

# Oracle Tablas Mutantes

# Oracle Tablas Mutantes

**Tabla mutante es una tabla que está siendo modificada por una sentencia SQL (insert, update, delete) o por el efecto de un DELETE CASCADE asociado a la sentencia SQL.**

**Restricciones sobre tablas mutantes:**

- **la acción de un regla de tipo FOR EACH ROW no puede consultar ni actualizar una tabla mutante para un evento.**
- **la acción de una regla de tipo FOR EACH STATEMENT activada como efecto de un DELETE CASCADE no puede consultar ni actualizar una tabla mutante para su evento.**

## Oracle Tablas Mutantes Solución

**Sea una regla R con evento E sobre la tabla T, en cuya acción se desea consultar o actualizar T (T es mutante para el evento de la regla R, entonces:**

- 1- Definir sobre la tabla T para el evento E una regla R1 de tipo FOR EACH ROW que copie en una estructura auxiliar A las tuplas actualizadas por el evento E. La estructura auxiliar A puede ser una variable estructurada type, local a la sesión de trabajo, o una tabla auxiliar de la BD.**
- 2- Definir sobre la tabla T para el evento E una regla R1\_1 de tipo FOR EACH STATEMENT que procese los elemento de A, haciendo las correspondientes consultas en T (T no es mutante para el evento de la regla R1\_1).**

## Ejemplo Oracle Tablas Mutantes

**Ejemplo : “cada producto no puede tener más de 10 presentaciones”**

**Punto de ejecución: AFTER**

**Qué eventos lo tienen que disparar:**

**INSERT OR UPDATE OF codigo\_p ON presentacion**

**Condición: sin condición**

**Granularidad: FOR EACH ROW**

**Acción:**

```
SELECT COUNT(*)
INTO    cant
FROM    presentacion
WHERE   codigo_p = :NEW.codigo_p; ERROR !!!!!
IF cant > 10 THEN
    RAISE_APPLICATION_ERROR(-20111, ' texto de error...');
END IF;
```

# Solución para Tablas Mutantes Oracle

```
CREATE TRIGGER control_present
AFTER INSERT OR UPDATE OF codigo_p ON presentacion
FOR EACH ROW
BEGIN
INSERT INTO tmp_presentacion VALUES (:NEW.codigo_p, :NEW.id_presentacion)
END;
```

```
CREATE TRIGGER control_present_1
AFTER INSERT OR UPDATE OF codigo_p ON presentacion
DECLARE Y NUMBER; X NUMBER;
CURSOR C IS SELECT DISTINCT codigo_p FROM tmp_presentacion
BEGIN
  OPEN C; FETCH C INTO Y,
  WHILE C%found LOOP
    SELECT COUNT(*) INTO X FROM presentacion WHERE codigo_p = Y;
    IF cant > 10 THEN RAISE_APPLICATION_ERROR(-20111, 'texto error..');
    END IF;
    FETCH C INTO Y
  END LOOP;
  DELETE FROM tmp_presentacion;
END;
```