Operation Analytics and Investigating Metric Spike

Project description

This project on operation analytics and investigating metric spike which has different 2 different case

1. Operation Analytics: As a Data Analyst my primary focus is on conducting comprehensive analysis of the end-to-end operations of the company. By working closely with various teams such as operations, support, and marketing, you help derive actionable insights from the data they collect.
2. Metric Spike Investigation: As a Data Analyst is to investigate metric spikes and provide explanations for fluctuations in performance indicators. By addressing questions such as why there is a dip in daily engagement or why sales have declined, understanding the factors impacting the company's performance. This daily investigation of metric spikes helps the organization make informed decisions, take corrective actions, and ensure continuous growth and success.

Here task is to extract information from user database to make the decisions.

Approach

Operation analytics and investigating metric spike project with the help of database created on MySQL sever, data from multiple tables is extracted using single row function joins and subquery concepts. Looking to the questions which need to be answered.

Excel for data cleaning which involved removing the data which was not needed organising the data to SQL specific format so it is easy for data extraction

With the help of SQL we will be joining multiple tables and removing the data which is not needed to answer the question

Tech-Stack Used

Software: MySql workbench 8.0 CE **Version:** 8.0

Why?

MySql manages the data RDBMS which is basically based on SQL.
As we can create private server for each case interface of MySql is better, easy to learn and execution is easier MySql.

Insights

I learned Importance of preparing the data for analysis which is cleaning formatting and also learned advanced SQL concepts , understood industry practices and how the projects work in real time .

Results

Questions to be answered

CASE STUDY 1 - OPERATION ANALYTICS

DATABASE USED FOR ANALYSIS

```
create table job_data(
job_id int,
actor_id int,
event varchar(255),
language varchar(255),
time_spent int,
org varchar(255),
ds date);
```

```
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'),
('2020-11-25', 19, 1003, 'transfer', 'Persian', 32, 'B'),
('2020-11-24', 15, 1006, 'decision', 'Arabic', 17, 'D'),
('2020-11-23', 18, 1005, 'skip', 'Hindi', 29, 'A'),
('2020-11-22', 12, 1002, 'transfer', 'English', 41, 'C'),
('2020-11-21', 24, 1001, 'decision', 'French', 14, 'B'),
('2020-11-20', 17, 1007, 'skip', 'Spanish', 73, 'D'),
('2020-11-29', 14, 1004, 'transfer', 'Italian', 27, 'A'),
('2020-11-28', 16, 1001, 'decision', 'German', 19, 'C'),
('2020-11-27', 13, 1005, 'skip', 'Russian', 38, 'B'),
('2020-11-26', 19, 1003, 'transfer', 'Persian', 26, 'D'),
('2020-11-25', 10, 1002, 'decision', 'Chinese', 62, 'A'),
('2020-11-24', 23, 1006, 'skip', 'Arabic', 49, 'C'),
('2020-11-23', 22, 1004, 'transfer', 'Hindi', 31, 'B'),
('2020-11-22', 21, 1007, 'decision', 'French', 12, 'D'),
('2020-11-22', 20, 1001, 'skip', 'Spanish', 27, 'A'),
('2020-11-21', 25, 1005, 'transfer', 'Russian', 18, 'C'),
('2020-11-29', 24, 1002, 'decision', 'German', 21, 'B'),
('2020-11-28', 17, 1006, 'skip', 'Chinese', 44, 'D'),
('2020-11-27', 16, 1003, 'transfer', 'Arabic', 28, 'A'),
('2020-11-26', 15, 1004, 'decision', 'Hindi', 16, 'C'),
('2020-11-25', 14, 1001, 'skip', 'French', 33, 'B'),
('2020-11-24', 13, 1007, 'transfer', 'Spanish', 19, 'D'),
('2020-11-23', 12, 1005, 'decision', 'Italian', 23, 'A'),
('2020-11-22', 11, 1002, 'skip', 'German', 51, 'C');
```

1) Calculate the number of jobs reviewed per hour per day for November 2020?

```
Query   SELECT ds as date , round(count(distinct job_id)/sum(time_spent)*3600) as
                     avg_job_revived
          FROM job_data

          WHERE ds between '2020-11-01' and '2020-11-30'

          GROUP BY ds

          ORDER BY ds desc ;
```

2) 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
        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 )sub ;
```

3) Calculate the percentage share of each language in the last 30 days?

```
Query SELECT language,COUNT(job_id) AS number_jobs ,  count(job_id)*100/sum(count(*))
                   OVER() as percentage_share_of_language
        FROM  job_data
        WHERE ds between '2020-11-01' and '2020-11-30'
        GROUP BY language ;
```

4) Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

```
Query SELECT sub.ds,sub.job_id,sub.actor_id,sub.event,sub.language,sub.time_spent,sub.org,
        CASE when sub.duplicates = 1 then "NOT DUPLICATE" else "DUPLICATE" end as
        DUPLICATE_NOT_DUPLICATE
        FROM  ( SELECT *, row_number() OVER (partition by ds, job_id, actor_id, event,
                   language, time_spent, org)as duplicates
                   FROM job_data ) as sub ;
```

CASE STUDY 2 - METRIC SPIKE INVESTIGATION

1)User Engagement: Calculate the weekly user engagement?

```
Query SELECT week(occurred_at) as Week,  count(DISTINCT user_id)as
                  Weekly_User_engagement
        FROM events
        GROUP BY Week(occurred_at)
        ORDER BY Week(occurred_at);
```

2) User Growth : Calculate the user growth for product?

```
Query SELECT year, week_number , number_active_users, sum(number_active_users)
        over(order by year, week_number rows between unbounded preceding and current row) as
        current_active_users
        FROM (SELECT extract(year from u.activated_at) as year, extract(week from
                            u.activated_at)as week_number, count(distinct user_id) as
                            number_active_users
                 FROM users as u
                 WHERE state='active'
                 GROUP BY 1, 2
                 ORDER BY 1, 2 desc )sub;
```

3) Weekly Retention: Calculate the weekly retention of users-sign up cohort?

```
SET @totalgrowth := 0;
Query SELECT sub.no_of_users, sub.date,( @totalgrowth := @totalgrowth + sub.no_of_users ) as
                  user_growth
        FROM ( SELECT count(user_id) as no_of_users, date(created_at) as date
                  FROM users WHERE state = "active"
                  GROUP BY date(created_at) ) sub;
```

4) Weekly Engagement: Calculate the weekly engagement per device?

Query
```
SELECT EXTRACT(WEEK FROM `occurred_at`) AS week_num, device,
                  COUNT(DISTINCT user_id) AS no_of_users
        FROM events
        WHERE event_type = 'engagement'
        GROUP BY 1 , 2
        ORDER BY 1 , 3 ;
```

5) **Email Engagement: Calculate the email engagement metrics?**

Query
```
SELECT week(occurred_at) as Week,
        count( DISTINCT ( CASE WHEN action = "sent_weekly_digest"
        THEN user_id end )) as weekly_digest,
        count( distinct ( CASE WHEN action = "sent_reengagement_email"
        THEN user_id end )) as reengagement_mail,
        count( distinct ( CASE WHEN action = "email_open"
        THEN user_id end )) as opened_email,
        count( distinct ( CASE WHEN action = "email_clickthrough"
        THEN user_id end )) as email_clickthrough
        FROM email_events
        GROUP BY week(occurred_at)
        ORDER BY week(occurred_at);
```