



Triggers

SQL avancé – Semaine
d'intégration 2019

- Un trigger est un bloc de PL/SQL associé à une vue ou une table, qui s'exécutera lorsqu'une instruction de manipulation de données (LMD) sera exécutée.

Avantages

- Le code est centralisé dans la bdd
- Le code est déclenché quelque soit l'outil qui effectue des manipulations de données sur la base

Inconvénients

- Son exécution utilise des ressources pouvant augmenter le temps des autres traitements (ex : modifications massives)
- Ne peut pas contenir de COMMIT ou ROLLBACK ou toute autre instruction SQL issue de la définition des données (LDD)

Exécution du trigger

- Le bloc PL/SQL constituant le trigger peut être exécuté avant ou après les contraintes d'intégrité.
- Choix du moment d'exécution à l'aide des mots clés :
 - BEFORE
 - AFTER
 - INSTEAD OF

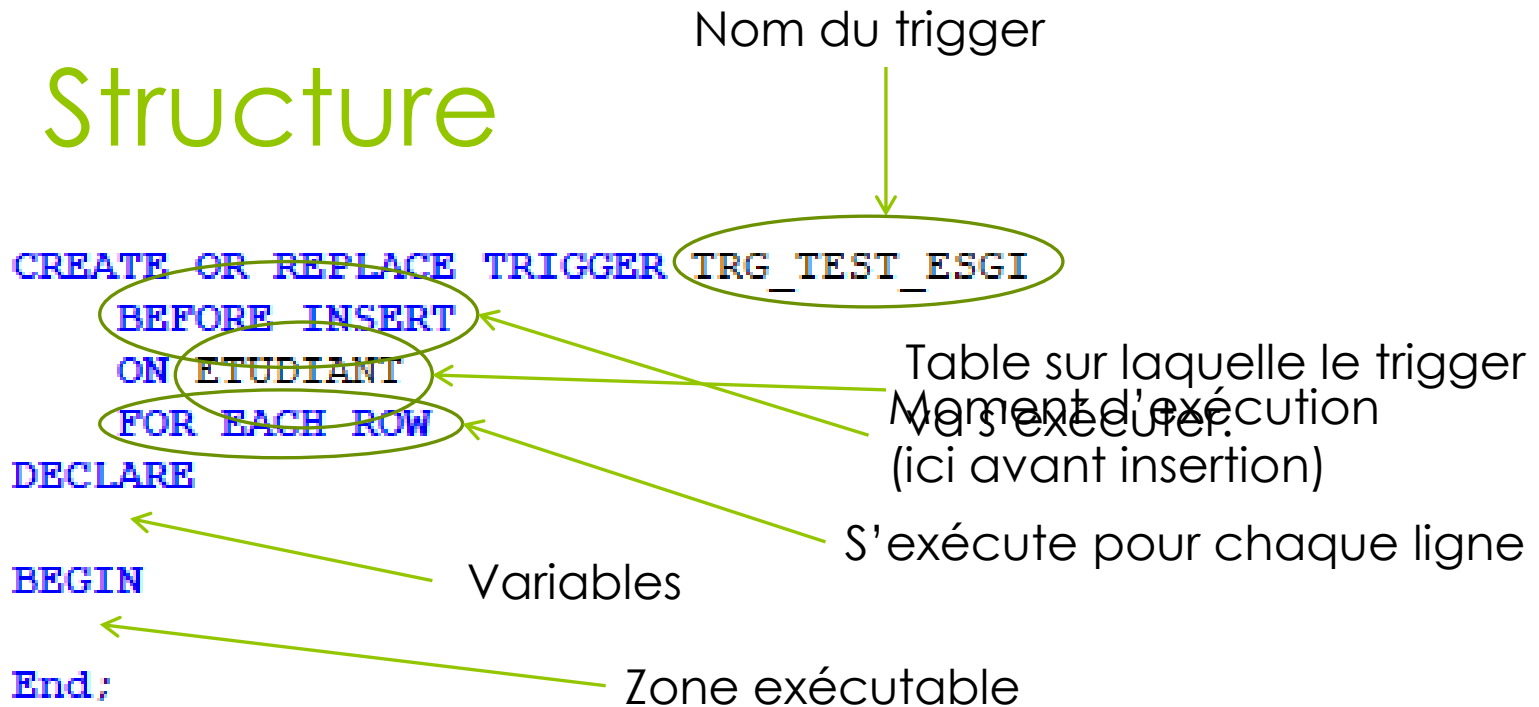
Manipulation des données

- Seules les colonnes de la ligne en cours de modification sont accessibles.
- Toujours préfixer les noms de colonnes de :
- Utilisation de deux variables prédéfinies :
 - OLD : valeur avant modification
 - NEW : valeur après modification
- Les deux variables sont utilisables dans la clause WHEN et dans la section exécutable

Remarques

- Utilisation : new.colonne ou old.colonne
- OLD n'est renseigné que pour DELETE et UPDATE. (Aucune utilité pour INSERT)
- NEW n'est renseigné que pour INSERT et UPDATE. (Aucune utilité pour DELETE)

Structure



Activer / Désactiver

- ALTER TRIGGER nom DISABLE
- ALTER TRIGGER nom ENABLE

- ALTER TABLE nom_table DISABLE ALL TRIGGERS
- ALTER TABLE nom_table ENABLE ALL TRIGGERS

Supprimer un trigger

- DROP TRIGGER nom

Variables

- Les variables locales sont définies dans la partie DECLARE
 - Variables de types ORACLE

```
DECLARE
    nom          VARCHAR2(10);
    quantite     NUMBER;
    datjour      DATE;
    prix         NUMBER(7,2);
BEGIN
    ...
END;
```

Variables

- Variables faisant référence au dictionnaire de données
 - TABLE.colonne%TYPE

```
DECLARE
    nom      emp.nom%TYPE; -- nom, de même type que colonne nom de emp
BEGIN

...

END;
```

Variables

- Initialisation à l'aide de `:=`

```
DECLARE
    nom          CHAR(10)    := 'HUGUES';
    quantite     NUMBER      := 15;
    datjour      DATE        := '20/05/02';
    reponse      BOOLEAN     := TRUE;
BEGIN
    ...
END;
```

Conditions

```
IF          condition1 THEN          traitement1;  
ELSIF      condition2 THEN          traitement2;  
ELSE      traitement3;  
END IF;
```

IF / THEN / END IF obligatoires