



ORAE505

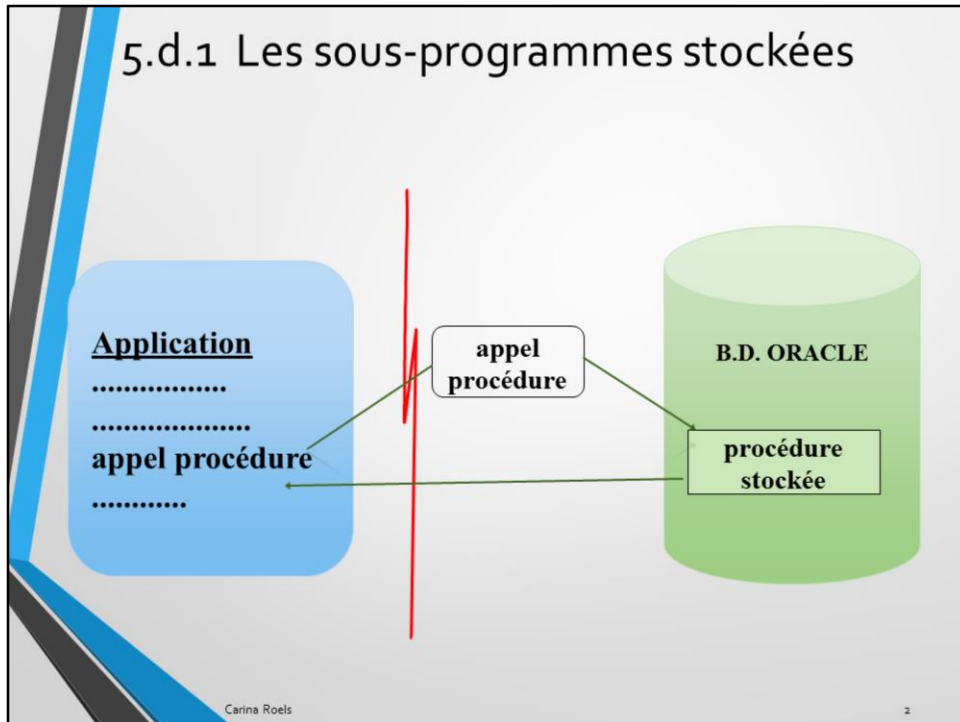
Conception et administration de B.D.

5. Le langage PL/SQL

d. Les sous-programmes stockés

Carina Roels

5.d.1 Les sous-programmes stockés



Un sous-programme stocké :

- Une FONCTION ou une PROCEDURE crée séparément, compilée et stockée dans la base de données.
- Un objet de la base de données prêt à l'emploi pour n'importe quelle application.

3 types de sous-programmes :

- sous-programmes stockés séparément.
- sous-programmes stockés dans un PACKAGE.
- sous-programmes locaux, donc définis dans un bloc PL/SQL ou dans un autre sous-programme. Ces sous-programmes ne sont pas directement utilisables par d'autres applications.

Un sous-programme peut être appelé à partir de :

- un trigger sur base
- un autre sous-programme stocké
- une application Pro*...
- SQL*PLUS
- Etc.

5.d.1 Les sous-programmes stockées

Procédure stockée

**Pour la réalisation d'actions spécifiques.
Pas de retour de valeur .**

Fonction stockée

**Pour la réalisation de calculs,
avec retour de valeur .**

5.d.2 Les actions possibles

Pour pouvoir gérer des programmes stockés, il faut avoir le privilège adéquat.

CREATE PROCEDURE

CREATE ANY
PROCEDURE

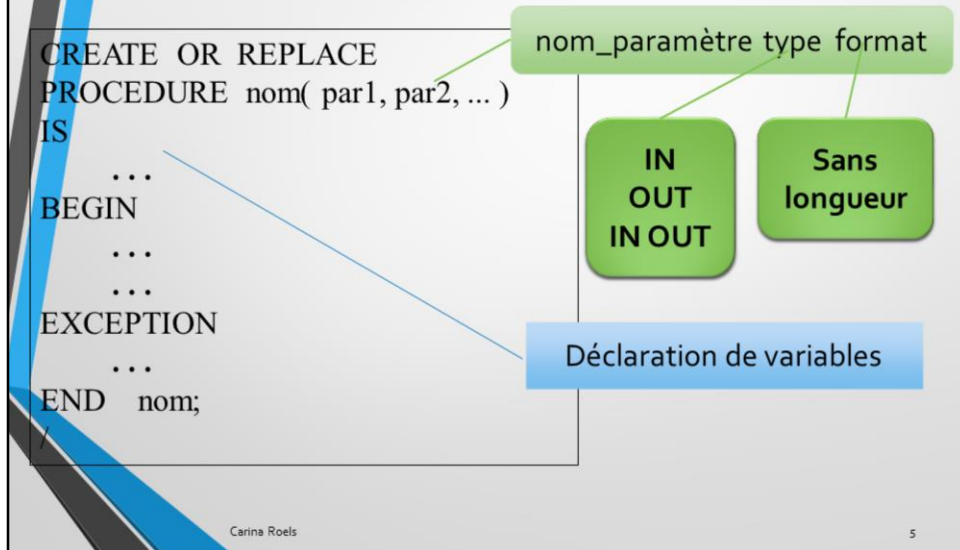
Les actions sur les programmes stockés :

Création ou modification

Compilation

Suppression

5.d.3 La création d'une procédure stockée



Les paramètres des procédures sont spécifiés de la manière suivante :

nom_variable type_de_paramètre format_sans_taille

type_de_paramètre : **IN OUT IN OUT**

Format_sans_taille : un format reconnu dans ORACLE
éventuellement suivi de **:= valeur_par_défaut**

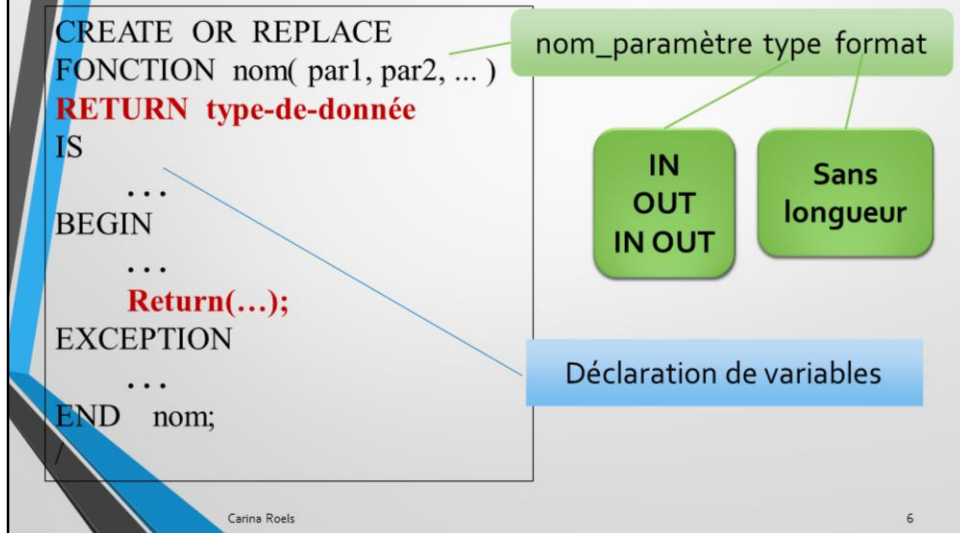
Exemple de procédure :

```
CREATE PROCEDURE credit( no_compte IN NUMBER , montant IN NUMBER )  
AS  
BEGIN  
    UPDATE COMPTE  
    SET balance = balance + montant  
    WHERE id_compte = no_compte ;  
END;  
/
```

Pour appeler la procédure :

credit(valeur_pour_no_compte , valeur_pour_montant);

5.d.4 La création d'une fonction stockée



Exemple de fonction :

```
CREATE FUNCTION lecture_balance( no_compte IN NUMBER )
RETURN NUMBER
IS
    bal_compte NUMBER( 11,2 )
BEGIN
    SELECT balance INTO bal_compte
    FROM comptes
    WHERE id_compte = no_compte ;

    RETURN( bal_compte );
END ;
/
```

Pour appeler la fonction :

```
balance = lecture_balance( valeur_pour_no_compte );
```

5.d.5 La compilation

```
ALTER FUNCTION [schéma.]nom_fonction COMPILE;  
ALTER PROCEDURE [schéma.]nom_procedure COMPILE;
```

La compilation d'un programme stocké →

- Détermination des meilleurs chemins d'accès pour toutes les requêtes contenues dans le programme (Index, lecture séquentielle, etc.)
- Stockage des informations dans le D.D.

5.d.6 La suppression

```
DROP FUNCTION [schéma.]nom_fonction;
```

```
DROP PROCEDURE [schéma.]nom_procedure;
```


5.d.7 La gestion des erreurs dans les programme stockés

```
RAISE_APPLICATION_ERROR( no_erreur , message_erreur );
```

Si dans un sous-programme, un `RAISE_APPLICATION_ERROR` est rencontré, les opérations suivantes sont exécutées :

- `ROLLBACK` de toutes les mises à jour
- Envoi du N° et du message définis vers le programme appelant. Ils peuvent ensuite être interceptés comme n'importe quel erreur ORACLE.
- Arrêt de l'exécution du sous-programme

Carina Roels

9

Exemple :

```
CREATE OR REPLACE PROCEDURE  augmentation( idemp NUMBER, montant  NUMBER )
IS
    salaire_actuel      NUMBER;
BEGIN
    SELECT      salaire INTO  salaire_actuel
    FROM        EMPLOYES
    WHERE       noemp  = idemp;

    IF  salaire_actuel IS NULL
    THEN
        RAISE_APPLICATION_ERROR( -20101, 'Pas de salaire !' );
    ELSE
        UPDATE  EMPLOYES
        SET      salaire = salaire_actuel + montant
        WHERE    noemp  = idemp;
    END IF;
END augmentation;
/
```

L'application appelante intercepte l'erreur à travers de `SQLCODE` et `SQLERRM`, par l'intermédiaire de `OTHERS` ou en utilisant `EXCEPTION_INIT`.

```
DECLARE
    .....;
    salaire_null EXCEPTION;
    PRAGMA      EXCEPTION_INIT( salaire_null , -20101 );
BEGIN
    .....;
    augmentation( :no_employe , :mont_augm );
    .....;
    .....;

EXCEPTION
    WHEN  salaire_null THEN
        .....;
        .....;
END;
/
```