Operation Analytics and Investigating Metric Spike

Approach-

THIS PROJECT WAS DEVELOPED USING SQL WORKBENCH1. A DATABASE IS CREATED MICRO_DB USING THE DATASET PROVIDED BY TRAINITY. THE DATA IS IMPORTED INTO SQL WORKBENCH. EXTENSIVE ANALYSIS IS CONDUCTED TO ANSWER CRITICAL QUESTIONS, SUCH AS IDENTIFYING REASONS FOR FLUCTUATIONS IN DAILY ENGAGEMENT AND SALES DIPS. THESE QUESTIONS NEED TO BE ADDRESSED DAILY, MAKING IT CRUCIAL TO INVESTIGATE METRIC SPIKES.

Tech stack used-

MYSQL WORKBENCH, SQL

TASK 1
JOB DATA ANALYSIS

Table used for Task- 1

Since the data provided for tast-1 was very less and was only a sample of data, so I curated the data and made some changes, inserted more rows.this is the table which has been used for task-1.

Job_data

job_id	actor_id	event	language	time_spent	org	ds
21	1001	skip	English	15	A	2020-11-30
22	1006	transfer	Arabic	25	B	2020-11-30
23	1003	decision	Persian	20	C	2020-11-29
23	1005	transfer	Persian	22	D	2020-11-28
25	1002	decision	Hindi	11	B	2020-11-28
11	1007	decision	French	104	D	2020-11-27
23	1004	skip	Persian	56	A	2020-11-26
20	1003	transfer	Italian	45	C	2020-11-25
21	1002	decision	Arabic	16	A	2020-11-24
20	1001	decision	Persian	22	C	2020-11-23
14	999	transfer	English	31	B	2020-11-24
19	998	transfer	French	54	C	2020-11-22
18	997	skip	Hindi	54	B	2020-11-21
18	996	decision	Arabic	26	A	2020-11-20
17	994	skip	Persian	33	C	2020-11-19
16	993	transfer	Hindi	53	B	2020-11-18
15	995	transfer	English	38	C	2020-11-17
13	993	decision	Hindi	53	C	2020-11-17
14	992	decision	English	36	A	2020-11-18
11	1009	transfer	Arabic	12	A	2020-11-15
10	1008	skip	English	40	B	2020-11-16
9	1007	decision	French	15	A	2020-11-15
12	1010	skip	Hindi	46	C	2020-11-14
26	1011	decision	Italian	56	D	2020-11-13
27	980	skip	Italian	26	B	2020-11-12
25	979	skip	Italian	14	C	2020-11-11
27	977	transfer	French	11	A	2020-11-11
28	976	decision	English	30	C	2020-11-10
29	975	transfer	Persian	29	C	2020-11-09
30	974	skip	Arabic	34	C	2020-11-08
31	973	skip	English	51	B	2020-11-07
32	988	transfer	French	46	C	2020-11-06
33	987	transfer	Hindi	49	B	2020-11-05
34	986	decision	French	47	C	2020-11-04
37	985	skip	French	36	D	2020-11-03
36	984	decision	Persian	15	D	2020-11-02
39	983	decision	Hindi	27	D	2020-11-01
40	982	transfer	English	23	A	2020-11-01
41	981	transfer	Italian	24	B	2020-11-05

Jobs Reviewed Over Time:

1. Calculate the number of jobs reviewed per hour for each day in November 2020.Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

Approach- To find the job reviewed per hour each day, I have used two approaches. In first one I have counted total no. of Job_id divided it in 30 days for each day and for per hour divided it with 24, which will give us the job reviewed per hour each day. In second i have group total time_spent on the date basis and to find job reviewed per hour divided it with 24.

Input-

```sql
/* Job Reviewed Over Time */
select count(job_id)/(30*24) as 'Job reviewed per hour each day' from job_data;
```

Or

```sql
/* Job Reviewed Over Time */
select ds as 'Date', count(job_id)/24 as 'Job reviewed per Hour' from job_data
where ds between '2020-11-01' and '2020-11-30'
group by ds
order by ds asc;
```

Output-

Job reviewed per hour each day
0.0542

Date	Job reviewed per Hour
2020-11-01	0.0833
2020-11-02	0.0417
2020-11-03	0.0417
2020-11-04	0.0417
2020-11-05	0.0833
2020-11-06	0.0417
2020-11-07	0.0417
2020-11-08	0.0417
2020-11-09	0.0417
2020-11-10	0.0417
2020-11-11	0.0833
2020-11-12	0.0417
2020-11-13	0.0417
2020-11-14	0.0417
2020-11-15	0.0833
2020-11-16	0.0417
2020-11-17	0.0833
2020-11-18	0.0833
2020-11-19	0.0417
2020-11-20	0.0417
2020-11-21	0.0417
2020-11-22	0.0417
2020-11-23	0.0417
2020-11-24	0.0833
2020-11-25	0.0417
2020-11-26	0.0417
2020-11-27	0.0417
2020-11-28	0.0833
2020-11-29	0.0417
2020-11-30	0.0833

Throughput Analysis:

2. Calculate the 7-day rolling average of throughput (number of events per second).Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

Answer - I prefer using the daily metrics, because it is more accurate than weekly metrics.

Input-

```sql
/* Daily Thoughput */
select ds as'Date',round(count(event)/sum(time_spent),2) as 'Daily throughput' from job_data
group by ds
order by ds;
```

```sql
/*Weekly Throughput*/
select week(ds),round(count(event)/sum(time_spent),2) as 'Weekly Throughput' from job_data
group by week(ds)
order by week(ds) asc;
```

Output-

week(ds)	Weekly Throughput
44	0.03
45	0.03
46	0.03
47	0.02
48	0.05

Date	Daily throughput
2020-11-01	0.04
2020-11-02	0.07
2020-11-03	0.03
2020-11-04	0.02
2020-11-05	0.03
2020-11-06	0.02
2020-11-07	0.02
2020-11-08	0.03
2020-11-09	0.03
2020-11-10	0.03
2020-11-11	0.08
2020-11-12	0.04
2020-11-13	0.02
2020-11-14	0.02
2020-11-15	0.07
2020-11-16	0.03
2020-11-17	0.02
2020-11-18	0.02
2020-11-19	0.03
2020-11-20	0.04
2020-11-21	0.02
2020-11-22	0.02
2020-11-23	0.05
2020-11-24	0.04
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

Language Share Analysis:

Calculate the percentage share of each language in the last 30 days. Write an SQL query to calculate the percentage share of each language over the last 30 days.

Input-

```sql
/* Language Share Analysis */
select count(*) as 'Total Rows' from job_data;
select language, concat((count(job_id)/39)*100,'%') as 'language share' from job_data
group by language;
```

Output-

	Total Rows
▶	39

language	language share
English	20.5128%
Arabic	12.8205%
Persian	17.9487%
Hindi	17.9487%
French	17.9487%
Italian	12.8205%

Duplicate Rows Detection:

Identify duplicate rows in the data. Write an SQL query to display duplicate rows from the job_data table.

Input-

```
/* Duplicate Rows Detection */
select actor_id, count(*) as duplicates
from job_data
group by actor_id
having count(*) > 1;
```

Output-

language	Language_share
English	12.5000 %
Arabic	12.5000 %
Persian	37.5000 %
Hindi	12.5000 %
French	12.5000 %
Italian	12.5000 %

TASK 2
INVESTIGATING METRIC SPIKE

Table used for Task- 2

These are the sample of the tables which has been used for task 2.

Users

user_id	company_id	language	state	created_at	activated_at
0	5737	english	active	2013-01-01 20:59:00	2013-01-01 21:01:00
3	2800	german	active	2013-01-01 18:40:00	2013-01-01 18:42:00
4	5110	indian	active	2013-01-01 14:37:00	2013-01-01 14:39:00
6	11699	english	active	2013-01-01 18:37:00	2013-01-01 18:38:00
7	4765	french	active	2013-01-01 16:19:00	2013-01-01 16:20:00
8	2698	french	active	2013-01-01 04:38:00	2013-01-01 04:40:00
11	3745	english	active	2013-01-01 08:07:00	2013-01-01 08:09:00

email_events

user_id	action	user_type	occurred_at
0	sent_weekly_digest	1	2014-05-06 09:30:00
0	sent_weekly_digest	1	2014-05-13 09:30:00
0	sent_weekly_digest	1	2014-05-20 09:30:00
0	sent_weekly_digest	1	2014-05-27 09:30:00
0	sent_weekly_digest	1	2014-06-03 09:30:00
0	email_open	1	2014-06-03 09:30:00

events

user_id	event_type	event_name	Location	device	user_type	occurred_at
10522	engagement	login	Japan	dell inspiron notebook	3	2014-05-02 11:02:00
10522	engagement	home_page	Japan	dell inspiron notebook	3	2014-05-02 11:02:00
10522	engagement	like_message	Japan	dell inspiron notebook	3	2014-05-02 11:03:00
10522	engagement	view_inbox	Japan	dell inspiron notebook	3	2014-05-02 11:04:00
10522	engagement	search_run	Japan	dell inspiron notebook	3	2014-05-02 11:03:00
10522	engagement	search_run	Japan	dell inspiron notebook	3	2014-05-02 11:03:00
10612	engagement	login	Netherlands	iphone 5	1	2014-05-01 09:59:00

Weekly User Engagement:

1. Measure the activeness of users on a weekly basis. Write an SQL query to calculate the weekly user engagement.

Input-

```sql
/*Weekly User Engagement*/
select extract(week from occurred_at) as Week,
count(distinct(user_id)) as 'Total Active Users'
from events
group by week;
```

Output-

Week	Total 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

User Growth Analysis:

Analyze the growth of users over time for a product. Write an SQL query to calculate the user growth for the product.

Input-

```sql
select Months, Users, round(((Users / LAG(Users, 1) over (order by Months)) - 1) * 100, 2) as "Growth in %"
from
    (
    select extract(month from created_at) as Months, count(activated_at) as Users
    from users
    where activated_at is not null
    group by Months
    order by Months
    ) sub;
```

Output-

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

Weekly Retention Analysis:

Analyze the retention of users on a weekly basis after signing up for a product. Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

Input-

```
Select First as 'Week Numbers',
Sum(case when Week_number = 0 then 1 else 0 end) as 'Week 0',
Sum(case when Week_number = 1 then 1 else 0 end) as 'Week 1',
Sum(case when Week_number = 2 then 1 else 0 end) as 'Week 2',
Sum(case when Week_number = 3 then 1 else 0 end) as 'Week 3',
Sum(case when Week_number = 4 then 1 else 0 end) as 'Week 4',
Sum(case when Week_number = 5 then 1 else 0 end) as 'Week 5',
Sum(case when Week_number = 6 then 1 else 0 end) as 'Week 6',
Sum(case when Week_number = 7 then 1 else 0 end) as 'Week 7',
Sum(case when Week_number = 8 then 1 else 0 end) as 'Week 8',
Sum(case when Week_number = 9 then 1 else 0 end) as 'Week 9',
Sum(case when Week_number = 10 then 1 else 0 end) as 'Week 10',
Sum(case when Week_number = 11 then 1 else 0 end) as 'Week 11',
Sum(case when Week_number = 12 then 1 else 0 end) as 'Week 12',
Sum(case when Week_number = 13 then 1 else 0 end) as 'Week 13',
Sum(case when Week_number = 14 then 1 else 0 end) as 'Week 14',
Sum(case when Week_number = 15 then 1 else 0 end) as 'Week 15',
Sum(case when Week_number = 16 then 1 else 0 end) as 'Week 16',
Sum(case when Week_number = 17 then 1 else 0 end) as 'Week 17',
Sum(case when Week_number = 18 then 1 else 0 end) as 'Week 18'
from
```

```
(
Select m.user_id, m.login_week, n.first, m.login_week-first as week_number
from
(select user_id, Extract(week from occurred_at) as login_week from events
group by 1, 2) as m,
(select user_id, min(Extract(week from occurred_at)) as first from events
group by 1) as n
where m.user_id = n.user_id
)sub
group by first
order by first;
```

Output-

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	663	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	596	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	427	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	358	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	317	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	326	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	328	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	339	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	305	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	288	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	292	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	274	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	270	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	294	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	215	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	267	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	286	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	279	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Weekly Engagement Per Device:

Measure the activeness of users on a weekly basis per device. Write an SQL query to calculate the weekly engagement per device.

Input-

```sql
select extract(week from occurred_at) as "Week Numbers",
count(distinct case when device in ('dell inspiron notebook') then user_id else null end) as "dell inspiron notebook",
count(distinct case when device in ('iphone 5') then user_id else null end) as "iphone 5",
count(distinct case when device in ('iphone 4s') then user_id else null end) as "iphone 4s",
count(distinct case when device in ('windows surface') then user_id else null end) as "windows surface",
count(distinct case when device in ('macbook air') then user_id else null end) as "macbook air",
count(distinct case when device in ('iphone 5s') then user_id else null end) as "iphone 5s",
count(distinct case when device in ('macbook pro') then user_id else null end) as "macbook pro",
count(distinct case when device in ('kindle fire') then user_id else null end) as "kindle fire",
count(distinct case when device in ('ipad mini') then user_id else null end) as "ipad mini",
count(distinct case when device in ('nexus 7') then user_id else null end) as "nexus 7",
count(distinct case when device in ('nexus 5') then user_id else null end) as "nexus 5",
count(distinct case when device in ('samsung galaxy s4') then user_id else null end) as "samsung galaxy s4",
count(distinct case when device in ('lenovo thinkpad') then user_id else null end) as "lenovo thinkpad",
count(distinct case when device in ('samsumg galaxy tablet') then user_id else null end) as "samsumg galaxy tablet",
count(distinct case when device in ('acer aspire notebook') then user_id else null end) as "acer aspire notebook",
count(distinct case when device in ('asus chromebook') then user_id else null end) as "asus chromebook",
count(distinct case when device in ('samsung galaxy note') then user_id else null end) as "samsung galaxy note",
count(distinct case when device in ('mac mini') then user_id else null end) as "mac mini",
count(distinct case when device in ('hp pavilion desktop') then user_id else null end) as "hp pavilion desktop",
count(distinct case when device in ('ipad air') then user_id else null end) as "ipad air",
count(distinct case when device in ('htc one') then user_id else null end) as "htc one",
count(distinct case when device in ('dell inspiron desktop') then user_id else null end) as "dell inspiron desktop",
count(distinct case when device in ('amazon fire phone') then user_id else null end) as "amazon fire phone",
count(distinct case when device in ('acer aspire desktop') then user_id else null end) as "acer aspire desktop",
count(distinct case when device in ('nokia lumia 635') then user_id else null end) as "nokia lumia 635",
count(distinct case when device in ('nexus 10') then user_id else null end) as "nexus 10"
from events
where event_type = 'engagement'
group by 1
order by 1;
```

Output-

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

Output-

	Week Numbers	samsumg galaxy tablet	acer aspire notebook	asus chromebook	samsung galaxy note	mac mini	hp pavilion desktop	ipad air	htc one	dell inspiron desktop	amazon fire phone	acer aspire desktop	nokia lumia 635	nexus 10
▶	17	8	20	21	7	6	14	27	16	18	4	9	17	16
	18	11	33	42	15	13	37	52	19	58	9	26	33	30
	19	6	41	27	11	18	40	55	30	36	12	23	23	25
	20	9	40	41	18	26	30	59	29	52	11	23	22	22
	21	6	47	38	20	18	44	51	21	41	5	29	25	25
	22	10	41	52	19	25	38	58	24	52	5	25	25	27
	23	14	43	49	14	18	54	41	20	53	16	22	31	45
	24	11	40	43	20	29	56	57	20	59	11	24	35	38
	25	12	47	38	14	21	52	57	21	52	13	28	37	29
	26	12	35	49	9	11	46	56	23	60	13	29	42	29
	27	15	49	52	15	15	56	55	27	53	10	29	31	37
	28	9	49	50	10	28	56	54	26	56	6	30	35	26
	29	13	53	49	16	31	58	52	31	54	12	28	43	25
	30	9	60	56	15	23	42	70	31	54	12	33	34	36
	31	8	55	56	14	24	51	55	13	44	14	31	28	24
	32	6	55	62	12	20	51	48	18	57	12	35	28	30
	33	12	46	49	13	32	38	40	19	37	14	39	27	23
	34	14	63	47	13	30	36	39	25	49	11	30	17	25
	35	0	3	6	1	2	1	0	2	1	0	1	2	2

Email Engagement Analysis:

Analyze how users are engaging with the email service. Write an SQL query to calculate the email engagement metrics.

Input-

```sql
SELECT Week,
ROUND((weekly_digest/total*100),2) AS "Weekly Digest Rate",
Round((email_opens/total*100),2) as "Email Open Rate",
Round((email_clickthroughs/total*100),2) as "Email Clickthrough Rate",
Round((Reengagement_emails/total*100),2) as "Reengagement Email Rate"
from
(
select extract(week from occurred_at) as Week,
count(case when action ='sent_weekly_digest' then user_id else null end) as weekly_digest,
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,
count(case when action ='sent_reengagement_email' then user_id else null end) as reengagement_emails,
count(user_id) as total
from email_events
group by 1
) sub
group by 1
Order by 1;
```

Output-

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

Result-

while doing this project, I learnt how to use subqueries in SQL tutorials which I applied it in multiple Problems. I have used case statements, I didn't knew how we can make Temporary columns, I applied it here. I had some doubts like I was doubtful about the data which was provided for Task 1 it was very liitle compared to previous project so I thought how I'll be able to to get the Monthly analytics on the basis of this much data then I watched the guide video in which sir had told to curated the data for task 1, so I curated it for the whole month.