

TP N°: 3

LES FPROCEDURES, LES FONCTIONS & LES EXCEPTIONS



ENSEIGNANTE : MME ONS BEN ROMDHANE

SEMESTRE : 2 - 2022/2023

MATIERE : ATELIERS SGBD

GROUPE : DSI2.2 G2

Objectifs :

-  - Traiter les exceptions prédéfinies : TOO_MANY_ROWS & NO_DATA_FOUND
-  - Gérer les exceptions personnalisées : RAISE & RAISE_APPLICATION_ERROR
- Manipuler les curseurs implicites, explicites et paramétrés
- Utiliser les attributs des curseurs implicites
- Pratiquer les différentes structures itératives : For, While & Loop
- Développer des procédures stockées (Procedure & Function)

EXERCICE 1 : EXCEPTIONS PREDEFINIES

Le bloc PL/SQL suivant accepte un identifiant d'un employé (id) en entrée et renvoie son nom (first_name et last_name) :

```
DECLARE
    name varchar2(50);
    emp_id employees.employee_id%TYPE ;
BEGIN
    emp_id := &id; --paramètre à saisir au clavier (en web, ça marche pas !)
    -- get the employee name by id
    SELECT first_name || ' ' || last_name INTO name
    FROM employees
    WHERE employee_id = emp_id;
    -- show the employee name
    dbms_output.put_line('Employee name is ' || name);
END;
```

1) Si vous exécutez ce bloc PL/SQL en utilisant la valeur zéro pour le paramètre id, Oracle provoque l'erreur suivante :

ORA-01403: no data found -- it is a predefined exception.

Modifier ce bloc pour gérer cette exception prédéfinie.

2) Remplacez la clause :

```
WHERE employee_id = emp_id
par
```

```
WHERE employee_id != emp_id
```

Gérer maintenant l'exception provoquée par Oracle.

EXERCICE 2 : FONCTION STOCKEE + EXCEPTION

- 1) Développer une fonction nommée **Get_FullName** qui permet d'afficher le nom complet d'un employé à partir de son **id** passé en paramètre.
- 2) Écrire un bloc PL/SQL qui fait appel à cette fonction en utilisant l'id **107**.
- 3) Réexécuter la fonction avec l'id **1007** (le résultat affiché sera **null**). Modifier la fonction **Get_FullName** pour qu'elle retourne la valeur *'Not Found !'* lorsque l'id est inexistant.
- 4) Modifier la fonction de telle sorte qu'elle lève l'exception au bloc appelant qui va traiter l'exception et afficher le message *'Not Found !'*.

EXERCICE 3 : PROCEDURE STOCKEE

- 1) Écrire une procédure stockée nommée **TOPK** permettant d'afficher le nom et le salaire des **k** (passé comme paramètre) employés les plus payés de la table **EMP**
- 2) Exécuter la procédure stockée en utilisant un bloc PL/SQL anonyme ou la commande **EXECUTE**.
- 3) Tester l'exécution de la procédure stockée avec plusieurs valeurs de **k**.

```
1- King      : 24000 €
2- Kochhar   : 17000 €
3- De Haan   : 17000 €
4- Russell   : 14000 €
5- Partners  : 13500 €
```

Indications :

- Définir un curseur qui liste les employés par ordre de salaire décroissant.
- Utiliser la fonction **RPAD()** pour bien justifier l'affichage à l'écran.
- Effectuer un parcours explicite du curseur qui s'arrête tant que **k** n-uplets n'ont pas été lus.

EXERCICE 4 : CURSEUR EXPLICITE ET EXCEPTION

Afin d'établir une corrélation, on souhaite connaître la différence de commission moyenne entre les employés successivement enregistrés dans la table **EMP**. La table **EMP** est remplie de commissions valuées (c'est-à-dire, pour lesquelles l'attribut **commission_pct** n'est pas **NULL**) ou non. Les commissions non valuées ne sont pas à prendre en compte. Écrire un bloc PL/SQL anonyme permettant de calculer la différence de commission moyenne entre les employés.

Indications :

- Créer un curseur contenant les commissions valuées.
- À l'aide d'un parcours explicite du curseur (FETCH), lire la première commission puis, pour toutes les commissions suivantes, cumuler la valeur absolue de commission courante – commission précédente (fonction ABS).
- Exception : Interrompre le programme immédiatement si la table EMP contient moins de deux commissions valuées (nb_com). Pour ce type de traitements, on utilise habituellement une requête COUNT (pour tester cette exception, on peut élever le nombre minimum de commissions à 40, par exemple).

	PRIME
1	5600
2	4050
3	3600
4	3300
5	2100
6	3000
7	2375
8	2250
9	1600
10	1500

... (35 lignes)

Résultat attendu =

(|5600 -4050| + |3600-4050| + |3300-3600| + |2100-3300|+ ...)
/ 34

Moyenne Prime = 0685.88 \$