1. Sobre la bd Jardinería crear un *TRIGGER* que bloquee actualizar el Código del producto en el proceso de actualización.

USE JARDINERIA;

go

CREATE TRIGGER TX\_PRODUCTO\_ACTUALIZA\_cod ON dbo.productos

FOR UPDATE AS

IF UPDATE (CodigoProducto)

BEGIN

PRINT 'NO SE PUEDE ACTUALIZAR EL ID DEL PRODUCTO'

ROLLBACK TRANSACTION

end

------------------

UPDATE Productos set CodigoProducto=1 where Nombre='Pala'

No se puede verificar pq hay restricción con la tabla detallePedidos

1. Sobre la BD Compras crear un trigger tal que cuando se vaya a introducir o modificar una línea de un artículo descatalogado no lo permita y devuelva el mensaje de error:

'No puede seleccionar un artículo descatalogado'

USE compras;

go

CREATE TRIGGER tr\_lineas ON dbo.Lineas AFTER INSERT, UPDATE AS BEGIN

DECLARE @errmsg char(255)

IF ( SELECT count(\*) FROM inserted i, articulos a

WHERE i.codigart=a.codigart and a.fecbaja is not null) >0

BEGIN

set @errmsg = 'No puede seleccionar un artículo descatalogado'

RAISERROR (@errmsg,16,1)

ROLLBACK TRANSACTION

RETURN

END

END

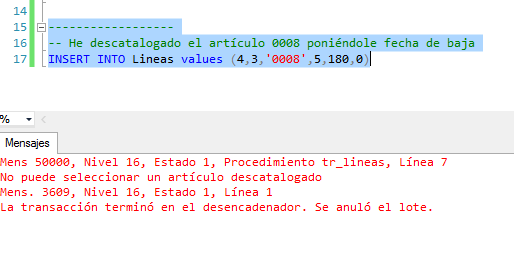
NOTA: con un IF EXISTS también funcionaría

------------------

-- He descatalogado el artículo 0008 poniéndole fecha de baja

INSERT INTO Lineas values (4,3,'0008',5,180,0)

Ver el error que se produce.



1. Sobre la bd de compras implementar el código necesario tal que cuando se intenten vender más de la mitad de las unidades que tengamos en stock de un artículo de una vez no se permita.

use compras;

go

create trigger tr\_stock on dbo.Articulos

FOR UPDATE

AS

IF EXISTS (SELECT \* from inserted i join deleted d on i.codigart=d.codigart

WHERE (d.stockart-i.stockart)>d.stockart/2)

BEGIN

RAISERROR('No se puede reducir el stock en más de un 50 %% de una vez.', 16,1)

ROLLBACK TRAN

END

-----------------------------

UPDATE dbo.Articulos

set stockart=25 where codigart='0001'

1. Sobre la bd Compras, crear una tabla eventos con los campos, fecha char(6) y motivo varchar(200). Dicha tabla se rellenará cada vez que en la tabla artículos el stock de un producto alcanza el mínimo establecido.

use compras;

go

CREATE table eventos(

fecha char(6),

motivo varchar(200))

go

CREATE TRIGGER tr\_SUP\_STOCK ON articulos

FOR UPDATE

AS

BEGIN

if UPDATE(stockart) BEGIN

INSERT INTO eventos (fecha, motivo)

SELECT getdate(), 'Stock minimo alcanzado ' + codigart FROM inserted

WHERE stockart<=stockmin

END

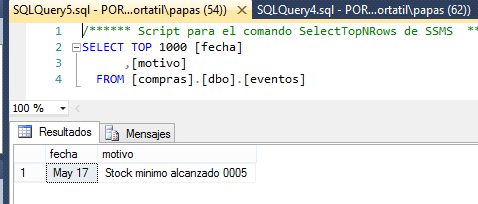
END

---------------------------

-- Deshabilito temporalmente el trigger de no disminuir de golpe más del 50%

DISABLE TRIGGER tr\_stock on articulos

UPDATE articulos set stockart=0 where codigart='0005'



1. Sobre la bd compras, cuando se quiera borra un artículo se sustituya por actualizar la fecha de baja en la tabla artículos.

use compras;

go

CREATE TRIGGER trd\_articulos on articulos

INSTEAD OF DELETE AS

IF @@ROWCOUNT = 0 – si no hay ningún artículo afectado por el delete que no haga nada

RETURN

UPDATE articulos SET fecbaja = getdate()

FROM articulos INNER JOIN deleted d ON d.codigart = articulos.codigart

------

DELETE from articulos where codigart='0005'

1. En la bd Jardinería trigger que avisa al usuario que se ha eliminado un registro de la tabla pagos.

use JARDINERIA;

go

CREATE TRIGGER BorraPago

ON pagos FOR DELETE AS

PRINT '¡Acaba de eliminar una fila! '

----

DELETE FROM pagos where codigoCliente=38

Te avisa que ha eliminado una fila y la borra. Si vuelvo a ejecutar lo anterior aunque no borra más filas (pq no hay ninguna que cumpla el criterio) te vuelve a avisar ( se dispara igual el trigger). Para evitarlo:

Solución1: Avisando el número de filas eliminadas. Si es cero lo avisa.

use JARDINERIA;

go

ALTER TRIGGER BorraPago

ON pagos FOR DELETE AS

declare @contador int

select @contador=count(\*) from deleted

PRINT '¡Ha eliminado '+convert(char(3),@contador)+'fila(s)!'

----

DELETE FROM pagos where codigoCliente=38

Solución 2

use JARDINERIA;

go

ALTER TRIGGER BorraPago

ON pagos FOR DELETE AS

IF @@ROWCOUNT=0

RETURN

PRINT '¡Ha eliminado fila(s)!'

----

DELETE FROM pagos where codigoCliente=38

1. Sobre la bd Banco crear un trigger que cree un registro en la tabla numrojos con los campos id\_fila autoincremental, dni\_cliente, id\_cuenta, fecha y saldo cada vez que algún cliente se quede en números rojos en alguna de sus cuentas.

SOLUCIÓN VÁLIDA MODIFIQUE SALDO DE UNA O VARIAS CUENTAS.

use banco;

go

alter TRiGGER tr\_numrojos on cuenta

for UPDATE as

begin

if UPDATE(saldo)

IF EXISTS (SELECT \* from inserted i

WHERE i.saldo <0)

INSERT INTO numrojos SELECT getdate(),c.dni,i.cod\_cuenta,i.saldo from cliente c , tiene t, inserted i where c.dni=t.dni and t.cod\_cuenta=i.cod\_cuenta and i.saldo <0

end

Válido si sólo modifico una

CREATE TRiGGER tr\_numrojos on cuenta

for UPDATE as

DECLARE @id\_cliente int

IF UPDATE(saldo)

IF EXISTS (SELECT \* from inserted i

WHERE i.saldo <0)

BEGIN

set @id\_cliente=(SELECT dni from tiene join inserted on tiene.cod\_cuenta=inserted.cod\_cuenta where inserted.saldo <0)

INSERT INTO numrojos VALUES(getdate(),@id\_cliente,(select cod\_cuenta from inserted i where i.saldo <0),

(select saldo from inserted i where i.saldo <0))

end

Solución cuando se actualiza más de un registro a la vez: cursores

use banco;

go

Create table numrojos(

id int identity primary key,

fecha date, cdni int,

ccuenta int,

saldo int)

go

create TRiGGER tr\_numrojos on cuenta

for UPDATE as

begin

DECLARE @id\_cliente int

DECLARE @c\_cuenta int

DECLARE @saldo int

IF EXISTS (SELECT \* from inserted i

WHERE i.saldo <0)

BEGIN

DECLARE MI\_CURSOR CURSOR for select c.dni,i.cod\_cuenta,i.saldo from cliente c , tiene t, inserted i where c.dni=t.dni and t.cod\_cuenta=i.cod\_cuenta and i.saldo <0

open MI\_CURSOR

FETCH NEXT FROM MI\_CURSOR INTO @id\_cliente, @c\_cuenta,@saldo

WHILE @@FETCH\_STATUS=0

BEGIN

INSERT INTO numrojos VALUES(getdate(),@id\_cliente,@c\_cuenta,@saldo)

FETCH NEXT FROM MI\_CURSOR INTO @id\_cliente, @c\_cuenta,@saldo

end

CLOSE MI\_CURSOR

DEALLOCATE MI\_CURSOR

END

end

----------------

UPDATE cuenta

SET saldo=saldo\*(-1) where cod\_cuenta between 2 and 3

1. Sobre la bd banco implementa un trigger talque imprime un mensaje en el cliente cuando alguien intenta cambiar datos en la tabla tiene.

USE banco

GO

CREATE TRIGGER tr\_NO\_Tocar ON tiene

AFTER INSERT, UPDATE

AS

BEGIN

RAISERROR ('No puede insertar ni modificar esta tabla', 16, 10)

1. Sobre la bd compras, creamos un disparador para que no permita eliminar más de un registro a la vez de la tabla Proveedores:

USE compras;

go

create trigger tr\_proveedores\_borrar on proveedores

for delete

as

if (select count(\*) from deleted)>1

begin

raiserror('No puede eliminar más de un 1 proveedor', 16, 1)

rollback transaction

end;

1. Hacer lo necesario para que cada vez que un cliente de la bd banco ingrese más de 1000 euros se le bonifique con 100, sólo para clientes con cuentas que superen los tres años de antigüedad entre el 1 de enero del 2015 y el 31 de Julio del 2015. Hacerlo sobre la tabla movimiento.

Para resolverlo creo un trigger y una función( para calcular la antigüedad de la cuenta)

EL TRIGGER SOLO SIRVE SI INSERTO UN ÚNICO MOVIMIENTO. SI INSERTO MÁS HABRÍA QUE USAR CURSORES POR EJEMPLO.

USE banco;

go

Create function ant\_cuenta(@ccuenta int) RETURNS INT

as

begin

declare @vfecha datetime;

declare @años int;

SET @vfecha=(SELECT fecha\_creacion from cuenta where cod\_cuenta=@ccuenta)

SET @años=datediff(yy,@vfecha,getdate())

RETURN(@años)

end

------------------------------------------------------

go

create Trigger tr\_bono on movimiento after INSERT

as

begin

IF (select cantidad from inserted)> 1000

begin

DECLARE @ant INT

DECLARE @ID\_cuenta int

SET @ID\_cuenta =(select cod\_cuenta from inserted)

SET @ant=dbo.ant\_cuenta (@ID\_cuenta)

IF(cast((select fecha from inserted) as date) >'2015-01-01' and cast((select fecha from inserted) as date)<'2015-07-31' and @ant >3)

UPDATE cuenta set saldo=saldo+100 where cod\_cuenta=(select cod\_cuenta from inserted)

end

end

---------------------

INSERT INTO movimiento values (getdate(),2000,117,1,31295)

1. Crear un desencadenador para la tabla productos en la bd Jardinería, que se active cada vez que se inserte un registro o se actualice la columna precio, la condición para aceptar al inserción o la actualización es que el precio costo no debe ser mayor que el precio venta.

USE JARDINERIA;

GO

CREATE TRIGGER TR\_PRODUCTO ON Productos AFTER INSERT, UPDATE AS

begin

IF UPDATE(PrecioVenta)

begin

IF(Select PrecioVenta from inserted) < (select PrecioProveedor from inserted)

begin

PRINT 'El precio de venta es menor que el precio de compra'

ROLLBACK TRAN

end

end

end

------

INSERT INTO Productos values ('kk', 'kkk','Herramientas','0,258','HiperGarden Tools','kkkk',15,8,7.04)

UPDATE Productos set PrecioVenta=18 where CodigoProducto='kk'

1. Crear trigger asociado a la tabla Pedidos, de manera que cuando queramos borrar un registro de la tabla pedidos elimine todos los registros relacionados de la tabla de detallepedidos. Hay que tener en cuenta que puedo borrar varios pedidos a la vez

SOLUCIÓN ÓPTIMA

CREATE TRIGGER tr\_borrar\_pedido on pedidos after delete

as

begin

IF (select count(\*) from deleted)>=1

DELETE FROM DetallePedidos where CodigoPedido in (select CodigoPedido from deleted)

end

OTRA SOLUCIÓN USANDO CURSORES

use JARDINERIA;

go

CREATE TRIGGER tr\_borrar\_pedido on pedidos after delete

as

begin

declare @ID\_pedido int

IF (select count(\*) from deleted)>=1

begin

DECLARE MI\_CURSOR cursor for select codigoPedido from deleted

open MI\_CURSOR

FETCH NEXT from MI\_CURSOR INTO @ID\_pedido

WHILE @@FETCH\_STATUS=0

begin

DELETE FROM DetallePedidos where CodigoPedido=@ID\_pedido

FETCH NEXT from MI\_CURSOR INTO @ID\_pedido

end

close MI\_CURSOR

DEALLOCATE MI\_CURSOR

end

end

------------------

delete from Pedidos where CodigoPedido >=49

Entra en conflicto con la restricción FK y que no se pueda eliminar nada cuando elimino un código de producto.