PL / SQL
Les déclencheurs

Ines BAKLOUTI

ines.baklouti@esprit.tn

Ecole Supérieure Privée d'Ingénierie et de Technologies

Plan

Introduction

- Un déclencheur ou un trigger est un bloc PL/SQL nommé et stocké qui se lance automatiquement lorsqu'un événement se produit.
- Un événement est toute modification de données se trouvant dans les tables ou les vues: instruction LMD (INSERT, UPDATE, DELETE).
- Un déclencheur est très utile pour contrôler ou appliquer des contraintes qu'il est impossible de les formuler de facon déclarative.
- Les informations sur les triggers sont stockées dans la vue: USER_TRIGGERS.

Plan

Création de déclencheurs

Syntaxe

```
CREATE [OR REPLACE] TRIGGER nom_trigger
{BEFORE|AFTER|INSTEAD OF}
{INSERT|UPDATE [ OF nom_colonne [, nom_colonneN]] |DELETE}
OR {INSERT|UPDATE [OF nom_colonne [, nom_colonneN ]] |DELETE}
REFERENCING {[OLD [AS] ancien] | [NEW [AS] nouveau]}
ON nom_table
[FOR EACH ROW]
[WHEN] (condition)
DECLARE
/* déclaration */
BEGIN
/* traitement */
   [EXCEPTION]
END;
```

Notez Bien

La création d'un trigger avec le mot clé INSTEAD OF est réservée pour les triggers sur des vues créees avec l'option WITH READ ONLY ou des vues créees sur plus qu'une table.

Plan

Types de déclencheurs

- Il existe deux types de triggers différents:
  - les triggers de table (STATEMENT): sont exécutés une seule fois lorsque des modifications surviennent sur une table (même si ces modifications concernent plusieurs lignes de la table).
  - Les triggers de lignes (ROW): sont exécutés « séparément » pour chaque ligne modifiée dans la table. Ils sont très utiles s'il faut mesurer une évolution pour certaines valeurs, effectuer des opérations pour chaque ligne en question.

Types de déclencheurs

- Il est possible de contrôler les conditions d'éxecution avec des clauses telles que:

  IF INSERTING THEN
  ...
  ELSIF UPDATING THEN
  ...
  ELSIF DELETING THEN
  ...

- Activer/Désactiver un trigger par son nom
  - ALTER TRIGGER nomTrigger ENABLE;
  - ALTER TRIGGER nomTrigger DISABLE;
- Supprimer un trigger
  - Drop trigger nom_Trigger

Déclencheur de table

Exemple 1

```
– création de table
CREATE TABLE log (table_name varchar2(30), date_ins date, action varchar2(100));
– création de trigger
CREATE TRIGGER TRIG_LOG
AFTER INSERT OR UPDATE
ON Employees
BEGIN
    IF INSERTING THEN
       INSERT INTO log(table_name, date_ins, action) VALUES ('Employees', sysdate, 'INSERT');
    ELSE
       INSERT INTO log(table_name, date_ins, action) VALUES ('Employees', sysdate, 'UPDATE');
    END IF;
END;
– test
UPDATE employees set first_name='Wiliam' where employee_id=153;
– vérification de l'insertion dans la table
SELECT * FROM log;
```

Déclencheur de table

Exemple 2

```
CREATE OR REPLACE TRIGGER trig_check
BEFORE INSERT OR UPDATE OF salary, department_id
OR DELETE ON employees
BEGIN
CASE
   WHEN INSERTING THEN
      DBMS_OUTPUT.PUT_LINE('Inserting');
   WHEN UPDATING('salary') THEN
      DBMS_OUTPUT.PUT_LINE('Updating salary');
   WHEN UPDATING('department_id') THEN
      DBMS_OUTPUT.PUT_LINE('Updating department ID');
   WHEN DELETING THEN
      DBMS_OUTPUT.PUT_LINE('Deleting');
END CASE;
END;
```

Déclencheur de ligne

- Dans un trigger ligne, on peut accéder aux anciennes et nouvelles valeurs de colonnes de la ligne. Les noms permettent de désigner ces deux valeurs sont:
  - :NEW contient les nouvelles valeurs
  - :OLD contient les anciennes valeurs

	:NEW	:OLD
INSERT	nouvelle valeur	null
UPDATE	nouvelle valeur	ancienne valeur
DELETE	null	ancienne valeur

Remarque

Un trigger crée avec le mot clé INSTEAD OF (trigger sur vue) peut lire les valeurs de :NEW et :OLD mais ne peut pas les changer. On peut concevoir un déclencheur INSTEAD OF pour déterminer quelle opération prévue et faire les opérations LMD appropriées sur les tables correspondantes.

Déclencheur de ligne

Exemple 1

```
CREATE OR REPLACE TRIGGER trig_dep
BEFORE INSERT OR UPDATE OR DELETE
ON DEPARTMENTS
FOR EACH ROW
BEGIN
   IF INSERTING THEN
      dbms_output.put_line('Insertion');
   END IF;
   IF UPDATING THEN
      dbms_output.put_line('Modification');
   END IF;
   IF DELETING THEN
      dbms_output.put_line('Suppression');
   END IF;
END;
-- test
UPDATE departments set department_name = lower(substr(department_name,1,1))
||substr(department_name,2,length(department_name)-1);
```

Déclencheur de ligne

Exemple 2

```
CREATE OR REPLACE TRIGGER trig_insert
BEFORE INSERT
ON departments
FOR EACH ROW
WHEN (new.department_id < 270)
BEGIN
:new.department_id := :new.department_id+10;
END;
```