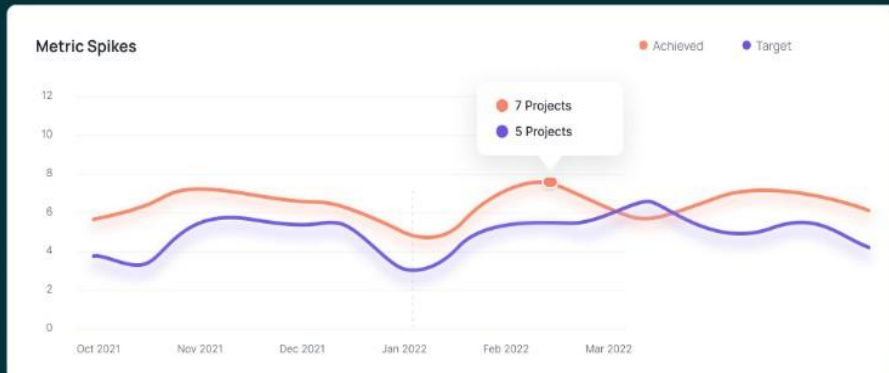


# OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

**trainity**

## Operation Analytics & Investigating metric spike case study



Case Study 1 :  
Job Data Analysis

Case Study 2 :  
Investigating Metric  
Spike

By - Anindya Das

# PROJECT DESCRIPTION

The project's objective is to leverage operational analytics for end-to-end company operations analysis, identify improvement areas, and provide actionable insights to cross-functional teams. As the data analyst lead at Microsoft, I will collaborate closely with departments like operations, support, and marketing to derive valuable insights from their data. The primary focus is on optimizing workflows, enhancing automation, and predicting the company's overall growth or decline. Additionally, the project will involve analyzing metric spikes in daily engagement and sales, investigating their causes, and providing insights to address any dips in these metrics.

# APPROACH

This project focuses on gathering data sets from various departments and ensuring quality and normalization for import into an SQL database. We will use SQL queries to analyze the data and uncover patterns, trends, and anomalies. Key performance metrics like daily engagement, sales, and customer satisfaction will be identified with cross-functional teams and tracked using SQL. Spikes in these metrics will be investigated through SQL subqueries, analyzing historical data, and collaborating for context. The insights gained will drive process improvements, optimize workflows, and support automation efforts.

# TECH-STACK USED

Used mysql workbench 8.0 CE which is owned by Oracle

# CASE STUDY 1 : JOB DATA ANALYSIS

Operation-1 (Table-1)



ds	job_id	actor_id	event	language	time_spent	org
2020-11-30	21	1001	skip	English	15	A
2020-11-30	22	1006	transfer	Arabic	25	B
2020-11-29	23	1003	decision	Persian	20	C
2020-11-28	23	1005	transfer	Persian	22	D
2020-11-28	25	1002	decision	Hindi	11	B
2020-11-27	11	1007	decision	French	104	D
2020-11-26	23	1004	skip	Persian	56	A
2020-11-25	20	1003	transfer	Italian	45	C

TABLE NAME : job\_data

# JOBS REVIEWED OVER TIME :

```
7 • create table job_data(  
8   ds varchar(50),  
9   job_id int,  
10  actor_id int,  
11  event varchar(50),  
12  language varchar(50),  
13  time_spent int,  
14  org varchar(50)  
15 );  
16 • select * from job_data;  
17 • Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/job_data.csv"  
18 into table job_data  
19 fields terminated by ','  
20 enclosed by '"'  
21 lines terminated by '\n'  
22 ignore 1 rows;  
23  
24 • select ds as date,  
25 round((count(job_id)/sum(time_spent))*3600) as "jobs viewd per hour per day" from job_data  
26 WHERE ds BETWEEN "01-11-2020" AND "30-11-2020" group by ds;  
27
```

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

Result Grid   Filter Rows: <input type="text"/>		
	date	jobs viewd per hour per 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

Result

Jobs reviewed per hour for each day in November 2020 – Query

\*Insights : In November 2020, job review activity varied significantly, with some days showing higher engagement than others. Investigating potential factors influencing this fluctuation, such as technical issues like website outages or slow loading times is crucial. External factors, like holidays or promotions, may have also impacted user activity. Analyzing user feedback during this period can help identify challenges and improve future engagement.

# THROUGHPUT ANALYSIS :

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

```
select round((count(event)/sum(time_spent)),2) as weekly_throughput  
from job_data;
```

7-day rolling average of throughput - Query

Result Grid		Filter
	weekly_throughput	
▶	0.03	

Result

```
select ds as date, round((count(event)/sum(time_spent)),2) as daily_metric  
from job_data group by date;
```

Daily metric

Result Grid			Filter Rows:
	date	daily_metric	
▶	11/30/2020	0.05	
	11/29/2020	0.05	
	11/28/2020	0.06	
	11/27/2020	0.01	
	11/26/2020	0.02	
	11/25/2020	0.02	

Result



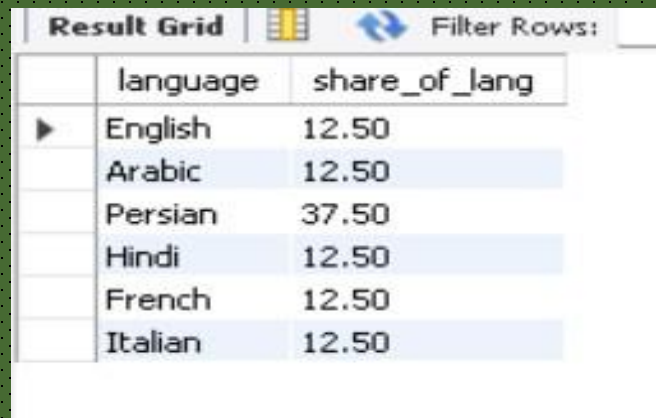
\*Insights : The 7-day rolling average of throughput offers a clearer perspective on data trends, helping to mitigate the impact of daily variations. Maintaining the use of this rolling average for throughput analysis is recommended, as it yields a more consistent portrayal of performance trends. This approach aids in recognizing long-term patterns and facilitates more informed decision-making.

# LANGUAGE SHARE ANALYSIS :

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

```
select language, round(((count(language)/8)*100),2) as share_of_lang  
from job_data group by language;
```

The percentage share of each language over the last 30 days - Query



The screenshot shows a 'Result Grid' window with a table of query results. The table has two columns: 'language' and 'share\_of\_lang'. There are seven rows of data. The first row is 'English' with a share of 12.50. The second row is 'Arabic' with a share of 12.50. The third row is 'Persian' with a share of 37.50. The fourth row is 'Hindi' with a share of 12.50. The fifth row is 'French' with a share of 12.50. The sixth row is 'Italian' with a share of 12.50. The table is displayed with alternating light blue and white rows. Above the table, there are icons for 'Result Grid', a grid icon, and a 'Filter Rows' button.

	language	share_of_lang
▶	English	12.50
	Arabic	12.50
	Persian	37.50
	Hindi	12.50
	French	12.50
	Italian	12.50

Result

\*Insights : The language distribution in the last 30 days is relatively balanced, with Persian having the highest share. To enhance user engagement, consider investing in language-specific content or features for those languages that have lower representation. This targeted approach can help cater to diverse audience needs and improve overall interaction.

# DUPLICATE ROWS DETECTION :

**Objective :** Identify duplicate rows in the data.

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

Duplicate rows - Query

Result Grid			Filter Row
	actor_id	tot_count	
▶	1003	2	

Result

\*Insights : There is one duplicate row in the data based on the actor\_id column. To maintain data integrity, it's essential to implement data validation mechanisms that prevent such duplicates in the future. This could include checks during data entry, unique constraints in the database, and regular audits to ensure the uniqueness of key identifiers. Taking these steps will help ensure cleaner data for analysis and reporting.

# CASE STUDY 2 : INVESTIGATING METRIC SPIKE

## Operation-2 (Table-1 (Users))

user_id	A unique ID per user. Can be joined to user_id in either of the other tables.
created_at	The time the user was created (first signed up)
state	The state of the user (active or pending)
activated_at	The time the user was activated, if they are active
company_id	The ID of the user's company
language	The chosen language of the user

Table name : users

## Operation-2 (Table-3(email\_events))

user_id	The ID of the user to whom the event relates. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
action	The name of the event that occurred. "sent_weekly_digest" means that the user was delivered a digest email showing relevant conversations from the previous day. "email_open" means that the user opened the email. "email_clickthrough" means that the user clicked a link in the email.

Table name : email\_events\_table

## Operation-2 (Table-2(events))

user_id	The ID of the user logging the event. Can be joined to user\_id in either of the other tables.
occurred_at	The time the event occurred.
event_type	The general event type. There are two values in this dataset: "signup_flow", which refers to anything occurring during the process of a user's authentication, and "engagement", which refers to general product usage after the user has signed up for the first time
event_name	The specific action the user took. Possible values include: create_user: User is added to Yammer's database during signup process enter_email: User begins the signup process by entering her email address enter_info: User enters her name and personal information during signup process complete_signup: User completes the entire signup/authentication process home_page: User loads the home page like_message: User likes another user's message login: User logs into Yammer search_autocomplete: User selects a search result from the autocomplete list search_run: User runs a search query and is taken to the search results page search_click_result_X: User clicks search result X on the results page, where X is a number from 1 through 10. send_message: User posts a message view_inbox: User views messages in her inbox
location:	The country from which the event was logged (collected through IP address).
device:	The type of device used to log the event.

Table name : events\_table

# WEEKLY USER ENGAGEMENT :

```
• create table events_table(  
  user_id varchar(50),  
  occurred_at varchar(50),  
  event_type varchar(50),  
  event_name varchar(50),  
  location varchar(50),  
  device varchar(50),  
  user_type int  
);  
  
• Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/events.csv"  
  into table events_table  
  fields terminated by ','  
  enclosed by '"'  
  lines terminated by '\n'  
  ignore 1 rows;  
  
• select * from events_table;  
  
• alter table events_table add column temp_occurred_at datetime;  
• update events_table set temp_occurred_at = str_to_date (occurred_at, '%d-%m-%Y %H:%i');  
• alter table events_table drop column occurred_at;  
• alter table events_table change column temp_occurred_at occurred_at datetime;
```

events\_table formation

```
select extract(week from occurred_at) as weeks,  
count(distinct user_id) as no_of_users from events_table  
where event_type="engagement"  
group by weeks order by weeks;
```

Weekly user engagement - Query

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

Result Grid			Filter Rows:
	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	

Result

\*Insights : User engagement peaked around week 30, displaying some fluctuations over the observed period. It's important to investigate patterns associated with content updates, marketing campaigns, or any external events that might have influenced user behavior. These insights can help in planning future engagement strategies effectively.



# USER GROWTH ANALYSIS :

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

```
SELECT
    DATE(created_at) AS date,
    COUNT(DISTINCT user_id) AS new_users,
    LAG(COUNT(DISTINCT user_id), 1) OVER (ORDER BY DATE(created_at)) AS previous_users,
    (COUNT(DISTINCT user_id) - LAG(COUNT(DISTINCT user_id), 1) OVER (ORDER BY DATE(created_at))) /
        LAG(COUNT(DISTINCT user_id), 1) OVER (ORDER BY DATE(created_at)) * 100 AS user_growth_percentage
FROM users
GROUP BY DATE(created_at)
ORDER BY DATE(created_at);
```

Growth of users over time - Query

date	new_users	previous_users	user_growth_percentage
2013-01-01	7	NULL	NULL
2013-01-02	7	7	0.0000
2013-01-03	6	7	-14.2857
2013-01-04	1	6	-83.3333
2013-01-05	2	1	100.0000
2013-01-06	3	2	50.0000
2013-01-07	4	3	33.3333
2013-01-08	2	4	-50.0000
2013-01-09	6	2	200.0000
2013-01-10	6	6	0.0000
2013-01-11	6	6	0.0000
2013-01-12	3	6	-50.0000
2013-01-13	2	3	-33.3333
2013-01-14	8	2	300.0000
2013-01-15	11	8	37.5000
2013-01-16	7	11	-36.3636
2013-01-17	9	7	28.5714
2013-01-18	10	9	11.1111
2013-01-19	1	10	-90.0000
2013-01-20	1	1	0.0000
2013-01-21	7	1	600.0000
2013-01-22	5	7	-28.5714
2013-01-23	7	5	40.0000
2013-01-24	5	7	-28.5714

Result - few outputs

\*Insights : User growth has generally shown a positive trend over time, although there have been some fluctuations along the way. To fully understand this growth, it's essential to analyze the factors driving these changes. Look into periods of increased growth and see if they coincide with specific product updates, targeted marketing efforts, or emerging market trends. By identifying these correlations, it will be possible to replicate successful strategies that can help sustain and accelerate user growth in the future.

# WEEKLY RETENTION ANALYSIS :

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

```
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;
```

Weekly retention of users based on their sign-up cohort - Query

Result Grid		Filter Rows:
	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

Result

\*Insights : Weekly user retention shows a gradual decline over time. It's essential to focus on improving user retention strategies. Identify key touchpoints in the user journey where users might be dropping off and work on enhancing user experience, engagement, and value during those stages. Implementing targeted interventions at these critical points can help in maintaining a loyal user base and fostering longer-term engagement.

# WEEKLY ENGAGEMENT PER DEVICE :

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

```
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;
```

Weekly engagement per device - Query

Result Grid		Filter Rows:	
	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
	samsung galaxy tablet	17	8
	samsung galaxy note	17	7
	samsung galaxy s4	17	52

Result - few outputs

\*Insights : Engagement varies across different devices and weeks, with some devices consistently demonstrating higher engagement levels. It's advisable to optimize the user experience for those devices that show lower engagement. Additionally, keep an eye on device trends over time to adapt your strategies and focus on enhancing user engagement on devices that present the highest potential.

# EMAIL ENGAGEMENT ANALYSIS :

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

```
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 metrics - Query

Result Grid			Filter Rows:
	action_count	action	
▶	57267	sent_weekly_digest	
	20459	email_open	
	9010	email_clickthrough	
	3653	sent_reengagement_email	

Result - action\_count

Result Grid			Filter Rows:
	open_rate	click_rate	
▶	33.5834	14.7899	

Result



\*Insights : The email engagement metrics indicate that the open rate is around 33.58% and the click rate is approximately 14.79%. To assess the effectiveness of these metrics, it's essential to compare them against industry benchmarks. If your engagement rates fall below average, think about enhancing various aspects of your emails, such as the content, subject lines, and targeting strategies, to boost engagement. Consistently testing and optimizing your email campaigns can lead to improved performance and better engagement rates over time.



# RESULT

Working on this project has deepened my understanding of operational analytics and its methodologies. I've gained skills in merging and normalizing diverse datasets for accurate analysis and sharpened my SQL techniques to uncover insights. Collaborating with cross-functional teams has also improved my communication and teamwork abilities, allowing me to translate data findings into actionable recommendations for process optimization and contributing to the project's success.

**THANK YOU !!**