

Triggers

Disparadores (triggers)

Un "trigger" (disparador o desencadenador) es un tipo de procedimiento almacenado que se ejecuta cuando se intenta modificar los datos de una tabla (o vista).

Se definen para una tabla (o vista) específica.

Se crean para conservar la integridad referencial y la coherencia entre los datos entre distintas tablas.

Si se intenta modificar (agregar, actualizar o eliminar) datos de una tabla en la que se definió un disparador para alguna de estas acciones (inserción, actualización y eliminación), el disparador se ejecuta (se dispara) en forma automática.

Los disparadores, a diferencia de las restricciones "check", pueden ser más complejos y hacer referencia a campos de otras tablas.

Sintaxis básica:

```
create trigger NOMBREDISPARADOR  
on NOMBRETABLA  
for EVENTO que puede ser insert, update o delete  
as  
SENTENCIAS
```

```

--Una librería almacena los datos de sus libros en una tabla
--denominada "libros" y en otra tabla llamada "ventas", las
--ventas de los mismos.
--Eliminamos las tablas si existen:
use pruebasql
go
if object_id('ventas') is not null
    drop table ventas;
if object_id('libros') is not null
    drop table libros;
go
--Creamos las tablas, con las siguientes estructuras:
create table libros(
    codigo int identity,
    titulo varchar(40),
    autor varchar(30),
    precio decimal(6,2),
    stock int,
    constraint PK_libros primary key(codigo)
);
create table ventas(
    numero int identity,
    fecha datetime,
    codigolibro int not null,
    precio decimal (6,2),
    cantidad int,
    constraint PK_ventas primary key(numero),
    constraint FK_ventas_codigolibro
        foreign key (codigolibro) references libros(codigo)
);
go
--Ingresamos algunos registros en "libros":
insert into libros values('Uno', 'Richard Bach', 15, 100);
insert into libros values('Ilusiones', 'Richard Bach', 18, 50);
insert into libros values('El aleph', 'Borges', 25, 200);
insert into libros values('Aprenda PHP', 'Mario Molina', 45, 200);
go

```

Preparamos los datos
para crear trigger

Creamos y probamos el trigger de inserción

```
--Cuando ingresamos un registro en "ventas", debemos
--controlar que exista stock y disminuir el stock en la tabla "libros"
create trigger DIS_ventas_insertar
on ventas
for insert
as
declare @stock int
select @stock= stock from libros
      join inserted
      on inserted.codigolibro=libros.codigo
      where libros.codigo=inserted.codigolibro
if (@stock>=(select cantidad from inserted))
update libros set stock=stock-inserted.cantidad
from libros
join inserted
on inserted.codigolibro=libros.codigo
where codigo=inserted.codigolibro
else
begin
  raiserror ('Hay menos libros en stock de los solicitados para la venta', 16, 1)
  rollback transaction
end
--probamos inserción
insert into ventas values('2007/04/01',1,15,1);
--comprobamos su inserción y el cambio en el stock
select *from ventas;
select *from libros where codigo=1;

--Ingresamos un registro en "ventas", solicitando una
--cantidad superior al stock, generándonos un error.
insert into ventas values('2007/04/01',2,18,100);
--otra inserción
insert into ventas values('2007/01/01',2,18,2);
```

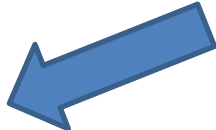



Tabla inserted tiene el conjunto de registros insertados



Rollback transaction para anular la inserción

Trigger de borrado

```
--Creamos un disparador para actualizar el campo "stock" de la tabla
-- "libros" cuando se elimina un registro de la tabla "ventas"
create trigger DIS_ventas_borrar
on ventas
for delete
as
    update libros set stock= libros.stock+deleted.cantidad
    from libros
    join deleted
    on deleted.codigolibro=libros.codigo;


--Eliminamos un registro de "ventas":
delete from ventas where numero=3;
-- Comprobamos stock de libros
select *from libros where codigo=2;
```



Tabla deleted tiene
el conjunto de
registros a borrar

Trigger de actualización

```
--No permitimos modificar el título
create trigger DIS_libros_actualizar_titulo
on libros
for update
as
    if update(titulo)
    begin
        raiserror('El título de un libro no puede modificarse.', 10, 1)
        rollback transaction
    end;
```

 Nos indica si se ha modificado un campo

```
select codigo,titulo,precio from libros
update libros set titulo='Alicia en el pais de las maravillas'
    where codigo=2;
update libros set precio=precio*1.1 where codigo=2;
```

Tenemos disponibles las tablas deleted e inserted con lo que se quita y lo que se añade

Sintaxis para deshabilitar o habilitar un disparador:

```
alter table NOMBRETABLA  
  ENABLE | DISABLE trigger NOMBREDISPARADOR;
```

El siguiente ejemplo deshabilita un trigger:

```
alter table empleados  
  disable trigger dis_empleados_borrar;
```