Operation Analytics and Investigating Metric Spike

PROJECT DESCRIPTION

Here we are trying to build a system through which we will be able to build better business opportunities and better client interaction. Through this operation analytics we will measure certain parameters which in turn will help us grow as a company and also most certainly help in understanding our flaws. By this way we would also be able to recommend best of our services and rectify the area we are lagging on. These operation analytics are the foundation pillar of the success of any organisation.

Investigating metric spike is also an important part of operation analytics as being a Data Analyst you 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 its very important to investigate metric spike.

APPROACH

I have tried to understand the dataset before trying to execute any of the query. I related each given data with what exactly I require to derive e.g. If I have been given the job dataset and asked to find out the job reviewed per hour, then I will build the connection between how many days it took review the job etc. So over in all my approach was quite simple I just kept on connecting the dots to build these query list.

TECH-STACK

I had used MySQL Workbench 8.0 CE available in MySQL.com. It is a unified visual tool for database architects, developers and DBA’s.

The reason for using it is that it has very user friendly interface and it is also hassle free with all the provided services such as creating a database, administering it, modifying it etc. I have particulary used it to create a database named operation_analytics and then I have used multiple queries to derive the outcome I required out of the given dataset.

INSIGHTS

I have gathered a great deal of knowledge from this assignment. I got a real exposure of how exactly an organisation works to get close to its error free environment and generate the best revenue out of the services or goods it offers. The ways in which here it is mentioned about the requirement from the given dataset it give you the opportunity to map the data like I had to find the percentage share of Persian language is the most (37.5%). Certainly there were other function such as join function, max, limit etc. These functions help in deriving the sorted data out of such a huge provided data.

In short these queries help us in getting the exact same amount of data or information we require out of the whole dataset.

RESULT

A.) CASE STUDY 1 (Job Data)

1.) Amount of jobs reviewed over time

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

2.) 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
 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  )a;
```

3.) Share of each language for different contents

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

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

B.) CASE STUDY 2 (Investigating metric spike)

1.) User Engagement:

```
select extract(week from occurred_at) as num_week,

count(distinct user_id) as no_of_distinct_user

 from table_events

group by num_week;
```

2.) Amount of users growing over time for a 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 total_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 table_users a
where state='active'
group by year, num_week
order by year, num_week    )a;
```

3.) Users getting retained weekly after signing-up for a product

```
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.) Users find quality in a product/service weekly

```
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;
```

5.) Users engaging with the Email service

```
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;
```

I have successfully been able to segregate the dataset with what I required using exact query to differentiate.

It gave me a great amount of pleasure to complete the project I learned all the functions used in queries with a live implication it get etched in my memory easily, I also learnt about aggregate functions.

Submitted by :- Bishwomkar Panigrahi