



Tarea HIT02

09.04.20XX

—

Base de Datos II

Unifranz

SedeelAlto

Hito2

Objetivos de la tarea.

1. Mostrar el manejo de **bases de datos relacionales**.
2. Mostrar el manejo de **funciones en MySQL**.

Consigna.

Esta tarea se divide en dos partes:

- La primera parte corresponde a la **parte TEÓRICA necesaria**, en donde se encuentra un conglomerado de preguntas relacionadas a BASES DE DATOS RELACIONALES.
- La segunda parte corresponde a la **parte PRÁCTICA necesaria** en donde deberá realizar y crear funciones manejando el motor de base de datos MySQL.

Consideraciones sobre la entrega.

La tarea es abierta, usted decide cómo entregar su tarea. Puede agregar imágenes, enlaces, etc. Sin embargo, considerar los siguientes enunciados.

- Deberá generar una presentación hecha en **POWER POINT**. Después este mismo archivo deberá ser convertido a un **archivo PDF**.
- Después de tener disponible la presentación, debe de **GENERAR UN VIDEO** explicando todo su contenido.
- Después de tener disponible el video, el video lo puede subir a youtube o a drive o a cualquier plataforma.
- Finalmente, todo lo generado, es decir:
 - a. El archivo **ppt** (powerpoint).
 - b. El archivo **pdf** (el powerpoint convertido a PDF).
 - c. El **video** subido a alguna plataforma.
 - d. IMPORTANTE:
 - Tiene que ser subido a la plataforma **GITHUB**.
 - Para hacer este proceso tiene que tener sus archivos ya disponibles.

- En la plataforma **MOODLE** solo deberá de subir una carátula referenciando a su tarea que se encuentra en github.

Ejemplo:

UNIVERSIDAD PRIVADA FRANZ TAMAYO
DEFENSA HITO 2 - TAREA FINAL
Estudiante: Univ. Nombres Apellidos
Asignatura: BASE DE DATOS II
Carrera: INGENIERÍA DE SISTEMAS
Paralelo: BDA (1)
Docente: Lic. William Barra Paredes
fecha: xx/xx/2022
GITHUB: https://github.com/dheeyi/base-de-datos-ii (aquí va el enlace a su cuenta de github)

A partir de este punto están las preguntas TEÓRICAS y PRÁCTICAS a resolverse.

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


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.

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

R.-MySQL permite almacenar y acceder a los datos a través de múltiples motores de almacenamiento

4. ¿Qué son las funciones de agregación?



R.-Una función de agregación es una función que resume las filas de un grupo en un solo valor.

5. ¿Qué llegaría a ser XAMPP?

R.-XAMPP es una plataforma de desarrollo para los sistemas operativos Linux, Windows, MacOS X y Solaris que incluye Apache, MySQL, PHP y Perl

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

R.-

7. ¿Para qué sirve el comando USE?

R.-El use nos sirve para posicionarnos en una base de datos exacta

8. Que es DML y DDL?

R.- El DML .-Utilizando instrucciones de SQL, permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

DDL.- Es la parte del SQL dedicada a la definición de la base de datos, consta de sentencias para definir la estructura de la base de datos, permite definir gran parte del nivel interno de la base de datos.

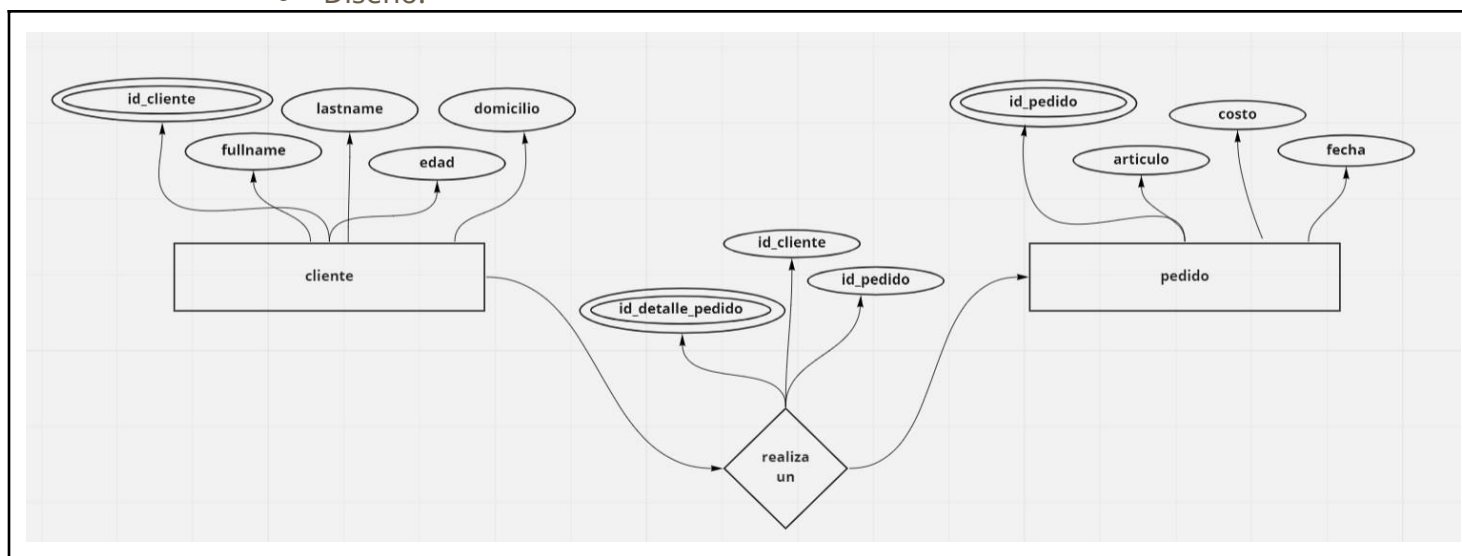
9. ¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc.

10. ¿Cómo crear, modificar y cómo eliminar una función?

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.

```
create DATABASE pollos_copa;
use pollos_copa;
create table cliente(
  cliente_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
  fullname VARCHAR(20),
  lastname VARCHAR(40),
  edad INT,
  domicilio VARCHAR(100)
);

create table pedido(
  pedido_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
  articulo VARCHAR(20),
  costo DECIMAL(7,2),
  fecha DATE
);

create table detalle(
  detalle_pedido_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
  cliente_id INT NOT NULL,
  pedido_id INT,
  FOREIGN KEY (cliente_id) REFERENCES cliente(cliente_id),
  FOREIGN KEY (pedido_id) REFERENCES pedido(pedido_id)
);

INSERT INTO cliente(fullname,lastname,edad,domicilio) VALUES('Miguel', 'Marino Condori', 20, 'Z Murillo');
INSERT INTO cliente(fullname,lastname,edad,domicilio) VALUES('Mariana', 'Marina Cespedes', 32, 'Z Los Andes');

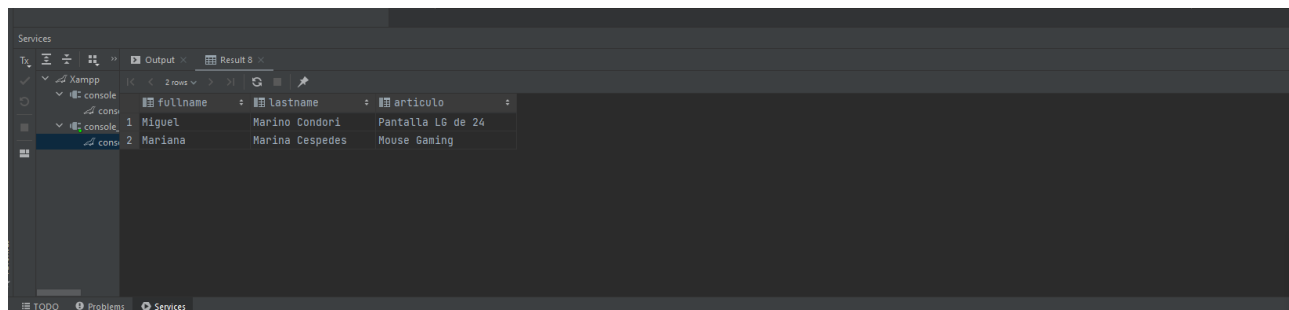
INSERT INTO pedido(articulo, costo, fecha) VALUES('Pantalla LG de 24', '1500', '2022-04-04');
INSERT INTO pedido(articulo, costo, fecha) VALUES('Mouse Gaming', '450', '2022-04-04');

INSERT INTO detalle(cliente_id, pedido_id) VALUES(1,1);
INSERT INTO detalle(cliente_id, pedido_id) VALUES(2,2);
```

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**.

```
SELECT c.fullname, c.lastname, p.articulo
FROM detalle d
INNER JOIN cliente c on c.cliente_id = d.cliente_id
INNER JOIN pedido p on p.pedido_id = d.pedido_id
```

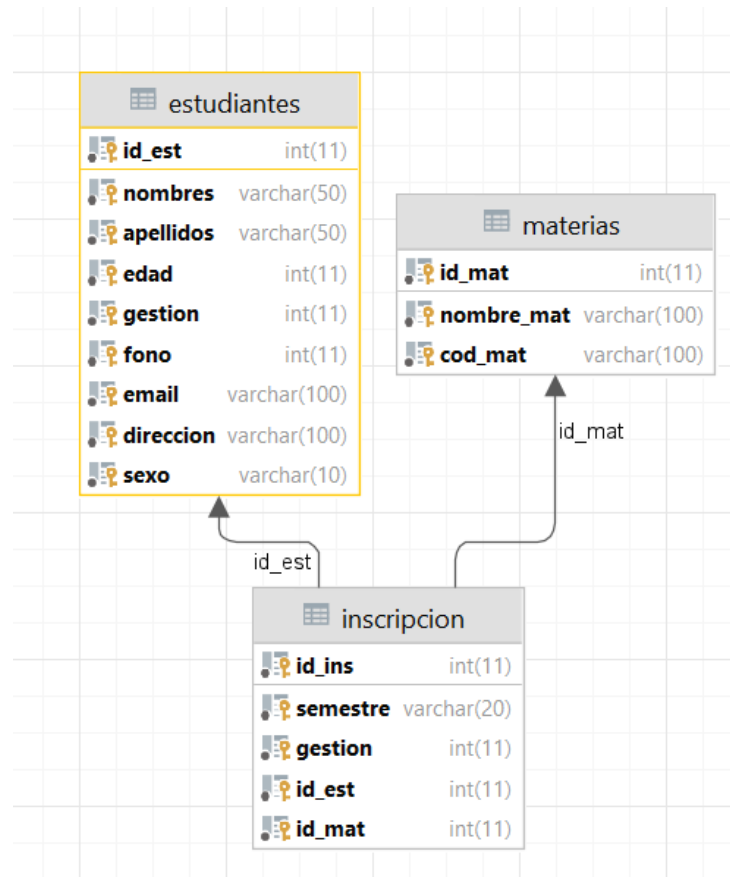


	fullname	lastname	articulo
1	Miguel	Marino Condori	Pantalla LG de 24
2	Mariana	Marina Cespedes	Mouse Gaming

13. Crear un función que compare dos códigos de materia.

- Recrear la siguiente base de datos:

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



```
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 Uria', 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 ('Introduccion a la Arquitectura', 'ARQ-101'),
      ('Urbanismo y Diseno', 'ARQ-102'),
      ('Dibujo y Pintura Arquitectonico', 'ARQ-103'),
      ('Matematica discreta', 'ARQ-104'),
      ('Fisica Basica', '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.
- Ejemplo:



```
SELECT est.id_est,
      nombres,
      apellidos,
      nombre_mat,
      cod_mat
FROM estudiantes AS est
WHERE comparaMaterias(mat.cod_mat, 'ARQ-105');
```

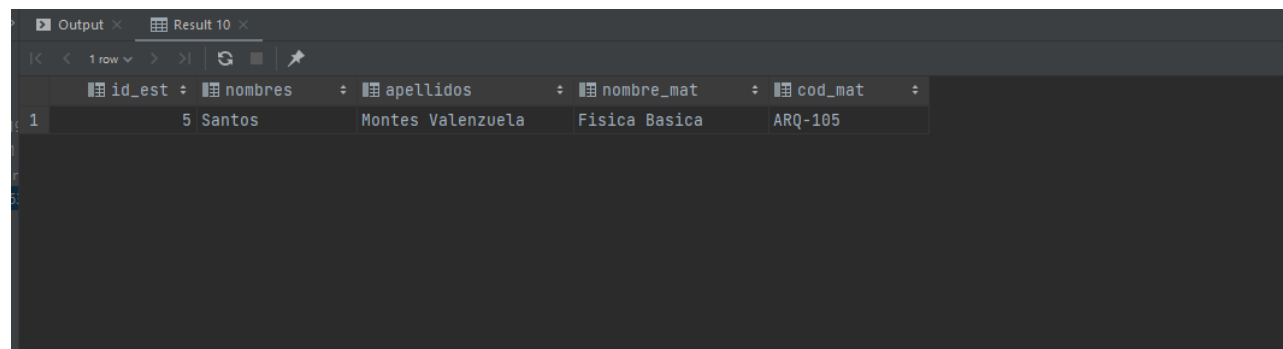
- El resultado al **ejecutar la consulta SQL** debería ser el siguiente:

id_est	nombres	apellidos	nombre_mat	cod_mat
5	Santos	Montes Valenzuela	Fisica Basica	ARQ-105

```
create function comparaMaterias(mat VARCHAR(50), val VARCHAR(50))
returns BOOLEAN
begin
return (select if(mat = val, TRUE, FALSE));
```

```
end;

SELECT e.id_est, e.nombres, e.apellidos, m.nombre_mat, m.cod_mat
FROM estudiantes e
INNER JOIN inscripcion i on i.id_est = e.id_est
INNER JOIN materias m on m.id_mat = i.id_mat
WHERE comparaMaterias(m.cod_mat, 'ARQ-105')
```



The screenshot shows a database query result window with two tabs: 'Output' and 'Result 10'. The 'Result 10' tab is active, displaying a single row of data. The window has a toolbar with navigation icons (back, forward, search, etc.) and a table structure header. The data row contains the following values:

	id_est	nombres	apellidos	nombre_mat	cod_mat
1	5	Santos	Montes Valenzuela	Fisica Basica	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.

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**

16. Crear una función de acuerdo a lo siguiente:

- Mostrar el **nombre, apellidos y el semestre** de todos los estudiantes que estén inscritos. Siempre y cuando la suma de las edades del sexo **femenino o masculino** sea par y **mayores** a cierta edad.
- Debe de crear una función que sume las edades (recibir como parámetro el sexo, y la edad).
 - Ejemplo: sexo='Masculino' y edad=22
 - Note que la función recibe 2 parámetros.
- La función creada anteriormente debe utilizarse en la consulta SQL. (Cláusula WHERE).

17. Crear una función de acuerdo a lo siguiente:

- Crear una función sobre la tabla estudiantes que compara un nombre y apellidos. (si existe este nombre y apellido mostrar todos los datos del estudiante).
 - La función devuelve un boolean.
 - La función debe recibir el nombre y sus apellidos.
- Ejemplo:

```
create function comparaNombreLastanem(nombres varchar(50), apellidos varchar(50))
```

- La función debería ser usada en la cláusula WHERE.