

26 - 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 "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 raiz 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.

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),
    primary key (codigo)
);
```

```
go
```

```
insert into libros (titulo,autor,editorial,precio)
    values('El aleph','Borges','Emece',25.33);
insert into libros
    values('Java en 10 minutos','Mario Molina','Siglo XXI',50.65);
insert into libros (titulo,autor,editorial,precio)
    values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',19.95);
```

```
-- Vamos a mostrar los precios de los libros redondeando el valor hacia abajo y hacia arriba:
select titulo,autor,precio,
    floor(precio) as abajo,
    ceiling(precio) as arriba
from libros;
```

Tenemos como resultado:

```
    precio decimal(6,2),
    primary key (codigo)
);

go

insert into libros (titulo,autor,editorial,precio)
values('El aleph','Borges','Emece',25.33);
insert into libros
values('Java en 10 minutos','Mario Molina','Siglo XXI',50.65);
insert into libros (titulo,autor,editorial,precio)
values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',19.95);

-- Vamos a mostrar los precios de los libros redondeando el valor hacia abajo y hacia arriba:
select titulo,autor,precio,
       floor(precio) as abajo,
       ceiling(precio) as arriba
from libros;
```

100 %

Resulta Messages

	titulo	autor	precio	abajo	arriba
1	El aleph	Borges	25.33	25	26
2	Java en 10 minutos	Mario Molina	50.65	50	51
3	Alicia en el pais de las maravillas	Lewis Carroll	19.95	19	20