Operation Analytics and Investigating metric spike

Project Description:

The primary objective of the project is to leverage advanced SQL skills to analyse various datasets and tables, providing valuable insights to different departments within the company. The focus is on understanding and explaining sudden changes in key metrics, such as user engagement and sales, to identify areas for improvement in the company's operations. This project aims to facilitate data-driven decision-making processes and ultimately enhance overall performance.

Approach:

Data exploration and Understanding: thoroughly exploring the provided datasets and tables to understand the structure, relationships, and relevance of the data. Identify key metrics and performance indicators crucial for operational analysis, such as daily user engagement etc.

Metric Spike Investigation: Defining criteria for identifying metric spikes based on the nature of the data and business objectives. Develop SQL queries to detect sudden changes or anomalies in key metrics over specific time periods.

Root cause Analysis: Utilize sql queries to drill down to granular data and identify patterns, correlations, or anomalies associated with the spikes.

Tech Stack use:

MYSQL Workbench: It is a open source tool which can support multiple database systems that can perform complex calculations and data manipulation with ease.

Insights

Casestudy_1

The dataset provides details about review process of applicants for various job profiles
The column details of the dataset are:

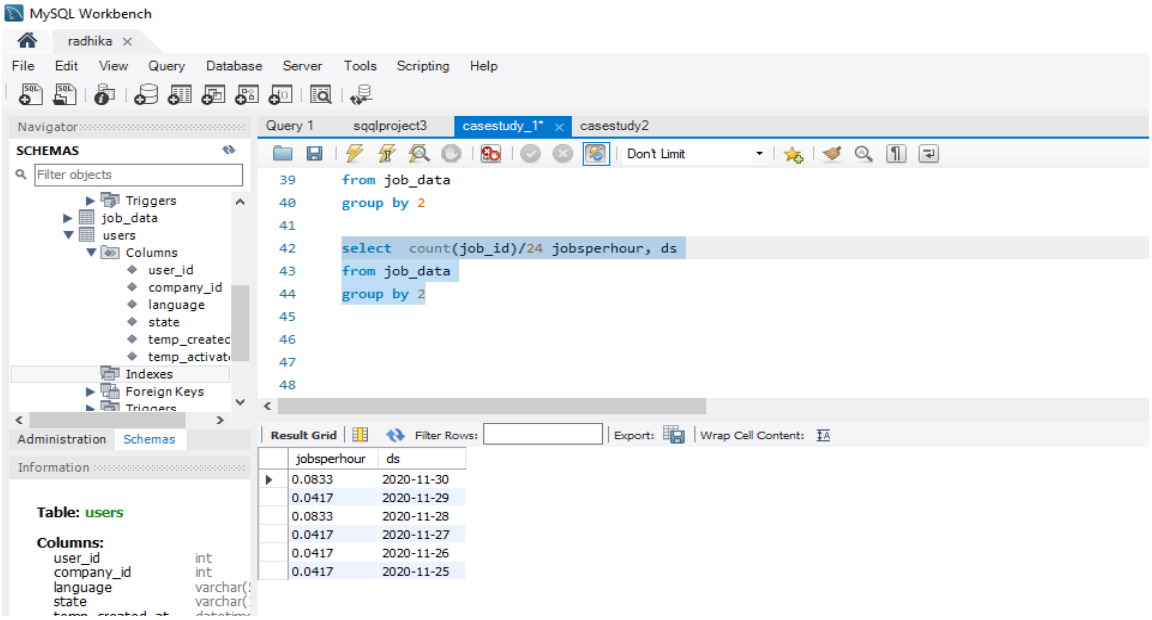
- **job_id:** Unique identifier of jobs
- **actor_id:** Unique identifier of actor
- **event:** The type of event (decision/skip/transfer).
- **language:** The Language of the content
- **time_spent:** Time spent to review the job in seconds.
- **org:** The Organization of the actor
- **ds:** The date in the format yyyy/mm/dd (stored as text).

Task 1: Jobs reviewed over time
No of jobs reviewed per hour each day in november 2020

SQL Code:

select count(job_id)/24 jobsperhour, ds
from job_data
group by 2

Analysing fluctuations in jobs across different days can help identify patterns in workload distribution and potentially correlate them with external factors such as weekdays, weekends, holidays or specific events. This can assist in resource planning, workload management, optimizing operational efficiency within the organization.

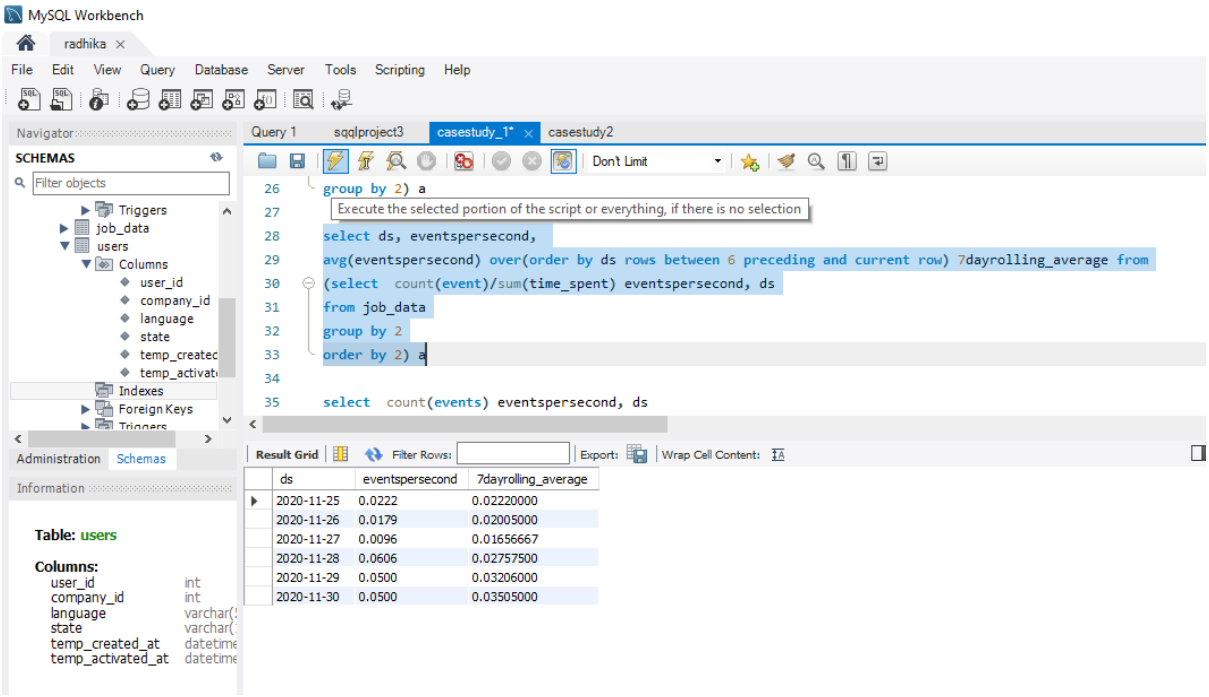


Task 2 (Throughput Analysis)

SQL Code:

select ds, eventspersecond,
avg(eventspersecond) over(order by ds rows between 6 preceding and current row) 7dayrolling_average from
(select count(event)/sum(time_spent) eventspersecond, ds
from job_data
group by 2
order by 2) a

the daily metric can give us a sudden rise or dip in the throughput which indicates the effects of seasonal factors like end of the month, holidays and weekend activity.
Using 7 day rolling average offers a smoothed-out view of throughput, reducing the impact of daily fluctuations and providing a more stable measure of overall performance trends. This approach is preferable for identifying long-term patterns and trends in throughput and is good for strategic decision making and trend analysis.

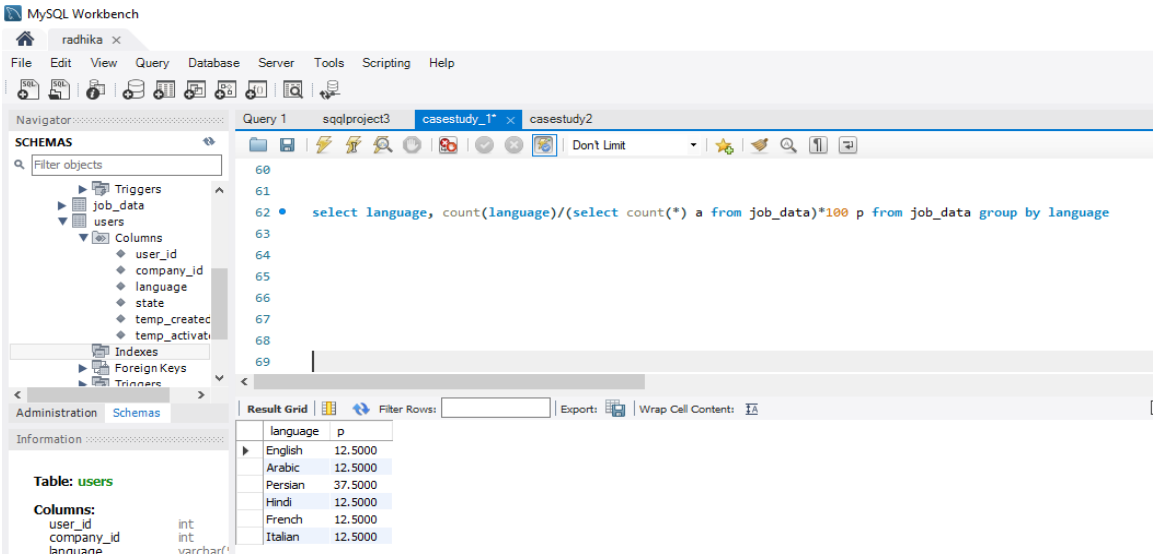


Task 3(language share)

SQL Code:
select language, count(language)/(select count(*) a from job_data)*100 p from job_data group by language

the query retrieves the percentage share of languages preferred or mostly in use. Very clearly from the query result, Persian language is most used.

Analysing the percentage share of languages can help identify the primary languages used in the events recorded in the last 30 days. This can indicate language sprcific marketing or understanding user demographics and preferences.

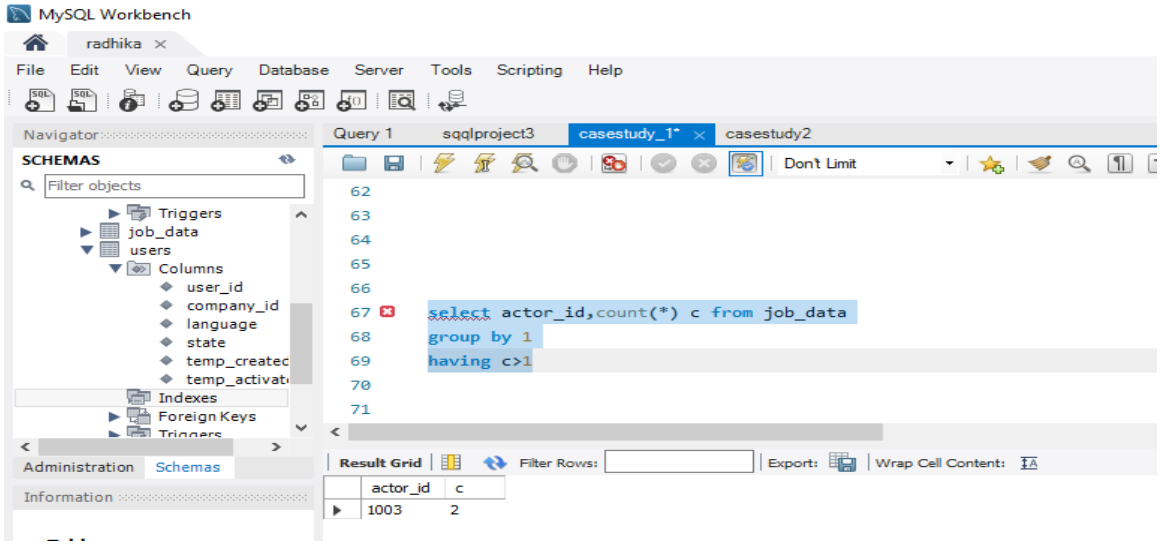


Task 4(duplicate rows)

To find duplicate rows we need to decide the column we are going to traverse for finding duplicates

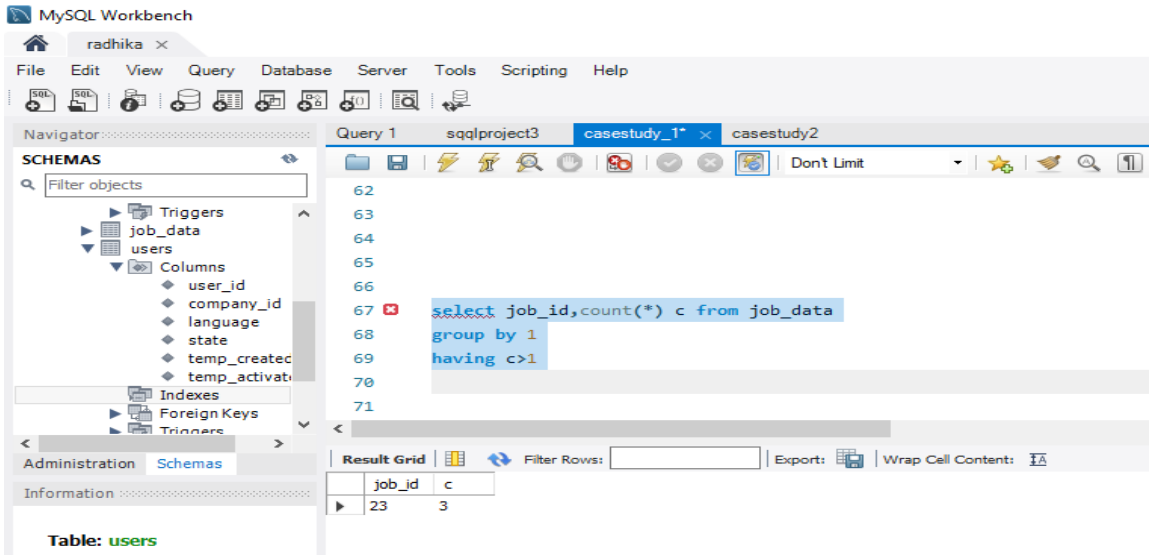
SQL Code:

```
select actor_id,count(*) c from job_data
group by 1
having c>1
```



If we consider actor_id there are 2 duplicate entries whereas if we consider job_id there are 3 duplicate entries

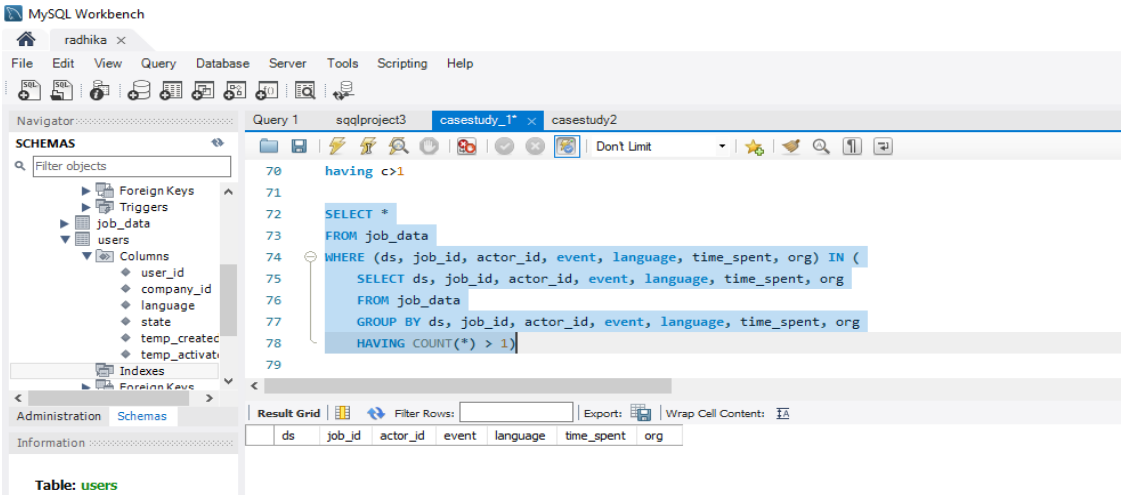
```
select job_id,count(*) c from job_data
group by 1
having c>1
```



But considering all the columns there are no duplicate entries in the table. duplicate job_id can indicate same job was being reviewed by different actors or duplicate actors could mean single actor was reviewing different jobs hence considering all the columns there are no duplicate rows in the table.

SQL Code:

```
SELECT *
FROM job_data
WHERE (ds, job_id, actor_id, event, language, time_spent, org) IN (
   SELECT ds, job_id, actor_id, event, language, time_spent, org
   FROM job_data
   GROUP BY ds, job_id, actor_id, event, language, time_spent, org
   HAVING COUNT(*) > 1)
```



Casestudy_2

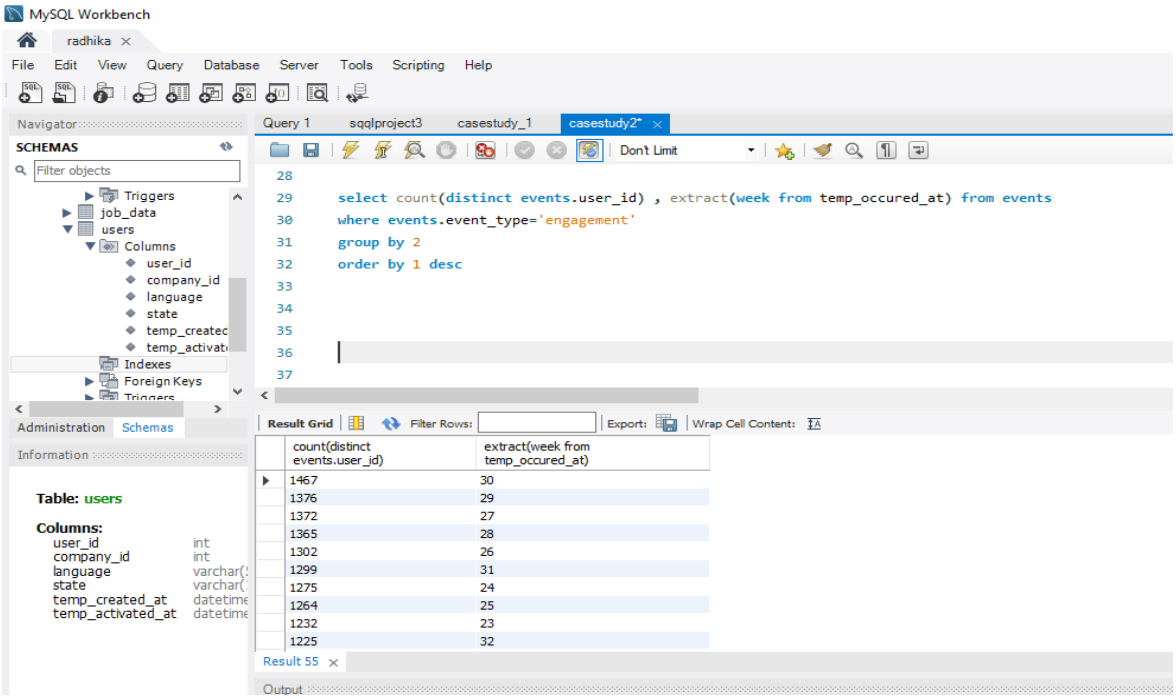
Task1(weekely User engagement)

SQL Code:

```
select count( distinct events.user_id), week(events.temp_occured_at) w from events
where events.event_type='engagement'
group by 2
order by 1 desc
```

the count is calculated on weekely basis and the lowest activity can be seen on week 35 with 104 users and highest activity on week 30 with 1467 users

Analysing weekly user engagement can inform decisions related to marketing campaigns, product updates and user experience enhancements to improve overall engagement and lower count can indicate potential issues with retention and activity.



Task 2(User growth over time)

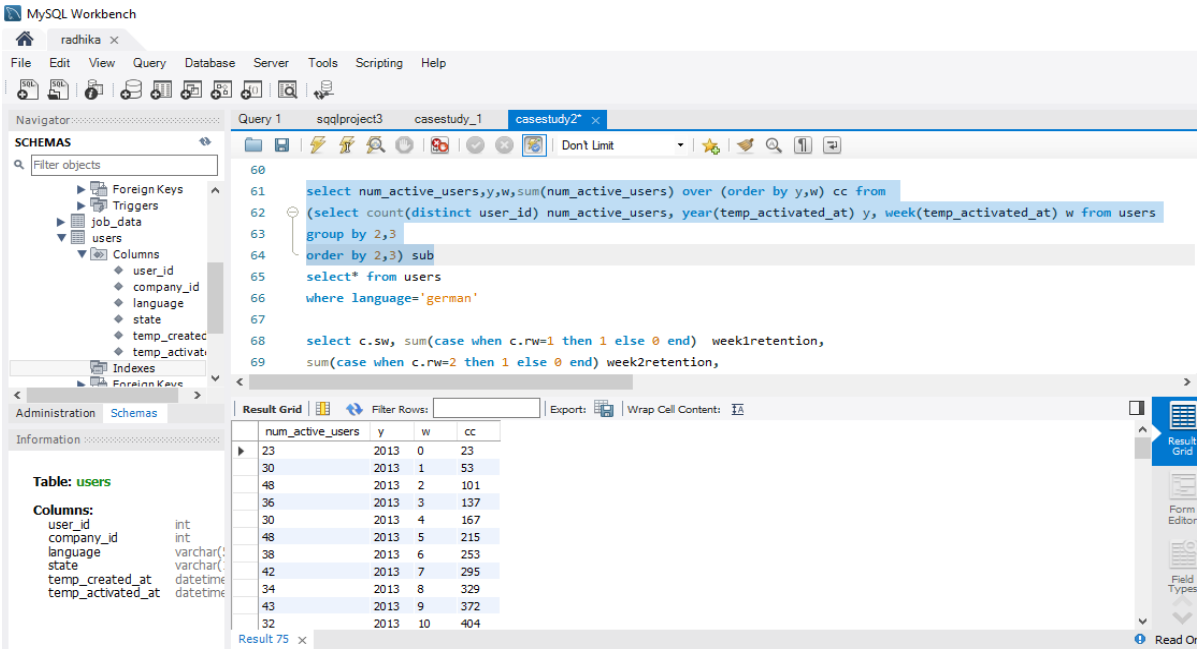
SQL Code:

select num_active_users,y,w,sum(num_active_users) over (order by y,w) cc from
(select count(distinct user_id) num_active_users, year(temp_activated_at) y,
week(temp_activated_at) w from users
group by 2,3
order by 2,3) sub

we can see the user count started at 23 at starting of the year 2013 and has wen till 9381 to the end of year 2014. if we compare for both years separately the user growth has increased over the weeks.

Analyzing growth trajectory of the products user base over time can inform strategic decisions related to marketing, product development and user engagement to sustain and accelerate growth.

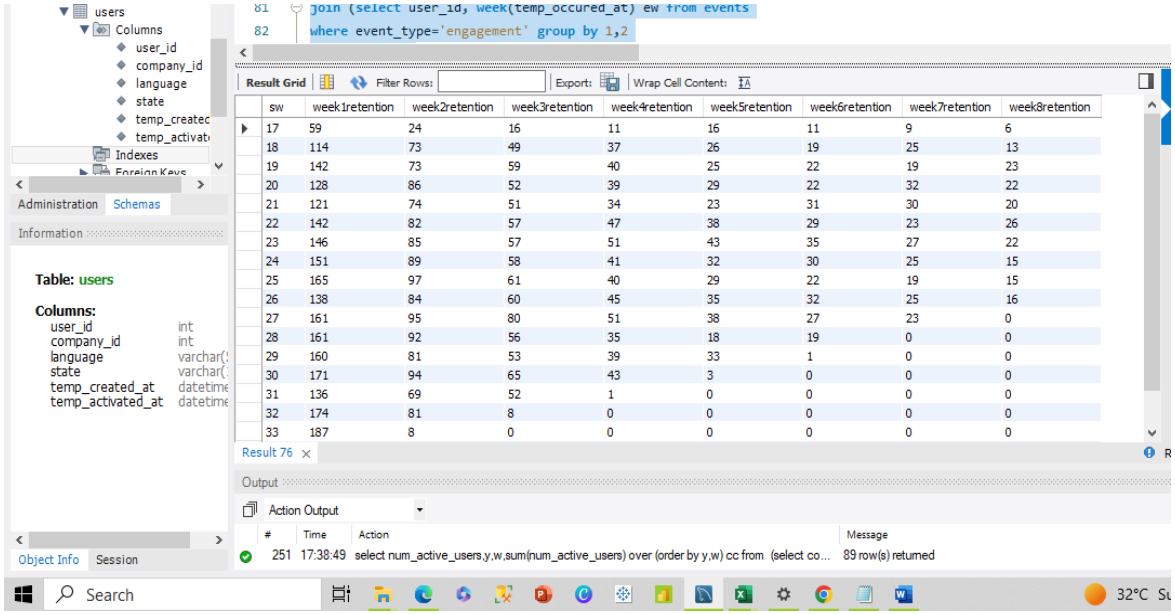
But we look into daily metric of num of active users created everyday the numbers may go low sometimes, considering the daily results and other related factors like seasonal factors like engagement on weekdays and weekends there can be adaption of improvements in acquisition strategies, products features, or market conditions



Task 3(weekely customer Retention)

SQL Code:

```
select c.sw, sum(case when c.rw=1 then 1 else 0 end)  week1retention,
sum(case when c.rw=2 then 1 else 0 end) week2retention,
sum(case when c.rw=3 then 1 else 0 end) week3retention,
sum(case when c.rw=4 then 1 else 0 end) week4retention,
sum(case when c.rw=5 then 1 else 0 end) week5retention,
sum(case when c.rw=6 then 1 else 0 end) week6retention,
sum(case when c.rw=7 then 1 else 0 end) week7retention,
sum(case when c.rw=8 then 1 else 0 end) week8retention
from
(select  a.user_id, a.sw, b.ew, b.ew-a.sw rw from
((select user_id, min(week(temp_occured_at)) sw from events
where event_type='signup_flow' and event_name='complete_signup' group by 1
order by 2) a
join (select user_id, week(temp_occured_at) ew from events
where event_type='engagement' group by 1,2
order by 2) b
on a.user_id=b.user_id)) c
group by c.sw
order by c.sw
```



The code written is for weekly retention for successive 8 weeks starting from the initial signup of the users for each week. we can add many weeks as we want to check retention. clearly from the results the retention has decreased over the weeks.
By analysing the retention rates, we can assess the effectiveness of onboarding process and engagement strategies higher retention rates indicate larger portion of users are continuing to engage with the platform and overall user experience is good but lower retention rates may indicate areas for improvement is user experience, feature adoption, or communication strategies to retain users.

Task 4(weekly engagement per device)

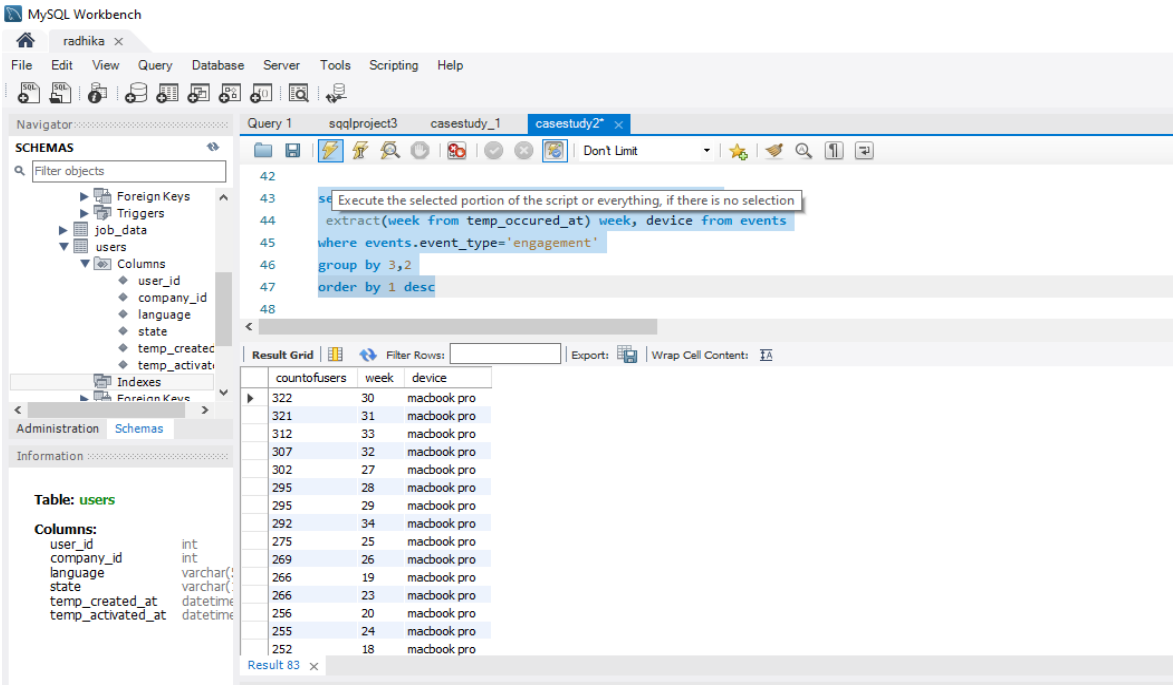
SQL Code:

```
select count(distinct events.user_id) countofusers,
extract(week from temp_occured_at) week, device from events
where events.event_type='engagement'
group by 3,2
order by 1 desc
```

the results can help identify trends in weekely user engagement and assess the impact of different devices on user interaction.

the top device used by the users are macbook pro followed by Lenovo thinkpad which can indicate on planning marketing campaigns targeting users of that device or improvements in the user experience specific to that device.

Overall this analysis provides valuable inisghts for optimizing user engagement startegies , device compatibility.



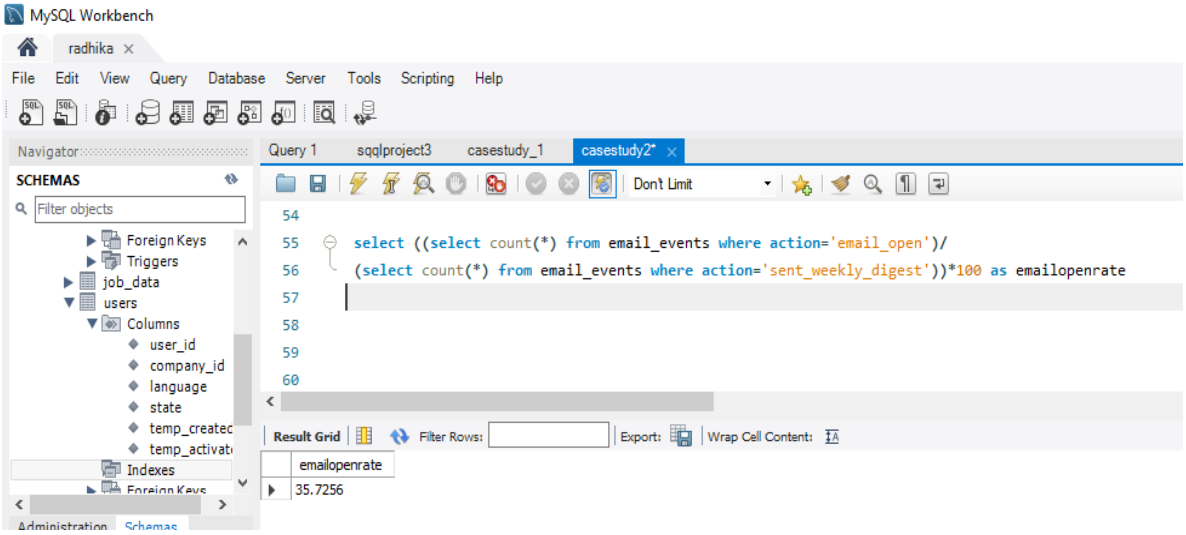
Task 4(Email Engagement Analysis)

There are variety of different email engagement rates to determine the effectiveness for email marketing strategy and content, based on the dataset here are few email engagement rates

1. **Open rate**- which is emails opened divided by total sent emails.
   Based on the recent industry standards for professional services the expected rate is 28.31% and this can vary based on the particular industry or companies goals and In this case the rate is 35.72% which exceeded the target and based on companies goals and objectives, marketing strategy should be improvised accordingly.

SQL Code:

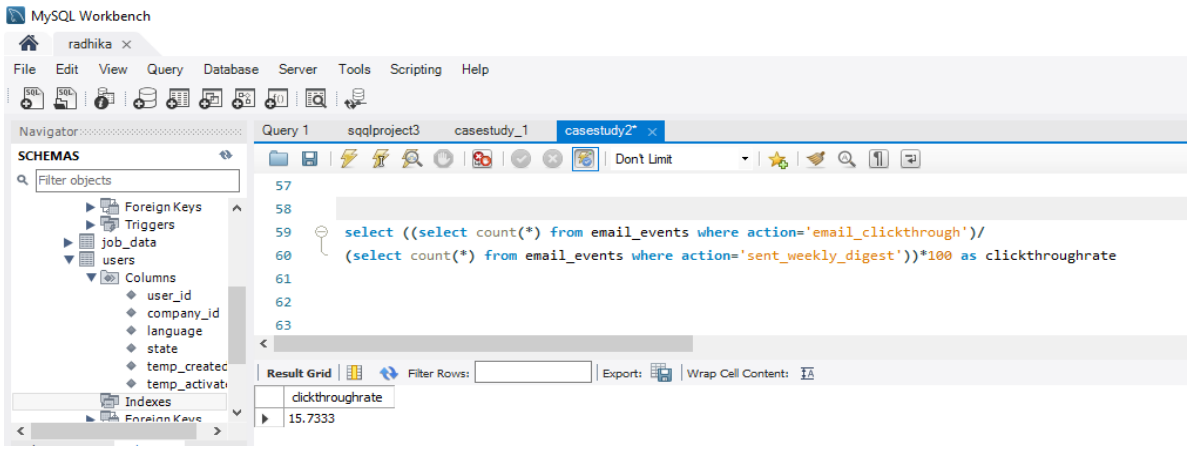
select ((select count(*) from email_events where action='email_open')/(select count(*) from email_events where action='sent_weekly_digest'))*100 as r



2. **Click-through rate(CTR)** - which is links clicked in email divided by total sent emails.
   Based on the industry standards for professional services the expected rate is 3.01% and In this case the rate is 15.73% which is good and can indicate the user might have opened links multiple times.

SQL Code:

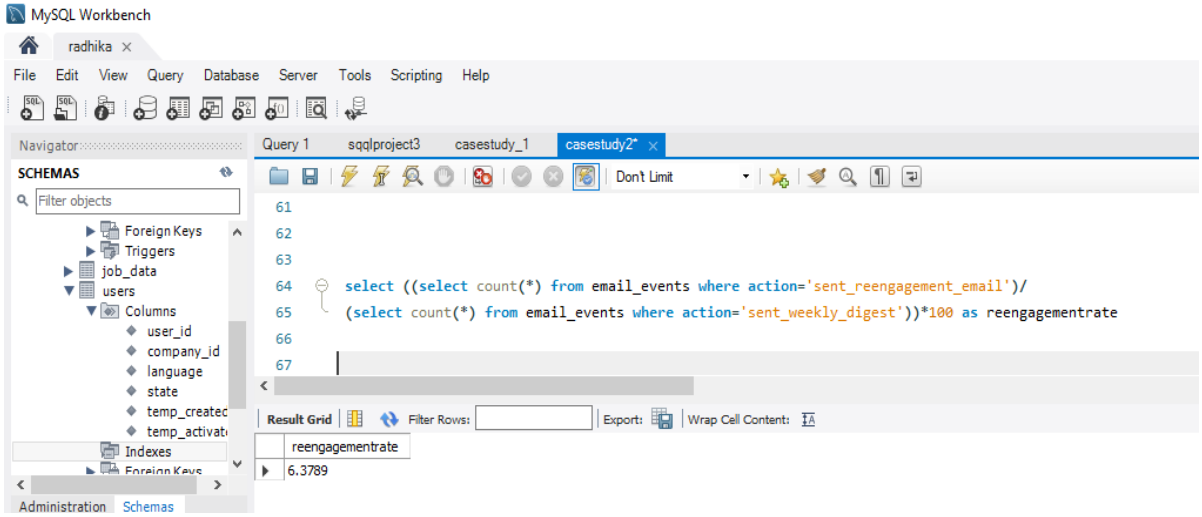
select ((select count(*) from email_events where action='email_clickthrough')/(select count(*) from email_events where action='sent_weekly_digest'))*100 as r



3. **Re-engagement rate**- which is emails sent to follow-up divided by total sent emails.
   The rate falls below 10% which is good as the follow-up mails to be sent could be less, but this clearly doesn't indicate the successful conversion has happened. So further analysis can be conducted on decision mailing.

SQL Code:

select ((select count(*) from email_events where action='sent_reengagement_email')/(select count(*) from email_events where action='sent_weekly_digest'))*100 as r



Result:

Understanding of operational processes: the project provide me with a deeper understanding of operational process, including data collection, analysis and interpretation to improve overall business operations.

Advanced SQL skills: by working with various datasets and tables and performing complex SQL queries to derive insights. I have enhanced my proficiency in SQL, particulary in data manipulation, aggregation and window functions.

Strategic Insights for performance improvement: Analyzing trends and patterns in metrics like user engagement and throughput has provided strategic insights for performance improvement, enabling data-driven decision-making to optimize operations and enhance overall business performance.

Preference for metric selection: The project has also reinforced the importance of selecting appropriate metrics for analysis, considering factors such as granularity, stability and relevance for business objectives. For example choosing between daily metrics and rolling averages for throughput analysis depends on the desired level of granularity and the need for short-term versus long-term, insights.