Microsoft SQL Server 2016 (T-SQL)

Aula 13

Professor Anderson Henrique

Assuntos tratados nessa aula:

01 – VIEWS (Exibições) – Criar, Alterar e Excluir

02 – Subconsultas (Subqueries) com Tabelas Derivadas, Operador Exists

03 – Subconsultas (Subqueries) com operadores SOME, ANY, ALL

04 – CTE (Common Table Expression (subconsultas))

Professor Anderson Henrique

01 – VIEWS

Uma Exibição (Visão) é uma tabela virtual baseada no conjunto de resultados de uma consulta SQL

Contém linhas e colunas como uma tabela real, e pode receber comandos como declarações JOIN, WHERE e funções como uma tabela normal

Mostra sempre resultados de dados atualizados, pois o motor do banco de dados recria os dados toda vez que um usuário consulta a visão

VIEWS – Criação

```
CREATE VIEW [Nome_Exibição]
AS SELECT colunas
FROM tabela
WHERE condições
```

```
CREATE VIEW vw_LivrosAutores
AS SELECT tbl_Livro.Nome_Livro AS Livro,
Tbl_Autor.Nome_Autor AS Autor
FROM tbl_Livro
INNER JOIN tbl_Autor ON tbl_Livro.ID_Autor = tbl_Autor.ID_Autor;
```

VIEWS - Alteração

```
ALTER VIEW vw_LivrosAutores
AS SELECT tbl_Livro.Nome_Livro AS Livro,
Tbl_Autor.Nome_Autor AS Autor, Preco_Livro AS Valor
FROM tbl_Livro
INNER JOIN tbl_Autor ON tbl_Livro.ID_Autor = tbl_Autor.ID_Autor;
```

VIEWS – Exclusão

```
DROP VIEW vw_LivrosAutores;
```

TECH
soluções em ti

02 – Subconsultas (Subqueries), Operador Exists

Uma subconsulta (subquery) é uma declaração SQL embutida em uma consulta externa

A subconsulta fornece uma resposta à consulta externa na forma de um valor escalar, lista de valores, ou conjunto de dados, equivalentes a uma expressão, lista ou tabela para a consulta externa

Exemplo simples:
SELECT (SELECT ‘Anderson’) AS Subconsulta;

Subconsultas com Tabelas Derivadas

```
--Subconsulta de uma única linha: retornam zero ou uma linha
--para instrução SQL externa
SELECT Nome_Autor, Sobrenome_Autor FROM tbl_Autor
WHERE ID_Autor =
(SELECT ID_Autor FROM tbl_Autor WHERE Sobrenome_Autor = 'Blum')

--Subconsulta usa a função para obter o preço médio dos produtos
-- é passado para a cláusula where da consulta externa
SELECT ID_Produto, Descricao, Preco
FROM tbl_Produto WHERE Preco >
(SELECT AVG(Preco) FROM tbl_Produto)
```

TECH
soluções em ti

Alterar a tabela tbl_Produto adicionando colunas

```
ALTER TABLE tbl_Produto ADD QTD_Vendas INT
ALTER TABLE tbl_Produto ADD UF VARCHAR(2)
```

Atualizar os registros

```
UPDATE tbl_Produto SET QTD_Vendas = 3, UF = 'DF' WHERE
ID_Produto = 1
UPDATE tbl_Produto SET QTD_Vendas = 10, UF = 'GO' WHERE
ID_Produto = 2
UPDATE tbl_Produto SET QTD_Vendas = 7, UF = 'DF' WHERE
ID_Produto = 3
```

TECH
soluções em ti

```
UPDATE tbl_Produto SET QTD_Vendas = 22, UF = 'RJ' WHERE
ID_Produto = 4
UPDATE tbl_Produto SET QTD_Vendas = 18, UF = 'GO' WHERE
ID_Produto = 5
UPDATE tbl_Produto SET QTD_Vendas = 4, UF = 'DF' WHERE
ID_Produto = 6
UPDATE tbl_Produto SET QTD_Vendas = 9, UF = 'RJ' WHERE
ID_Produto = 7
```

```
--Subconsulta utilizando função de agregação
SELECT UF, SUM(QTD_Vendas) AS 'Total de Vendas '
FROM tbl_Produto
GROUP BY UF

--Subconsulta retorna resultados para a consulta (Preco > (59.99))
SELECT Descricao, Preco FROM tbl_Produto
WHERE Preco > (SELECT MIN(Preco) + 20.00 FROM tbl_Produto)
```

TECH
soluções em ti

```
--Subconsulta em uma cláusula HAVING

SELECT UF, SUM(QTD_Vendas) AS 'Total de Vendas '
FROM tbl_Produto
GROUP BY UF
HAVING SUM(QTD_Vendas) >
(SELECT MAX(QTD_Vendas) FROM tbl_Produto)
```

02 – Operador Exists

O operador EXISTS é usado para testar a existência de qualquer registro em uma subconsulta. O operador EXISTS retornará true se a consulta retornar um ou mais registros

Sintaxe:
SELECT coluna
FROM tabela
WHERE EXISTS
(SELECT coluna FROM tabela WHERE condições)

Sintaxe:

```
SELECT coluna
FROM tabela
WHERE EXISTS
(SELECT coluna FROM tabela WHERE condições)
```

```
SELECT coluna
FROM tabela
WHERE NOT EXISTS
(SELECT coluna FROM tabela WHERE condições)
```

```
SELECT * FROM tbl_Pedido
```

```
SELECT cpf, nome, salario
FROM tbl_Cliente WHERE EXISTS
(SELECT *
FROM tbl_Pedido
WHERE tbl_Pedido.ID_Cliente = tbl_Cliente.ID_Cliente)
```

```
SELECT cpf, nome, salario
FROM tbl_Cliente WHERE NOT EXISTS
(SELECT *
FROM tbl_Pedido
WHERE tbl_Pedido.ID_Cliente = tbl_Cliente.ID_Cliente)
```

03 – Subconsultas com operadores [Some e Any], operador All

As consultas e subconsultas que utilizam estes operadores de comparação, retornam valores numéricos maior ou igual à zero para diversas funcionalidades, como cláusulas GROUP BY, HAVING ou subconsultas declaradas com EXISTS

O objetivo é facilitar a execução sobre a comparação de um ou mais registos, através de valores escalares, que atendam à uma condição booleana

Ambiente para Teste

```
CREATE TABLE tbl_Test(
        ID_Cliente smallint IDENTITY PRIMARY KEY,
        NM_Cliente varchar(20) NOT NULL
)

INSERT INTO tbl_Test(NM_Cliente)VALUES('CLIENTE 1')
INSERT INTO tbl_Test(NM_Cliente)VALUES('CLIENTE 2')
INSERT INTO tbl_Test(NM_Cliente)VALUES('CLIENTE 3')

SELECT * FROM tbl_Test
```

Os operadores SOME e ANY são semelhantes e lembram “um pouco” a instrução IN, mas retornam um valor booleano “true” ou “false”. Os valores para comparação sempre devem ser baseados em uma única coluna

Ao menos um dos valores deve atender a condição indicada:

```
--declarando uma variável
DECLARE @VALOR INT = 3

--É "menor" do que "algum" dos números desta coluna
SELECT iif((@VALOR < SOME (SELECT ID_Cliente FROM tbl_Test)),
                        'É menor que algum', 'Não é menor que algum')
```

```
--É "diferente" que "algum" dos números desta coluna
SELECT iif((@VALOR <> SOME (SELECT ID_Cliente FROM tbl_Test)),
                    'É diferente  de algum', 'Não é diferente de algum')

--É "menor" que "qualquer" dos números desta coluna
SELECT iif((@VALOR < ANY (SELECT ID_Cliente FROM tbl_Test)),
                    'É menor que qualquer', 'Não é menor que qualquer')

--É "diferente" que "qualquer" dos números desta coluna
SELECT iif((@VALOR <> ANY (SELECT ID_Cliente FROM tbl_Test)),
                    'É diferente  de qualquer', 'Não é diferente de qualquer')
```

Com o operador ALL, deverá ser verificado se o valor está entre um dos registros armazenados na coluna indicada

Também possui um retorno booleano, semelhante a SOME e ANY, porém, neste caso somente retornará o resultado “true” quando encontrar ao menos um valor comparado entre o conjunto de uma coluna

Você pode obter os mesmos resultado com o operador “<> ALL” semelhante a cláusula “NOT IN”

```
--declarando uma variável
DECLARE @VALOR INT = 1

--É "menor" que "todos" os números desta coluna
SELECT iif((@VALOR < ALL(SELECT ID_Cliente FROM tbl_Test)),
                        'É menor que todos', 'Não é menor que todos')

--É "diferente" que "todos" os números desta coluna
SELECT iif((@VALOR < ALL(SELECT ID_Cliente FROM tbl_Test)),
                        'É diferente de todos', 'Não é diferente de todos')
```

04 – CTE (Common Table Expression)

Common Table Expression | Expressão de Tabela Comum

Variação sintática de uma subconsulta, similar a uma exibição (view)

Pode ser acessada múltiplas vezes dentro da consulta principal, como se fosse uma exibição ou tabela

Sintaxe:

[WITH <common_table_expression> [,...n]]
<common_table_expression>::=
expression_name [(column_name [,...n])]
AS
(CTE_query_definition)

```
WITH CTE_Livro_Autor(ID, Titulo, ISBN, Data, Preco, Nome,
                     Sobrenome)
AS
(
        SELECT
        ID_Livro,
        Nome_Livro,
        ISBN_Livro,
        Data_Pub,
        Preco,
        Nome_Autor,
        Sobrenome_Autor
        FROM tbl_Livro
        FULL JOIN tbl_Autor
        ON tbl_Autor.ID_Autor = tbl_Livro.ID_Autor
)
```

Dúvidas?

Professor Anderson Henrique

Para a próxima aula

01 – Variáveis – Declaração e Atribuição de Valores

02 – Conversão de Tipos de Dados com Cast e Convert

03 – Condicional IF e ELSE – Estrutura de Decisão

Professor Anderson Henrique