

# Révision

## Questions de révision

1. Rechercher tous les restaurants servant une cuisine italienne.

*Indication : Utilisez le champ cuisine.*

```
db.restaurants.find( { cuisine : 'Italian' } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

1. Rechercher tous les restaurants situés dans le borough (arrondissement) de Manhattan.

*Indication : Utilisez le champ borough.*

```
db.restaurants.find( { borough : 'Manhattan' } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

1. Rechercher les restaurants dont le nom contient exactement “Domino’s Pizza”.

*Indication : Utilisez le champ name.*

```
db.restaurants.find( { name : "Domino's Pizza" } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

*// operateur regex*

*// \$regex c'est pour écrire le pattern de recherche ^ ça commence et \$ ça se termine*

*// \$options de la recherche ici i désigne le fait que c'est insensible à la casse*

```
db.restaurants.find( { name : { $regex: /^D/ , $options : "i" } } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

*// ça contient la lettre d ou D*

```
db.restaurants.find( { name : { $regex: /D/ , $options : "i" } } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

*// sensible à la casse*

```
db.restaurants.find( { name : { $regex: /D/ } } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

*// recherche insensible à la casse pour trouver les restaurants*

```
db.restaurants.find( { name : { $regex: /domino's pizza/ , $options : "i" } } , { _id : 0, name : 1 , 'address.coord' : 1 } )
```

1. Rechercher les restaurants ayant obtenu un score supérieur ou égal à 90 dans leurs inspections.

*Indication : Utilisez le champ grades.score.*

*// il suffit d'un score supérieur ou égale à 90 pour sélectionner le document.*

```
db.restaurants.find( {
  "grades.score" : { $gte: 90 } } ,
  { _id : 0, name : 1 , 'address.coord' : 1, "grades.score" : 1 }
)
```

Rechercher les restaurants dont tous les score(s) est/sont supérieur(s) ou égal(s) à 10 dans leurs inspections.

```
db.restaurants.find(
  { grades :
    {
```

```

    $not : { $elemMatch : { score : { $lt: 10 } } }
  }
},
{ _id : 0, name : 1, 'address.coord' : 1, "grades.score" : 1 }
)

```

1. Rechercher tous les restaurants qui ont une adresse située sur la rue “Broadway”.

*Indication : Utilisez le champ `address.street`.*

```

db.restaurants.find(
  { "address.street" : "Broadway" },
  { _id : 0, name : 1, 'address.street' : 1, "grades.score" : 1 }
)

```

2. Rechercher tous les restaurants avec une cuisine autre que “American”.

*Indication : Utilisez l’opérateur `$ne` pour exclure la cuisine américaine.*

```

db.restaurants.find(
  { cuisine : { $ne : "American" } },
  { _id : 0, name : 1, 'address.street' : 1, cuisine : 1 }
)

```

*// on peut aussi faire une black list avec American et Hamburgers on utilise l’opérateur `$nin`*

```

db.restaurants.find(
  { cuisine : { $nin : ["American", "Hamburgers" ] } },
  { _id : 0, name : 1, 'address.street' : 1, cuisine : 1 }
)

```

1. Rechercher tous les restaurants ayant été inspectés avant le 1er janvier 2020.

*Indication : Utilisez le champ `grades.date` et un opérateur de comparaison comme `$lt`.*

```

db.restaurants.find(
  { "grades.date" : { $lt : new Date("2020-01-01" ) } } // on peut utiliser new Date de Java
)

```

```

db.restaurants.find(
  { "grades.date" : { $lt : ISODate("2020-01-01" ) } } // on peut utiliser ISODate type MongoDB
)

```

1. Rechercher tous les restaurants situés dans le borough de Queens ou de Brooklyn.

*Indication : Utilisez l’opérateur `$in` avec le champ `borough`.*

```

db.restaurants.find(
  { borough : { $in : ["Queens", "Brooklyn"] } },

```

```
{ _id : 0, name : 1 , borough : 1, cuisine : 1}
)
```

2. Rechercher tous les restaurants dont le nom commence par la lettre "S".

*Indication : Utilisez une expression régulière avec le champ **name**.*

```
db.restaurants.find(
  { name : { $regex : /^s/, $options : "i" } },
  { _id : 0, name : 1 , borough : 1, cuisine : 1}
)
```

1. Rechercher tous les restaurants ayant au moins une note inférieure à 50.

*Indication : Utilisez un opérateur comme **\$elemMatch** pour examiner les éléments de **grades**.*

```
db.restaurants.find(
  { grades : { $elemMatch : { score : { $lt : 50 } } } },
  { _id : 0, name : 1 , "grades.score" : 1}
)
```

Indications, par exemple on peut utiliser cet opérateur pour faire une recherche avancée :

```
$elemMatch: { "grades.score": { $gt: 90 }, name: { $regex : /^D/} }
```