****

**Nested CASE Statement**

In this exercise, you'll modify a SQL query using a nested CASE statement to create a hierarchical user classification. Specifically, we'll create a "Special Influencer" category for users with at least 400 followers and retain the "Influencer" category for users with more than 300 followers.

**Step-by-Step Query:**

SELECT

User\_ID,

Account\_Name,

User\_Name,

Followers,

Posts,

CASE

-- Step 1: Check if the user is an Influencer (has more than 300 followers)

WHEN Followers > 300 THEN

-- Step 2: Inside the Influencer condition, check if they are a Special Influencer

CASE

WHEN Followers >= 400 THEN 'Special Influencer' -- Updated condition for Special Influencer

ELSE 'Influencer'

END

-- Step 3: If the user is not an Influencer, check if they are an Active User based on posts

WHEN Posts > 100 THEN 'Active User'

-- Step 4: Otherwise, categorise them as a Regular User

ELSE 'Regular User'

END AS User\_Category

FROM user\_data;

**Explanation:**

* Step 1: The outer CASE checks if the user has more than 300 followers.
* Step 2: If true, the nested CASE checks if the followers are at least 400 to categorise the user as a "Special Influencer."
* If the follower count is between 301 and 399, the user remains an "Influencer."
* The outer CASE then checks if the user is an "Active User" based on their post count, or a "Regular User" otherwise.

**Reflection:**

After executing this query, ask yourself the following questions:

* Why do we need to place the "Special Influencer" condition before the "Influencer" condition?
* What would happen if we placed the "Influencer" condition first?
* How does nesting CASE statements allow us to handle more complex classifications?
* Is there a more streamlined approach for our task?