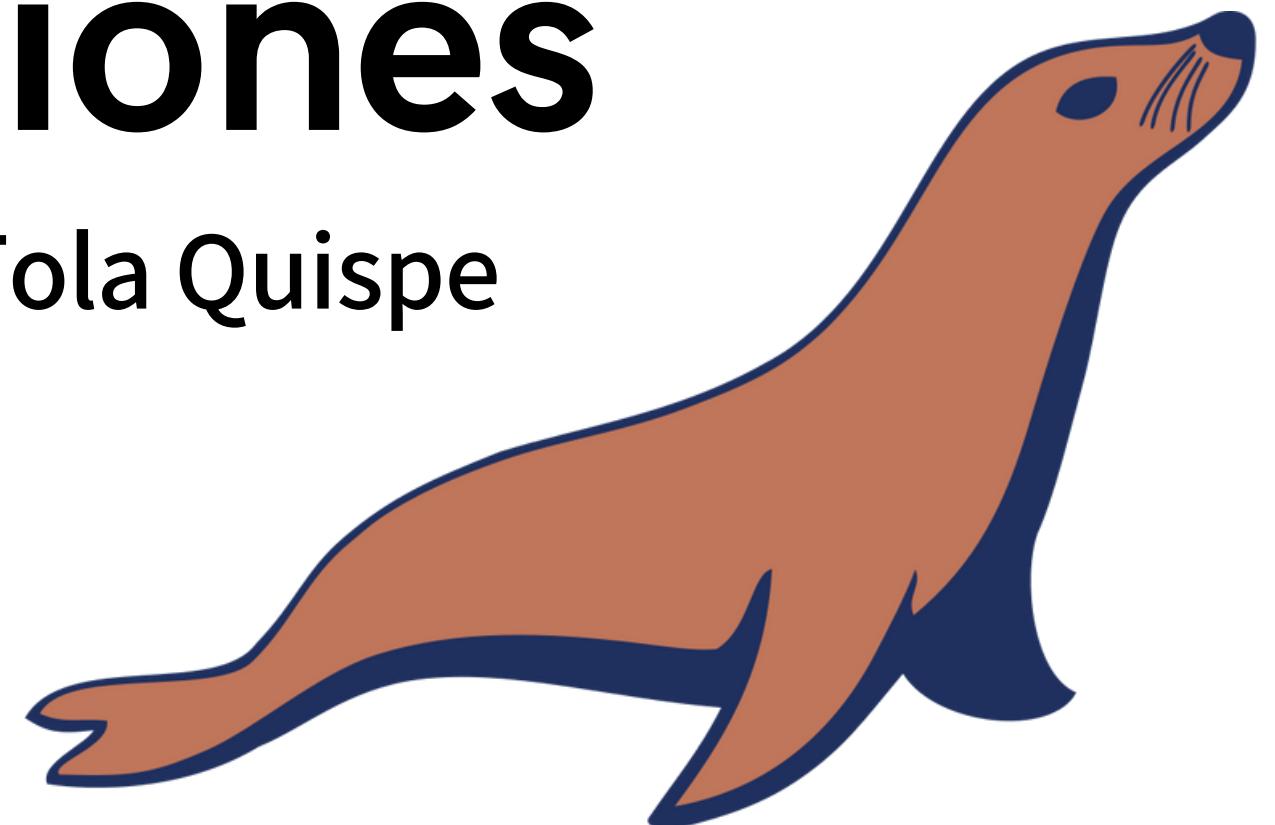


Base de datos II

Manejo de base de datos relacionales y funciones

Romario Tola Quispe



MariaDB

01 ¿A que se refiere cuando se habla de bases de datos relacionales?

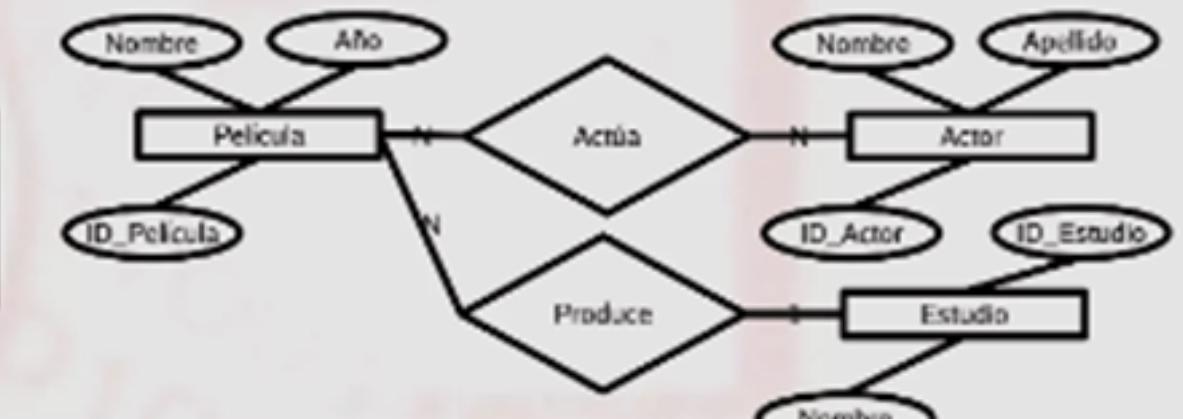
En un sistema de base de datos relacional, la información se almacena en tablas, donde cada tabla representa un tipo de entidad o relación. Cada tabla está compuesta por filas y columnas, donde las filas representan las instancias o registros individuales y las columnas representan los atributos o campos de datos.

Estudio	
ID_Estudio	Nombre
1	Ghibli
2	New Line Cinema
3	Lucastfilms
4	Sogecine

Actor		
ID_Actor	Nombre	Apellido
1	Mark	Hamill
2	Cristopher	Lee
3	Javier	Bardem
4	Hugo	Weaving

Actúa	
ID_Actor	ID_Película
1	1
2	2
3	3
4	2

Película			
ID_Película	Nombre	Año	ID_Estudio
1	La guerra de las galaxias	1977	3
2	El señor de los anillos 1	2001	2
3	Mar Adentro	2004	4
4	El viaje de Chihiro	2001	1



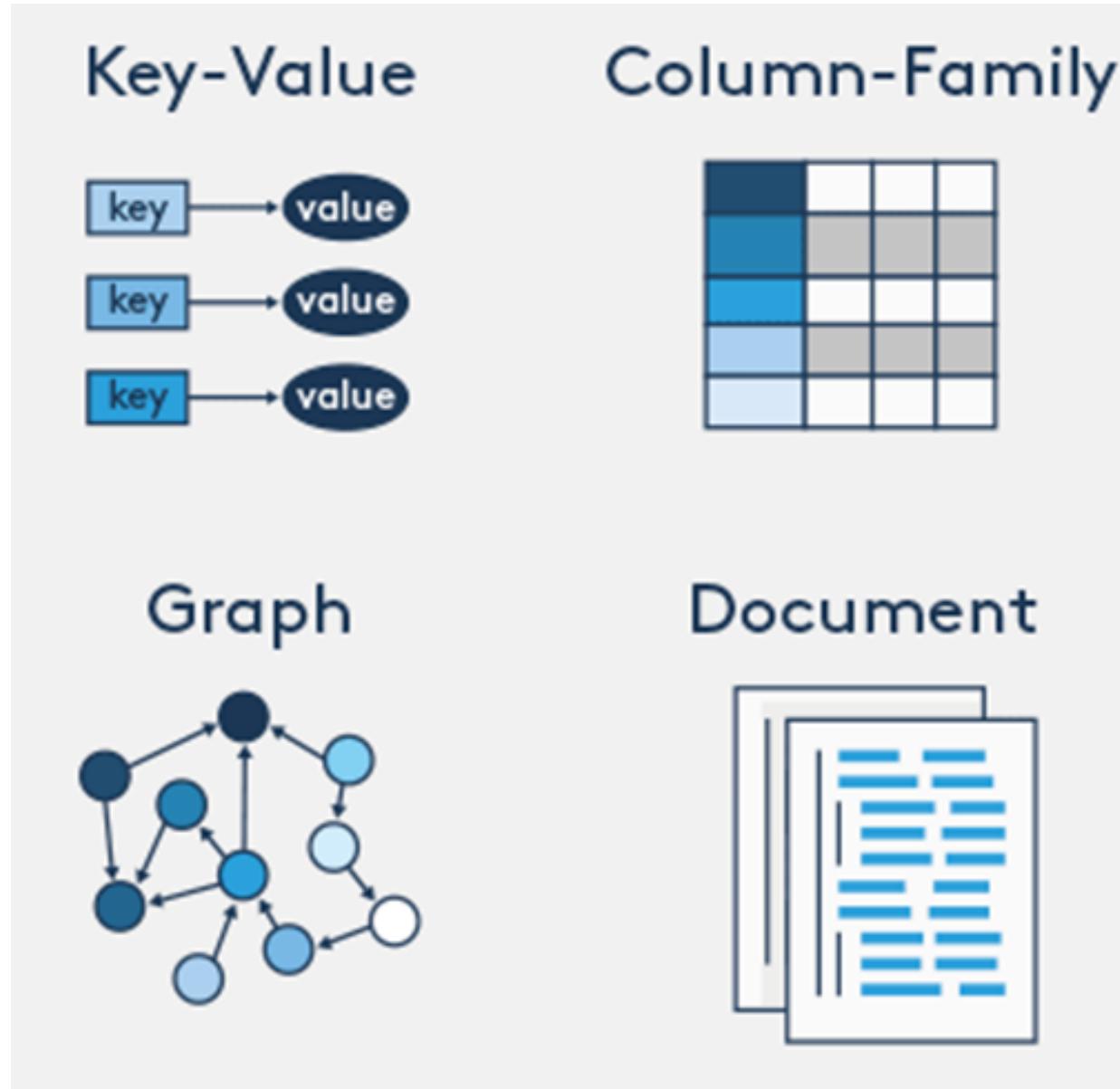
Ventajas de una base de datos relacional:

Facilidad de uso: el modelo relacional es intuitivo y fácil de entender, lo que hace que sea fácil diseñar y utilizar una base de datos relacional.

Escalabilidad: las bases de datos relacionales pueden crecer a medida que se agregan más datos y usuarios sin comprometer el rendimiento.

Seguridad: las bases de datos relacionales ofrecen un alto nivel de seguridad y control de acceso, lo que ayuda a proteger los datos de accesos no autorizados.

02 ¿A que se refiere cuando se habla de bases de datos no relacionales?



Las bases de datos no relacionales pueden almacenar datos en varios formatos, incluyendo documentos, gráficos, etc. Además, las bases de datos no relacionales pueden manejar grandes cantidades de datos como los datos de redes sociales y los datos generados por dispositivos de Internet de las cosas.

Ventajas de una base de datos no relacional:

Velocidad: Están diseñadas para manejar grandes cantidades de datos en tiempo real, lo que las hace ideales para aplicaciones de alto rendimiento como el procesamiento de transacciones financieras o el análisis de datos en tiempo real.

Escalabilidad horizontal: Pueden escalar horizontalmente agregando más servidores y distribuyendo la carga de trabajo entre ellos, lo que las hace ideales para aplicaciones de alta disponibilidad y rendimiento.

Flexibilidad de datos: Permiten almacenar datos de diferentes formatos y tamaños sin necesidad de establecer una estructura fija para ellos.



03 ¿Qué es MySQL y MariaDB?.

MySQL es un sistema de gestión de bases de datos relacional.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Es desarrollado por Michael Widenius —fundador de MySQL

04

¿Qué son las funciones de agregación?

Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos. Es decir, nos permiten obtener medias, máximos, etc...

Los mas usados son:

COUNT: devuelve el número de filas en una tabla o el número de valores distintos

SUM: devuelve la suma de los valores en una columna numérica.

AVG: devuelve el valor promedio

MAX: devuelve el valor máximo

MIN: devuelve el valor mínimo



05 ¿Qué llegaría a ser XAMPP, WAMP SERVER o LAMP?

XAMPP:

XAMPP es una distribución de Apache que incluye varios softwares libres. El paquete incluye una interfaz de usuario gráfica que facilita la gestión de los diferentes componentes del software, y permite crear y configurar servidores web y bases de datos de manera rápida y sencilla.



WAMP SERVER:

WAMP SERVER es un paquete de software libre que se utiliza para crear sitios web dinámicos en sistemas operativos Windows. Los componentes de WAMP Server incluyen el servidor web Apache, el lenguaje de programación PHP y el sistema de gestión de bases de datos MySQL (o MariaDB).



LAMP:

LAMP es un acrónimo que se refiere a un conjunto de software libre utilizado para crear sitios web dinámicos en Linux. Los componentes de LAMP incluyen el sistema operativo Linux, el lenguaje de programación PHP y el sistema de gestión de bases de datos MySQL (o MariaDB).



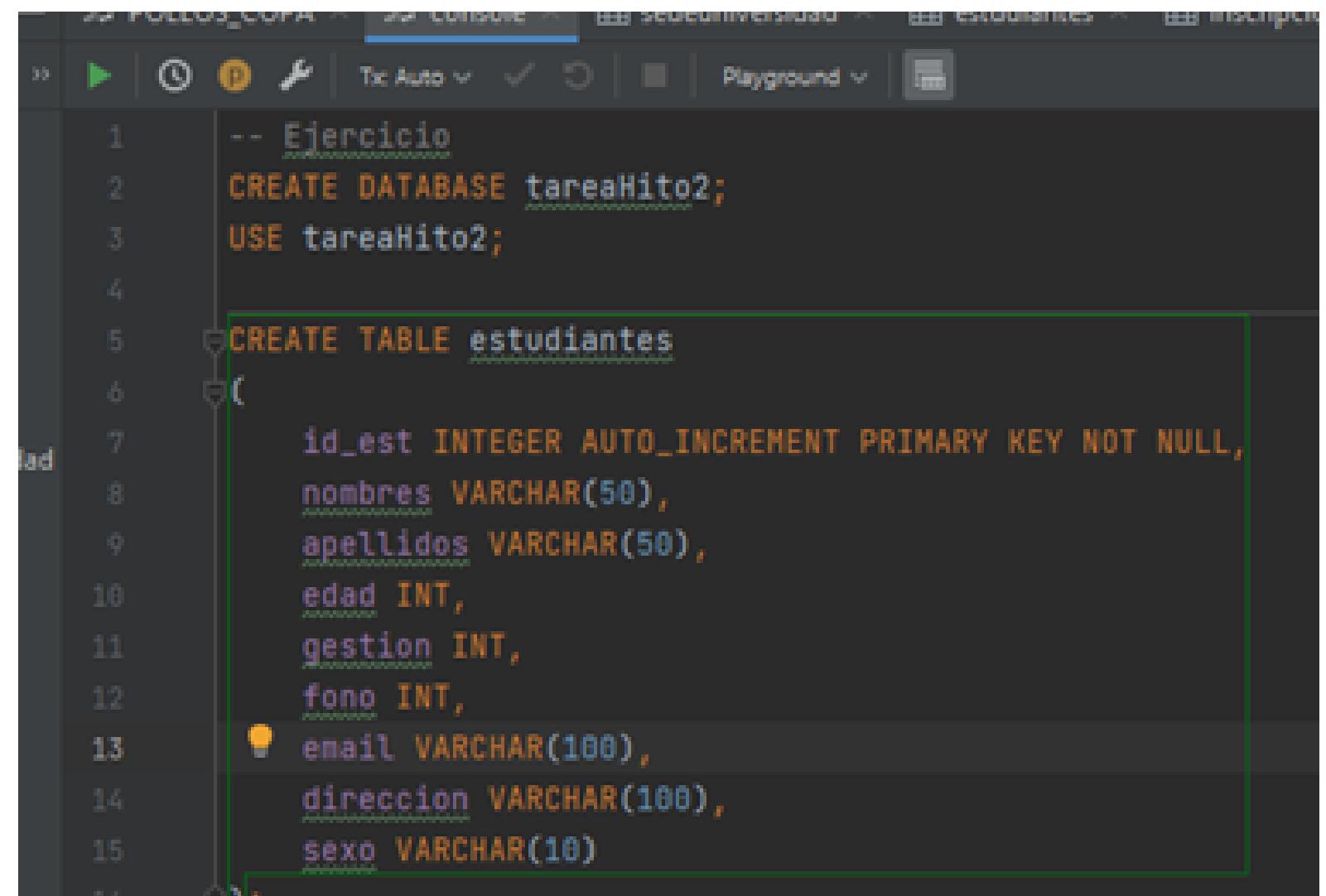
A close-up photograph showing the top right corner of a laptop keyboard and a small potted succulent plant. The keyboard keys visible include 'caps lock', 'shift', 'control', 'fn', 'option', 'command', and 'z'. The succulent has several green, pointed leaves and is in a white rectangular pot.

¿Cual es la diferencia entre las funciones de agregación y funciones creados por el DBA?

La principal diferencia entre las funciones de agregación y las funciones creadas por el usuario en SQL es que las funciones de agregación son funciones predefinidas que se utilizan comúnmente para resumir datos, mientras que las funciones creadas por el usuario son específicas para el conjunto de datos y las operaciones que se necesiten realizar.

¿Para qué sirve el comando USE?

El comando USE DATABASE se utiliza para designar una base externa como base de datos actual, en otras palabras, la base a la cual se dirigirán las próximas consultas SQL en el proceso actual.



```
-- Ejercicio
CREATE DATABASE tareaHito2;
USE tareaHito2;

CREATE TABLE estudiantes
(
    id_est INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
    nombres VARCHAR(50),
    apellidos VARCHAR(50),
    edad INT,
    gestion INT,
    fono INT,
    email VARCHAR(100),
    direccion VARCHAR(100),
    sexo VARCHAR(10)
```

08 ¿Qué es DML y DDL?

DDL: (Data Definition Language) es un conjunto de comandos SQL para definir el esquema de la base de datos.

- Create
- alter
- drop
- truncate

DML:(Data Manipulation Language) es un conjunto de comandos SQL utilizados para manipular los datos dentro del esquema creado por DDL.

- Select
- insert
- update
- delete

09 ¿Qué características tiene una función?

- Nombre de la función
- Parámetros de entrada
- Tipo de retorno
- Cuerpo de la función
- Sentencia RETURN

```
154
155          CREATE FUNCTION sumar(numero INT, numero2 INT)
156          RETURNS INT
157          BEGIN
158              DECLARE resultado INT;
159              SET resultado = numero + numero2;
160              RETURN resultado;
161          END
162
```

¿Cómo crear, modificar y cómo eliminar una función?

```
155  
156      CREATE FUNCTION sumar(numero INT, numero2 INT)  
157          RETURNS INT  
158          BEGIN  
159              RETURN numero + numero2;  
160          END  
161
```

```
2  
3      ALTER FUNCTION sumar(numero INT, numero2 INT)  
4          RETURNS INT  
5          BEGIN  
6              RETURN numero * numero2;  
7          END
```

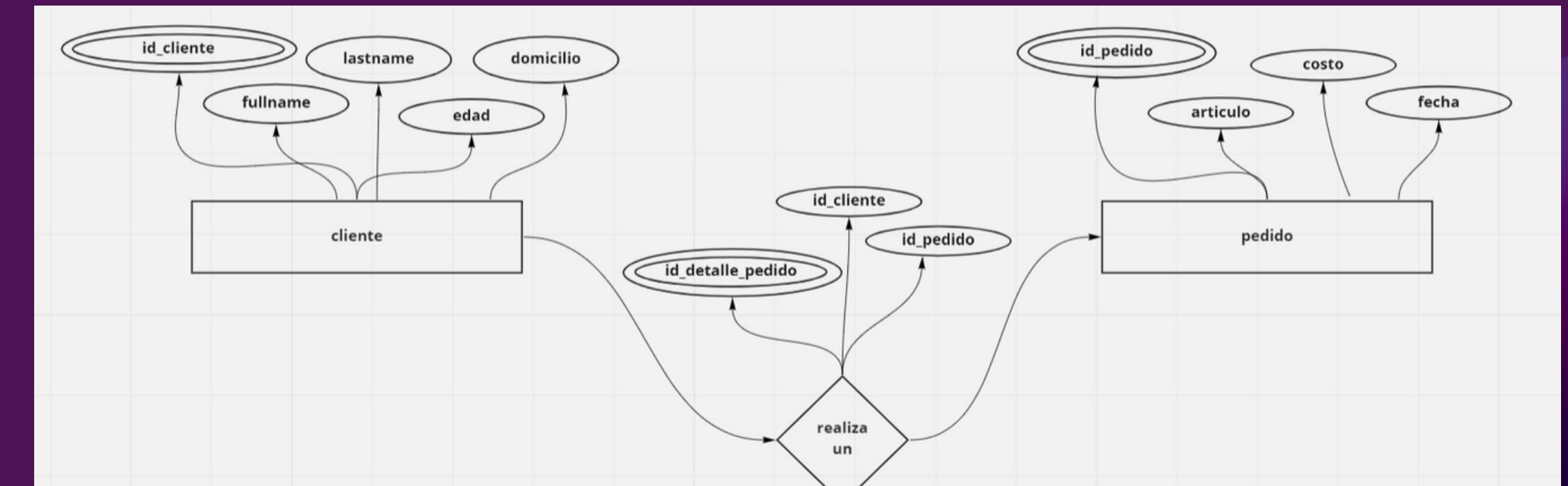
```
DROP FUNCTION sumar;
```

Parte practica

11

Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER.

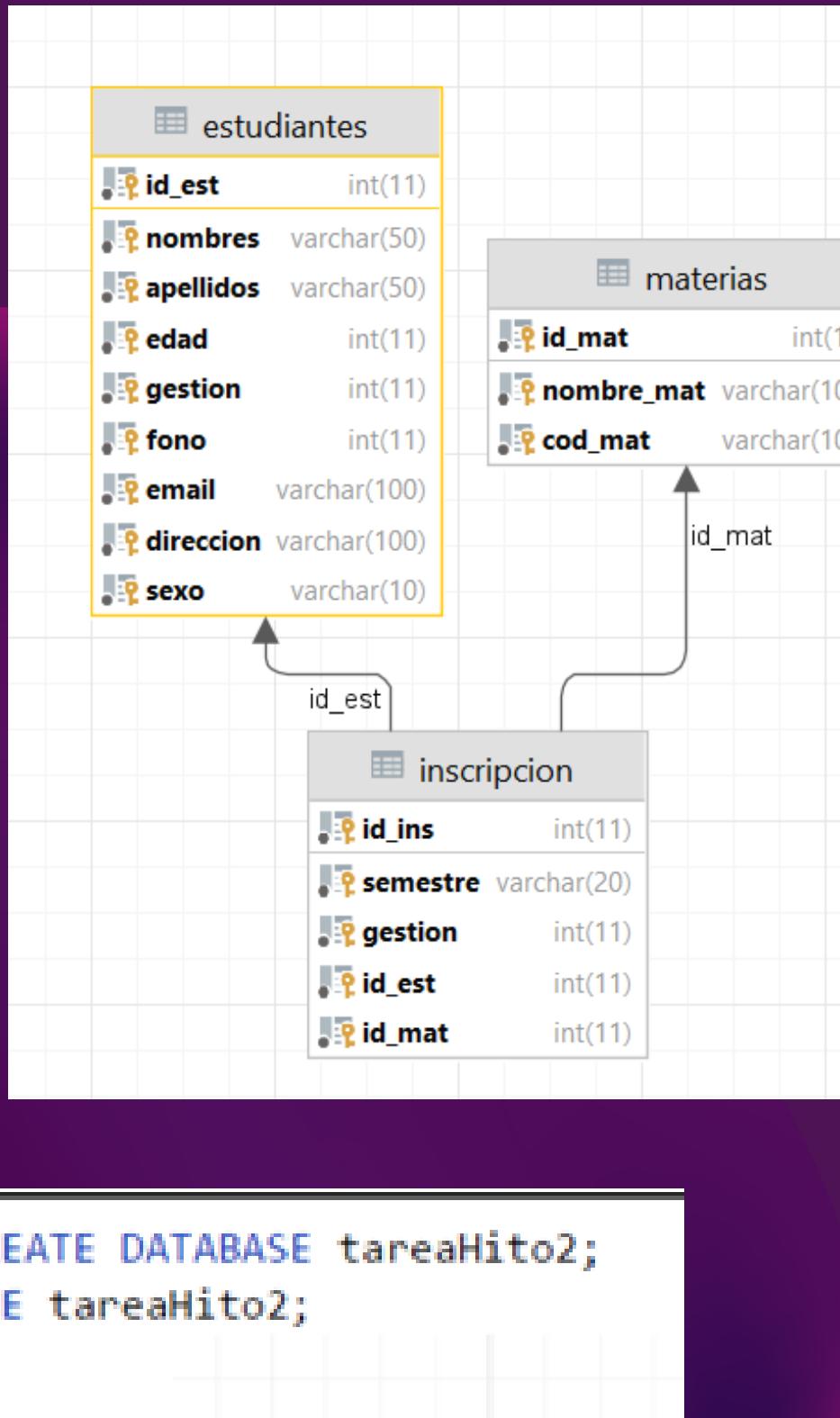
- Se sugiere crear una base de datos de nombre POLLOS_COPA y en ella crear las tablas:
 - cliente
 - detalle_pedido
 - pedido



Crear una consulta SQL en base al ejercicio anterior.

- Debe de utilizar las 3 tablas creadas anteriormente.
- Para relacionar las tablas utilizar JOINS.
- Adjuntar el código SQL generado.

13 Crear un función que compare dos códigos de materia.



```
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
```

```
VALUES ('Miguel', 'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino'), ('Sandra', 'Mavir Urias', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino'), ('Joel', 'Adubiri Mondar', 30, 2832117, 'joel@gmail.com',
```

```
'Av. 6 de Agosto', 'masculino'), ('Andrea', 'Arias Ballesteros', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino'), ('Santos', 'Montes Valenzuela', 24, 2832119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

```
INSERT INTO materias (nombre_mat, cod_mat)
```

```
VALUES ('Introducción a la Arquitectura', 'ARQ-101'), ('Urbanismo y Diseño', 'ARQ-102'), ('Dibujo y Pintura Arquitectónico', 'ARQ-103'), ('Matemática discreta', 'ARQ-104'), ('Física Básica', 'ARQ-105');
```

```
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
```

```
VALUES (1, 1, '1er Semestre', 2018), (1, 2, '2do Semestre', 2018), (2, 4, '1er Semestre', 2019), (2, 3, '2do Semestre', 2019), (3, 3, '2do Semestre', 2020), (3, 1, '3er Semestre', 2020), (4, 4, '4to Semestre', 2021), (5, 5, '5to Semestre', 2021);
```

Resolver lo siguiente:

■ Mostrar los nombres y apellidos de los estudiantes inscritos en la materia ARQ-105, adicionalmente mostrar el nombre de la materia.

■ Deberá de crear una función que reciba dos parámetros y esta función deberá ser utilizada en la cláusula WHERE.

```
CREATE DATABASE tareaHito2;
USE tareaHito2;
```

Crear una función que permita obtener el promedio de las edades del género masculino o femenino de los estudiantes inscritos en la asignatura ARQ-104.

- La función recibe como parámetro el género y el código de materia.

Crear una función que permita concatenar 3 cadenas.

- La función recibe 3 parámetros.
- Si las cadenas fuesen:
 - Pepito
 - Pep
 - 50
- La salida debería ser: (Pepito), (Pep), (50)
- La función creada utilizarlo en una consulta SQL.
 - Es decir podría mostrar el nombre, apellidos y la edad de los estudiantes.

Crear la siguiente VISTA:

- La vista deberá llamarse ARQUITECTURA_DIA_LIBRE
- El dia viernes tendrán libre los estudiantes de la carrera de ARQUITECTURA debido a su aniversario
 - Este permiso es solo para aquellos estudiantes inscritos en el año 2021.
 - La vista deberá tener los siguientes campos.
 1. Nombres y apellidos concatenados = FULLNAME
 2. La edad del estudiante = EDAD
 3. El año de inscripción = GESTION
 4. Generar una columna de nombre DIA_LIBRE
 - a. Si tiene libre mostrar LIBRE
 - b. Caso contrario mostrar NO LIBRE

Crear la siguiente VISTA:

- Agregar una tabla cualquiera al modelo de base de datos.
- Después generar una vista que maneje las 4 tablas
- La vista deberá llamarse PARALELO_DBA_I

GRACIAS !!