TP MongoDB - Restaurants et Employés

L. Ziad

EST-Essaouira

TP 5 - Restaurants

Exercices sur la collection restaurants

1) Lister les informations du restaurant "Cafe Henri" :

```
db.restaurants.find({name: "Cafe Henri"}).pretty()
```

2) Lister les restaurants n'ayant pas de quartier connu ("Missing") :

```
db.restaurants.find({borough: "Missing"})
```

3) Lister les restaurants ayant eu un score de 0 :

```
db.restaurants.find({"grades.score": 0})
```

4) Lister les restaurants qui ont le terme "Cafe" dans leur nom :

```
db.restaurants.find({name: /Cafe/})
```

5) Quelles sont les 10 plus grandes chaînes de restaurants (nom identique) ? :

```
db.restaurants.aggregate([
    {
        $group: {
            _id: "$name",
            count: {$sum: 1}
        }
    },
    {
        $sort: {count: -1}
    },
    {
        $limit: 10
    }
])
```

6) Donner le Flop 5 des types de cuisine, en terme de nombre de restaurants :

```
db.restaurants.aggregate([
    {
        $group: {
            _id: "$cuisine",
            count: {$sum: 1}
        }
    },
    {
        $sort: {count: 1}
```

```
    },
    {
        $limit: 5
    }
])
```

7) Donner les dates de début et de fin des évaluations :

```
db.restaurants.aggregate([
    {
        $unwind: "$grades"
    },
    {
        $group: {
            _id: null,
            date_debut: {$min: "$grades.date"},
            date_fin: {$max: "$grades.date"}
        }
    }
])
```

8) Quels sont les 10 restaurants (nom, quartier, adresse et score) avec le plus grand score moyen ? :

```
db.restaurants.aggregate([
    {
        $unwind: "$grades"
    },
    {
        $group: {
            _id: {
                na: "$name",
                id: "$restaurant_id",
                bo: "$borough",
                ad: "$address"
            },
            sc: {
                $avg: "$grades.score"
            }
        }
    },
    {
        $sort: { sc: -1 }
    },
    {
        $limit: 10
    },
    {
        $project: {
            _id: 0,
            name: "$_id.na",
            address: {
                $concat: [
                    "$_id.ad.building", " ", "$_id.ad.street", " , ", "
   $_id.bo"
                ]
            },
            score: "$sc"
        }
    }
]).pretty()
```

9) Compter le nombre d'évaluation par jour de la semaine :

```
db.restaurants.aggregate([
    {
        $unwind: "$grades"
    },
    {
        $project: {
            jour: { $dayOfWeek: "$grades.date" }
        }
    },
    {
        $group: {
            _id: "$jour",
            nb: { $sum: 1 }
        }
    },
    {
        $sort: { _id: 1 }
    }
])
```

TP 6 - Employés

1. Requêtes de base

Q1) Afficher tous les documents de la collection employes :

```
db.employes.find().pretty()
```

Q2) Trouver tous les employés vivant à Toulouse :

```
db.employes.find({ "adresse.ville": "Toulouse" })
```

Q3) Lister les employés ayant une ancienneté supérieure ou égale à 10 ans :

```
db.employes.find({ "anciennete": { $gte: 10 } })
```

Q4) Trouver tous les employés qui ont une prime :

```
db.employes.find({ "prime": { $exists: true } })
```

2. Mise à jour et suppression de documents

Q5) Ajouter une prime de 1000€ aux employés ayant une ancienneté de plus de 20 ans :

```
db.employes.updateMany(
    { "anciennete": { $gt: 20 } },
    { $set: { "prime": 1000 } }
)
```

Q6) Modifier le numéro de téléphone de l'employé "Franck Priou" pour "123456789" :

```
db.employes.updateOne(
    { "nom": "Priou", "prenom": "Franck" },
    { $set: { "telephone": "123456789" } }
)
```

Q7) Supprimer tous les employés ayant une ancienneté inférieure à 2 ans :

```
db.employes.deleteMany({ "anciennete": { $lt: 2 } })
```

3. Exercices avec aggregate

A. Comptage et statistiques

Q8) Calculer le nombre total d'employés :

```
db.employes.aggregate([
    { $count: "total_employes" }
])
```

Q9) Trouver l'ancienneté moyenne des employés :

```
db.employes.aggregate([
    {
        $group: {
            _id: null,
            anciennete_moyenne: { $avg: "$anciennete" }
        }
    }
])
```

Q10) Trouver l'ancienneté maximale et minimale des employés :

```
db.employes.aggregate([
    {
        $group: {
            _id: null,
            anciennete_max: { $max: "$anciennete" },
            anciennete_min: { $min: "$anciennete" }
        }
    }
])
```

B. Groupement et filtrage

Q11) Trouver le nombre d'employés par ville :

```
db.employes.aggregate([
    {
        $group: {
            _id: "$adresse.ville",
            nombre_employes: { $sum: 1 }
        }
    }
])
```

Q12) Calculer la prime moyenne des employés qui en ont une :

```
db.employes.aggregate([
    {
        $match: {
            "prime": { $exists: true }
        }
    },
    {
        $group: {
            _id: null,
            prime_moyenne: { $avg: "$prime" }
        }
    }
])
```

C. Manipulation avancée des données

Q13) Lister les employés par ancienneté décroissante (Top 5) :

```
db.employes.aggregate([
    {
        $sort: { anciennete: -1 }
    },
    {
        $limit: 5
    }
])
```

Q14) Regrouper les employés par prénom et compter combien partagent le même prénom :

```
db.employes.aggregate([
    {
        $group: {
            _id: "$prenom",
            count: { $sum: 1 }
        }
    },
    {
        $sort: { count: -1 }
    }
])
```

D. Combinaisons et transformations

Q15) Ajouter un champ "catégorie ancienneté" (Junior < 5 ans, Senior entre 5 et 15 ans, Expert > 15 ans) :

```
db.employes.aggregate([
    {
        $addFields: {
            categorie_anciennete: {
                $switch: {
                    branches: [
                        {
                            case: { $lt: ["$anciennete", 5] },
                            then: "Junior"
                        },
                        {
                            case: {
                                $and: [
                                    { $gte: ["$anciennete", 5] },
                                    { $lte: ["$anciennete", 15] }
                                ]
                            },
                            then: "Senior"
                        }
                    ],
                    default: "Expert"
                }
            }
        }
    }
])
```

E. Recherche avancée

Q16) Lister les employés avec un numéro de téléphone triés par ancienneté décroissante :

```
db.employes.aggregate([
    {
        $match: {
            "telephone": { $exists: true }
        }
    },
    {
        $sort: { anciennete: -1 }
    }
])
```