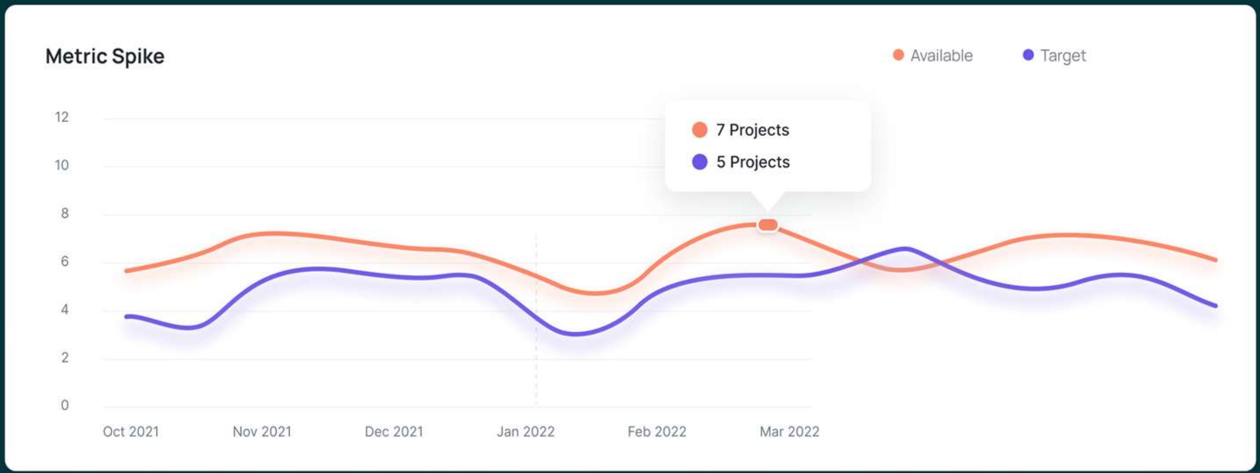
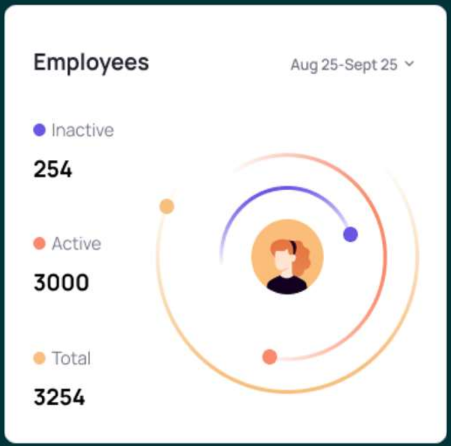
Operation Analytics & Investagating metric spike case study





--> By Kartik Roy

PROJECT DESCRIPTION

- As a Data Analyst, I focus on analyzing extensive datasets to extract insights that optimize operations. I collaborate with teams across marketing, operations, and support to address queries and resolve data-related challenges. A key aspect of my work involves diagnosing sudden shifts in critical metrics, such as decreases in sales or user engagement.
- Leveraging advanced SQL techniques, I uncover patterns and address data anomalies with precision. These insights support actionable strategies that drive informed decision-making and enhance overall company performance.

APPROACH

- Understanding the data , identify key metrices, using of Adv. SQL queries for analysis, generate actionable insights and communicate findings effectively.
- Using MS Excel's dynamic feature for visualizing the generated actionable insights for the given case studys.

TECH-STACK USED

- SQL -> For querying and analyzing the dataset.
- MySQL Workbench -> Database Management tool used for executing SQL queries and visualizing results.
- MS Excel
- MS PPT

```sql
create database project3;
use project3;

CREATE TABLE job_data(
ds date, job_id INT(10) ,
actor_id INT(10) PRIMARY KEY,
event VARCHAR(100), language VARCHAR(100),
time_spent INT(10), org VARCHAR(100));

select * from job_data;
show variables like 'SECURE_FILE_PRIV';
LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/job_data.csv'
INTO TABLE job_data
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

INSIGHTS

Case Study 1 -> **Job Data Analysis**

 # A. Jobs Reviewed Over Time:

select ds, count(job_id)

as jobs_per_day, sum(time_spent)/3600 as time_spent_dayfrom job_data

group by ds

order by ds;

ds	jobs_per_day	time_spent_day
2020-11-26	1	0.0156
2020-11-27	1	0.0289
2020-11-28	2	0.0092
2020-11-29	1	0.0056
2020-11-30	2	0.0111

INSIGHTS

Case Study 1 -> **Job Data Analysis**

 # B. Throughput Analysis:

select ds, jobs_reviewed,

avg(jobs_reviewed) over (order by ds rows between 6 preceding and current row) as throughput_7 from (

select ds, count(distinct job_id) as jobs_reviewed from job_data where ds between '2020-11-01' and '2020-11-30'group by ds) as a;

ds	jobs_reviewed	throughput_7
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.3333
2020-11-29	1	1.2500
2020-11-30	2	1.4000

INSIGHTS

\# Case Study 1 -> **Job Data Analysis**

 # C. Language Share Analysis:

```
select language, p.total_per_lan,
        round(p.total_per_lan/t.total_lan *100,2) as
lan_percent_share from(
select language, count(language) as total_per_lan
from job_data
group by language)as p,
(select count(language) as total_lan
from job_data)as t;
```

language	total_per_lan	lan_percent_share
English	1	14.29
Hindi	1	14.29
Persian	3	42.86
Arabic	1	14.29
French	1	14.29

INSIGHTS

Case Study 1 -> **Job Data Analysis**

 # D. Duplicate Rows Detection:

```
select * from
(select *,row_number() over (partition by job_id) as rownum from job_data)
as a
where rownum>1;
```

ds	job_id	actor_id	event	language	time_spent	org	rownum
2020-11-26	23	1004	skip	Persian	56	A	2
2020-11-28	23	1005	transfer	Persian	22	D	3

INSIGHTS

\# Case Study 2 -> **Investigating Metric Spike**

#A. Weekly User Engagement:

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

weeks	no_of_users
17	496
18	449
19	256
20	193
21	142
22	139
23	137

Result 1

INSIGHTS

Case Study 2 -> **Investigating Metric Spike**

#B. User Growth Analysis:

```
SELECT year, num_week,
num_active_users,SUM(num_active_users) OVER
(ORDER BY year, num_week ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW)AS
cumm_active_users FROM(

SELECT    EXTRACT(YEAR FROM a.activated_at) AS year,
EXTRACT(WEEK FROM a.activated_at) AS num_week,
COUNT(DISTINCT user_id) AS num_active_users

FROM users a

GROUP BY year, num_week

ORDER BY year, num_week) AS B;
```

year	num_week	num_active_users	cumm_active_users
NULL	NULL	9381	9381

INSIGHTS

Case Study 2 -> **Investigating Metric Spike**

#C. Weekly Retention Analysis:

```
SELECT user_id, COUNT(user_id) AS no_of_users, SUM(CASE WHEN
retention_week = 1       THEN 1       ELSE 0       END) AS per_week_retention

FROM (

SELECT a.user_id,          a.signup_week,          b.engagement_week,
DATEDIFF(b.engagement_week, a.signup_week) AS retention_week

FROM (       SELECT DISTINCT user_id,             EXTRACT(WEEK FROM
occurred_at) AS signup_week

FROM events

WHERE event_type = 'signup_flow'         AND event_name =
'complete_signup'         AND EXTRACT(WEEK FROM occurred_at) = 18    ) a
LEFT JOIN (       SELECT DISTINCT user_id,             EXTRACT(WEEK FROM
occurred_at) AS engagement_week

 FROM events

WHERE event_type = 'engagement'    ) b    ON a.user_id = b.user_id)

dGROUP BY user_id

ORDER BY user_id;
```

user_id	no_of_users	per_week_retention
11919	1	0
11920	1	0
11924	1	0
11926	1	0
11928	1	0
11929	1	0
11931	1	0
11933	1	0
11936	1	0

Result 5

INSIGHTS

Case Study 2 -> **Investigating Metric Spike**

#D. Weekly Engagement Per Device:

```
SELECT year(occurred_at) AS year,
week(occurred_at) AS no_of_weeks,
 device,
 COUNT(DISTINCT user_id) AS no_of_user
FROM events
WHERE event_type = 'engagement'
GROUP BY 1, 2, 3
ORDER BY 1, 2, 3;
```

year	no_of_weeks	device	no_of_user
2014	17	acer aspire desktop	6
2014	17	acer aspire notebook	10
2014	17	amazon fire phone	3
2014	17	asus chromebook	15
2014	17	dell inspiron desktop	12
2014	17	dell inspiron notebook	34
2014	17	hp pavilion desktop	5
2014	17	htc one	8
2014	17	ipad air	16

Result 4

INSIGHTS

Case Study 2 -> **Investigating Metric Spike**

#E. Email Engagement Analysis:

```
Select
week,num_users,time_weekly_digest_sent,time_weekly_digest_sent-lag(time_weekly_digest_sent) over(order by week)
as
time_weekly_digest_sent_growth,time_email_open,time_email_open-lag(time_email_open) over(order by week) as
time_email_open_growth,time_email_clickthrough,time_email_clickthrough-lag(time_email_clickthrough) over

(order by week) as time_email_clickthrough_growth From(

select week(occurred_at)as week,count(distinct user_id) as
num_users, sum(if(action='sent_weekly_digest',1,0)) as
time_weekly_digest_sent,sum(if(action='email_open',1,0))
as time_email_open,sum(if(action='email_clickthrough',1,0))
as time_email_clickthrough

from email_events

 group by 1

order by 1) a;
```

week	num_users	time_weekly_digest_sent	time_weekly_digest_sent_growth	time_email_open	time_email_open_growth	time_email_clickthrough	time_email_clickthrough_growth
17	950	908	NULL	0	NULL	0	NULL
18	1793	1694	786	0	0	0	0
19	167	63	-1631	0	0	0	0
20	186	68	5	0	0	0	0
21	174	89	21	0	0	0	0
22	185	89	0	0	0	0	0
23	196	92	3	0	0	0	0
24	219	102	10	0	0	0	0
25	217	102	0	0	0	0	0

Result 3

RESULT

Operational analytics is pivotal in driving company performance by thoroughly examining end-to-end operations and pinpointing areas for improvement. As a Data Analyst, applying advanced SQL techniques to investigate metric fluctuations is vital for uncovering reasons behind sudden shifts in key metrics like user engagement and sales. By delivering actionable insights from data analysis, departments such as operations, support, and marketing can make well-informed decisions that streamline operations and foster business growth. This structured method not only resolves daily issues but also contributes to strategic planning and ongoing organizational development.