

SGBD - 2^e

PL-SQL - Chapitre 8 - Les packages

Daniel Schreurs

10 août 2022

Haute École de la Province de Liège

Table des matières du chapitre i

1. Le concept du package
2. Persistance de la session
3. Principaux avantages des packages
4. Procédures et fonctions polymorphes et surchargées

Le concept du package

Le concept du package

- Un package est composé de **deux parties** de code bien distinctes : la **spécification** (specification) et le **corps** (body) ;
- La **spécification** du package contient les prototypes des procédures et fonctions et la déclaration des variables publiques
- Le **corps** du package contient l'implémentation des éléments définis dans la spécification. Le corps peut également contenir et définir des éléments privés.
- Le corps du package peut également contenir une section d'instructions appelée initialization section.

Le concept du package

- Le code défini publiquement est accessible à tout utilisateur possédant le **privilège EXECUTE** sur le package
- La sécurité repose sur des **GRANT EXECUTE**. Aucun accès aux données n'est accordé, uniquement aux procédures, fonctions et packages.
- Le code privé est inaccessible : ceci renforce encore l'indépendance données-programmes et la réutilisabilité

Le concept du package

Le serveur de bases de données assure que le code n'est exécuté qu'une seule fois par session lorsque le package est instancié

```
1 CREATE OR REPLACE PACKAGE PublicVarPersistante IS
2     vPublicVar CHAR := 'Y';
3 END PublicVarPersistante;
```

Persistence de la session

Il est possible :

- D'établir une session ;
- D'exécuter un programme qui assigne une valeur à une variable d'un package ;

Cette variable est donc initialisée et persistera jusqu'à la fin de la session

L'exemple le plus simple est celui d'un package ne comprenant qu'une spécification où une **variable publique** est déclarée.

```
1 CREATE OR REPLACE PACKAGE
2     PublicVarPersistante IS
3     vPublicVar CHAR := 'Y';
4 END PublicVarPersistante;
```

Persistence de la session

Accès à la variable publique :

- Tout utilisateur ayant reçu le privilège d'exécuter le package possède une instance de celui-ci dès qu'il y accède pour la première fois
- Ainsi, un tel utilisateur pourra exécuter la portion de code suivante où INPRES est le nom du schéma propriétaire du package.

Accès à la variable publique

```
1 BEGIN
2   DBMS_OUTPUT.PUT_LINE ('Valeur : ' ||
3     INPRES.PublicVarPersistence.VPublicVar);
4 END;
```

Pour rendre une variable privée, il suffit de ne pas la mettre dans la spécification.

```
1 CREATE OR REPLACE PACKAGE Session_Persistante IS
2     PROCEDURE AssigneVariable(P_Valeur CHAR);
3     FUNCTION AfficheVariable RETURN CHAR;
4 END SessionPersistante;
```

Important

Les variables publiques sont locales à une session.

Principaux avantages des packages

Principaux avantages des packages

- Un package permet de regrouper plusieurs unités de programmation dans un même container réduction du nombre d'objets stockés à maintenir
- Le fait de séparer l'implémentation de l'interface permet de maintenir ou modifier l'implémentation sans toucher à l'interface ni aux programmes appelants.
- Il n'est même pas nécessaire de recompiler les programmes appelants si l'interface n'est pas elle-même recompilée

Procédures et fonctions polymorphes et surchargées

Procédures et fonctions polymorphes et surchargées

Spécification du package contenant 2 fonctions polymorphes

```
1 CREATE OR REPLACE PACKAGE Pkgclients IS
2     -- ici refClient
3     FUNCTION RechercheClient(p_refClient IN clients.
4         refClient%TYPE)
5         RETURN Clients%ROWTYPE;
6     -- ici nomClient
7     FUNCTION RechercheClient(p_nomClient IN clients.
8         nomClient%TYPE)
9         RETURN Clients%ROWTYPE;
10 END PkgClients;
```


Procédures et fonctions polymorphes et surchargées i

Body du package contenant 2 fonctions polymorphes

```
1 CREATE OR REPLACE PACKAGE BODY Pkgclients IS
2     FUNCTION RechercheClient(p_refClient clients.
        refClient%TYPE)
3         RETURN Clients%ROWTYPE
4     IS
5         v_Client Clients%ROWTYPE;
6 BEGIN
7     SELECT *
8     INTO v_Client
9     FROM Clients
10    WHERE RefClient = p_refClient;
11
12    RETURN v_Client;
13 END;
```

Procédures et fonctions polymorphes et surchargées ii

```
14  FUNCTION RechercheClient(p_nomClient clients.  
    nomClient%TYPE)  
15      RETURN Clients%ROWTYPE  
16      IS  
17      v_Client Clients%ROWTYPE;  
18  BEGIN  
19      SELECT *  
20      INTO v_Client  
21      FROM Clients  
22      WHERE nomClient = p_nomClient;  
23  
24      RETURN v_Client;  
25  END;  
26  END PkgClients;
```