

Tutorial MySQL – Parte III

Aula 10: MySQL – AND, OR e NOT – Filtrar resultados de consultas

- Os operadores AND, OR e NOT são usados para filtrar registros baseados em mais de uma condição.
- O operador AND mostra um registro se ambas as condições forem verdadeiras.
- O operador OR mostra um registro se pelo menos uma das condições for verdadeira.
- O operador NOT é a negação de uma expressão (inverte seu estado lógico).

Exemplos de uso:

1 – Retornar todas as colunas da tabela de livros com os dados de livros de ID maior que 2 e ID de autor menor que 3, ao mesmo tempo:

```
SELECT * FROM tbl_Livro
WHERE ID_Livro > 2 AND ID_Autor < 3;
```

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora
3	Using Samba	123856789	2	2000-12-21	61.45	2

2 – Trazer todos os dados da tabela de livros de livros cujo ID é maior que 2 ou cujo ID do autor seja menor do que 3:

```
SELECT * FROM tbl_Livro
WHERE ID_Livro > 2 OR ID_Autor < 3;
```

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora
2	SSH, the Secure Shell	127658789	1	2009-12-21	58.30	2
3	Using Samba	123856789	2	2000-12-21	61.45	2
4	Fedora and Red Hat Linux	123346789	3	2010-11-01	62.24	1
5	Windows Server 2012 Inside Out	123356789	4	2004-05-17	66.80	3
6	Microsoft Exchange Server 2010	123366789	4	2000-12-21	45.30	3

Retornar todos os registros da tabela de livros cujo ID do livro seja maior do que 2 e o ID do autor **não** seja menor do que 3:

```
SELECT * FROM tbl_Livro
WHERE ID_Livro > 2 AND NOT ID_Autor < 3;
```

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora
4	Fedora and Red Hat Linux	123346789	3	2010-11-01	62.24	1
5	Windows Server 2012 Inside Out	123356789	4	2004-05-17	66.80	3
6	Microsoft Exchange Server 2010	123366789	4	2000-12-21	45.30	3

Aula 11: MySQL – DELETE e TRUNCATE TABLE – Excluir registros de uma tabela

Uma das tarefas mais comuns na manutenção de tabelas de bancos de dados é a exclusão de registros (linhas). É muito comum, por exemplo, excluir um produto de uma tabela que não é mais ofertado por uma loja, ou um cliente que se descadastrou de um sistema.

Além disso, em algumas situações muito especiais, pode ser necessário excluir todos os registros de uma tabela – ou seja, limpar a tabela.

Podemos excluir registros de uma tabela com SQL por meio de duas declarações: DELETE FROM e TRUNCATE TABLE.

Excluir um ou mais registros especificados – Cláusula DELETE

Com a cláusula DELETE podemos excluir registros específicos, indicados (filtrados) por meio de uma condição na cláusula WHERE

Sintaxe:

```
DELETE FROM tabela WHERE coluna = valor;
```

Exemplo – vamos excluir o autor da tabela de autores cujo id de autor é igual a 2:

```
DELETE FROM tbl_autores  
WHERE ID_Autor = 2;
```

Obs.: Sempre use a cláusula WHERE para evitar a perda de dados da tabela, caso contrário todos os registros serão excluídos, uma a uma!

TRUNCATE TABLE

O comando TRUNCATE TABLE permite remover todas as linhas de uma tabela em uma única operação, sem registrar as exclusões de linhas individuais.

O TRUNCATE TABLE equivale a executar a instrução DELETE, porém sem usar a cláusula WHERE. Portanto, é usada para apagar completamente o conteúdo de uma tabela no MySQL.

Entretanto, a cláusula TRUNCATE TABLE é mais rápida e utiliza menos recursos de sistema e log de transações durante sua execução.

Exemplo

Exemplo: Vamos excluir todos os registros presentes na tabela *tbl_teste_incremento*:

```
TRUNCATE TABLE tbl_teste_incremento;
```

Aula 12: MySQL – Alias com AS – Nomes alternativos para colunas e tabelas

Pode-se dar um nome diferente (e mais amigável) a uma coluna ou tabela ao realizar uma junção (join) ou retornar o resultado de uma consulta, de modo que seja mais fácil ou intuitivo entender os dados retornados. Para isso, usamos a cláusula **AS**.

Sintaxe de alias para colunas:

```
SELECT coluna  
AS alias_coluna  
FROM tabela AS alias_tabela;
```

Exemplo – Retornar a coluna Nome_Livro com o nome de Livro, simplesmente:

```
SELECT Nome_Livro  
AS Livro  
FROM tbl_Livro;
```

Podemos também aplicar um alias a uma coluna sem a necessidade de usar a palavra **AS**, bastando para isso inserir o alias desejado logo após o nome da coluna, sem separação por vírgulas. Veja o exemplo a seguir:

```
SELECT Nome_Livro Livro  
FROM tbl_Livro;
```

Para aplicar alias em mais de uma coluna, basta acrescentá-las normalmente, separando-as por vírgulas, e incluindo os alias logo após o nome de cada coluna.

Além disso, podemos criar alias usando palavras compostas, incluindo espaços, bastando para isso envolver o alias entre aspas. O exemplo a seguir mostra as duas possibilidades juntas:

```
SELECT Nome_Livro Livro, Preco_Livro 'Preço do Livro'  
FROM tbl_Livro;
```

Recomenda-se também usar aspas em aliases que contêm caracteres especiais ou que precisem respeitar diferenciação entre maiúsculas e minúsculas (case-sensitive).

Aula 13: MySQL – Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

Funções de agregação são funções SQL que permitem executar uma operação aritmética nos valores de uma coluna em todos os registros de uma tabela.

Retornam um valor simples baseado em um conjunto de valores de entrada.

Sintaxe básica:

```
função(ALL | DISTINCT expressão)
```

ALL – avalia todos os registros ao agregar o valor da função; é o comportamento padrão.

DISTINCT – Usa apenas valores distintos (sem repetição) ao avaliar a função.

As funções de agregação desconsideram valores **NULL** (com exceção da função COUNT(*)).

As principais funções de agregação (mais comuns) em MySQL são as seguintes:

- MIN = Valor Mínimo de um conjunto de valores
- MAX = Valor Máximo de um conjunto de valores
- AVG = Média Aritmética de um conjunto de valores
- SUM = Total (Soma) de um conjunto de valores
- COUNT = Contar quantidade total de itens

As funções SUM e AVG somente aceitam como entrada um conjunto de números; já as demais funções podem operar também com outros tipos de dados não-numéricos, como por exemplo strings (caracteres) ou datas.

Vamos a alguns exemplos usando o banco de dados db_biblioteca.

Exemplos

1 – Retornar o número total de autores cadastrados na tabela de autores:

```
SELECT COUNT(*) FROM tbl_autores;
```

2 – Contar o número de autores que possuem livros cadastrados na tabela de autores, sem repetições

```
SELECT COUNT(DISTINCT id_autor) FROM tbl_Livro;
```

3 – Descobrir o preço mais alto dos livros:

```
SELECT MAX(Preco_Livro) FROM tbl_Livro;
```

4 – Descobrir a data de publicação do livro mais antigo:

```
SELECT MIN(Data_Pub) FROM tbl_Livro;
```

5 – Retornar o preço médio dos livros cadastrados no banco:

```
SELECT AVG(Preco_Livro) FROM tbl_Livro;
```

6 – Descobrir o valor total dos livros presentes na tabela de livros:

```
SELECT SUM(Preco_Livro) FROM tbl_Livro;
```

É muito comum usar as funções agregadas em conjunto com a **cláusula GROUP BY**, aplicando-as a grupos de dados específicos.

Aula 14: MySQL – BETWEEN – Seleção de intervalos em consultas

Vamos aprender nesta aula a aprimorar nossas consultas em um banco de dados MySQL usando a cláusula BETWEEN, que nos permite selecionar intervalos de dados ao retornar os resultados de uma consulta.

Podemos usar a cláusula BETWEEN para, por exemplo, retornar registros cujos preços estejam entre dois valores distintos, ou registros contidos dentro de um intervalo de datas especificado.

A sintaxe para uso da cláusula BETWEEN é a seguinte:

SELECT *colunas* **FROM** *tabela*

WHERE *coluna* **BETWEEN** *valor1* **AND** *valor2*;

Usamos o operador lógico AND para auxiliar na criação do código de consulta.

Vejamos alguns exemplos.

Exemplo 1: Vamos retornar todos os livros da tabela tbl_livro cuja data de publicação esteja entre 17/05/2004 e 17/05/2011 (note como a data é fornecida no código: ano|mês|dia):

```
SELECT * FROM tbl_Livro  
WHERE Data_Pub BETWEEN '20040517' AND '20110517';
```

Resultado:

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora
1	Linux Command Line and Shell Scripting	143856969	5	2009-12-21	68.35	4
2	SSH, the Secure Shell	127658789	1	2009-12-21	58.30	2
4	Fedora and Red Hat Linux	123346789	3	2010-11-01	62.24	1
5	Windows Server 2012 Inside Out	123356789	4	2004-05-17	66.80	3

Exemplo 2: Agora vamos retornar os nomes dos livros e seus respectivos preços, da tabela tbl_livros, porém somente os livros cujos preços estiverem entre R\$ 40,00 e 60,00:

```
SELECT Nome_Livro AS Livro, Preco_Livro AS Preço  
FROM tbl_Livro  
WHERE Preco_Livro BETWEEN 40.00 AND 60.00;
```

Resultado:

Livro	Preço
SSH, the Secure Shell	58.30
Microsoft Exchange Server 2010	45.30

Aula 15: MySQL – LIKE e NOT LIKE – Padrões de caracteres em consultas

Quando realizamos uma consulta no MySQL, utilizamos a cláusula WHERE para realizar um filtro dos registros a retornar. Porém, com o WHERE, só podemos aplicar filtros de correspondência exata de palavras. E se precisarmos aplicar um filtro que verifique palavras de forma parcial, como palavras que iniciem ou terminem com determinados caracteres, ou que possuam sequências de caracteres específicas? Neste caso, usamos a cláusula LIKE:

- A cláusula LIKE determina se uma cadeia de caracteres (string) corresponde a um padrão especificado. Um padrão pode incluir caracteres normais e curingas.
- NOT LIKE inverte a comparação, verificando se a cadeia de caracteres NÃO corresponde ao padrão especificado.

Padrões específicos (metacaracteres)

Usamos diversos conjuntos de caracteres para especificar os padrões a serem filtrados pelas cláusulas LIKE e NOT LIKE. Por exemplo:

'%' — Qualquer cadeia de 0 ou mais caracteres

'_' — Sublinhado: qualquer caracter único

Exemplos:

1. Selecionar todos os registros da tabela tbl_livro cujo nome comece com a letra F:

```
SELECT * FROM tbl_Livro
WHERE Nome_Livro LIKE 'F%';
```

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora
4	Fedora and Red Hat Linux	123346789	3	2010-11-01	62.24	1

2. Selecionar todos os registros da tabela tbl_livro cujo nome não começa com a letra S:

```
SELECT * FROM tbl_Livro
WHERE Nome_Livro NOT LIKE 'S%';
```

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora
1	Linux Command Line and Shell Scripting	143856969	5	2009-12-21	68.35	4
3	Using Samba	123856789	2	2000-12-21	61.45	2
4	Fedora and Red Hat Linux	123346789	3	2010-11-01	62.24	1
5	Windows Server 2012 Inside Out	123356789	4	2004-05-17	66.80	3
6	Microsoft Exchange Server 2010	123366789	4	2000-12-21	45.30	3
7	Enciclopédia de Componentes Eletrônicos vol. 03	153642397	13	2016-05-05	63.39	5

3. Selecionar os nomes de livros da tabela tbl_livro cujo nome se inicie com uma letra qualquer e a segunda letra seja a letra i:

```
SELECT Nome_Livro
FROM tbl_Livro
WHERE Nome_Livro LIKE '_i%';
```

Nome_Livro
Linux Command Line and Shell Scripting
Windows Server 2012 Inside Out
Microsoft Exchange Server 2010

4. Selecionar os nomes dos livros e seus respectivos preços, na tabela de livros, cujo nome não comece com a letra F e que custem mais de R\$ 60,00:

```
SELECT Nome_Livro AS Livro, Preco_Livro AS Valor  
FROM tbl_livro  
WHERE Nome_Livro NOT LIKE 'F%'  
AND Preco_Livro > 60.00;
```

Livro	Valor
Linux Command Line and Shell Scripting	68.35
Using Samba	61.45
Windows Server 2012 Inside Out	66.80
Enciclopédia de Componentes Eletrônicos vol. 03	63.39