OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Name: PARITOSH GANDRE Email id: paritoshkrcg@gmail.com

PROJECT DESCRIPTION: The project is about Operation Analytics and Investigating Metric Spike. The project is about finding the user behaviour, user engagement, and user interaction with the product within a specified date or whole. The way I handled this project is going step by step with each question, I got stuck at few but I kept thinking and tried to solve every question by giving my best. From this project I am going to find out user metrics for the product company to make informed decisions about their user base, like who they want to target more, where they have to work more to retain the users, likewise.

APPROACH: My approach to this project was, firstly I understood the database which was provided in the portal. Then I sought for any similarities between all the tables. Following which I started my work from the very first question for the first table i.e Case Study 1 (Job Data), then following the sequence I tried to answer each question from both Case Studies.

TECH-STACK USED: The software I used to solve these queries is MySQL WORKBENCH 8.0 CE. The main reason behind using this software for me is I prefer software experience over web application experience. Also workbench has all the features like we can import our own data and start querying, create different workspaces and many more, which we do not get in any web browser application.

INSIGHTS: Talking about the insights I gained from this project are, first things first, this was a really tough project for me as I don’t have much experience with SQL, I found it difficult at times to get a desired result for a query, or to even design a query for a problem.

But getting through this project has helped me a lot in many ways, it has uplifted my confidence in SQL. We have to merge tables to get the desired result for a problem. Sometimes even have to make the most complex query to solve a problem which seems intimidating, but that’s how it is done in SQL or in any other programming language.

I got to know how to tackle questions which demand grouping of tables, creating new columns and inferring from those new columns. By making few changes whole result changes, so we have to be careful and clear about our thinking process and queries.

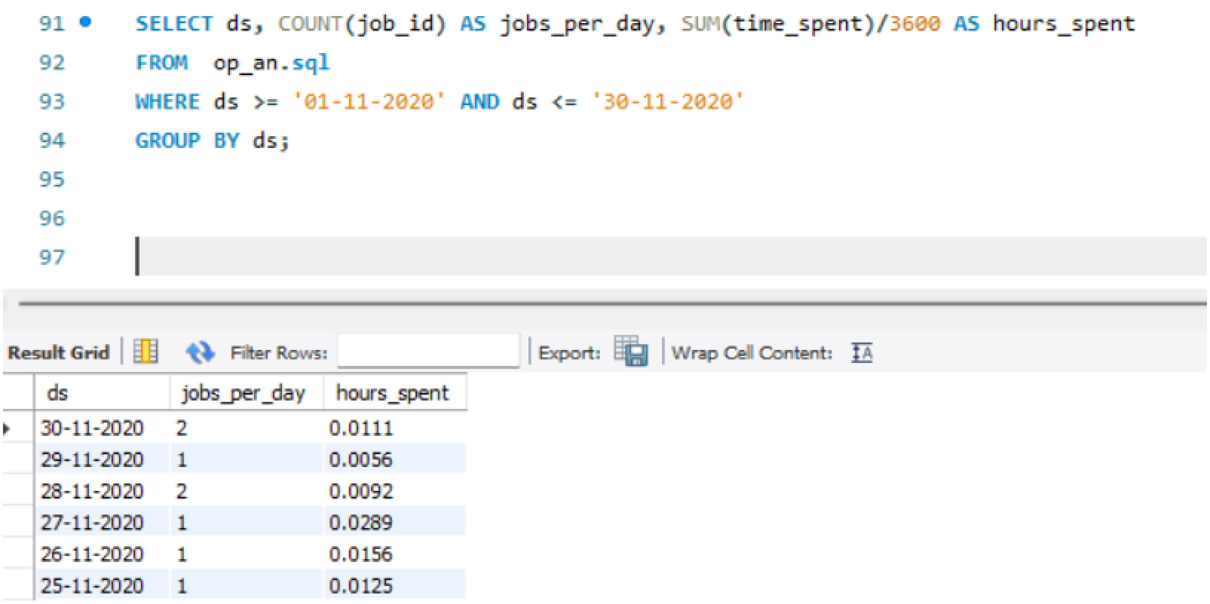
OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Name: PARITOSH GANDRE Email id: paritoshkrcg@gmail.com

RESULTS:

CASE STUDY 1 (JOB DATA):

A. Number of jobs reviewed: We have to calculate the number of jobs reviewed per hour per day in November 2020.
Since the table only has data of November 2020, it is quite easy to find out jobs reviewed per hour per day. Also the time spent column is in seconds we have to calculate it for an hour.



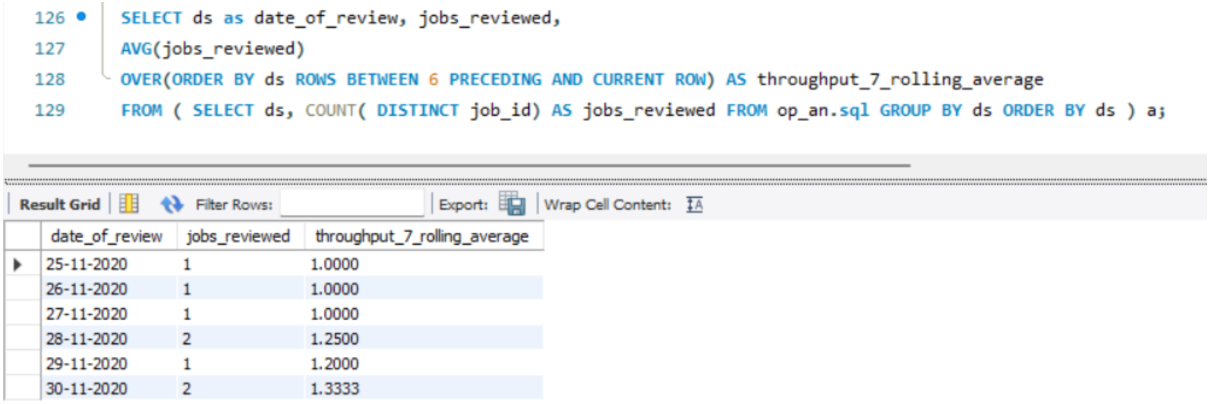
```sql
SELECT ds, COUNT(job_id) AS jobs_per_day, SUM(time_spent)/3600 AS hours_spent
FROM  op_an.sql
WHERE ds >= '01-11-2020' AND ds <= '30-11-2020'
GROUP BY ds;
```

ds	jobs_per_day	hours_spent
30-11-2020	2	0.0111
29-11-2020	1	0.0056
28-11-2020	2	0.0092
27-11-2020	1	0.0289
26-11-2020	1	0.0156
25-11-2020	1	0.0125

As you can see, the jobs reviewed on 30-11-2020 were of 2 ids and total time spent was 40 seconds.

B. Throughput: Here we are supposed to find out events happening per second for 7 days.

   a. What do I prefer between 7-day rolling or daily metric: Well it totally depends on the type of data. If the time is in seconds then I would definitely prefer a daily metric otherwise a 7-day rolling.



```sql
SELECT ds as date_of_review, jobs_reviewed,
AVG(jobs_reviewed)
OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS throughput_7_rolling_average
FROM ( SELECT ds, COUNT( DISTINCT job_id) AS jobs_reviewed FROM op_an.sql GROUP BY ds ORDER BY ds ) a;
```

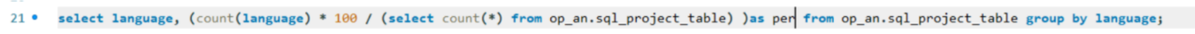
date_of_review	jobs_reviewed	throughput_7_rolling_average
25-11-2020	1	1.0000
26-11-2020	1	1.0000
27-11-2020	1	1.0000
28-11-2020	2	1.2500
29-11-2020	1	1.2000
30-11-2020	2	1.3333

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

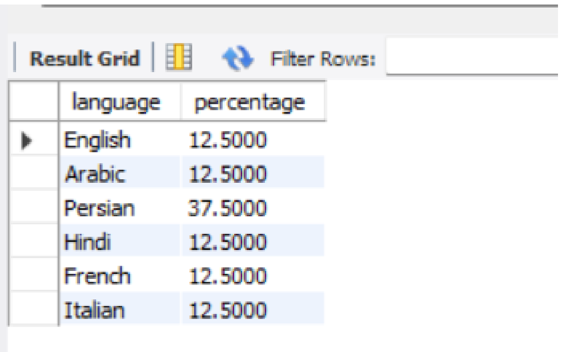
Name: PARITOSH GANDRE Email id: paritoshkrcg@gmail.com

C. Percentage share of each language: In this, we have to find out the percentage shared by each language within the table i.e English, Arabic, Persian, Hindi, French ,and Italian.

   a. To find this out, we have to take count of each language appearing in the table and divide it by row count.



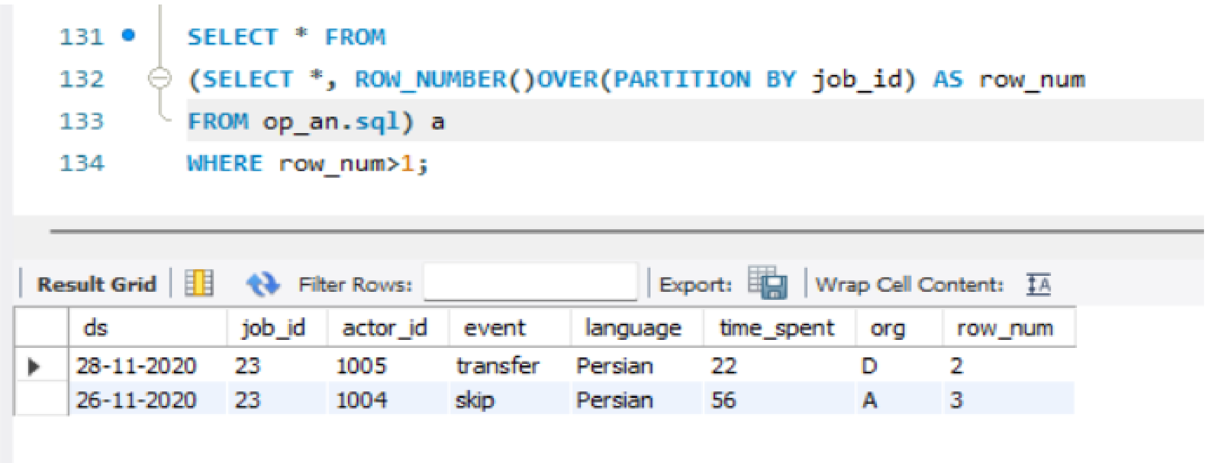
```sql
select language, (count(language) * 100 / (select count(*) from op_an.sql_project_table) )as per from op_an.sql_project_table group by language;
```



language	percentage
English	12.5000
Arabic	12.5000
Persian	37.5000
Hindi	12.5000
French	12.5000
Italian	12.5000

D. Duplicate Rows: Here we have to display the duplicate values or rows occurring in the table.

   a. Since the Job id cannot be duplicated because of the uniqueness of Ids, we will calculate if the table has duplicate Job ids.



```sql
SELECT * FROM
(SELECT *, ROW_NUMBER()OVER(PARTITION BY job_id) AS row_num
FROM op_an.sql) a
WHERE row_num>1;
```

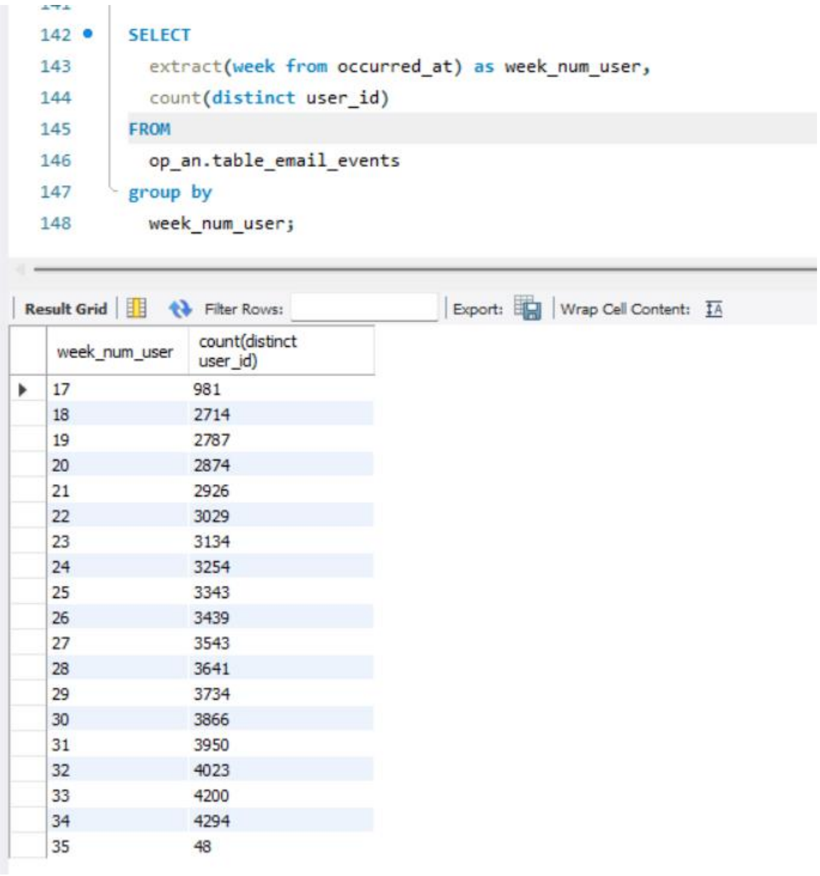
ds	job_id	actor_id	event	language	time_spent	org	row_num
28-11-2020	23	1005	transfer	Persian	22	D	2
26-11-2020	23	1004	skip	Persian	56	A	3

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Name: PARITOSH GANDRE Email id: paritoshkrcg@gmail.com

CASE STUDY 2 (INVESTING METRIC SPIKE) :

A. User Engagement: We have to calculate the weekly user engagement for the given table to measure activeness of users.

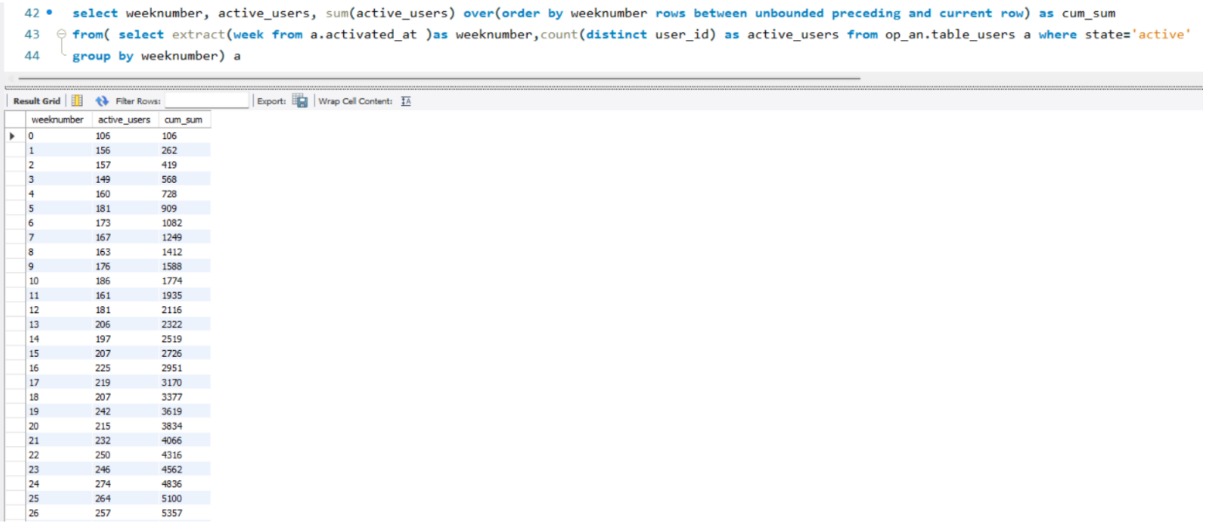


```sql
SELECT
  extract(week from occurred_at) as week_num_user,
  count(distinct user_id)
FROM
  op_an.table_email_events
group by
  week_num_user;
```

week_num_user	count(distinct user_id)
17	981
18	2714
19	2787
20	2874
21	2926
22	3029
23	3134
24	3254
25	3343
26	3439
27	3543
28	3641
29	3734
30	3866
31	3950
32	4023
33	4200
34	4294
35	48

B. User Growth: We have to find out the user growth for the product over time.

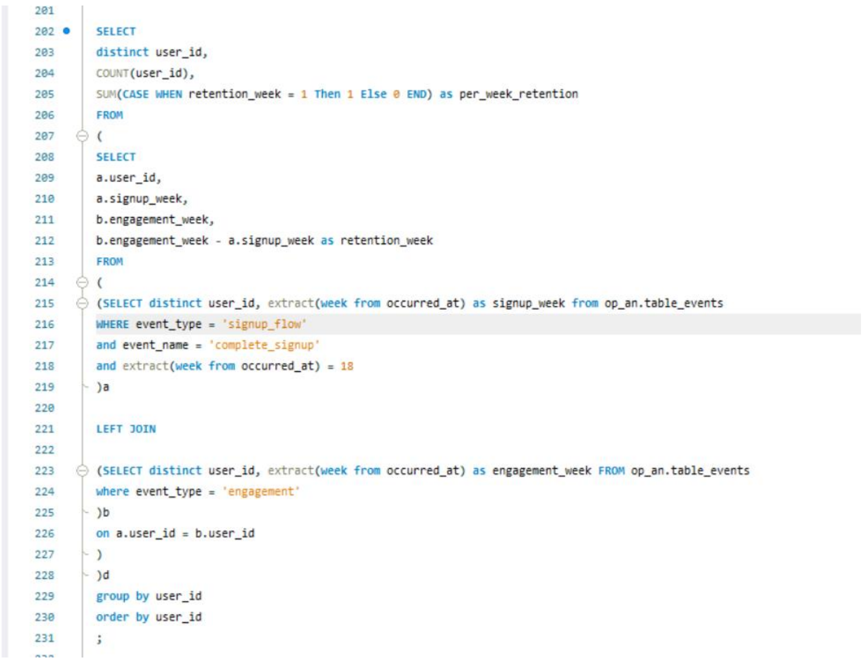
- The total user growth over time comes out to be **9381** users.



```sql
select weeknumber, active_users, sum(active_users) over(order by weeknumber rows between unbounded preceding and current row) as cum_sum
from( select extract(week from a.activated_at )as weeknumber,count(distinct user_id) as active_users from op_an.table_users a where state='active'
group by weeknumber) a
```

weeknumber	active_users	cum_sum
0	106	106
1	156	262
2	157	419
3	149	568
4	160	728
5	181	909
6	173	1082
7	167	1249
8	163	1412
9	176	1588
10	186	1774
11	161	1935
12	181	2116
13	206	2322
14	197	2519
15	207	2726
16	225	2951
17	219	3170
18	207	3377
19	242	3619
20	215	3834
21	232	4066
22	250	4316
23	246	4562
24	274	4836
25	264	5100
26	257	5357

C. Weekly Retention: We have to calculate the weekly retention of users-sign up cohort.

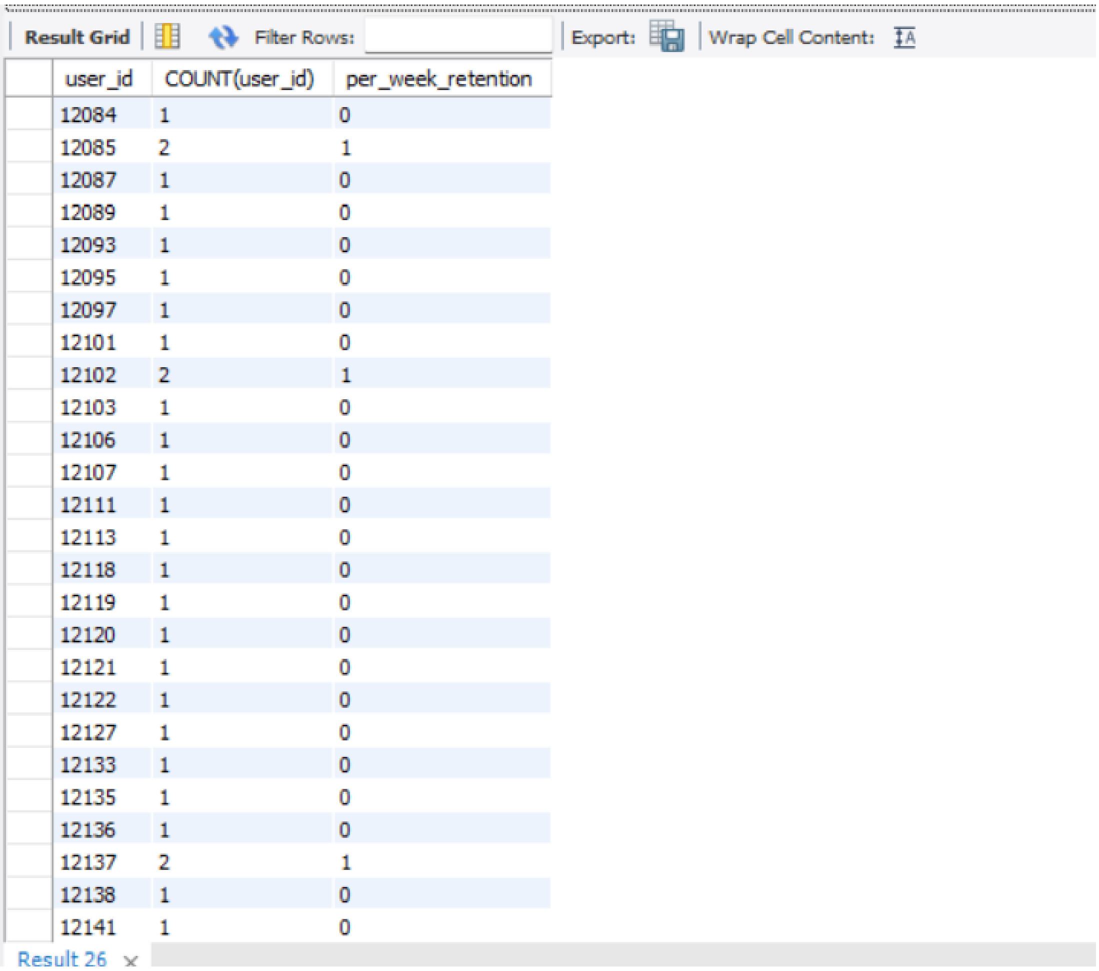


```

SELECT
distinct user_id,
COUNT(user_id),
SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as per_week_retention
FROM
(
SELECT
a.user_id,
a.signup_week,
b.engagement_week,
b.engagement_week - a.signup_week as retention_week
FROM
(
(SELECT distinct user_id, extract(week from occurred_at) as signup_week from op_an.table_events
WHERE event_type = 'signup_flow'
and event_name = 'complete_signup'
and extract(week from occurred_at) = 18
)a

LEFT JOIN

(SELECT distinct user_id, extract(week from occurred_at) as engagement_week FROM op_an.table_events
where event_type = 'engagement'
)b
on a.user_id = b.user_id
)
)d
group by user_id
order by user_id
;
```

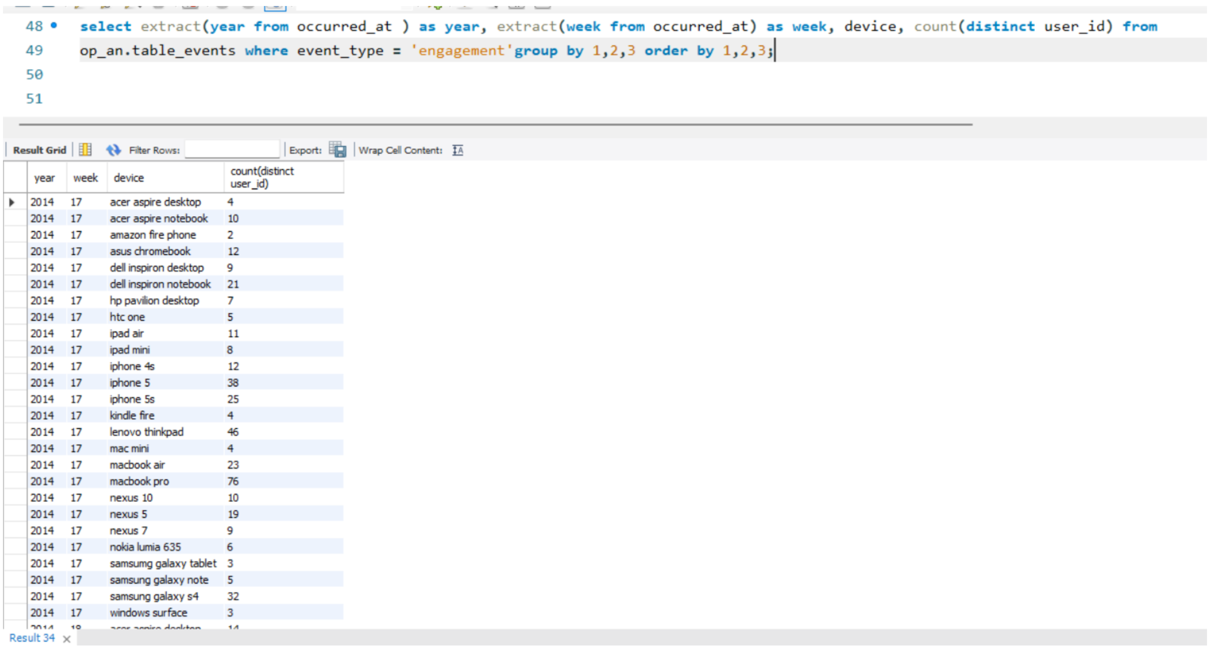


user_id	COUNT(user_id)	per_week_retention
12084	1	0
12085	2	1
12087	1	0
12089	1	0
12093	1	0
12095	1	0
12097	1	0
12101	1	0
12102	2	1
12103	1	0
12106	1	0
12107	1	0
12111	1	0
12113	1	0
12118	1	0
12119	1	0
12120	1	0
12121	1	0
12122	1	0
12127	1	0
12133	1	0
12135	1	0
12136	1	0
12137	2	1
12138	1	0
12141	1	0

OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Name: PARITOSH GANDRE Email id: paritoshkrcg@gmail.com

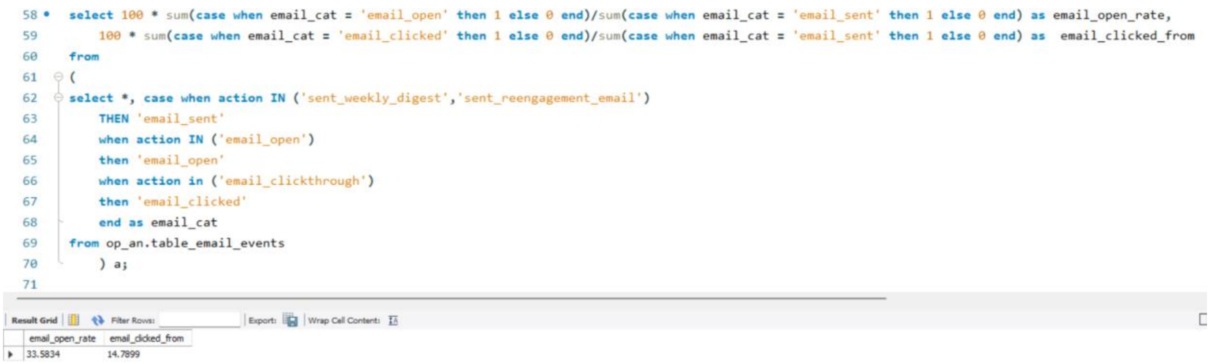
D. Weekly Engagement: Here we have to calculate weekly engagement per device by users.



```
select extract(year from occurred_at ) as year, extract(week from occurred_at) as week, device, count(distinct user_id) from
op_an.table_events where event_type = 'engagement'group by 1,2,3 order by 1,2,3;
```

year	week	device	count(distinct user_id)
2014	17	acer aspire desktop	4
2014	17	acer aspire notebook	10
2014	17	amazon fire phone	2
2014	17	asus chromebook	12
2014	17	dell inspiron desktop	9
2014	17	dell inspiron notebook	21
2014	17	hp pavilion desktop	7
2014	17	htc one	5
2014	17	ipad air	11
2014	17	ipad mini	8
2014	17	iphone 4s	12
2014	17	iphone 5	38
2014	17	iphone 5s	25
2014	17	kindle fire	4
2014	17	lenovo thinkpad	46
2014	17	mac mini	4
2014	17	macbook air	23
2014	17	macbook pro	76
2014	17	nexus 10	10
2014	17	nexus 5	19
2014	17	nexus 7	9
2014	17	nokia lumia 635	6
2014	17	samsumg galaxy tablet	3
2014	17	samsung galaxy note	5
2014	17	samsung galaxy s4	32
2014	17	windows surface	3
2014	18	acer aspire desktop	14

E. Email Engagement: We have to calculate the email engagement metrics.



```
select 100 * sum(case when email_cat = 'email_open' then 1 else 0 end)/sum(case when email_cat = 'email_sent' then 1 else 0 end) as email_open_rate,
    100 * sum(case when email_cat = 'email_clicked' then 1 else 0 end)/sum(case when email_cat = 'email_sent' then 1 else 0 end) as  email_clicked_from
from
(
select *, case when action IN ('sent_weekly_digest','sent_reengagement_email')
    THEN 'email_sent'
    when action IN ('email_open')
    then 'email_open'
    when action in ('email_clickthrough')
    then 'email_clicked'
    end as email_cat
from op_an.table_email_events
    ) a;
```

email_open_rate	email_clicked_from
33.5834	14.7899