

Cours Bases de données 2ème année IUT

Cours 5 : PL/SQL : ou comment faire plus avec ORACLE

3ème partie

Anne Vilnat

<http://www.limsi.fr/Individu/anne/cours>

Plan

- 1 Curseurs et mise à jour
- 2 Paquetages
 - Définition
 - Syntaxe
 - Exemple
 - Utilisation d'un paquetage
 - Le paquetage DBMS_OUTPUT
- 3 Déclencheurs
 - Définition
 - Syntaxe
 - Différents types : table, ligne
 - Exemples

Curseurs et mise à jour

Employe
Matricule Nom Prenom Salaire Prime ...

Un curseur : pour récupérer des lignes qu'on peut vouloir modifier.

Exemple

On veut augmenter de 10% la prime des employés gagnant moins de 3000 €, et de 20% celle des employés gagnant moins de 500 €.

Curseurs et mise à jour

Solution : on met dans un curseur les employés gagnant moins de 3000€, et suivant les cas on augmente leur prime de 10% ou 20%.

Exemple de mise à jour via un curseur

```
DECLARE
  CURSOR inc_prime IS
    SELECT *
      FROM employe
     WHERE emp_sal < 3000;
BEGIN
  .....
```

Curseurs et mise à jour

Exemple de mise à jour via un curseur (suite... pas simple!)

```
DECLARE
....
BEGIN
  FOR r_structure IN inc_prime LOOP
    IF r_structure.prime < 500 THEN
      UPDATE employe
        SET prime = prime * 1.2
        WHERE matricule = r_structure.matricule;
    ELSE UPDATE employe
        SET prime = prime * 1.1
        WHERE matricule = r_structure.matricule;
    END IF;
  END LOOP;
END;
```

curseurs et mise à jour : plus simple!

En déclarant le curseur FOR UPDATE

```
CURSOR inc_prime IS  
  SELECT * FROM employe  
    WHERE emp_sal < 3000;  
  FOR UPDATE;
```

en utilisant la clause CURRENT OF

```
IF r_structure.prime < 500 THEN  
  UPDATE employe  
    SET prime = prime * 1.2  
    WHERE CURRENT OF inc_prime;  
ELSE UPDATE employe  
    SET prime = prime * 1.1  
    WHERE CURRENT OF inc_prime;  
END IF;
```

Plan (suite)

- 1 Curseurs et mise à jour
- 2 Paquetages
 - Définition
 - Syntaxe
 - Exemple
 - Utilisation d'un paquetage
 - Le paquetage DBMS_OUTPUT
- 3 Déclencheurs
 - Définition
 - Syntaxe
 - Différents types : table, ligne
 - Exemples

Définition

Définition

Un paquetage est un groupe d'objets PL/SQL comme :

- des types
- des constantes,
- des variables,
- des procédures,
- des fonctions,
- des curseurs,
- des exceptions,
- des traitements d'exceptions.

Un paquetage a un nom,

Il forme un ensemble cohérent sur un sujet

Constitution d'un paquetage

2 parties

Un paquetage comporte :

- une partie spécification : les objets visibles de l'extérieur (PUBLIC),
- un corps (BODY) qui définit les objets déclarés dans la spécification et ceux non visibles à l'extérieur (PRIVATE).

Avantages

Avantages

- modifier le corps sans recompiler la spécification, ni changer les programmes qui font appel aux objets du package,
- surcharger,
- définir des variables globales et des curseurs communs à toutes les procédures ou fonctions du package,
- gérer plus efficacement les privilèges à accorder.

Syntaxe de la partie spécification

Création d'un paquetage

```
CREATE [ OR REPLACE ] PACKAGE nomPackage  
[IS | AS ]  
– Déclaration  
END [ nomPackage ];
```

[Déclarations] → objets du paquetage
Procédures et fonctions : [prototype](#)

Syntaxe de la partie corps

Création d'un paquetage

```
CREATE [ OR REPLACE ] PACKAGE BODY nomPackage
[IS | AS ]
[ Déclaration objets privés ]
[ Définition des procédures et des fonctions ]
[ BEGIN
    Instructions
[ EXCEPTION
    bloc exception
END [ nomPackage ];
```

→ **Corps** des procédures et les fonctions dont les **prototypes** sont dans la spécification

Instructions : bloc spécifique, exécuté au premier appel d'un composant du paquetage dans une session

Exceptions : levées lors de l'initialisation du paquetage

Exemple : spécification

Création d'un paquetage

```
CREATE OR REPLACE PACKAGE Statistiques
AS
  FUNCTION PlusDeNFilms (nbFilms NUMBER)
    RETURN NUMBER;
  FUNCTION nbFilmsActeur (
    numActeur Individu.numIndividu%TYPE)
    RETURN NUMBER;
END Statistiques ;
```

Similaire à C++ ou ADA

Exemple : corps

Création d'un paquetage

```
CREATE OR REPLACE PACKAGE Statistiques AS
  FUNCTION PlusDeNFilms (nbFilms NUMBER)
    RETURN NUMBER AS
      -- partie déclarations
    BEGIN
      -- Corps de la fonction
    EXCEPTION
      -- Traitement des exceptions
  END PlusDeNFilms;
  FUNCTION nbFilmsActeur(numAct Individu.numInd%TYPE)
    RETURN NUMBER AS
      -- Partie declaration
    BEGIN
      -- Corps de la fonction
    EXCEPTION
      -- Traitement des exceptions
  END nbFilmsActeur;
END statistiques
```

Suppression

Suppression d'un paquetage entier

```
DROP PACKAGE nomPackage;
```

Suppression du corps d'un paquetage

```
DROP PACKAGE BODY nomPackage;
```

Suppression du corps uniquement.

Utiliser un paquetage

Compilation

Un paquetage est compilé à la création (comme procédure...), avec éventuellement des erreurs dans les fonctions et/ou procédures à corriger

Utilisation

`[proprietaire].nomPackage.nomProcédure[(liste paramètres effectifs)]`

`[proprietaire].nomPackage.nomFonction[(liste paramètres effectifs)]`

Exemple de paquetage : DBMS_OUTPUT

Description

Paquetage fourni par Oracle, avec les procédures publiques :

- `ENABLE` (taille `IN INTEGER DEFAULT 2000`) : débloque l'affichage du buffer sur la console, à faire en premier
- `PUT (A { VARCHAR2 | NUMBER | DATE })` : met la valeur dans le tampon
- `NEW_LINE` : provoque l'écriture de la ligne sur la sortie standard
- `PUT_LINE (A { VARCHAR2 | NUMBER | DATE })` : provoque l'écriture de la ligne constituée sur la sortie standard
- `DISABLE` : bloque l'affichage
- ...

Plan (suite)

- 1 Curseurs et mise à jour
- 2 Paquetages
 - Définition
 - Syntaxe
 - Exemple
 - Utilisation d'un paquetage
 - Le paquetage DBMS_OUTPUT
- 3 Déclencheurs
 - Définition
 - Syntaxe
 - Différents types : table, ligne
 - Exemples

Déclencheurs : définition

Définition

Un déclencheur définit une action qui doit s'exécuter quand survient un événement dans la base de données.

Il peut servir à :

- ajouter des contraintes sur les valeurs des attributs d'une table,
- enregistrer des changements (suivi)
- ajouter des règles de gestion,...

Déclencheurs : définition

Quand et comment les utiliser

Un déclencheur peut être déclenché :

- à la création, suppression ou modification d'un objet (**CREATE**, **DROP**, **ALTER**),
- *à la connexion, déconnexion d'un utilisateur,*
- *démarrage ou arrêt d'une instance...*

Durée d'activité

L'action associée à un déclencheur est un bloc PL/SQL enregistré dans la base.

Un déclencheur est opérationnel jusqu'à la suppression de la table à laquelle il est lié.

Syntaxe de la création d'un déclencheur

Création

```
CREATE [OR REPLACE] TRIGGER [schema.]nomTrigger  
    { BEFORE | AFTER | INSTEAD OF } listeEvenement  
    ON nomTable  
    [FOR EACH ROW]  
    [ WHEN condition]  
    corpsTrigger
```

listeEvenement : liste d'événements séparés par une virgule.

événement : **DELETE**, **INSERT** ou **UPDATE**.

Si **UPDATE** on peut préciser les attributs concernés (**UPDATE OF listeAttributs**).

Trigger de niveau ligne ou de niveau table

Granularité

Le traitement mentionné dans un trigger peut s'effectuer :

- pour chaque ligne concernée par l'événement, (trigger de niveau **ligne**)
- une seule fois pour l'ensemble des lignes concernées par l'événement (trigger de niveau **table**).

Ligne

FOR EACH ROW → trigger de niveau ligne, sinon de niveau table.
WHEN condition : pour chaque ligne, trigger déclenché si vraie.

Quand et quoi...

Quand

BEFORE | AFTER

- pour trigger de niveau table : déclenché avant ou après l'événement
- pour trigger de niveau ligne : exécuté avant ou après la modification de **CHAQUE** ligne concernée

INSTEAD OF : spécifique aux vues.

Quoi

corpsTrigger : bloc PL/SQL effectué quand le trigger est déclenché

En plus :

IF INSERTING THEN ... END IF;

IF DELETING THEN ... END IF;

IF UPDATING THEN ... END IF;

Valeurs des attributs

Quelles valeurs sont testées?

Dans la clause WHERE ou dans le corps, on peut se référer à la valeur d'un attribut avant ou après que soit effectuée la modification déclenchant le trigger :

- :OLD.nomAttribut : la valeur avant la transaction UPDATE ou DELETE
- :NEW.nomAttribut : la valeur après la transaction UPDATE ou INSERT

Tables exemple

Une table contenant des étudiants (numero, nom, prénom...) avec leur moyenne.

Des tables reportEtudiant (pour le suivi), Alerte.→ voir plus loin!

Table etudiant

NumEtudiant	Nom	Prenom	Moyenne
123	DUPONT	JULES	4
234	DUPOND	ALFRED	5
567	DURAND	JULIE	14
598	DURANT	ALFREDINE	16

Exemple de déclencheur de niveau table

Niveau table

```
CREATE TRIGGER triggerEtudiant
  BEFORE INSERT, UPDATE ON etudiant
BEGIN
  DBMS_OUTPUT.ENABLE(10000);
  IF INSERTING THEN
    IF USER != 'VILNAT'
      THEN RAISE_APPLICATION_ERROR
        (-20001,'Utilisateur non autorisé.');
```

END IF;

END IF;

IF UPDATING THEN

DBMS_OUTPUT.PUT_LINE('Mise a jour de étudiant.');

END IF;

END;

Exemple de déclencheur de niveau ligne

Niveau ligne

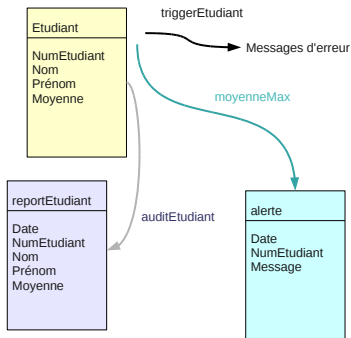
```
CREATE TRIGGER auditEtudiant
  AFTER UPDATE OR DELETE ON etudiant
  FOR EACH ROW
BEGIN
  INSERT INTO reportEtudiant
    VALUES (SYSDATE, :OLD.NumEtud, :OLD.nom,
            :OLD.prenom, :OLD:moyenneEtud);
END;
```

Exemple de déclencheur de niveau ligne avec clause WHERE

Niveau ligne

```
CREATE TRIGGER moyenneMax
  AFTER UPDATE ON etudiant
  FOR EACH ROW
  WHEN :NEW.moyenneEtud > 2*:OLD.moyenneEtud
BEGIN
  INSERT INTO alerte(datealerte, numEtudiant, message)
    VALUES ( SYSDATE, :OLD.nom,
              'MOYENNE a plus que doublé');
END;
```

Déclencheurs : exemples



Exemple

Les triggers qui vont “surveiller” les mises à jour sur la table Etudiant, et soit afficher des messages d’erreur en cas de problème, soit alimenter d’autres tables...

Ordre d'activation :

Si plusieurs triggers sur une même table, l'ordre d'activation est :

- 1 BEFORE niveau table
- 2 BEFORE niveau ligne, aussi souvent que de lignes concernées
- 3 AFTER niveau ligne, aussi souvent que de lignes concernées
- 4 AFTER niveau table