

Rapport : Avancée des Données dans un Système de Vente en Ligne avec MongoDB

Module : Base de Données MongoDB

Nom: Aya ouazzani akkar hammoudi

SOMMAIRE :

1. **Introduction**
2. **Créer des index pour améliorer les performances des requêtes de recherche.**
3. **Utiliser des vues pour simplifier l'accès aux données complexes.**
4. **Gérer la sécurité des données en créant des utilisateurs avec des rôles spécifiques.**
5. **Exporter les données dans un format JSON pour un partage et un traitement ultérieurs.**

1. Introduction

Dans le cadre de ce TP3, nous avons travaillé sur des fonctionnalités avancées de MongoDB appliquées à un système de vente en ligne.

Après avoir manipulé des requêtes et agrégations complexes dans le TP2, ce TP vise à approfondir les notions de performance, sécurité et structuration des données grâce à :

- La création d'index textuels pour accélérer les recherches.
- La mise en place de vues pour simplifier l'accès aux données.
- La gestion des utilisateurs et des rôles pour sécuriser la base.
- L'exportation de données pour la sauvegarde ou le partage.

2. Ajout d'un Champ Description et Création d'un Index Textuel

2.1 Ajout du champ "description"

Nous avons ajouté un champ description à chaque produit de la collection **produits**.

-Exemple d'ajout :

```
> db.produits.updateMany(  
  {},  
  { $set: { description: "Produit pratique et solide" } }  
)
```

```
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 100,
    modifiedCount: 100,
    upsertedCount: 0
}
```

2.2 Création d'un index textuel

Pour optimiser les recherches, nous avons créé un index textuel sur le champ description :

```
gestion_produits> db.produits.createIndex({ description: "text" });
```

```
> db.produits.getIndexes();
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'description_text',
    weights: { description: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
```

2.3 Recherche textuelle

Une fois l'index créé, nous pouvons utiliser \$text :

```
test> db.produits.find({ $text: { $search: "pratique solide" } }).
```

```
_id: ObjectId('6918c643bde91ea0a2a32d8c'),
nom: 'Pull',
categorie: 'Vêtements',
fournisseur: 'Fournisseur N',
prix: 47.39,
stock: 392,
avis: [
  'Livraison rapide',
  'Bon rapport qualité-prix'
],
description: 'Produit pratique et solide'
}
{
  _id: ObjectId('6918c643bde91ea0a2a32d8b'),
  nom: 'Manuel',
  categorie: 'Livres',
  fournisseur: 'Fournisseur L',
  prix: 75.56,
  stock: 40,
  avis: [
    'Livraison rapide',
    'Un peu cher'
  ]
}
```

```
  ],
  description: 'Produit pratique et solide'
}
{
  _id: ObjectId('6918c643bde91ea0a2a32d8a'),
  nom: 'Tablette',
  categorie: 'Électronique',
  fournisseur: 'Fournisseur T',
  prix: 27.07,
  stock: 227,
  avis: [
    'Conforme à la description',
    'Qualité moyenne'
  ],
  description: 'Produit pratique et solide'
}
{
  _id: ObjectId('6918c643bde91ea0a2a32d89'),
  nom: 'Café',
  categorie: 'Alimentation',
  fournisseur: 'Fournisseur M',
  prix: 6.83,
  stock: 40,
```

```

  ],
  description: 'Produit pratique et solide'
}
{
  _id: ObjectId('6918c643bde91ea0a2a32d79'),
  nom: 'Baskets',
  categorie: 'Chaussures',
  fournisseur: 'Fournisseur O',
  prix: 164.75,
  stock: 482,
  avis: [
    'Livraison rapide',
    'Solide'
  ],
  description: 'Produit pratique et solide'
```

3. Création de Vues (Views)

3.1 Vue : Produits avec un stock inférieur à 20

```
test> db.createView(  
    "produits_stockFaible",  
    "produits",  
    [  
        { $match: { stock: { $lt: 20 } } },  
        { $project: { nom: 1, stock: 1, prix: 1 } }  
    ]  
);|
```

```
< { ok: 1 }
```

```
test> db.produits_stockFaible.find();|
```

```
< {  
    _id: ObjectId('6918c643bde91ea0a2a32d31'),  
    nom: 'Mocassins',  
    prix: 197.04,  
    stock: 14  
}  
{  
    _id: ObjectId('6918c643bde91ea0a2a32d4a'),  
    nom: 'Sandales',  
    prix: 150.47,  
    stock: 18  
}  
{  
    _id: ObjectId('6918c643bde91ea0a2a32d7b'),  
    nom: 'Sandales',  
    prix: 146.99,  
    stock: 15  
}
```

3.2 Vue : Nombre de produits par catégorie

```
test> db.createView(  
    "nombreProduits_parCategorie",  
    "produits",  
    [  
        {  
            $group: {  
                _id: "$categorie",  
                nombreProduits: { $sum: 1 }  
            }  
        }  
    ]  
);|
```

```
< { ok: 1 }
```

```
test> db.nombreProduits_parCategorie.find();
```

```
< [
      {
        _id: 'Électronique',
        nombreProduits: 14
      },
      {
        _id: 'Chaussures',
        nombreProduits: 13
      },
      {
        _id: 'Jouets',
        nombreProduits: 7
      },
      {
        _id: 'Accessoires',
        nombreProduits: 8
      },
      {
        _id: 'Livres',
        nombreProduits: 13
      },
      {
        _id: 'Beauté',
        nombreProduits: 7
      },
      {
        _id: 'Maison',
        nombreProduits: 13
      },
      {
        _id: 'Alimentation',
        nombreProduits: 12
      },
      {
        _id: 'Vêtements',
        nombreProduits: 6
      },
      {
        _id: 'Sport',
        nombreProduits: 7
      }
    ]
```

3.3 Vue : Analyse des ventes par fournisseur

```
test > db.createView(
    "ventes_parFournisseur",
    "produits",
    [
        {
            $lookup: {
                from: "fournisseurs",
                localField: "fournisseur",
                foreignField: "nom",
                as: "detailsFournisseur"
            }
        },
        {
            $group: {
                _id: "$fournisseur",
                totalVentes: { $sum: { $multiply: ["$prix", "$stock"] } }
            }
        }
    ]
);
< { ok: 1 }
```

```
test > db.ventes_parFournisseur.find();
```

<pre>_id: 'Fournisseur G', totalVentes: 197863.16999999998 } { _id: 'Fournisseur K', totalVentes: 79470 } { _id: 'Fournisseur T', totalVentes: 64304.07 } { _id: 'Fournisseur P', totalVentes: 78198.02 } { _id: 'Fournisseur B', totalVentes: 135159.86 } { _id: 'Fournisseur O', totalVentes: 288949.12 }</pre>	<pre>_id: 'Fournisseur M', totalVentes: 121736.65 } { _id: 'Fournisseur H', totalVentes: 219990.5 } { _id: 'Fournisseur L', totalVentes: 60953.43 } { _id: 'Fournisseur C', totalVentes: 136409.88 } { _id: 'Fournisseur E', totalVentes: 43480.58 } { _id: 'Fournisseur A', totalVentes: 146073.68 }</pre>	<pre>_id: 'Fournisseur R', totalVentes: 236024.88 } { _id: 'Fournisseur S', totalVentes: 37242.81 } { _id: 'Fournisseur J', totalVentes: 131734.91999999998 } { _id: 'Fournisseur N', totalVentes: 122039.42 } { _id: 'Fournisseur Q', totalVentes: 95439.84 } { _id: 'Fournisseur F', totalVentes: 105604.12 }</pre>
---	---	---

4. Création d'un Utilisateur avec Rôle Limité

```
test> db.createUser({  
    user: "analyste",  
    pwd: "motdepasseSécurisé",  
    roles: [{ role: "read", db: "gestion_produits" }]  
});|
```

```
< { ok: 1 }
```

5. Exportation des Données au Format JSON:

Exportation de la collection produits :

```
C:\mongodb-database-tools-windows-x86_64-100.13.0\bin>mongoexport --db gestion_produits --collection produits --out produits.json  
2025-11-16T19:10:35.465+0100      connected to: mongodb://localhost/  
2025-11-16T19:10:35.615+0100      exported 100 records  
C:\mongodb-database-tools-windows-x86_64-100.13.0\bin>
```

Exportation de la collection fournisseurs :

```
C:\mongodb-database-tools-windows-x86_64-100.13.0\bin>mongoexport --db gestion_produits --collection fournisseurs --out fournisseurs.json  
2025-11-16T19:14:16.252+0100      connected to: mongodb://localhost/  
2025-11-16T19:14:16.327+0100      exported 20 records
```

6. Conclusion

Ce TP nous a permis d'approfondir des fonctionnalités avancées de MongoDB essentielles pour des applications professionnelles :

- **Index textuels** pour optimiser les recherches.
- **Vues** pour simplifier l'analyse et éviter la duplication.
- **Gestion des utilisateurs** pour renforcer la sécurité.
- **Exportation JSON** pour la portabilité des données.