OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

BY: PREETY MOHANTA

01

Introduction

The given project consists of 2 case studies:-

1. In first dataset it provides us the detail about the review of the applicants for various job profiles with the help of Operation Analytics where job data is provided and number of jobs reviewed , 7day rolling average of throughput, percentage share of language used and duplicates are found out.
2. The second dataset contains one row per user, with descriptive information about that user's account in this with the use of Investigating Metric Spike where user engagement, user growth, weekly retention, weekly engagement and email engagement is determined.

02

APPROACH

- ❖Get the data from the given source and go thoroughly to the data and understand the data and the tables provided.
- ❖Use MySQL Workbench and import the files in new database by applying various SQL queries
- ❖The insights I will be covering would provide to the company with the right solution to improve its operations and business growth.
- ❖I've used analytic logic to write Queries in MySQL to get the answers to our questions. I have used where clause, different joins, count, average, sum etc.

03

TECH STACK USED

MySQL 8.0 is used to harness the expected results/insights as per requirements given in the project description

Used in the second case study for better visualisation. It was utilised to get more hands on experience.

04

CASE 01

OPERATION ANALYTICS

INSIGHTS TASKS

Jobs Reviewed Over Time

Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.

Throughput Analysis

Objective: Calculate the 7-day rolling average of throughput (number of events per second).

Language Share Analysis

Objective: Calculate the percentage share of each language in the last 30 days.

Duplicate Rows Detection

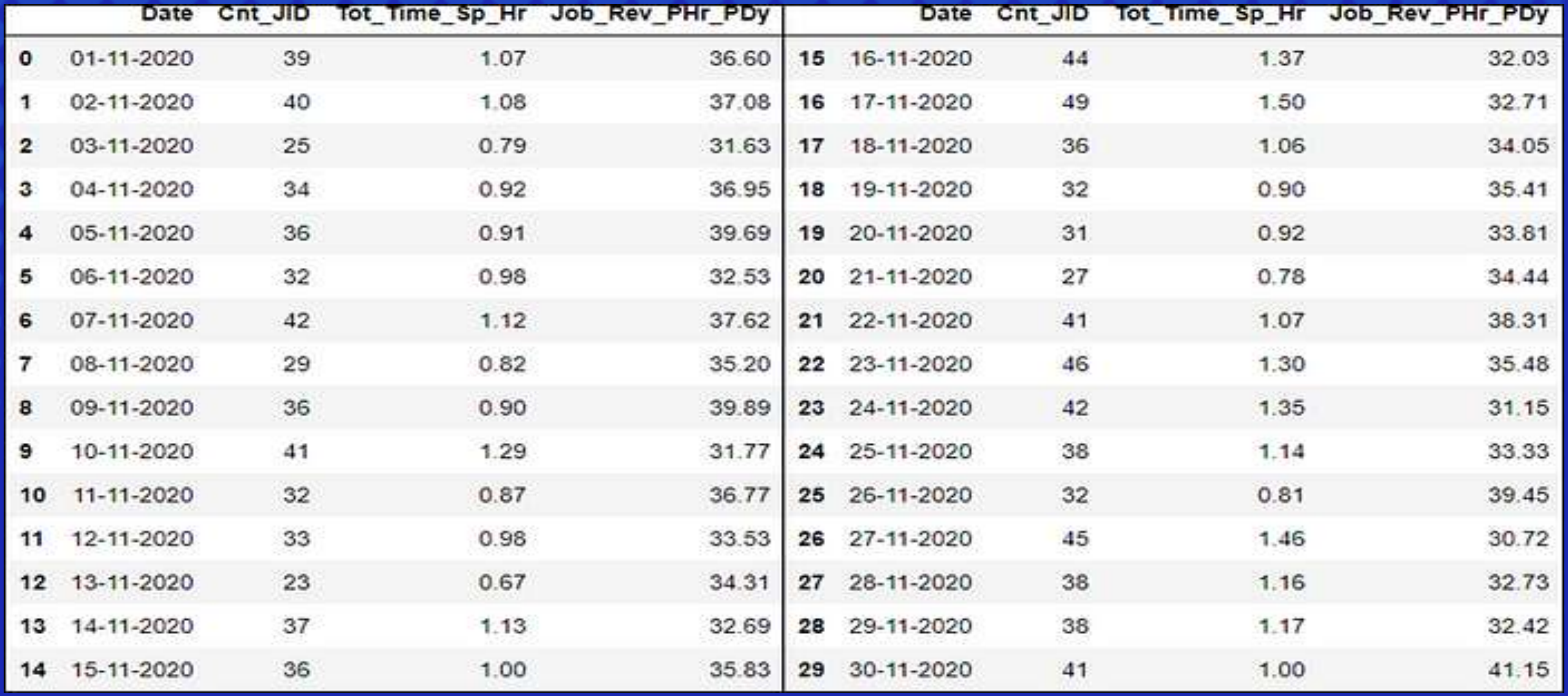
Objective: Identify duplicate rows in the data.

A. NUMBER OF JOBS REVIEWED

QUERY

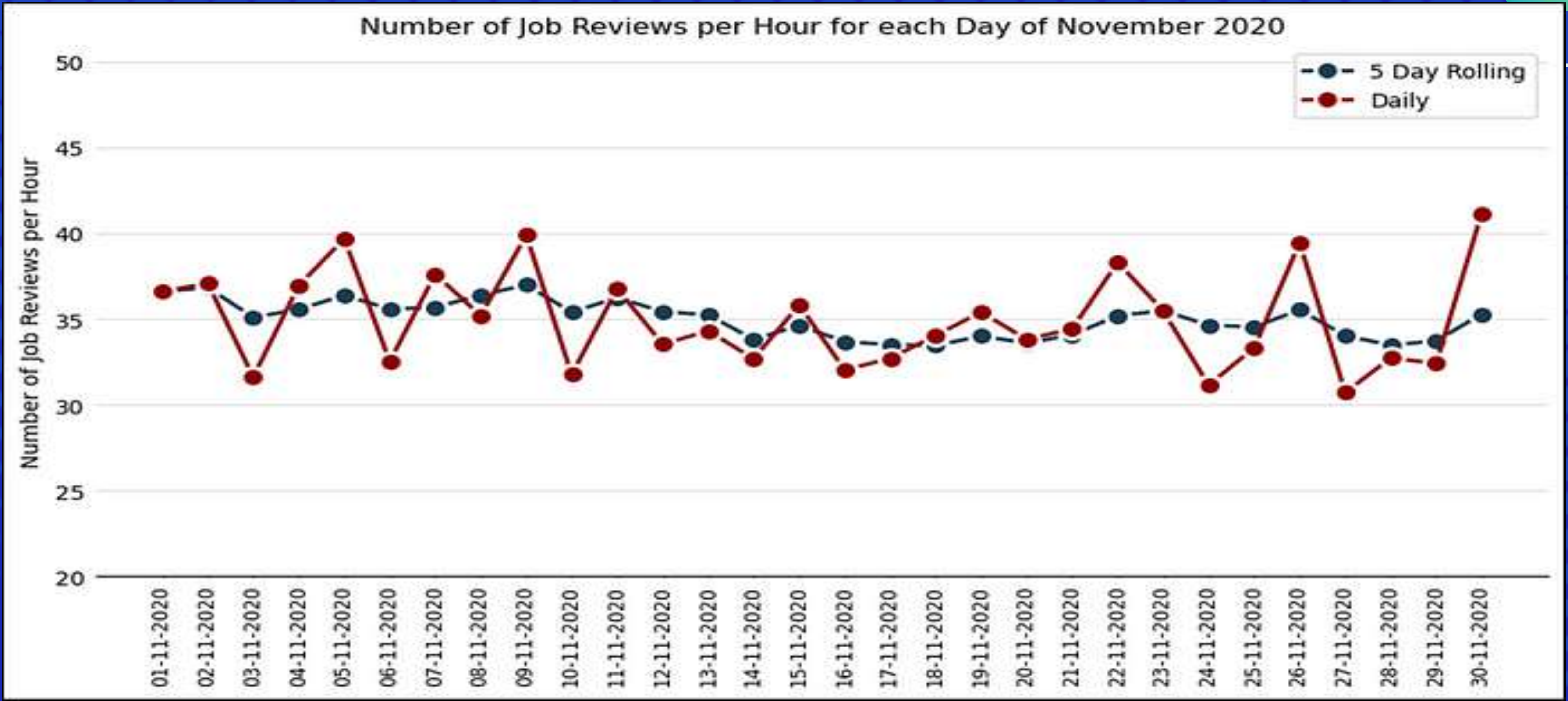
#no. of jobs viewed per hour per day

```
query="""SELECT ds AS Date, COUNT(job_id) AS Cnt_JID, ROUND((SUM(time_spent)/3600),2) AS Tot_Time_Sp_Hr,
ROUND((COUNT(job_id)/(SUM(time_spent)/3600)),2) AS Job_Rev_PHr_PDy
FROM cs_1
WHERE ds BETWEEN
\'01-11-2020\' AND \'30-11-2020\'
GROUP BY ds
ORDER BY ds"""
```



	Date	Cnt_JID	Tot_Time_Sp_Hr	Job_Rev_PHr_PDy
0	01-11-2020	39	1.07	36.60
1	02-11-2020	40	1.08	37.08
2	03-11-2020	25	0.79	31.63
3	04-11-2020	34	0.92	36.95
4	05-11-2020	36	0.91	39.69
5	06-11-2020	32	0.98	32.53
6	07-11-2020	42	1.12	37.62
7	08-11-2020	29	0.82	35.20
8	09-11-2020	36	0.90	39.89
9	10-11-2020	41	1.29	31.77
10	11-11-2020	32	0.87	36.77
11	12-11-2020	33	0.98	33.53
12	13-11-2020	23	0.67	34.31
13	14-11-2020	37	1.13	32.69
14	15-11-2020	36	1.00	35.83
15	16-11-2020	44	1.37	32.03
16	17-11-2020	49	1.50	32.71
17	18-11-2020	36	1.06	34.05
18	19-11-2020	32	0.90	35.41
19	20-11-2020	31	0.92	33.81
20	21-11-2020	27	0.78	34.44
21	22-11-2020	41	1.07	38.31
22	23-11-2020	46	1.30	35.48
23	24-11-2020	42	1.35	31.15
24	25-11-2020	38	1.14	33.33
25	26-11-2020	32	0.81	39.45
26	27-11-2020	45	1.46	30.72
27	28-11-2020	38	1.16	32.73
28	29-11-2020	38	1.17	32.42
29	30-11-2020	41	1.00	41.15

A. NUMBER OF JOBS REVIEWED



From this table, we can see that the number of **job reviews** done in between 30 and 40 for **most** days of November 2020.

B. THROUGHOUT ANALYSIS

QUERY

#calc 7 day rolling avg throughput(no. of events happening pere sec)

Daily Metrix: Code:

```
select
ds as date_of_record,
avg(count(event)) over() as no_events_per_day
from job_data
group by ds
order by ds asc;
```

7 Day Rolling Code:

```
select
a.data_of_record,
avg(a.no_events_per_day) over(rows between 6
preceding and current row) as
Avg_7_day_rolling
from
(select
ds as data_of_record,
count(event) as no_events_per_day
from job_data
group by ds
order by ds asc) as a;
```

B. THROUGHOUT ANALYSIS

OUTPUT:

Insight: I would prefer the 7-Day Rolling Average over daily metric for throughput. The reason being daily metrics can go up or down on a daily basis for factors that cannot be controlled by the organizations like seasonality, major events etc. Continue using the rolling average to observe trends without being influenced by daily fluctuations.

Date_of_record	No_events_per_day	Avg_7_day_rolling
25-11-2020	1.3333	1
26-11-2020	1.3333	1
27-11-2020	1.3333	1
28-11-2020	1.3333	1.25
29-11-2020	1.3333	1.2
30-11-2020	1.3333	1.3333

C. LANGUAGE SHARE ANALYSIS

QUERY

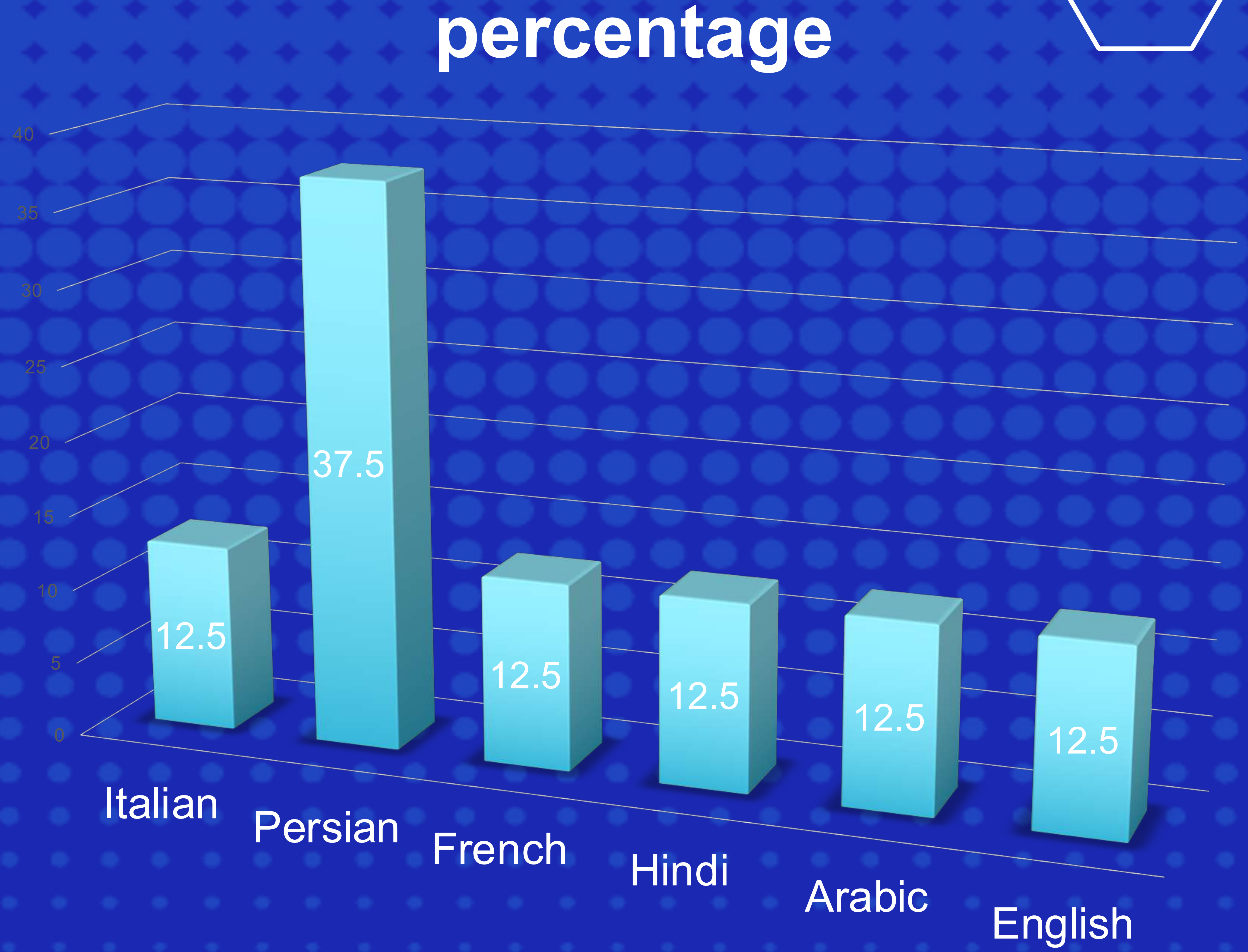
```
#calc percentage share of each language in 30 days

Select job_data.language,
-- count(job_id) as cnt,
-- (select count(job_id) from job_data) as total,
round(( (count(job_id) / ((select count(job_id) from job_data)))*100 ),1)
as Lang_Share
from job_data
group by job_data.language
order by Lang_Share desc;
```

language	percentage
Italian	12.5
Persian	37.5
French	12.5
Hindi	12.5
Arabic	12.5
English	12.5

C. LANGUAGE SHARE ANALYSIS

OUTPUT:

Insight: Language distribution is relatively balanced and it rectifies that Persian Language get the highest percentage share

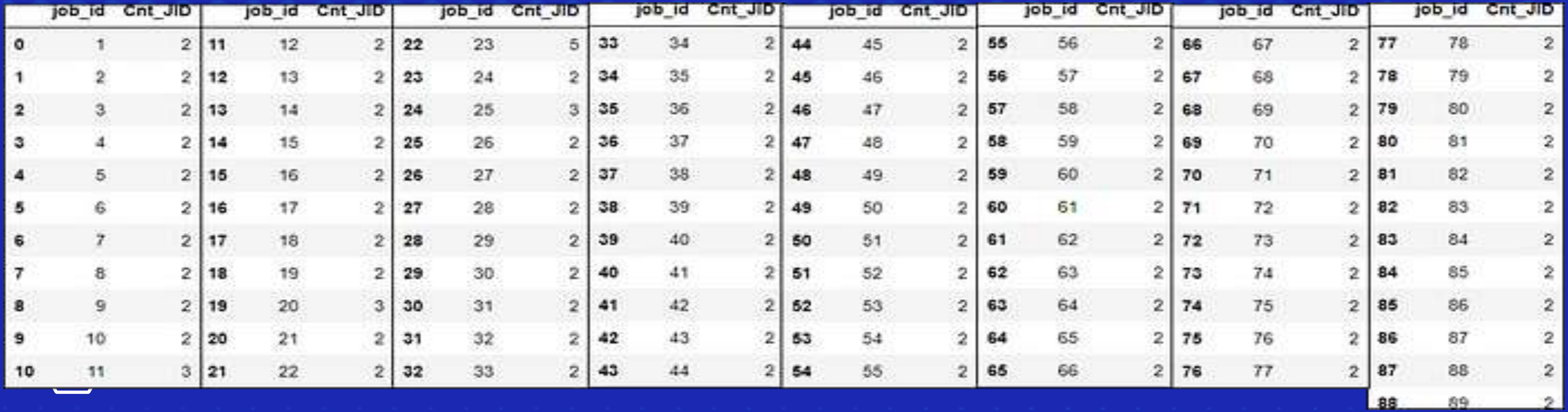


D. DUPLICATE ROWS

QUERY

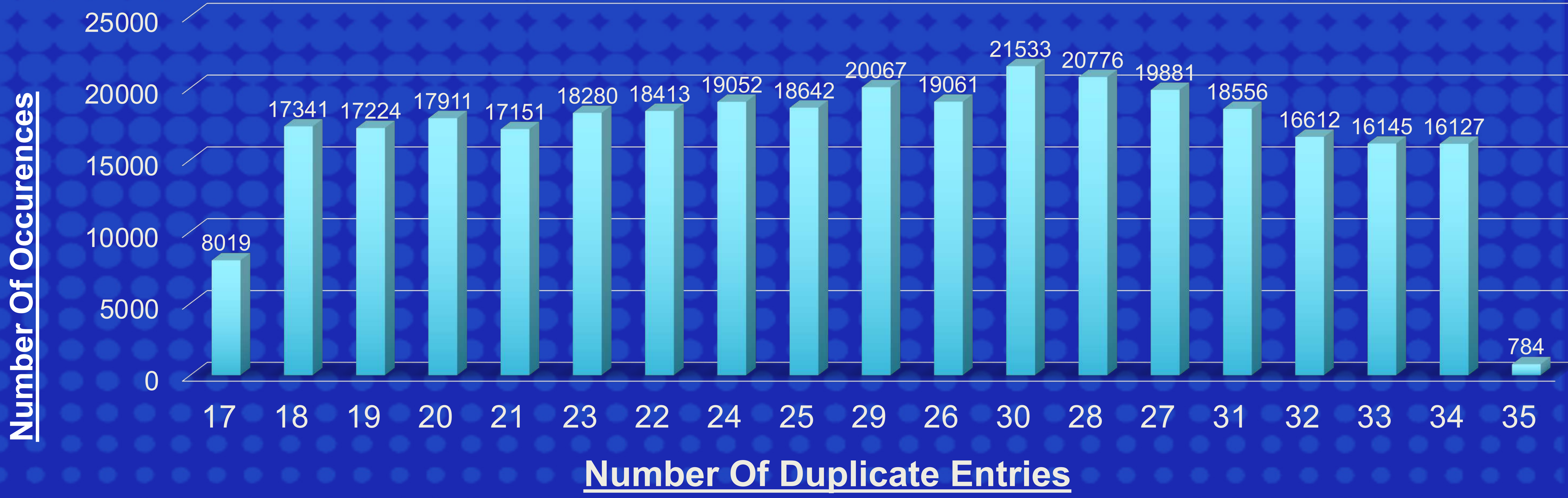
#duplicate rows count

"""SELECT job_id, COUNT(job_id) AS Cnt_JID FROM cs_1
GROUP BY job_id
HAVING Cnt_JID>1
ORDER BY job_id"""



	job_id	Cnt_JID
0	1	2
1	2	2
2	3	2
3	4	2
4	5	2
5	6	2
6	7	2
7	8	2
8	9	2
9	10	2
10	11	3
11	12	2
12	13	2
13	14	2
14	15	2
15	16	2
16	17	2
17	18	2
18	19	2
19	20	3
20	21	2
21	22	2
22	23	5
23	24	2
24	25	3
25	26	2
26	27	2
27	28	2
28	29	2
29	30	2
30	31	2
31	32	2
32	33	2
33	34	2
34	35	2
35	36	2
36	37	2
37	38	2
38	39	2
39	40	2
40	41	2
41	42	2
42	43	2
43	44	2
44	45	2
45	46	2
46	47	2
47	48	2
48	49	2
49	50	2
50	51	2
51	52	2
52	53	2
53	54	2
54	55	2
55	56	2
56	57	2
57	58	2
58	59	2
59	60	2
60	61	2
61	62	2
62	63	2
63	64	2
64	65	2
65	66	2
66	67	2
67	68	2
68	69	2
69	70	2
70	71	2
71	72	2
72	73	2
73	74	2
74	75	2
75	76	2
76	77	2
77	78	2
78	79	2
79	80	2
80	81	2
81	82	2
82	83	2
83	84	2
84	85	2
85	86	2
86	87	2
87	88	2
88	89	2

D. DUPLICATE ROWS



OUTPUT:

Insight: There are 89 number of rows with duplicate values of job_id.

05

CASE 02

INVESTIGATING METRIC SPIKE

INSIGHTS TASKS

Weekly User Engagement

Objective: Measure the activeness of users on a weekly basis.

User Growth Analysis

Objective: Analyze the growth of users over time for a product.

Weekly Retention Analysis

Objective: Analyze the retention of users on a weekly basis after signing up for a product.

Weekly Engagement Per Device

Objective: Measure the activeness of users on a weekly basis per device.

Email Engagement Analysis

Objective: Analyze how users are engaging with the email service

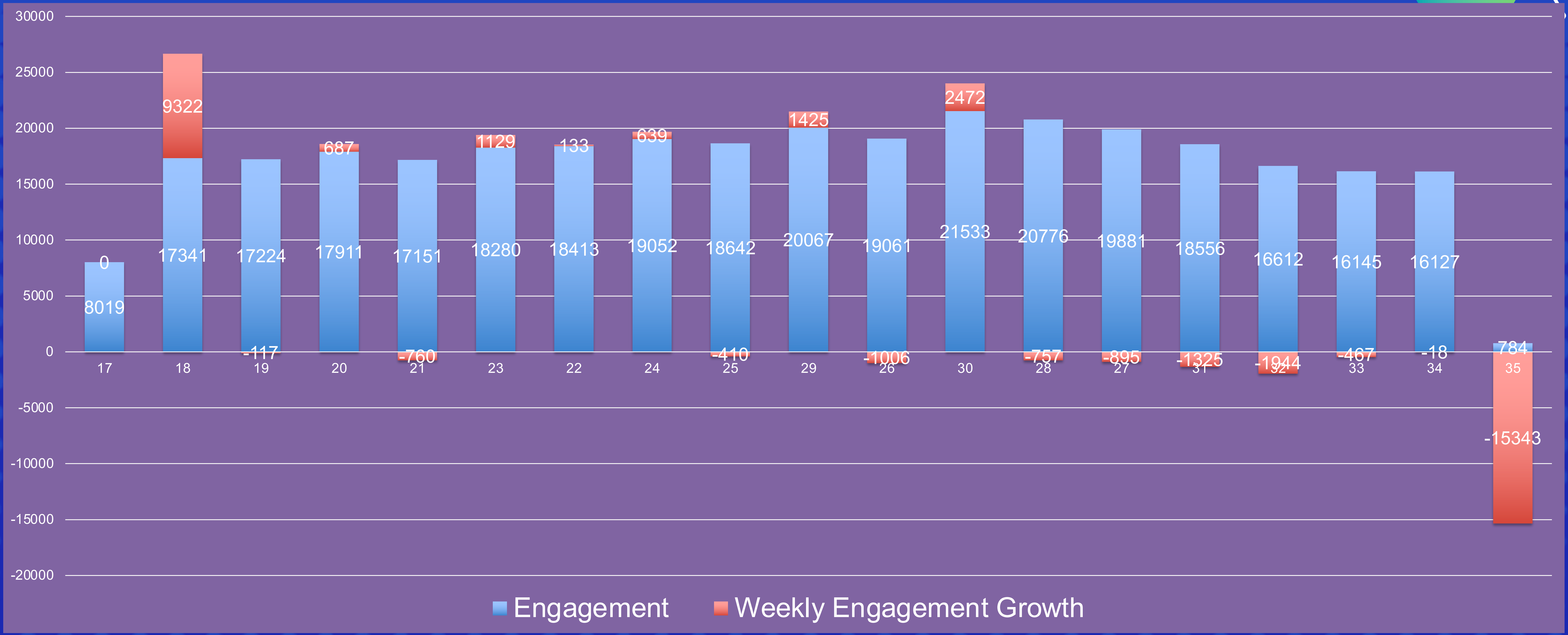
A. WEEKLY USER ENGAGEMENT

QUERY

```
#1. calc the weekly user engagement

select * from events_table;
select extract(week from occurred_at) as weeks,
count(distinct user_id) as no_of_users from events_table
where event_type="engagement"
group by weeks order by weeks;
```

Week Of The Year	Engagement	Weekly Engagement Growth
17	8019	Null
18	17341	9322
19	17224	-117
20	17911	687
21	17151	-760
23	18280	1129
22	18413	133
24	19052	639
25	18642	-410
29	20067	1425
26	19061	-1006
30	21533	2472
28	20776	-757
27	19881	-895
31	18556	-1325
32	16612	-1944
33	16145	-467
34	16127	-18
35	784	-15343

A. WEEKLY USER ENGAGEMENT



OUTPUT:

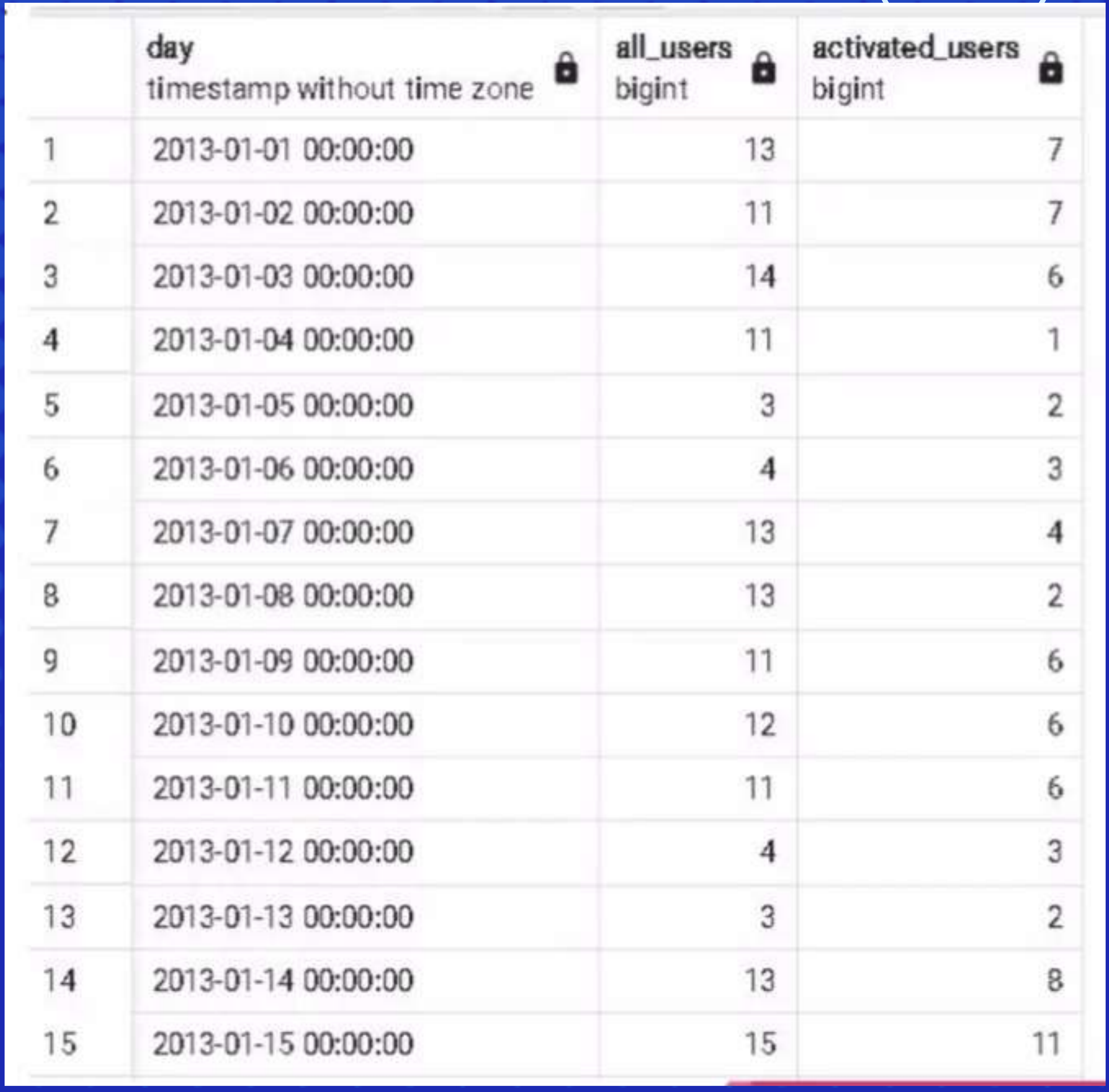
Insight: Language distribution is relatively balanced and it rectifies that Persian Language get the highest percentage share

B. USER GROWTH ANALYSIS

QUERY

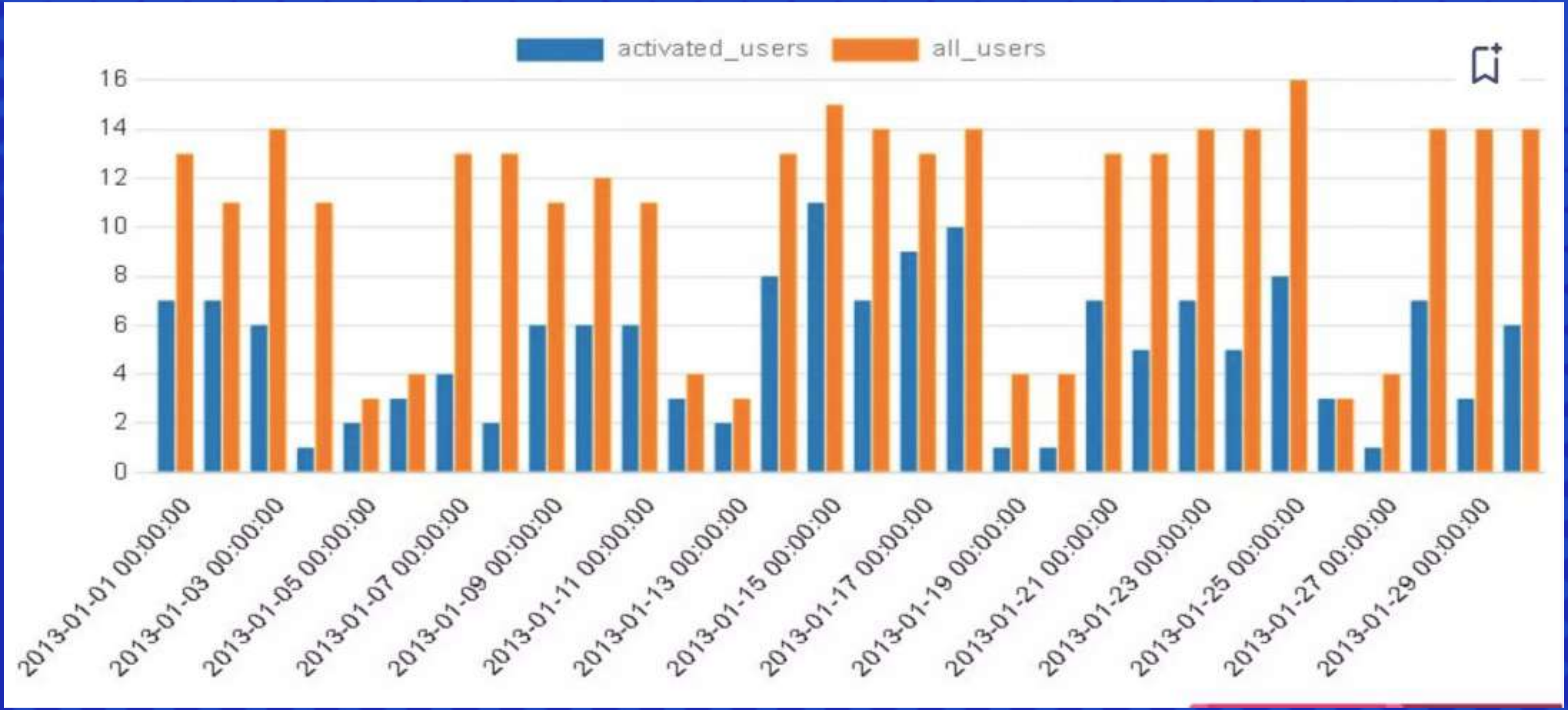
#2. calc the user growth over time for product

```
select week_num, year_num,
sum(active_users) over (order by week_num, year_num
rows between unbounded preceding and current row) as
cumulative_sum
from (
select extract(week from activated_at) as week_num,
extract(year from activated_at) as year_num,
count(distinct user_id) as active_users from users_table
where state= "active"
group by year_num, week_num
order by year_num, week_num) as alias;
```



	day timestamp without time zone	all_users bigint	activated_users bigint
1	2013-01-01 00:00:00	13	7
2	2013-01-02 00:00:00	11	7
3	2013-01-03 00:00:00	14	6
4	2013-01-04 00:00:00	11	1
5	2013-01-05 00:00:00	3	2
6	2013-01-06 00:00:00	4	3
7	2013-01-07 00:00:00	13	4
8	2013-01-08 00:00:00	13	2
9	2013-01-09 00:00:00	11	6
10	2013-01-10 00:00:00	12	6
11	2013-01-11 00:00:00	11	6
12	2013-01-12 00:00:00	4	3
13	2013-01-13 00:00:00	3	2
14	2013-01-14 00:00:00	13	8
15	2013-01-15 00:00:00	15	11

B. USER GROWTH ANALYSIS



OUTPUT:

Insight: User growth has been positive over time, with some fluctuations.

From 1st day to Last day in dataset of users there is 9381 users grow

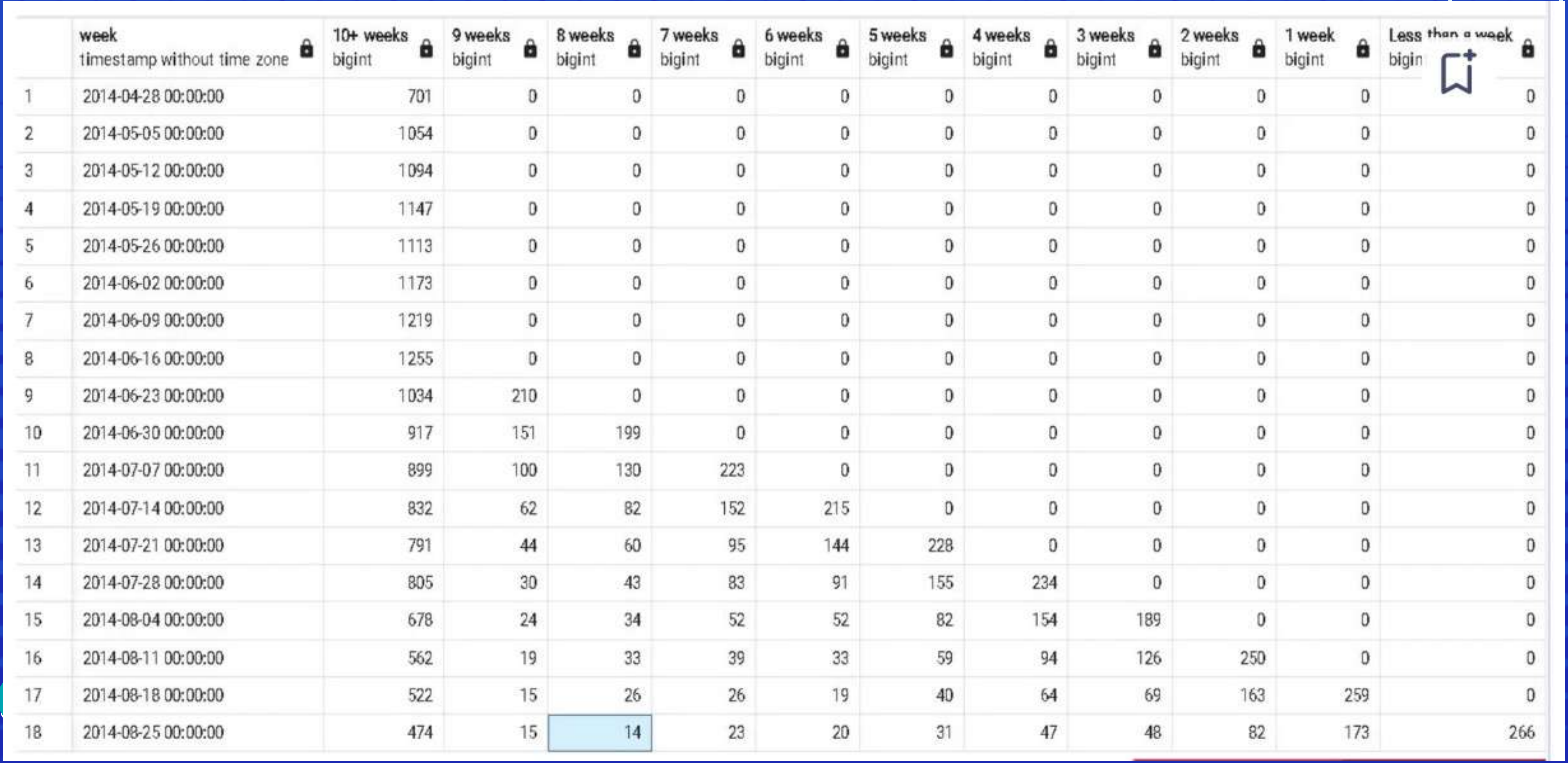
c. WEEKLY RETENTION ANALYSIS

QUERY

#3. calc the weekly retention of users-signup cohort

```
SELECT a.signup_week, b.engagement_week, count(a.user_id) as weekly_retention
FROM ( (SELECT distinct user_id , extract(week from occurred_at ) as signup_week
from trainity3.events WHERE event_type = 'signup_flow' and event_name =
'complete_signup' ) a LEFT JOIN (SELECT distinct user_id , extract(week from
occurred_at ) as engagement_week FROM trainity3.events where event_type =
'engagement' ) b on a.user_id = b.user_id ) Group by signup_week;
```

c. WEEKLY RETENTION ANALYSIS

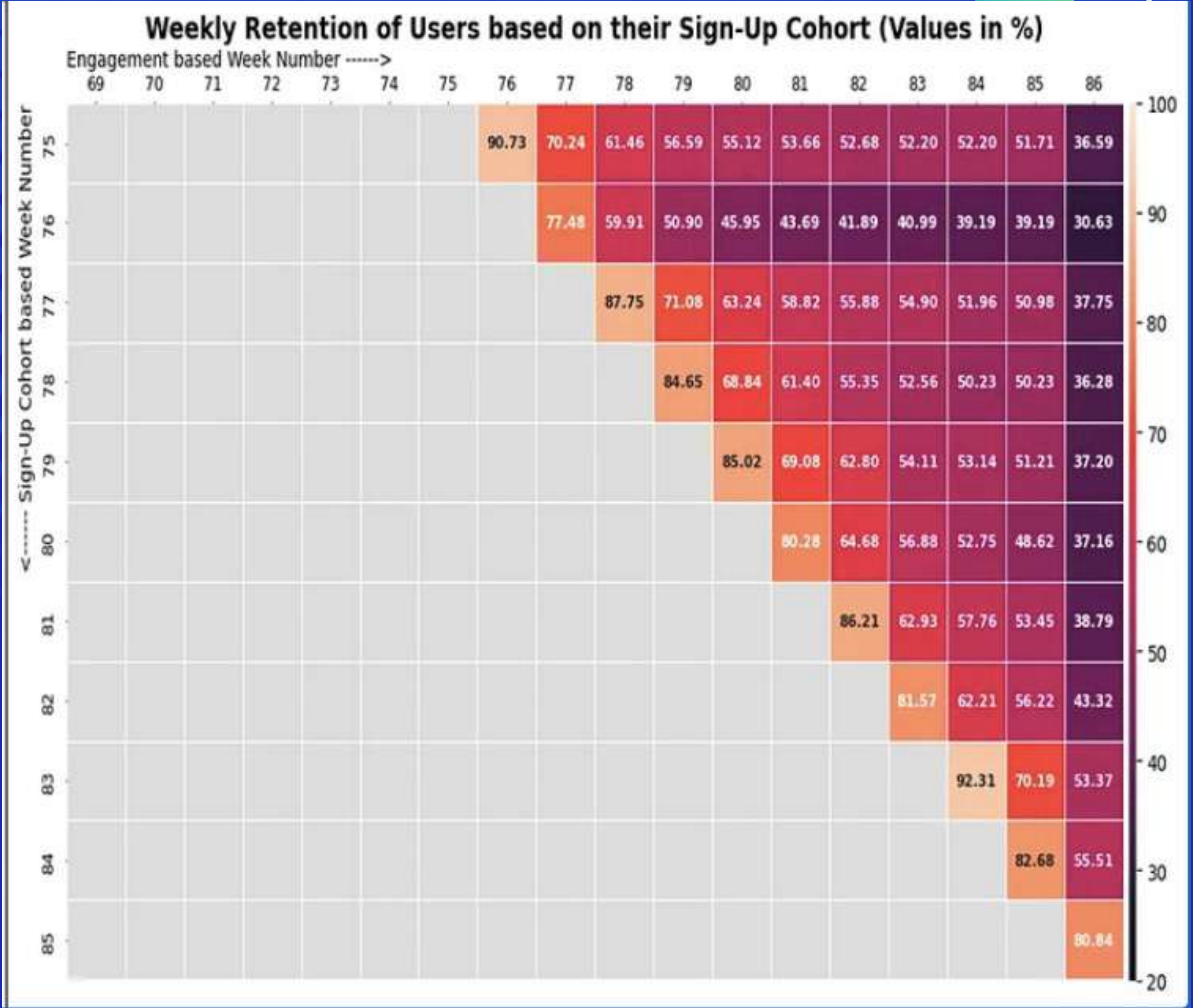


	week timestamp without time zone	10+ weeks bigint	9 weeks bigint	8 weeks bigint	7 weeks bigint	6 weeks bigint	5 weeks bigint	4 weeks bigint	3 weeks bigint	2 weeks bigint	1 week bigint	Less than a week bigin
1	2014-04-28 00:00:00	701	0	0	0	0	0	0	0	0	0	0
2	2014-05-05 00:00:00	1054	0	0	0	0	0	0	0	0	0	0
3	2014-05-12 00:00:00	1094	0	0	0	0	0	0	0	0	0	0
4	2014-05-19 00:00:00	1147	0	0	0	0	0	0	0	0	0	0
5	2014-05-26 00:00:00	1113	0	0	0	0	0	0	0	0	0	0
6	2014-06-02 00:00:00	1173	0	0	0	0	0	0	0	0	0	0
7	2014-06-09 00:00:00	1219	0	0	0	0	0	0	0	0	0	0
8	2014-06-16 00:00:00	1255	0	0	0	0	0	0	0	0	0	0
9	2014-06-23 00:00:00	1034	210	0	0	0	0	0	0	0	0	0
10	2014-06-30 00:00:00	917	151	199	0	0	0	0	0	0	0	0
11	2014-07-07 00:00:00	899	100	130	223	0	0	0	0	0	0	0
12	2014-07-14 00:00:00	832	62	82	152	215	0	0	0	0	0	0
13	2014-07-21 00:00:00	791	44	60	95	144	228	0	0	0	0	0
14	2014-07-28 00:00:00	805	30	43	83	91	155	234	0	0	0	0
15	2014-08-04 00:00:00	678	24	34	52	52	82	154	189	0	0	0
16	2014-08-11 00:00:00	562	19	33	39	33	59	94	126	250	0	0
17	2014-08-18 00:00:00	522	15	26	26	19	40	64	69	163	259	0
18	2014-08-25 00:00:00	474	15	14	23	20	31	47	48	82	173	266

C. WEEKLY RETENTION ANALYSIS

OUTPUT:

Insight: We can observe that the retention rate is remaining same for most of the middle event weeks. On further investigation, we found that event "sent_weekly_digest" forms majority of the events for most of the event weeks from 69 to 85 and the number of occurrence of this event is remaining constant for most of the middle overs for a given sign-up cohort week.



	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86
75								90.73	70.24	61.46	56.59	55.12	53.66	52.68	52.20	52.20	51.71	36.59
76									77.48	59.91	50.90	45.95	43.69	41.89	40.99	39.19	39.19	30.63
77										87.75	71.08	63.24	58.82	55.88	54.90	51.96	50.98	37.75
78											84.65	68.84	61.40	55.35	52.56	50.23	50.23	36.28
79												85.02	69.08	62.80	54.11	53.14	51.21	37.20
80													80.28	64.68	56.88	52.75	48.62	37.16
81														86.21	62.93	57.76	53.45	38.79
82															81.57	62.21	56.22	43.32
83																92.31	70.19	53.37
84																	82.68	55.51
85																		80.84

D. Weekly Engagement Per Device

QUERY

```
#4. calc the weekly user engagement per device

query1='''SELECT device, TIMESTAMPDIFF(WEEK, \'2013-01-01 04:40:10\',
DATE_FORMAT(STR_TO_DATE(occurred_at, \'%d-%m-%Y %H:%i\'), \'%Y-%m-%d
%H:%i:%S\')) AS wk, COUNT(user_id) as Cnt
FROM cs_2_t_2 WHERE event_type = \'engagement\‘ GROUP BY device, wk
ORDER BY device'''

query2='''SELECT device AS Device, ROUND(AVG(q1.Cnt), 2) AS
Avg_Week_Eng_P_Dev
FROM ({}) AS q1 GROUP BY device ORDER BY Avg_Week_Eng_P_Dev
DESC'''.format(query1)
```

D. Weekly Engagement Per Device

OUTPUT:

Insight: Given is average weekly engagement per device

The weekly data per device was very large (960 rows) hence calculated the weekly data

MacBook pro is used the most
Samsung galaxy table is used least
All three devices are laptops. It is understandable as these are formal events mostly used in corporate environment

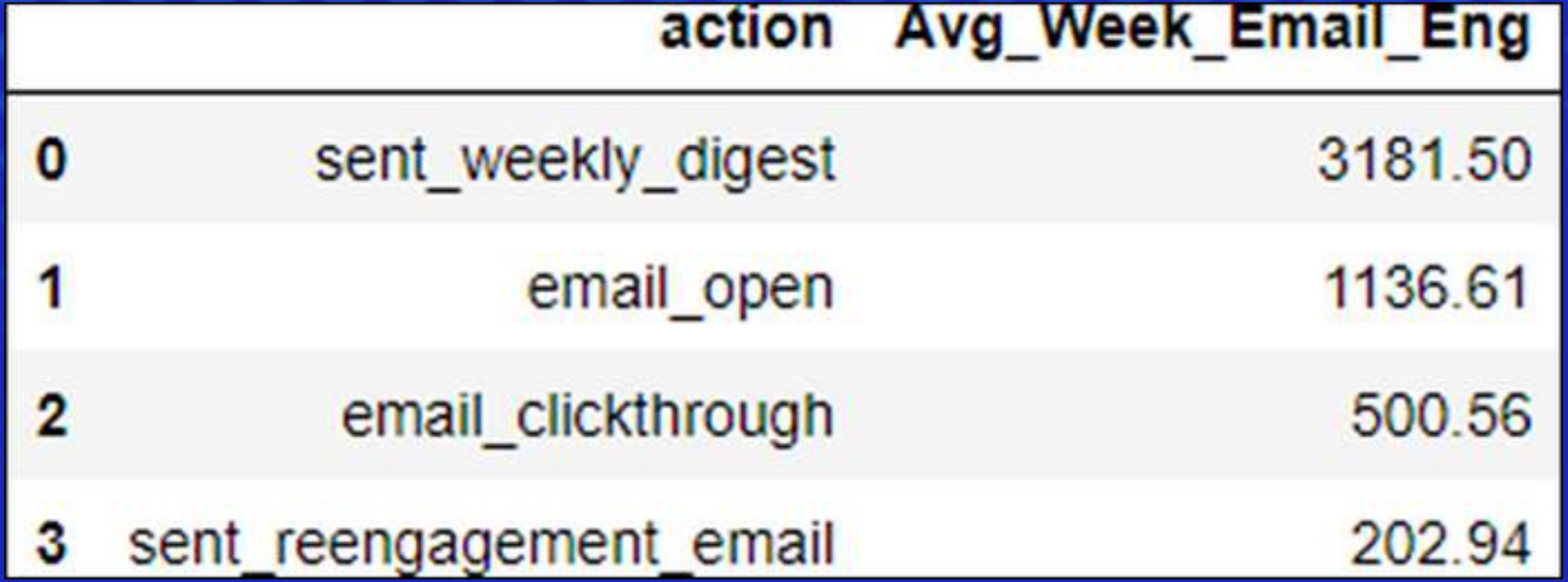
Device_name	Avg_weekly_users	Avg_times_used_weekly
Acer Aspire Desktop	26.00	32.95
Acer Aspire Notebook	43.16	56.84
Amazon Fire Phone	10.56	13.78
Asus Chromebook	43.53	58.89
Dell Inspiron Desktop	46.63	62.74
Dell Inspiron Notebook	91.11	123.47
Hp Pavilion Desktop	42.11	55.84
Htc One	21.84	27.68
Ipad Air	51.44	61.72
Ipad Mini	30.00	34.74
Iphone 4s	46.63	60.58
Iphone 5	123.16	161.21
Iphone 5s	73.32	96.79
Kindle Fire	21.16	25.53
Lenovo Thinkpad	172.95	232.58
Mac Mini	20.47	27.37
Macbook Air	123.16	164.89
Macbook Pro	260.16	358.16
Nexus 10	27.05	31.84
Nexus 5	76.37	99.63
Nexus 7	36.37	43.26
Nokia Lumia 635	28.16	36.26
Samsumg Galaxy Tablet	10.28	12.11
Samsung Galaxy Note	13.47	17.58
Samsung Galaxy S4	91.58	118.74
Windows Surface	18.21	21.53

E. EMAIL ENGAGEMENT ANALYSIS

QUERY

#5. calc the users email engagement metrics

```
query1='''SELECT action, TIMESTAMPDIFF(WEEK, \'2013-01-01 04:40:10\', occurred_at) AS wk,
COUNT(user_id) as Cnt
FROM cs_2_t_3 GROUP BY action, wk ORDER BY action'''

query2='''SELECT action, ROUND(AVG(q1.Cnt), 2) AS Avg_Week_Email_Eng
FROM ({}) AS q1 GROUP BY action ORDER BY Avg_Week_Email_Eng DESC'''.format(query1)
```



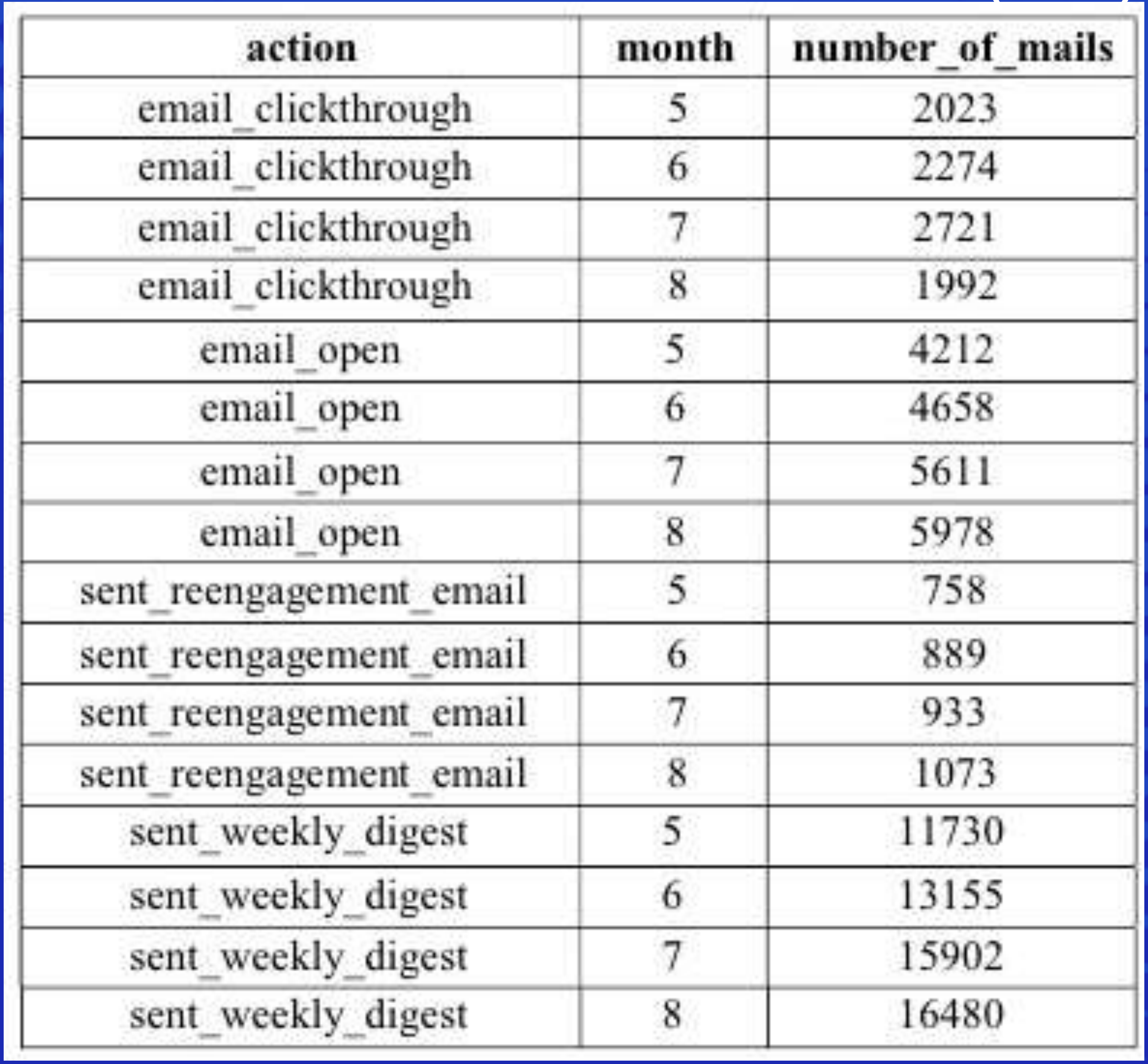
	action	Avg_Week_Email_Eng
0	sent_weekly_digest	3181.50
1	email_open	1136.61
2	email_clickthrough	500.56
3	sent_reengagement_email	202.94

E. EMAIL ENGAGEMENT ANALYSIS

OUTPUT:

Insight: Email engagement metrics include an open rate of approximately 33.58% and a click rate of about 14.79%

From the above table, we can observe that most email activity is related to sent_weekly_digest



action	month	number_of_mails
email_clickthrough	5	2023
email_clickthrough	6	2274
email_clickthrough	7	2721
email_clickthrough	8	1992
email_open	5	4212
email_open	6	4658
email_open	7	5611
email_open	8	5978
sent_reengagement_email	5	758
sent_reengagement_email	6	889
sent_reengagement_email	7	933
sent_reengagement_email	8	1073
sent_weekly_digest	5	11730
sent_weekly_digest	6	13155
sent_weekly_digest	7	15902
sent_weekly_digest	8	16480

05

Conclusion

This project has been highly beneficial as it allowed me to apply my SQL skills and gain hands-on experience in data analysis.
In this project of Operation Analytics and Investigating Metric Spike, I have achieved various Analytics and logical skills as well as technical skills to efficiently use MySQL. I learn how to understand dataset. What kind of questions we have to ask to get proper insights from data.
Whenever utilized correctly, operational analytics can achieve a significant positive effect

Thank you!