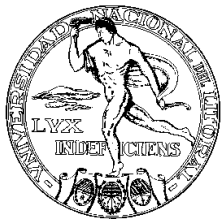


Universidad Nacional del Litoral

Facultad de Ingeniería y Ciencias Hídricas

Departamento de Informática

Bases de Datos

SQL: Guía de Trabajo Nro. 7

Triggers

Parte 2

Msc. Lic. Hugo Minni

2020

Ejercicio 1.

En SQL Server, cree una copia (`Autores`) de la tabla `Authors`. Luego defina un trigger llamado `tr_ejercicio1` asociado al evento `DELETE` sobre la misma. El trigger debe retornar un mensaje (usando PRINT) `"Se eliminaron n filas"` indicando la cantidad de filas afectadas en la operación. Dispare luego la siguiente sentencia SQL para probar el trigger.

```
DELETE
   FROM autores
   WHERE au_id = "172-32-1176" or
         au_id = "213-46-8915"
```

Ejercicio 2.

Implemente en T-SQL un trigger (`tr_ejercicio2`) asociado a la tabla `autores` para inserción y actualización. El trigger debe mostrar un mensaje `'Datos insertados en transaction log'`, y a continuación los datos insertados. Luego `'Datos eliminados en transaction log'` y a continuación los datos eliminados.

Modifique la configuración del menú *Query* a fin de que pueda visualizar la salida de la ejecución como texto. (*Query/Results to/Results to text*)

Luego inserte la siguiente fila y evalúe los resultados:

```
INSERT INTO autores
   SELECT '111-11-1111', 'Lynne', 'Jeff', '415 658-9932',
          'Galvez y Ochoa', 'Berkeley', 'CA', '94705', 1
```

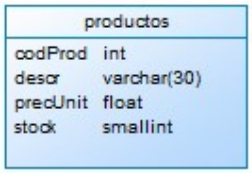
Modifique la fila insertada y evalúe los resultados:

```
UPDATE Autores
   SET au_fname = 'Nicanor' WHERE au_id = '111-11-1111'
```

Ejercicio 3.

Recordemos la tabla Productos, que creamos en la Guía de Trabajo Nro. 2:

```
CREATE TABLE productos
  (
   codProd    int           NOT NULL,
   descr      varchar(30)   NOT NULL,
   precUnit   float         NOT NULL,
   stock      smallint      NOT NULL
  )
```



Implemente un trigger T-SQL (`tr_ejercicio3`) para inserción sobre la misma que, ante la inserción de un producto con stock negativo, dispare el error de aplicación `'El stock debe ser positivo o cero'` con una severidad `12` y contexto `1` y deshaga la transacción. Testee su funcionamiento disparando la siguiente sentencia `INSERT`:

```
INSERT INTO Productos
   VALUES (10, 'Producto 10', 200, -6)
```

Ejercicio 4.

Implemente un trigger T-SQL (`tr_ejercicio4`) que impida insertar publicaciones de editoriales que no hayan vendido por más de `$1500` (tabla `Sales`).

Por ejemplo, La editorial `'1389'` posee un monto de ventas que debería permitir la inserción de sus publicaciones. En cambio, para la editorial `'0736'` seguramente se debería impedir la inserción de publicaciones.

Puede probar el trigger con las siguientes sentencias `INSERT`:

```
INSERT INTO titles
   SELECT 'PC4545', 'Prueba 1', 'trad_cook', '1389',
          14.99, 8000.00, 10, 4095, 'Prueba 1', CURRENT_TIMESTAMP

INSERT INTO titles
   SELECT 'PC4646', 'Prueba 2', 'trad_cook', '0736',
          14.99, 8000.00, 10, 4095, 'Prueba 1', CURRENT_TIMESTAMP
```

Ejercicio 5.

Escriba el mismo trigger como `tr_ejercicio5` en PL/pgSQL.

Ejercicio 6.

En PostgreSQL, agregue dos columnas adicionales a la tabla `Publishers`:
`FechaHoraAlta` está destinada a guardar la fecha y hora en que se da de alta una editorial. `UsuarioAlta` se utilizará para registrar el usuario que realizó la operación de inserción:

```
ALTER TABLE publishers
   ADD COLUMN FechaHoraAlta DATE NULL;

ALTER TABLE publishers
   ADD COLUMN UsuarioAlta VARCHAR(255) NULL;
```

Defina un trigger (`tr_ejercicio6`) que, ante la inserción de una editorial, permita registrar la fecha y hora de la operación (función `CURRENT_TIMESTAMP`) y el usuario que llevó a cabo la operación (función `SESSION_USER`).

Por ejemplo:

```
insert into publishers
   values('8888', 'Editorial Ejercicio 8', 'Boston', 'MA', 'USA');
```

Ejercicio 7.

Tenemos una tabla de registro con la siguiente estructura:

```
CREATE TABLE Registro
   (
    fecha DATE NULL,
    tabla varchar(100) NULL,
    operacion varchar(30) NULL,
    CantFilasAfectadas Integer NULL
   )
```

Se desean registrar en ella algunos movimientos que afectan varias filas, como `UPDATE` y `DELETE`.

Se desea crear un trigger (`tr_Ejercicio7`) para `DELETE` sobre la tabla `Employee` que por cada sentencia `DELETE` **que afecte más de una tupla** genere una entrada en la tabla `Registro`.

Se sugiere realizar una copia de la tabla Employee a fin de realizar las pruebas.

Para realizar las pruebas, elimine los cuatro empleados que existen con job_id 8:

```
Select * from employee where job_id = 8
```

Results | Messages

	emp_id	fname	minit	lname	job_id	job_lvl	pub_id	hire_date
1	ARD36773F	Anabela	R	Domingues	8	100	0877	1993-01-27 00:00:00.000
2	PSP68661F	Paula	S	Parente	8	125	1389	1994-01-19 00:00:00.000
3	M-P91209M	Manuel		Pereira	8	101	9999	1989-01-09 00:00:00.000
4	MMS49649F	Mary	M	Saveley	8	175	0736	1993-06-29 00:00:00.000

¿Cómo escribiría el trigger y en que lenguaje (T-SQL ó PL/pgSQL)?

Ejercicio 8.

Queremos auditar, por cada fila insertada en la tabla `Publishers`, fecha y hora de alta y usuario que realizó la misma (tal como hicimos en el Ejercicio 6). Podemos optar por:

A. Agregar dos columnas adicionales a la tabla `Publishers` (tal como hicimos en el Ejercicio 6) o bien

B. Tener una tabla de log independiente (como la tabla `Registro` del Ejercicio 7).

¿Que configuración de trigger PL/pgSQL elegiría si se adopta la opción A?

¿Que configuración de trigger PL/pgSQL elegiría si se adopta la opción B?