

## PROBLEMAS DE LA PRACTICA DEL HITO 4

Usando las tablas de los UNIFRANZITOS

```
create database Hito_4
go
```

```
use Trab_Hito_4_uwu
go
```

```
CREATE TABLE CAMPEONATO
(
ID_CAMPEONATO VARCHAR(12) PRIMARY KEY NOT NULL,
NOMBRE_DEL_CAMPEONATO VARCHAR (30) NOT NULL,
SEDE VARCHAR (30) NULL
);
```

```
CREATE TABLE EQUIPO
(
ID_EQUIPO VARCHAR(12) PRIMARY KEY NOT NULL,
ID_CAMPEONATO VARCHAR (12) NOT NULL,
NOMBRE_DE_EQUIPO VARCHAR (30) NOT NULL,
CATEGORIA VARCHAR (30) NULL,
FOREIGN KEY (ID_CAMPEONATO) REFERENCES CAMPEONATO (ID_CAMPEONATO )
);
```

```
CREATE TABLE JUGADOR
(
ID_JUGADOR VARCHAR (12) PRIMARY KEY NOT NULL,
NOMBRES VARCHAR (30),
APELLIDOS VARCHAR (30),
CI VARCHAR (30),
EDAD INTEGER,
ID_EQUIPO VARCHAR(12) NOT NULL,
FOREIGN KEY (ID_EQUIPO) REFERENCES EQUIPO(ID_EQUIPO)
);
```

--Inserts en la tabla campeonato

```
INSERT INTO CAMPEONATO(ID_CAMPEONATO,NOMBRE_DEL_CAMPEONATO,SEDE)
VALUES('CAMP-111','CAMPEONATO UNIFRANZ','EL ALTO')
INSERT INTO CAMPEONATO(ID_CAMPEONATO,NOMBRE_DEL_CAMPEONATO,SEDE)
VALUES('CAMP-222','CAMPEONATO UNIFRANZ','COCHABAMBA')
```

--Inserts en la tabla equipo

```
INSERT INTO EQUIPO(ID_EQUIPO,NOMBRE_DE_EQUIPO,CATEGORIA,ID_CAMPEONATO)
VALUES('EQUIP-111','GOOGLE','VARONES','CAMP-111')
INSERT INTO EQUIPO(ID_EQUIPO,NOMBRE_DE_EQUIPO,CATEGORIA,ID_CAMPEONATO)
VALUES('EQUIP-222','404 NOT FOUND ','VARONES','CAMP-111')
INSERT INTO EQUIPO(ID_EQUIPO,NOMBRE_DE_EQUIPO,CATEGORIA,ID_CAMPEONATO)
VALUES('EQUIP-333','GIRLS UNIFRANZ ','MUJERES','CAMP-111')
```

--Inserts en la tabla jugador

```

INSERT INTO JUGADOR(ID_JUGADOR,NOMBRES,APELLIDOS,CI,EDAD,ID_EQUIPO)
VALUES('JUG-111','CARLOS','VILLAS','8997811LP',19,'EQUIP-222')
INSERT INTO JUGADOR(ID_JUGADOR,NOMBRES,APELLIDOS,CI,EDAD,ID_EQUIPO)
VALUES('JUG-222','PEDRO','SALAS','8997822LP',20,'EQUIP-222')
INSERT INTO JUGADOR(ID_JUGADOR,NOMBRES,APELLIDOS,CI,EDAD,ID_EQUIPO)
VALUES('JUG-333','SAUL','ARAJ','8997833LP',21,'EQUIP-222')
INSERT INTO JUGADOR(ID_JUGADOR,NOMBRES,APELLIDOS,CI,EDAD,ID_EQUIPO)
VALUES('JUG-444','SANDRA','SOLIS','8997844LP',20,'EQUIP-333')
INSERT INTO JUGADOR(ID_JUGADOR,NOMBRES,APELLIDOS,CI,EDAD,ID_EQUIPO)
VALUES('JUG-555','ANA','MICA','8997855LP',23,'EQUIP-333')

```

```
SELECT * FROM dbo.CAMPEONATO
```

```
SELECT * FROM dbo.EQUIPO
```

```
SELECT * FROM dbo.JUGADOR
```

```

SELECT CONCAT (ju.NOMBRES, ' ', ju.APELLIDOS) as nombres_y_apellidos
FROM JUGADOR AS ju
inner JOIN EQUIPO AS eq
ON ju.ID_EQUIPO = eq.ID_EQUIPO
WHERE JU.EDAD > 20

```

```

SELECT CONCAT (ju.NOMBRES, ' ', ju.APELLIDOS, ' ', ju.EDAD) as nombres_y_apellidos_edad
FROM JUGADOR AS ju
inner JOIN EQUIPO AS eq
ON ju.ID_EQUIPO = eq.ID_EQUIPO
WHERE JU.EDAD > 20

```

```

select LOWER(eq.NOMBRE_DE_EQUIPO)
from EQUIPO as eq

```

--Muestra un ejemplo de DDL

```

alter database Hito_4
modify name = Trab_Hito_4_uwu

```

--Muestra un ejemplo de DML

```

select jug.NOMBRES, jug.EDAD
from JUGADOR as jug
where jug.NOMBRES like 'S%'

```

--Para que drive INNER JOIN

```

select cam.SEDE, cam.NOMBRE_DEL_CAMPEONATO, cam.ID_CAMPEONATO, eq.CATEGORIA
From CAMPEONATO AS cam
inner join EQUIPO AS eq
on eq.CATEGORIA like 'm%'

```

--DEFINE QUE ES UNA FUNCIÓN DE AGREGACIÓN

'Nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un unico valor agregado para todos ellos, nos permite obtener medias, maximos, etc...sobre un conjunto de valores'

--LISTE FUNCIONES DE AGREGACION QUE CONOZCA

```

--AVG()
--COUNT()
--MAX()
--MIN()
--SUM()
--STDDEV()
--VARIANCE()
--VAR()
--GROUPING()
--CHECKSUM_AGG()
--COUNT_BIG()

--MENCIONA ALGUNAS FUNCIONES PROPIAS DE SQL-SERVER

--AVG()
--COUNT()
--MAX()
--MIN()
--SUM()

--PARA QUE SIRVE LA FUNCION CONCAT

'Nos permite enlazar o vincular distintos tipos de valores
y llevarlos a un string de forma automatica donde ademas si
hay nullos no impactaran en el resultado'

--MUESTRA UN EJEMPLO DEL USO DE COUNT

select count(jug.NOMBRES)
FROM JUGADOR as jug
where jug.NOMBRES like 'S%'

--MUESTRA UN EJEMPLO DEL USO DE AVG

select avg(jug.EDAD)
FROM JUGADOR AS jug

--MUESTRA UN EJEMPLO DEL USO DE MIN-MAX

select MIN(jug.EDAD) EDAD_MINIMA
FROM JUGADOR AS jug
--MAXIMO--
select MAX(jug.EDAD) EDAD_MAXIMA
FROM JUGADOR AS jug

--SEGUNDA PARTE
--MANEJO DE CONSULTA

--MOSTRAR QUE JUGADORES FORMAN PARTE DEL EQUIPO equ-333

select count(jug.NOMBRES) JUGADORES
from JUGADOR AS jug
inner join EQUIPO AS eq
on jug.ID_EQUIPO = eq.ID_EQUIPO
where jug.ID_EQUIPO like 'equ-333%'

--crear una funcion que permita saber cuantos jugadores estan inscritos.

```

--la funcion debe llamarse Crear una funcion que permita saber cuantos jugadores estan inscritos.

--La funcion debe llamarse f1\_CantidadJugadores()()

```
create or alter function F1_CantidadJugadores()
returns integer as
begin
    declare @Cantidad_Jugadores integer;

    select @Cantidad_Jugadores = Count(jug.nombres)
    from jugador as jug
    return @Cantidad_Jugadores
end;
```

```
select dbo.F1_CantidadJugadores() as jugadores_inscritos
```

--Crear una funcion que permita saber cuantos jugadores estan inscritos y que sean de la categoria varones o mujeres.

```
create or alter function F2_CantidadJugadoresParam(@categoria varchar(30))
returns integer as
begin
    declare @Cantidad_Jugadores integer;

    select @Cantidad_Jugadores = Count(jug.nombres)
    from jugador as jug
    inner join equipo as equi on jug.id_equipo = equi.id_equipo
    where equi.categoria = 'Varones'
    return @Cantidad_Jugadores

end;
```

```
select dbo.F2_CantidadJugadoresParam('Varones') as Jugadores_Varones
```

--Crear una funcion que contenga el promedio de las edades mayores a una cierta edad.

```
create or alter function F3_PromedioEdades()
returns integer as
begin
    declare @Promedio_edades integer;

    select @Promedio_edades = AVG(jug.edad)
    from jugador as jug
    inner join equipo as equi on jug.id_equipo = equi.id_equipo
    where jug.edad > 21 and equi.categoria = 'Mujeres'
    return @Promedio_edades

end;
```

```
select dbo.F3_PromedioEdades() as edad_and_categoria
```

--crear una funcion que permita concatenar 3 parametros

--

```

ALTER FUNCTION F4_Concattems(@PAR1 VARCHAR(80), @PAR2 VARCHAR(80), @PAR3
VARCHAR(80))
    RETURNS VARCHAR(50) AS
    BEGIN
    DECLARE @RESULT VARCHAR(80)
    SET @RESULT = concat(@PAR1, ' ', @PAR2, ' ', @PAR3)
    RETURN @RESULT;
END;

```

```

SELECT dbo.F4_Concattems(JUG.NOMBRES, EQ.NOMBRE_DE_EQUIPO, CAMP.SEDE)
FROM JUGADOR AS JUG
INNER JOIN EQUIPO AS EQ ON JUG.ID_EQUIPO = EQ.ID_EQUIPO
INNER JOIN CAMPEONATO AS CAMP ON EQ.ID_CAMPEONATO = CAMP.ID_CAMPEONATO;

```

--GENERAR LA SERIE FIBONACCI

```

WITH Fibonacci (PrevN, N) AS
(
    SELECT 0, 1
    UNION ALL
    SELECT N, PrevN + N
    FROM Fibonacci
    WHERE N < 1000000000
)
SELECT Substring(
    (SELECT cast(', ' as varchar(max)) + cast(PrevN as varchar(max))
    FROM Fibonacci
    FOR XML PATH(''),3,10000000) AS list

```