

BASES DE DATOS - SQL SERVER PROCESUAL HITO 4

BASE DE DATOS I - 2022

CARRERA: INGENIERÍA DE SISTEMAS

ESTUDIANTE: HERLAN LUIS ALEJO MAMANI

DOCENTE: LIC. WILLIAM BARRA PAREDES

MUESTRA UN EJEMPLO DE DDL.

· DDL (DATA DEFINITION LANGUAGE - LENGUAJE DE DEFINICIÓN DE DATOS). LAS

SENTENCIAS DDL SON AQUELLAS UTILIZADAS PARA LA CREACIÓN DE UNA BASE DE DATOS Y

TODOS SUS COMPONENTES: TABLAS, ÍNDICES, RELACIONES, DISPARADORES (TRIGGERS),

PROCEDIMIENTOS ALMACENADOS, ETC.

COMANDOS SQL			
DDL	DML	DCL	TCL
CREATE	SELECT	GRANT	COMMIT
ALTER	INSERT	REVOKE	ROLLBACK
DROP	UPDATE		SAVEPOINT
TRUNCATE	DELETE		SET TRANSACTION
COMMENT	MERGE		
RENAME	CALL		
	EXPLAIN PLAN		
	LOCK TABLE		

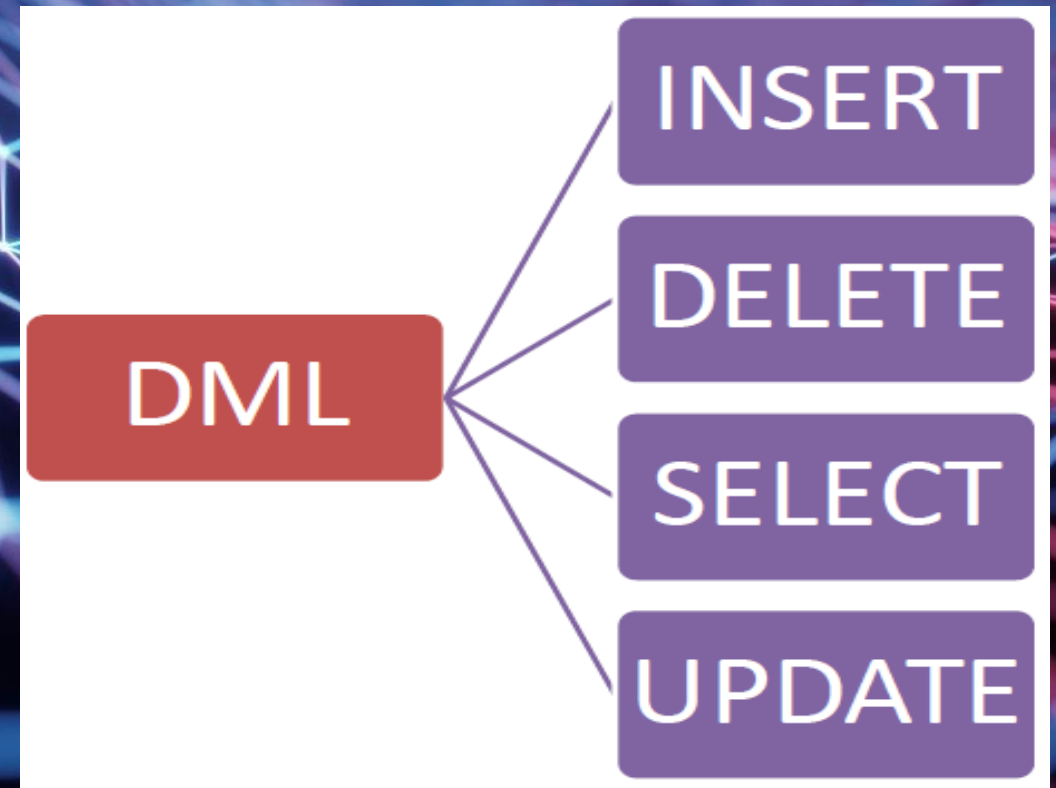
MUESTRA UN EJEMPLO DE DML

LAS SENTENCIAS DML SE UTILIZAN PARA CONTROLAR LA INFORMACIÓN CONTENIDA EN LA BASE DE DATOS. LAS LISTAS SIGUIENTES OFRECEN EJEMPLOS DE ESTOS TIPOS DE SENTENCIAS DML:

ADICIÓN DE REGISTROS A UNA TABLA
(MANDATO INSERT)

MODIFICACIÓN DE LA INFORMACIÓN DE
UNA TABLA (MANDATO UPDATE)

ELIMINACIÓN DE REGISTROS DE UNA TABLA
(MANDATO DELETE)



PARA QUE DRIVE INNER JOIN

ESTA CLÁUSULA BUSCA COINCIDENCIAS ENTRE 2 TABLAS, EN FUNCIÓN A UNA COLUMNA QUE TIENEN EN COMÚN. DE TAL MODO QUE SÓLO LA INTERSECCIÓN SE MOSTRARÁ EN LOS RESULTADOS.

Query 1 x

```
34 insert into libros (titulo, autor,codigoeditorial,precio,cantidad)
35 values('Cervantes y el quijote','Borges- Bioy Casares',3,25.5,300);
36
37
38 select 1.codigo,titulo,autor,nombre from libros as l
39 join editoriales as e
40 on 1.codigoeditorial=e.codigo;
41
42 -- Igual resultado obtenemos con inner join
43 select 1.codigo,titulo,autor,nombre from libros as l
44 inner join editoriales as e
45 on 1.codigoeditorial=e.codigo;
46
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents: |

	codigo	titulo	autor	nombre
1	1	El Aleph	Borges	Planeta
2	2	Alicia en el pais de las maravillas	Lewis Carroll	Emece
3	3	Aprenda PHP	Mario Perez	Paidós
4	4	Java en 10 minutos	Juan Lopez	Paidós
5	5	Alicia a través del espejo	Lewis Carroll	Paidós
6	6	Cervantes y el quijote	Borges- Bioy Casares	Planeta

Result 1 | Result 2 x

DEFINA QUE ES UNA FUNCIÓN DE AGREGACIÓN

LAS FUNCIÓN DE AGREGACIÓN QUE ESTÁN DISPONIBLES DEPENDEN DE LA ETAPA DESDE LA QUE HAYA ABIERTO EL CREADOR SQL. TODAS LAS VARIANTES DE SINTAXIS DE SQL INCLUYEN LAS FUNCIONES DE AGREGACIÓN AVG, COUNT, MAX, MIN, STDDEV Y VARIANCE.

SQLQuery1.sql - (localdb)\...n (52))*

```
SELECT COUNT(*) AS TotalFilas, COUNT(ShipRegion) AS FilasNoNulas,  
MIN(ShippedDate) AS FechaMin, MAX(ShippedDate) AS FechaMax,  
SUM(Freight) AS PesoTotal, AVG(Freight) AS PesoPromedio  
FROM Orders
```

Results Messages

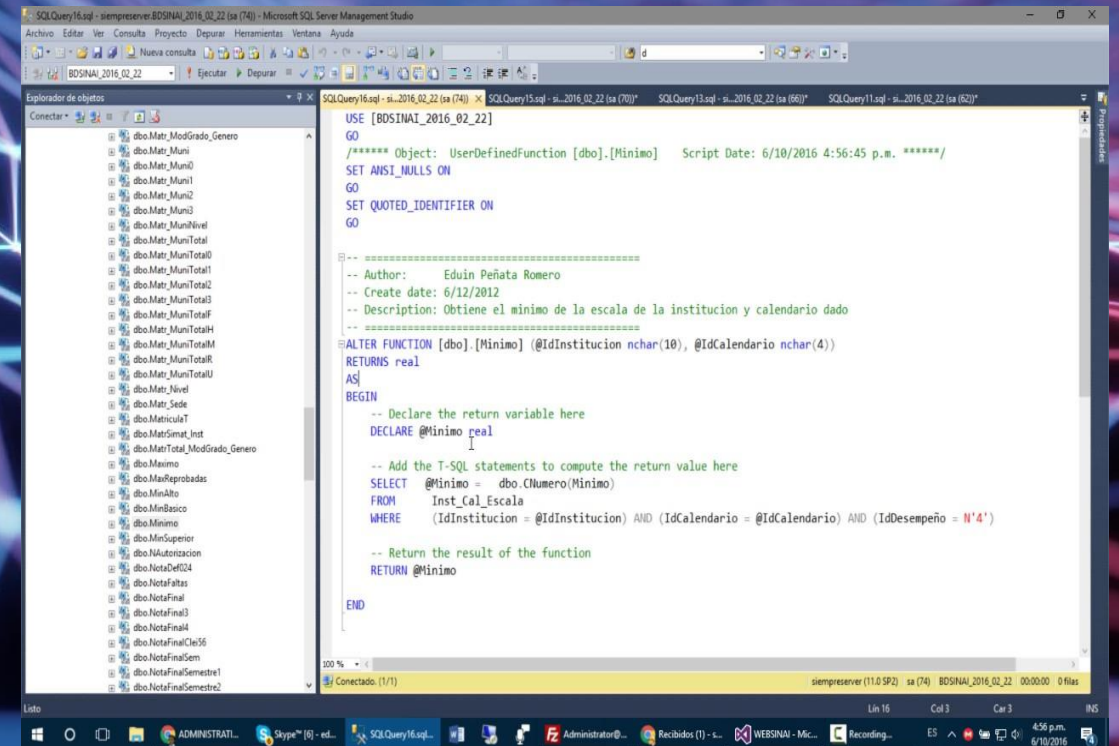
	TotalFilas	FilasNoNulas	FechaMin	FechaMax	PesoTotal	PesoPromedio
1	830	323	1996-07-10 00:00:00.000	1998-05-06 00:00:00.000	64942,69	78,2442

LISTE FUNCIONES DE AGREGACIÓN QUE CONOZCA

Funciones de Agregado	
Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

MENCIONE ALGUNAS FUNCIONES PROPIAS DE SQL-SERVER

LAS FUNCIONES INTEGRADAS EN SQL SERVER SON UNA SERIE DE RUTINAS ALMACENADAS QUE RECIBEN UNA SERIE DE PARÁMETROS CON LOS CUALES REALIZAN OPERACIONES CONCRETAS PARA RETORNAR UN RESULTADO ESPECÍFICO.



```
USE [BOSINAI_2016_02_22]
GO
/***** Object: UserDefinedFunction [dbo].[Minimo]    Script Date: 6/10/2016 4:56:45 p.m. *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

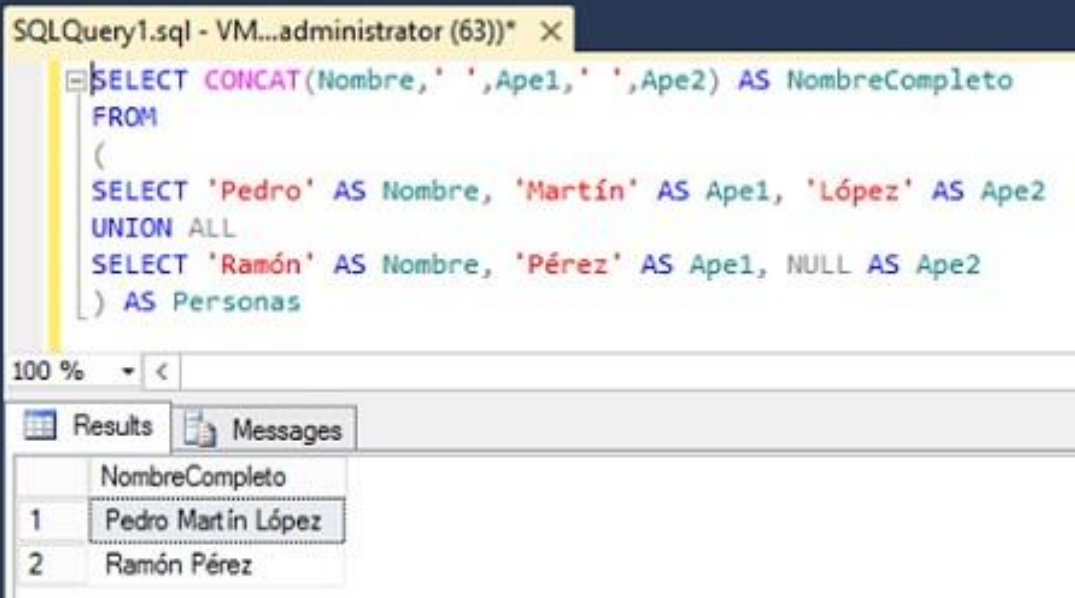
--
-- Author:      Eduin Peña Romero
-- Create date: 6/12/2012
-- Description: Obtiene el minimo de la escala de la institucion y calendario dado
--
ALTER FUNCTION [dbo].[Minimo] (@IdInstitucion nchar(10), @IdCalendario nchar(4))
RETURNS real
AS
BEGIN
    -- Declare the return variable here
    DECLARE @Minimo real

    -- Add the T-SQL statements to compute the return value here
    SELECT @Minimo = dbo.Clumero(Minimo)
    FROM Inst_Cal_Escala
    WHERE (IdInstitucion = @IdInstitucion) AND (IdCalendario = @IdCalendario) AND (IdDesempeño = N'4')

    -- Return the result of the function
    RETURN @Minimo
END
```


PARA QUÉ SIRVE LA FUNCIÓN CONCAT EN SQL-SERVER

CONCAT CONVIERTE IMPLÍCITAMENTE TODOS LOS ARGUMENTOS EN TIPOS DE CADENA ANTES DE LA CONCATENACIÓN. CONCAT CONVIERTE IMPLÍCITAMENTE LOS VALORES NULL EN CADENAS VACÍAS. SI CONCAT RECIBE ARGUMENTOS EN LOS QUE TODOS LOS VALORES SON NULL, DEVOLVERÁ UNA CADENA VACÍA DE TIPO VARCHAR(1).



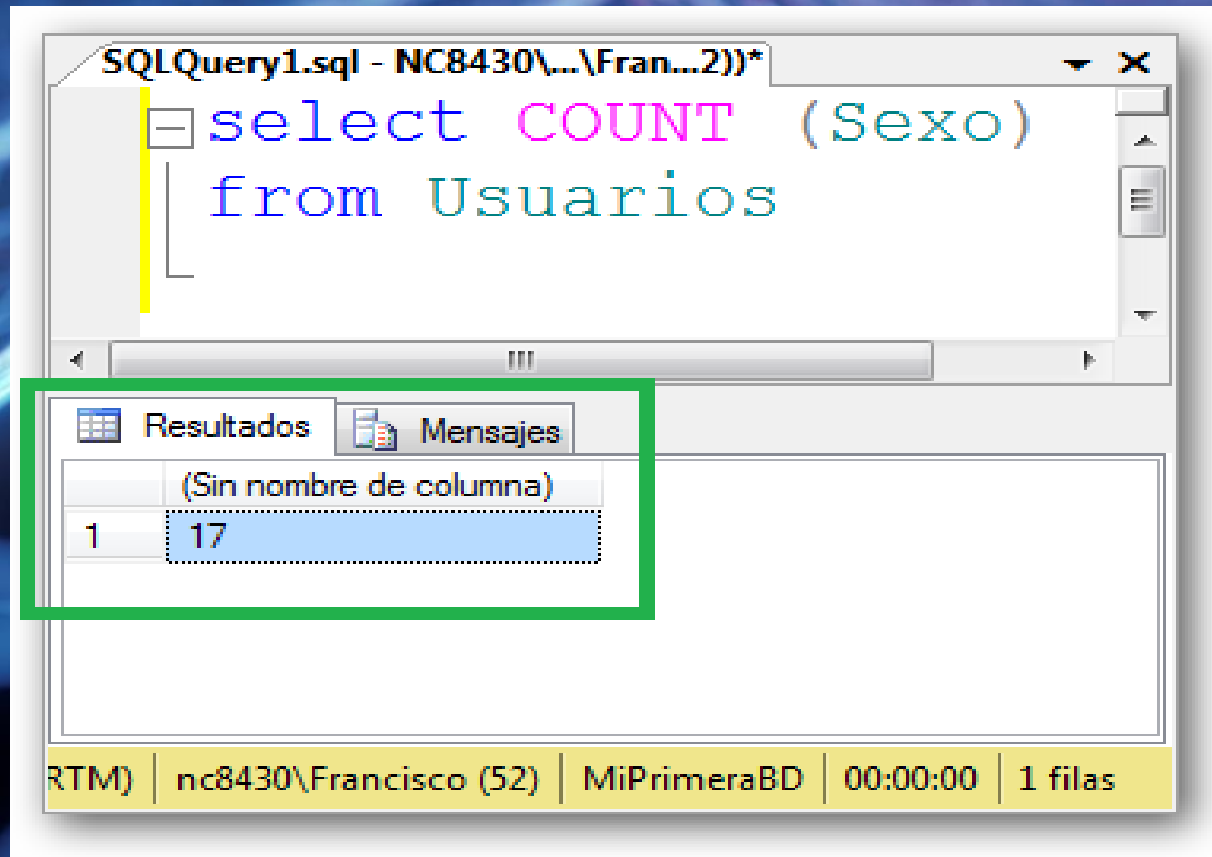
The screenshot shows a SQL query window titled "SQLQuery1.sql - VM...administrator (63))" with a query that uses the CONCAT function. The query is as follows:

```
SELECT CONCAT(Nombre, ' ', Ape1, ' ', Ape2) AS NombreCompleto
FROM
(
  SELECT 'Pedro' AS Nombre, 'Martín' AS Ape1, 'López' AS Ape2
  UNION ALL
  SELECT 'Ramón' AS Nombre, 'Pérez' AS Ape1, NULL AS Ape2
) AS Personas
```

Below the query, the "Results" tab is active, displaying the output of the query. The results are as follows:

	NombreCompleto
1	Pedro Martín López
2	Ramón Pérez

MUESTRA UN EJEMPLO DEL USO DE COUNT



The screenshot shows a SQL query window titled "SQLQuery1.sql - NC8430\...\Fran...2))*". The query text is:

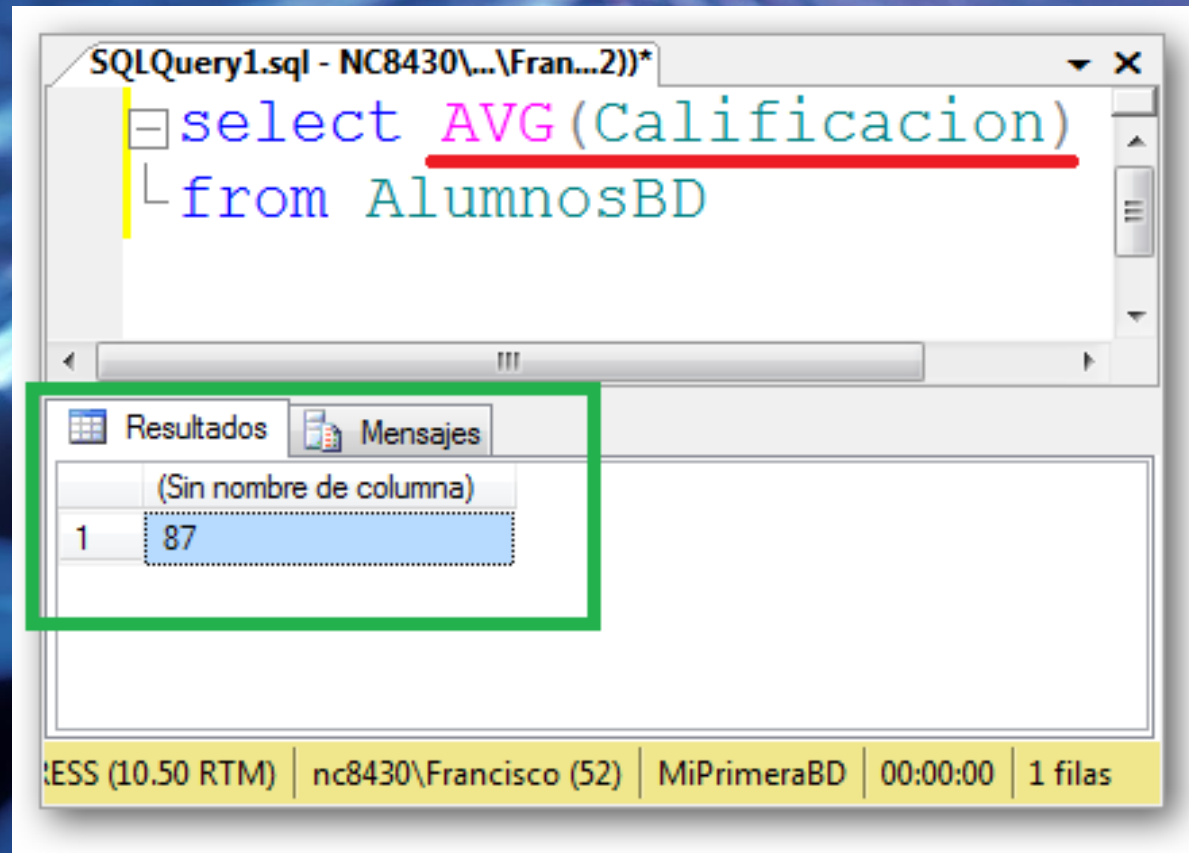
```
select COUNT (Sexo)  
from Usuarios
```

Below the query editor, the "Resultados" (Results) tab is active, displaying a single row of results. The column header is "(Sin nombre de columna)" and the value is 17. The results are highlighted with a green border.

	(Sin nombre de columna)
1	17

At the bottom of the window, the status bar shows: "RTM) | nc8430\Francisco (52) | MiPrimeraBD | 00:00:00 | 1 filas".

MUESTRA UN EJEMPLO DEL USOS DE AVG



The screenshot shows a SQL query window titled "SQLQuery1.sql - NC8430\...\Fran...2))*". The query text is:

```
select AVG(Calificacion)  
from AlumnosBD
```

The AVG(Calificacion) part of the query is underlined in red. Below the query editor, there is a tabbed interface with "Resultados" (Results) and "Mensajes" (Messages). The "Resultados" tab is active, showing a table with one row and one column. The column header is "(Sin nombre de columna)" and the value is 87. The row is numbered 1. The table is highlighted with a green border.

	(Sin nombre de columna)
1	87

At the bottom of the window, the status bar displays: "SQL Server Enterprise Edition (10.50 RTM) | nc8430\Francisco (52) | MiPrimeraBD | 00:00:00 | 1 filas".

MUESTRA UN EJEMPLO DEL USO DE MIN-MAX

SQLQuery5.sql - SQ...Administrador (53)) * X SQL.Ejemplo - dbo.Ventas

```
Use Ejemplo
Go

SELECT MAX(Ventas) AS 'Mayor:' from Ventas
SELECT Min(Ventas) AS 'Menor' from Ventas
SELECT Sum(Ventas) AS 'Total Ventas' from Ventas
SELECT Avg(Ventas) AS 'Media Ventas:' from Ventas
```

100 % <

Resultados Mensajes

	Mayor:
1	5432

	Menor
1	123

	Total Ventas
1	14814

	Media Ventas:
1	1646

MOSTRAR QUE JUGADORES QUE FORMEN PARTE DEL EQUIPO EQU-333

```
select equ.id_equipo, jug.nombres, jug.apellidos  
from jugador as jug  
inner join equipo as equ on jug.id_equipo = equ.id_equipo  
inner join campeonato as cam ON equ.id_campeonato = cam.id_campeonato  
where equ.id_equipo = 'equ-333';
```

100 %

Results Messages

	id_equipo	nombres	apellidos
1	equ-333	Sandra	Solis
2	equ-333	Ana	Mica

- CREAR UNA FUNCIÓN QUE PERMITA SABER CUÁNTOS JUGADORES ESTÁN INSCRITOS.**
- **LA FUNCIÓN DEBE LLAMARSE CREAR UNA FUNCIÓN QUE PERMITA SABER CUÁNTOS JUGADORES ESTÁN INSCRITOS.**
 - **LA FUNCIÓN DEBE LLAMARSE F1_CANTIDADJUGADORES00**

```
CREATE FUNCTION F1_CantidadJugadores()  
RETURNS integer AS  
BEGIN  
    DECLARE @response integer = 0;  
  
    select @response = count(jug.id_jugador)  
    from jugador as jug  
  
    RETURN @response;  
  
END;  
  
select dbo.F1_CantidadJugadores();
```

100 %

Results Messages

	(No column name)
1	5

CREAR UNA FUNCIÓN QUE PERMITA SABER CUÁNTOS JUGADORES ESTÁN INSCRITOS Y QUE SEAN DE LA CATEGORÍA VARONES O MUJERES.

- **LA FUNCIÓN DEBE LLAMARSE F2_CANTIDADJUGADORESPARAMO**
- **LA FUNCIÓN DEBE RECIBIR UN PARÁMETRO “VARONES” O “MUJERES”**

```
CREATE FUNCTION F2_CantidadJugadoresParam(@categoria varchar(8))
RETURNS integer AS
BEGIN
    DECLARE @response integer = 0;

    select @response = count(jug.id_jugador)
    from jugador as jug
    inner join equipo as equ on jug.id_equipo = equ.id_equipo
    where equ.categoria = @categoria

    RETURN @response;
END;

select dbo.F2_CantidadJugadoresParam('VARONES');
select dbo.F2_CantidadJugadoresParam('MUJERES');
```

00 %

Results Messages

	(No column name)
1	3

	(No column name)
1	2

- CREAR UNA FUNCIÓN QUE OBTenga EL PROMEDIO DE LAS EDADES MAYORES A UNA CIERTA EDAD.
- LA FUNCIÓN DEBE LLAMARSE F3_PROMEDIOEDADES()
 - LA FUNCIÓN DEBE RECIBIR COMO PARÁMETRO 2 VALORES.
 - LA CATEGORÍA. (VARONES O MUJERES)
 - LA EDAD CON LA QUE SE COMPARARA (21 AÑOS EJEMPLO)
 - ES DECIR MOSTRAR EL PROMEDIO DE EDADES QUE SEAN DE UNA CATEGORÍA Y QUE SEAN MAYORES A 21 AÑOS.

```
CREATE FUNCTION F3_PromedioEdades(@categoria varchar(8), @edad integer)
RETURNS integer AS
BEGIN
    DECLARE @response integer = 0;

    select @response = avg(jug.edad)
    from jugador as jug
    inner join equipo as equ on jug.id_equipo = equ.id_equipo
    where equ.categoria = @categoria and jug.edad > @edad

    RETURN @response;
END;

select dbo.F3_PromedioEdades('VARONES',18);
select dbo.F3_PromedioEdades('MUJERES',18);
```

100 %

Results Messages

	(No column name)
1	20

	(No column name)
1	21

CREAR UNA FUNCIÓN QUE PERMITA CONCATENAR 3 PARÁMETROS.

■ **LA FUNCIÓN DEBE LLAMARSE F4_CONCATITEMSO**

6

■ **LA FUNCIÓN DEBE DE RECIBIR 3 PARÁMETROS.**

■ **LA FUNCIÓN DEBE DE CONCATENAR LOS 3 VALORES.**

■ **PARA VERIFICAR LA CORRECTA CREACIÓN DE LA FUNCIÓN DEBE MOSTRAR LO SIGUIENTE.**

■ **MOSTRAR LOS NOMBRES DE LOS JUGADORES, EL NOMBRE DEL EQUIPO Y LA SEDE CONCATENADA, UTILIZANDO LA FUNCIÓN QUE ACABA DE CREAR**

```
ALTER FUNCTION F4_ConcatItems(@nombre varchar(30), @equipo varchar(30), @campeonato varchar(30))
RETURNS varchar(30) AS
BEGIN
    DECLARE @response varchar(30);

    select @response = CONCAT_WS(' ', jug.nombres, equ.nombre_equipo, cam.nombre_campeonato)
    from jugador as jug
    inner join equipo as equ on jug.id_equipo = equ.id_equipo
    inner join campeonato as cam ON equ.id_campeonato = cam.id_campeonato
    where jug.nombres = @nombre and equ.nombre_equipo= @equipo and cam.nombre_campeonato= @campeonato;

    RETURN @response;
END;

select dbo.F4_ConcatItems('Carlos', '484 Not found', 'Campeonato Unifranz');
```

83 %

Results Messages

	(No column name)
1	Carlos 484 Not found Campeonat

GENERAR LA SERIE FIBONACCI.

- EL OBJETIVO ES GENERAR UNA FUNCIÓN QUE RETORNE UNA CADENA CON LA SERIE DE LA FIBONACCI.
- LA FUNCIÓN SOLO RECIBE EL VALOR N.

```
ALTER FUNCTION fibonacci5(@num INT)
RETURNS VARCHAR(100) as
BEGIN
    DECLARE @str VARCHAR(255) = '0,1';
    DECLARE @fib1 INT = 0;
    DECLARE @fib2 INT = 1;
    DECLARE @fib3 INT = 0;
    BEGIN
        IF (@num = 1 )
            RETURN @fib1;
        ELSE IF (@num = 2 )
            RETURN CONCAT_WS(',',@fib1,@fib2);
        ELSE
            WHILE (@num > 2)
            begin
                SET @fib3 = @fib1 + @fib2;
                SET @fib1 = @fib2;
                SET @fib2 = @fib3;
                SET @num = @num -1;
                SET @str = CONCAT_WS(',',@str,@fib3);
            END;
            RETURN @str;
        END;
    END;

SELECT dbo.fibonacci5(5);
```

83 %

Results Messages

	(No column name)
1	0,1,1,2,3