

# OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

BY AKHILESH DHURI



# CONTENT

---

- Introduction
- Approach
- Tech-Stack used
- Insights
- Result

# INTRODUCTION

---

Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations to identify areas for improvement. Investigating metric spikes is one of the key aspects of Operational Analytics. It involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales.

We will be working on two case studies in this assignment, focusing on Operation Analytics and Investigating Metric Spike. Our goal is to use advanced SQL skills to analyze the data and provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

For Case Study 1, we will be working with a table named `job_data` and completing tasks such as calculating the number of jobs reviewed per hour, analyzing the 7-day rolling average of throughput, and identifying duplicate rows in the data.

For Case Study 2, we will be working with three tables named `users`, `events`, and `email events`. Our tasks will include calculating the weekly user engagement, analyzing the growth of users over time for a product, analyzing the weekly retention of users based on their sign-up cohort, measuring the activeness of users on a weekly basis per device, and analyzing how users are engaging with the email service.

Through these tasks, we will gain a deeper understanding of Operation Analytics and Investigating Metric Spike. We will also provide insights and interpretations of the results obtained from our queries.

# APPROACH

---

We will follow the following approach for Operation Analytics and Investigating Metric Spike:

- Analyze the datasets provided and identify the relevant columns and tables required to create database and table and import the data set into our tables.
- Understand the problem statement and the tasks assigned in each case study.
- Use SQL and MySQL Workbench.
- Write SQL queries to extract the required data.
- Organize the report in a clear manner, using headings and subheadings to structure the content.
- Present the output data in the report.

# TECH-STACK USED

---

For this project we are going to use following tech:

- MySQL Server 8.0
- MySQL Workbench (UI client for MySQL Server)

We have chosen MySQL Workbench because it is a popular and powerful tool for working with MySQL databases, and it provides a user-friendly interface for writing and executing SQL queries.

# INSIGHTS

While using the mentioned tech-stack for this project, we have gained insights for following analysis:

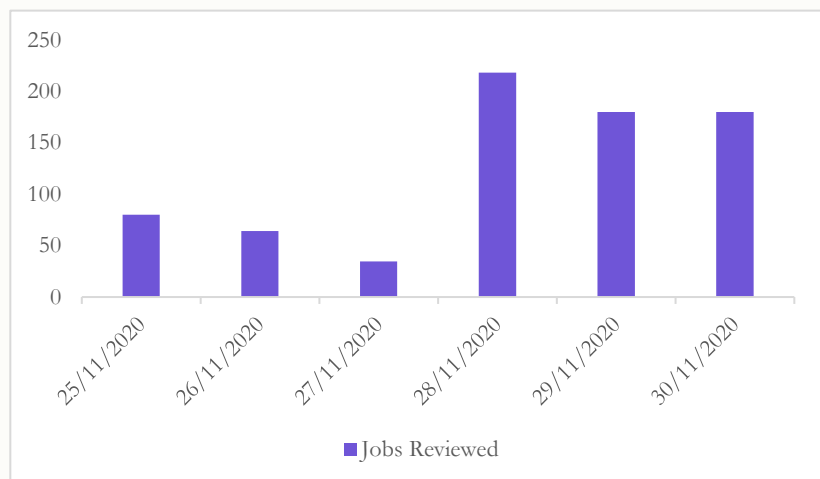
## ❖ Case Study 1: Job Data Analysis

### 1. JOBS REVIEWED OVER TIME:

Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.

```
SELECT
  ds AS days,
  COUNT(job_id) / (SUM(time_spent) / 3600) AS jobs_reviewed
FROM
  job_data
GROUP BY ds;
```

days	jobs_reviewed
2020-11-30	180.0000
2020-11-29	180.0000
2020-11-28	218.1818
2020-11-27	34.6154
2020-11-26	64.2857
2020-11-25	80.0000



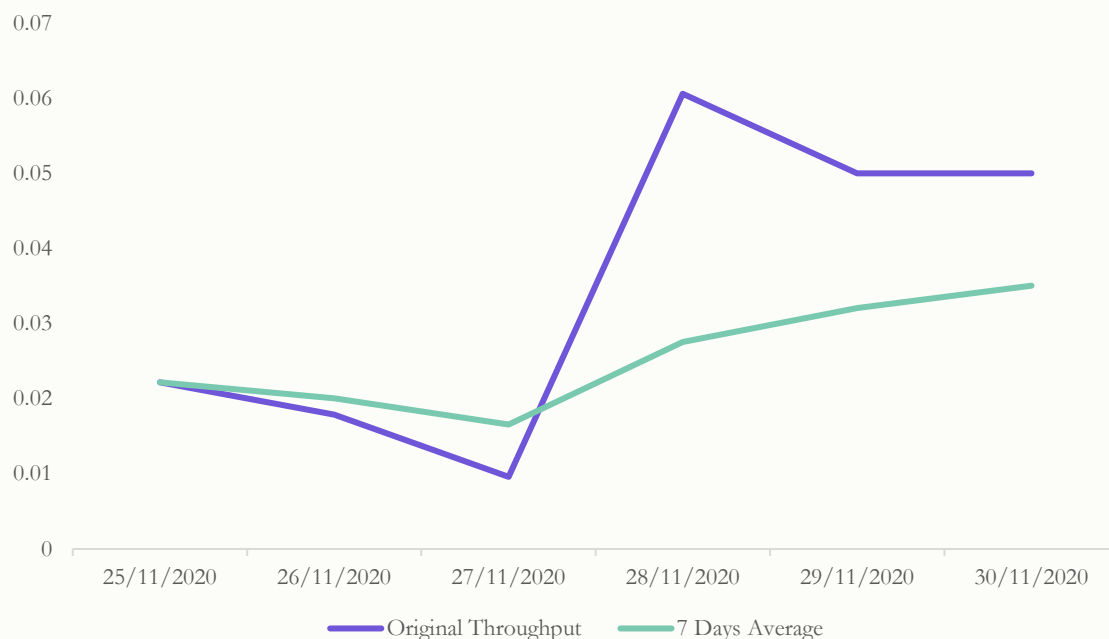
The highest number of jobs reviewed per hour was on November 28, 2020, with 218 jobs reviewed per hour. The lowest number of jobs reviewed per hour was on November 27, 2020, with only 34 jobs reviewed per hour.

## 2. THROUGHPUT ANALYSIS:

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

```
SELECT new_data.ds AS day,  
       new_data.throughput AS original_throughput,  
       AVG(new_data.throughput) OVER (ORDER BY new_data.ds ROWS BETWEEN 6 PRECEDING AND  
       CURRENT ROW) AS 7_day_avg_throughput  
FROM  
(SELECT ds, COUNT(job_id) / SUM(time_spent) AS throughput FROM job_data GROUP BY ds) AS  
new_data  
GROUP BY new_data.ds;
```

day	original_throughput	7_day_avg_throughput
2020-11-25	0.0222	0.02220000
2020-11-26	0.0179	0.02005000
2020-11-27	0.0096	0.01656667
2020-11-28	0.0606	0.02757500
2020-11-29	0.0500	0.03206000
2020-11-30	0.0500	0.03505000



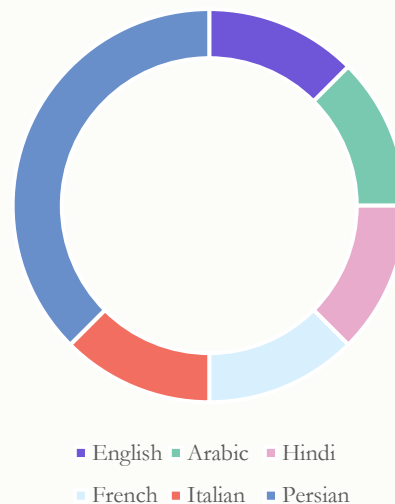
The given output shows that the throughput (number of events per second) of job reviews changes a lot from day to day. For example, on November 28th, the throughput was much higher than on November 27th. However, when we look at the 7-day rolling average of the throughput, we see a steady result. From November 25th to November 29th, the average throughput gradually increased, but it slightly decreased on November 30th. This information can help us understand how the throughput of job reviews changes over time and make decisions about resource allocation and scheduling.

### 3. LANGUAGE SHARE ANALYSIS:

Objective: Calculate the percentage share of each language in the last 30 days.

```
SELECT language, ROUND(100 * COUNT(*) / sum(count(*)) OVER(), 2) AS Percentage
FROM job_data
WHERE ds between '2020-11-01' and '2020-11-30'
GROUP BY language
```

Language	Percentage
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50



The SQL query output shows that in November 2020, most of the job postings were in Persian language, accounting for 37.50% of the total job postings. The other languages had an equal share of 12.50% each. This information can be used to hire candidates with language skills in these languages.

## 4. DUPLICATE ROWS DETECTION:

Objective: Identify duplicate rows in the data.

```
SELECT ds, job_id, actor_id, event, language, time_spent, org, COUNT(*) as count
FROM job_data
GROUP BY ds, job_id, actor_id, event, language, time_spent, org
HAVING COUNT(*) > 1;
```

No duplicates were available in the output.

## ❖ Case Study 2: Investigating Metric Spike

### 1. WEEKLY USER ENGAGEMENT:

Objective: Measure the activeness of users on a weekly basis.

```
SELECT
    WEEK(occurred_at) AS Week,
    COUNT(DISTINCT user_id) AS User_engagement
FROM
    events
GROUP BY
    Week;
```

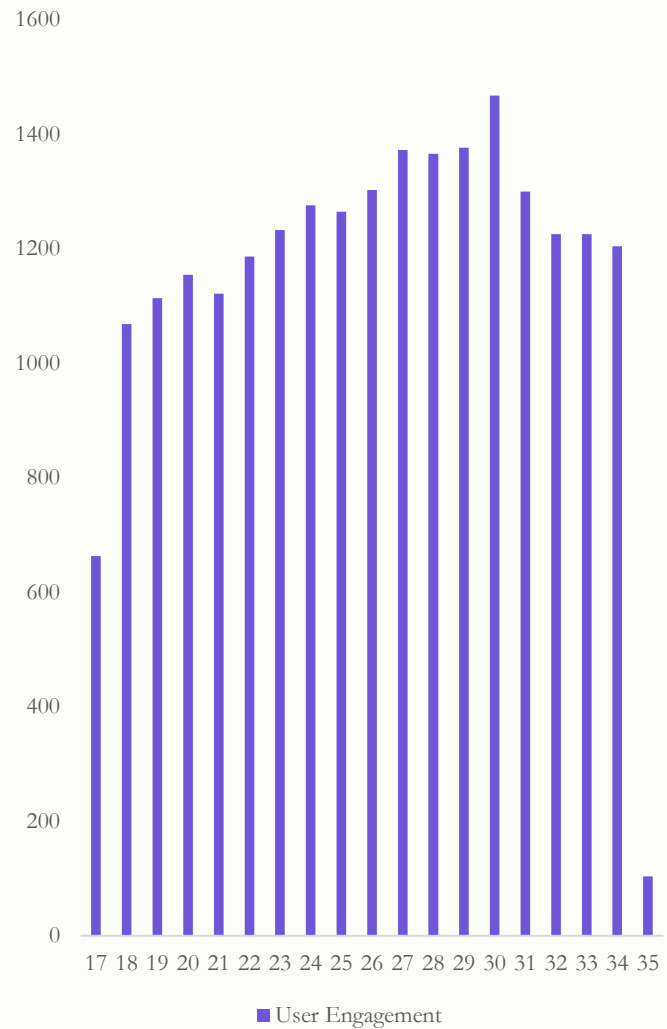
The analysis of the output below can provide insights into the activeness of users on a weekly basis. Here are some observations based on the given output:

- The user engagement increased from week 17 to week 30, with some difficulties.
- The highest user engagement was observed in week 30, with 1467 users being active.
- The user engagement started to decline from week 31 and reached its lowest point in week 35, with only 104 users being active.

The decline in user engagement from week 31 to week 35 suggests that there might be some issues that need to be addressed to maintain user engagement. Further analysis is required to identify the reasons behind the decline and take appropriate actions to improve user engagement.



Week	User_engagement
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

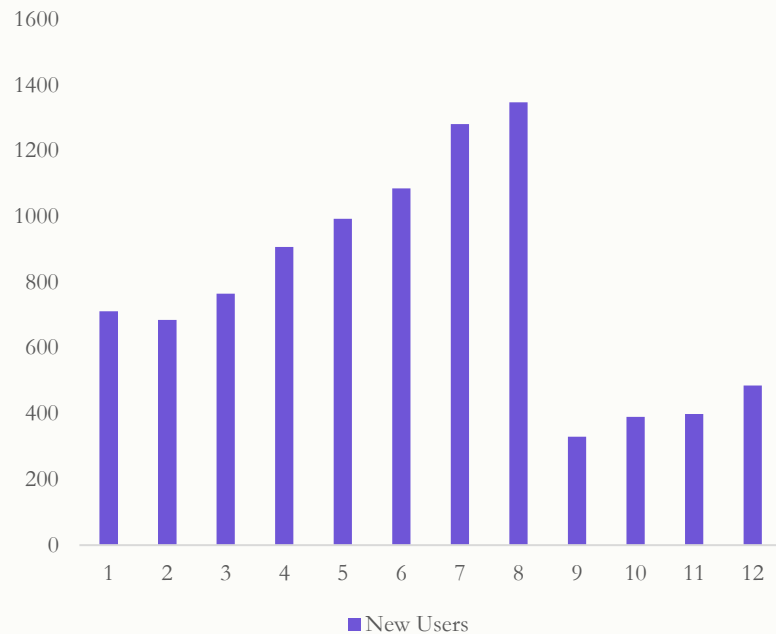


## 2. USER GROWTH ANALYSIS:

Objective: Analyze the growth of users over time for a product.

```
SELECT
  MONTH(created_at) AS month,
  COUNT(DISTINCT user_id) AS new_users
FROM
  users
GROUP BY month;
```

month	new_users
1	712
2	685
3	765
4	907
5	993
6	1086
7	1281
8	1347
9	330
10	390
11	399
12	486



The given output shows the number of new users who signed up each month for the product.

- The number of new users who signed up each month varies from 330 to 1347.
- There is a general trend of increasing user growth over time, with a few exceptions.
- Months 1 to 7 show a steady increase in user growth, with a sharp increase in month 7.
- Months 8 and 9 show a sudden drop in user growth.
- Months 10 to 12 show a slight increase in user growth, but not as much as in the earlier months.

### 3. WEEKLY RETENTION ANALYSIS:

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

```
SELECT DISTINCT
  WEEK(e.occurred_at) AS week,
  u.user_id,
  e.occurred_at,
  u.activated_at
FROM
  users u
  JOIN
  events e ON u.user_id = e.user_id
```

WHERE

```
u.activated_at > '2014-05-01'  
AND e.event_name = 'login'  
GROUP BY week;
```

week	user_id	occurred_at	activated_at
17	11768	2014-05-01 08:03:00	2014-05-01 08:03:00
18	11775	2014-05-09 12:21:00	2014-05-01 16:38:00
23	11778	2014-06-09 07:06:00	2014-05-01 18:49:00
21	11778	2014-05-28 14:10:00	2014-05-01 18:49:00
19	11787	2014-05-13 15:59:00	2014-05-01 18:23:00
20	11799	2014-05-20 07:56:00	2014-05-01 12:07:00
22	11901	2014-06-03 15:12:00	2014-05-03 10:32:00
24	12741	2014-06-21 07:57:00	2014-05-20 17:48:00
25	12882	2014-06-22 18:46:00	2014-05-22 09:13:00
29	13317	2014-07-20 12:18:00	2014-05-30 18:49:00
26	13441	2014-07-04 10:33:00	2014-06-02 13:42:00
30	13743	2014-07-30 06:10:00	2014-06-07 08:39:00
28	13757	2014-07-18 20:39:00	2014-06-08 15:41:00
27	13942	2014-07-12 08:53:00	2014-06-11 22:40:00
31	14237	2014-08-04 23:54:00	2014-06-16 15:30:00
32	14848	2014-08-15 14:03:00	2014-06-26 17:14:00
33	15469	2014-08-20 17:48:00	2014-07-07 15:14:00
34	15570	2014-08-27 11:19:00	2014-07-08 11:27:00
35	18864	2014-08-31 17:06:00	2014-08-28 10:35:00

## 4. WEEKLY ENGAGEMENT PER DEVICE:

Objective: Measure the activeness of users on a weekly basis per device.

```
SELECT
  WEEK(occurred_at) AS Week,
  device,
  COUNT(DISTINCT user_id) AS User_Engagement
FROM
  events
GROUP BY Week , device
```

### [Output](#)

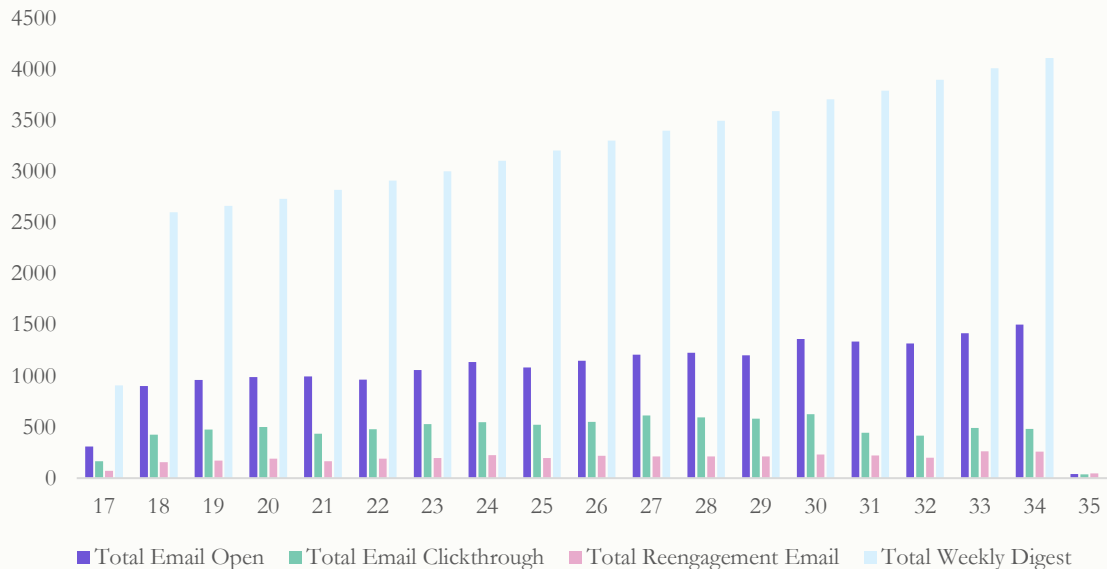
## 5. EMAIL ENGAGEMENT ANALYSIS:

Objective: Analyze how users are engaging with the email service.

```
SELECT WEEK(occurred_at) AS Week,
  COUNT(DISTINCT user_id ) AS total_actions,
  COUNT(DISTINCT CASE WHEN action = 'email_open' THEN user_id END) AS
total_email_open,
  COUNT(DISTINCT CASE WHEN action = 'email_clickthrough' THEN user_id END) AS
total_email_clickthrough,
  COUNT(DISTINCT CASE WHEN action = 'sent_reengagement_email' THEN user_id END) AS
total_reengagement_email,
  COUNT(DISTINCT CASE WHEN action = 'sent_weekly_digest' THEN user_id END) AS
total_weekly_digest
FROM email_events
GROUP BY Week;
```

Week	total_actions	total_email_open	total_email_clickthrough	total_reengagement_email	total_weekly_digest
17	981	310	166	73	908
18	2714	900	425	157	2602
19	2787	961	476	173	2665
20	2874	989	501	191	2733
21	2926	996	436	164	2822
22	3029	965	478	192	2911
23	3134	1057	529	197	3003
24	3254	1136	549	226	3105
25	3343	1084	524	196	3207
26	3439	1149	550	219	3302
27	3543	1207	613	213	3399
28	3641	1228	594	213	3499
29	3734	1201	583	213	3592
30	3866	1363	625	231	3706
31	3950	1338	444	222	3793
32	4023	1318	416	200	3897
33	4200	1417	490	264	4012
34	4294	1502	481	261	4111
35	48	41	38	48	0

## Email Engagement Analysis



The given output shows the email engagement metrics for each week, including the total number of actions, total email opens, total email clickthrough, total re-engagement emails, and total weekly digests.

- The total number of actions increased from week 17 to week 34, indicating overall engagement with the email service is increasing.
- The total number of email opens and clickthrough also increased over time, which is a positive sign for the email service.
- The number of re-engagement emails remained constant over time, indicating that the email service can retain users who may have become inactive.
- The number of weekly digests decreased significantly in week 35, which may indicate a problem with the weekly digest feature or a change in user behavior.

# RESULT

---

Through this project, working as a Data Analyst, I have achieved a better understanding of Operational Analytics and its importance in analyzing a company's end-to-end operations. I have learned how to investigate metric spikes and derive valuable insights from the data to answer questions posed by different departments within the company. I have also improved my SQL skills and learned how to calculate the number of jobs reviewed per hour for each day in November 2020, the 7-day rolling average of throughput, the percentage share of each language over the last 30 days, and how to display duplicate rows from the job\_data table. Additionally, I have learned how to measure the activeness of users on a weekly basis, analyze the growth of users over time for a product, analyze the retention of users on a weekly basis after signing up for a product, measure the activeness of users on a weekly basis per device, and analyze how users are engaging with the email service.

This project has contributed to my understanding of Operational Analytics Investigating Metric Spike and how it can help identify areas for improvement within a company. It has also improved my decision-making skills based on the insights derived from the data. By analyzing the data, I can provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

# THANK YOU