OPERATION ANALYTICS
AND
INVESTIGATING METRIC SPIKE

WHAT IS OPERATION ANALYTICS?

Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. We work closely with the ops team, support team, marketing team, etc and help them derive insights out of the data they collect.
Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company's fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.

ABOUT THIS PROJECT

This project focuses on analyzing the two case studies which are provided by the management team. My task here is to find the questions provided by the team and to find out their insights . So that these insights are then used by teams across the business.In this project i have 2 case studies from 1st case study i will talk about the number of jobs reviewed,throughput,percentage share of each language and duplicate rows.
From the 2 nd case study i will talk about the user engagement, user growth,weekly retention, email engagement and weekly engagement.

THE TOOLS AND TECHNOLOGIES

The tools and technologies that I am going to use in this project would be data provided by trainity and MySql workbench .
MySQL Workbench is widely used to handle structured data.MySQL 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. We can use MySQL to store, retrieve, manipulate and process data that is in the form of tables.

Why I am using the sql workbench?

- MySQL Workbench gives access to add and remove.
- MySQL Workbench grants and revokes privileges.
- We can modify global and database permissions on the MySQL server.

- We can change passwords using MySQL.
- We can audit to see who did what and when on the serve
- Multiple queries can be run at a time, and the result is automatically displayed.

Case Study 1 (Job Data)

In this case study I will find about the number of jobs reviewed over time , duplicate rows, percentage share of each language and the number of events happening per second.
This case study has 1 table (job_data) and below is the structure of the table with the definition of each column.

- Table-1: job_data

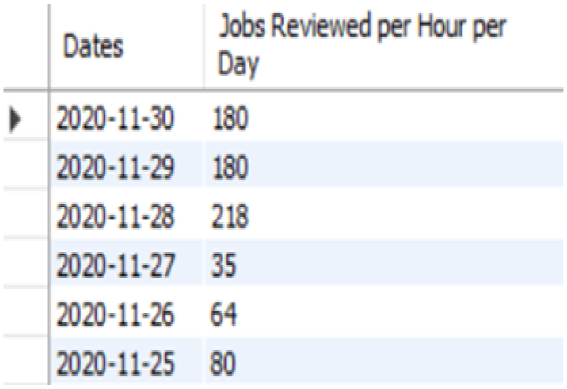
    - **job_id:** unique identifier of jobs
    - **actor_id:** unique identifier of actor
    - **event:** decision/skip/transfer
    - **language:** language of the content
    - **time_spent:** time spent to review the job in seconds
    - **org:** organization of the actor
    - **ds:** date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

INSIGHTS

Q1) 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 2020-11-28 there is a maximum number of jobs reviewed.

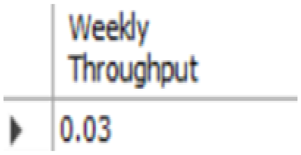


Dates	Jobs Reviewed per Hour per Day
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

Q2) Calculate 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;
```

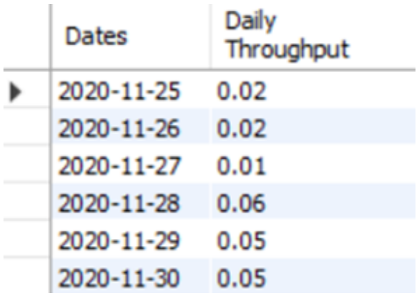
The weekly throughput is 0.03.



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;
```

On 2020-11-28 the throughput is highest.

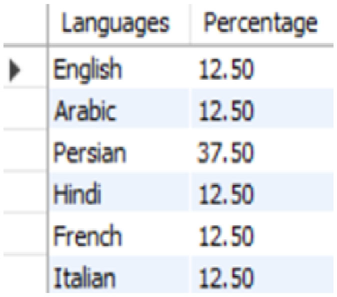


Dates	Daily Throughput
2020-11-25	0.02
2020-11-26	0.02
2020-11-27	0.01
2020-11-28	0.06
2020-11-29	0.05
2020-11-30	0.05

Q3) Calculate the percentage share of each language in the last 30 days?

```
select language ,round(100*count(*)/sum(count(*)) over (),2)
as percentage
from job_data
group by language;
```

The Persian language is the highest with 37.5% total.

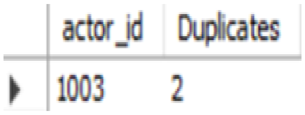


Languages	Percentage
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

Q4)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 duplicate rows.



actor_id	Duplicates
1003	2

Case Study 2 (Investigating metric spike)

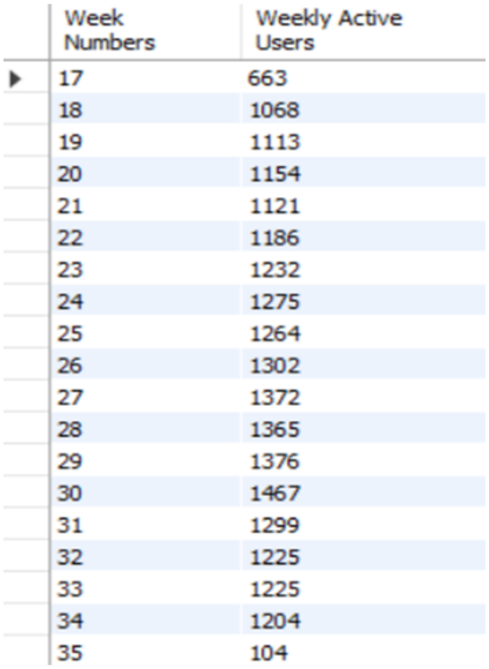
This case study I will find user engagement,user growth ,weekly retention,weekly engagement,and email engagement. This case study has 3 tables and the structure of the table are given below:

- **Table-1:** users
  This table includes one row per user, with descriptive information about that user's account.
- **Table-2:** events
  This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
- **Table-3:** email_events
  This table contains events specific to the sending of emails. It is similar in structure to the events table above.

INSIGHTS

Q1) Calculate the weekly user engagement?

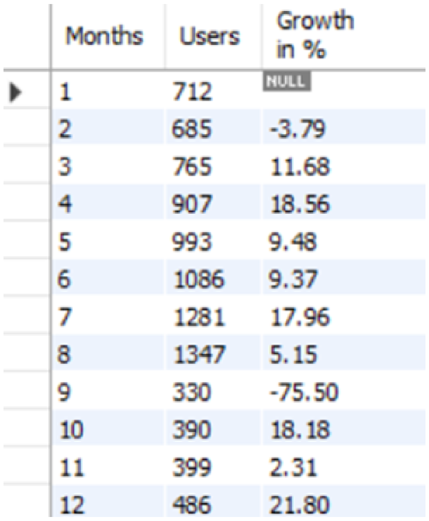
```
SELECT EXTRACT(WEEK FROM occurred_at) AS "Week Numbers", COUNT(DISTINCT
user_id) AS "Weekly Active Users"
FROM events
WHERE event_type = 'engagement'
GROUP BY 1;
```



Week Numbers	Weekly Active Users
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

Q2) calculate the user growth for the product?

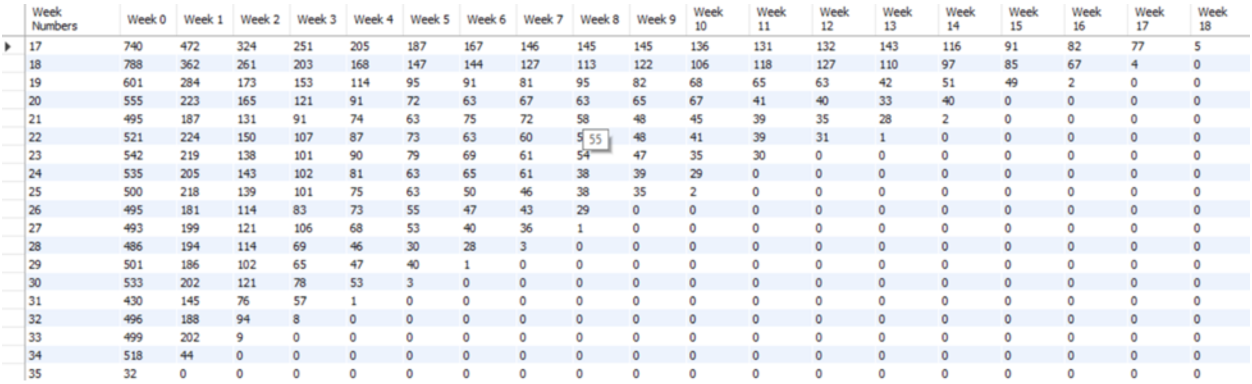
```
SELECT Months, Users, ROUND(((Users/LAG(Users, 1) OVER (ORDER BY Months) - 1)*100),
2) AS "Growth in %"
FROM
(
SELECT EXTRACT(MONTH FROM created_at) AS Months, COUNT(activated_at) AS Users
FROM users
WHERE activated_at NOT IN("")
GROUP BY 1
ORDER BY 1
) sub;
```



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

Q3) calculate the weekly retention of users-sign up cohort?

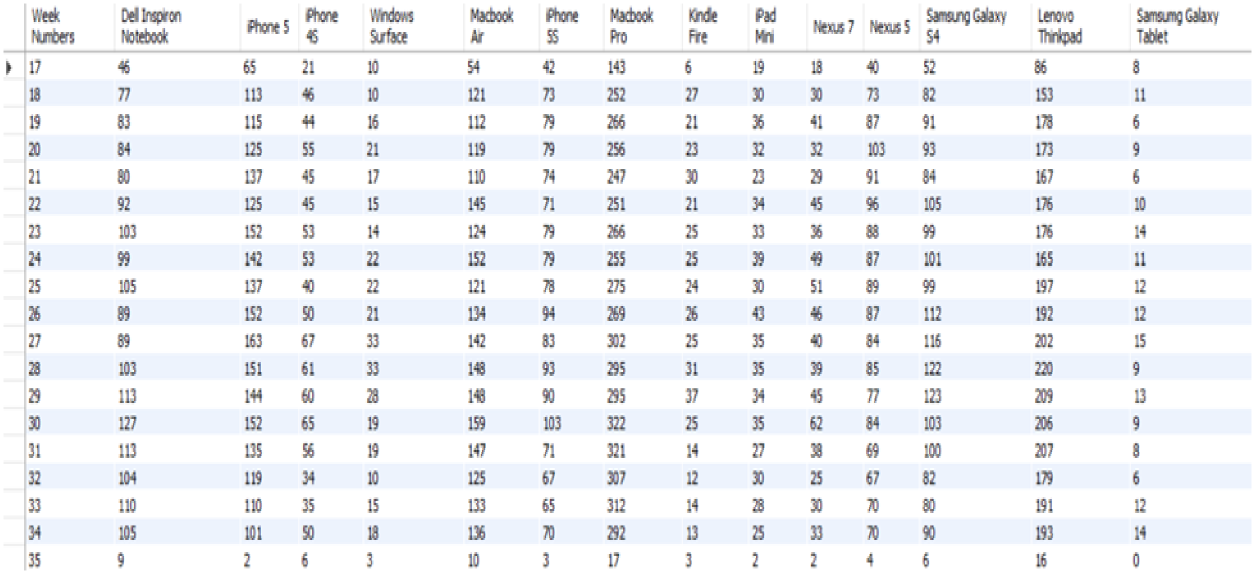
```
SELECT first AS "Week Numbers",
SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS "Week 0",
SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS "Week 1",
SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS "Week 2",
SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS "Week 3",
SUM(CASE WHEN week_number = 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 - first AS week_number
FROM
(SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS login_week FROM events
GROUP BY 1, 2) m,
(SELECT user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS first FROM events GROUP
BY 1) n
WHERE m.user_id = n.user_id
) sub
GROUP BY first
ORDER BY first;
```



Week Numbers	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
17	740	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	788	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	601	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	555	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	495	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	521	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	542	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	535	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	500	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	495	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	493	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	486	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	501	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	533	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	430	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	496	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	499	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	518	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Q4) calculate the weekly engagement per device?

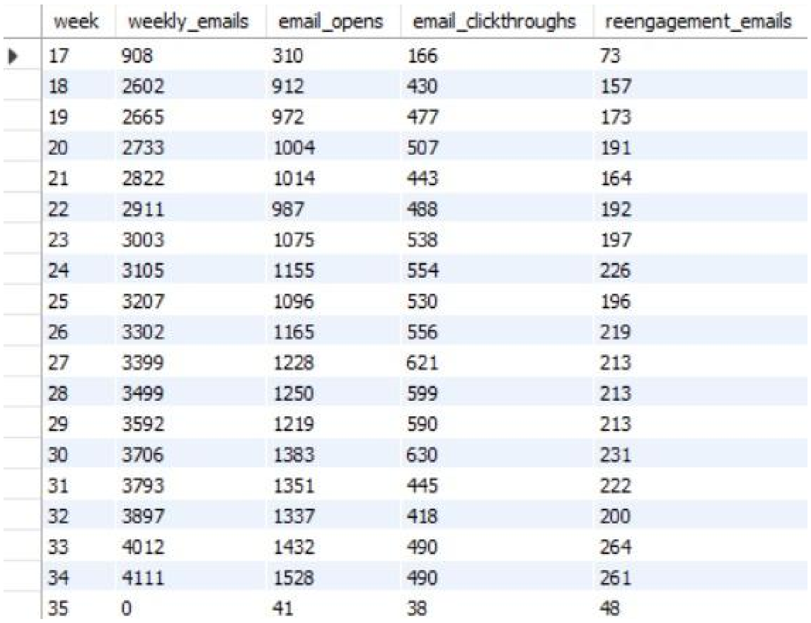
```
SELECT EXTRACT(WEEK FROM occurred_at) AS "Week Numbers",
COUNT(DISTINCT CASE WHEN device IN('dell inspiron notebook') THEN user_id ELSE NULL
END) AS "Dell Inspiron Notebook",
COUNT(DISTINCT CASE WHEN device IN('iphone 5') THEN user_id ELSE NULL END) AS
"iPhone 5",
COUNT(DISTINCT CASE WHEN device 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('samsumg galaxy tablet') THEN user_id ELSE
NULL END) AS "Samsumg 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 Pavilion Desktop",
COUNT(DISTINCT CASE WHEN device IN('dell inspiron desktop') THEN user_id ELSE NULL
END) AS "Dell Inspiron Desktop",
        SELECT EXTRACT(WEEK FROM 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('samsumg galaxy tablet') THEN user_id ELSE
NULL END) AS "Samsumg 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 Pavilion Desktop",
COUNT(DISTINCT CASE WHEN device IN('dell inspiron desktop') THEN user_id ELSE NULL END) AS "Dell Inspiron Desktop",
```



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	46	65	21	10	54	42	143	6	19	18	40	52	86	8
18	77	113	46	10	121	73	252	27	30	30	73	82	153	11
19	83	115	44	16	112	79	266	21	36	41	87	91	178	6
20	84	125	55	21	119	79	256	23	32	32	103	93	173	9
21	80	137	45	17	110	74	247	30	23	29	91	84	167	6
22	92	125	45	15	145	71	251	21	34	45	96	105	176	10
23	103	152	53	14	124	79	266	25	33	36	88	99	176	14
24	99	142	53	22	152	79	255	25	39	49	87	101	165	11
25	105	137	40	22	121	78	275	24	30	51	89	99	197	12
26	89	152	50	21	134	94	269	26	43	46	87	112	192	12
27	89	163	67	33	142	83	302	25	35	40	84	116	202	15
28	103	151	61	33	148	93	295	31	35	39	85	122	220	9
29	113	144	60	28	148	90	295	37	34	45	77	123	209	13
30	127	152	65	19	159	103	322	25	35	62	84	103	206	9
31	113	135	56	19	147	71	321	14	27	38	69	100	207	8
32	104	119	34	10	125	67	307	12	30	25	67	82	179	6
33	110	110	35	15	133	65	312	14	28	30	70	80	191	12
34	105	101	50	18	136	70	292	13	25	33	70	90	193	14
35	9	2	6	3	10	3	17	3	2	2	4	6	16	0

Q5) calculate the email engagement metrics?

```
SELECT EXTRACT(WEEK FROM occurred_at) AS week,
COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE NULL END) AS weekly_emails,
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
FROM email_events
GROUP BY 1
ORDER BY 1;
```



week	weekly_emails	email_opens	email_clickthroughs	reengagement_emails
17	908	310	166	73
18	2602	912	430	157
19	2665	972	477	173
20	2733	1004	507	191
21	2822	1014	443	164
22	2911	987	488	192
23	3003	1075	538	197
24	3105	1155	554	226
25	3207	1096	530	196
26	3302	1165	556	219
27	3399	1228	621	213
28	3499	1250	599	213
29	3592	1219	590	213
30	3706	1383	630	231
31	3793	1351	445	222
32	3897	1337	418	200
33	4012	1432	490	264
34	4111	1528	490	261
35	0	41	38	48

RESULT:

How this project helps me:This project helps me to understand the importance of operation analysis. Through this project I am able to understand how the companies use operation analysis as a secret weapon. With an informed and proactive approach, they can leverage Insights to make data-backed decisions that optimize their Instagram strategy and boost ROI.

Challenges that I faced in this project : Personally I feel that it took me a while to get really comfortable with joins and honestly I would probably take a while to come up with these answers. The challenge here is that the data in case study 2 is very huge, as the huge amount of data SQL Workbench is very slow to import. To tackle this situation I had used LOAD DATA statements.

Conclusion: Operational analytics focus on combining process design capabilities, business analysis, and data analysis to streamline your operations while improving productivity and reducing costs. It also facilitates better data-driven decision-making for the proper day-to-day management of your establishment.