

# Explorando a cláusula WHERE

Já tivemos alguns contatos com o comando WHERE e vimos que ele pode ser utilizado para filtrar os resultados de uma pesquisa.

Nos exemplos mostrados anteriormente, este comando foi utilizado em conjunto com os operadores de comparação. Nesta situação, o filtro é aplicado com uma expressão lógica. Mas, é possível adicionar mais de uma expressão através de operadores lógicos.

Além desses operadores, esta cláusula suporta outros tipos de operadores que veremos a seguir.

## Operadores lógicos

Os operadores lógicos testam as expressões e retornam um valor verdadeiro ou falso.

### Operador AND


O operador lógico **AND (E)** retorna verdadeiro se as duas expressões passadas para ele forem verdadeiras. A sua tabela-verdade é:

A	B	A AND B
FALSO	FALSO	FALSO
FALSO	VERDADEIRO	FALSO
VERDADEIRO	FALSO	FALSO
VERDADEIRO	VERDADEIRO	VERDADEIRO

Note que apenas quando A e B são verdadeiros o resultado é verdadeiro.

Assim, em um exemplo como o abaixo:

```
1
2 SELECT * FROM carro WHERE cod = 1 AND placa = 'asd0989';
3
```



A consulta só retornará dados se o campo *cod* for igual a 1 e se o campo *placa* for igual a “asd0989”. As duas condições precisam ser verdadeiras ao mesmo tempo.

## Operador OR


O operador **OR (OU)** retorna verdadeiro quando uma ou as duas expressões são verdadeiras. Veja a tabela-verdade do OR:

A	B	A OR B
FALSO	FALSO	FALSO
FALSO	VERDADEIRO	VERDADEIRO
VERDADEIRO	FALSO	VERDADEIRO
VERDADEIRO	VERDADEIRO	VERDADEIRO

Note que o operador OR só retorna falso se ambas as expressões forem falsas.

Por exemplo: na consulta abaixo:

```
1
2 SELECT * FROM aluno WHERE nome = 'wagner' OR cod = 5;
3
```



Serão retornados os registros cujo *nome* é “Wagner” **OU** o *cod* é igual a 5. Se qualquer uma das condições for considerada verdadeira, a tupla já é retornada.

## Operador NOT


O operador NOT inverte o resultado da expressão. Veja a tabela-verdade dele:

A	NOT A
---	-------

VERDADEIRO	FALSO
FALSO	VERDADEIRO

Se um valor for verdadeiro, ele vira falso. Se for falso, ele vira verdadeiro. Por exemplo:

```
1  
2 SELECT * FROM carro WHERE NOT cod = 2;  
3
```

  
MySQL

A consulta acima, irá retornar os registros onde *cod* não for 2.

## Condições IS NULL e IS NOT NULL

Como o NULL é uma expressão especial, ele não pode ser utilizado em uma comparação. Ou seja: para obtermos os registros com valores nulos, não é possível realizar a consulta abaixo:

```
1  
2 SELECT * FROM fornecedores  
3 WHERE CEP = NULL;  
4
```

  
MySQL

Assim, o SQL também introduziu o operador IS NULL. Com ele é possível verificar se o campo possui valores NULL. Veja o exemplo:

```
1  
2 SELECT * FROM fornecedores  
3 WHERE CEP IS NULL;  
4
```

  
MySQL

A consulta acima, irá retornar todos os registros cujo o campo *CEP* possua o valor *NULL*.

O operador IS NOT NULL realiza o filtro contrário, trazendo apenas os registros que **NÃO** são NULL. Por exemplo:

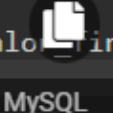
```
1  
2 SELECT * FROM fornecedores  
3 WHERE CEP IS NOT NULL;  
4
```

  
MySQL

## Operador BETWEEN

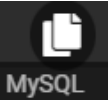
Podemos também selecionar elementos que estejam dentro de um intervalo de valores com o operador BETWEEN. A sintaxe desse comparador é a seguinte:

```
SELECT * FROM tabela WHERE campo BETWEEN valor_inicial AND valor_final
```



Geralmente o operador BETWEEN é utilizado para comparar um intervalo de datas, mas pode ser utilizado com qualquer tipo de dado. Por exemplo: para selecionarmos os fornecedores com código entre 3 e 7, a consulta poderia ser...

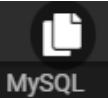
```
1  
2 SELECT * FROM fornecedores WHERE codigo BETWEEN 3 and 7;  
3
```



## Operador IN

Outro operador que podemos usar é o IN. Ele retorna apenas elementos que sejam iguais a um dos valores apresentados na lista. Veja seu uso:

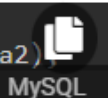
```
1  
2 SELECT * FROM fornecedores WHERE codigo IN (1, 5, 9);  
3
```



Acima, são selecionados os registros com código igual a 1, 5 ou 9.

O IN também pode ser utilizado com uma sub-consulta:

```
1  
2 SELECT * FROM tabela WHERE campo IN (SELECT campo FROM tabela2);  
3
```



## Operador LIKE

O operador LIKE é outro operador que pode ser utilizado com o comando WHERE. Porém, diferente dos operadores que vimos anteriormente, este só pode ser utilizado em cadeias de caracteres(string).

Com ele, é possível verificar se a cadeia de caracteres corresponde a um padrão especificado. Para definir este padrão, podemos utilizar uma série de caracteres curinga listados na tabela abaixo:

Caractere curinga	Descrição
<u>_</u> (sublinhado)	É válido para qualquer caractere nessa posição.
%	Qualquer cadeia de zero ou mais caracteres.
[]	Possibilita especificar um padrão para um caractere único, exemplo [a-f], [0-9], [A-Z].
[^]	Caractere único que não pertença ao padrão especificado.

## Operador REGEXP ou RLIKE

O REGEXP é parecido com o LIKE. Ele só pode ser utilizado em cadeias de caracteres(string) e verifica se a cadeia de caracteres corresponde ao padrão especificado. Este operador também possui um operador sinônimo que é o RLIKE. Ambos são equivalentes.

Nesses dois operadores pode-se utilizar expressões regulares para especificar o padrão da pesquisa.