

# Evaluación Procesual

UNIVERSIDAD PRIVADA FRANZ TAMAYO

DEFENSA HITO 3 – TAREA FINAL

**ESTUDIANTE:** RODRIGO CRISTHIAN TORREZ DE LA CRUZ

**ASIGNATURA:** ESTRUCTURA DE DATOS

**CARRERA:** INGENIERÍA DE SISTEMAS

PARALELO: EDD (1)

**DOCENTE:** LIC. WILLIAM BARRA PAREDES

**FECHA:** 04/05/23

**GITHUB:** [HTTPS://GITHUB.COM/RODRIGOT1/ESTRUCTURA-DE-DATOS](https://github.com/RODRIGOT1/ESTRUCTURA-DE-DATOS)

<https://github.com/rodrigoT1?tab=repositories>



UNIFRANZ

# Manejo de conceptos

## 1. DEFINA QUE ES LENGUAJE PROCEDURAL EN MySQL

Los procedimientos almacenados MySQL, también conocidos como Stored Procedure, se presentan como conjuntos de instrucciones escritas en el lenguaje SQL. Su objetivo es realizar una tarea determinada, desde operaciones sencillas hasta tareas muy complejas, el usuario da órdenes para que se realicen las tareas pertinentes con el objetivo de recuperar los datos requeridos

## 2. DEFINA QUE ES UNA FUNCION EN MySQL

Una función en MySQL es una rutina creada para tomar unos parámetros, procesarlos y retornar en un salida. Deben retornar en un valor con algún tipo de dato definido • Pueden usarse en el contexto de una sentencia SQL • Solo retornan un valor individual, no un conjunto de registros

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

Nombre.- podemos decir que será la descripción que en este caso será el nombre de la función

```
CREATE OR REPLACE FUNCTION cliente_estado_2(INGRESAR_VALOR integer)
```

Return.- su función será que no retornara el valor que puede ser numero(int) o cadena(varchar)

```
returns text
BEGIN
  DECLARE respuesta TEXT DEFAULT ',';

  if INGRESAR_VALOR > 50000 THEN SET respuesta = 'PLATINIUM';
  end if;
  if INGRESAR_VALOR >= 10000 AND INGRESAR_VALOR <= 50000 THEN
    SET respuesta = 'GOLD';
  end if;
  if INGRESAR_VALOR < 10000 THEN
    SET respuesta = 'SILVER';
  end if;

  RETURN respuesta;
end;
```

Parámetros.- los valores de entrada serán procesados a la función la cual tiene que tener un tipo de dato numérico o cadena

## Parámetros

```
create function QuitarValores(cadena1 varchar(20), cadena2 varchar(20))
```

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

Supongamos que debemos crear una función empezamos desde  
Un `create function` estaríamos creando una función

Para modificar tenemos un código para la modificación de una tabla  
O una función

```
alter function numeroFibonacci;
```

Para finalizar tenemos un código para eliminar es lo mismo con el  
`ALTER` solo que en este caso tenemos que poner `DROP` y el nombre

```
drop function numeroFibonacci;
```

Este `DROP` funciona con todo puede borrar la función, tabla o incluso  
Con la base de datos

```
create function numeroFibonacci(numero int)
returns text
begin
    declare resp text default '';
    declare numeros int default 0;
    declare cont int default 1;
    declare nummerio1 int default 0;
    declare numero2 int default 1;

    while(numero >= cont) DO
        set resp = concat(resp,nummerio1,',');
        set numeros = nummerio1 + numero2;
        set cont = cont+1;
        set nummerio1 = numero2;
        set numero2 = numeros;
    end while;
    return resp;
end;

select numeroFibonacci( numero: 7);
```

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

La función CONCAT sirve para unir en este caso, esta función nos permite unir cualquier cosa como numero, variables y cadenas y con todo lo que tenga lo convertirá en una sola cadena

Aquí tenemos una función pero queremos que no los Concatene

Simplemente debemos hacer normal el código pero

Tendremos que avisarle que no los concatene los parámetros que le pusimos a la función que en este caso será cadena1 y cadena2

```
create function QuitarValores(cadena1 varchar(20), cadena2 varchar(20))  
returns text  
begin  
    declare respuesta text default '';  
    declare cont int default 1;  
    declare NuevaCadena varchar(100) default concat(cadena1,'-',cadena2);  
    declare puntero char;
```

```
while(cont <= char_length(NuevaCadena)) DO  
    set puntero = substr(NuevaCadena, cont, 1);  
    if(puntero != 'A' AND puntero != 'E' AND puntero != 'I' AND puntero != 'O' AND puntero != 'U') then  
        set respuesta = concat(respuesta,puntero);  
        if(puntero='') then  
            set respuesta= concat(respuesta,'');  
        end if;  
    end if;  
    set cont = cont + 1;  
end while;  
return respuest
```



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

La función SUBSTRING su función es de que devuelve caracteres del valor cadena que comienzan desde la posición de caracteres especificada por inicio.

```
create function QuitarValores(cadena1 varchar(20), cadena2 varchar(20))
returns text
begin
    declare respuest text default '';
    declare cont int default 1;
    declare NuevaCadena varchar(100) default concat(cadena1,'-',cadena2);
    declare puntero char;

    while(cont <= char_length(NuevaCadena)) DO
        set puntero = substr(NuevaCadena, cont, 1);
        if(puntero != 'A' AND puntero != 'E' AND puntero != 'I' AND puntero != 'O' AND puntero != 'U') then
            set respuest = concat(respuest,puntero);
            if(puntero='') then
                set respuest= concat(respuest,'');
            end if;
        end if;
        set cont = cont + 1;
    end while;
    return respuest;
end;

select QuitarValores(cadena1: 'TALLER BDA II', cadena2: 'GESTION 2023');
```

QuitarValores('TALLER BDA II','GESTION 2023')

1 TLLRBD-GSTN2023

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

usa el conjunto de caracteres actual cuando realiza las comparaciones. Esto hace que el comportamiento de comparación por defecto no sea sensible al tipo, a no ser que uno o ambos operandos sean cadenas binarias.

```
create function quitarLetra(cadena1 varchar(30))
returns text
begin
  declare respuesta text default '';
  declare cont int default char_length(cadena1);
  declare puntero varchar(20);
  repeat
    set puntero=substr(cadena1,-cont,cont);
    set respuesta=concat(respuesta,puntero,',');
    set cont = cont -1;
  until cont = 0 end repeat;
  return respuesta;
end;
```

	<code>`quitarLetra('dbaii')`</code>	
1	dbaii,baii,aii,ii,i,	

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

La función *CHAR\_LENGTH* devuelve el número de caracteres presentes en una expresión aritmética.

```
while(cont <= char_length(NuevaCadena)) DO
    set punntero = substr(NuevaCadena, cont, 1);
    if(punntero != 'A' AND punntero != 'E' AND punntero != 'I' AND punntero != 'O' AND punntero != 'U') then
        set respuewsta = concat(respuewsta,punntero);
        if(punntero='') then
            set respuewsta= concat(respuewsta,'');
        end if;
    end if;
    set cont = cont + 1;
end while;
```

```
■ `QuitarValores('TALLER BDA II','GESTION 2023')`
```

```
1 TLLRBD-GSTN2023
```



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

La función principal de un DBA es implementar, mantener, optimizar y administrar estructuras de bases de datos para la organización. Básicamente, el DBA mantiene las bases de datos y las aplicaciones funcionando hasta PAR, o rendimiento, disponibilidad y capacidad de recuperación.

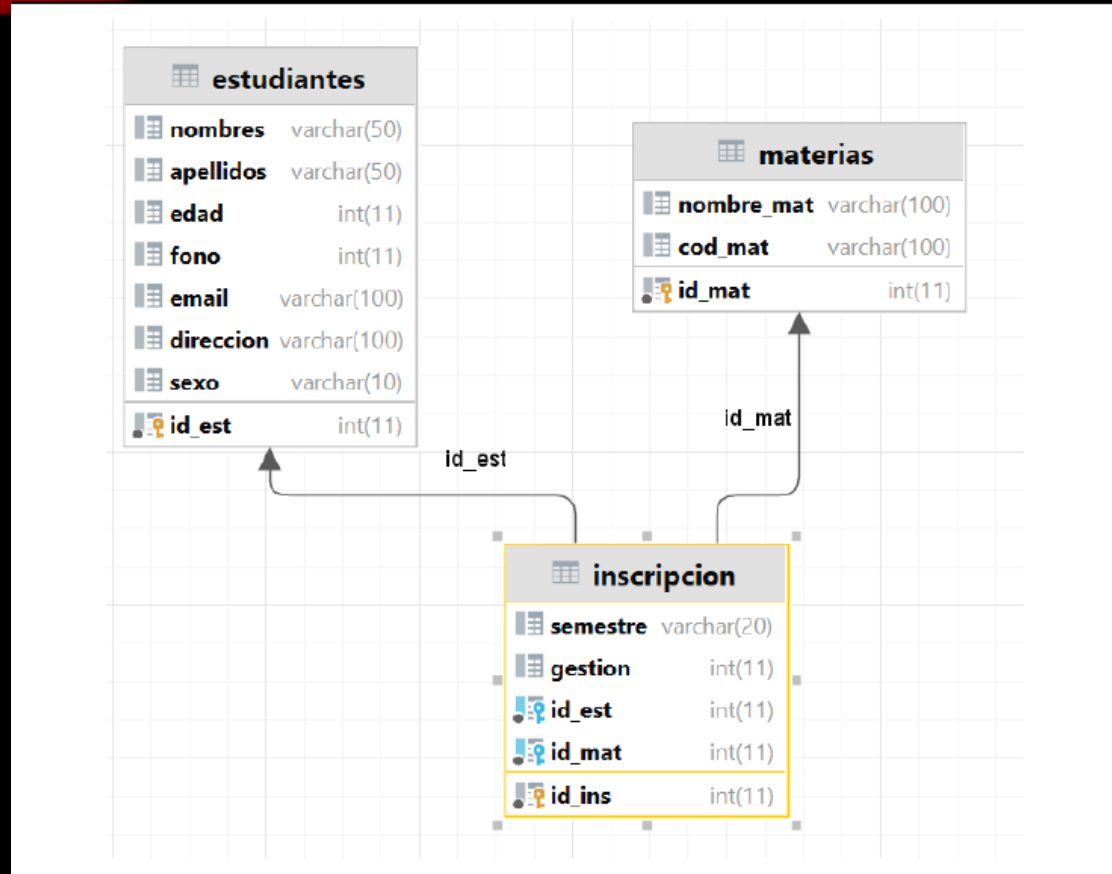
### **10.¿Busque y defina a qué se referirá cuando se habla de parámetros de entrada y salida en MySQL?**

Los parámetros de entrada permiten a quien realiza la llamada pasar un valor de datos a la función o al procedimiento almacenado. Los parámetros de salida permiten al procedimiento almacenado devolver un valor de datos o variable de cursor a quien realizó la llamada.

# MODELO PRACTICO



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



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



# SOLUCION

## EJERCICIO 1 1

# TABLA ESTUDIANTES

```
#####CREAR LAS TABLAS#####  
#####CREAR LA TABLA ESTUDIANTES  
create table Estudiantes(  
    id_est int auto_increment primary key,  
    Nombres varchar(30),  
    Apellido varchar(30),  
    Edad int,  
    Fono int,  
    Email varchar(30),  
    Direccion varchar(30),  
    Sexo varchar(30)  
);  
select * from Estudiantes;
```

	id_est	Nombres	Apellido	Edad	Fono	Email	Direccion	Sexo
1	1	Rodrigo	Torrez	19	60630082	rodrigo34torrez@gmail.com	Miraflores	Masculino
2	2	Andrea	Quisbert	19	78546811	Andreaquis23bert@gmail.com	Zona sur	Femenino
3	3	Rolando	Flores	26	87654921	rolando23flores@gmail.com	Achumani	Masculino
4	4	Lourdes	Apaza	28	68795438	lourdesapaza2@gmail.com	Villa fatima	Femenino
5	5	Ronald	De La Cruz	20	84765946	ronald123delacruz@gmail.com	San pedri	Masculino
6	6	Camila	Ramirez	25	98756489	camila32ramirez2@gmail.com	Satelite	Femenino



# TABLA INSCRIPCION

```
#####CREAR LA TABLA INSCRIPCION
create table Incripcion(
  id_ins int auto_increment primary key,
  Semestre varchar(30),
  Gestion int,
  id_est int,
  id_mat int,
  foreign key (id_est) references Estudiantes(id_est),
  foreign key (id_mat) references Materias(id_mat)
);
select * from Incripcion;
```

	id_ins	Semestre	Gestion	id_est	id_mat
1	7	1° Semestre	2023	1	1
2	8	3° Semestre	2019	2	3
3	9	3° Semestre	2015	2	4
4	10	2° Semestre	2018	3	1
5	11	5° Semestre	2021	4	4
6	12	4° Semestre	2022	5	5

# TABLA MATERIAS

```
#####CREAR LA TABLA MATERIAS
create table Materias(
    id_mat int auto_increment primary key,
    Nombre_Mat varchar(30),
    Cod_Mat varchar(30)
);
select * from Materias;
```

	id_mat	Nombre_Mat	Cod_Mat
1	1	Contaduria	CAR-101
2	2	Ingenieria Comercial	ING-101
3	3	Derecho	DER-101
4	4	Psicologia	PSI-101
5	5	Arquitectura	ARQ-101
6	6	Comercio Internacional	CIN-101



AGREGAR

VALORES A CADA TABLA

```
insert into Estudiantes(NOMBRES, APELLIDO, EDAD, FONO, EMAIL, DIRECCION, SEXO)
values('Rodrigo','Torrez',19,60630082,'rodrigo34torrez@gmail.com','Miraflores','Masculino'),
      ('Andrea','Quisbert',19,78546811,'Andreaquis23bert@gmail.com','Zona sur','Femenino'),
      ('Rolando','Flores',26,87654921,'rolando23flores@gmail.com','Achumani','Masculino'),
      ('Lourdes','Apaza',28,68795438,'lourdesapaza2@gmail.com','Villa fatima','Femenino'),
      ('Ronald','De La Cruz',20,84765946,'ronald123delacruz@gmail.com','San pedri','Masculino'),
      ('Camila','Ramirez',25,98756489,'camila32ramirez2@gmail.com','Satelite','Femenino');
```

```
insert into Inscripcion(Semestre, Gestion, id_est, id_mat)
values('1° Semestre',2023,1,1),
      ('3° Semestre',2019,2,3),
      ('3° Semestre',2015,2,4),
      ('2° Semestre',2018,3,1),
      ('5° Semestre',2021,4,4),
      ('4° Semestre',2022,5,5);
```

```
insert into Materias(nombre_mat, cod_mat)
values('Contaduria','CAR-101'),
      ('Ingenieria Comercial','ING-101'),
      ('Derecho','DER-101'),
      ('Psicologia','PSI-101'),
      ('Arquitectura','ARQ-101'),
      ('Comercio Internacional','CIN-101');
```



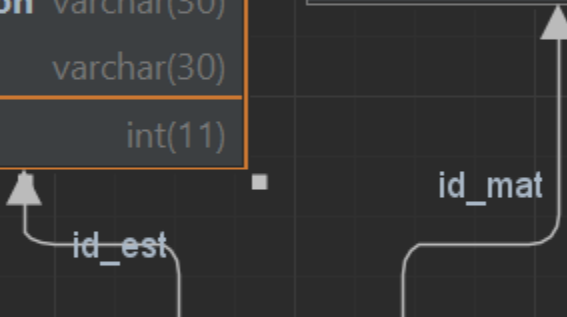
**DIAGRAMA SHOW**



estudiantes	
Nombres	varchar(30)
Apellido	varchar(30)
Edad	int(11)
Fono	int(11)
Email	varchar(30)
Direccion	varchar(30)
Sexo	varchar(30)
<b>id_est</b>	int(11)

materias	
Nombre_Mat	varchar(30)
Cod_Mat	varchar(30)
<b>id_mat</b>	int(11)

inscripcion	
Semestre	varchar(30)
Gestion	int(11)
<b>id_est</b>	int(11)
<b>id_mat</b>	int(11)
<b>id_ins</b>	int(11)



## 12. Crear una función que genere la serie Fibonacci.

```
create function numeroFibonacci(numero int)
returns text
begin
    declare resp text default '';
    declare numeros int default 0;
    declare cont int default 1;
    declare nummerio1 int default 0;
    declare numero2 int default 1;

    while(numero >= cont) DO
        set resp = concat(resp,nummerio1,',');
        set numeros = nummerio1 + numero2;
        set cont = cont+1;
        set nummerio1 = numero2;
        set numero2 = numeros;
    end while;
    return resp;
end;

select numeroFibonacci( numero: 7);
```

- o La función recibe un límite(number)
- o La función debe de retornar una cadena.
- o Ejemplo para n=7. OUTPUT: 0, 1, 1, 2, 3, 5, 8,
- o Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.

	<code>`numeroFibonacci(7)`</code>	
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 @limit = 7;
```

```
##no pondremos nada por que ya estamos poniendo una variable global que sera igual a 7
select numeroFibonacciVariableGlobal();
```

```
create function numeroFibonacciVariableGlobal()
returns text
begin
    declare resp text default '';
    declare numeros int default 0;
    declare cont int default 1;
    declare nummerio1 int default 0;
    declare numero2 int default 1;

    while(@limit >= cont) DO
        set resp = concat(resp,nummerio1,',');
        set numeros = nummerio1 + numero2;
        set cont = cont+1;
        set nummerio1 = numero2;
        set numero2 = numeros;
    end while;
    return resp;
end;
```

# 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.

```

create or replace function EdadMinima()
returns int
begin
    declare resp int default 0;
    select min(edad) into resp from Estudiantes;
    return resp;
end;

```

⌘ `ParesImpares()`

1 19,17,15,13,11,9,7,5,3,1,

```

create function ParesImpares()
returns text
begin
    declare resp text default '';
    declare par int default 0;
    declare imppar int default EdadMinima();

    if EdadMinima()%2=0 then
        repeat
            if(par %2=0) then
                set resp = concat(resp, par,',');
            end if;
            set par = par + 2;
        until par > EdadMinima() end repeat;
    else
        imapares: loop
            if imppar < 0 then
                leave imapares;
            end if;

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

            set imppar = imppar -1;

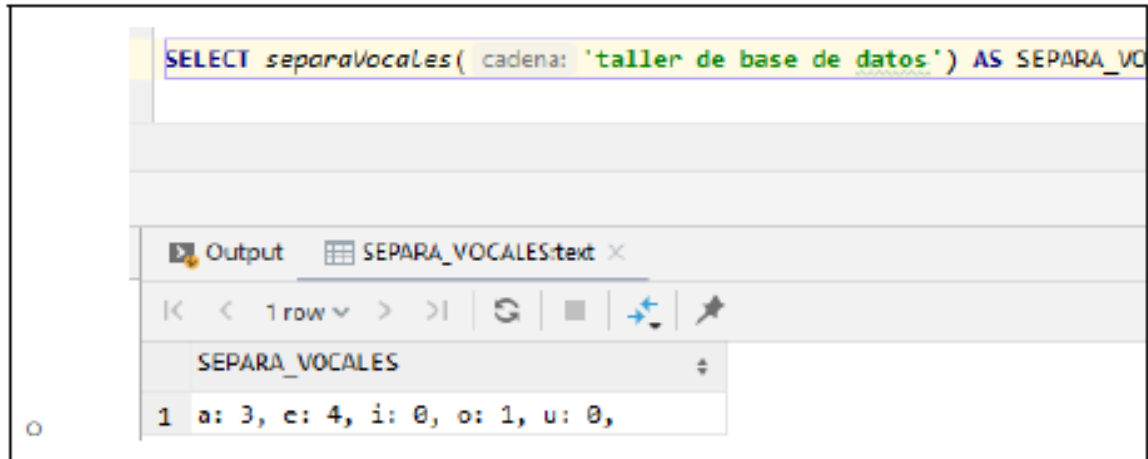
            iterate imapares;
        end loop;
    end if;
    return resp;
end;

```



# 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.



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

```
create function cuantasVecesRPVocales(cadena varchar(50))
returns text
begin
    declare respuesta text default ',';
    declare numero int default char_length(cadena);
    declare vocalA int default 0;
    declare vocalE int default 0;
    declare vocalI int default 0;
    declare vocalO int default 0;
    declare vocalU 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(numero > 0) DO
        if substr(cadena,numero,1)like'%A' then
            set vocalA=vocalA+1;
        end if;
        if substr(cadena,numero,1)like'%E' then
            set vocalE=vocalE+1;
        end if;
        if substr(cadena,numero,1)like'%I' then
            set vocalI=vocalI+1;
        end if;
        if substr(cadena,numero,1)like'%O' then
            set vocalO=vocalO+1;
        end if;
        if substr(cadena,numero,1)like'%U' then
            set vocalU=vocalU+1;
        end if;
        SET numero=numero-1;
    end while;
    set respuesta=concat('A:', vocalA , 'E:', vocalE , 'I:', vocalI , 'O:', vocalO , 'U:', vocalU );
end if;

```

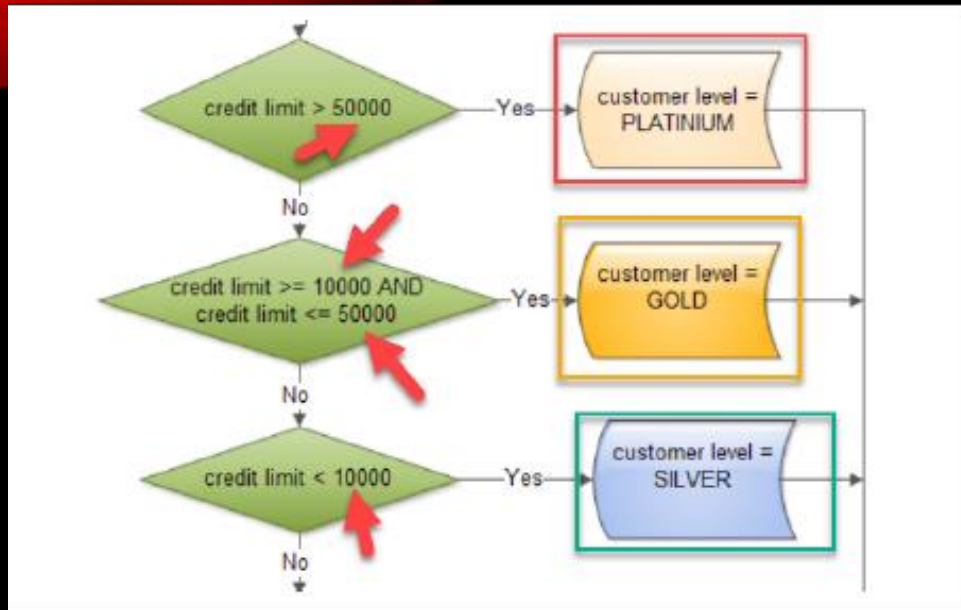
SEPARA\_VOCALES

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

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

1 A:3E:4I:0O:1U: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**.
- 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 PLATINIUM, GOLD o SILVER en base al valor del `credit_number`.
- Considere la imagen siguiente:

```
SELECT cliente_estado_2( INGRESAR_VALOR: 100000);
```

```
`cliente_estado_2(100000)`  
1 PLATINIUM
```

```
SELECT cliente_estado_2( INGRESAR_VALOR: 10000);
```

```
`cliente_estado_2(10000)`  
1 GOLD
```

```
SELECT cliente_estado_2( INGRESAR_VALOR: 999);
```

```
`cliente_estado_2(999)`  
1 SILVER
```

```
CREATE OR REPLACE FUNCTION cliente_estado_2(INGRESAR_VALOR integer)  
returns text  
BEGIN  
    DECLARE respuesta TEXT DEFAULT ',';  
  
    if INGRESAR_VALOR > 50000 THEN SET respuesta = 'PLATINIUM';  
    end if;  
    if INGRESAR_VALOR >= 10000 AND INGRESAR_VALOR <= 50000 THEN  
        SET respuesta = 'GOLD';  
    end if;  
    if INGRESAR_VALOR < 10000 THEN  
        SET respuesta = 'SILVER';  
    end if;  
  
    RETURN respuesta;  
end;
```

# 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 function QuitarValores(cadena1 varchar(20), cadena2 varchar(20))
returns text
begin
    declare respuestatext default '';
    declare cont int default 1;
    declare NuevaCadena varchar(100) default concat(cadena1,'-',cadena2);
    declare punntero char;

    while(cont <= char_length(NuevaCadena)) DO
        set punntero = substr(NuevaCadena, cont, 1);
        if(punntero != 'A' AND punntero != 'E' AND punntero != 'I' AND punntero != 'O' AND punntero != 'U') then
            set respuestatext = concat(respuestatext,punntero);
            if(punntero='') then
                set respuestatext= concat(respuestatext,'');
            end if;
        end if;
        set cont = cont + 1;
    end while;
    return respuestatext;
end;
```

```
SELECT `QuitarValores('TALLER DBA II','GESTION 2023')`
1 TLLRBD-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 function quitarLetra(cadena1 varchar(30))
returns text
begin
    declare respuesta text default '';
    declare cont int default char_length(cadena1);
    declare puntero varchar(20);
    repeat
        set puntero=substr(cadena1,-cont,cont);
        set respuesta=concat(respuesta,puntero,',');
        set cont = cont -1;
    until cont = 0 end repeat;
    return respuesta;
end;
```

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