

# Evaluación continua

---

Martin Josue Mamani Pilco

# 1. ¿A que se refiere cuando hablamos de bases de datos relacionales y no relacionales

Las bases de datos relacionales son las que utilizan un lenguaje SQL, suelen ser utilizadas por empresas o entidades que no tienen un tráfico masivo de datos. Además, este tipo de bases de datos utilizan tablas para almacenar los registros.

Las bases de datos no relacionales no utilizan un lenguaje SQL, suelen ser utilizadas por redes sociales o juegos en línea por el tráfico masivo de datos que se suelen manejar. A diferencia del anterior este tipo de bases de datos utiliza documentos para almacenar los registros, prescindiendo de las tablas.

## 2. ¿Qué es MySQL y MariaDB? Explique si son lo mismo o sin diferentes.

Estas bases de datos en un inicio fueron lo mismo, siendo MariaDB una copia de MySQL con una licencia totalmente free a diferencia de MySQL que cuando fue adquirido por Oracle cambio a una versión gratuita y una versión de paga. Hoy en dia MariaDB es una base de datos totalmente independiente y separada de MySQL.

### 3. ¿Que es una función de agregación? Adicionalmente muestra un ejemplo.

Una función de agregación es la que viene predeterminada en el sistema AVG, SUM, MAX, MIN. Esta función sólo puede usarse en la cláusula SELECT.

```
select count(est.id_est)  
from estudiantes as est;
```

## 4. Responda las siguientes preguntas

- Para qué sirve el comando USE

Sirve para utilizar una base de datos existente.

- Crear una consulta DML.

```
select *  
from estudiantes;
```

- Crear una consulta DDL.

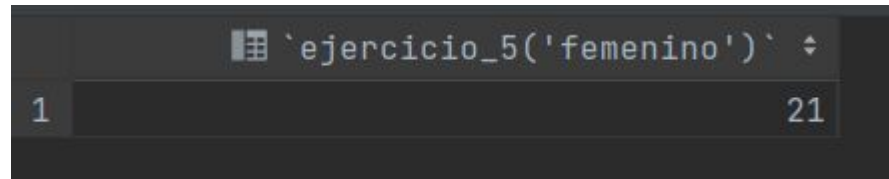
```
create table materias  
    (id mat int auto_increment primary key not null,  
nombre mat varchar(100),  
cod mat varchar(100));
```

## 5. Crear una funcion que permita saber cual es la menor edad de los estudiantes del sexo femenino.

- La función recibe un parámetro (género).
- La función debe retornar un numero (integer).
- La función debe retornar una variable de tipo numero (usar declare).
- Adjuntar la consulta SQL generada y una imagen de su correcto funcionamiento.

```
create function ejercicio_5(genero varchar(20))
returns int
begin
    declare edad int;
    set edad = (
        select min(est.edad)
        from estudiantes as est
        where est.genero = genero
    );
    return edad;
end;
```

```
select ejercicio_5('femenino');
```

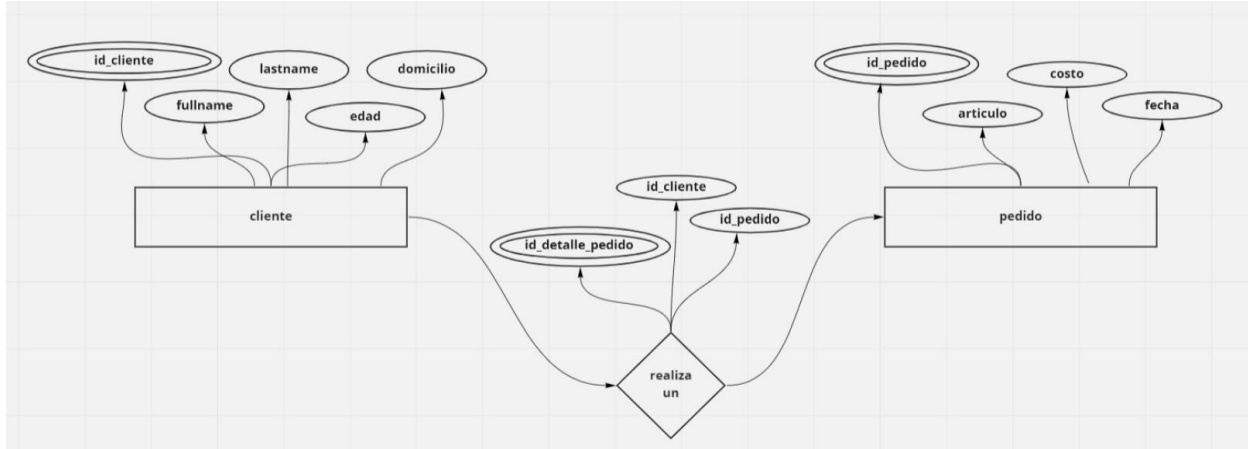


The screenshot shows a SQL query execution result. The query is ``ejercicio_5('femenino')``. The result is a single row with the value 21.

	<code>`ejercicio_5('femenino')`</code>
1	21

## 6. Generar la base de datos para el siguiente diagrama ER.

- Adjuntar la consulta SQL generada
- Adjuntar una imagen del modelo lógico general (Diagrama generado desde DATAGRIP)

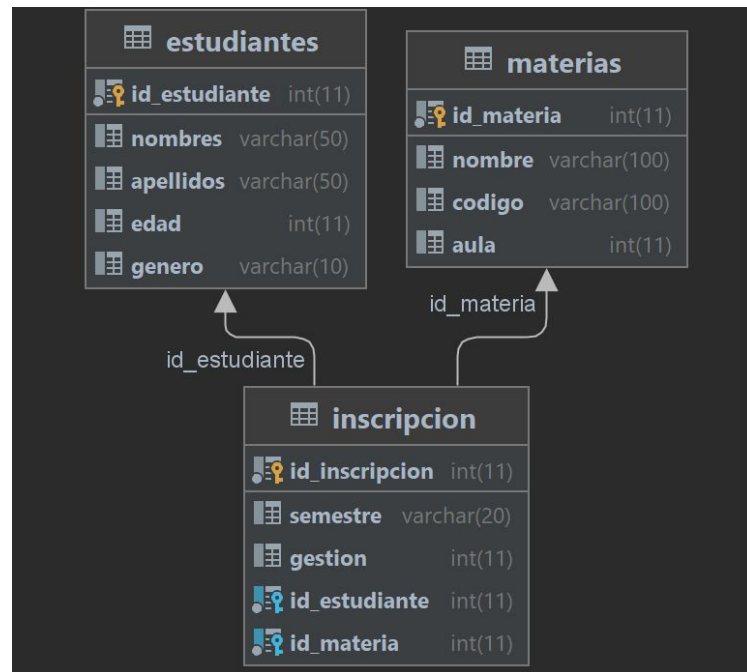




```

create table estudiantes(
    id_estudiante int auto_increment primary key not null,
    nombres varchar(50),
    apellidos varchar(50),
    edad int,
    genero varchar(10)
);
create table materias(
    id materia int auto_increment primary key not null,
    nombre varchar(100),
    codigo varchar(100),
    aula int
);
create table inscripcion(
    id inscripcion int auto_increment primary key not null,
    semestre varchar(20),
    gestion int,
    id estudiante int not null,
    id materia int not null,
    foreign key (id estudiante) references estudiantes (id estudiante),
    foreign key (id materia) references materias (id materia)
);




```



## 7. Mostrar los estudiantes (nombres y apellidos) y el nombre de la materia, inscritos en la gestión 2022.

- Debe de utilizar la base de datos del anterior ejercicio.
- Si tiene que relacionar tablas, utilizar JOINS.
- Adjuntar la consulta SQL generada.
- Adjuntar una imagen (captura de pantalla) del correcto funcionamiento.

```
select est.nombres, est.apellidos, m.nombre
from estudiantes as est
inner join 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	Andrea	Arias Ballesteros	Matematica discreta
2	Santos	Montes Valenzuela	Fisica Basica




8. Mostrar los nombres y apellidos de los estudiantes inscritos en la materia SIS-121, 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

- Debe de utilizar la base de datos del anterior ejercicio (Diseño ER de la imagen).
- Si tiene que relacionar tablas, utilizar JOINS.
- La función retorna un valor booleano.
- Adjuntar la consulta SQL generada.
- Adjuntar una imagen (captura de pantalla) del correcto funcionamiento.

```
create function ejercicio_8(materia varchar(15), codMateria varchar(15))  
returns boolean  
begin  
    return materia = codMateria;  
end;
```

```
select est.nombres, est.apellidos, m.nombre  
from estudiantes as est  
inner join inscripcion i on est.id_estudiante = i.id_estudiante  
inner join materias m on i.id_materia = m.id_materia  
where ejercicio_8(m.codigo, 'SIS-121');
```

	 nombres	 apellidos	 nombre
1	Miguel	Gonzales Veliz	Urbanismo y Diseno
2	Sandra	Mavir Uria	Matematica discreta
3	Andrea	Arias Ballesteros	Matematica discreta
4	Santos	Montes Valenzuela	Fisica Basica

9. 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 utilizar la base de datos del anterior ejercicio (Diseño ER de la imagen).
- Si tiene que relacionar tablas utilice JOINS
- Debe de crear una funcion que sume las edades (recibir como parametro el sexo y la edad)
- La función creada debe ser utilizada en la cláusula WHERE
- Adjuntar la consulta SQL generada
- Adjuntar una imagen del correcto funcionamiento

```

create or replace function ejercicio_9(genero varchar(20), edad int)
returns boolean
begin
    declare aux int default 0;
    select sum(est.edad) into aux
    from estudiantes as est
    where est.genero = genero
    and est.edad > edad;
    return aux;
end;

```

```

select e.nombres, e.apellidos, i.semestre
from inscripcion as i
inner join estudiantes e on i.id estudiante = e.id estudiante
inner join materias m on i.id materia = m.id materia
where ejercicio_9('femenino',20) % 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






10. Crear una función sobre la tabla estudiantes que compara el nombre y apellidos (si existe este nombre y apellido mostrar todos los datos del estudiante)

- Debe de utilizar la base de datos del ejercicio anterior (diagrama ER)
- Si tiene que relacionar, utilizar JOINS
- La función devuelve un booleano
- La función debe recibir el nombre y sus apellidos
- Adjuntar la consulta SQL generada
- Adjuntar una imagen del correcto funcionamiento



```
create or replace function ejercicio_10(nombres varchar(20), apellidos
varchar(20), nombresAcom varchar(20), apellidosAcom varchar(20))
returns bool
begin
    declare respuesta bool default false;
    set respuesta = (nombres = nombresAcom and apellidos = apellidosAcom) ;
    return respuesta;
end;
```

```
select est.*
from estudiantes as est
where ejercicio_10(est.nombres, est.apellidos, 'Santos', 'Montes
Valenzuela');
```

	 id_estudiante	 nombres	 apellidos	 edad	 genero
1	5	Santos	Montes Valenzuela	24	masculino