

Problema 1

```
/*
LA TABLA CUSTOMERS SE LE AGREGO EL CAMPO LIMITECREDITO, EL CUAL REPRESENTA
EL LIMITE DE CREDITO QUE TIENE EL CLIENTE PARA CADA ORDEN, REALIZAR UN TRIGGER QUE VALIDE QUE NO
PERMITA ORDENES QUE PASEN EL LIMITE DE CREDITO
*/
CREATE TRIGGER TR_problema1 ON Orders
FOR INSERT, UPDATE AS BEGIN
    DECLARE @CLIENTEID NCHAR
    DECLARE @TOTALCOSTO MONEY
    DECLARE @LIMITECREDITO MONEY

    SELECT
        @CLIENTEID = i.CustomerID,
        @TOTALCOSTO = SUM(OD.UnitPrice * OD.Quantity)
    FROM Inserted i
    INNER JOIN [Order Details] OD ON i.OrderID = OD.OrderID
    GROUP BY i.CustomerID

    SELECT @LIMITECREDITO = LimiteCredito
    FROM Customers
    WHERE @CLIENTEID = CustomerID

    IF @TOTALCOSTO > @LIMITECREDITO
    BEGIN
        ROLLBACK TRAN
        RAISERROR('Se excedio el limite de Credito', 16, 1)
    END
END
```

Problema 2

```
/*
VALIDAR QUE UNA REGION (USANDO LA TABLA REGION) NO TENGA MAS DE 10
EMPLEADOS ASIGNADOS. LA REGION SE REFIERE A LA TABLA REGION QUE ESTA
RELACIONADA CON TERRITORIES Y ESTA TABLA CON EMPLOYEE TERRITORIES
NOTA: NO USAR EL CAMPO REGION DE EMPLOYEES
*/
CREATE TRIGGER TR_problema2 ON Region
FOR INSERT, UPDATE AS BEGIN
    DECLARE @REGIONID INT
    DECLARE @COUNTEMPLOYEES INT

    SELECT @REGIONID = RegionID
    FROM inserted

    SELECT @COUNTEMPLOYEES = COUNT(ET.EmployeeID)
    FROM EmployeeTerritories ET
    INNER JOIN Territories T ON ET.TerritoryID = T.TerritoryID
    WHERE @REGIONID = T.RegionID
    GROUP BY ET.EmployeeID

    IF @COUNTEMPLOYEES > 10
    BEGIN
        ROLLBACK TRAN
        RAISERROR('No se permiten mas de 10 empleados por Region', 16, 1)
    END
END
```

Problema 3

```
/*
LA TABLA PRODUCTS TIENE EL CAMPO MAXIMOPIEZASVENDIDAS, DICHO CAMPO REPRESENTA EL MAXIMO DE PIEZAS
VENDIDAS EN CADA ORDEN. REALIZAR UN TRIGGER QUE VALIDE QUE NO SE VENDAN MAS PIEZAS SEÑALADAS EN ESE
CAMPO.
*/

CREATE TRIGGER TR_problema3 ON [Order Details]
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @PRODUCTID INT
    DECLARE @CANTIDAD INT
    DECLARE @MAXIMOPIEZASVENDIDAS INT

    SELECT
        @PRODUCTID = I.ProductID,
        @CANTIDAD = I.Quantity,
        @MAXIMOPIEZASVENDIDAS = P.MaximoPiezasVendidas
    FROM Inserted I
    INNER JOIN Products P ON I.ProductID = P.ProductID

    IF @CANTIDAD > @MAXIMOPIEZASVENDIDAS
    BEGIN
        ROLLBACK TRAN
        RAISERROR('No se puede vender mas de lo permitido', 16, 1)
    END
END
```

Problema 4

```
/*
VALIDAR QUE UN EMPLEADO NO REALICE MAS DE 50 ORDENES POR SEMANA
*/
CREATE TRIGGER TR_problema4 ON Orders
AFTER INSERT
AS
BEGIN
    DECLARE @EMPLEADOID INT
    DECLARE @TOTALORDENES INT
    DECLARE @INICIOSEMANA DATE
    DECLARE @FINSEMANA DATE

    SET @INICIOSEMANA = DATEADD(WEEK, DATEDIFF(WEEK, 0, GETDATE()), 0)
    SET @FINSEMANA = DATEADD(DAY, 6, @INICIOSEMANA)

    SELECT
        @EMPLEADOID = EmployeeID,
        @TOTALORDENES = COUNT(OrderID)
    FROM INSERTED
    WHERE OrderDate BETWEEN @INICIOSEMANA AND @FINSEMANA
    GROUP BY EmployeeID

    IF @TOTALORDENES > 50
    BEGIN
        ROLLBACK TRAN
        RAISERROR('Se ha excedido el límite de 50 órdenes por empleado en una semana.', 16, 1)
    END
END
```