

Práctica 6: Disparadores e índices

Bases de datos

Objetivos

- Programación con PL/SQL avanzado: disparadores (*triggers*).
- Creación y uso de índices para mejorar el rendimiento.

Enunciado

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

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

- Almacena cada pedido que los clientes hacen por la web.

```
CREATE TABLE pedidos(código char(6) primary key, fecha char(10),  
importe number(6,2), cliente char(20), notas char(1024));
```

contiene(pedido, plato, precio, unidades)

- 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.

```
CREATE TABLE contiene(pedido char(6), plato char(20), precio  
number(6,2), unidades number(2,0), primary key (pedido, plato));
```

auditoría(operación, tabla, fecha, hora)

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

```
CREATE TABLE auditoría(operación char(6), tabla char(50), fecha  
char(10), hora char(8));
```

resumenClientes(cliente, numPedidos, sumaImportes)

- Almacena en número de pedidos y suma de sus importes de los clientes que tienen pedidos

```
CREATE TABLE resumenClientes(cliente char(20), numPedidos integer,  
sumaImportes number(8,2));
```

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

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 **trigger_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. 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). Para determinar cuál es la operación en el cuerpo del disparador se usan las comprobaciones **IF INSERTING THEN** o **IF DELETING THEN** o **IF UPDATING THEN**. 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	06/01/2020	04:27:06
UPDATE	pedidos	06/01/2020	04:28:30
DELETE	pedidos	06/01/2020	04:29:16

Apartado 2. Disparador por fila

Para automatizar los pedidos, crea un disparador llamado **trigger_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 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, respectivamente). Comprueba los resultados de la ejecución del disparador con varias instrucciones de prueba.

Apartado 3. Disparador en cadena

Crea un disparador sobre la tabla de pedidos que mantenga actualizados los datos de la tabla resumenClientes cuando se modifique la tabla de pedidos.

La tabla resumen clientes debe tener siempre actualizado el número de pedidos y suma de sus importes de los clientes que tengan pedidos.

Si el cliente de un nuevo pedido no está en la tabla resumenClientes dicho cliente debe añadirse.

Si un cliente deja de tener pedidos se debe borrar su fila correspondiente en resumenClientes.