Operation and Metric Analysis

-Saksham Gupta

Description:

The project is based on Operation Analysis i.e., to perform end to end operations for company growth and which areas to improve on. I am going to find the following tasks: (1) The number of jobs reviewed per hour per day for November 2020; (2) 7 day rolling average of throughput; (3) The Percentage share of each language in the last 30 days; (4) Displaying Duplicates; (5) The weekly user engagement; (6)The user growth for the product; (7) The weekly retention of users-sign up cohort; (8) To measure the activeness of a user; (9) The email engagement metrics; from the provided database.

Approach:

- - I obtain the necessary information from the description, i.e., the tasks I must carry out.
- - To get the result, I used MYSQL Workbench to create new files and begin writing my queries.
- - I later ran the queries, and if any mistakes were found, I changed and corrected the code.
- - After execution was complete, I looked through the code.
- - I finished by adding my code to the file.

Tech Stack used:

❖ SQL ❖ Development tool – MYSQL Workbench version 8.0.30. The main purpose of using MySQL workbench is that it provides the console to simply editable and administer the MYSQL environments and to gain better results and insights of the data. It provides data modelling, SQL development and connecting servers and is the best tool to design, generate and manage the databases.

Code:

```
#the number of jobs reviewed per hour per day for November 2020
select
count(job_id)/(30*24) as num_jobs_reviewed
from job_data
where
ds between "2020-11-01" and "2020-11-30";
#7 day rolling average of throughput
#throughtput - calcualting rolling average
select ds,
 jobs_reviewed,
 avg(jobs_reviewed)over(order by ds rows between 6 preceding and
current row) as rolling_average
 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;

#the percentage share of each language in the last 30 days.
select language,
num_jobs,
100*(num_jobs/total_jobs) as pct_share
from
(select
ds,
language,
count(job_id) as num_jobs
from job_data
group by language)a
```

```
cross join(select count(job_id) as total_jobs from job_data)b;

# displaying duplicate rows
select * from(select *,row_number()over(partition by job_id) as rownum
from job_data)a where rownum>1;

#the weekly user engagement
select extract(week from occurred_at) as weeknum,count(distinct
user_id) from events
group by weeknum;

#the user growth for product
select year,weeknum,num_active_user,sum(num_active_user)
over(order by year,weeknum rows between unbounded preceding and
current row) as cum_active_users
from(select extract(year from activated_at) as year,extract(week from
activated_at)as weeknum,count(distinct user_id) as num_active_user
from opusers a where state="active" group by year,weeknum order by
year,weeknum)a;

#the weekly retention of users-sign up cohort.
select extract(year from occurred_at)as year,
extract(week from occurred_at)as week,
device,
count(distinct user_id)
from events
where event_type="engagement"
group by 1,2,3
order by 1,2,3;

#email engagement metrics
SELECT COUNT(user_id), SUM(CASE WHEN retention_week = 1 THEN 1
ELSE 0 END) as week_1
FROM ( SELECT a.user_id, a.signup_week, b.engagement_week,
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 )
ORDER BY a.user_id )a
```