

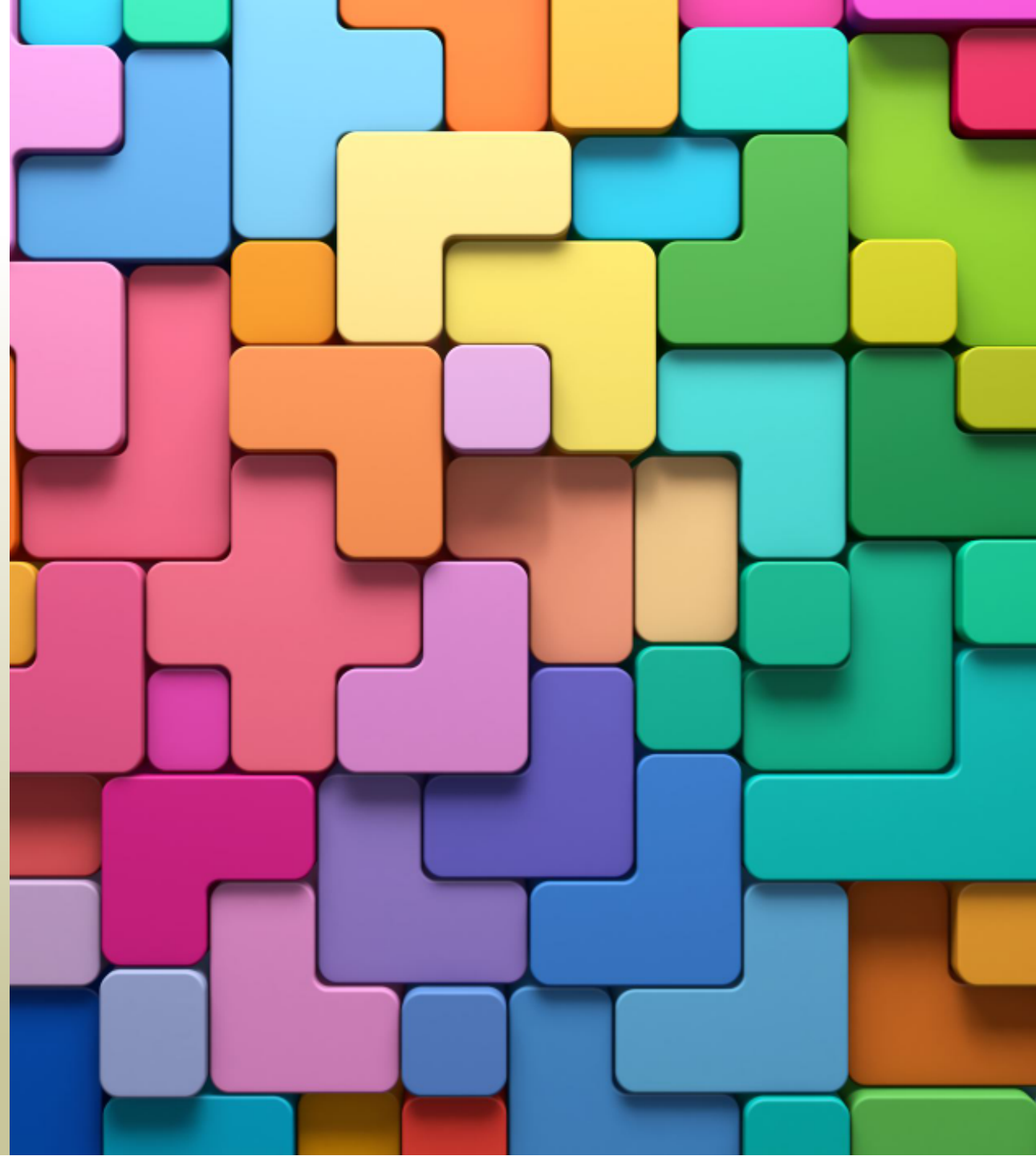
EVALUACIÓN PROCESUAL

HITO-4

BASE DE DATOS I

Estudiante: Jhanpol Machaca Aranibar

Cod_Sis: SIS9116774



MANEJO DE CONCEPTOS

1.-MUESTRA UN EJEMPLO DE DDL

```
create table campeonato
(
    id_campeonato varchar (10) primary key,
    nombre_campeonato varchar (20),
    sede varchar (25)
);

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

select * from campeonato;
```

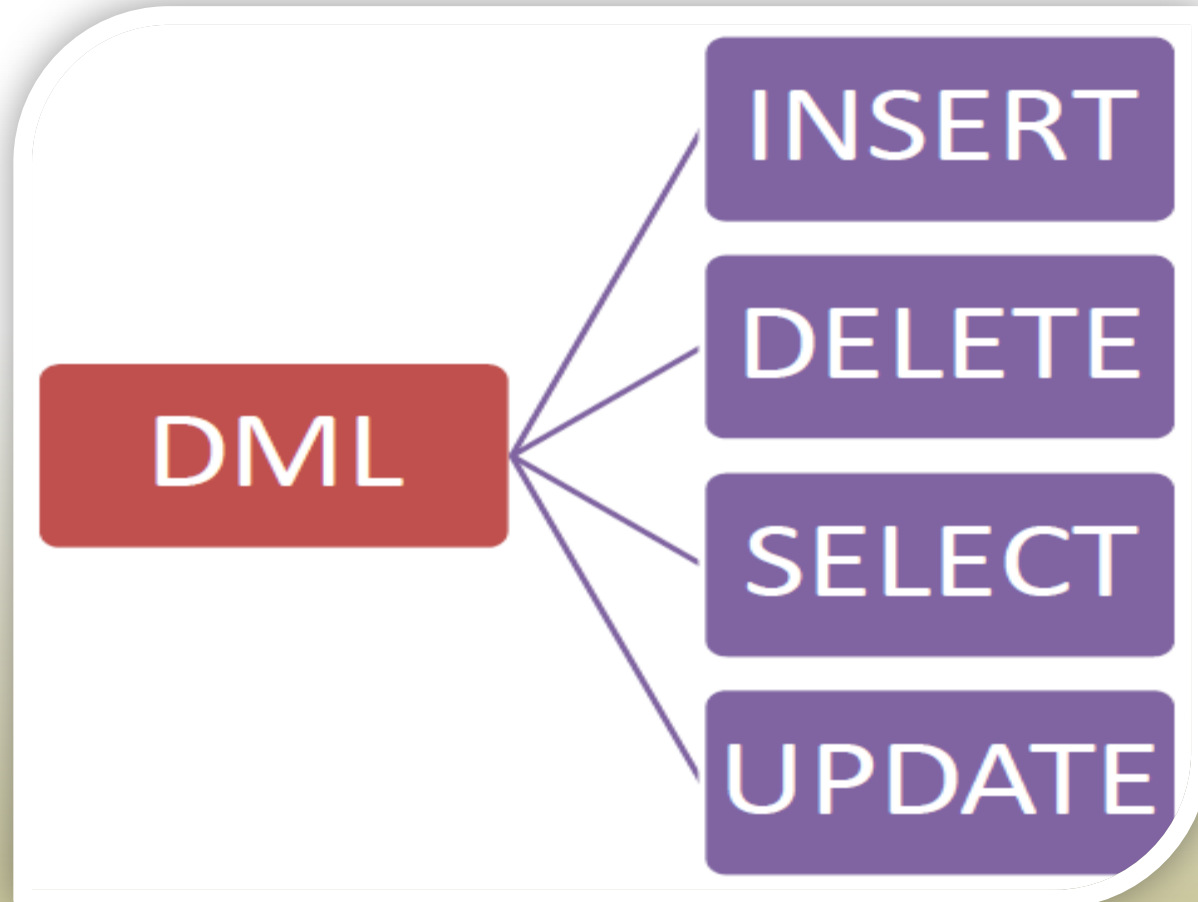
120 %

Results

Messages

	id_jugador	nombres	apellidos	CI	edad	id_equipo
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

2.-MUESTRA UN EJEMPLO DE DML



3.-PARA QUE SIRVE EL INNER JOIN

El inner join se utiliza para juntar las tablas algo así como una relación, para poder hacer esto las tablas deben tener mínimo un dato en común para que este funcione.

4.-DEFINA QUE ES UNA FUNCIÓN DE AGREGACIÓN

Las funciones nos ayudan a realizar operaciones para conseguir distintos tipos de resultados dependiendo de lo que necesitamos.

5.-LISTE FUNCIONES DE AGREGACIÓN

- **Count:** Este cuenta el numero de datos.
- **Min:** Nos dará el valor menor de una cosa en especifico.
- • **Max:** Nos dará el valor máximo de una cosa en especifico.

6.-MENCIONE ALGUNAS FUNCIONES PROPIAS DE SQL-SERVER

- **Count:** Este cuenta el numero de datos.
- **Min:** Nos dará el valor menor de una cosa en especifico.
- **Max:** Nos dará el valor máximo de una cosa en especifico.
- **Sum:** Suma los valores de los campos específicos.
- **AVG:** Nos devuelve el valor promedio de los campos (funciona sobre todo en divisiones).

7.-PARA QUE SIRVE LA FUNCIÓN CONCAT EN SQL-SERVER

Esta función lo que hace es tomar números de argumentos por cadena y los combina y concatena en una sola cadena, este necesita mínimo 2 datos de entrada.

8.-MUESTRA UN EJEMPLO DE COUNT

--mostrar un ejemplo de count--

```
SELECT COUNT (*)  
FROM jugador AS est;
```

145 %



Results



Messages

	(No column name)
1	5

9.-MUESTRA UN EJEMPLO DE AVG

```
--mostrar un ejemplo de AVG
SELECT avg(jug.edad)
FROM jugador AS jug;
```

145 %

Results Messages

	(No column name)
1	20

10.-MUESTRA UN EJEMPLO DE MIN-MAX

```
SELECT MAX(jug.edad)
FROM jugador AS jug
WHERE jug.id_equipo = 'equ-222';
```

```
SELECT MIN(jug.edad)
FROM jugador AS jug
WHERE jug.id_equipo = 'equ-222';
```

145 %

Results Messages

(No column name)	
1	21

(No column name)	
1	19

MANEJO DE CONSULTAS

1.-MOSTRAR QUE JUGADORES FORMEN PARTEN DEL EQUIPO EQU-333

```
Select nombres, id_equipo  
from jugador  
Where id_equipo = 'equ-222';
```

145 %

Results Messages

	nombres	id_equipo
1	carlos	equ-222
2	pedro	equ-222
3	saul	equ-222

2.-CREAR UNA FUNCIÓN QUE PERMITA SABER CUANTOS JUGADORES ESTÁN INSCRITOS

```
/*--2.-Crear una función que permita saber cuantos jugadores están inscritos--  
La función debe llamarse F1_CantidadJugadores()()*/
```

```
create function F1_CantidadJugadores(@id_jugador varchar(20))  
Returns Integer as  
Begin  
  
    Declare @response integer = 0;  
  
    SELECT @response COUNT(jug.id_jugador)  
    FROM jugador AS jug  
  
    end  
  
Return @response;  
  
end
```

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

```
/*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_CantidadJugadores(@id_jugador varchar(20))
Returns Integer as
Begin

    Declare @response integer = 0;

    SELECT @response COUNT(jug.id_jugador)
    FROM jugador AS jug
    inner join equipo as aqui on jug.id_jugador = aqui_jugador
    where equi.categotria = 'femenino' and jug.jugador

    end

Return @response;

end
```


4.- CREAR UNA FUNCIÓN QUE OBTENGA EL PROMEDIO DE LAS EDADES MAYORES A UNA CIERTA EDAD

- 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(@edad integer)
Returns Integer as
Begin

    Declare @response integer = 0;

    SELECT @response COUNT(jug.edad)
    FROM jugador AS jug
    WHERE jug.edad > @edad and equi.categotria = 'masculino';

    end

Return @response;

end
```

5.-CREAR UNA FUNCIÓN QUE PERMITA CONCATENAR 3 PARÁMETROS

```
/*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(@nombre varchar(20), nombre_equipo varchar (15), sede varchar (15))
Returns Integer as
Begin
    declare @respuesta varchar (60);

    select @respuesta = COUNT (jug.id_campeonato)
    from jugador as jug
    inner join campeonato as camp on jug.id_campeonato = camp.id_campeonato
    where jug.nombre and
           jug.nombre_equipo = '404 not found' and
           jug.sede 'El Alto';

    return @respuesta;

end;
```

GRACIAS POR SU ATENCIÓN
:D
