USTHB – Faculté d'Electronique et Informatique
Département Informatique

Architecture de S.G.B.D. relationnels

TP Oracle

RELARQUE IMPORTANTE
Pour l'ensemble des TP, on demande de réaliser un compte-rendu du travail effectué. Ce compte-rendu comprendra un rapport (manuscrit ou réalisé sur traitement de texte) détaillant les points intéressants abordés à chaque question :
- requêtes SQL correspondant aux questions posées
- résultat de l'exécution de la requête
- réponses aux questions soulevées dans le sujet,
- explication des parties non triviales des requêtes complexes,
- analyse et commentaire des jeux d'essais.

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 patients de la mutuelle MAAF par rang ensuite afficher le nombre de patients existant.

```
DECLARE
cursor cr is select  nom_patient from patient where mutuelle='MAAF';      -- la définition du curseur PL/SQL
 c_rec cr%rowtype;                                                        -- c_rec prend le même type que cr
i binary_integer;
vide EXCEPTION;                                                          -- basically an integer
BEGIN
   i := 1;
   for c_rec in cr loop                                                  -- mettre cr dans c_rec
            dbms_output.put_line('Le patient  N°' ||i|| ' est '|| c_rec.nom_patient);
            i := i+1;
            exit when cr%notfound;
   end loop;
if(i<2) then RAISE vide;
else
i := i-1;
dbms_output.put_line('La mutuelle MAAF contient ' ||i|| ' patients ');
end if;
EXCEPTION    WHEN vide THEN
    dbms_output.put_line('la mutuelle MAAF ne contient aucun patient');
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 chambre de chaque service le nombre de lits libres et occupés.

   Exemple : La chambre N° **101** de service **Cardiologie** possède **2** lit(s) occupé(s) et **1** lit(s) libre(s).

2. L'hôpital a décidé d'augmenter les salaires des infirmiers. Les infirmiers de rotation nuit auront une augmentation de 60% par contre ceux de jour est de 50%. Ecrire une fonction qui augmente le salaire de chaque infirmier. Désactiver la contrainte d'intégrité pour effectuer les mises à jour. Afficher pour chaque infirmier le nouveau salaire.
   Exemple : L'infirmier **BADI Hatem** de rotation **nuit** son ancien salaire est : **19470,61** DA et son nouveau salaire est **31152,976** DA.

3. Ecrire une fonction *Vérification (Salaire)* qui affiche « vérification positive » si le salaire de l'infirmier respecte la contrainte d'intégrité, et affiche « Vérification négative » sinon. Tester la procédure pour tous les infirmiers.

4. Ecrire une fonction qui retourne, pour chaque spécialité donnée, le nombre de médecins affectés. Exécuter la fonction pour plusieurs spécialités.

5. Créer une procédure qui permet d'ajouter un employé de type infirmier à partir de tous les attributs nécessaires. N'oublier pas de vérifier l'unicité de la clé et l'existence des clés étrangères vers **EMPLOYE** et **SERVICE**. Affichez les messages d'erreurs en cas de problèmes.