PROJECT 3

Operation Analytics and Investigating Metric Spike.

Project Description:

The company provided with 2 Datasets . In dataset 1 we have been given job_data and in dataset 2 it consists of users data ,email_events data , events data .It is the real-time data where we need to find meaningful trends and insights from the data .We were provided with Operational Analytics. Operational Analytics is a crucial process that involves analyzing a company's end-to-end operations. This analysis helps identify areas for improvement within the company.

Operational analytics is about using real-time data for daily decisions. Relevant business information flows from many sources into tools that analyze that data and identify problems and opportunities. This actionable data is then used by teams to inform their decision-making. companies must keep up with rapid changes in the business landscape. Operational analytics makes insights available in near-real-time for staff who need to address support tickets, repair remote equipment, or adjust pricing or sales tactics.

Operational analytics focuses on immediate action.

Keys

- Operational analytics enables companies to keep pace with rapid changes in the business landscape.
- Data from multiple sources flows into systems that provide actionable insights on the fly.
- The benefits of operational analytics include smoother operations, lower costs, better products and services and ultimately happier customers.
- You can use operational analytics platforms in many fields and functions, such as agile development, customer support and predictive maintenance.

Approach:

-We were assigned with 2 Datasets. One dataset consist of Job_Data.csv file and another dataset consist of user.csv ,events.csv and email_event.csv files.

- I first filtered the data , and formatted the date time which was there in the file ,then created a database and then imported the file in MYSQL WORKBENCH 8.0.

- I worked on each task /question provided and analysed basically on it , Then formatted and analyszed all the data and insights which are provided below.

-It made some challenges with the questions to understand thoroughly, as it contains huge data in data set 2.

-It was most challenging in figuring out the logics used in it and also the accurate solutions for it .

-It helped me to work on real-time data .

-This was the very first time I was working with huge dataset and analysing all the possible outcomes and then working on the solutions accurately.

Tech Stack:

MYSQL WORKBENCH 8.0 CE

INSIGHTS:

<u>Case Study 1: Job Data Analysis</u>

Tasks:

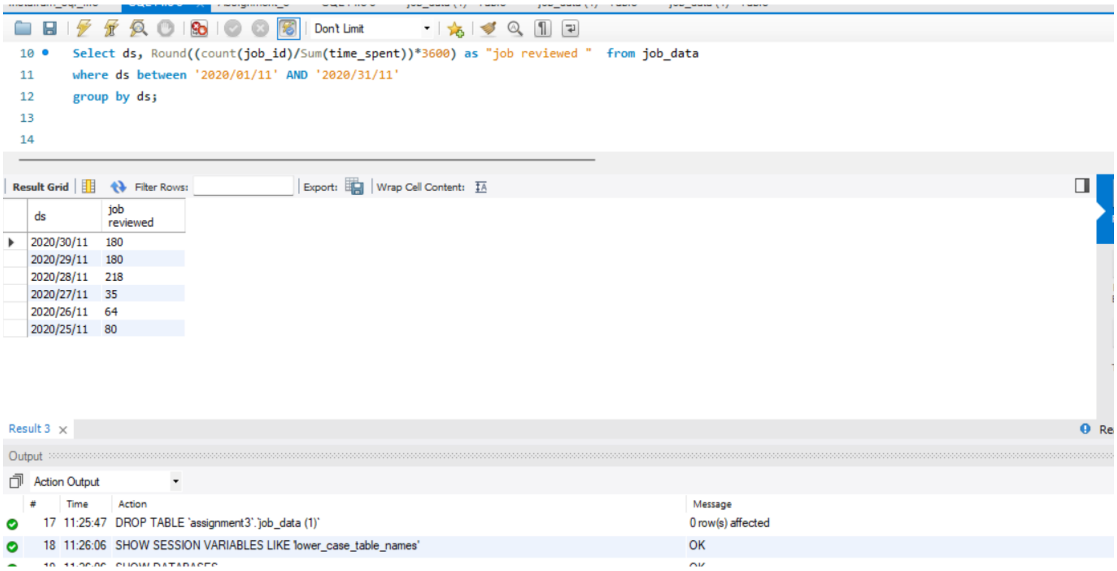
A. **Jobs Reviewed Over Time:**

- Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.
- Your Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

Answer: **Select ds, Round((count(job_id)/Sum(time_spent))*3600) as "job reviewed " from job_data**

where ds between '2020/01/11' AND '2020/31/11'

group by ds;



```
Select ds, Round((count(job_id)/Sum(time_spent))*3600) as "job reviewed "  from job_data
where ds between '2020/01/11' AND '2020/31/11'
group by ds;
```

ds	job reviewed
2020/30/11	180
2020/29/11	180
2020/28/11	218
2020/27/11	35
2020/26/11	64
2020/25/11	80

```
#   Time      Action                                                   Message
17  11:25:47  DROP TABLE `assignment3`.`job_data (1)`                  0 row(s) affected
18  11:26:06  SHOW SESSION VARIABLES LIKE 'lower_case_table_names'     OK
```

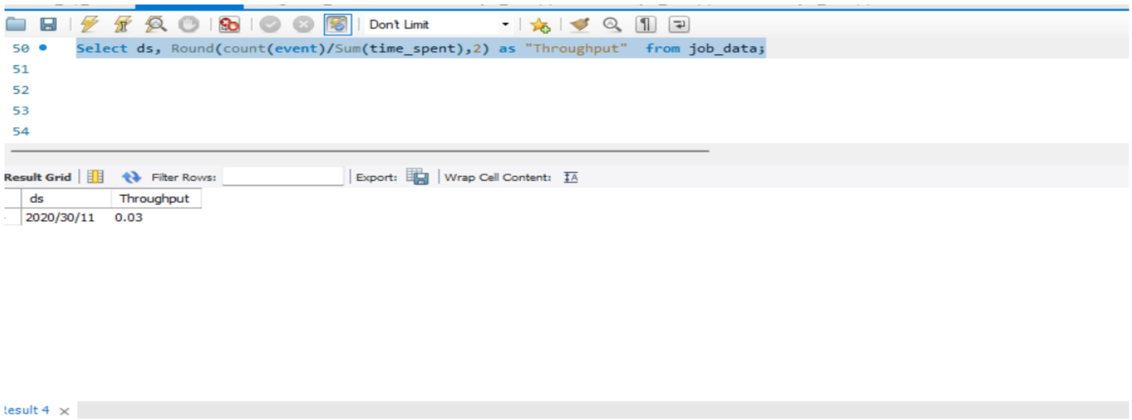
As a result, we can see that on 28th November 2020 , the job review was the highest i.e 218.

B.

Throughput Analysis:

- Objective: Calculate the 7-day rolling average of throughput (number of events per second).
- Your Task: 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.

Answer: **Select ds, Round(count(event)/Sum(time_spent),2) as "Throughput" from job_data;**



```
Select ds, Round(count(event)/Sum(time_spent),2) as "Throughput"  from job_data;
```

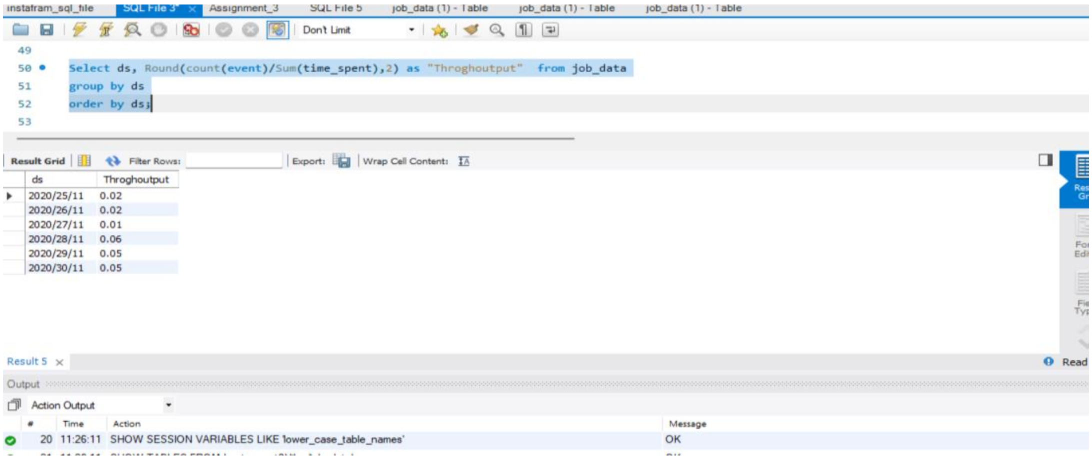
ds	Throughput
2020/30/11	0.03

It is a weekly throughoutput.

Select ds, Round(count(event)/Sum(time_spent),2) as "Throghoutput" from job_data

group by ds

order by ds;



```
Select ds, Round(count(event)/Sum(time_spent),2) as "Throghoutput"  from job_data
group by ds
order by ds;
```

ds	Throghoutput
2020/25/11	0.02
2020/26/11	0.02
2020/27/11	0.01
2020/28/11	0.06
2020/29/11	0.05
2020/30/11	0.05

#	Time	Action	Message
20	11:26:11	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK

The maximum throughoutput is 0.06.

Metrics will go up and down on daily basis or a weekly basis. So , we will get many numbers faster everyday or every minute. Hence rolling metric will be better for this type of count because it calculates trends over short periods of time using a set of data.

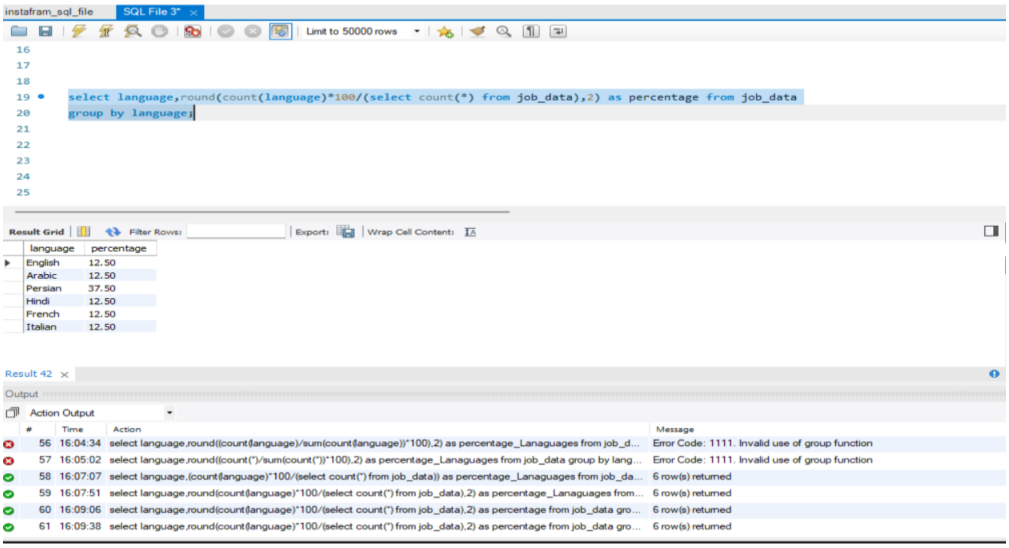
C. **Language Share Analysis:**

- Objective: Calculate the percentage share of each language in the last 30 days.
- Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

Answer: **select language,round(count(language)*100/(select count(*) from job_data),2) as percentage**

from job_data

group by language;



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

language	percentage
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

```
#   Time      Action                                                                                                 Message
56  16:04:34  select language,round((count(language)/sum(count(language))*100),2) as percentage_Lanaguages from job_d...  Error Code: 1111. Invalid use of group function
57  16:05:02  select language,round((count(*)/sum(count(*))*100),2) as percentage_Lanaguages from job_data group by lang...  Error Code: 1111. Invalid use of group function
58  16:07:07  select language,(count(language)*100/(select count(*) from job_data)) as percentage_Lanaguages from job_da...  6 row(s) returned
59  16:07:51  select language,round(count(language)*100/(select count(*) from job_data),2) as percentage_Lanaguages from...  6 row(s) returned
60  16:09:06  select language,round(count(language)*100/(select count(*) from job_data),2) as percentage from job_data gro...  6 row(s) returned
61  16:09:38  select language,round(count(language)*100/(select count(*) from job_data),2) as percentage from job_data gro...  6 row(s) returned
```

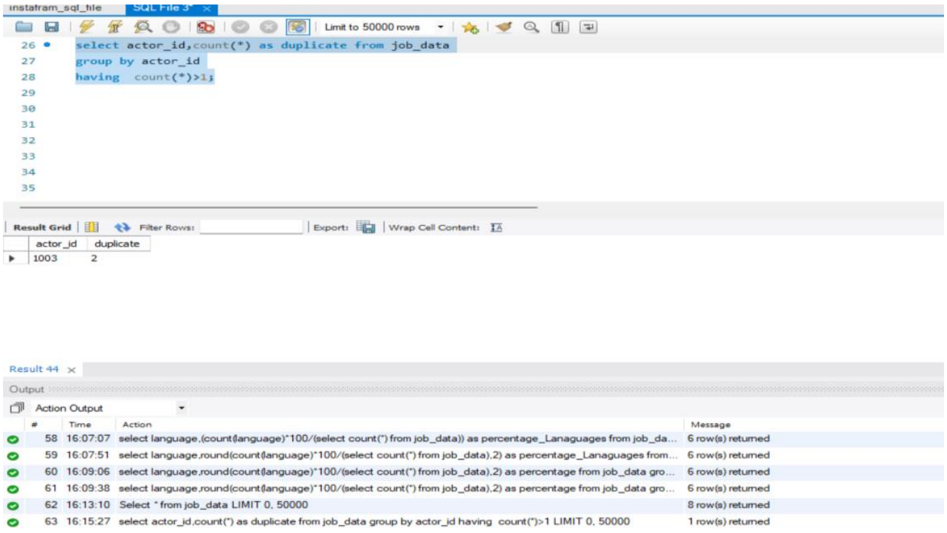
Persian Language is the highest ,it is 37%, rounded 38%.

D. **Duplicate Rows Detection:**
  - Objective: Identify duplicate rows in the data.
  - Your Task: Write an SQL query to display duplicate rows from the job_data table.

Answer: **select actor_id,count(*) as duplicate from job_data**

group by actor_id

having count(*)>1;



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

actor_id	duplicate
1003	2

```
#   Time      Action                                                                                                 Message
58  16:07:07  select language,(count(language)*100/(select count(*) from job_data)) as percentage_Lanaguages from job_da...  6 row(s) returned
59  16:07:51  select language,round(count(language)*100/(select count(*) from job_data),2) as percentage_Lanaguages from...  6 row(s) returned
60  16:09:06  select language,round(count(language)*100/(select count(*) from job_data),2) as percentage from job_data gro...  6 row(s) returned
61  16:09:38  select language,round(count(language)*100/(select count(*) from job_data),2) as percentage from job_data gro...  6 row(s) returned
62  16:13:10  Select * from job_data LIMIT 0, 50000                                                                        8 row(s) returned
63  16:15:27  select actor_id,count(*) as duplicate from job_data group by actor_id having  count(*)>1 LIMIT 0, 50000       1 row(s) returned
```

There are 2 duplicates for the ACTOR_Id= 1003.

Case Study 2: Investigating Metric Spike

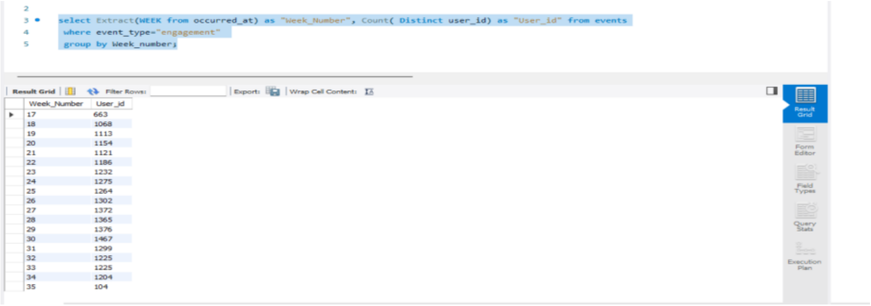
Tasks:

A. **Weekly User Engagement:**
  - Objective: Measure the activeness of users on a weekly basis.
  - Your Task: Write an SQL query to calculate the weekly user engagement.

Answer: **Select Extract(WEEK from occurred_at) as "Week_Number", Count(Distinct user_id) as "User_id" from events**

where event_type="engagement"

group by Week_number;



```
select Extract(WEEK from occurred_at) as "Week_Number", Count( Distinct user_id) as "User_id" from events
 where event_type="engagement"
 group by Week_number;
```

Week_Number	User_id
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

B. **User Growth Analysis:**

- Objective: Analyze the growth of users over time for a product.
- Your Task: Write an SQL query to calculate the user growth for the product.

Answer: **Select months,Num_of_Activated_Users,Round(((Num_of_Activated_Users/LAG(Num_of_Activated_Users,1) OVER (ORDER BY MONTHS)-1)*100),2) as "growth in %"**

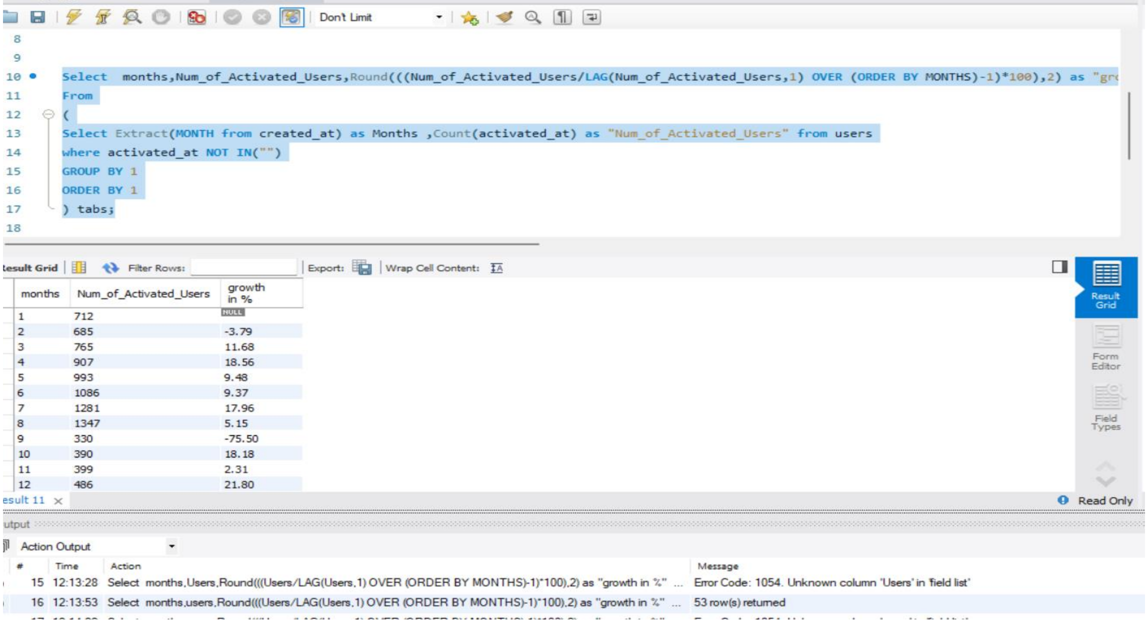
From

(Select Extract(MONTH from created_at) as Months ,Count(activated_at) as "Num_of_Activated_Users" from users where activated_at NOT IN("")

GROUP BY

ORDER BY 1)

tabs;



```
Select  months,Num_of_Activated_Users,Round(((Num_of_Activated_Users/LAG(Num_of_Activated_Users,1) OVER (ORDER BY MONTHS)-1)*100),2) as "gro
From
(
Select Extract(MONTH from created_at) as Months ,Count(activated_at) as "Num_of_Activated_Users" from users
where activated_at NOT IN("")
GROUP BY 1
ORDER BY 1
) tabs;
```

months	Num_of_Activated_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

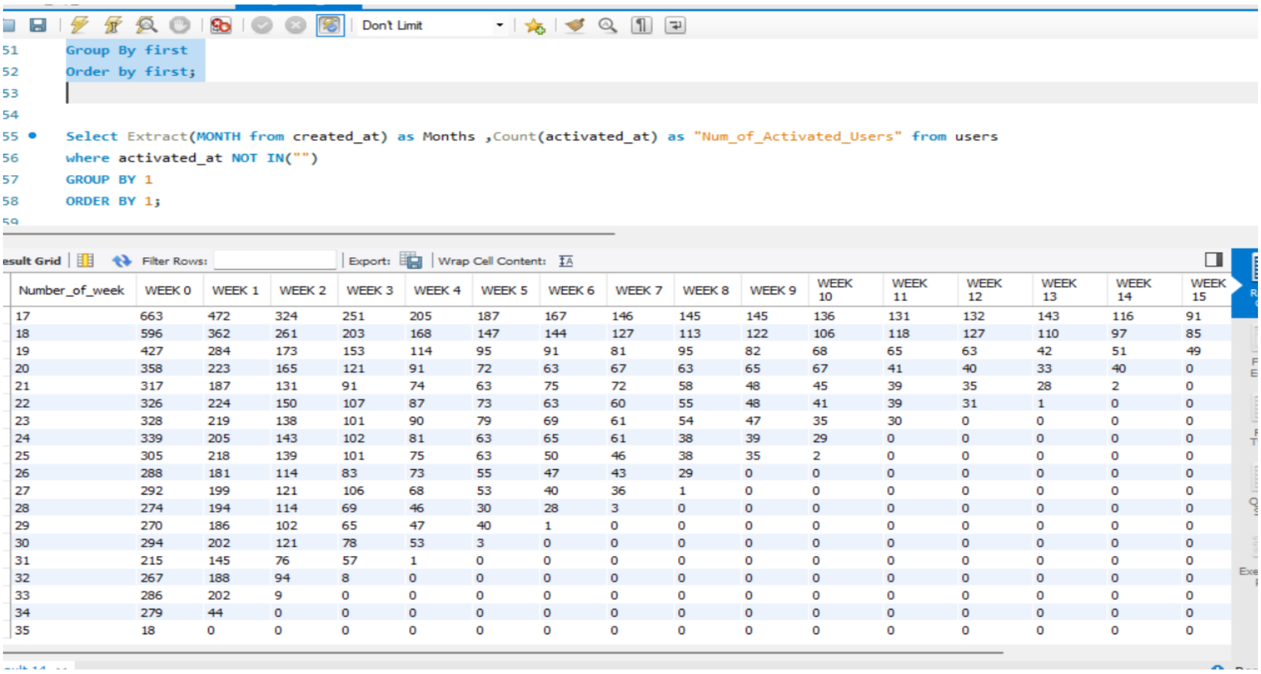
#	Time	Action	Message
15	12:13:28	Select months,Users,Round(((Users/LAG(Users,1) OVER (ORDER BY MONTHS)-1)*100),2) as "growth in %" ...	Error Code: 1054. Unknown column 'Users' in 'field list'
16	12:13:53	Select months,users,Round(((Users/LAG(Users,1) OVER (ORDER BY MONTHS)-1)*100),2) as "growth in %" ...	53 row(s) returned

C. **Weekly Retention Analysis:**

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

Answer:

```
Select first as "Number_of_week",
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 event1.user_id,event1.login_week,event2.first,event1.login_week-first AS Week_number
FROM
(
Select user_id, EXTRACT(WEEK FROM occurred_at) AS login_week from events
Group by user_id,login_week)event1,
(Select user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS First from events
Group by user_id)event2
Where event1.user_id=event2.user_id
)tabs
Group By first
 Order by first;
```



```
Group By first
Order by first;


Select Extract(MONTH from created_at) as Months ,Count(activated_at) as "Num_of_Activated_Users" from users
where activated_at NOT IN("")
GROUP BY 1
ORDER BY 1;
```

Number_of_week	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
17	663	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91
18	596	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85
19	427	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49
20	358	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0
21	317	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0
22	326	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0
23	328	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0
24	339	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0
25	305	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0
26	288	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0
27	292	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0
28	274	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0
29	270	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0
30	294	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0
31	215	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0
32	267	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0
33	286	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0
34	279	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D. **Weekly Engagement Per Device:**

- Objective: Measure the activeness of users on a weekly basis per device.
- Your Task: Write an SQL query to calculate the weekly engagement per device.

Answer: **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",

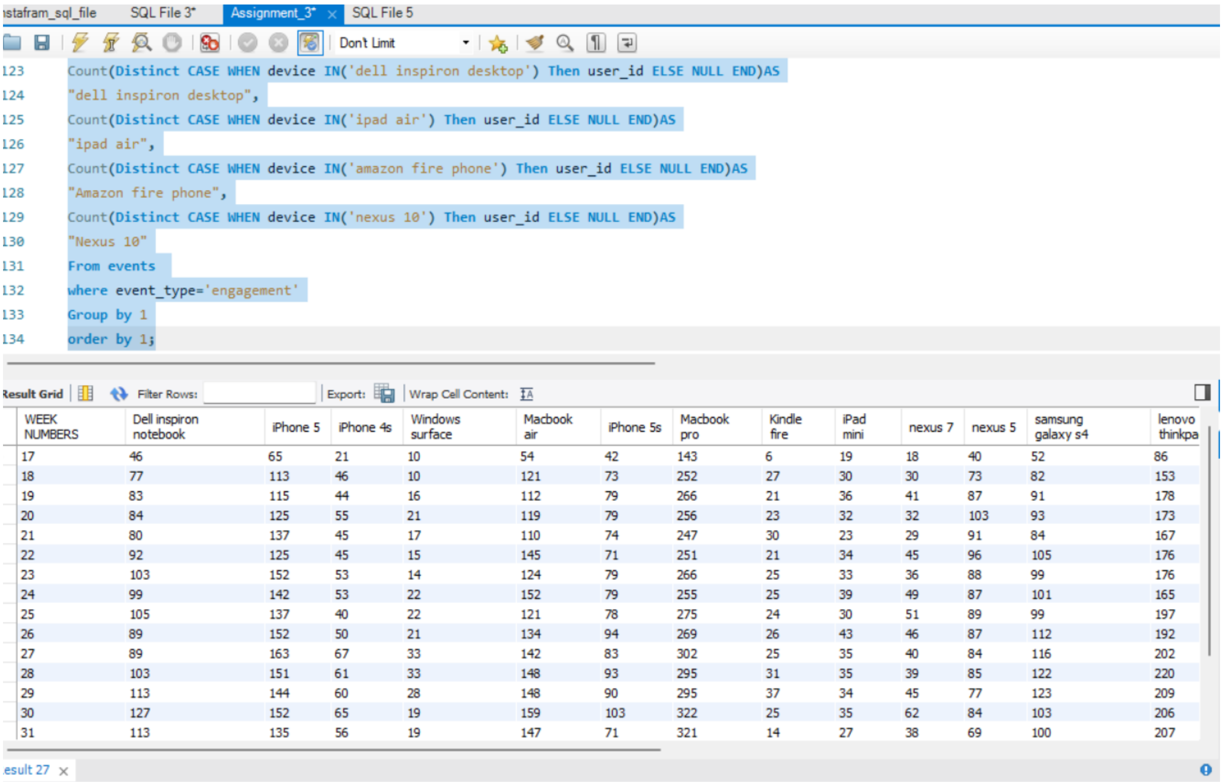
Count(Distinct CASE WHEN device IN('kindle fire') Then user_id ELSE NULL END)AS

"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('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('mac mini') Then user_id ELSE NULL END)AS
"mac mini",
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;
```



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

E. **Email Engagement Analysis:**

- Objective: Analyze how users are engaging with the email service.
- Your Task: Write an SQL query to calculate the email engagement metrics.

Answer: **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 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

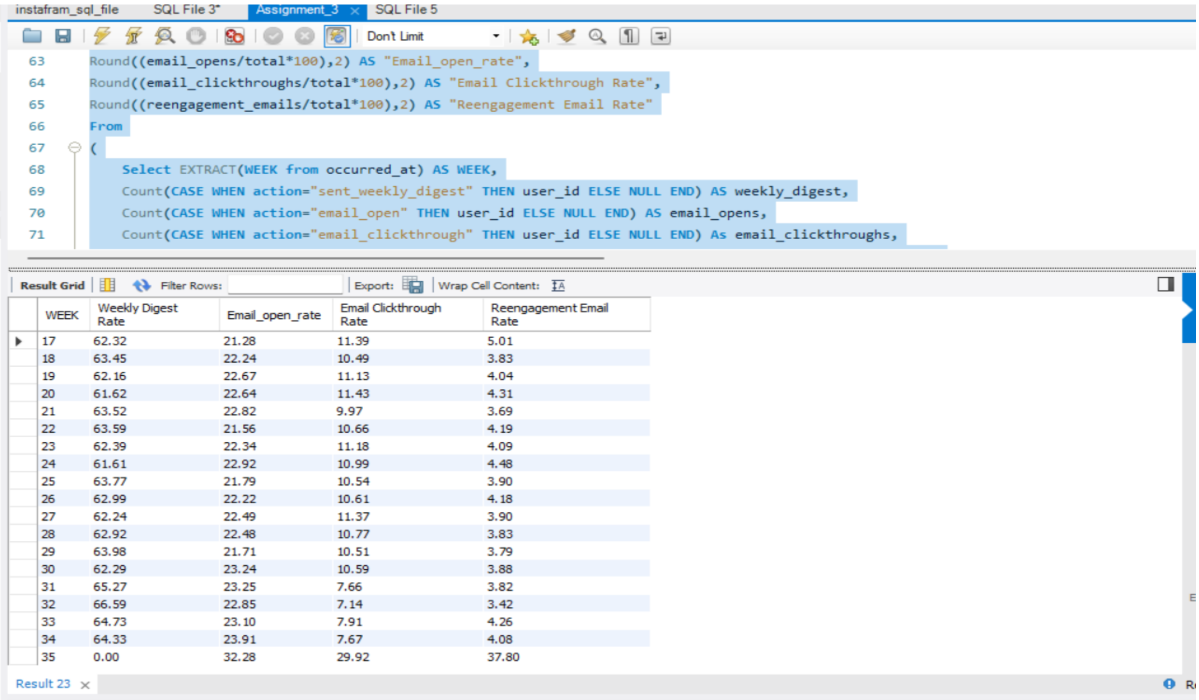
From email_events

Group BY 1

)tabs

Group By week

order By WEEK ;



```
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,
```

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

Results:

The project was the best way to boost up with my SQL knowledge. I have worked on throughout to analyse weekly and monthly data. I came to know about the data cleaning and data formatting in this. I can get the desired output while writing the queries.

It was the most difficult task as it consists to real time data and business insights. It helped me to brush up my knowledge on SQL and as well as, it helped me to face some challenges while working on Advanced SQL.

This Encouraged me to solve more problems until I am confident with it and helped me to create beautiful insights of the data.