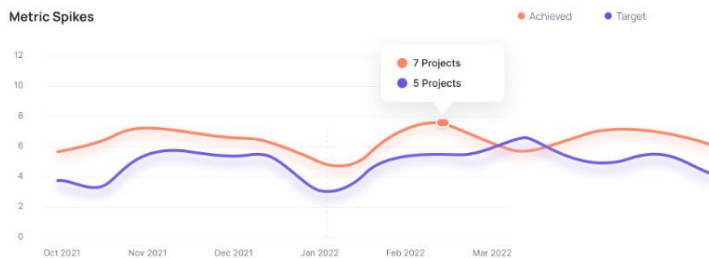


OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE.

Advanced SQL



Analysis



Employees

Aug 25-Sept 25

Inactive

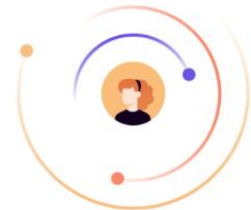
254

Active

3000

Total

3254



INTRODUCTION

This project focuses on analyzing the data which is provided by company. My task is to derive insights and answer the questions asked by different departments. So that these insights are then used by ops team, support team, marketing team, etc 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

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

WHAT IS OPERATION ANALYTICS?

Operational analytics focuses on measuring the existing and real-time operations of the company so that the company can monitor their day-to-day operations basis which they can take the necessary actions to improve customer satisfaction and bottom line.

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. 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.

Here are few examples:

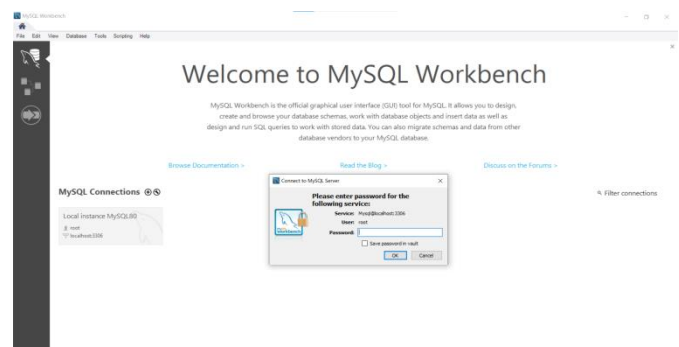
- **Ops:** Developers can use real-time data to look at how customers are using their products and make changes on the fly.
- **Marketing:** Businesses can optimize user engagement in real-time by using operational analytics to make personalized recommendations.

PROJECT APPROACH

This project is developed using SQL Workbench. First I need to create database by using dataset file which was provided by the company. Next step load the data into SQL Workbench then performed analysis and find the information that will help the ops team, support team, marketing team, etc to 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.

TECH-STACK USED

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



MySQL Workbench 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.

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.

KEY FUNCTIONALITIES OF MYSQL WORKBENCH

- **Visual SQL Editor:** MySQL Workbench is equipped with a visual SQL editor where developers can build, edit, and run queries. What's great about this is that it allows you to preview your changes before applying them.
- **Database Administration:** Aside from providing you with SQL editing tools, MySQL Workbench also comes with a database administration suite. This makes it easy for you to audit your databases, configure servers, and view logs.
- **Performance Monitoring:** MySQL Workbench gives users a dashboard where they can view the status of their queries, client timing, network latency, and index usage. This allows for simpler identification of possible ways to optimize SQL performance.

INSIGHTS :



CASE STUDY 1: JOB DATA ANALYSIS

A.) Jobs Reviewed Over Time:

Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

QUERY:

```
select ds, count(job_id) as jobs_per_day, sum(time_spent)/3600 as hours_spent  
from job_data  
where ds>='11/01/2020' and ds<='11/30/2020'  
group by ds;
```

Result Grid			 Filter Rows:	<input type="text"/>
	ds	jobs_per_day	hours_spent	
▶	11/30/2020	2	0.0111	
	11/29/2020	1	0.0056	
	11/28/2020	2	0.0092	
	11/27/2020	1	0.0289	
	11/26/2020	1	0.0156	
	11/25/2020	1	0.0125	


In this I have used sum and comparison operators in the query to get the result.

B.) Throughput Analysis:

Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

QUERY:

```
select round(count(event)/sum(time_spent),2) as "Weekly Throughput"  
from job_data;
```

Result Grid			Filter Rows:
	Weekly Throughput		
	0.03		

Here, the weekly throughput is 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 screenshot shows a 'Result Grid' with a 'Filter Rows' button. The table has two columns: 'dates' and 'Daily Throughput'. The data is as follows:

dates	Daily Throughput
11/25/2020	0.02
11/26/2020	0.02
11/27/2020	0.01
11/28/2020	0.06
11/29/2020	0.05
11/30/2020	0.05

On date 11/28/2020 the throughput is highest 0.06.

Metrics will always go up and down on a weekly and daily basis. You'll get members faster every day or minute if you want. As a result, rolling metrics are superb at showing if your metrics are trending up or down on a daily level.

C.) Language Share Analysis:

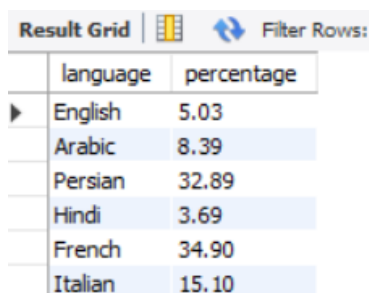
Write an SQL query to calculate the percentage share of each language over the last 30 days.

QUERY:

select language, round((sum(time_spent)/(select sum(time_spent) from job_data)) * 100 , 2) as percentage

from job_data

group by language;



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The table has two columns: 'language' and 'percentage'. The data is as follows:

language	percentage
English	5.03
Arabic	8.39
Persian	32.89
Hindi	3.69
French	34.90
Italian	15.10

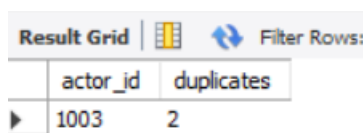
French language has highest percentage share with 34.90% total over the last 30 days.

D.) Duplicate Rows Detection:

Write an SQL query to display duplicate rows from the job_data table.

QUERY:

```
select actor_id, count(*) as duplicates
from job_data
group by actor_id
having count(*)>1;
```



actor_id	duplicates
1003	2

Actor ID 1003 has duplicates rows.

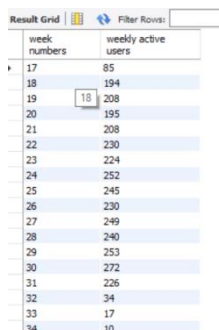
CASE STUDY 2: INVESTIGATING METRIC SPIKE

A.) Weekly User Engagement:

Write an SQL query to calculate the weekly user engagement.

QUERY:

```
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	85
18	194
19	208
20	195
21	208
22	230
23	224
24	252
25	245
26	230
27	249
28	240
29	253
30	272
31	226
32	34
33	17
34	10

B.) User Growth Analysis:

Write an SQL query to calculate the user growth for the product.

QUERY:

```
select registration_month, new_users, cumulative_users,  
  
case when lag(cumulative_users) over (order by registration_month) = 0 then null  
  
else round((new_users / lag(cumulative_users) over (order by registration_month)) * 100,  
2)  
  
end as growth_in_percentage  
  
from (  
  
select date_format(created_at, '%y-%m') as registration_month, count(distinct user_id) as  
new_users,  
  
@cumulative_users := @cumulative_users + count(distinct user_id) as cumulative_users  
  
from users  
  
cross join (select @cumulative_users := 0) as init  
  
group by registration_month ) growth_data  
  
order by registration_month;
```

Result Grid				
		Filter Rows:	Export:	Wrap Cell Content:
	registration_month	new_users	cumulative_users	growth_in_percentage
▶	2013-01	160	160	NULL
	2013-02	160	320	100
	2013-03	150	470	46.88
	2013-04	181	651	38.51
	2013-05	214	865	32.87
	2013-06	213	1078	24.62
	2013-07	284	1362	26.35
	2013-08	316	1678	23.2
	2013-09	330	2008	19.67
	2013-10	390	2398	19.42
	2013-11	399	2797	16.64
	2013-12	486	3283	17.38
	2014-01	552	3835	16.81
	2014-02	525	4360	13.69
	2014-03	615	4975	14.11
	2014-04	726	5701	14.59
	2014-05	779	6480	13.66
	2014-06	873	7353	13.47
	2014-07	997	8350	13.56
	2014-08	1031	9381	12.35

C.) Weekly Retention Analysis:

Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

QUERY:

```
select cohort_week, registration_week, count(distinct user_id) as initial_users,
sum(week_1_activity) as week_1_retention, sum(week_2_activity) as week_2_retention,
sum(week_3_activity) as week_3_retention, sum(week_4_activity) as week_4_retention
from ( select user_id, date_format(created_at, '%x-%v') as cohort_week,
week(created_at) as registration_week,
max(date(activated_at) <= date_add(created_at, interval 1 week)) as week_1_activity,
max(date(activated_at) <= date_add(created_at, interval 2 week)) as week_2_activity,
max(date(activated_at) <= date_add(created_at, interval 3 week)) as week_3_activity,
max(date(activated_at) <= date_add(created_at, interval 4 week)) as week_4_activity
from users
group by cohort_week, registration_week, user_id ) cohort_data
group by cohort_week, registration_week
order by cohort_week, registration_week;
```

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2013-01	0	23	23	23	23	23
2013-01	1	3	3	3	3	3
2013-02	1	27	27	27	27	27
2013-02	2	2	2	2	2	2
2013-03	2	46	46	46	46	46
2013-03	3	1	1	1	1	1
2013-04	3	35	35	35	35	35
2013-04	4	1	1	1	1	1
2013-05	4	29	29	29	29	29
2013-05	5	1	1	1	1	1
2013-06	5	47	47	47	47	47
2013-06	6	1	1	1	1	1
2013-07	6	37	37	37	37	37
2013-07	7	4	4	4	4	4
2013-08	7	38	38	38	38	38
2013-08	8	1	1	1	1	1
2013-09	8	33	33	33	33	33
2013-10	9	43	43	43	43	43
2013-11	10	32	32	32	32	32
2013-11	11	1	1	1	1	1
2013-12	11	30	30	30	30	30

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2013-23	22	50	50	50	50	50
2013-23	23	1	1	1	1	1
2013-24	23	49	49	49	49	49
2013-24	24	2	2	2	2	2
2013-25	24	43	43	43	43	43
2013-25	25	3	3	3	3	3
2013-26	25	54	54	54	54	54
2013-26	26	3	3	3	3	3
2013-27	26	53	53	53	53	53
2013-27	27	4	4	4	4	4
2013-28	27	48	48	48	48	48
2013-28	28	4	4	4	4	4
2013-29	28	68	68	68	68	68
2013-29	29	3	3	3	3	3
2013-30	29	64	64	64	64	64
2013-30	30	2	2	2	2	2
2013-31	30	65	65	65	65	65
2013-31	31	4	4	4	4	4
2013-32	31	63	63	63	63	63
2013-32	32	3	3	3	3	3
2013-33	32	68	68	68	68	68

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2013-12	12	2	2	2	2	2
2013-13	12	31	31	31	31	31
2013-13	13	2	2	2	2	2
2013-14	13	37	37	37	37	37
2013-14	14	3	3	3	3	3
2013-15	14	32	32	32	32	32
2013-15	15	3	3	3	3	3
2013-16	15	40	40	40	40	40
2013-16	16	2	2	2	2	2
2013-17	16	44	44	44	44	44
2013-17	17	4	4	4	4	4
2013-18	17	45	45	45	45	45
2013-18	18	3	3	3	3	3
2013-19	18	41	41	41	41	41
2013-19	19	4	4	4	4	4
2013-20	19	53	53	53	53	53
2013-20	20	2	2	2	2	2
2013-21	20	37	37	37	37	37
2013-21	21	4	4	4	4	4
2013-22	21	45	45	45	45	45
2013-22	22	4	4	4	4	4

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2013-33	33	5	5	5	5	5
2013-34	33	68	68	68	68	68
2013-34	34	3	3	3	3	3
2013-35	34	75	75	75	75	75
2013-35	35	4	4	4	4	4
2013-36	35	59	59	59	59	59
2013-36	36	6	6	6	6	6
2013-37	36	66	66	66	66	66
2013-37	37	5	5	5	5	5
2013-38	37	80	80	80	80	80
2013-38	38	4	4	4	4	4
2013-39	38	86	86	86	86	86
2013-39	39	6	6	6	6	6
2013-40	39	78	78	78	78	78
2013-40	40	3	3	3	3	3
2013-41	40	84	84	84	84	84
2013-41	41	4	4	4	4	4
2013-42	41	69	69	69	69	69
2013-42	42	5	5	5	5	5
2013-43	42	94	94	94	94	94
2013-43	43	3	3	3	3	3

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2013-44	43	86	86	86	86	86
2013-44	44	6	6	6	6	6
2013-45	44	90	90	90	90	90
2013-45	45	7	7	7	7	7
2013-46	45	84	84	84	84	84
2013-46	46	10	10	10	10	10
2013-47	46	78	78	78	78	78
2013-47	47	4	4	4	4	4
2013-48	47	98	98	98	98	98
2013-48	48	5	5	5	5	5
2013-49	48	92	92	92	92	92
2013-49	49	4	4	4	4	4
2013-50	49	112	112	112	112	112
2013-50	50	5	5	5	5	5
2013-51	50	119	119	119	119	119
2013-51	51	4	4	4	4	4
2013-52	51	98	98	98	98	98
2013-52	52	6	6	6	6	6
2014-01	0	83	83	83	83	83
2014-01	1	8	8	8	8	8
2014-01	52	41	41	41	41	41

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2014-12	12	8	8	8	8	8
2014-13	12	140	140	140	140	140
2014-13	13	11	11	11	11	11
2014-14	13	156	156	156	156	156
2014-14	14	5	5	5	5	5
2014-15	14	157	157	157	157	157
2014-15	15	9	9	9	9	9
2014-16	15	155	155	155	155	155
2014-16	16	10	10	10	10	10
2014-17	16	169	169	169	169	169
2014-17	17	7	7	7	7	7
2014-18	17	163	163	163	163	163
2014-18	18	9	9	9	9	9
2014-19	18	154	154	154	154	154
2014-19	19	6	6	6	6	6
2014-20	19	179	179	179	179	179
2014-20	20	7	7	7	7	7
2014-21	20	169	169	169	169	169
2014-21	21	8	8	8	8	8
2014-22	21	175	175	175	175	175
2014-22	22	11	11	11	11	11

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2014-02	1	118	118	118	118	118
2014-02	2	4	4	4	4	4
2014-03	2	105	105	105	105	105
2014-03	3	7	7	7	7	7
2014-04	3	106	106	106	106	106
2014-04	4	7	7	7	7	7
2014-05	4	123	123	123	123	123
2014-05	5	7	7	7	7	7
2014-06	5	126	126	126	126	126
2014-06	6	6	6	6	6	6
2014-07	6	129	129	129	129	129
2014-07	7	6	6	6	6	6
2014-08	7	119	119	119	119	119
2014-08	8	8	8	8	8	8
2014-09	8	121	121	121	121	121
2014-09	9	6	6	6	6	6
2014-10	9	127	127	127	127	127
2014-10	10	8	8	8	8	8
2014-11	10	146	146	146	146	146
2014-11	11	6	6	6	6	6
2014-12	11	124	124	124	124	124

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2014-23	22	185	185	185	185	185
2014-23	23	12	12	12	12	12
2014-24	23	184	184	184	184	184
2014-24	24	14	14	14	14	14
2014-25	24	215	215	215	215	215
2014-25	25	7	7	7	7	7
2014-26	25	200	200	200	200	200
2014-26	26	10	10	10	10	10
2014-27	26	191	191	191	191	191
2014-27	27	8	8	8	8	8
2014-28	27	214	214	214	214	214
2014-28	28	9	9	9	9	9
2014-29	28	206	206	206	206	206
2014-29	29	9	9	9	9	9
2014-30	29	212	212	212	212	212
2014-30	30	16	16	16	16	16
2014-31	30	222	222	222	222	222
2014-31	31	12	12	12	12	12
2014-32	31	181	181	181	181	181

cohort_week	registration_week	initial_users	week_1_retention	week_2_retention	week_3_retention	week_4_retention
2014-32	32	8	8	8	8	8
2014-33	32	237	237	237	237	237
2014-33	33	13	13	13	13	13
2014-34	33	248	248	248	248	248
2014-34	34	11	11	11	11	11
2014-35	34	248	248	248	248	248
2014-35	35	18	18	18	18	18

D.) Weekly Engagement Per Device:

4. Write an SQL query to calculate the weekly engagement per device.

QUERY:

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('mackbook_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) "mackbook 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 pavilion 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;

Result Grid		 Filter Rows:		 Export:		 Wrap Cell Content:																				
Week Numbers	Dell Inspiro Notebk	iPhoi 5	iPhoi 4s	Windo Surfac	Macb Air	iPhor 5S	Macbk Pro	Kind Fire	iPad Mini	Nexus 7	Nexus 5	Samsu Galaxy S4	Lenovo Thinkpae	Samsu Galaxy Tablet	Acer Aspi Note	Asus Chron	HTC One	Nokia Lumia 635	Samsung Galaxy Note	Acer Aspire Deskt	Mac Mini	HP Pavli Desk	Dell Inspi Deskt	iPad Air	Amazor Fire Phone	Nexus 10
17	4	11	3	0	0	5	13	0	3	4	4	7	8	0	2	3	2	2	1	2	1	2	1	1	1	0
18	12	7	4	1	0	8	41	5	7	5	9	12	28	0	4	4	2	4	4	4	1	6	3	8	2	2
19	7	19	7	3	0	9	37	4	1	7	15	14	21	0	8	3	6	2	4	0	2	5	1	10	4	3
20	9	16	5	2	0	13	38	2	6	3	9	9	22	0	4	6	3	6	5	2	2	2	11	9	1	3
21	12	24	9	3	0	10	33	3	4	3	16	9	22	0	4	9	4	1	4	6	1	7	4	8	0	5
22	16	10	9	2	0	13	33	3	4	7	14	11	29	0	6	9	2	5	0	0	2	4	9	9	2	8
23	11	19	8	2	0	13	34	5	6	5	8	11	26	0	8	10	1	2	0	5	2	11	10	5	1	4
24	11	24	7	7	0	12	41	4	5	9	10	18	24	0	5	8	4	1	1	5	2	7	8	8	2	6
25	21	19	3	5	0	7	42	2	3	7	8	18	30	0	7	8	4	8	0	3	3	5	11	11	2	4
26	11	25	8	5	0	17	40	2	4	8	7	11	25	0	6	8	2	8	2	5	3	7	7	6	2	3
27	8	26	16	5	0	14	52	4	2	9	11	16	24	0	7	7	3	2	2	4	3	6	6	5	2	6
28	16	18	7	5	0	10	40	6	4	5	10	21	24	0	8	5	2	4	2	2	4	7	9	10	1	5
29	15	22	8	3	0	12	41	7	7	6	10	16	32	0	1	5	4	10	2	3	2	5	8	9	2	4
30	16	24	11	2	0	16	36	1	5	6	19	16	37	0	10	9	1	7	3	6	6	8	7	10	2	8
31	13	28	7	4	0	9	47	1	3	4	13	18	22	0	8	6	1	6	2	3	1	4	7	11	0	3
32	2	5	1	0	0	3	9	1	0	0	3	3	4	0	1	1	0	0	1	0	0	0	1	0	1	0
33	0	0	1	0	0	1	2	0	1	1	1	0	0	0	1	1	0	0	1	3	0	1	1	1	0	0
34	1	1	1	0	0	2	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0

E.) Email Engagement Analysis:

5. Write an SQL query to calculate the email engagement metrics.

QUERY:

select week, round((weekly_digest/total*100),2) as "weekly digest rane",

round((email_opens/total* 100),2) as "email open rate",

round((email_clickthroughs/total* 100),2) as "email clickthrough rate",

round((reengagement_emails/total* 100),2) as "reengagement email rate"

from (select extract(week from 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;

Result Grid					
			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Week	Weekly Digest Rate	Email Open Rate	Email Clickthrough Rate	Reengagement 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
	35	0.00	32.28	29.92	37.80

RESULT

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

Challenges that I faced in this project: 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 it. To tackle this situation I have to use LOAD DATA statements. Now, there is another problem arises in the column user_type in events table that has datatype int which is stopping the process of importing. First I need to change its datatype to text then restart the process of loading the data into events table.

CONCLUSION:

Operational Analytics tackles the problem by synchronizing real time data. Operational Analytics has the capability to aggregate data from multiple data sources into a cumulative, organized, actionable solution capable of delivering analytical models in real-time to create individual customer profiles and a holistic view of operations for a company. This guarantees that your operational routines and systems are used efficiently. Whenever utilized correctly, operational analytics can achieve a significant positive effect on our general public and world everywhere and increment the general efficiency of specific areas.