OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKES

PROJECT DESCRIPTION:

This project will help us to analyse different the operational function in a company. This project deals with the topic Operation analytics under which the investigating metric spikes belongs. **Operation Analytics** provides real-time insights and in-depth analysis to optimize your business operations. This will help to make data-driven decisions effortlessly. In a company, the operation analytics will help them to analyse the day-to-day operations within the company so that they can make better decisions through the insights they get from the analysis. The main key aspect of the operation analytics is **Investigating metric spikes,** which will help to understand and explain the sudden changes in key metrics such as a high drop in the sale, or spike in the number of users etc.

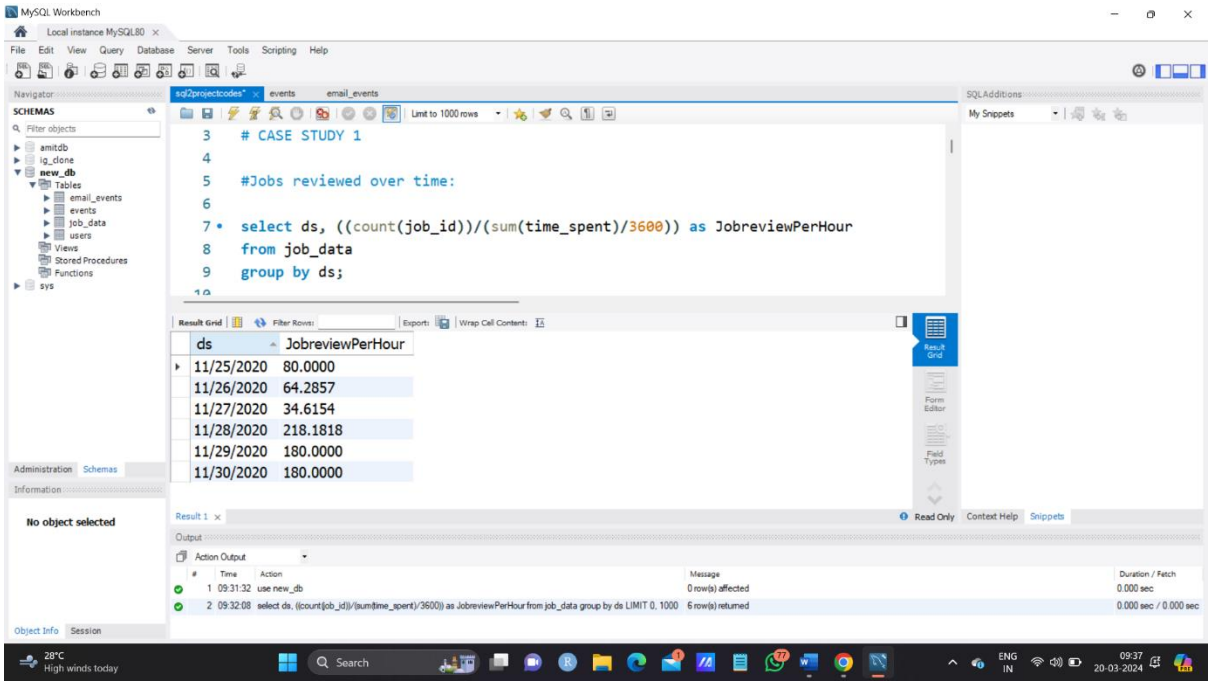
In this project we have two case studies, the first one is **Job data analysis** and the second one is **Investigating metric spike.** We will use SQL queries to find out different insights from the data according to the needs specified in the questions.

APPROACH:

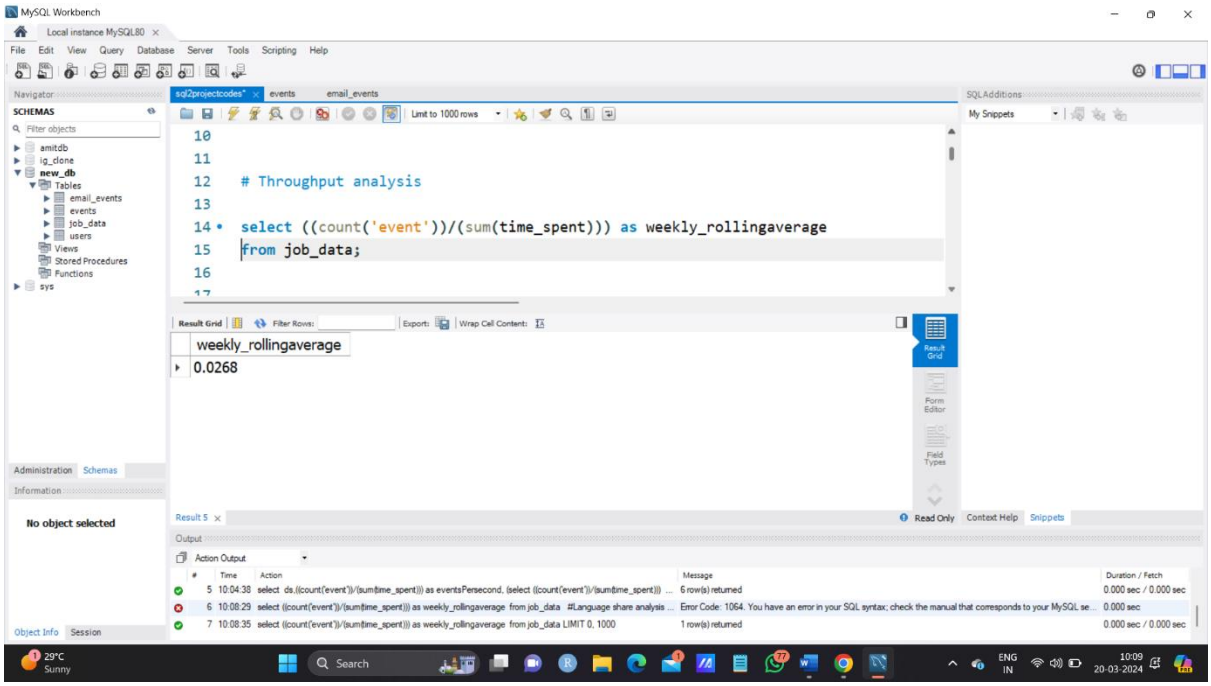
We first imported the data which is in csv file to the SQL workbench and then used the SQL workbench and SQL Queries to gain valuable insights from the different data that are given. The following are the steps that I took to analyse the data and for finding answers to the questions:

CASE STUDY 1: JOB DATA ANALYSIS

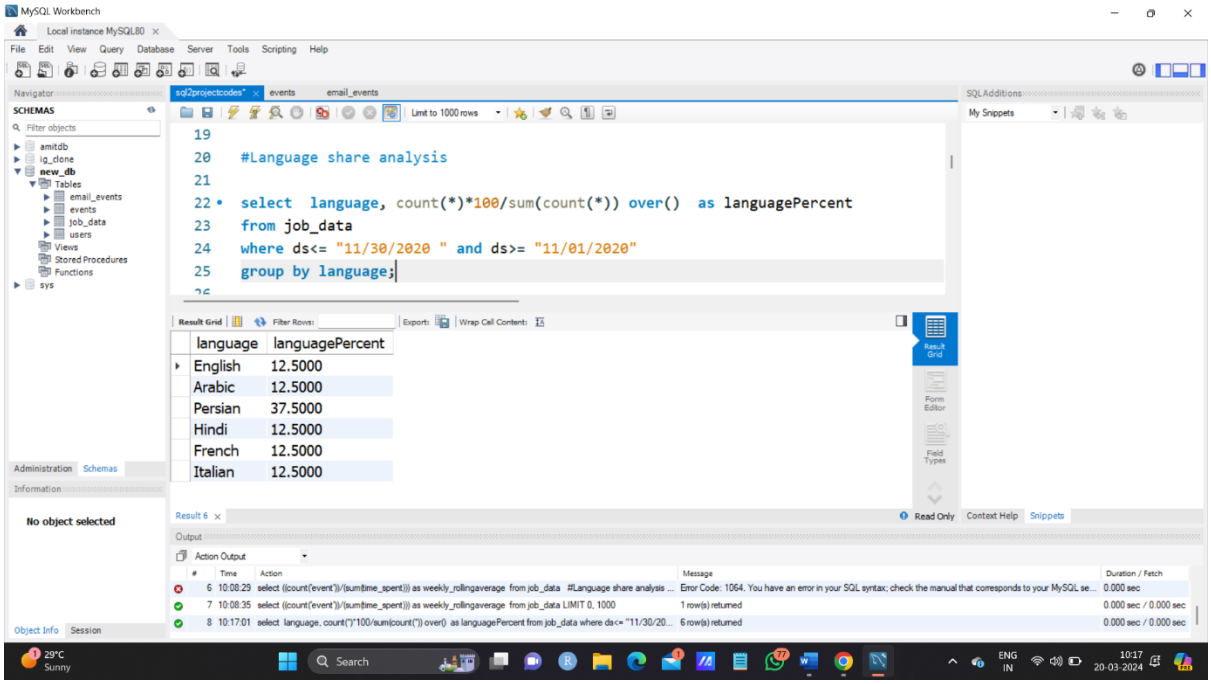
A) **Jobs Reviewed Over Time:** Here I write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020 from the given dataset job_data.



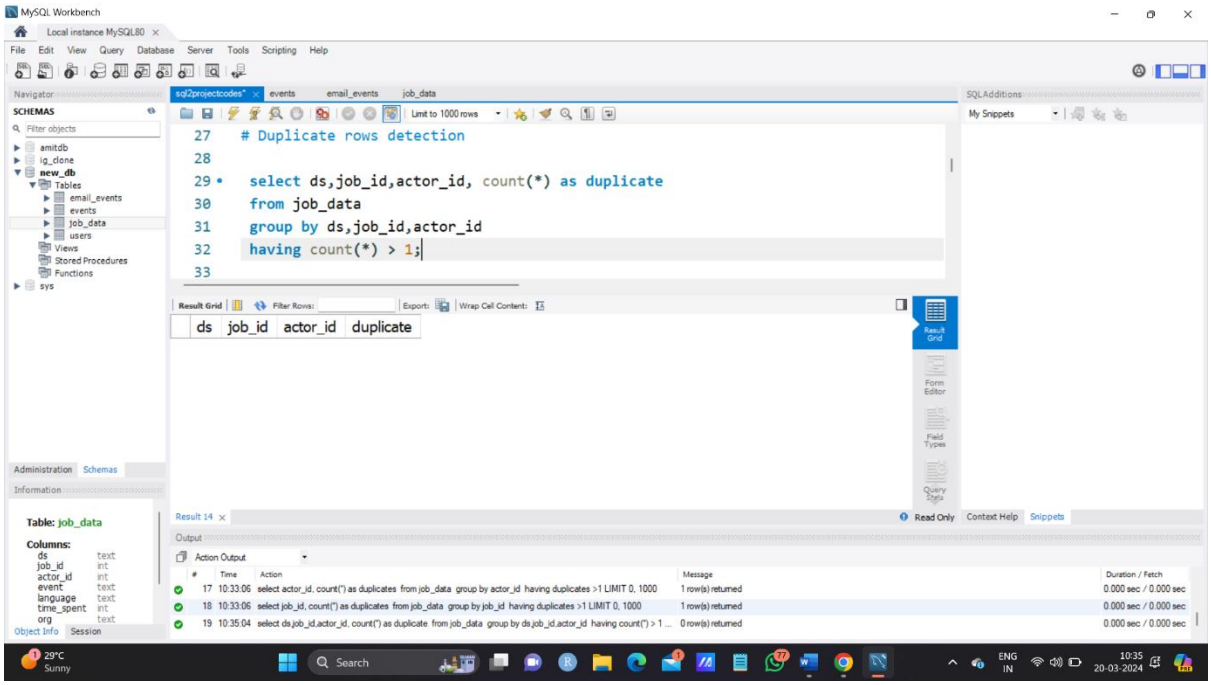
B) **Throughput Analysis:** Here I write an SQL query to calculate the 7-day rolling average of throughput (number of events per second).

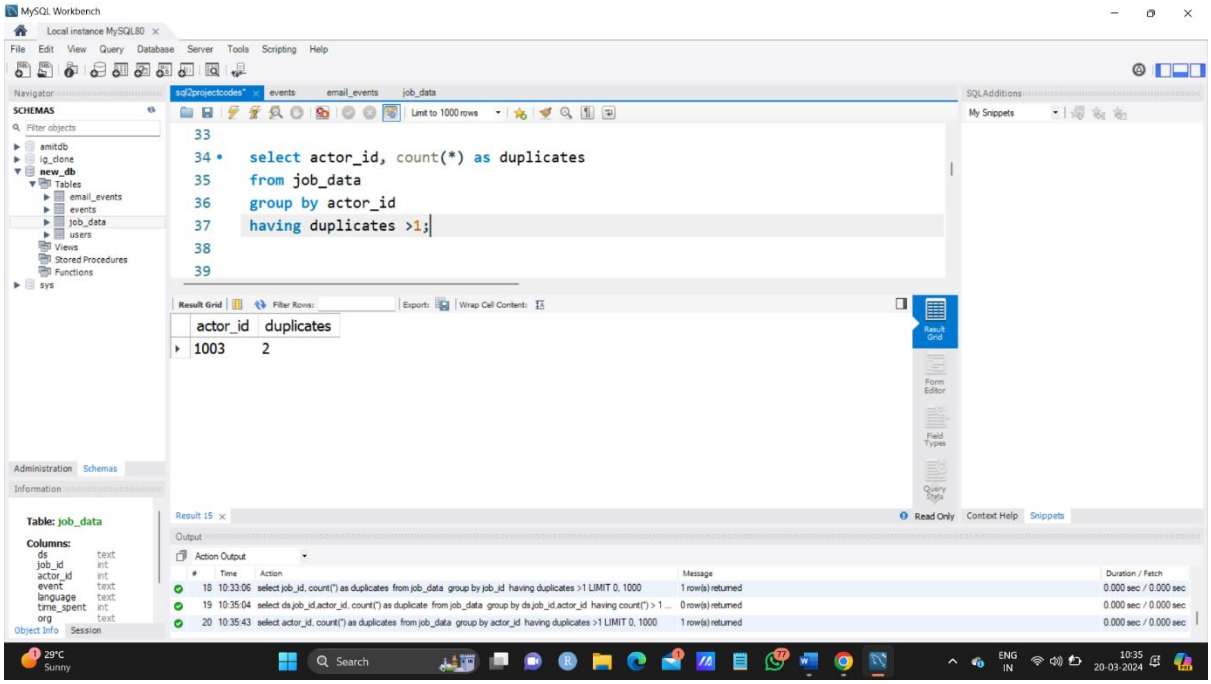


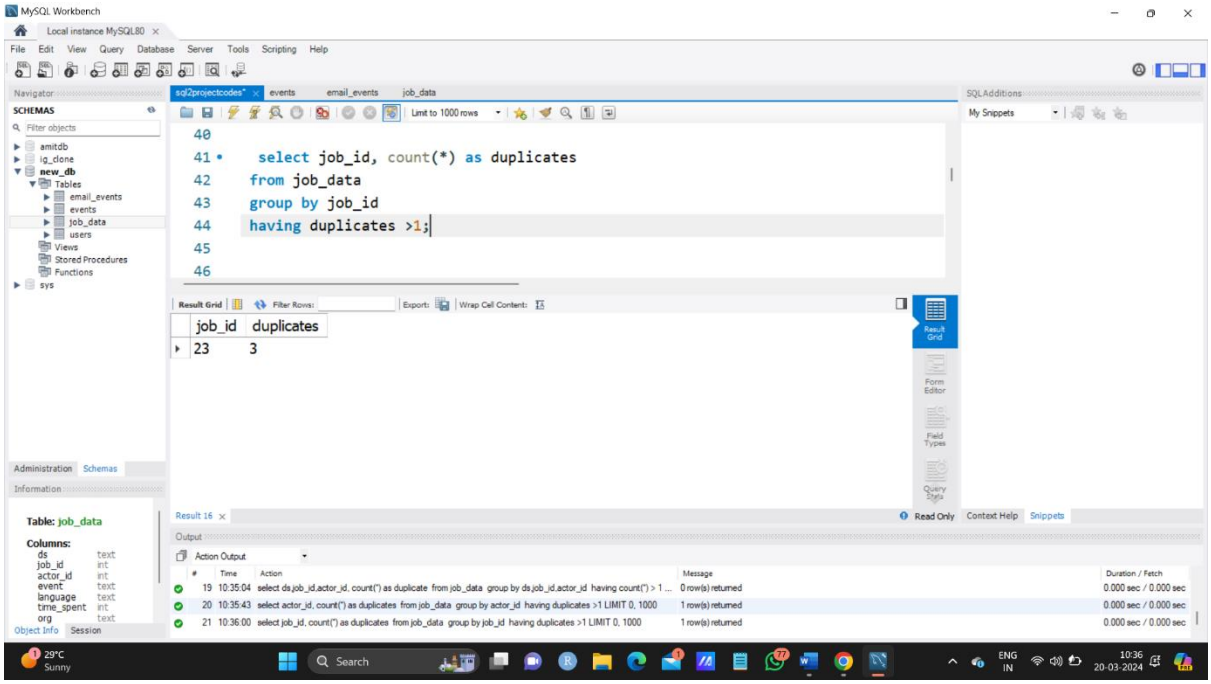
C) **Language Share Analysis:** Here I write an SQL query to calculate the percentage share of each language over the last 30 days.



D) **Duplicate Rows Detection:** Here I write an SQL query to display duplicate rows from the job_data table.



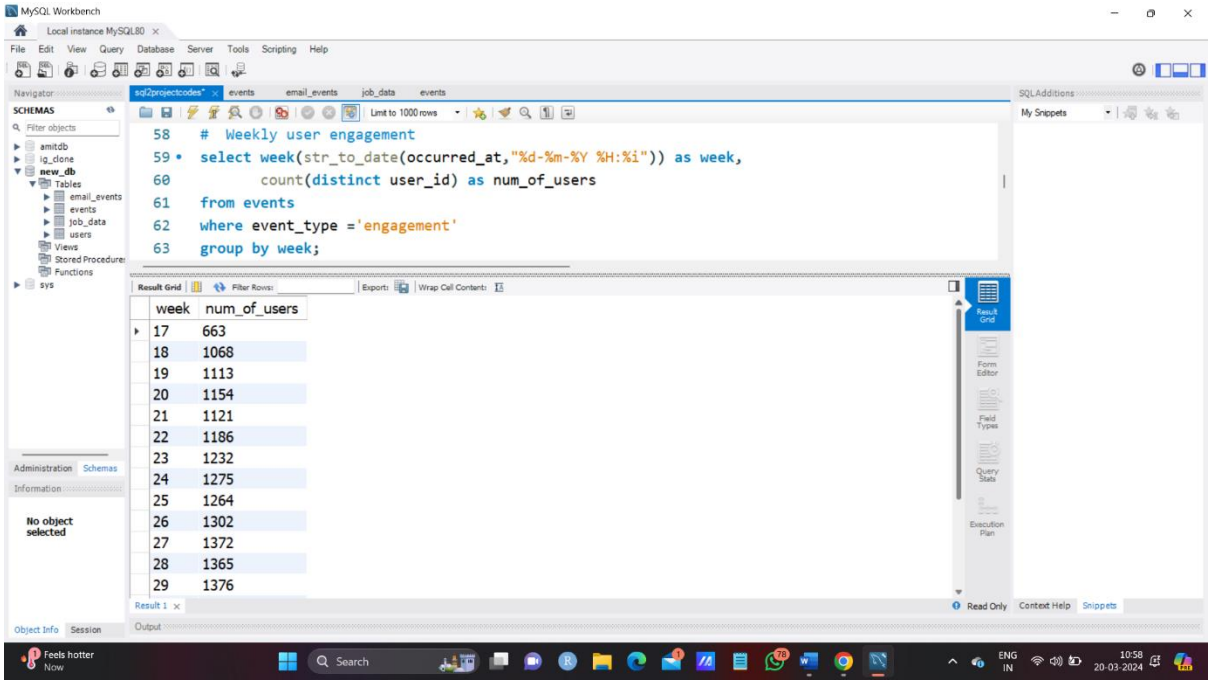


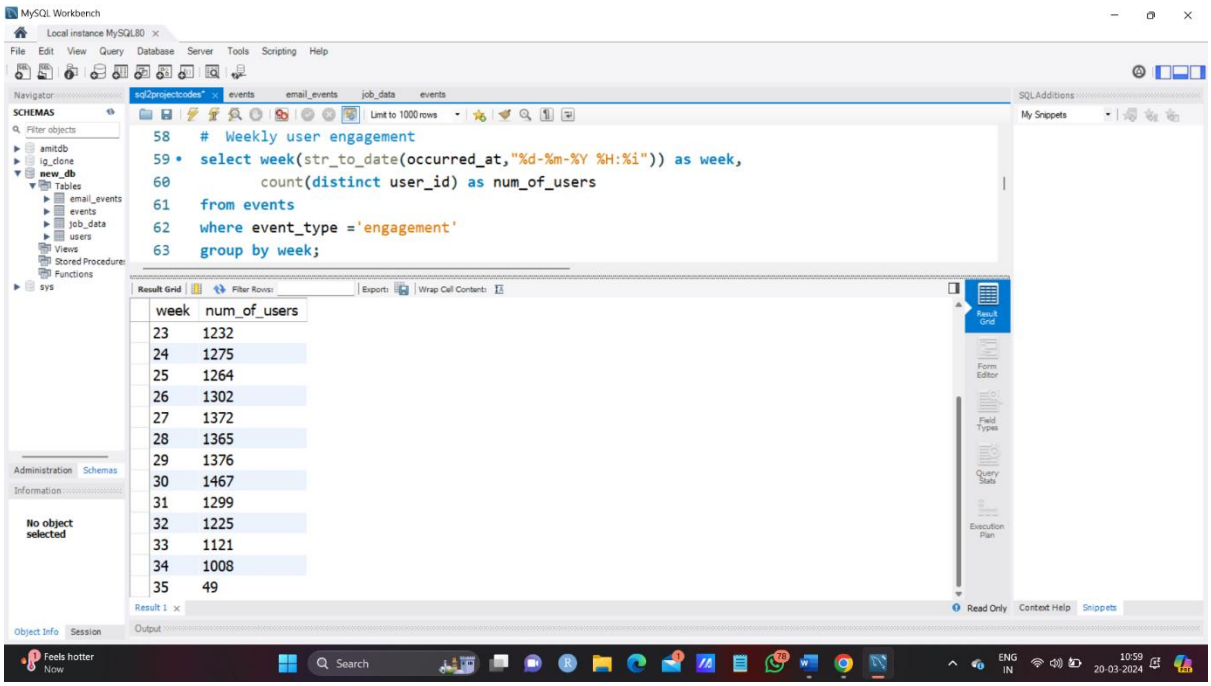


CASE STUDY 2: INVESTIGATING METRIC SPIKES

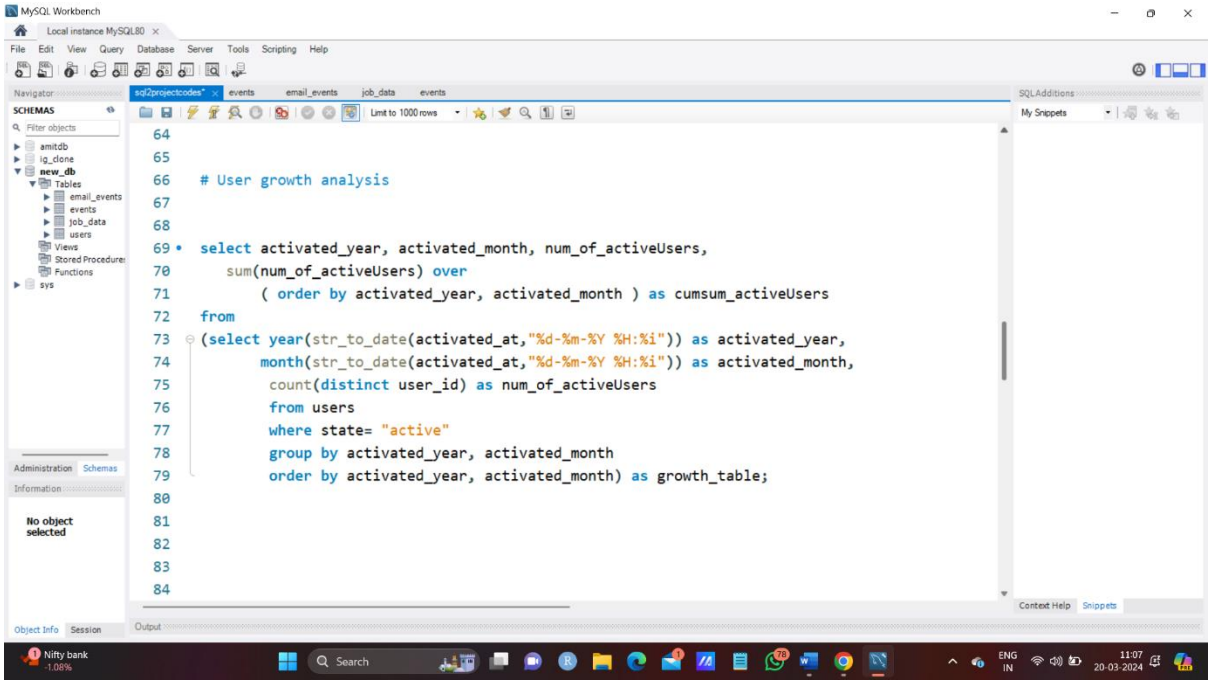
Here I am working with three Data sets that are related to each other to get insights from the data.

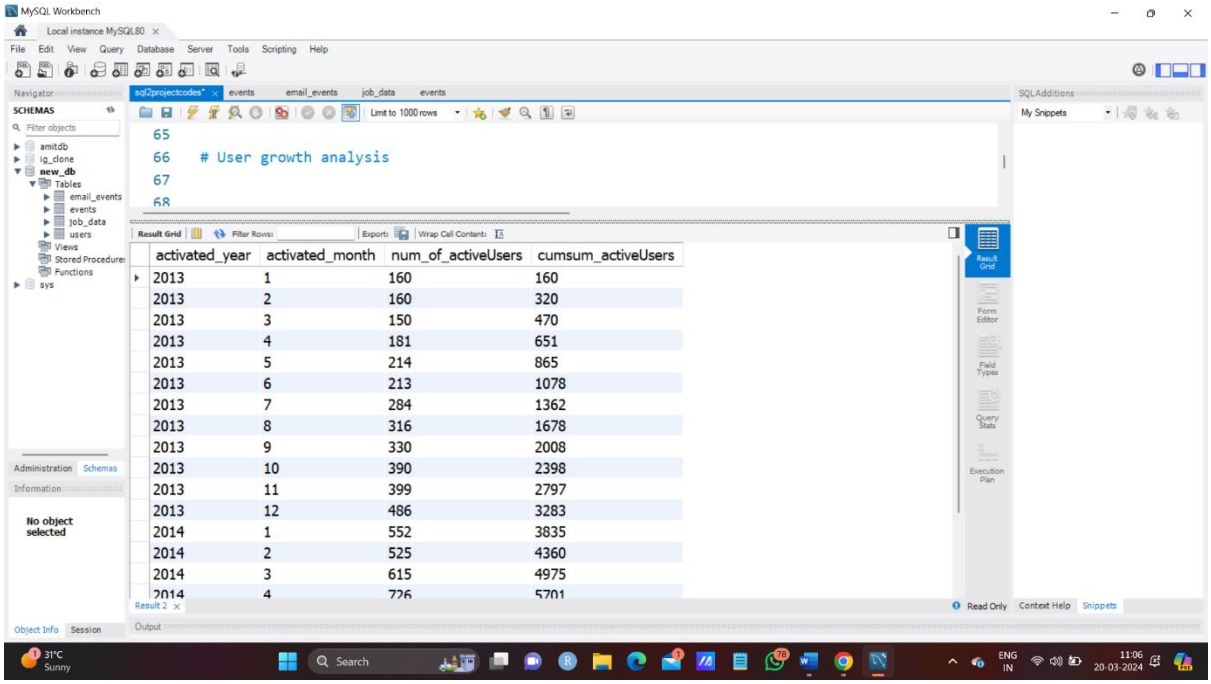
A) **Weekly User Engagement:** Here I write an SQL query to calculate the weekly user engagement.

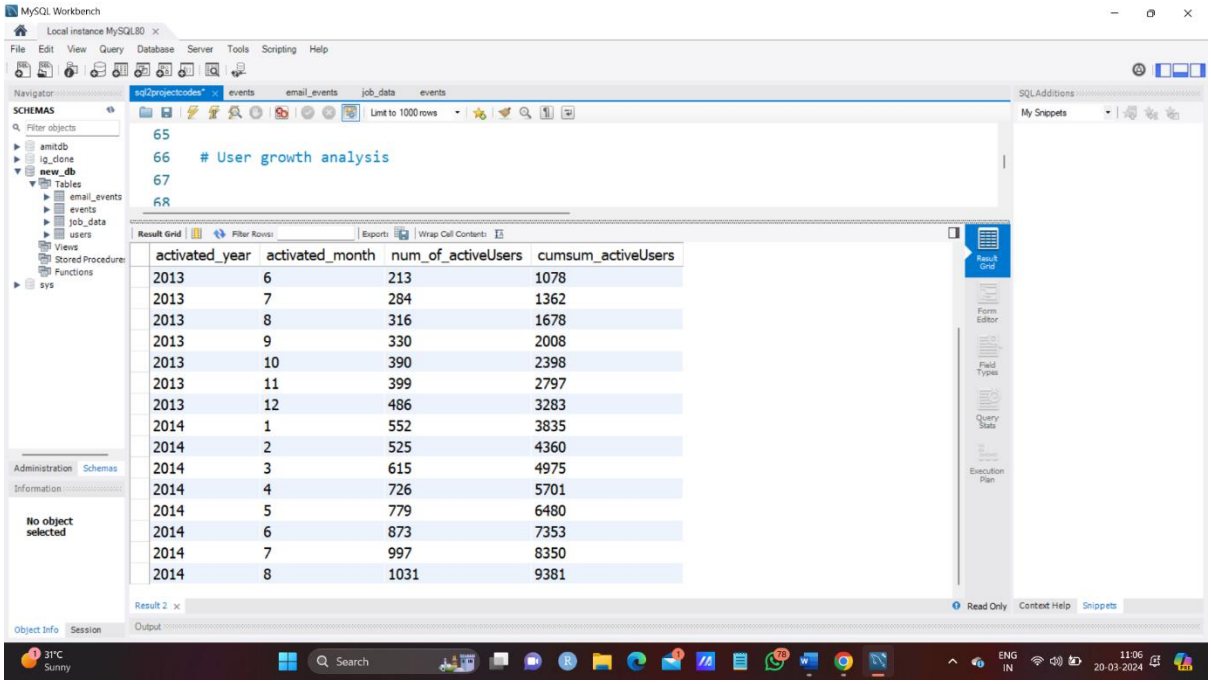




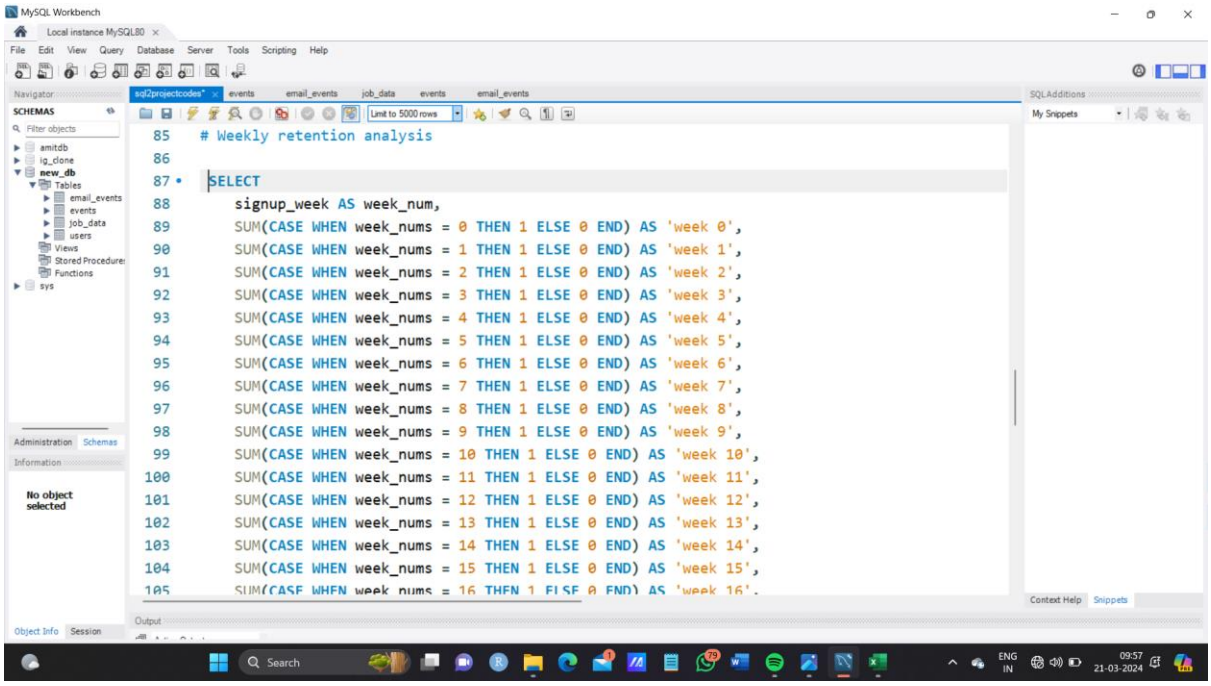
B) **User Growth Analysis:** Here I write an SQL query to calculate the user growth for the product over every month.

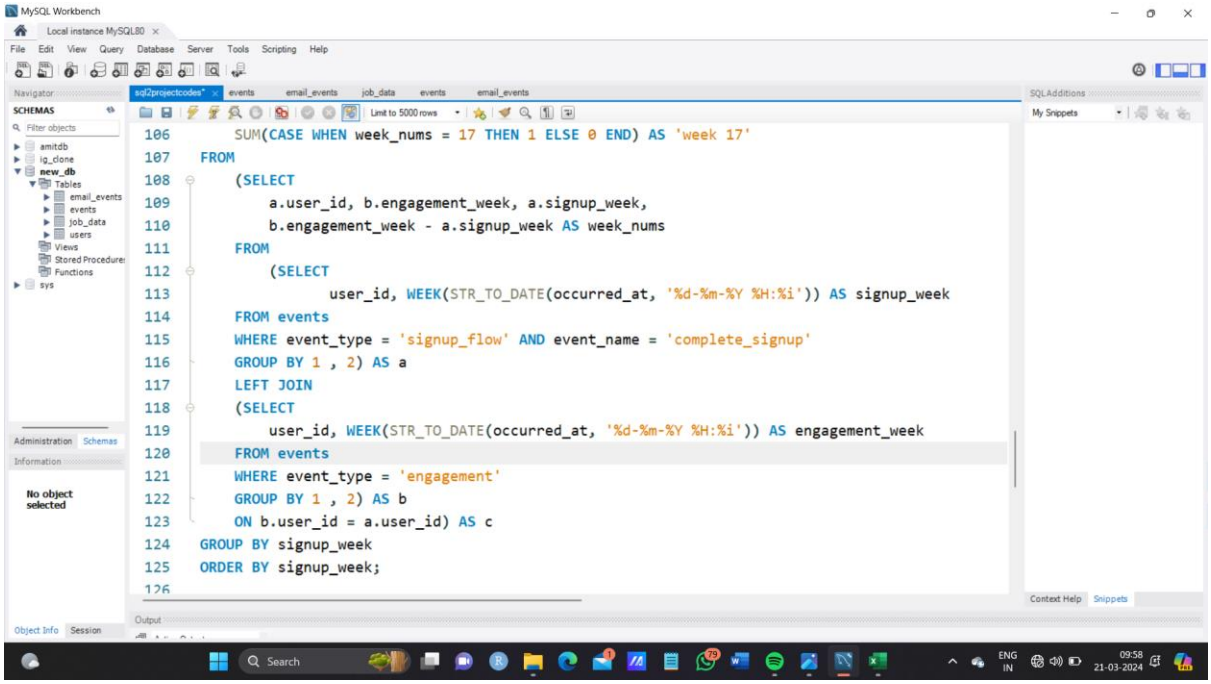

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
amitdb
ig_clone
new_db
Tables
email_events
events
job_data
users
Views
Stored Procedures
Functions
sys
sql2projectcodes*
events
email_events
job_data
events
Limit to 1000 rows
64
65
66 # User growth analysis
67
68
69 • select activated_year, activated_month, num_of_activeUsers,
70 sum(num_of_activeUsers) over
71 (order by activated_year, activated_month) as cumsum_activeUsers
72 from
73 (select year(str_to_date(activated_at,"%d-%m-%Y %H:%i")) as activated_year,
74 month(str_to_date(activated_at,"%d-%m-%Y %H:%i")) as activated_month,
75 count(distinct user_id) as num_of_activeUsers
76 from users
77 where state= "active"
78 group by activated_year, activated_month
79 order by activated_year, activated_month) as growth_table;
80
81
82
83
84
SQLAdditions
My Snippets
Administration Schemas
Information
No object selected
Object Info Session
Context Help Snippets
Output
Nifty bank -1.08%
Search
ENG IN
11:07 20-03-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
amitdb
ig_clone
new_db
Tables
email_events
events
job_data
users
Views
Stored Procedures
Functions
sys
sql2projectcodes*
events
email_events
job_data
events
Limit to 1000 rows
65
66 # User growth analysis
67
68
Result Grid
Filter Rows:
Export:
Wrap Cell Contents:
activated_year activated_month num_of_activeUsers cumsum_activeUsers
2013 1 160 160
2013 2 160 320
2013 3 150 470
2013 4 181 651
2013 5 214 865
2013 6 213 1078
2013 7 284 1362
2013 8 316 1678
2013 9 330 2008
2013 10 390 2398
2013 11 399 2797
2013 12 486 3283
2014 1 552 3835
2014 2 525 4360
2014 3 615 4975
2014 4 726 5701
Result 2
Read Only
SQLAdditions
My Snippets
Administration Schemas
Information
No object selected
Object Info Session
Context Help Snippets
Output
31°C Sunny
Search
ENG IN
11:06 20-03-2024

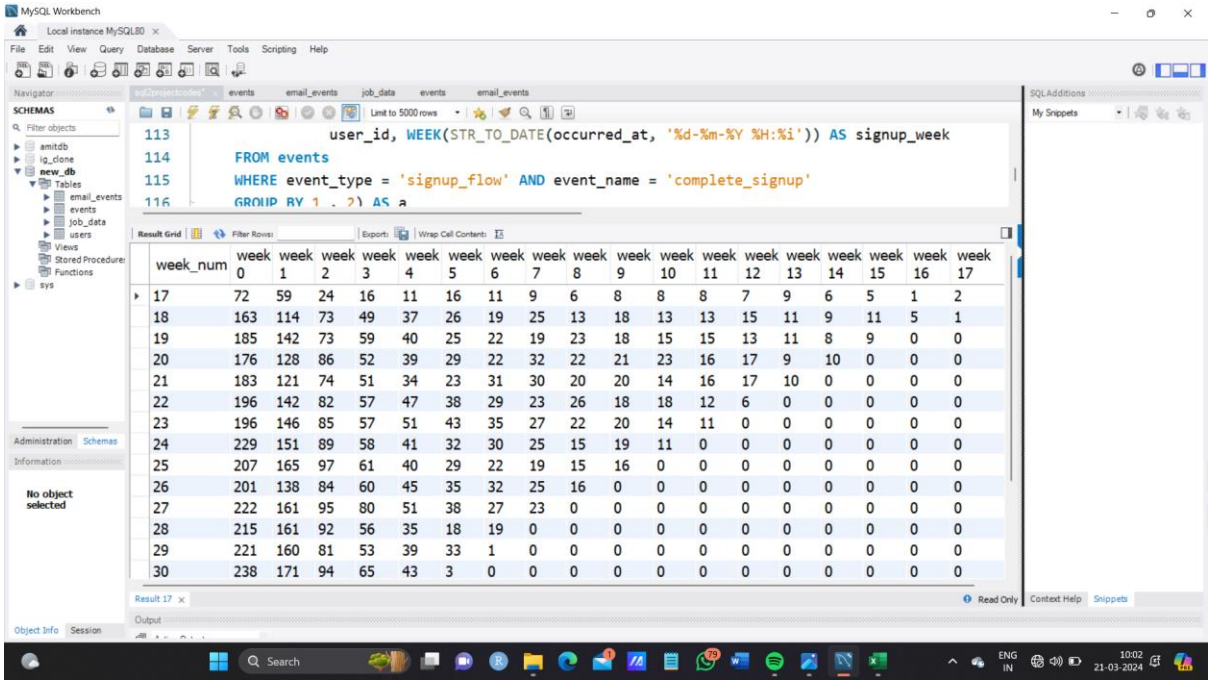


c) **Weekly Retention Analysis:** Here I write an SQL query to calculate the weekly retention of users based on their sign-up cohort.



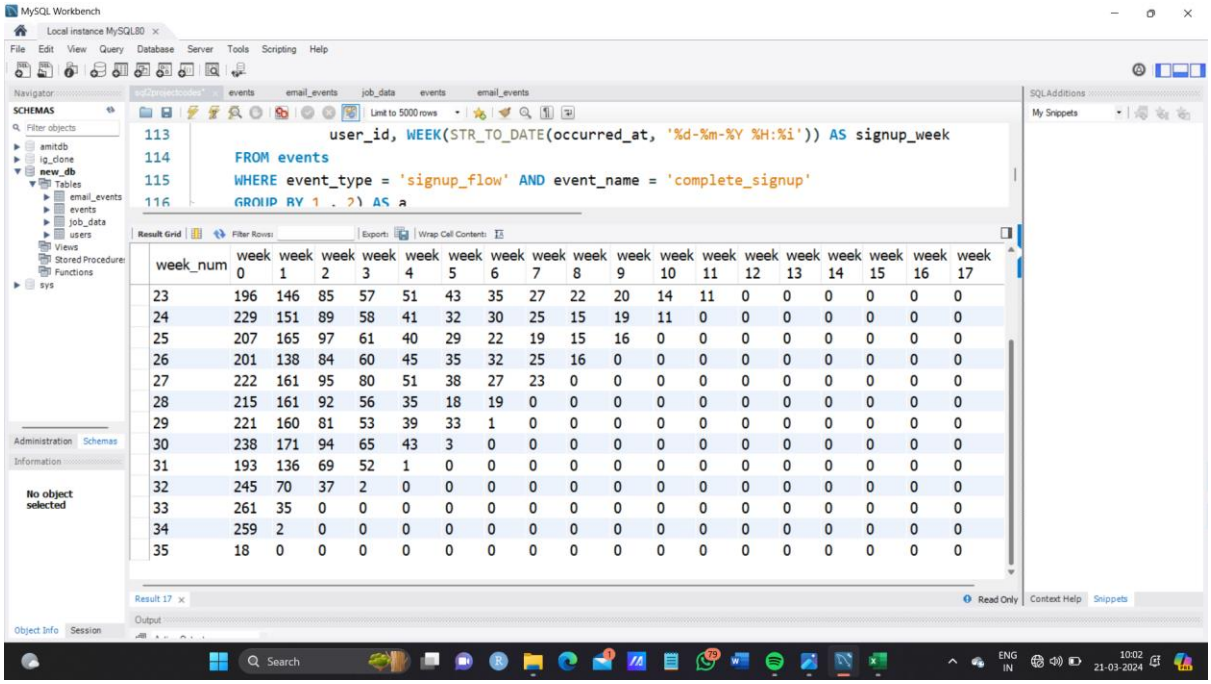


```
    SUM(CASE WHEN week_nums = 17 THEN 1 ELSE 0 END) AS 'week 17'
FROM
    (SELECT
        a.user_id, b.engagement_week, a.signup_week,
        b.engagement_week - a.signup_week AS week_nums
    FROM
        (SELECT
            user_id, WEEK(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS signup_week
    FROM events
    WHERE event_type = 'signup_flow' AND event_name = 'complete_signup'
    GROUP BY 1 , 2) AS a
    LEFT JOIN
    (SELECT
        user_id, WEEK(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS engagement_week
    FROM events
    WHERE event_type = 'engagement'
    GROUP BY 1 , 2) AS b
    ON b.user_id = a.user_id) AS c
GROUP BY signup_week
ORDER BY signup_week;
```



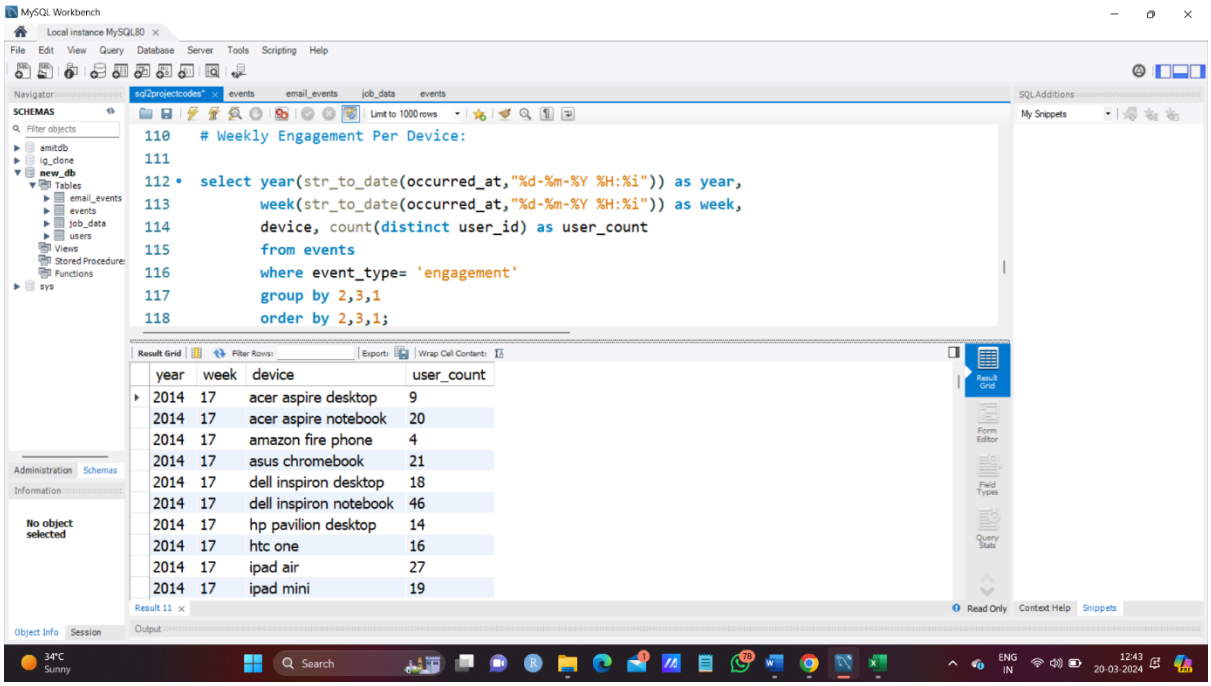
```
            user_id, WEEK(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS signup_week
    FROM events
    WHERE event_type = 'signup_flow' AND event_name = 'complete_signup'
    GROUP BY 1 , 2) AS a
```

week_num	week 0	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13	week 14	week 15	week 16	week 17
17	72	59	24	16	11	16	11	9	6	8	8	8	7	9	6	5	1	2
18	163	114	73	49	37	26	19	25	13	18	13	13	15	11	9	11	5	1
19	185	142	73	59	40	25	22	19	23	18	15	15	13	11	8	9	0	0
20	176	128	86	52	39	29	22	32	22	21	23	16	17	9	10	0	0	0
21	183	121	74	51	34	23	31	30	20	20	14	16	17	10	0	0	0	0
22	196	142	82	57	47	38	29	23	26	18	18	12	6	0	0	0	0	0
23	196	146	85	57	51	43	35	27	22	20	14	11	0	0	0	0	0	0
24	229	151	89	58	41	32	30	25	15	19	11	0	0	0	0	0	0	0
25	207	165	97	61	40	29	22	19	15	16	0	0	0	0	0	0	0	0
26	201	138	84	60	45	35	32	25	16	0	0	0	0	0	0	0	0	0
27	222	161	95	80	51	38	27	23	0	0	0	0	0	0	0	0	0	0
28	215	161	92	56	35	18	19	0	0	0	0	0	0	0	0	0	0	0
29	221	160	81	53	39	33	1	0	0	0	0	0	0	0	0	0	0	0
30	238	171	94	65	43	3	0	0	0	0	0	0	0	0	0	0	0	0

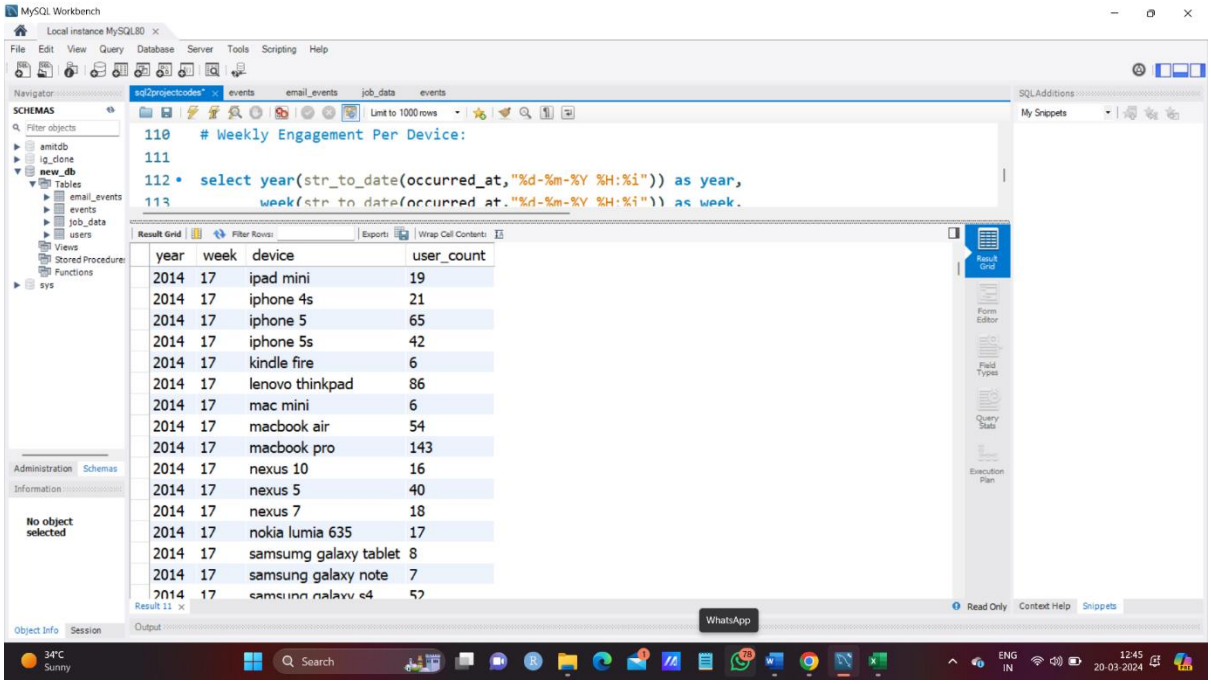


week_num	week 0	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13	week 14	week 15	week 16	week 17
23	196	146	85	57	51	43	35	27	22	20	14	11	0	0	0	0	0	0
24	229	151	89	58	41	32	30	25	15	19	11	0	0	0	0	0	0	0
25	207	165	97	61	40	29	22	19	15	16	0	0	0	0	0	0	0	0
26	201	138	84	60	45	35	32	25	16	0	0	0	0	0	0	0	0	0
27	222	161	95	80	51	38	27	23	0	0	0	0	0	0	0	0	0	0
28	215	161	92	56	35	18	19	0	0	0	0	0	0	0	0	0	0	0
29	221	160	81	53	39	33	1	0	0	0	0	0	0	0	0	0	0	0
30	238	171	94	65	43	3	0	0	0	0	0	0	0	0	0	0	0	0
31	193	136	69	52	1	0	0	0	0	0	0	0	0	0	0	0	0	0
32	245	70	37	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	261	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	259	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D) **Weekly Engagement Per Device:** Here I write an SQL query to calculate the weekly engagement per device.

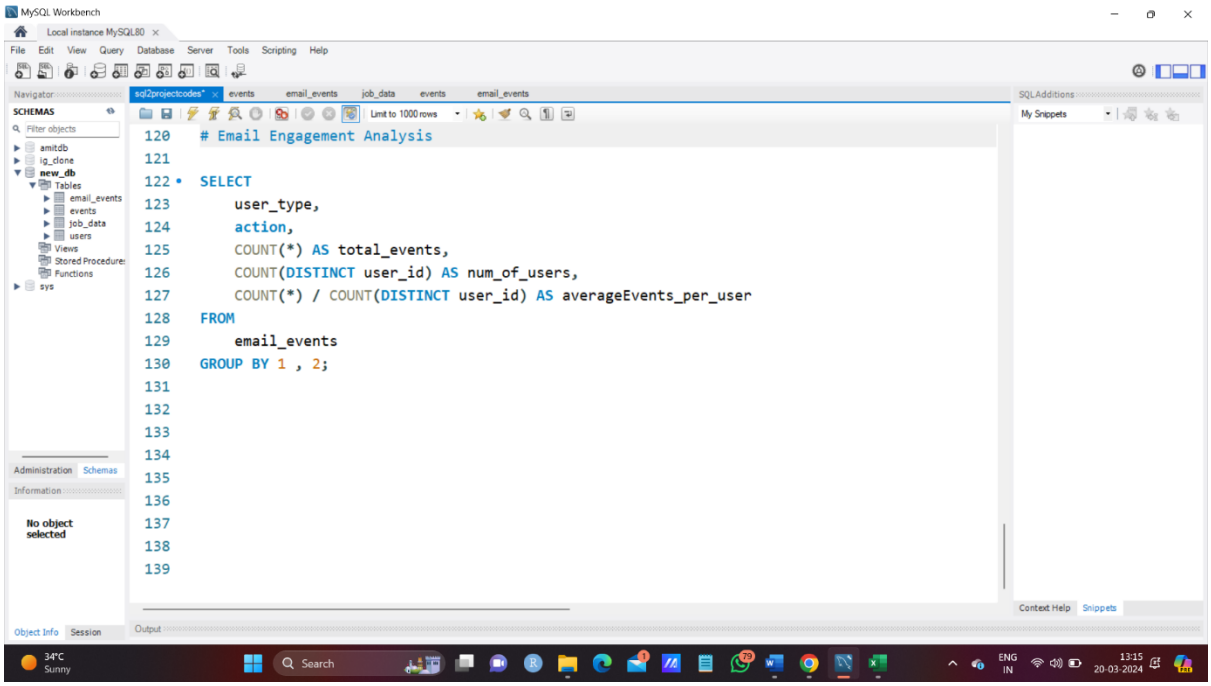


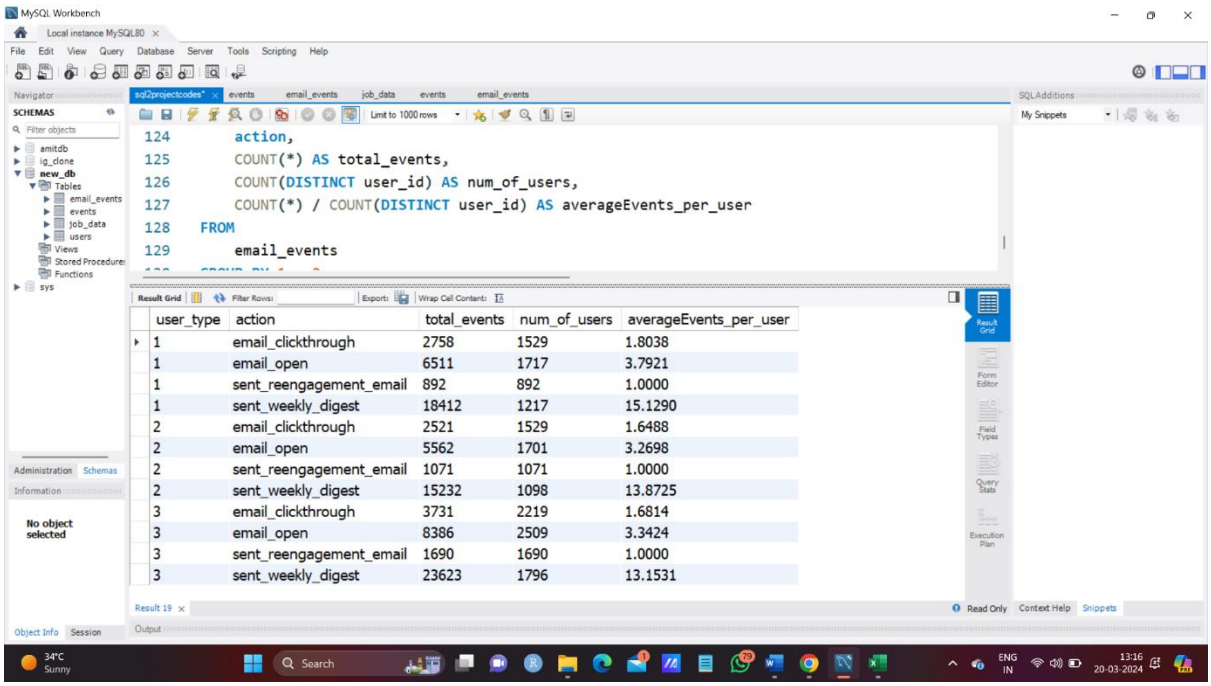
year	week	device	user_count
2014	17	acer aspire desktop	9
2014	17	acer aspire notebook	20
2014	17	amazon fire phone	4
2014	17	asus chromebook	21
2014	17	dell inspiron desktop	18
2014	17	dell inspiron notebook	46
2014	17	hp pavilion desktop	14
2014	17	htc one	16
2014	17	ipad air	27
2014	17	ipad mini	19



And continues…

E) **Email Engagement Analysis:** Here I write an SQL query to calculate the email engagement metrics.





TECH-STACK USED:

For the analysis of data, I used **the MySQL Workbench 8.0 CE**. And in this software, I imported the data sets given and use SQL queries to find different insights for different needs. The reason why we use the SQL is that since the database is relational and we want to join different data to get the information we want the SQL is the best choice. Also, for the dataset that is very large SQL is useful. For the reporting of the project, I used the **Microsoft Word.**

INSIGHTS AND RESULTS:

CASE STUDY 1: JOB DATA ANALYSIS

A) **Jobs reviewed overtime:** After the SQL query we can see that on an average of **218 jobs** were reviewed per hour on **28th NOV 2020,** which is the highest in all dates.

B) **Throughput Analysis:** The 7-day rolling average of the jobs reviewed per second is **0.0268.** Comparing the both averages that we have found about the jobs reviewed I think the average job reviewed per hour in each day will

make more sense to our analysis because this will help to tell that in which days of the week the mostly the jobs were reviewed, and what can we do to improve the work in the other days.

C) **Language Share Analysis:** In this analysis we were able to find that which language was there in the most of the content that was reviewed overtime and found out that **37.5%** of content was in **Persian language** amongst all other languages.

D) **Duplicate Rows Detection:** If we consider every column in each row, we cannot find any duplicate rows, that is by considering the entire row in a data we cannot find any duplicate rows. But when we consider about the unique actor id, we can find that there are **2 duplicates for the actor id 1003**. Similarly, when we consider the job id, we find that there are **3 duplicates for the job id 23.**

CASE STUDY 2: INVESTIGATING METRIC SPIKES

A) **Weekly User Engagement:** The user engagement is **highest in the week 30 with 1467 users** and the user engagement is **least in the week 35 with 49 users.**

B) **User Growth Analysis:** The year and month at which the **most users activated their account is 8th(august) month of 2014**. The Year and month in which the **least users activated their account is 3rd(march) month of 2013.** But in an overall view there was a significant increase in the number of users during this time.

C) **Weekly Retention Analysis:** Here we can see that the user engagement is gradually decreasing as the number of weeks increases after the sign up of the user. Even if there are many people who activated the account, there is not enough engagement after the signup of the account.

D) **Weekly Engagement Per Device:** This analysis helps us to understand that on a weekly basis how many users have used a particular device to login to their accounts. This will help the team to understand that which device is mostly used by users on a weekly basis.

E) **Email Engagement Analysis:** For the user_type 1, **on an average of 4 events of email_open was occurred for a particular user.** Similarly, we can interpret the rest of the results.

As a result, this project helps me to learn about a variety field such as operation analytics and the aspect of investigating metric spikes. Also, this project developed my SQL understanding a lot more. I have learned a bit about the advanced SQL and the queries I have written made me understand that SQL can interpret and analyse the data in a wide variety of ways.