SQL CASE Statement – Understanding Notes

---

1. Introduction to CASE Statement
The CASE statement allows conditional logic (if-then-else) inside SQL queries. It returns different results based on specified conditions.

2. Types of CASE Statements
Simple CASE:
CASE expression WHEN value1 THEN result1 WHEN value2 THEN result2 ELSE resultN END

Searched CASE:
CASE WHEN condition1 THEN result1 WHEN condition2 THEN result2 ELSE resultN END

3. Example 1 – Department Classification
SELECT name, CASE department WHEN 'IT' THEN 'IT Team' ELSE 'Other' END AS department_group FROM employees;

4. Example 2 – Age Group Classification
SELECT employee_id, name, age, department, CASE WHEN age < 25 THEN 'Junior' WHEN age BETWEEN 25 AND 30 THEN 'Young' ELSE 'Senior' END AS age_group FROM employees;

5. Example 3 – Nested CASE
SELECT name, CASE WHEN age < 30 THEN CASE WHEN department = 'Sales' THEN 'Jr Sales' ELSE 'Junior' END ELSE 'Senior' END AS employee_name FROM employees;

6. Example 4 – Multiple Nested Conditions
SELECT name, CASE WHEN age < 30 THEN CASE WHEN department = 'Sales' THEN 'Jr Sales' ELSE 'Junior' END WHEN age BETWEEN 30 AND 38 THEN CASE WHEN department = 'Sales' THEN 'Mid Sales' ELSE 'Middle' END ELSE 'Senior' END AS employee_name FROM employees;

7. Example 5 – Handling NULL Values
SELECT name, CASE WHEN department IS NULL THEN 'No Department Assigned' ELSE department END AS department_status FROM employees;

Conclusion
The CASE statement enhances SQL queries by enabling conditional logic directly within SELECT statements. It is essential for data classification, grouping, reporting, and handling special conditions like NULL values.