



UNIVERSIDAD TÉCNOLOGICA NACIONAL
FACULTAD REGIONAL GENERAL PACHECO

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

Base de Datos II

Teoría de Funciones

ING. ARIEL HERRERA



Contenido

Funciones	2
Funciones matemáticas	3
Funciones de Cadena.....	5
Funciones de fecha y hora.....	7



Funciones

Una función es un conjunto de sentencias que operan como una unidad lógica.

Una función tiene un nombre, retorna un parámetro de salida y opcionalmente acepta parámetros de entrada. Las funciones de SQL Server no pueden ser modificadas, las funciones definidas por el usuario, sí.

SQL Server ofrece varios tipos de funciones para realizar distintas operaciones. Se pueden clasificar de la siguiente manera:

- 1) de agregado: realizan operaciones que combinan varios valores y retornan un único valor. Son "count", "sum", "min" y "max".
- 2) escalares: toman un solo valor y retornan un único valor. Pueden agruparse de la siguiente manera:
 - de configuración: retornan información referida a la configuración. Ejemplo:

```
select @@version
```

retorna la fecha, versión y tipo de procesador de SQL Server.

- de cursores: retornan información sobre el estado de un cursor.
- de fecha y hora: operan con valores "datetime" y "smalldatetime". Reciben un parámetro de tipo fecha y hora y retornan un valor de cadena, numérico o de fecha y hora.
- matemáticas: realizan operaciones numéricas, geométricas y trigonométricas.
- de metadatos: informan sobre las bases de datos y los objetos.



- de seguridad: devuelven información referente a usuarios y funciones.
- de cadena: operan con valores "char", "varchar", "nchar", "nvarchar", "binary" y "varbinary" y devuelven un valor de cadena o numérico.
- del sistema: informan sobre opciones, objetos y configuraciones del sistema.

Ejemplo:

```
select user_name()
```

- estadísticas del sistema: retornan información referente al rendimiento del sistema.
- texto e imagen: realizan operaciones con valor de entrada de tipo text o image y retornan información referente al mismo.

3) de conjuntos de filas: retornan conjuntos de registros.

Se pueden emplear las funciones del sistema en cualquier lugar en el que se permita una expresión en una sentencia "select".

Estudiaremos algunas de ellas.

Funciones matemáticas

Las funciones matemáticas realizan operaciones con expresiones numéricas y retornan un resultado, operan con tipos de datos numéricos.

Microsoft SQL Server tiene algunas funciones para trabajar con números. Aquí presentamos algunas.



abs(x): retorna el valor absoluto del argumento "x". Ejemplo:

```
select abs(-20); retorna 20.
```

ceiling(x): redondea hacia arriba el argumento "x". Ejemplo:

```
select ceiling(12.34); retorna 13.
```

floor(x): redondea hacia abajo el argumento "x". Ejemplo:

```
select floor(12.34); retorna 12.
```

?: devuelve el resto de una división. Ejemplos:

```
select 10%3; retorna 1.
```

```
select 10%2; retorna 0.
```

power(x,y): retorna el valor de "x" elevado a la "y" potencia. Ejemplo:

```
select power(2,3); retorna 8.
```

round(numero,longitud): retorna un número redondeado a la longitud especificada.

"longitud" debe ser tinyint, smallint o int. Si "longitud" es positivo, el número de decimales es redondeado según "longitud"; si es negativo, el número es redondeado desde la parte entera según el valor de "longitud". Ejemplos:

```
select round(123.456,1); retorna desde el "123.500", es decir, redondea desde el primer decimal.
```

```
select round(123.456,2); retorna "123.460", es decir, redondea desde el segundo decimal.
```



`select round(123.456,-1);` retorna "120.000", es decir, redondea desde el primer valor entero (hacia la izquierda).

`select round(123.456,-2);` retorna "100.000", es decir, redondea desde el segundo valor entero (hacia la izquierda).

`sign(x)`; si el argumento es un valor positivo devuelve 1; -1 si es negativo y si es 0, 0.

NOTA: Todas las funciones se pueden aplicar a campos en una tabla de una base de datos. Se pueden emplear estas funciones matemáticas enviando como argumento el nombre de un campo de tipo numérico.

`select sign(campo) from tabla;`

`square(x)`: retorna el cuadrado del argumento. Ejemplo:

`select square(3);` retorna 9.

`sqrt(x)`: devuelve la raíz cuadrada del valor enviado como argumento.

`select sqrt(campo) from tabla;`

SQL Server dispone de funciones trigonométricas que retornan radianes.

Funciones de Cadena

`ascii(character_expression)`: Devuelve el valor del código ASCII del carácter más a la izquierda de una expresión de caracteres.

`select ASCII('Ariel');` retorna 65.

`select ASCII('A');` retorna 65.

`select ASCII('a');` retorna 97.



cast (integer_expression): Convierte un número int en caracteres numéricos.

`select CAST(65 as varchar(4));` retorna '65' como texto.

char (integer_expression): Convierte un código ASCII int en un caracter.

`select CHAR(65);` retorna 'A'.

left (character_expression, integer_expression): Devuelve la parte izquierda de una cadena de caracteres con el número de caracteres especificado.

`select LEFT('Articulo', 2)` retorna 'Ar'.

`select LEFT(campo, 2) from tabla` retorna las primeras dos letras del contenido de la columna 'campo' en la tabla 'tabla'.

len (string_expression): Devuelve el número de caracteres de la expresión de cadena especificada, excluidos los espacios en blanco finales.

`select LEN('Articulo')` retorna 8.

`select LEN(Nomb_prueba) from pruebas;` retorna las diferentes longitudes del contenido de la columna 'Nomb_prueba' en la tabla 'Pruebas', teniendo en cuenta que el tipo de dato corresponde a una cadena de longitud variable.

lower(character_expression): Devuelve una expresión de caracteres después de convertir en minúsculas los datos de caracteres en mayúsculas.

`select LOWER('Articulo')` retorna 'articulo'.

upper(character_expression): Devuelve una expresión de caracteres después de convertir en mayúsculas los datos de caracteres en minúsculas.

`select UPPER(Nomb_prueba) from pruebas;`



right (character_expression, integer_expression): Devuelve la parte derecha de una cadena de caracteres con el número de caracteres especificado.

`select RIGHT('Articulo', 2)` retorna 'lo'.

`select RIGHT(Articulo, 2) from Articulos` retorna las primeras últimas letras del contenido de la columna Articulo en la tabla Articulos.

Funciones de fecha y hora

Microsoft SQL Server ofrece algunas funciones para trabajar con fechas y horas.

Estas son algunas:

getdate(): retorna la fecha y hora actuales. Ejemplo:

`select getdate();`

datepart(partedefecha,fecha): retorna la parte específica de una fecha, el año, trimestre, día, hora, etc.

Los valores para "partedefecha" pueden ser: year (año), quarter (cuarto), month (mes), day (día), week (semana), hour (hora), minute (minuto), second (segundo) y millisecond (milisegundo). Ejemplos:

`select datepart(month,getdate());` retorna el número de mes actual;

`select datepart(day,getdate());` retorna el día actual;

`select datepart(hour,getdate());` retorna la hora actual;

`select datepart(MINUTE,getdate());` retorna los minutos de la hora actual;



`datetime(partedefecha,fecha)`: retorna el nombre de una parte específica de una fecha. Los valores para "partedefecha" pueden ser los mismos que se explicaron anteriormente. Ejemplos:

`select datetime(month,getdate());` retorna el nombre del mes actual;

`select datetime(dw,getdate());` retorna el nombre del día actual;

`dateadd(partedelafecha,numero,fecha)`: agrega un intervalo a la fecha especificada, es decir, retorna una fecha adicionando a la fecha enviada como tercer argumento, el intervalo de tiempo indicado por el primer parámetro, tantas veces como lo indica el segundo parámetro. Los valores para el primer argumento pueden ser: year (año), quarter (cuarto), month (mes), day (día), week (semana), hour (hora), minute (minuto), second (segundo) y millisecond (milisegundo). Ejemplos:

`select dateadd(day,3,'1980/11/02');` retorna "1980/02/14", agrega 3 días.

`select dateadd(month,3,'1980/11/02');` retorna "1981/05/11", agrega 3 meses.

`select dateadd(hour,2,'1980/11/02');` retorna "1980/02/11 2:00:00", agrega 2 horas.

`select dateadd(minute,16,'1980/11/02');` retorna "1980/02/02 00:16:00", agrega 16 minutos.

`datediff(partedelafecha,fecha1,fecha2)`: calcula el intervalo de tiempo (según el primer argumento) entre las 2 fechas. El resultado es un valor entero que corresponde a fecha2-fecha1. Los valores de "partedelafecha" pueden ser los mismos que se especificaron anteriormente. Ejemplos:

`select datediff(day,'2005/10/28','2006/10/28');` retorna 365 (días).

`select datediff(month,'2005/10/28','2006/11/29');` retorna 13 (meses).



day(fecha): retorna el día de la fecha especificada. Ejemplo:

```
select day(getdate());
```

month(fecha): retorna el mes de la fecha especificada. Ejemplo:

```
select month(getdate());
```

year(fecha): retorna el año de la fecha especificada. Ejemplo:

```
select year(getdate());
```

Se pueden emplear estas funciones enviando como argumento el nombre de un campo de tipo datetime o smalldatetime.



FORMATOS DE FECHA Y HORA ESTÁNDAR

FORMATOS DE DÍA	ESTÁNDAR	SENTENCIAS SQL	EJEMPLOS
Mon DD YYYY 1 HH:MIAM (or PM)	Default	SELECT CONVERT(VARCHAR(20), GETDATE(), 100)	Jan 1 2005 1:29PM
MM/DD/YY	USA	SELECT CONVERT(VARCHAR(8), GETDATE(), 1) AS [MM/DD/YY]	11/23/98
MM/DD/YYYY	USA	SELECT CONVERT(VARCHAR(10), GETDATE(), 101) AS [MM/DD/YYYY]	11/23/1998
YY.MM.DD	ANSI	SELECT CONVERT(VARCHAR(8), GETDATE(), 2) AS [YY.MM.DD]	72.01.01
YYYY.MM.DD	ANSI	SELECT CONVERT(VARCHAR(10), GETDATE(), 102) AS [YYYY.MM.DD]	1972.01.01
DD/MM/YY	British/French	SELECT CONVERT(VARCHAR(8), GETDATE(), 3) AS [DD/MM/YY]	19/02/72
DD/MM/YYYY	British/French	SELECT CONVERT(VARCHAR(10), GETDATE(), 103) AS [DD/MM/YYYY]	19/02/1972
DD.MM.YY	German	SELECT CONVERT(VARCHAR(8), GETDATE(), 4) AS [DD.MM.YY]	25.12.05
DD.MM.YYYY	German	SELECT CONVERT(VARCHAR(10), GETDATE(), 104) AS [DD.MM.YYYY]	25.12.2005
DD-MM-YY	Italian	SELECT CONVERT(VARCHAR(8), GETDATE(), 5) AS [DD-MM-YY]	24-01-98
DD-MM-YYYY	Italian	SELECT CONVERT(VARCHAR(10), GETDATE(), 105) AS [DD-MM-YYYY]	24-01-1998
DD Mon YY	-	SELECT CONVERT(VARCHAR(9), GETDATE(), 6) AS [DD MON YY]	04 Jul 06
DD Mon YYYY	-	SELECT CONVERT(VARCHAR(11), GETDATE(), 106) AS [DD MON YYYY]	04 Jul 2006
Mon DD, YY	-	SELECT CONVERT(VARCHAR(10), GETDATE(), 7) AS [Mon DD, YY]	Jan 24, 98
Mon DD, YYYY	-	SELECT CONVERT(VARCHAR(12), GETDATE(), 107) AS [Mon DD, YYYY]	Jan 24, 1998
HH:MM:SS	-	SELECT CONVERT(VARCHAR(8), GETDATE(), 108)	03:24:53
Mon DD YYYY HH:MI:SS.MMMAM (or PM) 1	Default + milliseconds	SELECT CONVERT(VARCHAR(26), GETDATE(), 109)	Apr 28 2006 12:32:29.253PM
MM-DD-YY	USA	SELECT CONVERT(VARCHAR(8), GETDATE(), 10) AS [MM-DD-YY]	01-01-06
MM-DD-YYYY	USA	SELECT CONVERT(VARCHAR(10), GETDATE(), 110) AS [MM-DD-YYYY]	01-01-2006
YY/MM/DD	-	SELECT CONVERT(VARCHAR(8), GETDATE(), 11) AS [YY/MM/DD]	98/11/23
YYYY/MM/DD	-	SELECT CONVERT(VARCHAR(10), GETDATE(), 111) AS [YYYY/MM/DD]	1998/11/23
YYMMDD	ISO	SELECT CONVERT(VARCHAR(6), GETDATE(), 12) AS [YYMMDD]	980124
YYYYMMDD	ISO	SELECT CONVERT(VARCHAR(8), GETDATE(), 112) AS [YYYYMMDD]	19980124
DD Mon YYYY HH:MM:SS.MMM(24h)	Europe default + milliseconds	SELECT CONVERT(VARCHAR(24), GETDATE(), 113)	28 Apr 2006 00:34:55.190
HH:MI:SS.MMM(24H)	-	SELECT CONVERT(VARCHAR(12), GETDATE(), 114) AS [HH:MI:SS.MMM(24H)]	11:34:23.013
YYYY-MM-DD HH:MI:SS(24h)	ODBC Canonical	SELECT CONVERT(VARCHAR(19), GETDATE(), 120)	1972-01-01 13:42:24
YYYY-MM-DD HH:MI:SS.MMM(24h)	ODBC Canonical (with milliseconds)	SELECT CONVERT(VARCHAR(23), GETDATE(), 121)	1972-02-19 06:35:24.489
YYYY-MM-DDTHH:MM:SS.MMM	ISO8601	SELECT CONVERT(VARCHAR(23), GETDATE(), 126)	1998-11-23T11:25:43.250
DD Mon YYYY HH:MI:SS.MMMAM	Kuwaiti	SELECT CONVERT(VARCHAR(26), GETDATE(), 130)	28 Apr 2006 12:39:32.429AM