

Les procédures –fonctions packages et triggers

Les procédures stockées:

Définition:

Une procédure est un code PL/SQL défini par l'utilisateur et stocké dans la base de données. Ce qui permet d'éliminer la redondance de code.

Avantages:

- Le code SQL est précompilé.
- Exécution plus rapide (puisque stockée dans le serveur)
- Moins de code redondant

Les procédures stockées

- Syntaxe générale:

```
CREATE OR REPLACE PROCEDURE schema.NOMProcedure  
(param1 [IN|OUT|IN OUT ] typeparam1,  
param2 [IN|OUT|IN OUT] Typeparam2, ...) AS|IS  
Déclarations des variables locales (sans DECLARE)  
BEGIN  
Bloc PL/SQL  
END;
```

Syntaxe (suite)

- CREATE indique que l'on veut créer une procédure.
- OR REPLACE (facultative) permet d'écraser une procédure portant le même nom
- Param1, param2 sont les paramètres de la procédure
- IN :indique que le paramètre transmis par le programme appelant n'est pas modifiable par la procédure. (entrée) par défaut les paramètres sont IN
- OUT: indique que les paramètres sont modifiables par la procédure (sortie)
- IN OUT: combinaison de IN et OUT

Exemple

```
CREATE OR REPLACE PROCEDURE Augmentation
(
  PNCODECOUR IN RESULTATS.CODECOURS%Type,
  PNNOTEIN NUMBER
)
IS
BEGIN
  -- augmentation de la note
  Update RESULTATS Set NOTE = NOTE + PNNOTEIN
  Where CODECOURS = PNCODECOUR ;
END;
```

EXERCICE 1

1. Écrire une petite procédure qui permet d'insérer dans la table DEPARTEMENTS. Cette procédure reçoit comme paramètres les valeurs à insérer .
2. Écrire une procédure qui met à jour les salaires des employés comme suit:
 - Lorsque le salaire d'un employé est null, alors on lui affecte le plus petit salaire (de tous les employés). Sinon on augmente le salaire avec une valeur souhaitée .
 - La procédure reçoit comme paramètres: le numéro d'employé (numemp) et l'augmentation du salaire (Augmentation)

Réponse

```
CREATE OR REPLACE  
PROCEDURE INSERECODETYPE  
(  
  CODE IN TYPELIVRE.CODETYPE%TYPE,  
  DESCR IN TYPELIVRE.DESCRPTION%TYPE)  
AS  
  
BEGIN  
INSERT INTO TYPELIVRE VALUES (CODE,DESCR);  
END ;
```

Remarques

- Pour exécuter une procédure dans SQL Developer , il suffit de taper l'instruction suivante:
EXECUTE nomProcédure(val1, val2,..)
EXECUTE augmentation('KED',10);
- Pour détruire une procédure, utiliser DROP PROCEDURE nomProcédure
- Lorsqu'une procédure a besoin de déclarer ses variables, pas besoins de DECLARE. Les variables sont déclarer entre IS et BEGIN

Les fonctions

- Les fonctions sont considérées comme des procédures qui retournent des valeurs.
- Synthaxe.

```
CREATE OR REPLACE FUNCTION NomFonction (param1 [IN|OUT|IN OUT ]
    typeparam1,
param2 [IN|OUT|IN OUT] Typeparam2, ...)
RETURN TypeVariable AS |IS
Declaration des variables locales (sans DECLARE)
BEGIN
    Bloc PL/SQL;
RETURN variable;
END ;
```

Exemple1

```
CREATE OR REPLACE FUNCTION afficher
```

```
(  
  CODER IN TYPELIVRE.CODETYPE%TYPE  
)  
  RETURN VARCHAR2 AS
```

```
DESCR varchar2(60);
```

```
BEGIN
```

```
  SELECT description into DESCR FROM TYPELIVRE where codetype = CODER;
```

```
  return DESCR ;
```

```
  DBMS_OUTPUT.PUT_LINE(DESCR);
```

```
END ;
```

cette fonction affiche la description (titre) d'un livre selon le code du livre

Exercice

1. Écrire une fonction qui permet de calculer le nombre d'employés dans un département . Cette fonction reçoit comme paramètre le code du département et retourne le nombre d'employés dans ce département.
2. Écrire une fonction qui permet d'insérer un enregistrement dans la table employés et qui retourne le numéro d'employé inséré. l'Insertion utilise une séquence.

Exécution et suppression

- Execution : (dans SQL developer)

```
SELECT nomFonction (liste des valeurs) from  
DUAL (ou SYS.DUAL);
```

```
select afficher(21) from dual;
```

```
select total from dual;
```

- Détruire une fonction:

```
DROP FUNCTION nomFonction;
```

Les Packages

- Un package est un objets de la base de données qui encapsule d'autres objets (procédures, fonctions ..)
- Un package a essentiellement deux parties:
 - la partie déclaration: dans cette partie sont déclarées les variables globales, les curseurs, les procédures et les fonctions
 - la partie corps du programme: dans cette partie, sont définies les procédures et les fonctions et les curseurs.

Exemple 1/2

```
CREATE OR REPLACE
PACKAGE GESTIONEMPLOYES AS

TYPE EMPENR IS REF CURSOR;

PROCEDURE INSERTION (NUM IN NUMBER , NOM IN
    VARCHAR2,PRENOM IN VARCHAR2, SALAIRE IN NUMBER );

PROCEDURE SUPPRESSION (NUM IN NUMBER);
PROCEDURE SELECTION (REQUETE OUT EMPENR );
FUNCTION TOTAL RETURN NUMBER;

END ;
```

Exemple 2/2

```
CREATE OR REPLACE PACKAGE BODY GESTIONEMPLOYES AS
  PROCEDURE INSERTION (NUM IN NUMBER , NOM IN VARCHAR2,PRENOM IN VARCHAR2, SALAIRE IN NUMBER ) AS
  BEGIN
  INSERT INTO EMPLOYES (numemp, nomemp, prenomemp, salaireemp ) VALUES (NUM,NOM,PRENOM ,SALAIRE);
  END;
  PROCEDURE SUPPRESSION (NUM IN NUMBER) AS
  BEGIN
  DELETE FROM EMPLOYES WHERE NUMEMP = NUM;
  END;
  FUNCTION TOTAL RETURN NUMBER AS
  TOTALEMP NUMBER;
  BEGIN
  SELECT COUNT(*) INTO TOTALEMP FROM EMPLOYES;
  RETURN TOTALEMP;
  END;
  PROCEDURE SELECTION(REQUETE OUT EMPENR )AS
  BEGIN
  OPEN REQUETE FOR SELECT * FROM EMPLOYES ;
  END;
END GESTIONEMPLOYES;
```