##### 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 sportifs conseillés par rang ensuite afficher le nombre des conseillés existant.

**declare**

**cursor cr is select distinct nom, prenom**

**from sportifs**

**where idsportif in (select distinct IdSportifConseiller from sportifs)**

**order by nom, prenom;**

**i integer;**

**vide EXCEPTION;**

**begin**

**i:=1;**

**for item in cr**

**loop**

**dbms\_output.put\_line('Le conseillé N°' ||i||' est '||item.nom||' '||item.prenom);**

**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||' sportifs conseillés ');**

**end if;**

**EXCEPTION**

**WHEN vide THEN dbms\_output.put\_line('La base de données ne contient aucun conseillé');**

**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 PLSQL qui permet d’afficher pour chaque sport le nombre total des gymnases qui organisent des séances pour ce sport.

**Exemple :** Le sport **« Volley ball»** est organisé par **11 gymnases**.

1. Ajouter la contrainte suivante : l’âge d’un sportif doit être entre 20 ans et 45 ans.

On veut augmenter l’âge des sportifs par 5 ans. Ecrire une procédure qui augmente l’âge de chaque sportif. Désactiver la contrainte d’intégrité pour effectuer les mises à jour. Afficher pour chaque sportif son nouvel âge.

Exemple :le sportif **DEBOUBA Yasser** son âge a passé de **30** ans à **35** ans

1. Ecrire une procédure qui effectue cet affichage pour chaque gymnase donner par jour d’ouverture les horaires des premières et dernières séances. Tester la procédure pour les gymnases de superficie >400 m²**.**
2. Ecrire une fonction qui retourne, pour chaque **sportif** donné, le **nombre** **des sports entrainés**. Exécuter la fonction pour plusieurs sportifs.

**Exemple :** Le sportif **BOUTAHAR Abderahim** entraine **5** sports.

1. Créer une procédure qui permet d’ajouter **une séance** 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.