
TRAVAUX PRATIQUES n° 17

Bases de données (SQLite)

Dans ce TP, on interroge une base de films avec le module `sqlite3` de Python (aucun logiciel externe). L'objectif est d'écrire des requêtes SQL : projection, sélection, tri, agrégation, puis une jointure.

Placez le fichier `"cinema.sql"` dans le même dossier que votre script Python.

Parcours minimal suggéré : mise en place + requêtes des blocs Projection / Sélection / Tris (questions principales) + agrégations 1 à 6. Le reste et la jointure sont pour aller plus loin.

I. Mise en place

On dispose du code suivant :

```
1 from os import chdir, path, remove
2 import sqlite3
3
4 chdir(r"C:\Users\Prenom\Documents\TP17")
5
6 # Création de la base à partir du script SQL
7 if path.exists("cinema.db"):
8     remove("cinema.db")
9 conn = sqlite3.connect("cinema.db")
10 with open("cinema.sql", encoding="utf8") as f:
11     # SQLite n'accepte pas les ALTER TABLE MySQL (AUTO_INCREMENT, clés étrangères)
12     conn.executescript(f.read().split("ALTER TABLE")[0])
13 conn.commit()
```

Adaptez le chemin dans `chdir` comme au TP 9 (chemin complet du dossier, précédé de `r`).

La table utile pour ce TP s'appelle `film_1_table` (titre, année, durée, catégorie, acteurs, réalisateur, popularité, récompense). La table `film` du dump est la version *normalisée* (exercice Jointure). Pour alléger les requêtes, on crée une vue :

```
1 conn.execute("CREATE VIEW IF NOT EXISTS films AS SELECT * FROM film_1_table")
2 conn.commit()
```

Pour exécuter une requête et afficher le résultat :

```
1 def requete(sql):
2     curseur = conn.execute(sql)
3     for ligne in curseur:
4         print(ligne)
5
6 requete("SELECT titre, annee FROM films LIMIT 5")
```

Le schéma d'une table s'obtient par :

```
PRAGMA table_info(films);
```

Attribut	Rôle
<code>id</code>	clé primaire
<code>titre</code>	titre du film
<code>annee</code>	année de sortie
<code>duree</code>	durée en minutes
<code>categorie</code>	genre
<code>acteur_principal</code>	acteur principal
<code>actrice_principale</code>	actrice principale
<code>realisateur</code>	réalisateur
<code>popularite</code>	note de popularité
<code>recompense</code>	1 si le film a été récompensé, 0 sinon

II. Requêtes sur une seule table

Dans les requêtes, on interroge la vue `films`.

1) Projection et limitation

Écrire une requête permettant d'afficher

- 1) tous les films (tous les attributs)
- 2) les attributs `titre`, `annee`, `categorie` et `popularite` seulement
- 3) les 20 premiers films (`LIMIT 20`)
- 4) les films 21 à 50 (`LIMIT 30 OFFSET 20`)
- 5) les acteurs et actrices principaux, colonnes renommées « Acteurs » et « Actrices » (`AS`)

2) Sélection (WHERE)

Écrire une requête permettant d'afficher

- 1) les films récompensés (`recompense = 1`)
- 2) les films sortis après 1980
- 3) les films sortis avant 1960
- 4) les films d'une durée comprise entre 100 et 120 minutes
- 5) les films de durée < 60 ou ≥ 120
- 6) les films de « Steven Spielberg »
- 7) les films dont l'acteur principal est « Gérard Depardieu »
- 8) les films qui ne sont ni des films de guerre, ni des films d'action

3) Tris (ORDER BY)

Écrire une requête permettant d'afficher

- 1) tous les films, du plus ancien au plus récent
- 2) tous les films, du plus récent au plus ancien
- 3) tous les films triés par catégorie
- 4) tous les films, du plus populaire au moins populaire
- 5) tous les films triés par popularité décroissante, puis récompense décroissante, puis année croissante

4) Valeurs distinctes (DISTINCT)

Écrire une requête permettant d'afficher

- 1) toutes les catégories, une seule fois (DISTINCT)
- 2) tous les réalisateurs, une seule fois

III. Agrégations

On utilise **COUNT**, **MIN**, **MAX**, **SUM**, **AVG**, éventuellement avec **GROUP BY**.

- 1) Combien y a-t-il de films en tout ?
- 2) Combien y a-t-il de films d'action ?
- 3) Combien y a-t-il de films de « Alfred Hitchcock » ?
- 4) Combien y a-t-il de films de « Alfred Hitchcock » dans la catégorie « Drame » ?
- 5) Combien d'acteurs, d'actrices et de réalisateurs distincts (une seule requête, trois colonnes) ?
- 6) Quels sont les films les moins populaires (en deux requêtes) ?
- 7) Quels sont les films les plus populaires (en deux requêtes) ? *(pour aller plus loin)*
- 8) Quel est le film le plus court ? le plus long ? *(pour aller plus loin)*
- 9) Combien de films ont été récompensés ? *(pour aller plus loin)*
- 10) Quelle est la durée totale des films ? la durée moyenne ? la popularité moyenne ? *(pour aller plus loin)*
- 11) Quelle catégorie a la moyenne de popularité la plus grande ? *(pour aller plus loin)*
- 12) Quels réalisateurs ont la moyenne de popularité la plus grande ? *(pour aller plus loin)*

Exercice 1 (*Jointure (pour aller plus loin)*) : La table `film` (normalisée, **distincte** de la vue `films`) relie un film à `categorie_id`. La table `categorie` contient `id` et le nom du genre.

- 1) Essayer :

```
SELECT film.titre , categorie.nom
FROM film JOIN categorie ON film.categorie_id = categorie.id
LIMIT 10;
```

Ici `film` désigne la table normalisée (avec `categorie_id`), distincte de la vue `films`.

- 2) Écrire une requête donnant, pour chaque catégorie, le nombre de films.