Ingénierie des systèmes d'information.
TP - SQL

Manon Ansart

ESIREM

Nous travaillons avec le système d'exploitation GNU/Linux. Normalement, SQLite et Python sont disponibles dans les salles GR01, GR05 et G06. De plus, depuis toutes les salles de l'Esirem, vous pouvez vous connecter sur le serveur GNU/Linux de la salle GR05 par ssh

```
ssh USERNAME@slocum.esirem-ad.ad.u-bourgogne.fr
```

ou bien

```
ssh USERNAME@10.169.20.12
```

1 Instructions

SQLite est une bibliothèque de gestion de base de données via le langage SQL. Vous pouvez téléchargez le code source et les fichiers binaires de SQLite ici : `https://www.sqlite.org/download.html`. La documentation officielle est disponible également sur le net : `https://www.sqlite.org/`.

SQLite est installé dans les salles de TP. L'interface standard de SQLite est appelée `sqlite3`, sa documentation est disponible via le système de mans des systèmes Unix : `man sqlite3`. Un grand nombre de langages de programmation fournissent des liaisons avec la bibliothèque SQLite : Smalltalk, Scheme, Tcl, D, C, Lua, Perl, R, Python, Go, Java, Closure, etc. Dans ce TP nous privilégerons Python, mais vous pouvez utiliser un autre language si vous le souhaitez.

Vous pouvez travailler sur une base de donnée `tp.db` en tapant dans votre terminal `sqlite3 tp.db`

Vous pouvez mettre toutes vos requêtes dans un fichier sql (par exemple requetes.sql), et executer toutes ces requêtes d'un coup dans `sqlite3` en faisant `.read requetes.sql`. Il est conseillé de procéder ainsi afin de conserver vos requêtes, et de pouvoir les modifier si elles donnent une erreur (vous ne pouvez pas accéder facilement à la requête précédente si vous la tapez directement).

Lors de l'**examen pratique** vous utiliserez le même système afin de rendre un fichier .sql qui contiendra toutes vos requêtes. À titre indicatif, les instructions exactes seront :

Vous rendrez :
- un fichier .sql (par exemple requetes.sql) contenant vos requêtes SQL
  - ce fichier doit pouvoir être appelé sans modifications dans `sqlite3` en faisant `.read requetes.sql`

— un fichier .py contenant le code python de la dernière partie, qui doit pouvoir être exécuté directement

Vous vous assurerez de respecter les points suivants :

— Travaillez sur une base de données `exam.db`
— Indiquez le numéro de la question en commentaire avant chaque partie correspondante, dans le fichier (.sql ou .py) dans lequel vous avez répondu.
— Chaque question doit avoir une réponse correspondant dans un des fichers (même les affichages). Tout numéro de question manquant sera considéré comme répondu
— Utilisez des commentaires pour justifiez vos réponses ou apporter des éléments supplémentaires quand c'est nécessaire.

2 Création de la base

L'objectif de ce TP est de créer une base de données pour un journal simple en appliquant les concepts vus en cours. Le journal est composé d'article. Chaque article possède un titre (pour plus de simplicité, le corps de l'article n'est pas stocké ici), et appartient à une rubrique (par exemple entertainment, business, sport, politics, tech). Chaque article peut être écrit par un journaliste ou plus. Un journaliste possède un nom et un prénom.

1. Proposer une modélisation de la base de données (tables, attributs, clés). Vous pouvez ajouter des attributs qui ne sont pas directement mentionnés dans le texte si ils vous semblent pertients.
2. Créer les tables que vous avez proposées dans votre base SQLite. Pour cela, vous réfléchirez aux contraintes d'intégrités qui vous semblent adaptées.
   — Si votre base de données contient des clés étrangères, réfléchissez à une gestion appropriée des suppressions et mises à jour (regardez les mots clés ON DELETE notamment). Voyez-vous des cas de suppressions qui sont mal gérés ?
   — Après la création de vos tables, vérifiez cette création en listant les tables de votre base puis en affichant le schéma de chaque table
   — Si vous utilisez un fchier .sql, ajoutez au début du fichier une suppression de chacune des tables si elle existe, afin de pouvoir relancer le .sql plusieurs fois sans erreur.
3. Créez des csv contenant les données de chacune des tables en générant les données de votre choix. Vous pouvez pour cela utiliser les données de la BBC : `http://mlg.ucd.ie/datasets/bbc.html` (utilisez les données de BBC, raw text files). Vous pouvez vous inspirer du fichier generation_csv.py fourni sur Teams
4. Utilisez les csv généré pour peupler vos tables sql. Vous pouvez par exemple voir `https://stackoverflow.com/questions/14947916/import-csv-to-sqlite`
5. Affichez les 10 premières lignes de chacune des tables pour vérifier l'insertion.
6. Vérifiez vos contraintes d'intégrité en inserant des tuples non valides, et en modifiant ou supprimant des lignes.

3 Manipulation de la base de données

Conservez bien chacune de vos requêtes dans un fichier sql

1. Affichez le nombre d'articles écrit par chaque journaliste (ainsi que ses informations)
2. Affichez le nombre d'articles écrit par chaque journaliste (ainsi que ses informations), en affichant d'abord les journalistes avec le plus d'articles. Nous n'avons pas vu ensemble comment trier alors qu'il y a un aggrégat, cela nécessite probablement une recherche google (en anglais).
3. Affichez le nombre d'articles dans la rubrique tech écrit par chaque journaliste (ainsi que ses informations).

4. Supprimez les articles de la catégorie tech écrits par le journaliste d'identifiant 10 (ou un autre identifiant de votre choix. Vous allez probablement avoir besoin d'une sous-requête en utilisant le mot clé "IN". Vous pouvez pour cela regarder les exemples de `https://www.geeksforgeeks.org/sql-subquery/`
5. Affichez les informations des journalistes ayant écrits au moins 30 (ou un nombre pertinent pour votre base) articles dans la catégorie tech
6. Affichez le nombre d'articles écrits par chaque journaliste (ainsi que ses informations) pour chaque rubrique. Nous n'avons pas vu ensemble comment faire un aggrégat sur plusieurs attributs, cela nécessite probablement une recherche google (en anglais).

4 Ajout de fonctionnalités en SQLite

1. Créez une vue articles_tech affichant uniquement les articles de la catégorie tech. Vérifiez que la vue est bien présente dans la base et affichez les 10 premières lignes pour vérifiez que le résultat est cohérent.
2. Affichez les id des journalistes ayant écrits des articles de la catégorie tech de 3 manières différentes :
   — sans utiliser la vue précédente, avec une sous-requête
   — en utilisant la vue précédente, avec une sous-requête
   — en utilisant la vue précédente, avec une jointure
   Dans tous les cas, assurez vous que les résultats soient présentés de façon agréable (id par ordre croissant, sans doublon). Le résultat doit être le même avec les 3 requêtes. Dans une question précédente vous avez retiré les articles tech d'un journaliste, vérifié que l'id de ce journaliste n'est pas retourné ici.

   Voyez-vous une autre manière de faire ce select ?
3. (a) Créez une table nb_articles_par_rubrique qui contient le nombre d'articles par rubrique. Vous pouvez vous aider de `https://www.techonthenet.com/sqlite/tables/create_table_as.php`
   (b) Ajoutez les triggers nécessaires pour mettre à jour cette table lors de l'ajout, suppression et mise à jour d'un article.
   (c) Vérifiez que vos triggers marchent en regardant votre table nb_articles_par_rubrique avant et après des suppressions, insertions et mises à jour.
   (d) Y a-t-il une autre solution pour accéder facilement à l'information du nombre d'article par rubrique ? Quelle solution est plus efficace dans quel situation ? Qu'est-ce qui vous semble le mieux pour un journal ?

   .....

   .....

   .....

   .....

   .....

5 Ajout de fonctionnalités externes

1. (a) En vous aidant de `https://docs.python.org/3/library/sqlite3.html`, connectez-vous à votre base de données, effectuez un select simple et affichez le résultat.
   (b) Le tutoriel utilise un curseur. À quoi servent les curseurs en base de données ? À quoi sert-il dans notre cas ?

.....

.....

.....

.....

.....

2. (a) En utilisant la même documentation, insérez 2 nouveaux journalistes et vérifiez le résultat.
   (b) À quoi sert le commit ? Tester avec et sans commit. Qu'est-ce qui change ?

.....

.....

.....

.....

.....

3. Ajouts de fonctions et aggrégats

Il est possible d'ajouter des fonctions utilisateur dans des bases de données. En SQLite, cela n'est pas possible depuis la base directement, mais uniquement depuis une application (Python par exemple). Il existe 2 types de fonctions utilisateur :

- Les fonctions scalaires, qui sont appliquées pour une ligne uniquement (une valeur par ligne si elle est appelée sur toutes les lignes). Ces fonctions sont créées avec `sqlite3` à l'aide de la fonction `create_function`.
- Les aggrégats, qui sont appliqués sur une table, et retourne donc une valeur pour la table. Les aggrégats sont créés avec `sqlite3` à l'aide de la fonction `create_aggregate`.

   (a) Ajoutez à votre base de donnée une fonction retournant la chaine de caractère "Prénom Nom" à partir du prénom et du nom d'un journaliste. Appliquez cette fonction à tous les journalistes et affichez le résultat. Vous pouvez vous aider des liens suivants :
   - `https://docs.python.org/3/library/sqlite3.html#sqlite3.Connection.create_function`
   - `https://www.geeksforgeeks.org/python-create-or-redefine-sqlite-functions/`

   (b) Créez une fonction Python qui, à partir d'un identifiant, retourne la chaine de caractère correspondant aux informations du journaliste. Cette fonction prendra uniquement en paramètre l'identifiant du journaliste, il faudra donc se connecter à la base à l'intérieur de la fonction.

   (c) Créez un aggrégat qui calcule la taille de la chaine la plus longue pour une colonne de chaines de caractères. Ajoutez cet aggrégat à votre base et testez-le sur les noms des journalistes. Vous pouvez vous aider de la doc : `https://docs.python.org/3/library/sqlite3.html#sqlite3.Connection.create_aggregate`

   (d) Créez une fonction Python qui calcule la taille du nom de journaliste le plus long en utilisant en appelant votre aggrégat.

   (e) On vous demande de créez une fonction Python qui retourne, pour un id de journaliste, sa rubrique principale (celle dans laquelle il a écrit le plus d'article). Utilisez vous une fonction ou un aggrégat ? Après avoir réfléchi par vous-même, vous pouvez vous aider en utilisant les étapes suivantes.

. . . . . .

. . . . . .

. . . . . .

i. De quelles tables avez-vous besoin ? Faut-il faire des jointures ?

. . . . . .

. . . . . .

. . . . . .

ii. A quoi ressemble la table la plus simple possible (obtenue avec le moins d'opération) dont vous auriez besoin en entrée de la fonction ou aggrégat ? Quelles sont les colonnes ? Peut-il y avoir plusieurs lignes ?

. . . . . .

. . . . . .

. . . . . .

6 Bonus

1. **(Question de reflexion, similaire à ce qui peut être demandé en examen)** On stocke le nombre de vues par article (dans une nouvelle table ou avec une nouvel attribut selon votre préférence). On veut ensuite regarder les performances des journalistes en utilisant cette métrique (avec une moyenne par exemple).
   (a) En vous inspirant de toutes les parties précédentes, proposer 3 manières de calculer le nombre de vues moyen par journaliste à partir du nombre de vues par article, sans qu'il soit nécessaire de taper une requête à chaque fois qu'on souhaite avoir cette information.

   . . . . . .

   . . . . . .

   . . . . . .

   . . . . . .

   . . . . . .

   (b) Comparez ces stratégies en terme de disponibilité. Dans quelle situation chaque startégie est la meilleure ? Dans quelle situation est-on ici ?

   . . . . . .

   . . . . . .

   . . . . . .

   . . . . . .

   . . . . . .

2. (Requête reprenant toutes les notions vues) Pour chaque journaliste, comptez le nombre d'articles que ce journaliste a écrit seul (qui n'ont pas d'autre auteur)

3. (Trigger avancé) Ajoutez un trigger qui, à la suppression d'un journaliste, vérifie qu'il n'a pas écrit d'article, et ne fait pas la suppression si le journaliste a écrit au moins un article.
4. (Requête qu'il aurait fallu ajouter pour que le système soit fiable) Ecrivez une requête permettant de vérifier si tous les articles sont bien écrits par au moins un journaliste