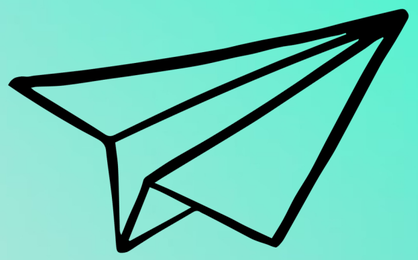
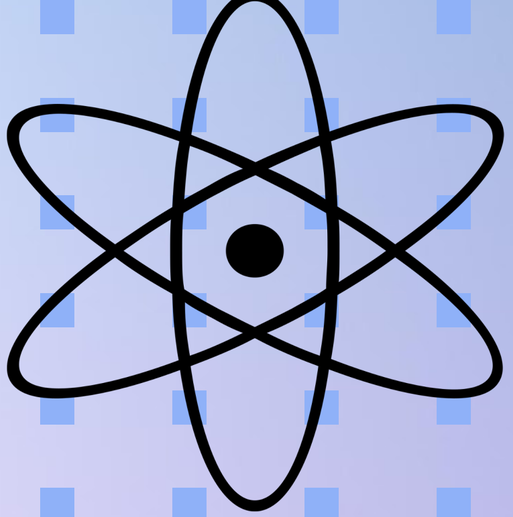


Operation Analytics and Investigating Metric Spike

— Anurag John Phillips

INDEX



Project Description

Case Study 1: In this project we are examining the number of jobs reviewed per hour per day in November 2020, the 7-day rolling average of throughput, language distribution in the last 30 days, and identifying duplicate rows using SQL queries.

Case Study 2: In this project we are analyzing data from three tables namely "users," "events," and "email_events" to measure User Engagement, Growth, and Email Service Performance. Weekly insights to help understand User activeness, Product Quality, and Email Engagement Metrics.

Approach

Case 1 Approach: I used SQL queries to analyse the dataset. I created more rows in the dataset. I calculated the number of jobs reviewed per hour per day and the 7-day rolling average of throughput (used Google to understand more about 7-day rolling average). Additionally, I used grouping and filtering techniques to determine the percentage share of each language in the last 30 days and identify any duplicate rows in the data.

Case 2 Approach: Here I had to create a database and then upload all the data which was in CSV to SQL, made three different tables and used SQL queries to extract, filter, and aggregate data from the provided tables. The final results were presented in a structured format to present the insights gained from the analysis. I have presented both the queries used and the output for better understanding.

Tech-Stack Used

- MySQL Workbench Version 8.0.33 for MacBook.
- Microsoft Excel for Mac Version 16.74.

Insight

Analysing the data helped us in answering these questions:

Case 1 – Job Data

Number of jobs reviewed: Number of jobs reviewed over time.
Throughput: Calculate 7 day rolling average of throughput?
Percentage share of each language: Share of each language for different contents.
Duplicate rows: Rows that have the same value present in them.

Case 2 – Investigating Metric Strike

User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.
User Growth: Number of users growing over time for a product.
Weekly Retention: Users getting retained weekly after signing-up for a product.
Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
Email Engagement: Users engaging with the email service.

```
-- Insert the data into the job_data table
INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org)
VALUES
('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'),
('2020-11-30', 21, 1004, 'decision', 'English', 30, 'A'),
('2020-11-29', 23, 1001, 'skip', 'Persian', 18, 'C'),
('2020-11-30', 25, 1001, 'decision', 'Hindi', 12, 'A'),
('2020-11-30', 22, 1003, 'transfer', 'Arabic', 20, 'B'),
('2020-11-28', 23, 1002, 'decision', 'Persian', 35, 'D'),
('2020-11-27', 11, 1004, 'decision', 'French', 40, 'D'),
('2020-11-26', 23, 1006, 'transfer', 'Persian', 40, 'A'),
('2020-11-25', 20, 1007, 'transfer', 'Italian', 30, 'C'),
('2020-11-29', 25, 1006, 'skip', 'Hindi', 18, 'B'),
('2020-11-30', 21, 1002, 'transfer', 'English', 25, 'B'),
('2020-11-29', 23, 1005, 'decision', 'Persian', 22, 'D'),
('2020-11-30', 21, 1006, 'decision', 'Arabic', 15, 'B'),
('2020-11-28', 23, 1003, 'transfer', 'Persian', 22, 'C'),
('2020-11-27', 11, 1004, 'decision', 'French', 104, 'D'),
('2020-11-26', 23, 1001, 'skip', 'Persian', 56, 'A'),
('2020-11-25', 20, 1005, 'transfer', 'Italian', 45, 'D'),
('2020-11-29', 25, 1003, 'skip', 'Hindi', 11, 'B'),
('2020-11-30', 22, 1007, 'transfer', 'Arabic', 30, 'B'),
('2020-11-28', 23, 1002, 'decision', 'Persian', 35, 'D'),
('2020-11-27', 11, 1006, 'decision', 'French', 40, 'D'),
('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),
('2020-11-25', 20, 1001, 'transfer', 'Italian', 45, 'C'),
('2020-11-29', 25, 1007, 'skip', 'Hindi', 18, 'B'),
('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'),
('2020-11-30', 21, 1004, 'decision', 'English', 30, 'A'),
('2020-11-29', 23, 1001, 'skip', 'Persian', 18, 'C'),
('2020-11-30', 25, 1001, 'decision', 'Hindi', 12, 'A'),
('2020-11-30', 22, 1003, 'transfer', 'Arabic', 20, 'B'),
('2020-11-28', 23, 1002, 'decision', 'Persian', 35, 'D'),
('2020-11-27', 11, 1004, 'decision', 'French', 40, 'D'),
('2020-11-26', 23, 1006, 'transfer', 'Persian', 40, 'A'),
('2020-11-25', 20, 1007, 'transfer', 'Italian', 30, 'C'),
('2020-11-29', 25, 1006, 'skip', 'Hindi', 18, 'B'),
('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'),
('2020-11-30', 21, 1004, 'decision', 'English', 30, 'A'),
('2020-11-29', 23, 1001, 'skip', 'Persian', 18, 'C'),
('2020-11-30', 25, 1001, 'decision', 'Hindi', 12, 'A');
```

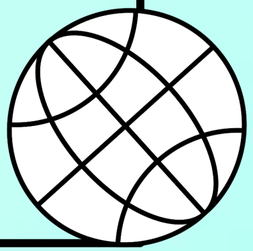
CASE STUDY-1

DATABASE Creation and Inserting the following values

```
-- Create the database
CREATE SCHEMA IF NOT EXISTS jobs_database;

-- Use the newly created database
USE jobs_database;

-- Create the job_data table
CREATE TABLE IF NOT EXISTS job_data (
    ds DATE,
    job_id INT,
    actor_id INT,
    event VARCHAR(10),
    language VARCHAR(20),
    time_spent INT,
    org VARCHAR(10)
);
```



Case Study 1 (Job Data)

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

```
SELECT ds AS date,
SUM(jobs_done) AS jobs_done_everyday,
SUM(jobs_done) / 24 AS hours_spent_everyday
FROM (
SELECT ds, COUNT(job_id) AS jobs_done
FROM job_data
WHERE ds >= '2020-11-01' AND ds <= '2020-11-30'
GROUP BY ds
) subquery
GROUP BY date;
```

date	jobs_done_everyd...	hours_spent_everyd...
2020-11-30	17	0.7083
2020-11-29	11	0.4583
2020-11-28	10	0.4167
2020-11-27	7	0.2917
2020-11-26	7	0.2917
2020-11-25	7	0.2917

Case Study 1 (Job Data)

B: Calculate 7 day rolling average of throughput?

For throughput, do you prefer daily metric or 7-day rolling and why?

```
SELECT
date_value,
AVG(throughput_per_second) OVER (ORDER BY date_value ROWS
BETWEEN 6 PRECEDING AND CURRENT ROW) AS
rolling_avg_throughput,
throughput_per_second
FROM (
SELECT ds AS date_value, COUNT(*) / (24 * 60 * 60) AS
throughput_per_second
FROM job_data
GROUP BY ds
) my_subquery
ORDER BY date_value;
```

	date_value	rolling_avg_throughput	throughput_per_second
	2020-11-25	0.00010000	0.0001
	2020-11-26	0.00010000	0.0001
	2020-11-27	0.00010000	0.0001
	2020-11-28	0.00010000	0.0001
	2020-11-29	0.00010000	0.0001
	2020-11-30	0.00011667	0.0002

For throughput we prefer, 7-day rolling average because it is like a special way of looking at the throughput that gives us a steadier and smoother view of how things are going over time. It does this by considering the data from the last 7 days and finding their average. This way, it takes out the ups and downs that happen daily.

Case Study 1 (Job Data)

C: Calculate the percentage share of each language in the last 30 days?

```
SELECT language,
ROUND(COUNT(*) * 100 / SUM(COUNT(*)) OVER (), 2)
AS percentage_share
FROM job_data
GROUP BY language;
```

language	percentage_sha...
English	11.86
Arabic	11.86
Persian	35.59
Hindi	16.95
French	11.86
Italian	11.86

By using this query, we will get the percentage share of each language in the **job_data** table.

Case Study 1 (Job Data)

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

```
SELECT  actor_id, COUNT(*) AS duplicates

FROM job_data

GROUP BY actor_id

HAVING COUNT(*) > 1;
```

actor_id	duplicates
1001	11
1006	9
1003	10
1005	5
1002	7
1007	7
1004	10

When we execute this query, it will display the actor_ids that have duplicates in the job_data table, along with the count of occurrences for each actor_id.

Case Study 2 (Investigating metric spike)

A: Calculate the weekly user engagement?

```
SELECT
week_num AS week_number,
COUNT(DISTINCT user_id) AS weekly_user_engagement
FROM
(
SELECT
 user_id,
DATE_FORMAT(occurred_at, '%U') AS week_num
FROM
 events
WHERE
event_type = 'engagement'
) weekly_events
GROUP BY
week_number;
```

week_number	weekly_user_engageme...
17	87
18	197
19	211
20	198
21	210
22	233
23	227
24	258
25	249
26	233
27	252
28	241
29	253
30	275
31	232
32	282
33	292
34	304
35	20

Case Study 2 (Investigating metric spike)

B: Calculate the user growth for product?

```
SELECT
month as 'year_month',
new_users,
ROUND((new_users - LAG(new_users) OVER (ORDER BY month)) /
LAG(new_users) OVER (ORDER BY month) * 100, 2) AS growth_percentage
FROM
(
SELECT
DATE_FORMAT(created_at, '%Y-%m') AS month,
COUNT(*) AS new_users
FROM
users
GROUP BY
DATE_FORMAT(created_at, '%Y-%m')
) user_growth
ORDER BY
'year_month';
```

year_month	new_users	growth_percentage
2013-01	332	NULL
2013-02	328	-1.20
2013-03	383	16.77
2013-04	410	7.05
2013-05	486	18.54
2013-06	485	-0.21
2013-07	608	25.36
2013-08	636	4.61
2013-09	699	9.91
2013-10	826	18.17
2013-11	816	-1.21
2013-12	972	19.12
2014-01	1083	11.42
2014-02	1054	-2.68
2014-03	1231	16.79
2014-04	1419	15.27
2014-05	1597	12.54
2014-06	1728	8.20
2014-07	1983	14.76
2014-08	1990	0.35

Case Study 2 (Investigating metric spike)

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

```
SELECT
week_period, cohort_size, cohort_retained,
cohort_retained / cohort_size AS percent_retained
FROM ( SELECT
week_period, COUNT(DISTINCT user_id) AS cohort_size,
COUNT(DISTINCT CASE WHEN is_retained = 1 THEN user_id
END) AS cohort_retained
FROM ( SELECT
cohort.week_period, cohort.user_id,
MAX(occurred_at = cohort.activated_at) AS is_retained
FROM ( SELECT
user_id, activated_at,
DATE_FORMAT(activated_at, '%Y-%U') AS week_period
FROM users
WHERE state = 'active'
) cohort
INNER JOIN ( SELECT
user_id, DATE_FORMAT(occurred_at, '%Y-%U') AS
week_period, MIN(occurred_at) AS occurred_at
FROM events
GROUP BY user_id, week_period
) cohort_2 ON cohort.user_id = cohort_2.user_id AND
cohort.week_period = cohort_2.week_period
GROUP BY cohort.week_period, cohort.user_id
) cohort_3
GROUP BY week_period
) cohort_4;
```

week_period	cohort_size	cohort_retained	percent_retained
2014-17	75	72	0.9600
2014-18	163	163	1.0000
2014-19	185	185	1.0000
2014-20	176	176	1.0000
2014-21	183	183	1.0000
2014-22	196	196	1.0000
2014-23	196	196	1.0000
2014-24	229	229	1.0000
2014-25	207	207	1.0000
2014-26	201	201	1.0000
2014-27	222	222	1.0000
2014-28	215	215	1.0000
2014-29	221	221	1.0000
2014-30	238	238	1.0000
2014-31	193	193	1.0000
2014-32	245	245	1.0000
2014-33	261	261	1.0000
2014-34	259	259	1.0000
2014-35	18	18	1.0000

Case Study 2 (Investigating metric spike)

D: Calculate the weekly engagement per device?

```
SELECT
DATE_FORMAT(occurred_at,
'%Y-%U’)
AS week,
device,
COUNT(*) AS
weekly_engagement
FROM
events
WHERE
event_type = 'engagement'
GROUP BY
DATE_FORMAT(occurred_at,
'%Y-%U'),
device
ORDER BY
week, device;
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	week	acer aspire desktop	acer aspire notebook	amazon fire phone	asus chromebook	dell inspiron desktop	dell inspiron notebook	hp pavilion desktop	htc one	ipad air	ipad mini	iphone 4s	iphone 5	iphone 5s	kindle fire	lenovo thinkpad	mac mini	macbook air	macbook pro	nexus 10	nexus 5	nexus 7	nokia lumia 635	samsung galaxy tablet	samsung galaxy note	samsung galaxy s4	windows surface
2	17	9	21	11	19	17	20	17	4	9	25	26	67	23	NULL	84	3	32	93	NULL	24	30	24	16	8	45	NULL
3	18	21	18	36	88	24	84	47	16	66	64	29	58	63	22	242	5	174	334	8	43	39	18	NULL	31	88	8
4	19	NULL	48	19	25	6	45	48	38	71	23	50	127	74	24	156	40	166	250	33	149	37	16	NULL	18	112	19
5	20	6	31	11	38	81	73	9	11	104	26	20	117	143	19	165	5	135	295	11	109	35	22	NULL	38	120	12
6	21	20	14	NULL	94	40	67	38	33	55	61	59	215	73	20	243	3	99	243	45	124	11	5	11	24	69	19
7	22	NULL	62	8	77	59	112	34	17	85	31	55	88	99	56	215	12	204	256	77	106	82	78	11	NULL	79	26
8	23	50	68	23	149	72	67	122	3	48	34	58	188	95	34	174	14	188	223	23	90	32	24	21	NULL	65	14
9	24	45	41	10	73	67	74	59	25	65	43	94	219	100	23	278	21	206	328	27	98	52	6	10	6	113	45
10	25	12	56	4	64	64	119	33	27	96	34	54	183	63	6	247	15	117	283	30	70	59	43	29	NULL	113	37
11	26	24	13	17	59	43	98	75	9	64	19	70	220	143	31	218	24	150	338	21	48	87	73	6	25	80	41
12	27	32	71	20	53	51	70	36	28	31	22	130	200	113	37	181	40	116	378	43	87	59	9	43	15	111	36
13	28	30	61	5	40	61	107	65	44	76	22	53	189	69	39	196	20	162	281	23	78	30	20	NULL	12	200	37
14	29	16	13	16	68	47	96	27	29	70	75	46	147	80	29	220	9	192	249	34	61	33	45	3	12	115	20
15	30	60	85	17	62	45	108	54	7	132	37	109	188	100	5	292	44	94	327	41	118	34	49	15	18	144	15
16	31	25	51	NULL	42	51	108	22	5	100	28	87	268	53	7	161	4	158	330	30	75	26	47	29	15	110	25
17	32	24	49	45	76	22	144	40	20	75	43	40	182	147	22	188	9	99	495	43	51	26	13	14	18	91	6
18	33	32	79	26	34	59	131	33	24	42	88	62	234	106	27	147	50	117	405	39	68	37	22	22	50	148	29
19	34	26	79	38	129	67	67	57	63	86	20	67	107	157	14	277	38	141	366	55	100	86	14	8	2	88	60
20	35	NULL	7	NULL	NULL	4	35	NULL	6	NULL	8	24	NULL	NULL	NULL	22	NULL	12	21	NULL	4	NULL	5	NULL	NULL	NULL	8
21																											

Showing the results in Excel for better visualization

Case Study 2 (Investigating metric spike)

E: Calculate the email engagement metrics

```
SELECT
week as Year_Week,
num_emails_sent as Total_Emails_Sent,
num_emails_opened as Total_Emails_Opened,
num_emails_clicked as Total_Emails_Clicked,
100.0 * num_emails_opened / num_emails_sent AS
Email_Opening_Rate,
100.0 * num_emails_clicked / num_emails_sent AS
Email_Clicking_Rate
FROM (
SELECT
DATE_FORMAT(occurred_at, '%Y-%U') AS week,
COUNT(CASE WHEN action IN ('sent weekly_digest',
'sent_reengagement_email') THEN 1 END) AS num_emails_sent,
COUNT(CASE WHEN action = 'email_open' THEN 1 END)
AS num_emails_opened,
COUNT(CASE WHEN action = 'email_clickthrough' THEN
1 END) AS num_emails_clicked
FROM
email_events
WHERE
action IN ('sent weekly_digest', 'sent_reengagement_email',
'email_open', 'email_clickthrough')
GROUP BY
DATE_FORMAT(occurred_at, '%Y-%U')
) Email_Eng;
```

Year_Week	Total_Emails_Sent	Total_Emails_Open...	Total_Emails_Clicked	Email_Opening_Rate	Email_Clicking_Rate
2014-22	192	987	488	514.06250	254.16667
2014-23	197	1075	538	545.68528	273.09645
2014-24	226	1155	554	511.06195	245.13274
2014-30	231	1383	630	598.70130	272.72727
2014-33	264	1432	490	542.42424	185.60606
2014-19	173	972	477	561.84971	275.72254
2014-20	191	1004	507	525.65445	265.44503
2014-25	196	1096	530	559.18367	270.40816
2014-26	219	1165	556	531.96347	253.88128
2014-32	200	1337	418	668.50000	209.00000
2014-27	213	1228	621	576.52582	291.54930
2014-34	261	1528	490	585.44061	187.73946
2014-31	222	1351	445	608.55856	200.45045
2014-29	213	1219	590	572.30047	276.99531
2014-18	157	912	430	580.89172	273.88535
2014-21	164	1014	443	618.29268	270.12195
2014-28	213	1250	599	586.85446	281.22066
2014-17	73	310	166	424.65753	227.39726
2014-35	48	41	38	85.41667	79.16667

Result

This project helped me a lot in understanding more about SQL. My learnings started from creating the database, then how to load the data where I spent a lot of time, because I kept getting error. Then eventually changing the user_type in events table from int to text and altering all the columns which contained dates from text to datetime. This all was new to me, so it helped me in learning new things, which although took a lot of time but this effort will help me in future.

It was also my first time in analysing such a big dataset.

Throughout this project, I have learned the importance of operational analytics and how it can empower businesses with valuable insights.