

Structure de la Base de Données `movie.db`

Dans cette section, nous allons détailler la structure de la base de données **SQLite** qui constituera la couche fondamentale de notre API cinéma. Cette base s'appuie sur le célèbre dataset **MovieLens**, et regroupe plusieurs entités essentielles du domaine cinématographique : les films, les notes des utilisateurs, les tags et les identifiants externes.

Table `movies`

```
CREATE TABLE movies (  
  movieId INTEGER PRIMARY KEY,  
  title TEXT NOT NULL,  
  genres TEXT  
);
```

Description :

Cette table contient **la liste complète des films**.

- `movieId` : identifiant unique du film. C'est la clé primaire.
- `title` : le titre du film, avec l'année entre parenthèses (ex : *Toy Story (1995)*).
- `genres` : une chaîne de caractères contenant les genres associés au film, séparés par des pipes (`|`).
Exemple : *Adventure/Animation/Children*.

Rôle :

C'est **la table principale**, référencée par les autres tables. Elle permet de connaître les films disponibles et leurs genres associés.

Table `ratings`

```
CREATE TABLE ratings (  
  userId INTEGER,  
  movieId INTEGER,  
  rating REAL CHECK(rating >= 0.5 AND rating <= 5.0),  
  timestamp INTEGER,  
  PRIMARY KEY (userId, movieId),  
  FOREIGN KEY (movieId) REFERENCES movies(movieId)  
);
```

Description :

Cette table contient **les notes attribuées par les utilisateurs** à chaque film.

- `userId` : identifiant de l'utilisateur.
- `movieId` : identifiant du film noté.
- `rating` : la note, comprise entre 0.5 et 5.0 (avec un pas de 0.5).
- `timestamp` : date et heure du rating (sous forme de timestamp UNIX).

Relations :

- Clé primaire composée : (`userId`, `movieId`) — un utilisateur ne peut noter un film qu'une seule fois.
 - `movieId` est une **clé étrangère** pointant vers la table `movies`.
-

Table `tags`

```
CREATE TABLE tags (  
  userId INTEGER,  
  movieId INTEGER,  
  tag TEXT,  
  timestamp INTEGER,  
  PRIMARY KEY (userId, movieId, tag),  
  FOREIGN KEY (movieId) REFERENCES movies(movieId)  
);
```

Description :

Cette table enregistre **les tags** (étiquettes) que les utilisateurs associent à un film.

- `userId` : identifiant de l'utilisateur.
- `movieId` : identifiant du film.
- `tag` : texte libre saisi par l'utilisateur (ex : "hilarious", "thriller", "based on a true story").
- `timestamp` : date et heure d'ajout du tag.

Relations :

- Clé primaire composée : (`userId`, `movieId`, `tag`) — un même utilisateur peut associer plusieurs tags à un film.
 - `movieId` est une **clé étrangère** vers la table `movies`.
-

Table `links`

```
CREATE TABLE links (  
  movieId INTEGER PRIMARY KEY,  
  imdbId TEXT,  
  tmdbId INTEGER,  
  FOREIGN KEY (movieId) REFERENCES movies(movieId)  
);
```

Description :

Cette table contient les **identifiants externes** des films.

- `movieId` : identifiant du film dans MovieLens (clé primaire).
- `imdbId` : identifiant du film dans la base **IMDb**.
- `tmdbId` : identifiant dans **The Movie Database (TMDb)**.

Rôle :

Elle permet de faire **le lien avec d'autres bases de données de films** pour enrichir l'analyse ou appeler des API externes comme IMDb ou TMDb.

Schéma de la Base de Données

- Toutes les tables secondaires (ratings, tags, links) référencent **movieId** dans la table `movies`.
 - Cette structure facilite les jointures pour l'analyse des notes, des tags et l'accès à des ressources externes.
-