OPERATION AND METRIC ANALYTICS

-RANAPRATAP GHOSH

Project Description

- Operations analysis is the analysis of the complete end-to-end operations of a business. With this, the companies,can identify areas for improvement.You work closely with operations teams, support teams and marketing teams to help them better understand the data they collect.
- Since it is one of the most important parts of a business,this analysis is further used to predict the overall growth or downfall of a startup or established business. This means better automation, better understanding between Cross functional teams, and more efficient workflows.
- Studying improving metrics is also an important part of operational analysis because you are a data analyst and you need to be able to understand or make other teams understand the following questions: Why engage daily decrease? Why are sales down? ETC. Questions like these need to be answered on a daily basis, so it's important to check that the metrics are increasing or decreasing.

Approach

- We'll go through the data already available in csv form, tables can be created from for the respective databases using different tools.

- When the tables are finally created, a query is written and executed to obtain the tables most relevant to the problem. The resulting table is then exported for further pro cessing.

- These charts help us better understand the data, help us research and provide our te ams with the information they need.

Tech-Stack Used

- PostgreSQL server
- PostgreSQL documentation
- MySQL Workbench 8.0

The software offers better data security, more than built-in features and on-demand scalability, and it also provides auto-completion functionality that allows developers to easily write queries.

Case Study 1

1. Number of jobs reviewed per hour per day for Nov 2020

Query-

```
select ds,count(job_id) as jobs_per_day, sum(time_spent)/3600 as hours_spent

from case_study_1

where ds >='2020-11-01' and ds <='2020-11-30'

group by ds

order by ds;
```

Result:

Result Grid | Filter Rows:

ds	job_per_day	hours_spent
2020-11-30	2	0.011111...
2020-11-29	1	0.005555...
2020-11-28	2	0.009166...
2020-11-27	1	0.028888...
2020-11-26	1	0.015555...
2020-11-25	1	0.0125

2. 7 day rolling average of throughput

Query:

```
WITH A AS ( SELECT ds, COUNT(job_id) AS jobs, SUM(time_spent) AS total_time
FROM case_study_1
GROUP BY ds)
SELECT ds, SUM(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) AS 7d_rolling_throughput_avg FROM A
```

Result:

ds	7d_rolling_throughput_avg
2020-11-25	0.0222
2020-11-26	0.0198
2020-11-27	0.0146
2020-11-28	0.0210
2020-11-29	0.0233
2020-11-30	0.0268

3. The percentage share of each language used in last 30 days

Query:

```
SELECT language, count(*) as num_jobs, sum(count(*)) over() as total_jobs,

count(*) * 100.0 / sum(count(*)) Over() as 'language Percentage'

 FROM case_study_1

GROUP BY language;
```

Result:

language	num_jobs	total_jobs	language Percentage
English	1	8	12.50000
Arabic	1	8	12.50000
Persian	3	8	37.50000
Hindi	1	8	12.50000
French	1	8	12.50000
Italian	1	8	12.50000

4. Display duplicates from the table

Query:

```
SELECT ds, any_value(job_id) as job_id, any_value(actor_id) as
actor_id, any_value(event) as event, any_value(language) as
language, any_value(time_spent) as time_spent, any_value(org) as
org
FROM case_study_1
GROUP BY ds
HAVING count(ds) > 1;
```

Result:

ds	job_id	actor_id	event	language	time_spent	org
2020-11-30	21	1001	skip	English	15	A
2020-11-28	23	1005	transfer	Persian	22	D

Case Study 2

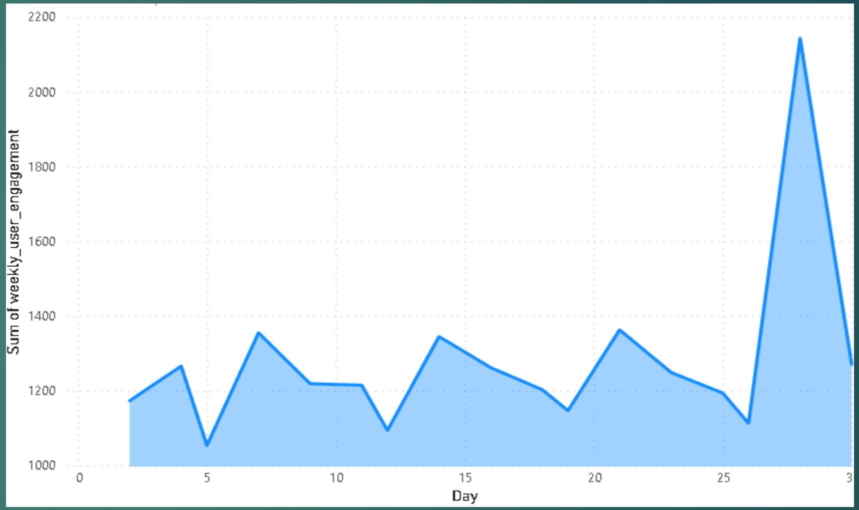
Calculate the weekly user engagement

Query:

```
select date_trunc('week', e.occurred_at) as date,
count(distinct e.user_id) as weekly_user_engagement
from events e
where e.event_type = 'engagement' AND
e.event_name = 'login'
group by 1
order by 1
```

Calculate the weekly user engagement

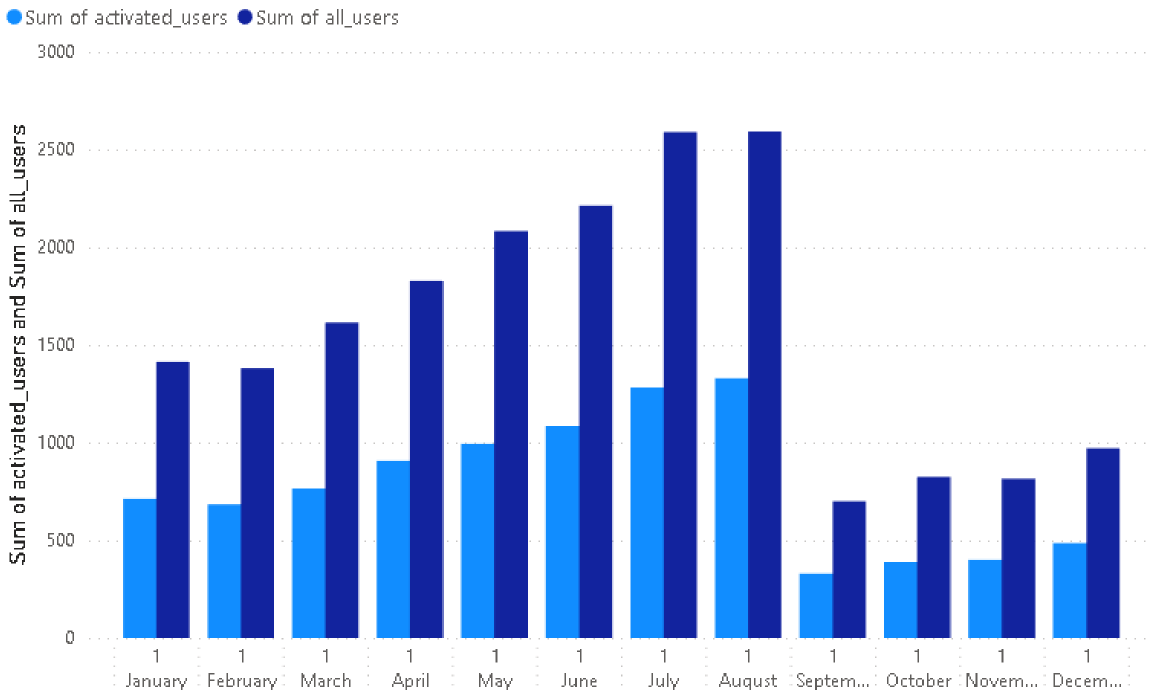
date	weekly_user_engagement
2014-04-28 00:00:00	701
2014-05-05 00:00:00	1054
2014-05-12 00:00:00	1094
2014-05-19 00:00:00	1147
2014-05-26 00:00:00	1113
2014-06-02 00:00:00	1173
2014-06-09 00:00:00	1219
2014-06-16 00:00:00	1262
2014-06-23 00:00:00	1249
2014-06-30 00:00:00	1271
2014-07-07 00:00:00	1355
2014-07-14 00:00:00	1345
2014-07-21 00:00:00	1363
2014-07-28 00:00:00	1442
2014-08-04 00:00:00	1266
2014-08-11 00:00:00	1215
2014-08-18 00:00:00	1203
2014-08-25 00:00:00	1194



Calculate the user growth for product

Query:

```
select date_trunc('month', u.created_at) as
month, count(*) as all_users,
count(case when u.activated_at is not null
then u.user_id else null end) as activated_users
from users u
where u.created_at >= '2013-01-01' and
u.created_at <= '2014-08-31'
group by 1
order by 1
```



	month	all_users	activated_users
1	2013-01-01 00:0...	332	160
2	2013-02-01 00:0...	328	160
3	2013-03-01 00:0...	383	150
4	2013-04-01 00:0...	410	181
5	2013-05-01 00:0...	486	214
6	2013-06-01 00:0...	485	213
7	2013-07-01 00:0...	608	284
8	2013-08-01 00:0...	636	316
9	2013-09-01 00:0...	699	330
10	2013-10-01 00:0...	826	390
11	2013-11-01 00:0...	816	399
12	2013-12-01 00:0...	972	486
13	2014-01-01 00:0...	1083	552
14	2014-02-01 00:0...	1054	525
15	2014-03-01 00:0...	1231	615
16	2014-04-01 00:0...	1419	726
17	2014-05-01 00:0...	1597	779
18	2014-06-01 00:0...	1728	873
19	2014-07-01 00:0...	1983	997
20	2014-08-01 00:0...	1958	1013

Calculate the user growth for product(Charts)

Calculate the weekly retention of users-sign up cohort

Query:

```
select date_trunc('week', z.occurred_at) as "week",
avg(z.age_at_event) as "Average age during week",
count(distinct case when z.user_age > 70 then z.user_id else null
end) as "10+ weeks",
count(distinct case when z.user_age < 70 and z.user_age >= 63
then z.user_id else null end) as "9 weeks",
 count(distinct case when z.user_age < 63 and z.user_age >= 56
then z.user_id else null end) as "8 weeks",
count(distinct case when z.user_age < 56 and z.user_age >= 49
then z.user_id else null end) as "7 weeks",
count(distinct case when z.user_age < 49 and z.user_age >= 42
then z.user_id else null end) as "6 weeks",
 count(distinct case when z.user_age < 42 and z.user_age >= 35
then z.user_id else null end) as "5 weeks",
count(distinct case when z.user_age < 35 and z.user_age >= 28
then z.user_id else null end) as "4 weeks",
```

Calculate the weekly retention of users-sign up cohort

Query:

```
count(distinct case when z.user_age < 28 and z.user_age >= 21 then
z.user_id else null end) as "3 weeks",
count(distinct case when z.user_age < 21 and z.user_age >= 14 then
z.user_id else null end) as "2 weeks",
 count(distinct case when z.user_age < 14 and z.user_age >= 7 then
z.user_id else null end) as "1 week",
count(distinct case when z.user_age < 7 then z.user_id else null end) as
"less than a week"
from( select e.occurred_at, u.user_id, date_trunc('week', u.activated_at)
as activation_week, extract('day' from e.occurred_at - u.activated_at)
as age_at_event,
extract('day' from '2014-08-31'::timestamp  - u.activated_at) as user_age
from users u
```

Calculate the weekly retention of users-sign up cohort

Query:

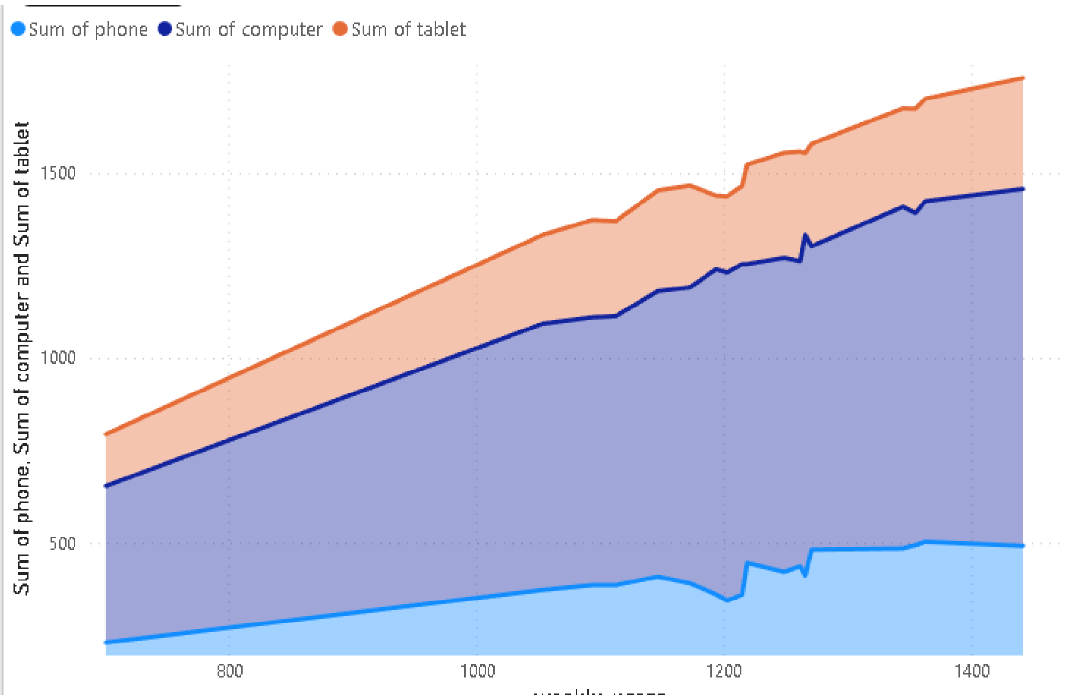
```
join events e on e.user_id = u.user_id
and e.event_type = 'engagement' and e.event_name = 'login' and
occurred_at >= '2014-05-01'  and occurred_at < '2014-08-31'
 where u.activated_at is not null) z
group by 1
order by 1
```

Calculate the weekly retention of users-sign up cohort(Charts)

week	Average age ...	10+ weeks	9 weeks	8 weeks	7 weeks	6 weeks	5 weeks	4 weeks	3 weeks	2 weeks	1 week	less than a w...
2014-04-28 0...	124.0072	701	0	0	0	0	0	0	0	0	0	0
2014-05-05 0...	124.3817	1054	0	0	0	0	0	0	0	0	0	0
2014-05-12 0...	131.9386	1094	0	0	0	0	0	0	0	0	0	0
2014-05-19 0...	132.3266	1147	0	0	0	0	0	0	0	0	0	0
2014-05-26 0...	132.3454	1113	0	0	0	0	0	0	0	0	0	0
2014-06-02 0...	131.8311	1173	0	0	0	0	0	0	0	0	0	0
2014-06-09 0...	131.0426	1219	0	0	0	0	0	0	0	0	0	0
2014-06-16 0...	136.4806	1246	7	0	0	0	0	0	0	0	0	0
2014-06-23 0...	136.2789	1025	205	10	0	0	0	0	0	0	0	0
2014-06-30 0...	136.4193	912	146	200	8	0	0	0	0	0	0	0
2014-07-07 0...	135.8888	896	96	129	222	9	0	0	0	0	0	0
2014-07-14 0...	143.4488	830	60	81	149	214	9	0	0	0	0	0
2014-07-21 0...	141.7028	788	41	59	94	143	219	16	0	0	0	0
2014-07-28 0...	144.0787	803	29	43	80	91	147	235	12	0	0	0
2014-08-04 0...	140.7322	677	22	35	51	52	77	152	191	8	0	0
2014-08-11 0...	125.9943	562	19	32	37	35	55	92	127	243	13	0
2014-08-18 0...	128.0217	520	15	25	27	18	37	64	68	156	260	11
2014-08-25 0...	131.7819	469	16	14	23	18	33	43	50	75	168	258

Calculate the weekly engagement per device

Query:
```
select date_trunc('week', e.occurred_at) as week,
count(distinct e.user_id) as weekly_users,
 count(distinct case when e.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 e.user_id else null end) as
computer,
 count(distinct case when e.device in ('iphone 5s','nokia lumia 635','amazon
fire phone','iphone 4s', 'htc one','iphone 5', 'samsung galaxy s4') then
e.user_id else null end) as phone,
 count(distinct case when e.device in ('kindle fire','samsung galaxy
note','ipad mini','nexus 7', 'nexus 10','samsung galaxy tablet','nexus 5','ipad
air') then e.user_id else null end) as tablet from events e where e.event_type
= 'engagement' and e.event_name = 'login'
group by 1
order by 1
```



	week	weekly_users	computer	phone	tablet
1	2014-04-28 00:00:00	701	423	231	140
2	2014-05-05 00:00:00	1054	720	373	240
3	2014-05-12 00:00:00	1094	724	386	263
4	2014-05-19 00:00:00	1147	772	409	272
5	2014-05-26 00:00:00	1113	727	386	257
6	2014-06-02 00:00:00	1173	800	391	275
7	2014-06-09 00:00:00	1219	808	446	269
8	2014-06-16 00:00:00	1262	825	437	296
9	2014-06-23 00:00:00	1249	849	422	284
10	2014-06-30 00:00:00	1271	820	482	277
11	2014-07-07 00:00:00	1355	898	494	282
12	2014-07-14 00:00:00	1345	924	485	266
13	2014-07-21 00:00:00	1363	921	503	277
14	2014-07-28 00:00:00	1442	965	492	300
15	2014-08-04 00:00:00	1266	921	412	221
16	2014-08-11 00:00:00	1215	894	360	211
17	2014-08-18 00:00:00	1203	887	344	206
18	2014-08-25 00:00:00	1194	879	361	199

Calculate the weekly engagement per device(Charts)

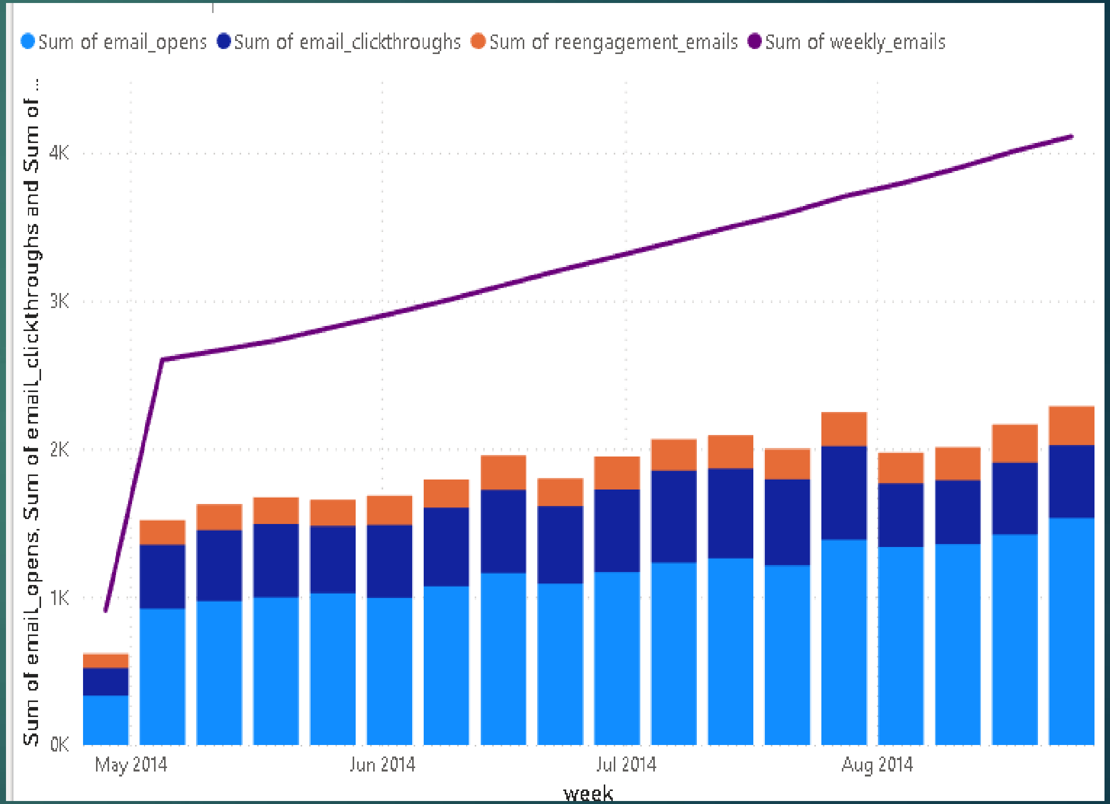
Calculate the email engagement matrices

Query:

```
select date_trunc('week',e.occurred_at) as week,
count(case when e.action = 'sent_weekly_digest' then
e.user_id else null end) as weekly_emails,
count(case when e.action =
'sent_reengagement_email' then e.user_id else null
end) as reengagement_emails, count(case when
e.action = 'email_open' then e.user_id else null end) as
email_opens,
count(case when e.action = 'email_clickthrough' then
e.user_id else null end) as email_clickthroughs
from email_events e
group by 1
order by 1
```

Calculate the email engagement matrices (Charts)

	week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
1	2014-04-28 00:00:00	908	98	332	187
2	2014-05-05 00:00:00	2602	164	919	434
3	2014-05-12 00:00:00	2665	175	971	479
4	2014-05-19 00:00:00	2733	179	995	498
5	2014-05-26 00:00:00	2822	179	1026	453
6	2014-06-02 00:00:00	2911	199	993	492
7	2014-06-09 00:00:00	3003	190	1070	533
8	2014-06-16 00:00:00	3105	234	1161	563
9	2014-06-23 00:00:00	3207	187	1090	524
10	2014-06-30 00:00:00	3302	222	1168	559
11	2014-07-07 00:00:00	3399	214	1230	622
12	2014-07-14 00:00:00	3499	226	1260	607
13	2014-07-21 00:00:00	3592	206	1211	584
14	2014-07-28 00:00:00	3706	230	1386	633
15	2014-08-04 00:00:00	3793	206	1336	432
16	2014-08-11 00:00:00	3897	224	1357	430
17	2014-08-18 00:00:00	4012	257	1421	487
18	2014-08-25 00:00:00	4111	263	1533	493



Conclusion

- In conclusion, operational analysis is one of the most powerful tools for examinin g matrix spikes in a organization's data and determining why that organization is succeeding or faili ng. This information then helps team members make recommendations that can help re solve the issues that the organization has faced.
- We can see from Case Study 1 that the throughput moving average is better bec ause it gives us those large occasional fluctuations, helping analysts and team members better under stand the data.
- From Case Study 2, we can see how users interact with the business and help the product grow. We can also see how well the organization is able to retain its leg acy users using the User Retention Group. We can also understand what type of device the users pri marily use to access products and how users interact with messaging services.

Thank You