




# **EVALUACION PROCESUAL HITO 3**

**BASE DE DATOS II  
LUIS ALVAREZ MEDINA**



# MANEJO DE CONCEPTOS

**01**

## **DEFINA QUE ES LENGUAJE PROCEDURAL EN MySQL**

El lenguaje procedural es la capacidad que tiene el servidor MySQL para crear y utilizar procedimientos almacenados y funciones definidas por el usuario, estos procedimientos, funciones pueden ser llamados desde cualquier lugar dentro de la base de datos MySQL estos pueden recibir parámetros de entrada y devolver valores de salida

**02**

## **DEFINA QUE ES UNA FUNCION EN MySQL**

La función en MySQL es un objeto de programación que puede ser definido por el usuario y utilizarlo para realizar operaciones específicas donde estos pueden recibir uno o varios valores de entrada los cuales serán procesados y devolverán un resultado

## QUE COSAS CARACTERISTICAS DEBE TENER UNA FUNCION EXPLIQUE? EL NOMBRE, EL RETURN, PAREMETROS, ETC.

Nombre: el nombre debe de describir el proceso que ara la función.

Parámetros: son el o los valores de entrada que serán procesados dentro de la función, estos tiene que tener un tipo de dato ya sea cadena o numérico.

Returns: este es donde pondremos el tipo de dato que devolverá esta función será numérico o cadena.

Procesos: dentro de este se cuenta las variables que declaremos o los bucles que usemos, condónales, etc, están todos los procedimientos de la función las cuales devolverán un valor de salida.

Return: este es el encargado de devolver dicho valor de salida donde de acuerdo al tipo de valor que nosotros le hallamos dado en el returns devolverá un numérico o una cadena.

```
create or replace function quita letra 1 en 1(cadena1 varchar(100))
returns TEXT
begin
    declare resp text default '';
    declare puntero varchar(100);
    declare cont int default char_length(cadena1);
    repeat
        set puntero = substr(cadena1,-cont,cont);
        set resp = concat(resp,puntero,',');
        set cont = cont-1;
    until cont = 0 end repeat;
    return resp;
end;
```

04

## ¿CÓMO CREAR, MODIFICAR Y CÓMO ELIMINAR UNA FUNCIÓN? ADJUNTE UN EJEMPLO DE SU USO.

```
create or replace function nombre(Parametros)
returns TEXT
begin
    declare resp text default "";
    return resp;
end;
```

Para crear o hacer modificaciones en una funcion utilizamos `create or replace function`, donde `create` lo utilizamos para crear si la funcion no existe y `replace` para remplazar si la funcion existe.

```
Drop function nombre;
```

Para eliminar una funcion utilizamos `drop function` y el nombre de dicha funcion.

05

## PARA QUE SIRVE LA FUNCION CONCAT Y COMO FUNCIONA EN MySQL

- ¿Crear una función que muestra de la función CONCAT?
- La función debe concatenar 3 cadenas.

La función **CONCAT**(paremtros), la funcion cancat sirve par unir una o varias cadenas, números en una sola cadena.

```
create or replace function concatenador(cadena1 varchar(100),cadena2 varchar(100),cadena3 varchar(100))
returns text
begin
  declare resp text default '';
  set resp = concat(cadena1,cadena2,cadena3);
  return resp;
end;

select concatenador('Luis','Alvarez','Medina');
```

```
📋 `concatenador('Luis','Alvarez','Medina')`
```

```
1 LuisAlvarezMedina
```

06

## Para qué sirve la función SUBSTRING y como funciona en MYSQL

- ¿Crear una función que muestre el uso de la función SUBSTRING?
- La función recibe un nombre completo INPUT: XIMENA CONDORI MAR
- La función solo retorna el nombre. OUTPUT XIMENA

La función SUBSTRING sirve para hacer cortes dentro de una cadena que ingresemos y tiene esta forma

`substr(cadena,posición,cantidad)`, la posición es en donde se encuentra la parte que deseamos cortar de la cadena si ingresamos un número positivo contara de izquierda a derecha y si ingresamos un número negativo ira de derecha a izquierda., la cantidad es cuantas letras queremos que se mantenga.

```
create or replace function utilizandoSubstring(cadena1 varchar(100))
returns text
begin
    declare resp text default "";
    set resp = substr(cadena1,1,6);
    return resp;
end;

select utilizandoSubstring('XIMENA CONDORI MAR');
```

```
create or replace function utilizandoSubstring(cadena1 varchar(100))
returns text
begin
    declare resp text default "";
    set resp = substr(cadena1,-18,6);
    return resp;
end;

select utilizandoSubstring('XIMENA CONDORI MAR');
```

```
`utilizandoSubstring('XIMENA CONDORI MAR')`
1 XIMENA
```

07

## Para qué sirve la función STRCMP y como funciona en MYSQL

- ¿Crear una función que muestre el uso de la función STRCMP?
- La función debe comparar 3 cadenas. Y deberá determinar si dos de ellas son iguales.

La función strcmp() sirve para comparar dos cadenas sin importar si es mayúscula o minúscula, este retornara 1 o -1 si las dos cadenas no son iguales y retornara 0 si los dos valores son iguales.

```
create or replace function utilizandostrcmp(cadena1 varchar(100),cadena2 varchar(100),cadena3 varchar(100))
returns text
begin
  declare resp text default '';
  if(strcmp(cadena1,cadena2)=0 or strcmp(cadena1,cadena3)=0 or strcmp(cadena2,cadena3)=0) then
    set resp = 'si hay son iguales dos cadenas';
  else
    set resp = 'no hay igualdad entre las cadenas';
  end if;
  return resp;
end;

select utilizandostrcmp('DBA II','DBB I','dba ii');
```

```
1 `utilizandostrcmp('DBA II','DBB I','dba ii')`
1 si hay son iguales dos cadenas
```



# Para qué sirve la función CHAR\_LENGTH y LOCATE y como funciona en MYSQL

■¿Crear una función que muestre el uso de ambas funciones?

La función `char_length(cadena)` la utilizamos para contar la cantidad de letras que se encuentran dentro de una cadena incluido los vacíos

La función `locate(buscamos,cadena,posicion)` la utilizamos para ver en que posición se encuentra lo que buscamos, la posición es desde donde nosotros queremos que busque.

```
create or replace function utilizandoLocateCharlegth(Buscar varchar(100),cadena1 varchar(100))
returns text
begin
  declare resp text default '';
  declare cont text default 0;
  declare pos text default 0;
  set pos = locate(Buscar,cadena1);
  if(pos>0) then
    set cont = char_length(Buscar);
    set resp = concat('la palabra que bsucas se encuentra en la posicion : ',pos,' y es de ', cont, ' palabras');
  end if;

  return resp;
end;
select utilizandoLocateCharlegth('Luis','Jose Luis Mamani Apaza');
```

```
`utilizandoLocateCharlegth('Luis','Jose Luis Mamani Apaza')`
1 la palabra que bsucas se encuentra en la posicion : 6 y es de 4 palabras
```

**09**

**¿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 agresión son las que viene con el servidor de MySQL, como puede ser count(), sum, max, etc.

Las funciones creados por el DBA son las funciones que los usuarios creamos para una tarea en específico.

**10**

**¿Busque y defina a qué se referirá cuando se habla de parámetros de entrada y salida en MySQL?**

Parametros de entrada: estos son los datos que ingresan una función ya sea numérico o tipo cadena, Iso caules pasan por el procedimiento de la función a al que le damos estos valores.

Salida: son esos datos resultantes de la función tras haber recibido uno o varios parámetros el cual se almacena en una variable de tipo cadena o tipo numéricos.



# **PARTIE PRACTICA**

# 11 Crear la siguiente base de datos y sus registros.

```
create table estudifiantes(
```

```
id_est int auto_increment primary key,  
nombres varchar(100),  
apellidos varchar(100),  
edad int,  
fono int,  
email varchar(100),  
direccion varchar(100),  
seco varchar(10)
```

```
);
```

```
create table inscripcion(
```

```
id_ins int auto_increment primary key,  
semestre varchar(20),  
gestion int,  
id_est int,  
id_mat int,  
  
foreign key (id_est) references estudifiantes(id_est),  
foreign key (id_mat) references materias(id_mat)
```

```
);
```

```
insert into inscripcion(semestre, gestion, id_est, id_mat)
```

```
values ('1er Semestre',2018,1,1),  
('2do Semestre',2018,1,2),  
('1er Semestre',2019,2,4),  
('2do Semestre',2019,2,3),  
('2do Semestre',2020,3,3),  
('3er Semestre',2020,3,1),  
('4to Semestre',2021,4,4),  
('5to Semestre',2021,5,5);
```

```
create table materias(
```

```
id_mat int auto_increment primary key,  
nombre_mat varchar(100),  
cod_mat varchar(100)
```

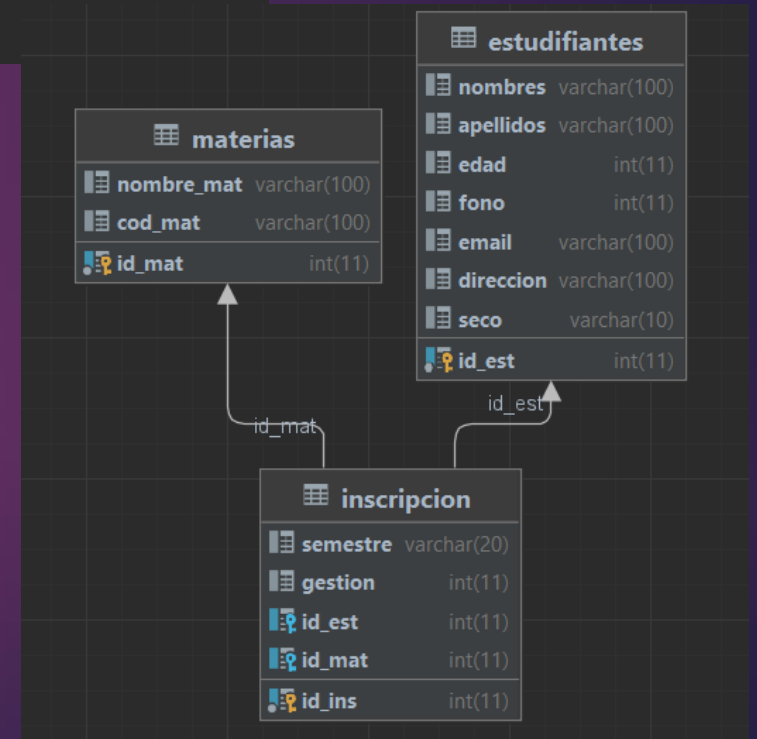
```
);
```

```
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 estudifiantes(nombres, apellidos, edad, fono, email, direccion, seco)
```

```
values ('Miguel','Gonzales Veliz',20,2832115,'miguel@gmail.com','Av. 6 de Agosto','masculino'),  
('Sandra','Mavir Uria',25,2832115,'sandra@gmail.com','Av. 6 de Agosto','femenino'),  
('Joel','Adubiri Mondar',30,2832115,'joel@gmail.com','Av. 6 de Agosto','masculino'),  
('Andrea','Arias Ballesteros',21,2832115,'andrea@gmail.com','Av. 6 de Agosto','femenino'),  
('Santos','Montes Valenzuela',24,2832115,'santos@gmail.com','Av. 6 de Agosto','masculino');
```



## 12

## Crear una función que genere la serie Fibonacci.

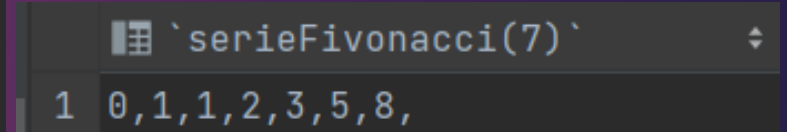
- La función recibe un límite(number)
- La función debe de retornar una cadena.
- Ejemplo para n=7. OUTPUT: 0, 1, 1, 2, 3, 5, 8

```
create or replace function serieFivonacci(numero int)
returns TEXT
begin
    declare respuesta TEXT default "";
    declare cont int default 1;
    declare numR int default 0;
    declare num1 int default 0;
    declare num2 int default 1;

    while(numero >= cont) do
        set respuesta = concat(respuesta,num1,',');
        set numR = num1+num2;
        set num1 = num2 ;
        set num2 = numR ;
        set cont = cont+1;

    end while;
    return respuesta;
end;

select serieFivonacci(7);
```



The screenshot shows a query window with the command ``serieFivonacci(7)`` and its result. The result is a single row with the value `1 0,1,1,2,3,5,8,`.

1	0,1,1,2,3,5,8,

## 13 Crear una variable global a nivel BASE DE DATOS.

- Crear una función cualquiera.
- La función debe retornar la variable global.

```
set @limit = 7;

create or replace function serieFivonacciConVariableGlobal()
returns TEXT
begin
    declare respuesta TEXT default '';
    declare cont int default 1;
    declare numR int default 0;
    declare num1 int default 0;
    declare num2 int default 1;

    while(@limit >= cont) do
        set respuesta = concat(respuesta,num1,',');
        set numR = num1+num2;
        set num1 = num2 ;
        set num2 = numR ;
        set cont = cont+1;

    end while;
    return respuesta;
end;

select serieFivonacciConVariableGlobal();
```

```
`serieFivonacciConVariableGlobal()`
1 0,1,1,2,3,5,8,
```

14

## Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o LOOP).

- **Previamente deberá de crear una función que obtenga la edad mínima de los estudiantes**
- **La función no recibe ningún parámetro.**
- **La función debe de retornar un número.(LA EDAD MÍNIMA).**
- **Si la edad mínima es PAR mostrar todos los pares empezando desde 0 a este ese valor de la edad mínima**
- Si la edad mínima es IMPAR mostrar descendentemente todos los impares hasta el valor 0.**

```
create or replace function ParesImpares()
returns text
begin
  declare resp text default '';
  declare par int default 0;
  declare impar int default EdadMinima();
  if EdadMinima() % 2 = 0 then
    repeat
      if(par % 2 = 0) then
        set resp = concat(resp,par,',');
      end if;
      set par = par+1;
    until par > EdadMinima() end repeat;
  else
    impares: LOOP
    if impar<0 then
      leave impares; /* andate*/
    end if;

    if impar % 2 != 0 then
      set resp= concat(resp,impar,',');
    end if;

    set impar=impar-1; /*seguinomas*/

    iterate impares;
  end loop;
end if;
return resp;
end;
```

```
create or replace function EdadMinima()
returns int
begin
  declare resp int default 0;

  select min(edad) into resp from estudifantes;

  return resp;
end;
```

```
select ParesImpares();
```

```
`ParesImpares()`
1 0,2,4,6,8,10,12,14,16,18,20,
```

15

## Crear una función que determina cuantas veces se repite las vocales.

- La función recibe una cadena y retorna un TEXT.
- Retornar todas las vocales ordenadas e indicando la cantidad de veces que se repite en la cadena.

```
select contadorDeVocales('Taller de base de datos');
```

```
1 | 'contadorDeVocales('Taller de base de datos')'
  |-----|
1 | A: 3 E: 4 I: 0 O: 1 U: 0
```

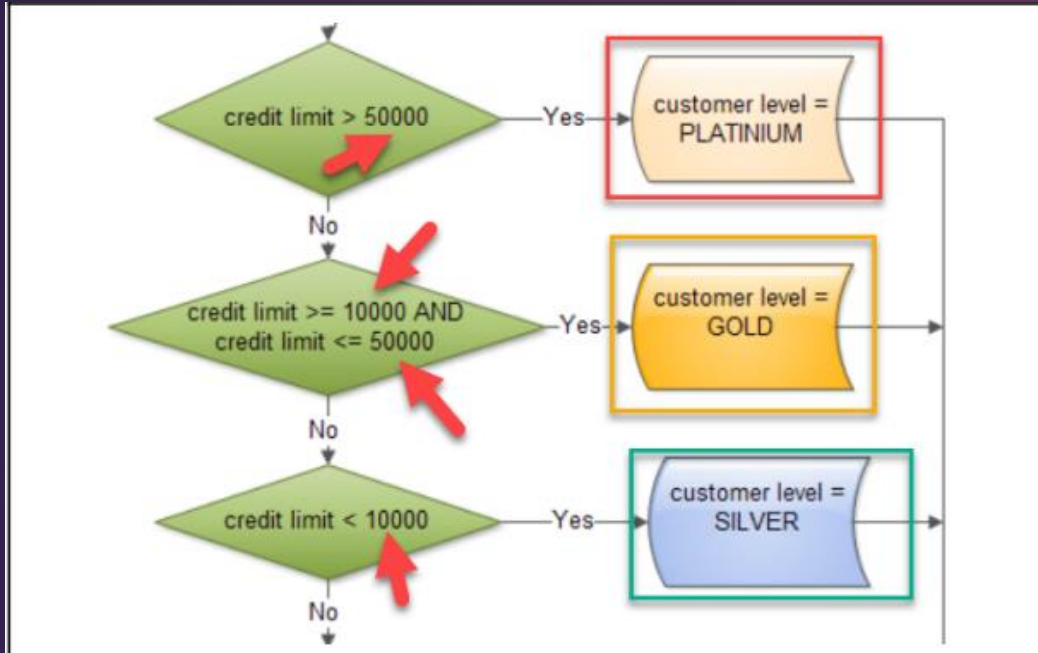
```
create or replace function contadorDeVocales(Cadena varchar(100))
returns TEXT
begin
    declare resp text default "";
    declare num int default char_length(Cadena);
    declare conA int default 0;
    declare conE int default 0;
    declare conI int default 0;
    declare conO int default 0;
    declare conU int default 0;
    if (locate('a', Cadena) > 0 or locate('e', Cadena) > 0 or locate('i', Cadena) > 0 or locate('o', Cadena) > 0 or locate('u', Cadena) > 0) then
        while (num > 0) do
            if substr(Cadena, num, 1) like '%a' then
                set conA = conA + 1;
            end if;
            if substr(Cadena, num, 1) like '%e' then
                set conE = conE + 1;
            end if;
            if substr(Cadena, num, 1) like '%i' then
                set conI = conI + 1;
            end if;
            if substr(Cadena, num, 1) like '%o' then
                set conO = conO + 1;
            end if;

            if substr(Cadena, num, 1) like '%U' then
                set conU = conU + 1;
            end if;
            set num = num - 1;
        end while;
        set resp = concat('A: ', conA, ' E: ', conE, ' I: ', conI, ' O: ', conO, ' U: ', conU);
    else
        set resp = 'No hay Vocales en la Cadena que ingreso';
    end if;
    return resp;
end;
```



16

## Crear una función que recibe un parámetro INTEGER



Para resolver debe de utilizar la instrucción **CASE - WHEN**.

```
create or replace function lvlCustomer(Dinero integer)
returns text
begin
  declare resp text default '';
  if dinero > 0 then
    case
      when Dinero > 50000 then set resp = 'PLATINIUM';
      when Dinero >= 10000 and Dinero <= 50000 then set resp = 'GOLD';
      when Dinero < 10000 then set resp = 'SILVER';
    end case;
  else
    set resp = concat('Usted Debe al Banco ', -Dinero, ' BS');
  end if;
  return resp;
end;

select lvlCustomer(9999);
```

```
lvlCustomer(9999)`
1 SILVER
```

17

## Crear una función que recibe 2 parámetros VARCHAR(20), VARCHAR(20)

- La función debe de retornar un texto TEXT como respuesta.
- Si las cadenas fueran “TALLER DBA II” y la segunda cadena fuese “GESTION 2023”.
- La nueva cadena debería ser “TLLR DB -GSTN 2023”.
- La nueva cadena es resultado de la concatenación de todos los valores distintos a las vocales.
- Retornar la nueva cadena concatenada.

```
create or replace function quitaVocales(cadena1 varchar(20),cadena2 varchar(20))
returns TEXT
begin
    declare resp text default '';
    declare puntero char;
    declare cont int default 1;
    declare newcadena varchar(100) default concat(cadena1,'-',cadena2);
    WHILE (cont <= char_length(newcadena)) do
        set puntero = substr(newcadena,cont,1);
        if(puntero != 'a' and puntero != 'e' and puntero != 'i' and puntero != 'o' and puntero != 'u')then
            set resp = concat(resp,puntero);
            if(puntero = '-') then
                set resp = concat(resp,' ');
            end if;
        end if;
        set cont = cont+1;
    end while;
    return resp;
end;

select quitaVocales('TALLER DBA II','GESTION 2023');
```

```
`quitaVocales('TALLER DBA II','GESTION 2023')`
1 TLLR DB -GSTN 2023
```

## Crear una función que reciba un parámetro TEXT

**-En donde este parámetro deberá de recibir una cadena cualquiera y retorna un TEXT de respuesta.**

**-Concatenar N veces la misma cadena reduciendo en uno en cada iteración hasta llegar a una sola letra.**

**-Utilizar REPEAT y retornar la nueva cadena concatenada.**

```
create or replace function quita letra1en1(cadena1 varchar(100))
returns TEXT
begin
  declare resp text default '';
  declare puntero varchar(100);
  declare cont int default char_length(cadena1);
  repeat
    set puntero = substr(cadena1,-cont,cont);
    set resp = concat(resp,puntero,',');
    set cont = cont-1;
  until cont = 0 end repeat;
  return resp;
end;
select quita letra1en1('dbaii');
```

```
`quita letra1en1('dbaii')`
1 dbaii,baii,aii,ii,i,
```