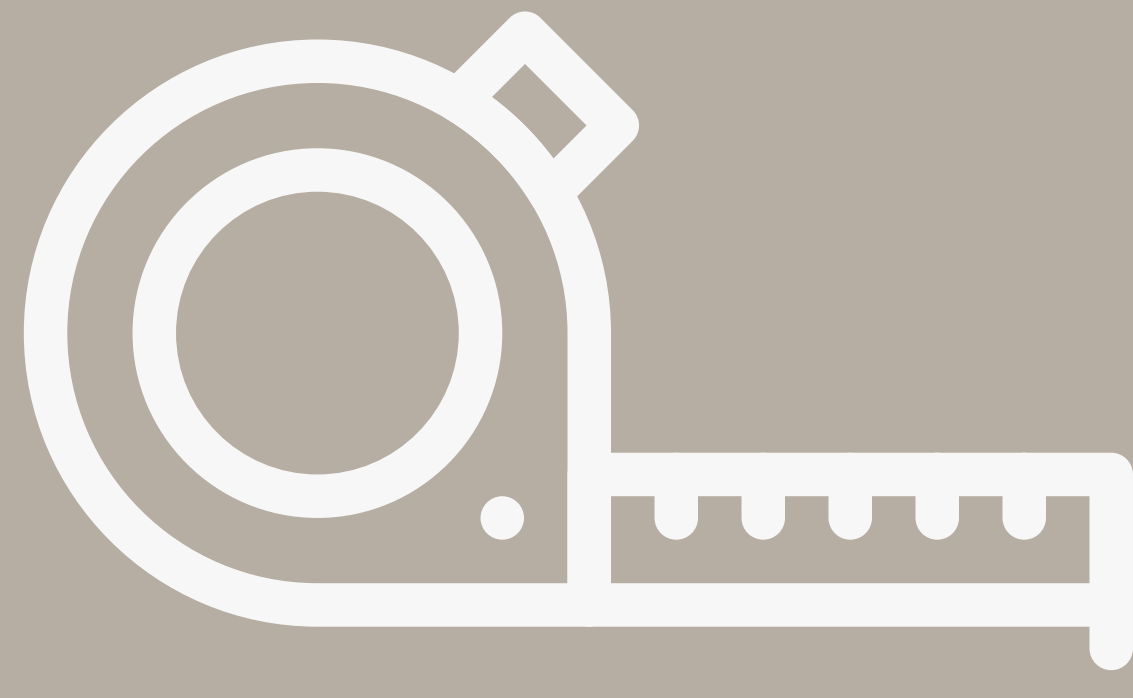

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Project report

Shubhangi Chaudhary



Content

- Project description
- Approach
- Tech Stack Used
- Job Data Analysis
- Investigating Metric Spike
- Insights
- Results
- Thank You





Project Description

Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company. Working closely with various teams, such as operations, support, and marketing, valuable insights can be derived from the data collected.

In this project, with various datasets and tables, and we will derive insights from the given data to answer questions posed by different departments within the company. The 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.

Approach

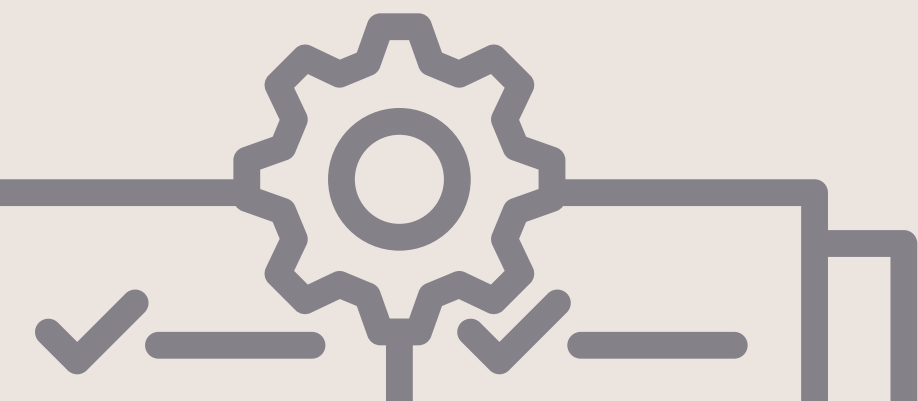
The basic steps involved the creation of the many databases and it was followed by effectively addressing the interaction and relevant behaviour, the focus was on the following:

Case Study 1: Job Data Analysis

- Jobs Reviewed Over Time
- Throughput Analysis
- Language Share Analysis
- Duplicate Rows Detection

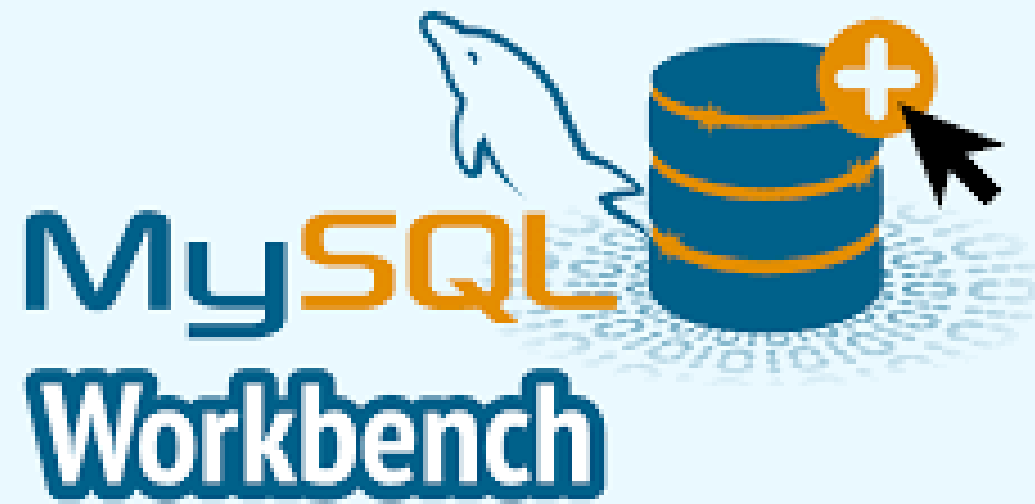
Case Study 2: Investigating Metric Spike

- Weekly User Engagement
- User Growth Analysis
- Weekly Retention Analysis
- Weekly Engagement Per Device
- Email Engagement Analysis



Tech-Stack Used

MySQL Workbench and Ms-Excel were used to carry out the following analysis. It allowed for the approach to be simpler and the results could be achieved in a comprehensive and user-friendly way.



Job Data Analysis

Jobs Reviewed Over Time: To calculate the number of jobs reviewed per hour for each day in November 2020.

1 • use jobs;

2 • select * from job_data;

3 /*To calculate the number of jobs reviewed per hour for each day in November 2020.*/

4 • select ds as date,

5 round((count(job_id)/sum(time_spent))*3600) as "jobs viewd per hour per day" from job_data

6 WHERE month(ds)=11

7 group by ds;

Result Grid

↕

Filter Rows:

Export:

Wrap Cell Content:

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

On November 11, 2010, a total of 218 jobs were reviewed, the most among all days.

Job Data Analysis

Throughput Analysis: To calculate the 7-day rolling average of throughput (number of events per second).

```
9      /*To calculate the 7-day rolling average of throughput*/
10 •   select round((count(event)/sum(time_spent)),2) as weekly_throughput
11     from job_data;
12     #calc daily metric throughput
13 •   select ds as date, round((count(event)/sum(time_spent)),2) as daily_throughput
14     from job_data group by date;
15
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	date	daily_throughput
▶	2020-11-30	0.05
	2020-11-29	0.05
	2020-11-28	0.06
	2020-11-27	0.01
	2020-11-26	0.02
	2020-11-25	0.02

Compared to the daily metric, the 7-day rolling average(0.03) for throughput gives a clearer trend over time.

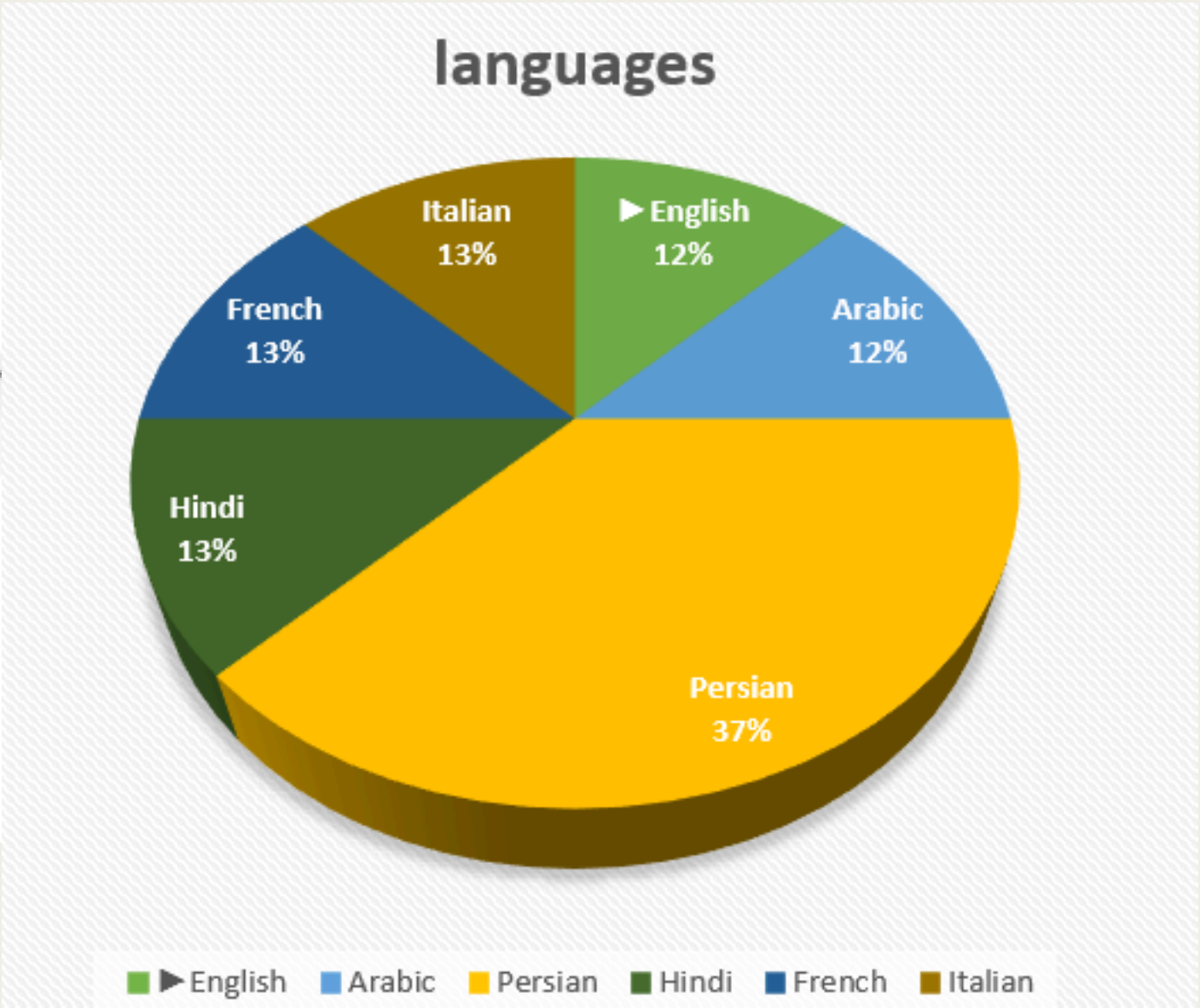


Job Data Analysis

Language Share Analysis: To calculate the percentage share of each language in the last 30 days.

```
16  /*To calculate the percentage share of each language in the last 30 days.*/
17  •  select language, round(((count(language)/8)*100),2) as percentage_share_per_language
18  from job_data group by language;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
language	percentage_share_per_language		
English	12.50		
Arabic	12.50		
Persian	37.50		
Hindi	12.50		
French	12.50		
Italian	12.50		



Among all languages, Persian has the largest percentage share (37.50%).

Job Data Analysis



Duplicate Rows Detection: To identify duplicate rows in the data.


20 /*To identify duplicate rows in the data.*/


21 • select actor_id, count(actor_id) as total_count from job_data

22 group by actor_id having total_count>1;

Result Grid

  Filter Rows:

Export: 

Wrap Cell Content: 

	actor_id	total_count
▶	1003	2

A duplicate row was present.

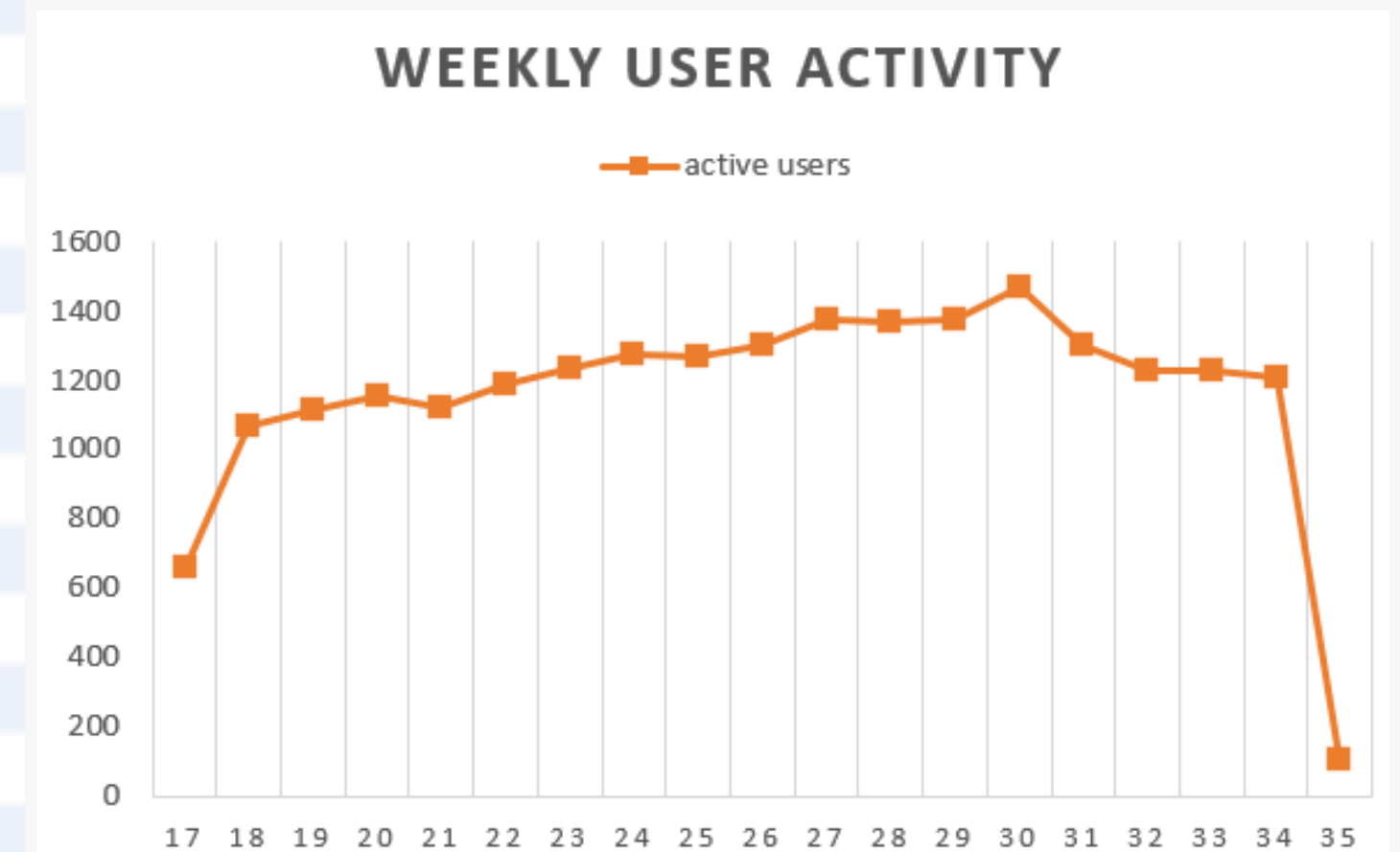


Investigating Metric Spike

Weekly User Engagement: To measure the activeness of users on a weekly basis.

```
/*To measure the activeness of users on a weekly basis.*/  
SELECT  
    *  
FROM  
    events_table;  
SELECT  
    EXTRACT(WEEK FROM occurred_at) AS weeks,  
    COUNT(DISTINCT user_id) AS users_numbers  
FROM  
    events_table  
WHERE  
    event_type = 'engagement'  
GROUP BY weeks  
ORDER BY weeks;
```

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



Week 35 had the fewest active users, while Week 30 had the most.

Investigating Metric Spike

User Growth Analysis: To analyze the growth of users over time for a product.

```
/*To analyze the growth of users over time for a product.*/
```

```
select week_num, year_num,
```

```
sum(active_users) over (order by week_num, year_num  
rows between unbounded preceding and current row) as cumulative_
```

```
from (  
select extract(week from activated_at) as week_num,  
extract(year from activated_at) as year_num,  
count(distinct user_id) as active_users from users_table  
where state= "active"  
group by year_num, week_num  
order by year_num, week_num) as alias;
```

week_num	year_num	cumulative_
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

The number of users for a product did show growth.

Investigating Metric Spike

Weekly Retention Analysis: To analyze the retention of users on a weekly basis after signing up for a product.

```
/*To analyze the retention of users on a weekly basis after signing up for a product.*/  
SELECT  
    *  
FROM  
    events_table;  
SELECT  
    EXTRACT(WEEK FROM occurred_at) AS weeks,  
    COUNT(DISTINCT user_id) AS no_of_users  
FROM  
    events_table  
WHERE  
    event_type = 'signup_flow'  
    AND event_name = 'complete_signup'  
GROUP BY weeks  
ORDER BY weeks;
```

weeks	no_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

User growth was at its highest in week 33 and its lowest in week 35.

Investigating Metric Spike

Weekly Engagement Per Device: To measure the activeness of users on a weekly basis per device.

```
/*To measure the activeness of users on a weekly basis per device.*/
SELECT
    *
FROM
    events_table;

SELECT
    device,
    EXTRACT(WEEK FROM occurred_at) AS weeks,
    COUNT(DISTINCT user_id) AS no_of_users
FROM
    events_table
WHERE
    event_type = 'engagement'
GROUP BY device , weeks
ORDER BY weeks;
```

device	weeks	no_of_users
acer aspire desktop	17	9
acer aspire notebook	17	20
amazon fire phone	17	4
asus chromebook	17	21
dell inspiron desktop	17	18
dell inspiron notebook	17	46
hp pavilion desktop	17	14
htc one	17	16
ipad air	17	27
ipad mini	17	19
iphone 4s	17	21
iphone 5	17	65
iphone 5s	17	42
kindle fire	17	6
lenovo thinkpad	17	86
mac mini	17	6
macbook air	17	54
macbook pro	17	143
nexus 10	17	16
nexus 5	17	40
nexus 7	17	18
nokia lumia 635	17	17

Week 30 had the most active users on the device MacBook Pro, while week 35 had the fewest active users on acer aspire desktop.

Investigating Metric Spike

Email Engagement Analysis: To analyze how users are engaging with the email service.

```
/*To analyze how users are engaging with the email service.*/
SELECT
    *
FROM
    email_events_table;
SELECT
    COUNT(action) AS action_count, action
FROM
    email_events_table
GROUP BY action;
SELECT
    (SUM(CASE
        WHEN email_category = 'email_opened' THEN 1
        ELSE 0
    END) / SUM(CASE
        WHEN email_category = 'email_sent' THEN 1
        ELSE 0
    END)) * 100 AS open_rate,
    (SUM(CASE
        WHEN email_category = 'email_clickthrough' THEN 1
        ELSE 0
    END) / SUM(CASE-----
        WHEN email_category = 'email_sent' THEN 1
        ELSE 0
    END)) * 100 AS click_rate
FROM
```

```
(SELECT
    *,
    CASE
        WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email') THEN ('email_sent')
        WHEN action IN ('email_open') THEN ('email_opened')
        WHEN action IN ('email_clickthrough') THEN ('email_clickthrough')
    END AS email_category
FROM
    email_events_table) AS alias;
```

Email engagement was highest in week 34, while lowest in week 35.

Insights

Job Data Analysis

- On November 11, 2010, a total of 218 jobs were reviewed, the most among all days.
- Compared to the daily metric, the 7-day rolling average(0.03) for throughput gives a clearer trend over time.
- Among all languages, Persian has the largest percentage share (37.50%).
- A duplicate row was present.

Investigating Metric Spike

- Week 35 had the fewest active users, while Week 30 had the most.
- The number of users for a product did show growth.
- User growth was at its highest in week 33 and its lowest in week 35.
- Week 30 had the most active users on the device MacBook Pro, while week 35 had the fewest active users on acer aspire desktop.
- Email engagement was highest in week 34, while lowest in week 35.



Results

The obtained results provided a better insight into the behaviour of the users and helped us understand and determine different aspects relevant to the needs of the team to optimize their approach and gain subsequent satisfactory results.



Thank you!

