

Migration SQLite vers MySQL

Objectif

Cette documentation va expliquer comment effectuer la migration de SQLite vers un SGBD, ici MySQL.

Requis :

Demander à votre administrateur réseau de créer la base de donnée nécessaire a la configuration. Ainsi que d'avoir le paquet sqlite3 installer sur votre serveur web à côté de l'installation de web2py. Avoir une sauvegarde de l'application.

Migration des données :

On s'assure que notre base de donnée sur le SGBD est bien disponible.

Pour effectuer la migration, nous allons commencer par exporter les données de sqlite sous forme de script SQL.

On se place d'abord dans le dossier contenant le fichier de stockage sqlite.

```
# cd /var/www/web2py/applications/MRBS/databases
```

Puis on réalise un export grâce a la commande qui suit :

```
# sqlite3 storage.sqlite .dump > data.sql
```

A l'aide de votre éditeur de texte préféré (ici vim sera utilisée) on modifie le fichier afin de supprimer les commandes inutiles tel que les commandes *PRAGMA*, *BEGIN TRANSACTION*, *CREATE TABLE* et *COMMIT*. On retire également toutes les données concernant *sqlite_sequence*.

On remplace les guillemets double (") par des guillemets inversées dite backquotes (`) avec la commande :

```
:%s/"`/`/g
```

On supprime le contenu du dossier databases dans le dossier */var/www/web2py/applications/MRBS* a l'aide de la commande suivante :

```
# rm -rf /var/www/web2py/applications/MRBS/databases/*
```

[On va maintenant configurer web2py afin d'effectuer la connexion vers MySQL.](#)

On modifie le fichier **db.py** dans l'édition de l'application MRBS. On commente la ligne d'origine (ligne 11 ci-dessous) puis on ajoute une nouvelle ligne qui est la ligne 12 ci-dessous. Sa syntaxe est la suivante : `<code>mysql:<Login>:<Mot de Passe>@<Hôte>/<Base de Donnée></code>`

```
11     ## db = DAL('sqlite://storage.sqlite')
12     db = DAL(['mysql://Devs:M2LTest@172.21.101.12/mrbs'])
```

Après le test de la connexion de l'application cette erreur apparaît :

```
<class 'gluon.contrib.pymysql.err.ProgrammingError'> (1064, u"You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'accessible INT,\n categorie_id INT, INDEX categorie_id__idx (categorie_id), F0' at line 5")
```

A l'aide du post sur le blog du PPE ainsi que nos recherches personnelles, nous avons trouvé la solution qui consiste à ajouter `“,ignore_field_case=False, entity_quoting=True”` dans la fonction DAL.

```
11     ## db = DAL('sqlite://storage.sqlite')
12     db = DAL(['mysql://Devs:M2LTest@172.21.101.12/mrbs', ignore_field_case=False, entity_quoting=True])
```

La dernière étape consiste à réimporter les données de l'application. Pour cela nous allons tout d'abord nous connecter en ligne de commande au serveur de base donnée (se référer à la documentation [“Mise en place d'un SGBD”](#) partie test de connexion).

La syntaxe de cette commande est la suivante :

```
mysql -u "Utilisateur" -p -h "Hôte" "Nom BDD" < "fichier"
```

Dans notre cas :

```
mysql -u Devs -p -h 172.21.101.12 mrbs < data.sql
```

Après réalisation des tests primaires sur l'application, nos développeurs confirment le bon fonctionnement de l'application après la migration.

From:
<https://wiki.viper61.fr/> - **Viper61's Wiki**

Permanent link:
https://wiki.viper61.fr/sio/ppe2/mission_2/mgrt_sqlitemysql

Last update: **18/09/2016 02:54**