

# TP2 : Requêtes en langage SQL

## Informatique CPP 2A, Semestre 1

CPP

Octobre - Novembre 2015

## 1 Mise en place

La base de données que nous utiliserons se trouve dans un fichier `musicbrainz-FR.db` (base de donnée SQLite 3) à télécharger sur la page du cours. Les logiciels existants pour accéder à une base de données SQLite sont assez complexes. Vos enseignants ont préparé une interface graphique minimaliste, taillée sur mesure pour le CPP : `sql_runner`, également téléchargeable sur la page du cours.

### 1.1 Lancement de l'interface `sql_runner`

Téléchargez `musicbrainz-FR.db` et `sql_runner.zip` sur la page du cours. Extrayez l'archive `sql_runner.zip` : ceci crée un répertoire `sql_runner`, qui contient entre autres un fichier `sql_runner.py`. Ouvrez `sql_runner.py` dans Spyder, et exécutez-le. L'interface graphique doit se lancer.

### 1.2 Connexion à la base de données

Dans le menu « Fichier », choisissez « Ouvrir un fichier DB » ( `Ctrl` - `O` ), et sélectionnez le fichier `musicbrainz-FR.db` que vous venez de télécharger. Vous devriez voir apparaître en bas de la fenêtre « Connecté au fichier .../musicbrainz-FR.db ».

### 1.3 Première requête

En guise d'échauffement, nous allons maintenant lister les artistes de la base de données. Entrez la requête suivante dans le champ texte « Code SQL à exécuter » en haut de la fenêtre : `SELECT * FROM Artistes;` Puis cliquez sur le bouton « Exécuter la requête » (ou `Ctrl` - `E` ). Le résultat apparaît dans le champ « Retour du Système de Gestion de Bases de Données » :



Affichons maintenant la ligne du chanteur Renaud, seule. Entrez la requête suivante :

```
SELECT * FROM Artistes WHERE nomArtiste = 'Renaud';
```

Exécutez la requête : 1 seule ligne s'affiche.

Attention, les minuscules/majuscules comptent dans les chaînes de caractères, si vous écrivez `'renaud'` à la place de `'Renaud'`, vous n'aurez aucun résultat. Par contre, les mots clés peuvent être écrits indifféremment en minuscule ou majuscule (la convention est de les écrire en majuscule).

Vous pouvez écrire vos requêtes sur plusieurs lignes, cela ne fait pas de différence pour SQLite. Par exemple, la requête ci-dessus est peut-être plus lisible comme ceci :

```
SELECT *
FROM Artistes
WHERE nomArtiste = 'Renaud'
```

## 1.4 Comment s'organiser ?

L'interface `sql_runner` vous permet d'exécuter des requêtes, mais pas directement de les sauvegarder. Quand vous avez écrit une requête et vérifié qu'elle marche, copiez-collez la à un endroit où vous pourrez la sauvegarder : un document LibreOffice à sauvegarder sur votre compte Ensimag, ou un mail à vous envoyer à vous-mêmes.

## 2 Et maintenant ?

Reprenez l'énoncé de TD du TD1, et traduisez les questions en SQL. Exécutez chaque requête pour vérifier que vous obtenez les bons résultats.

Pour vérifier vos résultats, pour la plupart des requêtes, nous vous donnerons le nombre de tuples attendus.<sup>1</sup> `sql_runner` affiche ce nombre de tuples sur la première ligne de sa réponse.

## 3 Travailler en dehors des séances

`sql_runner` n'a pas de dépendance particulière et devrait donc marcher sur n'importe quelle machine ayant Python 3 installée. Vous devriez donc pouvoir reproduire les manipulations de ce TP sur vos machines personnelles et celles du CPP.

Bien sûr, vous pouvez toujours venir travailler directement dans les salles machines de l'Ensimag !

## 4 Pour les curieux : particularités de SQLite3

Le logiciel SQLite3 que nous utilisons ici est un peu particulier comparé aux systèmes de gestion de bases de données les plus utilisés comme MySQL, PostgreSQL ou Oracle. Ici, nous avons une base de données stockée dans un fichier que l'on utilise directement. Les systèmes classiques fonctionnent avec un serveur de bases de données : une machine, en général distincte de celle sur laquelle on travaille, à qui on envoie des requêtes SQL et qui nous renvoie le résultat. Avec ces systèmes, l'utilisateur n'a pas d'accès direct aux données, il ne peut rien faire autrement qu'en passant par des requêtes SQL. Nous utilisons SQLite qui est moins puissant mais plus simple à utiliser.

## 5 Dépannage

Si en suivant les instructions ci-dessus, vous n'arrivez pas à lancer l'interface ou à exécuter une requête SQL, prévenez vos enseignants (c'est peut-être un bug que nous pouvons corriger).

En attendant, vous pouvez essayer :

- Fermer la console « IPython » dans Spyder, et utilisez le menu « Consoles » → « Ouvrir une console Python », puis relancez `sql_runner`.
- `sql_runner` limite l'utilisation de la mémoire pour vous éviter de faire planter votre machine quand vous vous trompez dans une requête. Cette limite peut mal interagir avec Python. Vous pouvez la désactiver en mettant en commentaire la ligne `limit_memory_usage()` dans le fichier `sqlite_utils.py`.

---

1. Bien entendu, obtenir le bon nombre de tuples est une condition nécessaire mais pas suffisante. Cependant, si vous obtenez le bon nombre de tuples, il y a peu de chance que vous vous soyez trompé.