

Procédure d'installation et de configuration de MySQL Workbench

Introduction

I. Installation

- 1/ Python 3.12.0
- 2/ MySQL Workbench 8.0.40 (64bits)
- 3/ Vérification du lancement et des chemins d'installation

II. Lancement de l'application et configuration de la connexion

- 1/ Vérification du service MySQL80
- 2/ Modifier le fichier os_utils.py
- 3/ Configurer la connexion dans MySQL Workbench
- 4/ Redémarrez MySQL80

III. Désactivation de certaines options dans MySQL

- 1/ Modifier le fichier « *my.ini* »
- 2/ Redémarrez le service MySQL80 pour prendre en compte les modifications

IV. Importer un Dump

- 1/ Ouvrez le fichier dump avec Notepad++ pour vérifier le nom du schéma attendu
- 2/ Importer le Dump dans MySQL Workbench

Conclusion

Introduction

Dans le cadre de l'évolution du système d'information, cette activité avait pour *objectif de migrer une base de données vers une version plus récente de MySQL*. J'ai été chargée de participer à la mise en place de ce nouvel environnement, notamment en installant et configurant les outils nécessaires, puis en important une base existante à partir d'un dump.

I. Installation

1/ Python 3.12.0

Python a été installé en amont, car *certaines fonctionnalités de MySQL Workbench reposent sur ce langage*. Cela permet d'assurer le bon fonctionnement de l'application.

2/ MySQL Workbench 8.0.40 (64bits)

MySQL Workbench a ensuite été installé. Il s'agit d'un *outil graphique permettant d'administrer un serveur MySQL*, notamment pour gérer les bases de données, les utilisateurs et exécuter des requêtes.

3/ Vérification du lancement et des chemins d'installation

Une fois les installations terminées, MySQL Workbench a été lancé afin de vérifier son bon fonctionnement. Cette étape permet de détecter d'éventuelles erreurs liées à l'environnement, comme un problème de dépendances ou de configuration.

Les fichiers exécutables des différents composants sont accessibles via les chemins suivants :

- MySQL Workbench : C:\ProgramData\Microsoft\Windows\Start Menu\Programs\MySQL

- MySQL Installer : C:\ProgramData\Microsoft\Windows\Start Menu\Programs\MySQL\MySQL Installer - Community
- Python : C:\Windows\py

La vérification de ces emplacements permet de s'assurer que les logiciels sont correctement installés et accessibles par le système.

Si au lancement, tout fonctionne correctement, vous pouvez passer l'étape II.

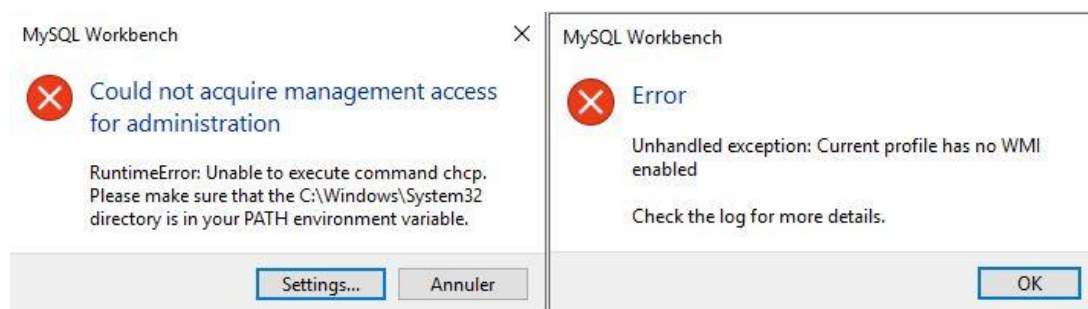
II. Lancement de l'application et configuration de la connexion

Lors du lancement de MySQL Workbench, des erreurs sont apparues empêchant l'accès aux fonctionnalités d'administration.

Les messages indiquaient notamment un problème lié à la variable d'environnement PATH ainsi qu'à l'absence de configuration WMI.

La variable PATH permet au système de *localiser les exécutables nécessaires au fonctionnement des applications*. Si elle est mal configurée, certaines commandes système ne peuvent pas être exécutées.

De plus, WMI (Windows Management Instrumentation) est un *composant de Windows permettant aux applications d'interagir avec le système*, notamment pour administrer des services comme MySQL. Son absence ou sa mauvaise configuration peut empêcher Workbench d'accéder aux fonctions d'administration.

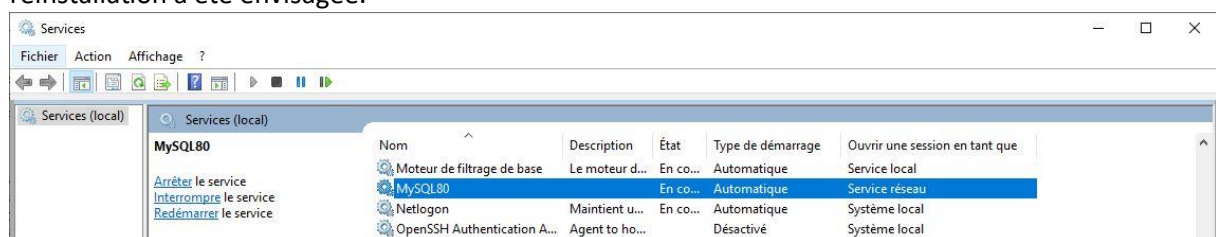


Cependant, après analyse, il est apparu que ces *messages pouvaient être trompeurs* et que le dysfonctionnement provenait en réalité d'une configuration interne de MySQL Workbench.

Afin de corriger les erreurs rencontrées lors du lancement de MySQL Workbench, plusieurs actions ont été réalisées :

1/ Vérification du service MySQL80

Le service MySQL80 a été vérifié via la console des services Windows (services.msc), car celui-ci doit être *actif* pour permettre le fonctionnement du serveur MySQL. En cas d'absence du service, une réinstallation a été envisagée.



2/ Modifier le fichier os_utils.py

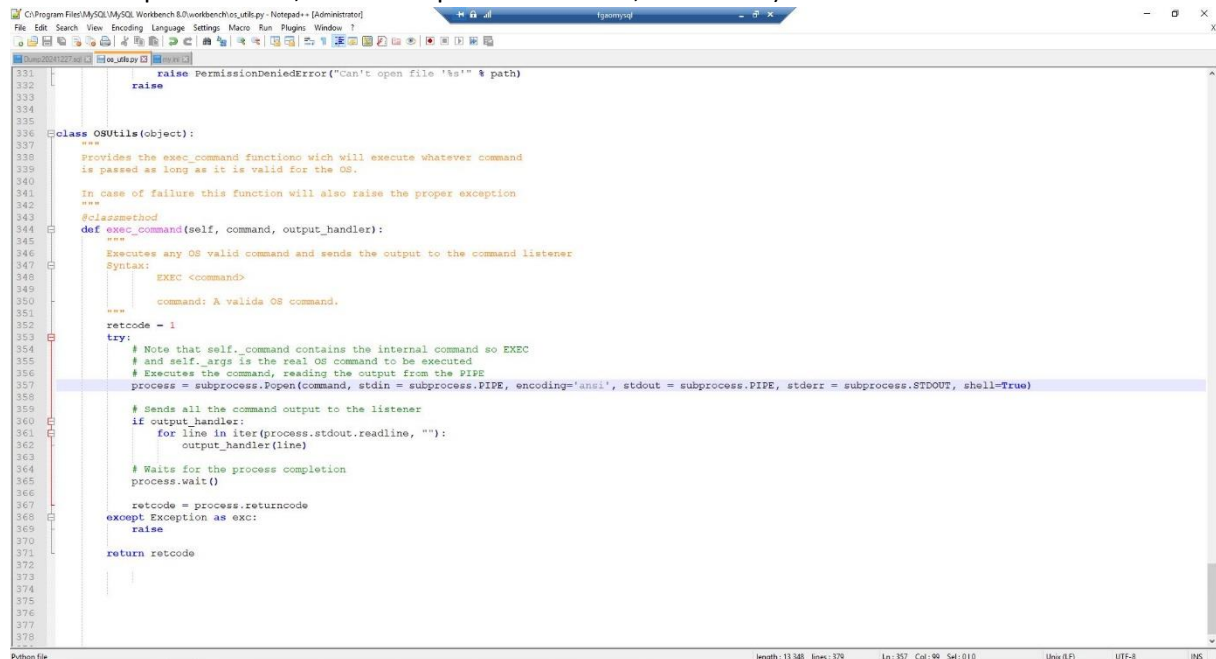
Ouvrez le fichier suivant avec un éditeur de texte comme Notepad++ :

C:\Program Files\MySQL\MySQL Workbench 8.0\workbench\os_utils.py.

À la ligne 357, remplacez le texte « UTF-8 » par « ansi ».

Exemple :

```
process = subprocess.Popen(command, stdin=subprocess.PIPE, encoding='ansi',
stdout=subprocess.PIPE, stderr=subprocess.STDOUT, shell=True)
```



The screenshot shows a Notepad++ window editing the file 'os_utils.py'. The code is as follows:

```
331         raise PermissionDeniedError("can't open file '%s'" % path)
332     raise
333
334
335
336 class OSUtils(object):
337     """
338     Provides the exec_command functions which will execute whatever command
339     is passed as long as it is valid for the OS.
340
341     In case of failure this function will also raise the proper exception
342     """
343     @classmethod
344     def exec_command(self, command, output_handler):
345         """
346         Executes any OS valid command and sends the output to the command listener.
347         Syntax: EXEC <command>
348
349         command: A valid OS command.
350
351         """
352         retcode = 1
353         try:
354             # Note that self._command contains the internal command so EXEC
355             # and self._args is the real OS command to be executed
356             # Executes the command, reading the output from the PIPE
357             process = subprocess.Popen(command, stdin = subprocess.PIPE, encoding='ansi', stdout = subprocess.PIPE, stderr = subprocess.STDOUT, shell=True)
358
359             # Sends all the command output to the listener
360             if output_handler:
361                 for line in iter(process.stdout.readline, ""):
362                     output_handler(line)
363
364             # Waits for the process completion
365             process.wait()
366
367             retcode = process.returncode
368         except Exception as exc:
369             raise
370
371         return retcode
372
373
374
375
376
377
378
```

Puis sauvegardez les modifications.

L'encodage ANSI est un standard relativement ancien qui *utilise un octet (8 bits) pour représenter un caractère*. Cela limite le nombre de caractères pouvant être représentés à 256.

Chaque région géographique a sa propre table ANSI, ce qui peut entraîner des problèmes d'incompatibilité lorsque l'on échange des données entre différents systèmes. Par exemple, l'ANSI français ne contient pas les mêmes caractères que l'ANSI russe.

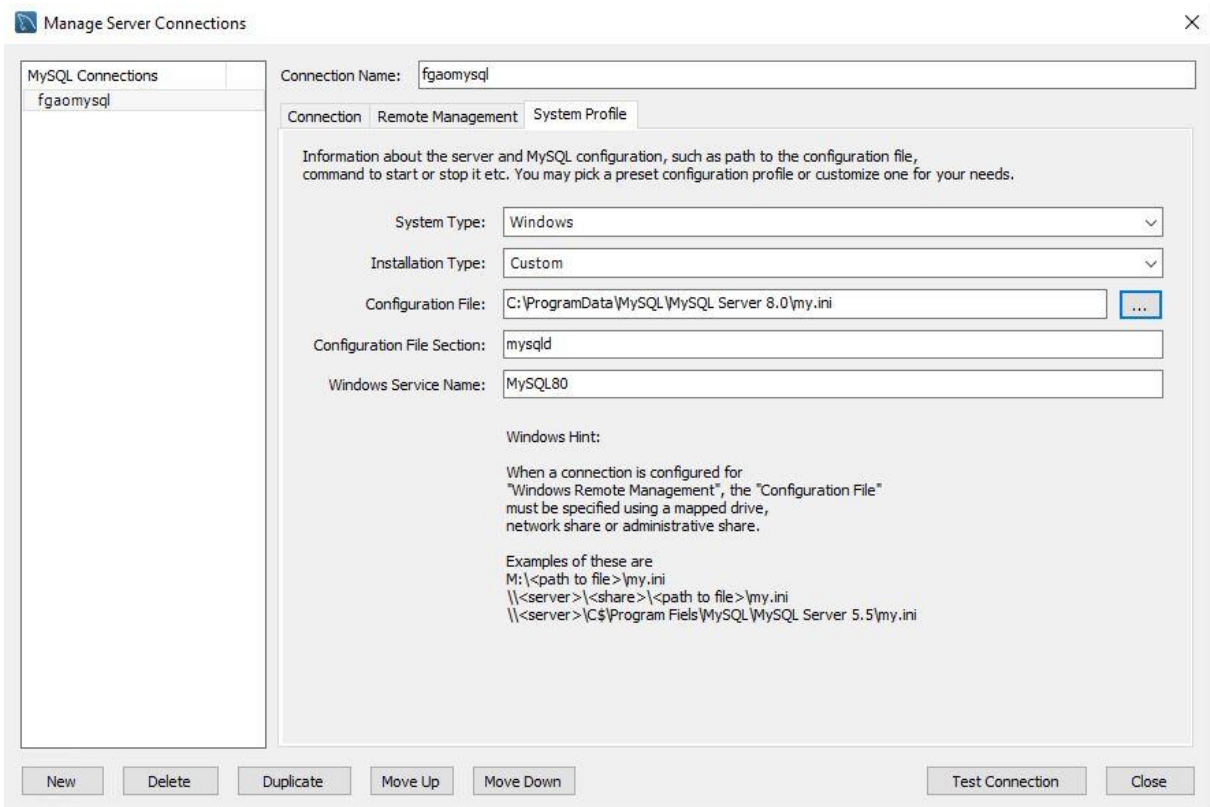
L'UTF-8 est un standard plus moderne et flexible qui *utilise un nombre variable d'octets pour représenter un caractère*. Cela permet de représenter un nombre beaucoup plus grand de caractères, incluant les caractères spéciaux, les symboles mathématiques, et les caractères de nombreuses langues du monde.

L'UTF-8 est une implémentation de l'Unicode, un standard international qui définit un répertoire universel de caractères.

L'UTF-8 est rétro compatible avec l'ASCII, ce qui signifie que les caractères ASCII sont représentés de la même manière en UTF-8 et en ANSI.

Nous avons choisi de remplacer l'encodage « UTF-8 » par « ansi » à la ligne 357 du fichier os_utils.py pour résoudre le problème. Bien que l'UTF-8 soit généralement recommandé pour sa compatibilité avec une large gamme de caractères, dans ce cas précis, l'encodage ANSI s'est avéré plus adapté. Il est important de noter que ce changement peut entraîner une perte de données si votre jeu de données contient des caractères spéciaux non pris en charge par l'encodage ANSI.

3/ Configurer la connexion dans MySQL Workbench



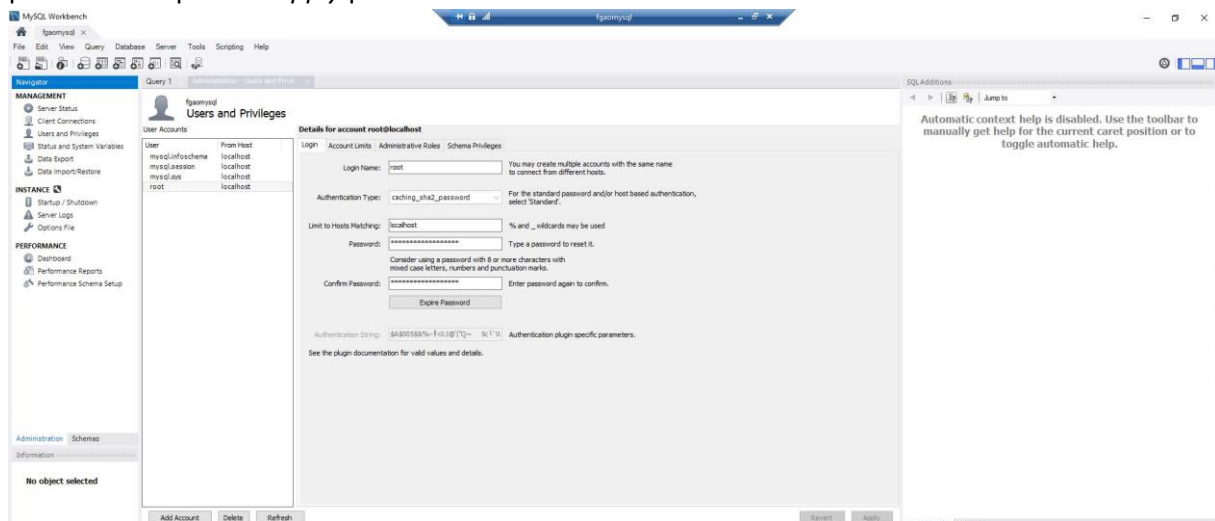
Dans MySQL Workbench, accédez à Database puis cliquez sur Manage Connections.

Dans la fenêtre *Manage Server Connections*, sélectionnez votre connexion dans la colonne de gauche, puis cliquez sur System Profile. Complétez les informations nécessaires comme l'indique l'image.

Cliquez sur *Test Connection* et saisissez le mot de passe si demandé.

Pour modifier votre mot de passe root :

Accédez au menu *User and Privileges*. Dans la colonne de gauche, sélectionnez l'utilisateur *root*, puis dans la section centrale, repérez le champ *Password*. Saisissez le nouveau mot de passe souhaité dans le champ password et confirmez le mot de passe en le ressaisissant dans le champ confirm password. Cliquez sur *Apply* pour valider les modifications.



4/ Redémarrez MySQL80

Fermez MySQL Workbench et redémarrez le service MySQL80 pour appliquer les changements.

III. Désactivation de certaines options dans MySQL

Le « *Performance Schema* » est une base de données interne contenant des tables fournissant des informations sur les performances et le fonctionnement du serveur MySQL, et non sur les données de la base elle-même. Bien qu'il soit utile pour surveiller les performances, il consomme des ressources, ce qui peut entraîner une saturation si les ressources système sont limitées. Désactiver cette option libère donc des ressources pour d'autres tâches.

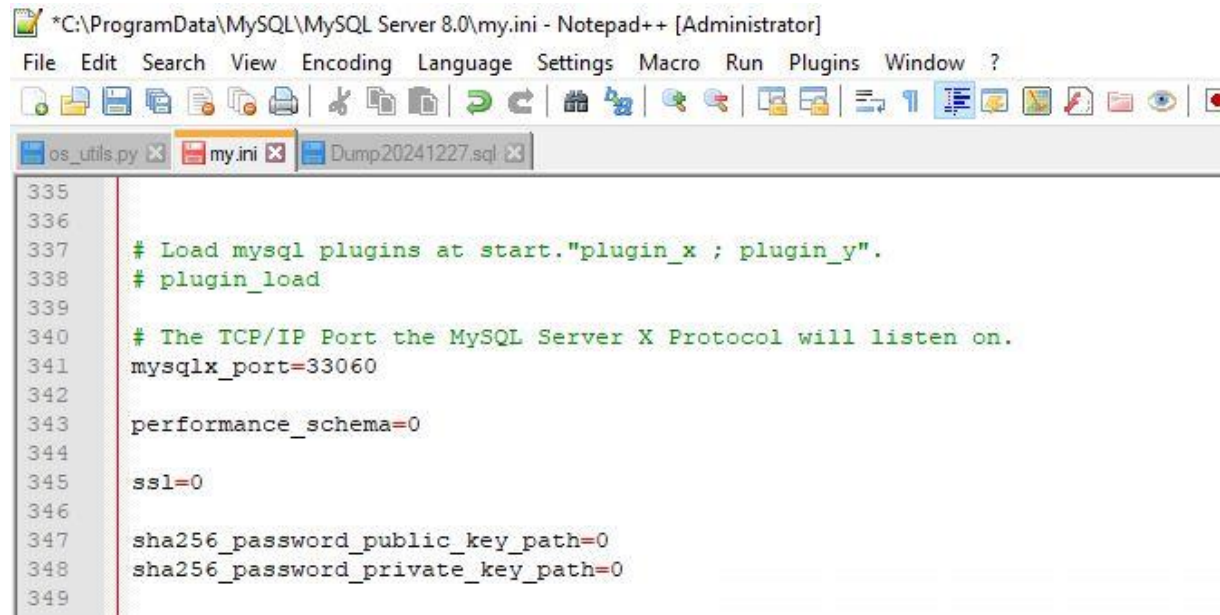
Pour le « *SSL Availability* » et le « *SHA256 Password Private/Public Key* », étant donné que les serveurs utilisant cette base sont internes, toute cette sécurité n'est pas nécessaire. Cela ajoute une complexité inutile pour les échanges internes. En revanche, pour les données qui transitent sur internet, il est important d'activer ces options pour garantir leur sécurité.

1/ Modifier le fichier « *my.ini* »

Ouvrez le fichier suivant avec Notepad++ : C:\ProgramData\MySQL\MySQL Server 8.0\my.ini.

Pour désactiver l'option *Performance Schema*, ajoutez cette ligne dans la section [mysqld] :
performance_schema=0 (de préférence tout à la fin)

Pour désactiver d'autres options comme *SSL Availability* ou *SHA256 Password Keys*, ajoutez les lignes correspondantes dans le fichier :



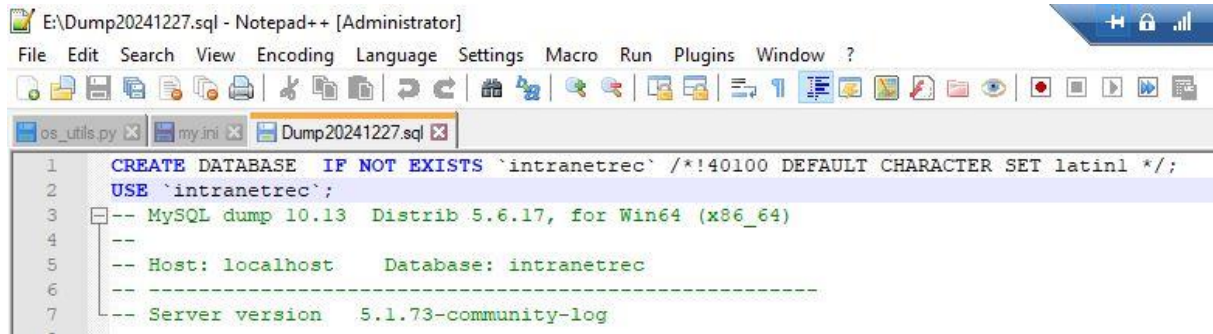
```
*C:\ProgramData\MySQL\MySQL Server 8.0\my.ini - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
os_utils.py x my.ini x Dump20241227.sql x
335
336
337 # Load mysql plugins at start."plugin_x ; plugin_y".
338 # plugin_load
339
340 # The TCP/IP Port the MySQL Server X Protocol will listen on.
341 mysqlx_port=33060
342
343 performance_schema=0
344
345 ssl=0
346
347 sha256_password_public_key_path=0
348 sha256_password_private_key_path=0
349
```

Puis sauvegardez les modifications dans my.ini.

2/ Redémarrez le service MySQL80 pour prendre en compte les modifications.

IV. Importer un Dump

1/ Ouvrez le fichier dump avec Notepad++ pour vérifier le nom du schéma attendu (souvent mentionné dans les premières lignes du fichier). Exemple : USE `intranetrec` ;



```
1 CREATE DATABASE IF NOT EXISTS `intranetrec` /*!40100 DEFAULT CHARACTER SET latin1 */;  
2 USE `intranetrec`;  
3 -- MySQL dump 10.13 Distrib 5.6.17, for Win64 (x86_64)  
4 --  
5 -- Host: localhost Database: intranetrec  
6 -----  
7 -- Server version 5.1.73-community-log
```

Pour que l'importation fonctionne correctement, le *nom du schema doit correspondre exactement à celui spécifié dans le fichier dump* (fichier contenant une sauvegarde d'une base de données, incluant sa structure et ses données). Si les noms ne correspondent pas, l'importation échouera.

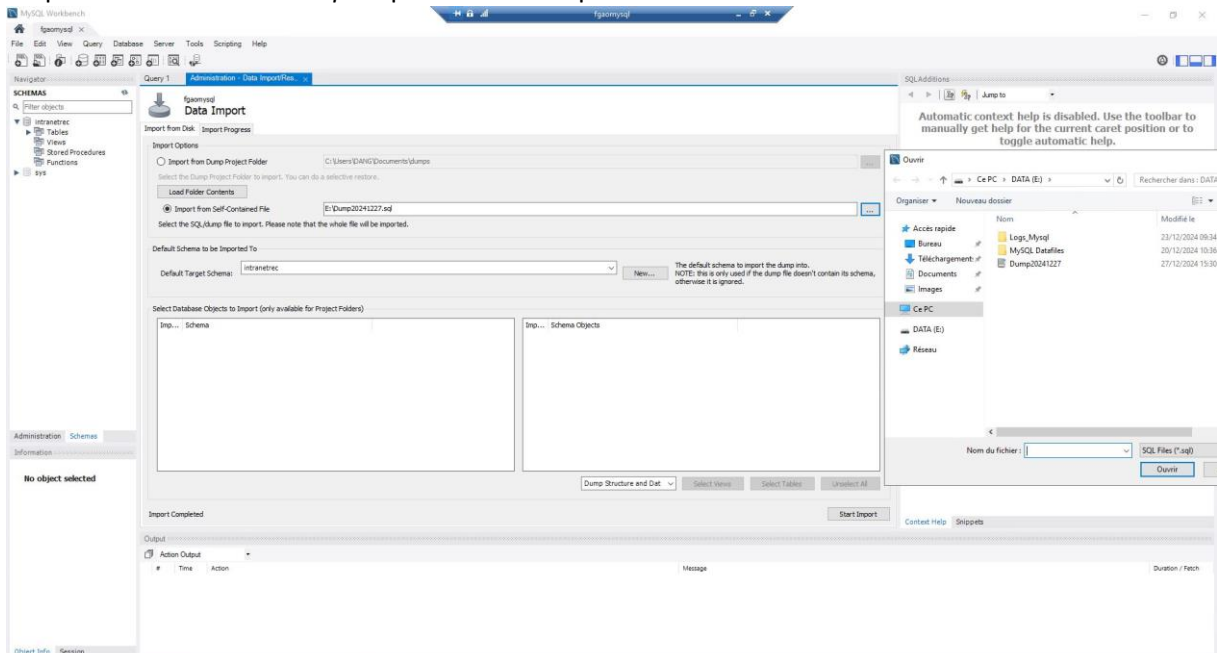
2/ Importer le Dump dans MySQL Workbench

Dans MySQL Workbench, accédez à *Server* puis *Data Import*.

Une fois sur la page *Data Import*, sélectionnez *Import from Self-Contained File* puis indiquez le *chemin vers le fichier dump* (un fichier dump fourni par l'administrateur, a été utilisé afin de restaurer une base de données existante dans MySQL).

Dans la section *Default Target Schema*, choisissez un schéma existant ou créez-en un nouveau en cliquant sur *New...* (rappelons que le nom du schema doit être identique à celui dans le fichier dump). Ici, le schéma existant *intranetrec* a été sélectionné comme cible, sans création d'un nouveau schéma.

Cliquez ensuite sur *Start Import* pour lancer l'importation.



Une fois terminé, faites un clic droit sur la colonne gauche puis sur *Refresh All* pour visualiser les changements.

Conclusion

Cette activité m'a permis de participer à une opération de migration de base de données, en assurant l'installation, la configuration et la restauration des données. Elle a contribué à la mise en place d'un environnement plus récent, améliorant la compatibilité, les performances et la maintenabilité du système.

Elle m'a également permis de développer mes compétences en administration de bases de données, notamment dans la gestion des configurations, l'analyse d'erreurs et l'importation de données.