





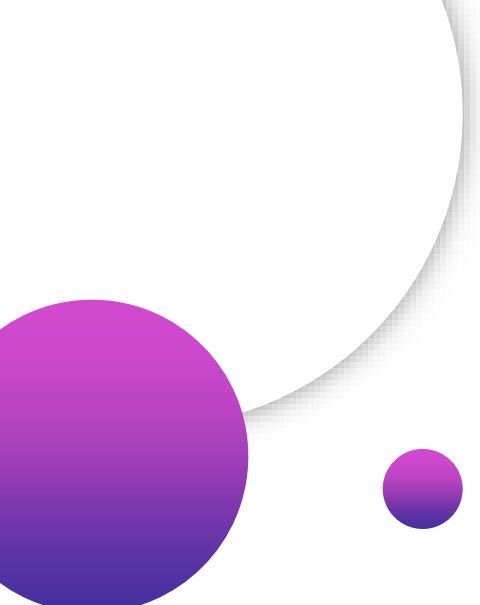
BASE DE DATOS II



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


Es una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas. Las tablas se utilizan para guardar información sobre los objetos que se van a representar en la base de datos.






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

Una base 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. Por ello también se les suele llamar **NoSQL**





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

MySQL.- Es un sistema de gestión de bases de datos relacionales de código abierto creado por MySQL AB para el desarrollo de aplicaciones web. Proporciona soporte para que los desarrolladores web puedan gestionar datos con facilidad utilizando tablas.


MariaDB.- Los desarrolladores construyeron el RDBMS para conservar la estructura y las características de MySQL. Temían que la adquisición del sistema por parte de Oracle, la corporación detrás de la base de datos Oracle, que era el mayor competidor de MySQL en ese momento, pusiera en peligro la base de datos.

- A pesar de tener una estructura y funcionalidad similares, hay varias diferencias clave entre MySQL y MariaDB.
- MariaDB sigue siendo completamente de código abierto (open-source), mientras que MySQL tiene ahora módulos de código cerrado. En general, MariaDB ofrece un mejor rendimiento, es más rápido y más ligero que MySQL.



¿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, sobre un conjunto de valores.

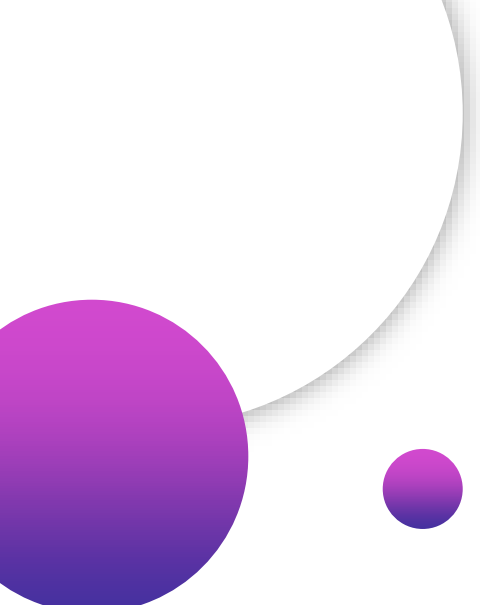




¿Qué llegaría a ser XAMPP, WAMP SERVER o LAMP?

Xampp es un paquete (como un kit de herramientas) para nuevos desarrolladores que contiene, un gestor de base de datos, el servidor apache, lenguajes de programación como php






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

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, sobre un conjunto de valores.


Funciones creados por el BDA se utiliza para definir una función de tabla, fila o escalar de SQL definida por el usuario. Una función escalar devuelve un solo valor cada vez que se invoca y en general es válida cuando una expresión SQL es válida.





¿Para qué sirve el comando USE?

El comando USE **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?

DML :


Permite recuperar, almacenar, modificar, eliminar, insertar y actualizar datos de una base de datos.

- **SELECT:** Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
- **INSERT:** Utilizado para cargar de datos en la base de datos en una única operación.
- **UPDATE:** Utilizado para modificar los valores de los campos y registros especificados
- **DELETE:** Utilizado para eliminar registros de una tabla de una base de datos.

DDL:


Este lenguaje permite definir las tareas de las estructuras que almacenarán los datos.


- **CREATE:** Utilizado para crear nuevas tablas, campos e índices.
- **ALTER:** Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
- **DROP:** Empleado para eliminar tablas e índices.
- **TRUNCATE:** Empleado para eliminar todos los registros de una tabla.
- **COMMENT:** Utilizado para agregar comentarios al diccionario de datos.
- **RENAME:** Tal como su nombre lo indica es utilizado para renombrar objetos.




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

Las funciones integradas en SQL Server son una serie de rutinas almacenadas que reciben una serie de parámetros con los cuales realizan operaciones concretas para retornar un resultado específico.





¿Cómo crear, modificar y cómo eliminar una función?

- Crear una función el código es CREATE FUNCTION
 - Modificar una función es REPLACE
 - Eliminar la función es DROP FUNCTION
- 

crear las tablas y 2 registros para cada tabla para el siguiente modelo ER.
Se sugiere crear una base de datos de nombre POLLOS_COPA y en ella crear las tablas:

- cliente
- detalle_pedido
- pedido

```
CREATE DATABASE POLLOS_COPA;
USE POLLOS_COPA;

CREATE TABLE Cliente
(
    id_cliente INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL ,
    fullname VARCHAR(30),
    lastname VARCHAR(30),
    edad INTEGER,
    domicilio VARCHAR(40)
);

CREATE TABLE DetallePedido
(
    id_detalle_pedido INTEGER AUTO_INCREMENT NOT NULL PRIMARY
    KEY,
    id_pedido INTEGER NOT NULL,
    id_cliente INTEGER NOT NULL,
    FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
);

CREATE TABLE Pedido
(
    id_pedido INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    articulo VARCHAR(30),
    costo FLOAT,
    fecha DATE
);
```

- 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 Detallepedido.id_detalle_pedido AS CODIGO_COMPRA,  
       CONCAT(Cliente.fullname, Cliente.Lastname) AS Nombre_completo,  
       Pedido.articulo, Pedido.costo  
  
FROM DetallePedido  
INNER JOIN Cliente ON Detallepedido.id_cliente = Cliente.id_cliente  
INNER JOIN Pedido ON Detallepedido.id_pedido = Pedido.id_pedido;
```

```
pollos_copa> INSERT INTO Cliente(fullname, lastname, edad, domicilio)  
              VALUES('Alejandro', 'Quiroz Savedra', 21, 'Rio Seco'),  
                     ('Lucia', 'Medina Romo ', 25, 'Cruce Villa Adela')  
[2022-09-11 19:15:52] 2 rows affected in 4 ms
```

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

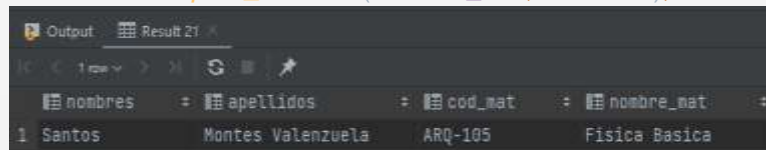
```
select est.nombres, est.apellidos, mat.cod_mat,  
mat.nombre_mat  
from estudiantes as est  
    inner join inscripcion as ins on est.id_est =  
ins.id_est  
    inner join materias as mat on ins.id_mat =  
mat.id_mat  
where mat.cod_mat = 'ARQ-105';
```

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

```
create function compara_materias(cod_mat varchar(20), nombre_mat varchar(20))  
    returns boolean  
begin  
    declare respuesta boolean;  
    if cod_mat = nombre_mat  
    then  
        set respuesta=1;  
    end if;  
    return respuesta;  
end;
```

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

```
select est.nombres, est.apellidos, mat.cod_mat,  
mat.nombre_mat  
from estudiantes as est  
    inner join inscripcion as ins on est.id_est =  
ins.id_est  
    inner join materias as mat on ins.id_mat =  
mat.id_mat  
where compara_materias(mat.cod_mat,'ARQ-105');
```



nombres	apellidos	cod_mat	nombre_mat
1 Santos	Montés Valenzuela	ARQ-105	Fisica Basica

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

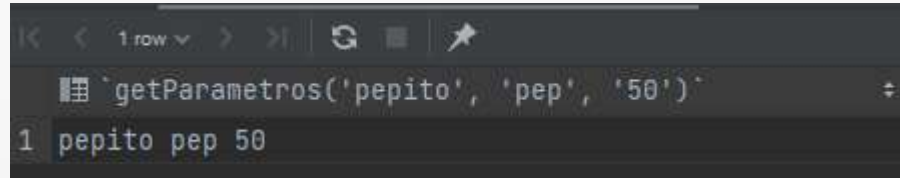
```
SELECT avg(est.edad)
FROM estudiantes AS est
      inner join inscripcion ins on est.id_est =
ins.id_est
      inner join materias mat on ins.id_mat =
mat.id_mat
where est.sexo = 'femenino' and mat.cod_mat =
'ARQ-104';
```

```
CREATE OR REPLACE FUNCTION get_avg_est(genero varchar(10),
codMateria varchar(10))
  RETURNS INTEGER
BEGIN
  declare avgEdad int default 0;
  SELECT avg(est.edad) into avgEdad
  FROM estudiantes AS est
        inner join inscripcion ins on est.id_est = ins.id_est
        inner join materias mat on ins.id_mat = mat.id_mat
  where est.sexo = genero and mat.cod_mat = codMateria;
  return avgEdad;
END;
```

Crear una función que permita concatenar 3 cadenas.

- La función recibe 3 parámetros.
- Si las cadenas fuesen: ■ Pepito ■ Pep ■ 50
- La salida debería ser: (Pepito), (Pep), (50)
- La función creada utilizarlo en una consulta SQL.

```
create function getParametros(par1 varchar(20), par2 varchar(20), par3  
varchar(20))  
  returns varchar(60)  
begin  
  declare resultado varchar(60);  
  set resultado = CONCAT(par1, ' ', par2, ' ', par3);  
  return resultado;  
end;
```



```
< < 1 row > > | [refresh] [clear] [star]  
[table icon] `getParametros('pepito', 'pep', '50')`  
1 pepito pep 50
```


Crear una función de acuerdo a lo siguiente:

- Mostrar el nombre, apellidos, edad y el semestre de todos los estudiantes que estén inscritos.
- Siempre y cuando la suma de las edades del sexo femenino (también puede ser 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 or replace function get_genero_edad(genero varchar(10),
edad int)
    returns boolean
begin
    declare resultado int default 0;
    declare ifRes boolean;

    select sum(est.edad) into resultado
    from estudiantes as est
    where est.sexo=genero;

    if resultado%2=0 and resultado>edad
    then
        set ifRes=1;
    end if;
    return ifRes;
end;
```

```
select est.nombres, est.apellidos, i.semestre
from estudiantes as est
inner join inscripcion i on est.id_est = i.id_est
where get_genero_edad('masculino', 22);
```

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). 7
- La función devuelve un boolean.
- La función debe recibir 4 parámetros, nombres y apellidos.
- Similar al siguiente ejemplo.
- La función debería ser usada en la cláusula WHERE.
- El objetivo es buscar a estudiantes a través de sus nombres y apellidos.

```
create or replace function comparaNombre(nombre varchar(50),apellido
varchar(50),nombreEst varchar(50), apellidoEst varchar(50))
  returns boolean
begin
  declare resultado boolean;
  if nombre=nombreEst and apellido=apellidoEst
  then
    set resultado=1;
  end if;
  return resultado;
end;

select est.*
from estudiantes as est
where comparaNombre(est.nombres,est.apellidos,'Joel','Aubiri Mondar')
```