

# EVALUACION PROCESUAL HITO 2

## BASE DE DATOS I



**PRESENTA:**  
**RONALD URIEL CHOQUE PACO**  
**SIS6972733**

EL ALTO, SETIEMBRE DE 2021

**UNIFRANZ**  
Internacionalízate

**INNOVACIÓN**  
EN EDUCACIÓN

# CONSIGNAS SOBRE LA ELABORACION DE LAS TABLAS.

## UNIFRANZITOS

### Problema

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**.

En tal sentido se deberá crear las siguientes **tablas**.

- **campeonato**
- **equipo**
- **jugador**

Detalle de las tablas.

#### campeonato

**id\_campeonato** => cadena de 12 caracteres y ademas llave primaria  
**nombre\_campeonato** => una cadena de 30 caracteres que no acepta valores nulos  
**sede** => una cadena de 20 caracteres que no acepta valores nulos




#### equipo

**id\_equipo** => cadena de 12 caracteres y ademas llave primaria  
**nombre\_equipo** => una cadena de 30 caracteres, que no acepta valores nulos  
**categoria** => esta columna recibe valores como (varones o mujeres), que no acepta valores nulos  
**id\_campeonato** => llave foreign key relacionado con la tabla campeonato





#### jugador

**id\_jugador** => cadena de 12 caracteres y ademas llave primaria  
**nombres** => una cadena de 30 caracteres, que no acepta valores nulos  
**apellidos** => una cadena de 50 caracteres, que no acepta valores nulos  
**ci** => una cadena de 15 caracteres (ejem: 8997899LP), que no acepta valores nulos  
**edad** => un valor numérico, que no acepta valores nulos  
**id\_equipo** => llave foreign key relacionado con la tabla equipo







#### tabla campeonato

 id_campeonato	 nombre_campeonato	 sede
camp-111	Campeonato Unifranz	El Alto
camp-222	Campeonato Unifranz	Cochabamba

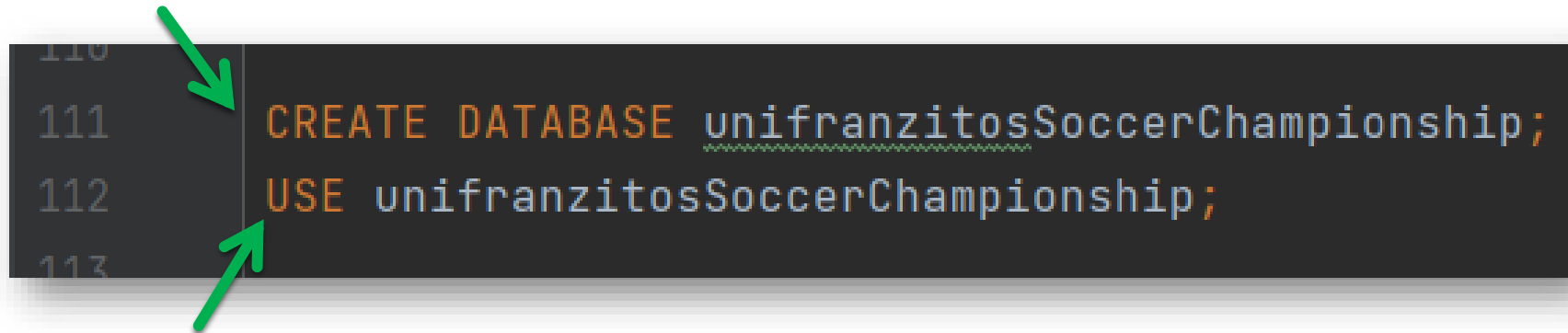
#### tabla equipo

 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

## PASO I: CREAR UNA NUEVA BASE DE DATOS



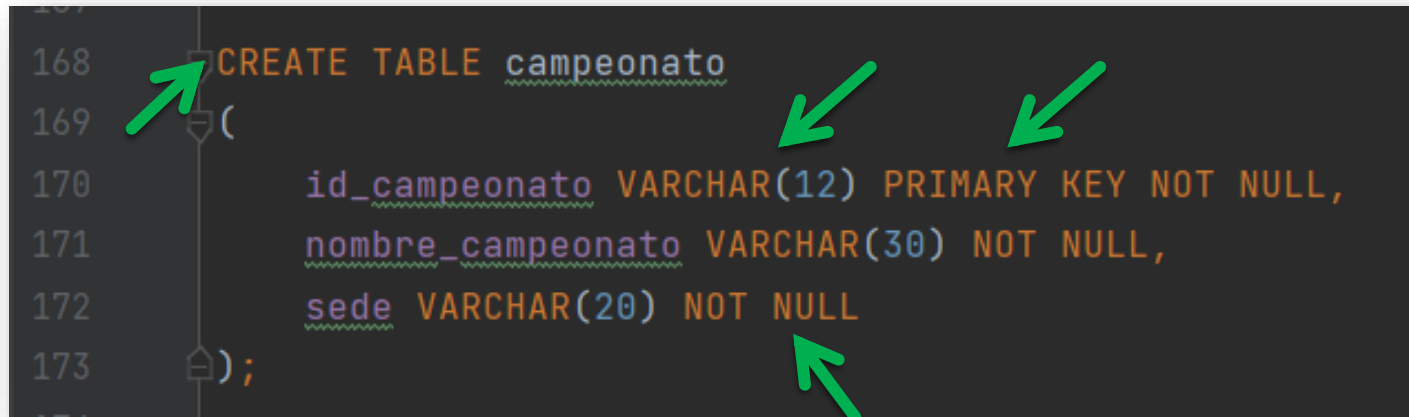
```
110  
111 CREATE DATABASE unifranzitosSoccerChampionship;  
112 USE unifranzitosSoccerChampionship;  
113
```

The image shows a code editor with two SQL commands. The first command, on line 111, is 'CREATE DATABASE unifranzitosSoccerChampionship;', where 'CREATE' and 'DATABASE' are in orange and the database name is underlined in blue. The second command, on line 112, is 'USE unifranzitosSoccerChampionship;', where 'USE' is in orange. Two green arrows point to the words 'CREATE' and 'USE' respectively.

El comando CREATE TABLE permite crear la base de datos, mientras que el comando USE da lugar al uso de la base de datos creada.

## PASO 2: CREAR LA TABLA CAMPEONATO

```
168 CREATE TABLE campeonato
169 (
170     id_campeonato VARCHAR(12) PRIMARY KEY NOT NULL,
171     nombre_campeonato VARCHAR(30) NOT NULL,
172     sede VARCHAR(20) NOT NULL
173 );
```




El comando `CREATE TABLE` se usa para la creación de tablas, dichas tablas deben presentar campos o variables las cuales pueden ser `VARCHAR` (similar a los string en c#) , `CHAR()` o `INTEGER` (variable de tipo entero).

`NOT NULL` es usado para que no se acepte campos vacíos o sin llenar.

`PRIMARY KEY` o llave primaria es para declarar a un campo el cual no puede repetirse y no puede ser nulo; ejemplo: para llenar el carnet en la tabla.

## PASO 2: CREAR LA TABLA EQUIPO


```
147 CREATE TABLE equipo
148 (
149     id_equipo VARCHAR(12) PRIMARY KEY NOT NULL,
150     nombre_equipo VARCHAR(30) NOT NULL,
151     categoria VARCHAR(8) NOT NULL,
152     id_campeonato VARCHAR(12) NOT NULL,
153     FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato)
154 );
```



Se usan los mismos conceptos que la anterior tabla, aumentando ahora el uso del FOREIGN KEY o clave foránea, sirve para relacionar tablas y sus columnas, haciendo referencia mediante REFERENCES a la columna de otra tabla en la tabla que se presenta.

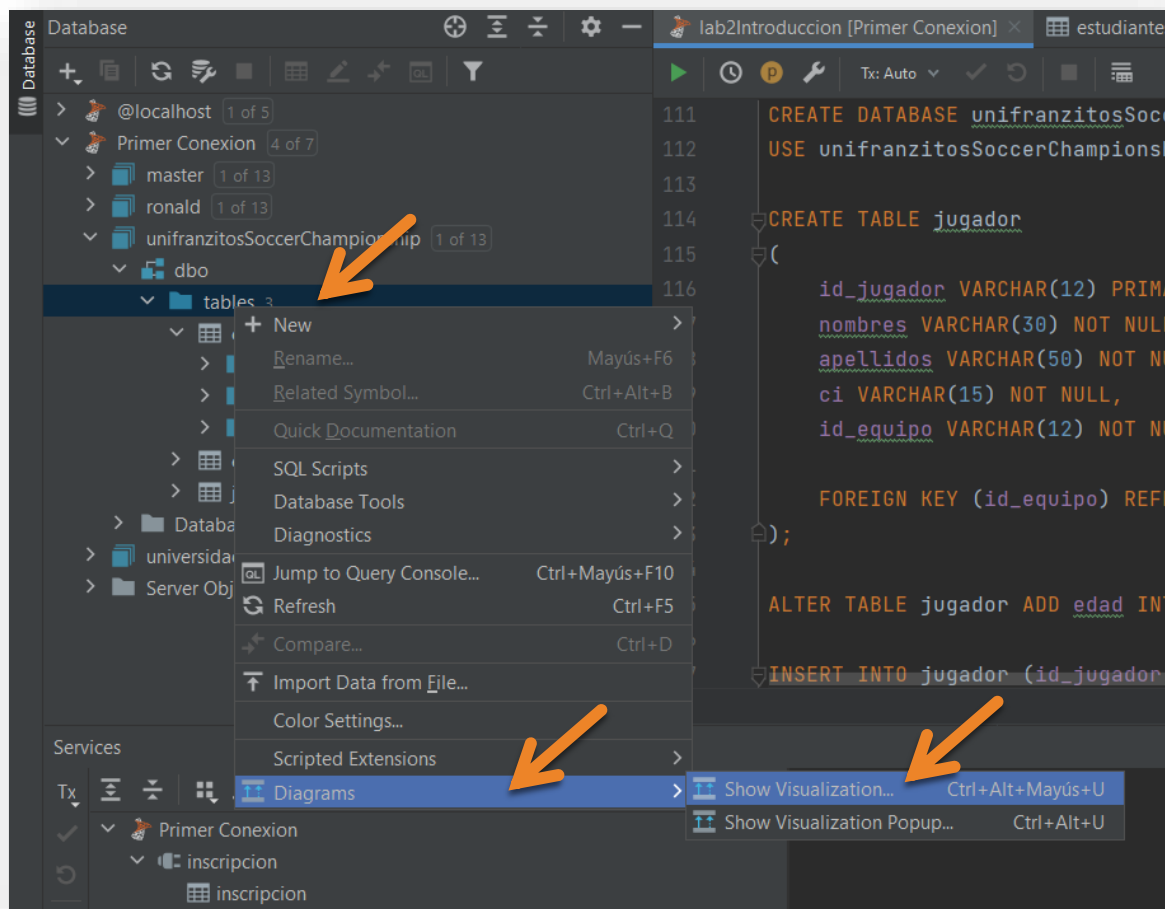
## PASO 2: CREAR LA TABLA JUGADOR

```
113  
114 CREATE TABLE jugador  
115 (  
116     id_jugador VARCHAR(12) PRIMARY KEY NOT NULL,  
117     nombres VARCHAR(30) NOT NULL,  
118     apellidos VARCHAR(50) NOT NULL,  
119     ci VARCHAR(15) NOT NULL,  
120     id_equipo VARCHAR(12) NOT NULL,  
121  
122     FOREIGN KEY (id_equipo) REFERENCES equipo (id_equipo)  
123 );  
124  
125 ALTER TABLE jugador ADD edad INTEGER;  
126
```

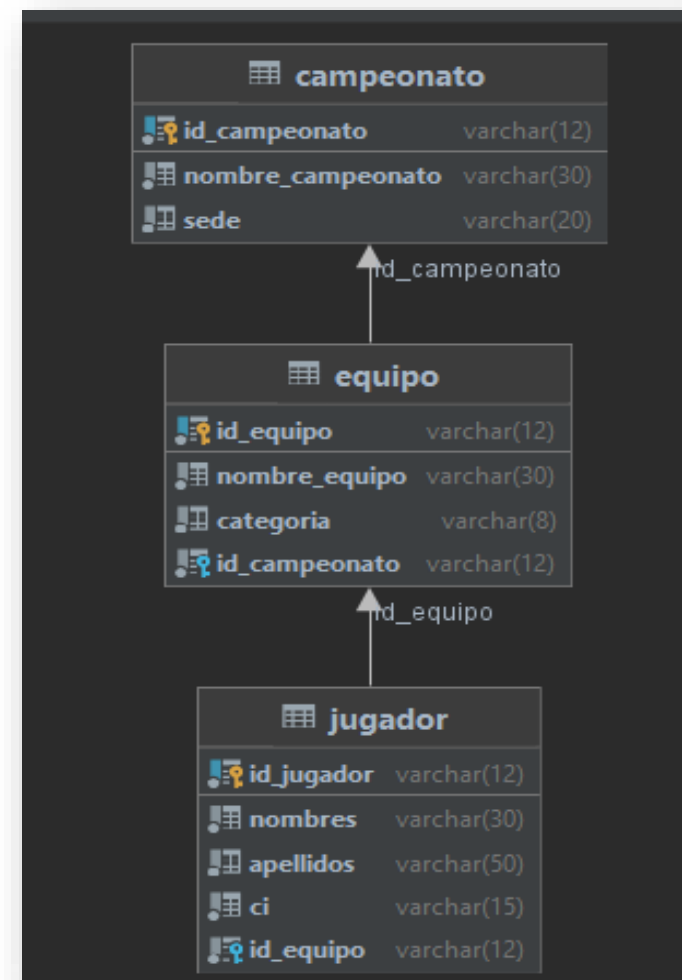


Se hace el uso de los mismos conceptos ya presentados, añadiendo como modo de repaso el uso de `ALTER TABLE` (sirve para alterar una tabla existente), añadiendo un nuevo campo mediante `ADD` y declarando su tipo de variable.

## PASO 5: VISUALIZACION DEL DIAGRAMA DE TABLAS

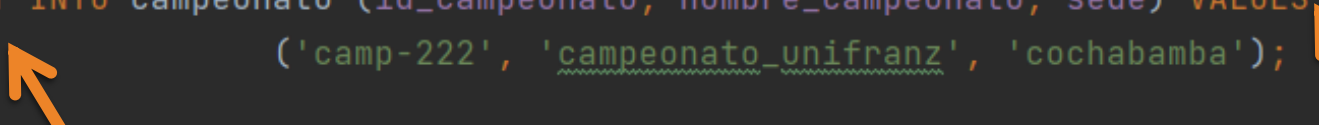


Haciendo un clic derecho sobre la carpeta de tablas, vaya y seleccione **DIAGRAMS** y la opción **SHOW VISUALIZATION** para poder ver el diagrama.



## PASO 6: LLENAR DATOS EN LA TABLA CAMPEONATO

```
172  
173 INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede) VALUES  
174      ('camp-111', 'campeonato_unifranz', 'el alto');  
175 INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede) VALUES  
176      ('camp-222', 'campeonato_unifranz', 'cochabamba');  
177
```



Mediante los comandos INSERT INTO () VALUES; podemos llenar datos a la tabla seleccionada, recordemos que cada dato debe ser llenado ya que se uso el NOT NULL que no acepta campos vacíos o nulos.



## PASO 7: LLENAR DATOS EN LA TABLA EQUIPO

```
156  INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato) VALUES
157      ('eqp-111', 'google', 'varones', 'camp-111');
158  INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato) VALUES
159      ('eqp-222', '404', 'varones', 'camp-111');
160  INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato) VALUES
161      ('eqp-333', 'girls_unifranza', 'mujeres', 'camp-111');
162
```

## PASO 8: LLENAR DATOS EN LA TABLA JUGADOR

```
126
127  INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad ,id_equipo) VALUES
128      ('jug-111', 'carlos', 'villa', '11111112LP', 19, 'eqp-222');
129  INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad ,id_equipo) VALUES
130      ('jug-222', 'pedro', 'salas', '11111117LP', 20, 'eqp-222');
131  INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad ,id_equipo) VALUES
132      ('jug-333', 'saul', 'araj', '11111134LP', 21, 'eqp-222');
133  INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad ,id_equipo) VALUES
134      ('jug-444', 'sandra', 'solis', '11111546LP', 20, 'eqp-333');
135  INSERT INTO jugador (id_jugador, nombres, apellidos, ci, edad ,id_equipo) VALUES
136      ('jug-555', 'ana', 'mica', '11117896LP', 21, 'eqp-333');
137
```

## SECCION: MANEJO DE CONSULTAS

Mostrar que jugadores que formen parte del equipo **equ-222**

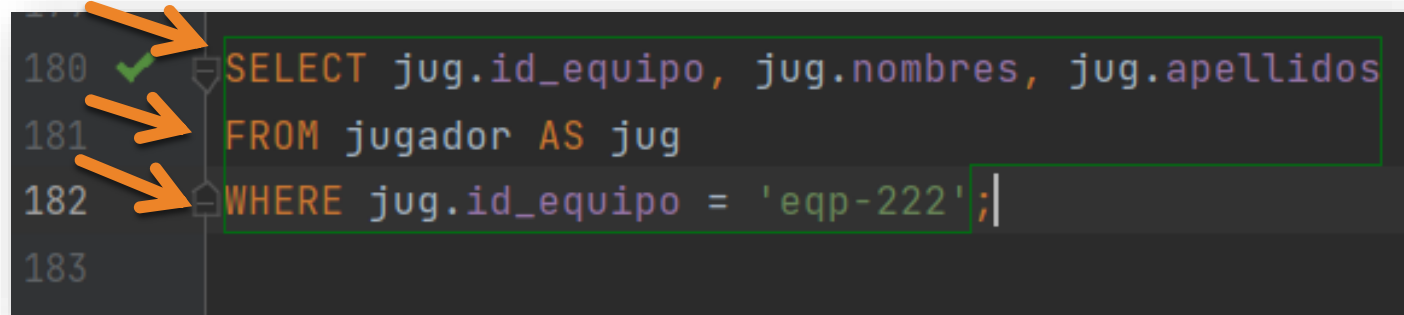
A recordar:

SELECT funciona para seleccionar un campo.

FROM indica de que tabla de extraerá dicha información.

WHERE indica la condición a cumplirse.

El usar AS simboliza una abreviación para una tabla, la cual sirve para no equivocar los datos de las distintas tablas.

A screenshot of a SQL editor showing a query. Three orange arrows point to the keywords 'SELECT', 'FROM', and 'WHERE' in the query. The query is: 

```
180 SELECT jug.id_equipo, jug.nombres, jug.apellidos
181 FROM jugador AS jug
182 WHERE jug.id_equipo = 'eqp-222';
183
```

```
SELECT jug.id_equipo, jug.nombres, jug.apellidos
FROM jugador AS jug
WHERE jug.id_equipo = 'eqp-222';
```

	id_equipo	nombres	apellidos
1	eqp-222	carlos	villa
2	eqp-222	pedro	salas
3	eqp-222	saul	araj

## SECCION: MANEJO DE CONSULTAS

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

```
184 ✓ SELECT jug.id_equipo, jug.nombres, jug.apellidos  
185 FROM jugador AS jug  
186 WHERE jug.id_equipo = 'eqp-333';  
187
```

	id_equipo	nombres	apellidos
1	eqp-333	sandra	solis
2	eqp-333	ana	mica

## SECCION: MANEJO DE CONSULTAS

Mostrar aquellos jugadores mayores o igual a **21** años


```
187  
188 ✓ SELECT jug.edad, jug.nombres, jug.apellidos  
189 FROM jugador AS jug  
190 WHERE edad >= 21  
191
```

	edad	nombres	apellidos
1	21	saul	araj
2	21	ana	mica

## SECCION: MANEJO DE CONSULTAS

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

```
192 ✓ SELECT eqp.id_equipo, eqp.categoria, eqp.nombre_equipo
193 FROM equipo AS eqp
194 WHERE eqp.id_campeonato = 'camp-111' AND
195        eqp.categoria = 'mujeres';
196
```



A recordar que el operador AND/OR (y/o) es usado para dar varias instrucciones.

1 row			Tx: Auto	DDL
	id_equipo	categoria	nombre_equipo	
1	eqp-333	mujeres	girls_unifranza	

## SECCION: MANEJO DE CONSULTAS

Mostrar el nombre del equipo del jugador con id\_jugador igual a **jug-333**

```
196  
197 ✓ SELECT jug.id_jugador, jug.nombres, jug.apellidos, eqp.id_equipo, eqp.nombre_equipo  
198 FROM jugador AS jug, equipo AS eqp  
199 WHERE jug.id_jugador = 'jug-333' AND  
200 eqp.id_equipo = jug.id_equipo;  
201
```

	id_jugador	nombres	apellidos	id_equipo	nombre_equipo
1	jug-333	saul	araj	eqp-222	404

## SECCION: MANEJO DE CONSULTAS

Mostrar el nombre del campeonato del jugador con id\_jugador igual a **jug-333**

```
201
202 ✓ SELECT jug.id_jugador, jug.nombres, jug.apellidos, eqp.id_equipo, eqp.nombre_equipo, camp.nombre_campeonato
203 FROM jugador AS jug, equipo AS eqp, campeonato AS camp
204 WHERE jug.id_jugador = 'jug-333' AND
205        eqp.id_equipo = jug.id_equipo AND
206        eqp.id_campeonato = camp.id_campeonato;
207
```

1 row						
	id_jugador	nombres	apellidos	id_equipo	nombre_equipo	nombre_campeonato
1	jug-333	saul	araj	eqp-222	404	campeonato_unifranz



## SECCION: MANEJO DE CONSULTAS

Determinar cuántos jugadores pertenecen a la categoría **VARONES**

```
207  
208 ✓ SELECT eqp.id_equipo, jug.nombres, jug.apellidos, eqp.categoria, eqp.nombre_equipo  
209 FROM equipo AS eqp, jugador AS jug  
210 WHERE eqp.categoria = 'varones' AND  
211 eqp.id_equipo = jug.id_equipo;  
212
```

	id_equipo	nombres	apellidos	categoria	nombre_equipo
1	eqp-222	carlos	villa	varones	404
2	eqp-222	pedro	salas	varones	404
3	eqp-222	saul	araj	varones	404

# SECCION: MANEJO DE CONCEPTOS.

- **Que es DDL.**

DDL es DATA DEFINITION LANGUAGE, todo lo relacionado con el diseño y creaciones de tablas.

- **Que es DML.**

DML es DATA MANIPULATION LANGUAGE, todo lo relacionado a la manipulación de tablas.

- **Que son Bases de Datos relacionales y no relacionales.**

**Relacional:** funciona a base de SQL manejando tablas bidimensionales (con filas y columnas), sus campos de relación deben ser del mismo tipo.

**No Relacional:** SQL no es su principal lenguaje, este tipo de base esta dedicada al rendimiento gracias a que puede almacenar grandes cantidades de datos, los cuales se van expandiendo.



## SECCION: MANEJO DE CONCEPTOS.

- **Qué es SQL.**

Es un lenguaje que facilita la comunicacion con bases de datos relacionales dando acceso a la manipulacion de los mismos.

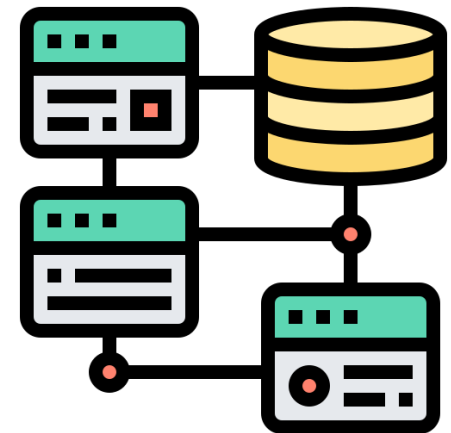
- **Cuando debería de usarse una Base de Datos relacional y no relacional.**

CUANDO USAR: cuando existen pocos datos a añadirse, algo controlable; cuando no se corre el riesgo de uso excesivo de usuarios al mismo tiempo.

CUANDO NO USAR: cuando exista exceso de demanda para proveer el servicio; cuando los datos son propensos a crecer rápido en momentos puntuales.

- **Qué tipo de base de datos es SQL Server.**

Es una base de datos relacional.



## SECCION: MANEJO DE CONCEPTOS.

- Mencione algunos métodos de autenticación para acceder a una instancia **SQL Server**.

Tenemos a: Windows Authentication (usuario SA) y SQL Server Authentication (usuario propio) entre los mas usados.

- Qué tipo de licencia tiene una base de datos **SQL Server**.

Posee licencia Microsoft EULA (información técnica).

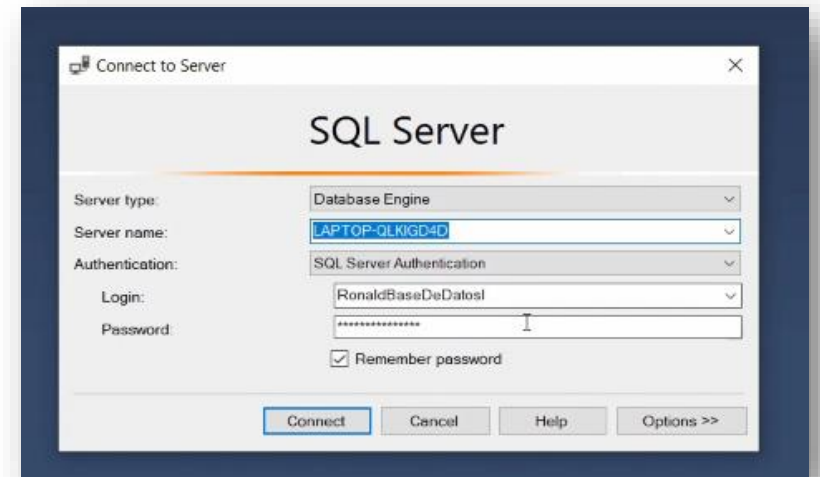
- Que es una tabla.

Son objetos que contienen los datos, organizándose mediante filas y columnas.

- Que significa **PRIMARY KEY** y **FOREIGN KEY**.

**PRIMARY KEY** (clave primaria) es para declarar a un campo el cual no puede repetirse y no puede ser nulo; ejemplo: para llenar el carnet en la tabla.

**FOREIGN KEY** (clave foránea) sirve para relacionar tablas y sus columnas, haciendo referencia a la columna de otra tabla en la tabla que se presenta.



# EVALUACION PROCESUAL HITO 2

## BASE DE DATOS I



**¡GRACIAS POR  
SU ATENCION!**

Ronald.choque2111@gmail.com

**SIS6972733**

EL ALTO, SETIEMBRE DE 2021

**UNIFRANZ**  
Internacionalízate

**INNOVACIÓN  
EN EDUCACIÓN**