📘 Intermediate SQL – Lesson 3: Conditional Logic with CASE Statements

🎯 What You'll Learn

In this lesson, you'll learn how to:

- Use `CASE` statements to return different outputs based on conditions
- Create labeled or categorized columns
- Apply conditional logic to perform calculations (like bonuses or raises)
- Build clean, readable outputs with customized results

🧠 What is a CASE Statement?

A **CASE** statement works like an if-then-else structure in SQL.
It allows you to check conditions and return a value when those conditions are met.

For example:

- Label employees as "Young" or "Old" based on age
- Apply salary increases based on job titles
- Customize outputs without altering your original data

🧾 Basic Usage

The structure follows:

- **WHEN** a condition is true
- **THEN** return a specific result
- Use **ELSE** for all other outcomes
- Close the logic with **END**

Only the **first true condition** will be applied — so order matters!

📂 Use Case 1: Categorizing Employees by Age

Let's say you want to group employees as:

- "Old" if their age is over 30
- "Young" if under or equal to 30
- "Baby" if they're very young (e.g., age ≤ 27)

This approach helps with labeling, segmentation, or reporting in dashboards and summaries.

📈 Use Case 2: Giving Out Raises Based on Job Title

Imagine your company had a successful year and you want to:

- Give **Salesmen** a 10% raise
- Give **Accountants** a 5% raise
- Give **HR** staff a 1% raise
- Give everyone else a 3% raise

Using a `CASE` statement, you can **apply this logic in your query** without touching the actual data — and calculate each employee's new salary instantly.

It’s a great way to add business rules directly into your SQL output.

💡 Pro Tips

- You can return **text labels** (like "Old", "Young", etc.)
- You can return **numeric calculations**, like updated salaries
- You can use as many `WHEN`/`THEN` conditions as needed
- Use `AS` to give your CASE output a **custom column name** (e.g., "Salary After Raise")
- The order of your `WHEN` clauses **matters** — the first match wins!

📌 Recap

✅ `CASE` statements let you apply logic and customization inside your queries
✅ They're perfect for labeling, categorizing, or conditional calculations
✅ You can use them with text or numbers
✅ Great for real-world applications like salary adjustments, bonus rules, or age-based groups

---

🚀 Coming Up Next: Data Updates with UPDATE and DELETE

In the next video, you’ll learn how to **modify data** in your tables using `UPDATE` and how to safely remove it using `DELETE`.