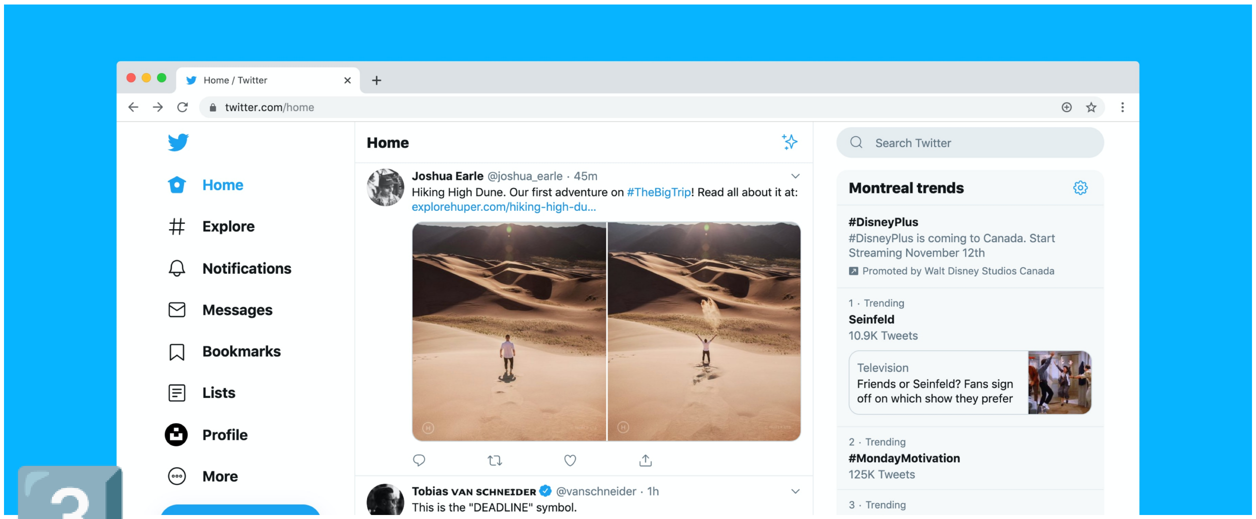

Joshua Earle @joshua_earle · 45m
Hiking High Dune. Our first adventure on #TheBigTrip! Read all about it at: explorehuper.com/hiking-high-du...

Tobias van Schneider @vanschneider · 1h
This is the "DEADLINE" symbol.

#DisneyPlus
#DisneyPlus is coming to Canada. Start Streaming November 12th
Promoted by Walt Disney Studios Canada

1 · Trending
Seinfeld
10.9K Tweets

Television
Friends or Seinfeld? Fans sign off on which show they prefer

2 · Trending
#MondayMotivation
125K Tweets

3 · Trending

Tweets' Rolling Averages (Twitter)

QUESTION

Given a table of tweet data over a specified time period, calculate the 3-day rolling average of tweets for each user. Output the user ID, tweet date, and rolling averages rounded to 2 decimal places.

Notes:

- A rolling average, also known as a moving average or running mean is a time-series technique that examines trends in data over a specified period of time.
- In this case, we want to determine how the tweet count for each user changes over a 3-day period.

Effective April 7th, 2023, the problem statement, solution and hints for this question have been revised.

`tweets` Table:

Column Name	Type
user_id	integer
tweet_date	timestamp
tweet_count	integer

`tweets` Example Input:

user_id	tweet_date	tweet_count
111	06/01/2022 00:00:00	2
111	06/02/2022	1

	00:00:00	
111	06/03/2022 00:00:00	3
111	06/04/2022 00:00:00	4
111	06/05/2022 00:00:00	5

Step 1: Identify the problem of the case

In this example, the output must include the user ID, tweet date, and alternative average, all rounded to two decimal places. Moving Average, also known as moving average or running average, is a time series approach that analyses data patterns over a specific time period. In this scenario, we want to see how the amount of tweets per user varies over the course of three days.

Output form:

Column Name	Type
user_id	integer
tweet_date	timestamp
rolling_avg_3d	medical

Step 2 : Analyze and solve problems

The following query estimates the average tweet count for each user ID and date to return the average rolling average tweet count for each user. We alter the preceding query by adding the `ROWS BETWEEN 2` preceding AND `CURRENT ROW` expression to the window function to determine the rolling average tweet count by 3-day period.

```
SELECT *
      , AVG(tweet_count) OVER(
        PARTITION BY user_id ORDER BY tweet_date
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
        )
FROM tweets
```

Finally, we use the `ROUND()` method in the preceding query to round up the rolling averages to the nearest two decimal points.

```
SELECT user_id, tweet_date
      , ROUND(AVG(tweet_count) OVER(
        PARTITION BY user_id ORDER BY tweet_date
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
        ),2) rolling_avg_3d
FROM tweets
```

OUTPUT

user_id	tweet_date	rolling_avg_3d
111	06/01/2022 00:00:00	2.00
111	06/02/2022 00:00:00	1.50
111	06/03/2022 00:00:00	2.00
111	06/04/2022 00:00:00	2.67
111	06/05/2022 00:00:00	4.00
111	06/06/2022 00:00:00	4.33
111	06/07/2022 00:00:00	5.00
199	06/01/2022 00:00:00	7.00
199	06/02/2022 00:00:00	6.00
199	06/03/2022 00:00:00	7.00
199	06/04/2022 00:00:00	5.00
199	06/05/2022 00:00:00	6.00
199	06/06/2022 00:00:00	3.67
199	06/07/2022 00:00:00	4.00
254	06/01/2022 00:00:00	1.00
254	06/02/2022 00:00:00	1.00
254	06/03/2022 00:00:00	1.33
254	06/04/2022 00:00:00	1.33
254	06/05/2022 00:00:00	2.00
254	06/06/2022 00:00:00	1.67
254	06/07/2022 00:00:00	2.33