



# Operation Analytics & Investigating Metric Spike

Trainity Project Report  
Rohit Kumar  
[rohitk.ug20.ce@nitp.ac.in](mailto:rohitk.ug20.ce@nitp.ac.in)

# Description

The study carried out for the entire end-to-end operations of a corporation is covered in this report's discussion of Operation Analytics and Investigating Metric Spike. This helps the business identify the areas where it needs to make improvements. The VP, top officials, operational team, support staff, marketing team, sales team, etc. can all benefit from this report.

Being one of the most crucial components of a business, this form of analysis is also utilized to forecast the general upward or downward trend in a company's fortune. Better automation, improved communication among cross-functional teams, and more efficient workflows are the results.

This report is based on the "Job data" dataset, which was used in the previous case study's operation analytics. Case Study 2 will use the "Yammer dataset" to investigate metric spikes and provide answers to queries about user engagement, user growth, email engagement metrics, and other topics.

# Approach

This report will be divided up primarily into two sections.

## 1) Operation Analytics

- I. **Number of jobs reviewed** - To number of jobs reviewed per hour per day for November 2020?
- II. **Throughput** - will calculate on weekly and daily basis
- III. **Percentage share of each language** -To find percentage share of each language in the last 30 days?
- IV. **Duplicate rows** - To highlight the duplicate rows

# Approach

## 2) Investigating Metric Spike

- I. **User Engagement** - User engagement basis on activeness of user on weekly basis
- II. **User Growth** - Growth of active user on weekly basis
- III. **Weekly Retention** - Applying cohort analysis to calculate the weekly retention of users-sign up.
- IV. **Weekly Engagement** - calculation of weekly engagement per device
- V. **Email Engagement** – Email Engagement metrics

# Tech-Stack Used

- For this Project I used **PostgreSQL PG Admin 4 version 6.8**

About pgAdmin 4	
Version	6.8
Application Mode	Desktop
Current User	pgadmin4@pgadmin.org
NW.js Version	0.55.0
Browser	Chromium 92.0.4515.107
Operating System	Windows-10-10.0.19045-SP0

- This project is based on the provided “Case\_Study” dataset.

[\[ CASE\\_STUDY\\_Dataset \]](#)



**ANALYSIS  
&  
RESULT**

Browser

- > ig\_clone
- > ops & investigation
  - > Casts
  - > Catalogs
  - > Event Triggers
  - > Extensions
  - > Foreign Data Wrappers
  - > Languages
  - > Publications
  - > Schemas (1)
    - > public
      - > Aggregates
      - > Collations
      - > Domains
      - > FTS Configurations
      - > FTS Dictionaries
      - > FTS Parsers
      - > FTS Templates
      - > Foreign Tables
      - > Functions
      - > Materialized Views
      - > Operators
      - > Procedures
      - > Sequences
      - > Tables
      - > Trigger Functions
      - > Types
      - > Views
    - > Subscriptions
  - > postgres
  - > Login/Group Roles

Dashboard Properties SQL Statistics Dependencies Dependents ops &amp; investigation/postgres@PostgreSQL 14 \*

ops &amp; investigation/postgres@PostgreSQL 14

Query Editor Query History

```
1 create table job_data
2 (
3     ds date ,
4     job_id int,
5     actor_id int,
6     event varchar,
7     language varchar,
8     time_spent int,
9     org varchar
10 );
11
12
13
14
```

Data Output Explain Messages Notifications

CREATE TABLE

Query returned successfully in 132 msec.

```
1  
2 select * from job_data;
```

[Data Output](#)   [Explain](#)   [Messages](#)   [Notifications](#)

ds	job_id	actor_id	event	language	time_spent	org
date	integer	integer	character varying	character varying	integer	character varying



pgAdmin 4

pgAdminFileObjectToolsHelp

BrowseDashboardPropertiesSQLStatisticsDependenciesDependentsops & investigation/postgres@PostgreSQL 14 \*X

ig\_clone

ops & investigation

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables

Trigger Functions

Types

Views

Subscriptions

postgres

Login/Group Roles

ops & investigation/postgres@PostgreSQL 14

Query EditorQuery History

1

2

3

4

5

6

```
COPY job_data (ds, job_id,actor_id,event,language,time_spent,org)
FROM 'E:\Trainity\Own\Operation Analytics & Investigating Metric Spikes\DATASET\Table-1 Job_data.csv'
DELIMITER ','
CSV HEADER;
```

Data Output

Explain

Messages

Notifications

COPY 11

Query returned successfully in 92 msec.

Query returned successfully in 92 msec.

23°C Haze

13:14

15-02-2023





**CASE STUDY**

**01**

# 1 (I) Numbers of Job Reviewed

- Number of jobs reviewed per hour per day for November 2020
- 0.0083 jobs reviewed per hour per day in November 2020

Query:-

```
SELECT  
COUNT(DISTINCT job_id)/(30 * 24.0) AS jobs_reviewed  
FROM job_data  
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30';
```

pgAdmin 4

pgAdminFileObjectToolsHelp

BrowseDashboardPropertiesSQLStatisticsDependenciesDependentsops & investigation/postgres@PostgreSQL 14 \*

ig\_clone

ops & investigation

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables

Trigger Functions

Types

Views

Subscriptions

postgres

Login/Group Roles

ops & investigation/postgres@PostgreSQL 14

Query EditorQuery History

1SELECT

2COUNT(DISTINCT job\_id)/(30 \* 24.0) AS jobs\_reviewed

3FROM job\_data

4WHERE ds BETWEEN '2020-11-01' AND '2020-11-30';

Data OutputExplainMessagesNotifications

jobs\_reviewed

numeric

10.008333333333333

23°C Haze

13:20

15-02-2023

# 1 (II) Throughput

- The weekly throughput is 0.03.
- On date 11/28/2020 the throughput is highest 0.06

## Weekly Throughput

weekly_throughput
0.03

## Daily Throughput

dates	daily_throughput
11/25/2020	0.02
11/26/2020	0.02
11/27/2020	0.01
11/28/2020	0.06
11/29/2020	0.05
11/30/2020	0.05

# 1 (II) Throughput

Query:-


Weekly Throughput

```
SELECT  
ROUND(COUNT(event)*1.00/SUM(time_spent), 2) AS Weekly_Throughput  
FROM job_data;
```

```
3 FROM job_data;
```

- public
- Aggregates

- Collations

>  FIS Configurations


```

> Aa FIS Parsers
  weekly_throughput
  numeric

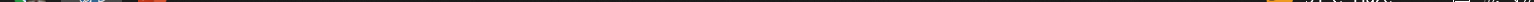
```

> Foreign Tables

>  Materialized Views

>  Procedures

>  Tables





# 1 (II) Throughput

Query:-

## Daily Throughput

```
SELECT  
ds AS Dates,  
ROUND(COUNT(event)*1.00/SUM(time_spent), 2) AS Daily_Throughput  
FROM job_data  
GROUP BY ds  
ORDER BY ds;
```

pgAdmin 4

FileObjectToolsHelp

Object Explorer

Servers (1)

PostgreSQL 15

Databases (3)

Instagram User Analytics

OPERATION AND METRIC ANALYTICS

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

1.3 Sequences

Tables

Trigger Functions

Types

Views

DashboardPropertiesSQLStatisticsDependenciesDependentsProcesses

OPERATION AND METRIC ANALYTICS/postgres@PostgreSQL 15\*

OPERATION AND METRIC ANALYTICS/postgres@PostgreSQL...

No limit

QueryQuery History

1SELECT

2dsASDates,

3ROUND(COUNT(event)\*1.00/SUM(time\_spent), 2)ASDaily\_Throughput

4FROMjob\_data

5GROUPBYds

6ORDERBYds;

7

Data OutputMessagesGraph VisualiserNotifications

datesdate

daily\_throughputnumeric

12020-11-250.02

22020-11-260.02

32020-11-270.01

42020-11-280.06

52020-11-290.05

62020-11-300.05

7[null][null]

Total rows: 7 of 7Query complete 00:00:00.135

Ln 3, Col 24

31°C Haze

12:21 AM7/28/2023

# 1 (III) Percentage share of each language

- find percentage share of each language in the last 30 days?

Language	num_jobs	total_jobs	perc_jobs
Arabic	1	8	12.50
English	1	8	12.50
French	1	8	12.50
Hindi	1	8	12.50
Italian	1	8	12.50
Persian	3	8	37.50

# 1 (III) Percentage share of each language

Query:-

```
WITH Base AS
( SELECT Language,
      COUNT(job_id) AS num_jobs
FROM job_data
      WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
      GROUP BY language),
total AS ( SELECT COUNT( job_id) AS total_jobs
FROM job_data
)
SELECT language, num_jobs, total_jobs, ROUND(100.0*num_jobs/total_jobs,2) AS perc_jobs
FROM Base
CROSS JOIN
total ORDER BY language
```

pgAdmin 4

pgAdminFileObjectToolsHelp

Browse

ig\_clone

ops & investigation

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables

Trigger Functions

Types

Views

Subscriptions

postgres

Login/Group Roles

DashboardPropertiesSQLStatisticsDependenciesDependentsops & investigation/postgres@PostgreSQL 14 \*

ops & investigation/postgres@PostgreSQL 14

Query EditorQuery History

```
1 WITH Base AS
2 ( SELECT Language,
3     COUNT(job_id) AS num_jobs
4 FROM job_data
5 WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
6 GROUP BY language),
7 total AS ( SELECT COUNT( job_id) AS total_jobs
8 FROM job_data
9 )
10 SELECT language, num_jobs, total_jobs, ROUND(100.0*num_jobs/total_jobs,2) AS perc_jobs
11 FROM Base
12 CROSS JOIN
13 total ORDER BY language
```

Data OutputExplainMessagesNotifications

	language character varying	num_jobs bigint	total_jobs bigint	perc_jobs numeric
1	Arabic	1	8	12.50
2	English	1	8	12.50
3	French	1	8	12.50
4	Hindi	1	8	12.50
5	Italian	1	8	12.50
6	Persian	3	8	37.50

23°C Haze

16:24

15-02-2023

# 1 (IV) Duplicate rows

- Highlight the duplicate rows:-
  - Here we can highlight duplicate rows basis on their row number ('dup\_flag' column) which is based on their occurrence repetition.
  - All the rows have dup\_flag higher than 1 are duplicate rows.

**Note** : Here all key column are consider for duplication entries There is no duplicate row to show present in dataset

# 1 (IV) Duplicate rows

Query:-

```
SELECT *  
FROM (  
  SELECT *, ROW_NUMBER() OVER (PARTITION BY job_id, actor_id, event, language)  
  AS dup_flag  
  FROM job_data ) AS a  
WHERE dup_flag > 1;
```

pgAdmin 4

pgAdminFileObjectToolsHelp

Browser

ig\_clone

ops & investigation

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables

Trigger Functions

Types

Views

Subscriptions

postgres

Login/Group Roles

DashboardPropertiesSQLStatisticsDependenciesDependentsops & investigation/postgres@PostgreSQL 14 \*

ops & investigation/postgres@PostgreSQL 14

Query EditorQuery History

```
1 SELECT *
2 FROM (
3 SELECT *,
4 ROW_NUMBER() OVER (PARTITION BY job_id, actor_id, event, language) AS dup_flag
5 FROM job_data
6 ) as a
7 WHERE
8 dup_flag > 1;
```

Data OutputExplainMessagesNotifications

	ds date	job_id integer	actor_id integer	event character varying	language character varying	time_spent integer	org character varying	dup_flag bigint
1	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2
2	[null]	[null]	[null]	[null]	[null]	[null]	[null]	3

Successfully run. Total query runtime: 128 msec. 2 rows affected.

Successfully run. Total query runtime: 99 msec. 0 rows affected.



# Insights

- i. This is very less amount of data to take insights from however,
- ii. We can conclude there is less quantity of work for each actor i.e getting 1 or 2 job per day
- iii. Dataset was clean and there is no single duplicate entry
- iv. We are getting equal quantity of work base on language and have stable work load for each day.

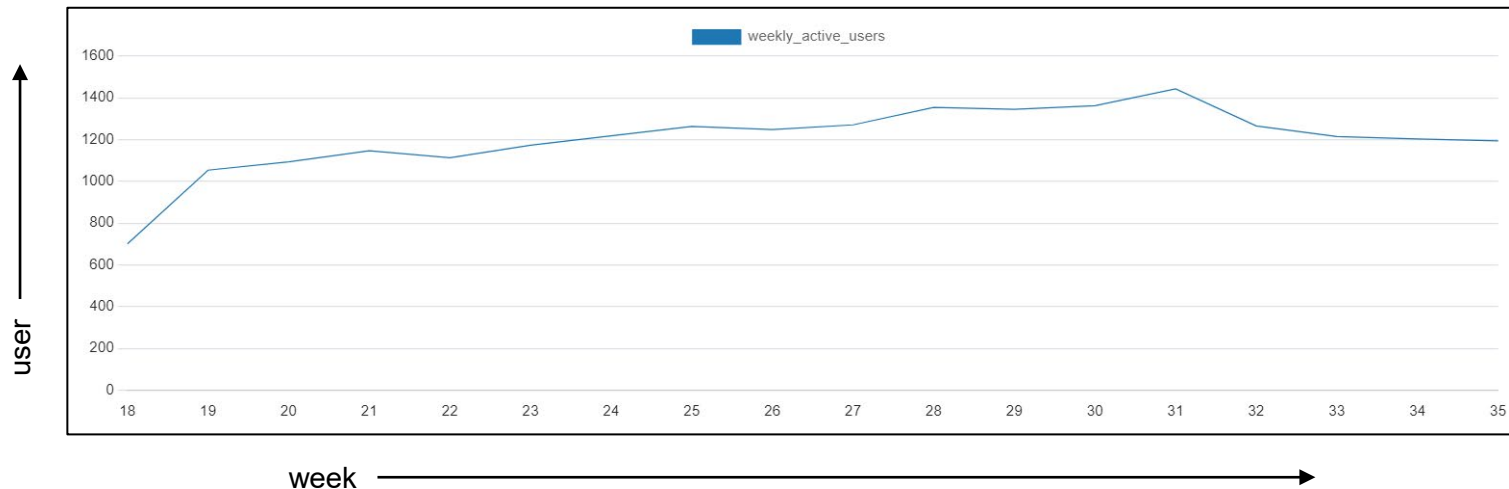
The graphic consists of three concentric circles. The innermost circle is a deep red color. The middle circle is a lighter, coral-like red. The outermost circle is a light orange or peach color. The text is centered within the innermost red circle.

**CASE STUDY**

**02**

## 2 (I) User Engagement

- User engagement basis on activeness of user on weekly basis



## 2 (I) User Engagement

Query:-

```
SELECT  
EXTRACT(WEEK FROM occurred_at) AS Week_Numbers,  
COUNT(DISTINCT user_id) AS Weekly_Active_Users  
FROM events  
WHERE event_type = 'engagement'  
GROUP BY Week_Numbers;
```

pgAdmin 4

FileObjectToolsHelp

Object Explorer

Servers (1)

PostgreSQL 15

Databases (3)

Instagram User Analytics

OPERATION AND METRIC ANALYTICS

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (4)

email\_events

events

job\_data

DashboardPropertiesSQLStatisticsDependenciesDependentsProcesses

OPERATION AND METRIC ANALYTICS/postgres@PostgreSQL 15\*

OPERATION AND METRIC ANALYTICS/postgres@PostgreSQL...

No limit

QueryQuery History

1SELECT

2EXTRACT(WEEK FROM occurred\_at) AS Week\_Numbers,

3COUNT(DISTINCT user\_id) AS Weekly\_Active\_Users

4FROM events

5WHERE event\_type = 'engagement'

6GROUP BY Week\_Numbers;

Data OutputMessagesGraph VisualiserNotifications

week\_numbers

numeric

week\_active\_users

bigint

118701

2191054

3201094

4211147

5221113

6231173

7241219

8251263

9261249

10271271

11281355

12291345

13301363

14311443

15321366

Total rows: 18 of 18

Query complete 00:00:00.346

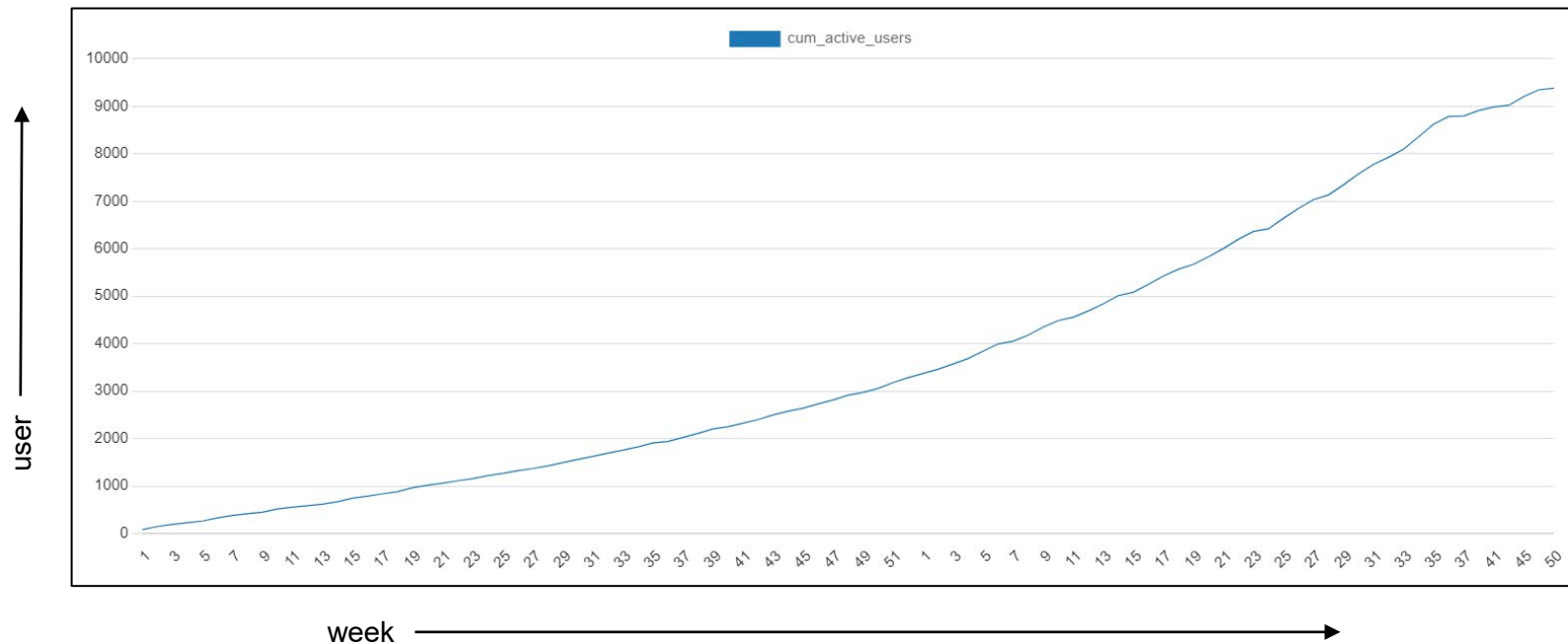
Ln 6, Col 23

29°C Mostly cloudy

3:19 AM7/28/2023

## 2 (II) User Growth

- Growth of active user on weekly basis



## 2 (II) User Growth

Query:-

```
SELECT year, weeknum, num_active_users,  
SUM(num_active_users) OVER (ORDER BY year, weeknum ROWS BETWEEN  
UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_active_users  
FROM  
(SELECT EXTRACT( YEAR FROM activated_at) AS year,  
EXTRACT(WEEK FROM activated_at) AS weeknum,  
COUNT(DISTINCT user_id) AS num_active_users  
FROM users a  
WHERE state='active'  
GROUP BY year, weeknum  
ORDER BY year, weeknum ) a;
```





## 2 (III) Weekly Retention

### Query:-

```
SELECT first_login_week,  
SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS week_0,  
SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS week_1,  
SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS week_2,  
SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS week_3,  
SUM(CASE WHEN week_number = 4 THEN 1 ELSE 0 END) AS week_4,  
SUM(CASE WHEN week_number = 5 THEN 1 ELSE 0 END) AS week_5,  
SUM(CASE WHEN week_number = 6 THEN 1 ELSE 0 END) AS week_6,  
SUM(CASE WHEN week_number = 7 THEN 1 ELSE 0 END) AS week_7,  
SUM(CASE WHEN week_number = 8 THEN 1 ELSE 0 END) AS week_8,  
SUM(CASE WHEN week_number = 9 THEN 1 ELSE 0 END) AS week_9,  
SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS week_10,  
SUM(CASE WHEN week_number = 11 THEN 1 ELSE 0 END) AS week_11,  
SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS week_12,  
SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS week_13,  
SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS week_14,  
SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS week_15,  
SUM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS week_16,  
SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS week_17,  
SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS week_18
```

## 2 (III) Weekly Retention

### Query (Continue):-

```
FROM
(SELECT log.user_id, login_week, first_login_week, login_week - first_login_week AS week_number
FROM (SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS login_week
      FROM events
      WHERE event_name = 'login'
      GROUP BY 1, 2) log
INNER JOIN
      (SELECT user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS first_login_week
      FROM events
      WHERE event_name = 'login'
      GROUP BY 1) first_log
ON log.user_id = first_log.user_id) sub_week_no
GROUP BY 1
ORDER BY 1;
```

[illegible]

pgAdmin 4

FileObjectToolsHelp

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesOPERATION AND METRIC ANALYTICS/postgres@PostgreSQL 15\*

Servers (1)OPERATION AND METRIC ANALYTICS/postgres@PostgreSQL 15\*

PostgreSQL

Databases

InstagrQueryQuery History

OPERA

Cast

Cata

Even

Exter

Forel

Lang

Publ

Sche

PL

FROM

(SELECT log.user\_id, login\_week, first\_login\_week, login\_week - first\_login\_week AS week\_number

FROM (SELECT user\_id, EXTRACT(WEEK FROM occurred\_at) AS login\_week

FROM events

WHERE event\_name = 'login'

GROUP BY 1, 2) log

INNER JOIN

(SELECT user\_id, MIN(EXTRACT(WEEK FROM occurred\_at)) AS first\_login\_week

FROM events

WHERE event\_name = 'login'

GROUP BY 1) first\_log

Total rows: 18 of 18Query complete 00:00:00.337

Ln 24, Col 4

```
1 SELECT first_login_week,
2 SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS week_0,
3 SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS week_1,
4 SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS week_2,
5 SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS week_3,
6 SUM(CASE WHEN week_number = 4 THEN 1 ELSE 0 END) AS week_4,
7 SUM(CASE WHEN week_number = 5 THEN 1 ELSE 0 END) AS week_5,
8 SUM(CASE WHEN week_number = 6 THEN 1 ELSE 0 END) AS week_6,
9 SUM(CASE WHEN week_number = 7 THEN 1 ELSE 0 END) AS week_7,
10 SUM(CASE WHEN week_number = 8 THEN 1 ELSE 0 END) AS week_8,
11 SUM(CASE WHEN week_number = 9 THEN 1 ELSE 0 END) AS week_9,
12 SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS week_10,
13 SUM(CASE WHEN week_number = 11 THEN 1 ELSE 0 END) AS week_11,
14 SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS week_12,
15 SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS week_13,
16 SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS week_14,
17 SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS week_15,
18 SUM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS week_16,
19 SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS week_17,
20 SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS week_18
21 FROM
22 (SELECT log.user_id, login_week, first_login_week, login_week - first_login_week AS week_number
23 FROM (SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS login_week
24 FROM events
25 WHERE event_name = 'login'
26 GROUP BY 1, 2) log
27 INNER JOIN
28 (SELECT user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS first_login_week
29 FROM events
30 WHERE event_name = 'login'
31 GROUP BY 1) first_log
```

29°C Mostly cloudy4:51 AM7/28/2023

## 2 (IV) Weekly Engagement

- calculation of weekly engagement per device

Query:-

```
SELECT
EXTRACT(YEAR FROM occurred_at) AS year,
EXTRACT(WEEK FROM occurred_at) AS week,
device,
COUNT(DISTINCT user_id )
FROM events
WHERE event_type = 'engagement'
GROUP BY 1,2,3
ORDER BY 1,2,3;
```



## 2 (v) Email Engagement

week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
28	3399	214	1230	622
30	3592	206	1211	584
34	4012	257	1421	487
33	3897	224	1357	430
23	2911	199	993	492
20	2665	175	971	479
35	4111	263	1533	493
21	2733	179	995	498
31	3706	230	1386	633
24	3003	190	1070	533
27	3302	222	1168	559
26	3207	187	1090	524
25	3105	234	1161	563
18	908	98	332	187
19	2602	164	919	434
22	2822	179	1026	453
32	3793	206	1336	432
29	3499	226	1260	607

## 2 (V) Email Engagement

Query:-

```
SELECT
EXTRACT(WEEK FROM 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
reengagement_emails,
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;
```





# Insights

- I. From week 17 to 30 there is up down trend of user engagement with overall upward trend and after 30 there is downward trend in both activeness of users and activity done by user
- II. On the above metric we can say there is direct relation between activeness of users and activity done by user
- III. There is tremendous growth on after week 17 and then have stable up & down trend while activeness of users has up and down trend
- IV. Till the week 30 user signup have positive growth afterward there is downfall while there is retention rate is good
- V. most of users using our product with macbook or laptops, and there is less no. Of users with cell phone device using our product.
- VI. Out of 100 sent emails there is only 33 emails get open and out of 33 emails of 14-15 emails getting encourage people for clicks, we have to improve our email standard.

# Results & Conclusions

- a) This project was very helpful to understand and strengthen key advance concept of sql
- b) Here i learn use of windows function, lead & lag function, diff date function, case function and application of it.
- c) It was helpful toward practicing of basic to intermediate sql quarries, and it was fun to write them, getting stuck and resolve error.



**THANK  
YOU !**