

TP 2

SQL : PL/SQL

A. Lancez *lipn.univ-paris13.fr/~cabanes/INFO1/ecole.sql*

B. Contraintes

- 1. Essayez de modifier les tables pour ajouter les contraintes suivantes en SQL :**
 - La note d'un étudiant doit être comprise entre 0 et 20.
 - Le sexe d'un étudiant doit être dans la liste: 'm', 'M', 'f', 'F' ou Null.
 - *Contrainte horizontale* : Le salaire de base d'un professeur doit être inférieur au salaire actuel.
 - *Contrainte verticale* : Le salaire d'un professeur ne doit pas dépasser le double de la moyenne des salaires des enseignants de la même spécialité.
- 2. Que constatez-vous ?**

C. Triggers

- 1. Créez un trigger permettant de vérifier la contrainte : « Le salaire d'un Professeur ne peut pas diminuer ».**
- 2. Gestion automatique de la redondance**
 - Créez la table suivante : `CREATE TABLE PROF_SPECIALITE (SPECIALITE VARCHAR2 (20), NB_PROFESSEURS NUMBER) ;`
 - Créez un trigger permettant de remplir et mettre à jour automatiquement cette table suite à chaque opération de MAJ (insertion, suppression, modification) sur la table des professeurs.
 - Testez le trigger sur des exemples de mise à jour.

3. **Mise à jour en cascade** : Créez un trigger qui met à jour la table CHARGE lorsqu'on supprime un professeur dans la table PROFESSEUR ou que l'on change son numéro.

4. **Sécurité : enregistrement des accès**

- Créez la table *audit_resultats* :

```
CREATE TABLE AUDIT_RESULTATS (  
  UTILISATEUR VARCHAR2(50),  
  DATE_MAJ date,  
  DESC_MAJ VARCHAR2(20),  
  NUM_ELEVE NUMBER (4) NOT NULL,  
  NUM_COURS NUMBER (4) NOT NULL,  
  POINTS NUMBER);
```

- Créez un trigger qui met à jours la table *audit_resultats* à chaque modification de la table RÉSULTAT. Il faut donner l'utilisateur qui a fait la modification (USER), la date de la modification et une description de la modification ('INSERT', 'DELETE', 'NOUVEAU', 'ANCIEN'). Par exemple pour une insertion :

```
INSERT INTO audit_resultats  
VALUES (USER, SYSDATE, 'INSERT', :NEW.NUM_ELEVE,  
:NEW.NUM_COURS, :NEW.POINTS ) ;
```

5. **Confidentialité** : On souhaite que seul l'utilisateur 'GrandChef' puisse augmenter les salaires des professeurs de plus de 20%. Le trigger doit retourner une erreur (No -20002) et le message 'Modification interdite' si la condition n'est pas respectée.

D. Fonctions et procédures

1. Créez une fonction *fn_moyenne* calculant la moyenne d'un étudiant passé en paramètre.
2. Créez une procédure *pr_resultat* permettant d'afficher la moyenne de chaque élève avec la mention adéquate : échec, passable, assez bien, bien, très bien.
3. Créez un package contenant ces fonction et procédures