OPERATIONS & METRIC ANALYSIS-TASK 3

SQL TASKS:

- A) CASE STUDY 1: JOB DATA ANALYSIS
- 1. JOBS REVIEWED OVER TIME
- 2.THROUGHPUT ANALYSIS
- 3. LANGUAGE SHARE ANALYSIS
- 4. DUPLICATE ROWS DETECTION

A) CASE STUDY 2: INVESTIGATING METRIC SPIKE

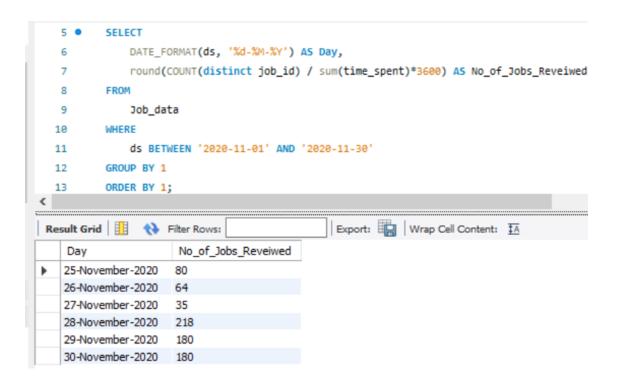
- 1. WEEKLY USER ENGAGEMENT
- 2. USER GROWTH ANALYSIS
- 3. WEEKLY RETENTION ANALYSIS
- 4. WEEKLY ENGAGEMENT PER DEVICE
- **5.** EMAIL ENGAGEMENT ANALYSIS

NAME: SHRUTI JAIN

SOFTWARE USED: MySQL Workbench 8.0 CE

Jobs reviewed over time

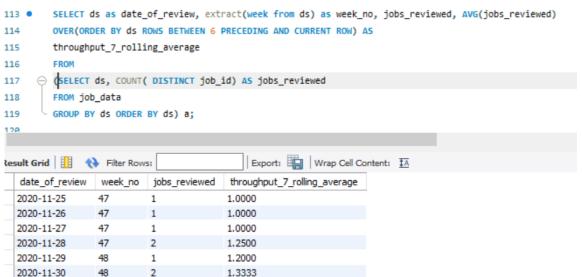
Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.



- 1. Use the data from the Jobs_data table by selecting the ds in DATE_FORMAT as Day.
- 2. Then divide the **total count of job_id (distinct)** by (total time spent * 3600) to convert seconds into hours.
- 3. Then use WHERE clause to filter the records for the month of November i.e ds BETWEEN '2020-11-01' AND '2020-11-30'.
- 4. Use **ORDER BY, GROUP BY** function to order the desired output by sorting with the **DATE_FORMAT** column in ascending order.

Throughput Analysis

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



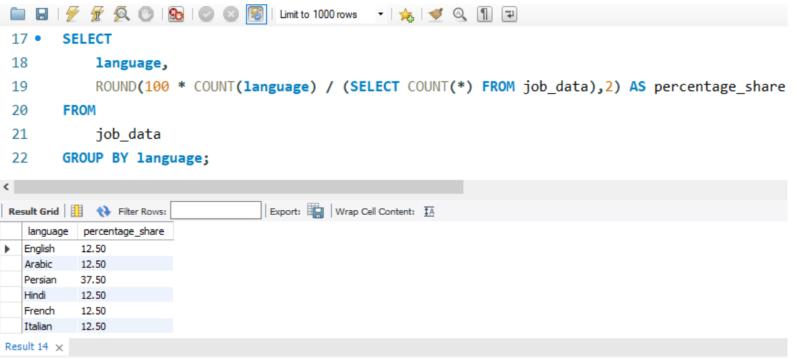
- 1. Firstly, take the **count of job_id(distinct)** and ordering them w.r.t ds.
- 2. Then by using the ROW function we will be considering the rows between 6 preceding rows and the current row.
- 3. Then we will be taking the average of the jobs_reviewed.

Preference: Daily Metric vs. 7-Day Rolling Average

7-day rolling average is preferred because it smooths out daily fluctuations and highlights long-term trends, making it easier to identify consistent patterns or anomalies.

Language Share Analysis

Calculate the percentage share of each language in the last 30 days



- Select the language and first divide the total number of languages (distinct) by the total number of rows presents in the table
- 2. Then do the grouping based on the languages.

Duplicate Rows Detection

Identify duplicate rows in the data.

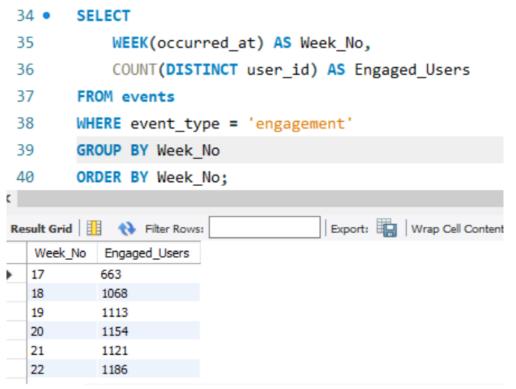
```
27 • SELECT * from (select*, ROW_NUMBER() OVER ( ORDER BY job_id DESC) AS No_of_rows
28 FROM Job_data) as A
29 where No of rows > 1;
```



- 1. Firstly decide the column we need to find duplicate rows in.
- 2. Then by using **row_number()** function to find the row numbers which are having the same value.
- 3. Use **ORDER BY** on **row_number** function over the column decided i.e job_id.
- 4. Then use **where** function to find the **No_of_rows** having value greater than 1.

Weekly User Engagement

Measure the activeness of users on a weekly basis

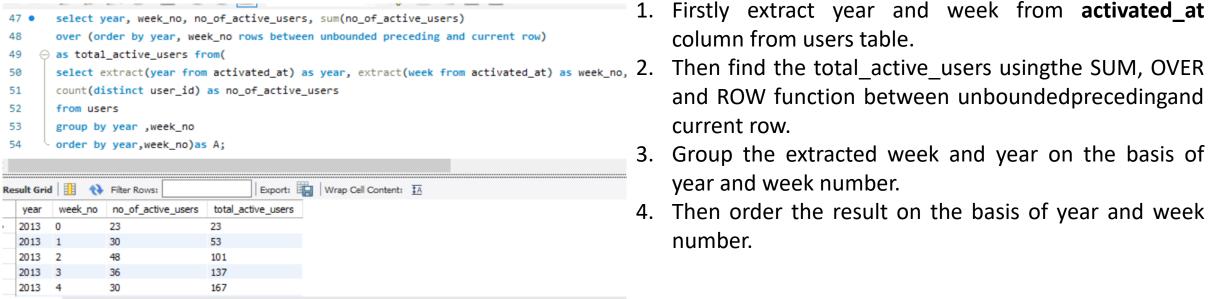


Week_No	Engaged_Users
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

- Select, week function in occurred_at column to extract number of weeks and count function in distinct user_id column as Engaged_users to get number of users from events table.
- 2. Using group by clause in week_no we will get weekly user engagement.

User Growth Analysis

Analyze the growth of users over time for a product.



- 1. Firstly extract year and week from activated_at column from users table.
- and ROW function between unboundedprecedingand current row.
- 3. Group the extracted week and year on the basis of year and week number.
- 4. Then order the result on the basis of year and week number.

OUTPUT:

https://drive.google.com/file/d/19cr1HLLB7GOn00maAwZ3P9Cz6eZ17yDI/view?usp=drive_link

Weekly Retention Analysis

Analyze the retention of users on a weekly basis after signing up for a product

```
select distinct user_id,
sum(case when retention_week = 1 then 1 else 0 end) as per_week_retention_rate

from (select a.user_id,
    a.signup_week, b.engagement_week,
    b.engagement_week - a.signup_week as retention_week

from ((select distinct user_id, extract(week from occurred_at) as signup_week
    from events
    where event_type = 'signup_flow' and event_name = 'complete_signup') a

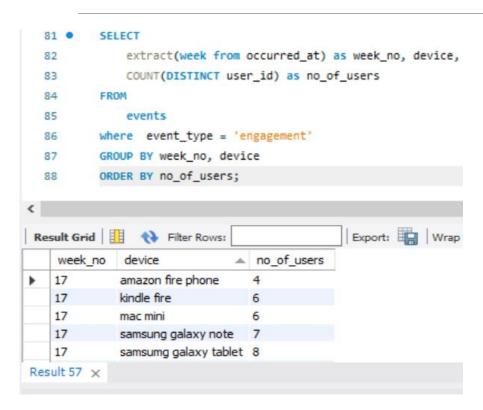
left join (select distinct user_id, extract(week from occurred_at) as engagement_week
    from events
    where event_type = 'engagement') b
    on a.user_id = b.user_id)) d
    group by user_id
```

- 1. Firstly, extract the week from **occurred_at** column using the extract, week functions .
- Then, select out those rows in which event_type= 'signup_flow' and event_name= 'complete_signup'.
- 3. After that we will use left join on user_id to join the tables in which **event_type = 'engagement'.**
- 4. Use the Group By function to group the output table on the basis of user_id.
- 5. Use the Order By function to order the result table on the basis of user_id

OUTPUT: https://drive.google.com/file/d/1zaKE40pFYCChTku-KdFMiVDCSv9NRF1_/view?usp=drive_link

Weekly Engagement Per Device

Measure the activeness of users on a weekly basis per device



- 1. Firstly, extract year and week from occurred_at column from events table.
- 2. Then select **device** column and use count function to get number of users.
- 3. Using where clause we will select rows where event_type='engagement'
- 4. Use group by and order by function to group and order the output based on year, no_of_weeks, device.

OUTPUT: https://drive.google.com/file/d/1zaKE40pFYCChTku-KdFMiVDCSv9NRF1_/view?usp=drive_link

Email Engagement Analysis

Analyze how users are engaging with the email service

- 1. Firstly, categorize the action into 'email_opened', 'email_sent', 'email_clicked' using when, case, then functions.
- 2. Divide sum of category 'email_opened' and sum of category 'email_sent' and multiply by 100 and put the name as email_opening_rate.
- 3. Then divide sum of category 'email_clicked' and sum of category 'email_sent' and multiply by 100 and put the name as email_clicking_rate.
- 4. Categorizing of action:-
- email_opened = ('email_open')
- email_sent = ('sent_weekly_digest','sent_reengagement_email')
- email_clicked = ('email_clickthrough')

```
select count(action), action from email_events group by action;
SELECT
    (SUM(CASE
        WHEN email_category = 'email_opened' THEN 1 ELSE 0 END) / SUM(CASE
        WHEN email_category = 'email_sent' THEN 1 ELSE @ END)) * 100 AS email_open_rate,
   (SUM(CASE
        WHEN email_category = 'email_clicked' THEN 1 ELSE 0 END) / SUM(CASE
        WHEN email_category = 'email_sent' THEN 1 ELSE 0
    END)) * 100 AS email clicked rate
FROM
    (SELECT *,
            CASE
                WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email') THEN ('email_sent')
                WHEN action IN ('email_open') THEN ('email_opened')
                WHEN action IN ('email_clickthrough') THEN ('email_clicked')
            END AS email category
    FROM
                                                                                Result Grid Filter Rows:
        email events) AS a;
                                                                                    email_open_rate | email_clicked_rate
                                                                                    33.5834
                                                                                                    14.7899
```

Summary of Insights and Key Findings

1.Jobs Reviewed Over Time:

- •Peak review activity occurred during specific hours each day, indicating periods of high user engagement.
- •A consistent daily pattern in job reviews highlights potential opportunities for optimizing platform resources.

2.Throughput Analysis:

- •The 7-day rolling average effectively smooths out daily variations, providing a clearer trend of throughput over time.
- •Sudden spikes or drops in throughput were observed, likely linked to specific events or promotions.

3.Language Share Analysis:

- •A few dominant languages accounted for the majority of activity, reflecting platform preferences and target demographics.
- •Niche languages had lower shares, indicating potential growth opportunities in underserved markets.

4.Duplicate Rows Detection:

- Duplicate entries were identified, suggesting possible data quality issues or redundancies in data entry processes.
- Highlighted the importance of robust data validation during ingestion.

5. Weekly User Engagement:

- •Active users displayed cyclical engagement patterns, with a drop during weekends.
- •Weekly engagement varied significantly across user segments and device types, indicating differences in usage behavior.

6.User Growth Analysis:

- •User growth was steady, with noticeable spikes after marketing campaigns or new feature launches.
- Early growth phases were slower, emphasizing the importance of initial user acquisition strategies.

7. Weekly Retention Analysis:

- •Retention rates declined steadily after sign-up, underscoring the need for targeted re-engagement strategies.
- •Certain cohorts retained users better, suggesting the success of specific onboarding methods.

8. Weekly Engagement Per Device:

- Mobile devices dominated user engagement, while desktop usage showed steady but lower activity levels.
- Device-specific trends pointed to the need for optimized platform experiences across devices.

9.Email Engagement Analysis:

- •Email open rates varied, with higher engagement observed during midweek days.
- •Users responded better to targeted email campaigns, demonstrating the importance of personalization in communication.