

# Rapport : Requêtes Avancées et Agrégations dans un Système de Vente en Ligne avec MongoDB

**Module :** Base de Données MongoDB

**Nom:** Aya ouazzani akkar hammoudi

**SOMMAIRE :**

**1. Introduction:**

**2 : Objectifs du TP:**

**.3 Base de données utilisée:**

**4 : Requêtes d'Agrégations:**

**5 : Conclusion:**

## 1. Introduction:

2. Ce travail pratique (TP2) s'inscrit dans la continuité du TP1 consacré à la manipulation de données dans MongoDB.  
L'objectif principal est d'explorer les fonctionnalités avancées d'agrégation afin d'extraire des informations analytiques à partir des collections de produits, catégories et fournisseurs.
3. Le pipeline d'agrégation de MongoDB permet d'appliquer plusieurs transformations successives sur les documents : filtrage, regroupement, calculs, tri, jointure, etc.

## 2 : Objectifs du TP:

1-Comprendre le fonctionnement du pipeline aggregate().

2-Utiliser les opérateurs d'agrégation tels que :

\$match, \$group, \$project, \$sort, \$limit, \$sum, \$avg, \$lookup.

3-Extraire des analyses avancées sur les produits d'un système de vente en ligne.

4-Produire un rapport détaillé présentant les requêtes et les résultats obtenus.

## 3 :Base de données utilisée:

Base : **gestion\_products**

Collections :

- produits
- categories
- fournisseurs

## 4 : Requêtes d'Agrégations:

### 4.1. Calculer le Chiffre d'Affaires Total par Produit:

```
>_MONGOSH
test> db.produits.aggregate([
  {
    $group: {
      _id: "$nom",
      chiffreAffaires: { $sum: { $multiply: ["$prix", "$stock"] } }
    }
  }
]);
```

#### Explication

- \$group : regroupe les documents par nom du produit.
- \$multiply : multiplie prix × stock.
- \$sum : additionne le résultat (utile si plusieurs lignes pour le même produit)

```
{
  _id: 'Manuel',
  chiffreAffaires: 56998.05
< {
  _id: 'Chaussures de sport',
  chiffreAffaires: 61736.57
}
{
  _id: 'Magazine',
  chiffreAffaires: 41437.26
}
{
  _id: 'Bottes',
  chiffreAffaires: 177521.13
}
{
  _id: 'Raquette',
  chiffreAffaires: 35020.89
}
{
  _id: 'Rideau',
  chiffreAffaires: 121231.54
}
```

### 4.2. Trier les Produits par Chiffre d'Affaires Décroissant

```
test> db.produits.aggregate([
  {
    $group: {
      _id: "$nom",
      chiffreAffaires: { $sum: { $multiply: ["$prix", "$stock"] } }
    }
  },
  { $sort: { chiffreAffaires: -1 } },
  { $limit: 5 }
]);
```

### Explication

- \$sort : tri décroissant.
- \$limit : affiche uniquement les 5 produits les plus rentables.

```
< [
  {
    _id: 'Bottes',
    chiffreAffaires: 177521.13
  }
  {
    _id: 'Bande dessinée',
    chiffreAffaires: 170095.54
  }
  {
    _id: 'Coussin',
    chiffreAffaires: 152149.01
  }
  {
    _id: 'Table',
    chiffreAffaires: 127016.2399999999
  }
  {
    _id: 'Rideau',
    chiffreAffaires: 121231.54
  }
]
```

#### 4.3. Calculer le Prix Moyen par Catégorie

```
test> db.produits.aggregate([
  {
    $group: {
      _id: "$categorie",
      prixMoyen: { $avg: "$prix" }
    }
  }
]);
```

#### Explication

- **\$group par catégorie.**
- **\$avg calcule la moyenne des prix.**

```
< [
    {
        _id: 'Alimentation',
        prixMoyen: 105.3108333333333
    }
    {
        _id: 'Livres',
        prixMoyen: 98.45142857142856
    }
    {
        _id: 'Accessoires',
        prixMoyen: 96.29444444444444
    }
    {
        _id: 'Jouets',
        prixMoyen: 96.72625
    }
    {
        _id: 'Beauté',
        prixMoyen: 94.01142857142858
    }
]
```

```
{
  {
    _id: 'Maison',
    prixMoyen: 119.83615384615383
  }
  {
    _id: 'Chaussures',
    prixMoyen: 114.19923076923077
  }
  {
    _id: 'Vêtements',
    prixMoyen: 77.11
  }
  {
    _id: 'Sport',
    prixMoyen: 112.19285714285715
  }
  {
    _id: 'Électronique',
    prixMoyen: 88.31500000000001
  }
}
```

#### 4.4. Produits avec Stock Faible et Prix Élevé

##### Condition

Stock < 50 et prix > 30.

```
test> db.produits.aggregate([
  {
    $match: { stock: { $lt: 50 }, prix: { $gt: 30 } }
  },
  {
    $sort: { prix: 1 }
  }
]);
```

##### Explication

- **\$match : filtre selon conditions.**
- **\$sort: tri croissant par prix**

```
< [
    {
        _id: ObjectId('6917a86ff865b0826df658f8'),
        nom: 'Manuel',
        categorie: 'Livres',
        fournisseur: 'Fournisseur L',
        prix: 75.56,
        stock: 40,
        avis: [
            'Livraison rapide',
            'Un peu cher'
        ]
    }
    {
        _id: ObjectId('6917a86ff865b0826df658e8'),
        nom: 'Sandales',
        categorie: 'Chaussures',
        fournisseur: 'Fournisseur E',
        prix: 146.99,
        stock: 15,
        avis: [
            'Je recommande',
            'Solide',
        ]
    }
]
```

```
{
  _id: ObjectId('6917a86ff865b0826df658b7'),
  nom: 'Sandales',
  categorie: 'Chaussures',
  fournisseur: 'Fournisseur A',
  prix: 150.47,
  stock: 18,
  avis: [
    'Solide',
    'Très pratique',
    'Conforme à la description'
  ]
}

{
  _id: ObjectId('6917a86ff865b0826df658d2'),
  nom: 'Biscuits',
  categorie: 'Alimentation',
  fournisseur: 'Fournisseur P',
  prix: 173.43,
  stock: 48,
  avis: [
    'Livraison rapide',
  ]
}
```

#### 4.5. Compter les Produits par Fournisseur

```
gestion_produits> db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      nombreProduits: { $sum: 1 }
    }
  }
]);
```

#### Explication

- **\$sum: 1** : compte le nombre de documents de chaque fournisseur.

```
\`1
{
  _id: 'Fournisseur L',
  nombreProduits: 5
}
{
  _id: 'Fournisseur M',
  nombreProduits: 6
}
{
  _id: 'Fournisseur P',
  nombreProduits: 8
}
{
  _id: 'Fournisseur T',
  nombreProduits: 5
}
{
  _id: 'Fournisseur R',
  nombreProduits: 7
}
{
  _id: 'Fournisseur B',
  nombreProduits: 7
```

```

{
  _id: 'Fournisseur 0',
  nombreProduits: 6
}
{
  _id: 'Fournisseur K',
  nombreProduits: 3
}
{
  _id: 'Fournisseur C',
  nombreProduits: 6
}
{
  _id: 'Fournisseur E',
  nombreProduits: 4
}
{
  _id: 'Fournisseur H',
  nombreProduits: 5
}
{
  _id: 'Fournisseur I',
  nombreProduits: 3
}

```

#### 4.6.Calculer la Quantité Totale en Stock par Fournisseur

```

gestion_produits> db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      stockTotal: { $sum: "$stock" }
    }
  }
]);

```

#### Explication

- **\$sum** additionne le stock total fourni par chaque fournisseur.

```
        {
            _id: 'Fournisseur N',
            stockTotal: 1289
        }
    {
        _id: 'Fournisseur H',
        stockTotal: 1515
    }
    {
        _id: 'Fournisseur I',
        stockTotal: 1042
    }
    {
        _id: 'Fournisseur T',
        stockTotal: 1132
    }
    {
        _id: 'Fournisseur P',
        stockTotal: 1113
    }
    {
        _id: 'Fournisseur A',
        stockTotal: 1090
    }
}
{
    _id: 'Fournisseur D',
    stockTotal: 515
}
{
    _id: 'Fournisseur C',
    stockTotal: 1430
}
{
    _id: 'Fournisseur K',
    stockTotal: 793
}
{
    _id: 'Fournisseur L',
    stockTotal: 733
}
{
    _id: 'Fournisseur E',
    stockTotal: 896
}
{
    _id: 'Fournisseur G',
    stockTotal: 1645
}
```

```
{
  {
    _id: 'Fournisseur B',
    stockTotal: 1368
  }
  {
    _id: 'Fournisseur F',
    stockTotal: 1047
  }
  {
    _id: 'Fournisseur J',
    stockTotal: 1714
  }
  {
    _id: 'Fournisseur Q',
    stockTotal: 1359
  }
  {
    _id: 'Fournisseur S',
    stockTotal: 659
  }
}
```

#### 4.7 Afficher les Catégories et le Nombre de Produits par Catégorie

```
gestion_produits> db.produits.aggregate([
  {
    $group: {
      _id: "$categorie",
      nombreProduits: { $sum: 1 }
    }
  },
  {
    $match: {
      nombreProduits: { $gte: 2 }
    }
  }
]);
```

#### Explication

- **\$group : compte produits par catégorie.**
- **\$match : affiche seulement catégories avec au moins 2 produits.**

```

        {
            _id: 'Électronique',
            nombreProduits: 14
        }
    {
        _id: 'Chaussures',
        nombreProduits: 13
    }
{
    _id: 'Maison',
    nombreProduits: 13
}
{
    _id: 'Jouets',
    nombreProduits: 8
}
{
    _id: 'Beauté',
    nombreProduits: 7
}
{
    _id: 'Accessoires',
    nombreProduits: 9
}
{
    _id: 'Sport',
    nombreProduits: 7
}
{
    _id: 'Livres',
    nombreProduits: 14
}
{
    _id: 'Vêtements',
    nombreProduits: 6
}
{
    _id: 'Alimentation',
    nombreProduits: 12
}

```

#### 4.8 Jointure entre Produits et Fournisseurs avec \$lookup

```

gestion_produits> db.produits.aggregate([
    {
        $lookup: {
            from: "fournisseurs",
            localField: "fournisseur",
            foreignField: "nom",
            as: "detailsFournisseur"
        }
    }
]);

```

#### Explication

- **\$lookup** : jointure entre les produits et leurs fournisseurs.
- **as** : crée un tableau "detailsFournisseur" contenant les infos du fournisseur.

```
< {
    _id: ObjectId('6917a814f865b0826df65893'),
    nom: 'Sac',
    categorie: 'Accessoires',
    fournisseur: 'Fournisseur T',
    prix: 140.62,
    stock: 187,
    avis: [
        'Qualité moyenne',
        'Excellent produit',
        'Je recommande'
    ],
    detailsFournisseur: [
        {
            _id: ObjectId('69144bf898e6c1b60dc85b62'),
            nom: 'Fournisseur T',
            contact: 'contactT@example.com'
        }
    ]
}

{
    _id: ObjectId('6917a814f865b0826df65894'),
    nom: 'Bande dessinée',
    categorie: 'Livres',
    fournisseur: 'Fournisseur P',
    prix: 104.14,
    stock: 178,
    avis: [
        'Livraison rapide',
        'Excellent produit',
        'Bon rapport qualité-prix',
        'Design sympa'
    ],
    detailsFournisseur: [
        {
            _id: ObjectId('69144bf898e6c1b60dc85b5e'),
            nom: 'Fournisseur P',
            contact: 'contactP@example.com'
        }
    ]
}
```

```
{  
    _id: ObjectId('6917a86ff865b0826df658a6'),  
    nom: 'Clavier',  
    categorie: 'Électronique',  
    fournisseur: 'Fournisseur D',  
    prix: 152.74,  
    stock: 143,  
    avis: [  
        'Un peu cher',  
        'Qualité moyenne'  
    ],  
    detailsFournisseur: [  
        {  
            _id: ObjectId('69144bf898e6c1b60dc85b52'),  
            nom: 'Fournisseur D',  
            contact: 'contactD@example.com'  
        }  
    ]  
}
```

## 5 : Conclusion:

Dans ce TP, nous avons exploré les principales fonctionnalités du pipeline d'agrégation de MongoDB. Les différentes requêtes réalisées montrent la puissance de MongoDB pour :

- analyser les ventes,
- regrouper les données par catégories ou fournisseurs,
- filtrer selon plusieurs conditions,
- effectuer des calculs analytiques,
- réaliser des jointures entre collections.