

## OPERADORES ESPECIALES DE FILTRO

**UNION:** une en vertical los resultados de dos o más consultas eliminando los datos duplicados. Ambas consultas deben tener:

- Mismo número de columnas
- Mismo tipo de datos
- La primera consulta decide el orden de los resultados

SELECT columnas, “tipo dato1” AS “columnaoutput” se le puede añadir una columna en el output para diferenciar una tabla de otra en la unión

FROM tabla1

UNION

SELECT columnas, “tipo dato2”

FROM tabla2;

**UNION ALL:** igual que UNION pero sin eliminar datos duplicados.

Es útil cuando queremos ver las tablas completas o realizar varias consultas.

SELECT columnas

FROM tabla1

UNION ALL

SELECT columnas

FROM tabla2

ORDER BY ordena alfabéticamente todos los datos por la columna indicada, solo se pueden poner columnas de la primera consulta

LIMIT; limita el número de resultados

SELECT CONCAT(columna1, “ “, columna2) AS alias para unir dos datos

**IN, NOT IN:** seleccionar registros cuyo valor coincida, o no, con los valores dados.

SELECT columnas

FROM tabla

WHERE columna IN (valor1, valor2, valor3) filtra solo en la columna que se indica

UNION ALL

SELECT columnas

FROM tabla2

**LIKE, NOT LIKE:** parecido a IN pero no busca la palabra completa, solo **busca un patrón**. Comodines:

- % : cualquier cantidad de caracteres. 'A%' empieza por A sin importar el número de caracteres
- \_: representa cada carácter. 'B\_' empieza por B y tiene dos caracteres más
- \_%: al menos un carácter. 'E\_%' empieza por E y tiene al menos dos caracteres
- % %: contiene la letra en cualquier parte de la palabra '%m%'
- \: si la palabra contiene algún carácter especial lleva delante del carácter la barra \. '%\%%' busca textos que lleven un símbolo %. Caracteres especiales:
  - \0: busca valores nulos
  - ': para buscar comilla simple duplicarla → LIKE '%%"'
  - \"": busca comillas dobles
  - \n: busca saltos de línea en cadenas de texto
  - \\: para buscar barra invertida duplicarla → LIKE '%\\\\\\%'
  - \%:
  - \\_:

`SELECT *`

`FROM empleadas`

`WHERE nombre NOT LIKE '%p%';`

Selecciona todas las columnas de la tabla en las que los nombres no contengan p

`SELECT *`

`FROM empleadas`

`WHERE CHAR_LENGTH(nombre) > 4 AND nombre LIKE '%p%';`

Selecciona todas las columnas de la tabla en las que los nombres tengan una longitud mayor que 4 y contengan p

**REGEXP:** permite **definir reglas** específicas **para encontrar caracteres o combinaciones**.

`SELECT columna1, columna2, ...`

`FROM tabla`

`WHERE columna REGEXP 'patron';`

## Operadores:

- `.`: Representa cualquier carácter individual en la cadena.
- `*`: Coincide con cero o más repeticiones del carácter o patrón anterior.
- `+`: Coincide con una o más repeticiones del carácter o patrón anterior.
- `?`: Coincide con cero o una repetición del carácter o patrón anterior.
- `|`: Funciona como un operador "o" para alternativas.
- `[]`: Permite definir un conjunto de caracteres válidos para una posición específica en la cadena.
- `[^]`: Excluye un conjunto de caracteres para una posición específica en la cadena.
- `()`: Agrupa elementos para aplicar operadores a un conjunto completo.
- `{n}`: Coincide exactamente con n repeticiones del carácter o patrón anterior.
- `{n, m}`: Coincide con un rango de repeticiones, desde n hasta m, del carácter o patrón anterior.

[Chuleta REGEX](#)