Operation Analytics and Investigating Metric Spike

Project Description

This project focuses on the operational analytics of a product-based company as well as the company's metric spikes. The project is separated into two parts: the first is about obtaining insights about the company's operation analytics, and the second is about investigating metric spikes.

Operation Analytics is the analysis of a company's whole end-to-end activities. The company can then use this to identify the areas where it needs to improve.

This type of study is also used to forecast a company's overall growth or decline. It means improved automation, better communication across cross-functional teams, and more efficient workflows.

Investigating metric spikes is also an important element of operation analytics, and by doing so, the organization can learn more about their product, such as why there is a decline in daily interaction. Why have sales dropped? Etc. Such questions must be answered on a regular basis, and it is critical to study metric spike.

I'm going to gather the following information for Case Study 1 (Job Data):

A. Estimated number of jobs assessed on a daily hourly basis for the month of November, 2020.
B. Calculating the rolling 7-day average of throughput.
C. Calculating each language's percentage share in November 2020.
D. Displaying duplicate rows from the table.

And for Case Study 2 (Investigating Metric Spikes), I'll gather the following information:

A. Analyzing a user's weekly activity/engagement.
B. Calculating the growth of a product's user base over time.
C. Measuring the number of users who are retained per week after joining up.
D. Calculating a user's weekly engagement per device.
E. Estimating metrics for people who interact with email services.

Approach

First, I examined the data set, including all of the tables and columns, to gain a sense of the information accessible to me. Then I examined and comprehended all of the case studies and summarized my method to solving these case studies. Then I compose the query.

Tech-Stack Used

I selected MySQL workbench 8.0 v8.0.31 software for Case Study 1 as the data set was small for this case study and also the software is quite user friendly and I had previously used it so I was comfortable with the interface.

And for Case Study 2, because the data set was so large, it was nearly difficult to import it into the MySQL database. As a result, I used mode.com's online dataset and compiler; the website has a variety of test data sets available on its server.

The following datasets are employed on the website:

For

- table1 users: tutorial.yammer_users
- table2 events: tutorial.yammer_events
- table3 email_events: tutorial.yammer_emails

Insights

I learned how data analysts analyze a company's end-to-end activities. Working with the operations team, support team, and marketing team, analysts can derive numerous helpful insights from the collected data, allowing them to forecast the general development or decline of a company.

I also learned about metric spikes, which can assist data analysts and teams in understanding how the company's product is functioning. How people are engaging with the product, why sales are down, and so on.

Result

By completing this project, I was able to solidify my understanding of numerous SQL clauses and statements such as GROUP BY, ORDER BY, JOINS, COUNT, various subqueries, EXTRACT function, CASE WHEN clause. It assists me in honing my SQL skills.

CASE STUDY 1: Job Data

A. **Number of jobs reviewed:** Amount of jobs reviewed over time.
Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

```
SELECT COUNT(DISTINCT job_id)/(24*30) AS jobs_rev

FROM job_data

WHERE

        EXTRACT(MONTH FROM ds) = 11;
```

jobs_rev
0.0083

B. **Throughput:** It is the no. of events happening per second.
Your 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?

```
SELECT ds, jobs_rev,
                AVG(jobs_rev)
                OVER(ORDER BY ds
                        ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS
rolling_avg_7d
FROM
(
SELECT ds, COUNT(DISTINCT job_id) AS jobs_rev
FROM job_data
WHERE
        EXTRACT(MONTH FROM ds) = 11
GROUP BY ds) AS sub;
```

ds	jobs_rev	rolling_avg_7d
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:** Share of each language for different contents.
Your task: Calculate the percentage share of each language in the last 30 days?

```
SELECT language,
        COUNT(*)/
        (SELECT COUNT(DISTINCT language) AS lang_count
            FROM job_data)*100 AS prct_share_lang
FROM job_data
GROUP BY language
ORDER BY language;
```

language	prct_share_lang
Arabic	16.6667
English	16.6667
French	16.6667
Hindi	16.6667
Italian	16.6667
Persian	50.0000

D. **Duplicate rows:** Rows that have the same value present in them.
Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

```
SELECT ds, job_id, actor_id, event, language, time_spent, org

FROM (

                SELECT *,

                ROW_NUMBER()OVER(PARTITION BY job_id) AS rptd_row

                FROM job_data

                ORDER BY job_id ) AS a

WHERE rptd_row > 1;
```

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

CASE STUDY 2: Investigating Metric Spikes

A. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
Your task: Calculate the weekly user engagement?

```
SELECT COUNT(DISTINCT user_id) AS no_of_users, week
FROM(
    SELECT user_id, EXTRACT(week FROM occurred_at) AS week
    FROM tutorial.yammer_events
    WHERE event_type = 'engagement') AS a
GROUP BY week;
```

	no_of_users	week
1	701	18
2	1054	19
3	1094	20
4	1147	21
5	1113	22
6	1173	23
7	1219	24
8	1263	25
9	1249	26
10	1271	27
11	1355	28
12	1345	29
13	1363	30
14	1443	31
15	1266	32
16	1215	33
17	1203	34
18	1194	35

B. **User Growth:** Amount of users growing over time for a product.
Your task: Calculate the user growth for product?

```
SELECT year, week, users_active,
    SUM(users_active) OVER(ORDER BY year, week
                           ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS
users_growth
FROM(
    SELECT
       EXTRACT(year FROM activated_at) AS year,
       EXTRACT(week FROM activated_at) AS week,
       COUNT(DISTINCT user_id) AS users_active
    FROM tutorial.yammer_users
    WHERE state = 'active'
    GROUP BY week, year
    ORDER BY year, week
    ) AS d;
```

Data Fields Source

	year	week	users_active	users_growth
40	2013	40	81	2116
41	2013	41	88	2204
42	2013	42	74	2278
43	2013	43	97	2375
44	2013	44	92	2467
45	2013	45	97	2564
46	2013	46	94	2658
47	2013	47	82	2740
48	2013	48	103	2843
49	2013	49	96	2939
50	2013	50	117	3056
51	2013	51	123	3179
52	2013	52	104	3283
53	2014	1	91	3374
54	2014	2	122	3496
55	2014	3	112	3608
56	2014	4	113	3721
57	2014	5	130	3851
58	2014	6	132	3983
59	2014	7	135	4118
60	2014	8	127	4245
61	2014	9	127	4372

C. **Weekly Retention:** Users getting retained weekly after signing-up for a product.
Your task: Calculate the weekly retention of users-sign up cohort?

```
SELECT user_id,
SUM(CASE WHEN retained_week >= 1 THEN 1 ELSE 0 END) AS weekly_retention
FROM
        (SELECT
            c.user_id,
            c.signup_week,
            e.engagement_week,
            e.engagement_week - c.signup_week AS retained_week
        FROM
            (SELECT
            user_id,
            EXTRACT(week FROM occurred_at) AS signup_week
            FROM tutorial.yammer_events
            WHERE event_name = 'complete_signup') AS c
        INNER JOIN
            (SELECT
            DISTINCT user_id,
            EXTRACT(week FROM occurred_at) AS engagement_week
            FROM tutorial.yammer_events
            WHERE event_type = 'engagement'
            ORDER BY 2) AS e
        USING (user_id)
        ORDER BY 1,2,3) AS g
GROUP BY 1;
```

	user_id	weekly_retention
1	11768	0
2	11770	0
3	11775	1
4	11778	2
5	11779	4
6	11780	1
7	11785	0
8	11787	2
9	11791	1
10	11793	5
11	11795	1
12	11798	5
13	11799	9
14	11801	1
15	11804	1
16	11806	0
17	11809	0
18	11811	1
19	11813	5
20	11816	2
21	11818	1
22	11820	3
23	11823	2
24	11824	6
25	11825	2
26	11826	1
27	11828	2
28	11829	2
29	11832	3
30	11833	13
31	11834	1

D. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
Your task: Calculate the weekly engagement per device?

```
SELECT
    EXTRACT(year FROM occurred_at) AS year,
    EXTRACT(week FROM occurred_at) AS week,
    COUNT(DISTINCT user_id) AS users_engage,
    device
FROM tutorial.yammer_events
GROUP BY 1,2,4
ORDER BY 1,2,3;
```

	year	week	users_engage	device
10	2014	18	19	htc one
11	2014	18	20	nokia lumia 635
12	2014	18	21	ipad mini
13	2014	18	22	nexus 7
14	2014	18	23	dell inspiron desktop
15	2014	18	25	acer aspire notebook
16	2014	18	26	asus chromebook
17	2014	18	27	iphone 4s
18	2014	18	33	ipad air
19	2014	18	50	iphone 5s
20	2014	18	50	nexus 5
21	2014	18	51	dell inspiron notebook
22	2014	18	62	samsung galaxy s4
23	2014	18	65	macbook air
24	2014	18	74	iphone 5
25	2014	18	100	lenovo thinkpad
26	2014	18	167	macbook pro
27	2014	19	9	amazon fire phone
28	2014	19	12	windows surface
29	2014	19	14	mac mini
30	2014	19	15	samsumg galaxy tablet
31	2014	19	18	samsung galaxy note
32	2014	19	20	htc one

« ‹ Page 1 of 1 › » Showing rows 1-100 of 100

E. **Email Engagement:** Users engaging with the email service.
Your task: Calculate the email engagement metrics?

```
SELECT
    ROUND(100.0*SUM(CASE WHEN email_events = 'email_opened' THEN 1 END)/SUM(CASE
WHEN email_events = 'email_delivered' THEN 1 END),2) AS open_rate,
    ROUND(100.0*SUM(CASE WHEN email_events = 'email_clicked' THEN 1 END)/SUM(CASE
WHEN email_events = 'email_delivered' THEN 1 END),2) AS click_rate
FROM
    (SELECT
    CASE WHEN action = 'email_open'
        THEN 'email_opened'
       WHEN action = 'email_clickthrough'
        THEN 'email_clicked'
       WHEN action IN ('sent_weekly_digest', 'sent_reengagement_email')
        THEN 'email_delivered'
    END AS email_events
    FROM tutorial.yammer_emails) AS e;
```

	open_rate	click_rate
1	33.58	14.79