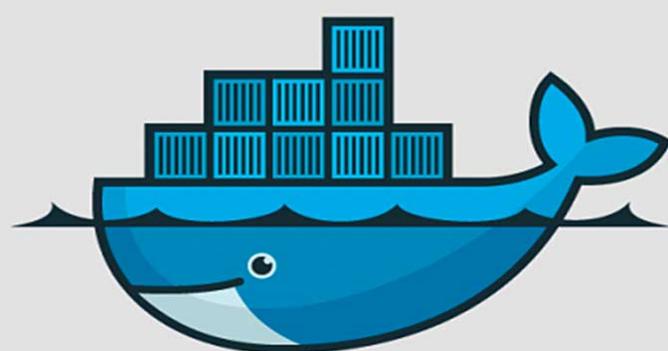


Tp Qualité des données



**l'école d'ingénierie
informatique**



docker

Sommaire

| | | |
|----|-------------------------------------|----|
| 1. | Prérequis | 3 |
| 2. | Introduction..... | 4 |
| 3. | Installations des dépendances | 4 |
| 4. | Exercice 1 | 5 |
| 5. | Exercice 2..... | 7 |
| 6. | Exercice 3 | 7 |
| 7. | Exercice 4..... | 10 |

Pré-requis

Nous devons avoir comme 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 :

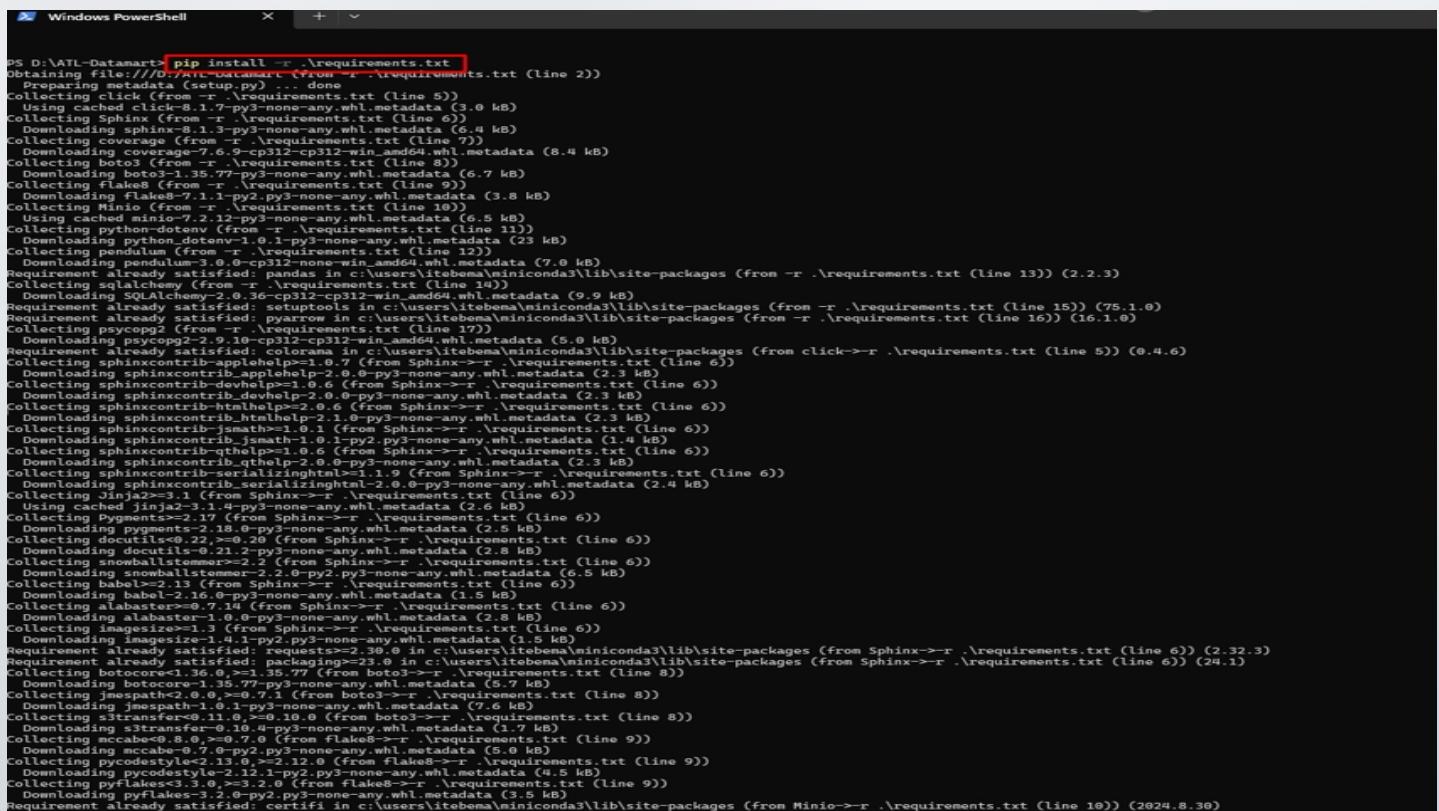
On a déjà fait l'Architecture Décisionnel, du coup on va réutiliser :

- 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 suivante nécessaires pour le TP:

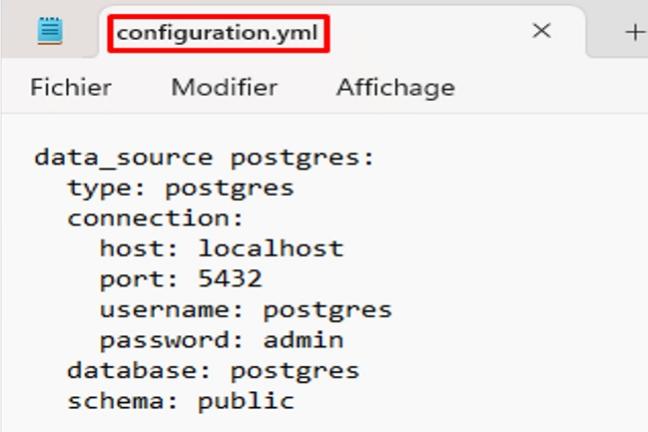
```
# pip install -r requirement.txt
```



```
D:\ATL-Datamart> pip install -r .\requirements.txt
Obtaining file:///D:/ATL-Datamart (from -r .\requirements.txt (line 2))
Preparing metadata (setup.py) ... done
Collecting click (from -r .\requirements.txt (line 5))
  Using cached click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting Sphinx (from -r .\requirements.txt (line 6))
  Downloading sphinx-8.1.3-py3-none-any.whl.metadata (6.4 kB)
Collecting coverage (from -r .\requirements.txt (line 7))
  Downloading coverage-7.6.1-py3-none-any.whl.metadata (8.4 kB)
Collecting boto3 (from -r .\requirements.txt (line 8))
  Downloading boto3-1.35.77-py3-none-any.whl.metadata (6.7 kB)
Collecting flake8 (from -r .\requirements.txt (line 9))
  Downloading flake8-7.1.1-py3-py3-none-any.whl.metadata (3.8 kB)
Collecting minio (from -r .\requirements.txt (line 10))
  Using cached minio-7.2.12-py3-none-any.whl.metadata (6.5 kB)
Collecting python-dotenv (from -r .\requirements.txt (line 11))
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Collecting pendulum (from -r .\requirements.txt (line 12))
  Downloading pendulum-2.0.0-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: pandas in c:\users\itebema\miniconda3\lib\site-packages (from -r .\requirements.txt (line 13)) (2.2.3)
Collecting sqlalchemy (from -r .\requirements.txt (line 14))
  Downloading SQLAlchemy-2.0.36-cp312-cp312-win_amd64.whl.metadata (9.9 kB)
Requirement already satisfied: sqluproot in c:\users\itebema\miniconda3\lib\site-packages (from -r .\requirements.txt (line 15)) (75.1.0)
Requirement already satisfied: alembic in c:\users\itebema\miniconda3\lib\site-packages (from -r .\requirements.txt (line 16)) (16.1.0)
Collecting psycopg2 (from -r .\requirements.txt (line 17))
  Downloading psycopg2-2.9.10-cp312-cp312-win_amd64.whl.metadata (5.0 kB)
Requirement already satisfied: colorama in c:\users\itebema\miniconda3\lib\site-packages (from click->r .\requirements.txt (line 5)) (0.4.6)
Collecting sphinxcontrib-apelhelp (from Sphinx->r .\requirements.txt (line 6))
  Downloading sphinxcontrib_apelhelp-0.0.1-py3-none-any.whl.metadata (2.3 kB)
Collecting sphinxcontrib-devhelp>=1.0.6 (from Sphinx->r .\requirements.txt (line 6))
  Downloading sphinxcontrib_devhelp-2.0.0-py3-none-any.whl.metadata (2.3 kB)
Collecting sphinxcontrib-htmlhelp>=2.0.6 (from Sphinx->r .\requirements.txt (line 6))
  Downloading sphinxcontrib_htmlhelp-2.0.6-py3-none-any.whl.metadata (2.3 kB)
Collecting sphinxcontrib-jsmath>=1.0.1 (from Sphinx->r .\requirements.txt (line 6))
  Downloading sphinxcontrib_jsmath-1.0.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting sphinxcontrib-qthelp>=1.0.6 (from Sphinx->r .\requirements.txt (line 6))
  Downloading sphinxcontrib_qthelp-2.0.0-py3-none-any.whl.metadata (2.3 kB)
Collecting sphinxcontrib-serializinghtml (from Sphinx->r .\requirements.txt (line 6))
  Downloading sphinxcontrib_serializinghtml-2.0.0-py3-none-any.whl.metadata (2.4 kB)
Collecting Jinja2>=2.3.1 (from Sphinx->r .\requirements.txt (line 6))
  Using cached jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting Payments>=2.1.0 (from Sphinx->r .\requirements.txt (line 6))
  Downloading payments-2.1.0-py3-none-any.whl.metadata (2.6 kB)
Collecting docutils<0.22,>=0.20 (from Sphinx->r .\requirements.txt (line 6))
  Downloading docutils-0.21.2-py3-none-any.whl.metadata (2.8 kB)
Collecting snowballstemmer>=2.1. (from Sphinx->r .\requirements.txt (line 6))
  Downloading snowballstemmer-2.1.0-py2.py3-none-any.whl.metadata (6.5 kB)
Collecting babel<3.0.0,>=2.13 (from Sphinx->r .\requirements.txt (line 6))
  Downloading babel-2.16.0-py3-none-any.whl.metadata (1.5 kB)
Collecting alabaster=>0.7.14 (from Sphinx->r .\requirements.txt (line 6))
  Downloading alabaster-1.0.0-py3-none-any.whl.metadata (2.8 kB)
Collecting packaging>=23.0 (from botocore->r .\requirements.txt (line 6))
  Downloading packaging-23.0-py3-none-any.whl.metadata (8.5 kB)
Collecting botocore<1.36.0,>=1.35.77 (from botocore->r .\requirements.txt (line 8))
  Downloading botocore-1.36.0-py3-none-any.whl.metadata (1.5 kB)
Collecting jmespath<0.0.0,>=0.7.1 (from botocore->r .\requirements.txt (line 8))
  Downloading jmespath-0.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from botocore->r .\requirements.txt (line 8))
  Downloading s3transfer-0.10.0-py3-none-any.whl.metadata (1.7 kB)
Collecting s3transfert<0.0.0,>=0.7.1 (from botocore->r .\requirements.txt (line 9))
  Downloading s3transfert-0.0.0-py3-none-any.whl.metadata (5.9 kB)
Collecting mcache<0.7.0,>=0.2.0 (from flask->r .\requirements.txt (line 9))
  Downloading mcache-0.7.0-py2.py3-none-any.whl.metadata (5.9 kB)
Collecting pycodestyle<2.13.0,>=2.12.0 (from flask8->r .\requirements.txt (line 9))
  Downloading pycodestyle-2.12.1-py2.py3-none-any.whl.metadata (4.5 kB)
Collecting pyflakes<3.0.0,>=2.0.0 (from flask8->r .\requirements.txt (line 9))
  Downloading pyflakes-2.0.0-py2.py3-none-any.whl.metadata (3.5 kB)
Requirement already satisfied: certifi in c:\users\itebema\miniconda3\lib\site-packages (from Minio->r .\requirements.txt (line 10)) (2024.8.30)
```

Exercice 1

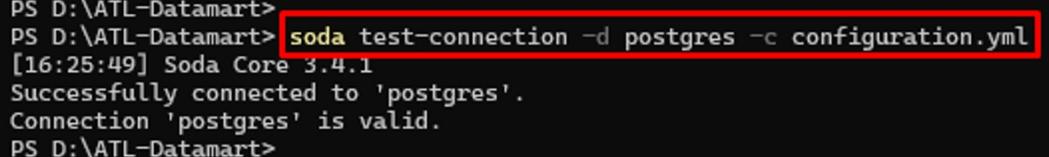
Création du fichier « configuration.yml »



```
data_source postgres:
  type: postgres
  connection:
    host: localhost
    port: 5432
    username: postgres
    password: admin
  database: postgres
  schema: public
```

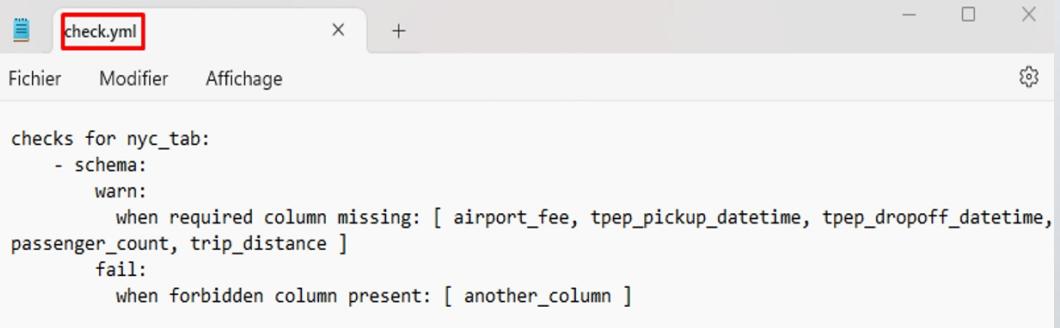
Vérification de la configuration de soda avec la commande :

```
#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 »



```
checks for nyc_tab:
  - schema:
    warn:
      when required column missing: [ airport_fee, tpep_pickup_datetime, tpep_dropoff_datetime,
passenger_count, trip_distance ]
    fail:
      when forbidden column present: [ another_column ]
```

Suite Exercices

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 :

```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml -V
[17:02:19] Soda Core 3.4.1
[17:02:19] Reading configuration file "configuration.yml"
[17:02:19] Reading SodAL file "check.yml"
[17:02:19] PostgreSQL connection properties: host="localhost", port="5432", database="postgres", user="postgres", options="-c search_path=public", connection_timeout="None"
[17:02:19] PostgreSQL connection properties: host="localhost", port="5432", database="postgres", user="postgres", options="-c search_path=public", connection_timeout="None"
[17:02:19] PostgreSQL connection properties: host="localhost", port="5432", database="postgres", user="postgres", options="-c search_path=public", connection_timeout="None"
SELECT column_name, data_type, is_nullable
FROM INFORMATION_SCHEMA.COLUMNS
WHERE lower(table_name) = 'nyc_tab'
AND lower(table_catalog) = 'postgres'
AND lower(column_name) = 'payment_type'
ORDER BY ORDINAL_POSITION
[17:02:19] 1/1 query OK
[17:02:19] 1/1 schema check OK
[17:02:19] 1/1 check PASSED:
[17:02:19]     Schema Check [PASSED]
[17:02:19] All is good. No failures. No warnings. No errors.
PS D:\ATL-Datamart>
```

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



```
checks for payment_type:
- row_count:
  warn: when > 90
  fail: when = 0
```

```
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
[14:08:09] Soda Core 3.4.1
[14:08:09] Counting invalid without valid or invalid specification does not make sense. ("invalid_percent(tpep_dropoff_datetime) < 1%" @ line=2,col=5 in dateval.yml)
[14:08:09] Counting invalid without valid or invalid specification does not make sense. ("invalid_percent(tpep_pickup_datetime) < 1%" @ line=3,col=5 in dateval.yml)
[14:08:12] Scan summary:
[14:08:12] 2/2 checks PASSED:
[14:08:12]     nyc_tab in postgres
[14:08:12]         invalid_percent(tpep_dropoff_datetime) < 1% [PASSED]
[14:08:12]         invalid_percent(tpep_pickup_datetime) < 1% [PASSED]
[14:08:12] 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.
```

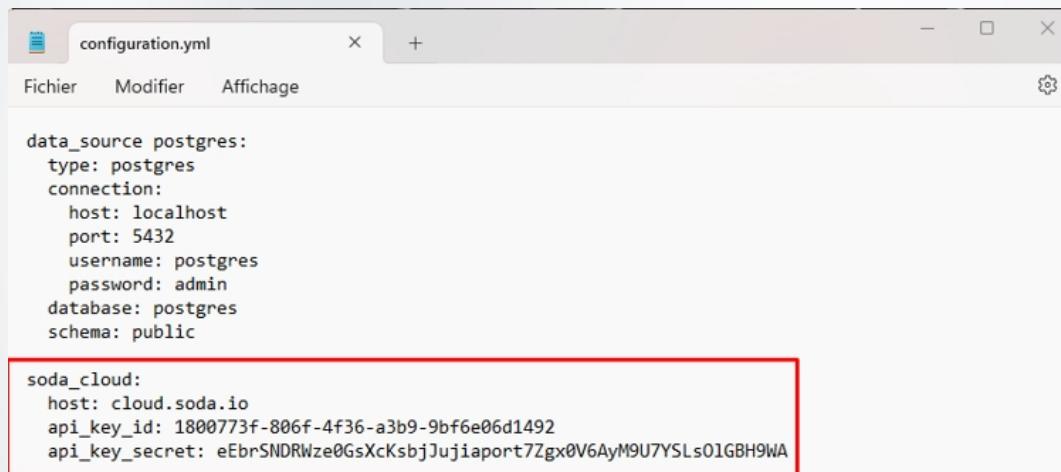
Suite Exercices

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



```
data_source postgres:
  type: postgres
  connection:
    host: localhost
    port: 5432
    username: postgres
    password: admin
    database: postgres
    schema: public

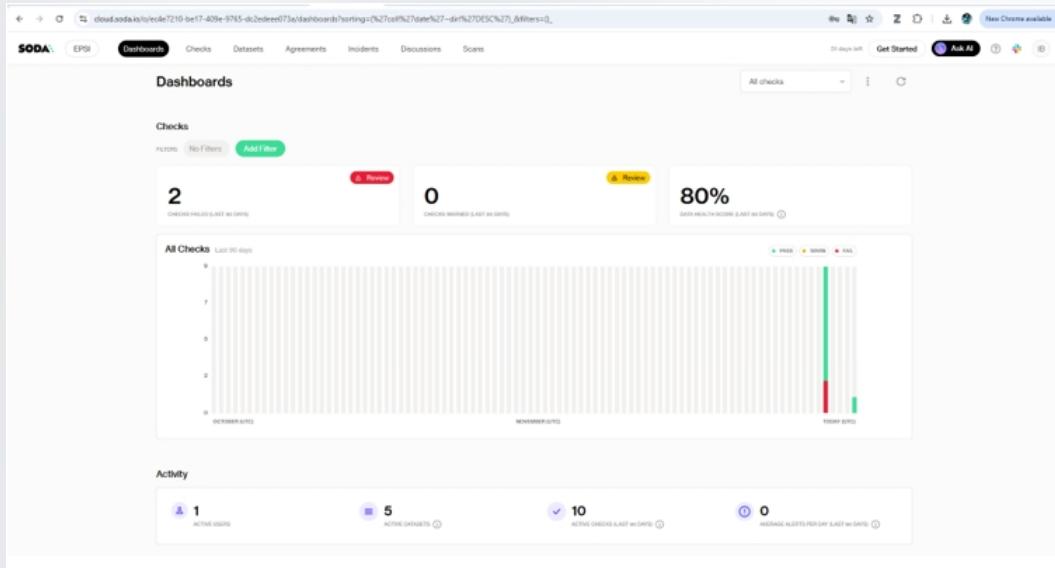
soda_cloud:
  host: cloud.soda.io
  api_key_id: 1800773f-806f-4f36-a3b9-9bf6e06d1492
  api_key_secret: eEbrSNDRWze0GsXcKsbjJujiaport7Zgx0V6AyM9U7YSLs01GBH9WA
```

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

```
PS D:\ATL-Datamart> soda scan -d postgres -c configuration.yml contenance.yml
[13:31:02] Soda Core 3.4.1
[13:31:07] Scan summary:
[13:31:07] 4/4 checks PASSED:
[13:31:07]   nyc_tab in postgres
[13:31:07]     row_count > 0 [PASSED]
[13:31:07]   location in postgres
[13:31:07]     row_count > 0 [PASSED]
[13:31:07]   vendors in postgres
[13:31:07]     row_count > 0 [PASSED]
[13:31:07]   nyc_tab_organised in postgres
[13:31:07]     row_count > 0 [PASSED]
[13:31:07] All is good. No failures. No warnings. No errors.
[13:31:07] Sending results to Soda Cloud
[13:31:08] Soda Cloud Trace: 1942967256771500667
```

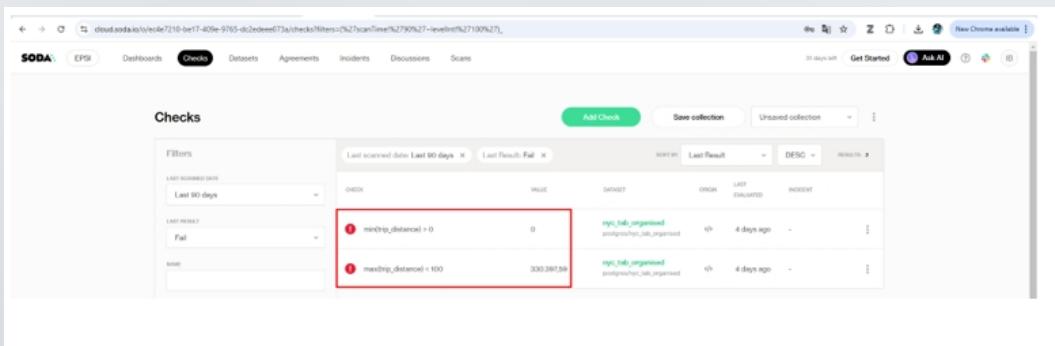
Exercice 4

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 inférieurs 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.

The screenshot shows the SODA Checks interface. On the left, there are filters for LAST SUBMITTED DATE, LAST RESULT, NAME, OWNER, DATASET, DATA SOURCE, OWNER, and CREATED DATE. The main area displays 10 check results, each with a status icon (green for success, red for failure), the check name, value, dataset, origin, last evaluated time, and incident details. The results are sorted by DESC.

| Check | Value | Dataset | Origin | Last Evaluated | Incident |
|--|------------|--------------------|-----------------------------|----------------|----------|
| max(bq_distance) > 0 | 0 | nyc_taxi_organized | postgres/nyc_taxi_organized | 4 days ago | - |
| max(bq_distance) < 100 | 300,987,59 | nyc_taxi_organized | postgres/nyc_taxi_organized | 4 days ago | - |
| now_count(warn > 90 fail when > 0) | 0 | payment_type | postgres/payment_type | 34 minutes ago | - |
| now_count(> 0) | 30,021,209 | nyc_taxi_organized | postgres/nyc_taxi_organized | 4 days ago | - |
| now_count(> 0) | 260 | location | postgres/location | 4 days ago | - |
| now_count(> 0) | 30,021,209 | nyc_taxi | postgres/nyc_taxi | 4 days ago | - |
| now_count(> 0) | 3 | vendors | postgres/vendors | 4 days ago | - |
| missing_count(bondedB > 0) | 0 | nyc_taxi | postgres/nyc_taxi | 4 days ago | - |
| invalid_parcels(prep_pickup_datetime) < % | 0 | nyc_taxi | postgres/nyc_taxi | 4 days ago | - |
| invalid_parcels(prep_dropoff_datetime) < % | 0 | nyc_taxi | postgres/nyc_taxi | 4 days ago | - |

Les checks ont été réalisés sur plusieurs datasets :

The screenshot shows the SODA Datasets interface. On the left, there are filters for LAST, FAILURES, OWNER, DATA SOURCE, ARRIVAL TIME, and CREATED DATE. The main area displays a list of datasets, each with a status icon (green for success, red for failure), name, failures, incidents, owner, and last score. The results are sorted by ASC.

| Name | Failures | Incidents | Owner | Last Score |
|--------------------|----------|-----------|------------------|------------------|
| location | 0 | - | Heleena Barandao | 4 days ago 13:01 |
| nyc_taxi | 0 | - | Heleena Barandao | 4 days ago 13:01 |
| nyc_taxi_organized | 2 | - | Heleena Barandao | 4 days ago 13:01 |
| payment_type | 0 | - | Heleena Barandao | today 13:10 |
| vendors | 0 | - | Heleena Barandao | 4 days ago 13:01 |

Fin du TP !

TP réalisé par :

Ahmed BEN MORRI MORRI,
KOUKOUTH A ARDEL KALEB
BARANDAO Itébéma

Machine utilisé pour ce TP :

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

