

TP N°06 (semaines 47) : PLSQL – Bloc Anonymes – Curseurs

Exercice N°1 : Curseurs

On veut écrire un bloc PL/SQL qui affiche **pour une date et un agent donnés**, la liste des chambres (avec son nb de couchages) dont il doit faire l'entretien (celles qui sont au planning à la date donnée). Il faudra afficher également le nombre de chambres pour cet agent à cette date.

Gérer les cas suivants : agent inexistant, agent sans chambre, autres erreurs (date incorrecte par ex.).



Un curseur est un pointeur vers une zone SQL privée qui stocke des informations sur le traitement d'une instruction SELECT, INSERT, UPDATE ou DELETE.

L'ensemble des résultats récupérés lors de l'exécution du SQL et les attributs du curseur y sont stockés. Les curseurs peuvent être classés en curseurs implicites et explicites.

- Oracle crée un curseur **implicite** pour toutes les instructions SQL incluses dans la section exécutable d'un bloc PL/SQL. Dans ce cas, le cycle de vie du curseur est géré par la base de données Oracle.
- Pour les curseurs **explicites**, le cycle d'exécution peut être contrôlé par l'utilisateur

Ex. : Pour l'agent A01 le 12/11/2007 → *Planning du 12/11/2007 pour l'agent A01 :
Chambre 5 avec 3 Couchages*

...

Le nombre de chambre(s) de l'agent A01 est : 3.

Pour l'agent A08 le 12/11/2007 → *Agent inexistant*

Pour l'agent A04 le 12/11/2007 → *Pas de chambre pour cet agent le 12/12/2007.*

Pour l'agent A04 le 12/18/2007 → *Erreur*

- 1) Utiliser un curseur explicite avec déclaration (CURSOR ..IS), ouverture (OPEN), parcours (FETCH) et fermeture (CLOSE) du curseur.



L'utilisation d'un curseur **explicite** comprend les étapes suivantes :

- Déclaration du curseur pour l'initialisation de la mémoire (SECTION DECLARE).
- Ouverture du curseur pour allouer la mémoire (OPEN).
- Lorsque le curseur est ouvert, les lignes peuvent être extraites du curseur une par une ou dans un bloc pour effectuer la manipulation des données (FETCH).
- Fermer le curseur pour libérer la mémoire allouée (CLOSE).

- 2) Utiliser un curseur avec une boucle FOR (dans sa forme condensée)



Il existe un moyen de manipuler le curseur sans déclaration au préalable et de laisser Oracle faire le reste, en utilisant la boucle for :

BEGIN

```
FOR cur IN (SELECT chp1, chp2 FROM maTable WHERE ...) LOOP
    -- Traitements sur la ligne courante
    -- accès aux données par la notation pointée :
    -- cur.chp1, cur.chp2
```

END LOOP;

END;

Exercice N°2 : Curseurs et exceptions

- 1) Ecrire un bloc PL/SQL qui affiche pour un type de téléphone et un numéro de client donnés, la liste des numéros correspondants. Afficher également le nombre de numéros récupérés.

Gérer les cas suivants : type de téléphone inexistant, client inexistant, client sans téléphone de ce type, éventuelles autres erreurs.

Ex. : Pour le type **TEL** et le client **59** → *N°1 : 01-42-02-54-47*
N°2 : 01-42-38-02-02
2 numéros correspondants

Pour le type **BUR** et le client **59** → *BUR : type de téléphone inexistant*
Pour le type **TEL** et le client **110** → *client 110 inexistant*
Pour le type **GSM** et le client **89** → *pas de téléphone de type GSM pour le client 89*

- 2) Ajouter dans la table TYPE un nouveau type de téléphone : BUR – Téléphone Bureau

Tester à nouveau le bloc précédemment écrit :

Ex. : Pour le type **BUR** et le client **59** → *pas de téléphone de type BUR pour le client 59*

- 3) Modifier le bloc PLSQL pour tenter de supprimer le type de téléphone dans le cas où aucun téléphone correspondant n'a été trouvé pour le client choisi.

En cas d'erreur lors de la suppression (type utilisé par d'autres clients), afficher un message.

Ex. : Pour le type **BUR** et le client **59** → *pas de téléphone de type BUR pour le client 59*
Type BUR inutilisé supprimé.

Pour le type **GSM** et le client **89** → *pas de téléphone de type GSM pour le client 89*
Type GSM utilisé ne peut être supprimé.