Titre : "PL/SQL Simplifié : Un Guide Pratique"

Chapitre 1 : Introduction à PL/SQL

- Qu'est-ce que PL/SQL ?

- Pourquoi utiliser PL/SQL ?

- Structure de base d'un bloc PL/SQL

Chapitre 2 : Les Curseurs

- Définition et utilité des curseurs

- Types de curseurs : implicites et explicites

- Exemple pratique : Affichage des employés avec un salaire supérieur à un montant spécifié (exo4.sql, exo1)

Chapitre 3 : Les Boucles

- Boucle FOR

- Boucle WHILE

- Exemple pratique : Affichage des départements (exo2.sql, exemple 1)

Chapitre 4 : Les Fonctions

- Création de fonctions

- Paramètres et valeurs de retour

- Exemple pratique : Calcul de la somme des salaires par département (exo4.sql, exo4)

Chapitre 5 : Les Procédures

- Création de procédures

- Différence entre fonctions et procédures

- Exemple pratique : Réduction des prix des produits d'une catégorie (exo4.sql, exo5)

Chapitre 6 : Manipulation de Données

- SELECT, INSERT, UPDATE, DELETE dans PL/SQL

- Exemple pratique : Augmentation des salaires (exo1.sql, EXO4)

Chapitre 7 : Gestion des Exceptions

- Types d'exceptions

- Capture et traitement des exceptions

- Exemple à ajouter (non présent dans les fichiers fournis)

Chapitre 8 : Techniques Avancées

- Combinaison de curseurs et boucles

- Exemple pratique : Affichage des employés par département (exo5.sql, exo2)

Chapitre 9 : Traitement de Données Complexes

- Agrégation et groupement de données

- Exemple pratique : Calcul des ventes par produit et par trimestre (exo5.sql, exo3)

Chapitre 10 : Optimisation et Bonnes Pratiques

- Conseils pour écrire du code PL/SQL efficace

- Gestion de la mémoire et des ressources

Pour chaque chapitre, nous utiliserions les exemples de code fournis pour illustrer les concepts, en expliquant chaque partie du code de manière détaillée et en montrant comment ces concepts s'appliquent dans des scénarios réels de gestion de base de données.

Ce livre permettrait aux lecteurs de comprendre progressivement les concepts de PL/SQL, en partant des bases jusqu'aux techniques plus avancées, tout en s'appuyant sur des exemples concrets et pratiques.

1. Curseurs

Un curseur en PL/SQL est un outil qui permet de parcourir un ensemble de résultats retournés par une requête SQL. C'est essentiellement un pointeur vers une zone de mémoire contenant ces résultats.

Types de curseurs :

- Curseurs implicites : Automatiquement créés par Oracle pour chaque instruction SQL

- Curseurs explicites : Définis manuellement par le programmeur

Exemple détaillé (de exo4.sql) :

```sql

CURSOR c\_employes (p\_salaire NUMBER) IS

SELECT Prenom, Nom, Salaire

FROM EMPLOYE

WHERE Salaire > p\_salaire;

```

Dans cet exemple :

- 'c\_employes' est le nom du curseur

- '(p\_salaire NUMBER)' est un paramètre du curseur

- La requête SELECT définit les données que le curseur va récupérer

Utilisation typique :

```sql

FOR emp IN c\_employes(3000) LOOP

DBMS\_OUTPUT.PUT\_LINE(emp.Prenom || ' ' || emp.Nom || ' : ' || emp.Salaire);

END LOOP;

```

2. Fonctions

Une fonction en PL/SQL est un bloc de code nommé qui effectue une tâche spécifique et retourne une valeur. Les fonctions sont utiles pour encapsuler la logique et la réutiliser.

Exemple détaillé (de exo3.sql) :

```sql

CREATE OR REPLACE FUNCTION NombreEmployes(p\_ID\_Departement IN NUMBER)

RETURN NUMBER IS

v\_NombreEmployes NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_NombreEmployes

FROM EMPLOYES

WHERE ID\_Departement = p\_ID\_Departement;

RETURN v\_NombreEmployes;

END;

```

Dans cette fonction :

- 'NombreEmployes' est le nom de la fonction

- '(p\_ID\_Departement IN NUMBER)' est le paramètre d'entrée

- 'RETURN NUMBER' spécifie le type de valeur retournée

- La logique dans le bloc BEGIN-END compte les employés d'un département

- 'RETURN v\_NombreEmployes' renvoie le résultat

Utilisation :

```sql

v\_Nombre := NombreEmployes(10);

DBMS\_OUTPUT.PUT\_LINE('Nombre d''employés : ' || v\_Nombre);

```

3. Procédures

Une procédure est similaire à une fonction, mais elle ne retourne pas nécessairement une valeur. Les procédures sont utilisées pour effectuer des actions ou des séries d'actions.

Exemple détaillé (de exo4.sql) :

```sql

CREATE OR REPLACE PROCEDURE reduire\_prix\_categorie(p\_id\_categorie NUMBER) IS

CURSOR c\_produits IS

SELECT ID\_Produit, Prix

FROM PRODUITS

WHERE ID\_Categorie = p\_id\_categorie

FOR UPDATE OF Prix;

BEGIN

FOR prod IN c\_produits LOOP

UPDATE PRODUITS

SET Prix = Prix \* 0.9

WHERE CURRENT OF c\_produits;

END LOOP;

COMMIT;

END;

```

Dans cette procédure :

- 'reduire\_prix\_categorie' est le nom de la procédure

- '(p\_id\_categorie NUMBER)' est le paramètre d'entrée

- La procédure utilise un curseur pour parcourir les produits d'une catégorie

- Elle met à jour le prix de chaque produit en le réduisant de 10%

- 'COMMIT' valide les modifications

Utilisation :

```sql

BEGIN

reduire\_prix\_categorie(1);

END;

```

Ces explications détaillées devraient aider à mieux comprendre les concepts fondamentaux de PL/SQL comme les curseurs, les fonctions et les procédures, en montrant leur structure et leur utilisation pratique.