

Initiation au PL/SQL

Par [Arnaud PHILIPPS](#) • Publié le 22/10/2015 à 16:48:29 • Noter cet article: ★★★★★ (9 votes)
Avis favorable du comité de lecture



Arnaud PHILIPPS

Introduction

Cet article a pour but de vous expliquer la structure du PL/SQL, une interface procédurale au système de gestion de base de données relationnelle Oracle. Il permet de combiner des requêtes SQL et des instructions procédurales (boucles, conditions) dans le but de créer des traitements complexes ...



Bonne lecture !

Structure

PL/SQL signifie Procedural Language / SQL. L'unité de base en PL/SQL est un bloc, tous les programmes PL/SQL sont constitués de blocs, pouvant être imbriqués les uns dans les autres. Un bloc a la structure suivante :

```
1  DECLARE
2
3      /* Section declarative : variables, types et sous programmes */
4
5  BEGIN
6
7      /* Section executable : déclaration de procédure */
8
9  EXCEPTION
10
11     /* Section de gestion des exceptions: gestion des erreurs */
12
13  END;
```

Seule la section exécutable est nécessaire. Les autres sections sont facultatives. Les seules instructions admises dans un programme PL / SQL sont SELECT, INSERT, UPDATE, DELETE et plusieurs autres déclarations de manipulation de données. Les instructions de définition de données comme CREATE, DROP ou ALTER ne sont pas autorisées. La section exécutable contient également des constructions telles que des affectations, des boucles, des appels de procédure. La PL/SQL n'est pas sensible à la casse et les commentaires sont similaires à ceux du C (/ * ... * /).

Variables et Types

Les informations sont transmises entre le programme PL/SQL et la base de données par le biais de variables. Chaque variable a un type spécifique qui lui est associé. Ce type peut être :

- L'un des types utilisés par SQL pour les colonnes de base de données.
- Un type générique utilisé en PL/SQL comme NUMBER.



- [Introduction](#)
- [Structure](#)
- [Variables et Types](#)
- [Boucle](#)
- [Curseur](#)
- [Procédure](#)
- [Conclusion](#)

exemple :

```
1 DECLARE
2
3     quantite  NUMBER;
4
5     biere     VARCHAR(20);
6
```

Notez que PL / SQL permet d'utiliser des variables BOOLEAN, même si Oracle ne supporte pas BOOLEAN comme un type pour les colonnes.

Le type en PL/SQL peut être délicat. Si il y a une incompatibilité de type, les affectations et les comparaisons de variables peuvent ne pas fonctionner comme vous le pensiez. Pour être sûr, au lieu de coder en dur le type d'une variable, vous pouvez utiliser %TYPE.

Exemple :

```
1 DECLARE
2
3     biere     Biere.name%TYPE;
4
```

Donne à la variable biere le type qui a été déclaré pour la colonne Biere.

La valeur initiale d'une variable, quel que soit son type, est NULL. Nous pouvons attribuer des valeurs aux variables, en utilisant l'opérateur « := » . L'affectation peut se produire après que la variable soit déclarée ou n'importe où dans la partie exécutable.

Exemple :

```
1 DECLARE
2
3     a  NUMBER := 3;
4
5 BEGIN
6
7     a := a + 1;
8
9 END;
10
```

Boucle

La PL/SQL nous permet de créer des boucles d'une manière assez familière !

```
1 IF <condition_0> THEN ... ELSE ... END IF;
2
3 IF <condition_1> THEN ...
4
5 ELSIF <condition_2> THEN ...
6
7 ...
8
9 ELSIF <condition_n> THEN ...
10
11 ELSE ...
12
13 END IF;
```

Curseur

La technique du curseur déclaré explicitement permet de parcourir une à une les lignes d'une requête. L'intérêt du curseur est donc de pouvoir récupérer (par un SELECT) plusieurs lignes.

Les étapes :

- On déclare le curseur
- On l'ouvre
- On en récupère les lignes une à une, en commençant par la première
- On le referme

Exemple :

```
5   ma_biere BIERE%ROWTYPE;
6 BEGIN
7   OPEN mesBieres;
8   LOOP
9     FETCH mesBieres INTO ma_biere;
10    ...
11    EXIT WHEN mesBieres%NOTFOUND;
12  END LOOP;
13  CLOSE mesBieres;
14 END;
```

Procédure

Les procédures en PL/SQL se comportent comme des procédures dans un autre langage de programmation. Une procédure est introduite par les mots-clés CREATE PROCEDURE suivi par le nom de la procédure et ses paramètres. Il est possible de suivre CREATE par OR CREATE, l'avantage est que si vous avez déjà fait la définition vous ne recevrez pas de message d'erreur. Mais si vous avez une procédure du même nom vous ne serez pas averti et l'ancienne sera perdue.

Il peut y avoir n'importe quel nombre de paramètre, chacun suivi d'un mode et d'un type. Les modes possibles sont IN (read-only), out (write-only) and INOUT (read and write).

```
1 CREATE TABLE Test (
2
3   a INTEGER,
4
5   b CHAR(10)
6
7 );
8
9 CREATE PROCEDURE addTup (i IN NUMBER) AS
10
11 BEGIN
12
13   INSERT INTO Test VALUES(i, 'xxx');
14
15 END addTup;
16
```

Conclusion

C'est ainsi que s'achève cet article sur l'initiation au PL/SQL. Vous avez pu constater finalement que la réalisation de procédure en PL/SQL n'est pas bien compliquée. Si vous avez des questions je serai ravi d'y répondre par mail.

Have Fun !