

QUESTION

User Consecutive Day Streak Analysis Hard

10 Points

Given a table with event logs, find the top five users with the longest continuous streak of visiting the platform in 2020.

Note: A continuous streak counts if the user visits the platform at least once per day on consecutive days.

Output Schema:

Column	Type
user_id	INT
streak_length	INT

TABLE SCHEMA

```
1 CREATE TABLE events (  
2   user_id INT,  
3   created_at DATETIME,  
4   url VARCHAR(255)  
5 );  
6  
7 --Generating comprehensive sample data with interrupted streaks  
8 INSERT INTO events (user_id, created_at, url) VALUES  
9  
10 (1, '2019-12-30 10:00:00', 'https://example.com/2019-page1'),  
11 (1, '2019-12-31 11:00:00', 'https://example.com/2019-page2'),  
12 (2, '2019-11-15 12:00:00', 'https://example.com/2019-profile1'),  
13 (2, '2019-11-16 13:00:00', 'https://example.com/2019-profile2'),  
14 (3, '2019-10-20 14:00:00', 'https://example.com/2019-blog1'),  
15 (4, '2019-09-25 16:00:00', 'https://example.com/2019-review1'),  
16 (4, '2019-09-26 17:00:00', 'https://example.com/2019-review2'),  
17 (5, '2019-08-30 18:00:00', 'https://example.com/2019-summer1'),  
18 (5, '2019-08-31 19:00:00', 'https://example.com/2019-summer2'),  
19 (6, '2019-07-15 20:00:00', 'https://example.com/2019-page1'),  
20 (6, '2019-07-16 21:00:00', 'https://example.com/2019-page2'),  
21 (1, '2020-01-01 10:00:00', 'https://example.com/page1'),  
22 (1, '2020-01-02 11:00:00', 'https://example.com/page2'),  
23 (1, '2020-01-04 12:00:00', 'https://example.com/page3'),  
24 (1, '2020-01-05 13:00:00', 'https://example.com/page4'),  
25 (1, '2020-01-06 14:00:00', 'https://example.com/page5'),  
26 (1, '2020-01-07 12:00:00', 'https://example.com/page7'),  
27 (1, '2020-01-08 12:00:00', 'https://example.com/page8'),  
28 (1, '2020-01-09 12:00:00', 'https://example.com/page9'),  
29 (1, '2020-01-10 12:00:00', 'https://example.com/page10'),  
30 (2, '2020-02-10 15:00:00', 'https://example.com/dashboard'),  
31 (2, '2020-02-11 16:00:00', 'https://example.com/profile'),  
32 (2, '2020-02-12 17:00:00', 'https://example.com/settings'),  
33 (2, '2020-02-14 18:00:00', 'https://example.com/messages'),  
34 (2, '2020-02-15 19:00:00', 'https://example.com/notifications'),  
35 (2, '2020-02-16 20:00:00', 'https://example.com/search'),  
36
```

SOLUTION

```
Day-12 Saisri

WITH event_rankings AS (
  SELECT
    user_id, DATE(created_at) AS event_date,
    ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY DATE(created_at)) AS
user_ranking
  FROM events
  WHERE strftime('%Y', created_at) = '2020'
  GROUP BY user_id, DATE(created_at)
),
grouped_dates AS (
  SELECT
    user_id, event_date, user_ranking,
    DATE(event_date, '-' || user_ranking || ' days') AS streak_group
  FROM event_rankings
),
streaks AS (
  SELECT user_id, streak_group, COUNT(*) AS streak_length
  FROM grouped_dates
  GROUP BY user_id, streak_group
),
max_streaks AS (
  SELECT user_id, MAX(streak_length) AS streak_length
  FROM streaks
  GROUP BY user_id
)
SELECT
  user_id, streak_length
FROM max_streaks
ORDER BY streak_length DESC
LIMIT 5;
```

OUTPUT

▼ Tables

user_id	streak_length
6	10
1	7
2	5
7	4
9	3

My Thought Process:

I started by filtering the data for 2020 and removed any duplicate visits on the same day. Then, I used ROW_NUMBER() to assign each visit a position in order. The trick I used was subtracting the row number from the visit date this groups all consecutive dates together. From there, I counted how many days were in each streak and picked the longest one per user. Finally, I selected the top 5 longest streaks.

