**Ejercicios Procedimientos y funciones:**

Sobre la base de datos test

1. Realiza un procedimiento que muestre el año actual.
2. Cree un procedimiento que muestre las tres primeras letras de una cadena pasada como parámetro, pasadas a mayúsculas.

CREATE PROCEDURE tres\_letras(cadena VARCHAR(20), OUT letras VARCHAR(3)) BEGIN

SELECT substr(cadena, 1,3) INTO letras;

END  
//  
CALL tres\_letras(“hola caracola”,

1. Crea un procedimiento que muestre dos cadenas pasadas como parámetros concatenadas y en mayúscula.

CREATE PROCEDURE mayus\_concat(cadena1 VARCHAR(20),cadena2 VARCHAR(20))

BEGIN

SELECT CONCAT(UPPER(cadena1),UPPER(cadena2)) AS resultado;

END//

1. Cree una función que devuelva el valor de la hipotenusa de un triángulo a partir de los valores de sus lados.

CREATE FUNCTION hipotenusa (lado1 DECIMAL, lado2 DECIMAL) RETURNS DECIMAL

BEGIN

RETURN SQRT(lado1\*lado1+lado2\*lado2);

END

//

SELECT hipotenusa(3,4)

CREATE FUNCTION hipotenusa(cateto1 DECIMAL(3,2),cateto2 DECIMAL(3,2)) RETURNS DECIMAL(3,2)

BEGIN

DECLARE h DECIMAL(3,2);

SELECT SQRT(cateto1\*cateto1+cateto2\*cateto2) INTO h;

RETURN h;

END//

1. Realiza una función que devuelva true si el número pasado como parámetro es par y false si es impar.

CREATE FUNCTION saber\_parimpar(num INTEGER) RETURNS BOOLEAN

BEGIN

IF (num MOD 2)=0 THEN RETURN TRUE;

ELSE RETURN FALSE;

END IF;

END

//

1. Crea un procedimiento que muestre la suma de los primeros n número enteros (n entero y parámetro de entrada).

CREATE PROCEDURE n\_enteros(num INTEGER, OUT suma INTEGER)

BEGIN

DECLARE bucle INTEGER DEFAULT 0;

SET suma=0;

WHILE bucle<=num DO

SET suma=suma+num;

SET bucle=bucle+1;

END WHILE;

END

//

CALL n\_enteros(4, @suma)//  
SELECT @suma//

1. Haga un procedimiento que muestre la suma de los términos 1/n con n entre 1 y m, es decir, 1/1+1/2+1/3+...+1/m, siendo m el parámetro de entrada. Tenga en cuenta que me no puede ser 0.

CREATE PROCEDURE suma\_fracciones (n INT, OUT suma DECIMAL)

BEGIN

DECLARE m INT DEFAULT 1;

SET suma=0;

WHILE (m<=n) DO

SET suma=suma+(1/m);

SET m=m+1;

END WHILE;

END

//

[CALL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/call.html) suma\_fracciones(0, @total)//

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) @total//

1. Cree una función que determine si un número es primo, devolviendo 0(falso) o 1 (verdadero).

CREATE FUNCTION num\_primo(num INTEGER) RETURNS INTEGER

BEGIN

DECLARE i INT DEFAULT 2;

DECLARE resul INT DEFAULT 1;

WHILE (i<(num/2)) DO

IF ((num MOD i) = 0) THEN SET resul=0;

END IF;

SET i=i+1;

END WHILE;

RETURN resul;

END

//

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) num\_primo(11)//

1. Usando la función anterior cree otra que calcule la suma de los primeros m números primos empezando en el 1.

CREATE FUNCTION suma\_primos (num INTEGER) RETURNS INTEGER

BEGIN

DECLARE a INT DEFAULT 1;

DECLARE suma INT DEFAULT 0;

DECLARE n INT DEFAULT 1;

WHILE a<=n DO

IF num\_primo(n) THEN

SET suma = suma + n;

SET a = a + 1;

END IF;

SET n = n+1;

END WHILE;

RETURN suma;

END

//

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) suma\_primos(3)//

1. Cree un procedimiento para generar y almacenar en la tabla primos (primos(id, numero)) de la base test los primeros números primos comprendidos entre 1 y m (parámetro de entrada).

CREATE PROCEDURE tabla\_primos(num INTEGER)

BEGIN

CREATE TABLE primos("ID" INT PRIMARY KEY, "Primo" INT);

IF(num\_primo(num)=1

INSERT INTO primos("ID", "Primo") VALUES (\*\*\*\*\*\*\*\*\*\*\*\*

1. Modifique el procedimiento anterior para almacenar en la variable de salida @np el número de primos almacenado.
2. Realiza un procedimiento que reciba un parámetro entero. Este procemiento llama a la función anterior y muestra por pantalla “*numero*  es par” o bien “ *numero* es impar”.