

BASE DE DATOS

MELANIE INGRID VILLCA COPA

LABORATORIO 2 H3

5 de octubre de 2022

```
CREATE DATABASE hito3 lenguaje procesual;
```

```
USE hito3 lenguaje procesual;
```

```
SET @admin = 'ADMIN';
```

```
SELECT @admin;
```

```
SET @admin = 'GUEST';
```

```
#DECLARE response INTEGER DEFAULT 0;
```

```
CREATE FUNCTION usando_variable_global()
```

```
RETURNS VARCHAR(20)
```

```
BEGIN
```

```
    DECLARE respuesta VARCHAR(20) DEFAULT ' ';
```

2

```
IF @admin = 'ADMIN'
```

```
THEN
```

```
SET respuesta = 'Usuario ADMIN';
```

```
ELSE
```

```
SET respuesta = 'Usuario Invitado';
```

```
END IF;
```

```
RETURN respuesta;
```

```
END;
```

```
SELECT usando_variable_global();
```

```
CREATE FUNCTION usando_variable_global v2()
```

```
RETURNS VARCHAR(20)
```

```
BEGIN
```

```
DECLARE respuesta VARCHAR(20) DEFAULT '';
```

```
CASE
```

```
WHEN @admin = 'ADMIN' THEN SET respuesta = 'Usuario ADMIN';
```

```
WHEN @admin = 'GUEST' THEN SET respuesta = 'Usuario Invitado';
```

```
ELSE SET respuesta = 'Usuario nuevo';
```

```
END CASE;
```

```
RETURN respuesta;
```

```
END;
```

```
SET @admin = 'William';
```

```
SELECT usando_variable_global v2();
```

```
#Crear una variable global de nombre PRECIO (numero)
```

```
#Crear una funcion que maneje CASE
```

```
#Si el precio es mayor a 10 y menor igual a 20 retornar el mensaje  
BASICO
```

```
#Si el precio es mayor a 20 y menor igual a 30 retornar el mensaje  
INTERMEDIO
```

```
#Si el precio es mayor a 30 y menor igual a 50 retornar el mensaje  
SUPERIOR
```

```
#Si el precio es mayor a 50 retornar el mensaje CARISIMO
```

```
SET @PRECIO = 35;
```

```
CREATE FUNCTION Precios()
```

```
RETURNS VARCHAR(50)
```

```
BEGIN
```

```
    DECLARE respuesta VARCHAR(50) DEFAULT '';
```

```
    CASE
```

```
        WHEN @PRECIO > 10 AND @PRECIO <=20 THEN SET respuesta =
        'BASICO';
```

```
        WHEN @PRECIO > 20 AND @PRECIO <=30 THEN SET respuesta =
        'INTERMEDIO';
```

```
        WHEN @PRECIO > 30 AND @PRECIO <=50 THEN SET respuesta =
        'SUPERIOR';
```

```
        WHEN @PRECIO > 50 THEN SET respuesta = 'CARISIMO';
```

```
        ELSE SET respuesta = 'ERROR';
```

```
    END CASE;
```

```
    RETURN respuesta;
```

```
END;
```

```
SELECT Precios();
```

```
#Crear una funcion que recibe un parametro INTEGER
```

```
#La funcion debe de retornar un texto TEXT como respuesta
```

```
#El parametro es un valor numerico credit number
```

```
#Si es mayor a 50000 es PLATINIUM
```

```
#Si es mayor igual 10000 y menor igual a 50000 es GOLD
```

```
#Si es menor a 10000 es SILVER
```

```
#La funcion debe retornar indicando si ese cliente es PLATINIUM GOLD  
o SILVER en base al valor del credit number
```

```
SET @credit number = 10000;
```

```
CREATE FUNCTION f_credit_number()
```

```
RETURNS VARCHAR(50)
```

```
BEGIN
```

```
    DECLARE respuesta VARCHAR(50) DEFAULT '';
```

```
    CASE
```

```
        WHEN @credit number > 50000 THEN SET respuesta = 'PLATINIUM';
```

```
        WHEN @credit number >= 10000 AND @credit number <= 50000 THEN  
SET respuesta = 'GOLD';
```

```
        WHEN @credit number < 10000 THEN SET respuesta = 'SILVER';
```

```
        ELSE SET respuesta = 'ERROR';
```

```
    END CASE;
```

```
    RETURN respuesta;
```

```
END;
```

```
SELECT f credit number();
```

```
#
```

```
CREATE FUNCTION numeros_pr(LimitWhile INTEGER)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
    DECLARE x INTEGER DEFAULT 1;
```

```
    DECLARE srt TEXT DEFAULT '';
```

```
    WHILE x <= LimitWhile DO
```

```
        SET srt = CONCAT(srt, x, ',');
```

```
        SET x = x + 1;
```

```
    END WHILE;
```

```
    RETURN srt;
```

```
END;
```

```
SELECT numeros_pr(20);
```

#

```
CREATE FUNCTION numeros_naturales(limite integer)
```

```
RETURNS TEXT
```

```
BEGIN
```

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

```
    DECLARE x INTEGER DEFAULT 1;
```

```
    WHILE x <= limite DO
```

```
        SET response = CONCAT(response, x, ' , ');
```

```
        SET x = x + 1;
```

```
    END WHILE;
```

```
    RETURN response;
```

```
END;
```

```
SELECT numeros_naturales(9);
```

#

```
CREATE FUNCTION numeros_pares(limite integer)
RETURNS TEXT
BEGIN
    DECLARE response TEXT DEFAULT '';
    DECLARE x INTEGER DEFAULT 2;

    WHILE x <= limite DO

        SET response = CONCAT(response, x, ', ');

        SET x = x + 2;

    END WHILE;

    RETURN response;
END;

SELECT numeros_pares(10);
```

#Crear una funcion que maneje WHILE para el siguiente escenario

#Esta funcion recibe un parametro (limite -> integer)

#Si ese numero es par, generar los pares hasta ese numero

#Si ese numero es impar, generar los impares hasta ese numero


```
CREATE FUNCTION numeros pares impares(limite integer)
RETURNS TEXT
BEGIN
    DECLARE respuesta TEXT DEFAULT '';
    DECLARE x INTEGER;

    IF limite % 2 = 0
    THEN
        SET x = 2;
        WHILE x <= limite DO
            SET respuesta = CONCAT(respuesta, x, ' ');
            SET x = x + 2;
        END WHILE;
    ELSE
        SET x = 1;
        WHILE x <= limite DO
            SET respuesta = CONCAT(respuesta, x, ' ');
            SET x = x + 2;
        END WHILE;
    END IF;
```

```
RETURN respuesta;
```

```
END;
```

```
SELECT numeros_pares_impares(12);
```

```
#Crear una funcion que maneje un solo WHILE para el siguiente  
escenario
```

```
#Esta funcion recibe un parametro (limite -> integer)
```

```
#Si ese numero es par, generar los pares hasta ese numero
```

```
#Si ese numero es impar, generar los impares hasta ese numero
```

```
CREATE FUNCTION numeros_pares_impares2(limite integer)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
DECLARE respuesta TEXT DEFAULT '';
```

```
DECLARE x INTEGER;
```

```
IF limite % 2 = 0
```

```
THEN
```

11

```
SET x = 2;
```

```
ELSE
```

```
SET x = 1;
```

```
END IF;
```

```
WHILE x <= limite DO
```

```
SET respuesta = CONCAT(respuesta, x, ' ',');
```

```
SET x = x + 2;
```

```
END WHILE;
```

```
RETURN respuesta;
```

```
END;
```

```
SELECT numeros_pares_impares2(16);
```

```
#Crear una funcion que maneje un solo WHILE para el siguiente  
escenario
```

```
#Esta funcion recibe un parametro (limite -> integer)
```

```
#Si ese numero es par, generar los pares hasta ese numero
```

```
#Si ese numero es impar, generar los impares hasta ese numero
```

```
#DESCENDENTE
```

```
CREATE FUNCTION numeros pares impares3(limite integer)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
    DECLARE respuesta TEXT DEFAULT '';
```

```
    DECLARE x INTEGER;
```

```
    IF limite % 2 = 0
```

```
    THEN
```

```
        SET x = 2;
```

```
    ELSE
```

```
        SET x = 1;
```

```
    END IF;
```

```
    WHILE x <= limite DO
```

```
        SET respuesta = CONCAT(x, ', ', respuesta);
```

```
        SET x = x + 2;
```

```
    END WHILE;
```

```
RETURN respuesta;
```

```
END;
```

```
SELECT numeros pares impares3(13);
```

```
#
```

```
CREATE FUNCTION numeros pares impares4(limite integer)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
    DECLARE respuesta TEXT DEFAULT '';
```

```
    WHILE limite > 0 DO
```

```
        SET respuesta = CONCAT(respuesta, limite, ', ');
```

```
        SET limite = limite - 2;
```

```
    END WHILE;
```

```
    RETURN respuesta;
```

```
END;
```

```
SELECT numeros_pares_impares4(11);
```

```
#WHILE -> SE EJECUTA MIENTRAS LA CONDICION SEA VERDAD
```

```
#REPEAT -> SE EJECUTA MIENTRAS LA CONDICION SEA FALSA
```

```
#REPEAT
```

```
CREATE FUNCTION usoDeRepeat(x int)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
DECLARE str TEXT DEFAULT '';
```

```
REPEAT
```

```
SET str = CONCAT(str, x, ',');
```

```
SET x = x - 1;
```

```
UNTIL x <= 0 END REPEAT;
```

```
RETURN str;
```

```
END;
```

```
SELECT usoDeRepeat(10);
```

```
#
```

```
CREATE FUNCTION usoDeRepeat2(limiteIteracion INTEGER)
```

```
RETURNS TEXT
```

```
BEGIN
```

```
    DECLARE str TEXT DEFAULT '';
```

```
    DECLARE x INTEGER DEFAULT limiteIteracion;
```

```
    REPEAT
```

```
        IF (x % 2 = 0)
```

```
        THEN
```

```
            SET str = CONCAT(str, x, ' -AA- ');
```

```
        ELSE
```

```
            SET str = CONCAT(str, x, ' -BB- ');
```

```
        END IF;
```

```
        SET x = x - 1;
```

```
UNTIL x <= 0 END REPEAT;
```

```
RETURN str;
```

```
END;
```

```
SELECT usoDeRepeat2(10);
```