Operation Analytics and Investigating Metric Spike

Sowmiya R **(Advanced sql)**

Introduction

Operational Analytics is a category of business analytics that's focused on syncing data from your warehouse directly to your business tools – thus ensuring that everyone across your organization has access to the same data regardless of their technical skills. Using operational analytics, you can review data in what's known as Near Real-Time

(NRT) as it enters your systems. This gives you immediate insights that can alert you to failures in your system before they escalate.

Case Study 1 (Job Data)

- **Number of jobs reviewed:** Amount of jobs reviewed over time.
- **Throughput:** It is the no. of events happening per second.
- **Percentage share of each language:** Share of each language for different contents.
- **Duplicate rows:** Rows that have the same value present in them.

Case Study 2 (Investigating metric spike)

- **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
- **User Growth:** Amount of users growing over time for a product.
- **Weekly Retention:** Users getting retained weekly after signing-up for a product.
- **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
- **Email Engagement:** Users engaging with the email service.

We were required to provide a detailed report for the below two operations mentioning the answers for the below related questions.........

PROJECT DESCRIPTION

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 it's very important to investigate metric spike.

APPROACH

The application that the below questions is been answered by the MySQL workbench. First, I need to analyze the given set of instructions through the dataset has been given with this project description. Then need to transfer the excel dataset to the SQL query.

TECH-STACK USED

MySQL Workbench is been used to compile the queries related to the questions . (MySQL 8.0.33)

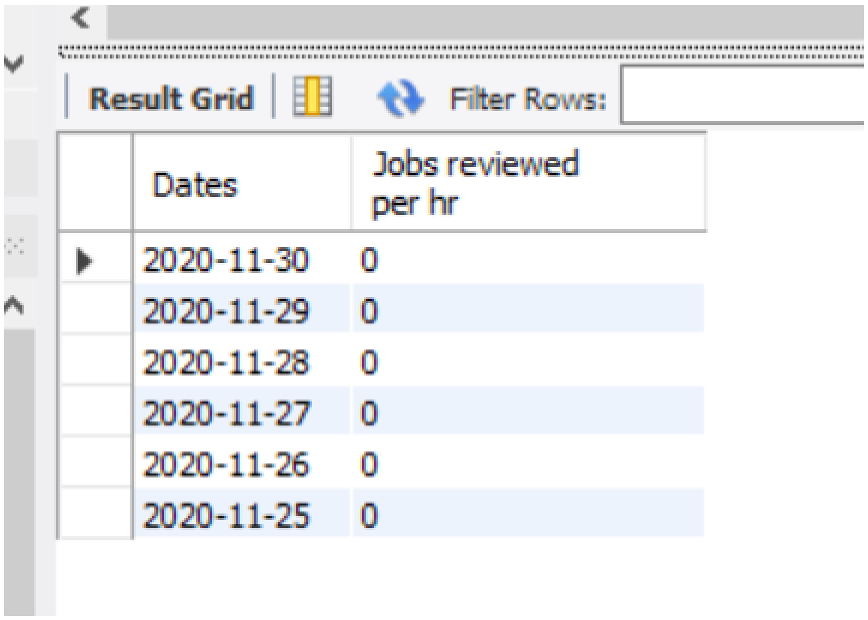
INSIGHTS

Case Study 1 (Job Data)

1. **Number of jobs reviewed:** Amount of jobs reviewed over time.

Our task is to Calculate the number of jobs reviewed per hour per day for November 2020.

```
SELECT ds AS Dates, (round(count(job_id)/sum(time_spent))*3600) as
'Jobs reviewed per hr'

from `sql project-1 table (1)`

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

group by ds;
```



Dates	Jobs reviewed per hr
2020-11-30	0
2020-11-29	0
2020-11-28	0
2020-11-27	0
2020-11-26	0
2020-11-25	0

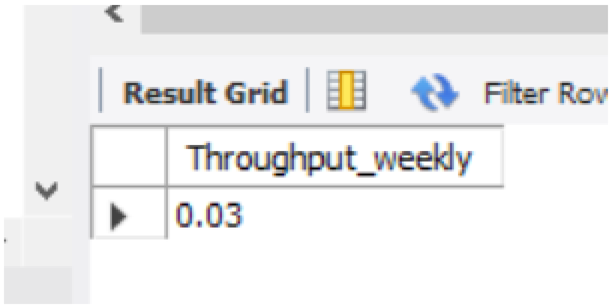
2. **Throughput:** It is the no. of events happening per second.

Our task is to Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

```
SELECT (ROUND(COUNT(event)/sum(time_spent),2)) as
Throughput_weekly
```

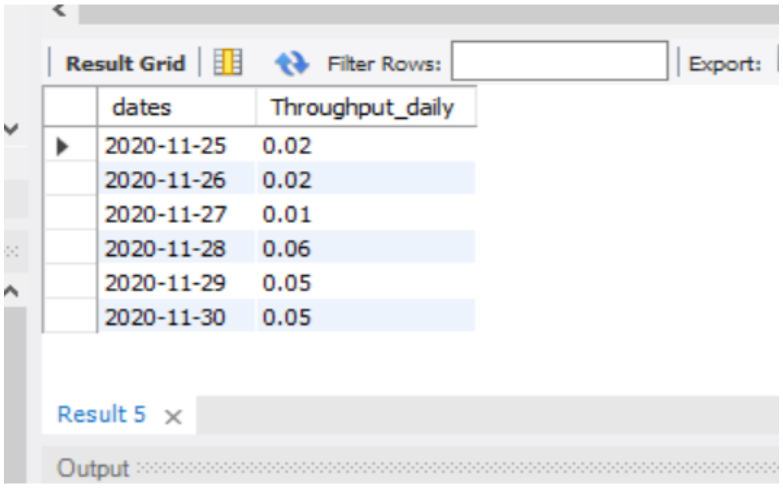
```
FROM `sql project-1 table (1)`;
```

The weekly throughput is 0.03



	Throughput_weekly
▶	0.03

```
select ds as dates,  (ROUND(COUNT(event)/sum(time_spent),2)) as
Throughput_daily

FROM `sql project-1 table (1)`

group by ds

order by ds;
```

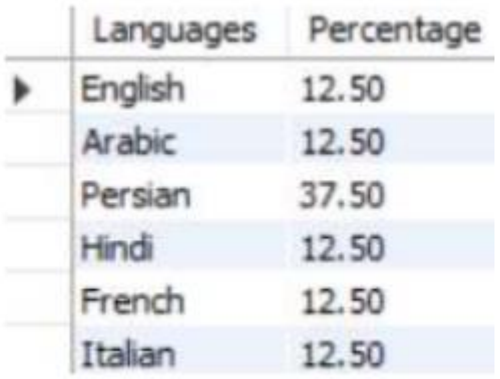


	dates	Throughput_daily
▶	2020-11-25	0.02
	2020-11-26	0.02
	2020-11-27	0.01
	2020-11-28	0.06
	2020-11-29	0.05
	2020-11-30	0.05

3. **Percentage share of each language:** Share of each language for different contents.

Our task is to calculate the percentage share of each language in the last 30 days.

```
SELECT  'language' AS Languages, (ROUND (100* COUNT(*)/total,2)) as
percentage_of_lang

FROM  `sql project-1 table (1)`

CROSS JOIN (SELECT COUNT(*) as total FROM `sql project-1 table (1)`)
sub_total

GROUP BY languages;
```



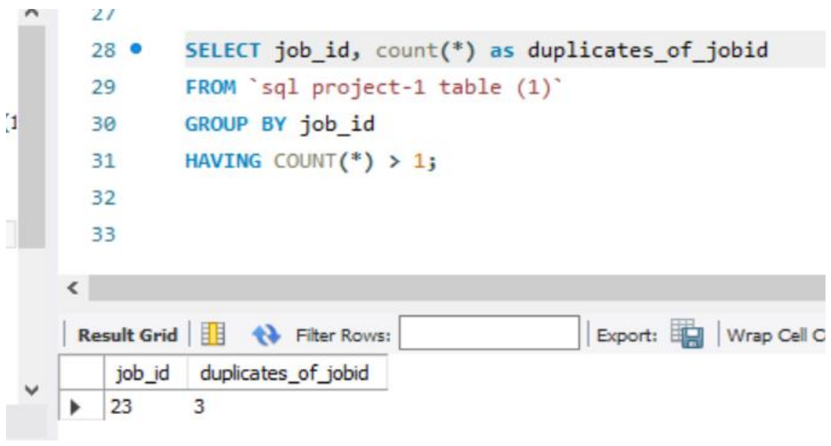
Languages	Percentage
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

4. **Duplicate rows:** Rows that have the same value present in them.

Our task is to the display duplicates from the table.

```
SELECT job_id, count(*) as duplicates_of_jobid
```

```
FROM `sql project-1 table (1)`

GROUP BY job_id

HAVING COUNT(*) > 1;
```



```
SELECT job_id, count(*) as duplicates_of_jobid
FROM `sql project-1 table (1)`
GROUP BY job_id
HAVING COUNT(*) > 1;
```

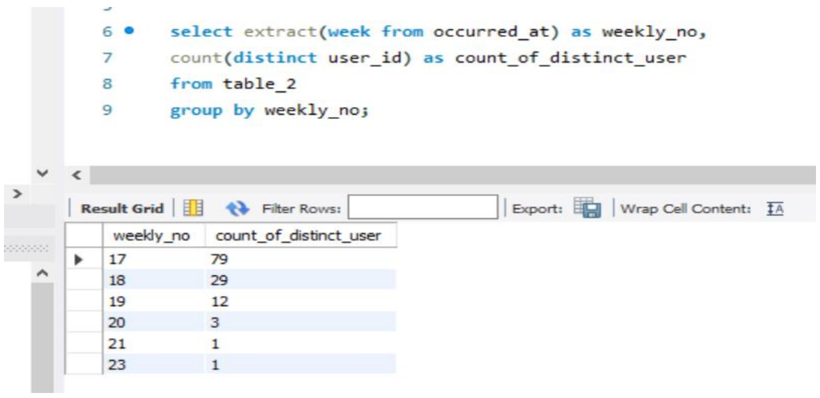
job_id	duplicates_of_jobid
23	3

Case Study 2 (Investigating metric spike)

1. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Our task is to calculate the weekly user engagement.

```
SELECT  EXTRACT(week FROM occurred_at) AS weekly_no,

COUNT(distinct user_id) AS count_of_distinct_user

FROM table_2

GROUP BY weekly_no;
```



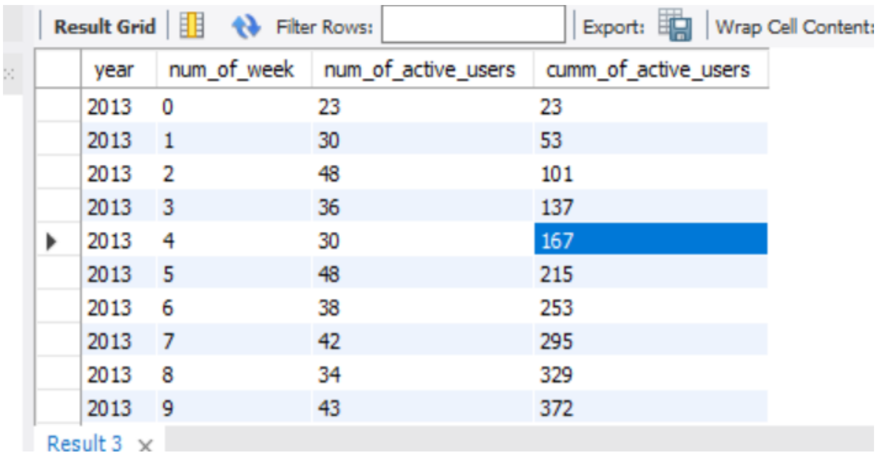
```
select extract(week from occurred_at) as weekly_no,
count(distinct user_id) as count_of_distinct_user
from table_2
group by weekly_no;
```

weekly_no	count_of_distinct_user
17	79
18	29
19	12
20	3
21	1
23	1

2. **User Growth:** Amount of users growing over time for a product.

Our task is to calculate the user growth for product.

```
SELECT year, num_of_week, num_of_active_users,

SUM(num_of_active_users) OVER(ORDER BY  year, num_of_week rows
BETWEEN unbounded preceding and current row)

AS cumm_of_active_users

FROM

(SELECT

   EXTRACT(year FROM a.activated_at) AS year,

   EXTRACT(week FROM a.activated_at)AS num_of_week,

   COUNT(distinct user_id) AS num_of_active_users

FROM table_1 a

WHERE state='active'

GROUP BY year, num_of_week

ORDER BY year, num_of_week

)a;
```

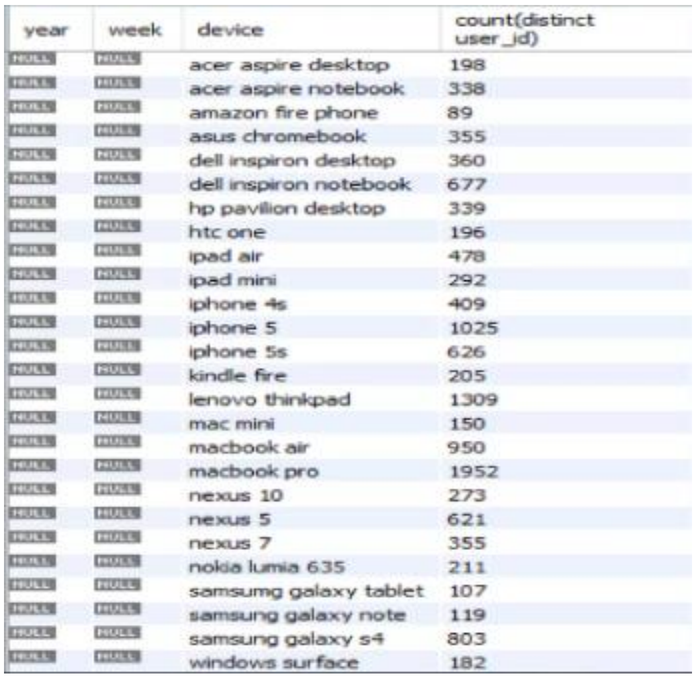


year	num_of_week	num_of_active_users	cumm_of_active_users
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372

3. **Weekly Retention:** Users getting retained weekly after signing-up for a product.

Our task is to calculate the weekly retention of users-sign up cohort.

```
select count(user_id),
       sum(case when retention_week = 1 then 1 else 0 end) as
retention_per_week
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
week_signup
from table_2
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 table_2
where event_type = 'engagement')b
on a.user_id = b.user_id
)
)
group by user_id
order by user_id;
```

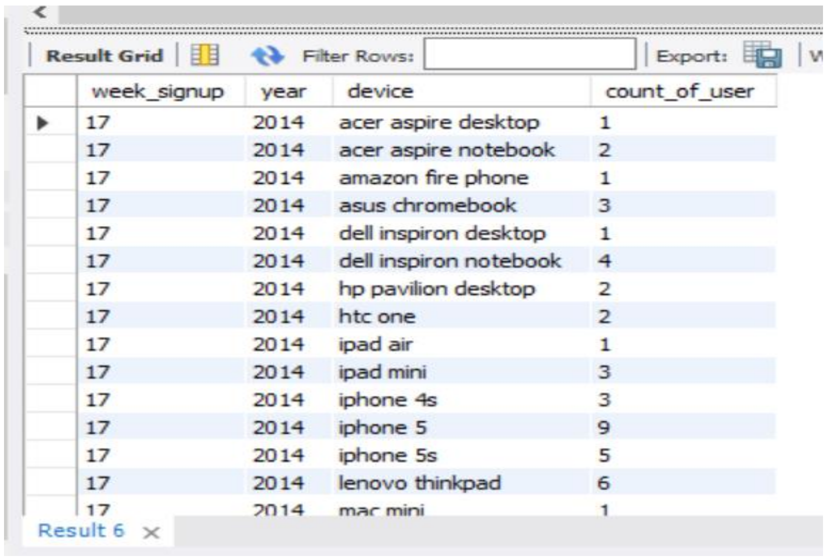


year	week	device	count(distinct user_id)
NULL	NULL	acer aspire desktop	198
NULL	NULL	acer aspire notebook	338
NULL	NULL	amazon fire phone	89
NULL	NULL	asus chromebook	355
NULL	NULL	dell inspiron desktop	360
NULL	NULL	dell inspiron notebook	677
NULL	NULL	hp pavilion desktop	339
NULL	NULL	htc one	196
NULL	NULL	ipad air	478
NULL	NULL	ipad mini	292
NULL	NULL	iphone 4s	409
NULL	NULL	iphone 5	1025
NULL	NULL	iphone 5s	626
NULL	NULL	kindle fire	205
NULL	NULL	lenovo thinkpad	1309
NULL	NULL	mac mini	150
NULL	NULL	macbook air	950
NULL	NULL	macbook pro	1952
NULL	NULL	nexus 10	273
NULL	NULL	nexus 5	621
NULL	NULL	nexus 7	355
NULL	NULL	nokia lumia 635	211
NULL	NULL	samsumg galaxy tablet	107
NULL	NULL	samsung galaxy note	119
NULL	NULL	samsung galaxy s4	803
NULL	NULL	windows surface	182

4. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

Our task is to calculate the weekly engagement per device.

```
select extract(week from occurred_at)as week_signup, extract(year from
occurred_at)as year, device,

count(distinct user_id)as 'count_of_user'

from table_2

where event_type='engagement'

group by 1,2,3

order by 1,2,3;
```

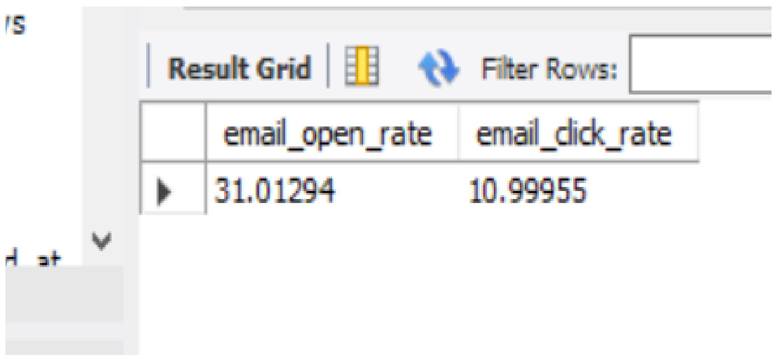


week_signup	year	device	count_of_user
17	2014	acer aspire desktop	1
17	2014	acer aspire notebook	2
17	2014	amazon fire phone	1
17	2014	asus chromebook	3
17	2014	dell inspiron desktop	1
17	2014	dell inspiron notebook	4
17	2014	hp pavilion desktop	2
17	2014	htc one	2
17	2014	ipad air	1
17	2014	ipad mini	3
17	2014	iphone 4s	3
17	2014	iphone 5	9
17	2014	iphone 5s	5
17	2014	lenovo thinkpad	6
17	2014	mac mini	1

5. **Email Engagement:** Users engaging with the email service.

Our task is 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_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_click_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 email_events
)a;
```



```
email_open_rate   email_click_rate
31.01294          10.99955
```

RESULT

In this project , I had learnt how to apply advanced SQL queries like group by and order by clauses together, cases and extracting dates and so on. And I have understand that how to find the valuable insights which help the business to grow, or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Questions like these must be answered daily and for that its very important to investigate metric spike.

DRIVE LINK

https://drive.google.com/file/d/1Bs5mmoI7THWJmUfB-hHOZ5MKXZ-e9n0a/view?usp=drive_link