**Question 1.1 : L ’extension CSV**

L’extension CSV (**C**omma-**S**eparated **V**alues) qui signifie valeurs séparées par des virgules en français, désigne un fichier texte dans lequel sont stockées des données sous forme de tableau avec des headers et des données.

Dans un fichier.csv les données sont organisées en ligne, chacune représente une ligne d’un tableau. La première ligne d’un fichier CSV décrit le nom des colonnes permettant ainsi d’avoir un aperçu des données contenues.

La fin d’une ligne est définie par un retour à la ligne :

Les virgules servent de séparateurs de valeur/données.

**Question 1.2 : Utilité**

Si les fichiers CSV sont couramment utilisés en programmation et dans les scripts, c’est pour leur structure simple qui facilite la manipulation, l’exportation et l’importation de données en grandes quantités.

En plus d’occuper peu d’espace de stockage, les données issues d'un fichier CSV sont simples à interpréter pour un humain et faciles à traiter, car même un éditeur de texte est suffisant.

Le format CSV peut être lu par de nombreux langages de programmation et de base de données ce qui facilite l’échange de données entre différents systèmes et applications.

**Question 1.3 : Choix du gestionnaire de MDP**

Une image contenant texte, Police, logo, Graphique

Description générée automatiquementPour la 1ère étape, nous avons choisi « Bitwarden » pour générer les mots de passe des utilisateurs. Si nous avons pris cette décision, c’est parce que c’est un gestionnaire de mot de passe gratuit qui utilise un chiffrement AES (**A**dvanced **E**ncryption **S**tandard) 256 bits à l’instar des banques et gouvernements. La génération de mots de passe a donc fait ses preuves et est très facile d’utilisation (simple copier-coller depuis l’extension).

**Question 1.4 : schéma hiérarchique**

**Déroulement des différentes étapes**

**A screenshot of a chat

Description automatically generated**

Pour chaque ligne du CSV importé, nous allons récupérer les OU

A finir

A screenshot of a chat

Description automatically generated

Pour chaque ligne du CSV importé, nous allons récupérer la date d’expiration qui a été insérée en accord avec les consignes de l’exercice.

IMAGE NEEDED CSV

Comme à notre habitude, on vérifie si les utilisateurs sont déjà dans l’AD. Si non, on signale avec le message ci-dessous qu’il n’a pas été trouvé puis on passe au suivant.

IMAGE NEEDED

Si oui, on met à jour la date d’expiration de l’utilisateur dans l’AD et ce, qu’il en a déjà une ou non.

Ce message s’affiche à chaque MàJ correctement effectuée :

IMAGE NEEDED

IMAGE NEEDED AD

Lorsque tous les utilisateurs ont été inséré, le script arrête son exécution.A screenshot of a chat

Description automatically generated

Pour chaque ligne du CSV importé, nous allons récupérer la date d’expiration qui a été insérée en accord avec les consignes de l’exercice.

IMAGE NEEDED CSV

Comme à notre habitude, on vérifie si les utilisateurs sont déjà dans l’AD. Si non, on signale avec le message ci-dessous qu’il n’a pas été trouvé puis on passe au suivant.

IMAGE NEEDED

Si oui, on met à jour la date d’expiration de l’utilisateur dans l’AD et ce, qu’il en a déjà une ou non.

Ce message s’affiche à chaque MàJ correctement effectuée :

IMAGE NEEDED

IMAGE NEEDED AD

Lorsque tous les utilisateurs ont été inséré, le script arrête son exécution.

A screenshot of a cell phone

Description automatically generated

Comme toujours, on importe le CSV.

On sélectionne ensuite chaque OU uniques et on fait une liste de tous les utilisateurs qui y sont. Le foreach est dans une variable pour la simple raison que l’on souhaite utiliser la longue liste d’utilisateur que l’on a généré en dehors de la boucle foreach.

Voici le résultat brut :

IMAGE NEEDED

Ensuite on export les données dans un CSV. On y ajoute le paramètre   
-NoTypeInformation pour éviter cette ligne qui s’avèrera être gênante si laissée tel quel

IMAGE NEEDED IWTHOUT NOTYPEINFO

Voici le résultat avec -NoTypeInformation, que l’on utilisera :

IMAGE NEEDED WITH NOTYPEINFO

On récupère la date actuelle car elle sera le nom du fichier compressé que l’on va créer. Le format n’étant pas spécifier dans la consigne, nous avons choisi le format de date ‘yyyyMMdd’.

Maintenant, la compression. Elle s’avère être simple à expliquer :

Tout d’abord, on précise le fichier que l’on veut compresser. Ensuite, nous décidons du chemin de destination. Le chemin d’accès précisé change le nom du fichier en la date actuelle (grâce à la variable $date) Et finalement, on décide de la méthode de compression. On a choisi optimal parce que ça nous paraissait la meilleure option. Un message s’affiche pour signaler le bon déroulement de la procédure :

IMAGE NEEDED

Dès la compression terminée, on déplace le fichier .zip dans la destination demandée par la consigne (C:\Workspace\Back\_up). Directory -Force sert à créer de force le dossier ‘Back\_up ‘ et ce même s’il existe déjà pour éviter une erreur. Voici le message de réussite :

IMAGE NEEDED

Le script arrête son exécution après avoir fini sa tâche.

**Problèmes rencontrés**

L’organisation avec nos plages horaires respectives et les complications personnelles a été un point compliqué de ce projet, mais malgré cela, nous avons su trouver le temps et l’énergie nécessaire à l’accomplissement du projet.

Nous avons eu de la peine à démarrer le projet dès le début et avons laissé le travail s’accumuler ce qui nous en a fourni d’avantage et en un temps restreint.

Difficulté à comprendre comment récupérer les données d’une colonne depuis le CSV et comment split un string.

**Conclusion du projet**

Ce projet était intéressant et complexe à la fois. Il a permis à chacun d’étendre ses connaissances respectives. Il nous a demandé un partage régulier de nos avancées et d’organiser à l’avance le rôle de chacun et les délais dans lesquels les tâches devaient être accomplies. Bien qu’une bonne organisation soit difficile à acquérir, cela s’est fait plutôt naturellement.

**Conclusion personnelle**

Loré :

Ce projet m’a aidé à comprendre le format CSV d’avantages et à comprendre son utilité en programmation. J’ai aussi appris à utiliser un fichier.csv avec PowerShell et j’ai eu un aperçu des travaux de gestion des identités et des accès.

David :

**Sources docu :**

<https://en.wikipedia.org/wiki/Comma-separated_values>

<https://www.bibl.ulaval.ca/geostat/statistiques/CSV-guide.pdf>

<https://bitwarden.com/products/>

**Sources code :**

<https://stackoverflow.com/questions/52012837/split-trim-in-a-single-step>

<https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.archive/compress-archive?view=powershell-7.4>

<https://www.alitajran.com/import-csv-delimiter-powershell/>

<https://lazyadmin.nl/powershell/get-aduser/#get-aduser-searchbase>

<https://stackoverflow.com/questions/34749333/powershell-guidelines-for-confirm-force-and-whatif>

<https://hodentekhelp.blogspot.com/2016/09/how-do-you-get-date-and-time.html>