-- Elimina la base de datos si ya existe (para fines de prueba) IF DB\_ID('SalesDB') IS NOT NULL DROP DATABASE SalesDB; GO

-- Crea la base de datos CREATE DATABASE SalesDB; GO

USE SalesDB; GO

-- Tabla de Clientes CREATE TABLE Clientes ( ClienteID INT IDENTITY(1,1) PRIMARY KEY, Nombre NVARCHAR(50) NOT NULL, Apellido NVARCHAR(50) NOT NULL, Email NVARCHAR(100) UNIQUE, Telefono NVARCHAR(20) ); GO

-- Tabla de Productos CREATE TABLE Productos ( ProductoID INT IDENTITY(1,1) PRIMARY KEY, NombreProducto NVARCHAR(100) NOT NULL, Precio DECIMAL(10,2) NOT NULL, Stock INT NOT NULL ); GO

-- Tabla de Pedidos CREATE TABLE Pedidos ( PedidoID INT IDENTITY(1,1) PRIMARY KEY, ClienteID INT NOT NULL, FechaPedido DATETIME NOT NULL DEFAULT GETDATE(), CONSTRAINT FK\_Pedidos\_Clientes FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID) ); GO

-- Tabla de DetallePedido CREATE TABLE DetallePedido ( DetalleID INT IDENTITY(1,1) PRIMARY KEY, PedidoID INT NOT NULL, ProductoID INT NOT NULL, Cantidad INT NOT NULL, PrecioUnitario DECIMAL(10,2) NOT NULL, CONSTRAINT FK\_DetallePedido\_Pedidos FOREIGN KEY (PedidoID) REFERENCES Pedidos(PedidoID), CONSTRAINT FK\_DetallePedido\_Productos FOREIGN KEY (ProductoID) REFERENCES Productos(ProductoID) ); GO

-- Tabla para registrar eliminaciones de pedidos CREATE TABLE HistorialPedidos ( HistorialID INT IDENTITY(1,1) PRIMARY KEY, PedidoID INT, ClienteID INT, FechaPedido DATETIME, FechaEliminacion DATETIME DEFAULT GETDATE() ); GO

-- Clientes INSERT INTO Clientes (Nombre, Apellido, Email, Telefono) VALUES ('Juan', 'Perez', '[juan.perez@example.com](mailto:juan.perez@example.com)', '111-222-333'), ('Ana', 'Gomez', '[ana.gomez@example.com](mailto:ana.gomez@example.com)', '222-333-444'), ('Luis', 'Martinez', '[luis.martinez@example.com](mailto:luis.martinez@example.com)', '333-444-555'); GO

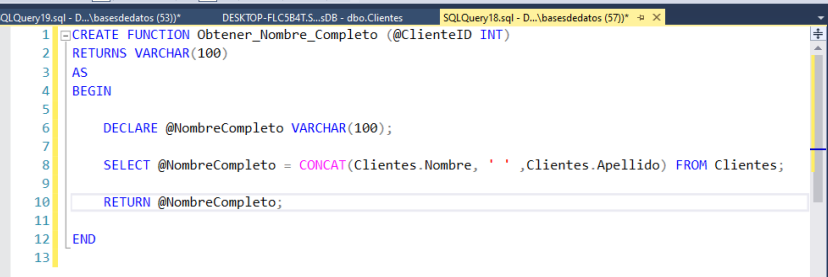
-- Productos INSERT INTO Productos (NombreProducto, Precio, Stock) VALUES ('Laptop', 800.00, 10), ('Smartphone', 500.00, 20), ('Tablet', 300.00, 15); GO

-- Pedidos INSERT INTO Pedidos (ClienteID, FechaPedido) VALUES (1, GETDATE()), (2, GETDATE()); GO

-- DetallePedido INSERT INTO DetallePedido (PedidoID, ProductoID, Cantidad, PrecioUnitario) VALUES (1, 1, 1, 800.00), (1, 2, 2, 500.00), (2, 2, 1, 500.00), (2, 3, 1, 300.00); GO

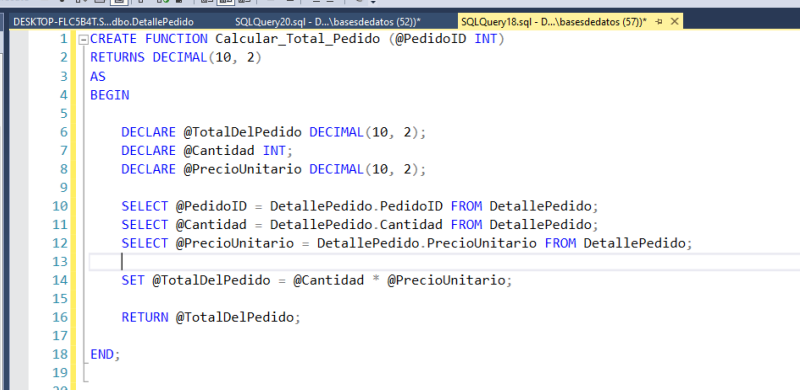
### **Enunciado 1: Función Escalar para Obtener Nombre Completo**

**Descripción:** Crea una función escalar llamada FN\_ObtenerNombreCompleto que reciba un ClienteID y retorne el nombre completo (concatenación de *Nombre* y *Apellido*) del cliente.



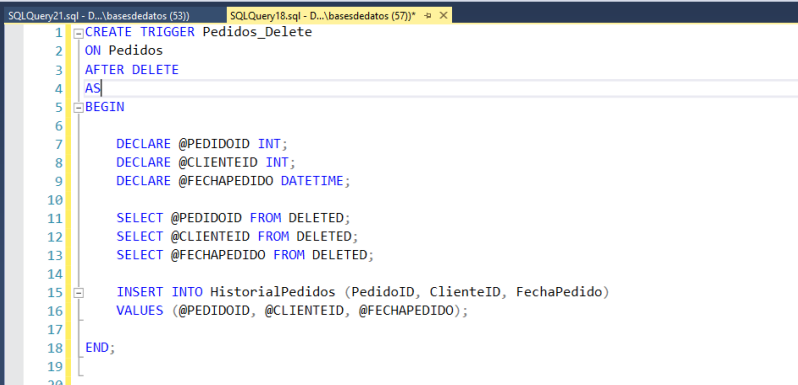
### **Enunciado 2: Función Escalar para Calcular el Total de un Pedido**

**Descripción:** Crea una función escalar llamada FN\_CalcularTotalPedido que reciba un PedidoID y retorne el total del pedido, calculando la suma de (Cantidad \* PrecioUnitario) de los registros correspondientes en la tabla **DetallePedido**.



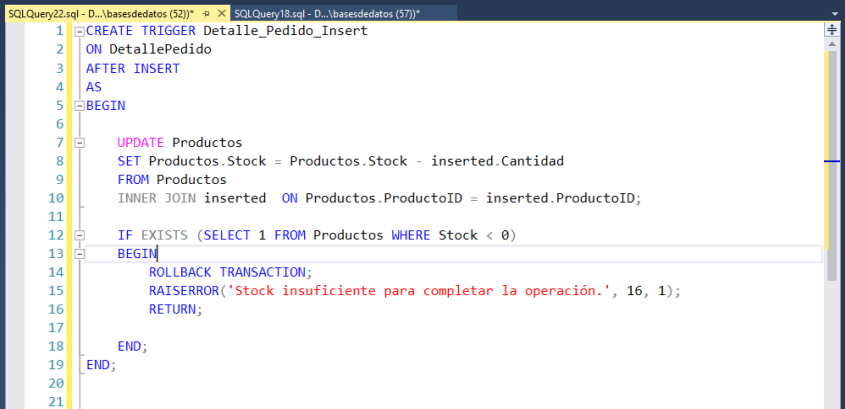
### **Enunciado 3: Trigger para Registrar Eliminación de Pedidos**

**Descripción:** Crea un trigger llamado TR\_Pedidos\_Delete sobre la tabla **Pedidos** que, después de eliminar un registro, inserte en la tabla **HistorialPedidos** los datos eliminados (PedidoID, ClienteID y FechaPedido).



### **Enunciado 4: Trigger para Actualizar Stock al Insertar en DetallePedido**

**Descripción:** Crea un trigger llamado TR\_DetallePedido\_Insert para la tabla **DetallePedido** que después de insertar un registro, actualice el stock del producto en la tabla **Productos** restándole la cantidad comprada. Si al actualizar el stock éste queda negativo, la transacción se revierte y se lanza un error.



### **Enunciado 5: Función de Tabla para Obtener Pedidos de un Cliente**

**Descripción:** Crea una función de tabla llamada TVF\_ObtenerPedidosCliente que reciba un ClienteID y retorne una tabla con los pedidos correspondientes a ese cliente, mostrando los campos *PedidoID*, *FechaPedido* y *TotalPedido* (obtenido a través de la función FN\_CalcularTotalPedido).

