

Lab Note 3: MongoDB

Subject

Preamble

Install MongoDB Server on this [link](#) and MongoDB Tools (to import our database) on this [link](#)

Tasks

1. Simple query

1. Combien y a-t-il d'utilisateurs dans la base de données ?

```
> db.users.count()
```

Result `6040`

2. Combien y a-t-il de film dans la base de données ?

```
> db.movies.count()
```

Result : 3883

3. Quelle est l'occupation de Clifford Johnathan ? Ecrivez une requêtes dont la réponse affiche uniquement son nom et son occupation.

```
db.users.find(
  {"name" : "Clifford Johnathan"},
  {"_id" : 0, "name" : 1,
   "occupation" : 1} )
```

Result :

```
{ "name" : "Clifford Johnathan", "occupation" : "technician/engineer" }
```

4. Combien d'utilisateurs ont entre 18 et 30 ans (inclus) ?

```
db.users.find(
  {"age":{$gte : 18},
   "age":{$lte : 30}}
).count()
```

Result 2587

5. Combien d'utilisateurs sont artistes (artist) ou scientifiques (scientist) ?

```
db.users.find(
  { $or:
    [ {"occupation":"artist"},
      {"occupation":"scientist"}
    ]}).count()
```

Result : 411

6. Quelles sont les dix femmes auteurs (writer) les plus âgées ?

```
db.users.find({
  "gender":"F",
  "occupation":"writer"},
  {"_id" : 0, "name" : 1, "age" : 1}
).sort(
  {"age": -1}
).limit(10)
```

Result :

```
{ "name" : "Kasha Glen", "age" : 76 }
{ "name" : "Alice Maurice", "age" : 54 }
{ "name" : "Nickole Jonathan", "age" : 54 }
{ "name" : "Daphine Humberto", "age" : 53 }
{ "name" : "Gisele Prince", "age" : 53 }
{ "name" : "Kassandra Elden", "age" : 53 }
{ "name" : "Alvina Dante", "age" : 50 }
{ "name" : "Marie Berry", "age" : 50 }
{ "name" : "Babara Jayson", "age" : 50 }
{ "name" : "Cordia Jess", "age" : 49 }
```

7. Quelles sont toutes les occupations présentes dans la base de données ?

```
db.users.distinct(
  'occupation')
```

Result :

```
... 'occupation')
[
  "K-12 student",
  "academic/educator",
  "artist",
  "clerical/admi",
  "college/grad student",
  "customer service",
  "developer",
  "doctor/health care",
  "executive/managerial",
  "farmer",
  "homemaker",
  "lawyer",
  "other",
  "retired",
  "sales/marketing",
  "scientist",
  "self-employed",
  "technician/engineer",
  "tradesman/craftsman",
  "unemployed",
```

```
    "writer"  
  ]  
}
```

2. Insertions, mises-à-jour et suppressions

8. Insérer un nouvel utilisateur dans la base de données (vous, par exemple).

```
db.users.save({  
  "_id" : 10000,  
  "name" : "Combe Antoine",  
  "gender" : "M",  
  "age" : 22,  
  "occupation" : "programmer",  
  "movies": {  
    "movieid" : 1408,  
    "rating" : 5,  
    "timestamp" : 956714023  
  }  
})
```

Result: WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 10000 })

9. Supprimer l'entrée de la base de données.

```
db.users.remove({  
  "name": "Combe Antoine",  
  "gender" : "M",  
  "age" : 22,  
  "occupation" : "programmer"  
})
```

Result: WriteResult({ "nRemoved" : 1 })

10. Pour tous les utilisateurs qui ont pour occupation "programmer", changer cette occupation en "developer".

```
db.users.update(  
  {"occupation" : "programmer"},  
  {$set : {"occupation" : "developer"}},  
  {"multi" : true}  
)
```

Result: WriteResult({ "nMatched" : 387, "nUpserted" : 0, "nModified" : 388 })

11. Les genres du film "Cinderella" devraient être Animation, Children's et Musical. Modifier en une seule requête le document correspondant pour qu'il contienne ces trois genres sans doublon.

```
db.movies.update(  
  {"title" : "Cinderella (1950)"},  
  {$set : {"genres" : "Animation|Children's|Musical"}}  
)
```

Result: WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

3. Expressions régulières

12. Combien de films sont sortis dans les années quatre-vingt ? (l'année de sortie est indiquée entre parenthèses à la fin du titre de chaque film)

```
db.movies.find(
  {"title" : /198/i}
).count()
```

Result : 0

13. Combien y a-t-il de films d'horreur ?

```
db.movies.find(
  {"genres" : /Horror/i}
).count()
```

Result : 343

14. Combien de films ont pour type à la fois "Musical" et "Romance" ?

```
db.movies.find( {
  $and: [
    { "genres" : /Romance/i },
    { "genres" : /Musical/i }
  ]
} ).count()
```

Result : 18

4. Requêtes sur des tableaux

15. Combien d'utilisateurs ont noté le film qui a pour id 1196 (Star Wars: Episode V - The Empire Strikes Back (1980)) ?

```
db.users.find({
  "movies.movieid": 1196 }
).count()
```

Result : 2990

16. Combien d'utilisateurs ont noté tous les films de la première trilogie Star Wars (id 260, 1196, 1210) ?

```
db.users.find({
  "movies.movieid": 1196,
  "movies.movieid": 260,
  "movies.movieid": 1210}
).count()
```

Result : 2883

17. Combien d'utilisateurs ont notés exactement 48 films ?

```
db.users.find({
  movies: {$size : 48}}
```

```
).count()
```

Result : 51

18. Pour chaque utilisateur, créer un champ num_ratings qui indique le nombre de films

```
db.users.aggregate([
  {
    "$addField": {
      "num_ratings": { "$size": "$movies" } }
  },
  { "$out": "users" }
])
```

19. Combien d'utilisateurs ont noté plus de 90 films ?

```
db.users.find(
  {"num_ratings": {$gt : 90}}
).count()
```

Result : 3114

20. Quels sont les trois derniers films notés par Jayson Brad ?

```
db.users.aggregate(
  { $match: {name:"Jayson Brad"}},
  { $unwind: '$movies' },
  { $sort: {'movies.timestamp': -1}},
  { $limit: 3 })
```

Result :

```
{ "_id" : 6016, "name" : "Jayson Brad", "gender" : "M", "age" : 47, "occupation" : "academic/educator", "movies" : { "m
{ "_id" : 6016, "name" : "Jayson Brad", "gender" : "M", "age" : 47, "occupation" : "academic/educator", "movies" : { "m
{ "_id" : 6016, "name" : "Jayson Brad", "gender" : "M", "age" : 47, "occupation" : "academic/educator", "movies" : { "m
```

5. Agrégats

21. Montrer combien de films ont été produits durant chaque année des années 90 ; ordonner les résultats de l'année la plus à la moins fructueuse.

```
db.movies.aggregate(
  { $match: {title:/199/i}},
  { $group : {}}
)
```

22. Quelle est la note moyenne du film Pulp Fiction, qui a pour id 296 ?

```
db.movies.aggregate([
  {$unwind : '$users'},
  {$match : { moviesid : 296 } }
  {$group :
    {"moviesid" : "$moviesid",
```

```

    "Title": {$max: "$title"},
    "Average" : {$avg : "$movies.rating"}
  }}
})

```

23. En une seule requête, retourner pour chaque utilisateur son id, son nom, les notes maximale, minimale et moyenne qu'il a données, et ordonner le résultat par note moyenne croissante.

```

db.users.aggregate([
  {$unwind : '$movies'},
  {$group :
    {
      "_id" : "$_id",
      "name": {$max: "$name"},
      "Average" : {$avg : "$movies.rating"},
      "Maximum" : {$max : "$movies.rating"},
      "Minimum" : {$min : "$movies.rating"}
    }
  },
  {$sort : {"Average" : 1}},
])

```

Result :

```

{ "_id" : 3598, "name" : "Billie Evan", "Average" : 1.0153846153846153, "Maximum" : 2, "Minimum" : 1 }
{ "_id" : 4486, "name" : "Logan Kendrick", "Average" : 1.0588235294117647, "Maximum" : 3, "Minimum" : 1 }
{ "_id" : 2744, "name" : "Ty Hank", "Average" : 1.3043478260869565, "Maximum" : 5, "Minimum" : 1 }
...

```

24. Quel est le genre le plus populaire en termes de nombre de notes ?

25. Quel est le genre le mieux noté (celui dont la moyenne de toutes les notes est la plus élevée) ?