



1 Le SQL et Logiciel DB Browser (SQLite)

Maintenant que nous avons vu ce qu'était une Base de Données, nous avons besoin d'un langage et d'un logiciel pour construire une Base De Données, créer des attributs, ajouter de données et modifier des données. Ce langage doit être capable de faire des requêtes. C'est à dire interroger une base de données afin d'obtenir des informations.

Nous allons utiliser le SQL (Structured Query Language). Le SQL est le langage le plus utilisé pour manipuler les bases de données relationnelles.

Le logiciel que nous utilisons est DB Browser que l'on peut télécharger sur <https://sqlitebrowser.org/>

- Lancer DB Browser (SQLite)
- Cliquer sur *Nouvelle Base de Données*
- Donner un nom à votre base de donnée : **maBaseFilm.db**

2 Les requêtes de création et transformation

2.1 Créer une table

Nous voulons créer la table **Film** dont le schéma est

Film : $\{(idFilm, entier), (titre, Texte), (idRéalisateur, entier)\}$

- Une première méthode : pour créer la table **Film**, on édite la définition de la table. Si celle ci n'apparaît pas, aller dans *Structure de la base de données* puis *Créer une table*.

Éditer la définition de la table

Table

Film

Avancé

Champs

Ajouter Supprimer Monter au début Monter Descendre Descendre à la fin

Nom	Type	NN	CP	IA	U	Défaut
idFilm	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
titre	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
idRéalisateur	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
CREATE TABLE "Film" (  
  "idFilm" INTEGER NOT NULL UNIQUE,  
  "titre" TEXT,  
  "idRéalisateur" INTEGER,  
  PRIMARY KEY("idFilm" AUTOINCREMENT)  
);
```

OK Annuler

- Une deuxième méthode en écrivant la requête (c'est à dire la demande) SQL en choisissant

Executer le SQL.

Créer la table **Artiste** dont le schéma est

Artiste : $\{(idArtiste, entier), (nom, Texte), (prénom, Text)\}$

en recopiant dans l'éditeur :

```
CREATE TABLE 'Artiste' (
  'idArtiste' INTEGER NOT NULL UNIQUE,
  'nom' TEXT,
  'prénom' TEXT,
  PRIMARY KEY ('idArtiste' AUTOINCREMENT)
);
```



Remarque

| Le SQL est très sensible au copié-collé, parfois il vaut mieux tout retaper

Lancer l'exécution en appuyant sur le triangle d'exécution ►.

Remarquer que les mots clés de SQL sont à écrire en MAJUSCULE!

2.2 Insertion des données

L'insertion des données se fait avec la syntaxe :

```
INSERT INTO 'tableau'
('attribut1', 'attribut2', 'attribut3', ...)
VALUES
(value1, value2, value3, ...),
(value1, value2, value3, ...),
...
;
```

Vous pouvez ainsi insérer des données de la table **film** en copiant collant le code suivant :

```
INSERT INTO 'Film'
('idFilm', 'titre', 'idRéalisateur')
VALUES
(1, 'La guerre des étoiles', 25),
(2, 'Apocalypse Now', 45),
(3, 'Impitoyable', 1),
(4, 'La vie est belle', 24),
(5, 'La vie est belle', 24),
(6, 'Le parrain', 4),
(8, 'Sur la route de madison', 1),
(9, 'La guerre des mondes', 32),
(10, 'La vie d Adèle', 105)
;
```

Vérifier en appuyant sur l'onglet *Parcourir les données*.



Exercice 1

Insérer les données de la table **artiste** suivante EN GARDANT LES FAUTES D'ORTHOGRAPHES :

Artiste			
idArtiste	nom	prénom	naissance
45	'Ford Copola'	'Francis'	1939
1	'Estwod'	'Clint'	1930
25	'Luca'	'George'	1944
24	'Benini '	'Robertto '	1952
4	'Donald'	'Duck'	1934
32	'Steven'	'Spielberg'	1946
105	'Abdellatif'	'Kechiche'	1960

2.3 Modification des données

2.3.1 Suppression

On peut supprimer des données avec la requête :

```
DELETE FROM
'table1'
WHERE
condition
;
```

On peut par exemple supprimer l'enregistrement 5 de la table Film qui est un doublon de la "La vie est belle".

```
DELETE FROM
'Film'
WHERE
idFilm=5
;
```

Vérifier en appuyant sur l'onglet *Parcourir les données*.



Exercice 2

| Supprimer l'artiste non réel de la table *Artiste*.

2.3.2 Modification

On peut modifier des données avec la requête :

```
UPDATE
'tableau'
SET
attribut=valeur
WHERE
condition
;
```

Par exemple "Le parrain" est de Francis Ford Coppola et pas de Donald Duck.

```

UPDATE
'Film'
SET
idRéalisateur=45
WHERE
titre='Le parrain'
;

```



Exercice 3

| Il y a une faute d'orthographe sur chaque réalisateur. Corrigez là avec UPDATE.

3 Les requête de recherche

3.1 Les requêtes simples (sur une seule table)

Pour que cela soit plus intéressant nous allons travailler sur une table plus fournie que celle que vous avez fabriquée mais gardez là pour comprendre les requêtes complexes.

Chargez la base de données de films [FilmCnam.db](#)

Son schéma est

```

Film{idFilm, titre, année, genre, résumé, idRéalisateur, codePays}
Pays{code, nom, langue}
Artiste{idArtiste, nom, prénom, annéeNaiss}
Rôle{idFilm, idActeur, nomRôle}
Internaute{email, nom, prénom, région}
Notation{email, idFilm, note}

```

Mais pour l'instant on ne s'intéressera qu'à la table Film.

On affiche les données avec SELECT suivant la syntaxe suivante :

```

SELECT
titre, année, genre
FROM
Film
;

```

On peut tout afficher avec * :

```

SELECT
*
FROM
Film
;

```

On va maintenant faire des requêtes conditionnelles.

Les conditions sont exprimées dans la clause WHERE. La sélection peut se faire sur 1 ou plusieurs critères à l'aide :

- Des opérateurs de comparaison (<, >, =, <=, >=, <>)
- Des opérateurs logiques (AND, OR, NOT)
- Des opérateurs arithmétiques (+, -, *, /, ROUND, ABS, SQRT)

- Des opérateurs IN(Appartenance), BETWEEN(Intervalle, bornes incluses), LIKE(Comparaison de chaînes), NULL.

On peut aussi utiliser la clause HAVING dont les principaux opérateurs sont :

AVG, COUNT, MAX, MIN, SUM

La clause GROUP BY est nécessaire dès que l'on utilise des fonctions de calculs statistiques avec des données brutes. Cette clause regroupe les occurrences ayant même valeur pour l'attribut(s) spécifié(s) dans la clause Group By et renvoie une seule ligne par groupe. Tous les attributs du SELECT et du GROUP BY doivent être identiques. Par défaut, GROUP BY trie par ordre croissant (ASC) de ses attributs. On utilise DESC pour un tri décroissant.

On utilise AS pour donner un nom d'affichage d'un attribut ou d'une statistique.

**Exercice 4**

| Comparer

```

SELECT      codePays
FROM        Film

SELECT      DISTINCT codePays AS 'code du Pays'
FROM        Film

```

Sur ce modèle, lister distinctement tous les genres et renommer la colonne en "type de film".

**Exercice 5**

| Essayer

```

SELECT
    *
FROM
    Film
WHERE
    titre='Gladiator'
;

```

Sur ce modèle, Chercher les caractéristiques de Terminator.

**Exercice 6**

| Essayer

```

SELECT
    titre, année, genre
FROM
    Film
WHERE
    année BETWEEN 1980 AND 2000
;

```

Cherchez les films entre 1900 et 1945 et changez la colonne 'titre' en 'Vieux films'.

**Exercice 7**

| Essayer

```

SELECT
    COUNT (codePays) , codePays
FROM
    Film
GROUP BY
    codePays
;

```

Sur ce modèle, Comptez le nombre de films dans chaque genre.

**Exercice 8**

| Essayer puis Enlever DISTINCT et comparer.

```

SELECT
    DISTINCT genre
FROM
    Film
ORDER BY genre ASC
;

```

Sur ce modèle, Faire la liste de tous les codePays et classer les par ordre alphabétique.



Exercice 9

Pour faire une recherche sur un mot en particulier. Le symbole % est un joker qui peut symboliser n'importe quelle chaîne de caractères.

```

SELECT
    DISTINCT titre
FROM
    Film
WHERE titre LIKE '%guerre%';
;

```

Sur ce modèle, Faire la liste de tous les titre contenant le mot "vie".



Exercice 10

Pour faire une opération élémentaire..

```

SELECT
    idFilm*5+3 as "Calcul sans intérêt"
FROM
    Film
;

```

Sur ce modèle, Donner le nom et l'âge des réalisateurs.



Exercice 11

Faire les recherches suivantes :

1. Titre de tous les Drames.
2. Titre de tous les Drames Français et nommer la colonne 'Film Français'.
3. Titre et année de tous les Drames Français après 1945.
4. Moyenne d'âge des réalisateurs.

3.2 Les requêtes complexes (sur plusieurs tables)

Pour bien comprendre, vous allez d'abord recharger la base données **maBaseFilm.db** qui ne contient que deux petites tables.

3.2.1 Le Produit Cartésien

Pour faire le produit cartésien de deux tables table1 et table2, on fait

```
SELECT
    attribut1,attribut2,attribut3,...
FROM
    table1,table2
;
```



Exercice 12

| Regarder le résultat de

```
SELECT
    *
FROM
    Film,Artiste
;
```

Évidemment cette opération agrège tous les résultats sans souci de sens. Pour faire une recherche dans cette table il faut faire

3.2.2 La Jointure Interne

On peut faire la jointure interne de différentes manières :

```
SELECT
    attribut1,attribut2,...
FROM
    table1 JOIN table2
WHERE
    condition
;
```

```
SELECT
    attribut1,attribut2,...
FROM
    table1 INNER JOIN table2
WHERE
    condition
;
```

```
SELECT
    attribut1,attribut2,..
FROM
    table1 , table2
WHERE
    condition
;
```



Exercice 13

| A quoi correspond le résultat de

```
SELECT
    titre,nom,prénom
FROM
    Artiste JOIN Film
WHERE
    idRéalisateur=idArtiste
;
```

```
SELECT
    titre,nom,prénom
FROM
    Artiste INNER JOIN Film
WHERE
    idRéalisateur=idArtiste
;
```

```
SELECT
    titre,nom,prénom
FROM
    Artiste, Film
WHERE
    idRéalisateur=idArtiste
;
```



Le cas des attributs avec le même nom

Il peut arriver que deux tables d'une même base de données contiennent un attribut avec le même nom. Pour éviter ambiguïtés lors d'une jointure, on utilise la syntaxe

table.attribut

pour nommer un attribut d'une table.

Créons la table Rôle dont le schéma est

Rôle : {(idFilm, entier), (idActeur, entier), (nom, Texte)}

La clé primaire est un couple ('idFilm', 'idActeur') et il y a une clé étrangère 'idActeur'. Cette dernière est liée à la table Artiste avec la clé idArtiste. On charge la table avec le code

```
CREATE TABLE 'Rôle' (
    'idFilm' INTEGER NOT NULL,
    'idActeur' INTEGER NOT NULL,
    'nom' TEXT,
    PRIMARY KEY ('idFilm' , 'idActeur')
);
```

```
INSERT INTO 'Rôle'
('idFilm', 'idActeur', 'nom')
VALUES
(1, 10, 'Han Solo'),
(2, 35, 'Tyrone Miller'),
(3, 1, 'William Munny'),
(4, 24, 'Guido Orefice'),
(6, 78, 'Vito Corleone'),
(8, 1, 'Robert Kincaid')
;
```

Et ajoutons quelques acteurs dans la table Artiste.

```
INSERT INTO 'Artiste'
('idArtiste', 'nom', 'prénom')
VALUES
(10, 'Ford', 'Harrisson'),
(35, 'Fishburne', 'Laurence'),
(78, 'Brando', 'Marlon'),
;
```

On peut voir maintenant les personnages de tous les films de la base avec la requête

```
SELECT
    titre, nom
FROM
    Rôle JOIN Film
WHERE
    Rôle.idFilm=Film.idFilm
;
```




Commentaires

Pour faire des commentaires :

- Sur une seule ligne avec deux tirets - - ,
- Sur plusieurs lignes avec les balises /* et */ :
/*
Mon commentaire
sur plusieurs lignes
*/

**Exercice 14**

Pour faire les exercices suivants vous les enregistrerez dans un fichier de requêtes appelé **mesRequetes-Film.sql** que vous ouvrirez en appuyant sur le bouton . Vous mettrez en commentaires les requêtes réussies.

Faites les requêtes suivantes :

1. Quel(s) rôle(s) a fait Marlon Brando? Utiliser seulement les tables Rôle et Artiste.
2. Qui est le réalisateur de 'La guerre des étoiles'?
3. Qui a joué le rôle de Han Solo?
4. Quel films ont un réalisateur qui joue dans le film? Faire apparaître le nom du réalisateur et l'identifiant du film.
5. Titre des films dans lesquels a joué Clint Eastwood. Donner aussi le rôle.
6. Dans quel film Clint Eastwood a-t-il joué le rôle de Bill Munny?

On peut maintenant s'entraîner à fond sur le site du CNAM : <http://deptfod.cnam.fr/bd/tp/>. Dans le cas où celle-ci ne fonctionne pas vous pouvez utiliser [FilmCnam.db](#) et répondre aux questions suivantes :

**Exercice 15**

1. Nom et année de naissance des artistes nés avant 1950. (solution)
2. Titre de tous les drames (solution)
3. Quels rôles a joué Bruce Willis (solution)
4. Qui est le réalisateur de Memento (solution)
5. Quelles sont les notes obtenues par le film Fargo (solution)
6. Qui a joué le rôle de Chewbacca? (solution)
7. Dans quels films Bruce Willis a-t-il joué le rôle de John McClane? (solution)
8. Nom des acteurs de 'Sueurs froides' (solution)
9. Quelles sont les films notés par l'internaute Prénom1 Nom1 (solution)
10. Films dont le réalisateur est Tim Burton, et l'un des acteurs Jonhhy Depp. (solution)
11. Titre des films dans lesquels a joué Woody Allen. Donner aussi le rôle. (solution)
12. Quel metteur en scène a tourné dans ses propres films? Donner le nom, le rôle et le titre des films. (solution)
13. Titre des films de Quentin Tarantino dans lesquels il n'a pas joué (solution)
14. Quel metteur en scène a tourné en tant qu'acteur? Donner le nom, le rôle et le titre des films dans lesquels cet artiste a joué. (solution)
15. Donnez les films de Hitchcock sans James Stewart (solution)
16. Dans quels films le réalisateur a-t-il le même prénom que l'un des interprètes? (titre, nom du réalisateur, nom de l'interprète). Le réalisateur et l'interprète ne doivent pas être la même personne. (solution)
17. Les films sans rôle (solution)
18. Quelles sont les films non notés par l'internaute Prénom1 Nom1 (solution)
19. Quels acteurs n'ont jamais réalisé de film? (solution)
20. Quelle est la moyenne des notes de Memento (solution) id, Nom et prénom des réalisateurs, et nombre de films qu'ils ont tournés. (solution)
21. Nom et prénom des réalisateurs qui ont tourné au moins deux films. (solution)
22. Quels films ont une moyenne des notes supérieure à 7 (solution)