

## TP 3 : LDD, Vues et PLSQL

### Partie 1 : SQL comme Langage de Description des Données (LDD)

#### A) CREATE, INSERT, UPDATE, ALTER, DELETE, DROP

On souhaite créer les schémas des relations suivantes :

**PERSONNE** (**nopers** number(6), **prenom** varchar2(20), **nom2fam** varchar2(25))

**SERVICE** (**noservice** number(4), **nomservice** varchar2(30), **noloc** number(6),  
**nochef** number(6))

**LOCALISATION** (**noloc** number(6), **rue** varchar2(40), **ville** varchar2(30))

**POSTE** (**noposte** varchar2(10), **nomposte** varchar2(35))

**SALAIRE** (**noposte** varchar2(10), **noservice** number(4,0), **salairemin** number(6),  
**salairemax** number(6))

**AFFECTATION** (**nopers** number(6), **noposte** varchar2(10), **noservice** number(4),  
**debservice** date, **finservice** date, **salaire** number(8))

Ces schémas sont complétés par les contraintes d'intégrité suivantes :

- les attributs soulignés sont les clés primaires des relations.
- les attributs en italique sont les clés étrangères
- le triplet (**nopers**, **noposte**, **noservice**) a une valeur **unique** pour chaque tuple d'AFFECTATION ; **noservice** peut être null contrairement aux deux premiers attributs
- l'attribut **nochef** dans SERVICE contient un numéro de personne (attribut **nopers** de PERSONNE)
- les valeurs des attributs suivants ne sont pas obligatoires : **prenom**, **salaire**, **nochef**, **rue**, **salairemin**, **salairemax**, **finservice** (et AFFECTATION.noservice)
- la **ville** par défaut dans la table LOCALISATION est 'Paris'
- pour une affectation donnée, la date de fin de service (**finservice**) est postérieure à celle de début de service (**debservice**)

1- Créer les schémas de ces relations.

2- Retrouver quelques informations sur le schéma de la table PERSONNE

- (a) En interrogeant la table USER\_TAB\_COLUMNS du dictionnaire, afficher le nom de chaque colonne (ou attribut), son type, la possibilité de valeur NULL (cf. attribut NULLABLE de la table USER\_TAB\_COLUMNS).

(b) Retrouver les mêmes informations en utilisant la fonction DESCRIBE.

3- Insérer dans les tables créées des données provenant de certaines tables HR

- (a) Insérer dans la table PERSONNE les informations utiles concernant tous les employés de la table EMPLOYEES de l'utilisateur HR (107 tuples).  
*N.B.* : first\_name correspond au prénom.
- (b) Peupler les tables SERVICE, LOCALISATION et POSTE à partir des tuples des tables HR.DEPARTMENTS, HR.LOCATIONS et HR.JOBS.
- (c) Initialiser la table SALAIRE à partir du produit cartésien des tables HR.JOBS et SERVICE. On supposera, dans un premier temps, que les salaires minimum et maximum ne dépendent que du numéro du poste (et pas du service) .
- (d) Peupler la table AFFECTATION à partir de la table HR.EMPLOYEES en initialisant à NULL la date de fin de service.

4- Mettre à jour ponctuellement les tables

- (a) Augmenter de 2% le salaire minimum pour tous les postes du service 'IT Support'.
- (b) Supprimer de la table SALAIRE les tuples concernant les postes dont le nom commence par 'Sales' dans les services dont le nom ne comporte pas 'sales'.
- (c) Supprimer le poste intitulé 'Purchasing Manager' de la table POSTE. Que se passe-t-il ? Renoncer à la mise à jour.

5- Ajouter un attribut **obsolète** à la table POSTE qui aura deux valeurs possibles **V** ou **F** selon qu'un poste est obsolète ou non. La valeur par défaut sera **F**.

6- Supprimer le schéma de toutes les tables créées.

## B) Vues

Sauf mention contraire, on considère à nouveau les tables de HR.

a- Créer une vue SALARIES\_SOUTHLAKE contenant certaines informations sur les salariés actuellement affectés dans les services localisés dans la ville de SOUTHLAKE<sup>1</sup>.

Le schéma de la vue est le suivant :

SALARIES\_SOUTHLAKE (NO\_SALARIE, NOM, PRENOM, NO\_POSTE, NOM\_SERVICE, ANCIENNETE)

L'ancienneté dans le service est exprimée en mois<sup>2</sup>.

b- Afficher le contenu actuel la vue SALARIES\_SOUTHLAKE, limité aux attributs NOM, PRENOM et ANCIENNETE, par ordre d'ancienneté décroissant.

c- Retrouver, pour chaque salarié de SOUTHLAKE, le nom et le prénom de son supérieur hiérarchique.

---

<sup>1</sup> fonctions UPPER, LOWER pour s'affranchir des problèmes de majuscule/minuscule dans la comparaison de chaînes

<sup>2</sup> fonctions FLOOR et MONTHS\_BETWEEN pour calculer l'ancienneté (cf. Cours ou Documentation)

## Partie 2 : PL/SQL

Sauf mention contraire, on considère les tables de HR

Remarque 1 : lorsqu'une erreur de compilation se produit en PL/SQL, les messages d'erreurs détaillés n'apparaissent pas immédiatement. Pour les obtenir, il faut exécuter la commande `SHOW ERRORS`.

Remarque 2 : pour activer l'affichage de message sur la console à partir de la commande `DBMS_OUTPUT_LINE` il faut préalablement donner la valeur `ON` à la variable `SERVEROUTPUT` avec la commande : `SET SERVEROUTPUT ON`

Rappel : L'appel d'une procédure peut se faire soit à partir d'un bloc PL/SQL selon la syntaxe :

```
NOM_PROCEDURE [ (PARAMETRES) ] ;
```

ou en dehors d'un bloc PL/SQL comme une commande du langage SQL selon la syntaxe :

```
EXECUTE NOM_PROCEDURE [ (PARAMETRES) ] ;
```

- 1) Créer un tableau T d'entiers. A l'aide d'une boucle, remplir T par les valeurs 2,4,6,...,30. Afficher le contenu de T dans l'ordre inverse (30, 28, ..., 2). Afficher aussi la taille du tableau T ou son cardinal.
- 2) Créer une fonction `CHARGES` qui retourne la valeur des charges pour un salaire donné en paramètre. Les charges représentent 45% du salaire.
- 2) Utiliser la fonction `CHARGES` pour afficher tous les montants des charges des salaires des employés actuellement en poste.

### PL/SQL – Curseurs - Procédures

- 1) A partir des tables de HR, afficher toutes les informations concernant les 5 employés ayant le salaire le plus élevé (bloc anonyme).
- 2) Transformer le bloc de la question a- pour en faire une procédure `TOP_N` qui affiche les noms et prénom des N salariés les mieux payés. N est un paramètre `IN` de la procédure. Penser à dérouter les cas où N n'est pas un nombre strictement positif.
- 3) Exécuter la procédure `TOP_N` pour N=5 puis N=11.