**Project – 3**

**Operation Analytics and Investigating Metric Spike**

**Project Description**

The project aims to perform operational analytics which helps organizations to analyse their performance and end to end operations. The Job Data Analysis and Investigating Metric Spikes case studies will be used to perform dataset analyses using advanced SQL queries. This will help to extract insights to help decision making in marketing and operations.

**Approach**

* First we load the dataset provided by importing CSV files using MySQL workbench.
* Then we observe the tables to understand the relationship between each columns and table.
* We then analysed the data as per the situations given by performing optimized SQL queries. It helps to derive key business insights.
* Finally we verify the queries by comparing it to expected outcome and checking its accuracy.

**Tech-Stack Used**

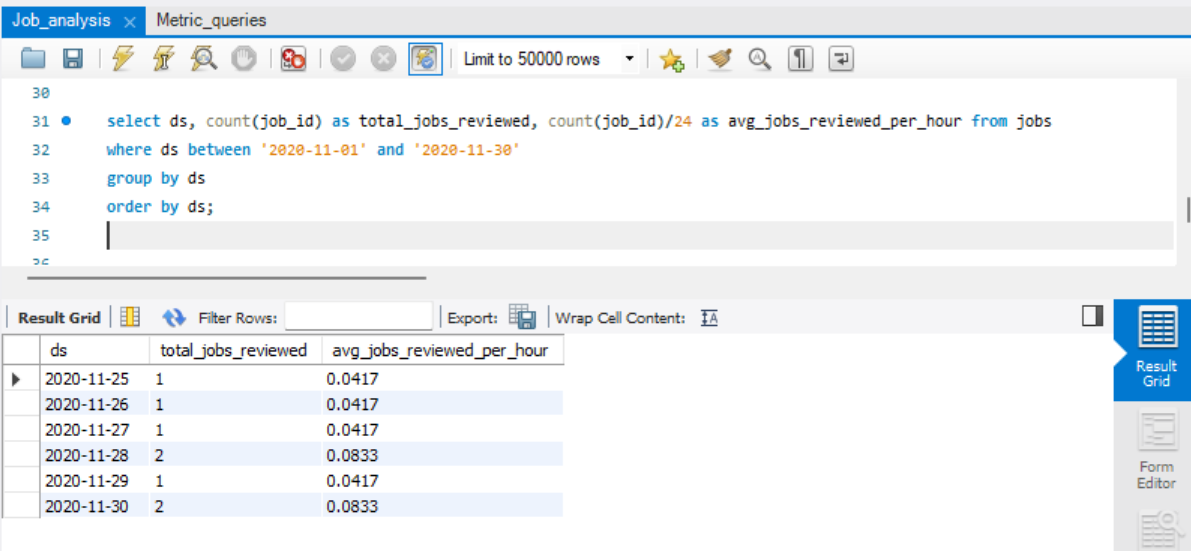
* Used MySQL workbench Version 8.0.41.0 for database management and query execution.
* Excel sheet for data visualization.

**Insights**

**Case Study 1: Job Data Analysis**

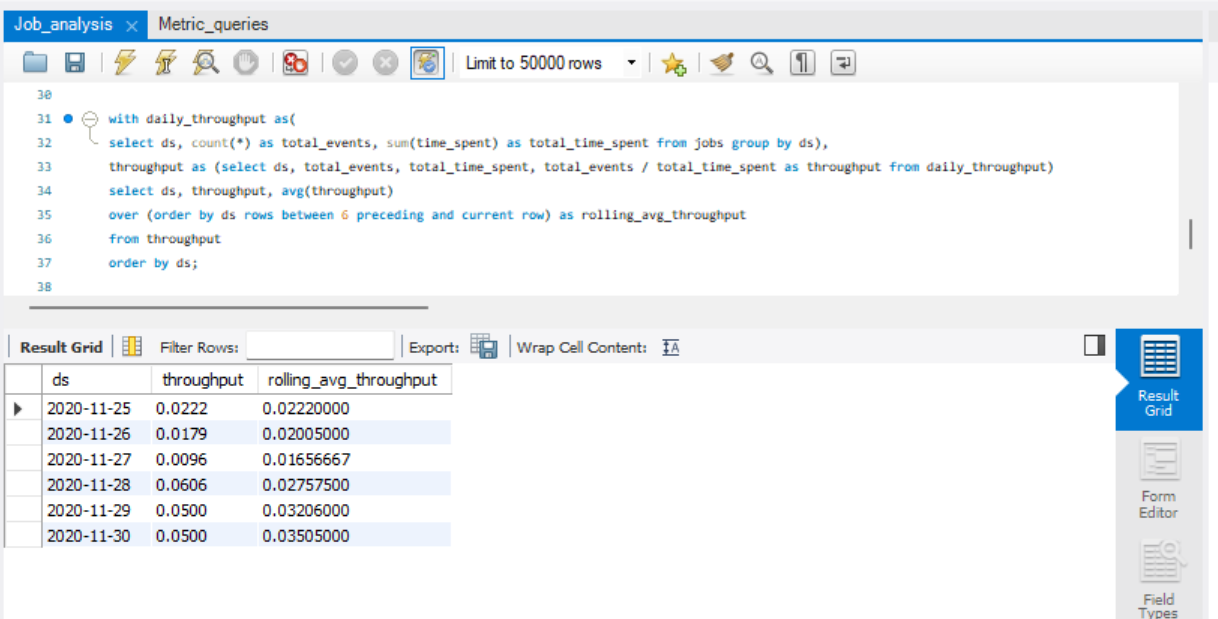
1. Jobs reviewed over time

We have calculated total number of jobs reviewed per day in November 2020 and the average jobs reviewed per hour by dividing the count by 24. Then it is grouped by data ds and it is then sorted in the ascending order.

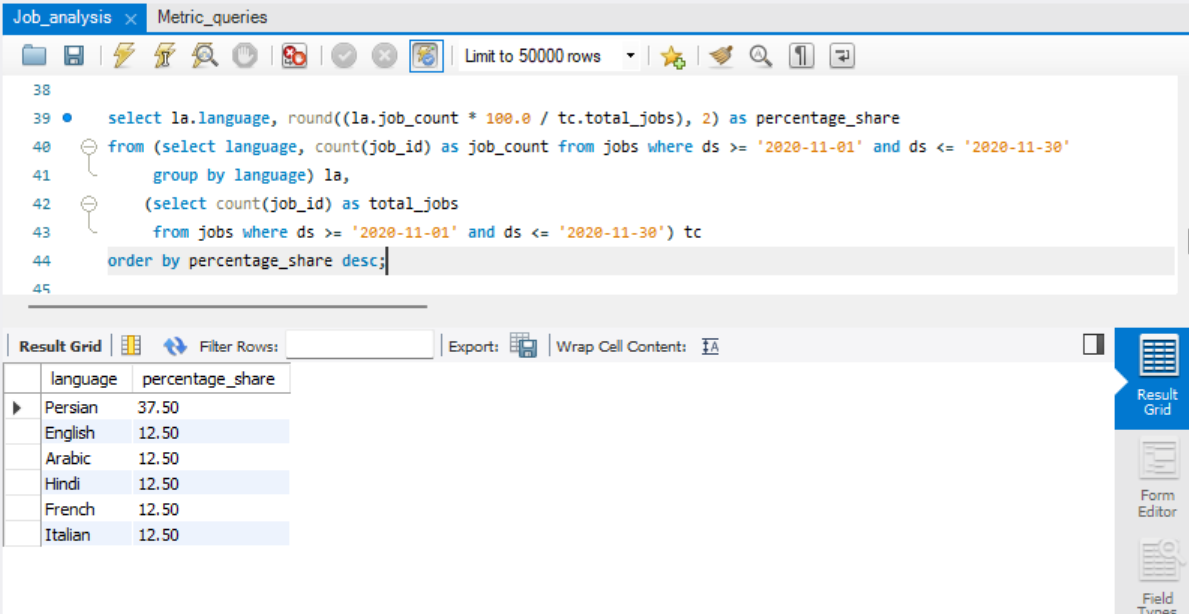
Insights- The above table shows the average jobs reviewed per hour.

1. Throughput analysis

Daily throughput is calculated by the number of events per total time spent. Then the rolling average throughput is calculated for 7 days considering the current day and previous 6 days. At the end the results are ordered by data ds To have an overview of trends overtime.

Insights- The above table shows the rolling average throughput as per the days.

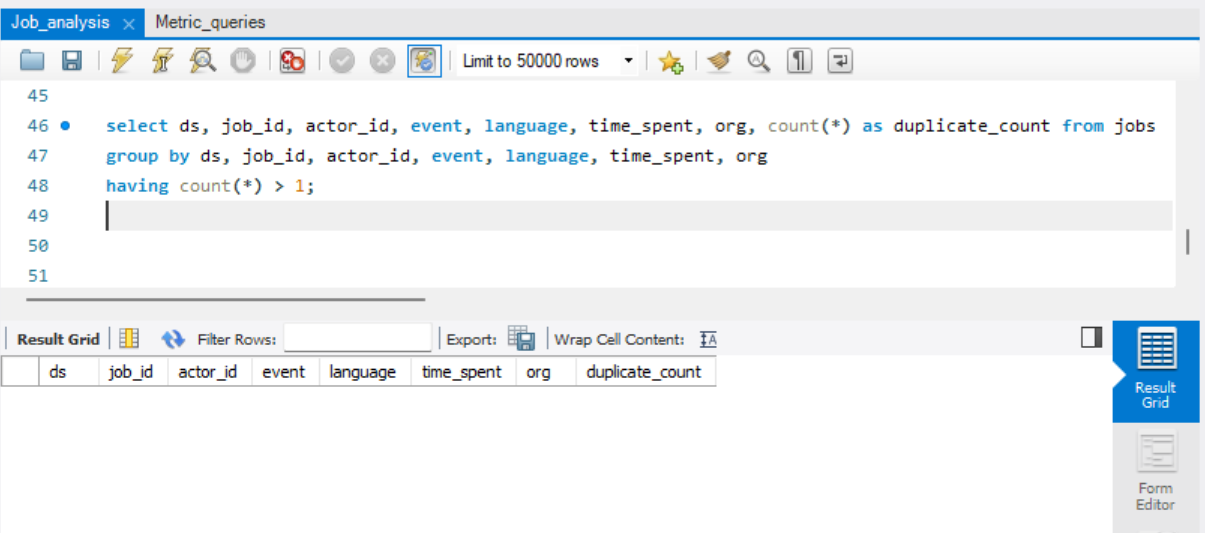
1. Language share analysis

We have calculated percentage share of each language by counting the number of jobs for each language la and total number of jobs tc. Then we have computed share of each language in percentage by dividing job count to total bs and into 100, then we have rounded it of to two decimal places and at the end we have printed it in descending order.

Insights- The above table shows the percentage of language share and shows that persian has the most percentage share.

1. Duplicate rows detection

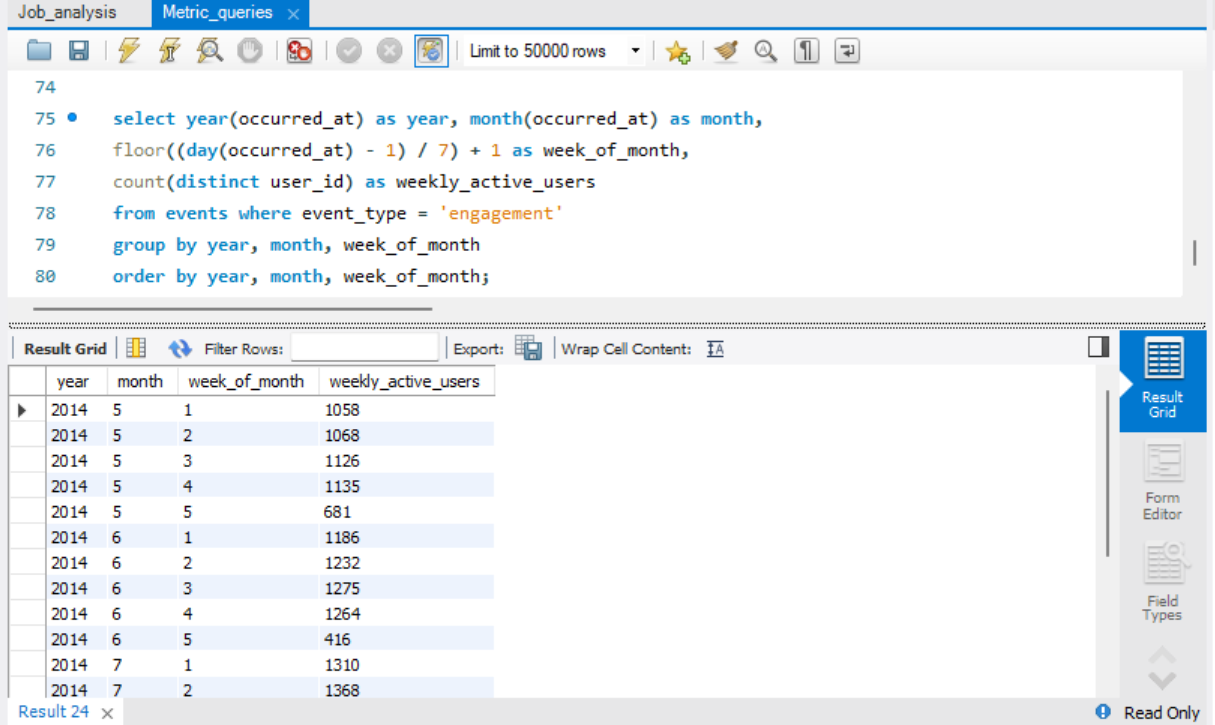
We have write this query to identify the duplicate rows in jobs table. We have first grouped the records by columns. We have then counted the number of the group has occurred and then filter it out. We have given the filter to only having more than one occurrence. Through this we are able to find duplicate count.

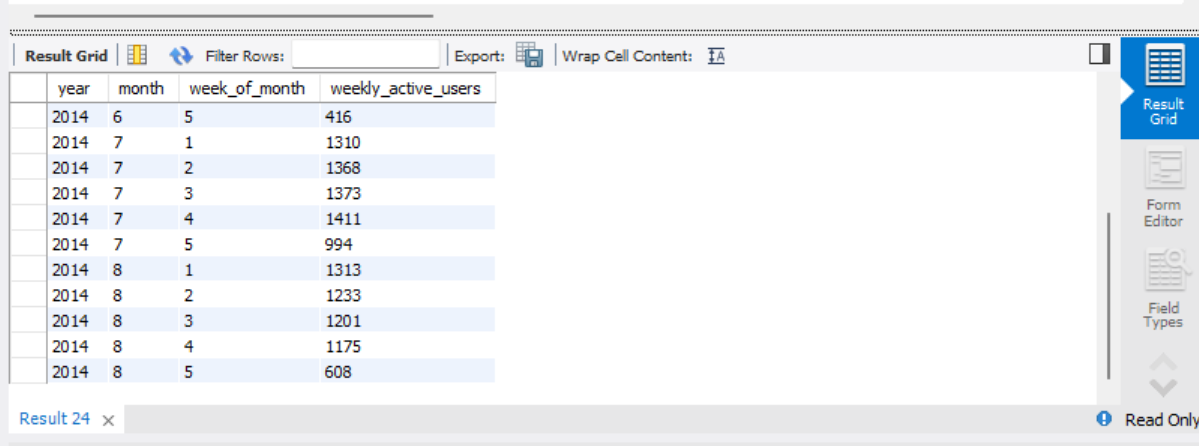
 Insights- The above output shows that according to the dataset there is no duplicate rows currently present in the dataset.

**Case Study 2: Investigating Metric Spike**

1. Weekly user engagement

Here we have aggregated weekly active users by year and month for events which have classified as engagement through this we can extract year and month from the column occurred\_at. We have used floor function to calculate the week of the month. Weekly active users are counted by distinct user\_id for each week. In the end we have grouped and ordered by year, month and week of month.

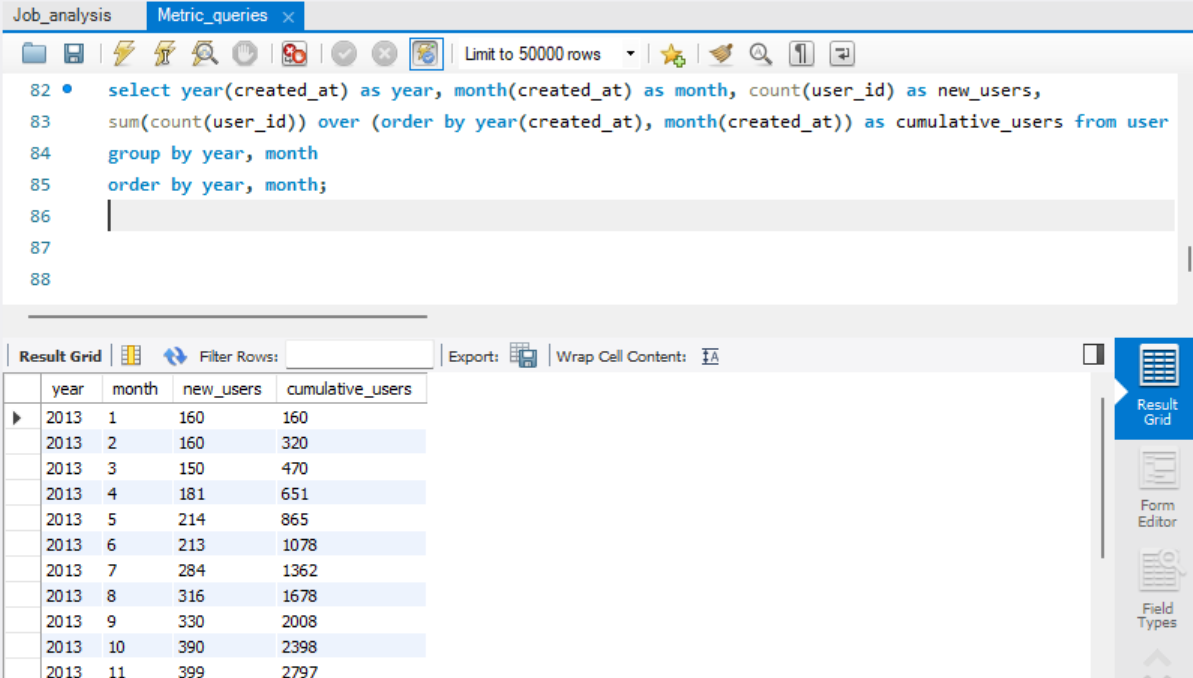


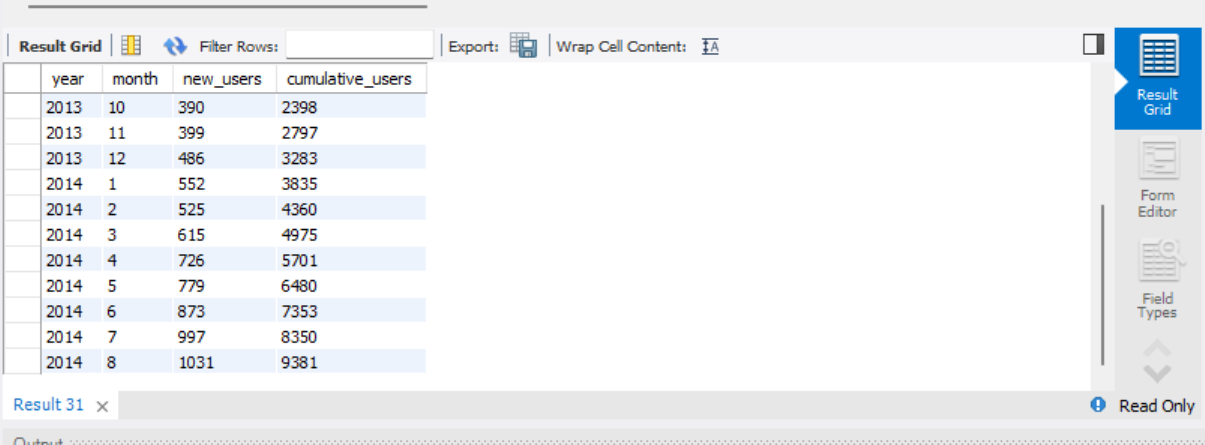


Insights- The above table shows the weekly active users according to the month.

1. User growth analysis

In this we have retrieved number of users added in each month as an active user and then we have cumulate number of users over time to find user growth. We first have extracted the year and month from the created\_at column from user table. To find monthly user we have counted user\_id for each month. We have then used sum window function to calculate cumulative total which is ordered by year and month. In the end the result is grouped and ordered by year and month.

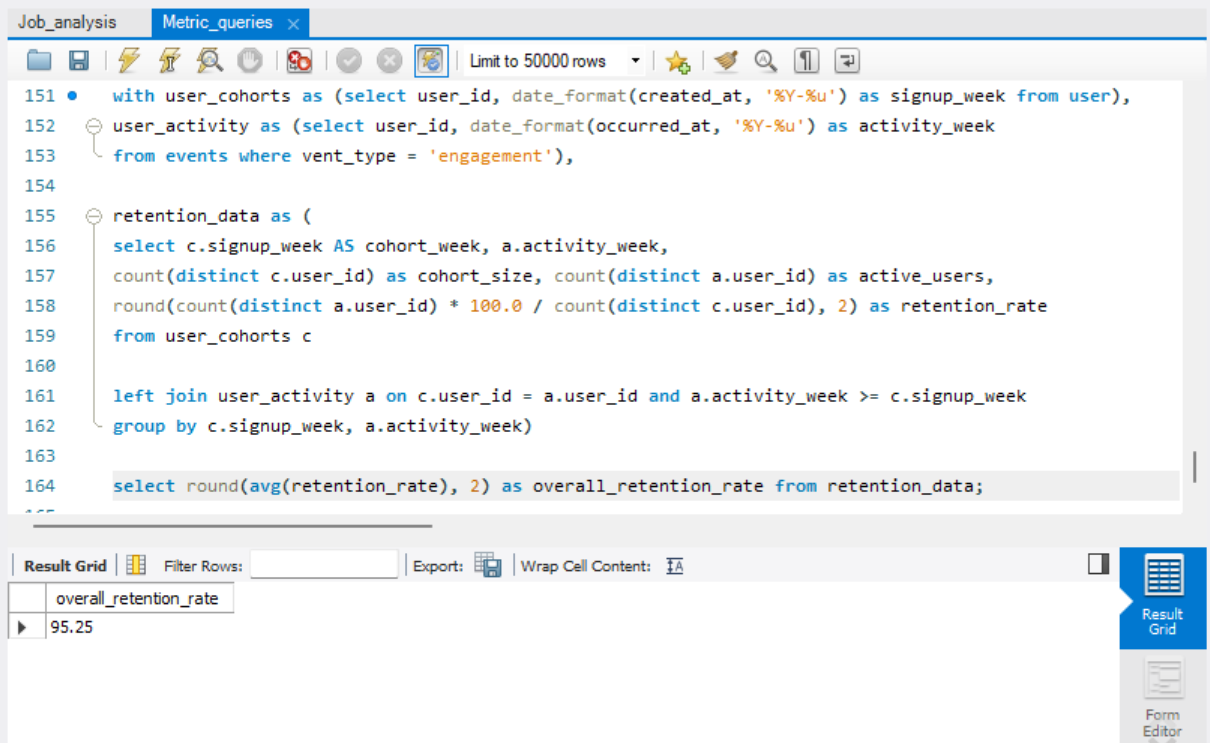




Insights- The above table shows the number of new users as per the month and then we have added the users to find the growth analysis and 9381 are number of users the growth has been seen.

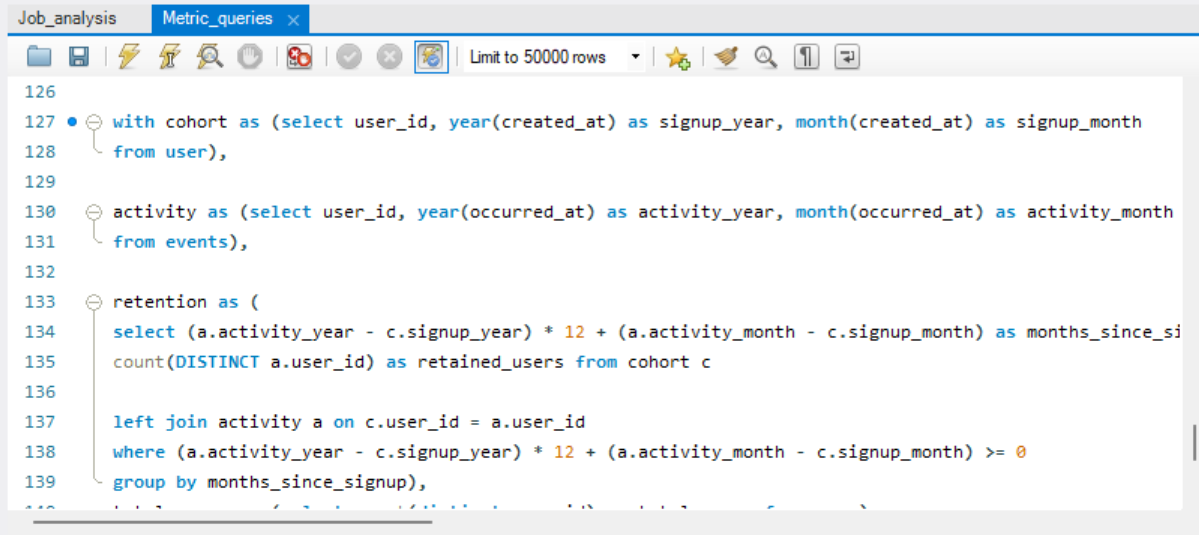
1. Weekly retention analysis

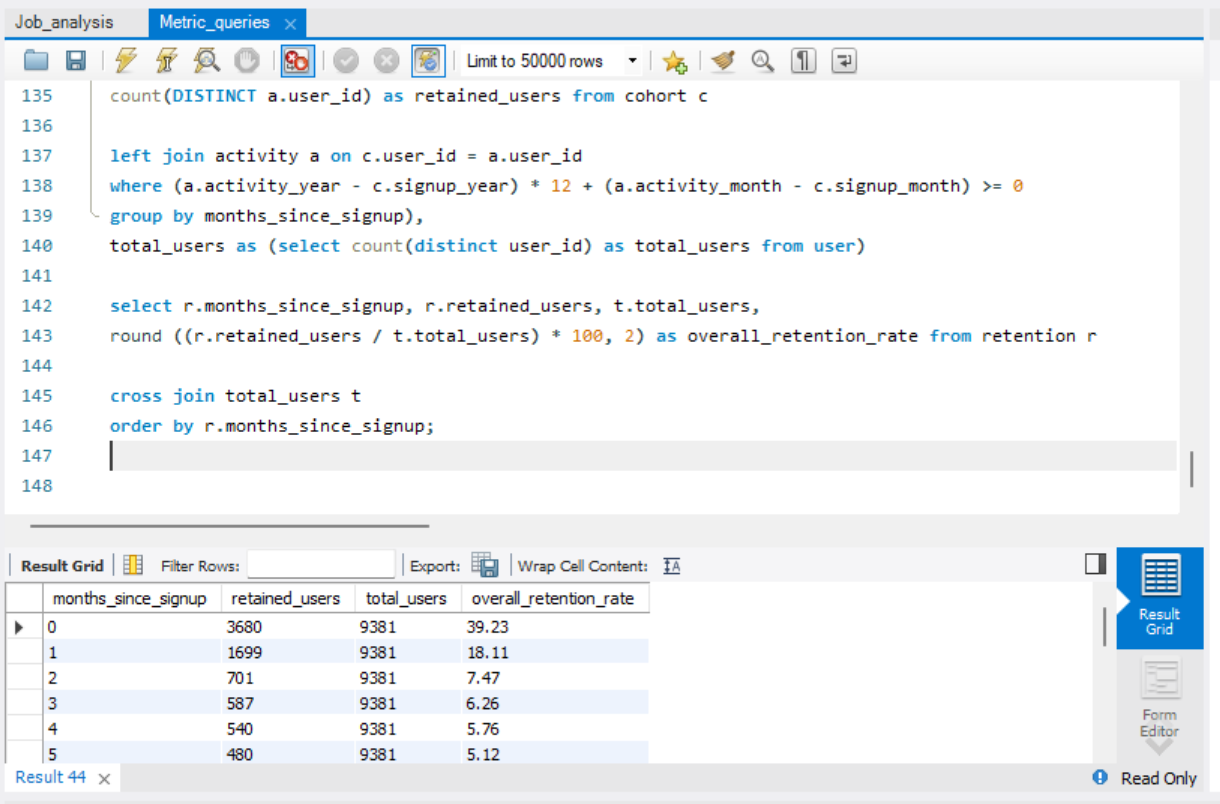
First to understand we have calculated the overall retention rate of user cohorts over time. By creating subqueries to determine the signup week and activity week for every user. Then the retention\_data subquery is used to group user by signup week and then to finally calculate retention rates for each cohort by comparing the number of active users to chort size. Then at the end finally we have calculated the average retention rate across all cohorts.

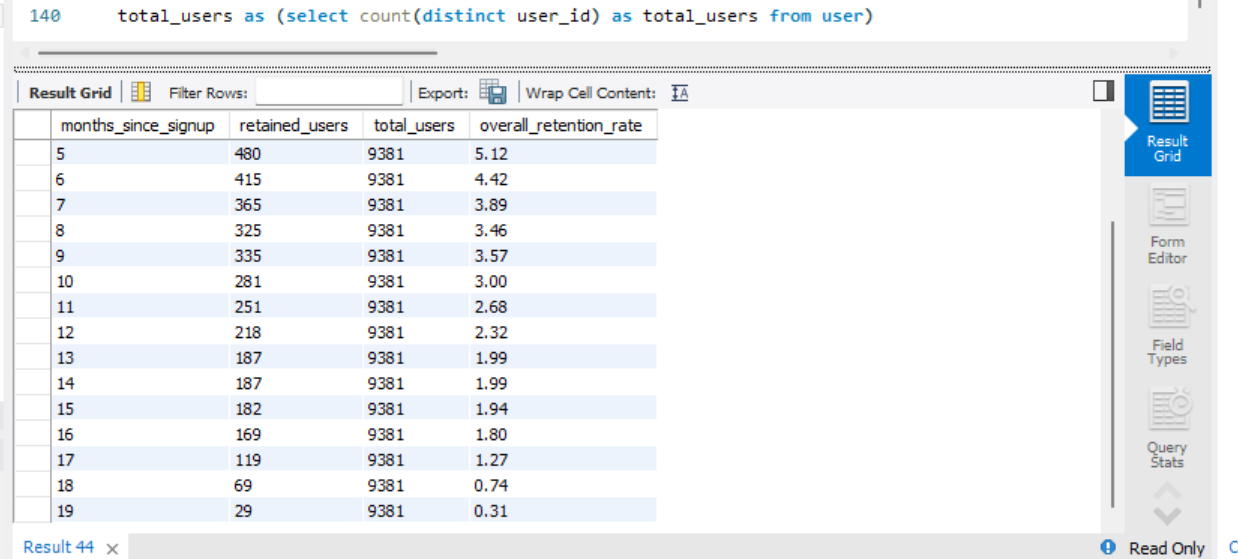


Insights- The above output shows that the retention rate is 95.25.

Here we have calculated overall retention rate of users over the time. First we have calculated subqueries first to track user signups second to track activities of users and third retention is made to determine number of retained users per month since signup. Then to find total number of users another query total\_users is made. Finally we have calculated overall retention rate by dividing number of retained users by total number of users. Finally we have grouped and ordered the result by months since their signup.

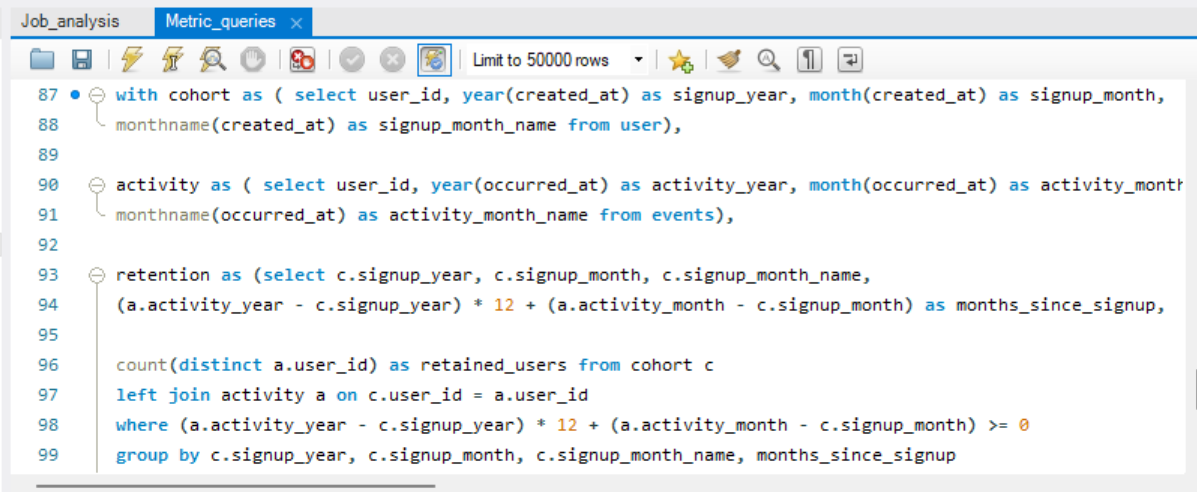


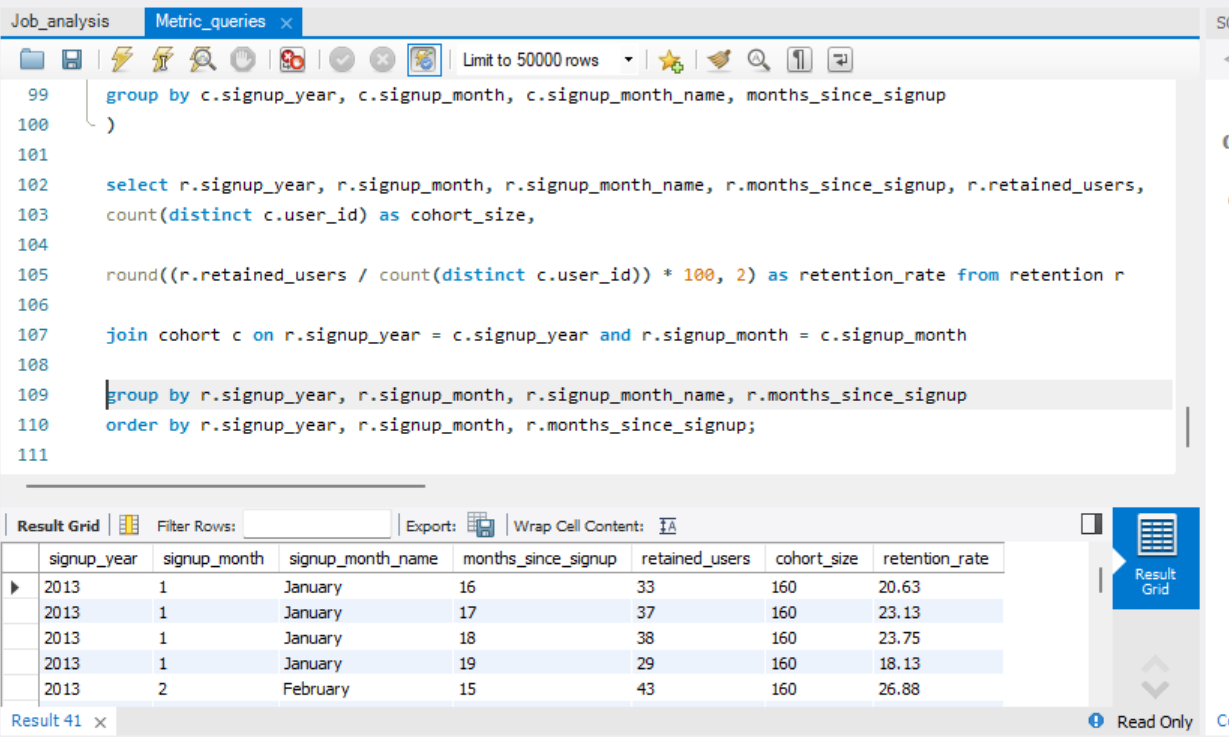


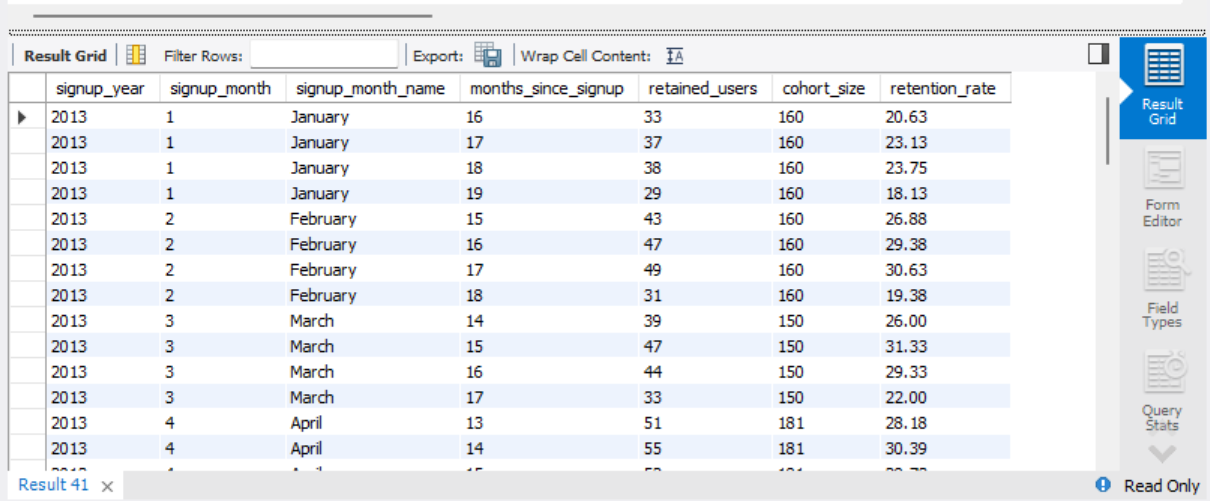


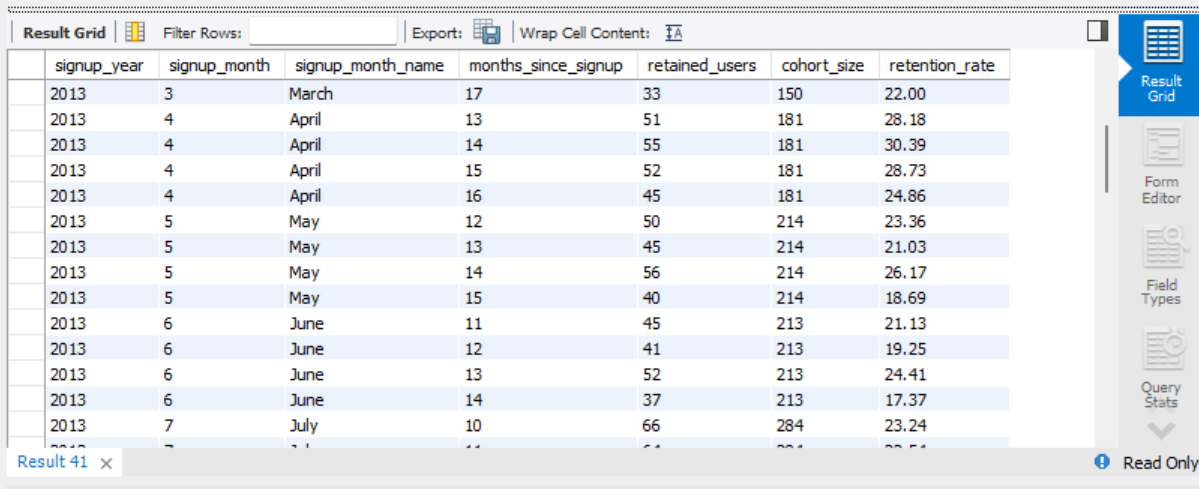
Insights- The above table shows the retention rate as per the month since signup which will help to understand when the retention rate was high.

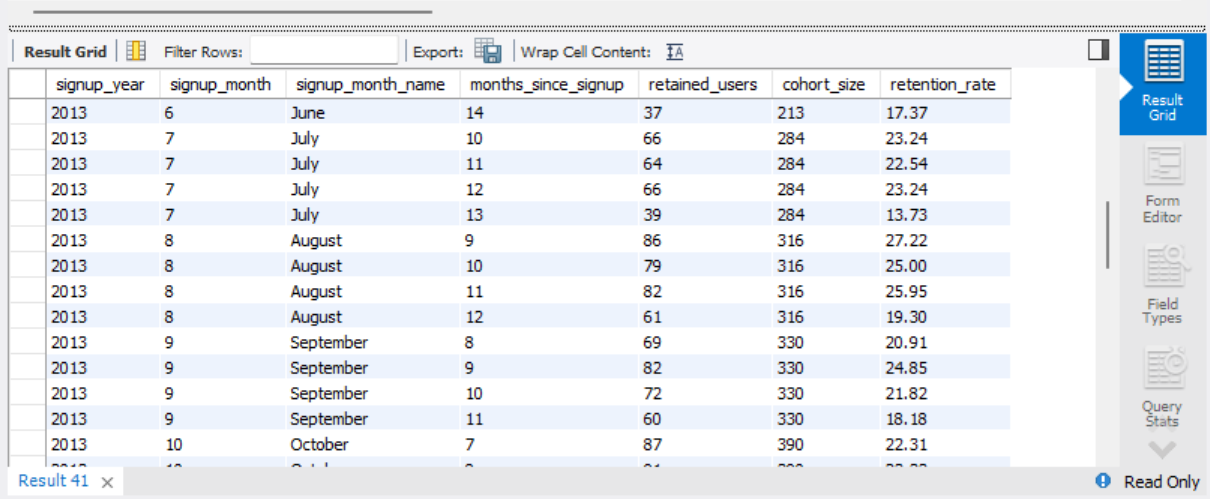
Here we have aimed to calculate retention time. By first creating subqueries to calculate cohort for tracking user signups then activity to track activities done by users and finally retention to calculate number of retained users per cohort. After this we have join the retention and cohert to obtain cohort size and then calculate retention rate which is obtained by dividing retained users by cohort size. Finally the result are then grouped and ordered by signup year and month and then months since signup.

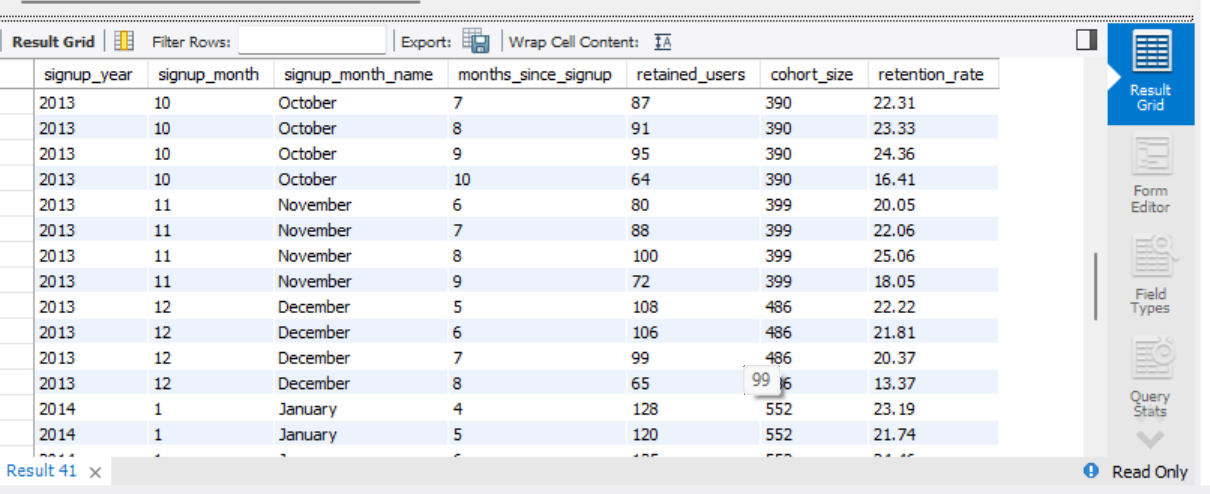


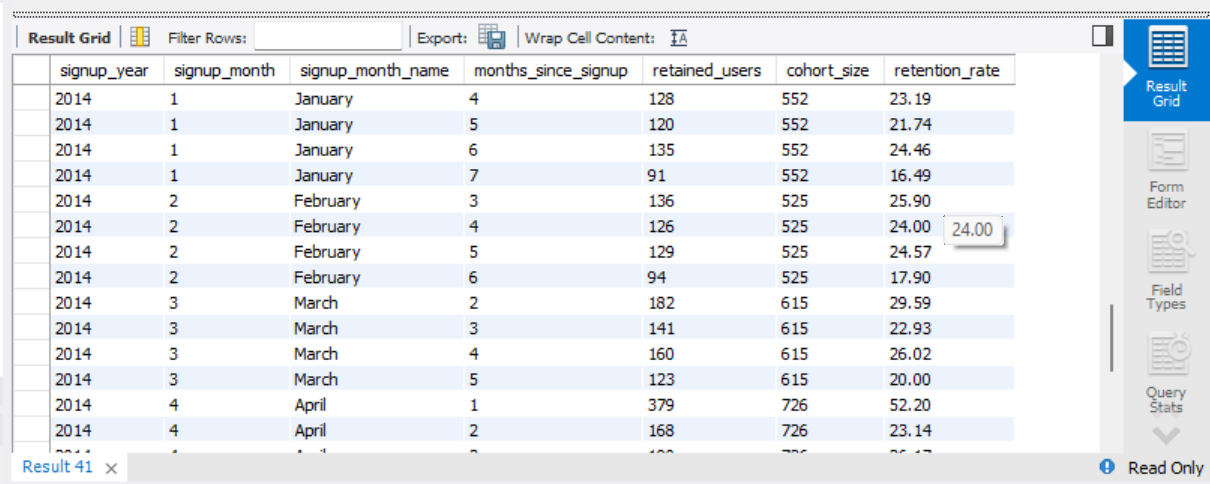








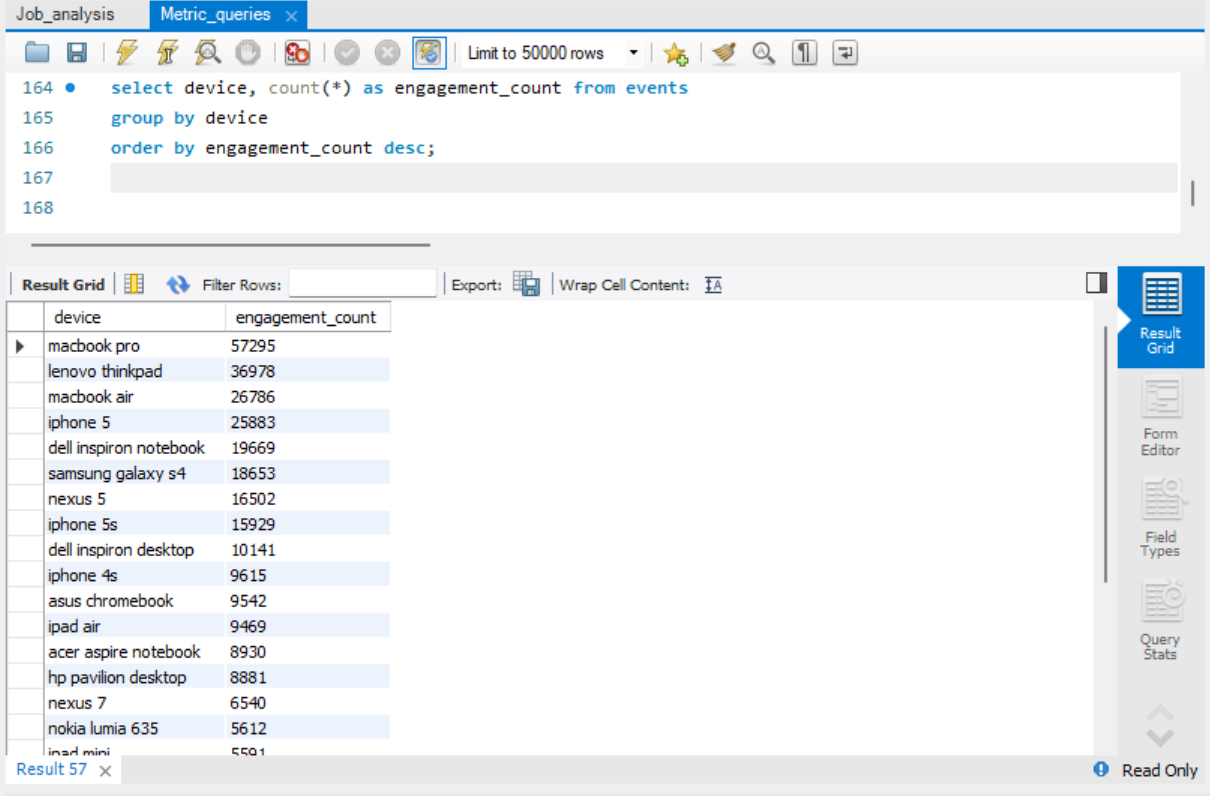




Insights- The table shows the retention rate as per the month and according to the month since sign up which helps to understand in which month and for how days until signup has retention rate.

1. Weekly engagement per device

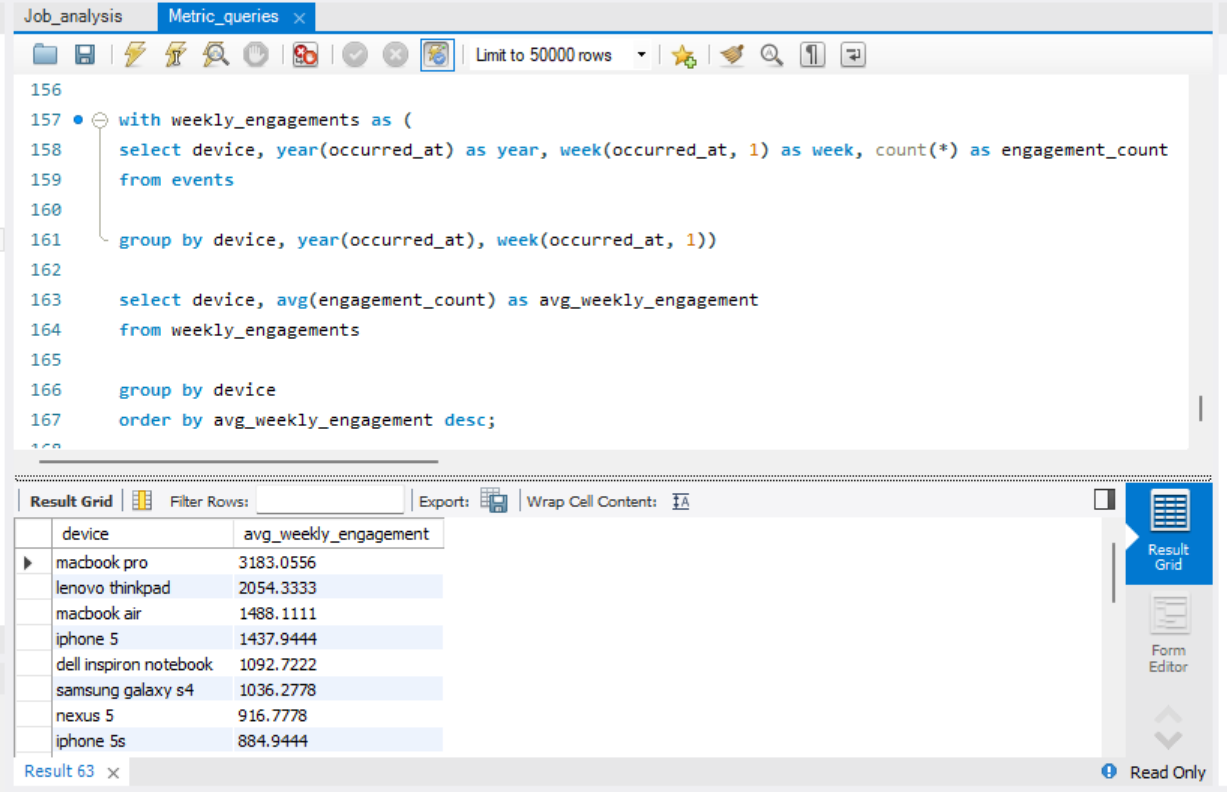
Here first to understand we have counted number of engagement events for each type of device. Then the result is grouped by device and is then ordered by engagement count to arrange in descending order to see the most used devices first.

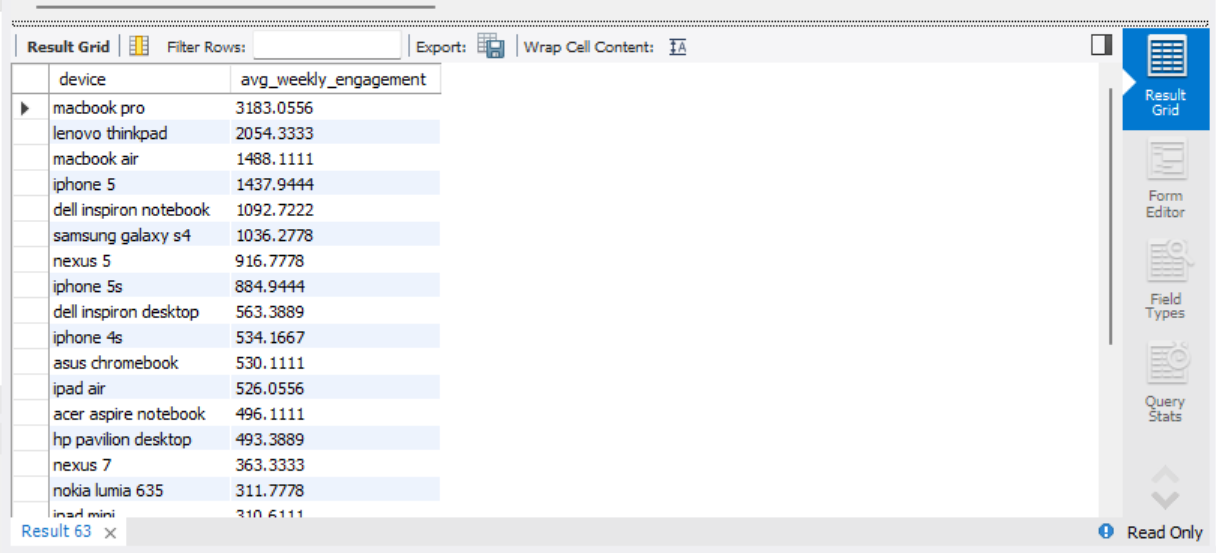
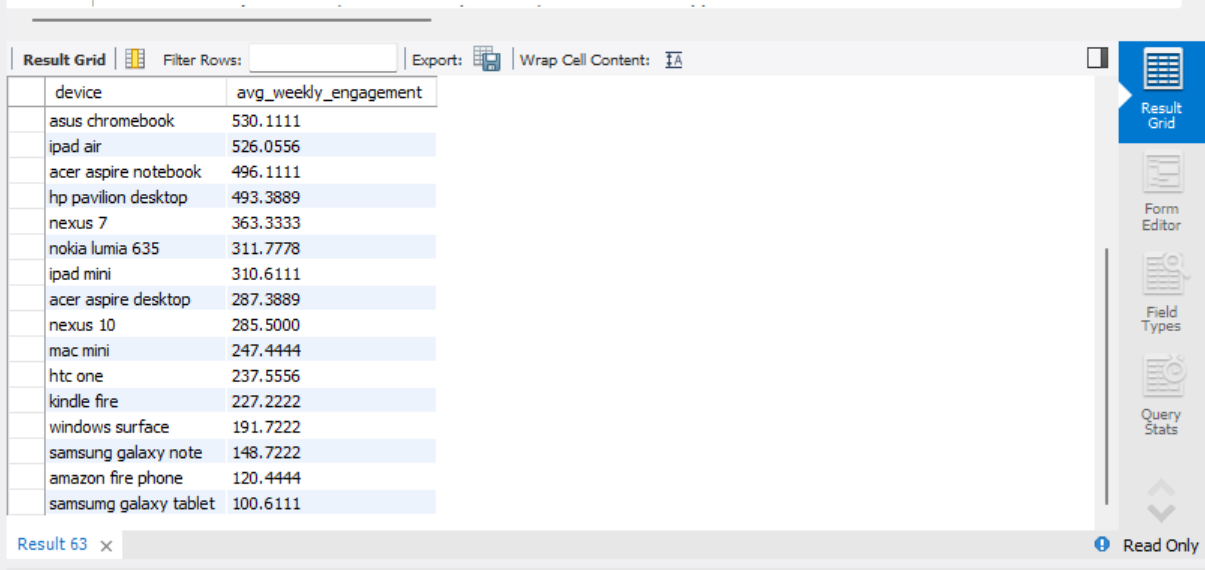




Insights- The table above table shows the total engagement count of users per device.

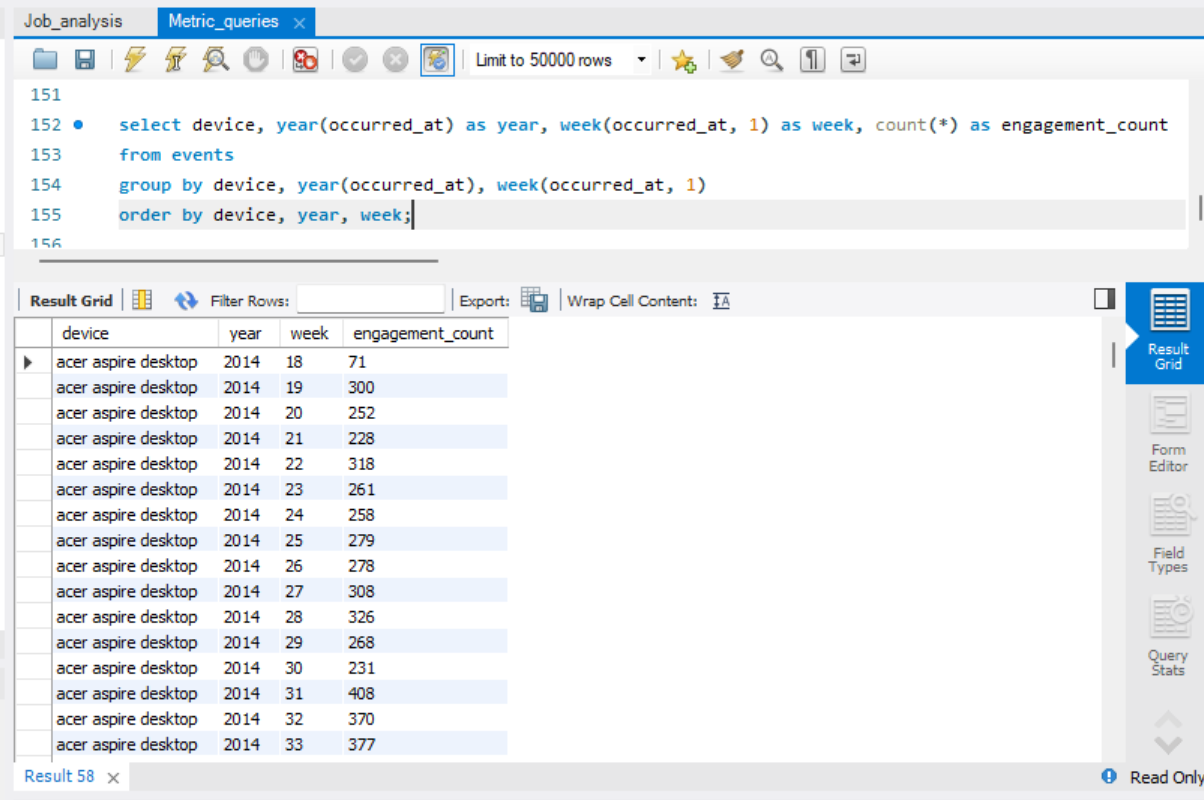
Here we have calculated average weekly engagement for every device. First we have created weekly \_engagement subquery to group events by device, year and week. It is done to count engagement. Then we have averaged out the weekly engagement counts for each device. At the end we have ordered it by average weekly engagement in descending orde

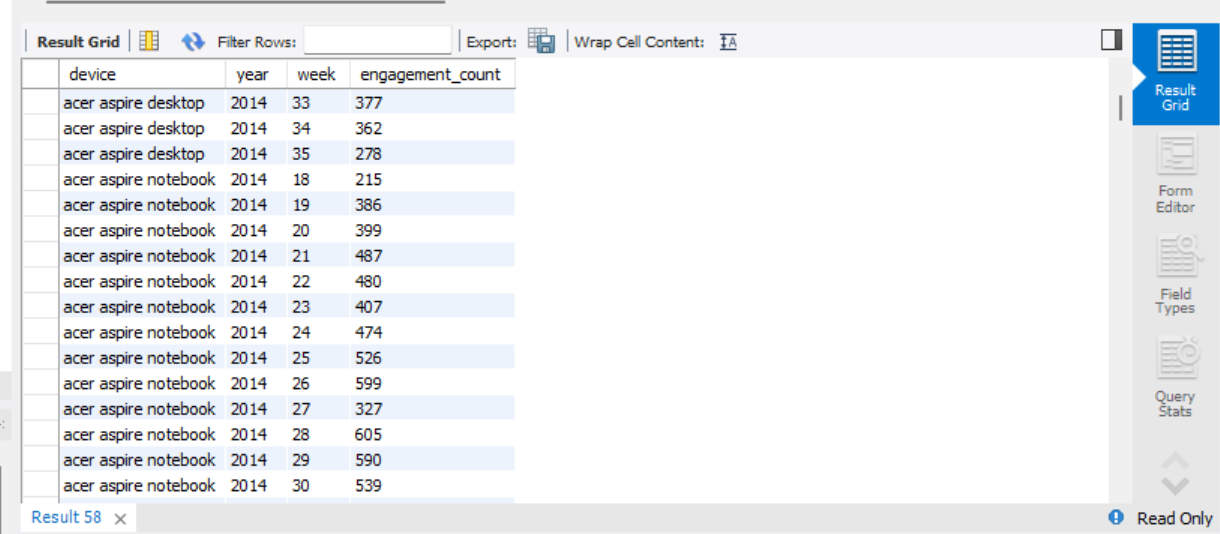


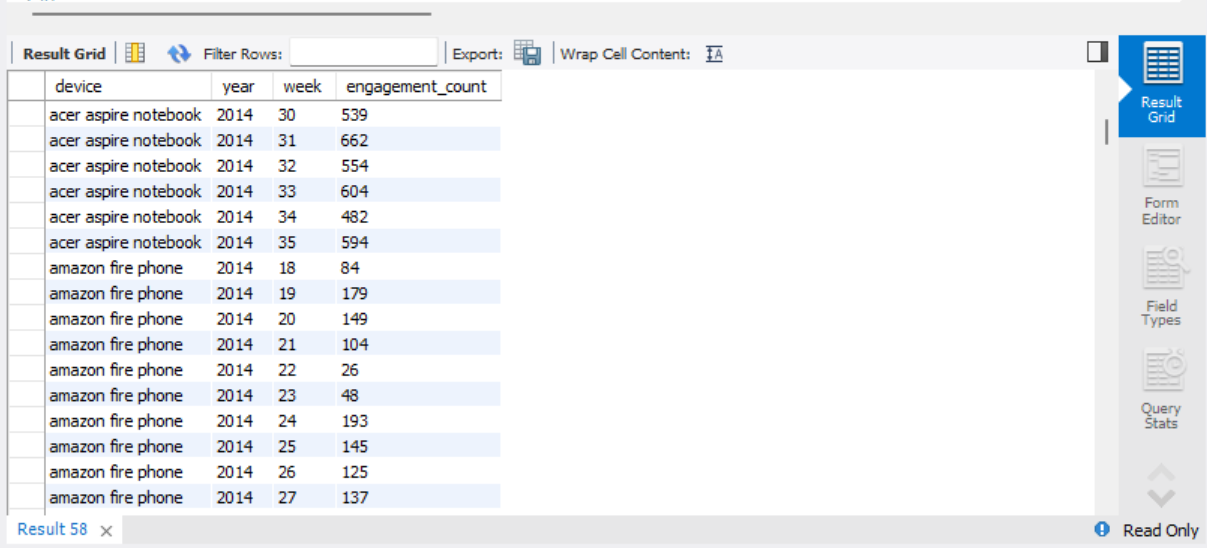
 

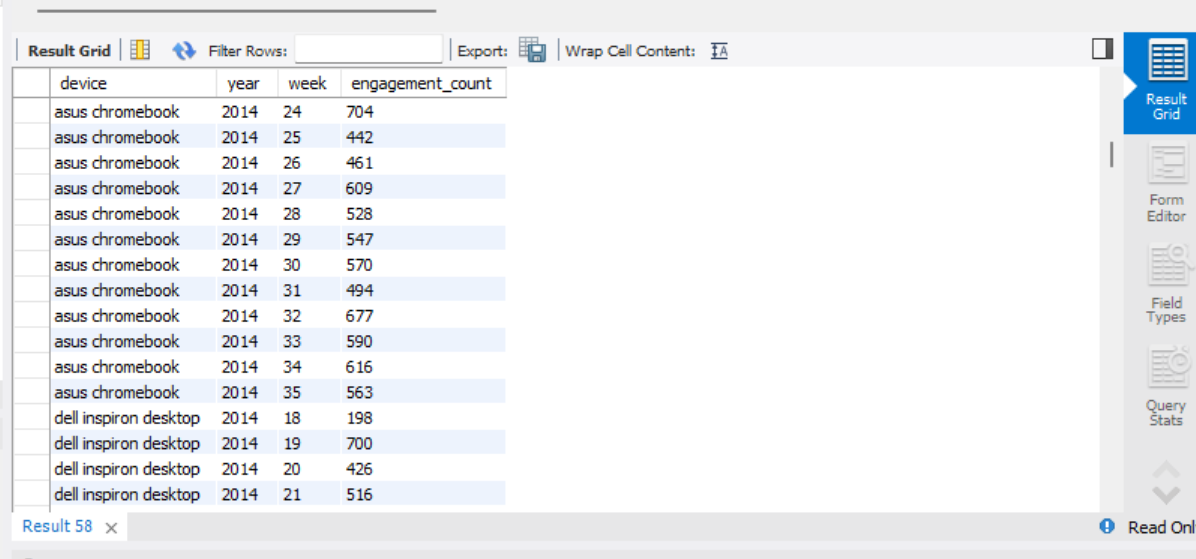
Insights- The above table shows the average user engagement per week. This will help to understand the growth and take operational decision.

In this we have calculated number of engagement events for each device on weekly basis. Then we have grouped the data by device, year and week. After that number of engagements for each group is counted. At the end the result is ordered by year, device and week to observe the trend of engagement over the time period for each device.



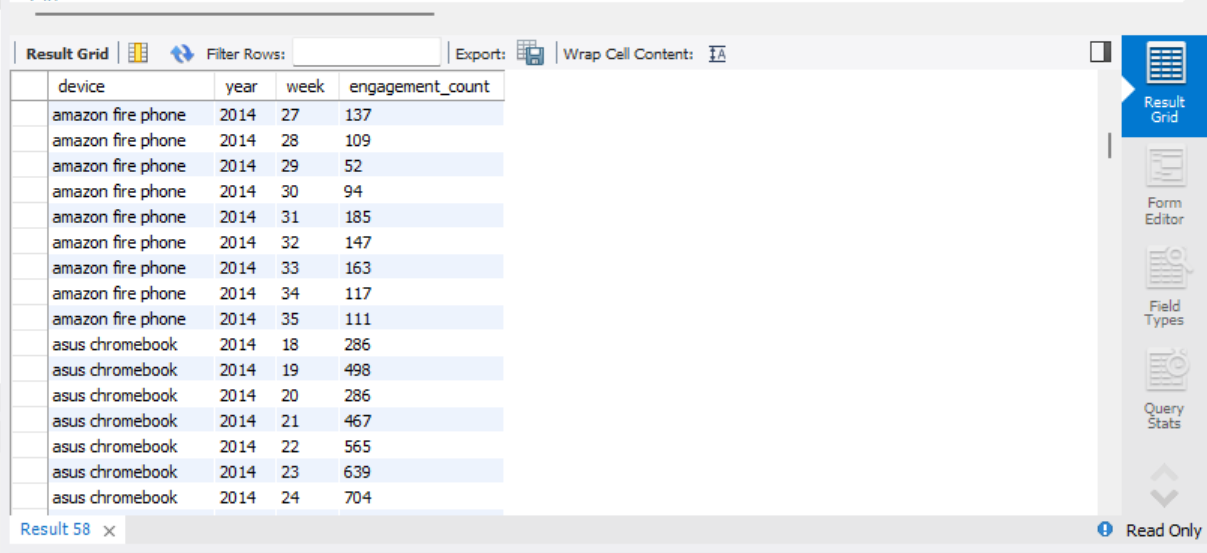


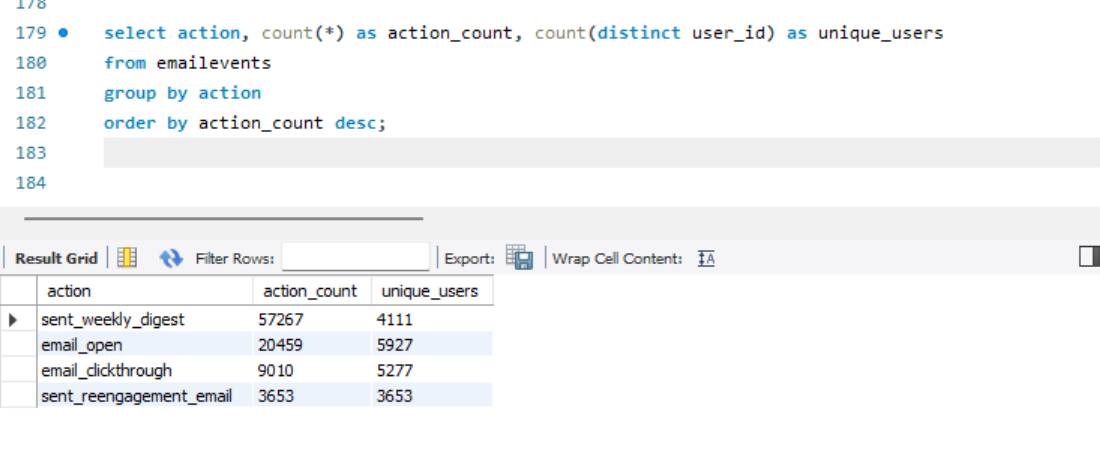




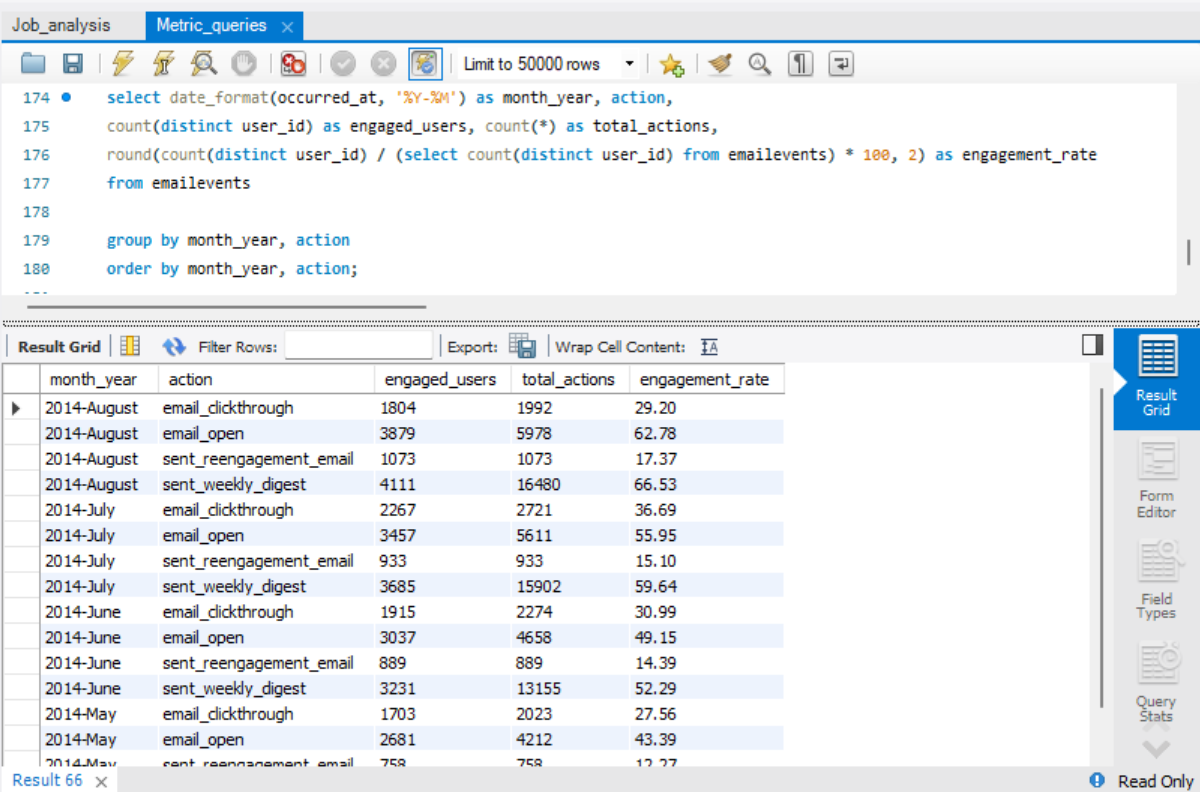
Insights- The above table shows the number of engagement count per week according to the sign up week and according to the device.

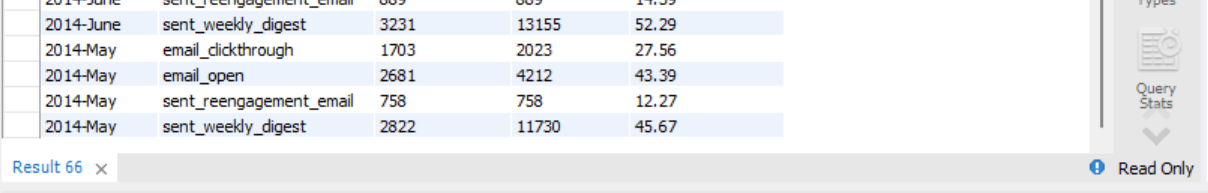
1. Email engagement analysis

In this we have examine email events to determine unique users and number of actions for each action type. Then we have counted total number of occurrences of each action as action\_count. Unique\_users are printed as distinct users performing each action. At the end the results asre grouped and ordered by action count in descending order as to see most frequent actions first in place.

Insights – The above table shows the total count of action of email as per user. The action is counted per user and the unique user. This will help to analyse the email engagement and it would be help to analyse the email performance.

Here we have analyzed user engagement of email events dataset. Here we have extracted occurred\_at for month and year. Then they are grouped by month-year and action. For each group the number of distinct engaged users, engagement rate and total actions are calculated. At and they are ordered by month year and action.





Insights- the above table shows the action and according to that total actions are given to find engagement rate is given. It helps to increase the performance of email as a platform.

**Result**

Through this project we learned to analyse the operations related to retention, user engagement, user activity and email interactions. Using MySQL workbench we found:

* User growth per month of new users
* Retention rate based on sign up cohorts weekly
* Usage of device per active user
* Average weekly activity engagement per device
* User monthly growth analysis
* Email engagement with email services

It helped to gain deeper knowledge to analyze the situation and gain more knowledge in SQL querying and relational database.