***TP 3 : Contraintes d’Intégrité Statiques et Dynamiques***

**Initiation au PL/SQL :**

**1- 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 chefs Algériens par rang ensuite afficher le nombre de chefs existant.

**DECLARE**

cursor cr is select nomc, pays, adresse from chef where pays ='ALGERIE'; -- la définition du curseur

c\_rec cr%rowtype; -- c\_rec prend le même type que cr

i binary\_integer; -- basically an integer

vide EXCEPTION;

**BEGIN**

i := 1;

for c\_rec in cr loop -- mettre cr dans c\_rec

dbms\_output.put\_line('Le Chef N°' ||i||' est '|| c\_rec.nomC);

i := i+1;

exit when cr%notfound;

end loop;

if(i<2) then RAISE vide;

else i := i-1;

dbms\_output.put\_line('Il y a ' ||i||’ chef(s) algérien(s) ');

end if;

**EXCEPTION** WHEN vide THEN

dbms\_output.put\_line('Il n'' y a aucun chef algérien!');

**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**

Pour voir les erreurs syntaxiques d’un code PL/SQL, il faut utiliser l’instruction : show errors.

**Fonctions et procédures**Le code PlSQL peut être sauvegardé dans une procédure ou fonction avec ou sans paramètres.  
Pour exécuter une procédure :

**CREATE [OR REPLACE] PROCEDURE Nom\_de\_procedure (arg1 type, arg2 type, …) IS**

/\* Declaration de variables locales \*/

**BEGIN**

/\*Instructions; \*/

**END**;

**2- TRIGGERS**

**Syntaxe de création d’un trigger**

***CREATE [OR REPLACE ] TRIGGER trigger\_name***

***{BEFORE | AFTER }***

***{INSERT [OR] | UPDATE [OR] | DELETE}***

***[OF col\_name]***

***ON table\_name***

***[FOR EACH ROW]WHEN (condition)***

***BEGIN***

***--- Instruction PLSQL***

***END;***

• ***CREATE [OR REPLACE ] TRIGGER trigger\_name*** *: pour créer ou écraser un trigger*  
*existant.*

• ***{BEFORE | AFTER | INSTEAD OF }*** : le moment du déclenchement du trigger (avant ou  
après l’opération de mise à jour).

• ***{INSERT [OR] | UPDATE [OR] | DELETE}*** : l’événement de mise à jour qui provoquera le  
déclenchement du trigger. Plusieurs événements séparés par OR sont possibles.

• ***[OF col\_name]*** *: utilisé dans le cas de l’opération Update appliquée sur une colonne*  
*particulière.*

• ***[ON table\_name]*** *: le nom de la table sur laquelle le trigger est défini*.

• ***[FOR EACH ROW]*** *: spécifie si le trigger est lancé pour chaque ligne affecté ou une seule*  
*fois.*

• ***WHEN (condition)*** *: le trigger est lance seulement lorsque la ligne affectée vérifie la*  
*condition.*

***Remarque*** *: pour générer une exception et empêcher le programme de continuer, l’utilisateur*  
*peut lancer la procédure* ***raise\_application\_error (-Num\_Message, 'Message à Afficher’) ;***  
*Num\_Message est compris entre 20000 et 20999.*

**Travail demandé**

En utilisant la BD créée dans le TP1, effectuer les manipulations suivantes :

**Contraintes d’Intégrité Statiques :**

1. Lister l’ensemble de contraintes d’intégrité pour chaque table de ce schéma.
2. Etendre la liste des rôles possibles avec la nouvelle fonction : « B-M à Domicile ».
3. Exigez que la date de naissance des patients soit antérieure à la date d’aujourd’hui. Tester avec quelques insertions.
4. Un ingénieur ne peut exercer en tant que Bio-Medical. Ajouter la contrainte et effectuer les tests nécessaires.
5. Supprimer la table Biologiste. Que remarquez-vous?

- Désactiver la CI qui bloque la suppression et réessayer. Que remarquez-vous?

- Supprimer cette contrainte et réessayer. Conclure.

- Recréer la table Biologiste.

**PL/SQL :**

1. Pour analyser l'organigramme du laboratoire, nous cherchons à afficher le personnel et leurs rôles.Ecrire un code PLSQL qui permet d’afficher pour chaque rôle, le nombre de biologistes correspondants.

**Exemple :** Il y a 2 personne(s) qui exerce(nt) en tant que «Aide-laboratoire».

1. Supposant que la norme mondiale pour le test antigénique Covid passe du seuil initial "0.5" à "0.8" pour que le résultat soit déclaré positif.Ajouter l'information, puis ajoutez la contrainte suivante : si le taux d'antigènes détectés est > 0.8 alors la conclusion doit être = "Positif".

Vérifiez après mis-à-j.

1. Ecrire une procédure *CasPositifs* qui affiche « Le patient "i" a été testé positif. » si le patient en question a un résultat positif aux tests Covid de différents types de prélèvement.
2. Ecrire une fonction qui retourne, pour un **Biologiste** donné, le **nombre** de prélèvements effectués. Exécuter la fonction pour plusieurs biologistes.

Exemple :Le Biologiste BADI a effectué 3 prélèvements.

1. Créer une procédure qui permet d’ajouter un prélèvement à partir de tous les attributs  
   nécessaires. N’oublier pas de vérifier l’unicité de la clé et l’existence de clé étrangère vers  
   Patient et Biologiste. Affichez les messages d’erreurs en cas de problèmes.
2. Créez un trigger qui affiche « un nouveau patient est ajouté» après chaque insertion  
   d’un patient. Répétez la même chose pour la modification ou la suppression.
3. Créer un trigger qui empêche la modification du numéro de biologiste de la table des prélèvements.
4. L’administrateur veut, pour un besoin interne, avoir le nombre total des prélèvements par biologistes. Pour cela, il ajoute un attribut : Nb\_Pr dans la table Biologistes.

* Ajoutez l’attribut.
* Créez un trigger **T1** qui permet de màj automatiquement l’attribut **Nb\_Pr**.

FIN