

## Consultas con cadenas de caracteres

Nos toca adentrarnos en la búsqueda de caracteres, algo que será extremadamente útil en los diseños web ya que siempre trabajamos con textos que han de ser consultados.

Incorporando el like y comodines:

Hasta el modulo anterior llegamos a buscar cadenas de caracteres usando el operador "=", donde la condición que se aplica es por una comparación exacta. Pero en la práctica no siempre vamos a conocer si la palabra que estamos buscando se escribió en diferente forma y va a depender de quien haya "insertado" o "actualizado" los datos en nuestra base pudiendo muchas veces ser el usuario de nuestro desarrollo web.

Es por esto que cuando busquemos por coincidencia no completa con ayuda de dos comodines vamos a poder potenciar nuestras consultas.

Comienzo por explicar los comodines y luego comenzaremos con los ejemplos.

- El **comodín %** (porcentaje) indica que podría haber más caracteres en ese lugar específico de la cadena de caracteres o patrón de comparación.
- El **comodín \_** (guion bajo) indica que se puede reemplazar por un carácter cualquiera, solo 1 dentro del patrón de comparación.

En ambos casos solo se pueden reemplazando el = por like.

Con la explicación de seguro nace la pregunta: Pero ¿cuándo podemos usar cada uno de ellos?

El comodín guion bajo:

En el caso del comodín "\_" nos será de extrema utilidad cuando busquemos una cadena de caracteres y nos encontremos con la problemática de no conocer el estado de un carácter. El ejemplo más común es cuando no sabemos si se ingresó un dato capitalizado (la primera letra en mayúscula) o completamente en singular, algo que ocurre mucho con el nombre de los meses o incluso con cualquier palabra considerando que los dispositivos móviles suelen recomendar automáticamente la primera letra de las palabras en mayúscula cuando interactúan con un formulario web.

Pasemos a un ejemplo suponiendo que tengo una tabla Clientes con la siguiente configuración:

Cliente	
Pk	NroCli
	Nyape
	domicilio
	localidad
	SaldoCli

Ejemplo 1:

Es necesario buscar a los clientes que pertenecen a Capital Federal, pero no podemos estar seguros que todos hayan sido ingresados con la "C" y la "F" en mayúscula.

La consulta quedaría similar a los siguiente, si no tuviéramos los comodines

```
SELECT *
FROM CLIENTE
WHERE localidad = 'Capital Federal'
or localidad = 'capital federal'
```

Pero con el uso del comodín podríamos tener algo reducido similar a esto.

```
SELECT *
FROM CLIENTE
WHERE localidad like ' _apital _ederal'
```

Y nos evitaríamos o mejor dicho tendríamos en cuenta ejemplos como este:

NROCLI	NYAPE	DOMICILIO	LOCALIDAD	SALDOCLI
302	Mercedes Fuentes	Rosario 571	Capital federal	0.00
301	Carlos Andrés Rosas	Mitre 134	capital federal	0.00
300	Matias Fuentes	Mitre 134	Capital Federal	-4777.00

Donde por algún motivo para el cliente con Nrocli = 302 aparece solo la primera parte capitalizada y en el segundo caso completamente en minúsculas a diferencia de último donde ambas palabras se encuentran capitalizadas.

Veamos otro ejemplo donde los errores ortográficos pueden complicarnos las consultas en especial las tildes.

#### Ejemplo 2:

Supongamos que requerimos un listado de clientes que pertenezcan a la localidad Morón. En este caso nos podemos encontrar con la situación de que no se haya incorporado la tilde en la “o”, además de no estar en mayúsculas la “M”.

Nuestra consulta podría quedar de la siguiente forma:

```
SELECT *  
FROM CLIENTE  
WHERE localidad like ' or n'
```

Los comodines pueden ser usados más de una vez en la sentencia e incluso combinarlos.

Y dada la consulta podría obtener los clientes con las diferentes características mencionadas.

NROCLI	NYAPE	DOMICILIO	LOCALIDAD	SALDOCLI
179	Natalia Lopéz	Chacabuco 476	moron	5764.67
230	Lautaro González	Alvarado 793	Morón	9294.67
270	Maria Dominguez	Moreno 435	Morón	-6880.00
299	Belen Silva	Cordoba 56	morón	-9756.67

Y como se puede ver en el extracto, logramos evitar los errores humanos que pudieran surgir, considerando “moron”, “morón” y la correcta “Morón”.

#### El comodín porcentaje:

El porcentaje, considero en mi experiencia, que es uno de los más usados y es extremadamente útil. Desde que lo aprendí en la mayoría de mis consultas buscando

cadenas de caracteres lo incluían. Pero veamos en cuáles ejemplos podría sernos útil, aunque seguro existirán muchos más donde poder sacarle provecho.

Los ejemplos más claros se darán cuando tengamos campos de caracteres con texto variado y estemos buscando algo muy específico. Es normal usarlo para cuando tenemos campos de texto que se completan con mucha información. Vamos a analizar un caso un poco más sencillo, realizando una búsqueda sobre un atributo que combina el nombre y apellido.

### Ejemplo 3:

Supongamos que me interesa encontrar en mi listado de clientes a aquellos con apellido González, pero desconozco los nombres y si poseen otro apellido. Este ejemplo es muy común en diseños web cuando queremos hacer búsqueda de productos incluso, donde solo ingresemos algunas palabras claves y la búsqueda arroje resultados semejantes.

Continuando con el ejemplo planteado, y considerando la misma tabla CLIENTE el ejemplo quedaría de la siguiente forma:

```
SELECT *  
FROM CLIENTE  
WHERE NYAPE like '%González%'
```

Cabe aclarar que se puede utilizar el comodín más de una vez, y en este caso aprovechamos esto para evitar problemas como que el apellido no sea lo último que se escribió y tenga otro apellido y, obviamente, sin importar los nombres que pudiera tener.

Y podríamos obtener un resultado similar a esto:

NROCLI	NYAPE	DOMICILIO	LOCALIDAD	SALDOCLI
60	Karina González Blanco	Pavón 1955	Capital Federal	0.00
62	Darío Pérez González	Rosario 2022	Capital Federal	0.00
230	Lautaro González	Alvarado 793	Morón	9294.67

Donde como se puede apreciar contamos con ejemplos variados:

- Un apellido adicional por detrás.
- Un nombre y un apellido por delante.
- Un único nombre por delante.

Y algo para aclarar, el “guion bajo” y el “porcentaje” se pueden combinar. En nuestro ejemplo podríamos reemplazar la “á” para evitar encontrarnos con algún olvido al cargar los datos, quedando de esta forma:

```
SELECT *  
FROM CLIENTE  
WHERE NYAPE like '%Gonz_lez%'
```

Permitiendo encontrar otro cliente que de la forma anterior posiblemente no hubiera sido factible de ser encontrado.

NROCLI	NYAPE	DOMICILIO	LOCALIDAD	SALDOCLI
60	Karina González Blanco	Pavón 1955	Capital Federal	0.00
62	Darío Pérez González	Rosario 2022	Capital Federal	0.00
230	Lautaro González	Alvarado 793	Morón	9294.67
401	Germán Gonzalez	Belgrano 2018	Vicente López	4050.00

Propongo otro ejemplo que se relacione a la búsqueda de productos.

Vamos a suponer que tenemos una aplicación web para consumo de canciones, la tabla tiene la siguiente configuración:

Track	
TrackId	
Name	
AlbumId	
MediaTypeId	
GenreId	
Composer	
Milliseconds	
Bytes	
UnitPrice	

#### Ejemplo 4:

Se requiere sacar un informe de aquellas canciones que empiecen con la palabra *love*.

Para poder resolver esta consulta vamos a hacer uso de una función con string que vimos en el módulo anterior para poder comparar directamente con minúsculas. Quedaría de la siguiente forma.

```
SELECT *  
FROM TRACK  
WHERE lower(name) like 'love%'
```

Dándonos un resultado similar a este:

TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer
24	Love In An Elevator	5	1	1	Steven Tyler, Joe Perry
56	Love, Hate, Love	7	1	1	Jerry Cantrell, Layne Staley
413	Loverman	35	1	3	Cave
440	Love Gun	37	1	1	Paul Stanley
493	Love Is Blind	40	1	1	David Coverdale/Earl Sclan
571	Love Of My Life	46	1	1	Carlos Santana & David Byrne
751	Love Child	58	1	1	Bolin/Coverdale
803	Love Conquers All	64	1	1	Blackmore, Glover, Marsh
808	Love Don't Mean a Thing	65	1	1	D.Coverdale/G.Hughes
828	Love Bites	67	1	1	NULL
1042	Love And Marriage	83	1	12	jimmy van heusen/sammy
1055	Loves Been Good To Me	83	1	12	rod mckuen

Y la lista continúa, por consiguiente, si nos encontramos ante una búsqueda específica o proponemos en nuestro sitio la búsqueda de algún texto podríamos hacer uso del comodín porcentaje, y regresando valores que se asemejen a las primeras cadenas de caracteres ingresadas.

Algo a tener en cuenta es que el motor de base de datos que elijamos va a sufrir considerablemente si la cantidad de registros es muy grande es por esto que aconsejo que siempre inicien una búsqueda con al menos 4 caracteres.

## Escapando comodines:

Al incorporar en nuestras sentencias el like y los comodines nos vamos a encontrar, cada tanto, con la problemática de realizar búsquedas que requieran buscar los comodines, guion bajo y porcentaje, como parte de la cadena de caracteres, por ejemplo, al querer buscar un atributo que contenga un porcentaje de descuento, de seguro tendrá incorporado el signo porcentaje.

“25 % de descuento en tu próxima compra...”

Si queremos buscar los tipos de porcentajes realizados, vamos a tener un serio problema o no lo vamos poder aplicar directamente.

Es por esto que los motores nos permiten “escapar” caracteres y cada uno lo puede llegar a implementar de diferente forma:

En SQL Server y MySQL se suele incorporar en la cláusula, la palabra reservada ESCAPE y determinar cuál será el símbolo a utilizar para esa consulta.

```
WHERE comment LIKE '%30!%%' ESCAPE '!'
```

En el ejemplo se busca un comentario que diga 30% y para encontrarlo se utiliza el símbolo “!” como escape, y de esta forma el segundo porcentaje es parte del texto a buscar.

## Un comodín adicional, los corchetes:

Este último comodín que veremos tiene una complejidad mayor, pero es bastante usado. La idea de incorporar corchetes es con el fin de buscar un carácter que se encuentre dentro de un rango específico.

Se suele usar cuando buscamos un nombre pero que se encuentre dentro de un grupo de letras que podremos definir dentro de los corchetes.

Veamos un ejemplo haciendo uso de la tabla customer que tiene el siguiente formato:

Customer	
CustomerId	
FirstName	
LastName	
Company	
Address	
City	
State	
Country	
PostalCode	
Phone	
Fax	
Email	
SupportRepId	

#### Ejemplo 5:

Supongamos que buscamos los clientes que comiencen sus apellidos con A, B o C con el fin de darles una promoción.

Para esto voy a combinar el uso de los corchetes y el porcentaje.

```
Select *
From customer
Where lastname like '[A-C]%'
```

Con los corchetes incorporando la “A” luego un guion medio y a continuación la “C” vamos a cubrir la A, B y C. Y agregando el porcentaje por detrás no va a importar con que continúen obteniendo un resultado similar a este:

CustomerId	FirstName	LastName	Company	Address	City	State
12	Roberto	Almeida	Riotur	Praça Pio X, 119	Rio de Janeiro	RJ
18	Michelle	Brooks	NULL	627 Broadway	New York	NY
21	Kathy	Chase	NULL	801 W 4th Street	Reno	NV
26	Richard	Cunningham	NULL	2211 W Berry Street	Fort Worth	TX
28	Julia	Barnett	NULL	302 S 700 E	Salt Lake City	UT
29	Robert	Brown	NULL	796 Dundas Street West	Toronto	ON
39	Camille	Bernard	NULL	4, Rue Milton	Paris	NA



También lo podemos usar para búsqueda de rango de números o combinarlos como mejor nos convenga.

Con la incorporación del like y los comodines vistos, nuestras consultas se volverán más dinámicas y robustas, ideal para implementar soluciones y funcionalidades en nuestros diseños de aplicaciones o incluso para la confección de informes.