

# Big Data

## Compte Rendu MongoDB

### Question 1 :

- Réponse :

```
show collections
```

- Résultat :

```
> show collections
< employees
```

### Question 2 :

- Réponse :

```
var collections = db.getCollectionNames();
collections.forEach(function(collectionName) {
    var cursor =
db.getCollection(collectionName).find();
    print("Documents de la collection " +
collectionName + ":");
    while (cursor.hasNext()) {
        printjson(cursor.next());
    }
    print("\n");
});
```

- Résultat :

```
< Documents de la collection employes:
< {
  _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' }
}
< {
```

### Question 3 :

- Réponse :

```
db.employes.find().count()
```

- Résultat :

```
> db.employes.find().count()
< 48
```

## Question 4 :

## ● Réponse :

```
db.employees.insert({nom : "Ghazi",prenom:"Chafar",  
anciennete:10});  
db.employees.insertOne({nom :  
"Ghazii",prenom:"Chafarr", prime:150});
```

## ● Résultat :

```
> db.employees.insert({nom : "Ghazi", prenom:"Chafar", anciennete:10});  
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.  
< {  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId("65800e5528d42a9fef420209")  
  }  
}
```

## Question 4 :

## ● Réponse :

```
db.employees.find({prenom:"David"})
```

## ● Résultat :

```
> db.employees.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'
  }
}
```

### Question 6 :

- Réponse :

```
db.employees.find({prenom:/^D.*|.*D $/});
```

- Résultat :

```
> db.employees.find({prenom:/^D.*|.*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'
  }
}
{
```

### Question 7 :

- Réponse :

```
db.employees.find({prenom:/^D[a-z]{4}$/});
```

- Résultat :

```
> db.employees.find({prenom:/^D[a-z]{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'
  }
}
```

### Question 8 :

- Réponse :

```
db.employees.find({prenom:/^[AEIOUY].*[aeiouy]$/});
```

- Résultat :

```
> db.employees.find({prenom:/^[AEIOUY].*[aeiouy]$/});
< {
  _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'
  },
}
```

### Question 9 :

- Réponse :

```
db.employees.find().forEach(function(p) {
  let pre = p.prenom.toLowerCase();
  if
  (pre.substr(0,1)==pre.substr(pre.length-1,1)) {
    print(pre)
  }
})
```

- Résultat :

```
> db.employees.find().forEach(function(p) {
  let pre = p.prenom.toLowerCase();
  if (pre.substr(0,1)==pre.substr(pre.length-1,1)) {
    print(pre)
  }
})
< david
< david
< roger
< david
< david
< elodie
< orlando
```

### Question 10 :

- Réponse :

```
db.employees.find(
  {anciennete : {$gt : 10}},
  {_id:0, nom:1, prenom : 1}
)
```

- Résultat :

```
> db.employees.find(
  {anciennete : {$gt : 10}},
  {_id:0, nom:1, prenom : 1}
)
< {
  nom: 'King',
  prenom: 'David'
}
{
  nom: 'Rupont',
  prenom: 'Jean'
}
{
  nom: 'Ving',
  prenom: 'David'
}
{
  nom: 'Bass',
  prenom: 'Vincent'
}
{
  nom: 'Kingaba',
  prenom: 'David'
}
```



## Question 11 :

## ● Réponse :

```
db.employees.find(  
  {'adresse.rue': {$exists : true}},  
  {nom : 1, adresse: 1}  
)
```

## ● Résultat :

```
> db.employees.find(  
  {'adresse.rue': {$exists : true}},  
  {nom : 1, adresse: 1}  
)  
< {  
  _id: ObjectId("511d0a9f181b16509ae5f7db"),  
  nom: 'Monnin',  
  adresse: {  
    numero: 80,  
    rue: 'General Leclerc',  
    codepostal: 31000,  
    ville: 'Toulouse'  
  }  
}  
{  
  _id: ObjectId("511d0aa0181b16509ae5f7e1"),  
  nom: 'Motin',  
  adresse: {  
    numero: 67,  
    rue: 'Jean Moulin',  
    codepostal: 31000,  
    ville: 'Toulouse'  
  }  
}
```

## Question 12 :

## ● Réponse :

```
db.employees.update(  
  {prime:{$exists : true}},  
  {$inc : {prime:200}},  
  {multi : true}  
)
```

## ● Résultat :

```
> db.employees.update(  
  {prime:{$exists : true}},  
  {$inc : {prime:200}},  
  {multi : true}  
)  
< DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 9,  
  modifiedCount: 9,  
  upsertedCount: 0  
}
```

## Question 13 :

## ● Réponse :

```
db.employees.find(  
  anciennete: {$exists : true},  
  {_id : 0}  
) .sort({anciennete : -1}).limit(3)
```

## ● Résultat :

```
> db.employees.find({
  anciennete: {$exists : true}},
  {_id : 0}
).sort({anciennete : -1}).limit(3)
< {
  nom: 'Bouras',
  prenom: 'Gerardo',
  anciennete: 27,
  adresse: {
    numero: 34,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
{
  nom: 'King',
  prenom: 'David',
  anciennete: 27,
  adresse: {
    numero: 78,
    codepostal: 33000,
    ville: 'Bordeaux'
  }
}
```

#### Question 14 :

- Réponse :

```
db.employees.find(
  {'adresse.ville': "Toulouse"},
  {_id : 0, nom:1, anciennete:1, prenom:1}
)
```

- Résultat :

```
db.employees.find(
  {'adresse.ville': "Toulouse"},
  {_id : 0, nom:1, anciennete:1, prenom:1}
)
[
  {
    nom: 'Dupond',
    prenom: 'Jean',
    anciennete: 10
  },
  {
    nom: 'Monnin',
    prenom: 'Gilles',
    anciennete: 2
  }
]
```

### Question 15 :

- Réponse :

```
db.employees.find(
  {
    prenom : "/^M.*/",
    "adresse.ville" : {
      $in : ["Foix", "Bordeaux"]
    }
  }
)
```

- Résultat :

```
> db.employees.find(
  {
    prenom : "/^M.*/",
    "adresse.ville" : {
      $in : ["Foix", "Bordeaux"]
    }
  }
)
<
```

## Question 16 :

## ● Réponse :

```
db.employees.update(  
  {prenom : "Dominique", nom:"Mani"},  
  {  
    $set : {  
      'adresse.numero' : 20,  
      'adresse.ville' : 'Marseille',  
      'adresse.codepostal': '13015',  
    },  
    $unset :{  
      'adresse.rue' : 1  
    }  
  }  
)
```

## ● Résultat :

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

## Question 17 :

## ● Réponse :

```
db. employees.updateMany (
  {$and :
    [
      {"adresse . ville":
        {snin : ["Paris ", " Toulouse ", "
Bordeaux " ]}}
    ],
    {prime : { sexists : false }}
  },
  {$set: {prime : 1500 }}
);
```

## ● Résultat :

```
> db. employees.updateMany (
  {$and :
    [
      {"adresse . ville":
        {snin : ["Paris ", " Toulouse ", " Bordeaux " ]}}
    ],
    {prime : { sexists : false }}
  },
  {$set: {prime : 1500 }}
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

## Question 18 :

## ● Réponse :

```
db.employees.find(
  {
    tel : {$exists : true}
  },
  {}
).forEach(
  function(t) {
    db.employees.updateMany(
      {_id : t._id},
      {
        $push : {telephone : t.tel},
        $unset : {tel : 1}
      }
    )
  }
)
```

## ● Résultat :

```
> db.employees.find(
  {
    tel : {$exists : true}
  },
  {}
).forEach(
  function(t) {
    db.employees.updateMany(
      {_id : t._id},
      {
        $push : {telephone : t.tel},
        $unset : {tel : 1}
      }
    )
  }
)
```

## Question 19 :

## ● Réponse :

```
db.employees.find(
  {
    prime : {$exists : 1}
  }
).forEach(
  function(doc) {
    var length = doc.adresse.ville.length;
    var newPrime = 100*length;
    db.employees.update(
      { _id : doc._id },
      {
        $inc : {prime : newPrime}
      }
    )
  }
)
```

## Question 20 :

## ● Réponse :

```
db.employees.find().forEach(
  function(p) {
    var email = p.nom + '.' + p.prenom +
    '@formation.fr';
    if (p.telephone) {
      var email = p.prenom + '.' + p.nom +
      '@formation.fr';
    }
    db.employees.updateMany(
```



```
        { _id : p._id },  
        { $set : { mail : email } }  
    )  
}  
)
```

### Question 21 :

- Réponse :

```
db.employees.aggregate([  
  {  
    $group: {  
      _id: "$prenom",  
      totalAnciennete: { $sum: "$anciennete" }  
    }  
  }  
])
```

- Résultat :

```
> db.employees.aggregate([  
  {  
    $group: {  
      _id: "$prenom",  
      totalAnciennete: { $sum: "$anciennete" }  
    }  
  }  
])  
< {  
  _id: 'Julien',  
  totalAnciennete: 8  
}  
{  
  _id: 'Melanie',  
  totalAnciennete: 1  
}  
{  
  _id: 'Arnaud',  
  totalAnciennete: 6  
}  
{  
  _id: 'Gilles',  
  totalAnciennete: 17  
}
```