USTHB – Faculté d'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 employés de catégorie **Assistant** par rang ensuite afficher le nombre total des employés assistants.

```
DECLARE
cursor cr is select  nomemp from employe where categorie ='Assistant';        -- la définition du curseur PL/SQL
i integer;
vide EXCEPTION;
begin
i:=1;
for item in cr
loop
dbms_output.put_line( item.nomemp ||' est l''employé  ' ||i||' de catégorie assistant');
i := i+1;
end loop;
if(i<2) then RAISE vide;
else
i := i-1;
dbms_output.put_line('La catégorie Assistant  contient ' ||i||' employé(s) ');
end if;
EXCEPTION
WHEN vide THEN
dbms_output.put_line('La catégorie Assistant ne contient aucun employé');
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 PLSQL qui permet d'afficher pour chaque marque le nombre de modèles.

   Exemple : La marque **« TOYOTA»** possède **2 modèles**.

2. Ajouter la contrainte suivante : le salaire d'un employé doit être entre 10000 DA et 30000 DA.
   Le centre de maintenance décide d'augmenter le salaire de l'employé de catégorie assistant par 30% et le mécanicien par 50%. Ecrire une procédure qui augmente le salaire de chaque employé. Désactiver la contrainte d'intégrité pour effectuer les mises à jour. Afficher pour chaque employé avec son nouveau salaire.

   Exemple : **L'employé OUSSEDIK Hakim** de catégorie **Mécanicien** son salaire a passé de **20000 DA** à **30000 DA**

3. Ecrire une procédure *Vérification (période intervention)* qui affiche « vérification positive » si la date début d'intervention est inférieur à la date de fin d'intervention, et affiche « Vérification négative » sinon. Tester la procédure pour toutes **les interventions dont les véhicules réparés sont d'année 1998.**

4. Ecrire une fonction qui retourne, pour chaque **employé** donné, le **nombre** d'interventions effectuées. Exécuter la fonction pour plusieurs employés.

   Exemple : L'employé **IGOUDJIL Redouane** a fait 3 interventions**.**

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