

ADVANCED SQL

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Trainity Project

PREPARED BY: REINE DB, DATA ANALYST TRAINEE

About the Project

Operation Analytics is the analysis for complete end to end operations of a company. In reference to the analysis, the company discovers the areas on where they should improve on. The analyst works closely with the ops team, support team, marketing team, etc. and assist with their insights out of the data collected.

Being one of the most crucial components of a business, this form of analysis is also utilized to forecast the general upward or downward trend in a company's fortune. Better automation, improved communication among cross-functional teams, and more efficient workflows are the results.

Analyzing **metric spikes** is a crucial component of operational analytics since a data analyst needs to be able to answer queries like, "Why is there a decline in daily engagement?" or at least help other teams answer these questions. Why have sales decreased? Etc. Daily answers to questions like these are required, so it is crucial to look into metric increase.

```
3.018", "deltaStartTimeMi  
RequestHandler", "met  
ration Log", "duration  
bParams": "null", "cla  
": "2017-06-03T18:43:3  
son", "class": "com.or  
wg0k", "sizeChars": "  
INFO", "webURL": "/app/  
47-498454af885d", "s  
"2017-06-03T18:46:92  
equestHandler", "met  
ration Log", "duration  
bParams": "file=char  
d7-6ead86e273d1", "s  
": "2017-06-03T18:42:1  
equestHandler", "met  
ration Log", "duration  
bParams": "null", "cla  
": "2017-06-03T18:43:3  
5-839146a20f09", "se  
": "2017-06-03T18:43:3  
on", "class": "com.or  
wg0k", "sizeChars": "  
INFO", "webURL": "/app/  
47-498454af885d", "s  
'2017-06-03T18:46:921  
equestHandler", "metr
```

Project's Approach

The project approach adopted involved the use of MS SQL Server Management Studio (SSMS), a powerful tool for managing and administering Microsoft SQL Server. The tool was instrumental in facilitating the creation and management of databases, database objects, and data manipulation tasks. It also provided an intuitive user interface for monitoring server activity, managing security, and troubleshooting issues that arose during development. The use of MS SQL Server Management Studio was essential to the success of the project, as it allowed for efficient collaboration and streamlined development processes of the large data sets. Overall, the tool proved to be a valuable asset in delivering a high-quality, robust database solution that met the project requirements.

Tech-Stack Used

- MS SQL Server Management Studio (SSMS), to execute and run SQL queries and subqueries.
- The purpose behind using this tool was due to its strong managing and querying SQL Server databases. Its advanced features, such as debugging and performance tuning, makes it a better choice for effective data management and manipulation of the objects..

INSIGHTS

- Gained deeper understanding of SQL queries and comparative analysis of different syntax formats. How they can be used to extract meaningful insights from complex data sets.
- Furthermore, this project allowed me to develop my skills in advance aggregate functions, sub-queries and dealing with large data sets.
- Additionally, Gained knowledge in calculating metrics and models about being mindful of the datatypes and their conversion so as to use them for better and meaningful results of the metrics.
- Overall, working on this project using MS SQL Server as a tool has expanded my skill set and provided me with valuable experience in data analytics and investigating metrics.

[Link to Project](#)

Results

trainity

Operation Analytics & Investigating metric spike case study



Operation-1 (Table-1)

ds	job_id	actor_id	event	language	time_spent	org
2020-11-30	21	1001	skip	English	15	A
2020-11-30	22	1006	transfer	Arabic	25	B
2020-11-29	23	1003	decision	Persian	20	C
2020-11-28	23	1005	transfer	Persian	22	D
2020-11-28	25	1002	decision	Hindi	11	B
2020-11-27	11	1007	decision	French	104	D
2020-11-26	23	1004	skip	Persian	56	A
2020-11-25	20	1003	transfer	Italian	45	C

CASE STUDY 1 (JOB DATA)

A. Number of jobs reviewed per hour per day for November 2020

dates	Jobs_reviewed
2020-11-25 00:00:00.000	80
2020-11-26 00:00:00.000	64
2020-11-27 00:00:00.000	35
2020-11-28 00:00:00.000	218
2020-11-29 00:00:00.000	180
2020-11-30 00:00:00.000	180

```
/* Amount of jobs reviewed over time.
number of jobs reviewed per hour per day for November 2020 */
SELECT ds as dates,
ROUND((COUNT(job_id)/(SUM(CAST(time_spent AS FLOAT))/3600)), 0) AS Jobs_reviewed
FROM job_data
WHERE ds >= '2020-11-01' AND ds <= '2020-11-30'
GROUP BY ds;
```

B. THROUGHPUT

- **Throughput:** It is the no. of events happening per second.
- **7 day rolling average of throughput.** Daily metric or 7-day rolling.

```
SELECT
ROUND(COUNT(event)/(SUM(TRY_CAST(time_spent AS FLOAT))), 2) AS "weekly_throughput"
FROM job_data;
```

weekly_throughput
0.03

```
)SELECT
ds AS Dates,
ROUND(CAST(COUNT(event) AS float),2)/
ROUND(SUM(CAST(time_spent AS FLOAT)), 2) AS "Daily Throughput"
FROM job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds
ORDER BY ds;
```

Dates	Daily Throughput
2020-11-25 00:00:00.000	0.0222222222222222
2020-11-26 00:00:00.000	0.0178571428571429
2020-11-27 00:00:00.000	0.00961538461538462
2020-11-28 00:00:00.000	0.0606060606060606
2020-11-29 00:00:00.000	0.05
2020-11-30 00:00:00.000	0.05

The purpose of using a rolling average is to smooth out any daily fluctuations and provide a more accurate representation of the overall trend. However, a daily metric is useful for monitoring day-to-day changes and identifying any sudden spikes or dips in throughput.

A 7-day rolling average can provide a more stable and accurate picture of the overall trend, which can be useful for forecasting and long-term planning.

In results it shows that the highest Daily throughput is 0.06 on 2020-11-28 and the weekly throughput is 0.03

C. PERCENTAGE SHARE OF EACH LANGUAGE

- Share of each language for different contents.
- The percentage share of each language in the last 30 days.

```
SELECT language AS languages, ROUND(100*COUNT(*)/total, 2) AS percentage
FROM job_data
CROSS JOIN (SELECT COUNT(*) AS total FROM job_data) AS sub
GROUP BY language, total;
```

	languages	percentage
1	NULL	11
2	Arabic	11
3	English	11
4	French	11
5	Hindi	11
6	Italian	11
7	Persian	33

Persian language has the highest percentage share amongst the total of 9 languages including 'NULL' into consideration with a value of 33%

D. DUPLICATE ROWS

- Rows that have the same value present in them.
- Display duplicates from the table

```
/* duplicate rows */
SELECT actor_id, COUNT(*) AS duplicates
FROM job_data
GROUP BY actor_id
HAVING COUNT(*) > 1;
```

	actor_id	duplicates
1	1003.0	2

CASE STUDY 2 (INVESTIGATING METRIC SPIKE)

A. USER ENGAGEMENT

To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Weekly user engagement.

```
/* weekly user engagement */
SELECT DATEPART(WEEK, occurred_at) AS "Week Numbers",
COUNT(DISTINCT user_id) AS "Weekly Active Users"
FROM [Operation Analytics].dbo.events
WHERE event_type = 'engagement'
GROUP BY DATEPART(WEEK, occurred_at)
ORDER BY [Week Numbers];
```

	Week Numbers	Weekly Active Users
1	18	663
2	19	1068
3	20	1113
4	21	1154
5	22	1121
6	23	1186
7	24	1232
8	25	1275
9	26	1264
10	27	1302
11	28	1372
12	29	1365
13	30	1376
14	31	1467
15	32	1299
16	33	1225
17	34	1225
18	35	1204
19	36	104

Operation-2 (Table-1 (Users))

user_id	A unique ID per user. Can be joined to user_id in either of the other tables.
created_at	The time the user was created (first signed up)
state	The state of the user (active or pending)
activated_at	The time the user was activated, if they are active
company_id	The ID of the user's company
language	The chosen language of the user

B. USER GROWTH

Amount of users growing over time for a product.

The user growth for product

```
/* user growth for product */

SELECT Months, Users,
       ROUND(((Users/LAG(Users, 1) OVER
              (ORDER BY Months) -1)*100), 2) AS [Growth in %]
FROM
(
    SELECT
        DATEPART(MONTH, created_at) AS Months,
        COUNT(activated_at) AS Users
    FROM [Operation Analytics].dbo.users
    WHERE activated_at IS NOT NULL
    GROUP BY DATEPART(MONTH, created_at)
) sub;
```

Months	Users	Growth in %
1	712	NULL
2	685	-100
3	765	0
4	907	0
5	993	0
6	1086	0
7	1281	0
8	1347	0
9	330	-100
10	390	0
11	399	0
12	486	0

C. WEEKLY RETENTION

Users getting retained weekly after signing-up for a product.

The weekly retention of users-sign up cohort

```
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 - n.first AS week_number
FROM
(SELECT user_id, DATEPART(WEEK, occurred_at) AS login_week
FROM [Operation Analytics].dbo.events
GROUP BY user_id, DATEPART(WEEK, occurred_at)) m
JOIN
(SELECT user_id, MIN(DATEPART(WEEK, occurred_at)) AS first
FROM [Operation Analytics].dbo.events
GROUP BY user_id) n
ON m.user_id = n.user_id
) sub
GROUP BY first
ORDER BY first;
```


D. WEEKLY ENGAGEMENT

- To measure the activeness of a user.
- Measuring if the user finds quality in a product/service weekly.**
- The weekly engagement per device.

Operation-2 (Table-2(events))

user_id	The ID of the user logging the event. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
event_type	The general event type. There are two values in this dataset: "signup_flow", which refers to anything occurring during the process of a user's authentication, and "engagement", which refers to general product usage after the user has signed up for the first time
event_name	The specific action the user took. Possible values include: create_user: User is added to Yammer's database during signup process enter_email: User begins the signup process by entering her email address enter_info: User enters her name and personal information during signup process complete_signup: User completes the entire signup/authentication process home_page: User loads the home page like_message: User likes another user's message login: User logs into Yammer search_autocomplete: User selects a search result from the autocomplete list search_run: User runs a search query and is taken to the search results page search_click_result_X: User clicks search result X on the results page, where X is a number from 1 through 10. send_message: User posts a message view_inbox: User views messages in her inbox
location:	The country from which the event was logged (collected through IP address).
device:	The type of device used to log the event.

```

SELECT DATEPART(WEEK, occurred_at) AS "Week Numbers",
COUNT(DISTINCT CASE WHEN device = 'dell inspiron notebook' THEN user_id END) AS "Dell Inspiron Notebook",
COUNT(DISTINCT CASE WHEN device = 'iphone 5' THEN user_id END) AS "iPhone 5",
COUNT(DISTINCT CASE WHEN device = 'iphone 4s' THEN user_id END) AS "iPhone 4S",
COUNT(DISTINCT CASE WHEN device = 'windows surface' THEN user_id END) AS "Windows Surface",
COUNT(DISTINCT CASE WHEN device = 'macbook air' THEN user_id END) AS "Macbook Air",
COUNT(DISTINCT CASE WHEN device = 'iphone 5s' THEN user_id END) AS "iPhone 5S",
COUNT(DISTINCT CASE WHEN device = 'macbook pro' THEN user_id END) AS "Macbook Pro",
COUNT(DISTINCT CASE WHEN device = 'kindle fire' THEN user_id END) AS "Kindle Fire",
COUNT(DISTINCT CASE WHEN device = 'ipad mini' THEN user_id END) AS "iPad Mini",
COUNT(DISTINCT CASE WHEN device = 'nexus 7' THEN user_id END) AS "Nexus 7",
COUNT(DISTINCT CASE WHEN device = 'nexus 5' THEN user_id END) AS "Nexus 5",
COUNT(DISTINCT CASE WHEN device = 'samsung galaxy s4' THEN user_id END) AS "Samsung Galaxy S4",
COUNT(DISTINCT CASE WHEN device = 'Lenovo Thinkpad' THEN user_id END) AS "Lenovo Thinkpad",
COUNT(DISTINCT CASE WHEN device = 'Samsung Galaxy Tablet' THEN user_id END) AS "Samsung Galaxy Tablet",
COUNT(DISTINCT CASE WHEN device = 'Acer Aspire Notebook' THEN user_id END) AS "Acer Aspire Notebook",
COUNT(DISTINCT CASE WHEN device = 'Asus Chromebook' THEN user_id END) AS "Asus Chromebook",
COUNT(DISTINCT CASE WHEN device = 'HTC One' THEN user_id END) AS "HTC One",
COUNT(DISTINCT CASE WHEN device = 'Nokia Lumia 635' THEN user_id END) AS "Nokia Lumia 635",
COUNT(DISTINCT CASE WHEN device = 'Samsung Galaxy Note' THEN user_id END) AS "Samsung Galaxy Note",
COUNT(DISTINCT CASE WHEN device = 'Acer Aspire Desktop' THEN user_id END) AS "Acer Aspire Desktop",
COUNT(DISTINCT CASE WHEN device = 'Mac Mini' THEN user_id END) AS "Mac Mini",
COUNT(DISTINCT CASE WHEN device = 'HP Pavilion Desktop' THEN user_id END) AS "HP Pavilion Desktop",
COUNT(DISTINCT CASE WHEN device = 'Dell Inspiron Desktop' THEN user_id END) AS "Dell Inspiron Desktop",
COUNT(DISTINCT CASE WHEN device = 'iPad Air' THEN user_id END) AS "iPad Air",
COUNT(DISTINCT CASE WHEN device = 'Amazon Fire Phone' THEN user_id END) AS "Amazon Fire Phone",
COUNT(DISTINCT CASE WHEN device = 'Nexus 10' THEN user_id END) AS "Nexus 10"
FROM [Operation Analytics].dbo.events
WHERE event_type = 'engagement'
GROUP BY DATEPART(WEEK, occurred_at)
ORDER BY DATEPART(WEEK, occurred_at);

```

Week Numbers	Dell Inspiron Notebook	iPhone 5	iPhone 4S	Windows Surface	Macbook Air	iPhone 5S	Macbook Pro	Kindle Fire	iPad Mini	Nexus 7
18	46	65	21	10	54	42	143	6	19	18
19	77	113	46	10	121	73	252	27	30	30
20	83	115	44	16	112	79	266	21	36	41
21	84	125	55	21	119	79	256	23	32	32
22	80	137	45	17	110	74	247	30	23	29
23	92	125	45	15	145	71	251	21	34	45
24	103	152	53	14	124	79	266	25	33	36
25	99	142	53	22	152	79	255	25	39	49
26	105	137	40	22	121	78	275	24	30	51
27	89	152	50	21	134	94	269	26	43	46
28	89	163	67	33	142	83	302	25	35	40
29	103	151	61	33	148	93	295	31	35	39
30	113	144	60	28	148	90	295	37	34	45
31	127	152	65	19	159	103	322	25	35	62
32	113	135	56	19	147	71	321	14	27	38
33	104	119	34	10	125	67	307	12	30	25
34	110	110	35	15	133	65	312	14	28	30
35	105	101	50	18	136	70	292	13	25	33
36	9	2	6	3	10	3	17	3	2	2

Nexus 5	Samsung Galaxy S4	Lenovo Thinkpad	Samsung Galaxy Tablet	Acer Aspire Notebook	Asus Chromebook	HTC One	Nokia Lumia 635
40	52	86	8	20	21	16	17
73	82	153	11	33	42	19	33
87	91	178	6	41	27	30	23
103	93	173	9	40	41	29	22
91	84	167	6	47	38	21	25
96	105	176	10	41	52	24	25
88	99	176	14	43	49	20	31
87	101	165	11	40	43	20	35
89	99	197	12	47	38	21	37
87	112	192	12	35	49	23	42
84	116	202	15	49	52	27	31
85	122	220	9	49	50	26	35
77	123	209	13	53	49	31	43
84	103	206	9	60	56	31	34
69	100	207	8	55	56	13	28
67	82	179	6	55	62	18	28
70	80	191	12	46	49	19	27
70	90	193	14	63	47	25	17
4	6	16	0	3	6	2	2

Samsung Galaxy Note	Acer Aspire Desktop	Mac Mini	HP Pavilion Desktop	Dell Inspiron Desktop	iPad Air	Amazon Fire Phone	Nexus 10
7	9	6	14	18	27	4	16
15	26	13	37	58	52	9	30
11	23	18	40	36	55	12	25
18	23	26	30	52	59	11	22
20	29	18	44	41	51	5	25
19	25	25	38	52	58	5	27
14	22	18	54	53	41	16	45
20	24	29	56	59	57	11	38
14	28	21	52	52	57	13	29
9	29	11	46	60	56	13	29
15	29	15	56	53	55	10	37
10	30	28	56	56	54	6	26
16	28	31	58	54	52	12	25
15	33	23	42	54	70	12	36
14	31	24	51	44	55	14	24
12	35	20	51	57	48	12	30
13	39	32	38	37	40	14	23
13	30	30	36	49	39	11	25
1	1	2	1	1	0	0	2

E. EMAIL ENGAGEMENT

- Users engaging with the email service.
- The email engagement metrics.

Operation-2 (Table-3(email_events))

user_id	The ID of the user to whom the event relates. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
action	The name of the event that occurred. "sent_weekly_digest" means that the user was delivered a digest email showing relevant conversations from the previous day. "email_open" means that the user opened the email. "email_clickthrough" means that the user clicked a link in the email.

```

SELECT *
FROM [Operation Analytics].[dbo].[email_events];

SELECT sub.Week,
ROUND((CAST(sub.weekly_digest AS float)/CAST(sub.total AS float)*100),2) AS "weekly_digest_rate",
ROUND((CAST(sub.emailOpens AS float)/CAST(sub.total AS float)*100),2) AS "Email Open Rate",
ROUND((CAST(sub.emailClickthroughs AS float)/CAST(sub.total AS float)*100),2) AS "Email Clickthrough Rate",
ROUND((CAST(sub.reengagement_emails AS float)/CAST(sub.total AS float)*100),2) AS "Reengagement Email Rate"
FROM
(
    SELECT DATEPART(WEEK, occurred_at) AS Week,
    COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE NULL END) AS weekly_digest,
    COUNT(CASE WHEN action = 'email_open' THEN user_id ELSE NULL END) AS emailOpens,
    COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE NULL END) AS emailClickthroughs,
    COUNT(CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE NULL END) AS reengagement_emails,
    COUNT(user_id) AS total
    FROM [Operation Analytics].[dbo].[email_events]
    GROUP BY DATEPART(WEEK, occurred_at)
) sub
JOIN [Operation Analytics].[dbo].[email_events] AS email_events
ON DATEPART(WEEK, email_events.occurred_at) = sub.Week
GROUP BY sub.Week, sub.weekly_digest, sub.emailOpens, sub.emailClickthroughs, sub.reengagement_emails, sub.total
ORDER BY sub.Week;

```

	Week	weekly_digest_rate	Email Open Rate	Email Clickthrough Rate	Reengagement Email Rate
1	18	62.32	21.28	11.39	5.01
2	19	63.45	22.24	10.49	3.83
3	20	62.16	22.67	11.13	4.04
4	21	61.62	22.64	11.43	4.31
5	22	63.52	22.82	9.97	3.69
6	23	63.59	21.56	10.66	4.19
7	24	62.39	22.34	11.18	4.09
8	25	61.61	22.92	10.99	4.48
9	26	63.77	21.79	10.54	3.9
10	27	62.99	22.22	10.61	4.18
11	28	62.24	22.49	11.37	3.9
12	29	62.92	22.48	10.77	3.83
13	30	63.98	21.71	10.51	3.79
14	31	62.29	23.24	10.59	3.88
15	32	65.27	23.25	7.66	3.82
16	33	66.59	22.85	7.14	3.42
17	34	64.73	23.1	7.91	4.26
18	35	64.33	23.91	7.67	4.08
19	36	0	32.28	29.92	37.8