

Funciones

Es un objeto que almacena un conjunto de instrucciones que desempeñan una tarea específica y que se ejecutan en bloques.

Características:

- ❖ Puede aceptar o no parámetros y retornan un valor único o una tabla
- ❖ No se pueden usar para Insertar, Actualizar o Eliminar registros en una tabla.
- ❖ Almacena una lógica que se puede ejecutar dentro de una sentencia SQL

Dos tipos de funciones:

- ❖ Escalar: Retorna un solo valor como resultado de todas las operaciones de la función
- ❖ De tabla: Retorna una tabla como resultado de todas las operaciones de la función (no se utilizan mucho)

Diferencia entre un SP y Función:

1. De una Función no puedo llamar a un SP. Mientras que de un SP si puedo llamar a una función
2. Los SP no trabajan con la palabra "RETURN"

Ejemplo 1

```
CREATE FUNCTION calcularpromedio2 (  
    @valor1 AS DECIMAL(10,2),  
    @valor2 AS DECIMAL(10,2))  
  
RETURNS DECIMAL(10,2)  
  
AS  
BEGIN  
    DECLARE @resultado DECIMAL(10,2)  
    SET @resultado = (@valor1 + @valor2)/2  
    RETURN @resultado  
END  
  
SELECT dbo.calcularpromedio2 (12,13) AS Promedio  
  
PRINT 'Promedio: ' + CAST(dbo.calcularpromedio2(12,13) AS VARCHAR(10))
```

Results		Messages
	Promedio	
1	12.50	

Results Messages

(1 row affected)
Promedio: 12.50

Completion time: 2022-09-12T22:04:12.2536817-03:00

Ejemplo 2

CREATE FUNCTION fechalettras (@fecha AS DATE)

RETURNS VARCHAR(20)

AS

BEGIN

DECLARE @resultado VARCHAR(20)

SET @resultado = CONCAT(DAY(@fecha), ' de ',
 DATENAME(MONTH, @fecha), ' del ',
 YEAR(@fecha))

RETURN @resultado

END

SELECT dbo.fechalettras('2015-08-08') AS 'Fecha Actual'

Results		Messages
	Fecha Actual	
1	8 de August del 2015	

Ejemplo 3

Implementar una **función de tabla** en línea que muestre los registros de la tabla "Sales.SalesTerritory" dependiendo del Grupo (Group) al que pertenezca.

CREATE FUNCTION departamentoempleado (@departamento AS CHAR(30))

RETURNS TABLE

AS

RETURN(SELECT E.emp_no, (E.emp_fname + ' ' + E.emp_lname) AS Empleado,
 D.dept_no AS Departamento
 FROM employee E
 INNER JOIN department D
 ON D.dept_no = E.dept_no
 WHERE D.dept_no = @departamento)

SELECT * FROM dbo.departamentoempleado('d1')

Results

Messages

	emp_no	Empleado		Departamento
1	18316	John	Barimore	d1
2	28559	Sybill	Moser	d1

Ejemplo 4

```
USE DBRESERVA
```

```
CREATE FUNCTION calculocobroextra (@extprecio DECIMAL(9,2))
```

```
RETURNS DECIMAL(9,2)
```

```
AS
```

```
BEGIN
```

```
    RETURN @extprecio * 0.19
```

```
END
```

```
SELECT idtarifa,  
       clase,  
       precio,  
       '0.19' AS Impuesto,  
       dbo.calculocobroextra(precio) AS Total
```

```
FROM TARIFA
```

Results		Messages			
	idtarifa	clase	precio	Impuesto	Total
1	1	Supervip	1200,00	0.19	300.00
2	2	Vip	1000,00	0.19	250.00
3	3	Nacional	800,00	0.19	200.00
4	4	Económica	500,00	0.19	125.00

Ejemplo 5

Utilizando el ejemplo anterior:

Que pasaria si tengo más lógica con respecto al precio y al % de Impuesto:

1 a 20 ==> 18%
21 a 50 ==> 20%
51 a 100 ==> 22%
101 a más ==> 25%

```
USE DBRESERVA
```

```
CREATE FUNCTION calculocobroextra (@extprecio DECIMAL(9,2))
```

```
RETURNS DECIMAL(9,2)
```

```
AS
```

```
BEGIN
```

```
    DECLARE @result DECIMAL(9,2)
```

```
    SET @result =
```

```
        CASE
```

```
            WHEN @extprecio BETWEEN 1 AND 20 THEN (@extprecio * 0.18)
```

```
            WHEN @extprecio BETWEEN 21 AND 50 THEN (@extprecio * 0.20)
```

```
            WHEN @extprecio BETWEEN 51 AND 100 THEN (@extprecio * 0.22)
```

```
            ELSE @extprecio * 0.25
```

```
        END
```

```
    RETURN @result
```

```
END
```

```

SELECT idtarifa,
       clase,
       precio,
       dbo.calculocobroextra(precio) AS Total
FROM TARIFA

```

	idtarifa	clase	precio	Total
1	1	Supervip	1200,00	300.00
2	2	Vip	1000,00	250.00
3	3	Nacional	800,00	200.00
4	4	Económica	500,00	125.00

Ejemplo 6

```
USE DBRESERVA
```

```

CREATE FUNCTION FN_Concatenar(@param1 VARCHAR(50),
                              @param2 VARCHAR(75),
                              @param3 VARCHAR(5))
RETURNS VARCHAR(200)

```

```

AS
BEGIN
    RETURN @param1+@param3+@param2
END

```

```
SELECT dbo.FN_Concatenar('Alfonso','Pérez',' ')
```

	(No column name)
1	Alfonso Pérez

Ejemplo 7

```

CREATE OR ALTER FUNCTION FN_Correo(@Nombre VARCHAR(50),
                                   @Apellido VARCHAR(50))
RETURNS VARCHAR(MAX)

```

```

AS
BEGIN
    DECLARE @SubNombre VARCHAR(5);
    DECLARE @SubApellido VARCHAR(5);
    DECLARE @Union VARCHAR(100);

    SET @SubNombre = SUBSTRING(@Nombre,1,3)
    SET @SubApellido = SUBSTRING(@Apellido,1,3)
    SET @Union = dbo.FN_Concatenar(@SubNombre,@SubApellido,('.') +
                                   '@gmail.com'

```

```
    RETURN LOWER(@Union)
```

```
END
```

```
SELECT dbo.FN_Correo ('Alfonso','Perez')
```

	(No column name)
1	alf.per@gmail.com

Ejemplo 8

USE ADDC

```
CREATE OR ALTER FUNCTION FN_Edad (@FecNac DATE)
RETURNS INT
AS
BEGIN
    RETURN DATEDIFF(YEAR,@FecNac, GETDATE())
END

SELECT dbo.FN_Edad ('1996-12-10')
```