

TP 7 : PL/SQL – Les Procédures et les Fonctions

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, Qtef)

Avant de répondre aux exercices, créer un nouvel utilisateur nommé TP7A, 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 TP7A identified by tp7;
- Grant connect, resource, create view to TP7A;
- Connect TP7A/tp7
- Start c:\tp7\creationTP7.sql -- on suppose que le dossier tp7 est dans c:\
- Start c:\tp7\insertionTP7.sql
- Set serveroutput on

Exercice 1 :

- 1) Ecrire une procédure PL/SQL qui pour un produit donné retourne son prix et sa quantité. La procédure doit afficher les messages suivants :
 - « Produit en rupture de stock » si la quantité est égale à 0.
 - « Produit nécessite un approvisionnement » si la quantité est inférieure à 5.
 - « Produit en promotion » si la quantité dépasse 50.

Procédure **Prix_Qte**(refProd IN **Produit.Refp%type**, Prix OUT **Produit.PU%type**, Qte OUT **Produit.qtes%type**)

Ou bien

Procédure **Prix_Qte**(refProd IN **varchar2**, Prix OUT **Number**, Qte OUT **Number**)

- 2) Ecrire un programme PL/SQL qui fait appel à **Prix_Qte** qui permet de saisir la référence d'un produit et d'afficher son prix et sa quantité en stock.
 - Si la quantité est inférieure à 5 alors majorer le prix de 15%.
 - Si la quantité dépasse 50 alors diminuer le prix de 20%.

Exercice 2 :

- 1) Ecrire une fonction qui pour un fournisseur donné permet de retourner son chiffre d'affaire (c'est le total des produits achetés de ce fournisseur).

Function **CA_Frs**(CodFour **Fournisseur.codf%**type) Return **Number**

- 2) Exécuter la fonction dans SQL-Plus en mettant le résultat dans une variable de liaison.
- 3) Ecrire un programme PL/SQL qui permet de saisir le code d'un fournisseur et affiche son chiffre d'affaire en utilisant la fonction **CA_FRS**.

Exercice 3 :

- 1) Ecrire une procédure qui pour un produit donné retourne le prix d'achat le moins cher ainsi que le code du fournisseur correspondant.

Procedure **Moins_Cher**(RefProd **IN Varchar2**, Prix **OUT Number**, CodFour **OUT Number**)

Ou bien

Procedure **Moins_Cher**(RefProd **IN Produit.refp%**type, Prix **OUT frs_prod.puf%**type, CodFour **OUT frs_prod.codf%**type)

- 2) Exécuter la procédure dans SQL-Plus en mettant les résultats dans des variables de liaison.
- 3) Ecrire un programme PL/SQL qui permet de saisir le code d'un produit et affiche son prix d'achat le moins cher et le fournisseur correspondant

Exercice 4 :

- 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.

Procedure **Affiche** /*Les procédures sans paramètres sont déclarées sans parenthèses*/

- 2) Exécuter la procédure dans SQL-Plus

Exercice 5 :

- 1) Ecrire une procédure qui pour un client donné permet de retourner sa catégorie et son état de crédit selon les conditions suivantes :

- Catégorie du client :

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

Procedure **Cat_Etat**(code **Client.Codc%**type, Categorie **OUT varchar2**, EtatCred **OUT varchar2**)

- 2) Ecrire un programme PL/SQL qui utilise la procédure Cat_Etat et permet de classer les clients dans des catégories. 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 6 :

- 1) Ecrire une fonction qui pour un produit donné retourne la somme des quantités commandées.
Function Qte_Com(RefProd varchar2) return Number
- 2) Ecrire une fonction qui permet de déterminer la quantité de réapprovisionnement (**Q_reap**) d'un produit donné. Etant donné que cette quantité dépend de la demande observée du produit on la calculera selon la procédure suivante :

Q_tot_com = Σ (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)

Function Qte_Reap(refProd varchar2) return Number

- 3) Ecrire un bloc PL/SQL qui permet de déterminer la quantité de réapprovisionnement de chaque produit. 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

Exercice 7

- 1) Ecrire une procédure qui permet d'afficher le détail d'une commande donné selon le format suivant :

```
| ***** |
| * Commande n° : 10 * Date : 10/10/2004 * Client : 100 **** |
| ***** |
| *****|** Référence **|**Qté Cdée**|** PU **|** Total **|
| *****|** BU44 **|** 10 **|** 125.5 **|* 1255 *|
| *****|** BU11 **|** 2 **|**1050.5 **|* 2101 *|
```

Procédure Affiche_Com(NumCde Commande.Nume%type)

- 2) Ecrire un programme PL/SQL qui permet d'afficher le détail de toutes les commandes.

Exercice 8

- 1) Ecrire une fonction qui permet de retourner le chiffre d'affaire réalisé par un client entre deux dates données.

Function **CA_entre_Date**(codclient **Client.Codc%**type, debut **Date**, fin **Date**) return **Number**

- 2) Ecrire une procédure qui permet d'afficher à l'écran pour un client donné ses commandes entre 2 dates données selon le format suivant :

```
| ***** |
| * Client : 100 * STS * Sousse * CA : 6289.7 ***** |
| ***** |
| *****|** Num Cde **|** Date **|** Montant **|***** |
| *****|** 50 **|** 15/01/16 **|** 2334.7 **|***** |
| *****|** 30 **|** 04/01/16 **|** 1826.5 **|***** |
| *****|** 10 **|** 02/01/15 **|** 2128.5 **|***** |
```

Procédure **Affiche_Cli**(CodClient **Number**, debut **Date**, fin **Date**)

- 3) Afin de bien faire le suivi des commandes des clients on vous demande d'écrire un bloc PL/SQL qui saisit deux dates et affiche à l'écran pour chaque client les commandes entre ces deux dates.

Exercice 9

- 1) Ecrire une fonction qui permet de calculer pour une commande donnée, le taux de remise possible.

- Si le montant de commande < 500 alors la remise=0.00
- Si le montant de commande ≥ 500 et <1500 alors la remise=0.05
- Si le montant de commande ≥ 1500 alors la remise=0.1

L'entête de la fonction étant le suivant :

Fonction **Remise_cde** (Num_Cde **IN Commande.NumC%**type) return **number**

- 2) Ecrire un bloc PL/SQL qui permet d'afficher toutes les commandes avec pour chaque commande:

le **Numc**, la **Datec**, le **Codc**, le **Mntc**, la **Remise** et le **montant à régler après remise**.

Utiliser pour ce bloc la fonction **Remise_cde**.