

# Project Description :-

This project involved using MySQL to analyze operational data and investigate metric spikes. The aim was to generate insights and answer queries related to the organization's operations.

#### **CASE STUDY 01:**

- A. Number of jobs reviewed: Amount of jobs reviewed over time. Your task: Calculate the number of jobs reviewed per hour per day for November 2020?
- B. Throughput: It is the no. of events happening per second. Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?
- C. Percentage share of each language: Share of each language for different contents. Your task: Calculate the percentage share of each language in the last 30 days?
- D. Duplicate rows: Rows that have the same value present in them. Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

#### **CASE STUDY 2:**

- A. User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service. Your task: Calculate the weekly user engagement?
- B. User Growth: Amount of users growing over time for a product. Your task: Calculate the user growth for product?
- C. Weekly Retention: Users getting retained weekly after signing-up for a product. Your task: Calculate the weekly retention of users-sign up cohort?
- D. Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly. Your task: Calculate the weekly engagement per device?
- E. Email Engagement: Users engaging with the email service. Your task: Calculate the email engagement metrics?

#### Approach:

Use SQL to create a database that can store the data and support the queries that we need to perform the analysis and load the data into the database. Use SQL queries to perform the analysis and retrieve the insights you want to extract.

#### **Tech-Stack Used:**

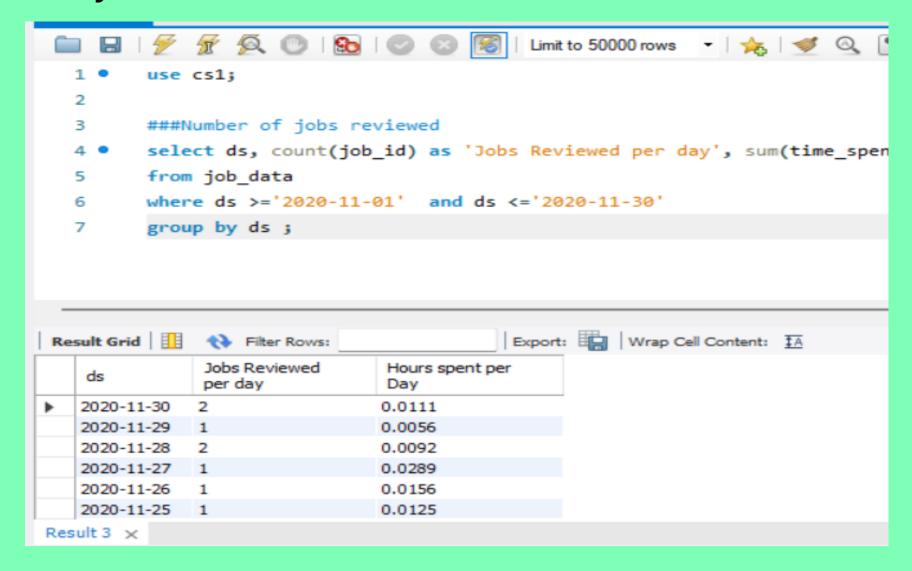
MySQL Workbench 8.0 CE

### **Insights:**

The queries were performed and recorded the observations based on the questions that were asked and the followings were results.

# Case Study 1 (Job Data)

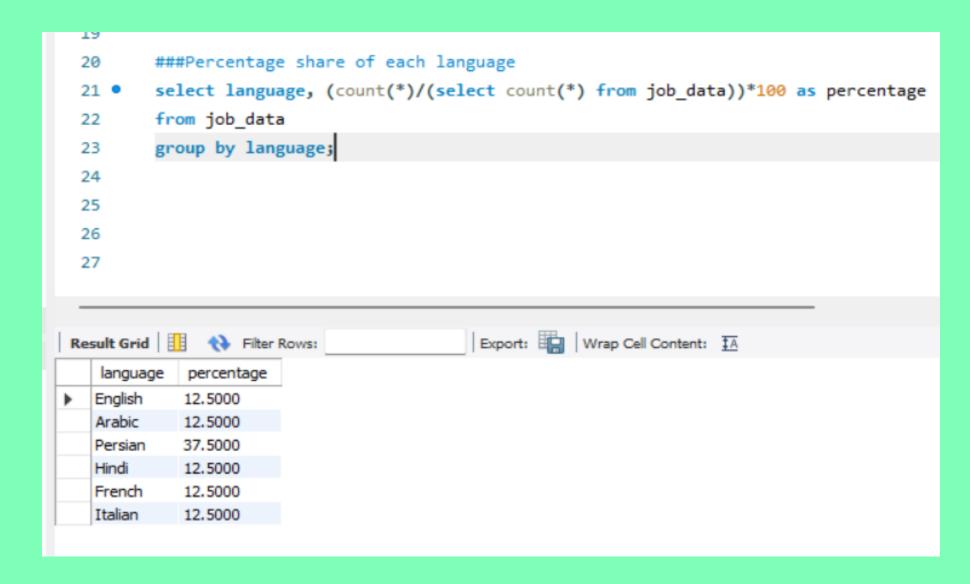
#### 1. Number of jobs reviewed:



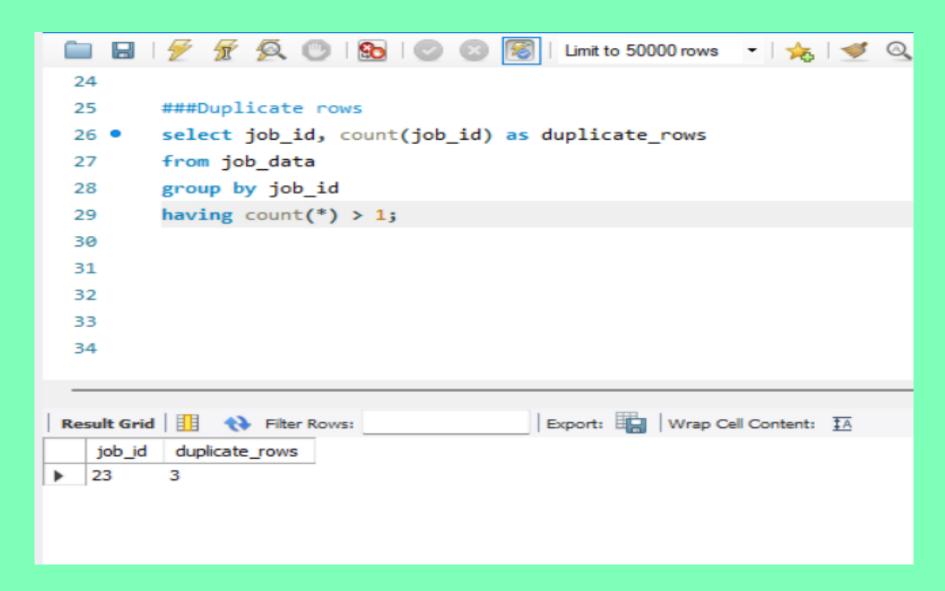
#### 2.Throughput:

```
###Throughput
 9
10 •
       SELECT ds,
       sum(jobs) over (order by ds rows between 6 preceding and current row) /
11
       sum(total_time) as throughput_7day_rolling_avg
12
13
       from
14
     15
       FROM job data where ds >= '2020-11-01' and ds <='2020-11-30'
       GROUP BY ds
16
17
       ) d
       group by ds;
18
 19
Result Grid Filter Rows:
                                     Export: Wrap Cell Content: TA
  ds
            throughput_7d_rolling_avg
  2020-11-25 0.0222
  2020-11-26 0.0357
  2020-11-27 0.0288
  2020-11-28 0.1515
                      0.1515
  2020-11-29 0.3000
  2020-11-30 0.2000
```

#### 3. Percentage share of each language:

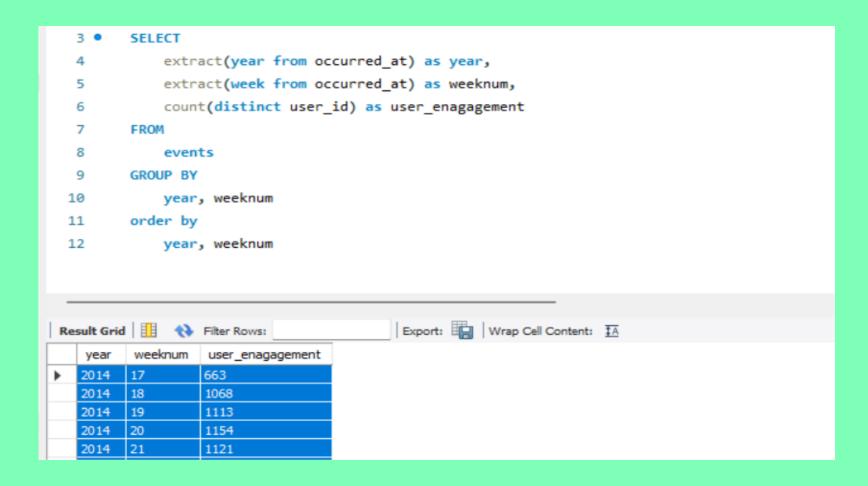


#### **4.Duplicate rows:**



# Case Study 2 (Investigating metric spike)

#### 1.User Engagement:



#### 2.User Growth:

```
3 •
        select
 4
        year_num,
        week_num,
 5
        num_active_users,

⇒ SUM(num_active_users)OVER(ORDER BY year_num, week_num ROWS BETWEEN

        UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_active_users
 8
 9
        from
     ⊖ (
 10
        select
 11
 12
        extract(year from activated at) as year num,
13
        extract(week from activated_at) as week_num,
14
        count(distinct user_id) as num_active_users
15
        from
 16
        users
        WHERE
17
        state = 'active'
 18
        group by year_num, week_num
 19
 20
        order by year num, week num
 21
        ) a;
                                       Export: Wrap Cell Content: IA
year_num week_num num_active_users cum_active_users
                     23
  2013
           0
                                   23
                                   53
  2013
                     30
  2013
                                   101
           3
                     36
                                   137
  2013
                     30
                                   167
  2013
           4
                     48
  2013
           5
                                   215
  2013
           6
                     38
                                   253
                     47
  2013
                                   295
```

#### 3. Weekly Retention:

11933

```
23 •
        SELECT
                                                                                                             29
                                                                                                                     SELECT
        distinct user_id,
 24
                                                                                                             30
                                                                                                                     a.user id,
 25
        COUNT(user_id),
                                                                                                                     a.signup_week,
                                                                                                             31
        SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as per_week_retention
                                                                                                                     b.engagement_week,
 26
                                                                                                             32
                                                                                                                     b.engagement_week - a.signup_week as retention_week
 27
        FROM
                                                                                                             33
                                                                                                             34
 29
        SELECT
                                                                                                             35
 30
        a.user_id,
                                                                                                                     (SELECT distinct user_id, extract(week from occurred_at) as signup_week from events_data
        a.signup_week,
                                                                                                                     WHERE event_type = 'signup_flow'
 31
                                                                                                             37
        b.engagement_week,
                                                                                                                     and event name = 'complete signup'
 32
                                                                                                             38
        b.engagement_week - a.signup_week as retention_week
                                                                                                                     and extract(week from occurred_at) = 18
 33
                                                                                                             39
                                                                                                                    )a
        FROM
 34
                                                                                                              40
 35
                                                                                                             41
                                                                                                                     LEFT JOIN
        (SELECT distinct user id, extract(week from occurred at) as signup week from events data
                                                                                                                     (SELECT distinct user_id, extract(week from occurred_at) as engagement_week FROM events_data
        WHERE event_type = 'signup_flow'
                                                                                                                     where event type = 'engagement'
 37
                                                                                                             43
        and event_name = 'complete_signup'
                                                                                                             44
                                                                                                                     )b
 38
        and extract(week from occurred_at) = 18
                                                                                                                     on a.user_id = b.user_id
 39
                                                                                                             45
                                                                                                              46
 40
                                                                                                                     )d
        LEFT JOIN
                                                                                                             47
        (SELECT distinct user_id, extract(week from occurred_at) as engagement_week FROM events_data
                                                                                                                     group by user id
        where event type = 'engagement'
                                                                                                                     order by user id;
 43
                                                                                                             49
 \Lambda \Lambda
Export: Wrap Cell Content: IA
                                                                                                            Export: Wrap Cell Content: IA
   user_id COUNT(user_id) per_week_retention
                                                                                                               user_id COUNT(user_id) per_week_retention
  11926
                                                                                                               11936
                                                                                                               11939 3
  11928
                                                                                                               11940
  11929
                                                                                                                                    0
  11931
                                                                                                               11942
```

11944 1

0

## 4. Weekly Engagement:

```
50
 51 •
         SELECT
         extract(year from occurred_at) as year_num,
 52
         extract(week from occurred_at) as week_num,
 53
         device,
 54
         COUNT(distinct user_id) as no_of_users
 55
 56
         FROM
 57
         events data
 58
         where event type = 'engagement'
         GROUP by 1,2,3
 59
         order by 1,2,3;
 60
 61
 62
 63
 64
Result Grid Filter Rows:
                                             Export: Wrap Cell Content: IA
            week_num
                        device
                                           no_of_users
   year_num
  2014
             17
                       acer aspire desktop
                                           2
  2014
             17
                       acer aspire notebook
                                           2
             17
                       amazon fire phone
  2014
                       asus chromebook
  2014
             17
                                           3
   2014
             17
                        dell inspiron desktop
```

#### **5.Email Engagement:**

```
DI
 62 •
        SELECT
 63
        engagements,
 64
        total users,
        engagements/total_users*100 AS engagement_rate
 65
 66
      from ( SELECT
 67
                     count(distinct(user id)) AS total users,

    ○ COUNT(DISTINCT)

 68
 69
      70
        WHEN action = 'email open' THEN user id
       WHEN action = 'email_clickthrough' THEN user_id END) AS engagements
 71
 72
                 FROM email events) as counts;
 73
 74
 75
Result Grid
                                          Export: Wrap Cell Content: TA
             Filter Rows:
   engagements
              total_users
                         engagement_rate
  5927
              6179
                         95.9217
```

## Result:

Extracted many useful analysis that could help in making data driven decisions making for the business.

In this task, I have many basics as well as advanced concepts related to SQL and this project helped me to understand those concepts experimentally.

# All the tables and data that were extracted where attached in the following sheet:

