

# Data Analytics

-



# CASE WHEN



# 1. Usos avanzados CASE WHEN

## CASE

Retorna un valor específico basado en condiciones booleanas. Si se da la condición es True, entonces realizará la instrucción.

Se ubica, generalmente, en el SELECT.

Es la lógica que ya conocemos **if-else**.

Esta instrucción es como clasificar la información según las condiciones especificadas en cada WHEN y añadirá una columna a la salida de la query con el nombre que se le haya indicado en el END AS.

Esto no modifica la base de datos, únicamente se verá reflejado en la salida de la query.

```
SELECT columna1, columna2
CASE
  WHEN condición THEN haz_esto
  WHEN condición THEN haz_esto_otro
  ELSE haz_esto_último
END AS clasificación
FROM tabla;
```

Quiero ver los alumnos, con su nota y el tipo de calificación que supone esa nota (excelente, notable, suficiente, suspenso).

```
SELECT nombre, nota
CASE
  WHEN nota BETWEEN 10 AND 9 THEN 'excelente'
  WHEN nota BETWEEN 8.99 AND 7 THEN 'notable'
  WHEN nota BETWEEN 6.99 AND 5 THEN 'suficiente'
  ELSE 'suspenso'
END AS calificación
FROM alumnos;
```



## 2. Usos en ORDER BY

Personalizar el orden de los resultados usando condiciones lógicas.

En este ejemplo, utilizamos CASE WHEN dentro de una cláusula ORDER BY para ordenar personas primero por aquellos que son mayores de 65 años, luego por menores de 18, y finalmente todos los demás. Es una forma muy poderosa de controlar cómo queremos que se presenten los datos en nuestros informes o aplicaciones.

```
SELECT nombre, edad
FROM habitantes
ORDER BY
CASE
    WHEN edad >= 65 THEN 1
    WHEN edad < 18 THEN 2
    ELSE 3
END AS bloques;
```



# 3. Usos en GROUP BY

Para agrupar los resultados basados en categorías definidas condicionalmente.

Este ejemplo agrupa personas en categorías de edad antes de contarlas, utilizando **CASE** dentro de **GROUP BY**.

```
SELECT  
  
CASE  
  
    WHEN edad < 20 THEN 'joven'  
  
    WHEN edad BETWEEN 20 AND 64 THEN 'adulto'  
  
    ELSE 'senior'  
  
END AS grupo_edad  
  
COUNT (*)  
  
FROM alumnos  
  
GROUP BY grupo_edad;
```



## 4. Usos en HAVING

Para filtrar grupos de datos basados en una condición evaluada dinámicamente.

Este ejemplo muestra cómo se pueden filtrar los departamentos que tienen un salario promedio alto, utilizando **CASE** en **HAVING**.

```
SELECT
    departamento_id
    , AVG(sueldo) AS sueldo_medio
FROM empleados
GROUP BY departamento_id
HAVING
    CASE
        WHEN AVG(sueldo) > 10000 THEN 'Alto'
        ELSE 'Bajo'
    END = 'Alto';
```



# 5. Usos en funciones y operaciones

Para modificar valores dentro de funciones o durante cálculos complejos.

Este ejemplo ajusta el salario de los empleados basado en su salario actual.

```
SELECT
    empleado_id
    , sueldo
    , sueldo + (CASE
        WHEN sueldo < 10000 THEN 500
        ELSE 200
        END) AS ajuste
FROM empleados ;
```



## 6. Usos en UPDATE

Para actualizar valores en una tabla basado en condiciones específicas.

Este ejemplo ajusta los precios de los productos basándose en su categoría.

```
UPDATE productos
SET precio = CASE
    WHEN categoria = 'Electrónica' THEN price * 0.9
    WHEN categoria = 'Ropa' THEN price * 1.1
    ELSE price
END
WHERE producto_id IN (1, 2, 3);
```

