

1: Jobs Reviewed Over Time

SELECT

DATE_FORMAT(ds, '%Y-%m-%d %H:00') AS hour,

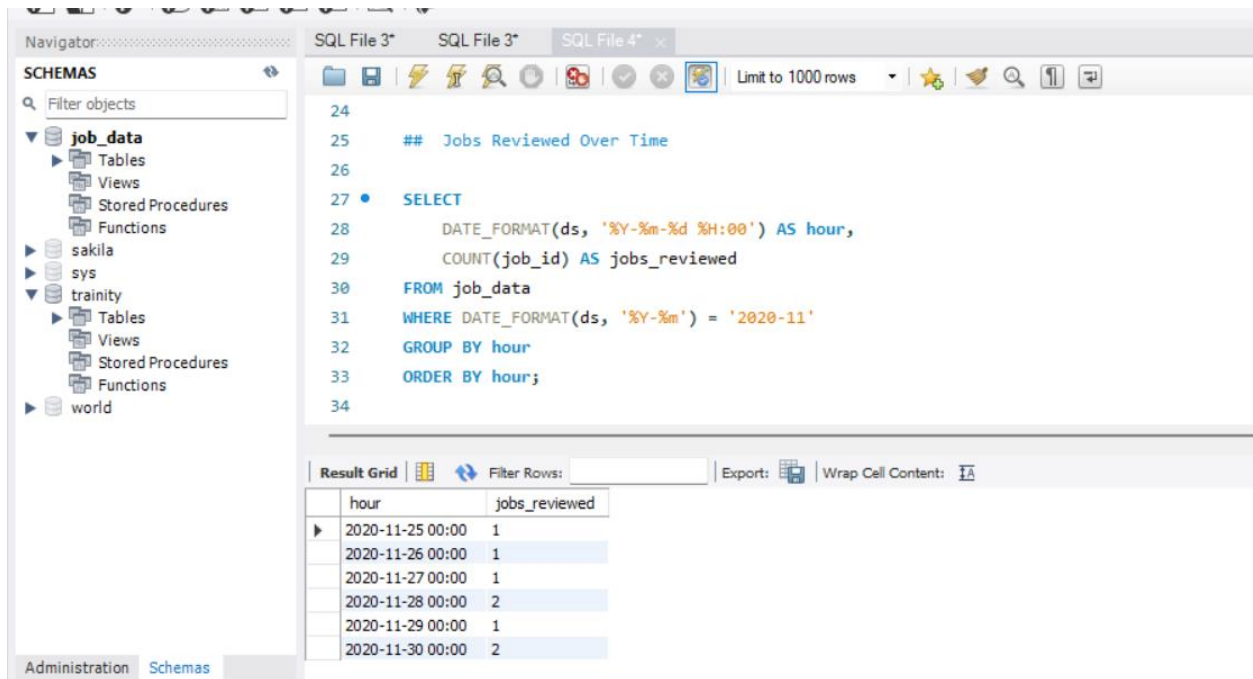
COUNT(job_id) AS jobs_reviewed

FROM job_data

WHERE DATE_FORMAT(ds, '%Y-%m') = '2020-11'

GROUP BY hour

ORDER BY hour;



The screenshot shows a SQL IDE interface with a Navigator on the left, a SQL editor in the center, and a Result Grid at the bottom. The Navigator shows a tree structure with schemas: job_data, sakila, sys, trainity, and world. The SQL editor contains the following query:

```
24
25  ## Jobs Reviewed Over Time
26
27  • SELECT
28      DATE_FORMAT(ds, '%Y-%m-%d %H:00') AS hour,
29      COUNT(job_id) AS jobs_reviewed
30  FROM job_data
31  WHERE DATE_FORMAT(ds, '%Y-%m') = '2020-11'
32  GROUP BY hour
33  ORDER BY hour;
34
```

The Result Grid shows the following data:

hour	jobs_reviewed
2020-11-25 00:00	1
2020-11-26 00:00	1
2020-11-27 00:00	1
2020-11-28 00:00	2
2020-11-29 00:00	1
2020-11-30 00:00	2

2: Throughput Analysis

WITH daily_throughput AS (

SELECT

ds,

```

COUNT(*) AS total_events,

SUM(time_spent) AS total_time_spent

FROM job_data

GROUP BY ds

),

throughput_with_rolling_avg AS (

SELECT

ds,

total_events / total_time_spent AS throughput_per_second,

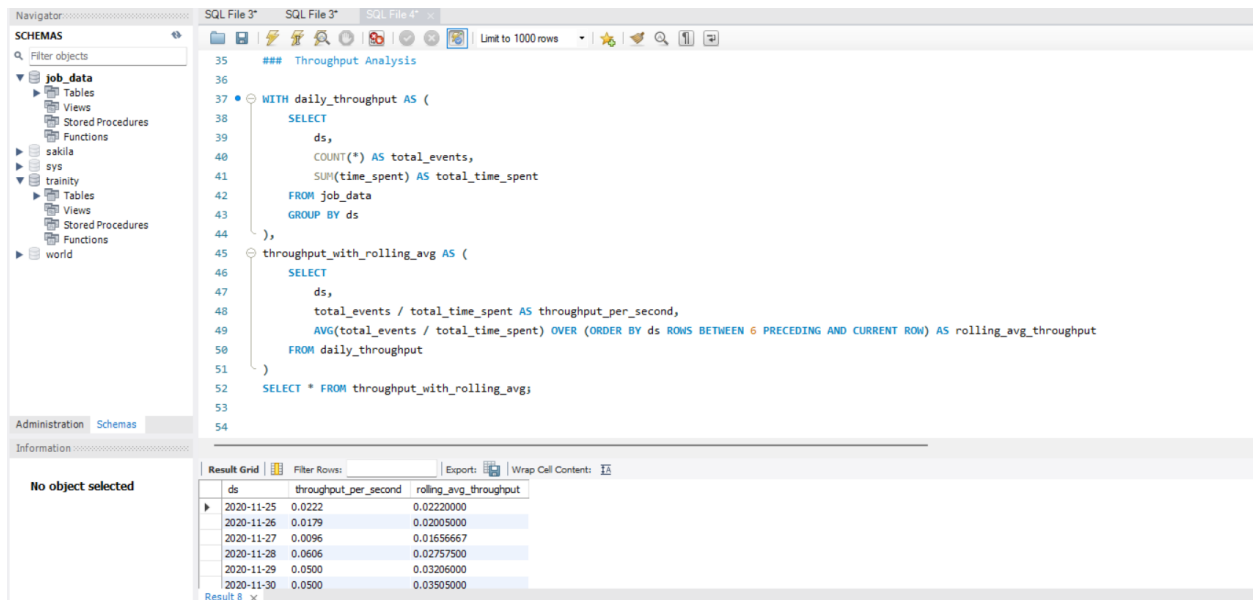
AVG(total_events / total_time_spent) OVER (ORDER BY ds ROWS
BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg_throughput

FROM daily_throughput

)

SELECT * FROM throughput_with_rolling_avg;

```

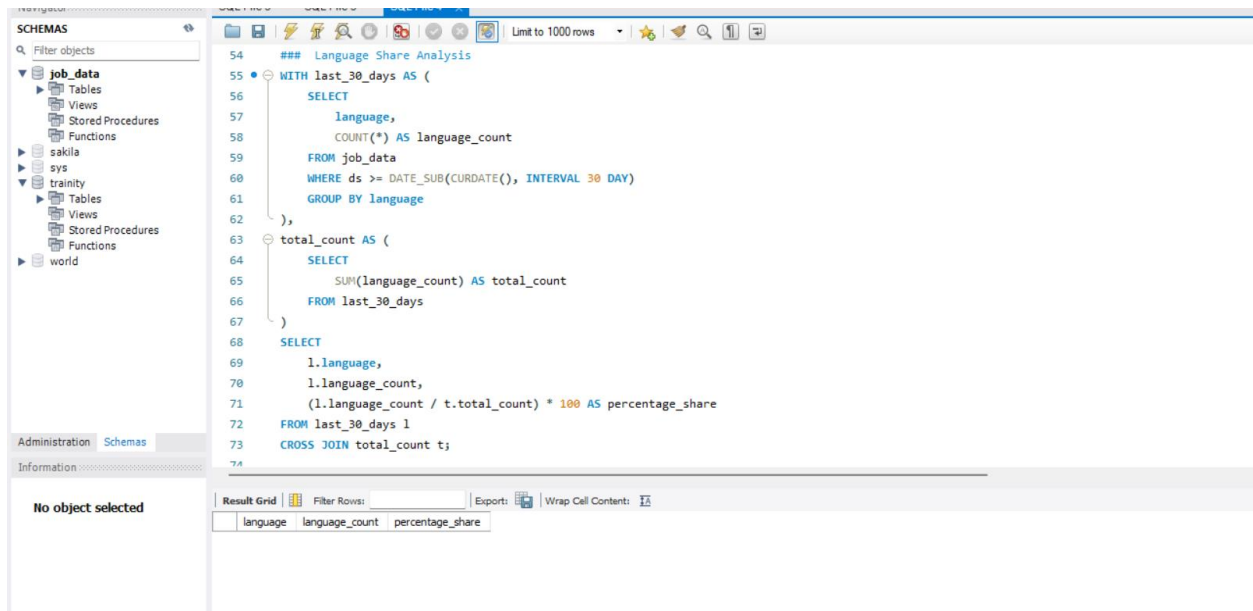


The screenshot shows a SQL IDE interface. The left sidebar displays a schema tree with databases like 'job_data', 'sakila', 'sys', 'trainity', and 'world'. The main query window contains a SQL script with two CTEs: 'daily_throughput' and 'throughput_with_rolling_avg'. The 'throughput_with_rolling_avg' CTE uses a window function with a 6-row rolling average. The bottom pane shows the 'Result Grid' with 8 rows of data.

ds	throughput_per_second	rolling_avg_throughput
2020-11-25	0.0222	0.02220000
2020-11-26	0.0179	0.02005000
2020-11-27	0.0096	0.01656667
2020-11-28	0.0606	0.02757500
2020-11-29	0.0500	0.03206000
2020-11-30	0.0500	0.03505000

3: Language Share Analysis

```
WITH last_30_days AS (  
    SELECT  
        language,  
        COUNT(*) AS language_count  
    FROM job_data  
    WHERE ds >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)  
    GROUP BY language  
)  
total_count AS (  
    SELECT  
        SUM(language_count) AS total_count  
    FROM last_30_days  
)  
SELECT  
    l.language,  
    l.language_count,  
    (l.language_count / t.total_count) * 100 AS percentage_share  
FROM last_30_days l  
CROSS JOIN total_count t;
```



4: Duplicate Rows Detection

SELECT

job_id,

actor_id,

event,

language,

time_spent,

org,

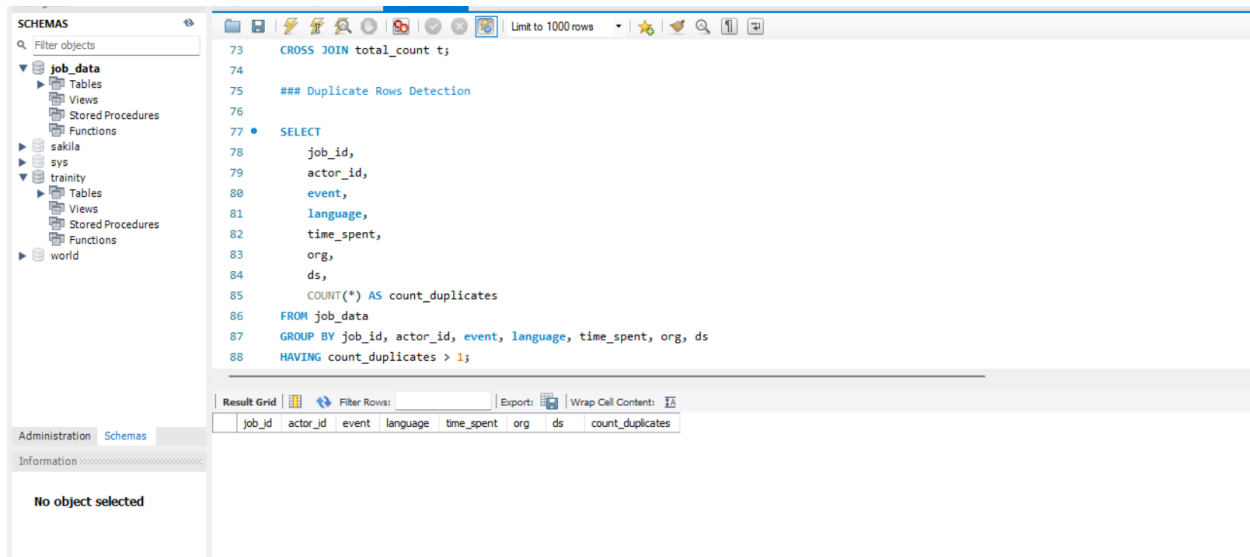
ds,

COUNT(*) AS count_duplicates

FROM job_data

GROUP BY job_id, actor_id, event, language, time_spent, org, ds

HAVING count_duplicates > 1;



Project Description:

This project involves operational analytics where we analyze the company's end-to-end operations using data. The objective is to derive insights from the dataset related to job events, helping different departments to understand and improve various metrics.

Approach:

Data Understanding: Initially, the dataset was loaded and inspected to understand its structure and content.

Data Preprocessing: The date column was converted to a standard format suitable for SQL operations.

Analysis Execution: SQL queries were crafted to address the project tasks, including temporal analysis, rolling average calculation, language distribution, and duplicate detection.

Tech-Stack Used

MySQL Workbench: Used for querying and analyzing the dataset.

Insights

Jobs Reviewed Over Time: This analysis provides a detailed breakdown of job reviews per hour, helping to identify peak review times.

Throughput Analysis: By calculating the 7-day rolling average, we get a smoothed view of throughput, which is useful for identifying trends without the noise of daily fluctuations.

Language Share: Understanding the distribution of job languages helps in allocating resources effectively.

Duplicate Rows: Detecting duplicates ensures data integrity and accuracy in analysis.

Result

The project provided valuable insights into job reviews, throughput performance, language distribution, and data integrity. These insights can help in making informed decisions to enhance operational efficiency and resource allocation.