

25 - 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);
```

retorna "A".

- **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 orden 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.

Servidor de SQL Server instalado en forma local.

Ingresemos el siguiente lote de comandos en el SQL Server Management Studio:

```
if object_id ('libros') is not null
    drop table libros;

create table libros(
    codigo int identity,
    titulo varchar(40) not null,
    autor varchar(20) default 'Desconocido',
    editorial varchar(20),
    precio decimal(6,2),
    cantidad tinyint default 0,
    primary key (codigo)
);

go

insert into libros (titulo,autor,editorial,precio)
    values('El aleph','Borges','Emece',25);
insert into libros
    values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
insert into libros (titulo,autor,editorial,precio,cantidad)
    values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);

-- Mostramos sólo los 12 primeros caracteres de los títulos de los libros y
-- sus autores, empleando la función "substring()":
select substring(titulo,1,12) as titulo
    from libros;

-- Mostramos sólo los 12 primeros caracteres de los títulos de los libros y
-- sus autores, ahora empleando la función "left()":
select left(titulo,12) as titulo
    from libros;

-- Mostramos los títulos de los libros y sus precios convirtiendo este último a cadena
-- de caracteres con un solo decimal, empleando la función "str":
select titulo,
    str(precio,6,1)
    from libros;

-- Mostramos los títulos de los libros y sus precios convirtiendo este último a cadena
-- de caracteres especificando un solo argumento:
select titulo,
    str(precio)
    from libros;

-- Mostramos los títulos, autores y editoriales de todos libros, al último
-- campo lo queremos en mayúsculas:
```

```
select titulo, autor, upper(editorial)
      from libros;
```