# MANIPULAÇÃO DE DADOS

## CONVERT

Vamos iniciar falando sobre a função CONVERT, essa função é usada para converter um valor de um tipo de dado para outro. Isso pode ser útil em diversas situações, como quando é necessário formatar uma data de maneira específica, converter um tipo de dados para outro para realizar cálculos ou para exibir dados de uma maneira diferente.

A sintaxe básica da função CONVERT é a seguinte:

CONVERT(data\_type, expression [, style])

data\_type: Especifica o tipo de dados para o qual a expressão deve ser convertida.

expression: A expressão que se deseja converter.

style (opcional): Especifica um estilo de formatação para a conversão, geralmente usado com tipos de dados de data e hora.

Aqui estão alguns exemplos de como a função CONVERT pode ser usada:

Converter uma string em um tipo de dados numérico:

SELECT CONVERT(int, '123') AS valor\_convertido;

Isso converte a string '123' em um valor inteiro.

Converter um valor numérico em uma string:

SELECT CONVERT(varchar, 123) AS valor\_convertido;

Isso converte o valor numérico 123 em uma string.

Converter uma data em um formato diferente:

SELECT CONVERT(varchar, GETDATE(), 101) AS valor\_convertido;

Isso converte a data atual para o formato 'mm/dd/yyyy'. O terceiro parâmetro (estilo) é opcional e define o formato de saída da data. Aqui, '101' é um estilo de formatação específico.

Converter uma data em um tipo de dados diferente:

SELECT CONVERT(datetime, '2022-01-01') AS valor\_convertido;

Isso converte a string '2022-01-01' em um tipo de dados de data.

Converter uma string em um tipo de dados float:

SELECT CONVERT(float, '123.45') AS valor\_convertido;

Isso converte a string '123.45' em um valor de ponto flutuante.

Converter um valor float em uma string:

SELECT CONVERT(varchar, 123.45) AS valor\_convertido;

Isso converte o valor de ponto flutuante 123.45 em uma string.

Converter uma data em um formato diferente (dia/mês/ano):

SELECT CONVERT(varchar, GETDATE(), 103) AS data\_convertida;

Isso converte a data atual para o formato 'dd/mm/yyyy'. O terceiro parâmetro (estilo) é opcional e define o formato de saída da data. Aqui, '103' é um estilo de formatação específico.

Converter uma data em um formato ISO (YYYY-MM-DD):

SELECT CONVERT(varchar, GETDATE(), 23) AS data\_convertida;

Isso converte a data atual para o formato ISO 'YYYY-MM-DD'. Aqui, '23' é um estilo de formatação específico.

Converter uma data e hora em um formato de 24 horas:

SELECT CONVERT(varchar, GETDATE(), 108) AS data\_convertida;

Isso converte a data e hora atual para o formato de 24 horas. Aqui, '108' é um estilo de formatação específico.

Converter uma data em um formato curto:

SELECT CONVERT(varchar, GETDATE(), 12) AS data\_convertida;

Isso converte a data atual para um formato curto. Aqui, '12' é um estilo de formatação específico.

Converter uma data em um formato numérico (AAAAMMDD):

SELECT CONVERT(int, CONVERT(varchar, GETDATE(), 112)) AS data\_convertida;

Isso converte a data atual para um formato numérico 'AAAAMMDD', que pode ser útil para comparações numéricas.

Converter um valor decimal em uma string com casas decimais:

SELECT CONVERT(varchar, CONVERT(decimal(10,2), 123.456)) AS valor\_convertido;

Isso converte o valor decimal 123.456 em uma string com duas casas decimais.

Converter uma string em um tipo de dados money:

SELECT CONVERT(money, '123.45') AS valor\_convertido;

Isso converte a string '123.45' em um valor de dinheiro.

Converter um valor money em uma string com formatação de moeda:

SELECT CONVERT(varchar, CONVERT(money, 123.45), 1) AS valor\_convertido;

Isso converte o valor de dinheiro 123.45 em uma string com formatação de moeda. O terceiro parâmetro (estilo) define a formatação de moeda.

Converter uma string em um tipo de dados booleano:

SELECT CONVERT(bit, 'true') AS valor\_convertido;

Isso converte a string 'true' em um valor booleano.

Converter um valor booleano em uma string:

SELECT CONVERT(varchar, 1) AS valor\_convertido;

Isso converte o valor booleano 1 em uma string.

Converter uma string em uma data e hora:

SELECT CONVERT(datetime, '2024-02-06 15:30:00') AS data\_convertida;

Isso converte a string '2024-02-06 15:30:00' em um tipo de dados de data e hora.

Converter uma data e hora em um formato específico com separadores:

SELECT CONVERT(varchar, GETDATE(), 103) + ' ' + CONVERT(varchar, GETDATE(), 108) AS data\_convertida;

Isso converte a data e hora atual para um formato 'dd/mm/yyyy hh:mm:ss'.

Formatos de estilos disponíveis, se aplica apenas na conversão para data.

101: Formato dos EUA - mm/dd/aaaa

102: Formato ANSI - aaaa.mm.dd

103: Formato Britânico/Francês - dd/mm/aaaa

104: Formato Alemão - dd.mm.aaaa

105: Formato Italiano - dd-mm-aaaa

106: Formato Espanhol - dd mm aaaa

107: Formato Padrão - mês dd, aaaa

108: Hora - hh:mi:ss

109: Padrão com hora - mm-dd-aaaa hh:mi:ss

110: EUA - mm-dd-aaaa

111: Japonês - aaaa/mm/dd

112: ISO - aaaa-mm-dd

113: Formato Europeu com hora - dd mês aaaa hh:mi:ss

114: Hora sem segundos - hh:mi

120: ODBC com timestamp - aaaa-mm-dd hh:mi:ss

121: ODBC sem frações de segundo - aaaa-mm-dd hh:mi:ss

126: ISO8601 com frações de segundo - aaaa-mm-ddThh:mi:ss.mmm

127: ISO8601 sem frações de segundo - aaaa-mm-ddThh:mi:ssZ

130: Hijri - dd mês aaaa hh:mi:ss

131: Hijri - dd/mm/aaaa

É importante observar que a função CONVERT pode gerar erros se a conversão não for possível. Por exemplo, tentar converter uma string que não representa um número em um tipo de dados numérico resultará em um erro. Além disso, nem todas as conversões são suportadas para todos os tipos de dados. Por exemplo, nem todas as strings podem ser convertidas em datas.

## CAST

Falando agora sobre a função CAST, ela também é usada para converter um valor de um tipo de dados para outro. É semelhante à função CONVERT, mas com algumas diferenças fundamentais em seu comportamento e sintaxe.

Aqui está a sintaxe básica da função CAST:

CAST(expression AS data\_type)

expression: A expressão que você deseja converter.

data\_type: O tipo de dados para o qual você deseja converter a expressão.

O CAST é especialmente útil quando você precisa de uma conversão direta entre tipos de dados sem formatação adicional, pois não oferece opções de formatação como a função CONVERT. No entanto, ele é mais limitado em termos de tipos de dados que pode manipular em comparação com CONVERT.

Aqui estão alguns exemplos de como você pode usar a função CAST:

Converter uma string em um tipo de dados numérico:

SELECT CAST('123' AS int) AS valor\_convertido;

Isso converte a string '123' em um valor inteiro.

Converter um valor numérico em uma string:

SELECT CAST(123 AS varchar) AS valor\_convertido;

Isso converte o valor numérico 123 em uma string.

Converter uma string em um tipo de dados de data:

SELECT CAST('2022-01-01' AS date) AS data\_convertida;

Isso converte a string '2022-01-01' em um tipo de dados de data.

Converter uma data em um tipo de dados de data e hora:

SELECT CAST('2022-01-01' AS datetime) AS data\_convertida;

Isso converte a string '2022-01-01' em um tipo de dados de data e hora.

## ISNULL

A função ISNULL no SQL Server é usada para substituir valores nulos por um valor especificado. Essa função é especialmente útil ao lidar com consultas que podem retornar valores nulos em determinadas circunstâncias. A sintaxe básica da função ISNULL é a seguinte:

ISNULL(expression, replacement\_value)

expression: A expressão que será verificada quanto a nulos.

replacement\_value: O valor a ser retornado se a expressão for nula.

Quando a expressão não é nula, a função ISNULL retorna o valor da expressão original. No entanto, se a expressão for nula, o ISNULL retorna o valor especificado como replacement\_value.

Aqui está um exemplo de como a função ISNULL pode ser usada:

SELECT ISNULL(column\_name, 'N/A') AS column\_alias

FROM table\_name

Neste exemplo, se column\_name for nula em alguma linha da tabela, a função ISNULL substituirá o valor nulo por 'N/A'. Se column\_name não for nula, seu valor original será mantido.

## NULLIF

A função NULLIF no A SQL Server é usada para comparar duas expressões e retornar NULL se as expressões forem iguais. Se as expressões forem diferentes, a função retorna a primeira expressão. A sintaxe básica da função NULLIF é a seguinte:

NULLIF(expression1, expression2)

expression1: A primeira expressão a ser comparada.

expression2: A segunda expressão a ser comparada.

Se expression1 for igual a expression2, a função NULLIF retornará NULL. Caso contrário, ela retornará o valor de expression1.

Aqui está um exemplo de como a função NULLIF pode ser usada:

SELECT NULLIF(column\_name, 0) AS column\_alias

FROM table\_name

Neste exemplo, se column\_name for igual a 0 em alguma linha da tabela, a função NULLIF retornará NULL. Caso contrário, o valor original de column\_name será mantido.

A função NULLIF é útil para lidar com situações em que deseja substituir um valor específico por NULL. Se expression1 e expression2 forem iguais, NULLIF fornece uma maneira conveniente de retornar NULL.