SQL Practice - Day-32 - 20250311

Problem Statement:

Ride Duration Analysis
A bike-sharing company wants to analyze ride durations to understand user behavior better. They are interested in identifying "long rides"—rides that lasted significantly longer than usual for a user.

📂 Table Schema:
RIDES
RIDE_ID (Unique identifier for each ride)
USER_ID (ID of the user)
START_TIME (Timestamp when the ride started)
END_TIME (Timestamp when the ride ended)

🎯 Your Task:
Write a query to find users who had at least two rides in the past 30 days where the duration was more than 1.5 times their average ride duration in this period.

📝 Output:
USER_ID
LONG_RIDE_COUNT (Number of rides exceeding 1.5x their average duration)
AVG_DURATION (User's average ride duration in minutes)

⚡ Conditions:
Consider only rides from the last 30 days.
Ride duration is calculated as (END_TIME - START_TIME).
Only include users who had at least two long rides in this period.
Sort the results by LONG_RIDE_COUNT in descending order, then by USER_ID.

** QUERY:

```
WITH BASE AS(
SELECT *,
TIMEDIFF(END_TIME, START_TIME) AS DURATION
FROM RIDES
WHERE START_TIME >= CURRENT_DATE() - INTERVAL 30 DAY
)
,AVERAGE_TIME AS(
SELECT USER_ID,
AVG(DURATION) AS AVG_TIME
FROM BASE
GROUP BY 1
)
, BASE2 AS(
SELECT A.*, COUNT(*) AS ABV_AVG_RIDES
FROM BASE AS A
LEFT JOIN AVERAGE_TIME AS B
ON A.USER_ID = B.USER_ID
WHERE A.DURATION > 1.5 * AVG_TIME
GROUP BY A.USER_ID
HAVING COUNT(*) > 1
)
SELECT USER_ID, AVG_TIME AS AVG_DURATION, SUM(*) AS LONG_RIDE_COUNT
FROM BASE2
GROUP BY 1,2
;
```