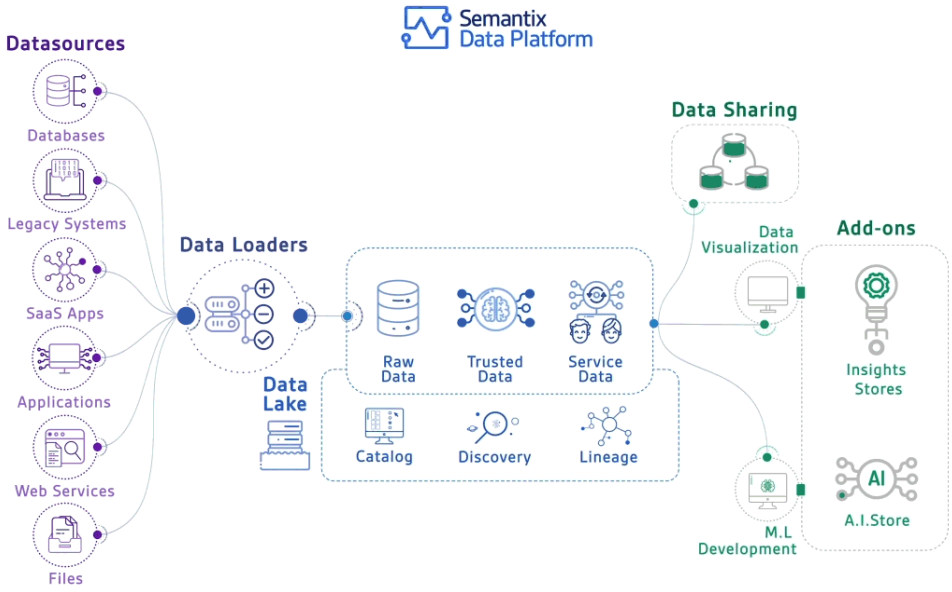
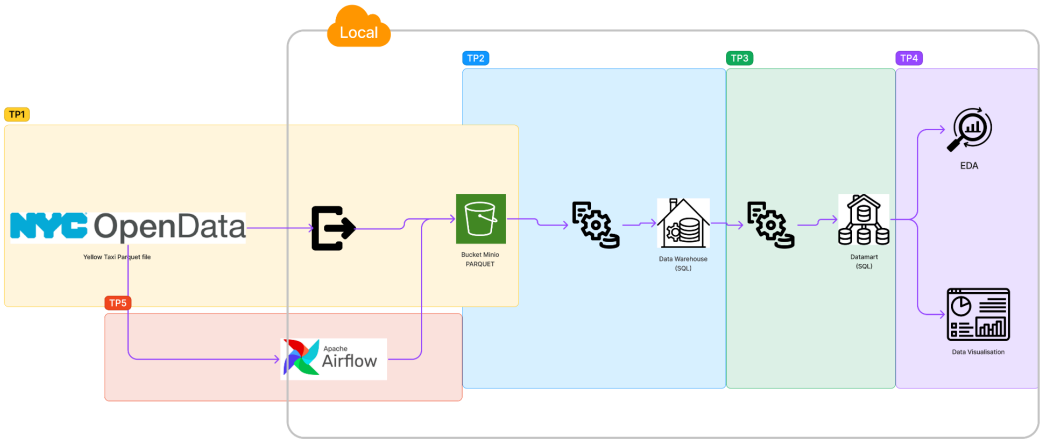
Architecture des données





Lucas CHIPAN, Belkis COSKUN, Lyes BOUMRAH, Victory MBANZILA DIMBOU

20/01/2025

Ingénierie des données I1

INTRODUCTION : But du projet / Problématique

L'objectif principal de ce projet est de construire une architecture décisionnelle robuste et industrialisée, capable de traiter et d'analyser efficacement des volumes importants de données. Plus précisément, le projet vise à répondre aux besoins d'une entreprise de VTC basée à New York en lui fournissant des outils décisionnels basés sur ses données opérationnelles.

La problématique repose sur plusieurs axes :

1. **Automatisation des flux de données** : Comment garantir une récupération continue et sans erreur des données brutes provenant de sources externes, comme le site de l'État de New York ?
2. **Transformation et structuration des données** : Comment transformer ces données hétérogènes en informations exploitables pour des analyses approfondies ?
3. **Optimisation des performances** : Quelle architecture mettre en place pour répondre aux exigences de rapidité et d'efficacité des requêtes sur les données massives ?
4. **Création de valeur via la visualisation** : Comment exploiter ces données pour générer des tableaux de bord interactifs et dynamiques qui facilitent la prise de décision ?

À travers ce projet, nous allons déployer une solution "data-driven" comprenant les étapes clés suivantes :

- Récupération des données brutes et leur stockage dans un Data Lake.
- Transformation des données pour alimenter un Data Warehouse.
- Optimisation et structuration des données dans un Data Mart pour améliorer les performances analytiques.
- Conception de tableaux de bord dynamiques via des outils de visualisation comme Tableau ou Power BI.
- Automatisation des processus pour garantir une maintenance réduite et une mise à jour continue des données.

Ce projet s'inscrit dans un cadre pédagogique visant à développer des compétences

concrètes dans la mise en œuvre d'une architecture décisionnelle de bout en bout.

HYPOTHÈSE : Comment répondre à cette problématique (cité ce qu'on a déjà code etc et ce qu'on y ajoute)

Pour répondre à la problématique, nous mettrons en œuvre les éléments suivants :

1. **Code existant** :
    - Scripts en Python ou Scala pour récupérer les jeux de données de novembre et décembre 2023 depuis la source officielle et les stocker dans un Data Lake (Minio).
2. **Ajouts prévus** :
    - Automatisation des processus avec Apache Airflow, notamment pour la récupération des données mensuelles.
    - Développement de pipelines ETL pour transformer et transférer les données entre les différentes couches (Data Lake, Data Warehouse, Data Mart).
    - Conception de modèles de données en étoile pour optimiser les performances des requêtes.
    - Création de tableaux de bord interactifs avec Tableau ou Power BI

MATÉRIEL (Technologie utilisé)

Outils de stockage et manipulation des données :

- **Minio** : Utilisé comme Data Lake pour stocker les données brutes.
- **PostgreSQL** : Base de données relationnelle pour le Data Warehouse et le Data Mart.

Langages de programmation :

- Python ou Scala pour le développement des pipelines ETL. Nous avons privilégié python.

Outils de visualisation :

- Tableau Software ou Power BI pour la création des tableaux de bord.

Outils d'automatisation :

- Apache Airflow pour la gestion des tâches répétitives.

Configuration matérielle :

- PC avec au moins 16 Go de RAM et 10 Go d'espace disque disponible.

PROCÉDURE (Tp)

1. **TP1 : Récupération des données**
    a. Création de scripts Python/Scala pour télécharger les données depuis la source officielle et les stocker dans Minio.
    b. Mise en place d'une structure d'alimentation mensuelle automatisée.
2. **TP2 : ETL vers le Data Warehouse**
    a. Développement de pipelines ETL pour extraire les données de Minio, les transformer, et les charger dans une base PostgreSQL.
3. **TP3 : Création du Data Mart**
    a. Conception d'un modèle de données en étoile pour répondre aux besoins de visualisation.
    b. Transfert des données traitées vers une base OLAP dédiée.

4. **TP4 : Visualisation des données**
    a. Conception et publication d'un tableau de bord dynamique connecté au Data Mart.
5. **TP5 : Automatisation**
    a. Mise en œuvre d'une DAG (Directed Acyclic Graph) dans Apache Airflow pour automatiser la récupération et l'intégration des données.

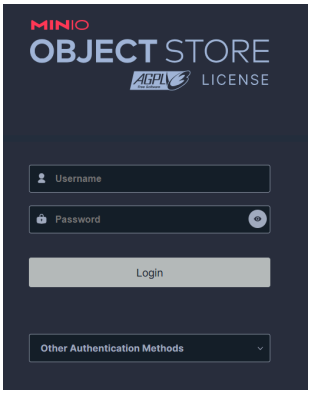
Processus pour la réalisation de chaque TP

Voici de quelle manière nous avons abordé et résolut chaque TP ainsi qu'une partie dédiée aux résultats :

TP 1 : Récupération des données et stockage dans Minio

Processus :

1. **Préparation :**
    - Cloner le dépôt GitHub contenant les scripts nécessaires.
    - Puis nous avons créé un environnement python grâce à miniconda avec la commande **conda create -n "ATL-Datamart" python=3.12**
    - Une fois l'environnement créé nous l'avons activé **conda activate "ATL-Datamart"** puis **pip install -r requirements.txt**
    - Configurer Docker et lancer l'environnement via `docker compose up`.
2. **Complétion du script `grab_parquet.py` :**
    - Ajout des fonctions pour télécharger les fichiers Parquet pour les mois de janvier à août 2023 et le mois dernier.
    - Utilisation de la bibliothèque Minio pour stocker les fichiers téléchargés dans un bucket.
3. **Vérifications :**
    - Lancer le script et s'assurer que les fichiers sont présents dans Minio ou se connecter à l'interface web ou à l'adresse http://localhost:9001/login puis dans le bucket utilisé en tant que DataLake (ici nous prenons l'exemple d'un fichier du au capacité de notre ordinateur mais le script est capable d'en traiter plusieurs)



Résultats :

- **Succès :** Les données des mois spécifiés sont correctement téléchargées et stockées dans Minio.
- **Vérifications visuelles :** Les fichiers sont visibles dans le bucket "yellow-taxi-data".

TP 2 : Chargement des données dans PostgreSQL

Processus :

1. **Configuration PostgreSQL :**
    - Lancer un conteneur PostgreSQL via Docker avec les paramètres de connexion nécessaires.
    - Créer la base de données `data_warehouse` et une table `mon_schema.yellow_tripdata` pour stocker les données.
2. **Complétion du script `dump_to_sql.py` :**
    - Télécharger les fichiers depuis Minio.
    - Charger les fichiers Parquet dans la table PostgreSQL à l'aide de pandas et Psycopg2.

3. **Gestion des erreurs :**
    - Ajustement des noms de colonnes dans le script pour résoudre les incompatibilités entre les fichiers et la table PostgreSQL.

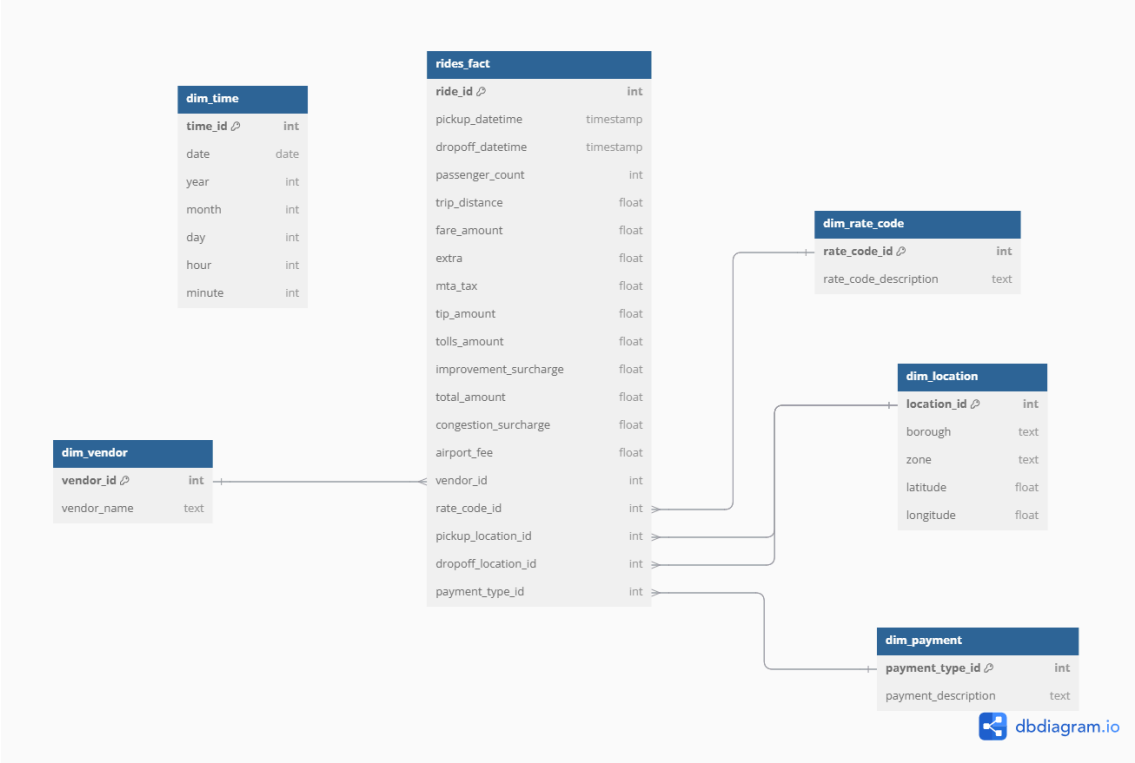
Résultats :

- **Succès :** Les fichiers sont correctement insérés dans la table `mon_schema.yellow_tripdata` .
- **Vérifications :**
    - Requête `SELECT * FROM mon_schema.yellow_tripdata LIMIT 10;` montre un échantillon des données.
    - Comptage des lignes pour valider l'intégration complète.

TP 3 : Création et alimentation du Data Mart

Processus :

1. **Configuration du Data Mart :**
    - Lancer un deuxième conteneur PostgreSQL pour le Data Mart.
    - Créer la base de données `datamart` et les tables dimensionnelles et factuelles en modèle étoile via un script SQL.
2. **Alimentation des dimensions et des faits :**



- Utiliser `dblink` pour connecter le Data Warehouse et le Data Mart.
- Remplir les dimensions (vendor, rate code, location, payment type, time) avec des requêtes SQL.
- Charger la table factuelle `rides_fact` avec les données du dataWarehouse.

```
datamart=# \i alter.sql
UPDATE 0
INSERT 0 3
INSERT 0 2
UPDATE 0
psql:alter.sql:55: ERROR:  column "service_zone" of relation "dim_location" does not exist
LINE 1: ...owflake.dim_location (location_id, borough, zone, service_zo...
```

3. **Validation :**
    - Vérification des clés primaires et des contraintes.
    - Inspection manuelle des dimensions et de la table factuelle.

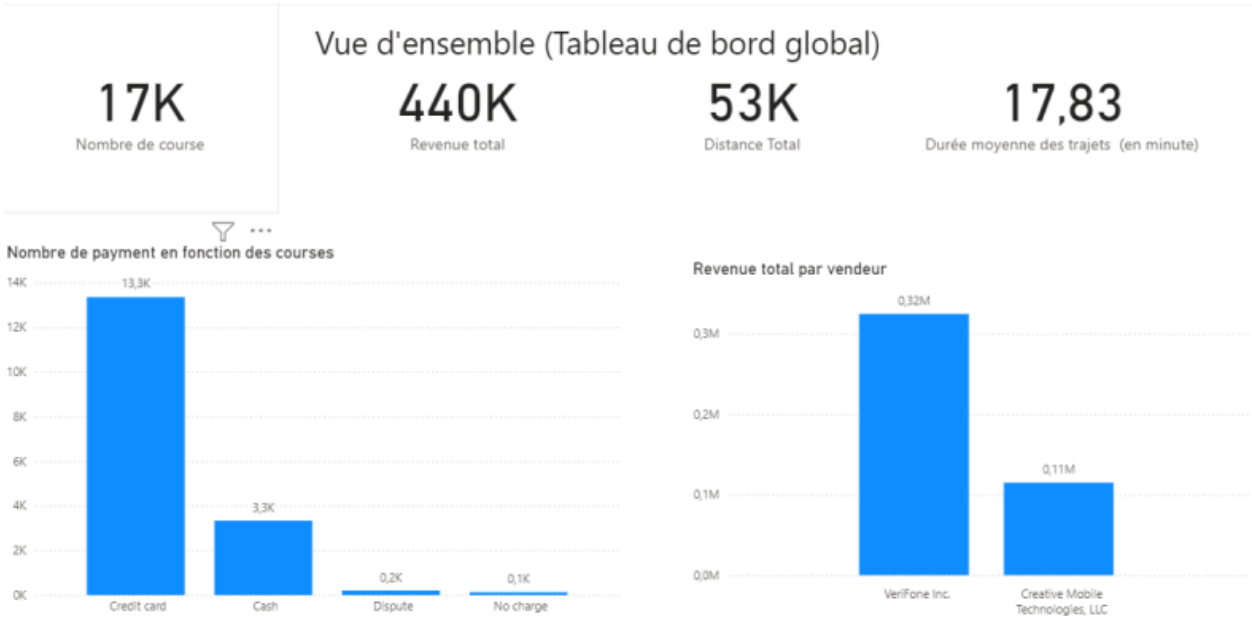
Résultats :

- **Succès :** Les tables du modèle étoile sont créées et alimentées avec succès.
- **Vérifications :**
    - Les données des dimensions et de la table factuelle sont cohérentes avec les données source.
    - Les fichiers complémentaires (comme `taxi_zone_lookup.csv`) ont été intégrés.

TP 4 : Visualisation des données

Processus :

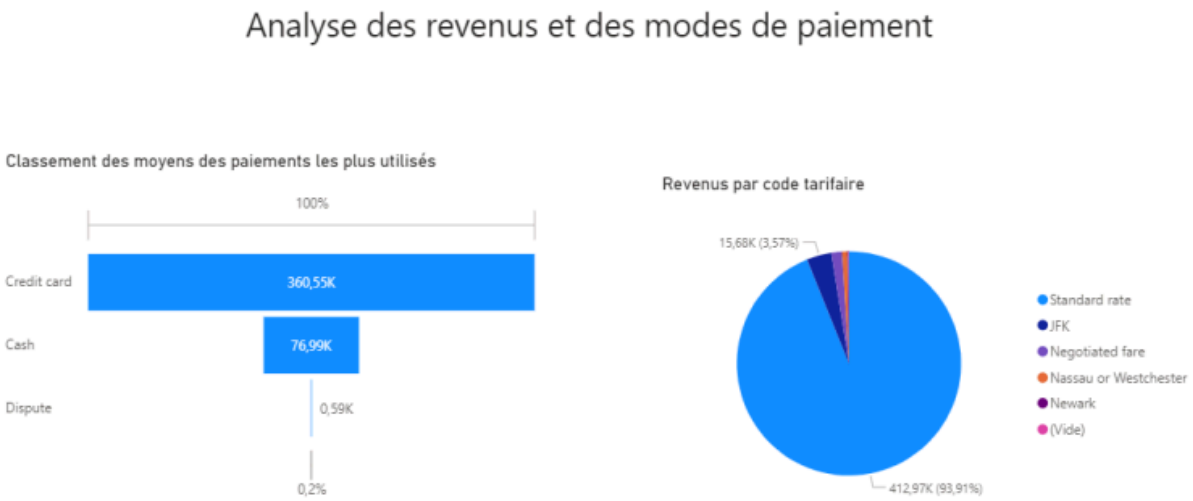
1. **Connexion au Data Mart :**
    - Nous avons installé les pilotes nécessaires pour Power BI.
    - Configuré la source de données pour pointer vers le Data Mart PostgreSQL.
2. **Exploration des données (EDA) :**
    - Analyser les données pour identifier les KPI clés (par exemple, distance moyenne, revenus par type de paiement) qui sont pertinents et nous permettent de relever des insights.
3. **Création des visualisations :**
    - Conception de graphiques pour chaque KPI
    - Assemblage des visualisations dans un tableau de bord interactif sur Power Bi.
4. **Mise à jour automatique :**
    - Configurer les visualisations pour qu'elles se mettent à jour automatiquement à partir du Data Mart.



Analyse temporelle

Analyse géographique

Répartition total des trajet



CONCLUSION : Réponse à la problématique et méthode d'approche

La problématique initiale posait plusieurs défis liés à la gestion et à l'analyse de données volumineuses, notamment pour une entreprise de VTC à New York. Les objectifs principaux étaient d'automatiser la récupération des données, de les structurer pour optimiser leur exploitation, et d'en extraire des insights exploitables à travers des tableaux de bord interactifs.

Pour répondre à cette problématique, plusieurs étapes ont été suivies de manière méthodique :

1. **Automatisation de la récupération des données** :
   La solution a permis de télécharger automatiquement les données de trajets en format Parquet via des scripts Python, puis de les stocker dans un Data Lake (Minio). Cette automatisation assure un flux continu des données mensuelles sans intervention manuelle.
2. **Transformation des données** :
   Les fichiers récupérés ont été intégrés dans une base de données relationnelle (PostgreSQL) grâce à des pipelines ETL, permettant un stockage structuré dans un Data Warehouse. Ces données brutes ont ensuite été organisées dans un modèle en étoile au sein d'un Data Mart, offrant une base optimisée pour les requêtes analytiques.
3. **Extraction de valeur par la visualisation** :

Les données du Data Mart ont été utilisées pour concevoir des tableaux de bord interactifs via Tableau ou Power BI. Ces tableaux de bord présentent des indicateurs clés tels que les montants totaux, les distances moyennes, et les pourboires selon le type de paiement, permettant ainsi une prise de décision éclairée.

4. **Industrialisation et maintenance** :
   Enfin, normalement, la mise en œuvre d'une DAG dans Apache Airflow a permis d'automatiser le processus de récupération et d'intégration des données, garantissant une actualisation régulière et fiable.

En conclusion, cette architecture décisionnelle répond pleinement à la problématique en mettant en place un système robuste, automatisé, et scalable pour la gestion des données de trajets. L'approche adoptée a permis non seulement de résoudre les défis initiaux, mais également de fournir une solution évolutive, prête à être enrichie ou étendue en fonction des besoins futurs de l'entreprise. Le projet a démontré l'efficacité d'une solution "data-driven" pour transformer des données brutes en insights stratégiques exploitables.