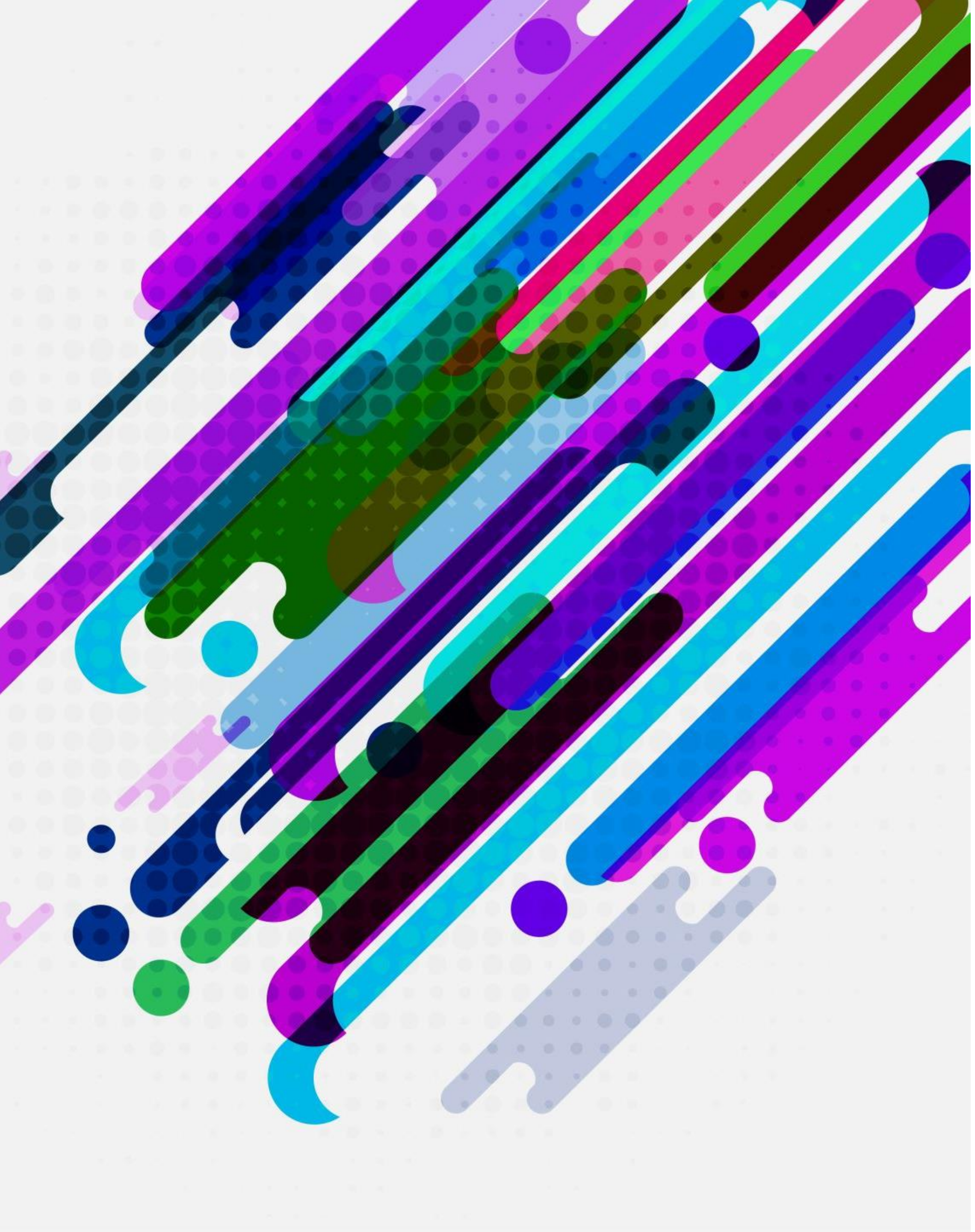


OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

By Anjali Gupta

PROJECT DESCRIPTION

- Operation Analytics is the analysis done for the complete end-to-end operation of a company
- This overall analysis is done to predict the growth or decline of a company fortune

APPROACH

- In this project, we have approached the matter systematically. We first read the problem carefully and then tried to find the possible solution for them.

TECH_STACK USED

- For completing this project we have used MySQL Oracle Workbench

INSIGHTS

The project is divided into two parts:-

- Case Study 1-Job Data
- Case Study 2 – Investigating Metric Spike

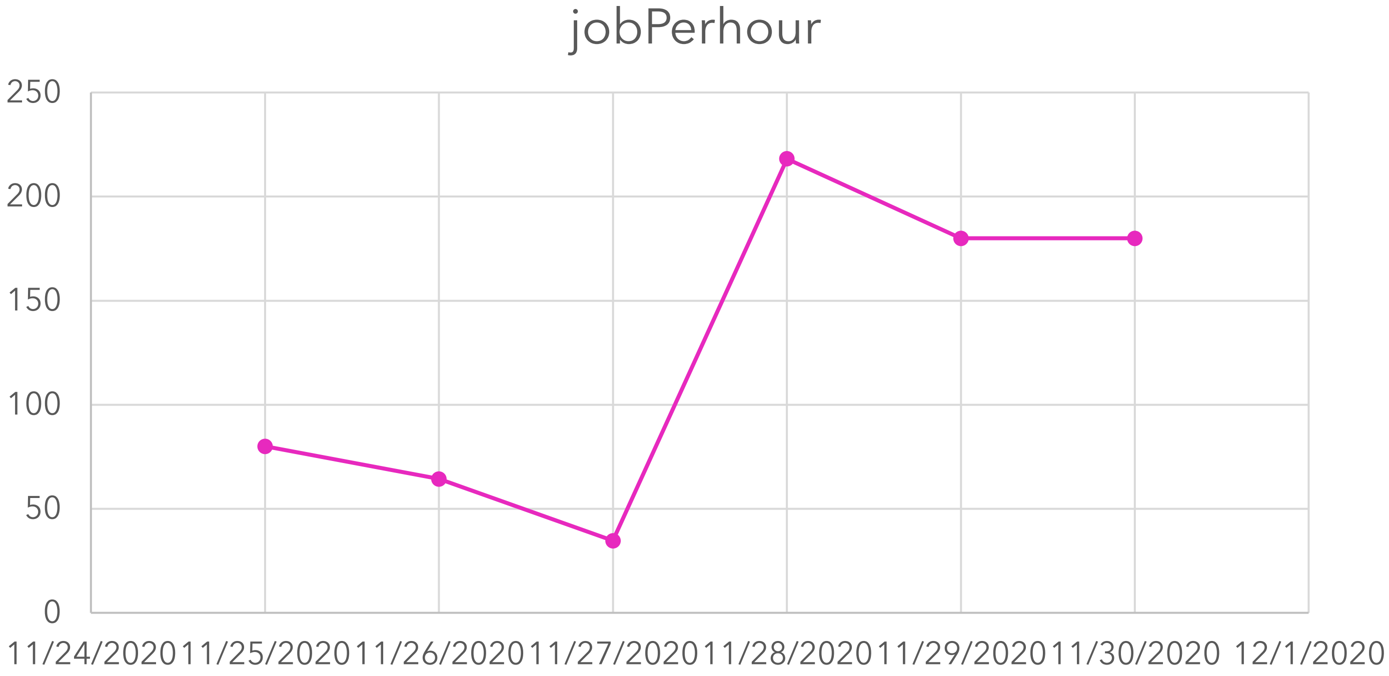
CASE STUDY 1

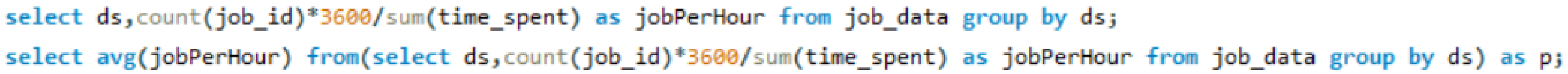
In this case study insights on job data will be provided

Number of jobs reviewed

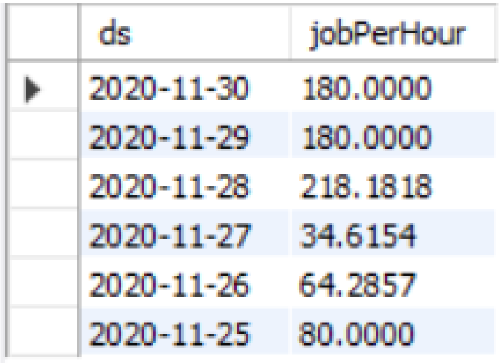
We have to calculate the number of jobs reviewed per hour per day for November.

As we can see from the graph the maximum number of jobs were reviewed on 28 November 2020 as inferred from the data. Which resulted in a total average of 126.18048333

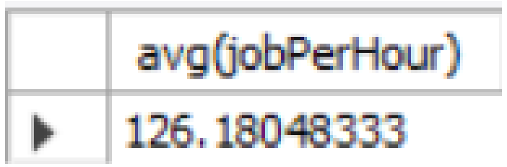




```
select ds,count(job_id)*3600/sum(time_spent) as jobPerHour from job_data group by ds;
select avg(jobPerHour) from(select ds,count(job_id)*3600/sum(time_spent) as jobPerHour from job_data group by ds) as p;
```



ds	jobPerHour
2020-11-30	180.0000
2020-11-29	180.0000
2020-11-28	218.1818
2020-11-27	34.6154
2020-11-26	64.2857
2020-11-25	80.0000

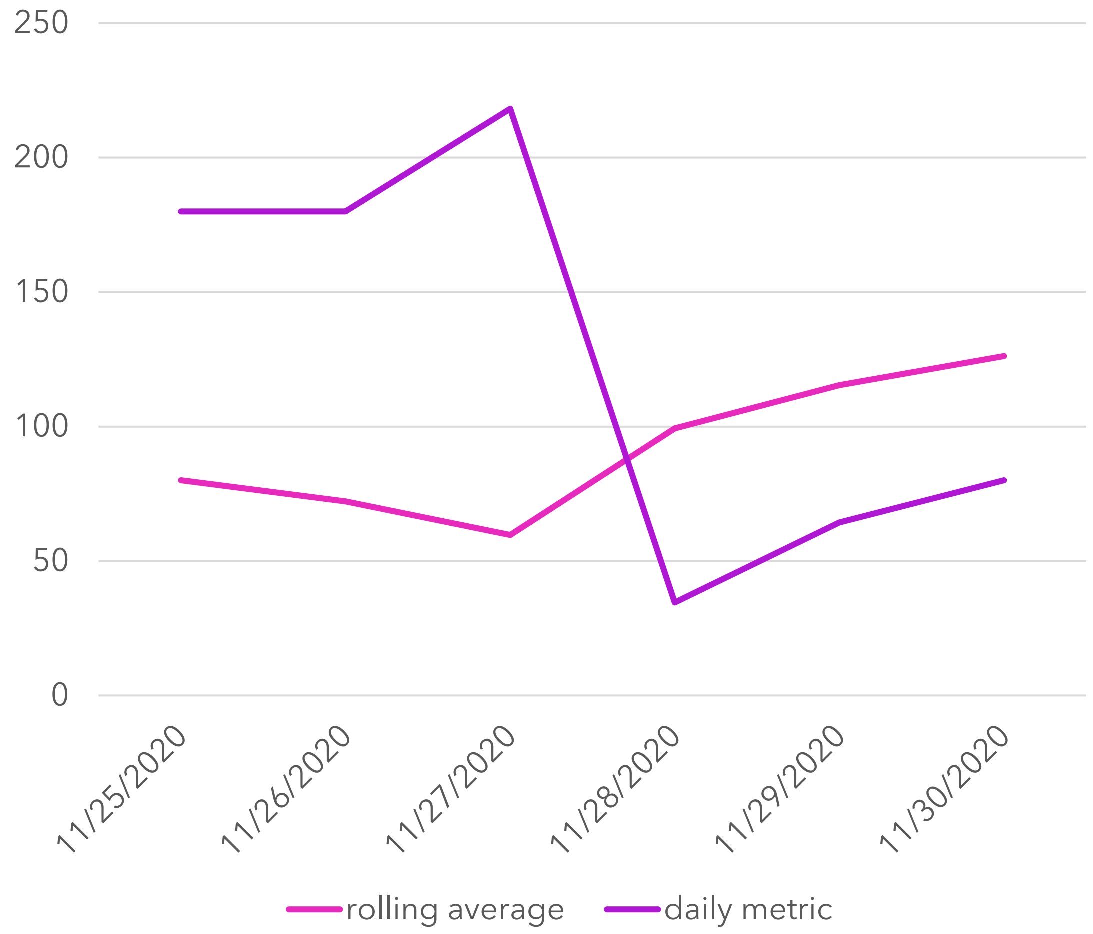


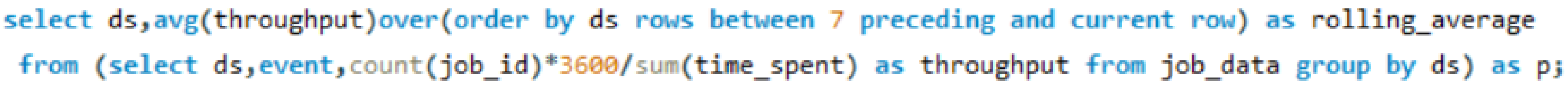
avg(jobPerHour)
126.18048333

Throughput

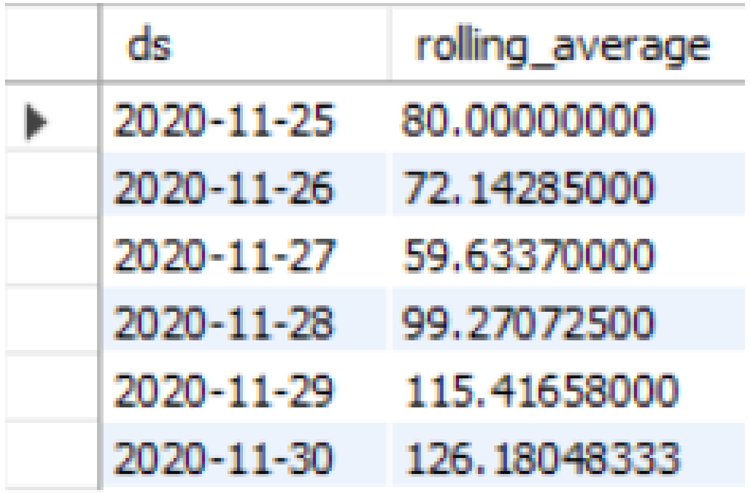
We have to calculate 7 days rolling average of throughput

A rolling average is a more preferable technique as it will give an idea of what is happening over a period of time.





```
select ds,avg(throughput)over(order by ds rows between 7 preceding and current row) as rolling_average
 from (select ds,event,count(job_id)*3600/sum(time_spent) as throughput from job_data group by ds) as p;
```

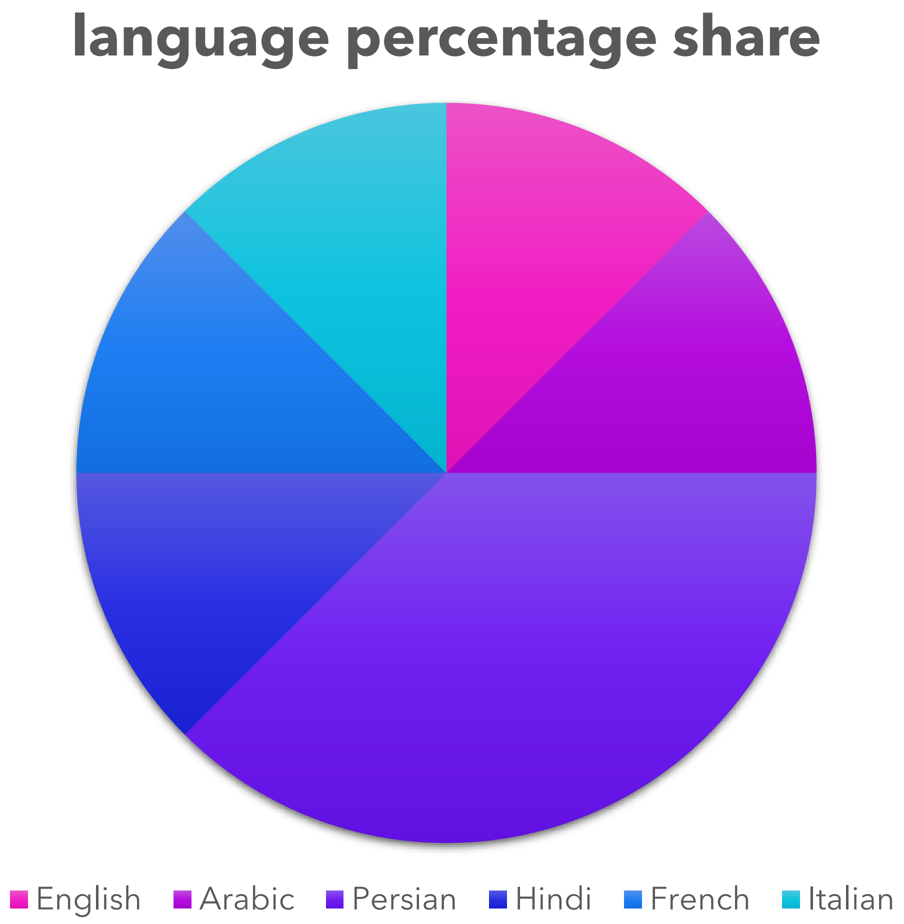


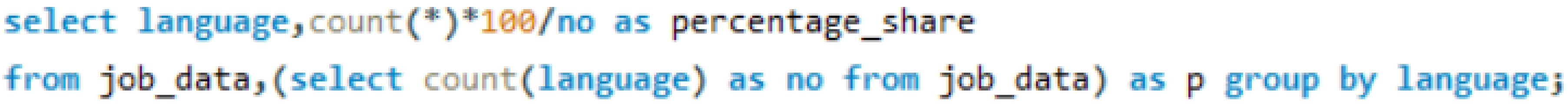
ds	rolling_average
2020-11-25	80.00000000
2020-11-26	72.14285000
2020-11-27	59.63370000
2020-11-28	99.27072500
2020-11-29	115.41658000
2020-11-30	126.18048333

Percentage share of each language

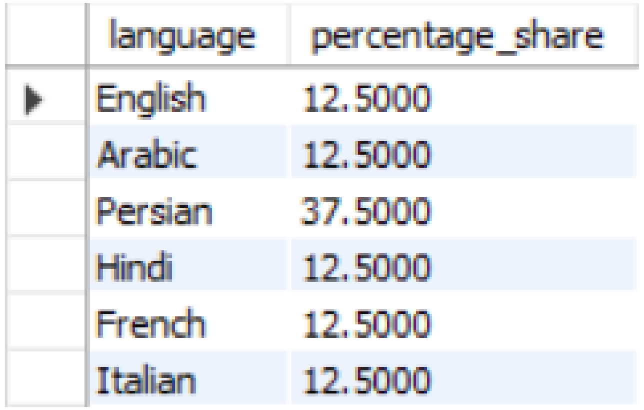
We need to calculate the percentage share of each language.

From the pie chart, we can infer that Persian has the highest share amongst all the languages.





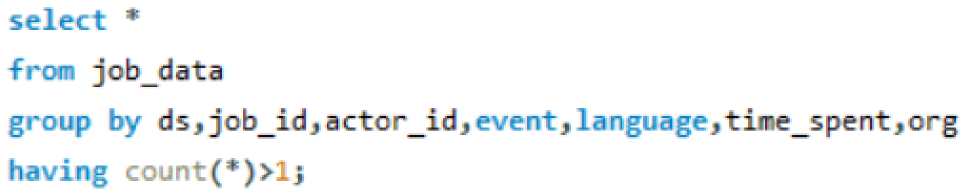
```
select language,count(*)*100/no as percentage_share
from job_data,(select count(language) as no from job_data) as p group by language;
```



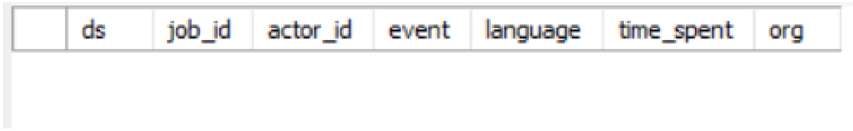
language	percentage_share
English	12.5000
Arabic	12.5000
Persian	37.5000
Hindi	12.5000
French	12.5000
Italian	12.5000

Duplicate rows

We have inferred that there are no two identical rows in the dataset provided



```
select *
from job_data
group by ds,job_id,actor_id,event,language,time_spent,org
having count(*)>1;
```



	ds	job_id	actor_id	event	language	time_spent	org

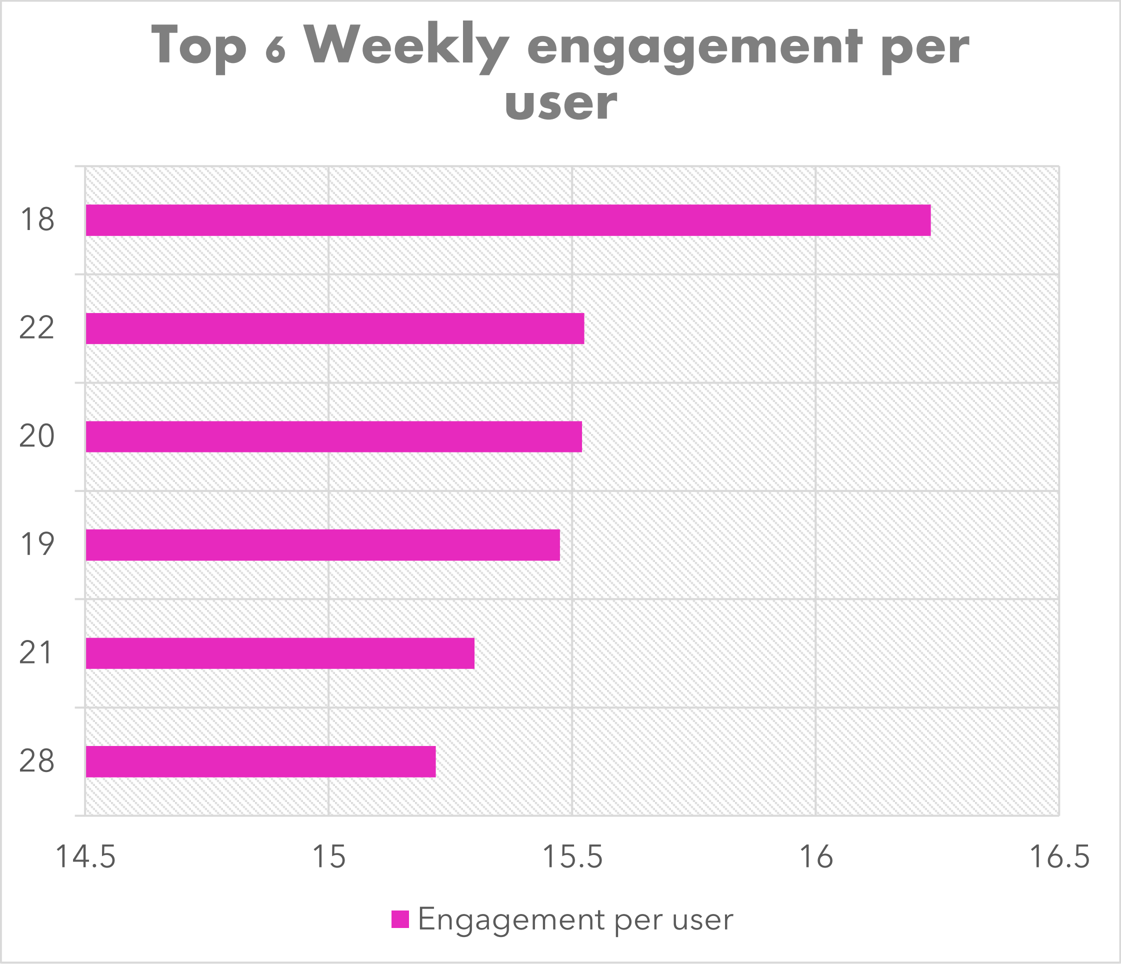
CASE STUDY 2

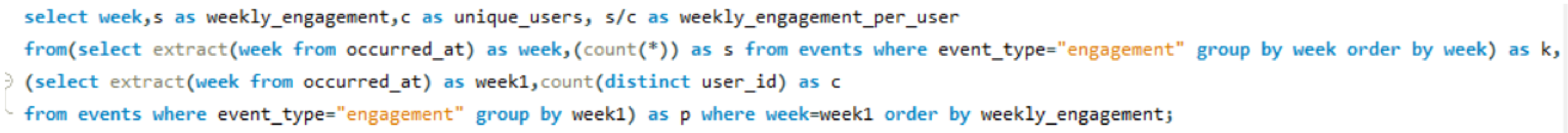
In this case study insights on metric spike will be provided

User Engagement

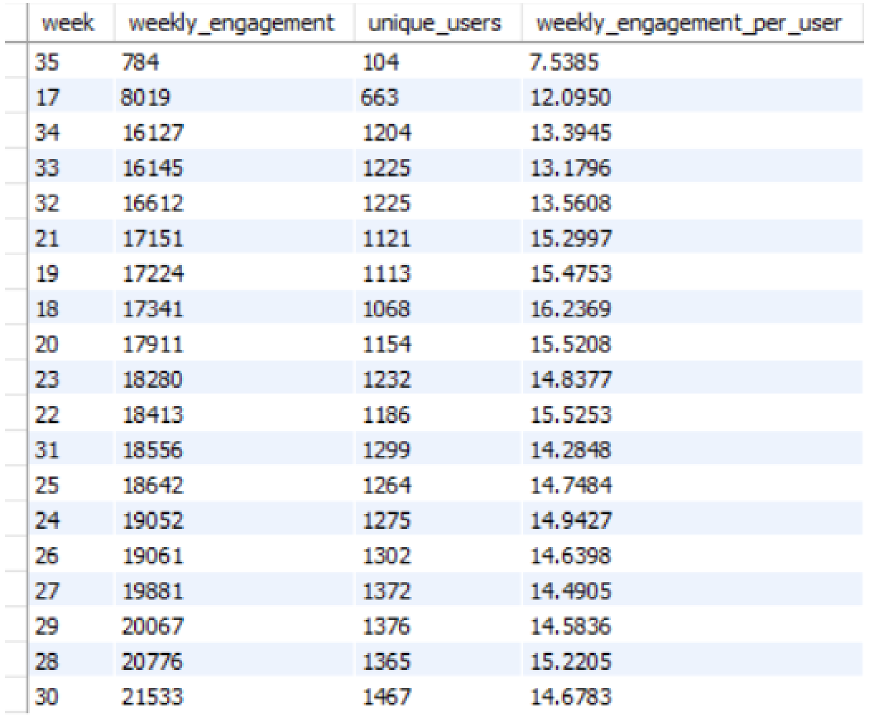
Our task is to calculate the weekly user engagement.

From the stats, we can see that the maximum user engagement was on week 18 of 2014. the average user engagement was 14.22382797





```
select week,s as weekly_engagement,c as unique_users, s/c as weekly_engagement_per_user
from(select extract(week from occurred_at) as week,(count(*)) as s from events where event_type="engagement" group by week order by week) as k,
(select extract(week from occurred_at) as week1,count(distinct user_id) as c
from events where event_type="engagement" group by week1) as p where week=week1 order by weekly_engagement;
```

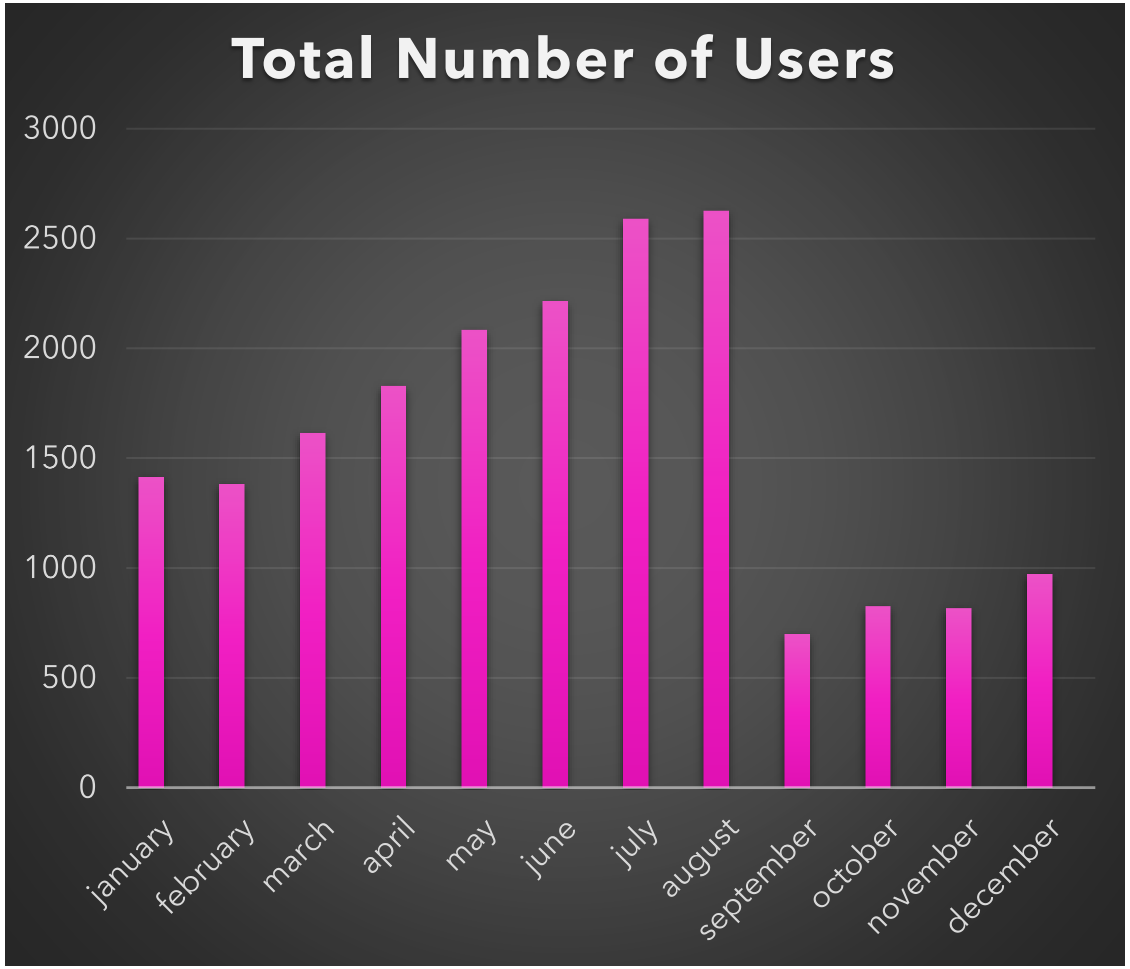


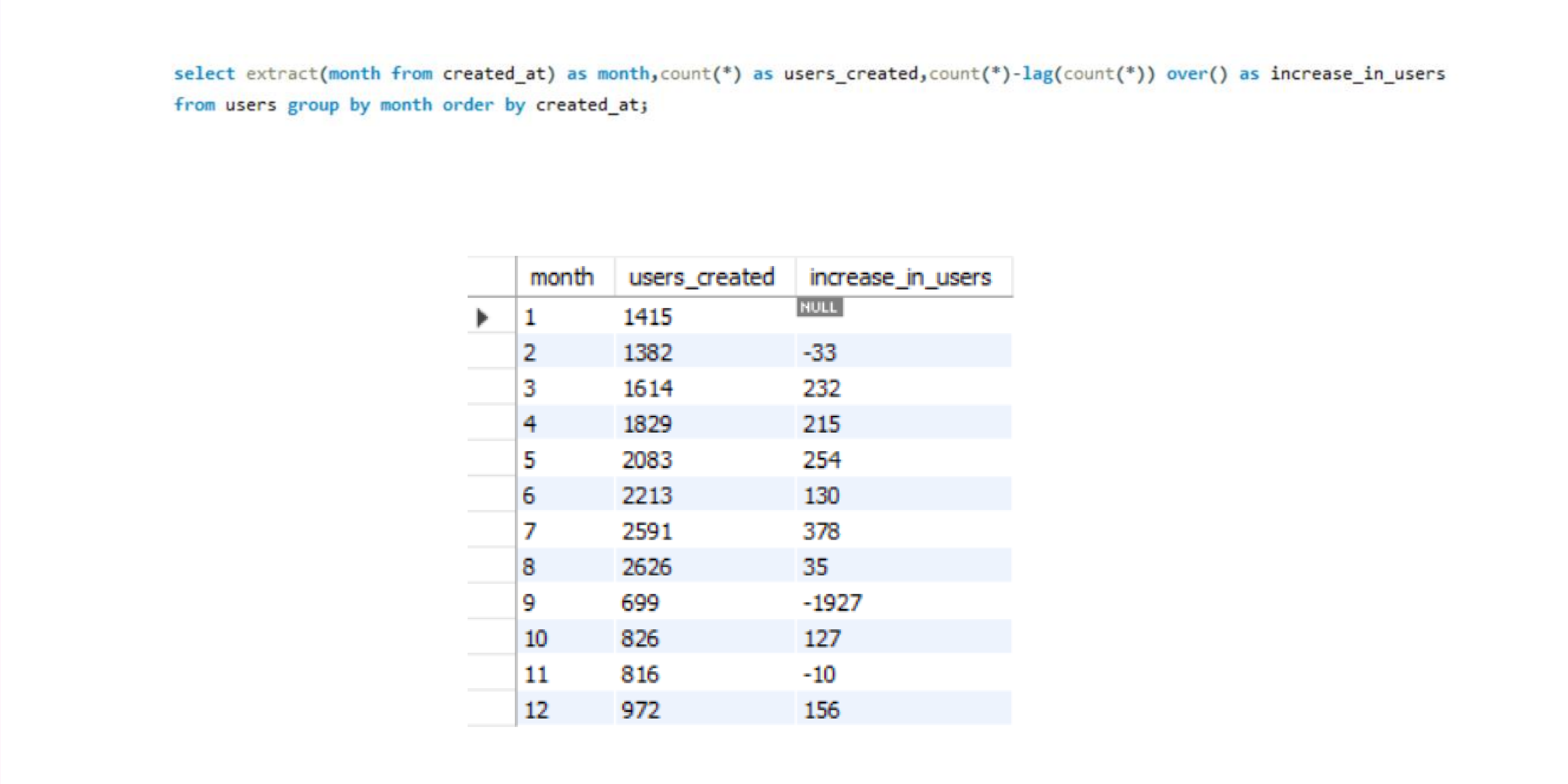
week	weekly_engagement	unique_users	weekly_engagement_per_user
35	784	104	7.5385
17	8019	663	12.0950
34	16127	1204	13.3945
33	16145	1225	13.1796
32	16612	1225	13.5608
21	17151	1121	15.2997
19	17224	1113	15.4753
18	17341	1068	16.2369
20	17911	1154	15.5208
23	18280	1232	14.8377
22	18413	1186	15.5253
31	18556	1299	14.2848
25	18642	1264	14.7484
24	19052	1275	14.9427
26	19061	1302	14.6398
27	19881	1372	14.4905
29	20067	1376	14.5836
28	20776	1365	15.2205
30	21533	1467	14.6783

User Growth

The number of users growing over time is given on the next page

As we can see that the maximum number of users created an account in august 2014





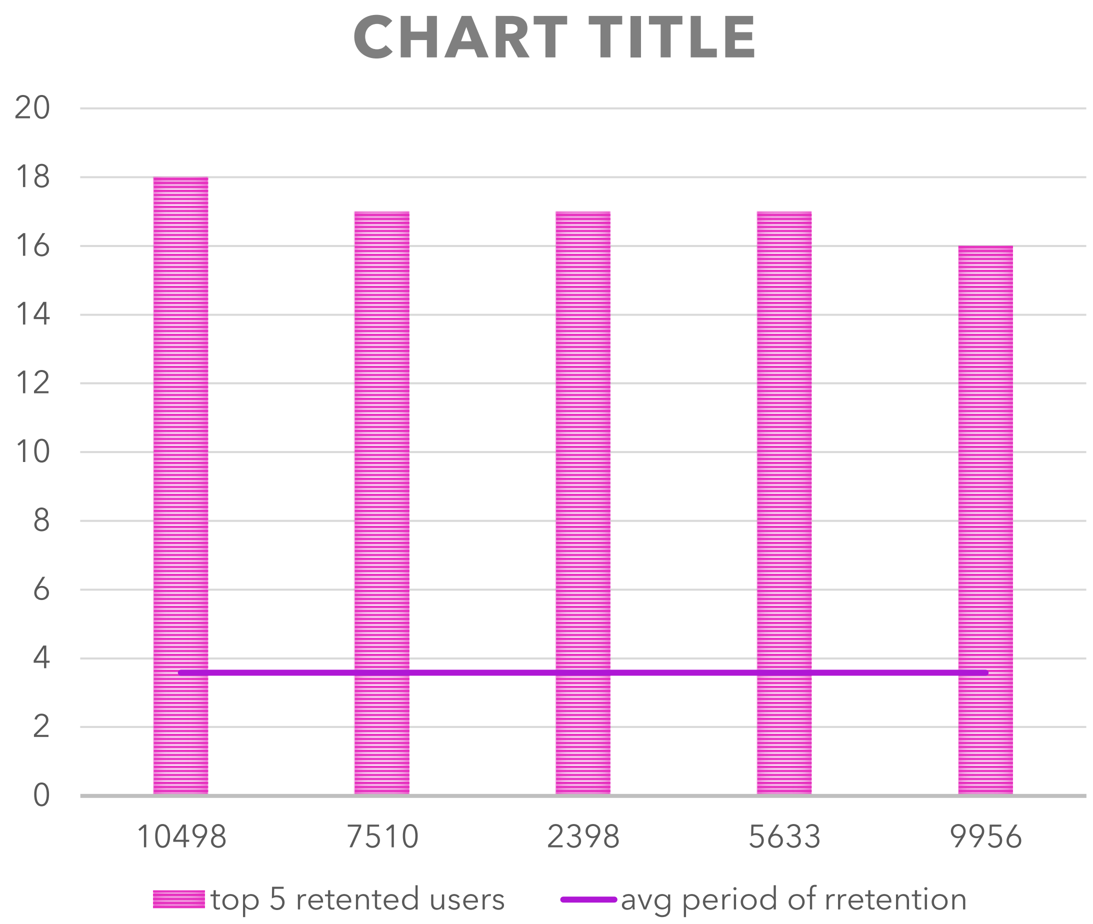
```sql
select extract(month from created_at) as month,count(*) as users_created,count(*)-lag(count(*)) over() as increase_in_users
from users group by month order by created_at;
```

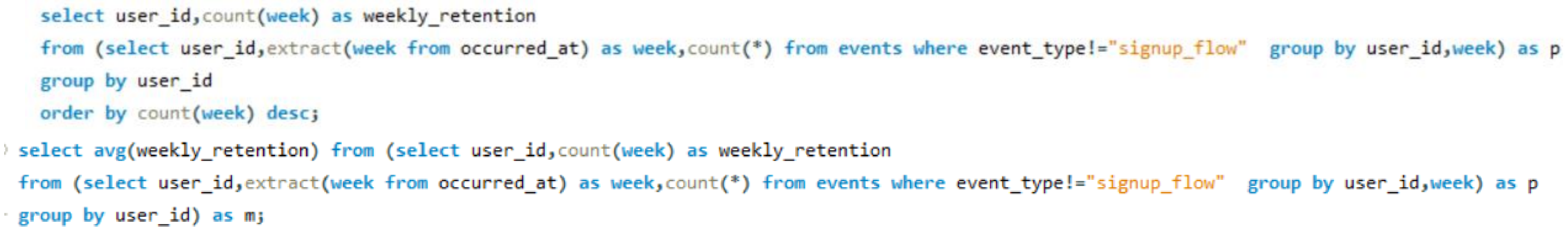
month	users_created	increase_in_users
1	1415	NULL
2	1382	-33
3	1614	232
4	1829	215
5	2083	254
6	2213	130
7	2591	378
8	2626	35
9	699	-1927
10	826	127
11	816	-10
12	972	156

Weekly Retention

A user can be maximum retained for a period of 18 week.

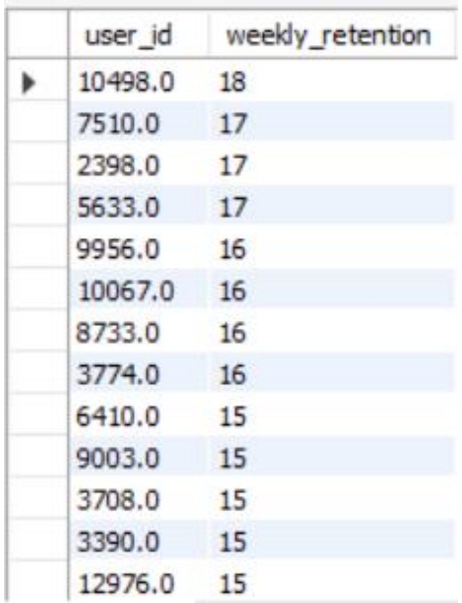
On an average a user can be retained for 3-4 weeks



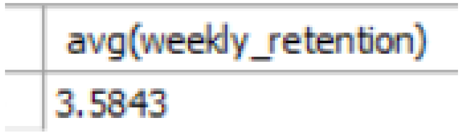


```
select user_id,count(week) as weekly_retention
from (select user_id,extract(week from occurred_at) as week,count(*) from events where event_type!="signup_flow"  group by user_id,week) as p
group by user_id
order by count(week) desc;
```

```
select avg(weekly_retention) from (select user_id,count(week) as weekly_retention
from (select user_id,extract(week from occurred_at) as week,count(*) from events where event_type!="signup_flow"  group by user_id,week) as p
group by user_id) as m;
```



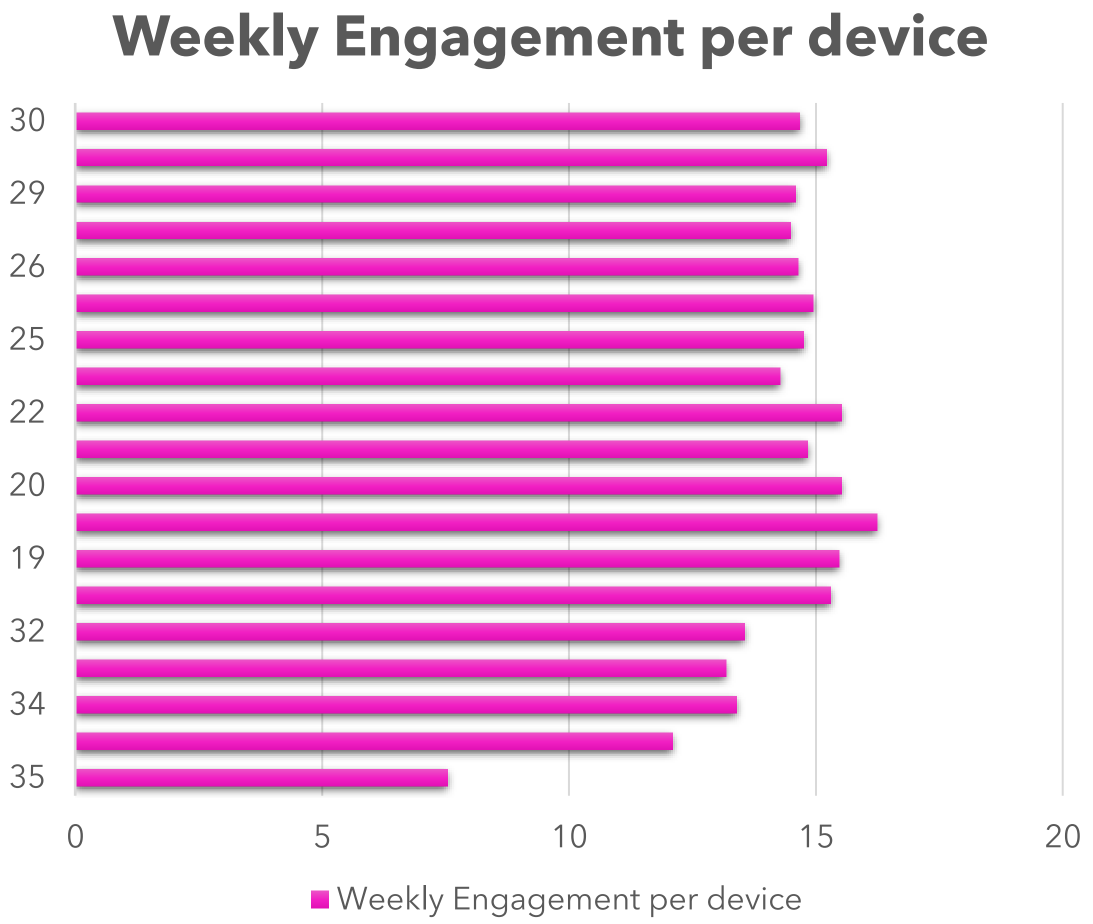
user_id	weekly_retention
10498.0	18
7510.0	17
2398.0	17
5633.0	17
9956.0	16
10067.0	16
8733.0	16
3774.0	16
6410.0	15
9003.0	15
3708.0	15
3390.0	15
12976.0	15

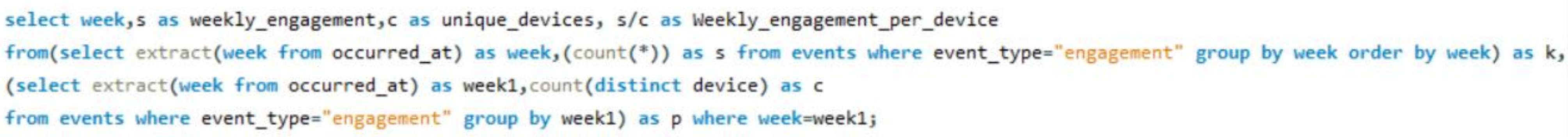


avg(weekly_retention)
3.5843

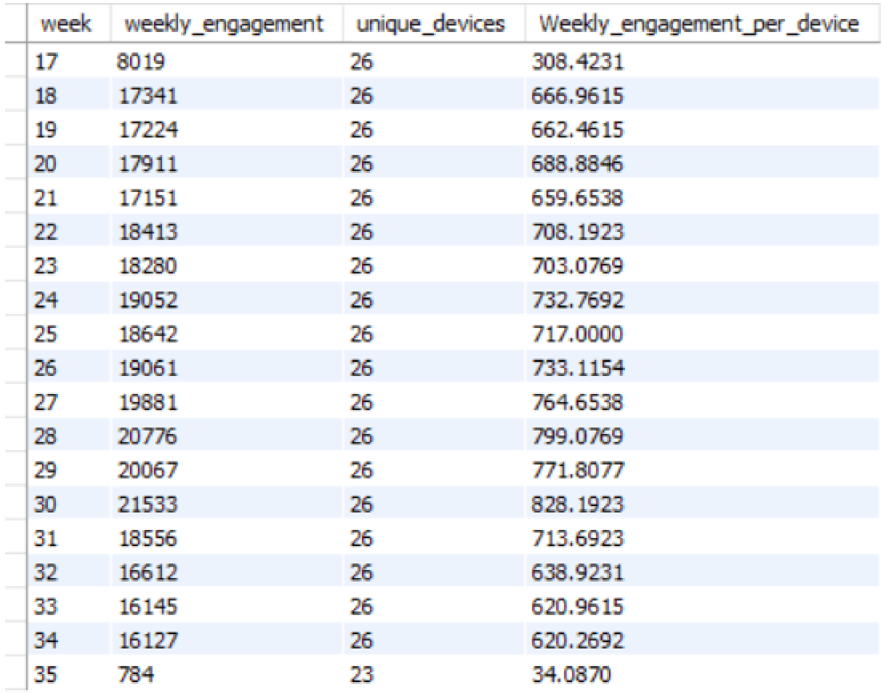
Weekly engagement

Weekly engagement per device is





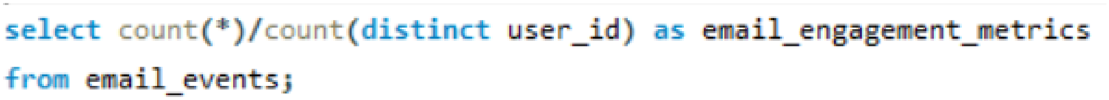
```
select week,s as weekly_engagement,c as unique_devices, s/c as Weekly_engagement_per_device
from(select extract(week from occurred_at) as week,(count(*)) as s from events where event_type="engagement" group by week order by week) as k,
(select extract(week from occurred_at) as week1,count(distinct device) as c
from events where event_type="engagement" group by week1) as p where week=week1;
```



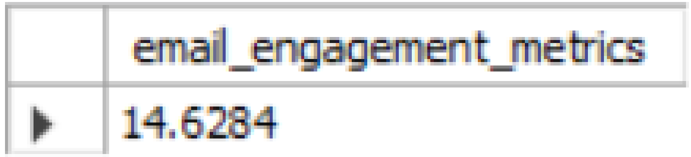
week	weekly_engagement	unique_devices	Weekly_engagement_per_device
17	8019	26	308.4231
18	17341	26	666.9615
19	17224	26	662.4615
20	17911	26	688.8846
21	17151	26	659.6538
22	18413	26	708.1923
23	18280	26	703.0769
24	19052	26	732.7692
25	18642	26	717.0000
26	19061	26	733.1154
27	19881	26	764.6538
28	20776	26	799.0769
29	20067	26	771.8077
30	21533	26	828.1923
31	18556	26	713.6923
32	16612	26	638.9231
33	16145	26	620.9615
34	16127	26	620.2692
35	784	23	34.0870

Email Engagement

The email engagement metrics of the product is 14.6284



```
select count(*)/count(distinct user_id) as email_engagement_metrics
from email_events;
```



	email_engagement_metrics
▶	14.6284

RESULT

- From this project, we have learnt advanced SQL Queries and understood their importance in data analytics.
- We have understood the importance of metric spikes in data analytics.

THANK YOU