

Travaux Pratiques PL SQL

Les Curseurs

Objectifs :

- Déclarer et utiliser des **curseurs explicites**.
- Gérer des **curseurs avec paramètres** pour des requêtes dynamiques.
- Utiliser des curseurs pour les **misés à jour conditionnelles** avec FOR UPDATE et CURRENT OF

Partie 1 : Curseurs explicites

1. Afficher les projets en cours

- a. Créez un curseur explicite qui sélectionne les projets ayant le statut '**En cours**'.
- b. Parcourez les résultats avec une boucle et affichez : *Nom du projet, Budget et Date de début*.

2. Lister les employés d'un projet

- a. Créez un curseur explicite qui affiche les **noms et postes des employés** associés au projet ayant `id_projet = 1`.
- b. Ajoutez un compteur pour afficher le nombre total d'employés trouvés.

Partie 2 : Curseurs avec paramètres

3. Filtrer les tâches d'un projet donné

- a. Créez un curseur avec un paramètre `p_id_projet` qui sélectionne les tâches associées à un projet donné.
- b. Parcourez les résultats pour afficher : *Description, Date limite, et Statut des tâches*.

4. Calculer le total des salaires d'un projet

- a. Créez un curseur avec un paramètre `p_id_projet` qui sélectionne les employés d'un projet donné.
- b. Utilisez une boucle pour calculer et afficher le **salaire total** des employés travaillant sur ce projet.

Partie 3 : Curseurs avec mises à jour conditionnelles

5. Modifier le statut des tâches non commencées

- Créez un curseur explicite pour sélectionner les tâches ayant le statut '**Non commencé**'.
- Utilisez la clause `FOR UPDATE` pour verrouiller ces lignes.
- Parcourez le curseur et mettez à jour leur statut en '**En cours**' à l'aide de `WHERE CURRENT OF`.

6. Augmenter les salaires des développeurs

- Créez un curseur pour sélectionner les employés ayant le poste '**Développeur**'.
- Parcourez les résultats pour augmenter leur salaire de **10%** si celui-ci est inférieur à **5000**.
- Affichez les noms des développeurs et leur nouveau salaire.

Partie 4 : Combinaison des curseurs

7. Vérifier et mettre à jour les budgets des projets

- Créez un curseur pour sélectionner tous les projets.
- Pour chaque projet :
 - Parcourez ses dépenses via un curseur imbriqué.
 - Si le total des dépenses dépasse le budget, affichez un message : *"Le budget du projet [Nom] est dépassé."*

Partie 5 : Archivage et Table Auxiliaire

8. Archivage des tâches terminées :

Utiliser un curseur explicite pour déplacer les tâches **terminées** de la table `taches` vers une table auxiliaire appelée `archive_taches`.

- Créer une table `archive_taches` avec les mêmes colonnes que `taches` :

```
CREATE TABLE archive_taches AS SELECT * FROM taches WHERE 1=0;
```

- Créer un curseur pour sélectionner toutes les tâches avec statut = 'Terminé'.
- Parcourir le curseur et insérer les données dans `archive_taches`.
- Supprimer les tâches déplacées de la table `taches`.
- Créez un bloc PL/SQL qui :
 - Identifie les tâches terminées.
 - Archive ces tâches dans la table `archive_taches`.
 - Supprime les tâches archivées de la table `taches`.

9. **Exercice 9 : Vérification des budgets et archivage :** Créer

une vérification avancée sur les budgets des projets.

- a. Créer une table `budget_historique` pour archiver les projets dépassant leur budget :

```
CREATE TABLE budget_historique AS SELECT * FROM projets WHERE 1=0;
```

- b. Utiliser un curseur pour parcourir les projets.
- c. Calculer le total des dépenses pour chaque projet en utilisant un curseur imbriqué ou une requête dynamique.
- d. Si le total des dépenses dépasse le budget :
- Insérer le projet dans `budget_historique`.
 - Mettre à jour son statut en '**Suspendu**'.
- e. Créez un bloc PL/SQL qui :
- Vérifie pour chaque projet si les dépenses totales dépassent le budget.
 - Archive les projets concernés dans `budget_historique`.
 - Met à jour leur statut en Suspendu.