**CASE Study1: Job Data Analysis**

**Project Description:**

The project stresses on analysis of job data for insights into the job review patterns, throughput, and language share analysis.

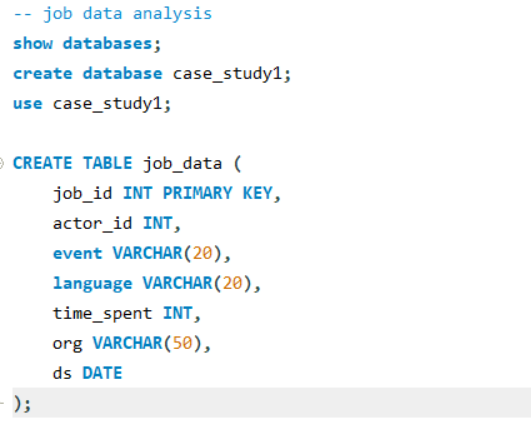
**Approach:**

1. Created the table job\_data with columns job\_id, actor\_id, event, language, time\_spent, org, and ds using the database case\_study1.
2. Entries are added randomly utilizing the SOL functions.
3. Executed SQL queries for each task.
4. Reviewed obtained outputs to identify patterns.
5. Drew insights and interpretations from the results.

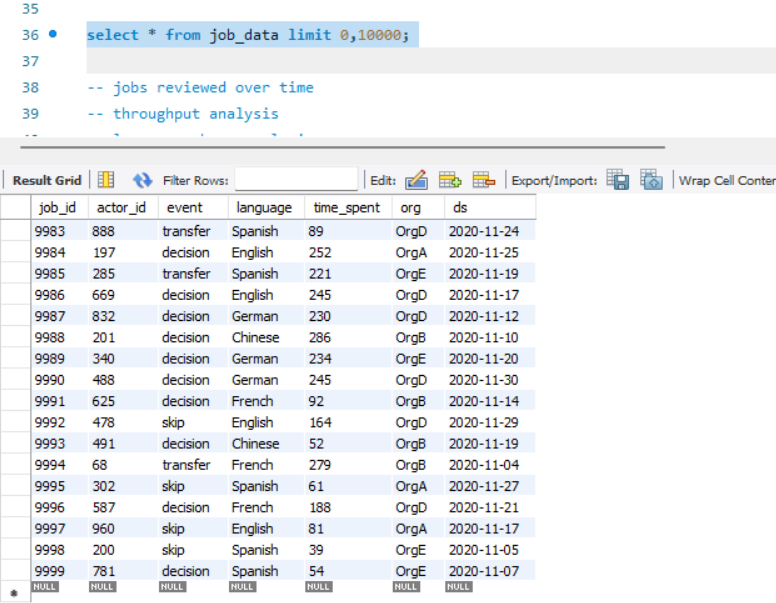
**Tech-Stack Used:**

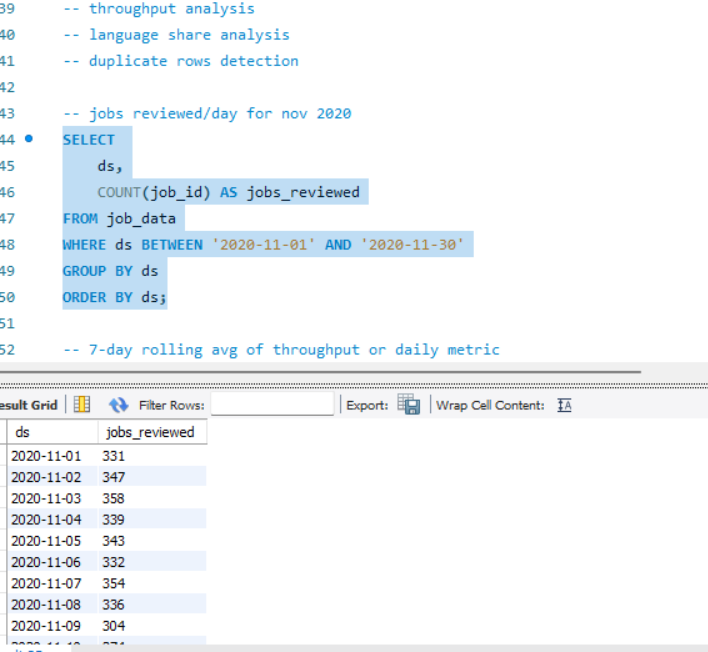
1. MySQL Workbench: For executing the SQL queries.
2. SQL: For data analysis.

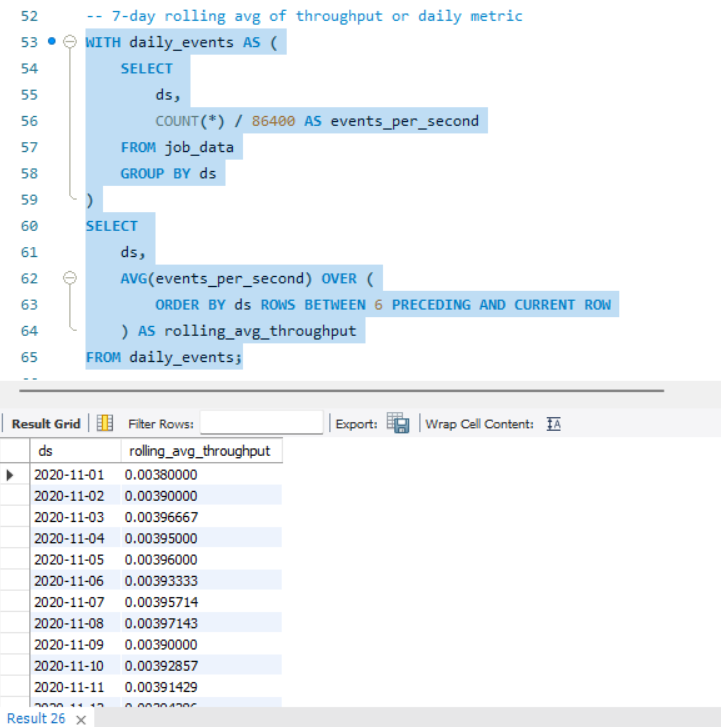
**Queries:**

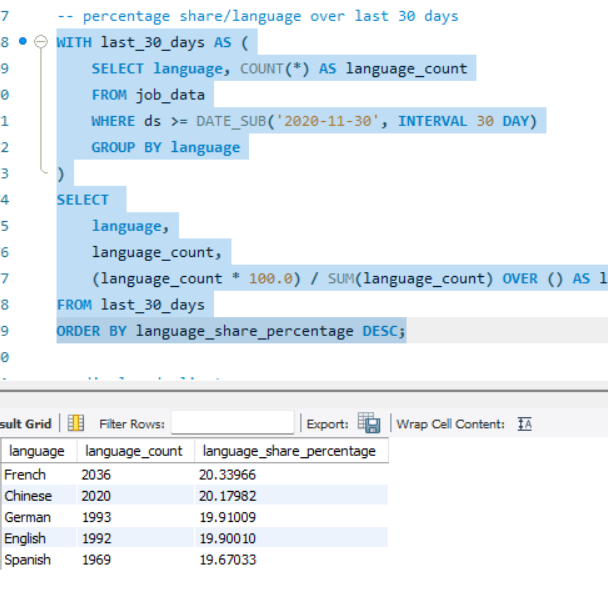
****

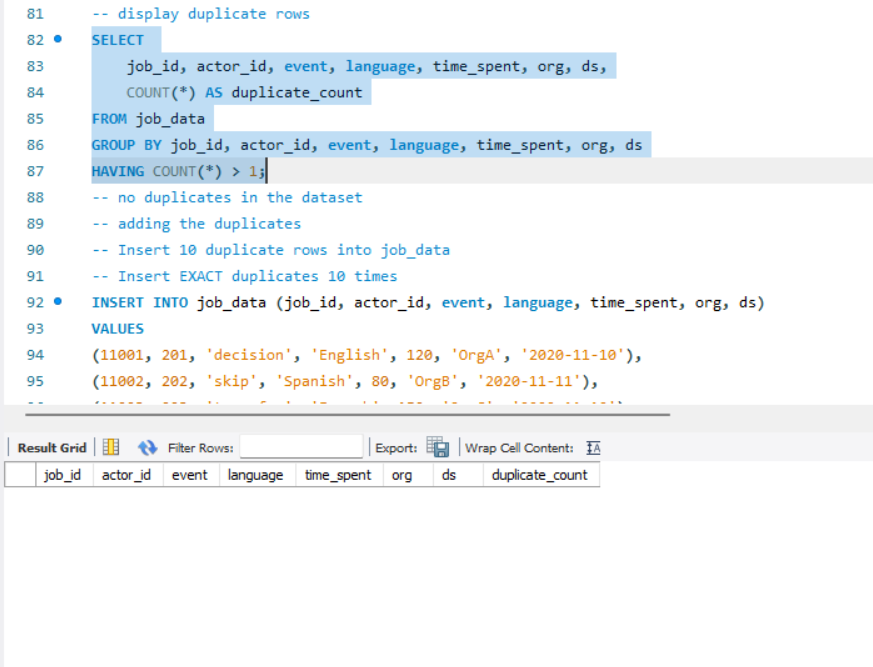
****

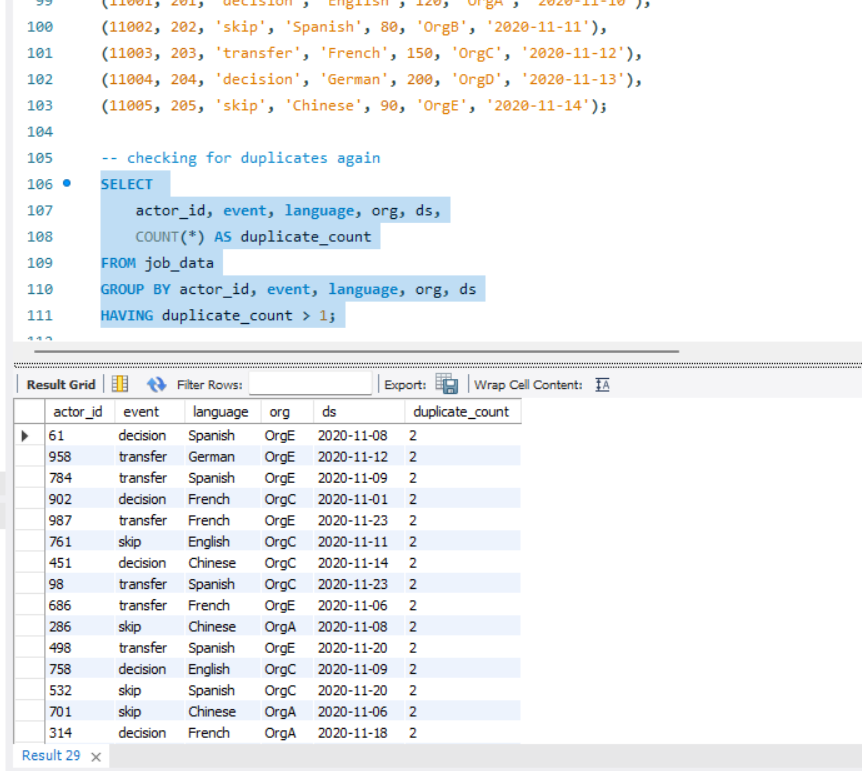
****

****

****

****

****

****

**Insights:**

1. Jobs reviewed/day: the daily review count shows job activity patterns. Spikes indicate high-priority days/ system anomalies while dips highlight holidays/outages.
2. Throughput analysis: the rolling average overcome daily fluctuations, hence providing long-term patterns. It is preferred over daily metrics for identifying consistent patterns whereas in case of spotting sudden spikes or dips, daily metric should be preferred.
3. Language share analysis: highest for French, least for Spanish.
4. Duplicate rows detection: for data inconsistency identification. There were no duplicate rows initially, so after inserting exact duplicates, checking whether the query works correctly. It did.

**Results:**

1. Developed and executed SQL queries for all tasks.
2. Provided insights into job review patterns, throughput trends, and language shares.
3. Validated duplicate detection with sample data.

Case study2: Investigating metric spike

**Project description:**

This project focuses on analysing user engagement, growth, retention, and email activity to identify and investigate metric spikes. The goal is to uncover meaningful insights that can guide business decisions and product improvements. SQL queries were used to extract and analyse data from three key datasets: users, events, and email\_events.

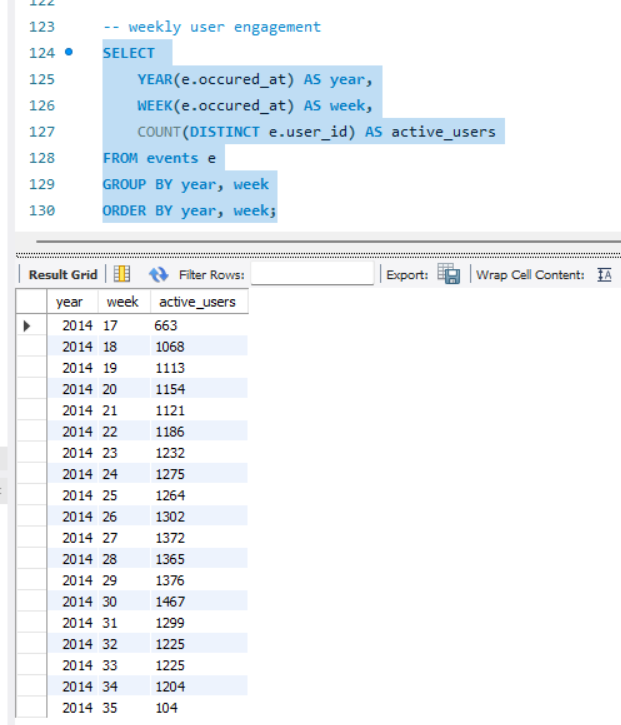
**Approach:**

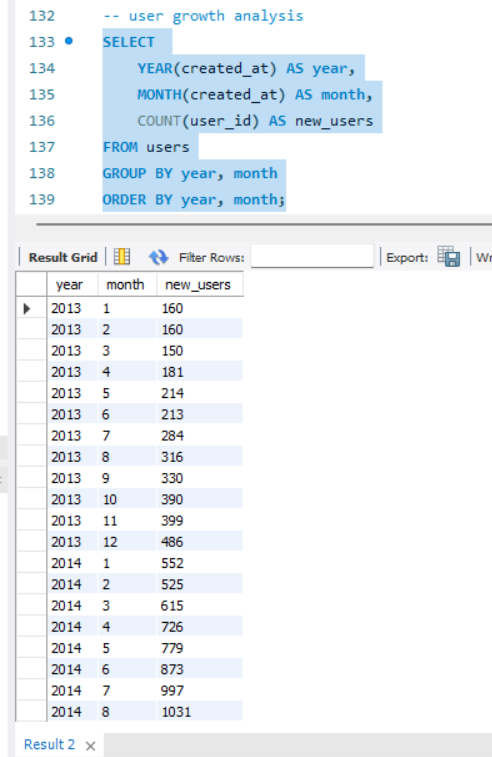
1. Data Exploration: Reviewed the structure of the three datasets:
   * users: Contains user information, including signup dates.
   * events: Logs user actions, including timestamps and device information.
   * email\_events: Tracks email interactions with columns for user actions and timestamps.
2. SQL Query Development: Constructed queries to address key metrics such as user engagement, growth, and email interaction.
3. Results Analysis: Evaluated outputs for trends, spikes, and anomalies.
4. Report Compilation: Documented queries, outputs, and insights.

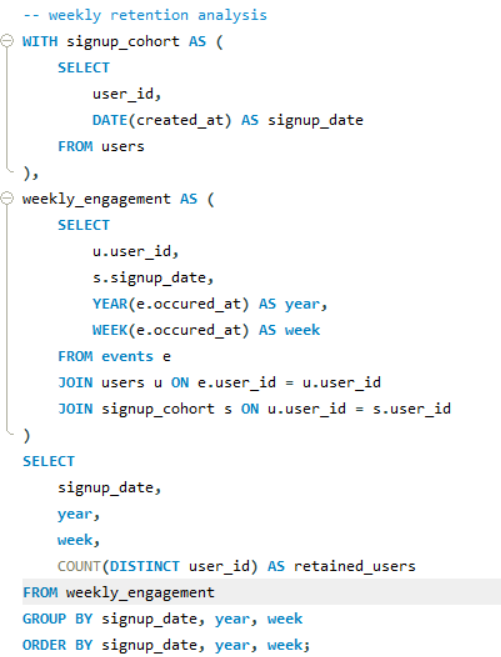
**Tech-stack used:**

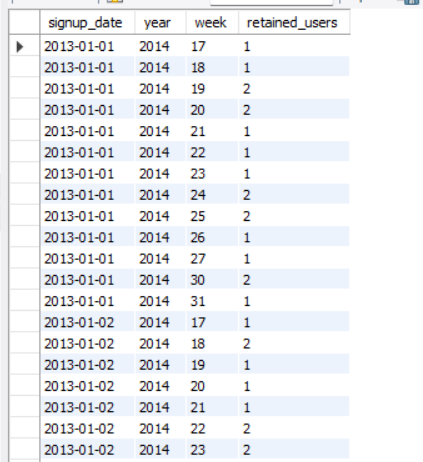
1. MySQL Workbench: For writing and executing SQL queries.
2. SQL: Structured Query Language for data analysis

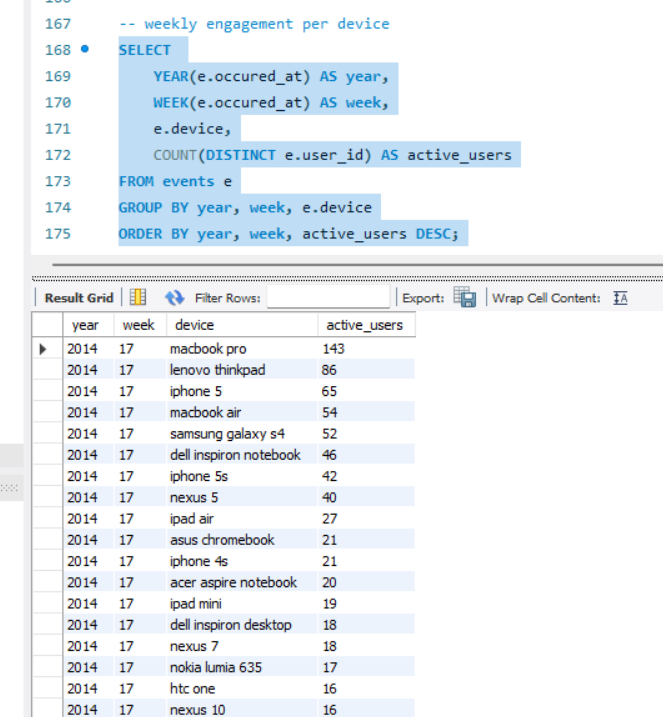
**Queries:**

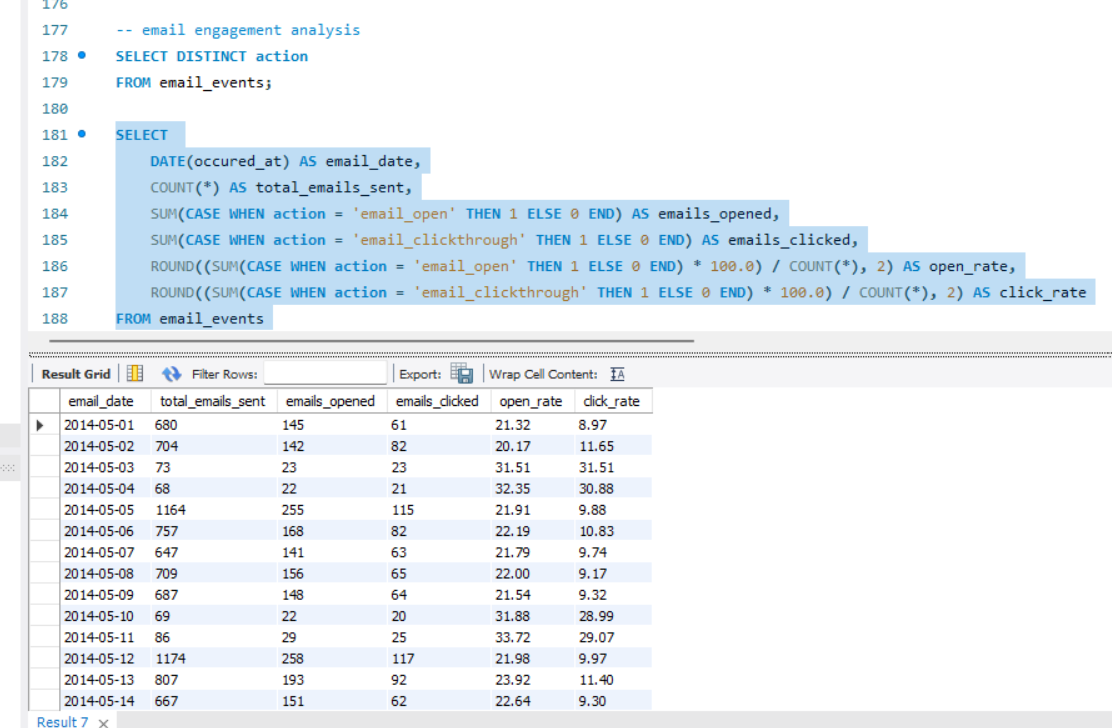
****

****

****

****

****

****

**Insights:**

1. Weekly user engagement trends highlight active user patterns. Spikes may indicate successful feature launches or marketing campaigns.
2. Growth trends show the pace at which users join the platform. Sharp increases may point to effective onboarding strategies or viral events.
3. Retention metrics help identify how well the platform retains users after sign-up. Strong weekly retention suggests user satisfaction and engagement.
4. Device-level engagement provides insights into user preferences, helping prioritize mobile or desktop experiences.
5. Tracking email action rates evaluates the effectiveness of email campaigns. Low engagement may signal the need for improved content or targeting.

**Results:**

1. Successfully executed SQL queries to investigate user engagement, growth, retention, and email activity.
2. Identified potential metric spikes linked to user actions and email interactions.
3. Delivered actionable insights for product and marketing teams.