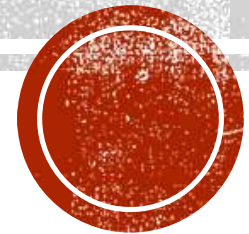


# PROCESUAL

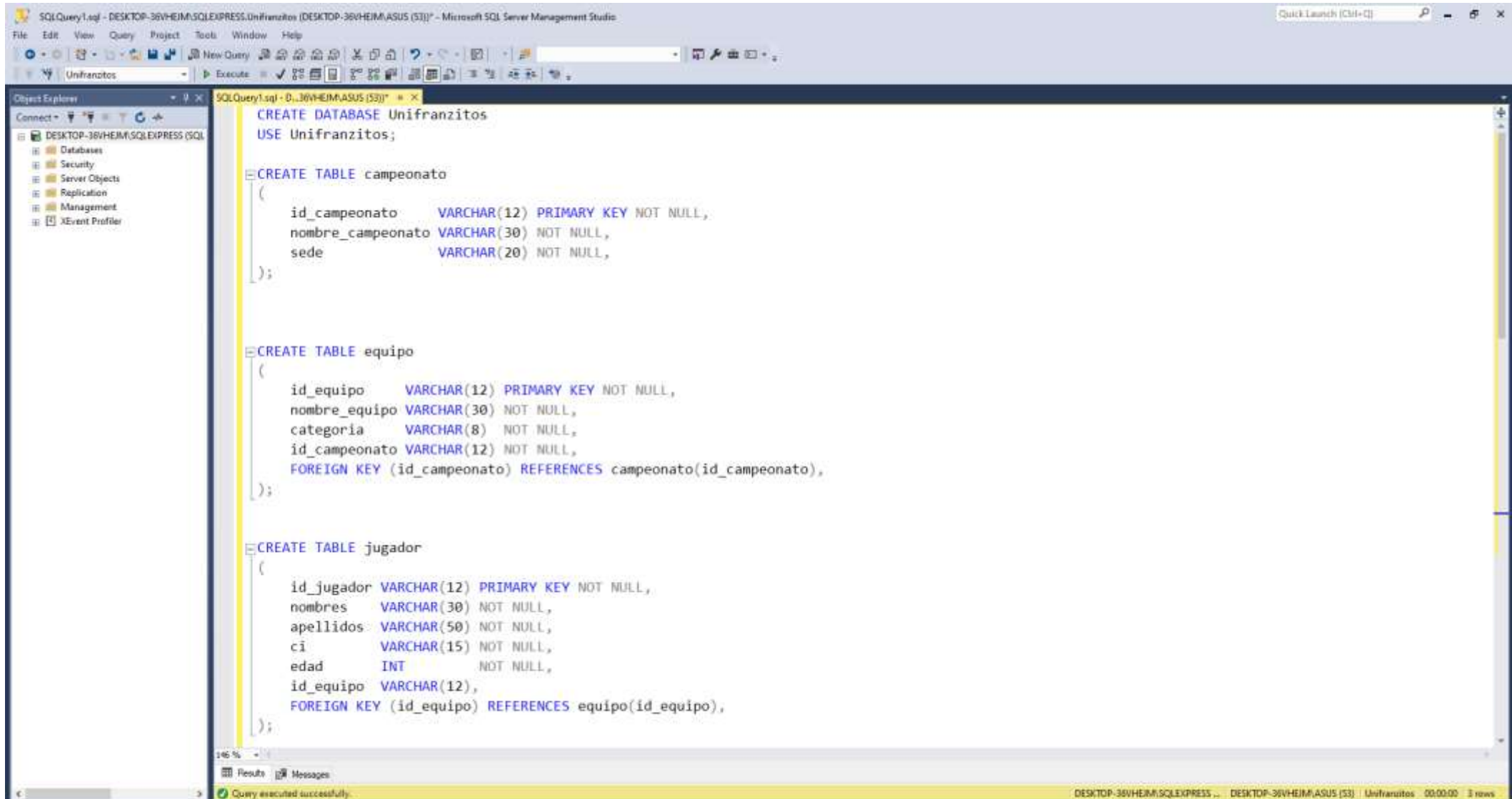
**EJERCICIO CON SU RESPECTIVO DIAGRAMA**

**NOMBRE: EZEQUIEL LAUTARO MORAN CARVAJAL**



# 1. DISEÑO DE BASE DE DATOS

## 1.1 CREAR EL DISEÑO PARA UN CAMPEONATO



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left shows the 'Unifranzitos' database selected. The 'SQL Query' window on the right contains the following SQL script:

```
CREATE DATABASE Unifranzitos
USE Unifranzitos;

CREATE TABLE campeonato
(
    id_campeonato VARCHAR(12) PRIMARY KEY NOT NULL,
    nombre_campeonato VARCHAR(30) NOT NULL,
    sede VARCHAR(20) NOT NULL,
);

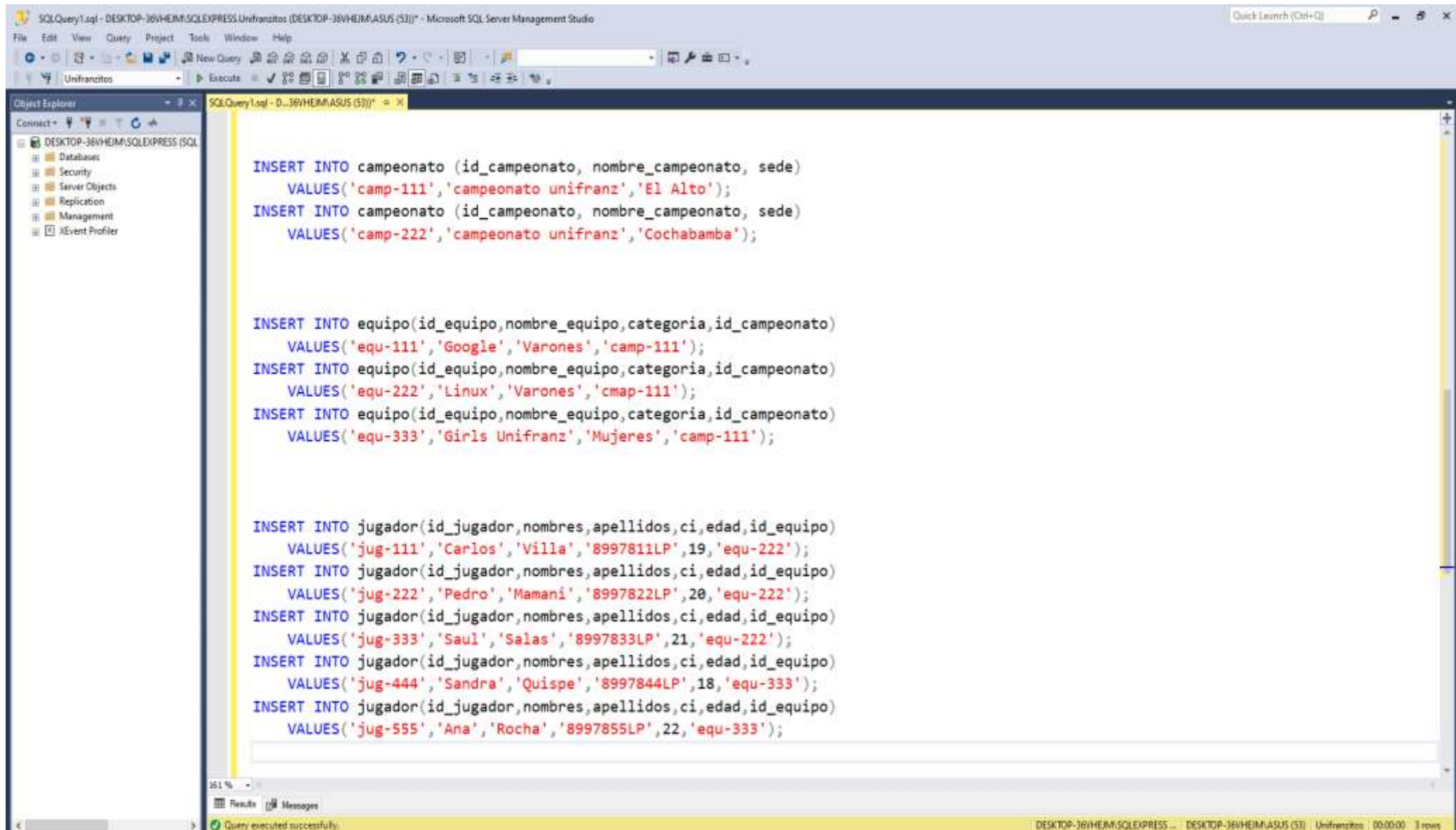
CREATE TABLE equipo
(
    id_equipo VARCHAR(12) PRIMARY KEY NOT NULL,
    nombre_equipo VARCHAR(30) NOT NULL,
    categoria VARCHAR(8) NOT NULL,
    id_campeonato VARCHAR(12) NOT NULL,
    FOREIGN KEY (id_campeonato) REFERENCES campeonato(id_campeonato),
);

CREATE TABLE jugador
(
    id_jugador VARCHAR(12) PRIMARY KEY NOT NULL,
    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),
);
```

The status bar at the bottom indicates 'Query executed successfully.' and '3 rows'.



# 12.CREAR LOS REGISTROS DE LAS TABLAS



The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows the server 'DESKTOP-36VHEIM\SQLEXPRESS (SQL)'. The main query window contains the following SQL code:

```
INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)
VALUES('camp-111','campeonato unifranz','El Alto');
INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)
VALUES('camp-222','campeonato unifranz','Cochabamba');

INSERT INTO equipo(id_equipo,nombre_equipo,categoria,id_campeonato)
VALUES('equ-111','Google','Varones','camp-111');
INSERT INTO equipo(id_equipo,nombre_equipo,categoria,id_campeonato)
VALUES('equ-222','Linux','Varones','camp-111');
INSERT INTO equipo(id_equipo,nombre_equipo,categoria,id_campeonato)
VALUES('equ-333','Girls Unifranz','Mujeres','camp-111');

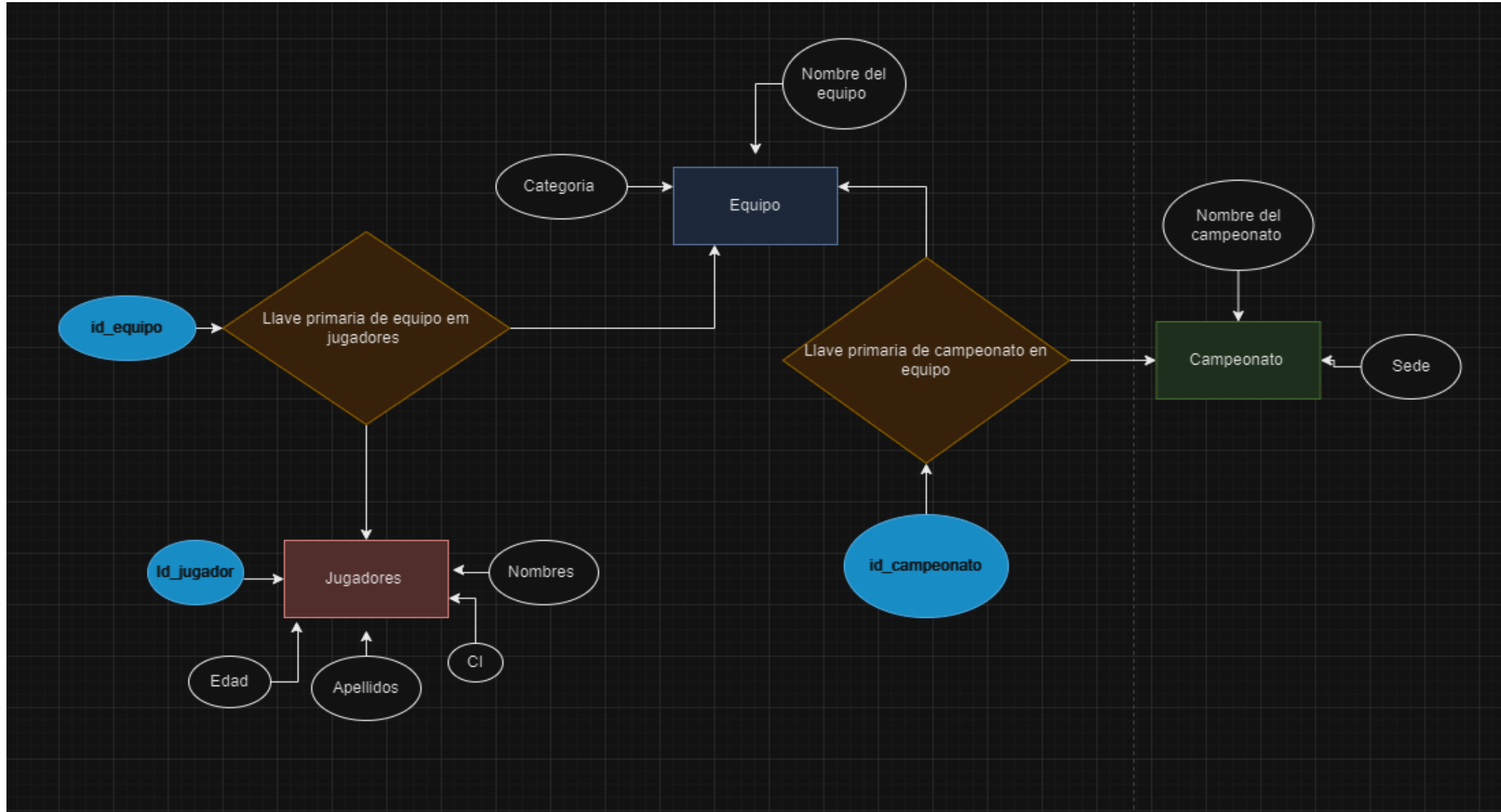
INSERT INTO jugador(id_jugador,nombres,apellidos,ci,edad,id_equipo)
VALUES('jug-111','Carlos','Villa','8997811LP',19,'equ-222');
INSERT INTO jugador(id_jugador,nombres,apellidos,ci,edad,id_equipo)
VALUES('jug-222','Pedro','Mamani','8997822LP',20,'equ-222');
INSERT INTO jugador(id_jugador,nombres,apellidos,ci,edad,id_equipo)
VALUES('jug-333','Saul','Salas','8997833LP',21,'equ-222');
INSERT INTO jugador(id_jugador,nombres,apellidos,ci,edad,id_equipo)
VALUES('jug-444','Sandra','Quispe','8997844LP',18,'equ-333');
INSERT INTO jugador(id_jugador,nombres,apellidos,ci,edad,id_equipo)
VALUES('jug-555','Ana','Rocha','8997855LP',22,'equ-333');
```

The status bar at the bottom indicates 'Query executed successfully.' and shows the server name 'DESKTOP-36VHEIM\SQLEXPRESS', the database 'DESKTOP-36VHEIM\ASUS (3)', and the time '00:00:00'.



# 2.MANEJO DE CONCEPTOS

## 2.1 ADJUNTAR EL DIAGRAMA.





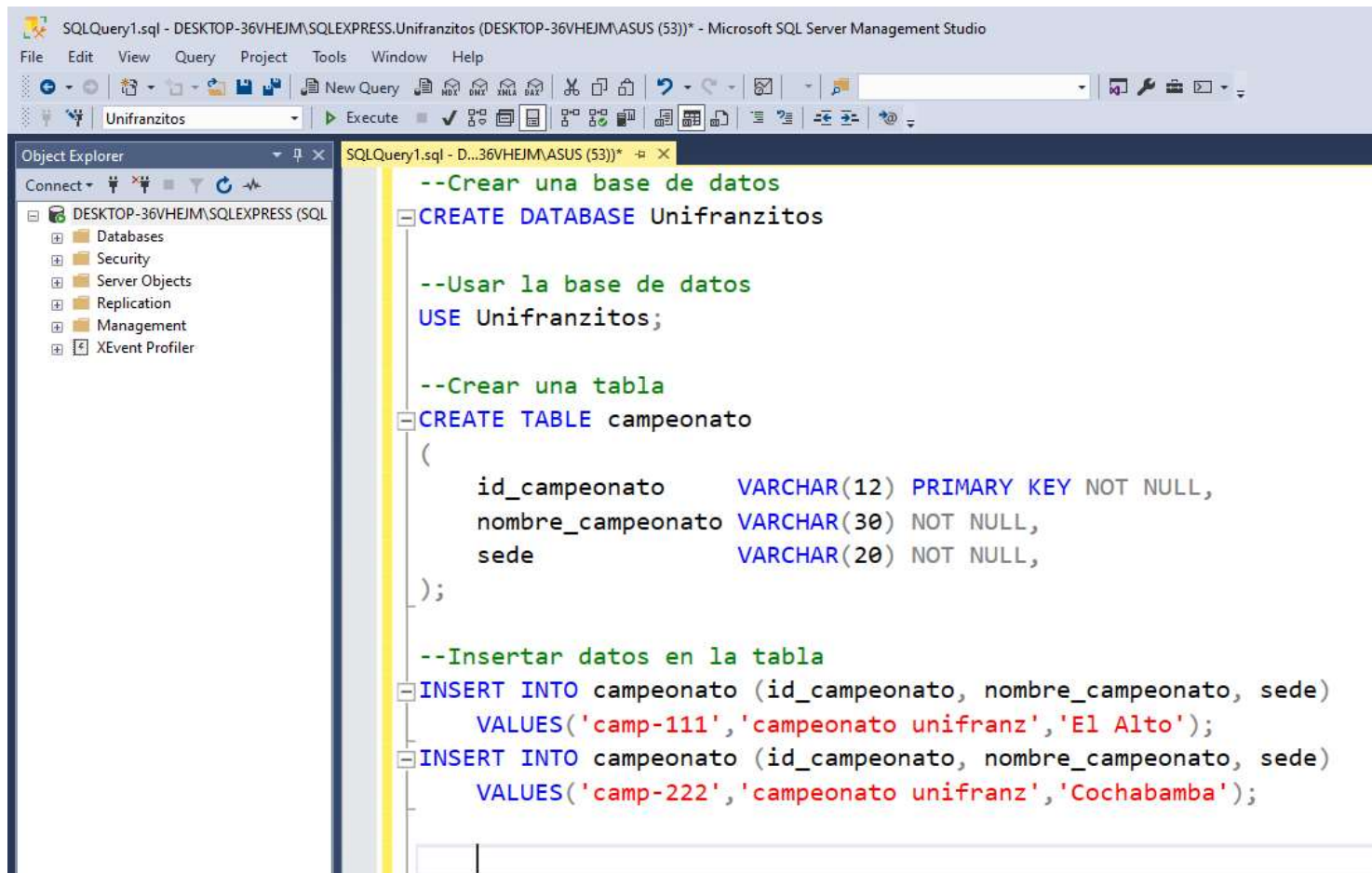
## 2.2 ¿QUÉ ES DDL Y MDL? ADICIONALMENTE MUESTRA UN EJEMPLO EN LA BASE DE DATOS UNIFRANZITOS.

R.- DDL (DATA DEFINITION LANGUAGE) SE UTILIZA PARA DEFINIR LA ESTRUCTURA DE LA BASE DE DATOS.

DML (DATA MANIPULATION LANGUAGE) SE UTILIZA PARA MANIPULAR LOS DATOS QUE VIVEN EN ESAS ESTRUCTURAS.

EN EL CÓDIGO, SE UTILIZA DDL PARA CREAR UNA TABLA LLAMADA "CAMPEONATO" CON TRES COLUMNAS: "ID\_CAMPEONATO", "NOMBRE\_CAMPEONATO" Y "SEDE"

EN EL CÓDIGO, SE UTILIZA DML PARA INSERTAR DATOS EN LA TABLA "CAMPEONATO".



```
SQLQuery1.sql - DESKTOP-36VHEJM\SQLEXPRESS.Unifranzitos (DESKTOP-36VHEJM\ASUS (53)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
Unifranzitos
Object Explorer
Connect
DESKTOP-36VHEJM\SQLEXPRESS (SQL
  Databases
  Security
  Server Objects
  Replication
  Management
  XEvent Profiler
SQLQuery1.sql - D...36VHEJM\ASUS (53))
--Crear una base de datos
CREATE DATABASE Unifranzitos

--Usar la base de datos
USE Unifranzitos;

--Crear una tabla
CREATE TABLE campeonato
(
    id_campeonato    VARCHAR(12) PRIMARY KEY NOT NULL,
    nombre_campeonato VARCHAR(30) NOT NULL,
    sede             VARCHAR(20) NOT NULL,
);

--Insertar datos en la tabla
INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)
VALUES('camp-111','campeonato unifranz','El Alto');
INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)
VALUES('camp-222','campeonato unifranz','Cochabamba');
```



## **2.3. QUE SIGNIFICA PRIMARY KEY Y FOREIGN KEY.**

R.-EN SQL, PRIMARY KEY Y FOREIGN KEY SON RESTRICCIONES QUE SE UTILIZAN PARA GARANTIZAR LA INTEGRIDAD DE LOS DATOS EN LAS TABLAS DE LA BASE DE DATOS. PRIMARY KEY SE UTILIZA PARA DEFINIR UNA CLAVE PRINCIPAL EN UNA TABLA, QUE ES UNA COLUMNA O UNA COMBINACIÓN DE COLUMNAS QUE IDENTIFICA DE FORMA ÚNICA CADA FILA DE LA TABLA. FOREIGN KEY SE UTILIZA PARA DEFINIR UNA CLAVE FORÁNEA EN UNA TABLA, QUE ES UNA COLUMNA O UNA COMBINACIÓN DE COLUMNAS QUE HACE REFERENCIA A LA CLAVE PRINCIPAL DE OTRA TABLA.

## **2.4. DEFINA QUE ES UNA TABLA Y EL USO DE IDENTITY.**

R.-UNA TABLA ES UNA ESTRUCTURA DE DATOS QUE SE UTILIZA PARA ALMACENAR INFORMACIÓN EN FILAS Y COLUMNAS. CADA FILA REPRESENTA UN REGISTRO EN LA TABLA, Y CADA COLUMNA REPRESENTA UN CAMPO DE DATOS EN EL REGISTRO. IDENTITY ES UNA PROPIEDAD QUE SE UTILIZA PARA CREAR UNA COLUMNA DE IDENTIDAD EN UNA TABLA. LA COLUMNA DE IDENTIDAD SE UTILIZA PARA GENERAR VALORES DE CLAVE ÚNICOS PARA CADA FILA EN LA TABLA. CUANDO SE INSERTA UNA NUEVA FILA EN LA TABLA, EL VALOR DE LA COLUMNA DE IDENTIDAD SE INCREMENTA AUTOMÁTICAMENTE EN UNO. ESTO GARANTIZA QUE CADA FILA TENGA UN VALOR ÚNICO EN LA COLUMNA DE IDENTIDAD.

## **2.5. PARA QUE SE UTILIZA LA CLÁUSULA WHERE.**

R.-SE UTILIZA PARA FILTRAR LAS FILAS DE UNA TABLA QUE CUMPLAN UNA CONDICIÓN.

## **2.6. PARA QUE SE UTILIZA LA INSTRUCCIÓN INNER JOIN.**

R.-SE UTILIZA PARA COMBINAR DOS O MÁS TABLAS EN UNA SOLA TABLA. ESTA INSTRUCCIÓN DEVUELVE SOLO LAS FILAS QUE TIENEN VALORES COINCIDENTES EN AMBAS TABLAS.

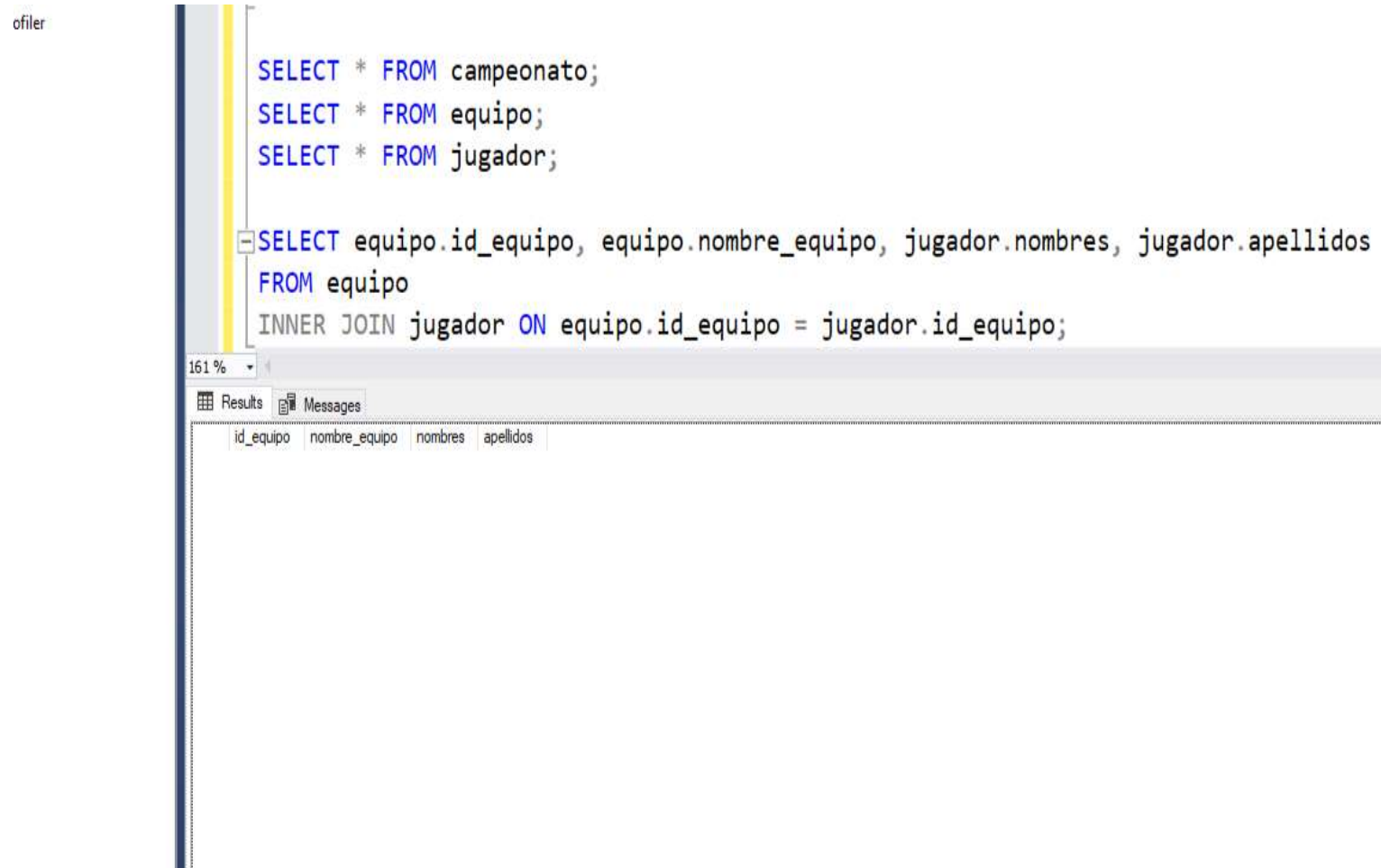


## 2.7. APOYÁNDONOS EN EL CONCEPTO DE CONJUNTOS MUESTRE LOS SIGUIENTE:

### 2.7.1. EJEMPLO DE INNER JOIN

R.-EN ESTE EJEMPLO SE UTILIZA INNER JOIN PARA COMBINAR LAS TABLAS “EQUIPO”Y “JUGADOR” EN UNA SOLA TABLA, EL “ON” SE UTILIZA PARA ESPECIFICAR LA CONDICIÓN DE COMBINACIÓN, QUE ES QUE LOS VALORES EN LA COLUMNA “ID\_EQUIPO” DE AMBAS TABLAS DEBEN COINCIDIR.

### 2.7.2. ADJUNTAR UNA IMAGEN DE CONJUNTOS Y LA CONSULTA SQL QUE REFLEJE EL INNER JOIN



The screenshot shows a SQL IDE interface. The top pane contains the following SQL code:

```
SELECT * FROM campeonato;  
SELECT * FROM equipo;  
SELECT * FROM jugador;  
  
SELECT equipo.id_equipo, equipo.nombre_equipo, jugador.nombres, jugador.apellidos  
FROM equipo  
INNER JOIN jugador ON equipo.id_equipo = jugador.id_equipo;
```

The bottom pane shows the results of the query, with a zoom level of 161%. The results are displayed in a table with the following columns: id\_equipo, nombre\_equipo, nombres, and apellidos.

id_equipo	nombre_equipo	nombres	apellidos
-----------	---------------	---------	-----------



## 2.8. APOYÁNDONOS EN EL CONCEPTO DE CONJUNTOS MUESTRE LOS SIGUIENTE:

### 2.8.1. EJEMPLO DE LEFT JOIN

R-EN ESTE EJEMPLO, SE UTILIZA LEFT JOIN PARA COMBINAR LAS TABLAS “EQUIPO”Y “JUGADOR” EN UNA SOLA TABLA. LA CLÁUSULA ON SE UTILIZA PARA ESPECIFICAR LA CONDICIÓN DE COMBINACIÓN, QUE ES QUE LOS VALORES EN LA COLUMNA “ID\_EQUIPO” DE AMBAS TABLAS DEBEN COINCIDIR. LA CONSULTA DEVUELVE LAS COLUMNAS “ID\_EQUIPO”, “NOMBRE\_EQUIPO”, “NOMBRES”Y “APELLIDOS” DE TODAS LAS FILAS DE LA TABLA “EQUIPO”, INCLUSO SI NO HAY COINCIDENCIAS EN LA TABLA “JUGADOR”.

### 2.8.2. ADJUNTAR UNA IMAGEN DE CONJUNTOS Y LA CONSULTA SQL QUE REFLEJE EL LEFT JOIN

Databases  
Security  
Server Objects  
Replication  
Management  
XEvent Profiler

```
SELECT * FROM jugador;  
  
SELECT equipo.id_equipo, equipo.nombre_equipo, jugador.nombres, jugador.apellidos  
FROM equipo  
INNER JOIN jugador ON equipo.id_equipo = jugador.id_equipo;  
  
SELECT equipo.id_equipo, equipo.nombre_equipo, jugador.nombres, jugador.apellidos  
FROM equipo  
LEFT JOIN jugador ON equipo.id_equipo = jugador.id_equipo;
```

161 %

Results Messages

	id_equipo	nombre_equipo	nombres	apellidos
1	equ-111	Google	NULL	NULL



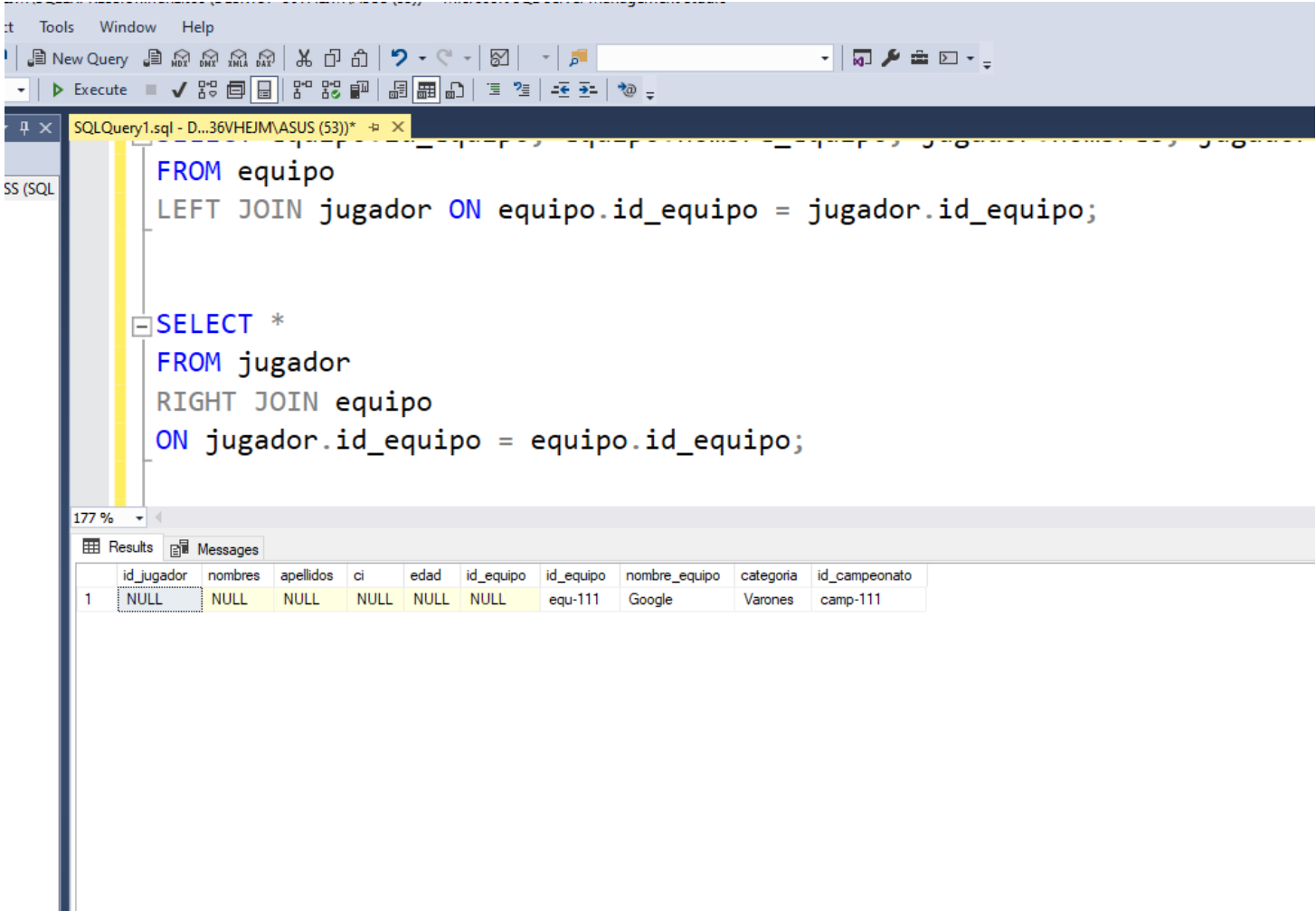


## 2.9. APOYÁNDONOS EN EL CONCEPTO DE CONJUNTOS MUESTRE LOS SIGUIENTE:

### 2.9.1. EJEMPLO DE RIGHT JOIN

R-EN ESTE EJEMPLO, SE UTILIZA RIGHT JOIN PARA COMBINAR LAS TABLAS “EQUIPO”Y “JUGADOR” EN UNA SOLA TABLA. LA CLÁUSULA ON SE UTILIZA PARA ESPECIFICAR LA CONDICIÓN DE COMBINACIÓN, QUE ES QUE LOS VALORES EN LA COLUMNA “ID\_EQUIPO” DE AMBAS TABLAS DEBEN COINCIDIR. LA CONSULTA DEVUELVE LAS COLUMNAS “ID\_EQUIPO”, “NOMBRE\_EQUIPO”, “NOMBRES”Y “APELLIDOS” DE TODAS LAS FILAS DE LA TABLA “JUGADOR”, INCLUSO SI NO HAY COINCIDENCIAS EN LA TABLA “EQUIPO”.

### 2.9.2. ADJUNTAR UNA IMAGEN DE CONJUNTOS Y LA CONSULTA SQL QUE REFLEJE EL RIGHT JOIN



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor displays a SQL query that performs a RIGHT JOIN between the 'jugador' and 'equipo' tables. The results pane shows a single row of data where the 'id\_jugador' is 1, and the other columns are NULL, indicating that there is no matching record in the 'equipo' table for this player.

```
SQLQuery1.sql - D:\36VHEJM\ASUS (53)) * X
FROM equipo
LEFT JOIN jugador ON equipo.id_equipo = jugador.id_equipo;

SELECT *
FROM jugador
RIGHT JOIN equipo
ON jugador.id_equipo = equipo.id_equipo;
```

Results

	id_jugador	nombres	apellidos	ci	edad	id_equipo	id_equipo	nombre_equipo	categoria	id_campeonato
1	NULL	NULL	NULL	NULL	NULL	NULL	equ-111	Google	Varones	camp-111



## 2.10. CREAR 3 TABLAS Y CREAR UNA CONSULTA SQL QUE MUESTRA EL USO DE INNER JOIN.

SQLQuery1.sql - D:\38VHEIM\ASUS (53))

```
CREATE TABLE cliente (  
    id_cliente INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    direccion VARCHAR(100),  
    telefono VARCHAR(20)  
);  
  
CREATE TABLE producto (  
    id_producto INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    descripcion VARCHAR(100),  
    precio DECIMAL(10, 2)  
);  
  
CREATE TABLE venta (  
    id_venta INT PRIMARY KEY,  
    fecha DATE,  
    id_cliente INT,  
    id_producto INT,  
    cantidad INT,  
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente),  
    FOREIGN KEY (id_producto) REFERENCES producto(id_producto)  
);
```

Results

	id_cliente	nombre	direccion	telefono	id_venta	fecha	id_cliente	id_producto	cantidad	id_producto	nombre	descripcion	precio
1	1	Juan Perez	Calle 123	555-1234	1	2023-10-16	1	1	2	1	Producto 1	Descripcion del producto 1	10.99
2	2	Maria González	Calle 456	555-5678	2	2023-10-17	2	2	1	2	Producto 2	Descripcion del producto 2	20.99
3	3	Pedro Pérez	Calle 789	555-9012	3	2023-10-18	3	3	2	3	Producto 3	Descripcion del producto 3	30.99
4	1	Juan Perez	Calle 123	555-1234	4	2023-10-19	1	1	3	1	Producto 1	Descripcion del producto 1	10.99

```
INSERT INTO cliente (id_cliente, nombre, direccion, telefono)  
VALUES (1, 'Juan Perez', 'Calle 123', '555-1234');  
INSERT INTO cliente (id_cliente, nombre, direccion, telefono)  
VALUES (2, 'Maria González', 'Calle 456', '555-5678');  
INSERT INTO cliente (id_cliente, nombre, direccion, telefono)  
VALUES (3, 'Pedro Pérez', 'Calle 789', '555-9012');  
  
INSERT INTO producto (id_producto, nombre, descripcion, precio)  
VALUES (1, 'Producto 1', 'Descripcion del producto 1', 10.99);  
INSERT INTO producto (id_producto, nombre, descripcion, precio)  
VALUES (2, 'Producto 2', 'Descripcion del producto 2', 20.99);  
INSERT INTO producto (id_producto, nombre, descripcion, precio)  
VALUES (3, 'Producto 3', 'Descripcion del producto 3', 30.99);  
  
INSERT INTO venta (id_venta, fecha, id_cliente, id_producto, cantidad)  
VALUES (1, '2023-10-16', 1, 1, 2);  
INSERT INTO venta (id_venta, fecha, id_cliente, id_producto, cantidad)  
VALUES (2, '2023-10-17', 2, 2, 1);  
INSERT INTO venta (id_venta, fecha, id_cliente, id_producto, cantidad)  
VALUES (3, '2023-10-18', 3, 3, 2);  
INSERT INTO venta (id_venta, fecha, id_cliente, id_producto, cantidad)  
VALUES (4, '2023-10-19', 1, 1, 3);  
  
SELECT *  
FROM cliente  
INNER JOIN venta ON cliente.id_cliente = venta.id_cliente  
INNER JOIN producto ON venta.id_producto = producto.id_producto;
```

Results

	id_cliente	nombre	direccion	telefono	id_venta	fecha	id_cliente	id_producto	cantidad	id_producto	nombre	descripcion	precio
1	1	Juan Perez	Calle 123	555-1234	1	2023-10-16	1	1	2	1	Producto 1	Descripcion del producto 1	10.99
2	2	Maria González	Calle 456	555-5678	2	2023-10-17	2	2	1	2	Producto 2	Descripcion del producto 2	20.99
3	3	Pedro Pérez	Calle 789	555-9012	3	2023-10-18	3	3	2	3	Producto 3	Descripcion del producto 3	30.99
4	1	Juan Perez	Calle 123	555-1234	4	2023-10-19	1	1	3	1	Producto 1	Descripcion del producto 1	10.99