

QUESTION 1

The table below contains information about tweets over a given period of time. Calculate the 3-day rolling average of tweets published by each user for each date that a tweet was posted. Output the user id, tweet date, and rolling averages rounded to 2 decimal places.

Important Assumptions:

1. Rows in this table are consecutive and ordered by date.
2. Each row represents a different day
3. A day that does not correspond to a row in this table is not counted. The most recent day is the next row above the current row.

Note: Rolling average is a metric that helps us analyse data points by creating a series of averages based on different subsets of a dataset. It is also known as a moving average, running average, moving mean, or rolling mean.

Tweets' Rolling Averages [Twitter SQL Interview Question]

Accepted

Congrats 🎉 - Share this problem, and your solution, on LinkedIn or Twitter!

[Share on Twitter](#) [Share on LinkedIn](#)

In your post, don't forget to tag Nick Singh, so that he can comment on and share your post with his audience of 90k+ followers on LinkedIn and 6k+ followers on Twitter (which will give your post and profile more visibility)!

Output

user_id	tweet_date	rolling_avg_3d
111	06/01/2022 04:00:00	2.00
111	06/02/2022 04:00:00	1.50
111	06/04/2022 04:00:00	1.33
111	06/15/2022 04:00:00	1.00
199	06/08/2022 04:00:00	2.00
199	06/22/2022 04:00:00	1.50
254	06/10/2022 04:00:00	2.00
254	06/11/2022 04:00:00	2.00

Expected

```
1 WITH temp AS (  
2   SELECT  
3     user_id,  
4     tweet_date,  
5     COUNT(DISTINCT tweet_id) AS tweet_num  
6   FROM  
7     tweets  
8   GROUP BY  
9     user_id, tweet_date  
10  )  
11  
12  SELECT  
13    user_id,  
14    tweet_date,  
15    ROUND(  
16      AVG(tweet_num) OVER(  
17        PARTITION BY user_id  
18        ORDER BY user_id, tweet_date  
19        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW), 2)  
20    AS rolling_avg_3d  
21  FROM  
22    temp;
```

PostgreSQL 14

[Run Code](#) [Submit](#)

Solution to Question 1

QUESTION 2

You are given the tables below containing information on Robinhood trades and users. Write a query to list the top three cities that have the most completed trade orders in descending order.

Output the city and number of orders.

The screenshot shows a web browser with the DataLemur website. The page title is "Cities With Completed Trades [Robinhood SQL Interview Question]". The "Submissions" tab is active, showing a successful submission. The SQL query is as follows:

```
1 SELECT
2   users.city,
3   COUNT(trades.order_id) AS total_orders
4
5 FROM trades
6 LEFT JOIN
7   users ON
8   trades.user_id = users.user_id
9
10 WHERE
11   trades.status = 'Completed'
12
13 GROUP BY
14   users.city
15
16 ORDER BY
17   total_orders DESC
18
19 LIMIT 3;
```

The "Output" section shows the following results:

city	total_orders
San Francisco	4
Boston	3
Denver	2

The "Expected" section shows the same results. The "Run Code" button is highlighted in red.

Solution to Question 2