Bases de Données
SQL - définition et manipulation de données

Travaux Pratiques

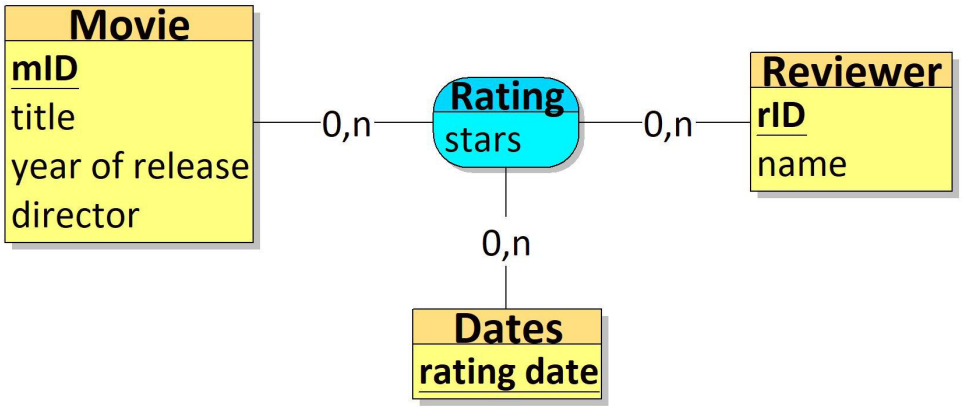
18 septembre 2023

Exercice 1 : création de la base

Vous souhaitez créer un site d'appréciation de films. On gère les films identifiés par un numéro, qui ont un titre obligatoire, une date de réalisation et un réalisateur. Les évaluateurs ont un identifiant unique et obligatoirement un nom. Chaque évaluation est définie par le évaluateur, le film et la date d'évaluation ; elle donne lieu à une note (nombre d'étoiles). Il ne peut pas y avoir d'évaluation sans note.

Précision : pour simplifier, le nom du réalisateur sera une simple information pour chaque film et ne représentera pas une entité de l'application.

Le schéma Entité/Association ci-dessous est proposé pour représenter ces données.



Après traduction du diagramme EA et simplification (on a supprimé la relation 'Date' qui n'apportait rien), on obtient le schéma relationnel suivant.

- *movies*(*id_movie*, *title_movie*, *year_movie*, *director_movie*)
- *reviewers*(*id_reviewer*, *name_reviewer*)
- *ratings*(*#id_reviewer*, *#id_movie*, *date_rating*, *stars_rating*)

1. Créez cette base de données en utilisant la commande SQL **CREATE TABLE** (...) pour chaque relation. Par exemple celui de la relation "Reviewer" est :

```
CREATE TABLE Reviewer(
  id_reviewer integer,
  name_reviewer varchar(30) NOT NULL,
  PRIMARY KEY (id_reviewer)
  );
```

Pensez à créer vos clés étrangères avec la commande sur le modèle suivant :

```
CONSTRAINT nom-contrainte FOREIGN KEY(attribut_source)
                          REFERENCES table_cible(attribut_cible)
```

2. Avec une commande de type **ALTER TABLE** ... **ADD CONSTRAINT** ..., déclarez une contrainte pour la relation "Movie" permettant de vérifier que la même année, deux films ne peuvent pas avoir le même titre.
3. Copiez-collez le script TP1_moviedata.sql (voir sur la page WEB de l'UE) qui insère des tuples dans cette base de données.

Exercice 2 : requêtes simples

Exprimer en SQL les requêtes permettant de retrouver les informations suivantes :

1. Le nom du relecteur 205.
2. Les titres de films.
3. Les titres de films par ordre croissant.
4. Les films réalisés par Steven Spielberg.
5. Les titres de films dont le réalisateur n'est pas renseigné
6. Les évaluations, avec les informations sur les films et les réalisateurs issus des tables correspondantes. Nommez cette requête par un alias (vue) "v_detail_evaluations", que vous utiliserez comme une relation dans la suite à chaque fois que vous en aurez besoin.
   Indication : Utilisez la commande **CREATE VIEW AS** (requeteSQL). Cette vue est la jointure naturelle des trois relations, avec tous les tuples "reconstitués" grâce aux clés étrangères.
7. Les années, dans l'ordre croissant, qui ont un film qui a reçu une note de 4 ou 5.
8. Le nom des personnes qui ont noté le film Gone with the Wind.
9. L'intégralité des évaluations, avec un résultat sous la forme (nom de l'examinateur, titre du film, nombre d'étoiles). Trier le résultat, d'abord par le nom de relecteur, puis par le titre de film, et enfin par le nombre d'étoiles.
10. Pour chaque évaluation où l'examinateur est identique au réalisateur du film (même nom), le nom de l'examinateur, le titre du film, et le nombre d'étoiles.
11. Les titres des films non encore examinés par Chris Jackson.
12. Pour tous les cas où la même personne note deux fois le même film et donne une note plus élevée la seconde fois, le nom de l'examinateur et le titre du film.

Exercice 3 : Fonctions agrégatives

Les fonctions agrégatives calculent une valeur unique à partir d'un ensemble de tuples, par exemple : ***max****(x),* ***min****(x),* ***count****(x),* ***avg****(x), ... Elles le font sur l'ensemble des tuples retournés par les clauses* ***FROM*** *...* ***WHERE*** *..., ou bien sur chaque groupe formé par l'utilisation de* ***GROUP BY****.*

1. Retourner le nom de l'examinateur, le titre du film, et le nombre d'étoiles pour tous les films qui ont actuellement la plus mauvaise note dans la base.
2. Pour chaque film, trouver la meilleure note reçue. Retourner le titre de film et le nombre d'étoiles. Trier par rapport au titre de film (ordre alphabétique).
3. Donnez le nom des évaluateurs qui ont évalué tous les films.
4. Lister les titres de films, leur note moyenne, leur meilleure et plus mauvaise note.
5. Trouver le nom de tous les examinateurs qui ont fait au moins 3 évaluations.
6. Trouver le(s) film(s) ayant la meilleure moyenne de note. Retourner le titre de film, et sa note moyenne.
7. Pour chaque film, donner sa meilleure note et le nom du rapporteur qui l'a donnée (une ligne par meilleure note s'il elle a été donnée plusieurs fois). Ordonnez sur les titres de films.
8. La différence entre la note moyenne des films réalisés avant 1980 et ceux réalisés à partir de 1980. [1]
9. Pour chaque film, retourner le titre et la différence entre la meilleure et la plus mauvaise note. Trier par rapport à cette amplitude puis en fonction du titre.

1. On veut la moyenne des notes moyennes de chaque film.

Exercice 4 : Fonctions de fenêtrage

*Les fonctions de fenêtrages sont des fonctions agrégatives, mais directement suivies de la clause **OVER** (). Cela a pour effet que la fonction n'est pas évaluée sur les groupes des blocs **FROM** ... **WHERE** ...**GROUP BY** ..., mais sur la partition "à la volée" définie dans la clause **OVER**.*

1. Pour chaque évaluation, retourner le nom du reviewer, le titre du film évalué, la note attribué au film par le reviewer et la moyenne de toutes les notes attribuées par ce reviewer.
2. Pour chaque évaluateur, donnez le nombre de films rapportés, la moyenne des notes qu'il a données, la plus petite et la meilleure note. Ordonnez le résultat par le nombre de films rapportés.
   Indication : Utilisez une clause **WINDOW** w **AS** ... en fin de requête qui définit une partition ; vous pourrez utiliser cette partition dans la clause **SELECT** sans avoir à la ré-écrire.
3. Pour chaque film, afficher sa note moyenne et son rang de classement dans l'ordre des notes moyennes parmi l'ensemble des films.
4. Pour chaque évaluateur, et chaque film qu'il a évalué, afficher la meilleure note qu'il a donnée à ce film, ainsi que la moyenne de toutes les notes qui ont été données à ce film. La réponse sera ordonnée par nom de évaluateur et titre de film.

Exercice 5 : Parcours de cubes

*Lorsqu'on analyse une grandeur, on peut vouloir le faire selon différentes combinaisons d'attributs. Ainsi, on peut ajouter des commandes particulières à **GROUP BY** permettant d'explorer plusieurs combinaisons dans la même requête - avec les mots clés **GROUPING SETS**, **CUBE** ou **ROLLUP**.*

1. Donner la moyenne des notes pour chaque évaluateur, ainsi que pour chaque film qu'il a rapporté.
2. Même question, mais pour les lignes correspondant à tous les films d'un évaluateur, remplacer la valeur NULL par "TOUS LES FILMS" (utilisation de la fonction **coalesce**).
3. Analyser la moyenne des notes selon les trois dimensions film, évaluateur et année d'évaluation. Le résultat devra être défini sur les attributs titre du film, nom du évaluateur et année.