Case Study 1: Job Data Analysis

Creating database and table

```
create database casestudy_1;
       CREATE TABLE job_data
2 •
3
    ⊖ (
4
           ds DATE,
5
           job_id INT NOT NULL,
           actor_id INT NOT NULL,
6
7
           event VARCHAR(15) NOT NULL,
8
           language VARCHAR(15) NOT NULL,
           time spent INT NOT NULL,
9
10
           org CHAR(2)
       );
11
12
13 •
       INSERT INTO job_data (ds, job_id, actor_id, event, language, time_spent, org)
       VALUES ('2020-11-30', 21, 1001, 'skip', 'English', 15, 'A'),
           ('2020-11-30', 22, 1006, 'transfer', 'Arabic', 25, 'B'),
15
16
           ('2020-11-29', 23, 1003, 'decision', 'Persian', 20, 'C'),
           ('2020-11-28', 23, 1005, 'transfer', 'Persian', 22, 'D'),
17
           ('2020-11-28', 25, 1002, 'decision', 'Hindi', 11, 'B'),
18
           ('2020-11-27', 11, 1007, 'decision', 'French', 104, 'D'),
19
           ('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),
20
           ('2020-11-25', 20, 1003, 'transfer', 'Italian', 45, 'C');
21
22
```

Jobs Reviewed Over Time:

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

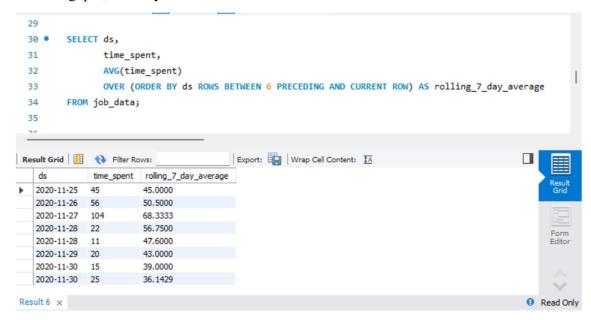
Your Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.



Throughput Analysis:

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

Your Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.



Reason:

Daily Metric: This provides the raw, day-to-day values of throughput without any smoothing or averaging. It might be useful when examining short-term trends or fluctuations. Daily metrics are sensitive to sudden changes and can provide insights into specific days' performances.

7-Day Rolling Average: This smoothed metric helps in identifying overall trends and patterns by averaging the data over a 7-day period, reducing the impact of daily fluctuations or irregularities. It's particularly useful for understanding long-term trends, identifying broader patterns, and eliminating noise in the data.

Language Share Analysis:

Objective: Calculate the percentage share of each language in the last 30 days.

Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.



Duplicate Rows Detection:

Objective: Identify duplicate rows in the data.

Your Task: Write an SQL query to display duplicate rows from the job_data table.

