

Projet 1SIO Janv 2024

Table des matières

Cahier des charges.....	2
Documents à produire.....	3
Documents mis à disposition.....	3
Matériel à disposition.....	3
Évaluation.....	3
Les itérations du projet (étapes).....	3
Étape 00 - Constitution des équipes.....	3
Étape 01 - Préparation des VM.....	3
Étape 02 - Lecture et écriture dans un fichier CSV en Python.....	4
Étape 03 - Création d'une base de données et d'un utilisateur ayant des privilèges dessus en SQL.....	5
Étape 04 : Extraction des utilisateurs de l'annuaire Active Directory.....	5
Étape 05a : Création de mots de passe robustes (phase 1 - tableur).....	10
Étape 05b : Création de mots de passe robustes (phase 2 – script python).....	11
Étape 05c : Validation respect des contraintes d'une liste de mots de passe.....	11
Étape 06 : Programme Python de génération du script SQL (CREATE ...).....	12
Étape 07 : Test du script SQL généré (CREATE ...).....	12
Étape 08 : Programme Python de génération du script SQL (DROP ...).....	13
Étape 09 : Test du script SQL généré (DROP ...).....	13
Étape 10 : Rédaction d'un mode opératoire.....	13
Étape 11 : Evolution 1 - Suppression des caractères spéciaux dans les noms d'utilisateur.....	13
Étape 12 : Evolution 2 - Paramètres en ligne de commandes.....	14
Étape 13 : Evolution 3 - Mise en œuvre de "pattern" pour créer ou supprimer les utilisateurs.....	14
Étape 14 : Evolution 4 - Exécution automatique du script SQL.....	15
Étape 15 : Evolution 5 - Requêtes SQL exécutées par python.....	15
Étape 16 : Evolution 6 - Mise en œuvre d'une interface graphique.....	15
ANNEXE - Récapitulatif des programmes et scripts demandés.....	17

Cahier des charges

La société STESIO dispose d'un contrôleur de domaine CDSTESIO et d'un serveur de base de données WEB-BD-STESIO.

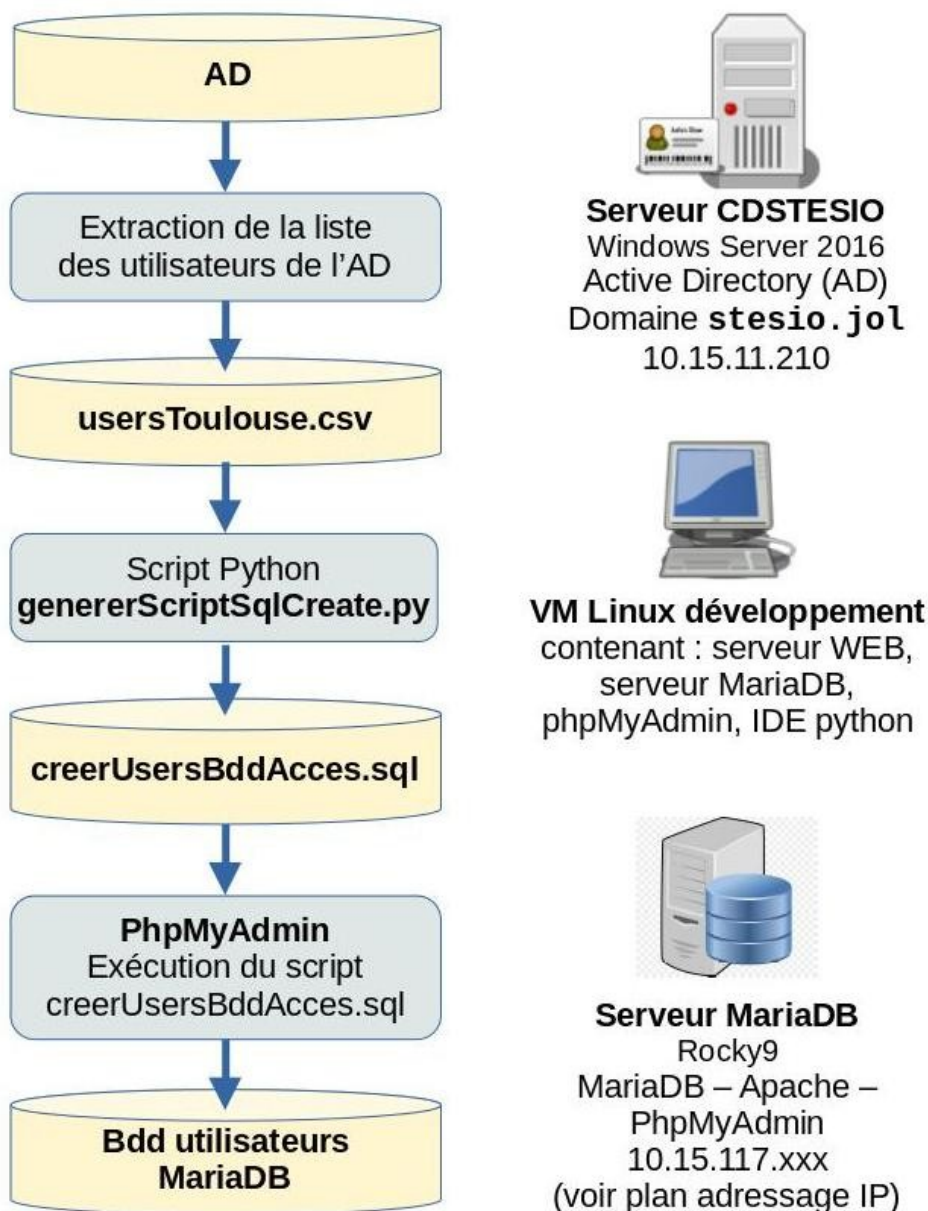
Le client souhaite que chaque utilisateur d'un de ses sites ayant un compte dans la société STESIO dispose de sa propre base de données sur le serveur de base de données avec son propre accès.

Il est possible de **recupérer la liste des utilisateurs** en interrogeant l'annuaire Active Directory (AD) du contrôleur de domaine et de la copier dans un fichier CSV (fichier texte avec séparation des informations).

Un script python générera à partir de ce fichier CSV les ordres SQL de création des comptes utilisateurs avec leur mot de passe, leur base de données et leur accès.

Le script SQL ainsi créé sera enfin exécuté dans le SGBD MariaDB pour créer les utilisateurs, les bases de données et leur accès.

Ainsi, chaque utilisateur pourra se connecter à distance à sa propre base de données (seulement la sienne) sur le serveur de base de données de la société.



Documents à produire

A chaque étape, vous devez remettre des documents dans un fichier d'archive ZIP en respectant la convention de nommage suivante : **pyt24_bBB_eEE_nom1_nom2.zip**

BB : n° binôme (01 à ...), **EE** : n° étape (01 à 16), **nom1** et **nom2** : noms étudiants du binôme

Exemple : pyt24_b07_e01_morane_lefort.zip

Un travail a été ouvert à cet effet sur la plateforme Moodle.

Documents mis à disposition

- Le présent cahier des charges
- Une documentation Python sur la gestion de fichiers et le traitement des chaînes de caractères

Matériel à disposition

Chaque étudiant dispose de plusieurs VM :

- **1 serveur MariaDB** : **VOTRELOGIN_T_rocky9_mariadb_apache_phpmyadmin**
située dans l'arborescence Datacenter - 1SIO - 1SIOx - Projet_python_x dans la baie VMWare.
Compte utilisateur : **btssio / btssio** compte administrateur : **root / joliverie**
- **1 VM Windows** (utile pour extraire des objets Active Directory)
- **1 VM Fedora** (utile pour développer en python)

Évaluation

Le vendredi, chaque binôme évaluera la réalisation de deux autres binômes : la grille d'évaluation est remplie en testant le jeu d'essai et le mode opératoire fournis par le binôme évalué.

Les itérations du projet (étapes)

Étape 00 - Constitution des équipes

Vous devez constituer un binôme et le déclarer aux enseignants.

Étape 01 - Préparation des VM

Un 1er objectif est de mettre en service votre serveur MariaDB avec le bon nom de machine, les bons paramètres IP et le bon VLAN.

Un 2ème objectif est de disposer d'un environnement de développement (IDE) Python, **SPYDER** par exemple. Ce dernier facilitera le développement de vos futurs scripts, notamment à l'aide de la **complétion automatique** (en tapant les premières lettres d'une instruction, variables... l'IDE vous propose des choix possibles) et le **débogueur** (utile à la mise au point du script).

Préparation du serveur MariaDB

Pour toutes les opérations suivantes, consultez le plan d'adressage IP fourni dans Moodle.

- Sélectionnez le VLAN qui vous a été attribué
- Renommez votre serveur en respectant le format suivant :
SRV-SQL-PNNNN P : première lettre de votre prénom NNNN : 4 premières lettres de votre nom
- Configurer les paramètres IP de votre serveur en respectant le le plan d'adressage IP.

- Authentifiez-vous auprès du Stormshield pour accéder à Internet (<https://10.15.254.254>). Testez l'accès à Internet
- Accédez à votre espace réseau personnel :
`smb://srvnas.jolsio.net/homes2025$/votrelogin/`
- Testez l'accès en transférant le fichier de votre choix de votre espace réseau personnel à votre VM.

Préparation de la station de travail de développement

- Assurez-vous que la VM que vous allez utiliser ait bien été renommée
Ce nom doit être personnalisé. Par exemple : FED-PNNNN-AP ou WIN10-PNNNN-AP
- Installez, si cela n'a pas été fait, un IDE (environnement de développement), Spyder par exemple. Testez votre IDE.

Voir doc fourni dans la partie Langage Python Découverte (voir module AP – Moodle) :

- docpy01_spyder_et_environn_virtuel_Linux_v1b.pdf
- environnement_python_spyder.pdf
- dossier Scripts python pour test environnement spyder

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape
- Une capture d'écran montrant le nom de machine du serveur et ses paramètres IP.
- Une capture d'écran montrant l'environnement python utilisé en précisant le système d'exploitation utilisé (Windows ou Linux)

Étape 02 - Lecture et écriture dans un fichier CSV en Python

Cette étape a pour but de vous former aux commandes Python permettant de lire et d'écrire dans un fichier CSV (Cf. doc. Python).

- Enregistrez les lignes suivantes dans un fichier **testEntree.csv** :

```
Prenom;NOM  
Emile;ZOLA  
Victor;HUGO  
George;SAND
```
- Réalisez ensuite un programme effectuant les actions suivantes :
 - ouvrir le fichier **testEntree.csv** en lecture et un fichier **testSortie.csv** en écriture.
 - pour chaque ligne du fichier **testEntree.csv** :
 - créer un login
 - extraire la première lettre du prénom et le nom puis les concaténer
 - convertir la chaîne obtenue en minuscules
 - écrire cette chaîne dans le fichier **testSortie.csv**. La première ligne du fichier contiendra le titre de la colonne comme tous les fichiers CSV, à savoir **login**.

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape et une capture d'écran du contenu des fichiers CSV et python
- Le programme python commenté **gererFichierCsv.py**
- Les fichiers **testEntree.csv** et **testSortie.csv**

Étape 03 - Création d'une base de données et d'un utilisateur ayant des privilèges dessus en SQL

- Recherchez les ordres SQL pour :
 - **Créer une base de données** ayant pour nom votre login
 - **Créer un utilisateur** MariaDB portant le nom de votre login avec pour mot de passe **P@ssw0rd**
 - **Donner les privilèges nécessaires** à cet utilisateur pour qu'il puisse effectuer les opérations suivantes sur sa base de données :
 - Créer, modifier et supprimer des tables et des colonnes
 - Insérer, modifier et supprimer des données
 - Interroger **sa base de données**
- Testez ces commandes en utilisant **phpmyadmin** ou en **ligne de commandes** à partir d'un terminal et vérifiez les points suivants :
 - l'utilisateur peut :
 - se connecter à sa base de données, mais pas à une autre
 - créer une table, créer des colonnes dans cette table puis ajouter des enregistrements
 - récupérer des informations de cette table avec une requête
 - modifier et supprimer des enregistrements
 - modifier, ajouter et supprimer des colonnes
 - supprimer une table

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, l'explication des ordres SQL, les tests de fonctionnement avec les captures d'écran associées et commentées.

Étape 04 : Extraction des utilisateurs de l'annuaire Active Directory

Vous allez maintenant extraire certaines informations sur certains utilisateurs (noms et prénoms) de l'annuaire d'un serveur de domaine Windows. Ces informations nous permettront de créer les comptes utilisateurs et les bases de données dans MariaDB.

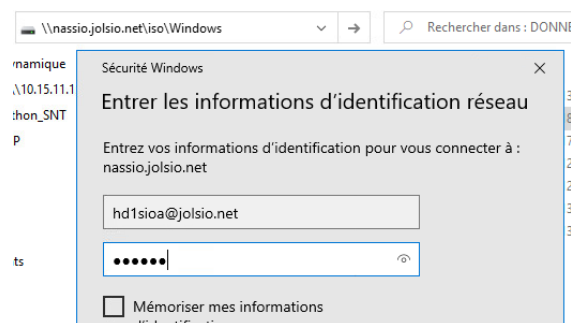
Marche à suivre

Pour se connecter à l'annuaire Active Directory du serveur **CDSTESIO**, vous utiliserez le logiciel "LDAPBrowser" de l'éditeur "Softerra", disponible sur :

`\\nassio.jolsio.net\iso\Windows\`

- Lancez votre VM Windows, téléchargez le fichier d'installation `LDAPbrowser-xxx.msi` sur `nassio`

Pour accéder au `nassio`, il faudra s'authentifier avec `votrelogin@jolsio.net / votremotdepasse`.



- Lancez l'installation par défaut de `LDAPbrowser-xxx.msi`

- Lancez l'application LDAPbrowser.

Définissez-la comme client LDAP par défaut (mémoriser) si on vous le demande

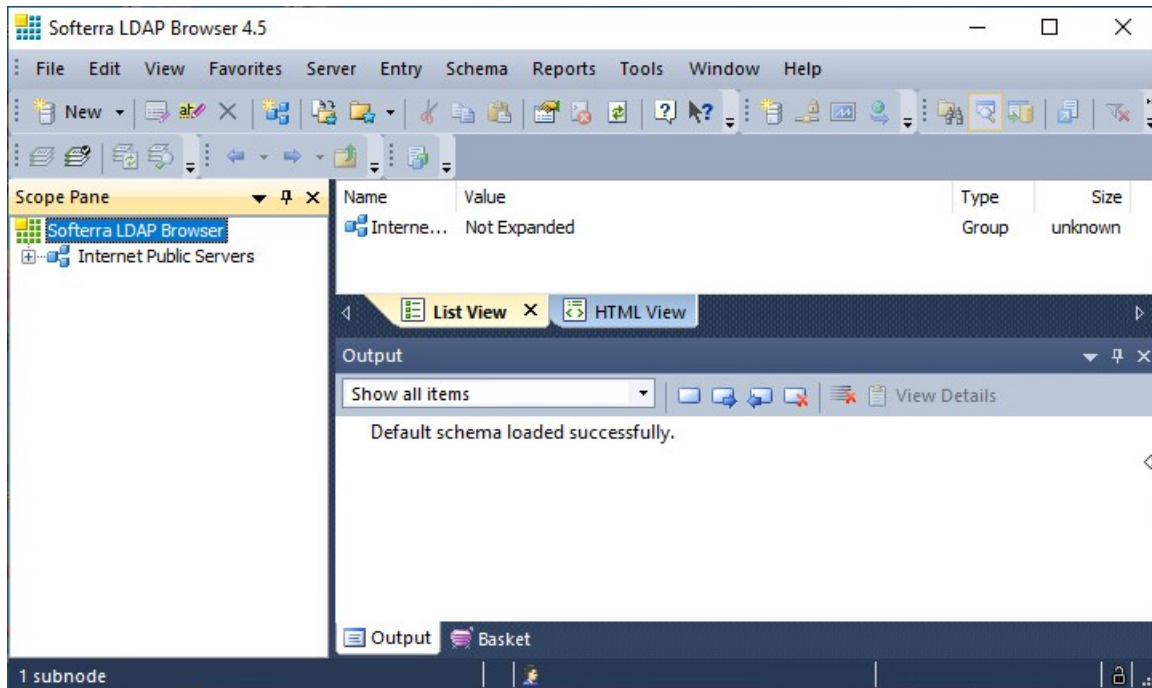
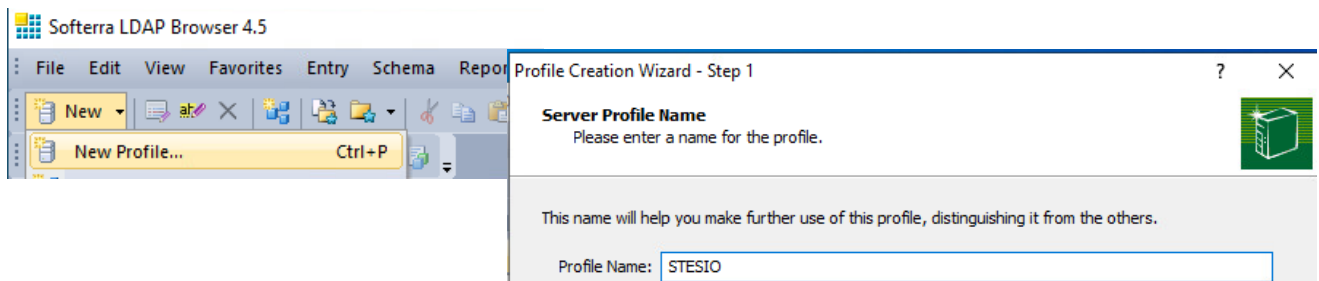


Figure 1: L'application LDAPbrowser a démarré. Il s'agit de la même application que LDAPAdministrator sans les droits de modification (lecture seule).

- dans LDAPBrowser, créer un nouveau profil **STESIO** pour vous connecter à l'Active Directory du serveur **CDSTESIO**

1. Cliquer **New – New Profile**



2. Complétez les informations du serveur **cdstesio.stesio.jol**

En cliquant sur **Lookup Servers...** puis en renseignant le domaine **stesio.jol**, on retrouve le contrôleur de domaine **CDSTESIO**. Cela permet de remplir automatiquement la zone **Host**

Profile Creation Wizard - Step 2

Profile General Information
Please provide general information.

Please specify server host information and adjust general security options.

Host Information

Host: Port:

Base DN:

Security Options

☐ Use secure connection (SSL)

Specify an LDAP URL for the other fields to be filled based on it.

LDAP URL:

< Précédent **Suivant >** Terminer Annuler Aide

Lookup LDAP Servers

Lookup in domain:

Available servers:

Figure 2: Base DN : CN=btssio,CN=Users,DC=stesio,DC=jol

3. Ajoutez l'utilisateur / mot de passe qui a le droit d'interroger l'annuaire Active Directory.

L'utilisateur est **btssio@stesio.jol** / **P@ssw0rd**

soit en requête LDAP : **CN=btssio,CN=Users,DC=stesio,DC=jol**

Profile Creation Wizard - Step 3

User Authentication Information
Bind using one of the following authentication options.

☐ Anonymous user
☐ Currently logged on user (Active Directory only)
☐ External (SSL Certificate)
☒ Other credentials

Mechanism:

Principal:
Example: cn=User,ou=People,o=Company

Password:
☒ Save password

☒ Try matching the credentials required for referral rebind.

< Précédent **Suivant >** Terminer

Profile Creation Wizard - Step 4

LDAP Settings
Please adjust LDAP settings for the profile.

Connection Options

Filter: Timeout: Size limit:

Referrals Handling

☐ None - Disable handling referrals.
☐ Opaque - All the referral's subentries are displayed in place of a referral entry.
☒ Manual - Referrals are displayed as an entry.

Dereference Aliases

☐ Searching - Dereference aliases in subordinates of the base object being searched.
☐ Finding - Dereference aliases in locating the base object of the search.

< Précédent **Suivant >** **Terminer** Annuler Aide

4. Cliquez **Terminer**, la création du profil doit réussir

Le profil **STESIO** est maintenant créé et vous pouvez accéder à l'Active Directory du serveur CDSTESIO.

The screenshot shows the Softerra LDAP Browser 4.5 interface. The left pane displays the directory tree for 'STESIO' under 'Internet Public Servers'. The right pane shows a list of attributes and their values for the selected entry.

Scope Pane:

- Internet Public Servers
 - STESIO
 - CN=Configuration
 - CN=Schema
 - DC=DomainDnsZones
 - DC=ForestDnsZones
 - DC=stesio
 - CN=Builtin
 - CN=Computers
 - OU=Domain Controllers
 - OU=Finances
 - OU=Calais
 - OU=Dijon
 - OU=Limoges
 - OU=Madrid
 - OU=Marly-Gomont
 - OU=Marseille
 - OU=Nice
 - OU=Paris
 - OU=Perpignan
 - OU=Toulouse
 - CN=ForeignSecurityPrincipals
 - CN=Infrastructure
 - CN=Keys

Find what: Search in: Names, Descriptions

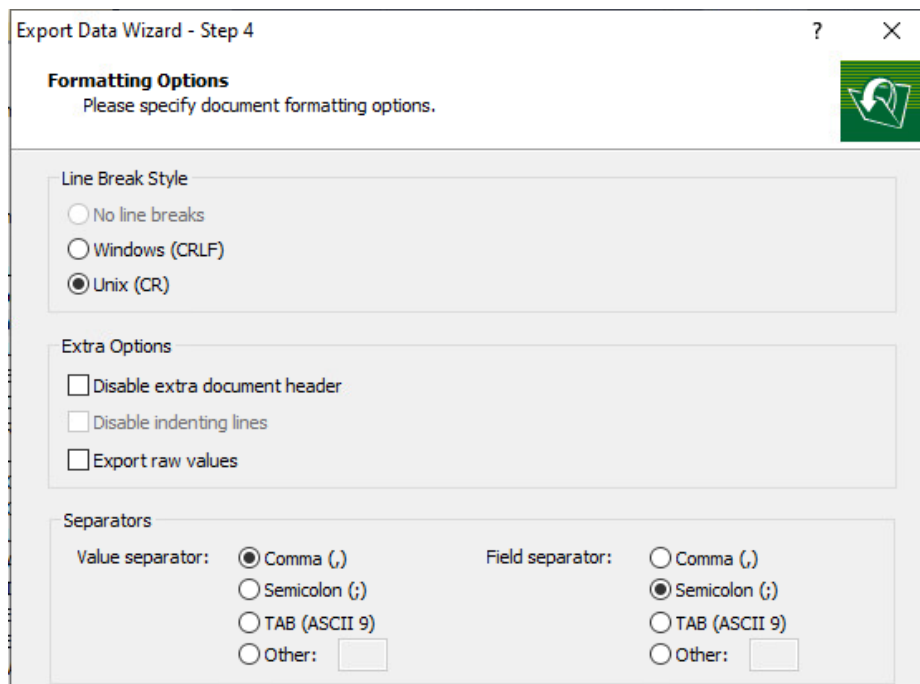
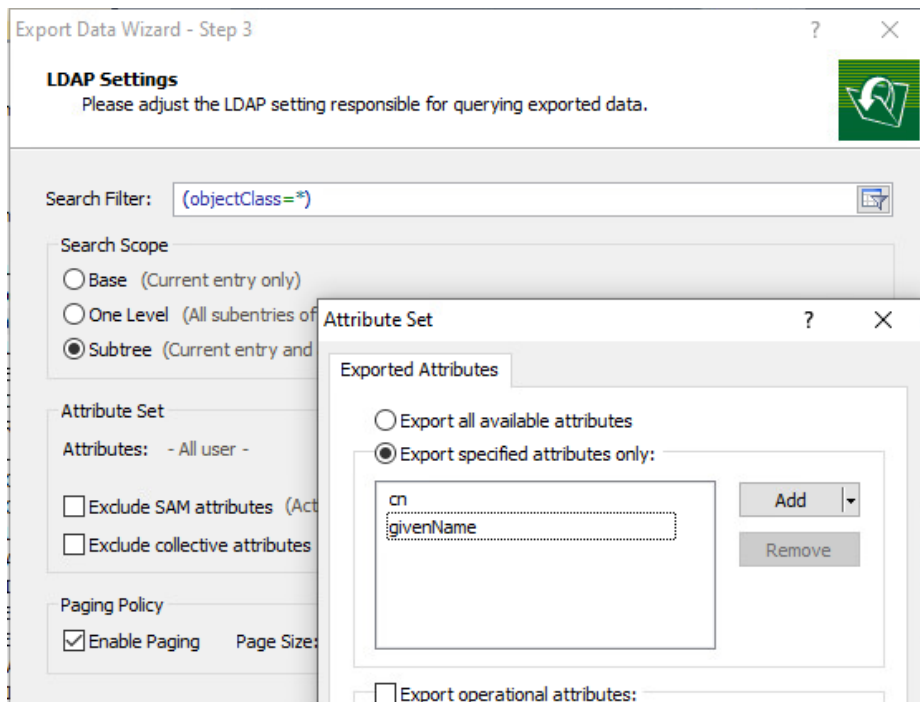
Name	Value
CN	Configuration
CN	Schema
DC	DomainDnsZones
DC	ForestDnsZones
DC	stesio
currentTime	27/12/2023 21:35:43
dsServiceName	CN=NTDS Settings,CN=CDSTESIO,CN=Servers,CN=Default-First-Site-Name...
defaultName	DC=stesio,DC=jol
schemaName	CN=Schema,CN=Configuration,DC=stesio,DC=jol
configurationName	CN=Configuration,DC=stesio,DC=jol
rootDomainName	DC=stesio,DC=jol
supportedLDAPVersion	MaxPoolThreads
supportedLDAPVersion	MaxPercentDirSyncRequests
supportedLDAPVersion	MaxDatagramRecv
supportedLDAPVersion	MaxReceiveBuffer

Output:

Show all items

Default schema loaded successfully.
Schema for cdstesio.stesio.jol:389 loaded successfully.

- Extrayez les utilisateurs de l'unité d'organisation (OU) **Toulouse** dans un fichier CSV comme suit :
 - Développez le nœud **STESIO**, puis l'OU **Finances** et l'OU **Toulouse**
 - Sélectionnez le nœud **Utilisateurs**,
 - cliquez droit sur **Export Data** au format **Plain text**
 - renommez le fichier en **usersToulouse.csv**
 - dans **Attribute Set (Edit)**, précisez les attributs qui vous intéressent (**cn** et **givenName**)
 - Précisez comme fin de ligne le format Unix. comme séparateur de valeur la "," (comma) et comme séparateur de champ le ";" (semicolon).



Vous devez obtenir un fichier comme celui ci-dessous

```

1 |DN;cn;givenName
2 |"OU=Utilisateurs,OU=Toulouse,OU=Finances,DC=stesio,DC=jol";
3 |"CN=RUIZ Charlotte,OU=Utilisateurs,OU=Toulouse,OU=Finances,DC=stesio,DC=jol";"RUIZ Charlotte";"Charlotte"
4 |"CN=SAMSON Noemie,OU=Utilisateurs,OU=Toulouse,OU=Finances,DC=stesio,DC=jol";"SAMSON Noemie";"Noemie"
5 |"CN=BONHOMME Myrtille,OU=Utilisateurs,OU=Toulouse,OU=Finances,DC=stesio,DC=jol";"BONHOMME Myrtille";"Myrtille"
6 |"CN=LEFEBVRE Arthur,OU=Utilisateurs,OU=Toulouse,OU=Finances,DC=stesio,DC=jol";"LEFEBVRE Arthur";"Arthur"

```

- Transférez le fichier créé (**usersToulouse.csv**) de votre VM Win10 vers votre VM de développement en passant par votre espace réseau personnel
 - Créez un lecteur réseau dans Win10
 - Explorateur de fichiers - clic droit sur Réseau - Connecter un lecteur réseau
 - Choisir une lettre et saisir le chemin UNC de votre espace réseau personnel
\\srvnas.jolsio.net\homes2025\$\votrelogin\
 - Faites la même chose dans votre VM de développement (si Linux : smb://srvnas.jolsio.net\homes2025\$\votrelogin\).
 - Vous pouvez maintenant vous échanger des fichiers entre vos 2 VM

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape et une capture d'écran du contenu du fichier CSV.
- Le fichier **usersToulouse.csv**.

Étape 05a : Création de mots de passe robustes (phase 1 - tableur)

La société STESIO a été récemment victime de plusieurs cyberattaques, sans conséquence pour l'instant. Cependant, le service informatique veut renforcer sa politique de mot de passe. Dans un premier temps, chaque mot passe devra respecter les critères de sécurité suivants :

- longueur minimum : 10 caractères
- doit comporter au moins 1 majuscule, 1 minuscule et 1 chiffre
- ne doit pas comporter les caractères suivants : espace, lettre O majuscule et chiffre 0 (pour éviter confusion entre O et 0)

Dans un 2ème temps, d'autres contraintes seront ajoutées (étape 05b).

Pour créer les mots de passe, le service informatique préconise d'utiliser un tableur qui à l'aide de formules générera un mot de passe répondant aux critères de sécurité pour chaque utilisateur.

- Créez un fichier nommé **usersPassword** à l'aide d'un tableur (Calc, Excel) permettant de créer ces mots de passe. Vous devez avoir à votre disposition autant de mots de passe qu'il y a d'utilisateurs dans le fichier **usersToulouse.csv**.

Pour simplifier la réalisation, le mot de passe sera constitué de la manière suivante :

C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 avec

C1 : lettre minuscule	C2 : lettre majuscule	C3 : chiffre	C4 : lettre minuscule
C5 : lettre minuscule	C6 : chiffre	C7 : chiffre	C8 : lettre majuscule
C9 : lettre minuscule	C10 : lettre majuscule		

- Générez à partir de votre fichier tableur un fichier CSV nommé **usersPassword.csv** avec comme titre de colonne **password**. Ce fichier ne doit comporter que les mots de passe, un par ligne, plus le titre de colonne en début de fichier.

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, expliquant la méthode utilisée pour générer les mots de passe demandés (indiquer les formules utilisées) et la capture d'écran du contenu du fichier CSV obtenu
- fichier tableur **usersPassword.ods** (si Calc) ou **usersPassword.xlsx** (si Excel)
- fichier **usersPassword.csv**

Étape 05b : Création de mots de passe robustes (phase 2 – script python)

Finalement, le service informatique décide de durcir sa dernière politique de mot de passe.

Voici les contraintes imposées :

- Longueur minimale : 12 caractères.
 - Le mot de passe doit contenir au moins 2 majuscules, 2 minuscules, 2 chiffres et 2 caractères spéciaux.
 - Toutes les lettres majuscules sont autorisées, à l'exception de 'O' pour éviter toute confusion avec le chiffre '0'.
 - Tous les chiffres sont autorisés, à l'exception de '0' pour éviter toute confusion avec la lettre 'O'.
 - Les caractères spéciaux autorisés sont : ! @ # \$ % & * + - = . < > / ? :
 - L'utilisation d'espaces est interdite.
 - Le mot de passe ne doit pas contenir de mots issus d'un dictionnaire (français, anglais, etc.), même s'ils sont modifiés en remplaçant certaines lettres par des caractères spéciaux.
 - Les différents types de caractères ne doivent pas avoir de positions spécifiques dans le mot de passe (par exemple, tous les mots de passe ne doivent pas se terminer par un chiffre). Cependant, les mots de passe ne doivent pas commencer ni se terminer par un caractère spécial.
 - Chaque utilisateur doit avoir son propre mot de passe, qui doit être différent des autres utilisateurs.
- Créez un programme Python nommé **"createPassword.py"** qui génère un fichier CSV, **"usersPassword2.csv"**, contenant un nombre spécifié de mots de passe fournis en tant qu'argument. Le fichier CSV doit avoir une première ligne indiquant le nom de la colonne (**"password"**) et contenir autant de lignes que de mots de passe générés. Chaque mot de passe doit être conforme aux contraintes préalablement spécifiées.

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape et une capture d'écran du contenu des fichiers python et CSV
- Le programme python commenté **createPassword.py**
- Le fichier **usersPassword2.csv**

Étape 05c : Validation respect des contraintes d'une liste de mots de passe

Le service informatique exprime le besoin d'un programme capable de vérifier la validité d'une liste de mots de passe.

- Élaborez un programme Python nommé **"testPassword.py"** qui examine les contraintes des mots de passe fournis dans un fichier CSV spécifié en argument. Les contraintes prises en compte sont celles définies à l'étape 05b, à l'exception de la vérification de l'utilisation de mots issus d'un dictionnaire. Le programme doit indiquer si le mot de passe est valide, sinon il doit signaler la nature du problème rencontré. N'oubliez pas de vérifier qu'il n'y a pas de doublons dans la liste à valider.

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape et une capture d'écran du contenu des fichiers python et CSV
- Une capture d'écran du test du programme python commentée
- Le programme python commenté **testPassword.py**
- Le fichier d'entrée CSV utilisé pour les tests

Étape 06 : Programme Python de génération du script SQL (CREATE ...)

- Vous devez réaliser une première version du programme Python demandé : **genererScriptSqlCreate.py**.

Données du programme	le fichier CSV que vous avez récupéré : usersToulouse.csv le fichier de mots de passe : usersPassword.csv
Résultat	Le script SQL creerUsersBddAcces.sql permettant dans le SGBD mariaDB : <ul style="list-style-type: none">- de créer les utilisateurs au format pnom- de créer une base de données pour chaque utilisateur (dbpnom) avec les privilèges suivant sur celle-ci :<ul style="list-style-type: none">- Créer, modifier et supprimer des tables et des colonnes- Insérer, modifier et supprimer des données- Interroger sa base de données

Le compte MariaDB sera créé de la façon suivante :

première lettre du prénom suivie du nom, le tout en minuscule.

Exemple : l'utilisateur **Alain LAMBERT** doit avoir le compte MariaDB **alambert** et avoir comme base de donnée **dbalambert**

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape et les captures d'écran du contenu des fichiers CSV, python et sql
- le programme python commenté **genererScriptSqlCreate.py**
- le script obtenu **creerUsersBddAcces.sql**.

Étape 07 : Test du script SQL généré (CREATE ...)

- Exécutez le script généré **creerUsersBddAcces.sql** sur votre serveur MariaDB **local** (VM Centos8).
- Réalisez ensuite les tests suivants pour 2 utilisateurs pris au hasard :
 - se connecter à sa base de données
 - créer 1 table
 - créer des colonnes dans cette table puis ajouter 2 enregistrements
 - récupérer des informations de cette table avec une requête
 - modifier un enregistrement puis le supprimer
 - ajouter une colonne dans une table puis la supprimer
 - supprimer la table

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, les tests réalisés avec les captures d'écran des résultats obtenus commentés.

Étape 08 : Programme Python de génération du script SQL (DROP ...)

Vous devez réaliser un second programme Python nommé **genererScriptSqlDrop.py** générant un script de suppression des utilisateurs et de leur base de donnée.

Données du programme	le fichier CSV que vous avez récupéré : usersToulouse.csv
Résultat	le script SQL de suppression des utilisateurs nommé et de leur base de données supprimerUsersBddAcces.sql

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape et les captures d'écran du contenu des fichiers CSV, python et sql
- le programme python commenté **genererScriptSqlDrop.py**
- le script obtenu **supprimerUsersBddAcces.sql**.

Étape 09 : Test du script SQL généré (DROP ...)

Exécutez le script **supprimerUsersBddAcces.sql** sur votre serveur MariaDB local. Vérifiez que tous les utilisateurs et leurs objets ont disparu.

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, les tests réalisés avec les captures d'écran des résultats obtenus commentés.

Étape 10 : Rédaction d'un mode opératoire

Vous réaliserez à l'intention de l'administrateur système, un mode opératoire en français puis sa traduction en anglais ou en espagnol. La traduction sera réalisée jeudi après-midi. Ce mode opératoire précisera :

- la récupération des utilisateurs *dans l'Active Directory*
- l'utilisation des programmes Python
- l'utilisation des scripts SQL générés

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, le mode opératoire et sa traduction

Étape 11 : Evolution 1 - Suppression des caractères spéciaux dans les noms d'utilisateur

Reprendre les fonctionnalités du programme **genererScriptSqlCreate.py** et mettre au point une nouvelle version de celui-ci, nommée **genererScriptSqlCreateV2.py** permettant de respecter les règles imposées par l'administrateur du système :

- prénoms composés : on juxtapose les initiales de chaque prénom
- le nom d'utilisateur ne contient que des caractères alphabétiques en majuscules ('A' à 'Z') ; tous les autres caractères seront éliminés.

Exemple : Jean-Baptiste LE FLOC'H doit donner **jblefloch**

Xavier Bertrand DE LA VILLETANAY doit donner **xbdelavilletanay**

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, un jeu d'essai et les captures d'écran du contenu des fichiers CSV, python et sql
- le programme python commenté **genererScriptSqlCreateV2.py**
- le script obtenu **creerUsersBddAcces.sql**

Étape 12 : Evolution 2 - Paramètres en ligne de commandes

Reprendre les fonctionnalités du programme **genererScriptSqlCreateV2.py** et mettre au point une nouvelle version du programme Python, nommée **genererScriptSqlCreateV3.py** permettant de fournir en paramètre sur la ligne de commande, le nom du fichier des utilisateurs ainsi que le nom du fichier résultat.

Exemple d'utilisation :

```
python3 genererScriptSqlCreateV3.py usersToulouse.csv creerUsersBddAcces.sql
```

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, une nouvelle version du mode opératoire avec captures d'écran, une capture d'écran du contenu du fichier **genererScriptSqlCreateV3.py**
- le programme python commenté **genererScriptSqlCreateV3.py**

Étape 13 : Evolution 3 - Mise en œuvre de "pattern" pour créer ou supprimer les utilisateurs

L'administrateur système souhaiterait n'avoir qu'un seul programme python permettant de générer au choix le script de **création** ou de **suppression** des utilisateurs et de leur base de données.

Reprendre les fonctionnalités du programme **genererScriptSqlCreateV3.py** et mettre au point une nouvelle version du programme Python, nommée **genererScriptSqlV4.py** permettant de générer les ordres SQL pour chaque utilisateur d'après un modèle (pattern) contenu dans un autre fichier CSV. Le script SQL généré sera appelé soit **creerUsersBddAcces.sql** (si création) ou **supprimerUsersBddAcces.sql** (si suppression).

Utilisation du fichier pattern de création **patternSqlCreate.csv**

Le fichier pattern contient les ordres SQL nécessaires pour créer un utilisateur "générique" (XXXXX), sa base de données et les privilèges associés. Le programme python récupère les commandes SQL du pattern, remplace le motif XXXXXX par le nom de chacun des utilisateurs (**pnom**) inscrits dans le fichier **usersToulouse.csv** et remplace le motif PPPPP par un mot de passe sécurisé à partir du fichier **usersPassword.xxx**

Exemple :

Si le fichier **patternSqlCreate.csv** contient CREATE USER XXXXX ...

Votre script Python générera les ordres SQL en remplaçant la chaîne XXXXX par les identifiants des utilisateurs.

```
CREATE USER malheinc ...
CREATE USER aallain ...
CREATE USER jaoustin ...
...
```

Utilisation du fichier pattern de suppression **patternSqlDrop.csv**

Le fichier pattern contient les ordres SQL nécessaires pour supprimer un utilisateur "générique" (XXXXX) et sa base de données. Le programme python récupère les commandes SQL du pattern et remplace le motif XXXXXX par le nom de chacun des utilisateurs (**pnom**).

Spécifications supplémentaires :

- Le fichier modèle pourra contenir plusieurs lignes.
- Le programme python acceptera les paramètres suivants sur la ligne de commande :
 - le nom du fichier modèle
 - le nom du fichier des utilisateurs
 - le nom du fichier de script SQL résultant

Exemple d'utilisation :

```
python3 genererScriptSqlV4.py patternSqlCreate.csv usersToulouse.csv creerUsersBddAcces.sql
```


A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, une nouvelle version du mode opératoire avec captures d'écran, une capture d'écran du contenu des fichiers CSV, python et SQL.
- le programme python commenté **genererScriptSqlV4.py**
- Les fichiers pattern **patternSqlCreate.csv** et **patternSqlDrop.csv**

Étape 14 : Evolution 4 - Exécution automatique du script SQL

On souhaite éviter les manipulations nécessaires pour exécuter les scripts SQL (par exemple, le démarrage d'un client MariaDB).

- Comment exécuter un script SQL depuis une ligne de commande Linux ?
- Comment exécuter une commande Linux depuis un script Python ?

Reprendre les fonctionnalités du programme **genererScriptSqlV4.py** et mettre au point une nouvelle version du programme Python, nommée **genererScriptSqlV5.py** permettant d'exécuter automatiquement le script SQL qu'il vient de générer.

Exemple d'utilisation :

```
python3 genererScriptSqlV5.py patternSqlCreate.csv usersToulouse.csv
```

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, une nouvelle version du mode opératoire avec captures d'écran, une capture d'écran du contenu des fichiers CSV, python et SQL.
- le programme python commenté **genererScriptSqlV5.py**

Étape 15 : Evolution 5 - Requêtes SQL exécutées par python

Il s'agit de créer un programme Python qui exécute directement les requêtes SQL d'ajout-suppression d'utilisateurs-bases-privilèges **dans le SGBD MariaDB** présent dans votre machine virtuelle.

Voici ci-dessous les étapes que vous pouvez aborder :

- Documentez-vous sur internet pour connaître le(s) nouveau(x) module(s) à installer et les ordres qui permettent de se connecter et d'exécuter des ordres SQL sur un SGBDR MariaDB à partir de python 3.

<https://mariadb.com/fr/resources/blog/how-to-connect-python-programs-to-mariadb/>

- Reprendre les fonctionnalités du programme **genererScriptSqlV5.py** et mettre au point une nouvelle version du programme Python, nommée **genererScriptSqlV6.py** permettant donc d'exécuter directement les ordres SQL dans le SGBD MariaDB et de mémoriser ceux-ci dans un fichier texte *.

** Les ordres SQL exécutés dans MariaDB sont mémorisés dans un fichier texte nommé automatiquement dans le programme python de la manière suivante :*

ordres_sql_AAAA_MM_JJ_hh_mm.txt (date et heure courante dans le nom du fichier)

AAAA : année **MM** : mois **JJ** : jour **hh** : heures **mm** : minutes

exemple : **ordres_sql_2021_12_17_15_30.txt**

Exemple d'utilisation :

```
python3 genererScriptSqlV6.py patternSqlCreate.csv usersToulouse.csv
```

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, une nouvelle version du mode opératoire avec captures d'écran, une capture d'écran du contenu des fichiers CSV, python et txt.
- le programme python commenté **genererScriptSqlV6.py**
- Les fichiers textes générés par les jeux d'essai (création..., suppression...)

Étape 16 : Evolution 6 - Mise en œuvre d'une interface graphique

Dans les étapes précédentes vous avez pu constater que les informations à fournir au script sont les suivantes :

- nom du fichier modèle (pattern) utilisé
- nom du fichier contenant la liste des utilisateurs en entrée
- nom du fichier de script SQL généré par votre programme Python

Il est possible d'envisager d'utiliser une interface graphique offrant la possibilité de saisir (ou de choisir) ces mêmes informations grâce à des éléments adaptés (zone de saisie par exemple) puis de lancer le traitement en cliquant simplement sur un bouton (ou même dans un menu).

Python permet de réaliser des interface graphiques grâce à une bibliothèque nommée **tkinter**.

<https://docs.python.org/fr/3/library/tkinter.html>

Pour pouvoir utiliser cette bibliothèque, il faut préalablement l'installer, car elle n'est pas nécessairement installée par défaut lorsque l'on installe Python.

Il ne vous reste alors plus qu'à réaliser une nouvelle version du script.

Le nouveau script, **genererScriptSqlV7.py**, pourra être lancé dans un terminal et sans paramètres particuliers, ce qui lancera l'interface graphique que vous aurez prévue, dans laquelle l'utilisateur n'aura plus qu'à saisir ou choisir les noms des fichiers pré-cités, puis à valider.

Dans l'interface graphique, il faut donc prévoir :

- 1 champ de saisie (ou de choix) du nom du fichier qui contiendra le modèle à utiliser
- 1 champ de saisie (ou de choix) du nom du fichier contenant la liste des utilisateurs
- 1 champ de saisie (ou de choix) du nom du fichier qui contiendra le script SQL généré
- 1 bouton (censé être cliqué après cette saisie) qui déclenchera le traitement
- 1 bouton permettant de quitter l'interface

Par contre, bien qu'il soit possible d'enjoliver l'interface à loisir, vous pourrez vous contenter d'une interface « moins esthétique », du moment qu'elle soit fonctionnelle.

Pour réaliser l'interface graphique et la rendre fonctionnelle, vous pourrez vous aider des nombreuses ressources disponibles sur Internet.

Points importants à prendre également en compte :

- (1) Comme Python est couramment utilisé en 2 versions : la 2 et la 3, des éléments de syntaxe peuvent changer entre ces 2 versions.

Par exemple la bibliothèque « tkinter » s'appelle

- **Tkinter** (avec « T » majuscule) pour Python 2
- **tkinter** pour Python 3 → **celle que vous utiliserez donc ici**

Il faut donc importer avec le bon nom de version dans le script Python.

A remettre :

- Un compte-rendu contenant l'énoncé de l'étape, une nouvelle version du mode opératoire avec captures d'écran, une capture d'écran du contenu des fichiers CSV, python et sql.
- le programme python commenté **genererScriptSqlV7.py**

ANNEXE - Récapitulatif des programmes et scripts demandés

Programme / Script	Fonctionnalités
usersToulouse.csv	Contient le nom et le prénom de tous les utilisateurs de l'unité d'organisation (OU) Toulouse.
usersPassword.ods (Calc) ou usersPassword.xlsx (Excel)	Contient une liste de mots de passe sécurisés, autant qu'il y a d'utilisateurs dans l'OU Toulouse
createPassword.py	Créer un fichier usersPassword2.csv contenant les mots de passe respectant des contraintes fortes
genererScriptSqlCreate.py	<p>Génère le script SQL permettant dans le SGBD mariaDB :</p> <ul style="list-style-type: none"> - de créer les utilisateurs au format pnom à partir du fichier usersToulouse.csv et leur mot de passe à partir du fichier usersPassword.xxx - de créer une base de données pour chaque utilisateur (dbpnom) avec les privilèges suivant sur celle-ci : <ul style="list-style-type: none"> - Créer, modifier et supprimer des tables et des colonnes - Insérer, modifier et supprimer des données - Interroger sa base de données <p>Résultat sortie : script SQL creerUsersBddAcces.sql</p>
creerUsersBddAcces.sql	<p>Contient les ordres SQL permettant, pour le SGBD mariaDB :</p> <ul style="list-style-type: none"> - de créer les utilisateurs (pnom) et leur mot de passe - de créer une base de données pour chaque utilisateur (dbpnom) avec les privilèges suivant sur celle-ci : <ul style="list-style-type: none"> - Créer, modifier et supprimer des tables et des colonnes - Insérer, modifier et supprimer des données - Interroger sa base de données
genererScriptSqlDrop.py	<p>Génère le script SQL permettant, dans le SGBD mariaDB :</p> <ul style="list-style-type: none"> - de supprimer les utilisateurs indiqués dans le fichier usersToulouse.csv - de supprimer la base de données de chaque utilisateur <p>Résultat sortie : script SQL supprimerUsersBddAcces.sql</p>
supprimerUsersBddAcces.sql	Contient les ordres SQL pour le SGBD mariaDB permettant de supprimer les utilisateurs indiqués et leur base de données
genererScriptSqlCreateV2.py	<p>Amélioration de genererScriptSqlCreate.py</p> <p>Permet en plus de respecter les règles imposées par l'administrateur du système :</p> <ul style="list-style-type: none"> - prénoms composés : initiales de chaque prénom juxtaposées - suppression dans le nom d'utilisateur de tous les caractères autres que les lettres A-Z.

Programme / Script	Fonctionnalités
genererScriptSqlCreateV3.py	Amélioration de genererScriptSqlCreateV2.py Permet en plus de fournir en paramètre sur la ligne de commande, le nom du fichier des utilisateurs usersToulouse.csv ainsi que le nom du fichier résultat creerUsersBddAcces.sql .
genererScriptSqlV4.py	Amélioration de genererScriptSqlCreateV3.py Génère le script SQL creerUsersBddAcces.sql ou supprimerUsersBddAcces.sql en fonction du modèle (pattern) choisi pour le SGBD mariaDB. - Si pattern patternSqlCreate <ul style="list-style-type: none"> - récupère les ordres SQL du pattern - pour chaque utilisateur trouvé dans le fichier usersToulouse.csv <ul style="list-style-type: none"> - recopie ces ordres SQL en remplaçant le motif XXXXX par le nom d'utilisateur (pnom) et le motif PPPPP par un mot de passe sécurisé à partir du fichier usersPassword.xxx Résultat sortie : script SQL creerUsersBddAcces.sql - Si pattern patternSqlDrop <ul style="list-style-type: none"> - récupère les ordres SQL du pattern - pour chaque utilisateur trouvé dans le fichier usersToulouse.csv <ul style="list-style-type: none"> - recopie ces ordres SQL en remplaçant le motif XXXXX par le nom d'utilisateur (pnom) Résultat sortie : script SQL supprimerUsersBddAcces.sql
genererScriptSqlV5.py	Amélioration de genererScriptSqlCreateV4.py Exécute en plus automatiquement le script SQL après sa génération

Programme / Script	Fonctionnalités
genererScriptSqlV6.py	<p>Amélioration de genererScriptSqlV5.py Exécution des ordres SQL directement dans le SGBD MariaDB.</p> <ul style="list-style-type: none"> - Si pattern patternSqlCreate <ul style="list-style-type: none"> - récupère les ordres SQL du pattern - pour chaque utilisateur trouvé dans le fichier usersToulouse.csv <ul style="list-style-type: none"> - recopie ces ordres SQL en remplaçant le motif XXXXX par le nom d'utilisateur (pnom) et le motif PPPPP par un mot de passe sécurisé à partir du fichier usersPassword.xxx - exécute tous les ordres SQL dans le SGBD MariaDB et les mémorise dans un fichier journal * - Si pattern patternSqlDrop <ul style="list-style-type: none"> - récupère les ordres SQL du pattern - pour chaque utilisateur trouvé dans le fichier usersToulouse.csv <ul style="list-style-type: none"> - recopie ces ordres SQL en remplaçant le motif XXXXX par le nom d'utilisateur (pnom) - exécute tous les ordres SQL dans le SGBD MariaDB et les mémorise dans un fichier journal * <p><i>* Les ordres SQL exécutés dans MariaDB sont mémorisés dans un fichier texte nommé automatiquement par le programme python de la manière suivante :</i> ordres_sql_AAAA_MM_JJ_hh_mm.txt <i>(date et heure courante dans le nom du fichier)</i> AAAA : année MM : mois JJ : jour hh : heures mm : minutes <i>exemple : ordres_sql_2021_12_17_15_30.txt</i></p>
genererScriptSqlV7.py	<p>Amélioration de genererScriptSqlV6.py Mise en œuvre d'une interface graphique permettant de sélectionner (ou choisir) les différents fichiers nécessaires à l'exécution, puis de lancer l'exécution à l'aide d'un bouton.</p>