

NoSQL

Module: 165

Utiliser des bases de données
NoSQL

Base de données NoSQL

Objectifs

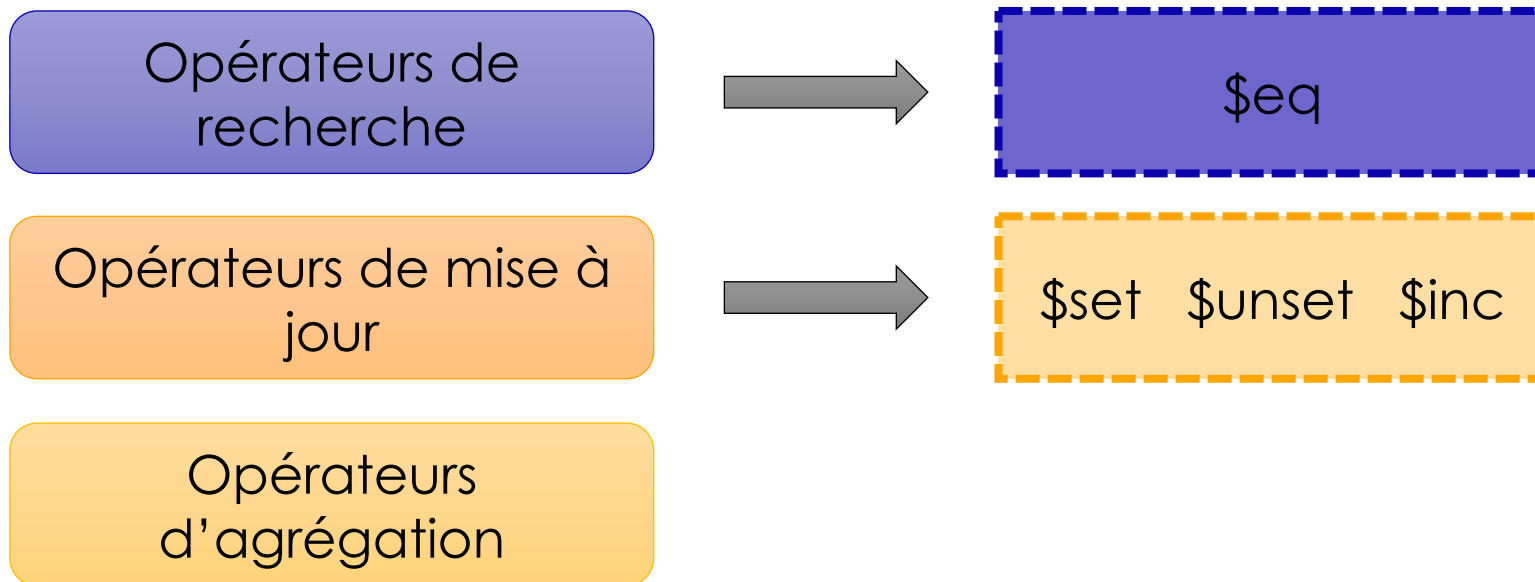
2. Implémenter une base de données NoSQL et y insérer les données.

MongoDB

Les types d'opérateurs de recherche

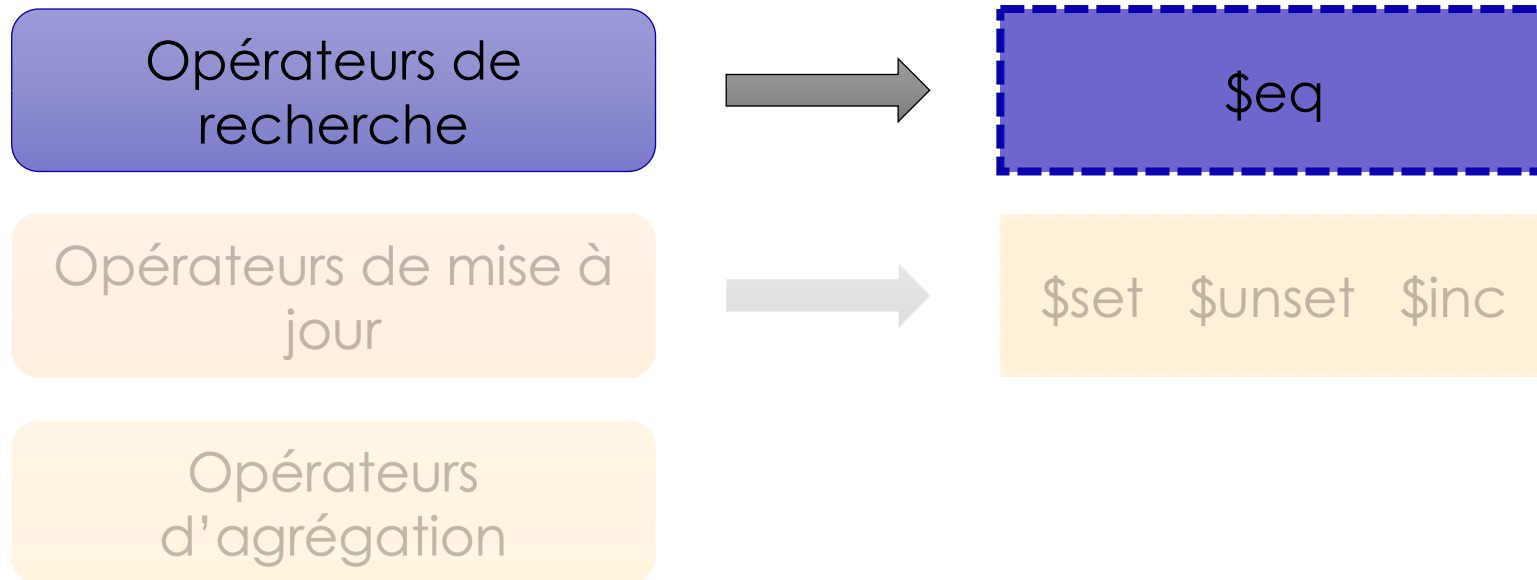
MongoDB

Les types d'opérateurs



MongoDB

Les opérateurs de recherche



MongoDB

Les opérateurs

Opérateurs de
recherche

Opérateurs de comparaison : **\$eq, \$ne, \$gt, \$gte, \$lt, \$lte, \$in, \$nin**

Opérateurs logiques : **\$and, \$or, \$nor, \$not**

Opérateurs d'évaluation : **\$expr**

Opérateurs d'élément : **\$exists, \$type**

MongoDB

Les opérateurs de comparaison

MongoDB

Les opérateurs de comparaison

Opérateurs de
recherche

Opérateurs de comparaison : **\$eq, \$ne, \$gt, \$gte, \$lt, \$lte, \$in, \$nin**

Opérateurs logiques : **\$and, or, \$nor, \$not**

Opérateurs d'évaluation : **\$expr**

Opérateurs d'élément : **\$exists, \$type**

MongoDB

Les opérateurs de comparaison

```
use('sample_mflix');
db.movies.find(
  { 'awards.wins': { $eq: 55 } }
)
```

L'opérateur **\$eq** (pour égalité, en anglais equality) permet de sélectionner les documents dont la valeur d'un champ est égale à la valeur précisée.

L'opérateur **\$ne** (pour not equal) permet de sélectionner les documents dont la valeur d'un champ n'est pas égale à la valeur précisée.

```
use('sample_mflix');
db.movies.countDocuments(
  { 'awards.wins': { $ne: 0 } }
)
```

MongoDB

Les opérateurs de comparaison

Les opérateurs **\$in** (pour dans) et **\$nin** (pour pas dans) permettent de sélectionner les documents où la valeur du champ se trouve dans le tableau spécifié, ou au contraire dont la valeur du champ n'est pas dans le tableau spécifié.

```
use('sample_mflix');
db.movies.countDocuments(
  { runtime: { $in: [60, 90, 120] } }
)
```

```
use('sample_mflix');
db.movies.countDocuments(
  { year: { $nin: [1983, 2004, 2008] } }
)
```

MongoDB

Les opérateurs logiques

MongoDB

Les opérateurs logiques

Opérateurs de
recherche

Opérateurs de comparaison : **\$eq, \$ne, \$gt, \$gte, \$lt, \$lte, \$in, \$nin**

Opérateurs logiques : **\$and, or, \$nor, \$not**

Opérateurs d'évaluation : **\$expr**

Opérateurs d'élément : **\$exists, \$type**

MongoDB

Les opérateurs logiques

```
use('sample_mflix');
db.movies.countDocuments(
  {
    'imdb.rating': { $gt: 9 },
    'tomatoes.viewer.rating': { $gte: 4 }
  }
)
```

En conséquence, on utilise rarement l'opérateur. Le seul cas d'utilisation est lorsque l'on doit utiliser deux fois un autre opérateur logique.

L'opérateur **\$and** permet de cumuler plusieurs conditions qui sont spécifiées dans un tableau.

```
use('sample_mflix');
db.movies.countDocuments(
  {
    $and: [
      {'imdb.rating': { $gt: 9 }},
      {'tomatoes.viewer.rating': { $gte: 4 }}
    ]
  }
)
```

MongoDB

Les opérateurs logiques

L'opérateur **\$not** permet de spécifier une expression qui ne doit pas être valide dans un tableau.

```
use('sample_mflix');
db.movies.countDocuments(
  {
    'imdb.rating': { $not: { $lte: 9 } }
  }
)
```

Attention ! Ce n'est pas la même chose que **{ \$gt: 9 }**, car les documents qui n'ont pas le champ spécifié sont aussi sélectionnés.

MongoDB

Les opérateurs d'évaluation

MongoDB

Les opérateurs d'évaluation

Opérateurs de
recherche

Opérateurs de comparaison : **\$eq, \$ne, \$gt, \$gte, \$lt, \$lte, \$in, \$nin**

Opérateurs logiques : **\$and, or, \$nor, \$not**

Opérateurs d'évaluation : **\$expr**

Opérateurs d'élément : **\$exists, \$type**

MongoDB

Les opérateurs d'évaluation

L'opérateur **\$regex** permet d'utiliser des expressions régulières, également appelées expressions rationnelles.

```
use('sample_mflix');
db.movies.countDocuments(
  {
    title: { $regex: /^The land/, $options: 'i' }
  }
)
```

Les deux principales options disponibles sont :

- **i** pour activer l'insensibilité à la casse (insensible aux majuscules / minuscules).
- **s** pour que le caractère . sélectionne tous les caractères incluant les caractères de nouvelle ligne.

MongoDB

Les opérateurs d'élément

MongoDB

Les opérateurs d'élément

Opérateurs de
recherche

Opérateurs de comparaison : **\$eq, \$ne, \$gt, \$gte, \$lt, \$lte, \$in, \$nin**

Opérateurs logiques : **\$and, or, \$nor, \$not**

Opérateurs d'évaluation : **\$expr**

Opérateurs d'élément : **\$exists, \$type**

MongoDB

Les opérateurs d'élément

L'opérateur **\$exists** permet de sélectionner les documents qui ont (true) ou non (false) le champ spécifié.

```
use('sample_mflix');
db.movies.countDocuments(
  {
    'tomatoes.viewer.rating': { $exists: false }
  }
)
```

Attention ! Notez que l'existence ne vérifie pas la valeur mais uniquement l'existence du champ spécifié. Ainsi, même si la valeur est **null**, le document sera sélectionné avec **{ \$exists: true }**

MongoDB

Les opérateurs d'élément

L'opérateur **\$type** permet de sélectionner les documents où la valeur d'un champ est d'un type **BSON** spécifié.

Vous pouvez utiliser le numéro du type **BSON** ou son **alias** par exemple **string** ou **double**.

```
use('sample_mflix');
db.movies.find(
  {
    'imdb.rating': { $type: 2 }
  }
)
```

```
use('sample_mflix');
db.movies.find(
  {
    'imdb.rating': { $type: 'string' }
  }
)
```

Retrouvez tous les types **BSON** ici:

<https://docs.mongodb.com/manual/reference/bson-types/>

MongoDB

Les opérateurs d'élément

Pour vérifier qu'un champ a le type **null** ou 10, il faut faire :

```
use('sample_mflix');
db.movies.find(
  {
    'tomatoes': null
  }
)
```

```
use('sample_mflix');
db.movies.find(
  {
    'imdb.rating': { $type: 'null' }
  }
)
```

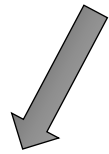
Faites très attention **champ: null** retournera les documents dont la valeur du champ est **null** ET les documents qui n'ont pas le champ.

MongoDB

Les opérateurs sur les tableaux

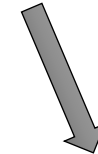
MongoDB

Les opérateurs sur les tableaux



Opérateurs de recherche

\$all, \$elemMatch, \$size



Opérateurs de mise à jour

**\$push, \$pullAll, \$pop, \$pull,
\$addToSet, \$**

MongoDB

Les opérateurs sur les tableaux

L'opérateur **\$all** permet de rechercher les tableaux qui contiennent plusieurs éléments spécifiés.

```
use('sample_mflix');
db.movies.countDocuments(
  {
    genres: {$all: ['Drama', 'Comedy']}
  }
)
```

MongoDB

Les opérateurs sur les tableaux

L'opérateur **\$elemMatch** permet de rechercher les tableaux qui contiennent un élément répondant à plusieurs conditions.

```
use('sample_supplies');
db.sales.countDocuments(
  {
    items: { $elemMatch: { name: "printer paper", quantity: { $gte: 9 } } }
  }
)
```

Cela comptera les documents qui ont un tableau *items* où au moins un élément a pour le champ *name* la valeur "printer paper" ET une valeur supérieure ou égale à 9 pour le champ *quantity*.

MongoDB

Les opérateurs sur les tableaux

```
use('school');
db.students.findOneAndUpdate({},
  {
    $push: {
      grades: {
        $each: [{score: 15}, {score: 18}],
        $sort: {score: -1},
        $slice: 5
      },
    },
  },
  {returnDocument: 'after'}
)
```

Vous pouvez bien sûr utiliser les trois modificateurs pour effectuer des opérations complexes.

MongoDB

Les opérateurs sur les tableaux

Pour supprimer toutes les valeurs spécifiées d'un tableau il faut utiliser l'opérateur **\$pullAll**.

```
use('school');
db.students.findOneAndUpdate({},
  {
    $pullAll: {
      grades: [{score: 12}, {score: 13}],
    }
  },
  {returnDocument: 'after'}
)
```

Cela retirera tous les éléments **{score: 12}** et **{score: 13}** du tableau *grades*.

MongoDB

Les opérateurs de mise à jour

MongoDB

Les opérateurs de mise à jour

\$inc

\$set

\$unset

Tableau

\$push

\$addToSet

\$pop

\$pull

\$pullAll

\$. \$[]

\$currentDate

\$max

\$min

\$rename

\$mul

MongoDB

Les projections

MongoDB

Les projections

Une projection permet de restreindre les champs retournés.

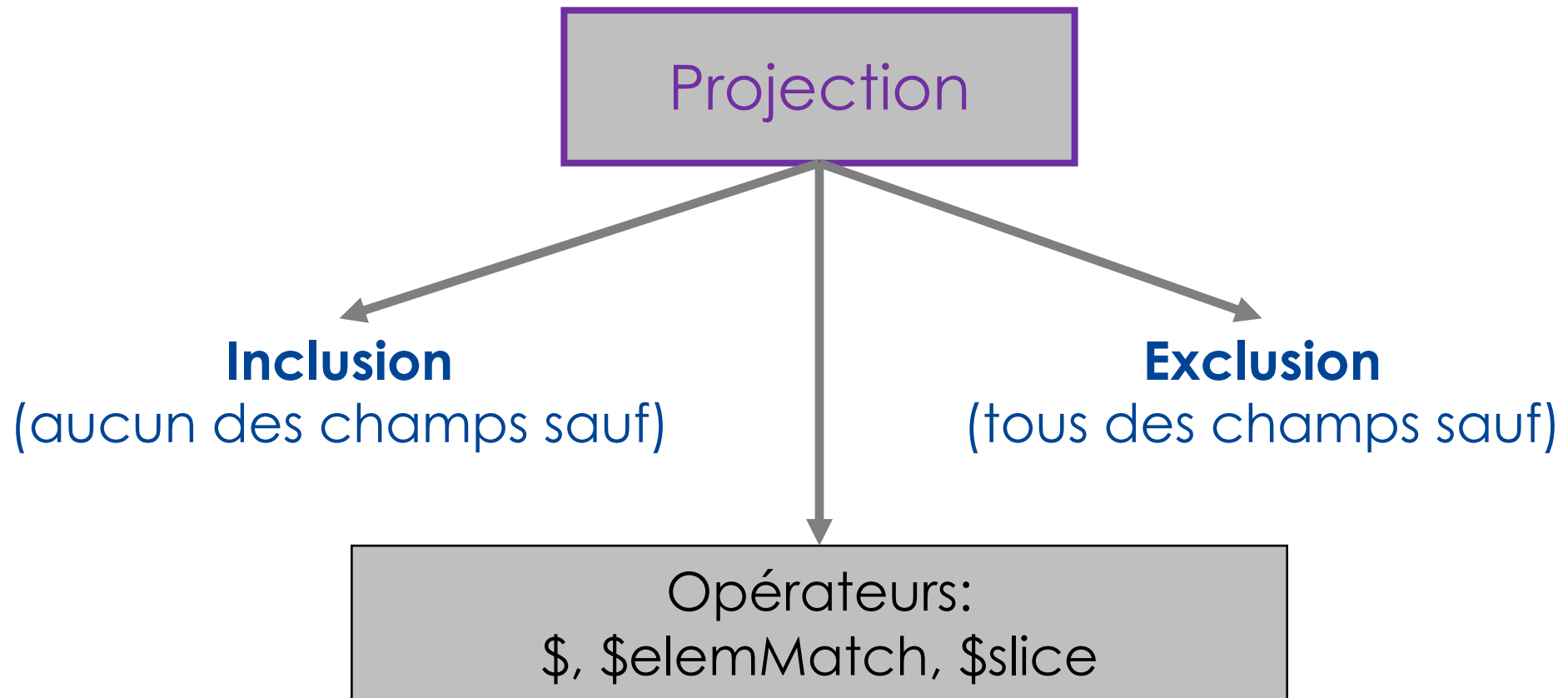
```
find(query, projection)
```

```
findOne(query, projection)
```

```
findOneAndUpdate(query, update, { projection: {} })
```


MongoDB

Les projections



MongoDB

Les projections: syntaxe

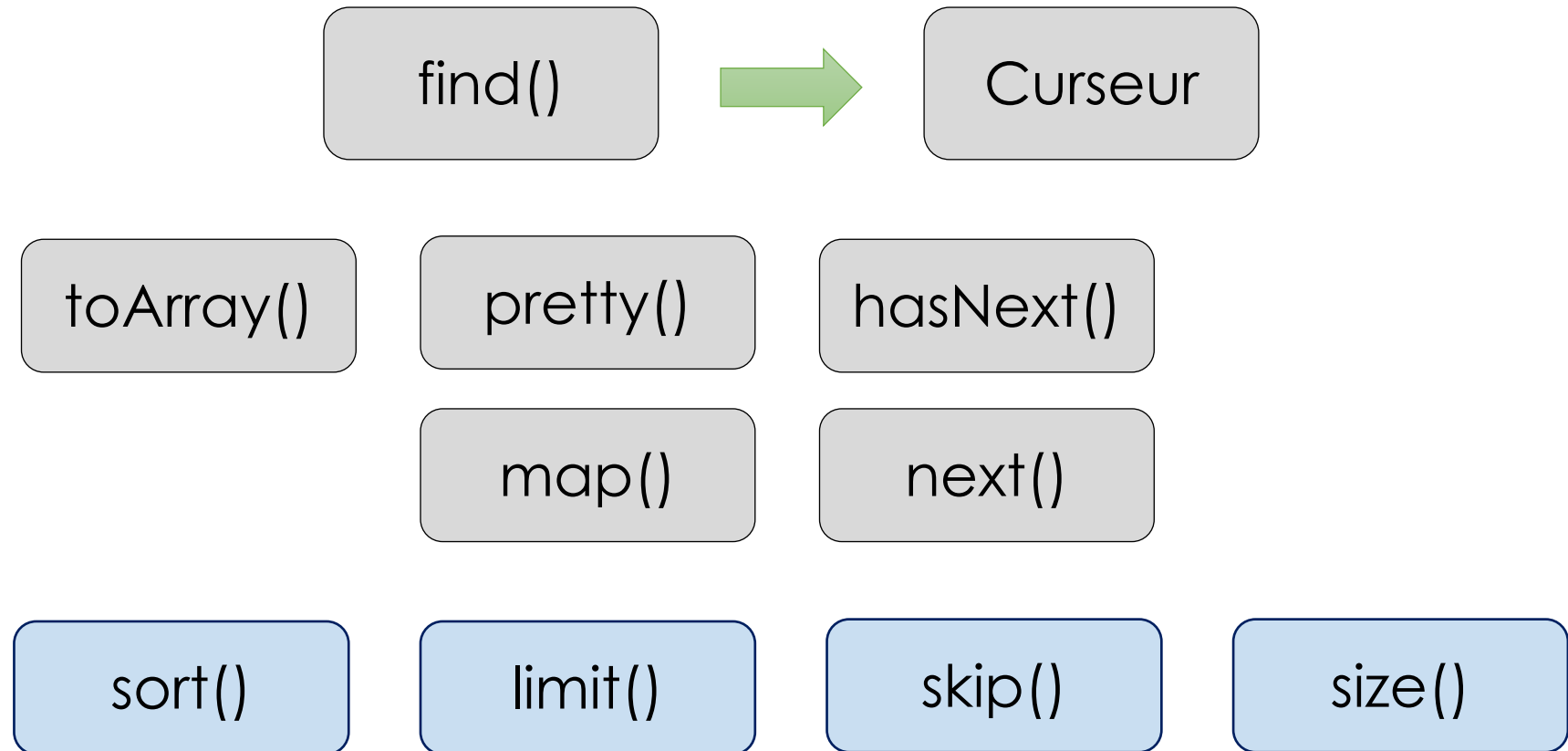
```
use('sample_mflix');
db.movies.findOneAndUpdate(
  { _id: ObjectId("573a1390f29313caabcd63d6") },
  { $inc: { runtime: 2 } },
  {
    projection: { title: 1, runtime : 1 },
    returnDocument: 'after'
  }
);
```

MongoDB

Les méthodes de curseur

MongoDB

Les méthodes de curseur



MongoDB

La méthode de curseur

La méthode **sort()** permet de modifier l'ordre dans lequel la requête va retourner les documents.

La méthode **limit()** permet de limiter le nombre de documents retournés par le curseur.

```
use('sample_mflix');
db.movies.find(
  {
    'imdb.rating': {
      $ne: ""
    }
  }
).sort({'imdb.rating': -1, 'tomatoes.viewer.rating': -1, '_id': 1}).limit(2);
```

MongoDB

La méthode de curseur

La méthode **skip()** permet de contrôler où MongoDB va commencer à retourner des résultats.

```
use('sample_mflix');  
db.movies.find()  
  .sort( { title: 1, _id: 1 } )  
  .skip( (50 - 1) * 20 )  
  .limit( 20 )
```

Le cas d'utilisation le plus commun de **skip()** est la mise en place d'une pagination.

La méthode **size()** permet de retourner le nombre de documents qui sont retournés dans le curseur.

MongoDB

Exercice de recherche avancé

- E-165-MongoDB-Query-Exercise-I.pdf
- db_restaurant.gz

Importez la collection **restaurants** dans la base de données **db_restaurant**.

Puis exécuter les exercices de recherche des documents de la collection.