REPORT

Operation Analytics and Investigating Metric Spike

Project Description:

In operation analytics and investigating metric spike project we have to analyse end to end operations of company that's called Operation Analytics. We have to derive some useful insights which will help the company to find out the key areas which it must improve. Using these analysis company can cut down costs and also it helps in improving decision making. In these project we have given some datasets and questions by different departments in company we have derive useful insights from that datasets and need to provide the results to departments so that they can take correct actions for companies growth.

For implementing these project I have used advanced SQL concepts which helps to perform complex operation on big datasets.

Approach:

First I convert the given data into database tables using converters. I have used advance SQL to find out the insights from big data sets. To perform the analysis and to find the information which will help the different departments in the company I have write SQL queries using different functions like window function, date and time function and so on.

Tech-Stack Used:

I used XAMP server as in these software we can easily view the structure of the stored databases. And also it provides graphical representation of stored data.

Version : v3.3.0

Insights:

While implementing these project I came to know where advance SQL's Command and functions such as window function, date and time function and nested queries. I got to learn various advance SQL's functions such as over(), extract() and so on.

Case Study 1 (Job Data) :

A) Number of jobs reviewed:

```
select ds, count(job_data.job_id) as no_of_jobs, (sum(job_data.time_spent)/3600) as jobs_r
eviewed from job_data where job_data.ds between '2020-11-01' And '2020-11-
30' GROUP BY ds;
```

Output:

ds	no_of_jobs	jobs_reviewed
2020-11-25	1	0.0125
2020-11-26	1	0.0156
2020-11-27	1	0.0289
2020-11-28	2	0.0092

2020-11-29	1	0.0052
2020-11-30	2	0.0111

B) Throughput:

select job_data.ds,count(job_data.job_id) as no_of_jobs,AVG(COUNT(job_data.job_id)) over(order by ds rows between 6 preceding and current row) as throughput from job_data GROUP by ds;

ds	no_of_jobs	throughput
2020-11-25	1	1.0000
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.2500
2020-11-29	1	1.2000
2020-11-30	2	1.3333

C) Percentage share of each language:

select job_data.language,COUNT(*) as language_count, count(*) * 100.0 / sum(count(*)) over() as percentageshare from job_data GROUP By job_data.LANGUAGE order by job_data.LANGUAGE;

language	language_count	percentageshare
Arabic	1	12.50000
English	1	12.50000
French	1	12.50000
Hindi	1	12.50000
Italian	1	12.50000
Persian	3	37.50000

D) Duplicate rows:

select * from (select *, row_number() over(PARTITION by job_data.job_id) as row_num from job_data) a where row_num > 1;

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

Case Study 2 (Investigating metric spike):

A) User Engagement:

select EXTRACT(week from events.occurred_at) as week, count(events.user_id) as weekly_user_engagement from events where events.event_type = 'engagement' And events.event_name='login' Group by 1 order by 1;

week	weekly_user_engagement
17	887
18	1985
19	2030
20	2093
21	1986
22	2157
23	2188
24	2265
25	2244
26	2266
27	2397
28	2493
29	2433
30	2583
31	2278
32	2098
33	2071
34	2052
35	104

B) User Growth:

select Month, user_count,
((user_count/LAG(user_count, 1) over (order by Month) - 1)*100) As Growth from (select extract(month from users.created_at) as Month, count(*) as user_count from users where users.activated_at IS NOT NULL group by 1 order by 1) a;

Month	user_count	Growth
1	712	NULL
2	685	-3.7921
3	765	11.6788
4	907	18.5621
5	993	9.4818
6	1086	9.3656
7	1281	17.9558
8	1347	5.1522
9	330	-75.5011
10	390	18.1818
11	399	2.3077
12	486	21.8045

C) Weekly Retention:

```
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) m, (select user_id,
MIN(EXTRACT(WEEK FROM occurred_at)) AS first FROM events GROUP BY 1) n WHERE
m.user_id = n.user_id ) z
Group By first order By first;
```

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

D) Weekly Engagement:

```
select extract(week from e.occurred_at) As week, COUNT(DISTINCT CASE WHEN
e.device IN("macbook pro")
THEN e.user_id ELSE NULL END) AS macbook_pro,COUNT(DISTINCT CASE WHEN e.device
IN("iphone 5")
THEN e.user_id ELSE NULL END) AS iphone_5,COUNT(DISTINCT CASE WHEN e.device
IN("samsung galaxy s4")
```

```
THEN e.user_id ELSE NULL END) AS samsung_galaxy_s4,
COUNT(DISTINCT CASE WHEN e.device IN("lenovo thinkpad")
THEN e.user_id ELSE NULL END) AS lenovo_thinkpad,COUNT(DISTINCT CASE WHEN
e.device IN("macbook air")
THEN e.user_id ELSE NULL END) AS macbook_air,COUNT(DISTINCT CASE WHEN e.device
IN("nexus 5")
THEN e.user_id ELSE NULL END) AS nexus_5,COUNT(DISTINCT CASE WHEN e.device
IN("iphone 5s")
THEN e.user_id ELSE NULL END) AS iphone_5s,
COUNT(DISTINCT CASE WHEN e.device IN("dell inspiron notebook")
THEN e.user_id ELSE NULL END) AS dell_inspiron_notebook,COUNT(DISTINCT CASE
WHEN e.device IN("asus chromebook")
THEN e.user_id ELSE NULL END) AS asus_chromebook,COUNT(DISTINCT CASE WHEN
e.device IN("dell inspiron desktop")
THEN e.user_id ELSE NULL END) AS dell_inspiron_desktop,COUNT(DISTINCT CASE WHEN
e.device IN("iphone 4s")
THEN e.user_id ELSE NULL END) AS iphone_4s,COUNT(DISTINCT CASE WHEN e.device
IN("nokia lumia 635")
THEN e.user_id ELSE NULL END) AS nokia_lumia_635,
COUNT(DISTINCT CASE WHEN e.device IN("acer aspire notebook")
THEN e.user_id ELSE NULL END) AS acer_aspire_notebook,COUNT(DISTINCT CASE WHEN
e.device IN("hp pavilion desktop")
THEN e.user_id ELSE NULL END) AS hp_pavilion_desktop,COUNT(DISTINCT CASE WHEN
e.device IN("htc one")
THEN e.user_id ELSE NULL END) AS htc_one,
COUNT(DISTINCT CASE WHEN e.device IN("acer aspire desktop")
THEN e.user_id ELSE NULL END) AS acer_aspire_desktop,COUNT(DISTINCT CASE WHEN
e.device IN("mac mini")
THEN e.user_id ELSE NULL END) AS mac_mini,COUNT(DISTINCT CASE WHEN e.device
IN("samsung galaxy note")
THEN e.user_id ELSE NULL END) AS samsung_galaxy_note,COUNT(DISTINCT CASE WHEN
e.device IN("amazon fire phone")
THEN e.user_id ELSE NULL END) AS amazon_fire_phone,COUNT(DISTINCT CASE WHEN
e.device IN("ipad air")
THEN e.user_id ELSE NULL END) AS ipad_air,COUNT(DISTINCT CASE WHEN e.device
IN("nexus 7")
THEN e.user_id ELSE NULL END) AS nexus_7,COUNT(DISTINCT CASE WHEN e.device
IN("ipad mini")
THEN e.user_id ELSE NULL END) AS ipad_mini,COUNT(DISTINCT CASE WHEN e.device
IN("nexus 10")
THEN e.user_id ELSE NULL END) AS nexus_10,COUNT(DISTINCT CASE WHEN e.device
IN("kindle fire")
THEN e.user_id ELSE NULL END) AS kindle_fire,COUNT(DISTINCT CASE WHEN e.device
IN("windows surface")
THEN e.user_id ELSE NULL END) AS windows_surface,COUNT(DISTINCT CASE WHEN
e.device IN("samsumg galaxy tablet")
THEN e.user_id ELSE NULL END) AS samsung_galaxy_tablet
from events e
```

where e.event_type = 'engagement'
group by 1
order by 1
limit 100;

week	macbook_pro	iphone_5	samsung_galaxy_s4	lenovo_thinkpad	macbook_air	nexus_5	iphone_5s	dell_inspiron_notebook	asus_chromebook	dell_inspiron_desktop	iphone_4s	nol
17	143	65	52	86	54	40	42	46	21	18	21	17
18	252	113	82	153	121	73	73	77	42	58	46	33
19	266	115	91	178	112	87	79	83	27	36	44	23
20	256	125	93	173	119	103	79	84	41	52	55	22
21	247	137	84	167	110	91	74	80	38	41	45	25
22	251	125	105	176	145	96	71	92	52	52	45	25
23	266	152	99	176	124	88	79	103	49	53	53	31
24	255	142	101	165	152	87	79	99	43	59	53	35
25	275	137	99	197	121	89	78	105	38	52	40	37
26	269	152	112	192	134	87	94	89	49	60	50	42
27	302	163	116	202	142	84	83	89	52	53	67	31
28	295	151	122	220	148	85	93	103	50	56	61	35
29	295	144	123	209	148	77	90	113	49	54	60	43
30	322	152	103	206	159	84	103	127	56	54	65	34
31	321	135	100	207	147	69	71	113	56	44	56	28
32	307	119	82	179	125	67	67	104	62	57	34	28
33	312	110	80	191	133	70	65	110	49	37	35	27
34	292	101	90	193	136	70	70	105	47	49	50	17
35	17	2	6	16	10	4	3	9	6	1	6	2

nokia_lumia_635	acer_aspire_notebook	hp_pavilion_desktop	htc_one	acer_aspire_desktop	mac_mini	samsung_galaxy_note	amazon_fire_phone	ipad_air	nexus_7	ipad_mini	nexus_10
17	20	14	16	9	6	7	4	27	18	19	16
33	33	37	19	26	13	15	9	52	30	30	30
23	41	40	30	23	18	11	12	55	41	36	25
22	40	30	29	23	26	18	11	59	32	32	22
25	47	44	21	29	18	20	5	51	29	23	25
25	41	38	24	25	25	19	5	58	45	34	27
31	43	54	20	22	18	14	16	41	36	33	45
35	40	56	20	24	29	20	11	57	49	39	38
37	47	52	21	28	21	14	13	57	51	30	29
42	35	46	23	29	11	9	13	56	46	43	29
31	49	56	27	29	15	15	10	55	40	35	37
35	49	56	26	30	28	10	6	54	39	35	26
43	53	58	31	28	31	16	12	52	45	34	25
34	60	42	31	33	23	15	12	70	62	35	36
28	55	51	13	31	24	14	14	55	38	27	24
28	55	51	18	35	20	12	12	48	25	30	30
27	46	38	19	39	32	13	14	40	30	28	23
17	63	36	25	30	30	13	11	39	33	25	25
Console	3	1	2	1	2	1	0	0	2	2	2

nexus_10	kindle_fire	windows_surface	samsung_galaxy_tablet
16	6	10	8
30	27	10	11
25	21	16	6
22	23	21	9
25	30	17	6
27	21	15	10
45	25	14	14
38	25	22	11
29	24	22	12
29	26	21	12
37	25	33	15
26	31	33	9
25	37	28	13
36	25	19	9
24	14	19	8
30	12	10	6
23	14	15	12
25	13	18	14
2	3	3	0

E) Email Engagement:

select week, ((weekly_emails/total)*100) AS Weekly_Digest_Rate,
((email_opens/total)*100) AS Email_Open_Rate, ((email_clickthroughs/total)*100) AS

```
Email_Clickthrough_Rate, ((reengagement_emails/total)*100) AS
Reengagement_Emails_Rate FROM (select extract(week from
email_events.occurred_at) AS week, COUNT(CASE WHEN email_events.action
IN("sent_weekly_digest") THEN email_events.user_id ELSE NULL END) AS
weekly_emails, COUNT(CASE WHEN email_events.action IN("email_open") THEN
email_events.user_id ELSE NULL END) AS email_opens, COUNT(CASE WHEN
email_events.action = "email_clickthrough" THEN email_events.user_id ELSE NULL END)
AS email_clickthroughs, COUNT(CASE WHEN email_events.action =
"sent_reengagement_email" THEN email_events.user_id ELSE NULL END) AS
reengagement_emails, COUNT(email_events.user_id) as total from email_events
GROUP BY 1) z GROUP BY 1 ORDER BY 1;
```

week	Weekly_Digest_Rate	Email_Open_Rate	Email_Clickthrough_Rate	Reengagement_Emails_Rate
17	62.3198	21.2766	11.3933	5.0103
18	63.4479	22.2385	10.4852	3.8283
19	62.1647	22.6732	11.1267	4.0355
20	61.6234	22.6381	11.4318	4.3067
21	63.5156	22.8224	9.9707	3.6912
22	63.5867	21.5596	10.6597	4.1940
23	62.3935	22.3353	11.1781	4.0931
24	61.6071	22.9167	10.9921	4.4841
25	63.7701	21.7936	10.5389	3.8974
26	62.9912	22.2243	10.6066	4.1778
27	62.2413	22.4867	11.3715	3.9004
28	62.9203	22.4780	10.7714	3.8302
29	63.9829	21.7136	10.5094	3.7941
30	62.2857	23.2437	10.5882	3.8824
31	65.2728	23.2490	7.6579	3.8203
32	66.5926	22.8469	7.1429	3.4176
33	64.7306	23.1042	7.9058	4.2594
34	64.3349	23.9124	7.6682	4.0845
35	0.0000	32.2835	29.9213	37.7953

Result:

From these project I understood the importance and use of operation analytics and how companies use metric spike to improve the growth of company. I got to learn many SQL functions and how to use them to get the results. The challenged that I faced during implementation of project is while converting big data to tables / insert commands. I used bulk insert command to resolve this issue.

Drive Link:

https://drive.google.com/file/d/1QVC2eCOS0-WR8_zPMfH0vRQqlQ7zAvM4/view?usp=sharing