Operation Analytics and Investigating Metric Spike

Project Description: Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. You work closely with the ops team, support team, marketing team, etc and help them derive insights out of the data they collect.

Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company's fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.

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.

You are working for a company like Microsoft designated as Data Analyst Lead and is provided with different data sets, tables from which you must derive certain insights out of it and answer the questions asked by different departments.

You are required to provide a detailed report for the below two operations mentioning the answers for the related questions:

Case Study 1 (Job Data)
Below is the structure of the table with the definition of each column that you must work on:

- **Table-1:** job_data
    - **job_id:** unique identifier of jobs
    - **actor_id:** unique identifier of actor
    - **event:** decision/skip/transfer
    - **language:** language of the content
    - **time_spent:** time spent to review the job in seconds
    - **org:** organization of the actor
    - **ds:** date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

```
create database operation_analytics;

use operation_analytics;

create table job_data(

ds date,

job_id int,

actor_id int,

event varchar(255),

language varchar(255),

time_spent int,
```

```
org varchar(255)
);
select * from job_data;
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org)
VALUES ('2020-11-30', 21, 1001, 'skip', 'English', 15, 'A'),
    ('2020-11-30', 22, 1006, 'transfer', 'Arabic', 25, 'B'),
    ('2020-11-29', 23, 1003, 'decision', 'Persian', 20, 'C'),
    ('2020-11-28', 23, 1005,'transfer', 'Persian', 22, 'D'),
    ('2020-11-28', 25, 1002, 'decision', 'Hindi', 11, 'B'),
    ('2020-11-27', 11, 1007, 'decision', 'French', 104, 'D'),
    ('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),
    ('2020-11-25', 20, 1004, 'transfer', 'Italian', 45, 'C');
```

Use the dataset attached in the Dataset section below the project images then answer the questions that follows

A. **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?

Answer:

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

B. **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?

Answer:

```
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;
```

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

Answer:

```
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;
```

D. **Duplicate rows:** Rows that have the same value present in them.
Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

Answer:

```
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)

The structure of the table with the definition of each column that you must work on is present in the project image

- **Table-1:** users
  This table includes one row per user, with descriptive information about that user's account.

- **Table-2:** events
  This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
- **Table-3:** email_events
  This table contains events specific to the sending of emails. It is similar in structure to the events table above.

Use the dataset attached in the Dataset section below the project images then answer the questions that follows

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

Answer:

```
select extract(week from occurred_at) as weeknum,count(distinct user_id) from events
group by weeknum;
```

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

Answer:

```
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;
```

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

Answer:

```
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.
Your task: Calculate the weekly engagement per device?

Answer:

```
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;
```

E. **Email Engagement:** Users engaging with the email service.
Your task: Calculate the email engagement metrics?

Answer:

```
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
```