

**AULA 07****SQL - DQL – Linguagem de Consulta de Dados****1. Alterando o Heading (Cabeçalho) da coluna**

**Sintaxe:**     SELECT nome da coluna AS “cabeçalho da coluna”  
               FROM nome da tabela

**Exemplo:**

SELECT codven AS Número, nomeven AS Nome, salfixo AS rendimentos, Faixa AS comissão  
 FROM vendedor;

**2. Manipulando dados numéricos: Operadores Aritméticos**

Símbolo	Operação	Pode ser usado com (MS-SQL Server 7.0)
+	Adição	<i>Int, smallint, tinyint, numeric, decimal, float, real, money</i>
-	Subtração	<i>Int, smallint, tinyint, numeric, decimal, float, real, money</i>
/	Divisão	<i>Int, smallint, tinyint, numeric, decimal, float, real, money</i>
*	Multiplicação	<i>Int, smallint, tinyint, numeric, decimal, float, real, money</i>
%	Módulo	<i>Int, smallint e tinyint</i>

**Exemplo:**

SELECT nomeven, (salfixo \* 2) AS Salário FROM vendedor;

Operadores de Comparação		Operadores Lógicos	
=	Igual	AND	E lógico
<> ou !=	diferente	OR	OU lógico
<	menor do que	NOT	negação
>	maior do que		
>=	maior ou igual do que		
<=	menor ou igual do que		
!>	não maior		
!<	não menor		

## Problemas para Resolver

### **Problema 1:**

- Listar o num\_pedido, o código\_produto e a quantidade dos itens do pedido com a quantidade igual a 35 da tabela item\_do\_pedido.

**. Sintaxe:**       SELECT numped, codprod, quant  
                  FROM itempedido  
                  WHERE quant = 35;

### **Problema 2:** - Quais os clientes que moram em Niterói?

**Sintaxe:**       SELECT nomecli  
                  FROM cliente  
                  WHERE cidcli = 'Niterói';

### **Problema 3:** - Listar os produtos que tenham unidade igual a 'M' e valor unitário igual a R\$1,05 da tabela produto.

**Sintaxe:**       SELECT descricao  
                  FROM produto  
                  WHERE unidade = 'M' AND valunit = 1.05;

### **Problema 4:** - Liste os clientes e seus respectivos endereços, que moram em 'SÃO PAULO' ou estejam na faixa de CEP entre '30077000' e '30079000'.

**. Sintaxe:**       SELECT nomecli, endcli  
                  FROM cliente  
                  WHERE (cepcli >= '30077000' AND cepcli <= '30079000') OR cidcli = 'São Paulo';

A utilização dos parênteses é fundamental para a construção correta da sentença, pois sem eles as consultas podem ser analisadas de forma errada, devido à prioridade do operador AND ser maior que a prioridade do operador OR.

### **Problema 5:** - Mostrar todos os pedidos que não tenham prazo de entrega igual a 15 dias.

**. Sintaxe 1:**    SELECT numped  
                  FROM pedido  
                  WHERE NOT (prazo = 15);

Ou podemos alternativamente utilizar um operador de comparação <> que irá realizar a mesma operação de seleção.

**. Sintaxe 2:**    SELECT numped  
                  FROM pedido  
                  WHERE (prazo <> 15);

### 3. Operadores Between e NOT Between

**. Sintaxe:**

- WHERE <nome da coluna> BETWEEN <valor1> AND <valor2>
- WHERE <nome da coluna> NOT BETWEEN <valor1> AND <valor2>

Este operador propicia a pesquisa dentro de uma faixa determinada de valores. Tanto o VALOR1 quanto o VALOR2 têm de ser do mesmo tipo de dado da coluna.

**Problema 6:** - Listar o código e a descrição dos produtos que tenham o valor unitário na faixa de R\$ 0,32 até R\$2,00.

**. Sintaxe:**      SELECT codprod, descricao  
                     FROM produtos  
                     WHERE valunit between 0.32 and 2.00;

### 4. Operadores baseados em string de caracteres LIKE e NOT LIKE

**. Sintaxe:**

- WHERE <nome da coluna> LIKE <valor>;
- WHERE <nome da coluna> NOT LIKE <valor>;

Os operadores LIKE e NOT LIKE só trabalham sobre colunas que sejam do tipo CHAR. Eles têm praticamente o mesmo funcionamento que os operadores = e < >, porém o poder desses operadores está na utilização dos símbolos (%) e C) que podem fazer o papel de "curinga":

% => substitui uma palavra

\_ => substitui um caractere

Ex: LIKE 'LÁPIS %' pode enxergar todos os registros que começam com LÁPIS.  
'LÁPIS PRETO'      'LÁPIS CERA'      'LÁPIS BORRACHA'

LIKE 'BROCA N\_' pode enxergar todos os registros com o último caractere diferente:  
'BROCA N1'      'BROCA N9'      'BROCA N3'

LIKE '%ão' pode enxergar qualquer nome que termine em "ão".

Vamos ver mais alguns exemplos de utilização da cláusula LIKE. .

**Problema 7:** - Listar todos os produtos que tenham o seu nome começando por Q.

**Sintaxe:**      SELECT Codprod, descricao  
                     FROM produto  
                     WHERE descricao LIKE 'Q%';

**Problema 8:**

Listar os vendedores que não começam por 'Jo'.

**Sintaxe ANSI:**       SELECT Codven, nomeven  
                          FROM vendedor  
                          WHERE nomeven NOT LIKE 'Jo%';

## 5. Operadores baseados em listas IN e NOT IN

- WHERE <nome da coluna> IN <valores>;
- WHERE <nome da coluna> NOT IN <valores>;

Esses operadores pesquisam registros que estão ou não contidos no conjunto de <valores> fornecido. Estes operadores minimizam o uso dos operadores =, <>, AND e OR.

**Problema 9:** - Listar os vendedores que são da faixa de comissão A e B.

**Sintaxe:**       SELECT nomeven  
                          FROM vendedor  
                          WHERE faixa\_comis IN ('A', 'B');

## 6. Operadores baseados em valores desconhecidos: IS NULL e IS NOT NULL

- WHERE <nome da coluna> IS NULL;
- WHERE <nome da coluna> IS NOT NULL;

A utilização do valor nulo (NULL) é muito problemática, pois cada implementação da linguagem pode adotar qualquer representação para o valor nulo. O resultado da aplicação destes operadores permite o tratamento de valores nulos em colunas de uma tabela, selecionando as linhas correspondentes.

**Problema 10:** - Mostrar os clientes que não tenham inscrição estadual.

**Sintaxe:**       Select \* from clientes where IE is null

**Sintaxe:** SELECT nomeven, (salfixo \*1.75 + 120) AS “Novo Salário”  
FROM vendedor  
WHERE faixa = 'C'  
ORDER BY nomeven;

## 9. Utilizando Funções de Agregação sobre Conjuntos

### 9.1- Buscando Máximo e Mínimos (MAX,MIN)

```
SELECT MIN(sal_fixo), MAX(sal_fixo)
FROM vendedor;
```

### 9.2- Totalizando os valores de Colunas (SUM)

```
SELECT SUM(quant)
FROM itempedido
WHERE codprod = 78;
```

### 9.3- Calculando Médias (AVG)

```
SELECT AVG(sal_fixo)
FROM vendedor;
```

### 9.4- Contando os Registros (COUNT)

```
SELECT COUNT(*)
FROM vendedor
WHERE sal_fixo > 2500;
```

### 9.5- Utilizando a Cláusula DISTINCT

```
SELECT DISTINCT unidade,
FROM produto;
```

### 9.6- Agrupando Informações Seleccionadas (GROUP BY e HAVING)

```
SELECT numped, COUNT(*) as "total_produtos"
FROM itempedido
GROUP BY numped;
```

### 9.7- Utilizando com HAVING

```
SELECT numped, total_produtos = COUNT(*)
FROM itempedido
GROUP BY numped
HAVING COUNT(*) > 3;
```