





Introdução

Vamos aprender neste capítulo os operadores aritméticos, relacionais e lógicos de uma forma um pouco mais detalhada e alguns operadores auxiliares.

Continuaremos trabalhando com nosso Banco de Dados de estudo.

Portanto, inicie o *MySQL Command Line Client* e ative o BD *empresa*.



3



Operadores Aritméticos

É possível realizar operações e cálculos matemáticos simples dentro de uma instrução SQL. Veja as operações possíveis:

Operação	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto da Divisão	%





SELECT com Op. Aritméticos

É possível realizar uma operação matemática na consulta de uma informação. Vamos testar de acordo com os passos abaixo:

1º - Consulta comum:

SELECT NOME, SALARIO FROM funcionario;

- 2º Observe os salários que estão armazenados nesta tabela.
- 3º Agora experimente este comando:

SELECT NOME, SALARIO + 100 FROM funcionario;

- 4º Compare os salários exibidos, com os do 1º passo realizado.
- 5º Agora repita o 1º passo e tire suas conclusões.
- 6º Faça agora uma consulta onde os salários sejam exibidos com 20% de reajuste (somente em um comando)



5



SELECT com Op. Aritméticos (Cont.)

Você também pode realizar simples operações matemáticas, independente do que existir armazenado nas tabelas.

Ex:

```
SELECT 2 + 3 * 5;

SELECT (2 + 3) * 5;

SELECT 3 / 2 + 4;

SELECT 3 % 2;
```



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







SELECT com Op. de Comparação

Faça alguns testes:

```
SELECT * FROM funcionario WHERE DEPTO = `5';

SELECT * FROM funcionario WHERE FUNCAO = `VENDEDOR';

SELECT * FROM funcionario WHERE SALARIO <= 1700;

SELECT * FROM funcionario WHERE SALARIO > 1700 + 50;
```





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. Para estes casos é necessário trabalhar com operadores lógicos, também conhecidos como operadores booleanos. Um operador lógico pode ser:

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



9



Operador Lógico AND ou &&

O operador **AND** ou **&&** é utilizado quando dois ou mais relacionamentos lógicos de uma determinada condição necessitam ser verdadeiros. Em seguida, é apresentada a tabela verdade para esse tipo de operador:

Tabela Verdade - Operador AND ou &&		
Condição 1	Condição 2	Resultado
FALSA	FALSA	FALSO
VERDADEIRA	FALSA	FALSO
FALSA	VERDADEIRA	FALSO
VERDADEIRA	VERDADEIRA	VERDADEIRO

Ex:

SELECT * FROM funcionario WHERE (DEPTO='5')
AND (FUNCAO='VENDEDOR');





Operador Lógico OR ou | |

O operador **OR** ou | | é utilizado quando pelo menos um dos relacionamentos lógicos (quando houver mais de um relacionamento) de uma condição necessita ser verdadeiro:

Tabela Verdade - Operador OR ou		
Condição 1	Condição 2	Resultado
FALSA	FALSA	FALSO
VERDADEIRA	FALSA	VERDADEIRO
FALSA	VERDADEIRA	VERDADEIRO
VERDADEIRA	VERDADEIRA	VERDADEIRO

Ex:

SELECT * FROM funcionario WHERE (DEPTO=\5') OR
(FUNCAO=\VENDEDOR');



11



Operador Lógico NOT ou!

O operador **NOT** ou **!** é utilizado quando é preciso estabelecer que uma determinada condição deve não ser falsa. O operador **NOT** inverte o estado lógico de uma condição.

Tabela Verdade - Operador NOT ou!	
Condição	Resultado
VERDADEIRA	FALSO
FALSA	VERDADEIRO

Ex:

SELECT * FROM funcionario WHERE
NOT (FUNCAO = VENDEDOR');





Operador Lógico XOR

O operador lógico de disjunção exclusiva **XOR** é utilizado quando se necessita que as condições sejam opostas para resultado lógico verdadeiro:

Tabela Verdade - Operador XOR		
Condição 1	Condição 2	Resultado
FALSA	FALSA	FALSO
VERDADEIRA	FALSA	VERDADEIRO
FALSA	VERDADEIRA	VERDADEIRO
VERDADEIRA	VERDADEIRA	FALSO

Ex:

SELECT * FROM funcionario WHERE
DEPTO=\5') XOR (FUNCAO=\SUPERVISOR');

Será exibida uma lista com funcionários do departamento 5 que não são supervisores, e dos funcionários supervisores que não são do departamento 5.



13



Operadores Auxiliares

A linguagem de consulta estruturada SQL possui alguns outros operadores auxiliares que facilitam a definição de condições a serem utilizadas com os comandos **SELECT**, **UPDATE** e **DELETE**, quando for utilizada a **WHERE**:

Operador	Descrição
IS NULL	Verifica se um campo é vazio
BETWEEN	Verifica um valor numa faixa de valores
IN	Verifica se um valor existe na tabela
LIKE	Verifica um valor buscando seus semelhantes





Operador Auxiliar IS NULL

Este operador 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.

Vamos testar, primeiramente alterando as tabelas, propositalmente:

ALTER TABLE funcionario ADD FILHOS SMALLINT; ALTER TABLE morto ADD FILHOS SMALLINT;

Agora faça uma consulta de todos os registros na tabela funcionario. Percebe-se que a coluna FILHOS foi criada, e está toda NULL.

Esta tarefa continua no próximo slide



15

Operador Auxiliar IS NULL (Cont.)

Continuação da tarefa do slide anterior:

Agora insira alguns registros de FILHOS de funcionários:

UPDATE funcionario SET FILHOS = 1 WHERE CODFUN = 3; UPDATE funcionario SET FILHOS = 3 WHERE CODFUN = 10;

Faça novamente uma consulta de todos os registros da tabela funcionario.

Agora sim, vamos testar o operador IS NULL:

SELECT CODFUN, NOME, FILHOS FROM funcionario WHERE FILHOS IS NULL;

Faça também este teste:

SELECT CODFUN, NOME, FILHOS FROM funcionario WHERE NOT FILHOS IS NULL;





Operador Auxiliar BETWEEN

O operador auxiliar **BETWEEN** permite definir uma faixa de valores para a cláusula **WHERE** em uma condição. Desta forma, é possível extrair consultas de registros situados em uma faixa de valores.

Considere a necessidade de obter-se uma consulta de todos os registros de funcionários que recebem salários entre 1700 e 2000.

Assim sendo, utilize a instrução:

SELECT * FROM funcionario WHERE SALARIO BETWEEN 1700 AND 2000;

Experimente também o comando abaixo a analise a diferença com o anterior:

SELECT * FROM funcionario WHERE SALARIO NOT BETWEEN 1700 AND 2000;

17





Operador Auxiliar IN

O operador auxiliar **IN** possibilita fazer a busca de um valor específico dentro de uma lista de valores definidos, retornando TRUE caso o valor específico esteja na lista. Os valores da lista devem ser indicados entre apóstrofos.

Por exemplo, apresentar uma consulta de registros cujo campo DEPTO possua valores 2, 3 ou 5. Utilize a seguinte instrução:

```
SELECT * FROM funcionario
WHERE DEPTO IN ('2', '3', '5');
```

Agora uma busca com a negação:

SELECT * FROM funcionario
WHERE DEPTO NOT IN ('2', '3', '5');





Operador Auxiliar LIKE

O operador auxiliar **LIKE** aceita o uso de operadores curingas, aumentando assim sua capacidade de operação. Neste sentido, são aceitos como operadores curingas os símbolos de porcentagem (%) que representam zero, um ou vários caracteres, e underline (_) que representa sempre um único caractere. Desta forma, seguem alguns exemplos como referências:

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



19



Operador Auxiliar LIKE (Cont.)

O exemplo a seguir faz a consulta dos registros de todos os funcionários cujo nome comece com a caractere A. Para tanto, execute a instrução:

SELECT * FROM funcionario WHERE NOME LIKE 'A%';

O próximo exemplo apresenta os registros de todos os funcionários que possuam a sequência de caracteres AN em qualquer posição de seu nome. Execute a instrução:

SELECT * FROM funcionario WHERE NOME LIKE '%AN%';

E agora experimente este comando e tire suas conclusões:

SELECT * FROM funcionario WHERE NOME NOT LIKE '%AN%';





