7️⃣

Unsubscribe Rates (Twitter)

Question:

Twitter wants to roll out more **push notifications** to users because they think users are missing out on good content. Twitter decides to do this in an A/B test.

Say that after more notifications are released, there is a sudden increase in the total number of unsubscribes.

We're given two tables: events where actions are 'login', 'nologin', and 'unsubscribe' and another table called variants where user's are bucketed into a control and a variant A/B test.

Given these tables, write a query to display a graph to understand how unsubscribes are affecting login rates over time.

Note: Let's say that all users are automatically put into the A/B test.

`events` table

column	Type
user_id	INTEGER
created_at	DATETIME
action	STRING

`variants` table

Column	Type
user_id	INTEGER
experiment	STRING
variant	STRING

Solving:

Step 1: Start Each SQL Case Study by Making Assumptions

This question asks us to compare multiple variables at play here. Specifically, we're looking at:

- There is a new notification system.
- We're interested in the effect the new notifications are having on unsubscribes.

We're not sure how unsubscribes are affecting login rates, but **we can plot a graph** that would help us **visualize how the login rates change before and after an unsubscribe from a user**.

We can also **see how the login rates compare for unsubscribes** for each bucket of the A/B test. Given that we want to measure two different changes, we have to eventually do a GROUP BY of two different variables:

1. Date
2. Bucket variant

Step 2: Develop a Hypothesis for the SQL Case Question

In order to visualize this, we'll need to plot two lines on a 2D graph.

- T**he x-axis** represents days until unsubscribing with a range of -30 to 0 to 30, in which -30 is thirty days before unsubscribing and 30 is 30 days after unsubscribing.
- **The y-axis** represents the average login rate for each day. We'll be plotting two lines for each of the A/B test variants, control and test.

Now that we have what we're going to graph, it's a matter of writing a SQL query to get the dataset for the graph.

We can make sure our dataset looks something like this:

variant	days_since_unsub	login_rate
control	-30	90%
test	-30	91%

Each column represents a different axis or line for our graph.

Step 3: SQL Coding + Analysis

We know that we have to get every user that has unsubscribed, so we'll first INNER JOIN the abtest table to the events table, where there exists an unsubscribe event. Now we've isolated all users that have ever unsubscribed.

Additionally, we have to then get every event in which the user has logged in, and divide it by the total number of users that are eligible within the timeframe.

```
SELECT
    variant
    , DATEDIFF(e1.created_at, e2.created_at) AS days_since_unsub
    , ROUND(COUNT(DISTINCT CASE WHEN e2.action = 'login'
THEN 1 ELSE 0 END)/COUNT(DISTINCT abtest.user_id) * 100, 2)AS login_rate
FROM variants as abtest
INNER JOIN events AS e1
    ON abtest.user_id = e1.user_id
        AND e1.action = 'unsubscribe'
INNER JOIN events AS e2
    ON abtest.user_id = e2.user_id
        AND e2.action IN ('login', 'nologin')
        AND DATEDIFF(e1.created_at, e2.created_at)
            BETWEEN -30 AND 30
GROUP BY 1,2
```