

6



Packages



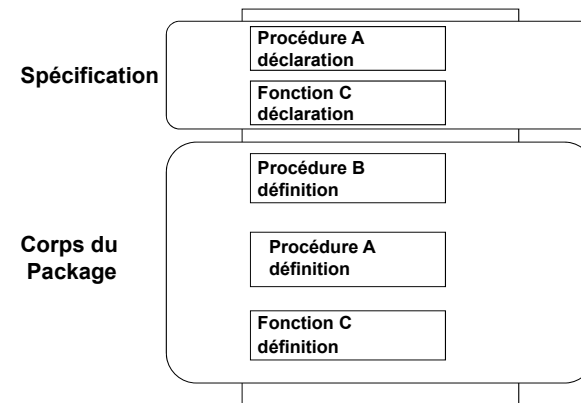
Généralités

- Regroupement *logique* de types PL/SQL, variables, procédures et fonctions
- Deux parties
 - Spécification
 - Corps
- Ne peut pas être appelé
- Permet au serveur Oracle de charger plusieurs objets simultanément en mémoire

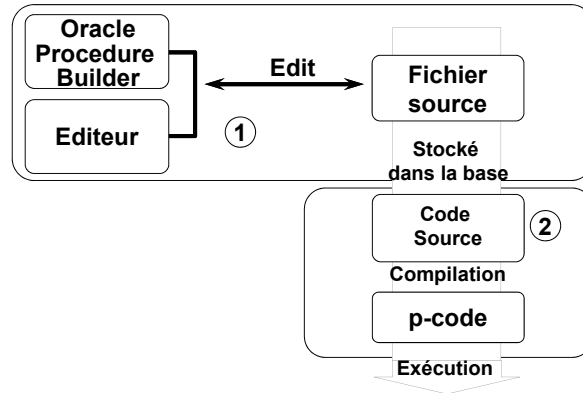
Avantages des packages

- Meilleur développement des applications
- Modularité
- Informations privées
- Ajout de fonctionnalités

Structure d'un package



Développement d'un package



294

Bases de données - © Christine Bonnet

Développement d'un package (suite)

- Écrire dans deux fichiers différents les parties *spécification* et *corps* facilite les modifications ultérieures
- La partie *spécification* peut exister sans la partie *corps*
- La partie *corps* ne peut pas exister sans la partie *spécification*

295

Bases de données - © Christine Bonnet



Création de la spécification du package

Syntaxe

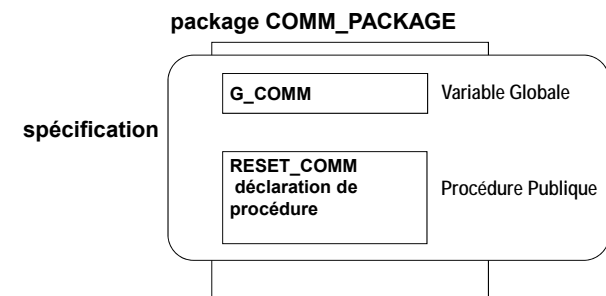
```

CREATE [OR REPLACE] PACKAGE nom_package
IS | AS
    type public et déclarations de variables
    spécifications de procédures et/ou fonctions
END nom_package;
    
```

296

Bases de données - © Christine Bonnet

Déclaration d'éléments publics



297

Bases de données - © Christine Bonnet

Déclaration d'éléments publics

Exemple

```
CREATE OR REPLACE PACKAGE comm_package IS
  g_comm NUMBER := 10; --initialisé à 10
  PROCEDURE reset_comm (v_comm IN NUMBER);
END comm_package;
```

298

Bases de données - © Christine Bonnet

Utilisation d'une variable globale

```
SQL>EXECUTE comm_package.g_comm := 5
```

Utilisation d'une procédure publique

```
SQL>EXECUTE comm_package.reset_comm(8)
```

299

Bases de données - © Christine Bonnet



Création d'un corps de package

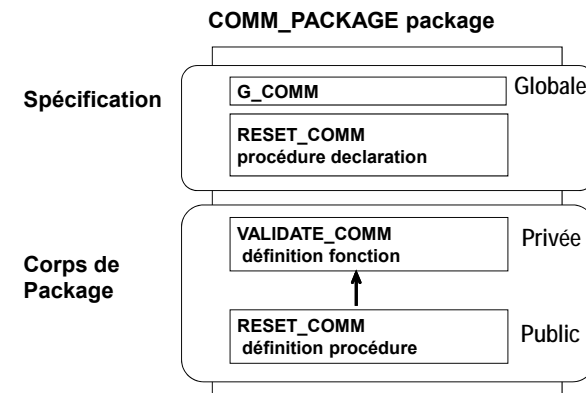
Syntaxe

```
CREATE [OR REPLACE] PACKAGE BODY nom_package
IS | AS
  type privé et déclarations de variables
  corps de procédures et/ou fonctions
END nom_package;
```

300

Bases de données - © Christine Bonnet

Éléments publics et privés



301

Bases de données - © Christine Bonnet

Création d'un corps de package Exemple

```
CREATE OR REPLACE PACKAGE BODY comm_package IS
  FUNCTION validate_comm (v_comm IN NUMBER)
  RETURN BOOLEAN
  IS
    v_max_comm NUMBER;
  BEGIN
    SELECT MAX(comm) INTO v_max_comm FROM emp;
    IF v_comm > v_max_comm THEN RETURN(FALSE);
    ELSE RETURN(TRUE);
    END IF;
  END validate_comm;
```

302

Bases de données - © Christine Bonnet

Exemple - suite

```
PROCEDURE reset_comm(v_comm IN NUMBER)
IS
  v_valid BOOLEAN;
BEGIN
  v_valid := validate_comm(v_comm);
  IF v_valid = TRUE THEN
    g_comm := v_comm;
  ELSE
    RAISE_APPLICATION_ERROR(-20210, 'Invalid
    commission');
  END IF;
  END reset_comm;
END comm_package;
```

303

Bases de données - © Christine Bonnet

Appel d'un composant d'un package

Exemple 1: appel d'une fonction à partir d'une procédure du même package

```
CREATE OR REPLACE PACKAGE BODY comm_package IS
  . . .
  PROCEDURE reset_comm(v_comm IN NUMBER)
  IS
    v_valid BOOLEAN;
  BEGIN
    v_valid := validate_comm(v_comm); -- nom fonction
    IF v_valid = TRUE THEN
      g_comm := v_comm;
    ELSE
      RAISE_APPLICATION_ERROR (-20210, 'Invalid comm');
    END IF;
  END reset_comm;
END comm_package;
```

304

Bases de données - © Christine Bonnet

Exemple 2: Appel d'une procédure d'un package avec SQL*Plus

Nom_package.nom_procedure

```
SQL> EXECUTE comm_package.reset_comm(1500);
```

Exemple 3: Appel d'une procédure d'un package d'un schéma différent

Nom_schéma.nom_package.nom_procedure

```
SQL> EXECUTE scott.comm_package.reset_comm(1500);
```

305

Bases de données - © Christine Bonnet

Exemple 4: Appel d'une procédure d'un package d'une base distante

Nom_package.nom_procedure@lienBD

```
SQL> EXECUTE comm_package.reset_comm@ny (1500);
```

306

Bases de données - © Christine Bonnet

Référence à une variable globale à l'intérieur d'un package

Exemple

```
CREATE OR REPLACE PACKAGE BODY comm_package IS
    . . .
    PROCEDURE reset_comm(v_comm IN NUMBER)
    IS
        v_valid    BOOLEAN;
    BEGIN
        v_valid := validate_comm(v_comm);
        IF v_valid = TRUE THEN
            g_comm := v_comm; -- nom variable
        ELSE
            RAISE_APPLICATION_ERROR (-20210,'Invalid comm');
        END IF;
    END reset_comm;
END comm_package;
```

307

Bases de données - © Christine Bonnet

Référence à une variable globale par une procédure externe

Exemple

```
CREATE OR REPLACE PROCEDURE hire_emp
(v_ename IN      emp.ename%TYPE,
v_mgr   IN      emp.mgr%TYPE,
v_job   IN      emp.job%TYPE,
v_sal   IN      emp.sal%TYPE)
IS
    v_comm emp.comm%TYPE;
    . . .
BEGIN
    . . .

    v_comm := comm_package.g_comm;
    -- nom_package.nom_variable
    . . .
END hire_emp;
```

308

Bases de données - © Christine Bonnet



Suppression d'un package

Suppression de la spécification et du corps :

```
DROP PACKAGE nom_package;
```

Suppression du corps seul :

```
DROP PACKAGE BODY nom_package;
```

309

Bases de données - © Christine Bonnet



Exercices