

Informatique tronc commun – TP n° 14

Bases de données et langage SQL

Mercredi 27 avril 2016

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
3. Vous ne rendrez pas de compte rendu : l'objectif est d'écrire *vous-même* le plus de programmes (fonctionnels!) possible.
4. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez les!
5. Vous devez être autonome. Ainsi, avant de poser une question à l'enseignant, merci de commencer par :
 - relire l'énoncé du TP (beaucoup de réponses se trouvent dedans);
 - relire les passages du cours¹ relatifs à votre problème;
 - effectuer une recherche dans l'aide disponible sur votre ordinateur (ou sur internet) concernant votre question.

Il est alors raisonnable d'appeler votre enseignant pour lui demander des explications ou une confirmation!

1 Introduction

Le but de ce TP est d'effectuer des requêtes sur une base de données. Nous utiliserons pour cela le client SQLite en ligne de commande, `sqlite3`.

Vous trouverez sur le site du TP une base de données `peche.sqlite`. Téléchargez-la et déposez-la dans un de vos répertoires.

Important : Vous ne parviendrez pas à ouvrir la base de données avec `sqlite` si celle-ci est située dans une partie de votre répertoire stockée sur le réseau (répertoire `~/Documents/`) en raison de limitations du système de fichiers réseau utilisé.

Pour lancer le client SQLite, ouvrez un terminal et tapez

```
sqlite3 pêche.sqlite
```

1. Dans le cas fort improbable où vous ne vous en souviendriez pas.

après vous être placé dans le répertoire où vous avez déposé `peche.sqlite`. Cela démarre le client SQLite. Celui-ci affiche quelques lignes d'aide puis affiche l'invite `sqlite>` pour signifier qu'il attend vos commandes.

2 La base de données utilisée

La base de données décrit les prises de pêches déclarées en France pour 2009 et 2010.² Elle contient 4 tables :

ESPECES Chaque ligne de cette table décrit une espèce animale et contient les champs suivants :

ESP_COD_FAO Code utilisé par la FAO³ pour décrire l'espèce considérée.

ESP_LIB_FAO_FRANCAIS Nom en français de l'espèce considérée.

groupe Groupe auquel appartient l'espèce.

LIEUX Chaque ligne de cette table décrit un lieu de pêche. Elle contient les champs suivants :

LIEU_REF Code pour décrire le lieu.

LIEU_LIB_REF Nom du lieu.

REGION_LIB_REF Nom de la région correspondante.

CAPT2009 Chaque ligne de cette table décrit une déclaration de prise de pêche pour l'année 2009. Elle contient les champs suivants :

id Un identifiant numérique interne à la base de données (utilisé comme clé primaire)

PAVILLON Le pavillon de la flotte de pêche (le pays où elle est immatriculée ?)

LIEU_REF Le lieu de pêche, sous forme du code du lieu.

ESP_COD_FAO Le code FAO de l'espèce.

QUANT_POIDS_VIF La masse pêchée.

MONTANT_EUROS Le prix de cette pêche.

CAPT2010 Même contenu que **CAPT2009** mais pour l'année 2010.

3 Travail demandé

La plupart des questions demandées peuvent être faites avec un simple listing papier des données. Le but du jeu est évidemment de faire faire ce travail à la machine en lui donnant les bonnes requêtes.

Q1 Pour obtenir les commandes qui ont été utilisées pour créer les tables (pas pour les remplir), vous pouvez utiliser la commande « `.schema` ». Il s'agit d'une commande spécifique à SQLite et non d'une requête SQL, en revanche, ce qu'elle affiche est une suite de commandes SQL qui permettent de créer les tables.

Utilisez-la pour comprendre comment ont été créés les tables.

Le code FAO est-il un entier ou une chaîne de caractère ? Même question pour le code de lieu.

2. Source : <http://data.gouv.fr>.

3. *Food and Agriculture Organization* des Nations-Unies.

- Q2** Obtenir la liste des lieux de pêche.
- Q3** Quel est le code du port de 'Nice' ?
- Q4** Afficher les prises réalisées à Nice en 2009.
- Q5** Quels sont les différents groupes d'espèces existantes ? On fera attention à ne lister qu'une fois chaque groupe.
- Q6** Combien y en a t-il ?
- Q7** Lister toutes les déclarations de prises pour 2009 d'espèces appartenant au groupe des Poissons **diadromes**.
- Q8** Combien y a t-il de prises pour 2009 d'espèces appartenant au groupe des Poissons **amphibiotiques** ?
- Q9** Quelle est la masse totale des prises en 2009 appartenant au groupe des Poissons **amphibiotiques** ?
- Q10** Quel est, groupe par groupe, le montant total des prises en 2009 ?
- Q11** Quel est le nombre d'éléments de la table **LIEUX** ? Quel est le nombre de descriptions de lieux distincts ? En déduire qu'il y a des doublons (des lieux ayant plusieurs codes distincts) et les trouver (avec une requête bien choisie).
- Q12** Quel est, par lieu, le nombre d'espèces pêchées en 2009 ?
- Q13** Quel en est le maximum ? Dans quel lieu ce nombre est-il atteint ?
- Q14** Quels sont les lieux où on a pêché des 'Coquillages' en 2009 ? Ceux où on n'en a pas pêché ?

Remarques

Pour afficher l'aide dans **sqlite3**, vous pouvez utiliser la commande suivante.

```
.help
```

Pour afficher les entêtes des tables (noms des colonnes), vous pouvez utiliser la commande **.header** (voir l'aide).

Pour récupérer les résultats d'une requête sous forme de fichier lisible par un tableur, combinez **.mode** pour demander un affichage en CSV et **.output** pour sauver les résultats des requêtes dans des fichiers.

Enfin, on préférera souvent écrire ses requêtes dans un script SQL puis exécuter ces scripts via **sqlite3**. Une fois la requête écrite dans le script (on préférera lui donner comme extension **.sql**), on pourra procéder de deux manières.

— Lire les scripts dans **sqlite3**, en utilisant la commande **.read** comme suit.

```
.read monscript.sql
```

— Directement dans le terminal, on pourra écrire

```
sqlite3 peche.sqlite < monscript.sql
```

pour obtenir le résultat de la requête dans le terminal. On pourra préférer obtenir le résultat de la requête dans un fichier CSV (par exemple) en redirigeant ce résultat comme suit.

```
sqlite3 peche.sqlite < monscript.sql > res.csv
```