

PL/SQL : Procédures – Fonctions – Packages

1. Les procédures

Une procédure est un bloc PL/SQL nommé qui exécute une ou plusieurs actions.
Une procédure peut être stockée dans la base de données, comme tout autre objet, et peut être réutilisée à son souhait.

Syntaxe :

```
CREATE [OR REPLACE] PROCEDURE nom_proc  
  (param1 [mode] type_donnée,  
   param2 [mode] type_donnée,  
   ....  
  )  
IS  
  Bloc PL/SQL
```

Où :

Mode : IN (par défaut) | out | IN OUT

Type_donnée : char, date, varchar2, number, %type,....

Utiliser 'SHOW ERRORS' pour voir les erreurs de la compilation.

Pour exécuter la procédure : execute nom_proc (paramètres)

Exo #1 :

```
CREATE OR REPLACE PROCEDURE proc_sal (p_empno EMP.empno%type) IS  
  v_ename EMP.ename%type;  
  v_sal EMP.sal%type;  
BEGIN  
  SELECT ename, sal INTO v_ename, v_sal  
  FROM EMP  
  WHERE empno = p_empno;  
  DBMS_OUTPUT.PUT_LINE(p_empno || ' ' || v_ename || ' ' || v_sal);  
END;  
/
```

N.B. :

Pour tester une procédure :

SQL> EXECUTE proc_sal(7369)

Ou créer un bloc PL/SQL dans lequel on fait appel à la procédure :

```
DECLARE  
  v_num EMP.empno%type;  
BEGIN  
  v_num := 7369;  
  proc_sal(v_num);  
END;
```

Suppression d'une procédure : DROP PROCEDURE nom_proc

Dictionnaire de données : USER_SOURCE, USER_OBJECTS, USER_ERRORS

2. Les fonctions

C'est aussi des blocs PL/SQL nommés qui retournent une valeur.

Une fonction peut être stockée dans la base de données comme les autres objets de la base de données.

Syntaxe :

```
CREATE [OR REPLACE] FUNCTION nom_fonction
  (param1 [mode] type_donnée,
   param2 [mode] type_donnée,
   ....
 )
RETURN type_donnée IS
  Bloc PL/SQL;
```

Où :

Mode : IN seulement

Return type_donnée : ne doit pas avoir de dimension.

Exo #1 :

```
CREATE OR REPLACE FUNCTION sal_emp (v_num IN EMP.empno%type)
RETURN number IS
  v_sal EMP.sal%type := 0;
BEGIN
  SELECT sal INTO v_sal FROM EMP WHERE empno = v_num;
  Return (v_sal);
END;
/
```

N.B. :

Pour tester une procédure :

```
SQL> select sal_emp(7369) from dual;
```

ou créer un bloc PL/SQL dans lequel on fait appel à la fonction :

```
DECLARE
  v_sal number := 0;
BEGIN
  v_sal := sal_emp(7369);
  DBMS_OUTPUT.PUT_LINE('Salaire : '||v_sal);
END;
```

Suppression d'une fonction : DROP FUNCTION nom_fonction

3. Les packages

Un package est un ensemble de procédures, de fonctions, de variables, de constantes, de curseurs et d'exceptions stockés dans la base de données Oracle.

Pour concevoir un package, on doit créer une spécification et un corps.

Ne peut être appelé ni paramétré (package).

A. Spécification

C'est la partie déclaration de tous les composants du package.

Syntaxe :

```
CREATE [OR REPLACE] PACKAGE nom_package IS
```

```
    Section declaration;
```

```
END;
```

Où :

Section déclaration : consiste à déclarer les variables, les constantes, les fonctions, les procédures, les curseurs et les exceptions.

B. Corps du package

Définit les procédures, les fonctions, les variables, les constantes, les curseurs et les exceptions du package.

Syntaxe :

```
CREATE PACKAGE BODY nom_package IS
```

```
    Corps des procédures;
```

```
    Corps des fonctions;
```

```
    Déclaration des variables;
```

```
    Déclaration des constantes;
```

```
    Corps des curseurs;
```

```
    Déclaration des exceptions;
```

```
END;
```

Exo #1:

Créer un package (pack1) contenant :

- une variable global initialisée à 10
- une fonction (fonct1) avec 1 paramètre en entrée (no employé) et retourne le salaire de l'employé
- un curseur (cur_max) qui ramène le salaire maximal de tous les employés
- une fonction (fonct2) qui retourne le salaire maximal de tous les employés en utilisant le curseur.
- une procédure (proc1) avec 1 paramètre en entrée (le no employé) et 2 paramètres en sortie le nom et le salaire de l'employé.

Réponse :

Spécification :

```
CREATE OR REPLACE PACKAGE pack1 IS
```

```
    p_global number := 10;
```

```
    FUNCTION fonct1(p_num number) return number;
```

```
    CURSOR cur_max IS SELECT max(sal) FROM EMP;
```

```
    FUNCTION fonct2 return number;
```

```
    PROCEDURE proc1(p_num number, p_nom OUT char, p_sal OUT number);
```

```
END;
```

Body (corps)

```
CREATE OR REPLACE PACKAGE BODY pack1 IS
    FUNCTION fonct1(p_num number) return number IS
        v_sal number;
    BEGIN
        SELECT sal INTO v_sal FROM EMP WHERE empno = p_num;
        Return (v_sal);
    END;

    FUNCTION fonct2 return number IS
        v_max number;
    BEGIN
        OPEN cur_max;
        FETCH cur_max INTO v_max;
        CLOSE cur_max;
        Return (v_max);
    END;

    PROCEDURE proc1(p_num number, p_nom OUT char, p_sal OUT number) IS
    BEGIN
        SELECT ename, sal INTO p_nom, p_sal
        FROM EMP
        WHERE empno = p_num;
    END;
END;
```

Test sous SQL*PLUS

```
SQL> Variable g_var number
      Execute :g_var := pack1.p_global
      Print g_var

Variable g_sal number
      Execute :g_sal := pack1.fonct1(7934)
      Print g_sal

Variable g_nom char(15)
      Execute pack1.proc1(7369, :g_nom, :g_sal)
      Print g_nom
      Print g_sal

Variable g_max number
      Execute :g_max := pack1.fonct2
      Print g_max
```

Test avec un bloc PL/SOL

DECLARE

 p_nb number;

 p_char char(15);

BEGIN

 p_nb := pack1.fonct1(7369);

 DBMS_OUTPUT.PUT_LINE('Le salaire de l''employé 7369 est : '|| p_nb);

 DBMS_OUTPUT.PUT_LINE('Le salaire de l''employé 7369 est : '||
 pack1.fonct1(7369));

 pack1.proc1(7698, p_char, p_nb);

 DBMS_OUTPUT.PUT_LINE('Le nom et le sal de l''employé 7698 est : '||
 p_char||' '||p_nb));

 DBMS_OUTPUT.PUT_LINE('Le sal max des employés est : '||pack1.fonct2);
END;

Suppression des packages :

- Pour supprimer le corps et la spécification d'un package :

 DROP PACKAGE nom_package

- Pour supprimer le corps d'un package :

 DROP PACKAGE BODY nom_package