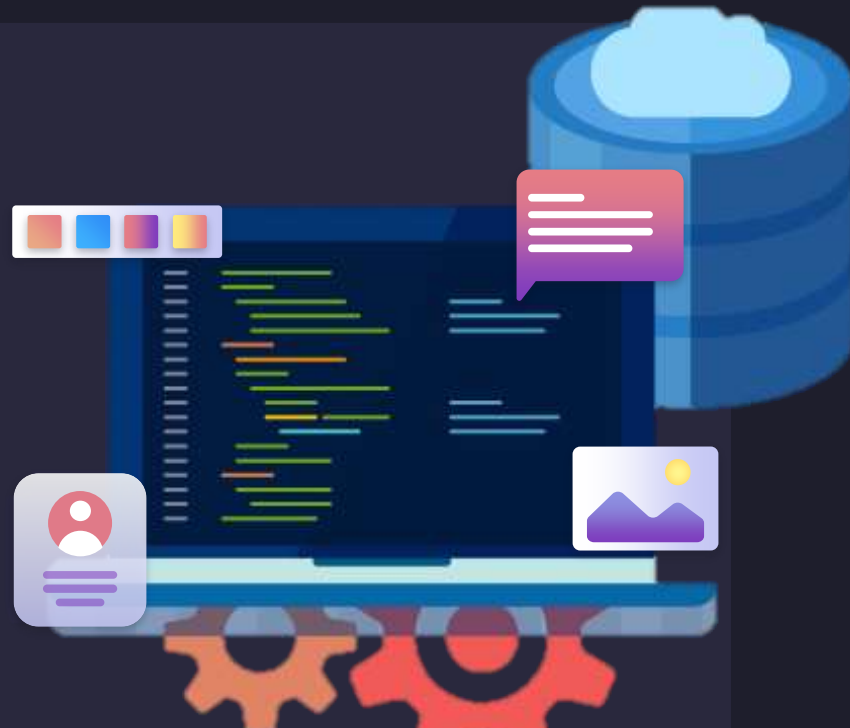




/DEFENSA FINAL HITO 3

BASE DE DATOS II

EDSON IVER CONDORI CONDORI
SIS10929449



EL ALTO, MAYO DE 2023

UNIFRANZ
Internacionalízate

INNOVACIÓN
EN EDUCACIÓN



/CONTENIDO (DBAII).



/01

/PARTE TEORICA

Se explicara
conceptos
elementales de
DBAII.



/02

/PARTE PRACTICA

Se presentara la
aplicacion de la
parte teorica para
la resolucion de los
ejercicios.





/COMENZEMOS

> /BDAII



MARIA DB



/01 /PARTE TEORICA



/CONCEPTOS



1. Defina Que Es Lenguaje Procedural En MySQL.

- ❖ El lenguaje procedural se refiere a la capacidad de la base de datos de permitir la creación de procedimientos almacenados, funciones y desencadenadores utilizando una sintaxis similar a la de los lenguajes de programación convencionales.
- ❖ Estos programas pueden ser utilizados para automatizar y simplificar tareas relacionadas con la manipulación y gestión de los datos almacenados en la base de datos.
- ❖ En resumen, el lenguaje procedural en MySQL permite a los desarrolladores crear programas dentro de la base de datos para mejorar la eficiencia y el rendimiento de las operaciones de la base de datos.





/CONCEPTOS



2. Defina Que Es Una Función En MySQL.

Una función en MySQL es un programa que realiza una tarea específica y devuelve un valor como resultado, que puede ser utilizado en diferentes partes de una consulta SQL. Las funciones son objetos almacenados en la base de datos y se pueden crear utilizando la instrucción "CREATE FUNCTION".

```
CREATE FUNCTION nombre_de_funcion ()  
RETURNS TEXT  
BEGIN  
    #lo que se quier realizar  
  
    RETURN /*lo que se desea retornar*/;  
END;
```



/CONCEPTOS



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

Una función debe de tener:

- Un nombre: A la función que creamos le podemos dar un nombre al cual este relacionado nuestro problema a resolver.
- Parámetros de entrada (opcional): Nuestra Función puede o no recibir Parámetros de entrada.
- Lo que se desea realizar dentro de la función: En aquí se desarrolla la solución del problema.
- Debe de Retornar algo: Nuestra Función nos debe de retornar un valor ya sea numérico o de tipo texto.

/CONCEPTOS



Creación de
la Función

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

```
CREATE OR REPLACE FUNCTION Categoria_cliente(credit_number INT)
  RETURNS TEXT
  BEGIN
    declare response text default '';
    CASE
      WHEN credit_number > 50000
        THEN SET response = 'PLATINIUM';
      WHEN credit_number >= 10000 AND credit_number <= 50000
        THEN SET response = 'GOLD';
      WHEN credit_number < 10000 AND credit_number >= 0
        THEN SET response = 'SILVER';
    END CASE;
    RETURN response;
  END;
```

Nombre de la
Función

Parámetro de
Entrada

Lo que
deseamos
realizar

Lo que
queremos
devolver

/CONCEPTOS



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

- ❖ Para CREAR una Función debemos de Utilizar el comando CREATE FUNTION

Creación de
la Función



```
CREATE FUNCTION nombre_de_funcion ()  
RETURNS TEXT  
BEGIN  
    #lo que se quier realizar  
  
    RETURN /*lo que se desea retornar*/;  
END;
```

/CONCEPTOS



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

- ❖ Para MODIFICAR una Función debemos de Utilizar el comando OR REPLACE FUNTION

Creación de
la Función



```
CREATE OR REPLACE FUNCTION nombre_de_funcion ()  
RETURNS TEXT  
BEGIN  
    #Lo que se quier realizar  
  
    RETURN /*lo que se desea retornar*/;  
END;
```

Modificando
la Función



/CONCEPTOS



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

- ❖ Para **ELIMINAR** una Función debemos de Utilizar el comando **DROP FUNCTION**

Creación de
la Función



```
CREATE OR REPLACE FUNCTION nombre_de_funcion ()  
RETURNS TEXT  
BEGIN  
    #Lo que se quier realizar  
  
    RETURN /*lo que se desea retornar*/;  
END;  
  
DROP FUNCTION nombre_de_funcion;
```

Modificando
la Función



Eliminando la
Función Creada



/CONCEPTOS



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

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

Creación de
Función usando
Concat

```
1# USO DEL CONCAT
2
3CREATE OR REPLACE FUNCTION uso_de_concat(cad1 text,cad2 text,cad3 text)
4RETURNS TEXT
5BEGIN
6    DECLARE response TEXT DEFAULT '';
7    SET response = concat(cad1,' ',cad2,' ',cad3);
8    RETURN response;
9END;
10
11SELECT uso_de_concat( cad1: 'Bienvenidos', cad2: 'a', cad3: 'BDA-II');
```

Realizando la
Consulta a la
Función

Salida de Consola

```
1 uso_de_concat('Bienvenidos','a','BDA-II')
1 Bienvenidos a BDA-II
```

/CONCEPTOS



6. Para qué sirve la función SUBSTRING y como funciona en MYSQL

- ¿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

La Función **SUBSTRING** nos permite poder extraer una subcadena de una cadena.

SUBSTRING(cadena, inicio, longitud)

/CONCEPTOS



6. Para qué sirve la función SUBSTRING y como funciona en MYSQL

```
CREATE OR REPLACE FUNCTION Uso_de_substring(cad text)
RETURNS TEXT
BEGIN
    DECLARE response TEXT DEFAULT ' ';
    DECLARE contador INTEGER DEFAULT locate(' ', cad);
    SET response = SUBSTRING(cad, 1, contador);

    RETURN response;
END;
```

Extrayendo la
Subcadena



Retornando
el valor



Realizando la
Consulta a la
Función



```
SELECT Uso_de_substring( cad; 'Ximena Condori Mar')
```

Salida de Consola



```
'Uso_de_substring('Ximena Condori Mar')'
1 Ximena
```

/CONCEPTOS



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

La función de STRCMP nos sirve para poder comparar dos cadenas de texto y devolver un valor numérico que indica si las cadenas son iguales o son diferentes.

- 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

/CONCEPTOS

Creación de
Función usando
STRCMP

```
CREATE OR REPLACE FUNCTION Uso_de_strcmp(cad TEXT, cad2 TEXT, cad3 TEXT)
RETURNS TEXT
BEGIN
    DECLARE response TEXT DEFAULT ' ';

    CASE
        WHEN STRCMP(cad, cad2) = 0
            THEN SET response = CONCAT('La cadena 1: ', cad, ' y la cadena 2: ', cad2, ' son iguales');
        WHEN STRCMP(cad2, cad3) = 0
            THEN SET response = CONCAT('La cadena 2: ', cad2, ' y la cadena 3: ', cad3, ' son iguales');
        WHEN STRCMP(cad, cad3) = 0
            THEN SET response = CONCAT('Las cadenas 1: ', cad, ' y la cadena 3: ', cad3, ' son iguales');
        ELSE SET response = 'Ninguna de las cadenas son iguales';
    END CASE;
    RETURN response;
END;
```

Realizando la
Consulta a la
Función

```
SELECT Uso_de_strcmp(cad: 'BDAII', cad2: 'BDAII', cad3: 'EDD');
```

Salida de Consola

```
1 'Uso_de_strcmp('BDAII', 'BDAII', 'EDD')'
1 La cadena 1: BDAII y la cadena 2: BDAII son iguales
```


/CONCEPTOS



8. Para qué sirve la función `CHAR_LENGTH` y `LOCATE` y como funciona en `MYSQL`

En MySQL las funciones de `CHAR_LENGTH` nos permite contar el numero de caracteres que existen dentro de una cadena.

Por otra parte la Función de `LOCATE` nos permitirá buscar una palabra deseada que se encuentre en la cadena.

- Crear una función que muestre el uso de ambas funciones

8. Para qué sirve la función CHAR_LENGTH y LOCATE y como funciona en MYSQL

Creación de
Función usando
CHAR_LENGTH y
LOCATE

```
CREATE OR REPLACE FUNCTION uso_charleng_locate(cadena VARCHAR(50), letra CHAR)
RETURNS TEXT
BEGIN
    DECLARE response TEXT DEFAULT 'La letra no se encuentra en la cadena';
    DECLARE contador INT DEFAULT 1;
    DECLARE nVeces INT DEFAULT 0;
    DECLARE puntero CHAR;
    IF LOCATE(letra, cadena) > 0 THEN
        WHILE contador <= char_length(cadena) DO
            SET puntero = substr(cadena, contador, 1);
            IF puntero = letra THEN
                SET nVeces = nVeces + 1;
            END IF;
            SET contador = contador + 1;
        END WHILE;
        SET response = CONCAT('La letra " ', letra, ' " se repite ', nVeces, ' veces');
    END IF;
    RETURN response;
END;
```

Realizando la
Consulta a la
Función

```
SELECT uso_charleng_locate( cadena: 'HOLA MUNDO', letra: 'A');
```

Salida de Consola

```
1 'uso_charleng_locate('HOLA MUNDO', 'A')'
1 La letra " A " se repite 1 veces
```

/CONCEPTOS



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

Las diferencia entre estas son:

Funciones de Agregación: Estas básicamente ya están preestablecidas dentro de la base de datos es decir que las podemos usar en cualquier momento algunas de estas son: MAX, MIN, AVG, CONCAT, etc.

Por otra parte las Funciones que son Creadas por el DBA es decir nosotros, son funciones las cuales nosotros les damos una operación en especifica para resolver un problema, para poder crear una función debemos de usar el comando CREATE FUNCTION.

/CONCEPTOS

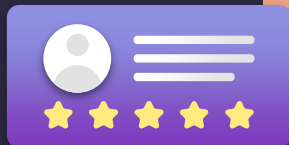
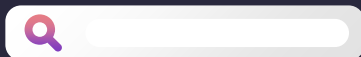


10.¿Busque y defina a qué se referirá cuando se habla de parámetros de entrada y salida en MySQL?

Cuando hablamos de parámetros de entrada y salida nos referimos a los valores que les damos a una función al momento de ejecutarla (parámetros de entrada) y los valores que la función que se devuelve después de su ejecución (parámetros de salida).

IN : Lo usamos para definir un parámetro de entrada en una función.

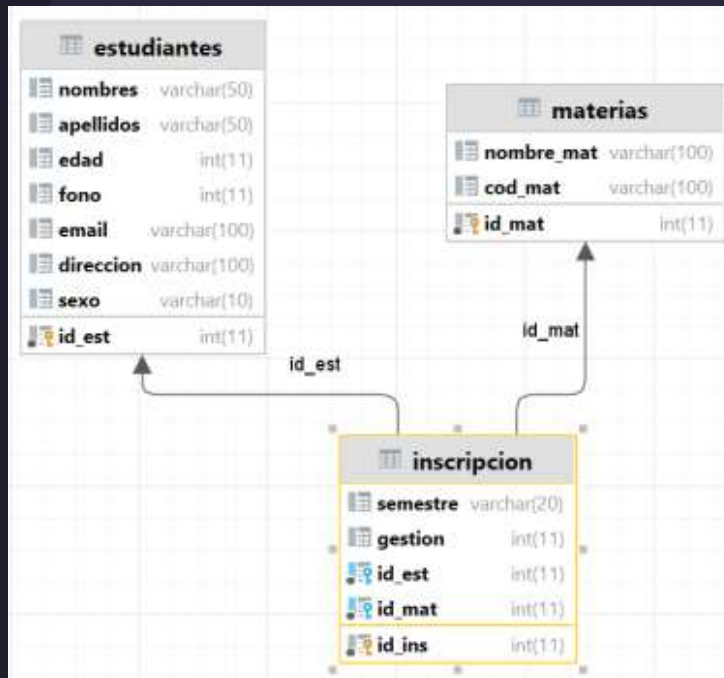
INOUT : Lo usamos para definir un parámetro de entrada y salida en una función.



/02 /PARTE PRACTICA



/ 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 Uria	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

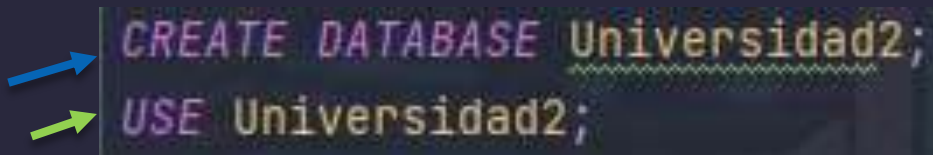
id_mat	nombre_mat	cod_mat
1	Introduccion a la Arquitectura	ARQ-101
2	Urbanismo y Diseno	ARQ-102
3	Dibujo y Pintura Arquitectonico	ARQ-103
4	Matematica discreta	ARQ-104
5	Fisica Basica	ARQ-105

/ 11. Crear la siguiente Base de datos y sus registros.



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

/CREACION DE LA DBA.



```
CREATE DATABASE Universidad2;  
USE Universidad2;
```

El comando CREATE DATABASE es el que genera la DBA.

Comando USE para posicionarnos en la DBA para trabajar en el esta.

/CREACION DE LAS TABLAS.

```
CREATE TABLE estudiantes (  
→ id_est Integer AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  nombres VARCHAR (50) NOT NULL, ←  
  apellidos VARCHAR(50) NOT NULL,  
  edad INTEGER NOT NULL,  
  fono INTEGER NOT NULL,  
  email VARCHAR(100),  
  direccion VARCHAR(100),  
  genero VARCHAR(10)  
);  
CREATE TABLE materias(  
→ id_mat INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  nombre_mat VARCHAR (100) NOT NULL ,  
  cod_mat VARCHAR (100) NOT NULL  
);
```

```
CREATE TABLE inscripcion (  
→ id_inscripcion INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  semestre VARCHAR(20) NOT NULL ,  
  gestion INTEGER NOT NULL,  
  
  id_est INTEGER NOT NULL ,  
  id_mat INTEGER NOT NULL ,  
  
  → FOREIGN KEY (id_est) REFERENCES estudiantes(id_est),  
  → FOREIGN KEY (id_mat) REFERENCES materias(id_mat)  
);  
  
INSERT INTO inscripcion (semestre, gestion, id_est, id_mat)  
VALUES  ('1er Semestre',2018,1,1),  
        ('2do Semestre',2018,1,2),
```

Definimos a los PRIMARY KEY.

El comando FOREIGN KEY es utilizado para relacionar las tablas mediante los PRIMARY KEY.

NOT NULL para no tener columnas sin registro.

/LLENADO DE LAS TABLAS.



```
→ INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, genero)
VALUES ('Miguel', 'Gonzalez Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
       ('Sandra', 'Mavir Uribe', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino'),
       ('Joel', 'Aduviri Mondar', 30, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
       ('Andrea', 'Arias Ballesteros', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino'),
       ('Santos', 'Montes Valenzuela', 24, 2832119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

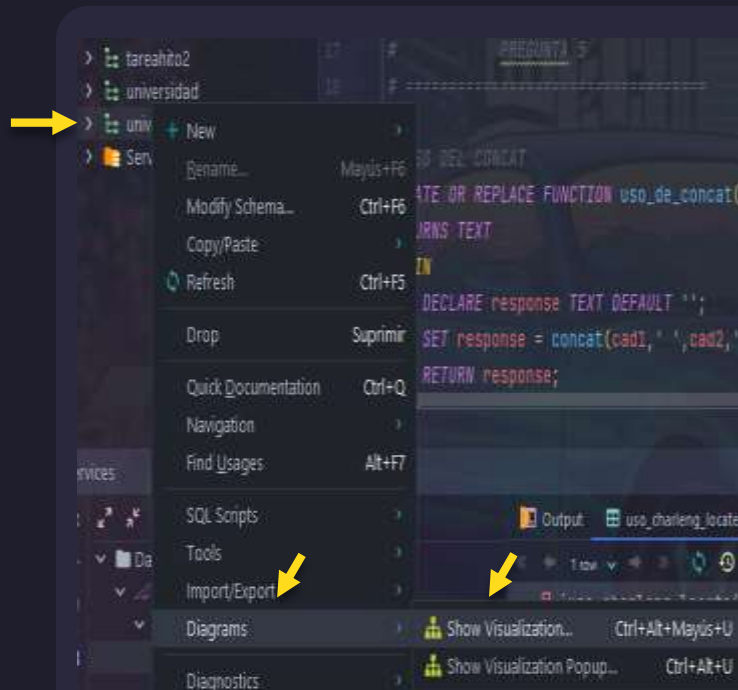
El llenado de la tabla puede acortarse mediante el uso de “,” al final de una cadena de valores, cortando el proceso con “;”

Los INSERT fueron realizados de forma replicada a la especificada en el doc.

```
→ INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Introduccion a la Arquitectura', 'ARQ-101'),
       ('Urbanismo y Diseño', 'ARQ-102'),
       ('Dibujo y Pintura Arquitectonica', 'ARQ-103'),
       ('Matematica Discreta', 'ARQ-104'),
       ('Fisica Basica', 'ARQ-105');
```

```
→ INSERT INTO inscripcion (semestre, gestion, id_est, id_mat)
VALUES ('1er Semestre', 2018, 1, 1),
       ('2do Semestre', 2018, 1, 2),
       ('1er Semestre', 2019, 2, 4),
       ('2do Semestre', 2019, 2, 3),
       ('2do Semestre', 2020, 3, 3),
       ('3er Semestre', 2020, 3, 1),
       ('4to Semestre', 2021, 4, 4),
       ('5to Semestre', 2021, 5, 5);
```

/DIAGRAMA LOGICO DEL PROYECTO.



Haciendo un clic derecho sobre la carpeta de principal, dirijase y seleccione DIAGRAMS y la opción SHOW VISUALIZATION para poder ver el diagrama.



/12.Crear una función que genere la serie Fibonacci.

```
CREATE OR REPLACE FUNCTION fibonacci(NumerLimit INT)
RETURNS TEXT
BEGIN

    DECLARE num1 INT DEFAULT 0;
    DECLARE cont INT DEFAULT 1;
    DECLARE num2 INT DEFAULT 1;
    DECLARE result TEXT DEFAULT '';

    SerieFigonaci: LOOP
        #verifica si se sigue cumpliendo la secuencia
        IF cont > NumerLimit THEN
            LEAVE SerieFigonaci; #LEAVE SALE DEL BUCLE
        END IF;

        SET result = CONCAT(result, num2, ',');
        SET num2 = num1 + num2;
        SET num1 = num2 - num1;
        SET cont = cont + 1;
    END LOOP SerieFigonaci;

    #RETURN SUBSTRING(result, 1/*, LENGTH(result) - 1*/);
    RETURN result;

end;

SELECT fibonacci( NumerLimit: 6);
```

Código que
Resuelve el
Problema

/SUGERENCIAS

- o La función recibe un límite(number)
- o La función debe de retornar una cadena.
- o Ejemplo para n=7. OUTPUT:
0, 1, 1, 2, 3, 5, 8,

/12.Crear una función que genere la serie Fibonacci.



LLAMANDO A LA FUNCION Y
MOSTRANDO EN CONSOLA

Llamando a la
Función

```
SELECT fibonacci( NumerLimit: 6);
```

Output Cadena_Nueva_sin_Vac...A II'

1 row

fibonacci(6)
1, 1, 1, 2, 3, 5, 8,

Mostrando la
Consola de
Salida

/13.Crear una variable global a nivel BASE DE DATOS.

```
SET @Variable_gloval = 'HOLA MI NOMBRE ES EDSON';  
SELECT @Variable_gloval;  
  
CREATE OR REPLACE FUNCTION retorna_el_nombre()  
RETURNS TEXT  
BEGIN  
    RETURN SUBSTR(@Variable_gloval, -5);  
END;
```

Código que
Resuelve el
Problema

```
SELECT retorna_el_nombre();
```

← Llamando a la Función

Output Cadena_Nueva_sin_Vac...A II,'GESTI

1 row

Mostrando la Consola de Salida

```
1 EDSON
```

/SUGERENCIAS

- o Crear una función cualquiera.
- o La función debe retornar la variable global
- Crear una variable global de nombre LIMIT.
- Este valor debe almacenar un valor entero.

Creación var global

```
SET @Limit = 10;  
SELECT @Limit;
```

@Limit :

1	10
---	----

/13.Crear una variable global a nivel BASE DE DATOS.

```
CREATE OR REPLACE FUNCTION figonaci_v2()
RETURNS TEXT
BEGIN
    DECLARE num1 INT DEFAULT 0;
    DECLARE cont INT DEFAULT 1;
    DECLARE num2 INT DEFAULT 1;
    DECLARE result TEXT DEFAULT '';

    SerieFigonaci: LOOP
        #verifica si se siguira cumpliendo la secuencia
        IF cont > @Limit THEN
            LEAVE SerieFigonaci; #LEAVE SALE DEL BUCLE
        END IF;

        SET result = CONCAT(result, num2, ',');
        SET num2 = num1 + num2;
        SET num1 = num2 - num1;
        SET cont = cont + 1;
    END LOOP SerieFigonaci;

    #RETURN SUBSTRING(result, 1,*,LENGTH(result) - 1*/);
    RETURN result;
END;
```

Código que
Resuelve el
Problema

/SUGERENCIAS

- o Crear una función cualquiera.
- o La función debe retornar la variable global
- Crear una variable global de nombre LIMIT.
- Este valor debe almacenar un valor entero.
- Crear una función que genere la serie fibonacci hasta ese valor LIMIT.
- o Note que el valor LIMIT debe ser usado en la función
- o La función no recibe ningún parámetro.

/13.Crear una variable global a nivel BASE DE DATOS.



LLAMANDO A LA FUNCION Y
MOSTRANDO EN CONSOLA

Llamando a la
Función →

```
SELECT figonaci_v2();
```

Output Cadena_Nueva_sin_Vac...A II', 'GE

1 row

figonaci_v2()

1	1,1,2,3,5,8,13,21,34,55,
---	--------------------------

Mostrando la
Consola de
Salida →

/14.Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o LOOP).



/SUGERENCIAS

- 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.

```
`paresImpares()`  
1 0,2,4,6,8,10,12,14,16,18,20,22,24,
```



/14.Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o LOOP).



/SUGERENCIAS

- Si la edad mínima es IMPAR mostrar descendentemente todos los impares hasta el valor 0.

```
`paresImpares()`  
1 25,23,21,19,17,15,13,11,9,7,5,3,1,
```

- Retornar la nueva cadena concatenada.
- Adjuntar el código SQL generado y una imagen de su correcto funcionamiento.
- Nota: Esta función está llamando a otra función, considere eso.



/14.Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o LOOP).



Primera Función
que busca la edad
Mínima

```
# BUSCANDO LA EDAD MINIMA
CREATE OR REPLACE FUNCTION edad_Minima_Est()
RETURNS INT
BEGIN
    DECLARE response INTEGER;
    SELECT MIN(edad) FROM estudiantes INTO response;
    RETURN response;
END;
```

Llamando a la
función y
Mostrando en
Consola

```
SELECT edad_Minima_Est();
```

Output Cadena_Nueva_sin_Vac...A II', 'GE!

<< + 1 row >> ↺ ⌛ 📌

'edad_Minima_Est()' :

1

20

/14.Crear una función no recibe parámetros (Utilizar WHILE, REPEAT o LOOP).



```
CREATE OR REPLACE FUNCTION Edad_Minima_est_bucle()
RETURNS TEXT
BEGIN
    DECLARE response TEXT DEFAULT '';
    DECLARE limite INTEGER;
    # DECLARE limite INTEGER DEFAULT 19; #SI LA EDAD MIN FUERA IMPAR
    DECLARE num INTEGER;

    #Llamando a la función anterior y almacenando en una variable
    # COMENTAR LA LINEA DE ABAJO PARA HACER LA PRUEBA CON UN MIN EDAD IMPAR
    SELECT MIN(edad) FROM estudiantes INTO limite;

    IF limite%2<=0
    THEN
        SET num = 0;
        WHILE num <= limite DO
            SET response = CONCAT(response,num,',');
            SET num = num +2;
        end while;
    ELSE
```

Código que
Resuelve el
Problema

Implementando la
función anterior

```
SET num = 1;
    WHILE num <= limite DO
        SET response = CONCAT(num,',',response);
        SET num = num + 2;
    end while;
END IF;
RETURN response;
END;
```

```
Select Edad_Minima_est_bucle();
```

Output Cadena_Nueva_sin_Vac...A II,"GESTION 2023

<< 1 ROW >> ↺ ↻

```
Edad_Minima_est_bucle()
1 19,17,15,13,11,9,7,5,3,1,
```

Llamando Función y
Mostrando Consola de
Salida

MARIA DB

/15.Crear una función que determina cuantas veces se repite las vocales.



/SUGERENCIAS

- 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.

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

Output SEPARA_VOCALES:text ×

1 row

SEPARA_VOCALES
1 a: 3, e: 4, i: 0, o: 1, u: 0,

/15. Crear una función que determina cuantas veces se repite las vocales.



```
CREATE OR REPLACE FUNCTION cuenta_vocales_cad(Cadena TEXT)
```

```
    RETURNS TEXT
```

```
BEGIN
```

```
    DECLARE puntero CHAR;
```

```
    DECLARE x Int DEFAULT 1; #nos permitira avanzar a la sig letra de la cadena
```

```
    DECLARE cont Int DEFAULT 0;
```

```
    DECLARE cont2 Int DEFAULT 0;
```

```
    DECLARE cont3 Int DEFAULT 0;
```

```
    DECLARE cont4 Int DEFAULT 0;
```

```
    DECLARE cont5 Int DEFAULT 0;
```

```
    WHILE x <= CHAR_LENGTH(Cadena) DO
```

```
        SET puntero = SUBSTR(Cadena, x, 1);
```

```
        IF puntero = 'a' THEN
```

```
            SET cont = cont + 1;
```

```
        end if;
```

```
        IF puntero = 'e' THEN
```

Código que
Resuelve el
Problema



/15.Crear una función que determina cuantas veces se repite las vocales.



```
        SET cont2 = cont2 + 1;
    END IF;

    IF puntero = 'i' THEN
        SET cont3 = cont3 + 1;
    END IF;

    IF puntero = 'o' THEN
        SET cont4 = cont4 + 1;
    END IF;

    IF puntero = 'u' THEN
        SET cont5 = cont5 + 1;
    END IF;

    SET X = X + 1; #nos permitira avanzar a la sig letra/caracter de la cadena
end while;

RETURN CONCAT('a:', cont , ' e:', cont2 , ' i:', cont3 , ' o:', cont4 , ' u:',cont5);
end;
```

Código que
Resuelve el
Problema



/15. Crear una función que determina cuantas veces se repite las vocales.

LLAMANDO A LA FUNCION Y
MOSTRANDO EN CONSOLA

Llamando a la
Función →

```
select cuenta_vocales_cad( Cadena: 'TALLER DE BASE DE DATOS');
```

Output Cadena_Nueva_sin_Vac...A II', 'GESTION 2023') X reducir_cadena('BD/

1 row

```
'cuenta_vocales_cad('TALLER DE BASE DE DATOS')'
```

1	a:3	e:4	i:0	o:1	u:0
---	-----	-----	-----	-----	-----

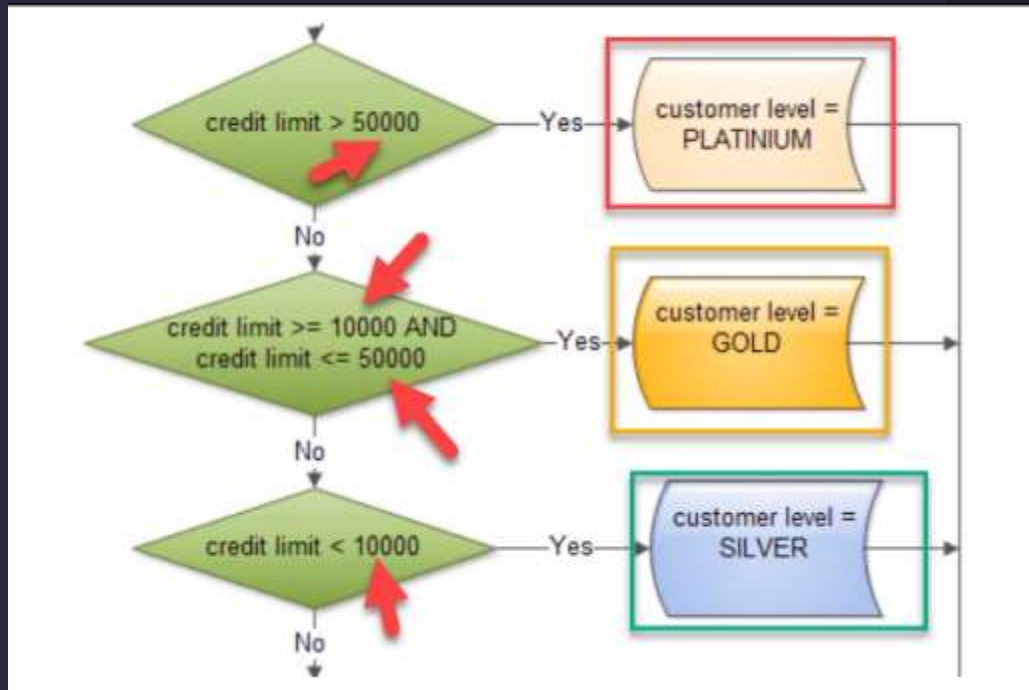
Mostrando la
Consola de
Salida →

/16.Crear una función que recibe un parámetro INTEGER.



/SUGERENCIAS

- La función debe de retornar un texto(TEXT) como respuesta.
- El parámetro es un valor numérico credit_number.
- Si es mayor a 50000 es PLATINIUM.
- 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 PLATINIUM, GOLD o SILVER en base al valor del credit_number.



/16.Crear una función que recibe un parámetro INTEGER. ●●●

```
CREATE OR REPLACE FUNCTION Categoria_cliente(credit_number INT)
RETURNS TEXT
BEGIN
    declare response text default '';

    CASE
    WHEN credit_number > 50000
        THEN SET response = 'PLATINIUM';
    WHEN credit_number >= 10000 AND credit_number <= 50000
        THEN SET response = 'GOLD';
    WHEN credit_number < 10000 AND credit_number >= 0
        THEN SET response = 'SILVER';
    END CASE;
    RETURN response;
END;
```

Código que
Resuelve el
Problema

```
SELECT Categoria_cliente(credit_number: 50001);
```

Output Cadena_Nueva_sin_Vac...A II,'GESTION 2023'] ×

1 row

Categoria_cliente(50001)

1	PLATINIUM
---	-----------

Llamado de Función y
Muestra Consola



/17. Crear una función que recibe 2 parámetros VARCHAR(20), VARCHAR(20).



/SUGERENCIAS

- 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.



/17. Crear una función que recibe 2 parámetros VARCHAR(20), VARCHAR(20).



```
CREATE OR REPLACE FUNCTION nueva_Cadena_sin_vocales(cadena1 VARCHAR(200), cadena2 VARCHAR(200))  
RETURNS TEXT  
BEGIN
```

```
    DECLARE response TEXT DEFAULT '';
```

```
    DECLARE cadena_concatenada TEXT DEFAULT CONCAT(cadena1, '-', cadena2);
```

```
    DECLARE puntero CHAR;
```

```
    DECLARE contador INTEGER DEFAULT 1;
```

```
    #El bucle se ejecuta si el cont sea menor o igual a la longitud de la cadena concatenada
```

```
    WHILE contador <= CHAR_LENGTH(cadena_concatenada) DO
```

```
        SET puntero = SUBSTRING(cadena_concatenada, contador, 1);
```

```
        # FIND_IN_SET() Permite buscar un valor dentro de una cadena/lista separada por comas
```

```
        # además de que nos devuelve la posición
```

```
        # entonces si la posición es mayor a 0 entonces existe una vocal
```

```
        IF FIND_IN_SET(puntero, 'a,e,i,o,u') > 0 THEN
```

```
            SET contador = contador + 1;
```

```
        ELSE # si no encuentra vocal sigue avanzando pero verifica si existe espacios
```

```
            IF puntero = ' ' THEN
```

Código que
Resuelve el
Problema



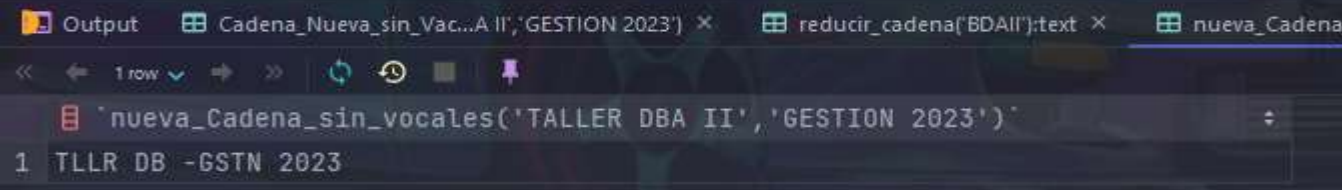
/17. Crear una función que recibe 2 parámetros VARCHAR(20), VARCHAR(20).



```
SET response = CONCAT(response,puntero,' ');  
ELSE # si no encuentra vocal sigue avanzando  
SET contador = contador +1;  
SET response = CONCAT(response,puntero);  
END IF;  
END IF;  
END WHILE;  
  
RETURN response;  
END;
```

Código que
Resuelve el
Problema

```
SELECT nueva_Cadena_sin_vocales( cadena1: 'TALLER DBA II', cadena2: 'GESTION 2023');
```



Llamado de
Función y
Muestra
Consola

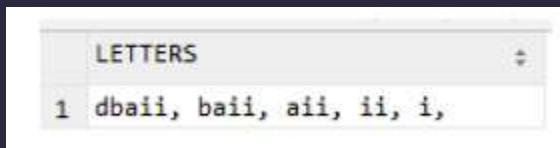


/18.Crear una función que reciba un parámetro TEXT



/SUGERENCIAS

- 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:



/18.Crear una función que reciba un parámetro TEXT



```
CREATE OR REPLACE FUNCTION uso_charleng_locate(cadena VARCHAR(50), letra CHAR)
RETURNS TEXT
BEGIN
    DECLARE response TEXT DEFAULT 'La letra no se encuentra en la cadena';
    DECLARE contador INT DEFAULT 1;
    DECLARE nVeces INT DEFAULT 0;
    DECLARE puntero CHAR;
    IF LOCATE(letra, cadena) > 0 THEN
        WHILE contador <= char_length(cadena) DO
            SET puntero = substr(cadena, contador, 1);
            IF puntero = letra THEN
                SET nVeces = nVeces + 1;
            END IF;
            SET contador = contador + 1;
        END WHILE;
        SET response = CONCAT('La letra " ', letra, ' " se repite ', nVeces, ' veces');
    END IF;
    RETURN response;
END;
```

Código que
Resuelve el
Problema

/18.Crear una función que reciba un parámetro TEXT



LLAMANDO A LA FUNCION Y
MOSTRANDO EN CONSOLA

Llamando a la
Función →

```
SELECT uso_charleng_locate( cadena: 'HOLA MUNDO', letra: 'A');
```

Output uso_charleng_locate(...OLA MUNDO', 'A');text x

1 row

1 'uso_charleng_locate('HOLA MUNDO', 'A')'

1 La letra " A " se repite 1 veces

Mostrando la
Consola de
Salida →



¡GRACIAS POR SU ATENCION!



Email: condoedsoniver@gmail.com
Telefono: +591 72096981



BASE DE DATOS II

