

# EVALUCION PROCESUAL HITO 3

BASE DE DATOS II

NILBER MAYTA CUNO

# PARTE TEORICA

MANEJO DE CONCEPTOS

# “ 1. Defina que es lenguaje procedural en MySQL.

Es una extensión de MySQL que permite que el propio usuario pueda desarrollar sus propias funciones, procedimientos almacenados y desencadenadores que nos permite un mejor control sobre los datos

## 2. Defina que es una función en MySQL.

**Una función dentro de MySQL es una simplificación de un proceso al cual podemos llamar en cualquier momento, estos pueden estar ya integrados en MySQL o podemos agregar mas según nuestras necesidades.**

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

Primero tenemos la creación de la función:

```
create function
```

Después se le asigna un nombre mas los parámetros si es necesario

```
1 sumarNumeros(num1 int, num2 int)
```



Después esta el “returs” mas el tipo de variable que retornara la función

```
returns integer
```

El “begin” nos indica el inicio de la función y el “end” el final

```
begin
```

```
end;
```



El “declare” nos ayuda a declarar variables que podremos usar dentro la función y el “default” ★ asigna un valor inicial.

```
declare resp int default 0;
```

Con la clausula “set” asignamos valores en procesos

```
set resp = num1 + num2;
```



Finalmente tenemos el “return” que será el valor que nos retorna la función.

```
return resp;
```

Todo junto nos quedaría así:

```
create function sumarNumeros(num1 int, num2 int)
  returns integer
begin
  declare resp int default 0;
  set resp = num1 + num2;
  return resp;
end;
```

Y para llamar a la función usamos el “select”

```
select sumarNumeros( num1: 7, num2: 5);
```



## 4. ¿Cómo crear, modificar y cómo eliminar una función? Adjunte un ejemplo de su uso.

Para crear una función usamos la clausula “create”

```
create function sumarNumeros(num1 int, num2 int)
    returns integer
begin
    declare resp int default 0;
    set resp = num1 + num2;
    return resp;
end;
```

```
select sumarNumeros( num1: 7, num2: 5);
```

`sumarNumeros(7,5)`	
1	12

```
create or replace function sumarNumeros(num1 int, num2 int)
  returns int
begin
  declare resp int default 0;
  set resp = num1 - num2;
  return resp;
end;
```

Para modificar una función usamos la clausula “create or replace”

```
select sumarNumeros( num1: 7, num2: 5);
```

	<code>`sumarNumeros(7,5)`</code>	
1		2

Y para eliminar la función usamos el “drop funcion”

```
drop function sumarNumeros;
```

```
) FUNCTION evaluacion_procesual_h3.sumarNumeros does not exist
```

## “ 5. Para qué sirve la función CONCAT y como funciona en MYSQL

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

**La función CONCAT nos sirve para poder concatenar cadenas de texto, es decir que tiene la capacidad de unir varias cadenas de texto en una sola.**



```
create or replace function Concatenacion(txt1 text, txt2 text, txt3 text)
  returns text
begin
  declare resp text default '';
  set resp = concat(txt1,txt2,txt3);
  return resp;
end;
```

```
select Concatenacion( txt1: 'hola', txt2: 'Mundo', txt3: 'Concatenado');
```

```
1 `Concatenacion('ho
```

```
1 1 ho
```

## 6. 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 nos ayuda a extraer una parte de una cadena de texto o mejor dicho sacar un sub texto de una cadena de texto.**

```
create or replace function SubTextos(txt text)
  returns text
begin
  declare resp text default '';
  set resp = SUBSTRING(txt , 1 , 7);
  return resp;
end;
```

```
select SubTextos( txt: 'Ximena Condori Mar');
```

```
SubTextos('Ximena Condori Mar')`  
1 Ximena
```

## 7. 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.

Esta función nos ayuda a comparar cadenas de texto devolviéndonos un valor de tipo booleano que en este caso es representado con 0 como verdadero(true) y 1 o -1 como falso (false)



```
create or replace function CompararCadenas(txt1 text, txt2 text, txt3 text)
    returns text
begin
    declare resp text;
    if(strcmp(txt1, txt2) = 0 or strcmp(txt1, txt3) = 0 or strcmp(txt2, txt3) = 0) then
        set resp = 'Dos cadenas son iguales';
    else
        set resp = 'No son iguales';
    end if;
    return resp;
end;
```

```
select CompararCadenas( txt1: 'hola', txt2: 'hola', txt3: 'hole');
```

```
`CompararCadenas('hola', 'hola', 'hole')`
```

```
1 Dos cadenas son iguales
```

```
select CompararCadenas( txt1: 'hola', txt2: 'holi', txt3: 'hole');
```

```
`CompararCadenas('hola', 'holi', 'hole')`
```

```
1 No son iguales
```



## 8. 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 nos sirve para obtener la longitud de una cadena de texto mientras que el locate nos proporciona la posición donde se encuentra una sub cadena.**

```

create or replace function ContarCaracteres(cadena varchar(50), letra char)
    returns TEXT
BEGIN
    declare resp text default 'La letra no esta en la cadena';
    declare cont integer default 1;
    declare nVeces int default 0;
    declare puntero char;

    if locate(letra,cadena)>0 then
        WHILE cont <= char_length(cadena) do
            set puntero = substr(cadena,cont,1);
            if puntero = letra then
                set nVeces = nVeces+1;
            end if;
            set cont = cont+1;
        end while;
        set resp = concat('la letra ',letra,' se repite ',nVeces);
    end if;

    return resp;
end;

```

```
select ContarCaracteres( cadena: 'yyyyyyyyy', letra: 'y');
```

```

`ContarCaracteres('yyyyyyyyy','y')`
1 la letra y se repite 9

```

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

Las funciones de agregación son funciones que ya están dentro de SQL como ser el cont, concat, etc. Mientras que las funciones creados por el DBA son funciones que uno mismo crea para el manejo de datos.

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

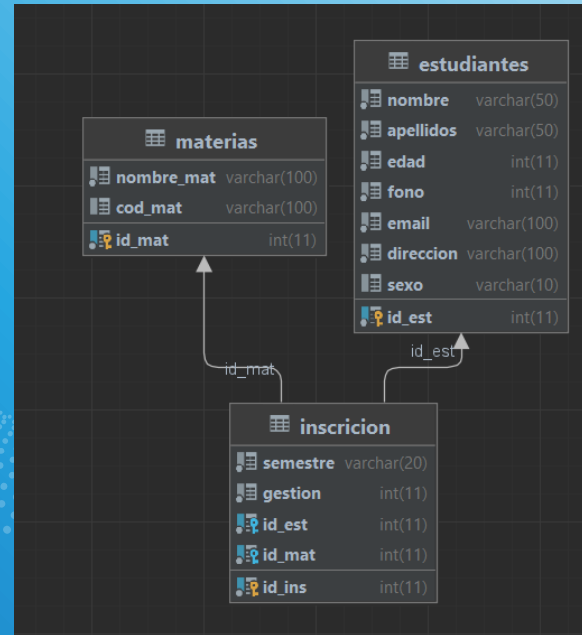
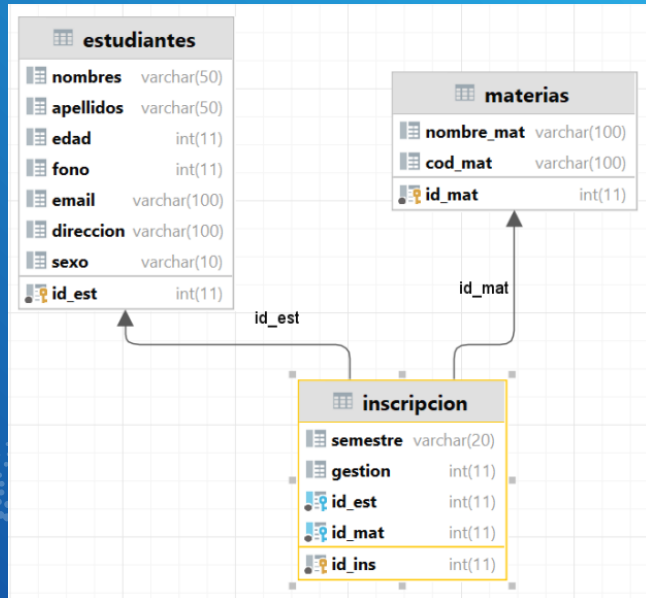
○ Es decir IN INOUT, etc.

**Los parámetros de entrada son aquellos que entran dentro de una función, mientras que los de salida son aquellos que nos retorna la función.**

# PARTE PRACTICA

The background features a solid blue color. Overlaid on this are several wavy, horizontal lines composed of small, dark blue dots. These lines create a sense of motion and depth, with some lines appearing more prominent than others, creating a layered effect.

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



```
create table estudiantes(
    id_est integer primary key auto_increment not null,
    nombre varchar(50) not null ,
    apellidos varchar(50)not null ,
    edad integer not null ,
    fono integer not null ,
    email varchar(100)not null ,
    direccion varchar(100)not null ,
    sexo varchar(10)
);
```

```
create table materias (
    id_mat integer primary key auto_increment not null ,
    nombre_mat varchar(100) not null ,
    cod_mat varchar(100)
);
```

```
create table inscripcion(
    id_ins integer primary key auto_increment not null ,
    semestre varchar(20) not null ,
    gestion integer not null ,
    id_est integer not null ,
    id_mat integer not null ,
    foreign key (id_est) references estudiantes(id_est),
    foreign key (id_mat) references materias(id_mat)
);
```



## DATOS TABLA ESTUDIANTES

id_est	nombres	apellidos	edad	fono	email	direccion	sexo
1	Miguel	Gonzales Veliz	20	2832115	miguel@gmail.com	Av. 6 de Agosto	masculino
2	Sandra	Mavir Uria	25	2832116	sandra@gmail.com	Av. 6 de Agosto	femenino
3	Joel	Adubiri Mondar	30	2832117	joel@gmail.com	Av. 6 de Agosto	masculino
4	Andrea	Arias Ballesteros	21	2832118	andrea@gmail.com	Av. 6 de Agosto	femenino
5	Santos	Montes Valenzuela	24	2832119	santos@gmail.com	Av. 6 de Agosto	masculino

## DATOS TABLA MATERIAS

id_mat	nombre_mat	cod_mat
1	Introduccion a la Arquitectura	ARQ-101
2	Urbanismo y Diseno	ARQ-102
3	Dibujo y Pintura Arquitectonico	ARQ-103
4	Matematica discreta	ARQ-104
5	Fisica Basica	ARQ-105

## DATOS TABLA INSCRIPCION

id_ins	semestre	gestion	id_est	id_mat
1	1er Semestre	2018	1	1
2	2do Semestre	2018	1	2
3	1er Semestre	2019	2	4
4	2do Semestre	2019	2	3
5	2do Semestre	2020	3	3
6	3er Semestre	2020	3	1
7	4to Semestre	2021	4	4
8	5to Semestre	2021	5	5



```

insert into estudiantes(nombre,apellidos,edad,fono,email,direccion,sexo)
values('Miguel','Gonzales Veliz',20,2832115,'miguel@gmail.com','Av. 6 de Agosto','masculino'),
      ('Sandra','Mavir Uria',25,2832116,'sandra@gmail.com','AV. 6 de Agosto','femenino'),
      ('Joel','Aduviri Mondar',30,2832117,'joel@gmail.com','AV. 6 de Agosto','masculino'),
      ('Andrea','Arias Ballesteros',21,2832118,'andrea@gmail.com','AV. 6 de Agosto','femenino'),
      ('Santos','Montes Valenzuela',24,2832119,'santos@gmail.com','Av. 6 de Agosto','masculino');

```

```

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 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',2020,4,4),
      ('5to Semestre',2021,5,5);

```

	id_est	nombre	apellidos	edad	fono	email	direccion	sexo
1	1	Miguel	Gonzales Veliz	20	2832115	miguel@gmail.com	Av. 6 de Agosto	masculino
2	2	Sandra	Mavir Uria	25	2832116	sandra@gmail.com	AV. 6 de Agosto	femenino
3	3	Joel	Aduviri Mondar	30	2832117	joel@gmail.com	AV. 6 de Agosto	masculino
4	4	Andrea	Arias Ballesteros	21	2832118	andrea@gmail.com	AV. 6 de Agosto	femenino
5	5	Santos	Montes Valenzuela	24	2832119	santos@gmail.com	Av. 6 de Agosto	masculino

	id_mat	nombre_mat	cod_mat
1	1	Introduccion a la Arquitectura	ARQ-101
2	2	Urbanismo y Diseno	ARQ-102
3	3	Dibujo y Pintura Arquitectonico	ARQ-103
4	4	Matematica discreta	ARQ-104
5	5	Fisica Basica	ARQ-105

	id_ins	semestre	gestion	id_est	id_mat
1	1	1er Semestre	2018	1	1
2	2	2do Semestre	2018	1	2
3	3	1er Semestre	2019	2	4
4	4	2do Semestre	2019	2	3
5	5	2do Semestre	2020	3	3
6	6	3er Semestre	2020	3	1
7	7	4to Semestre	2020	4	4
8	8	5to Semestre	2021	5	5

## 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,
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

```
create or replace function fibonacci(num int)
returns text
begin
    declare a int default 0;
    declare b int default 1;
    declare resp text default '';

    while num>0 do
        set resp = concat(resp,a,' ', ' ');
        set b = a + b;
        set a = b - a;
        set num = num - 1;
    end while;
    return resp;
```

```
select fibonacci( num: 7);
```

```
`fibonacci(7)`  
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.
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

- Crear una variable global de nombre LIMIT.
- Este valor debe almacenar un valor entero.
  - Ejemplo, LIMIT = 7
  - OUTPUT: 0,1,1,2,3,5,8
- Crear una función que genere la serie Fibonacci hasta ese valor LIMIT.
  - Note que el valor LIMIT debe ser usado en la función
  - La función no recibe ningún parámetro.



```
set @limite = 7;
```

```
create or replace function fibonacci_v2()  
  returns text  
begin  
  declare a int default 0;  
  declare b int default 1;  
  declare resp text default '';  
  declare num int default 0;  
  set num = @limite;  
  
  while num>0 do  
    set resp = concat(resp,a,' ', ' ');  
    set b = a + b;  
    set a = b - a;  
    set num = num - 1;  
  end while;  
  return resp;  
end;
```

```
select fibonacci_v2();
```

```
`fibonacci_v2()`  
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.

```
`paresImpares()`  
1 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24,
```

○ Si la edad mínima es IMPAR mostrar descendentemente todos los impares hasta el valor 0.

```
`paresImpares()`  
1 25,23,21,19,17,15,13,11,9,7,5,3,1,
```

- Retornar la nueva cadena concatenada.
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.
- Nota: Esta función está llamando a otra función, considere eso.



```
select min(est.edad)
from estudiantes as est;
```

```
select edadMinima();
```

```
create or replace function edadMinima()
    returns int
begin
    declare resp int;
    select min(est.edad) into resp
    from estudiantes as est;
    return resp;
end;
```

`edadMinima()`	
1	20

```

create or replace function ParesImpares()
    returns text
begin
    declare cont int default 0;
    declare resp text default '';
    declare aux int;
    set aux = edadMinima();
    if aux % 2 = 0 then
        while cont <= aux do
            set resp = concat(resp,cont,' , ');
            set cont = cont+2;
        end while;
    else
        repeat
            set resp = concat(resp,aux,' , ');
            set aux = aux - 2;
        until aux <= 0 end repeat;
    end if;
    return resp;
end;

```

```
select ParesImpares();
```

```

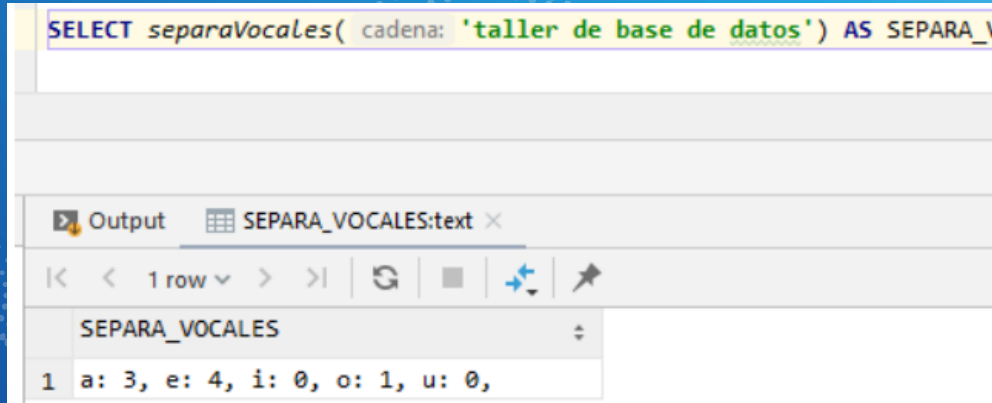
`ParesImpares()`
1 0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 , 20 ,

```

# “ 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.
- Resultado esperado.

```
SELECT separaVocales( cadena: 'taller de base de datos') AS SEPARA_V
```



SEPARA_VOCALES
1 a: 3, e: 4, i: 0, o: 1, u: 0,

○ Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

```

create or replace function ContarVocales(cadena varchar(50))
    returns TEXT
BEGIN
    declare resp text default '';
    declare cont integer default 1;
    declare nVecesa int default 0;
    declare nVecese int default 0;
    declare nVecesi int default 0;
    declare nVeceso int default 0;
    declare nVecesu int default 0;
    declare puntero char;

    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 cont <= char_length(cadena) do
            set puntero = substr(cadena,cont,1);
            if puntero = 'a' then
                set nVecesa = nVecesa + 1;
            end if;
            if puntero = 'e' then
                set nVecese = nVecese + 1;
            end if;
            if puntero = 'i' then
                set nVecesi = nVecesi + 1;
            end if;
        END WHILE
    end if;

```

```

end if;
if puntero = 'o' then
    set nVeceso = nVeceso + 1;
end if;
if puntero = 'u' then
    set nVecesu = nVecesu + 1;
end if;
set cont = cont+1;
end while;
set resp = concat(' a: ',nVecesa,', e: ',nVecese,', i: ',nVecesi,', o: ',nVeceso,', u: ',nVecesu);
end if;
return resp;
end;

```

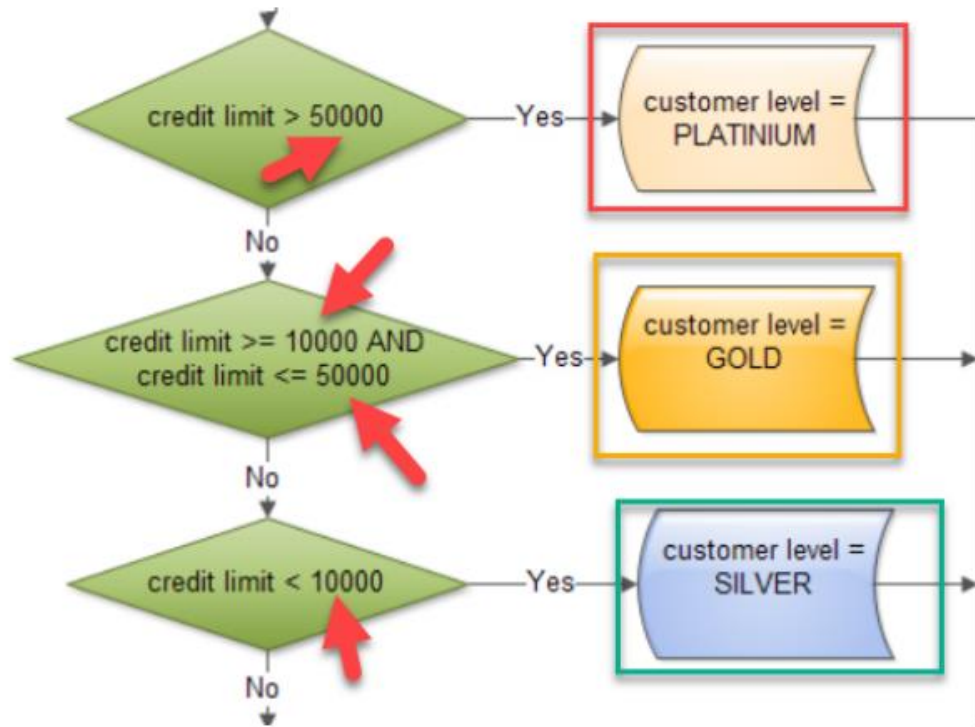
```
select ContarVocales( cadena: 'taller de base de datos');
```

```
`ContarVocales('taller de base de datos')`
```

```
1 a: 3, e: 4, i: 0, o: 1, u: 0
```

## 16. Crear una función que recibe un parámetro INTEGER.

- La función debe de retornar un texto(TEXT) como respuesta.
- El parámetro es un valor numérico credit\_number.
- Si es mayor a 50000 es PLATINIUM. 8
- Si es mayor igual a 10000 y menor igual a 50000 es GOLD.
- Si es menor a 10000 es SILVER
- La función debe retornar indicando si ese cliente es PLATINUM, GOLD o SILVER en base al valor del credit\_number.
- Considere la imagen siguiente:




○ Para resolver debe de utilizar la instrucción CASE - WHEN.  
○ Adjuntar el código SQL generado y una imagen de su correcto funcionamiento

```
create or replace function tipocredito(credit_number integer)
    returns text
BEGIN
    declare resp text default '';


    if credit_number > 50000 then
        set resp = 'cliente platino';
    end if;
    if credit_number >=10000 and credit_number <=50000 then
        set resp = 'cliente gold';
    end if;
    if credit_number < 10000 then
        set resp = 'cliente silver';
    end if;

    return resp;
end;
```


```
select tipocredito( credit_number: 5000000);
select tipocredito( credit_number: 1);
select tipocredito( credit_number: 9999);
```

 `tipocredito(5000000)`

1 cliente platino

 `tipocredito(1)`

1 cliente silver

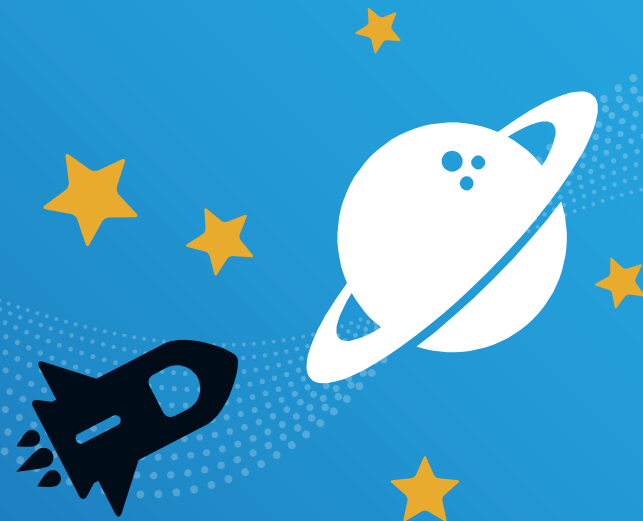
 `tipocredito(9999)`

1 cliente 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 SinVocales(txt1 varchar(20), txt2 varchar(20))
    returns text
begin
    declare resp text default '';
    declare aux char(1);
    declare cont int default 1;
    declare nuevaCadena varchar(100) default concat(txt1,'-',txt2);

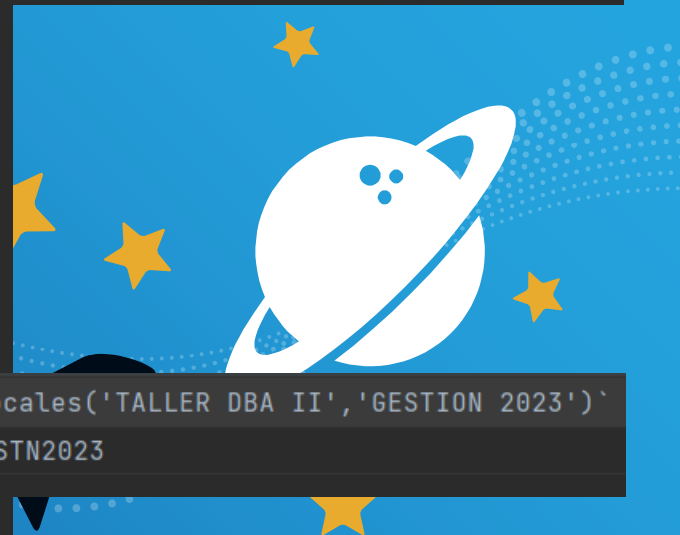
    while (cont <= char_length(nuevaCadena)) do
        set aux = substr(nuevaCadena, cont, 1);
        if (aux not in ('a','e','i','o','u'))then
            set resp = concat(resp, aux);
            if (aux = '-') then
                set resp = concat(resp, aux);
            end if;
        end if;
        set cont = cont + 1;
    end while;
    return resp;
end;

```

```
select SinVocales( txt1: 'TALLER DBA II', txt2: 'GESTION 2023');
```

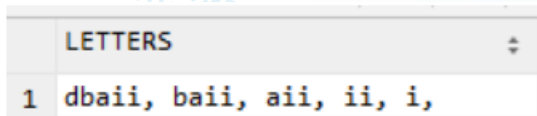
```
`SinVocales('TALLER DBA II','GESTION 2023')`
```

```
1 TLLRDB-GSTN2023
```



# 18. 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. ○ Considerar la siguiente imagen:



	LETTERS
1	dbaii, baii, aii, ii, i,

- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

```
create or replace function Deletreado(txt varchar(100))  
  returns text  
begin  
  declare resp text default '';  
  declare aux varchar(100);  
  declare cont int default char_length(txt);  
  
  repeat  
    set aux = substr(txt,-cont,cont);  
    set resp = concat(resp,aux,', ');  
    set cont = cont-1;  
  until cont=0 end repeat;  
  return resp;  
  
end;
```

```
select Deletreado( txt: 'dbaii')
```

```
`Deletreado('dbaii')`  
1 dbaii, baii, aii, ii, i,
```

# Gracias!

