

TP : Agrégations MongoDB

Préparation de la base de données

Insertion des clients

```
db.clients.insertMany([
  { _id: 1, nom: "Ibn Battuta", pays_id: 101 },
  { _id: 2, nom: "Tariq Ibn Ziyad", pays_id: 102 },
  { _id: 3, nom: "Fatima al-Fihriya", pays_id: 101 },
  { _id: 4, nom: "Ahmed al-Mansour", pays_id: 101 },
  { _id: 5, nom: "Abdelkrim al-Khattabi", pays_id: 101 }
])
```

Insertion des pays natale

```
db.pays.insertMany([
  { _id: 101, nom: "Maroc", continent: "Afrique" },
  { _id: 102, nom: "Espagne", continent: "Europe" },
  { _id: 103, nom: "Algérie", continent: "Afrique" }
])
```

Insertion des commandes

```
db.commandes.insertMany([
  { _id: 1, client_id: 1, produit_id: 201, date: new Date("2024-01-15") },
  { _id: 2, client_id: 2, produit_id: 202, date: new Date("2024-02-10") },
  { _id: 3, client_id: 3, produit_id: 203, date: new Date("2024-02-15") },
  { _id: 4, client_id: 4, produit_id: 204, date: new Date("2024-03-05") },
  { _id: 5, client_id: 5, produit_id: 201, date: new Date("2024-03-20") }
])
```

Insertion des produits

```
db.produits.insertMany([
  { _id: 201, nom: "Tapis berbère", origine: "Atlas", prix: 1500 },
  { _id: 202, nom: "Huile d'argan", origine: "Souss", prix: 300 },
  { _id: 203, nom: "Thé vert à la menthe", origine: "Nord", prix: 50 },
  { _id: 204, nom: "Babouches artisanales", origine: "Fès", prix: 200 }
])
```

Partie 1 : Requêtes entre clients et pays natale

1. Liste des clients avec leur pays natale

Affichez les noms des clients et leur pays d'origine.

```
db.clients.aggregate([
  { $lookup: {
    from: "pays",
    localField: "pays_id",
    foreignField: "_id",
    as: "details_pays"
  }},
  { $project: {
    nom: 1,
    "details_pays.nom": 1,
    _id: 0
  }}
])
```

2. Clients africains uniquement

Affichez les noms des clients originaires du continent africain.

```
db.clients.aggregate([
  { $lookup: {
    from: "pays",
    localField: "pays_id",
    foreignField: "_id",
    as: "details_pays"
  }},
  { $match: { "details_pays.continent": "Afrique" }},
  { $project: { nom: 1, _id: 0 }}
])
```

3. Nombre de clients par continent

Calculez combien de clients sont originaires de chaque continent.

```
db.clients.aggregate([
  { $lookup: {
    from: "pays",
    localField: "pays_id",
    foreignField: "_id",
    as: "details_pays"
  }},
  { $group: {
    _id: "$details_pays.continent",
    nombre_clients: { $sum: 1 }
  }}
])
```

```
    }}  
  ])
```

Partie 2 : Requêtes entre clients, commandes et produits

1. Liste des produits achetés par chaque client

Affichez les noms des clients et les produits qu'ils ont achetés.

```
db.commandes.aggregate([  
  { $lookup: {  
    from: "clients",  
    localField: "client_id",  
    foreignField: "_id",  
    as: "details_clients"  
  }},  
  { $lookup: {  
    from: "produits",  
    localField: "produit_id",  
    foreignField: "_id",  
    as: "details_produits"  
  }},  
  { $project: {  
    "details_clients.nom": 1,  
    "details_produits.nom": 1,  
    _id: 0  
  }}  
])
```

2. Afficher les noms des clients et leurs commandes

Affichez les noms des clients et les produits qu'ils ont achetés.

```
db.commandes.aggregate([  
  { $lookup: {  
    from: "clients",  
    localField: "client_id",  
    foreignField: "_id",  
    as: "details_clients"  
  }},  
  { $project: {  
    "details_clients.nom": 1,  
    date: 1,  
    _id: 0  
  }}  
])
```

3. Lister les produits commandés avec leurs détails

Affichez les noms des clients et les produits qu'ils ont achetés.

```
db.commandes.aggregate([
  { $lookup: {
    from: "produits",
    localField: "produit_id",
    foreignField: "_id",
    as: "details_produits"
  }},
  { $project: {
    produit_id: 1,
    "details_produits.nom": 1,
    "details_produits.prix": 1,
    _id: 0
  }}
])
```

4. Total des dépenses par client

Calculez le montant total dépensé par chaque client.

```
db.commandes.aggregate([
  { $lookup: {
    from: "clients",
    localField: "client_id",
    foreignField: "_id",
    as: "details_clients"
  }},
  { $lookup: {
    from: "produits",
    localField: "produit_id",
    foreignField: "_id",
    as: "details_produits"
  }},
  { $group: {
    _id: "$details_clients.nom",
    total_depenses: { $sum: { $arrayElemAt: ["$details_produits.prix", 0] } }
  }}
])
```

5. Nombre de commandes par client

Identifiez les produits les plus achetés, triés par leur popularité.

```
db.commandes.aggregate([
  { $group: {
    _id: "$client_id",
    nombre_commandes: { $sum: 1 }
  }}
])
```

6. Total des quantités commandées par produit

Identifiez les produits les plus achetés, triés par leur popularité.

```
db.commandes.aggregate([
  { $group: {
    _id: "$produit_id",
    total_quantite: { $sum: "$quantite" }
  }}
])
```

7. Produits les plus populaires

Identifiez les produits les plus achetés, triés par leur popularité.

```
db.commandes.aggregate([
  { $group: {
    _id: "$produit_id",
    nombre_achats: { $sum: 1 }
  }},
  { $lookup: {
    from: "produits",
    localField: "_id",
    foreignField: "_id",
    as: "details_produits"
  }},
  { $project: {
    "details_produits.nom": 1,
    nombre_achats: 1,
    _id: 0
  }},
  { $sort: { nombre_achats: -1 } }
])
```

Partie 3 : Exercice d'application

Objectif

Dans cette partie, vous travaillerez sur une base de données fictive dédiée à une entreprise de logistique, avec les collections suivantes : **livraisons**, **clients**, **véhicules**, et **chauffeurs**. Votre tâche est d'écrire les requêtes MongoDB nécessaires pour répondre aux questions ci-dessous.

Collections disponibles

- **livraisons** : Contient les informations sur les livraisons (client_id, chauffeur_id, véhicule_id, date_livraison, poids_colis).
- **clients** : Contient les informations sur les clients (nom, ville).

- ****véhicules**** : Contient les informations sur les véhicules (marque, capacité_max).
- ****chauffeurs**** : Contient les informations sur les chauffeurs (nom, ancienneté, type_permis).

Questions

1. Affichez la liste des livraisons avec les noms des clients et des chauffeurs associés.
2. Identifiez les chauffeurs ayant transporté plus de 500 kg au total.
3. Trouvez les véhicules ayant été utilisés pour le plus grand nombre de livraisons.
4. Calculez la moyenne des poids des colis livrés par chaque chauffeur.
5. Affichez les villes ayant reçu le plus de livraisons, triées par ordre décroissant.