

Microsoft SQL Server 2016 (T-SQL)

Aula 15

Professor Anderson Henrique

Assuntos tratados nessa aula:

01 – Case – Estrutura de Decisão

02 – Loop While – Estrutura de Repetição

03 – Exercícios

Professor Anderson Henrique

01 – Case

Avalia uma lista de condições e retorna uma das várias expressões de resultado possíveis

A expressão CASE tem dois formatos:

- A expressão case simples compara uma expressão com um conjunto de expressões simples para determinar o resultado
- A expressão CASE pesquisada avalia um conjunto de expressões booleanas para determinar o resultado

Sintaxe:

CASE input_expression.

 WHEN when_expression THEN result_expression [...n]
 [ELSE else_result_expression]

END

CASE

 WHEN boolean_expression THEN result_expression [...n]
 [ELSE else_result_expression]

END

input_expression – é a expressão avaliada quando o formato CASE simples é usado, input_expression é qualquer expressão válida

WHEN when_expression – é uma expressão simples com a qual input_expression é comparada quando o formato CASE simples é usado. When_expression é qualquer expressão válida. Os tipos de dados devem ser os mesmo ou devem ser uma conversão implícita

THEN result_expression – é a expressão retornada quando input_expression igual a when_expression é avaliada como true ou boolean_expression é avaliada como true, result_expression é qualquer expressão válida

ELSE else_result_expression – é a expressão retornada se nenhuma operação for avaliada como **true**. Se esse argumento for omitido e nenhuma operação retornar **true**, CASE retornará **NULL**, else_result_expression é qualquer expressão válida. Os tipos de dados devem ser os mesmos ou devem ser uma conversão implícita

WHEN boolean_expression – é a expressão booleana avaliada quando CASE simples é usado, boolean_expression é qualquer expressão válida

Tipos de retorno

Expressão CASE simples opera comparando a primeira expressão com a expressão em cada cláusula WHEN por equivalência, se forem equivalentes, a expressão será retornada

- Permite somente uma verificação de igualdade
- Na ordem especificada, avalia `input_expression = when_expression` para cada cláusula WHEN
- Retorna a `result_expression` da primeira `input_expression = when_expression` que é avaliada como true
- Se nenhuma `input_expression = when_expression` for avaliada como true, o Mecanismo de Banco de Dados retornará a `else_result_expression`, caso uma cláusula ELSE exista, senão retornará NULL

Expressão CASE pesquisada:

- Avalia, na ordem especificada, a `boolean_expression` de cada cláusula `WHEN`
- Retorna `result_expression` da primeira `boolean_expression` avaliada como `true`
- Se nenhuma `boolean_expression` for avaliada como `true`, o Mecanismo de Banco de Dados retornará a `else_result_expression`, caso uma cláusula `ELSE` exista, senão retornará `NULL`

-- Case simples

```
SELECT nome, genero =  
    CASE genero  
        WHEN 'M' THEN 'Masculino'  
        WHEN 'F' THEN 'Feminino'  
        ELSE 'Outro Gênero'  
    END,  
    salario  
FROM tbl_Cliente  
ORDER BY nome DESC
```

```
SELECT cpf, nome, genero, salario, uf =  
    CASE uf  
        WHEN 'DF' THEN 'Distrito Federal'  
        WHEN 'GO' THEN 'Goiás'  
        ELSE 'Outra UF'  
    END  
FROM tbl_Cliente  
ORDER BY salario DESC
```

-- Case pesquisada

```
SELECT nome, genero, "salario" =  
    CASE  
        WHEN salario < 1000 THEN 'Salário Baixo'  
        WHEN salario >= 1000 AND salario < 2500 THEN 'Salário Mediano'  
        WHEN salario >= 2500 AND salario < 4000 THEN 'Salário Satisfatório'  
        WHEN salario >= 4000 AND salario < 8000 THEN 'Salário Muito Satisfatório'  
        ELSE 'Sonho de Vida Alcançado'  
    END,  
    salario  
FROM tbl_Cliente
```

02 – Loop While

O comando WHILE é usado para executar código enquanto uma condição for verdadeira

ele executa apenas um comando; se for necessário executar um bloco de comandos utilize BEGIN/END

Sintaxe:

WHILE condição

BEGIN

Bloco de códigos

END;

Primeiro Exemplo:

```
DECLARE @valor INT  
SET @valor = 0
```

```
WHILE @valor < 10  
BEGIN  
    PRINT 'Número:' + CAST(@valor AS VARCHAR(2))  
    SET @valor = @valor + 1  
END;
```

Exemplo 2: Imprimir Livro e Preço dos Livros de Ids 1 a 5

```
DECLARE @codigo INT
```

```
SET @código = 1
```

```
WHILE @código < 6
```

```
    BEGIN
```

```
        SELECT ID_Livro AS ID, Nome_Livro AS Livro, Preço_Livro AS  
Preço
```

```
        FROM tbl_Livro
```

```
        WHERE ID_Livro = @código
```

```
        SET @código = @código + 1
```

```
    END;
```

03 – Exercícios

Dúvidas?

Professor Anderson Henrique



Para a próxima aula

01 – Stored Procedures – Criação e Execução

02 – Stored Procedures – Alteração e Parâmetros de Entrada

03 – Stored Procedures – Parâmetros de Entrada e INSERT

Professor Anderson Henrique

