

Stockage de traitements en bases de données

Les packages PL/SQL

PL/SQL : Les packages 1/10

- Un package permet de regrouper des modules de traitement qui ont un lien logique entre eux sous une seule entité.
- Les modules de traitement peuvent être :
 - procédures
 - fonctions
 - triggers
 - exceptions
 - variables
 - curseurs
 - constantes

PL/SQL : Les packages 2/10

- Un packages est composé de deux parties créées et compilées séparément :
 - Une partie **déclaration** (spécification) qui contient la déclaration des modules qui sont accessibles de l'extérieur (**PUBLIC**)
 - Une partie **body** (corps) qui contient les définitions des modules déclarés dans la partie déclaration ainsi que les déclarations et les définitions des modules de type **PRIVE**.

PL/SQL : Les packages 3/10

- On choisira de créer des packages dans le but :
 - d'une meilleur organisation des programmes (regroupement des objets par thème comme par exemple un packages qui regroupera tous les objets chargés de gérer les clients et dans un autre packages tous les objets chargés de gérer les commandes ...)
 - d'une approche objet

PL/SQL : Les packages 4/10

- **Partie déclaration**

```
CREATE [OR REPLACE] packages  
  nom_package
```

```
[IS|AS]
```

```
{[déclaration de procédure ;]
```

```
|[déclaration de fonctions ;]
```

```
|[déclaration de variable ;]
```

```
|...}
```

PL/SQL : Les packages 5/10

- **Partie body**

```
CREATE [OR REPLACE] packages BODY  
  nom_package
```

```
[IS|AS]
```

```
{[déclaration et corps de procédure ;]
```

```
 |[déclaration et corps de fonctions ;]
```

```
 |[déclaration de variable ;]
```

```
 |...}
```

```
END nom_package
```

PL/SQL : Les packages 6/10

- **Exemple**

```
CREATE OR REPLACE PACKAGE exemple1  
IS
```

```
-- déclaration d'une variable globale au package--
```

```
vglobal NUMBER :=0 ;
```

```
PROCEDURE test(pnom in client.nom%TYPE);
```

```
FUNCTION moyenne RETURN number;
```

```
END exemple1;
```

```
/
```

PL/SQL : Les packages 7/10

- **Exemple**

```
create or replace package body exemple1  
as
```

```
    procedure test(pnom in client.nom%type) is
```

```
    begin
```

```
    end test;
```

```
    function moyenne return number is
```

```
    begin
```

```
    return(1);
```

```
    end moyenne;
```

```
end exemple1;
```

```
/
```

PL/SQL : Les packages 8/10

-- initialise la variable vglobal à 500--

SQL>exec exemple1.vglobal := 500

-- déclaration de la variable moyenne de
type number--

SQL>VARIABLE moyenne NUMBER

-- appel de la fonction moyenne--

EXECUTE :moyenne := exemple1.moyenne

-- affichage du résultat de la fonction--

Print moyenne

PL/SQL : Les packages 9/10

- **Pour modifier la partie spécification ou la partie corps d'un package, il suffit de modifier le texte source correspondant et d'exécuter l'un des ordres suivants :**

REPLACE PACKAGE nom_package

REPLACE PACKAGE BODY nom_package

PL/SQL : Les packages 10/10

- **Pour supprimer un package entier**
DROP PACKAGE nom_package;
- **Pour supprimer la partie BODY d'un packages**
DROP PACKAGE BODY nom_package;

PL/SQL : Les rôles

- **Un rôle est un ensemble nommé de privilèges.**
- **Un rôle peut être attribué à des utilisateurs ou à d'autres rôles.**

Deux aspects d'application :

- **rôle utilisateur**
- **rôle d'application**

PL/SQL : Les rôles

CREATE ROLE nom_rôle

**[not identified|identified {by
mot_de_passe}]**

- **Par défaut un rôle est créé initialement sans aucune caractéristique.**
- **Il est alimenté par l'ordre GRANT.**
- **Le rôle doit être activé par l'ordre SET.**

PL/SQL : Les rôles

Exemple :

**Create user U_dev identified by
motdepasse;**

CREATE ROLE R_dev ;

GRANT connect,resource to R_dev ;

GRANT R_dev to U_dev ;

PL/SQL : Les rôles

Exemple de rôle définis par ORACLE :

CONNECT (CREATE session,create cluster,create view,create table,create sequence ...)

RESOURCE (CREATE cluster, CREATE procedure,create sequence,create table,create trigger...)

DBA tous les privilèges