

Projet E-commerce MongoDB / PyMongo

Contexte

- Projet de base de données NoSQL sous MongoDB.
- Domaine choisi : e-commerce
- Utilisation de PyMongo pour manipuler les données.
- Hébergement local + MongoDB Atlas pour le bonus sécurité.

Modélisation

- **3 collections :**
 - **produits** : liste des produits du catalogue
 - **clients** : liste des clients
 - **commandes** : liste des commandes passées (embedding des produits commandés)

Liste des 20 requêtes avec explications

CRUD & requêtes simples

1. Ajouter un produit 'Webcam Logitech'

- But : Insérer un nouveau produit dans la collection **produits**.
- Lecture : Retourne le document inséré pour confirmer.

2. Lister tous les produits de la catégorie 'Informatique'

- But : Filtrer les produits par **categorie**.
- Lecture : Affiche tous les produits correspondants.

3. Incrémenter le stock de la Webcam Logitech de +10

- But : Mettre à jour le champ **stock** avec **\$inc**.
- Lecture : Confirme le nombre de documents modifiés.

4. Supprimer la Webcam Logitech

- But : Supprimer un produit précis par **nom**.
- Lecture : Affiche la confirmation de suppression.

5. Trouver un client nommé 'Jean Dupont'

- But : Faire un **find_one** sur **nom**.
- Lecture : Retourne le document du client.

6. Lister les commandes avec total et date

- But : Projection pour ne montrer que **total** et **date_commande**.
- Lecture : Affiche uniquement ces champs.

Filtres et requêtes avancées

1. Lister les clients inscrits en juillet 2025 avec email @example

- But : Utiliser `$and` avec un filtre `date` et `regex`.
- Lecture : Affiche tous les clients correspondant.

2. Trouver produits avec stock <10 ou prix >300

- But : Combiner deux conditions avec `$or`.
- Lecture : Liste des produits qui remplissent au moins une condition.

3. Lister clients dont le nom commence par 'J'

- But : Filtre via `$regex`.
- Lecture : Montre uniquement ces clients.

4. Trouver 'Tapis de course' ou 'Aspirateur Dyson'

- But : `$in` sur le champ `nom`.
- Lecture : Liste les produits précis.

5. Lister les commandes contenant un champ 'produits'

- But : Vérifier présence via `$exists`.
- Lecture : Montre les commandes valides.

6. Lister clients par date d'inscription décroissante

- But : Sort `date_inscription` avec projection.
- Lecture : Permet de voir les plus récents.

Agrégations

1. Calculer le total dépensé par chaque client

- But : `$group` par `client_id` avec `$sum`.
- Lecture : Retourne un total par client.

2. Top 3 clients les plus gros acheteurs

- But : `$group`, `$sort`, `$limit`.
- Lecture : Liste les 3 meilleurs clients.

3. Ajouter un champ TVA calculé à 20% sur commandes

- But : `$project` pour créer `TVA`.
- Lecture : Affiche chaque commande avec son montant TVA.

4. Jointure commandes + infos clients

- But : `$lookup` pour enrichir.
- Lecture : Chaque commande contient un tableau `client_info`.

5. Lister chaque produit commandé individuellement

- But : `$unwind` du tableau `produits`.
- Lecture : Chaque ligne = un produit commandé.

6. Compter combien de fois chaque produit a été commandé

- But : `$unwind`, puis `$group` et `$sort`.
- Lecture : Stats nb de commandes par produit.

7. Calculer total et moyenne des montants des commandes

- But : `$group` global avec `$sum` et `$avg`.
- Lecture : Donne deux indicateurs globaux.

8. Lister commandes de clients avec email 'dupont'

- But : `$lookup`, `$unwind` puis `$match`.
- Lecture : Montre uniquement ces commandes.

Sécurité / Atlas

- Création d'un utilisateur `appUser` avec rôle `readWrite` sur `Projet_Ecommerce`.
- Cluster hébergé sur MongoDB Atlas (M0 gratuit).
- IP whitelisted `0.0.0.0/0` pour tests.
- Données importées via `mongoimport` à partir des JSON exportés localement.

Exécution du projet

```
# Remplir la base en local
python insert_data.py

# Créer l'utilisateur MongoDB local
python create_user.py

# Lancer toutes les requêtes
python requetes.py
```