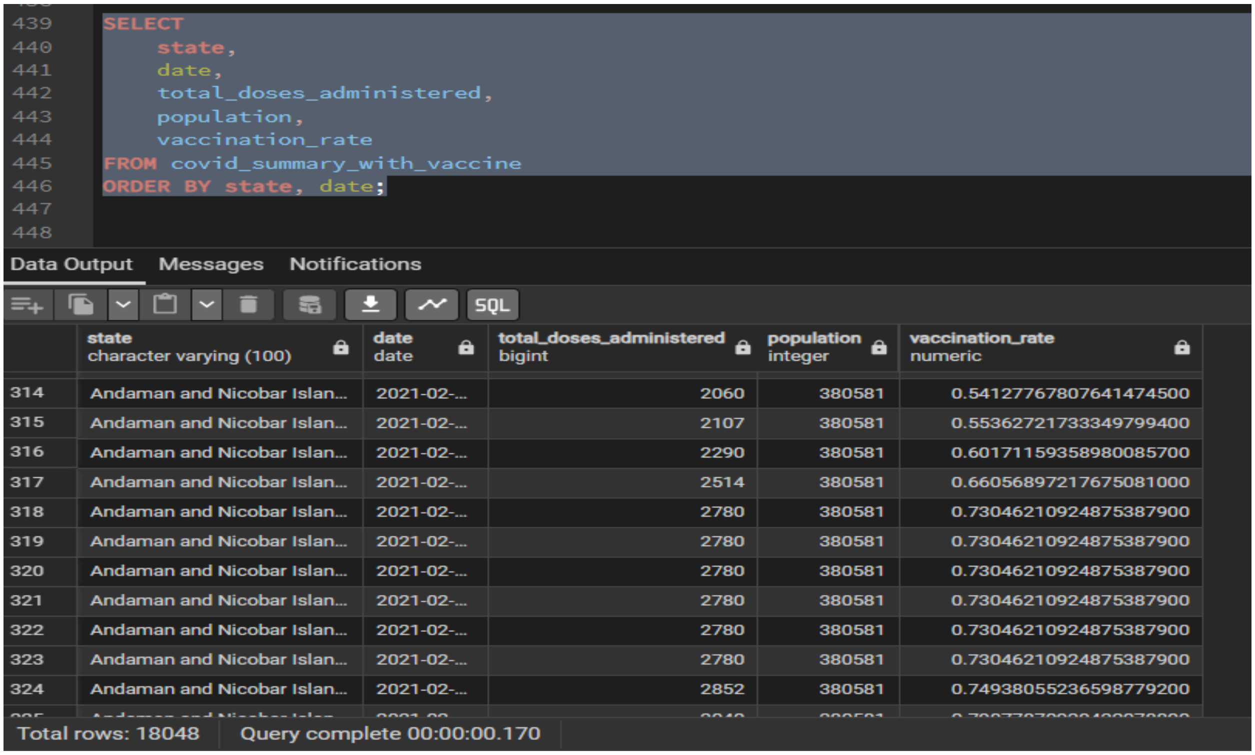
Vaccination Rate Calculation

Population data was added.

Formula:

Vaccination Rate = Total Doses / Population × 100

This shows vaccination progress.



```sql
SELECT
    state,
    date,
    total_doses_administered,
    population,
    vaccination_rate
FROM covid_summary_with_vaccine
ORDER BY state, date;
```

	state character varying (100)	date date	total_doses_administered bigint	population integer	vaccination_rate numeric
314	Andaman and Nicobar Islan...	2021-02-...	2060	380581	0.54127767807641474500
315	Andaman and Nicobar Islan...	2021-02-...	2107	380581	0.55362721733349799400
316	Andaman and Nicobar Islan...	2021-02-...	2290	380581	0.60171159358980085700
317	Andaman and Nicobar Islan...	2021-02-...	2514	380581	0.66056897217675081000
318	Andaman and Nicobar Islan...	2021-02-...	2780	380581	0.73046210924875387900
319	Andaman and Nicobar Islan...	2021-02-...	2780	380581	0.73046210924875387900
320	Andaman and Nicobar Islan...	2021-02-...	2780	380581	0.73046210924875387900
321	Andaman and Nicobar Islan...	2021-02-...	2780	380581	0.73046210924875387900
322	Andaman and Nicobar Islan...	2021-02-...	2780	380581	0.73046210924875387900
323	Andaman and Nicobar Islan...	2021-02-...	2780	380581	0.73046210924875387900
324	Andaman and Nicobar Islan...	2021-02-...	2852	380581	0.74938055236598779200

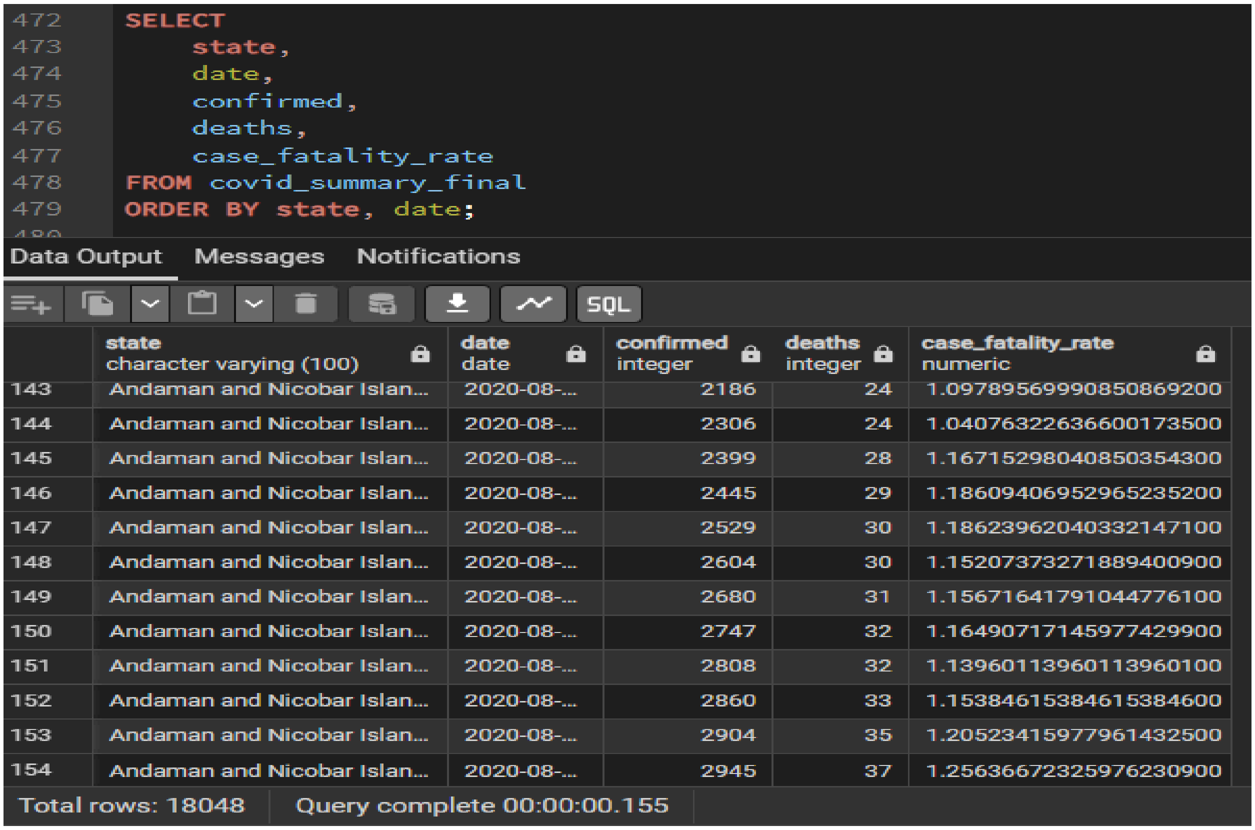
Total rows: 18048   Query complete 00:00:00.170

Case Fatality Rate Calculation

Formula:

Case Fatality Rate = Deaths / Confirmed × 100

This shows disease severity.



```sql
SELECT
    state,
    date,
    confirmed,
    deaths,
    case_fatality_rate
FROM covid_summary_final
ORDER BY state, date;
```

	state character varying (100)	date date	confirmed integer	deaths integer	case_fatality_rate numeric
143	Andaman and Nicobar Islan...	2020-08-...	2186	24	1.09789569990850869200
144	Andaman and Nicobar Islan...	2020-08-...	2306	24	1.04076322636600173500
145	Andaman and Nicobar Islan...	2020-08-...	2399	28	1.16715298040850354300
146	Andaman and Nicobar Islan...	2020-08-...	2445	29	1.18609406952965235200
147	Andaman and Nicobar Islan...	2020-08-...	2529	30	1.18623962040332147100
148	Andaman and Nicobar Islan...	2020-08-...	2604	30	1.15207373271889400900
149	Andaman and Nicobar Islan...	2020-08-...	2680	31	1.15671641791044776100
150	Andaman and Nicobar Islan...	2020-08-...	2747	32	1.16490717145977429900
151	Andaman and Nicobar Islan...	2020-08-...	2808	32	1.13960113960113960100
152	Andaman and Nicobar Islan...	2020-08-...	2860	33	1.15384615384615384600
153	Andaman and Nicobar Islan...	2020-08-...	2904	35	1.20523415977961432500
154	Andaman and Nicobar Islan...	2020-08-...	2945	37	1.25636672325976230900

Total rows: 18048 Query complete 00:00:00.155

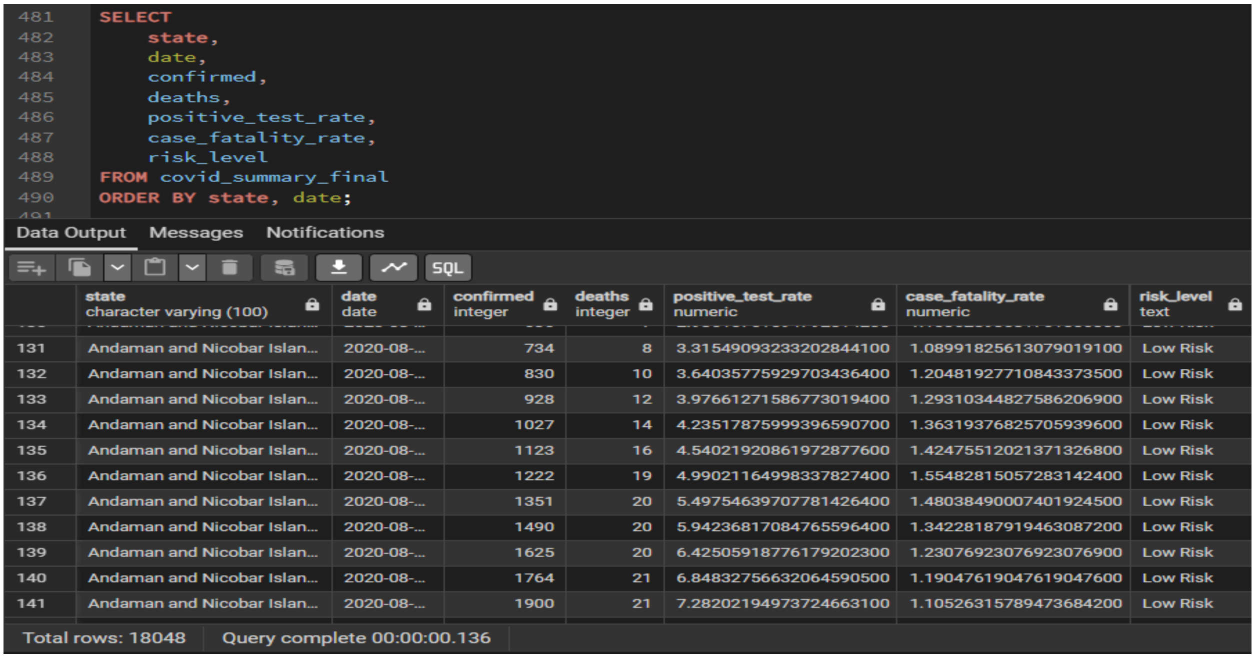
Risk Level Classification

Risk levels were created based on:

- High Risk
- Medium Risk
- Low Risk

Conditions were applied using CASE statements.

This helps identify dangerous states.



```sql
SELECT
    state,
    date,
    confirmed,
    deaths,
    positive_test_rate,
    case_fatality_rate,
    risk_level
FROM covid_summary_final
ORDER BY state, date;
```

	state character varying (100)	date date	confirmed integer	deaths integer	positive_test_rate numeric	case_fatality_rate numeric	risk_level text
131	Andaman and Nicobar Islan...	2020-08-...	734	8	3.31549093233202844100	1.08991825613079019100	Low Risk
132	Andaman and Nicobar Islan...	2020-08-...	830	10	3.64035775929703436400	1.20481927710843373500	Low Risk
133	Andaman and Nicobar Islan...	2020-08-...	928	12	3.97661271586773019400	1.29310344827586206900	Low Risk
134	Andaman and Nicobar Islan...	2020-08-...	1027	14	4.23517875999396590700	1.36319376825705939600	Low Risk
135	Andaman and Nicobar Islan...	2020-08-...	1123	16	4.54021920861972877600	1.42475512021371326800	Low Risk
136	Andaman and Nicobar Islan...	2020-08-...	1222	19	4.99021164998337827400	1.55482815057283142400	Low Risk
137	Andaman and Nicobar Islan...	2020-08-...	1351	20	5.49754639707781426400	1.48038490007401924500	Low Risk
138	Andaman and Nicobar Islan...	2020-08-...	1490	20	5.94236817084765596400	1.34228187919463087200	Low Risk
139	Andaman and Nicobar Islan...	2020-08-...	1625	20	6.42505918776179202300	1.23076923076923076900	Low Risk
140	Andaman and Nicobar Islan...	2020-08-...	1764	21	6.84832756632064590500	1.19047619047619047600	Low Risk
141	Andaman and Nicobar Islan...	2020-08-...	1900	21	7.28202194973724663100	1.10526315789473684200	Low Risk

Total rows: 18048 Query complete 00:00:00.136

2.5 Final Output Generation

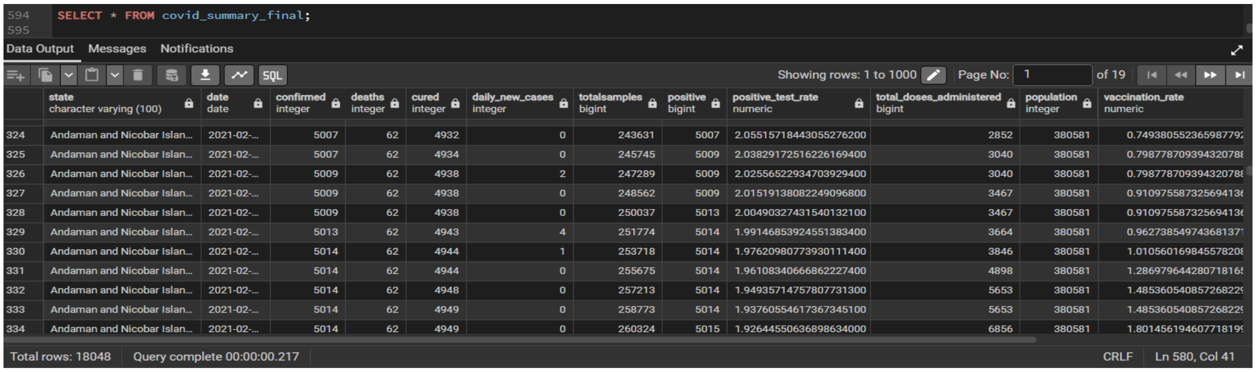
After completing all transformation steps, the final summary table `covid_summary_final` was created.

This table contains all required metrics for further analysis and visualization.

The final table includes the following columns:

- State
- Date
- Confirmed Cases
- Deaths
- Cured
- Daily New Cases
- Total Samples Tested
- Positive Cases
- Positive Test Rate
- Total Doses Administered
- Population
- Vaccination Rate
- Case Fatality Rate
- Risk Level

This table combines data from all three datasets into a single structured format.



```
594    SELECT * FROM covid_summary_final;
595
```

	state	date	confirmed	deaths	cured	daily_new_cases	totalsamples	positive	positive_test_rate	total_doses_administered	population	vaccination_rate
	character varying (100)	date	integer	integer	integer	integer	bigint	bigint	numeric	bigint	integer	numeric
324	Andaman and Nicobar Islan...	2021-02-...	5007	62	4932	0	243631	5007	2.05515718443055276200	2852	380581	0.749380552365987792
325	Andaman and Nicobar Islan...	2021-02-...	5007	62	4934	0	245745	5009	2.03829172516226169400	3040	380581	0.798778709394320788
326	Andaman and Nicobar Islan...	2021-02-...	5009	62	4938	2	247289	5009	2.02556522934703929400	3040	380581	0.798778709394320788
327	Andaman and Nicobar Islan...	2021-02-...	5009	62	4938	0	248562	5009	2.01519138082249096800	3467	380581	0.910975587325694136
328	Andaman and Nicobar Islan...	2021-02-...	5009	62	4938	0	250037	5013	2.00490327431540132100	3467	380581	0.910975587325694136
329	Andaman and Nicobar Islan...	2021-02-...	5013	62	4943	4	251774	5014	1.99146853924551383400	3664	380581	0.962738549743681371
330	Andaman and Nicobar Islan...	2021-02-...	5014	62	4944	1	253718	5014	1.97620980773930111400	3846	380581	1.010560169845578208
331	Andaman and Nicobar Islan...	2021-02-...	5014	62	4944	0	255675	5014	1.96108340666862227400	4898	380581	1.286979644280718165
332	Andaman and Nicobar Islan...	2021-02-...	5014	62	4948	0	257213	5014	1.94935714757807731300	5653	380581	1.485360540857268229
333	Andaman and Nicobar Islan...	2021-02-...	5014	62	4949	0	258773	5014	1.93760554617367345100	5653	380581	1.485360540857268229
334	Andaman and Nicobar Islan...	2021-02-...	5014	62	4949	0	260324	5015	1.92644550636898634000	6856	380581	1.801456194607718199

2.6 Data Export

The final table was exported into CSV format using PostgreSQL export functionality.

This file is named:

covid_summary_final.csv

This file will be used by the Excel Team for:

- Statistical Analysis
- Dashboard Creation
- Visualization
- Insight Generation

2.6 Observations

- Early stage data contains missing values
- Testing and vaccination started later
- Duplicate entries were found in raw data
- Forward fill was used to maintain continuous records

2.6 Conclusion of ETL Phase

The ETL process successfully completed the following:

- Created structured tables from raw CSV files
- Cleaned and standardized data
- Combined multiple datasets
- Calculated required metrics
- Generated final summary table
- Exported data for Excel analysis

This final dataset serves as the foundation for further statistical and dashboard analysis.

Section 3: Excel Analysis

1. Data Sources

Three main datasets were imported via `Data → From Text/CSV` and converted into structured tables (`Ctrl + T`) to make formula referencing and PivotTable analysis easier:

- **SummaryData:** Daily confirmed, recovered, deceased, and positive cases.
- **VaccineData:** Daily cumulative vaccination counts.
- **TestingData:** Daily state-wise testing sample counts.

2. Data Cleaning and Preparation

- **2.1 Date Normalization:** The dates were already formatted correctly from the SQL export, so the format was simply verified instead of requiring manual transformation.
- **2.2 Handling Missing Values:** `Go To Special → Blanks` was used to find empty numeric cells, which were then replaced with `0`. This prevented `#DIV/0!` calculation errors and ensured mathematical consistency.

3. Data Integration (Merging Testing Data)

To merge `TestingData` into `SummaryData` (mimicking a database `JOIN`), a composite key was created to uniquely identify each row:

- `=TRIM([@state]) & TEXT([@date],"dd-mm-yyyy")`

Next, the `TotalSamples` data was pulled into the main tracking table using `INDEX` and `MATCH`:

- `=INDEX(TestingData[TotalSamples], MATCH([@Key], TestingData[Key], 0))`

4. Data Validation (Recalculating Daily Cases)

To verify data accuracy, the daily new cases were recalculated to ensure they mathematically matched the running cumulative totals. The `OFFSET` function was utilized to compare each day's total to the previous day:

- `=IF([@state]=OFFSET([@state],-1,0), [@confirmed]-OFFSET([@confirmed],-1,0), 0)`

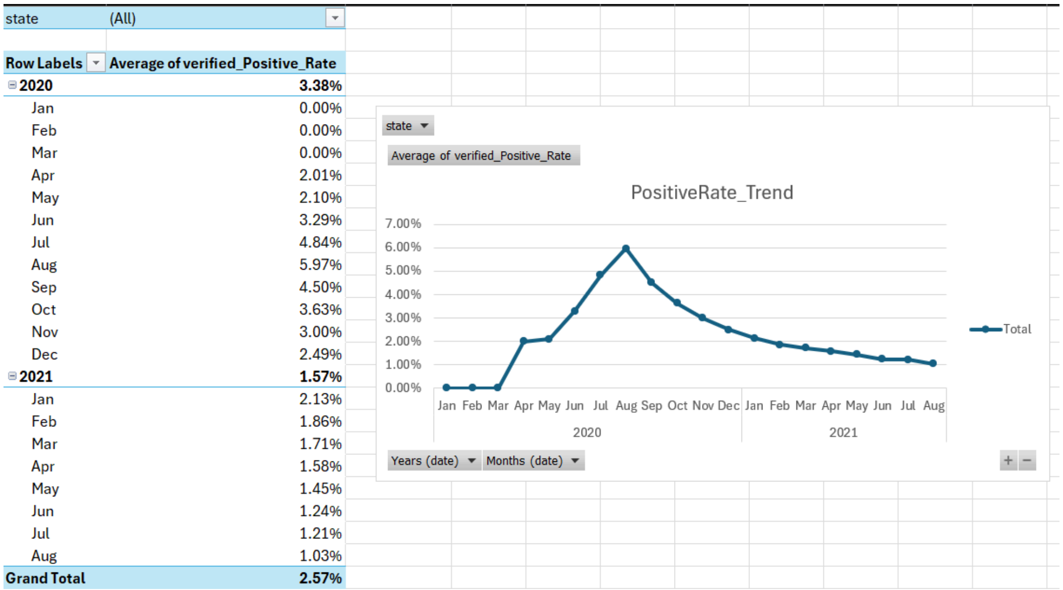
5. Key Metrics Calculation

- **5.1 Positive Rate:** Measures testing efficiency and how fast the infection is spreading.
    - `=IF([@TotalSamples]=0, 0, [@positive]/[@TotalSamples])`
- **5.2 Recovery Rate:** Shows the proportion of confirmed cases that successfully recovered.
    - `=IF([@confirmed]=0, 0, [@recovered]/[@confirmed])`
- **5.3 Cumulative Vaccination Rate:** Because vaccine data is a running total, the final recorded value represents total coverage.
    - *Pivot Setup:* `Rows → State` | `Values → MAX of Total Doses Administered`
    - *Calculation:* `MAX_Total_Doses / Population`

6. Visualizations and Interpretation

6.1 Positive Rate Trend (Line Chart)

Pivot grouped by Month-Year with Average Positive Rate. This chart identified peaks corresponding to pandemic waves.

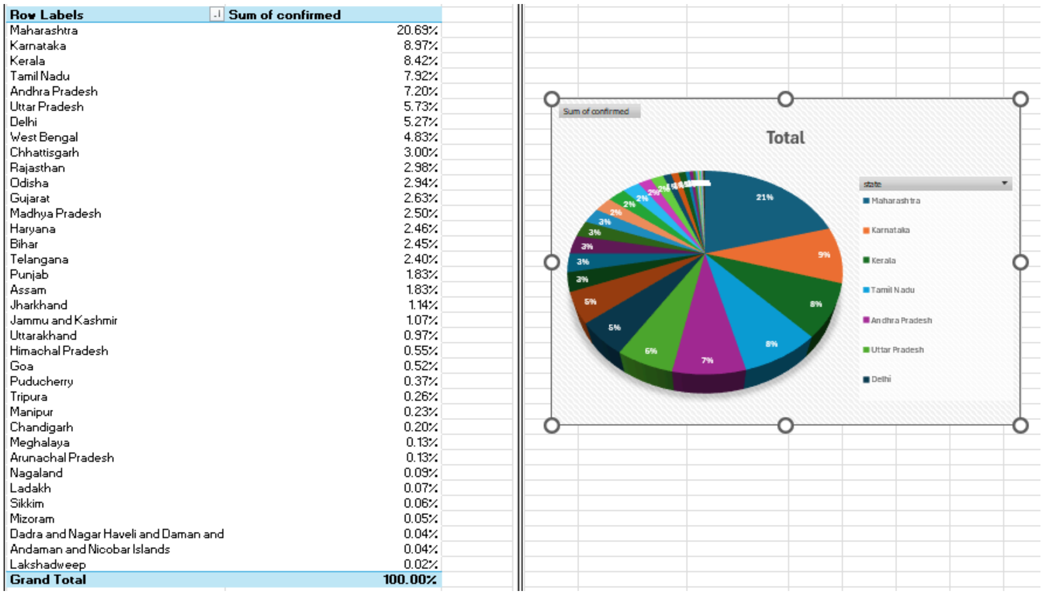


state	(All)
Row Labels	**Average of verified_Positive_Rate**
2020	**3.38%**
Jan	0.00%
Feb	0.00%
Mar	0.00%
Apr	2.01%
May	2.10%
Jun	3.29%
Jul	4.84%
Aug	5.97%
Sep	4.50%
Oct	3.63%
Nov	3.00%
Dec	2.49%
2021	**1.57%**
Jan	2.13%
Feb	1.86%
Mar	1.71%
Apr	1.58%
May	1.45%
Jun	1.24%
Jul	1.21%
Aug	1.03%
Grand Total	**2.57%**

state
Average of verified_Positive_Rate
PositiveRate_Trend
7.00%
6.00%
5.00%
4.00%
3.00%
2.00%
1.00%
0.00%
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2020
Jan Feb Mar Apr May Jun Jul Aug
2021
Total
Years (date)
Months (date)

6.2 State Contribution

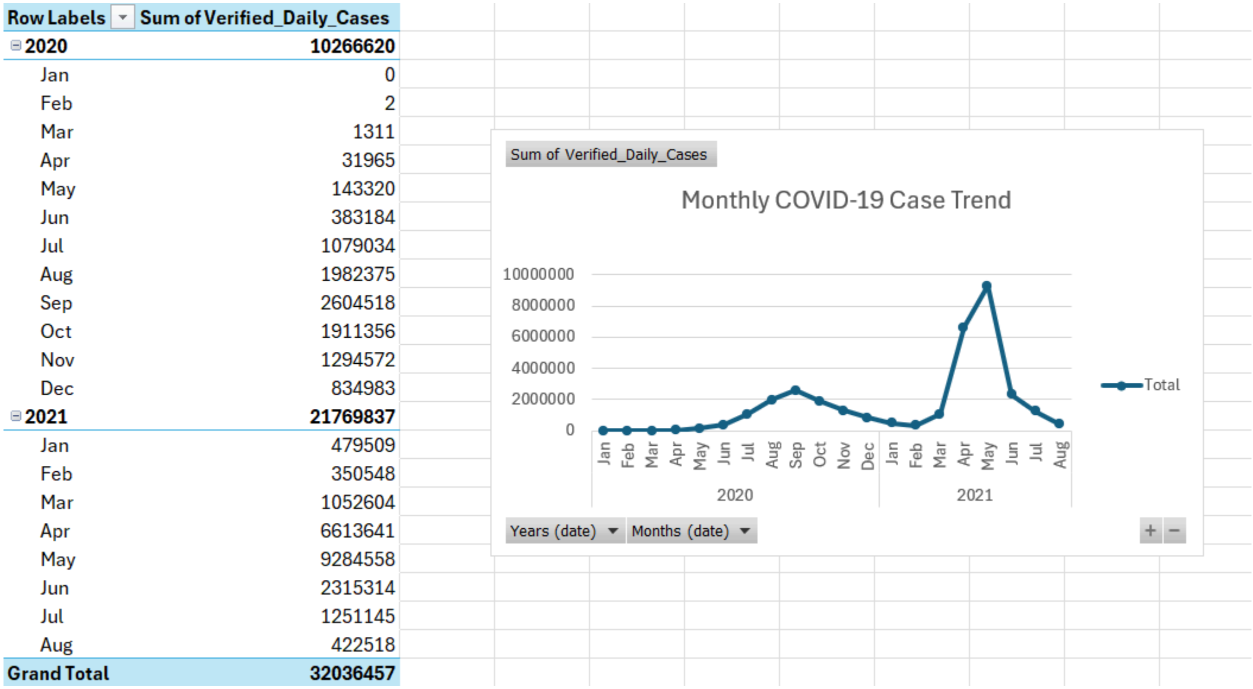
Pivot with MAX Confirmed cases displayed as % of Grand Total. This highlighted each state's contribution to total infections.



Row Labels	Sum of confirmed
Maharashtra	20.69%
Karnataka	8.97%
Kerala	8.42%
Tamil Nadu	7.92%
Andhra Pradesh	7.20%
Uttar Pradesh	5.73%
Delhi	5.27%
West Bengal	4.83%
Chhattisgarh	3.00%
Rajasthan	2.98%
Odisha	2.94%
Gujarat	2.63%
Madhya Pradesh	2.50%
Haryana	2.46%
Bihar	2.45%
Telangana	2.40%
Punjab	1.83%
Assam	1.83%
Jharkhand	1.14%
Jammu and Kashmir	1.07%
Uttarakhand	0.97%
Himachal Pradesh	0.55%
Goa	0.52%
Puducherry	0.37%
Tripura	0.26%
Manipur	0.23%
Chandigarh	0.20%
Meghalaya	0.13%
Arunachal Pradesh	0.13%
Nagaland	0.09%
Ladakh	0.07%
Sikkim	0.06%
Mizoram	0.05%
Dadra and Nagar Haveli and Daman and	0.04%
Andaman and Nicobar Islands	0.04%
Lakshadweep	0.02%
Grand Total	**100.00%**

6.3 Monthly Case Trend (Line Chart)

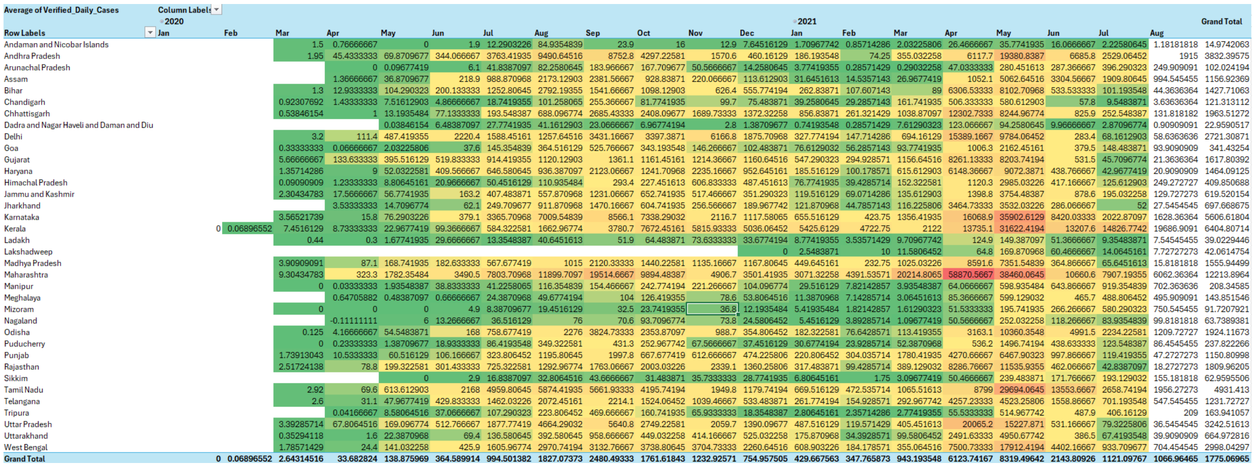
Pivot grouped by Month-Year using SUM of Verified_Daily_Cases. Clear waves were visible over time.



Row Labels	Sum of Verified_Daily_Cases
⊟ 2020	**10266620**
Jan	0
Feb	2
Mar	1311
Apr	31965
May	143320
Jun	383184
Jul	1079034
Aug	1982375
Sep	2604518
Oct	1911356
Nov	1294572
Dec	834983
⊟ 2021	**21769837**
Jan	479509
Feb	350548
Mar	1052604
Apr	6613641
May	9284558
Jun	2315314
Jul	1251145
Aug	422518
Grand Total	**32036457**

6.4 Heatmap (Average Monthly Cases)

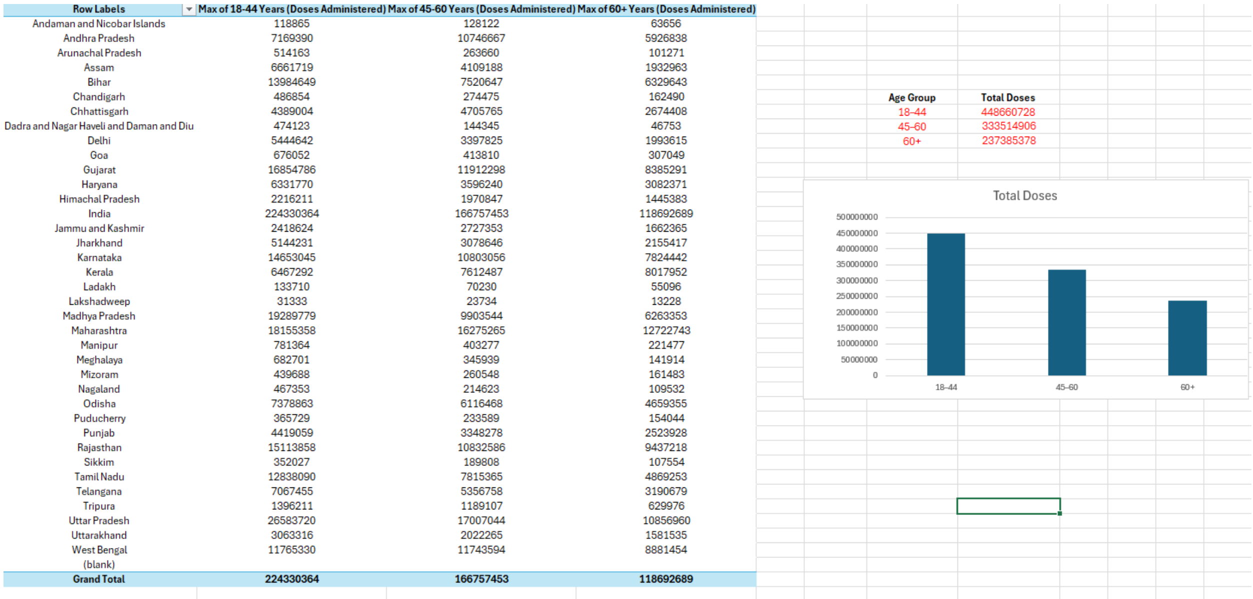
Pivot with Rows → State and Columns → Month-Year using Average Daily Cases. Conditional formatting color scale created a heatmap to highlight hotspots.



Average of Verified_Daily_Cases	Column Labels																			
	2020											2021								Grand Total
Row Labels	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	
Andaman and Nicobar Islands			1.5	0.76666667	0	1.9	12.2903226	84.9354839	23.9	16	12.9	7.64516129	1.70967742	0.85714286	2.03225806	26.4666667	35.7741935	16.0666667	2.22580645	1.18181818	14.9742063
Andhra Pradesh			1.95	45.4333333	69.8709677	344.066667	3763.41935	9490.64516	8752.8	4297.22581	1570.6	460.16129	186.193548	74.25	355.032258	6117.7	19380.8387	6685.8	2529.06452	1915	3832.39575
Arunachal Pradesh				0	0.09677419	6.1	41.8387097	82.2580645	183.966667	167.709677	50.5666667	14.2580645	3.77419355	0.28571429	0.29032258	47.0333333	280.451613	287.366667	396.290323	249.909091	102.024194
Assam				1.36666667	36.8709677	218.9	988.870968	2173.12903	2381.56667	928.83871	220.066667	113.612903	31.6451613	14.5357143	26.9677419	1052.1	5062.64516	3304.56667	1909.80645	994.545455	1156.92369
Bihar			1.3	12.9333333	104.290323	200.133333	1252.80645	2792.19355	1541.66667	1098.12903	626.4	555.774194	262.83871	107.607143	89	6306.53333	8102.70968	533.533333	101.193548	44.3636364	1427.71063
Chandigarh			0.92307692	1.43333333	7.51612903	4.86666667	18.7419355	101.258065	255.366667	81.7741935	99.7	75.483871	39.2580645	29.2857143	161.741935	506.333333	580.612903	57.8	9.5483871	3.63636364	121.313112
Chhattisgarh			0.53846154	1	13.1935484	77.1333333	193.548387	688.096774	2685.43333	2408.09677	1689.73333	1372.32258	856.83871	261.321429	1038.87097	12302.7333	8244.96774	825.9	252.548387	131.818182	1963.51272
Dadra and Nagar Haveli and Daman and Diu					0.03846154	6.48387097	27.7741935	41.1612903	23.0666667	6.96774194	2.8	1.38709677	0.74193548	0.28571429	7.61290323	123.066667	94.2580645	9.96666667	2.87096774	0.90909091	22.9590517
Delhi			3.2	111.4	487.419355	2220.4	1588.45161	1257.64516	3431.16667	3397.3871	6166.8	1875.70968	327.774194	147.714286	694.16129	15389.1667	9784.06452	283.4	68.1612903	58.6363636	2721.30871
Goa			0.33333333	0.06666667	2.03225806	37.6	145.354839	364.516129	525.766667	343.193548	146.266667	102.483871	76.6129032	56.2857143	93.7741935	1006.3	2162.45161	379.5	148.483871	93.9090909	341.43254
Gujarat			5.66666667	133.633333	395.516129	519.833333	914.419355	1120.12903	1361.1	1161.45161	1214.36667	1160.64516	547.290323	294.928571	1156.64516	8261.13333	8203.74194	531.5	45.7096774	21.3636364	1617.80392
Haryana			1.35714286	9	52.0322581	409.566667	646.580645	936.387097	2123.06667	1241.70968	2235.16667	952.645161	185.516129	100.178571	615.612903	6148.36667	9072.3871	438.766667	42.9677419	20.9090909	1464.09125
Himachal Pradesh			0.09090909	1.23333333	8.80645161	20.9666667	50.4516129	110.935484	293.4	227.451613	606.833333	487.451613	76.7741935	39.4285714	152.322581	1120.3	2985.03226	417.166667	125.612903	249.272727	409.850688
Jammu and Kashmir			2.30434783	17.5666667	56.7741935	163.2	407.483871	557.870968	1231.06667	652.741935	517.466667	351.290323	119.516129	69.0714286	135.612903	1398.8	3754.48387	878.6	195.032258	129.727273	619.520154
Jharkhand				3.53333333	14.7096774	62.1	249.709677	911.870968	1470.16667	604.741935	256.566667	189.967742	121.870968	44.7857143	116.225806	3464.73333	3532.03226	286.066667	52	27.5454545	697.668675
Karnataka			3.56521739	15.8	76.2903226	379.1	3365.70968	7009.54839	8566.1	7338.29032	2116.7	1117.58065	655.516129	423.75	1356.41935	16068.9	35902.6129	8420.03333	2022.87097	1628.36364	5606.61804
Kerala	0	0.06896552	7.4516129	8.73333333	22.9677419	99.3666667	584.322581	1662.96774	3780.7	7672.45161	5815.93333	5036.06452	5425.6129	4722.75	2122	13735.1	31622.4194	13207.6	14826.7742	19686.9091	6404.80714
Ladakh			0.44	0.3	1.67741935	29.6666667	13.3548387	40.6451613	51.9	64.483871	73.6333333	33.6774194	8.77419355	3.53571429	9.70967742	124.9	149.387097	51.3666667	9.35483871	7.54545455	39.0229446
Lakshadweep												0	2.5483871	10	11.5806452	64.8	169.870968	60.4666667	14.0645161	7.72727273	42.0614754
Madhya Pradesh			3.90909091	87.1	168.741935	182.633333	567.677419	1015	2120.33333	1440.22581	1135.16667	1167.80645	449.645161	232.75	1025.03226	8591.6	7351.54839	364.866667	65.6451613	15.8181818	1555.94499
Maharashtra			9.30434783	323.3	1782.35484	3490.5	7803.70968	11899.7097	19514.6667	9894.48387	4906.7	3501.41935	3071.32258	4391.53571	20214.8065	58870.5667	38460.0645	10660.6	7907.19355	6062.36364	12213.8964
Manipur			0	0.03333333	1.93548387	38.8333333	41.2258065	116.354839	154.466667	242.774194	221.266667	104.096774	29.516129	7.82142857	3.93548387	64.0666667	598.935484	643.866667	919.354839	702.363636	208.34585
Meghalaya				0.64705882	0.48387097	0.66666667	24.3870968	49.6774194	104	126.419355	78.6	53.8064516	11.3870968	7.14285714	3.06451613	85.3666667	599.129032	465.7	488.806452	495.909091	143.851546
Mizoram			0	0	0	4.9	8.38709677	19.4516129	32.5	23.7419355	36.8	12.1935484	5.41935484	1.82142857	1.61290323	51.5333333	195.741935	266.266667	580.290323	750.545455	91.7207921
Nagaland				-0.11111111	6	13.2666667	36.516129	76	70.6	93.7096774	73.8	24.5806452	5.4516129	3.89285714	1.09677419	50.5666667	252.032258	118.266667	83.9354839	99.8181818	63.7389381
Odisha			0.125	4.16666667	54.5483871	168	758.677419	2276	3824.73333	2353.87097	988.7	354.806452	182.322581	76.6428571	113.419355	3163.1	10360.3548	4991.5	2234.22581	1209.72727	1924.11673
Puducherry			0	0.23333333	1.38709677	18.9333333	86.4193548	349.322581	431.3	252.967742	67.5666667	37.4516129	30.6774194	23.9285714	52.3870968	536.2	1496.74194	438.633333	123.548387	86.4545455	237.822266
Punjab			1.73913043	10.5333333	60.516129	106.166667	323.806452	1195.80645	1997.8	667.677419	612.666667	474.225806	220.806452	304.035714	1780.41935	4270.66667	6467.90323	997.866667	119.419355	47.2727273	1150.80998
Rajasthan			2.51724138	78.8	199.322581	301.433333	725.322581	1292.96774	1763.06667	2003.03226	2339.1	1360.25806	317.483871	99.4285714	389.129032	8286.76667	11535.9355	462.066667	42.8387097	18.2727273	1809.96205
Sikkim					0	2.9	16.8387097	32.8064516	43.6666667	31.483871	35.7333333	28.7741935	6.80645161	1.75	3.09677419	50.4666667	239.483871	171.766667	193.129032	155.181818	62.9595506
Tamil Nadu			2.92	69.6	613.612903	2168	4959.80645	5874.41935	5661.93333	4195.74194	1949.8	1179.74194	669.516129	472.535714	1065.51613	8799	29694.0645	13553.6667	2658.74194	1956.27273	4931.413
Telangana			2.6	31.1	47.9677419	429.833333	1462.03226	2072.45161	2214.1	1524.06452	1039.46667	533.483871	261.774194	154.928571	292.967742	4257.23333	4523.25806	1558.86667	701.193548	547.545455	1231.72727
Tripura				0.04166667	8.58064516	37.0666667	107.290323	223.806452	469.666667	160.741935	65.9333333	18.3548387	2.80645161	2.35714286	2.77419355	55.5333333	514.967742	487.9	406.16129	209	163.941057
Uttar Pradesh			3.39285714	67.8064516	169.096774	512.766667	1877.77419	4664.29032	5640.8	2749.22581	2059.7	1390.09677	487.516129	119.571429	405.451613	20065.2	15227.871	531.166667	79.3225806	36.5454545	3242.51613
Uttarakhand			0.35294118	1.6	22.3870968	69.4	136.580645	392.580645	958.666667	449.032258	414.166667	525.032258	175.870968	34.3928571	99.5806452	2491.63333	4950.67742	386.5	67.4193548	39.9090909	664.972816
West Bengal			1.78571429	24.4	141.032258	425.9	1605.96774	2970.74194	3132.76667	3738.80645	3704.73333	2260.64516	608.903226	184.178571	355.064516	7500.73333	17912.4194	4402.16667	933.709677	704.454545	2998.04297
Grand Total	**0**	**0.06896552**	**2.64314516**	**33.682824**	**138.875969**	**364.589914**	**994.501382**	**1827.07373**	**2480.49333**	**1761.61843**	**1232.92571**	**754.957505**	**429.667563**	**347.765873**	**943.193548**	**6123.74167**	**8319.49642**	**2143.80926**	**1121.09767**	**1066.96465**	**1775.06965**

6.5 Age Group Vaccination (Bar Chart)

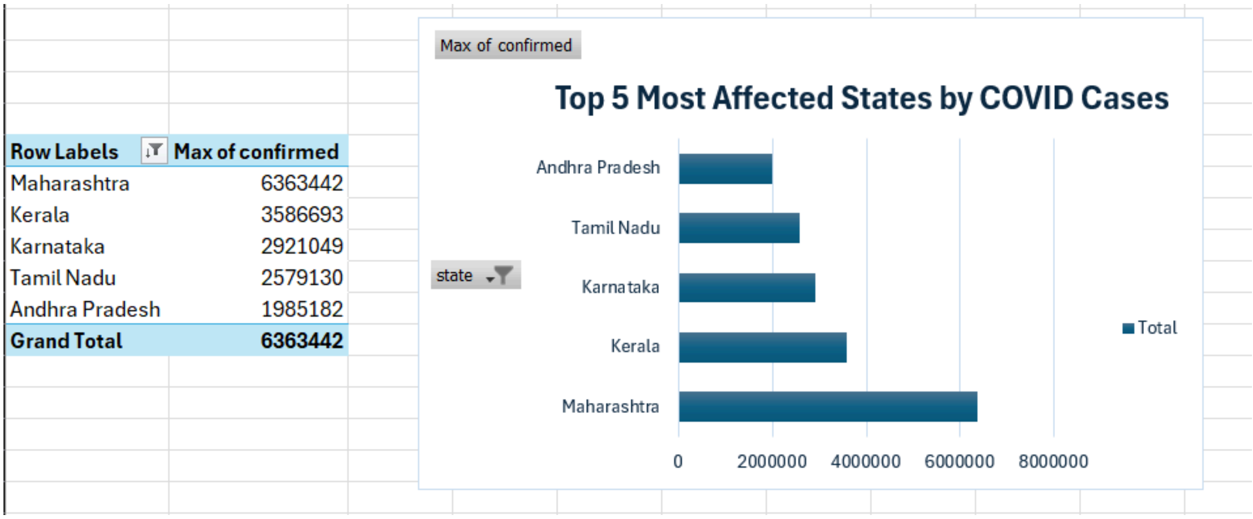
MAX cumulative doses per age group were calculated and compared. The 18–44 age group showed the highest vaccination count.



Row Labels	Max of 18-44 Years (Doses Administered)	Max of 45-60 Years (Doses Administered)	Max of 60+ Years (Doses Administered)
Andaman and Nicobar Islands	118865	128122	63656
Andhra Pradesh	7169390	10746667	5926838
Arunachal Pradesh	514163	263660	101271
Assam	6661719	4109188	1932963
Bihar	13984649	7520647	6329643
Chandigarh	486854	274475	162490
Chhattisgarh	4389004	4705765	2674408
Dadra and Nagar Haveli and Daman and Diu	474123	144345	46753
Delhi	5444642	3397825	1993615
Goa	676052	413810	307049
Gujarat	16854786	11912298	8385291
Haryana	6331770	3596240	3082371
Himachal Pradesh	2216211	1970847	1445383
India	224330364	166757453	118692689
Jammu and Kashmir	2418624	2727353	1662365
Jharkhand	5144231	3078646	2155417
Karnataka	14653045	10803056	7824442
Kerala	6467292	7612487	8017952
Ladakh	133710	70230	55096
Lakshadweep	31333	23734	13228
Madhya Pradesh	19289779	9903544	6263353
Maharashtra	18155358	16275265	12722743
Manipur	781364	403277	221477
Meghalaya	682701	345939	141914
Mizoram	439688	260548	161483
Nagaland	467353	214623	109532
Odisha	7378863	6116468	4659355
Puducherry	365729	233589	154044
Punjab	4419059	3348278	2523928
Rajasthan	15113858	10832586	9437218
Sikkim	352027	189808	107554
Tamil Nadu	12838090	7815365	4869253
Telangana	7067455	5356758	3190679
Tripura	1396211	1189107	629976
Uttar Pradesh	26583720	17007044	10856960
Uttarakhand	3063316	2022265	1581535
West Bengal	11765330	11743594	8881454
(blank)			
Grand Total	**224330364**	**166757453**	**118692689**

Age Group	Total Doses
18–44	448660728
45–60	333514906
60+	237385378

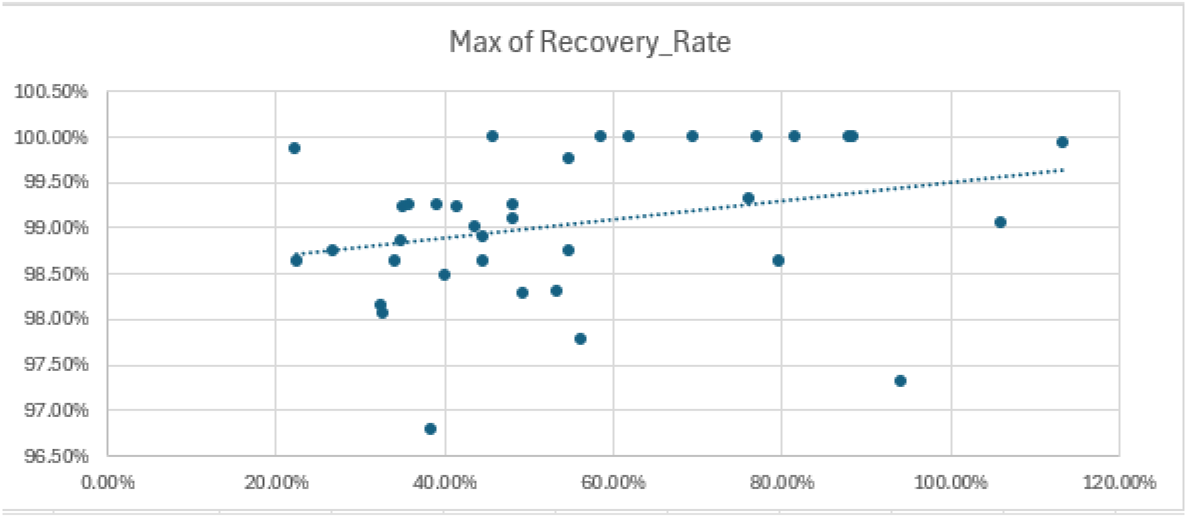
6.6 Top 5 States States ranked using MAX confirmed cases and filtered to Top 5. These states experienced the highest infection burden.



Row Labels	Max of confirmed
Maharashtra	6363442
Kerala	3586693
Karnataka	2921049
Tamil Nadu	2579130
Andhra Pradesh	1985182
Grand Total	**6363442**

6.7 Vaccination vs Recovery

Pivot used MAX Recovery Rate and MAX Vaccination Rate per state.



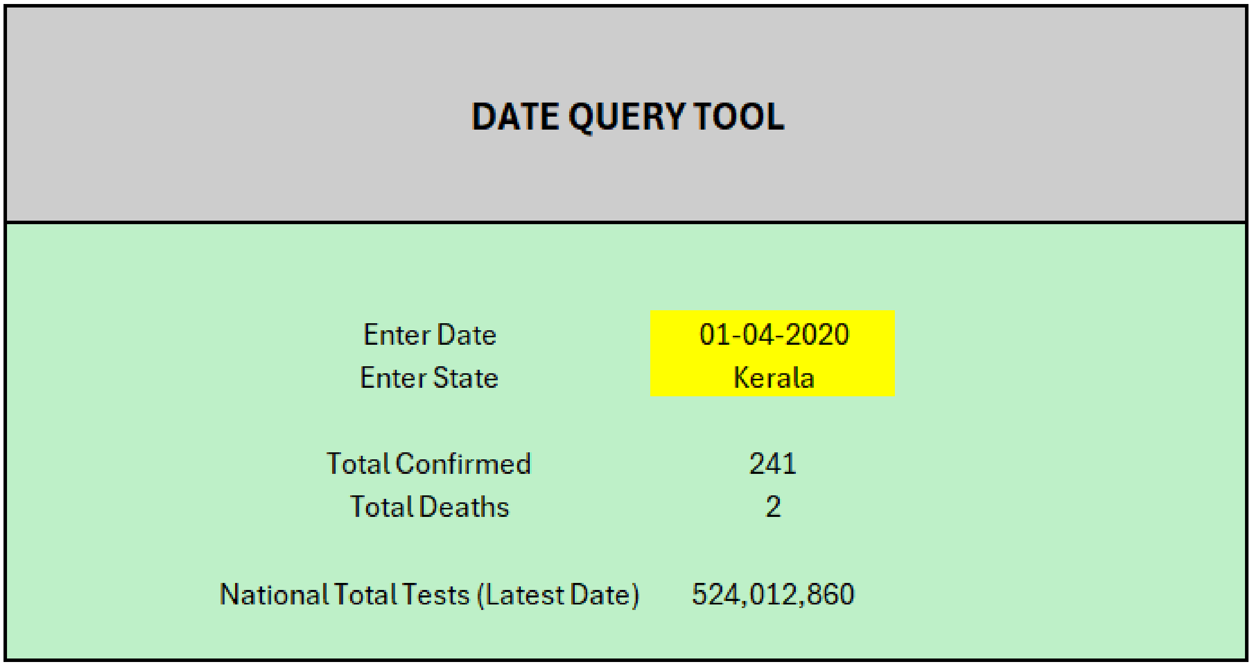
7. Feature Engineering & Logic Tools

Three custom features were engineered to drive the dashboard's logic matrix:

- `active_cases`: Calculated dynamically as `=[@confirmed] - [@cured] - [@deaths]`.
- `excel_risk_level`: A nested conditional logic gate calculating severity: `=IF([@active_cases]>50000, "High Risk", IF([@active_cases]>10000, "Medium Risk", "Low Risk"))`.
- `day_of_week`: Extracted from the timestamp using `=TEXT([@date], "dddd")` to enable seasonal tracking.

7.1 Advanced Dashboard Modules

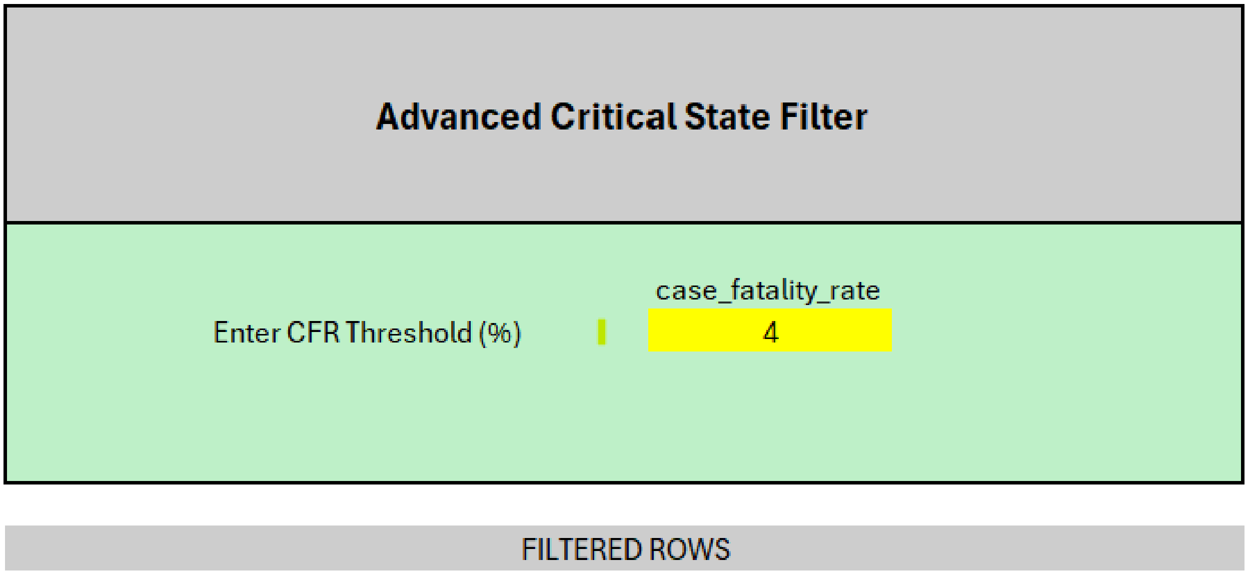
- **The Date Query Engine:** An interactive, two-criteria search tool was engineered using nested boolean arrays. By combining `INDEX` and `MATCH` functions (e.g., `=MATCH(1, (CovidData[date]=$C$9)*(CovidData[state]=$C$10), 0)`), the tool allows users to input any Date and State to instantly retrieve historical confirmed cases and deaths.



DATE QUERY TOOL

Enter Date	01-04-2020
Enter State	Kerala
Total Confirmed	241
Total Deaths	2
National Total Tests (Latest Date)	524,012,860

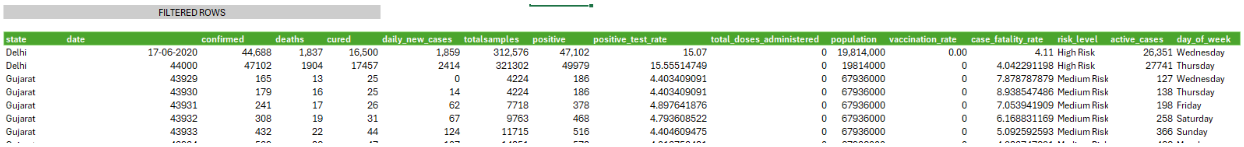
- **National Testing Matrix:** To calculate the cumulative national tests without double-counting historical running totals, a `SUMIFS` logic gate was deployed: `=SUMIFS(CovidData[totalsamples],CovidData[date],MAX(CovidData[date]))`.
- **Dynamic Critical State Filter:** While the project initially scoped a static "Advanced Filter" to extract states breaching a specific Case Fatality Rate (CFR), we engineered a superior solution. Utilizing the dynamic array `=FILTER(P4_Data, P4_Data[case_fatality_rate] > $G$2, "No states found")`, the dashboard now features a parameterized input box, allowing stakeholders to dynamically adjust the CFR threshold on the fly.



Advanced Critical State Filter

	case_fatality_rate
Enter CFR Threshold (%)	4

FILTERED ROWS



FILTERED ROWS

state	date	confirmed	deaths	cured	daily_new_cases	totalsamples	positive	positive_test_rate	total_doses_administered	population	vaccination_rate	case_fatality_rate	risk_level	active_cases	day_of_week
Delhi	17-06-2020	44,688	1,837	16,500	1,859	312,576	47,102	15.07	0	19,814,000	0.00	4.11	High Risk	26,351	Wednesday
Delhi	44000	47102	1904	17457	2414	321302	49979	15.55514749	0	19814000	0	4.042291198	High Risk	27741	Thursday
Gujarat	43929	165	13	25	0	4224	186	4.403409091	0	67936000	0	7.878787879	Medium Risk	127	Wednesday
Gujarat	43930	179	16	25	14	4224	186	4.403409091	0	67936000	0	8.938547486	Medium Risk	138	Thursday
Gujarat	43931	241	17	26	62	7718	378	4.897641876	0	67936000	0	7.053941909	Medium Risk	198	Friday
Gujarat	43932	308	19	31	67	9763	468	4.793608522	0	67936000	0	6.168831169	Medium Risk	258	Saturday
Gujarat	43933	432	22	44	124	11715	516	4.404609475	0	67936000	0	5.092592593	Medium Risk	366	Sunday

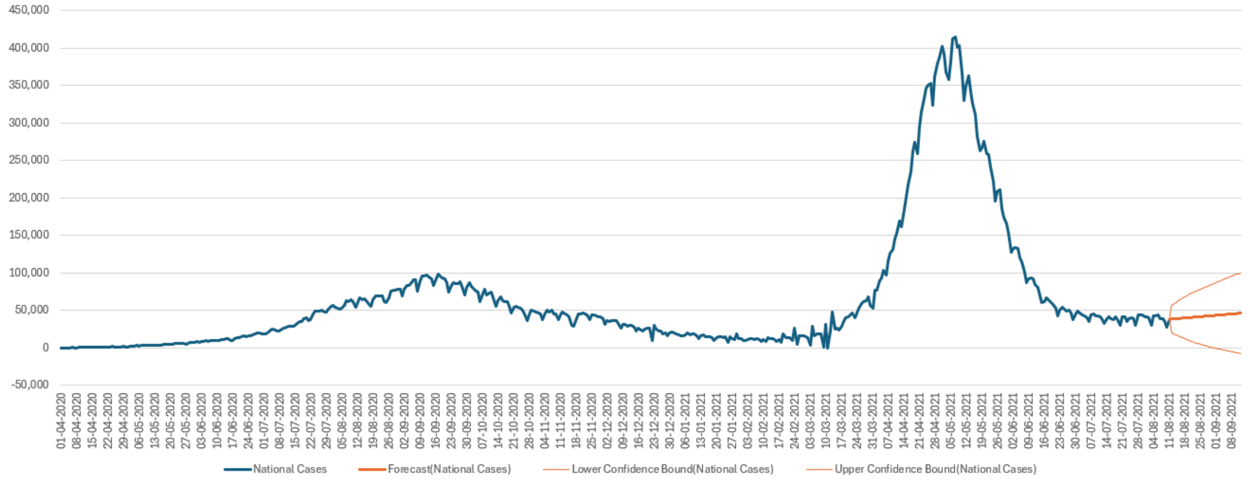
7.2 Analytical Insights & Strategic Recommendations

- **Insight 1: Administrative Seasonality (Weekday Analysis)**

- ○ ***Finding****:* Pivot table analysis evaluating the average daily cases per weekday revealed a distinct anomaly. While most days averaged between 1,821 and 1,928 cases, Tuesdays consistently showed a pronounced dip to 1,621 cases.

- ○ ***Actionable Recommendation****:* This represents an "administrative lag" caused by reduced testing and reporting lab capacity over the weekend, resulting in an artificial drop in reported numbers on Tuesdays. Healthcare supply chains and hospital staffing must not scale down operations based on Tuesday data, as the viral spread remains constant despite the reporting artifact.

- **Insight 2: Predictive Trajectory (Time Series)**
  - ○ ***Finding****:* Utilizing a single-state isolation method, an Exponential Smoothing (ETS) algorithmic forecast was successfully deployed to project daily new cases 30 days into the future.



  - ○ ***Actionable Recommendation****:* By mapping the 95% confidence intervals generated by this forecast, policymakers can anticipate upper-bound scenarios for hospital bed utilization a full month in advance.

- **Insight 3: Dynamic Risk Stratification**
  - ○ ***Finding****:* The dynamic CFR filtering array successfully identified states with historically critical death rates.
  - ○ ***Actionable Recommendation****:* Federal resource allocation (such as ventilators and supplemental oxygen) should be routed using this live, ratio-based severity matrix rather than relying solely on cumulative gross case counts, ensuring aid reaches the most mathematically vulnerable populations first.

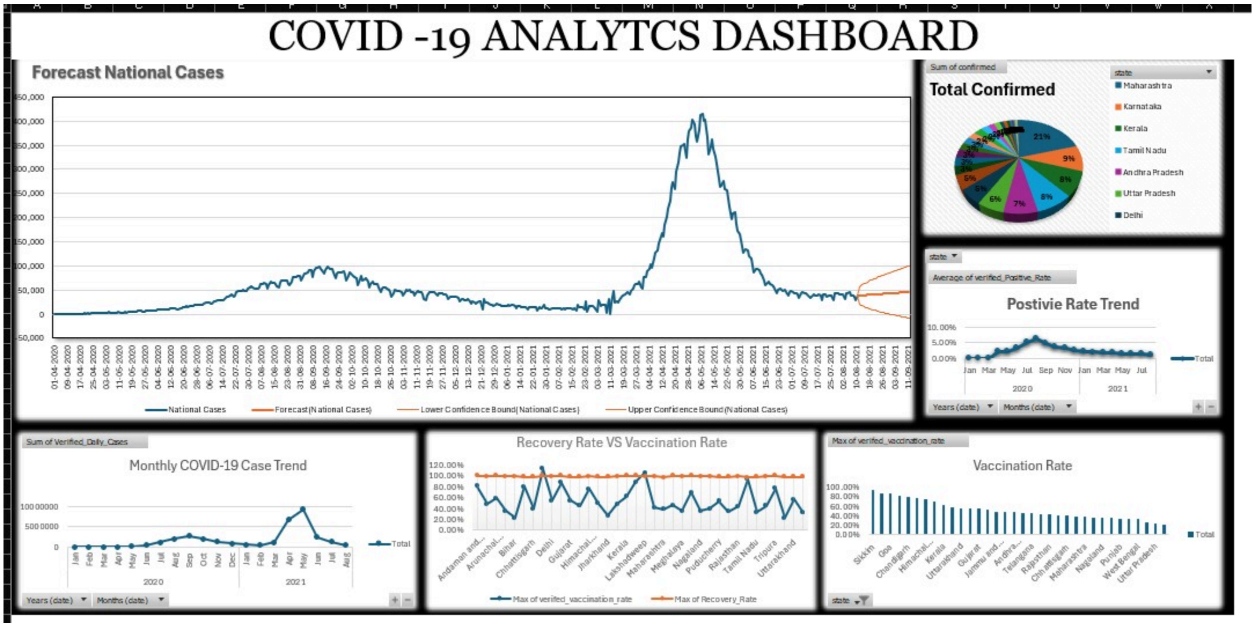
Section 4: Interactive Dashboard & State-Specific Analysis (P5)

4.1 Dashboard Architecture & Integration

The final phase of the project involved synthesizing the backend data models, predictive analytics, and static charts from previous phases into a unified, interactive Graphical User Interface (GUI). All disparate datasets (`SummaryData`, `VaccineData`) were merged into a single Master Workbook. During integration, external reference errors were resolved by forcefully redirecting the Pivot Caches to the local master tables.

4.2 COVID-19 Analytics Dashboard

To provide policymakers with an immediate, high-level overview of the pandemic's national trajectory, a centralized dashboard was engineered.


COVID -19 ANALYTCS DASHBOARD
Forecast National Cases
National Cases
Forecast(National Cases)
Lower Confidence Bound(National Cases)
Upper Confidence Bound(National Cases)
Sum of confirmed
Total Confirmed
state
Maharashtra
Karnataka
Kerala
Tamil Nadu
Andhra Pradesh
Uttar Pradesh
Delhi
21%
9%
8%
8%
7%
6%
5%
5%
Average of verified_Positive_Rate
Postivie Rate Trend
Total
2020
2021
Years (date)
Months (date)
Sum of Verified_Daily_Cases
Monthly COVID-19 Case Trend
Recovery Rate VS Vaccination Rate
Max of verified_vaccination_rate
Max of Recovery_Rate
Max of verifed_vaccination_rate
Vaccination Rate

Key Visual Modules Integrated:

- **National Predictive Trajectory:** The "Forecast National Cases" chart utilizes the Exponential Smoothing (ETS) algorithm to plot the aggregated national daily cases, projecting a future trendline with upper and lower 95% confidence bounds.
- **Geospatial Burden:** A "Total Confirmed" donut chart visually isolates the states carrying the highest percentage of the national viral burden (e.g., Maharashtra at 21%).

- **Temporal Wave Analysis:** Line charts mapping the "Positive Rate Trend" and "Monthly Case Trend" clearly visualize the distinct waves of the pandemic.
- **Efficacy & Adoption:** The dashboard maps the "Vaccination Rate" side-by-side with a "Recovery Rate vs. Vaccination Rate" dual-axis trend, visually correlating vaccine rollouts with improved survival outcomes.

4.3 State-Specific Analysis Module

While the main dashboard provides a national macro-view, a dedicated **"State-Specific Analysis"** tool was built to allow deep-dive drill-downs into individual regional metrics.

Interactive Engineering Elements:

- **Dynamic Parameterization (The Dropdown):** A Data Validation list was engineered in cell `A2`. This allows the user to actively select any of the 36 States or Union Territories (e.g., "Himachal Pradesh").
- **KPI Retrieval Engine:** Upon selecting a state, a backend array of formulas instantly fetches the localized data:
    - **Total Cases & Deaths:** `SUMIFS` formulas were deployed to dynamically aggregate the historical data for the explicitly selected state.
        - *Cases Formula:* `=SUMIFS(SummaryData!C:C, SummaryData!A:A, 'State-Specific Analysis '!A2)`
        - *Deaths Formula:* `=SUMIFS(SummaryData!D:D, SummaryData!A:A, 'State-Specific Analysis '!A2)`
    - **Temporal Sync (Date):** To ensure the dashboard reports the most current snapshot, a `MAXIFS` array isolates the absolute latest recorded timestamp for that specific state.
        - *Date Formula:* `=MAXIFS(SummaryData!B:B, SummaryData!A:A, 'State-Specific Analysis '!A2)`

Advanced Multi-Criteria Logic (Risk Level): Standard Excel lookup functions cannot evaluate two criteria simultaneously (e.g., finding the Risk Level for a specific *State* on a specific *Date*). To bypass this limitation, an advanced in-memory array was engineered:

- *Risk Formula:* `=VLOOKUP($A$2&$D$2, CHOOSE({1,2}, SummaryData!A:A&SummaryData!B:B, SummaryData!N:N), 2, FALSE)`
- *Methodology:* This formula concatenates the State and Date (`$A$2&$D$2`) to create a unique composite key. The `CHOOSE({1,2}...` function dynamically

generates a "virtual table" in Excel's memory, allowing the `VLOOKUP` to accurately fetch the Risk Level from column `N` without altering the raw dataset structure.

Visual Drill-Down: A localized Bar Chart is dynamically linked to these KPI outputs. As the user cycles through different states in the dropdown, the chart instantly updates to visually compare the Total Cases against the Total Deaths for that specific geographic region.