Gestión y Modelación de Datos

Sesión Practica: Triggers

MARÍA CONSTANZA PABÓN

MCPABON@JAVERIANACALI.EDU.CO

PL – Triggers (disparadores)

- Un ***trigger*** permite definir un programa que el motor de base de datos **ejecuta implicitamente** cuando ocurre un **evento DML** en los datos
  - Eventos DML: insert, update, delete

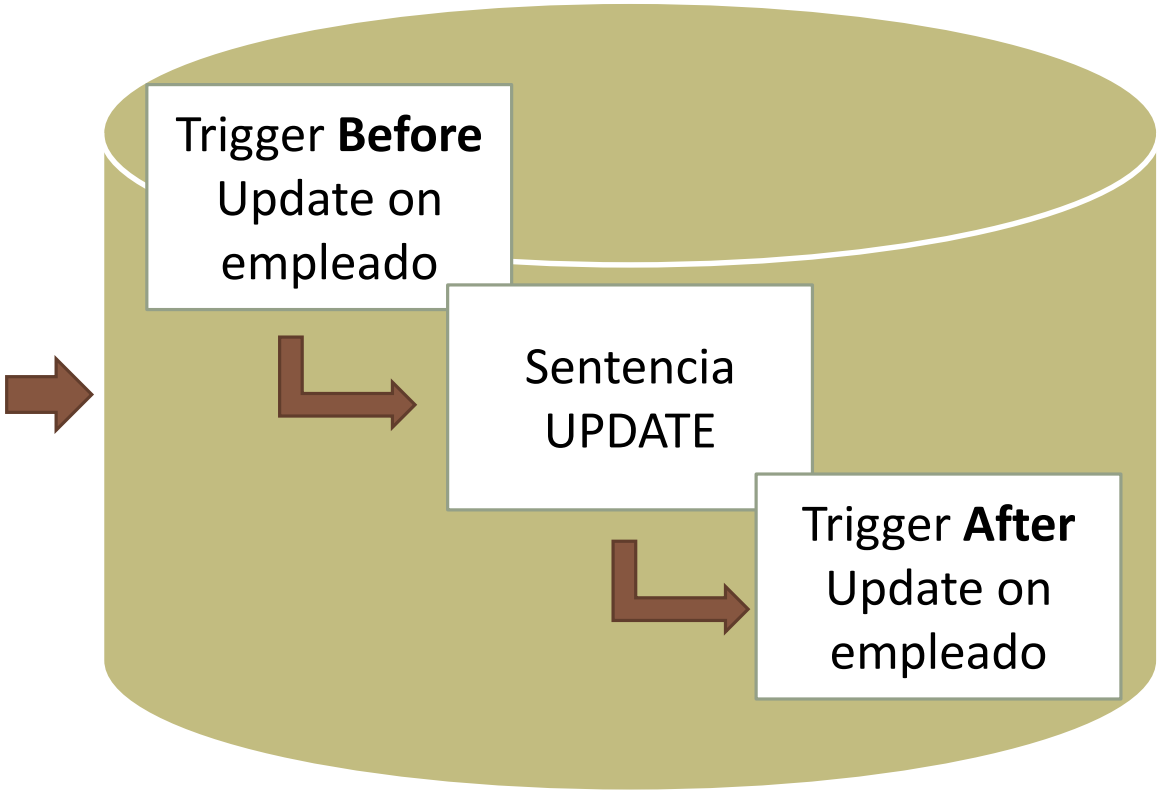
La declaración de un trigger incluye

- Un **evento**: **operación** que se efectúa sobre **una** tabla
  - **INSERT**, **UPDATE**, **DELETE**
- Un **momento**: cuándo se dispara el trigger
  - ***BEFORE***: el trigger se ejecuta antes de que se ejecute el evento
  - ***AFTER*** : el trigger se ejecuta después de que se ejecute el evento

Secuencia de ejecución

Dados los triggers before y after update en la tabla empleado:

```
UPDATE
   Empleado
SET
   salario =
      salario *1,05;
```



Usos

- Prevenir transacciones inválidas
  - Verificar datos
- Generar valores de atributos derivados
- Reforzar seguridad
  - Ej. Permitir ciertas operaciones solamente en horario laboral
- Implementar reglas de negocio complejas
- Temas de auditoria
  - Ej. Registrar usuario y hora (timestamp) de cada operación
- Sincronizar réplicas

Ejemplo

- Si se tiene una tabla **totalSalarios** que tiene un atributo que guarda la suma de los salarios de los empleados, el siguiente trigger mantiene la tabla actualizada:

```
CREATE OR REPLACE TRIGGER trCambioSalarios
AFTER INSERT OR UPDATE OR DELETE ON Empleado
BEGIN
    UPDATE totalSalarios
    SET    total = (SELECT SUM(salario) FROM empleado);
END;
```

Ejemplo

- Una vez creado el trigger

```
CREATE OR REPLACE TRIGGER trTotalSalarios
AFTER INSERT OR UPDATE OR DELETE ON Empleado
BEGIN
   UPDATE totalSalarios
   SET    total = (SELECT SUM(salario) FROM empleado);
END;
```

SELECT * FROM totalSalarios;

total
100.000.000

INSERT INTO Empleado (cc, nombre, salario) VALUES (1001, 'Juan Robles', 1000000);

SELECT * FROM totalSalarios;

total
101.000.000

Sintaxis

```
CREATE [OR REPLACE] TRIGGER nombre_trigger
momento evento ON tabla
[FOR EACH ROW [WHEN condición]]
bloque de PL/SQL
-- El bloque comienza con DECLARE o BEGIN
```

Tipo de trigger

- El **número de veces** que se ejecuta el cuerpo del trigger (programa)
  - **FOR EACH ROW**: el cuerpo del trigger se ejecuta por cada registro que se procesa
  - **Trigger de operación**: el cuerpo del trigger se ejecuta una sola vez

Trigger de fila (FOR EACH ROW)

- Pueden acceder a la información de la fila que se va a procesar con las variables **:OLD** (valor anterior del registro) y **:NEW** (nuevo valor del registro)

Evento	:OLD	:NEW
INSERT	NULL	El registro que se inserta
DELETE	El registro que se borra	NULL
UPDATE	El registro antes de la actualizarse	El registro después de actualizarse

- **:OLD** y **:NEW** son registros del mismo tipo del registro de la tabla para la cual se declara el trigger

Cuando un trigger BEFORE genera un error la operación no se ejecuta

```
CREATE OR REPLACE TRIGGER trValor
BEFORE INSERT ON Empleado FOR EACH ROW
DECLARE minSal NUMBER(10) DEFAULT 0;
BEGIN
   SELECT MIN(salario) INTO minSal FROM empleado;
   IF :NEW.salario < minSal THEN
      Raise_application_error(-20000, 'Salario fuera de
                              rango: ' ||:NEW.salario);
   END IF;
END;
```

Cuando un trigger BEFORE genera un error la operación no se ejecuta

```
CREATE OR REPLACE TRIGGER trValor
 BEFORE INSERT ON Empleado FOR EACH ROW
 DECLARE minSal NUMBER(10) DEFAULT 0;
 BEGIN
    SELECT MIN(salario) INTO minSal FROM empleado;
    IF :NEW.salario < minSal THEN
       Raise_application_error(-20000, 'Salario fuera de rango: ' ||:NEW.salario);
    END IF;
 END;
```

INSERT INTO Empleado (cc, nombre, salario) VALUES (1002, 'Mary Diaz', 150000);

```
Error que empieza en la línea: 1 del comando :
INSERT INTO Empleado (cc, nombre, salario)  VALUES (1002, 'Mary Diaz', 150000)
Informe de error -
ORA-20000: Salario fuera de rango:150000
ORA-06512: at "BD19.TRVALOR", line 5
ORA-04088: error during execution of trigger 'BD19.TRVALOR'
```

En un trigger BEFORE de fila, se pueden cambiar los valores que se van a guardar en la base de datos

- Si se agrega el atributo usuario para registrar el usuario que hizo los cambios en los datos:

```
CREATE OR REPLACE TRIGGER trUser
BEFORE INSERT OR UPDATE ON Empleado
FOR EACH ROW
BEGIN
    SELECT USER INTO :NEW.Usuario FROM DUAL;
END;
```

En un trigger BEFORE de fila, se pueden cambiar los valores que se van a guardar en la base de datos

```
CREATE OR REPLACE TRIGGER trUser
BEFORE INSERT OR UPDATE ON Empleado
FOR EACH ROW
BEGIN
    SELECT USER INTO :NEW.Usuario FROM DUAL;
END;
```

SELECT * FROM empleado WHERE cc = 234;

cc	nombre	salario	usuario
234	José	2000000	(null)

UPDATE empleado SET nombre = 'José M.' WHERE cc = 234;

SELECT * FROM empleado WHERE cc = 234;

cc	nombre	salario	usuario
234	José M.	2000000	BD19

Usando :OLD

```
CREATE OR REPLACE TRIGGER trUser
BEFORE UPDATE ON Empleado
FOR EACH ROW
BEGIN
    IF :NEW.salario < :OLD.salario THEN
       Raise_application_error(-20100, 'El nuevo
       salario ('||:NEW.salario||') debe ser mayor
       que el anterior ('||:OLD.salario||')');
    END IF;
END;
```

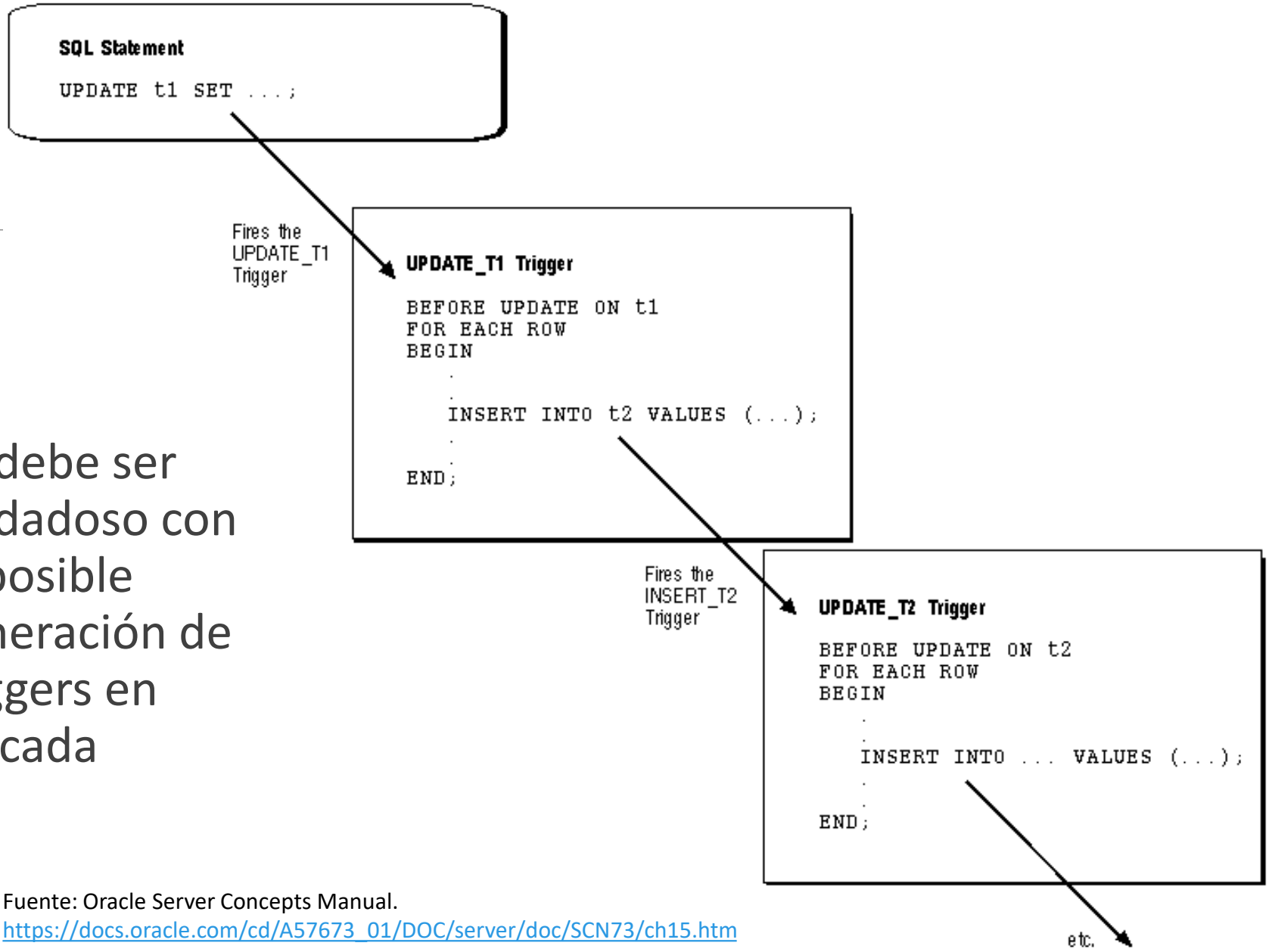
Comportamientos diferentes para insert, delete y update

- Predicados condicionales INSERTING, DELETING, UPDATING

```
CREATE OR REPLACE TRIGGER trEjemplo BEFORE INSERT OR DELETE OR
UPDATE ON Empleado FOR EACH ROW
BEGIN
   CASE
      WHEN INSERTING THEN
        …
      WHEN UPDATING THEN
        …
      WHEN DELETING THEN
        …
   END CASE;
END;
```

Modelo de ejecución de una sentencia en la que se aplican triggers

1. Ejecutar todos los disparadores BEFORE de tipo operación que aplican
2. Para cada fila afectada por la sentencia SQL:
   a) Ejecutar todos los disparadores BEFORE de fila que aplican
   b) Bloquear y cambiar la fila, y realizar la comprobación de las restricciones de integridad
   c) Ejecutar todos los activadores AFTER de fila que aplican
3. Ejecutar todos los activadores AFTER de tipo operación que aplican



- Se debe ser cuidadoso con la posible generación de triggers en cascada

Fuente: Oracle Server Concepts Manual.
https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch15.htm

Referencias

Manual de Oracle

En línea:
https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/triggers.htm#LNPLS020

https://docs.oracle.com/cd/B28359_01/server.111/b28318/triggers.htm#CNCPT1672