[**https://carbon.now.sh/?bg=rgba%28171%2C+184%2C+195%2C+1%29&t=seti&wt=none&l=sql&width=680&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=5px&ph=5px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=2x&wm=false&code=SELECT%2520CONCAT\_WS%28%27%252C%2520%27%252C%2520Endereco%252C%2520Cidade%252C%2520Estado%29%2520AS%2520EnderecoCompleto%250AFROM%2520dbo.Clientes%253B**](https://carbon.now.sh/?bg=rgba%28171%2C+184%2C+195%2C+1%29&t=seti&wt=none&l=sql&width=680&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=5px&ph=5px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=2x&wm=false&code=SELECT%2520CONCAT_WS%28%27%252C%2520%27%252C%2520Endereco%252C%2520Cidade%252C%2520Estado%29%2520AS%2520EnderecoCompleto%250AFROM%2520dbo.Clientes%253B)

No mundo dos bancos de dados, a manipulação de diferentes tipos de dados é uma tarefa corriqueira. Para realizar essa mágica, o SQL Server oferece duas funções poderosas: **CAST** e **CONVERT**. Ambas permitem transformar um valor de um tipo de dados para outro, abrindo um leque de possibilidades para você.

**CAST: A Transformação Elegante**

Com a sintaxe concisa e elegante CAST (expressao AS tipo\_de\_dados [(comprimento)]), a função CAST se destaca pela sua simplicidade. Imagine que você precisa concatenar um texto com um valor numérico. A função CAST entra em cena para converter o número em texto, permitindo a união harmoniosa entre os diferentes tipos de dados.

**Exemplo:**

SELECT 'O preço total é ' + CAST(Preco AS VARCHAR(10)) AS PrecoTotal

FROM dbo.Produtos;

Neste exemplo, a função CAST transforma o valor numérico da coluna "Preco" em uma string, possibilitando a concatenação com a frase "O preço total é".

**CONVERT: A Versatilidade em Ação**

A função CONVERT, com sua sintaxe CONVERT (tipo\_de\_dados [(comprimento)], expressao [, estilo]), oferece uma gama maior de opções. O parâmetro "estilo" permite controlar o formato da conversão, especialmente útil ao lidar com datas e horas.

**Exemplo:**

SELECT CONVERT(VARCHAR, GETDATE(), 103) AS DataFormatada;

Neste caso, a função CONVERT transforma a data atual, obtida com GETDATE(), em uma string no formato "dd/mm/yyyy", especificado pelo estilo 103.

**Conversões Implícitas: A capa da invisibilidade**

O SQL Server, em sua inteligência, realiza conversões implícitas entre tipos de dados compatíveis, sem a necessidade de usar CAST ou CONVERT. Por exemplo, ao comparar um valor smallint com um int, o SQL Server converte automaticamente o smallint para int antes da comparação.

**Exemplo:**

DECLARE @smallint SMALLINT = 10;

DECLARE @int INT = 20;

IF @smallint < @int

PRINT 'smallint é menor que int';

**Truncando e Arredondando: Lidando com as Sutilezas**

Ao converter dados, é crucial estar atento à possibilidade de truncamento ou arredondamento. Se o tamanho do valor convertido exceder o tamanho do tipo de dados de destino, o SQL Server pode truncar os dados, levando à perda de informações.

**Exemplo:**

DECLARE @decimal DECIMAL(5,2) = 123.45;

DECLARE @int INT;

SET @int = CAST(@decimal AS INT);

SELECT @int;

Neste exemplo, a conversão de decimal para int causa o truncamento da parte decimal, resultando no valor 123.

**Conclusão**

As funções CAST e CONVERT são ferramentas essenciais para qualquer desenvolvedor SQL Server. Dominar suas nuances e compreender o impacto das conversões implícitas é fundamental para garantir a integridade e a precisão dos seus dados. Com este conhecimento, você estará pronto para manipular os dados com maestria, extraindo o máximo do poder do SQL Server.

**TRY\_CAST, TRY\_CONVERT, TRY\_PARSE: Tentativas Seguras de Conversão**

As funções **TRY\_CAST**, **TRY\_CONVERT** e **TRY\_PARSE** tentam converter um valor para um tipo de dados específico. Se a conversão for bem-sucedida, a função retorna o valor convertido. Caso contrário, a função retorna NULL, evitando erros que podem interromper a execução do código.

**Sintaxe:**

TRY\_CAST (expressao AS tipo\_de\_dados)

TRY\_CONVERT (tipo\_de\_dados, expressao [, estilo])

TRY\_PARSE (expressao\_string AS tipo\_de\_dados USING cultura)

**Exemplo:**

SELECT TRY\_CAST('123' AS INT) AS Numero; -- Retorna 123

SELECT TRY\_CAST('abc' AS INT) AS Numero; -- Retorna NULL

**Expandindo o Repertório: Mais Funções Essenciais para Manipulação de Strings**

Além das poderosas funções **CAST** e **CONVERT**, o SQL Server oferece um arsenal de outras funções para manipulação de strings, prontas para levar suas habilidades de manipulação de dados a um novo patamar.

**Concatenando Textos com Maestria: CONCAT**

A função **CONCAT**, com sua sintaxe CONCAT (argumento1, argumento2 [, argumentoN]...), permite unir dois ou mais valores de string, criando uma única string resultante. Imagine que você precisa combinar o nome e o sobrenome de um cliente em um único campo. A função CONCAT entra em ação para realizar essa fusão textual.

**Exemplo:**

SELECT CONCAT(Nome, ' ', Sobrenome) AS NomeCompleto

FROM dbo.Clientes;

Neste exemplo, a função CONCAT une o valor da coluna "Nome", um espaço em branco e o valor da coluna "Sobrenome", gerando o campo "NomeCompleto".

**CONCAT\_WS: Concatenação com Estilo**

A função **CONCAT\_WS** (Concat With Separator), disponível a partir do SQL Server 2017, oferece a funcionalidade de CONCAT com um toque especial: a capacidade de especificar um separador entre os valores concatenados.

**Sintaxe:**

CONCAT\_WS (separador, argumento1, argumento2 [, argumentoN]...)

**Exemplo:**

SELECT CONCAT\_WS(', ', Endereco, Cidade, Estado) AS EnderecoCompleto

FROM dbo.Clientes;

Neste caso, a função CONCAT\_WS une os valores das colunas "Endereco", "Cidade" e "Estado", utilizando uma vírgula e um espaço como separador.

**Delimitando Strings com Precisão: SUBSTRING**

A função **SUBSTRING**, com a sintaxe SUBSTRING (expressao, inicio, comprimento), extrai uma parte específica de uma string. Imagine que você precisa obter os primeiros quatro dígitos de um código postal. A função SUBSTRING permite isolar essa porção da string com facilidade.

**Exemplo:**

SELECT SUBSTRING(CEP, 1, 4) AS PrefixoCEP

FROM dbo.Enderecos;

Neste exemplo, a função SUBSTRING extrai os caracteres da posição 1 até a posição 4 da coluna "CEP", gerando o campo "PrefixoCEP".

**LEN: Revelando o Tamanho da String**

A função **LEN**, com a sintaxe simples LEN (expressao\_string), retorna o número de caracteres de uma string, desconsiderando espaços em branco à direita.

**Exemplo:**

SELECT Nome, LEN(Nome) AS TamanhoNome

FROM dbo.Clientes;

Neste caso, a função LEN calcula o número de caracteres da coluna "Nome", exibindo o resultado no campo "TamanhoNome".

**Formatando com Elegância: FORMAT**

A função **FORMAT**, com a sintaxe FORMAT(valor, formato [, cultura]), formata um valor de acordo com um padrão específico e uma cultura opcional. Essa função é ideal para formatar datas, horas e números de acordo com as necessidades da sua aplicação.

**Exemplo:**

SELECT FORMAT(DataNascimento, 'dd/MM/yyyy') AS DataFormatada

FROM dbo.Clientes;

Neste exemplo, a função FORMAT formata a data da coluna "DataNascimento" no padrão brasileiro "dd/MM/yyyy".

**Removendo Espaços Indesejados: LTRIM e RTRIM**

As funções **LTRIM** e **RTRIM** removem espaços em branco do início e do fim de uma string, respectivamente. Imagine que você precisa limpar um campo de texto que contém espaços extras à esquerda ou à direita. Essas funções entram em cena para garantir que a string esteja formatada corretamente.

**Sintaxe:**

LTRIM (expressao\_string)

RTRIM (expressao\_string)

**Exemplo:**

SELECT LTRIM(RTRIM(Nome)) AS NomeLimpo

FROM dbo.Clientes;

Neste caso, a função LTRIM remove espaços à esquerda e a função RTRIM remove espaços à direita da coluna "Nome", resultando no campo "NomeLimpo" sem espaços extras.

**TRIM: A Ferramenta Completa**

A função **TRIM**, introduzida no SQL Server 2017, oferece uma solução abrangente para remover espaços em branco ou outros caracteres especificados do início, do fim ou de ambos os lados de uma string. Essa função combina a funcionalidade de LTRIM e RTRIM em uma única ferramenta poderosa.

**Sintaxe:**

TRIM ([caracteres FROM] string)

**Exemplo:**

SELECT TRIM('.,! ' FROM ' # test .') AS Resultado

FROM dbo.Tabela;

Neste exemplo, a função TRIM remove os caracteres '.,! ' do início e do fim da string '# test .', resultando na string 'test'.

**Substituindo Caracteres com Precisão: REPLACE**

A função **REPLACE**, com a sintaxe REPLACE (string\_expressao, string\_a\_ser\_substituida, string\_de\_substituicao), substitui todas as ocorrências de uma substring por outra substring dentro de uma string.

**Exemplo:**

SELECT REPLACE(Descricao, 'velho', 'novo') AS DescricaoAtualizada

FROM dbo.Produtos;

Neste exemplo, a função REPLACE substitui todas as ocorrências da palavra "velho" por "novo" na coluna "Descricao", gerando o campo "DescricaoAtualizada".

**TRANSLATE: Substituindo Múltiplos Caracteres**

A função **TRANSLATE**, introduzida no SQL Server 2017, oferece a capacidade de substituir múltiplos caracteres de uma string por outros caracteres correspondentes. Imagine que você precisa substituir todos os caracteres especiais por seus equivalentes sem acentos. A função TRANSLATE torna essa tarefa simples e eficiente.

**Sintaxe:**

TRANSLATE (string\_entrada, caracteres, traducoes)

**Exemplo:**

SELECT TRANSLATE('àáâãä', 'àáâãä', 'aaaaa') AS TextoSemAcento

FROM dbo.Tabela;

Neste exemplo, a função TRANSLATE substitui os caracteres acentuados 'àáâãä' por seus equivalentes sem acento 'aaaaa'.

**Conclusão**

Com este arsenal de funções à sua disposição, você está pronto para enfrentar qualquer desafio de manipulação de strings no SQL Server. Ao combinar o poder das funções CAST e CONVERT com as demais ferramentas apresentadas, você estará apto a transformar, formatar, extrair e manipular strings com maestria, elevando suas habilidades de SQL a um novo patamar.

Gostaríamos de expressar nossa profunda gratidão por você, leitor, que embarcou nesta jornada mágica pelo universo do SQL Server. Assim como Harry Potter precisou de determinação e coragem para desvendar os mistérios de Hogwarts, você demonstrou um espírito explorador ao se aventurar no fascinante mundo da transformação de dados. Este eBook foi criado como um “Mapa do Maroto” para guiar sua trajetória, revelando atalhos, segredos e encantamentos técnicos que tornam o SQL Server uma ferramenta tão poderosa quanto uma varinha das varinhas.

Por fim, agradecemos por permitir que este livro seja parte de sua jornada. Que ele sirva como um \*Pomo de Ouro\*, sempre lembrando que o aprendizado contínuo é a verdadeira vitória em qualquer busca. Que seu caminho seja repleto de insights valiosos, e que sua habilidade com SQL Server brilhe como o feitiço Lumos, iluminando novos horizontes em sua carreira. Até a próxima aventura no mundo da programação – estarei sempre por aqui, pronto para ajudá-lo a conjurar novas soluções!