

Instituto Tecnológico de Culiacán



Carrera: Ingeniería en Sistemas Computacionales

Materia: Taller de Base de Datos

Alumnos:

Cesar Alfredo Astorga Ochoa

José Alfredo García Aguilar

Trabajo: U4 T2 Triggers

Fecha: 16-Noviembre-2021

Horario de clase: 08:00 – 09:00 am

Profesor: Daniel Esparza Soto

1.- Utilizando trigger, validar que solo se vendan ordenes de lunes a viernes.

```
create trigger tr_ordenes_lunvier
on orders for insert as
declare @fecha date
select @fecha = orderdate from inserted

if (datepart(dw,@fecha) in (1,7))
begin
    rollback tran
    raiserror('Solo permitido de Lunes a Viernes...',16,1)
end
go
--ejecucion
insert orders values(14000, 'HANAR', 8, '05-08-2020 00:00:00.000', '02-01-2020
00:00:00.000', NULL, 2, 10.95, 'MiniMerca', 'Ssss 8', 'Charlie', NULL, 1310, 'Ger')
select * from orders

insert orders values(14000, 'HANAR', 8, '11-20-2020 00:00:00.000', '09-11-2020
00:00:00.000', NULL, 2, 10.95, 'MiniMerca', 'Ssss 8', 'Charlie', NULL, 1310, 'Ger')
select * from orders
```

2.- Validar que no se vendan mas de 20 ordenes por empleado en una semana.

```
create trigger tr_ordenes
on orders for insert as
declare @id_emp int, @primer_fecha int, @segunda_fecha date, @contador int

select @id_emp = employeeid
from inserted
select @segunda_fecha = OrderDate
from inserted
where employeeid = @id_emp
select @primer_fecha = datepart(ww,orderdate)
from orders
where employeeid = @id_emp and OrderDate = @segunda_fecha
select @contador = count(*)
from orders
where employeeid = @id_emp
group by datepart(ww,orderdate), datepart(yy,orderdate)
order by 1 asc

if @contador > 20
begin
    if @primer_fecha > 1
    begin
        rollback tran
        raiserror('Solo se pueden vender hasta 20 ordenes...',16,1)
    end
end
go
```

3.- Validar que el campo firstname en la tabla employees solamente tenga nombres que inicien con vocal.

```
create trigger tr_firstnameVocal
on employees for insert as
declare @nom char(50)

select @nom = firstname from inserted

if @nom not like '[aeiou]%'
begin
    rollback tran
    raiserror('Los nombres solo deben iniciar con vocal...',16,1)
end
go

--ejecucion
insert employees values(25,'Garcia','Alfredo','Manager','Señor.','04-19-1977','16-11-2000','Buenos Aires','Culiacan',NULL,'80199',
'MX',6673879461,667,NULL,NULL,25,NULL)
select * from employees

insert employees values(10,'Astorga','Cesar','Manager','Señor','08-07-1971','05-16-2020','Aguiles Serdan','Culiacan',NULL,'80317',
'MX',6675489034,667,NULL,NULL,44,NULL)
select * from employees
```

4.- validar que el importe de venta de cada orden no sea mayor a \$10,000.

```
create trigger tr_impor
on [order details] for insert as
declare @clave int, @importe numeric(12,2)
select @clave = orderid
from inserted
select @importe = sum(quantity*unitprice)
from [order details]
where orderid = @clave

if @importe > 10000
begin
    rollback tran
    raiserror('El importe no puede ser mayor a $10,000...',16,1)
end
go

insert [order details] values(16519,64,18.00,260,0)
insert [order details] values(16519,60,11.00,700,0)
select * from [order details]

insert [order details] values(16519,58,19.00,110,0)
insert [order details] values(16519,62,21.00,690,0)
select * from [order details]
```

5.- validar que no se puedan eliminar ordenes que se hicieron los lunes.

```
create trigger tr_ordenesL
on orders for delete as
declare @fecha date
select @fecha = orderdate from deleted

if (datepart(dw,@fecha) in (2))
begin
    rollback tran
    raiserror('No es posible eliminar ordenes hechas en dia lunes...',16,1)
end
go

--ejecucion
delete orders where orderid = 10865
select * from orders

delete orders where orderid = 10878
select * from orders
```

6.- Validar que no se realicen inserciones masivas en la tabla products

```
create trigger tr_insercion
on products for insert as
declare @contador int
select @contador = count(*)
from inserted

if @contador>1
begin
    rollback tran
    raiserror('No es posible hacer inserciones masivas en la tabla
Products',16,1)
end

--ejecucion
insert products values(99,'Carlos V',18,2,'18 Piezas',50.90,60.0,17,0)
select * from products

insert products values(99,'Carlos V',18,2,'18 Piezas',50.90,60,0,17,0),(100,'Ferrero
Rocher',18,2,'18 Piezas',28.50,25,0,25,0)
select * from products
```

7.- Validar que no se pueda modificar el campo unitprice de la tabla [order details].

```
create trigger tr_modify
on [order details] for update as
if update (unitprice)
begin
    rollback tran
    raiserror('No es posible modificar el UnitPrice de Order Details',16,1)
end

--ejecucion
```

```

update [order details] set quantity = 7 where orderid = 10248 and productid = 25
select * from [order details]

update [order details] set unitprice = 10.90 where orderid = 10000 and productid =
25
select * from [order details]

```

8.- Validar que solo se pueda actualizar una sola vez el nombre del cliente.

```

alter table customers add contador int
update customers set contador = 0
alter table customers drop column contador

create trigger tr_actualizarNombreCli
on customers for update as
declare @clave nchar(5), @contadorr int

select @clave = customerid
from inserted
select @contadorr = isNull(contador,0)
from inserted

if update (contactname)
begin
if @contadorr>0
begin
rollback tran
raiserror('Solo es posible actualizar una vez el nombre del cliente...',16,1)
end
else
update customers set contador = @contadorr+1 where customerid = @clave
end

--ejecucion
update customers set contactname = 'Alfredo García'
where customerid = 'AROUT'
select * from customers

update customers set contactname = 'Cesar Astorga'
where customerid = 'AROUT'
select * from customers

```

9.- Validar que no se puedan eliminar categorías que tengan una clave impar.

```

create trigger tr_categoriasI
on categories for delete as
declare @clave int
select @clave = categoryid
from deleted

if (@clave % 2 = 1)
begin
rollback tran
raiserror('No se pueden eliminar categorías que tengan una clave
impar...',16,1)

```

```

end
go

--ejecucion
delete categories
where categoryid = 10
select * from categories

delete categories
where categoryid = 13
select * from categories

```

10.- Validar que no se puedan insertar ordenes que se realicen en domingo.

```

create trigger tr_ordenDom
on orders for insert as
declare @fecha date
select @fecha = orderdate from inserted

if (datepart(dw,@fecha) in (1))
begin
    rollback tran
    raiserror('No se pueden insertar ordenes realizadas en domingo...',16,1)
end
go

--Ejecucion
insert orders values(11050,'FOLKO',3,'06-02-2020 00:00:00.000','03-01-2020
00:00:00.000',NULL,4,20.83,'MercaAhorro Supercito','Buenos Aires
54','Cln',NULL,1820,'Mexico')
select * from orders

insert orders values(11050,'FOLKO',3,'01-14-2020 00:00:00.000','07-08-2020
00:00:00.000',NULL,4,20.83,'MercaAhorro Supercito','Buenos Aires
54','Cln',NULL,1820,'Mexico')
select * from orders

```