



# **PRÁCTICAS DE SISTEMAS DE BASES DE DATOS**

**GRADO EN INGENIERÍA INFORMÁTICA  
2º CURSO**

## *SESIÓN 4: Integridad. Disparadores*

Ana Belén Gil González  
Ana de Luis Reboredo

Departamento de Informática y Automática  
Universidad de Salamanca



Ana Belén Gil González, Ana de Luis Reboredo, © Marzo 2016  
Location: Salamanca

## 5. INTEGRIDAD. DISPARADORES

### 5.1. Creación de disparadores

**Sintaxis:**

```
CREATE TRIGGER nombre_disparador
{BEFORE | AFTER } { INSERT | UPDATE [OF listacolumnas] |
DELETE}
ON nombre_tabla
[REFERENCING OLD [ ROW | TABLE ] [ AS ] nombre
    [ NEW [ ROW | TABLE ] [ AS ] nombre ] ]
[FOR EACH ROW | FOR EACH STATEMENT]
[WHEN (condición)]
BEGIN
/* Cuerpo del trigger */ ...
END;
```

Un **disparador** es una orden que el sistema ejecuta de manera automática como efecto secundario de la modificación de la base de datos.

Para diseñar un mecanismo disparador, se debe:

- Especificar qué acción sobre la base de datos es la que provocará la ejecución del disparador
- Especificar las condiciones que determinan la ejecución del disparador.
- Especificar las acciones que se van a realizar cuando se ejecute el disparador.

El evento del disparador puede ser `insert`, `delete` o `update`.

La cláusula `on` especifica la relación en la que se especifica la regla.

Las palabras claves opcionales `FOR EACH ROW` especifican que la regla se activará una vez para cada fila que se vea afectada por el evento disparador.

La cláusula opcional `when` se empleará para especificar cualquier condición que haya que verificarse después de que se active la regla pero antes de que se ejecute la acción.

Se puede hacer referencia a los valores de los atributos antes y después de una actualización (notación Oracle)

`new` : para insertar y actualizaciones

`old`: para borrar y actualizaciones

## EJERCICIOS PROPUESTOS

1. Se desea llevar un control de las actualizaciones que se realizan sobre una base de datos que está compuesta por las siguientes tablas:

**PROYECTO** (COD\_PROY, NOMBRE, PRESUPUESTO)

**DEPARTAMENTO** (COD\_DPTO, NOMBRE, DIRECCION, NUM\_EMPLEADOS)

Para ello, se crea una tabla donde se registrará cada acción que se realice sobre las tablas anteriores. Dicha tabla tendrá el siguiente esquema:

**REGISTRO** (ID, FECHA, USUARIO, TABLA, COD\_ITEM, ACCION)

En la tabla REGISTRO se incluirá una tupla por cada acción que se realice en las tablas anteriores y que contendrá los siguientes atributos:

- ID. Será la clave de la tabla registro y se gestionará automáticamente mediante un disparador que obtenga el valor correspondiente a partir de una secuencia.
- Fecha en la que se ha realizado la modificación
- Usuario que ha realizado la acción
- Nombre de la tabla modificada (PROYECTO o DEPARTAMENTO)
- Clave de la tupla insertada, cambiada o borrada
- Acción que se ha realizado (INSERT, UPDATE o DELETE)

a) Crear las tres tablas indicadas y los disparadores necesarios para registrar los datos de modificación de las tablas.

b) Insertar, modificar y borrar varias tuplas en las tablas PROYECTO y DEPARTAMENTO y consultar el contenido de la tabla REGISTRO para comprobar que los disparadores han funcionado correctamente.

2. Crear una nueva tabla EMPLEADO (DNI, NOMBRE, APELLIDO, COD\_DEPTO). Crear los disparadores precisos para que el atributo derivado NUM\_EMPLEADOS de la tabla DEPARTAMENTO se mantenga consistente con el contenido de la tabla EMPLEADOS de modo automático. Comprobar el funcionamiento de los disparadores en los siguientes casos:

- Se insertan varios empleados en distintos departamentos
- Se cambia el departamento al que está asignado un empleado
- Se elimina un usuario
- Se eliminan varios usuarios
- Se inserta un empleado sin departamentos asignado y posteriormente se modifica para asignarlo a un departamento existente
- Se modifica un empleado asignado a un departamento para que deje de estar asignado a ninguno

3. Crear dos tablas con los mismos esquemas de las tablas DISPONE y la tabla PRESTAMO de la base de datos usada en las prácticas (no es necesario definir en ellas las claves externas correspondientes al resto de las tablas de la base de datos de prácticas). Crear los disparadores necesarios para que el atributo derivado NUM\_DISPONIBLES de la tabla creada a imagen de DISPONE se mantenga consistente de manera automática.

Se desea impedir que en la tabla creada a imagen de PRESTAMO se realicen modificaciones sobre las columnas ISBN o COD\_SUC. Crear un disparador que garantice que no se producirán modificaciones de este tipo.

4. La biblioteca desea incentivar los hábitos de lectura de sus socios estableciendo una clasificación de los mismos en función del número de prestamos que han realizado. Solo se incluirán en la clasificación aquellos lectores que hayan realizado como mínimo 10 préstamos. En el caso de que varios lectores coincidan con el mismo nº de prestamos, se les asignarán números consecutivos en la clasificación sin importar el criterio. Para ello, se desea crear una tabla que contenga las siguientes columnas: nº de orden en la clasificación a fecha de hoy, código del lector y nº de prestamos realizados.
  - a. Crear la tabla anterior tomando como clave primaria el nº de orden en la clasificación.
  - b. Crear una secuencia que se utilizará para obtener los valores de la clave primaria de la tabla anterior.
  - c. Crear un trigger que genere de forma automática durante la inserción los valores para la clave de la tabla.
  - d. Rellenar la tabla con los valores correspondientes a partir del contenido de la Base de Datos en el momento actual.
5. Eliminar todos los objetos de la base de datos creados a lo largo de esta sesión.