EL OPERADOR $REGEX

En esta práctica vamos a trabajar con el operador $REGEX el cual nos da la posibilidad de buscar una serie de expresiones dentro de la colección con la que estamos trabajando. La sintaxis que utiliza este operador es la siguiente:

{ <field>: { $regex: /pattern/, $options: '<options>' } }

{ <field>: { $regex: 'pattern', $options: '<options>' } }

{ <field>: { $regex: /pattern/<options> } }

Como se puede apreciar la sintaxis de este operador cuenta con diferentes campos:

1. **Field: Tendremos que especificar el campo del documento con el que queremos trabajar.**
2. **$regex: Para proceder a buscar la expresión del documento tendremos que especificar el operador $regex**
3. **/pattern/: En este apartado tendremos que especificar que queremos buscar , por ejemplo si queremos buscar en que en el campo nombre contenga la cadena /mes/**
4. **Options: En opciones podemos especificar determinadas premisas de este operador, por ejemplo la opción "i" que lo que hace es no distinguir entre mayúsculas y minúsculas.**

Ejemplo de consulta:

{ <**Descripcion**>: {$regex: /B.Bcc/, $i}} En este caso buscaríamos en el campo descripción de aquel documento que contenga la cadena "b(Cualquier caracter)bcc"

Definition

$regex

Provides regular expression capabilities for pattern matching strings in queries. MongoDB uses Perl compatible regular expressions (i.e. "PCRE") version 8.42 with UTF-8 support.

Option   Description   Syntax Restrictions

i   Case insensitivity to match upper and lower cases. For an example, see Perform Case-Insensitive Regular Expression Match.

m

For patterns that include anchors (i.e. ^ for the start, $ for the end), match at the beginning or end of each line for strings with multiline values. Without this option, these anchors match at beginning or end of the string. For an example, see Multiline Match for Lines Starting with Specified Pattern.

If the pattern contains no anchors or if the string value has no newline characters (e.g. \n), the m option has no effect.

x

"Extended" capability to ignore all white space characters in the $regex pattern unless escaped or included in a character class.

Additionally, it ignores characters in-between and including an un-escaped hash/pound (#) character and the next new line, so that you may include comments in complicated patterns. This only applies to data characters; white space characters may never appear within special character sequences in a pattern.

The x option does not affect the handling of the VT character (i.e. code 11).

Requires $regex with $options syntax

s   Allows the dot character (i.e. .) to match all characters including newline characters. For an example, see Use the . Dot Character to Match New Line.
        Requires $regex with $options syntax

Option	Description	Syntax Restrictions
i	Case insensitivity to match upper and lower cases. For an example, see Perform Case-Insensitive Regular Expression Match.	
m	For patterns that include anchors (i.e. ^ for the start, $ for the end), match at the beginning or end of each line for strings with multiline values. Without this option, these anchors match at beginning or end of the string. For an example, see Multiline Match for Lines Starting with Specified Pattern.	

Option	Description	Syntax Restrictions
	If the pattern contains no anchors or if the string value has no newline characters (e.g. `\n`), the `m` option has no effect.	
x	"Extended" capability to ignore all white space characters in the `$regex` pattern unless escaped or included in a character class.

Additionally, it ignores characters in-between and including an un-escaped hash/pound (`#`) character and the next new line, so that you may include comments in complicated patterns. This only applies to data characters; white space characters may never appear within special character sequences in a pattern.

The `x` option does not affect the handling of the VT character (i.e. code 11). | Requires `$regex` with `$options` syntax |
| s | Allows the dot character (i.e. `.`) to match all characters *including* newline characters. For an example, see Use the . Dot Character to Match New Line. | Requires `$regex` with `$options` syntax |