

## Deuxième partie – Les Procédures et Fonctions stockées :

### Fonctions stockées

#### 6) Fonction stockée nbEtudiantsParGroupe.

- En vous inspirant de ce que vous avez écrit lors des questions 2 ou 3, écrire une fonction stockée nbEtudiantsParGroupe qui retourne le nombre d'étudiants qui appartiennent au groupe dont l'identifiant p\_idGroupe est passé en paramètre. Cette fonction doit avoir la signature suivante :

```
FUNCTION nbEtudiantsParGroupe(p_idGroupe IN Groupes.idGroupe%TYPE)
                                RETURN NUMBER
```

Si on passe en paramètre de cette fonction un identifiant de groupe qui n'existe pas, on souhaite que la fonction retourne NULL (et non pas 0).

Si lors de la création de votre fonction, vous avez une erreur de compilation (Warning: Function created with compilation errors) vous pouvez demander à Oracle de vous afficher la liste des erreurs grâce à l'instruction SHOW ERRORS.

On rappelle que sous Oracle, pour tester une fonction stockée, on peut utiliser la pseudo-table DUAL.

- Résultats attendus :

```
SELECT nbEtudiantsParGroupe('T1')
FROM DUAL ;
```

```
nbEtudiantsParGroupe('T1')
-----
6
```

```
SELECT nbEtudiantsParGroupe('T4')
FROM DUAL ;
```

```
nbEtudiantsParGroupe('T4')
-----
```

// quand le résultat est NULL, par défaut, iSQL\*Plus n'affiche rien

#### 7) Fonction stockée nbEtudiantsParPromotion.

- Ecrire une fonction stockée nbEtudiantsParPromotion qui retourne le nombre d'étudiants qui appartiennent à une promotion dont l'identifiant p\_idPromotion est passé en paramètre. Cette fonction doit avoir la signature suivante :

```
FUNCTION nbEtudiantsParPromotion(
                                p_idPromotion IN Promotions.idPromotion%TYPE)
                                RETURN NUMBER
```

Dans le corps de cette fonction, on pourra appeler la fonction qui a été réalisée à la question précédente (nbEtudiantsParGroupe). Cette fonction pourra être appelée dans le SELECT d'une requête.

On ne vous demande pas de gérer explicitement le cas où l'identifiant de promotion qui est passé en paramètre n'existe pas, ou bien le cas où la promotion existe mais qu'elle n'a pas de groupe (dans ces cas là, votre fonction devrait retourner NULL par défaut sans que vous soyez obligé de le programmer).

- Résultat attendu :

```
SELECT nbEtudiantsParPromotion('A1')
FROM DUAL ;
```

```
nbEtudiantsParPromotion('A1')
-----
16
```

- Ecrire une requête SQL qui, en utilisant la fonction écrite ci-dessus, met à jour l'attribut nbEtudiantsPromotion de toutes les lignes de la table *Promotions*. Puis, vérifier que dans la table *Promotions* on a bien 16 étudiants en A1 et 11 en A2.

**Procédures stockées**8) Procédure stockée `affichageInfosEtudiant`.

- En vous inspirant de ce que vous avez écrit lors de la question 4, écrire une procédure stockée `affichageInfosEtudiant` qui affiche toutes les informations nominatives qui concernent l'étudiant dont l'identifiant `p_idEtudiant` est passé en paramètre. Cette procédure doit avoir la signature suivante :

```
PROCEDURE affichageInfosEtudiant(
                                p_idEtudiant IN Etudiants.idEtudiant%TYPE)
```

On rappelle que sous Oracle, on peut appeler une procédure stockée avec l'instruction `CALL`.

- Résultat attendu :

```
CALL affichageInfosEtudiant('E1');

Identifiant étudiant : E1
Nom étudiant       : Alizan
Prénom étudiant    : Gaspard
Sexe étudiant      : M
Date naissance étudiant : 14.07.1997
Groupe étudiant    : T1

CALL affichageInfosEtudiant('E30');
L'étudiant E30 n'existe pas
```

9) Procédure stockée `miseAJourCoefficientModules`.

- Ecrire une procédure stockée qui met à jour l'attribut `coefficientModule` de toutes les lignes de la table *Modules*. Le coefficient d'un module doit être égal à la somme des coefficients des matières qui sont incluses dans le module. Cette procédure doit avoir la signature suivante :

```
PROCEDURE miseAJourCoefficientModules
```

- Résultat attendu :

```
CALL miseAJourCoefficientModules();
```

```
SELECT idModule, coefficientModule
FROM Modules
```

idModule	coefficientModule
M111	6
M112	6
M113	3
M121	6
M122	6
M123	3
M211	6
M212	6
M213	3
M221	6
M222	6