



# EVALUACIÓN BDA-II H3

SERGIO ANDRES MENDOZA ALVARADO

# ¿QUE ES EL LENGUAJE PROCEDURAL EN MYSQL?

- El lenguaje procedural en bases de datos se refiere a un conjunto de instrucciones y estructuras de control que permiten la creación de programas o rutinas que pueden ser almacenados en la base de datos y ejecutados dentro del motor de la misma.

```
CREATE OR REPLACE FUNCTION separar_vocales(cadena TEXT)
RETURNS TEXT
BEGIN

    DECLARE letter_a, letter_e, letter_i, letter_o, letter_u INT DEFAULT 0;
    DECLARE b,c,d INT DEFAULT 1;
    DECLARE text1,text2,text3, resp TEXT DEFAULT '';

    WHILE b <= CHAR_LENGTH(cadena) DO

        SET text1 = SUBSTRING(cadena, b, 1);
        IF text1 = 'a' THEN
            SET letter_a = letter_a+1;
        end if;

    END WHILE;

    SET resp = letter_a;

    RETURN resp;
END;
```

```
CREATE FUNCTION nombre_de_la_funcion()
```

Creación de una función

```
CREATE FUNCTION nombre_de_la_funcion()  
  
RETURNS TIPO_DE_DATO_QUE_RETORNARA
```

Parámetro y tipo de dato  
que retornara

```
CREATE FUNCTION nombre_de_la_funcion()  
  
RETURNS TIPO_DE_DATO_QUE_RETORNARA  
  
BEGIN  
  
    RETURN valor_a_retornar;  
  
END;
```

Inicio y final de la funcion

¿QUÉ ES UNA FUNCIÓN? PARTES DE UNA FUNCIÓN

# ¿CÓMO CREAR, ELIMINAR Y MODIFICAR UNA FUNCIÓN?

Para crear una función simplemente debemos contrar con los parámetros que pide una función, un parámetro, su RETURNS, begin, end, con eso creamos una funcion

```
154 CREATE FUNCTION nombre_de_la_funcion()  
155     RETURNS TEXT  
156     BEGIN  
157  
158     DECLARE a TEXT;  
159     SET a = 'Hola mundo';  
160     RETURN a;  
161 END;  
162  
163 ✓ SELECT nombre_de_la_funcion();
```

nombre\_de\_la\_funcion()

Output
nombre_de_la_funcion():text
1 Hola mundo

Para modificar una función escribimos entre CREATE FUNCTION lo siguiente: OR REPLACE.  
Para eliminar la función escribimos:  
DROP FUNCTION nombreDeLafuncion;

```
53  
54 create OR REPLACE function nombre_de_la_funcion()  
55     RETURNS TEXT  
56     BEGIN  
57  
58     DECLARE a TEXT;  
59     SET a = 'Hola mundo';  
60     RETURN a;  
61 END;  
62  
63 ✓ DROP FUNCTION nombre_de_la_funcion;  
64
```

```
CREATE FUNCTION concatenador()  
RETURNS TEXT  
BEGIN  
  
    DECLARE a,b,c,r TEXT;  
    SET a = 'Hola';  
    SET b = ' Como';  
    SET c = ' estas causa?';  
  
    SET r = (SELECT CONCAT(a,b,c));  
    RETURN r;  
END;  
  
SELECT concatenador();
```

Output concatenador():text

	`concatenador()``
1	Hola Como estas causa?

Bien

```
262  
263 CREATE OR REPLACE FUNCTION concatenador()  
264 RETURNS TEXT  
265 BEGIN  
266  
267     DECLARE a,b,c,r TEXT;  
268     SET a = 'Hola';  
269     SET b = ' Como';  
270     SET c = ' estas causa?';  
271  
272     SET r = a+b+c;  
273     RETURN r;  
274     END;  
275  
276 SELECT concatenador();
```

[22007][1292] (conn=3) Truncated incorrect DOUBLE value: ' estas causa?'

Mal

¿PARA QUE SIRVE LA FUNCIÓN CONCAT?

Nos ayuda a concatenar datos, tanto como cadenas como enteros u otros, debemos recordar que la forma de concatenar en MySQL no es igual comparada a un lenguaje de programación como Python, C# o Java, es programación basada en base de datos, y eso siempre debemos recordar.

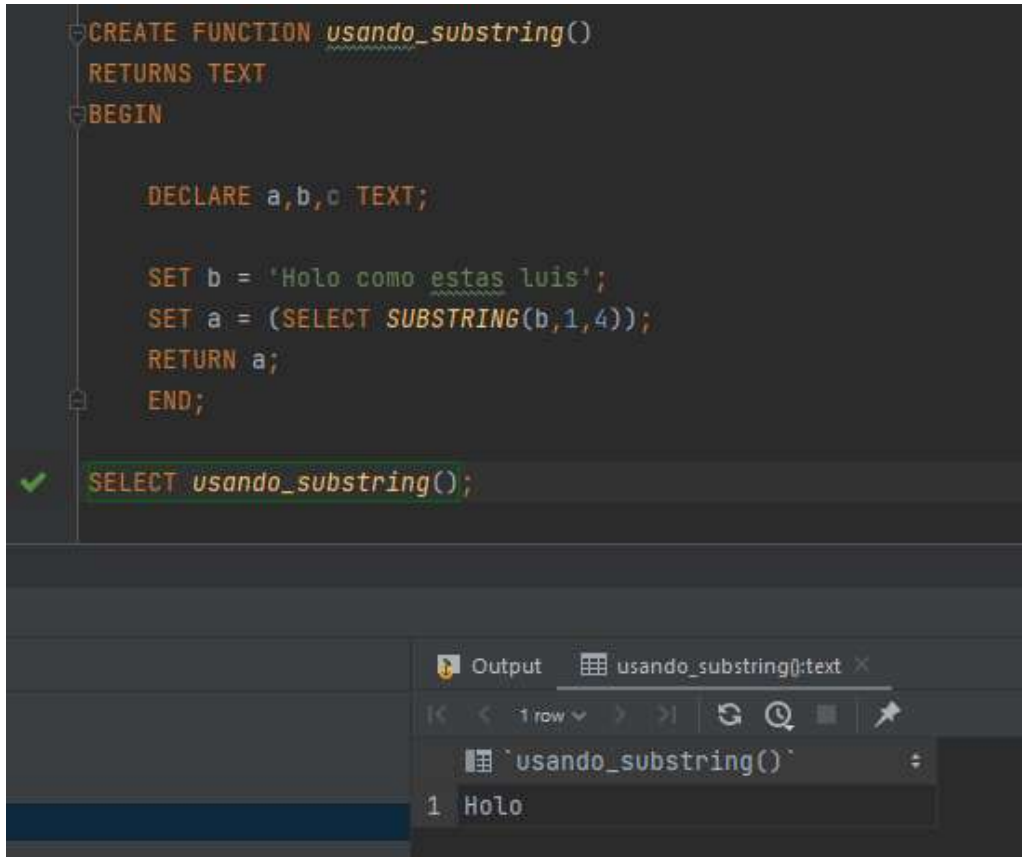


# ¿PARA QUÉ SIRVE LA FUNCIÓN SUBSTRING Y COMO FUNCIONA EN MYSQL?

```
CREATE FUNCTION usando_substring()
RETURNS TEXT
BEGIN
    DECLARE a,b,c TEXT;

    SET b = 'Holo como estas luis';
    SET a = (SELECT SUBSTRING(b,1,4));
    RETURN a;
END;

SELECT usando_substring();
```



usando_substring()
Holo

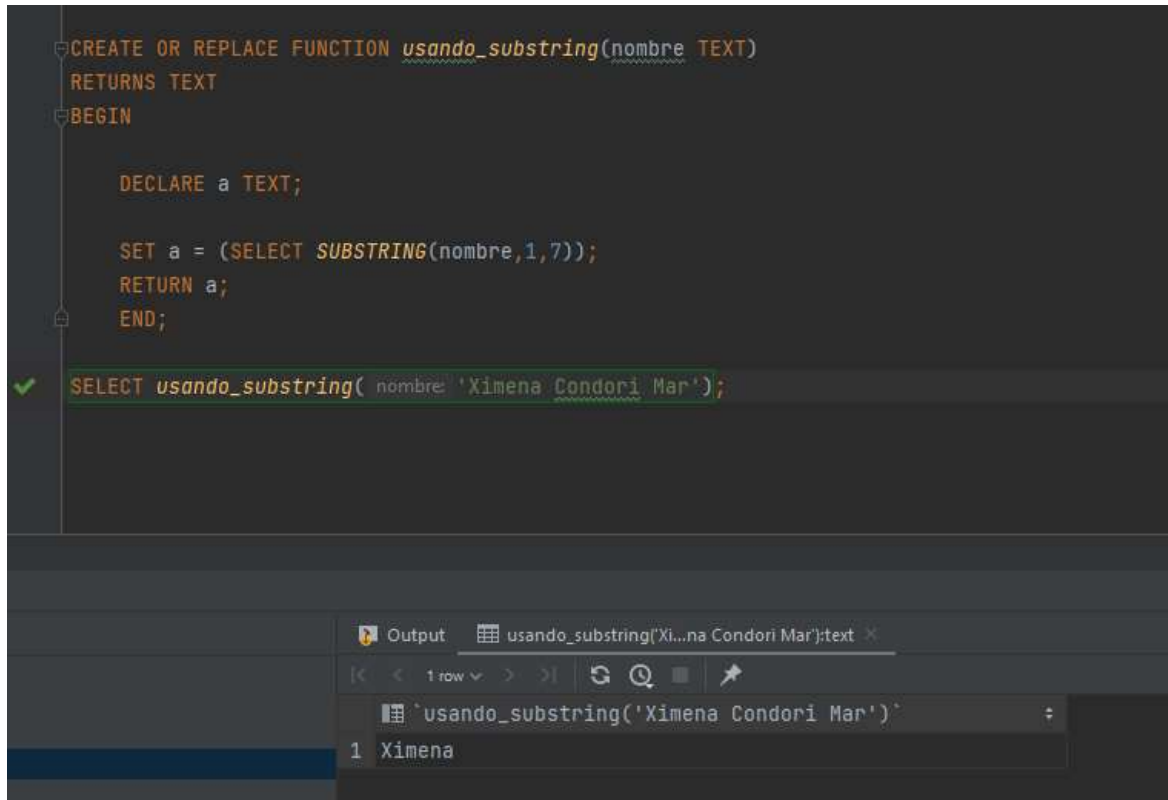
La función SUBSTRING recibe como parámetros 3 datos, como primer parámetro una cadena, como segundo parámetro de donde empezara a contar la cadena, y como tercer parámetro hasta donde contara, luego de realizar ese proceso como se ve en la imagen empieza desde 1 y termina en 4, por lo que imprime solamente “Holo” en vez de “Holo como estas luis”.

## EJEMPLO 6

```
CREATE OR REPLACE FUNCTION usando_substring(nombre TEXT)
  RETURNS TEXT
  BEGIN
    DECLARE a TEXT;

    SET a = (SELECT SUBSTRING(nombre,1,7));
    RETURN a;
  END;

SELECT usando_substring( nombre: 'Ximena Condori Mar');
```



Output
usando_substring('Ximena Condori Mar'):text
1 Ximena

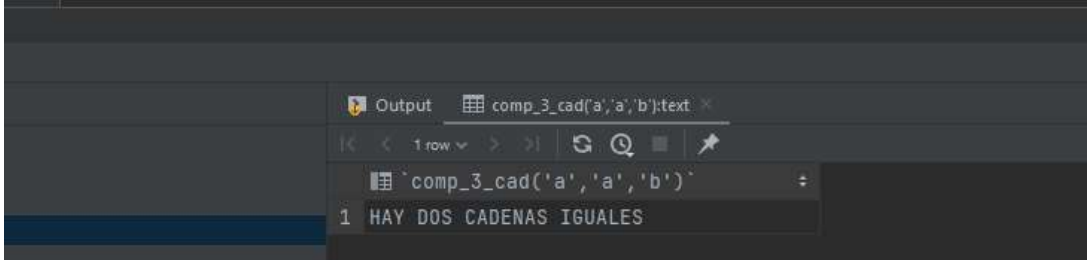
Recibimos como parámetro un nombre, en este caso el nombre de “Ximena Condori Mar”, ese es nuestro input, y utilizando la función de agregación SUBSTRING podemos acortar el nombre como nos pide el ejercicio, dándonos como output solamente “Ximena”.

# ¿PARA QUÉ SIRVE LA FUNCIÓN STRCMPY COMO FUNCIONA EN MYSQL?

```
CREATE OR REPLACE FUNCTION comp_3_cad(nom1 TEXT, nom2 TEXT, nom3 TEXT)
  RETURNS TEXT
  BEGIN
    DECLARE message TEXT;

    IF STRCMP(nom1, nom3) OR STRCMP(nom2, nom3) = FALSE THEN
      SET message = 'HAY DOS CADENAS IGUALES';
    ELSE
      SET message = 'NO HAY CADENAS MAYOR A DOS QUE SEAN IGUALES';
    end if;
    RETURN message;
  END;

SELECT comp_3_cad( nom1: 'a', nom2: 'a', nom3: 'b');
```



The screenshot shows a MySQL IDE with a dark theme. The main editor contains a MySQL script that defines a function named `comp_3_cad`. The function takes three text arguments: `nom1`, `nom2`, and `nom3`. It declares a local variable `message` of type `TEXT`. An `IF` statement checks if `STRCMP(nom1, nom3) OR STRCMP(nom2, nom3) = FALSE`. If true, it sets `message` to `'HAY DOS CADENAS IGUALES'`; otherwise, it sets `message` to `'NO HAY CADENAS MAYOR A DOS QUE SEAN IGUALES'`. The function then returns `message`. Below the function definition, a `SELECT` statement calls `comp_3_cad( 'a', 'a', 'b')`. At the bottom, the 'Output' window shows the result of the query: a single row with the value `HAY DOS CADENAS IGUALES`.

La función STRCMP compara dos cadenas, y dependiendo si son iguales o no, retorna un valor booleano, es por ello que aprovechamos ese valor booleano en este ejercicio, y si al hacer la comparación 2 cadenas son iguales entonces nos dira que hay dos cadenas iguales, en otro caso nos dirá que no hay cadenas iguales.



## Caso 1: Cuando hay una “h”

```
CREATE OR REPLACE FUNCTION char_and_locate()
RETURNS TEXT
BEGIN
    DECLARE a,b,c TEXT;

    SET a = 'hola'; #MIRAR AQUI
    SET b = (SELECT CHAR_LENGTH(a));

    IF b = 4 THEN
        SET c = (SELECT CONCAT('En la posicion: ',LOCATE('H',a),' hay una "h"'));
    ELSE
        SET c = 'no hay la letra H en esta cadena';
    END IF;

    RETURN c;
END;

SELECT char_and_locate();
```

Output | char\_and\_locate:text

	char_and_locate()
1	En la posicion: 1 hay una "h"

## Caso 2: Cuando no hay una H

```
CREATE OR REPLACE FUNCTION char_and_locate()
RETURNS TEXT
BEGIN
    DECLARE a,b,c TEXT;

    SET a = 'ola'; #MIRAR AQUI
    SET b = (SELECT CHAR_LENGTH(a));

    IF b = 4 THEN
        SET c = (SELECT CONCAT('En la posicion: ',LOCATE('H',a),' hay una "h"'));
    ELSE
        SET c = 'no hay la letra H en esta cadena';
    END IF;

    RETURN c;
END;

SELECT char_and_locate();
```

Output | char\_and\_locate:text

	char_and_locate()
1	no hay la letra H en esta cadena

## ¿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 predeterminadas con el respectivo lenguaje de programación, como por ejemplo en el caso de Python la función “.lower” nos ayuda a evitar confusiones con cadenas que estén escritas en mayúsculas o minúsculas, ahora como vimos en MYSQL tenemos muchas funciones de agregación, tales como unas cuantas con MAX, MIN, AVG, CONCAT, SUBSTRING, CHAR\_LENGTH, LENGTH, LOCATE y entre otras mas, son muy útiles para ahorrarnos código, supongamos que no existiera la función CONCAT entonces nosotros mismos tendríamos que desarrollar tal función, esa seria una función creada por el usuario, al igual que la funciones de agregación esta nos ayudara a solucionar un problemas mas concreto respecto a nuestros requerimientos.

# ¿BUSQUE Y DEFINA A QUÉ SE REFERIRÁ CUANDO SE HABLA DE PARÁMETROS DE ENTRADA Y SALIDA EN MYSQL?

Cuando se habla de parámetros de entrada y salida en MySQL, se refiere a los valores que se pueden pasar a una función o procedimiento almacenado para que realice ciertas operaciones y devuelva resultados.

Los parámetros de entrada son valores que se pasan a una función o procedimiento almacenado cuando se llama. Estos valores se utilizan dentro de la función o procedimiento para realizar las operaciones necesarias y producir un resultado. Por ejemplo, en una función que calcule la suma de dos números, los dos números serían los parámetros de entrada.



GRACIAS