SMI-S5

SGBD ORACLE : Déclencheurs

Pr. Issam QAFFOU

Laboratoire Ingénierie des Systèmes d'Information
Département d'Informatique
FSSM-UCA

C’est quoi un déclencheur?

- On les appelle en anglais: triggers
- Un déclencheur est un bloc de PL/SQL qui est **invoqué** implicitement à chaque fois qu'un **événement** particulier se produit dans la base de données.
- Un tel événement est lié à une manipulation particulière des données d'une table telles que l'insertion, la suppression ou la modification (INSERT, UPDATE ou DELETE) d'une ligne d'une table.
- Les déclencheurs ont un nom, une partie déclaration, un corps et une partie exception.
- Les déclencheurs n’acceptent pas d’arguments et ne peuvent pas être tirés explicitement.
- Les déclencheurs s’exécutent automatiquement par Oracle d’une manière transparente à l’utilisateur.

Utilité

- On peut utiliser les déclencheurs:
  - Contraintes d'intégrité complexes qui ne peuvent être implémentées comme des contraintes déclaratives d'une table.
  - Audit: Par exemple, pour garder une trace des modifications apportées à une table.
  - Pour effectuer automatiquement une action lorsqu'une autre action aura lieu. Par exemple, la mise à jour d'une table à chaque fois qu'il y a une insertion d'une ligne dans une autre table.
  - Le maintien des valeurs dérivées facilement; par exemple, un système d'achat d'actions ne doit jamais réellement changer la quantité en stock. On la maintient avec un déclencheur.
  - Système d'archivage - les données archivées peuvent être obtenues très facilement avec des déclencheurs.

Types de triggers

- Parler des types de triggers, c'est dire quand est ce qu'ils se déclenchent.
- Il y a 12 types divisés en 2 catégories:
  - Catégorie de ceux qui sont tirés pour répondre à une instruction DML.
  - Catégorie de ceux qui sont tirés pour toutes les lignes affectées par une instruction DML.
- Ils peuvent aussi être divisés:
  - Déclencheur de **synchronisation**:
    - BEFORE: tirés avant l'exécution de l'instruction.
    - AFTER: tirés après l'exécution de l'instruction.
  - Déclencheur **d'évènement**:
    - INSERT, UPDATE et DELETE: pour déterminer le type de l'instruction qui tire le déclencheur.

Créer un trigger

- On utilise la commande CREATE TRIGGER pour créer un déclencheur d'une base de données.
- On utilise aussi:
  - Nom du trigger,
  - La table associée,
  - Quand un déclencheur est tiré: before ou after
  - La commande qui invoque le déclencheur: update, delete ou insert
  - Condition de filtrage
  - Le bloc PL/SQL qui sera exécuté quand le déclencheur s'est tiré.

Création

- Syntaxe

```
CREATE [OR REPLACE] TRIGGER nom_trigger
{BEFORE | AFTER}
{DELETE | INSERT | UPDATE [OF colonnes]}
        [OR {DELETE | INSERT |UPDATE [OF colonnes]}]...
ON nom_table
[FOR EACH ROW [WHEN condition]]
[REFERENCING [OLD AS old] [NEW AS new]]
PL/SQL block
```

Créer un trigger

- CREATE OR REPLACE

  Recréer le déclencheur s'il existe déjà.

- BEFORE | AFTER

  Avant ou après l'événement déclenchant le trigger, qui peut être une insertion, suppression ou mise à jour (INSERT | DELETE | UPDATE) sur une table.

- L'option FOR EACH ROW [WHEN (condition)] fait exécuter le trigger à chaque modification d'une ligne de la table spécifiée (on dit que le trigger est de "niveau-ligne").

  En l'absence de cette option, le trigger est exécuté une seule fois ("niveau table").

- When est utilisé pour ne déclencher le trigger que si la condition est vérifiée.

Créer un trigger

- L'option OF permet de spécifier les colonnes concernées pour déclencher le trigger.
- <corps du trigger> représente un bloc de code à exécuter au déclenchement d'un trigger.
- Les noms de corrélation: NEW, OLD
  - Les noms de corrélation par défaut sont NEW pour de nouvelles valeurs et OLD pour des anciennes valeurs de la ligne.

    Ainsi, afin **d'accéder aux valeurs d'une nouvelle ligne** on utilise NEW et pour **accéder aux valeurs d'une ligne déjà existante** on utilise OLD.
  - Il est possible de changer les noms de corrélation en utilisant REFERENCING

Créer un trigger

- **Exemple**: on crée un trigger qui maintient une fiche de stock basée sur une table « transaction ». Ce trigger doit satisfaire ces conditions:
  - On a une table transaction qui enregistre des commandes et des reçus.
  - On a une table « stock » qui contient les données relatives aux chiffres des stocks actuels.
  - On veut maintenir automatiquement la quantité de stock actuelle basée sur les transactions uniquement du type ISS (Commande) et RCT (reçu).
  - Tester et déboguer des déclencheurs est un peu difficile, parce qu'on ne peut rien afficher directement à partir d'un déclencheur, on doit l'exécuter vers une autre table ou un fichier externe.
  - Pour vraiment déclencher un trigger on doit générer l'événement déclencheur avec une commande DML saisie directement dans SQL*PLUS.

Créer un trigger

- Exemple (suite)

```
CREATE OR REPLACE TRIGGER stock_quantity
        BEFORE INSERT
        ON transactions
        FOR EACH ROW
        WHEN new.transaction_type IN ('ISS','RCT')
DECLARE
        l_use_quantity stock.quantity%TYPE;
BEGIN
   IF :new.transaction_type = 'ISS'
   THEN
      l_use_quantity := :new.quantity - 1;
   ELSE
      l_use_quantity := :new_quantity;
   END IF;
      UPDATE stock
      SET quantity = quantity + l_use_quantity
      WHERE item = :new.item;
END;
```

Créer un trigger

- Le trigger de l'exemple précédent sera tiré avant qu'une ligne ne soit insérée dans la table *transaction*, mais quand le type de *transaction* est soit ISS ou RCT.
- Si la transaction est une commande alors la quantité est mise négative pour s'assurer que le stock a été réduit.
- La quantité dans la table des valeurs est mise à jour.

Le trigger instead of

- Ces triggers sont définis pour des vues.
- Ils sont utilisés pour modifier des vues qui ne peuvent pas être modifiées par des commandes DML.
- Au contraire des triggers normaux qui sont déclenchés lors de l'exécution d'une commande DML, ceux-ci se déclenchent au lieu d'exécuter ces commandes.
- Exemple: soit la vue NewEtudiant créée à partir de Etudiant et Payement.

```
create view NewEtudiant
as
select s.rollno, nom, bcode, sexe, total
from Etudiant s, Payement p
where s.rollno = p.rollno;
```

Le trigger instead of

Si on veut insérer une ligne dans la table NewEtudiant, Oracle retourne une erreur.

```
SQL> insert into NewEtudiant values (15,'Joe','b2','m',2000);
ERROR at line 1:
ORA-01779: cannot modify a column which maps to a non key-preserved table
```

Il est possible de l'insérer dans les tables Etudiant et Payement en utilisant:

```
create or replace trigger NewEtudiant _it_bi_row
instead of insert
on NewEtudiant
for each row
begin
        -- insérer dans Etudiant d'abord
        insert into Etudiant(rollno,bcode,nom,sexe,dj)
        values(:new.rollno, :new.bcode, :new.nom, :new.sexe,sysdate);
        -- insérer une ligne dans PAYEMENT
        insert into payments values(:new.rollno, sysdate, :new.total);
end;
```

Trigger actif ou inactif

- Un trigger peut être soit activé ou désactivé.
- Désactiver un trigger peut améliorer la performance lorsqu'une grande quantité de données d'une table doit être modifiée.
- Par exemple: UPDATE Etudiant set nom = upper(nom);

  S'exécutera plus vite si tous les déclencheurs tirés par UPDATE sur la table Etudiant sont désactivés.
- Pour activer ou désactiver un trigger on utilise alter trigger.
  - **alter trigger nom_trigger disable;**
  - **alter trigger nom_trigger enable;**
- Pour désactiver tous les triggers d'une table:
  - **alter table nom_table disable all triggers;**
- Pour supprimer un trigger:
  - **drop trigger nom_trigger ;**