OPERATION ANALYTICS AND INVESTIGATION METRIC SPIKE

PROJECT DESCRIPTION

- Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. This kind of analysis is further used to predict the overall growth or decline of a company's fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.
- Investigating metric spike is also an important part of operation analytics as being a Data Analyst we must be able to understand or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily and for that it's very important to investigate metric spike.
- In this project, we are provided with different data sets, tables from which we must derive certain insights out of it and answer the questions asked by different departments.

APPROCH

Initially I tried to understand the data and study it thoroughly to get the understanding of data by running some basic SQL queries. After I get comfortable with data, I started to look for answers to questions

TECH-STACK USED

I used MY SQL to complete this project because I am comfortable using it.

Case Study 1 (Job Data):

a) **Number of jobs reviewed -** Calculate the number of jobs reviewed per hour per day for November 2020?

QUERY:-

```
 select count(distinct job_id)/(30*24) as num_jobs_reviewed
from job_data
Where
 ds between '2020-11-01' and '2020-11-30';
```

Num_jobs_reviewed
0.0083

b) Throughput: - Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

QUERY:-

```
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
 GROUP BY ds
ORDER BY ds) sub;
```

ds	jobs_reviewed	throughput_7
2020-11-25	1	1.0000
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.2500
2020-11-29	1	1.2000
2020-11-30	2	1.3333

c) **Percentage share of each language -** Calculate the percentage share of each language in the last 30 days?

Query:-

```
select language, count(language) as total_language,
count(*)*100.0/sum(count(*)) over() as percentage
from job_data
where ds >='2020-11-01' and ds <='2020-11-30'
 group by language
 order by language;
```

language	total_language	percentage
Arabic	1	12.50000
English	1	12.50000
French	1	12.50000
Hindi	1	12.50000
Italian	1	12.50000
Persian	3	37.50000

d) Duplicate rows - Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

QUERY:-

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

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

Case Study 2 (Investigating Metric Spike):

a) **User Engagement:** Calculate the weekly user engagement?

QUERY:-

```
SELECT EXTRACT(WEEK FROM occurred_at) AS week_num ,
 COUNT(DISTINCT user_id) AS num_users
FROM events
where event_type = 'engagement'
GROUP BY weeknum
ORDER BY week_num;
```

week-num	num_users
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
…..	……

***This table has 19 rows**

b) User Growth: Calculate the user growth for product?

QUERY:-

```
select year, week_num, num_users,
sum(num_users)over(order by year, week_num) as
cum_usersfrom
(select EXTRACT(YEAR from created_at) as year,
EXTRACT(week from created_at) as week_num,
count(DISTINCT user_id) as num_users
from users
 where state = 'active'
group by year, week_num
order by year, week_num)sub;
```

year	week-num	num_users	Cum_users
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435

***This table has 89 rows**

c) Weekly Retention: Calculate the weekly retention of users-sign up cohort?

QUERY:-

```
SELECT COUNT(user_id)as users,
SUM(CASE WHEN retention_week = 1 THEN 1 ELSE 0 END ) AS  week_1,
SUM(CASE WHEN retention_week = 2 THEN 1 ELSE 0 END ) AS  week_2,
SUM(CASE WHEN retention_week = 3 THEN 1 ELSE 0 END ) AS  week_3,
SUM(CASE WHEN retention_week = 4 THEN 1 ELSE 0 END ) AS  week_4,
SUM(CASE WHEN retention_week = 5 THEN 1 ELSE 0 END ) AS  week_5
FROM (SELECT a.user_id,
a.sign_up_week,
b.engagement_week,
b.engagement_week - a.sign_up_week as retention_week
FROM (
(select distinct user_id, extract(week from occurred_at) as sign_up_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
)order by a.user_id )a;
```

users	week_1	week_2	week_3	week_4	week_5
182	11	4	3	0	1

d) Weekly Engagement: Calculate the weekly engagement per device?

QUERY:-

```
SELECT EXTRACT(month from occurred_at) as month_num,
EXTRACT(week from occurred_at) as week_num,device,
count(user_id)
from events
group by month_num, week_num, device
order by month_num, week_num, device;
```

month_num	week_num	device	count(user_id)
5	17	acer aspire desktop	11
5	17	acer aspire notebook	22
5	17	amazon fire phone	12
5	17	asus chromebook	14
5	17	dell inspiron desktop	18
5	17	dell inspiron notebook	23
5	17	hp pavilion desktop	19
5	17	htc one	6
5	17	ipad air	10

***This table has 493 rows**

e) Email Engagement: Calculate the email engagement metrics?

QUERY:-

```
SELECT EXTRACT(week FROM occurred_at) AS week,
 COUNT(CASE WHEN e.action = 'sent_weekly_digest' THEN e.user_id ELSE NULL END) AS weekly_emails,
COUNT(CASE WHEN e.action = 'sent_reengagement_email' THEN e.user_id ELSE NULL END) AS reengagement_emails,
COUNT(CASE WHEN e.action = 'email_open' THEN e.user_id ELSE NULL END) AS email_opens,
 COUNT(CASE WHEN e.action = 'email_clickthrough' THEN e.user_id  ELSE NULL END) AS email_clickthroughs
FROM email_events e
GROUP BY 1;
```

Weekly_emails	Reengagment_emails	Email_opens	Email_clickthroughs
57267	3653	20459	9010

Insights:

- The number of distinct jobs reviewed per hour per day for November 2020 is 83%.
- We used the 7-day rolling average of throughput as it gives the average for all the days right from day 1 to day 7 whereas, daily metric gives the average for only that particular day itself.
- The percentage share of Persian language is the most (37.5%).
- There are two duplicate rows if we partition the data by job_id. But if we look the overall columns, all the rows are unique.
- The overall count of weekly engagement per device used is the most for MacBook users and iPhone users.

RESULT

- In this project, I learned how to apply advanced SQL concepts like Windows Functions, etc. I understood how the real-world industry works. It helped me in mastering my SQL concepts. I learned how to ask the right questions given the circumstances. From the given data and questions, which columns to consider and how to find the valuable insights which help the business to grow. I learned how the company find different areas related to the company to improve it further. I got to know about investigating metric spike