



MariaDB®

Lenguaje procedural

BASE DE DATOS II

Romario Tola Quispe

Preguntas teóricas

1. Defina que es lenguaje procedural en MySQL.

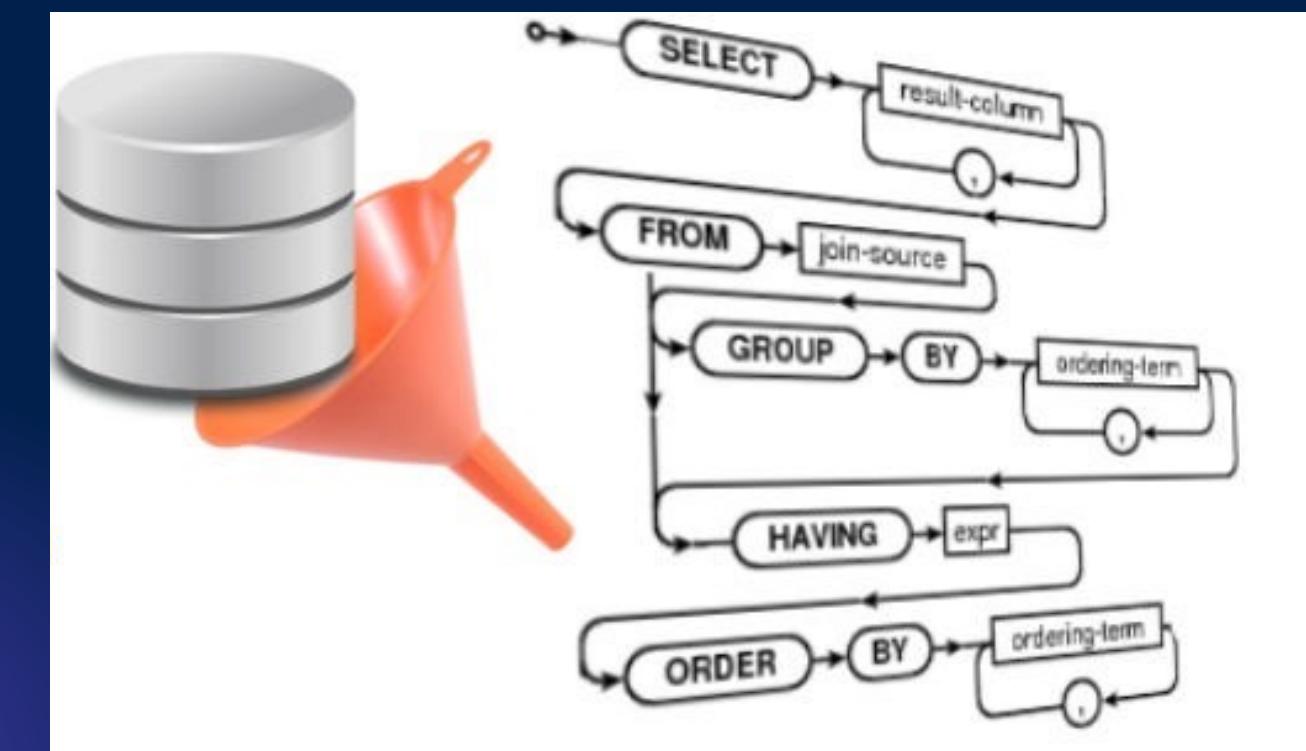


Es un tipo de lenguaje de programación que se basa en una serie de instrucciones que se ejecutan de manera secuencial para completar una tarea o resolver un problema.

El SQL procedimental de MySQL permite a los desarrolladores escribir bloques de código que se ejecutan de manera secuencial

2. Defina que es una función en MySQL.

Las funciones son piezas de código que reciben datos de entrada, realizan operaciones con ellos y luego devuelven un resultado.



3. ¿Qué características debe de tener una función?

- Nombre de la función
- Parámetros de entrada
- Tipo de retorno
- Cuerpo de la función
- Sentencia RETURN

```
CREATE FUNCTION sumar(num1 INT, num2 INT)
RETURNS INT
BEGIN
    DECLARE resultado INT;
    SET resultado = num1 + num2;
    RETURN resultado;
END;
```

4. ¿Cómo crear, modificar y cómo eliminar una función?

```
CREATE FUNCTION sumar(num1 INT, num2 INT)
RETURNS INT
BEGIN
    DECLARE resultado INT;
    SET resultado = num1 + num2;
    RETURN resultado;
END;
```

```
ALTER FUNCTION sumar(num1 INT, num2 INT, num3 INT)
RETURNS INT
BEGIN
    DECLARE resultado INT;
    SET resultado = num1 + num2 + num3;
    RETURN resultado;
END;
```

```
DROP FUNCTION IF EXISTS sumar;
```

5. Para qué sirve la función CONCAT y como funciona en MySQL

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

```
CREATE OR REPLACE FUNCTION uso_del_concat (cadena1 text, cadena2 text, cadena3 text)
returns text
begin
    declare concatenacion text ;
    set concatenacion = concat(cadena2 , ' ',cadena3, ' ',cadena1);

    return concatenacion;
end;
select uso_del_concat( cadena1: 'Quispe', cadena2: 'Romario', cadena3: 'Tola');
```

6. 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

```
CREATE OR REPLACE FUNCTION uso_del_substring (cadena text)
returns text
begin
    declare salida text default '';
    set salida = substr(cadena, 1,7);

    return salida;
end;
select uso_del_substring( cadena: 'Ximena Condori Mar');
```

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

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

```
create or replace function uso_del_strcmp(cadena1 text, cadena2 text, cadena3 text)
    returns text
begin
    -- 1 y -1 => diferentes
    -- 0 => iguales
    DECLARE respuesta text default 'NO HAY TOCAYOS';
    IF (strcmp(cadena1,cadena2)=0 OR strcmp(cadena1,cadena3)=0 OR strcmp(cadena3,cadena2)=0 ) THEN
        SET respuesta = 'SI HAY TOCAYOS';
    end if;
    RETURN respuesta;
end;

select uso_del_strcmp( cadena1: 'Juan',  cadena2: 'Esteban',  cadena3: 'Juana');
```

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?

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

Las funciones de agregación son funciones predefinidas que se utilizan comúnmente para resumir datos, mientras que las funciones creadas por el usuario son específicas para el conjunto de datos y las operaciones que se necesiten realizar.



■ Creación de una función

```
USE Northwind  
GO  
CREATE FUNCTION fn_NewRegion  
    (@myinput nvarchar(30))  
RETURNS nvarchar(30)  
BEGIN  
    IF @myinput IS NULL  
        SET @myinput = 'Not Applicable'  
    RETURN @myinput  
END
```

10. ¿A qué se referirá cuando se habla de parámetros de entrada y salida en MySQL?

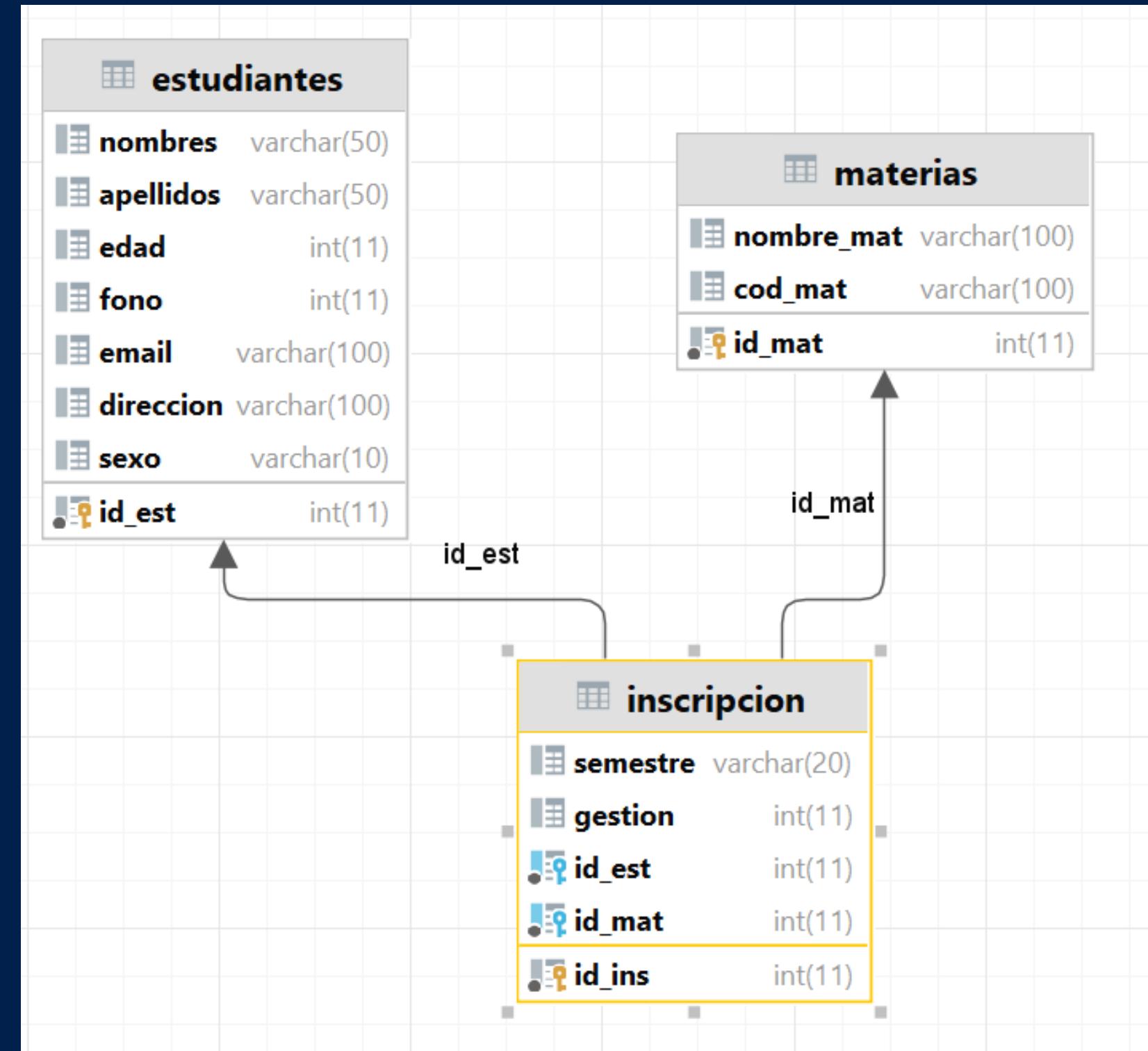
En MySQL, los parámetros de entrada y salida se refieren a los valores que se pasan a una función almacenada y los valores que se devuelven por esa función, respectivamente.

Para definir parámetros de entrada y salida en MySQL, se utilizan las palabras clave IN y OUT, respectivamente, en la declaración de la función almacenada.

```
CREATE FUNCTION FUNCION(param1 INT, param2 INT, OUT result INT)
BEGIN
    SET result = param1 + param2;
END;
SELECT FUNCION( param1: 1, param2: 2)
```

Preguntas prácticas

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 Urias	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

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 fibonacci_serie(limite INT)
RETURNS TEXT
BEGIN
    DECLARE cadena TEXT DEFAULT '';
    DECLARE i INT DEFAULT 0;
    DECLARE num INT DEFAULT 0;
    DECLARE nuevoNumero INT DEFAULT 1;
    DECLARE acum INT;

    WHILE (i < limite) DO
        SET cadena = concat(cadena, num, ',');
        SET acum = num;
        SET num = nuevoNumero;
        SET nuevoNumero = acum + nuevoNumero;

        SET i = i+1;
    END WHILE;

    RETURN cadena;

```

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

13. Crea una variable global a nivel BASE DE DATOS.

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

- Crea una variable global de nombre LIMIT.
- Este valor debe almacenar un valor entero.
- Ejemplo, LIMIT = 7
- OUTPUT: 0,1,1,2,3,5,8
- Crea 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.

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 descendemente todos los impares hasta el valor 0.**
- **Retornar la nueva cadena concatenada.**

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.

The screenshot shows a database interface with a query window and an output window.

In the query window, the following SQL code is written:

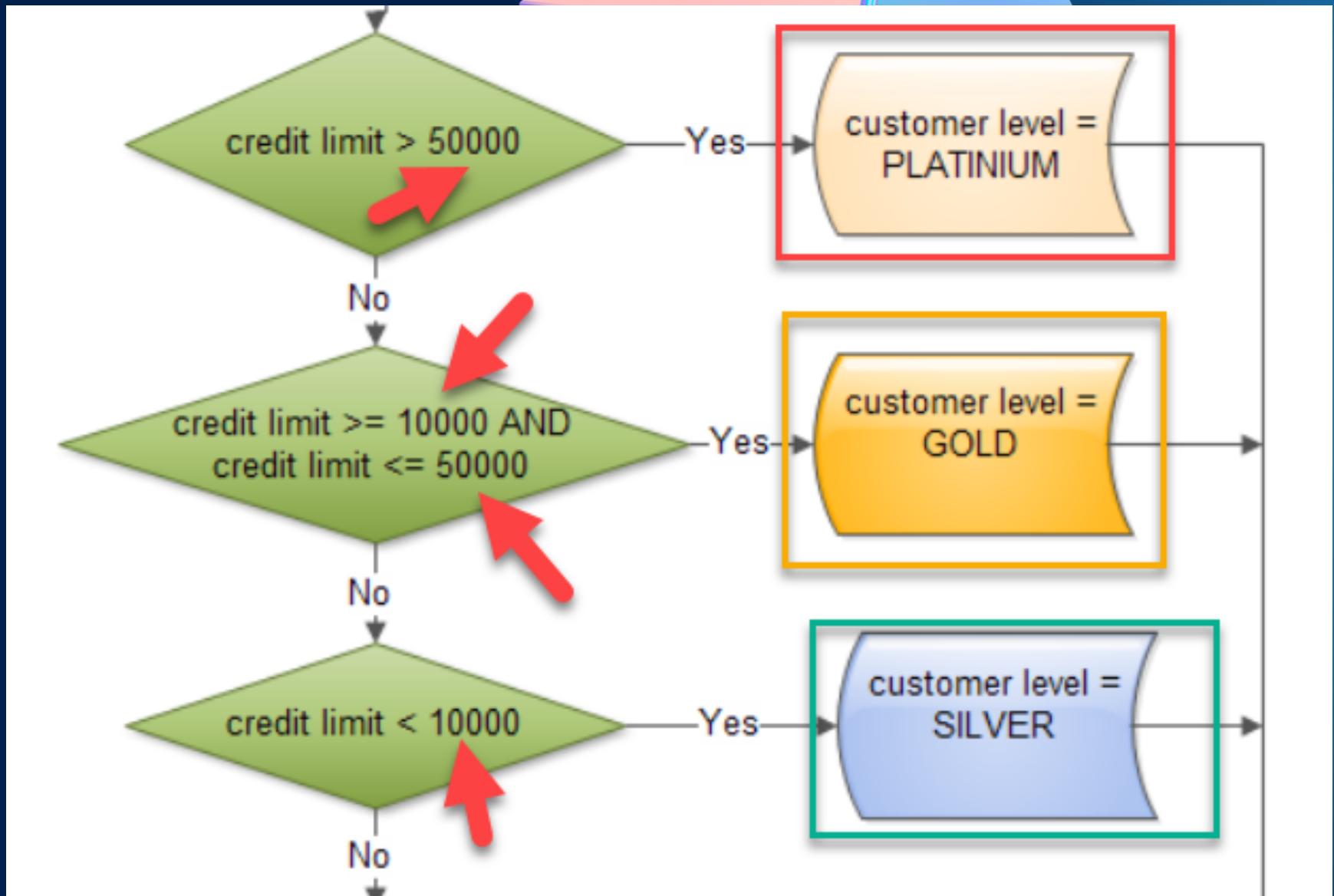
```
SELECT separaVocales( cadena: 'taller de base de datos' ) AS SEPARA_VOCALES;
```

In the output window, the results are displayed in a table:

	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 PLATINUM.
- 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.



```
CREATE OR REPLACE FUNCTION numero_de_credito(credit_number INT)
RETURNS text
BEGIN
    DECLARE respuesta TEXT DEFAULT '';
    CASE
        WHEN (credit_number > 50000) THEN set respuesta = 'PLATINUM';
        WHEN (credit_number >= 10000) AND credit_number <= 50000 THEN set respuesta = 'GOLD';
        WHEN (credit_number < 10000) THEN set respuesta = 'SILVER';
    END CASE;
    return respuesta;
end;
SELECT numero_de_credito( credit_number: 50001);
```

Output numero_de_credito(50001):text ×

1 row ▾

	numero_de_credito(50001)
1	PLATINUM

17. Crea^r 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.

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,

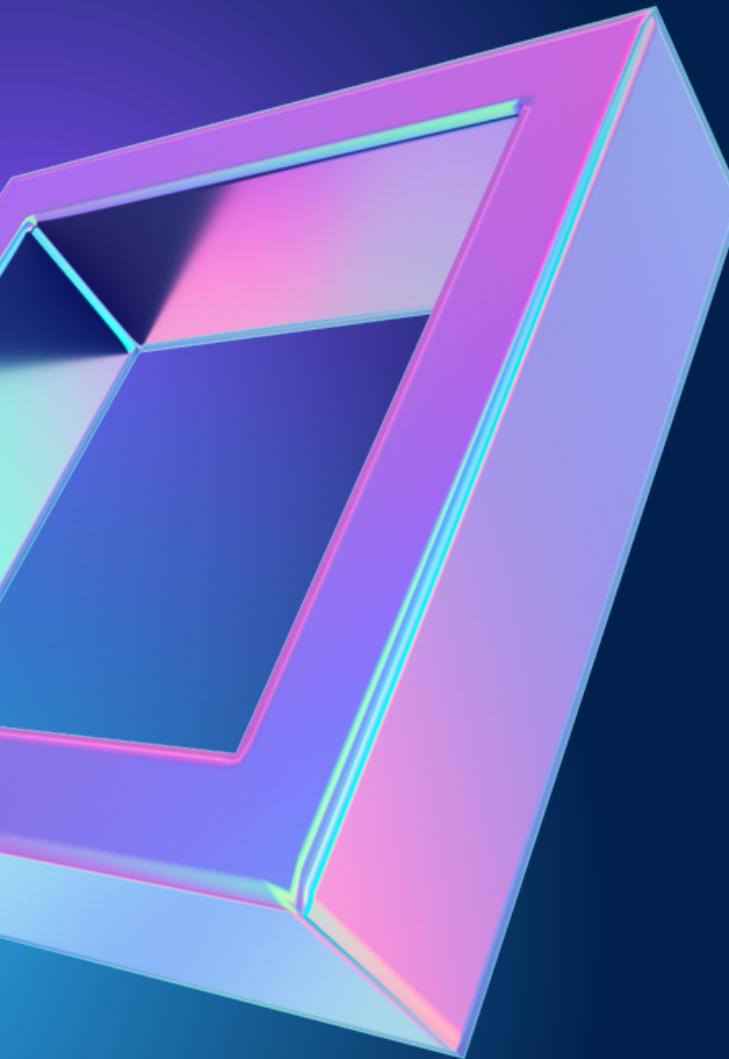
```
CREATE OR REPLACE FUNCTION reduce_cadena(cadena TEXT)
RETURNS TEXT
BEGIN
    DECLARE respuesta TEXT DEFAULT cadena;
    DECLARE cadena_cortada text default substr(cadena,2);
    declare contador int default 3;

    REPEAT
        SET respuesta = concat(respuesta, ', ',cadena_cortada);
        set cadena_cortada = substr(cadena,contador);
        set contador = contador +1;

    UNTIL  cadena_cortada = '' END REPEAT;
    return respuesta;
END;
SELECT reduce_cadena( cadena: 'dbaii');
```

Output `reduce_cadena('dbaii'):text`

	reduce_cadena('dbaii')
1	dbaii, baii, ai, i



GRACIAS!!