

TD N 1 - Le langage SQL

Partie 1

Schéma de la base de données

On veut établir un classement des films, et pour cela on va collecter les données provenant des examinateurs classant les différents films. Le schéma de la base de données est le suivant :

Movie (mID, title, year, director)

Chaque film possède un identificateur ID, un titre, l'année de production et un réalisateur.

Reviewer (rID, name)

Un examinateur possède un identificateur ID et un nom.

Rating (rID, mID, stars, ratingDate)

Pour le classement des films, l'examineur **rID** donne un nombre d'étoiles (1-5) au film **mID** dans une date donnée.

Exercice 1

1. Ecrire une requête qui va créer la table « Movie » avec des contraintes convenables.
2. Ecrire une requête qui va créer la table « Reviewer » avec des contraintes convenables.
3. Ecrire une requête qui va créer la table « Rating » avec des contraintes convenables.

Exercice 2

1. Trouver les titres des films réalisés par « James Cameron »
2. Trouver les années possédant un film qui a reçu un classement de 4 ou 5 étoiles, en les triant par ordre décroissant.
3. Trouver les titres des films qui n'ont pas reçu un classement
4. Quelques examinateurs n'ont pas fourni une date avec leurs classement. Trouver les noms de tous les examinateurs qui n'ont pas fourni une date avec leurs classement.
5. Ecrire une requête qui retourne les données classées dans un format plus lisible : nom de l'examineur, titre du film, étoiles et la date de classement. Trier les données, premièrement par nom de l'examineur, ensuite par titre du film et finalement par le nombre des étoiles reçu.
6. Pour les cas où le même examinateur a classé deux fois le même film en donnant un classement supérieur dans la deuxième fois, trouver le nom de l'examineur et le nom du film (challenge).
7. Pour chaque film qui possède au moins un classement, trouver le nombre maximal des étoiles reçu par ce film. Afficher le titre du film et le nombre des étoiles. Trier les résultats par titre des films.

8. Pour chaque film, afficher le titre et la 'propagation du classement', c'est-à-dire, la différence entre le classement maximal et minimal donné pour ce film. Trier les résultats par ordre décroissant selon la 'propagation du classement'.

Exercice 3

1. Trouver les noms de tous les examinateurs qui ont classés *'Gone with the Wind'*.
2. Pour chaque classement dont l'examineur est le réalisateur du film, retourne le nom de l'examineur, le titre du film et le nombre des étoiles reçues.
3. Trouver à la fois tous les examinateurs et les titres des films dans une seule liste, classée par ordre alphabétique.
4. Trouver les titres des films qui ne sont pas examinés par *'Chris Jackson'*.
5. Pour tous les paires des examinateurs tels que les deux examinateurs classent le même film, retourne les noms des deux examinateurs. Eliminer les doublons, n'associez pas les examinateurs avec eux-mêmes, inclure chaque paire une seule fois. Pour chaque paire, retourne les noms par ordre alphabétique.
6. Pour chaque classement dont le nombre des étoiles est le plus bas dans la base de données, retourne le nom de l'examineur, le nom du film et le nombre des étoiles.
7. Trouver les titres des films et la moyenne des classements, du plus haut au plus bas classement. Si deux ou plus de films possèdent la même moyenne de classement, en procède à un classement par ordre alphabétique.
8. Trouver les noms de tous les examinateurs qui ont contribué à trois ou plus de classements.
9. Quelques réalisateurs ont réalisés plus d'un film. Pour tous ces réalisateurs, trouver les titres de tous les films qui ont été réalisés avec le nom du réalisateur. Classer les résultats par nom, ensuite par le titre du film.
10. Pour chaque réalisateur, trouver son nom à la fois avec le titre(s) du film(s) réalisés et ont reçu le plus haut classement parmi tous ses films, et la valeur de ce classement. On ignore les films dont le réalisateur est NULL.

Exercice 4

1. Ajouter à la base de données l'examineur *'Roger Ebert'* avec un rID de 209.
2. Insérer dans la base de données un classement de 5-étoiles donné par James Cameron pour tous les films. Laissez le champ *'date'* NULL.
3. Pour tous les films possédant une moyenne de classement de 4-étoiles ou plus, ajouter 25 à l'année de production (mettre à jour les tuples).
4. Supprimer tous les classements des films dont l'année de production est avant 1970 ou après 2000, et le classement est moins de 4 étoiles.

Partie 2

Schéma de la base de données

Les étudiants du '*Génie Informatique*' à l'EST d'Agadir ont décidé d'organiser leurs réseaux sociaux en utilisant les bases de données. Ils ont collecté des informations concernant 16 étudiants chacun possède un identificateur, un nom et une note. Le schéma de la base de données est le suivant :

Student (ID, name, mark)

Un étudiant possède un identificateur ID, un nom et une note.

Friend (ID1, ID2)

L'étudiant avec l'identificateur ID1 est un ami de l'étudiant dont l'identificateur est ID2. Notons que si (123, 456) figure dans la table *Friend*, il est de même pour (456, 123).

Likes (ID1, ID2)

L'étudiant avec un identificateur ID1 aime l'étudiant avec un identificateur ID2. Notons que ce n'est pas nécessaire si (123, 456) figure dans la table *Likes*, (456, 123) figure aussi.

Exercice 5

1. Trouver les noms des étudiants qui sont des amis avec 'Gabriel'.
2. Pour chaque étudiant qui aime un autre qui possède une note inférieure de 2 ou plus, retourner le nom et la note de cet étudiant, ainsi que le nom et la note de l'étudiant qui aime.
3. Pour chaque paire d'étudiants dont l'un aime l'autre, afficher le nom et la note des deux étudiants. Afficher chaque paire une seule fois en ordre alphabétique.
4. Trouver tous les étudiants qui n'apparaissent pas dans la table '*Likes*' et retourner leurs noms et leurs notes. Trier les résultats par note, ensuite par nom.
5. Pour chaque situation dont un étudiant A aime un étudiant B, mais on ne possède aucune information sur les étudiants aimés par l'étudiant B (c'est-à-dire, B n'apparaît pas comme ID1 dans la table '*Likes*'), retourner les noms et les notes de A et B.
6. Pour chaque étudiant A qui aime un étudiant B dont les deux ne sont pas des amis, trouver s'ils possèdent un ami C en commun. Pour ces trios, afficher le nom et la note de A, B et C.

7. Trouver la différence entre le nombre des étudiants à l'école et le nombre de différents noms.
8. Trouver le nom et la note de tous les étudiants qui sont aimés par plus d'un autre étudiant.