**EJERCICIOS INTRODUCCIÓN FUNCIONES Y PROCEDIMIENTOS 2020**

**OPCIONAL**

**Para generar un número aleatorio en el rango a <= n <= b, puede usar la siguiente fórmula**

**a + RAND() \* (b - a + 1))**

**En nuestro caso, como queríamos generar números aleatorios >=5 y <10, tendríamos que poner:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `aleat5y10`()

BEGIN

SELECT 5+RAND()\*5;

END

1. **Modifica la función anterior para que si el número es impar devuelva la cadena “impar” y si es par devuelva la cadena “par”.**

CREATE FUNCTION `poi2`(num int)

RETURNS varchar(10)

BEGIN

declare valor varchar(10);

IF MOD(num,2)=0 THEN SET valor="PAR";

else SET valor="IMPAR";

END IF;

RETURN valor;

END

1. **Sobre el esquema de pruebas test crea un procedimiento para mostrar el año actual.**

CREATE PROCEDURE `año`()

BEGIN

select curdate() as año\_actual;

END

1. **Sobre el esquema test, crea una función a la que pasaremos como parámetro un dato de tipo fecha y la función me devolverá el número de días que han pasado hasta hoy.**

CREATE FUNCTION `dias`(a date) RETURNS int(11)

BEGIN

declare d int;

set d=datediff(curdate(),a);

RETURN d;

END

1. **Crea una función que muestre las tres primeras letras de una cadena pasada como parámetro en mayúsculas. Usa las funciones UPPER para pasar a mayúsculas (LOWER pasa a minúsculas) y LEFT(cadena,n) para quedarse con los n primeros caracteres de la izquierda.**

**Ejemplo: Si hago left(“perro”,3) me devuelve “per”. Si a esto le aplicas UPPER 🡪 UPPER(LEFT(“perro”,3)) me devuelve PER.**

CREATE DEFINER=`root`@`localhost` FUNCTION `tresletras`(a varchar(15)) RETURNS varchar(3) CHARSET utf8mb4

BEGIN

declare x varchar(3);

set x=upper(left(a,3));

RETURN x;

END

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

CREATE PROCEDURE doscadenas (cadena1 varchar(10), cadena2 varchar(10))

BEGIN

declare nuevacad varchar(30);

set nuevacad=concat(upper(cadena1), upper(cadena2));

select nuevacad as concatenado;

END

CREATE DEFINER=`root`@`localhost` PROCEDURE `doscadenas`(cadena1 varchar(10), cadena2 varchar(10))

BEGIN

SELECT concat(upper(cadena1), upper(cadena2)) AS CONCATENADO;

END

1. **Crea una función que devuelva el valor de la hipotenusa de un triángulo a partir de los valores de sus lados. Busca las funciones que hacen referencia a la raíz cuadrada y a la elevación al cuadrado en el manual de MySQL**

CREATE DEFINER=`root`@`localhost` FUNCTION `hipotenusa`(lado1 decimal(4,2), lado2 decimal(4,2)) RETURNS decimal(8,2)

BEGIN

DECLARE hip decimal(8,2);

set hip=sqrt(pow(lado1,2)+pow(lado2,2));

return hip;

END