

Operation Analytics and Investigating Metric Spike

- Rukmini Annadata
rukmini2k2@gmail.com

Project Description:

Operational Analytics is pivotal for scrutinizing a company's end-to-end operations, identifying areas for enhancement. As a Data Analyst, one's role involves collaborating with teams like operations, support, and marketing to extract valuable insights from collected data.

Central to Operational Analytics is deciphering metric spikes. This means grasping sudden shifts in key metrics, be it a dip in user engagement or sales. Daily, you'll tackle such questions, necessitating adeptness in investigating metric anomalies.

In this project, assume the mantle of Lead Data Analyst at a company akin to Microsoft. Armed with diverse datasets, you'll employ advanced SQL skills to analyse data. Your mission: furnish insights to queries posed by various company sectors. The goal is to enhance operations and fathom sudden metric changes, contributing to informed decision-making.

Approach:

Case Study 1: Job Data Analysis

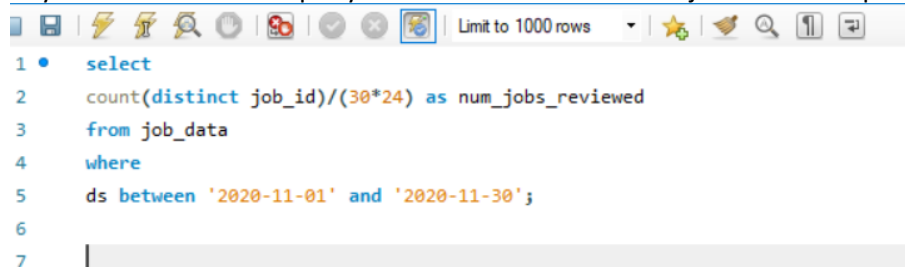
You will be working with a table named `job_data` with the following columns:

- **job_id:** Unique identifier of jobs
- **actor_id:** Unique identifier of actor
- **event:** The type of event (decision/skip/transfer).
- **language:** The Language of the content
- **time_spent:** Time spent to review the job in seconds.
- **org:** The Organization of the actor
- **ds:** The date in the format yyyy/mm/dd (stored as text).

Tasks:

- A. **Jobs Reviewed Over Time:** To display the amount of jobs reviewed over time.

My Task: Write an SQL query to calculate the number of jobs reviewed per hour for each



```
1 • select
2   count(distinct job_id)/(30*24) as num_jobs_reviewed
3   from job_data
4   where
5     ds between '2020-11-01' and '2020-11-30';
6
7 |
```

B. Throughput Analysis: To display the no. of events happening per second.

My Task: To write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

```
• 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;
```

C. Language Share Analysis: Share of each language for different contents.

My Task: To write an SQL query to calculate the percentage share of each language over the last 30 days.

```
19 • select language, num_jobs,
20      100.0* num_jobs/total_jobs as pct_share_jobs
21      from
22      (
23        select language, count(distinct job_id) as num_jobs
24        from job_data
25        group by language
26      )a
27      cross join
28      (
29        select count(distinct job_id) as total_jobs
30        from job_data
31      )b;
```

D. Duplicate Rows Detection:

- Objective: To Identify duplicate rows in the data.

My Task: To write an SQL query to display duplicate rows from the job_data table.

```
33 • select * from
34      (
35        select *,
36        row_number()over(partition by job_id) as rownum
37        from job_data
38      )a
39      where rownum>1;
```

Case Study 2: Investigating Metric Spike

You will be working with three tables:

users: Contains one row per user, with descriptive information about that user's account.

events: Contains one row per event, where an event is an action that a user has taken (e.g., login, messaging, search).

email_events: Contains events specific to the sending of emails.

Tasks:

A. Weekly User Engagement:

- Objective: To measure the activeness of a user. Measuring if the user finds quality in a product/service.
- My Task: Write an SQL query to 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;
```

B. User Growth Analysis:

- Objective: Analyze the growth of users over time for a product.(amount of users)
- My Task: Write an SQL query to calculate the user growth for the product.

```
48 • select year, num_week, num_active_users,
49      sum(num_active_users) over(order by year, num_week rows between unbounded
50      preceding and current row)
51      as cumm_active_users
52      from
53      (select
54          extract(year from a.activated_at) as year,
55          extract(week from a.activated_at) as num_week,
56          count(distinct user_id) as num_active_users
57      from tutorial.yammer_users a
58      where state='active'
59      group by year, num_week
60      order by year, num_week
61      )a;
```

C. Weekly Retention Analysis:

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- My Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

```
63 • select count(user_id),
64      sum(case when retention_week = 1 then 1 else 0 end) as
65      per_week_retention
66      from
67      (
68          select a.user_id,
69              a.sign_up_week,
70              b.engagement_week,
71              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 tutorial.yammer_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 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 Per Device:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

- Objective: Measure the activeness of users on a weekly basis per device.
- My Task: To write an SQL query to calculate the weekly engagement per device.

```

extract(year from occurred_at) as year_num,
extract(week from occurred_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;

```

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

- Objective: Analyze how users are engaging with the email service.
- Your Task: Write an SQL query to 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;

```

Tech-Stack Used:

- MySQL Workbench (Version 8.0 CE): Employed MySQL Workbench for tasks encompassing data modeling, SQL development, and administrative configurations. The platform provides an intuitive graphical interface to interact with databases, enabling efficient creation of databases and analysis to address posed questions.
- Mode.com: Leveraged Mode.com for swift execution of advanced analytics, yielding valuable insights without necessitating any downloads or installations. Seamlessly connected to data warehouses via Mode, and employed it to undertake Case Study 2 involving the investigation of metric spikes.
- Microsoft Word 2021: Utilized Microsoft Word 2021 to generate a comprehensive report in PDF format, destined for presentation to the leadership team. This facilitated effective communication of findings and recommendations derived from the analysis.

Insights:

Case Study 1 - Job Data Analysis:

- During November 2020, an average of 83% distinct jobs were reviewed per hour per day.
- To capture an inclusive view, we employed a 7-day rolling average for throughput analysis. This approach presents an averaged perspective encompassing all seven days, unlike daily metrics that pertain to individual days.
- Among language shares, Persian takes the lead, constituting 37.5% of the total.
- A noteworthy observation is the presence of two duplicate rows when partitioned by `job_id`. However, considering all columns holistically, each row remains distinct.

Case Study 2 - Investigation of Metric Spike:

- User engagement demonstrated an upward trend from the 18th to the 31st week, followed by a subsequent decline. This trend indicates a decline in perceived value by some users during the latter weeks.
- The analysis identified a cumulative count of 9381 active users spanning from the first week of 2013 to the 35th week of 2014.
- Notably, users employing MacBook and iPhone devices exhibited the highest weekly engagement counts, marking these as prime engagement tools.

Result:

During this project, I gained a deep understanding of using advanced techniques in SQL, like Windows Functions, and more. This experience gave me really valuable knowledge about how things actually work in real industries. It made me much better at using SQL effectively. I learned how to create specific questions that fit the situation, figure out which parts of the data are most important, and discover crucial information that helps a business to grow. This project helped me get better at finding out how different parts of a company can be improved. It also taught me about investigating sudden ups and downs in data, which is really important for understanding how a business is doing.