OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Project Description:

Performing Operational analytics, which involves analyzing a company's end-to-end operations to identify the areas for improvement within the company. It's important to understand and explain the sudden changes in key aspects of the company's metrics in various departments such as operations, support, and marketing. This process is called as investigating metric spike.

In this project, I will as a data analyst will perform to different case studies with two different databases. In case study-1, a sample data is provided as asked to create a table based on it by "Data Creation" and answer the questions by analyzing the end-to-end operations data and to provide insights regarding rolling average, jobs reviewed over time and so on.

In case study-2, a database with 3 tables are given, I have to work with this data table to learn about the changes in user metrics over time to understand the user behavior regarding the product and provide any insights and interpretations of the results for future developments to stakeholders.

Approach:

Ask: I will first define the questions to solve, these questions are regarding analyzing end-to-end operations and investigating metric spikes.

- ***Case study1 - Job Data Analysis:***
    1. How many jobs are reviewed per hour for each day in November, 2020?
    2. What is the 7-day rolling average of throughput and explaining whether using daily metric or the 7-day rolling average is good?
    3. What is percentage share of each language over the last 30 days?
    4. Identifying the duplicate rows in the data.
- ***Case study2 – Investigating Metric Spike:***
    1. Measuring the activeness of users on a weekly basis.
    2. What is the user growth for the product?
    3. What is the weekly retention rate of users based on their sign-up cohort?
    4. What is the weekly engagement per device?
    5. What is the email engagement metrics like how many clicked and opened a mail and how many sent a mail?

Prepare: For case study1, a sample data is given. We have to create a sample table with process called 'Data Creation'. I used MS. Excel to create this sample data by using RANDBETWEEN(), INDEX(), TEXT(), DATE() functions to create this sample of 500 rows.

This sample table is named job_data with 7 columns namely job_id, actor_id, event, language, time_spent, org, ds. This data is then uploaded to MySQL workbench under database called case_study1. Click here for datset.

For case study2, I am given 3 tables namely users, events, email_events. The blank spaces in these tables are removed and can be uploaded to MySQL workbench by creating a database called project3. These tables are linked by a primary key called user_id.

Process: The data in both databases is checked for null values and no null values are found. In database project3, for tables we have to change the data type for columns created_at, activated-at in users table, occurred_at in events table, occurred_at in email_events table from 'varchar' to 'date' which can be done by STR_TO_DATE() function.

Analyze: Finally, the data is ready for analyzing. Here we will analyze the end-to-end operations for any changes in user metrics to provide data-driven insights for our stakeholders.

For this we use MySQL workbench,

To remind the workbench we are using the "case_study1", "project3" database, we have to revoke the database by using the following syntax,

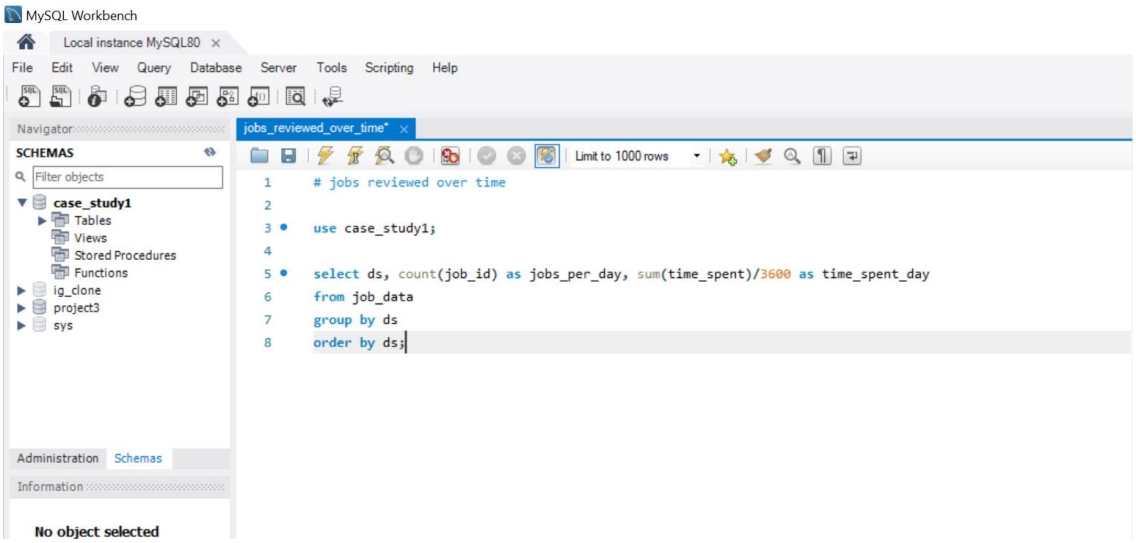
USE case_study1; # For answering questions related to job_data table

USE project3; # for answering questions related to user metrics

now SQL will know to use this database for the future queries until the end of the session

Case study1 – Job Data Analysis:

- **Jobs Reviewed Over Time:** Here we are going to calculate number of jobs reviewed per hour for each day in November, 2020.



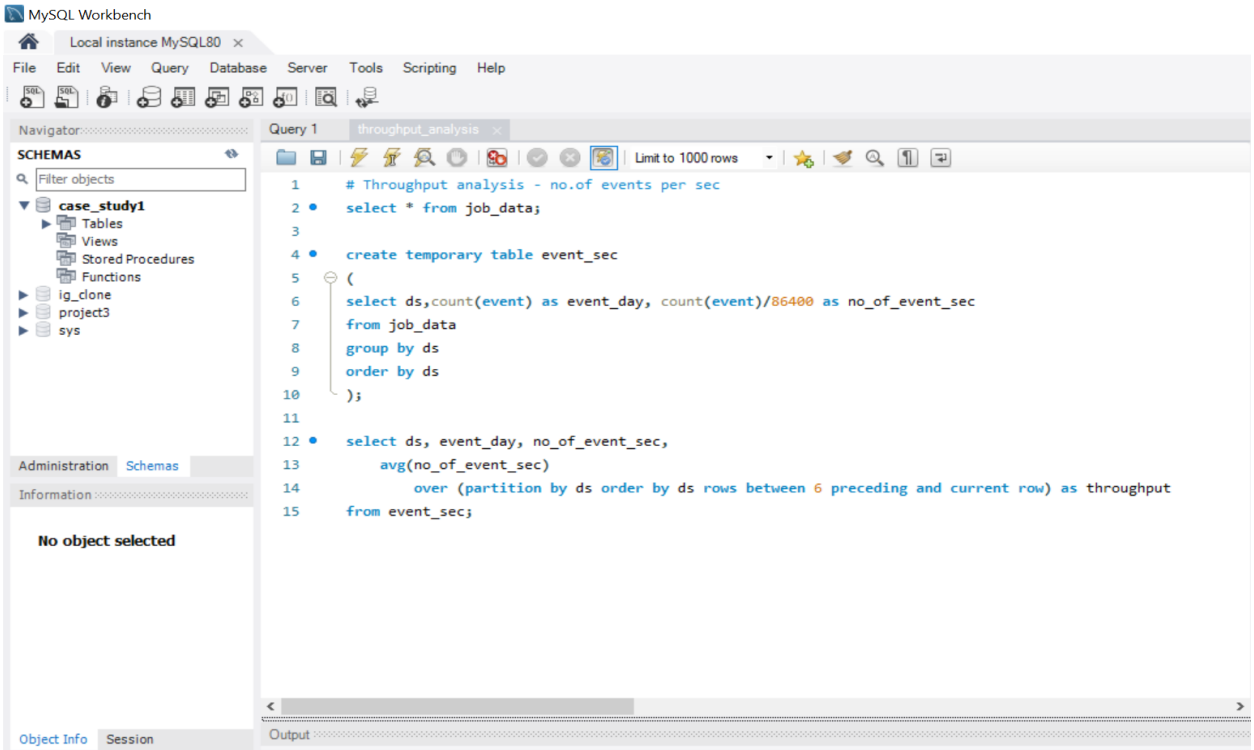
***Output*:** Query returns 30 rows for each day of November, 2020. The result below gives us the number of jobs reviewed each day and time spent to review them in seconds.

Result Grid | Filter Rows:

ds	jobs_per_day	time_spent_day
2020-11-01	27	0.6833
2020-11-02	18	0.4103
2020-11-03	10	0.2189
2020-11-04	21	0.6022
2020-11-05	16	0.4258
2020-11-06	12	0.3447
2020-11-07	8	0.1400
2020-11-08	15	0.3117
2020-11-09	16	0.3303
2020-11-10	16	0.3081
2020-11-11	19	0.5000
2020-11-12	18	0.4397
2020-11-13	26	0.5908
2020-11-14	19	0.4419
2020-11-15	23	0.6458
2020-11-16	14	0.3486
2020-11-17	16	0.4514
2020-11-18	13	0.3469
2020-11-19	18	0.5517
2020-11-20	13	0.3781
2020-11-21	24	0.6825
2020-11-22	15	0.4064
2020-11-23	14	0.3350
2020-11-24	17	0.3964
2020-11-25	23	0.5042
2020-11-26	15	0.3975
2020-11-27	11	0.2319
2020-11-28	19	0.5797
2020-11-29	14	0.4006
2020-11-30	10	0.3614

Result 1

- **Throughput Analysis:** I am going to calculate the 7-day rolling average of throughput i.e., number of events per second. Here I am going to create a temporary table called event_sec, which gives number of events per second per day. Then use this temporary table to calculate 7-day rolling average of throughput.



Output: This query gives us 30 rows, number of events per second as well as 7-day rolling average.

Result Grid | Filter Rows: | Export:

ds	event_day	no_of_event_sec	throughput
2020-11-01	27	0.0003	0.00030000
2020-11-02	18	0.0002	0.00020000
2020-11-03	10	0.0001	0.00010000
2020-11-04	21	0.0002	0.00020000
2020-11-05	16	0.0002	0.00020000
2020-11-06	12	0.0001	0.00010000
2020-11-07	8	0.0001	0.00010000
2020-11-08	15	0.0002	0.00020000
2020-11-09	16	0.0002	0.00020000
2020-11-10	16	0.0002	0.00020000
2020-11-11	19	0.0002	0.00020000
2020-11-12	18	0.0002	0.00020000
2020-11-13	26	0.0003	0.00030000
2020-11-14	19	0.0002	0.00020000
2020-11-15	23	0.0003	0.00030000
2020-11-16	14	0.0002	0.00020000
2020-11-17	16	0.0002	0.00020000
2020-11-18	13	0.0002	0.00020000
2020-11-19	18	0.0002	0.00020000
2020-11-20	13	0.0002	0.00020000
2020-11-21	24	0.0003	0.00030000
2020-11-22	15	0.0002	0.00020000
2020-11-23	14	0.0002	0.00020000
2020-11-24	17	0.0002	0.00020000
2020-11-25	23	0.0003	0.00030000
2020-11-26	15	0.0002	0.00020000
2020-11-27	11	0.0001	0.00010000
2020-11-28	19	0.0002	0.00020000
2020-11-29	14	0.0002	0.00020000
2020-11-30	10	0.0001	0.00010000

Result 2

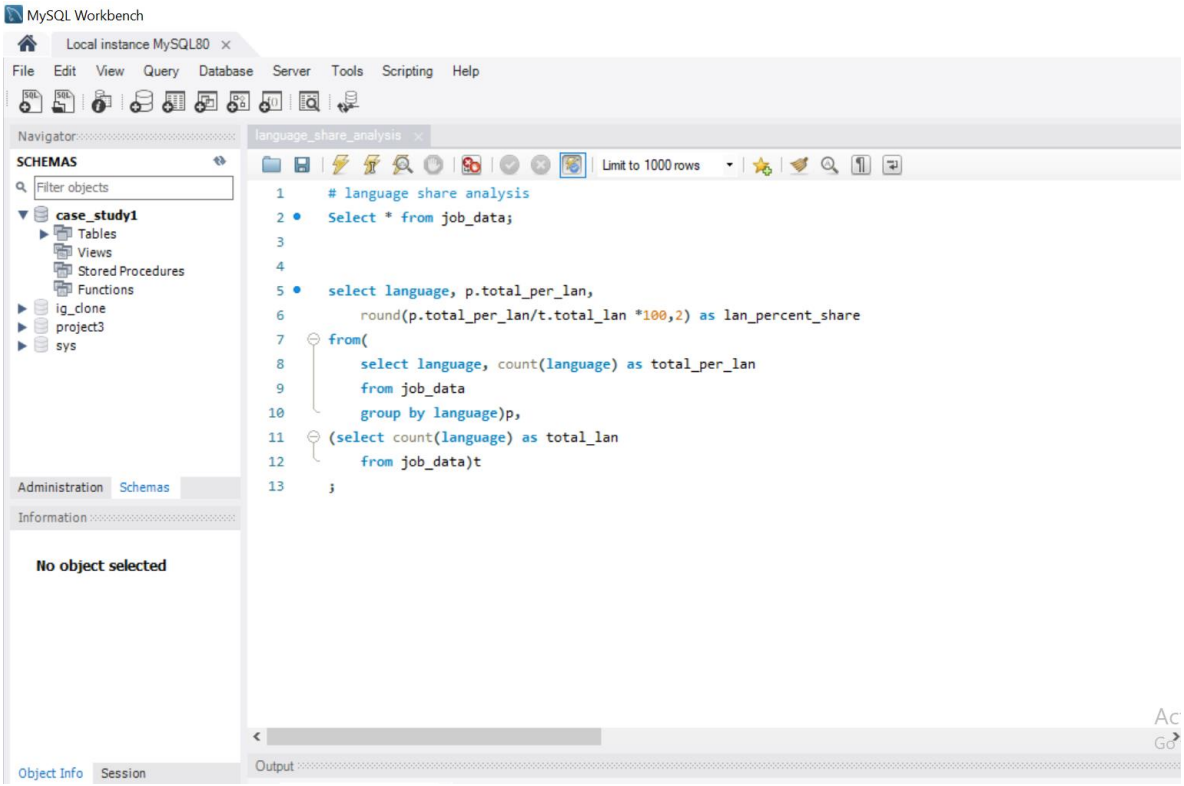
7-day rolling average is calculated by joining every row within 7-days and taking the average of it. For this we used core window function.

Q- Do you prefer 7-day rolling average or daily metric?

A- *I prefer 7-day rolling average to using daily metric because, rolling average uses smaller parts of the data that allows analysts to discover the way average changes over time (7 days, 30 days etc.,)*

Rolling average is also useful for finding long-term trends while a daily metrics are affected by short-term fluctuations in market or production process which will not affect the long-term trends but provide ineffective insights about trends and leads to inaccurate interpretations of the results.

- **Language Share Analysis:** Query about the percentage share of each language in the last 30 days,

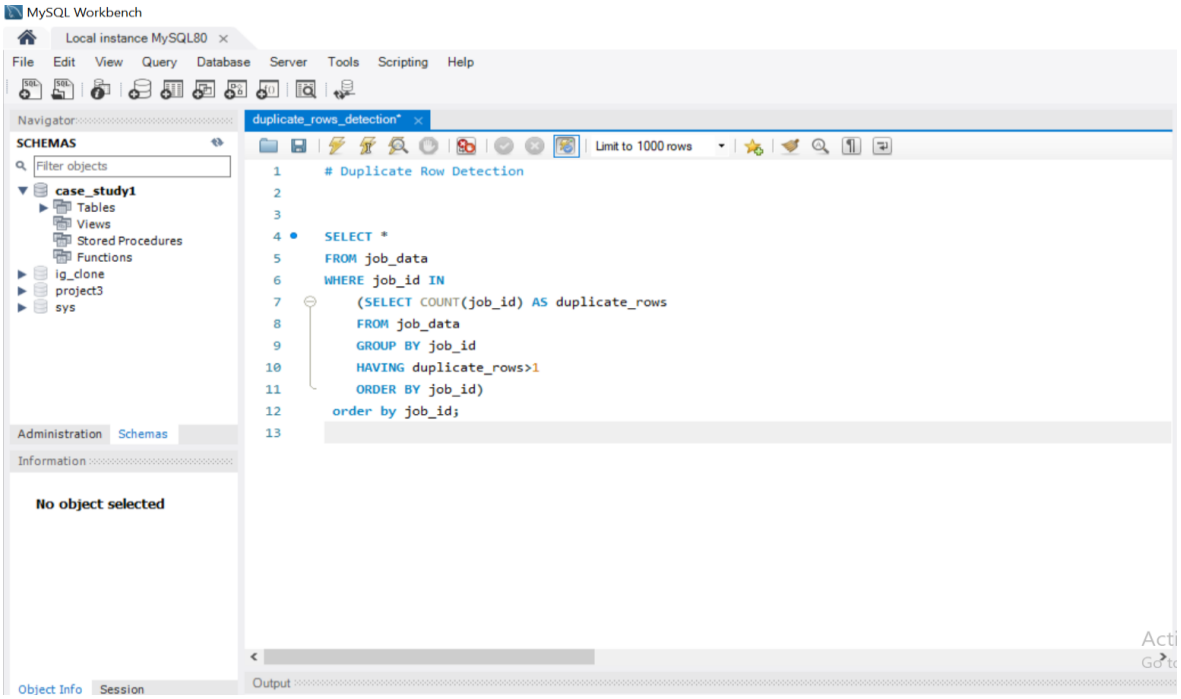


Output: We have percentage share of each language in last 30days. There are 5 languages of the context. The largest share of 18.80% goes to Italian and Persian while the smallest share is French with 15%.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

language	total_per_lan	lan_percent_share
French	75	15.00
Hindi	82	16.40
Italian	94	18.80
Persian	94	18.80
English	81	16.20
Arabic	74	14.80

- **Duplicate Rows Detection:** Identifying the duplicate rows in data,



Output: This query shows all the duplicate rows in the job_data table on the basis of job_id. There are 225 duplicate rows in the table.

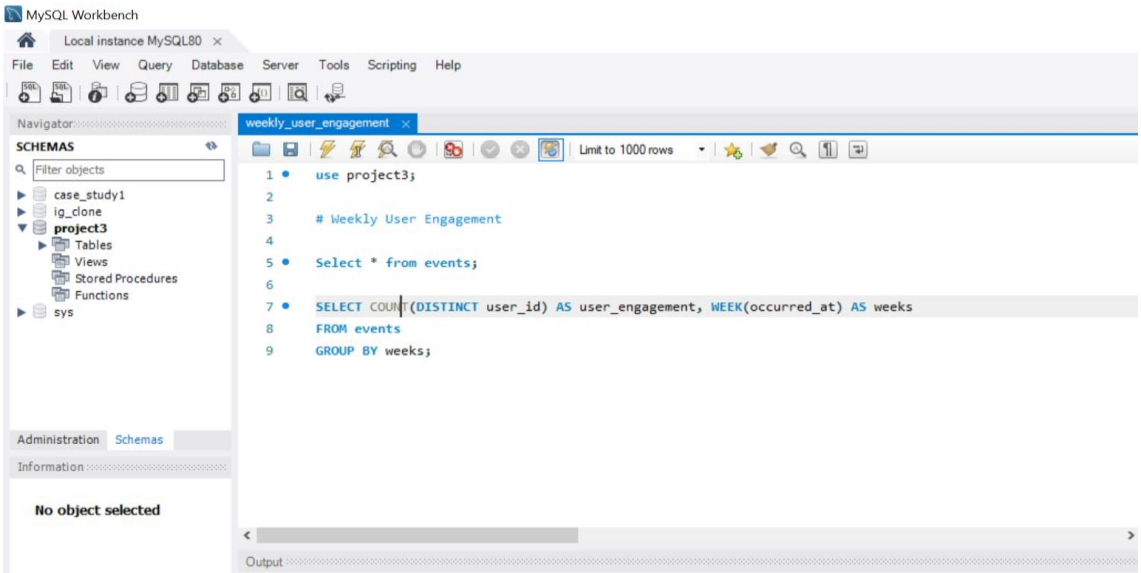
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

job_id	actor_id	event	language	time_spent	org	ds
11	1094	skip	Persian	117	C	2020-11-03
11	1064	tranfer	Italian	50	C	2020-11-25
11	1029	decision	Persian	20	D	2020-11-09
11	1073	skip	Hindi	141	B	2020-11-01
11	1070	skip	Arabic	79	B	2020-11-23
11	1053	decision	Hindi	155	A	2020-11-04
11	1005	skip	Arabic	84	B	2020-11-22
11	1018	skip	French	43	A	2020-11-25
11	1092	tranfer	Hindi	73	D	2020-11-08
11	1018	tranfer	Arabic	157	C	2020-11-08
11	1073	decision	Italian	102	A	2020-11-21
11	1093	tranfer	Italian	71	C	2020-11-29
12	1073	tranfer	Persian	66	D	2020-11-15
12	1071	decision	French	95	C	2020-11-01
12	1021	tranfer	Persian	173	B	2020-11-14
12	1071	tranfer	Hindi	140	A	2020-11-30
12	1035	tranfer	English	165	B	2020-11-28

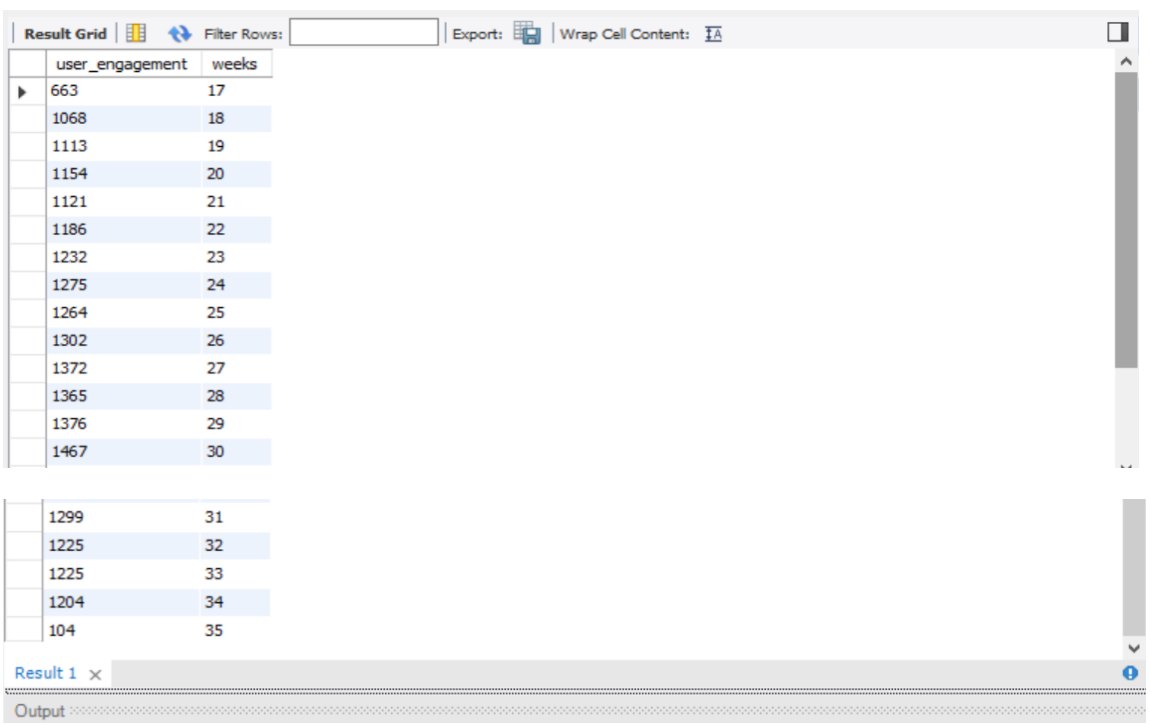
4 22:57:10 SELECT * FROM job_data WHERE job_id IN (SELECT COUNT(job_id) AS duplicat... 225 row(s) returned

Case Study2 – Investigating Metric Spike:

- **Weekly User Engagement:** Calculating the weekly user engagement i.e., activeness of the users on weekly basis. I am going to use events table for this query,

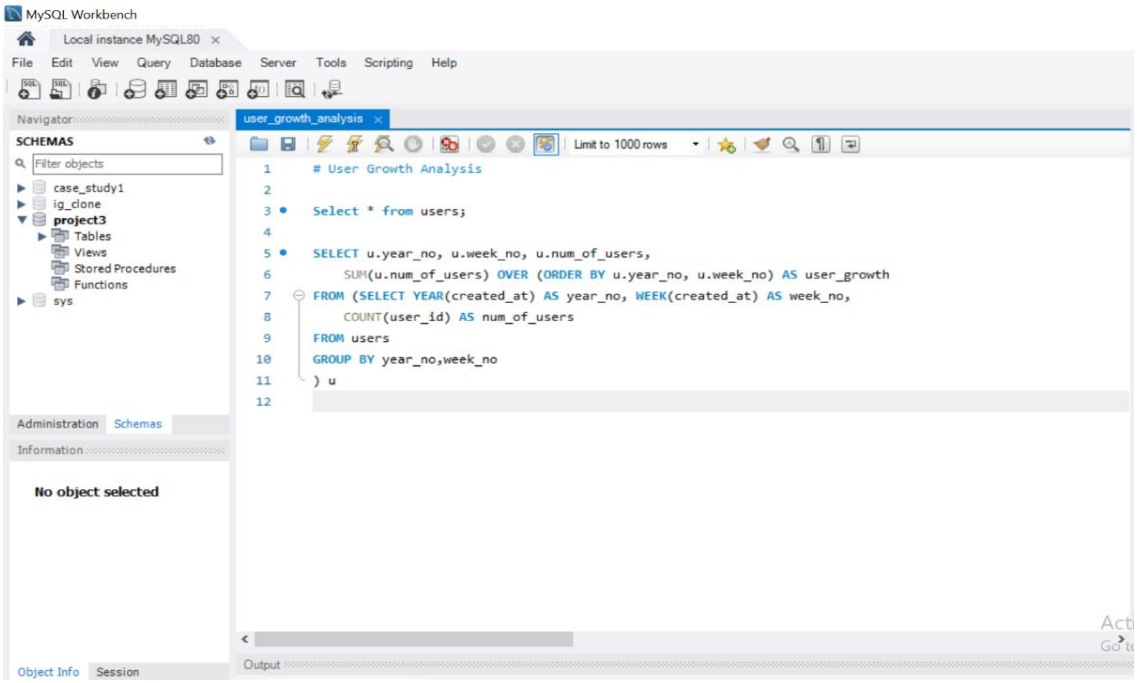


Output: User engagement on weekly basis, we can observe a gradual increase in users with 30th week have more user engagement than slight decrease can be seen after that week and lowest been 35th week with only 104 users.



user_engagement	weeks
663	17
1068	18
1113	19
1154	20
1121	21
1186	22
1232	23
1275	24
1264	25
1302	26
1372	27
1365	28
1376	29
1467	30
1299	31
1225	32
1225	33
1204	34
104	35

- **User Growth Analysis:** Analyzing the growth of the users over time for product,



Output: The query returns 89 rows with cumulative count of users, we can observe a steady growth of users.

Result Grid | Filter Rows: | Ex

year_no	week_no	num_of_users	user_growth
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542

Result Grid | Filter Rows: | Export:

year_no	week_no	num_of_users	user_growth
2013	15	43	585
2013	16	46	631
2013	17	49	680
2013	18	44	724
2013	19	57	781
2013	20	39	820
2013	21	49	869
2013	22	54	923
2013	23	50	973
2013	24	45	1018
2013	25	57	1075
2013	26	56	1131
2013	27	52	1183
2013	28	72	1255
2013	29	67	1322

Result Grid | Filter Rows: | Expo

year_no	week_no	num_of_users	user_growth
2013	30	67	1389
2013	31	67	1456
2013	32	71	1527
2013	33	73	1600
2013	34	78	1678
2013	35	63	1741
2013	36	72	1813
2013	37	85	1898
2013	38	90	1988
2013	39	84	2072
2013	40	87	2159
2013	41	73	2232
2013	42	99	2331
2013	43	89	2420
2013	44	96	2516

Result 2

Result Grid | Filter Rows: | Expo

year_no	week_no	num_of_users	user_growth
2013	45	91	2607
2013	46	88	2695
2013	47	102	2797
2013	48	97	2894
2013	49	116	3010
2013	50	124	3134
2013	51	102	3236
2013	52	47	3283
2014	0	83	3366
2014	1	126	3492
2014	2	109	3601
2014	3	113	3714
2014	4	130	3844
2014	5	133	3977
2014	6	135	4112

Result 2

Result Grid | Filter Rows: | Exp

year_no	week_no	num_of_users	user_growth
2014	7	125	4237
2014	8	129	4366
2014	9	133	4499
2014	10	154	4653
2014	11	130	4783
2014	12	148	4931
2014	13	167	5098
2014	14	162	5260
2014	15	164	5424
2014	16	179	5603
2014	17	170	5773
2014	18	163	5936
2014	19	185	6121
2014	20	176	6297
2014	21	183	6480

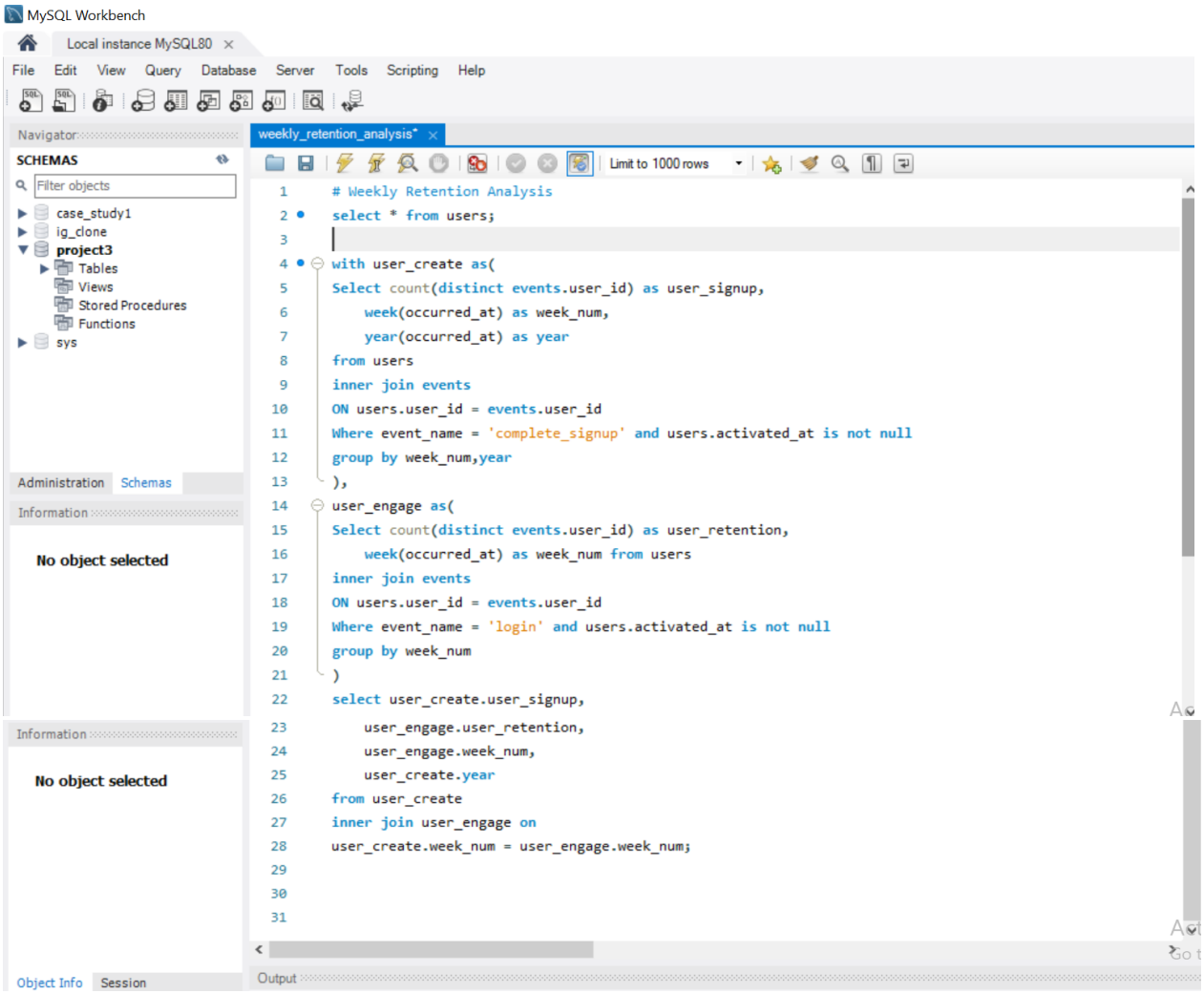
Result 2

Result Grid | Filter Rows: | Ex

year_no	week_no	num_of_users	user_growth
2014	22	196	6676
2014	23	196	6872
2014	24	229	7101
2014	25	207	7308
2014	26	201	7509
2014	27	222	7731
2014	28	215	7946
2014	29	221	8167
2014	30	238	8405
2014	31	193	8598
2014	32	245	8843
2014	33	261	9104
2014	34	259	9363
2014	35	18	9381

Result 2

- **Weekly Retention Analysis:** Analyzing the retention of users on a weekly basis based on sign-up cohort,

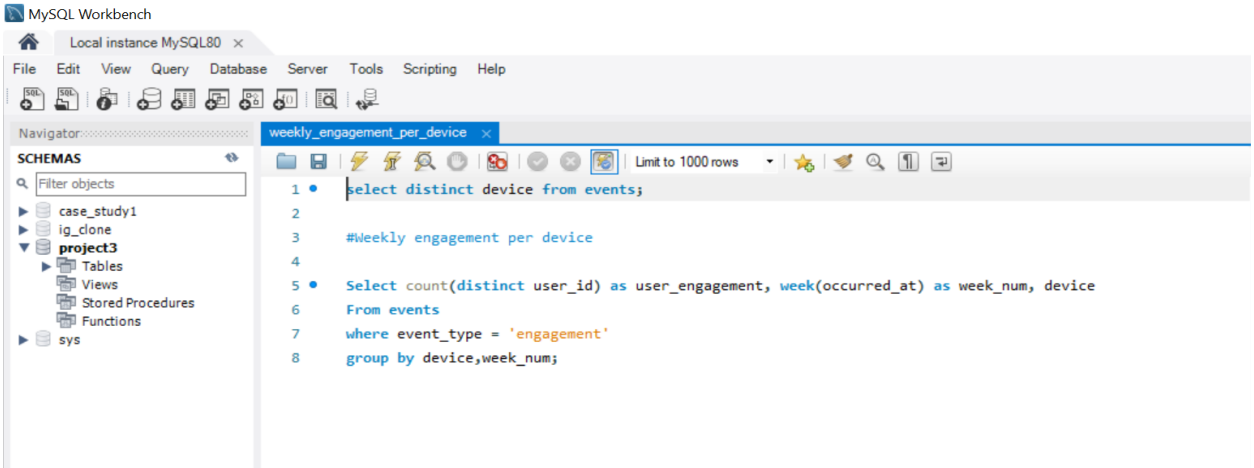


Output: 19 rows are returned. It shows count of users who are signing up for the product and count of user retention on weekly basis, we can see a decrease in retention rate on 31st,32nd and 33rd week of 2014. User engagement data for 2013 is not available in the given table.

Result Grid | Filter Rows: | Export

user_signup	user_retention	week_num	year
72	663	17	2014
163	1068	18	2014
185	1113	19	2014
176	1154	20	2014
183	1121	21	2014
196	1186	22	2014
196	1232	23	2014
229	1275	24	2014
207	1264	25	2014
201	1302	26	2014
222	1372	27	2014
215	1365	28	2014
221	1376	29	2014
238	1467	30	2014
193	1299	31	2014
245	1225	32	2014
261	1225	33	2014
259	1204	34	2014
18	104	35	2014

- **Weekly Engagement per Device:** Measuring the weekly activeness of users per device,



Output: This query returns numbers of users engaging in different devices on weekly basis. It returns 491 rows for different devices in different weekly of 2014.

Result Grid | Filter Rows: | Exp

user_engagement	week_num	device
9	17	acer aspire desktop
26	18	acer aspire desktop
23	19	acer aspire desktop
23	20	acer aspire desktop
29	21	acer aspire desktop
25	22	acer aspire desktop
22	23	acer aspire desktop
24	24	acer aspire desktop
28	25	acer aspire desktop
29	26	acer aspire desktop
29	27	acer aspire desktop
30	28	acer aspire desktop
28	29	acer aspire desktop
33	30	acer aspire desktop
31	31	acer aspire desktop
35	32	acer aspire desktop
39	33	acer aspire desktop

6 22:04:17 Select count(distinct user_id) as user_engagement, week(occurred_at) as week_nu... 491 row(s) returned

- **Email Engagement Analysis:** analyzing the users' engagement with email service by calculating the email metrics,

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- case_study1
- ig_clone
- project3
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Administration Schemas

Information

No object selected

Object Info Session

email_engagement_analysis*

Limit to 1000 rows

```
# Email engagement analysis

select count(action), action
from email_events
group by action; # But we have calculate the metrics like click rates and open rates of emails in percents

SELECT
    100*sum(case
        when m.email_metrics = 'email_opens' then 1 else 0 end)/sum(case
        when m.email_metrics = 'email_sent'then 1 else 0 end) as email_open_rate,
    100*sum(case
        when m.email_metrics = 'email_clicks' then 1 else 0 end)/sum(case
        when m.email_metrics = 'email_sent'then 1 else 0 end) as email_click_rate
FROM(Select *,
CASE
    when action in ('sent_weekly_digest', 'sent_reengagement_email') Then 'email_sent'
    when action = 'email_open' then 'email_opens'
    when action = 'email_clickthrough' then 'email_clicks'
    end as email_metrics
    from email_events
    ) m
```

Output

Output: We have email engagement metrics as for Emails we have two options like opening the email, click-through action. From the result, we can see that on average of 33.59 users open their emails and replied to them while average of 14.79 users only clicked through it.

email_open_rate	email_click_rate
33.5834	14.7899

Thus, conclude our analysis.

Tech-Stack used:

The Tech-stack used is

- A relational database - MySQL workbench 8.0 CE to perform analysis and execute the queries for obtaining meaningful insights.
- Microsoft Excel Office 2016 to create sample data table and also to clean the tables.

Insights:

Share & Act: After analyzing the data, in MySQL workbench we have to share out insights with the primary stakeholders and secondary stakeholders (Team members). So, that data-driven decisions are taken for the success of the program.

My insights are as follows,

Case study1- Job Data Analysis

- 7-day rolling average is better than daily metrics as these are subjected to short term fluctuations and sudden spikes in the metrics which doesn't have any effect in long-term trends will lead to fallacious results.
- Jobs reviewed over time have a slight decrease towards the end of the month.
- Italian and Persian with 18.80% is the most used context language while least used is French with 15%.

Case study2 – Investigating Metric Spike

- We can observe an exponential rise in user engagement and user growth analysis which indicates the success of the product.
- But we can observe a slight decrease in retention of users in the final week of 2014 and necessary measures is to be taken to improve the retention rate. We should encourage the users to engage more or update the application according to the latest trends in the market which appeals more to the non-users resulting in more users completing the signing up and engage more with the application.
- We can see the growth of weekly engagement per device, there is a rise in number of users per device on weekly basis.

Result:

The stakeholder's questions about over time job reviewed, language most used for the context, 7-day rolling average for finding long-term trends and measures is to be taken to reduce the duplicate rows.

The stakeholders should focus more on user retention for the product, and encourage others to sign-up for the product by advertisement and encourage users to engage more with the product as well as emails.