

Le sujet comprend 3 parties.

1 Vidéos à la demande (donner les réponses sur le sujet que vous joindrez à votre copie)

Dans tous les exercices suivants, on fixe une base de données utilisée par un service de streaming de séries télévisées. Le service permet aux utilisateurs de regarder les différents épisodes qui constituent les séries et éventuellement de noter les épisodes qu'ils ont vus.

La base de données respecte le schéma ci-dessous, où les clefs primaires sont soulignées et les clefs étrangères listées à la suite.

```
serie(idS, nom, genre, agemin)
episode(idE, titre, idS, numero, duree)
utilisateur(pseudo, mdp, email, age)
regarde(pseudo, idE, note)
```

FK : idS dans `episode` fait référence à idS dans `serie` ;

FK : pseudo dans `regarde` fait référence à pseudo dans `utilisateur` ;

FK : idE dans `regarde` fait référence à idE dans `episode`.

La table `utilisateur` contient les utilisateurs inscrits sur le service, avec leur pseudo-nyme, mot de passe, adresse e-mail et âge. La table `serie` regroupe les informations sur les séries disponibles sur la plateforme : nom, genre et âge minimum recommandé. La table `episode` contient les épisodes de chaque série. Chaque épisode a un titre et un numéro au sein de la série dont il fait partie, et on connaît de plus sa durée en minutes.

Par exemple, la série *The Good Place* est une *comédie* recommandée pour les spectateurs de 13 ans et plus. Son épisode numéro 1 est intitulé *Everything Is Fine* et dure 24 minutes.

Finalement, la table `regarde` indique quand l'utilisateur `pseudo` a vu l'épisode `idE` et la note qu'il lui a donnée.

Les exercices sont très largement indépendants.

Exercice 1 : Lecture du schéma relationnel

Pour chacune des questions suivantes, écrivez **autorisé** si le schéma permet les situations décrites et **interdit** sinon.

1) Deux utilisateurs ont la même adresse email.

2) Une série ne contient aucun épisode.

3) Un épisode n'a pas de titre.

4) Une série a un épisode numéro 2 mais pas d'épisode numéro 1.

- 5) Un utilisateur regarde un épisode d'une série alors qu'il n'a pas l'âge requis.

- 6) Un utilisateur n'a pas de pseudonyme.

Exercice 2 : création des tables

Pour chacune des questions suivantes, dites quels choix faire (contraintes, types de données, etc.) à la création des tables pour obtenir les comportements voulus.

La syntaxe SQL précise n'est pas obligatoire : indiquez seulement la nature du choix, et les tables et attributs concernés.

- 1) Un utilisateur ne peut pas avoir de mot de passe identique à son pseudonyme.

- 2) Les pseudonymes des utilisateurs sont d'une longueur maximum de 25 caractères.

- 3) Le titre et la durée de chaque épisode sont forcément renseignés.

- 4) Une même série ne peut avoir deux épisodes portant le même numéro.

- 5) Lorsqu'une série est retirée du système, tous les épisodes de la série sont automatiquement supprimés.

Exercice 3 : requêtes SQL

Proposez des requêtes SQL permettant d'extraire les informations demandées.

- 1) La liste des genres de série, sans doublons.

- 2) La liste des utilisateurs qui n'ont pas renseigné d'adresse email.

--

- 3) Les utilisateurs qui ont regardé un épisode d'une série alors qu'ils n'avaient pas l'âge.

--

- 4) la liste des séries, avec pour chacune son nom et son nombre total d'épisodes, triée par nombre total d'épisodes croissant.

--

- 5) Les séries de science fiction qui contiennent au moins 30 épisodes.

--

- 6) Les utilisateurs qui ont regardé un épisode d'une série alors qu'ils n'ont pas vu l'épisode précédent.

--

- 7) La liste des séries, avec pour chacune son nom et le titre de son épisode le mieux noté en moyenne.

Exercice 3 : requêtes SQL

Proposez des requêtes d'algèbre relationnelle permettant d'extraire les informations demandées.

- 1) Le titre du premier épisode de la série "Rome".

- 2) Les utilisateurs qui n'ont jamais attribué la note 0 à un épisode.

- 3) Les utilisateurs (**pseudo**) qui ont vu tous les épisodes de toutes les séries.
(Indice : si vous avez compris l'opération algébrique de division, c'est le moment de l'utiliser. Il est également possible d'écrire la requête sans cet opérateur.)

2 Modélisation

Un service de jeux de société en ligne utilise une base de données pour stocker les informations sur les jeux, gérer les profils des joueurs et leur permettre d'organiser leurs parties.

Les joueurs peuvent s'inscrire en fournissant un pseudonyme unique, une adresse e-mail et un mot de passe, et le système retient la date de leur inscription. Une fois inscrits, les joueurs peuvent consulter la liste des jeux disponibles sur le service, où sont renseignées les informations suivantes : titre du jeu, nombre de joueurs minimal et maximal par partie, et texte des règles. Le service indique également lorsqu'un jeu est une variante d'un autre jeu, et un même jeu peut avoir plusieurs variantes. Les joueurs peuvent indiquer les jeux qu'ils connaissent, et dans ce cas, laisser leur avis sous la forme d'un commentaire textuel et d'une note.

Les joueurs peuvent utiliser le service pour organiser leurs parties. Pour chaque partie, on connaît le jeu utilisé, le joueur qui l'organise, la date à laquelle elle aura lieu, et son statut : 'ouverte', 'en cours' ou 'terminée'. Les joueurs peuvent participer aux parties créées et les organisateurs sont libres de participer ou non à leurs propres parties. Finalement, à la fin de chaque partie, le résultat de chaque participant est enregistré sous la forme d'un classement et d'un score.

1. Proposer une modélisation E/R pour décrire l'activité de ce service de jeux de société en ligne. On précisera les entités, les clefs primaires, les associations, on indiquera pour chaque association les cardinalités.
2. Proposer une traduction dans le modèle relationnel. On précisera les clefs primaires et les contraintes référentielles (clefs étrangères).