Operation Analytics and Investigating Metric Spike

Project Description:

Operation Analytics involves thoroughly examining a company's entire operations. By doing this, the company identifies areas that need improvement. I collaborate closely with various teams like operations, support, and marketing, assisting them in uncovering valuable insights from the data they gather.

This study is critical for determining a company's overall success or failure. It improves automation, team understanding, and productivity of workflows.

In this project, we delve into questions like why there's a dip in daily engagement or why sales have taken a hit. Understanding these questions daily is essential, requiring an investigation into metric spikes.

Project Approach:

At first, I took some time to educate myself with the data and tables given. While reviewing the data, I clarified questions, such as understanding what job_id, actor_id, and event meant, and focused on the key components.

This project was created using SQL Workbench. First, I created a database using the dataset file provided by the company. The next phase involves loading data into SQL Workbench and analyzing it to answer queries such as

1. Why daily engagement has decreased?
2. Why sales dropped?

These types of questions have must be answered on a daily basis, and investigating metric spikes is critical.

The tech stack I used is **My SQL Workbench 8.0CE.**

Insights:

Case Study 1 – Job Data Analysis

Task1:

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

SQL Query:

```
Select
count(distinct job_id)/(30*24) as num_jobs_reviewed
from job_data
where
ds between '2020-11-01' and '2020-11-30';
```

Result:

Dates	Jobs Reviewed per Hour per Day
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

Task2:

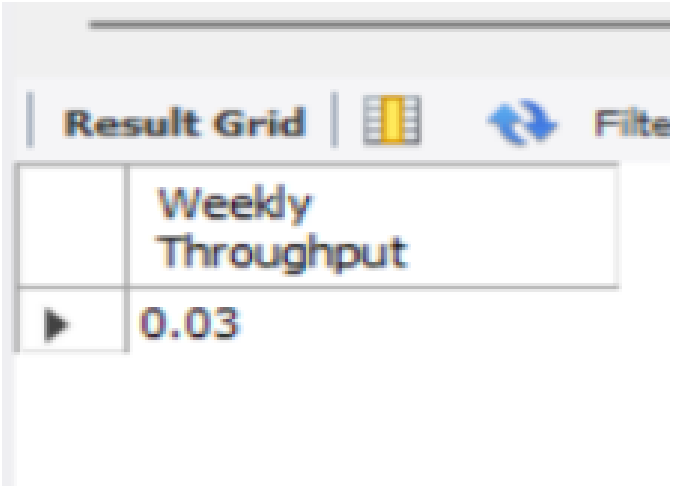
Calculate the 7-day rolling average of throughput (number of events per second). For throughput, do you prefer daily metric or 7-day rolling and why?

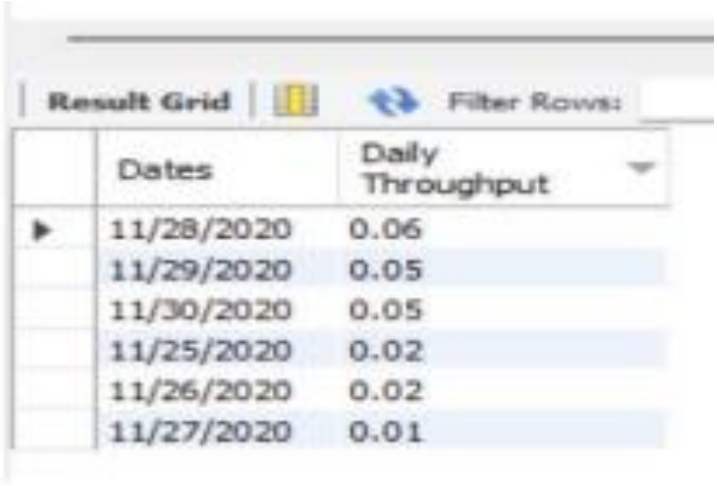
SQL Query:

```
select ds, jobs_reviewed,
avg(jobs_reviewed)over(order by ds rows between 6 preceding and current row)
```

```
as throughput_7_rolling_avg
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
order by ds
)a;
```

Result:





Task3:

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

SQL Query:

```
select language, num_jobs,
100.0* num_jobs/total_jobs as pct_share_jobs
from
(
select language, count(distinct job_id) as num_jobs
from job_data
group by language
)a
cross join
(
select count(distinct job_id) as total_jobs
from job_data
)b;
```

Result:

Result Grid | Filter Rows: | Export:

LANGUAGE	LANGUAGE_COUNT	Percentage_Share
Persian	3	37.5000
Italian	1	12.5000
Hindi	1	12.5000
French	1	12.5000
English	1	12.5000
Arabic	1	12.5000

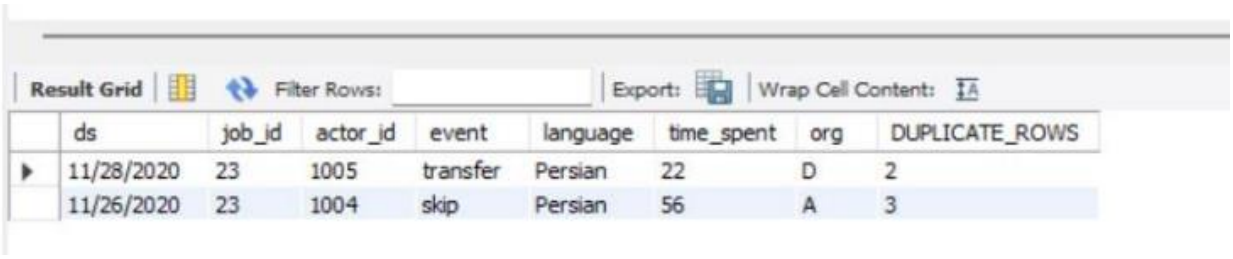
Task4:

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

SQL Query:

```
select * from
(
select *,
row_number()over(partition by job_id) as rownum
from job_data
)a
where rownum>1;
```

Result:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	job_id	actor_id	event	language	time_spent	org	DUPLICATE_ROWS
11/28/2020	23	1005	transfer	Persian	22	D	2
11/26/2020	23	1004	skip	Persian	56	A	3

My Insights:

1. The number of distinct jobs reviewed per hour per day for November 2020 is in the table above.
2. We used the 7-day rolling average of throughput as it gives the average for all the days right from day 1 to day 7 whereas, daily metric gives the average for only that particular day itself.
3. The percentage share of Persian language is the most (37.5%).
4. There are two duplicate rows if we partition the data by job_id.

Case Study 2- Investigating metric spike

Task1:

Calculate the weekly user engagement?

SQL Query:

```
select
 extract(week from occurred_at) as num_week,
 count(distinct user_id) as no_of_distinct_user
from tutorial.yammer_events
group by num_week;
```

Result:

Week Numbers	Weekly Active Users
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

Task2:

Calculate the user growth for product?

SQL Query:

```
select year, num_week, num_active_users,
sum(num_active_users) over(order by year, num_week rows between unbounded
preceding and current row)
as cumm_active_users
from
(select
 extract(year from a.activated_at) as year,
 extract(week from a.activated_at)as num_week,
 count(distinct user_id) as num_active_users
from tutorial.yammer_users a
where state='active'
group by year, num_week
order by year, num_week
)a;
```

Result:

	Months	Users	Growth in %
▶	1	712	NULL
	2	685	-3.79
	3	765	11.68
	4	907	18.56
	5	993	9.48
	6	1086	9.37
	7	1281	17.96
	8	1347	5.15
	9	330	-75.50
	10	390	18.18
	11	399	2.31
	12	486	21.80

Task3:

Calculate the weekly retention of users-sign up cohort?

SQL Query:

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

Result:

Week Numbers	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
17	740	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	788	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	601	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	555	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	495	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	521	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	542	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	535	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	500	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	495	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	493	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	486	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	501	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	533	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	430	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	496	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	499	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	518	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Task4:

Calculate the weekly engagement per device?

SQL Query:

```
select
extract(year from occured_at) as year_num,
extract(week from occured_at) as week_num,
device,
count(distinct user_id) as no_of_users
from tutorial.yammer_events
where event_type = 'engagement'
group by 1,2,3
order by 1,2,3;
```

Result:

Week Numbers	Dell Inspiron Notebook	iPhone 5	iPhone 4S	Windows Surface	Macbook Air	iPhone 5S	Macbook Pro	Kindle Fire	iPad Mini	Nexus 7	Nexus 5	Samsung Galaxy S4	Lenovo Thinkpad	Samsung Galaxy Tablet
17	46	65	21	10	54	42	143	6	19	18	40	52	86	8
18	77	113	46	10	121	73	252	27	30	30	73	82	153	11
19	83	115	44	16	112	79	266	21	36	41	87	91	178	6
20	84	125	55	21	119	79	256	23	32	32	103	93	173	9
21	80	137	45	17	110	74	247	30	23	29	91	84	167	6
22	92	125	45	15	145	71	251	21	34	45	96	105	176	10
23	103	152	53	14	124	79	266	25	33	36	88	99	176	14
24	99	142	53	22	152	79	255	25	39	49	87	101	165	11
25	105	137	40	22	121	78	275	24	30	51	89	99	197	12
26	89	152	50	21	134	94	269	26	43	46	87	112	192	12
27	89	163	67	33	142	83	302	25	35	40	84	116	202	15
28	103	151	61	33	148	93	295	31	35	39	85	122	220	9
29	113	144	60	28	148	90	295	37	34	45	77	123	209	13
30	127	152	65	19	159	103	322	25	35	62	84	103	206	9
31	113	135	56	19	147	71	321	14	27	38	69	100	207	8
32	104	119	34	10	125	67	307	12	30	25	67	82	179	6
33	110	110	35	15	133	65	312	14	28	30	70	80	191	12
34	105	101	50	18	136	70	292	13	25	33	70	90	193	14
35	9	2	6	3	10	3	17	3	2	2	4	6	16	0

Week Numbers	Acer Aspire Notebook	Asus Chromebook	HTC One	Nokia Lumia 635	Samsung Galaxy Note	Acer Aspire Desktop	Mac Mini	HP Pavilion Desktop	Dell Inspiron Desktop	iPad Air	Amazon Fire Phone	Nexus 10
17	20	21	16	17	7	9	6	14	18	27	4	16
18	33	42	19	33	15	26	13	37	58	52	9	30
19	41	27	30	23	11	23	18	40	36	55	12	25
20	40	41	29	22	18	23	26	30	52	59	11	22
21	47	38	21	25	20	29	18	44	41	51	5	25
22	41	52	24	25	19	25	25	38	52	58	5	27
23	43	49	20	31	14	22	18	54	53	41	16	45
24	40	43	20	35	20	24	29	56	59	57	11	38
25	47	38	21	37	14	28	21	52	52	57	13	29
26	35	49	23	42	9	29	11	46	60	56	13	29
27	49	52	27	31	15	29	15	56	53	55	10	37
28	49	50	26	35	10	30	28	56	56	54	6	26
29	53	49	31	43	16	28	31	58	54	52	12	25
30	60	56	31	34	15	33	23	42	54	70	12	36
31	55	56	13	28	14	31	24	51	44	55	14	24
32	55	62	18	28	12	35	20	51	57	48	12	30
33	46	49	19	27	13	39	32	38	37	40	14	23
34	63	47	25	17	13	30	30	36	49	39	11	25
35	3	6	2	2	1	1	2	1	1	0	0	2

Task5:

Calculate the email engagement metrics?

SQL Query:

```
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_opening_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_clicking_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 tutorial.yammer_events
)a;
```

Result:

Week	Weekly Digest Rate	Email Open Rate	Email Clickthrough Rate	Reengagement Email Rate
17	62.32	21.28	11.39	5.01
18	63.45	22.24	10.49	3.83
19	62.16	22.67	11.13	4.04
20	61.62	22.64	11.43	4.31
21	63.52	22.82	9.97	3.69
22	63.59	21.56	10.66	4.19
23	62.39	22.34	11.18	4.09
24	61.61	22.92	10.99	4.48
25	63.77	21.79	10.54	3.90
26	62.99	22.22	10.61	4.18
27	62.24	22.49	11.37	3.90
28	62.92	22.48	10.77	3.83
29	63.98	21.71	10.51	3.79
30	62.29	23.24	10.59	3.88
31	65.27	23.25	7.66	3.82
32	66.59	22.85	7.14	3.42
33	64.73	23.10	7.91	4.26
34	64.33	23.91	7.67	4.08
35	0.00	32.28	29.92	37.80

My Insights:

1. The weekly user engagement increased from week 18th to week 31st and then started declining from then onwards. This means that some of the users do not find much quality in the product/service in the last of the weeks.
2. There are in total 9381 active users from 1st week of 2013 to the 35th week of 2014.
3. The overall count of weekly engagement per device used is the most for MacBook users and iPhone users as given in the table above.
4. The email opening rate is around 34% and email clicking rate is around 15%. The users are engaging with the email service which is good for the company to expand.