

TD 2 : Bases de données NoSQL - MongoDB

Soit la structure de la collection «restaurants» suivante :

```
{
  "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A",
"score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A",
"score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A",
"score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A",
"score": 9 },
    { "date": { "$date": 1299715200000 }, "grade": "B",
"score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

Questions

- 1) Ecrivez une requête MongoDB pour afficher tous les documents de la collection "restaurants".
- 2) Écrivez une requête MongoDB pour afficher les champs "restaurant_id, name, borough and cuisine" pour tous les documents de la collection "restaurant".
- 3) Écrivez une requête MongoDB pour afficher les champs "restaurant_id, name, borough and cuisine", mais excluez le champ _id pour tous les documents de la collection "restaurant".

- 4) Écrivez une requête MongoDB pour afficher les champs "restaurant_id, name, borough and zip code", mais excluez le champ _id pour tous les documents de la collection "restaurant".
- 5) Rédigez une requête MongoDB pour afficher tous les restaurants du quartier Bronx. (borough=quartier)
- 6) Rédigez une requête MongoDB pour afficher les 5 premiers restaurants qui se trouvent dans le quartier Bronx. (borough=quartier)
- 7) Rédigez une requête MongoDB pour afficher les 5 prochains restaurants après avoir ignoré les 5 premiers qui se trouvent dans le quartier Bronx.
- 8) Écrivez une requête MongoDB pour trouver les restaurants qui ont obtenu un score supérieur à 90.
- 9) Écrivez une requête MongoDB pour trouver les restaurants qui ont obtenu un score supérieur à 80 mais inférieur à 100.
- 10) Écrivez une requête MongoDB pour trouver les restaurants dont la latitude est inférieure à -95.754168.
- 11) Écrivez une requête MongoDB pour trouver les restaurants qui ne préparent pas de cuisine «américaine» et leur note supérieure à 70 et leur latitude inférieure à -65.754168.
- 12) Écrivez une requête MongoDB pour trouver les restaurants qui ne préparent pas de cuisine «américaine» et ont obtenu un score supérieur à 70 et situés dans la longitude inférieure à -65,754168. Remarque: effectuez cette requête sans utiliser l'opérateur \$and .
- 13) Écrivez une requête MongoDB pour trouver les restaurants qui ne préparent pas de cuisine «américaine» et ont obtenu un point «A» n'appartenant pas à l'arrondissement de Brooklyn. Le document doit être affiché selon la cuisine en ordre décroissant.
- 14) Écrivez une requête MongoDB pour trouver l'identifiant, le nom, quartier et la cuisine du restaurant pour les restaurants qui contiennent «Wil» comme trois premières lettres de son nom.
- 15) Écrivez une requête MongoDB pour trouver l'ID, le nom, quartier et la cuisine du restaurant pour les restaurants qui contiennent «ces» comme trois dernières lettres pour son nom.
- 16) Écrivez une requête MongoDB pour trouver l'identifiant, le nom, quartier et la cuisine du restaurant pour les restaurants qui contiennent «Reg» en trois lettres quelque part dans son nom.
- 17) Rédigez une requête MongoDB pour trouver les restaurants qui appartiennent au quartier Bronx et qui ont préparé un plat américain ou chinois.
- 18) Rédigez une requête MongoDB pour trouver l'identifiant, le nom, le quartier et la cuisine du restaurant pour les restaurants appartenant au quartier Staten Island ou Queens ou Bronx ou Brooklyn.
- 19) Rédigez une requête MongoDB pour trouver l'identifiant, le nom, le quartier et la cuisine du restaurant pour les restaurants qui n'appartiennent pas au quartier Staten Island ou Queens ou Bronx ou Brooklyn.
- 20) Écrivez une requête MongoDB pour trouver l'identifiant, le nom, quartier et la cuisine du restaurant pour les restaurants qui ont obtenu un score qui n'est pas supérieur à 10.

Solutions

- 1) `db.restaurants.find();`
- 2) `db.restaurants.find({}, {"restaurant_id" : 1, "name":1, "borough":1, "cuisine" :1});`
- 3) `db.restaurants.find({}, {"restaurant_id" : 1, "name":1, "borough":1, "cuisine" :1, "_id":0});`
- 4) `db.restaurants.find({}, {"restaurant_id" : 1, "name":1, "borough":1, "address.zipcode" :1, "_id":0});`
- 5) `db.restaurants.find({"borough": "Bronx"});`
- 6) `db.restaurants.find({"borough": "Bronx"}).limit(5);`
- 7) `db.restaurants.find({"borough": "Bronx"}).skip(5).limit(5);`
- 8) `db.restaurants.find({grades : { $elemMatch: {"score": {$gt : 90}}}});`
- 9) `db.restaurants.find({grades : { $elemMatch: {"score": {$gt : 80 , $lt :100}}}});`
- 10) `db.restaurants.find({"address.coord" : {$lt : -95.754168}});`
- 11) `db.restaurants.find({$and: [{"cuisine" : {$ne : "American "}}, {"grades.score" : {$gt : 70}}, {"address.coord" : {$lt : -65.754168}}]});`
- 12) `db.restaurants.find({ "cuisine" : {$ne : "American "}, "grades.score" : {$gt: 70}, "address.coord" : {$lt : -65.754168} });`
- 13) `db.restaurants.find({ "cuisine" : {$ne : "American "}, "grades.grade" : "A", "borough": {$ne : "Brooklyn"} }).sort({"cuisine":-1});`
- 14) `db.restaurants.find({name: /^Wil/}, { "restaurant_id" : 1, "name":1, "borough":1, "cuisine" :1 });`
- 15) `db.restaurants.find({name: /ces$/},`

```

        {
            "restaurant_id" : 1,
            "name":1,"borough":1,
            "cuisine" :1
        }
    );
16) db.restaurants.find(
    { "name": /. *Reg.*/ },
    {
        "restaurant_id" : 1,
        "name":1,"borough":1,
        "cuisine" :1
    }
);
17) db.restaurants.find(
    {
        "borough": "Bronx" ,
        $or : [
            { "cuisine" : "American " },
            { "cuisine" : "Chinese" }
        ]
    }
);
18) db.restaurants.find(
    { "borough" : {$in : ["Staten
Island", "Queens", "Bronx", "Brooklyn"] } },
    {
        "restaurant_id" : 1,
        "name":1,"borough":1,
        "cuisine" :1
    }
);
19) db.restaurants.find(
    { "borough" : {$nin : ["Staten
Island", "Queens", "Bronx", "Brooklyn"] } },
    {
        "restaurant_id" : 1,
        "name":1,"borough":1,
        "cuisine" :1
    }
);
20) db.restaurants.find(
    { "grades.score" :
    { $not:
    { $gt : 10 }
    }
    },
    {
        "restaurant_id" : 1,

```

```
"name":1,"borough":1,  
"cuisine" :1  
}  
);
```