

# CLÁUSULA WHERE

## FILTROS

OPERADORES RELACIONAIS : **>, <, =, >=, <=, <>**

OPERADORES LÓGICOS : **AND, OR, NOT**

DA LINGUAGEM SQL : **IS NULL, IS NOT NULL,  
LIKE, NOT LIKE, IN, NOT IN, BETWEEN,  
NOT BETWEEN, EXISTS, NOT EXISTS**

## SQL - OPERADORES

IS NULL	Verifica se o conteúdo da coluna é NULL (vazio)
IS NOT NULL	Negação do operador IS NULL
LIKE	<p>Compara cadeia de caracteres utilizando padrões de comparação:</p> <ul style="list-style-type: none"><li>% substitui zero, um ou mais caracteres</li><li>_ substitui um caractere</li></ul> <p>LIKE 'A%' inicia com a letra A</p> <p>LIKE '%A' termina com a letra A</p> <p>LIKE '%A%' tem a letra A em qualquer posição</p> <p>LIKE 'A_' string de dois caracteres, inicia com a letra A</p> <p>LIKE '_A' string de dois caracteres, termina com a letra A</p> <p>LIKE '_A_' string de três caracteres, letra A na segunda posição</p> <p>LIKE '%A_' tem a letra A na penúltima posição</p> <p>LIKE '_A%' tem a letra A na segunda posição</p>
NOT LIKE	Negação do operador LIKE
IN	Testa se um valor pertence a um conjunto de valores.
NOT IN	Negação do operador IN
BETWEEN	Determina um intervalo de busca: BETWEEN 'valor1' AND 'valor2'
NOT BETWEEN	Negação do operador BETWEEN
EXISTS	Verifica se um valor existe em um conjunto, levando em conta os valores nulos. Fato não considerado pelo operador IN.
NOT EXISTS	Negação do operador EXISTS.

# SELECT

A instrução SELECT seleciona e mostra um conjunto de dados de acordo com as condições estabelecidas. Quando é usada como linha de comando em um SGBD, os dados são visualizados na tela do computador.

Consiste de 3 cláusulas:

SELECT <.....**1**.....> FROM <.....**2**.....> WHERE <.....**3**.....>

**1 – Pode ter :**

**colunas de tabelas (atributos)**

**expressões com atributos**

**constantes**

**funções**

**\* representa todas as colunas de uma tabela.**

2 – Pode ter:

- Tabelas  
As colunas especificadas devem existir nas tabelas.

3 – Pode ter:

- Condições de busca relacionadas com os atributos selecionados

Vamos usar a tabela de CLIENTE como exemplo para ilustrar o resultado da pesquisa nesta tabela, onde IdCliente é a chave primária da tabela.

<u>IdCliente</u>	Nome	Endereço	<u>Cep</u>	<u>DtFunda</u>	<u>TotEmp</u>	Faturamento
123	Cia de malhas	Rua <u>Abc</u> ,6	12356-569	<b>01/02/2007</b>	10	12000000,00
124	Tatuzinho <u>Ltda</u>	Av. <u>Porc</u> 34	05485-000	<b>01/12/1990</b>	200	20000000,00
125	Coca-cola	<u>Av Cursino</u> ,2	03527-001	<b>25/02/1950</b>	2540	90000000,00
126	Agua de <u>fiori</u>	<u>Av Guarilhos</u> ,1	02458-256	<b>20/06/1986</b>	1200	15000000,10
127	Avon	Av. Paes leme	23658-987	<b>20/01/2000</b>	20	12345000,10
128	<u>Marabras</u>	<u>Av Floriano</u> , 23	06895-001	<b>29/01/1986</b>	200	12355,00
129	Renner	<u>Av Jurupis</u> 345	04569-256	<b>30/01/1995</b>	123	25000000,00

## EXEMPLO 1 - Projeção

Os atributos Nome, Cep e Faturamento são as colunas de dados da tabela CLIENTE escolhidos para serem mostrados na tela.

```
SELECT Nome, Cep, Faturamento FROM CLIENTE
```

Nome	<u>Cep</u>	Faturamento
Cia de malhas	12356-569	12000000,00
Tatuzinho Ltda	05485-000	20000000,00
Coca-cola	03527-001	90000000,00
Agua de <u>fiori</u>	02458-256	15000000,10
Avon	23658-987	12345000,10
<u>Marabras</u>	06895-001	12355,00
Renner	04569-256	25000000,00

## EXEMPLO 2 - Projeção e Seleção

Selecionando todas as colunas de Cliente que possuem menos de 501 funcionários, usando operadores relacionais.

```
SELECT * FROM CLIENTE WHERE TotEmp <= 500
```

IDCliente	Nome	Endereço	Cep	DtFunda	TotEmp	Faturamento
123	Cia de malhas	Rua Abc,6	12356-569	01/02/2007	10	12000000,00
124	Tatuzinho Ltda	Av. Porc 34	05485-000	01/12/1990	200	20000000,00
127	Avon	Av. Paes leme	23658-987	20/01/2000	20	12345000,10
128	Marabras	Av Floriano, 23	06895-001	29/01/1986	200	12355,00
129	Renner	Av Jurupis 345	04569-256	30/01/1995	123	25000000,00



### EXEMPLO 3 - Projeção e Seleção

Selecionando o Nome e o Faturamento da tabela CLIENTE calculando o Imposto de Renda do Faturamento de 27.5%.

**SELECT Nome, Faturamento, (Faturamento \* 27.5/100) FROM CLIENTE**



Exemplo 3		
Nome	Faturamento	IR
Cia de malhas	R\$ 12.000.000,00	3300000
Tatuzinho Ltda	R\$ 20.000.000,00	5500000
Coca-cola	R\$ 90.000.000,00	24750000
Agua de fiori	R\$ 15.000.000,10	4125000,0275
Avon	R\$ 12.345.000,10	3394875,0275
Marabras	R\$ 12.355,00	3397,625
Renner	R\$ 25.000.000,00	6875000



## Projeção e Seleção

Selecionando os dados com a cláusula WHERE podem ser usados os predicados

- Operadores Relacionais
- BETWEEN
- NULL
- LIKE
- EXISTS
- IN

## Operadores relacionais

Operador	Significado
=	Igual
!=	Diferente
>	Maior
!>	Não maior (menor ou igual )
<	Menor
!<	Não menor (maior ou igual )
>=	Maior ou igual
<=	Menor ou igual

## BETWEEN

Seleciona uma faixa de valores.

O exemplo abaixo seleciona a coluna Nome da tabela CLIENTE onde o total de empregados esteja entre os valores de 10 a 100

```
SELECT Nome FROM CLIENTE WHERE ToTemp BETWEEN 10 and 100
```

## IN

Seleciona dados que estão contidos em uma lista.

O exemplo abaixo seleciona todas as colunas da tabela CLIENTE onde o IDCliente está contido em uma lista.

```
SELECT * FROM CLIENTE WHERE IDCliente in ( 123,125,127)
```

## NULL

Seleciona conteúdos nulos.

O exemplo abaixo seleciona todas as colunas da tabela CLIENTE onde o endereço é nulo.

```
SELECT * FROM CLIENTE WHERE Endereço is NULL
```

Podemos usar o operador lógico |not|

```
SELECT Nome FROM CLIENTE WHERE Endereço is |not| NULL
```

Neste caso será mostrado todos os nomes onde o endereço não contém nulo.

## EXERCÍCIOS:

Usando a tabela CLIENTE escrever a instrução que seleciona cada um dos itens abaixo:

- a) todos os nomes, endereço e cep que contém o faturamento maior que 20.000.000,00.
- b) Todo IDCliente contido na lista (121, 122, 123, 126,127)
- c) Considerando que 1% do faturamento é pró-labore do proprietário, listar quais empresas possuem o pró-labore maior que 200.000,00.
- d) escrever as tabelas do resultado da seleção dos itens a, b, c.

## SELECT

A maioria das pesquisas em banco de dados é feita com várias tabelas ao mesmo tempo.

O produto cartesiano ocorre quando precisamos consultar dados em duas ou mais tabelas.

Cada tupla ( linha ) de uma tabela está associada a todas as tuplas da segunda tabela.

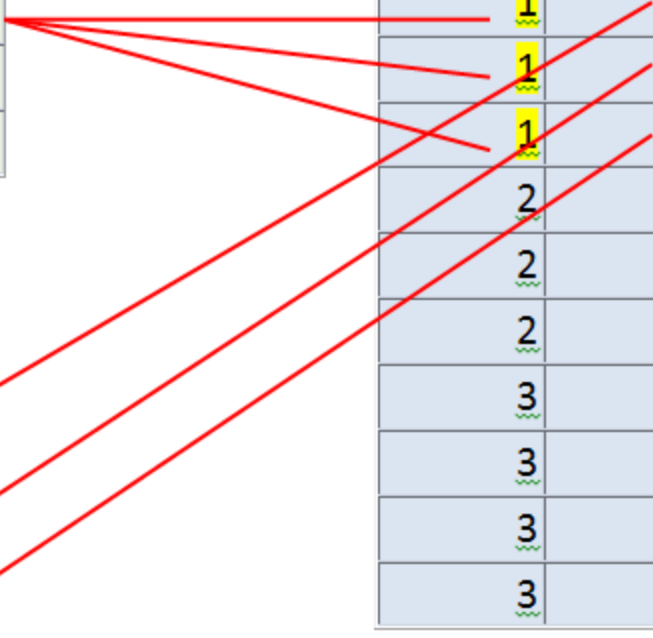
Ex.: Considerando um banco de dados com as seguintes tabelas: Uma tabela de programador com os dados pessoais, uma tabela de linguagem e uma terceira tabela Conhecimento, que associa cada programador com uma linguagem que ele conhece. Por exemplo, o programador de código (1) chama-se Arthur e está associado às linguagens (1) Algol, (3) Java e (4) Visual Basic.



Programador			
CodProg	Nome	Endereço	Fone
1	Arthur	R. Quatá 3	4513-4569
2	Raquel	R. Cabo 23	4562-9876
3	Ana Maria	R. ABC 908	1234-5698

Linguagem	
CodLing	Linguagem
1	Algol
2	Cobol
3	Java
4	Visual Basic

Conhecimento		
CodProg	CodLing	TempoExperiência
1	1	2
1	3	5
1	4	10
2	1	1
2	2	1
2	3	10
3	2	4
3	3	2
3	4	4
3	2	9



## EXEMPLO 1:

SELECT Programador .\*, Linguagem .\* FROM Programador, Linguagem

EXEMPLO 1					
<u>CodProg</u>	Nome	Endereço	Fone	<u>CodLing</u>	Linguagem
<u>1</u>	Arthur	R. Quatá <u>3</u>	4513-4569	<u>1</u>	<u>Algol</u>
<u>1</u>	Arthur	R. Quatá <u>3</u>	4513-4569	<u>2</u>	<u>Cobol</u>
<u>1</u>	Arthur	R. Quatá <u>3</u>	4513-4569	<u>3</u>	Java
<u>1</u>	Arthur	R. Quatá <u>3</u>	4513-4569	<u>4</u>	Visual Basic
<u>2</u>	Raquel	R. Cabo 23	4562-9876	<u>1</u>	<u>Algol</u>
<u>2</u>	Raquel	R. Cabo 23	4562-9876	<u>2</u>	<u>Cobol</u>
<u>2</u>	Raquel	R. Cabo 23	4562-9876	<u>3</u>	Java
<u>2</u>	Raquel	R. Cabo 23	4562-9876	<u>4</u>	Visual Basic
<u>3</u>	Ana Maria	R. ABC 908	1234-5698	<u>1</u>	<u>Algol</u>
<u>3</u>	Ana Maria	R. ABC 908	1234-5698	<u>2</u>	<u>Cobol</u>
<u>3</u>	Ana Maria	R. ABC 908	1234-5698	<u>3</u>	Java
<u>3</u>	Ana Maria	R. ABC 908	1234-5698	<u>4</u>	Visual Basic

### EXEMPLO 3:

Seleciona o nome do programador, o nome da linguagem e o tempo de experiência.

```
SELECT Programador.Nome, Linguagem.Linguagem,  
Conhecimento.TempoExperiência  
FROM Programador,Linguagem,Conhecimento  
WHERE Programador.CodProg = Conhecimento.CodProg  
AND Linguagem.CodLing = Conhecimento.CodLing;
```

### EXEMPLO 3

Nome	Linguagem	TempoExperiência
Arthur	Algol	2
Arthur	Java	5
Arthur	Visual Basic	10
Raquel	Algol	1
Raquel	Cobol	1
Raquel	Java	10
Ana Maria	Cobol	4
Ana Maria	Java	2
Ana Maria	Visual Basic	4

#### EXEMPLO 4:

Para selecionar quais programadores conhecem a linguagem Visual Basic e mostrar o nome e tempo de experiência, podemos selecionar da seguinte forma:

```
SELECT Programador.Nome, Conhecimento.TempoExperiência  
FROM Programador, Linguagem, Conhecimento  
WHERE Programador.CodProg = Conhecimento.CodProg  
AND Linguagem.CodLing = Conhecimento.CodLing  
AND Linguagem.Linguagem = 'Visual Basic';
```

Resulta:

EXEMPLO 4	
Nome	TempoExperiência
Arthur	10
Ana Maria	4

## Exercícios:

Usando as mesmas tabelas dos exemplos, selecione:

- 1) O nome, endereço e o telefone dos programadores que conhecem a linguagem JAVA
- 2) O Nome dos programadores que possuem mais de 2 anos de experiência em todas as linguagem que conhecem.
- 3) O nome dos programadores que não conhecem Visual Basic.