Advanced Database

Assignement 3
TP MongoDB

Jacques Polart
ing4 – SI – Groupe 4
(03/12/2021)

Requêtes simples

Question 1. Combien y a-t-il d'utilisateurs dans la base de données ?
```
>db.users.countDocuments()  // or db.users.count() but it's deprecated

>6040
```

Question 2. Combien y a-t-il de films dans la base de données ?
```
>db.movies.countDocuments()

>3883
```

Question 3. Quelle est l'occupation de Clifford Johnathan ? Ecrivez une requêtes dont la réponse affiche uniquement son nom et son occupation.
```
>db.users.findOne({name: 'Clifford Johnathan'}, {name: 1, occupation: 1, _id: 0})

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

Question 4. Combien d'utilisateurs ont entre 18 et 30 ans (inclus) ?
```
>db.users.countDocuments({$and: [{age: {$gte: 18}}, {age: {$lte: 30}}]})

>2365
```

Question 5. Combien d'utilisateurs sont artistes (artist) ou scientifiques (scientist) ?
```
>db.users.countDocuments({occupation: {$in: ['artist', 'scientist']}})

>411
```

Question 6. Quelles sont les dix femmes auteurs (writer) les plus âgées ?
```
>db.users.aggregate([
        {$match: {$and: [{gender: 'F'}, {occupation: 'writer'}]}},
        {$sort: {age: -1}}, >{$limit: 10},
        {$project: {_id: 0, name: 1, age: 1}}
])

>{ 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: 'Kasandra Elden', age: 53 }
{ name: 'Alvina Dante', age: 50 }
{ name: 'Marie Berry', age: 50 }
{ name: 'Babara Jayson', age: 50 }
{ name: 'Cordia Jess', age: 49 }
```

Question 7. Quelles sont toutes les occupations présentes dans la base de données ?
>db.users.distinct('occupation')

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

Insertions, mises-à-jour et suppressions

Question 8. Insérer un nouvel utilisateur dans la base de données (vous, par exemple).
>db.users.insertOne({name: 'Polart Jacques', gender: 'M', age: 22, occupation: 'student', movies: []})

```
>{ acknowledged: true,
  insertedId: ObjectId("61a737733fddd2d8b81417cc") }
```

Question 9. Supprimer l'entrée de la base de données.
>db.users.deleteOne({name: 'Polart Jacques'})

```
>{ acknowledged: true, deletedCount: 1 }
```

Question 10. Pour tous les utilisateurs qui ont pour occupation "programmer", changer cette occupation en "developer".
>db.users.updateMany({occupation: 'programmer'}, {$set: {occupation: 'developer'}})

```
>{ acknowledged: true,
  insertedId: null,
  matchedCount: 388,
  modifiedCount: 388,
  upsertedCount: 0 }
```

Question 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.updateOne({title: {$regex: /Cinderella/}}, {$set: {genre: "Animation|Children's|
Musical"}})

>{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

Expressions régulières

Question 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.count({title: {$regex: /\(198/}})

>598
```

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

```
>db.movies.count({genres: {$regex: /Horror/}})

>343
```

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

```
>db.movies.count({$and: [{genres: {$regex: /Romance/}}, {genres: {$regex: /Musical/}}]})

>18
```

Requêtes sur des tableaux

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

```
>db.users.count({movies: {$elemMatch: {movieid: 1196}}})

>2990
```

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

```
>db.users.count({movies: {$all: [{$elemMatch: {movieid: 260}}, {$elemMatch: {movieid: 1196}}, {$elemMatch: {movieid: 1210}}]}})

>1926
```

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

```
>db.users.count({movies: {$size: 48}})

>51
```

Question 18. Pour chaque utilisateur, créer un champ num_ratings qui indique le nombre de films qu'il a notés.

```
>db.users.updateMany({}, [{$set: {num_ratings: {$size: '$movies'}}}])

>{ acknowledged: true,
  insertedId: null,
  matchedCount: 6040,
  modifiedCount: 6040,
  upsertedCount: 0 }
```

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

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

>3114
```

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

```
>db.users.find({name: 'Jayson Brad'}, {_id: 0, movies: {$slice: -3}, movie: 1})

>{ movies:
   [ { movieid: 1097, rating: 3, timestamp: 956776340 },
     { movieid: 2043, rating: 2, timestamp: 956778658 },
     { movieid: 3783, rating: 4, timestamp: 963610480 } ] }
```

Agrégats

Question 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([
        {$match: {title: {$regex: /\(199\d\)/}}},
        {$project: {output: {$regexFind: {input: '$title', regex:/(199\d)/}}}},
        {$group: {_id: '$output.match', count: {$count: {}}}},
        {$sort: {count: -1}}
])

>{ _id: '1996', count: 345 }
{ _id: '1995', count: 342 }
{ _id: '1998', count: 337 }
{ _id: '1997', count: 315 }
{ _id: '1999', count: 283 }
{ _id: '1994', count: 257 }
{ _id: '1993', count: 165 }
{ _id: '1992', count: 102 }
{ _id: '1990', count: 77 }
{ _id: '1991', count: 60 }
```

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

```
>db.users.aggregate([
        {$unwind: '$movies'},
        {$match: {'movies.movieid': 296}},
        {$group: {_id: null, avg: {$avg: '$movies.rating'}}},
        {$project: {_id: 0}}
])

>{ avg: 4.278212805158913 }
```

Question 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'},
        {$group: {
                _id: {_id: '$_id', name: '$name'},
                max: {$max: '$movies.rating'},
                min: {$min: '$movies.rating'},
                avg: {$avg: '$movies.rating'}}
        },
        {$project: {_id: '$_id._id', name: '$_id.name', min: '$min', max: '$max', avg: '$avg'}},
        {$sort: {avg: 1}}
])

>{ _id: 3598,
  name: 'Billie Evan',
  min: 1,
```

```
  max: 2,
  avg: 1.0153846153846153 }
{ _id: 4486,
  name: 'Logan Kendrick',
  min: 1,
  max: 3,
  avg: 1.0588235294117647 }
  .
  .
  .
```

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

```
>db.users.aggregate([
        {$unwind: '$movies'},
        {$lookup: {from: 'movies', localField: 'movies.movieid', foreignField: '_id', as: 'movie'}},
        {$unwind: '$movie'},
        {$project: {output: {$regexFind: {input: '$movie.genres', regex:/([^|])+/}}}},
        {$group: {_id: '$output.match', reviews: {$count: {}}}},
        {$sort: {reviews: -1}},
        {$limit: 1}
])

>{ _id: 'Comedy', reviews: 276923 }
```

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

```
>db.users.aggregate([
        {$unwind: '$movies'},
        {$lookup: {from: 'movies', localField: 'movies.movieid', foreignField: '_id', as: 'movie'}},
        {$unwind: '$movie'},
        {$project: {
                output: {$regexFind: {input: '$movie.genres', regex:/([^|])+/}},
                rating: '$movies.rating'}
        },
        {$group: {_id: '$output.match', avg: {$avg: '$rating'}}},
        {$sort: {avg: -1}},
        {$limit: 1}
])

>{ _id: 'Film-Noir', avg: 4.167612116022691 }
```