

PROCESUAL H2 BASE DE DATOS I

ESTUDIANTE: BRITTANY IBLING MARINO QUISPE

UNIVERSIDAD: UNIVERSIDAD FRANZ TAMAYO

CARRERA: INGENIERÍA DE SISTEMAS

CÓDIGO: SIS13181271

1. DISEÑO DE BASE DE DATOS.

1.1. DADO EL DETALLE EXPLICADO EN LA PARTE INICIAL DE ESTE DOCUMENTO DEBERÍA GENERAR UNA BASE DE DATOS SIMILAR AL SIGUIENTE.

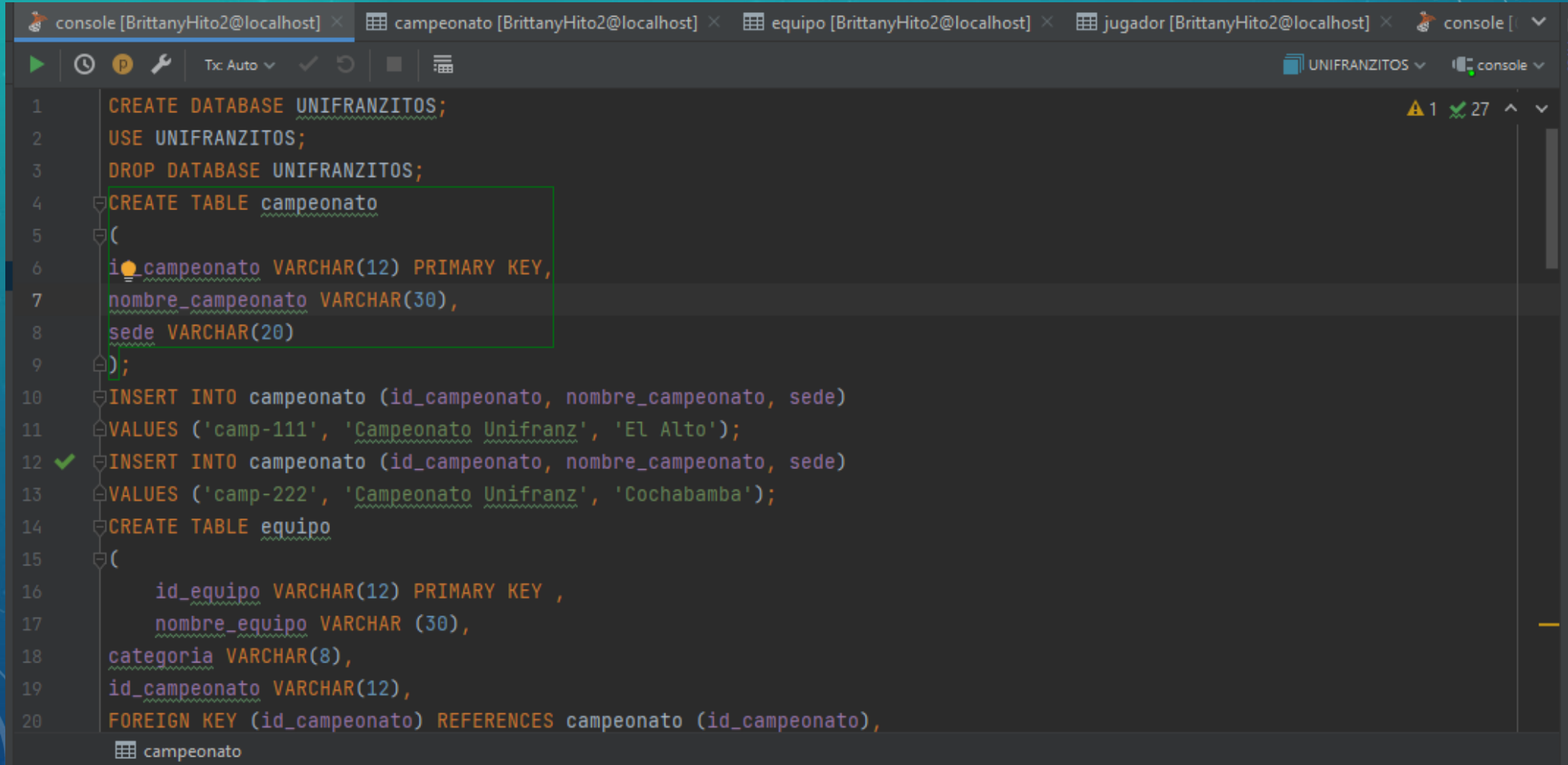
Detalle del problema				
UNIFRANZITOS				
Problema	<p>Se tiene como contexto un CAMPEONATO DE FÚTBOL en el cual se tiene 3 entidades principales el campeonato como tal los equipos que participaran en el campeonato y en donde cada equipo tendrá una cantidad de jugadores.</p>			
	<p>En tal sentido se deberá crear las siguientes tablas.</p> <ul style="list-style-type: none">• campeonato• equipo• jugador			
	<p>Detalle de las tablas.</p> <table><tr><th>campeonato</th></tr><tr><td><code>id_campeonato</code> => cadena de 12 caracteres y ademas llave primaria</td></tr><tr><td><code>nombre_campeonato</code> => una cadena de 30 caracteres que no acepta valores nulos</td></tr><tr><td><code>sede</code> => una cadena de 20 caracteres que no acepta valores nulos</td></tr></table>	campeonato	<code>id_campeonato</code> => cadena de 12 caracteres y ademas llave primaria	<code>nombre_campeonato</code> => una cadena de 30 caracteres que no acepta valores nulos
campeonato				
<code>id_campeonato</code> => cadena de 12 caracteres y ademas llave primaria				
<code>nombre_campeonato</code> => una cadena de 30 caracteres que no acepta valores nulos				
<code>sede</code> => una cadena de 20 caracteres que no acepta valores nulos				

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

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

Ejecute el comando CREATE DATABASE UNIFRANZITOS

Posicionarnos en la base que queremos trabajar. Ejecute el comando: USE UNIFRANZITOS



The screenshot shows a SQL IDE interface with a dark theme. The top bar displays several tabs: 'console [BrittanyHito2@localhost]', 'campeonato [BrittanyHito2@localhost]', 'equipo [BrittanyHito2@localhost]', 'jugador [BrittanyHito2@localhost]', and another 'console [BrittanyHito2@localhost]'. The main editor area contains SQL code for creating a database, using it, dropping it, and then creating two tables: 'campeonato' and 'equipo'. The 'campeonato' table has columns 'id_campeonato' (primary key), 'nombre_campeonato', and 'sede'. The 'equipo' table has columns 'id_equipo' (primary key), 'nombre_equipo', 'categoria', 'id_campeonato' (foreign key), and 'FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato)'. The code is numbered from 1 to 20. A green box highlights the 'CREATE TABLE campeonato' statement. A green checkmark is visible next to line 12. The bottom status bar shows 'campeonato'.

```
1 CREATE DATABASE UNIFRANZITOS;
2 USE UNIFRANZITOS;
3 DROP DATABASE UNIFRANZITOS;
4 CREATE TABLE campeonato
5 (
6     id_campeonato VARCHAR(12) PRIMARY KEY,
7     nombre_campeonato VARCHAR(30),
8     sede VARCHAR(20)
9 );
10 INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)
11 VALUES ('camp-111', 'Campeonato Unifranz', 'El Alto');
12 INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)
13 VALUES ('camp-222', 'Campeonato Unifranz', 'Cochabamba');
14 CREATE TABLE equipo
15 (
16     id_equipo VARCHAR(12) PRIMARY KEY ,
17     nombre_equipo VARCHAR (30),
18     categoria VARCHAR(8),
19     id_campeonato VARCHAR(12),
20     FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato),
```

PARA CREAR UNA TABLA ME APOYE EN LA CLÁUSULA: CREATE TABLE
NOMBRE_TABLA (...);
CÓMO AGREGAR REGISTROS A UNA TABLA. ME APOYE EN LA CLÁUSULA
INSERT INTO NOMBRE_TABLA

```
console [BrittanyHito2@localhost]
Add Configuration...
console [BrittanyHito2@localhost] x campeonato [BrittanyHito2@localhost] x equipo [BrittanyHito2@localhost] x jugador [BrittanyHito2@localhost] x console [
Tic Auto ✓ ↺
19 id_campeonato VARCHAR(12),
20 FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato),
21 );
22
23 INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)
24 VALUES ('equ-111', 'Google', 'VARONES', 'camp-111');
25 INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)
26 VALUES ('equ-222', '404 Not found', 'VARONES', 'camp-111');
27 INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)
28 VALUES ('equ-333', 'girls Unifranz', 'MUJERES', 'camp-111');
29
30 CREATE TABLE jugador
31 (
32 id_jugador VARCHAR(12) PRIMARY KEY,
33 nombres VARCHAR (30),
34 apellidos VARCHAR(50),
35 ci VARCHAR(15),
36 edad INT,
37 id_equipo VARCHAR(12),
38 FOREIGN KEY (id_equipo) REFERENCES equipo (id_equipo),
campeonato
[2021-09-16 23:39:29] 1 row affected in 3 ms
```


2. MANEJO DE CONCEPTOS

- **2.1. Que es DDL.**
- **Un lenguaje de base de datos o lenguaje de definición de datos, un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los programadores de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.**
- **2.2. Que es DML.**
- **Lenguaje de Manipulación de Datos (Data Manipulation Language, DML) es un lenguaje proporcionado por los sistemas gestores e bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos.**
- **2.3. Que son Bases de Datos relacionales y no relacionales.**
- **- Las bases de datos relacionales utilizan un lenguaje de consulta estructurado para la manipulación de datos, estas se conforman por filas, columnas y registros y se almacenan por tablas. Para manipular los datos en SQL, se requiere primero determinar la estructura de estos, si se cambia la estructura de uno de los datos, puede perjudicar todo el sistema, ya que las tablas están relacionadas.**
- **-Las bases de datos no relacionales tienen un esquema dinámico, no se requiere la estructura de los datos para su manipulación. Los datos se pueden almacenar de cualquier manera, columnas, documentos, gráficos, etc, y cada documento puede tener su propia estructura, sin afectar los demás, puede agregar más campos a medida que se avanza.**

- **2.4. Qué es SQL.**
- **Es un tipo de lenguaje de programación que ayuda a solucionar problemas específicos o relacionados con la definición, manipulación e integridad de la información representada por los datos que se almacenan en las bases de datos. Algunos aspectos de SQL están basados en el cálculo relacional, algunos en el álgebra relacional que provienen del modelo relacional y otros a ninguno de los dos sino que son parte de SQL.**
- **2.5. Cuando debería de usarse una Base de Datos relacional y no relacional.**
- **- Las bases de datos relacionales se basan en la organización de la información en partes pequeñas que se integran mediante identificadores; a diferencia de las bases de datos no relacionales que, como su nombre lo indica, no tienen un identificador que sirva para relacionar dos o más conjuntos de datos. Además, son más robustas, es decir, tienen mayor capacidad de almacenamiento, y son menos vulnerables ante fallas, estas son sus principales características. - La información se organiza normalmente mediante documentos y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar.**
- **2.6. Qué tipo de base de datos es SQL Server.**
- **Es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.**

- **2.7. Mencione algunos métodos de autenticación para acceder a una instancia SQL Server.**
- Hay dos posibles modos: modo de autenticación de Windows y modo mixto. El modo de autenticación de Windows habilita la autenticación de Windows y deshabilita la autenticación de SQL Server . El modo mixto habilita tanto la autenticación de Windows como la de SQL Server . La autenticación de Windows está disponible siempre y no se puede deshabilitar.
- **2.8. Qué tipo de licencia tiene una base de datos SQL Server.**
- El tipo de licencia es: Microsoft EULA
- **2.9. Que es una tabla**
- Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro único y cada columna un campo dentro del registro.
- **2.10. Que significa PRIMARY KEY y FOREIGN KEY**
- - PRIMARY KEY (Llave primaria) se utiliza para definir la clave principal de la tabla. Las columnas que forman la clave principal no pueden contener valores nulos ni puede haber valores duplicados de la combinación de columnas.
- -FOREIGN KEY (Llave foránea) sirve para definir una clave foránea sobre una columna o una combinación de columnas. Una clave foránea es una columna o conjunto de columnas que contiene un valor que hace referencia a una fila de otra tabla.

3. Manejo de consultas

3.1. Mostrar que jugadores que formen parte del equipo equ-222

```
SELECT *  
FROM jugador  
WHERE jugador.id_equipo = 'EQU-222';
```

```
1 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
2 VALUES ('jug-111', 'carlos', 'Villa', '8997811LP','19', 'EQU-222');  
3 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
4 VALUES ('jug-222', 'Pedro', 'Salas', '8997822LP','20', 'EQU-222');  
5 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
6 VALUES ('jug-333', 'Saul', 'Araj', '8997833LP','21', 'EQU-222');  
7 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
8 VALUES ('jug-444', 'Sandra', 'Solis', '8997844LP','20', 'EQU-333');  
9 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
0 VALUES ('jug-555', 'Ana', 'Mica', '8997855LP','23', 'EQU-333');  
1 ✓ SELECT *  
2 FROM jugador  
3 WHERE jugador.id_equipo = 'EQU-222';  
4
```

Output x UNIFRANZITOS.dbo.jugador x

	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

3.2. Mostrar que jugadores que formen parte del equipo equ-333

```
SELECT *  
FROM jugador  
WHERE jugador.id_equipo = 'EQU-333';
```

```
41 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
42 VALUES ('jug-111', 'carlos', 'Villa', '8997811LP','19', 'EQU-222');  
43 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
44 VALUES ('jug-222', 'Pedro', 'Salas', '8997822LP','20', 'EQU-222');  
45 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
46 VALUES ('jug-333', 'Saul', 'Araja', '8997833LP','21', 'EQU-222');  
47 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
48 VALUES ('jug-444', 'Sandra', 'Solis', '8997844LP','20', 'EQU-333');  
49 INSERT INTO jugador (id_jugador, nombres, apellidos,ci,edad,id_equipo)  
50 VALUES ('jug-555', 'Ana', 'Mica', '8997855LP','23', 'EQU-333');  
51 ✓ SELECT *  
52 FROM jugador  
53 WHERE jugador.id_equipo = 'EQU-333';  
54
```

Output x UNIFRANZITOS.dbo.jugador x

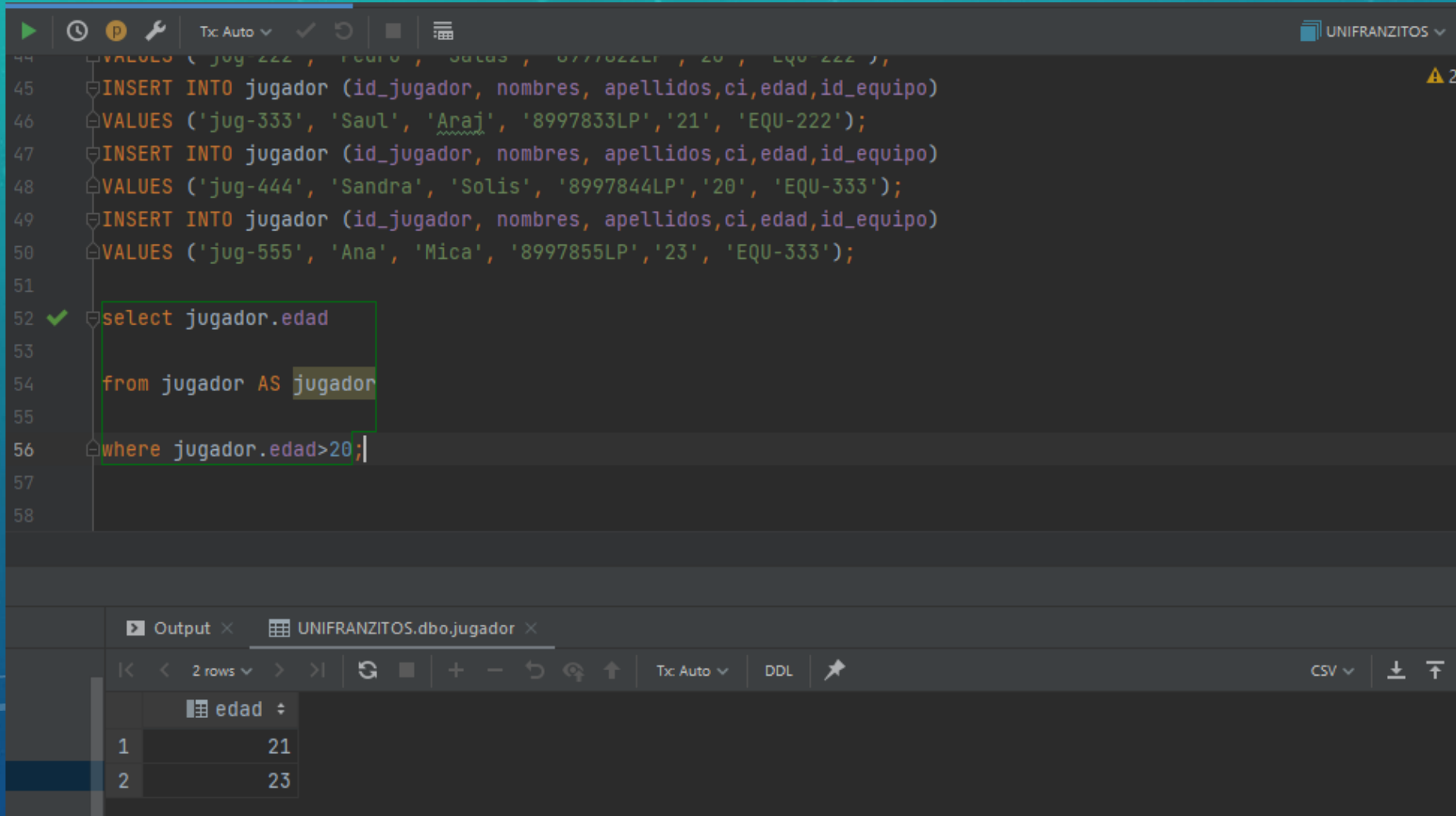
	id_jugador	nombres	apellidos	ci	edad	id_equipo
1	jug-444	Sandra	Solis	8997844LP	20	EQU-333
2	jug-555	Ana	Mica	8997855LP	23	EQU-333

3.3. Mostrar aquellos jugadores mayores o igual a 21 años

```
select jugador.edad
```

```
from jugador AS jugador
```

```
where jugador.edad>20;
```



The screenshot shows a SQL IDE interface. The main editor contains a script with several INSERT statements and a SELECT statement. The SELECT statement is highlighted with a green box. Below the editor, there is a results window titled "Output" and "UNIFRANZITOS.dbo.jugador". The results window shows two rows of data, with the column "edad" selected.

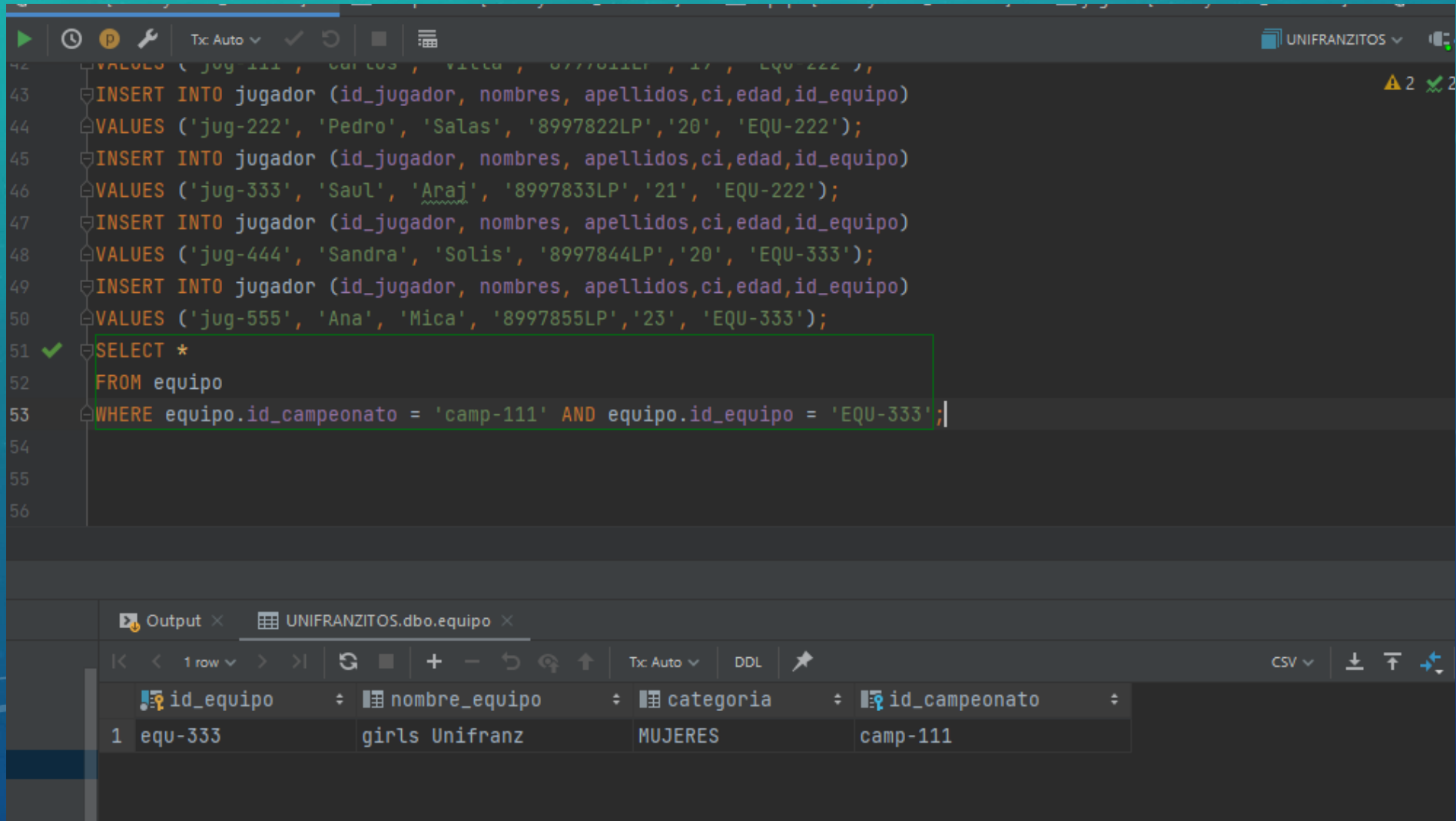
```
44 VALUES ('jug-222', 'Pedro', 'Salas', '8997822LP', '20', 'EQU-222');
45 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
46 VALUES ('jug-333', 'Saul', 'Araja', '8997833LP', '21', 'EQU-222');
47 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
48 VALUES ('jug-444', 'Sandra', 'Solis', '8997844LP', '20', 'EQU-333');
49 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
50 VALUES ('jug-555', 'Ana', 'Mica', '8997855LP', '23', 'EQU-333');
51
52 select jugador.edad
53
54 from jugador AS jugador
55
56 where jugador.edad>20;
57
58
```

Output: UNIFRANZITOS.dbo.jugador

	edad
1	21
2	23

3.4. Mostrar que equipos forman parte del campeonato camp-111 y además sean de la categoría MUJERES.

```
SELECT *  
FROM equipo  
WHERE equipo.id_campeonato = 'camp-111' AND equipo.id_equipo = 'EQU-333';
```



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
VALUES ('jug-111', 'Carlos', 'Villar', '8777011LP', '17', 'EQU-222');  
INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
VALUES ('jug-222', 'Pedro', 'Salas', '8997822LP', '20', 'EQU-222');  
INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
VALUES ('jug-333', 'Saul', 'Araja', '8997833LP', '21', 'EQU-222');  
INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
VALUES ('jug-444', 'Sandra', 'Solis', '8997844LP', '20', 'EQU-333');  
INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
VALUES ('jug-555', 'Ana', 'Mica', '8997855LP', '23', 'EQU-333');  
SELECT *  
FROM equipo  
WHERE equipo.id_campeonato = 'camp-111' AND equipo.id_equipo = 'EQU-333';
```

The results pane shows the output of the query, displaying a single row of data for the team 'equ-333'.

id_equipo	nombre_equipo	categoria	id_campeonato
1 equ-333	girls Unifranz	MUJERES	camp-111

3.5. Mostrar el nombre del equipo del jugador con id_jugador igual a jug-333

```
SELECT *  
FROM jugador  
WHERE jugador.id_jugador = 'jug-333';
```

console [BrittanyHito2@localhost] x campeonato [BrittanyHito2@localhost] x equipo [BrittanyHito2@localhost] x jugador [BrittanyHito2@localhost] x

UNIFRANZITOS

```
42 VALUES ('jug-111', 'Carlos', 'Villa', '8997811LP', '17', 'EQU-222');  
43 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
44 VALUES ('jug-222', 'Pedro', 'Salas', '8997822LP', '20', 'EQU-222');  
45 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
46 VALUES ('jug-333', 'Saul', 'Araja', '8997833LP', '21', 'EQU-222');  
47 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
48 VALUES ('jug-444', 'Sandra', 'Solis', '8997844LP', '20', 'EQU-333');  
49 INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
50 VALUES ('jug-555', 'Ana', 'Mica', '8997855LP', '23', 'EQU-333');  
51  
52 ✓ SELECT *  
53 FROM jugador  
54 WHERE jugador.id_jugador = 'jug-333';  
55  
56
```

Output x UNIFRANZITOS.dbo.jugador x

1 row

	id_jugador	nombres	apellidos	ci	edad	id_equipo
1	jug-333	Saul	Araja	8997833LP	21	EQU-222

3.6. Mostrar el nombre del campeonato del jugador con id_jugador igual a jug-333

```
SELECT *  
FROM jugador , equipo  
WHERE jugador.id_jugador = 'jug-333' AND equipo.id_equipo='equ-222';
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure, including the 'jugador' and 'equipo' tables. The central pane shows a SQL query being executed. The bottom pane displays the results of the query in a table format.

Query Executed:

```
INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
VALUES ('jug-444', 'Sandra', 'Solis', '8997844LP', '20', 'equ-333');  
INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)  
VALUES ('jug-555', 'Ana', 'Mica', '8997855LP', '23', 'equ-333');  
SELECT *  
FROM jugador , equipo  
WHERE jugador.id_jugador = 'jug-333' AND equipo.id_equipo='equ-222';  
SELECT *
```

Query Results (1 row):

	nombres	apellidos	ci	edad	jugador.id_equipo	equipo.id_equipo	nombre_equipo	categoria	id_campeonato
1	Saul	Araj	8997833LP	21	equ-222	equ-222	404 Not found	VARONES	camp-111

3.7. Determinar cuántos jugadores pertenecen a la categoría VARONES.

```
SELECT *  
FROM equipo,  
jugador  
WHERE equipo.categoria = 'VARONES' AND jugador.id_equipo='equ-222';
```

The screenshot shows a database IDE with a console window displaying a SQL query and its results. The query is:

```
SELECT *  
FROM equipo,  
jugador  
WHERE equipo.categoria = 'VARONES' AND jugador.id_equipo='equ-222';
```

The results are displayed in a table with 10 columns: equipo.id_equipo, nombre_equipo, categoria, id_campeonato, id_jugador, nombres, apellidos, ci, edad, and jugador. The table contains 6 rows of data.

	equipo.id_equipo	nombre_equipo	categoria	id_campeonato	id_jugador	nombres	apellidos	ci	edad	jugador
1	equ-111	Google	VARONES	camp-111	jug-111	carlos	Villa	8997811LP	19	equ-222
2	equ-111	Google	VARONES	camp-111	jug-222	Pedro	Salas	8997822LP	20	equ-222
3	equ-111	Google	VARONES	camp-111	jug-333	Saul	Araj	8997833LP	21	equ-222
4	equ-222	404 Not found	VARONES	camp-111	jug-111	carlos	Villa	8997811LP	19	equ-222
5	equ-222	404 Not found	VARONES	camp-111	jug-222	Pedro	Salas	8997822LP	20	equ-222
6	equ-222	404 Not found	VARONES	camp-111	jug-333	Saul	Araj	8997833LP	21	equ-222

LINK DEL VIDEO: