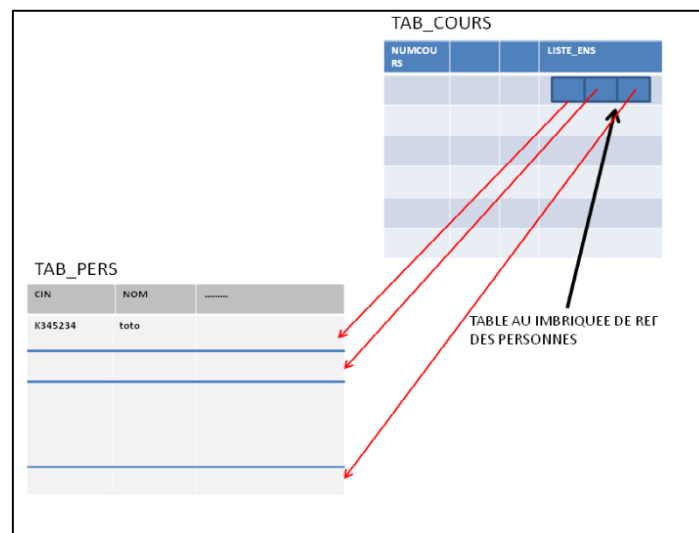


## LE TYPE COLLECTION: VARRAY (TABLEAU IMBRIQUE)

On se propose dans ce TP de manipuler le premier type de collection proposé dans Oracle, soit VARRAY. Pour ce, on reprend les mêmes types utilisés dans le précédent TP (donc prenez le même script et le modifier). Et, on ajoute un nouveau type T\_COURS défini par :

NUMCOURS de type NUMBER et qui est Le numéro du cours  
 INTIT de type VARCHAR2 de taille 20 et qui est l'intitulé du cours  
 VOLH de type NUMBER est qui est le volume horaire du cours  
 LISTE\_ENS la liste des références des personnes intervenant dans le cours et qui sont au maximum 3 (voir figure).



### I. DECLARATIONS DU TYPE T\_COURS ET DE SES METHODES

I.1. Créer le type T\_LISTE\_ENS de type tableau de références sur les personnes de la table TAB\_PERS et de taille maximale 3.

I.2. Créer le type T\_COURS avec les méthodes suivantes (juste les déclarations) :

- Un constructeur avec comme argument le numéro du cours, l'intitulé et le volume horaire ;
- Un constructeur avec comme argument le numéro du cours et l'intitulé ;
- Une fonction VOLH\_COURS qui retourne le volume horaire d'un cours;
- Une fonction LISTE\_PROF qui retourne la liste des intervenants dans le cours;
- Une procédure modif\_INTIT (NEW\_IN IN varchar2) qui modifie l'intitulé d'un cours en le mettant à NEW\_INTIT;
- Une procédure AJOUTER\_PROF (CIN\_PROF IN varchar2) qui ajoute la personne de CIN CIN\_PROF à la liste des intervenants si la limite n'est pas dépassée ;
- Une procédure AJOUTER\_LISTE\_PROF (CLISTE\_PROF IN T\_LISTE\_ENS) qui ajoute la liste CLISTE\_PROF à la liste des intervenants ;
- Une procédure RETIRER\_PROF (CIN\_PROF IN varchar2) qui retire la personne de CIN CIN\_PROF de la liste des intervenants du cours ;
- Une procédure AJOUTER\_COURS(N IN NUMBER, I IN varchar2, V IN NUMBER, LP IN T\_LISTE\_ENS) qui ajoute un nouveau cours;
- Une procédure AJOUTER\_COURS(C IN T\_COURS) qui ajoute le cours C dans la table des cours,
- Une procédure SUPPRIMER\_COURS(N IN NUMBER) qui supprime le cours numéro N.

I.3. Créer la table des cours TAB\_COURS avec comme clé primaire le numéro de cours et l'intitulé qui ne peut être NULL.

#### I.4. Insertion des données dans la table des cours

Insérer dans la table des cours les données suivantes :

NUMCOURS	INTIT	VOLH	LISTE_ENS (CIN DONNE)
1	BD	30	VIDE MAIS CREE
2	BDO	45	'd12434', 'X12434', 'F12434'
3	BDR	45	'F12434', 'X12434'
5	MATH	75	'X12434', 'g12434'
4	RES	50	NON DONNEE

## II. LES REQUETES SELECT

II.1. Afficher la liste la liste des cours ;

II.2. Afficher la liste des cours dont la liste des intervenants est vide ;

II.3. Afficher la liste des intervenants dans le cours numéro 2 (uniquement leurs ref) ;

II.4. Afficher la liste des intervenants dans le cours numéro 2 mais cette fois leurs CIN ;

II.5. Afficher par une requête la liste des cours ou intervient le professeur de CIN X12434 ;

II.6. Afficher par une requête la liste des professeurs qui n'interviennent dans aucun cours ;

## III. CODAGE DES METHODES

III.1. Coder les méthodes suivantes : les deux constructeurs, VOLH\_COURS, LISTE\_PROF et MODIF\_INTIT ;

III.2. Tester ces méthodes dans des blocs PL/SQL

III.3. Coder la procédure AJOUTER\_PROF (CIN\_PROF IN varchar2). La procédure doit traiter les exceptions suivantes : le prof passé en argument n'existe pas, le prof est déjà dans la liste et la liste est saturée ;

III.4. Tester la méthode précédente dans un bloc PL/SQL en testant les différents cas de figure (les exceptions traitées) ;

III.5. Coder la procédure AJOUTER\_LISTE\_PROF (CLISTE\_PROF IN T\_LISTE\_ENS) ;

III.6. Tester la méthode précédente dans un bloc PL/SQL ;

III.7. Coder la procédure RETIRER\_PROF (CIN\_PROF IN varchar2). La procédure doit traiter les exceptions suivantes : le prof passé en argument n'existe pas, le prof n'est pas dans la liste, la liste est vide ;

III.8. Tester la méthode précédente dans un bloc PL/SQL en testant les différents cas de figure (les exceptions traitées);

III.9. Coder la procédure AJOUTER\_COURS(N IN NUMBER, I IN varchar2, V IN NUMBER, LP IN T\_LISTE\_ENS)en traitant le cas de la violation de la primary key;

III.10. Tester la méthode précédente dans un bloc PL/SQL ;

III.11. Coder les autres méthodes qui restent et les tester.

## IV. MODIFICATION DU TYPE T\_PERS

Dans cette partie, on se propose d'enrichir le type T\_PERS par une méthode qui retourne la liste des cours assurés par une personne. Pour ce :

IV.1. Créer le type de tableau T\_LISTE\_COURS\_PROF de taille maximale 5 et de type de données NUMBER

IV.2. Ajouter la fonction LISTE\_COURS\_PROF RETURN T\_LISTE\_COURS\_PROF ;

IV.3. Coder la méthode précédente et recompiler le corps du type T\_PERS ;

IV.4. Tester la méthode précédente dans un bloc PL/SQL en affichant les cours assurés par la personne de CIN : X12434;