

Banco de Dados

Rebeca Barros



Consultando dados nas tabelas

MySQL

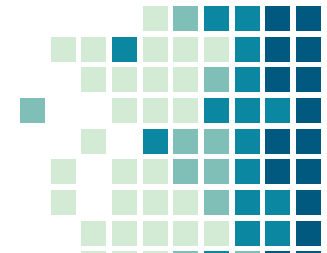


Seleção

- O comando usado para extrair informações de um banco de dados é o **SELECT**;
- É um comando com amplas possibilidades que permitem recuperar desde a simples extração do conteúdo de todas as linhas e colunas de uma tabela até a união de diversas tabelas, cálculos, agrupamentos, ordenações e filtragem das linhas e colunas.



Seleção



```
SELECT [DISTINCT | ALL] { * | nome_coluna [, nome_coluna, ...] }  
FROM nome_tabela
```

- Onde:
 - **DISTINCT** é usado para não mostrar valores repetidos;
 - **ALL** é usado para mostrar todos os valores. É o padrão se **DISTINCT** não for definido.
 - ***** indica que devem ser mostrada todas as colunas da tabela
 - **coluna1...colunan** são as colunas que serão recuperadas;
 - **tabela** é a tabela em que será realizada a busca.

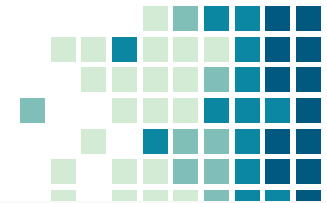


Seleção

recuperar todas as linhas e colunas da tabela CD
`SELECT * FROM CD;`

recuperar apenas id e nome da tabela CD
`SELECT idCD, nome FROM CD;`

Seleção – Ordenando o resultado



sintaxe do Select com ordenação

```
SELECT [DISTINCT | ALL] {*} | nome_coluna [, nome_coluna,...]  
FROM nome_tabela  
ORDER BY coluna1 [, coluna2,..]
```

- Onde:
 - **coluna1...colunan** é a lista de colunas na ordem de precedência de classificação.



Seleção – Ordenando o resultado

recuperar id e nome do cd em ordem alfabética
`SELECT idCD, nome FROM CD ORDER BY nome;`

recuperar id e nome do cd em ordem alfabética decrescente
`SELECT idCD, nome FROM CD ORDER BY nome DESC;`

Seleção – Ordenando o resultado

```
/* recuperar id da Gravadora e nome do cd em ordem  
de id e depois de nome */  
SELECT gravadora_idGravadora, nome FROM CD  
ORDER BY gravadora_idGravadora, nome;
```


Seleção – Filtrando linhas

```
-- sintaxe do Select com Where (filtrando linhas)
SELECT [DISTINCT | ALL] {*} | nome_coluna [, nome_coluna,...]}
FROM nome_tabela
WHERE condicao
```

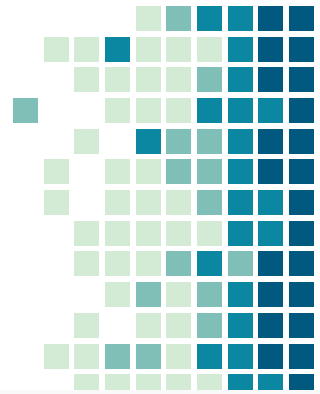
- Onde:
 - **condicao** é a expressão lógica que será avaliada para definir quais linhas irão aparecer no resultado.

Seleção – Filtrando linhas

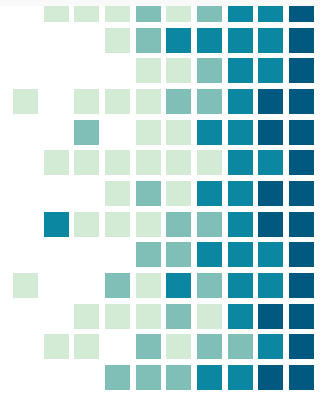
- **Operadores relacionais:** definem um tipo de condição básica. Devem ser colocados entre os argumentos que estão sendo comparados.

Operador	Significado	Exemplo
=	igual	idCompositor = 2
<	Menor que	preco_venda < 10
<=	Menor ou igual a	Preco_venda <=10
>	Maior que	preco_venda > 10
>=	Maior ou igual a	preco_venda >=10
!= ou <>	diferente	idCompositor !=2 ou idCompositor <> 2

Seleção – Filtrando linhas



```
/* recuperar id e nome apenas dos cds com  
preço de venda maior que 10. */  
select idCD, nome from cd where preco_venda > 10;
```



Seleção – Filtrando linhas

- **Operadores lógicos:** usados quando for necessário atender a mais de uma condição.

Operador	Significado	Exemplo
AND	e	condicao1 AND condicao2
OR	ou	condicao1 OR condicao2
NOT ou !	não/negação	NOT condicao

Seleção – Filtrando linhas

- **AND:** indica que as duas condições devem ser verdadeiras para que seja mostrada a linha

```
/* recuperar id e nome apenas dos cds com preço de venda maior que 10 e da gravadora com id=2. */  
select idCD, nome from cd where preco_venda > 10 AND  
gravadora_idGravadora = 2;
```

Seleção – Filtrando linhas

- **OR:** indica que pelo menos uma das condições deve ser verdadeira para que seja mostrada a linha

```
/* recuperar id e nome dos cds com preço de venda  
maior que 11 ou que sejam da gravadora com id=3. */  
select nome, preco_venda, gravadora_idGravadora from cd  
where preco_venda > 11 OR  
gravadora_idGravadora = 3;
```

Seleção – Filtrando linhas

- **NOT**: utilizado para inverter o resultado de uma expressão lógica, negando o resultado dessa condição. Isto é, se a condição for verdadeira, será retornado falso e vice-versa.

```
/* recuperar nome e preco de venda dos cds com preço de venda  
que não seja maior que 15 */  
select nome, preco_venda from cd  
where NOT (preco_venda < 15);
```

Seleção – Operadores Especiais

- **IS NULL**: usado quando deseja-se saber se o conteúdo da coluna foi preenchido (inicializado) ou não.

```
/* recuperar nome das gravadoras que não  
tiveram o endereço preenchido */  
select nomeGravadora from gravadora  
where enderecoGravadora IS NULL;
```


Seleção – Operadores Especiais

- **IS NOT NULL**: negação do IS NULL, usado quando deseja-se recuperar apenas linhas com conteúdo na coluna

```
/* recuperar nome das gravadoras que  
tiveram o endereço preenchido */  
select nomeGravadora from gravadora  
where enderecoGravadora IS NOT NULL;
```

Seleção – Operadores Especiais

- **BETWEEN**: usado para determinar um intervalo de busca.

```
/* recuperar nome e data de lançamento dos cds lançados entre  
01 de janeiro de 1999 e 01 de dezembro de 2001 */  
select nome, data_lancamento from cd  
where data_lancamento BETWEEN '1999-01-01' AND  
'2001-12-01';
```

Seleção – Operadores Especiais

- **BETWEEN**: usado para determinar um intervalo de busca.

```
/* recuperar nome e preço de venda dos cds cujo o  
preço não esteja entre 9 e 11 */  
select nome, preco_venda from cd  
where preco_venda NOT BETWEEN 9 AND 11;
```

Seleção – Operadores Especiais

- **LIKE:** usado para comparar cadeias de caracteres utilizando padrões de comparação (*wildcards*). Normalmente o percentual (%) substitui zero, um ou mais caracteres e o underline (_) substitui um caractere.

Seleção – Operadores Especiais

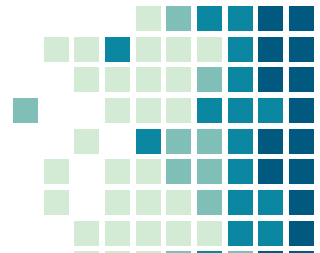
Expressão	Significado
LIKE 'A%'	Todas as palavras que iniciem com a letra A
LIKE '%A'	Todas as palavras que terminem com a letra A
LIKE '%A%'	Todas as palavras que tenham a letra A em qualquer posição
LIKE 'A_'	String de 2 caracteres onde o primeiro é a letra A
LIKE 'A_'	String de 2 caracteres onde o último é a letra A
LIKE '_A_'	String de 3 caracteres cuja segunda letra seja A
LIKE '%A_'	Todas as palavras que tenham letra A na penúltima posição
LIKE '_A%'	Todas as palavras que tenham letra A na segunda posição

Seleção – Operadores Especiais

```
/* recuperar nome dos compositores que comecem  
com a letra R */  
select * from compositor  
where nome LIKE 'R%';
```

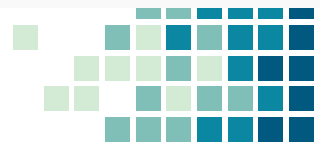
```
/* recuperar nome dos compositores que tenham l  
como segunda letra do nome */  
select * from compositor  
where nome LIKE '_l%';
```

Seleção – Operadores Especiais



```
/* recuperar nome dos compositores que comece  
com a letra C e tenha R como terceira letra do nome */  
select * from compositor  
where nome LIKE 'C_r%';
```

```
/* recuperar id, nome e preço dos cd que tenham  
preço começando com o algarismo 1 */  
select idCD, nome, preco_venda from cd  
where preco_venda LIKE '1%';
```



Seleção – Operadores Especiais

- **IN:** permite comparar o valor de uma coluna com um conjunto de valores. Normalmente usado para substituir uma série de comparações seguidas da cláusula OR.

```
/* recuperar os compositores que tenham o id  
1, 10 ou 20 */  
select * from compositor  
WHERE idCompositor IN (1, 10, 20);
```


Referências

- Beaulieu, A. (2009). *Learning SQL: Master SQL Fundamentals*. "O'Reilly Media, Inc."
- de OLIVEIRA, C. H. P. (2002). *SQL: curso prático*. Novatec.
- Documentação oficial do MySQL. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/>
- Curso Completo de MySQL (Bóson Treinamentos). Disponível em: <http://www.bosontreinamentos.com.br/curso-completo-de-mysql/>