

## Bases de Datos - SQL Server

Procesual Hito 4

Base de Datos I - 2022

Nombre: Joel Reynaldo Condori Tumiri

Diseñe un sistema de Base de Datos Relacional utilizando el gestor de Base de Datos **SQL Server** teniendo como premisa el uso de buenas prácticas en diseño de la base de datos aplicados al siguiente escenario.

Detalle del problema

Problema

### UNIFRANZITOS

Se tiene como contexto un **CAMPEONATO DE FÚTBOL** en el cual se tiene 3 entidades principales el **campeonato** como tal, los **equipos** que participaran en el campeonato y en donde cada equipo tendrá una cantidad de **jugadores**.

En tal sentido se deberá crear las siguientes tablas.

- campeonato
- equipo
- jugador

Detalle de las tablas.

Create database Unifranzitos
use Unifranzitos

#### campeonato

id\_campeonato => cadena de 12 caracteres y ademas llave primaria
nombre\_campeonato => una cadena de 30 caracteres que no acepta valores nulos
sede => una cadena de 20 caracteres que no acepta valores nulos

#### equipo

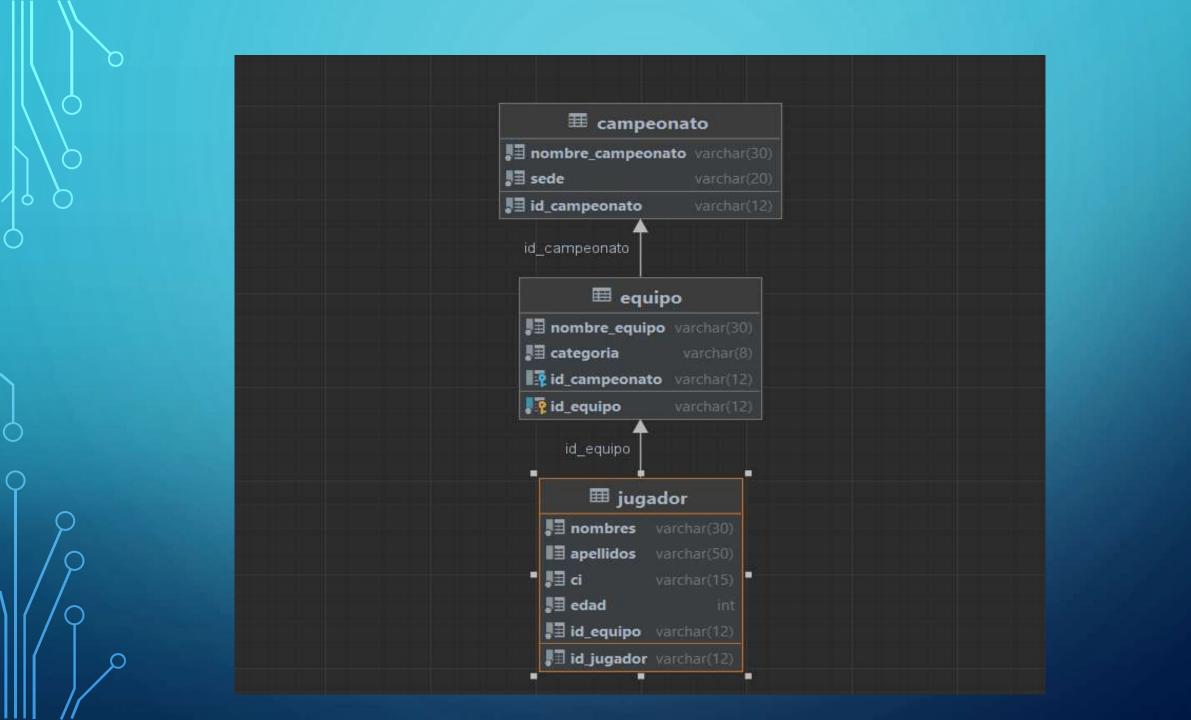
id\_equipo => cadena de 12 caracteres y ademas llave primaria
nombre\_equipo => una cadena de 30 caracteres, que no acepta valores nulos
categoria => esta columna recibe valores como (varones o mujeres), que no acepta
valores nulos
id campeonato => llave foreign key relacionado con la tabla campeonato

### jugador

id\_jugador => cadena de 12 caracteres y ademas llave primaria
nombres => una cadena de 30 caracteres, que no acepta valores nulos
apellidos => una cadena de 50 caracteres, que no acepta valores nulos
ci => una cadena de 15 caracteres (ejem: 8997899LP), que no acepta valores nulos
edad => un valor numérico, que no acepta valores nulos
id\_equipo => llave foreign key relacionado con la tabla equipo

```
create table campeonato
    id_campeonato varchar (12) primary key,
    nombre_campeonato varchar (30) not null,
    sede varchar (20) not null,
create table equipo
    id_equipo varchar (12) primary key,
    nombre_equipo varchar (30) not null,
    categoria varchar (8) not null,
    id_campeonato varchar (12),
    Foreign key (id_campeonato) references campeonato (id_campeonato)
create table jugador
    id_jugador varchar (12) primary key,
    nombres varchar (30) not null,
    apellidos varchar (50) not null,
    ci varchar (15) not null,
    edad int not null,
```

# DE FOREIGN KEY Y PRIMARY KEY



# tabla campeonato in id\_campeonato camp-111 camp-222 Campeonato Unifranz Campeonato Unifranz Cochabamba

tabla equipo			
<pre>id_equipo</pre>	: 💷 nombre_equipo	: 💷 categoria	<ul> <li>Imid_campeonato</li> </ul>
equ-111	Google	VARONES	camp-111
equ-222	404 Not found	VARONES	camp-111
equ-333	girls unifranz	MUJERES	camp-111

tabla jugador					
<pre>id_jugador</pre>	* # nombres	: ⊯ apellidos	## ci	# edad :	mid_equipo
jug-111	Carlos	Villa	8997811LP	19	equ-222
jug-222	Pedro	Salas	8997822LP	20	equ-222
jug-333	Saul	Araj	8997833LP	21	equ-222
jug-444	Sandra	Solis	8997844LP	20	equ-333
jug-555	Ana	Mica	8997855LP	23	equ-333

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

insert into equipo values ('equ-111', 'Google', 'VARONES', 'camp-111')
insert into equipo values ('equ-222', '404 Not found', 'VARONES', 'camp-111')
insert into equipo values ('equ-333', 'girls unifranz', 'MUJERES', 'camp-111')

insert into jugador values ('jug-111', 'Carlos', 'Villa', '8997811LP', 19, 'equ-222')
insert into jugador values ('jug-222', 'Pedro', 'Salas', '8997822LP', 20, 'equ-222')
insert into jugador values ('jug-333', 'Saul', 'Araj', '8997833LP', 21, 'equ-222')
insert into jugador values ('jug-444', 'Sandra', 'Solis', '8997844LP', 20, 'equ-333')
insert into jugador values ('jug-555', 'Ana', 'Mica', '8997855LP', 23, 'equ-333')
```

### MANEJO DE CONCEPTOS

### MUESTRA UN EJEMPLO DE DLL

### MUESTRA UN EJEMPLO DE DML

```
--muestra un ejemplo de DML
insert into equipo values ('equ-111','google','varones','camp-111'),
delete from jugador where nombres = 'carlos'
```

¿PARA QUE SIRVE UN INNER JOIN? R.- COMBINA LOS REGISTROS DE DOS TABLAS SI HAY VALORES COINCIDENTES EN UN CAMPO COMÚN ¿DEFINA QUE ES UNA FUNCIÓN DE AGREGACIÓN? R.- NOS PERMITEN OBTENER EL MÁXIMO VALOR DE UNA EXPRESIÓN A EVALUAR. ¿LISTA FUNCIONES DE AGREGACIÓN QUE CONOZCA? R.-AVG, COUNT, MAX, MIN, SUM ¿MENCIONE ALGUNAS FUNCIONES PROPIAS DE SQL-SERVER.? R.- AVG, COUNT, MAX, MIN, SUM, GROUP BY ¿PARA QUE SIRVE LA FUNCIÓN CONCAT EN SQL-SERVER? PARA CONCATENAR UNA DE LAS FUNCIONES DE TEXTO PARA U OS O MAS CADENAS DE TEXTO EN UNA SOLA

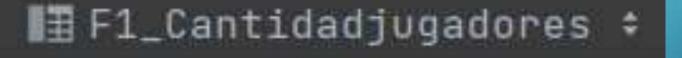
```
--muestra un ejemplo del uso de count
         select count(jug.nombres)
         from jugador as jug
64
         --muestra un ejemplo del uso de AVG
        select AVG(jug.edad)
        from jugador as jug
       --muestra un ejemplo del uso de min
       select min (jug.edad)
       from jugador as jug
       inner join equipo as equi on jug.id_equipo = equi.id_equipo
       where equi.categoria = 'varones'
      --muestra un ejemplo del uso de max
73
      select max (jug.edad)
      from jugador as jug
      inner join equipo as equi on jug.id_equipo = equi.id_equipo
      where equi.categoria = 'mujeres'
```

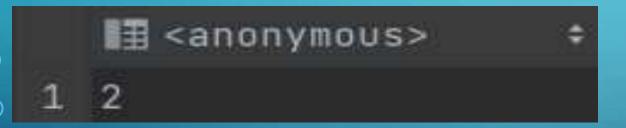
## MANEJO DE CONSULTAS

```
--mostrar que jugadores que formen parte del equipo equ-333

| select jug.nombres, jug.apellidos, equi.id_equipo, equi.nombre_equipo
| from equipo as equi
| inner join jugador as jug on equi.id_equipo = jug.id_equipo
| where equi.id_equipo = 'equ-333'
```

Г	I⊞ nombres	÷	<b>I</b> ∄ apellidos	Ŷ.	<b>Ⅲ</b> id_equipo	÷	■ nombre_equipo	*
1	Sandra		Solis		equ-333		girls unifranz	
2	Ana		Mica		equ-333		girls unifranz	



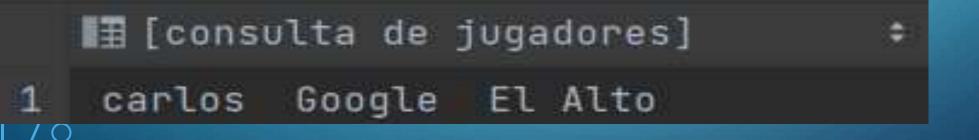


```
-- La categoría. (Varones o Mujeres)
-- La edad con la que se comparara (21 años ejemplo)
-- Es decir mostrar el promedio de edades que sean de una categoría y que sean mayores a 21 años.
alter function F3_PromedioEdades(@categoria varchar (8))
returns varchar (20) as
    begin
        declare @edades int
        select @edades = avg(jug.edad)
        from jugador as jug
        inner join equipo as equi on jug.id_equipo = equi.id_equipo
        where equi.categoria = @categoria and jug.edad > 21
        return @edades
select dbo.F3_PromedioEdades('mujeres')
```

## ∎≣ <anonymous>

1 23

```
-- Crear una función que permita concatenar 3 parámetros
-- La función debe de recibir 3 parámetros.
create function F4_ConcotItems(@parametro1 varchar (30), @parametro2 varchar (30), @parametro3 varchar (30))--paramtros valores que nosotro ingresamos
returns varchar (30) as
    declare @concatenado varchar (30)
    set @concatenado = concat(@parametro1,@parametro2,@parametro3)
    return @concatenado
    end
select dbo.F4_ConcatItems(' carlos ', ' Google ', ' El Alto ' ) as 'consulta de jugadores'
|select dbo.F4_ConcatItems( jug.nombres, equi.nombre_equipo, cam.sede )
from jugador as jug
inner join equipo as equi on jug.id_equipo = equi.id_equipo
inner join campeonato as cam on equi.id_campeonato = cam.id_campeonato
```



```
🗦--concatenado 2.0
 alter function F4_ConcatItems2(@par1 varchar (80), @par2 varchar (80), @par3 varchar (80))
 returns varchar (80) as
 begin
     declare @concatenado1 varchar (80)
    declare @concatenado2 varchar (80)
     declare @concatenado3 varchar (80)
     declare @respuesta varchar (80)
     set @concatenado1 = concat(' nombre de jugadores: ',@par1)
     set @concatenado2 = concat(' nombres de equipo: ',@par2)
     set @concatenado3 = concat(' sede: ',@par3)
     set @respuesta = @concatenado1+@concatenado2+@concatenado3
     return @respuesta
 select dbo.F4_ConcatItems2('marco','dbo','santa cruz')
 select dbo.F4_ConcatItems2( jug.nombres, equi.nombre_equipo, cam.sede )
 from jugador as jug
 inner join equipo as equi on jug.id_equipo = equi.id_equipo
 inner join campeonato as cam on equi.id_campeonato = cam.id_campeonato
```