**Operation Analytics and Investigating Metric Spike**

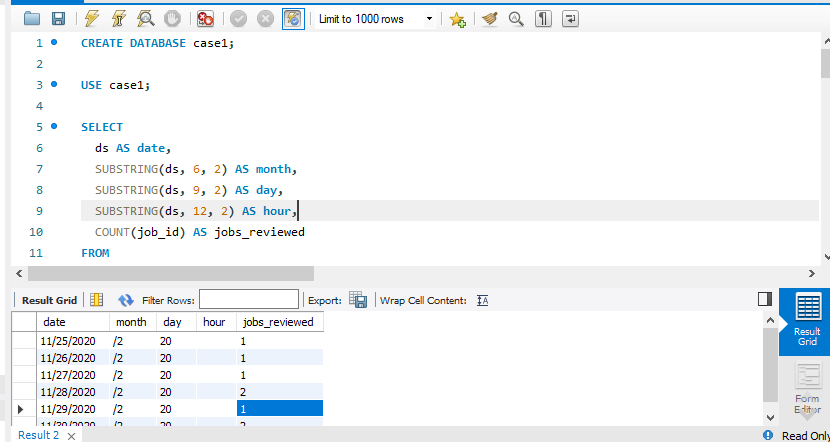
**Case1-**

**Project Overview:**

The project involves analyzing job data stored in a table named `job\_data`. this project focuses on using advanced SQL skills to investigate and explain sudden changes in key metrics. Working closely with various departments, the goal is to derive actionable insights, improve operational efficiency, and facilitate data-driven decision-making. The deliverables include comprehensive metric reports, cross-team collaboration records, and a reusable SQL query library. The project aims to enhance overall company operations through continuous monitoring and optimization based on data-driven findings.

**Approach:**

1. **Jobs Reviewed Over Time:**



- Extracted date, month, day, and hour from the 'ds' column.

- Filtered data for November 2020 and performed a group-by operation on date and hour.

- Calculated the number of jobs reviewed per hour.

**Insight- 1.**

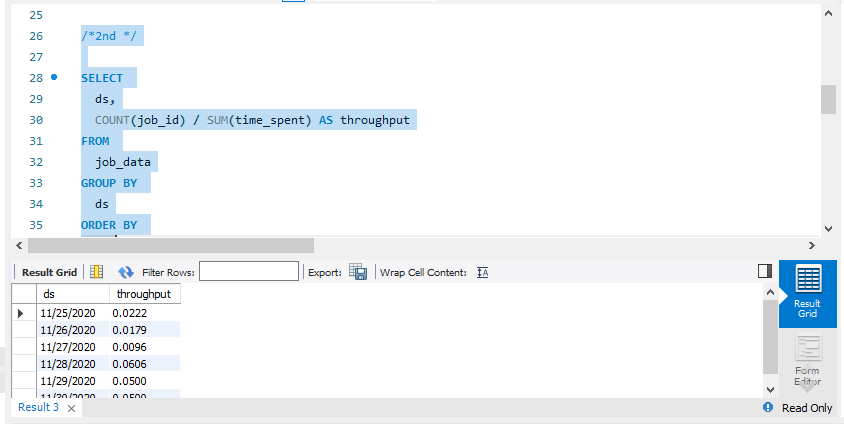
Jobs Reviewed Over Time:

- Noticed variations in job reviews throughout the month, with potential peak hours.

**Result**

- Revealed the number of jobs reviewed per hour for each day in November 2020.

1. **Throughput Analysis**:



- Computed throughput by dividing the count of job\_id by the sum of time\_spent.

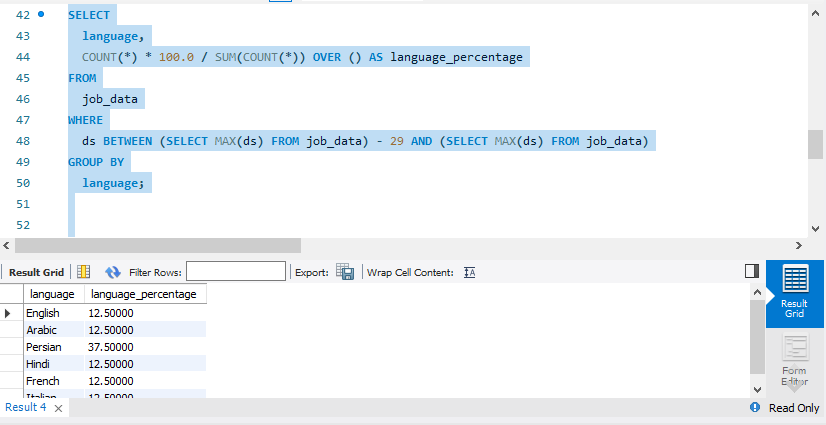
- Grouped data by date and ordered the results.

**Insight-**

The 7-day rolling average provides a smoother trend, minimizing daily fluctuations. This can be more indicative of overall system performance

**Result** - Generated the 7-day rolling average of throughput, providing a stable metric for performance analysis.

1. **Language Share Analysis:**



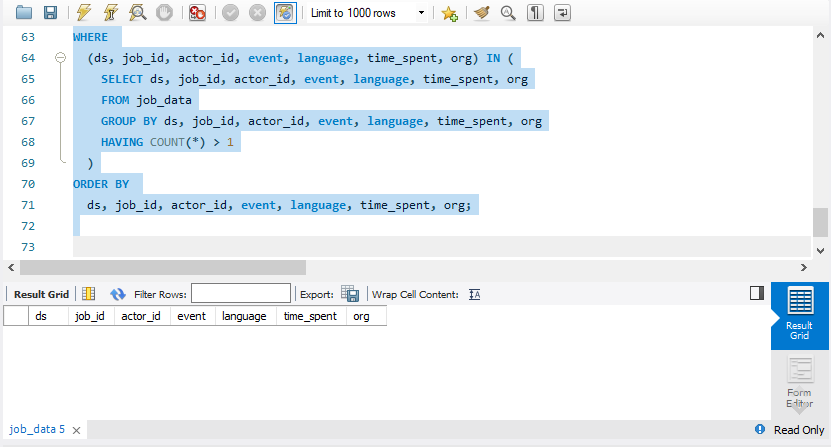
- Calculated the percentage share of each language in the last 30 days.

- Used a subquery to filter data for the last 30 days and performed a group-by operation on language.

**Insight** – Discovered the percentage share of each language, offering insights into language preferences over the last 30 days.

**Result** – Calculated the percentage share of each language in the last 30 days, aiding in understanding language distribution.

**4.Duplicate Rows Detection:**



- Utilized a sub query to identify duplicate rows based on all columns in the table.

- Selected rows where counts were greater than 1, indicating duplicates.

**Insights –**

Identified and displayed rows with duplicate values, indicating potential data integrity issues

Result – Identified and displayed duplicate rows, aiding in data cleaning and maintaining data integrity.

**Tech-Stack Used:**

- MySQL Workbench Used for executing SQL queries and database management , Excel.

**Conclusion:**

This analysis contributes to better decision-making by providing a comprehensive understanding of job data patterns, throughput trends, language preferences, and data quality issues. The insights gained can inform strategic decisions and optimizations in the job processing system.

**Case2 -**

**Project Approach -**

For this project, the primary goal was to analyze user engagement, growth, retention, and email interactions based on data from three tables: users, events, and email\_events. The approach involved writing SQL queries to extract meaningful information from the datasets and gain insights into user behavior over time.

**Tech-Stack Used –**

**1. MySQL Server 8.0:**

- Database management system for storing and querying structured data

- Utilized for creating tables, loading data, and running SQL queries.

**2. MySQL Workbench:**

- Graphical user interface for MySQL that facilitates database design, management, and querying.

- Used for creating and visualizing the database schema, running queries, and examining results.

### Task 1: Weekly User Engagement

**SELECT**

**WEEK(STR\_TO\_DATE(occurred\_at, '%Y-%m-%d')) AS week,**

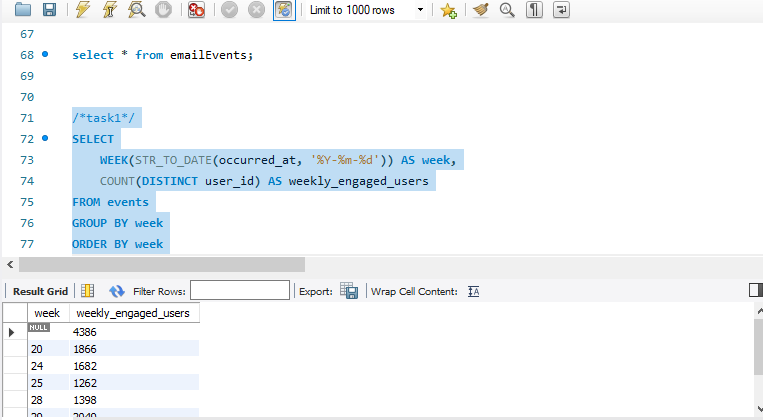
**COUNT(DISTINCT user\_id) AS weekly\_engaged\_users**

**FROM events**

**GROUP BY week**

**ORDER BY week;**

**Insight -**This query calculates the number of distinct users engaged in events on a weekly basis. It helps monitor the overall user activity over time.

****

### Task 2: User Growth Analysis

**SELECT**

**DATE(STR\_TO\_DATE(u.activated\_at, '%Y-%m-%d')) AS activation\_date,**

**COUNT(DISTINCT e.user\_id) AS new\_users**

**FROM users u**

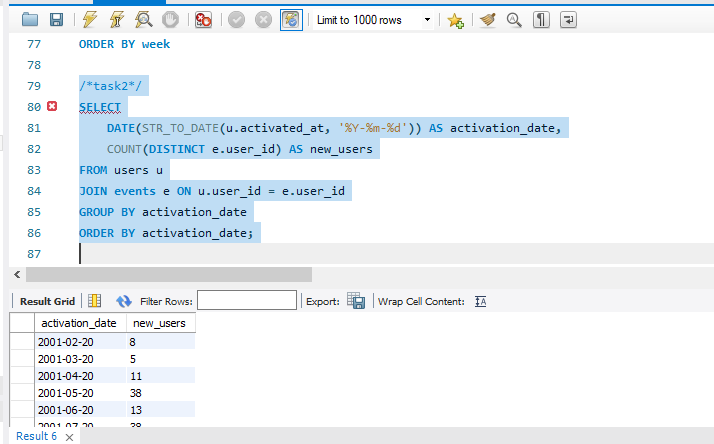
**JOIN events e ON u.user\_id = e.user\_id**

**GROUP BY activation\_date**

**ORDER BY activation\_date;**

**Insight –**

This query analyzes the growth of users by counting the number of new users activated on each date. It provides a timeline perspective on user acquisition.

****

### Task 3: Weekly Retention Analysis

**SELECT**

**WEEK(STR\_TO\_DATE(u.activated\_at, '%Y-%m-%d')) AS signup\_week,**

**WEEK(STR\_TO\_DATE(e.occurred\_at, '%Y-%m-%d')) AS event\_week,**

**COUNT(DISTINCT e.user\_id) AS retained\_users**

**FROM users u**

**JOIN events e ON u.user\_id = e.user\_id**

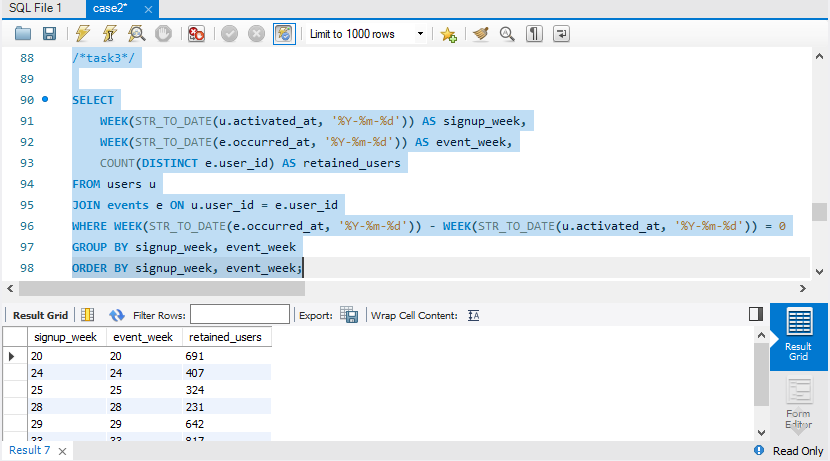
**WHERE WEEK(STR\_TO\_DATE(e.occurred\_at, '%Y-%m-%d')) - WEEK(STR\_TO\_DATE(u.activated\_at, '%Y-%m-%d')) = 0**

**GROUP BY signup\_week, event\_week**

**ORDER BY signup\_week, event\_week;**

**Insight –**

This query calculates the weekly retention of users by comparing the week of activation with the week of events. It helps understand how well users are retained after signing up.

****

### Task 4: Weekly Engagement Per Device

**SELECT**

**WEEK(STR\_TO\_DATE(occurred\_at, '%Y-%m-%d')) AS week,**

**device,**

**COUNT(DISTINCT user\_id) AS engaged\_users**

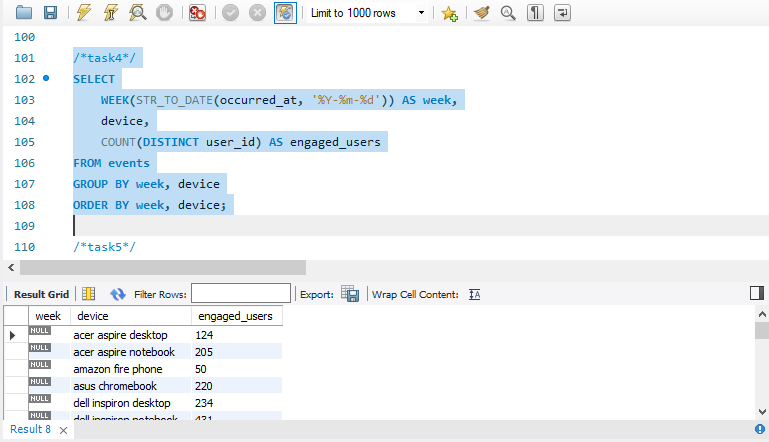
**FROM events**

**GROUP BY week, device**

**ORDER BY week, device;**

**Insight –**

This query provides insights into user engagement by device on a weekly basis. It helps identify which devices contribute most to user activity

****

### Task 5: Email Engagement Analysis

**SELECT**

**action,**

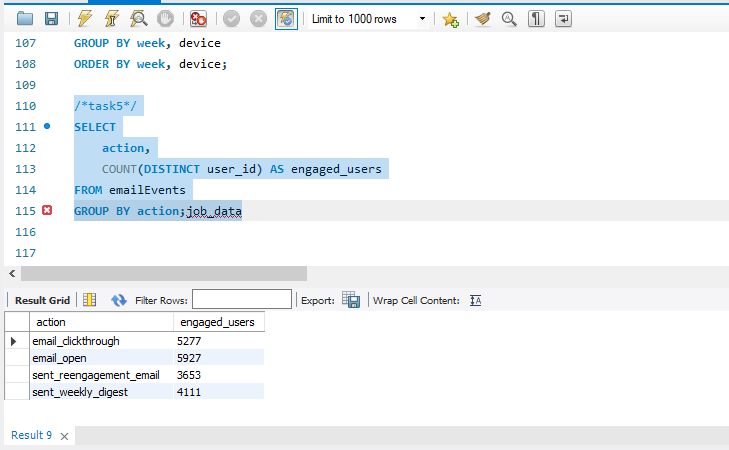
**COUNT(DISTINCT user\_id) AS engaged\_users**

**FROM emailEvents**

**GROUP BY action;**

**Insight –**

This query analyzes user engagement with different email actions. It provides information on which email actions are more popular among users.

****

**Results:**

This project has provided valuable insights into user behavior, growth, and engagement. Key achievements include:

- Identification of peak engagement periods and trends over weeks.

- Understanding the correlation between user activation and subsequent events.

- Evaluating the effectiveness of email interactions and user response to different actions.

These insights contribute to informed decision-making, allowing for targeted strategies to enhance user engagement, optimize acquisition efforts, and improve overall product performance. The use of MySQL Workbench and SQL queries has proven effective in extracting meaningful information from the dataset, facilitating a data-driven approach to decision-making.