BASE DE DATOS 2

ESTUDIANTE: ISRAEL ALEJANDRO ZAMBRANA

DOCENTE: LIC WILLIAM BARRA PAREDES

SEMESTRE: TERCER SEMESTRE

Manejo de conceptos.

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

R. Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

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

R. Las bases de datos no relacionales son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas. Esto no significa que no puedan usar el lenguaje SQL, pero no lo hacen como herramienta de consulta, sino como apoyo.

3. ¿Qué es MySQL y MariaDB?. Explique si existen diferencias o son iguales, etc.

R. Aunque MariaDB es una bifurcación de MySQL, estos dos sistemas de gestión de bases de datos siguen siendo bastante diferentes: MariaDB tiene licencia GPL mientras que MySQL tiene un enfoque de doble licencia. Cada mango se acumula de una manera diferente. MariaDB soporta muchos motores de almacenamiento diferentes.

4. ¿Qué son las funciones de agregación?

R. Una función de agregación es una función en la que los valores de varias filas se agrupan para formar un único valor de resumen. Las funciones agregadas comunes incluyen: Promedio Contar Máximo Mediana Mínimo Modo Rango Suma etc.

5. ¿Qué llegaría a ser XAMPP?

R. Es un programa capas de controlar otros programas diferentes como mysql, apache etc.

6. ¿Cual es la diferencia entre las funciones de agresión yfunciones creados por el DBA? Es decir funciones creadas por el usuario.

R. cada platafoma de dba tiene sus respectivos comandos para las fucions de agregación como por defento y agregados por el usuario.

7. ¿Para qué sirve el comando USE?

R. Hacer **que** una base **de** datos determinada sea la actual mediante el uso **de** la sentencia **USE** no descarta **que** se pueda acceder a tablas **de** otras bases **de** datos.

8. Que es DML y DDL?

R. Las sentencias SQL se dividen en dos categorías: lenguaje de definición de datos (**DDL**) y lenguaje de manipulación de datos (**DML**). Las sentencias **DDL** se utilizan para describir una base de datos, para definir su estructura, para crear sus objetos y para crear los subobjetos de la tabla.

9. ¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc.

R. Crea una función definida por el usuario. Una función definida por el usuario es una rutina de Transact-SQL o Common Language Runtime (CLR) que acepta parámetros, realiza una acción, como un cálculo complejo, y devuelve el resultado de esa acción como un valor. El valor devuelto puede ser un valor escalar (único) o una tabla. Utilice esta instrucción para crear una rutina reutilizable que se pueda utilizar de estas formas:

En instrucciones Transact-SQL como SELECT

En las aplicaciones que llaman a la función

En la definición de otra función definida por el usuario

Para parametrizar una vista o mejorar la funcionalidad de una vista indizada

Para definir una columna en una tabla

Para definir una restricción CHECK en una columna

Para reemplazar un procedimiento almacenado

Usar una función insertada como predicado de filtro de la directiva de seguridad

10.¿Cómo crear,modificar ycómo eliminar una función?

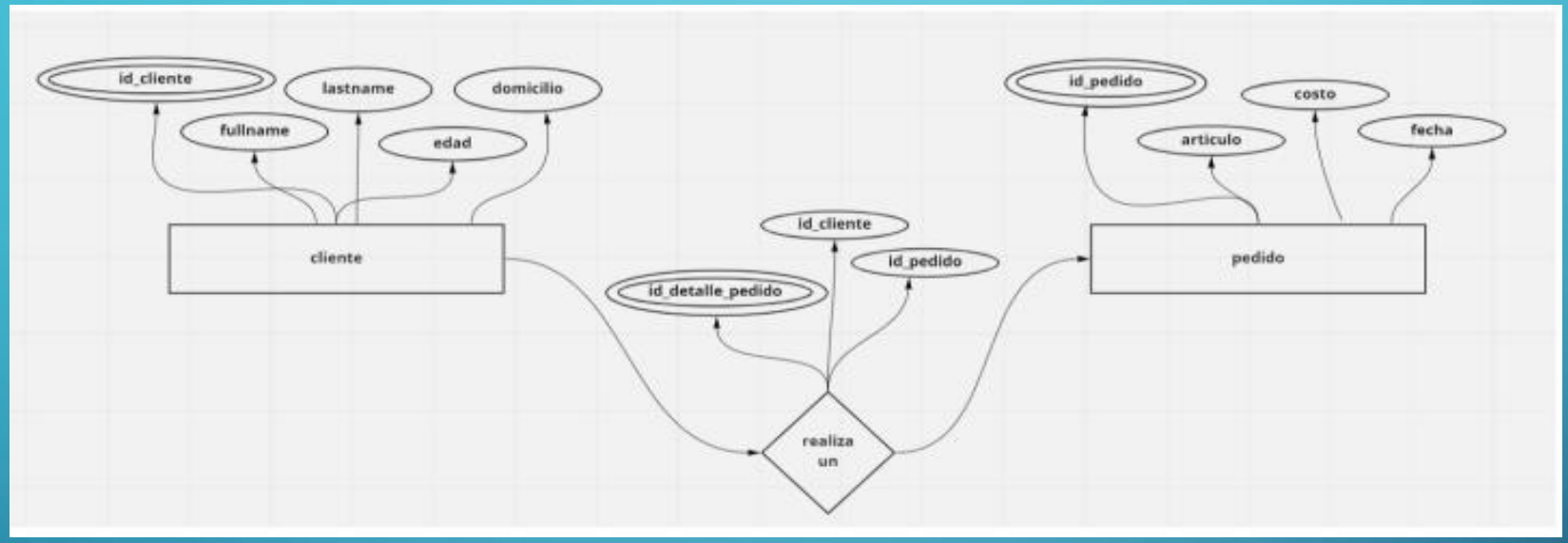
R. CREATE Para crear

Drop para eliminar

Alter funtion para modificar

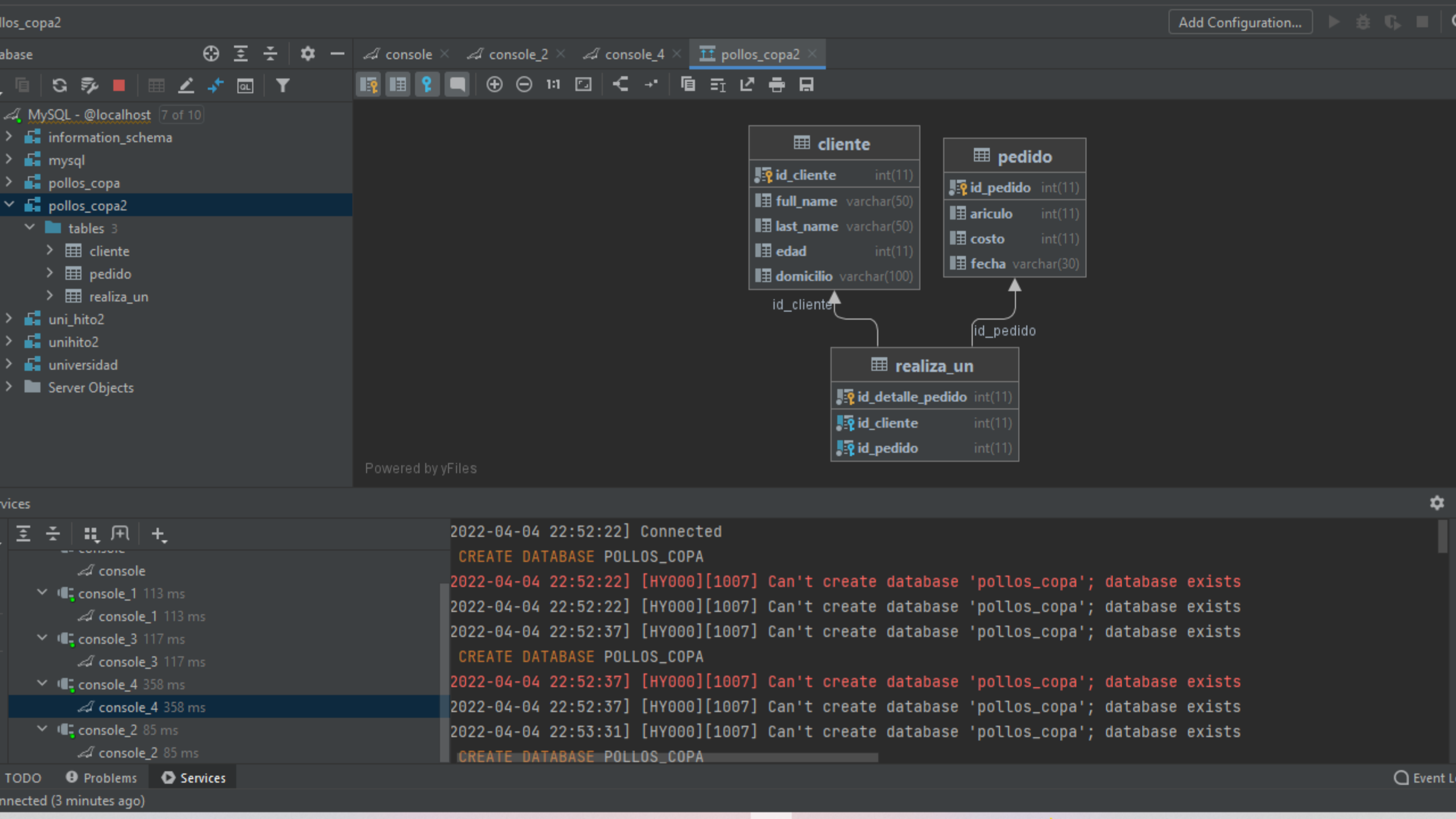
Parte practica

11. Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER.
○ Diseno. ○ Se sugiere crear una base de datos de nombre POLLOS_COPA y en ella crear las tablas:



- cliente
- detalle_pedido
- pedido ○ Adjuntar el código SQL generado. 4 12.Crear una consulta SQL en base al ejercicio ant

```
CREATE DATABASE POLLOS_COPA2;
USE POLLOS_COPA2;
CREATE TABLE realiza_un
(
 id_detalle_pedido INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
 id_cliente INT NOT NULL,
 id_pedido INT NOT NULL,
 FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente),
 FOREIGN KEY (id_pedido) REFERENCES pedido (id_pedido)
);
CREATE TABLE cliente
(
 id_cliente INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
 full_name VARCHAR(50),
 last_name VARCHAR(50),
 edad INTEGER,
 domicilio VARCHAR(100)
);
CREATE TABLE pedido
(
 id_pedido INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL ,
 ariculo INTEGER,
 costo INTEGER,
 fecha VARCHAR(30)
);
```


llos_copa2
Add Configuration...
abase
console
console_2
console_4
pollos_copa2
MySQL - @localhost 7 of 10
information_schema
mysql
pollos_copa
pollos_copa2
tables 3
cliente
pedido
realiza_un
uni_hito2
unihito2
universidad
Server Objects
cliente
id_cliente int(11)
full_name varchar(50)
last_name varchar(50)
edad int(11)
domicilio varchar(100)
pedido
id_pedido int(11)
ariculo int(11)
costo int(11)
fecha varchar(30)
id_cliente
id_pedido
realiza_un
id_detalle_pedido int(11)
id_cliente int(11)
id_pedido int(11)
Powered by yFiles
vices
console
console_1 113 ms
console_1 113 ms
console_3 117 ms
console_3 117 ms
console_4 358 ms
console_4 358 ms
console_2 85 ms
console_2 85 ms
2022-04-04 22:52:22] Connected
CREATE DATABASE POLLOS_COPA
2022-04-04 22:52:22] [HY000][1007] Can't create database 'pollos_copa'; database exists
2022-04-04 22:52:22] [HY000][1007] Can't create database 'pollos_copa'; database exists
2022-04-04 22:52:37] [HY000][1007] Can't create database 'pollos_copa'; database exists
CREATE DATABASE POLLOS_COPA
2022-04-04 22:52:37] [HY000][1007] Can't create database 'pollos_copa'; database exists
2022-04-04 22:52:37] [HY000][1007] Can't create database 'pollos_copa'; database exists
2022-04-04 22:53:31] [HY000][1007] Can't create database 'pollos_copa'; database exists
CREATE DATABASE POLLOS_COPA
TODO
Problems
Services
Event L
nnected (3 minutes ago)

12.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.

```
CREATE TABLE estudiante
(
id_est INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombres VARCHAR(50),
apellidos VARCHAR(50),
edad INTEGER,
gestion INTEGER,
fono INTEGER,
email VARCHAR(100),
direccion VARCHAR(100),
sexo VARCHAR(10)
);
CREATE TABLE inscripcion
(
id_ins INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
semestre VARCHAR(20),
gestion INTEGER,
id_est INT NOT NULL,
id_mat INT NOT NULL,
 FOREIGN KEY (id_est) REFERENCES estudiante (id_est),
 FOREIGN KEY (id_mat) REFERENCES materias (id_mat)
);
CREATE TABLE materias
(
 id_mat INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombre_mat VARCHAR(100),
cod_mat VARCHAR(100)
);
```

```
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Miguel', 'Gonzales Veliz' , 20, 2832115,'miguel@gmail.com' , 'Av. 6 de Agosto' , 'masculino' );
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Sandra', 'Mavir Uria' , 25, 2832116, 'sandra@gmail.com' ,'Av. 6 de Agosto' , 'femenino');
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Joel', 'Adubiri Mondar' , 30, 2832117, 'joel@gmail.com' ,'Av. 6 de Agosto' , 'masculino' );
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Andrea', 'Arias Ballesteros' , 21, 2832118,'andrea@gmail.com' , 'Av. 6 de Agosto' ,
'femenino');
INSERT INTO estudiante ( nombres, apellidos, edad, fono, email,direccion, sexo)
VALUES ('Santos', 'Montes Valenzuela' , 24, 2832119,'santos@gmail.com' , 'Av. 6 de Agosto' ,
'masculino' );
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Introduccion a la Arquitectura' , 'ARQ-101');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Urbanismo y Diseno' , 'ARQ-102');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Dibujo y Pintura Arquitectonico' , 'ARQ-103');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Matematica discreta' , 'ARQ-104');
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Fisica Basica' , 'ARQ-105');
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES (1, 1, '1er Semestre', 2018);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(1, 2, '2do Semestre', 2018);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(2, 4, '1er Semestre', 2019);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(2, 3, '2do Semestre', 2019);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(3, 3, '2do Semestre', 2020);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(3, 1, '3er Semestre', 2020);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(4, 4, '4to Semestre', 2021);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES(5, 5, '5to Semestre', 2021);
```

File Edit View Navigate Code Refactor Run Tools Git Window Help Base de Datos II (2022) - tareahito2

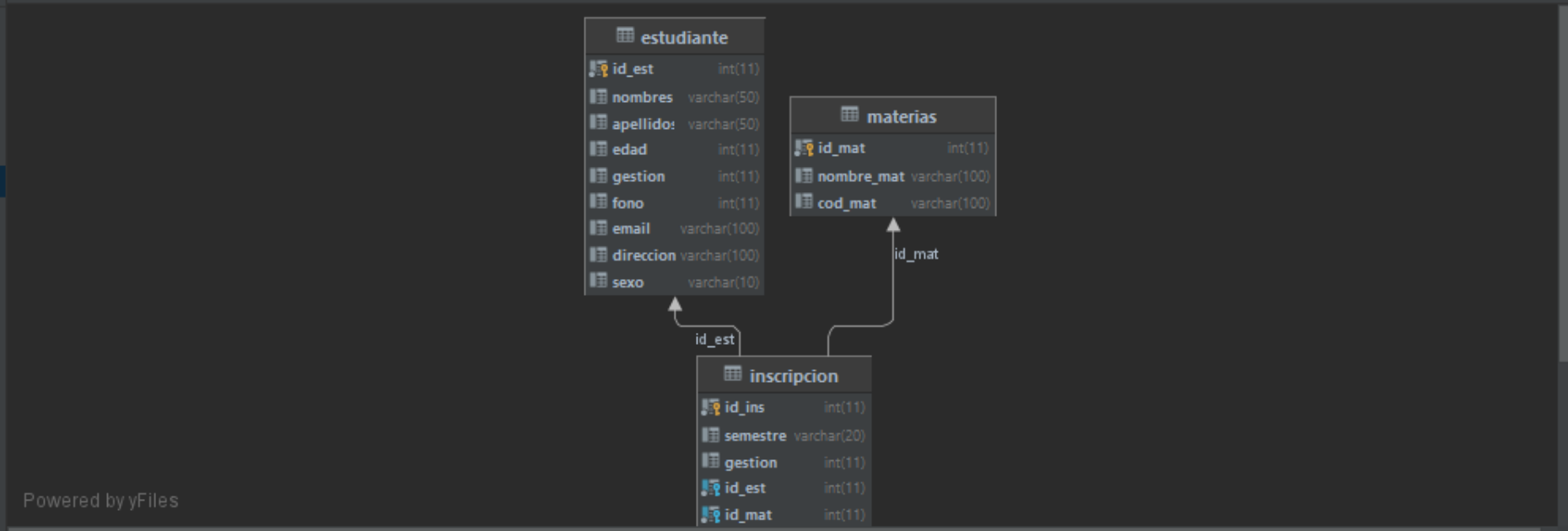
tareahito2

Add Configuration...

Database

MySQL - @localhost 8 of 11
- information_schema
- mysql
- pollos_copa
- pollos_copa2
- tareahito2
  - tables 3
    - estudiante
    - inscripcion
    - materias
- uni_hito2
- unihito2
- universidad
- Server Objects

console | console_2 | console_4 | console_5 | tareahito2



Services

- console_1 113 ms
- console_3 117 ms
  - console_3 117 ms
- console_4 358 ms
  - console_4 358 ms
- console_2 85 ms
  - console_2 85 ms
- console_5 2 s 156 ms
  - console_5 83 ms

```
22-04-04 23:02:35] Connected
REATE DATABASE tareaHito2
22-04-04 23:02:35] 1 row affected in 5 ms
SE tareaHito2
22-04-04 23:02:35] completed in 2 ms
eahito2> CREATE TABLE estudiante
         (
         id_est INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
         nombres VARCHAR(50),
         apellidos VARCHAR(50),
```

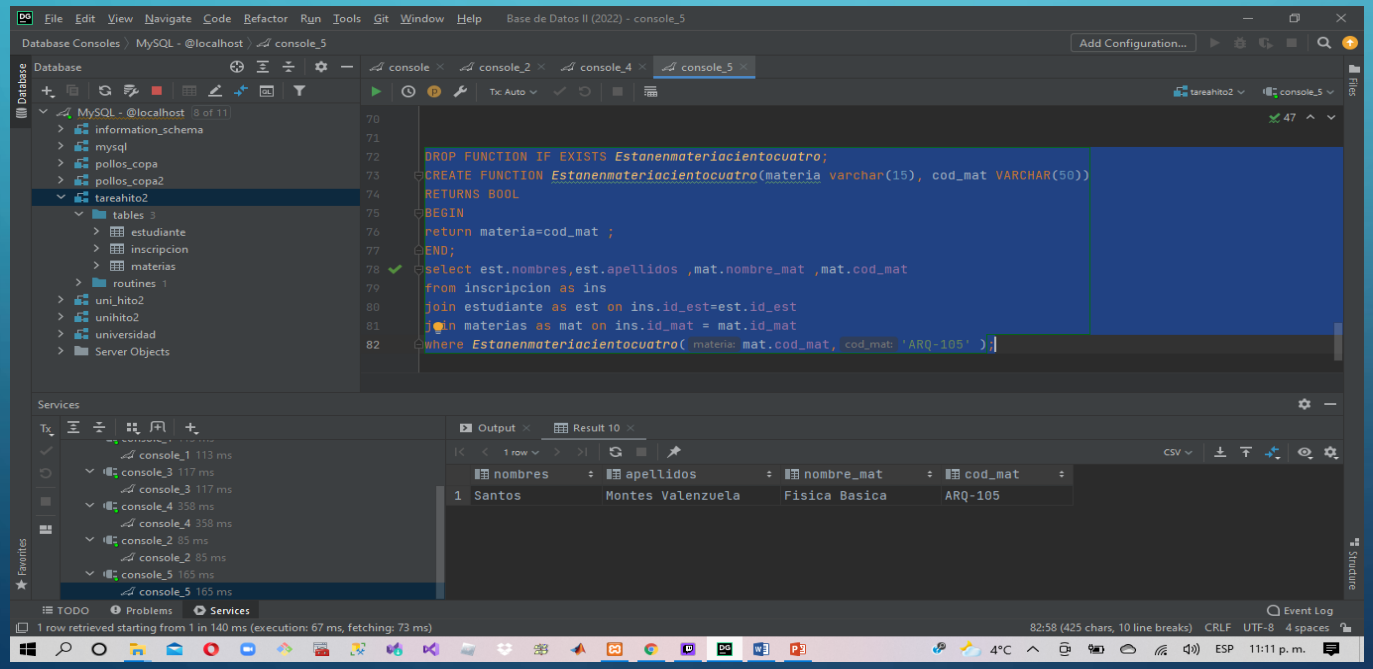
TODO Problems Services Event Log

Connected (5 minutes ago)

4°C ESP 11:08 p. m.

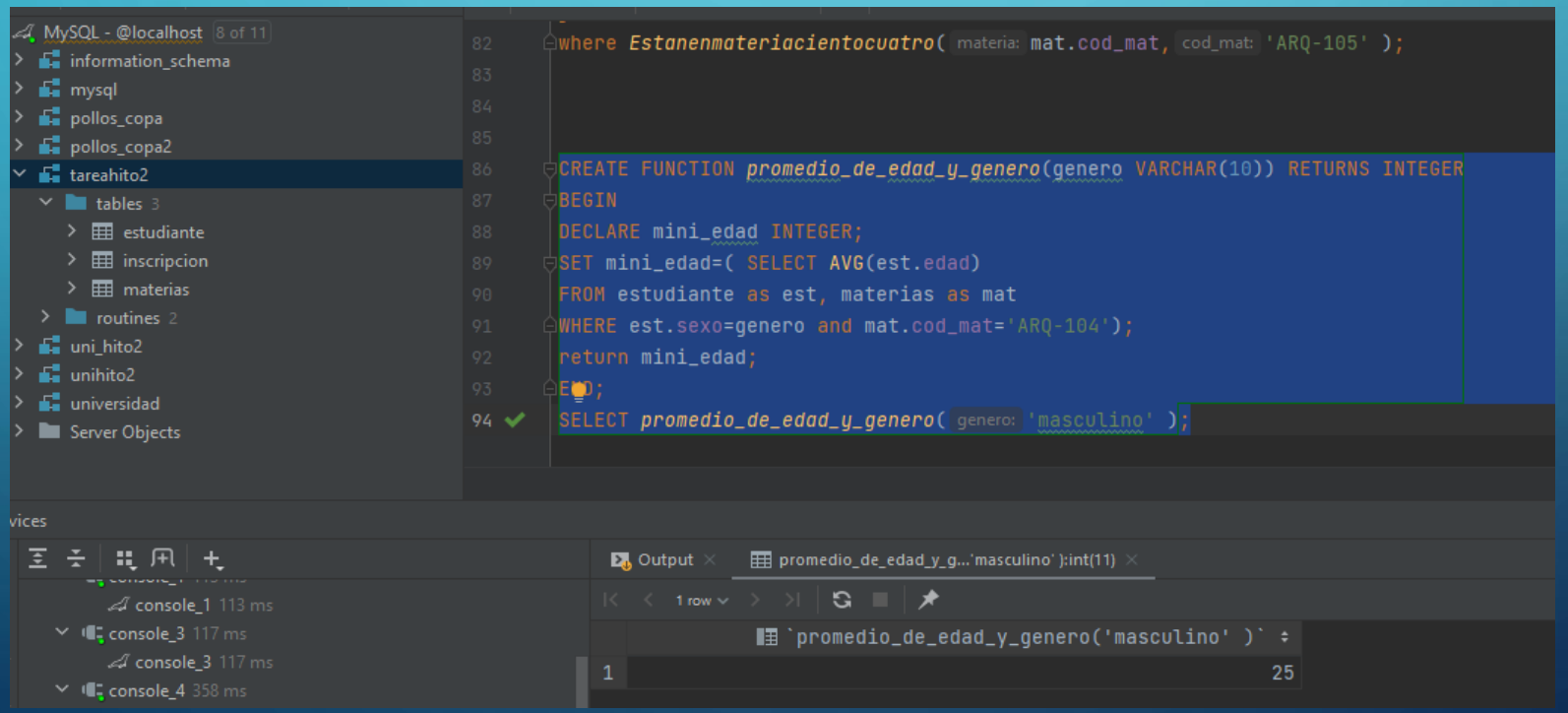
Resolver lo siguiente: ■ Mostrar los nombres y apellidos de los estudiantes inscritos en la materia ARQ-105, adicionalmente mostrar el nombre de la materia. ■ Deberá de crear una función que reciba dos parámetros y esta función deberá ser utilizada en la cláusula WHERE.

```
DROP FUNCTION IF EXISTS Estanenmateriacientocuatro;
CREATE FUNCTION Estanenmateriacientocuatro(materia varchar(15), cod_mat VARCHAR(50))
RETURNS BOOL
BEGIN
return materia=cod_mat ;
END;
select est.nombres,est.apellidos ,mat.nombre_mat ,mat.cod_mat
from inscripcion as ins
join estudiante as est on ins.id_est=est.id_est
join materias as mat on ins.id_mat = mat.id_mat
where Estanenmateriacientocuatro(mat.cod_mat,'ARQ-105' );
```



14.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 solo el género. ○ La función retorna un valor numérico.

```
CREATE FUNCTION promedio_de_edad_y_genero(genero VARCHAR(10)) RETURNS INTEGER
BEGIN
DECLARE mini_edad INTEGER;
SET mini_edad=( SELECT AVG(est.edad)
FROM estudiante as est, materias as mat
WHERE est.sexo=genero and mat.cod_mat='ARQ-104');
return mini_edad;
END;
SELECT promedio_de_edad_y_genero('masculino' );
```



15.Crear una función que permita concatenar 3 cadenas. ○ La función recibe 3 parámetros. ○ Si la cadenas fuesen: ■ Pepito ■ Perez ■ 50 ○ La salida debería ser: Pepito - Perez - 50

```
CREATE FUNCTION concatenaresta(nombre VARCHAR(50), apellido VARCHAR(50), edad INTEGER)
RETURNS VARCHAR (104)
BEGIN
declare info_per varchar(104);
set info_per= concat(nombre,'-',apellido,'-',edad);
return info_per;
END;
SELECT concatenaresta('Pepito','Perez',25);
```