

# EXPRESIONES REGULARES EN MySQL

Las expresiones regulares son una poderosa herramienta para determinar si una cadena de texto se ajusta a un determinado patrón, que también es posible utilizar en la cláusula WHERE de una sentencia en MySQL.

En este artículo comentamos las posibilidades que ofrece, la sintaxis y las peculiaridades del uso de expresiones regulares en este sistema de gestión de bases de datos, con ejemplos prácticos.

## Palabras clave REGEXP y RLIKE

La comparación entre una cadena de texto y una expresión regular se realiza con la palabra clave REGEXP. También se puede utilizar indistintamente el sinónimo RLIKE.

Así, por ejemplo, para seleccionar todos los campos de la tabla “usuarios” en donde el campo “nombre” empieza por una vocal, podemos utilizar la sentencia:

```
SELECT * FROM usuarios WHERE nombre REGEXP '^[aeiouAEIOU]';  
o bien  
SELECT * FROM usuarios WHERE nombre RLIKE '^[aeiouAEIOU]';
```

En la expresión regular utilizada en el ejemplo, el carácter especial “^” indica el comienzo de la cadena. El primer carácter de los nombres coincidentes puede ser cualquiera de los indicados en el interior de los corchetes cuadrados.

## Caracteres especiales en expresiones regulares

### “^” y “\$” – Comienzo y fin de la cadena

Ya hemos visto que el carácter “^” indica el comienzo de la cadena. De la misma forma, “\$” indica el final de la misma. Así, para seleccionar todos los nombres que terminan en vocal en la tabla usuarios del ejemplo, utilizaríamos la sentencia:

```
SELECT * FROM usuarios WHERE nombre REGEXP '[aeiouAEIOU]$';
```

### “.” – Coincidencia con cualquier carácter

El carácter especial “.” coincide con cualquier carácter que aparezca en la cadena. Por ejemplo, para seleccionar todos los nombres de tres caracteres, sean cuales sean, podemos utilizar la sentencia:

```
SELECT * FROM usuarios WHERE nombre REGEXP '^...$';
```

La expresión regular en este caso indica que el comienzo de la cadena (“^”) debe ir seguido de cualquier carácter (“.”), cualquier carácter (“.”), cualquier carácter (“.”), y el fin de la cadena (“\$”).

## “?”, “+” y “\*” – Indicadores de repetición

El carácter especial “?” en una expresión regular indica que el carácter que le precede puede aparecer cero o una veces en la cadena

El carácter especial “+” indica que el carácter que le precede puede aparecer una o más veces en la cadena

Por último, el carácter “\*” indica que el carácter que le precede puede aparecer cero, una o más veces en la cadena.

Por ejemplo, para seleccionar todos los nombres que empiezan por A y terminan en “o” o en “os”, podemos utilizar la sentencia:

```
SELECT * FROM usuarios WHERE nombre REGEXP '^A.*os?$';
```

La expresión regular indica que el comienzo de la cadena (“^”) debe ir seguido de una “A” (“A”), seguida de cualquier carácter (“.”) cero o más veces (“\*”), seguidos de una “o”, opcionalmente seguida de una “s” (“s?”) al final de la cadena (“\$”).

Si queremos que los nombres seleccionados contengan al menos un carácter entre la “A” inicial y la “o”, utilizaríamos en su lugar la sentencia:

```
SELECT * FROM usuarios WHERE nombre REGEXP '^A.+os?$';
```

## {n,m} – Indicador genérico de repetición

Con la construcción {n}, expresamos que el carácter precedente debe aparecer exactamente n veces consecutivas.

Con la construcción {n,m} expresamos que el carácter precedente debe aparecer en la cadena un mínimo de n veces consecutivas, y un máximo de m veces. Si no se especifica m, el número máximo de apariciones no está limitado.

Los indicadores de repetición presentados en el apartado anterior se pueden escribir también utilizando esta notación:

- “a?” es equivalente a “a{0,1}”
- “a+” es equivalente a “a{1,}”
- “a\*” es equivalente a “a{0,}”

El valor de n y m no puede ser superior al parámetro de MySQL RE\_DUP\_MAX (por defecto, 255)

## “|” – Coincidencia de una u otra secuencia de caracteres

Con el carácter “|” (barra vertical, pipa) podemos escribir una expresión regular que se cumple si la cadena contiene cualquiera de entre dos (o más) secuencias de caracteres. Por ejemplo, para seleccionar los nombres que contienen “Juan”, “Pedro” o “Luis”, podemos utilizar la sentencia:

```
SELECT * FROM usuarios WHERE nombre REGEXP 'Juan|Pedro|Luis';
```

### “( )” – Secuencia de caracteres

Una secuencia de caracteres se puede encerrar entre paréntesis, para indicar que un carácter especial que le sigue afecta a toda la secuencia. Por ejemplo, para escribir una expresión regular que se cumple si en la cadena aparece la secuencia “Pérez ” dos o más veces, utilizamos:

```
(Pérez ){2,}
```

### “[]”, “[^]” – Coincidencia/no coincidencia con un carácter de entre un conjunto de caracteres

Encerrando un conjunto de caracteres entre corchetes cuadrados, indicamos que la coincidencia se cumple si en la cadena aparece cualquiera de ellos: la expresión “[arq]” se cumple si la cadena contiene el carácter a, el carácter r or el carácter q (una expresión equivalente a la de este ejemplo, utilizando el carácter especial “|”, sería: “a|r|q”).

También podemos indicar un rango de caracteres con un guión: “[c-f3-7]” se cumple si la cadena contiene cualquiera de los caracteres c, d, e, f, 3, 4, 5, 6 o 7

Para incluir el carácter “-” entre los caracteres posibles, debe colocarse al principio o al final. Por ejemplo, la expresión “[abc-]” se cumple si en la cadena aparecen los caracteres “a”, “b”, “c” o “-”.

Para incluir el carácter “[” entre los caracteres posibles, debe colocarse inmediatamente después del carácter “[”. Por ejemplo: “[[]abc]”

Utilizando el carácter “^” inmediatamente después del corchete de apertura, indicamos que los caracteres a continuación no deben aparecer en la cadena. Por ejemplo, “[^a-z]” se cumple si la cadena no contiene ningún carácter alfabético entre la “a” y la “z”.

### [.carácter.] – Coincidencia con un carácter

La construcción [.carácter.] puede aparecer en el interior de una construcción “[ ]” para representar un determinado carácter. “carácter” es o bien el nombre de un carácter, o el propio carácter. Por ejemplo, para buscar registros en donde el nombre contiene el carácter “line feed” o el carácter “escape”:

```
SELECT * FROM usuarios WHERE nombre REGEXP '[.LINEFEED.][.ESCAPE.]';
```

La tabla siguiente muestra los nombres de carácter que se pueden utilizar en esta construcción:

Name	Character	Name	Character
NUL	0	SOH	001
STX	002	ETX	003

<b>Name</b>	<b>Character</b>	<b>Name</b>	<b>Character</b>
EOT	004	ENQ	005
ACK	006	BEL	007
alert	007	BS	010
backspace	'b'	HT	011
tab	't'	LF	012
newline	'n'	VT	013
vertical-tab	'v'	FF	014
form-feed	'f'	CR	015
carriage-return	'r'	SO	016
SI	017	DLE	020
DC1	021	DC2	022
DC3	023	DC4	024
NAK	025	SYN	026
ETB	027	CAN	030
EM	031	SUB	032
ESC	033	IS4	034
FS	034	IS3	035
GS	035	IS2	036
RS	036	IS1	037
US	037	space	' '
exclamation-mark	'!'	quotation-mark	'\"'
number-sign	'#'	dollar-sign	'\$'
percent-sign	'%'	ampersand	'&'
apostrophe	'\"'	left-parenthesis	' ('
right-parenthesis	')'	asterisk	'*'
plus-sign	'+'	comma	','
hyphen	'-'	hyphen-minus	'-'
period	'.'	full-stop	'.'
slash	'/'	solidus	'/'
zero	'0'	one	'1'
two	'2'	three	'3'
four	'4'	five	'5'
six	'6'	seven	'7'
eight	'8'	nine	'9'
colon	':'	semicolon	','
less-than-sign	'<'	equals-sign	'='
greater-than-sign	'>'	question-mark	'?'
commercial-at	'@'	left-square-bracket	'['
backslash	'\"'	reverse-solidus	'\"'
right-square-bracket	']'	circumflex	'^'
circumflex-accent	'^'	underscore	'_'
low-line	'_'	grave-accent	'`'

Name	Character	Name	Character
left-brace	'{'	left-curly-bracket	'{'
vertical-line	' '	right-brace	'}'
right-curly-bracket	'}'	tilde	'~'
DEL	177		

### **[[:character\_class:]]**

[[:character\_class:]] puede utilizarse en el interior de una construcción “[” para representar todos los caracteres que pertenecen a una misma clase. La tabla siguiente muestra las clases que se pueden utilizar en esta expresión:

Character Class Name	Meaning
alnum	Alphanumeric characters
alpha	Alphabetic characters
blank	Whitespace characters
cntrl	Control characters
digit	Digit characters
graph	Graphic characters
lower	Lowercase alphabetic characters
print	Graphic or space characters
punct	Punctuation characters
space	Space, tab, newline, and carriage return
upper	Uppercase alphabetic characters
xdigit	Hexadecimal digit characters

### **[[:<:]], [[:>:]]**

Estos marcadores indican el comienzo y el fin de una palabra, respectivamente.

Por ejemplo, para seleccionar todos los nombres que contienen la palabra “pedro”:

```
SELECT * from usuarios WHERE nombre REGEXP '[[:<:]]pedro[[:>:]]';
```

## **Literales de caracteres especiales en una expresión regular**

Para utilizar un literal de un carácter especial en una expresión regular, hay que precederlo de dos caracteres “\” (backslash). Por ejemplo, para comprobar si una cadena de texto contiene la secuencia “1+2”, dado que el signo “+” es un carácter especial, debemos utilizar la expresión regular “1\+2”.