OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Advanced SQL

PROJECT

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.

Here we work closely with the operations team, support team, marketing team, etc and help them derive insights out of the data they collect.

DESCRIPTION

Number of jobs reviewed

Percentage of each language

Remove duplicate rows

Rolling average throughout

CASE STUDY 1

Number of jobs reviewed per day ,per hour in November

```
SELECT DS, COUNT(JOB_ ID) AS JOBS_ PER_ DAY, SUM(TIME_ SPENT)/3600 AS HOURS_
SPENT
FROM MS _ DATA
WHERE DS >='2020-11-01'  AND DS <='2020-11-30'
GROUP BY DS ;
```

BY TAKING THE DATE IN TABLE AND COUNT HE JOBS IN THE TABLE BY GROUPING DATES COUNT ONCE AS PER DAY AND ONCE ASPER HOURS SUM(TOTAL TIME)/3600

JOBS REVIEWED PER DAY AND HOUR

date	Jobs per hour	Jobs per hour
2020-11-30	2	0.0111
2020-11-29	1	0.0056
2020-11-28	2	0.0092
2020-11-27	1	0.0289
2020-11-26	1	0.0156
2020-11-25	1	0.0125

CALCULATE 7 DAY ROLLING AVERAGE OF THROUGHPUT? FOR THROUGHPUT, DO YOU PREFER DAILY METRIC OR 7-DAY ROLLING AND WHY?

THROUGHPUT MEASURES HOW MANY UNITS OF INFORMATION A SYSTEM CAN PROCESS IN A PERIOD OF TIME

```
WITH CTE AS ( SELECT DS, COUNT(JOB_ID) AS NUM_JOBS,
SUM(TIME_SPENT)  AS TOTAL_TIME
FROM MS_DATA
WHERE DS BETWEEN '2020-11-01' AND '2020-11-30' GROUP BY DS  )
SELECT DS, ROUND(1.0*SUM(NUM_JOBS)
OVER (ORDER BY DS ROWS BETWEEN 6 PRECEDING AND CURRENT
ROW)  / SUM(TOTAL_TIME)
OVER (ORDER BY DS ROWS BETWEEN 6 PRECEDING AND CURRENT
ROW),2) AS THROUGHPUT_7D FROM CTE
```

- IN THE ABOVE SQL QUERY I USED COMMON TABLE EXPRESSION TO CREATE AN TEMPORARY TABLE TO ROUND OVER THE 7 DAYS RESULT AND PASS IT TO MAIN OUTPUT .HERE ROUND IS AN WINDOWS FUNCTION

date	Through put 7d
2020-11-25	0.02
2020-11-26	0.02
2020-11-27	0.01
2020-11-28	0.02
2020-11-29	0.02
2020-11-30	0.03

FOR THROUGHPUT, DO YOU PREFER DAILY METRIC OR 7-DAY ROLLING AND WHY?

- It depends on the specific use case and the goals of the organization.
- A daily metric would be useful for tracking short-term changes and identifying specific issues that arise on a day-to-day basis. This can be helpful for quickly identifying and addressing problems as they occur.
- A 7-day rolling metric, on the other hand, would be useful for identifying longer-term trends and patterns. This can be helpful for identifying and addressing underlying issues that may not be immediately apparent from daily metrics. It can also smooth out fluctuations that may occur on a day-to-day basis and provide a more accurate overall picture of performance.
- Ultimately, both daily and 7-day rolling metrics can be useful and it is important to consider the specific needs of the organization when deciding which metric to use.

languag e	cat	ds	percentag e
English	1	2020-11-30	12.5000
Arabic	1	2020-11-30	12.5000
Persian	3	2020-11-29	37.5000
Hindi	1	2020-11-28	12.5000
French	1	2020-11-27	12.5000
Italian	1	2020-11-25	12.5000

select sub.* ,cat*100 /8 as perentage from (SELECT language,count(*) as cat,max(ds) FROM ms_ data group by language limit 30)sub group by language;

- **Here I used subquery and took count of individual languages ,and find max date and limit it 30 so we get last 30 days, now calculate the count into percentage in main query cat*100/total no of languages**

SHARE OF EACH LANGUAGE FOR DIFFERENT CONTENTS FOR LAST 30 DAYS.

DUPLICATE ROWS

```
SELECT ds,job_id,actor_id,event,language,time_spent,org,  COUNT(*) AS QTY

FROM ms_data

GROUP BY ds,job_id,actor_id,event,language,time_spent,org  HAVING
COUNT(*)>1;
```

There are no duplicate rows in table .if we check the individual columns then we can see duplicates .if want to show only duplicates just count he column you want to find duplicates then use where count(*)>1 then you can find duplicates of that particular column.

CASE STUDY 2

- User Engagement
- User Growth
- Weekly Retention
- Weekly Engagement
- Email Engagement

USER ENGAGEMENT WEEKLY

SELECT DATE_FORMAT(occurred at,'%W') as date, COUNT(DISTINCT user id) AS weekly active users FROM events WHERE event type = 'engagement' AND event name = 'login' GROUP BY 1 ORDER BY 1

date	Weekly active users
Friday	303
Monday	251
Saturday	81
Sunday	84
Thursday	328
Tuesday	294
Wednesday	305

USER GROWTH OVER A PERIOD OF TIME FOR PRODUCT

```
SELECT date_format( occurred_at,"%M") AS Month,
COUNT(DISTINCT user_id) AS weekly_users,
COUNT(DISTINCT CASE WHEN device IN ('macbook pro', 'acer aspire notebook','acer aspire desktop','lenovo thinkpad', 'mac mini', 'dell inspiron desktop','dell inspiron notebook','windows surface','macbook air','asus chromebook','hp pavilion desktop') THEN user_id ELSE NULL END) AS computer,
COUNT(DISTINCT CASE WHEN device IN ('iphone 5s','nokia lumia 635','amazon fire phone','iphone 4s','htc one','iphone 5','samsung galaxy s4') THEN user id ELSE NULL END) AS phone,
COUNT(DISTINCT CASE WHEN device IN ('kindle fire','samsung galaxy note','ipad mini','nexus 7','nexus 10','samsumg galaxy tablet','nexus 5','ipad air') THEN user id ELSE NULL END) AS tablet
FROM events
WHERE event type = 'engagement'
AND event name = 'login'
GROUP BY 1
ORDER BY 1 ;
```

USER GROWTH FOR PRODUCT

month	Monthly users	Computers	phone	tablets
August	1	0	0	1
July	16	11	7	1
June	603	375	163	110
May	824	469	238	175

USERS GETTING RETAINED WEEKLY AFTER SIGNING-UP FOR A PRODUCT.

```
SELECT date_format( occurred_at,"%W") AS week,
 COUNT(CASE WHEN event_type = 'signup_flow' THEN user_id ELSE NULL END) AS signup
 FROM events
GROUP BY 1
ORDER BY 1
```

week	signup
Friday	246
Monday	216
Saturday	56
Sunday	67
Thursday	263
Tuesday	238
Wednesday	251

TO MEASURE THE ACTIVENESS OF A USER. MEASURING IF THE USER FINDS QUALITY IN A PRODUCT/SERVICE WEEKLY. WEEKLY ENGAGEMENT PER DEVICE

```
select distinct(device),
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Monday') THEN user_id ELSE NULL END) AS Monday,
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Tuesday') THEN user_id ELSE NULL END) AS Tuesday,
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Wednesday') THEN user_id ELSE NULL END) AS
Wednesday,
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Thursday') THEN user_id ELSE NULL END) AS Thursday,
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Friday') THEN user_id ELSE NULL END) AS Friday,
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Saturday') THEN user_id ELSE NULL END) AS Saturday,
count(distinct CASE WHEN date_format(occurred_at,'%W') IN ('Sunday') THEN user_id ELSE NULL END) AS Sunday
from  events
WHERE event_type = 'engagement'
group by 1
```

	acer aspire desktop	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	Acer aspire desktop	5	1	5	3	9	1	0
	acer aspire notebook	4	9	12	8	6	2	1
	amazon fire phone	3	1	3	2	4	1	0
	asus chrome book	6	10	16	8	8	2	2
	dell inspiron desktop	8	6	9	5	14	2	5
	dell inspiron notebook	11	14	17	15	18	2	5
	hp pavilion desktop	8	13	4	10	9	1	1
	htc one	5	4	4	4	7	0	0
	ipad air	11	13	8	13	10	3	3
	ipad mini	4	6	3	5	11	0	3
	iphone 4s	6	9	7	11	13	4	4
	iphone 5	19	19	22	31	26	8	12
	iphone 5s	10	22	14	12	18	5	1
	kindle fire	1	4	4	7	5	3	0
	lenovo thinkpad	24	28	42	36	34	13	11

	Mac mini	4	4	1	2	1	1	0
	macbook air	30	26	28	40	23	3	5
	macbook pro	38	59	51	56	43	14	16
	nexus 10	9	6	4	5	3	2	0
	nexus 5	14	14	15	17	13	5	4
	nexus 7	6	6	12	9	6	2	3
	nokia lumia 635	4	4	6	3	6	3	1
	samsumg galaxy tablet	0	2	3	2	4	0	1
	samsung galaxy note	3	2	3	5	4	1	0
	samsung galaxy s4	17	18	12	23	16	4	4
	windows surface	3	6	5	4	2	0	2

EMAIL ENGAGEMENT: USERS ENGAGING WITH THE EMAIL SERVICE.

CALCULATE THE EMAIL ENGAGEMENT METRICS?

```
SELECT date_format( occurred_at,"%W") AS week,
COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE NULL END) AS weekly_emails,
COUNT(CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE NULL END) AS reengagement_emails,
COUNT(CASE WHEN action = 'email_open' THEN user_id ELSE NULL END) AS email_opens,
COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE NULL END) AS email_clickthroughs
FROM email_events
GROUP BY 1
ORDER BY 1
```

week	Weekly email	Re engagment email	Email opens	Email click throughs
Friday	10514	666	3711	1651
Monday	15927	635	5431	2230
Saturday	0	635	561	516
Sunday	0	711	619	565
Thursday	10883	177	3430	1289
Tuesday	10323	660	3711	1649
Wednesday	9620	169	2996	1110

APPROACH

For this project, I have used my SQL to extract the required data from the given database

using the Join function, subqueries, Aggregation, where condition, Group by, Distinct and windows functions required.

In consideration provided all the reports asked by the marketing department and ops team, support team,

TECH-STACK USED

- MySQL Server -8.0
- MySQL shell - 8.0.31
- MySQL Work Bench - 8.0CE

INSIGHTS & RESULT

- In this project I have gained knowledge on sub query ,window function ,how to use aggregation function with conditions .
- From this project, I got an idea about how as a business or data analyst we work on real-time data to take any data-driven decision.
- To get the weekly retention per device first we take the distinct device column and then we count the user id when the week is 'Monday' as 'Monday' Column so on with rest of the weeks too.It was a very good experience working on such kind of project.
- It helped me a lot to understand the analysis process well, and to provide insights for the best decision possible