Project: Operation Analytics and Investigating Metric Spike

NAME: MUKESH CHANDRA KAMILA

Description:

In this project, I was designated as Data Analyst Lead and provided with different data sets. I was focusing on analysing the data which is provided by the company. So, the task is to derive insights and answer the questions asked by different departments. So that these insights are then used by operation team, support team and marketing team to predict the overall growth or decline of the company. It means better automation, better understanding between cross effective workflows.

In Case Study 1, there is job_data table wherever in Case Study 2 there are users, events and email_event tables.

Approach:

To execute the project, I used SQL. So, the first approach was to run SQL queries to create a database using the raw data provided. Once the database was created, I imported the raw data and run various sorting and data extracting queries to get the required insights.

Tech-Stack: PostgreSQL 6.19

Used:

PostgreSQL 6.19 was used in this project execution. The ease of access and set up with convenient user interface made it a good tool for the project.

Insights:

In this project, I learned about advanced SQL. And how to analyse the problem statement and the functions I can use in SQL to solve the problem statement and write the queries to get required output. Following are the questions that has been answered with the help of SQL.

Result:

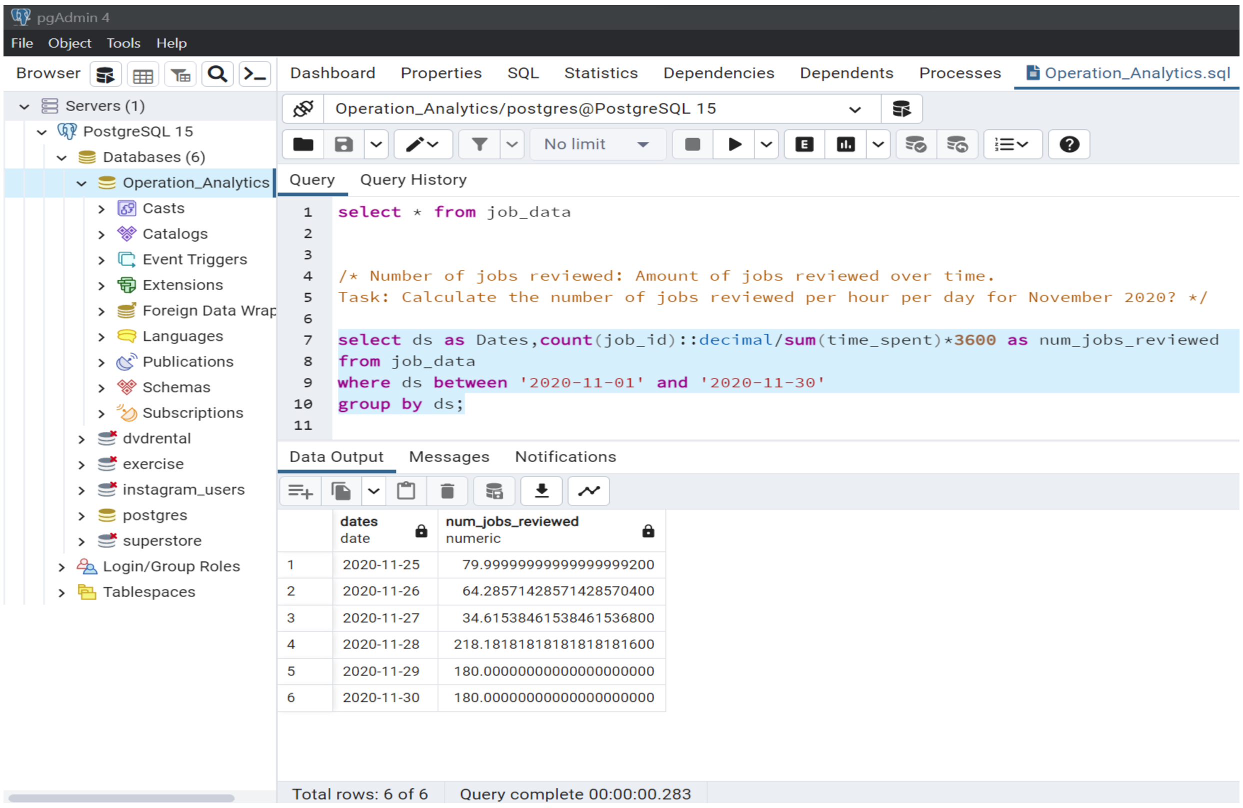
In this project, I have achieved and gain knowledge how to deal with data with help of SQL. And how to interact and run queries to get desired output from database.

Operation Analytics

Q1. Task: Calculate the number of jobs reviewed per hour per day for November 2020?

Query:

```
select ds as Dates, count(job_id)::decimal/sum(time_spent)*3600
as num_jobs_reviewed
from job_data
where ds between '2020-11-01' and '2020-11-30'
group by ds;
```



Operation_Analytics/postgres@PostgreSQL 15

```
select * from job_data


/* Number of jobs reviewed: Amount of jobs reviewed over time.
Task: Calculate the number of jobs reviewed per hour per day for November 2020? */

select ds as Dates,count(job_id)::decimal/sum(time_spent)*3600 as num_jobs_reviewed
from job_data
where ds between '2020-11-01' and '2020-11-30'
group by ds;

```

	dates date	num_jobs_reviewed numeric
1	2020-11-25	79.99999999999999999200
2	2020-11-26	64.28571428571428570400
3	2020-11-27	34.61538461538461536800
4	2020-11-28	218.18181818181818181600
5	2020-11-29	180.00000000000000000000
6	2020-11-30	180.00000000000000000000

Total rows: 6 of 6   Query complete 00:00:00.283

Q2. Task: Calculate 7 day rolling average of throughput?

For throughput, do you prefer daily metric or 7-day rolling and why?

Query:

--Weekly throughput

select round(cast(count(event)as decimal)/sum(time_spent),2)

as weekly_throughput

from job_data;

--or

select round(count(event)::decimal /sum(time_spent),2) as weekly_throughput

from job_data;

--Daily throughput

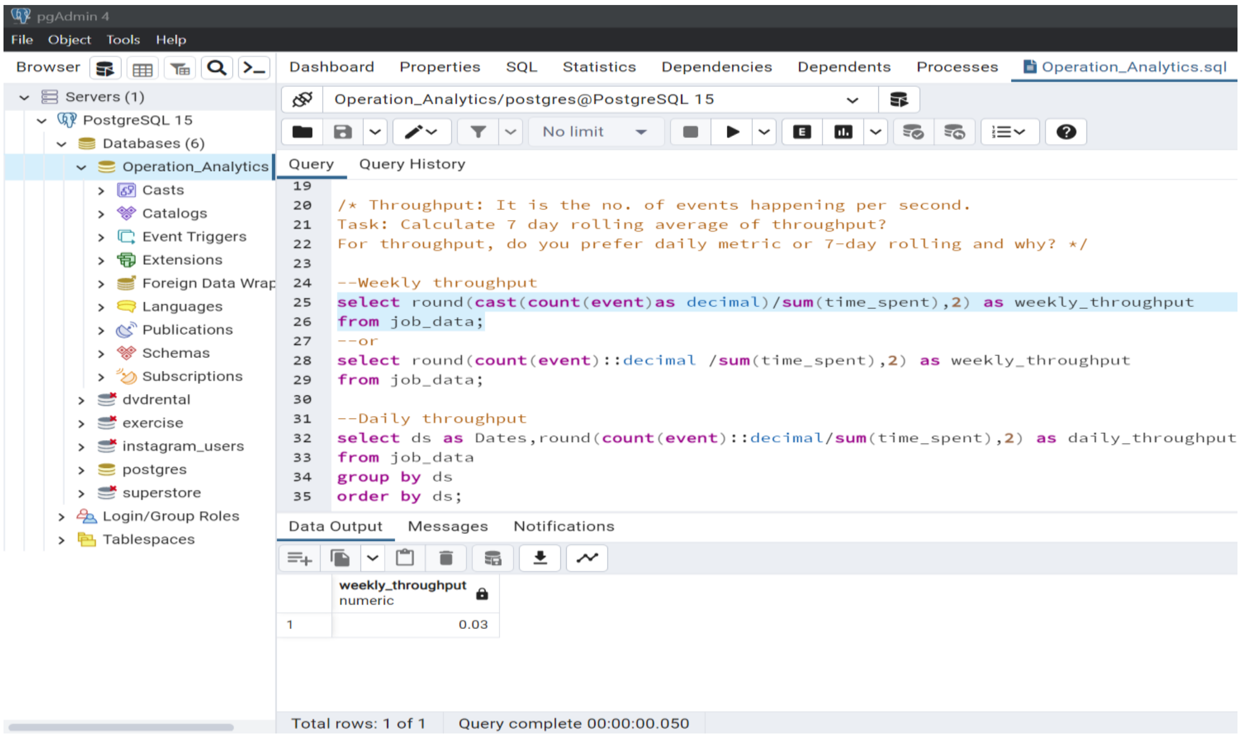
select ds as Dates,round(count(event)::decimal/sum(time_spent),2)

as daily_throughput

from job_data

group by ds

order by ds;



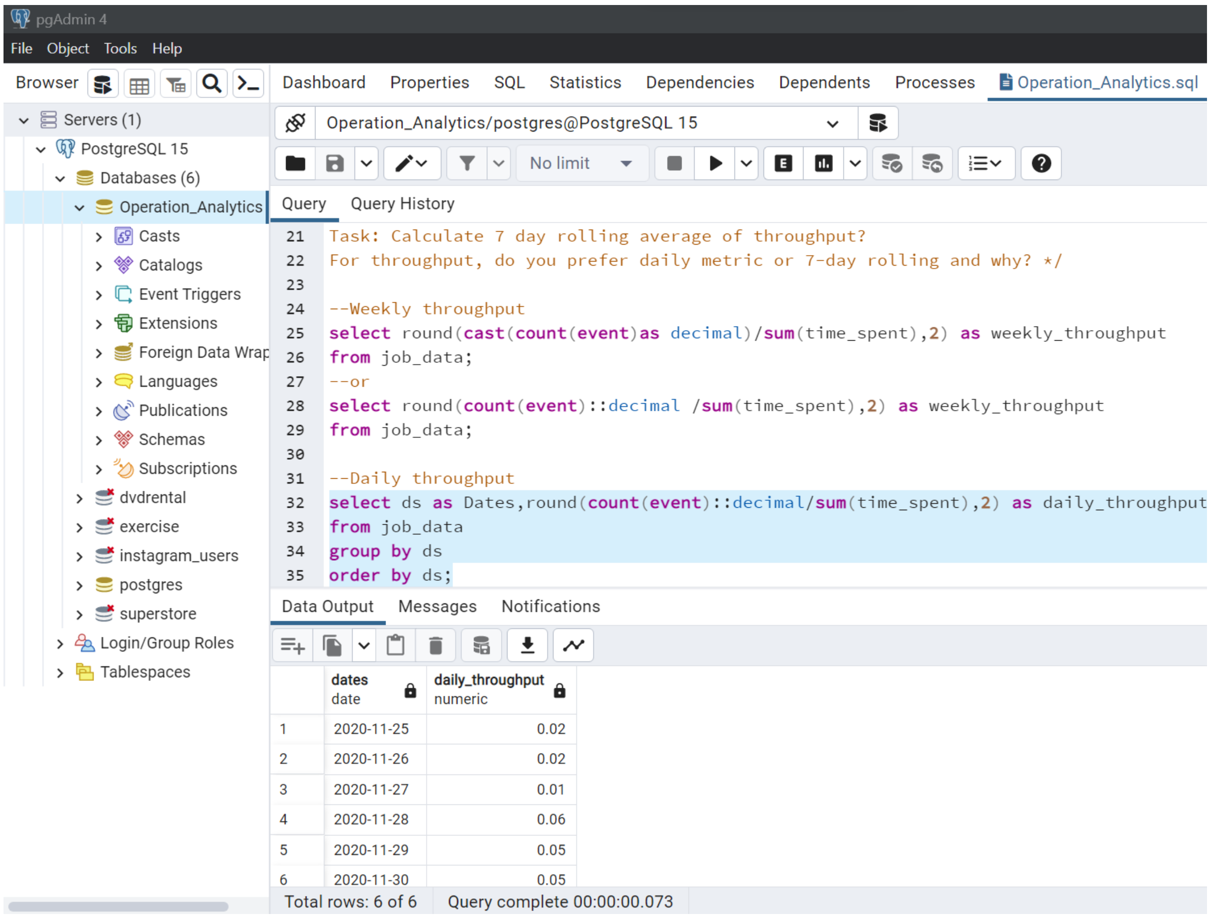
```
/* Throughput: It is the no. of events happening per second.
Task: Calculate 7 day rolling average of throughput?
For throughput, do you prefer daily metric or 7-day rolling and why? */

--Weekly throughput
select round(cast(count(event)as decimal)/sum(time_spent),2) as weekly_throughput
from job_data;
--or
select round(count(event)::decimal /sum(time_spent),2) as weekly_throughput
from job_data;

--Daily throughput
select ds as Dates,round(count(event)::decimal/sum(time_spent),2) as daily_throughput
from job_data
group by ds
order by ds;
```

```
   weekly_throughput
   numeric
1               0.03

Total rows: 1 of 1    Query complete 00:00:00.050
```



```
Task: Calculate 7 day rolling average of throughput?
For throughput, do you prefer daily metric or 7-day rolling and why? */

--Weekly throughput
select round(cast(count(event)as decimal)/sum(time_spent),2) as weekly_throughput
from job_data;
--or
select round(count(event)::decimal /sum(time_spent),2) as weekly_throughput
from job_data;

--Daily throughput
select ds as Dates,round(count(event)::decimal/sum(time_spent),2) as daily_throughput
from job_data
group by ds
order by ds;
```

	dates date	daily_throughput numeric
1	2020-11-25	0.02
2	2020-11-26	0.02
3	2020-11-27	0.01
4	2020-11-28	0.06
5	2020-11-29	0.05
6	2020-11-30	0.05

Total rows: 6 of 6   Query complete 00:00:00.073

Q3. Task: Calculate the percentage share of each language in the last 30 days?

Query:

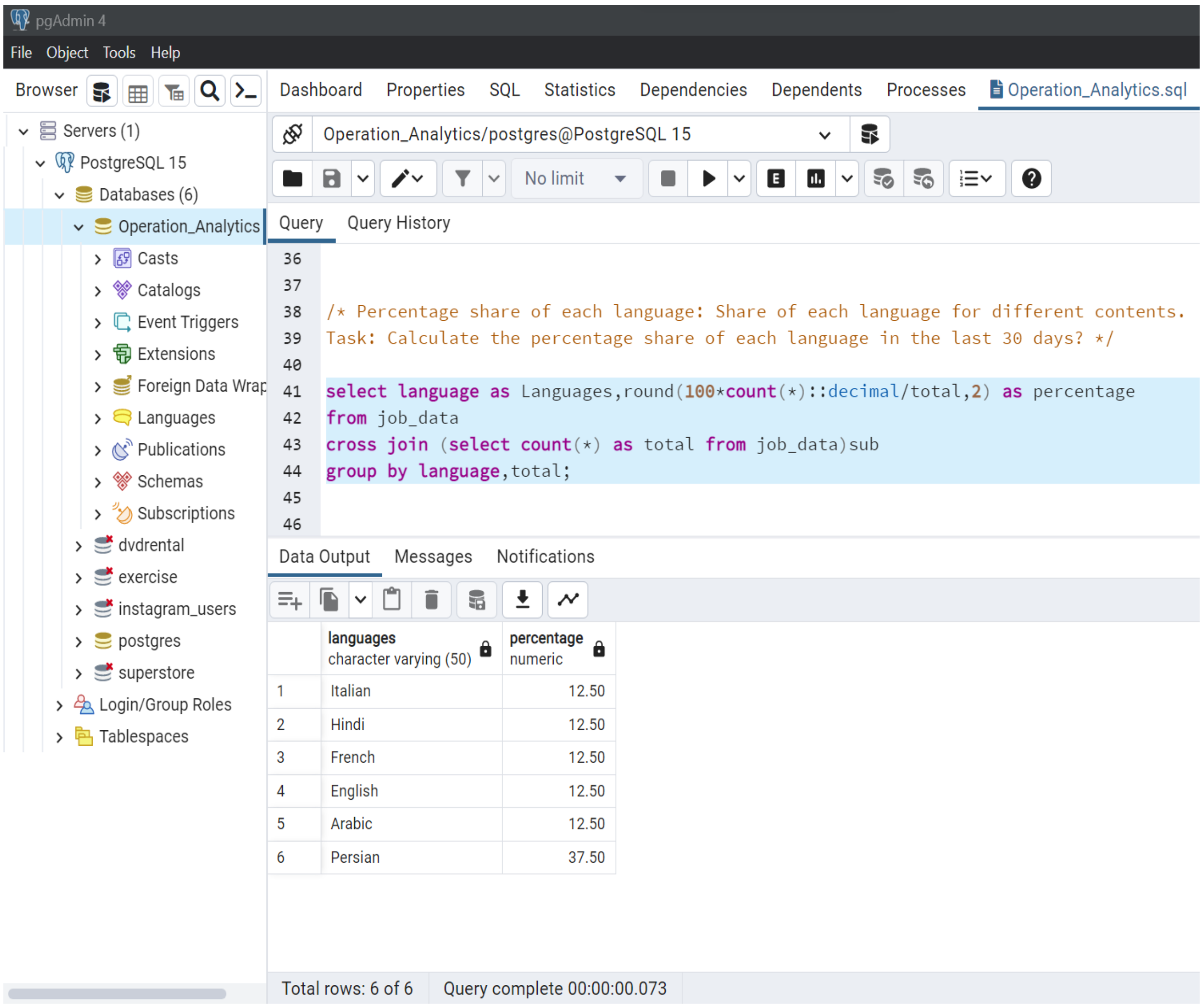
select language as Languages,round(100*count(*)::decimal/total,2)

as percentage

from job_data

cross join (select count(*) as total from job_data)sub

group by language,total;



```
/* Percentage share of each language: Share of each language for different contents.
Task: Calculate the percentage share of each language in the last 30 days? */

select language as Languages,round(100*count(*)::decimal/total,2) as percentage
from job_data
cross join (select count(*) as total from job_data)sub
group by language,total;
```

	languages character varying (50)	percentage numeric
1	Italian	12.50
2	Hindi	12.50
3	French	12.50
4	English	12.50
5	Arabic	12.50
6	Persian	37.50

Total rows: 6 of 6 Query complete 00:00:00.073

Q4. Task: display duplicates rows in the data.

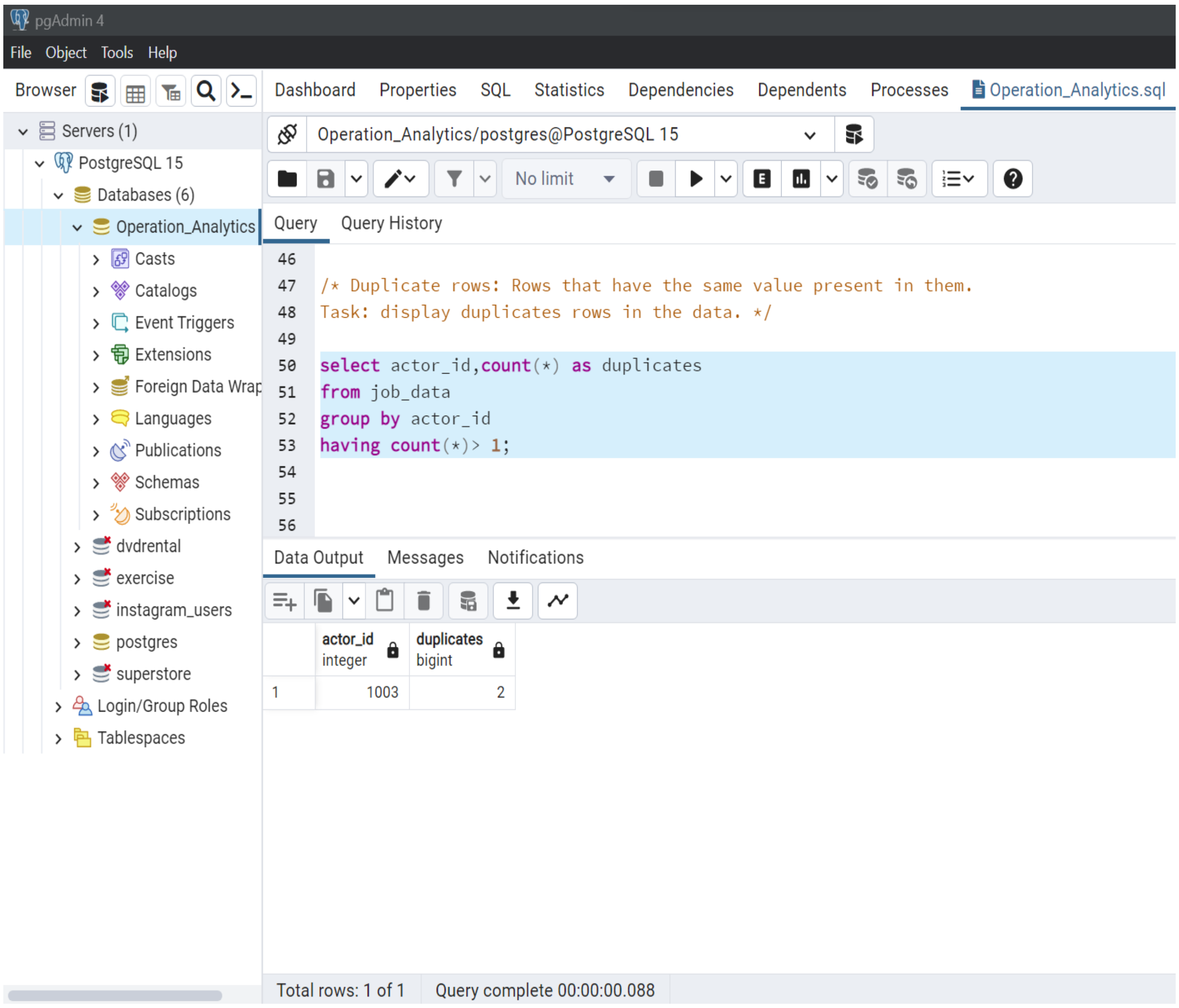
Query:

select actor_id,count(*) as duplicates

from job_data

group by actor_id

having count(*)> 1;



```
/* Duplicate rows: Rows that have the same value present in them.
Task: display duplicates rows in the data. */

select actor_id,count(*) as duplicates
from job_data
group by actor_id
having count(*)> 1;
```

	actor_id integer	duplicates bigint
1	1003	2

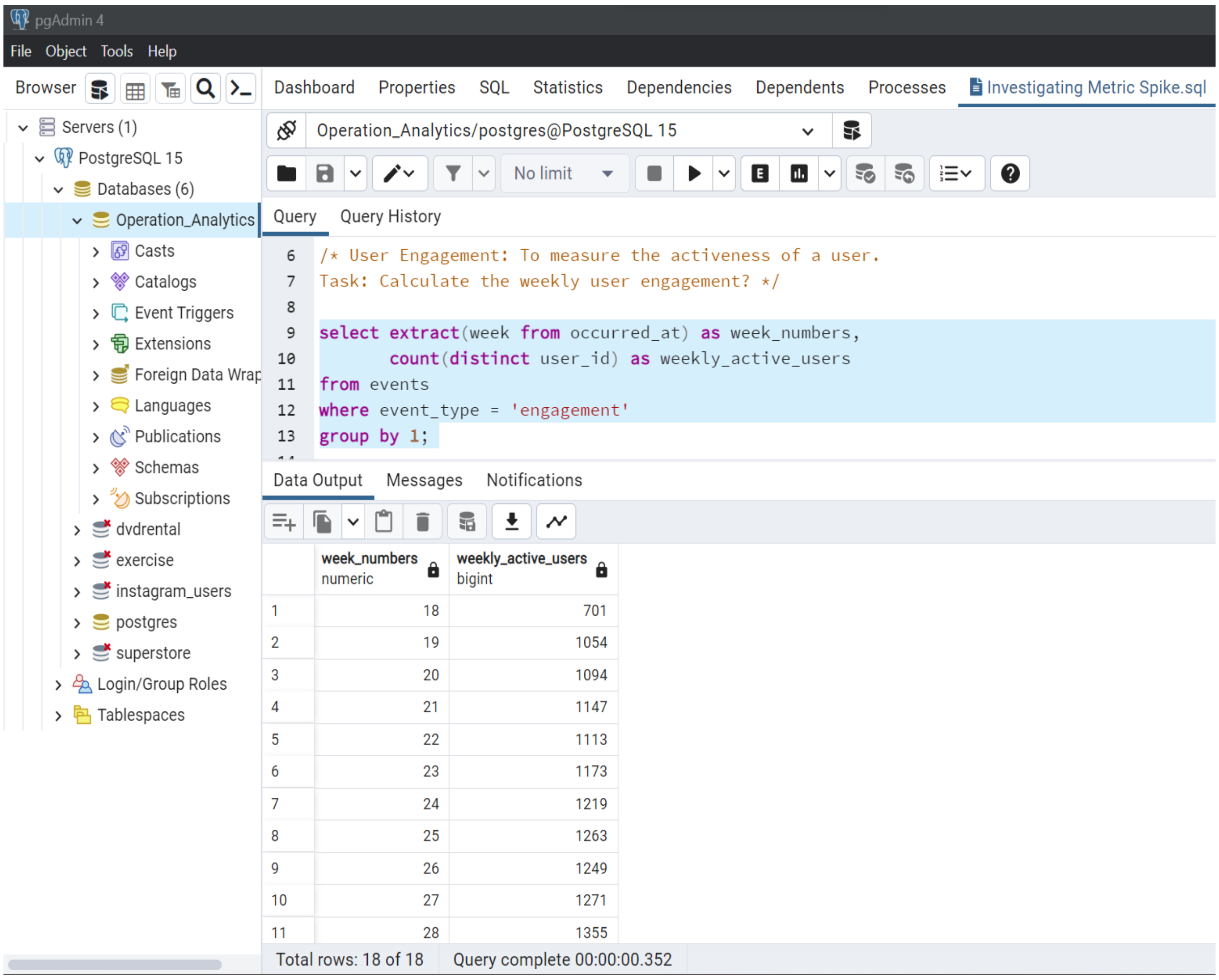
Total rows: 1 of 1   Query complete 00:00:00.088

Investigating Metric Spike

Q1. Task: Calculate the weekly 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 1;
```



```
/* User Engagement: To measure the activeness of a user.
Task: Calculate the weekly user engagement? */

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 1;
```

	week_numbers numeric	weekly_active_users bigint
1	18	701
2	19	1054
3	20	1094
4	21	1147
5	22	1113
6	23	1173
7	24	1219
8	25	1263
9	26	1249
10	27	1271
11	28	1355

Total rows: 18 of 18 Query complete 00:00:00.352

Q2. Task: Calculate the user growth for product?

Query:

select year, num_week, num_active_users,

sum(num_active_users) over(order by year, num_week rows between unbounded preceding and current row)

as cumm_active_users

from

(select

extract(year from a.activated_at) as year,

extract(week from a.activated_at)as num_week,

count(distinct user_id) as num_active_users

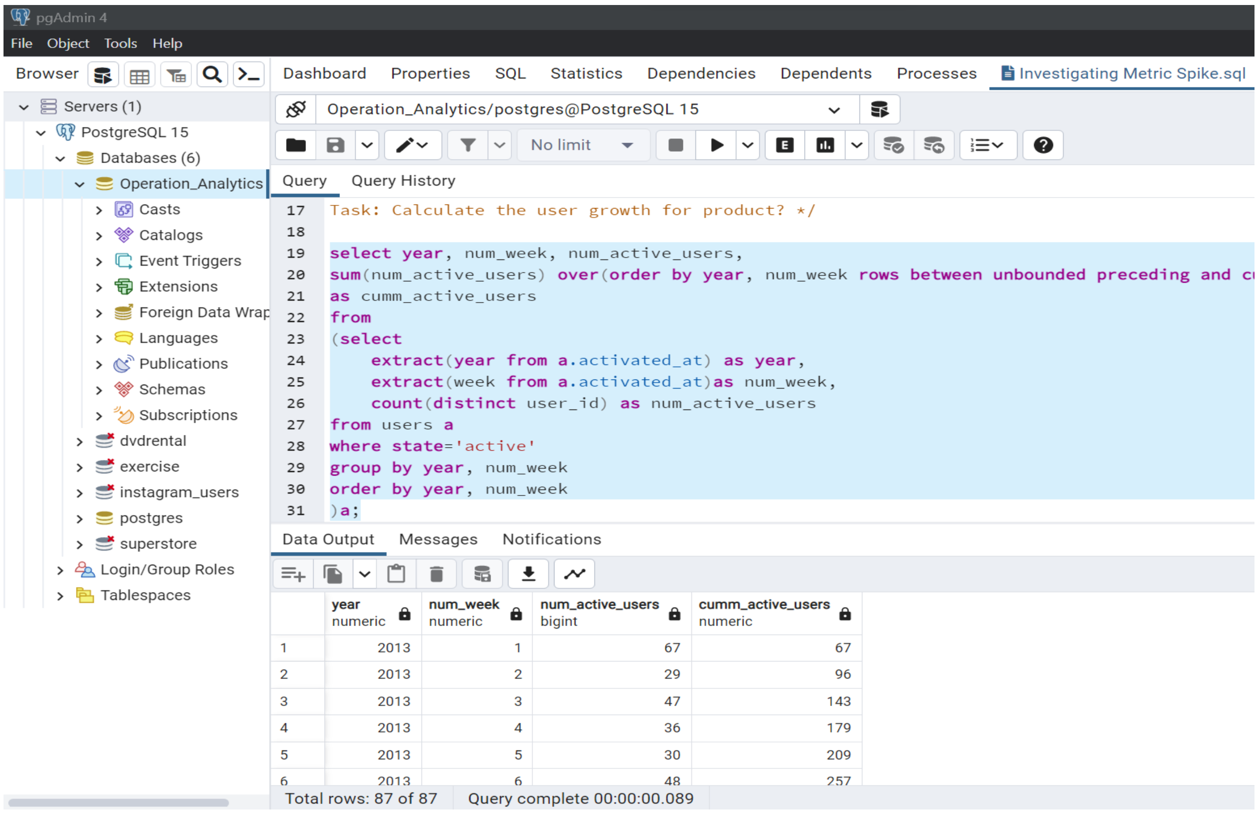
from users a

where state='active'

group by year, num_week

order by year, num_week

)a;



```
Task: Calculate the user growth for product? */

select year, num_week, num_active_users,
sum(num_active_users) over(order by year, num_week rows between unbounded preceding and cu
as cumm_active_users
from
(select
    extract(year from a.activated_at) as year,
    extract(week from a.activated_at)as num_week,
    count(distinct user_id) as num_active_users
from users a
where state='active'
group by year, num_week
order by year, num_week
)a;
```

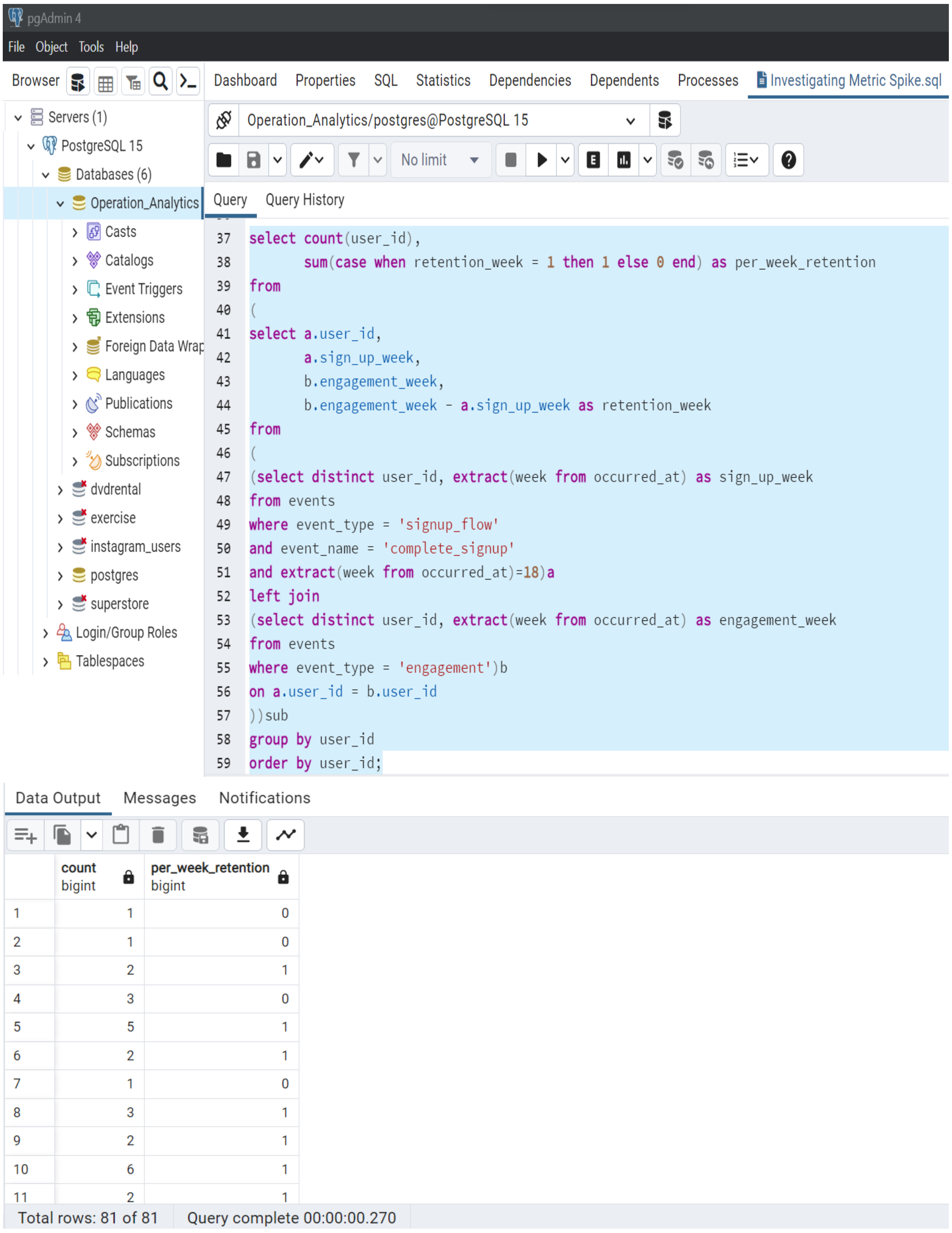
	year numeric	num_week numeric	num_active_users bigint	cumm_active_users numeric
1	2013	1	67	67
2	2013	2	29	96
3	2013	3	47	143
4	2013	4	36	179
5	2013	5	30	209
6	2013	6	48	257

Total rows: 87 of 87 Query complete 00:00:00.089

Q3. Task: Calculate the weekly retention of users-sign up cohort?

Query:

```
select count(user_id),
       sum(case when retention_week = 1 then 1 else 0 end) as per_week_retention
from
(
select a.user_id,
       a.sign_up_week,
       b.engagement_week,
       b.engagement_week - a.sign_up_week as retention_week
from
(
(select distinct user_id, extract(week from occurred_at) as sign_up_week
from events
where event_type = 'signup_flow'
and event_name = 'complete_signup'
and extract(week from occurred_at)=18)a
left join
(select distinct user_id, extract(week from occurred_at) as engagement_week
from events
where event_type = 'engagement')b
on a.user_id = b.user_id
))sub
group by user_id
order by user_id;
```



```sql
select count(user_id),
       sum(case when retention_week = 1 then 1 else 0 end) as per_week_retention
from
(
select a.user_id,
       a.sign_up_week,
       b.engagement_week,
       b.engagement_week - a.sign_up_week as retention_week
from
(
(select distinct user_id, extract(week from occurred_at) as sign_up_week
from events
where event_type = 'signup_flow'
and event_name = 'complete_signup'
and extract(week from occurred_at)=18)a
left join
(select distinct user_id, extract(week from occurred_at) as engagement_week
from events
where event_type = 'engagement')b
on a.user_id = b.user_id
))sub
group by user_id
order by user_id;
```

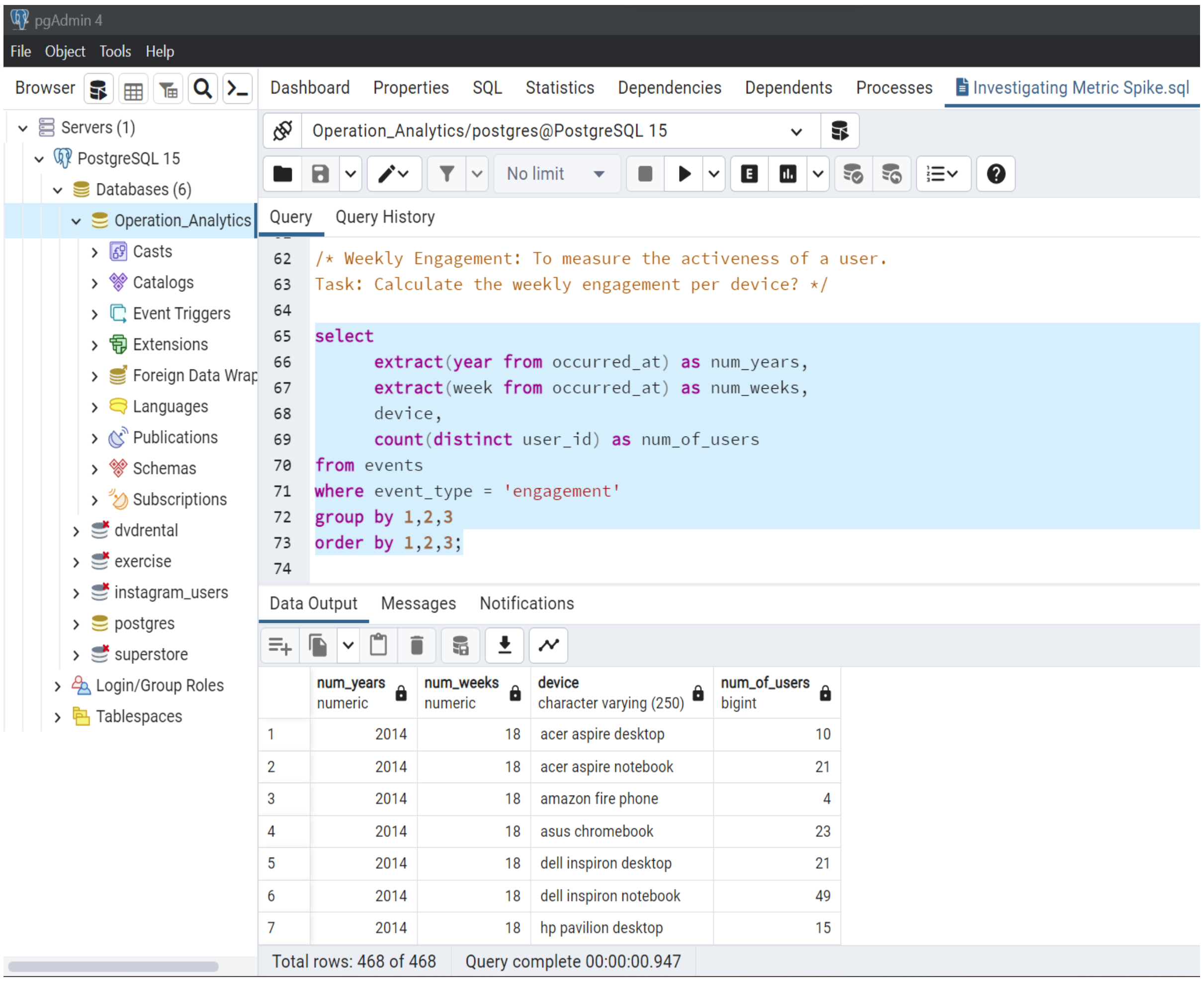
	count bigint	per_week_retention bigint
1	1	0
2	1	0
3	2	1
4	3	0
5	5	1
6	2	1
7	1	0
8	3	1
9	2	1
10	6	1
11	2	1

Total rows: 81 of 81   Query complete 00:00:00.270

Q4. Task: Calculate the weekly engagement per device?

Query:

```
select
        extract(year from occurred_at) as num_years,
        extract(week from occurred_at) as num_weeks,
        device,
        count(distinct user_id) as num_of_users
from events
where event_type = 'engagement'
group by 1,2,3
order by1,2,3;
```



```
/* Weekly Engagement: To measure the activeness of a user.
Task: Calculate the weekly engagement per device? */

select
      extract(year from occurred_at) as num_years,
      extract(week from occurred_at) as num_weeks,
      device,
      count(distinct user_id) as num_of_users
from events
where event_type = 'engagement'
group by 1,2,3
order by 1,2,3;
```

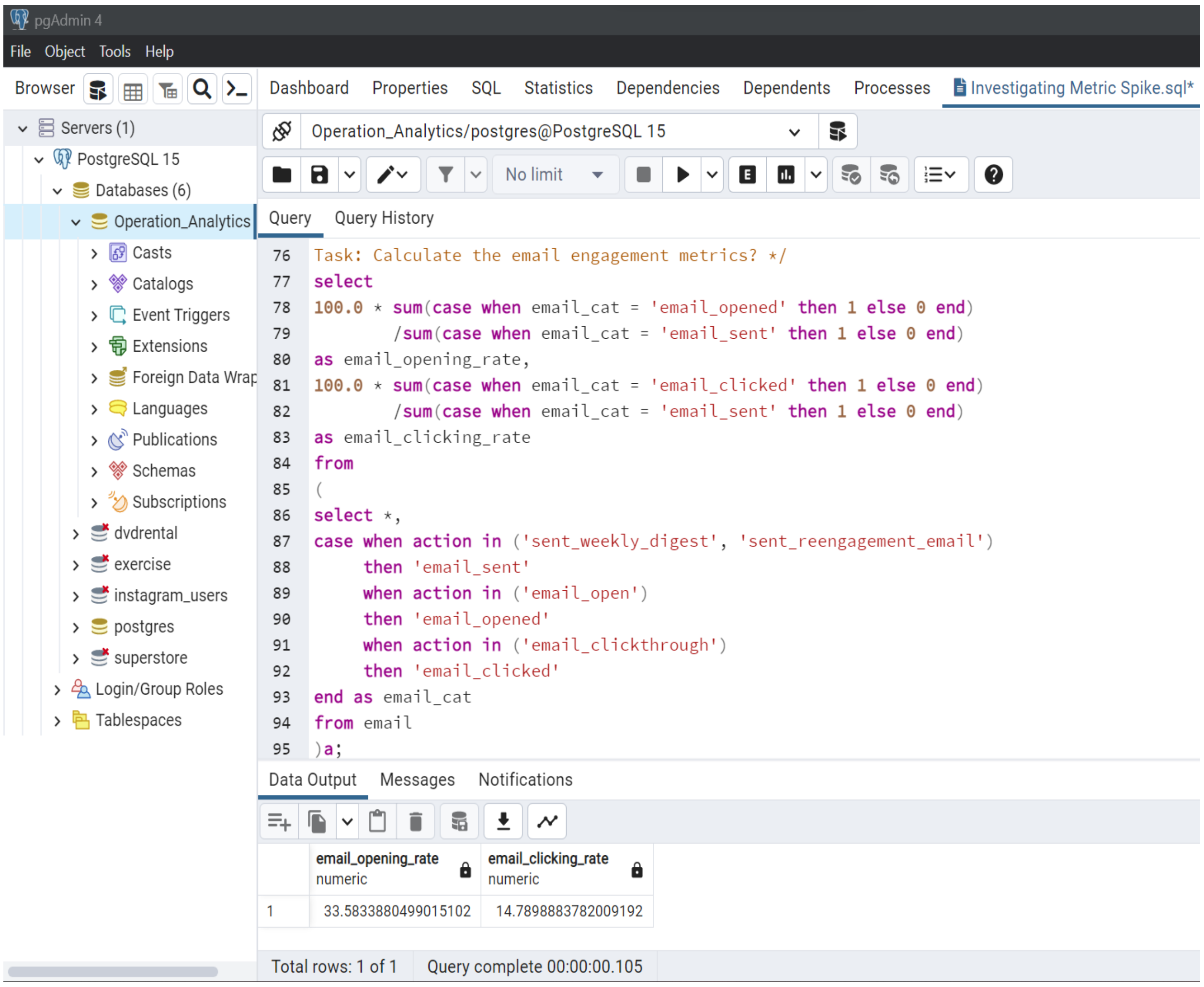
	num_years numeric	num_weeks numeric	device character varying (250)	num_of_users bigint
1	2014	18	acer aspire desktop	10
2	2014	18	acer aspire notebook	21
3	2014	18	amazon fire phone	4
4	2014	18	asus chromebook	23
5	2014	18	dell inspiron desktop	21
6	2014	18	dell inspiron notebook	49
7	2014	18	hp pavilion desktop	15

Total rows: 468 of 468 Query complete 00:00:00.947

Q5. Task: Calculate the email engagement metrics?

Query:

```
select
100.0 * sum(case when email_cat = 'email_opened' then 1 else 0 end)
      /sum(case when email_cat = 'email_sent' then 1 else 0 end)
as email_opening_rate,
100.0 * sum(case when email_cat = 'email_clicked' then 1 else 0 end)
      /sum(case when email_cat = 'email_sent' then 1 else 0 end)
as email_clicking_rate
from
(
select *,
case when action in ('sent_weekly_digest', 'sent_reengagement_email')
    then 'email_sent'
    when action in ('email_open')
    then 'email_opened'
    when action in ('email_clickthrough')
    then 'email_clicked'
end as email_cat
from email
)a;
```



```sql
Task: Calculate the email engagement metrics? */
select
100.0 * sum(case when email_cat = 'email_opened' then 1 else 0 end)
        /sum(case when email_cat = 'email_sent' then 1 else 0 end)
as email_opening_rate,
100.0 * sum(case when email_cat = 'email_clicked' then 1 else 0 end)
        /sum(case when email_cat = 'email_sent' then 1 else 0 end)
as email_clicking_rate
from
(
select *,
case when action in ('sent_weekly_digest', 'sent_reengagement_email')
     then 'email_sent'
     when action in ('email_open')
     then 'email_opened'
     when action in ('email_clickthrough')
     then 'email_clicked'
end as email_cat
from email
)a;
```

	email_opening_rate numeric	email_clicking_rate numeric
1	33.5833880499015102	14.7898883782009192

Total rows: 1 of 1   Query complete 00:00:00.105