

## LE TYPE COLLECTION: NESTED TABLE (TABLE IMBRIQUEE)

On se propose dans ce TP de manipuler le deuxième type de collection proposé dans Oracle, soit NESTED TABLE. Pour ce, on reprend les mêmes types utilisés dans le précédent TP (donc prenez le même script et le modifier, la solution du TP3 est fournie). Et, on ajoute les nouveaux types selon le schéma suivant :

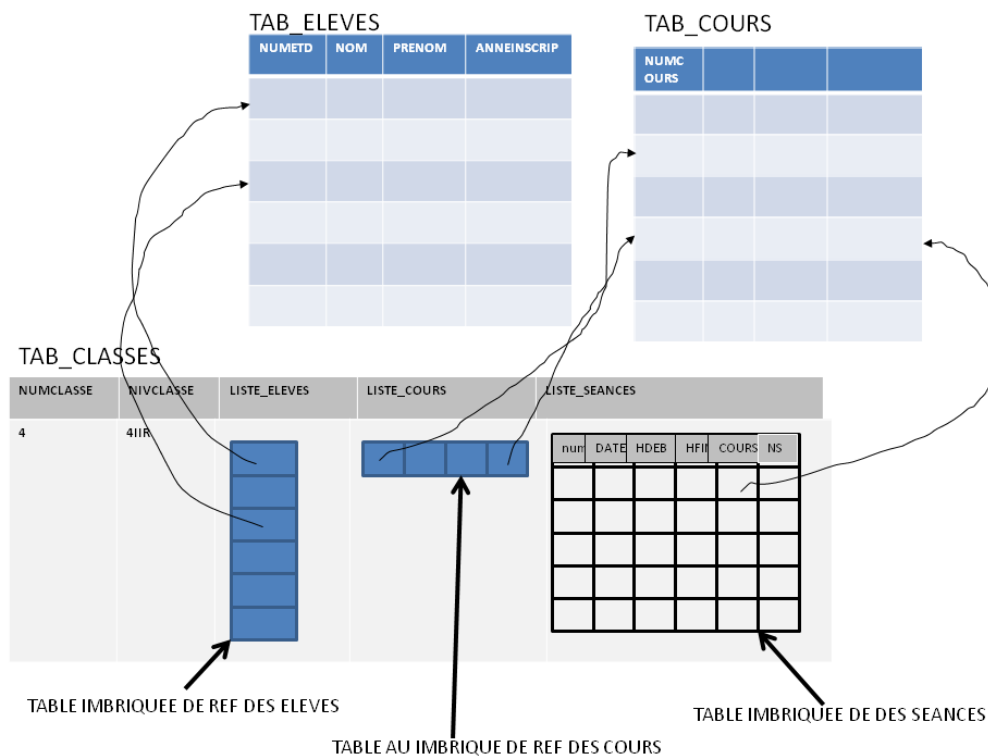


Schéma des types et des tables à ajouter

### I. CREATION DU TYPE T\_ELEVE

I.1. Créer le type T\_ELEVE défini par : NUMETD, NOM, PRENOM et ANNEINSCRIP (année d'inscription) de type respectivement de type NUMBER(10), VARCHAR2(20), VARCHAR2(20) et DATE.

I.2. Créer la table TAB\_ELEVES avec comme PRIMARY KEY le NUMETD et ANNEINSCRIP NOT NULL.

I.3. Insérer quelques éléments pour les tests.

### II. CREATION DU TYPE T\_SEANCE

II.2. Créer les type T\_SEANCE défini par : NUMSEAN, DATESEAN, HDEBUT, HFIN, COURS et NUMSALLE respectivement de type NUMBER(7), DATE, NUMBER(2), NUMBER(2), REF T\_COURS et NUMBER(2).

### III. CREATION DU TYPE T\_CLASSE

III.1. Créer le type T\_LISTE\_ELEVES de type table de référence sur T\_ELEVE ;

III.2. Créer le type T\_LISTE\_SEANCES de type table de T\_SEANCE;

III.3. Créer le type T\_LISTE\_COURS de type tableau de taille maximale 10 de référence sur T\_COURS ;

**III.4.** Créer le type T\_CLASSE défini par : NUMCLASSE, NIVCLASSE, LISTE\_ELEVES, LISTE\_COURS et LISTE\_SEANCES de type respectivement NUMBER(2), VARCHAR2(20), T\_LISTE\_ELEVES, T\_LISTE\_COURS et T\_LISTE\_SEANCES ;

**III.5.** Créer la table TAB\_CLASSES avec comme PRIMARY KEY le NUMCLASSE et NIVCLASSE NOT NULL.

**III.6.** Insérer des données dans la table classes.

#### **IV. MANIPULATION DE TABLE IMBRIQUEE PAR DES REQUETES SQL**

A partir de maintenant, les données à utiliser vous sont fournies pour les tests.

**IV.1.** Ajouter l'étudiant numéro 60 dans la classe numéro 4

**IV.2.** Tester l'insertion en affichant par une requête le numéro et le nom des élèves de la classe 4

**IV.3.** Mettre la date d'inscription de tous les étudiants de la classe 4 à 1/1/2010.

**IV.4.** Afficher, par une requête SQL, la liste des références des cours programmés et non prévus par pour la classe numéro 4.

**IV.5.** Afficher, par une requête SQL, la liste des références des cours prévus et non programmés pour la classe numéro 4.

**IV.6.** La même chose que dans mais affichant le numéro du cours et son intitulé

**IV.7.** Ajouter, pour la classe 4, la séance numéro 99 programmée aujourd'hui de 15h à 17h à la salle 88 pour le cours RES.

**IV.8.** Changer l'heure de fin de la séance 99 à 16h pour la classe 4.

**IV.9.** Supprimer la séance 99 de la classe 4.

#### **V. MANIPULATION DE TABLE IMBRIQUEE DANS DES BLOC PL/SQL**

**V.1.** Ecrire un bloc PL/SQL qui affiche la liste des étudiants d'une classe (classe numéro 4 par exemple) sans utiliser de cursor.

**V.2.** Faire la même chose que dans V.1. en utilisant un cursor.

**V.3.** Ecrire un bloc PL/SQL qui affiche le volume horaire des cours programmés (des séances) pour la classe 4.

**V.4.** Ecrire un bloc PL/SQL qui affiche le volume horaire globale des cours prévus (des séances) pour la classe 4.

#### **VI. LES METHODES POUR LE TYPE T\_CLASSE**

**VI.1.** Proposer des méthodes pour les différents types ajoutés dans ce TP, les coder et les tester.