Project Description:

- This project explains operations of a company for its better growth.
- It analysis various aspects of the company to work on areas of improvement.
- As a data analyst, we can deep dive into the database of the company and bring out some insights for the company's future.

Approach:

- I have created the needed database (Project3) and tables in MySQL workbench and imported the entire csv file in the database as tables so that we could analyse the data.
- We write MySQL queries to extract information that is needed for better analysis as the data is huge.
- The tables that are created:
    1. Job_Data
    2. Events
    3. Email_Events
    4. Users

Insights:

- We are going to find few insights from the data like-
- Number of jobs reviewed.
- Calculating the percentage share of each language over a period
- Analysing the throughput using window function
- User engagement in a week
- Engagement per device on a weekly basis and more.

Result:

- The project has helped me learn MySQL queries in a very vast manner.
- It expanded my knowledge and improved the ability to think and analyse the data.
- It has helped me to draw conclusions and the requirements for a better understanding of the business growth.
- It has helped me to better understand the MySql Workbench.

MySQL queries and output Screenshots:

Jobs Reviewed over time----------

```sql
SELECT
    ds,
    COUNT(job_id) AS num_of_jobs,
    COUNT(*) / 24 AS jobs_per_hr,
    SUM(time_spent) / 3600 AS hrs_spent
FROM
    job_data
WHERE
    ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds;
```

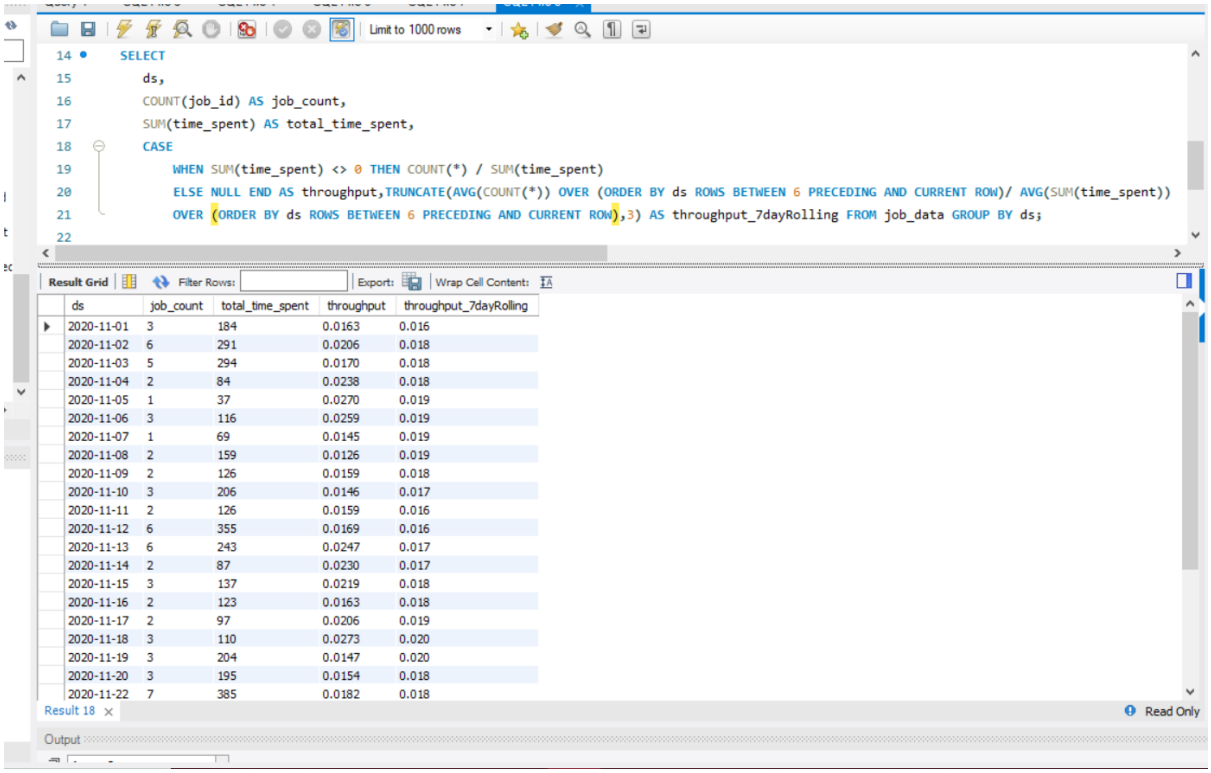
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	num_of_jobs	jobs_per_hour	hours_spent
2020-11-30	4	0.1667	0.0575
2020-11-29	3	0.1250	0.0306
2020-11-28	5	0.2083	0.0469
2020-11-27	4	0.1667	0.0500
2020-11-26	6	0.2500	0.0961
2020-11-25	2	0.0833	0.0367
2020-11-15	3	0.1250	0.0381
2020-11-02	6	0.2500	0.0808
2020-11-20	3	0.1250	0.0542
2020-11-01	3	0.1250	0.0511
2020-11-22	7	0.2917	0.1069
2020-11-16	2	0.0833	0.0342
2020-11-13	6	0.2500	0.0675
2020-11-05	1	0.0417	0.0103
2020-11-09	2	0.0833	0.0350
2020-11-19	3	0.1250	0.0567

Result 14

The query shows number of hours in a day spent on number of jobs in last one month.

7 day Rolling average:



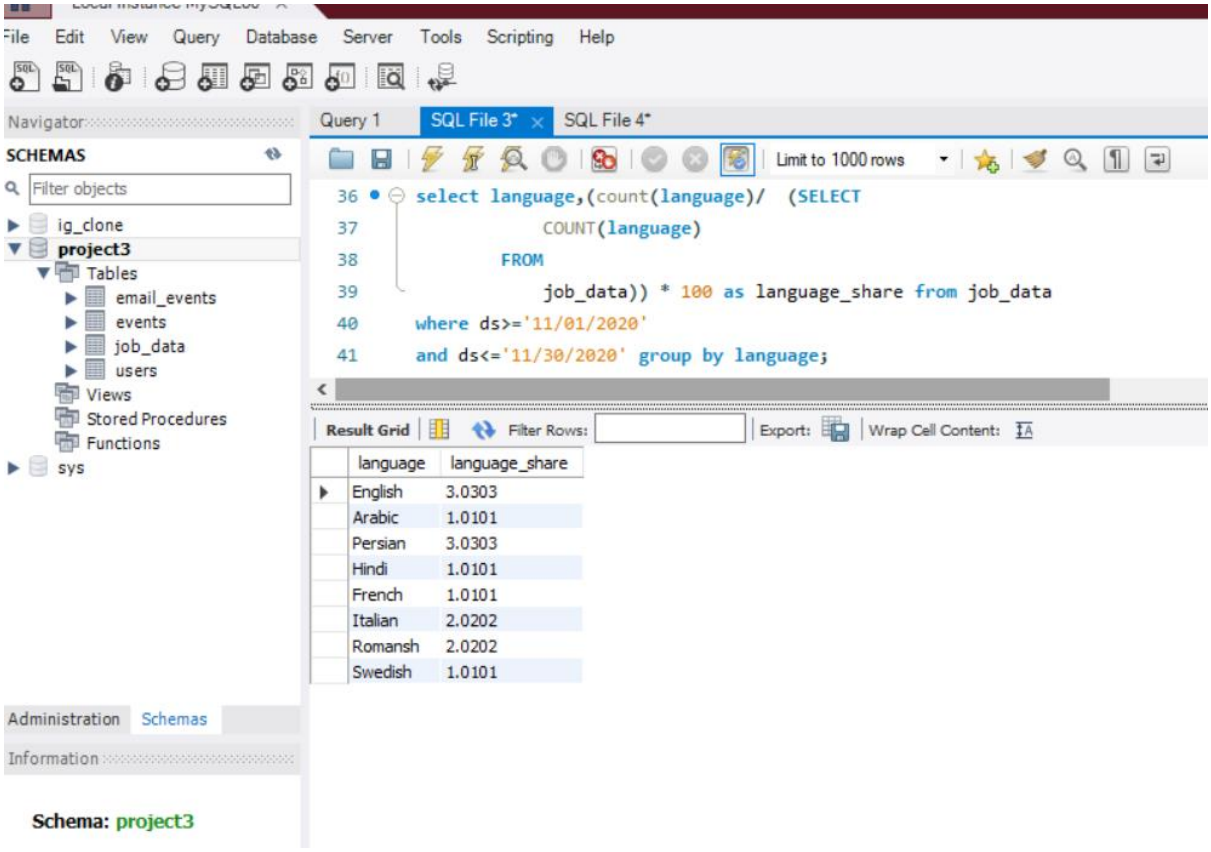
```
SELECT
    ds,
    COUNT(job_id) AS job_count,
    SUM(time_spent) AS total_time_spent,
    CASE
        WHEN SUM(time_spent) <> 0 THEN COUNT(*) / SUM(time_spent)
        ELSE NULL END AS throughput,TRUNCATE(AVG(COUNT(*)) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)/ AVG(SUM(time_spent))
        OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW),3) AS throughput_7dayRolling FROM job_data GROUP BY ds;
```

Result Grid

ds	job_count	total_time_spent	throughput	throughput_7dayRolling
2020-11-01	3	184	0.0163	0.016
2020-11-02	6	291	0.0206	0.018
2020-11-03	5	294	0.0170	0.018
2020-11-04	2	84	0.0238	0.018
2020-11-05	1	37	0.0270	0.019
2020-11-06	3	116	0.0259	0.019
2020-11-07	1	69	0.0145	0.019
2020-11-08	2	159	0.0126	0.019
2020-11-09	2	126	0.0159	0.018
2020-11-10	3	206	0.0146	0.017
2020-11-11	2	126	0.0159	0.016
2020-11-12	6	355	0.0169	0.016
2020-11-13	6	243	0.0247	0.017
2020-11-14	2	87	0.0230	0.017
2020-11-15	3	137	0.0219	0.018
2020-11-16	2	123	0.0163	0.018
2020-11-17	2	97	0.0206	0.019
2020-11-18	3	110	0.0273	0.020
2020-11-19	3	204	0.0147	0.020
2020-11-20	3	195	0.0154	0.018
2020-11-22	7	385	0.0182	0.018

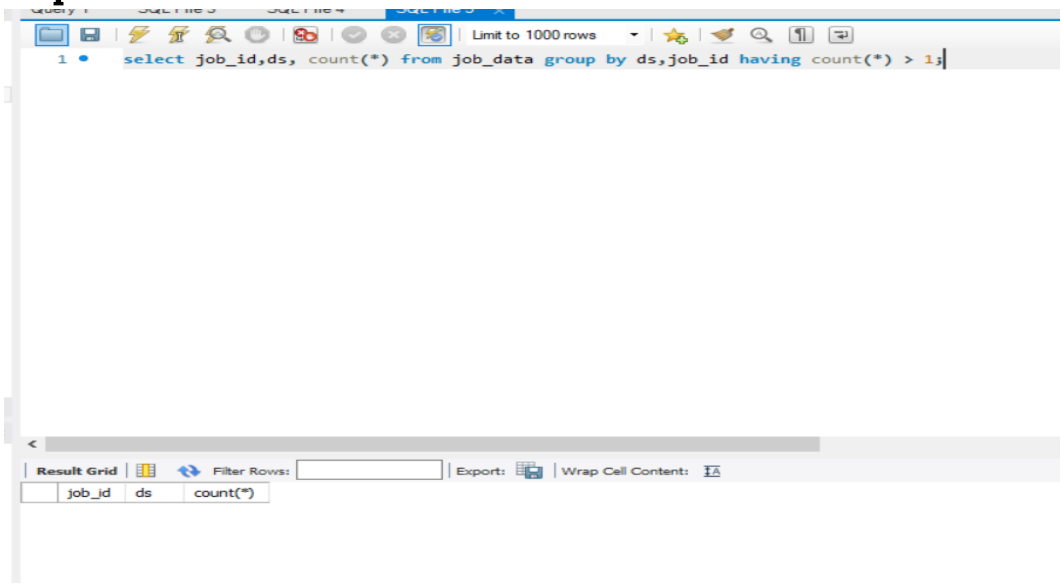
Here we calculate the 7 day rolling average. This is better for huge set of data. Hence this would be preferred more.

Language Share Analysis----------



Using this query we can find the share of each language from job_data

Duplicate Rows Detection----------



There are no duplicate rows which is a good sign of a good data.

Weekly User Engagement----------

```sql
SELECT
    EXTRACT(WEEK FROM occured_at) AS weeknum,
    COUNT(DISTINCT user_id) AS weekly_user_engmnt
FROM
    events a
GROUP BY weeknum;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

weeknum	weekly_user_engmnt
17	740
18	1260
19	1287
20	1351
21	1299
22	1381
23	1446
24	1471
25	1459
26	1509
27	1573
28	1577
29	1607
30	1706
31	1514
32	1454
33	1438
34	1443
35	118

This query shows how much users are engaged in a week.

User Growth Analysis----------

```sql
SELECT year, weeknum, num_active_user,
SUM(num_active_user)OVER(ORDER BY year,weeknum
ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_active_users FROM
(
SELECT EXTRACT(year from a.activated_at) AS year,
EXTRACT(week from a.activated_at) AS weeknum,
COUNT(DISTINCT user_id) AS num_active_user FROM users a
 GROUP BY year, weeknum ORDER BY year, weeknum ) a;
```

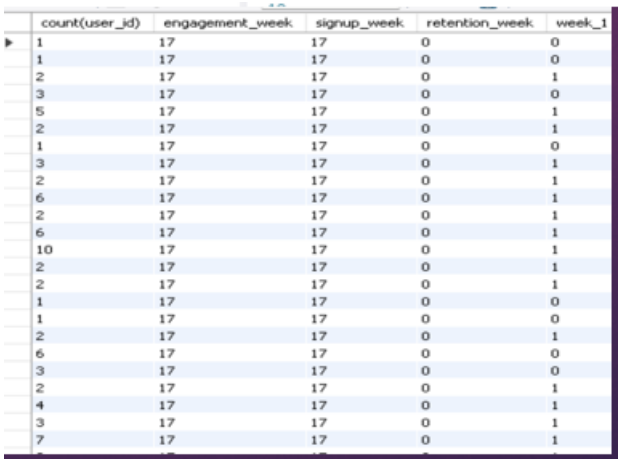
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	weeknum	num_active_user	cum_active_users
NULL	NULL	9685	9685
2013	0	23	9708
2013	1	30	9738
2013	2	48	9786
2013	3	36	9822
2013	4	30	9852
2013	5	48	9900
2013	6	38	9938
2013	7	42	9980
2013	8	34	10014
2013	9	43	10057
2013	10	32	10089
2013	11	31	10120
2013	12	33	10153
2013	13	39	10192
2013	14	35	10227
2013	15	43	10270
2013	16	46	10316

This query finds the growth of number of users for the product.

Weekly Retention----------

```sql
select
count(user_id),engagement_week,signup_week,retention_week,sum(case when retention_week = 1 then 1 else 0 end) as week_1
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 occured_at) as signup_week from events where event_type = 'signup_flow'
and event_name = 'complete_signup' and extract(week from occured_at) ) a left join( select distinct  user_id,
extract(week from occured_at) as engagement_week
from events where event_type = 'engagement'
) b on a.user_id = b.user_id)order by
a.user_id
) c group by user_id;
```



count(user_id)	engagement_week	signup_week	retention_week	week_1
1	17	17	0	0
1	17	17	0	0
2	17	17	0	1
3	17	17	0	0
5	17	17	0	1
2	17	17	0	1
1	17	17	0	0
3	17	17	0	1
2	17	17	0	1
6	17	17	0	1
2	17	17	0	1
6	17	17	0	1
10	17	17	0	1
2	17	17	0	1
2	17	17	0	1
1	17	17	0	0
1	17	17	0	0
2	17	17	0	1
6	17	17	0	0
3	17	17	0	0
2	17	17	0	1
4	17	17	0	1
3	17	17	0	1
7	17	17	0	1

This query shows how many users retain after signing up for the cohort that is shown by engagement week.

Weekly Engagament per Device----------

```
-- Weekly Engagement Per Device:
SELECT
EXTRACT(year FROM occurred_at) AS year,
EXTRACT(week from occurred_at) AS week,
device,
COUNT(distinct user_id) as weekly_engmnt_count
FROM events WHERE event_type ='engagement' GROUP BY 1,2,3 ORDER by 1,2,3;
```

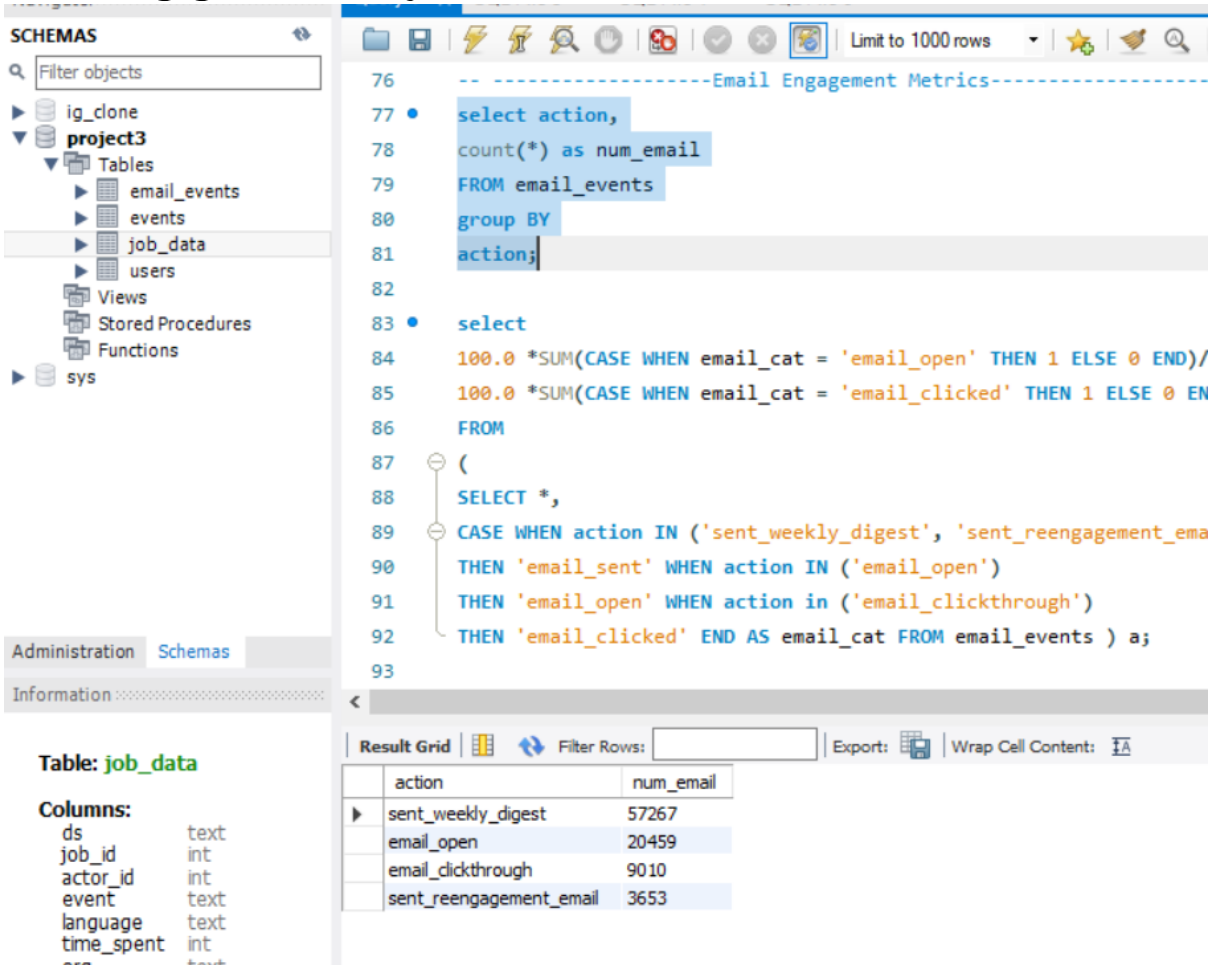
year	week	device	weekly_engmnt_count
2014	17	acer aspire desktop	9
2014	17	acer aspire notebook	20
2014	17	amazon fire phone	4
2014	17	asus chromebook	21
2014	17	dell inspiron desktop	18
2014	17	dell inspiron notebook	46
2014	17	hp pavilion desktop	14
2014	17	htc one	16
2014	17	ipad air	27
2014	17	ipad mini	19
2014	17	iphone 4s	21
2014	17	iphone 5	65
2014	17	iphone 5s	42
2014	17	kindle fire	6
2014	17	lenovo thinkpad	86
2014	17	mac mini	6
2014	17	macbook air	54
2014	17	macbook pro	143
2014	17	nexus 10	16
2014	17	nexus 5	40
2014	17	nexus 7	18
2014	17	nokia lumia 635	17
2014	17	samsumg galaxy tablet	8

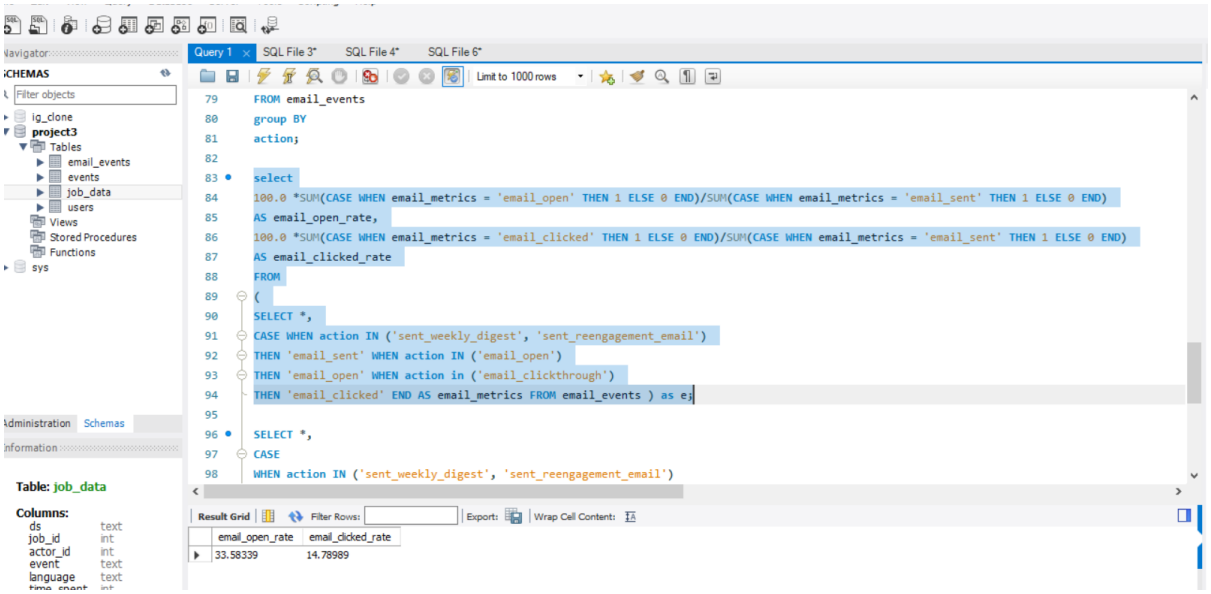
year	week	device	weekly_engmnt_count
2014	18	acer aspire notebook	33
2014	18	amazon fire phone	9
2014	18	asus chromebook	42
2014	18	dell inspiron desktop	58
2014	18	dell inspiron notebook	77
2014	18	hp pavilion desktop	37
2014	18	htc one	19
2014	18	ipad air	52
2014	18	ipad mini	30
2014	18	iphone 4s	46
2014	18	iphone 5	113
2014	18	iphone 5s	73
2014	18	kindle fire	27
2014	18	lenovo thinkpad	153
2014	18	mac mini	13
2014	18	macbook air	121
2014	18	macbook pro	252
2014	18	nexus 10	30
2014	18	nexus 5	73
2014	18	nexus 7	30
2014	18	nokia lumia 635	33
2014	18	samsumg galaxy tablet	11
2014	18	samsung galaxy note	15

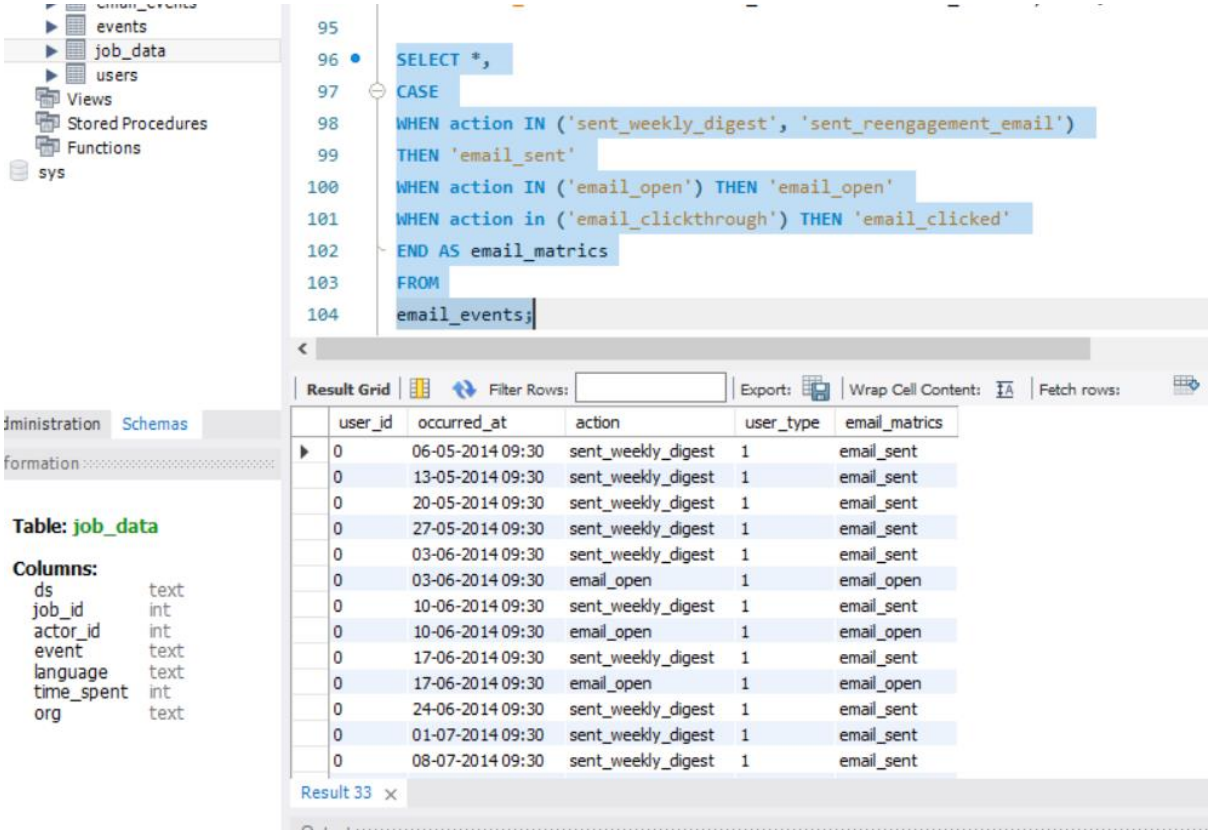
year	week	device	weekly_engmnt_count
2014	19	dell inspiron notebook	83
2014	19	hp pavilion desktop	40
2014	19	htc one	30
2014	19	ipad air	55
2014	19	ipad mini	36
2014	19	iphone 4s	44
2014	19	iphone 5	115
2014	19	iphone 5s	79
2014	19	kindle fire	21
2014	19	lenovo thinkpad	178
2014	19	mac mini	18
2014	19	macbook air	112
2014	19	macbook pro	266
2014	19	nexus 10	25
2014	19	nexus 5	87
2014	19	nexus 7	41
2014	19	nokia lumia 635	23
2014	19	samsumg galaxy tablet	6
2014	19	samsung galaxy note	11
2014	19	samsung galaxy s4	91
2014	19	windows surface	16
2014	20	acer aspire desktop	23
2014	20	acer aspire notebook	40

In this query, weekly engagement per device is calculated, if its more the users are satisfied with the product.

Email Engagement Analysis----------







This result gives us an overview of the engagement of the users with emails.

CODE FOR CREATING THE TABLE AND IMPORTING DATA

```
CREATE database Project3;
Use project3;

Create Table job_data(
ds DATE,
job_id INT NOT NULL,
actor_id INT NOT NULL,
event Varchar(10) NOT NULL,
language Varchar(15) NOT NULL,
time_spent INT NOT NULL,
org CHAR(3)
);


CREATE TABLE users
(
        user_id int null,
    created_at datetime null,
    company_id int null,
    language VARCHAR(10) null,
   activated_at datetime null,
    state varchar(25) null
    );
```

```
CREATE TABLE events
(
       user_id int null,
    occured_at datetime null,
    event_type VARCHAR(100) null,
    event_name VARCHAR(100) null,
   location varchar(215) null,
    device varchar(215) null,
    user_type int null
    );


CREATE TABLE email_events
(
       user_id int null,
    occurred_at datetime null,
    action VARCHAR(225) null,
    user_type int null
    );

LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/My SQLServer
8.0/Uploads/events.csv'
INTO TABLE events
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;


show variables like "secure_file_priv";
```

Load data result:

✓	15	13:20:57	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/email_events.csv' INTO TABLE e...	90389 row(s) affected Records: 90389 Deleted: 0 Skipped: 0 Warnings: 0
✗	16	13:35:08	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv' INTO TABLE users FIE...	Error Code: 1146. Table 'project3.users' doesn't exist
✓	17	13:35:23	CREATE TABLE users (user_id int null, created_at date null, company_id int null, language VARCHAR...	0 row(s) affected
✓	18	13:35:28	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users.csv' INTO TABLE users FIE...	9381 row(s) affected Records: 9381 Deleted: 0 Skipped: 0 Warnings: 0