Operational Analytics And Investigating Metric Spikes

Project Description

To perform operational Analytics that involves analyzing company's end-to-end operations on daily basis. This involves understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales.

Tech Stack Used

For this project, I have used Mysql workbench 8.0.35 CE. I have chosen this because it is an easy-to-use Mysql platform. It is used to create databases and perform various SQL queries. I have also used MS Excel to change the date formats in all the sheets and MS Word to generate the report on operational analytics and then converted into PDF format for submission.

Case Study-1

Approach

After analyzing the dataset that contains information on jobs reviewed by people looking to become actor on daily basis and the other activities performed by them, I have found

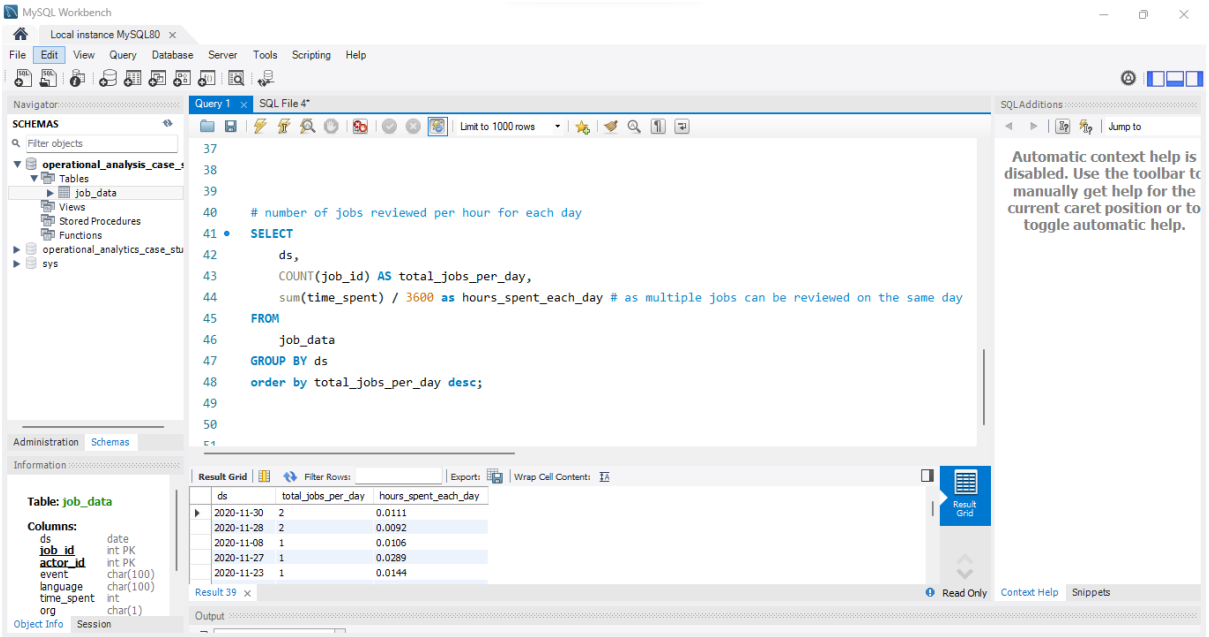
1. Number of jobs reviewed per hour for each day in November 2020

The output is :

ds	total_jobs_per_day	hours_spent_each_day
2020-11-30	2	0.0111
2020-11-28	2	0.0092
2020-11-08	1	0.0106
2020-11-27	1	0.0289
2020-11-23	1	0.0144
2020-11-17	1	0.0258
2020-11-19	1	0.0064
2020-11-15	1	0.0272
2020-11-04	1	0.0272
2020-11-18	1	0.0147
2020-11-10	1	0.0172

2020-11-21	1	0.0211
2020-11-13	1	0.0111
2020-11-14	1	0.0089
2020-11-12	1	0.0042
2020-11-22	1	0.0186
2020-11-03	1	0.0186
2020-11-24	1	0.0103
2020-11-16	1	0.0086
2020-11-25	1	0.0125
2020-11-02	1	0.0067
2020-11-09	1	0.0208
2020-11-29	1	0.0056
2020-11-26	1	0.0156
2020-11-11	1	0.0106
2020-11-05	1	0.0086
2020-11-07	1	0.0172
2020-11-20	1	0.0219

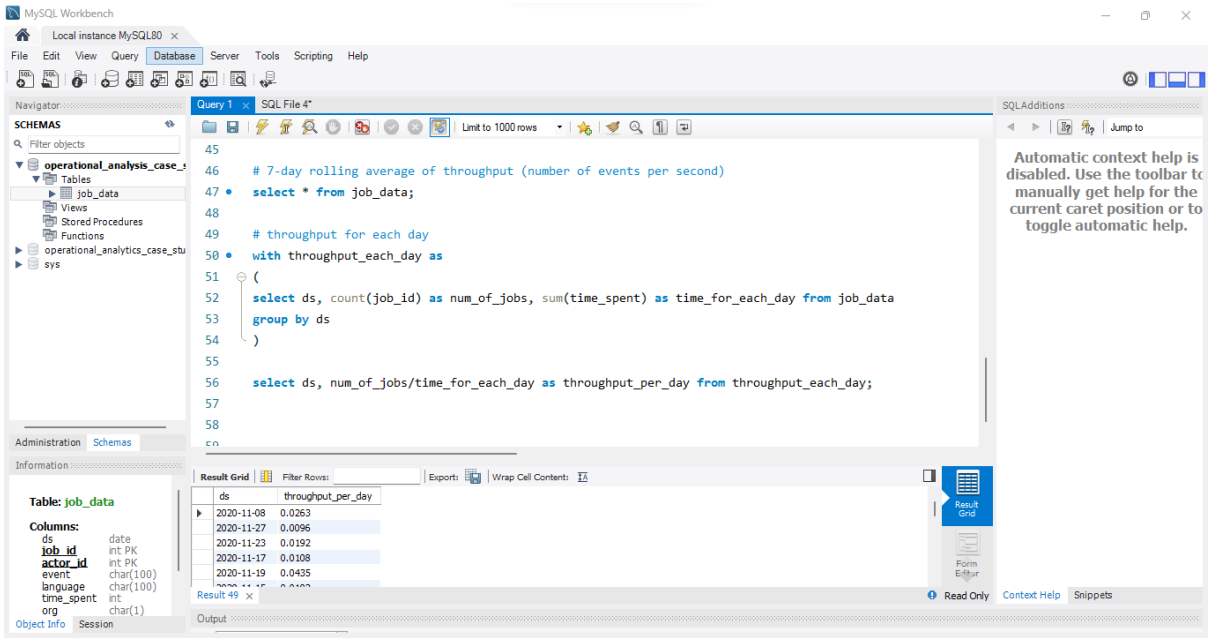
SQL Query With Output



2. 7-day rolling average of throughput (number of events per second)

After analysis, it has been observed that calculating every-day average throughput is much better than 7-day rolling average throughput as it provides a better view of fluctuations happening each day. It helps to closely monitor everything.

SQL Query With Output for each day

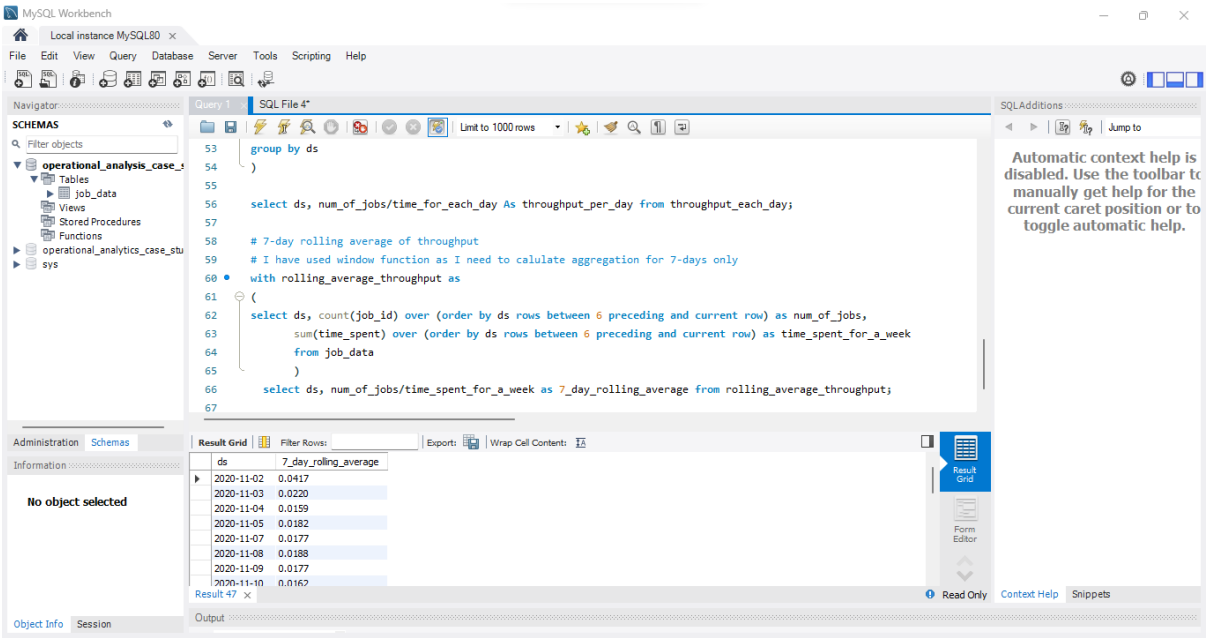


Output is:

ds	throughput_per_day
2020-11-08	0.0263
2020-11-27	0.0096
2020-11-23	0.0192
2020-11-17	0.0108
2020-11-19	0.0435
2020-11-15	0.0102
2020-11-04	0.0102
2020-11-18	0.0189
2020-11-10	0.0161
2020-11-21	0.0132
2020-11-13	0.0250
2020-11-14	0.0313
2020-11-12	0.0667
2020-11-22	0.0149
2020-11-03	0.0149

2020-11-24	0.0270
2020-11-16	0.0323
2020-11-25	0.0222
2020-11-30	0.0500
2020-11-02	0.0417
2020-11-09	0.0133
2020-11-29	0.0500
2020-11-26	0.0179
2020-11-28	0.0606
2020-11-11	0.0263
2020-11-05	0.0323
2020-11-07	0.0161
2020-11-20	0.0127

SQL Query With Output for 7-day rolling Average



Output is:

ds	7_day_rolling_average
2020-11-02	0.0417
2020-11-03	0.0220
2020-11-04	0.0159
2020-11-05	0.0182
2020-11-07	0.0177
2020-11-08	0.0188
2020-11-09	0.0177

2020-11-10	0.0162
2020-11-11	0.0173
2020-11-12	0.0218
2020-11-13	0.0212
2020-11-14	0.0233
2020-11-15	0.0194
2020-11-16	0.0222
2020-11-17	0.0202
2020-11-18	0.0193
2020-11-19	0.0189
2020-11-20	0.0171
2020-11-21	0.0155
2020-11-22	0.0166
2020-11-23	0.0158
2020-11-24	0.0181
2020-11-25	0.0185
2020-11-26	0.0170
2020-11-27	0.0160
2020-11-28	0.0183
2020-11-28	0.0214
2020-11-29	0.0237
2020-11-30	0.0256
2020-11-30	0.0277

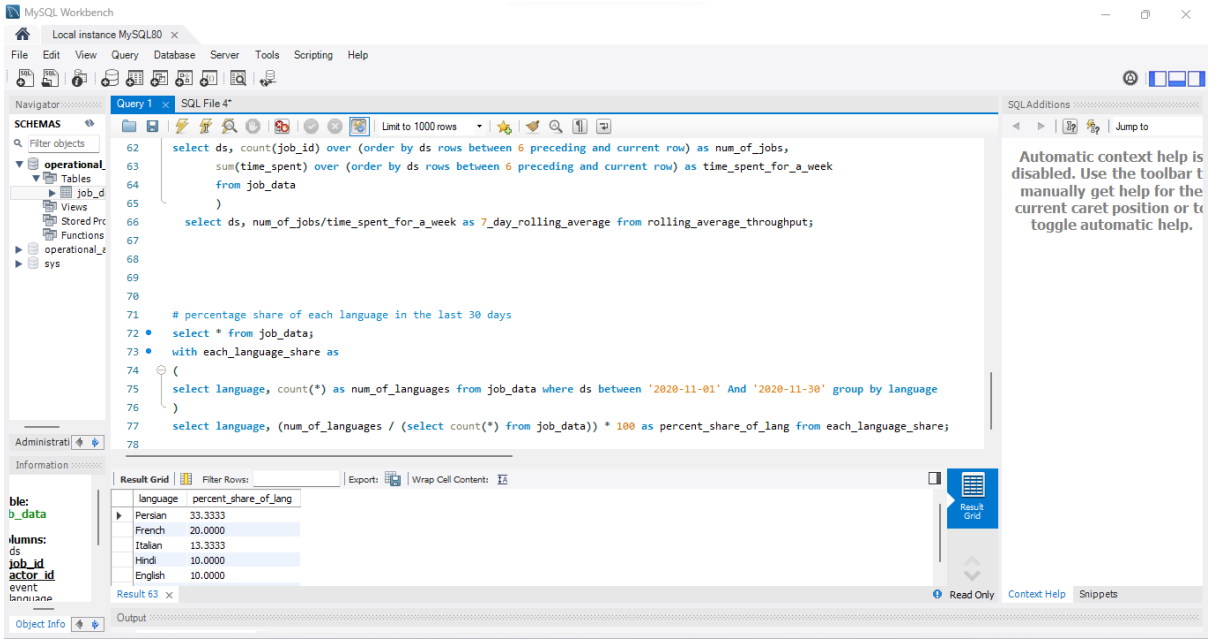
3. Percentage share of each language in the last 30 days

After analysis, it has been found that Persian is the language whose share is more than all other languages.

Output is:

language	**percent_share_of_lang**
Persian	33.3333
French	20.0000
Italian	13.3333
Arabic	13.3333
Hindi	10.0000
English	10.0000

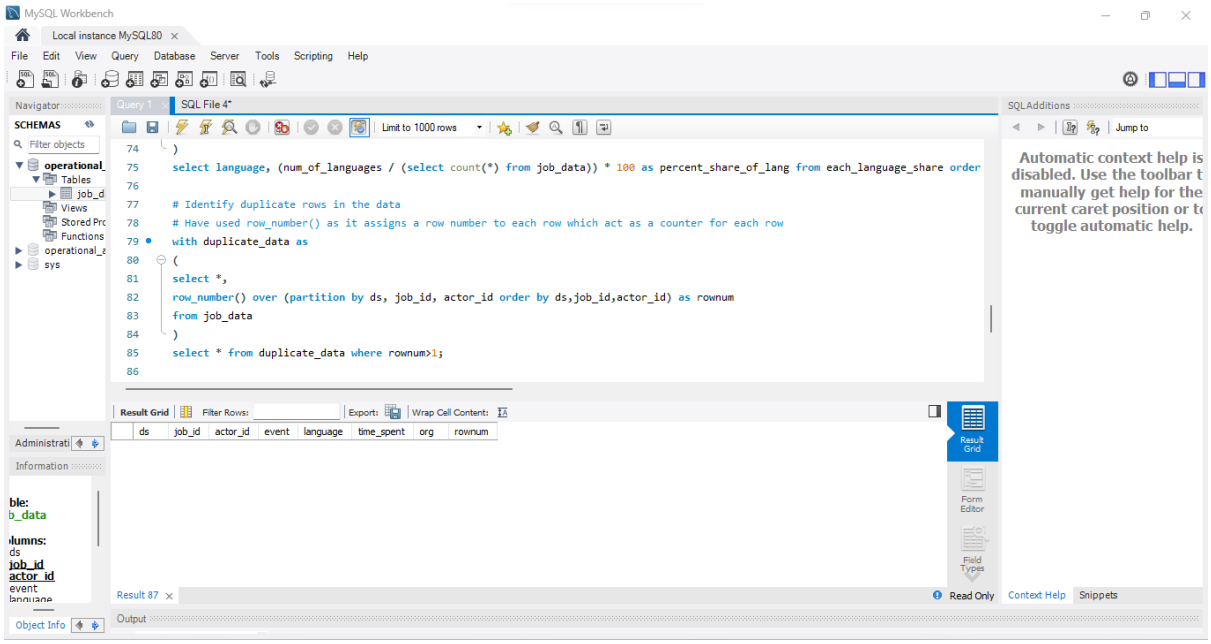
SQL Query With Output



4. Identify duplicate rows in the data

After analysis, it has been found that the given dataset doesn't contain any duplicate records. Although, each column does contain duplicate values individually. But, no duplicate rows has been found.

SQL Query With Output



Case Study – 2

Approach

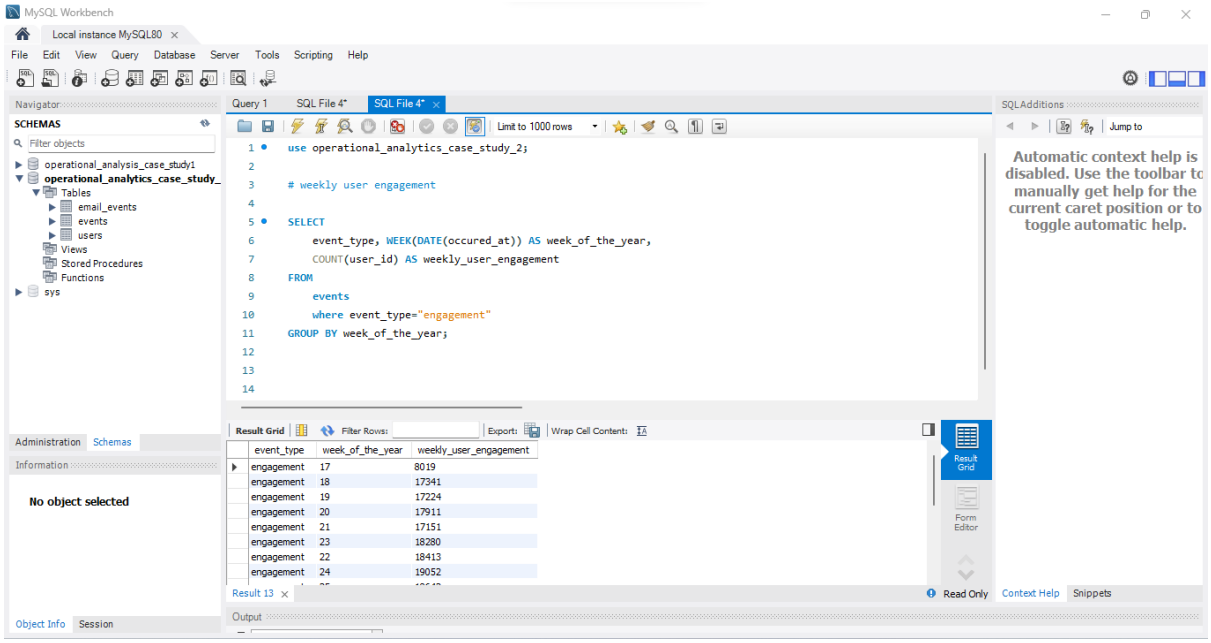
After analyzing the dataset that contains information on users and their engagement and the other activities performed by them, I have found

1. Measure the activeness of users on a weekly basis

Below is the list that shows the count of active user engagement per week. The overall user engagement per week is good.

event_type	week_of_the_year	weekly_user_engagement
engagement	17	8019
engagement	18	17341
engagement	19	17224
engagement	20	17911
engagement	21	17151
engagement	23	18280
engagement	22	18413
engagement	24	19052
engagement	25	18642
engagement	29	20067
engagement	26	19061
engagement	30	21533
engagement	28	20776
engagement	27	19881
engagement	31	18556
engagement	32	16612
engagement	33	16145
engagement	34	16127
engagement	35	784

SQL Query With Output



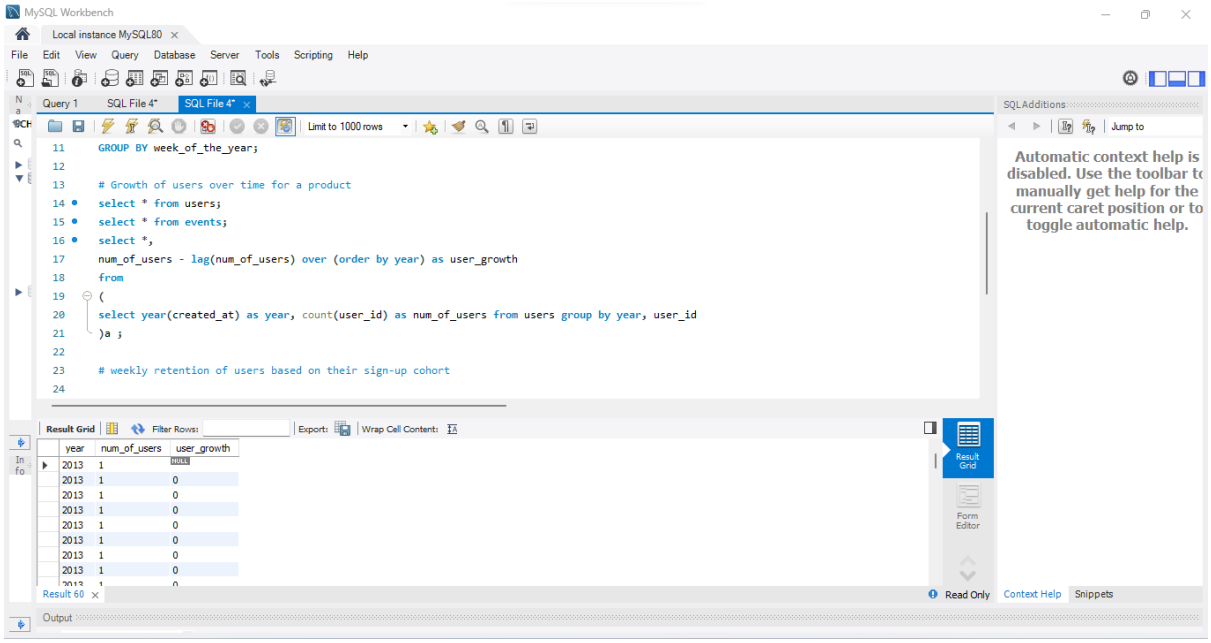
2. Growth of users over time for a product

Below are results showing the number of users joined each year. As, it is clear that number of users joined in 2014 are more than number of users in 2013 which means that more users have engaged with the product.

year	num_of_users	user_growth
2013	1	NULL
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0
2013	1	0

Result 60

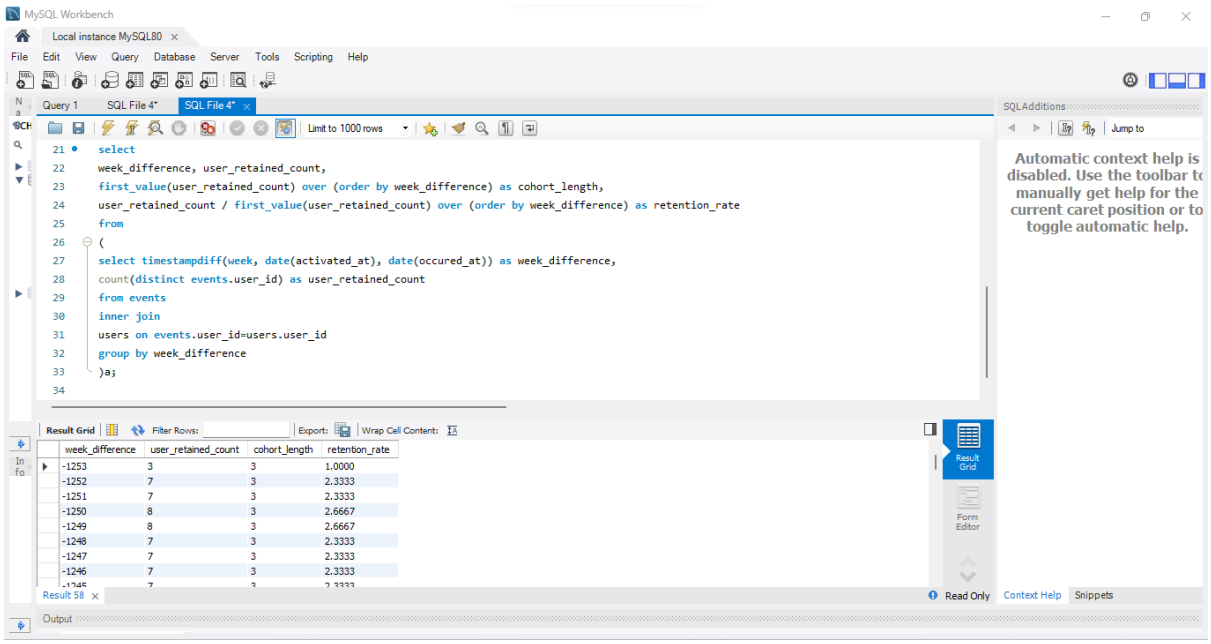
SQL Query With Output



3. Retention of users on a weekly basis after signing up for a product

Here we calculated Cohort Analysis which is an analysis of multiple cohorts or group of people with the objective of getting a deeper understanding of user behaviors, market trends, etc. It is a method of analyzing a metric by comparing its behavior between different groups of users.

SQL Query With Output



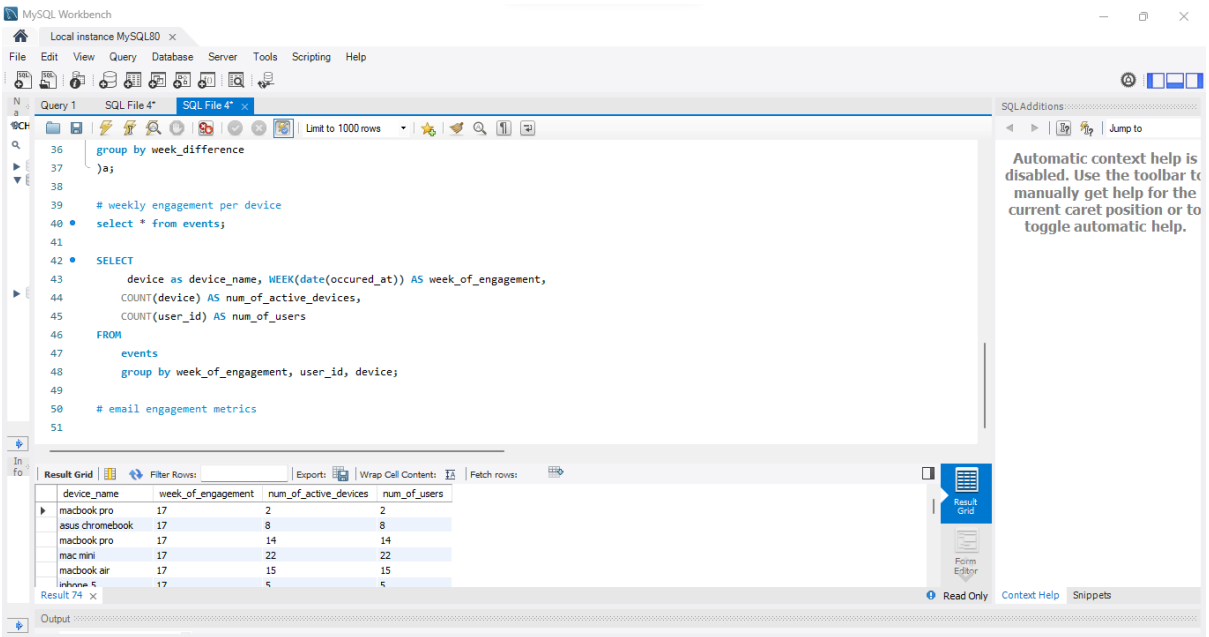
4. Measure the activeness of users on a weekly basis per device

Below is the list of devices with number of users using them weekly. User engagement seems good.

device_name	week_of_engagement	num_of_active_devices	num_of_users
macbook pro	17	2	2
asus chromebook	17	8	8
macbook pro	17	14	14
mac mini	17	22	22
macbook air	17	15	15
iphone 5	17	5	5
nexus 10	17	4	4
ipad air	17	31	31
lenovo thinkpad	17	8	8
hp pavilion desktop	17	9	9
lenovo thinkpad	17	14	14
macbook pro	17	4	4
iphone 5	17	16	16
dell inspiron noteb...	17	5	5
macbook pro	17	5	5
iphone 5	17	7	7
samsung galaxy s4	17	8	8
macbook air	17	3	3
samsung galaxy s4	17	5	5
asus chromebook	17	28	28

Result 74

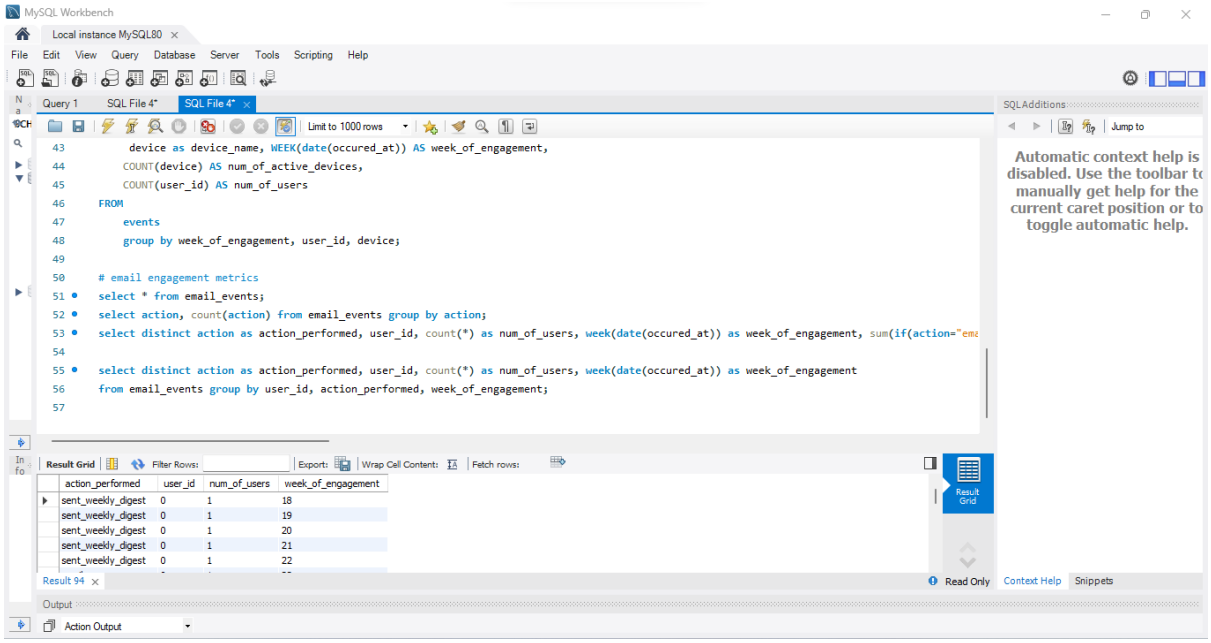
SQL Query With Output

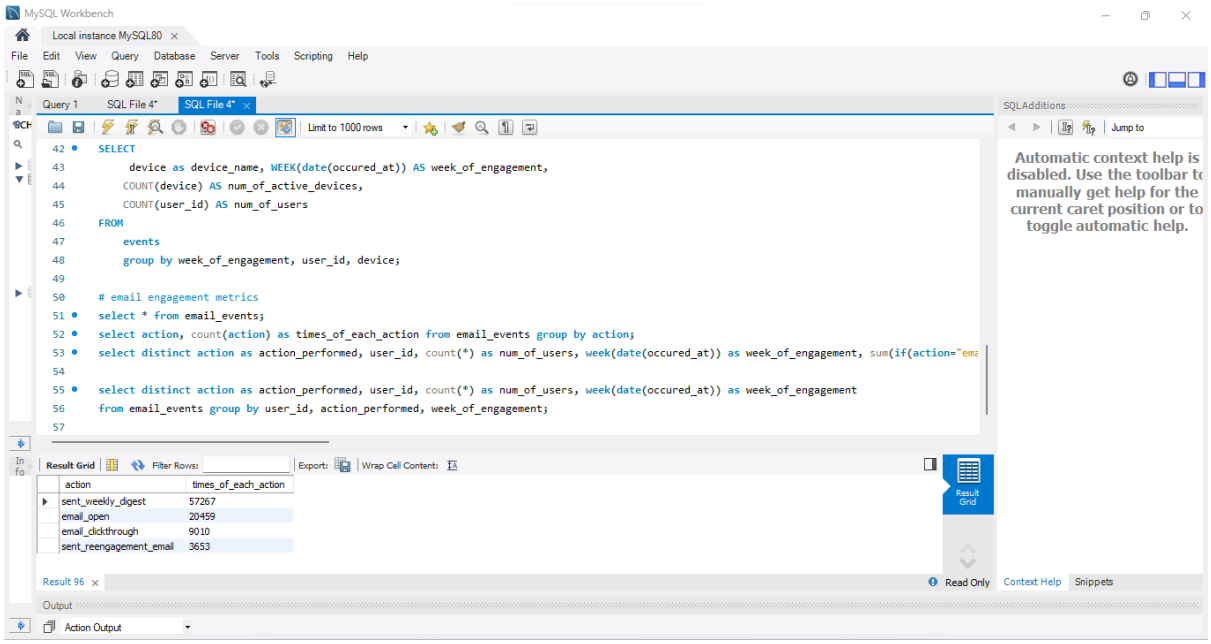


5. How users are engaging with the email service

Below is the list of users and the action taken by them weekly along with number of times each action is performed. Overall email engagements seems to be increasing.

SQL Query With Output





Result

This project is really engaging and at the same time difficult too. It not only level up my understanding about window functions in SQL but also introduced me to terms like cohort analysis, rolling average. It does gave me the confidence in using window functions for future.

Note:

It’s a request to please include a explaination of the each task so that it becomes clear that what insight we should be looking for in each task.