

Hito 4 – Ingeniería de sistemas



# Manejo de consultas y funciones

# 1. Diseño de la base de datos

campeonato
<b>id_campeonato</b> => cadena de 12 caracteres y ademas llave primaria
<b>nombre_campeonato</b> => una cadena de 30 caracteres que no acepta valores nulos
<b>sede</b> => una cadena de 20 caracteres que no acepta valores nulos

  

equipo
<b>id_equipo</b> => cadena de 12 caracteres y ademas llave primaria
<b>nombre_equipo</b> => una cadena de 30 caracteres, que no acepta valores nulos
<b>categoria</b> => esta columna recibe valores como (varones o mujeres), que no acepta valores nulos
<b>id_campeonato</b> => llave foreign key relacionado con la tabla campeonato

  

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



**tabla campeonato**

🔑 id_campeonato	📄 nombre_campeonato	📄 sede
camp-111	Campeonato Unifranz	El Alto
camp-222	Campeonato Unifranz	Cochabamba

**tabla equipo**

🔑 id_equipo	📄 nombre_equipo	📄 categoria	🔑 id_campeonato
equ-111	Google	VARONES	camp-111
equ-222	404 Not found	VARONES	camp-111
equ-333	girls unifranz	MUJERES	camp-111

**tabla jugador**

🔑 id_jugador	📄 nombres	📄 apellidos	📄 ci	📄 edad	🔑 id_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

## 2. Manejo de conceptos

2.1. Muestra un ejemplo de DDL.

```
CREATE TABLE BDA1(nombre varchar(50),  
aula varchar(20),  
gestion int);
```

```
INSERT INTO BDA1(nombre,aula,gestion)VALUES  
( 'Juan Carlos Roque', 'BDA I-218',2022),  
( 'Pedro Pascal Mamani', 'BDA I-218',2021);  
  
SELECT *  
FROM BDA1;
```

2.2. Muestra un ejemplo de DML.

### 2.3. Para que sirve INNER JOIN

Inner join combina los registros de dos tablas siempre y cuando haya valores que coincidan en un campo en comun en ambas tablas.

```
SELECT jug.nombres,eq.nombre_equipo
FROM jugador as jug
inner join equipo as eq on eq.id_equipo=jug.id_equipo
WHERE eq.id_equipo='equ-222';
```

equipo	
id_equipo	varchar(12)
nombre_equipo	varchar(30)
categoria	varchar(8)
id_campeonato	varchar(12)

id\_equipo

jugador	
id_jugador	varchar(12)
nombres	varchar(30)
apellidos	varchar(50)
ci	varchar(15)
edad	int
id_equipo	varchar(12)

### 2.4. Defina que es una función 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. Es decir, nos permiten obtener medias, máximos, etc... sobre un conjunto de valores.

```
SELECT COUNT (*) AS numero_de_estudiantes
FROM estudiantes AS est;
```

	numero_de_estudiantes
1	7

2.5. Liste funciones de agregación que conozca

```
SELECT COUNT (*) AS numero_de_estudiantes  
FROM estudiantes AS est;  
  
SELECT MIN (est.edad) edad_minima  
FROM estudiantes AS est;  
  
SELECT MAX (est.edad) edad_maxima  
FROM estudiantes AS est;  
  
SELECT AVG (est.edad) prom_edad  
FROM estudiantes AS est;  
  
SELECT SUM (est.edad) suma_edad  
FROM estudiantes AS est;
```

2.6. Mencione algunas funciones propias de SQL-Server

```
SELECT STDEV (est.edad) desviacion  
FROM estudiantes AS est;  
  
SELECT VAR (est.edad)  
FROM estudiantes AS est;
```

	numero_de_estudiantes
1	7
	edad_minima
1	20
	edad_maxima
1	30
	prom_edad
1	22
	suma_edad
1	160
	desviacion
1	3.76069902316805
	varianza
1	14.1428571428571



2.7. Para qué sirve la función CONCAT en SQL-Server

```
SELECT CONCAT('Hola', ' ', 'compañeros de BASE DE DATOS I');
```

	(No column name)
1	Hola compañeros de BASE DE DATOS I

2.8. Muestra un ejemplo del uso de COUNT

```
SELECT COUNT(est.id_est) estudiantes_escuela  
FROM estudiantes as est  
INNER join escuela as esc ON esc.id_esc=est.id_esc  
WHERE esc.nombre='Amor de Dios Fe y Alegria';
```

	estudiantes_escuela
1	2

2.9. Muestra un ejemplo  
del usos de AVG

```
SELECT AVG (jug.edad) prom_edad  
FROM jugador as jug;
```

	prom_edad
1	20

2.10. Muestra un ejemplo  
del uso de MIN-MAX

```
SELECT MIN (jug.edad) edad_minima  
FROM jugador as jug;
```

```
SELECT MAX (jug.edad) edad_maxima  
FROM jugador as jug;
```

	edad_minima
1	19

  

	edad_maxima
1	23



### 3. 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.

- La función debe llamarse Crear una función que permita saber cuántos jugadores están inscritos.
- La función debe llamarse **F1\_CantidadJugadores()**

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”

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.

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.

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

```
SELECT [dbo].fibonacci2(5);
```

Output x [dbo].fibonacci2(5):varchar(100) x

|< < 1 row v > >| | ↺ ■ ↗

<anonymous>

1 0, 1, 1, 2, 3,