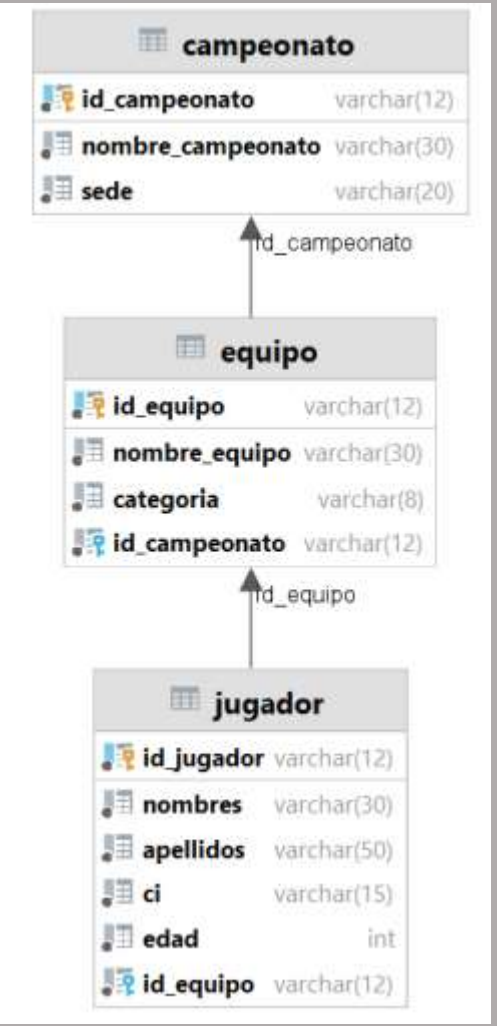
Base de Datos I

Nombre : Cristian Tantani Aguilar
Docente : William Roddy Barra Paredes
Año : 2023

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.



Base de datos generado.

```sql
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(10) 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 INTEGER NOT NULL,
id_equipo VARCHAR(12) NOT NULL,
FOREIGN KEY (id_equipo) REFERENCES equipo (id_equipo),
);
```

```sql
INSERT INTO campeonato(id_campeonato,nombre_campeonato,sede)VALUES
('camp-111','Campeonato Unifranz E','El Alto'),
('camp-222','Campeonato Unifranz C','Cochabamba');
INSERT INTO equipo(id_equipo,nombre_equipo,categoria,id_campeonato)VALUES
('equ-111','Google','VARONES','camp-111'),
('equ-222','404 Not found','VARONES','camp-111'),
('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'),
('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');
```

1.2. Los registros de cada tabla deberían quedar de la siguiente forma

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

Registros de las tablas.

Results | Messages

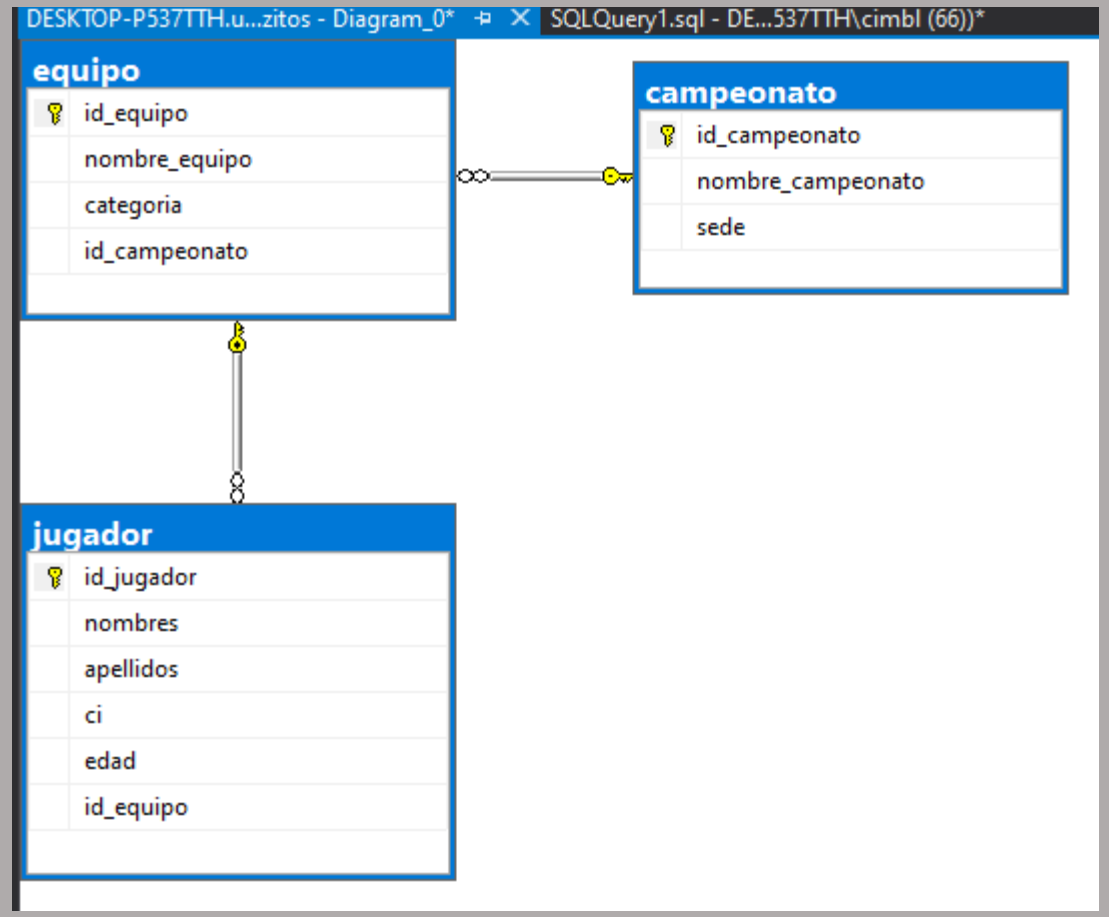
	id_campeonato	nombre_campeonato	sede
1	camp-111	Campeonato Unifranz E	El Alto
2	camp-222	Campeonato Unifranz C	Cochabamba

	id_equipo	nombre_equipo	categoria	id_campeonato
1	equ-111	Google	VARONES	camp-111
2	equ-222	404 Not found	VARONES	camp-111
3	equ-333	girls unifranz	MUJERES	camp-111

	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
4	jug-444	Sandra	Solis	8997844LP	20	equ-333
5	jug-555	Ana	Mica	8997855LP	23	equ-333

2. Manejo de conceptos.

- **2.1. Adjuntar el diagrama E-R GENERADO por su editor (DATAGRIP o SQL SERVER MANAGEMENTS STUDIO)**



2.2. Que es DDL y DML, adicionalmente muestra un ejemplo en la base de datos UNIFRANZITOS.

- DDL significa "Data Definition Language" (Lenguaje de Definición de Datos) y se utiliza para definir la estructura de la base de datos, así como para definir los objetos de la base de datos, como tablas, vistas, índices, restricciones y otros objetos de base de datos. Algunos ejemplos de instrucciones DDL son CREATE TABLE (crear una tabla), ALTER TABLE (alterar una tabla), DROP TABLE (eliminar una tabla) y CREATE INDEX (crear un índice).
- DML significa "Data Manipulation Language" (Lenguaje de Manipulación de Datos) y se utiliza para manipular los datos almacenados en la base de datos. Con DML, se pueden agregar, modificar o eliminar datos en una o varias tablas de la base de datos. Algunos ejemplos de instrucciones DML son SELECT (seleccionar datos de una tabla), INSERT (insertar datos en una tabla), UPDATE (actualizar datos en una tabla) y DELETE (eliminar datos de una tabla).

2.3. Que significa PRIMARY KEY y FOREIGN KEY

- PRIMARY KEY es una columna o conjunto de columnas en una tabla de base de datos que identifica de manera única cada fila de la tabla. Es decir, la PRIMARY KEY garantiza que no habrá duplicados en la tabla. Cada tabla de base de datos debe tener una PRIMARY KEY, y esta clave puede consistir en una o varias columnas de la tabla. Además, la PRIMARY KEY se utiliza para establecer relaciones entre tablas, a través de las FOREIGN KEY.

- FOREIGN KEY es una columna o conjunto de columnas en una tabla que establece una relación con la PRIMARY KEY de otra tabla en la base de datos. La FOREIGN KEY se utiliza para asegurar que los datos almacenados en la tabla estén relacionados con los datos almacenados en otra tabla. Por ejemplo, si tenemos una tabla de clientes y una tabla de pedidos, podemos utilizar la PRIMARY KEY de la tabla de clientes como FOREIGN KEY en la tabla de pedidos para relacionar cada pedido con un cliente específico.

2.4. Defina que es una TABLA y el uso de IDENTITY.

- Una tabla es un objeto fundamental en una base de datos relacional. Una tabla es una colección de datos organizados en filas y columnas. Cada fila representa una entidad o un conjunto de entidades, mientras que cada columna representa un atributo o una propiedad de la entidad. Las tablas son utilizadas para almacenar información de manera estructurada y fácil de consultar.

- IDENTITY es una propiedad que se puede aplicar a una columna de una tabla en una base de datos relacional. Cuando se define una columna con la propiedad IDENTITY, la base de datos automáticamente asigna valores únicos a cada nueva fila que se inserta en la tabla. Es decir, el valor de la columna se genera automáticamente, y se incrementa en 1 para cada nueva fila insertada en la tabla. La propiedad IDENTITY es útil cuando se necesita un valor único y creciente, como un identificador único para cada fila de una tabla.

2.5. Para que se utiliza la cláusula WHERE.

- La cláusula WHERE se utiliza en SQL (Structured Query Language) para filtrar los datos de una tabla en función de una o varias condiciones.
- La cláusula WHERE va seguida de una condición que debe ser verdadera para que se incluya una fila en los resultados de la consulta. La condición puede incluir uno o varios operadores de comparación, como "=", "<", ">", "<=", ">=", "<>", "BETWEEN", "LIKE", "IN", "NOT IN", etc.

2.6. Para que se utiliza la instrucción INNER JOIN.

- La instrucción INNER JOIN se utiliza en SQL para combinar dos o más tablas en una consulta, utilizando una columna común en ambas tablas. La instrucción INNER JOIN devuelve solo las filas que tienen una coincidencia en ambas tablas. En otras palabras, la instrucción INNER JOIN se utiliza para obtener los datos comunes a dos o más tablas.

2.7. Apoyándonos en el concepto de conjuntos muestre los siguiente:

- 2.7.1. Ejemplo de INNER JOIN

```
SELECT*FROM jugador AS jg INNER JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo
WHERE jg.edad>=20 AND eq.categoria = 'VARONES';
```

2.7.2. Adjuntar una imagen de conjuntos y la consulta SQL que refleje el INNER JOIN

```sql
INSERT INTO campeonato(id_campeonato,nombre_campeonato,sede)VALUES
('camp-111','Campeonato Unifranz E','El Alto'),
('camp-222','Campeonato Unifranz C','Cochabamba');
INSERT INTO equipo(id_equipo,nombre_equipo,categoria,id_campeonato)VALUES
('equ-111','Google','VARONES','camp-111'),
('equ-222','404 Not found','VARONES','camp-111'),
('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'),
('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');
-------------------------------------------------------
SELECT*FROM jugador AS jg INNER JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo
WHERE jg.edad>=20 AND eq.categoria = 'VARONES';
-------------------------------------------------------
```

Results | Messages

	id_jugador	nombres	apellidos	ci	edad	id_equipo	id_equipo	nombre_equipo	categoria	id_campeonato
1	jug-222	Pedro	Salas	8997822LP	20	equ-222	equ-222	404 Not found	VARONES	camp-111
2	jug-333	Saul	Araj	8997833LP	21	equ-222	equ-222	404 Not found	VARONES	camp-111

2.8. Apoyándonos en el concepto de conjuntos muestre los siguiente:

- 2.8.1. Ejemplo de LEFT JOIN
- 2.8.2. Adjuntar una imagen de conjuntos y la consulta SQL que refleje el LEFT JOIN

```sql
SELECT jg.id_jugador,eq.id_equipo FROM jugador
AS jg LEFT JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo;
```

132 %

Results | Messages

	id_jugador	id_equipo
1	jug-111	equ-222
2	jug-222	equ-222
3	jug-333	equ-222
4	jug-444	equ-333
5	jug-555	equ-333

2.9. Apoyándonos en el concepto de conjuntos muestre los siguiente:

2.9.1. Ejemplo de RIGHT JOIN

2.9.2. Adjuntar una imagen de conjuntos y la consulta SQL que refleje el RIGHT JOIN

```sql
SELECT jg.id_jugador,eq.id_equipo FROM jugador
AS jg RIGHT JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo;
-------------------------------------------------------------
```

32 %

Results | Messages

	id_jugador	id_equipo
1	NULL	equ-111
2	jug-111	equ-222
3	jug-222	equ-222
4	jug-333	equ-222
5	jug-444	equ-333
6	jug-555	equ-333

2.10. Crear 3 tablas y crear una consulta SQL que muestra el uso de INNER JOIN.

```sql
CREATE DATABASE ejemplo;
USE ejemplo;
CREATE TABLE cliente
(
id_cliente VARCHAR(12) PRIMARY KEY NOT NULL,
nombre VARCHAR(20)NOT NULL,
apellido VARCHAR(20)NOT NULL,
edad INTEGER NOT NULL,
);
CREATE TABLE producto
(
id_produc VARCHAR(12) PRIMARY KEY NOT NULL,
nombre VARCHAR(20)NOT NULL,
precio INTEGER NOT NULL,
);
CREATE TABLE compra
(
id_compra VARCHAR(12) PRIMARY KEY NOT NULL,
pago INTEGER NOT NULL,
id_produc VARCHAR(12) NOT NULL,
id_cliente VARCHAR(12)NOT NULL,
FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente),
FOREIGN KEY (id_produc) REFERENCES producto(id_produc),
);
-----------------------------------------------------------
```

```sql
-----------------------------------------------------------
INSERT INTO cliente(id_cliente,nombre,apellido,edad)VALUES
('10090906','Cristian','Aguilar',20);
INSERT INTO producto(id_produc,nombre,precio)VALUES
('000-111','shampu',3);
INSERT INTO compra(id_compra,pago,id_produc,id_cliente)VALUES
('123-AAA',3,'000-111','10090906');
-----------------------------------------------------------
SELECT cli.id_cliente,com.id_compra,pro.id_produc
FROM cliente AS cli INNER JOIN compra AS com
ON cli.id_cliente=com.id_cliente
INNER JOIN producto AS pro ON pro.id_produc=com.id_produc;
```

132 %

Results | Messages

	id_cliente	id_compra	id_produc
1	10090906	123-AAA	000-111

3 Manejo de Consultas

- 3.1. Mostrar que jugadores que son del equipo equ-222

```sql
SELECT jg.nombres,eq.nombre_equipo,eq.id_equipo FROM jugador AS jg
INNER JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo
WHERE eq.id_equipo = 'equ-222';
```

132 %

Results | Messages

	nombres	nombre_equipo	id_equipo
1	Carlos	404 Not found	equ-222
2	Pedro	404 Not found	equ-222
3	Saul	404 Not found	equ-222

- 3.2. Mostrar que jugadores(nombres, apellidos) que juegan en la sede de El Alto.

```sql
SELECT jg.nombres,jg.apellidos,camp.sede FROM jugador AS jg
INNER JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo
INNER JOIN campeonato AS camp ON camp.id_campeonato=eq.id_campeonato
WHERE camp.sede = 'El Alto';
```

132 %

Results | Messages

	nombres	apellidos	sede
1	Carlos	Villa	El Alto
2	Pedro	Salas	El Alto
3	Saul	Araj	El Alto
4	Sandra	Solis	El Alto
5	Ana	Mica	El Alto

3.3. Mostrar aquellos jugadores mayores o igual a 21 años que sean de la categoría VARONES.

```sql
SELECT * FROM jugador AS jg
INNER JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo
WHERE jg.edad>=21 AND eq.categoria = 'VARONES';
```

132 %

Results | Messages

	id_jugador	nombres	apellidos	ci	edad	id_equipo	id_equipo	nombre_equipo	categoria	id_campeonato
1	jug-333	Saul	Araj	8997833LP	21	equ-222	equ-222	404 Not found	VARONES	camp-111

- 3.4. Mostrar a todos los estudiantes en donde su apellido empiece con la letra S.
- 3.4.1. Podría utilizar la instrucción LIKE

```
-------------------------------------------------------
SELECT * FROM jugador AS jg WHERE jg.apellidos LIKE  '%S%';
```

132 %

Results | Messages

	id_jugador	nombres	apellidos	ci	edad	id_equipo
1	jug-222	Pedro	Salas	8997822LP	20	equ-222
2	jug-444	Sandra	Solis	8997844LP	20	equ-333

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

```sql
SELECT * FROM equipo AS eq
INNER JOIN campeonato AS camp ON eq.id_campeonato=camp.id_campeonato
WHERE camp.id_campeonato = 'camp-111' AND eq.categoria = 'MUJERES';
```

132 %

Results | Messages

	id_equipo	nombre_equipo	categoria	id_campeonato	id_campeonato	nombre_campeonato	sede
1	equ-333	girls unifranz	MUJERES	camp-111	camp-111	Campeonato Unifranz E	El Alto

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

```sql
-------------------------------------------------------------
SELECT eq.nombre_equipo FROM equipo AS eq
INNER JOIN jugador AS jg ON eq.id_equipo=jg.id_equipo
WHERE jg.id_jugador = 'jug-333';
```

132 %

Results | Messages

	nombre_equipo
1	404 Not found

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

```sql
----------------------------------------------------------
SELECT camp.nombre_campeonato FROM campeonato AS camp
INNER JOIN equipo AS eq ON camp.id_campeonato=eq.id_campeonato
INNER JOIN jugador AS jg ON jg.id_equipo=eq.id_equipo
WHERE jg.id_jugador = 'jug-333';
----------------------------------------------------------
```

132 %

Results | Messages

	nombre_campeonato
1	Campeonato Unifranz E

3.8. Crear una consulta SQL que maneje las 3 tablas de la base de datos.

```sql
SELECT jg.nombres,camp.id_campeonato,eq.id_equipo
FROM campeonato AS camp
INNER JOIN equipo AS eq ON camp.id_campeonato=eq.id_campeonato
INNER JOIN jugador AS jg ON jg.id_equipo=eq.id_equipo
WHERE jg.nombres LIKE '%C%';
-------------------------------------------------------
```

132 %

Results | Messages

	nombres	id_campeonato	id_equipo
1	Carlos	camp-111	equ-222

- 3.9. ¿Qué estrategia utilizaría para determinar cuántos equipos inscritos hay?
- 3.9.1. Podría utilizar la función de agregación COUNT

```sql
SELECT COUNT(eq.nombre_equipo) FROM equipo AS eq
INNER JOIN campeonato AS camp ON eq.id_campeonato=camp.id_campeonato
```

132 %

Results | Messages

	(No column name)
1	3

- 3.10. ¿Qué estrategia utilizaría para determinar cuántos jugadores pertenecen a la categoría VARONES o Categoría MUJERES.
- 3.10.1. Para esto puede utilizar la función de agregación COUNT

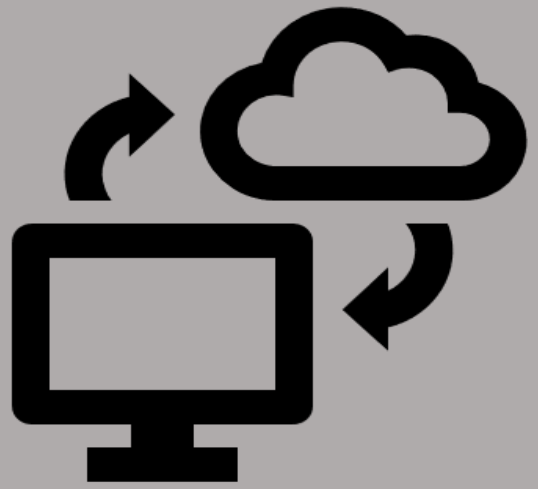
```sql
SELECT COUNT(jg.nombres) FROM jugador AS jg
INNER JOIN equipo AS eq ON jg.id_equipo=eq.id_equipo
WHERE eq.categoria = 'VARONES';
-------------------------------------------------------
```

132 %

Results | Messages

	(No column name)
1	3

FIN DE LA PRESENTACION



GRACIAS POR SU ATENCION.

REDES SOCIALES:

https://www.youtube.com/channel/UC13mhRh9d8ICAq3v4StKv2Q