

Tarea de defensa

Hito 2

Martin Josue Mamani Pilco

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

Parte teórica

Manejo de conceptos

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

Las bases de datos relacionales son las que se conforman por tablas para almacenar los registros, por lo general utilizados por empresas con un tráfico de datos no muy grande. Además, este tipo de bases de datos utilizan un lenguaje SQL.

ORACLE®
DATABASE



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

Las bases de datos no relacionales son las que no utilizan tablas, ya que suelen manejar un tráfico masivo de datos. No utilizan un lenguaje SQL a diferencia de las bases de datos relacionales.



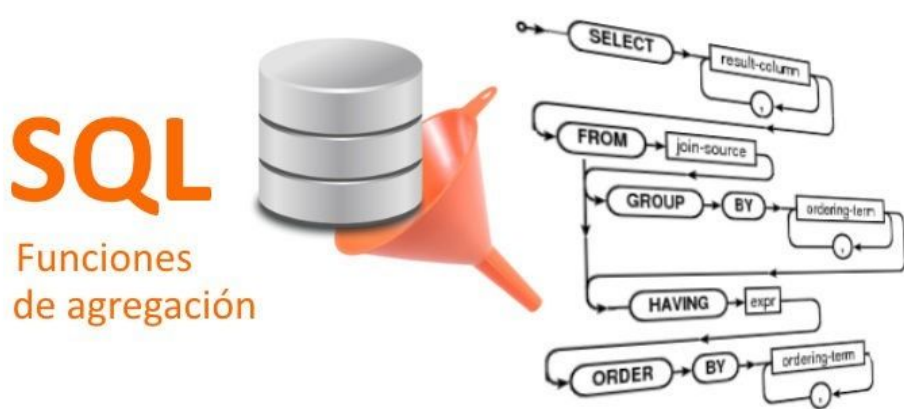
3. ¿Qué es MySQL y MariaDB? Explique si existen diferencias o son iguales.

Los lenguajes MySQL y MariaDB tienen la misma base, puesto que MariaDB es una copia de MySQL que puede ser utilizado de manera gratuita, ya que para utilizar MySQL se debe pagar una licencia.



4. ¿Qué son las funciones de agregación?

Las funciones de agregación son las que vienen predeterminadas en un lenguaje como AVG, SUM, MAX, entre otros.



5. ¿Qué llegaría a ser XAMPP?

Es una herramienta que contiene un paquete de software libre, que funciona como un servidor local de varios lenguajes. La palabra es un abreviatura donde X es multiplataforma, A es Apache, M es MySQL y las dos p son PHP y Perl.



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

Las funciones tienen funciones predeterminadas y solo realizan una acción ya establecida, en cambio las funciones creadas por el DBA pueden tener diferentes parámetros y realizar las acciones que el usuario programe. Además de que las funciones creadas por el DBA pueden utilizarse en la cláusula Where, a diferencia de las funciones de agregación que solo pueden ser utilizadas en la cláusula Select.

7. ¿Para qué sirve el comando USE?

El comando USE sirve para posicionarse en una base de datos existente

USE GRANJA;

- comando para poder seleccionar y usar la base datos.



8. ¿Que es DML y DDL?

DML es Data Manipulation Language y abarca elementos para manipular datos como Select, Insert, Delete entre otros

DDL es Data Definition Language y permite definir las estructuras que almacenarán los datos, disponiendo de sentencias como Create, Alter y Drop.



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

En una función creada por el usuario el nombre puede ser cualquiera, pero no debe tener espacios, si desea recibir parámetros debe declarar las variables pertinentes entre paréntesis. Posteriormente debe especificar que tipo de resultado desea con el return, ya sea un integer, un varchar u otra variable. Después se debe poner el desarrollo dependiendo que es lo que el usuario desea realizar dentro de un begin y un end, cerrando la función.

10. ¿Como crear, modificar y cómo eliminar una función?

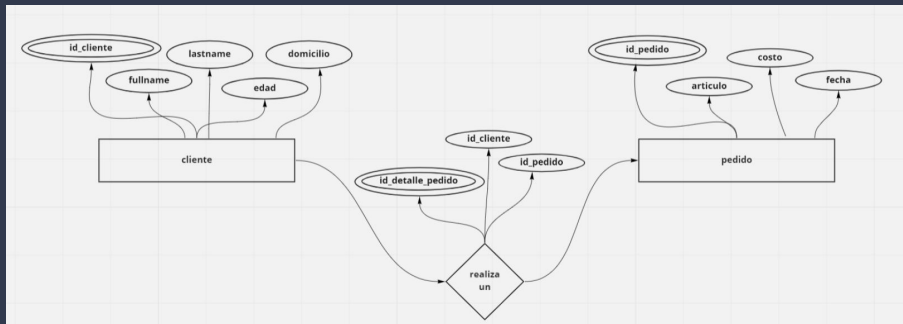
Para crear una función se debe colocar create seguido del nombre que se desea ponerle a una función, para luego poner return con el tipo de variable que se desea retornar y el procedimiento dentro de begin y end.

Para modificar utilizamos replace en vez de create, luego el procedimiento que sigue es el mismo.

Para eliminar se debe poner drop function y el nombre de la función.

Parte práctica

11. 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 int primary key not null,
    fullname varchar(30),
    lastname varchar(30),
    edad int,
    domicilio varchar(100)
);
insert into cliente (id_cliente, fullname, lastname, edad, domicilio)
values (1, 'Alex', 'Ticona', 20, 'domicilio1'),
       (2, 'Paul', 'Magne', 18, 'domicilio2'),
       (3, 'Carlos', 'Quisbert', 23, 'domicilio3'),
       (4, 'Luz', 'Lopez', 19, 'domicilio4');
create table pedido(
    id pedido int primary key not null,
    articulo varchar(20),
    costo int,
    fecha date
);
insert into pedido (id pedido, articulo, costo, fecha)
values (1, 'artprimero', 30, '10/08/20'),
       (2, 'artsegundo', 15, '03/10/19'),
       (3, 'arttercero', 40, '21/06/21'),
       (4, 'artcuarto', 60, '18/04/22');
create table detalle_pedido(
    id detalle_pedido int primary key not null,
    id cliente int,
    id pedido int,
    foreign key (id cliente) references cliente (id cliente),
    foreign key (id pedido) references pedido (id pedido)
);
insert into detalle_pedido(id_detalle_pedido, id_cliente, id_pedido)
values (1, 1, 1),
       (2, 2, 2),
       (3, 3, 3),
       (4, 4, 4);
```

12. Crear una consulta SQL en base al ejercicio anterior

```
select cli.fullname, cli.lastname,  
cli.edad  
from cliente as cli  
inner join detalle pedido dp on  
cli.id_cliente = dp.id_cliente  
inner join pedido p on dp.id_pedido  
= p.id_pedido  
where costo >10;
```

	fullname	lastname	edad
1	Alex	Ticona	20
2	Paul	Magne	18
3	Carlos	Quisbert	23
4	Luz	Lopez	19

13. Crear una función que compare dos códigos de materia

```
create or replace function
comparaMaterias(codigo varchar(20), materia
varchar(20))
returns boolean
begin
    declare flag boolean;
    set flag = 0;
    if(codigo = materia)
    then
        set flag=1;
    end if;
    return (flag);
end;

select est.id est, est.nombres, est.apellidos,
mat.nombre mat, mat.cod mat
from estudiantes as est
inner join inscripcion i on est.id est =
i.id est
inner join materias mat on i.id mat = mat.id mat
where comparaMaterias(mat.cod mat, 'ARQ-105');
```

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

14. 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

```
create function prom_edades(genero varchar(10))  
returns int  
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 cod_mat = 'ARQ-104' and est.sexo =  
genero);  
end;  
select prom_edades('femenino');
```

`prom_edades('femenino')`	
1	23

15. Crear una función que permita concatenar 3 cadenas

```
create or replace function concatenar(param1  
varchar(20), param2 varchar(20), param3  
varchar(20))  
returns varchar(100)  
begin  
    declare concat varchar(80);  
    set concat = concat(param1, ' - ', param2, '  
- ', param3);  
    return (concat);  
end;
```

```
select concatenar('Pepito','Perez','50');
```

```
1 `concatenar('Pepito','Perez','50')`  
1 Pepito - Perez - 50
```

16. 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 crear una función que sume las edades (recibir como parametro el sexo y la edad)
- La función creada anteriormente debe utilizarse en la consulta SQL (cláusula WHERE)

```
create or replace function sum edades gen(sexo
varchar(15), edad int)
returns int
begin
    return (select sum(est.edad)
            from estudiantes as est
            where est.sexo = sexo 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 sum edades gen('masculino',22) % 2 = 0 and
est.sexo = 'masculino' and est.edad > 20;
```

	nombres	apellidos	semestre
2	Joel	Adubiri Mondar	3er Semestre
3	Santos	Montes Valenzuela	5to Semestre

17. Crear una función de acuerdo a lo siguiente

- Crear una función sobre la tabla estudiantes que compare 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
- La función debería ser usada en la cláusula WHERE

```
create or replace function comparaNombresLastnem(nombres
varchar(50), apellidos varchar(50))
returns boolean
begin
    declare flag boolean;
    declare nombres y apellidos varchar(100);
    set nombres y apellidos = (
        select est.nombres
        from estudiantes as est
        where est.nombres = nombres and est.apellidos =
apellidos);
    if(nombres y apellidos is not null)
    then
        set flag = 1;
    else
        set flag = 0;
    end if;
    return (flag);
end;
select est.*
from estudiantes as est
where comparaNombresLastnem('Miguel','Gonzales Veliz') and
est.nombres = 'Miguel' and est.apellidos = 'Gonzales Veliz';
```

id_est	nombres	apellidos	edad	gestion	fono	email	direccion	sexo
1	Miguel	Gonzales Veliz	20	<null>	2832115	miguel@gmail.com	Av. 6 de Agosto	masculino