

CONSULTAS COM OPERADORES

- Os **operadores aritméticos** são as ferramentas responsáveis pelas operações e cálculos matemáticos simples dentro de uma instrução SQL.

ADIÇÃO	+
SUBTRAÇÃO	-
MULTIPLICAÇÃO	*
DIVISÃO	/
RESTO DA DIVISÃO	%

CONSULTAS COM OPERADORES

- A empresa Virtual Ltda. deseja fazer uma **simulação** de aumento de salário em R\$ 100,00:

```
SELECT nome, salario + 100 FROM funcionario;
```

```
SELECT * FROM funcionario; # alteração não persiste
```

CONSULTAS COM OPERADORES

- Qual seria o valor dos salários se eles fossem reajustados em 20%?

```
SELECT nome, salario * 1.20 FROM funcionario;
```

ORDEM DE PRECEDÊNCIA

Caso queira mudar a ordem de precedência, será necessário usar parênteses:

SELECT 2 + 3 * 5;

SELECT (2 + 3) * 5;

OPERADORES DE COMPARAÇÃO

- Os operadores de comparação, também conhecidos como **operadores relacionais**, são ferramentas utilizadas na definição de **condições** em que há necessidade de comparação entre dois valores e assim extrair uma variedade de consultas de uma determinada tabela.

OPERADOR	DESCRIÇÃO
>	Maior que
<	Menor que
=	Igual
<> ou !=	Diferente de
>=	Maior ou igual a
<=	Menor ou igual a
<=>	Igual para NULL

OPERADORES LÓGICOS

- Existem ocasiões em que é necessário trabalhar com o relacionamento de **duas ou mais condições ao mesmo tempo**, e realizar testes múltiplos para obter consultas mais elaboradas.

OPERADORES LÓGICOS

- Para estes casos é necessário trabalhar com operadores lógicos:

OPERADOR	DESCRIÇÃO
AND ou &&	Operador de conjunção
OR ou	Operador de disjunção
NOT ou !	Operador de negação
XOR	Operador de disjunção exclusiva

OPERADORES LÓGICOS

- Apresentar uma consulta dos funcionários que sejam programadores e estejam locados no departamento 3:

```
SELECT *  
  
FROM funcionario  
  
WHERE (departamento = '3')  
  
AND (funcao = 'programador');
```


OPERADORES LÓGICOS

- Apresentar uma consulta dos funcionários que estejam locados no departamento 3 ou departamento 5:

```
SELECT *
```

```
FROM funcionario
```

```
WHERE (departamento = '3')
```

```
OR (departamento = '5');
```

OPERADORES LÓGICOS

- Apresentar uma consulta dos funcionários que não sejam vendedores:

```
SELECT *
```

```
FROM funcionario
```

```
WHERE NOT (funcao = 'vendedor');
```

XOR – DISJUNÇÃO EXCLUSIVA

- Apresentação dos funcionários do departamento 5 que não ocupam o cargo de programador e dos funcionários que ocupam o cargo de programador em qualquer outro departamento, que não seja o departamento 5:

```
SELECT *  
  
FROM funcionario  
  
WHERE (departamento = '5')  
  
XOR (funcao = 'programador');
```

Tabela Verdade – Operador XOR		
Condição 1	Condição 2	Resultado
Falsa	Falsa	Falso
Verdadeira	Falsa	Verdadeiro
Falsa	Verdadeira	Verdadeiro
Verdadeira	Verdadeira	Falso

OPERADORES AUXILIARES

- A linguagem SQL possui, além dos operadores aritméticos, relacionais e lógicos, alguns outros **operadores auxiliares** que facilitam a definição de condições a serem utilizadas com os comandos SELECT, UPDATE e DELETE, quando o WHERE for utilizado.

OPERADORES AUXILIARES

- A linguagem SQL possui, além dos operadores aritméticos, relacionais e lógicos, alguns outros **operadores auxiliares** que facilitam a definição de condições a serem utilizadas com os comandos SELECT, UPDATE e DELETE, quando o WHERE for utilizado.

OPERADORES AUXILIARES

OPERADOR	DESCRIÇÃO
IS NULL	Verifica se um campo é vazio
BETWEEN	Verifica um valor em uma faixa
IN	Verifica se um valor existe na tabela
LIKE	Verifica um valor buscando seus semelhantes

OPERADORES AUXILIARES | IS NULL

- O operador auxiliar **IS NULL** é utilizado quando há necessidade de verificar se um determinado campo da tabela possui registrado o valor **NULL**, ou seja, se um determinado campo está vazio.

OPERADORES AUXILIARES | IS NULL

- Agora será feita a inserção de um novo campo na tabela funcionario para registrar o número de filhos.

ALTER TABLE funcionario ADD filhos SMALLINT.

OPERADORES AUXILIARES | IS NULL

- Antes de testar o operador auxiliar **IS NULL**, é necessário definir para alguns registros de funcionários o número de filhos.

```
UPDATE funcionario SET filhos = 1 WHERE id = 2;
```

```
UPDATE funcionario SET filhos = 3 WHERE id = 3;
```

```
UPDATE funcionario SET filhos = 2 WHERE id = 5;
```

```
UPDATE funcionario SET filhos = 1 WHERE id = 9;
```

```
UPDATE funcionario SET filhos = 4 WHERE id = 20;
```

```
UPDATE funcionario SET filhos = 3 WHERE id = 25;
```

OPERADORES AUXILIARES | IS NULL

- A partir desse momento já possível usar o operador auxiliar **IS NULL** para visualizar todos os registros que estejam com o campo filho sinalizado como **NULL**:

```
SELECT id, nome, filhos FROM funcionario;
```

```
SELECT id, nome, filhos FROM func... WHERE filhos IS NULL;
```

```
SELECT id, nome, filhos FROM func... WHERE NOT filhos IS NULL;
```

OPERADORES AUXILIARES | BETWEEN

- O operador auxiliar **BETWEEN** permite definir uma faixa de valores para cláusula **WHERE** em uma condição.
- Considere a necessidade de obter-se uma consulta de todos os registros de funcionários que recebem salários entre 1700 e 2000:

```
SELECT * FROM funcionario
```

```
WHERE salario BETWEEN 1700 AND 2000;
```

OPERADORES AUXILIARES | BETWEEN

- Considere a necessidade de apresentar os funcionários que ganham abaixo de 1700 e acima de 2000:

```
SELECT * FROM funcionario
```

```
WHERE salario NOT BETWEEN 1700 AND 2000;
```

OPERADORES AUXILIARES | IN

- O operador auxiliar **IN** possibilita fazer a busca de um valor específico dentro de uma lista de valores definidos, retornado **TRUE** caso o valor esteja na lista.

OPERADORES AUXILIARES | IN

- Apresentar uma consulta de registros cujo campo departamento possua os valores 2 ou 3:

```
SELECT *
```

```
FROM funcionario
```

```
WHERE departamento IN ('2', '3');
```

Negação:

```
WHERE departamento NOT IN ('2', '3');
```

OPERADORES AUXILIARES | LIKE

- O operador **LIKE** é usado para verificar e comparar sequências de caracteres dentro de um determinado campo, sendo utilizado com a cláusula **WHERE**.

OPERADORES AUXILIARES | LIKE

- Aceita o uso de operadores curingas, aumentando assim sua capacidade.
- São aceitos como curingas os símbolos de
porcentagem (%) que representam zero, um ou vários caracteres, e
underline (_) que representa sempre um único caractere.

OPERADORES AUXILIARES | LIKE

Referência	Operação
WHERE salario LIKE '11%'	Encontra valores que começam com 11.
WHERE salario LIKE '%8%'	Valores que tenham 8 em qualquer posição.
WHERE salario LIKE '_0%'	Valores que tenham 0 na segunda posição.
WHERE salario LIKE '1_%_ %'	Valores que começam com 1 e tenham 3 caracteres de comprimento.
WHERE salario LIKE '%6'	Qualquer valor que termine com 6.
WHERE salario LIKE '_1%6'	Qualquer valor que tenha 1 na segunda posição e que termine com 6.

OPERADORES AUXILIARES | LIKE

- Consultar os funcionários cujo nome comece com o caractere A:

```
SELECT * FROM funcionario WHERE nome LIKE 'A%';
```