

Funciones SQL Server

Funciones de Cadena

Función	Descripción
ASCII (<i>expression</i>)	<p>Devuelve el valor de código ASCII del carácter situado más a la izquierda de una expresión de cadena.</p> <p>Argumentos <i>expression</i>: cualquier expresión válida de un tipo String ASCII.</p> <p>Valor devuelto Valor de tipo Int32.</p> <p>Ejemplo <code>ASCII('A')</code></p>
CHAR (<i>expression</i>)	<p>Convierte un código Int32 en una cadena ASCII.</p> <p>Argumentos <i>expression</i>: valor de tipo Int32.</p> <p>Valor devuelto Valor de tipo String ASCII.</p> <p>Ejemplo <code>char(97)</code></p>
CHARINDEX (<i>expression1</i> , <i>expression2</i> [, <i>start_location</i>])	<p>Devuelve la posición inicial de la expresión especificada en una cadena de caracteres.</p> <p>Argumentos <i>expression1</i>: expresión que contiene la secuencia de caracteres que se va a buscar. La expresión puede ser de un tipo String (ASCII o Unicode) o Binary. <i>expression2</i>: expresión, que normalmente es una columna, en la que se encuentra la secuencia especificada. La expresión puede ser de un tipo String (ASCII o Unicode) o Binary. <i>start_location</i>: (opcional) valor de tipo Int64 (no se devuelve en SQL Server 2000) o Int32 que representa la posición de caracteres en la que debe comenzar la búsqueda de <i>expression1</i> en <i>expression2</i>. Si no se especifica <i>start_location</i>, es un número negativo o es igual a cero, la búsqueda comienza al principio de <i>expression2</i>.</p> <p>Valor devuelto Valor de tipo Int32.</p> <p>Ejemplo <code>CHARINDEX('h', 'habcdefgh', 2)</code></p>

DIFFERENCE (<i>expression</i> , <i>expression</i>)	<p>Compara los valores de SOUNDEX de dos cadenas y evalúa la similitud entre ambas.</p> <p>Argumentos Tipo String Unicode o ASCII. <i>expression</i> puede ser una constante, una variable o una columna.</p> <p>Valor devuelto Devuelve un valor de tipo Int32 que representa la diferencia entre los valores de SOUNDEX de dos expresiones de caracteres. El intervalo está comprendido entre 0 y 4. El valor 0 indica una similitud escasa o inexistente, y el valor 4 indica una elevada similitud o que los valores son iguales.</p> <p>Ejemplo <pre>// The following example returns a DIFFERENCE value of 4, //the least possible difference or the best match. DIFFERENCE('Green','Greene');</pre></p>
LEFT (<i>expression</i> , <i>count</i>)	<p>Devuelve la parte izquierda de una cadena de caracteres con el número de caracteres especificado.</p> <p>Argumentos <i>expression</i>: tipo String Unicode o ASCII. Use la función CAST para convertir character_expression explícitamente. <i>count</i>: valor de tipo Int64 (no se devuelve en SQL Server 2000) o Int32 que especifica cuántos caracteres de character_expression se devolverán.</p> <p>Valor devuelto Valor String Unicode o ASCII.</p> <p>Ejemplo <pre>LEFT('SQL Server', 4)</pre></p>
LEN (<i>expression</i>)	<p>Devuelve el número de caracteres de la expresión de cadena especificada, excluidos los espacios en blanco finales.</p> <p>Argumentos <i>expression</i>: expresión de un tipo String (ASCII o Unicode) o un tipo Binary.</p> <p>Valor devuelto Valor de tipo Int32.</p> <p>Ejemplo <pre>LEN('abcd')</pre></p>
LOWER (<i>expression</i>)	<p>Devuelve una expresión de String después de convertir a minúsculas los datos de caracteres en mayúsculas.</p> <p>Argumentos <i>expression</i>: cualquier expresión válida del tipo String.</p> <p>Valor devuelto Valor de tipo String.</p> <p>Ejemplo</p>

	<p><code>LOWER('AbB')</code></p>
<p>LTRIM(<i>expression</i>)</p>	<p>Devuelve una expresión String tras quitar los espacios iniciales en blanco.</p> <p>Argumentos <i>expression</i>: cualquier expresión válida del tipo String.</p> <p>Valor devuelto Un objeto String.</p> <p>Ejemplo <code>LTRIM(' d')</code></p>
<p>NCHAR(<i>expression</i>)</p>	<p>Devuelve el valor de tipo String Unicode correspondiente al código entero dado, tal como se define en el estándar Unicode.</p> <p>Argumentos <i>expression</i>: valor de tipo Int32.</p> <p>Valor devuelto Valor de tipo String Unicode.</p> <p>Ejemplo <code>NCHAR(65)</code></p>
<p>PATINDEX('%pattern %',<i>expression</i>)</p>	<p>Devuelve la posición inicial de la primera aparición de un patrón en una expresión String especificada.</p> <p>Argumentos '%pattern%': valor de tipo String ASCII o Unicode. Se pueden utilizar caracteres comodín; no obstante, el carácter % debe ir delante y detrás del patrón (excepto cuando se busque el primer o último carácter).</p> <p><i>expression</i>: cadena de tipo String ASCII o Unicode en la que buscar el patrón especificado.</p> <p>Valor devuelto Valor de tipo Int32.</p> <p>Ejemplo <code>PATINDEX('abc', 'ab')</code></p>
<p>QUOTENAME('char_string' [, 'quote_char'])</p>	<p>Devuelve un valor de tipo String Unicode con los delimitadores agregados para convertirla en un identificador delimitado válido de SQL Server 2005.</p> <p>Argumentos <i>char_string</i>: valor de tipo String Unicode. <i>quote_char</i>: cadena de un solo carácter que se utiliza como delimitador. Puede ser una comilla simple ('), un corchete izquierdo o derecho ([]) o una comilla doble ("). Si no se especifica <i>quote_char</i>, se utilizarán corchetes.</p> <p>Valor devuelto Valor de tipo String Unicode.</p> <p>Ejemplo <code>QUOTENAME('abc[]def')</code></p>

<p>REPLACE(<i>expression1</i>,<i>expression2</i>, <i>expression3</i>)</p>	<p>Repite una expresión de carácter un número especificado de veces.</p> <p>Argumentos <i>expression1</i>: expresión de cadena que se va a buscar. <i>expression1</i> puede ser de un tipo String Unicode o ASCII. <i>expression2</i>: subcadena que se va a buscar. <i>expression2</i> puede ser de un tipo String Unicode o ASCII. <i>expression3</i>: cadena de reemplazo. <i>expression3</i> puede ser de un tipo String Unicode o ASCII.</p> <p>Ejemplo REPLACE('aabbcc', 'bc', 'zz')</p>
<p>REPLICATE(<i>char_expression</i>, <i>int_expression</i>)</p>	<p>Repite una expresión de carácter un número especificado de veces.</p> <p>Argumentos <i>char_expression</i>: tipo String Unicode o ASCII. <i>int_expression</i>: Int64 (no se admite en SQL Server 2000) o Int32.</p> <p>Valor devuelto Tipo String Unicode o ASCII.</p> <p>Ejemplo REPLICATE('aa', 2)</p>
<p>REVERSE(<i>expression</i>)</p>	<p>Devuelve un valor de tipo String Unicode o ASCII con las posiciones de los caracteres invertidas con respecto a la cadena de entrada.</p> <p>Argumentos <i>expression</i>: tipo String Unicode o ASCII.</p> <p>Valor devuelto Tipo String Unicode o ASCII.</p> <p>Ejemplo REVERSE('abcd')</p>
<p>RIGHT(<i>char_expression</i>, <i>count</i>)</p>	<p>Devuelve la parte derecha de una cadena de caracteres con el número de caracteres especificado.</p> <p>Argumentos <i>char_expression</i>: tipo String Unicode o ASCII. Use la función CAST para convertir <i>character_expression</i> explícitamente. <i>count</i>: valor de tipo Int64 (no se devuelve en SQL Server 2000) o Int32 que especifica cuántos caracteres de <i>character_expression</i> se devolverán.</p> <p>Valor devuelto Un tipo String ASCII.</p> <p>Ejemplo RIGHT('SQL Server', 6)</p>
<p>RTRIM(<i>expression</i>)</p>	<p>Devuelve un valor de tipo String Unicode o ASCII después de quitar los espacios finales.</p> <p>Argumentos</p>

	<p><i>expression</i>: tipo String Unicode o ASCII.</p> <p>Valor devuelto Tipo String Unicode o ASCII.</p> <p>Ejemplo <code>RTRIM(' d e ')</code></p>
SOUNDEX (<i>expression</i>)	<p>Devuelve un código de cuatro caracteres (SOUNDEX) para evaluar la similitud de dos cadenas.</p> <p>Argumentos <i>expression</i>: tipo String Unicode o ASCII.</p> <p>Valor devuelto Valor de tipo String ASCII. Un código de cuatro caracteres (SOUNDEX) es una cadena que evalúa la semejanza de dos cadenas.</p> <p>Ejemplo <code>Select SOUNDEX('Smith'), SOUNDEX('Smythe') FROM {1}</code></p> <p>Valores devueltos ----- S530 S530</p>
SPACE (<i>int_expression</i>)	<p>Devuelve un valor de tipo String ASCII de espacios repetidos.</p> <p>Argumentos <i>int_expression</i>: valor de tipo Int64 (no se devuelve en SQL Server 2000) o Int32 que indica el número de espacios.</p> <p>Valor devuelto Valor de tipo String ASCII.</p> <p>Ejemplo <code>SPACE(2)</code></p>
STR (<i>float_expression</i> [, <i>length</i> [, <i>decimal</i>]])	<p>Devuelve un valor String ASCII convertido a partir de datos numéricos.</p> <p>Argumentos <i>float_expression</i>: expresión de un tipo de datos (Double) numérico aproximado con un separador decimal. <i>length</i>: (opcional) valor de tipo Int32 que representa la longitud total. Incluye el separador decimal, el signo, los dígitos y los espacios. El valor predeterminado es 10. <i>decimal</i>: (opcional) valor de tipo Int32 que representa el número de lugares a la derecha del separador decimal. decimal debe ser menor o igual que 16. Si decimal es mayor que 16, el resultado se trunca a dieciséis lugares a la derecha del separador decimal.</p> <p>Valor devuelto Valor de tipo String ASCII.</p> <p>Ejemplo <code>STR(212.0)</code></p>
STUFF (<i>str_expression</i> , <i>start</i> , <i>length</i> , <i>str_expression_to_ins</i>)	<p>Elimina una cantidad especificada de caracteres e inserta otro juego de caracteres a partir del punto inicial</p>

<p>ert)</p>	<p>especificado de una expresión de cadena.</p> <p>Argumentos <i>str_expression</i>: tipo String Unicode o ASCII. <i>start</i>: valor Int64 (no se devuelve en SQL Server 2000) o Int32 que especifica la ubicación al principio de la eliminación e inserción. <i>length</i>: valor Int64 (no se devuelve en SQL Server 2000) o Int32 que especifica el número de caracteres que se van a eliminar. <i>str_expression_to_insert</i>: valor String Unicode o ASCII.</p> <p>Valor devuelto Valor String Unicode o ASCII.</p> <p>Ejemplo <code>STUFF('abcd', 2, 2, 'zz')</code></p>
<p>SUBSTRING(<i>str_expression</i>, <i>start</i>, <i>length</i>)</p>	<p>Devuelve parte de una expresión String.</p> <p>Argumentos <i>str_expression</i>: expresión de un tipo String (ASCII o Unicode) o un tipo Binary. <i>start</i>: valor de tipo Int64 (no se devuelve en SQL Server 2000) o Int32 que especifica dónde comienza la subcadena. 1 se refiere al primer carácter de la cadena. <i>length</i>: valor Int64 (no se devuelve en SQL Server 2000) o Int32 que especifica cuántos caracteres de la expresión se devolverán.</p> <p>Valor devuelto Valor de tipo String (ASCII o Unicode) o Binary.</p> <p>Ejemplo <code>SUBSTRING('abcd', 2, 2)</code></p>
<p>UNICODE(<i>expression</i>)</p>	<p>Devuelve el valor entero, según la definición del estándar Unicode, para el primer carácter de la expresión de entrada.</p> <p>Argumentos <i>expression</i>: valor String Unicode.</p> <p>Valor devuelto Valor de tipo Int32.</p> <p>Ejemplo <code>UNICODE('a')</code></p>
<p>UPPER(<i>expression</i>)</p>	<p>Devuelve una expresión String después de convertir a mayúsculas los datos de caracteres que están en minúsculas.</p> <p>Argumentos <i>expression</i>: expresión de un tipo String ASCII o Unicode.</p> <p>Valor devuelto Tipo String ASCII o Unicode.</p> <p>Ejemplo <code>UPPER('AbB')</code></p>

Funciones Matemáticas

Función	Descripción
ABS(<i>expression</i>)	<p>Lleva a cabo la función que devuelve el valor absoluto.</p> <p>Argumentos <i>expression</i>: valor de tipo Int32, Int64, Double o Decimal.</p> <p>Valor devuelto Valor absoluto de la expresión especificada.</p> <p>Ejemplo ABS(-2)</p>
ACOS(<i>expression</i>)	<p>Devuelve el valor del arcocoseno de la expresión especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo ACOS(.9)</p>
ASIN(<i>expression</i>)	<p>Devuelve el valor del arcoseno de la expresión especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Objeto Double.</p> <p>Ejemplo ASIN(.9)</p>
ATAN(<i>expression</i>)	<p>Devuelve el valor del arcotangente de la expresión numérica especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo ATAN(9)</p>
ATN2(<i>expression,expression</i>)	<p>Devuelve el ángulo, en radianes, cuya tangente se encuentra entre las dos expresiones numéricas especificadas.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo ATN2(9, 8)</p>

CEILING (<i>expression</i>)	<p>Convierte la expresión especificada al número entero más pequeño mayor o igual que él.</p> <p>Argumentos <i>expression</i>: valor de tipo Int32, Int64, Double o Decimal.</p> <p>Valor devuelto Valor de tipo Int32, Int64, Double o Decimal.</p> <p>Ejemplo SQL SELECT VALUE product FROM AdventureWorksEntities.Products AS product WHERE product.ListPrice = CEILING(product.ListPrice)</p>
COS (<i>expression</i>)	<p>Calcula el coseno trigonométrico del ángulo especificado, en radianes.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo COS(45)</p>
COT (<i>expression</i>)	<p>Calcula la cotangente trigonométrica del ángulo especificado, en radianes.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo COT(60)</p>
DEGREES (<i>radians</i>)	<p>Devuelve el ángulo correspondiente en grados.</p> <p>Argumentos <i>expression</i>: valor de tipo Int32, Int64, Double o Decimal.</p> <p>Valor devuelto Valor de tipo Int32, Int64, Double o Decimal.</p> <p>Ejemplo DEGREES(3.1)</p>
EXP (<i>expression</i>)	<p>Calcula el valor exponencial de la expresión numérica especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo</p>

	EXP(1)
FLOOR (<i>expression</i>)	<p>Convierte la expresión especificada al número entero más grande que sea menor o igual que ella.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo SQL SELECT VALUE product FROM AdventureWorksEntities.Products AS product WHERE product.ListPrice = FLOOR(product.ListPrice)</p>
LOG (<i>expression</i>)	<p>Calcula el logaritmo natural de la expresión float especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo LOG(100)</p>
LOG10 (<i>expression</i>)	<p>Devuelve el logaritmo en base 10 de la expresión Double especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Objeto Double.</p> <p>Ejemplo LOG10(100)</p>
PI()	<p>Devuelve el valor constante de Pi como un Double.</p> <p>Valor devuelto Objeto Double.</p> <p>Ejemplo PI()</p>
POWER (<i>numeric_expression</i> , <i>power_expression</i>)	<p>Calcula el valor de la expresión especificada elevada a la potencia indicada.</p> <p>Argumentos <i>numeric_expression</i>: valor de tipo Int32, Int64, Double o Decimal. <i>power_expression</i>: valor de tipo Double que representa la potencia a la que se va a elevar <i>numeric_expression</i>.</p> <p>Valor devuelto Valor de la <i>numeric_expression</i> especificada a la</p>

	<p><i>power_expression</i> especificada.</p> <p>Ejemplo</p> <p>POWER(2,7)</p>
RADIANS (<i>expression</i>)	<p>Convierte grados en radianes.</p> <p>Argumentos</p> <p><i>expression</i>: valor de tipo Int32, Int64, Double o Decimal.</p> <p>Valor devuelto</p> <p>Valor de tipo Int32, Int64 Double o Decimal .</p> <p>Ejemplo</p> <p>RADIANS(360.0)</p>
RAND ([inicialización])	<p>Devuelve un valor aleatorio de 0 a 1.</p> <p>Argumentos</p> <p>Retruns el valor de inicialización como Int32. Si la inicialización no se especifica, el motor de base de datos de SQL Server asigna uno de forma aleatoria. Para un valor de inicialización especificado, el resultado devuelto es siempre el mismo.</p> <p>Valor devuelto</p> <p>Valor Double aleatorio de 0 a 1.</p> <p>Ejemplo</p> <p>RAND()</p>
ROUND (<i>numeric_expression</i> , <i>length</i> [, <i>function</i>])	<p>Devuelve una expresión numérica, redondeada a la longitud o precisión especificadas.</p> <p>Argumentos</p> <p><i>numeric_expression</i>: valor de tipo Int32, Int64, Double o Decimal.</p> <p><i>length</i>: Valor de tipo Int32 que representa la precisión a la que se va a redondear <i>numeric_expression</i>. Si <i>length</i> es un número positivo, <i>numeric_expression</i> se redondea al número de posiciones decimales que especifica <i>length</i>. Si <i>length</i> es un número negativo, <i>numeric_expression</i> se redondea a la izquierda del separador decimal, según se especifica en <i>length</i>.</p> <p><i>function</i>: (opcional) Valor Int32 que representa el tipo de operación que se va a realizar. Si la función se omite o tiene el valor 0 (predeterminado), <i>numeric_expression</i> se redondea. Si se especifica un valor distinto de 0, <i>numeric_expression</i> se trunca.</p> <p>Valor devuelto</p> <p>Valor de la <i>numeric_expression</i> especificada a la <i>power_expression</i> especificada.</p> <p>Ejemplo</p> <p>ROUND(748.58, -3)</p>

SIGN (<i>expression</i>)	<p>Devuelve el signo positivo (+1), cero (0) o negativo (-1) de la expresión especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Int32, Int64, Double o Decimal</p> <p>Valor devuelto Valor de tipo Int32, Int64, Double o Decimal.</p> <p>Ejemplo SIGN(-10)</p>
SIN (<i>expression</i>)	<p>Calcula el seno trigonométrico de un ángulo especificado, en radianes, y devuelve una expresión de tipo Double.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo SIN(20)</p>
SQRT (<i>expression</i>)	<p>Devuelve la raíz cuadrada de la expresión especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo SQRT(3600)</p>
SQUARE (<i>expression</i>)	<p>Devuelve el cuadrado de la expresión especificada.</p> <p>Argumentos <i>expression</i>: valor de tipo Double.</p> <p>Valor devuelto Valor de tipo Double.</p> <p>Ejemplo SQUARE(25)</p>
TAN (<i>expression</i>)	<p>Calcula la tangente de una expresión especificada.</p> <p>Argumentos <i>expression</i>: Double</p> <p>Valor devuelto Double</p> <p>Ejemplo TAN(45.0)</p>

Funciones de fecha y hora

Función	Descripción
DATEADD (<i>datepart,number, date</i>)	<p>Devuelve un valor DateTime nuevo que resulta de sumar un intervalo a la fecha especificada.</p> <p>Argumentos</p> <p><i>datepart</i>: valor de tipo String que representa qué parte de la fecha se devuelve como el valor nuevo.</p> <p><i>number</i>: valor de tipo Int32, Int64, Decimal o Double que se usa para incrementar el valor de <i>datepart</i>.</p> <p><i>date</i>: expresión que devuelve un valor de tipo DateTime, DateTimeOffset o Time con precisión = [0-7], o una cadena de caracteres en un formato de fecha.</p> <p>Valor devuelto</p> <p>Valor de tipo DateTime, DateTimeOffset o Time nuevo con precisión = [0-7].</p> <p>Ejemplo</p> <pre>DATEADD('day', 22, cast('6/9/2006' as DateTime))</pre>
DATEDIFF (<i>datepart,startdate, enddate</i>)	<p>Devuelve el número de límites de fecha y hora entre dos fechas especificadas.</p> <p>Argumentos</p> <p><i>datepart</i>: String que representa la parte de la fecha para calcular la diferencia.</p> <p><i>startdate</i>: la fecha de comienzo para el cálculo es una expresión que devuelve un valor de tipo DateTime, DateTimeOffset o Time con precisión = [0-7], o una cadena de caracteres en formato de fecha.</p> <p><i>enddate</i>: : la fecha de fin para el cálculo es una expresión que devuelve un valor de tipo DateTime, DateTimeOffset o Time con precisión = [0-7], o una cadena de caracteres en formato de fecha.</p> <p>Valor devuelto</p> <p>Valor de tipo Int32.</p> <p>Ejemplo</p> <pre>DATEDIFF('day', cast('6/9/2006' as DateTime), cast('6/20/2006' as DateTime))</pre>
DATENAME (<i>datepart,date</i>)	<p>Devuelve una cadena de caracteres que representa el <i>datepart</i> especificado de la fecha especificada.</p> <p>Argumentos</p> <p><i>datepart</i>: valor de tipo String que representa qué parte de la fecha se devuelve como el valor nuevo.</p> <p><i>date</i>: expresión que devuelve un valor de tipo</p>

	<p>DateTime,DateTimeOffset o Time con precisión = [0-7], o una cadena de caracteres en un formato de fecha.</p> <p>Valor devuelto</p> <p>La cadena de caracteres que representa el datepart especificado de la fecha especificada.</p> <p>Ejemplo</p> <pre>DATENAME('year', cast('6/9/2006' as DateTime))</pre>
DATEPART(datepart,date)	<p>Devuelve un número entero que representa el datepart especificado de la fecha dada.</p> <p>Argumentos</p> <p><i>datepart</i>: valor de tipo String que representa qué parte de la fecha se devuelve como el valor nuevo.</p> <p><i>date</i>: expresión que devuelve un valor de tipo DateTime,DateTimeOffset, o Time con precisión = [0-7], o una cadena de caracteres en un formato de fecha.</p> <p>Valor devuelto</p> <p>El datepart especificado de la fecha especificada como un valor de tipo Int32.</p> <p>Ejemplo</p> <pre>DATEPART('year', cast('6/9/2006' as DateTime))</pre>
DAY(date)	<p>Devuelve el día de la fecha especificada como un número entero.</p> <p>Argumentos</p> <p><i>date</i>:expresión de tipo DateTime o DateTimeOffset con precisión = 0-7.</p> <p>Valor devuelto</p> <p>Día de la fecha especificada como un valor de tipo Int32.</p> <p>Ejemplo</p> <pre>DAY(cast('6/9/2006' as DateTime))</pre>
GETDATE()	<p>Genera la fecha y hora actuales en el formato interno de SQL Server para los valores datetime.</p> <p>Valor devuelto</p> <p>La fecha y hora actuales del sistema como DateTime con una precisión de 3.</p> <p>Ejemplo</p> <pre>GETDATE()</pre>
GETUTCDATE()	<p>Devuelve el valor datetime en formato de hora universal coordinada (UTC) o del meridiano de Greenwich.</p> <p>Valor devuelto</p> <p>El valor DateTime con una precisión de 3 en formato UTC.</p>

	<p>Ejemplo GETUTCDATE()</p>
MONTH(date)	<p>Devuelve el mes de la fecha especificada como un número entero.</p> <p>Argumentos <i>date</i>:expresión de tipo DateTime o DateTimeOffset con precisión = 0-7.</p> <p>Valor devuelto Mes de la fecha especificada como un Int32.</p> <p>Ejemplo MONTH(cast('6/9/2006' as DateTime))</p>
YEAR(date)	<p>Devuelve el año de la fecha especificada como un número entero.</p> <p>Argumentos <i>date</i>:expresión de tipo DateTime o DateTimeOffset con precisión = 0-7.</p> <p>Valor devuelto Año de la fecha especificada como un valor de tipo Int32.</p> <p>Ejemplo YEAR(cast('6/9/2006' as DateTime))</p>
SYSDATETIME()	<p>Devuelve un valor DateTime con una precisión de 7.</p> <p>Valor devuelto Valor DateTime con una precisión de 7.</p> <p>Ejemplo SYSDATETIME()</p>
SYSUTCDATE()	<p>Devuelve el valor datetime en formato de hora universal coordinada (UTC) o del meridiano de Greenwich.</p> <p>Valor devuelto Valor DateTime con una precisión = 7 en formato UTC.</p> <p>Ejemplo SYSUTCDATE()</p>
SYSDATETIMEOFFSET()	<p>Devuelve un valor DateTimeOffset con una precisión de 7.</p> <p>Valor devuelto Valor DateTimeOffset con una precisión de 7 en formato UTC.</p> <p>Ejemplo SYSDATETIMEOFFSET()</p>

Los valores de datepart pueden ser: year (año), quarter (cuarto), month (mes), day (día), week (semana), hour (hora), minute (minuto), second (segundo), millisecond (milisegundo), dw (día de la semana).

<i>datepart</i>	Abreviaturas
year	yy , yyyy
quarter	qq , q
month	mm , m
dayofyear	dy , y
day	dd , d
week	wk , ww
weekday	dw
hour	hh
minute	mi, n
second	ss , s
millisecond	ms
microsecond	mcs
nanosecond	ns
TZoffset	tz
ISO_WEEK	isowk , isoww

Estilos de fecha y hora para convert

CONVERT (data_type [(length)] , expression [, style])

Para campos FechaHora:

CONVERT (varchar , campoFechaHora , style])

Cuando *expression* es un tipo de datos de fecha u hora, *style* puede ser uno de los valores que se muestran en la tabla siguiente. Otros valores se procesan como 0. SQL Server admite el formato de fecha en estilo árabe mediante el algoritmo kuwaití.

Sin el siglo (aa) ⁽¹⁾	Con el siglo (aaaa)	Estándar	Entrada/salida ⁽³⁾
-	0 o 100 ^(1,2)	Predeterminado	mes dd aaaa hh:mia.m. (o p.m.)
1	101	EE. UU.	1 = mm/dd/aa 101 = mm/dd/aaaa
2	102	ANSI	2 = aa.mm.dd 102 = aaaa.mm.dd
3	103	Británico/ Francés	3 = dd/mm/aa 103 = dd/mm/aaaa
4	104	Alemán	4 = dd.mm.aa 104 = dd.mm.aaaa
5	105	Italiano	5 = dd-mm-aa

			105 = dd-mm-aaaa
6	106 ⁽¹⁾	-	6 = dd mes aa 106 = dd mes aaaa
7	107 ⁽¹⁾	-	7 = Mes dd, aa 107 = Mes dd, aaaa
8	108	-	hh:mi:ss
-	9 ó 109 ^(1,2)	Valor predeterminado + milisegundos	mes dd aaaa hh:mi:ss:mmma.m. (o p.m.)
10	110	EE. UU.	10 = mm-dd-aa 110 = mm-dd-aaaa
11	111	JAPÓN	11 = aa/mm/dd 111 = aaaa/mm/dd
12	112	ISO	12 = aammdd 112 = aaaammdd
-	13 ó 113 ^(1,2)	Europeo predeterminado + milisegundos	dd mes aaaa hh:mi:ss:mmm(24h)
14	114	-	hh:mi:ss:mmm(24h)
-	20 ó 120 ⁽²⁾	ODBC canónico	aaaa-mm-dd hh:mi:ss(24h)

-	21 ó 121⁽²⁾	ODBC canónico (con milisegundos)	aaaa-mm-dd hh:mi:ss.mmm(24h)
-	126⁽⁴⁾	ISO8601	aaaa-mm-ddThh:mi:ss.mmm (sin espacios) <div> <p>Nota</p> <p>Cuando el valor para milisegundos (mmm) es 0, no se muestra el valor de milisegundos. Por ejemplo, el valor '2012-11-07T18:26:20.000' se muestra como '2012-11- 07T18:26:20'.</p> </div>
-	127^(6,7)	ISO8601 con zona horaria Z.	aaaa-mm-ddThh:mi:ss.mmmZ (sin espacios) <div> <p>Nota</p> <p>Cuando el valor para milisegundos (mmm) es 0, no se muestra el valor de milisegundos. Por ejemplo, el valor '2012-11- 07T18:26:20.000' se muestra como '2012-11-07T18:26:20'.</p> </div>
-	130^(1,2)	Hijri ⁽⁵⁾	dd mes aaaa hh:mi:ss:mmma.m.
-	131⁽²⁾	Hijri ⁽⁵⁾	dd/mm/aa hh:mi:ss:mmma.m.

¹ Estos valores de estilo devuelven resultados no deterministas. Incluye todos los estilos (aa) (sin el siglo) y un subconjunto de estilos (aaaa) (con el siglo).

² Los valores predeterminados (*style* **0** o **100**, **9** o **109**, **13** o **113**, **20** o **120** y **21** o **121**) siempre devuelven el siglo (aaaa).

³ Entrada cuando se convierte en **datetime**; salida cuando se convierte en datos de caracteres.

CAST y CONVERT

Sintaxis para CAST:

CAST (expression AS data_type [(length)])

Sintaxis para CONVERT:

CONVERT (data_type [(length)] , expression [, style])

A:																												
De:	binary	varbinary	char	varchar	nchar	nvarchar	datetime	smalldatetime	decimal	numeric	float	real	bigint	int(INT4)	smallint(INT2)	tinyint(INT1)	money	smallmoney	bit	timestamp	uniqueidentifier	image	ntext	text	sql_variant	xml	CLR UDT	
binary																												
varbinary																												
char																												
varchar																												
nchar																												
nvarchar																												
datetime																												
smalldatetime																												
decimal																												
numeric																												
float																												
real																												
bigint																												
int(INT4)																												
smallint(INT2)																												
tinyint(INT1)																												
money																												
smallmoney																												
bit																												
timestamp																												
uniqueidentifier																												
image																												
ntext																												
text																												
sql_variant																												
xml																												
CLR UDT																												

● Conversión explícita

○ Conversión implícita

○ No se permite la conversión

★ Requiere una función CAST explícita para evitar la pérdida de precisión o escala que puede producirse en una conversión implícita.

● Las conversiones implícitas entre tipos de datos de XML se admiten sólo si el origen o el destino es xml sin tipo. De lo contrario, deben ser explícitas.

Las conversiones implícitas son aquellas conversiones que tienen lugar sin especificar las funciones CAST o CONVERT. El SQL Server las intenta realizar cada vez que los datos no casan. Las conversiones explícitas son las realizadas a través de las funciones cast y convert.

Ejemplos:

```
CAST(ListPrice AS int)
SELECT 'The list price is ' + CAST(ListPrice AS varchar(12)) AS ListPrice
FROM Production.Product
SELECT CAST(10.6496 AS int)
SELECT CAST(10.3496847 AS money)
```