

## PARTE 1

```
CREATE DATABASE tienda_online;
```

- Función para nombre completos

```
77 DELIMITER //
```

```
78
```

```
79 * CREATE FUNCTION nombreCompleto(p_cliente_id INT)
```

```
80 RETURNS VARCHAR(200)
```

```
81 DETERMINISTIC
```

```
82 BEGIN
```

```
83     RETURN
```

```
84     (SELECT CONCAT(nombre, ' ', apellido)
```

```
85     FROM Clientes
```

```
86     WHERE id_cliente = p_cliente_id
```

```
87     LIMIT 1);
```

```
88 END //
```

```
89
```

```
90 DELIMITER ;
```

```
91
```

```
92 * SELECT nombreCompleto(1) AS NombreCompleto;
```

```
93
```

```
94
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	NombreCompleto				
	Juan Pérez				

- Función para calcular el descuento

```
96
97 DELIMITER //
98
99 * CREATE FUNCTION calculardescuento(precio DECIMAL(10,2), descuento DECIMAL(5,2))
100 RETURNS DECIMAL(10,2)
101 DETERMINISTIC
102 BEGIN
103     RETURN precio - (precio * (descuento / 100));
104 END //
105
106 DELIMITER ;
107
108 * SELECT calculardescuento(50.00, 10) AS Descuento;
109
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Descuento
45.00

- Función para calcular el total de un pedido

```
111 DELIMITER //
112
113 * CREATE FUNCTION calculartotal(id_pedido INT)
114 RETURNS DECIMAL(10,2)
115 DETERMINISTIC
116 BEGIN
117     DECLARE total DECIMAL(10,2);
118     SELECT SUM(cantidad * precio_unitario) INTO total
119     FROM Detalles_Pedido
120     WHERE id_pedido = id_pedido;
121     RETURN total;
122 END //
123
124 DELIMITER ;
125
126 * SELECT calculartotal(1);
127
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

calculartotal(1)
470.00

- Función para verificar si cumple un stock con booleanos

```
131 • CREATE FUNCTION verificarstocks(id_producto INT, cantidad INT)
132     RETURNS BOOLEAN
133     DETERMINISTIC
134     BEGIN
135         DECLARE stock_actual INT;
136         SELECT stock INTO stock_actual
137         FROM Productos
138         WHERE id_producto = id_producto
139         LIMIT 1;
140
141         IF stock_actual >= cantidad THEN
142             RETURN TRUE;
143         ELSE
144             RETURN FALSE;
145         END IF;
146     END //
147
148     DELIMITER ;
149
150 • SELECT verificarstocks(1, 10) AS VerificarStock_1TRUE_0FALSE;
```

Result Grid

VerificarStock_1TRUE_0FALSE
1

- Función para calcular antigüedad de un cliente

```
154 DELIMITER //
```

```
155
```

```
156 * CREATE FUNCTION calcularantiguedads(id_clientes INT)
```

```
157 RETURNS INT
```

```
158 DETERMINISTIC
```

```
159 BEGIN
```

```
160     RETURN TIMESTAMPDIFF(
```

```
161         YEAR,
```

```
162         (SELECT fecha_registro FROM Clientes WHERE id_cliente = id_clientes LIMIT 1),
```

```
163         CURDATE()
```

```
164     );
```

```
165 END //
```

```
166
```

```
167 DELIMITER ;
```

```
168
```

```
169 * SELECT calcularantiguedads(1);
```

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	calcularantiguedads(1)
▶	4

## PARTE 2

- CREAR BASE

```
1 -- Crear la base de datos
```

```
2 * CREATE DATABASE tiendaonline;
```

```
3
```

```
4 -- Usar la base de datos
```

```
5 * USE tiendaOnline;
```

```
6
```

- CREAR TABLAS

```

7
8  -- Crear tabla Productos
9  * CREATE TABLE Productos (
10     ProductoID INT AUTO_INCREMENT PRIMARY KEY,
11     nombre VARCHAR(255) NOT NULL,
12     precio DECIMAL(10, 2) NOT NULL,
13     Existencia INT NOT NULL
14 );
15
16  -- Crear tabla Ordenes
17  * CREATE TABLE Ordenes (
18     OrdenID INT AUTO_INCREMENT PRIMARY KEY,
19     producto_id INT NOT NULL,
20     cantidad INT NOT NULL,
21     FOREIGN KEY (producto_id) REFERENCES Productos(ProductoID)
22 );
23
24  -- Crear tabla Usuarios
25  * CREATE TABLE Usuarios (
26     UsuarioID INT AUTO_INCREMENT PRIMARY KEY,
27     nombre VARCHAR(255) NOT NULL,
28     fecha_nacimiento DATE NOT NULL
29 );
30
31  -- Crear tabla Cuentas
32  * CREATE TABLE Cuentas (
33     CuentaID INT AUTO_INCREMENT PRIMARY KEY,
34     titular VARCHAR(255) NOT NULL
35 );
36

```

```

37  -- Crear tabla Transacciones
38  * CREATE TABLE Transacciones (
39     TransaccionID INT AUTO_INCREMENT PRIMARY KEY,
40     cuenta_id INT NOT NULL,
41     tipo_transaccion ENUM('deposito', 'retiro') NOT NULL,
42     monto DECIMAL(10, 2) NOT NULL,
43     fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
44     FOREIGN KEY (cuenta_id) REFERENCES Cuentas(CuentaID)
45 );
46

```

- AGREGAR REGISTROS
- CREAR FUNCIONES
- Crear función CalcularTotalOrden

*Calcula el total de una orden, sumando el precio de los productos multiplicado por la cantidad de cada uno y luego agregar el IVA*

```

86 DELIMITER //
87 * -- Crear función CalcularTotalOrden
88 CREATE FUNCTION CalcularTotalOrden(id_orden INT)
89 RETURNS DECIMAL(10, 2)
90 DETERMINISTIC
91 BEGIN
92     DECLARE total DECIMAL(10, 2);
93     DECLARE iva DECIMAL(10, 2);
94
95     SET iva = 0.15;
96
97     SELECT SUM(P.precio * O.cantidad) INTO total
98     FROM Ordenes O
99     JOIN Productos P ON O.producto_id = P.ProductoID
100    WHERE O.OrderID = id_orden;
101
102    SET total = total + (total * iva);
103
104    RETURN total;
105 END //
106
107 DELIMITER ;
108
109 * SELECT CalcularTotalOrden(1);
110

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	CalcularTotalOrden(1)			
	230.00			

- Crear función CalcularEdad

*Calcula la edad de una persona mediante su fecha de nacimiento usando la diferencia en años entre la fecha de nacimiento y la fecha actual.*

```

112 DELIMITER //
113
114 * -- Crear función CalcularEdad
115 CREATE FUNCTION CalcularEdad(fecha_nacimiento DATE)
116 RETURNS INT
117 DETERMINISTIC
118 BEGIN
119     DECLARE edad INT;
120     SET edad = TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE());
121     RETURN edad;
122 END //
123
124 DELIMITER ;
125
126 * SELECT CalcularEdad("1999,12,21") AS EDDAD;
127
128

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	EDDAD			
▶	25			

- Crear función VerificarStock  
Verifica si un producto tiene stock caso contrario retorna TRUE (1) si hay del y FALSE (0) si no la hay.

```

128
129     DELIMITER //
130
131 •   -- Crear función VerificarStock
132     CREATE FUNCTION VerificarStock(producto_id INT)
133     RETURNS BOOLEAN
134     DETERMINISTIC
135     BEGIN
136         DECLARE stock INT;
137
138         SELECT Existencia INTO stock
139         FROM Productos
140         WHERE ProductoID = producto_id;
141
142     IF stock > 0 THEN
143         RETURN TRUE;
144     ELSE
145         RETURN FALSE;
146     END IF;
147 END //
148
149     DELIMITER ;
150
151 •   SELECT VerificarStock(1);
152

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	VerificarStock(1)			
▶	1			

- Crear función CalcularSaldo  
Calcula el saldo de una cuenta sumando los depósitos y restando los retiros registrados en las transacciones de una cuenta.



```

153 DELIMITER //
154
155 • -- Crear función CalcularSaldo
156 CREATE FUNCTION CalcularSaldo(id_cuenta INT)
157 RETURNS DECIMAL(10, 2)
158 DETERMINISTIC
159 BEGIN
160     DECLARE saldo DECIMAL(10, 2);
161
162     SELECT SUM(CASE
163         WHEN tipo_transaccion = 'deposito' THEN monto
164         WHEN tipo_transaccion = 'retiro' THEN -monto
165         ELSE 0
166     END) INTO saldo
167     FROM Transacciones
168     WHERE cuenta_id = id_cuenta;
169
170     RETURN saldo;
171 END //
172
173 DELIMITER ;
174
175 • SELECT CalcularSaldo(2)

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	CalcularSaldo(2)			
▶	400.00			