

Exercice 2 (6 points)

Cet exercice porte sur les bases de données relationnelles, le langage SQL et la programmation.

Une ludothèque municipale a décidé de moderniser sa gestion en créant une base de données informatique. Cette base de données permettra de suivre les jeux disponibles, les emprunts effectués par les adhérents, ainsi que les avis laissés sur les différents jeux. Pour commencer, quatre tables principales ont été identifiées : `jeu`, `adhérent`, `emprunt` et `avis`. Ces tables et leurs relations vont permettre de stocker toutes les informations essentielles au bon fonctionnement de la ludothèque. On va considérer que la ludothèque n'a **qu'un exemplaire** de chaque jeu (deux jeux de la ludothèque ne peuvent donc pas avoir le même nom).

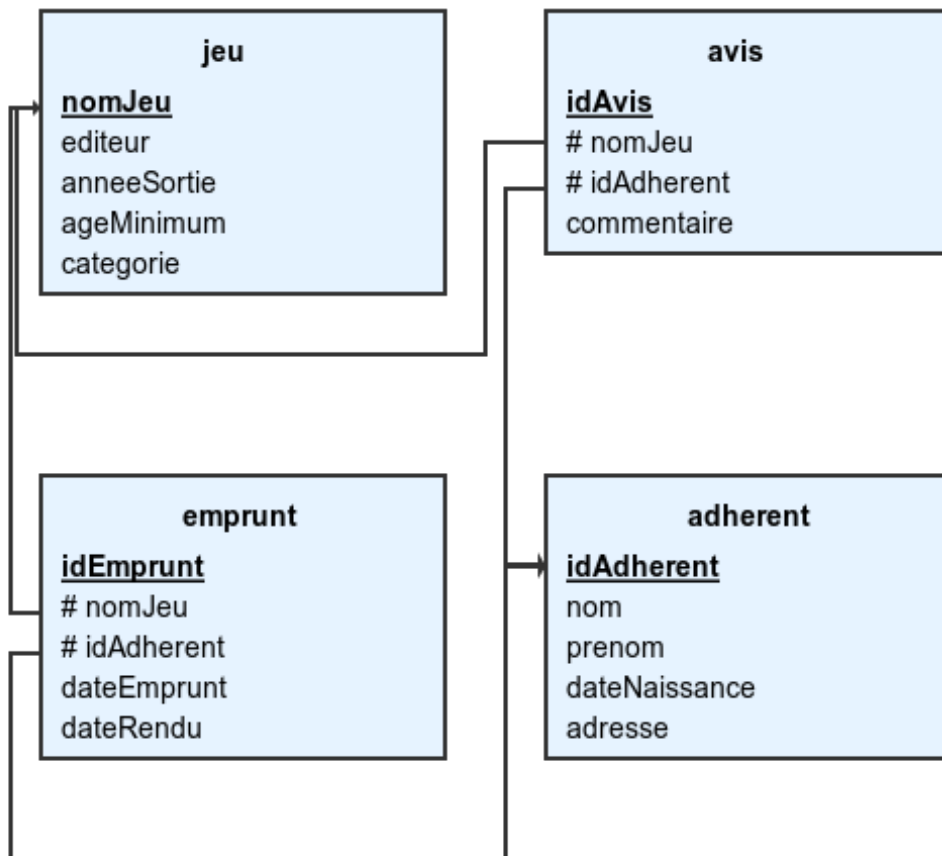


Figure 1. La base de données de la ludothèque

Dans la figure ci-dessus, les clés primaires de chacune des tables sont soulignées et les clés étrangères sont précédées du symbole #.

Dans cet exercice, on pourra utiliser les clauses du langage SQL pour :

- construire des requêtes d'interrogation à l'aide de `SELECT`, `FROM`, `WHERE` (avec les opérateurs logiques `AND`, `OR`) et `JOIN . . . ON` ;
- construire des requêtes d'insertion et de mise à jour à l'aide de `UPDATE`, `INSERT` et `DELETE` ;
- affiner les recherches à l'aide de `DISTINCT` et `ORDER BY` ;
- réaliser des agrégations à l'aide de `COUNT`.

Par exemple, l'instruction SQL :

```
SELECT COUNT(nomJeu) FROM jeu;
```

donne le nombre de jeux présents dans la table `jeu`.

1. Expliquer pourquoi on ne peut pas prendre l'attribut `nom` comme clé primaire pour la relation `adherent`.
2. Décrire ce que donne la requête SQL suivante :

```
SELECT nomJeu, editeur  
FROM jeu  
ORDER BY nomJeu;
```

Lorsque qu'un jeu est emprunté et n'a pas encore été rendu, la valeur de l'attribut `dateRendu` de la table `emprunt` est à `NULL`.

3. Écrire une requête permettant de connaître le nom de tous les jeux qui sont en cours d'emprunt.
4. Écrire une requête SQL pour afficher le nom et le prénom de tous les adhérents qui ont emprunté le jeu "Catan".
5. Claire VOYANT, adhérente de longue date à cette ludothèque, a emprunté le jeu "Catan" et l'a rendu le 3 juin 2025. Lors de l'emprunt, la valeur de `id_emprunt` était 1538.

Écrire une requête SQL qui a permis de mettre à jour la base de données afin qu'elle prenne en compte que ce jeu a été rendu. Toutes les dates de la base de données sont écrites sous le format 'AAAA-MM-JJ'.

6. Écrire une requête SQL qui permet de trouver le nom et la catégorie de tous les jeux de la ludothèque sortis à partir de 2010 et dont l'âge minimum est strictement inférieur à 10 ans.

La ludothèque décide d'organiser des événements. Pour cela, elle ajoute une relation `evenement` à sa base de données. En outre, pour chaque événement, elle souhaite garder en mémoire une trace des adhérents qui y ont participé. À cette fin, elle complète sa base avec une relation `participation`.

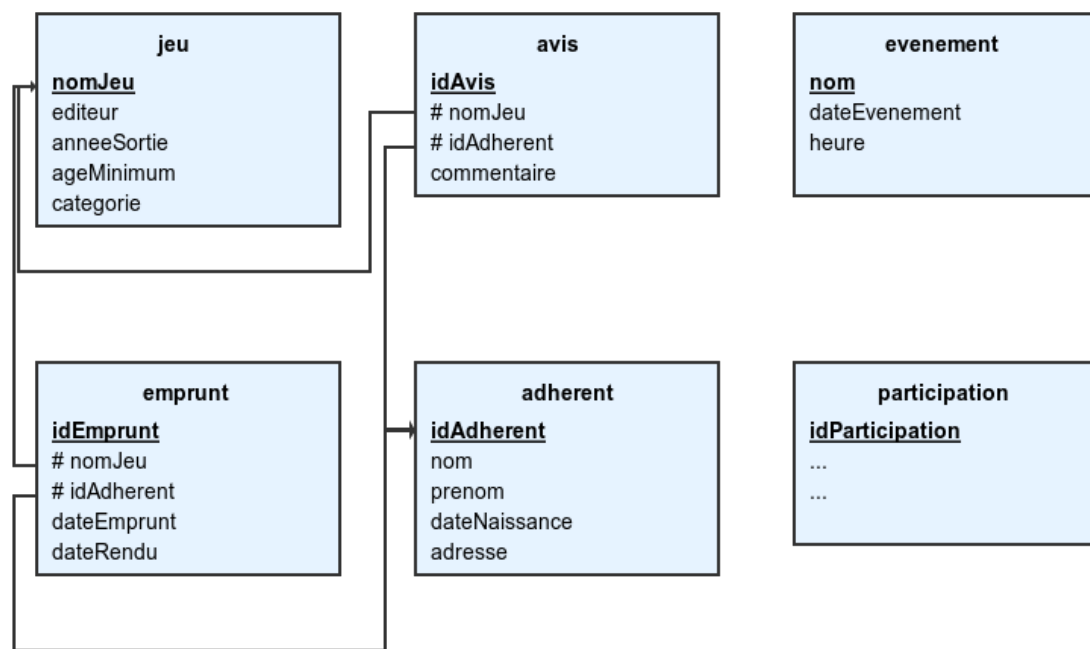


Figure 2. La base de données de la ludothèque actualisée

- Proposer les clés étrangères de la table `participation` en précisant le nom des attributs auxquels elles font référence.

Le programme Python suivant permet de créer la liste de tous les jeux empruntés, sachant que, dans celle-ci, un jeu va apparaître autant de fois qu'il a été emprunté.

```

1 import sqlite3
2
3 # Connexion à la base de données
4 connection = sqlite3.connect("ludothèque.db")
5 curseur = connection.cursor()
6
7 # Exécution de la requête
8 curseur.execute("SELECT nomJeu FROM emprunt")
9
10 # Récupération des résultats
11 jeux = curseur.fetchall()
12
13 liste = []
14 # Création de la liste des jeux empruntés
15 for jeu in jeux:
16     liste.append(jeu[0])
17
18 # Fermeture de la connexion

```

```
19 curseur.close()
20 connection.close()
```

8. Écrire un script Python permettant de créer le dictionnaire `dict_emprunts` qui, à chaque jeu emprunté, associe le nombre de fois où il a été emprunté.

On veut créer un podium des jeux les plus souvent empruntés. Comme il peut y avoir des égalités à la première, deuxième ou troisième place, il peut y avoir plus de trois jeux sélectionnés sur le podium.

Par exemple, si le dictionnaire des emprunts est :

```
1 dict_emprunts = {
2     "Terraforming Mars": 25,
3     "Codenames": 22,
4     "Agricola": 18,
5     "Puerto Rico": 18,
6     "Caylus": 18,
7     "Dominion": 22,
8     "Dixit": 12
9 }
```

il y aura sur le podium les jeux "Agricola", "Puerto Rico" et "Caylus" puis les jeux "Dominion" et "Codenames" et enfin le jeu "Terraforming Mars".

Pour modéliser ce podium en Python, on va utiliser une liste de trois listes.

Pour l'exemple précédent, cette liste sera :

```
[["Agricola", "Puerto Rico", "Caylus"], ["Dominion",
"Codenames"], ["Terraforming Mars"]].
```

9. Proposer un script Python permettant de générer ce podium.

Exercice 3 (8 points)

Cet exercice porte sur la programmation de base en Python, la sécurisation des communications et les réseaux.

Partie A - La méthode du *masque jetable*

Dans cette partie, on s'intéresse à une méthode de chiffrement dite du *masque jetable*. Voici ce que l'on peut lire sur le site Wikipédia :

Le chiffrement par la méthode du masque jetable consiste à combiner le message en clair avec une clé présentant les caractéristiques très particulières suivantes :

- *la clé doit être une suite de caractères au moins aussi longue que le message à chiffrer ;*