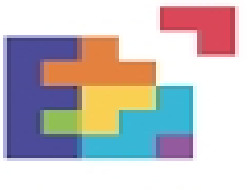
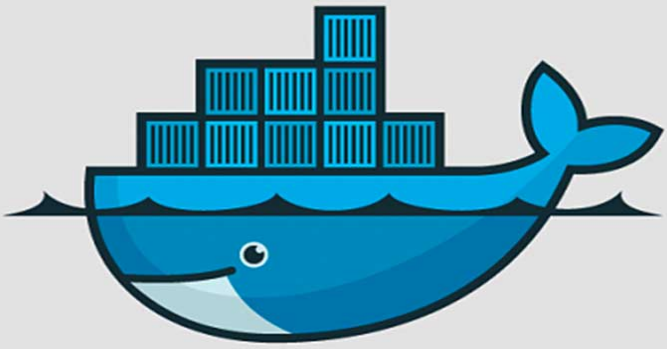
Documentation d'installation

Tp Qualité des donées



epsi

l'école d'ingénierie informatique



docker

Ahmed BEN MORRI MORRI, KOUKOUTHA ARDELKALEB BARANDAO Itébéma

Sommaire

Pré-requis

Nous devons avoir come pré-requis:

- Docker
- Python
- Fin du TP 2 de Architecture Décisionnel (Travail déjà effectuer)

Nous sommes sur Windows

Introduction : Importation de données

Dans ce projet nous sommes
dans le cas 1 car :

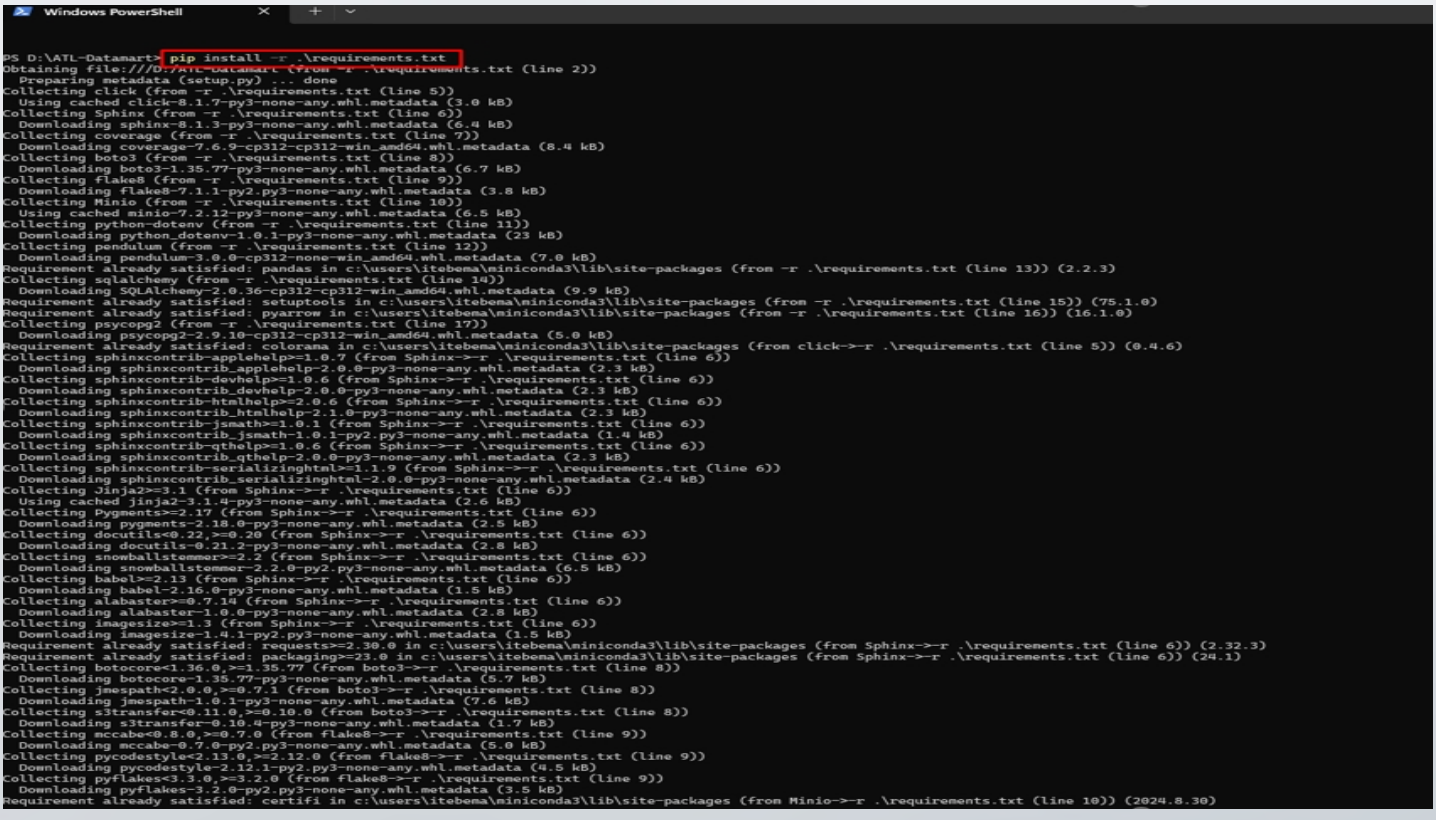
On a déjà fait l'Architecture Décisionnel, du coup on va réutiliser :

- La base de données sur les taxis de New-York pour continuer la suite du TP.
- Nous utiliserons les données brutes, sans formattage.

Installations des Dépendances

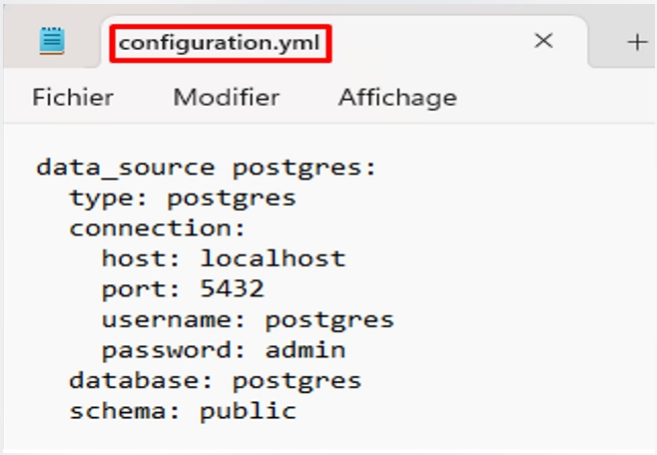
Il faut donc taper la commande suiavntes nécessaires pour le TP:

pip install -r requirement.txt



Exercice 1

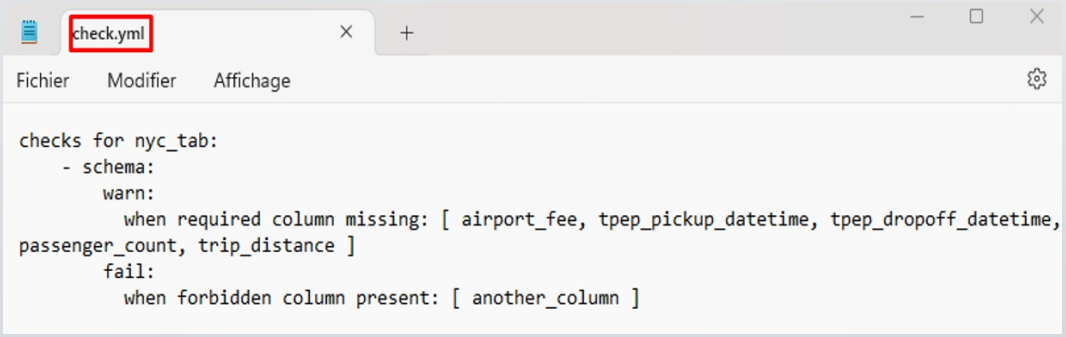
Création du fichier « configuration.yml »



Vérification de la configuration de soda avec la commande :
#soda test-connection -d postgres -c configuration.yml

```
PS D:\ATL-Datamart>
PS D:\ATL-Datamart> soda test-connection -d postgres -c configuration.yml
[16:25:49] Soda Core 3.4.1
Successfully connected to 'postgres'.
Connection 'postgres' is valid.
PS D:\ATL-Datamart>
```

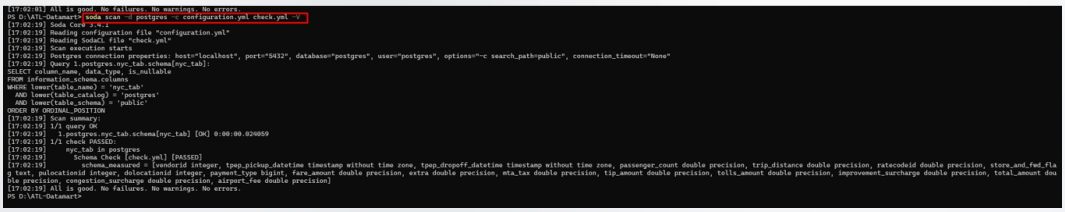
Création du fichier « check.yml »



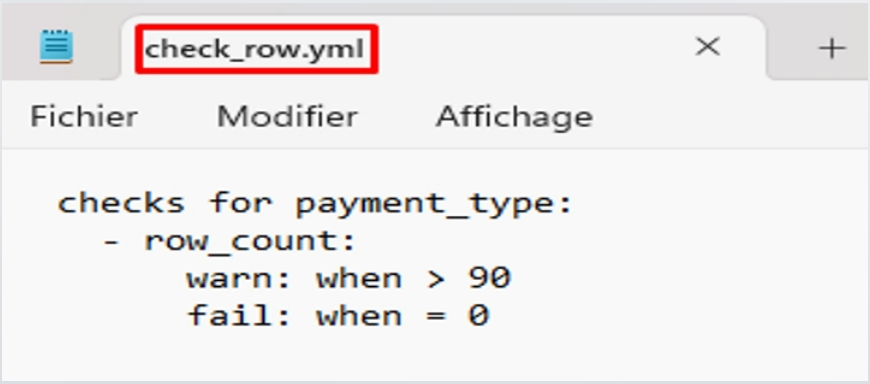
Test de la commande :
#soda scan -d postgres -c configuration.yml check.yml

```
PS D:\ATL-Datamart>
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml check.yml
[17:02:01] Soda Core 3.4.1
[17:02:01] Scan summary:
[17:02:01] 1/1 check PASSED:
[17:02:01]     nyc_tab in postgres
[17:02:01]       Schema Check [PASSED]
[17:02:01] All is good. No failures. No warnings. No errors.
```

Maintenant, ajoutez le -V en fin de commande, on obtient ceci :



Nombre de ligne supérieur à 0 avec row_count dans le fichier « check_row.yml » :




```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml check_row.yml
[13:10:31] Soda Core 3.4.1
[13:10:32] Scan summary:
[13:10:32] 1/1 check PASSED:
[13:10:32]     payment_type in postgres
[13:10:32]       row_count warn when > 90 fail when = 0 [PASSED]
[13:10:32] All is good. No failures. No warnings. No errors.
```

Exercices 2 & 3

Check1 : Vérifier que les tables « nyc_tab », « location » et « vendors » ne sont pas vide.

```
PS D:\ATL-Datamart>
PS D:\ATL-Datamart>  soda scan -d postgres -c configuration.yml contenance.yml
[12:44:42] Soda Core 3.4.1
[12:44:46] Scan summary:
[12:44:46] 3/3 checks PASSED:
[12:44:46]     nyc_tab in postgres
[12:44:46]       row_count > 0 [PASSED]
[12:44:46]     location in postgres
[12:44:46]       row_count > 0 [PASSED]
[12:44:46]     vendors in postgres
[12:44:46]       row_count > 0 [PASSED]
[12:44:46] All is good. No failures. No warnings. No errors.
PS D:\ATL-Datamart>
```

Check2 : Vérifier que la colonne « vendorid » de la table « nyc_tab » n'ait pas de valeur nulle.

```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml vendorval.yml
[14:14:44] Soda Core 3.4.1
[14:14:48] Scan summary:
[14:14:48] 1/1 check PASSED:
[14:14:48]     nyc_tab in postgres
[14:14:48]       missing_count(vendorid) = 0 [PASSED]
[14:14:48] All is good. No failures. No warnings. No errors.
```

Check3 : Vérifier que les colonnes « tpep_pickup_datetime » et « tpep_dropoff_datetime » ont des valeurs valides à hauteur de 99%

```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml dateval.yml
[14:40:09] Soda Core 3.4.1
[14:40:09] Counting invalid without valid or invalid specification does not make sense. ("invalid_percent(tpep_dropoff_datetime) < 1%" @ line=2,col=5 in dateval.yml)
[14:40:09] Counting invalid without valid or invalid specification does not make sense. ("invalid_percent(tpep_pickup_datetime) < 1%" @ line=3,col=5 in dateval.yml)
[14:40:12] Scan summary:
[14:40:12] 2/2 checks PASSED:
[14:40:12]     nyc_tab in postgres
[14:40:12]       invalid_percent(tpep_dropoff_datetime) < 1% [PASSED]
[14:40:12]       invalid_percent(tpep_pickup_datetime) < 1% [PASSED]
[14:40:12] All is good. No failures. No warnings. No errors.
```

Check4 : Vérifier que les valeurs dans la colonne « trip_distance » de la table « nyc_tab » sont toutes supérieurs à 0

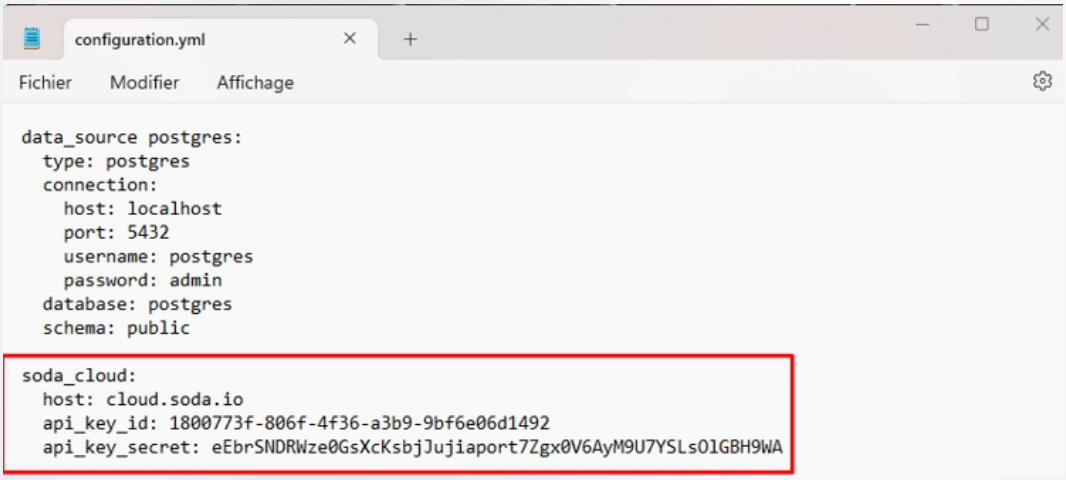
```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml distance.yml
[14:50:41] Soda Core 3.4.1
[14:50:47] Scan summary:
[14:50:47] 1/1 check FAILED:
[14:50:47]     nyc_tab in postgres
[14:50:47]       min(trip_distance) > 0 [FAILED]
[14:50:47]         check_value: 0.0
[14:50:47] Oops! 1 failures. 0 warnings. 0 errors. 0 pass.
```

Check5 : Vérifier que les valeurs dans la colonne « trip_distance » de la table « nyc_tab_organised » sont supérieurs à 0 et inférieurs à 100

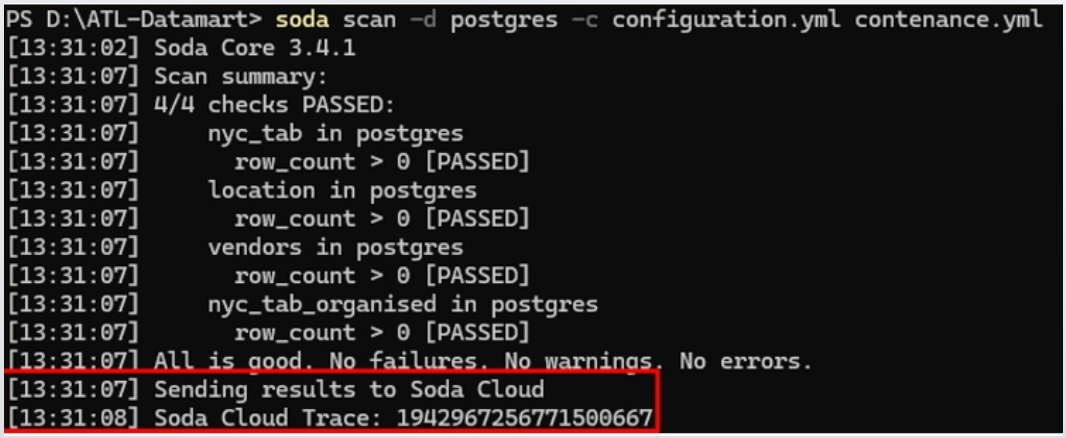
```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml distance.yml
[15:24:23] Soda Core 3.4.1
[15:24:31] Scan summary:
[15:24:31] 2/2 checks FAILED:
[15:24:31]     nyc_tab_organised in postgres
[15:24:31]       min(trip_distance) > 0 [FAILED]
[15:24:31]         check_value: 0.0
[15:24:31]       max(trip_distance) < 100 [FAILED]
[15:24:31]         check_value: 330397.59
[15:24:31] Oops! 2 failures. 0 warnings. 0 errors. 0 pass.
```

Exercice 4

Nous avons choisi « soda cloud » pour réaliser le Dashboard. Pour se faire il faut créer son compte sur soda cloud et ensuite compléter notre fichier de configuration « configuration.yml » avec quelques lignes pour permettre à soda d'envoyer les résultats des checks sur soda cloud pour la génération automatique d'un dashboard de qualité des données

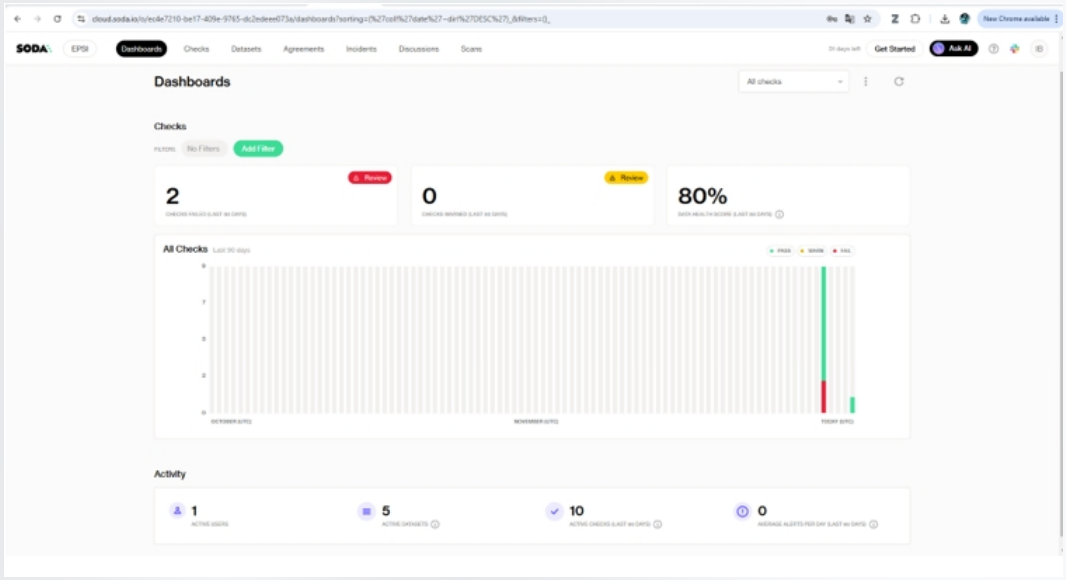


A l'exécution de chaque check, on aura un message supplémentaire qui nous confirmera l'envoie des résultats du check vers soda cloud.



Exercice 4

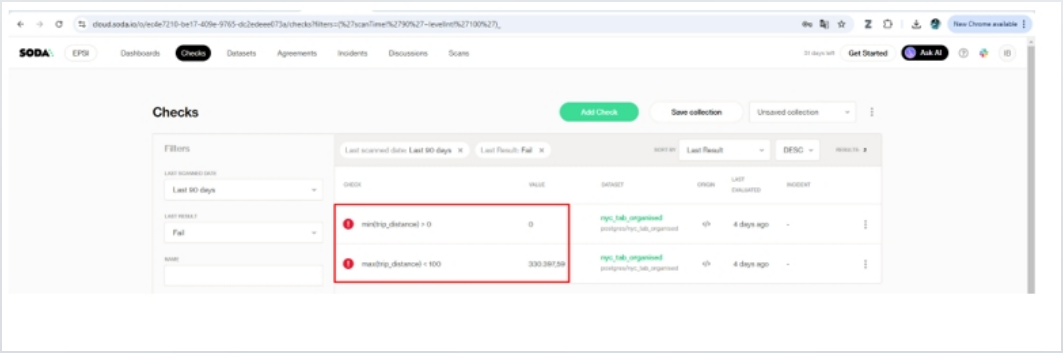
Les Dashboard :



2 checks échoués : ce qui veut dire que les données dans nos bases de données ne respectent pas ces deux règles. Qui sont :

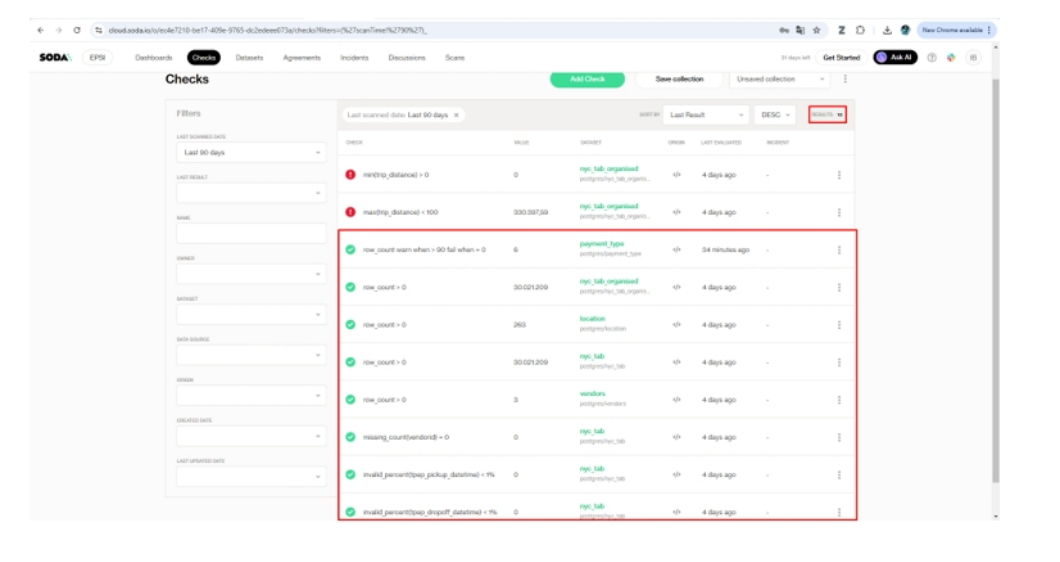
• Les valeurs dans la colonne « trip_distance » ne doivent jamais être inferieurs ni égales à 0. Mais ici dans le retour du check, on voit qu'il y'a bien une valeur qui est égale à 0. D'où l'échec de la règle

• Les valeurs dans la colonne « trip_distance » ne doivent jamais être supérieurs à 100. Mais ici dans le retour du check, on voit qu'il y'a bien une valeur(330.397,59) qui est supérieur à 100. D'où l'échec de la règle.

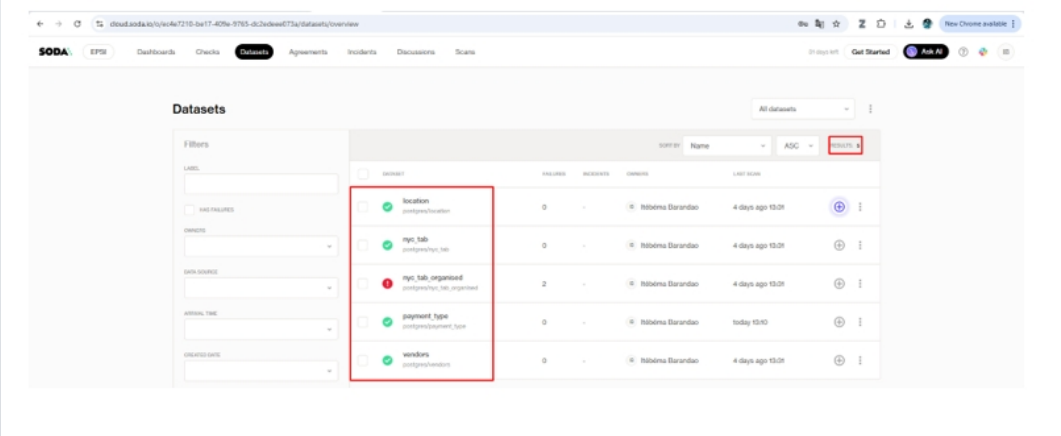


Exercice 4

10 checks ont été réalisés au total et les données des bases de données sont conformes à 8 d'entre elles.



Les checks ont été réalisés sur plusieurs datasets :



Fin du TP !

TP réalisé par :

Ahmed BEN MORRI MORRI,
KOUKOUTHA ARDEL KALEB
BARANDAO Itébéma

Machine utilisé pour ce TP :

Asus Tuf f15
16 Ram
1 téra
Rtx 3060
Intel Core i7