Operation Analytics and Investigating Metric Spike

By Santosh Shinde

Description

Operation Analytics

Operation analytics is the analysis performed for a company's whole end-to-end operations.

With the aid of this, the business determines the areas in which it needs to develop, working closely with the operations team, support team, and marketing team to assist them in drawing conclusions from the data they gather.

Being one of the most crucial components of a business, this form of analysis is also utilised to forecast the general upward or downward trend in a company's fortune.

Additionally, it is used for better workflows, cross-functional team communication, and automation.

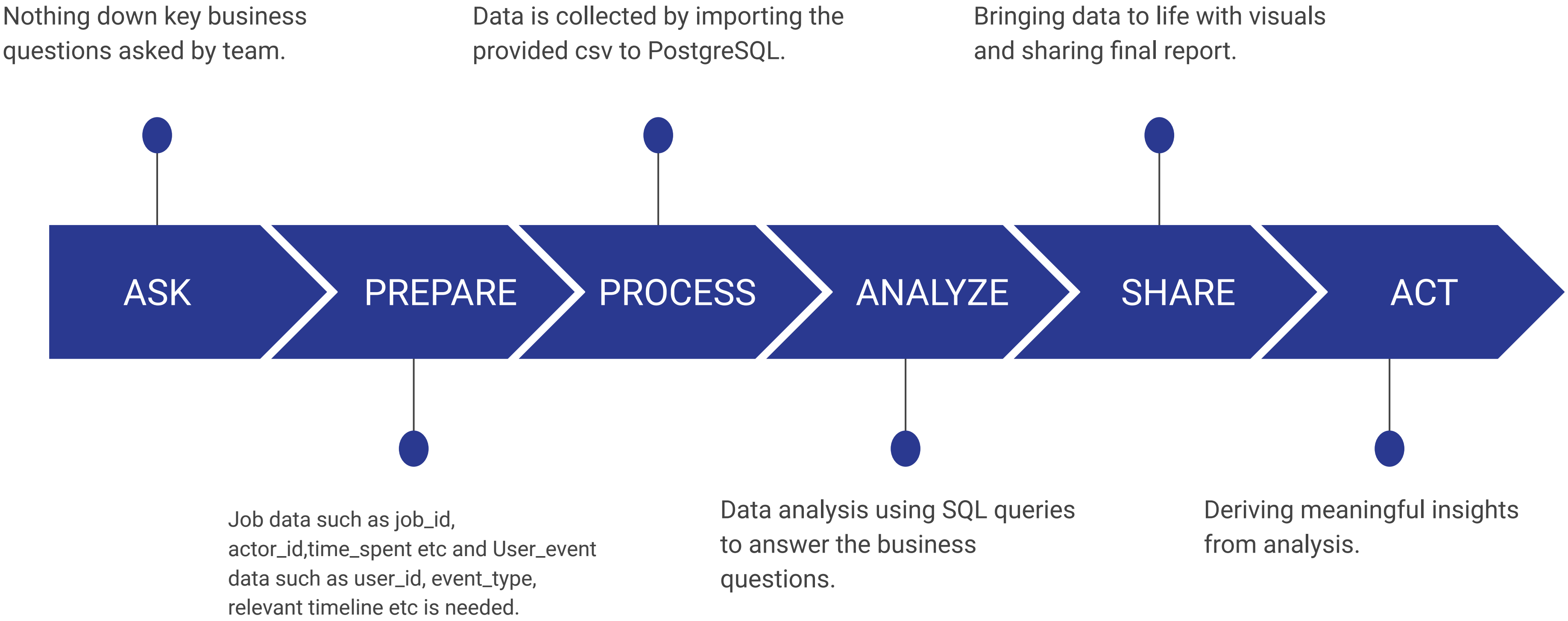
Investigating Metric Spike

Analyzing metric spikes is a crucial component of operational analytics since a data analyst needs to be able to answer queries like, "Why is there a decline in daily engagement?" or at least help other teams answer these questions.

Why have sales decreased?

Daily answers to questions like these are required, so it is crucial to look into metric increase.

Approach



ASK

Nothing down key business questions asked by team.

PREPARE

Job data such as job_id, actor_id,time_spent etc and User_event data such as user_id, event_type, relevant timeline etc is needed.

PROCESS

Data is collected by importing the provided csv to PostgreSQL.

ANALYZE

Data analysis using SQL queries to answer the business questions.

SHARE

Bringing data to life with visuals and sharing final report.

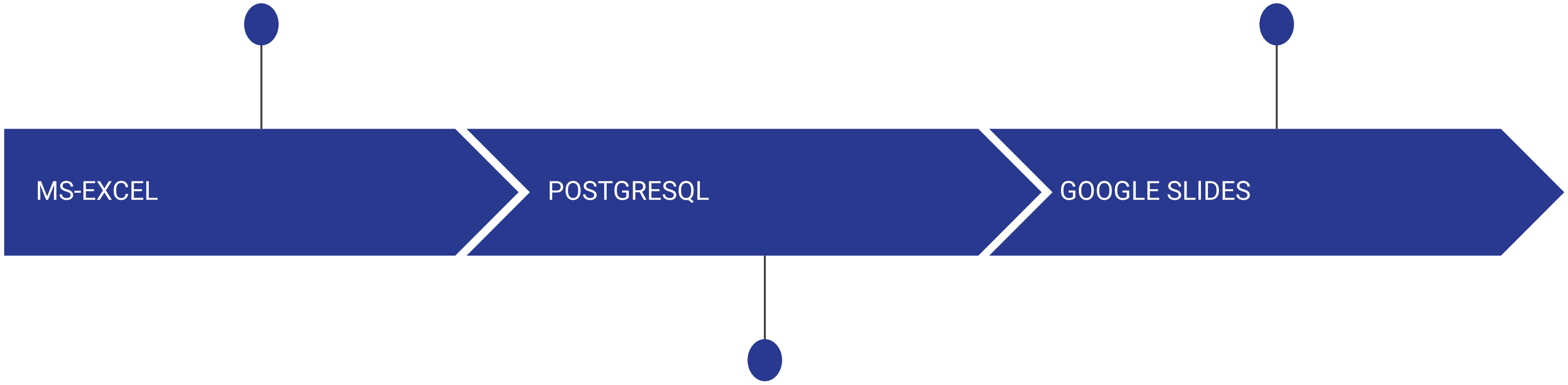
ACT

Deriving meaningful insights from analysis.

Tech-Stack Used

Data is in the form of .csv files from which the data will be processed

All the analyzed data will be visualise in the form PPT.



MS-EXCEL

POSTGRESQL

GOOGLE SLIDES

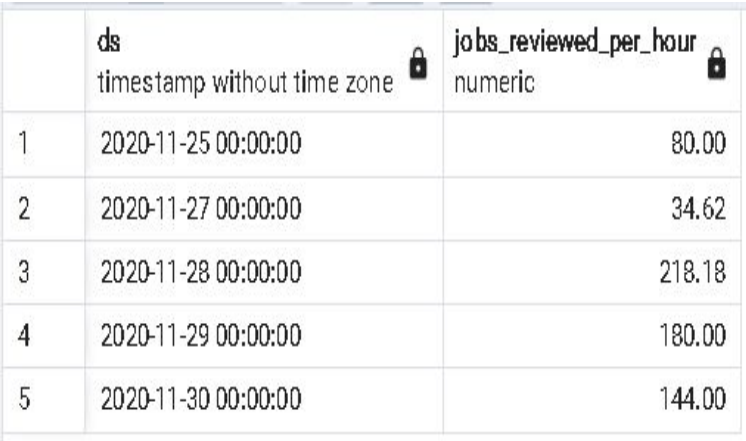
Data will imported from csv to sql format. And then the data will be analyzed according to the conditions asked by the organization.

Insights

JOB_DATA

1.Total number of jobs examined: The total number of jobs examined over time.
Your job: Determine how many jobs will be reviewed each hour and day in November 2020.

```
-- Calculate the number of jobs reviewed per hour per day for November 2020?
SELECT
        ds,
        ROUND(1.0 * COUNT(job_id) * 3600 / SUM(time_spent),
        2) AS jobs_reviewed_per_hour
FROM
        job_table
WHERE
        ds BETWEEN '2020-11-01' AND '2020-11-30'
AND event IN ('transfer' , 'decision')
GROUP BY ds
```

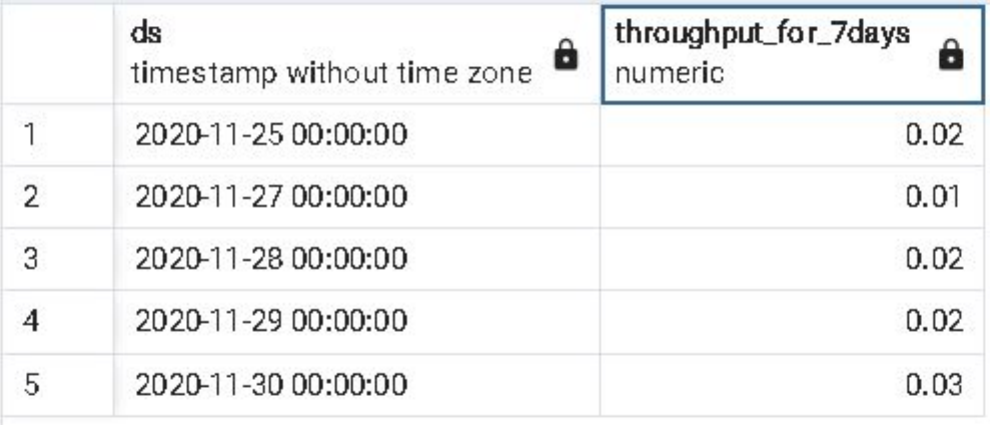


	ds timestamp without time zone	jobs_reviewed_per_hour numeric
1	2020-11-25 00:00:00	80.00
2	2020-11-27 00:00:00	34.62
3	2020-11-28 00:00:00	218.18
4	2020-11-29 00:00:00	180.00
5	2020-11-30 00:00:00	144.00

Insights

2.Throughput: It is the no. of events happening per second.
Your job: Let’s say the above metric is called throughput. Calculate 7 day rolling average of throughput.

```
-- Calculate 7 day rolling average of throughput?
with CTE AS (
SELECT
DS,
COUNT(job_id) as num_jobs,
SUM(time_spent) as total_time
From job_table
Where event In ('transfer','decision')
And ds Between '2020-11-01' and '2020-11-30'
Group by ds)
Select ds,round(1.0*sum(num_jobs) over
(Order by ds rows between 6 Preceding and current row) /
sum(total_time) OVER
(order by ds rows between 6 Preceding and current row),2) as throughput_for_7days
From CTE;
```

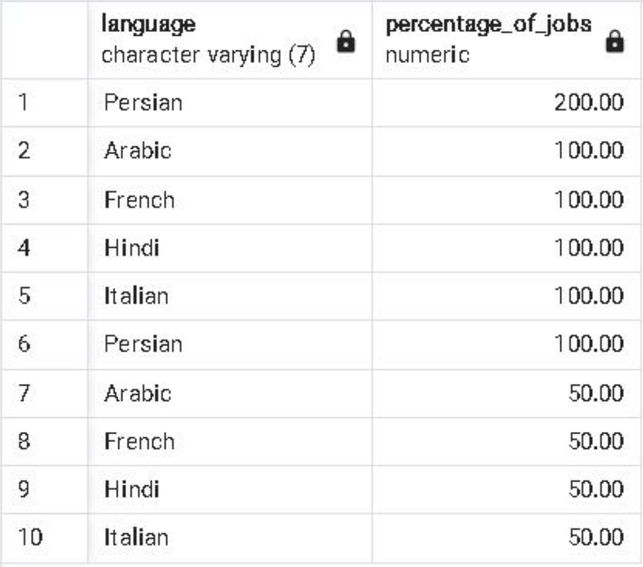


	ds timestamp without time zone	throughput_for_7days numeric
1	2020-11-25 00:00:00	0.02
2	2020-11-27 00:00:00	0.01
3	2020-11-28 00:00:00	0.02
4	2020-11-29 00:00:00	0.02
5	2020-11-30 00:00:00	0.03

Insights

3.Percentage share of each language: Share of each language for different contents.
Your job: Calculate the percentage share of each language in the last 30 days

```
-- Calculate the percentage share of each language in the last 30 days
WITH CTE AS (
SELECT Language,
COUNT(job_id) as num_jobs
From job_table
Where event In('transfer','decision')
And ds Between '2020-11-01' and '2020-11-30'Group by language),
Total as(Select COUNT(job_id) as total_jobs
From job_table
Where event In('transfer','decision')
And ds Between '2020-11-01' and '2020-11-30'
Group by language)
Select distinct Language,
Round(100*num_jobs / total_jobs,2) as percentage_of_jobs
From CTE
cross join total
Order by percentage_of_jobs DESC;
```

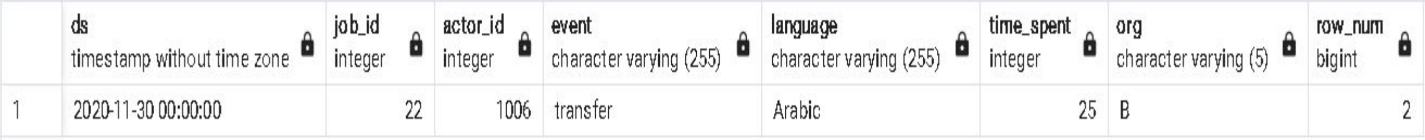


	language character varying (7)	percentage_of_jobs numeric
1	Persian	200.00
2	Arabic	100.00
3	French	100.00
4	Hindi	100.00
5	Italian	100.00
6	Persian	100.00
7	Arabic	50.00
8	French	50.00
9	Hindi	50.00
10	Italian	50.00

Insights

4.Duplicate rows: Rows that have the same value present in them.
Your job: Let’s say you see some duplicate rows in the data. How will you display duplicates from the table.

```
--Here we have created new table with similar values, after that inserted one more row.
-- Displaying duplicate rows
SELECT
*
FROM (SELECT
*,
ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id, event,
language,time_spent, org ORDER BY job_id) AS row_num
FROM job_table_2) a
WHERE row_num >1
```



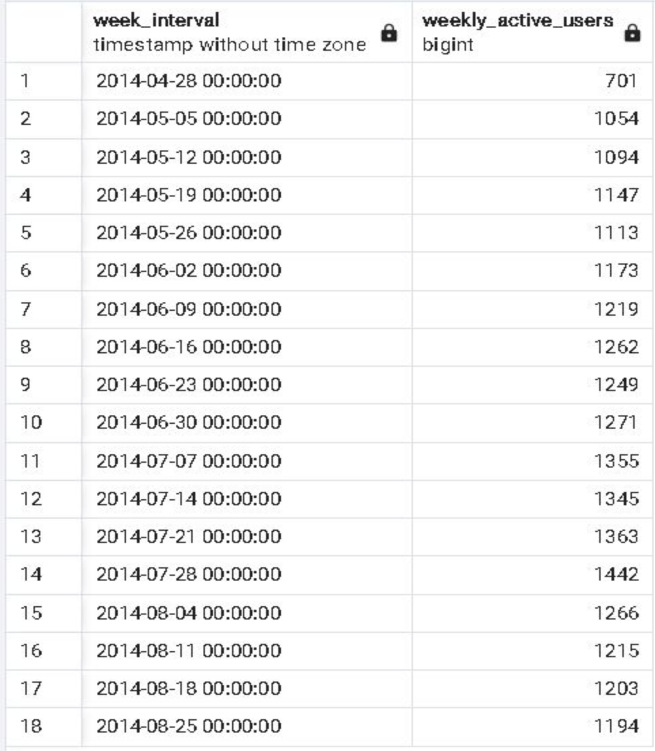
	ds	job_id	actor_id	event	language	time_spent	org	row_num
	timestamp without time zone	integer	integer	character varying (255)	character varying (255)	integer	character varying (5)	bigint
1	2020-11-30 00:00:00	22	1006	transfer	Arabic	25	B	2

Insights

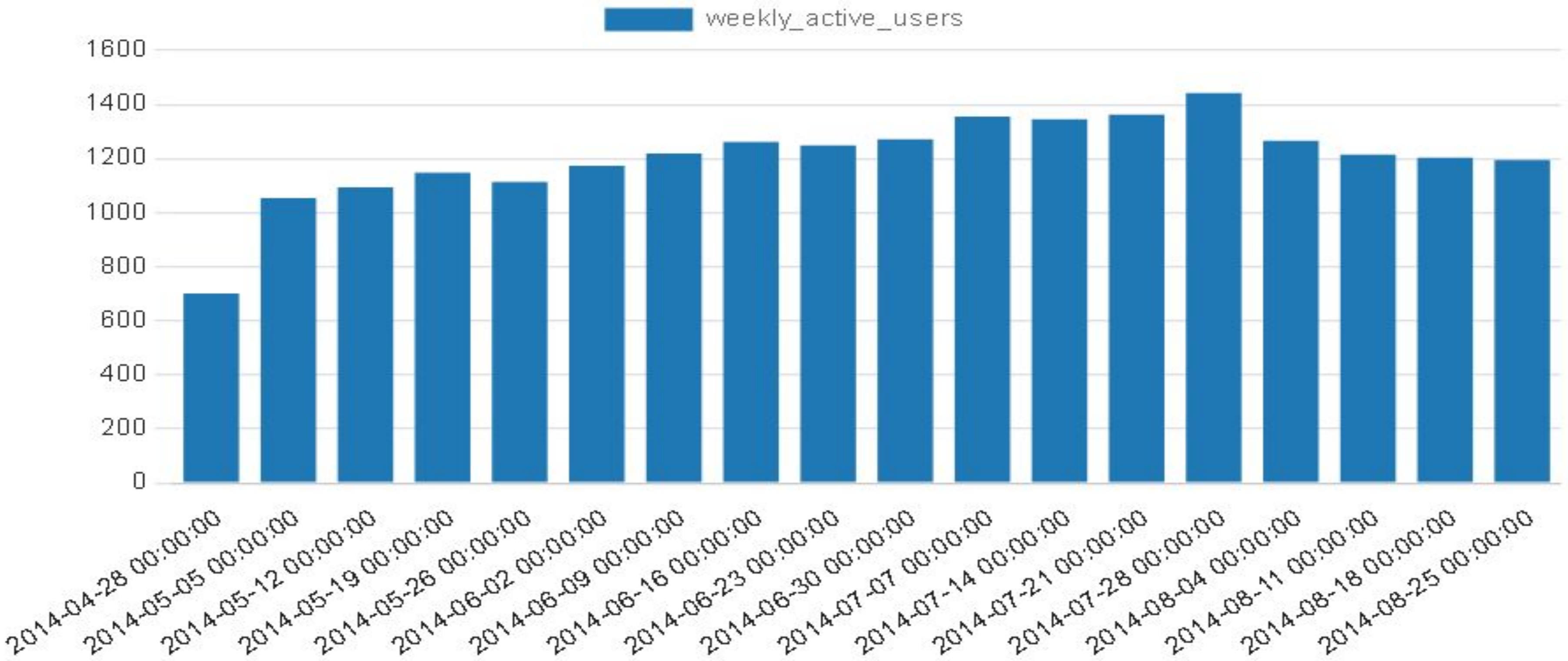
INVESTIGATING_METRIC SPIKE

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

```
-- Calculate the weekly user engagement?
SELECT
DATE_TRUNC('week', e.occurred_at) AS week_interval,
COUNT(DISTINCT e.user_id) AS weekly_active_users
FROM
events e
WHERE
e.event_type = 'engagement'
AND e.event_name = 'login'
GROUP BY
week_interval
ORDER BY
week_interval
```



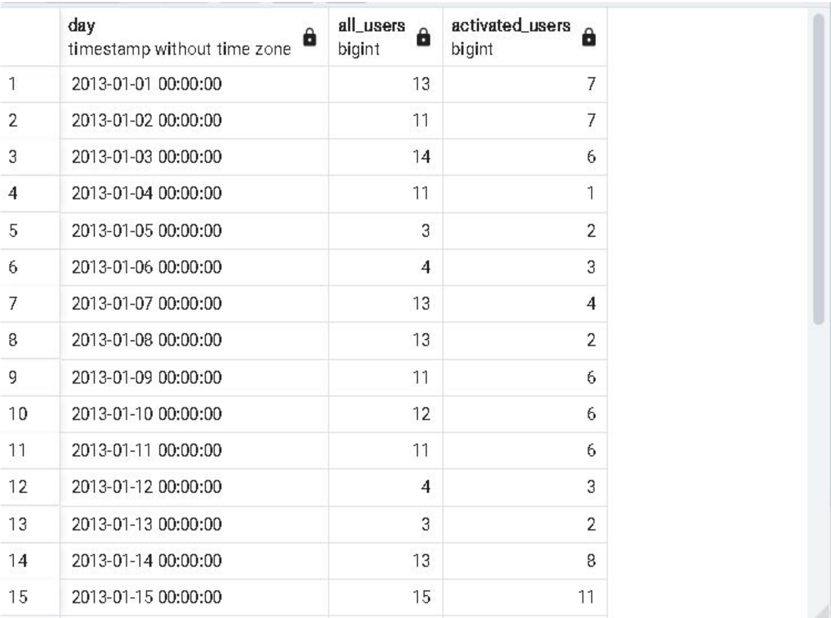
	week_interval timestamp without time zone	weekly_active_users bigint
1	2014-04-28 00:00:00	701
2	2014-05-05 00:00:00	1054
3	2014-05-12 00:00:00	1094
4	2014-05-19 00:00:00	1147
5	2014-05-26 00:00:00	1113
6	2014-06-02 00:00:00	1173
7	2014-06-09 00:00:00	1219
8	2014-06-16 00:00:00	1262
9	2014-06-23 00:00:00	1249
10	2014-06-30 00:00:00	1271
11	2014-07-07 00:00:00	1355
12	2014-07-14 00:00:00	1345
13	2014-07-21 00:00:00	1363
14	2014-07-28 00:00:00	1442
15	2014-08-04 00:00:00	1266
16	2014-08-11 00:00:00	1215
17	2014-08-18 00:00:00	1203
18	2014-08-25 00:00:00	1194


weekly_active_users
1600
1400
1200
1000
800
600
400
200
0
2014-04-28 00:00:00
2014-05-05 00:00:00
2014-05-12 00:00:00
2014-05-19 00:00:00
2014-05-26 00:00:00
2014-06-02 00:00:00
2014-06-09 00:00:00
2014-06-16 00:00:00
2014-06-23 00:00:00
2014-06-30 00:00:00
2014-07-07 00:00:00
2014-07-14 00:00:00
2014-07-21 00:00:00
2014-07-28 00:00:00
2014-08-04 00:00:00
2014-08-11 00:00:00
2014-08-18 00:00:00
2014-08-25 00:00:00

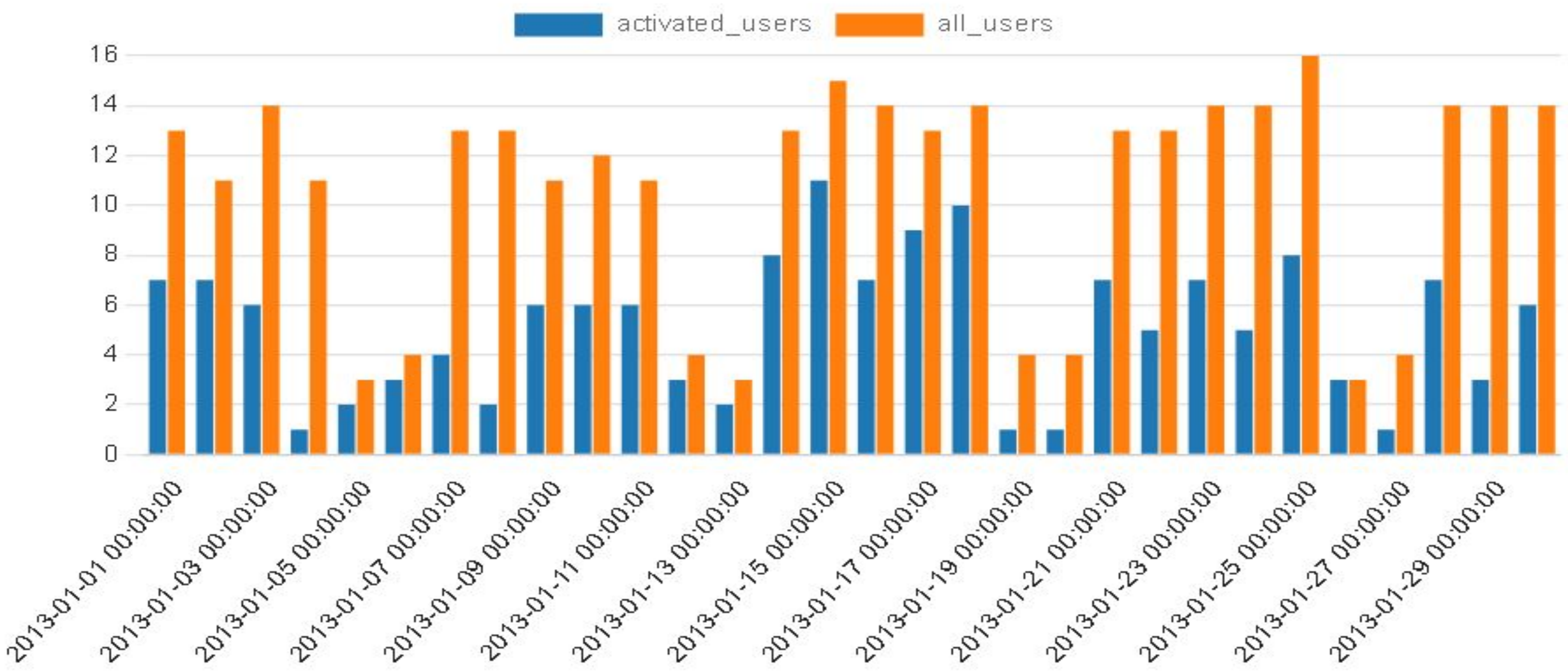
Insights

2.User Growth: Amount of users growing over time for a product.
Your job: Calculate the user growth for product.

```
-- Calculate the user growth for product?
SELECT
DATE_TRUNC('day', created_at) AS DAY,
COUNT(*) AS all_users,
COUNT(
CASE
WHEN activated_at IS NOT NULL THEN u.user_id
ELSE NULL
END
) AS activated_users
FROM
users u
WHERE created_at >= '2013-01-01'
AND created_at < '2013-01-31'
GROUP BY DAY
ORDER BY DAY
```



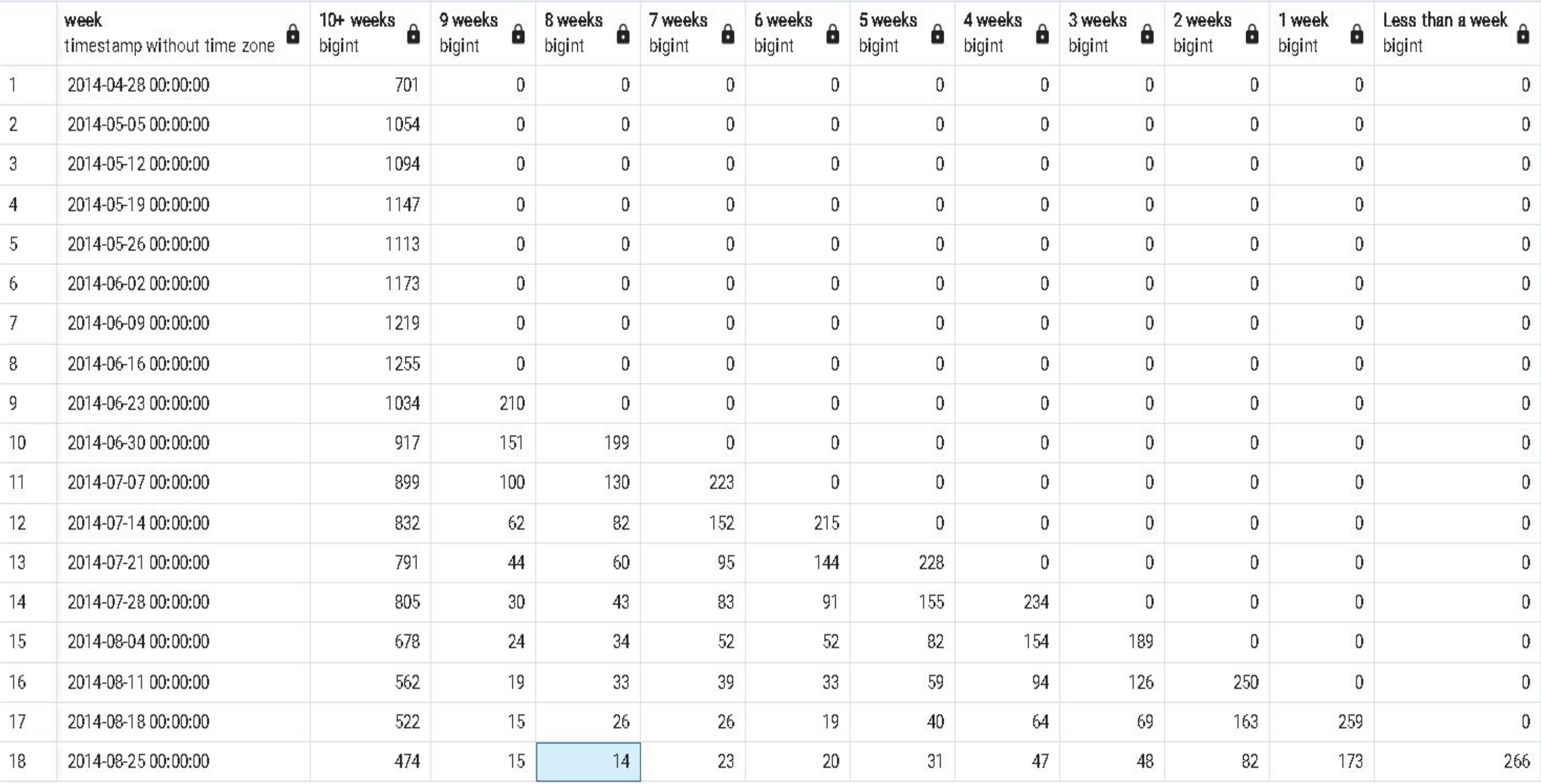
	day timestamp without time zone	all_users bigint	activated_users bigint
1	2013-01-01 00:00:00	13	7
2	2013-01-02 00:00:00	11	7
3	2013-01-03 00:00:00	14	6
4	2013-01-04 00:00:00	11	1
5	2013-01-05 00:00:00	3	2
6	2013-01-06 00:00:00	4	3
7	2013-01-07 00:00:00	13	4
8	2013-01-08 00:00:00	13	2
9	2013-01-09 00:00:00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


activated_users
all_users
16
14
12
10
8
6
4
2
0
2013-01-01 00:00:00
2013-01-03 00:00:00
2013-01-05 00:00:00
2013-01-07 00:00:00
2013-01-09 00:00:00
2013-01-11 00:00:00
2013-01-13 00:00:00
2013-01-15 00:00:00
2013-01-17 00:00:00
2013-01-19 00:00:00
2013-01-21 00:00:00
2013-01-23 00:00:00
2013-01-25 00:00:00
2013-01-27 00:00:00
2013-01-29 00:00:00

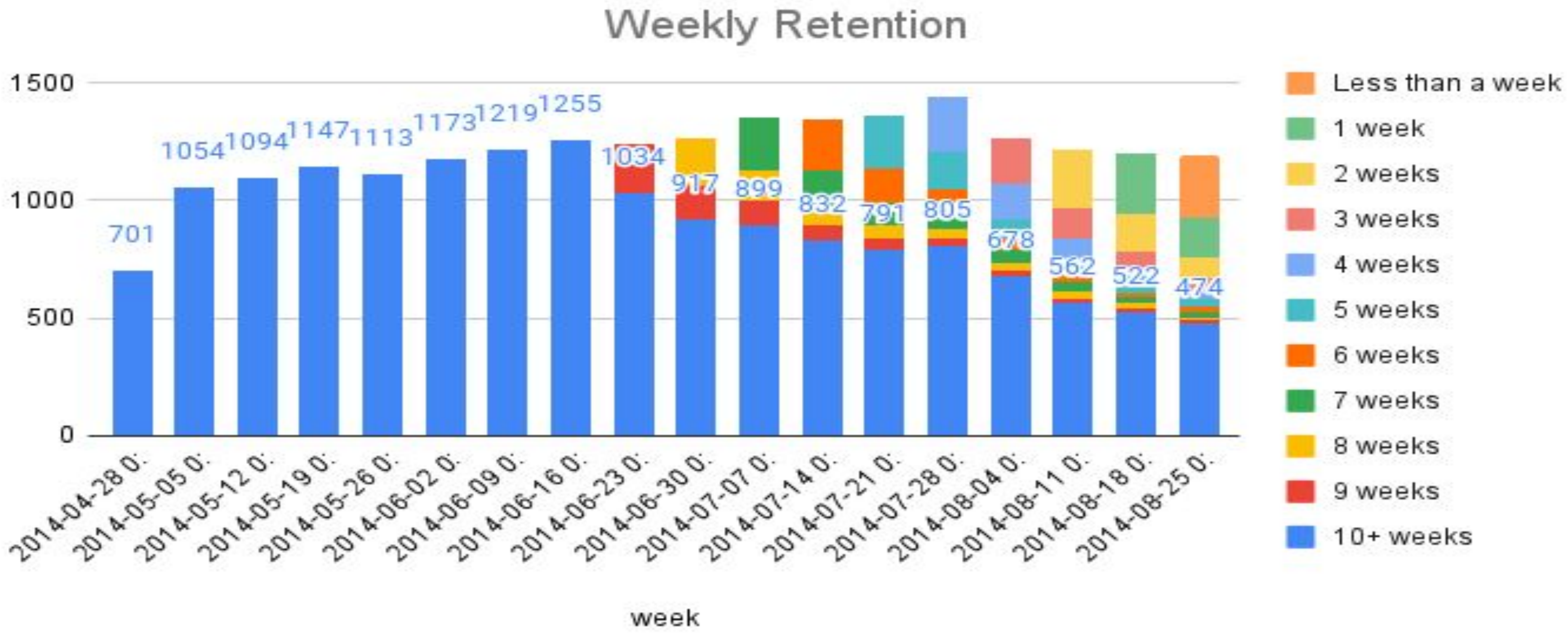
Insights

3.Weekly Retention: Users getting retained weekly after signing-up for a product.
Your job: Calculate the weekly retention of users-sign up cohort?

```
SELECT DATE_TRUNC('week',a.occurred_at) AS "week",
COUNT(DISTINCT CASE WHEN a.user_age > 70 THEN a.user_id ELSE NULL END) AS "10+ weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 70 AND a.user_age >= 63 THEN a.user_id ELSE NULL END) AS "9 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 63 AND a.user_age >= 56 THEN a.user_id ELSE NULL END) AS "8 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 56 AND a.user_age >= 49 THEN a.user_id ELSE NULL END) AS "7 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 49 AND a.user_age >= 42 THEN a.user_id ELSE NULL END) AS "6 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 42 AND a.user_age >= 35 THEN a.user_id ELSE NULL END) AS "5 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 35 AND a.user_age >= 28 THEN a.user_id ELSE NULL END) AS "4 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 28 AND a.user_age >= 21 THEN a.user_id ELSE NULL END) AS "3 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 21 AND a.user_age >= 14 THEN a.user_id ELSE NULL END) AS "2 weeks",
COUNT(DISTINCT CASE WHEN a.user_age < 14 AND a.user_age >= 7 THEN a.user_id ELSE NULL END) AS "1 week",
COUNT(DISTINCT CASE WHEN a.user_age < 7 THEN a.user_id ELSE NULL
END) AS "Less than a week"
FROM (
SELECT e.occurred_at,
u.user_id,
DATE_TRUNC('week',u.activated_at) AS activation_week,
EXTRACT('day' FROM '2014-09-01'::TIMESTAMP -
u.activated_at) 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 >= '2014-05-01'
AND e.occurred_at < '2014-09-01'
WHERE u.activated_at IS NOT NULL
) a
GROUP BY DATE_TRUNC('week',a.occurred_at)
ORDER BY DATE_TRUNC('week',a.occurred_at)
LIMIT 100
```



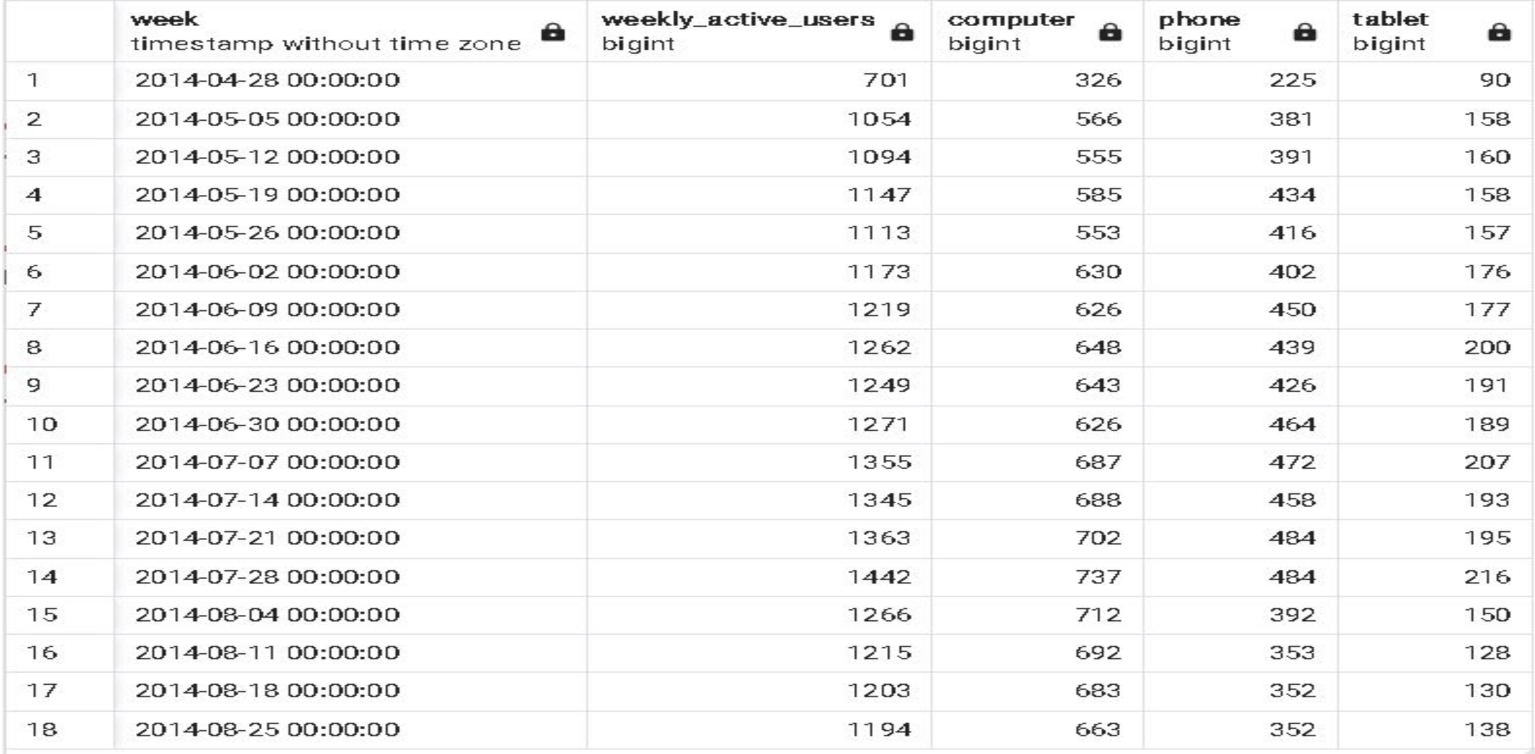
	week	10+ weeks	9 weeks	8 weeks	7 weeks	6 weeks	5 weeks	4 weeks	3 weeks	2 weeks	1 week	Less than a week
	timestamp without time zone	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint	bigint
1	2014-04-28 00:00:00	701	0	0	0	0	0	0	0	0	0	0
2	2014-05-05 00:00:00	1054	0	0	0	0	0	0	0	0	0	0
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


Weekly Retention
1500
1000
500
0
701
1054
1094
1147
1113
1173
1219
1255
1034
917
899
832
791
805
678
562
522
474
2014-04-28 0:
2014-05-05 0:
2014-05-12 0:
2014-05-19 0:
2014-05-26 0:
2014-06-02 0:
2014-06-09 0:
2014-06-16 0:
2014-06-23 0:
2014-06-30 0:
2014-07-07 0:
2014-07-14 0:
2014-07-21 0:
2014-07-28 0:
2014-08-04 0:
2014-08-11 0:
2014-08-18 0:
2014-08-25 0:
week
Less than a week
1 week
2 weeks
3 weeks
4 weeks
5 weeks
6 weeks
7 weeks
8 weeks
9 weeks
10+ weeks

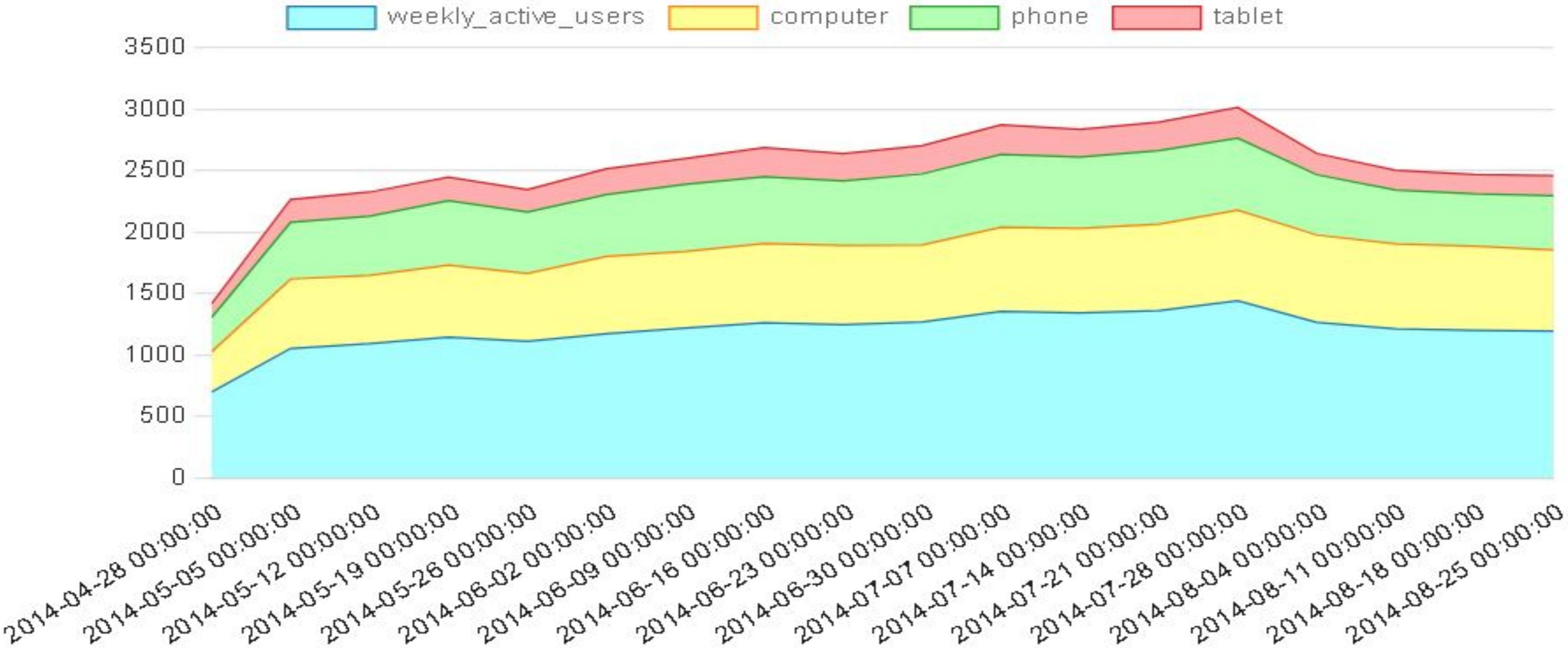
Insights

4.Weekly Engagement: To measure the activeness of a user.
Measuring if the user finds quality in product/service_weekly.
Your job: Calculate the weekly engagement per device?

```
-- Calculate the weekly user engagement?
SELECT DATE_TRUNC('week', occurred_at) AS week,
COUNT(DISTINCT e.user_id) AS weekly_active_users,
COUNT(DISTINCT CASE WHEN e.device IN ('macbook pro','lenovo
thinkpad','macbook air','dell inspiron notebook',
'asus chromebook','dell inspiron desktop','acer aspire
notebook','hp pavilion desktop','acer aspire desktop','mac mini')
THEN e.user_id ELSE NULL END) AS computer,
COUNT(DISTINCT CASE WHEN e.device IN ('iphone 5','samsung galaxy s4','nexus 5','iphone 5s','iphone 4s','nokia lumia 635', 'htc
one','samsung galaxy note','amazon fire phone') THEN e.user_id ELSE NULL END) AS phone,
COUNT(DISTINCT CASE WHEN e.device IN ('ipad air','nexus 7','ipad mini','nexus 10','kindle fire','windows surface', 'samsumg galaxy
tablet') THEN e.user_id ELSE NULL END) AS tablet FROM events e
WHERE e.event_type = 'engagement' AND e.event_name = 'login'
GROUP BY week
ORDER BY week
LIMIT 100
```



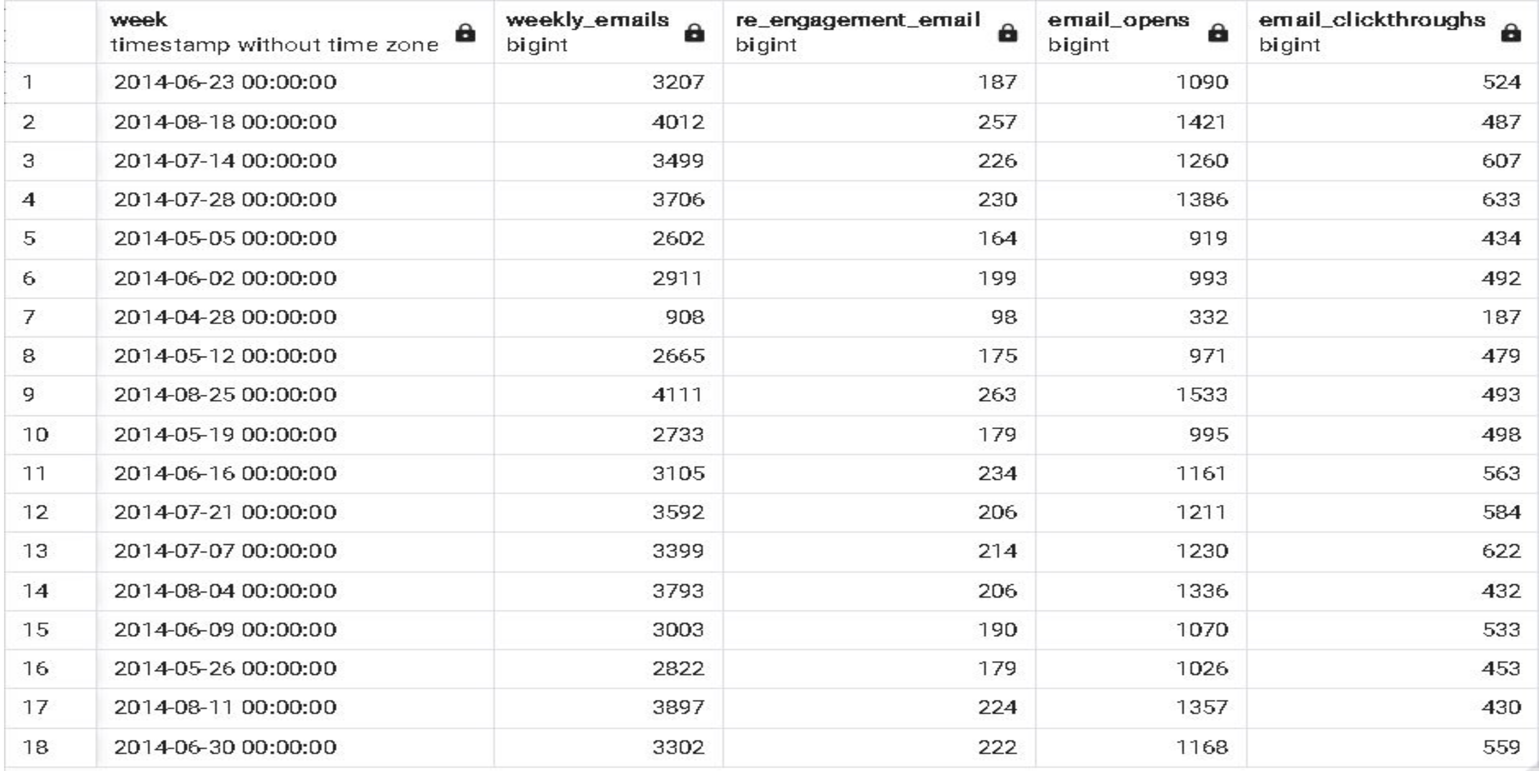
	week timestamp without time zone	weekly_active_users bigint	computer bigint	phone bigint	tablet bigint
1	2014-04-28 00:00:00	701	326	225	90
2	2014-05-05 00:00:00	1054	566	381	158
3	2014-05-12 00:00:00	1094	555	391	160
4	2014-05-19 00:00:00	1147	585	434	158
5	2014-05-26 00:00:00	1113	553	416	157
6	2014-06-02 00:00:00	1173	630	402	176
7	2014-06-09 00:00:00	1219	626	450	177
8	2014-06-16 00:00:00	1262	648	439	200
9	2014-06-23 00:00:00	1249	643	426	191
10	2014-06-30 00:00:00	1271	626	464	189
11	2014-07-07 00:00:00	1355	687	472	207
12	2014-07-14 00:00:00	1345	688	458	193
13	2014-07-21 00:00:00	1363	702	484	195
14	2014-07-28 00:00:00	1442	737	484	216
15	2014-08-04 00:00:00	1266	712	392	150
16	2014-08-11 00:00:00	1215	692	353	128
17	2014-08-18 00:00:00	1203	683	352	130
18	2014-08-25 00:00:00	1194	663	352	138


weekly_active_users
computer
phone
tablet
3500
3000
2500
2000
1500
1000
500
0
2014-04-28 00:00:00
2014-05-05 00:00:00
2014-05-12 00:00:00
2014-05-19 00:00:00
2014-05-26 00:00:00
2014-06-02 00:00:00
2014-06-09 00:00:00
2014-06-16 00:00:00
2014-06-23 00:00:00
2014-06-30 00:00:00
2014-07-07 00:00:00
2014-07-14 00:00:00
2014-07-21 00:00:00
2014-07-28 00:00:00
2014-08-04 00:00:00
2014-08-11 00:00:00
2014-08-18 00:00:00
2014-08-25 00:00:00

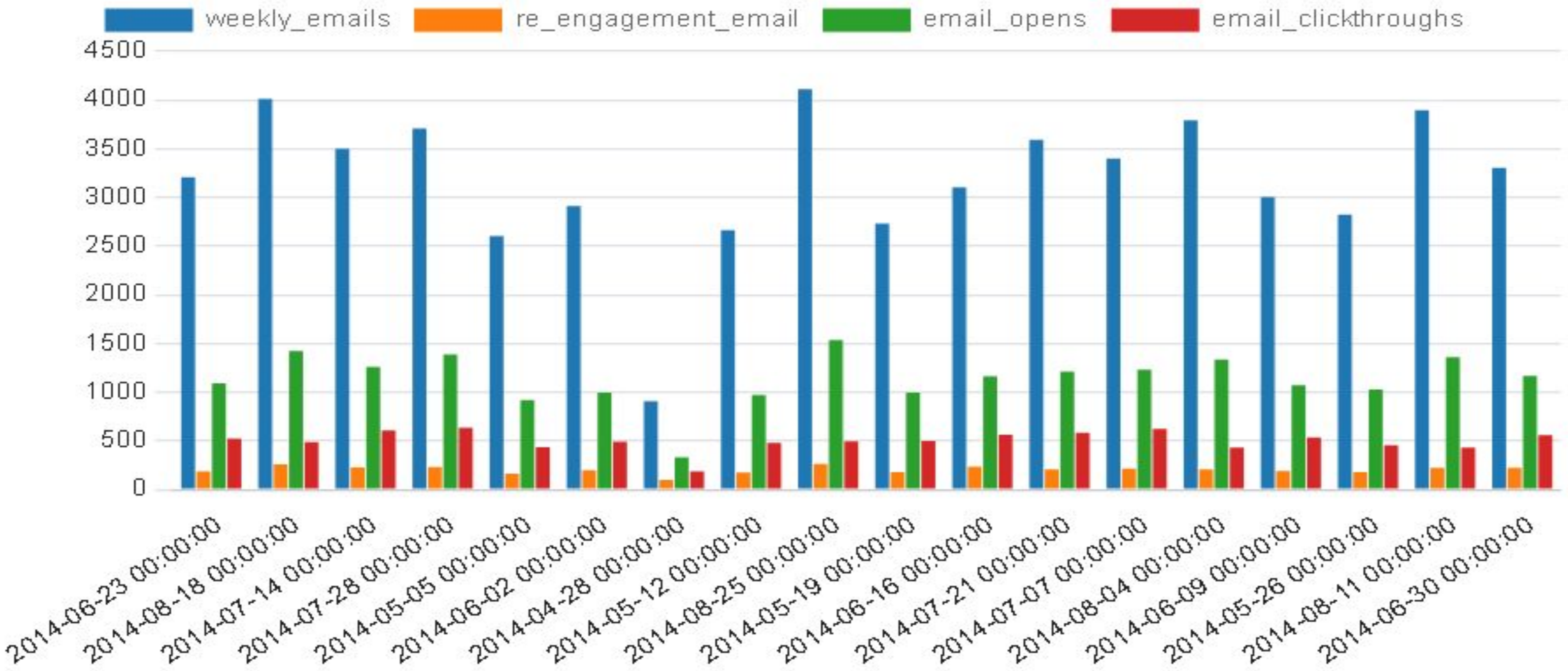
Insights

5.Email Engagement: Users engaging with the email service.
Your job: Calculate the email engagement metrics?

```
-- Calculate the email engagement metrics
select date_trunc('week', occurred_at) as week,
count(case when e.action='sent_weekly_digest' then e.user_id else null end) as weekly_emails,
count(case when e.action='sent_reengagement_email' then e.user_id else null end) as re_engagement_email,
count(case when e.action='email_open' then e.user_id else null end) as email_opens,
count(case when e.action='email_clickthrough' then e.user_id else null end) as email_clickthroughs
from email_events e
group by 1;
```



	week timestamp without time zone	weekly_emails bigint	re_engagement_email bigint	email_opens bigint	email_clickthroughs bigint
1	2014-06-23 00:00:00	3207	187	1090	524
2	2014-08-18 00:00:00	4012	257	1421	487
3	2014-07-14 00:00:00	3499	226	1260	607
4	2014-07-28 00:00:00	3706	230	1386	633
5	2014-05-05 00:00:00	2602	164	919	434
6	2014-06-02 00:00:00	2911	199	993	492
7	2014-04-28 00:00:00	908	98	332	187
8	2014-05-12 00:00:00	2665	175	971	479
9	2014-08-25 00:00:00	4111	263	1533	493
10	2014-05-19 00:00:00	2733	179	995	498
11	2014-06-16 00:00:00	3105	234	1161	563
12	2014-07-21 00:00:00	3592	206	1211	584
13	2014-07-07 00:00:00	3399	214	1230	622
14	2014-08-04 00:00:00	3793	206	1336	432
15	2014-06-09 00:00:00	3003	190	1070	533
16	2014-05-26 00:00:00	2822	179	1026	453
17	2014-08-11 00:00:00	3897	224	1357	430
18	2014-06-30 00:00:00	3302	222	1168	559


weekly_emails
re_engagement_email
email_opens
email_clickthroughs
4500
4000
3500
3000
2500
2000
1500
1000
500
0
2014-06-23 00:00:00
2014-08-18 00:00:00
2014-07-14 00:00:00
2014-07-28 00:00:00
2014-05-05 00:00:00
2014-06-02 00:00:00
2014-04-28 00:00:00
2014-05-12 00:00:00
2014-08-25 00:00:00
2014-05-19 00:00:00
2014-06-16 00:00:00
2014-07-21 00:00:00
2014-07-07 00:00:00
2014-08-04 00:00:00
2014-06-09 00:00:00
2014-05-26 00:00:00
2014-08-11 00:00:00
2014-06-30 00:00:00

Results

- **User engagement:** We observe a significant increase and a sharp decline around the end of July and the beginning of August; this could be the result of an advertising event.

- **User growth:** Growth is still robust during the week and low on the weekends.

- **Weekly retention:** The engagement of users who have been registered for more than 10 weeks is decreasing, as shown in the graph.

- **Weekly engagement by device:** There is a significant decline in phone engagement rates when compared to computer and tablet engagement rates, indicating that there may be a problem with the service's mobile app that is preventing long-term user retention.

- **Email engagement:** There may be an issue with the content in the email as indicated by the sudden decline in weekly emails, re-engagement emails, email opens, and email click throughs in the late 4th week of April 2014. As a result, the organisation should concentrate on email content marketing.

Thank You.