--1. Hacer un TRIGGER denominado TR\_stock\_dventas, que despues de insertar una fila a la tabla dventas valide si hay stock suficiente del producto (usar PA\_HayStockxAlmacen), si no hay Stock suficiente se debe cancelar el INSERT usando ROLLBACK

CREATE FUNCTION TR\_stock\_dventas(@cprd int, @calm int)

RETURNS DECIMAL(12,2)

AS

BEGIN

DECLARE @cant DECIMAL(12,2)

SET @cant = (SELECT isnull(sum(cant), 0) FROM sumi WHERE cprd=@cprd AND calm=@calm)

RETURN @cant

END

DROP TRIGGER TR\_stock\_deventas

CREATE TRIGGER TR\_stock\_deventas

ON dventas

FOR INSERT

AS

DECLARE @cprd INT, @calm INT, @cant DECIMAL, @nvta INT

SELECT @cprd = cprd, @nvta = nvta, @cant=cant FROM INSERTED

SELECT @calm = calm FROM pventas WHERE nvta = @nvta

IF @cant > dbo.TR\_stock\_dventas(@cprd, @calm)

BEGIN

RAISERROR('El producto no tiene stock suficiente...', 16, 1)

ROLLBACK

RETURN

END

--PRUEBA

delete from dventas where nvta = 1

delete from pventas where nvta = 1

select \* from pventas

select \* from dventas

print dbo.stock(2, 1)

insert into pventas values(1, 'Julanito', 1, GETDATE(), 0, 0)

insert into dventas values(1, 1, 200, 4, 800)

insert into dventas values(1, 2, 100, 2, 200)

--2. Hacer un TRIGGER denominado TR\_del\_dventas, que después de borrar una fila en la tabla dventas actualice el Importe Total en la tabla pventas (usar PA\_TotalPreVenta).

CREATE FUNCTION PA\_TotalPreVenta(@nvta INT)

RETURNS DECIMAL(12,2)

AS

BEGIN

DECLARE @ntotal DECIMAL

SET @ntotal = (SELECT isnull(sum(impt), 0) FROM dventas WHERE nvta = @nvta)

RETURN @ntotal

END

CREATE TRIGGER TR\_del\_Preventas

ON dventas

FOR DELETE

AS

BEGIN

DECLARE @nvta INT, @nitot DECIMAL

SELECT @nvta = nvta FROM DELETED

SET @nitot = dbo.PA\_TotalPreVenta(@nvta)

UPDATE pventas SET itot = @nitot WHERE nvta = @nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 2

delete from pventas where nvta = 2

select \* from pventas where nvta = 2

select \* from dventas where nvta = 2

insert into pventas values(2, 'PEDRO', 1, GETDATE(), 26, 0)

insert into dventas values(2, 1, 1, 5, 5)

insert into dventas values(2, 2, 3, 7, 21)

delete from dventas where nvta = 2 and cprd = 2

--3. Hacer un TRIGGER denominado TR\_ins\_dventas, que después de insertar una fila en la tabla dventas actualice el Importe Total en la tabla pventas (usar PA\_TotalPreVenta).

CREATE TRIGGER TR\_ins\_dventas

ON dventas

FOR INSERT

AS

BEGIN

DECLARE @nvta INT, @nitot DECIMAL

SELECT @nvta = nvta FROM INSERTED

SET @nitot = dbo.PA\_TotalPreVenta(@nvta)

UPDATE pventas SET itot = @nitot WHERE nvta = @nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 3

delete from pventas where nvta = 3

select \* from pventas where nvta = 3

select \* from dventas where nvta = 3

insert into pventas values(3, 'PEDRO', 1, GETDATE(), 0, 0)

insert into dventas values(3, 1, 2, 15, 30)

insert into dventas values(3, 2, 3, 7, 21)

--4. Hacer un TRIGGER denominado TR\_upd\_dventas, que después de actualizar una fila en la tabla dventas actualice el Importe Total (itot) en la tabla pventas (usar PA\_TotalPreVenta).

CREATE TRIGGER TR\_upd\_dventas

ON dventas

FOR UPDATE

AS

BEGIN

DECLARE @nvta INT, @nitot DECIMAL

SELECT @nvta = nvta FROM INSERTED

SET @nitot = dbo.PA\_TotalPreVenta(@nvta)

UPDATE pventas SET itot = @nitot WHERE nvta = @nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 4

delete from pventas where nvta = 4

select \* from pventas where nvta = 4

select \* from dventas where nvta = 4

insert into pventas values(4, 'JUAN', 1, GETDATE(), 0, 0)

insert into dventas values(4, 1, 2, 15, 30)

insert into dventas values(4, 2, 3, 7, 21)

update dventas set cant=5, prec=8, impt=40 where nvta=4

--5. Hacer un TRIGGER denominado TR\_del\_dventas, que después de borrar una fila en la tabla dventas actualice el Importe de descuento (ides) en la tabla pventas (usar PA\_DescPreVenta). Ademas si ya no hay mas filas para esa venta en la tabla dventas que también eliminé la filade la tabla pventas.

DROP FUNCTION PA\_DescPreVenta

CREATE FUNCTION PA\_DescPreVenta(@nvta INT)

RETURNS DECIMAL(12,2)

AS

BEGIN

DECLARE @itot DECIMAL(12,2)

SET @itot = dbo.PA\_TotalPreVenta(@nvta) \* 0.1

RETURN @itot

END

DROP TRIGGER TR\_del\_dventas\_2

CREATE TRIGGER TR\_del\_dventas\_2

ON dventas

FOR DELETE

AS

BEGIN

DECLARE @nvta INT, @ides DECIMAL(12,2), @cant INT

SELECT @nvta=nvta from DELETED

select @cant=count(\*) from dventas where nvta=@nvta

IF(@cant = 0)

BEGIN

delete from pventas where nvta=@nvta

RETURN

END

SET @ides = dbo.PA\_DescPreVenta(@nvta)

update pventas set ides=@ides where nvta=@nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 5

delete from pventas where nvta = 5

select \* from pventas where nvta = 5

select \* from dventas where nvta = 5

insert into pventas values(5, 'CARLOS', 1, GETDATE(), 0, 2.6)

insert into dventas values(5, 1, 1, 5, 5)

insert into dventas values(5, 2, 3, 7, 21)

delete from dventas where nvta = 5 and cprd = 2

--6. Hacer un TRIGGER denominado TR\_ins\_dventas, que después de insertar una fila en la tabla dventas actualice el Importe de descuento (ides) en la tabla pventas (usar PA\_DescPreVenta).

DROP TRIGGER TR\_ins\_dventas\_2

CREATE TRIGGER TR\_ins\_dventas\_2

ON dventas

FOR INSERT

AS

BEGIN

DECLARE @nvta INT, @ides DECIMAL(12,2), @cant INT

SELECT @nvta=nvta from INSERTED

SET @ides = dbo.PA\_DescPreVenta(@nvta)

update pventas set ides=@ides where nvta=@nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 6

delete from pventas where nvta = 6

select \* from pventas where nvta = 6

select \* from dventas where nvta = 6

insert into pventas values(6, 'CARLOS', 1, GETDATE(), 0, 0)

insert into dventas values(6, 1, 1, 5, 5)

insert into dventas values(6, 2, 3, 7, 21)

--7. Hacer un TRIGGER denominado TR\_upd\_dventas, que después de actualizar una fila en la tabla dventas actualice el Importe de descuento (ides) en la tabla pventas (usar PA\_DescPreVenta).

DROP TRIGGER TR\_upd\_dventas\_2

CREATE TRIGGER TR\_upd\_dventas\_2

ON dventas

FOR UPDATE

AS

BEGIN

DECLARE @nvta INT, @ides DECIMAL(12,2), @cant INT

SELECT @nvta=nvta from DELETED

SET @ides = dbo.PA\_DescPreVenta(@nvta)

update pventas set ides=@ides where nvta=@nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 7

delete from pventas where nvta = 7

select \* from pventas where nvta = 7

select \* from dventas where nvta = 7

insert into pventas values(7, 'JUAN', 1, GETDATE(), 0, 0)

insert into dventas values(7, 1, 2, 15, 30)

insert into dventas values(7, 2, 3, 7, 21)

update dventas set cant=5, prec=8, impt=40 where nvta=7

--8. Hacer un TRIGGER denominado TR\_del\_PreVentas, que borre toda la venta incluyendo su detalle (usar PA\_DelPreVentas).

DROP FUNCTION PA\_DelPreVentas

CREATE FUNCTION PA\_DelPreVentas(@nvta INT)

RETURNS INT

AS

BEGIN

DECLARE @cant INT

select @cant=count(\*) from dventas where nvta=@nvta

RETURN @cant

END

DROP TRIGGER TR\_del\_PreVentas\_2

CREATE TRIGGER TR\_del\_PreVentas\_2

ON pventas

AFTER DELETE

AS

BEGIN

DECLARE @nvta INT, @susc INT

select @nvta=nvta from deleted

SET @susc = dbo.PA\_DelPreVentas(@nvta)

delete from dventas where nvta=@nvta

RETURN

END

--PRUEBA

delete from dventas where nvta = 8

delete from pventas where nvta = 8

select \* from pventas where nvta = 8

select \* from dventas where nvta = 8

insert into pventas values(8, 'JUAN', 1, GETDATE(), 0, 0)

insert into dventas values(8, 1, 2, 15, 30)

insert into dventas values(8, 2, 3, 7, 21)

delete from pventas where nvta=8

--9. Hacer un TRIGGER denominado TR\_ins\_log, que despues de insertar una fila en la tabla pventas inserte los datos en la tabla log\_pventas

DROP TRIGGER TR\_ins\_log

CREATE TRIGGER TR\_ins\_log

ON pventas

FOR INSERT

AS

BEGIN

DECLARE @nvta INT

select @nvta=nvta from inserted

insert into log\_pventas values(@nvta, getdate(), user, 'I')

RETURN

END

--PRUEBA

select \* from pventas

select \* from log\_pventas

insert into pventas values(9, 'JUAN', 1, GETDATE(), 0, 0)

--10. Hacer un TRIGGER denominado TR\_del\_log, que despues de borarr una fila en la tabla pventas inserte los datos en la tabla log\_pventas

DROP TRIGGER TR\_del\_log

CREATE TRIGGER TR\_del\_log

ON pventas

FOR DELETE

AS

BEGIN

DECLARE @nvta INT

select @nvta=nvta from deleted

insert into log\_pventas values(@nvta, getdate(), user, 'D')

RETURN

END

--PRUEBA

select \* from pventas

select \* from log\_pventas

insert into pventas values(10, 'JUAN', 1, GETDATE(), 0, 0)

delete from pventas where nvta=10

--11. Hacer un TRIGGER denominado TR\_upd\_log, que despues de actualizar una fila en la tabla pventas inserte los datos en la tabla log\_pventas

DROP TRIGGER TR\_upd\_log

CREATE TRIGGER TR\_upd\_log

ON pventas

FOR UPDATE

AS

BEGIN

DECLARE @nvta INT

select @nvta=nvta from deleted

insert into log\_pventas values(@nvta, getdate(), user, 'U')

RETURN

END

--PRUEBA

select \* from pventas

select \* from log\_pventas

insert into pventas values(11, 'JUAN', 1, GETDATE(), 0, 0)

update pventas set nomc = 'JUAN2' where nvta=11