|  |
| --- |
| Pré-requis |
| * **WMF 5.1 (Windows Management Framework) :** [**https://docs.microsoft.com/fr-fr/powershell/wmf/5.1/install-configure**](https://docs.microsoft.com/fr-fr/powershell/wmf/5.1/install-configure) * **PSSQLITE Module pour PowerShell :** [**https://github.com/RamblingCookieMonster/PSSQLite**](https://github.com/RamblingCookieMonster/PSSQLite) * **Scripts Project DBLite :** [**https://github.com/OxyDeV2/Project-DBLite.git**](https://github.com/OxyDeV2/Project-DBLite.git) |
| Partie 1 : Installation |
| Pour pouvoir utiliser les différents scripts proposés ici il vous faudra installer l’environnement adéquats, c’est-à-dire Windows Management Framework (WMF).  Une fois WMF installé, assurez-vous d’avoir la dernière version de Powershell  Lancez le terminal de commande « Windows+R » > cmd.  Une fois dans l’invite de commande, tapez « Powershell », et ensuite « $PsVersionTable ».    Pour le bon fonctionnement des scripts, il faut ***IMPERATIVEMENT*** avoir télécharger le module PSSQLITE. Placez le dossier dans  **C:\Program Files\WindowsPowerShell\Modules** ou **C:\Users\dacosta\Documents\WindowsPowerShell\Modules** |
|  |
| Partie 2 : Démarrage et explication des différents scripts |

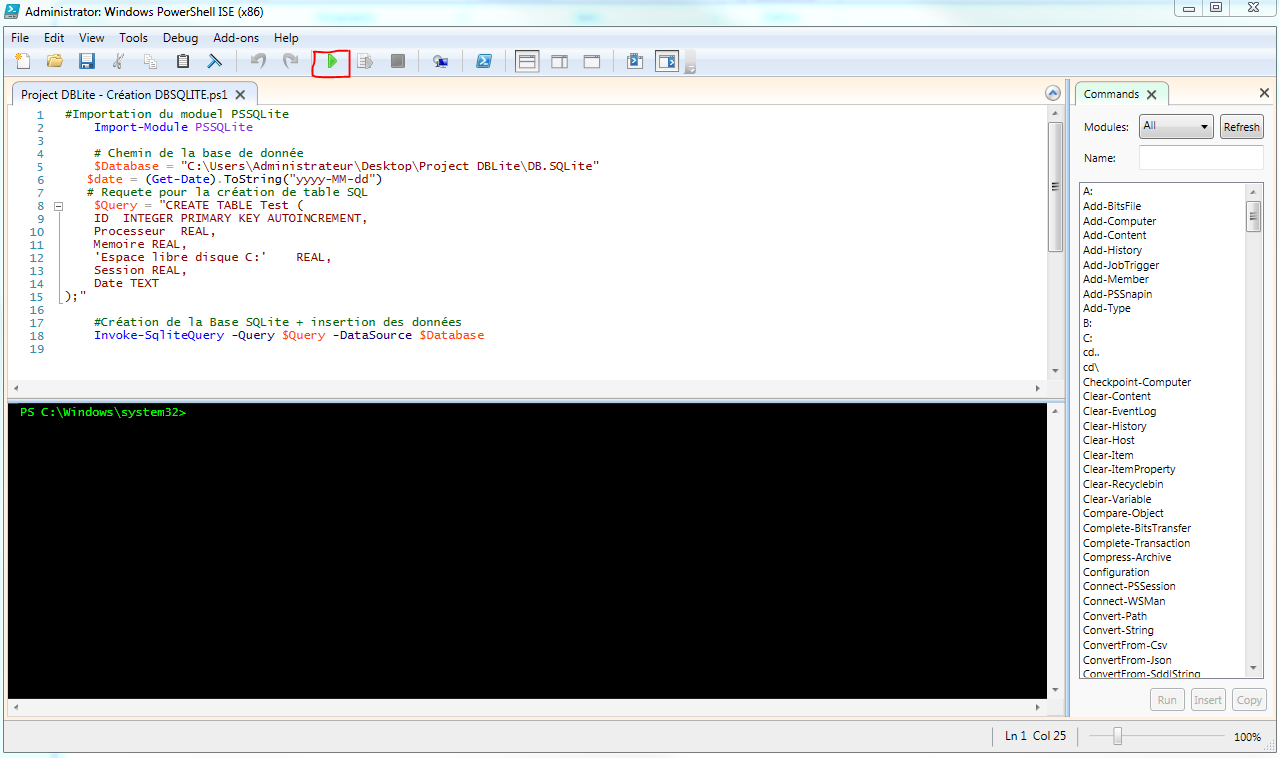
Pour commencer la partie 2, il vous faudra télécharger les scripts.

Une fois ceci accomplie, ouvrez les Scripts en mode Powershell ISE.

Clic droit > Ouvrir avec > Powershell ISE.

En mode édition vous pourrez donc voir comment le scripts a été composé (etc..)

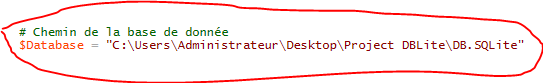
Pour le lancer notre premier script il vous suffira de cliquer sur la petite flèche verte en haut, le script va se lancer et donc générer la base de données.

****

**/ ! \ VEILLEZ A BIEN CHANGER LES CHEMIN DE DESTINATION / ! \**

Une fois le script exécuté, la base de donnée à été créer à l’emplacement spécifié dans le script.

Dans notre cas le chemin utilisé est celui-ci.



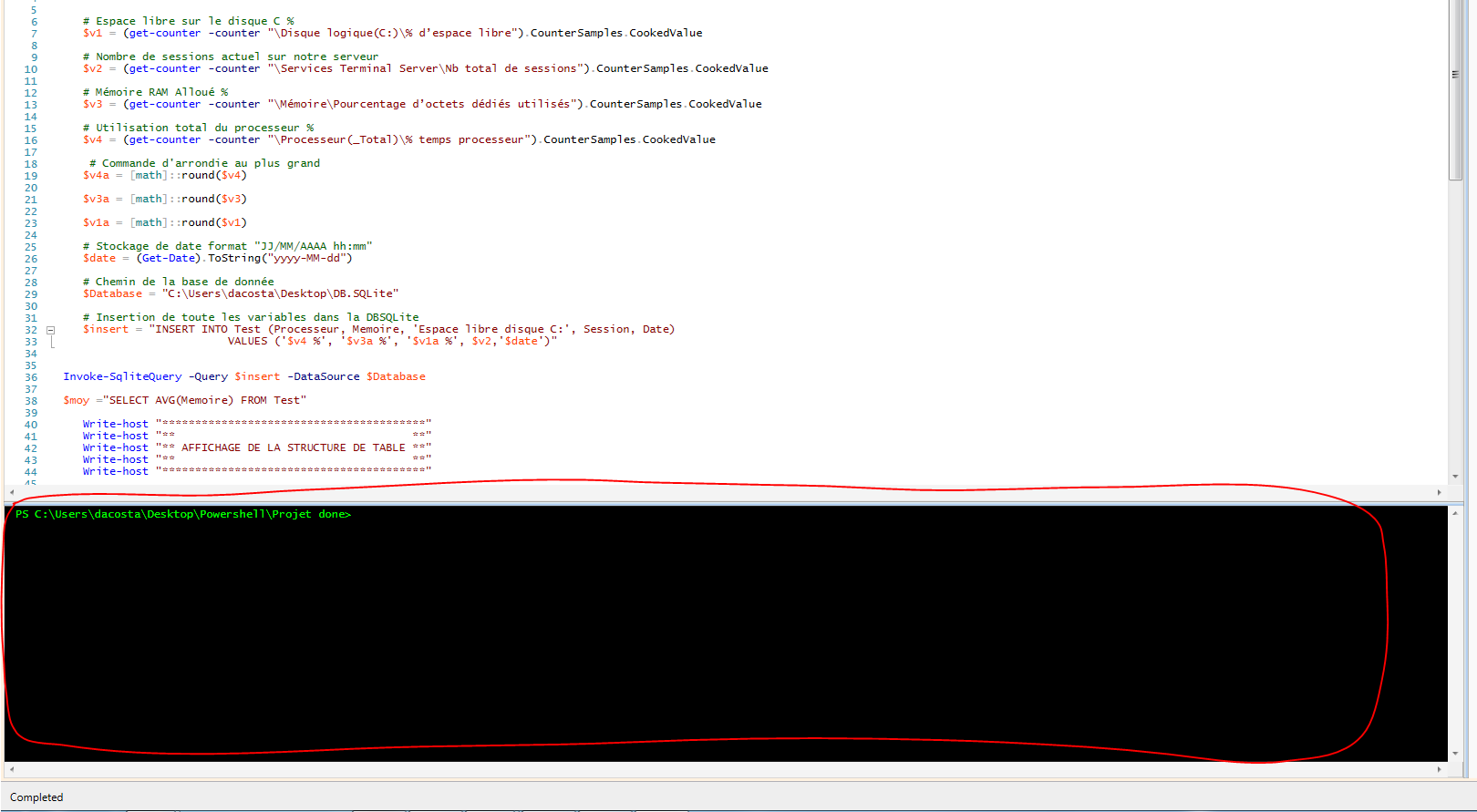
Le script a généré un fichier nommé « DB.SQLite ».

Nous pouvons maintenant passer sur le script suivant : L’insertion des différentes valeurs , sur les performances de la machines , dans la base donnée , pour pouvoir par la suite, traiter les données.

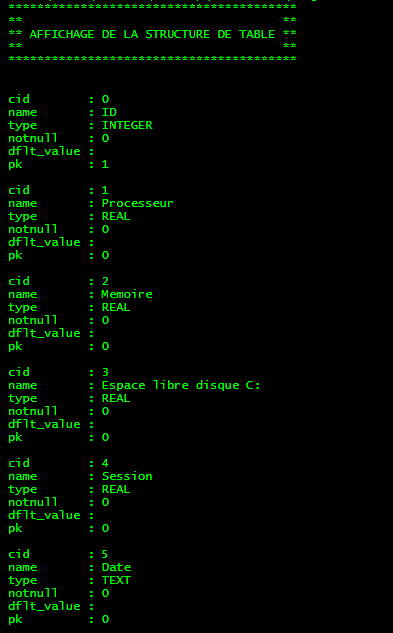
Ouvrez le second script nommé « Insertion DBSQLITE »  pour pouvoir remplir la base de données.

Comme dans le cas précèdent n’oubliez pas de changer l’emplacement de destination de la base de données !

La console Powershell ISE situé en bas vous indique si votre programme a été exécuté correctement et nous sert d’affichage pour différentes utilisations



Exécuter le script et voyons ce qu’il nous retourne :



Le script va afficher la structure détaillée de la table

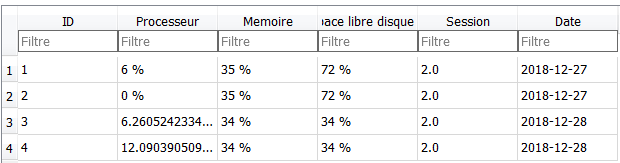
Ensuite, si l’on regarde un peu plus bas



Nous avons le contenue de chaque requête effectué le 27 et 28 déc. 18

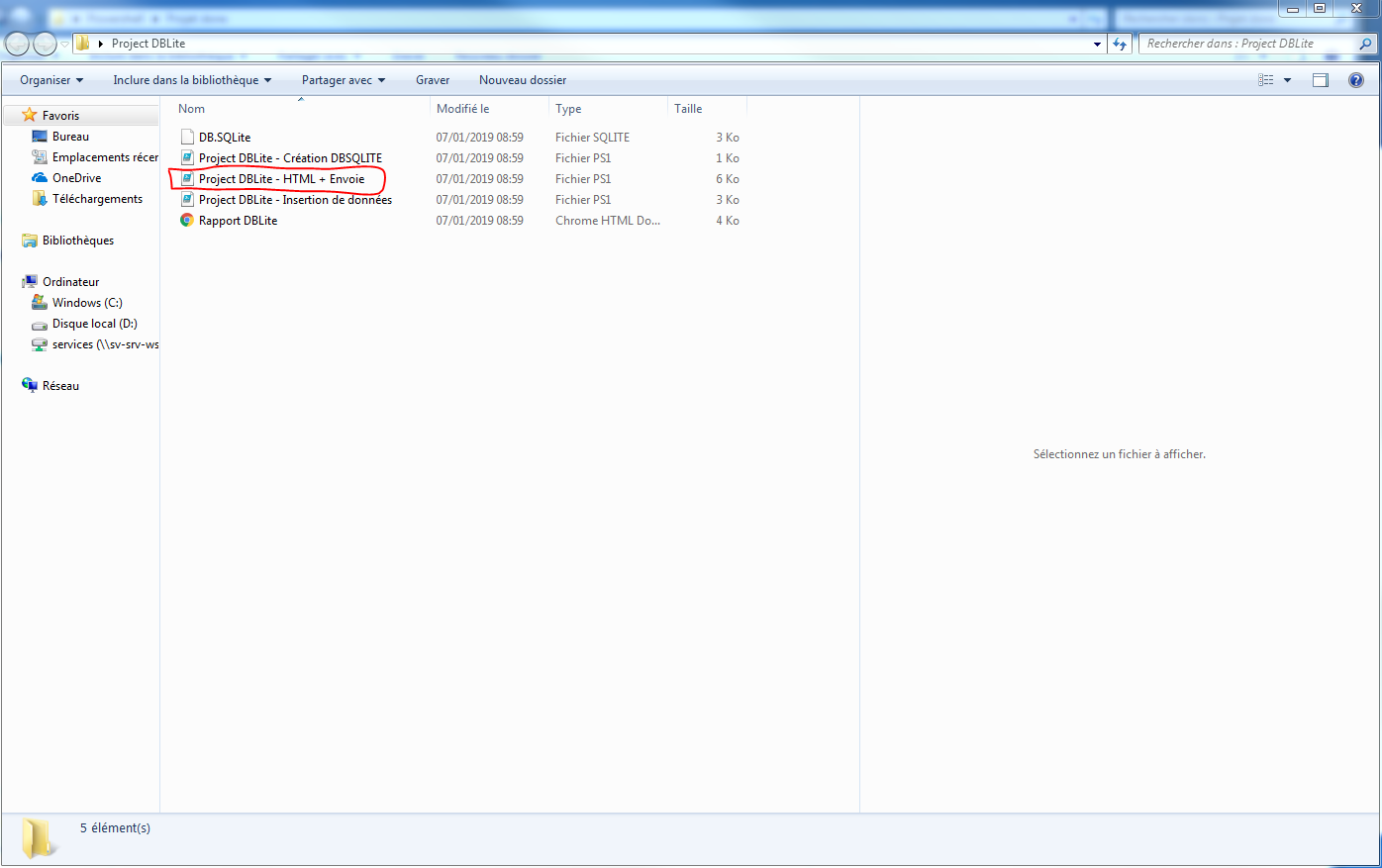
Ainsi que tous les champs : Processeur, mémoire, ID, Espace libre, Session, Date

Donc le script a écrit dans la base DBLite , si nous allons vérifier avec un « Browser » SQL on peut voir que requêtes y sont bien parvenues :



Les 4 requêtes sont bien parvenues à écrire dans la base.

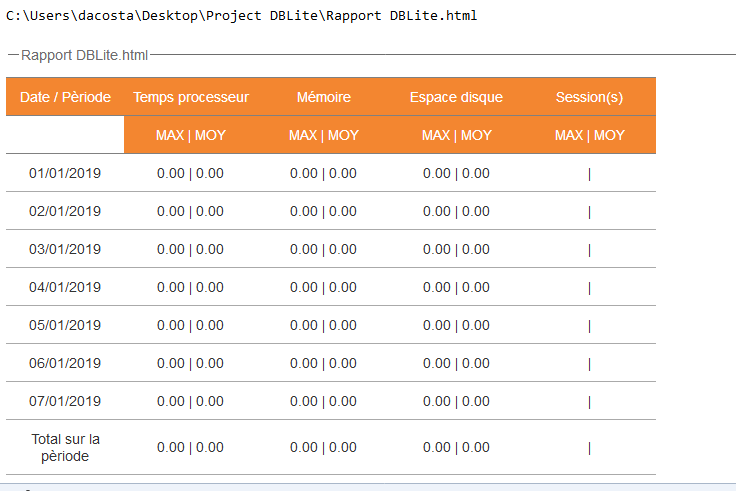
Nous pouvons donc lancer le script finale qui vas nous permettre générer le tableau HTML avec toutes nos données dans la base , et de nous envoyez finalement ce rapport par e-mail.



Double-cliquez sur le fichier pour analyser la composition de ce script

Le fonctionnement du script :

* Insérer le tableau html à l’intérieur d’une variable
* Création d’une requête de récupération SQL
* Créer une boucle qui récupère les données sur 7 jours pour les implanter directement dans notre tableau html
* Génération des données dans le tableau html



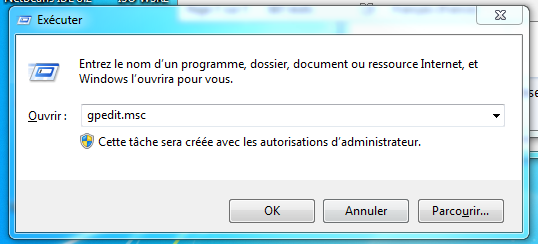
Voici le résultat de notre acquisition (Affichage de 0 car , nous n’avons collectez aucune données) automatique !

|  |
| --- |
| **Partie 4 : Planification des tâches** |

Nous pouvons maintenant faire démarrer nos scripts automatiquement grâce au planificateur de tâche Windows

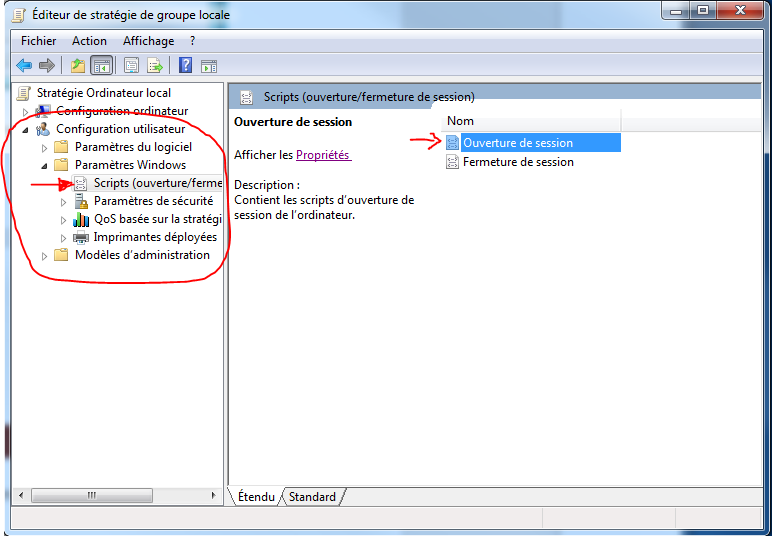
Appuyez simultanément sur les touches « Windows+R » , puis demandez l’exécution de

« gpedit.msc »

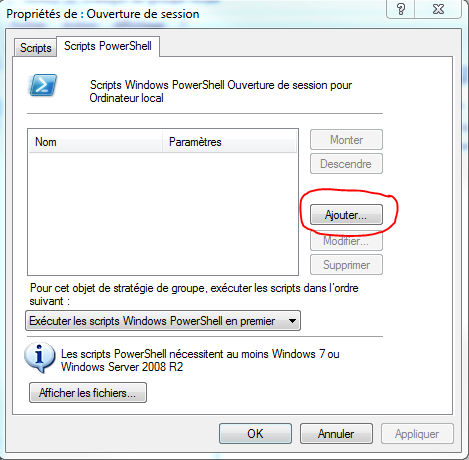


Nous voulons exécuter les scripts pour chaque connexion à une sessions particulière

Pour cela il faudra vous dirigez dans « Configuration utilisateur > Paramètres Windows > Scripts (ouverture/fermeture de session) »



Ensuite , nous devons ajouter le script que nous voulons voir se démarrer , et avec les propriétés que vous aurez choisis au préalable



|  |
| --- |
| **Partie 5 : Conclusion** |

Grâce à nos 3 scripts nous allons pouvoir générer des rapports sur les serveurs que l’on souhaite analyser et le tout automatiquement !