Case Study 2(Investigating metric spike)

Project Description

In this project we are trying to analyse the given dataset and draw key insights from it. I am given data for users,events,email_events data.

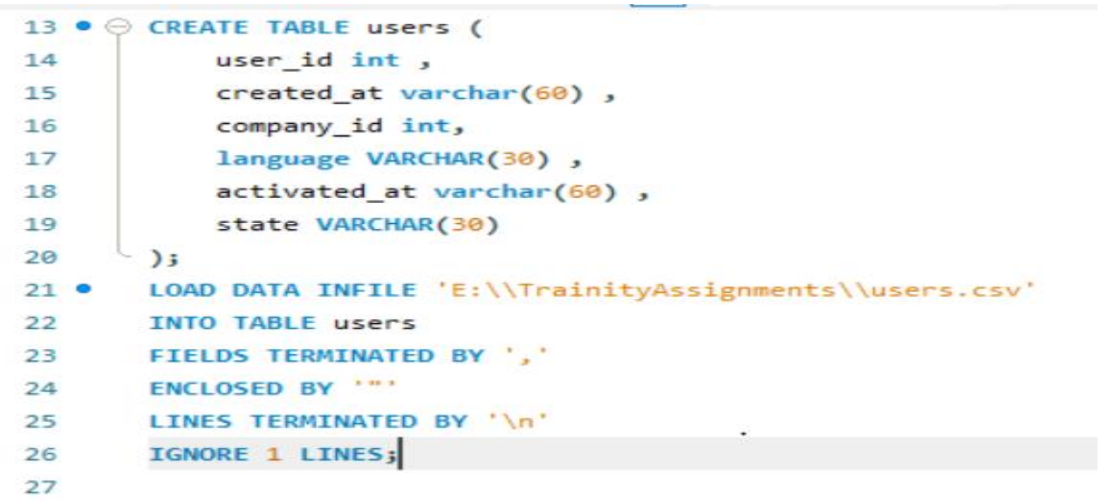
Users table tells us about descriptive information about that user's account like userid,when the user was created for the first time, when the user was activated, current state of the user i.e activated or inactive, the language the user speaks.

Events table tells us about descriptive information about an event where an event is an action that a user has taken. This tells us which user has taken what type of event,event name, when he has performed the action and the device from which user has performed the action. This also gives the location from which the action was performed. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.

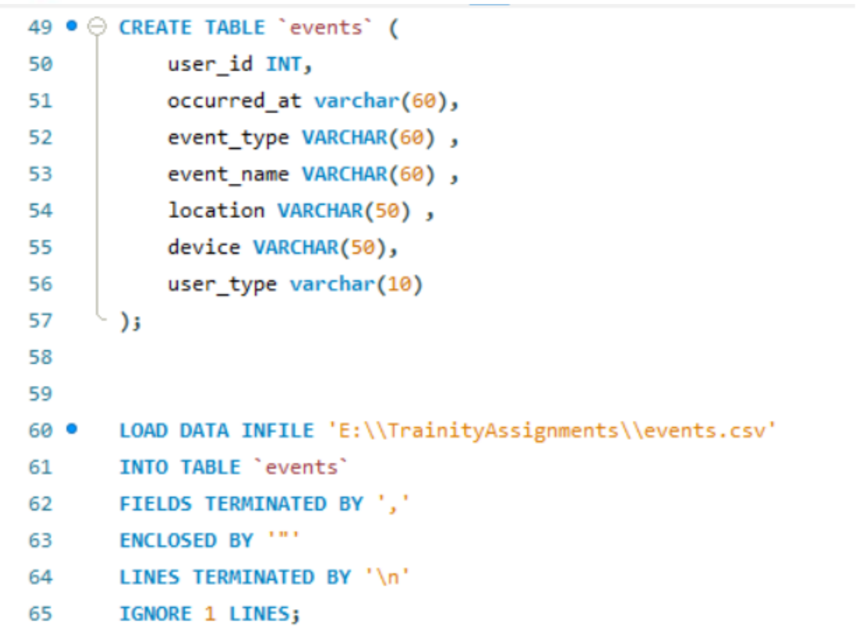
Email_Events table tells us about descriptive information about events related to emails only.

Approach

The csv datas were given. I have created the tables. Then I have loaded data from the csv files.

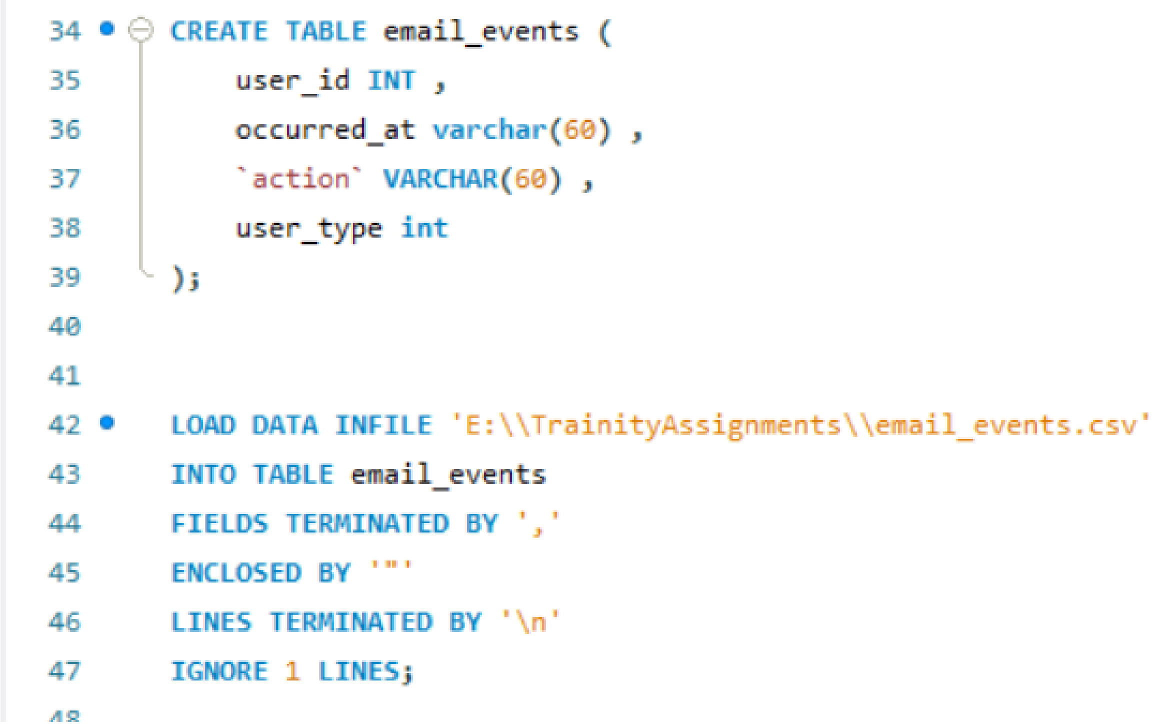


```
CREATE TABLE users (
        user_id int ,
        created_at varchar(60) ,
        company_id int,
        language VARCHAR(30) ,
        activated_at varchar(60) ,
        state VARCHAR(30)
);
LOAD DATA INFILE 'E:\\TrainityAssignments\\users.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```



```
CREATE TABLE `events` (
    user_id INT,
    occurred_at varchar(60),
    event_type VARCHAR(60) ,
    event_name VARCHAR(60) ,
    location VARCHAR(50) ,
    device VARCHAR(50),
    user_type varchar(10)
);


LOAD DATA INFILE 'E:\\TrainityAssignments\\events.csv'
INTO TABLE `events`
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```



```sql
CREATE TABLE email_events (
    user_id INT ,
    occurred_at varchar(60) ,
    `action` VARCHAR(60) ,
    user_type int
);


LOAD DATA INFILE 'E:\\TrainityAssignments\\email_events.csv'
INTO TABLE email_events
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

I was having some issues loading the fields related to date fields so I had to take date fields as varchar same goes for user_type.

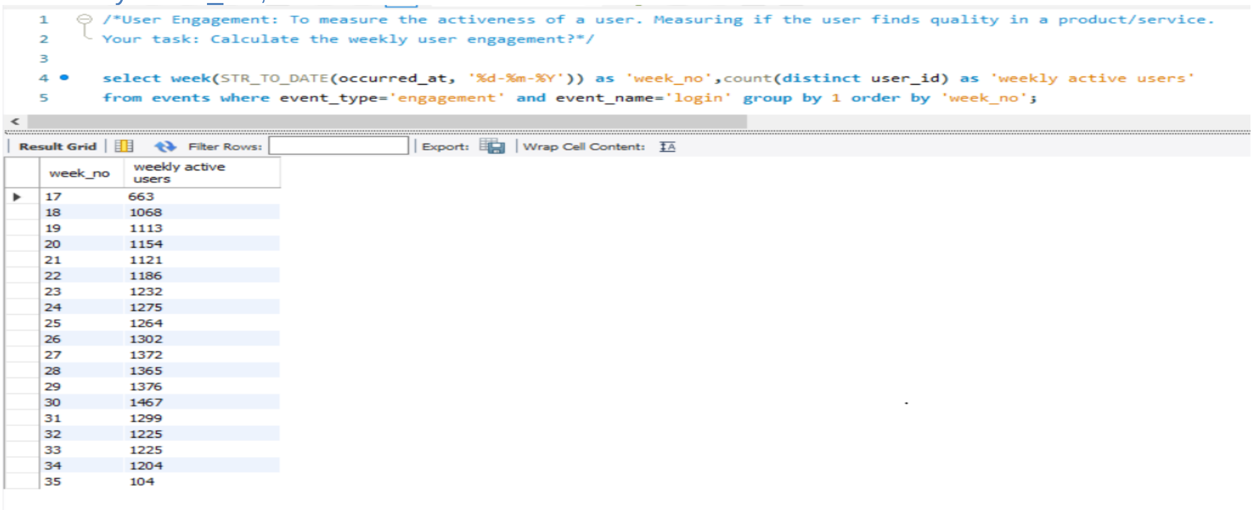
Tech-Stack Used

MySQL Server as the database, MySQL WORKBENCH as a query editor.
G-Drive to upload the assignment.

Insights

1. User Engagement: To measure the activeness of a user. Measuring if the userfinds quality in a product/service. We have to calculate the weekly user engagement.

select week(STR_TO_DATE(occurred_at, '%d-%m-%Y')) as 'week_no',count(distinct user_id) as 'weekly active users' from events where event_type='engagement' and event_name='login' group by 1 order by 'week_no';



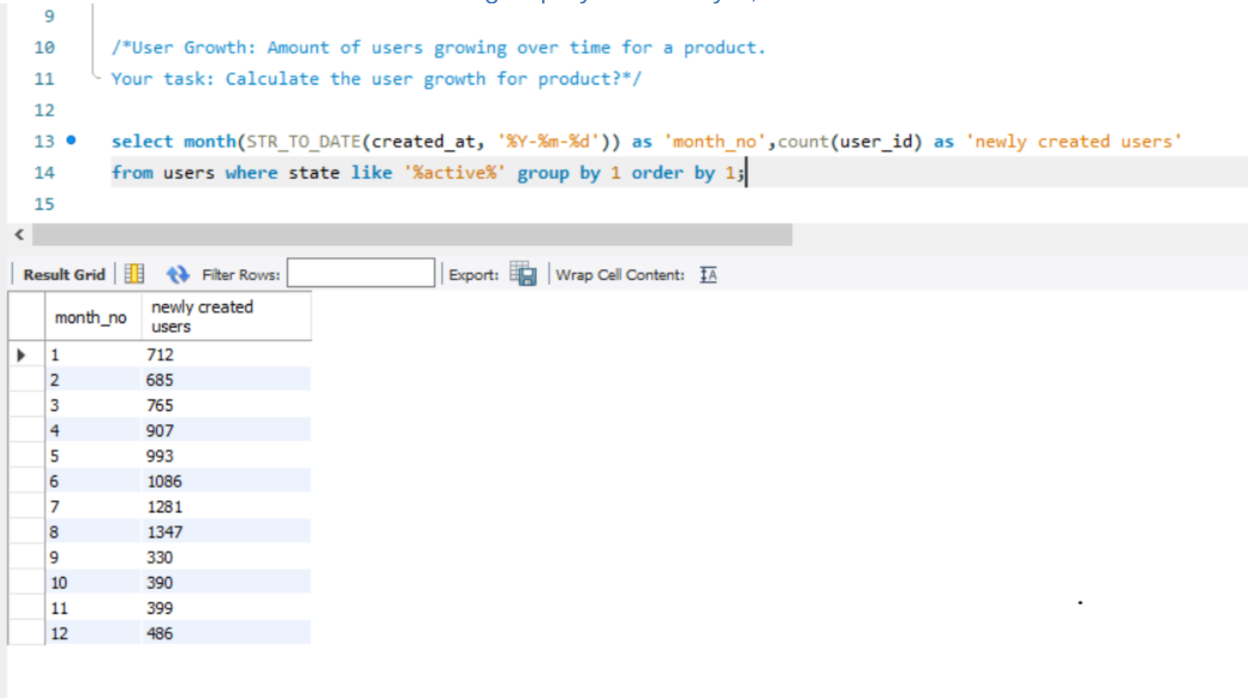
```sql
/*User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.
Your task: Calculate the weekly user engagement?*/

select week(STR_TO_DATE(occurred_at, '%d-%m-%Y')) as 'week_no',count(distinct user_id) as 'weekly active users'
from events where event_type='engagement' and event_name='login' group by 1 order by 'week_no';
```

week_no	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

2. User Growth: Number of users growing over time for a product. We have to calculate the user growth for product. – Here I am calculating growth per month

```
select month(STR_TO_DATE(created_at, '%Y-%m-%d')) as 'month_no',count(user_id) as 'newly created users'
from users where state like '%active%' group by 1 order by 1;
```



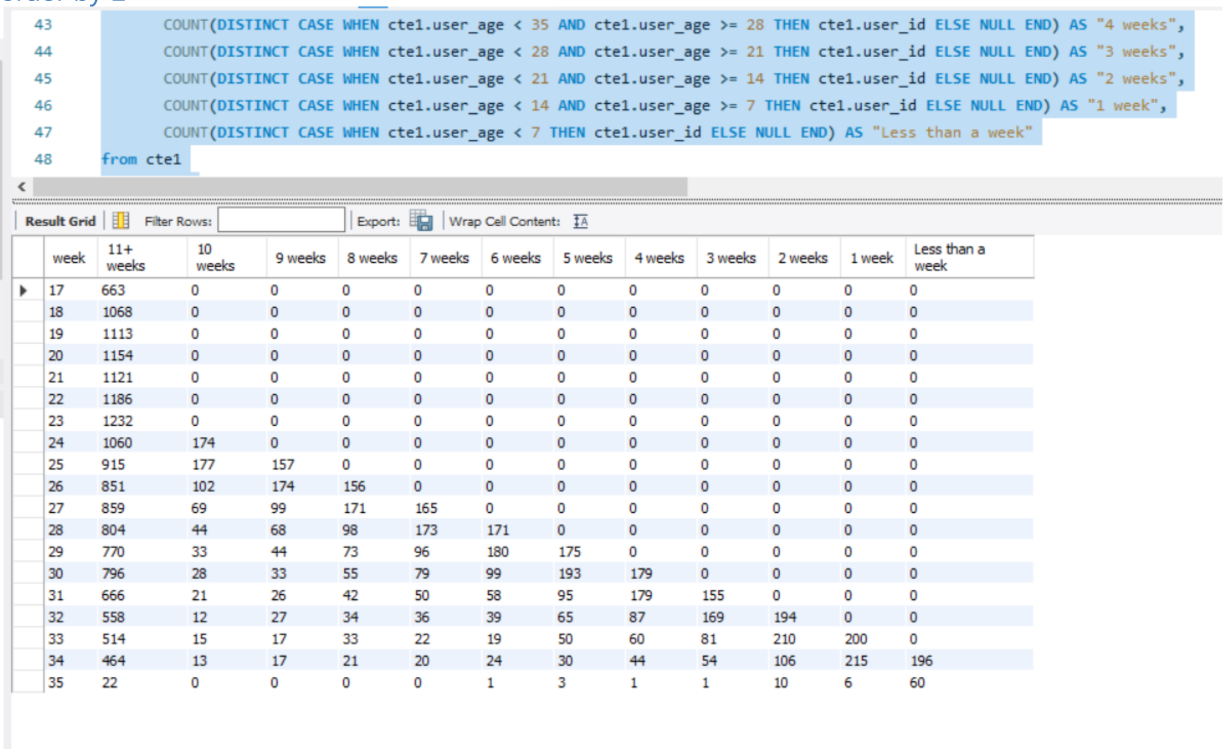
```
/*User Growth: Amount of users growing over time for a product.
Your task: Calculate the user growth for product?*/

select month(STR_TO_DATE(created_at, '%Y-%m-%d')) as 'month_no',count(user_id) as 'newly created users'
from users where state like '%active%' group by 1 order by 1;
```

month_no	newly created users
1	712
2	685
3	765
4	907
5	993
6	1086
7	1281
8	1347
9	330
10	390
11	399
12	486

3. Weekly Retention: Users getting retained weekly after signing-up for a product.We have to calculate the weekly retention of users-sign up cohort.

```
with cte1 as(
select u.user_id,e.occurred_at as occurred_at,u.activated_at,
               week(str_to_date(u.activated_at,'%Y-%m-%d')) AS activation_week,
               datediff(str_to_date(e.occurred_at,'%d-%m-%Y'),str_to_date(u.activated_at,'%Y-%m-%d')) AS age_at_event,
               datediff('2014-09-01 00:00' , str_to_date(u.activated_at,'%Y-%m-%d')) AS user_age
          FROM users u
          JOIN events e
            ON e.user_id = u.user_id
           AND e.event_type = 'engagement'
           AND e.event_name = 'login'
           #AND e.occurred_at >= '01-05-2014'
           #AND e.occurred_at < '01-09-2014'
         WHERE u.activated_at  <>''
         order by user_id)
SELECT week(str_to_date(cte1.occurred_at,'%d-%m-%Y')) AS "week",
       #AVG(cte1.age_at_event) AS "Average age during week",
       COUNT(DISTINCT CASE WHEN cte1.user_age > 77 THEN cte1.user_id ELSE NULL END) AS "11+ weeks",
       COUNT(DISTINCT CASE WHEN cte1.user_age < 77 AND cte1.user_age >= 70 THEN cte1.user_id ELSE NULL END) AS "10 weeks",
       COUNT(DISTINCT CASE WHEN cte1.user_age < 70 AND cte1.user_age >= 63 THEN cte1.user_id ELSE NULL END) AS "9 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 63 AND cte1.user_age >= 56 THEN cte1.user_id ELSE NULL END) AS "8 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 56 AND cte1.user_age >= 49 THEN cte1.user_id ELSE NULL END) AS "7 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 49 AND cte1.user_age >= 42 THEN cte1.user_id ELSE NULL END) AS "6 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 42 AND cte1.user_age >= 35 THEN cte1.user_id ELSE NULL END) AS "5 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 35 AND cte1.user_age >= 28 THEN cte1.user_id ELSE NULL END) AS "4 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 28 AND cte1.user_age >= 21 THEN cte1.user_id ELSE NULL END) AS "3 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 21 AND cte1.user_age >= 14 THEN cte1.user_id ELSE NULL END) AS "2 weeks",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 14 AND cte1.user_age >= 7 THEN cte1.user_id ELSE NULL END) AS "1 week",
	COUNT(DISTINCT CASE WHEN cte1.user_age < 7 THEN cte1.user_id ELSE NULL END) AS "Less than a week"
from cte1
group by 1
order by 1
```

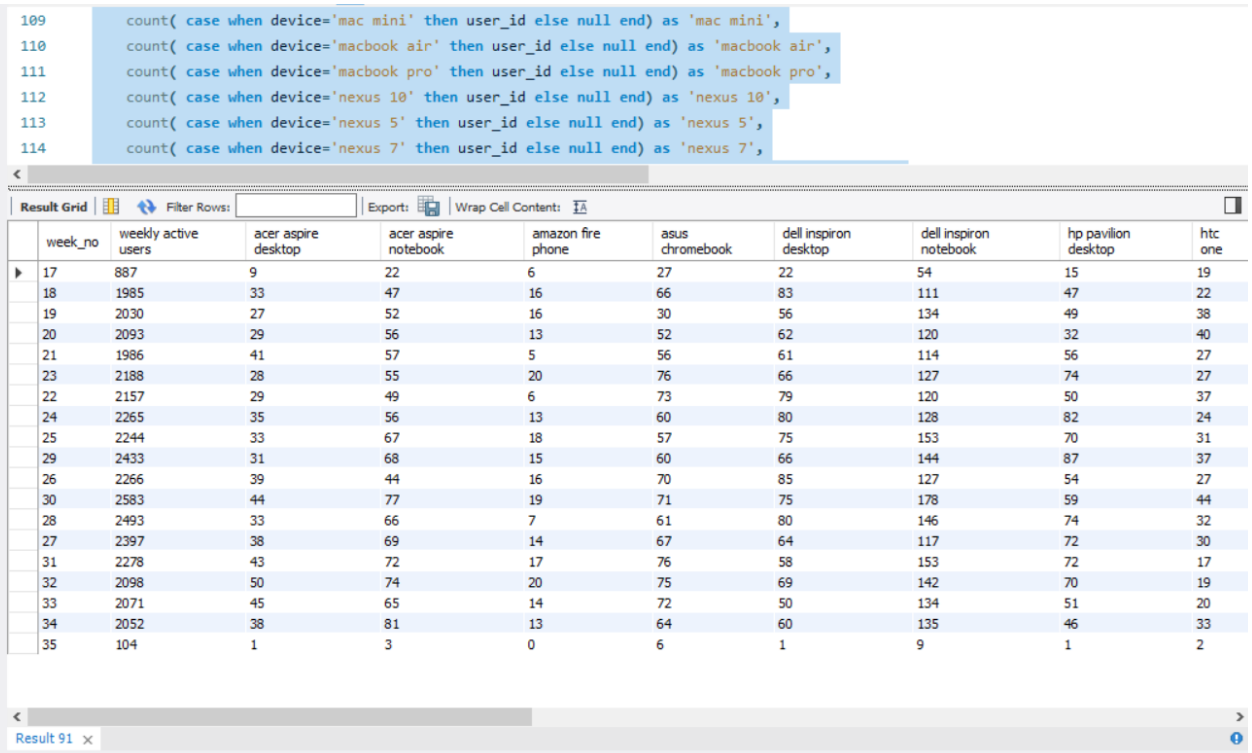


```
        COUNT(DISTINCT CASE WHEN cte1.user_age < 35 AND cte1.user_age >= 28 THEN cte1.user_id ELSE NULL END) AS "4 weeks",
        COUNT(DISTINCT CASE WHEN cte1.user_age < 28 AND cte1.user_age >= 21 THEN cte1.user_id ELSE NULL END) AS "3 weeks",
        COUNT(DISTINCT CASE WHEN cte1.user_age < 21 AND cte1.user_age >= 14 THEN cte1.user_id ELSE NULL END) AS "2 weeks",
        COUNT(DISTINCT CASE WHEN cte1.user_age < 14 AND cte1.user_age >= 7 THEN cte1.user_id ELSE NULL END) AS "1 week",
        COUNT(DISTINCT CASE WHEN cte1.user_age < 7 THEN cte1.user_id ELSE NULL END) AS "Less than a week"
from cte1
```

week	11+ weeks	10 weeks	9 weeks	8 weeks	7 weeks	6 weeks	5 weeks	4 weeks	3 weeks	2 weeks	1 week	Less than a week
17	663	0	0	0	0	0	0	0	0	0	0	0
18	1068	0	0	0	0	0	0	0	0	0	0	0
19	1113	0	0	0	0	0	0	0	0	0	0	0
20	1154	0	0	0	0	0	0	0	0	0	0	0
21	1121	0	0	0	0	0	0	0	0	0	0	0
22	1186	0	0	0	0	0	0	0	0	0	0	0
23	1232	0	0	0	0	0	0	0	0	0	0	0
24	1060	174	0	0	0	0	0	0	0	0	0	0
25	915	177	157	0	0	0	0	0	0	0	0	0
26	851	102	174	156	0	0	0	0	0	0	0	0
27	859	69	99	171	165	0	0	0	0	0	0	0
28	804	44	68	98	173	171	0	0	0	0	0	0
29	770	33	44	73	96	180	175	0	0	0	0	0
30	796	28	33	55	79	99	193	179	0	0	0	0
31	666	21	26	42	50	58	95	179	155	0	0	0
32	558	12	27	34	36	39	65	87	169	194	0	0
33	514	15	17	33	22	19	50	60	81	210	200	0
34	464	13	17	21	20	24	30	44	54	106	215	196
35	22	0	0	0	0	1	3	1	1	10	6	60

4. Weekly Engagement: To measure the activeness of a user. Measuring if the userfinds quality in a product/service weekly. We have to calculate the weeklyengagement per device.

```
select
        week(STR_TO_DATE(occurred_at, '%d-%m-%Y')) as 'week_no',
        count( user_id) as 'weekly active users',
   count( case when device='acer aspire desktop' then  user_id else null end) as 'acer aspire desktop',
   count( case when device='acer aspire notebook' then  user_id else null end) as 'acer aspire notebook',
   count( case when device='amazon fire phone' then user_id else null end) as 'amazon fire phone',
   count( case when device='asus chromebook' then user_id else null end) as 'asus chromebook',
   count( case when device='dell inspiron desktop' then user_id else null end) as 'dell inspiron desktop',
   count( case when device='dell inspiron notebook' then user_id else null end) as 'dell inspiron notebook',
   count( case when device='hp pavilion desktop' then user_id else null end) as 'hp pavilion desktop',
   count( case when device='htc one' then user_id else null end) as 'htc one',
   count( case when device='ipad air' then user_id else null end) as 'ipad air',
   count( case when device='ipad mini' then user_id else null end) as 'ipad mini',
   count( case when device='iphone 4s' then user_id else null end) as 'iphone 4s',
   count( case when device='iphone 5' then user_id else null end) as 'iphone 5',
   count( case when device='iphone 5s' then user_id else null end) as 'iphone 5s',
   count( case when device='kindle fire' then user_id else null end) as 'kindle fire',
   count( case when device='lenovo thinkpad' then user_id else null end) as 'lenovo thinkpad',
   count( case when device='mac mini' then user_id else null end) as 'mac mini',
   count( case when device='macbook air' then user_id else null end) as 'macbook air',
   count( case when device='macbook pro' then user_id else null end) as 'macbook pro',
   count( case when device='nexus 10' then user_id else null end) as 'nexus 10',
   count( case when device='nexus 5' then user_id else null end) as 'nexus 5',
   count( case when device='nexus 7' then user_id else null end) as 'nexus 7',
   count( case when device='nokia lumia 635' then user_id else null end) as 'nokia lumia 635',
   count( case when device='samsumg galaxy tablet' then user_id else null end) as 'samsumg galaxy tablet',
   count( case when device='samsung galaxy note' then user_id else null end) as 'samsung galaxy note',
   count( case when device='samsung galaxy s4' then user_id else null end) as 'samsung galaxy s4',
   count( case when device='windows surface' then user_id else null end) as 'windows surface'
from events where event_type='engagement'
and event_name='login'
group by 1
order by 'week_no';
```



```
    count( case when device='mac mini' then user_id else null end) as 'mac mini',
    count( case when device='macbook air' then user_id else null end) as 'macbook air',
    count( case when device='macbook pro' then user_id else null end) as 'macbook pro',
    count( case when device='nexus 10' then user_id else null end) as 'nexus 10',
    count( case when device='nexus 5' then user_id else null end) as 'nexus 5',
    count( case when device='nexus 7' then user_id else null end) as 'nexus 7',
```

week_no	weekly active users	acer aspire desktop	acer aspire notebook	amazon fire phone	asus chromebook	dell inspiron desktop	dell inspiron notebook	hp pavilion desktop	htc one
17	887	9	22	6	27	22	54	15	19
18	1985	33	47	16	66	83	111	47	22
19	2030	27	52	16	30	56	134	49	38
20	2093	29	56	13	52	62	120	32	40
21	1986	41	57	5	56	61	114	56	27
23	2188	28	55	20	76	66	127	74	27
22	2157	29	49	6	73	79	120	50	37
24	2265	35	56	13	60	80	128	82	24
25	2244	33	67	18	57	75	153	70	31
29	2433	31	68	15	60	66	144	87	37
26	2266	39	44	16	70	85	127	54	27
30	2583	44	77	19	71	75	178	59	44
28	2493	33	66	7	61	80	146	74	32
27	2397	38	69	14	67	64	117	72	30
31	2278	43	72	17	76	58	153	72	17
32	2098	50	74	20	75	69	142	70	19
33	2071	45	65	14	72	50	134	51	20
34	2052	38	81	13	64	60	135	46	33
35	104	1	3	0	6	1	9	1	2

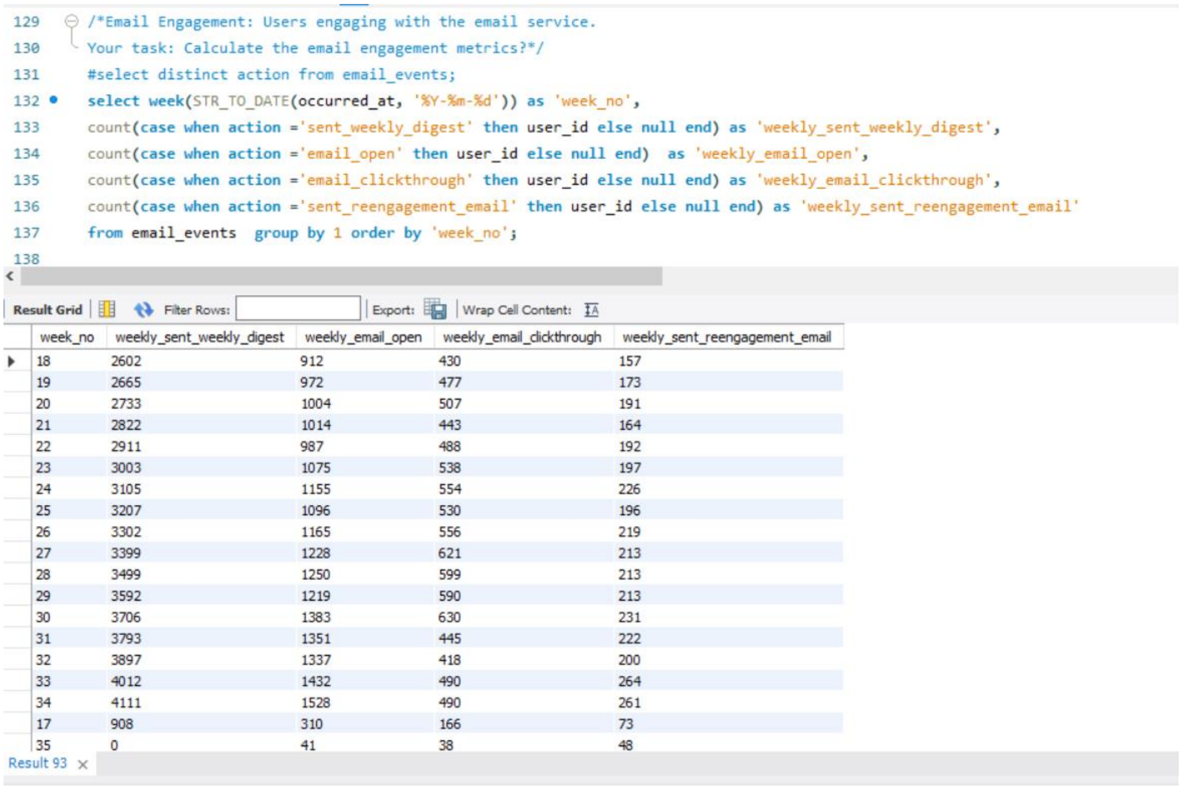
Full Results : Results

UsersBydeviceTypePerWeek.csv

5. Email Engagement: Users engaging with the email service. We have to calculate the email engagement metrics.

#select distinct action from email_events;

```
select week(STR_TO_DATE(occurred_at, '%Y-%m-%d')) as 'week_no',
count(case when action ='sent_weekly_digest' then user_id else null end) as
'weekly_sent_weekly_digest',
count(case when action ='email_open' then user_id else null end)  as 'weekly_email_open',
count(case when action ='email_clickthrough' then user_id else null end) as
'weekly_email_clickthrough',
count(case when action ='sent_reengagement_email' then user_id else null end) as
'weekly_sent_reengagement_email'
from email_events  group by 1 order by 'week_no';
```



```sql
/*Email Engagement: Users engaging with the email service.
Your task: Calculate the email engagement metrics?*/
#select distinct action from email_events;
select week(STR_TO_DATE(occurred_at, '%Y-%m-%d')) as 'week_no',
count(case when action ='sent_weekly_digest' then user_id else null end) as 'weekly_sent_weekly_digest',
count(case when action ='email_open' then user_id else null end)  as 'weekly_email_open',
count(case when action ='email_clickthrough' then user_id else null end) as 'weekly_email_clickthrough',
count(case when action ='sent_reengagement_email' then user_id else null end) as 'weekly_sent_reengagement_email'
from email_events  group by 1 order by 'week_no';
```

week_no	weekly_sent_weekly_digest	weekly_email_open	weekly_email_clickthrough	weekly_sent_reengagement_email
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
17	908	310	166	73
35	0	41	38	48

Result:

1. I was able to create meaningful insights from the dataset which can be crucial for business. I was able to sharpen my SQL skills from here. Successfully completing this project boosted my confidence.
2. Created the database from scratch, loading huge data (above 3lakhs) boosted my confidence
3. Wrote many complex queries. Faced many issues like I was not able to load the date fields so had to change the data type and load data then had to convert the string to date filed in each time