

# EVALUACION PROCESUAL HITO IV

**DOCENTE.-** WILLIAM RODDY BARRA PAREDES

**PRESENTA:** WILLIAM GARRI ESCALANTE JIHUAÑA

**MATERIA.-** BASE DE DATOS I

**FECHA:** 28 / 11 / 2022



# INDICE

## MANEJO DE CONCEPTOS

2.1	Muestra un ejemplo de DDL.
2.2	Muestra un ejemplo de DML
2.3	Para que drive INNER JOIN.
2.4	Defina que es una función de agregación.
2.5	Liste funciones de agregación que conozca.
2.6	Mencione algunas funciones propias de SQL-Server.
2.7	Para qué sirve la función CONCAT en SQL-Server
2.8	Muestra un ejemplo del uso de COUNT
2.9	Muestra un ejemplo del usos de AVG
2.10	Muestra un ejemplo del uso de MIN-MAX

# INDICE

## MANEJO DE CONSULTAS

3.1	Mostrar que jugadores que formen parte del equipo equ-333
3.2	Crear una función que permita saber cuántos jugadores están inscritos.
3.3	Crear una función que permita saber cuántos jugadores están inscritos y que sean de la categoría varones o mujeres.
3.4	Crear una función que obtenga el promedio de las edades mayores a una cierta edad.
3.5	Crear una función que permita concatenar 3 parámetros.
3.6	Generar la serie fibonacci.



# MANEJO DE CONCEPTOS



2.1.0



## Muestra un ejemplo de DDL



### DDL ( Lenguaje de definición de datos ):

DDL se utilizan para describir una base de datos, para definir su estructura, para crear sus objetos y para crear los sub objetos de la tabla.

```
create table campeonato(  
  id_campeonato varchar(20) primary key not null,  
  nombre_campeonato varchar(30) not null,  
  sede varchar(20) not null,  
);
```

2.2.0



## Muestra un ejemplo de DML



- DDL (Lenguaje de manipulación de datos):

DML se utilizan para controlar la información contenida en la base de datos.

```
insert into campeonato(id_campeonato, nombre_campeonato,sede)
values('camp-111', 'Campeonato Unifranz', 'El Alto'),
      ('camp-222', 'Campeonato Unifranz', 'Cochabamba')
```

## 2.3.0

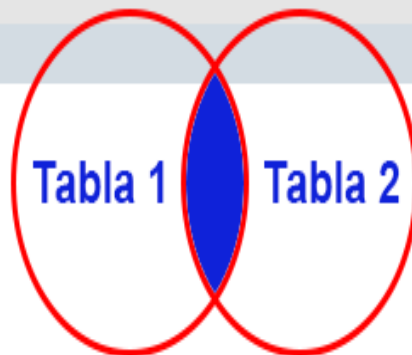


### Para que sirve INNER JOIN



```
SELECT *  
FROM Tabla1 a  
INNER JOIN Tabla2 b ON a.id = b.id  
LIMIT 0, 30
```

### INNER JOIN MySQL



+ Opciones

id	letra	id	valores
3	Tres	3	Tres
4	Cuatro	4	Cuatro

#### ❑ “ INNER JOIN ”:

Inner Join se utiliza para poder combina los registros de dos tablas, si hay valores iguales en las tablas seleccionadas se mostrará los registros que son iguales de ambas tablas.

## 2.4.0



### Defina que es una función de agregación



#### ■ FUNCIONES DE AGREGACION:

Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos.

MIN  
STDEV  
SUM AVG  
MAX COUNT



## 2.5.0



### Liste funciones de agregación que conozca



- ❑ **COUNT:** devuelve el número total de filas seleccionadas por la consulta.
- ❑ **MIN:** devuelve el valor mínimo del campo que especifiquemos.
- ❑ **MAX:** devuelve el valor máximo del campo que especifiquemos.
- ❑ **SUM:** suma los valores del campo que especifiquemos. Sólo se puede utilizar en columnas numéricas.
- ❑ **AVG:** devuelve el valor promedio del campo que especifiquemos. Sólo se puede utilizar en columnas numéricas.



## 2.6.0



### Mencione algunas funciones propias de SQL-Server



#### FUNCIONES



SQL Server proporciona al usuario la posibilidad de definir sus propias funciones, conocida como UDF (user defined functions). Existen tres tipos de funciones.

Estas son:

- Funciones escalares.
- Funciones en línea.
- Funciones en línea de múltiples sentencias.

## 2.7.0



### Para qué sirve la función CONCAT en SQL-Server



#### ■ CONCAT en SQL-Server

La función CONCAT() agrega dos o más cadenas juntas. Si CONCAT recibe argumentos en los que todos los valores son NULL, devolverá una cadena vacía de tipo varchar(1)

## 2.8.0



### Muestra un ejemplo del uso "COUNT"



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

```
select COUNT (Edad)
from Usuarios
where Edad > '25'
```

The query is executed, and the results are displayed in a table with one row and one column. The result is 7. The table is highlighted with a green box.

(Sin nombre de columna)	
1	7

At the bottom of the window, the status bar shows: <PRESS (10.50 RTM) | nc8430\Francisco (52) | MiPrimeraBD | 00:00:00 | 1 filas

## 2.9.0



# Muestra un ejemplo del uso "AVG "



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

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

The word "AVG" is highlighted in pink and underlined in red. Below the query, the "Resultados" (Results) tab is active, showing a single row with the value 87. The "Mensajes" (Messages) tab is also visible. The status bar at the bottom indicates "SQL Server Enterprise Edition (10.50 RTM) | nc8430\Francisco (52) | MiPrimeraBD | 00:00:00 | 1 filas".

	(Sin nombre de columna)
1	87

SQL Server Enterprise Edition (10.50 RTM) | nc8430\Francisco (52) | MiPrimeraBD | 00:00:00 | 1 filas

2.10.0



## Muestra un ejemplo del uso “ MIN-MAX ”



```
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

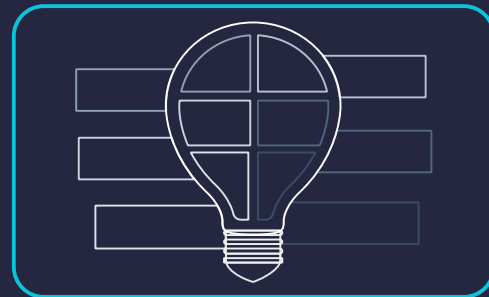
Mensajes

Mayor:	
1	5432
Menor	
1	123
Total Ventas	
1	14814
Media Ventas:	
1	1646



# MANEJO DE

# CONSULTAS



### 3.1.0

Mostrar que jugadores que formen parte del equipo equ-333

```
--3.1. Mostrar que jugadores que formen parte del equipo equ-333
```

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



## 3.2.0

Crear una función que permita saber cuántos jugadores están 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 function F1_CantidadJugadores()  
returns integer as  
begin  
    declare @resultado integer = 0  
    select @resultado = COUNT(jug.id_jugador)  
    from jugadores as jug  
    return @resultado  
end  
  
select dbo.F1_CantidadJugadores() as 'jugadores están inscritos'
```

### 3.3.0

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

```
--3.3. 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"

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

select dbo.F2_CantidadJugadoresParam('Varones')
```

### 3.4.0

Crear una función que obtenga el promedio de las 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(20), @edad integer)  
returns integer as  
begin  
    declare @resultado integer=0  
    select @resultado = AVG(jug.edad)  
    from jugadores 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)
```

## 3.5.0

### Crear una función que permita concatenar 3 parámetros.

```
--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(@categoria varchar(20), @id_equipo varchar(20), @id_cam varchar(20))
returns varchar(60) as
begin
    declare @resultado varchar(100)
    select @resultado = CONCAT(jug.nombres, ' es del equipo ', eq.nombre_equipo, ' ,juega en la sede ', cam.sede)
    from jugadores as jug
    inner join equipo as eq on jug.id_equipo = eq.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_cam and eq.categoria = @categoria
    return @resultado
end

select dbo.F4_ConcatItems('varones', 'equ-222', 'cam-111')
```

### 3.6.0

## 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 fibonacci(@cantidad integer)  
returns @numbers table(NUMBER int)  
as  
begin  
    declare @n1 integer=0, @n2 integer=1, @i integer=0, @temp integer  
    insert into @numbers values (@n1),(@n2)  
    while (@i<=@cantidad-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.fibonacci(8)
```