

# Rapport : atelier5 exercice1 et 2

**Module** : Base de Données MongoDB

**Nom**: Aya ouazzani akkar hammoudi

## Exercice 1 :

### SOMMAIRE :

1. PARTIE 1 — Création de la base & des collections
2. PARTIE 2 — Peupler les collections
3. PARTIE 3 — TD Python : Toutes les compétences étape par étape :
  - Exercice 1 — Se connecter à MongoDB en Python
  - Exercice 2 — Afficher les collections
  - Exercice 3 — Lire les utilisateurs
  - Exercice 4 — Insérer un nouvel utilisateur via Python
  - Exercice 5 — Filtrage (find)
  - Exercice 6 — Mise à jour
  - Exercice 7 — Supprimer
  - Exercice 8 — Index
  - Exercice 9 — Agrégation (statistiques)
  - Exercice 10 — Jointure : afficher commandes + utilisateurs + produits

### 1. PARTIE 1 — Création de la base & des collections



### PARTIE 2 — Peupler les collections

#### 1. Insérer les utilisateurs :

```
test> db.utilisateurs.insertMany([
  {nom: "Ali", email: "ali@test.com", age: 23, ville: "Tanger", hobbies: ["football","gaming"]},
  {nom: "Sara", email: "sara@test.com", age: 29, ville: "Rabat", hobbies: ["lecture","voyage"]},
  {nom: "Nadia", email: "nadia@test.com", age: 35, ville: "Casa", hobbies: ["cuisine"]},
  {nom: "Yassine",email:"yassine@test.com",age:27, ville: "Fès", hobbies: ["fitness","musique"]},
  {nom: "Hassan",email: "hassan@test.com", age:31, ville: "Tanger", hobbies: ["lecture"]}
])
```

```

    acknowledged: true,
    insertedIds: {
      '0': ObjectId('691d8b4221e7e5821d083e88'),
      '1': ObjectId('691d8b4221e7e5821d083e89'),
      '2': ObjectId('691d8b4221e7e5821d083e8a'),
      '3': ObjectId('691d8b4221e7e5821d083e8b'),
      '4': ObjectId('691d8b4221e7e5821d083e8c')
    }
  }
}

```

## 2. Insérer les produits :

```

test> db.produits.insertMany([
  { nom: "PC Portable", prix: 7000, stock: 12 },
  { nom: "Souris Gaming", prix: 250, stock: 50 },
  { nom: "Clavier Mécanique", prix: 600, stock: 30 },
  { nom: "Écran 24 pouces", prix: 1200, stock: 18 }
])

```

```

< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('691d8b8921e7e5821d083e8d'),
    '1': ObjectId('691d8b8921e7e5821d083e8e'),
    '2': ObjectId('691d8b8921e7e5821d083e8f'),
    '3': ObjectId('691d8b8921e7e5821d083e90')
  }
}

```

## ◆ 3. Insérer les commandes :

1. Récupérer les `_id` des utilisateurs et 2. Récupérer les `_id` des produits :

```

test> user1 = db.utilisateurs.findOne({nom: "Ali"})._id
      user2 = db.utilisateurs.findOne({nom: "Sara"})._id

      prod1 = db.produits.findOne({nom: "PC Portable"})._id
      prod2 = db.produits.findOne({nom: "Souris Gaming"})._id

```

```

< ObjectId('691d8b8921e7e5821d083e8e')

```

3. Insérer les commandes (sans texte parasite) :

```
test> db.commandes.insertMany([
  { idUtilisateur: user1, idProduit: prod1, quantite: 1, dateCommande: new Date("2024-02-01") },
  { idUtilisateur: user1, idProduit: prod2, quantite: 2, dateCommande: new Date("2024-02-10") },
  { idUtilisateur: user2, idProduit: prod2, quantite: 1, dateCommande: new Date("2024-03-01") },
  { idUtilisateur: user2, idProduit: prod1, quantite: 1, dateCommande: new Date("2024-03-12") },
  { idUtilisateur: user2, idProduit: prod2, quantite: 3, dateCommande: new Date("2024-03-20") }
])
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('691d8d3221e7e5821d083e91'),
    '1': ObjectId('691d8d3221e7e5821d083e92'),
    '2': ObjectId('691d8d3221e7e5821d083e93'),
    '3': ObjectId('691d8d3221e7e5821d083e94'),
    '4': ObjectId('691d8d3221e7e5821d083e95')
  }
}
```

```
> db.utilisateurs.find().pretty()
< {
  _id: ObjectId('691d8b4221e7e5821d083e88'),
  nom: 'Ali',
  email: 'ali@test.com',
  age: 23,
  ville: 'Tanger',
  hobbies: [
    'football',
    'gaming'
  ]
}
{
  _id: ObjectId('691d8b4221e7e5821d083e89'),
  nom: 'Sara',
  email: 'sara@test.com',
  age: 29,
  ville: 'Rabat',
  hobbies: [
    'lecture',
    'voyage'
  ]
}
```

```
> db.produits.find().pretty()
< {
  _id: ObjectId('6918c643bde91ea0a2a32d29'),
  nom: 'Sac',
  categorie: 'Accessoires',
  fournisseur: 'Fournisseur T',
  prix: 140.62,
  stock: 187,
  avis: [
    'Qualité moyenne',
    'Excellent produit',
    'Je recommande'
  ],
  description: 'Produit pratique et solide'
}
{
  _id: ObjectId('6918c643bde91ea0a2a32d2a'),
  nom: 'Bande dessinée',
  categorie: 'Livres',
  fournisseur: 'Fournisseur P',
  prix: 104.14,
  stock: 178,
```

```

> db.commandes.find().pretty()
< {
  _id: ObjectId('691d8d3221e7e5821d083e91'),
  idUtilisateur: ObjectId('691d8b4221e7e5821d083e88'),
  idProduit: ObjectId('691d8b8921e7e5821d083e8d'),
  quantite: 1,
  dateCommande: 2024-02-01T00:00:00.000Z
}
{
  _id: ObjectId('691d8d3221e7e5821d083e92'),
  idUtilisateur: ObjectId('691d8b4221e7e5821d083e88'),
  idProduit: ObjectId('691d8b8921e7e5821d083e8e'),
  quantite: 2,
  dateCommande: 2024-02-10T00:00:00.000Z
}
{
  _id: ObjectId('691d8d3221e7e5821d083e93'),
  idUtilisateur: ObjectId('691d8b4221e7e5821d083e89'),
  idProduit: ObjectId('691d8b8921e7e5821d083e8e'),
  quantite: 1,
  dateCommande: 2024-03-01T00:00:00.000Z
}

```

## PARTIE 3 — TD Python : Toutes les compétences étape par étape

### Exercice 1 — Se connecter à MongoDB en Python

```

atelier6.py X
atelier6.py > ...
1  from pymongo import MongoClient
2
3  client = MongoClient("mongodb://localhost:27017/")
4  db = client["atelierPythonMongo"]
5
6  print("Base connectée !")
7

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + v [

```

PS C:\Users\LENOVO\Desktop\2eme annee full1203\Tps\public\pdfs> python atelier6.py
Base connectée !
PS C:\Users\LENOVO\Desktop\2eme annee full1203\Tps\public\pdfs>

```

### Exercice 2 — Afficher les collections :

```
collection_produits = db["produits"]

print("Collection 'produits' créée !")
```

Collection 'produits' créée !

### Exercice 3 — Lire les utilisateurs

```
✓ db.utilisateurs.insert_many([
    {"nom": "Ali", "email": "ali@test.com", "age": 23, "ville": "Tanger", "hobbies": ["football", "gaming"]},
    {"nom": "Sara", "email": "sara@test.com", "age": 29, "ville": "Rabat", "hobbies": ["lecture", "voyage"]},
    {"nom": "Nadia", "email": "nadia@test.com", "age": 35, "ville": "Casa", "hobbies": ["cuisine"]},
    {"nom": "Yassine", "email": "yassine@test.com", "age": 27, "ville": "Fès", "hobbies": ["fitness"]},
    {"nom": "Hassan", "email": "hassan@test.com", "age": 31, "ville": "Tanger", "hobbies": ["lecture"]}
])

print("Utilisateurs ajoutés !")
✓ for utilisateur in db.utilisateurs.find():
    print("ID :", utilisateur["_id"])
    print("Nom :", utilisateur["nom"])
    print("Email :", utilisateur["email"])
    print("Âge :", utilisateur["age"])
    print("Ville :", utilisateur["ville"])
    print("Hobbies :", utilisateur["hobbies"])
    print("-----")
```

```
Utilisateurs ajoutés !
ID : 691da1c248d8056c56bb9492
Nom : Ali
Email : ali@test.com
Âge : 23
Ville : Tanger
Hobbies : ['football', 'gaming']
-----
ID : 691da1c248d8056c56bb9493
Nom : Sara
Email : sara@test.com
Âge : 29
Ville : Rabat
Hobbies : ['lecture', 'voyage']
-----
ID : 691da1c248d8056c56bb9494
Nom : Nadia
Email : nadia@test.com
Âge : 35
Ville : Casa
Hobbies : ['cuisine']
-----
ID : 691da1c248d8056c56bb9495
Nom : Yassine
Email : yassine@test.com
Âge : 27
```

### Exercice 4 — Insérer un nouvel utilisateur via Python

```
db.utilisateurs.insert_one({
    "nom": "Karim",
    "email": "karim@test.com",
    "age": 26,
    "ville": "Agadir",
    "hobbies": ["surf", "musique"]
})

print("Utilisateur Karim ajouté !")
```

Utilisateur Karim ajouté !

#### Exercice 5 — Filtrage (find)

```
print("Utilisateurs de plus de 28 ans :")
for utilisateur in db.utilisateurs.find({"age": {"$gt": 28}}):
    print("Nom :", utilisateur["nom"], "| Âge :", utilisateur["age"])
```

```
Utilisateurs de plus de 28 ans :
Nom : Sara | Âge : 29
Nom : Nadia | Âge : 35
Nom : Hassan | Âge : 31
```

#### Exercice 6 — Mise à jour

-Modifier l'âge d'Ali :

```
db.utilisateurs.update_one(
    {"nom": "Ali"},
    {"$set": {"age": 24}}
)
print("Âge de Ali modifié à 24 !")
```

Âge de Ali modifié à 24 !

-Ajouter un hobby :

```
db.utilisateurs.update_one(
    {"nom": "Ali"},
    {"$push": {"hobbies": "cinéma"}}
)
print("Hobby 'cinéma' ajouté à Ali !")
```

Hobby 'cinéma' ajouté à Ali !

#### Exercice 7 — Supprimer

Supprimer Hassan :



```
db.utilisateurs.delete_one({"nom": "Hassan"})
print("Utilisateur Hassan supprimé !")
```

```
Utilisateur Hassan supprimé !
```

#### Exercice 8 — Index

```
db.utilisateurs.create_index([("email", 1)], unique=True)
print("Index unique créé !")
```

```
Index unique créé !
```

#### Exercice 9 — Agrégation (statistiques)

```
pipeline = [
    {
        "$group": {
            "_id": "$ville",
            "age_moyen": {"$avg": "$age"}
        }
    },
    {
        "$sort": {"age_moyen": -1}
    }
]

print("Âge moyen par ville :")
for doc in db.utilisateurs.aggregate(pipeline):
    print("Ville :", doc["_id"], "| Âge moyen :", round(doc["age_moyen"], 2))
```

```
Âge moyen par ville :
Ville : Casa | Âge moyen : 35.0
Ville : Rabat | Âge moyen : 29.0
Ville : Tanger | Âge moyen : 27.5
Ville : Fès | Âge moyen : 27.0
Ville : Agadir | Âge moyen : 26.0
```

#### Exercice 10 — Jointure : afficher commandes + utilisateurs + produits

```

pipeline = [
    {
        "$lookup": {
            "from": "utilisateurs",
            "localField": "idUserUtilisateur",
            "foreignField": "_id",
            "as": "detailsUtilisateur"
        }
    },
    {
        "$lookup": {
            "from": "produits",
            "localField": "idProduit",
            "foreignField": "_id",
            "as": "detailsProduit"
        }
    }
]
for commande in db.commandes.aggregate(pipeline):
    print("Commande ID :", commande["_id"])
    user = commande["detailsUtilisateur"][0]
    print("Utilisateur :", user["nom"], "| Email :", user["email"], "| Ville :", user["ville"])

    produit = commande["detailsProduit"][0]
    print("Produit :", produit["nom"], "| Prix :", produit["prix"], "| Stock :", produit["stock"])
    print("Quantité commandée :", commande["quantite"])
    print("Date de commande :", commande["dateCommande"])
    print("-----")

```