1.

```
CREATE VIEW view_Customers
AS
SELECT c.CustomerID,c.CompanyName,
       CASE
           WHEN SUM(od.UnitPrice*od.Quantity) < 5000 THEN 'BAJA'
             WHEN SUM(od.UnitPrice*od.Quantity) BETWEEN 5000 and 25000
THEN 'MEDIA'
         WHEN SUM(od.UnitPrice*od.Quantity) > 25000 THEN 'ALTA'
        END as Franja
FROM Orders o, OrderDetails od, Customers c
WHERE o.OrderID=od.OrderID and
      o.CustomerID=c.CustomerID
GROUP BY c.CustomerID,c.CompanyName
```

2.

```
Create Procedure RebajarInvendibles
@PorcentajeRebaja int, @CantidadProductosRebajados int OUTPUT
AS
BEGIN
      IF @PorcentajeRebaja BETWEEN 1 and 100
      BEGIN
      update Products
      set UnitPrice = UnitPrice - (UnitPrice*@PorcentajeRebaja/100)
      where not exists ( select * from OrderDetails OD where od.ProductID
= Products.productid )

      select @CantidadProductosRebajados= count(*)
            from Products where not exists ( select * from OrderDetails
OD where od.ProductID = Products.productid )
      END
    ELSE
       SELECT 'Rango invalido' as Mensaje
END
```

3.

```
CREATE FUNCTION DarMayorCantidad(@ProductID int)
  RETURNS decimal
AS
BEGIN
DECLARE @retorno decimal
SELECT @retorno=MAX(od.quantity)
FROM OrderDetails od
WHERE od.ProductID=@ProductID
GROUP BY od.ProductID

RETURN @retorno

END

-- Consulta utilizando la funcion
SELECT ProductID,ProductName,dbo.DarMayorCantidad(ProductID) as Mayor
FROM Products
```

4.

```
CREATE TRIGGER trg_Ej14
ON Products
INSTEAD OF insert
AS
BEGIN

  insert into Products SELECT i.* -- Cuando se pruebe poner todos los
campos ya que hay Identity
                       FROM inserted i,Suppliers s
                                WHERE i.SupplierID=s.SupplierID and
                                      s.Region is not null OR s.Phone not
like '%(02)%'

-- Mostramos los datos que no pasan la validacion
SELECT i.ProductName,i.SupplierID,s.CompanyName,'Proveedor no pasa la
validacion' as Mensaje
FROM inserted i,Suppliers s
WHERE i.SupplierID=s.SupplierID and
      s.Region is null OR s.Phone like '%(02)%'


END
```

5.

```
CREATE TABLE AuditSupplier(idAudit int identity primary key,
                           fchAudit datetime,
                                     SupplierID int,
                                     CompanyName varchar(40),
                                     Country varchar(15),
                                     NewCountry varchar(15),
                                     PostalCode varchar(10),
                                     NewPostalCode varchar(10))

CREATE TRIGGER trg_Ej7
ON Suppliers
AFTER update
AS
BEGIN
IF UPDATE(Country) OR UPDATE(PostalCode)
       insert into AuditSupplier SELECT
getdate(),i.SupplierID,i.CompanyName,
                                          d.Country,i.Country,

      d.PostalCode,i.PostalCode
                                   FROM inserted i, deleted d
                                               WHERE
i.SupplierID=d.SupplierID
END
```