

Livrable 1

Lecture et écriture dans un fichier CSV en Python

Objectif de l'étape :

L'objectif de cette étape est d'apprendre à lire un fichier CSV et à écrire dans un autre fichier CSV en Python. Le script permet de générer automatiquement des logins à partir d'un prénom et d'un nom, en respectant un format précis.

```
loginGeneratorBasic.py X
C: > Users > G L O R Y X > Desktop > PROJET PYTHON > TEST-LOGIN-GENERATOR > loginGeneratorBasic.py
1  # On ouvre le fichier d'entrée et le fichier de sortie
2  with open("testEntree.csv", "r", encoding="utf-8") as fichier_entree, \
3      | open("testSortie.csv", "w", encoding="utf-8") as fichier_sortie:
4
5      # On ignore la première ligne (l'en-tête)
6      fichier_entree.readline()
7
8      # On lit chaque ligne du fichier d'entrée
9      for ligne in fichier_entree:
10         # On enlève les retours à la ligne
11         ligne = ligne.strip()
12
13         # Si la ligne n'est pas vide
14         if ligne:
15             # On sépare le prénom et le nom
16             prenom, nom = ligne.split(";")
17
18             # On crée le login (1ère lettre du prénom + nom)
19             login = (prenom[0] + nom).lower()
20
21             # On écrit le login dans le fichier de sortie
22             fichier_sortie.write(login + "\n")
23
24
```

Avec ce script on passe de :

Prenom;NOM Emile;ZOLA Victor;HUGO Georges;SAND

à

ezola vhugo gsand

Livrable 2

Normalisation avancée des logins

Objectif de l'étape :

L'objectif de cette étape est de générer des logins normalisés à partir de prénoms et de noms complexes. Le script permet de gérer les prénoms composés en utilisant leurs initiales, de traiter les noms à particules ou composés, et de supprimer les caractères spéciaux ainsi que les accents. Cette normalisation garantit des logins simples, cohérents et compatibles avec les standards d'entreprise.

```
1  import unicodedata
2  import re
3
4  def supprimer_accents(texte):
5      texte = unicodedata.normalize("NFD", texte)
6      return "".join(c for c in texte if unicodedata.category(c) != "Mn")
7
8  def initiales_prenom(prenom):
9      parties = re.split(r"[ -]", prenom.strip())
10     return "".join(p[0] for p in parties if p)
11
12 def normaliser_nom(nom):
13     nom = supprimer_accents(nom)
14     nom = re.sub(r"[\s\-\_]", "", nom)
15     return nom.lower()
16
17 fichier_entree = "testEntreeSpeciaux.csv"
18 fichier_sortie = "testSortieNormaliser.csv"
19
20 with open(fichier_entree, "r", encoding="utf-8") as fin, \
21     open(fichier_sortie, "w", encoding="utf-8") as fout:
22
23     # en-tête du fichier de sortie
24     fout.write("login\n")
25
26     # on ignore l'en-tête du fichier source
27     next(fin)
28
29     for ligne in fin:
30         prenom, nom = ligne.strip().split(";")
31
32         prenom = supprimer_accents(prenom)
33         initiaux = initiales_prenom(prenom).lower()
34         nom_normalise = normaliser_nom(nom)
35
36         login = initiaux + nom_normalise
37         fout.write(login + "\n")
```

Avec ce script on passe de :

```
Prenom;NOM  
Alain;Lambert  
Jean-Baptiste;Le Floc'h  
Sophie;O'Connor  
François Xavier;DE LA VILLETANAY  
Émilie;DUPONT-THIBERT  
Anaïs;MARTÍNEZ-GÓMEZ  
Pierre-Paul;D'ALEMBERT  
Marie-Josée;ST-PIERRE
```

à

```
login  
alambert  
jblefloch  
soconnor  
fxdelavilletanay  
edupontthibert  
amartinezgomez  
ppdalembert  
mjstpierre
```

Livrable 3

Base de données et gestion des privilèges utilisateur

Objectif de l'étape :

L'objectif de cette étape est d'apprendre à créer une base de données MariaDB et un utilisateur associé en respectant le principe du moindre privilège. Il s'agit de configurer des identifiants personnalisés, d'attribuer uniquement les droits nécessaires à l'utilisateur sur sa propre base de données, puis de vérifier que les accès sont correctement restreints afin de garantir la sécurité du système.

Commandes SQL exécutées :

Création de la base et de l'utilisateur :

```
MariaDB [(none)]> CREATE DATABASE db_ZOZIO_N;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'nzozio'@'localhost' IDENTIFIED BY 'joliverie';
Query OK, 0 rows affected (0,002 sec)
```

Gestion des privilèges de l'utilisateur :

```
MariaDB [(none)]> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP ON db_ZOZIO_N.* TO 'nzozio'@'localhost';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'nzozio'@'localhost';
+-----+
| Grants for nzozio@localhost |
+-----+
| GRANT USAGE ON *.* TO 'nzozio'@'localhost' IDENTIFIED BY PASSWORD '*9BB7AEA65709C5196D4EE7D17BA412DE58D9CCB4' |
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON `db_ZOZIO_N`.* TO 'nzozio'@'localhost' |
+-----+
2 rows in set (0,000 sec)
```

Tests :

Création d'une table :



L'utilisateur n'accède uniquement qu'à sa base de données

Insérer des éléments :

```
1 INSERT INTO Test (ID, Prenom, NOM, Adresse, Ville)
2 VALUES (1, 'Jean', 'JACQUES', 'Nantes', '12 rue du Chêne');
```

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0.0005 seconde(s).)

SELECT * FROM `Test`

☐ Profilage [[Éditer en ligne](#)] [[Éditer](#)] [[Expliquer SQL](#)] [[Créer le code source PHP](#)] [[Actualiser](#)]

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtre les lignes:

Options supplémentaires

ID	Prenom	NOM	Adresse	Ville
1	Jean	JACQUES	Nantes	12 rue du Chêne

Pour finir la base de donnée est supprimée avec :

```
DROP DATABASE db_ZOZIO_N ;
```

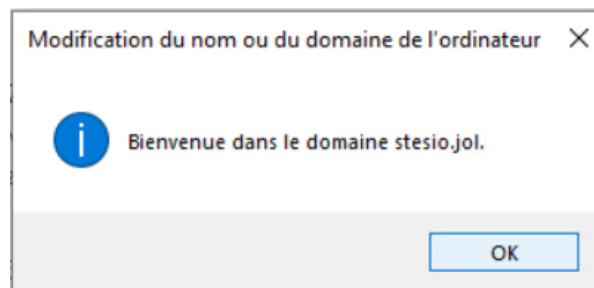
Livrable 4

Intégration d'une station Win10 et extraction d'informations Active Directory

Objectif de l'étape :

L'objectif de cette étape est d'intégrer une station Windows 10 au domaine **stesio.jol** afin de permettre l'accès aux ressources du domaine et l'exécution de scripts PowerShell. Elle vise également à vérifier le bon rattachement de la machine au domaine à l'aide de tests de connectivité et de commandes d'identification, et à préparer l'extraction d'informations utilisateurs depuis l'Active Directory dans un contexte sécurisé et contrôlé

Intégration au domaine stesio.jol :



Test de connectivité au domaine (LDAP) :

```
PS C:\Users\binome37> Test-NetConnection -ComputerName stesio.jol -Port 389

ComputerName      : stesio.jol
RemoteAddress     : 10.15.11.210
RemotePort        : 389
InterfaceAlias    : Ethernet0
SourceAddress     : 10.15.12.143
TcpTestSucceeded  : True

PS C:\Users\binome37>
```

Extraction des utilisateurs de l'OU Perpignan à l'aide d'un script powershell :

```
# Parametres
$Server = "10.15.11.210"
$OUPath = "OU=Perpignan,OU=Finances,DC=stesio,DC=jol"
$OutputFile = "D:\powershell\usersPerpignan.csv"
# Recupere les utilisateurs et exporte en CSV

try {
    Get-ADUser -Server $Server -SearchBase $OUPath -Filter * -Properties givenName,
    sn, displayName, mail, telephoneNumber |
    Select-Object givenName, sn, displayName, mail, telephoneNumber |
    Export-Csv -Path $OutputFile -Delimiter ";" -Encoding UTF8
    -NoTypeInformation -Force
} catch {
    Write-Host "Erreur : $_" -ForegroundColor Red
    Write-Host "Verifie : "
    Write-Host "1. Le chemin de l'OU : $OUPath"
    Write-Host "2. Tes droits d'accès sur ce serveur et cette OU."
    Write-Host "3. Que le module ActiveDirectory est installé."
}
```

```
PS C:\Users\binome37> C:\Users\binome37\Desktop\extractUsersPerpignan.ps1
Export réussi : D:\powershell\usersPerpignan.csv

PS C:\Users\binome37> |
```

Résultat après exécution :

```
1  {"givenName";"sn";"displayName";"mail";"telephoneNumber"
2  "Julie";"PRIGENT";"PRIGENT Julie";"julie.prigent@stesio.jol";"0159000373"
3  "Caroline";"PARMENTIER";"PARMENTIER Caroline";"caroline.parmentier@stesio.jol";"0159000383"
4  "Eva";"DUCLOS";"DUCLOS Eva";"eva.duclos@stesio.jol";"0459000393"
5  "Enzo";"HAMEL";"HAMEL Enzo";"enzo.hamel@stesio.jol";"0459000403"
6  "Pierre";"COMTE";"COMTE Pierre";"pierre.comte@stesio.jol";"0459000413"
7  "Julie";"SIMONIN";"SIMONIN Julie";"julie.simonin@stesio.jol";"0159000773"
8  "Caroline";"DE PLATEL";"DE PLATEL Caroline";"caroline.deplatel@stesio.jol";"0159000783"
9  "Eva";"REDON";"REDON Eva";"eva.redon@stesio.jol";"0159000793"
10 "Enzo";"LARCHER";"LARCHER Enzo";"enzo.larcher@stesio.jol";"0159000803"
11 "Pierre";"MANCEAU";"MANCEAU Pierre";"pierre.manceau@stesio.jol";"0159000813"
12 "Jean-Baptiste";"LE FLOC'H";"LE FLOC'H Jean-Baptiste";;
13 "Enzo";"THOMAS";"THOMAS Enzo";"enzo.thomas@stesio.jol";"0159000803"
14 "Pierre";"LEFEBVRE";"LEFEBVRE Pierre";"pierre.lefebvre@stesio.jol";"0159000813"
15 "Maxime";"LAMBERT";"LAMBERT Maxime";"maxime.lambert@stesio.jol";"0159000823"
16 "Camille";"WEBER";"WEBER Camille";"camille.weber@stesio.jol";"0159000833"
17 "Julie";"ETIENNE";"ETIENNE Julie";"julie.etienne@stesio.jol";"0159000843"
18 "Caroline";"LEMAITRE";"LEMAITRE Caroline";"caroline.lemaitre@stesio.jol";"0159000853"
19 "Eva";"PICHON";"PICHON Eva";"eva.pichon@stesio.jol";"0159000863"
20 "Enzo";"OLLIVIER";"OLLIVIER Enzo";"enzo.ollivier@stesio.jol";"0159000873"
21 "Camille";"GALLET";"GALLET Camille";"camille.gallet@stesio.jol";"0159000883"
22
```

Extraction des utilisateurs de l'OU Toulouse à l'aide du même script powershell :

```
"givenName";"sn";"displayName";"mail";"telephoneNumber"
"Charlotte";"RUIZ";"RUIZ Charlotte";"charlotte.ruiz@stesio.jol";"0159000382"
"Noemie";"SAMSON";"SAMSON Noemie";"noemie.samson@stesio.jol";"0159000392"
"Myrtille";"BONHOMME";"BONHOMME Myrtille";"myrtille.bonhomme@stesio.jol";"0459000402"
"Arthur";"LELEU";"LELEU Arthur";"arthur.leleu@stesio.jol";"0459000412"
"Emilie";"BRIERE";"BRIERE Emilie";"emilie.briere@stesio.jol";"0159000772"
"Charlotte";"AUVRAY";"AUVRAY Charlotte";"charlotte.auvray@stesio.jol";"0159000782"
"Noemie";"PAGE";"PAGE Noemie";"noemie.page@stesio.jol";"0159000792"
"Malcom";"PORTIER";"PORTIER Malcom";"malcom.portier@stesio.jol";"0159000802"
"Arthur";"VERGNE";"VERGNE Arthur";"arthur.vergne@stesio.jol";"0159000812"
"Xavier Bertrand";"DE LA VILLETANAY";"DE LA VILLETANAY Xavier Bertrand";;
"Élodie";"DUPONT-MARTIN";"DUPONT-MARTIN Élodie";"elodie.dupont-martin@stesio.jol";"0259000102"
"Jean-Luc";"O'CONNOR";"O'CONNOR Jean-Luc";"jean-luc.oconnor@stesio.jol";"0259000112"
"Anaïs";"DE LA FONTAINE";"DE LA FONTAINE Anaïs";"anaïs.delafontaine@stesio.jol";"0259000122"
"Pierre-François";"LEFÈVRE";"LEFÈVRE Pierre-François";"pierre-francois.lefevre@stesio.jol";"0259000132"
"Mélanie";"D'ANGELIS";"D'ANGELIS Mélanie";"melanie.dangelis@stesio.jol";"0259000142"
"Sophie";"VAN DEN BERGHE";"VAN DEN BERGHE Sophie";"sophie.vandenberghes@stesio.jol";"0259000152"
"Michael";"BERNARD";"BERNARD Michael";"michael.bernard@stesio.jol";"0159000002"
"François Xavier";"SAINT-ETIENNE";"SAINT-ETIENNE François Xavier";"francoisxavier.saint-etienne@stesio.jol";"0259000162"
"Cécile";"MÜLLER";"MÜLLER Cécile";"cecile.muller@stesio.jol";"0259000172"
"Marie-Claire";"LO'PEZ";"LO'PEZ Marie-Claire";"marie-claire.lopez@stesio.jol";"0259000192"
"Théo";"VAN HOUTEN";"VAN HOUTEN Théo";"theo.vanhouten@stesio.jol";"0259000202"
"Léa";"DUPUIS-MOREAU";"DUPUIS-MOREAU Léa";"lea.dupuis-moreau@stesio.jol";"0259000212"
"Marc-André";"LOISELLE";"LOISELLE Marc-André";"marc-andre.loiselle@stesio.jol";"0259000302"
"Arthur";"MICHEL";"MICHEL Arthur";"arthur.michel@stesio.jol";"0159000012"
"Sebastien";"DUPONT";"DUPONT Sebastien";"sebastien.dupont@stesio.jol";"0159000022"
"Ines";"MARCHAL";"MARCHAL Ines";"ines.marchal@stesio.jol";"0159000162"
"Emilie";"GILLET";"GILLET Emilie";"emilie.gillet@stesio.jol";"0159000172"
"Charlotte";"GAY";"GAY Charlotte";"charlotte.gay@stesio.jol";"0159000182"
"Noemie";"CORDIER";"CORDIER Noemie";"noemie.cordier@stesio.jol";"0159000192"
"Martin";"BUISSON";"BUISSON Martin";"martin.buisson@stesio.jol";"0159000202"
"Ines";"DOS SANTOS";"DOS SANTOS Ines";"ines.dos.santos@stesio.jol";"0159000362"
"Emilie";"LEVY";"LEVY Emilie";"emilie.levy@stesio.jol";"0159000372"
```

Livrable 5

Génération de mots de passe robustes avec un script Python

Objectif de l'étape :

L'objectif de cette étape est de mettre en place des scripts Python permettant de générer et de vérifier des mots de passe robustes conformément à une politique de sécurité stricte. Il s'agit de produire automatiquement des mots de passe uniques respectant des contraintes précises de longueur et de complexité, de les stocker dans un fichier CSV, puis de valider une liste de mots de passe afin de détecter les non-conformités et les doublons, dans une démarche de renforcement de la sécurité des comptes utilisateurs.

Script disponible en pièce jointe

Exécution du Script :

On obtient un fichier .csv avec le nombre de mots de passe spécifié :

```
1 password
2 gVeu9.&*Y6w/
3 z.fBPC2-4Bnw
4 F*/G9>sY2qIp
5 P%. :95UvF=x-
6 -8VfY.t5=9WT
7 a=-rpLy/G12B
8 1-6c@QLd#vgk
9 2Tws?EY1Q<FV
10 5BA5<hXuRn@b
11 5Sc8J=Puk#qe
12 vIP?6Dq2xGf!
13 :P2yQ:*@5vqf
```

Il nous servira plus tard pour générer les mots de passe des utilisateurs de la base de donnée

Livrable 6

Génération des logins utilisateurs pour MariaDB

Objectif de l'étape :

Nous allons créer un script Python nommé 'genererLoginToulouse.py' qui lira les fichiers CSV pour récupérer les informations des utilisateurs, nettoiera et normalisera les noms et prénoms en suivant la logique du script loginNormaliser.py, puis générera un fichier loginToulouse.csv
(Script disponible en pièce jointe)

Exécution du Script :

```
1 login;password;lastname;firstname
2 cruiz;gVeu9.&*Y6w/;RUIZ;Charlotte
3 nsamson;z.fBPC2-4Bnw;SAMSON;Noemie
4 mbonhomme;F*/G9>sY2qIp;BONHOMME;Myrtille
5 aleleu;P%.:95UvF=x-;LELEU;Arthur
6 ebriere;-8VfY.t5=9WT;BRIERE;Emilie
7 cauvray;a=-rpLy/G12B;AUVRAY;Charlotte
8 npage;1-6c@QLd#vgk;PAGE;Noemie
9 mportier;2Tws?EY1Q<FV;PORTIER;Malcom
10 avergne;5BA5<hXuRn@b;VERGNE;Arthur
11 xdelavilletanay;5Sc8J=Puk#qe;DE LA VILLETANAY;Xavier bertrand
```

Ce fichier CSV servira à la création des utilisateurs de la base de donnée.

Livrable 7

Création du script SQL pour utilisateurs et bases MariaDB

Objectif de l'étape :

L'objectif de cette étape est d'automatiser la création des comptes utilisateurs et de leurs bases de données sur le serveur MariaDB en générant un script SQL à partir des données normalisées contenues dans **loginToulouse.csv**. Le script permettra de créer les utilisateurs et leurs bases associées avec les privilèges nécessaires pour gérer leurs tables et données, tout en garantissant que chaque utilisateur n'ait accès qu'à sa propre base. Cette étape inclut également la vérification du bon fonctionnement des comptes et des droits sur la base de données.

(Script disponible en pièce jointe)

Exécution du Script :

```
import csv # Import du module pour manipuler les fichiers CSV

# Nom du fichier CSV contenant les logins, mots de passe, noms et prénoms
fichier_csv = "loginToulouse.csv"

# Nom du fichier SQL qui sera généré
fichier_sql = "creerUsersBddAcces.sql"

# Ouverture du fichier CSV en lecture et du fichier SQL en écriture
with open(fichier_csv, newline='', encoding="utf-8") as f_csv, \
    open(fichier_sql, "w", encoding="utf-8") as f_sql:

    # Création d'un lecteur CSV qui interprète la première ligne comme des noms de colonnes
    reader = csv.DictReader(f_csv, delimiter=';')

    # Parcours de chaque ligne du fichier CSV
    for row in reader:
        # Récupération du login et du mot de passe depuis le CSV
        login = row["login"]
        password = row["password"]

        # Nom de la base de données associé à l'utilisateur (préfixe st_ pour STESIO)
        bdd = f"st_{login}"

        # Écriture des commandes SQL dans le fichier
        # Création de la base de données si elle n'existe pas
        f_sql.write(f"CREATE DATABASE IF NOT EXISTS {bdd};\n")

        # Création de l'utilisateur MariaDB avec son mot de passe
        f_sql.write(
            f"CREATE USER IF NOT EXISTS '{login}'@'localhost' IDENTIFIED BY '{password}';\n"
        )

        # Attribution des privilèges nécessaires sur la base de données de l'utilisateur
        f_sql.write(
            f"GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER "
            f"ON {bdd}.* TO '{login}'@'localhost';\n"
```

Fichier de sortie :

```
CREATE DATABASE IF NOT EXISTS st_cruiz;
CREATE USER IF NOT EXISTS 'cruiz'@'localhost' IDENTIFIED BY 'gVeu9.&*Y6w/';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_cruiz.* TO 'cruiz'@'localhost';
FLUSH PRIVILEGES;

CREATE DATABASE IF NOT EXISTS st_nsamson;
CREATE USER IF NOT EXISTS 'nsamson'@'localhost' IDENTIFIED BY 'z.fBPC2-48nw';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_nsamson.* TO 'nsamson'@'localhost';
FLUSH PRIVILEGES;

CREATE DATABASE IF NOT EXISTS st_mbonhomme;
CREATE USER IF NOT EXISTS 'mbonhomme'@'localhost' IDENTIFIED BY 'F*/G9>sY2qIp';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_mbonhomme.* TO 'mbonhomme'@'localhost';
FLUSH PRIVILEGES;

CREATE DATABASE IF NOT EXISTS st_aleleu;
CREATE USER IF NOT EXISTS 'aleleu'@'localhost' IDENTIFIED BY 'P%.:95UvF=x-';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_aleleu.* TO 'aleleu'@'localhost';
FLUSH PRIVILEGES;
```

Cela nous permettra d'exécuter directement le script dans phpmyadmin

Livrable 8

Création du script SQL de suppression utilisateurs et BDD

Objectif de l'étape :

Nous allons automatiser la suppression des comptes utilisateurs MariaDB et de leurs bases de données associées en générant un script SQL à partir d'un fichier CSV, puis en l'exécutant afin de garantir une suppression sécurisée sur le serveur.

(Script disponible en pièce jointe)

Exécution du Script :

```
#!/usr/bin/env python3
import csv

fichier_csv = "loginToulouseDelete.csv"
fichier_sql = "supprimerUsersBddAcces.sql"

with open(fichier_csv, newline='', encoding="utf-8") as f_csv, \
    open(fichier_sql, "w", encoding="utf-8") as f_sql:

    reader = csv.DictReader(f_csv, delimiter=';')

    for row in reader:
        login = row["login"]
        bdd = f"st_{login}"

        f_sql.write(f"DROP DATABASE IF EXISTS {bdd};\n")
        f_sql.write(f"DROP USER IF EXISTS '{login}'@'localhost';\n")
        f_sql.write("FLUSH PRIVILEGES;\n\n")

print(f"Script SQL généré : {fichier_sql}")
```

Fichier de sortie (extrait) :

```
1 DROP DATABASE IF EXISTS st_cruiz;
2 DROP USER IF EXISTS 'cruiz'@'localhost';
3 FLUSH PRIVILEGES;
4
5 DROP DATABASE IF EXISTS st_nsamson;
6 DROP USER IF EXISTS 'nsamson'@'localhost';
7 FLUSH PRIVILEGES;
8
9 DROP DATABASE IF EXISTS st_mbonhomme;
10 DROP USER IF EXISTS 'mbonhomme'@'localhost';
11 FLUSH PRIVILEGES;
12
13 DROP DATABASE IF EXISTS st_aleleu;
14 DROP USER IF EXISTS 'aleleu'@'localhost';
15 FLUSH PRIVILEGES;
16
17 DROP DATABASE IF EXISTS st_ebriere;
18 DROP USER IF EXISTS 'ebriere'@'localhost';
19 FLUSH PRIVILEGES;
20
21 DROP DATABASE IF EXISTS st_cauvray;
22 DROP USER IF EXISTS 'cauvray'@'localhost';
23 FLUSH PRIVILEGES;
```

Cela nous permettra d'exécuter directement le script dans phpmyadmin

Importation de nos scripts .sql dans phpmyadmin :

Extrait du script de création des utilisateurs et de leurs bases de données :

```
CREATE DATABASE IF NOT EXISTS st_mbuisson;
CREATE USER IF NOT EXISTS 'mbuisson'@'localhost' IDENTIFIED BY '/m3.K>?9sCkZ';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_mbuisson.* TO 'mbuisson'@'localhost';
FLUSH PRIVILEGES;

CREATE DATABASE IF NOT EXISTS st_idossantos;
CREATE USER IF NOT EXISTS 'idossantos'@'localhost' IDENTIFIED BY '676*5a.ahaTS';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_idossantos.* TO 'idossantos'@'localhost';
FLUSH PRIVILEGES;

CREATE DATABASE IF NOT EXISTS st_elevy;
CREATE USER IF NOT EXISTS 'elevy'@'localhost' IDENTIFIED BY 'jJ&7rx-S96*j';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON st_elevy.* TO 'elevy'@'localhost';
FLUSH PRIVILEGES;
```

Exécution :

Activités Firefox 7 janv. 22:33

localhost / localhost | php x Courrier - ZOZIO Nicolas x SQL Apostrophe Error Fix x +

http://localhost/phpmyadmin/index.php?route=/server/privileges&viewing_mode=server

phpMyAdmin

Récentes Préférences

performance_schema

st

st_adelafontaine

st_aleleu

st_amichel

st_avergne

st_cauvray

st_cgay

st_emiller

Serveur : localhost

	Bases de données	SQL	État	Comptes utilisateurs	Exporter	Importer	Para
☐	edupontmartin	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	egillet	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	elevy	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	fsaintetienne	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	idossantos	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	imarchal	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	joconnor	localhost	Oui	USAGE	Non	Exporter	Verrouiller
☐	ldupuismoreau	localhost	Oui	USAGE	Non	Exporter	Verrouiller

Extrait du script de suppression des utilisateurs :

```
DROP DATABASE IF EXISTS st_aleleu;  
DROP USER IF EXISTS 'aleleu'@'localhost';  
FLUSH PRIVILEGES;  
  
DROP DATABASE IF EXISTS st_ebriere;  
DROP USER IF EXISTS 'ebriere'@'localhost';  
FLUSH PRIVILEGES;  
  
DROP DATABASE IF EXISTS st_cauvray;  
DROP USER IF EXISTS 'cauvray'@'localhost';  
FLUSH PRIVILEGES;  
  
DROP DATABASE IF EXISTS st_npage;  
DROP USER IF EXISTS 'npage'@'localhost';  
FLUSH PRIVILEGES;
```

Exécution :

