

Desarrollo de Bases de Datos II

Instituto Profesional de Chile - IPCHILE

INGENIERÍA EN INFORMÁTICA

Docente Jonathan Toloza Carachi

Republica

- 1.- Funciones
- 2.- **Funciones para el manejo de cadenas**
- 3.- **Funciones matemáticas**
- 4.- **Funciones para el uso de fechas y horas**

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 si.

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 para el manejo de cadenas

Microsoft SQL Server tiene algunas funciones para trabajar con cadenas de caracteres. Estas son algunas:

- `substring(cadena,inicio,longitud)`: devuelve una parte de la cadena especificada como primer argumento, empezando desde la posición especificada por el segundo argumento y de tantos caracteres de longitud como indica el tercer argumento. Ejemplo:

```
select substring('Buenas tardes',8,6);  
retorna "tardes".
```

- **str(numero,longitud,cantidaddecimales)**: convierte números a caracteres; el primer parámetro indica el valor numérico a convertir, el segundo la longitud del resultado (debe ser mayor o igual a la parte entera del número más el signo si lo tuviese) y el tercero, la cantidad de decimales. El segundo y tercer argumento son opcionales y deben ser positivos. String significa cadena en inglés.

Ejemplo: se convierte el valor numérico "123.456" a cadena, especificando 7 de longitud y 3 decimales:

```
select str(123.456,7,3);
```

```
select str(-123.456,7,3);  
retorna '-123.46';
```

Si no se colocan el segundo y tercer argumento, la longitud predeterminada es 10 y la cantidad de decimales 0 y se redondea a entero. Ejemplo: se convierte el valor numérico "123.456" a cadena:

```
select str(123.456);  
retorna '123';
```

```
select str(123.456,3);  
retorna '123';
```

Si el segundo parámetro es menor a la parte entera del número, devuelve asteriscos (*). Ejemplo:

select str(123.456,2,3);

retorna "*".**

- stuff(cadena1,inicio,cantidad,cadena2):
inserta la cadena enviada como cuarto argumento, en la posición indicada en el segundo argumento, reemplazando la cantidad de caracteres indicada por el tercer argumento en la cadena que es primer parámetro. Stuff significa rellenar en inglés. Ejemplo:

select stuff('abcde',3,2,'opqrs');
retorna "abopqrse". Es decir, coloca en la posición 2 la cadena "opqrs" y reemplaza 2 caracteres de la primer cadena.

Los argumentos numéricos deben ser positivos y menor o igual a la longitud de la primera cadena, caso contrario, retorna "null".

Si el tercer argumento es mayor que la primera cadena, se elimina hasta el primer carácter.

- len(cadena): retorna la longitud de la cadena enviada como argumento. "len" viene de length, que significa longitud en inglés.

Ejemplo:

select len('Hola');
devuelve 4.

- char(x): retorna un caracter en código ASCII del entero enviado como argumento. Ejemplo:

select char(65);

- left(cadena,longitud): retorna la cantidad (longitud) de caracteres de la cadena comenzando desde la izquierda, primer caracter.

Ejemplo:

select left('buenos dias',8);
retorna "buenos d".

- right(cadena,longitud): retorna la cantidad (longitud) de caracteres de la cadena comenzando desde la derecha, último caracter.

Ejemplo:

select right('buenos dias',8);
retorna "nos dias".

-lower(cadena): retornan la cadena con todos los caracteres en minúsculas. lower significa reducir en inglés. Ejemplo:

select lower('HOLA ESTUDIAnte');
retorna "hola estudiante".

-upper(cadena): retornan la cadena con todos los caracteres en mayúsculas. Ejemplo:

select upper('HOLA ESTUDIAnte');

-ltrim(cadena): retorna la cadena con los espacios de la izquierda eliminados. Trim significa recortar. Ejemplo:

```
select ltrim('  Hola  ');  
retorna "Hola ".
```

- rtrim(cadena): retorna la cadena con los espacios de la derecha eliminados. Ejemplo:

```
select rtrim('  Hola  ');  
retorna " Hola".
```

- **replace(cadena,cadenareemplazo, cadenareemplazar):** retorna la cadena con todas las ocurrencias de la subcadena reemplazo por la subcadena a reemplazar. Ejemplo:

select replace('xxx.sqlserverya.com','x','w');
retorna "www.sqlserverya.com".

- **reverse(cadena):** devuelve la cadena invirtiendo el order de los caracteres. Ejemplo:

select reverse('Hola');
retorna "aloH".

- patindex(patron,cadena): devuelve la posición de comienzo (de la primera ocurrencia) del patrón especificado en la cadena enviada como segundo argumento. Si no la encuentra retorna 0.

Ejemplos:

select patindex('%Luis%', 'Jorge Luis Borges');
retorna 7.

select patindex('%or%', 'Jorge Luis Borges');
retorna 2.

select patindex('%ar%', 'Jorge Luis Borges');
retorna 0.

- charindex(subcadena,cadena,inicio): devuelve la posición donde comienza la subcadena en la cadena, comenzando la búsqueda desde la posición indicada por "inicio". Si el tercer argumento no se coloca, la búsqueda se inicia desde 0. Si no la encuentra, retorna 0. Ejemplos:

**select charindex('or','Jorge Luis Borges',5);
retorna 13.**

**select charindex('or','Jorge Luis Borges');
retorna 2.**

select charindex('or','Jorge Luis Borges',14);
retorna 0.

select charindex('or', 'Jorge Luis Borges');
retorna 0.

- replicate(cadena,cantidad): repite una cadena la cantidad de veces especificada. Ejemplo:

**select replicate ('Hola',3);
retorna "HolaHolaHola";**

- space(cantidad): retorna una cadena de espacios de longitud indicada por "cantidad", que debe ser un valor positivo. Ejemplo:

**select 'Hola'+space(1)+'que tal';
retorna "Hola que tal".**

Se pueden emplear estas funciones enviando como argumento el nombre de un campo de tipo caracter.

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.

-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 "123.400", 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.

-square(x): retorna el cuadrado del argumento.

Ejemplo:

select square(3); retorna 9.

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

SQL Server dispone de funciones trigonométricas que retornan radianes.

Se pueden emplear estas funciones enviando como argumento el nombre de un campo de tipo numérico.

Funciones para el uso de fechas y horas

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 (dia), 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;

- datetimename(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 datetimename(month,getdate());
retorna el nombre del mes actual;

select datetimename(day,getdate());

- **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 (dia), week (semana), hour (hora), minute (minuto), second (segundo) y millisecond (milisegundo).

Ejemplos:

select dateadd(day,3,'1980/11/02');
retorna "1980/11/05", agrega 3 días.

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

select dateadd(hour,2,'1980/11/02');
retorna "1980/02/02 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.