



Les bases de données



Résumé¹

Une base de données (en anglais database), permet de stocker et de retrouver des données structurées, semi-structurées ou des données brutes ou de l'information, souvent en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles.

Leurs données peuvent être stockées sous une forme très structurées (base de données relationnelles par exemple), ou bien sous la forme de données brutes peu structurées voire déstructurées (avec les bases de données NoSQL par exemple).

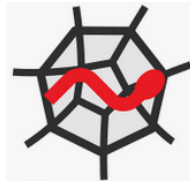
Une base de données peut être localisée dans un même lieu et sur un même support informatisé, ou réparties sur plusieurs machines à plusieurs endroits.

Sommaire

1	Présentation des Bases de Données	2
1.1	<i>Données et bases de données</i>	2
1.2	<i>Quel support pour nos tables ?</i>	3
1.3	<i>Le SGBD</i>	4
2	Une base de données	5
2.1	<i>Schéma relationnel, vocabulaire</i>	5
2.2	<i>Un bon schéma relationnel.....</i>	5
2.3	<i>Clés primaires, clés étrangères.....</i>	5
	Clés primaires.....	6
	Clés étrangères	6
2.4	<i>Conception des tables.....</i>	6
	Analyse de la table Film.....	6
	Schémas des tables de la base	7
2.5	<i>Résumé</i>	8
3	Interrogation de la base.....	9
3.1	<i>Le langage SQL pour l'interrogation par l'exemple</i>	9
	Essais sur la base Film avec SQLiteDatabaseBrowser.	9
	Requêtes sur une seule table	9
	Requêtes sur deux tables	11
	Requête sur trois tables en même temps.....	11
	Quelques ressources.....	11
3.2	<i>Exercices 'papier' avec la base bibliothèque</i>	12
	Le schéma utilisé	12
	Écriture de requêtes SQL	12



¹ Source Wikipédia



1 Présentation des Bases de Données²

1.1 Données et bases de données

Les données sont multiples, une fois numérisées elles sont disponibles dans nos ordinateurs sous la forme de chaînes de caractères, d'entiers, de dates etc. Pour pouvoir être exploitées elles ne sont pas stockées 'en vrac'. Une certaine organisation est nécessaire. Bien plus les données peuvent avoir des relations entre elles, relations qui permettent à l'aide du traitement adéquat d'obtenir des informations pertinentes.

Retrouvons les tables vues en 1^{ère} NSI³

La table film :

Titre	Directeur	Acteur
Le crabe tambour	Schoendoerffer	Rochefort
Le crabe tambour	Schoendoerffer	Perrin
...		
Vingt Mille Lieues sous les mers	Fleischer	Douglas
Vingt Mille Lieues sous les mers	Fleischer	Mason
Vingt Mille Lieues sous les mers	Fleischer	Lukas

Cette table relie les informations suivantes : des titres de films, des directeurs (metteurs en scène) et des acteurs. Le fait que trois de ces informations se retrouvent sur une même ligne crée une information supplémentaire : l'acteur *Rochefort* a tourné dans le film *Le crabe tambour* sous la direction de *Schoendoerffer*. Nous voyons ici que l'organisation de l'information permet d'obtenir des informations supplémentaires : appelée dans le langage des bases de données une relation. Donc ici une relation entre Titre, Directeur, Acteur.

Nous pouvons compléter notre table :

La table coordonnées

Salle	Adresse	Téléphone
Pathé	Grenoble 21 Boulevard Maréchal Lyautey	08 92 69 66 96
Le Mèlies	Grenoble 28 Allée Henri Frenay	04 76 47 99 31
...		
Les 6 Rex	Grenoble 13 Rue Saint-Jacques	04 76 44 06 82
Ciné Club de Grenoble	Grenoble 4 Rue Hector Berlioz	04 76 44 70 38

La table séance

Salle	Titre	Horaire
Pathé	Mais qui a tué Harry ?	18H30
Pathé	Vingt Mille Lieues sous les mers	19H00
...		
Les 6 Rex	Mais qui a tué Harry ?	18H00
Les 6 Rex	Le souper	22H00

L'analyse effectuée sur la table film peut se faire sur ces deux nouvelles tables. Mais bien plus des colonnes contenant des données identiques vont nous permettre d'étendre le mécanisme de relation à un ensemble de tables.

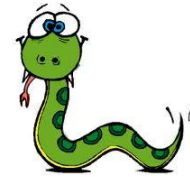
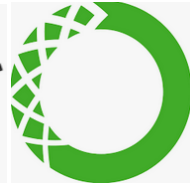
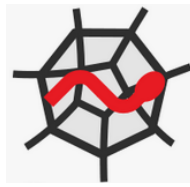
Nous allons pouvoir interroger non pas une, mais plusieurs tables pour obtenir des réponses à des questions plus complexes.

Quels sont les adresses des salles qui projettent des films entre 18H30 et 19H15 ?



² Rédigé avec le support du cours <http://sql.bdpedia.fr/intro.html> Philippe Rigaux, Cnam

³ http://sti2dvox.patque.com/NSI_1ERE/Fichiers_2021/007_DONNEES_EN_TABLES/NSI_DONNEES_EN_TABLE.pdf



Les tables vues sur la page précédente constituent une base de données :

Une base de données : c'est une collection de données structurées qui sont stockées sur un support persistant.

1.2 Quel support pour nos tables ?

Comme nous l'avons vu l'année dernière un support possible pour stocker nos tables est l'utilisation de fichier CSV. Voilà un extrait de la table FILM.csv :

```

FILM.csv
1 Titre;Directeur;Acteur
2 Mais qui a tué Harry ?;Hitchcock;Gwenn
3 Mais qui a tué Harry ?;Hitchcock;Forsythe
4 Mais qui a tué Harry ?;Hitchcock;MacLaine
5 Mais qui a tué Harry ?;Hitchcock;Hitchcock
6 Cris et chuchotements;Bergman;Andersson
7 Cris et chuchotements;Bergman;Sylwan
    
```

Notre base de données serait alors structurée de la manière suivante⁴, un ensemble de fichiers csv.

Cette manière de faire oblige chacun des utilisateurs à concevoir son programme de gestion des données pour pouvoir les exploiter, en ajouter, en supprimer.

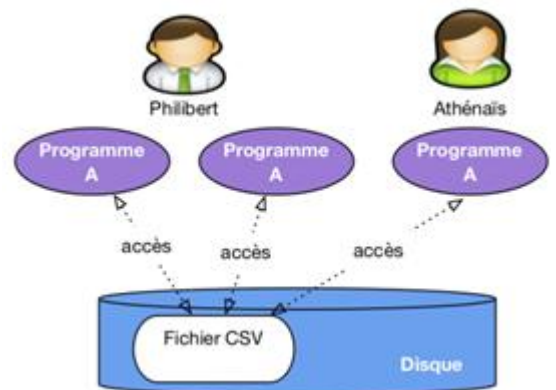
Cela ne va pas sans poser les problèmes suivants⁵ :

Lourdeur d'accès aux données. En pratique, pour chaque accès, même le plus simple, il faudrait écrire un programme adapté à la structure du fichier. La production et la maintenance de tels programmes seraient extrêmement coûteuses.

Risques élevés pour l'intégrité et la sécurité. Si tout programmeur peut accéder directement aux fichiers, il est impossible de garantir la sécurité et l'intégrité des données. Quelqu'un peut très bien par exemple, en toute bonne foi, faire une fausse manœuvre qui rend le fichier illisible.

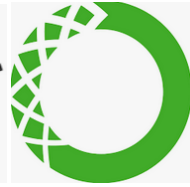
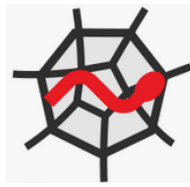
Pas de contrôle de concurrence. Dans un environnement où plusieurs utilisateurs accèdent aux mêmes fichiers, comme illustré sur la figure ci-dessus des problèmes de concurrence d'accès se posent, notamment pour les mises à jour. Comment gérer par exemple la situation où deux utilisateurs souhaitent en même temps ajouter une ligne au fichier ?

Performances. Tant qu'un fichier ne contient que quelques centaines de lignes, on peut supposer que les performances ne posent pas de problème, mais que faire quand on atteint les Gigaoctet (1,000 Mégaoctets), ou même le Téraoctet (1,000 Gigaoctet) ? Maintenir des performances acceptables suppose la mise en œuvre d'algorithmes ou de structures de données demandant des compétences très avancées, probablement hors de portée du développeur d'application qui a, de toute façon, mieux à faire.



⁴ Illustrations issues de supra Philippe Rigaux, Cnam

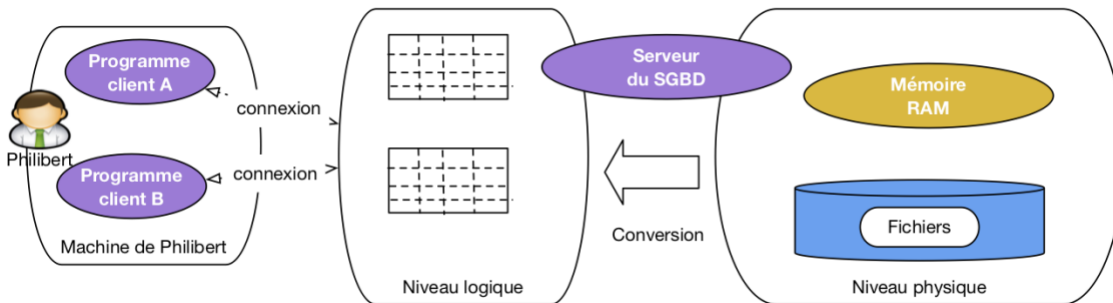
⁵ <http://sql.bdpedia.fr/files/cbd-sql.pdf> page 14 et 15.



Chacun de ces problèmes soulève de redoutables difficultés techniques. Leur combinaison nécessite la mise en place de systèmes d'une très grande complexité, capable d'offrir à la fois un accès simple, sécurisé, performant au contenu d'une base, et d'accomplir le tour de force de satisfaire de tels accès pour des dizaines, centaines ou même milliers d'utilisateurs simultanés, le tout en garantissant l'intégrité de la base même en cas de panne. De tels systèmes sont appelés Systèmes de Gestion de Bases de Données, ou SGBD.

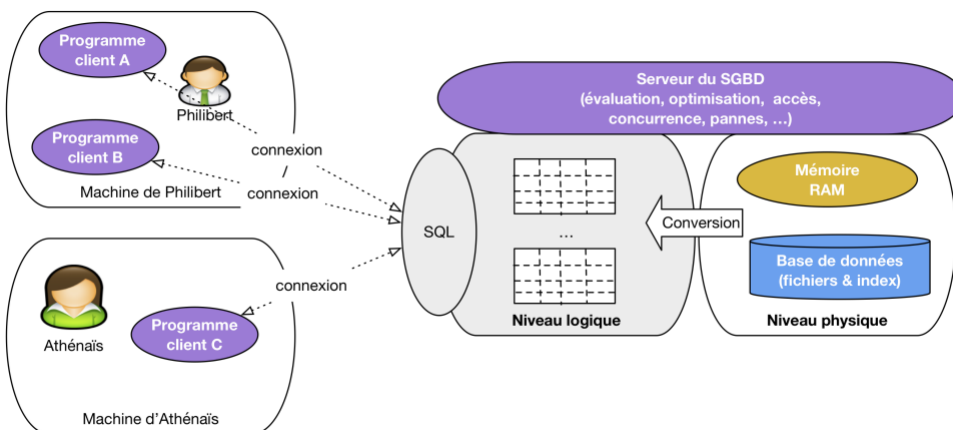
1.3 Le SGBD

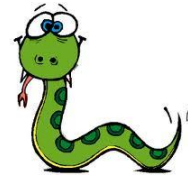
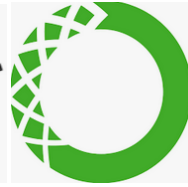
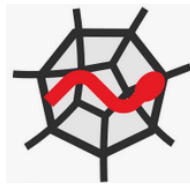
Avec un SGBD l'utilisateur n'a plus accès aux bases que par l'intermédiaire d'une interface logique. C'est par le biais de cette interface que l'utilisateur envoie des requêtes. Le système de gestion analysera les requêtes et produira les réponses à partir des fichiers enregistrés sur disques. L'utilisateur ne sera jamais directement relié aux fichiers physiques constituant la base.



Il utilisera pour l'interrogation des bases un ou des langages mettant en œuvre des concepts fondés sur l'algèbre relationnel. Ces langages permettent de traduire les requêtes en actions à réaliser sur les différentes tables.

Le plus connu est le langage SQL : Structured Query Language





2 Une base de données

2.1 Schéma relationnel, vocabulaire

La relation décrite par la table film relie les colonnes appelées **les attributs** donc ici : Titre, Directeur, Acteur.

Chaque attribut prend ses valeurs dans **un domaine** donc par exemple un entier, une chaîne de caractères, un flottant, une date ... Chaque attribut ne contient qu'une seule valeur.

Le schéma d'une relation précise la liste ordonnée des attributs et de leurs domaines, donc pour la table film :

Schéma : ((Titre, chaîne), (Directeur, chaîne), (Acteur, chaîne))

Une table correspondant à une relation est donc un ensemble fini d'éléments représenté par des n-uplets dont toutes les valeurs respectent le schéma.

Un élément d'une table s'appelle **un enregistrement**.

Titre	Directeur	Acteur
Le crabe tambour	Schoendoerffer	Rochefort
Le crabe tambour	Schoendoerffer	Perrin
...		
Vingt Mille Lieues sous les mers	Fleischer	Douglas
Vingt Mille Lieues sous les mers	Fleischer	Mason
Vingt Mille Lieues sous les mers	Fleischer	Lukas

Un

2.2 Un bon schéma relationnel

Nous pourrions penser qu'il suffit de créer une seule table et de mettre toutes les données dedans pour répondre à toutes les interrogations sur cette table. **En fait cette façon de procéder est inappropriée** car il est alors possible de représenter la même information plusieurs fois dans ce cas on dit qu'il y a redondance de l'information.

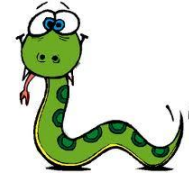
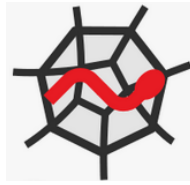
Cette redondance pose problème dans le cas de modifications par exemple : est-on certain d'avoir modifié tous les enregistrements concernés ? De plus cette redondance conduit à une occupation de place pour le stockage préjudiciable aux bonnes performances de la base.

Une base de données bien décrite est un schéma relationnel mettant en œuvre plusieurs tables. Comme dans notre exemple d'introduction.

2.3 Clés primaires, clés étrangères

Il faut garder à l'esprit que les bases de données peuvent avoir des tailles très importantes, de l'ordre du petaoctet 2^{50} octets ! Pour pouvoir accéder rapidement aux données et assurer une bonne intégrité d'une base de données nous allons utiliser des attributs particuliers appelés des clés : clés primaires et clés étrangères.





Clés primaires

Pour permettre la rapidité des recherches un attribut particulier jouera le rôle de l'indexage. La colonne correspondante est définie comme une clé primaire PRIMARY KEY. Cela impose que toutes les valeurs de cette colonne soient différentes.

Choisir soi-même les valeurs des clés n'est pas très facile pour les bases importantes. Il est possible de rendre ce processus automatique. C'est le gestionnaire de la base qui s'occupera dans ce cas de choisir la bonne valeur de la clé.

Attention : dans ce cas c'est à toi de bien choisir les valeurs des clés.



Clés étrangères

Nous avons besoin d'un mécanisme permettant d'assurer une bonne cohérence de nos bases de données. En effet dans les deux tables ci-dessous qui peut nous garantir que les films dans la table séance sont bien présents dans la table film ?

Pour assurer cette cohérence il faut utiliser des clés spécifiques appelées clés étrangères. Elles ne remplacent pas les clés primaires mais elles permettent d'assurer automatiquement des vérifications entre les attributs de tables différentes. Il sera de cette manière impossible de rentrer un enregistrement dans la table séance avec un Titre de film non défini dans la table film.

Une clé étrangère pourra être définie sur plusieurs colonnes si la situation l'exige.

La table film :

Titre	Directeur	Acteur
Le crabe tambour	Schoendoerffer	Rochefort
Le crabe tambour	Schoendoerffer	Perrin
...		
Vingt Mille Lieues sous les mers	Fleischer	Douglas
Vingt Mille Lieues sous les mers	Fleischer	Mason
Vingt Mille Lieues sous les mers	Fleischer	Lukas

La table séance

Salle	Titre	Horaire
Pathé	Mais qui a tué Harry ?	18H30
Pathé	Vingt Mille Lieues sous les mers	19H00
	...	
Les 6 Rex	Mais qui a tué Harry ?	18H00
Les 6 Rex	Le souper	22H00

Dans le schéma ci-dessus la colonne Titre de la table Séance est reliée à la colonne Titre de la table Film par une clé étrangère. Cela permet d'assurer une bonne cohérence de la base.

2.4 Conception des tables

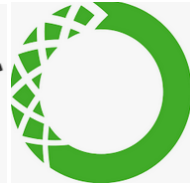
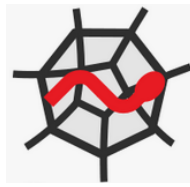
Nous voyons avec tous les éléments décrits ci-dessus que la bonne conception des bases ne peut pas se faire sans arrières pensées sur l'utilisation de celles-ci. La structuration retenue devra permettre de bonnes performances tout en assurant une bonne cohérence des données contenues.

Analyse de la table Film

Q1. En examinant le schéma actuel de la table film donnez votre avis sur la pertinence de la conception.

Q2. Comment résoudre le problème posé ?





Q3. Quelle colonne prendre comme clé primaire dans la table Film ?

Q4. Comment résoudre le problème posé à la question précédente ?

Schémas des tables de la base

Un schéma est une représentation synthétique des tables d'une base de données relationnelle. Pour ces schémas les clés primaires seront indiquées en caractères soulignés et les clés étrangères en italique. Les domaines de valeur des différentes colonnes peuvent également être précisés.

Voilà la base Cinéma représentée de cette manière :

Acteur

<u>idActeur</u>	nomActeur	anneeNaissance	<i>#idFilm</i>
integer	varchar2(40)	integer	integer

Coordonnees

<u>idCoord</u>	nomSalle	Adresse	Telephone
integer	varchar2(40)	varchar2(40)	varchar2(15)

Film

<u>idFilm</u>	Titre	nomDirecteur	anneeNaissance
integer	varchar2(40)	varchar2(40)	integer

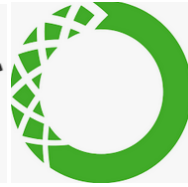
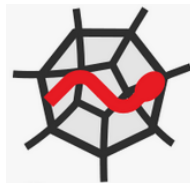
Seance

<u>idSeance</u>	<i>#idSalle (idCoord)</i>	<i>#idFilm</i>	Horaire
integer	integer	integer	time

Q5. Listez les différences avec les trois tables présentées en introduction.

Q6. Commentez l'intégrité obtenue avec les clés étrangères présentes dans la description.





Q7. Proposez un schéma pour une base de données gérant la bibliothèque municipale⁶ de votre quartier et présentant les différents ouvrages disponibles dans le fonds de cette bibliothèque.

Q8. Proposez une extension au schéma de votre base pour gérer les emprunteurs des ouvrages.

2.5 Résumé⁷

SGBD

Un Système de Gestion de Bases de Données (SGBD) est un système qui fournit un accès efficace, pratique et multi-utilisateur à des ensembles de données (nombreuses) qu'il gère et stocke de manière persistante. 3 fonctions importantes et liées entre elles :

- garantir un état cohérent
- contrôle de la concurrence
- reprise après panne (récupération)

Le SGBD doit garantir des exécutions qui vérifient les propriétés suivantes :

- Atomicité
- Cohérence
- Isolation
- Durabilité

Atomicité

Une transaction doit être exécutée de manière atomique (tout ou rien). Elle doit être exécutée dans sa totalité ou annulée en totalité. Elle doit être considérée comme une unique action non divisible.

Cohérence

La transaction transforme la base d'un état cohérent en un autre état cohérent. Cette responsabilité peut être partagée entre le SGBD et les développeurs d'applications.

Isolation

Des transactions doivent pouvoir s'exécuter en parallèle de manière indépendante les unes des autres. ie: les effets partiels des transactions incomplètes ne doivent pas être visibles par les autres transactions.

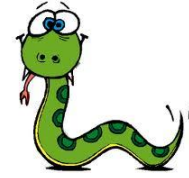
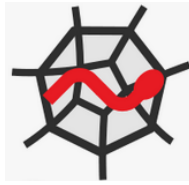
Durabilité

Les effets d'une transaction validée sont inscrits de manière durable et ne doivent pas subir de perte suite à une défaillance.



⁶ D'après un exercice proposé dans NSI Terminale, S. Bays, p. 190

⁷ Source DIUEIL Grenoble 2020 C. Labbé




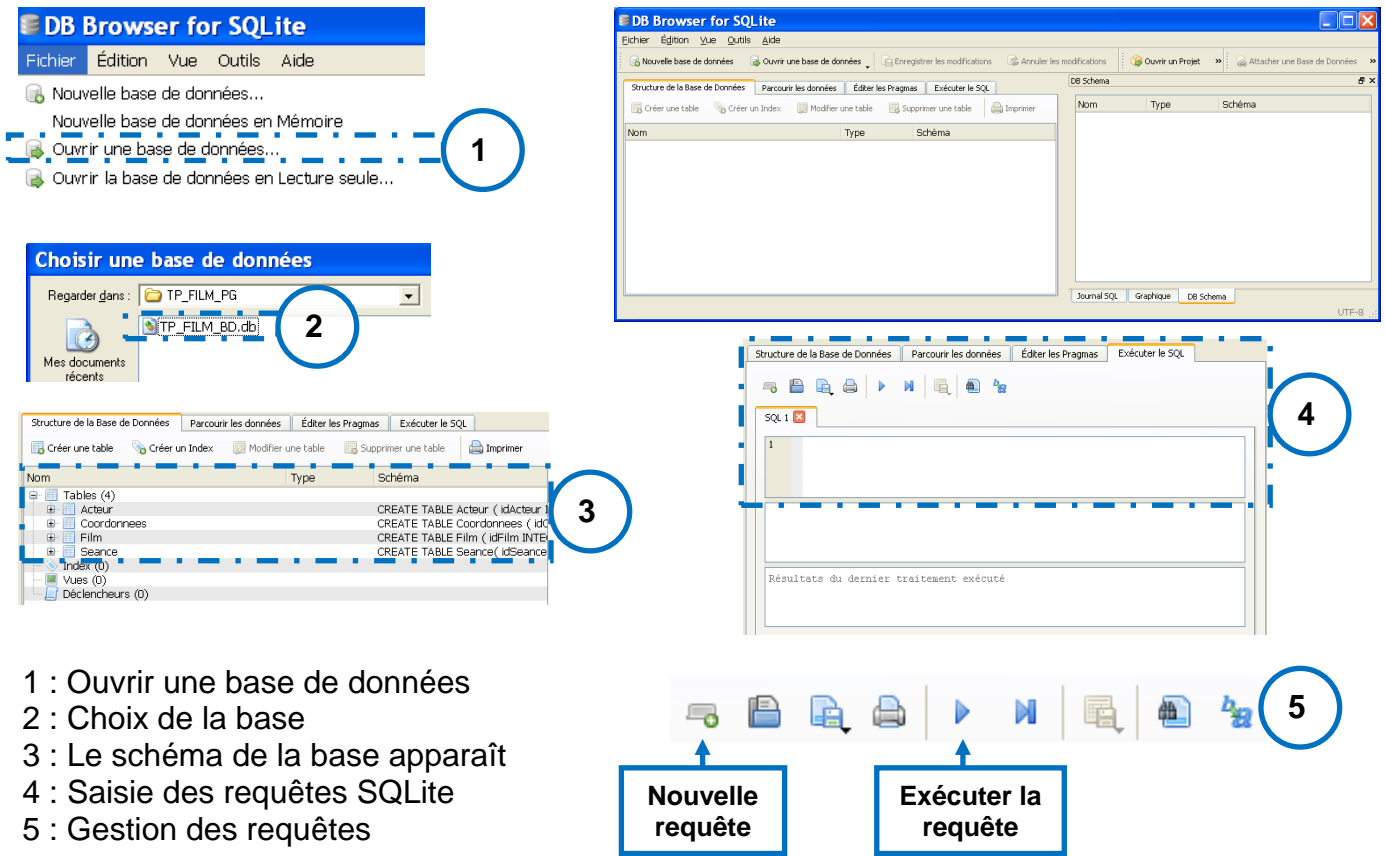
3 Interrogation de la base TP_FILM_BD.db

3.1 Le langage SQL pour l'interrogation par l'exemple

Essais sur la base Film avec SQLiteDatabaseBrowser.

Ce logiciel gratuit sans installation et simple d'emploi permet de visualiser les bases SQLite et permet aussi de les interroger avec des requêtes SQL. Nous allons l'utiliser pour découvrir les possibilités.

 SQLiteDatabaseBrowserPortable.exe



The screenshot shows the DB Browser for SQLite interface with five numbered steps:

- 1: Clicking on 'Ouvrir une base de données...' in the menu.
- 2: Selecting the 'TP_FILM_BD.db' file in the file browser.
- 3: The database schema is displayed in the 'DB Schema' pane, showing tables like Acteur, Coordonnees, Film, and Seance.
- 4: The SQL editor pane is active, ready for input.
- 5: The toolbar at the bottom, with 'Nouvelle requête' and 'Exécuter la requête' buttons highlighted.

- 1 : Ouvrir une base de données
- 2 : Choix de la base
- 3 : Le schéma de la base apparaît
- 4 : Saisie des requêtes SQLite
- 5 : Gestion des requêtes

Il est vivement conseiller de tester en direct les exemples proposés ci-dessous.

Requêtes sur une seule table

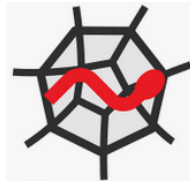
Requête_1. Lister tous les enregistrements d'une table.

```
SELECT * FROM film
```

	idFilm	Titre	nomDirecteur	anneeNaissance
1	1	Mais qui a tué Harry ?	Hitchcock	1899
2	2	Cris et chuchotements	Bergman	1918
3	3	Vingt Mille Lieues sous les mers	Fleischer	1916

Result: 14 enregistrements ramenés en 20ms
At line 1:
SELECT * FROM Film





Requête_2. Donnez la liste de tous les directeurs (metteur en scène) des films

```
SELECT nomDirecteur FROM Film
```

	nomDirecteur
1	Hitchcock
2	Bergman
3	Fleischer

Result: 14 enregistrements ramenés en 9ms
At line 1:
SELECT nomDirecteur FROM Film

	nomDirecteur
1	Hitchcock
2	Bergman
3	Fleischer

Result: 12 enregistrements ramenés en 9ms
At line 1:
SELECT DISTINCT nomDirecteur FROM Film

Requête_3. Donnez tous les directeurs mais éviter les doublons

```
SELECT DISTINCT nomDirecteur FROM Film
```

Requête_4. Combien de fois le metteur en scène Truffaut apparaît dans la base ?

```
SELECT COUNT(nomDirecteur) FROM Film WHERE nomDirecteur = "Truffaut"
```

	COUNT(nomDirecteur)
1	2

Requête_5. Idem 4 mais mettre le nom du metteur en scène

```
SELECT nomDirecteur, COUNT(nomDirecteur) FROM Film WHERE nomDirecteur = "Truffaut"
```

	nomDirecteur	COUNT(nomDirecteur)
1	Truffaut	2

Requête_6. Lister tous les metteurs en scène avec leur nombre de films dans la base

```
SELECT nomDirecteur, COUNT(*) FROM Film GROUP BY nomDirecteur
```

	nomDirecteur	COUNT(*)
1	Autant-Lara	1
2	Bergman	1
3	Dhéry	1
4	Fleischer	1
5	Girault	1
6	Hitchcock	1
7	Lucas	1
8	Molinaro	1
9	Oury	1
10	Schoendoerffer	1
11	Spielberg	2
12	Truffaut	2

Requête_7. Comptez le nombre de séances qui sont comprises entre 19H et 21H30

```
SELECT count(*) FROM Seance WHERE Horaire >= "19:00" and Horaire <="21:30"
```

	count(*)
1	8

Requête_8. Donner tous les acteurs dont le nom commence par la lettre 'D'

```
SELECT DISTINCT nomActeur FROM Acteur WHERE nomActeur LIKE "D%"
```

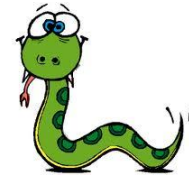
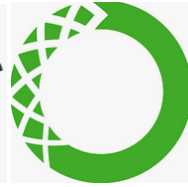
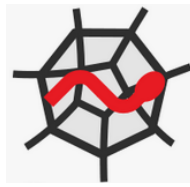
	nomActeur
1	Douglas
2	De Funès
3	Dhéry
4	Denner
5	Duflilio
6	Dreyfuss
7	Dillon
8	Decomble

Requête_9. Donner la liste des noms d'acteurs par ordre alphabétique.

```
SELECT DISTINCT nomActeur FROM Acteur ORDER BY nomActeur
```

	nomActeur
1	Andersson
2	Balaban
3	Batti
4	Bouquet
5	Bourvil
6	Brasseur
7	Brialy





Requête_10. Donner la liste des acteurs nés pendant la seconde guerre mondiale et leurs années de naissance ordonnée suivant les dates décroissantes.

```
SELECT DISTINCT nomActeur,anneeNaissance FROM Acteur
WHERE anneeNaissance >= 1939 AND anneeNaissance <= 1945
ORDER BY anneeNaissance DESC
```



	nomActeur	anneeNaissance
1	Balaban	1945
2	Holgado	1944
3	Garr	1944
4	Léaud	1944
5	Ford	1942
6	Perrin	1941
7	Sylwan	1940
8	Dillon	1939

- ORDER BY DESC
- ORDER BY ASC

Décroissant DESC
Croissant ASC

Requêtes sur deux tables

Requête_11. Combien y a-t-il d'acteurs qui ont été directeur de films ?

```
SELECT DISTINCT nomActeur FROM Acteur A, Film F WHERE A.nomActeur = F.nomDirecteur
```

	nomActeur
1	Hitchcock
2	Dhéry
3	Truffaut

Requête_12. Donnez le titre et les horaires des films qui sont compris entre 19H et 21H30

```
SELECT F.Titre, S.Horaire FROM Film F, Seance S
WHERE
  Horaire >= "19:00" AND Horaire <="21:30"
  AND S.idFilm = F.idFilm
```



	Titre	Horaire
1	Vingt Mille Lieues sous les mers	19:00:00
2	Vingt Mille Lieues sous les mers	20:00:00
3	Rencontres du troisième type	19:00:00
4	Le crabe tambour	20:00:00
5	Les quatre cents coups	19:00:00
6	La mariée était en noir	20:00:00
7	Cris et chuchotements	21:00:00
8	Rencontres du troisième type	21:00:00

Requête sur trois tables en même temps

Requête_13. Donnez les titres et horaires des films dans lequel joue De Funès et qui passent entre 18H et 23H30.

```
SELECT F.Titre, S.Horaire FROM Film F, Seance S, Acteur A
WHERE (Horaire >= "18:00" and Horaire <="23:30")
AND S.idFilm = F.idFilm
AND F.idFilm = A.idFilm
AND A.nomActeur="De Funès"
```

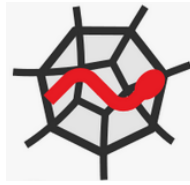
	Titre	Horaire
1	La grande vadrouille	18:00:00

Quelques ressources

<https://www.w3schools.com/sql/>

<https://sql.sh/cours/>





3.2 Exercices 'papier' avec la base bibliothèque

Le schéma utilisé

Livres

<u>idTitre</u>	Titre	annee	#idGenre	#idEditeur
integer	varchar2(40)	integer	integer	integer

Auteurs

<u>idAuteur</u>	nom	prenom
integer	varchar2(40)	varchar2(40)

Editeurs

<u>idEditeur</u>	nom
integer	varchar2(40)

Ecrits

<u>idAuteur</u>	<u>idTitre</u>
integer	integer

Genres

<u>idGenre</u>	Genre
integer	varchar2(40)

Écriture de requêtes SQL

Écrire les requêtes SQL qui répondent aux interrogations ci-dessous :

Fin requête

Les titres des livres de la base	SELECT Titre FROM Livres ;
Les titres et années de parution des livres	
Le nombre d'auteurs contenu dans la base	
Les noms et prénoms de tous les auteurs par ordre alphabétique de nom.	
Les noms des auteurs dont le prénom est Pierre	
Les titres et années de parution des livres parus de 2010 à 2015 ordonnés suivant l'année de parution de manière décroissante.	

