Operation Analytics and Investigating Metric Spike

BY

Rahul Patil

◆ Project Description:

- 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.

- 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.

- In this project we are going to analyze company operation and user engagement with product and services, growth of product and increase or decrease in user engagement which will help to find the area which required improvement which further help to generate more revenue.

Approach:

- To successfully carry out this project we are going to use **SIX STEP** of Data Analysis Process i.e (Ask, Prepare, Process, Analyze, Share, Act)
- Ask step include asking right set of question which justify goal and give motivation to carry out analysis

- We have following set of question (reasons) to justify goal of this project.
  - Calculate the number of jobs reviewed per hour per day for November 2020?
  - Calculate 7-day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?
  - Calculate the percentage share of each language in the last 30 days?
  - How will you display duplicates from the table?
  - Calculate the weekly user engagement?
  - Calculate the user growth for product?
  - Calculate the weekly retention of users-sign up cohort?
  - Calculate the weekly engagement per device?
  - Calculate the email engagement metrics?

- Prepare: We have users and event data store in RDBMS in different table like users, events, email_events, job_data.
- This step includes selecting right data, tools, data source to make project successful
- Process: Data we have is already in process format i.e data store in RDBMS is process, clean and useful data.
- Analyze: We are using SQL to analyze data to draw insight.
- Share: we are showing data obtain from analysis in the form of row and column as well as chart wherever required for better and easy understanding.

- Act: Step include taking decision based on insight opt from this project.

Tech-Stack Used:

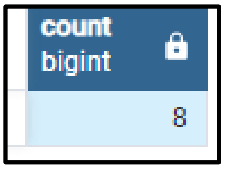
- Data is stored in RDBMS (Relational Database Management System) in different table
- We could use jupyter notebook and programming language to carry out this project. Which requires programming skills as well further extra step such as connecting to database, loading data from DBMS to notebook.
- While RDBMS come with inbuilt data query language called SQL (Structure Query Language) which can successfully carry out entire analysis for this project.
- Tech we are going to use for this project is PostgreSQL which is RDBMS and excel to visualize output using charts and graph.

Insights:

Case Study 1 (Job Data):

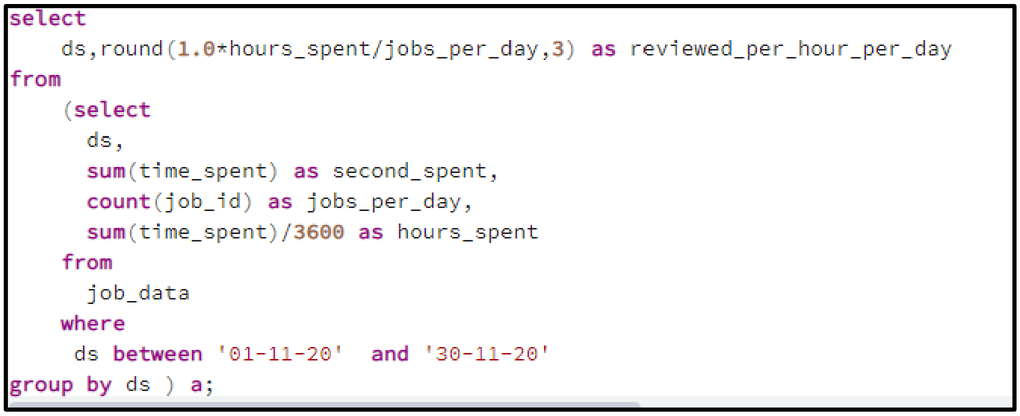
1.Calculate the number of jobs reviewed per hour per day for November 2020.

```
select count(*) from job_data
```

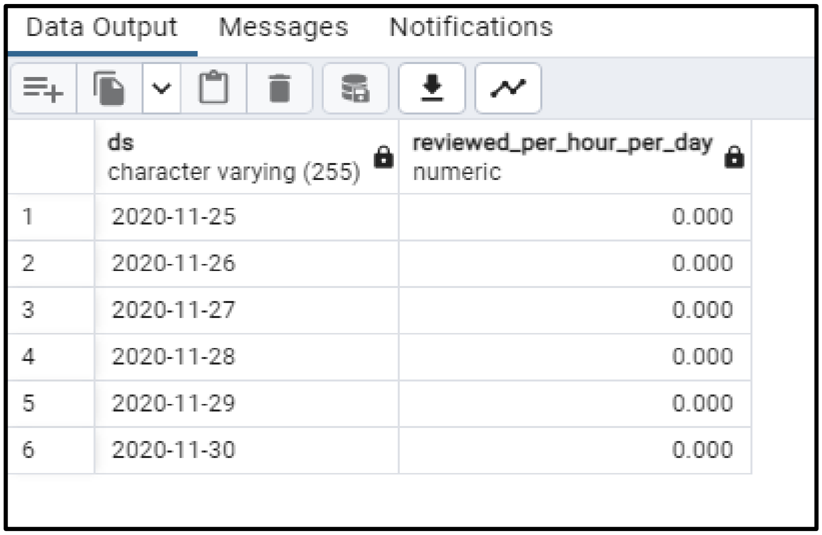


```
count
bigint
      8
```

Total number of record are very few it will difficult to do analysis and take decision when data is limited.



```
select
    ds,round(1.0*hours_spent/jobs_per_day,3) as reviewed_per_hour_per_day
from
    (select
      ds,
      sum(time_spent) as second_spent,
      count(job_id) as jobs_per_day,
      sum(time_spent)/3600 as hours_spent
    from
      job_data
    where
     ds between '01-11-20'  and '30-11-20'
group by ds ) a;
```

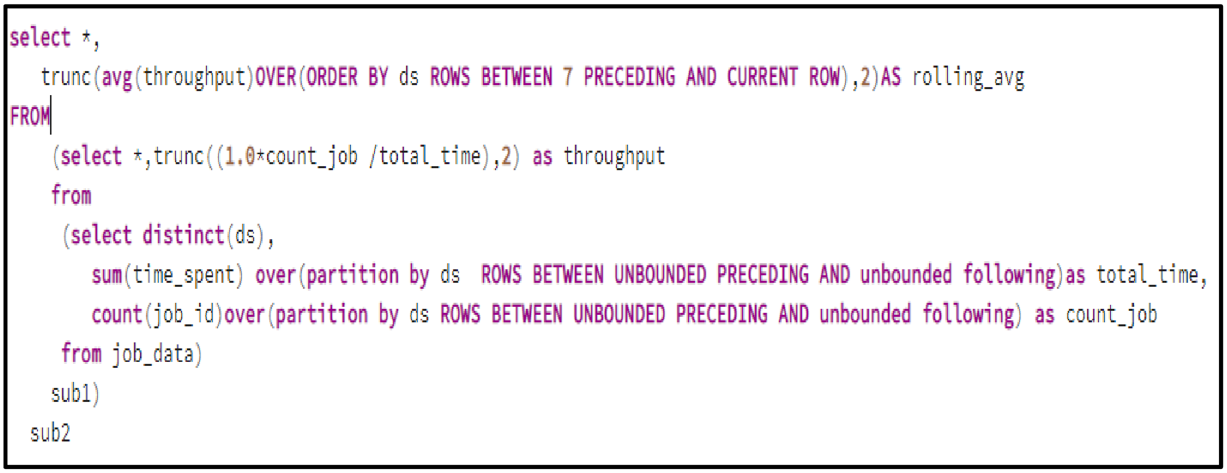


	ds character varying (255)	reviewed_per_hour_per_day numeric
1	2020-11-25	0.000
2	2020-11-26	0.000
3	2020-11-27	0.000
4	2020-11-28	0.000
5	2020-11-29	0.000
6	2020-11-30	0.000

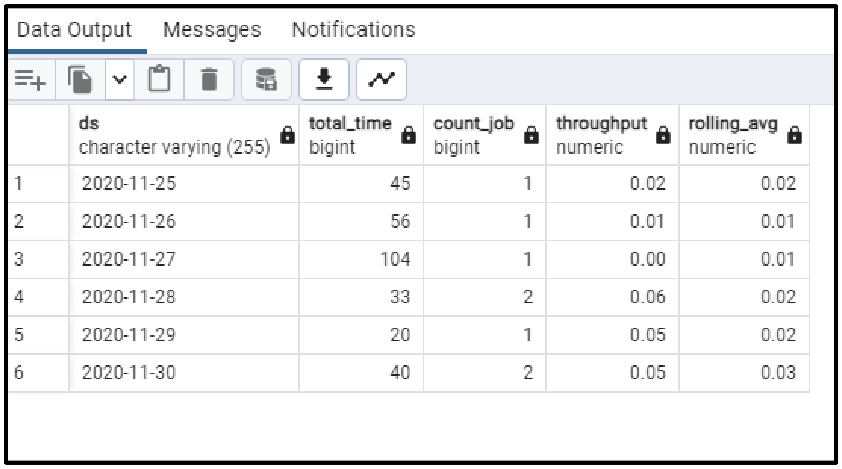
most job review in seconds, hardly 1 minute spend that's why jobs reviewed per hour per day is zero

2.Calculate 7-day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

Throughput: It is the no. of events happening per second



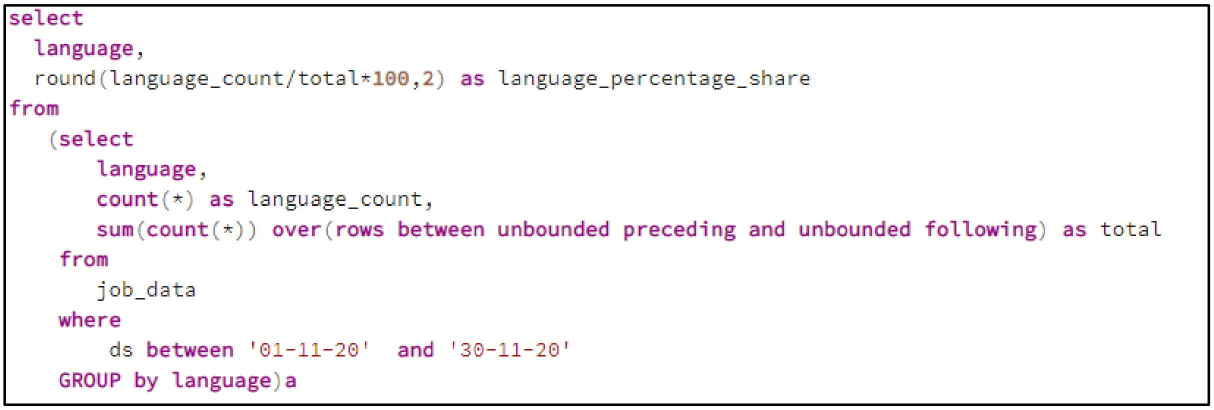
```
select *,
   trunc(avg(throughput)OVER(ORDER BY ds ROWS BETWEEN 7 PRECEDING AND CURRENT ROW),2)AS rolling_avg
FROM
    (select *,trunc((1.0*count_job /total_time),2) as throughput
    from
     (select distinct(ds),
        sum(time_spent) over(partition by ds  ROWS BETWEEN UNBOUNDED PRECEDING AND unbounded following)as total_time,
        count(job_id)over(partition by ds ROWS BETWEEN UNBOUNDED PRECEDING AND unbounded following) as count_job
     from job_data)
    sub1)
  sub2
```



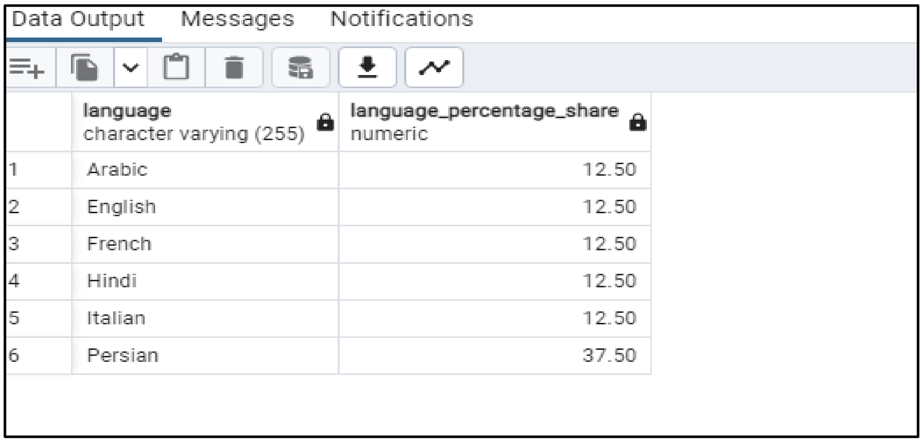
	ds character varying (255)	total_time bigint	count_job bigint	throughput numeric	rolling_avg numeric
1	2020-11-25	45	1	0.02	0.02
2	2020-11-26	56	1	0.01	0.01
3	2020-11-27	104	1	0.00	0.01
4	2020-11-28	33	2	0.06	0.02
5	2020-11-29	20	1	0.05	0.02
6	2020-11-30	40	2	0.05	0.03

Data is limited so we can’t conclude whether daily metric or 7-day rolling average should we prefer, we required more data to make conclusion.

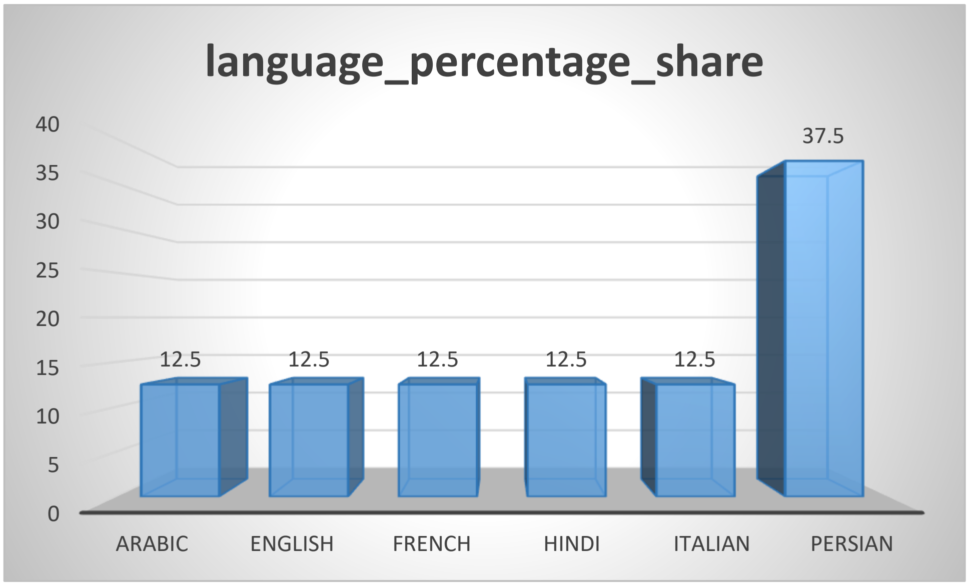
3.Calculate the percentage share of each language in the last 30 days?



```
select
  language,
  round(language_count/total*100,2) as language_percentage_share
from
   (select
       language,
       count(*) as language_count,
       sum(count(*)) over(rows between unbounded preceding and unbounded following) as total
    from
       job_data
    where
        ds between '01-11-20'  and '30-11-20'
    GROUP by language)a
```

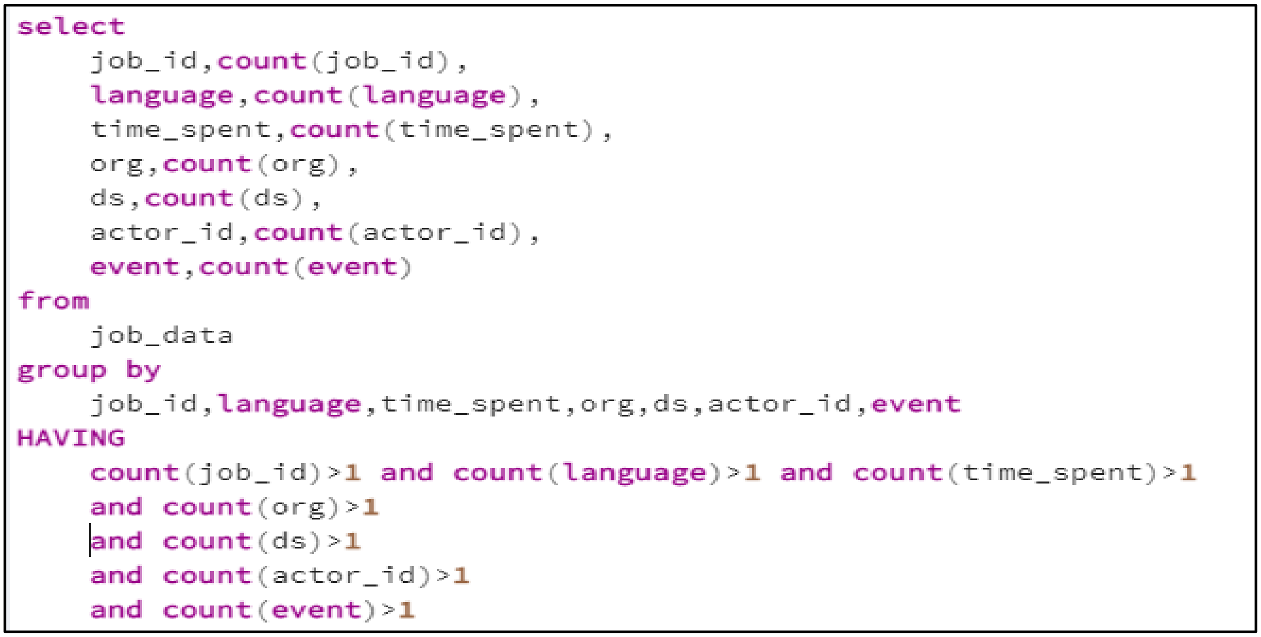


	language character varying (255)	language_percentage_share numeric
1	Arabic	12.50
2	English	12.50
3	French	12.50
4	Hindi	12.50
5	Italian	12.50
6	Persian	37.50

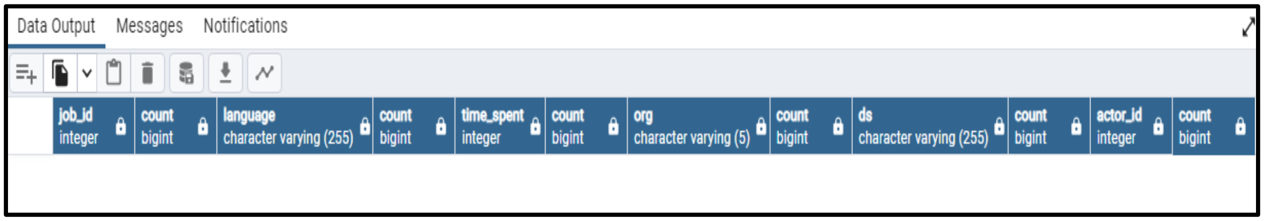


Each language has equal weightage of 12.50% except Persian language having higher weightage of 37.50%

4.How will you display duplicates from the table?



```
select
    job_id,count(job_id),
    language,count(language),
    time_spent,count(time_spent),
    org,count(org),
    ds,count(ds),
    actor_id,count(actor_id),
    event,count(event)
from
    job_data
group by
    job_id,language,time_spent,org,ds,actor_id,event
HAVING
    count(job_id)>1 and count(language)>1 and count(time_spent)>1
    and count(org)>1
    and count(ds)>1
    and count(actor_id)>1
    and count(event)>1
```



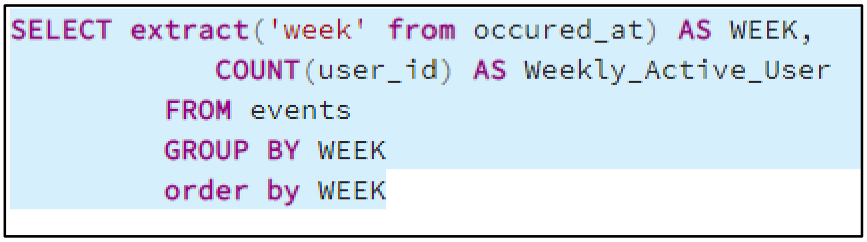
job_id	count	language	count	time_spent	count	org	count	ds	count	actor_id	count
integer	bigint	character varying (255)	bigint	integer	bigint	character varying (5)	bigint	character varying (255)	bigint	integer	bigint

No duplicate records found for given data

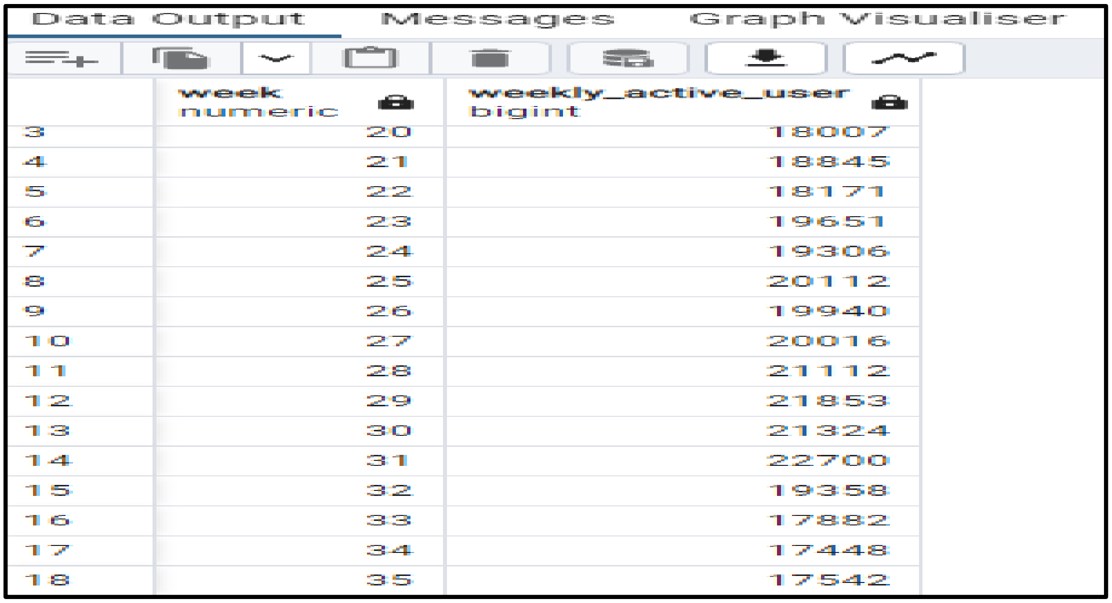
Insights:

Case Study 2 (Investigating metric spike)

1.Calculate the weekly user engagement?



```
SELECT extract('week' from occured_at) AS WEEK,
            COUNT(user_id) AS Weekly_Active_User
         FROM events
         GROUP BY WEEK
         order by WEEK
```

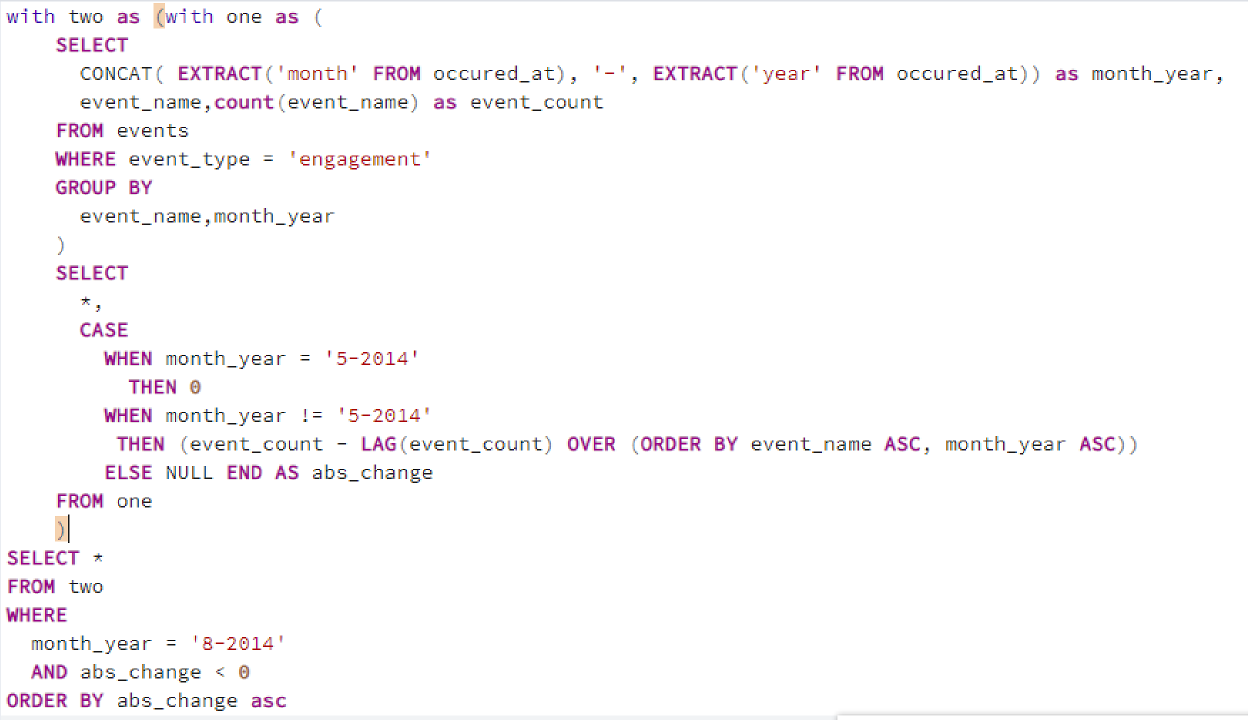


	week numeric	weekly_active_user bigint
3	20	18007
4	21	18845
5	22	18171
6	23	19651
7	24	19306
8	25	20112
9	26	19940
10	27	20016
11	28	21112
12	29	21853
13	30	21324
14	31	22700
15	32	19358
16	33	17882
17	34	17448
18	35	17542

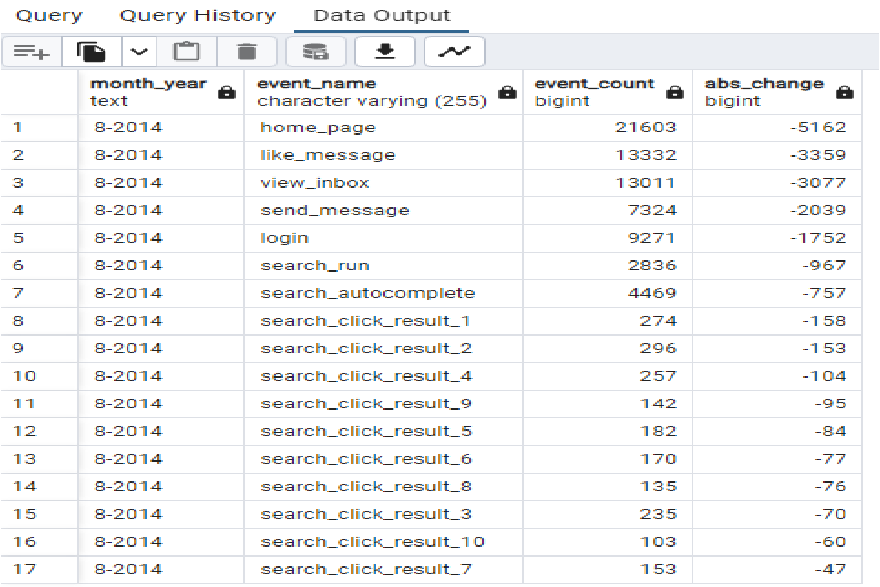
From above query its clearly visible that there is decline in user engagement from week 32.

The next thing I wanted to do was count the number of occurrences of each ‘engagement’ event month over month as decline in user in between July and August

to see if there’s any significant event(s) that caused the dip after 32 week



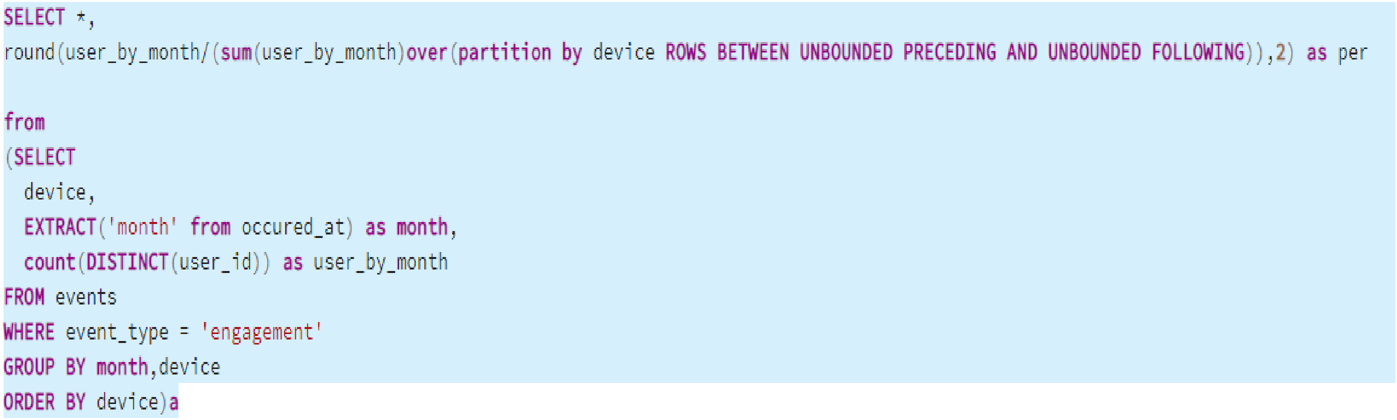
```
with two as (with one as (
    SELECT
      CONCAT( EXTRACT('month' FROM occured_at), '-', EXTRACT('year' FROM occured_at)) as month_year,
      event_name,count(event_name) as event_count
    FROM events
    WHERE event_type = 'engagement'
    GROUP BY
      event_name,month_year
    )
    SELECT
      *,
      CASE
        WHEN month_year = '5-2014'
          THEN 0
        WHEN month_year != '5-2014'
         THEN (event_count - LAG(event_count) OVER (ORDER BY event_name ASC, month_year ASC))
        ELSE NULL END AS abs_change
    FROM one
    )
SELECT *
FROM two
WHERE
  month_year = '8-2014'
  AND abs_change < 0
ORDER BY abs_change asc
```



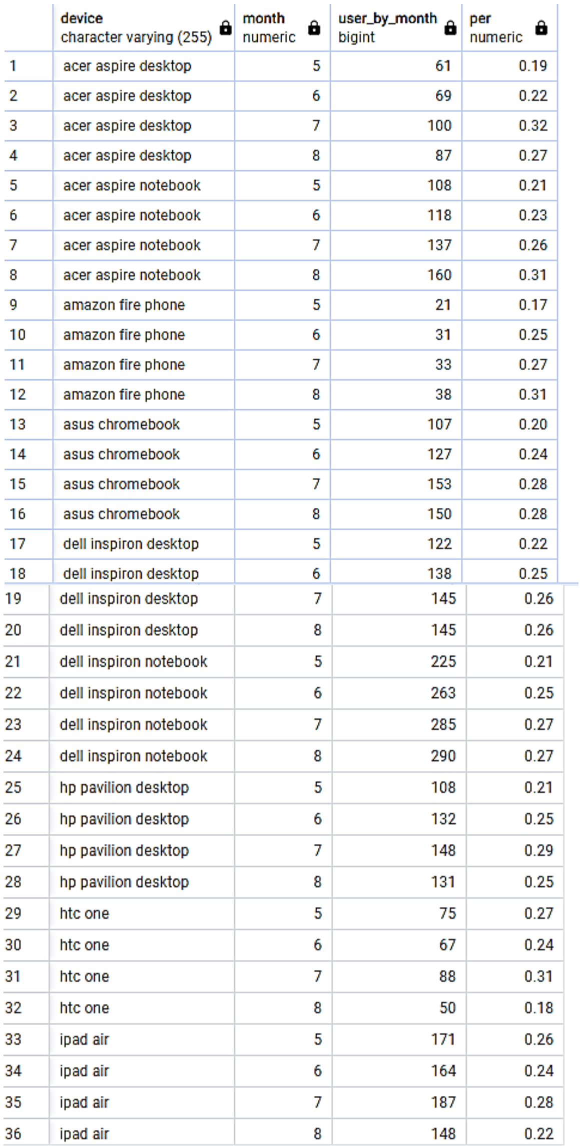
	month_year text	event_name character varying (255)	event_count bigint	abs_change bigint
1	8-2014	home_page	21603	-5162
2	8-2014	like_message	13332	-3359
3	8-2014	view_inbox	13011	-3077
4	8-2014	send_message	7324	-2039
5	8-2014	login	9271	-1752
6	8-2014	search_run	2836	-967
7	8-2014	search_autocomplete	4469	-757
8	8-2014	search_click_result_1	274	-158
9	8-2014	search_click_result_2	296	-153
10	8-2014	search_click_result_4	257	-104
11	8-2014	search_click_result_9	142	-95
12	8-2014	search_click_result_5	182	-84
13	8-2014	search_click_result_6	170	-77
14	8-2014	search_click_result_8	135	-76
15	8-2014	search_click_result_3	235	-70
16	8-2014	search_click_result_10	103	-60
17	8-2014	search_click_result_7	153	-47

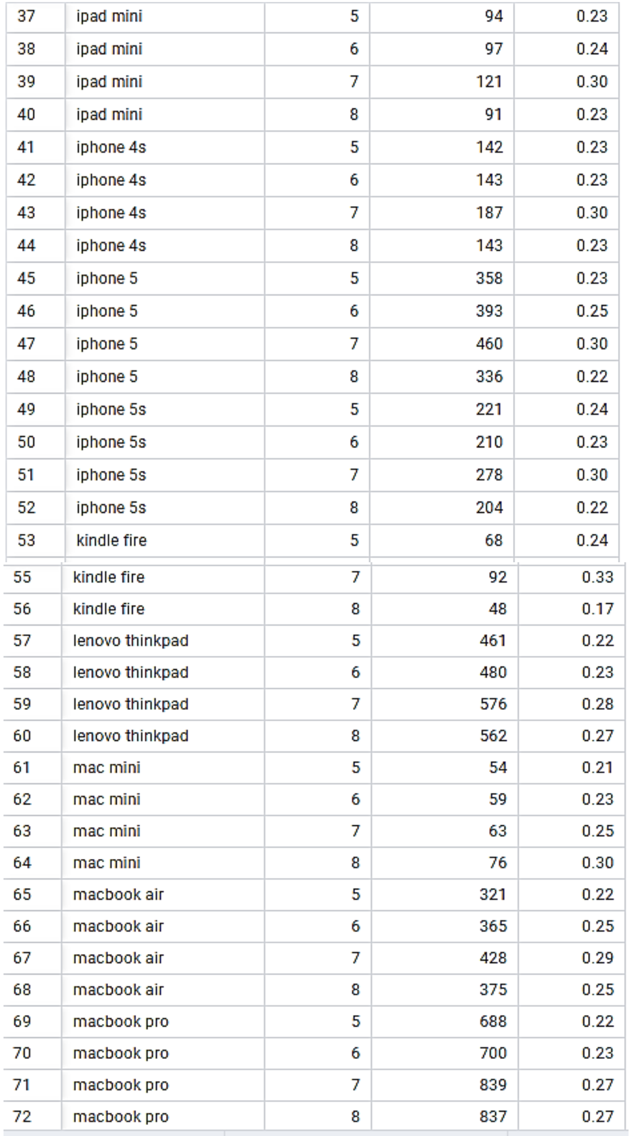
The dip in engagement was largely attributed to home_page, like_message_ view_inbox, send_message, and login. It seems like the drop in all of these events are simply related to the fact that users are loging in less.

2.Calculate the user growth for product?



```
SELECT *,
round(user_by_month/(sum(user_by_month)over(partition by device ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)),2) as per

from
(SELECT
  device,
  EXTRACT('month' from occured_at) as month,
  count(DISTINCT(user_id)) as user_by_month
FROM events
WHERE event_type = 'engagement'
GROUP BY month,device
ORDER BY device)a
```



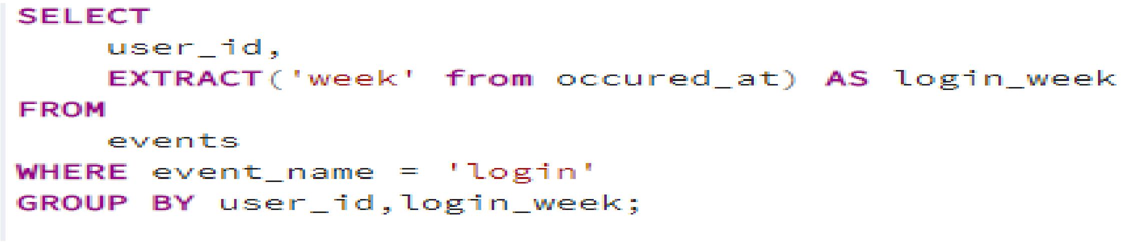


	device character varying (255)	month numeric	user_by_month bigint	per numeric
1	acer aspire desktop	5	61	0.19
2	acer aspire desktop	6	69	0.22
3	acer aspire desktop	7	100	0.32
4	acer aspire desktop	8	87	0.27
5	acer aspire notebook	5	108	0.21
6	acer aspire notebook	6	118	0.23
7	acer aspire notebook	7	137	0.26
8	acer aspire notebook	8	160	0.31
9	amazon fire phone	5	21	0.17
10	amazon fire phone	6	31	0.25
11	amazon fire phone	7	33	0.27
12	amazon fire phone	8	38	0.31
13	asus chromebook	5	107	0.20
14	asus chromebook	6	127	0.24
15	asus chromebook	7	153	0.28
16	asus chromebook	8	150	0.28
17	dell inspiron desktop	5	122	0.22
18	dell inspiron desktop	6	138	0.25
19	dell inspiron desktop	7	145	0.26
20	dell inspiron desktop	8	145	0.26
21	dell inspiron notebook	5	225	0.21
22	dell inspiron notebook	6	263	0.25
23	dell inspiron notebook	7	285	0.27
24	dell inspiron notebook	8	290	0.27
25	hp pavilion desktop	5	108	0.21
26	hp pavilion desktop	6	132	0.25
27	hp pavilion desktop	7	148	0.29
28	hp pavilion desktop	8	131	0.25
29	htc one	5	75	0.27
30	htc one	6	67	0.24
31	htc one	7	88	0.31
32	htc one	8	50	0.18
33	ipad air	5	171	0.26
34	ipad air	6	164	0.24
35	ipad air	7	187	0.28
36	ipad air	8	148	0.22
37	ipad mini	5	94	0.23
38	ipad mini	6	97	0.24
39	ipad mini	7	121	0.30
40	ipad mini	8	91	0.23
41	iphone 4s	5	142	0.23
42	iphone 4s	6	143	0.23
43	iphone 4s	7	187	0.30
44	iphone 4s	8	143	0.23
45	iphone 5	5	358	0.23
46	iphone 5	6	393	0.25
47	iphone 5	7	460	0.30
48	iphone 5	8	336	0.22
49	iphone 5s	5	221	0.24
50	iphone 5s	6	210	0.23
51	iphone 5s	7	278	0.30
52	iphone 5s	8	204	0.22
53	kindle fire	5	68	0.24
55	kindle fire	7	92	0.33
56	kindle fire	8	48	0.17
57	lenovo thinkpad	5	461	0.22
58	lenovo thinkpad	6	480	0.23
59	lenovo thinkpad	7	576	0.28
60	lenovo thinkpad	8	562	0.27
61	mac mini	5	54	0.21
62	mac mini	6	59	0.23
63	mac mini	7	63	0.25
64	mac mini	8	76	0.30
65	macbook air	5	321	0.22
66	macbook air	6	365	0.25
67	macbook air	7	428	0.29
68	macbook air	8	375	0.25
69	macbook pro	5	688	0.22
70	macbook pro	6	700	0.23
71	macbook pro	7	839	0.27
72	macbook pro	8	837	0.27

Number of user per device is decline in month of August mostly for tablet and mobiles.

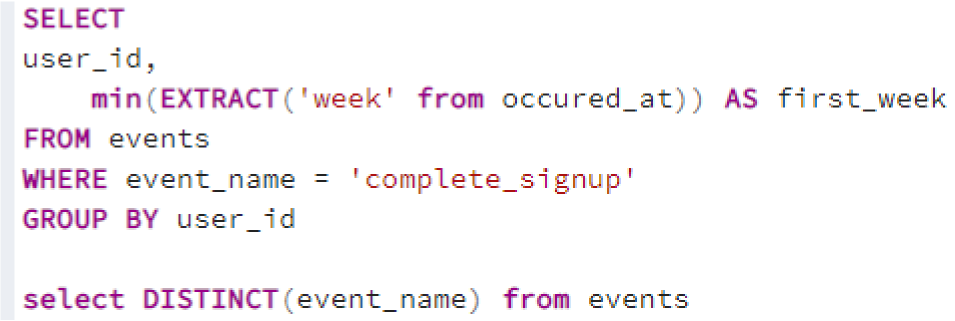
3.Calculate the weekly retention of users-sign up cohort?

first calculate user login by week



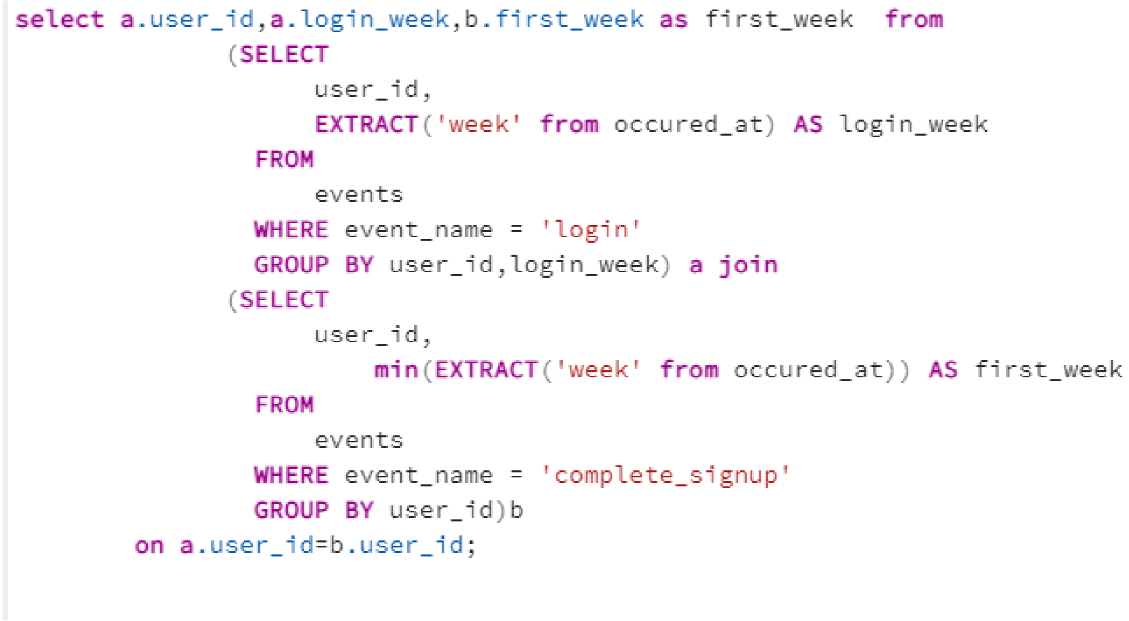
```
SELECT
    user_id,
    EXTRACT('week' from occured_at) AS login_week
FROM
    events
WHERE event_name = 'login'
GROUP BY user_id,login_week;
```

calculate user who completed sign up we use this user whether this user login on weekly basis or not



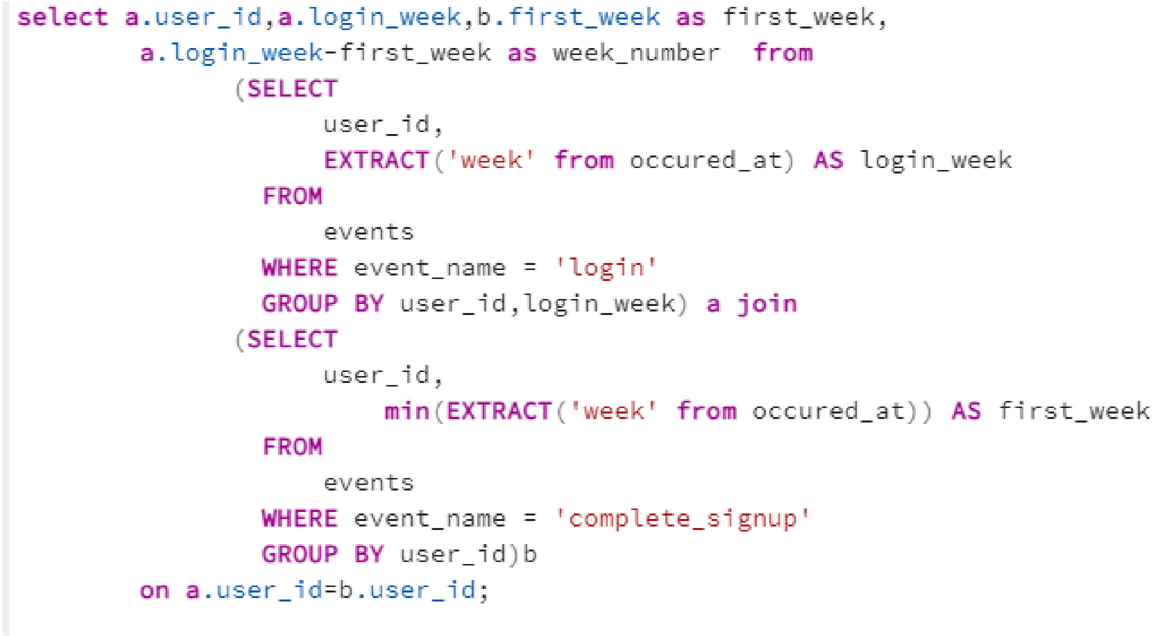
```
SELECT
user_id,
    min(EXTRACT('week' from occured_at)) AS first_week
FROM events
WHERE event_name = 'complete_signup'
GROUP BY user_id

select DISTINCT(event_name) from events
```

we get login_week and first_week side by side for each user using the query below, with an INNER JOIN



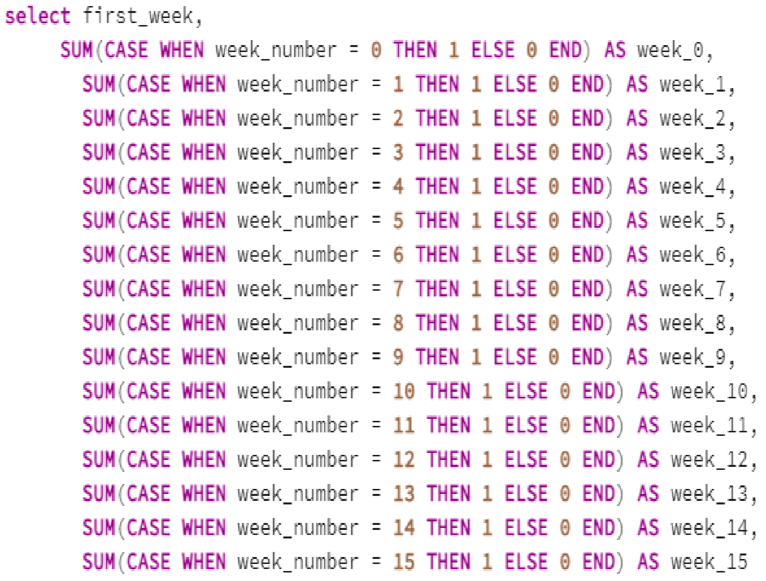
```
select a.user_id,a.login_week,b.first_week as first_week  from
            (SELECT
                  user_id,
                  EXTRACT('week' from occured_at) AS login_week
             FROM
                  events
             WHERE event_name = 'login'
             GROUP BY user_id,login_week) a join
            (SELECT
                  user_id,
                      min(EXTRACT('week' from occured_at)) AS first_week
             FROM
                  events
             WHERE event_name = 'complete_signup'
             GROUP BY user_id)b
      on a.user_id=b.user_id;
```

calculate the difference between login_week and first_week to calculate week_number (number of week)

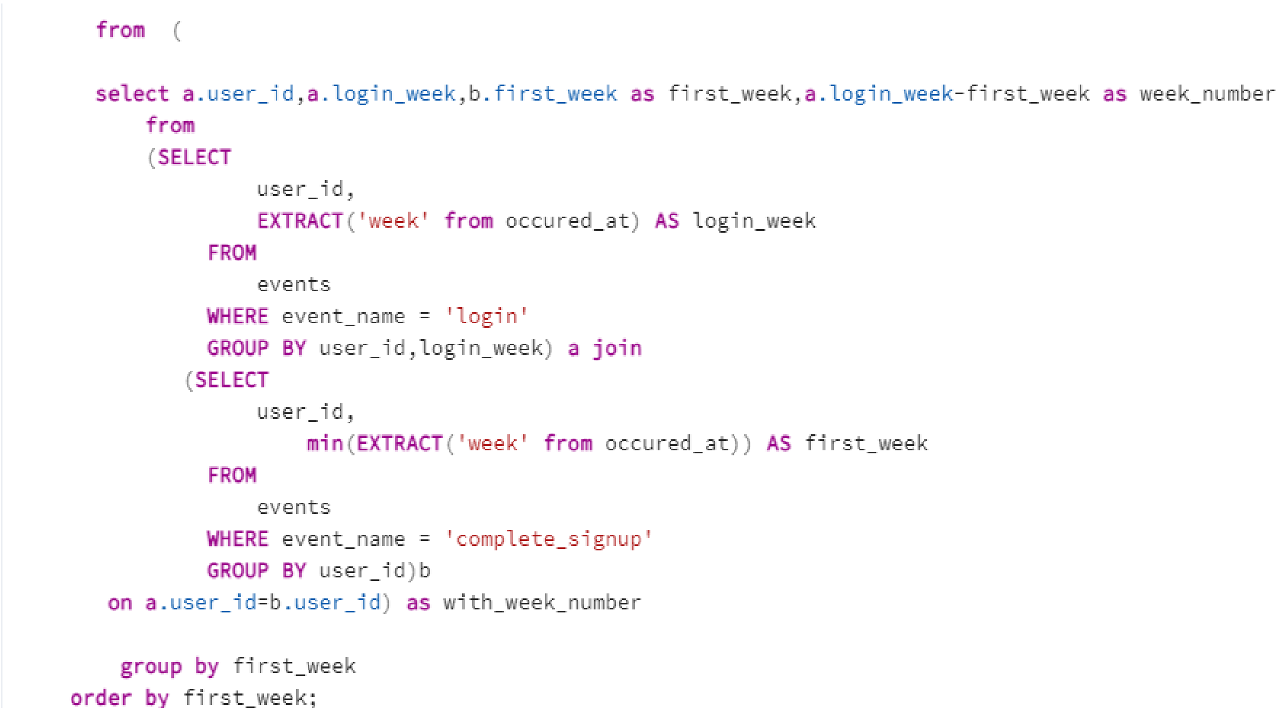


```
select a.user_id,a.login_week,b.first_week as first_week,
        a.login_week-first_week as week_number  from
              (SELECT
                    user_id,
                    EXTRACT('week' from occured_at) AS login_week
                FROM
                    events
                WHERE event_name = 'login'
                GROUP BY user_id,login_week) a join
              (SELECT
                    user_id,
                        min(EXTRACT('week' from occured_at)) AS first_week
                FROM
                    events
                WHERE event_name = 'complete_signup'
                GROUP BY user_id)b
        on a.user_id=b.user_id;
```

Combining all above query



```
select first_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
```

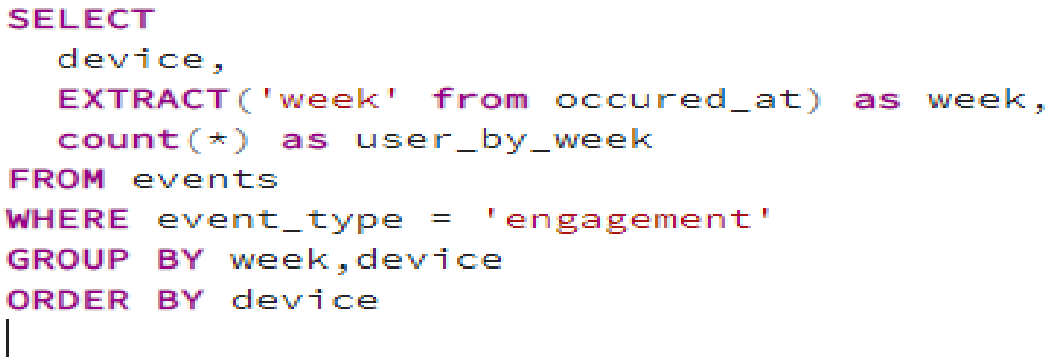


```
        from  (

        select a.user_id,a.login_week,b.first_week as first_week,a.login_week-first_week as week_number
            from
            (SELECT
                     user_id,
                     EXTRACT('week' from occured_at) AS login_week
                 FROM
                     events
                 WHERE event_name = 'login'
                 GROUP BY user_id,login_week) a join
               (SELECT
                     user_id,
                         min(EXTRACT('week' from occured_at)) AS first_week
                 FROM
                     events
                 WHERE event_name = 'complete_signup'
                 GROUP BY user_id)b
         on a.user_id=b.user_id) as with_week_number

          group by first_week
      order by first_week;
```

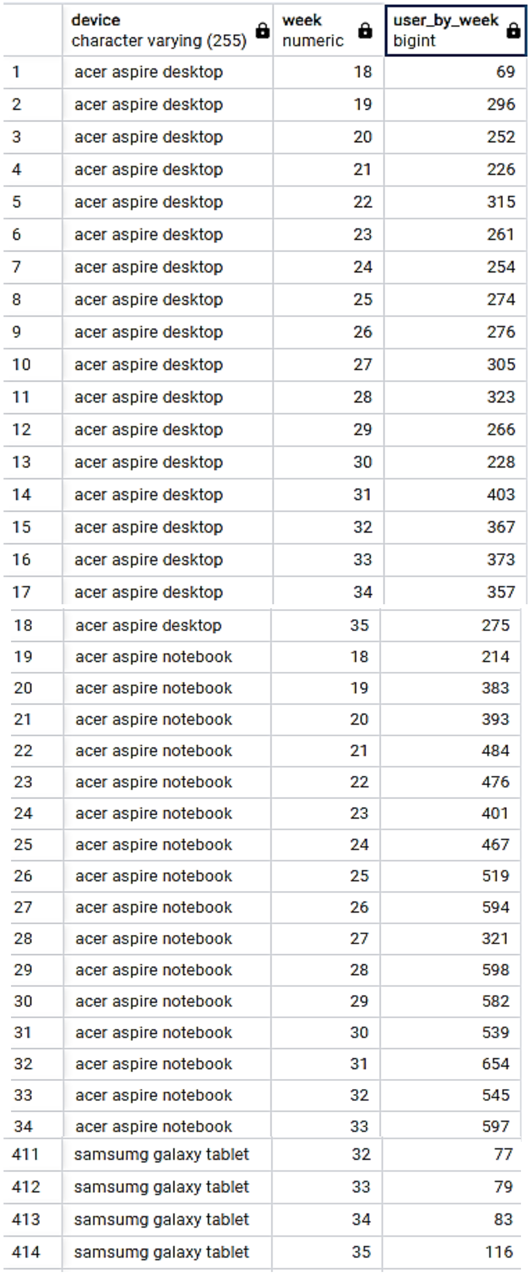
first_week	week_0	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8	week_9	week_10	week_11	week_12	week_13	week_14	week_15
18	81	64	27	19	15	19	13	10	7	9	9	9	8	9	7	5
19	160	104	67	45	34	22	18	24	11	17	12	11	14	10	9	9
20	186	144	77	64	40	26	20	19	23	18	15	15	13	11	8	10
21	177	121	79	50	40	28	21	32	23	23	23	18	18	10	9	0
22	186	117	78	53	34	24	33	30	18	18	13	14	15	9	0	0
23	197	133	83	60	49	42	29	25	26	17	17	12	7	0	0	0
24	198	146	85	56	44	41	34	28	22	21	14	10	0	0	0	0
25	222	135	89	57	41	31	28	25	15	17	12	0	0	0	0	0
26	210	151	100	62	44	30	24	19	15	15	0	0	0	0	0	0
27	199	130	82	60	43	34	33	26	14	0	0	0	0	0	0	0
28	223	152	95	83	52	39	26	23	0	0	0	0	0	0	0	0
29	215	144	91	52	33	19	20	0	0	0	0	0	0	0	0	0
30	228	155	82	59	40	31	0	0	0	0	0	0	0	0	0	0
31	234	154	94	64	47	0	0	0	0	0	0	0	0	0	0	0
32	189	126	69	48	0	0	0	0	0	0	0	0	0	0	0	0
33	250	163	82	0	0	0	0	0	0	0	0	0	0	0	0	0
34	259	173	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	266	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

From 2nd week user retention is almost half of the 1st week

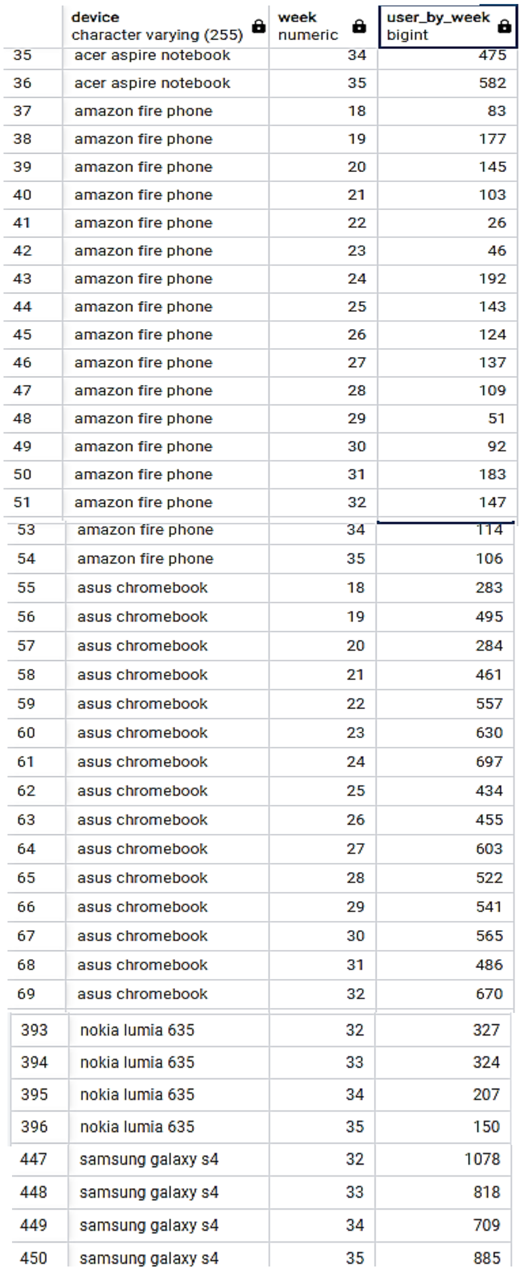
4.Calculate the weekly engagement per device?



```
SELECT
    device,
    EXTRACT('week' from occured_at) as week,
    count(*) as user_by_week
FROM events
WHERE event_type = 'engagement'
GROUP BY week,device
ORDER BY device
```



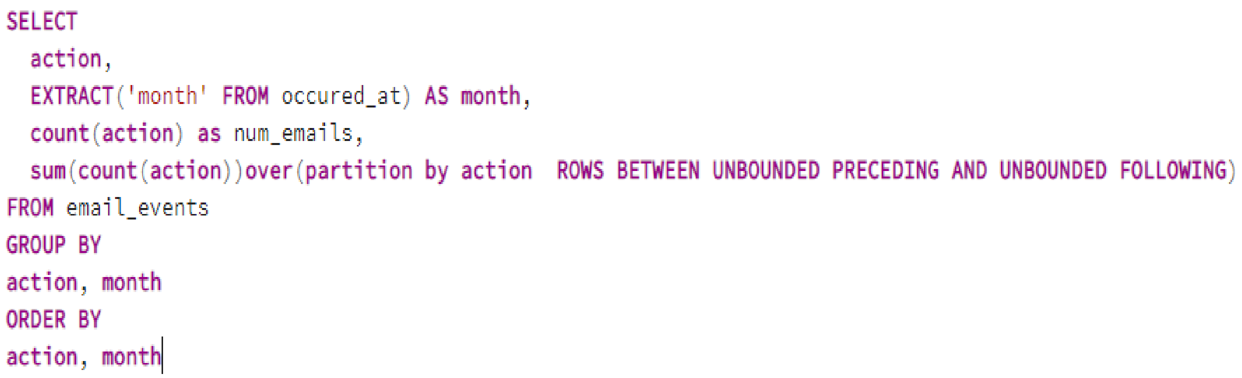
	device character varying (255)	week numeric	user_by_week bigint
1	acer aspire desktop	18	69
2	acer aspire desktop	19	296
3	acer aspire desktop	20	252
4	acer aspire desktop	21	226
5	acer aspire desktop	22	315
6	acer aspire desktop	23	261
7	acer aspire desktop	24	254
8	acer aspire desktop	25	274
9	acer aspire desktop	26	276
10	acer aspire desktop	27	305
11	acer aspire desktop	28	323
12	acer aspire desktop	29	266
13	acer aspire desktop	30	228
14	acer aspire desktop	31	403
15	acer aspire desktop	32	367
16	acer aspire desktop	33	373
17	acer aspire desktop	34	357
18	acer aspire desktop	35	275
19	acer aspire notebook	18	214
20	acer aspire notebook	19	383
21	acer aspire notebook	20	393
22	acer aspire notebook	21	484
23	acer aspire notebook	22	476
24	acer aspire notebook	23	401
25	acer aspire notebook	24	467
26	acer aspire notebook	25	519
27	acer aspire notebook	26	594
28	acer aspire notebook	27	321
29	acer aspire notebook	28	598
30	acer aspire notebook	29	582
31	acer aspire notebook	30	539
32	acer aspire notebook	31	654
33	acer aspire notebook	32	545
34	acer aspire notebook	33	597
411	samsumg galaxy tablet	32	77
412	samsumg galaxy tablet	33	79
413	samsumg galaxy tablet	34	83
414	samsumg galaxy tablet	35	116



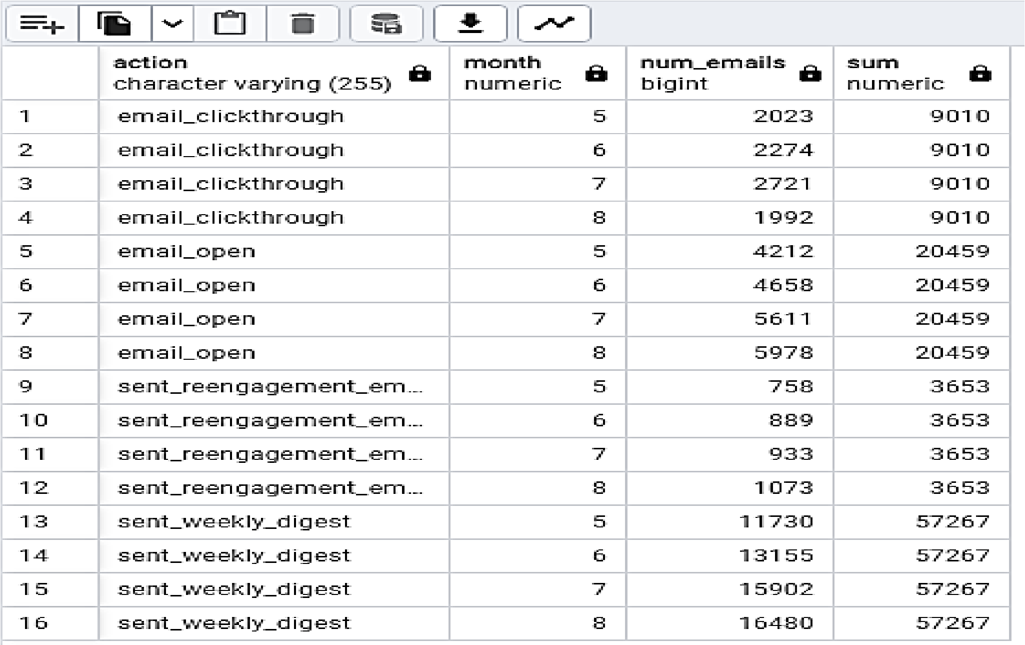
	device character varying (255)	week numeric	user_by_week bigint
35	acer aspire notebook	34	475
36	acer aspire notebook	35	582
37	amazon fire phone	18	83
38	amazon fire phone	19	177
39	amazon fire phone	20	145
40	amazon fire phone	21	103
41	amazon fire phone	22	26
42	amazon fire phone	23	46
43	amazon fire phone	24	192
44	amazon fire phone	25	143
45	amazon fire phone	26	124
46	amazon fire phone	27	137
47	amazon fire phone	28	109
48	amazon fire phone	29	51
49	amazon fire phone	30	92
50	amazon fire phone	31	183
51	amazon fire phone	32	147
53	amazon fire phone	34	114
54	amazon fire phone	35	106
55	asus chromebook	18	283
56	asus chromebook	19	495
57	asus chromebook	20	284
58	asus chromebook	21	461
59	asus chromebook	22	557
60	asus chromebook	23	630
61	asus chromebook	24	697
62	asus chromebook	25	434
63	asus chromebook	26	455
64	asus chromebook	27	603
65	asus chromebook	28	522
66	asus chromebook	29	541
67	asus chromebook	30	565
68	asus chromebook	31	486
69	asus chromebook	32	670
393	nokia lumia 635	32	327
394	nokia lumia 635	33	324
395	nokia lumia 635	34	207
396	nokia lumia 635	35	150
447	samsung galaxy s4	32	1078
448	samsung galaxy s4	33	818
449	samsung galaxy s4	34	709
450	samsung galaxy s4	35	885

Number of user per device per week decline from 32 Week for mobile and tablet

5.Calculate the email engagement metrics?



```
SELECT
  action,
  EXTRACT('month' FROM occured_at) AS month,
  count(action) as num_emails,
  sum(count(action))over(partition by action  ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
FROM email_events
GROUP BY
action, month
ORDER BY
action, month
```



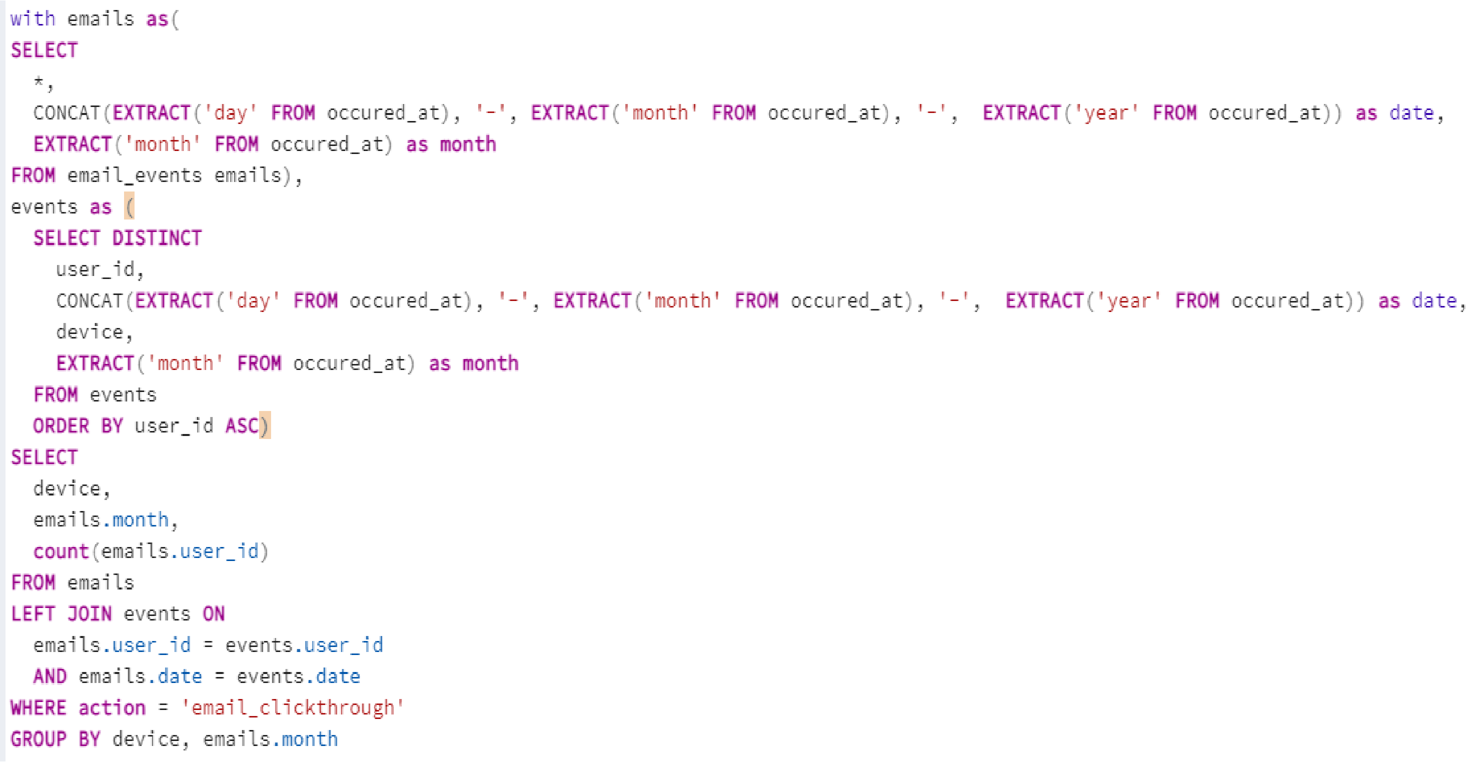
	action character varying (255)	month numeric	num_emails bigint	sum numeric
1	email_clickthrough	5	2023	9010
2	email_clickthrough	6	2274	9010
3	email_clickthrough	7	2721	9010
4	email_clickthrough	8	1992	9010
5	email_open	5	4212	20459
6	email_open	6	4658	20459
7	email_open	7	5611	20459
8	email_open	8	5978	20459
9	sent_reengagement_em...	5	758	3653
10	sent_reengagement_em...	6	889	3653
11	sent_reengagement_em...	7	933	3653
12	sent_reengagement_em...	8	1073	3653
13	sent_weekly_digest	5	11730	57267
14	sent_weekly_digest	6	13155	57267
15	sent_weekly_digest	7	15902	57267
16	sent_weekly_digest	8	16480	57267

there was a steady increase in the number of emails sent

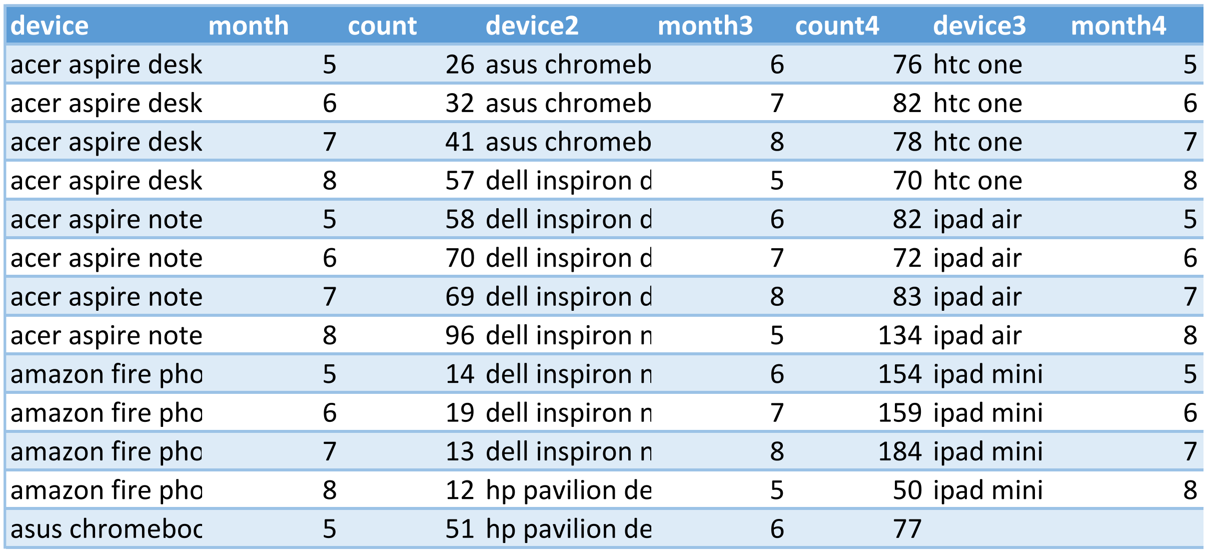
(weekly digest and re-engagement) and emails opened

but but there was a decrease in clickthrough rates. From July to August,

there was a 6.5% increase in emails open but a 27% decline in clickthrough rates.



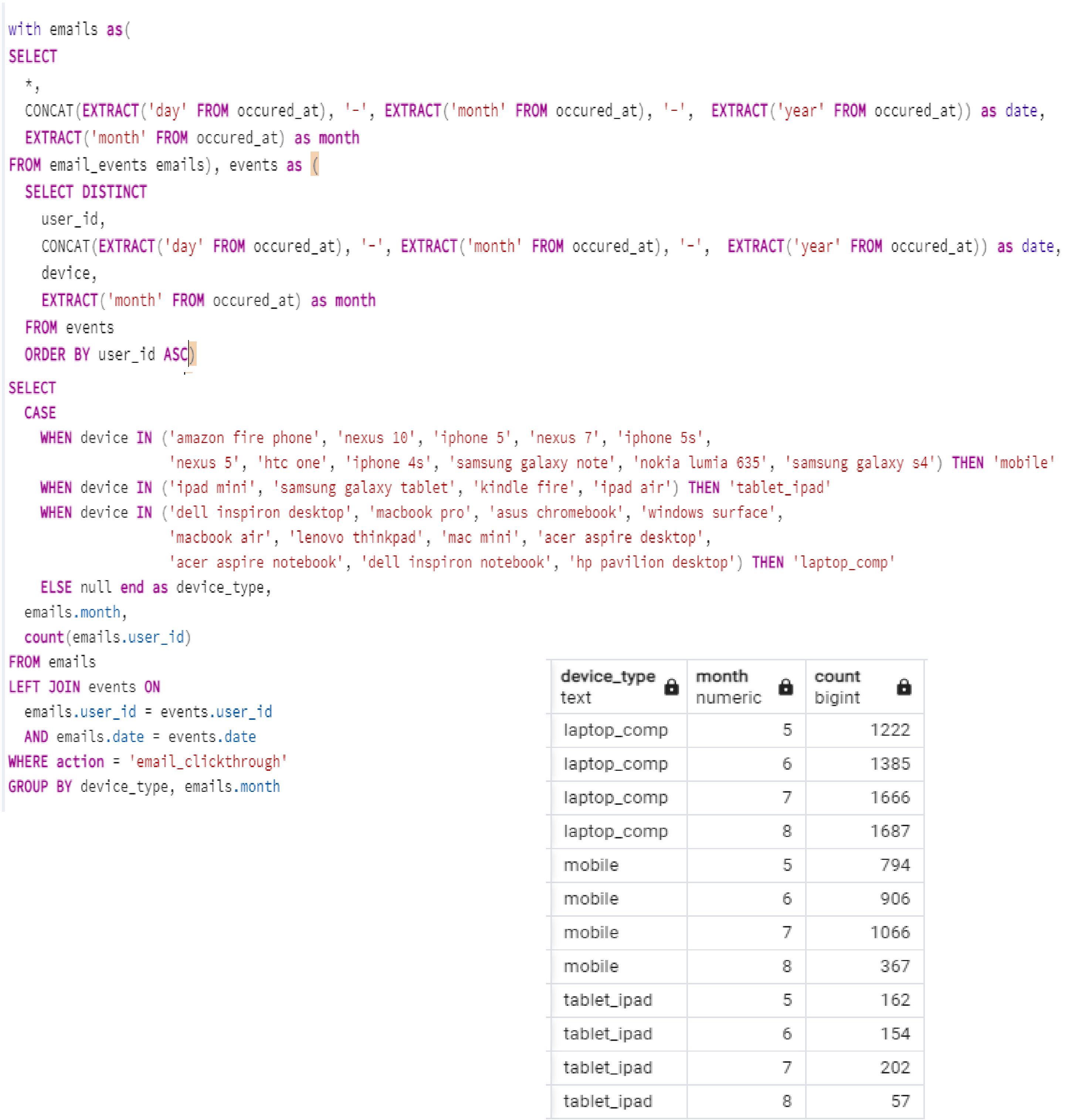
```
with emails as(
SELECT
  *,
  CONCAT(EXTRACT('day' FROM occured_at), '-', EXTRACT('month' FROM occured_at), '-',  EXTRACT('year' FROM occured_at)) as date,
  EXTRACT('month' FROM occured_at) as month
FROM email_events emails),
events as (
  SELECT DISTINCT
    user_id,
    CONCAT(EXTRACT('day' FROM occured_at), '-', EXTRACT('month' FROM occured_at), '-',  EXTRACT('year' FROM occured_at)) as date,
    device,
    EXTRACT('month' FROM occured_at) as month
  FROM events
  ORDER BY user_id ASC)
SELECT
  device,
  emails.month,
  count(emails.user_id)
FROM emails
LEFT JOIN events ON
  emails.user_id = events.user_id
  AND emails.date = events.date
WHERE action = 'email_clickthrough'
GROUP BY device, emails.month
```



device	month	count	device2	month3	count4	device3	month4
acer aspire desk	5	26	asus chromeb	6	76	htc one	5
acer aspire desk	6	32	asus chromeb	7	82	htc one	6
acer aspire desk	7	41	asus chromeb	8	78	htc one	7
acer aspire desk	8	57	dell inspiron d	5	70	htc one	8
acer aspire note	5	58	dell inspiron d	6	82	ipad air	5
acer aspire note	6	70	dell inspiron d	7	72	ipad air	6
acer aspire note	7	69	dell inspiron d	8	83	ipad air	7
acer aspire note	8	96	dell inspiron n	5	134	ipad air	8
amazon fire pho	5	14	dell inspiron n	6	154	ipad mini	5
amazon fire pho	6	19	dell inspiron n	7	159	ipad mini	6
amazon fire pho	7	13	dell inspiron n	8	184	ipad mini	7
amazon fire pho	8	12	hp pavilion de	5	50	ipad mini	8
asus chromeboc	5	51	hp pavilion de	6	77		

Using the query above, I noticed that the clickthrough rates on laptops and computers were stable from July to August,

but not the tablets and cellphones



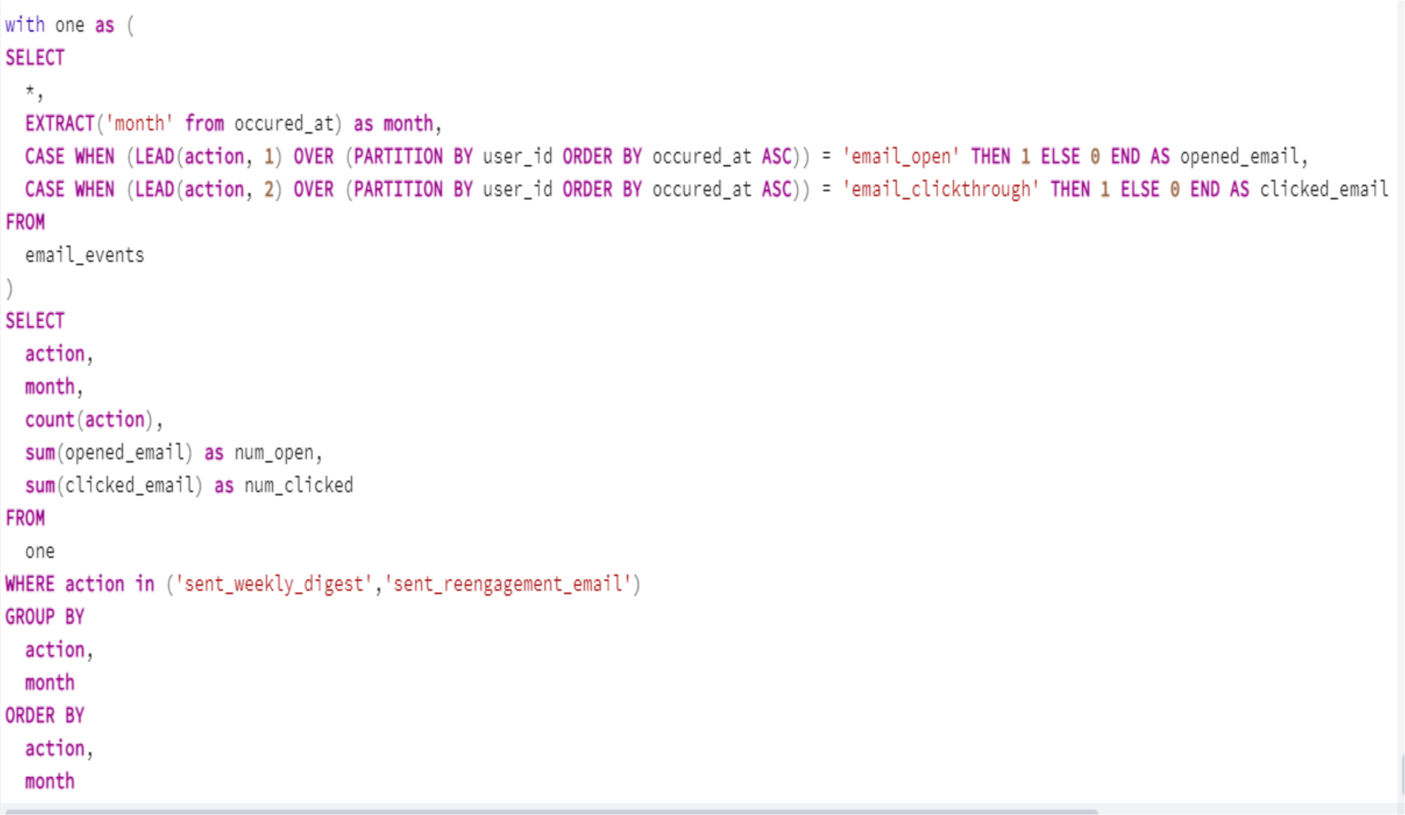
```
with emails as(
SELECT
  *,
  CONCAT(EXTRACT('day' FROM occured_at), '-', EXTRACT('month' FROM occured_at), '-',  EXTRACT('year' FROM occured_at)) as date,
  EXTRACT('month' FROM occured_at) as month
FROM email_events emails), events as (
  SELECT DISTINCT
    user_id,
    CONCAT(EXTRACT('day' FROM occured_at), '-', EXTRACT('month' FROM occured_at), '-',  EXTRACT('year' FROM occured_at)) as date,
    device,
    EXTRACT('month' FROM occured_at) as month
  FROM events
  ORDER BY user_id ASC)
SELECT
  CASE
    WHEN device IN ('amazon fire phone', 'nexus 10', 'iphone 5', 'nexus 7', 'iphone 5s',
                    'nexus 5', 'htc one', 'iphone 4s', 'samsung galaxy note', 'nokia lumia 635', 'samsung galaxy s4') THEN 'mobile'
    WHEN device IN ('ipad mini', 'samsung galaxy tablet', 'kindle fire', 'ipad air') THEN 'tablet_ipad'
    WHEN device IN ('dell inspiron desktop', 'macbook pro', 'asus chromebook', 'windows surface',
                    'macbook air', 'lenovo thinkpad', 'mac mini', 'acer aspire desktop',
                    'acer aspire notebook', 'dell inspiron notebook', 'hp pavilion desktop') THEN 'laptop_comp'
    ELSE null end as device_type,
  emails.month,
  count(emails.user_id)
FROM emails
LEFT JOIN events ON
  emails.user_id = events.user_id
  AND emails.date = events.date
WHERE action = 'email_clickthrough'
GROUP BY device_type, emails.month
```

device_type text	month numeric	count bigint
laptop_comp	5	1222
laptop_comp	6	1385
laptop_comp	7	1666
laptop_comp	8	1687
mobile	5	794
mobile	6	906
mobile	7	1066
mobile	8	367
tablet_ipad	5	162
tablet_ipad	6	154
tablet_ipad	7	202
tablet_ipad	8	57

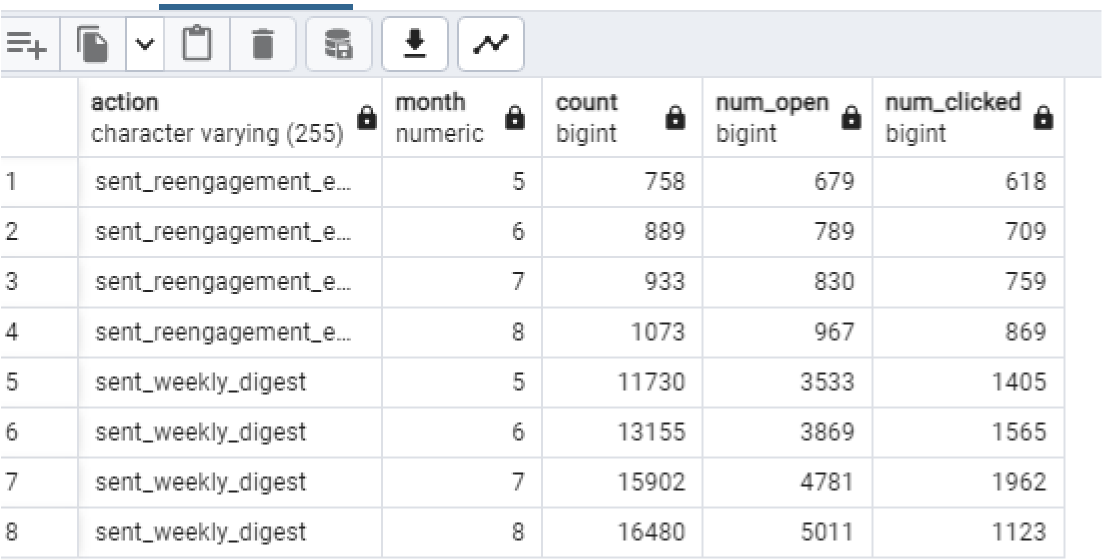
it seems to be the case that the drop in clickthrough rates was attributed specifically to mobile devices and tablets.

lack of engagement is due to a decrease in email clickthrough rates from July to August. To gather more information,

Let’s see if we can narrow the problem even further by email type.



```
with one as (
SELECT
  *,
  EXTRACT('month' from occured_at) as month,
  CASE WHEN (LEAD(action, 1) OVER (PARTITION BY user_id ORDER BY occured_at ASC)) = 'email_open' THEN 1 ELSE 0 END AS opened_email,
  CASE WHEN (LEAD(action, 2) OVER (PARTITION BY user_id ORDER BY occured_at ASC)) = 'email_clickthrough' THEN 1 ELSE 0 END AS clicked_email
FROM
  email_events
)
SELECT
  action,
  month,
  count(action),
  sum(opened_email) as num_open,
  sum(clicked_email) as num_clicked
FROM
  one
WHERE action in ('sent_weekly_digest','sent_reengagement_email')
GROUP BY
  action,
  month
ORDER BY
  action,
  month
```



	action	month	count	num_open	num_clicked
	character varying (255)	numeric	bigint	bigint	bigint
1	sent_reengagement_e…	5	758	679	618
2	sent_reengagement_e…	6	889	789	709
3	sent_reengagement_e…	7	933	830	759
4	sent_reengagement_e…	8	1073	967	869
5	sent_weekly_digest	5	11730	3533	1405
6	sent_weekly_digest	6	13155	3869	1565
7	sent_weekly_digest	7	15902	4781	1962
8	sent_weekly_digest	8	16480	5011	1123

◆ Result:

- In this project we understand how to use SIX step process of Data analysis.
- Project help to understand how to use SQL joins as data store in different table for simple data analytics job.
- Project help to understand how to use date and window functions
- Key project insight are:
    - Due to lack of data first case is difficult to analyze still we are able to find out number of share for each language

    - Persian language has higher per share of 37.5% while other languages have 12.5%

    - The drop in engagement was mainly attributed to a drop in five engagement events (home_page, like_message_ view_inbox, send_message, and login)

    - Decrease in events was caused by a reduction in total active users from August, as well as a decrease in engagement per user.

    - After I took a close look at the emails table and I noticed that there was a significant decrease in click-through rates from July to August even though there was an increase in the number of emails opened.

    - By segmenting the clickthrough rates by device type (mobile, tablet, laptop), I noticed that the drop in clickthrough rates was attributed to mobile and tablet devices.

    - The decline in click-through rates is attributed to the weekly digest email and not the re-engagement email.

    - It’s possible that there’s a technical problem, making it difficult for users to click the email or simply a UX problem, where the content and layout of the email are not enticing users to click. A good first step would be to see what changes have been made from July to August and working backward.