

```
use Northwind
go
```

```
/* Ejercicio 1 - Programar un procedimiento almacenado que reciba por
parámetro un identificador de Producto y en el caso que se cumple que las
unidades en stock ( UnitsInStock ) sean cero,
y el producto no está Discontinuado: debe setear el producto como
Discontinuado. Y se debe retornar un booleano indicando si efectivamente
se marcó al producto como Discontinuado.
*/
```

```
CREATE or alter PROCEDURE Ejercicio1 @ProductId int , @SeActualizo bit
OUTPUT AS
BEGIN
    SET NOCOUNT ON
    SET @SeActualizo = 0
    IF EXISTS(SELECT * FROM Products where ProductId = @ProductId and
UnitsInStock=0 and Discontinued=0)
        BEGIN
            update Products set Discontinued=1 where ProductId =
@ProductId and Discontinued=0
            set @SeActualizo = 1
        END
    END
END
GO
```

```
update Products set UnitsInStock = 0, Discontinued =0 where ProductID = 1
go
DECLARE @RC int
DECLARE @ProductId int
DECLARE @SeActualizo bit
set @ProductId = 1

EXECUTE @RC = [dbo].[Ejercicio1] @ProductId ,@SeActualizo OUTPUT
GO
select Discontinued from Products where ProductID = 1
go
```

```
/* Ejercicio 2 - Programar una función que dado un id de Empleado y un
año: retorne el monto de descuento total de todas sus órdenes en ese año
(Aclaración: el campo Discount almacena el porcentaje de descuento, no el
monto de descuento. Entonces para obtener el descuento total, este
porcentaje debe aplicarse
sobre el Monto que se obtiene multiplicando el Precio por la Cantidad que
aparece en el Detalle de la Orden)
*/
```

```
CREATE or alter FUNCTION Ejercicio2 (@EmployeeId int, @Anio int) returns
int AS
BEGIN
    RETURN ( SELECT sum(UnitPrice*Quantity*Discount) from Orders O,
[Order Details] OD where O.OrderID = OD.OrderID and o.EmployeeID =
@EmployeeId and year(o.OrderDate) = @Anio)
END
GO
select dbo.Ejercicio2(1,1996)
```

```
SELECT * from Orders O, [Order Details] OD where O.OrderID = OD.OrderID
and o.EmployeeID = 1 and year(o.OrderDate) = 1996
```

```
/* Ejercicio 3 - Implementar un disparador que luego de una actualización
del campo UnitsInStock de la tabla Products, controle si se llegó al
nivel de stock de producto en el cual
se debe hacer una reposición (ReorderLevel). El objetivo es listar el/los
productos que requieren reposición, que son: los que se modificaron en la
sentencia de update que disparó el trigger,
que están activos (no-discontinuados) y que el nivel de stock ahora quedó
igual o menor al nivel de reposición del producto). Nota: Considere
modificaciones múltiples
*/
```

```
CREATE OR ALTER TRIGGER Ejercicio3 ON Products AFTER UPDATE AS
BEGIN
    IF update(UnitsInStock)
    BEGIN
        select * from Products where productid in (select ProductId
from Inserted) and UnitsInStock <= ReorderLevel and Discontinued = 0
    END
END
GO
select * from products
```

```
/* Ejercicio 4 - Implementar un disparador que no permita insertar "Order
Details" de productos que no hay suficiente saldo en stock para
satisfacer la orden
(dicho de otra forma: quedarían con saldo negativo de stock). En el caso
en que una misma inserción contenga mezcla de productos que quedarán con
stock válido (mayor o igual que cero) y
productos que quedarían con saldo negativo de stock: Se debe insertar
solamente los que quedan con stock válido, e ignorar la inserción de los
que quedarían con stock negativo.
Nota: Considere inserciones múltiples
*/
```

```
CREATE TRIGGER Ejercicio4 ON [Order Details] INSTEAD OF INSERT AS
BEGIN
    insert into [Order Details]
        select i.*
        from inserted i, Products p
        where i.ProductID = p.ProductID and p.Discontinued = 0 and
(p.UnitsInStock - i.Quantity) <0
END
GO
```

```
/* Ejercicio 5a - Crear la vista "ProductOrdered" con 2 campos: Id de
producto y cantidad total de unidades ordenadas (Quantity) en el año
actual (OrderDate) de ese producto. */
```

```
CREATE VIEW ProductOrdered AS
    SELECT OD.ProductID, sum(quantity) as CantidadOrdenada
    from Orders O inner join [Order Details] OD on o.OrderID =
OD.OrderID
    where year(OrderDate) = year(GetDate())
    GROUP BY OD.ProductID
```

```
/* Ejercicio 5b - Usar esa vista "ProductOrdered" para listar los  
productos de los que se ordenó 2000 o más unidades en el año actual.*/
```

```
select ProductId from ProductOrdered where CantidadOrdenada >= 2000
```