PROJECT- OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

- Advanced SQL

Project Description:

This project is about Operation Analytics and Investigating Metric Spike. Operation Analytics is the analysis done for the complete end to end operations of a company. This helps the company finds the areas on which it needs improvements. You work cooperating with the operations team, support team, marketing team, etc and aid them derive insights from the data they collect. Operation analytics is an important process and is further used to predict the overall growth or decline of a company's wealth. It implies better automation, better understanding between cross-functional teams, and more effective workflows. Investigating metric spike is also an essential part of operation analytics since a data analyst must be able to understand or make other teams understand questions. And by answering these questions, important insights can be derived.

Through this project, I will be finding out the number of jobs reviewed, number of events happening per second, percentage share of languages used, duplicate rows in the data, weekly user engagement and user growth, weekly retention of users, email engagement, etc. and many other insights from the provided data, as asked by the team.

Approach:

I tried to understand the purpose of this project. I read each and every tasks provided by the team and understood what data they needed from it. SQL is used to perform the analysis. The datasets for case study 1 and case study 2 were provided by the team and it contain the details for creating the database and tables for doing operation analytics and for investigating the metric spike. It contains the details of job data, user data, languages spoken by users, events, email events, etc. For case study 1, I imported the data to the DB Fiddle into the Schema SQL section and executed it. For case study 2, I solved the questions in the Mode.com, which is a SQL editor website. I analyzed the data completely and started querying. I performed several SQL commands and gained insights from the results got, which the operations team, support team and marketing team needed.

Tech-Stack Used:

I used DB Fiddle for solving Case Study 1 questions and used Mode.com for solving Case Study 2 questions. They are useful and free online SQL editor websites to learn and practice SQL coding for a wider range of databases. It also generates results quickly.

Insights:

While doing the Operation analytics and investigating metric spike project, I performed several SQL queries as per the given tasks, gained many insights and knowledge, and provided a detailed report.

After getting the desired output data, I understood about the number of jobs reviewed, throughput, i.e., the number of events happening per second, percentage share of languages spoken by the users, number of duplicate rows in the data, actions of users, time spent for events or actions done by the users, measure of activeness of users, weekly user engagement, user growth for product, weekly retention of users after sign up for a product, weekly engagement of users per device, email engagement, user behaviour and interaction, etc. By analyzing this output data, we can analyse the reasons for dip in sales and daily engagement of users , if any and it helps to predict the overall growth or decline of company's fortune. It can help us understand the areas where the company needs improvement. These insights helps in doing predictive maintenance and implement strategies to find opportunities and to build better customer experiences. It can help in bringing better user experience, better automation, better collaboration between cross-functional teams, more effective optimized data workflows and making better business decisions. So that the complete end to end operations of the company can be done efficiently and smoothly. It helps in boosting the company's growth.

Result:

While making this project, I have achieved the confidence to carry out the tasks provided by the team and got the required output. It has helped me to derive useful insights for the team, from the output data. This project has helped me to gain a clear understanding about advanced SQL and helped me to use my imagination to improve my practical knowledge in it. I learned to apply real time SQL knowledge in solving the tasks of this project.

Below are each of the tasks given by the teams, the SQL queries used to carry out the tasks and their respective outputs.

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)
FROM
      job_data
WHERE
      ds BETWEEN '2020-11-01' AND '2020-11-30'
```

Output:

COUNT(DISTINCT job_id)/(30*24)
0.0111

2. **Throughput:** It is the no. of events happening per second.
   Task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

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 job_data
 WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
 GROUP BY ds
 ORDER BY ds
) a
```

Output:

ds	jobs_reviewed	throughput_7
2020-11-25T00:00:00.000Z	1	1.0000000000000000000
2020-11-26T00:00:00.000Z	1	1.0000000000000000000
2020-11-27T00:00:00.000Z	1	1.0000000000000000000
2020-11-28T00:00:00.000Z	2	1.2500000000000000
2020-11-29T00:00:00.000Z	1	1.2000000000000000
2020-11-30T00:00:00.000Z	2	1.3333333333333333

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 pct_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
```

Output:

language	num_jobs	pct_jobs
Arabic	1	12.5000000000000000
English	1	12.5000000000000000
French	1	12.5000000000000000
Hindi	1	12.5000000000000000
Italian	1	12.5000000000000000
Persian	3	37.5000000000000000

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 rownum
  FROM job_data
) a
WHERE rownum>1
```

Output:

job_id	actors_id	event	language	time_spent	org	ds	rownum
28	1008	transfer	Italian	45	C	2020-11-25T00:00:00.000Z	2

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:

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

Output:

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

```sql
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_users
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
) a
```

Output:

year	weeknum	num_active_user	cum_active_users
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
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
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:

```sql
SELECT
      COUNT(user_id) AS Users_Signup_Cohort,
      SUM(CASE WHEN retention_week = 1 THEN 1 ELSE 0 END) AS Retained_on_week_1,
      SUM(CASE WHEN retention_week = 2 THEN 1 ELSE 0 END) AS Retained_on_week_2,
      SUM(CASE WHEN retention_week = 3 THEN 1 ELSE 0 END) AS Retained_on_week_3,
      SUM(CASE WHEN retention_week = 4 THEN 1 ELSE 0 END) AS Retained_on_week_4
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
```

Output:

users_signup_cohort	retained_on_week_1	retained_on_week_2	retained_on_week_3	retained_on_week_4
317	64	27	19	15

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:

```sql
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
```

Output:

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
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
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
```

Output:

email_open_rate	email_clicked_rate
33.5834	14.7899