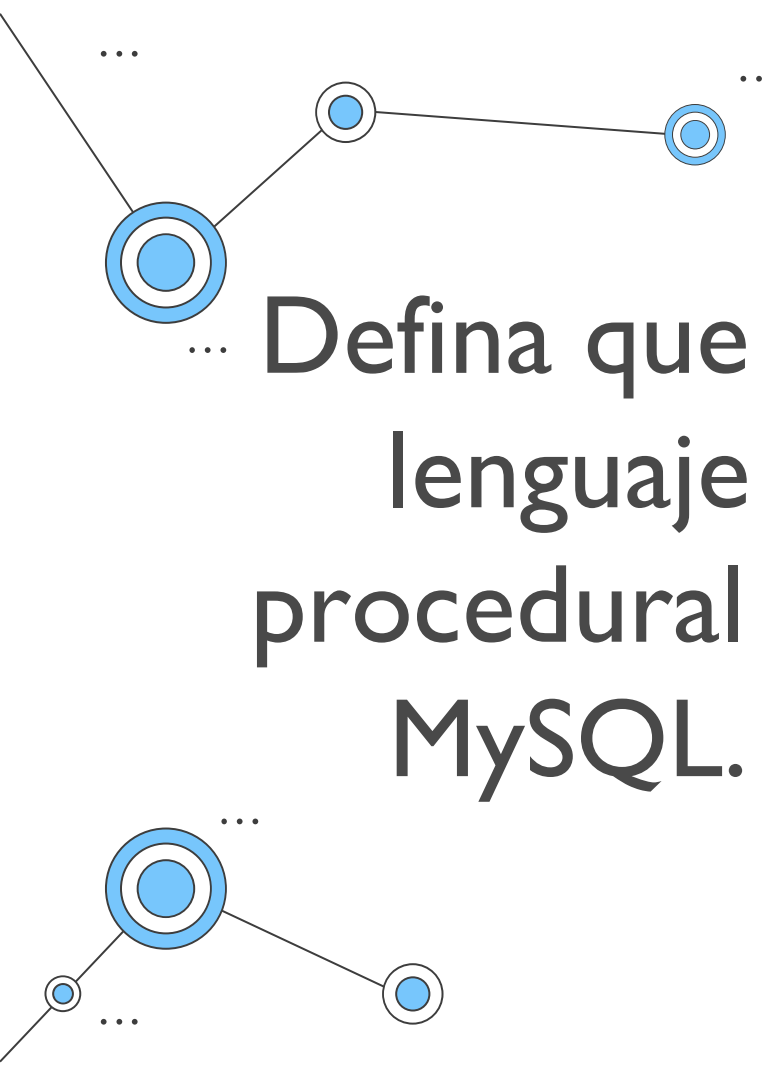



BASE DE DATOS II



... Defina que es lenguaje procedural en MySQL.

Los procedimientos almacenados MySQL contienen una o más instrucciones SQL además de un procesamiento manipulador o lógico.

La característica fundamental de los procedimientos almacenados MySQL es que estos comandos se quedan almacenados y se ejecutan en el servidor o en el motor de bases de datos.



Defina que es una función 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.



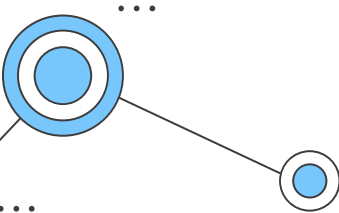
¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parámetros, etc.

NOMBRE FUNCIÓN .- Es el nombre que llevara la función que estamos creando la cual también tiene que llegar a tener un nombre único.

RETURN .- La instrucción return indica el final de la función pero también el valor que devuelve la función

PARAMETROS .- Los parámetros son locales al procedimiento, es decir, existen solamente dentro del mismo. Pueden declararse varios parámetros por procedimiento, se separan por comas. Cuando el procedimiento es ejecutado, deben explicitarse valores para cada uno de los parámetros en el orden que fueron definidos.

INSTRUCCIONES .- El cuerpo de la función, se define en un bloque "begin -end" que contiene las instrucciones que retornan el valor.



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

Crear Función

```
CREATE FUNCTION sumadosvariables(a INT, b  
INT)  
Returns int  
Begin  
Declare numero INTEGER default 0;  
Set numero =a+b;  
Return numero;  
End;
```

Modificar Función

```
CREATE OR REPLACE FUNCTION sumadosvariables (a  
int, b int)  
Returns int  
Begin  
Declare numero INTEGER default 0;  
Set numero=a+b;  
Return numero;  
End;
```

Eliminar Función

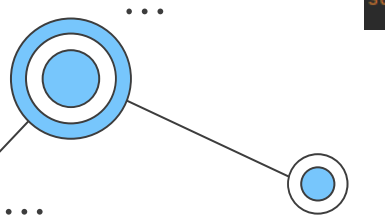
```
DROP FUNCTION sumadosvariables;
```

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

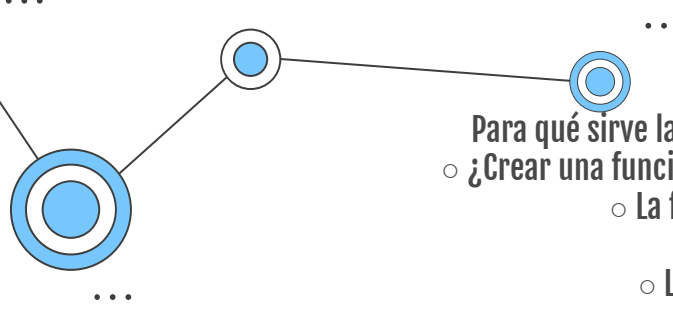
La función CONCAT funciona para unir dos o mas cadenas en una.

```
create or replace function concat_cadenas(a varchar(50),b varchar(50),c varchar(50))
returns varchar(50)
begin
declare CADENA varchar(60) default '';
set CADENA= CONCAT(a,' ',b,' ',c);
return CADENA;
end;

select concat_cadenas( a: 'demonios', b: 'gump', c: 'eres realmente un genio');
```



```
`concat_cadenas('demonios','gump','eres realmente un genio')`
demonios gump eres realmente un genio
```

- 
- Para qué sirve la función SUBSTRING y como funciona en MYSQL
- ¿Crear una función que muestre el uso de las función SUBSTRING?
 - La función recibe un nombre completo.
 - INPUT: Ximena Condori Mar
 - La función solo retorna el nombre.
 - OUTPUT: Ximena

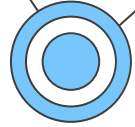
SUBSTRING .- La función SUBSTRING en SQL nos permite extraer una porción de una string, o cadena de caracteres. Esto es muy útil para realizar diversas operaciones sobre las strings, como concatenarlas, dividir las, etc



```
create or replace function substring_cadena(nombre varchar(50),posicion integer,longitud integer)
returns varchar(50)
begin
declare cadena varchar(50) default '';
set cadena= substring(nombre,posicion,longitud);
return cadena;
end;

select substring_cadena( nombre: 'Ximena Condori Mar', posicion: 1, longitud: 6);
```

```
mysql> `substring_cadena('Ximena Condori Mar',1,6)`
1 Ximena
```



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() en MySQL se usa para comparar dos strings. Si ambas strings son iguales, devuelve 0, si el primer argumento es más pequeño que el segundo según el orden definido, devuelve -1 y devuelve 1 cuando el segundo es más pequeño que el primero.

```
create or replace function comparar_cadenas (a varchar(50),b varchar(50),c varchar(50))  
returns varchar(50)  
begin  
declare respuesta varchar(50) default '';  
if strcmp(a,b)=0 then set respuesta='dos cadenas son iguales';  
elseif strcmp(a,c)=0 then set respuesta='dos cadenas son iguales';  
elseif strcmp(b,c)=0 then set respuesta='dos cadenas son iguales';  
else  
set respuesta='ninguna cadena es igual ';  
end if;  
return respuesta;  
end;  
  
select comparar_cadenas (a: 'salar de uyuni', b: 'litio para elon musk', c: 'salar de uyuni');
```

```
1 'comparar_cadenas ('salar de uyuni','litio para elon musk','salar de uyuni')'  
1 dos cadenas son iguales
```


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 .- En MySQL se usa para encontrar la longitud de una string dada en caracteres. Cuenta el número de caracteres e ignora si los caracteres son de un solo byte o de varios bytes.

```
create or replace function Clength_cadena(a varchar(50))
returns integer
begin
declare respuesta varchar(50) default '';
set respuesta=char_length(a);
return respuesta;
end;

select Clength_cadena(a: 'Maravilloso');
```

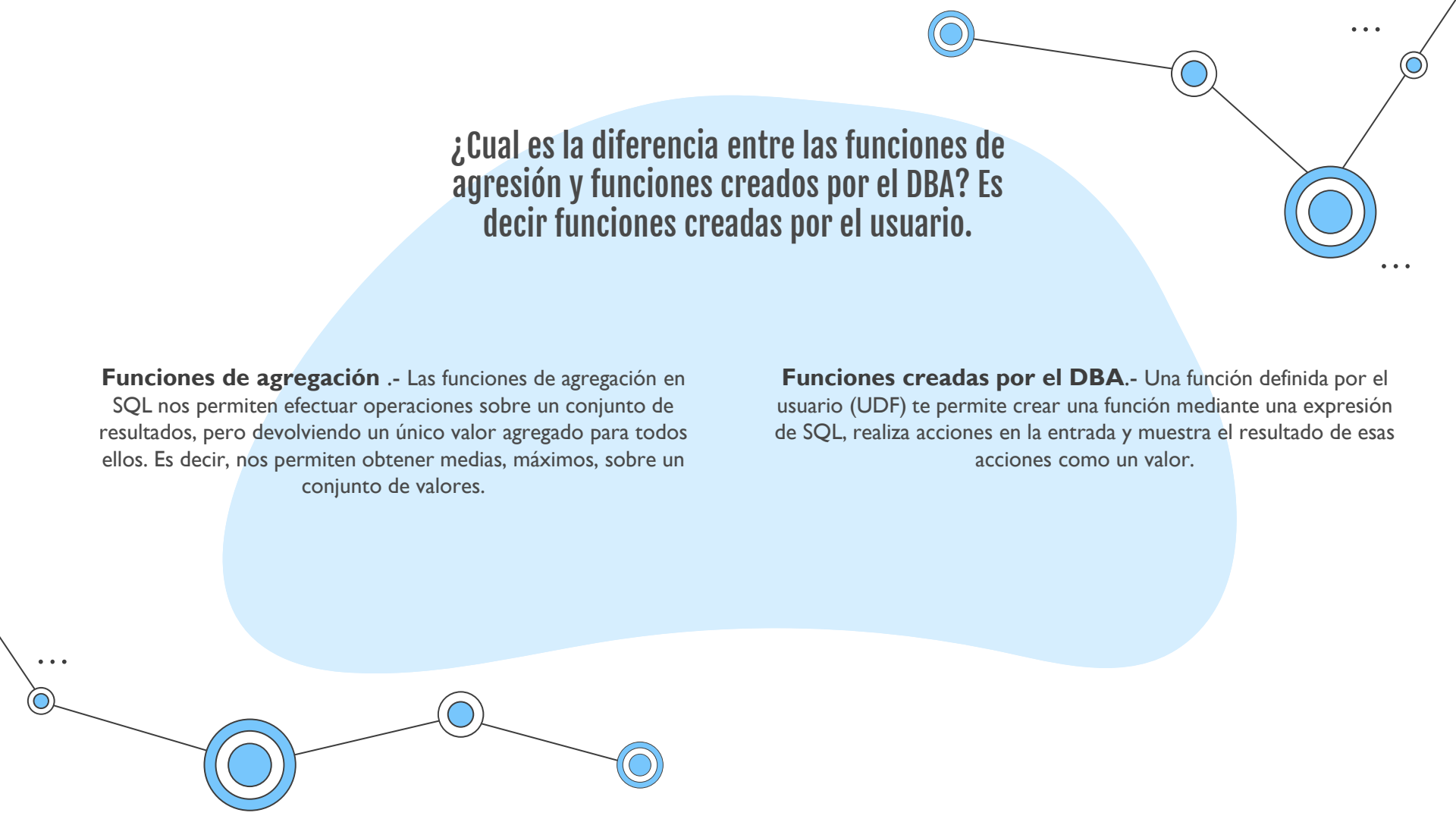
	1	11
1	`Clength_cadena('Maravilloso')`	

La función LOCATE .- En MySQL se usa para encontrar la ubicación de una substring en una string.

```
create or replace function locate_cadena(locatecadena varchar(50),cadena varchar(50))
returns integer
begin
declare resp integer default 0;
set resp=locate(cadena,locatecadena);
return resp;
end;

select locate_cadena( locatecadena: 'darksiders', cadena: 'siders');
```

	1	5
1	`locate_cadena('darksiders','siders')`	



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

Funciones de agregación .- Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos. Es decir, nos permiten obtener medias, máximos, sobre un conjunto de valores.

Funciones creadas por el DBA.- Una función definida por el usuario (UDF) te permite crear una función mediante una expresión de SQL, realiza acciones en la entrada y muestra el resultado de esas acciones como un valor.

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

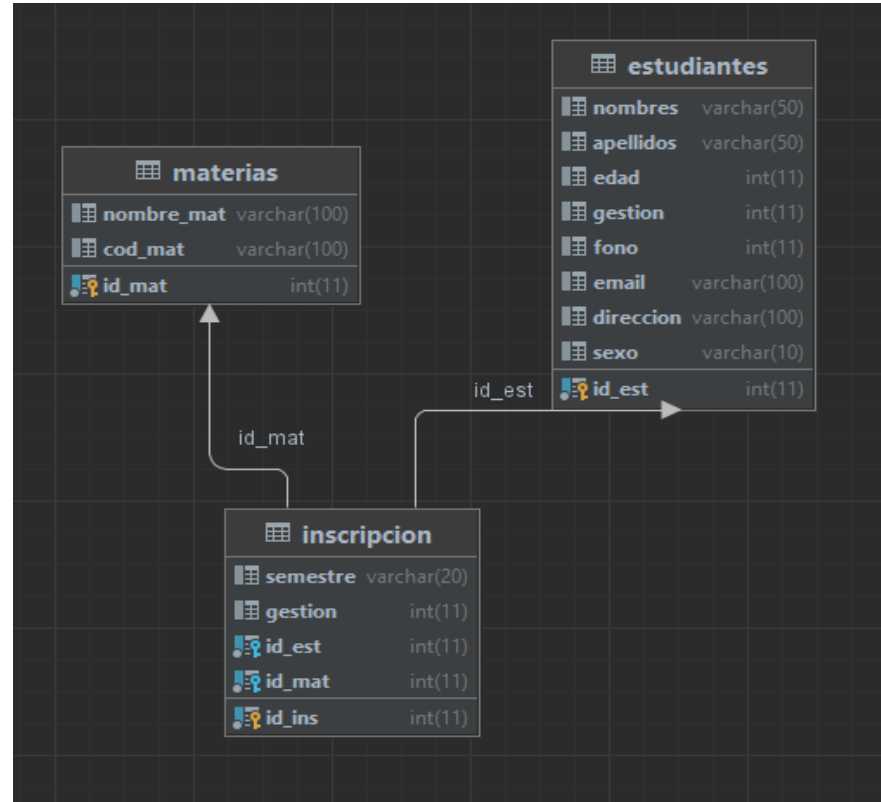
- Es decir IN INOUT, etc

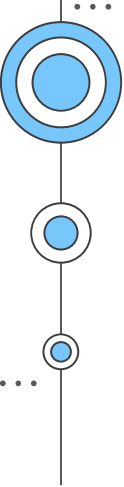
IN: Es el tipo de parámetro que se usa por defecto. La aplicación o código que invoque al procedimiento tendrá que pasar un argumento para este parámetro. El procedimiento trabajará con una copia de su valor, teniendo el parámetro su valor original al terminar la ejecución del procedimiento.

OUT: El valor de este parámetros puede ser cambiado en el procedimiento, y además su valor modificado será enviado de vuelta al código o programa que invoca el procedimiento.

INOUT: Es una mezcla de los dos conceptos anteriores. La aplicación o código que invoca al procedimiento puede pasarle un valor a éste, devolviendo el valor modificado al terminar la ejecución.

Crear la siguiente Base de datos y sus registros.

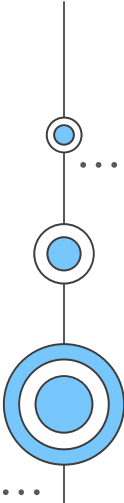




	id_est	nombres	apellidos	edad	gestion	fono	email	direccion	sexo
1	1	Miguel	Gonzales Veliz	20	<null>	2832115	miguel@gmail.com	Av. 6 de Agosto	masculino
2	2	Sandra	Mavir Uria	25	<null>	2832116	sandra@gmail.com	Av. 6 de Agosto	femenino
3	3	Joel	Aduhiri Mondar	30	<null>	2832117	joel@gmail.com	Av. 6 de Agosto	masculino
4	4	Andrea	Arias Ballesteros	21	<null>	2832118	andrea@gmail.com	Av. 6 de Agosto	femenino
5	5	Santos	Montes Valenzuela	24	<null>	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	2021	4	4



Crear una función que genere la serie Fibonacci.

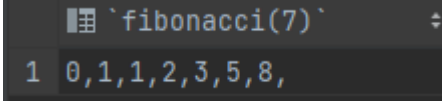
```
/*12.Crear una función que genere la serie Fibonacci.
La función recibe un limite(number)
La función debe de retornar una cadena.
Ejemplo para n=7. OUTPUT: 0, 1, 1, 2, 3, 5, 8,
Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.*/

CREATE OR REPLACE FUNCTION fibonacci(limite INT)
RETURNS TEXT
DETERMINISTIC
BEGIN
    DECLARE fibo1 INT DEFAULT 0;
    DECLARE fibo2 INT DEFAULT 1;
    DECLARE fibo3 INT DEFAULT 0;
    DECLARE str VARCHAR(255) DEFAULT '0,1,';

    IF limite = 1 THEN
        RETURN fibo1;
    ELSEIF limite = 2 THEN
        RETURN CONCAT(fibo1, fibo2);
    ELSE
        WHILE limite > 2 DO
            SET fibo3 = fibo1 + fibo2;
            SET fibo1 = fibo2;
            SET fibo2 = fibo3;
            SET limite = limite - 1;
            SET str = CONCAT(str, fibo3, ',');
        END WHILE;
        RETURN str;
    END IF;
END;

select fibonacci( limite: 7);
```

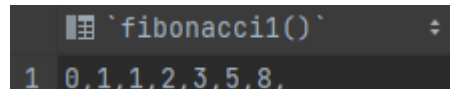
Resultado



The screenshot shows a database query result for the function `fibonacci(7)`. The output is displayed in a single row: `1 0,1,1,2,3,5,8,`. The first column contains the number 1, and the second column contains the string '0,1,1,2,3,5,8,'.

Crear una variable global a nivel BASE DE DATOS.

```
/*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.*/  
  
set @limit=7;  
CREATE OR REPLACE FUNCTION fibonacci1()  
RETURNS TEXT  
DETERMINISTIC  
BEGIN  
    DECLARE fib1 INT DEFAULT 0;  
    DECLARE fib2 INT DEFAULT 1;  
    DECLARE fib3 INT DEFAULT 0;  
    DECLARE str VARCHAR(255) DEFAULT '0,1,';  
  
    IF @limit = 1 THEN  
        RETURN fib1;  
    ELSEIF @limit = 2 THEN  
        RETURN CONCAT(fib1, fib2);  
    ELSE  
        WHILE @limit > 2 DO  
            SET fib3 = fib1 + fib2;  
            SET fib1 = fib2;  
            SET fib2 = fib3;  
            SET @limit = @limit - 1;  
            SET str = CONCAT(str, fib3,',');  
        END WHILE;  
        RETURN str;  
    END IF;  
END;  
  
select fibonacci1();
```



The screenshot shows a database query result for the function `fibonacci1()`. The output is a single row with the value `1 0,1,1,2,3,5,8,`, which represents the first seven Fibonacci numbers concatenated with commas.

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

```
/* 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 edadMinima2()
returns TEXT
begin
    declare respuesta text default '';
    declare limite int;
    declare x int default 1;
    select min(est.edad) into limite
    from estudiantes as est;

    if limite %2=0
    then
        set x=2;
    end if;
    while x<=limite do
        set respuesta= concat(respuesta,x,',');
        set x=x+2;
    end while;
    return respuesta;
END;

select edadMinima2();
```

```
`edadMinima2()`
```

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


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

```
/* 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. */  
  
create or replace function vowel_count111(str varchar(1024))  
returns text  
begin  
    return concat(  
        concat (' a: ', (LENGTH(str) - LENGTH(REPLACE(str, 'a', '')))/LENGTH('a')),  
        concat (' e: ', (LENGTH(str) - LENGTH(REPLACE(str, 'e', '')))/LENGTH('e')),  
        concat (' i: ', (LENGTH(str) - LENGTH(REPLACE(str, 'i', '')))/LENGTH('i')),  
        concat (' o: ', (LENGTH(str) - LENGTH(REPLACE(str, 'o', '')))/LENGTH('o')),  
        concat (' u: ', (LENGTH(str) - LENGTH(REPLACE(str, 'u', '')))/LENGTH('u'))  
    );  
end;  
  
select vowel_count111( str: 'trivialidades');
```

```
`vowel_count111('trivialidades')`  
1  a: 2.0000 e: 1.0000 i: 3.0000 o: 0.0000 u: 0.0000
```

Crear una función que recibe un parámetro INTEGER.

```
/*16.Crear una función que recibe un parámetro INTEGER.
o La función debe de retornar un texto(TEXT) como respuesta.
o El parámetro es un valor numérico credit_number.
o Si es mayor a 50000 es PLATINIUM.
o Si es mayor igual a 10000 y menor igual a 50000 es GOLD.
o Si es menor a 10000 es SILVER
o La función debe retornar indicando si ese cliente es PLATINUM, GOLD o
SILVER en base al valor del credit_number.
*/

create or replace function funcion_credito( CREDIT_NUMBER INTEGER)
returns text
begin
declare RESPUESTA text default '';
case
when CREDIT_NUMBER >= 50000 then set RESPUESTA = 'PLATINIUM';
when CREDIT_NUMBER >= 10000 and CREDIT_NUMBER < 50000 then set RESPUESTA = 'GOLD';
when CREDIT_NUMBER < 10000 then set RESPUESTA = 'SILVER';
else set RESPUESTA = 'NO PERTENCE A NINGUNA CLASE';
end case ;
return RESPUESTA;
end;

select funcion_credito( CREDIT_NUMBER: 10500);
```

```
1 `funcion_credito(10500)`
1 GOLD
```

Crear una función que reciba un parámetro TEXT

```
/* 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.*/

create or replace function disminuir_Cadena (CADENA TEXT)
returns text
begin
declare RESPUESTA text default '';
declare CONTADOR integer default CHAR_LENGTH(CADENA);
declare dis_cad integer default 1;
declare dis_cad2 integer default CHAR_LENGTH(CADENA);
repeat
set RESPUESTA= CONCAT(RESPUESTA,',',SUBSTRING(CADENA,dis_cad,dis_cad2));
set CONTADOR=CONTADOR-1;
set dis_cad=dis_cad+1;
until CONTADOR <=0 end repeat;
return RESPUESTA;
end;

select disminuir_Cadena( CADENA: 'DBAII');
```

```
1 `disminuir_Cadena('DBAII')`
1 ,DBAII,BAII,AII,II,I
```