

EJERCICIOS FUNCIONES

1. Escribe una función que reciba un número entero de entrada y devuelva TRUE si el número es par o FALSE en caso contrario.

DELIMITER //

CREATE FUNCTION es_par(numero INT) RETURNS BOOLEAN

DETERMINISTIC

BEGIN

 RETURN (MOD(numero, 2) = 0);

END //

DELIMITER ;

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

$h = \text{raíz cuadrada } (a^2 + b^2)$

SQRT(POW(cateto1, 2) + POW(cateto2, 2))

DELIMITER //

CREATE FUNCTION hipotenusa(cateto1 DOUBLE, cateto2 DOUBLE)
RETURNS DOUBLE

DETERMINISTIC

BEGIN

 RETURN SQRT(POW(cateto1, 2) + POW(cateto2, 2));

END //

DELIMITER ;

3. Escribe una función que reciba como parámetro de entrada un valor numérico que represente un día de la semana y que devuelva una cadena de caracteres con el nombre del día de la semana correspondiente. Por ejemplo, para el valor de entrada 1 debería devolver la cadena lunes.

DELIMITER //

CREATE FUNCTION nombre_dia(numero_dia INT) RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

 RETURN CASE numero_dia

 WHEN 1 THEN 'Lunes'

 WHEN 2 THEN 'Martes'

 WHEN 3 THEN 'Miércoles'

 WHEN 4 THEN 'Jueves'

 WHEN 5 THEN 'Viernes'

 WHEN 6 THEN 'Sábado'

 WHEN 7 THEN 'Domingo'

 ELSE 'Día inválido'

 END;

END //

DELIMITER ;

4. Escribe una función que reciba tres números reales como parámetros de entrada y devuelva el mayor de los tres.

DELIMITER //

CREATE FUNCTION maximo_tres(a DOUBLE, b DOUBLE, c DOUBLE)

 RETURNS DOUBLE

DETERMINISTIC

BEGIN

```
    RETURN GREATEST(a, b, c);  
END //
```

```
DELIMITER ;
```

5. Escribe una función que devuelva el valor del área de un círculo a partir del valor del radio que se recibirá como parámetro de entrada.

```
DELIMITER //
```

```
CREATE FUNCTION area_circulo(radio DOUBLE) RETURNS DOUBLE  
DETERMINISTIC  
BEGIN  
    RETURN PI() * POW(radio, 2);  
END //
```

```
DELIMITER ;
```

6. Escribe una función que devuelva como salida el número de años que han transcurrido entre dos fechas que se reciben como parámetros de entrada. Por ejemplo, si pasamos como parámetros de entrada las fechas 2018-01-01 y 2008-01-01 la función tiene que devolver que han pasado 10 años. Para realizar esta función puede hacer uso de las siguientes funciones que nos proporciona MySQL:

DATEDIFF y TRUNCATE

```
DELIMITER //
```

```
CREATE FUNCTION anios_entre_fechas(fecha1 DATE, fecha2 DATE)  
RETURNS INT  
DETERMINISTIC  
BEGIN  
    RETURN TRUNCATE(DATEDIFF(fecha1, fecha2) / 365, 0);  
END //
```

```
DELIMITER ;
```

7. Escribe una función que reciba una cadena de entrada y devuelva la misma cadena pero sin acentos. La función tendrá que reemplazar todas las vocales que tengan acento por la misma vocal pero sin acento. Por ejemplo, si la función recibe como parámetro de entrada la cadena María la función debe devolver la cadena Maria.

```
REPLACE( cadena, caracter_sustituir, caracter_nuevo)
```

```
DELIMITER //
```

```
CREATE FUNCTION quitar_acentos(texto VARCHAR(255)) RETURNS  
VARCHAR(255)  
DETERMINISTIC  
BEGIN
```

```

SET texto = REPLACE(texto, 'á', 'a');
SET texto = REPLACE(texto, 'é', 'e');
SET texto = REPLACE(texto, 'í', 'i');
SET texto = REPLACE(texto, 'ó', 'o');
SET texto = REPLACE(texto, 'ú', 'u');
SET texto = REPLACE(texto, 'Á', 'A');
SET texto = REPLACE(texto, 'É', 'E');
SET texto = REPLACE(texto, 'Í', 'I');
SET texto = REPLACE(texto, 'Ó', 'O');
SET texto = REPLACE(texto, 'Ú', 'U');
RETURN texto;
END //

DELIMITER ;

```

8. FUNCIONES SOBRE LA BASE DE DATOS TIENDA.

Realiza las siguientes funciones:

- a. Escribe una función para la base de datos tienda que devuelva el número total de productos que hay en la tabla productos.

DELIMITER //

```

CREATE FUNCTION total_productos() RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total FROM producto;
    RETURN total;
END //

```

DELIMITER ;

- b. Escribe una función para la base de datos tienda que devuelva el valor medio del precio de los productos de un determinado fabricante que se recibirá como parámetro de entrada. El parámetro de entrada será el nombre del fabricante.

DELIMITER //

```

CREATE FUNCTION precio_medio_por_fabricante(nombre_fabricante
    VARCHAR(100)) RETURNS DOUBLE
DETERMINISTIC
BEGIN
    DECLARE media DOUBLE;
    SELECT AVG(p.precio)
    INTO media
    FROM producto p

```

```
JOIN fabricante f ON p.id_fabricante = f.id
WHERE f.nombre = nombre_fabricante;
RETURN media;
END //
```

DELIMITER ;

- c. Escribe una función para la base de datos tienda que devuelva el valor máximo del precio de los productos de un determinado fabricante que se recibirá como parámetro de entrada. El parámetro de entrada será el nombre del fabricante.

DELIMITER //

```
CREATE FUNCTION precio_maximo_por_fabricante(nombre_fabricante
    VARCHAR(100)) RETURNS DOUBLE
DETERMINISTIC
BEGIN
    DECLARE maximo DOUBLE;
    SELECT MAX(p.precio)
    INTO maximo
    FROM producto p
    JOIN fabricante f ON p.id_fabricante = f.id
    WHERE f.nombre = nombre_fabricante;
    RETURN maximo;
END //
```

DELIMITER ;

- d. Escribe una función para la base de datos tienda que devuelva el nombre del fabricante con más productos.

DELIMITER //

```
CREATE FUNCTION fabricante_con_mas_productos() RETURNS
    VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE nombre_fabricante VARCHAR(100);
    SELECT f.nombre
    INTO nombre_fabricante
    FROM fabricante f
    JOIN producto p ON f.id = p.id_fabricante
    GROUP BY f.id
    ORDER BY COUNT(p.id) DESC
    LIMIT 1;
    RETURN nombre_fabricante;
```

```
END //
```

```
DELIMITER ;
```

- e. Escribe una función para la base de datos tienda que devuelva el nombre del producto más barato.

```
DELIMITER //
```

```
CREATE FUNCTION producto_mas_barato() RETURNS VARCHAR(100)  
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE nombre_producto VARCHAR(100);
```

```
    SELECT nombre
```

```
    INTO nombre_producto
```

```
    FROM producto
```

```
    ORDER BY precio ASC
```

```
    LIMIT 1;
```

```
    RETURN nombre_producto;
```

```
END //
```

```
DELIMITER ;
```