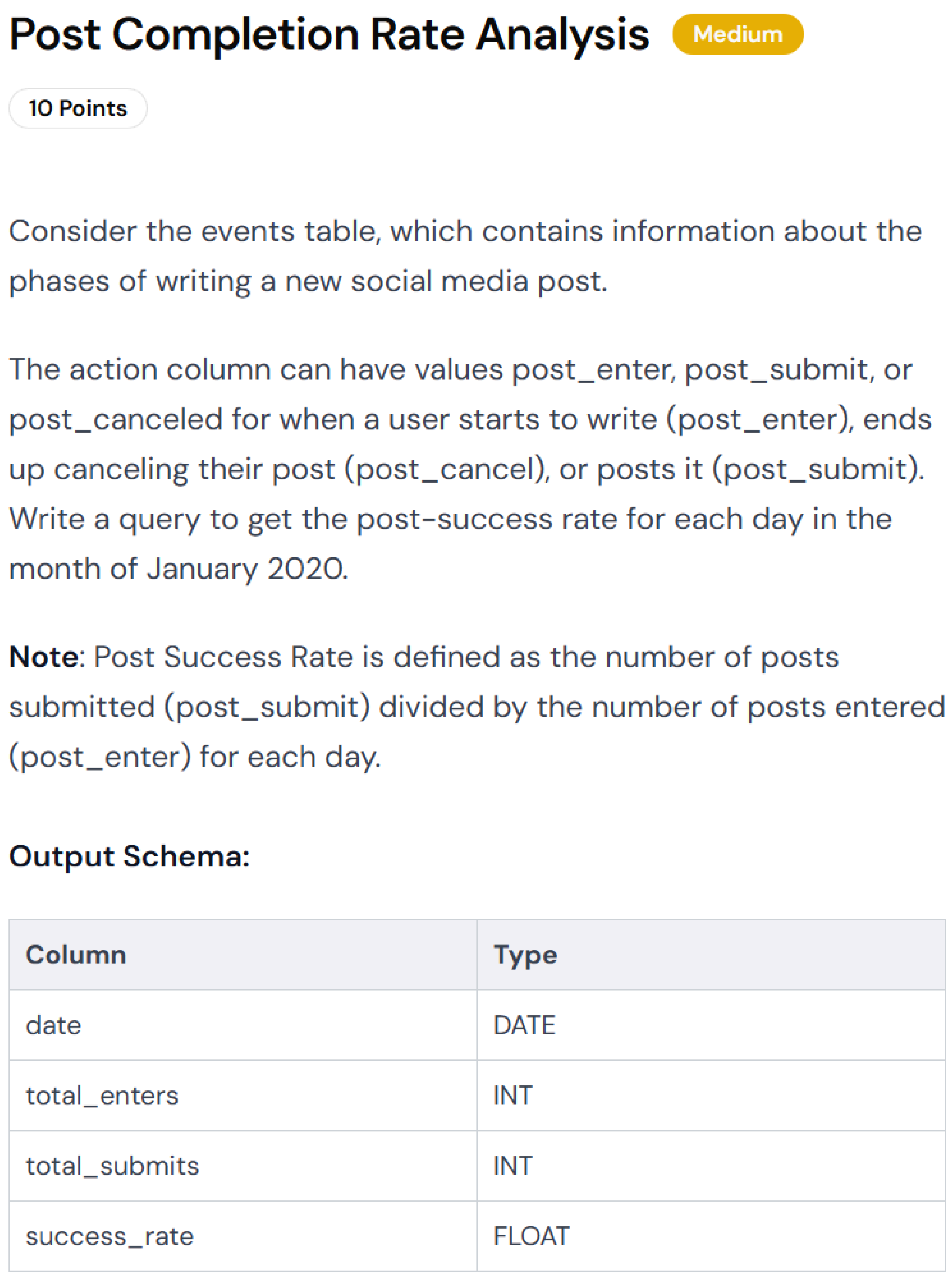
QUESTION



Post Completion Rate Analysis

Consider the events table, which contains information about the phases of writing a new social media post.

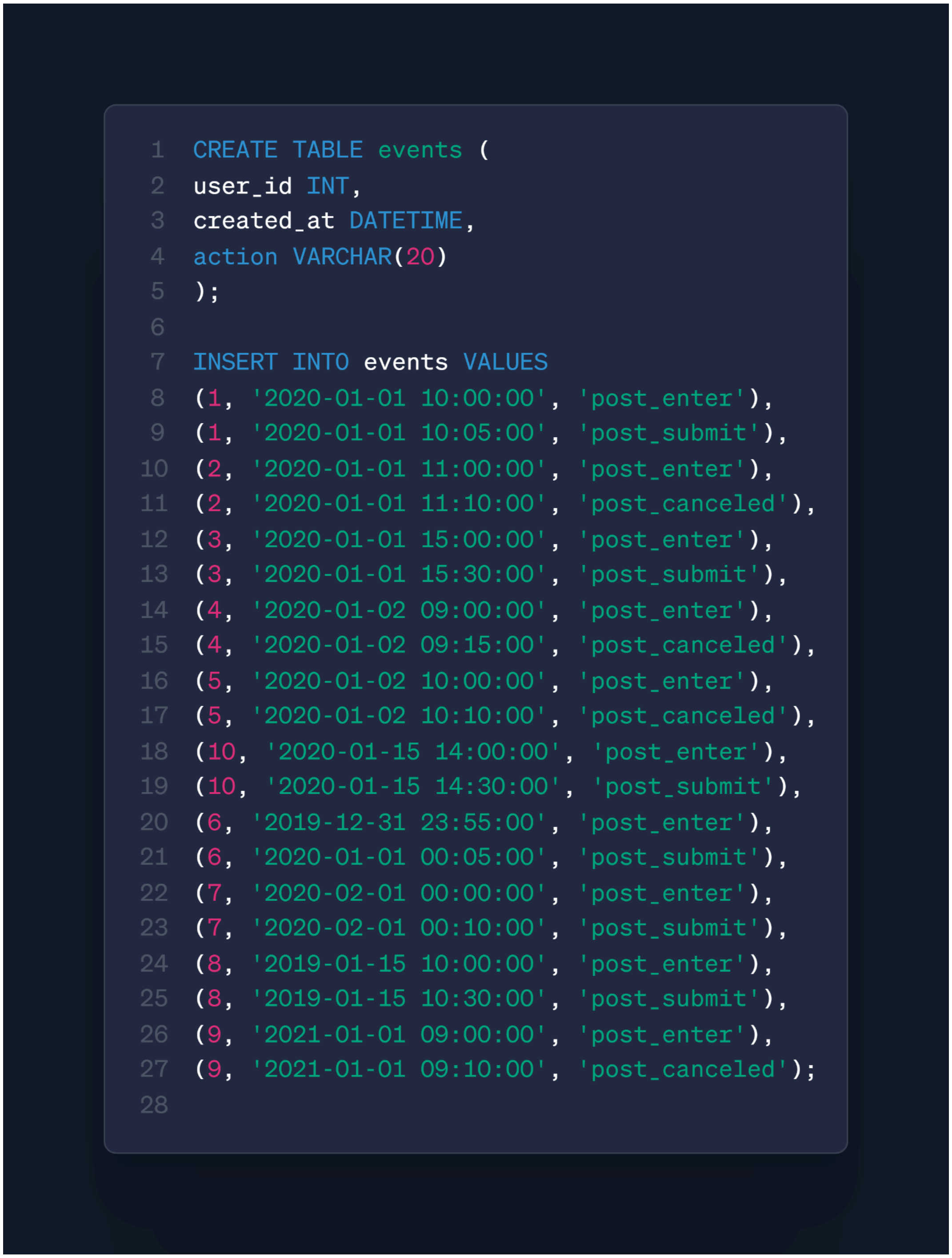
The action column can have values post_enter, post_submit, or post_canceled for when a user starts to write (post_enter), ends up canceling their post (post_cancel), or posts it (post_submit). Write a query to get the post-success rate for each day in the month of January 2020.

Note: Post Success Rate is defined as the number of posts submitted (post_submit) divided by the number of posts entered (post_enter) for each day.

Output Schema:

Column	Type
date	DATE
total_enters	INT
total_submits	INT
success_rate	FLOAT

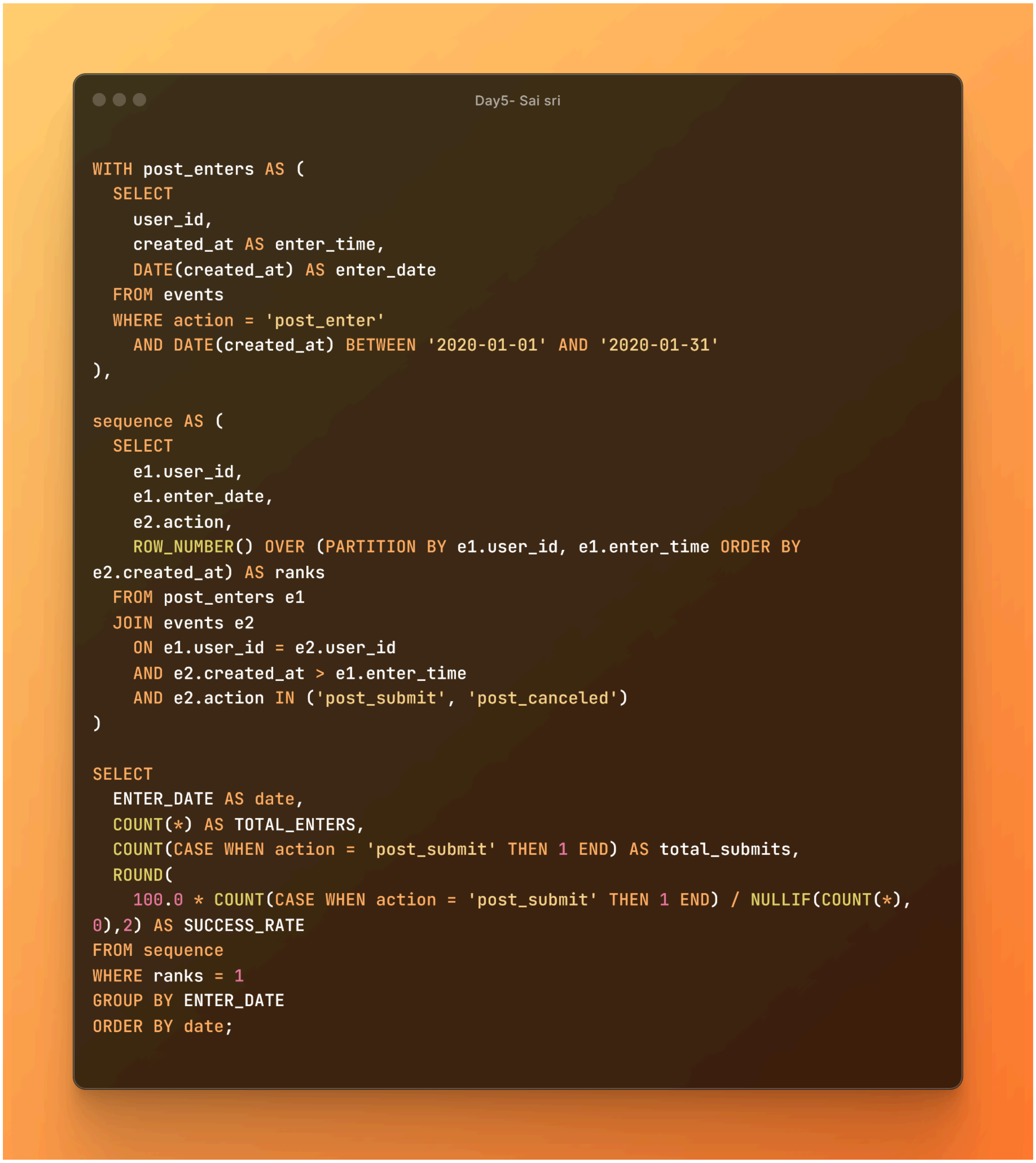
TABLE SCHEMA



```sql
CREATE TABLE events (
user_id INT,
created_at DATETIME,
action VARCHAR(20)
);

INSERT INTO events VALUES
(1, '2020-01-01 10:00:00', 'post_enter'),
(1, '2020-01-01 10:05:00', 'post_submit'),
(2, '2020-01-01 11:00:00', 'post_enter'),
(2, '2020-01-01 11:10:00', 'post_canceled'),
(3, '2020-01-01 15:00:00', 'post_enter'),
(3, '2020-01-01 15:30:00', 'post_submit'),
(4, '2020-01-02 09:00:00', 'post_enter'),
(4, '2020-01-02 09:15:00', 'post_canceled'),
(5, '2020-01-02 10:00:00', 'post_enter'),
(5, '2020-01-02 10:10:00', 'post_canceled'),
(10, '2020-01-15 14:00:00', 'post_enter'),
(10, '2020-01-15 14:30:00', 'post_submit'),
(6, '2019-12-31 23:55:00', 'post_enter'),
(6, '2020-01-01 00:05:00', 'post_submit'),
(7, '2020-02-01 00:00:00', 'post_enter'),
(7, '2020-02-01 00:10:00', 'post_submit'),
(8, '2019-01-15 10:00:00', 'post_enter'),
(8, '2019-01-15 10:30:00', 'post_submit'),
(9, '2021-01-01 09:00:00', 'post_enter'),
(9, '2021-01-01 09:10:00', 'post_canceled');
```

SOLUTION

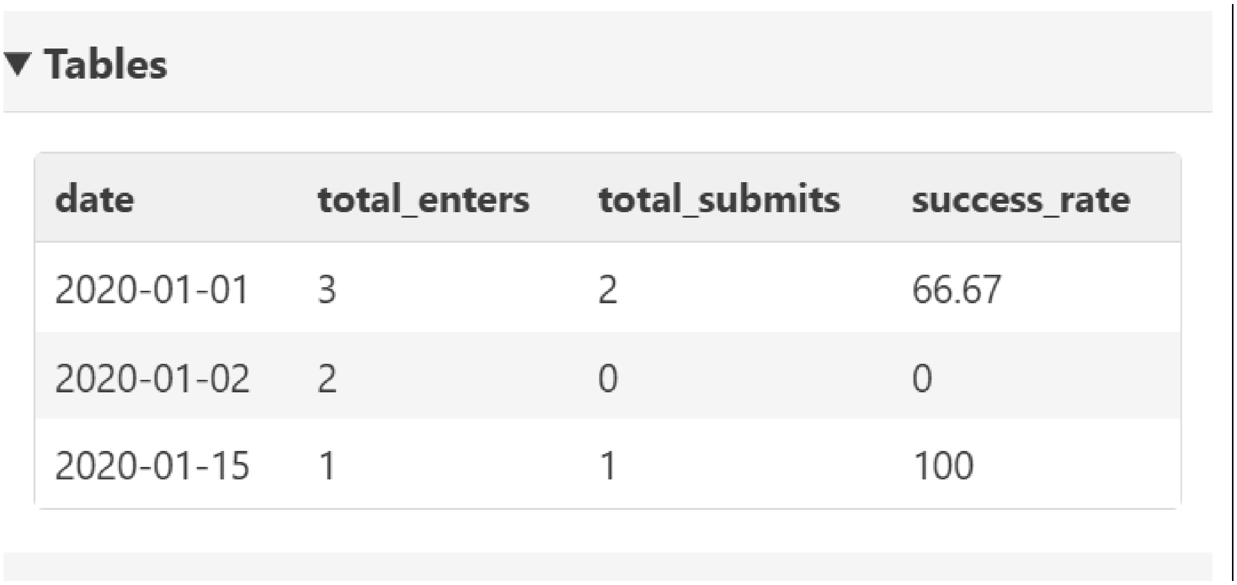


```sql
WITH post_enters AS (
  SELECT
    user_id,
    created_at AS enter_time,
    DATE(created_at) AS enter_date
  FROM events
  WHERE action = 'post_enter'
    AND DATE(created_at) BETWEEN '2020-01-01' AND '2020-01-31'
),

sequence AS (
  SELECT
    e1.user_id,
    e1.enter_date,
    e2.action,
    ROW_NUMBER() OVER (PARTITION BY e1.user_id, e1.enter_time ORDER BY
e2.created_at) AS ranks
  FROM post_enters e1
  JOIN events e2
    ON e1.user_id = e2.user_id
    AND e2.created_at > e1.enter_time
    AND e2.action IN ('post_submit', 'post_canceled')
)

SELECT
  ENTER_DATE AS date,
  COUNT(*) AS TOTAL_ENTERS,
  COUNT(CASE WHEN action = 'post_submit' THEN 1 END) AS total_submits,
  ROUND(
    100.0 * COUNT(CASE WHEN action = 'post_submit' THEN 1 END) / NULLIF(COUNT(*),
0),2) AS SUCCESS_RATE
FROM sequence
WHERE ranks = 1
GROUP BY ENTER_DATE
ORDER BY date;
```

OUTPUT



▼ Tables

date	total_enters	total_submits	success_rate
2020-01-01	3	2	66.67
2020-01-02	2	0	0
2020-01-15	1	1	100

My Thought Process:

I first filtered all post_enter events in January. Then, for each entry, I looked for the very next action (post_submit or post_canceled) by the same user using ROW_NUMBER(). This helped me pair each entry with its correct outcome.

After that, I grouped everything by the original entry date and calculated the total number of posts started, the number submitted, and the overall success rate.

Business Impact:

This type of analysis can be used to track user flow in any multi-step process like form completions, job applications, or checkouts. It gives real insights into user behavior, helps identify where drop-offs happen, and can drive better UX and design decisions.