

École Polytechnique de Montréal
Département Génie Informatique et Génie Logiciel
INF3710 – Fichiers et Bases de données

TP 3 – Manipulation avancée de bases de données en SQL

Objectif: Manipulation avancée d'une base de données (vues, triggers, fonctions, SQL avancé)

1. Informations générales

<i>Pondération</i>	5%
<i>Taille de l'équipe</i>	2 personnes

Notez bien:

1. Tout retard dans la remise du TP entraîne automatiquement une pénalité comme discuté dans le plan de cours.
2. Aucun TP ne sera corrigé, s'il est soumis par une équipe dont la taille est différente **de deux (2) étudiants** sans l'approbation préalable du chargé de laboratoire. Cette approbation ne sera accordée à une équipe qu'en cas de nombre impair d'étudiants dans le laboratoire. Sinon, la note de zéro sera attribuée aux étudiants concernés.
3. Soumission du TP par **Moodle** uniquement (<https://moodle.polymtl.ca>). Aucune soumission "hors **Moodle**" ne sera corrigée. La note de zéro sera attribuée aux étudiants concernés.

2. Travail à faire

Partie 1 (Non notée)

Veuillez résoudre les exercices présentés dans TP3_Tutoriel.pdf.

Partie 2 (Notée)

Soit la base de données suivante (les PK sont soulignées)

Etudiant(**sid**, snom, sexe, age, moyenne)

Dept(**dID**, nombrephds)

Prof(**pID**, pnom, dep)
FK dep reference Dept(dID)

Cours(**cno**, cnom, dep)
FK dep reference dept(dID)

Section(**cno**, **sectno**, pID)

```
FK pID references Prof(pID)
FK cno references Cours(cno)

Inscription(sid, cno, sectno, note)
FK sid reference Etudiant(sid)
FK (cno, sectno) references section (cno,sectno)
```

On vous demande de :

1. (2 points) Créer la base de données *University* correspondante. Le script DDL de création de la BD doit se nommer `University_schema.sql`. Vous devez respecter les contraintes suivantes :
 - Les sections de cours doivent être automatiquement supprimées si un cours est supprimé
 - Lorsqu'un professeur est supprimé, sa valeur dans les tuples qui le référencent dans les sections doit être mise à null
 - On doit s'assurer que le sexe d'un étudiant est 'M' ou 'F' seulement
2. (1 point) Insérer des données dans toutes les tables. Votre script d'insertion des données doit se nommer `University_data.sql` (les données à mettre dans vos tables vous sont communiquées dans le fichier *data.txt*)
 - Que se passe-t-il si vous entrez un étudiant dans Enrollment avec un numéro de cours qui n'existe pas dans la table des cours ? (Mettez la réponse en commentaire dans votre fichier *University_data.sql*)
3. (14 points) Ecrire les requêtes suivantes. Votre script de requêtes SQL doit se nommer `University_query.sql`. Toutes vos requêtes doivent être précédées d'un commentaire qui reprend le texte de la requête en Français

Requêtes SQL :

- 1) (0.5 point) Retourner tous les étudiants par ordre croissant sur leur nom
- 2) (0.5 point) Retourner le nom des professeurs et leur département. Nommez les colonnes Professeur et Dep
- 3) (1 point) Retourner le nom des professeurs qui travaillent dans un département de plus de 10 PHDs
- 4) (1 point) Retourner la plus haute note du cours '105' (toutes sections confondues)
- 5) (1 point) Retourner la plus haute note du cours '105' par section
- 6) (1 point) Pour chaque cours, retourner le nombre d'étudiants par section pour les sections de plus d'un étudiant (exemple : *c3, s1, 20*)
- 7) (1 point) Retourner les infos des étudiants et de leurs inscriptions. La table doit également comprendre les étudiants qui ne sont inscrits dans aucun cours
- 8) (1 point) Retourner l'info des étudiants qui ne sont inscrits à aucun cours - Utilisez une sous-requête
- 9) (1 point) Imprimer les informations des cours qui parlent de géométrie (toutes les combinaisons de titres possibles)

- 10) (1 point) Imprimer le nom des étudiants qui suivent un cours de géométrie (toutes les combinaisons de titres possibles)
 - 11) (1 point) Imprimer le nom des étudiants qui sont inscrits à au moins un cours du département GIGL et au moins un cours du département de mathématiques - Utilisez INTERSECT
 - 12) (1 point) Imprimer le nom des étudiants qui suivent un cours du département GIGL OU un cours du département de mathématiques
 - 13) (1 point) Quelle est la différence d'âge entre le plus vieux et le plus jeune étudiant ? Affichez le résultat dans une colonne nommée Difference
 - 14) (1 point) Quel est le nombre d'étudiants dont la moyenne est supérieure à la moyenne de tous les étudiants ?
 - 15) (1 point) Quels sont le ou les étudiants avec la plus grande moyenne ? Affichez le nom des étudiants et leur moyenne
4. Triggers (3 points)
- Créez un fichier *University_trigger_function.sql* qui contient les éléments suivants :
- 1) (0.5 points) Une fonction *nbEtudiants* en PL/pgSQL qui retourne le nombre total d'étudiants;
 - 2) (2.5 points) Un trigger *auditGrade* qui crée un enregistrement dans une table archive Audit. Ce trigger se déclenche lorsque la note du cours est mise à jour. L'archive devra conserver les informations d'inscription ainsi que la date où la modification de la note du cours a été effectuée ;

5. Modalités de remise

Vous devez soumettre sur Moodle un fichier zip *matricule1_matricule2_TP3.zip* contenant

- a. Le fichier *University_Schema.sql*
- b. Le fichier *University_data.sql*
- c. Le fichier *University_query.sql*
- d. Le fichier *University_trigger_function.sql*