

# TP MongoDB - Gestion des employés

L. Ziad

EST-Essaouira

## Installation et configuration

- 1) Lancer la commande :

```
1 mongod
```

- 2) Télécharger et décompresser l'archive `employees.rar`

- 3) Dans la console, exécuter la commande :

```
1 mongorestore --db gescom cheminabsolutdufichier/employees.bson --port  
numeroPort
```

- 4) Se connecter à la base `gescom` avec la commande :

```
1 mongo --port numeroPort gescom
```

## Exercices

- 1)- Afficher toutes les collections de la base :

```
1 show collections
```

- 2)- Afficher tous les documents de la base :

```
1 db.getCollectionNames().forEach(function(collection) {  
2   print("Collection: " + collection);  
3   db[collection].find().forEach(printjson);  
4 });
```

- 3)- Compter le nombre de documents de la collection `employees` :

```
1 db.employees.countDocuments()
```

- 4)- Insérer de deux manières différentes deux employés :

**Méthode 1 :**

```
1 db.employees.insertOne({  
2   nom: "Dupont",  
3   prenom: "Pierre",  
4   prime: 500  
5 })
```

**Méthode 2 :**

```

1 db.employees.insertOne({
2     nom: "Martin",
3     prenom: "Sophie",
4     anciennete: 3
5 })

```

- 5)- Afficher la liste des employés dont le prénom est David :

```

1 db.employees.find({prenom: "David"})

```

- 6)- Afficher la liste des employés dont le prénom commence ou se termine par D :

```

1 db.employees.find({
2     $or: [
3         {prenom: /^D/},
4         {prenom: /D$/}
5     ]
6 })

```

- 7)- Afficher la liste des personnes dont le prénom commence par D et contient exactement 5 lettres :

```

1 db.employees.find({
2     prenom: /^D....$/
3 })

```

- 8)- Afficher la liste des personnes dont le prénom commence et se termine par une voyelle :

```

1 db.employees.find({
2     prenom: /^[aeiouAEIOU].*[aeiouAEIOU]$/
3 })

```

- 9)- Afficher la liste des personnes dont le prénom commence et se termine par une même lettre :

```

1 db.employees.find({
2     $expr: {
3         $eq: [
4             {$substrCP: ["$prenom", 0, 1]},
5             {$substrCP: ["$prenom", -1, 1]}
6         ]
7     }
8 })

```

- 10)- Afficher les nom et prénom de chaque employé ayant une ancienneté > 10 :

```

1 db.employees.find(
2     {anciennete: {$gt: 10}},
3     {nom: 1, prenom: 1, _id: 0}
4 )

```

- 11)- Afficher les nom et adresse complète des employés ayant un attribut rue dans l'objet adresse :

```

1 db.employees.find(
2     {"adresse.rue": {$exists: true}},
3     {nom: 1, adresse: 1, _id: 0}
4 )

```

- 12)- Incrémenter de 200 la prime des employés ayant déjà le champ prime :

```

1 db.employees.updateMany(
2   {prime: {$exists: true}}, 
3   {$inc: {prime: 200}}
4 )

```

- 13)- Afficher les trois premières personnes ayant la plus grande valeur d'ancienneté :

```

1 db.employees.find()
2   .sort({anciennete: -1})
3   .limit(3)

```

- 14)- Regrouper les personnes dont la ville de résidence est Toulouse (afficher nom, prénom et ancienneté) :

```

1 db.employees.find(
2   {"adresse.ville": "Toulouse"}, 
3   {nom: 1, prenom: 1, anciennete: 1, _id: 0}
4 )

```

- 15)- Afficher les personnes dont le prénom commence par M et la ville de résidence est soit Foix soit Bordeaux :

```

1 db.employees.find({
2   prenom: /^M/, 
3   "adresse.ville": {$in: ["Foix", "Bordeaux"]}}
4 )

```

- 16)- Mettre à jour l'adresse de Dominique Mani :

```

1 db.employees.updateOne(
2   {nom: "Mani", prenom: "Dominique"}, 
3   {$set: {adresse: {numero: 20, ville: "Marseille", codepostal: 
4   "13015"}}}
5 )

```

- 17)- Attribuer une prime de 1500 à tous les employés n'ayant pas de prime et dont la ville de résidence est différente de Toulouse, Bordeaux et Paris :

```

1 db.employees.updateMany(
2   {
3     prime: {$exists: false}, 
4     "adresse.ville": {$nin: ["Toulouse", "Bordeaux", "Paris"]}
5   },
6   {$set: {prime: 1500}}
7 )

```

- 18)- Remplacer le champ tel par un tableau nommé téléphone :

```

1 db.employees.updateMany(
2   {tel: {$exists: true}}, 
3   [
4     {$set: {telephone: ["$tel"]}}, 
5     {$unset: "tel"}
6   ]
7 )

```

- 19)- Créer un champ prime pour les documents qui n'en disposent pas et l'affecter à 100 × nombre de caractères du nom de la ville :

```

1 db.employees.updateMany(
2     {prime: {$exists: false}},
3     [
4         {$set: {
5             prime: {
6                 $multiply: [
7                     100,
8                     {$strLenCP: "$adresse.ville"}
9                 ]
10            }
11        }
12    }
13 )

```

20)- Créer un champ mail :

```

1 db.employees.updateMany(
2     {},
3     [
4         {$set: {
5             mail: {
6                 $cond: {
7                     if: {$not: ["$telephone"]},
8                     then: {$concat: ["$nom", ".", "$prenom", "@formation
9 .fr"]},
9                     else: {$concat: ["$prenom", ".", "$nom", "@formation
10 .fr"]}
11                 }
12             }
13         }
14 )

```

21)- Calculer et afficher la somme de l'ancienneté pour les employés disposant du même prénom :

```

1 db.employees.aggregate([
2     {$group: {
3         _id: "$prenom",
4         sommeAnciennete: {$sum: "$anciennete"}
5     }}
6 ])

```