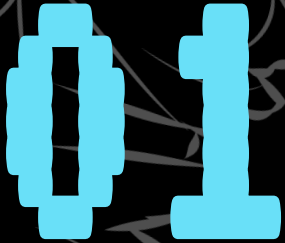




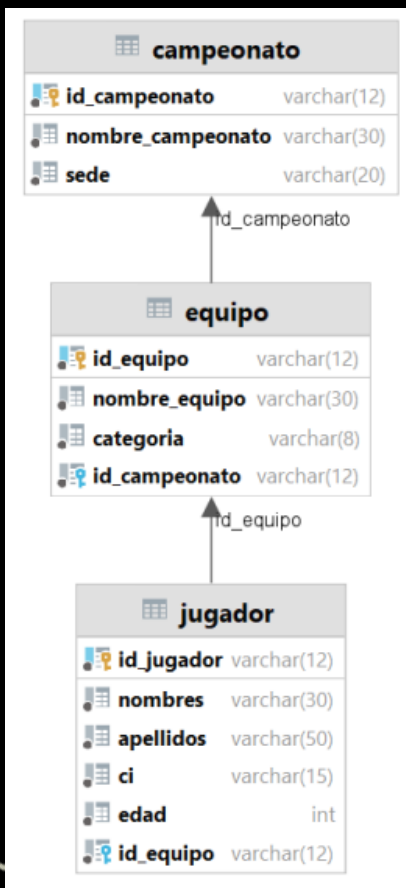
BASE DE DATOS

FREDDY MACHACA MAMANI



01

DISEÑO DE BASE DE DATOS



| 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 |

```
Create database Unifranzitos
use Unifranzitos
```

```
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,

  id_equipo varchar (12),
  Foreign Key (id_equipo) references equipo (id_equipo)
```

```
)
```

```
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')
```



02

MANEJO DE CONCEPTOS

2.1. Muestra un ejemplo de DDL.

DDL

Esta formado por un conjunto de sentencias llamadas ddl que nos sirve para la creación de una base de datos y todos sus componentes

-CREATE

```
Create database Unifranzitos  
use Unifranzitos
```

Crea base de datos y nos permite crear tablas

-DROP

```
drop table jugador
```

```
drop database Unifranzitos
```

Drop table borra la tabla

-ALTER

```
alter table jugador  
add  
seleccion varchar (12)
```

Modifica la estructura de una tabla

-TRUNCATE

```
truncate table jugador
```

Truncate vacía la tabla

2.2. Muestra un ejemplo de DML

DML

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

-INSERT

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

Insertar registros a una tabla

-UPDATE

```
update campeonato set sede = 'El alto'  
where id_campeonato = 'camp-111'
```

Modificación de la información de una tabla

-DELETE

```
delete from jugador  
where id_jugador = 'jug-333'
```

Eliminar registros de una tabla

2.3. Para que sirve INNER JOIN.

Combina los registros de dos tablas si hay valores coincidentes en un campo común. De tal modo que sólo la intersección se mostrará en los resultados.

```
select camp.nombre_campeonato, jug.id_jugador, jug.nombres, jug.apellidos
from jugador as jug
inner join equipo as equ on jug.id_equipo = equ.id_equipo
inner join campeonato as camp on equ.id_campeonato=camp.id_campeonato
where jug.id_jugador='jug-333'
```

2.4. Defina que es una función de **agregación**.

2.5. Liste funciones de **agregación** que conozca.

Las funciones de agregación operan en un conjunto de valores para devolver un valor único. Por ejemplo, pueden proporcionar el valor de un indicador clave de rendimiento o una medida basada en un conjunto de métricas.

Las funciones de agregación que empleo y conozco son sum, max, min y avg.

2.6. Mencione algunas funciones propias de SQL-Server.

AVG

```
select avg(jug.edad)  
from jugador AS JUG
```

La media de valores.

COUNT

```
select COUNT(jug.id_jugador)  
from jugador AS JUG
```

El numero de filas.

MAX

```
select max(jug.edad)  
from jugador AS JUG
```

El valor mas grande.

MIN

```
select min(jug.edad)  
from jugador AS JUG
```

El valor mas pequeño.

SUM

```
select sum(jug.edad)  
from jugador AS JUG
```

La suma de los valores

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

La función CONCAT() agrega dos o más cadenas juntas.

SINTAXIS

CONCAT(string1, string2,, string_n)

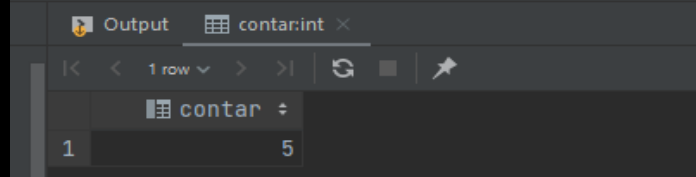
```
create function F4_ConcatItems(@nombres varchar(max), @nombre_equipo varchar(max), @sede varchar(max))
returns varchar(max) as
begin
    declare @return varchar(max)
    declare @conc1 varchar(max)
    declare @conc2 varchar(max)
    declare @conc3 varchar(max)
    set @conc1 = concat('Nombres jugadores: ',@nombres)
    set @conc2 = concat('Nombres Equipos: ',@nombre_equipo)
    set @conc3 = concat('Sede: ',@sede)
    set @return=@conc1+@conc2+@conc3
    return @return
end

select dbo.F4_ConcatItems( @nombres: 'Augusto Salazar ', @nombre_equipo: 'Sao Paulo ', @sede: 'La paz')
```

- 2.8. Muestra un ejemplo del uso de **COUNT**
- 2.9. Muestra un ejemplo del usos de **AVG**

COUNT

```
-- determinar cuantos usuarios registrados tengo
select count (us.ID_USUARIOS) as contar
from USUARIOS us
```



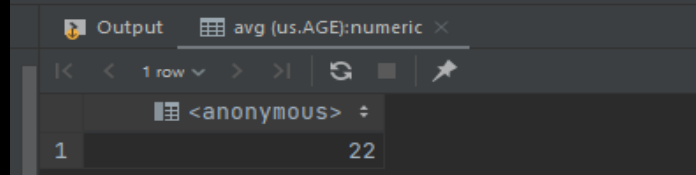
The screenshot shows the output of the SQL query. The 'Output' tab is active, displaying a table with one row. The column is named 'contar' and the value is 5.

| contar |
|--------|
| 5 |

La cantidad de usuarios es 5.

AVG

```
-- determinar cual es el promedio de las edades
select avg (us.AGE)
from USUARIOS as us
```



The screenshot shows the output of the SQL query. The 'Output' tab is active, displaying a table with one row. The column is named '<anonymous>' and the value is 22.

| <anonymous> |
|-------------|
| 22 |

El promedio es 22.

2.10. Muestra un ejemplo del uso de MIN-MAX

MAX

```
-- determinar cual es la edad maxima que se tiene en los resultados
select max (us.AGE)
from USUARIOS as us
```

Output max (us.AGE):numeric x

| | |
|---|----|
| 1 | 32 |
|---|----|

La edad máxima es 32.

MIN

```
-- determinar cual es la edad minima que se tiene en los resultados
select min (us.AGE)
from USUARIOS as us
```

Output min (us.AGE):numeric x

| | |
|---|----|
| 1 | 15 |
|---|----|

La edad mínima es 15

The background is black with several white lines of varying lengths and orientations. A small cyan square is located near the top center. On the left side, there is a large, stylized number '03' in a light blue, pixelated font. The number '0' has a vertical line through its center, and the '3' is composed of several horizontal and vertical segments.

03

MANEJO DE CONSULTAS

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

```
--3.1. Mostrar que jugadores que formen parte del equipo equ-333
select jug.nombres, jug.apellidos, equ.id_equipo, equ.nombre_equipo
from jugador as jug
inner join equipo as equ on jug.id_equipo = equ.id_equipo
where equ.id_equipo = 'equ-333'
```

Output 3.1. Mostrar que jug...te del equipo equ-333

| | nombres | apellidos | id_equipo | nombre_equipo |
|---|---------|-----------|-----------|----------------|
| 1 | Sandra | Solis | equ-333 | girls unifranz |
| 2 | Ana | Mica | equ-333 | girls unifranz |

- 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()**

```
--3.2. Crear una función que permita saber cuántos jugadores están inscritos.  
create function F1_CantidadJugadores()  
returns int as  
begin  
    declare @Cant_Jug int  
    SELECT @Cant_Jug = count (jug.id_jugador)  
    from jugador as jug  
    return @Cant_Jug  
end  
  
select dbo.F1_CantidadJugadores()
```

| Output | |
|--------------------------------|---|
| dbo.F1_CantidadJugadores():int | |
| 1 row | |
| <anonymous> | |
| 1 | 5 |

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"

```
--3.3. 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(8))  
returns varchar as
```

```
begin
```

```
    declare @Cant_Jug varchar
```

```
    select @Cant_Jug = count(jug.id_jugador)
```

```
    from jugador as jug
```

```
    inner join equipo as equ 1..n<->1: on jug.id_equipo = equ.id_equipo
```

```
    where equ.categoria =@categoria
```

```
    return @Cant_Jug
```

```
end
```

```
select dbo. F2_CantidadJugadoresParam( @categoria: 'VARONES')
```

Output dbo. F2_CantidadJuga... 'VARONES');varchar(1) ×

1 row ↺ ↻ ⚡

<anonymous> ⌵

1 3

- 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.

```
--3.4. 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(8), @edad int)  
RETURNS int AS  
BEGIN  
    DECLARE @Prom_edad int  
    SELECT @Prom_edad = avg (jug.edad)  
    FROM jugador as jug  
    INNER JOIN equipo AS equ 1..n<->1: on jug.id_equipo = equ.id_equipo  
    where equ.categoria = @categoria and jug.edad > @edad  
    RETURN @Prom_edad  
end  
  
select dbo.F3_PromedioEdades( @categoria: 'varones', @edad: 19)
```

Output | dbo.F3_PromedioEdades('varones',19):int

|< < 1 row > >|

<anonymous>

| | |
|---|----|
| 1 | 20 |
|---|----|

- 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(@nombres varchar(max), @nombre_equipo varchar(max), @sede varchar(max))
returns varchar(max) as
begin
    declare @return varchar(max)
    declare @conc1 varchar(max)
    declare @conc2 varchar(max)
    declare @conc3 varchar(max)
    set @conc1 = concat('Nombres jugadores: ',@nombres)
    set @conc2 = concat('Nombres Equipos: ',@nombre_equipo)
    set @conc3 = concat('Sede: ',@sede)
    set @return=@conc1+@conc2+@conc3
    return @return
end

select dbo.F4_ConcatItems( @nombres: 'Augusto Salazar ', @nombre_equipo: 'Sao Paulo ', @sede: 'La paz')
```

Output

|< < 1 row > >| ↺ ■ ↻

<anonymous>

1 Nombres jugadores: Augusto Salazar Nombres Equipos: Sao Paulo Sede: La paz

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 fn_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(@n1),(@n2)
    WHILE (@i<=@max-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.fn_Fibonacci( @max: 5)
```

Output Result 7

6 rows

| | number |
|---|--------|
| 1 | 0 |
| 2 | 1 |
| 3 | 1 |
| 4 | 2 |
| 5 | 3 |