

TP MongoDB - Gestion des employés

L. Ziad

EST-Essaouira

Installation et configuration

- 1) Lancer la commande :

```
1 mongod  
2
```

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

- 3) Dans la console, exécuter la commande :

```
1 mongorestore --db gescom cheminabsoludufichier/employes.bson --  
port numeroPort  
2
```

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

```
1 mongo --port numeroPort gescom  
2
```

Exercices

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

```
1 show collections  
2
```

- 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 };  
5
```

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

```
1 db.employes.countDocuments()  
2
```

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

Méthode 1 :

```

1     db.employes.insertOne({
2         nom: "Dupont",
3         prenom: "Pierre",
4         prime: 500
5     })
6

```

Méthode 2 :

```

1     db.employes.insertOne({
2         nom: "Martin",
3         prenom: "Sophie",
4         anciennete: 3
5     })
6

```

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

```

1     db.employes.find({prenom: "David"})
2

```

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

```

1     db.employes.find({
2         $or: [
3             {prenom: /^D/},
4             {prenom: /D$/}
5         ]
6     })
7

```

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

```

1     db.employes.find({
2         prenom: /^D....$/
3     })
4

```

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

```

1     db.employes.find({
2         prenom: /^[aeiouAEIOU].*[aeiouAEIOU]$/
3     })
4

```

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

```

1     db.employes.find({
2         $expr: {
3             $eq: [
4                 {$substrCP: ["$prenom", 0, 1]},
5                 {$substrCP: ["$prenom", -1, 1]}
6             ]
7         }
8     })
9

```

- 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   )
5

```

- 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   )
5

```

- 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   )
5

```

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

```

1   db.employees.find()
2     .sort({"anciennete": -1})
3     .limit(3)
4

```

- 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   )
5

```

- 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   })
5

```

- 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: "13015"}}}
4   )
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
8

```

- 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
8

```

- 19)- Créer un champ `prime` pour les documents qui n'en disposent pas et l'affecter à $100 \times$ 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}
14

```

- 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.fr"]
9               }],
10                  else: {$concat: ["$prenom", ".", "$nom", "@formation.fr"]}
11              }
12          }
13      }
14  ]
15

```

- 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 ])
7
```