

TP 2

SQL - Définition de Données

Il faudra garder dans un fichier un compte-rendu du travail fait en TP (requêtes + résultats)

A- Définition de schémas pour une BD Ressources humaines

On souhaite créer les schémas des relations suivantes :

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

SERVICE (**noservice** number(4), **nomservice** varchar(30), **noLOC** number(6),
nochef number(6))

LOCALISATION (**noLOC** number(6), **rue** varchar(40), **ville** varchar(30))

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

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

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

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

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

- 1- Créer les schémas de ces relations avec toutes les contraintes.

2- Retrouver les informations suivantes sur le schéma de la table PERSONNE en utilisant la fonction DESCRIBE. Cette commande permet d'afficher le nom de chaque colonne (ou attribut), son type, la possibilité de valeur NULL (*nullable* ? yes/no)

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 certaines 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.
- (d) Ajouter un attribut COURRIEL à la table PERSONNE dont 'confidentiel' sera la valeur par défaut. Vérifier.
- (e) Renseigner l'attribut COURRIEL à partir de l'attribut EMAIL de la table HR.EMPLOYEES en complétant chaque valeur de email par la chaîne '@technissimo.fr'. Vérifier.

5- Supprimer le schéma de toutes les tables créées. Vérifier au moins pour une des tables.

B- Requêtes LMD + LDD sur une BD Projets (partie à finir seul(e))

Des relations ont été définies par les commandes suivantes :

```
create table employe(emp# varchar(10), nom_e varchar(20), grade char(3), chef#  
varchar(10)) ;
```

```
create table projet(proj# int, nom varchar(20), cout number, debut date, duree int) ;
```

```
create table financement(proj# int, organisme varchar(30), montant number) ;
```

```
create table affectation(emp# varchar(10), proj# int, charge number(5,2)) ;
```

```
create table direction(emp# varchar(10), proj# int) ;
```

Précisions :

- Les relations appartiennent à l'utilisateur **smail5** et elles sont interrogeables par tous.
- Aucune contrainte n'a été spécifiée lors de la création de ces relations.
- Dans la relation *employe*, *chef#* indique le supérieur hiérarchique immédiat.
- La relation *financement* indique le montant et la ou les sources de financement des projets (*projet.organisme*).
- La relation *direction* indique quels sont les chefs de projets, une personne pouvant diriger plusieurs projets et inversement (un projet peut avoir plusieurs chefs)
- La relation *affectation* indique, quant à elle, les participations des employés à des projets (un employé peut être affecté à plusieurs projets) ainsi que leur taux d'implication par projet (attribut *affectation.charge* qui prend ses valeurs dans l'intervalle [10..100]).

Définition des données

0- Copier les relations dans votre schéma par la commande

```
CREATE TABLE XX AS (SELECT * FROM SMAIL5.XX) ;
```

- 1- Ajouter aux schémas des relations, les contraintes de clés primaires et étrangère (commande ALTER TABLE)
- 2- Exprimer la contrainte suivante : *employe.grade* ∈ {'ANL', 'ING', 'CP', 'STG', 'PGM'}, signifiant, respectivement, *analyste*, *ingénieur*, *chef de projet*, *stagiaire*, *développeur*.

Requêtes SELECT

- 3- Combien de tuples comporte chacune des relations ?
- 4- Trouver les noms et les grades des personnes qui participent au projet numéro 1, ainsi que leur taux de participation à ce projet.
- 5- Quels sont les projets financés par les organismes 'CEE' et 'Conseil regional' (donner les noms des projets et les apports financiers respectifs) ?

- 6- Quels sont les projets dont le seul financeur est l'UL ? Pour cette question, il est conseillé de passer par une formulation logique (projet financé par l'UL et pour lequel il n'existe aucun autre organisme financeur).