

SELECT – DISTINCT – WHERE
– BETWEEN – LIKE – IN –
AND – OR – NOT – ORDER BY

Prof. Ewerton

SELECT

O comando SELECT é usado para selecionar dados de um banco de dados. Os dados retornados são armazenados em uma tabela de resultados, chamada de conjunto de resultados. A sintaxe básica do comando SELECT é:

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela;
```

Aqui, coluna1, coluna2, ... são os nomes dos campos da tabela que você deseja selecionar os dados.

DISTINCT

O comando `SELECT DISTINCT` é usado para retornar apenas valores distintos (diferentes) em uma coluna. Dentro de uma tabela, uma coluna geralmente contém muitos valores duplicados e às vezes você só deseja listar os valores diferentes (distintos). A sintaxe do `SELECT DISTINCT` é a seguinte:

```
SELECT DISTINCT nome_da_coluna FROM nome_da_tabela;
```

Você pode especificar várias colunas separando-as por vírgulas após o `SELECT DISTINCT`. Por exemplo:

```
SELECT DISTINCT nome_da_coluna1, nome_da_coluna2 FROM  
nome_da_tabela;
```

WHERE

A cláusula WHERE é usada para filtrar resultados em uma consulta SQL. Ela é usada para especificar uma condição que deve ser atendida para que as linhas sejam retornadas em uma consulta. Por exemplo:

```
SELECT nome_da_coluna1, nome_da_coluna2  
FROM nome_da_tabela  
WHERE nome_da_coluna1 = 'valor';
```

Neste exemplo, apenas as linhas em que o valor da coluna “nome_da_coluna1” é igual a “valor” serão retornadas.

A cláusula WHERE não é usada apenas em instruções SELECT, ela também é usada em UPDATE, DELETE, etc.!

SQL requer aspas simples em torno de valores de texto (a maioria dos sistemas de banco de dados também permite aspas duplas). No entanto, os campos numéricos não devem ser colocados entre aspas.

Com WHERE você pode usar qualquer operador de comparação (como =, <>, >, <, >=, <=) e operadores lógicos (como AND, OR e NOT) para criar condições de filtro.

Operadores na cláusula WHERE

=	Igual
>	Maior que
<	Menor que
>=	Maior ou igual
<=	Menor ou igual
<>	Não igual. Observação: em algumas versões do SQL, esse operador pode ser escrito como !=
BETWEEN	Entre um determinado intervalo
LIKE	Pesquisar um padrão
IN	Para especificar vários valores possíveis para uma coluna

BETWEEN

A cláusula BETWEEN é usada para selecionar valores dentro de um intervalo em uma consulta no MySQL. A sintaxe é a seguinte:

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

Esta consulta retorna todas as linhas da tabela “tabela” onde o valor da coluna “coluna” está entre “valor1” e “valor2”, inclusive.

Aqui está um exemplo:

```
SELECT * FROM times WHERE pontos BETWEEN 10 AND 20;
```

Esta consulta retorna todas as linhas da tabela “times” onde o valor da coluna “pontos” está entre 10 e 20, inclusive.

LIKE

A cláusula LIKE é usada para fazer buscas parciais em campos do tipo texto (varchar, text etc) utilizando coringas % e _. A sintaxe é a seguinte:

```
SELECT * FROM tabela WHERE coluna LIKE 'padrão';
```

Esta consulta retorna todas as linhas da tabela “tabela” onde o valor da coluna “coluna” corresponde ao padrão especificado.

Aqui está um exemplo:

```
SELECT * FROM clientes WHERE cliente_nome LIKE 'Luiz%';
```

Esta consulta retorna todas as linhas da tabela “clientes” onde o valor da coluna “cliente_nome” começa com “Luiz”.

IN

A cláusula IN é usada para verificar se um valor específico corresponde a algum valor presente em uma lista ou subconsulta passada como parâmetro a uma cláusula de filtragem WHERE. A sintaxe é a seguinte:

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

Esta consulta retorna todas as linhas da tabela “tabela” onde o valor da coluna “coluna” corresponde a qualquer um dos valores presentes na lista passada.

Aqui está um exemplo:

```
SELECT * FROM clientes WHERE cliente_id IN (1, 2, 3);
```

Esta consulta retorna todas as linhas da tabela “clientes” onde o valor da coluna “cliente_id” é igual a 1, 2 ou 3.

AND – OR – 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).

Aqui está um exemplo:

```
SELECT * FROM clientes WHERE cliente_id = 1 AND idade > 18;
```

Esta consulta retorna todas as linhas da tabela “clientes” onde o valor da coluna “cliente_id” é igual a 1 e o valor da coluna “idade” é maior que 18.

Exemplo de uso dos operadores AND e OR:

```
SELECT * FROM clientes WHERE cliente_id = 1 AND idade > 18 OR  
cidade = 'São Paulo';
```

Esta consulta retorna todas as linhas da tabela “clientes” onde o valor da coluna “cliente_id” é igual a 1 e o valor da coluna “idade” é maior que 18 ou o valor da coluna “cidade” é igual a “São Paulo”.

Aqui está um exemplo de uso do operador NOT:

```
SELECT * FROM clientes WHERE NOT cidade = 'São Paulo';
```

Esta consulta retorna todas as linhas da tabela “clientes” onde o valor da coluna “cidade” não é igual a “São Paulo”.

ORDER BY

A cláusula ORDER BY é usada para classificar os resultados de uma consulta em ordem crescente ou decrescente. Por padrão, os resultados são classificados em ordem crescente. Para classificar em ordem decrescente, use a palavra-chave DESC.

Aqui está um exemplo:

```
SELECT * FROM clientes ORDER BY nome DESC;
```

Esta consulta retorna todas as linhas da tabela “clientes” ordenadas pelo valor da coluna “nome” em ordem decrescente.

```
SELECT * FROM jogadores WHERE tme_id = 1 OR tme_id = 3 ORDER BY tme_id, jog_posicao, jog_nome;
```

```
SELECT * FROM jogadores WHERE NOT tme_id = 5 ORDER BY tme_id, jog_posicao DESC, jog_nome;
```

```
SELECT * FROM jogadores WHERE tme_id = 1 AND (jog_posicao = 'ATA' OR jog_posicao = 'MEI');
```