TP Info 7 | Bases de données

Le but de ce TP est d'interroger en SQL une base de données préconstruite sur le thème du cinéma.

1. Ce qu'il faut savoir

Projection

▷ C'est la commande **SELECT** qui permet d'effectuer une projection (et non une sélection...) :

SELECT col1,col2,... FROM table

▷ On utilise **SELECT *** pour projeter sur toutes les colonnes.

▷ Attention, la commande **SELECT** n'élimine pas les doublons. On obtiendra une sélection au sens de l'algèbre relationnelle en précisant **SELECT DISTINCT** :

SELECT DISTINCT col1,col2,... FROM table

La sélection

▷ On peut effectuer une sélection en ajoutant une clause **WHERE** à **SELECT** :

SELECT col1,... FROM table WHERE condition

▷ On peut construire des conditions en utilisant les opérateurs logiques **AND**, **OR** et **NOT**.

Renommage

▷ C'est le mot-clé **AS** qui permet le renommage :

SELECT * AS tablebis FROM table

Produit cartésien, jointure et division cartésienne

▷ On décrit un produit cartésien de tables en énumérant les tables après **FROM**.

▷ **SELECT ... FROM table1,table2**

▷ Il n'existe aucune commande spécifique à la division cartésienne dans le langage SQL. On peut cependant la simuler au moyen d'autres commandes.

▷ Une jointure s'effectue au moyen de **JOIN ... ON**.

▷ **SELECT col1,col2,... FROM (table1 JOIN table2 ON condition)**

▷ On peut aussi créer une jointure en parcourant un produit cartésien avec une clause **WHERE**. Cette méthode est plus lisible en cas de jointures multiples.

▷ **SELECT col1,col2,... FROM table1,table2 WHERE condition**

Opérations ensemblistes

▷ L'opérateur **UNION** réalise la réunion de deux tables.

▷ **SELECT * FROM table1 UNION SELECT * FROM table2**

▷ L'intersection n'existe pas sous SQL mais on peut la simuler à l'aide de jointures ou de sélections.

▷ Idem pour l'opération de différence.

Groupement

▷ Le comptage des enregistrements d'une table s'effectue au moyen de la fonction **COUNT**.

▷ **SELECT COUNT(*) AS total FROM tab**

▷ **MAX**, **AVG**, **MIN** et **SUM** permettent de déterminer le maximum, la moyenne, le minimum et la somme d'une colonne.

▷ **SELECT MAX(col1) AS maximum FROM table**

▷ On peut appliquer une fonction à un groupe de lignes en utilisant la commande **GROUP BY**.

▷ **SELECT col2,SUM(col1) as total FROM table GROUP BY col2**

▷ La clause **HAVING** remplace **WHERE** lorsque les colonnes intervenant dans la condition proviennent d'une fonction.

▷ **SELECT col2,SUM(col1) as total FROM table GROUP BY col2 HAVING condition sur total**

La bibliothèque sqlite3

a) Il est possible de manipuler une base de données directement depuis un interpréteur Python.

b) On utilise pour cela la bibliothèque sqlite3.

c) On commence par créer *une connexion* à la base de données au moyen de la méthode `connect` :

conn=sqlite3.connect('C ://nom.db')

d) On crée alors *un curseur* au moyen de la méthode `cursor` :

c=conn.cursor()

e) On peut alors utiliser du code SQL encapsulé au moyen de `execute` :

c.execute('SELECT...')

f) On récupère le résultat au moyen de la méthode `fetchall` :

print(c.fetchall())

Exemples

```
▷ >>> import sqlite3
▷ >>> conn=sqlite3.connect('E://films.db')
▷ >>> c=conn.cursor()
▷ >>> c.execute('SELECT prenom FROM acteur')
▷ <sqlite3.Cursor object at 0xb582e760>
▷ >>> print(c.fetchall())
▷ [(u'Tim',), (u'Morgan',), (u'Bob',),...
```

2. Exercices

2.1. Avant de commencer la séance

Nous allons illustrer le cours par une base de données sur le cinéma : films.db. Elle est constituée de six relations. Voici son schéma relationnel :

- ▷ acteur[id,prenom,nom]
- ▷ entree[id_utilisateur,id_film,note]
- ▷ film[id,titre,annee,id_realisateur]
- ▷ jeu[id_film,id_acteur]
- ▷ realisateur[id,prenom,nom]
- ▷ utilisateur[id,prenom,nom]

Ouvrir pyzo, importer la bibliothèque `sqlite3`, créer une connexion `conn` et un curseur `c`.

Effectuer quelques requêtes afin de comprendre la structure et le type d'informations contenues dans chaque relation.

2.2. Requêtes

a) Afficher les films sortis l'année de votre naissance.

b) Afficher les films de **Akira Kurosawa**.

c) Afficher les films de **Jim Carrey**.

d) Afficher les acteurs du film **Inception**.

e) Afficher les acteurs ayant été dirigés par **Francis Ford Coppola**.

f) Afficher les films d'**Alfred Hitchcock** avec **Cary Grant**.

g) Afficher la liste des réalisateurs accompagnés du nombre de leurs films dans la base de données.

h) Afficher les acteurs ayant joué dans au moins trois films de cette base de données.

i) Afficher le total des films par année par valeurs décroissantes.

j) Afficher le total de films par réalisateur par valeurs décroissantes.

k) Afficher chaque utilisateur avec la moyenne de ses notes.

l) Afficher les moyennes des notes des films par ordre décroissant.

m) Proposer à votre voisin une requête futile de votre choix.