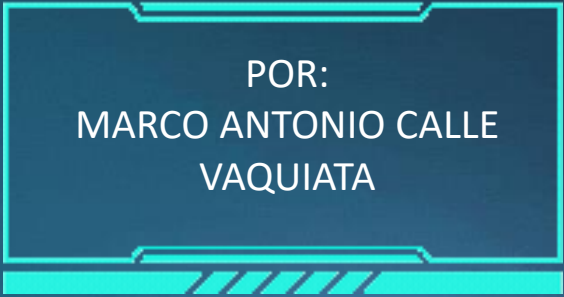





BASE DE DATOS II

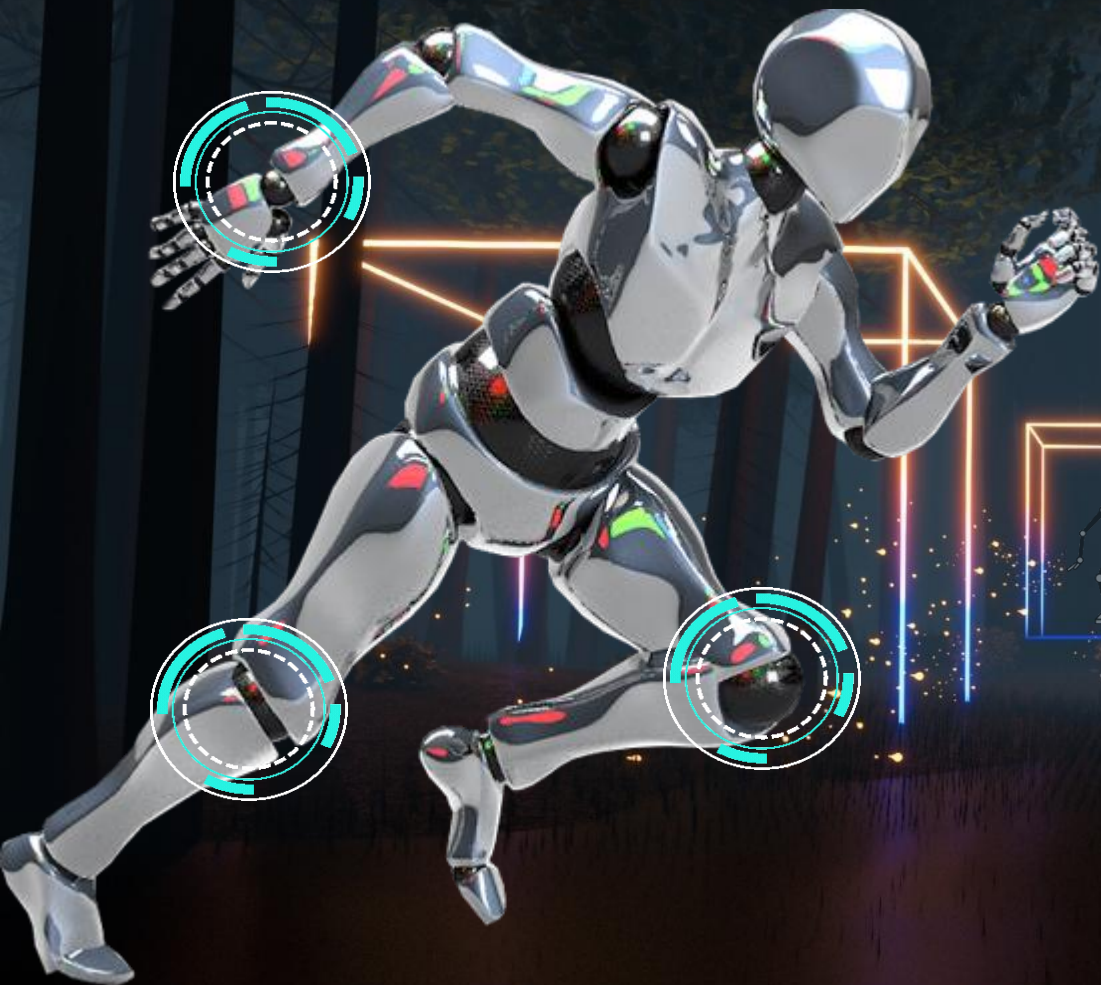
HITO-2



POR:
MARCO ANTONIO CALLE
VAQUIATA



Índice



Manejo de Conceptos

Parte Practica

Conclusión

Conceptos

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

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.

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

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.





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

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.

¿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... sobre un conjunto de valores.

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

WAMP es un acrónimo que significa Windows, Apache, MySQL y PHP. Es un stack o conjunto de soluciones de software que significa que cuando instalas WAMP, estás instalando Apache, MySQL y PHP en tu sistema operativo (Windows en el caso de WAMP). Aunque puede instalarlos por separado, por lo general son empaquetados, y también por una buena razón

¿Cuál es la diferencia entre las funciones de agregación y funciones creados por el DBA? Es decir funciones creadas por el usuario

Las funciones de agregación es el que ya esta determinado tanto en el SQL/DATAGRIP, ya están determinadas(SUM,REST,MAX,MIN,ETC) mientras que las funciones creadas por el usuario son las que ponemos de acuerdo a la necesidades

¿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.

¿Que es DML y DDL??

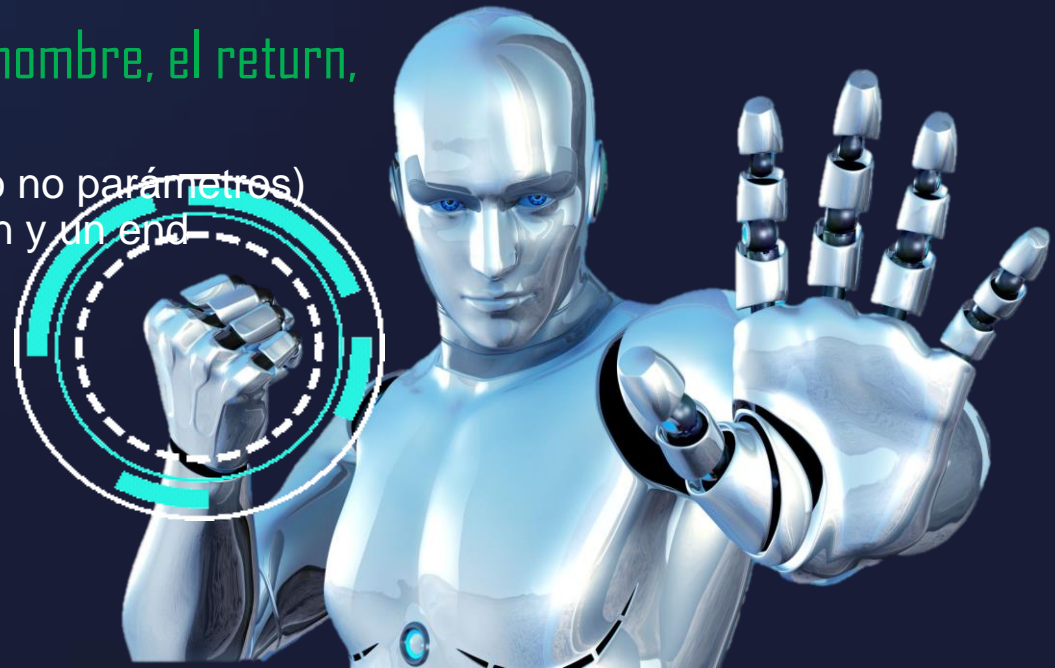
Las sentencias SQL se dividen en dos categorías: **lenguaje de definición de datos (DDL)** y **lenguaje de manipulación de datos (DML)**. Las sentencias DDL se utilizan para describir una base de datos, para definir su estructura, para crear sus objetos y para crear los sujetos de la tabla.

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

Las características que debe de tener son el nombre de la función(si tiene o no parámetros) un returns dependiendo si va a ser varchar o integer después un begin y un end

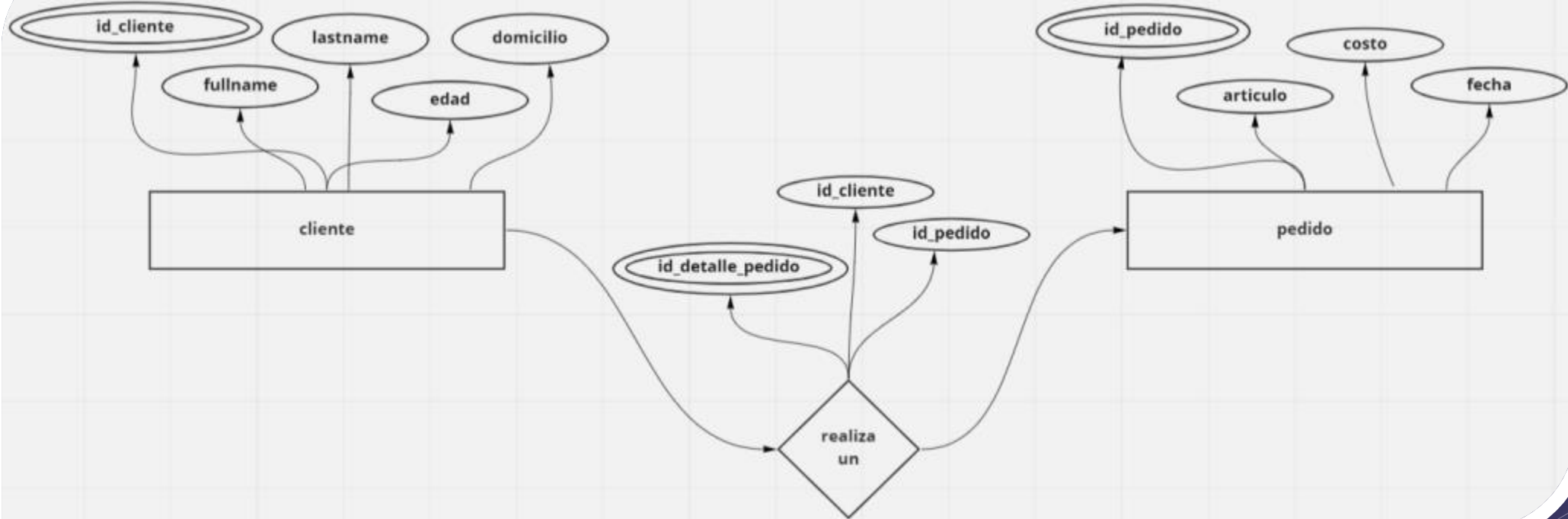
¿Cómo crear, modificar y cómo eliminar una función?

CREATE	Cree un índice o tabla que aún no existe.
ALTER	Modifique una tabla o columna existente.
DROP	Eliminar una tabla, columna o restricción existente.



Parte practica

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



Resultados

```
create database POLLOS_COPA;
use POLLOS_COPA;
create table cliente(
    id_cliente integer primary key,
    name varchar (30),
    lastname varchar(30),
    edad integer,
    domicilio varchar(50));
create table pedido(
    id_pedido integer primary key ,
    articulo varchar (30),
    costo integer,
    fecha varchar (20)
```

```
create table detalle_pedido(
    id_detalle_pedido integer primary key,
    id_cliente integer,
    id_pedido integer,
    foreign key (id_cliente) references cliente (id_cliente),
    foreign key(id_pedido)references pedido(id_pedido)
);
insert into cliente(id_cliente,name,lastname,edad,domicilio)
values (2,'MARCO','CALLE',20,'Av.flores calle 11'),
(5,'MARIA','CELESTE',19,'Av. civica calle 2');
insert into pedido(id_pedido,articulo,costo,fecha)
values (1,'ROPA',530,'23/12/2013'),
(4,'ZAPATO',600,'12/02/2006');
```

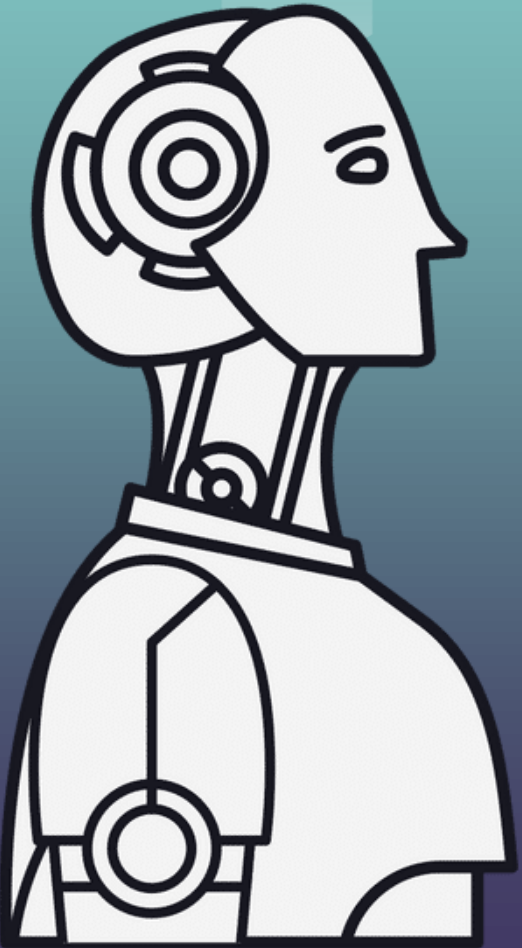
```
insert into detalle_pedido(id_detalle_pedido, id_cliente, id_pedido)
values (1,2,3),(4,5,6);
```



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*  
from cliente as c  
inner join detalle_pedido dp on  
id_detalle_pedido = dp.id_detalle_pedido  
inner join pedido p on dp.id_pedido = p.id_pedido
```



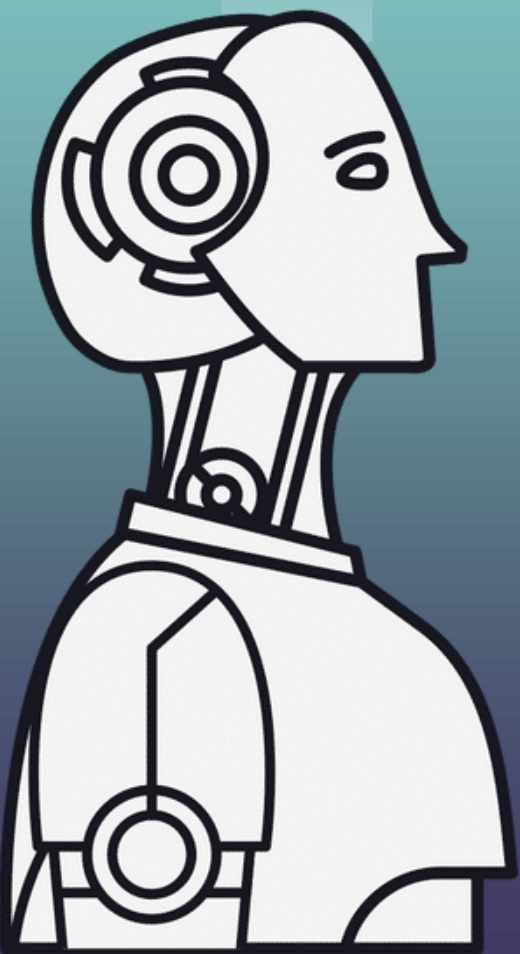
Crear un función que compare dos códigos de materia.

Recrear la siguiente base de datos:

```
create database tareaHito2;
use tareaHito2;
CREATE TABLE estudiantes
(
  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 materias
(id_mat INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
 nombre_mat VARCHAR(100),
 cod_mat VARCHAR(100));
CREATE TABLE inscripcion
(id_ins INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
 id_est INT NOT NULL,
 id_mat INT NOT NULL,
 semestre VARCHAR(20),
 gestion INTEGER,
 FOREIGN KEY (id_est) REFERENCES estudiantes (id_est),
 FOREIGN KEY (id_mat) REFERENCES materias (id_mat))
```





```
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion)
VALUES ('Miguel', 'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino');
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion)
VALUES ('Sandra', 'Mavir Uria', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino');
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion)
VALUES ('Joel', 'Adubiri Mondar', 30, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino');
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion)
VALUES ('Andrea', 'Arias Ballesteros', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino');
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion)
VALUES ('Santos', 'Montes Valenzuela', 24, 2832119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

```
lidos, edad, fono, email, direccion, sexo)
20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino');
lidos, edad, fono, email, direccion, sexo)
2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino');
lidos, edad, fono, email, direccion, sexo)
, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino');
lidos, edad, fono, email, direccion, sexo)
', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino');
lidos, edad, fono, email, direccion, sexo)
', 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 Arquitectonica', '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', 2015);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (1, 2, '2do Semestre', 2015);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (2, 4, '1er Semestre', 2016);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (2, 3, '2do Semestre', 2016);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (3, 3, '2do Semestre', 2017);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (3, 1, '3er Semestre', 2017);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (4, 4, '4to Semestre', 2017);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES (5, 5, '5to Semestre', 2017);

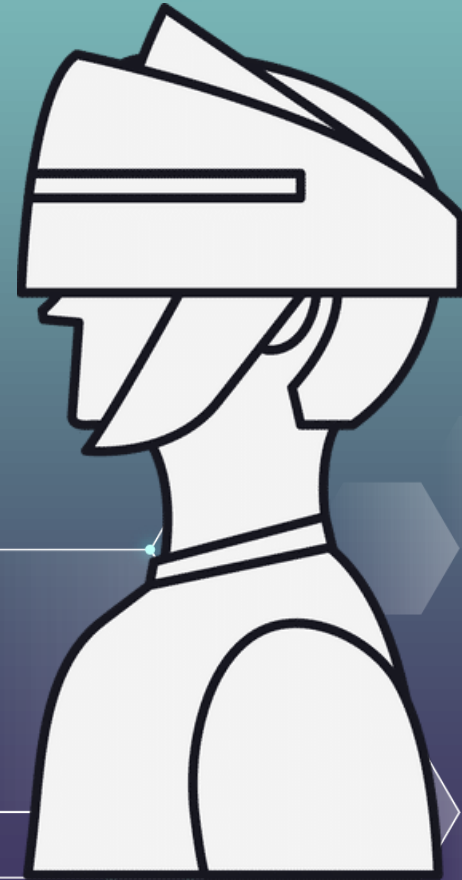
```

*"Que tu actitud sea positiva sin
importar el pesimismo de quienes te
rodean."*

```

(mat) VALUES ('Introduccion a la Arquitectura', 'ARQ-101');
(mat) VALUES ('Urbanismo y Diseno', 'ARQ-102');
(mat) VALUES ('Dibujo y Pintura Arquitectonica', 'ARQ-103');
(mat) VALUES ('Matematica discreta', 'ARQ-104');
(mat) VALUES ('Fisica Basica', 'ARQ-105');
t, semestre, gestion) VALUES (1, 1, '1er Semestre', 2015);
t, semestre, gestion) VALUES (1, 2, '2do Semestre', 2015);
t, semestre, gestion) VALUES (2, 4, '1er Semestre', 2016);
t, semestre, gestion) VALUES (2, 3, '2do Semestre', 2016);
t, semestre, gestion) VALUES (3, 3, '2do Semestre', 2017);
t, semestre, gestion) VALUES (3, 1, '3er Semestre', 2017);
t, semestre, gestion) VALUES (4, 4, '4to Semestre', 2017);
t, semestre, gestion) VALUES (5, 5, '5to Semestre', 2017);

```



Resolver lo siguiente:

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

```
select nombres,apellidos,nombre_mat
from estudiantes as es
inner join materias as ma on id_mat=ma.id_mat
where cod_mat='ARQ-105'
```

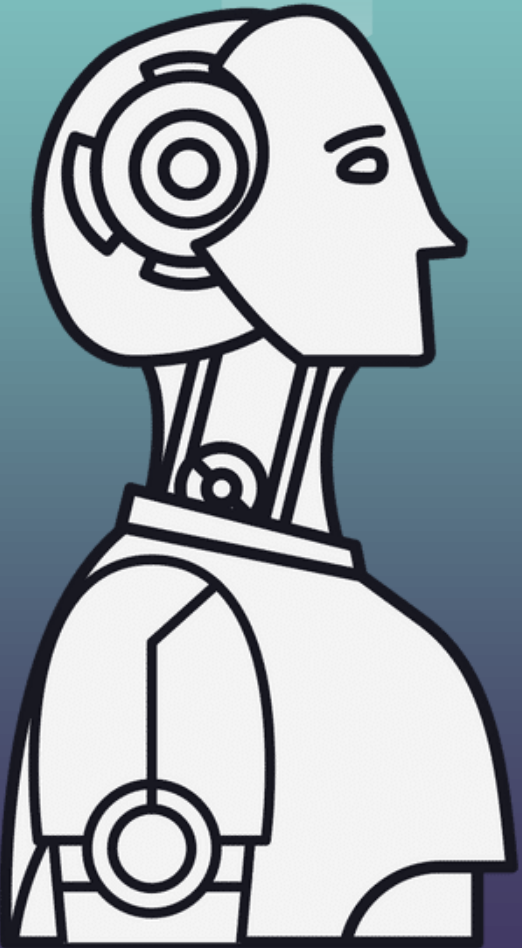
	■ nombres	÷ ■ apellidos	÷ ■ nombre_mat
1	Miguel	Gonzales Veliz	Fisica Basica
2	Sandra	Mavir Uria	Fisica Basica
3	Joel	Adubiri Mondar	Fisica Basica
4	Andrea	Arias Ballesteros	Fisica Basica
5	Santos	Montes Valenzuela	Fisica Basica

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



```
CREATE or replace FUNCTION DOS(P1 VARCHAR(50),P2 INT)
RETURNS integer
BEGIN
    declare pan integer default 0;
    SELECT MAX(edad) into pan
    FROM estudiantes AS ES
    WHERE sexo=P1 && EDAD>P2;
    return pan;
end;
select nombres,apellidos
from estudiantes as es
where DOS( P1: 'masculino', P2: 23);
```

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.



```
create function get_Avg1(p1 varchar(20),p2 varchar (20))
    RETURNS integer
begin
    DECLARE EDADES REAL DEFAULT 0;
    SELECT Avg(est.edad) INTO EDADES
    from inscripcion as ins
    inner join estudiantes as est on ins.id_est = est.id_est
    inner join materias mat on ins.id_mat = mat.id_mat
    WHERE est.sexo = p1 and cod_mat=p2;
    return EDADES;
end;
```

```
select est.nombres, est.apellidos, mat.nombre_mat
from estudiantes as est
inner join materias mat on est.id_est = mat.id_mat
where get_Avg1( p1: 'masculino', p2: 'ARQ-104|')
```


Crear una función que permita concatenar 3 cadenas.

```
create function concatenar2(p1 varchar(50),p2 varchar(50),p3 int)
returns varchar (100)
begin
    declare concatenado varchar(100) default '';
    set concatenado= CONCAT_WS(' ','NOMBRES:',p1,'APELLIDOS:',p2,'EDADES:',p3);
    RETURN concatenado;
end;

SELECT concatenar2( p1: 'MARCO', p2: 'CALLE', p3: 20) AS DATOS;
```

■ DATOS

1 NOMBRES: MARCO APELLIDOS: CALLE EDADES: 20



Crear una función de acuerdo a lo siguiente:

```
create or replace function SUMA_EDADES2(p1 varchar(50),p2 varchar(50)) returns varchar (100)
BEGIN
    RETURN
        (SELECT SUM(e.edad)
         FROM estudiantes AS e
         WHERE e.sexo=p1 and e.edad>p2
        );
END;
```

```
select SUMA_EDADES2( p1: 'masculino', p2: 22)as datos;
```

```
select nombres,apellidos,edad,semestre
from estudiantes e
inner join inscripcion i on e.id_est = i.id_ins
where e.edad=SUMA_EDADES2( p1: 'masculino', p2: 22)%2=0;
```



· Crear una función de acuerdo a lo siguiente:

```
CREATE OR REPLACE FUNCTION BUSCAR_PE(P1 VARCHAR(50),P2 VARCHAR(50),P3 VARCHAR(50),P4 VARCHAR(50)) RETURNS BOOLEAN
BEGIN
    DECLARE BUSCAR BOOLEAN DEFAULT FALSE;

    if ((P1=P2)AND (P3=P4))
    then
        set BUSCAR=TRUE;
    ELSE
        SET BUSCAR=FALSE;
    end if;
    RETURN BUSCAR ;
end;

SELECT BUSCAR_PE( P1: 'A', P2: 'A', P3: 'B', P4: 'B');

SELECT nombres,apellidos
FROM estudiantes AS ES
WHERE BUSCAR_PE( P1: ES.nombres, P2: 'Sandra', P3: ES.apellidos, P4: 'Mavir Uria');
```



Conclusión

En este tema nuestros objetivos fueron:

Mostrar el manejo de bases de datos relacionales.

Mostrar el manejo de funciones en MySQL.

Se concluyo que alcanzamos exitosamente con los objetivos

Pudiendo alcanzar nuevos temas y aprendes mas cada día.

**“En la lucha por la supervivencia, quien gana es quien permanece,
así no alcance la victoria, porque esto es una lucha diaria”**

A futuristic, metallic robotic hand is shown reaching out from the left side of the frame. The hand has a sleek, silver design with blue glowing joints at the wrist, elbow, and fingers. The background is a deep black space filled with numerous small, distant stars. A bright, glowing blue light source is visible in the upper right, creating a lens flare effect. A rectangular frame with a cyan border and a diagonal hatch pattern at the bottom encloses the word "Gracias".

Gracias