

Project – 03

Operation Analytics and Investigating Metric Spike

Project Description:

This project mainly focuses on analyzing the data which is provided by the respective company. My task is to derive insights and answer the questions asked by different departments. So that these insights are used by the operation team, support team, marketing team, etc to predict the overall growth or decline of a company's fortune. So that the company gets better automation, better understanding between cross-functional teams, and more effective workflows.

In case study 1, there is a job_data table; in case study 2 there are users, events, and email_events tables.

In Case Study 1 the insights are found based on the following questions:

- (i) **Number of jobs reviewed:** Amount of jobs reviewed over time.
- (ii)
- (iii) **Throughput:** Calculate the 7-day rolling average of throughput. For throughput, do you prefer daily metric or 7-day rolling and why?
- (iv) **Percentage share of each language:** Share of each language for different contents.
- (v) **Duplicate rows:** Rows that have the same value present in them

In case study 2 the insights are found based on the following questions:

- (i) **User engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
- (ii) **User Growth:** A product's users grow over time.
- (iii) **Weekly Retention:** Users are retained weekly after signing up for a product.
- (iv) **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
- (v) **Email Engagement:** Users engaging with the email service.

Using operation analytics, a business can determine which areas must be improved and uncover areas in which it can save money. Since this is one of a company's most critical components, it can also predict the future profitability of the business.

Project Approach:

This project is developed using MySQL Workbench. At first, I need to create a database by using a dataset file provided by the company. After that load, the data is loaded into the SQL workbench then performed analysis and find information that will help the operation team, support team, marketing team, etc.

Tech-Stack Used:

I used MySQL Workbench. MySQL Workbench is a visual editor that unifies data modeling, SQL development, and database administration in one interface. It allows you to visually design, generate and manage databases.

It is widely used to handle structured data. It is an open-source Relational Database Management System (RDBMS) developed by Oracle Corporation, Sun Microsystems that uses Structured Query Language (SQL) to interact with databases. And MySQL Workbench offers database migration options, making it easier to move data to and from the Microsoft SQL Server, Microsoft Access, and other RDBMS tables.

Insights:

Case Study – 1

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

SELECT ds AS Dates, ROUND((COUNT(job_id)/SUM(time_spent))*3600) AS "Jobs reviewed per Hour per Day"

FROM job_data

WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'

GROUP BY ds;

On the date 2020-11-28, the maximum number of jobs reviewed is 218.

	Dates	Jobs reviewed per Hour per Day
▶	2020-11-30	180
	2020-11-29	180
	2020-11-28	218
	2020-11-27	35
	2020-11-26	64
	2020-11-25	80

(B) Calculate the 7-day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

```
SELECT ROUND(COUNT(event)/SUM(time_spent),2) AS "Weekly  
Throughput"  
FROM job_data;
```

The weekly throughput is 0.03.

	Weekly Throughput
▶	0.03

```
SELECT ds AS Dates, ROUND (COUNT (event)/SUM (time_spent),2) AS  
"Daily Throughput"  
FROM job_data  
GROUP BY ds  
ORDER BY ds;
```

The highest throughput on the date 2020-11-28 is 0.06

	Dates	Daily Throughput
▶	2020-11-25	0.02
	2020-11-26	0.02
	2020-11-27	0.01
	2020-11-28	0.06
	2020-11-29	0.05
	2020-11-30	0.05

Metrics go up and down on a daily and weekly basis always. You will get the numbers faster every day and every minute you want. So rolling metrics are superb at showing if your metrics are trending up or down on a daily level. Finally, I prefer the daily metric rolling.

(C) Calculate 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) sub  
GROUP BY language;
```

The Persian Language has the highest percentage which is about 37.50%.

	Languages	Percentage
►	English	12.50
	Arabic	12.50
	Persian	37.50
	Hindi	12.50
	French	12.50
	Italian	12.50

(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 1003 has 2 duplicate rows.

	actor_id	Duplicates
►	1003	2

Case Study - 02

(A) Calculate the weekly user engagement?

```
SELECT WEEK(occurred_at) AS "Week Numbers", COUNT(DISTINCT
user_id) AS "Weekly Active Users"
FROM events
WHERE event_type = 'engagement'
GROUP BY 1
ORDER BY 1;
```

	Week Numbers	Weekly Active Users
►	17	258
	18	458
	19	485
	20	465
	21	461
	22	489
	23	482
	24	540
	25	512
	26	496
	27	517
	28	521
	29	520
	30	540
	31	448
	32	458
	33	461
	34	475
	35	29

(B) Calculate the user growth for the product?

```
SELECT Months, Users, ROUND(((Users/LAG(Users,1) OVER (ORDER BY
Months) - 1)*100),2) AS "Growth in %"
FROM (
```

```

SELECT MONTH(created_at) AS Months, COUNT(activated_at) AS
Users
FROM users
WHERE activated_at NOT IN ("")
GROUP BY 1
ORDER BY 1
) sub;

```

	Months	Users	Growth in %
▶	1	712	NULL
	2	685	-3.79
	3	765	11.68
	4	907	18.56
	5	993	9.48
	6	1086	9.37
	7	1281	17.96
	8	1347	5.15
	9	330	-75.50
	10	390	18.18
	11	399	2.31
	12	486	21.80

(C) Calculate the weekly retention of the 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 - first AS
week_number
    FROM (SELECT user_id, WEEK(occurred_at) AS login_week FROM
events GROUP BY 1, 2) m,
        (SELECT user_id, MIN(WEEK(occurred_at)) AS first FROM
events GROUP BY 1) n
    WHERE m.user_id = n.user_id
) sub
GROUP BY first
ORDER BY first;
```

[illegible]

(D) Calculate the weekly engagement per device?

```
SELECT WEEK(occurred_at) AS "Week Numbers",
COUNT(DISTINCT CASE WHEN device IN ('dell Inspiron Notebook') THEN
user_id ELSE NULL END) AS "Dell Inspiron Notebook",
COUNT(DISTINCT CASE WHEN device IN ('iphone 5') THEN user_id ELSE
NULL END) AS "iPhone 5",
COUNT(DISTINCT CASE WHEN device IN ('iphone 4s') THEN user_id ELSE
NULL END) AS "iPhone 4s",
COUNT(DISTINCT CASE WHEN device IN ('windows surface') THEN user_id
ELSE NULL END) AS "Windows Surface",
COUNT(DISTINCT CASE WHEN device IN ('macbook air') THEN user_id
ELSE NULL END) AS "Macbook Air",
COUNT(DISTINCT CASE WHEN device IN ('iphone 5s') THEN user_id ELSE
NULL END) AS "iPhone 5s",
COUNT(DISTINCT CASE WHEN device IN ('macbook pro') THEN user_id
ELSE NULL END) AS "Macbook Pro",
COUNT(DISTINCT CASE WHEN device IN ('kindle fire') THEN user_id ELSE
NULL END) AS "Kindle Fire",
COUNT(DISTINCT CASE WHEN device IN ('ipad mini') THEN user_id ELSE
NULL END) AS "iPad Mini",
COUNT(DISTINCT CASE WHEN device IN ('nexus 7') THEN user_id ELSE
NULL END) AS "Nexus 7",
COUNT(DISTINCT CASE WHEN device IN ('nexus 5') THEN user_id ELSE
NULL END) AS "Nexus 5",
COUNT(DISTINCT CASE WHEN device IN ('samsung galaxy s4') THEN
user_id ELSE NULL END) AS "Samsung Galaxy S4",
COUNT(DISTINCT CASE WHEN device IN ('lenovo thinkpad') THEN user_id
ELSE NULL END) AS "Lenovo Thinkpad",
COUNT(DISTINCT CASE WHEN device IN ('samsung galaxy tablet') THEN
user_id ELSE NULL END) AS "Samsung Galaxy Tablet",
```



```
COUNT(DISTINCT CASE WHEN device IN ('acer aspire notebook') THEN
user_id ELSE NULL END) AS "Acer Aspire Notebook",
COUNT(DISTINCT CASE WHEN device IN ('asus chromebook') THEN
user_id ELSE NULL END) AS "Asus Chromebook",
COUNT(DISTINCT CASE WHEN device IN ('htc one') THEN user_id ELSE
NULL END) AS "HTC One",
COUNT(DISTINCT CASE WHEN device IN ('nokia lumia 635') THEN user_id
ELSE NULL END) AS "Nokia Lumia 635",
COUNT(DISTINCT CASE WHEN device IN ('samsung galaxy note') THEN
user_id ELSE NULL END) AS "Samsung Galaxy Note",
COUNT(DISTINCT CASE WHEN device IN ('acer aspire desktop') THEN
user_id ELSE NULL END) AS "Acer Aspire Desktop",
COUNT(DISTINCT CASE WHEN device IN ('mac mini') THEN user_id ELSE
NULL END) AS "Mac Mini",
COUNT(DISTINCT CASE WHEN device IN ('hp pavilion desktop') THEN
user_id ELSE NULL END) AS "HP Pavillion Desktop",
COUNT(DISTINCT CASE WHEN device IN ('dell inspiron desktop') THEN
user_id ELSE NULL END) AS "Dell Inspiron Desktop",
COUNT(DISTINCT CASE WHEN device IN ('ipad air') THEN user_id ELSE
NULL END) AS "iPad Air",
COUNT(DISTINCT CASE WHEN device IN ('amazon fire phone') THEN
user_id ELSE NULL END) AS "Amazon Fire Phone",
COUNT(DISTINCT CASE WHEN device IN ('nexus 10') THEN user_id ELSE
NULL END) AS "Nexus 10"
FROM events
WHERE event_type = 'engagement'
GROUP BY 1
ORDER BY 1;
```

	Week Numbers	Dell Inspiron Notebook	iPhone 5	iPhone 4s	Windows Surface	Macbook Air	iPhone 5s	Macbook Pro	Kindle Fire	iPad Mini	Nexus 7	Nexus 5	Samsung Galaxy S4	Lenovo Thinkpad	Samsung Galaxy Tablet
▶	17	17	30	7	2	18	21	57	2	6	6	16	24	33	0
	18	31	39	18	2	50	24	112	13	13	11	26	35	78	0
	19	34	50	18	4	48	28	114	6	12	16	41	37	75	0
	20	32	42	18	7	42	28	115	6	12	8	39	27	71	0
	21	36	56	21	8	40	27	98	12	11	7	39	30	57	0
	22	41	35	17	4	57	32	92	7	10	19	40	40	70	0
	23	38	56	26	3	45	25	89	12	12	8	37	26	64	0
	24	37	61	27	13	58	29	98	12	10	19	31	44	63	0
	25	46	53	19	10	39	20	107	7	9	14	29	42	73	0
	26	34	52	23	7	51	34	104	9	10	15	29	39	68	0
	27	26	53	35	10	54	24	123	12	10	12	31	42	70	0
	28	35	49	26	12	65	28	106	11	9	12	33	55	65	0
	29	41	48	20	6	59	32	103	13	8	15	33	41	84	0
	30	48	41	26	7	43	36	105	2	8	17	36	30	82	0
	31	28	48	18	7	44	19	111	2	6	10	31	33	64	0
	32	27	43	13	4	41	24	116	6	6	9	21	25	57	0
	33	28	34	14	7	42	26	115	6	11	12	21	32	58	0

	Samsung Galaxy Tablet	Acer Aspire Notebook	Asus Chromebook	HTC One	Nokia Lumia 635	Samsung Galaxy Note	Acer Aspire Desktop	Mac Mini	HP Pavilion Desktop	Dell Inspiron Desktop	iPad Air	Amazon Fire Phone	Nexus 10
▶	0	8	10	4	6	3	4	4	0	8	7	1	4
	0	11	14	6	14	8	9	5	0	16	16	3	12
	0	20	13	15	11	8	6	9	0	12	22	5	10
	0	17	16	9	7	9	7	14	0	21	24	1	9
	0	12	19	8	4	7	16	7	0	13	18	0	13
	0	15	22	5	10	4	7	12	0	17	23	4	16
	0	18	20	3	7	5	10	10	0	19	14	4	12
	0	16	13	7	10	7	10	13	0	19	17	5	15
	0	17	18	8	10	3	9	10	0	23	26	3	12
	0	13	16	7	18	3	12	6	0	22	16	5	14
	0	17	14	6	9	7	6	5	0	17	18	2	15
	0	16	15	7	14	5	7	10	0	23	18	2	12
	0	12	18	14	20	5	8	7	0	16	19	2	11
	0	23	23	7	12	6	9	8	0	13	24	2	16
	0	20	17	3	12	5	7	7	0	15	17	1	6
	0	15	22	5	7	8	12	5	0	13	19	6	11
	0	13	17	6	6	7	14	12	0	11	11	5	9

(E) Calculate the email engagement metrics?

```

SELECT Week,
ROUND((weekly_digest/total*100),2) AS "Weekly Digest Rate",
ROUND((email_opens/total*100),2) AS "Email Open Rate",
ROUND((email_clickthroughs/total*100),2) AS "Email Click Through
Rate",
ROUND((reengagement_emails/total*100),2) AS "Re-engagement Email
Rate"
FROM (
SELECT 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 email_opens,
COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE
NULL END) AS email_clickthroughs,
COUNT(CASE WHEN action = 'sent_reengagement_email' THEN user_id
ELSE NULL END) AS reengagement_emails,
COUNT(user_id) AS total
FROM email_events
GROUP BY 1
) sub
GROUP BY 1
ORDER BY 1;

```

	Week	Weekly Digest Rate	Email Open Rate	Email Click Through Rate	Re-engagement Email Rate
▶	17	62.32	21.28	11.39	5.01
	18	63.45	22.24	10.49	3.83
	19	62.16	22.67	11.13	4.04
	20	61.62	22.64	11.43	4.31
	21	63.52	22.82	9.97	3.69
	22	63.59	21.56	10.66	4.19
	23	62.39	22.34	11.18	4.09
	24	61.61	22.92	10.99	4.48
	25	63.77	21.79	10.54	3.90
	26	62.99	22.22	10.61	4.18
	27	62.24	22.49	11.37	3.90
	28	62.92	22.48	10.77	3.83
	29	63.98	21.71	10.51	3.79
	30	62.29	23.24	10.59	3.88
	31	65.27	23.25	7.66	3.82
	32	66.59	22.85	7.14	3.42
	33	64.73	23.10	7.91	4.26
	34	64.33	23.91	7.67	4.08

Result 22

Result:

This project helps me to understand the importance of operation analytics in a company. Through this project, I can understand how companies use metric spikes as a secret weapon. They can use insights to improve their strategy and increase ROI if they take a thoughtful and proactive approach.

Challenges I faced in the project: The only issue here is that case study 2 contains a tonne of data, and SQL Workbench is having a difficult time importing all of it. Therefore, I must use LOAD DATA statements to solve this issue. The process of importing is currently being halted due to a new issue that occurs in the column with datatype int. Before I can restart the process of loading the data into the events table, I need to modify the datatype of the object.

Conclusion:

Operational Analytics tackles the problem by synchronizing real-time data. Operation analytics can combine data from various data sources into a comprehensive, organized, and actionable solution that can give analytical models in real-time to develop customer profiles and a comprehensive perspective of operations for a business. This ensures that operational procedures and processes are applied effectively. When properly applied, operational analytics can significantly improve the lives of the general public, the global community, and the general effectiveness of particular regions.