

Situación Base

Una empresa desea implementar funciones en su base de datos para mejorar la eficiencia de las consultas. Necesitan funciones que realicen diversos cálculos y transformaciones de datos.

```
CREATE DATABASE empresa;
```

```
USE empresa;
```

```
-- Tabla de empleados
```

```
CREATE TABLE empleados (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    salario_mensual DECIMAL(10,2)  
);
```

```
INSERT INTO empleados (nombre, salario_mensual) VALUES  
( 'Carlos Pérez', 1500),  
( 'Ana Gómez', 1800),  
( 'Luis Fernández', 2000);
```

```
-- Tabla de productos
```

```
CREATE TABLE productos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    precio DECIMAL(10,2),  
    iva DECIMAL(5,2),  
    descuento DECIMAL(5,2)
```

```
);
```

```
INSERT INTO productos (nombre, precio, iva, descuento) VALUES
```

```
('Laptop', 1000, 21, 10),
```

```
('Mouse', 50, 21, 5),
```

```
('Teclado', 80, 21, 8);
```

```
-- Tabla de ventas
```

```
CREATE TABLE ventas (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    producto_id INT,
```

```
    cantidad INT,
```

```
    FOREIGN KEY (producto_id) REFERENCES productos(id)
```

```
);
```

```
INSERT INTO ventas (producto_id, cantidad) VALUES
```

```
(1, 2),
```

```
(2, 5),
```

```
(3, 3);
```

```
-- Tabla de rectángulos
```

```
CREATE TABLE rectangulos (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    base INT,
```

```
    altura INT
```

```
);
```

```
INSERT INTO rectangulos (base, altura) VALUES
```

```
(10, 5),
```

```
(8, 4),
```

```
(15, 7);
```

```
-- Tabla de usuarios
```

```
CREATE TABLE usuarios (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    nombre VARCHAR(100)
```

```
);
```

```
INSERT INTO usuarios (nombre) VALUES
```

```
('Carlos'),
```

```
('Ana'),
```

```
('Luis');
```

Problemas

1. Función para calcular el área de un rectángulo

Crea una función llamada `calcular_area` que reciba la base y la altura de un rectángulo y retorne su área.

2. Función para determinar si un número es par

Crea una función llamada `es_par` que reciba un número entero y retorne `1` si es par y `0` si es impar.

3. Función para obtener el salario anual

Crea una función llamada `salario_anual` que reciba el salario mensual de un empleado y retorne su salario anual considerando 12 meses.

4. Función para convertir temperaturas de Celsius a Fahrenheit

Crea una función llamada `celsius_a_fahrenheit` que reciba una temperatura en grados Celsius y la convierta a Fahrenheit con la fórmula:

$$F = (C \times 9/5) + 32$$

5. Función para obtener la inicial de un nombre

Crea una función llamada `inicial_nombre` que reciba un nombre completo y retorne la primera letra del nombre en mayúscula.

6. Función para calcular el IVA

Crea una función llamada `calcular_iva` que reciba un monto y un porcentaje de IVA y retorne el monto total con IVA incluido.

7. Función para calcular el descuento en una compra

Crea una función llamada `calcular_descuento` que reciba el precio de un producto y el porcentaje de descuento y retorne el precio final después del descuento.

8. Función para contar caracteres en un texto

Crea una función llamada `contar_caracteres` que reciba una cadena de texto y retorne la cantidad de caracteres que tiene.

9. Función para calcular el promedio de tres números

Crea una función llamada `promedio_tres_numeros` que reciba tres números y retorne el promedio.

10. Función para determinar el mayor de tres números

Crea una función llamada `mayor_de_tres` que reciba tres números y retorne el mayor de ellos.

Respuestas (Implementación en SQL)

1. Función para calcular el área de un rectángulo

```
CREATE FUNCTION calcular_area(base INT, altura INT)
RETURNS INT DETERMINISTIC
BEGIN
    RETURN base * altura;
END;
```

 **Ejemplo de uso:**

```
SELECT calcular_area(10, 5);
```

Salida: 50

2. Función para determinar si un número es par

```
CREATE FUNCTION es_par(numero INT)
RETURNS INT DETERMINISTIC
BEGIN
    RETURN numero % 2 = 0;
END;
```

 **Ejemplo de uso:**

```
SELECT es_par(8);
```

Salida: 1

3. Función para obtener el salario anual

```
CREATE FUNCTION salario_anual(salario_mensual DECIMAL(10,2))
RETURNS DECIMAL(10,2) DETERMINISTIC
BEGIN
    RETURN salario_mensual * 12;
END;
```

Ejemplo de uso:

```
SELECT salario_anual(1500);
```

Salida: 18000.00

4. Función para convertir temperaturas de Celsius a Fahrenheit

```
CREATE FUNCTION celsius_a_fahrenheit(celsius FLOAT)
RETURNS FLOAT DETERMINISTIC
BEGIN
    RETURN (celsius * 9/5) + 32;
END;
```

Ejemplo de uso:

```
SELECT celsius_a_fahrenheit(25);
```

Salida: 77.0

5. Función para obtener la inicial de un nombre

```
CREATE FUNCTION inicial_nombre(nombre VARCHAR(100))
RETURNS CHAR(1) DETERMINISTIC
BEGIN
    RETURN UPPER(LEFT(nombre, 1));
END;
```

Ejemplo de uso:

```
SELECT inicial_nombre('Carlos');
```

Salida: 'C'

6. Función para calcular el IVA

```
CREATE FUNCTION calcular_iva(monto DECIMAL(10,2), iva DECIMAL(5,2))
RETURNS DECIMAL(10,2) DETERMINISTIC
BEGIN
  RETURN monto + (monto * iva / 100);
END;
```

 Ejemplo de uso:

```
SELECT calcular_iva(1000, 21);
```

Salida: 1210.00

7. Función para calcular el descuento en una compra

```
CREATE FUNCTION calcular_descuento(precio DECIMAL(10,2), descuento DECIMAL(5,2))
RETURNS DECIMAL(10,2) DETERMINISTIC
BEGIN
  RETURN precio - (precio * descuento / 100);
END;
```

 Ejemplo de uso:

```
SELECT calcular_descuento(200, 10);
```

Salida: 180.00

8. Función para contar caracteres en un texto

```
CREATE FUNCTION contar_caracteres(texto VARCHAR(255))
RETURNS INT DETERMINISTIC
BEGIN
  RETURN CHAR_LENGTH(texto);
END;
```

 Ejemplo de uso:

```
SELECT contar_caracteres('Hola Mundo');
```

Salida: 10

9. Función para calcular el promedio de tres números

```
CREATE FUNCTION promedio_tres_numeros(num1 FLOAT, num2 FLOAT, num3 FLOAT)
RETURNS FLOAT DETERMINISTIC
BEGIN
    RETURN (num1 + num2 + num3) / 3;
END;
```

 **Ejemplo de uso:**

```
SELECT promedio_tres_numeros(8, 9, 10);
```

Salida: 9.0

10. Función para determinar el mayor de tres números

```
CREATE FUNCTION mayor_de_tres(n1 INT, n2 INT, n3 INT)
RETURNS INT DETERMINISTIC
BEGIN
    RETURN GREATEST(n1, n2, n3);
END;
```

 **Ejemplo de uso:**

```
SELECT mayor_de_tres(12, 25, 8);
```

Salida: 25