

TP n2 en Big Data

Framework d'agrégation en MongoDB

Le framework d'agrégation comporte un ensemble large d'opérateurs :

- \$group : regroupements et calculs à effectuer.
 - \$project : création de nouvelle variable
 - \$match : restriction sur les documents à utiliser.
 - \$sort : tri sur les agrégats.
 - \$unwind : découpage de tableaux, permet d'éclater le tableau en autant de documents
 - Une requête d'agrégation est de la forme : `db.collection.aggregate([{ ... }, { ... }, ...])`.
- Fonctions d'agrégations en mongoDB: \$sum, \$max, \$min, \$count, ...

```
db.coll.aggregate([
  { $group: { _id: <expression>, // Group By Expression
    <field1>: { <accumulator1> : <expression1> },
    ... }
  }
])
```

L'opérateur \$group doit contenir obligatoirement une clé de groupement (_id), puis une clé (field) à laquelle on associe la fonction d'agrégation.

Mongodb

```
db.collection.aggregate([
  { $group: { _id: <expression>,
    nbre: { $sum: "$attribut" } }
})
db.livres.aggregate( [
  { $group: { _id: "null", count: { $count: { " " } } }
  }
])
db.emprunts.aggregate([
  { $group: { _id: "$noliv", nb_pret: { $sum: 1 } } },
  { $match: { "nb_pret": { $gt: 1 } } }
])
```

SQL

```
SELECT SUM(attribut) AS nbre
FROM collection
```

```
SELECT COUNT(*) AS count FROM sales
```

```
SELECT COUNT(*) AS nb_pret FROM
emprunts group by noliv HAVING nb_pret > 1
```

Exemple 1: Total des PU des livres par éditeur

```
db.livres.aggregate([ { $group: { "_id": "$Editeur", TotPU: { $sum: "$PU" } } } ])
```

Resultat: { _id: 'Hermann', TotPU: 540 },
{ _id: 'Dunod', TotPU: 30805 },
.....

Exemple 2: \$unwind

```
db.courses.insertOne({ "_id" : 1,
  "name" : "ali", "cours": [ "MongoDB", "Java", "Python" ] })
```

```
db.courses.aggregate([ { $unwind: "$cours" } ])
[
  { _id: 1, name: 'ali', cours: 'MongoDB' },
  { _id: 1, name: 'ali', cours: 'Java' },
  { _id: 1, name: 'ali', cours: 'Python' }
]
```

Exercice 1.

1. Se connecter à MongoDB via le shell : `C:\mongosh-1.10.6-win32-x64\bin\mongosh`
2. Afficher les bases de données existantes
3. Afficher la liste de collections de la base biblio
4. Compter le nombre de documents de la collection 'livres'
5. Compter le nombre de documents par Editeur
6. Rechercher dans la collection livres tous les documents (titre, et prix) qui ont un prix entre 200 et 400.
7. Compter le nombre de documents ayant PU > 500
8. Rechercher la valeur maximale puis minimale que prend la variable PU sur tous les documents
9. Afficher le nom de l'éditeur, le prix du livre le plus cher et le moins cher pour cet éditeur
10. Afficher le nom de l'éditeur, le prix du livre le plus cher et le moins cher pour cet éditeur, ceci pour les éditeurs ayant édité plus d'un livre.
11. Récupérer le premier document (titre, auteur et PU) dont le titre contient « dévelop ».
12. Créer dans la base de données biblio les deux collections « étudiants », et « emprunts »
13. Insérer dans la collection etudiants les données du fichier dataetudiant.json, et vérifier l'insertion
14. Insérer dans la collection emprunts les données suivantes, et vérifier l'insertion

```
{ noetud: 'E1', noliv: 2, datesortie: '15/01/2023', dateretour: " " },
{ noetud: 'E2', noliv: 1, datesortie: '17/01/2023', dateretour: '18/01/2023' },
{ noetud: 'E1', noliv: 2, datesortie: '20/01/2023', dateretour: " " },
{ noetud: 'E3', noliv: 4, datesortie: '15/01/2023', dateretour: " " },
{ noetud: 'E3', noliv: 5, datesortie: '15/01/2023', dateretour: '16/01/2023' },
{ noetud: 'E6', noliv: 6, datesortie: '16/01/2023', dateretour: " " },
{ noetud: 'E4', noliv: 7, datesortie: '14/01/2023', dateretour: '16/01/2023' },
{ noetud: 'E4', noliv: 12, datesortie: '20/01/2023', dateretour: " " },
{ noetud: 'E5', noliv: 7, datesortie: '22/01/2023', dateretour: " " }
```
15. Lister le nombre d'emprunt de chaque livre (noliv).
16. Sélectionner les livres ayant été empruntés plus qu'une seule fois.
17. Lister noliv, et date sortie des livres empruntés
18. Sélectionner les livres (codeLivre, et Titre) qui n'ont jamais empruntés.
19. Lister noliv, titre, et date sortie des livres empruntés
20. Trouver noliv, titre, des livres empruntés (non retourné), numéro et le nom de leur emprunteur.
21. Quels sont les noms et prénoms des étudiants qui ont emprunté (et non retourné) un livre après la date du 16/01/2010.
Insérer dans la collection emprunts, le document suivant. { noetud: 'E7', noliv: 10, datesortie: '16/02/2023', dateretour: " " },
22. Modifier les dates d'emprunt du livre 10 sorti le 16/02/2020 à la date de sortie 17/02/2020 et la date retour 18/02/2020, et vérifier la modification.
23. Supprimer dans la collection "emprunts" tous les livres de dateretour est nul, et vérifier la suppression.

Exercice 2. Soit les trois relations (Etudiant, Matiere, et Evaluer) relatives à la gestion des notes annuelles d'une promotion d'étudiants

1. Créer dans une nouvelle base de données les trois collections : etudiant, matiere, et evaluer
2. Quel est le nombre total d'étudiants ?
3. Quelles sont les moyennes par matière pour chaque étudiant ?
4. Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse pour chaque matière
5. Quelles sont les moyennes de chaque étudiant dans chacune des matières
6. Quelle est la moyenne générale de chaque étudiant ?
7. Quelle est la moyenne générale de la promotion ?

Netud	Nom		Codemat	Libelle	Coef		Netud	Codemat	NoteCC	NoteCF
A10	N1		Math	Mathématique	13		A11	Math	12	13
A11	N2		SI	Syst. Information	7		A11	SI	10	12
A12	N11		BD	Base Données	7		A11	BD	9	14
							A10	Math	14	11
							A10	SI	13	13
							A10	BD	11	10,5
							A12	Math	8	11
							A12	SI	14	15
							A12	BD	9	14