

Práctica 5: Disparadores

Bases de datos

Objetivos

- Programación con PL/SQL avanzado: disparadores (*triggers*).
- Mejorar la docencia con otra encuesta (¡oh, no!).

Enunciado

Considera las siguientes tablas de una base de datos de una pizzería por internet:

pedidos(código, fecha, importe, cliente, notas, especial)

- Esta tabla almacena cada pedido que los clientes hacen por la web.

- Tipos: **char(6), char(10), number(6,2), char(20), char(1024), char(1) ∈ {'S', 'N'}**

contiene(pedido, plato, precio, unidades)

- Esta tabla almacena un código de pedido, cada plato en cada pedido con su precio individual y el número de platos en total. El precio se entiende por cada unidad.

- Tipos: **char(6), char(20), number(6,2), number(2,0)**

auditoría(operación, tabla, fecha, hora)

- Esta tabla almacena el tipo de operación sobre cada tabla, indicando la fecha y hora en que se realizó.

- Tipos: **char(6), char(50), char(10), char(8)**

- *Sin clave primaria. Responde el porqué.*

Añade las restricciones de integridad que consideréis oportunas (integridad referencial, limitación de valores...)

Como resultado de esta práctica se debe subir al CV un **documento PDF con las respuestas a las preguntas planteadas, instrucciones usadas, disparadores y resultados de las ejecuciones** para cada uno de los apartados siguientes:

Apartado 1. Disparador por tabla

- a) Crea las tablas anteriores. Más tarde las rellenarás con los datos que veas necesarios para comprobar el resto de apartados.
- b) Crea y comprueba el funcionamiento de un disparador denominado **tr_pedidos** sobre la tabla **pedidos** de manera que se auditen los cambios producidos por inserciones, borrados y actualizaciones: se incluirá una fila en la tabla **auditoría** con el tipo de operación realizada (**INSERT, UPDATE o DELETE**), el nombre de la tabla (**pedidos**), la fecha y la hora actuales. Para conseguir estos dos últimos datos se usan las funciones **to_char(sysdate, 'dd/mm/yyyy')** y **to_char(sysdate, 'hh:mi:ss')** respectivamente. Este disparador se ejecutará *después* de la actualización (**AFTER INSERT OR DELETE OR UPDATE**) y para la tabla global (no por cada fila). Comprueba el resultado de la ejecución del disparador con varias instrucciones de prueba. Un ejemplo de su resultado podría ser:

OPERACIÓN	TABLA	FECHA	HORA
-----	-----	-----	-----
INSERT	pedidos	06/11/2021	04:27:06
UPDATE	pedidos	06/11/2021	04:28:30
DELETE	pedidos	06/11/2021	04:29:16

Apartado 2. Disparador por fila

Para automatizar los pedidos, crea un disparador llamado **tr_contiene** que se asocie a todas las operaciones posibles de actualización (**INSERT**, **DELETE** y **UPDATE**) sobre la tabla **contiene**, que opere *después* de la modificación (**AFTER**) y por cada fila (**FOR EACH ROW**). Al insertar una nueva fila en esta tabla, se deberá incrementar el valor del campo **importe** de la tabla **pedidos** con el nuevo valor de la tabla **contiene** (**:NEW.precio**) multiplicado por el número de unidades (**:NEW.unidades**). Si se produce la eliminación (**DELETE**) o modificación (**UPDATE**) de una fila, el importe se debe ajustar según la modificación introducida (restando al antiguo valor **:OLD.precio** el nuevo, o restando el antiguo y sumando el nuevo a donde corresponda, respectivamente). Ten en cuenta que también puede cambiar el código de pedido al modificar con **UPDATE**. Comprueba los resultados de la ejecución del disparador con varias instrucciones de prueba.

Apartado 3. Disparador ¿mutante?

Escribe un disparador denominado **tr_especial** que actualice el campo **especial** de la tabla **pedidos** cada vez que se inserte (con una instrucción **INSERT**) o modifique un nuevo pedido (con una instrucción **UPDATE**). Al insertar consideramos que se proporcionan todos los valores salvo el del campo especial, que será responsabilidad de este disparador proporcionarle un valor. El pedido se considera especial (campo **especial='S'**) si el importe supera el importe promedio de todos los pedidos (incluido él mismo); en caso contrario no es especial ((campo **especial='N'**)). Escribe cómo podría ser con granularidad **FOR EACH ROW**, inserta algunas filas y comprueba el resultado. Actualiza ahora el importe de una de las filas y describe lo que ocurre. Si encuentras problemas al implementarlo así, ¿podrías resolverlo de otro modo? (Ayuda: lee la transparencia sobre temporalidad de eventos).

Apartado 4. Encuesta DESweb

Al igual que la encuesta Docentia, esta encuesta es anónima (así que critica todo lo que quieras, no habrá represalias :-). Cada alumno debe rellenar su encuesta. Accede desde el CV en la pestaña General (este es el [enlace](#)). En el documento pdf debéis indicar como respuesta: Realizada o No realizada.