**Projet Lecture Log**

Sommaire

[**Introduction** 3](#_Toc190426843)

[**Problématique** 3](#_Toc190426844)

[**Solution** 3](#_Toc190426845)

[**Détails des solutions** 4](#_Toc190426846)

[Dossier de logs 4](#_Toc190426847)

[Filtrage des logs 4](#_Toc190426848)

[Requête SQL 5](#_Toc190426849)

[Base de données 6](#_Toc190426850)

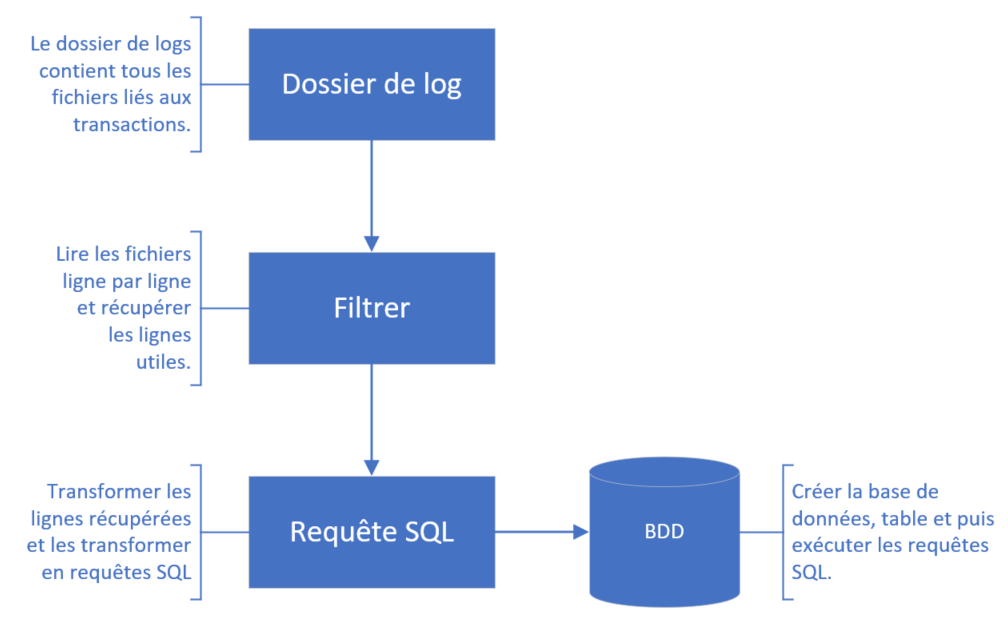
# **Introduction**

## **Problématique**

Nous devons récupérer toutes les informations liées aux transactions dans plusieurs fichiers situés dans le dossier « log » et faire en sorte de pouvoir les utiliser par la suite. Donc il faut trouver un moyen de filtrer les informations que l’on veut et de pouvoir les sauvegarder de manière lisible dans un endroit où nous pourrons les utiliser.

## **Solution**

Voici le plan pour résoudre cette problématique :



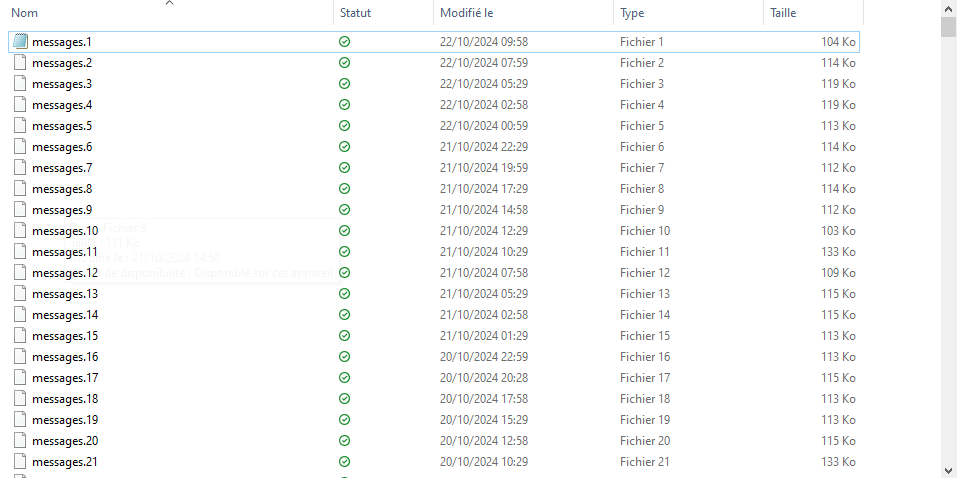
Tout d'abord, nous devons lire chaque dossier un par un, ligne par ligne, afin de ne pas surcharger la mémoire de l'ordinateur.

Ensuite, nous filtrerons chaque ligne lue pour pouvoir les conserver et créer les requêtes SQL. Une fois cela fait, nous créerons la base de données, la table, et nous exécuterons les requêtes SQL. À ce stade, nous devrions pouvoir consulter les informations dans la base de données de manière plus lisible.

# **Détails des solutions**

## Dossier de logs

Voici ce que contient le dossier « log » :

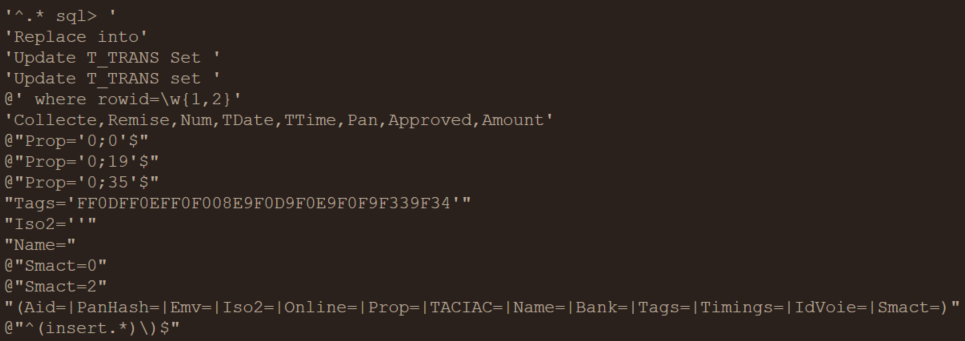


## Filtrage des logs

Pour commencer, nous devons filtrer les lignes. Il faut que les lignes :

1. Respectent l’expression régulière : '(sql> (BEGIN|Replace|Update|COMMIT))'.
2. Qu’elles ne contiennent pas : ("BEGIN TRANSACTION", "COMMIT TRANSACTION", "T\_REMISE", "MagColl")
3. Qu’elles ne contiennent pas l’expression régulière : "^Emv="
4. Qu’elles ne contiennent pas l’expression régulière : "^Collecte=\d{2} where"

Ensuite, nous pouvons commencer à transformer nos lignes. A noter que certaines lignes ci-dessous sont des expressions régulières et commence par un « @ ». Donc il faudra donc remplacer toutes ces lignes que nous avons filtrées :



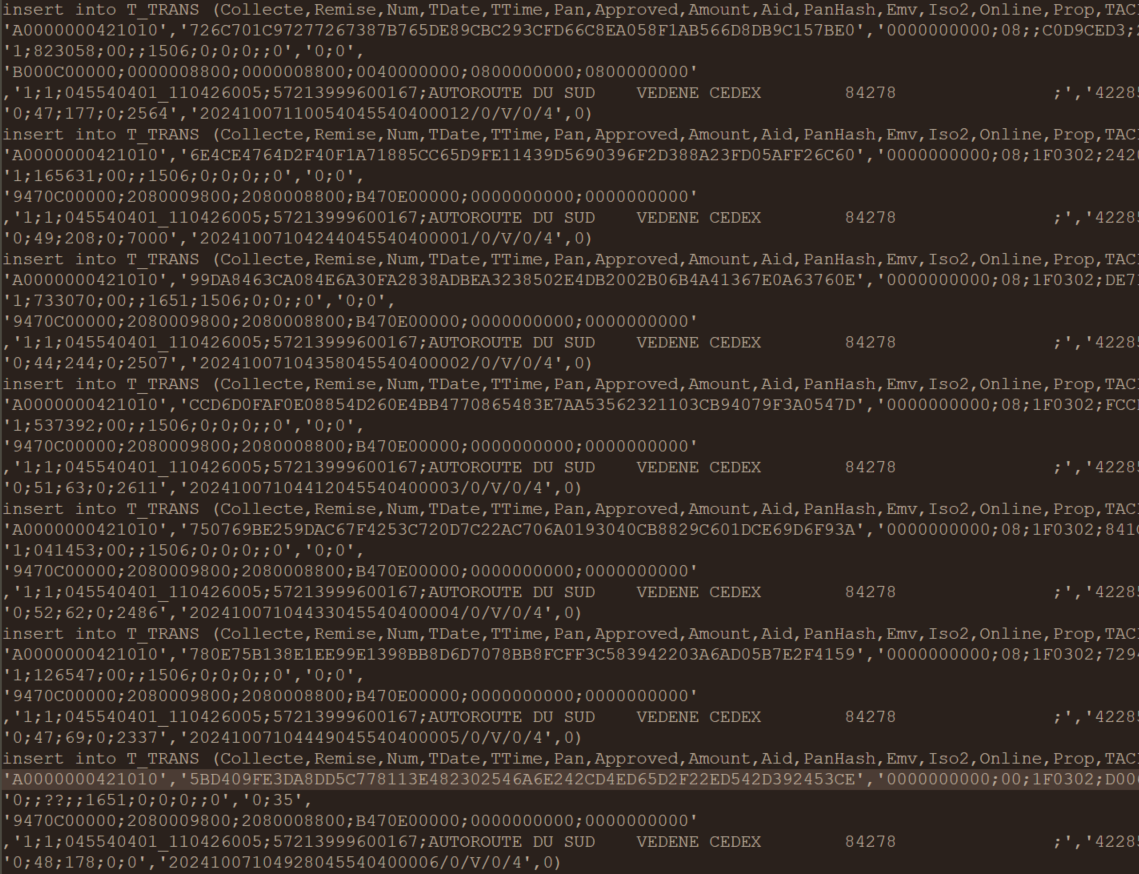
Et les remplacer par :



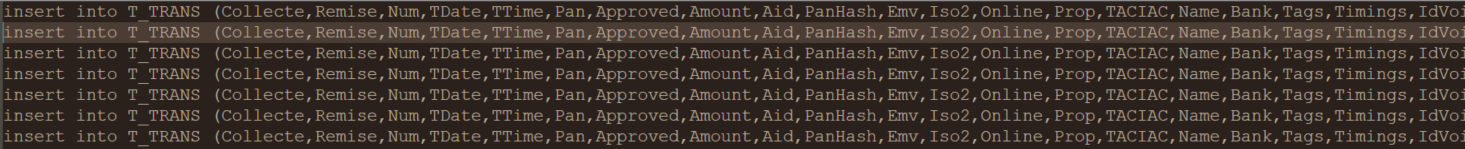
Maintenant nous avons les informations que l’on veut, il faut en faire des commandes SQL.

## Requête SQL

Maintenant que les lignes sont filtrées, il faut les convertir en requêtes SQL afin de pouvoir les exécuter.   
Les lignes que nous avons filtrées devraient ressembler à ça :



Nous avons déjà la commande SQL, mais il y a des retours à la ligne qui sont gênants. Donc tout ce que nous devons faire, c’est de retirer les retours à la ligne qu’il faut pour avoir ce résultat :



Maintenant nous avons nos requêtes SQL !

## Base de données

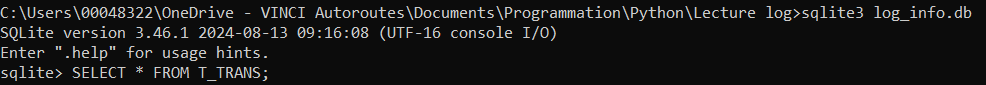
Maintenant, il faut créer la base de données, exécuter les requêtes SQL et vérifier si nous avons bien nos informations.

Pour commencer, créez une base de données (par exemple : « log\_info.db »).

Ensuite pour créer la table, voici la requête SQL :

|  |
| --- |
| CREATE TABLE T\_TRANS ( TDate CHAR(6), Remise INT, Num INT, TTime CHAR(6), Approved INT, Collecte INT, Amount INT, Aid CHAR(32), Pan CHAR(19), Iso2 CHAR(8), TacIac CHAR(66), Online CHAR(80), Emv TEXT, Prop CHAR(20), PanHash CHAR(64), Name CHAR(48), Bank CHAR(64), Tags CHAR(40), IdVoie CHAR(40), Smact INT, Timings CHAR(30), PRIMARY KEY (Remise, Num) ); |

Il ne reste plus qu'à exécuter toutes les requêtes SQL que nous avons transformées.

Maintenant, nous voulons vérifier que les informations sont bien dans la base de données. Pour ce faire, ouvrez une invite de commande, allez dans le répertoire où se situe la base de données et exécutez ces commandes :  


Après avoir exécuté la dernière commande, vous devriez voir toutes les informations comme ci-dessous :

Voici ce que doit faire votre programme.