--Lista de Exercícios – Programação para Banco de Dados Triggers

--Prof. Me. Mateus Guilherme Fuini

--Utilizando o BD Pedido, desenvolva os exercícios a seguir:

--1. Usando comandos da DDL, insira com campo chamado estado (char(2)) na tabela Clientes.

ALTER TABLE dbo.cliente ADD estado char(20)

SELECT \* FROM dbo.cliente

--2. Escreva uma trigger que evite que sejam inseridos novos clientes na tabela clientes em que o campo UF seja “RJ” ou “ES”. Esta trigger será criada para a ação INSERT. Na tentativa de inserção de cliente de um destes estados emitir uma mensagem utilizando raiserror: “INSERÇÃO DO REGISTRO CANCELADA” e retroceder a transação. Para os demais clientes emitir mensagem “ESTADO PERMITIDO”

CREATE TRIGGER EX\_2

ON dbo.cliente

FOR INSERT

AS

BEGIN

Declare @campoUF char(2)

set @campoUF = (select UF from inserted)

IF @campoUF = 'RJ' OR @campoUF = 'ES'

BEGIN

RAISERROR('INSERÇÃO DO REGISTRO CANCELADO', 10, 1)

ROLLBACK

END

ELSE

BEGIN

PRINT('ESTADO PERMITIDO')

END

END

insert into dbo.cliente (Cod\_cliente,Nome\_cliente,UF) values(22,'Feranndo','ES')

--3. Escreva uma trigger que seja disparada quando da tentativa de inclusão de novos produtos na tabela produtos. Esta trigger deve emitir um aviso que as inserções dos produtos estão suspensas e registrar em uma tabela chamada tlogproduto com a seguinte estrutura:

Create table tlogproduto(

usuario varchar(30),

dataalteracao datetime,

nome\_produto varchar(20))

<--EXEMPLO-->

EXEMPLO

CREATE TRIGGER TG\_NAOALTERA

ON tlogproduto

FOR INSERT

AS

BEGIN

print 'INSERÇÕES CANCELADAS'

ROLLBACK

END

INSERT INTO tlogproduto (usuario) VALUES ('fernando')

--4. Escreva um trigger que ao cadastrar um item da nota, atualize a tabela de produtos da seguinte forma:

-- Se qtd\_Est (tab. produto) >= qtde (Pedida - tab inserted)

--Então atualizar:

-- qtde estoque (tab. produto) = qtde estoque(tab. produto) - qtde (tab inserted)

-- Caso contrário

--Utilizar raiserror para emitir mensagem: ”ESTOQUE INSUFICIENTE”, incluindo também na mensagem o código do produto que está em falta Retroceder a transação.

ALTER TABLE dbo.produto ADD qtd\_Est int

insert into dbo.produto (cod\_produto,qtd\_Est) values(55,2)

select \* from item\_pedido

insert into dbo.item\_pedido (num\_pedido,cod\_produto,quantidade) values(8,55,3)

select \* from dbo.produto

CREATE TRIGGER checkEstoque

ON dbo.item\_pedido

FOR INSERT

AS

BEGIN

DECLARE @qtdInserida int

SET @qtdInserida = (select quantidade from inserted)

DECLARE @codProdInserido int;

SET @codProdInserido = (select cod\_produto from inserted)

DECLARE @qtdEmEstoque int

SET @qtdEmEstoque = (select qtd\_Est from produto where cod\_produto = @codProdInserido )

if @qtdInserida > @qtdEmEstoque

BEGIN

RAISERROR('ESTOQUE INSUFICIENTE DO PRODUTO: %d', 10, 1,@codProdInserido)

ROLLBACK

END

else

BEGIN

update produto set qtd\_Est = (qtd\_Est-@qtdInserida)

END

END

select \* from dbo.item\_pedido;

INSERT INTO dbo.item\_pedido (num\_pedido,cod\_produto,quantidade,pco\_unit) VALUES (9,56,50,20)

--5. Escreva um trigger que ao ser incluído um pedido na tabela pedido verifique se a data de entrega está programada para domingo ( IF datepart( dw, dataentrega) =1). Neste caso alterar a data de entrega para a segunda-feira próxima

CREATE TRIGGER check\_Pedido

ON dbo.pedido

AFTER INSERT

AS

BEGIN

DECLARE @numPedido int;

SET @numPedido = (SELECT Num\_pedido FROM inserted)

DECLARE @dataDeCompra date

SET @dataDeCompra = (SELECT data\_compra FROM inserted)

DECLARE @dataDeEntrega date

SET @dataDeEntrega = (SELECT data\_entrega FROM inserted)

print @dataDeEntrega

DECLARE @newDate date;

if(datepart(dw,@dataDeEntrega) = 1)

BEGIN

--Soma mais +1

SET @newDate = DATEADD(DD,1,@dataDeEntrega)

update pedido set data\_entrega = @newDate where Num\_pedido = @numPedido

END

END

insert into pedido (Num\_pedido,data\_compra,data\_entrega,cod\_cliente) VALUES(25,getDate(),getDate()+3,5)

-- 6. Escreva um Trigger que ao ser inserido um cliente verifique se o campo Fone está nulo, se estiver emita mensagem de erro utilizando raiserror, incluindo também na mensagem o código do cliente, e retroceda a transação, caso contrário efetive a transação.

-- 7. Escreva um trigger que ao ser inserido um produto verifique se o preço é maior que $0.00. Se não for emitir mensagem de erro utilizando raiserror, incluindo também na mensagem o código do produto e, retroceder a transação, caso contrário efetive a transação.

-- 8. Fazer uma trigger que dispare após qualquer inserção ou deleção na tabela item\_pedido. Para cada atualização deve ser inserida uma linha em tlogitem com as seguintes informações: código do usuário que fez a atualização(current\_user), data da atualização(data do sistema), num\_nota, cod\_produto e o tipo de atualização (inserção ou deleção).

ALTER TABLE dbo.item\_pedido ADD num\_nota varchar(15)

ALTER TABLE dbo.produto ADD nome\_produto varchar(30)

ALTER TABLE dbo.tlogproduto ALTER COLUMN cod\_produto int

ALTER TABLE dbo.tlogproduto ALTER COLUMN tipo\_atuali

ALTER TRIGGER ex8

ON dbo.item\_pedido

AFTER INSERT, DELETE

AS

DECLARE @codUsuario INT

SET @codUsuario = CURRENT\_USER

DECLARE @dataUpdated DATE

SET @dataUpdated = GETDATE()

DECLARE @numNota INT

SET @numNota = (select num\_nota from inserted)

DECLARE @codProd INT

SET @codProd = (SELECT cod\_produto FROM inserted)

DECLARE @tipoDeModif varchar(15)

SET @tipoDeModif = 'TESTE - DELETE'

DECLARE @nomeProduto varchar(30)

SET @nomeProduto = (select nome\_produto from dbo.produto where cod\_produto = @codProd)

--Obs. Criar Tlogitem com os campos descritos acima.

--9. Escreva um trigger que ao ser alterada uma linha na tabela vendedor seja gerada uma linha na tabela Tlog com o usuário que fez a alteração, a data do sistema, o código do vendedor alterado e os valores anteriores à atualização, caso tenham sido alterados).

--Obs.: Antes de iniciar a trigger a tabela Tlog (se não existir) deve ser criada com a seguinte estrutura:

Create table tlog (

usuario varchar(30),

dataalteracao datetime,

codigo\_vendedor int,

salario\_fixo Money,

faixa\_comissao char(1),

porc\_comissao float)

--10. Escreva um trigger que ao ser alterado o valor unitário de um produto, insira uma linha na tabela histórico de preços que possui a seguinte estrutura: codigo - int, identity, chave primária, cod\_produto - int, valor\_unitario(antigo) - Money, data\_alteração - datetime. Após a inclusão na tabela históricos verificar o valor de @@error, caso seja diferente de zero imprimir mensagem de erro utilizando raiserror e retroceder a transação, caso contrário efetivar a transação.

--11. Fazer uma trigger que dispare após qualquer atualização (inserção, deleção, alteração) na tabela cliente. Para cada atualização deve ser inserida uma linha em tlogcliente com as seguintes informações: código do usuário que fez a atualização, data da atualização(data do sistema), código do cliente e o tipo de atualização (inserção, deleção ou alteração)

--12. Escreva um trigger que ao ser alterada a data da entrega de um pedido na tabela de pedido verifique se a nova data de entrega é um domingo. Neste caso some 1 na data de entrega.

--13. Refazer os exercícios 2, 3, 4, 6 e 8 e utilizar Triggers do tipo Instead Of.