

[Página Principal](#) / [Mis cursos](#) / [Base de Datos](#) / [UNIDAD 3](#) / [Lenguaje SQL](#)

Lenguaje SQL

Sentencias DML: Cláusula IN y Cláusula LIKE

Cláusula IN:

Ahora veremos, cómo obtenemos los códigos de ciertos productos específicos, por ejemplo Harina, Azúcar y Leche:



SQL

```
SELECT codigo FROM producto
WHERE descripción = 'Harina' OR
      descripción = 'Azúcar' OR
      descripción = 'Leche'
```

Pero si tenemos una lista larga de posibilidades, escribir todas estas cláusulas **OR** encadenadas sería muy tedioso, entonces usamos la sentencia **IN** que funciona de manera equivalente:



SQL

```
SELECT codigo FROM producto
WHERE descripción IN ('Harina' , 'Azúcar' , 'Leche')
```



Siempre después de la cláusula IN va una lista de una sola columna o atributo, y siempre antes va un atributo que debe coincidir en su tipo de dato con el tipo de dato de la lista.

Cláusula LIKE

Otra cláusula muy potente del lenguaje SQL es la que se utiliza para comparaciones con campos de tipo de cadenas de texto. Esta sentencia se podría utilizar por ejemplo para consultar cuáles son los clientes que viven en una calle que contiene el nombre Martín, pero que no se sabe si se ha escrito Martín o Martin (o sea que podría estar sin acento). ¡O sea podríamos tener en esta lista gente que viva en la localidad de San Martín o en la localidad Martín Coronado o en Martínez!

La cláusula de la que estamos hablando es el **LIKE**. Veamos cómo se utiliza en este caso:



SQL

```
SELECT * FROM Clientes c
WHERE calle LIKE '%Mart[ií]n%'
```

Analicemos un poco esta comparación para entender mejor como trabaja el **LIKE**:

Más info...

Primero que nada:

Las comparaciones que trabajan con **LIKE** van todas entre comillas simples.

Existen comodines como ser:

% : este comodín representa una cadena de cualquier largo que incluye texto, números y espacios en blanco

_ : este comodín representa un solo carácter pero este puede ser una letra o un número

? : este comodín representa un solo carácter de tipo numérico (o sea un dígito)

[] : entre corchetes vamos a poder colocar todos aquellos caracteres o números posibles que pueden ir en un solo lugar. O sea es como si pusiéramos el guion bajo pero le diéramos solo una lista de opciones posibles refinando así los valores que se pueden ingresar en ese lugar.

Veamos entonces en detalle lo que pasó con nuestra consulta:

Al colocar el % al comienzo y al final estamos representando un texto que no nos preocupa cómo comienza ni cómo termina, siempre y cuando contenga la palabra que nos interesa.

Como no sabíamos si iba a estar escrito o no con acento entonces colocamos entre corchetes las dos opciones de "i" (o sea "i" e "í").



Veamos otro ejemplo:

Buscar los nombres de las calles que comiencen con N o J, luego viene una vocal y a continuación un texto cualquiera que termina con dos números. Esta condición sería así:



SQL

```
SELECT * FROM Clientes c
WHERE calle LIKE '[NJ][aeiou]%%'
```

< Anterior

Siguiente >