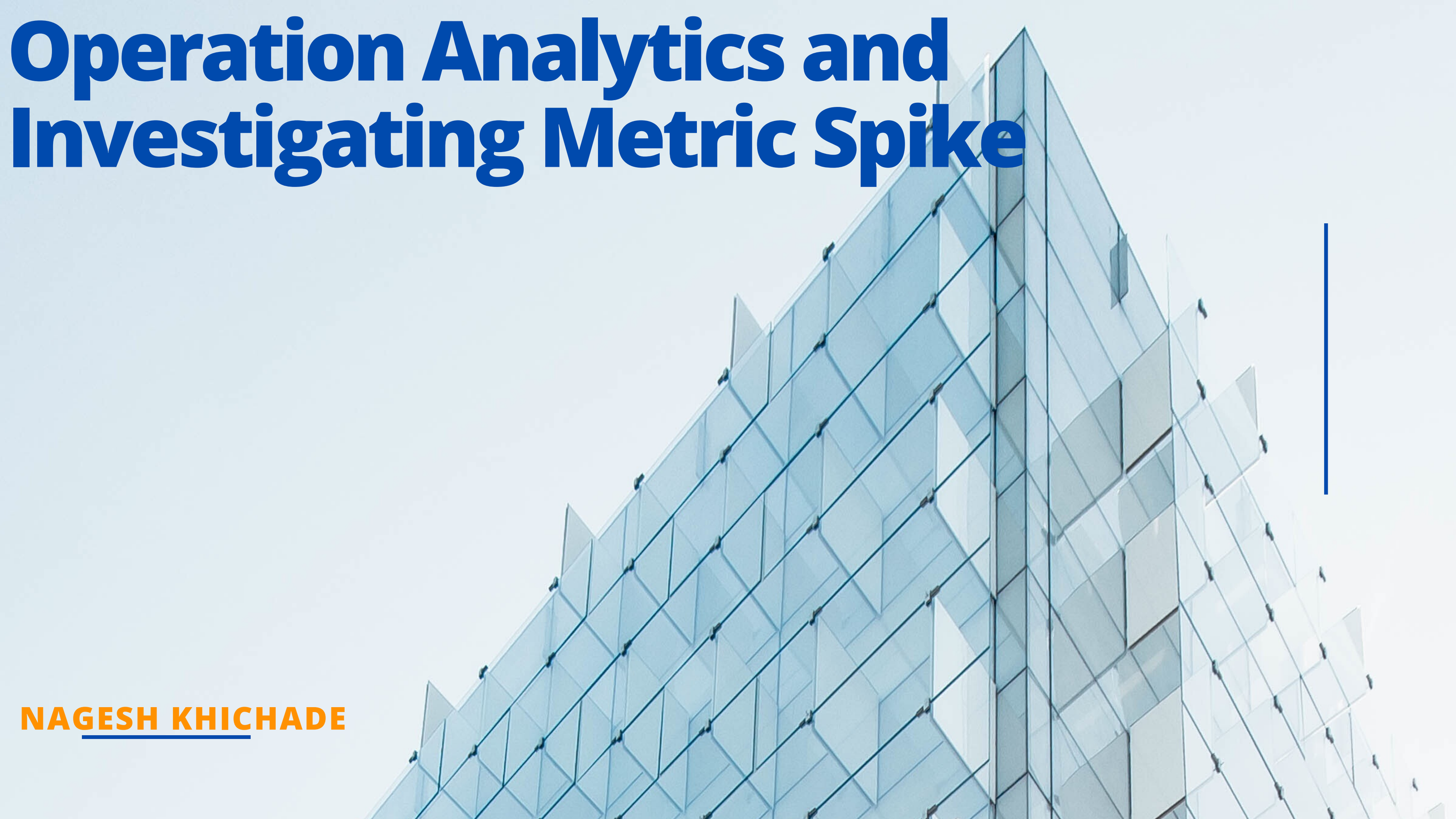
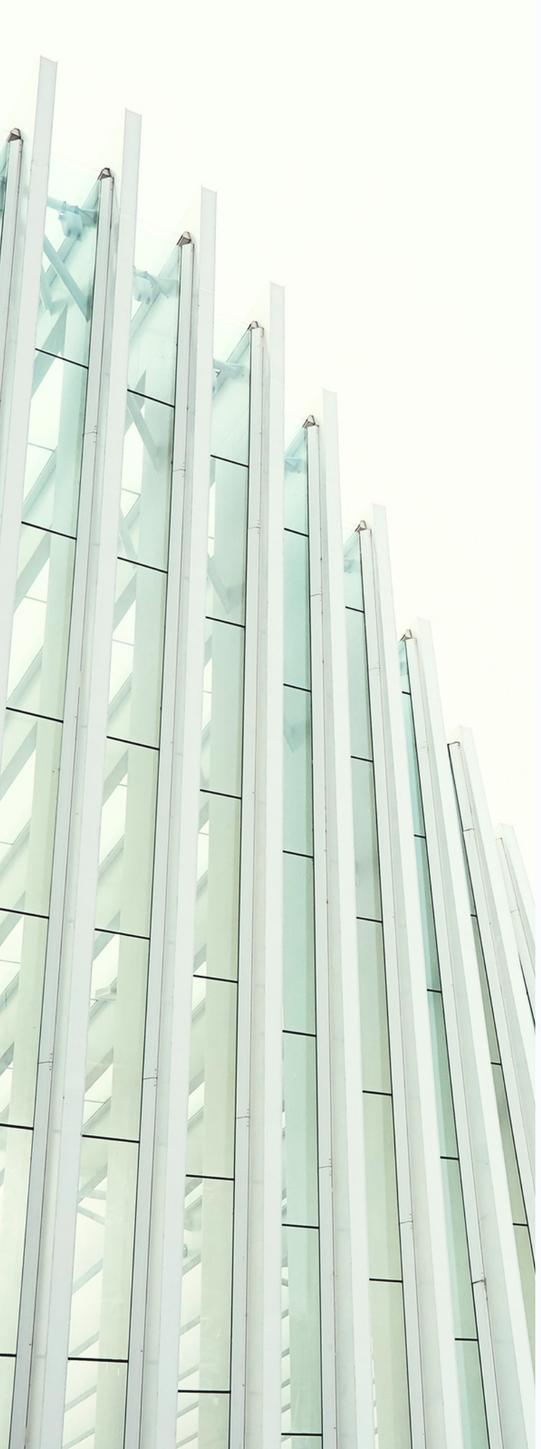
# Operation Analytics and Investigating Metric Spike



NAGESH KHICHADE



# Table of Contents


01 Project Description

02 Approach

03 Tech Stack Used

04 Insights

05 Results

06 Conclusion

# Project Description

Operational analytics is a crucial process that involves analyzing company's end to end operations. This helps in finding the areas of improvement. One of the key aspects are metric spikes. This involves understanding of sudden changes in key metrices

This project's objective is to leverage operational analytics for end-to-end analysis. The primary focus is optimizing workflows, enhancing automation, predicting sales etc. Additionally, this project involve analyzing metric spikes in daily engagement  and insights.

# Approach

## Data Collection & Preparation

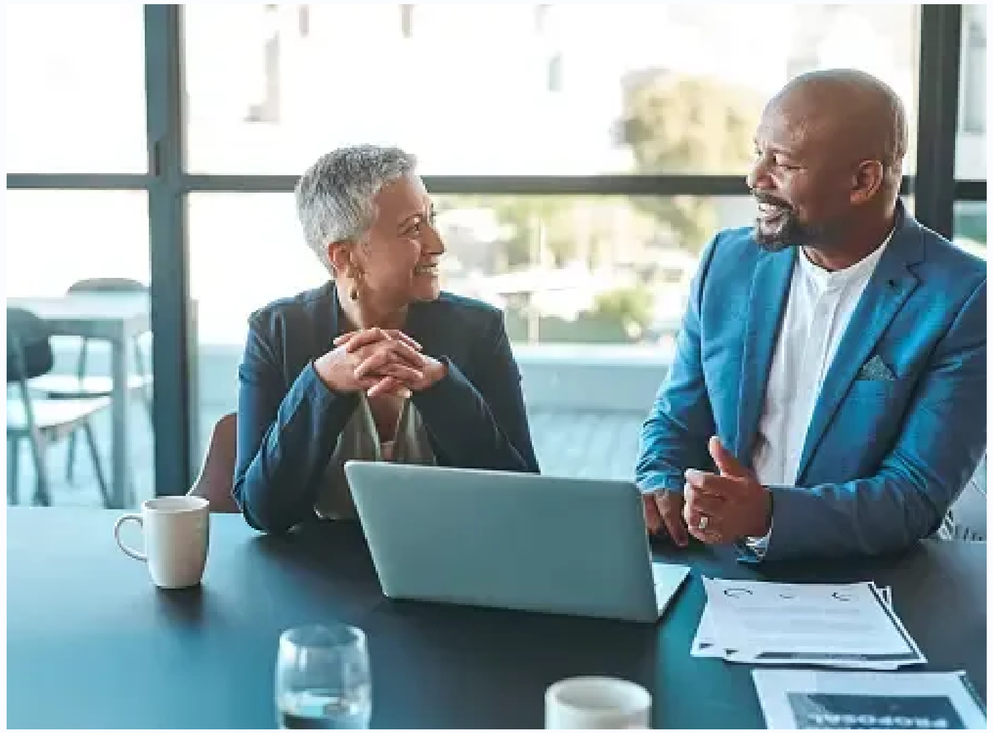
This involves gathering data sets from various departments into csv files. Then I cleaned the data from excel to find missing values.

## Problem Understanding

What needs to be analyzed throgh the key metrices and spikes.

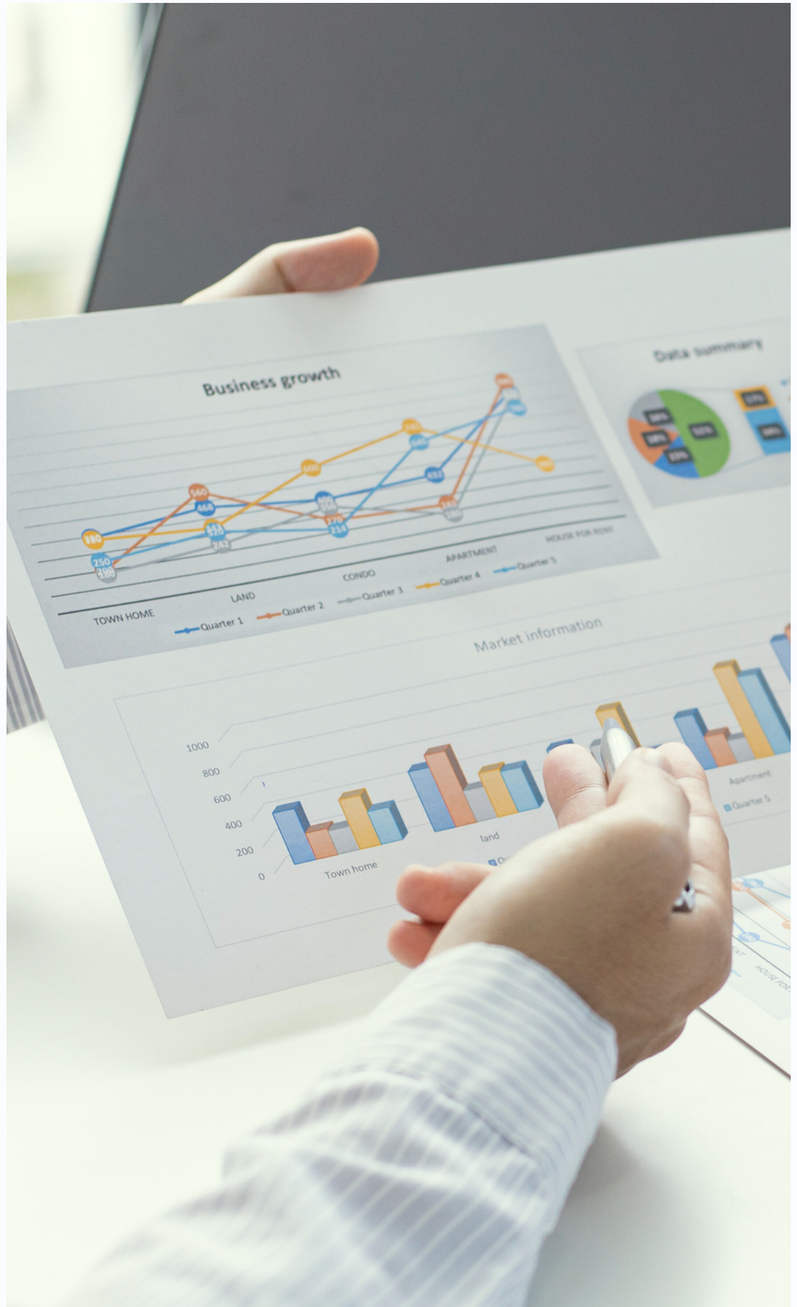
## Solving the Problem

By making use of Excel and Advanced MySQL soled the problems.



# Insights

This projects has helped me in gaining the practical knowledge of Advanced MySQL commands. This experience boosted my confidence in the SQL and also in Excel. I learned how to import the csv files in MySQL workbench. Also the large amount of data how to import and what changes to be made I learned.




## Tech-Stack

MySQL Workbench 8.3 CE
MS-Excel

# CASE STUDY 1

## Task 1: Number of Jobs  reviewed per hour per day in November 2020

Query 1 | SQL File 2* | job_data

Limit to 1000 rows

```sql
## No. of jobs viewed in november 2020
select ds as date,
round((count(job_id)/sum(time_spent))*60*60) as "jobs viewd hour/day" from job_data
WHERE ds BETWEEN "01-11-2020" AND "30-11-2020" group by ds;
```

Result Grid | Filter Rows: | Export:

| date | jobs viewd hour/day |
|---|---|
| 11/30/2020 | 180 |
| 11/29/2020 | 180 |
| 11/28/2020 | 218 |
| 11/27/2020 | 35 |
| 11/26/2020 | 64 |
| 11/25/2020 | 80 |

The number of jobs reviewed in November varies pretty neat with the days with most jobs reviewed at the end o f the month.

This could be because people mostly because users want to find switch from jobs.

# CASE STUDY 1

## Task 2: Calculate the 7-day rolling average of throughput (number of events per second).

```
6
7     ## 7-day rolling average of throughput
8 •   select round((count(event)/sum(time_spent)),2) as weekly_throughput
9     from job_data;
10
11
```

Result Grid | Filter Rows:

| | weekly_throughput |
|---|---|
| ▶ | 0.03 |

Insight:

The 7 day rolling average throughput allows to observe the trends over a time without daily fluctuations. This is more preferred as at a particular day there can be same trends.

# CASE STUDY 1

## Task 3: Percentage share of each language

```
16      # Language Percentage
17  •   select language, round(((count(language)/8)*100),2) as share_of_lang
18      from job_data group by language;
19
```

Result Grid | Filter Rows:

| language | share_of_lang |
|---|---|
| English | 12.50 |
| Arabic | 12.50 |
| Persian | 37.50 |
| Hindi | 12.50 |
| French | 12.50 |
| Italian | 12.50 |

Insight:

The language distribution shows that it is mostly balanced , except for Persian. It is the mostly spoken language as per the data.

# CASE STUDY 1

## Task 3: Display duplicate row count from the table

```
select actor_id, count(actor_id) as tot_count from job_data
group by actor_id having tot_count>1;
```

```
select job_id, count(job_id) as tot_count from job_data
group by job_id having tot_count>1;
```

| ds | job_id | actor_id | event |
|---|---|---|---|
| 11/29/2020 | 23 | 1003 | decision |
| 11/25/2020 | 20 | 1003 | transfer |

| ds | job_id | actor_id | event |
|---|---|---|---|
| 11/29/2020 | 23 | 1003 | decision |
| 11/28/2020 | 23 | 1005 | transfer |
| 11/26/2020 | 23 | 1004 | skip |

Insight:

Here we can see the duplicate row count for 2 entries, actor and job.

# CASE STUDY 2

## Task1: Weekly User Engagement

```
77      ##Calculate the weekly user engagement
78 •    select * from events;
79 •    select extract(week from occurred_at) as weeks,
80      count(distinct user_id) as no_of_users from events_table
81      where event_type="engagement"
82      group by weeks order by weeks;
```

| weeks | no_of_users |
|---|---|
| 17 | 663 |
| 18 | 1068 |
| 19 | 1113 |
| 20 | 1154 |
| 21 | 1121 |
| 22 | 1186 |
| 23 | 1232 |
| 24 | 1275 |
| 25 | 1264 |
| 26 | 1302 |
| 27 | 1372 |
| 28 | 1365 |
| 29 | 1376 |
| 30 | 1467 |
| 31 | 1299 |
| 32 | 1225 |
| 33 | 1225 |
| 34 | 1204 |
| 35 | 104 |

Insight:

User engagement is peaked mostly throughout the weeks as the values shows above 1000. It's comparatively low in week 17 that is only 663.

# CASE STUDY 2

## Task2: User growth Analysis

```sql
##Analyze the growth of users over time for a product.

SELECT year,
    week_number,
    Number_of_users,
    SUM(Number_of_users) OVER (ORDER BY year, week_number) AS cumulative_users
FROM(
    SELECT EXTRACT(YEAR FROM created_at) AS year,
        EXTRACT(WEEK FROM created_at) AS week_number,
        COUNT(DISTINCT user_id) AS Number_of_users
    FROM users
    WHERE state = 'active'
    GROUP BY year, week_number
    ) AS sub;
```

| week_num | year_num | cumulative_sum |
|---|---|---|
| 0 | 2013 | 23 |
| 0 | 2014 | 106 |
| 1 | 2013 | 136 |
| 1 | 2014 | 262 |
| 2 | 2013 | 310 |
| 2 | 2014 | 419 |
| 3 | 2013 | 455 |
| 3 | 2014 | 568 |
| 4 | 2013 | 598 |
| 4 | 2014 | 728 |
| 5 | 2013 | 776 |
| 5 | 2014 | 909 |
| 6 | 2013 | 947 |
| 6 | 2014 | 1082 |
| 7 | 2013 | 1124 |
| 7 | 2014 | 1249 |
| 8 | 2013 | 1283 |
| 8 | 2014 | 1412 |
| 9 | 2013 | 1455 |
| 9 | 2014 | 1588 |
| 10 | 2013 | 1620 |
| 10 | 2014 | 1774 |
| 11 | 2013 | 1805 |
| 11 | 2014 | 1935 |
| 36 | 2013 | 7911 |
| 37 | 2013 | 7996 |
| 38 | 2013 | 8086 |
| 39 | 2013 | 8170 |
| 40 | 2013 | 8257 |
| 41 | 2013 | 8330 |
| 42 | 2013 | 8429 |
| 43 | 2013 | 8518 |
| 44 | 2013 | 8614 |
| 45 | 2013 | 8705 |
| 46 | 2013 | 8793 |
| 47 | 2013 | 8895 |
| 48 | 2013 | 8992 |
| 49 | 2013 | 9108 |
| 50 | 2013 | 9232 |
| 51 | 2013 | 9334 |
| 52 | 2013 | 9381 |

Insight:

This show very positive growth of users over the span of year. This is amazing to have such growth.

# CASE STUDY 2

## Task3: Analyze the retention of users on a weekly basis after signing up for a product.

```
SELECT * FROM events;
SELECT extract(week FROM occurred_at) AS weeks, COUNT(DISTINCT user_id) AS COUNT_of_users
FROM events
WHERE event_type="signup_flow" AND event_name="complete_signup"
GROUP BY weeks
ORDER BY weeks;
```

| weeks | COUNT_of_users |
|---|---|
| 17 | 72 |
| 18 | 163 |
| 19 | 185 |
| 20 | 176 |
| 21 | 183 |
| 22 | 196 |
| 23 | 196 |
| 24 | 229 |
| 25 | 207 |
| 26 | 201 |
| 27 | 222 |
| 28 | 215 |
| 29 | 221 |
| 30 | 238 |
| 31 | 193 |
| 32 | 245 |
| 33 | 261 |
| 34 | 259 |
| 35 | 18 |

Insight:

User growth is generally positive in all ways with some low score on few of the weeks. This could be due to a busy week or it can be of holidyas.

# CASE STUDY 2

## Task4: Weekly Engagement Per Device

```
103 •    SELECT week(occurred_at) as Weeks,
104      device,
105      count(distinct user_id)as User_engagement
106      FROM events
107      GROUP BY device,
108      week(occurred_at)
109      ORDER BY week(occurred_at);
```

| device | User_engagement |
|---|---|
| acer aspire desktop | 198 |
| acer aspire notebook | 338 |
| amazon fire phone | 89 |
| asus chromebook | 355 |
| dell inspiron desktop | 360 |
| dell inspiron notebook | 677 |
| hp pavilion desktop | 339 |
| htc one | 196 |
| ipad air | 478 |
| ipad mini | 292 |
| iphone 4s | 409 |
| iphone 5 | 1025 |
| iphone 5s | 626 |
| kindle fire | 205 |
| lenovo thinkpad | 1309 |
| mac mini | 150 |
| macbook air | 950 |
| macbook pro | 1952 |

Insight:

Engagement across various devices differ variously.  Most of the users are Mobile phone users or the latest technology electronic gadgets. The devices which show lower engagement rate must be focused on developing the better UI/UX interfaces.

# CASE STUDY 2

## Task5: Analyze how users are engaging with the email service.

```sql
##Engageent Analysis

SELECT week(occured_at) AS week,
    count(CASE WHEN action = 'email_open' THEN user_id ELSE null END) AS email_open,
    count(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE null END) AS email_clickthrough,
    count(CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE null END) AS sent_weekly_digest,
    count(CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE null END) AS sent_reengagement
FROM emailEvents
GROUP BY 1, week
ORDER BY 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| week | email_open | email_clickthrough | sent_weekly_digest | sent_reengagement |
|---|---|---|---|---|
| NULL | 20459 | 9010 | 57267 | 3653 |

Insight:

Most of the email are opened but their clickthrough rates are very low. This could be the indication of less engaging content of the emails.
Reengagement shows that the conversation is started but not converted into the clickthrough.

# Thank You


Phone

**8862086486**

LinkedIn

**https://www.linkedin.com/in/nagesh-khichade-210875212/**

Email

**nageshkhichade00@gmail.com**