

Réalisé par :

BARANDAO Itébéma, BEN MORRI Ahmed, KKOUKOUTH A ARDEL KALEB

## 1. Introduction

- **Contexte :**

Ce projet a été réalisé dans le cadre du cours d'atelier Architecture Décisionnelle Datamart (TRDE704). Il vise à développer et automatiser un pipeline de données, en passant par des étapes de traitement, stockage, et visualisation des données.

- **Objectifs :**

- Comprendre et mettre en œuvre les principes d'architecture décisionnelle.
- Gérer l'infrastructure à l'aide de Docker et de fichiers de configuration.
- Automatiser le traitement des données et leur restitution via des outils de visualisation.

## 2. Méthodologie

- **Infrastructure :**

L'infrastructure a été gérée à l'aide de Docker, avec les commandes suivantes :

- Lancer l'infrastructure : `docker compose up`
- Arrêter l'infrastructure : `docker compose down`

- **Approche par étapes :**

- **TP1** : Complétion des fonctions dans `grab_parquet.py`.
- **TP2** : Adaptation du code pour récupérer des fichiers depuis Minio et les injecter dans PostgreSQL.
- **TP3** : Création de tables en modèle flocon avec des scripts SQL.
- **TP4** : Restitution des données via l'outil de dataviz PowerBI.
- **TP5** : Automatisation des tâches avec Airflow à travers un dag.

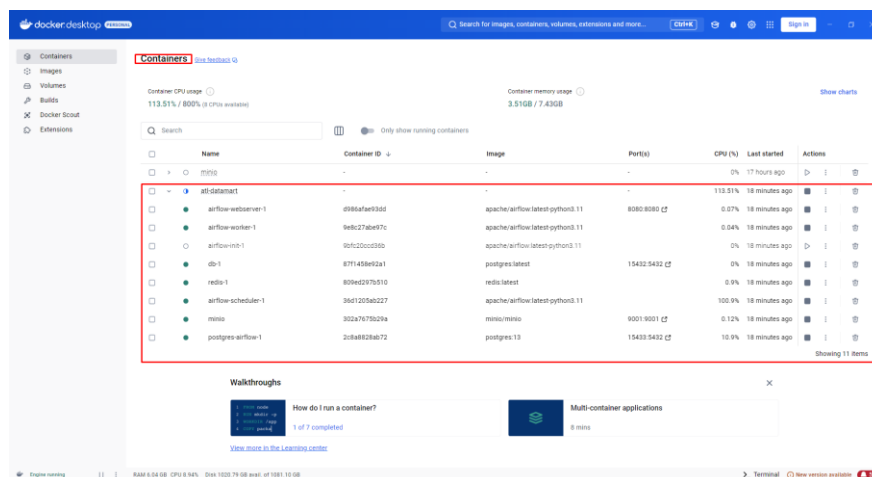
## 3. Résultats et analyses

# TP1 : Extraction des données

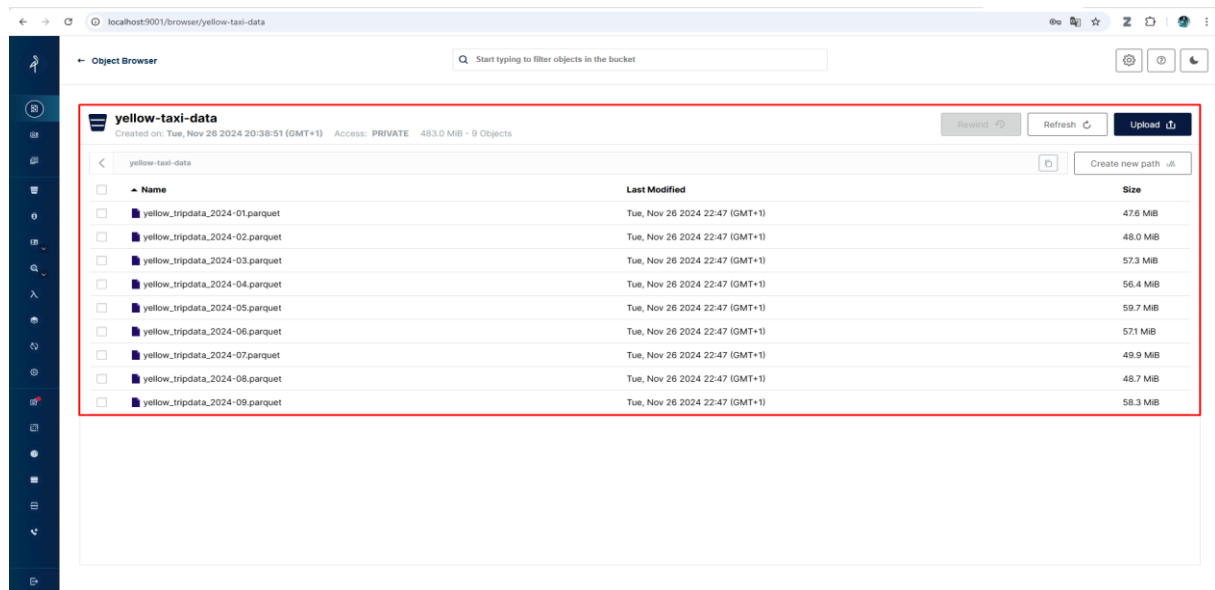
Modifications apportées au script `grab_parquet.py` (`src/data/grab_parquet.py`) pour récupérer les données (fichiers `.parquet`) depuis le site [www.nyc.gov](http://www.nyc.gov) vers minio.

## Initialisation de Docker, de Minio, d'Airflow

```
Windows PowerShell
PS D:\All-Datamart> docker compose up
time="2024-12-10T13:05:22+0100" level=warning msg="D:\\All-Datamart\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 9/0
  ✓ Container atl-datamart-db-1      Running      0.0s
  ✓ Container minio                  Created      0.0s
  ✓ Container atl-datamart-redis-1    Running      0.0s
  ✓ Container atl-datamart-postgres-airflow-1 Running      0.0s
  ✓ Container atl-datamart-airflow-init-1 Created      0.0s
  ✓ Container atl-datamart-airflow-webserver-1 Running      0.0s
  ✓ Container atl-datamart-airflow-scheduler-1 Running      0.0s
  ✓ Container atl-datamart-airflow-worker-1 Running      0.0s
  ✓ Container atl-datamart-airflow-triggerer-1 Running      0.0s
Attaching to airflow-init-1, airflow-scheduler-1, airflow-triggerer-1, airflow-webserver-1, airflow-worker-1, db-1, postgres-airflow-1, redis-1, minio
minio
minio  Main Object Storage Server
minio  Copyright: 2015-2024 MinIO, Inc.
minio  License: GNU AGPLv3 = https://www.gnu.org/licenses/agpl-3.0.html
minio  Version: RELEASE.2024-11-0708-52-202 (gpl.23.3 linux/amd64)
minio  API: http://172.18.0.9:9000 http://127.0.0.1:9000
minio  MinIO: http://172.18.0.9:9001 http://127.0.0.1:9001
minio
minio  Docs: https://docs.min.io
airflow-init-1 The container is run as root user. For security, consider using a regular user account.
airflow-init-1 /home/airflow/.local/lib/python3.11/site-packages/airflow/configuration.py:859 FutureWarning: section/key [core/sql_alchemy_conn] has been deprecated, you should use [database/sql_alchemy_conn] instead. Please update your 'conf.get*' call to use t
airflow-init-1 he new name
airflow-init-1 DB: postgresql+psycopg2://airflow***@postgres-airflow:5432/airflow
airflow-init-1 Performing upgrade to the metadata database postgresql+psycopg2://airflow***@postgres-airflow:5432/airflow
airflow-init-1 [2024-12-10T14:06:37.694+0000] (migration.py:207) INFO - Context impl PostgresqlImpl.
airflow-init-1 [2024-12-10T14:06:37.679+0000] (migration.py:218) INFO - Will assume transactional DDL.
airflow-init-1 [2024-12-10T14:06:37.679+0000] (migration.py:207) INFO - Context impl PostgresqlImpl.
airflow-init-1 [2024-12-10T14:06:37.679+0000] (migration.py:218) INFO - Will assume transactional DDL.
airflow-init-1 [2024-12-10T14:06:37.680+0000] (db.py:1078) INFO - Creating tables
airflow-init-1 INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
airflow-init-1 INFO [alembic.runtime.migration] Will assume transactional DDL.
airflow-init-1 INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
airflow-init-1 INFO [alembic.runtime.migration] Will assume transactional DDL.
airflow-init-1 Database migrating done.
airflow-init-1 /home/airflow/.local/lib/python3.11/site-packages/airflow/configuration.py:859 FutureWarning: section/key [core/sql_alchemy_conn] has been deprecated, you should use [database/sql_alchemy_conn] instead. Please update your 'conf.get*' call to use t
airflow-init-1 he new name
airflow-init-1 /home/airflow/.local/lib/python3.11/site-packages/lask_linter/extension.py:333 UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://fl
airflow-init-1 ask-linter.readthedocs.io/en/latest/storage-backend for documentation about configuring the storage backend.
airflow-init-1 airflow already exist in the db
airflow-init-1 /home/airflow/.local/lib/python3.11/site-packages/airflow/configuration.py:859 FutureWarning: section/key [core/sql_alchemy_conn] has been deprecated, you should use [database/sql_alchemy_conn] instead. Please update your 'conf.get*' call to use t
airflow-init-1 he new name
airflow-init-1 [ 2.10.3
airflow-init-1 airflow-init-1 exited with code 0
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:05:06] "GET /health HTTP/1.1" 200 -
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:05:21 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10T14:06:08.288+0000] (triggerer-job_runner.py:518) INFO - 0 triggers currently running
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:06:18] "GET /health HTTP/1.1" 200 -
airflow-worker-1 127.0.0.1 - - [10/Oct/2024 14:06:21 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:06:26] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:06:35 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10T14:07:08.360+0000] (triggerer-job_runner.py:518) INFO - 0 triggers currently running
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:07:13] "GET /health HTTP/1.1" 200 -
airflow-worker-1 127.0.0.1 - - [10/Oct/2024 14:07:15] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:07:21 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:07:40] "GET /health HTTP/1.1" 200 -
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:07:51 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10T14:08:08.392+0000] (triggerer-job_runner.py:518) INFO - 0 triggers currently running
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:08:16] "GET /health HTTP/1.1" 200 -
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:08:25 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10T14:08:48.406+0000] (triggerer-job_runner.py:518) INFO - 0 triggers currently running
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:08:46] "GET /health HTTP/1.1" 200 -
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:08:51 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10T14:09:08.409+0000] (triggerer-job_runner.py:518) INFO - 0 triggers currently running
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:09:15] "GET /health HTTP/1.1" 200 -
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:09:21 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10T14:09:47.421+0000] (triggerer-job_runner.py:518) INFO - 0 triggers currently running
airflow-scheduler-1 127.0.0.1 - - [10/Oct/2024 14:09:47] "GET /health HTTP/1.1" 200 -
airflow-webserver-1 127.0.0.1 - - [10/Oct/2024 14:09:52 +0000] "GET /health HTTP/1.1" 200 318 "-" "curl/7.88.1"
airflow-triggerer-1 [2024-12-10 14:09:57.107 UTC [29] LOG: checkpoint starting: time
db-1
```



## Récupération effective des fichiers depuis le [www.nyc.gov](http://www.nyc.gov) vers minio à travers script grab\_parquet.py



## TP2 : Injection des données dans le Data Warehouse

- Modification du script « **dump\_to\_sql.py** » (src/data/dump\_to\_sql.py) pour l'insertion dans une base de données.
- Récupération des fichiers stockés sur minio et injection dans une base de données PostgreSQL en passant par un stockage temporaire en local

pgAdmin 4

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@Archidata

Query: Query History

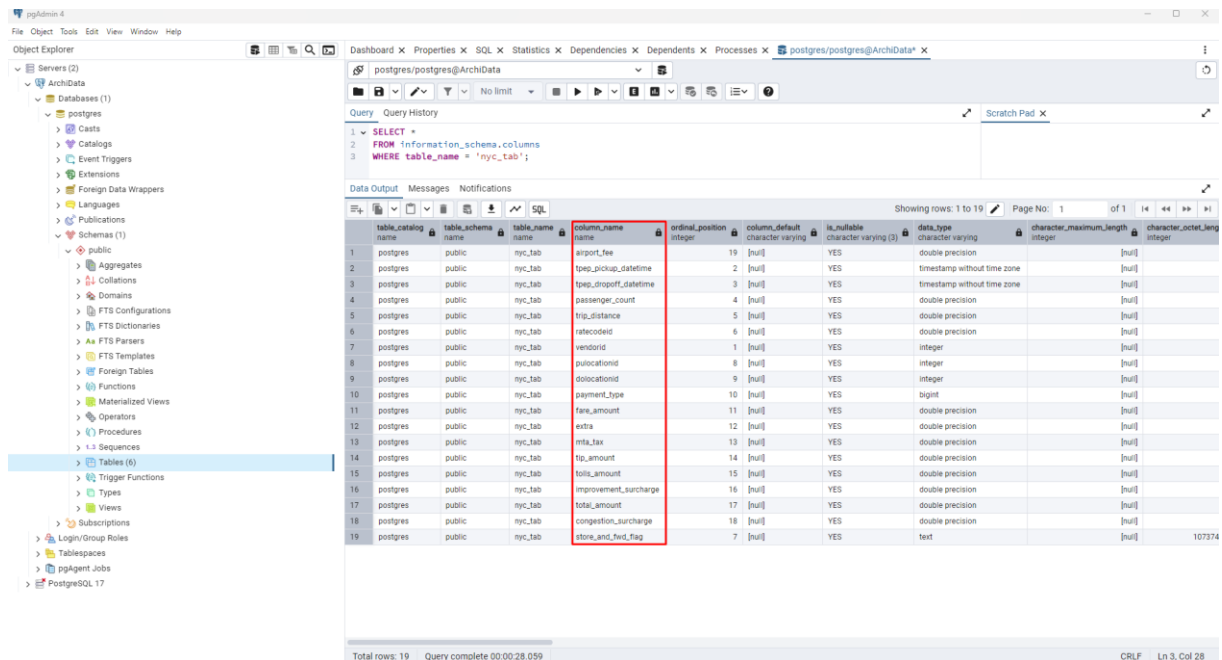
1 SELECT \* FROM taxi LIMIT 100;

Data Output Messages Notifications

vendor_id	taxi_pickup_datetime	taxi_dropoff_datetime	passenger_count	trip_distance	ratecode_id	store_and_fwd_flag	publication_id	declassification_id	payment_type	fare_amount
1	2024-01-01 00:17:06	2024-01-01 00:35:31	1	4.7	1	N	234	79	1	1
4	2024-01-01 00:36:38	2024-01-01 00:44:56	1	1.4	1	N	79	211	1	1
5	2024-01-01 00:46:51	2024-01-01 00:52:37	1	0.6	1	N	211	146	1	1
6	2024-01-01 00:54:08	2024-01-01 01:26:31	1	4.7	1	N	146	141	1	1
7	2024-01-01 00:49:44	2024-01-01 01:15:47	2	10.82	1	N	138	161	1	1
8	2024-01-01 00:36:40	2024-01-01 00:56:40	0	3	1	N	246	231	2	1
9	2024-01-01 00:26:01	2024-01-01 00:54:12	1	5.44	1	N	161	261	2	1
10	2024-01-01 00:28:08	2024-01-01 00:29:16	1	0.04	1	N	113	113	2	1
11	2024-01-01 00:25:22	2024-01-01 00:41:41	2	0.75	1	N	107	127	1	1
12	2024-01-01 00:25:00	2024-01-01 00:34:03	2	1.2	1	N	188	246	1	1
13	2024-01-01 00:35:16	2024-01-01 01:11:32	2	8.2	1	N	246	190	1	1
14	2024-01-01 00:43:27	2024-01-01 00:47:11	2	0.4	1	N	68	90	1	1
15	2024-01-01 00:51:53	2024-01-01 00:55:43	1	0.6	1	N	90	68	2	1
16	2024-01-01 00:50:09	2024-01-01 01:03:57	1	5	1	N	132	216	2	1
17	2024-01-01 00:41:06	2024-01-01 00:53:42	1	1.5	1	N	164	79	1	1
18	2024-01-01 00:52:04	2024-01-01 00:52:28	1	0	1	N	237	237	2	1
19	2024-01-01 00:56:38	2024-01-01 01:03:17	1	1.5	1	N	141	263	1	1
20	2024-01-01 00:32:34	2024-01-01 00:49:33	1	2.57	1	N	161	263	1	1
21	2024-01-01 00:52:30	2024-01-01 00:57:37	1	0.66	1	N	263	263	1	1
22	2024-01-01 00:36:20	2024-01-01 01:13:33	2	1.7	1	N	246	170	1	1
23	2024-01-01 00:44:24	2024-01-01 00:51:57	1	0.94	1	N	158	113	1	1
24	2024-01-01 00:14:29	2024-01-01 00:14:29	1	0	1	N	236	264	2	1
25	2024-01-01 00:42:05	2024-01-01 01:16:49	1	23.9	5	N	263	263	1	1
26	2024-01-01 00:12:35	2024-01-01 00:19:21	2	1.08	1	N	146	4	1	1
27	2024-01-01 00:20:11	2024-01-01 00:42:53	1	5.88	1	N	4	238	1	1
28	2024-01-01 00:44:01	2024-01-01 00:54:31	2	2.22	1	N	238	90	1	1

Total rows: 100 Query complete: 00:00:00.244 CRLF Ln 1, Col 32

## Vérifications de la présence de toutes les colonnes des données transférées depuis minio. (Présence des 19 colonnes encadrées en rouge)



The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane shows the database structure, with 'Tables (6)' selected under the 'public' schema. The main pane displays a SQL query and its results.

Query:

```
1 SELECT *
2 FROM information_schema.columns
3 WHERE table_name = 'nyc_tax';
```

Data Output:

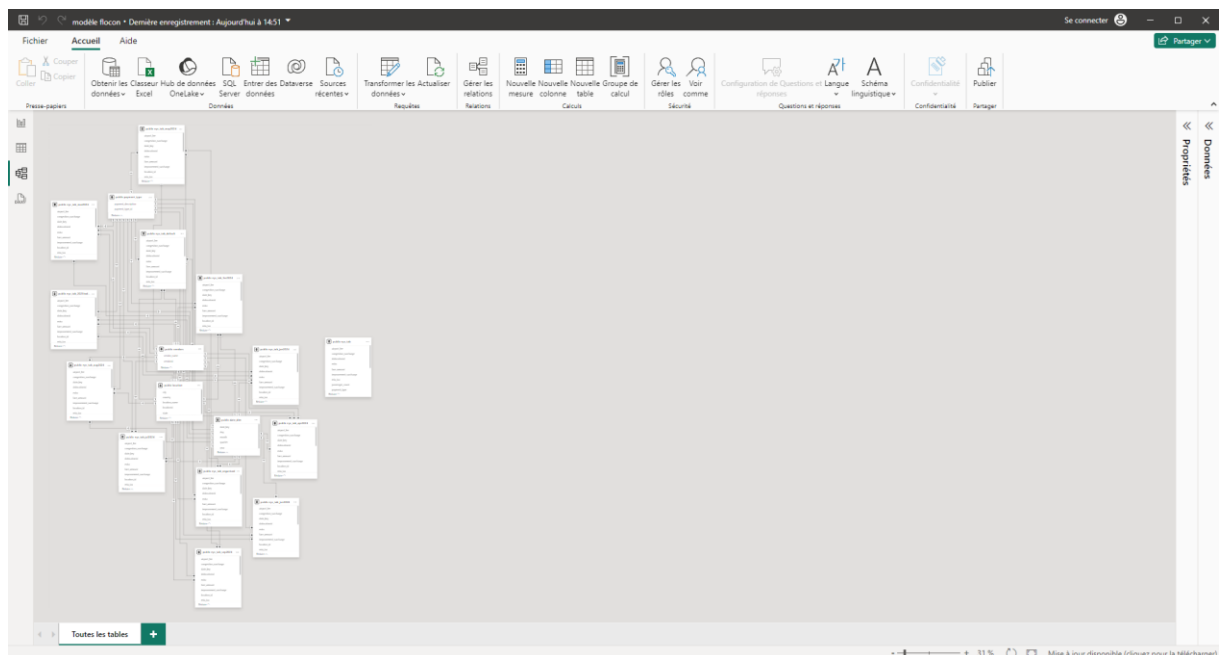
table_catalog	table_schema	table_name	column_name	ordinal_position	column_default	is_nullable	data_type	character_maximum_length	character_octet_length
postgres	public	nyc_tax	airport_fee	1		YES	double precision		
postgres	public	nyc_tax	trip_pickup_datetime	2		YES	timestamp without time zone		
postgres	public	nyc_tax	trip_dropoff_datetime	3		YES	timestamp without time zone		
postgres	public	nyc_tax	passenger_count	4		YES	double precision		
postgres	public	nyc_tax	trip_distance	5		YES	double precision		
postgres	public	nyc_tax	ratecodeid	6		YES	double precision		
postgres	public	nyc_tax	vendorid	7		YES	integer		
postgres	public	nyc_tax	pulocationid	8		YES	integer		
postgres	public	nyc_tax	dolocationid	9		YES	integer		
postgres	public	nyc_tax	payment_type	10		YES	bigint		
postgres	public	nyc_tax	fare_amount	11		YES	double precision		
postgres	public	nyc_tax	extra	12		YES	double precision		
postgres	public	nyc_tax	mta_tax	13		YES	double precision		
postgres	public	nyc_tax	tip_amount	14		YES	double precision		
postgres	public	nyc_tax	tolls_amount	15		YES	double precision		
postgres	public	nyc_tax	improvement_surcharge	16		YES	double precision		
postgres	public	nyc_tax	total_amount	17		YES	double precision		
postgres	public	nyc_tax	congestion_surcharge	18		YES	double precision		
postgres	public	nyc_tax	store_and_fwd_flag	19		YES	text		

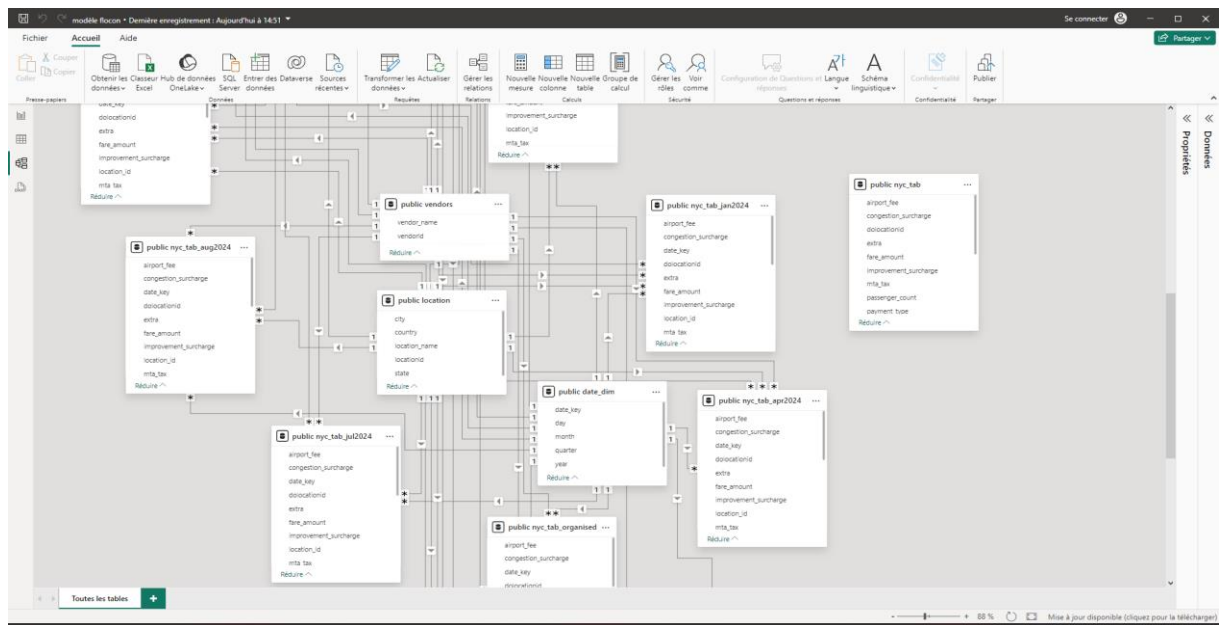
Total rows: 19 Query complete 00:00:28.059

## TP3 : Modélisation en flocon

Création des tables en flocons avec les contraintes associés. A trouver dans : **flocon\_sql.pdf** (docs/ flocon\_sql.pdf).

Visualisation des liens entre les différentes colonnes



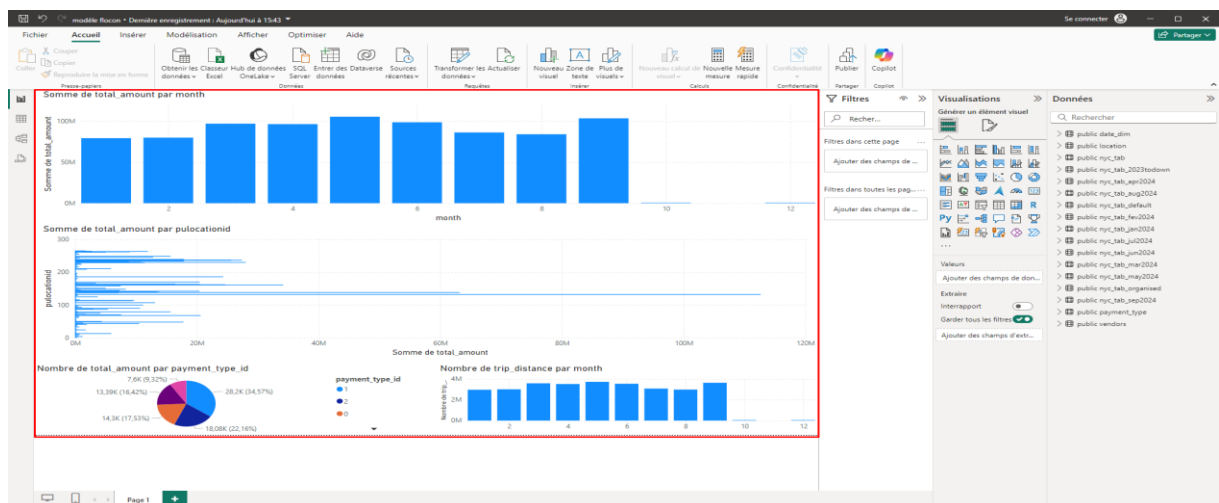


## TP4 : Visualisation des données

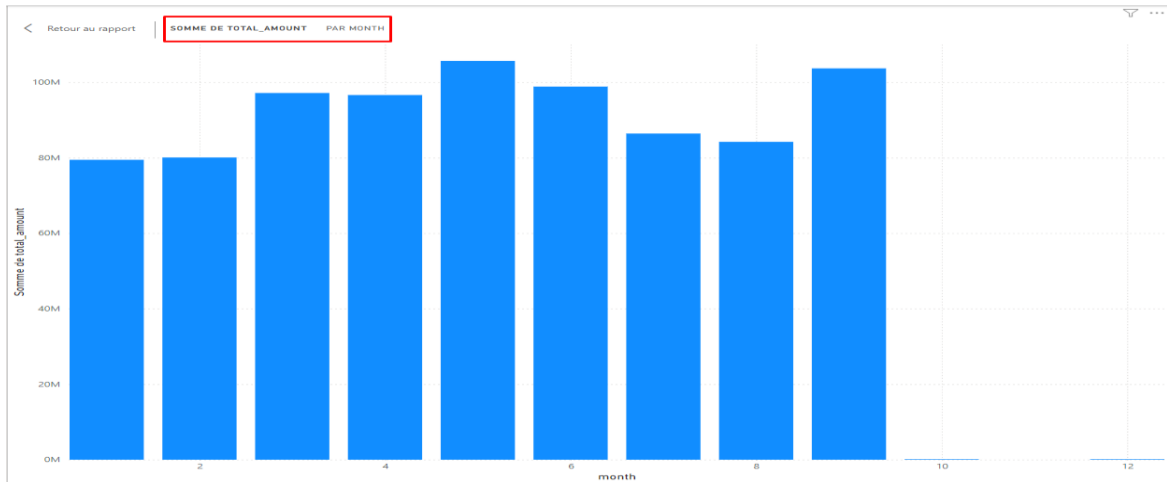
- Outil utilisé : PowerBI
- Résultats obtenus :

## Les Dashboard dans PowerBI

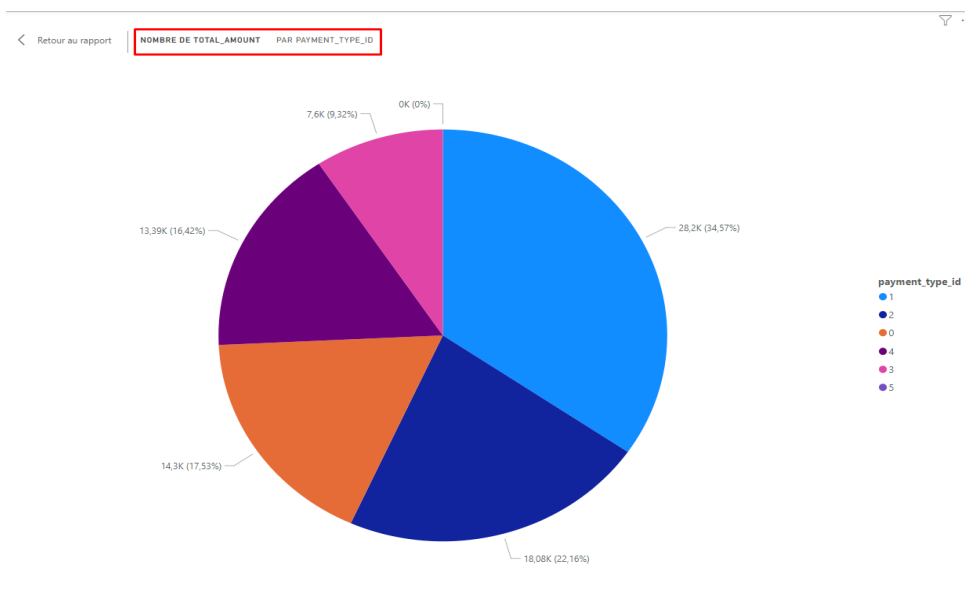
### ➤ Vue d'ensemble



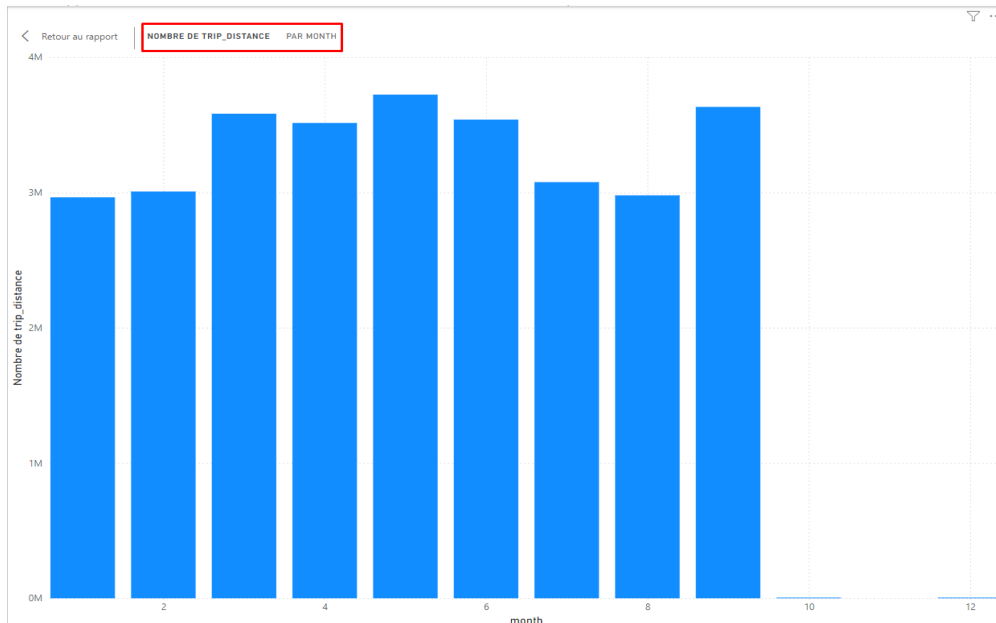
## ➤ Sommes totale des recettes par mois



## ➤ Nombre de paiement par type de paiement



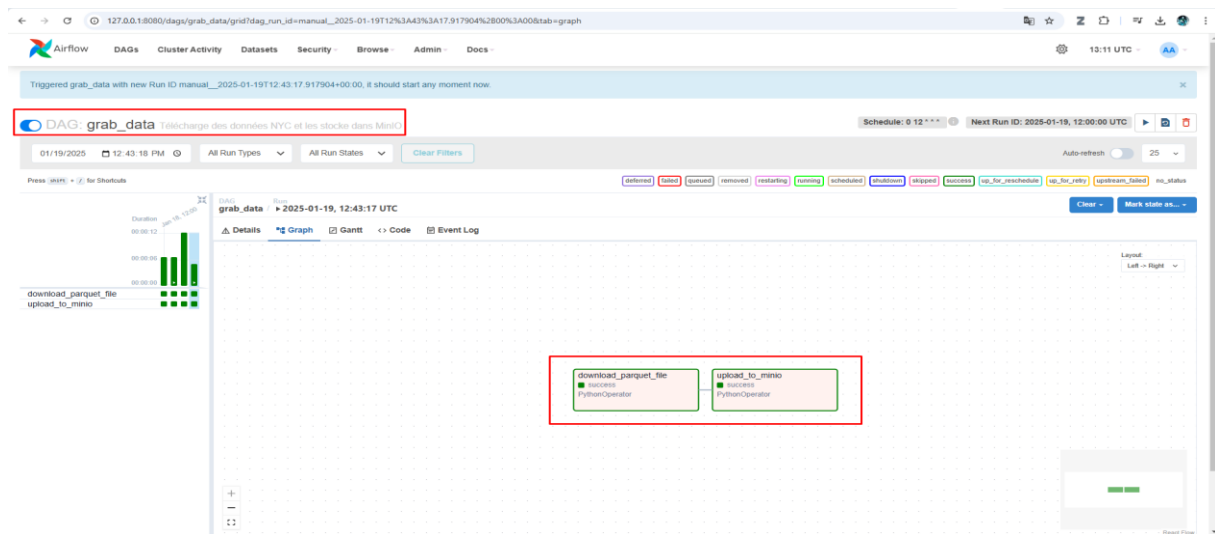
➤ **Distance parcourue par les taxis par mois**



## TP5 : Automatisation du TP1 (récupération des fichiers depuis [www.nyc.gov](http://www.nyc.gov) vers minio) avec Airflow

- Le script du dag est à retrouver dans le fichier « **dag\_script.py** » (airflow/dags/dag\_script.py).

➤ Activation et résultat du lancement du dag sur Airflow



➤ Résultat et mise à jour du fichier téléchargé dans minio

