## 

línea horizontal



Actividad UT 15-FUNCIONES Y CURSORES

Base de Datos

Lunes 6 de Abril de 2024

**─**

Hirahi Mejías Delgado

1 DAM-A

**EJERCICIOS A REALIZAR**

**EJERCICIO 1. -**

**1.- Crear una función que devuelva cuántos libros hay de precio mayor que el que**

**suministremos como parámetro.**

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(5,2)

);

insert into libros values('Alicia en el pais de las maravillas','Lewis

Carroll','Emece',20.00);

insert into libros values('Alicia en el pais de las maravillas','Lewis

Carroll','Plaza',35.00);

insert into libros values('Aprenda PHP','Mario Molina','Siglo XXI',40.00);

insert into libros values('El aleph','Borges','Emece',10.00);

insert into libros values('Ilusiones','Richard Bach','Planeta',15.00);

insert into libros values('Java en 10 minutos','Mario Molina','Siglo XXI',50.00);

insert into libros values('Martin Fierro','Jose Hernandez','Planeta',20.00);

insert into libros values('Martin Fierro','Jose Hernandez','Emece',30.00);

insert into libros values('Uno','Richard Bach','Planeta',10.00);

**Funcion:**

CREATE FUNCTION ContarLibrosPrecioMayor(@precio decimal(5,2))

RETURNS INT

AS

BEGIN

DECLARE @CantidadLibros INT;

SELECT @CantidadLibros = COUNT(\*)

FROM libros

WHERE precio > @precio;

RETURN @CantidadLibros;

END;

****

**2- Crear una función escalar que tenga como parámetros el DNI y la letra del NIF y**

**nos valide si es correcta o no. Usar la función con los datos de una tabla que**

**contenga nombre, apellidos, fechanacimiento, dni y la letra del nif.**

CREATE FUNCTION dbo.ValidarNIF (@dni CHAR(8), @letra CHAR(1))

RETURNS BIT

AS

BEGIN

DECLARE @letraCorrecta CHAR(1)

-- Verificar si el DNI tiene 8 caracteres y si son todos dígitos

IF LEN(@dni) <> 8 OR PATINDEX('%[^0-9]%', @dni) <> 0

BEGIN

RETURN 0; -- DNI incorrecto

END

-- Calcular la letra correcta correspondiente al DNI

SET @letraCorrecta = SUBSTRING('TRWAGMYFPDXBNJZSQVHLCKE', ((CONVERT(INT, @dni) % 23) + 1), 1)

-- Comparar la letra proporcionada con la letra correcta

IF @letra = @letraCorrecta

BEGIN

RETURN 1; -- NIF válido

END

ELSE

BEGIN

RETURN 0; -- NIF incorrecto

END

RETURN 0;

END;

SELECT

nombre,

apellidos,

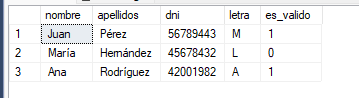
dni,

letra,

dbo.ValidarNIF(dni, letra) AS es\_valido

FROM

personas;



**3.- Crear una función que nos devuelva los años de diferencia respecto al actual a**

**partir de la fecha pasada como parámetro. Usar la función con la tabla anterior.**

CREATE FUNCTION calcular\_anios\_diferencia(@fecha datetime)

RETURNS INT

AS

BEGIN

DECLARE @anios\_diferencia INT;

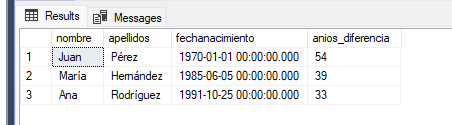
SELECT @anios\_diferencia = DATEDIFF(YEAR, @fecha, GETDATE());

RETURN @anios\_diferencia;

END;

SELECT nombre, apellidos, fechanacimiento, dbo.calcular\_anios\_diferencia(fechanacimiento) AS anios\_diferencia

FROM personas;

****

**4.- Crear función que dada fecha como cadena de caracteres devuelva que no es**

**correcta o en caso contrario el nombre del mes.**

CREATE FUNCTION validar\_fecha(@fecha varchar(10))

RETURNS VARCHAR(20)

AS

BEGIN

DECLARE @nombre\_mes VARCHAR(20);

IF ISDATE(@fecha) = 0

BEGIN

SET @nombre\_mes = 'Fecha no válida';

END

ELSE

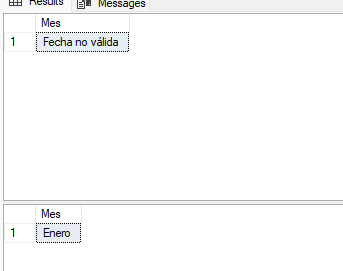
BEGIN

SET @nombre\_mes = DATENAME(MONTH, @fecha);

END

RETURN @nombre\_mes;

END;

****

**5.- Crear una función de tabla que devuelva los libros de precio mayor que el que**

**suministremos como parámetro.**

CREATE FUNCTION LibrosMasCaros(@precio decimal(5,2))

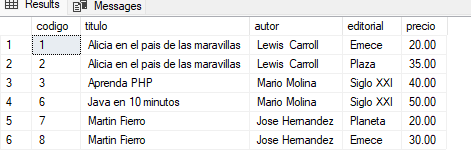
RETURNS TABLE

AS

return

(select \* from libros where precio > @precio)

select \* from dbo.LibrosMasCaros(15)

****

**6.- Función que devuelva el máximo precio de la tabla libros**

CREATE FUNCTION LibroMasCaro()

RETURNS DECIMAL(5,2)

AS

BEGIN

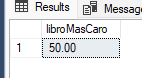
Declare @Max\_precio decimal(5,2)

select @Max\_precio=MAX(precio) from libros;

return @Max\_precio

end

select dbo.LibroMasCaro() as libroMasCaro;

****

**7.- Función que devuelva una tabla con el nombre y dni de las personas de dos**

**tablas (personal y alumnado), pasando como parámetro: personal (saca sólo los de**

**la tabla personal; alumnado (saca sólo los de la tabla alumnado; ambos (saca los de**

**ambas tablas).**

CREATE FUNCTION ObtenerDatos (@tipo VARCHAR(10))

RETURNS @resultado TABLE (nombre VARCHAR(100), apellidos VARCHAR(100))

AS

BEGIN

IF @tipo = 'personal'

BEGIN

INSERT INTO @resultado

SELECT nombre, apellidos FROM personal;

END

ELSE IF @tipo = 'alumnado'

BEGIN

INSERT INTO @resultado

SELECT nom AS nombre, apell AS apellidos FROM alumnado;

END

ELSE IF @tipo = 'ambos'

BEGIN

INSERT INTO @resultado

SELECT nombre, apellidos FROM personal;

INSERT INTO @resultado

SELECT nom AS nombre, apell AS apellidos FROM alumnado;

END

RETURN;

END;

-- Para obtener solo los datos de la tabla personal

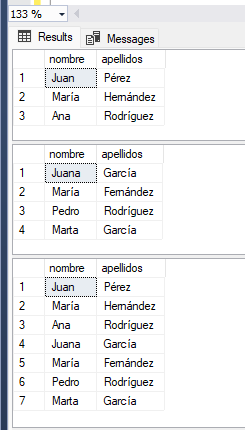
SELECT \* FROM dbo.ObtenerDatos('personal');

-- Para obtener solo los datos de la tabla alumnado

SELECT \* FROM dbo.ObtenerDatos('alumnado');

-- Para obtener los datos de ambas tablas

SELECT \* FROM dbo.ObtenerDatos('ambos');

****

**8.- Función que devuelva el nº de días del mes de una fecha pasada como**

**parámetro.**

CREATE FUNCTION obtener\_dias\_mes(@fecha date)

RETURNS INT

AS

BEGIN

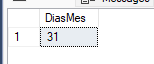
DECLARE @dias\_mes INT;

SET @dias\_mes = DAY(EOMONTH(@fecha));

RETURN @dias\_mes;

END;

select dbo.obtener\_dias\_mes('2003-10-28') as DiasMes

****

**9.- Crear función que valide si una cadena de caracteres es un DNI correcto. Que**

**contenga 8 dígitos y una letra y la letra se corresponda con la correcta. Probarlo con**

**la tabla personas.**

CREATE FUNCTION ValidarDNI (@dni CHAR(9))

RETURNS BIT

AS

BEGIN

DECLARE @numeros CHAR(8);

DECLARE @letra CHAR(1);

DECLARE @letra\_correcta CHAR(1);

DECLARE @resultado BIT;

-- Verificar si la cadena tiene exactamente 9 caracteres

IF LEN(@dni) <> 9

BEGIN

SET @resultado = 0; -- DNI incorrecto

END

ELSE

BEGIN

-- Extraer los primeros 8 caracteres (los números)

SET @numeros = LEFT(@dni, 8);

-- Extraer la última letra

SET @letra = UPPER(RIGHT(@dni, 1));

-- Calcular la letra correcta correspondiente a los números

SET @letra\_correcta = CHAR(64 + (CAST(@numeros AS INT) % 23));

-- Comparar la letra proporcionada con la letra correcta y devolver un valor booleano

IF @letra = @letra\_correcta

BEGIN

SET @resultado = 1; -- DNI válido

END

ELSE

BEGIN

SET @resultado = 0; -- DNI no válido

END

END

RETURN @resultado;

END;

SELECT

nombre,

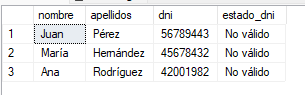
apellidos,

dni,

CASE WHEN dbo.ValidarDNI(dni) = 1 THEN 'Válido' ELSE 'No válido' END AS estado\_dni

FROM

personas;

****

**EJERCICIO 2. -**

**Crear un procedimiento almacenado que recorra una tabla con un cursor. Esta tabla**

**contendrá en un campo el nombre de la tabla a crear y en otros dos las**

**descripciones de los dos campos que compondrán dicha tabla crear. El**

**procedimiento deberá crear todas las tablas que se indiquen en TablasCrear.**

create table TablasCrear

(Tabla nvarchar(100),

DescripcionCampo1 nvarchar(100),

DescripcionCampo2 nvarchar(100))

go

insert into TablasCrear (Tabla, DescripcionCampo1,

DescripcionCampo2)

values (N'alum',N'DNI char(9)',N'Nombre varchar(50)')

insert into TablasCrear (Tabla, DescripcionCampo1,

DescripcionCampo2)

values (N'libro',N'Titulo varchar(50)',N'Autor varchar(50)')

go

**Procedimiento:**

CREATE PROCEDURE CrearTablas

AS

BEGIN

DECLARE @Tabla nvarchar(100);

DECLARE @DescripcionCampo1 nvarchar(100);

DECLARE @DescripcionCampo2 nvarchar(100);

DECLARE @SQL nvarchar(max);

DECLARE tabla\_cursor CURSOR FOR

SELECT Tabla, DescripcionCampo1, DescripcionCampo2

FROM TablasCrear;

OPEN tabla\_cursor;

FETCH NEXT FROM tabla\_cursor INTO @Tabla, @DescripcionCampo1, @DescripcionCampo2;

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @SQL = 'CREATE TABLE ' + QUOTENAME(@Tabla) + ' (' +

@DescripcionCampo1 + ',' +

@DescripcionCampo2 + ');';

EXEC sp\_executesql @SQL;

FETCH NEXT FROM tabla\_cursor INTO @Tabla, @DescripcionCampo1, @DescripcionCampo2;

END;

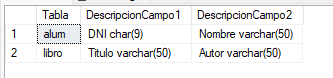
CLOSE tabla\_cursor;

DEALLOCATE tabla\_cursor;

END;

exec CrearTablas;

select \* from TablasCrear

****