



Selecionando Registros

1

Banco de Dados II



Banco de Dados II

Selecionando Registros

Consultas Simples em MySQL

Podemos dizer que a tarefa mais importante que um banco de dados MySQL realiza, do ponto de vista do usuário, é permitir a realização de consultas aos dados armazenados. O comando básico que utilizamos para realizar as consultas é o comando `SELECT`.





Banco de Dados II

Selecionando Registros

Consultas Simples em MySQL

Comando SELECT

Sintaxe:

```
SELECT coluna(s)  
FROM Tabela;
```





Banco de Dados II

Selecionando Registros

Consultas Simples em MySQL

Comando SELECT

Exemplos:

Consultar nomes dos autores cadastrados na tabela de autores:

```
SELECT Nome_Autor  
FROM tbl_Autores;
```





Banco de Dados II

Selecionando Registros

Consultas Simples em MySQL

Comando SELECT

Exemplos:

Consultar os nomes dos livros (apenas) na tabela de livros:

```
SELECT Nome_Livro  
FROM tbl_livro;
```

Consultar nomes de livros e IDs de autores respectivos na tabela de livros:

```
SELECT Nome_Livro, ID_Autor  
FROM tbl_livro;
```





Banco de Dados II

Selecionando Registros

Consultas Simples em MySQL

Comando SELECT

Exemplos:

Consultar os nomes dos livros (apenas) na tabela de livros:

```
SELECT Nome_Livro  
FROM tbl_livro;
```

Consultar nomes de livros e IDs de autores respectivos na tabela de livros:

```
SELECT Nome_Livro, ID_Autor  
FROM tbl_livro;
```





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE

A cláusula WHERE permite filtrar registros nos resultados de uma consulta, de modo a trazer apenas as informações desejadas (e não o conteúdo completo das colunas).

```
SELECT colunas  
FROM tabela  
WHERE coluna = valor;
```





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE

Retornar o nome e a data de publicação do livro cujo ID do autor é 1:

```
SELECT Nome_Livro, Data_Pub  
FROM tbl_Livro  
WHERE ID_Autor = 1;
```

Nome_Livro	Data_Pub
SSH, the Secure Shell	2009-12-21

Trazer o ID e Nome do autor do autor cujo sobrenome é Stanek:

```
SELECT ID_Autor, Nome_Autor  
FROM tbl_autores  
WHERE Sobrenome_Autor = 'Stanek';
```

ID_Autor	Nome_Autor
4	William





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores AND, OR e NOT

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).





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores **AND**, OR e NOT

Exemplos de uso

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores AND, **OR** e NOT

Exemplos de uso

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores AND, OR e **NOT**

Exemplos de uso

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores **IN** e **NOT IN**

O operador SQL **IN** permite verificar se um valor específico corresponde a algum valor presente em uma lista ou subconsulta, passada como parâmetro a uma cláusula de filtragem WHERE.

```
SELECT coluna(s)  
FROM tabela  
WHERE expressão | coluna  
IN (valor1, valor2, ...);
```

O operador **IN** retorna o valor 1 (true) se o valor da expressão ou da coluna na cláusula WHERE corresponder a qualquer um dos valores presentes na lista passada, e retorna 0 (false) se não houver nenhuma correspondência.





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores **IN** e **NOT IN**

Retornar livros cujas editoras tem os códigos 2 ou 4:

```
SELECT NomeLivro, IdEditora  
FROM tbl_livro  
WHERE IdEditora IN (2,4);
```





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores IN e **NOT IN**

Podemos combinar o operador **IN** com o operador lógico NOT SQL para determinar se um valor **NÃO** corresponde com nenhum valor na lista ou subconsulta realizada:

```
SELECT coluna(s)  
FROM tabela  
WHERE expressão | coluna  
NOT IN (valor1, valor2,...);
```





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - Operadores IN e **NOT IN**

Retornar livros cuja edição não é a primeira nem a segunda:

```
SELECT NomeLivro, Edicao  
FROM tbl_livro  
WHERE Edicao NOT IN (1,2);
```

NomeLivro	Edicao
Microsoft Exchange Server 2010	3
Practical Electronics for Inventors	4





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - **BETWEEN** - Seleção de intervalos em consultas

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.

```
SELECT colunas  
FROM tabela  
WHERE coluna  
        BETWEEN valor1 AND valor2;
```

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - **BETWEEN** - Seleção de intervalos em consultas

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';
```

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

Cláusula WHERE - **BETWEEN** - Seleção de intervalos em consultas

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;
```

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

Obs.: A Cláusula **AS** nomeia a coluna de saída do resultado consulta.





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

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:





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

LIKE e NOT LIKE - Padrões de caracteres em consultas

- 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.





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

LIKE e NOT LIKE – Padrões de caracteres em consultas

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

LIKE e NOT LIKE - Padrões de caracteres em consultas

Padrões específicos (metacaracteres)

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

LIKE e NOT LIKE - Padrões de caracteres em consultas

Padrões específicos (metacaracteres)

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

LIKE e NOT LIKE - Padrões de caracteres em consultas

Padrões específicos (metacaracteres)

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





Banco de Dados II

Selecionando Registros

Filtrar resultados de consultas

LIKE e NOT LIKE - Padrões de caracteres em consultas

Padrões específicos (metacaracteres)

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

```
SELECT Nome_Livro AS Livro, Preço_Livro AS Valor
FROM tbl_livro
WHERE Nome_Livro NOT LIKE 'F%'
AND Preço_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





Referências

- **Boson Treinamentos**

www.bosontreinamentos.com.br/mysql/mysql-select-realizar-consultas-simples-em-tabelas-12/

