

PRACTICA BASE DE DATOS II

ESTUDIANTE: HENRY JAVIER HUARACHI QUISPE

DOCENTE: WILLIAM BARRA



MANEJO DE CONCEPTOS

1. DEFINA QUE ES LENGUAJE PROCEDURAL EN MYSQL

Los procedimientos almacenados MySQL son conjuntos de comandos SQL almacenados en el servidor. Esto nos permite trabajar con grandes cantidades de datos.

2. DEFINA QUE ES UNA FUNCIÓN EN MYSQL.

- Es una rutina creada para tomar unos parámetros, procesarlos y retornar en un salida

3. ¿QUÉ COSAS CARACTERÍSTICAS DEBE DE TENER UNA FUNCIÓN? EXPLIQUE SOBRE EL NOMBRE, EL RETURN, PARÁMETROS, ETC.

Una function debe Tener : nombre , operaciones , parametros , returns , return

```
CREATE OR REPLACE FUNCTION fibonacci1()
  RETURNS text          #text nos permite admitir numeros y tetos
  DETERMINISTIC
BEGIN
  DECLARE fib1 INT DEFAULT 0;
  DECLARE fib2 INT DEFAULT 1;
  DECLARE fib3 INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT '0,1,'; # valor por defecto 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,','); #llama str
    END WHILE;
    RETURN str;
  END IF;
END;
```

4. ¿CÓMO CREAR, MODIFICAR Y CÓMO ELIMINAR UNA FUNCIÓN? ADJUNTE UN EJEMPLO DE SU USO.

Una function se crea con : create function (nombre)

Se elimina con : drop function(nombre de la funcion)

Create or replace function (nombre) para modificar la funcion

```
create or replace function conteo_vocales(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 conteo_vocales( str: 'camisa');
drop function conteo_vocales;
```

5. PARA QUÉ SIRVE LA FUNCIÓN CONCAT Y COMO FUNCIONA EN MYSQL

El CONCAT nos sirve para enlazar dos cadenas y que muestre una mejor visualización

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

```
#ejemplos
create or replace function ejempp(x varchar(10),y varchar (10),z varchar(10))
returns varchar (30)
begin
    declare unioh varchar(30) default '';
    set unioh=concat (x,' ',y,' ',z);
    return unioh;

end;

select ejempp('claudio','perez','tapia');
```

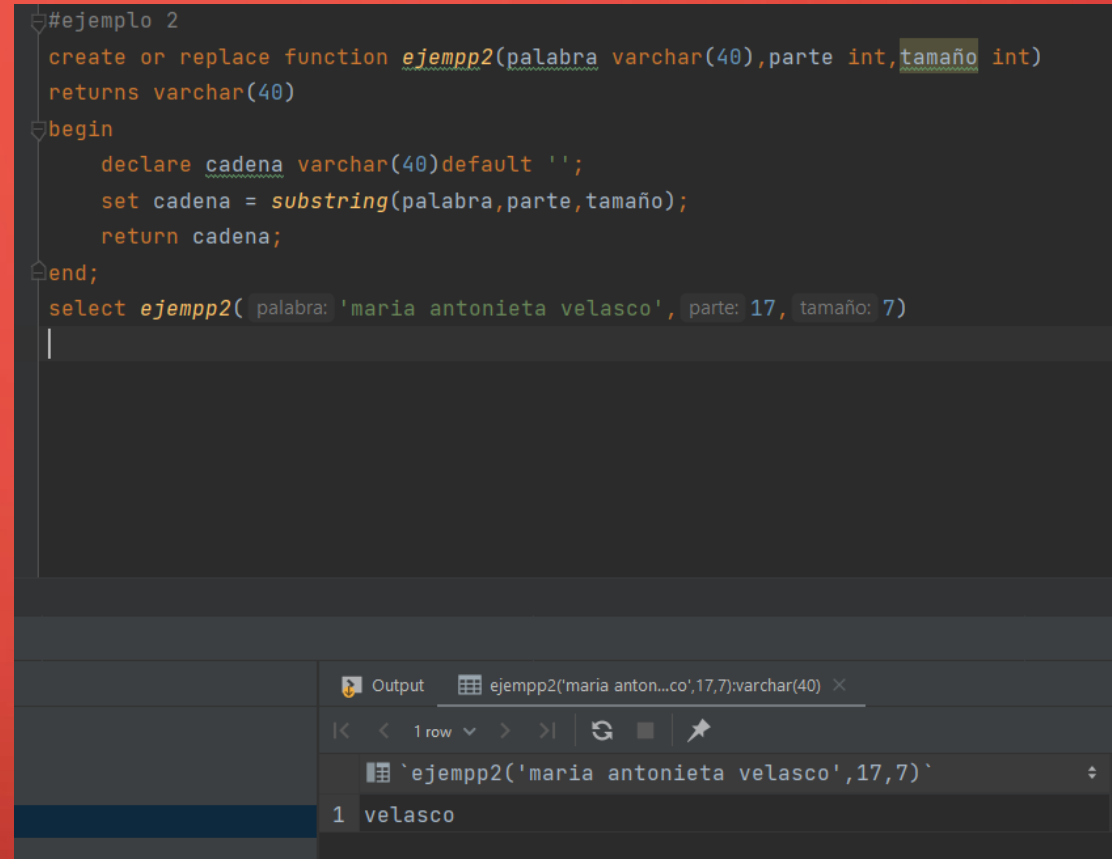
1 row
<code>`ejempp('claudio','perez','tapia')`</code>
1 claudio perez tapia

6. PARA QUÉ SIRVE LA FUNCIÓN SUBSTRING Y COMO FUNCIONA EN MYSQL

Esta función nos sirve para tomar una parte de la cadena.

- ¿Crear una función que muestre el uso de la función SUBSTRING?
- La función recibe un nombre completo.
 - INPUT: Ximena Condori Mar
- La función solo retorna el nombre.
 - OUTPUT: Ximena

```
#ejemplo 2
create or replace function ejempp2(palabra varchar(40),parte int,tamaño int)
returns varchar(40)
begin
    declare cadena varchar(40)default '';
    set cadena = substring(palabra,parte,tamaño);
    return cadena;
end;
select ejempp2( palabra: 'maria antonieta velasco', parte: 17, tamaño: 7)
```



Output	
ejempp2('maria anton...co',17,7);varchar(40)	
1	velasco

7. PARA QUÉ SIRVE LA FUNCIÓN STRCMP Y COMO FUNCIONA EN MYSQL

Nos permite comparar cadenas.

- ¿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.

```
#ejemplo 3
create or replace function ejempp3 (palabra1 varchar(30),palabra2 varchar(30),palabra3 varchar(30))
returns varchar(90)
begin
    declare resultado varchar(60)default '';
    if STRCMP(palabra1,palabra2)=0 THEN SET resultado= 'HAY DOS IGUALES';
    ELSEIF STRCMP(palabra1,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';
    ELSEIF STRCMP(palabra2,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';
    ELSE
        SET resultado='NINGUNO ES IGUAL A OTRO';
    END IF;
    RETURN resultado;
end;

SELECT ejempp3( palabra1: 'CAMISA', palabra2: 'FUTBOL', palabra3: 'CAMISA')
```

Output	
ejempp3('CAMISA','FU...CAMISA'):varchar(90) ×	
1 row	
`ejempp3('CAMISA','FUTBOL','CAMISA')`	
1	HAY DOS IGUALES

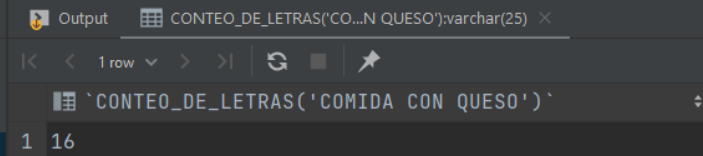
8. PARA QUÉ SIRVE LA FUNCIÓN CHAR_LENGTH Y LOCATE Y COMO FUNCIONA EN MYSQL

La función char_length nos ayuda a contar los espacios y las letras.

La función locate nos ayuda a encontrar cadenas dentro de una cadena.

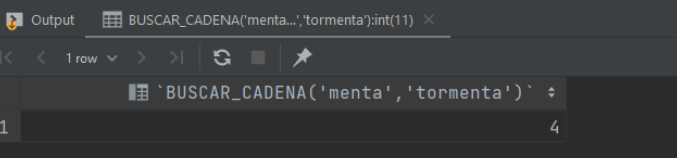
- ¿Crear una función que muestre el uso de ambas funciones?

```
#EJEMP4 CHAR
CREATE OR REPLACE FUNCTION CONTEO_DE_LETRAS(PALABRA VARCHAR (25))
RETURNS VARCHAR(25)
BEGIN
    DECLARE RESULTADO VARCHAR (20) DEFAULT '';
    SET RESULTADO = CHAR_LENGTH(PALABRA);
    RETURN RESULTADO;
end;
SELECT CONTEO_DE_LETRAS( PALABRA: 'COMIDA CON QUESO')
```



	CONTEO_DE_LETRAS('COMIDA CON QUESO')
1	16

```
#EJEMP5 LOCATE
CREATE OR REPLACE FUNCTION BUSCAR_CADENA(SUBCADENA VARCHAR(25),CADENA varchar (20))
RETURNS INTEGER
BEGIN
    DECLARE resultado INTEGER DEFAULT 0;
    SET resultado=locate(SUBCADENA,CADENA);
    return resultado;
end;
select BUSCAR_CADENA( SUBCADENA: 'menta', CADENA: 'tormenta');
```



	BUSCAR_CADENA('menta','tormenta')
1	4

9. ¿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 agregacion son aquellas funciones que vienen por defecto en el MySQL

Las funciones creadas por el DBA

Son funciones que el usuario necesita crear en base a su necesidad usando Tambien funciones de agregacion

MIN
STDEV
SUM
MAX
AVG
COUNT

```
#ejemplo 3
create or replace function ejempp3 (palabra1 varchar(30),palabra2 varchar(30),palabra3 varchar(30))
returns varchar(90)
begin
    declare resultado varchar(60)default '';
    if STRCMP(palabra1,palabra2)=0 THEN SET resultado= 'HAY DOS IGUALES';
    ELSEIF STRCMP(palabra1,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';
    ELSEIF STRCMP(palabra2,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';
    ELSE
        SET resultado='NINGUNO ES IGUAL A OTRO';
    END IF;
    RETURN resultado;
end;

SELECT ejempp3( palabra1: 'CAMISA', palabra2: 'FUTBOL', palabra3: 'CAMISA')
```

Output ejempp3('CAMISA','FU...('CAMISA');varchar(90) x

1 row
ejempp3('CAMISA','FUTBOL','CAMISA')
1 HAY DOS IGUALES

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

Los parámetros de entrada (in) es un parámetro que entra al procedimiento que tendremos y se ara operaciones.

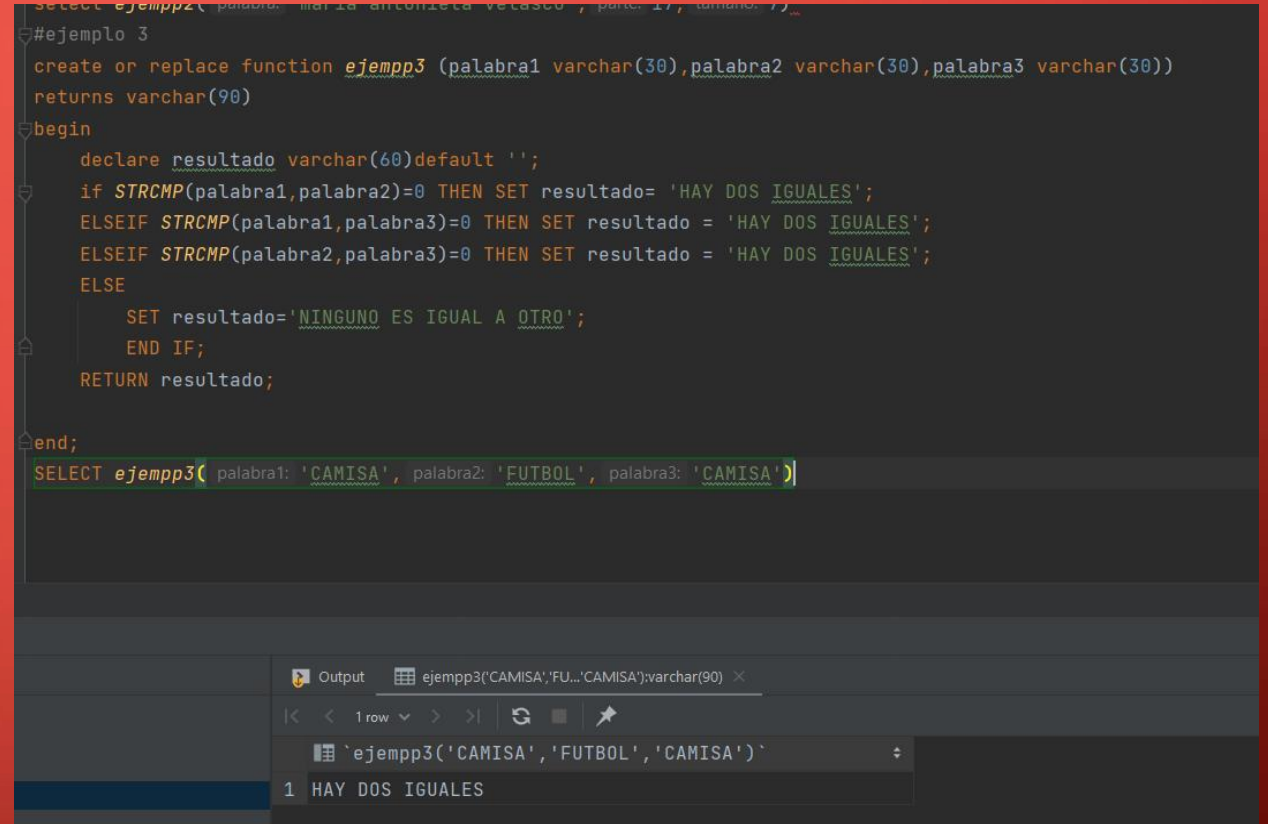
El parámetro de salida (out) es un dato que se le devolverá al usuario al utilizar una función.

El parámetro de entrada y salida (inout)

Es un parámetro que entrara dentro de una operación y dicho resultado se mostrara al usuario.

- Es decir IN INOUT, etc.

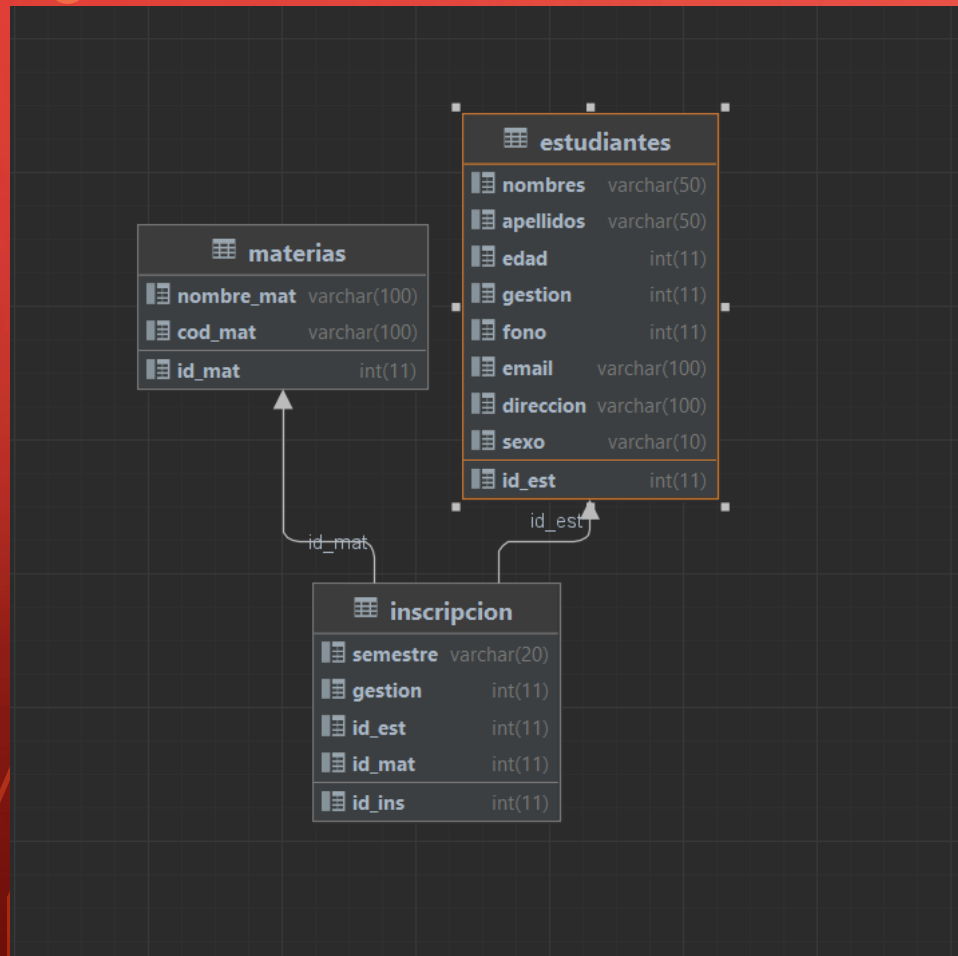
```
select ejempp2( palabra: 'maria antonietta velasco', parametro: 17, tamaño: 7);  
#ejemplo 3  
create or replace function ejempp3 (palabra1 varchar(30),palabra2 varchar(30),palabra3 varchar(30))  
returns varchar(90)  
begin  
    declare resultado varchar(60)default '';  
    if STRCMP(palabra1,palabra2)=0 THEN SET resultado= 'HAY DOS IGUALES';  
    ELSEIF STRCMP(palabra1,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';  
    ELSEIF STRCMP(palabra2,palabra3)=0 THEN SET resultado = 'HAY DOS IGUALES';  
    ELSE  
        SET resultado='NINGUNO ES IGUAL A OTRO';  
    END IF;  
    RETURN resultado;  
end;  
SELECT ejempp3( palabra1: 'CAMISA', palabra2: 'FUTBOL', palabra3: 'CAMISA')
```



ejempp3('CAMISA','FU...CAMISA');varchar(90) ×	
1 HAY DOS IGUALES	

PARTE PRACTICA

11. 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	Adubiri 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_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
8	8	5to Semestre	2021	5	5

	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

12. CREAR UNA FUNCIÓN QUE GENERE LA SERIE FIBONACCI.

```
/*12 SERIE FIBONACCI*/
CREATE OR REPLACE FUNCTION fibonacci(limite INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE fib1 INT DEFAULT 0;
    DECLARE fib2 INT DEFAULT 1;
    DECLARE fib3 INT DEFAULT 0;
    DECLARE str VARCHAR(255) DEFAULT '01';

    IF limite = 1 THEN
        RETURN fib1;
    ELSEIF limite = 2 THEN
        RETURN CONCAT(fib1, fib2);
    ELSE
        WHILE limite > 2 DO
            SET fib3 = fib1 + fib2;
            SET fib1 = fib2;
            SET fib2 = fib3;
            SET limite = limite - 1;
            SET str = CONCAT(str, fib3);
        END WHILE;
        RETURN str;
    END IF;
END;
```

	`fibonacci(7)`
1	112358

13. CREAR UNA VARIABLE GLOBAL A NIVEL BASE DE DATOS.

```
`fibonacci1()`  
1 0,1,1,2,3,5,8,
```

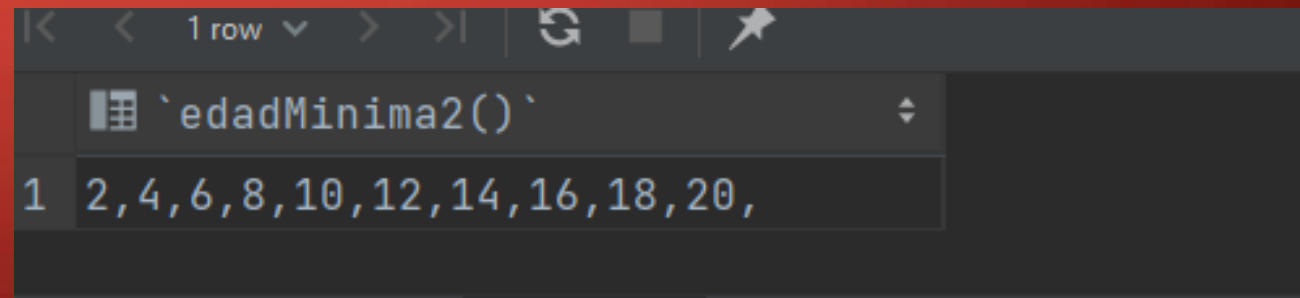
```
#13 CREAR UNA VARIABLE GLOBAL A NIVEL BASE DE DATOS  
set @limit=7; # set @nombre de la variable = asignacion valor(int o varchar)  
  
CREATE OR REPLACE FUNCTION fibonacci1()  
RETURNS text  
DETERMINISTIC  
BEGIN  
    DECLARE fib1 INT DEFAULT 0;  
    DECLARE fib2 INT DEFAULT 1;  
    DECLARE 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;
```

14. CREAR UNA FUNCIÓN NO RECIBE PARÁMETROS (UTILIZAR WHILE, REPEAT O LOOP).

```
#14 CREAR UNA FUNCION QUE NO RECIBE PARAMETROS
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();
```



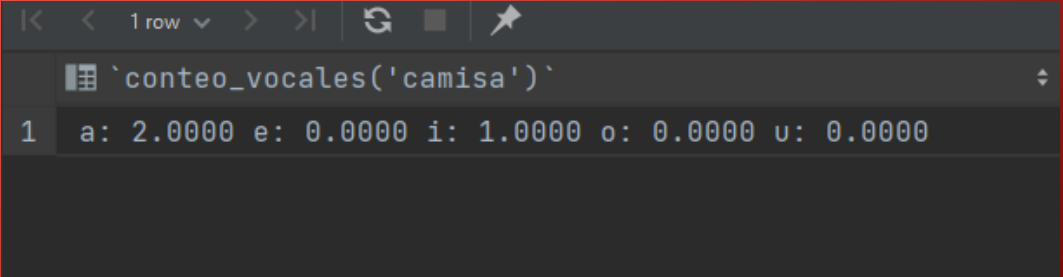
The screenshot shows a database query result window. At the top, it says "1 row". Below that, the query is displayed as ``edadMinima2()``. The result is shown in a table with one row containing the string `2,4,6,8,10,12,14,16,18,20,`.

	`edadMinima2()`
1	2,4,6,8,10,12,14,16,18,20,

15. CREAR UNA FUNCIÓN QUE DETERMINA CUANTAS VECES SE REPITE LAS VOCALES.

```
# 15 crear una funcion cuantas veces se repite las vocales resuelto
create or replace function conteo_vocales(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 conteo_vocales(str: 'camisa');
```



The screenshot shows a database query result with a single row. The query is `'conteo_vocales('camisa')'`. The result is a string showing the count of each vowel: `a: 2.0000 e: 0.0000 i: 1.0000 o: 0.0000 u: 0.0000`.

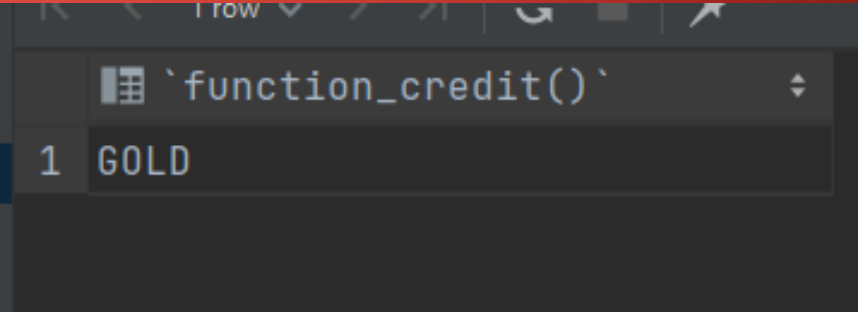
	<code>'conteo_vocales('camisa')'</code>
1	a: 2.0000 e: 0.0000 i: 1.0000 o: 0.0000 u: 0.0000

16. CREAR UNA FUNCIÓN QUE RECIBE UN PARÁMETRO INTEGER.

```
#16 crear una funcion que recibe un parametro integer devuelva txt
set @creditNumber=50000;

create or replace function function_credit()
returns varchar(20)
begin
    declare respuesta varchar(20) default '';
    case
        when @creditNumber>50000 then set respuesta = 'PLATINIUM';
        when @creditNumber<=50000 and @creditNumber >=10000 then set respuesta = 'GOLD';
        when @creditNumber<10000 then set respuesta = 'SILVER';
        else set respuesta='caso desconocido';
    end case;
    return respuesta;
end;

select function_credit();
```



The screenshot shows a database query result window. The title bar indicates the query is 'function_credit()'. The result is displayed in a table with one row and one column, showing the value 'GOLD'.

	function_credit()
1	GOLD

17. CREAR UNA FUNCIÓN QUE RECIBA UN PARÁMETRO TEXT

```
#17 crear una funcion que reciba un parametro txt
CREATE OR REPLACE FUNCTION decreciendo (letra TEXT)
RETURNS TEXT
BEGIN
    DECLARE RESPUESTA TEXT DEFAULT '';
    DECLARE CONTADOR INTEGER DEFAULT CHAR_LENGTH(letra);
    DECLARE AUX INTEGER DEFAULT 1;
    DECLARE AUX2 INTEGER DEFAULT CHAR_LENGTH(letra);

    REPEAT
        SET RESPUESTA= CONCAT(RESPUESTA,' ',SUBSTRING(letra,AUX,AUX2));
        SET CONTADOR=CONTADOR-1;
        SET AUX=AUX+1;

    until CONTADOR <=0 end repeat;
    RETURN RESPUESTA;
end;

SELECT decreciendo( letra: 'francesco');
```

```
`decreciendo('francesco')`
1 ,francesco,ranchesco,anchesco,nchesco,chesco,hesco,esco,sco,co,o
```