

Rapport

Projet Base de Données

-

Netflop

Réalisé par Nicolas Deronsart

I/ Description du projet et de ses objectifs

Pour la réalisation de ce projet, j'ai créé avec sqlite3 la base de données d'une nouvelle plateforme de streaming fictive nommée Netflop, ainsi qu'une interface graphique en python pour interagir avec elle de manière simplifiée.

Cette base de données permet de gérer les différentes œuvres (films et séries) proposées par la plateforme, ainsi que les différents acteurs, réalisateurs, studios ou chaînes, et entreprises impliqués dans leur création.

Elle permet également de répertorier les utilisateurs et les différentes œuvres qu'ils ont vues et notées, afin de pouvoir leur recommander plus efficacement d'autres œuvres de la plateforme.

L'interface est alors organisée en 3 parties, la première permet de parcourir les différentes tables de la base de données, la deuxième permet d'ajouter et de retirer des éléments, et la troisième propose des œuvres différentes selon les préférences de chaque utilisateur. Toutes ces fonctionnalités sont permises grâce à des requêtes SQL.

Dans la réalisation de ce projet j'ai tout d'abord conçu le modèle entité-association, représentant la base de données, ainsi que son schéma relationnel pour pouvoir écrire dans un script sql la création de toutes les tables nécessaires.

J'ai de plus créé plusieurs fichiers .csv dans lesquels j'ai répertorié films, séries, acteurs, réalisateurs, studios, chaînes et entreprises, et créé quelques utilisateurs, qui seront lus et ajoutés à la base de donnée pour permettre d'avoir une base déjà remplie lorsque l'on ouvre l'interface.

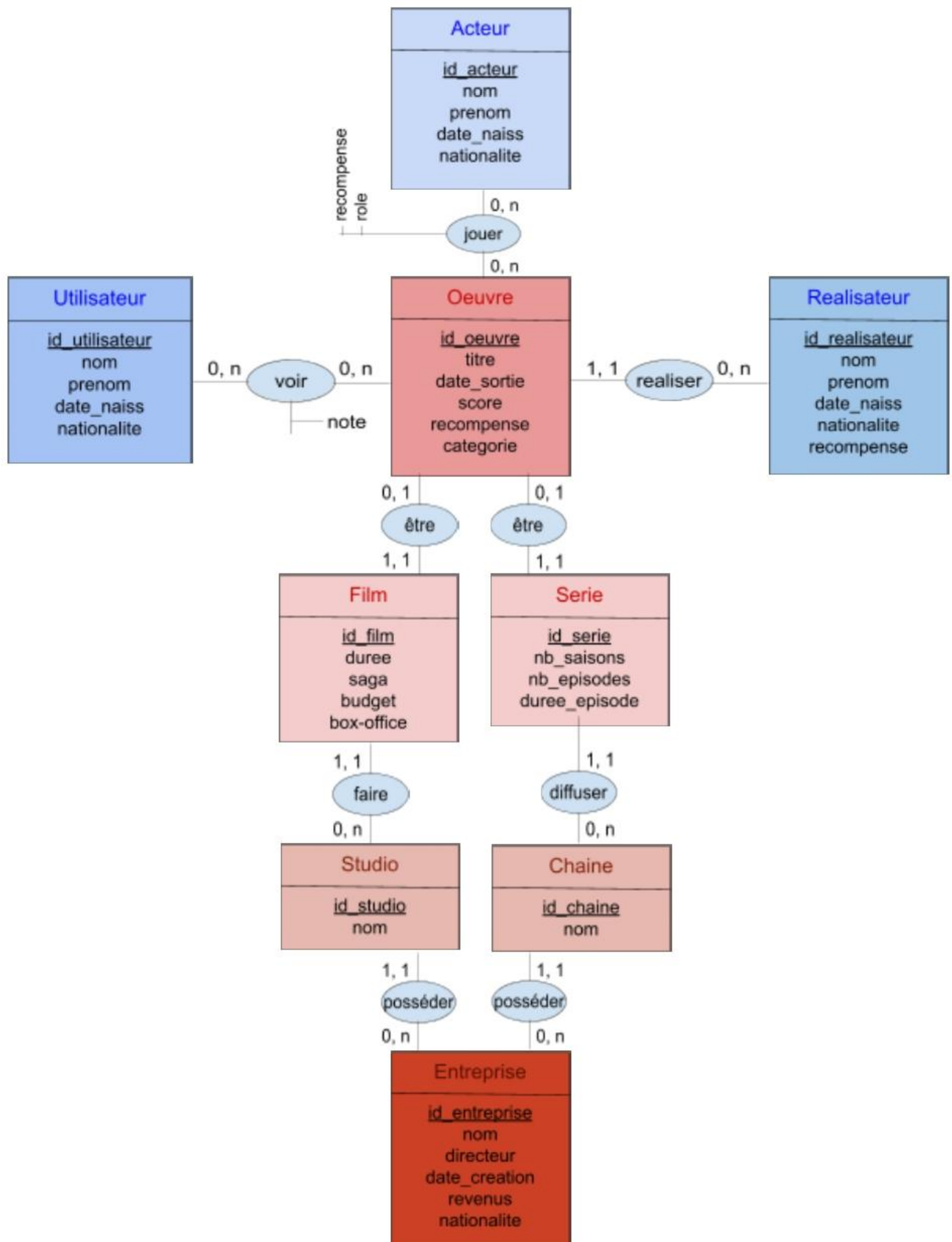
Des vues et déclencheurs sont également ajoutés à la base de données, à l'aide de scripts SQL, dès sa création.

Les vues permettent d'afficher les films ou séries créés par Netflop, les meilleures ou pires œuvres selon les notes données par la presse, le meilleur film ou la meilleure série selon les utilisateurs, ou encore quels acteurs ont été récompensés pour quelles œuvres. Toutes ces vues sont utilisées dans l'interface.

Nous reviendrons plus tard dans ce rapport sur les différents déclencheurs.

Vous pourrez trouver dans le dossier application les différents scripts SQL utilisés, les fichiers csv et le fichier netflop.py qui lance l'interface graphique.

II/ Modèle entité association



III/ Schéma relationnel

Voici le schéma relationnel de ma base de données, tiré du modèle entité-association ci-dessus.

acteur(id_acteur, nom, prenom, date_naiss, nationalite)
utilisateur(id_utilisateur, nom, prenom, date_naiss, nationalite)
realisateur(id_realisateur, nom, prenom, date_naiss, nationalite, recompense)
oeuvre(id_oeuvre, titre, date_sortie, score, recompense, categorie, id_realisateur*)
jouer(id_acteur*, id_oeuvre*, role, recompense)
voir(id_utilisateur*, id_oeuvre*, note)
entreprise(id_entreprise, nom, directeur, date_creation, revenus, nationalite)
studio(id_studio, nom, id_entreprise*)
chaine(id_chaine, nom, id_entreprise*)
film(id_film, duree, saga, budget, box_office, id_oeuvre*, id_studio*)
serie(id_serie, nb_saisons, nb_episodes, duree_episode, id_oeuvre*, id_chaine*)

Types des attributs :

- integer
- varchar
- date

Clés :

- Primaires → nom_attribut
- Étrangères → nom_attribut*

Contraintes:

- Tous les attributs ont une contrainte “not NULL”, à l'exception de l'attribut **saga**.
- Parmi les entiers les attributs **recompense** se trouvent toujours dans l'intervalle [0, 1], **note** et **score** sont dans l'intervalle [0, 5], **duree**, **budget**, **box_office**, **nb_saisons**, **nb_episodes** et **duree_episode** sont supérieurs ou égaux à 1, et **revenus** est supérieur ou égal à 0.
- Les chaînes de caractères ont une taille maximale de 30 et l'attribut **role** doit être soit “first”, “second” ou “antagonist”.
- Les dates doivent être comprises entre 1900 et 2022, et sont toutes du format YYYY-MM-DD (année-mois-jour).

IV/ Déclencheurs (signature et signification)

- **suppression_voir** permet, lorsqu'il y a suppression d'un utilisateur, de supprimer tous les éléments de la table voir impliquant cet utilisateur supprimé.
- **suppression_jouer** permet, lorsqu'il y a suppression d'un acteur, de supprimer tous les éléments de la table jouer impliquant cet acteur supprimé.
- **maj_recompense_oeuvre** permet, lorsque l'on ajoute un élément à la table jouer, de mettre à jour la récompense de l'œuvre à vraie (1) si l'acteur a reçu une récompense pour ce rôle et si cette œuvre n'avait pas de récompense pour le moment (recompense=0).
- **supprime_film_serie** permet, lorsque l'on supprime une oeuvre, de supprimer le film ou la serie correspondant à cette oeuvre supprimée.
- **realisateur_inconnu** permet, lorsque l'on supprime un realisateur, de changer l'id_realisateur pour toutes ses œuvres par l'id 0 qui renvoie vers John Doe (nom utilisé pour nommer une personne inconnue aux Etats-Unis).