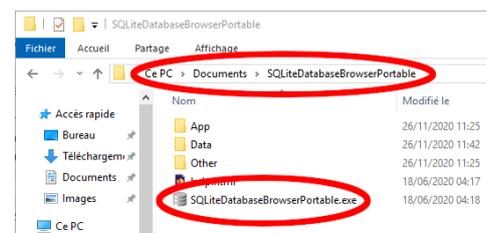
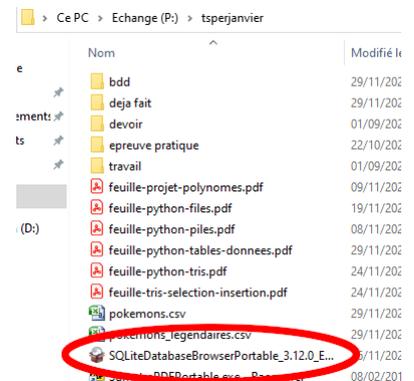
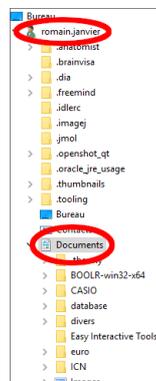
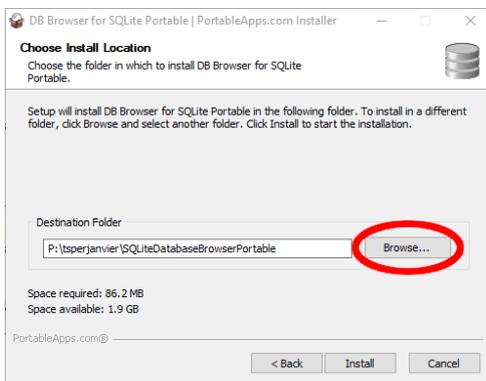


Découverte de SQL

Installation du logiciel

Afin de manipuler des bases de données, nous allons utiliser DB Browser qui manipule des bases SQLite. Dans le dossier Echange du groupe, vous trouverez une version portable s'appelant SQLiteDatabaseBrowserPortable_XXX.exe.

EXERCICE 1 : Copiez ce fichier dans votre dossier personnel ou exécutez le directement afin d'installer le logiciel. Lorsqu'on vous demande où l'installer, vérifiez bien que vous le faites dans vos documents. Vous pouvez alors lancer le logiciel.



EXERCICE 2 :

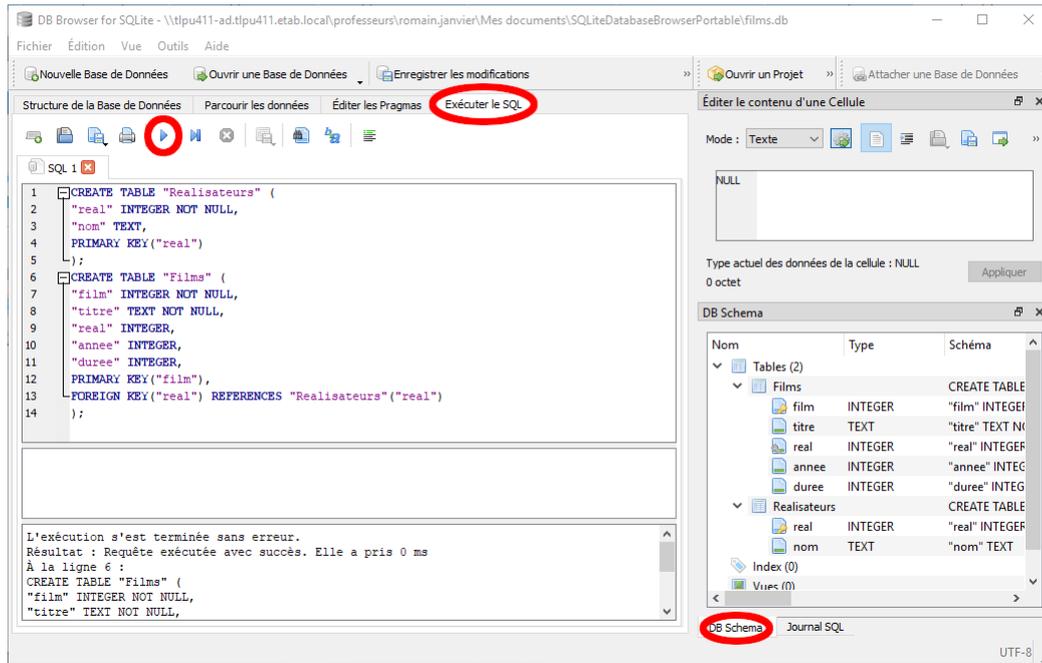
1) Créez une nouvelle base que vous appellerez films.db. Vous pouvez fermer la fenêtre qui s'ouvrira.



2) Allez dans l'onglet "Exécuter le SQL" et copier le code ci-dessous :

```
CREATE TABLE "Realisateurs" (  
    "real" INTEGER NOT NULL,  
    "nom" TEXT,  
    PRIMARY KEY("real")  
);  
  
CREATE TABLE "Films" (  
    "film" INTEGER NOT NULL,  
    "titre" TEXT NOT NULL,  
    "real" INTEGER,  
    "annee" INTEGER,  
    "duree" INTEGER,  
    PRIMARY KEY("film"),  
    FOREIGN KEY("real") REFERENCES "Realisateurs"("real")  
);
```

- 3) Exécuter le code, puis, en cliquant sur l'onglet "DB Schema" vérifier que les deux tables ont bien été créées.

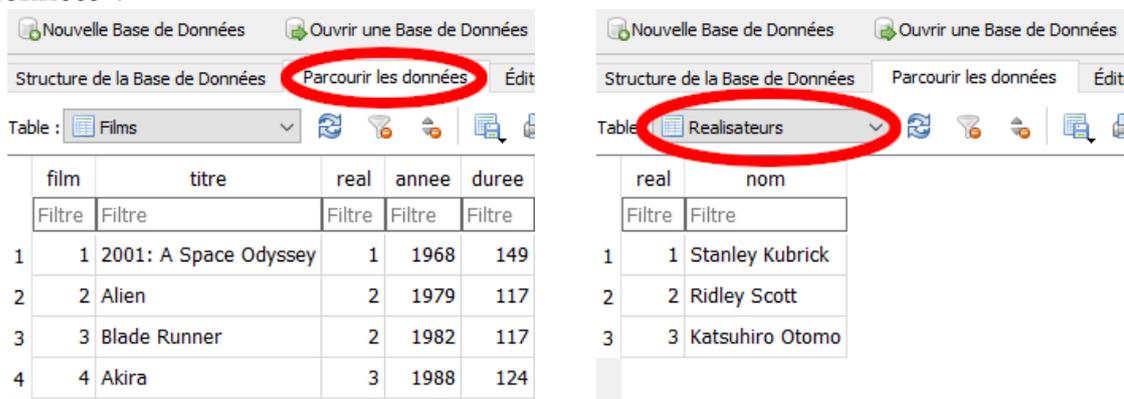


- 4) Vous pouvez maintenant effacer les commandes pour mettre les suivantes :

```
INSERT INTO "Realisateurs" VALUES(1, "Stanley Kubrick");
INSERT INTO "Realisateurs" VALUES(2, "Ridley Scott");
INSERT INTO "Realisateurs" VALUES(3, "Katsuhiro Otomo");
INSERT INTO "Films" VALUES(1, "2001: A Space Odyssey", 1, 1968, 149);
INSERT INTO "Films" VALUES(2, "Alien", 2, 1979, 117);
INSERT INTO "Films" VALUES(3, "Blade Runner", 2, 1982, 117);
INSERT INTO "Films" VALUES(4, "Akira", 3, 1988, 124);
```

Contrairement à Python, lorsque vous exécutez de nouvelles commandes, les anciennes ne sont pas annulées. La base de données est sauvegardée dans un fichier indépendant et toute modification faite sur cette base est permanente.

- 5) Vous pouvez vérifier que les données sont bien rentrées en allant dans l'onglet "Parcourir les données".

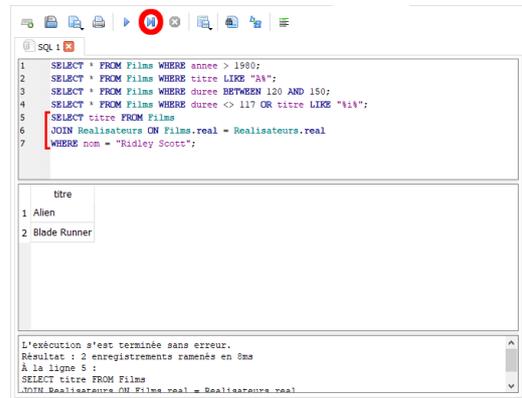
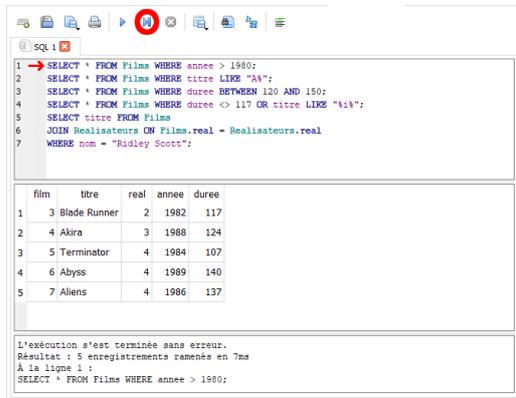


EXERCICE 3 : À l'aide d'une commande SQL, rajouter le réalisateur "James Cameron".

EXERCICE 4 : Rajouter ces trois films qu'il a réalisés :

- "Terminator", sorti en 1984 et durant 107 minutes ;
- "Abyss", sorti en 1989 et durant 140 minutes ;
- "Aliens", sorti en 1986 et durant 137 minutes.

Pour les exercices suivants, vous pouvez vous inspirer des requêtes présentées dans la feuille de cours sur SQL. Vous pouvez mettre les requêtes les unes à la suite des autres en mettant bien ";" à la fin de chacune. Pour plus de lisibilité, une requête peut être sur plusieurs lignes. Vous pouvez exécuter la requête de votre choix en mettant le curseur dessus et en appuyant sur le bouton indiqué ci-dessous.



EXERCICE 5 : Faire une requête pour faire afficher le titre, la durée et l'année de sortie de chacun des films.

EXERCICE 6 : Faire une requête pour obtenir la liste des titres de films de plus de 120 minutes.

EXERCICE 7 : Faire une requête pour obtenir la liste des titres et des années de sortie des films commençant par "Alien".

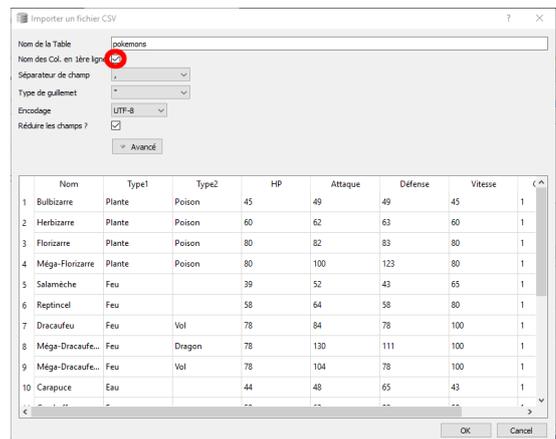
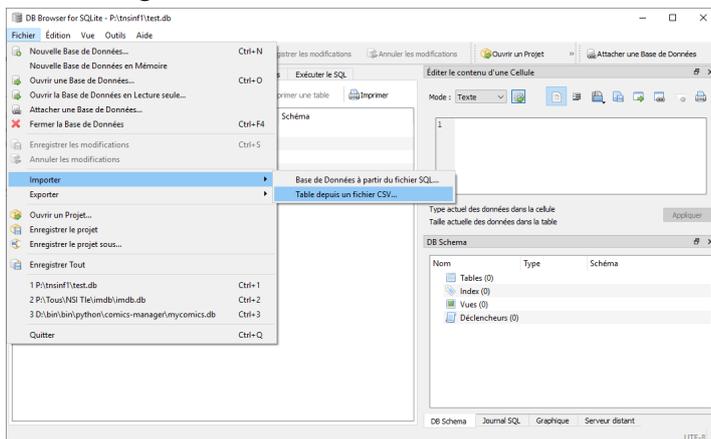
EXERCICE 8 : Faire une requête pour obtenir la liste des films classés par année de sortie. Pour cela, vous devez rajouter à la fin de votre requête **ORDER BY** attribut.

EXERCICE 9 : Faire une requête pour obtenir la liste des films classés par ordre inverse de durée. Pour cela, vous devez rajouter à la fin de votre requête **ORDER BY** attribut **DESC**.

EXERCICE 10 : Faire une requête pour obtenir la liste des noms des réalisateurs ayant fait des films de moins de 120 minutes. Attention, les noms ne doivent apparaître qu'une seule fois. Il faut faire une jointure.

Importation d'une table de Pokémon

EXERCICE 11 : Créer une nouvelle base de données, que vous appellerez pokemons.db. Nous allons importer dedans le fichier pokemons.csv. Attention, il faut bien cocher "Nom de Col. en 1ère ligne".



EXERCICE 12 : Faire une requête pour obtenir les informations de "Bulbizarre".

EXERCICE 13 : Faire une requête pour obtenir la vitesse de "Bulbizarre". Notez bien cette valeur.

EXERCICE 14 : Faire une requête pour obtenir le nom et la vitesse de tous les Pokémons strictement plus rapides que Bulbizarre. Vous pouvez utiliser la réponse de l'exercice précédent.

Au lieu d'écrire le résultat de la requête précédente, il est possible de remettre cette requête entre parenthèses à la place de la valeur. On parle alors de **requêtes imbriquées**. Dans ce cas, il faut que la requête renvoie une seule valeur. Ce type de requêtes est hors programme, mais cela permet de voir la puissance de SQL.

EXERCICE 15 : Faire une requête pour obtenir le nom et la vitesse de tous les Pokémons strictement plus rapides que Bulbizarre en utilisant des requêtes imbriquées. Vous pouvez compléter l'exemple ci-dessous.

```
SELECT ... FROM ... WHERE ... > (SELECT ... FROM ... WHERE ...)
```

EXERCICE 16 : Faire une requête pour obtenir tous les Pokémons dont le nom comporte exactement 4 lettres.

EXERCICE 17 : Faire une requête pour obtenir le nom de tous les Pokémons de type Feu. Attention, les types sont répartis dans deux colonnes.

EXERCICE 18 : Faire une requête qui affiche tous les noms de Pokémons triés par ordre alphabétique.

EXERCICE 19 : Faire une requête qui affiche les noms et l'attaque de tous les Pokémons triés par ordre décroissant d'attaque puis par ordre alphabétique. Il est possible de mettre plusieurs attributs dans **ORDER BY** en les séparant par des virgules et **DESC** ne concerne que l'attribut après lequel il se trouve.

EXERCICE 20 : Faire une requête qui affiche les 5 Pokémons les plus rapides. Pour limiter le nombre de résultats aux n premiers, il faut ajouter **LIMIT n**.

EXERCICE 21 : Faire une requête pour obtenir le nom et la vie de tous les Pokémons dont la vie est entre 50 et 60. Afficher les résultats par ordre décroissant de vie, puis par ordre alphabétique.

EXERCICE 22 : La commande **SELECT AVG(attribut) FROM table** renvoie la valeur moyenne de attribut. Faire une requête pour obtenir la valeur moyenne de la défense des Pokémons.

De la même manière, il existe des fonctions **MAX(attribut)** et **MIN(attribut)**.

EXERCICE 23 : Déterminer la vitesse minimale et maximale des Pokémons de type Vol.

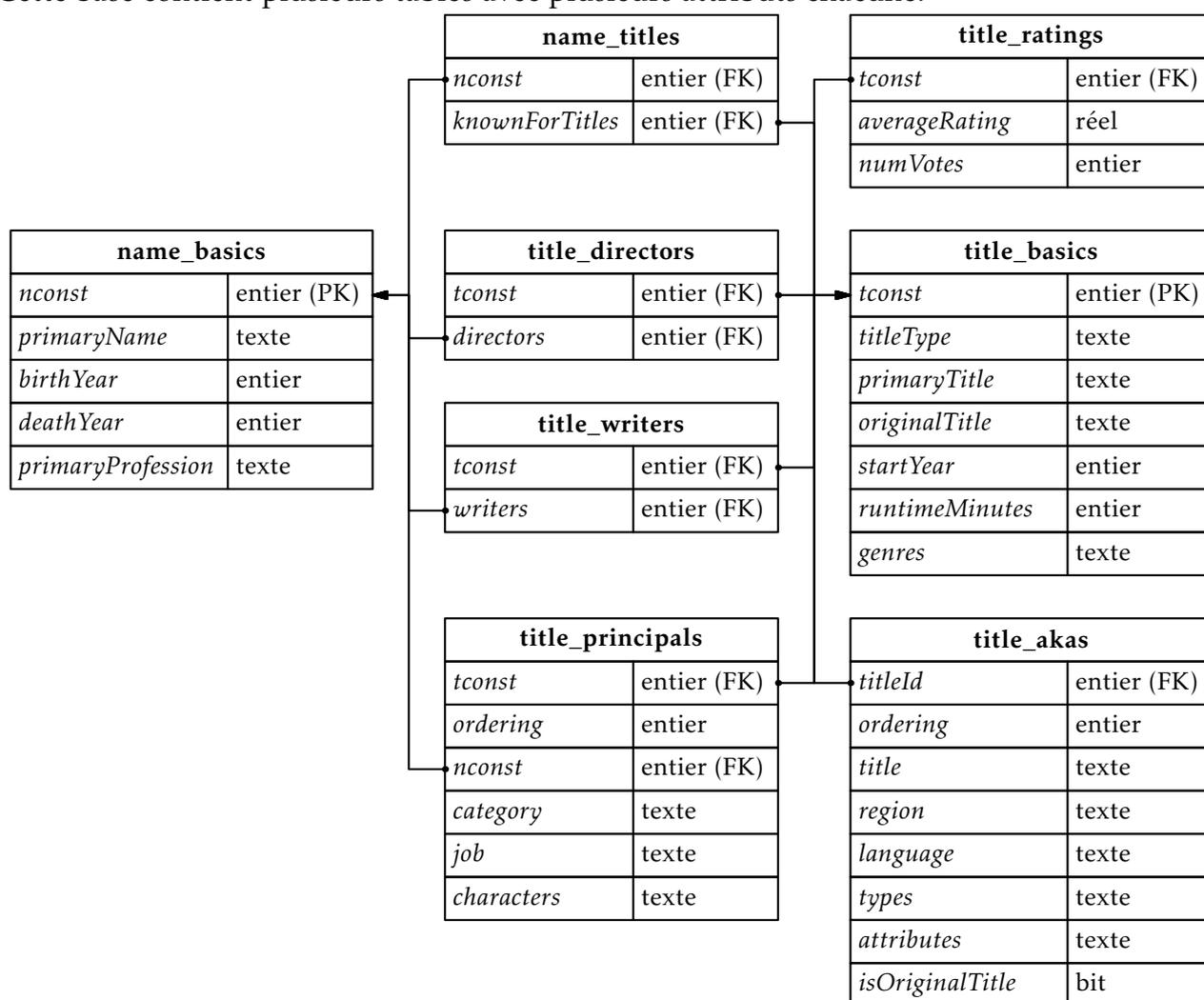
EXERCICE 24 : Il est possible de faire des calculs afin d'obtenir de "nouveaux" attributs. Il est également possible de donner un nom à ces attributs. Testez la commande suivante et expliquer ce qu'elle fait :

```
SELECT nom, attaque - défense AS différence  
FROM pokemons WHERE attaque > défense  
ORDER BY différence DESC
```

Avec une base de données plus massive

Nous allons maintenant utiliser une base de données contenant des dizaines de milliers de films, de réalisateurs, d'acteurs et de scénaristes.

EXERCICE 25 : Copier sur le bureau ou sur une clef USB la base de données s'appelant `imdb.db` se trouvant dans le dossier `bdd` dans le dossier de votre groupe dans `Echange`. Cette base contient plusieurs tables avec plusieurs attributs chacune.



Le nombre de résultats étant souvent très important, il est souvent plus intéressant de demander combien il y a de réponses que de les regarder. Pour cela il faut utiliser la fonction `COUNT(...)` d'une des manières ci-dessous :

```
SELECT COUNT(*) FROM table
SELECT COUNT(*) FROM table WHERE ...
SELECT COUNT(attribut) FROM table WHERE ...
SELECT COUNT(DISTINCT attribut) FROM table WHERE ...
```

EXERCICE 26 : La table `name_basics` contient les noms des personnes (réalisateurs, scénaristes, acteurs...) de la base. Déterminer le nombre de personnes dans cette table.

EXERCICE 27 : Certaines personnes sont des homonymes.

- 1) Déterminer le nombre de personnes s'appelant David Smith.
- 2) Déterminer le nombre d'acteurs s'appelant David Smith. Pour cela, il faut chercher si `actor` se trouve dans `primaryProfession`.

EXERCICE 28 : La table **title_ratings** contient la note moyenne (sur 10) donnée sur Imdb aux films, ainsi que le nombre de votes.

- 1) Déterminer le nombre de votes moyens.
- 2) Déterminer la moyenne des notes moyennes.
- 3) Déterminer la note maximale.
- 4) Déterminer le nombre maximal de votes.
- 5) Déterminer combien de films ont eu plus d'un million de votes.

EXERCICE 29 : La table **title_basics** contient le titre des films, l'année de sortie et leur durée.

- 1) À l'aide d'une jointure, lister les 100 films ayant les meilleures moyennes, classés par ordre décroissant de note.
- 2) Lister les films, les notes et le nombre de votes des films ayant plus d'un million de votes, classés par ordre décroissant de nombre de votes.
- 3) Lister les titres des films ayant une moyenne supérieure à 9 avec plus de 10 000 votes.

EXERCICE 30 : La table **title_akas** contient les titres des films dans les différentes langues.

- 1) En n'utilisant que cette table, déterminer le numéro identifiant le film "Le fabuleux destin d'Amélie Poulain".
- 2) Déterminer le titre de travail de ce film. Il faut que le type soit "working".
- 3) Combien est-ce qu'il y a de titres différents pour ce film?
- 4) Combien est-ce qu'il faut de symboles pour écrire le titre en japonais?

EXERCICE 31 : La table **title_directors** permet d'associer chaque film avec son, ou ses, réalisateur(s). Il faut faire une double jointure pour pouvoir associer le nom d'un réalisateur avec le titre d'un de ses films.

- 1) Lister tous les titres et réalisateurs de films dont le titre contient "Ninja".
- 2) Les classer par ordre chronologique.
- 3) En faisant une nouvelle jointure avec **title_ratings**, les classer par ordre décroissant de notes, en ne gardant que ceux qui ont plus de 1 000 votes.

EXERCICE 32 : Pour les questions suivantes, vous pouvez avoir à utiliser les autres tables et faire plusieurs jointures.

- 1) En quelle année est sortie le film "The Godfather"?
- 2) En quelle année est sortie le premier film "Superman"?
- 3) Quel est le titre original du film "Les dents de la mer"?
- 4) Quel est le métier d'Olivier Nakache?
- 5) Quels sont les films d'Olivier Nakache?
- 6) Qui a écrit le scénario du film "Taxi" sorti en 1998?
- 7) Quelles sont les noms et rôles (category et job) des personnes intervenant dans la production du film "Return of the Jedi"?
- 8) Quelle sont les 5 comédies romantiques les mieux notées?
- 9) Quels sont les 10 films d'animation ayant reçu plus de 1 000 votes les mieux notés?
- 10) Combien de films durent plus de 3 heures?
- 11) Quelle est la durée moyenne d'un film?
- 12) Quel est le film le plus long?
- 13) Quels sont les 5 films les plus longs?
- 14) Quels sont les titres des films les plus connus de Sean Connery?
- 15) Quels sont les acteurs ayant joué le rôle de James Bond, et dans quels films?
- 16) Quel sont les réalisateurs ayant fait les cinq film les mieux notés? Indiquer les noms des films correspondants.