Operation Analytics and Investigating Metric Spike
Advanced SQL

PROJECT DESCRIPTION:

In this project i.e. Operation Analytics and Investigating Metric Spike, I provided support in the effort to produce business insights for the marketing, product, and by giving the data to the development teams working on this project using SQL (structured query language). I am utilising a variety of SQL queries during this process to obtain the relevant data. Based on the data obtained, I managed to figure out-

- Number of jobs reviewed.
- Number of events happening per second.
- Percentage share of each language in the company.
- Duplicate rows.

As a Data Analyst, investigating metric spikes is a crucial component of operation analytics. We used different SQL queries to gather information on user engagement, growth, weekly retention, email engagement and more which helps the company to determine the areas which require improvement.

APPROACH:

I initially examined the objective to find the relevant data that the team needed and then I imported the data into SQL and performed multiple queries to analyse the data and find the team's insights in requirement for marketing, support and development.

TECH-STACK USED:

I carried out the project using **MySQL Workbench 8.0 CE** and **Mode.com**

INSIGHTS:

I identify and clarify the main objective and by using SQL, I learn more about real-time analytics and obtain insights by applying various SQL commands on this project. I identified the users' and employees' info that was provided. I gained various insights into the number of jobs, email engagement, user growth, retention, and weekly engagement. I also looked for duplicate rows, comprehended the SQL process and learned how to use practical SQL skills in any organisation.

RESULTS:

By the completion of the project, I had improved my SQL skills, learned how to work with real-world data and experienced how to approach and resolve problems.

Based on the data provided, I was able to achieve a number of results which are listed below.

A) Case Study 1 (Job Data):

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

Task: Calculate the number of jobs reviewed per hour per day for November 2020

Query:

```
SELECT
COUNT(distinct job_id)/(30*24) as Jobs_reviewed_per_hour
FROM operation_analytics.job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
```

Result:

Jobs_reviewed_per_hour
0.0083

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

Task: Calculate 7 day rolling average of throughput.

Query:

```
SELECT ds,
jobs_reviewed,
AVG(jobs_reviewed) OVER (ORDER BY ds ROWS BETWEEN 6
PRECEDING AND CURRENT ROW) AS throughput_7
FROM (SELECT ds, COUNT(DISTINCT job_id) AS
jobs_reviewed
FROM operation_analytics.job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds ORDER BY ds ) a
```

Result:

ds	jobs_reviewed	throughput_7
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

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

Task: Calculate the percentage share of each language in the last 30 days.

Query:

```
SELECT language, num_jobs,
100.0*num_jobs/total_jobs as pc_jobs
FROM
(SELECT language, COUNT(DISTINCT job_id) as num_jobs
FROM operation_analytics.job_data
GROUP BY language) a
CROSS JOIN
(SELECT COUNT(DISTINCT job_id) as total_jobs
FROM operation_analytics.job_data) b
```

Result:

language	num_jobs	pc_jobs
Arabic	1	16.66667
English	1	16.66667
French	1	16.66667
Hindi	1	16.66667
Italian	1	16.66667
Persian	1	16.66667

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

Task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table.

Query:

```
SELECT * FROM
(SELECT *, ROW_NUMBER()
OVER(PARTITION BY job_id) AS row_num
FROM operation_analytics.job_data ) a WHERE row_num>1
```

Result:

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

B) 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.

Task: Calculate the weekly user engagement.

Query:

```
SELECT EXTRACT(week from occurred_at) AS weeknum,
COUNT(DISTINCT user_id) FROM tutorial.yammer_events a
GROUP BY weeknum
```

Result:

weeknum	count
18	791
19	1244
20	1270
21	1341
22	1293
23	1366
24	1434
25	1462
26	1443

weeknum	count
27	1477
28	1556
29	1556
30	1593
31	1685
32	1483
33	1438
34	1412
35	1442

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

Task: Calculate the user growth for product.

Query:

```
SELECT year, weeknum, num_active_user,
SUM(num_active_user)
OVER(ORDER BY year, weeknum ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS
cum_active_user FROM
(SELECT EXTRACT(year from a.activated_at) AS year,
EXTRACT(week from a.activated_at)AS weeknum,
COUNT(DISTINCT user_id) AS num_active_user FROM
tutorial.yammer_users a
WHERE state='active'
GROUP BY year, weeknum
ORDER BY year, weeknum) b
```

Result:

year	weeknum	num_active_user	cum_active_user
2013	1	67	67
2013	2	29	96
2013	3	47	143
2013	4	36	179
2013	5	30	209
2013	6	48	257
2013	7	41	298
2013	8	39	337
2013	9	33	370
2013	10	43	413
2013	11	33	446
2013	12	32	478
2013	13	33	511
2013	14	40	551
2013	15	35	586
2013	16	42	628
2013	17	48	676
2013	18	48	724
2013	19	45	769
2013	20	55	824
2013	21	41	865
2013	22	49	914
2013	23	51	965
2013	24	51	1016
2013	25	46	1062
2013	26	57	1119
2013	27	57	1176
2013	28	52	1228
2013	29	71	1299
2013	30	66	1365
2013	31	69	1434
2013	32	66	1500
2013	33	73	1573
2013	34	70	1643
2013	35	80	1723
2013	36	65	1788
2013	37	71	1859
2013	38	84	1943
2013	39	92	2035
2013	40	81	2116
2013	41	88	2204
2013	42	74	2278

year	weeknum	num_active_user	cum_active_user
2013	43	97	2375
2013	44	92	2467
2013	45	97	2564
2013	46	94	2658
2013	47	82	2740
2013	48	103	2843
2013	49	96	2939
2013	50	117	3056
2013	51	123	3179
2013	52	104	3283
2014	1	91	3374
2014	2	122	3496
2014	3	112	3608
2014	4	113	3721
2014	5	130	3851
2014	6	132	3983
2014	7	135	4118
2014	8	127	4245
2014	9	127	4372
2014	10	135	4507
2014	11	152	4659

year	weeknum	num_active_user	cum_active_user
2014	12	132	4791
2014	13	151	4942
2014	14	161	5103
2014	15	166	5269
2014	16	165	5434
2014	17	176	5610
2014	18	172	5782
2014	19	160	5942
2014	20	186	6128
2014	21	177	6305
2014	22	186	6491
2014	23	197	6688
2014	24	198	6886
2014	25	222	7108
2014	26	210	7318
2014	27	199	7517
2014	28	223	7740
2014	29	215	7955
2014	30	228	8183
2014	31	234	8417
2014	32	189	8606
2014	33	250	8856
2014	34	259	9115
2014	35	266	9381

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

Task: Calculate the weekly retention of users-sign up cohort?

Query:

```
SELECT COUNT(user_id),
SUM(CASE WHEN retention_week = 1 THEN 1
ELSE 0
END) AS week_1 FROM
(SELECT a.user_id, a.signup_week, b.engagement_week,
b.engagement_week - a.signup_week AS retention_week
FROM((SELECT DISTINCT user_id,
EXTRACT(WEEK FROM occurred_at) AS signup_week
FROM tutorial.yammer_events
WHERE event_type = 'signup_flow' AND
event_name = 'complete_signup' AND
EXTRACT(WEEK FROM occurred_at) = 18) a
LEFT JOIN (SELECT DISTINCT user_id,
```

```
EXTRACT(WEEK FROM occurred_at) AS engagement_week
FROM tutorial.yammer_events
WHERE event_type = 'engagement') b
ON a.user_id = b.user_id)
ORDER BY a.user_id) a
```

Result:

count	week_1
317	64

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

Task: Calculate the weekly engagement per device?

Query:

```
SELECT EXTRACT(year FROM occurred_at) AS year,
EXTRACT(week from occurred_at) AS week,
device, COUNT(distinct user_id) FROM
tutorial.yammer_events
WHERE event_type ='engagement'
GROUP BY 1,2,3
ORDER by 1,2,3
```

year	week	device	count
2014	18	acer aspire desktop	10
2014	18	acer aspire notebook	21
2014	18	amazon fire phone	4
2014	18	asus chromebook	23
2014	18	dell inspiron desktop	21
2014	18	dell inspiron notebook	49
2014	18	hp pavilion desktop	15
2014	18	htc one	16
2014	18	ipad air	30
2014	18	ipad mini	21
2014	18	iphone 4s	21
2014	18	iphone 5	70
2014	18	iphone 5s	45
2014	18	kindle fire	6
2014	18	lenovo thinkpad	90
2014	18	macbook air	57
2014	18	macbook pro	154
2014	18	mac mini	8
2014	18	nexus 10	16
2014	18	nexus 5	43

year	week	device	count
2014	18	nexus 7	20
2014	18	nokia lumia 635	19
2014	18	samsumg galaxy tablet	8
2014	18	samsung galaxy note	7
2014	18	samsung galaxy s4	56
2014	18	windows surface	10
2014	19	acer aspire desktop	26
2014	19	acer aspire notebook	34
2014	19	amazon fire phone	9
2014	19	asus chromebook	42
2014	19	dell inspiron desktop	58
2014	19	dell inspiron notebook	78
2014	19	hp pavilion desktop	37
2014	19	htc one	19
2014	19	ipad air	52
2014	19	ipad mini	29
2014	19	iphone 4s	47
2014	19	iphone 5	114
2014	19	iphone 5s	70
2014	19	kindle fire	26

year	week	device	count
2014	19	lenovo thinkpad	155
2014	19	macbook air	119
2014	19	macbook pro	248
2014	19	mac mini	12
2014	19	nexus 10	30
2014	19	nexus 5	73
2014	19	nexus 7	29
2014	19	nokia lumia 635	34
2014	19	samsumg galaxy tablet	11
2014	19	samsung galaxy note	15
2014	19	samsung galaxy s4	80
2014	19	windows surface	10
2014	20	acer aspire desktop	22
2014	20	acer aspire notebook	40
2014	20	amazon fire phone	12
2014	20	asus chromebook	26
2014	20	dell inspiron desktop	36
2014	20	dell inspiron notebook	82
2014	20	hp pavilion desktop	40
2014	20	htc one	32
2014	20	ipad air	53
2014	20	ipad mini	37
2014	20	iphone 4s	40
2014	20	iphone 5	113
2014	20	iphone 5s	77
2014	20	kindle fire	20
2014	20	lenovo thinkpad	176
2014	20	macbook air	110
2014	20	macbook pro	261
2014	20	mac mini	19
2014	20	nexus 10	25
2014	20	nexus 5	84
2014	20	nexus 7	41
2014	20	nokia lumia 635	22
2014	20	samsumg galaxy tablet	6
2014	20	samsung galaxy note	11
2014	20	samsung galaxy s4	90
2014	20	windows surface	15
2014	21	acer aspire desktop	23
2014	21	acer aspire notebook	40
2014	21	amazon fire phone	10
2014	21	asus chromebook	39
2014	21	dell inspiron desktop	52
2014	21	dell inspiron notebook	84
2014	21	hp pavilion desktop	31
2014	21	htc one	27
2014	21	ipad air	54
2014	21	ipad mini	32
2014	21	iphone 4s	56
2014	21	iphone 5	128
2014	21	iphone 5s	75
2014	21	kindle fire	22
2014	21	lenovo thinkpad	177
2014	21	macbook air	119
2014	21	macbook pro	256
2014	21	mac mini	25
2014	21	nexus 10	23
2014	21	nexus 5	99
2014	21	nexus 7	31
2014	21	nokia lumia 635	21

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

Task: Calculate the email engagement metrics?

Query:

```
SELECT 100.0 * SUM(CASE
WHEN email_cat = 'email_open' 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_clicked_rate
FROM(SELECT *, CASE
WHEN action IN ('sent_weekly_digest' ,
'sent_reengagement_email') THEN 'email_sent'
WHEN action IN ('email_open') THEN 'email_open'
WHEN action IN ('email_clickthrough') THEN 'email_clicked'
END AS email_cat
FROM tutorial.yammer_emails) a
```

Results:

email_open_rate	email_clicked_rate
33.5834	14.7899