USTHB – Faculté d'Electronique et Informatique
Département Informatique

Architecture de S.G.B.D. relationnels

TP Oracle

PLSQL

Le langage PL/SQL (Procedural Language /SQL) est une extension du langage SQL qui offre un environnement procédural au langage SQL. Les fonctionnalités de PL/SQL sont les suivantes :
- Définition de variables, Traitements conditionnels, Traitements répétitifs, Traitements des curseurs, Traitements des erreurs

Les programmes PL/SQL sont organisés et sont interprétés en blocs. Un bloc est un ensemble de commandes, il est structuré en trois sections comme suit :

```
--BLOC PLSQL
DECLARE
/* Déclaration des variables, des types, des curseurs, fonctions et procédures */
BEGIN
/* Instructions PLSQL ; toute instruction est terminée par ; */
EXCEPTION
/* Traitement des erreurs */
END;
- - Fin du bloc PL/SQL
```

Remarque :
Le traitement des erreurs se fait en initialisant une variable de type EXCEPTION et ensuite l'utiliser dans la partie EXCEPTION.

Exemple :
Afficher les noms des produits par rang ensuite afficher le nombre des produits existant.

```
declare
cursor cr is select distinct refprod, nomprod
                        from produit
                        order by refprod, nomprod ;
i integer;
vide EXCEPTION;
begin
i:=1;
for item in cr
loop
dbms_output.put_line('Le produit N°' ||i||' est '||item.refprod||' '||item.nomprod);
i:=i+1;
end loop;

if(i<2) then RAISE vide;
else
i := i-1;
dbms_output.put_line('La base de données contient ' ||i||' produits ');
end if;
EXCEPTION
WHEN vide THEN dbms_output.put_line('La base de données ne contient aucun produit');
close cr;
end;
/
```

Pour afficher un texte vous utilisez le package DBMS_OUTPUT. Pour rendre les affichages visibles dans SQLPLUS, il faut utiliser la commande suivante : **SET SERVEROUTPUT ON**

Fonctions et procédures

Le code PlSQL peut être sauvegardé dans une procédure ou fonction avec ou sans paramètres.

```
CREATE [OR REPLACE] PROCEDURE Nom_de_procedure (arg1 type, arg2 type, ...) IS
Declaration de variables locales
BEGIN
Instructions;
END;
```

Pour exécuter une procédure :

```
SQL> EXECUTE Nom_de_procedure(valeurs des arguments);
```

Remarque : pour voir les erreurs syntaxiques commises lors de la déclaration une procédure, il faut utiliser l'instruction :

show errors procedure Nom_de_procedure;

Questions

Supposons que les tables des TP précédents sont crées et remplies.

1. Ecrire un code PL/SQL qui permet d'afficher pour chaque catégorie le nombre total de ses produits.

   Exemple : La catégorie **« Produits laitiers»** possède **4 produits**.

2. Ecrire une procédure qui supprime les produits qui n'étaient jamais commandés.

3. Ecrire une procédure qui affiche le nom, la fonction et le chiffre d'affaire généré des employés dont le chiffre d'affaire généré par ses commandes est entre 10000DA et 8000DA.

4. Ecrire une fonction qui retourne, pour chaque **client** donné, le **nombre de ses commandes**. Exécuter la fonction pour plusieurs clients.

   Exemple : Le client **Frankenversand** a **5** commandes.

5. Créer une procédure qui permet d'ajouter **une commande** à partir de tous les attributs nécessaires. N'oublier pas de vérifier l'unicité de la clé et l'existence de clé étrangère vers **les tables référencées**. Affichez les messages d'erreurs en cas de problèmes.