Parte superior do formulário

**Funções de Conversão e Manipulação de Strings no SQL Server**

Este documento resume os principais temas e ideias extraídos de documentações do Microsoft Learn sobre funções de conversão de dados e manipulação de strings no SQL Server.

**Temas Principais:**

* **Conversão de Tipos de Dados:** O SQL Server oferece funções para converter dados entre diferentes tipos, como CAST e CONVERT.
* **Manipulação de Strings:** Diversas funções facilitam a manipulação de strings, incluindo LEFT, RIGHT, SUBSTRING, LEN, REPLACE, LTRIM, RTRIM, TRIM, CONCAT e CONCAT\_WS.
* **Tratamento de Erros:** Funções como TRY\_CAST, TRY\_CONVERT e TRY\_PARSE permitem a conversão de dados com tratamento de erros, retornando NULL em caso de falha.
* **Formatação de Dados:** Funções como FORMAT e STR permitem formatar dados numéricos e de data/hora de acordo com padrões específicos.

**Ideias e Fatos Importantes:**

**Conversão de Dados:**

* CAST e CONVERT: Ambas convertem uma expressão de um tipo de dados para outro. CONVERT oferece mais opções de estilo para controlar a conversão, especialmente para datas e horas.
* Sintaxe: CAST (expressão AS tipo\_de\_dados [(comprimento)]) e CONVERT (tipo\_de\_dados [(comprimento)], expressão [, estilo])
* Tipos de Dados Suportados: A maioria dos tipos de dados do SQL Server, incluindo xml, bigint e sql\_variant.
* Conversões Implícitas vs. Explícitas: Conversões implícitas ocorrem automaticamente em algumas situações, enquanto as explícitas exigem o uso de funções.
* Precedência de Tipo de Dados: Em expressões mistas, o SQL Server usa regras de precedência para determinar a ordem das conversões.
* Exemplo: DECLARE @string VARCHAR(10); SET @string = 1; SELECT @string + ' is a string.' AS Result (o valor inteiro 1 é convertido implicitamente para VARCHAR).

**Manipulação de Strings:**

* LEFT, RIGHT, SUBSTRING: Extraem partes de uma string.
* Exemplo: SELECT RIGHT('abcdefg', 2); retorna 'fg'.
* LEN: Retorna o número de caracteres em uma string, excluindo espaços à direita.
* Observação: Para obter o tamanho em bytes, use DATALENGTH.
* REPLACE: Substitui todas as ocorrências de uma substring por outra.
* Exemplo: SELECT REPLACE('abcdefghicde','cde','xxx'); retorna 'abxxxfghixxx'.
* LTRIM, RTRIM, TRIM: Removem espaços em branco ou caracteres especificados do início, fim ou ambos os lados de uma string.
* A partir do SQL Server 2022, o argumento characters permite especificar quais caracteres remover.
* CONCAT e CONCAT\_WS: Concatenam strings, sendo que CONCAT\_WS permite especificar um separador.
* CONCAT\_WS ignora NULLs durante a concatenação.

**Tratamento de Erros:**

* TRY\_CAST, TRY\_CONVERT, TRY\_PARSE: Semelhantes a CAST, CONVERT e PARSE, mas retornam NULL em caso de erro de conversão.

**Formatação de Dados:**

* FORMAT: Formata dados de acordo com um padrão e cultura específicos, usando regras do .NET Framework.
* Suporta formatação de números, datas, horas e outros tipos de dados.
* Exemplo: SELECT FORMAT(GETDATE(), 'dd/MM/yyyy', 'pt-BR') AS 'Data';
* STR: Converte dados numéricos em strings, com opções de comprimento e casas decimais.
* Exemplo: SELECT STR(123.45, 6, 2); retorna '123.45'.

**Observações Adicionais:**

* **Compatibilidade com CLR:** Algumas funções, como PARSE e TRY\_PARSE, dependem do .NET Framework CLR e não podem ser usadas em contextos remotos.
* **Níveis de Compatibilidade:** Algumas funcionalidades, como o argumento characters em TRIM, exigem um nível de compatibilidade de banco de dados específico.
* **Ordenação:** Funções de string como REPLACE consideram a ordenação dos dados. É possível usar COLLATE para especificar uma ordenação diferente.

**Recursos Adicionais:**

* Documentação completa das funções no Microsoft Learn.
* Exemplos de código para cada função.
* Diagramas de conversão de tipo de dados.
* Tabelas de mapeamento de culturas do SQL Server para o .NET Framework.

**Conclusão:**

As funções de conversão de dados e manipulação de strings são ferramentas essenciais no SQL Server, permitindo transformar e formatar dados para atender às necessidades de diversas aplicações. É importante entender as diferentes funções, suas opções e o tratamento de erros para garantir resultados precisos e eficientes.

Parte inferior do formulário

convert\_to\_textTransformar em fonte

Aqui está um cartão de referência para as funções de conversão e cadeia de caracteres disponíveis no SQL Server:

**Funções de Conversão de Tipo de Dados**

* **CAST**: Converte uma expressão de um tipo de dados para outro.
  + Sintaxe: CAST ( expressao AS tipo\_de\_dados [ ( comprimento ) ] )
* **CONVERT**: Converte uma expressão de um tipo de dados para outro, com opções de estilo adicionais.
  + Sintaxe: CONVERT ( tipo\_de\_dados [ ( comprimento ) ] , expressao [ , estilo ] )
* **TRY\_CAST**: Tenta converter um valor para um tipo de dados especificado. Retorna o valor convertido se a conversão for bem-sucedida, caso contrário, retorna NULL.
  + Sintaxe: TRY\_CAST ( expressao AS tipo\_de\_dados [ ( comprimento ) ] )
* **TRY\_CONVERT**: Tenta converter um valor para um tipo de dados especificado. Retorna o valor convertido se a conversão for bem-sucedida, caso contrário, retorna NULL.
  + Sintaxe: TRY\_CONVERT ( tipo\_de\_dados [ ( comprimento ) ] , expressao [ , estilo ] )
* **PARSE**: Converte um valor de cadeia de caracteres em um tipo de dados especificado, usando uma cultura opcional.
  + Sintaxe: PARSE ( valor\_cadeia\_caracteres AS tipo\_de\_dados [ USING cultura ] )
* **TRY\_PARSE**: Tenta converter um valor de cadeia de caracteres em um tipo de dados especificado, usando uma cultura opcional. Retorna o valor convertido se a conversão for bem-sucedida, caso contrário, retorna NULL.
  + Sintaxe: TRY\_PARSE ( valor\_cadeia\_caracteres AS tipo\_de\_dados [ USING cultura ] )

**Funções de Cadeia de Caracteres**

* **CONCAT**: Concatena dois ou mais valores de cadeia de caracteres.
  + Sintaxe: CONCAT ( argumento1 , argumento2 [ , argumentoN ] ... )
* **CONCAT\_WS**: Concatena dois ou mais valores de cadeia de caracteres, separando-os com um delimitador especificado.
  + Sintaxe: CONCAT\_WS ( separador , argumento1 , argumento2 [ , argumentoN ] ... )
* **FORMAT**: Retorna um valor formatado com o formato e a cultura opcional especificados.
  + Sintaxe: FORMAT( valor , formato [ , cultura ] )
* **LEFT**: Retorna os *n* caracteres mais à esquerda de uma cadeia de caracteres.
  + Sintaxe: LEFT ( expressao\_caracteres , expressao\_inteiro )
* **LEN**: Retorna o número de caracteres de uma cadeia de caracteres, excluindo espaços à direita.
  + Sintaxe: LEN ( expressao\_cadeia\_caracteres )
* **LTRIM**: Remove espaços à esquerda de uma cadeia de caracteres.
  + Sintaxe: LTRIM ( expressao\_caracteres [ , caracteres ] )
* **REPLACE**: Substitui todas as ocorrências de uma cadeia de caracteres especificada por outra cadeia de caracteres.
  + Sintaxe: REPLACE ( expressao\_cadeia\_caracteres , padrao\_cadeia\_caracteres , substituicao\_cadeia\_caracteres )
* **RIGHT**: Retorna os *n* caracteres mais à direita de uma cadeia de caracteres.
  + Sintaxe: RIGHT ( expressao\_caracteres , expressao\_inteiro )
* **RTRIM**: Remove espaços à direita de uma cadeia de caracteres.
  + Sintaxe: RTRIM ( expressao\_caracteres [ , caracteres ] )
* **STR**: Converte um valor numérico em uma cadeia de caracteres.
  + Sintaxe: STR ( expressao\_float [ , comprimento [ , decimal ] ] )
* **SUBSTRING**: Retorna uma parte de uma cadeia de caracteres.
  + Sintaxe: SUBSTRING ( expressao , inicio , comprimento )
* **TRANSLATE**: Retorna uma cadeia de caracteres após converter caracteres especificados em um conjunto de caracteres de destino especificado.
  + Sintaxe: TRANSLATE ( cadeia\_entrada, caracteres, traducoes )
* **TRIM**: Remove espaços em branco ou outros caracteres especificados do início e do final de uma cadeia de caracteres.
  + Sintaxe: TRIM ( [ INICIAL | FINAL | AMBOS ] [caracteres DE] cadeia\_caracteres )