

Ingénierie des systèmes d'information.

TD 3 – Contraintes d'intégrités, triggers et vues

Manon Ansart

ESIREM, 2022

Exercice 1

Soient les tables suivantes :

- GROUPE(ID, SPECIALITE, NUM_TD)
- ENSEIGNANT(ID, NOM, PRENOM)
- COURS(ID, GROUPE, ENSEIGNANT, HEURE_DEBUT, DUREE, NB_ETUDIANTS)
- ETUDIANT(ID, NOM, PRENOM, GROUPE)

1. Quelles sont les clés des différentes tables ?
2. Ces tables sont-elles en 2ème forme normale ? 3ème forme normale ?
3. Ecrivez les requêtes SQL permettant de créer ces tables, en ajoutant toutes les contraintes d'intégrité appropriées.
 - On considère qu'il faut au moins 10 étudiants dans un cours
4. Ajoutez un trigger qui, à l'insertion d'un étudiant, met à jour le nombre d'étudiants dans les cours correspondants
5. Ajoutez un trigger qui, à la suppression d'un groupe, supprime les cours de ce groupe
6. Quels problèmes voyez-vous avec ce schéma ? Quel impact y a-t-il en terme de cohérence, robustesse et disponibilité ?

Exercice 2

On décide de supprimer l'attribut NB_ETUDIANTS de la relation COURS, qui s'écrit donc maintenant COURS(ID, GROUPE, ENSEIGNANT, HEURE_DEBUT, DUREE)

1. Proposez une requête permettant de calculer le nombre d'étudiants par groupe
2. Proposez une requête permettant de calculer le nombre d'étudiants par cours
3. Créer une vue calculant le nombre d'étudiants de chaque cours
4. Que pensez vous de cette solution en terme de cohérence, robustesse et disponibilité ?

Exercice 3

1. Créez une vue permettant d'afficher les groupes SQR
2. Créez une vue permettant d'afficher les cours des SQR
3. Comptez le nombre d'heures par spécialité
4. Comptez le nombre d'heures que chaque enseignant passe avec les SQR de 2 manières différentes
5. Bonus : Retournez le nom et prénom des enseignants donnant des cours à des groupes de plus de 20 personnes