



# ACTIVIDAD FINAL

## HITO 3

**Docente:** Ing. William Roddy Barra Paredes

**Nombre completo:** Ludwing Antoni Vargas Ibarra

**Semestre:** III/2022

# Defina que es lenguaje procedural en MySQL.

- Lenguajes procedurales o procedimentales: El usuario da órdenes para que se realicen las tareas pertinentes con el objetivo de recuperar los datos requeridos. Es la base del lenguaje de consulta SQL.

# 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.

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

```
CREATE FUNCTION max_edad_est_4maculino(nombres VARCHAR(20), apellidos VARCHAR(20),eda integer)
RETURNS INTEGER
BEGIN
  return
  (
    SELECT max(est.edad)
    FROM estudiantes AS est
    where est.nombres = nombres AND est.edad > eda
  );
END;
select max_edad_est_4maculino( nombres: 'Miguel', apellidos: 'Gonzales Veliz', eda: 18);
```

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

```
CREATE or replace FUNCTION SUM_edad (genero varchar(20))
  RETURNS INTEGER
  BEGIN
    return
    (
      SELECT SUM(est.edad)
      FROM estudiantes AS est
      WHERE est.sexo = genero
    );
  END;

SELECT est.nombres, est.apellidos
  FROM estudiantes AS est
 where SUM_edad( genero: 'masculino') % 2 = 0;
```

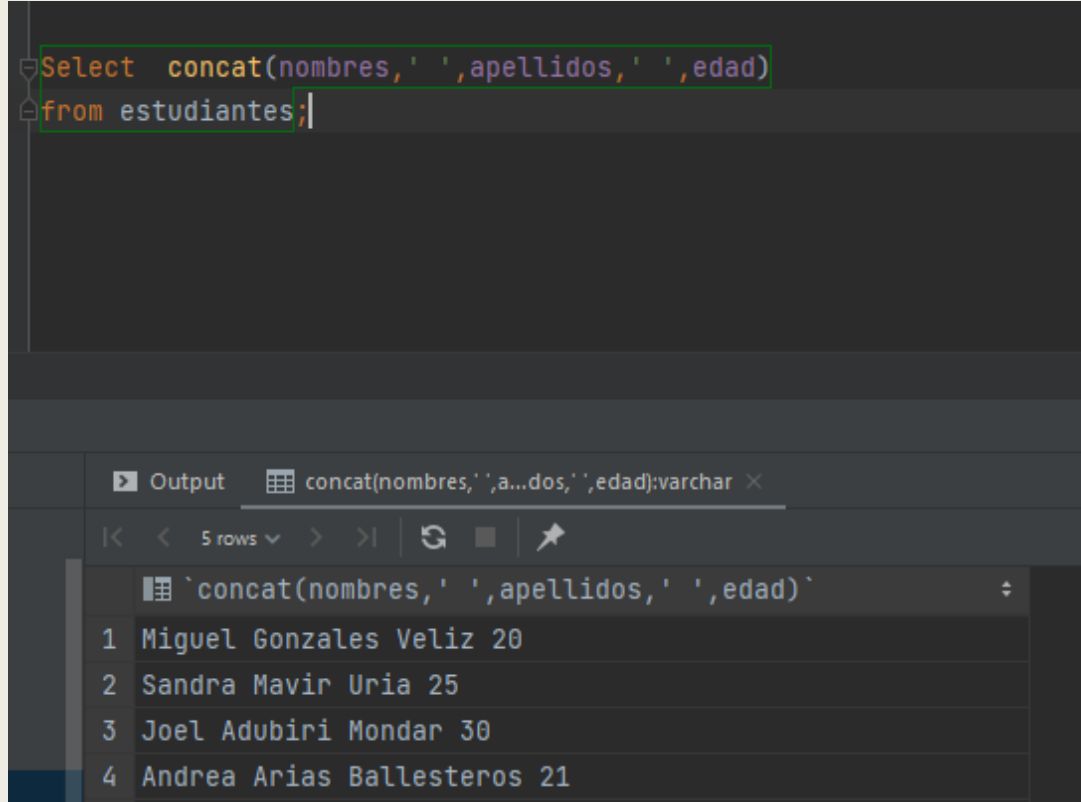
Output tareahito2.estudiantes

	nombres	apellidos
1	Miguel	Gonzales Veliz
2	Sandra	Mavir Uria
3	Joel	Adubiri Mondar
4	Andrea	Arias Ballesteros

# Para qué sirve la función CONCAT y como funciona en MYSQL

- La función CONCAT puede concatenar o combinar valores de varias columnas en una sola columna. La sintaxis de la función CONCAT es la siguiente:

```
Select concat(nombres, ' ', apellidos, ' ', edad)
from estudiantes;
```



	concat(nombres, ' ', apellidos, ' ', edad): varchar
1	Miguel Gonzales Veliz 20
2	Sandra Mavir Uria 25
3	Joel Adubiri Mondar 30
4	Andrea Arias Ballesteros 21

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

- La función de subcadena de MySQL se utiliza para extraer una subcadena o una parte de la cadena contra la cadena de entrada. Como sugiere el nombre, la función Substring opera en una cadena de entrada y devuelve una subcadena más pequeña contra las opciones especificadas.

```
create or replace function subCadena(cadena varchar(20), position integer, position2 integer)
returns text
begin
declare subCadena text default '';
set subCadena = substr(cadena, position, position2);
return subCadena;
end;

select subCadena( cadena: 'Ximena Condori Mar', position: 1, position2: 7);
```

Output: subCadena('Ximena Condori Mar',1,7):text

1 row

1 Ximena

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

- 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.

```
-->
declare verificaIguales integer;
declare mensajeretorno text default '';
set verificaIguales = strcmp(cadena1,cadena2);
if (verificaIguales = 0)
then
set mensajeretorno = concat('CADENAS IGUALES', ' : ', verificaIguales);
else
set mensajeretorno = concat('CADENAS DIFERENTES', ' : ', verificaIguales);
end if;
end;
select comparaCadenas( cadena1: 'dba II', cadena2: 'dba II');

SELECT nombres, apellidos,
STRCMP(nombres, apellidos) AS Cmp_Value
FROM estudiantes;
```

Output Result 44 x

5 rows

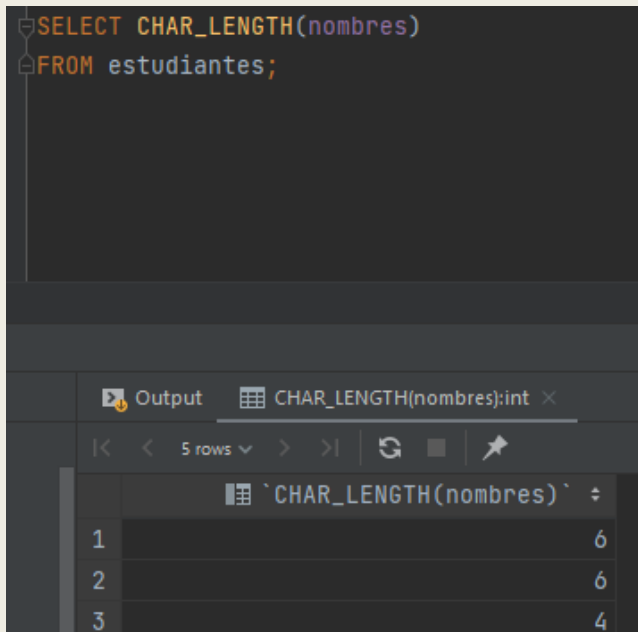
	nombres	apellidos	Cmp_Value
3	Joel	Adubiri Mondar	1
4	Andrea	Arias Ballesteros	-1
5	Santos	Montes Valenzuela	1



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

- 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.
- La función LOCATE() en MySQL se usa para encontrar la ubicación de una substring en una string. Devolverá la ubicación de la primera aparición de la substring en la string. Si la substring no está presente en la string, devolverá 0

```
SELECT CHAR_LENGTH(nombres)
FROM estudiantes;
```

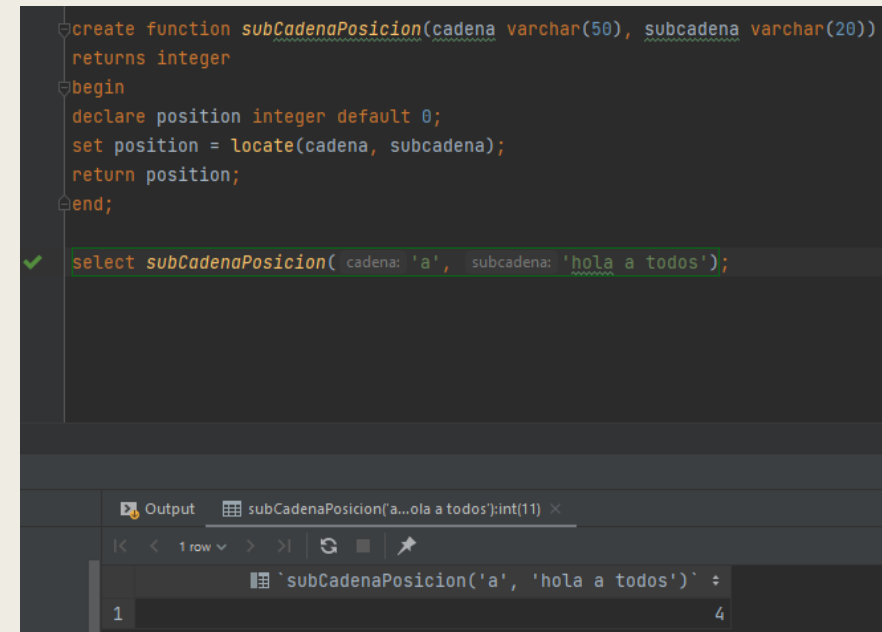


The screenshot shows a MySQL query result for the query `SELECT CHAR_LENGTH(nombres) FROM estudiantes;`. The output is displayed in a table with 5 rows. The first three rows show the character length of the names in the 'nombres' column. The values are 6, 6, and 4.

	CHAR_LENGTH(nombres)
1	6
2	6
3	4

```
create function subCadenaPosicion(cadena varchar(50), subcadena varchar(20))
returns integer
begin
declare position integer default 0;
set position = locate(cadena, subcadena);
return position;
end;

select subCadenaPosicion(cadena: 'a', subcadena: 'hola a todos');
```



The screenshot shows a MySQL query result for the query `select subCadenaPosicion(cadena: 'a', subcadena: 'hola a todos');`. The output is displayed in a table with 1 row. The value is 4.

	subCadenaPosicion('a', 'hola a todos')
1	4

# ¿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 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, etc... sobre un conjunto de valores.

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

- Los procedimientos almacenados pueden recibir y devolver información; para ello se emplean parámetros, de entrada y salida, respectivamente.
- Veamos los primeros. Los parámetros de entrada posibilitan pasar información a un procedimiento.
- Para que un procedimiento almacenado admita parámetros de entrada se deben declarar variables como parámetros al crearlo

# .Crear una función que genere la serie Fibonacci.

```
create or replace function concatenaNumerosPares(numLimit integer)
returns varchar(200)
begin
declare str varchar(200) default '';
declare x integer default 0;
declare y integer default 0;
declare z integer default 0;
declare c integer;
WHILE y <= numLimit DO
set str = concat(str, y, ',');
set c=y+z;
set y=z;
set z=c;
set y = y + 1;
END WHILE ;
return str;
end;
select concatenaNumerosPares(7);
```

```
begin
declare str varchar(200) default '';
declare x integer default 0;
declare y integer default 0;
declare z integer default 0;
declare c integer;
WHILE y <= numLimit DO
set str = concat(str, y, ',');
set c=y+z;
set y=z;
set z=c;
set y = y + 1;
END WHILE ;
return str;
end;
select concatenaNumerosPares( numLimit: 7);
```

concatenaNumerosPares()

Output usoDeRepeat(10):text x concatenaNumero

1 row

concatenaNumerosPares(7)

1 0,1,1,2,3,5,

# Crear una variable global a nivel BASE DE DATOS.

```
set @userAll = 'ADMIN';

create function global()
returns text
begin
    return @userAll;
end;

select global();
```

```
set @userAll = 'ADMIN';

create function global()
returns text
begin
    return @userAll;
end;

select global();
```

Output

usoDeRepeat(10):text

1 row

@userAll

1	ADMIN
---	-------

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

```
CREATE or replace FUNCTION min_edad(nombres VARCHAR(20), apellidos
VARCHAR(20),eda integer)
RETURNS INTEGER
BEGIN
return
(
SELECT max(est.edad)
FROM estudiantes AS est
where est.nombres = nombres AND est.edad < eda
);
END;
CREATE or replace FUNCTION usoDeRepeat(eda INTEGER) RETURNS TEXT
BEGIN
DECLARE respuesta TEXT DEFAULT '';
DECLARE x INTEGER DEFAULT eda;
REPEAT
IF (x % 2 = 0)
THEN
SET respuesta = CONCAT(respuesta, x, ' ');
ELSE
SET respuesta = CONCAT(respuesta, x, ' ');
END IF;
SET x = x - 2;
UNTIL x <= 0 END REPEAT ;
RETURN respuesta;
END;
SELECT usoDeRepeat(min_edad('Miguel','Gonzales Veliz',18));
```

```
DECLARE respuesta TEXT DEFAULT '';
DECLARE x INTEGER DEFAULT eda;
REPEAT
IF (x % 2 = 0)
THEN
SET respuesta = CONCAT(respuesta, x, ' ');
ELSE
SET respuesta = CONCAT(respuesta, x, ' ');
END IF;
SET x = x - 2;
UNTIL x <= 0 END REPEAT ;
RETURN respuesta;
END;
SELECT usoDeRepeat(eda: min_edad(nombres: 'Miguel', apellidos: 'Gonzales Veliz', eda: 18));
```

Output    usoDeRepeat(10);text ×    usoDeRepeat(max\_edad...ales Veliz',18);text ×

1 row ▾    ↺    ⏏    ⚡

```
`usoDeRepeat(max_edad_est_4maculino('Miguel','Gonzales Veliz',18))`
```

1	20	18	16	14	12	10	8	6	4	2
---	----	----	----	----	----	----	---	---	---	---

# Crear una función que recibe un parámetro INTEGER.

```
CREATE or replace FUNCTION usoDeRepeat(credit_number INTEGER)
RETURNS TEXT
BEGIN
DECLARE creditlimit INTEGER DEFAULT credit_number;
CASE
WHEN creditlimit > 50000 THEN 'PLATINIUM'
WHEN creditlimit >= 10000 AND creditlimit <= 50000 THEN 'GOLD'
WHEN creditlimit < 10000 THEN 'SILVER'
ELSE 'SILVER'
END;
END;
SELECT usoDeRepeat(10000);
```

# Crear una función que reciba un parámetro TEXT

```
create or replace function subCadena(cadena varchar(20), position integer, position2 integer)
returns text
begin
declare subCadena text default '';
set subCadena = substr(cadena, position, position2);
set subCadena = substr(cadena, position, position2);
set subCadena = substr(cadena, position, position2);
return subCadena;
end;

select subCadena('Ximena Condori Mar',1,7) ,
       subCadena('Ximena Condori Mar',2,6),
       subCadena('Ximena Condori Mar',3,5) ,
       subCadena('Ximena Condori Mar',4,4);
```

```
create or replace function subCadena(cadena varchar(20), position integer, position2 integer)
returns text
begin
declare subCadena text default '';
set subCadena = substr(cadena, position, position2);
set subCadena = substr(cadena, position, position2);
set subCadena = substr(cadena, position, position2);
return subCadena;
end;

select subCadena(cadena: 'Ximena Condori Mar', position: 1, position2: 7) ,
       subCadena(cadena: 'Ximena Condori Mar', position: 2, position2: 6),
       subCadena(cadena: 'Ximena Condori Mar', position: 3, position2: 5) ,
       subCadena(cadena: 'Ximena Condori Mar', position: 4, position2: 4);
```

Output Result 190

1	Ximena	imena	mena
---	--------	-------	------