**Operation Analytics and Investigating Metric Spike**

**DESCRIPTION:**

Operation Analytics is the analysis done for the complete end-to-end operations of a company. With the help of this analysis, the company finds the areas in which it must do something to improve. In this project, I will work closely with the ops team, support team, marketing team, etc., and help them derive insights from the data they collect. As one of the most important parts of a company, this analysis is further used to predict the overall growth or decline of a company’s fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.

Investigating Metric spikes is also an important part of operation analytics as Data Analysts. I must understand or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily, so it is very important to investigate metric spikes.

**Project Approach Used:**

According to the project’s scenario, I am working for a company like Microsoft, designated as Data Analyst Lead, and provided with different data sets and tables from which I must derive certain insights and answer the questions asked by different departments. This project is completely based on Structured Query Language (SQL). To create a complete database containing different tables, SQL queries were provided. To answer all the questions asked, I used SQL Basic to Advanced functions that generated useful insights. This report gives a better understanding of that in the ‘insight.’ segment.

**Tech-Stack Used:**

* MySQL Workbench with the latest released Version 8.0 CE (Community Edition) by Oracle Corporation – For Importing Database, Running SQL Queries to get insights
* Excel 2016 by Microsoft Corporation – For extracting & manipulating data
* Word 2016 by Microsoft Corporation – For creating a project report

I used it because MySQL Workbench is a visual database design tool that combines SQL development, administration, database design, creation, and maintenance into a single development environment.

**Project Insights**

1. **Case Study 1 (Job Data)**
2. **Number of jobs reviewed:**

Here, I need to calculate the number of jobs reviewed per hour per day for November 2020 from the given database.

For fulfilling the required query, I selected ds and used the COUNT function on job\_id and named that column jobs\_per\_day. Then I used the SUM function on the total time spent (I assumed that the time spent was in minutes) and divided it by 60 to get time in hours and used table opsanalytics.jobs\_data (Table I created in MYSQL using the given dataset), and lastly, used the WHERE function to narrow down the search results for the month of November.

**SQL Statement:**

SELECT

ds,

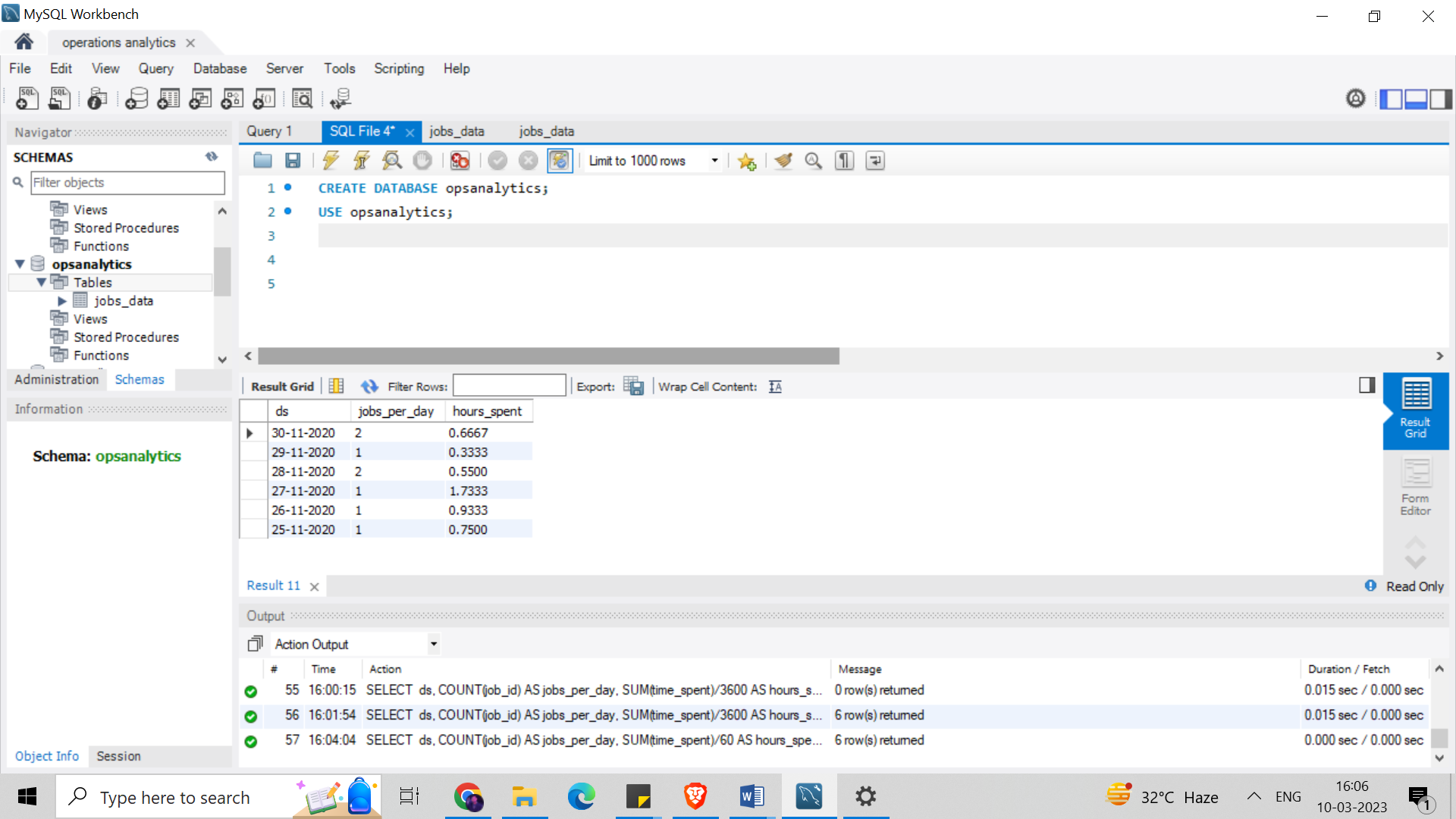
COUNT(job\_id) AS jobs\_per\_day,

SUM(time\_spent)/60 AS hours\_spent

FROM opsanalytics.jobs\_data

WHERE ds >='01-11-2020' AND ds <='30-11-2020' GROUP BY ds;

**Result:**



**Conclusion:**

Following are the number of jobs reviewed per hour per day for the month of November 2020.

1. **Throughput or the no. of events happening per second.**

Here I need to calculate the 7-day rolling average of throughput, and for throughput, do I prefer daily metric or 7-day rolling and why?

We know that a rolling average is a calculation that lets us analyze data points by creating a series of averages based on different subsets of a data set called a moving average or running average.

A 7-day moving average is a short-term trend indicator. It is simply the average closing values of the last seven days. I prefer a 7-day rolling average because it is reliable, and it is a simple indicator of changing the price/production of services or goods.

**SQL Statement:**

SELECT x.\*,

AVG(dailyusage) over (partition by job\_id order by job\_id, ds rows

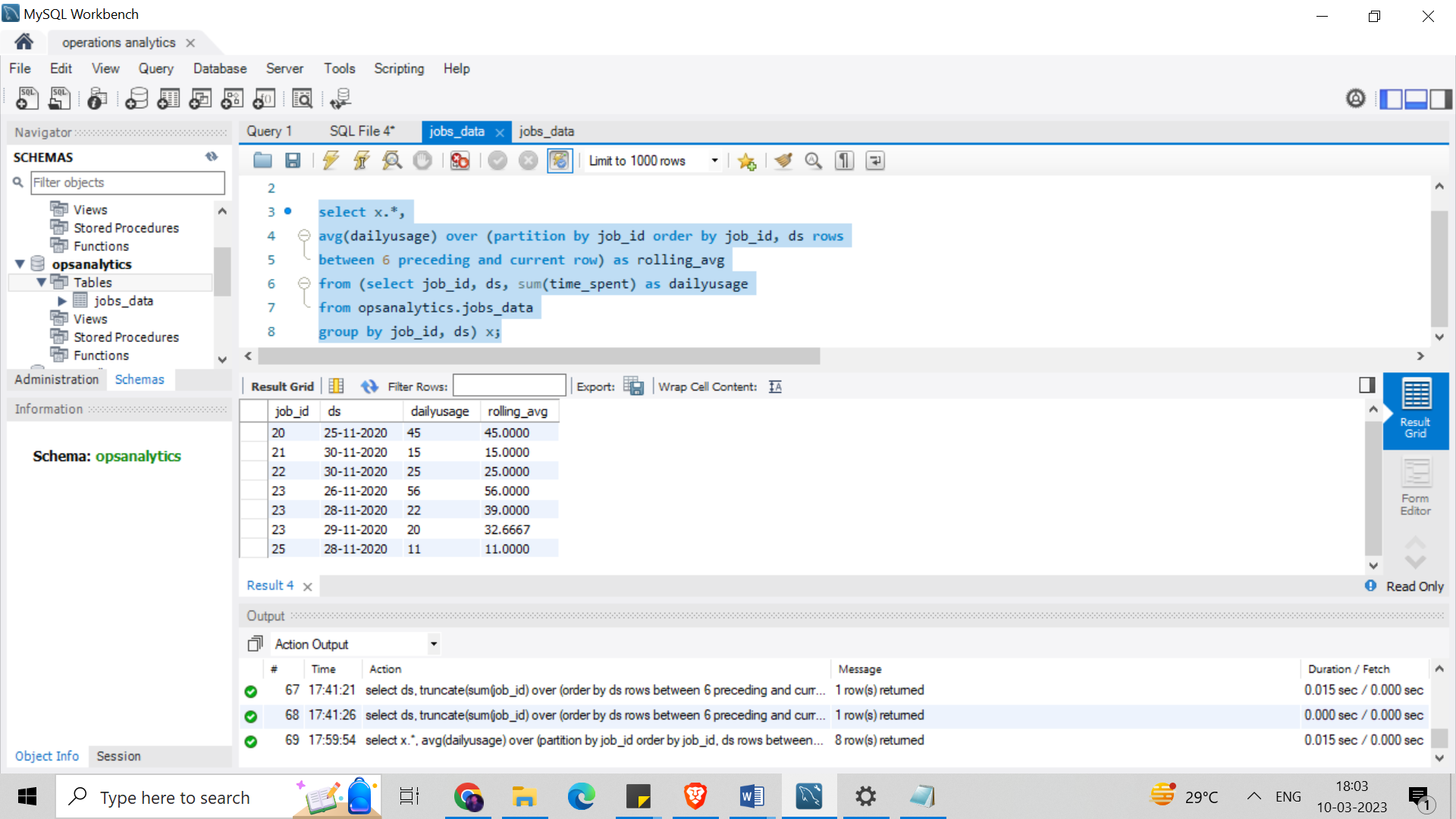
between 6 preceding and current row) AS rolling\_avg

FROM (select job\_id, ds, sum(time\_spent) as dailyusage

FROM opsanalytics.jobs\_data

GROUP BY job\_id, ds) x;

**Result:**



**Conclusion:**

A 7-day rolling average is what I prefer because rolling averages can be used to uncover long-term trends that are occasionally masked by volatility.

1. **Percentage share of each language:**Share of each language for different contents.

Here I need to calculate the percentage share of each language in the last 30 days.

**SQL Statement:**

SELECT ds, language, event,

COUNT(\*) as lang\_usage,

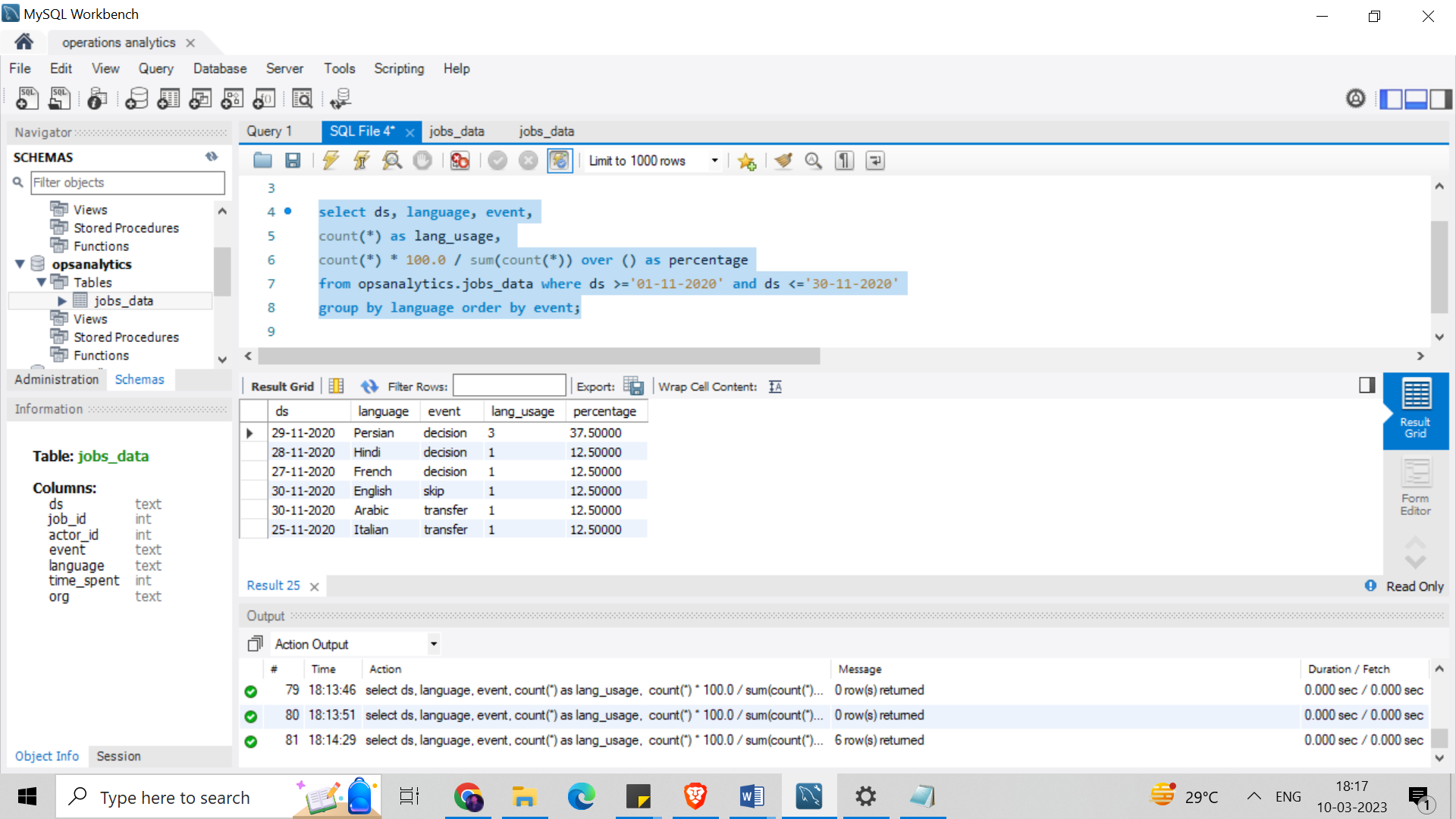
COUNT(\*) \* 100.0 / sum(COUNT(\*)) OVER () AS percentage

FROM opsanalytics.jobs\_data

WHERE ds >='01-11-2020' and ds <='30-11-2020'

GROUP BY language ORDER BY event;

**Result:**



**Conclusion:**

Following is the percentage share of each language for events, where we can conclude Persian is the most used language, and the rest, like Hindi, French, etc., have an equal share with a percentage share of 37.5% and 12.5%, respectively.

1. **Duplicate rows:** Rows that have the same value present in them.

Here I need to display duplicates from the table.

**SQL Statement:**

SELECT job\_id, actor\_id,

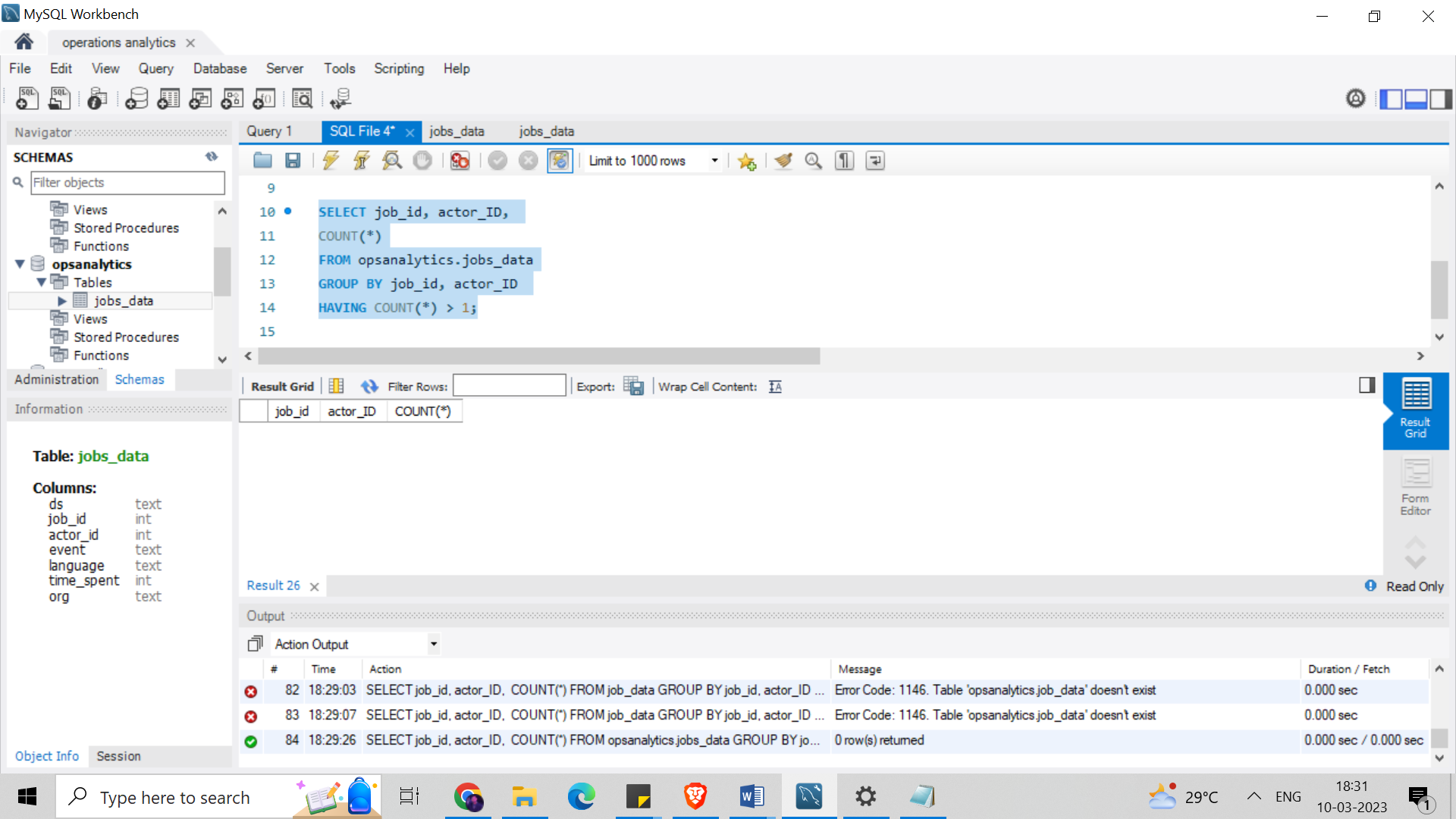
COUNT(\*)

FROM opsanalytics.jobs\_data

GROUP BY job\_id, actor\_id

HAVING COUNT(\*) > 1;

**Result:**



**Conclusion:**

Here all the rows have unique values. No rows were found that have the same value present in them.

Hence, there are no duplicate values to be displayed.

1. **Case Study 2 (Investigating metric spike)**
2. **User Engagement:**To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Here I need to calculate the weekly user engagement.

**SQL statement:**

SELECT

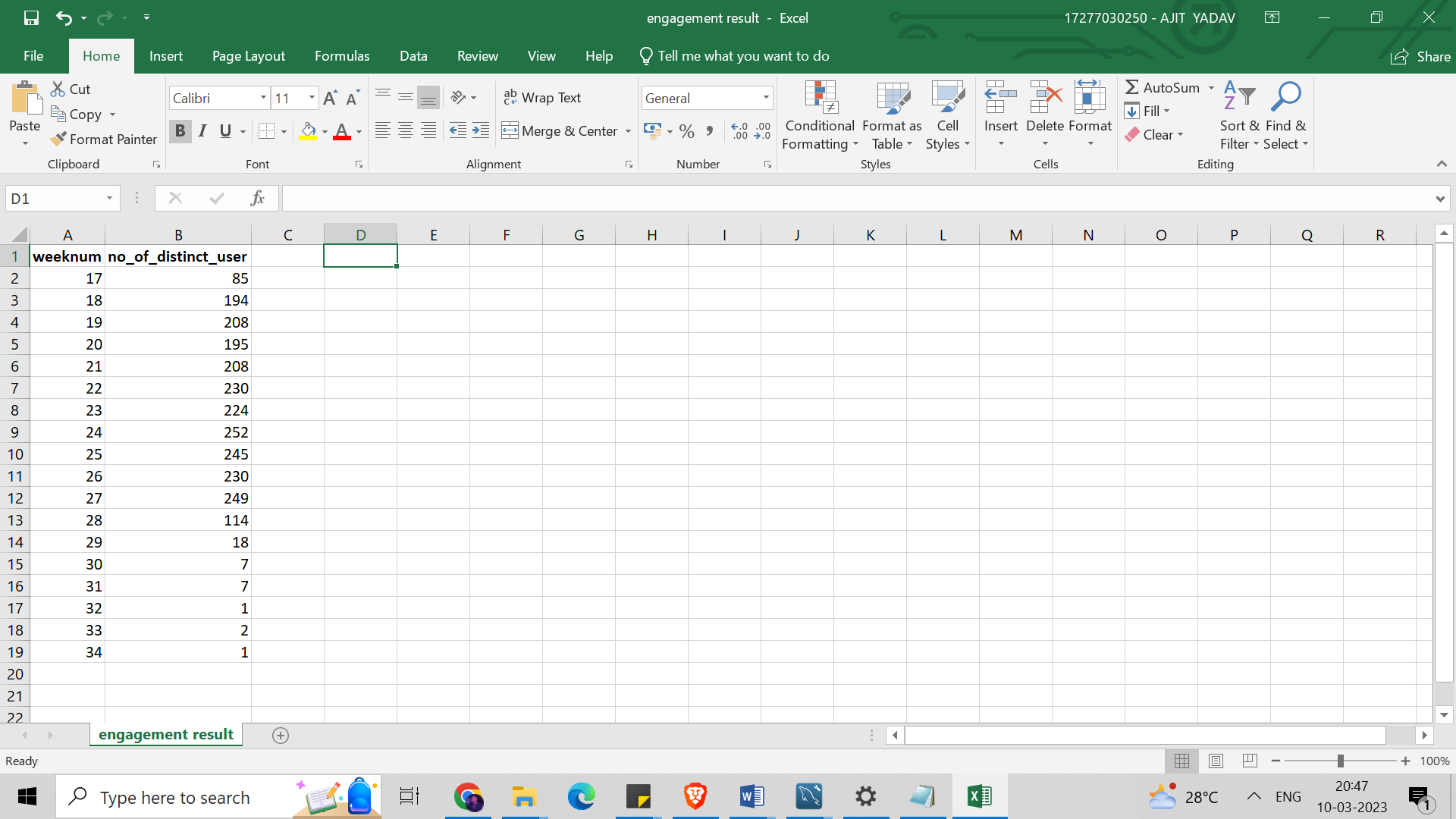
EXTRACT(WEEK FROM occurred\_at) AS weeknum,

COUNT(DISTINCT user\_id) AS no\_of\_distinct\_user

FROM `opsanalytics`.`table-2 events`

GROUP BY weeknum

**Conclusion: Result:**



Following is the weakly user engagement.

1. **User Growth**: Amount of users growing over time for a product.

Here I need to calculate the user growth for the product.

**SQL statement:**

SELECT

activated\_at, weeknum, num\_active\_user,

SUM(num\_active\_user),

OVER(ORDER BY activated\_at, 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\_user

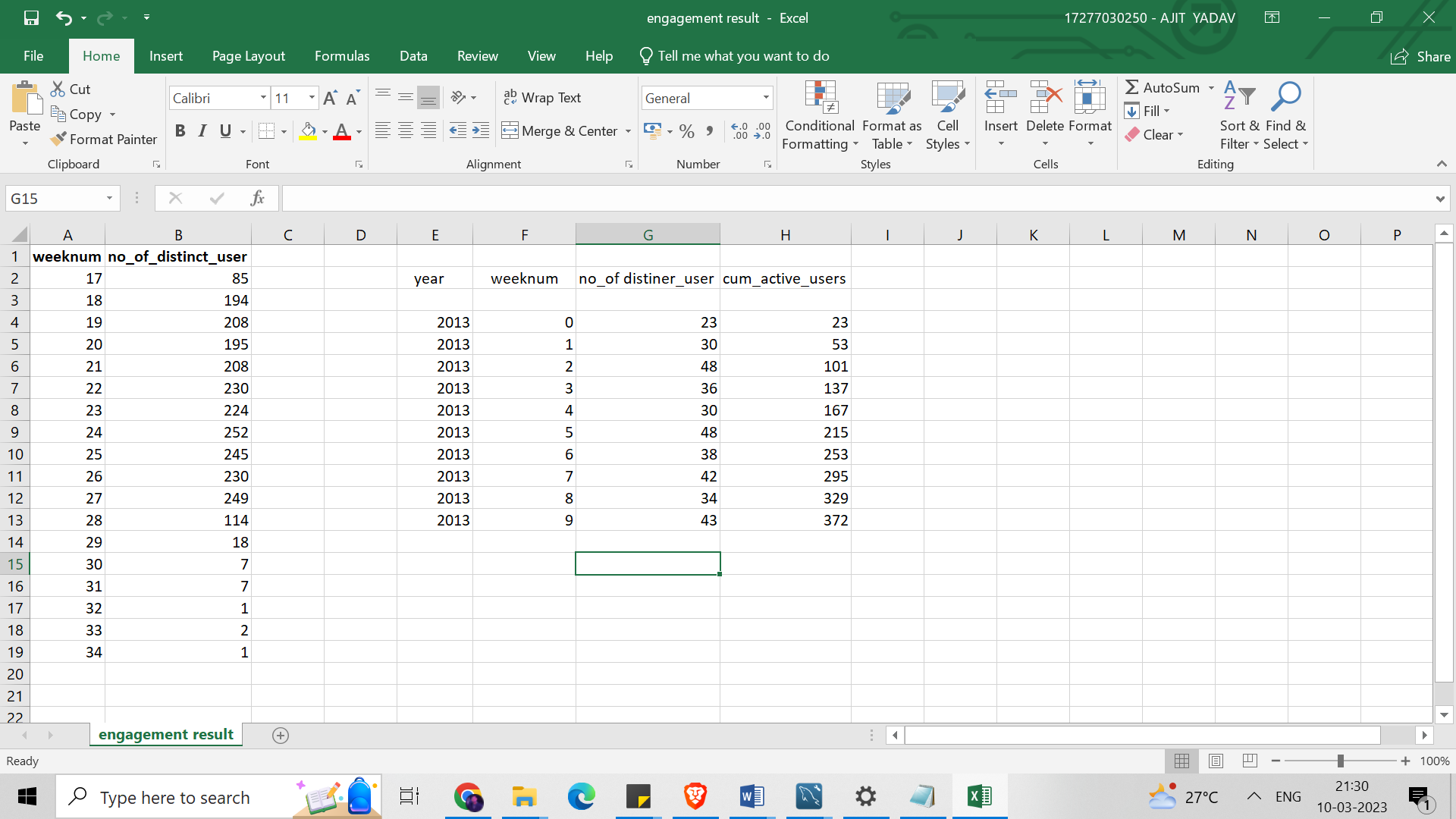
FROM `opsanalytics`.`table-1 users` a

WHERE state='active'

GROUP BY year, weeknum

ORDER BY year, weeknum) a;

**Result:**



**Conclusion:** Following is the output for the query having the amount of users growing over time for a product.

1. **Weekly Retention:**

Here I need to find users getting retained weekly after signing up for a product.

**SQL statement:**

SELECT week (occurred\_at) AS week,

COUNT(CASE WHEN e.event\_type = 'engagement' THEN e.user\_id

ELSE NULL END) AS engagement,

COUNT(CASE WHEN e.event\_type = 'signup\_flow' THEN e.user\_id

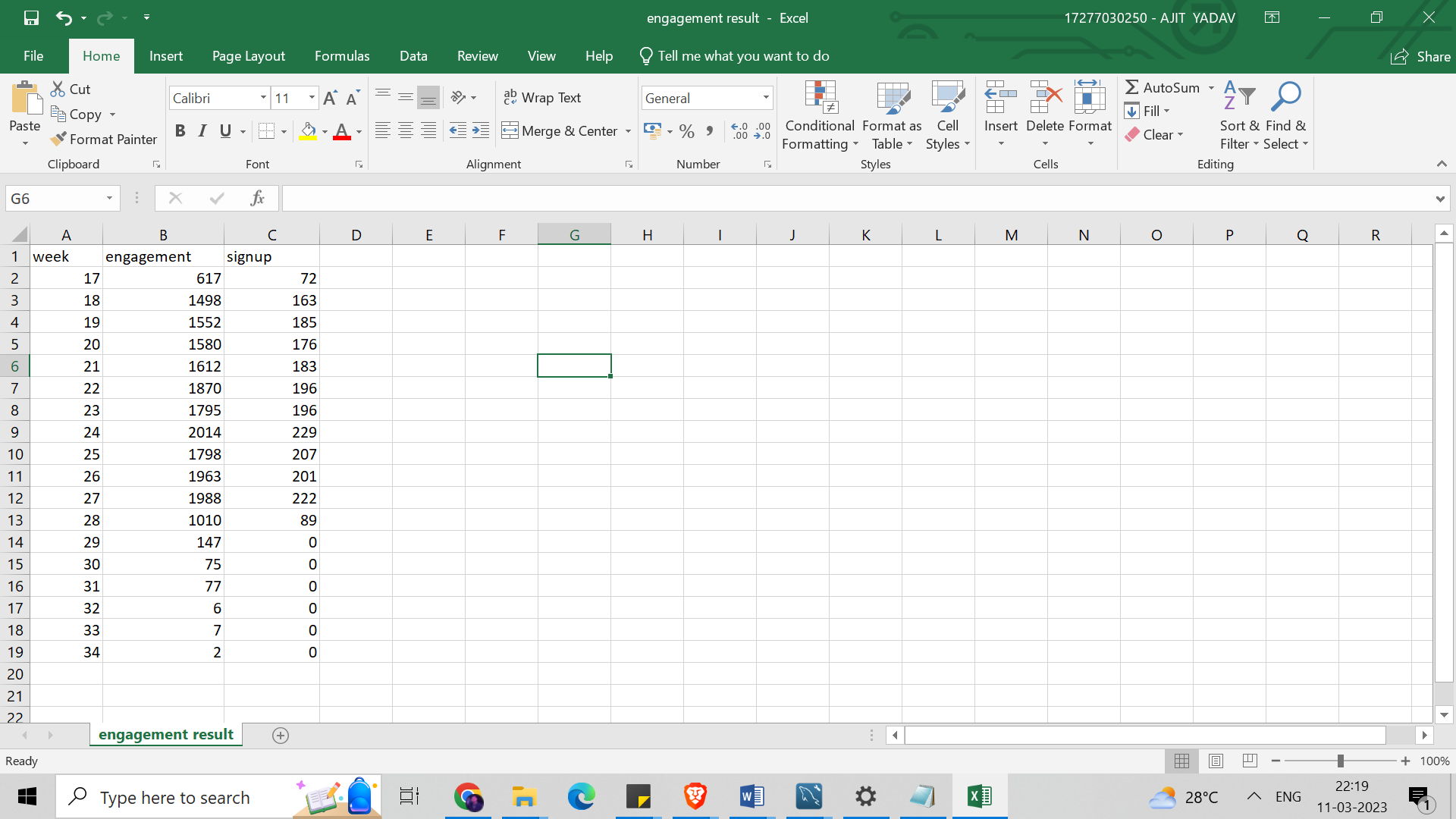
ELSE NULL END) AS signup

FROM `opsanalytics`.`table-2 events` e

GROUP BY 1

ORDER BY 1

**Conclusion: Result:**

Following mentioned above are the

metrics of the users getting retained

weekly after signing up for a product,

and the engagements related to it.

1. **Weekly Engagement:**To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

Here I need to calculate the weekly engagement per device.

**SQL Statement:**

SELECT week (occurred\_at) AS week,

COUNT(DISTINCT e.user\_id) AS weekly\_users,

COUNT(DISTINCT CASE WHEN e. device IN ('macbook pro', 'aceraspire notebook',

'acer aspire desktop','lenovo thinkpad', 'mac mini', 'dell inspirondesktop',

'dell inspiron notebook','windows surface','macbook air','asus chromebook','hppavilion desktop')

THEN e.user\_id ELSE NULL END) AS computer,

COUNT(DISTINCT CASE WHEN e. device IN ('iphone 5s',

'nokia lumia635','amazon fire phone','iphone 4s','htc one','iphone 5','samsung galaxy s4')

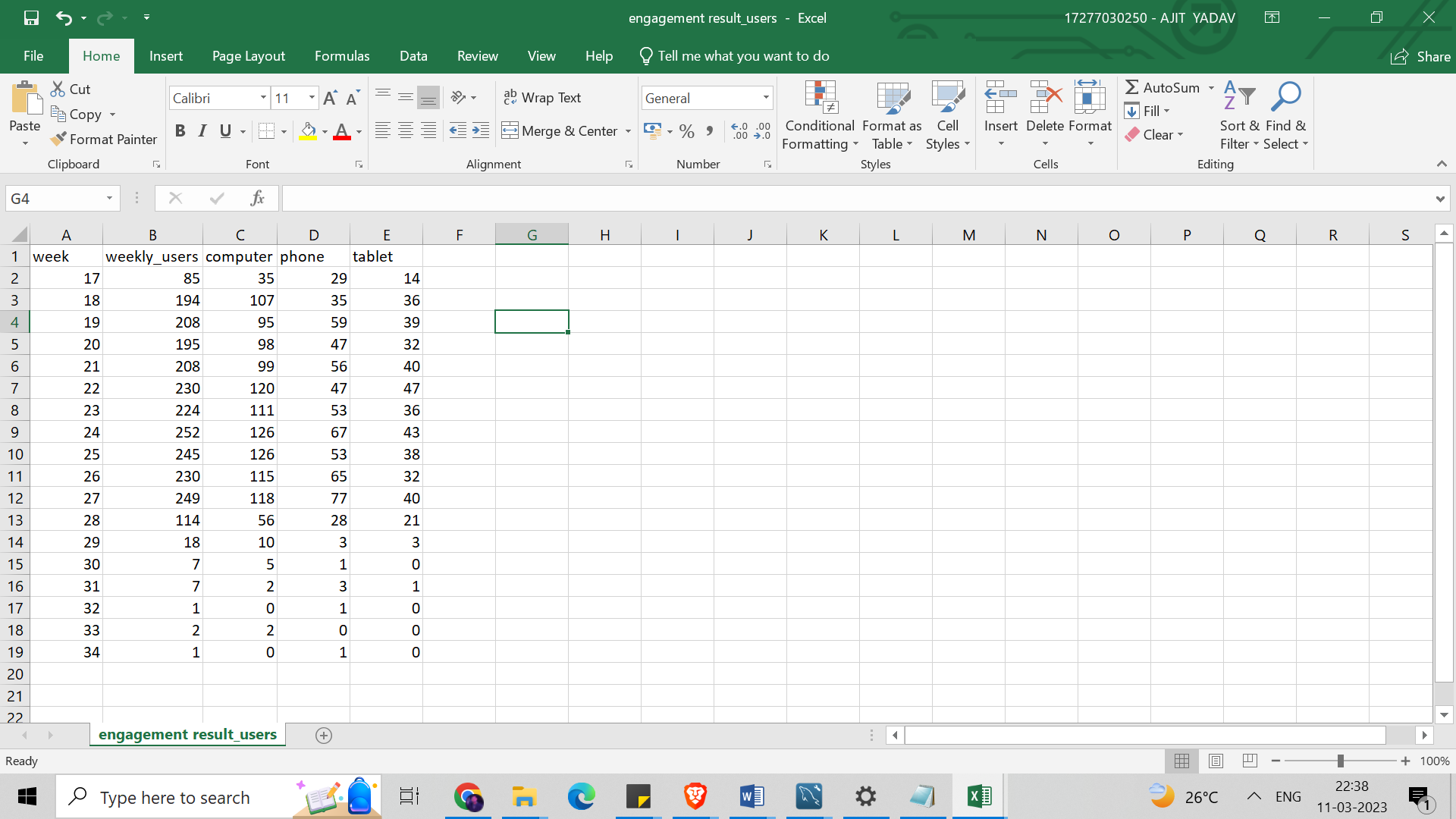
THEN e.user\_id ELSE NULL END) AS phone,

COUNT(DISTINCT CASE WHEN e. device IN ('kindle fire',

'samsunggalaxy note','ipad mini','nexus 7','nexus 10','samsumg galaxy tablet','nexus 5', 'ipad air')

THEN e.user\_id ELSE NULL END) AS tablet

FROM `opsanalytics`.`table-2 events` e **Result:**

WHERE e. event\_type = 'engagement'

AND e. event\_name = 'login'

GROUP BY 1

ORDER BY 1

**Conclusion:**

Following is the weakly engagement per device, where users are using devices like computers, phone and tablets.

1. **Email Engagement:**Users engaging with the email service.

Here I need to calculate the email engagement metrics.

**SQL statement:**

SELECT

EXTRACT(WEEK FROM occurred\_at) AS weeknum,

COUNT(DISTINCT e.user\_id) AS weekly\_users,

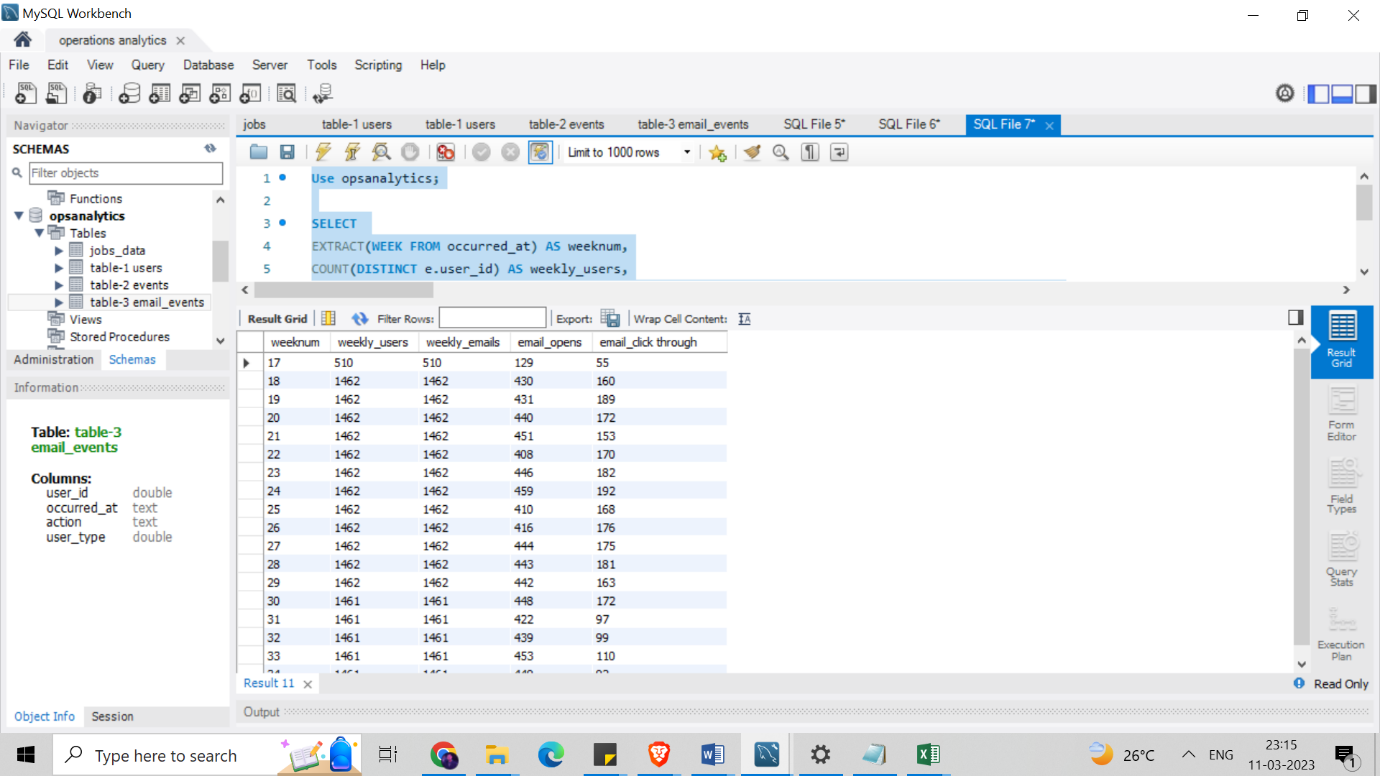
COUNT(CASE WHEN e.action = 'sent\_weekly\_digest' THEN e.user\_id ELSE NULL END) AS 'weekly\_emails',

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\_click through'

FROM `opsanalytics`.`table-3 email\_events` e

GROUP BY 1

**Conclusion: Result:**

Following are the email metrics, where I found 18 rows of the metrics like weekly users, weekly emails, email opens, email clickthrough etc.

**Result:**

I accomplished a great deal while working on the Operation Analytics and Investigating Metric Spike Report project. During the project, I gained knowledge about various techniques utilized for investigating metric spikes, and I also learned about the tools utilized for analyzing data. This project has provided me with practical knowledge about these concepts, enhancing my understanding of operation analytics and data analysis. Furthermore, the project has improved my problem-solving and analytical abilities.

DRIVE LINK:

<https://drive.google.com/file/d/1m186RQGVljZULBtQ9ReMMtxUjTZijS7N/view?usp=share_link>