|  |
| --- |
|  |
| Ejemplo triggers |
|  |
| *1.Integridad\_Compleja.sql* |

Ejemplo triggers

# TRIGGER para garantizar un tipo complejo de unicidad

Imaginemos una tabla Employees y necesitamos almacenar el Nº de carnet de conducir del empleado y como no todos los empleados conducen, esta columna debe permitir valores NULL. Sin embargo, queremos garantizar que en esta columna haya unicidad, ya que dos empleados no pueden tener el mismo número de licencia de conducir.

Veamos las diversas formas de hacer cumplir esta regla:

1- No podemos crear una restricción UNIQUE porque puede haber más de un empleado

sin carnet de conducir. Las restricciones UNIQUE solo aceptan un valor NULL en la columna de la tabla.

2- No podemos crear una restricción CHECK para controlar si ya existe el mismo valor en alguna otra fila de la tabla, porque las restricciones CHECK no pueden acceder a otras filas en la misma tabla.

3- No podemos crear un objeto RULE, porque estos objetos solo controlan los valores en la columna seleccionada y en la fila afectada.

4- Podemos crear un procedimiento almacenado que inserte los datos en la tabla y controlar la unicidad del valor dentro del procedimiento. En este caso necesitamos dos procedimientos:

- uno para insertar datos.

- otro para modificarlos.

Como estos dos procedimientos son lo único que garantiza el cumplimiento de esta regla, debemos asegurarnos de que las aplicaciones cliente solo inserten o modifiquen los datos de los empleados a través de estos procedimientos. Pero los admininistradores pueden modificar las tablas directamente y de ese modo violar esta regla de negocio.

5- Podemos crear un trigger que controle la unicidad del valor que se va a insertar y aceptar o rechazar la entrada. Esta sería la solución más razonable y la que exponemos en el código.

#### Descripción de la tabla Employees:

**Descripción**

Datos personales de empleados

****

#### Modificar la tabla para contener el nuevo campo:

USE Northwind

GO

-- Añadimos una columna llamada NoConductor a la tabla Employees

ALTER TABLE Employees

ADD NoConductor varchar(15) NULL

GO

#### Creación del trigger:

USE Northwind

GO

-- Creamos el trigger para chequear

-- la unicidad en la columna NoConductor

CREATE TRIGGER isrEmployees

ON Employees

FOR INSERT, UPDATE

AS

--Nos aseguramos de que la actualización se ha hecho en el campo

--Nºconductor

IF UPDATE (NoConductor)

--No queremos permitir que haya más de un conductor con el mismo numero de --carnet de conducir, por lo que si existe un numero de conductor igual al

--que queremos modificar no se aceptará el cambio.

IF EXISTS (

SELECT NoConductor, COUNT(\*)

FROM Employees

WHERE NoConductor IS NOT NULL

AND NoConductor IN (

SELECT DISTINCT NoConductor

FROM Inserted

)

GROUP BY NoConductor

HAVING COUNT(\*) > 1

)

BEGIN

RAISERROR ('Nº de carnet de conducir no válido, INSERT o UPDATE cancelada', 16, 1)

ROLLBACK TRAN

END

GO

#### Testear el trigger y sus consecuencias:

-- Este mandato tiene éxito

UPDATE Employees

SET NoConductor = '74914173'

WHERE EmployeeID = 5

-- Este mandato falla

-- El número de carnet no es único

UPDATE Employees

SET NoConductor = '74914173'

WHERE EmployeeID = 6

-- Este mandato falla

-- Se intenta insertar un carnet repetido

UPDATE Employees

SET NoConductor = '74914175'

WHERE EmployeeID BETWEEN 6 AND 10

-- Inicializamos a NULL ya que los valores NULL no los

-- tenemos en consideración en el trigger

UPDATE Employees

SET NoConductor = NULL

#### Deshacer ejemplo:

USE Northwind

GO

ALTER TABLE Employees

DROP COLUMN NoConductor

GO

IF EXISTS (SELECT name FROM sys.objects

WHERE name = 'isrEmployees' AND type = 'TR')

DROP TRIGGER isrEmployees;

GO

# Más información

|  |  |
| --- | --- |
| C/ Miracruz, 10 (Bº de Gros) 20001 Donostia  Telf.: 943 275819  email: [seim@centroseim.com](mailto:seim@centroseim.com) |  |

Logotipo SEIM