


# PROCESUAL HITO 4

BASE DE DATOS I

Presenta:

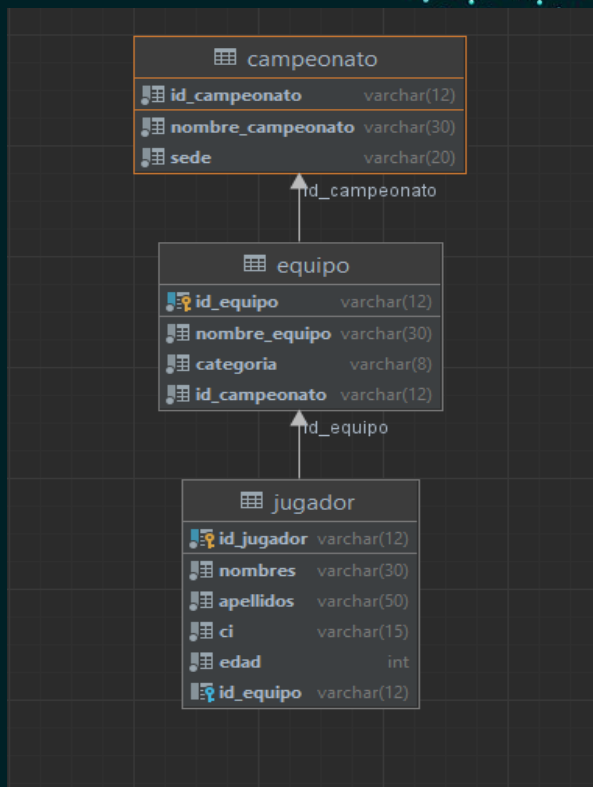
Ilia Araceli Sarzo Laura- SIS14125434  
2doSEMESTRE INGENIERIA DE SISTEMAS

The background features a dark teal gradient. On the left side, there is a dynamic pattern of small, glowing teal dots that form a semi-circular arc. From the center of this arc, several bright teal light rays emanate, spreading outwards and creating a sense of depth and movement. The overall aesthetic is modern and technological.

# **MANEJO DE CONCEPTOS**

## 2.1

# EJEMPLO DE DDL



## EJEMPLO DE DML

	id_jugador	nombres	apellidos	ci	edad	id_equipo
1	jug-222	Pedro	Salas	8997822LP	20	equ-222
2	jug-444	Sandra	Solis	8997844LP	20	equ-333

## 2.3

# EJEMPLO DE INNER JOIN

```
select cam.nombre_campeonato , equi.nombre_equipo, jug.nombres, jug.apellidos
from jugador as jug
inner join equipo as equi 1..n<->1: on jug.id_equipo = equi.id_equipo
inner join campeonato as cam 1..n<->1: on equi.id_campeonato = cam.id_campeonato
where jug.id_equipo = 'equ-222';
```



## ¿Que es un Funcion?

---

es un parámetro opcional En la gestión de bases de datos, una función agregada es una función en la que los valores de varias filas se agrupan como entrada en ciertos criterios para formar un valor único de significado más significativo o medición, como un conjunto, una bolsa o una lista.



2.5

## **FUNCIONES DE REGRESION**

**1.-COUNT**

**2.-MIN**

**3.-MAX**

**4.-SUM**

**5.-AVG**

**6.- CONCAT**

## 2.7

# FUNCION CONCAT

**permite concatenar valores sin límite, de forma que el resultado siempre será una cadena de textos con todos los valores introducidos.**

```
create function sumar_resta_dos_numer(
@num1 int,
@num2 int,
@type varchar(20)
)
returns varchar(20) as
begin
    declare @respuesta int;
    declare @concatenado varchar(20)
    if @type = 'sumar'
    begin
        set @respuesta = @num1 + @num2;
        set @concatenado = CONCAT('SUMA: ',@respuesta)
    end;

    if @type = 'resta'
    begin
        set @respuesta = @num1 - @num2;
        set @concatenado = CONCAT('RESTA: ',@respuesta)
    end;

    return @concatenado;

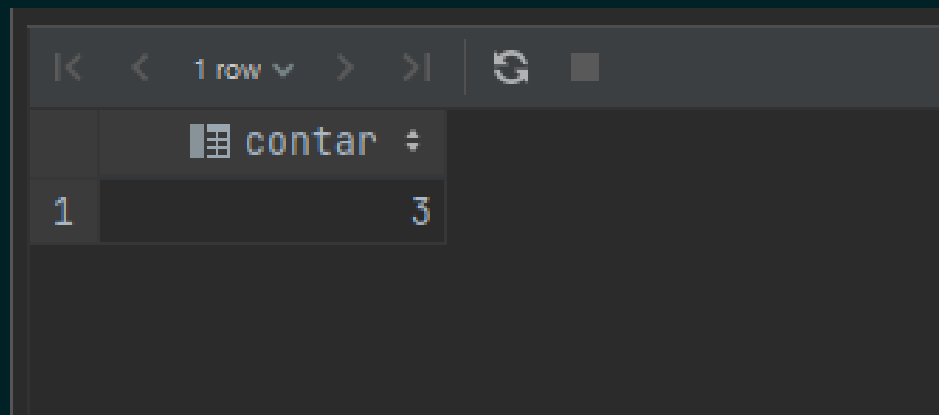
End;
```



## 2.8

# FUNCION COUNT

```
-- mostrar ejemplo con count  
SELECT COUNT(equi.id_equipo) as contar  
FROM equipo as equi
```



The screenshot shows a database query result viewer. At the top, there are navigation controls: a back arrow, a forward arrow, a dropdown menu showing '1 row', and a refresh button. Below these controls is a table with one column named 'contar' and one row with the value '3'.

	contar
1	3

## 2.9

# FUNCION AVG

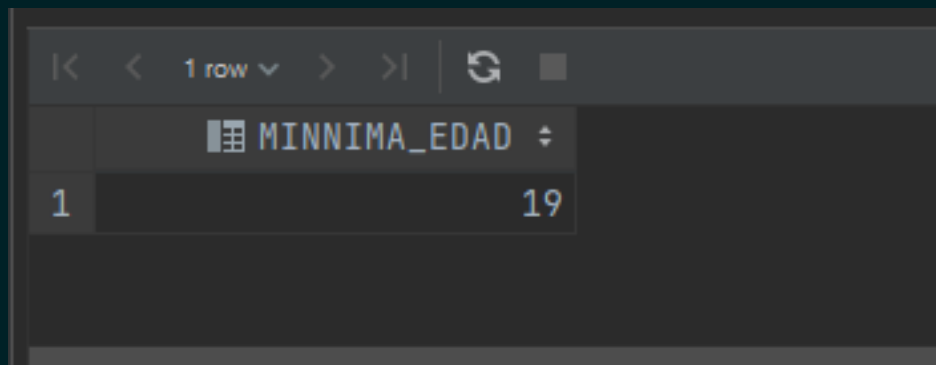
```
-- mostrar un ejemplo con usando avg  
--mostrar un promedio de edad de los jugadores  
select avg(jug.EDAD) as promeido_edad  
from JUGADOR as jug
```

1 row	
	promeido_edad
1	20

## 2.10

# FUNCION MIN

```
-💡 MOSTRAR LA EDAD MINIMA QUE TIENEN LOS JUGADORES  
SELECT MIN(JUG.EDAD) AS MINNIMA_EDAD  
FROM JUGADOR AS JUG;
```



The screenshot shows a database query result interface. At the top, there are navigation controls: a left arrow, a right arrow, a dropdown menu showing '1 row', and a refresh button. Below these controls is a table with one column labeled 'MINNIMA\_EDAD' and one row with the value '19'.


	MINNIMA_EDAD
1	19

## 2.10

# FUNCION MAX

```
-- mostrar edad maxima que se tiene de jugadores  
select MAX(JUG.EDAD) AS MAXIMA_EDAD  
FROM JUGADOR AS JUG
```

1 row	
	MAXIMA_EDAD
1	23

An abstract digital graphic on the left side of the image. It features a dense cluster of small, bright blue dots that form a curved, fan-like shape. From this cluster, several bright blue light rays or beams of light extend outwards towards the left edge of the frame. The background is a solid, dark teal color.

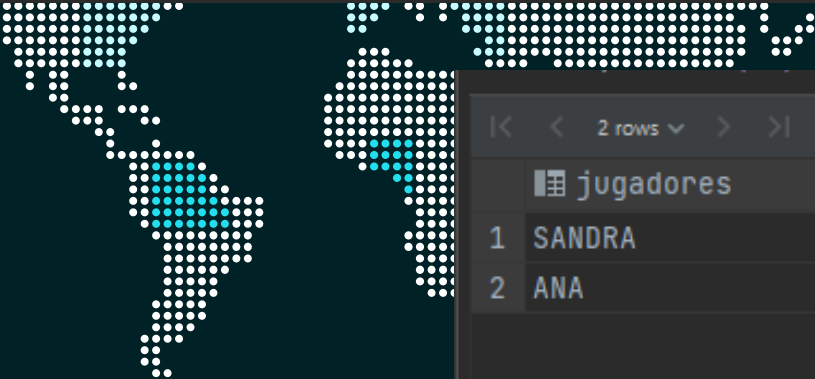
# **MANEJO DE CONSULTAS**



## 3.1

# MOSTRAR QUE JUGADORES QUE FORMAN PARTE DELEQUIPO EQU-333

```
--3.1 Mostrar que jugadores que formen parte del equipo equ-333  
  
select jug.nombres as jugadores  
from jugador as jug  
inner join EQUIPO as equi on jug.ID_EQUIPO = equi.ID_EQUIPO  
WHERE equi.id_equipo = 'equip-333';
```

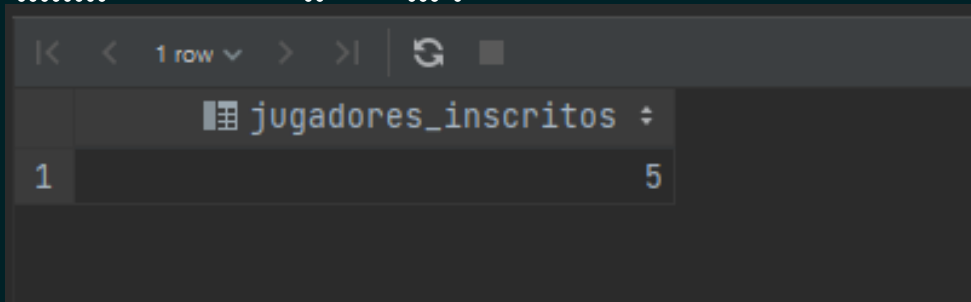


2 rows	
jugadores	
1	SANDRA
2	ANA

## 3.2

# CREAR UNA FUNCION QUE PERMITA SABER CUANTOS JUGADORES ESTAN INSCRITOS

```
--3.2. 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_CantidadJugadores()  
  
create or alter function F1_CantidadJugadores()  
returns integer as  
begin  
    declare @Cantidad_Jugadores integer;  
  
    select @Cantidad_Jugadores = Count(jug.nombres)  
    from jugador as jug  
    return @Cantidad_Jugadores  
end;  
  
select dbo.F1_CantidadJugadores() as jugadores_inscritos
```



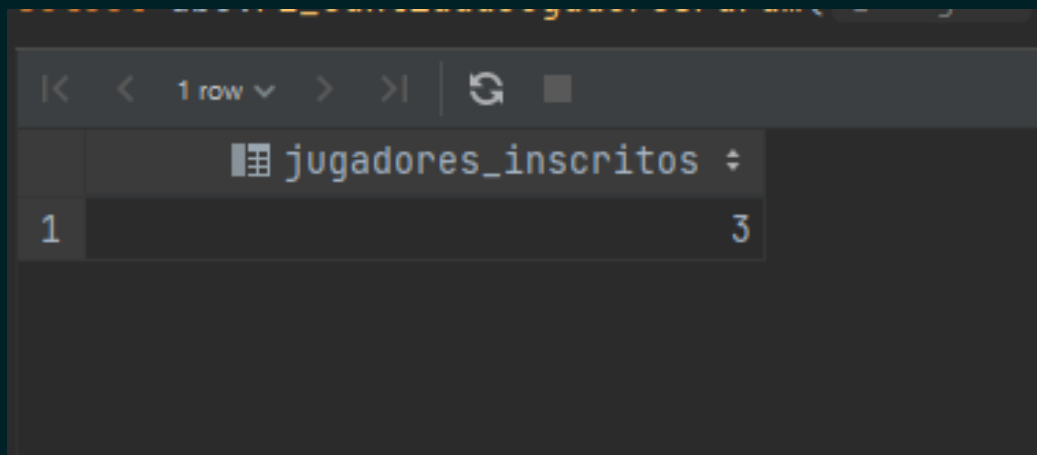
	jugadores_inscritos
1	5

## 3.3

## CREAR UNA FUNCION QUE PERMITA SABER CUANTOS JUGADORES ESTAN INSCRITOS Y QUE SEAN DE CATEGORIA VARONES O MUJERES

```
--3.3 crear una funcion que permita saber cuantos jugadores estan inscritos.  
--la funcion debe llamarse Crear una funcioonn que permita saber cuantos jugadores estan inscritos.  
--La funcion debe llamarse f1_CantidadJugadores()  
  
create function F2_CantidadJugadoresParam(@categoria varchar(30))  
  
returns integer as  
begin  
    declare @Cantidad_Jugadores integer;  
  
    select @Cantidad_Jugadores = Count(jug.nombres)  
    from jugador as jug  
    inner join EQUIPO as equi on jug.ID_EQUIPO = equi.ID_EQUIPO  
    where equi.CATEGORIA = @categoria  
    return @Cantidad_Jugadores  
end;  
  
select dbo.F2_CantidadJugadoresParam( @categoria: 'varones') as jugadores_inscritos
```

### 3.3



A screenshot of a database viewer interface. The table is named 'jugadores\_inscritos'. It has one row with two columns. The first column contains the value '1' and the second column contains the value '3'. The interface includes navigation buttons at the top and a table structure icon next to the table name.

	jugadores_inscritos
1	3

## 3.4

## CREAR UNA FUNCION QUE OBTENGA EL PROMEDIO DE EDADES MAYORES A UNA CIERTA EDAD

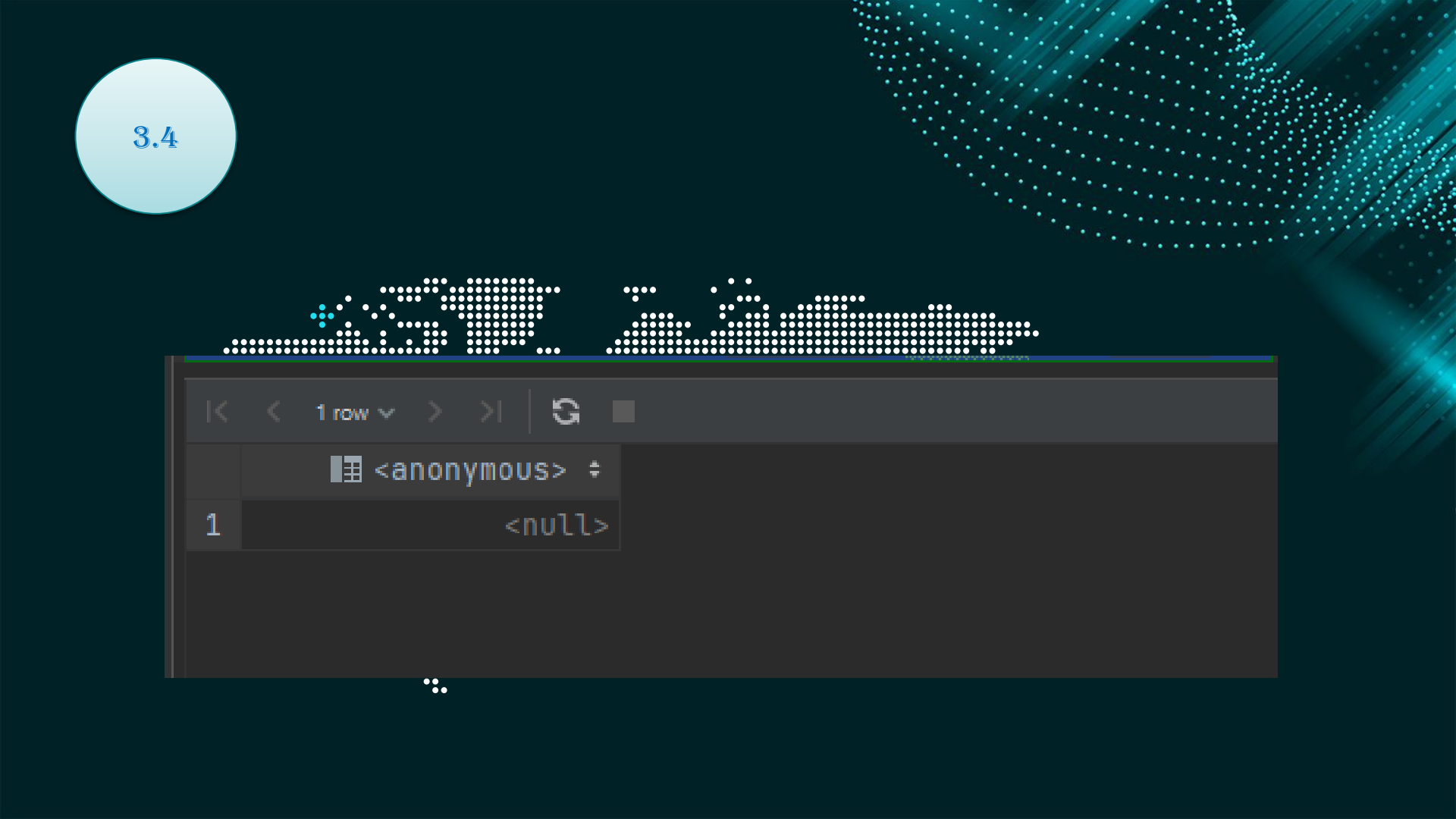
```
-- 3.4 Crear una función que obtenga el promedio de las edades mayores a una cierta edad.  
-- La función debe llamarse F3_PromedioEdades()  
-- La función debe recibir como parámetro 2 valores.  
-- La categoría. (Varones o Mujeres)  
-- La edad con la que se comparara (21 años ejemplo)  
-- Es 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 int)  
returns int as  
begin  
    declare @Prom_edad int  
    select @Prom_edad = avg (jug.edad)  
    from JUGADOR as jug  
    inner join EQUIPO as equi 1..n<->1: on jug.ID_EQUIPO = equi.ID_EQUIPO  
    where equi.CATEGORIA = @categoria and jug.edad > @edad  
    return @Prom_edad  
end
```

```
SELECT DBO.F3_PromedioEdades( @categoria: 'VARONES', @edad: 21)
```



3.4



< < 1 row ▾ > >		↺	■
		<anonymous> ⚙	
1		<null>	

## 3.5

## CREAR UNA FUNCION QUE PERMITA CONCATENAR 3 PARAMETROS

```
-- 3.5 Crear una función que permita concatenar 3 parámetros.
-- ■ La función debe llamarse F4_ConcatItems()
-- ■ 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.

CREATE FUNCTION F4_ConcatItems(
@PAR1 VARCHAR(80),
@PAR2 VARCHAR(80),
@PAR3 VARCHAR(80)
)
RETURNS VARCHAR(80) AS
BEGIN
DECLARE @RESPUESTA VARCHAR(80)
DECLARE @CONCATENADO1 VARCHAR(80)
DECLARE @CONCATENADO2 VARCHAR(80)
DECLARE @CONCATENADO3 VARCHAR(80)
SET @CONCATENADO1 = CONCAT(' NOMBRES DE LOS JUGADORES: ',@PAR1);
SET @CONCATENADO2 = CONCAT(' NOMBRE DE EQUIPO: ',@PAR2);
SET @CONCATENADO3 = CONCAT(' SEDE: ',@PAR3);
SET @RESPUESTA = @CONCATENADO1 + @CONCATENADO2 +@CONCATENADO3

RETURN @RESPUESTA;
END;
```

## 3.5

```
SELECT DBO.F4_Concattems( @PAR1: JUG.NOMBRES, @PAR2: EQUI.NOMBRE_DE_EQUIPO, @PAR3: CAMP.SEDE)
FROM JUGADOR AS JUG
INNER JOIN EQUIPO AS EQUI 1..n<->1: ON JUG.ID_EQUIPO = EQUI.ID_EQUIPO
INNER JOIN CAMPEONATO AS CAMP 1..n<->1: ON EQUI.ID_CAMPEONATO = CAMP.ID_CAMPEONATO
```



5 rows

CSV



<anonymous>

1	NOMBRES DE LOS JUGADORES: CARLOS NOMBRE DE EQUIPO: 404 NOT FOUND SEDE: EL ALTO
2	NOMBRES DE LOS JUGADORES: PEDRO NOMBRE DE EQUIPO: 404 NOT FOUND SEDE: EL ALTO
3	NOMBRES DE LOS JUGADORES: SAUL NOMBRE DE EQUIPO: 404 NOT FOUND SEDE: EL ALTO
4	NOMBRES DE LOS JUGADORES: SANDRA NOMBRE DE EQUIPO: GIRLS UNIFRANZ SEDE: EL ALT
5	NOMBRES DE LOS JUGADORES: ANA NOMBRE DE EQUIPO: GIRLS UNIFRANZ SEDE: EL ALTO

## 3.6

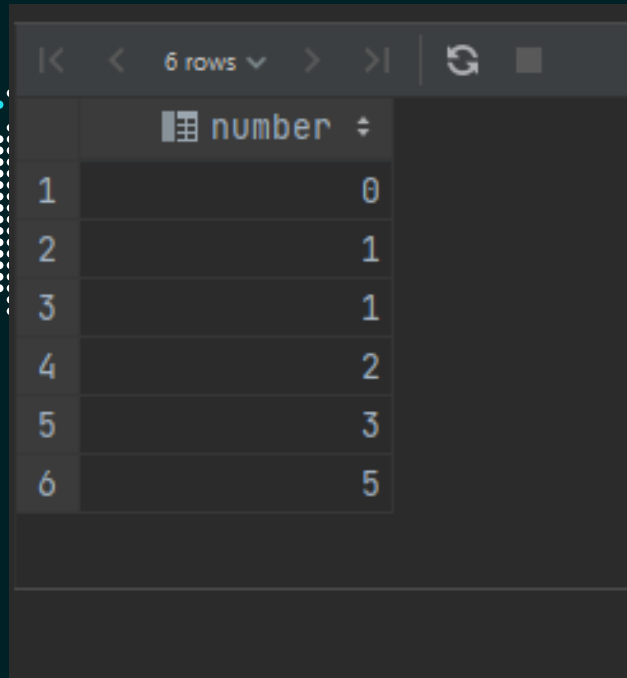
## GENERAR LA SERIE FIBONACCI

```
--3.6. 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.  
-- Comportamiento esperado
```

```
CREATE FUNCTION fn_Fibonacci(@max int)  
RETURNS @numbers TABLE(number int)  
AS  
BEGIN  
    Declare @n1 int = 0,@n2 int =1,@i int=0,@temp int  
    Insert Into @numbers Values(@n1),(@n2)  
    WHILE (@i<=@max-2)  
    BEGIN  
        Insert Into @numbers Values(@n2+@n1)  
        set @temp = @n2  
        Set @n2 = @n2 + @n1  
        Set @n1 = @temp  
        Set @i += 1  
    END  
    RETURN  
END
```

```
Select * from dbo.fn_Fibonacci( @max: 5)
```

3.6



	number
1	0
2	1
3	1
4	2
5	3
6	5



# GRACIAS POR SU ATENCION

---

[eate.iliaaraceli.sarzo.la@unifranz.edu.bo](mailto:eate.iliaaraceli.sarzo.la@unifranz.edu.bo)

UNIFRANZ  
Internacionalízate

INNOVACIÓN  
EN EDUCACIÓN

=X  
FACULTAD  
DE CIENCIAS