Operation Analytics and Investigating Metric Spike

(Advanced SQL) **BY-Shaurya Gairola**

Project Description

Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. You work closely with the ops team, support team, marketing team, etc and help them derive insights out of the data they collect.

Being one of the most important parts of a company, this kind of 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 spike is also an important part of operation analytics as being a Data Analyst you must be able to 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 and for that its very important to investigate metric spike.

We have been provided two datasets , Case Study-1 Dataset(Job_Data) and Case Study-2 Dataset (Investigating Metric Spike) and required to provide a detailed report for the below two operations mentioning the answers for the related questions:

Case Study 1 (Job Data)

Below is the structure of the table with the deflnition of each column that you must work on:

- **Table-1:** job_data
  - **job_id:** unique identifier of jobs
  - **actor_id:** unique identifier of actor
  - **event:** decision/skip/transfer
  - **language:** language of the content
  - **time_spent:** time spent to review the job in seconds
  - **org:** organization of the actor
  - **ds:** date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

Use this dataset answer the questions that follows

1. **Number of jobs reviewed:** Amount of jobs reviewed over time.
   My task: Calculate the number of jobs reviewed per hour per day for November 2020?
2. **Throughput:** It is the no. of events happening per second.
   My task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?
3. **Percentage share of each language:** Share of each language for different contents.
   My task: Calculate the percentage share of each language in the last 30 days?
4. **Duplicate rows:** Rows that have the same value present in them.
   My task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

Case Study 2 (Investigating metric spike)

- **Table-1:** users
  This table includes one row per user, with descriptive information about that user's account.
- **Table-2:** events
  This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
- **Table-3:** email_events
  This table contains events specific to the sending of emails. It is similar in structure to the events table above.

Using this dataset answer the questions that follows

1. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
   My task: Calculate the weekly user engagement?
2. **User Growth:** Amount of users growing over time for a product.
   My task: Calculate the user growth for product?
3. **Weekly Retention:** Users getting retained weekly after signing-up for a product.
   My task: Calculate the weekly retention of users-sign up cohort?
4. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
   My task: Calculate the weekly engagement per device?
5. **Email Engagement:** Users engaging with the email service.
   My task: Calculate the email engagement metrics?

Approach

First of all I have imported the dastabase on my MySQL Workbench. Then I analyzed the database carefully. Observing all the tables, columns, rows, and relationship among all the table, and created ER Diagram of complete database provided.

Before finding the answers of the questions I need to have the data understanding of the database provided as well as the business understanding. Then I have done Data Profiling and created a Data Model like numbers of rows and columns we have in every Table, Datatypes,Keys,Relationships.

After doing all this , I started to find answers of the questions provided to me by the Operations Team by Querying the database.

Tech-Stack Used

I have used MySQL Workbench v8.0.31 by Oracle for project execution in order to query the database. The ease of access and setup, troubleshooting support as well as the GUI made it a good tool for the project.

Insights

Case Study 1 (Job Data)

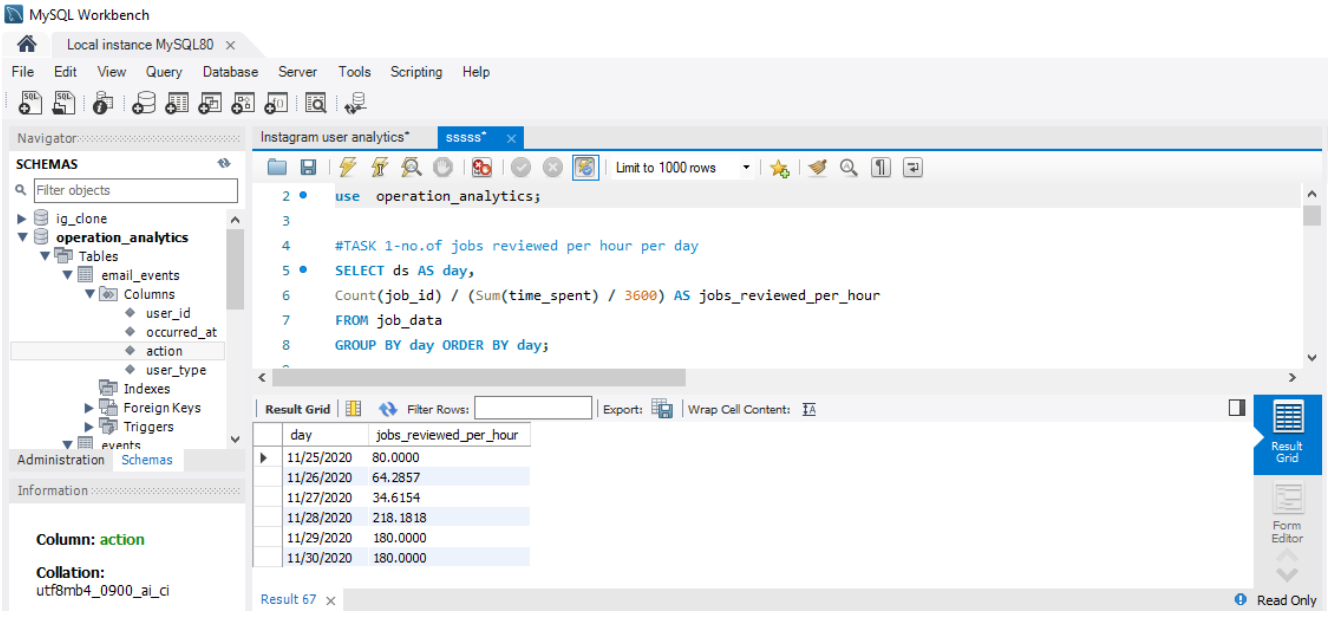
A) Number of jobs reviewed: Amount of jobs reviewed over time.

My task: Calculate the number of jobs reviewed per hour per day for November 2020?

QUERY:-

```
SELECT ds AS day,

Count(job_id) / (Sum(time_spent) / 3600) AS jobs_reviewed_per_hour

FROM job_data

GROUP BY day

ORDER BY day;
```

OUTPUT:-



B) Throughput: It is the no. of events happening per second.

My task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

QUERY:-

```
SELECT ds AS day,new.throughput,

avg(new.throughput) OVER ( ORDER BY ds rows BETWEEN 6 PRECEDING AND CURRENT row ) AS
7_day_avg_of_throughput

FROM

( SELECT ds, count(job_id) / sum(time_spent) AS throughput

FROM job_data

GROUP BY ds ) AS new

GROUP BY ds;
```

OUTPUT:-

Server Tools Scripting Help

Instagram user analytics* sssss*

Limit to 1000 rows

```
#TASK 2- 7 day rolling avg of throughput
SELECT ds AS day,new.throughput,
avg(new.throughput) OVER ( ORDER BY ds rows BETWEEN 6 PRECEDING AND CURRENT row ) AS
7_day_avg_of_throughput
FROM
( SELECT ds, count(job_id) / sum(time_spent) AS throughput FROM job_data GROUP BY ds ) AS new
GROUP BY ds;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

day	throughput	7_day_avg_of_throughput
11/25/2020	0.0222	0.02220000
11/26/2020	0.0179	0.02005000
11/27/2020	0.0096	0.01656667
11/28/2020	0.0606	0.02757500
11/29/2020	0.0500	0.03206000
11/30/2020	0.0500	0.03505000

Result 68

CONCLUSION:-

I think 7-Day Rolling Average would be much better than daily metric for better performance of any business , as it relates with only the recent trends and help us to compare which group of days perform better and it helps to understand why it is so, thus we can stay in trend and keep updating ourselves.

C.) Percentage share of each language: Share of each language for different contents.

My task: Calculate the percentage share of each language in the last 30 days?

QUERY:-

```
SELECT language,
count(job_id) as no_of_jobs,
count(job_id)*100 / sum(count(job_id)) OVER() as percentage_share
FROM job_data
GROUP by language;
```

OUTPUT:-

job_id	language	LANG_COUNT	PERCENTAGE
21	English	1	17.00
22	Arabic	1	17.00
23	Persian	3	50.00
25	Hindi	1	17.00
11	French	1	17.00
20	Italian	1	17.00

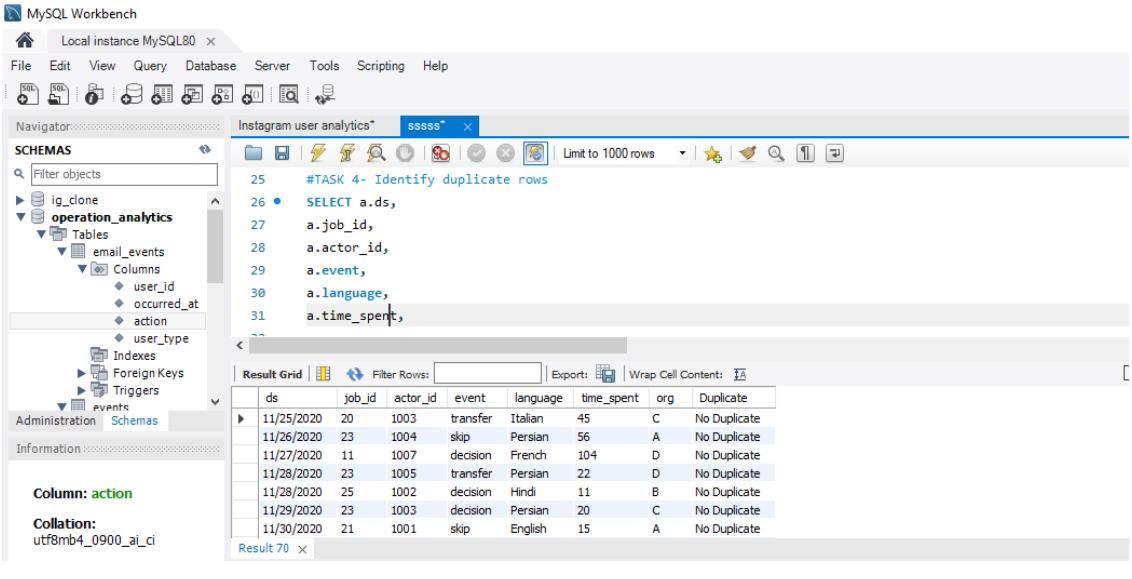
D.) Duplicate rows: Rows that have the same value present in them.

My task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

QUERY:-

```
SELECT a.ds,
a.job_id,
a.actor_id,
a.event,
a.language,
a.time_spent,
a.org,
CASE when a.duplicates = 1 then "No Duplicate" else "Duplicate" end as Duplicate
FROM
( SELECT *, row_number() OVER (partition by ds, job_id, actor_id, event, language, time_spent, org)
as duplicates FROM job_data ) as a ;
```

OUTPUT:-



ds	job_id	actor_id	event	language	time_spent	org	Duplicate
11/25/2020	20	1003	transfer	Italian	45	C	No Duplicate
11/26/2020	23	1004	skip	Persian	56	A	No Duplicate
11/27/2020	11	1007	decision	French	104	D	No Duplicate
11/28/2020	23	1005	transfer	Persian	22	D	No Duplicate
11/28/2020	25	1002	decision	Hindi	11	B	No Duplicate
11/29/2020	23	1003	decision	Persian	20	C	No Duplicate
11/30/2020	21	1001	skip	English	15	A	No Duplicate

Case Study 2 (Investigating metric spike)

A.) User Engagement: To measure the activeness of a user. Measuring if the user flnds quality in a product/service.

My task: Calculate the weekly user engagement?

QUERY:-

```
SELECT  WEEK(str_to_date(occured_at,'%Y-%m-%d'))  as  weekNumber,
        count(user_id) as UserCount
        from events
        group by weekNumber;
```

OUTPUT:-

weekNumber	UserCount
17	8404
18	18242
19	18178
20	18866
21	18112
23	19345
22	19455
24	20210
25	19717
29	21233
26	20126
30	22775
28	21908
27	21021
31	19585
32	17872
33	17445
34	17466
35	872

CONCLUSION:- We can see here the Week Number and the number of users active in that week. In this way we canmeasure the weekly engagement of users.

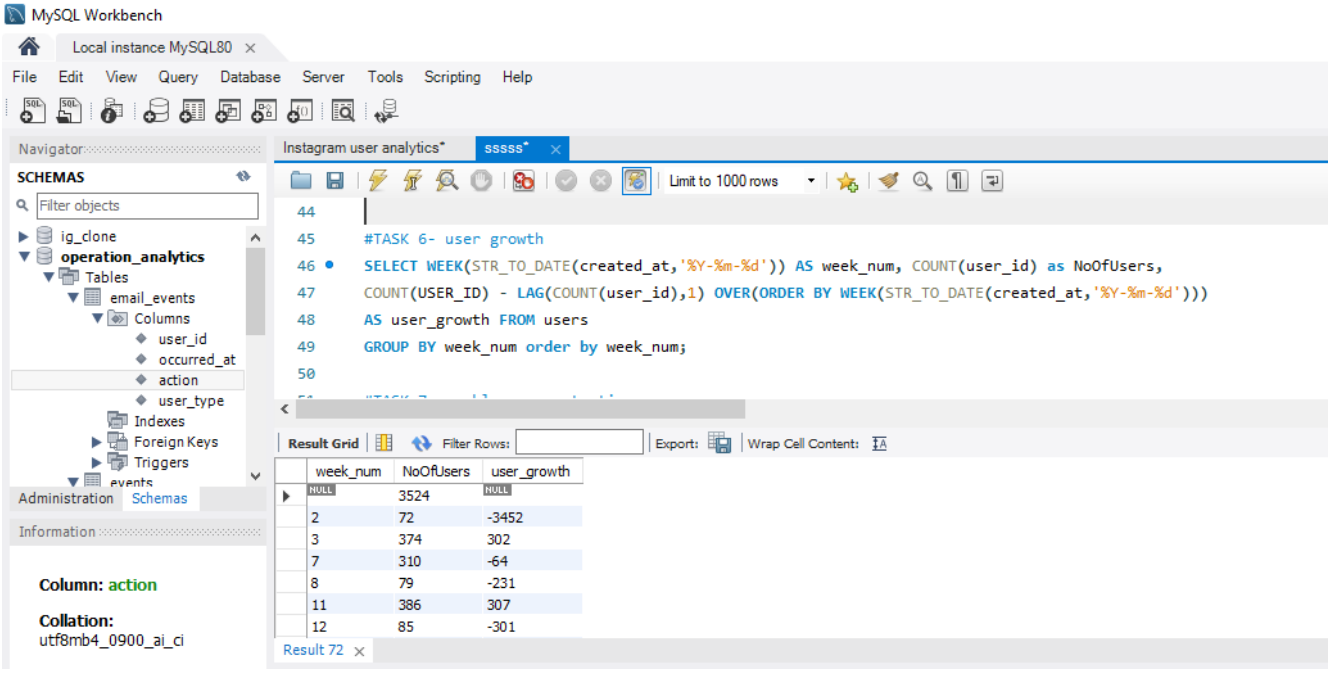
B) User Growth: Amount of users growing over time for a product.

My task: Calculate the user growth for product?

QUERY:-

```
SELECT  WEEK(STR_TO_DATE(created_at,'%Y-%m-%d'))  AS  week_num,

COUNT(user_id) NoOfUsers,


COUNT(USER_ID) - LAG(COUNT(user_id),1) OVER(ORDER BY WEEK(STR_TO_DATE(created_at,'%Y-%m-%d')))
AS user_growth

FROM users


GROUP BY week_num


order by week_num;
```

OUTPUT:-



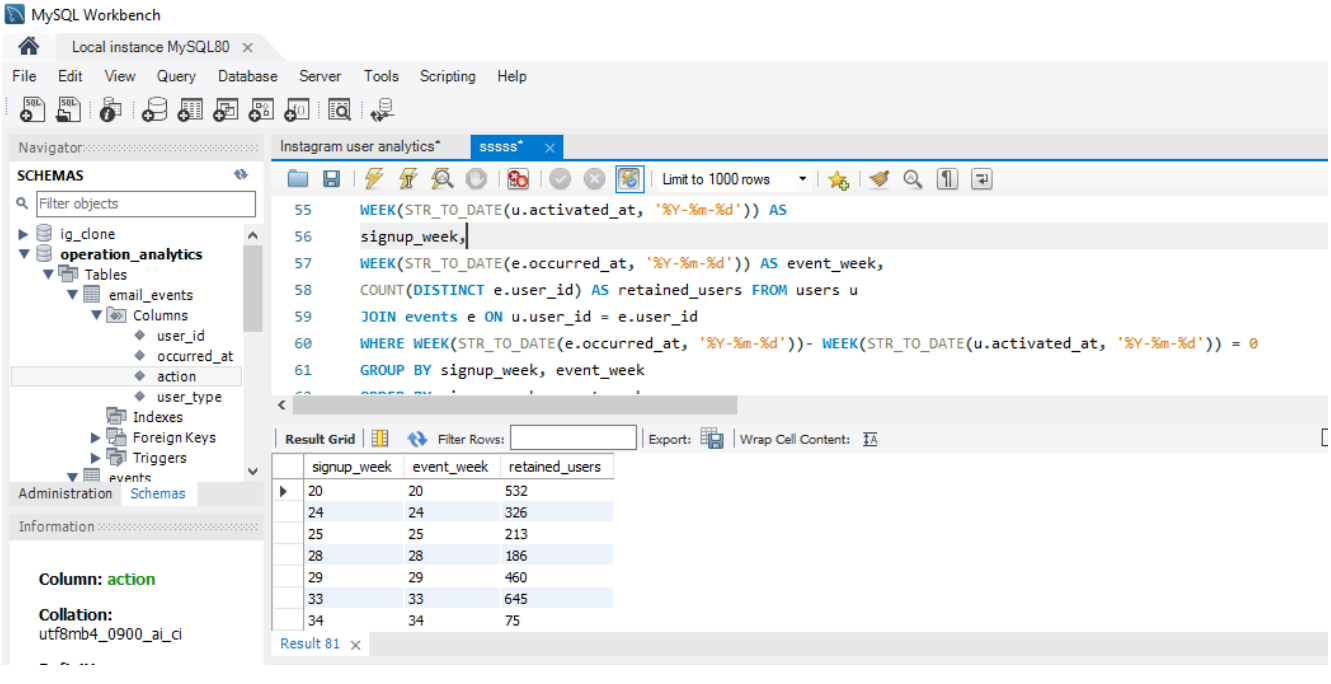
C.) Weekly Retention: Users getting retained weekly after signing-up for a product.

My task: Calculate the weekly retention of users-sign up cohort?

QUERY:-

```
SELECT

WEEK(STR_TO_DATE(u.activated_at, '%Y-%m-%d')) AS signup_week,

WEEK(STR_TO_DATE(e.occurred_at, '%Y-%m-%d')) AS event_week,

COUNT(DISTINCT e.user_id) AS retained_users

FROM users u

JOIN events e ON u.user_id = e.user_id

WHERE WEEK(STR_TO_DATE(e.occurred_at, '%Y-%m-%d'))- WEEK(STR_TO_DATE(u.activated_at, '%Y-%m-
%d')) = 0

GROUP BY signup_week, event_week

ORDER BY signup_week, event_week;
```

OUTPUT:-



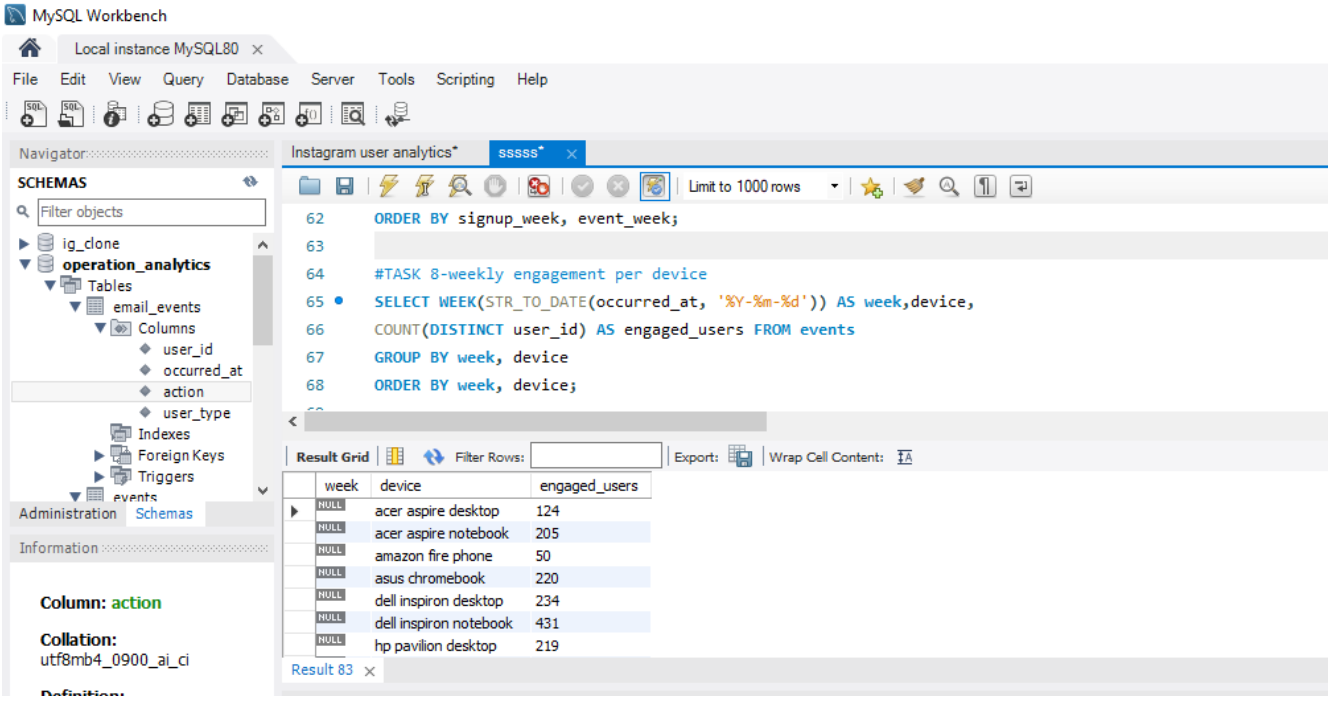
D.) Weekly Engagement: To measure the activeness of a user. Measuring if the user flnds quality in a product/service weekly.

My task: Calculate the weekly engagement per device?

QUERY:-

```
SELECT WEEK(STR_TO_DATE(occurred_at, '%Y-%m-%d')) AS week,device,

COUNT(DISTINCT user_id) AS engaged_users FROM events

GROUP BY week, device

ORDER BY week, device;
```

OUTPUT:-



Result 9 ×

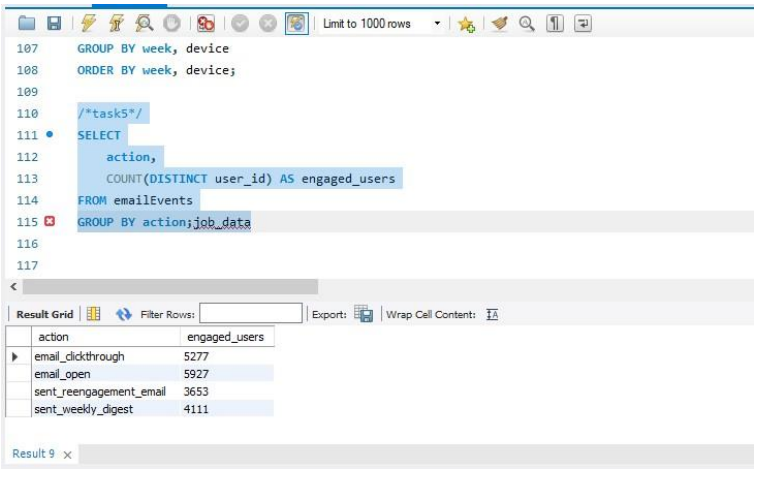
E.) Email Engagement: Users engaging with the email service.

My task: Calculate the email engagement metrics?

QUERY:-

```
SELECT
action,

COUNT(DISTINCT user_id) AS engaged_users

FROM email_events

GROUP BY action;
```

OUTPUT:-



Result

This initiative has yielded invaluable insights into user behaviors, growth patterns, and engagement dynamics. Notable accomplishments encompass:

- Identification of peak engagement periods and trends across weeks.
- Discerning the relationship between user activation and subsequent events.
- Assessing the efficacy of email interactions and gauging user responses to diverse actions.

These revelations contribute significantly to well-informed decision-making, enabling the formulation of targeted strategies to enhance user engagement, optimize acquisition endeavors, and elevate overall product performance. The application of MySQL Workbench alongside strategic SQL queries has proven to be efficacious in extracting meaningful information from the dataset, fostering a data-centric approach to decision-making