

Analysis of COVID19 data using Snowflake and Power BI

DOMAIN - HEALTHCARE

Members Of The Team

NAME	PS No.
Garima Singh	10683205
Numan Salim Shaikh	10683782
Mukesh Parmar	10683809
Atharva Jadhav	10683762

Problem Statement

- Recent Covid-19 pandemic has raised alarms over one of the most overlooked areas to focus: Healthcare Management. While healthcare management has various use cases for using data science.
- Increasing the impact of this virus as the number of cases increased exponentially .
- As a Snowflake developer/ consultant, we are helping the data analytical team to explore the dataset using snowflake , history data is loaded and current data is loaded with snowpipes into database.
- Our approach is to merge all the datasets based on the common key and create a single table and cleansing of noise or missing data.

Technical specifications

- **Data sources:** Local Datasets, S3
- **Data Storage :** AWS S3 , Snowflake
- **File formats :** CSV , JSON, Parquet
- **Data Ingestion :** Snowpipes
- **Data cleansing and scrubbing :** SQL /Python, Snowflake (Stage)
- **Data computation and Analysis :** SnowSQL CLI, Snowsql WebClient
- **ELT :** Snowflake
- **Visualization :** PowerBI



Snowflake Mechanisms

- Zero Copy Cloning
- Time Travel
- Optimised Compression Storage
- Dynamic Caching

Data Workflow

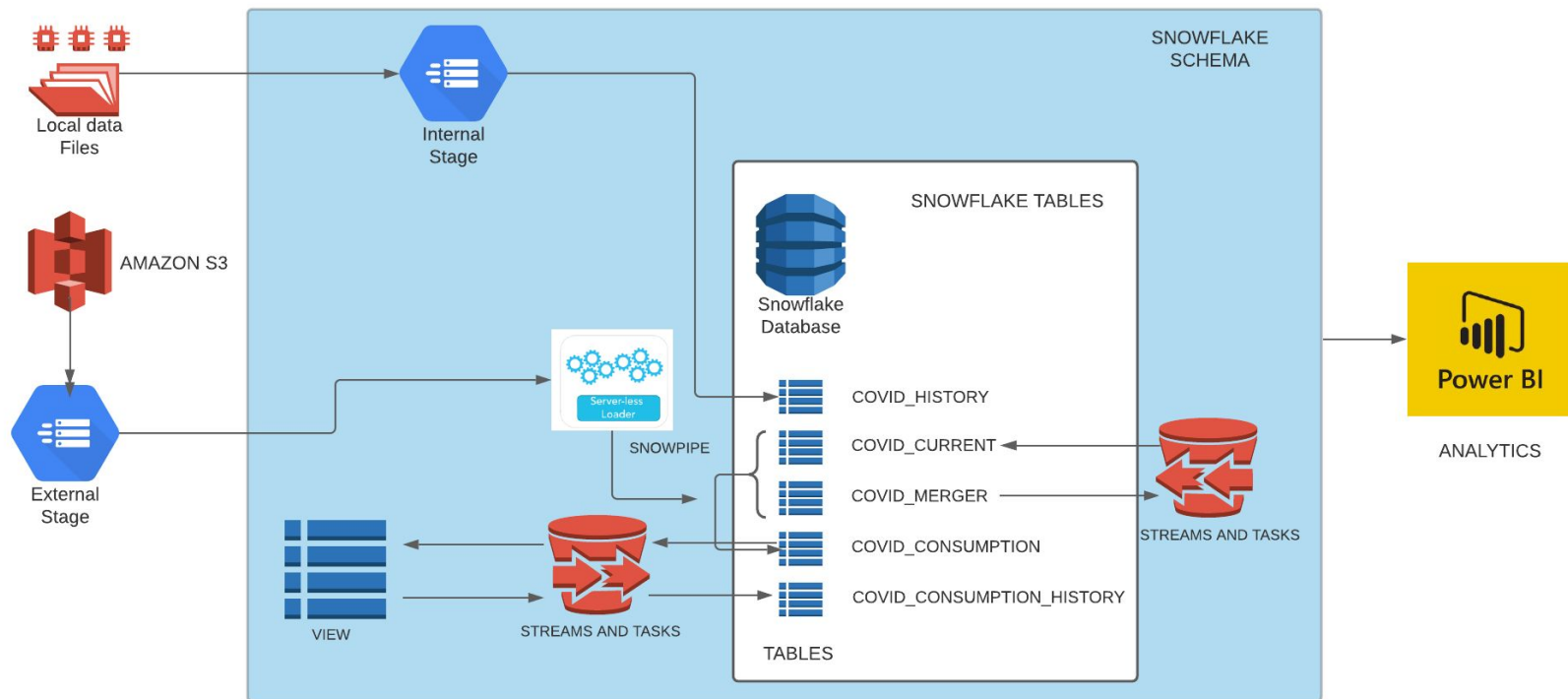
There exists multiple approaches to load Data into Snowflake, some of the techniques that we used while loading the given datasets were as follows:

(A) Data Flow From Internal Storage to SnowDB

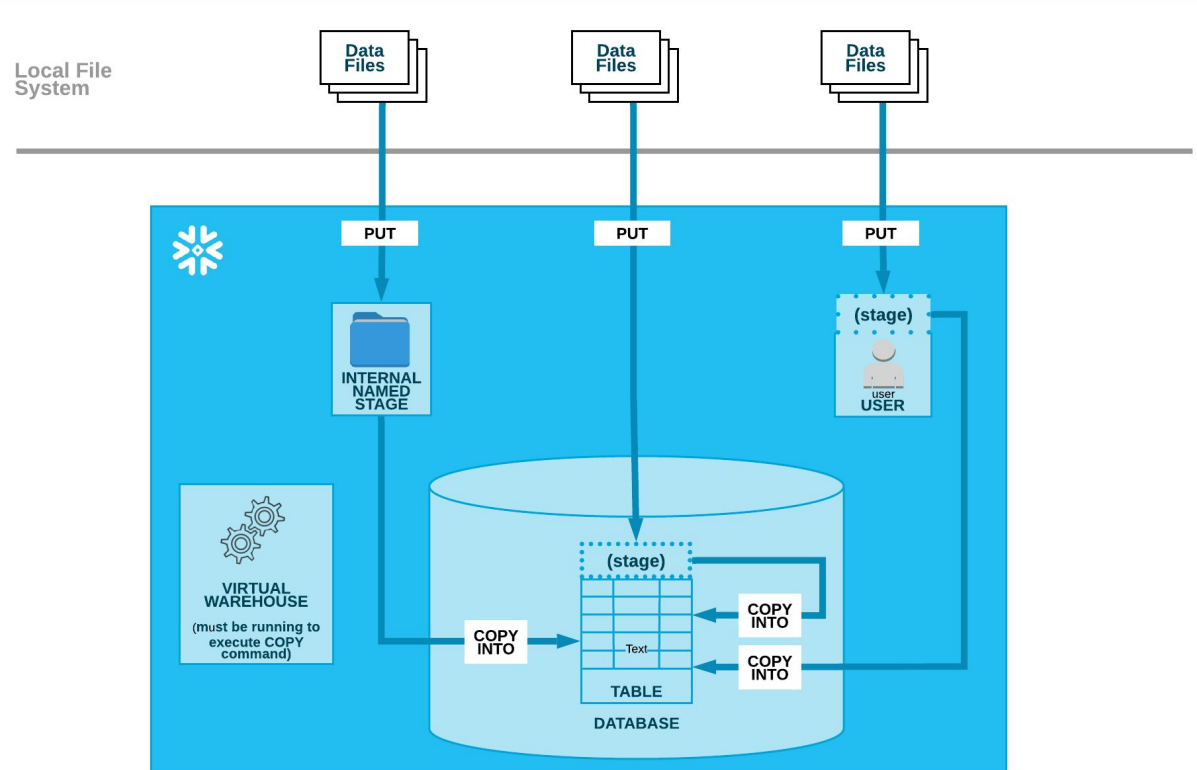
(B) Bulk Data Flow From External Storage(S3) to SnowDB

(C) Continuous Data Flow From External Storage(S3) to SnowDB

ARCHITECTURE



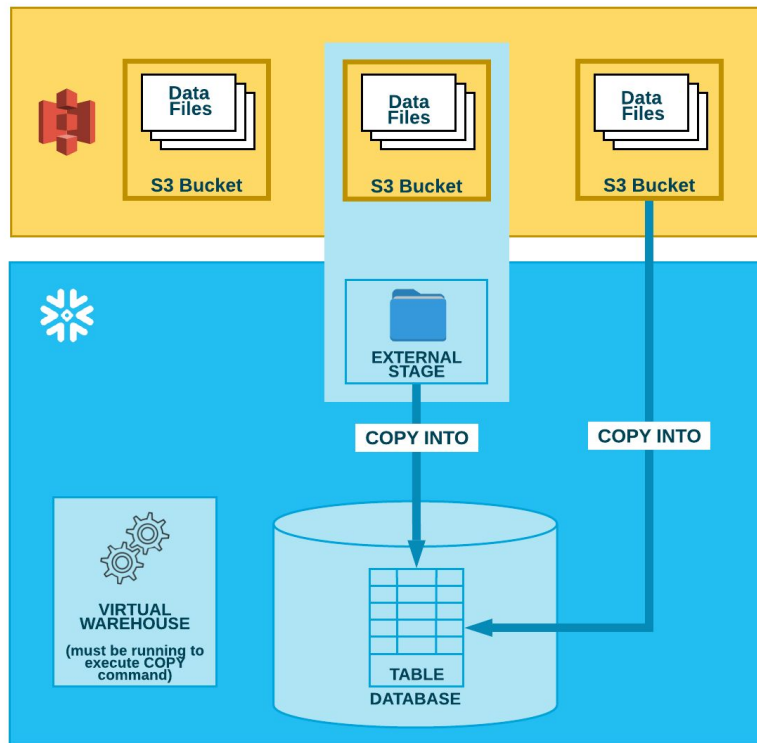
Data Flow From Internal Storage to SnowDB



Internal Stage Creation

```
NUMAN#COMPUTE_WH@COVID_DB.SCHEMA>CREATE STAGE "COVID_DB"."SCHEMA".covid_int;  
+-----+  
| status |  
+-----+  
| Stage area COVID_INT successfully created. |  
+-----+  
1 Row(s) produced. Time Elapsed: 1.226s  
NUMAN#COMPUTE_WH@COVID_DB.SCHEMA>
```

Bulk Data Flow From External Storage(S3) to SnowDB



Bucket Folder Creation on AWS S3

Amazon S3 > snow-extstage-grp1

snow-extstage-grp1 [Info](#)

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

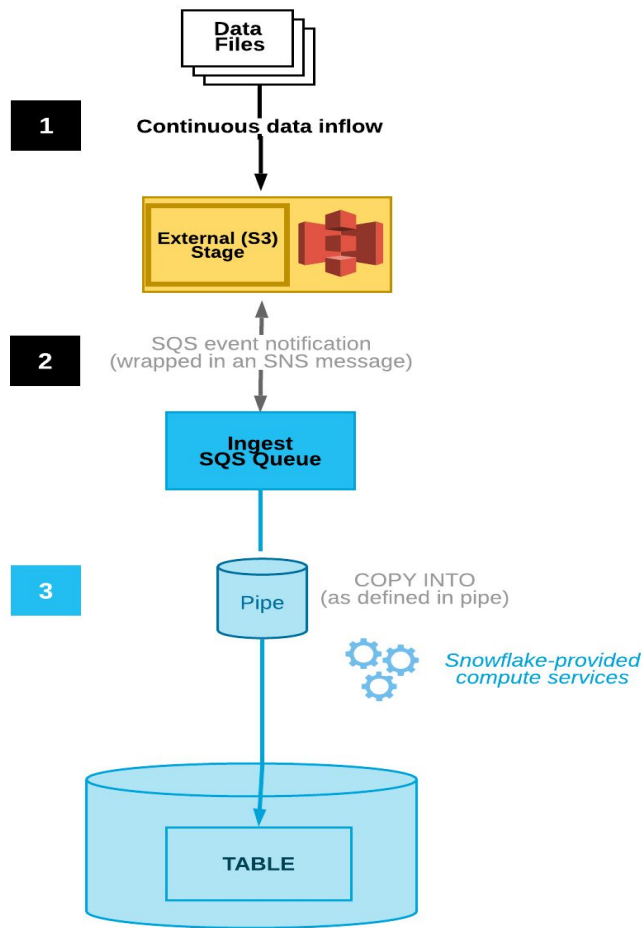
Copy S3 URI Copy URL Download Open Delete

Create folder Upload

☐ Show versions < 1 >

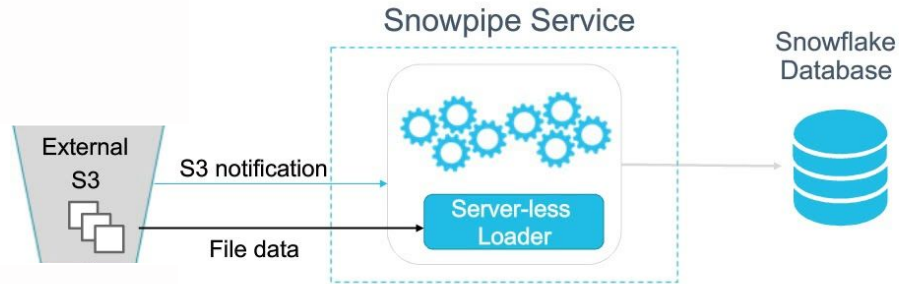
<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	covid/	Folder	-	-	-

Continuous Data Flow From External Storage(S3) to SnowDB



Snowpipes

Snowpipe Scenario: Automatic Loading from S3



Cluster Keys

Our **covid_history** and **covid_current** tables reflect the cluster key (**country_region,province_state**)

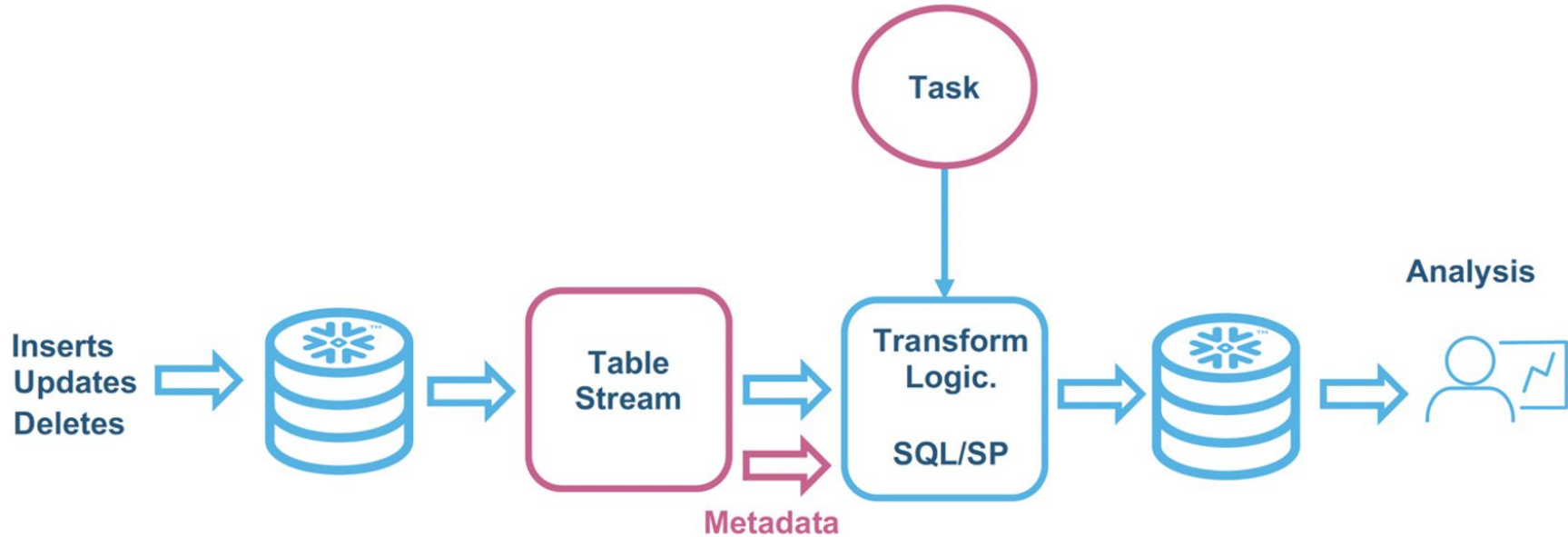
```
| cluster_by |  
tion_progress | search_optimization_bytes  
+-----+  
+-----+  
| LINEAR(COUNTRY_REGION,province_state) |  
NULL | NULL  
| LINEAR(COUNTRY_REGION,province_state) |  
NULL | NULL  
+-----+
```

Findings & Challenges - Phase 1

- The data has 4 schemas having 6,8,12,14 columns.
- The data type for few columns were mismatched and also the columns needed to be renamed to standardize them for merge operation.
- The column 'Combined_Key' contained comma separated value and were treated as values for 3 different columns causing mismatch data type.
- The data type assigned to latitude and longitude columns was earlier **number** with a precision of 4,4 which was generating incorrect values so we changed the type to **float**.



Streams



Tasks

Tasks can be combined with table streams for continuous ELT workflows to process recently changed table rows. Tasks can also be used independently to generate periodic reports by inserting or merging rows into a report table or perform other periodic work.

```
A>create or replace task covid_task
  warehouse = compute_wh
  schedule  = '1 minute'
  when
    system$stream_has_data('ncov_stream')
  as
    insert into covid_current select fips,admin2,province_state,COUNTRY_REGION,last_update,lat,long,confirmed,deaths,recovered,active,c
    ombined_key,incident_rate,case_fatality_ratio
    from ncov_stream;
-----
```

Implementation of Slowly Changing Dimensions

- A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse. It is considered one of the most critical ETL (extract, transform, load) tasks in tracking the history of dimension records.
- Type 1 SCDs - Overwriting In a Type 1 SCD the new data overwrites the existing data. Thus the existing data is lost as it is not stored anywhere else. This is the default type of dimension you create. You do not need to specify any additional information to create a Type 1 SCD.
- Type 2 SCDs - Creating another dimension record. A Type 2 SCD retains the full history of values. When the value of a chosen attribute changes, the current record is closed and a new record is created with the changed data values.
- Type 3 SCDs - Creating a current value field. A Type 3 SCD stores two versions of values for certain selected level attributes. Each record stores the previous value and the current value of the selected attribute. When the value of any of the selected attributes changes, the current value is stored as the old value, and the new value becomes the current value.

SCD Type - 2

```
8 Row(s) produced. Time Elapsed: 2.223s
NUMAN#COMPUTE_WH@COVID_DB.SCHEMA>select fips,admin2,last_update,confirmed from covid_consumption where FIPS = 1001 and confirmed=663
union
select fips,admin2,last_update,confirmed from covid_consumption where FIPS = 1001 and confirmed=708
union
select fips,admin2,last_update,confirmed from covid_consumption where FIPS = 1001 and confirmed=872
union
select fips,admin2,last_update,confirmed from covid_consumption where FIPS = 1001 and confirmed=1029;
```

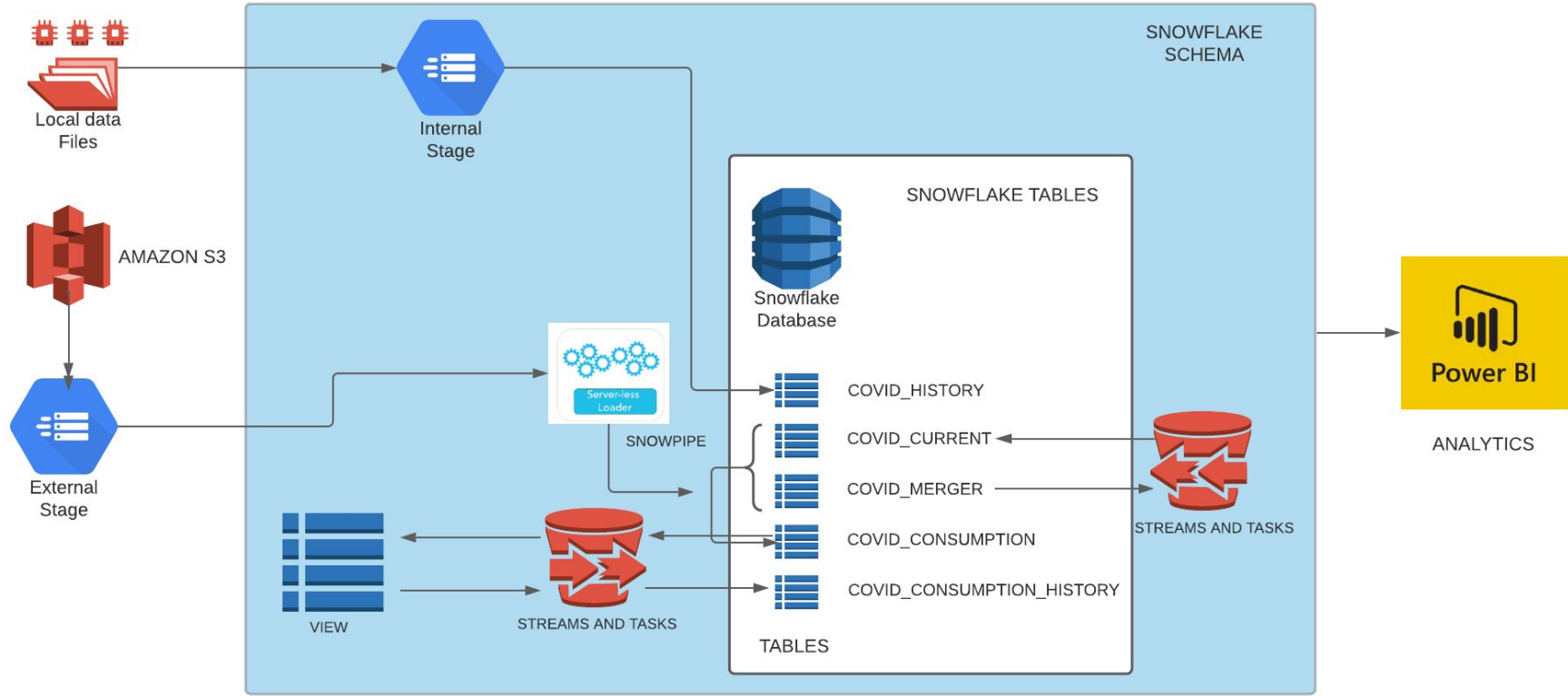
FIPS	ADMIN2	LAST_UPDATE	CONFIRMED
1001	Mukesh	2020-07-31 04:35:18.000	1029
1001	Numan	2020-07-09 04:34:23.000	663
1001	Atharva	2020-07-12 04:34:30.000	708
1001	Garima	2020-07-21 04:38:46.000	872

```
4 Row(s) produced. Time Elapsed: 1.170s
NUMAN#COMPUTE_WH@COVID_DB.SCHEMA>select fips,admin2,last_update,confirmed,current_flag from covid_consumption_history where FIPS = 1001 and confirmed=663
union
select fips,admin2,last_update,confirmed,current_flag from covid_consumption_history where FIPS = 1001 and confirmed=708
union
select fips,admin2,last_update,confirmed,current_flag from covid_consumption_history where FIPS = 1001 and confirmed=872
union
select fips,admin2,last_update,confirmed,current_flag from covid_consumption_history where FIPS = 1001 and confirmed=1029;
```

FIPS	ADMIN2	LAST_UPDATE	CONFIRMED	CURRENT_FLAG
1001	Atharva	2020-07-12 04:34:30.000	708	1
1001	Numan	2020-07-09 04:34:23.000	663	1
1001	Garima	2020-07-21 04:38:46.000	872	1
1001	Mukesh	2020-07-31 04:35:18.000	1029	1
1001	Autauga	2020-07-31 04:35:18.000	1029	0
1001	Autauga	2020-07-12 04:34:30.000	708	0
1001	Autauga	2020-07-21 04:38:46.000	872	0
1001	Autauga	2020-07-09 04:34:23.000	663	0

```
8 Row(s) produced. Time Elapsed: 0.306s
```

WORKFLOW



Time Travel

Continuous Data Protection Lifecycle

Standard operations allowed:
Queries, DDL, DML, etc.

Time Travel allowed:
SELECT ... AT | BEFORE ...
CLONE ... AT | BEFORE ...
UNDROP ...

No user operations allowed
(data recoverable only by
Snowflake)

Current Data
Storage

Time Travel
Retention
(1-90 Days)

Fail-Safe
(transient: 0 days,
Permanent: 7 days)

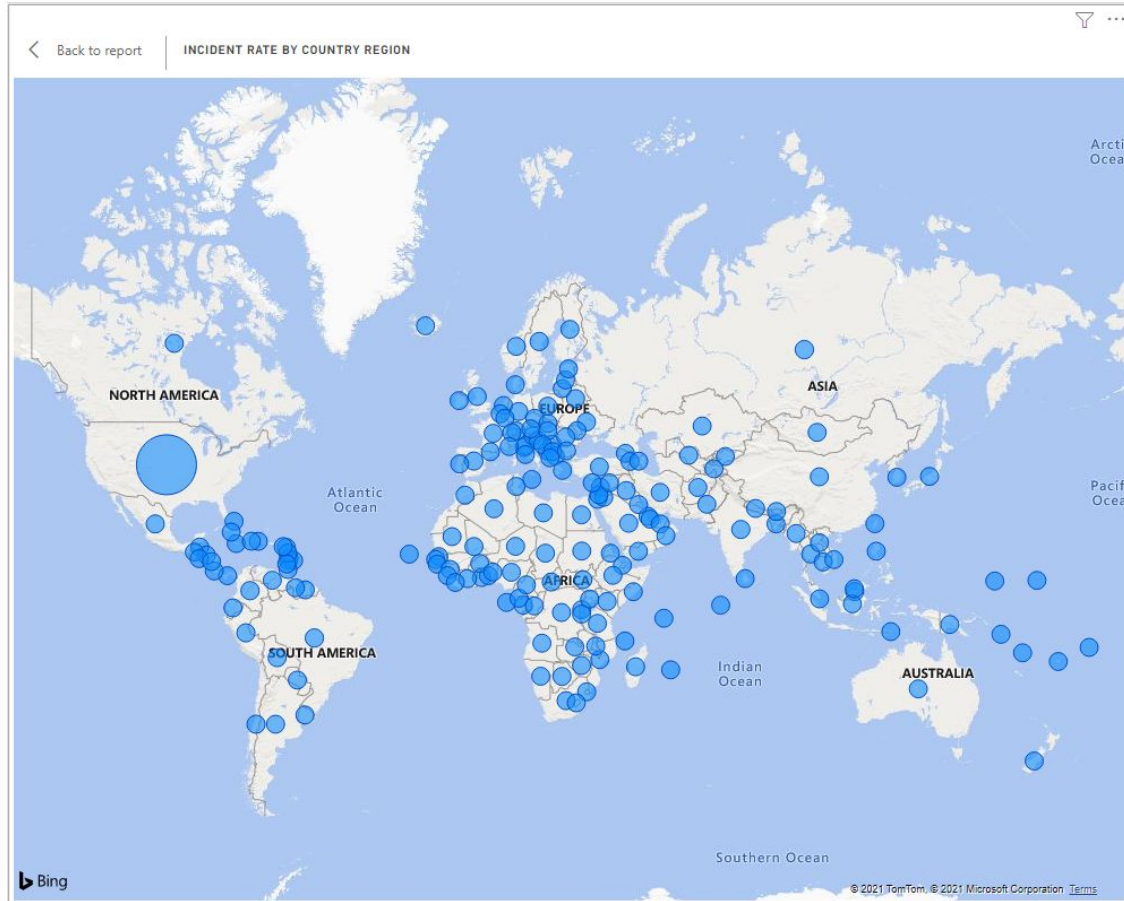
Findings & Challenges- Phase 2

- While implementing the slowly changing dimensions using view, we came to know there were no columns that could uniquely identify a row in the table. If each row is not uniquely identified then the update and delete changes would not be possible.
- To overcome this we used a pair of keys (composite keys) in order to identify each row and to help implement the type -2 SCD.

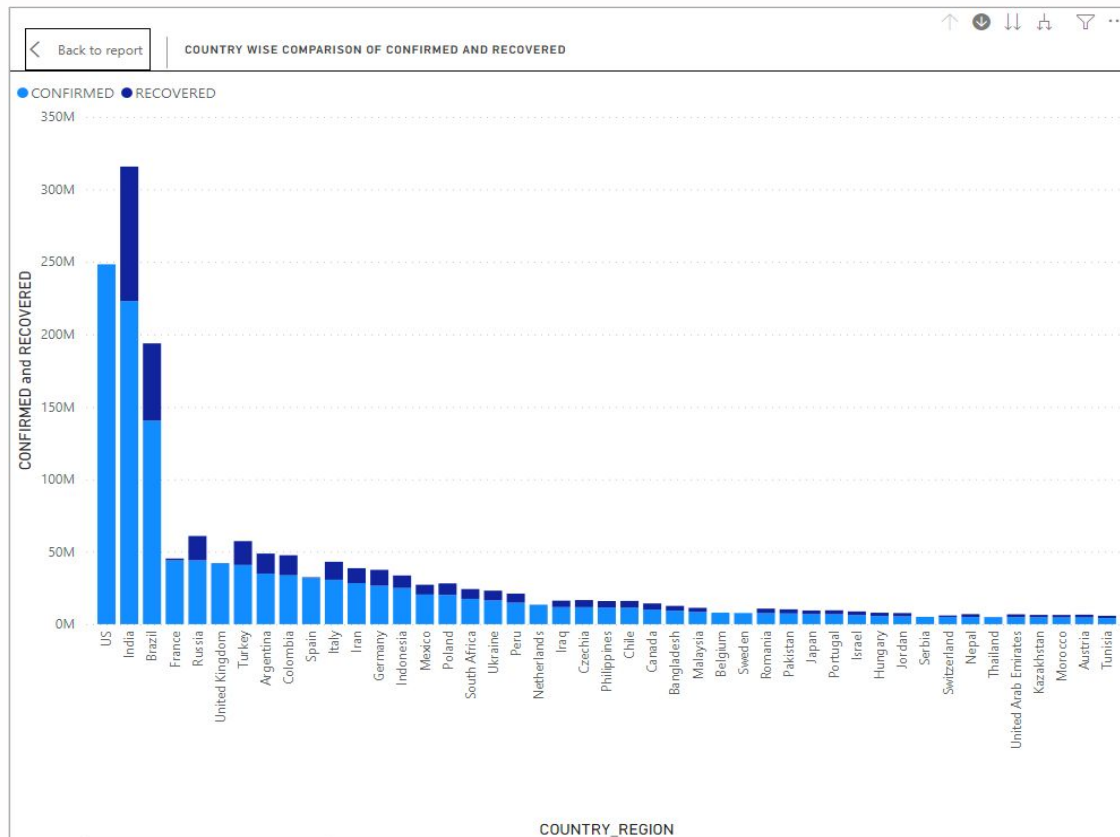


REPORTS USING BUSINESS QUERIES

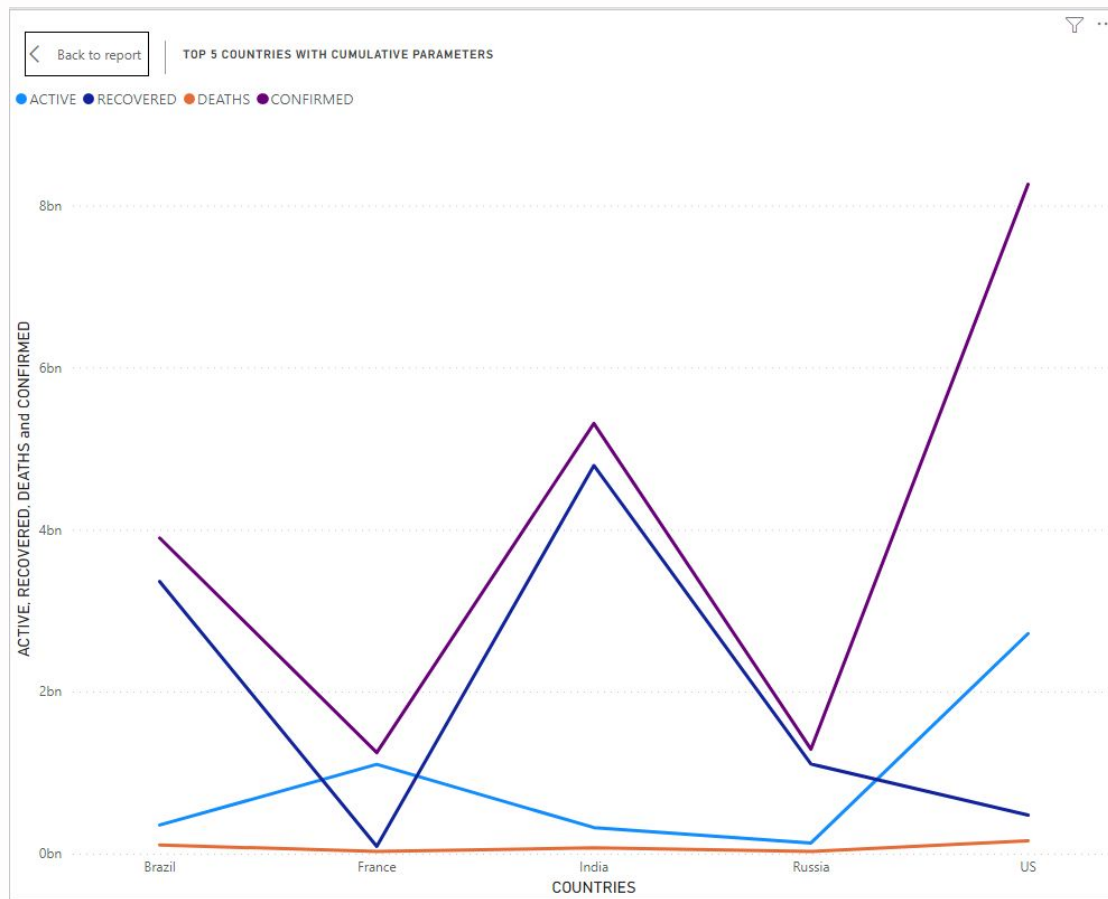
INCIDENT RATE BY COUNTRY



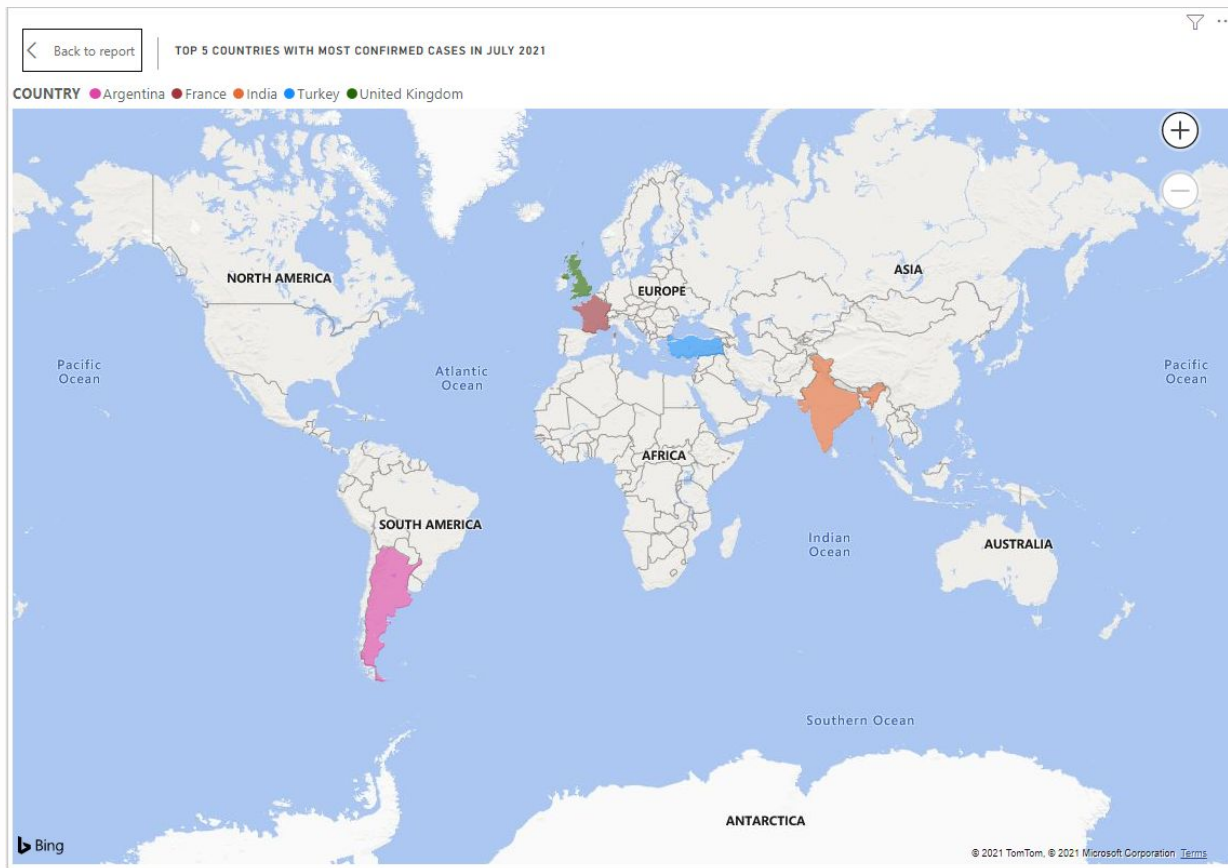
COUNTRY WISE COMPARISON OF CONFIRMED AND RECOVERED CASES



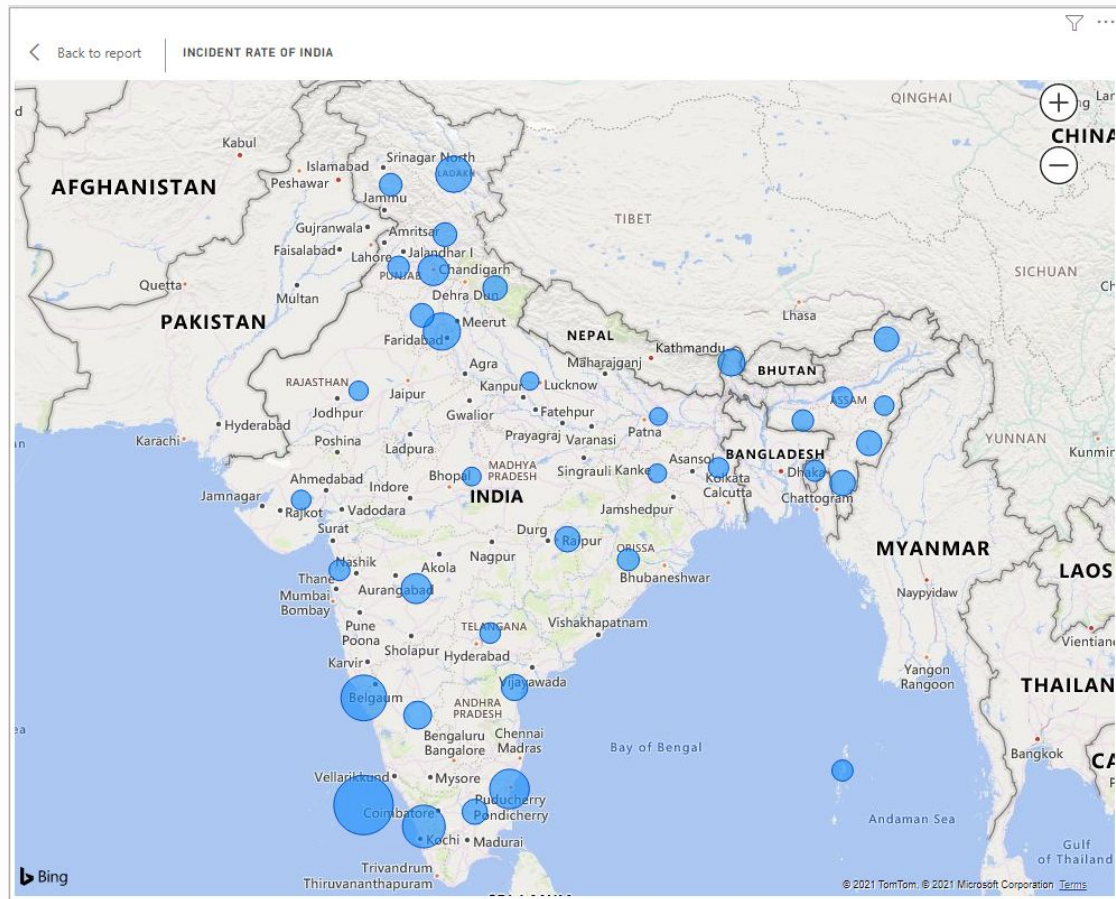
TOP 5 COUNTRIES WITH CUMULATIVE PARAMETERS



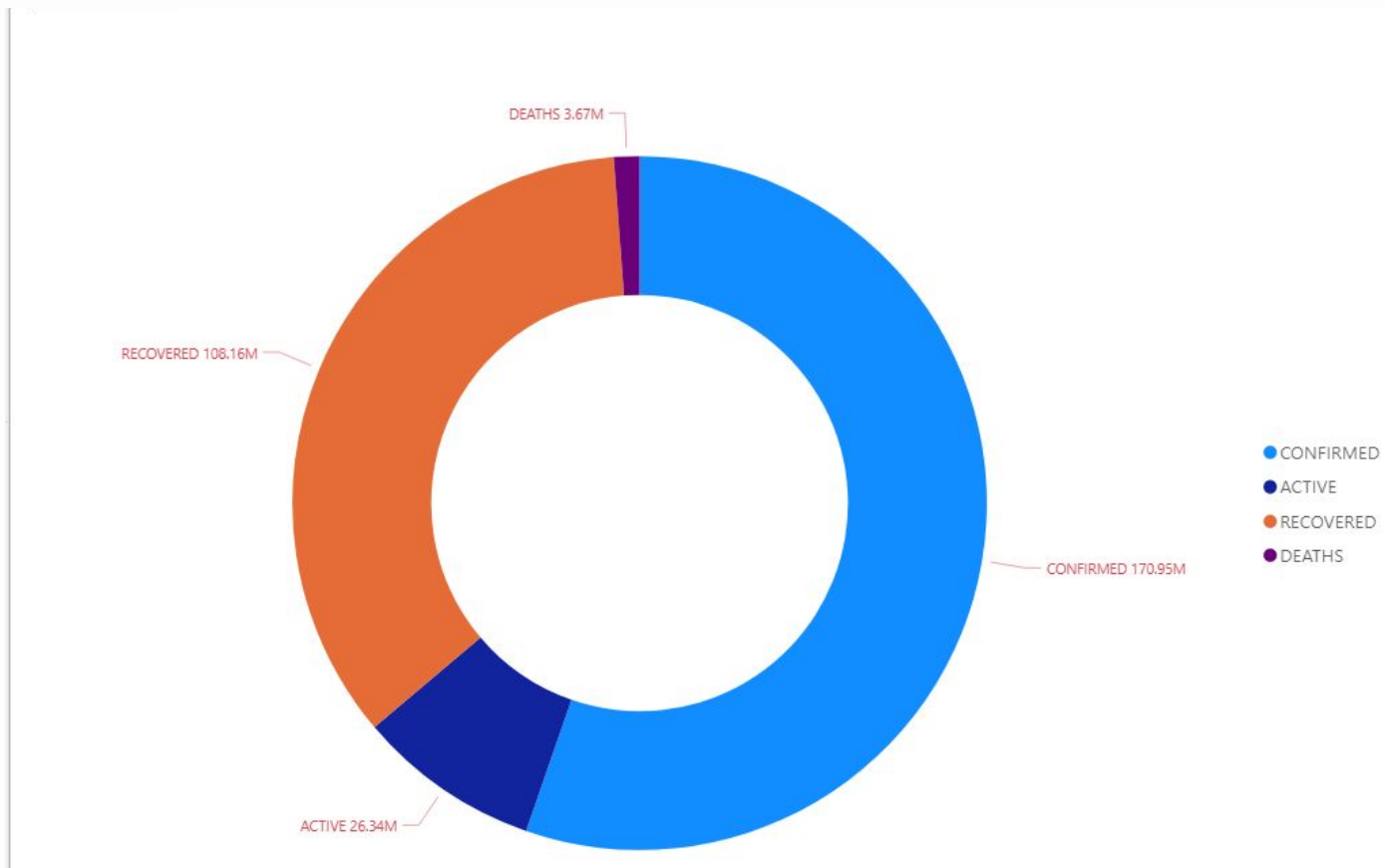
TOP 5 COUNTRIES WITH MOST CONFIRMED IN JULY 2021



INCIDENT RATE OF INDIA BASED ON STATES



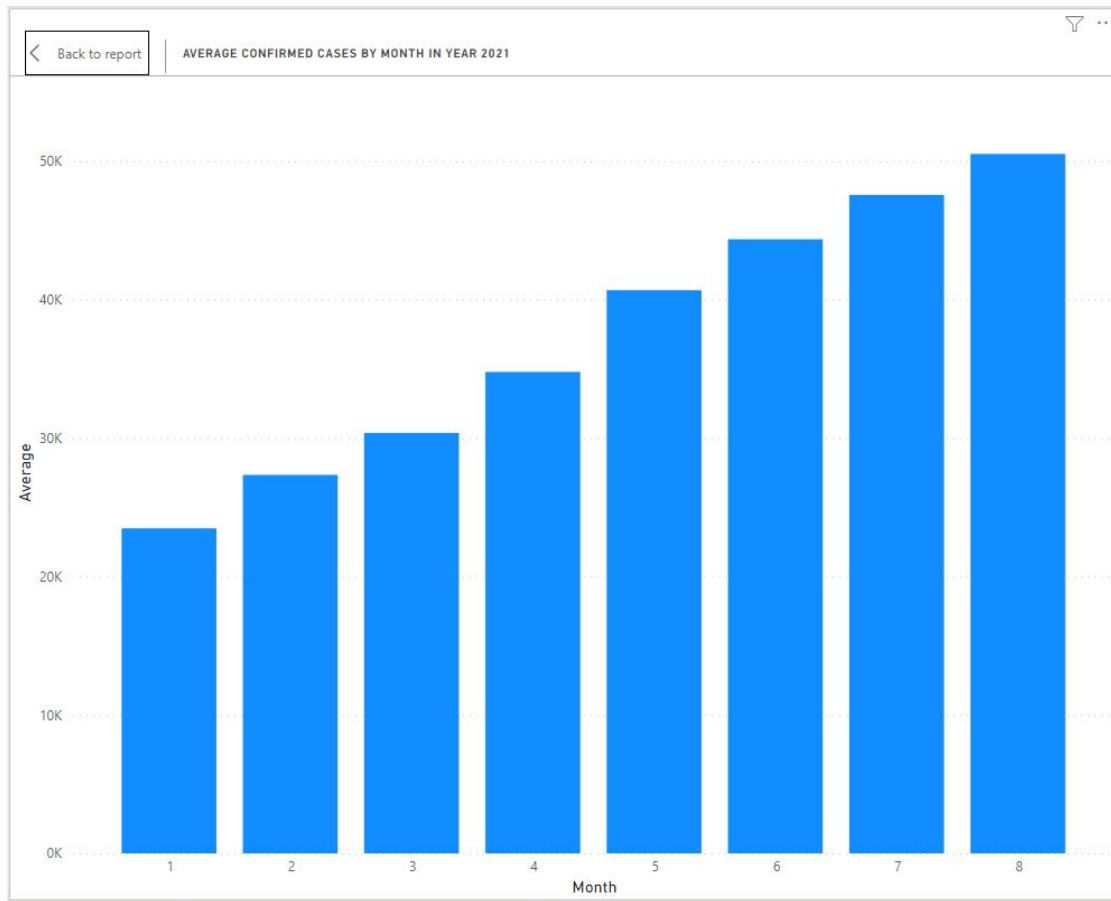
TOTAL RECORD OF INDIA WITH CUMULATIVE PARAMETERS



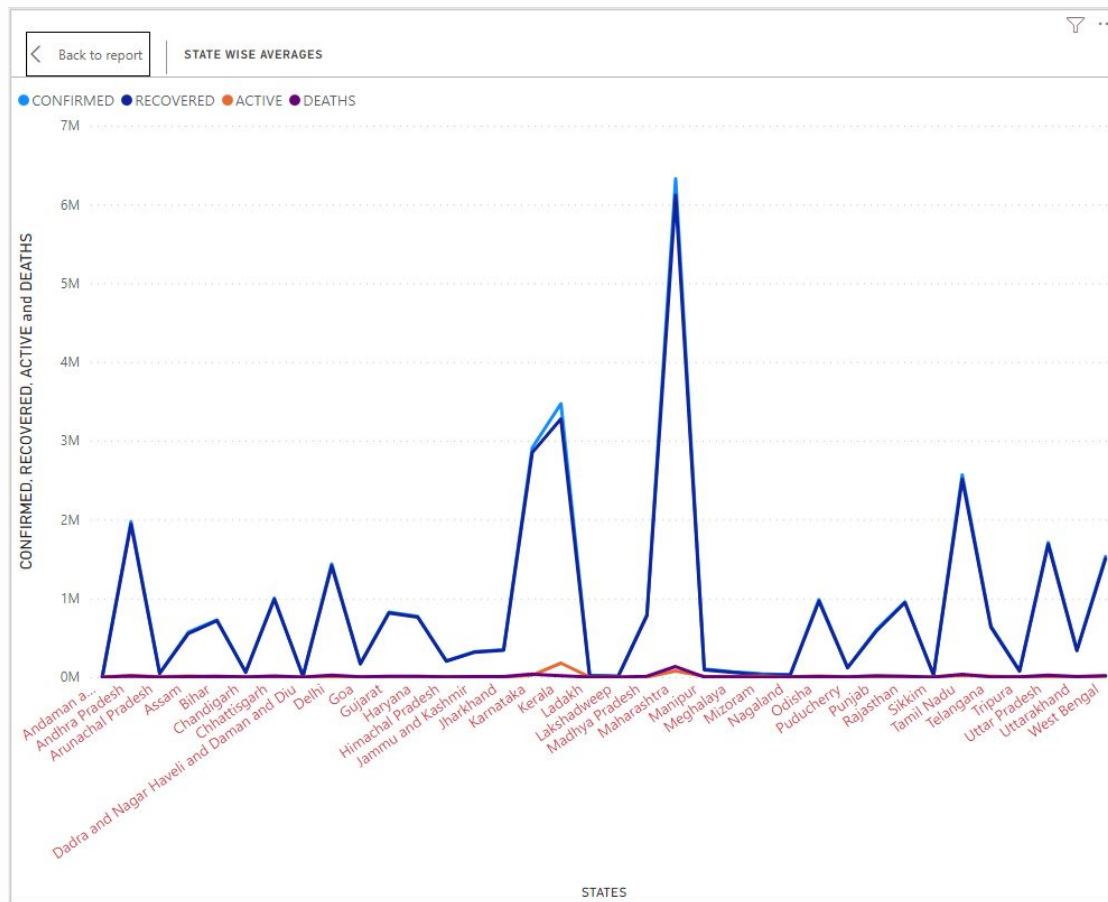
TOP 5 INDIAN STATES WITH MOST CONFIRMED CASES

STATE	CONFIRMED	RECOVERED	ACTIVE	DEATHS
Maharashtra	5,746,892.00	5,395,370.00	256,178.00	95,344.00
Karnataka	2,604,431.00	2,261,590.00	313,751.00	29,090.00
Delhi	1,426,240.00	1,390,963.00	11,040.00	24,237.00
Tamil Nadu	2,096,516.00	1,770,503.00	301,781.00	24,232.00
Uttar Pradesh	1,691,488.00	1,633,947.00	37,044.00	20,497.00
West Bengal	1,376,377.00	1,273,788.00	87,048.00	15,541.00
Chhattisgarh	971,463.00	922,674.00	35,741.00	13,048.00
Andhra Pradesh	1,693,085.00	1,528,360.00	153,795.00	10,930.00
Kerala	2,526,579.00	2,310,385.00	207,379.00	8,815.00
Rajasthan	939,958.00	888,919.00	42,654.00	8,385.00

AVERAGE CONFIRMED CASES BY MONTH IN YEAR 2021



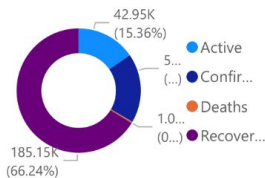
STATE WISE AVERAGES WITH CUMULATIVE PARAMETERS



DASHBOARD - 01 (GLOBAL)

COVID 19 REPORTS (2020-21)	200.24M	130.90M	30.37M	4.26M	Last Updated 05 August 2021
	CONFIRMED	RECOVERED	ACTIVE	DEATHS	

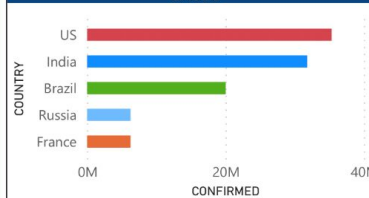
AVERAGE BY CUMULATIVE PARAMETERS



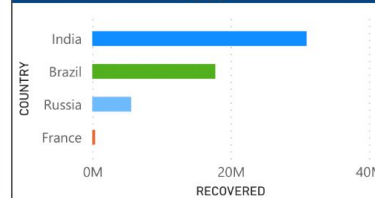
INCIDENT RATE BY COUNTRY



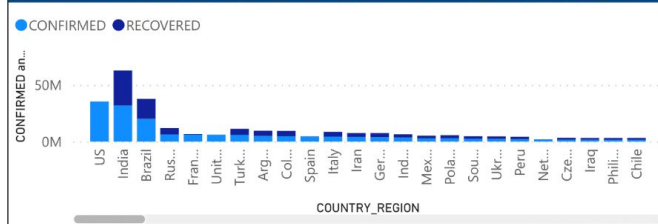
TOP 5 COUNTRIES WITH MOST CONFIRMED CASES



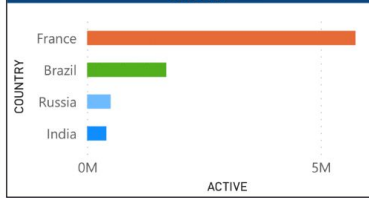
TOP 5 COUNTRIES WITH MOST RECOVERED CASES



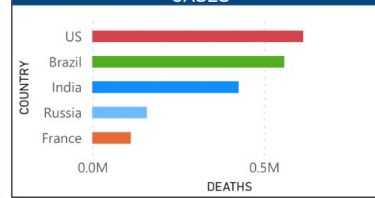
COUNTRY AND STATE WISE COMPARISON OF CONFIRMED AND RECOVERED



TOP 5 COUNTRIES WITH MOST ACTIVE CASES



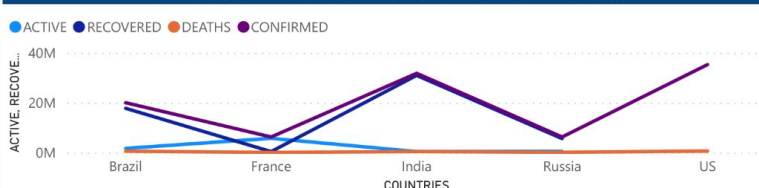
TOP 5 COUNTRIES WITH MOST DEATH CASES



TOP 5 COUNTRIES WITH MOST CONFIRMED CASES IN JULY 2021



TOP 5 COUNTRIES WITH CUMULATIVE PARAMETERS



DASHBOARD - 02 (INDIA)

COVID 19 REPORTS (2020-21) - INDIA

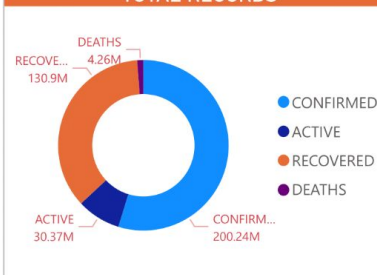
Last Updated

05 August 2021

INCIDENT RATE OF INDIA



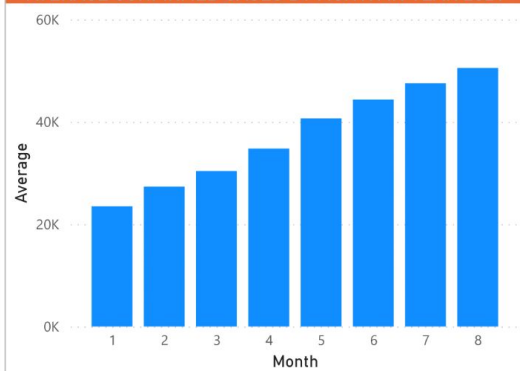
TOTAL RECORDS



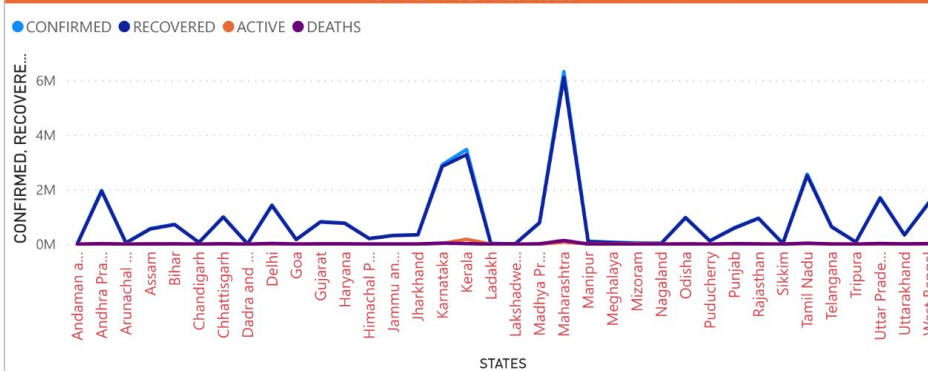
TOP 5 STATES WITH MOST NUMBER OF CASES

STATE	CONFIRMED	RECOVERED	ACTIVE	DEATHS
Maharashtra	63,27,194.00	61,17,560.00	76,224.00	1,33,410.00
Karnataka	29,11,727.00	28,50,717.00	24,330.00	36,680.00
Tamil Nadu	25,67,401.00	25,13,087.00	20,117.00	34,197.00
Delhi	14,36,518.00	14,10,947.00	513.00	25,058.00
Uttar Pradesh	17,08,623.00	16,85,170.00	686.00	22,767.00
West Bengal	15,30,850.00	15,01,925.00	10,745.00	18,180.00
Kerala	34,71,563.00	32,77,788.00	1,76,564.00	17,211.00
Chhattisgarh	10,02,735.00	9,87,298.00	1,906.00	13,531.00
Andhra Pradesh	19,73,996.00	19,40,368.00	20,184.00	13,444.00
Odisha	9,82,181.00	9,63,718.00	12,295.00	6,168.00

AVERAGE CONFIRMED CASES BY MONTH IN YEAR 2021



STATE WISE AVERAGES

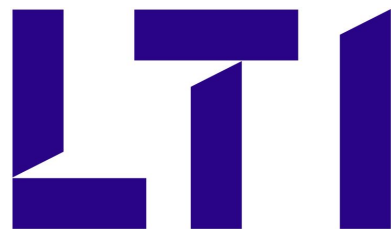


Findings & Challenges - Phase 3

- Error in forming reports due to warehouse consumption error. Fixed this by changing to a smaller warehouse (S) in Snowflake and resuming said warehouse from its suspended state.
- Since there were multiple timestamps for the same day, we transformed our data and changed the 'timestamp' data type to 'date'.
- Countries/Regions can be drilled down to Provinces/States and vice versa.



QUESTIONS?



Let's Solve