Apuntes PL-SQL

Triggers I

Triggers

- ¿Qué son?

  Bloques de código PL-SQL que se ejecutan automáticamente cuando se produce un evento.

- Tipos de triggers.
  - Aplicación: Asociados a un evento del interfaz de usuario (clic de un botón, pérdida de foco…). Se usan en Oracle Developer.
  - Datos: Asociados a una modificación de los datos que hay en las tablas: INSERT, UPDATE o DELETE.
  - Sistema: Asociados a un evento del sistema. (Login, Error del Servidor, Arranque de la BD, etc…).

Creación de triggers de datos

- Momento: BEFORE o AFTER o INSTEAD OF.
- Evento: INSERT ó UPDATE ó DELETE
- Nombre Tabla: On NombreTabla
- Tipo de Disparo: Por fila o por sentencia.
- Condición de disparo: Claúsula WHEN.
  (sólo en triggers por fila).
- Cuerpo del trigger:
  DECLARE
  BEGIN
  END;

Elementos de un trigger

- Momento: ¿Cuándo se va a disparar?

  BEFORE: Antes de la instrucción DML, indicado si el trigger sirve para comprobar si la operación DML debe hacerse o no.

  AFTER: Si queremos que la sentencia se ejecute antes que el trigger y además no se prevee que el trigger vaya a hacer fallar la sentencia.

  INSTEAD OF: Solo para modificar vistas que no son actualizables directamente.

Elementos de un trigger

- Evento: ¿Por qué se ejecuta el trigger?

- INSERT
- UPDATE (de cualquier columna o de algunas en concreto).
- DELETE
- Combinación de las anteriores.

Elementos de un trigger

- Tipo de Disparo: ¿Cuántas veces se va a disparar cuando se produce el evento?

  – Por sentencia: Se ejecuta una sola vez, incluso aunque no haya filas afectadas.

    Se usan si la acción del trigger no depende de los valores de los datos afectados por la operación.

  – Por fila: Se ejecuta tantas veces como filas se vean afectadas por la operación. Se usan si la acción del trigger depende de los datos afectados.

Elementos del trigger

- Cuerpo del trigger: ¿Qué acción va a realizar el trigger?

  El cuerpo del trigger es un bloque PL/SQL:

  ```
  DECLARE
  BEGIN
  EXCEPTION
  END;
  ```

  Puede contener variables, cursores, excepciones…

  Si el trigger es de fila se permite acceder al valor de los datos afectados por la orden DML empleando :old y :new.

Orden de disparo

- Supongamos cuatro triggers definidos sobre la tabla emp llamados:

  BeforeStatement

  BeforeRow

  AfterRow

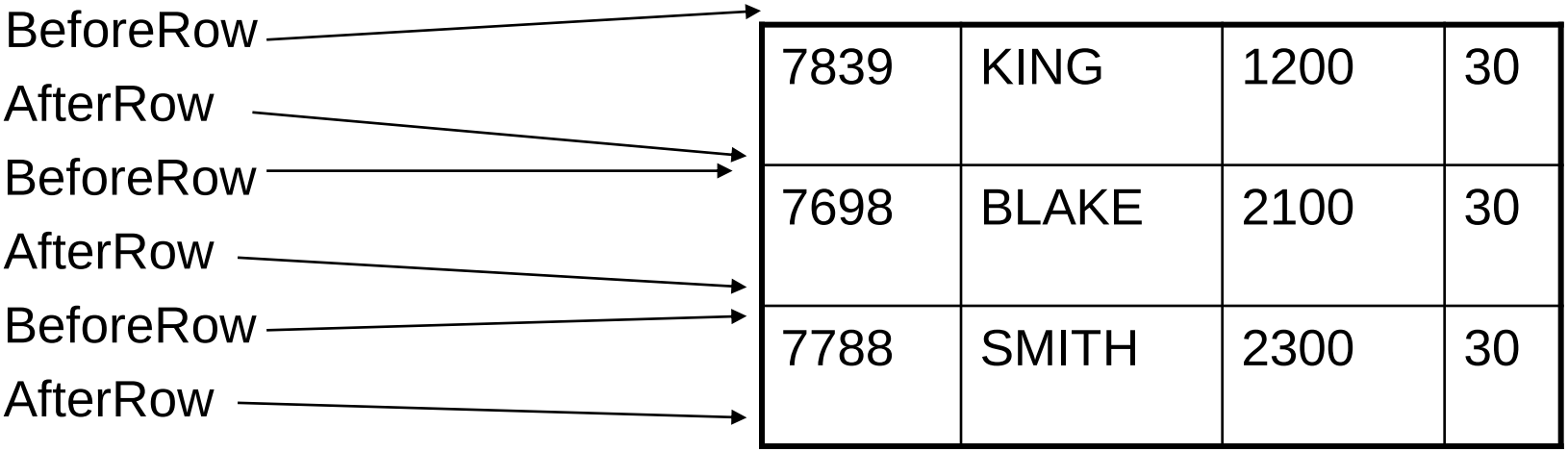
  AfterStatement

  y una instrucción DML que afecta a tres registros, por ejemplo:

```
UPDATE emp
SET sal = sal *2
WHERE deptno=30;
```

Orden de disparo

BeforeStatement



AfterStatement

Uso de predicados condicionales

- Cuando un trigger puede ser disparado por varias operaciones distintas, hay una forma de saber dentro del código del trigger cual de ellas lo disparó, se llaman predicados condicionales y se usan así:

    - IF INSERTING THEN…
    - IF UPDATING [(‘nombreColumna’)] THEN…
    - IF DELETING THEN…

Ejemplos de Triggers

- Supongamos que queremos hacer un trigger que impida insertar datos en la tabla emp fuera del horario normal de oficina.

Veamos que tipo de trigger es:

- Momento: BEFORE, ya que es un trigger cuyo objetivo es comprobar si la operación debe hacerse o no.

- Tipo de Disparo: Por sentencia, ya que el contenido de los datos no es relevante.

Ejemplos de triggers

- El código del trigger anterior sería:

```
CREATE OR REPLACE TRIGGER SeguridadEmp
BEFORE INSERT ON emp
BEGIN
   IF (TO_CHAR(sysdate, 'DY') IN ('SAT', 'SUN') OR
       TO_CHAR(sysdate,'HH24) NOT BETWEEN '08' AND '15' THEN
           RAISE_APPLICATION_ERROR( -20100, 'No puedes insertar
           registros fuera del horario normal de oficina');
   END IF;
END;
/
```

- Nota: Cuando un trigger termina con excepciones no tratadas, la sentencia que lo disparó hace un ROLLBACK automáticamente.

Uso de :old y :new

- Solo para triggers por fila.
- Sirven para referirse al valor anterior y posterior a una modificación.

- INSERT: solo existe :new
- DELETE: solo existe :old
- UPDATE: existen :old y :new

- Para usarlos en una claúsula WHEN se le quitan los dos puntos.

Restricciones en el uso de triggers

- No pueden ejecutarse instrucciones DDL.
- No pueden ejecutarse instrucciones de control de transacciones (COMMIT, ROLLBACK, …)
- Por sentencia: No tiene sentido el uso de :old y :new.
- Por fila: No puedo consultar los datos de la tabla que ha disparado el trigger, esto es, no puedo hace una SELECT de esa tabla (problema de tablas mutantes).

Ejemplos de triggers

- Supongamos que queremos hacer un trigger que no permita a un empleado ganar más de 5000 si no es el presidente (restricción de integridad de datos compleja).

Veamos que tipo de trigger es:

- Momento: BEFORE, ya que es un trigger cuyo objetivo es comprobar si la operación debe permitirse o no.

- Tipo de Disparo: Por fila, ya que el contenido de los datos es necesario.

- Evento: Se disparará cuando haya un INSERT o un UPDATE del sueldo o un UPDATE del oficio.

Ejemplos de triggers

- El código del trigger anterior sería:

```
CREATE OR REPLACE TRIGGER ControlSueldo
BEFORE INSERT OR UPDATE ON EMP
FOR EACH ROW

BEGIN

    IF :new.sal>5000 AND :new.job!='PRESIDENT' THEN
            RAISE_APPLICATION_ERROR(-20100, 'No puede ganar
                                                tanto si no es presidente');
    END IF;

END;
/
```

Triggers INSTEAD OF

- Se utilizan para modificar datos en vistas que no son actualizables directamente porque les falta algún campo requerido en las tablas en que se basa la vista.
- Se ejecutan en lugar de la sentencia que provoca su disparo.
- Ver ejemplo de la página 254 del libro del martillo.

Gestión de triggers

- Cuando se crea un trigger, se activa automáticamente. Puede ser interesante desactivarlo para que no interfiera en la fase de pruebas de otros triggers.

- Para desactivar un trigger:

  ALTER TRIGGER NombreTrigger DISABLE;

- Para activar un trigger:

  ALTER TRIGGER NombreTrigger ENABLE;

- Para activar o desactivar todos los triggers asociados a una tabla:

  ALTER TABLE NombreTabla ENABLE o DISABLE ALL TRIGGERS;

- Para borrar un trigger definitivamente:

  DROP TRIGGER NombreTrigger;

Triggers de sistema

- Pueden funcionar para un esquema (o usuario) concreto o para toda la base de datos (todos los usuarios).
- Los eventos que se pueden considerar son:
  - CREATE, ALTER o DROP
  - Conexión y desconexión
  - Arranque y parada de la base de datos.
  - Cuando se produzca un error concreto o un error cualquiera.

Sintáxis de triggers de sistema

Evento DDL (creación, modificación o borrado de un objeto):

```
CREATE OR REPLACE TRIGGER NombreTrigger
[BEFORE | AFTER]
[CREATE|ALTER|DROP] OR [CREATE|ALTER|DROP] …
ON [DATABASE | SCHEMA]

DECLARE
BEGIN
EXCEPTION
END;
```

Sintáxis de triggers de sistema

Eventos del sistema:

CREATE OR REPLACE TRIGGER NombreTrigger
[eventosys] OR [eventosys] … ON [DATABASE | SCHEMA]
DECLARE
BEGIN
EXCEPTION
END;

eventosys puede ser uno de los siguientes:

- AFTER SERVERERROR
- AFTER LOGON
- BEFORE LOGOFF
- AFTER STARTUP
- BEFORE SHUTDOWN

Ejemplo de trigger de sistema

Supongamos que queremos guardar en una tabla las conexiones y desconexiones de la base de datos de un usuario concreto para saber cuanto tiempo ha estado conectado a la base de datos.

```
CREATE OR REPLACE TRIGGER ControlEntrada
AFTER LOGON ON SCHEMA
BEGIN
    INSERT INTO UsoBD (cod_user, hora, accion)
    VALUES (USER, SYSDATE, 'Entrada');
END;

CREATE OR REPLACE TRIGGER ControlSalida
BEFORE LOGOFF ON SCHEMA
BEGIN
    INSERT INTO UsoBD (cod_user, hora, accion)
    VALUES (USER, SYSDATE, 'Salida');
END;
```

Usos más frecuentes de los triggers

- Seguridad.
- Auditorías.
- Integridad de datos.
- Integridad referencial.
- Replicación de tablas.
- Cálculo de datos derivados.
- Control de Eventos.

Triggers de seguridad

- Los permisos de usuario permiten modificar o no las tablas a un usuario concreto, pero con triggers podemos hacer que ciertos valores de los datos solo puedan ser introducidos por ciertos usuarios.
- Otro ejemplo de trigger de seguridad es el que solo permite trabajar a unas horas determinadas.

Triggers de auditoría

- Son aquellos triggers que permiten guardar información sobre los usuarios que realizan determinadas operaciones en la base de datos, incluyendo los valores introducidos en dicha operación, cuando las realizan, etc...
- Hemos visto varios ejemplos de triggers de auditoría.

Triggers de integridad de datos

- Como hemos visto, en ocasiones hay restricciones en los valores de los datos que no se pueden implementar con un simple CHECK en la creación de la tabla.

- Algunos ejemplos que hemos visto ya:

    - Sólo el presidente puede ganar más de 5000.
    - Sólo los vendedores pueden tener comisiones.

Triggers de integridad referencial

- Ya sabemos que ORACLE mantiene la integridad referencial definiendo FOREIGN KEYs y la opción ON DELETE CASCADE.

- Así, por ejemplo, no podemos introducir en el campo deptno de la tabla emp un valor que no exista en la tabla dept o borrar un departamento sin borrar los empleados asociados, pero no existe la posibilidad de actualizar en cascada si no se emplea un trigger para ello.

Triggers de integridad referencial

- Veamos un trigger de actualización en cascada:

```
CREATE OR REPLACE TRIGGER OnUpdateCascade
AFTER UPDATE OF deptno ON dept
FOR EACH ROW
BEGIN
   UPDATE emp
   SET emp.deptno = :new.deptno
   WHERE emp.deptno = :old.deptno
END;
/
```

Triggers de replicación de tablas

- Los triggers de replicación se usan para mantener copias remotas en tiempo real de ciertas tablas.

- Simplemente consisten en realizar en las réplicas las mismas operaciones DML que se van haciendo en las tablas maestras.

- Las réplicas se pueden hacer en la misma base de datos o en otra base de datos si usamos un enlace de base de datos.

Triggers de cálculo de datos derivados

- En ocasiones, para evitar que se realice repetidamente un mismo cálculo a partir de los datos, es conveniente introducir un cierto grado de redundancia en la base de datos.

- Un ejemplo de esto sería la creación de una columna total_salarios en la tabla dept donde se guarde la suma de los salarios de todos los empleados de cada departamento.

- El problema de introducir redundancia es la posible inconsistencia de los datos, para asegurar esta consistencia se emplean triggers de cálculo de datos derivados.

Triggers de cálculo de datos derivados

- Vamos a ver como se haría el trigger para mantener la consistencia, usando por primera vez una llamada a un procedimiento desde un trigger:

```
CREATE OR REPLACE TRIGGER MantenerTotalSalarios
AFTER INSERT OR UPDATE OF SAL OR DELETE ON emp
FOR EACH ROW
BEGIN
    IF DELETING THEN actualizartotal (:old.deptno, -1 * :old.sal);
    ELSIF UPDATING THEN actualizartotal(:new.deptno, :new.sal-:old.sal);
    ELSE – insercion
            actualizartotal(:new.deptno, :new.sal);
    END IF;
END;
/
```

Triggers de cálculo de datos derivados

- Finalmente, vamos a ver el código del procedimiento actualizartotal:

```
CREATE OR REPLACE actualizartotal
(v_deptno IN dept.deptno%TYPE, v_sal IN dept.totalsalarios%TYPE)
IS
BEGIN
    UPDATE dept
    SET totalsalarios = NVL(totalsalarios, 0) + v_sal
    WHERE deptno = v_deptno;
END;
/
```

Triggers de control de eventos

- Un ejemplo típico de trigger de control de eventos es el que se diseña en una base de datos de almacén para que se generen pedidos automáticamente para aquellos productos cuyo stock esté por debajo de un umbral de existencias predeterminado.

  En este caso, el evento es la bajada de las existencias por debajo del umbral aceptable.

- Este problema será resuelto en una de las prácticas que vais a llevar a cabo.