

## **Actividad de aprendizaje 8-8**

### **Se trata de:**

- Programar y comprobar triggers.

### **Objetivos:**

- Crear y programar triggers.
- Asociar evento y tabla a un trigger.
- Usar los operadores OLD y NEW,
- Comprobar la correcta ejecución de un trigger.
- Eliminar un trigger.
- Listar los triggers asociados una base de datos.

### **Tareas**

Las tareas de esta actividad se realizan sobre la base de datos **ligatercera**.

1.- Realiza un trigger **insertarEquipo** que provoca que, al insertar un nuevo equipo en la tabla **equipos**, lo inserta automáticamente en la tabla **clasificación**.

2.- Realiza un trigger **insertarPartido** que provoca que, al insertar un nuevo partido en la tabla partidos:

- No permite que se carguen valores en goles de equipo local y goles de equipo visitante. Por ello, si se hubieran cargado en una instrucción INSERT, el trigger fuerza a que vuelvan a valor null.

3.- Modifica el trigger **insertarPartido** para añadirle la siguiente funcionalidad:

- Si en la jornada ya hay algún partido que contiene a los equipos del partido a insertar, se debe hacer algo que impida la grabación del partido. No significa esto que haya un partido jugado entre esos dos equipos en la jornada sino que alguno de ellos aparezca ya en algún partido de la jornada. Para que se pueda rechazar la inserción, podemos poner, por ejemplo, el nuevo número de jornada que se guardará a valor nulo. Como en esa columna no se admiten nulos, se rechazará la inserción.

4.- Se supone que los partidos, de acuerdo a lo realizado anteriormente, se cargan con valores null en goles del equipo local y goles del equipo visitante. Para establecer que un partido se inicia, debemos poner valores cero en esas dos columnas de goles. Por ello

deberemos ejecutar una instrucción UPDATE que haga que esas columnas pasen a valer 0 partiendo de un valor anterior null.

Realizar un trigger **partidoEnJuego** que controla las modificaciones realizadas sobre el resultado de un partido en la tabla partidos y desencadena las siguientes acciones:

- Si al hacer la modificación, los valores anteriores de las columnas de goles de los equipos local y visitante son **null**, se pondrán a cero esos goles y, si se produce esta condición:
  1. En la tabla clasificación, se incrementa en 1 el número de partidos jugados por los dos equipos.
  2. En la tabla clasificación, se incrementa en 1 el número de partidos empatados de los dos equipos.
  3. En la tabla clasificación, se incrementan en 1 (debido al empate registrado) los puntos de los dos equipos.

5.- Añade funcionalidad al trigger **partidoEnJuego** anterior para que actualice la clasificación cuando se modifica el resultado desde un resultado anterior no nulo.

Por ejemplo, cuando el resultado era 1-1 y pasa a ser 2-1, habría que hacer lo siguiente:

- El equipo local tiene un partido menos empatado.
- El equipo visitante tiene un partido menos empatado.
- El equipo local tiene un partido más ganado.
- El equipo visitante tiene un partido más perdido.
- Al equipo local se le suma un gol a favor y no se le suman goles en contra. Esto se debe hacer por las diferencias entre los nuevos goles y los anteriores goles.
- Al equipo visitante se le suma un gol en contra y no se le suman goles a favor.
- Se obtienen y cargan los puntos de los equipos local y visitante ( $3*pg+pe$ ).

En definitiva:

- Hay que saber cuál era el resultado del partido y, en función de ello descontar el número de partidos ganados, empatados o perdidos por los equipos.
- Hay que saber cuál es el nuevo resultado del partido y, en función de ello incrementar el número de partidos ganados, empatados o perdidos por los equipos.
- Hay que saber la diferencia entre los nuevos goles y los anteriores del local para sumárselos a favor al local y descontárselos al visitante en la clasificación.
- Hay que saber la diferencia entre los nuevos goles y los anteriores del visitante para sumárselos a favor al visitante y descontárselos al local en la clasificación.