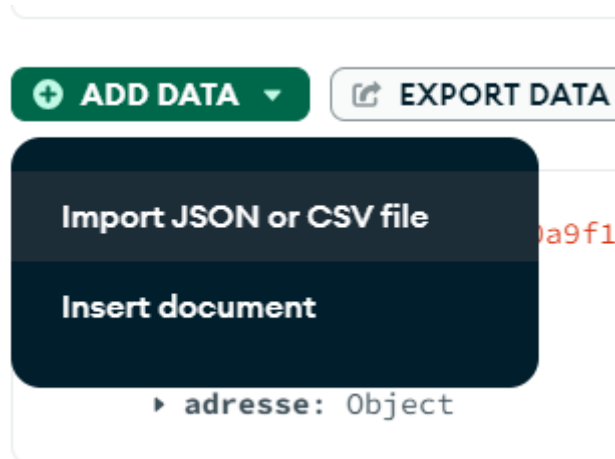


# TP MongoDB

On commence par une Importation de base de donnée



1)

-selectionner la base de donnée :

```
> use gescom
< switched to db gescom
gescom>
```

-afficher les collection

```
> show collections
< employe
```

2) afficher tous les documents de la base

```
> db.employe.find({})
< {
  _id: ObjectId("511d0a9f181b16509ae5f7d8"),
  nom: 'Dupond',
  prenom: 'Jean',
  anciennete: 10,
  adresse: {
    numero: 77,
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

```

{
  _id: ObjectId("511d0a9f181b16509ae5f7da"),
  nom: 'Ossola',
  prenom: 'Christophe',
  anciennete: 7,
  adresse: {
    numero: 75,
    codepostal: 75000,
    ville: 'Paris'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7db"),
  nom: 'Monnin',
  prenom: 'Gilles',
  anciennete: 2,
  adresse: {
    numero: 80,
    rue: 'General Leclerc',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}

```

3) compter le nombre de documents de la collection employes

```

> db.employe.find({}).count()
< 48

```

4) inserer de deux manieres differentes deux employes avec les champs nom, prenom et soit prime soit anciennete

Methode1 :

```
> db.employees.insertOne({
  nom: "Guesmi",
  prenom: "Molka",
  prime: 5000
});
< {
  acknowledged: true,
  insertedId: ObjectId("6565f7b258ea870cb72d24f9")
}
> db.employees.insertOne({
  nom: "Guesmi",
  prenom: "Rami",
  prime: 5000
});
< {
  acknowledged: true,
  insertedId: ObjectId("6565f7e258ea870cb72d24fa")
}
```

Methode2 :

```

    }
    > db.employees.insertMany([
      {
        nom: "Nom1",
        prenom: "Prenom1",
        prime: 5000
      },
      {
        nom: "Nom2",
        prenom: "Prenom2",
        prime: 7000
      }
    ]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6565f81e58ea870cb72d24fb"),
    '1': ObjectId("6565f81e58ea870cb72d24fc")
  }
}

```

5)

afficher la liste des employes dont le prenom est David

```
> db.employe.find ({prenom:'David'})
< {
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7df"),
  nom: 'Ving',
  prenom: 'David',
  anciennete: 17,
  adresse: {
    numero: 28,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
  nom: 'Kingaba',
  prenom: 'David',
  anciennete: 23,
  adresse: {
    numero: 38,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7fd"),
  nom: 'Buyot',
  prenom: 'David',
  anciennete: 14,
  adresse: {
    numero: 8,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

6)afficher la liste des employes dont le prenom commence ou se termine par D

```
> db.employe.find({$or:[{prenom:/^D/},{prenom:/D$/}]})
```

```
< {
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7df"),
  nom: 'Ving',
  prenom: 'David',
  anciennete: 17,
  adresse: {
    numero: 28,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```



```
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
  nom: 'Kingaba',
  prenom: 'David',
  anciennete: 23,
  adresse: {
    numero: 38,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7e7"),
  nom: 'Mani',
  prenom: 'Dominique',
  anciennete: 2,
  adresse: {
    numero: 47,
    rue: 'Lavoisier',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{  
  _id: ObjectId("511d0aa0181b16509ae5f7fd"),  
  nom: 'Buyot',  
  prenom: 'David',  
  anciennete: 14,  
  adresse: {  
    numero: 8,  
    codepostal: 33000,  
    ville: 'Bordeaux'  
  }  
}
```

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

```
> db.employe.find({
  prenom: {
    $regex: /^D.{4}$/
  }
})
< {
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7df"),
  nom: 'Ving',
  prenom: 'David',
  anciennete: 17,
  adresse: {
    numero: 28,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
  nom: 'Kingaba',
  prenom: 'David',
  anciennete: 23,
  adresse: {
    numero: 38,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7fd"),
  nom: 'Buyot',
  prenom: 'David',
  anciennete: 14,
  adresse: {
    numero: 8,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

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

```
> db.employe.find({$and: [{prenom: /^[AUIOEYaeuioy]/}, {prenom: /[AUIOEYaeuioy]$/}]})
< {
  _id: ObjectId("511d0aa0181b16509ae5f7f6"),
  nom: 'Aumont',
  prenom: 'Audrey',
  anciennete: 10,
  adresse: {
    numero: 18,
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f801"),
  nom: 'Baron',
  prenom: 'Elodie',
  anciennete: 16,
  adresse: {
    numero: 9,
    codepostal: 9500,
    ville: 'Foix'
  },
  tel: 96934634,
  prime: 1500
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f805"),
  nom: 'Batin',
  prenom: 'Orlando',
  anciennete: 2,
  adresse: {
    numero: 27,
    rue: 'Les huissiers',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

9) afficher la liste des personnes dont le prenom commence et se termine par une meme lettre

```
}  
> db.employe.find({prenom: /^(\\w).*\\1$/i })  
< {  
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),  
  nom: 'King',  
  prenom: 'David',  
  anciennete: 27,  
  adresse: {  
    numero: 78,  
    codepostal: 33000,  
    ville: 'Bordeaux'  
  }  
}  
{  
  _id: ObjectId("511d0a9f181b16509ae5f7df"),  
  nom: 'Ving',  
  prenom: 'David',  
  anciennete: 17,  
  adresse: {  
    numero: 28,  
    codepostal: 33000,  
    ville: 'Bordeaux'  
  }  
}  
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7e1"),
  nom: 'Motin',
  prenom: 'Roger',
  anciennete: 2,
  adresse: {
    numero: 67,
    rue: 'Jean Moulin',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
  nom: 'Kingaba',
  prenom: 'David',
  anciennete: 23,
  adresse: {
    numero: 38,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7fd"),
  nom: 'Buyot',
  prenom: 'David',
  anciennete: 14,
  adresse: {
    numero: 8,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}

{
  _id: ObjectId("511d0aa0181b16509ae5f801"),
  nom: 'Baron',
  prenom: 'Elodie',
  anciennete: 16,
  adresse: {
    numero: 9,
    codepostal: 9500,
    ville: 'Foix'
  },
  tel: 96934634,
  prime: 1500
}
```



```
{
  _id: ObjectId("511d0aa0181b16509ae5f805"),
  nom: 'Batin',
  prenom: 'Orlando',
  anciennete: 2,
  adresse: {
    numero: 27,
    rue: 'Les huissiers',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

10) afficher les nom et prenom de chaque employe ayant une anciennete > 10

```
> db.employe.find({anciennete:{$gt:10}})
< {
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7de"),
  nom: 'Rupont',
  prenom: 'Jean',
  anciennete: 11,
  adresse: {
    numero: 15,
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  _id: ObjectId("511d0a9f181b16509ae5f7df"),
  nom: 'Ving',
  prenom: 'David',
  anciennete: 17,
  adresse: {
    numero: 28,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7e0"),
  nom: 'Bass',
  prenom: 'Vincent',
  anciennete: 12,
  adresse: {
    codepostal: 75000,
    ville: 'Paris'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
  nom: 'Kingaba',
  prenom: 'David',
  anciennete: 23,
  adresse: {
    numero: 38,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7ea"),
  nom: 'Caponi',
  prenom: 'Jean',
  anciennete: 12,
  adresse: {
    numero: 77,
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7eb"),
  nom: 'Bouras',
  prenom: 'Gerardo',
  anciennete: 27,
  adresse: {
    numero: 34,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7f0"),
  nom: 'Rigal',
  prenom: 'Cyril',
  anciennete: 14,
  adresse: {
    numero: 108,
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7f7"),
  nom: 'Julien',
  prenom: 'Marie',
  anciennete: 11,
  adresse: {
    numero: 19,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f7f9"),
  nom: 'David',
  prenom: 'Christophe',
  anciennete: 12,
  adresse: {
    numero: 80,
    rue: 'Les tuiles',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7fb"),
  nom: 'Has',
  prenom: 'Stephen',
  anciennete: 13,
  adresse: {
    numero: 7,
    codepostal: 9500,
    ville: 'Foix'
  },
  tel: 96930154,
  prime: 5000
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7fd"),
  nom: 'Buyot',
  prenom: 'David',
  anciennete: 14,
  adresse: {
    numero: 8,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7ff"),
  nom: 'Perrez',
  prenom: 'Gilles',
  anciennete: 15,
  adresse: {
    numero: 2,
    rue: 'Les sabliers',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  _id: ObjectId("511d0aa0181b16509ae5f801"),
  nom: 'Baron',
  prenom: 'Elodie',
  anciennete: 16,
  adresse: {
    numero: 9,
    codepostal: 9500,
    ville: 'Foix'
  },
  tel: 96934634,
  prime: 1500
}
```



```
{
  _id: ObjectId("511d0aa0181b16509ae5f803"),
  nom: 'Moore',
  prenom: 'Roland',
  anciennete: 17,
  adresse: {
    numero: 37,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

11) afficher les nom et adresse complete des employes ayant un attribut rue dans l'objet adresse

```
> db.employe.find(
  { "adresse.rue": { $exists: true } },
  { _id: 0, nom: 1, adresse: 1 }
)
< {
  nom: 'Monnin',
  adresse: {
    numero: 80,
    rue: 'General Leclerc',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  nom: 'Motin',
  adresse: {
    numero: 67,
    rue: 'Jean Moulin',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  nom: 'Mani',
  adresse: {
    numero: 47,
    rue: 'Lavoisier',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  nom: 'Autran',
  adresse: {
    numero: 41,
    rue: 'General DeBase',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  nom: 'Potin',
  adresse: {
    numero: 15,
    rue: 'Jean Sens',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  nom: 'David',
  adresse: {
    numero: 80,
    rue: 'Les tuiles',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
{
  nom: 'Perrez',
  adresse: {
    numero: 2,
    rue: 'Les sabliers',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

```
{
  nom: 'Batin',
  adresse: {
    numero: 27,
    rue: 'Les huissiers',
    codepostal: 31000,
    ville: 'Toulouse'
  }
}
```

12) incrementer de 200 la prime des employes ayant deja le champ prime

```
> db.employe.updateMany({prime:{$exists:true}},{$inc:{prime:200}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 9,
  modifiedCount: 9,
  upsertedCount: 0
}
```

13) afficher les trois premieres personnes ayant la plus grande valeur d'anciennete

```
> db.employe.find({}).sort({
  anciennete:-1}).limit(3)
< {
  _id: ObjectId("511d0aa0181b16509ae5f7eb"),
  nom: 'Bouras',
  prenom: 'Gerardo',
  anciennete: 27,
  adresse: {
    numero: 34,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  _id: ObjectId("511d0a9f181b16509ae5f7d9"),
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

```
{
  _id: ObjectId("511d0aa0181b16509ae5f7e5"),
  nom: 'Kingaba',
  prenom: 'David',
  anciennete: 23,
  adresse: {
    numero: 38,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

14) regrouper les personnes dont la ville de residence est Toulouse (afficher nom, prenom et anciennete)

```
> db.employe.find(
  { "adresse.ville": "Toulouse" },
  { _id: 0, nom: 1, prenom: 1, anciennete: 1 }
)
< {
  nom: 'Dupond',
  prenom: 'Jean',
  anciennete: 10
}
{
  nom: 'Monnin',
  prenom: 'Gilles',
  anciennete: 2
}
{
  nom: 'Rupont',
  prenom: 'Jean',
  anciennete: 11
}
{
  nom: 'Motin',
  prenom: 'Roger',
  anciennete: 2
}
```



```
{
  nom: 'Cuponi',
  prenom: 'Eric',
  anciennete: 10
}
{
  nom: 'Mani',
  prenom: 'Dominique',
  anciennete: 2
}
{
  nom: 'Caponi',
  prenom: 'Jean',
  anciennete: 12
}
{
  nom: 'Autran',
  prenom: 'Vincent',
  anciennete: 2
}
```

```
{
  nom: 'Aumont',
  prenom: 'Audrey',
  anciennete: 10
}
{
  nom: 'David',
  prenom: 'Christophe',
  anciennete: 12
}
{
  nom: 'Voneschen',
  prenom: 'Henri',
  anciennete: 10
}
{
  nom: 'Perrez',
  prenom: 'Gilles',
  anciennete: 15
}
```

```

{
  nom: 'Pasqua',
  prenom: 'Jean',
  anciennete: 5
}
{
  nom: 'Batin',
  prenom: 'Orlando',
  anciennete: 2
}

```

15) afficher les personnes dont le prenom commence par M et la ville de residence est soit Foix soit Bordeaux

```

> db.employe.find({$and:[{nom:/^M/},{adresse.ville:{$in:["Foix", "Bordeaux"]}}]})
< {
  _id: ObjectId("511d0aa0181b16509ae5f7ee"),
  nom: 'Menard',
  prenom: 'Eric',
  anciennete: 5,
  adresse: {
    numero: 42,
    codepostal: 9500,
    ville: 'Foix'
  },
  tel: 561908352
}
{
  _id: ObjectId("511d0aa0181b16509ae5f803"),
  nom: 'Moore',
  prenom: 'Roland',
  anciennete: 17,
  adresse: {
    numero: 37,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}

```

16) mettre a jour l'adresse de Dominique Mani : nouvelle adresse ({ numero : 20, ville : 'Marseille', codepostal : '13015' }). Attention, il n'y aura plus d'attribut rue dans adresse

```
> db.employe.updateOne(  
  { nom: 'Mani', prenom: 'Dominique' },  
  {  
    $set: {  
      adresse: {  
        numero: 20,  
        ville: 'Marseille',  
        codepostal: '13015'  
      }  
    }  
  }  
)  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

17) attribuer une prime de 1 500 `a tous les employes n'ayant pas de prime et dont la ville de residence est differente de Toulouse, Bordeaux et Paris

```

> db.employe.updateOne(
  { prime:{$exists:false} ,"adresse.ville":{$nin:["Toulouse","Bordeaux"," Paris"]}},
  {
    $set: {
      prime:1500
    }
  }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

18) remplacer le champ tel, pour les documents ayant un champ tel), par un tableau nomme telephone contenant la valeur du champ tel (le champ tel est `a supprimer)

```

> db.employe.updateMany(
  { tel: { $exists: true } },
  {
    $set: {
      telephone: ["$tel"]
    },
    $unset: {
      tel: ""
    }
  }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 16,
  modifiedCount: 16,
  upsertedCount: 0
}

```

19) creer un champ prime pour les documents qui n'en disposent pas et de l'affecter `a 100 \* nombre de caractere du nom de la ville

```

> db.employe.updateMany(
  { prime: { $exists: false } },
  {
    $set: {
      prime: {
        $multiply: [
          { $strLenCP: "$adresse.ville" },
          100
        ]
      }
    }
  }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}

```

20) creer un champ mail dont la valeur est egale soit `a nom.prenom@formation.fr pour les employes ne disposant pas d'un champ telephone, soit `a prenom.nom@formation.fr (nom et prenom sont `a remplacer par les vraies valeurs de chaque employe)

```

gescom> db.employe.updateMany(
  { telephone: { $exists: false } },
  {
    $set: {
      mail: {
        $concat: [
          "$nom",
          ".",
          "$prenom",
          "@formation.fr"
        ]
      }
    }
  })
db.employe.updateMany(
  { telephone: { $exists: true } },
  {
    $set: {
      mail: {
        $concat: [
          "$prenom",
          ".",
          "$nom",
          "@formation.fr"
        ]
      }
    }
  })

```

```

< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 16,
  modifiedCount: 0,
  upsertedCount: 0
}

```

21) calculer et afficher la somme de l'anciennete pour les employes disposant du meme prenom

```
> db.employe.aggregate([
  {
    $group: {
      _id: "$prenom",
      sommeAnciennete: { $sum: "$anciennete" },
      count: { $sum: 1 }
    }
  },
  {
    $project: {
      _id: 0,
      prenom: "$_id",
      sommeAnciennete: 1,
      count: 1
    }
  }
])
```



```
< {  
  sommeAnciennete: 4,  
  count: 1,  
  prenom: 'Bertrand'  
}  
{  
  sommeAnciennete: 8,  
  count: 1,  
  prenom: 'Julien'  
}  
{  
  sommeAnciennete: 5,  
  count: 1,  
  prenom: 'Sylvie'  
}  
{  
  sommeAnciennete: 17,  
  count: 1,  
  prenom: 'Roland'  
}
```

```
{
  sommeAnciennete: 1,
  count: 1,
  prenom: 'Melanie'
}
{
  sommeAnciennete: 5,
  count: 1,
  prenom: 'Edouard'
}
{
  sommeAnciennete: 5,
  count: 1,
  prenom: 'James'
}
{
  sommeAnciennete: 7,
  count: 1,
  prenom: 'Garard'
}
```

```
{
  sommeAnciennete: 81,
  count: 4,
  prenom: 'David'
}
{
  sommeAnciennete: 27,
  count: 1,
  prenom: 'Gerardo'
}
{
  sommeAnciennete: 3,
  count: 1,
  prenom: 'Nicolas'
}
{
  sommeAnciennete: 0,
  count: 1,
  prenom: 'rami'
}
{
  sommeAnciennete: 38,
  count: 4,
  prenom: 'Jean'
}
```

```
{
  sommeAnciennete: 33,
  count: 4,
  prenom: 'Christophe'
}
{
  sommeAnciennete: 17,
  count: 2,
  prenom: 'Gilles'
}
{
  sommeAnciennete: 11,
  count: 1,
  prenom: 'Marie'
}
{
  sommeAnciennete: 4,
  count: 2,
  prenom: 'Stephanie'
}
{
  sommeAnciennete: 7,
  count: 1,
  prenom: 'Marc'
}
```

```
{
  sommeAnciennete: 13,
  count: 1,
  prenom: 'Stephen'
}
{
  sommeAnciennete: 15,
  count: 2,
  prenom: 'Eric'
}
```