



---

# PROCESUAL BASE DE DATOS II

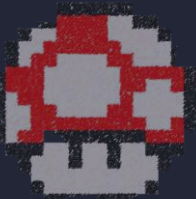
---

---

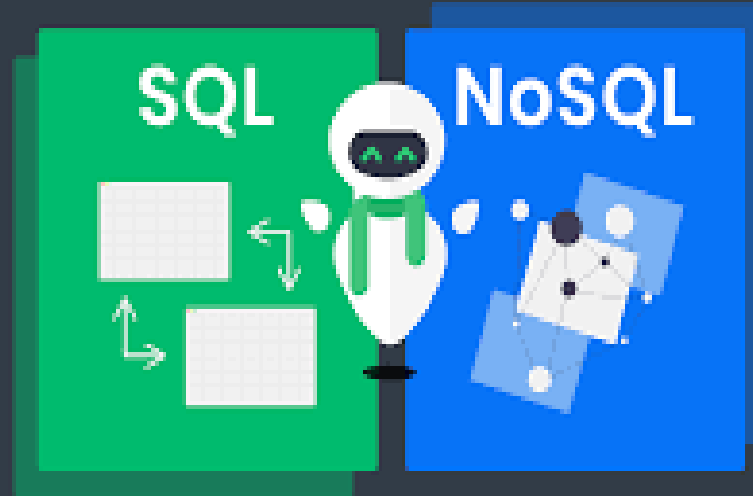
ANA CRISTINA  
CALDERÓN ORTEGA

# Parte Teórica

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



Las bases de datos relacionales son un conjunto de tablas. Las no relacionales no son propias del lenguaje SQL.



# Parte Teórica

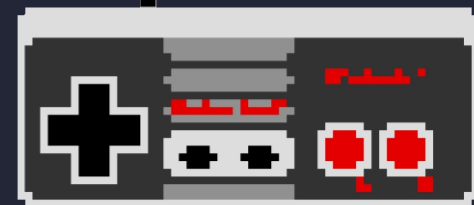
¿Qué es MySQL y MariaDB? Explique si son iguales o diferentes



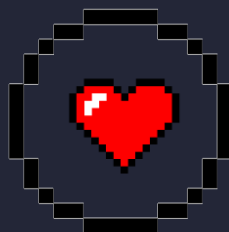
MySQL es un sistema de gestión de bases de datos relacionales de código abierto. MariaDB es un sistema de base de datos proveniente de MySQL pero con licencia, son diferentes por que ambos son autónomos.



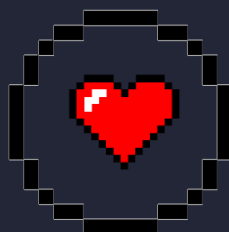
# Parte Teorica



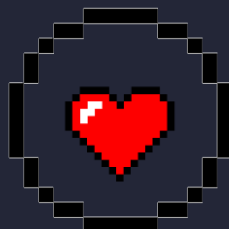
```
SELECT NOMBRE_CLIENTE,  
MAX(TOTAL) AS VENTAS_TOTALES  
FROM VENTAS  
GROUP BY NOMBRE_CLIENTE
```



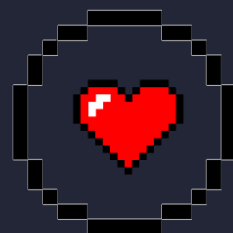
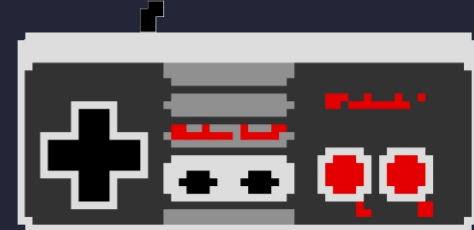
Que es una función de agregación? Adicionalmente muestre un ejemplo



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



# Parte Teórica



Para que sirve el comando USE

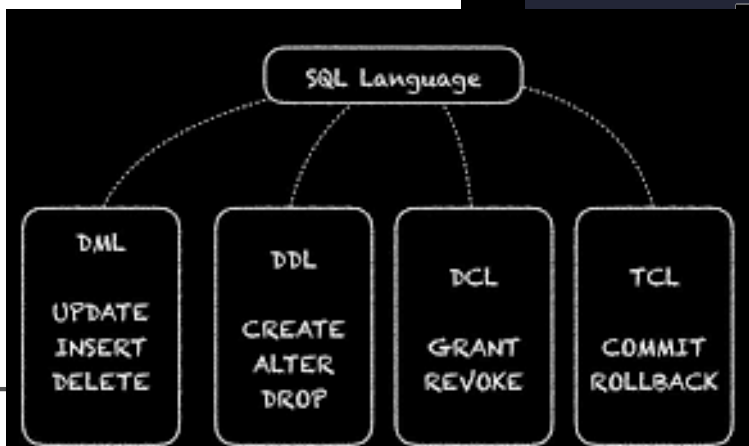
La sentencia USE db\_name indica a MySQL que use la base de datos

Crear una consulta DML

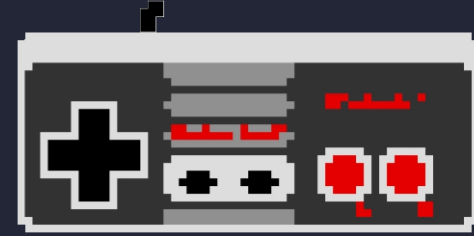
```
insert into estudiante (nombres,
apellidos, edad, genero ) values
('Carlos', 'Perez', 18, 'Masculino');
```

Crear una consulta DDL

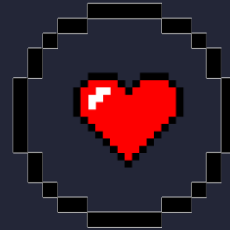
```
create database UNI_Hito2;
```



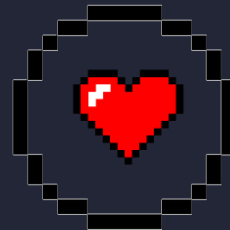
# Parte Practica



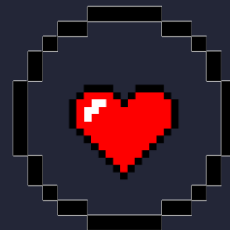
1 row	
	Edad
1	18



Crear una función que permita saber cual es la menor edad de los estudiantes de sexo femenino



```
create function sexo_edad_estudiantes()  
returns integer  
begin  
    return (  
        select MIN(est.edad)  
        from estudiante as est  
        where est.genero='Femenino'  
    );  
end;
```



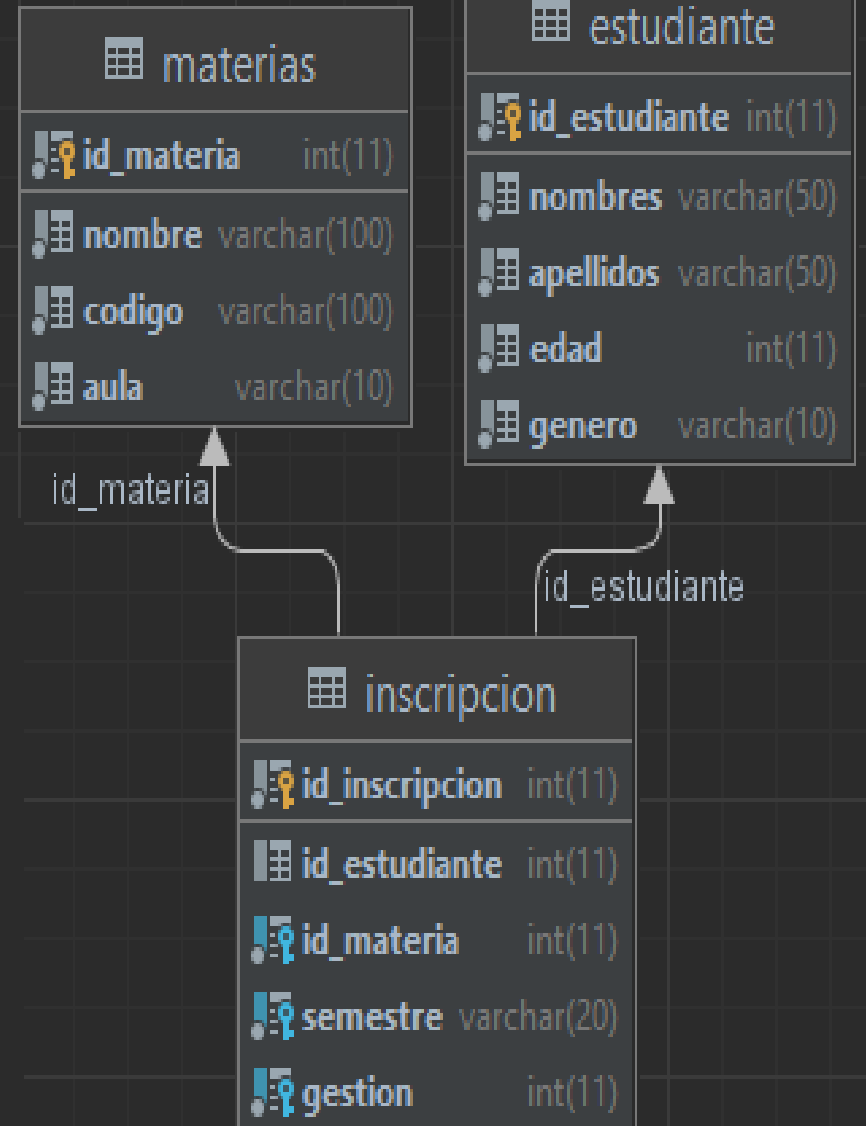
```
select sexo_edad_estudiantes() as Edad;
```

Generar la base de datos para el siguiente modelo ER

```
insert into estudiante (nombres, apellidos, edad, genero ) values
('Carlos', 'Perez', 18, 'Masculino'),
('Fernanda', 'Barrera', 20, 'Femenino'),
('Camilo', 'Cortez', 19, 'Masculino'),
('Carolina', 'Jimenez', 20, 'Femenino');
```

```
insert into materias(nombre, codigo, aula) values
('Calculo', 'CAL-213', 'A-217'),
('Sistemas', 'SIS-121', 'C-214'),
('Administracion', 'ADM-117', 'A-219'),
('Investigacion de Operaciones', 'IOP-120', 'B-220');
```

```
INSERT INTO inscripcion(id_estudiante, id_materia, semestre,
gestion) VALUES
(1, 1, '2do semestre', 2021),
(2, 2, '5to semestre', 2022),
(3, 2, '5to semestre', 2022),
(4, 4, '3er semestre', 2020);
```



Mostrar los estudiantes  
(nombres y apellidos) y  
el nombre de la materia  
de la gestión 2022

```
select est.nombres, est.apellidos,  
m.nombre  
from estudiante as est  
inner join uni_hito2.inscripcion i on  
est.id_estudiante = i.id_estudiante  
inner join materias m on i.id_materia =  
m.id_materia  
where i.gestion=2022;
```



	nombres	apellidos	nombre
1	Fernanda	Barrera	Sistemas
2	Camilo	Cortez	Sistemas



# Mostrar los nombres y apellidos de los estudiantes inscritos en la materia SIS-121

```
create function materiasis(materia
varchar(20), codigo varchar(20))
returns bool
begin
    return materia=codigo;
end;
```

```
select est.nombres, est.apellidos, m.nombre
from estudiante as est
inner join uni_hito2.inscripcion i on
est.id_estudiante = i.id_estudiante
inner join materias m on i.id_materia =
m.id_materia
where materiasis(m.codigo, 'SIS-121');
```

	nombres	apellidos	nombre
1	Fernanda	Barrera	Sistemas
2	Camilo	Cortez	Sistemas

Mostrar el nombre, apellido 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

```
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('femenino', 19)%2=0;
```

	id_estudiante	nombres	apellidos	edad	genero
1	1	Carlos	Perez	18	Masculino
2	2	Fernanda	Barrera	20	Femenino
3	3	Camilo	Cortez	19	Masculino
4	4	Carolina	Jimenez	20	Femenino

# CREAR UNA FUNCION QUE COMPARE EL NOMBRE Y EL APELLIDO

```
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 estudiante as est
where comparenomb(est.nombres,
est.apellidos, 'Carlos', 'Perez');
```

id_estudiante		nombres	
1		Carlos	
apellidos		edad	
Perez		18	
genero		Masculino	