# PROJECT 3

# **Operation Analytics and Investigating Metric Spike**

**Project Description:** Project involves analysing user interactions and engagement with the Instagram app to provide valuable insights that can help the business grow. will be using MY SQL Workbench to provide insights which will help in making decisions in future requirements.

Tech Stack Used: MySQL Workbench 8.0 CE, MS EXCEL.

### Pre-requisite's:

- Creating Database and tables. Importing data from CSV Files to SQL.
- Alter the data type of necessary date fields which are in Text to Timestamp.
- Understand the Table Columns before proceeding with the analysis.

### **Analysis:**

# Case Study 1: Job Data Analysis

### 1. Jobs Reviewed Over Time:

- Use Job Data Analysis Database.
- From table job\_data, filter the records for Nov 2020.
- Calculated no.of jobs ran per sum of time spent for each job as jobs reviewed per hr in a day .
- group by day and sort by jobs\_reviewed\_perhr\_perdy in descending order.

**Insights:** Through this analysis, observed the hourly time review the jobs in a day. With 27/11/2020 being the lowest jobs reviewed with count of 34.61 and 28/11/2020 with highest jobs reviewed with a count of 218.18.



#### 2. Throughput Analysis:

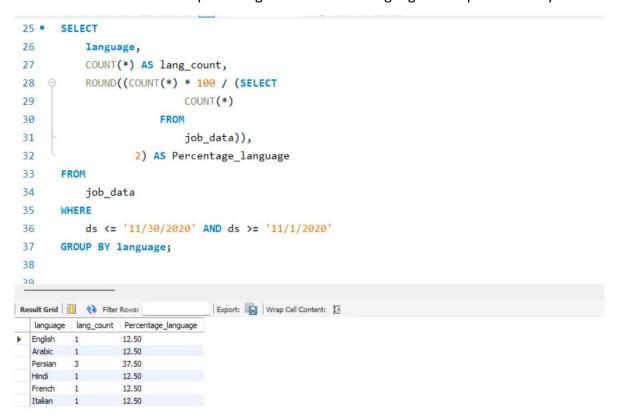
- Using Common Table Expressions Query the data no of jobs, sum of time spent as total time from job data table.
- Group the results by ds.
- In the main query. calculated the 7-day rolling average for throughput (no of jobs per second).
- Using window functions fetch the sum of jobs/sum of total time for 6 preceding and current row of ds column and rounded to 2 digits from CTE table.

**Insights:** Rolling Average gives better idea, as it considers the previous data and is capable of projecting future insights. I prefer rolling average to daily metrics as rolling average in time series data to analyze trends, especially when short-term fluctuations can hide a longer-term trend or cycle.

```
13
        #throughput rolling average.
 14 ● ⊖ WITH CTE AS (
 15
        SELECT ds, COUNT(job_id) AS num_jobs, SUM(time_spent) AS total_time_spent
 16
        FROM job_data
      GROUP BY ds)
 17
 18
        SELECT ds.
 19
        SUM(num_jobs) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)
 20
        / SUM(total_time_spent) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg_7day
 21
 22
Result Grid | Filter Rows:
                                    Export: Wrap Cell Content: IA
           rolling_avg_7day
  11/25/2020 0.0222
  11/26/2020 0.0198
  11/27/2020 0.0146
  11/28/2020 0.0210
  11/29/2020 0.0233
  11/30/2020 0.0268
```

### 3. Language Share Analysis:

• Calculated the percentage share of each language for a span of 30 days.



### 4. Duplicate Rows Detection:

• With this query detected the duplicate rows, as there are no duplicate rows in the data, results are empty. But the same approach can be used for the duplicate row's detection.

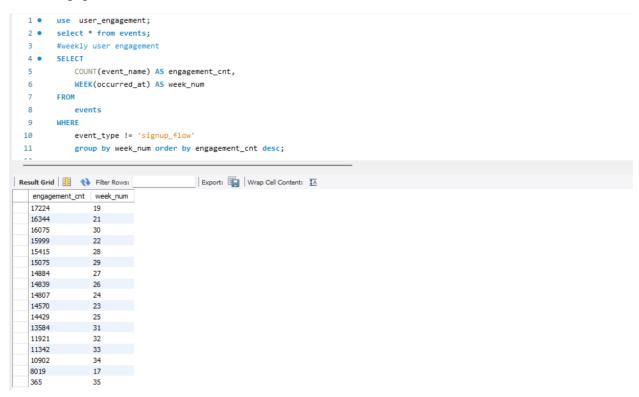
```
🔤 📼 | 🗡 📜 🎉 🤍 | 🚾 | 🤝 | 🤝 | 🐷 | Limit to 10000 rows 💌 🚜 | 💌 🕓 🕕 🖭
37
      # Duplicate detection in a table.
38 •
39
           ds, job_id, actor_id, COUNT(*) AS dup_count
40
41
          job_data
      \ensuremath{\mathsf{GROUP}} BY ds , job_id , actor_id
42
43
      HAVING COUNT(*) > 1;
44
46
47
48
49
Export: Wrap Cell Content: IA
```

# **Case Study 2: Investigating Metric Spike**

### 1. Weekly User Engagement:

- Use database user\_engagement.
- From the events table, consider the data where event type is not sign-up flow.
- Query the engagement count and calculate week of occurred\_at as week\_num using week function.
- Group and order the results by week\_num.

**Insights**: Analyzing the weekly user engagement. With highest engagement in week 19 and lowest in week 35. This insight will further help in the analysis of weeks with lesser engagement.

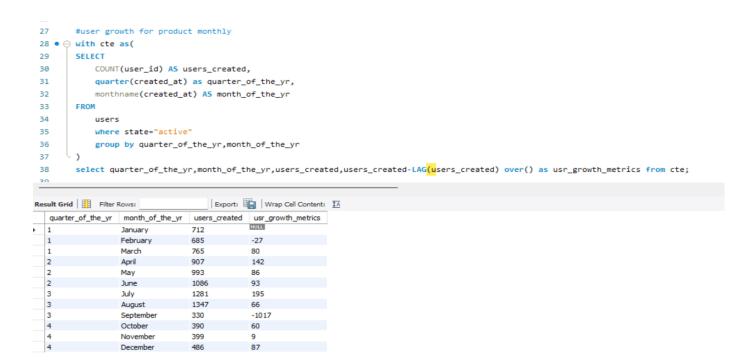


#### 2. User Growth Analysis:

- Created CTE for users table and filtered the data only for active users using where condition
- to query the count of user\_id as user\_growth, monthname and quarter from the created at column using the respective functions.
- From CTE, query the User growth, user\_growth\_metrics i.e., difference in growth compared to previous month.

### Insights:

- From the previous insights there is a fluctuation in the consecutive weeks and a major drop in user engagement from week 34-35. Which raised the question, is it because of the drop in newly created users?
- With this insight we will calculate the user growth and decrease monthly as weekly data is more, analyzing each week will be difficult.



### 3. Weekly Retention Analysis:

In the sub query m and n, calculated the login week and first login from events table

- In the sub query sub, calculated the week number i.e., difference of login week and first login.
- In the main query calculated the weekly retention based on the weekly cohort.

**Insights:** From this analysis, users who have remained is not a major number.

Which might result in user engagement drop.

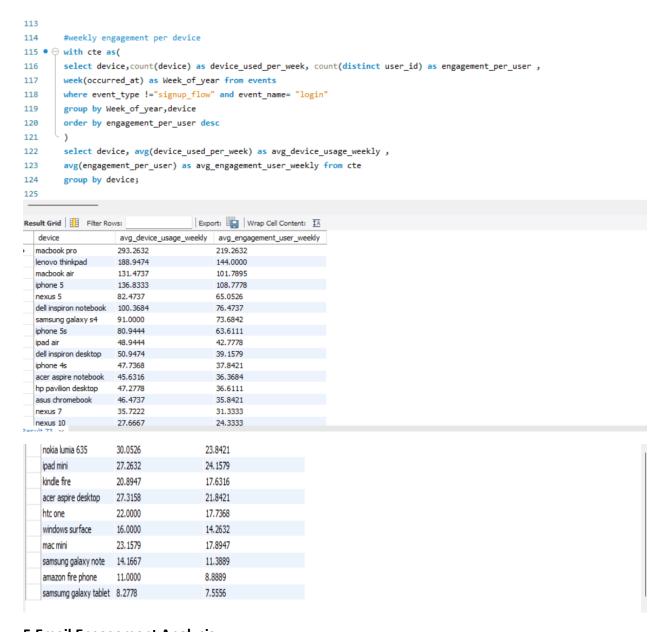
```
86 •
        SELECT first AS "Week Numbers",
        SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS "Week 17",
 87
        SUM(CASE WHEN week_number = 1 THEN 1 ELSE @ END) AS "Week 18",
        SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS "Week 19",
 89
        SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS "Week 20",
 91
        SUM(CASE WHEN week_number = 4 THEN 1 ELSE @ END) AS "Week 21",
        SUM(CASE WHEN week_number = 5 THEN 1 ELSE @ END) AS "Week 22",
        SUM(CASE WHEN week_number = 6 THEN 1 ELSE @ END) AS "Week 23",
 94
        SUM(CASE WHEN week_number = 7 THEN 1 ELSE @ END) AS "Week 24",
 95
        SUM(CASE WHEN week_number = 8 THEN 1 ELSE @ END) AS "Week 25",
 96
        SUM(CASE WHEN week number = 9 THEN 1 ELSE @ END) AS "Week 26".
 97
        SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS "Week 27",
        SUM(CASE WHEN week number = 11 THEN 1 ELSE @ END) AS "Week 28".
98
        SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS "Week 29",
100
        SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS "Week 30",
        SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS "Week 31",
102
        SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS "Week 32",
103
        SUM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS "Week 33",
104
        SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS "Week 34",
105
        SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS "Week 35"
 106
      ⊖ from (
         SELECT m.user_id, m.login_week, n.first, m.login_week - first as week_number FROM
 107
         (SELECT user_id, week(occurred_at) as login_week FROM events GROUP BY 1, 2) m,
 108
 109
         (SELECT user_id, MIN(week(occurred_at)) AS first from events group BY 1) n
         WHERE m.user_id = n.user_id ) sub
 110
 111
         GROUP BY first
         ORDER BY first;
 112
                                                                                                                                             Export: Wrap Cell Content: IA
   17
                663
                                 324
                                                  205
                                                          187
                                                                  167
                                                                           146
                                                                                   145
                                                                                                                    132
                                                                                                            131
                                                                                                                                     116
                        362
                                 261
                                                          147
                                                                  144
                                                                          127
                                                                                   113
                                                                                                           118
                                                                                                                            110
                                                                                           82
                358
                        192
                                 147
                                         111
                                                         67
                                                                                                           41
                                                                                                                    40
   22
                326
                        107
                                93
                                         71
                                                 66
                                                         59
                                                                  56
                                                                          57
                                                                                   53
                                                                                           47
                                                                                                   40
                                                                                                           39
                                                                                                                    31
                                                                                                                                             0
   23
                328
                                                          62
                                                                                           45
                                                                                                            30
   24
                339
                        88
                                79
                                         65
                                                 64
                                                         55
                                                                  59
                                                                          58
                                                                                   38
                                                                                           38
                                                                                                   29
                                                                                                           0
                                                                                                                                             0
   25
                305
                                 66
                                                          52
                                                                  42
                                                                                                           0
   26
                288
                        73
                                50
                                         39
                                                 50
                                                         37
                                                                  35
                                                                          34
                                                                                   26
                                                                                                                                             0
   27
                292
                                                  41
                                                          34
                                                                  29
                                                                          29
                                42
                                         32
                                                         20
   28
                274
                        66
                                                 24
                                                                  24
                                                                          3
                                                                                  0
                                                                                          0
                                                                                                   0
                                                                                                           0
                                                                                                                                             0
   29
                270
                                 45
                                         28
                                                 23
                                                          28
   30
                294
                        52
                                45
                                         28
                                                 28
                215
   31
                                 19
                                         22
   32
                                28
   35
```

Week 33	Week 34	Week 35
82	77	5
67	4	0
2	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

### 4. Weekly Engagement Per Device

- From events table, filter the results where event\_type is not sign-up flow and event, and name is login.
- Using CTE, query count(device) as device user per week, distinct count of users, week num.
- Group the results weekly and by device.
- Order the results by engagement per user in desc.
- From CTE calculate the avg engagement per user, avg weekly device usage.

**Insights:** From this analysis, I've found that mac book pro users have more engagement, while Samsung tablet has the least.



## 5.Email Engagement Analysis:

- From sub query calculated the distinct count of users.
- Using if function calculated sum of users based on the action from events table.
- In the main query, calculated the weekly growth for each action using the window function.

**Insights:** Overall picture of email engagement metrics is observed., And metrics seems to be in the positive side as the no of weekly digest emails sent increased weekly so as the email open and email clickthrough.

```
143
        #email engagement metrics:
145
146
        num users,
147
        time_weekly_digest_sent,
        time_weekly_digest_sent-lag(time_weekly_digest_sent) over(order by week) as time_weekly_digest_sent_growth,
149
        time_email_open,time_email_open-lag(time_email_open) over(order by week) as time_email_open_growth,
        time_email_clickthrough, time_email_clickthrough-lag(time_email_clickthrough) over(order by week) as time_email_clickthrough_growth
150
151
152 ⊝ (select week(occurred_at)as week,
153
        count(distinct user_id) as num_users,
         sum(if(action='sent_weekly_digest',1,0)) as time_weekly_digest_sent,
155
        sum(if(action='email_open',1,0)) as time_email_open,
        sum(if(action='email_clickthrough',1,0)) as time_email_clickthrough
156
157
        from email events
        group by 1
Export: Wrap Cell Content: IA
   week num_users time_weekly_digest_sent time_weekly_digest_sent_growth time_email_open_growth time_email_dickthrough_growth
                  908
                                                                310
                                                                                                  166
                  2602
                                                                              602
  20
                  2733
                                                                                                                    30
  21
         2926
                  2822
                                                                1014
  22
        3029
                                                                                                                    45
         3134
                  3003
                                                                1075
  24
  25
                  3207
                                                                1096
  26
        num_users time_weekly_digest_sent time_weekly_digest_sent_growth
                                                               time_email_open time_email_open_growth time_email_clickthrough
                                                                                                                   time_email_clickthrough_growth
  28
                                      100
                                                                             22
                                                                                                                   -22
  30
        3866
                  3706
                                      114
                                                               1383
                                                                             164
                                                                                                630
                                                                                                                   40
  31
                                                                                                                   -185
  32
        4023
                  3897
                                      104
                                                               1337
                                                                                                418
                                                                                                                   -27
  34
        4294
                  4111
                                                               1528
                                                                             96
                                                                                                490
  35
                                                                                                                   -452
```

### **Result:**

- Gained the confidence in writing window functions, CTE.
- Wherever possible tried to reduce the query execution time for each question by including where clause to filter the data as this data set in huge, grouping the results to reduce the row size.
- Avoided writing Subqueries as much as possible.