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. .

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.

I am going to handle with two types of datasets in this project for case study 1 and case study 2 respectively. I will go through some sql queries before starting this project. I read the tasks given and tried to understand the problem statements.

Approach:

First, I created a database in mysql workbench and imported the excel data tables in mysql in csv format. I read the tasks again and tried to apply my sql knowledge to solve the problem statements or tasks given in each of the two case studies. Then I performed several sql commands on mysql workbench to retrieve the exact data required in the problem

Tech-stack Used:

I used MySQL Workbench 8 0 CE

Insights:

After I performed several SQL Commands, I got the insights and a knowledge of how to perform a real time SQL knowledge on this project I understood the data given of users and employees I derived several insights about weekly user engagement, user growth, weekly retention, email engagement, number of jobs, searched for duplicate rows. Understood the process of SQL and learn how to apply real time SQL knowledge in any company.

Result:

I retrieved the data from the given datasets by coding queries on mysql workbench and the queries are mentioned below as follows.

Case Study 1 (JOB_DATA)

Task-1:

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

Query:

```
USE metric_analytics:

SELECT COUNT(DISTINCT job_id)/(30*24) AS jobreview_perhr_perday

FROM job_data;
```

Result:

jobreview_perhr_perday
0.0083

Task-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:

```
USE metric_analytics;

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

GROUP BY ds ORDER BY ds ) AS job_rev
```

Result:

ds	jobs_reviewed	throughput_7
25-11-2020	1	1.0000
26-11-2020	1	1.0000
27-11-2020	1	1.0000
28-11-2020	2	1.2500
29-11-2020	1	1.2000
30-11-2020	2	1.3333

Rolling average would be much better than daily metric for better performance of any business ,as It relates with only the recent trends and help us to compare which group of days perform better and it helps to understand why is it so ,thus we can stay in trend and keep updating our stuff

Task 3:

Calculate the percentage share of each language in the last 30 days

Query:

```
USE metric_analytics;

SELECT ds,job_id,language,COUNt(language) AS
tot_count,ROUND((COUNT(language)/6),2)*100 AS perct_share

FROM job_data

GROUP BY job_id,language

ORDER BY ds;
```

Result:

ds	job_id	language	tot_count	perct_share
25-11-2020	20	Italian	1	17.00
27-11-2020	11	French	1	17.00
28-11-2020	25	Hindi	1	17.00
29-11-2020	23	Persian	3	50.00
30-11-2020	21	English	1	17.00
30-11-2020	22	Arabic	1	17.00

Result 29

Task 4:

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

Query:

```
USE metric_analytics;

SELECT ds,job_id,event,org,COUNT(job_id) AS dup_id

FROM job_data

GROUP BY job_id

HAVING COUNT(job_id)>1

ORDER BY ds;
```

Result:

ds	job_id	event	org	dup_id
29-11-2020	23	decision	C	3

Case Study 2 (Investigating metric spike)

Task 1:

Calculate the weekly user engagement

Query:

```
USE metric_analytics;
SELECT WEEK(STR_TO_DATE(occurred_at,'%d-%m-%y'))AS week_num,
COUNT(user_id) as total_users
FROM events
GROUP BY week_num;
```

Result:

I exported the data in excel sheet for better display of results.

week_num	total_users	week_num2	total_users3
17	765	26	2539
18	2174	30	2740
19	2325	28	2571
20	2193	31	2457
21	2411	27	2596
23	2421	32	2663
22	2728	33	2723
24	2912	34	2670
25	2451	35	374
29	2491		

Task 2:

Calculate the user growth for product

Query:

```
USE metric_analytics;
SELECT WEEK(STR_TO_DATE(created_at,'%d-%m-%y')) AS week_num,
COUNT(user_id),
COUNT(user_id)-LEAD(COUNT(user_id))OVER w AS user_growth
FROM users
GROUP BY week_num
WINDOW w AS(PARTITION BY WEEK(STR_TO_DATE(created_at,'%d-%m-%y')))
```

Task 3:

Calculate the weekly retention of users-sign up cohort?

Query:

```
USE metric_analytics;
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
```

Task 4:

Calculate the weekly engagement per device?

Query:

```
USE metric_analytics;
SELECT device, WEEK(STR_TO_DATE(occurred_at,'%d-%m-%y'))AS week_num,
COUNT(user_id) as total_users
FROM events
WHERE event_type='engagement'
GROUP BY device
ORDER BY total_users DESC;
```

Result:

device	week_num	total_users	device2	week_num3	total_users4
macbook pro	18	7343	hp pavilion desktop	17	1203
lenovo thinkpad	18	4862	nexus 7	18	949
iphone 5	17	4224	nexus 10	18	864
macbook air	18	3659	ipad mini	18	805
samsung galaxy s4	18	2329	nokia lumia 635	17	618
iphone 5s	19	2151	mac mini	17	591
dell inspiron notebook	17	2111	acer aspire desktop	17	551
nexus 5	17	1932	windows surface	19	532
iphone 4s	18	1883	htc one	17	469
ipad air	17	1650	kindle fire	19	465
dell inspiron desktop	17	1413	amazon fire phone	17	374
acer aspire notebook	19	1315	samsung galaxy note	17	327
asus chromebook	18	1311	samsumg galaxy tablet	17	273

Task 5:

Calculate the email engagement metrics?

Query:

```
USE metric_analytics;
SELECT
 100.0 * SUM(CASE
 WHEN email_cat = 'email_open' THEN 1
```

```
 ELSE 0
 END) / SUM(CASE
 WHEN email_cat = 'email_sent' THEN 1
 ELSE 0
 END) AS email_open_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_clicked_rate
FROM
 (SELECT
 *,
 CASE
 WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email') THEN
'email_sent'
 WHEN action IN ('email_open') THEN 'email_open'
 WHEN action IN ('email_clickthrough') THEN 'email_clicked'
 END AS email_cat
 FROM
 email_events) a
```

Result:

	email_open_rate	email_clicked_rate
▶	33.58339	14.78989