

TP 6 : PL/SQL – Les curseurs

Soit le schéma relationnel de la base de données commerciale "BDCOM":

Client (Codc, Nomc, Ville, CA, Cred, CredMax)

Commande (Numc, Datec, Mntc, #Codc) -- '#' indique que l'attribut est une clé étrangère

Produit (RefP, Libp, PU, Qtes, Seuil)

Fournisseur (Codf, Nomf, Villef)

Lig_cde (#Numc, #RefP, Qtec)

Frs_Prod (#Codf, #RefP, PUf, Qtcf)

Avant de répondre aux exercices, créer un nouvel utilisateur nommé TP4, lui attribuer les privilèges nécessaires, créer les tables et insérer les données. Ainsi, on vous demande de suivre les étapes suivantes :

- `Connect system/mot de passe de system`
- `Create user TP6 identified by tp6;`
- `Grant connect, resource, create view to TP6;`
- `Connect TP6/tp6`
- `Start c:\tp6\creationTP6.sql` -- on suppose que le dossier tp6 est dans c:\
- `Start c:\tp6\insertionTP6.sql`
- `Set serveroutput on`

Exercice 1 :

Ecrire un programme PL/SQL qui permet d'afficher la liste des produits. On veut afficher pour chaque produit sa référence, son libellé et son prix.

Exercice 2 :

Ecrire un programme PL/SQL qui permet de parcourir les produits, calculer pour chaque produit le nombre de pièces vendues et de sauvegarder le résultat dans la table **Vente_Prod** à créer avant l'exécution du programme.

Le schéma de la table Vente_Prod est le suivant :

Vente_Prod(Refp, Libp, PU, QteVendu)

N'oublier pas de vider la table Vente_Prod au début du programme.

Exercice 3 :

Ecrire un programme PL/SQL qui parcourt la liste des fournisseurs et calcule pour chacun le nombre de ses produits ainsi que le chiffre d'affaire réalisé avec ce fournisseur.

Les fournisseurs seront classés selon leur chiffre d'affaire :

- Si CA <1000 → Fournisseur passager
- Si 1000 ≤ CA <10000 → Fournisseur Moyen
- Si CA ≥ 10000 → Fournisseur important

Le résultat sera sauvegardé dans la table **Cat_Four** à créer avant l'exécution du programme.

Cat_Four(codf, nomf, nbProd, CA, Categorie)

Exercice 4 :

Ecrire un bloc PL/SQL qui permet de classer les **clients** dans les catégories suivantes :

- | | |
|-------------------|-----------------------------|
| 'Client fidèle' | si CA > 10 000 |
| 'Bon Client' | si CA entre 5 000 et 10 000 |
| 'Client passager' | si CA < 5 000 |

Rajouter à ce client une des mentions suivantes par rapport à l'état de son crédit :

- | | |
|----------------------|---|
| 'Client douteux' | si Cred > CredMax |
| 'Client doit régler' | si Cred entre 50% de son CredMax et CredMax |
| 'Client solvable' | si Cred < 50% de son CredMax |

Insérer les informations déterminées dans la table **Cat_Cli** à créer avant l'exécution du bloc PL/SQL :

Cat_Cli (Codc, Nomc, CA, Cred, CredMax, Categorie, Etat_Cred)

Categorie et Etat_Cred sont de type VARCHAR2(20).

Exercice 5 :

Ecrire un bloc PL/SQL qui permet de déterminer la quantité de réapprovisionnement (**Q_reap**) de chaque produit. Etant donné que cette quantité dépend de la demande observée du produit on la calculera selon la procédure suivante :

Q_tot_com = \sum (Qtec du produit).

Si (Q_tot_com > (2 * (Qtes-Seuil))) alors Q_reap = (Q_tot_com – Qtes) * 1.2

Si (Q_tot_com > (3 * (Qtes-Seuil))) alors Q_reap = (Q_tot_com – Qtes) * 1.3

Si (Q_tot_com > (4 * (Qtes-Seuil))) alors Q_reap = (Q_tot_com – Qtes) * 1.4

Si (Q_tot_com ≤ (2 * (Qtes-Seuil))) alors Q_reap = (Q_tot_com – Qtes)

Le résultat du bloc sera récupéré dans une table appelée :

Reapp (Refp, Libp, Q_reap, Mnt_reap, New_Stock)
Mnt_reap=Q_reap * PU & New_Stock=Qtes + Q_reap