TP 2 - NoSQL

I - Lecture

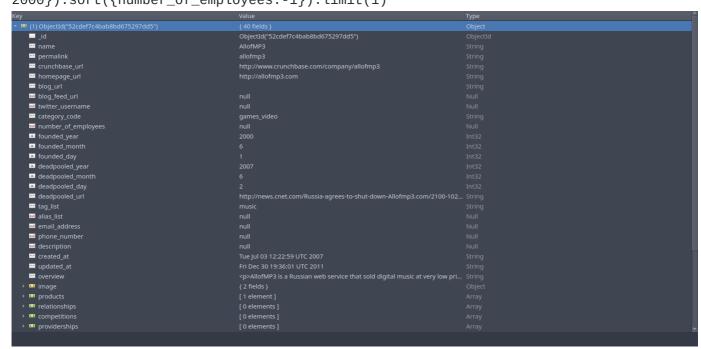
1. Combien de sociétés ont été créées durant toute l'année 2010 ?

```
db.getCollection('companies').find({founded_year: 2010}).count()

db.getCollection('companies').find({founded_year: 2010}).count()

0.01 sec.
33
```

2. Sur les sociétés fondées en 2000, laquelle possède le plus d'employé ? Et Combien ? db.getCollection('companies').find({founded_year: 2000}).sort({number_of_employees:-1}).limit(1)



3. Combien de sociétés ont des bureaux localisés dans la ville de Seattle ?
db.getCollection('companies').find({ "offices.city" : 'Seattle' }).count()

4. Combien de sociétés sont implantées en France pour au moins un bureau ?
db.getCollection('companies').find({ 'offices.country_code': 'FRA' }).count()

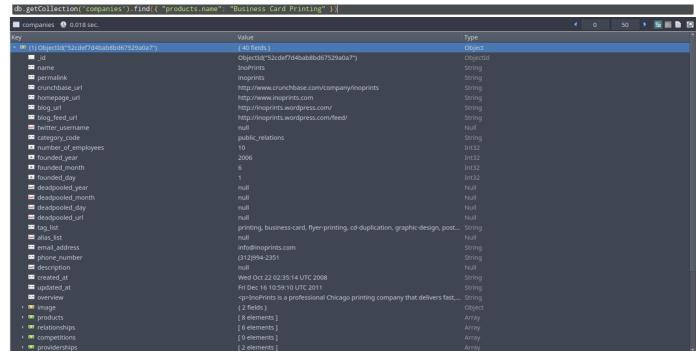
https://md2pdf.netlify.app 1/7

```
db.getCollection('companies').find({ 'offices.country_code': 'FRA' }).count()

     0.018 sec.
184
```

5. Combien de sociétés ont des employés au poste de : Software Engineer ? db.getCollection('companies').find({ 'relationships.title': 'Software Engineer' }).count()

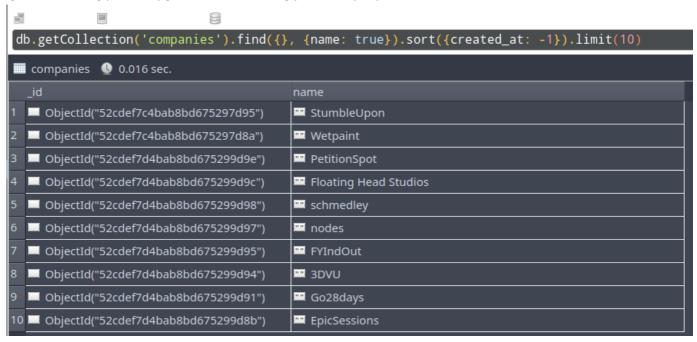
6. Quelle société propose le produit : Business Card Printing
 db.getCollection('companies').find({ "products.name": "Business Card Printing" })



7. Ecrivez un query qui donne le nom des 10 sociétés les plus récentes du jeu de données. **Je ne savais pas si je devais effectuer cette requête sur la base de created_at, ou de founded_year, founded_month et founded_day** db.getCollection('companies').find({},

https://md2pdf.netlify.app 2/7

{name: true}).sort({created_at: -1}).limit(10)



8. Donnez la liste des sociétés créées en mai 2000.

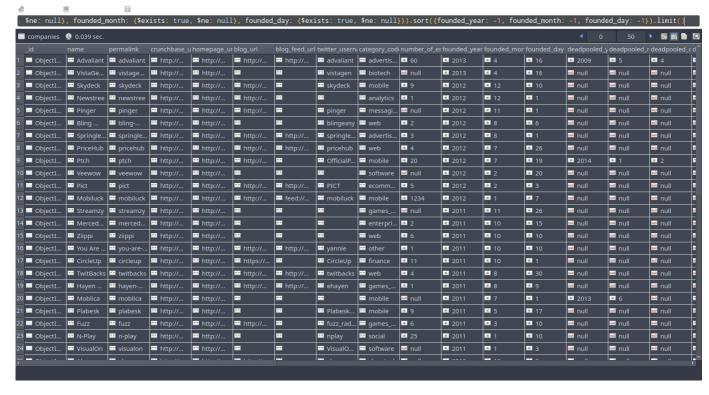
db.getCollection('companies').find({founded_year: 2000, founded_month: 5})



9. Sachant que le sort() en croissant retourne en premier les valeurs nulles. Trouvez un moyen (avec une ou plusieurs queries) de donner la liste des 10 plus anciennes sociétés du jeu de données. db.getCollection('companies').find({founded_year: {\$exists: true, \$ne: null}, founded_month: {\$exists: true, \$ne: null}, founded_day: {\$exists: true, \$ne:

https://md2pdf.netlify.app 3/7

null}}).sort({founded_year: -1, founded_month: -1, founded_day: -1}).limit()



- 10. Donnez le temps d'exécution des requêtes suivantes :
- 11. db.companies.find();

```
use('tp2');
const startTime = Date.now();
db.companies.find();
const endTime = Date.now();
console.log(endTime - startTime);
```

```
tp2> load("/home/dk/Documents/studies/B3/nosql/tp2/script.js")
6ms
true
```

12. db.companies.find().sort({founded_year: -1 })

```
use('tp2');
const startTime = Date.now();
db.companies.find().sort({ founded_year: -1 });
const endTime = Date.now();
console.log(endTime - startTime + 'ms');
```

```
tp2> load("/home/dk/Documents/studies/B3/nosql/tp2/script.js")
10ms
true
```

II - Écriture

https://md2pdf.netlify.app 4/7

1. Créez une collection nommée products contenant 1000 produits, avec une propriété de compteur (exemple : product number) et d'autres champs de votre choix.

```
use('tp2');
db.createCollection('products');
const products = [];
for (let i = 0; i < 1000; i++) {
  products.push({
    product_number: i,
    name: 'Product ' + i,
    price: Math.random() * 100,
    description: 'Description ' + i,
    category: 'Category ' + i,
    sub_category: 'SubCategory ' + i,
    tags: ['tag1', 'tag2', 'tag3'].slice(0, Math.floor(Math.random() * 3)),
    created_at: new Date(),
  });
}
db.products.insertMany(products);
```

2. Ajoutez au produit numéro 737 une propriété nommée version de type tableau contenant les chaines de caractère : ['MAX-7', 'MAX-8', 'MAX-9', 'MAX-10']

[0 elements]

3. Supprimez les produits numéro 13 et 666.

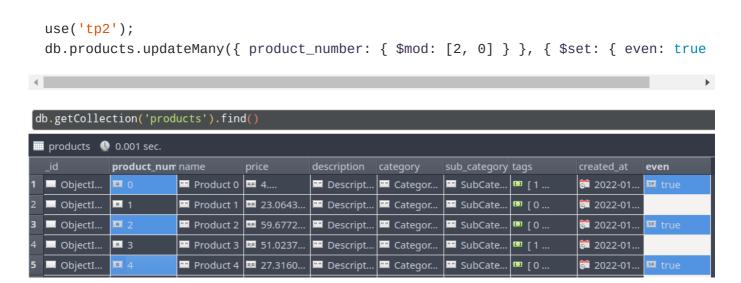
description
category
sub_category

created_at
version
[0]
[1]
[1]

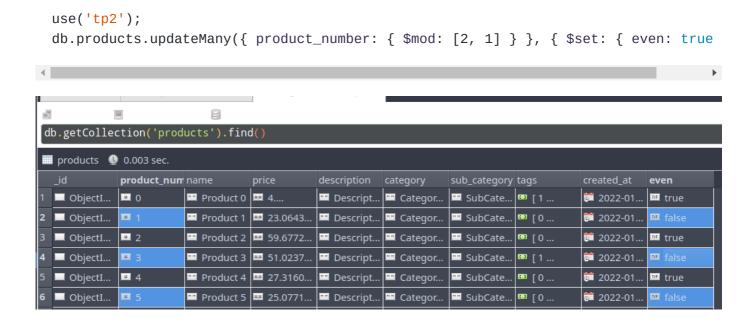
```
use('tp2');
db.products.deleteOne({ product_number: 13 });
db.products.deleteOne({ product_number: 666 });
```

https://md2pdf.netlify.app 5/7

4. Pour tout les produits dont le numéro de compteur est pair, ajoutez un champ nommé even de type booléen à true.



5. Idem pour tout les produits dont le numéro de compteur est impair, ajoutez un champ nommé even de type booléen à false.



6. Afficher le nombre de documents avec le champ even à true.

```
use('tp2');
db.getCollection('products').find({ even: true }).count();
```

https://md2pdf.netlify.app 6/7

7. Incrémentez de 1 le numéro du produit ayant pour id 1000.

8. Corrigez la valeur du champ even du document précédemment modifié.

```
use('tp2');
db.getCollection('products').updateOne({ product_number: 1001 }, { $set: { even: f}
```

- 9. Supprimez tout les produits :
 - i. Dont la propriété even est à true à l'aide d'une boucle.

```
use('tp2');
for (let i = 0; i < 1002; i += 1) {
  db.products.deleteOne({ product_number: i, even: true });
}</pre>
```

ii. Dont la propriété even est à false à l'aide d'une seule commande.

```
use('tp2');
db.getCollection('products').deleteMany({ even: false });
```

https://md2pdf.netlify.app 7/7