

Partie I : MongoDB

1. Créer les collections : produits, clients et commandes dans une base de données ecommerceDB.

```
db.produits.insertMany([ {Nom : "Smartphone XYZ", Catégorie : "Électronique", Prix : 599.99,
{Nom : "Chaussures de sport", Catégorie : "Mode", Prix : 75.00, Stock : 30 },
{Nom : "Laptop ABC", Catégorie : "Électronique", Prix : 899.99, Stock : 5 },
{Nom : "Livre de recettes", Catégorie : "Livres", Prix : 20.00, Stock : 50 },
{Nom : "Casque audio", Catégorie : "Électronique", Prix : 49.99, Stock : 25 }])

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('690f3f3d5e2a2f00a9128ec8'),
    '1': ObjectId('690f3f3d5e2a2f00a9128ec9'),
    '2': ObjectId('690f3f3d5e2a2f00a9128eca'),
    '3': ObjectId('690f3f3d5e2a2f00a9128ecb'),
    '4': ObjectId('690f3f3d5e2a2f00a9128ecc')
  }
}
```

```
> db.clients.insertMany([
  {Nom : " Ahmed Ben Saïd", Email : " ahmed.ben.saïd@mail.com", Adresse : " 10 Rue Al Madina, Casablanca, Maroc", Téléphone : " 06 77 77 77 77"},
  {Nom : " Fatima El Khalil", Email : "fatima.el.khalil@mail.com", Adresse : " 25 Avenue des Palmiers, Rabat, Maroc", Téléphone : " 06 77 77 77 77"},
  {Nom : " Noura Bakkar", Email : " noura.bakkar@mail.com", Adresse : " 22 Rue des Palmiers, Tunis, Tunisie", Téléphone : " 06 77 77 77 77"},
  {Nom : " Aisha Darwish", Email : " aisha.darwish@mail.com", Adresse : " 35 Rue Al Kindi, Damas, Syrie", Téléphone : " 06 77 77 77 77"},
  {Nom : " Hassan Al-Haddad", Email : "hassan.haddad@mail.com", Adresse : "12 Rue Al Mutanabi, Bagdad, Irak", Téléphone : " 06 77 77 77 77"}
])
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('690f40545e2a2f00a9128ecd'),
    '1': ObjectId('690f40545e2a2f00a9128ece'),
    '2': ObjectId('690f40545e2a2f00a9128ecf'),
    '3': ObjectId('690f40545e2a2f00a9128ed0'),
    '4': ObjectId('690f40545e2a2f00a9128ed1')
  }
}
```

```
db.commandes.insertMany([
  {
    Client: "Fatima El Khalil",
    Produits: "1x Smartphone XYZ, 2x Livre de recettes",
    Date_commande: "2024-11-01",
    Statut: "livrée",
    Montant_total: 639.99
  },
  {
    Client: "Hassan Al-Haddad",
    Produits: "1x Laptop ABC, 1x Casque audio",
    Date_commande: "2024-11-03",
    Statut: "en cours",
    Montant_total: 949.98
  }
])
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('690f41605e2a2f00a9128ed2'),
    '1': ObjectId('690f41605e2a2f00a9128ed3')
  }
}
```

3. Afficher tous les produits en stock.

```
> db.produits.find({ Stock: { $gt: 0 } })
< {
  _id: ObjectId('690f3f3d5e2a2f00a9128ec8'),
  Nom: 'Smartphone XYZ',
  'Catégorie': 'Électronique',
  Prix: 599.99,
  Stock: 10
}
{
  _id: ObjectId('690f3f3d5e2a2f00a9128ec9'),
  Nom: 'Chaussures de sport',
  'Catégorie': 'Mode',
  Prix: 75,
  Stock: 30
}
{
  _id: ObjectId('690f3f3d5e2a2f00a9128eca'),
  Nom: 'Laptop ABC',
  'Catégorie': 'Électronique',
  Prix: 899.99,
  Stock: 5
}
{
  _id: ObjectId('690f3f3d5e2a2f00a9128ecb'),
```

4. Rechercher tous les clients dont l'adresse contient "Paris".

```
> db.clients.find({ adresse: { $regex: "Paris", $options: "i" } })
<
```

5. Afficher toutes les commandes d'un client particulier (par exemple, Alice Dupont).

```
> db.commandes.find({ Client: "Alice Dupont" })
<
```

6. Rechercher toutes les commandes livrées.

```
> db.commandes.find({ Statut: "livrée" })
< {
  _id: ObjectId('690f41605e2a2f00a9128ed2'),
  Client: 'Fatima El Khalil',
  Produits: '1x Smartphone XYZ, 2x Livre de recettes',
  Date_commande: '2024-11-01',
  Statut: 'livrée',
  Montant_total: 639.99
}
```

7. Le stock du produit "Smartphone XYZ" doit être mis à jour suite à une nouvelle livraison (ajoutez 10 unités de stock).

```
> db.produits.updateOne(
  { Nom: "Smartphone XYZ" },
  { $inc: { Stock: 10 } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

8. Modifier le statut de la commande de Hassan Al-Haddad pour passer de "en cours" à "expédiée".

```
> db.commandes.updateOne({Nom:"Hassan Al-Haddad"},{$set:{Statut:"expédiée"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
ecommerceDB >
```

9. Supprimer un produit qui est épuisé (par exemple, un produit dont le stock est égal à 0).

```
> db.produits.deleteOne({ stock: 0 })
< {
  acknowledged: true,
  deletedCount: 0
}
```

10. Supprimer une commande annulée.

```
> db.commandes.deleteMany({Statut:"annulée"})
< {
  acknowledged: true,
  deletedCount: 0
}
```

11. Trouver tous les produits de la catégorie "Électronique" dont le prix est inférieur à 100.

```
> db.produits.find({Catégorie:"Électronique",Prix:{$lt:100}})
< {
  _id: ObjectId('690f4ac75e2a2f00a9128ed8'),
  Nom: 'Casque audio',
  'Catégorie': 'Électronique',
  Prix: 49.99,
  Stock: 25
}
```

12. Compter le nombre total de commandes passées par chaque client.

```
> db.commandes.aggregate([{$group: {_id: "$Client",total_commandes: { $sum: 1 }}}])
< {
  _id: 'Hassan Al-Haddad',
  total_commandes: 1
}
{
  _id: 'Fatima El Khalil',
  total_commandes: 1
}
```

13. Afficher les 5 produits les plus chers dans la collection.

```
> db.produits.find()
      .sort({ prix: -1 })
      .limit(5)
< {
  _id: ObjectId('690f4ac75e2a2f00a9128ed7'),
  Nom: 'Livre de recettes',
  'Catégorie': 'Livres',
  Prix: 20,
  Stock: 50
}
{
  _id: ObjectId('690f4ac75e2a2f00a9128ed8'),
  Nom: 'Casque audio',
  'Catégorie': 'Électronique',
  Prix: 49.99,
  Stock: 25
}
{
  _id: ObjectId('690f4ac75e2a2f00a9128ed5'),
```

Partie II : Pymongo

[lien github](#)

Partie III : Gestion des utilisateurs :

1. Créer 3 utilisateurs de la base de données avec des mots de passe :

a. Administrateur ayant le rôle dbAdmin

b. Utilisateur ayant le rôle userAdmin

c. Visiteur ayant le rôle read

```
> use ecommerceDB
< switched to db ecommerceDB
> use admin
< switched to db admin
> db.createUser({
  user: "Administrateur",
  pwd: "motdepasseAdmin",
  roles: [ { role: "dbAdmin", db: "ecommerceDB" } ]
})
< { ok: 1 }
> db.createUser({
  user: "Utilisateur",
  pwd: "motdepasseUser",
  roles: [ { role: "userAdmin", db: "ecommerceDB" } ]
})
< { ok: 1 }
> db.createUser({
  user: "Visiteur",
  pwd: "motdepasseVisiteur",
  roles: [ { role: "read", db: "ecommerceDB" } ]
})
< { ok: 1 }
```

2. Donner la syntaxe de connexion en tant que visiteur

mongosh -u Visiteur -p motdepasseVisiteur --authenticationDatabase ecommerceDB

3. Supprimer les utilisateurs créés

use ecommerceDB

```
db.dropUser("Administrateur")
```

```
db.dropUser("Utilisateur")
```

```
db.dropUser("Visiteur")
```