

Le langage SQL – TD N 1

Partie 1

Schéma de la base de données

Movie (mID, title, year, director)

Reviewer (rID, name)

Rating (rID, mID, stars, ratingDate)

Exercice 1

Créer la base de données.

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.
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 4

Question 1

Ajouter à la base de données l'examineur 'Roger Ebert' avec un rID de 209.

Question 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.

Question 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).

Question 4

Supprimer tous les classement des films dont l'année de production est avant 1970 ou après 2000, et le classement est moins de 4 étoiles.

Partie 2

Student (ID, name, mark)

Friend (ID1, ID2)

Likes (ID1, ID2)

Exercice 5

1. Trouver les noms des étudiants qui sont des amis avec 'Gabriel'.
2. Pour chaque étudiant qui aime un autre dont le dernier 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 pair des étudiants dont l'un aime l'autre, afficher le nom et la note des deux étudiants. Afficher chaque pair 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 ce 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.