

# BASE DE DATOS 1

A futuristic digital cityscape with glowing blue and pink lines, data points, and architectural structures. The scene is set against a dark blue background with floating particles and light trails, creating a high-tech, cybernetic atmosphere.






# TEMA: DEFENSA HITO-4

NOMBRE: LUIS DAVID APAZA  
ROCA  
ASIGNATURA: BASE DE DATOS  
CARRERA: INGENIERIA EN  
SISTEMAS  
DOCENTE: LIC. WILLIAM BARRA  
PAREDES

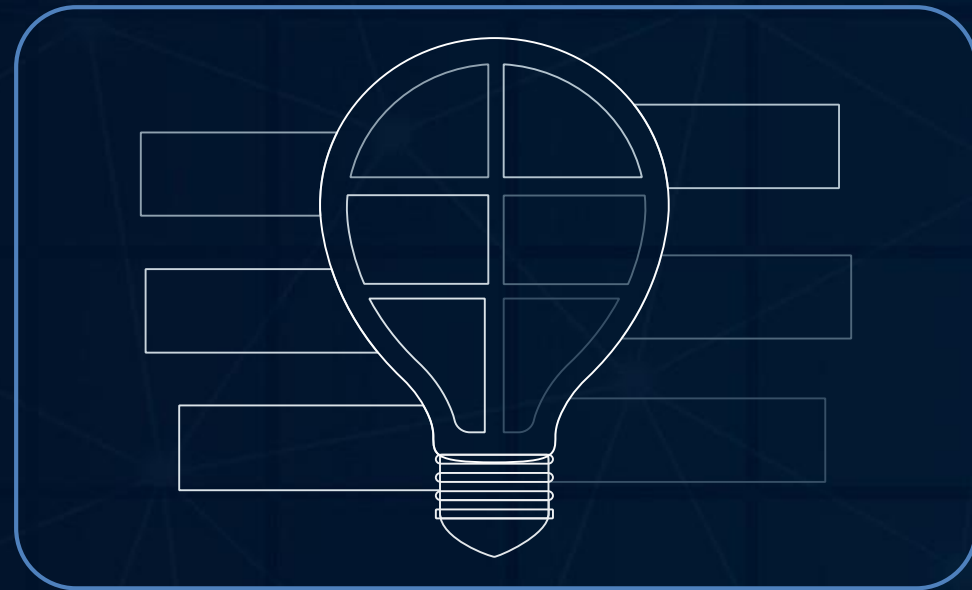


# ÍNDICE

- 1: Mostrar ejemplo DDL.
  - 2: Mostrar ejemplo DML.
  - 3: Para que sirve el INNER JOIN.
  - 4: Defina que es una función de agregación.
  - 5: Listar funciones de agregación que no conozcas.
  - 6: Mencionar algunas funciones propias del SQL-SERVER.
  - 7: Para que sirve la función CONCAT en SQL-SERVER.
  - 8: Mostrar un ejemplo de uso del COUNT.
  - 9: Mostrar un ejemplo de uso del AVG.
  - 10: Mostrar un ejemplo de uso del
- 



# MANEJO DE CONCEPTOS

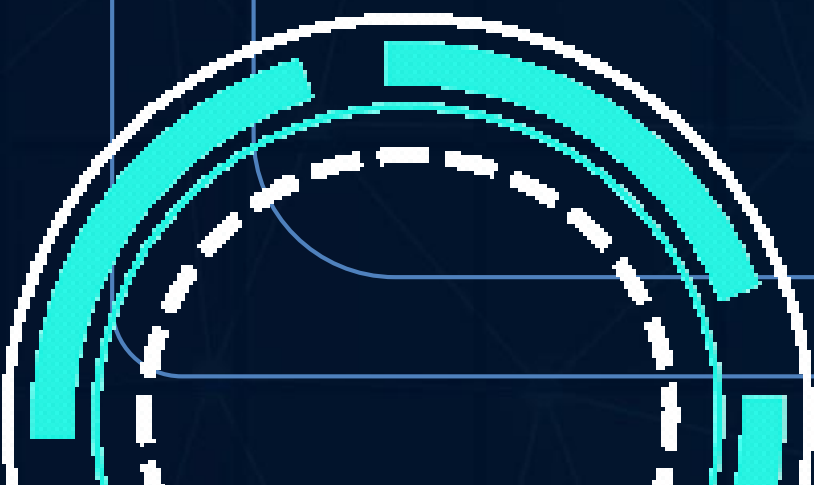


## Ejemplo DDL

```
25
26 -- CREATE
27 CREATE TABLE Person(
28     PersonID INT IDENTITY (1,1) CONSTRAINT PK_PersonID PRIMARY KEY,
29     FirstName NVARCHAR(20),
30     LastName NVARCHAR(25)
31 );
32
33 -- ALTER
34 ALTER TABLE Person
35 ADD BirthDate DATETIME
36
37 -- DROP
38 DROP TABLE Person
39
```

### □ Definición:

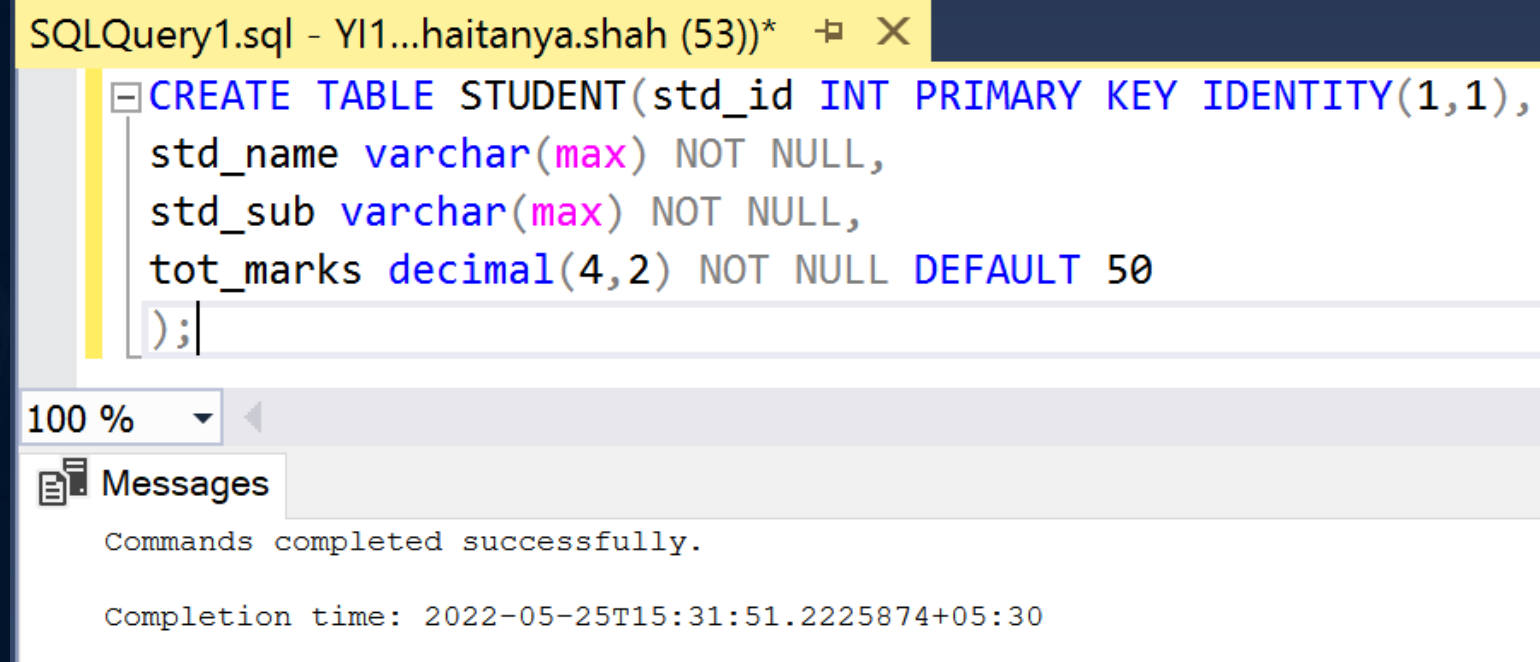
El DDL es un subconjunto de SQL. Es un lenguaje para describir los datos y sus relaciones en una base de datos.





## Ejemplo DML

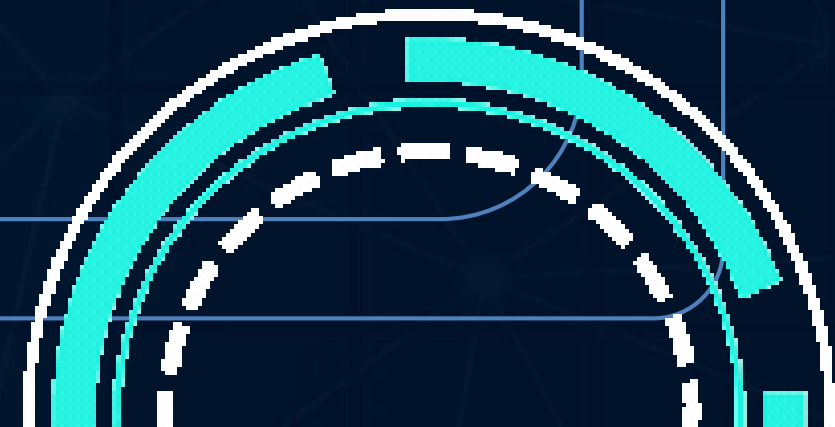
### ❑ Definicion:



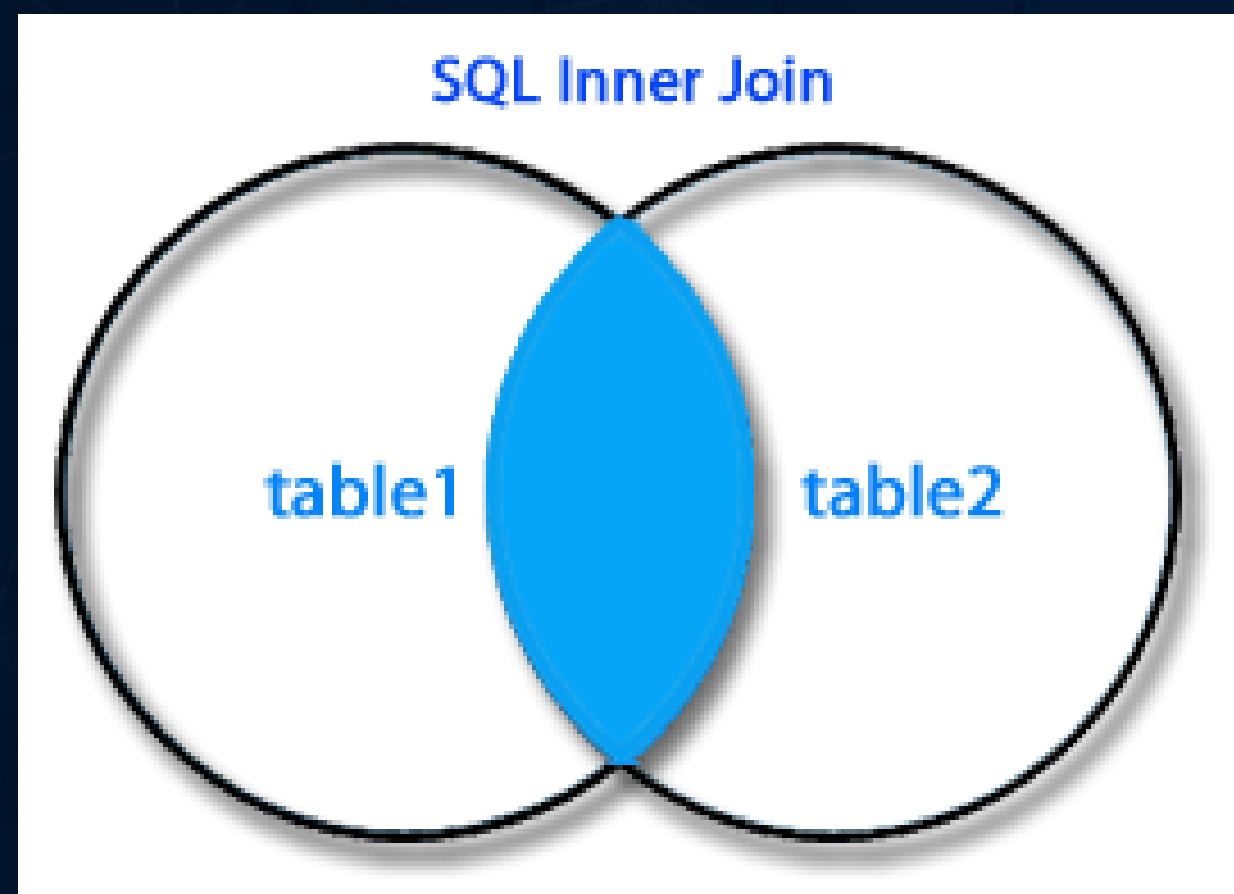
The screenshot shows a SQL query editor window titled "SQLQuery1.sql - YI1...haitanya.shah (53))". The query text is: `CREATE TABLE STUDENT(std_id INT PRIMARY KEY IDENTITY(1,1), std_name varchar(max) NOT NULL, std_sub varchar(max) NOT NULL, tot_marks decimal(4,2) NOT NULL DEFAULT 50);`. Below the query editor, a "Messages" pane shows the status: "Commands completed successfully." and "Completion time: 2022-05-25T15:31:51.2225874+05:30".

```
SQLQuery1.sql - YI1...haitanya.shah (53))*  X
CREATE TABLE STUDENT(std_id INT PRIMARY KEY IDENTITY(1,1),
std_name varchar(max) NOT NULL,
std_sub varchar(max) NOT NULL,
tot_marks decimal(4,2) NOT NULL DEFAULT 50
);
100 %
Messages
Commands completed successfully.
Completion time: 2022-05-25T15:31:51.2225874+05:30
```

El DML constituyen un tipo especial de procedimiento almacenado que se inicia automáticamente cuando tiene lugar un evento de **lenguaje de manipulación de datos (DML)** que afecta a la tabla o la vista definida en el desencadenador.



## ¿Para que sirve el INNER JOIN?



### ❑ Definicion inner join:

Esta clausula busca coincidencia entre dos tablas, en función a una columna que tienen en común. De tal modo que solo la intreseccion se mostrara en los resultados

## ¿Que es función de agregacion?

### □ Definición:

Las funciones de agrupación es SQL nos permite efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos, es decir nos permite obtener medias, máximas ,etc.... Sobre un conjunto de valores.





## Lista de funciones de agregacion

### ❑ LISTA DE FUNCIONES DE AGRUPACION:

- AVG = Devuelve el promedio de los valores de una columna.
- COUNT = Utilizada para devolver el numer de registros de la seleccion.
- SUM = Utilizada para devolver la suma de todos los valores de un campo especificado.
- MAX = Utilizada para devolver el valor mas alto de un campo especificado.
- MIN = Utilizada para devolver el valor mas bajo de un campo especificado.



## Funciones propias del SQL-SERVER

### ❑ FUNCIONES PROPIAS:

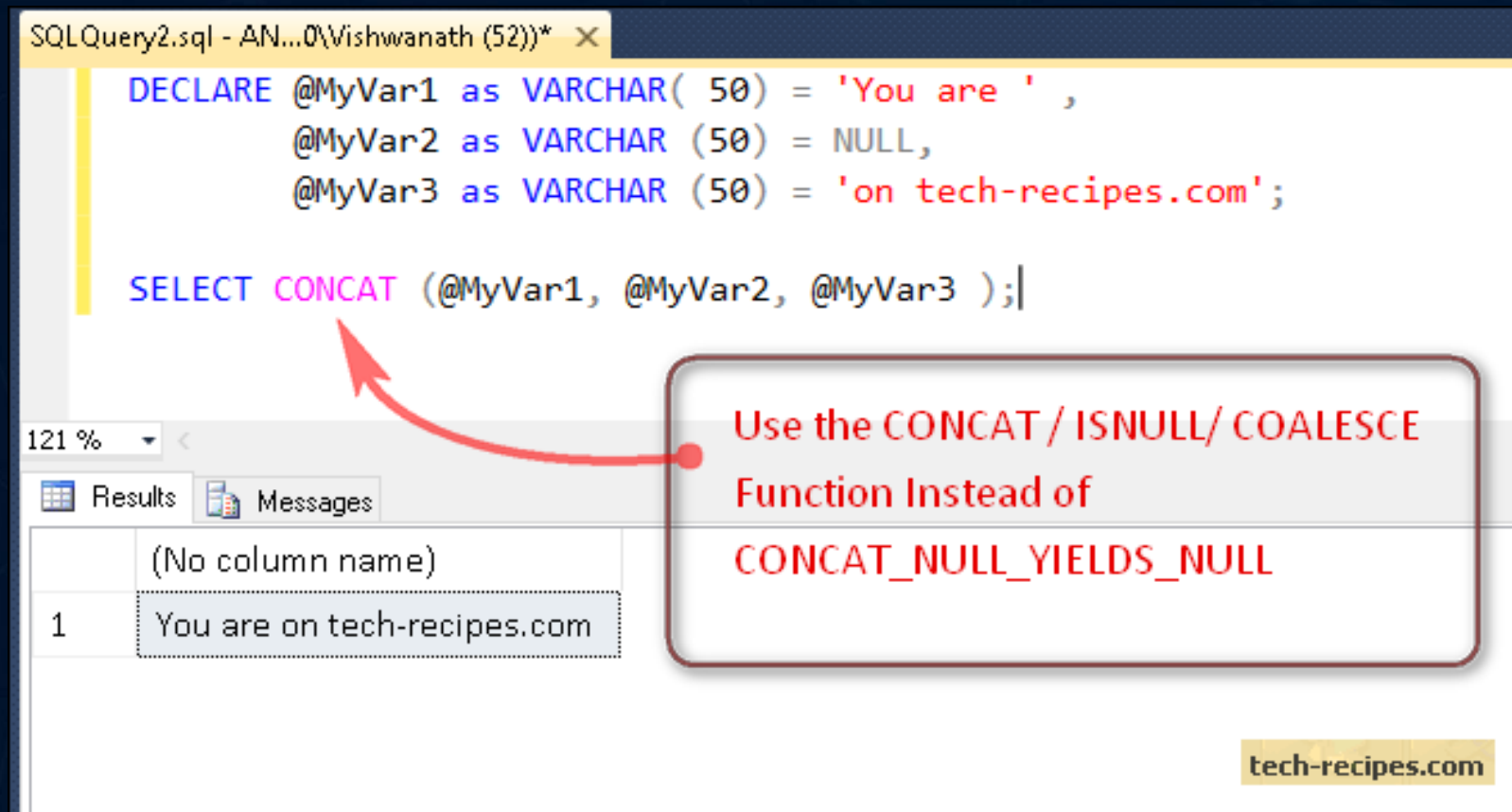
En SQL-SERVER se pueden utilizar una serie de funciones que vienen integradas con el sistema:

- Funciones Analíticas.
- Funciones de Categorías.
- Funciones de Conjunto de filas.
- Funciones Escalares.





## ¿Para que sirve la funcion CONCAT?



```
SQLQuery2.sql - AN...0\Vishwanath (52)) * X
DECLARE @MyVar1 as VARCHAR( 50) = 'You are ' ,
        @MyVar2 as VARCHAR (50) = NULL,
        @MyVar3 as VARCHAR (50) = 'on tech-recipes.com';

SELECT CONCAT (@MyVar1, @MyVar2, @MyVar3 );|
```

121 %

Results Messages

	(No column name)
1	You are on tech-recipes.com

tech-recipes.com

Use the CONCAT / ISNULL / COALESCE Function Instead of CONCAT\_NULL\_YIELDS\_NULL

### ❑ Definicion:

Concat toma un numero variable de argumentos de cadena y los conecta a una sola cadena, necesita un mínimo de dos valores de entrada de lo contrario produce un error en concat.

## Ejemplo de uso del COUNT

```
SELECT COUNT(*) Numero_de_Jugadores_Inscritos  
FROM Campeonato AS camp  
INNER JOIN Equipo AS equi on camp.id_Campeonato = equi.id_campeonato  
INNER JOIN Jugador as jug on equi.id_equipo = jug.id_equipo
```

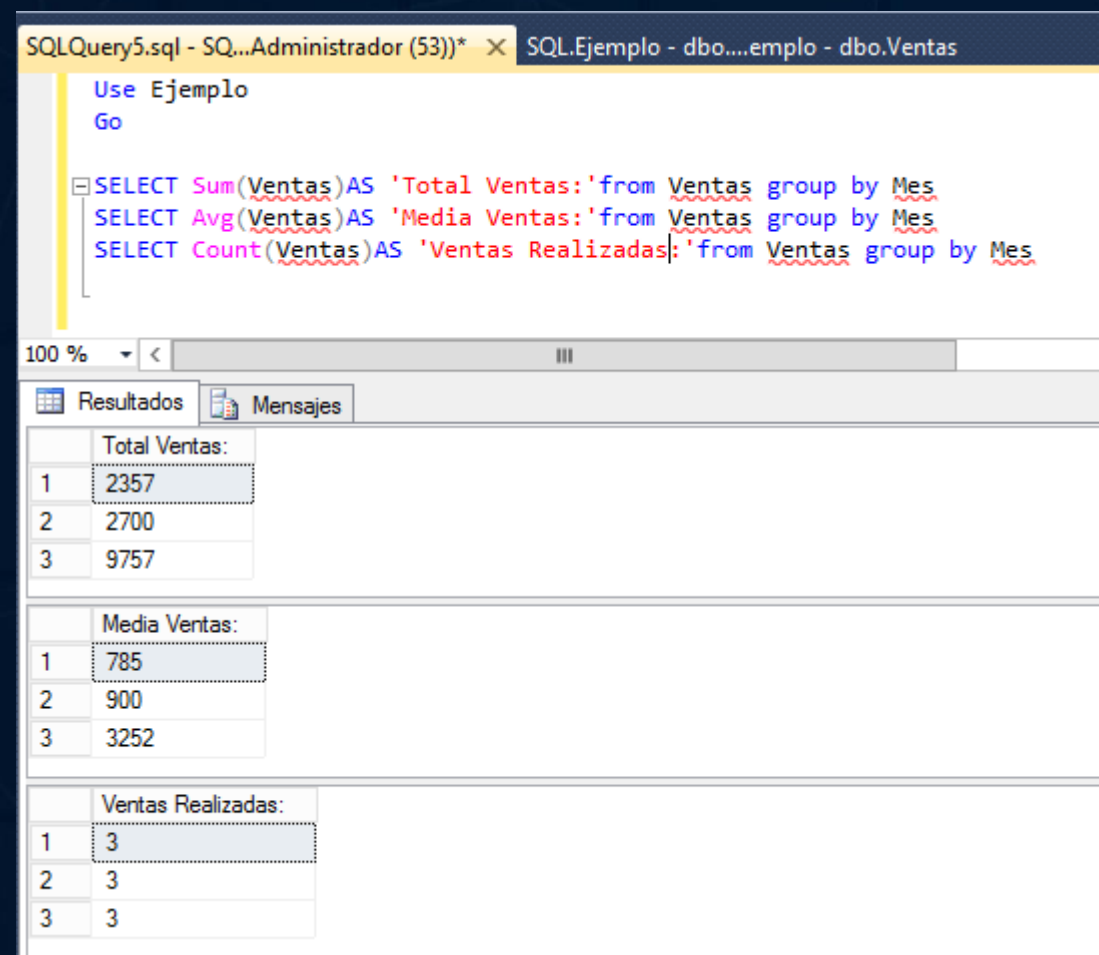
### ❑ Uso del COUNT:

El COUNT() es una función agregada del SQL que devuelve el número de elementos encontrados en un conjunto





## Ejemplo de uso del AVG



The screenshot shows a SQL query window with the following text:

```
Use Ejemplo
Go

SELECT Sum(Ventas)AS 'Total Ventas:'from Ventas group by Mes
SELECT Avg(Ventas)AS 'Media Ventas:'from Ventas group by Mes
SELECT Count(Ventas)AS 'Ventas Realizadas:'from Ventas group by Mes
```

Below the query, the 'Resultados' (Results) tab is active, displaying three tables of data:

Total Ventas:	
1	2357
2	2700
3	9757

Media Ventas:	
1	785
2	900
3	3252

Ventas Realizadas:	
1	3
2	3
3	3

### ❑ Uso del AVG:

La función AVG() de SQL-SERVER es una función agregada que devuelve el valor promedio de un grupo de valores.

## Ejemplo de uso del MIN-MAX

SQLQuery5.sql - SQL...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

### ❑ Uso del MIN-MAX:

La función MIN permite encontrar el valor mínimo de un conjunto de valores.

La función MAX permite encontrar el valor maximo de un conjunto de valores.





# ÍNDICE

**2.1:** Mostrar que jugadores que fromen parte del equipo equ-333.

**2.2:** 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.

**2.3:** Crear una función que permita saber cuántos jugadores están inscritos y que sean de la categoría varones o mujeres.


- La función debe llamarse F2\_CantidadJugadoresParam.
- La función debe recibir un parámetro “Varones” o “Mujeres”.

**2.4:** 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.

**2.5:** 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.

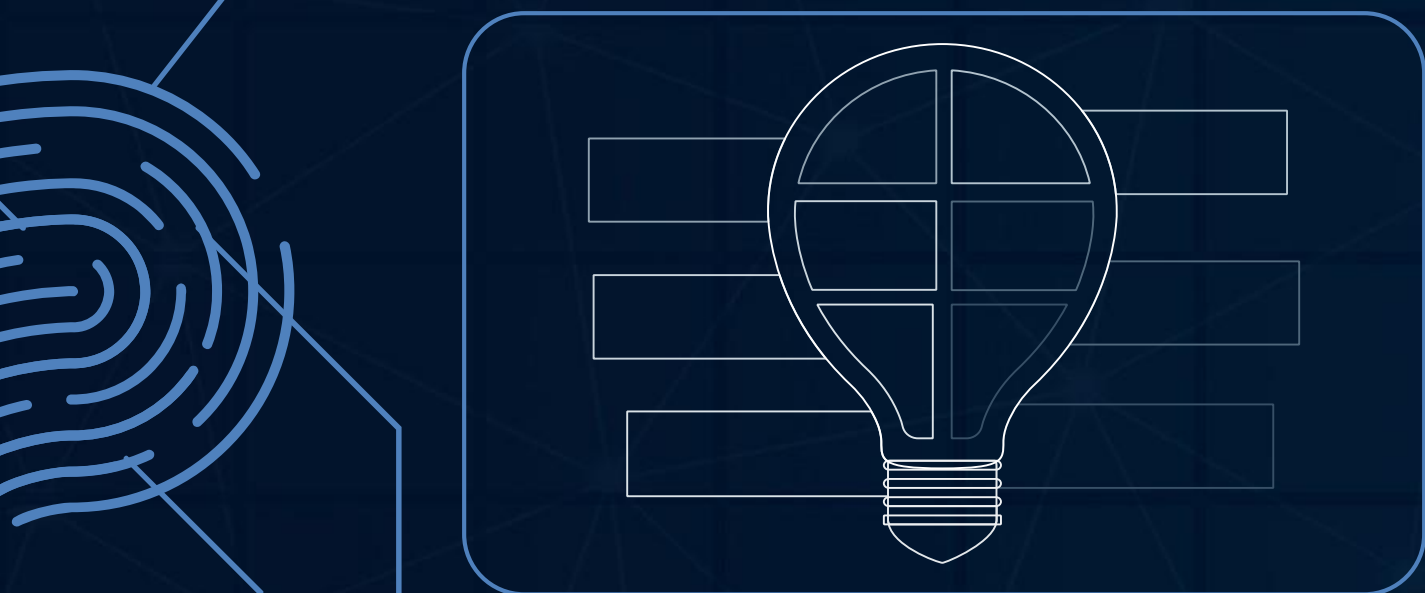
## 2.6: Generar la serie Fibonacci.

- El objetivo es generar una función que retorne una cadena con la serie de la fibonacci.
  - La función solo recibe el valor N.
- 





# MANEJO DE CONSULTAS



## 2.1

Mostrar que jugadores que formen parte del equipo equ-333

--1.-Mostrar que jugadores que formen parte del equipo equ-333

```
select jug.*  
from jugador as jug  
inner join equipo as eq on jug.id_equipo =eq.id_equipo  
where eq.id_equipo = 'equ-333'
```

121 %

Results Messages

	id_jugador	nombres	apellidos	ci	edad	id_equipo
1	jug-444	Sandra	Solis	8997844LP	20	equ-333
2	jug-555	Ana	Mica	8997855LP	23	equ-333

## 2.2

### FUNCION JUGADORES INSCRITOS

--2.-Crear una función que permita saber cuántos jugadores están inscritos

```
create function F1_CantidadJugadores()  
returns integer as  
begin  
    declare @resultado integer = 0  
    select @resultado = count(jug.id_jugador)  
    from jugador as jug  
  
    return @resultado  
end
```

```
select dbo.F1_CantidadJugadores() as 'Jugadores inscritos'
```

121 %

Results Messages

Jugadores inscritos	
1	5



## 2.3

### FUNCION CUANTOS JUGADORES VARON O MUJER

```
--3.-Crear una función que permita saber cuántos jugadores están inscritos y que sean de la categoría varones o mujeres

create function F2_CantidadJugadoresParam(@categoria varchar(20))
returns integer as
begin
    declare @resultado integer = 0
    select @resultado = count(jug.id_jugador)
    from jugador as jug
    inner join equipo as eq on eq.id_equipo = jug.id_equipo
    where eq.categoria = @categoria

    return @resultado
end

select dbo.F2_CantidadJugadoresParam('Mujeres') as 'Registrados'
```

0 %

Results Messages

Registrados
2

```
--3.-Crear una función que permita saber cuántos jugadores están inscritos y que sean de la categoría varones o mujeres

create function F2_CantidadJugadoresParam(@categoria varchar(20))
returns integer as
begin
    declare @resultado integer = 0
    select @resultado = count(jug.id_jugador)
    from jugador as jug
    inner join equipo as eq on eq.id_equipo = jug.id_equipo
    where eq.categoria = @categoria

    return @resultado
end

select dbo.F2_CantidadJugadoresParam('varones') as 'Registrados'
```

100 %

Results Messages

Registrados
3

## 2.4

### FUNCION PROMEDIO DE EDADES

--4.-Crear una función que obtenga el promedio de las edades mayores a una cierta edad

```
create function F3_PromedioEdades(@categoria varchar(20), @edad integer)
returns integer as
begin
    declare @resultado integer = 0
    select @resultado = avg(jug.edad)
    from jugador as jug
    inner join equipo as eq on jug.id_equipo = eq.id_equipo
    where eq.categoria = @categoria and jug.edad > @edad

    return @resultado
end

select dbo.F3_PromedioEdades('mujeres', 21) as 'Promedio de edad'
```

100 %

Results Messages

	Promedio de edad
1	23

## 2.5

### FUNCION CON 3 PARAMETROS

```
SQLQuery2.sql - DE...5PSA\Usuario (52))* ✕
-- 5.-Crear una función que permita concatenar 3 parámetros.

alter function F4_ConcatItems(@categoria varchar(20), @id_equipo varchar(12), @id_camp varchar(12))
returns
begin
    declare @resultado varchar(50)
    select @resultado=CONCAT(jug.nombres, ' forma parte del equipo ', eq.nombre_equipo, ' , y juega en la sede ', cam.sede)
    from jugador as jug
    inner join equipo as eq on eq.id_equipo = jug.id_equipo
    inner join campeonato as cam on eq.id_campeonato = cam.id_campeonato
    where jug.id_equipo = @id_equipo and cam.id_campeonato = @id_camp and eq.categoria = @categoria

    return @resultado

end

select dbo.F4_ConcatItems('Varones', 'equ-222', 'camp-111') as 'Resultado'
```

Incorrect syntax: 'ALTER FUNCTION' must be the only statement in the batch.

100 %

Results Messages

	Resultado
1	Carlos forma parte del equipo 404 Not Found . y ju



## 2.6

### SERIE FIBONACCI

```
SQLQuery2.sql - DE...5PSA\Usuario (52))*  X
--
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;

select dbo.Fibonacci5(5);
```

100 %

Results Messages

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

GRACIAS

