Operation Analytics and Investigation Metrics

Case Study 1: Operation Analytics

Project Description

- This analysis is to understand the complete end to end operations of a company.
- By this analyzing process company can find out the areas on which it must improve.

Approach

- First I spent some time in understanding the given data and problem statements.
- Then, I studied the attributes and the relations between tables.
- Moving forward, using various SQL commands and functions I solved the given questions and derived insights from the datasets provided.

Execution

1. **Number of jobs reviewed:** Amount of jobs reviewed over time.
   Your task: 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';
```

2. **Throughput:** It is the no. of events happening per second.
   Your task: Let's say the above metric is called 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_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;
```

3. **Percentage share of each language:** Share of each language for different contents.
   My task: Calculate the percentage share of each language in the last 30 days?

   Query:

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

4. **Duplicate rows:** Rows that have the same value present in them.
   My task: 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;
```

Case Study 2: Investigating Metric Spike

1. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
   My task: Calculate the weekly user engagement?

   Query:

```
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;
```

2. **User Growth:** Amount of users growing over time for a product.
   My task: Calculate the user growth for product?

   Query:

```
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;
```

3. **Weekly Retention:** Users getting retained weekly after signing-up for a product.
   My task: Calculate the weekly retention of users-sign up cohort?

   Query:

```
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;
```

4. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
   My task: Calculate the weekly engagement per device?

   Query:

```
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;
```

Tech-Stack Used:

- I used MySQL 8.0 to create and perform analysis answering the given questions.
- I used MS-Word to make report.

Insights:

Case Study 1:

For November 2020, 83% of all unique jobs were examined each hour. We chose the 7-day moving average of throughput because it provides averages for days 1 through 7 while daily metrics only provide averages for a single day. Persian has the highest percentage of speakers (37.5%). If we divide the data by job_id, we get two duplicate rows. But if we examine the overall columns, each row is distinct.

Case Study 2:

The weekly user engagement increased from week 18th to week 31st and then started decreasing which means that some of the users do not find much quality in the product/service in the last weeks. There are in total 9381 active users from 1st week of 2013 to the 35th week of 2014. MacBook and iPhone are the most used device according to the weekly count. 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.

Result

This project helped to understand the importance of operation analytics and apply advanced SQL concepts. I came to know how insights are used by industries in improvisation. It helped me to understand the aggregation of data from multiple sources into a cumulative and organized solution of delivering analytical models.