Operation Analytics and Investigating Metric Spike - (Project 3)

Project Description –

A. Operation analytics –

This is refers to the use of data analysis and advanced analytical techniques to gain insights and optimize various operational aspects of a business or organization. It involves collecting, processing, and analyzing data from various operational processes to improve efficiency, reduce costs, enhance productivity, and make better-informed decisions.

B. Metric spike –

This is refers to a sudden and significant increase or surge in a particular metric or key performance indicator (KPI) within a specific timeframe. Metric spikes can occur for various reasons and are typically monitored closely by organizations to understand the underlying causes and take appropriate actions

C. Approach –

1. Download all Data for this project
2. Create database with the dataset given for Case study 1 & 2
3. Get details of database insights
4. Write Queries to find answers
5. Analyze the information and use the data for decision making

D.Tech-Stack Used

1.Mysql Workbench 8.0

E. Resources used -

1.Mysql workbench 8.0 to run SQL queries
2.Datasets provided Case Study 1 (Job Data), Case Study 2 (Job Data)

F. Insights –

Case Study 1 (Job Data):

A. Number of jobs reviewed: Amount of jobs reviewed over time.

- Calculate the number of jobs reviewed per hour per day for November 2020?

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

Result - The number of distinct jobs reviewed per hour per day for November 2020 is 83%

B. Throughput: It is the no. of events happening per second.
Calculate 7 day rolling average of throughput?

```
select ds, jobs_reviewed,
avg(jobs_reviewed)over(order by ds rows between 6 preceding and current row)
as throughput_7_rolling_avg
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
order by ds
)a;
```

Result – 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.

C. Percentage share of each language: Share of each language for different contents.

- Calculate the percentage share of each language in the last 30 days?

```
select language, num_jobs,
100.0* num_jobs/total_jobs as pct_share_jobs
```

```
Result from
(
select language, count(distinct job_id) as num_jobs
from job_data
group by language
)a
cross join
(
select count(distinct job_id) as total_jobs
from job_data
)b;
```

Result - The percentage share of Persian language is the most (37.5%).

D. Duplicate rows: Rows that have the same value present in them.

- How will you display duplicates from the table?

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

Result - There are two duplicate rows if we partition the data by job_id. if we look the overall columns, all the rows are unique.

Case Study 2 (Investigating metric spike):

A. User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.

- Calculate the weekly user engagement?

```
select
extract(week from occurred_at) as num_week,
count(distinct user_id) as no_of_distinct_user
from tutorial.yammer_events
group by num_week;
```

Result - The weekly user engagement increased from week 18th to week 31st and then started declining from then onwards. This means that some of the users do not find much quality in the product/service in the last of the weeks.

B. User Growth: Amount of users growing over time for a product.

- Calculate the user growth for product?

```
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 tutorial.yammer_users a
where state='active'
group by year, num_week
order by year, num_week
)a;
```

Result - There are in total 9381 active users from 1st week of 2013 to the 35th week of 2014.

C .Weekly Retention: Users getting retained weekly after signing-up for a product.

- Calculate the weekly retention of users-sign up cohort?

```
select count(user_id),
 sum(case when retention_week = 1 then 1 else 0 end) as
per_week_retention
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 occured_at) as sign_up_week
from tutorial.yammer_events
where event_type = 'signup_flow'
and event_name = 'complete_signup'
and extract(week from occured_at)=18)a
left join
(select distinct user_id, extract(week from occured_at) as engagement_week
from tutorial.yammer_events
where event_type = 'engagement')b
on a.user_id = b.user_id
)
group by user_id
order by user_id;
```

D. Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

- Calculate the weekly engagement per device?

```
select
extract(year from occured_at) as year_num,
extract(week from occured_at) as week_num,
device,
count(distinct user_id) as no_of_users
from tutorial.yammer_events
where event_type = 'engagement'
group by 1,2,3
order by 1,2,3;
```

Result - The overall count of weekly engagement per device used is the most for MacBook users and iPhone users.

E. Email Engagement: Users engaging with the email service.

- Calculate the email engagement metrics?

```
select
100.0 * sum(case when email_cat = 'email_opened' then 1 else 0 end)
 /sum(case when email_cat = 'email_sent' then 1 else 0 end)
as email_opening_rate,
100.0 * sum(case when email_cat = 'email_clicked' then 1 else 0 end)
 /sum(case when email_cat = 'email_sent' then 1 else 0 end)
as email_clicking_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_clicked'
end as email_cat
from tutorial.yammer_events
)a;
```

Result – The email opening rate is around 34% and email clicking rate is around 15%. The users are engaging with the email service which is good for the company to expand

Final Result & Outcomes –

This project is helpful to understand operational analytics with SQL queries workflow on different aspects. I learn how we can used data for different task and how to ask questions with Mysql workbench. This will be helpful for me to get the insights to get the insights of the company with proper data management.