

```

/*
| *Função*           | *Descrição*           |
|   | *Sintaxe e Parâmetros*           |
| *Exemplo de Uso*           | *Exemplo de           |
| Resposta*           |
|-----|-----|
|-----|-----|
|-----|-----|
|-----|-----|
| *DATEDIFF()*           | Calcula a diferença entre duas datas em uma unidade |
| específica (dia, mês, ano). | DATEDIFF(unidade, data_inicio, data_fim) |
|           | DATEDIFF(YEAR, '2000-01-01', GETDATE()) |
|           | 24 (diferença em anos) |
| *YEAR()*           | Extrai o ano de uma data. |
|           | YEAR(data) |
| YEAR('2024-08-09')           | 2024 |
|
| *SUBSTRING()*           | Extrai uma porção específica de uma string. |
|           | SUBSTRING(texto, posição_inicial, comprimento) |
| SUBSTRING('SQL Database', 5, 8)           | Database |
|
| *CONCAT()*           | Concatena duas ou mais strings. |
|           | CONCAT(string1, string2, ...) |
| CONCAT('Hello', ' ', 'World!')           | Hello World! |
|
| *LEN()*           | Retorna o comprimento de uma string. |
|           | LEN(texto) |
| LEN('OpenAI')           | 6 |
|
| *CAST()*           | Converte um valor de um tipo de dado para outro. |
|           | CAST(valor AS tipo_destino) |
| CAST(123.45 AS INT)           | 123 |
|
| *CONVERT()*           | Converte um valor de um tipo de dado para outro, com |
| opções de estilo. | CONVERT(tipo_destino, valor, estilo) |
|           | CONVERT(VARCHAR, GETDATE(), 103) |
| 09/08/2024 (DD/MM/YYYY) |
| *GETDATE()*           | Retorna a data e hora atuais do sistema. |
|           | GETDATE() |
| GETDATE()           | 2024 -08-09 |
| 10:00:00.000 (exemplo) |
| *RAND()*           | Gera um número aleatório entre 0 e 1. |
|           | RAND() |
| RAND()           | 0.547923 |
| (exemplo) |
| *DATEPART()*           | Extrai uma parte específica de uma data (ano, mês, | |
| dia, etc.). | DATEPART(parte, data) |
|           | DATEPART(MONTH, '2024-08-09') | 8 (mês de |
| agosto) |
| *DATENAME()*           | Retorna o nome textual de uma parte específica de uma |
| data. | DATENAME(parte, data) |
| DATENAME(WEEKDAY, '2024-08-09')           | Friday |
|

```

...Drive\Área de Trabalho\Fatec\aula 2\funcoes_nativas.sql 2

```
| *LOWER()*          | Converte uma string para letras minúsculas.  2
|         | LOWER(texto)                                | 2
| LOWER('SQL SERVER')          | sql server 2
|
| *UPPER()*          | Converte uma string para letras maiúsculas.  2
|         | UPPER(texto)                                | 2
| UPPER('sql server')          | SQL SERVER 2
|
| *LEFT()*           | Retorna um número especificado de caracteres do 2
| início de uma string. | LEFT(texto, número_caracteres)                2
|         | LEFT('Database', 4)                            | Data 2
|
| *RIGHT()*           | Retorna um número especificado de caracteres do final 2
| de uma string. | RIGHT(texto, número_caracteres)                2
|         | RIGHT('Database', 4)                            | base 2
|
| *ROUND()*           | Arredonda um número para um número específico de 2
| casas decimais. | ROUND(valor, casas_decimais)                    2
|         | ROUND(123.456, 2)                                | 123.46 2
|
| *COALESCE()*        | Retorna o primeiro valor não nulo em uma lista de 2
| argumentos. | COALESCE(valor1, valor2, ...)                    2
|         | COALESCE(NULL, 'Default', 'Another')            | Default 2
|
```

Explicação dos Parâmetros

```
- *unidade*: A unidade de medida para calcular a diferença entre datas (pode ser 2
  DAY, MONTH, YEAR, etc.).
- *data_inicio* e *data_fim*: As datas entre as quais a diferença será 2
  calculada.
- *texto*: A string a ser manipulada.
- *posição_inicial*: A posição inicial de onde a substring deve começar.
- *comprimento*: O número de caracteres a serem extraídos na substring.
- *string1, **string2*: Strings que serão concatenadas.
- *valor*: O valor a ser convertido ou manipulado.
- *tipo_destino*: O tipo de dado para o qual o valor será convertido.
- *parte*: A parte da data que se deseja extrair (pode ser YEAR, MONTH, DAY, 2
  HOUR, WEEKDAY, etc.).
- *número_caracteres*: O número de caracteres a serem retornados de uma string.
- *casas_decimais*: O número de casas decimais para o qual um valor deve ser 2
  arredondado.
- *valor1, **valor2*, etc.: Valores a serem verificados por COALESCE(), 2
  retornando o primeiro não nulo.
*/
```

DECLARE @nome VARCHAR(8000) = 'Paola Bracho';

```
DECLARE @n INT = LEN(@nome);

PRINT('Número de caracteres no nome: ' + CONVERT(VARCHAR,@n));
PRINT('Substring começando em 5 e avançando 10 caracteres: ' + SUBSTRING(@nome, 5, 10)); -- SQL é 1-indexed.
PRINT('Substring dos 5 primeiros caracteres: ' + LEFT(@nome,5));
PRINT('Substring dos 5 últimos caracteres: ' + RIGHT(@nome,5));
PRINT('Ativação CAPS LOCK: ' + UPPER(@nome));
PRINT('Desativação CAPS LOCK: ' + LOWER(@nome));
PRINT(REPLACE(@nome, 'Paola', 'Paulina'));

DECLARE @resultado_de_concat VARCHAR(8000) = CONCAT('Sra. ', 'Paola', ' ', 'Bracho', ' ', 'MsC. ');
PRINT(@resultado_de_concat);

/*
SQL_VARIANT_PROPERTY gera uma informação tabelada acerca da variável:
*/
SELECT SQL_VARIANT_PROPERTY(@n, 'BaseType') AS Tipo,
       SQL_VARIANT_PROPERTY(@n, 'Precision') AS 'Precisão',
       SQL_VARIANT_PROPERTY(@n, 'Scale') AS Escala;

/*
Funções matemáticas.
*/

DECLARE @PI FLOAT = 3.14159265;

DECLARE @sinpi2 FLOAT = SIN(@PI / 2),
        @cospi2 FLOAT = COS(@PI / 2),
        @tanpi2 FLOAT = TAN(@PI / 2);

PRINT(CONCAT('sen(PI/2) = ', @sinpi2));
PRINT(CONCAT('cos(PI/2) = ', @cospi2));
PRINT(CONCAT('tan(PI/2) = ', @tanpi2));

DECLARE @x_1 FLOAT = 0.1,
        @y_1 FLOAT = 0.2,
        @x_2 FLOAT = 0.3,
        @y_2 FLOAT = 0.4;
/*
DECLARE @z FLOAT = SQRT(POWER(@x_1 - @x_2, 2) + POWER(@y_1 - @y_2, 2));

PRINT(@z);*/

DECLARE @z FLOAT = POWER(POWER(@x_1 - @x_2, 2) + POWER(@y_1 - @y_2, 2), 0.5);

PRINT(@z);
```

```
/*
Funções de arredondamento:

*/

DECLARE @E FLOAT = 2.718281828;

PRINT(CONCAT('CEILING(', @E, ') = ', CEILING(@E)));
PRINT(CONCAT('FLOOR(', @E, ') = ', FLOOR(@E)));
PRINT(CONCAT('ROUND(', @E, ', 2) = ', ROUND(@E, 2))); -- ROUND(<variável>,
    <qtde_de_casas_decimais>)

/*
Geração de números aleatórios: RAND() gera números pseudorandoms entre 0 e 1.
*/

PRINT(RAND());

/*
Tratamento de datas!

Funções:
GETDATE(),
DATEPART(<parte_da_data>, <data>),
DATENAME(<parte_da_data>, <data>),
DATEDIFF(<unidade>, <data1>, <data2>),
YEAR(<data>), DAY(<data>), MONTH(<data>)

*/

DECLARE @data_de_hoje DATETIME = GETDATE(),
        @data_niver DATETIME = '1992-04-12';

PRINT(CONCAT('Dia de hoje: ', @data_de_hoje));

PRINT(YEAR(@data_de_hoje));
PRINT(DATENAME(MONTH, @data_de_hoje));
PRINT(DAY(@data_de_hoje));

PRINT(CONCAT('Dias de vida: ', DATEDIFF(DAY, @data_de_hoje, @data_niver)));

PRINT(CONCAT('Segundo atual: ', DATEPART(SECOND, @data_de_hoje)));

/*
Função de geração de código hexadecimal randômico: NEWID()

Gera valores literais do tipo [8]-[4]-[4]-[4]-[12].

```

Exemplo de código gerado: A87100DC-975E-4891-8F6A-D4FF809AE823.
*/

PRINT(NEWID());