

Microsoft SQL Server 2016 (T-SQL)

Aula 12

Professor Anderson Henrique

Assuntos tratados nessa aula:

01 – Concatenação de Strings e Funções

02 – Collation – Agrupamento de Caracteres ou Colaço

03 – Funções de Data e Matemáticas

04 – Cláusula WITH TIES

Professor Anderson Henrique



01 – Concatenação de Strings

É possível concatenar strings usando o operador de concatenação (+), unir ou juntar textos

Sintaxe:

String1 | coluna + string2 | coluna

```
SELECT 'Anderson ' + ' da LogusTI Treinamentos' AS 'Logus'
```

```
SELECT Nome_Autor +' '+ Sobrenome_Autor AS  
'Nome Completo' FROM tbl_Autor
```

```
SELECT 'Eu gosto do livro ' + Nome_Livro As 'Meu Livro' FROM tbl_Livro  
WHERE ID_Autor = 2
```

01 – Funções String

As funções string manipulam a estrutura de caracteres alfanuméricos definidos em um banco de dados. Veremos a função **ASCII()** – retorna o código ASCII do primeiro caractere de uma string informada

Exemplo:

```
SELECT ASCII('A')
```

```
SELECT ASCII('B')
```

```
SELECT ASCII('ABACATE')
```

A função **CHAR()** – retorna o caractere da tabela ASCII a partir de um valor do tipo inteiro fornecido entre 0 e 255

Exemplo:

```
SELECT CHAR(65)
```

```
SELECT CHAR(66)
```

```
SELECT CHAR(67)
```

A função **LEN()** – retorna o número de caracteres da expressão alfanumérica, ou seja, retorna o tamanho da string, excluindo os espaços em branco à direita

Exemplo:

```
SELECT LEN('Anderson Henrique') as 'Qtd de caracteres'  
SELECT LEN('Constituição')  
SELECT LEN('SQL Server')
```

As funções **LEFT**(caractere,posição) e **RIGHT**(caractere,posição) – retorna a parte esquerda/direita de uma cadeia de caracteres com o número de caracteres especificado

Exemplo:

```
SELECT LEFT('Anderson Henrique', 5) as '5 caracteres à esquerda'  
SELECT RIGHT('SQL Server', 3) as '3 caracteres à direita'
```


As funções **UPPER**(expressão) e **LOWER**(expressão) – retorna em formato maiúsculo/minúsculo a cadeia de caracteres informada

Exemplo:

```
SELECT UPPER('Anderson Henrique') as 'converte maiúsculo'  
SELECT LOWER('SQL Server') as 'converte minúsculo'
```

As funções **LTRIM**(expressão) e **RTRIM**(expressão) – remove de uma cadeia de caracteres os espaços em branco que porventura existam na parte esquerda/direita da string

Exemplo:

SELECT LTRIM('Anderson Henrique')	as 'elimina esquerda'
SELECT RTRIM('SQL Server')) as 'elimina direita'

A função **REPLACE**(cadeia, outra, trecho) – substitui todas as ocorrências de uma cadeia de caracteres por outra cadeia de caracteres a partir de um trecho dessa cadeia

Exemplo:

```
SELECT REPLACE('O Palmeiras é o melhor time do Brasil',  
'Palmeiras','Cruzeiro')
```

```
SELECT REPLACE('O rato roeu a roupa do rei de Roma', 'Roma','Copas')
```

A função **SUBSTRING**(cadeia, início, tamanho) – retorna parte de uma cadeia de caracteres a partir da delimitação de um ponto inicial e de um tamanho como ponto final. Os argumentos início e tamanho são valores do tipo inteiro

Exemplo:

```
SELECT SUBSTRING('COMPUTADOR', 4, 4)  
SELECT SUBSTRING('CONSTITUIÇÃO', 5, 8)
```

02 – Collation (‘Colação’ / ‘Agrupamento’)

Trata-se da codificação dos caracteres em uma ordem padrão

Muitos sistemas de colação são baseados em ordem numérica ou alfabética

A colação usada pelos bancos de dados é configurada durante a instalação do sistema

Collation-charts.org – site com cartas de agrupamento em vários alfabetos e idiomas

Para ver as opções de agrupamento

SELECT * FROM fn_helpcollations

Para ver o esquema de colação usada atualmente

SELECT SERVERPROPERTY('Collation') AS Colacao_Usada

Alterar o esquema de agrupamento atual

ALTER DATABASE db_Biblioteca

COLLATE Greek_CI_AI

Verificar o esquema de agrupamento de um Banco de Dados

SELECT DATABASEPROPERTYEX('db_Biblioteca', 'Collation')

```
SELECT * FROM tbl_Livro  
ORDER BY Nome_Livro  
COLLATE Icelandic_CI_AI
```

03 – Funções de DATA

A função **GETDATE()** retorna a data e a hora atuais do sistema de banco de dados. Esse valor é oriundo do sistema operacional do computador no qual a instância do SQL Server está sendo executada

Exemplo:

```
SELECT GETDATE() as 'Data-Hora Atuais';
```


Funções **DAY(data)**, **MONTH(data)** e **YEAR(data)**, elas retornam como valor inteiro, respectivamente, os números do dia, mês e ano de uma determinada data fornecida como argumento

Exemplo :

```
SELECT DAY('2019-07-22') as 'Dia'  
SELECT MONTH('2019-07-22') as 'Mês'  
SELECT YEAR('2019-07-22') as 'Ano'
```

A função **DATEADD**(parte, valor, data) retorna uma determinada data com base num valor inteiro adicionado à parte especificada dessa data

Os argumentos parte pode ser uma das constantes **year** (ano), **quarter** (trimestre), **month** (mês), **dayofyear** (dia do ano), **day** (dia), **week** (semana), **weekday** (dia da semana), **hour** (hora), **minute** (minuto), **second** (segundo), **millisecond** (milissegundos), **microsecond** (microssegundos) e **nanosecond** (nanossegundos); valor é o valor que será usado para a operação de adição

Exemplo :

```
SELECT DATEADD(Month, 1, '26/03/1965')  
SELECT DATEADD(Day, 5, '22/07-2020')
```

A função **DATEDIFF**(parte, início, fim) retorna como contagem um valor inteiro relacionado aos limites definidos de uma parte em relação a um início e fim especificados

Os argumentos parte pode ser uma das constantes **year** (ano), **quarter** (trimestre), **month** (mês), **dayofyear** (dia do ano), **day** (dia), **week** (semana), **weekday** (dia da semana), **hour** (hora), **minute** (minuto), **second** (segundo), **millisecond** (milissegundos), **microsecond** (microssegundos) e **nanosecond** (nanossegundos); início é o valor da data inicial e fim é o valor da data final

Exemplo :

```
SELECT DATEDIFF(Day, '18-07-2020', '25-07-2020')  
SELECT DATEDIFF(Year, '18-07-2016', '18-07-2020')
```

A função **DATENAME**(parte, data) retorna uma cadeia de caracteres que representa a parte especificada da data informada

Os argumentos parte pode ser uma das constantes **year** (ano), **quarter** (trimestre), **month** (mês), **dayofyear** (dia do ano), **day** (dia), **week** (semana), **weekday** (dia da semana), **hour** (hora), **minute** (minuto), **second** (segundo), **millisecond** (milissegundos), **microsecond** (microssegundos) e **nanosecond** (nanossegundos); data é o valor da data a ser utilizada

Exemplo :

```
SELECT DATENAME(Day, '22/07/2020') as 'Dia'  
SELECT DATENAME(Month, '22/07/2020') as 'Mês'
```

```
SELECT 'Hoje é dia ' + datename(Day, '22/07/2020') + ' de ' +  
datename(Month, '22/07/2020') + ' de '  
+ datename(year, '22/07/2020')
```

03 – Funções Matemáticas

A função **ABS(expressão)** retorna o valor positivo, ou seja, o valor absoluto de um valor numérico inteiro ou real positivo ou negativo fornecido como expressão

Exemplo :

```
SELECT ABS(10)
```

```
SELECT ABS(-8.238765)
```

```
SELECT ABS(-9)
```

A função **FLOOR(expressão)** retorna como resultado o arredondamento para baixo de um valor do tipo real fornecido. Desta forma obtém-se com essa função a parte inteiro (expoente) de um valor fornecido

Exemplo :

```
SELECT FLOOR(1.1)
```

```
SELECT FLOOR(1.4)
```

```
SELECT FLOOR(2.1)
```

```
SELECT FLOOR(2.4)
```

A função **LOG(expressão)** retorna como resultado o logaritmo natural de um valor do tipo real fornecido

Exemplo :

```
SELECT LOG(3)  
SELECT LOG(10)
```

A função **PI()** retorna como resultado o valor da constante matemática pi

```
SELECT PI()  
SELECT PI() / 2
```


A função **POWER(base, expoente)** retorna resultado da potência de uma base elevada a um expoente que pode ser um valor do tipo inteiro ou real

Exemplo :

```
SELECT POWER(2.0, 3.0)
```

```
SELECT POWER(2, 3)
```

```
SELECT POWER(25, 0.5)
```

A função **SQRT(expressão)** retorna resultado da raiz quadrada em tipo real, representado pela expressão de um valor numérico do tipo real ou inteiro positivo fornecido

Exemplo :

```
SELECT SQRT(9)  
SELECT SQRT(2)  
SELECT SQRT(2.5)
```

A função **RAND([expressão])** retorna como resultado um valor do tipo real pseudoaleatório entre 0 e 1. O valor da expressão, que é opcional, deve ser do tipo inteiro

Exemplo :

-- sorteia um valor entre 0 e 1

```
SELECT RAND()
```

--simula o sorteio de valores entre 1 e 5

```
SELECT FLOOR(RAND() * 4 + 1)
```

04 – Cláusula WITH TIES

Quando utilizamos o comando TOP, as vezes alguns itens adjacentes deixam de ser exibidos, principalmente quando eles possuem o mesmo valor do último item da lista retornada

Podemos exibir os 3 primeiros times de uma lista que possuem as 3 maiores pontuações, pode existir um quarto time com a mesma pontuação do terceiro, ele ficará de fora da lista

```
SELECT TOP(3) WITH TIES Nome_Time, Pontos  
FROM tbl_Times  
ORDER BY Pontos DESC
```

```
SELECT TOP(5) WITH TIES Nome_Time, Pontos  
FROM tbl_Times  
ORDER BY Pontos DESC
```

Dúvidas?

Professor Anderson Henrique



Para a próxima aula

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

02 – Subconsultas (Subqueries) com Tabelas Derivadas

03 – CTE (Common Table Expression (subconsultas))

Professor Anderson Henrique

