

# Funciones

## Funciones

SQL Server ofrece varios tipos de funciones para realizar distintas operaciones.

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

No todas las sentencias SQL son válidas dentro de una función. NO es posible emplear en ellas funciones no determinadas (como getdate()) ni sentencias de modificación o actualización de tablas o vistas.

SQL Server admite 3 tipos de funciones definidas por el usuario clasificadas según el valor retornado:

- escalares: retornan un valor escalar
- de tabla de varias instrucciones (retornan una tabla)
- de tabla en línea (retornan una tabla).

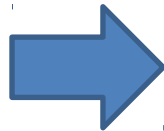
Las funciones definidas por el usuario se crean con la instrucción "create function" y se eliminan con "drop function".

# Funciones escalares

```
create function NOMBRE  
(@PARAMETRO TIPO=VALORPORDEFEECTO,...)  
returns TIPO  
begin  
    INSTRUCCIONES...  
    return VALOR  
end;
```

```
create function f_promedio  
(@valor1 decimal(4,2),  
 @valor2 decimal(4,2)  
)  
returns decimal(6,2)  
as  
begin  
    declare @resultado decimal(6,2)  
    set @resultado=(@valor1+@valor2)/2  
    return @resultado  
end;
```

Ejecución



```
select dbo.f_promedio(5.5,8.5);
```

```
create function f_letraNIF(@dni varchar(8))
returns char
as
begin
return SUBSTRING('TRWAGMYFPDXBNJZSQVHLCKE', @dni % 23 + 1, 1)
end;
print dbo.f_letraNIF('56789443')
```

```
if object_id('Cubo') is not null
    drop function Cubo;
CREATE FUNCTION dbo.Cubo( @fNumero float)
RETURNS float
AS
BEGIN
RETURN (@fNumero * @fNumero * @fNumero)
END
print dbo.Cubo(3)
select precio, dbo.cubo(precio)
from LIB_Libros
```



Varias formas  
de ejecución

```
CREATE FUNCTION dbo.Factorial ( @iNumber int )
RETURNS INT
AS
BEGIN
DECLARE @i int
IF @iNumber <= 1
    SET @i = 1
ELSE
    SET @i = @iNumber * dbo.Factorial( @iNumber - 1 )
RETURN (@i)
END
print dbo.Factorial(4)
```

```
CREATE FUNCTION [dbo].[fnEdad]
(@fNacimiento datetime,
@preferencia datetime
)
RETURNS int
AS
BEGIN
DECLARE @ResultVar int
set @ResultVar = DATEDIFF(year,@fNacimiento,@preferencia )+
case
when ( Month(@preferencia) < Month(@fNacimiento) Or
(Month(@preferencia) = Month(@fNacimiento) And Day(@preferencia) <
day(@fNacimiento))) Then -1 else 0 end
RETURN @ResultVar
END
```

# Funciones con valores de tabla en línea

Una función con valores de tabla en línea retorna una tabla que es el resultado de una única instrucción "select".

```
create function NOMBREFUNCION  
(@PARAMETRO TIPO=VALORPORDEFECTO)  
returns table  
as  
return (  
  select CAMPOS  
  from TABLA  
  where CONDICION  
);
```

```
-- funciones que devuelven una tabla en línea
-- como una vista con parámetros
CREATE FUNCTION dbo.f_VerLibrosDeAutores (@autor varchar(40) )
RETURNS TABLE
AS
RETURN
(
select autor,titulo
from LIB_Libros as l
inner join LIB_Autores as a
on l.CodAutor=a.CodAutor
where Autor like '%' + @autor + '%'
)
go
select * from dbo.f_VerLibrosDeAutores ('García')
go
```



# Funciones de tabla de varias instrucciones

Este tipo de función es similar a un procedimiento almacenado; la diferencia es que la tabla retornada por la función puede ser referenciada en el "from" de una consulta, pero el resultado de un procedimiento almacenado no.

También es similar a una vista; pero en las vistas solamente podemos emplear "select", mientras que en funciones definidas por el usuario podemos incluir sentencias como "if", llamadas a funciones, procedimientos, etc.

```
create function NOMBREFUNCION
(@PARAMETRO TIPO)
returns @NOMBRETABLARETORNO table-- nombre de la tabla
--formato de la tabla
(CAMPO1 TIPO,
CAMPO2 TIPO,
CAMPO3 TIPO
)
as
begin
insert @NOMBRETABLARETORNO
select CAMPOS
from TABLA
where campo OPERADOR @PARAMETRO
RETURN
end
```

```
--funciones que devuelven una tabla dando su estructura
if object_id('libros') is not null
    drop table libros;
create table libros(
    codigo int identity,
    titulo varchar(40),
    autor varchar(30),
    editorial varchar(20),
    precio decimal(6,2)
);

--Ingresamos algunos registros:
insert into libros values('Uno','Richard Bach','Planeta',15);
insert into libros values('Ilusiones','Richard Bach','Planeta',10);
insert into libros values('El aleph','Borges','Emece',25);
insert into libros values('Aprenda PHP','Mario Molina','Siglo XXI',55);
insert into libros values('Alicia en el pais','Lewis Carroll','Paidos',35);
insert into libros values('Matematica estas ahi','Paenza','Nuevo siglo',25);

--Eliminamos la función "f_ofertas" si existe.
if object_id('f_ofertas') is not null
    drop function f_ofertas;
```

```
create function f_ofertas
  (@minimo decimal(6,2)
  )
  returns @ofertas table-- nombre de la tabla
  --formato de la tabla
  (codigo int,
   titulo varchar(40),
   autor varchar(30),
   precio decimal(6,2)
  )
  as
  begin
    insert @ofertas
      select codigo,titulo,autor,precio
      from libros
      where precio<@minimo
    return
  end;
select * from dbo.f_ofertas(20);
```

```
create function ver_factura_cliente
(@codcliente int)
returns varchar(max)
begin
    declare @delimitador varchar, @nfac varchar(max)
    set @delimitador = '#'
    set @nfac = ''
    select @nfac=@nfac + str(nfactura)+@delimitador
        from FAC_T_factura where codcliente=@codcliente
    return @nfac
end
go

select NombreCliente, dbo.ver_factura_cliente(codcliente)
from FAC_T_Cliente

go
```

```

--8 Hacer una función que nos a la que le pasaremos un parámetro que podrá ser
--DNI ó Matrícula.
--Si es DNI nos sacará el DNI de todos los Alquileres de la tabla AlquilerCoche.
--Si es matrícula nos sacará los valores almacenados en el campo Matricula para
--todos los Alquileres de la tabla AlquilerCoche.
use Adm88
go
if object_id ('f_verdatos') is not null
    drop function f_verdatos;
go
create function f_verdatos
    (@campo varchar(20))
returns @datos table
    (dato varchar(10)
    )
as
begin
    if @campo='DNI'
        begin
            insert @datos
                select dni
                from AlquilerCoche
        end
    else
        begin
            insert @datos
                select matricula
                from AlquilerCoche
        end
    return
end;
go
select * from dbo.f_verdatos('DNI');
select * from dbo.f_verdatos('Matrícula');

```