Tarea

ANA CRISTINA CALDERÒN ORTEGA

Manejo de Conceptos

Base de Datos II

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

R.- Las bases de datos relacionales son un conjunto de tablas.

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

R.- Las bases de datos relacionales son un conjunto de tablas.

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

R.- Es un sistema de gestión de bases de datos relacionales de código abierto. MariaDB es un sistema de base de datos proveniente de MySQL pero con licencia, son diferentes por que ambos son autónomos.

¿Qué llegaría a ser XAMPP?

R.- XAMPP llegaría a se un servidor para gestionar bases de datos en MySQL.

¿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.

Funciones de Agregado	
Función	**Descripción**
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

¿Para qué sirve el comando USE?

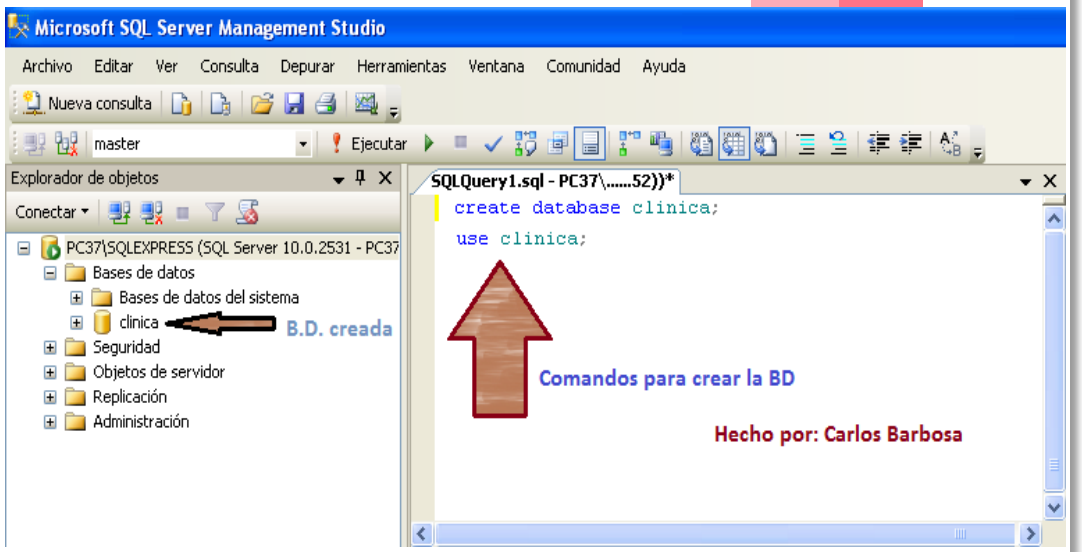
R.- La sentencia USE db_name indica a MySQL que use la base de datos

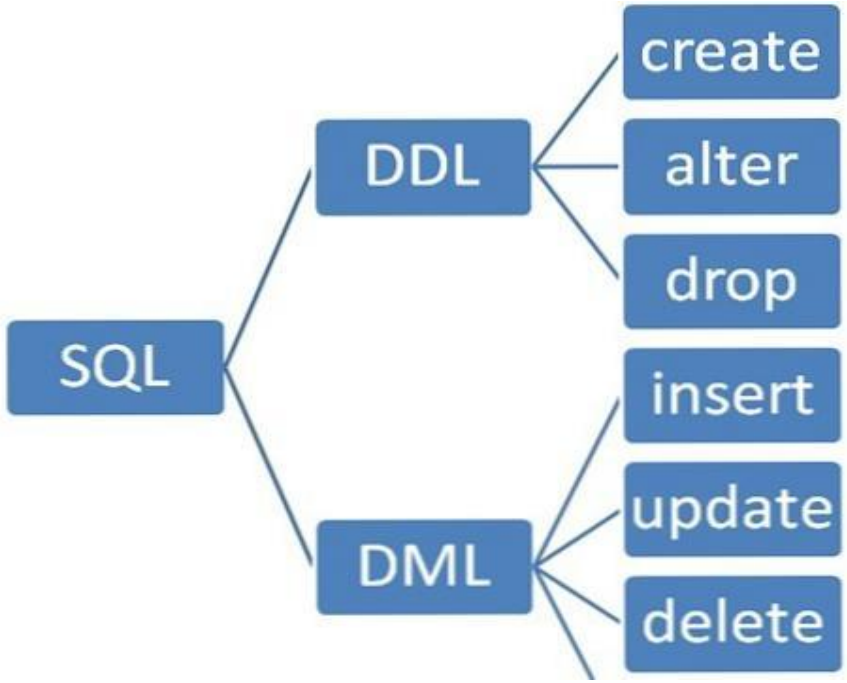
¿Que es DML y DDL?

R.- DDL es el lenguaje que permite definir datos.
Lenguaje de manipulación de datos (DML)

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

R.- En las funciones de agregación son creadas por el propio Software a comparación de las DBA que el usuario o administrador de la base puede crear para realizar tareas específicas.





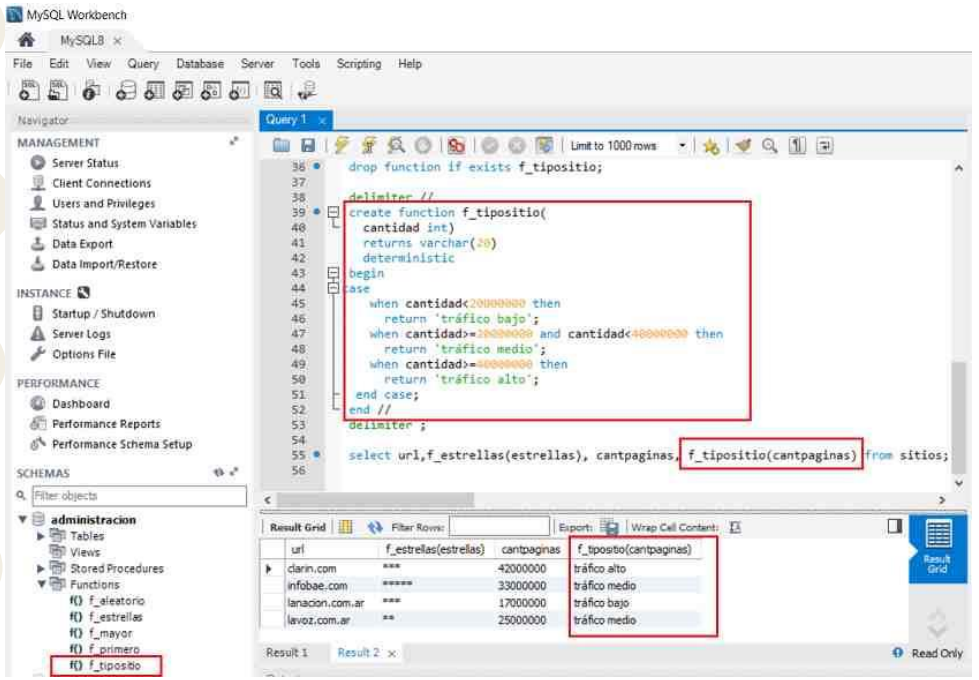
¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc.

R.- Una función debe estar en la clausula select cuando realizamos una consulta al crear una función propia debemos usar comando como créate funtion el nombre de la función, darle parámetros e indicar el tipo, utilizar el retorn, begin end y podemos utilizar el select, from y where también podemos crear una función y ponerla en where junto a los inner joins.

¿Cómo crear, modificar y cómo eliminar una función?

R.- Para eliminar una función
DROP FUNCTION IF EXISTS
max_edad_estudiantes;

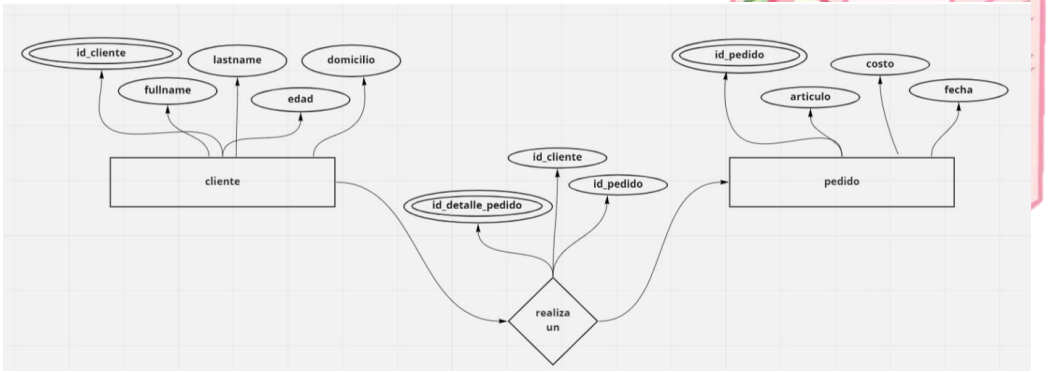
Para crear y alterar una función
CREATE OR REPLACE FUNCTION
max_edad_estudianes() RETURNS
intBEGIN
return
(
);
SELECT max(est.edad)
FROM estudiantes AS est
END;



TAREA

Parte practica

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




```
create database POLLOS_COPA;
USE POLLOS_COPA;

CREATE TABLE CLIENTE
(
   ID_CLIENTE INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL ,
   FULLNAME VARCHAR(50) NOT NULL,
   LASTNAME VARCHAR(50) NOT NULL,
   EDAD INTEGER NOT NULL,
   DOMICILIO VARCHAR(200) NOT NULL
);

CREATE TABLE PEDIDO
(
   ID_PEDIDO INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
   ARTICULO VARCHAR(50) NOT NULL,
   COSTO INTEGER NOT NULL,
   FECHA DATE NOT NULL
);
```

```sql
CREATE TABLE DETALLE_PEDIDO
(
   ID_DETALLE_PEDIDO INTEGER AUTO_INCREMENT PRIMARY KEY NOT
NULL,
   ID_CLIENTE INTEGER NOT NULL,
   ID_PEDIDO INTEGER NOT NULL
);

INSERT INTO CLIENTE(FULLNAME, LASTNAME, EDAD, DOMICILIO) VALUES
('Karla Albina', 'Calderon Arroyo', 54, 'Calle Chuquisaca, La Paz'),
('Jaime', 'Fernandez Quilla', 45, 'San Pedro'),
('Fernando Carlos', 'Torrez Mujica', 21, 'Zona 16 de julio El Alto');

INSERT INTO PEDIDO(ARTICULO, COSTO, FECHA) VALUES
('1/4 de pollo con papas y fideo', 23, '2021-05-04');
INSERT INTO PEDIDO(ARTICULO, COSTO, FECHA) VALUES
('1/8 de pollo con papas y uns coca cola de 1.5lt', 40, '2021-05-18'),
('2 unid de 1/8 pollo con arroz y 2 personales', 66, '2021-06-01');

INSERT INTO DETALLE_PEDIDO(ID_CLIENTE, ID_PEDIDO) values
(1,1),
(2,2),
(3,3);
```

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.

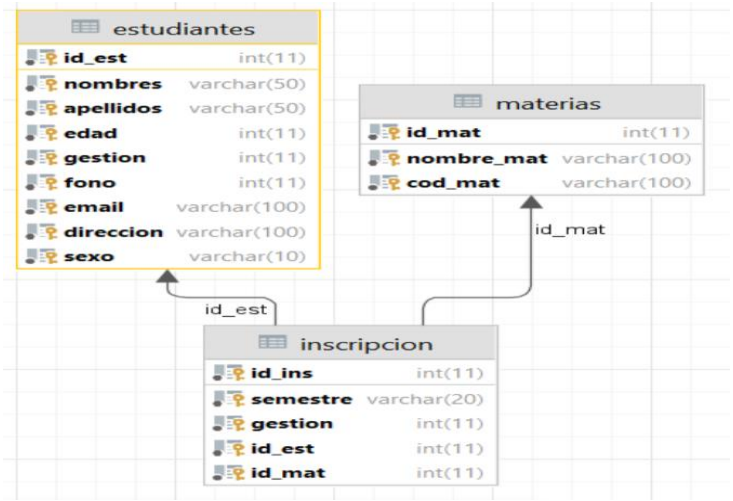
```sql
select C.FULLNAME, C.FULLNAME
from DETALLE_PEDIDO as dp
inner join CLIENTE C on dp.ID_CLIENTE = C.ID_CLIENTE
    inner join PEDIDO P on dp.ID_PEDIDO = P.ID_PEDIDO
where P.COSTO>20;
```

3 rows

	C.FULLNAME	C.FULLNAME
1	Karla Albina	Karla Albina
2	Jaime	Jaime
3	Fernando Carlos	Fernando Carlos

Crear un función que compare dos códigos de materia.

- Recrear la siguiente base de datos:



```sql
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, 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);
```

```sql
CREATE FUNCTION COMPARARCODIGOS (CODIGO VARCHAR(20), materiaco
varchar(20))
RETURNS bool
BEGIN
      DECLARE CODI varchar(20);
      set CODI= (CODIGO=materiaco);
      RETURN (CODI);
end;

SELECT mat.*
from materias as mat
where COMPARARCODIGOS( mat.cod_mat, 'ARQ-105');
```

	id_mat	nombre_mat	cod_mat
1	5	Fisica Basica	ARQ-105

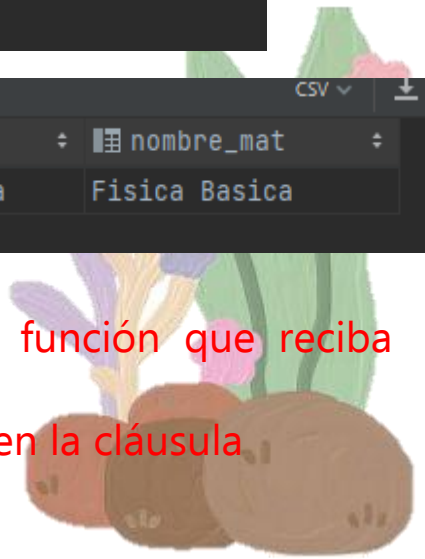
Resolver lo siguiente:

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

```
SELECT e.nombres, e.apellidos, m.nombre_mat
FROM inscripcion AS insc
inner join estudiantes e on insc.id_est = e.id_est
inner join materias m on insc.id_mat = m.id_mat
where m.cod_mat='ARQ-105';
```

1 row CSV

	nombres	apellidos	nombre_mat
1	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 function MAXIDEST(genero varchar(10), EDAD INTEGER )
  returns integer
  begin
    declare MAXID integer default 0;

    select MAX(est.id_est) INTO MAXID
    from estudiantes AS est
    where est.sexo = genero AND est.edad=EDAD;

    return MAXID;

  end;

select est.id_est, est.nombres, est.apellidos, m.nombre_mat,
m.cod_mat
from estudiantes as est
INNER JOIN inscripcion i on est.id_est = i.id_est
INNER JOIN materias m on i.id_mat = m.id_mat
where est.id_est= MAXIDEST('masculino', 24) ;
```

	id_est	nombres	apellidos	nombre_mat	cod_mat
1	5	Santos	Montes Valenzuela	Fisica Basica	ARQ-105

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.

```sql
CREATE FUNCTION PROMEDIOEDADES(GENERO VARCHAR(20))
RETURNS INTEGER
   BEGIN
       RETURN (select avg(est.edad)
      from estudiantes as est
      inner join inscripcion i on est.id_est = i.id_est
      inner join materias m on i.id_mat = m.id_mat
      where est.sexo=genero and m.cod_mat='ARQ-104');
   end;

SELECT PROMEDIOEDADES('femenino');
```

	`PROMEDIOEDADES('femenino')`
1	23

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

```sql
create function getNombreCompleto(
   nombres varchar(50),
   apellidos varchar(50),
   edad int
 )
 returns varchar(100)
 begin
   declare resultado varchar(100) default ''; #resultado = ''
   set resultado = concat(nombres, ' - ', apellidos,  ' - ', edad);#resultado = 'WillBar'
   return resultado;
 end;

select getNombreCompleto(est.nombres, est.apellidos, est.edad)
from estudiantes as est;
```

	`getNombreCompleto(est.nombres, est.apellidos, est.edad)`
1	Miguel - Gonzales Veliz - 20
2	Sandra - Mavir Uria - 25
3	Joel - Adubiri Mondar - 30
4	Andrea - Arias Ballesteros - 21
5	Santos - Montes Valenzuela - 24

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).

```
CREATE FUNCTION SUMEDAD(GENERO VARCHAR(20), EDAD INT)
RETURNS INTEGER
   BEGIN
      RETURN (SELECT SUM(EST.EDAD)
         FROM estudiantes AS EST
         WHERE EST.sexo=GENERO AND EST.edad>EDAD);
   end;

SELECT EST.nombres, EST.apellidos, i.semestre
FROM estudiantes AS EST
INNER JOIN inscripcion i on EST.id_est = i.id_est
inner join materias m on i.id_mat = m.id_mat
WHERE EST.edad=SUMEDAD('masculino', 22)%2=0;
```

	nombres	apellidos	semestre
1	Miguel	Gonzales Veliz	1er Semestre
2	Miguel	Gonzales Veliz	2do Semestre
3	Sandra	Mavir Uria	1er Semestre
4	Sandra	Mavir Uria	2do Semestre
5	Joel	Adubiri Mondar	2do Semestre
6	Joel	Adubiri Mondar	3er Semestre
7	Andrea	Arias Ballesteros	4to Semestre
8	Santos	Montes Valenzuela	5to Semestre

```
create function comparenomb(nombre varchar(20), apellido varchar(20), nombrecompa
varchar(20), apellidocompa varchar(20))
returns boolean
   begin
      declare comparar bool default false;

      set comparar=(nombre=nombrecompa and apellido=apellidocompa);
      return comparar;
   end;

select est.*
from estudiantes as est
where comparenomb(est.nombres, est.apellidos, 'Andrea', 'Arias Ballesteros');
```

1 row | Tx: Auto | DDL | CSV

	id_est	nombres	apellidos	edad	gestion	f
1	4	Andrea	Arias Ballesteros	21	<null>	28

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.

GRACIAS