OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

By: Abhishek Kumar

CONTENTS

PROJECT DESCRIPTION

With the help of operation analytics, the company can find the areas on which it must improve upon. Being one of the most important parts of a company, this kind of analysis is further used to understanding between cross-functional teams, and more effective workflows. Investigating metric spike is also an important part of operation analytics as being a Data Analyst we must be able to understand or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily and for that it's very important to investigate metric spike.

Working for a company like Microsoft designated as Data Analyst Lead and is provided with different data sets, tables from which I must derive certain insights out of it and answer the questions asked by different departments.

APPROACH

Database creation : Created and inserted the values in the database using the DDL & DML SQL queries provided by the product manager(as per project) in the MySQL database using MySQL workbench.

Extraction of insights : After creating the database required insights are generated from the database tables by running SQL queries in MySQL workbench.

TECH-STACK USED

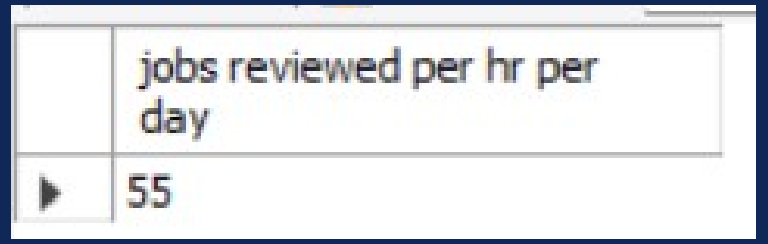
Used MySQL Community Server - GPL Version 8.0.36.0 and Connector Version C++ 8.0.36.0 for creating my project as MySQL Community Server - GPL is a free and open-source relational database management system that uses SQL.

INSIGHTS

Case Study 1 (Operation Analytics)

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

```
WITH jobs_review_per_hr
     AS (SELECT ds AS "review date",
     Round(Count(job_id) / Sum(time_spent) * 3600) AS "review_pe
r_hr"
     FROM   job_data
     WHERE  Month(ds) = 11
     GROUP  BY ds)
SELECT Round(Sum(review_per_hr) / 30) AS "jobs reviewed per hr p
er day"
FROM   jobs_review_per_hr;
```



	jobs reviewed per hr per day
▶	55

INSIGHTS

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?

```
WITH jobs_review_per_sec
     AS (SELECT ds AS "review_date",
                Round(Count(job_id) / SUM(time_spent), 3) AS
                "jobs_review_per_sec"
         FROM   job_data
         WHERE  Month(ds) = 11
         GROUP  BY ds)
SELECT *,
       Round(Avg(jobs_review_per_sec)
       over(ORDER BY review_date ROWS BETWEEN 6 preceding
       AND CURRENT ROW),3) AS
       "7-days rolling average"
FROM   jobs_review_per_sec;
```

review_date	jobs_review_per_sec	7-days rolling average
2020-11-01	0.019	0.019
2020-11-02	0.012	0.016
2020-11-03	0.019	0.017
2020-11-04	0.017	0.017
2020-11-05	0.012	0.016
2020-11-06	0.020	0.017
2020-11-07	0.013	0.016
2020-11-08	0.012	0.015
2020-11-09	0.013	0.015
2020-11-10	0.014	0.014
2020-11-11	0.015	0.014
2020-11-12	0.021	0.015
2020-11-13	0.014	0.015
2020-11-14	0.012	0.014
2020-11-15	0.013	0.015
2020-11-16	0.019	0.015
2020-11-17	0.014	0.015
2020-11-18	0.017	0.016

INSIGHTS

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?

```
SELECT language,
       Round(Count(*) / Sum(Count(*))
       OVER() * 100, 2) AS "percentage share(%)"
FROM   job_data
WHERE  Month(ds) = 11
GROUP  BY language
ORDER  BY Round(Count(*) / Sum(Count(*))
          OVER() * 100, 2) DESC;
```

	language	percentage share(%)
▶	French	17.39
	English	17.10
	Arabic	17.10
	Hindi	16.52
	Persian	16.52
	Italian	15.36

INSIGHTS

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?

```
SELECT *,
       Count(*) AS "count_of_dulplicate_rows"
FROM   job_data
GROUP  BY ds,
          job_id,
          actor_id,
          event,
          language,
          time_spent,
          org
HAVING Count(*) > 1;
```

ds	job_id	actor_id	event	language	time_spent	org	count_of_dulplicate_rows
2020-10-16	8	1032	decision	Hindi	80	C	2
2020-10-19	23	1006	decision	Italian	84	A	2
2020-10-20	14	1026	transfer	French	94	B	2
2020-10-21	14	1015	skip	Persian	85	A	3
2020-10-23	6	1047	skip	English	81	D	3
2020-10-24	11	1018	transfer	Italian	81	B	2
2020-10-25	22	1021	decision	Italian	74	D	2
2020-10-25	23	1014	decision	English	56	C	3
2020-10-25	20	1029	transfer	Hindi	75	D	2
2020-10-26	12	1024	decision	Hindi	112	D	2
2020-10-26	18	1021	skip	Persian	107	B	2
2020-10-30	5	1042	transfer	Hindi	96	A	4
2020-10-30	19	1026	transfer	Persian	60	C	3
2020-11-01	21	1017	skip	English	44	C	2
2020-11-03	5	1034	transfer	Hindi	34	D	2
2020-11-04	24	1042	skip	Hindi	44	A	2
2020-11-05	22	1050	decision	Hindi	119	A	2
2020-11-08	10	1046	decision	French	111	C	2
2020-11-10	7	1050	skip	Arabic	104	B	3
2020-11-10	14	1046	decision	Hindi	63	D	2
2020-11-10	9	1019	decision	Italian	62	D	3
2020-11-14	19	1043	decision	Persian	108	D	3
2020-11-15	15	1022	skip	Arabic	79	B	3
2020-11-15	4	1014	skip	Hindi	108	B	2
2020-11-17	24	1024	decision	Arabic	104	B	3

INSIGHTS

Case Study 2 (Investigating Metric Spike)

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 Concat("week-", Week(occurred_at), ”
”, Year(occurred_at)) AS
   "week number",
   Count(user_id)  AS
   "count of weekly engagement"
FROM   events_table
WHERE  event_type = "engagement"
GROUP  BY 1
ORDER  BY 1;
```

week number	count of weekly engagement
week-17 2014	8019
week-18 2014	17341
week-19 2014	17224
week-20 2014	17911
week-21 2014	17151
week-22 2014	18413
week-23 2014	18280
week-24 2014	19052
week-25 2014	18642
week-26 2014	19061
week-27 2014	19881
week-28 2014	20776
week-29 2014	20067
week-30 2014	21533
week-31 2014	18556
week-32 2014	16612
week-33 2014	16145
week-34 2014	16127
week-35 2014	784

INSIGHTS

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

```
WITH new_active_users
     AS (SELECT Date_format(activated_at, "%m %y") AS "Months",
                Count(user_id)                     AS "New_users"
         FROM   users
         WHERE  state = "active"
         GROUP  BY 1)
SELECT *,
       Round(( new_users / Lag(new_users, 1, 160)
                             OVER(ORDER BY "months") - 1 ) * 100, 2)
                             AS "% growth rate"
FROM   new_active_users
```

Months	New_users	% growth rate
January 2013	160	0.00
February 2013	160	0.00
March 2013	150	-6.25
April 2013	181	20.67
May 2013	214	18.23
June 2013	213	-0.47
July 2013	284	33.33
August 2013	316	11.27
September 2013	330	4.43
October 2013	390	18.18
November 2013	399	2.31
December 2013	486	21.80
January 2014	552	13.58
February 2014	525	-4.89
March 2014	615	17.14
April 2014	726	18.05
May 2014	779	7.30
June 2014	873	12.07
July 2014	997	14.20
August 2014	1031	3.41

INSIGHTS

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

```
WITH signup_users
     AS (SELECT user_id,
                Week(occurred_at) AS "sign_up_week"
                      FROM   events_table
                      WHERE  event_type = "signup_flow"
                      AND event_name = "complete_signup"),
     engagement_users
     AS (SELECT user_id,
                Week(occurred_at) AS "first_engagement_week"
                      FROM   events_table
                      WHERE  event_type = "engagement"),
     cal_table
     AS (SELECT e.user_id,
                e.first_engagement_week              AS "week_num",
                first_engagement_week - sign_up_week AS "retention_week"
                       FROM   engagement_users e
                       LEFT JOIN signup_users s
                       ON e.user_id = s.user_id)
SELECT week_num,
       Sum(CASE
             WHEN retention_week = 1 THEN 1
             ELSE 0
           END) AS "No. of users weekly retain"
FROM   cal_table
GROUP  BY week_num
ORDER  BY week_num;
```

	week_num	No. of users weekly retain
▶	17	168
	18	365
	19	364
	20	377
	21	362
	23	387
	22	389
	24	404
	25	394
	29	425
	26	403
	30	456
	28	438
	27	420
	31	392
	32	357
	33	349
	34	349

INSIGHTS

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 Concat("week-
", Week(occurred_at), "", Year(occurred_at)) AS
       "week number",
       device,
       Count(user_id)  AS
       "count of weekly engagement"
FROM   events_table
WHERE  event_type = "engagement"
GROUP  BY 1,
          2
ORDER  BY 1,
          2;
```

week number	device	count of weekly engagement
week-17 2014	acer aspire desktop	67
week-17 2014	acer aspire notebook	206
week-17 2014	amazon fire phone	83
week-17 2014	asus chromebook	251
week-17 2014	dell inspiron desktop	187
week-17 2014	dell inspiron notebook	503
week-17 2014	hp pavilion desktop	132
week-17 2014	htc one	190
week-17 2014	ipad air	330
week-17 2014	ipad mini	205
week-17 2014	iphone 4s	217
week-17 2014	iphone 5	706
week-17 2014	iphone 5s	473
week-17 2014	kindle fire	57
week-17 2014	lenovo thinkpad	793
week-17 2014	mac mini	59
week-17 2014	macbook air	490
week-17 2014	macbook pro	1516
week-17 2014	nexus 10	145
week-17 2014	nexus 5	382
week-17 2014	nexus 7	177
week-17 2014	nokia lumia 635	128
week-17 2014	samsumg galaxy tablet	70
week-17 2014	samsung galaxy note	116

INSIGHTS

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

```
WITH email_metric
     AS (SELECT Week(occurred_at) AS week_num,
                CASE
                  WHEN action = 'sent_weekly_digest' THEN 1
                  ELSE 0
                END                AS "sent_weekly_digest",
                CASE
                  WHEN action = 'email_open' THEN 1
                  ELSE 0
                END                AS "email_open",
                CASE
                  WHEN action = 'email_clickthrough' THEN 1
                  ELSE 0
                END                AS "email_clickthrough",
                CASE
                  WHEN action = 'sent_reengagement_email' THEN 1
                  ELSE 0
                END                AS "sent_reengagement_email"
         FROM   email_events)
SELECT week_num,
       Avg(sent_weekly_digest)      AS "sent_weekly_digest Rate",
       Avg(email_open)              AS "email_open Rate",
       Avg(email_clickthrough)      AS "email_clickthrough Rate",
       Avg(sent_reengagement_email) AS "sent_reengagement_email Rate"
FROM   email_metric
GROUP  BY week_num
ORDER  BY week_num;
```

week_num	sent_weekly_digest Rate	email_open Rate	email_clickthrough Rate	sent_reengagement_email Rate
17	0.6232	0.2128	0.1139	0.0501
18	0.6345	0.2224	0.1049	0.0383
19	0.6216	0.2267	0.1113	0.0404
20	0.6162	0.2264	0.1143	0.0431
21	0.6352	0.2282	0.0997	0.0369
22	0.6359	0.2156	0.1066	0.0419
23	0.6239	0.2234	0.1118	0.0409
24	0.6161	0.2292	0.1099	0.0448
25	0.6377	0.2179	0.1054	0.0390
26	0.6299	0.2222	0.1061	0.0418
27	0.6224	0.2249	0.1137	0.0390
28	0.6292	0.2248	0.1077	0.0383
29	0.6398	0.2171	0.1051	0.0379
30	0.6229	0.2324	0.1059	0.0388
31	0.6527	0.2325	0.0766	0.0382
32	0.6659	0.2285	0.0714	0.0342
33	0.6473	0.2310	0.0791	0.0426
34	0.6433	0.2391	0.0767	0.0408
35	0.0000	0.3228	0.2992	0.3780

RESULTS

In this project, I learned how to apply advanced SQL concepts like Windows Functions, etc. I understood how the real-world industry works. It helped me in mastering my SQL concepts. I learned how to ask the right questions given the circumstances. From the given data and questions, which columns to consider and how to find the valuable insights which help the business to grow. I learned how the company find different areas related to the company to improve it further. I got to know about investigating metric spike (why there is a boom and why there is a dip).

THANK YOU