

Mini-Projet

A présenter le 8 Janvier 2026

Big Data Warehouse NoSQL – Simulation de flux de commandes multicanal

1. Contexte général

La société **MultiMarket** opère plusieurs canaux de vente :

- Un **site web**,
- Une **application mobile**,
- Des **boutiques physiques**.

Chaque canal génère des **commandes client** à différents moments de la journée.

Dans un environnement de **production réel**, chaque canal génère **un fichier JSON** pour chaque commande.

Ces fichiers sont **déposés** dans un **répertoire de serveur** partagé, accessible au service décisionnel.

Votre mission est de **simuler cet environnement**, de **collecter les commandes** depuis ce serveur, et de **construire un entrepôt de données Big Data Warehouse** basé sur **MongoDB**.

2. Objectifs pédagogiques

- Simuler un **flux de données hétérogène** en provenance de différentes sources.
- Gérer l'**arrivée continue de données**.
- Construire un **Data Warehouse NoSQL** capable d'ingérer des commandes en continu.
- Maîtriser l'**analyse par agrégation** sur des structures semi-structurées (JSON/MongoDB).

3. Travail demandé

Partie 1 – Simulation de la production de données

- Écrire **3 scripts simulateurs (en python)** :
 - site_web.py
 - application_mobile.py

- boutique_physique.py

Chaque script doit :

- Générer une nouvelle **commande** sous forme de fichier JSON de manière aléatoire.
- Déposer ce fichier dans un répertoire spécifique :
 - /data/sources/site_web/
 - /data/sources/application_mobile/
 - /data/sources/boutique_physique/
- Générer une nouvelle commande toutes les **2 à 5 secondes**.

Structure minimale d'une commande :

```
{
  "order_id": "234",
  "client_id": "123",
  "produits": [
    {
      "produit_id": "P015",
      "nom": "Laptop",
      "quantite": 2,
      "prix_unitaire": 8500.00
    }
  ],
  "date_commande": "2025-04-26T14:00:00",
  "canal": "web",
  "statut_commande": "payée",
  "livraison": {
    "adresse": "10 Av Roudani, Centre ville",
    "ville": "Casablanca"
  }
}
```

Notes :

- Les commandes **web** et **mobile** contiennent une **adresse de livraison**.
- Les commandes **boutique n'ont pas d'adresse** (L'achat est fait sur place).
- Il faut générer un volume d'au moins **500 commandes** par source pour avoir un corpus suffisant à analyser (avant de passer à la suite).

Partie 2 – Processus d'intégration des données (Projet BI)

- Écrire un **script de collecte** qui :
 - Surveille en continu les répertoires /data/sources/*/.
 - Récupère chaque fichier JSON nouvellement arrivé.
 - Valide les données (format JSON correct).
 - Insère les commandes dans la **base MongoDB** (nommée multi_market), dans une collection nommée « **commandes** ».
 - Après traitement, **déplace les fichiers** vers /data/archive/ pour archivage.

Exigences techniques :

- Détection automatique des nouveaux fichiers.
- Traitement en "quasi temps réel" (toutes les 10 secondes maximum).

- Robustesse en cas d'erreur (ex : ignorer un fichier corrompu).

Partie 3 – Agrégation et Analyse décisionnelle

- Utiliser MongoDB pour produire les rapports suivants :
 - Chiffre d'affaires **total** par **mois** et par **canal** (web, mobile, boutique).
 - **Top 10 des produits** les plus vendus (en quantité).
 - **Taux de commandes annulées** par canal.
 - **Chiffre d'affaires moyen** par commande pour chaque canal.
- Utiliser les **opérateurs MongoDB** :
 - \$group
 - \$project
 - \$match
 - \$sort

Partie 4 – Reporting visuel

Créer un **tableau de bord** :

- Utiliser **MongoDB Charts**, puis **Power BI**.
- Afficher :
 - Chiffre d'affaires **total** par **mois** et par **canal** (web, mobile, boutique).
 - **Top 10 des produits** les plus vendus (en quantité).
 - **Taux de commandes annulées** par canal.
 - **Chiffre d'affaires moyen** par commande pour chaque canal.