

TAREA

HITO - 3

ESTUDIANTE:

ANDRÉS VLADIMIR QUIROGA HUARISTE SIS9208422

DOCENTE:

ING. WILLIAM RODDY BARRA PAREDES

MATERIA:

BASE DE DATOS II

Manejo de conceptos

- ▶ **Defina que es lenguaje procedural en mysql.**

- ▶ R. Conjunto de instrucciones SQL, mas una serie de estructuras de control que pueden almacenarse en el servidor.

- ▶ **Defina que es una función en mysql**

- ▶ R. Son definidas por el usuario, son rutinas que aceptan parámetros, realiza una acción y devuelven el resultado como un único valor .

- ▶ **¿Que cosas características debe tener una función? explique sobre el nombre, el return, parámetros, etc....**

- ▶ R. La sentencia Return se utiliza para concluir la ejecución de una rutina, para las funciones o métodos en sql, devuelve el resultado.

- ▶ Los parámetros se usan para intercambiar datos entre las funciones y los procedimientos almacenados.

- ▶ Las características de una función son:

- ❑ Integridad de los datos.

- ❑ Lenguaje estandarizado

- ❑ Sencilles y Claridad

- ¿Cómo crear, modificar y como eliminar una función? adjunte un ejemplo de su uso.

R.

```
drop function ejemplo;  
  
create or replace function ejemplo() returns varchar(60)  
begin  
    declare respuesta varchar(60) default '';  
    if @admin = 'admin'  
    then  
        set respuesta = 'Usuario Admin';  
    else  
        set respuesta = 'Usuario Invitado';  
    end if;  
    return respuesta;  
end;
```

- Para que sirve la función concat y como funciona en mysql.

R. La función sirve para unir datos dentro de una consulta.

```
create or replace function numeros_naturales(limite integer) returns text
begin
    declare response text default '';
    declare x integer default 1;

    while x <= limite do
        set response = CONCAT(response , x , ',');
        set x = x + 1;
    end while;
    return response;
end;
```

- Para que sirve la función substring como funciona en mysql.

R. Devuelve el subconjunto de una cadena basado en la posición inicial especificada.

```
select substring('Ximena Condori Mar' , 1 , 6);
```

1 row	
	<code>`substring('Ximena Condori Mar' , 1 , 6)`</code>
1	Ximena

- Para qué sirve la función strcmp y como funciona en mysql

la función strcmp devolverá 0 si las cadenas son iguales. en caso de que la primera sea "menor" que la segunda, devolverá un número negativo, y finalmente si la primera es mayor que la segunda, devolverá un número positivo.



```
select strcmp('dia' , 'dias');
```

<code>`strcmp('dia' , 'dias')`</code>	
1	-1

<code>select strcmp('dias' , 'dia');</code>	
Output <code>strcmp('dias' , 'dia');tinyint</code>	
1 row	
	<code>`strcmp('dias' , 'dia')`</code>
1	1

- Para qué sirve la función char_length y locate y como funciona en mysql

R. Char_length: Devuelve el largo de texto

Locate: Retorna la posición de la primera ocurrencia de la subcadena enviada como primer argumento en la cadena enviada como segundo argumento, empezando en la posición enviada como tercer argumento. Devuelve 0 si la subcadena no se encuentra en la cadena.

```
select char_length('Hola Mundo');
```

char_length('Hola Mundo'):int	
1	10

```
select locate('x' , 'Andres');
```

locate('x' , 'Andres'):int	
1	0

```
select locate('s' , 'Andres');
```

locate('s' , 'Andres'):int	
1	6

► ¿Cual es la diferencia entre las funciones de agregación y funciones creados por el dba? es decir funciones creadas por el usuario.

R. Las funciones de agregación son aquellas que ya viene con el gestor de base de datos.

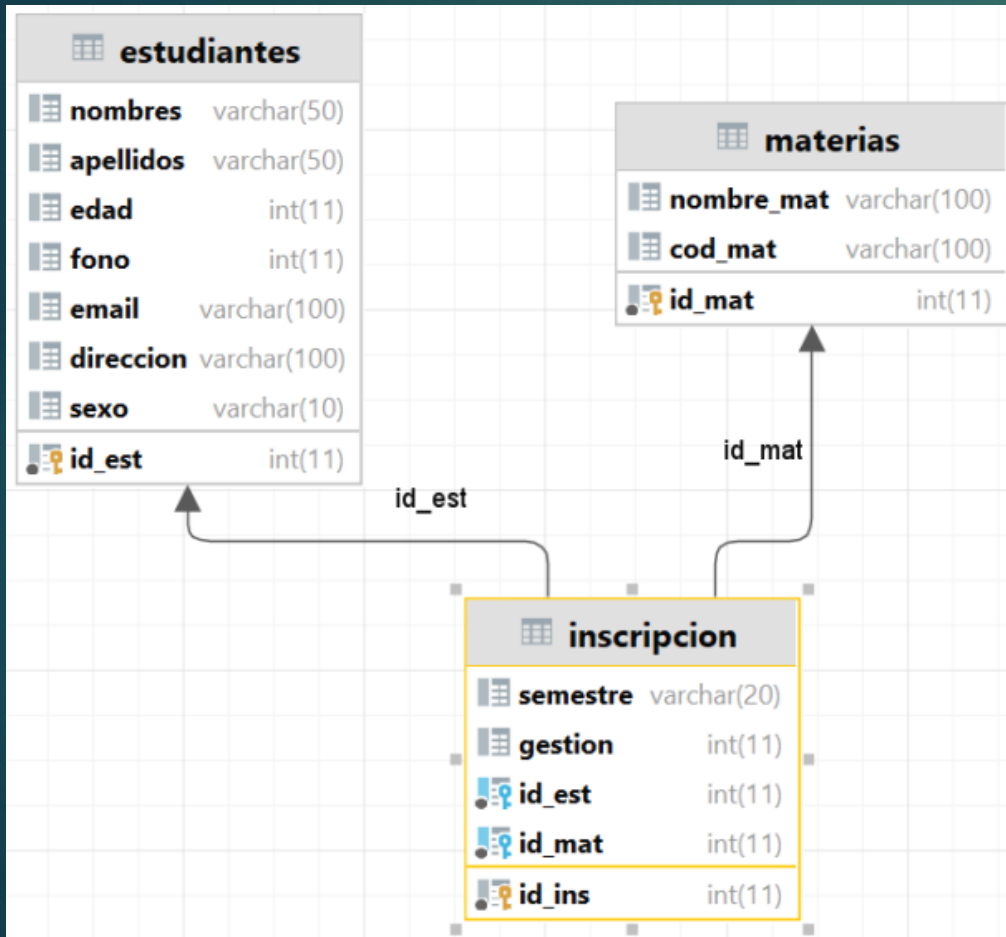
Y las funciones creadas por la base de datos nos permiten procesar y manipular datos dependiendo el lenguaje de base de datos que sea.

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

R. 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.

PARTE PRACTICA

- 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 Uriá	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


```

create table estudiantes(
    id_est integer primary key,
    nombres varchar(50),
    apellidos varchar(50),
    edad integer,
    fono integer,
    email varchar(100),
    direccion varchar(100),
    sexo varchar(100)
);

drop table materias;

create table materias(
    id_mat integer primary key ,
    nombre_mat varchar(100),
    cod_mat varchar(100)
);

create table inscripcion(
    semestre varchar(20),
    gestion integer,
    id_est integer,
    id_mat integer,
    id_ins integer
);

```

```

insert into estudiantes(id_est , nombres, apellidos, edad, fono, email, direccion, sexo)
values (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');

insert into materias(id_mat , nombre_mat, cod_mat)
values (1 , 'Introduccion a la Arquitectura' , 'ARQ-101'),
(2 , 'Urbanismo y Diseño' , 'ARQ-102'),
(3 , 'Dibujo y Pintura Arquitectonico' , 'ARQ-103'),
(4 , 'Matematica Discreta' , 'ARQ-104'),
(5 , 'Fisica Basica' , 'ARQ-105');

insert into inscripcion(id_ins , semestre, gestion, id_est, id_mat)
values (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);

```

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

La función recibe un límite (number)

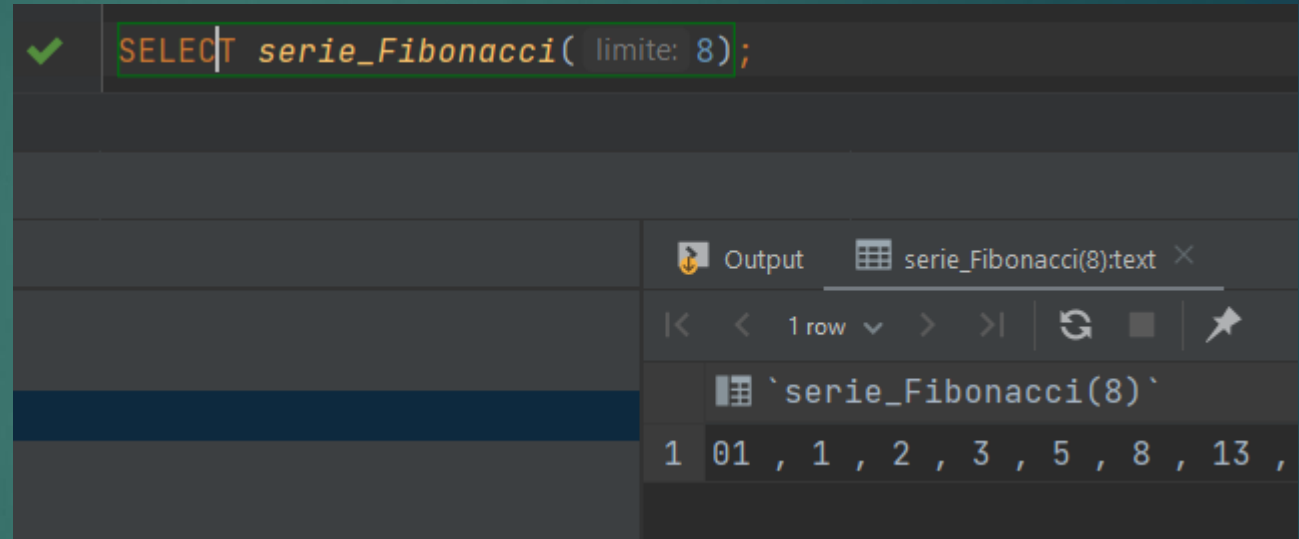
La función debe de retornar una cadena.

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 serie_Fibonacci(limite integer) returns text
BEGIN
  DECLARE A INT DEFAULT 0;
  DECLARE B INT DEFAULT 1;
  declare i int default 0;
  declare siguiente int default 0;
  declare respuesta text default "";
  while i <= limite DO
    SET i=i+1;
    if i <= 1
      then
        set siguiente = i;
      else
        set siguiente=A+B;
        set A=B;
        SET B=siguiente;
      end if;
    set respuesta= concat(respuesta,siguiente,', ');
  end while ;
  return respuesta;
END;

SELECT serie_Fibonacci(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.

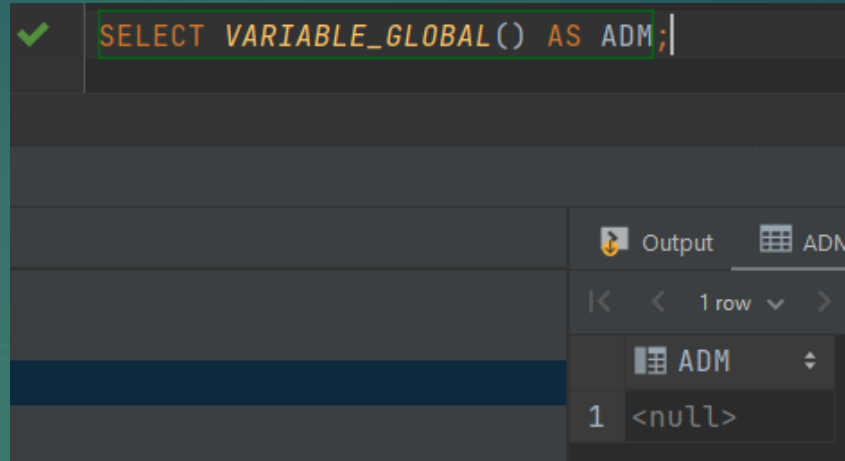
- **Crear una variable global de nombre LIMIT.**
- **Este valor debe almacenar un valor entero.**
 - Ejemplo, **LIMIT = 7**
 - **OUTPUT: 0,1,1,2,3,5,8**
- **Crear una función que genere la serie fibonacci hasta ese valor LIMIT.**
 - **Note que el valor LIMIT debe ser usado en la función**
 - **La función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

```
SET @USER='Andres';
```

```
CREATE FUNCTION  
VARIABLE_GLOBAL()  
RETURNS TEXT  
BEGIN  
    RETURN @USER;
```

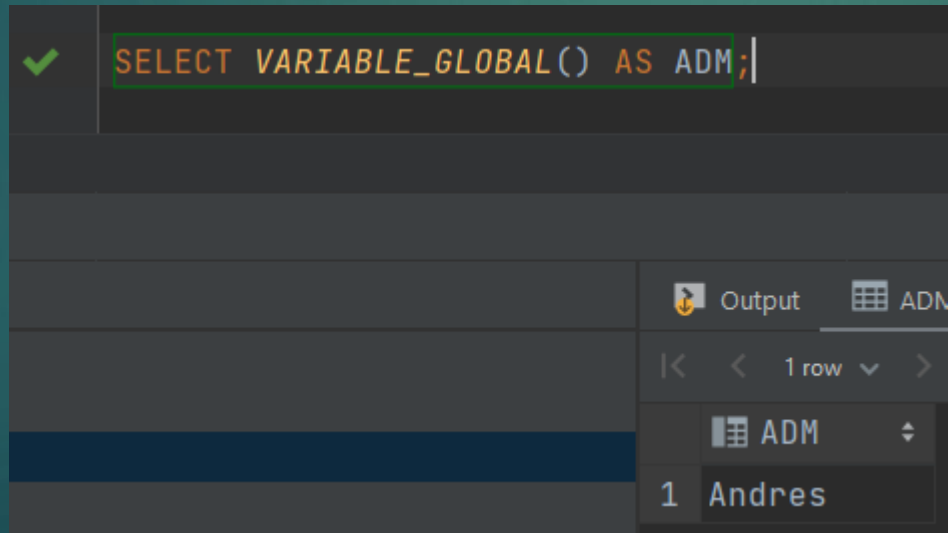
```
end;
```

```
SELECT VARIABLE_GLOBAL() AS  
ADM;
```



A screenshot of a SQL query execution interface. The query `SELECT VARIABLE_GLOBAL() AS ADM;` is entered in the top bar. Below the query, a table with one row is displayed. The row is highlighted in blue. The table has a column header `ADM` and a single row with the value `<null>`. The interface includes a green checkmark icon on the left, a 'Output' tab, and navigation controls for the result set.

ADM
<null>



A screenshot of a SQL query execution interface, similar to the one above. The query `SELECT VARIABLE_GLOBAL() AS ADM;` is entered. The result table now shows the value `Andres` in the single row. The interface elements are consistent with the previous screenshot, including the green checkmark, 'Output' tab, and navigation controls.

ADM
Andres

- ▶ 14. Crear una función que no reciba 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.

```

SET @LIMIT=8;

CREATE OR REPLACE FUNCTION parametros()
RETURNS text
BEGIN
    DECLARE A INT DEFAULT 0;
    DECLARE B INT DEFAULT 1;
    declare i int default 0;
    declare siguiente int default 0;
    declare respuesta text default '';
    while i <= @LIMIT DO
        SET i=i+1;
        if i < 1
            then
                set siguiente = i;
            else
                set siguiente=A+B;
                set A=B;
                SET B=siguiente;
            end if;
        set respuesta= concat(respuesta,siguiente,' ',
    ');

        end while ;
    return respuesta;
END;

SELECT parametros() AS ADMIN;

```

```

CREATE OR REPLACE FUNCTION MIN_EDAD()
RETURNS INT
BEGIN
    RETURN
        (
            SELECT MIN(est.edad)
            FROM estudiantes AS est

        );
end;

SELECT MIN_EDAD();

```

```

CREATE OR REPLACE FUNCTION SERIE_EDAD(X INT)
RETURNS TEXT
BEGIN
    DECLARE A TEXT DEFAULT '';
    REPEAT
        IF X % 2=0
            THEN
                SET A=CONCAT(' ',X,A);
                SET X=X-2;
            ELSE IF X%2=1
                THEN
                    SET A= CONCAT(A,X,' ');
                    SET X=X-2;
                end if;
            end if;
    until X <=0
        end repeat;
    RETURN A;
end;

SELECT SERIE_EDAD(MIN_EDAD());

SELECT SERIE_EDAD(23);

```

```

SET @LIMT=8;

CREATE OR REPLACE FUNCTION parametros()
RETURNS text
BEGIN
    DECLARE A INT DEFAULT 0;
    DECLARE B INT DEFAULT 1;
    declare i int default 0;
    declare siguiente int default 0;
    declare respuesta text default '';
    while i <= @LIMT DO
        SET i=i+1;
        if i < 1
            then
                set siguiente = i;
            else
                set siguiente=A+B;
                set A=B;
                SET B=siguiente;
            end if;
        set respuesta= concat(respuesta,siguiente,' , ');
    end while ;
    return respuesta;
END;

```

```
SELECT parametros() AS ADMIN;
```

```

CREATE OR REPLACE FUNCTION MIN_EDAD()
RETURNS INT
BEGIN
    RETURN
        (
            SELECT MIN(est.edad)
            FROM estudiantes AS est
        );
end;

```

```
SELECT MIN_EDAD();
```

```

CREATE OR REPLACE FUNCTION SERIE_EDAD(X INT)
RETURNS TEXT
BEGIN
    DECLARE A TEXT DEFAULT '';
    REPEAT
        IF X % 2=0
            THEN
                SET A=CONCAT(' , ',X,A);
                SET X=X-2;
            ELSE IF X%2=1
                THEN
                    SET A= CONCAT(A,X,' , ');
                    SET X=X-2;
                end if;
            end if;
    until X <=0
        end repeat;
    RETURN A;
end;

```

```
SELECT SERIE_EDAD( X: MIN_EDAD());
```

```
SELECT SERIE_EDAD( X: 20);
```

```
SELECT SERIE_EDAD( X: MIN_EDAD());
```

Output SERIE_EDAD(MIN_EDAD()):text X

1 row

SERIE_EDAD(MIN_EDAD())`

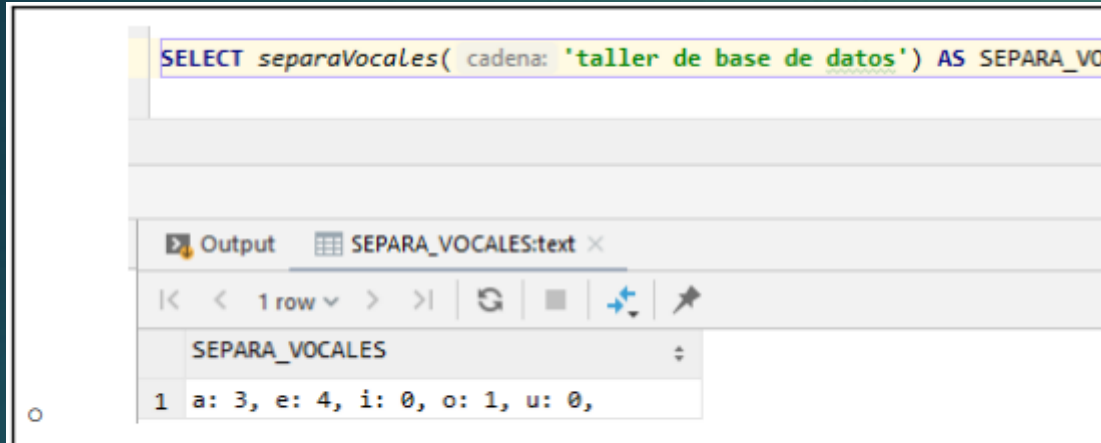
1 , 0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 , 20

- 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.



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


```

create or replace function vocales(par1 text)
returns text
begin
  declare x int default 1;
  declare response text default '';
  declare letra char default '';
  declare limite int default char_length(par1);
  declare a int default 0;
  declare e int default 0;
  declare i int default 0;
  declare o int default 0;
  declare u int default 0;
  while x <= limite do
    set letra = substring(par1, x, 1);
    if letra = 'a'
    then
      set a = a + 1;
    else if letra = 'e'
    then
      set e = e + 1;
    else if letra = 'i'
    then
      set i = i + 1;
    else if letra = 'o'
    then
      set o = o + 1;
    else if letra = 'u'
    then
      set u = u + 1;
    end if; end if; end if; end if; end if;
    set x = x + 1;
  end while;
  set response = concat('A: ', a, ', E: ', e, ', I: ', i, ', O: ', o, ', U: ', u);
  return response;
end;

```

```

select vocales('taller de base de datos') as SEPARA_VOCALES;

```

✓ `select vocales(par1: 'taller de base de datos') as SEPARA_VOCALES;`

Output SEPARA_VOCALES:text

1 row

SEPARA_VOCALES

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

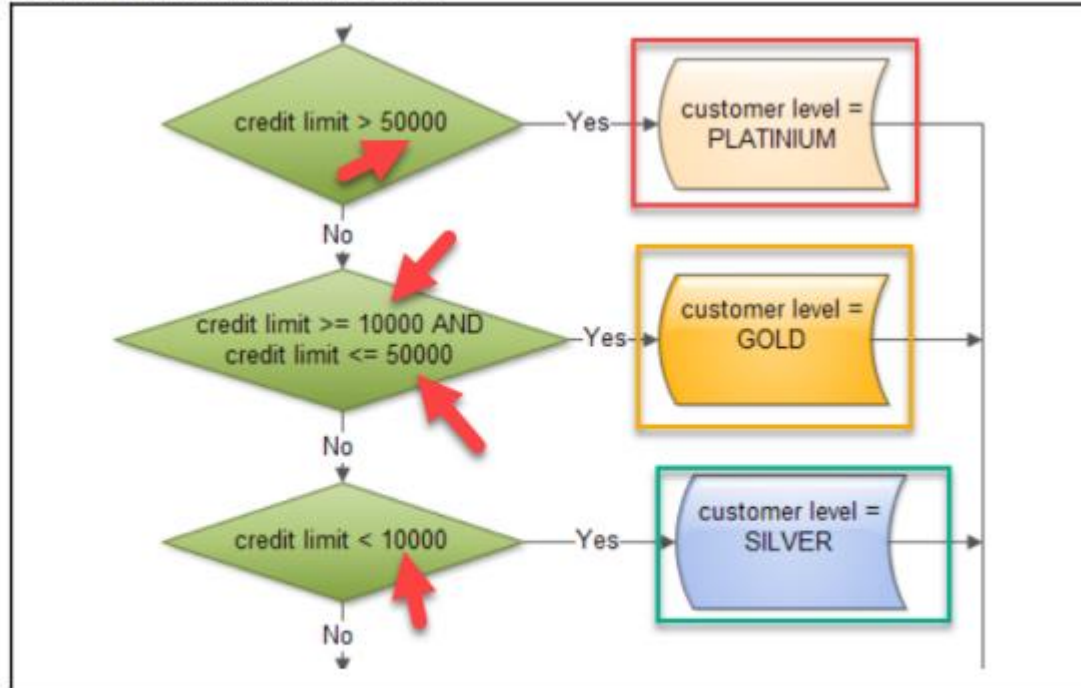
Si es mayor igual a 10000 y menor igual a 50000 es GOLD.

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:

- 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
parametro_int(parametro int)
returns text
begin
declare numero int default 0;
declare respuesta text default '';
set numero = parametro;
case
when numero > 50000 then set
respuesta='PLATIUM';
when numero >= 10000 and numero <=
50000 then set respuesta='GOLD';
when numero < 10000 then set
respuesta='SILVER';
ELSE SET respuesta='NO EXISTE';
END CASE;
RETURN respuesta;
end;

SELECT parametro_int(60000) as USUARIO;
```

```
create or replace function parametro_int(parametro int)
returns text
begin
declare numero int default 0;
declare respuesta text default '';
set numero = parametro;
case
when numero > 50000 then set respuesta='PLATIUM';
when numero >= 10000 and numero <= 50000 then set respuesta='GOLD';
when numero < 10000 then set respuesta='SILVER';
ELSE SET respuesta='NO EXISTE';
END CASE;
RETURN respuesta;
end;

SELECT parametro_int( parametro: 10000) as USUARIO;
```

Output		USUARIO:text	
		1 row	
USUARIO			
1	GOLD		

- ▶ 17. Crear una función que reciba un parámetro TEXT

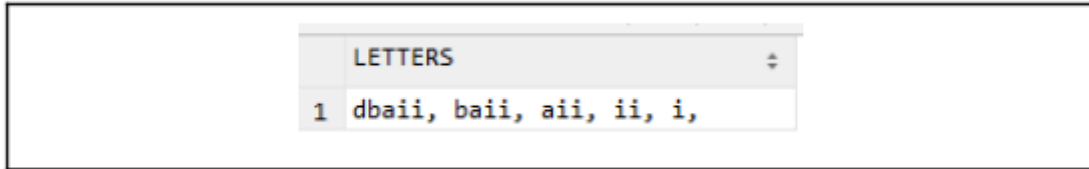
En donde este parámetro deberá de recibir una cadena cualquiera y retorna un TEXT de respuesta.

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:

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 parametro_text(x text)
returns text
begin
  declare str text default '';
  declare legth_concat int default char_length(x);
  declare limite int default 1;
  declare sub int default legth_concat;

  repeat
    if legth_concat >= limite
    then
      set str = concat(str, substr(x, legth_concat, sub - 1), ' ', '');
      set legth_concat = legth_concat - 1;
    end if;
  until legth_concat <= 0
  end repeat;
  return str;
end;

select parametro_text('BBaII');

```

```

create or replace function parametro_text(x text)
returns text
begin
  declare str text default '';
  declare legth_concat int default char_length(x);
  declare limite int default 1;
  declare sub int default legth_concat;

  repeat
    if legth_concat >= limite
    then
      set str = concat(str, substr(x, legth_concat, sub - 1), ' ', '');
      set legth_concat = legth_concat - 1;
    end if;
  until legth_concat <= 0
  end repeat;
  return str;
end;

select parametro_text( x: 'BBaII');

```

Output parametro_text('BBaII').text parametro_text('BBaII').te

1 row

1 parametro_text('BBaII')

1 I , II , AII , BAII , BBAI ,