

ACTIVIDAD PROCESUAL H4



BASE DE DATOS



ingeniería de sistemas - unifrnaz

CRISTIAN MACHICADO
FLORES

```
= CREATE DATABASE defensa_hito3_unifranzitos;  
USE defensa_hito3_unifranzitos;  
= CREATE TABLE campeonato(  
  id_campeonato Varchar (25) PRIMARY KEY,  
  nombre_campeonato VARCHAR(100),  
  sede varchar (30)  
);  
= CREATE TABLE equipo(  
  id_equipo Varchar (25) PRIMARY KEY,  
  nombre_equipo VARCHAR(100),  
  categoria varchar (30),  
  id_campeonato varchar(30)  
);  
= CREATE TABLE jugador(  
  id_jugador Varchar (25) PRIMARY KEY,  
  nombres VARCHAR(100),  
  apellidos VARCHAR(100),  
  ci VARCHAR(50),  
  edad INTEGER,  
  id_equipo varchar(50)  
);
```

diseño de base de datos

1.2 REGISTROS

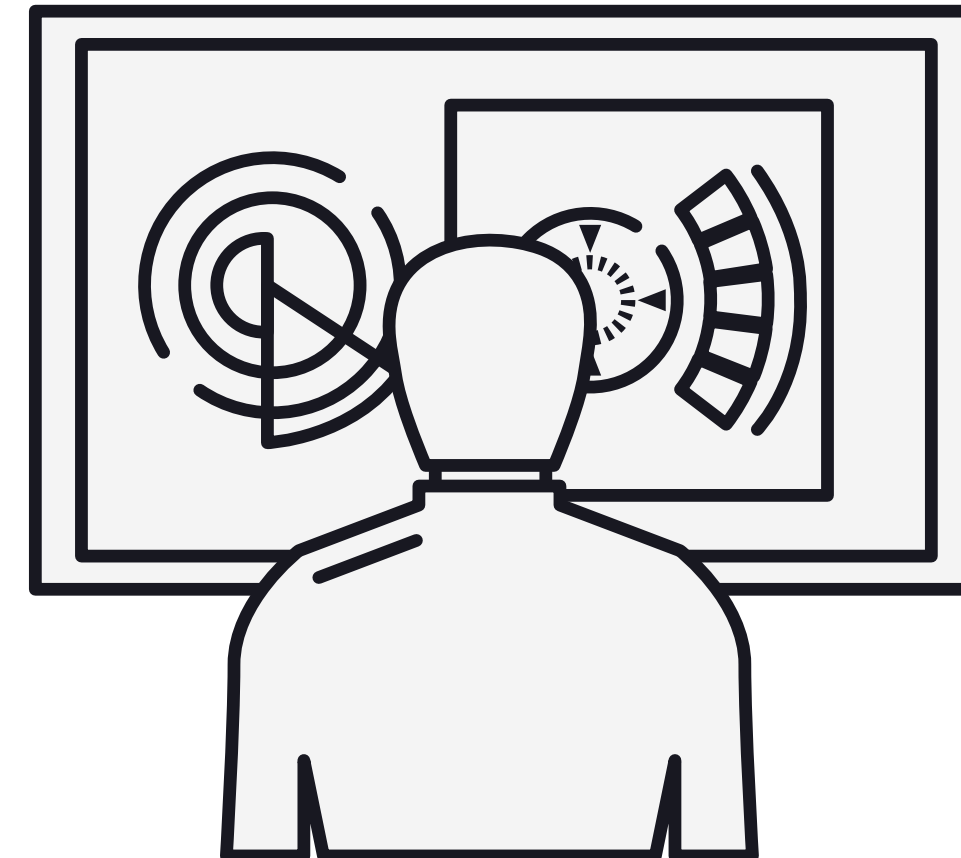
	id_juga...	nombres	apellid...	ci	ed...	id_equi...
1	jug-111	CARLOS	VILLA	8997811LP	19	equ-222
2	jug-222	PEDRO	SALAS	8997822LP	20	equ-222
3	jug-333	SAUL	ARAJ	8997833LP	21	equ-222
4	jug-444	SANDRA	SOLIS	8997844LP	20	equ-333
5	jug-555	ANA	MICA	8997855LP	23	equ-333

	id_equi...	nombre_equi...	categoria	id_campeon...
1	equ-111	Google	VARONES	camp-111
2	equ-222	484 Not Found	VARONES	camp-111
3	equ-333	Girls unifranz	MUJERES	camp-111

	id_campeon...	nombre_campeonato	sede
1	camp-111	campeonato unifranz	El alto
2	camp-222	campeonato unifranz	Cochabamba



2.1 DIAGRAMA



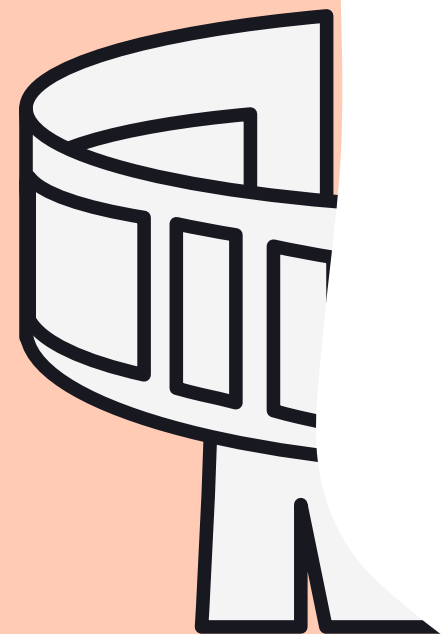
2.2 QUE ES DDL Y DML

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.

```
SELECT * from jugador
```

las sentencias DML Utilizan instrucciones de SQL, permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

```
truncate table equipo
```



2.3 INSTRUCCIÓN INNER JOIN.

En una instrucción SQL, inner join Combina los registros de dos tablas si hay valores coincidentes en un campo común.



```
SELECT j1.nombres, j1.apellidos, j1.edad, j1.id_equipo, eq.nombre_equipo, ca.sede from jugador as j1
INNER JOIN equipo as eq on eq.id_equipo=j1.id_equipo
INNER JOIN campeonato as ca on ca.id_campeonato=eq.id_campeonato
where j1.id_equipo = 'equ-333' and j1.edad>20 and ca.sede='El alto';
```

	nombr...	apellid...	ed...	id_equi...	nombre_equ...	sede
1	ANA	MICA	23	equ-333	Girls unifranz	El alto

2.4 DEFINA QUE ES UNA FUNCION DE AGREGACIÓN.

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



```
SELECT COUNT(*) FROM equipo;
```

100 % <

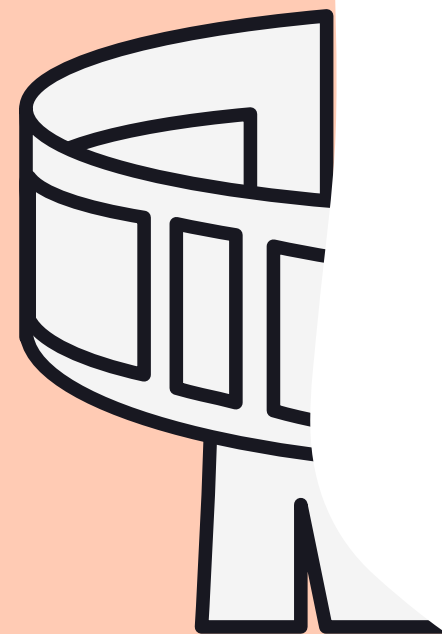
Results Messages

	(No column na...
1	3

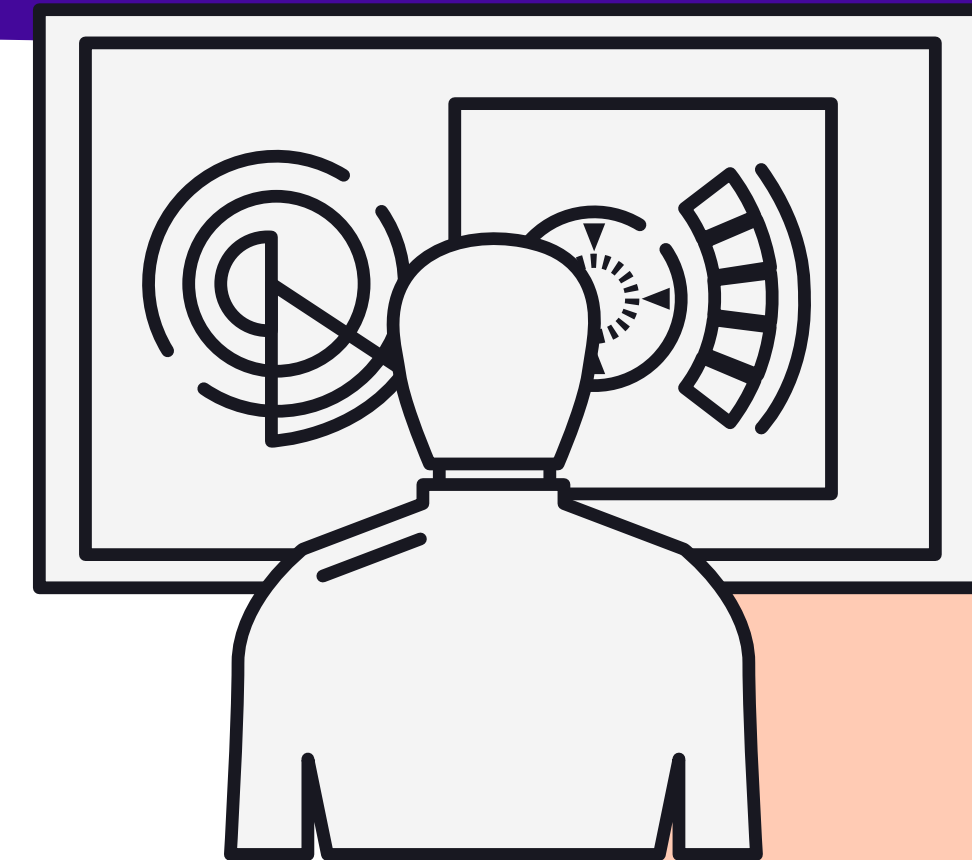
2.5 LISTE FUNCIONES DE AGREGACIÓN QUE CONOZCA.

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

- ▶ **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 MENCIONE ALGUNAS FUNCIONES PROPIAS DE SQL-SERVER.



- **Datetime()**. Para obtener la fecha, utiliza parámetros como @Date part o @Expression.
- **SUM()**. Suma de distintos valores.
- **MID()**. Extrae valores de un campo tipo texto.
- **AVG()**. Proporciona la media de una serie de valores.

2.7 PARA QUÉ SIRVE LA FUNCION CONCAT EN SQL-SERVER

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 the following SQL code:

```
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 window, the 'Results' tab is active, displaying the output of the query:

	NombreCompleto
1	Pedro Martín López
2	Ramón Pérez



2.8 MUESTRA UN EJEMPLO DEL USO DE COUNT

```
SELECT COUNT(*) FROM jugador where id_equipo='equ-222';
```

100 %

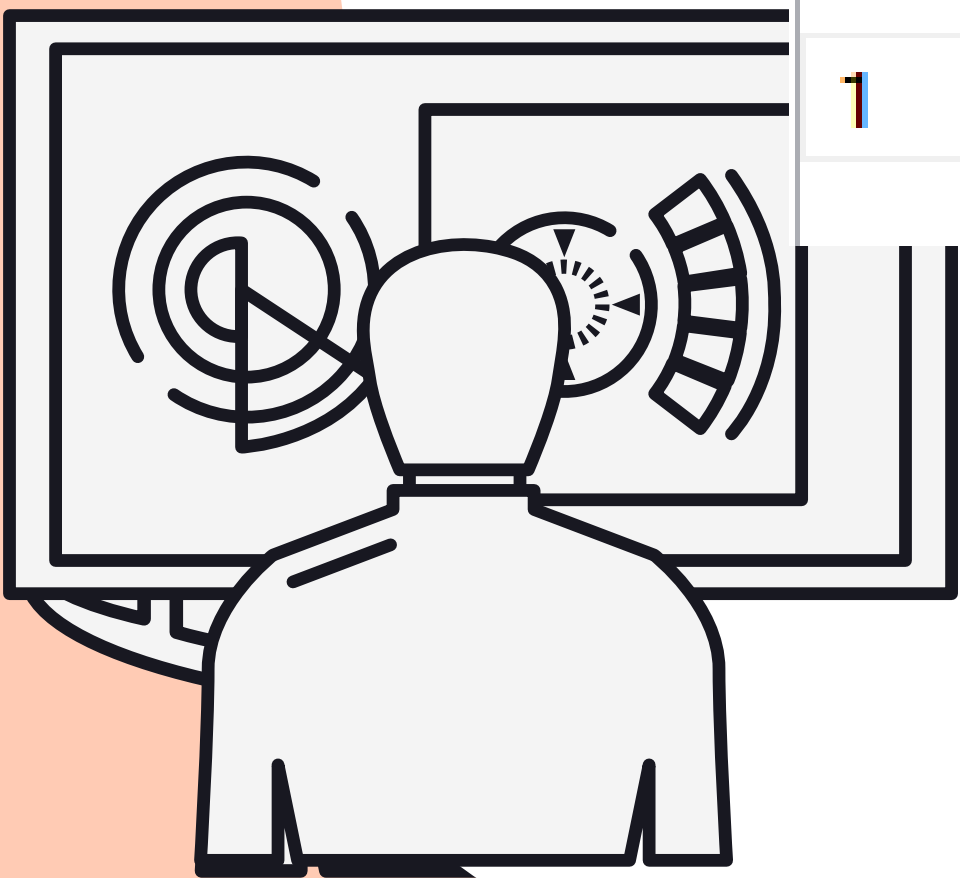


Results



Messages

	(No column na...
1	3



2.9 MUESTRA UN EJEMPLO DEL USO DE AVG

```
SELECT AVG(edad) FROM jugador;
```

100 %

Results Messages

	(No column na...
1	20



2.10 MUESTRA UN EJEMPLO DEL USO DE MIN-MAX

```
SELECT MAX(edad) FROM jugador;
```

100 %



Results



Messages

(No column na...

1

23



3. MANEJO DE CONSULTAS

Mostrar que jugadores que formen parte del equipo equ-333

```
SELECT * FROM jugador where id_equipo='equ-333';
```

100 %



Results



Messages

	id_juga...	nombres	apellid...	ci	ed...	id_equi...
1	jug-444	SANDRA	SOLIS	8997844LP	20	equ-333
2	jug-555	ANA	MICA	8997855LP	23	equ-333



3. MANEJO DE CONSULTAS

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(30))  
returns integer as  
begin  
    declare @Cantidad_Jugadores int;  
    select @Cantidad_Jugadores = Count(jug.nombres)  
    from jugador as jug  
    inner join equipo as equ on jug.id_equipo = equ.id_equipo  
    where equ.categoria = @categoria  
    return @Cantidad_Jugadores  
end;  
select dbo.F2_CantidadJugadoresParam(@categoria:'VARONES') as jugadores_inscritos
```



3. MANEJO DE CONSULTAS

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

```
create function F4_ConcatItems(  
@p1 varchar(100), @p2 varchar(100), @p3 varchar(100)  
)  
returns varchar(100) as  
begin  
declare @respuesta varchar(100)  
declare @concat1 varchar(100)  
declare @concat2 varchar(100)  
declare @concat3 varchar(100)  
set @concat1 = CONCAT('Nombre de los jugadores: ',@p1);  
set @concat2 = CONCAT('Nombre del equipo: ',@p2);  
set @concat3 = CONCAT('Sede: ',@p3);  
set @respuesta = @concat1 + @concat1 + @concat3  
  
return @respuesta;  
  
end;
```



3. MANEJO DE CONSULTAS

Generar la serie fibonacci.

```
alter function funcion_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(@n2),(@n1)
    while(@i<=@max-2)
    begin
        insert into @numbers values(@n2+@n1)
        set @temp = @n2
        set @n2 = @n2 + @n1
        set @i += 1
    end
    return
end
```

```
select * from dbo.funcion_fibonacci(@max: 5)
```

```
SELECT * FROM dbo.funcion_fibonacci(5);
```

% < Results Messages

number
0
1
1
2
3
5
8
13
21

GRACIAS

