



SQL (DML)

Cláusulas especiais

A cláusula **CASE** **percorre condições e retorna um valor quando a primeira condição é atendida**. Assim, uma vez que uma condição seja verdadeira, a expressão para de avaliar as demais e retorna o resultado. Se nenhuma condição for verdadeira, ela retorna o valor especificado na cláusula **ELSE**. Caso não haja uma parte **ELSE** e nenhuma condição seja verdadeira, ela retornará **NULL**.

A sintaxe da cláusula básica da cláusula CASE é:

CASE

WHEN condição1 **THEN** resultado1

WHEN condição2 **THEN** resultado2

WHEN condiçãoN **THEN** resultadoN

ELSE resultado

END;

EXEMPLIFICANDO!!!

Dada a tabela Produtos a seguir:

ProdutoID	Nome_do_Produto	FornecedorID	CategoriaID	Unidade	Preco
1	Leite	1	1	Litros	3
2	Banana	1	1	Kilogramas	5
3	Melancia	1	2	Unidade	6
4	Pão	2	2	Pacote	4

Ao executar o seguinte comando:

SELECT ProdutoID, Preco

CASE WHEN Preco > 4 **THEN** 'O preço é MAIOR que 4'

WHEN Preco = 4 **THEN** 'O preço é 4'

ELSE 'O preço é MENOR que 4'

END AS QuantidadeEmTexto

FROM Produto;

ProdutoID	Preco	QuantidadeEmTexto
1	3	O preço é MENOR que 4
2	5	O preço é MAIOR que 4
3	6	O preço é MAIOR que 4
4	4	O preço é 4

As cláusulas **TOP**, **LIMIT** ou **FETCH FIRST** servem para **especificar o número de registros a serem retornados em uma consulta**. A cláusula exata depende do SGBD, mas vou deixar aqui pelo menos a sintaxe básica delas para caso apareçam em questões:

SELECT TOP numero * FROM tabela WHERE condição; (SQL Server/MS Access)

SELECT colunas FROM tabela WHERE condição LIMIT numero; (MySQL / PostgreSQL)

SELECT colunas FROM tabela WHERE condição FETCH FIRST numero ROWS ONLY; (Oracle)

EXEMPLIFICANDO!!!

Dada a tabela Clientes a seguir:

IDCliente	Nome_Cliente	Nome_Contato	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

Caso deseje selecionar apenas os 2 primeiros clientes, é possível usar:

SELECT TOP 2 * FROM Clientes; (SQL Server / MS Access)

ou

SELECT * FROM Clientes LIMIT 2; (MySQL / PostgreSQL)

ou

SELECT * FROM Clientes FETCH FIRST 2 ROWS ONLY; (Oracle)

IDCliente	Nome_Cliente	Nome_Contato	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

A cláusula **OFFSET** é usada para **pular um número específico de registros antes de começar a exibir** os resultados.

A sintaxe básica é:

SELECT colunas **FROM** tabela **WHERE** condição **OFFSET** **numero**;

EXEMPLIFICANDO!!!

Dada a tabela Clientes a seguir:

IDCliente	Nome_Cliente	Nome_Contato	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

Case deseje selecionar os registros, pulando os 2 primeiros, é possível usar:

SELECT * FROM Clientes OFFSET 2;

IDCliente	Nome_Cliente	Nome_Contato	Endereco	Cidade	CEP	Pais
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

EXEMPLIFICANDO!!!

Dada a tabela Clientes a seguir:

IDCliente	Nome_Cliente	Nome_Contato	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

Case deseje selecionar 1 registro, pulando os 2 primeiros, é possível usar:

SELECT * FROM Clientes LIMIT 1 OFFSET 2;

IDCliente	Nome_Cliente	Nome_Contato	Endereco	Cidade	CEP	Pais
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

A cláusula **SELECT INTO** é usada para **copiar os dados de uma tabela para uma nova tabela**.

A sintaxe básica é:

SELECT colunas **INTO** nova_tabela **FROM** tabela_original **WHERE** condicao;

É possível também criar a tabela com a atribuição de valores, usando a sintaxe a seguir:

SELECT valor1 AS coluna1, valor2 AS coluna2... **INTO** nova_tabela;

EXEMPLIFICANDO!!!

```
SELECT * INTO Clientes_Brasileiros FROM Clientes WHERE Pais = 'Brasil';
```

```
SELECT 1 AS código, 'João' AS nome INTO alunos;
```

(CESPE / CEBRASPE - 2024 – MPO - Analista de Planejamento e Orçamento) Julgue o item seguinte, relativos à linguagem de definição de dados SQL, à linguagem de manipulação de dados SQL, à linguagem de controle de dados SQL e à linguagem de consulta de dados SQL.

No exemplo de comando UPDATE a seguir, é utilizada a construção case, em que as atualizações nas tuplas são realizadas com um único comando, evitando-se problemas com a ordem das atualizações.

UPDATE tabela

SET coluna = case

 When predicado1 then resultado1

 When predicado2 then resultado2

 ...

 When predicado3 then resultado

 Else resultado

End;

(CESPE / CEBRASPE - 2023 – SEPLAN-RR - Analista de Planejamento e Orçamento) Com pertinência à linguagem SQL, julgue o item abaixo.

Considere-se o seguinte script SQL.

```
select report_code, year, month, day, wind_speed,  
  
case  
  
    when wind_speed >= 40 then 'HIGH'  
  
    when wind_speed >= 30 then 'MODERATE'  
  
    else 'LOW'  
  
end as wind_severity  
  
from station_data
```

O resultado da execução do script resultará em erro, pois, caso haja, na tabela `station_data`, algum registro no campo `wind_speed` com valor superior a 40, não será possível predizer se o valor da variável `wind_severity` será igual a 'HIGH'.

(FCC - 2023 – TRT 18ª Região - Analista Judiciário) Em um banco de dados PostgreSQL 13 aberto e funcionando em condições ideais, deseja-se criar a tabela Tab_TRT18 utilizando o comando SELECT, com o valor 1 no campo Vara, o valor Juiz1 no campo NomeJuiz e o valor vt1goiania@trt18.jus.br no campo Email. O comando correto é:

- a) SELECT Vara AS 1, NomeJuiz AS 'Juiz1', Email AS 'vt1goiania@trt18.jus.br' INTO Tab_TRT18;
- b) SELECT 1 AS Vara, 'Juiz1' AS NomeJuiz, 'vt1goiania@trt18.jus.br' AS Email INTO Tab_TRT18;
- c) SELECT INTO Tab_TRT18 (1 AS Vara, 'Juiz1' AS NomeJuiz, 'vt1goiania@trt18.jus.br' AS Email);
- d) SELECT INTO Tab_TRT18 Vara AS 1, NomeJuiz AS 'Juiz1', Email AS 'vt1goiania@trt18.jus.br';
- e) SELECT VALUES (1 AS Vara, 'Juiz1' AS NomeJuiz, 'vt1goiania@trt18.jus.br' AS Email) INTO Tab_TRT18;

(FGV - 2024 – ALETO – Analista Legislativo) Considere uma tabela fictícia que registre as doações recebidas por projetos sociais de uma determinada prefeitura, criada com o seguinte comando SQL:

```
CREATE TABLE doacoes (  
    SiglaProjeto text,  
    DataDoacao text,  
    Valor REAL,  
    DocumentoDoador text,  
    NomeDoador text);
```

Considere, ainda, que uma certa consulta SQL produziu a seguinte saída:

SiglaProjeto	total_recebido
-----	-----
PA	57865.01
PB	37199.87
PC	18025.90
PD	17828.83
PE	8201.47

Assinale a opção que indica, corretamente, a consulta que poderia ter produzido a saída acima.

- a) `SELECT SiglaProjeto, count(valor) as total_recebido FROM doacoes GROUP BY SiglaProjeto ORDER BY total_recebido DESC LIMIT 5.`
- b) `SELECT SiglaProjeto, sum(Valor) as total_recebido FROM doacoes GROUP BY SiglaProjeto ORDER BY total_recebido DESC LIMIT 5.`
- c) `SELECT SiglaProjeto, sum(Valor) as total_recebido FROM doacoes GROUP BY SiglaProjeto ORDER BY total_recebido LIMIT 5.`
- d) `SELECT SiglaProjeto, count(valor) as total_recebido FROM doacoes GROUP BY SiglaProjeto ORDER BY total_recebido LIMIT 5.`
- e) `SELECT SiglaProjeto, total_recebido FROM doacoes GROUP BY SiglaProjeto ORDER BY total_recebido DESC LIMIT 5.`

(VUNESP - 2024 – Pref. SBC – Técnico) Considerando o sistema gerenciador de bancos de dados MySQL 8.0, o comando para obter os 20 primeiros resultados de uma consulta à tabela Teste, para obter a coluna denominada Inicial dessa tabela, é:

- a) `SELECT Inicial (20) FROM Teste;`
- b) `SELECT Teste(Inicial) LIMIT 20;`
- c) `SELECT Inicial FROM Teste LIMIT 20;`
- d) `SELECT Teste (Inicial) FIRST 20;`
- e) `SELECT FIRST 20 FROM Teste (Inicial);`

(FUNDATEC - 2023 – CAU RS – Analista Superior) Assinale a alternativa que corresponde ao resultado esperado pela execução do comando SQL abaixo:

SELECT *

INTO BKP_VENDAS

FROM VENDAS;

- a) Será criada uma nova tabela, nomeada como BKP_VENDAS, com todos os dados da tabela VENDAS
- b) Serão atualizados apenas os registros da tabela BKP_VENDAS que tiverem chave correspondente na tabela VENDAS
- c) A tabela VENDAS será renomeada para BKP_VENDAS
- d) Será criada uma nova tabela vazia, nomeada como BKP_VENDAS
- e) Serão copiados os dados da tabela BKP_VENDAS para a tabela VENDAS