TD MongoDB

3IF

Előd EGYED-ZSIGMOND

elod.egyed-zsigmond@insa-lyon.fr

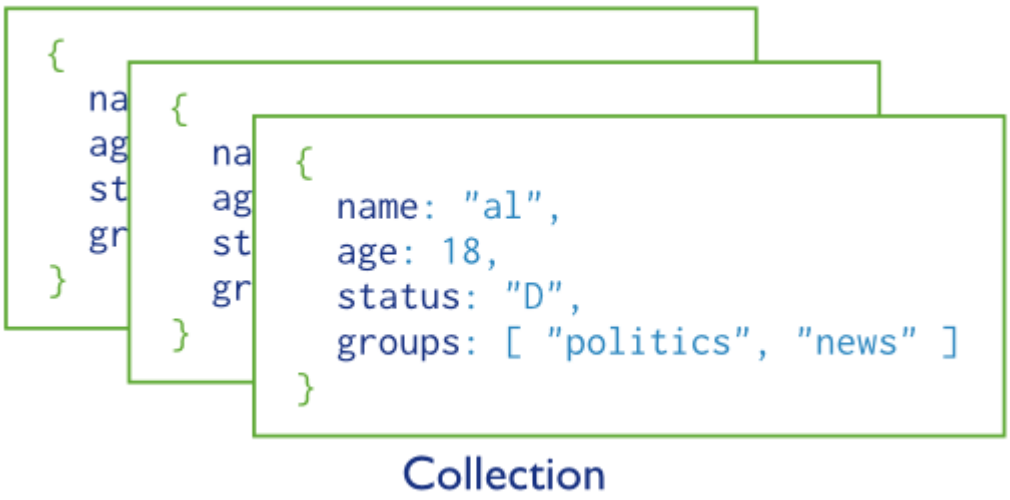
Plan

- Introduction
- Requêtes CRUD
- Agrégats
- Indexes

Vocabulaire

- Document

- Collection



http://docs.mongodb.org/manual/core/crud-introduction/

Pratique

MongoDB vient avec un shell : bin/mongo

Démarrage avec : bin/mongod

Quelques arguments :

--dbpath <path> : Chemin de stockage des données.

--port <port> : Port du serveur

--replSet <nom> : Introduire le serveur dans un cluster de réplicas.

Shell

- Afficher la base de données courante :
  - `db`
- Afficher la liste des bases de données :
  - `show dbs`
- Sélectionner une base de données :
  - `use <name>`
- Afficher les collections :
  - `show collections`

Chargement de données

- Charger des documents présents dans les fichiers : movies_refs.json et artists.json dans la base MongoDB :

```
mongoimport --host localhost:27017 --db test --collection
  films < movies_refs.json –jsonArray
```

```
mongoimport --host localhost:27017 --db test --collection
  Artists < artists.json --jsonArray
```

Requêtes dans MongoDB

- Requêtes en JSON
- On retrouve l'équivalent des projections de SQL
- Quelques agrégations par défaut

… Map/Reduce pour les autres

Plan

Introduction

Requêtes CRUD

Agrégats

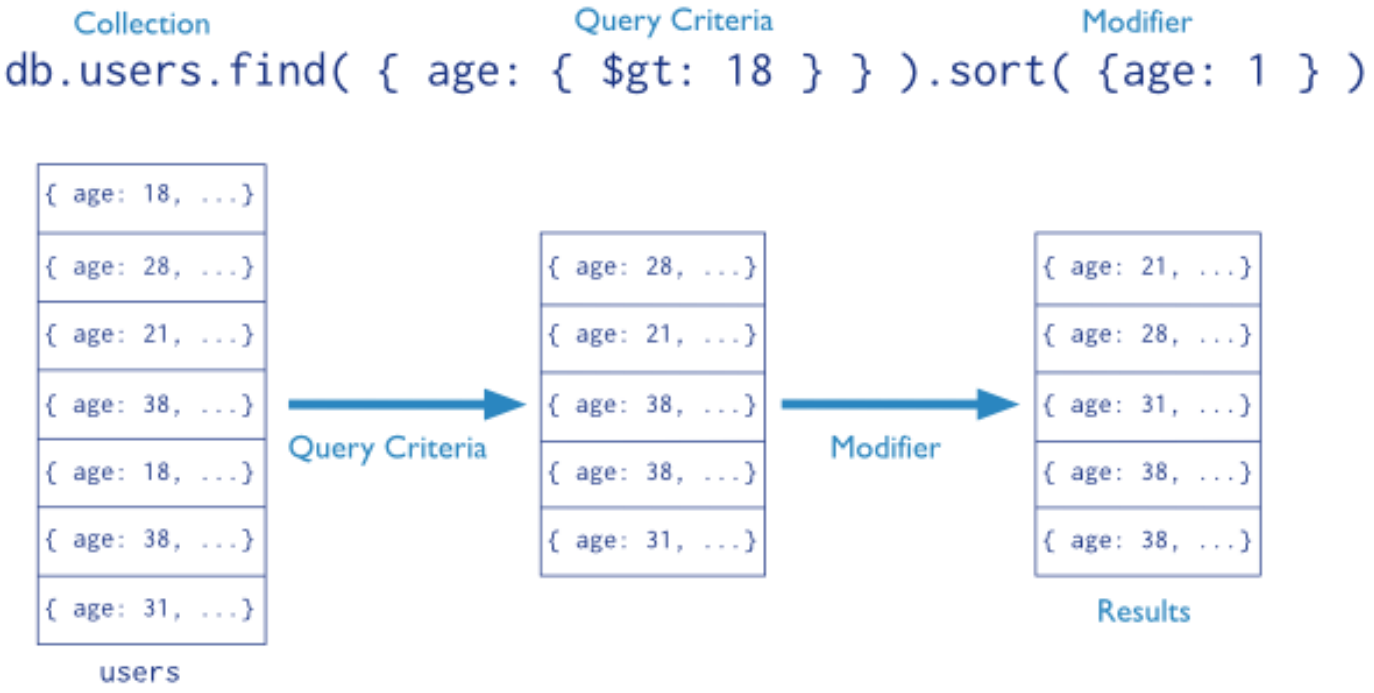
Indexes

- Requêtes en JSON
- On retrouve l'équivalent des projections de SQL
- Quelques agrégations par défaut

… Map/Reduce pour les autres

CRUD

Requêtes



http://docs.mongodb.org/manual/

CRUD

Requêtes

```
db.users.find(                          ← collection
   { age: { $gt: 18 } },                ← query criteria
   { name: 1, address: 1 }              ← projection
).limit(5)                              ← cursor modifier
```

En SQL

```
SELECT _id, name, address   ← projection
FROM   users                ← table
WHERE  age > 18             ← select criteria
LIMIT  5                    ← cursor modifier
```

```
db.films.find({"country": {$ne:"USA"}},{_id:0,title:1}).limit(5)
```

http://docs.mongodb.org/manual/

Opérateurs

$eq	=	valeurs qui sont égales à une valeur spécifiée.
$gt	>	valeurs qui sont supérieures à une valeur spécifiée.
$gte	>=	à des valeurs qui sont supérieures ou égales à une valeur spécifiée.
$lt	<	valeurs qui sont inférieures à une valeur spécifiée.
$lte	<=	valeurs qui sont inférieures ou égales à une valeur spécifiée.
$ne	!=	toutes les valeurs qui ne sont pas égales à une valeur spécifiée.
$in	∈	une des valeurs spécifiées dans un tableau.
$nin	∉	aucune des valeurs spécifiées dans un tableau.
$or		
$and		
$not		
$nor		
…		

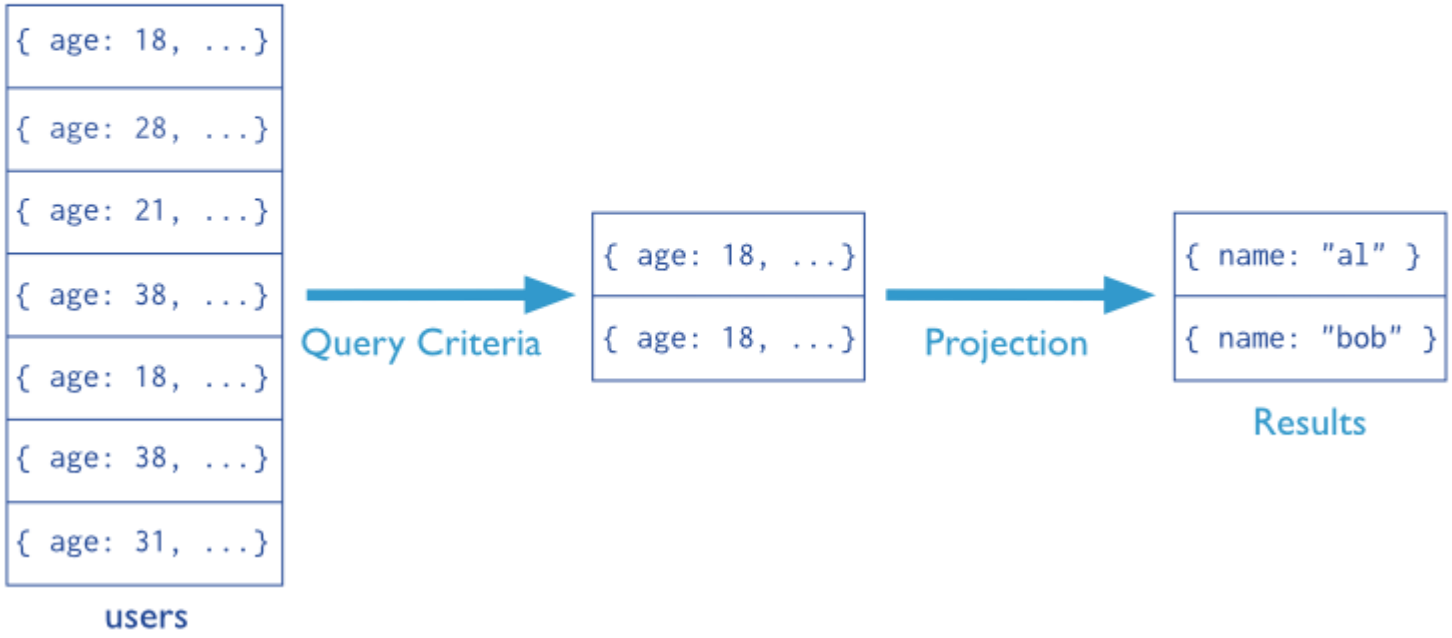
http://docs.mongodb.org/manual/reference/operator/query/

CRUD

Projections

Collection Query Criteria Projection

```
db.users.find( { age: 18 }, { name: 1, _id: 0 } )
```




```
db.films.find({"country":{$ne:"USA"}},{_id:0,title:1})
```

CRUD

Exemples

Exclure l'année :

```
db.films.find({"country":"USA"},{year:0})
```

Retourner le titre et le genre et le _id

```
db.films.find({"country":"USA"},{title:1,genre:1})
```

Retourner les films possédant un rôle donné

```
db.films.find({"actors.role":"William Munny"})
```

Retourner que le titre et le genre

```
db.films.find({"country":"USA"},{title:1,genre:1, _id:0})
```

Trier les résultats

```
db.films.find({"year": {$gt: 2000}}).sort({year:-1})
```

CRUD

Exemples

```
db.films.find({$and : [{"year": 1992}, {"country" : "USA"}]} )
```

```
db.films.find({$or : [{"year": 1992}, {"country" : "USA"}]} )
```

CRUD

Curseurs

```
db.collection.find()
db.films.find({"country": "USA"})
```

Parcours un à un des résultats

```
var myCursor = db.films.find( { country: 'FR' } );
while (myCursor.hasNext()) {
   print(tojson(myCursor.next()));
}

var myCursor = db.films.find( { country: 'FR' } );
myCursor.forEach(printjson);
```

CRUD

Curseurs, méthodes

- *limit()*: pour récupérer les n premiers résultats uniquement
- *sort()*: pour trier les résultats
- *skip()*: pour "sauter" n résultats

Par exemple, pour avoir l'antépénultième film selon le nom, on pourrait faire:

```
var myCursor = db.films.find( {  } );
myCursor.sort( {title: -1} ).limit(1).skip(3);
```

CRUD

Curseurs

Transformation en tableau

```
var lesFilms = db.films.find( { country: "FR" } ).toArray()
printjson (lesFilms);
```

Analyser la performance des requêtes

```
db.films.find( { country: 'FR' } ).explain();
db.films.createIndex( { country: 1 } )
```

Jointure en javascript

```
var myCursor = db.films.find({country:"FR"});
while (myCursor.hasNext()) {
   film = myCursor.next();
   print(film.title)
   film.actors.forEach(function (actor){
        //print(actor);
        var l_artist_c = db.Artists.find({"_id" : actor.old_id});
        var artist_name = l_artist_c.hasNext() ?
  l_artist_c.next().last_name : "";
        print ("- "+artist_name);
   });
}
```

CRUD

Modification:

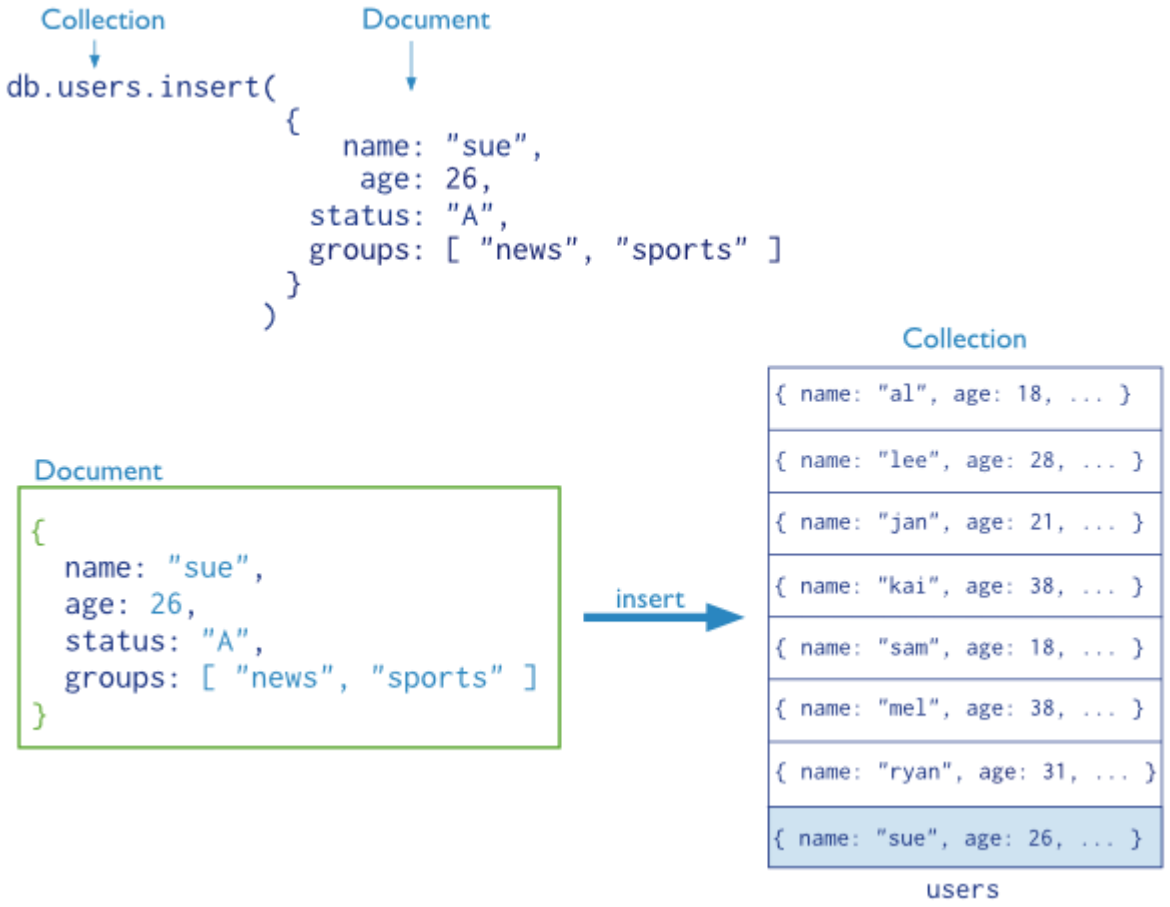
- Insert
- Update
- Remove

Les opérations s'appliquent sur une collection

http://docs.mongodb.org/manual/

CRUD

Insertion



Le champ "_id" est ajouté automatiquement

http://docs.mongodb.org/manual/

CRUD

Update

```
db.collection.update(
    <query>,
    <update>,
    {
      upsert: <boolean>,
      multi: <boolean>,
      writeConcern: <document>
    }
)
```

Mêmes contraintes que pour find

$set, $unset, …

```
db.users.update(
   { age: { $gt: 18 } },
   { $set: { status: "A" } },
   { multi: true }
)
```

collection

update criteria

update action

update option

par défaut la mise à jour concerne un document

http://docs.mongodb.org/manual/

CRUD

Update

```
db.films.update({"year": {$lt: 2000}},{$set : {old:true}})

db.films.find({old:true})

db.films.update({"year": {$lt: 2000}},{$unset : {old:true}})

db.films.update({"year": {$lt: 2000}},{$set : {old:true}},
{multi:true})
```

CRUD

Update, mise à jour d'un tableau

- On va ajouter un tableau à un document

```
db.produits.insert({compteur:100001, tab:['a','b','c']})
```

- Vous pouvez le consulter avec la commande suivante:

```
db.produits.find({compteur:100001})
```

- Maintenant pour ajouter un élément, on utilise l'opérande **$push**:

```
db.produits.update({compteur:100001}, {$push : {tab : 'd'}})
```

Pour en ajouter plusieurs d'un coup, il existe $**pushAll** pour ajouter un tableau de valeur.

CRUD

Update, mise à jour d'un tableau

- Avec l'opérande **$pop** on va supprimer le dernier élément:

```
db.produits.update({compteur:100001}, {$pop : {tab:1}})
```

- Le tableau a perdu son dernier élément "d".
- Pour supprimer le premier élément:

```
db.produits.update({compteur:100001}, {$pop : {tab:-1}})
```

- De façon analogue avec la méthode sort, en mettant -1 on supprime les éléments dans l'autre sens.
- Avec l'opérande **$addToSet**, on ajoute sans doublon :

```
db.produits.update({compteur:100001}, {$addToSet : {tab :
  'b'}})
```

- La valeur "b" était présente dans le tableau donc il n'y pas eu d'ajout.

CRUD

- **Supprimer un document**
- La suppression d'un document se fait grâce à la méthode remove.

Par exemple:

```
db.produits.remove({mill: 600})
```

- Supprime tous les documents qui ont pour valeur 600 à la propriété *mill*.

Plan

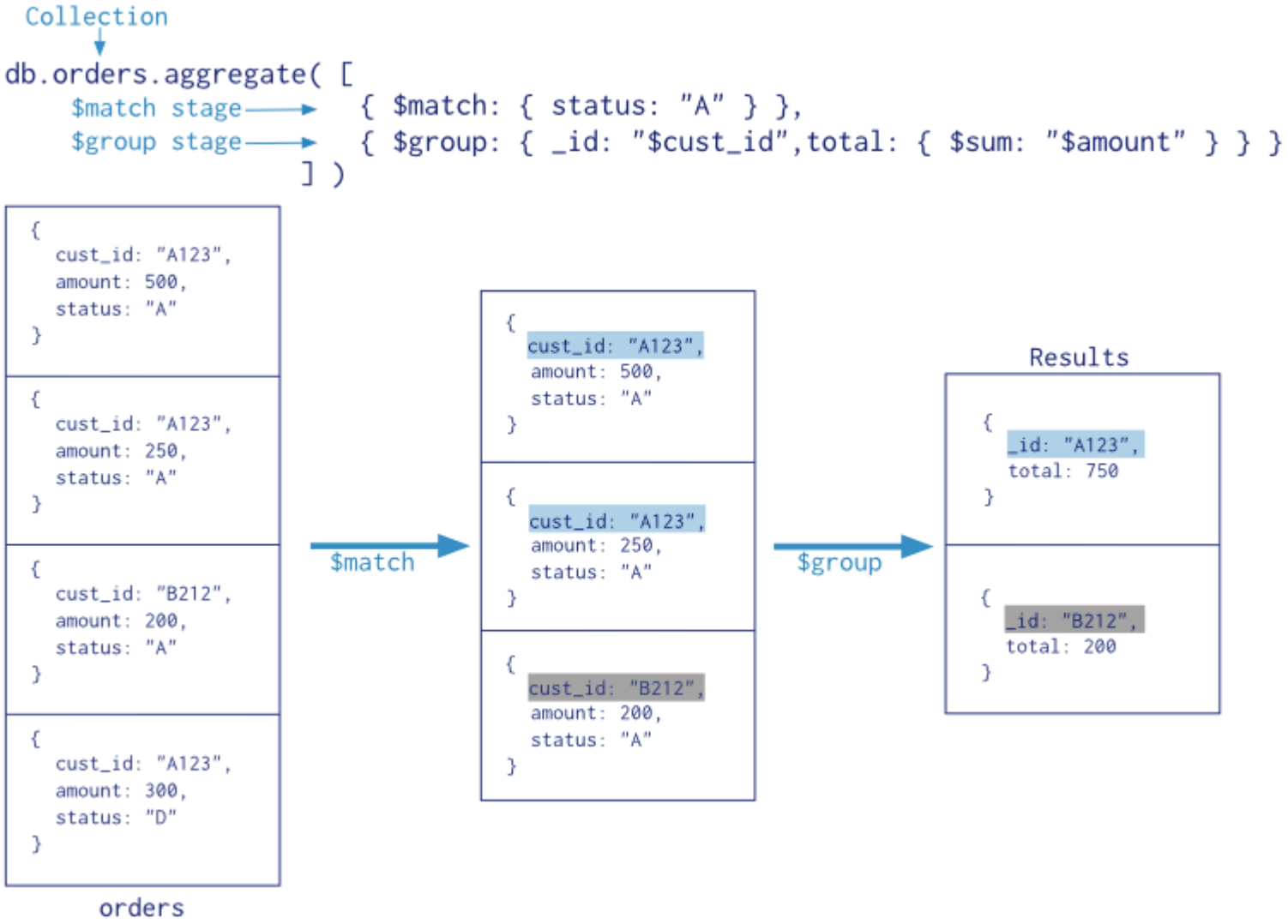
Introduction

Requêtes CRUD

Agrégats

Indexes

Agrégats



https://docs.mongodb.org

Agrégats

Nb de films par genre

```
db.films.aggregate([
  {$group:{_id:"$genre",count:{$sum:1}}}
])
```

Nb de films américains par genre

```
db.films.aggregate([
{$match:{country:"USA"}},
{$group:{_id:"$genre",count:{$sum:1}}}
])
```

Date du premier film d'horreur

```
db.films.aggregate([
{$match:{genre:"Horreur"}},
{$group:{_id:"$genre",debut:{$min:"$year"}}}
])
```

Plan

Introduction

Requêtes CRUD

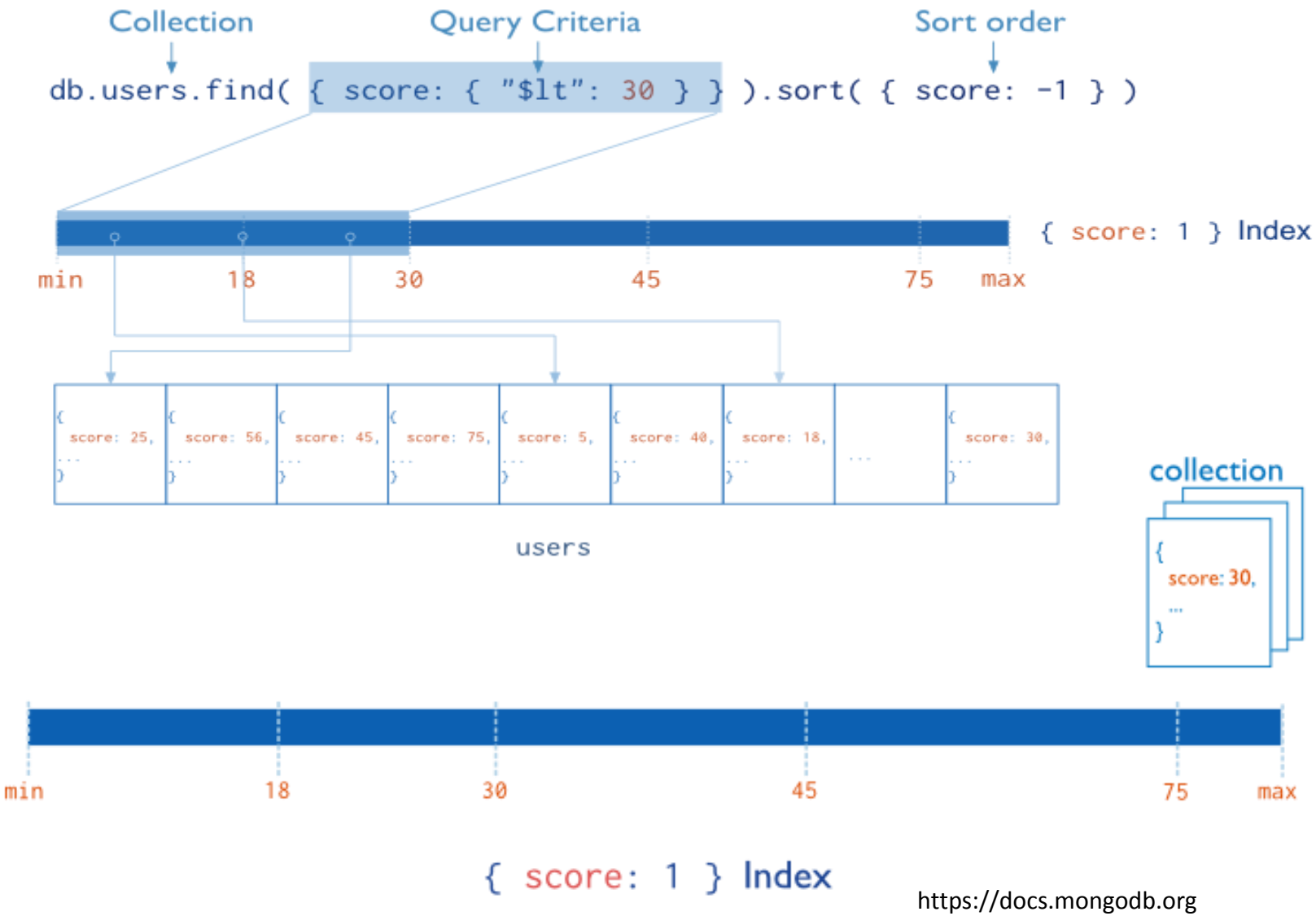
Agrégats

Indexes

Indexes

- Très similaire aux SGBDR, l'indexation dans MongoDB se fait sur un ou plusieurs champs.
- Permet d'améliorer les performances de recherche.
- Les indexes sont stockés au niveau des collections.
- Apporte une surcharge pour les opérations d'écriture.
- Le fonctionnement interne est très proche de ce que l'on trouve dans les SGBD actuels.

Indexes



https://docs.mongodb.org

Indexes

Types d’indexes

- Champ unique
- Champ composé
- Multi clef
- Géo spatiale
- Texte
- Haché

Indexes

- Penser à utiliser les indexes de manière efficace.
- Un champ peu requêté n'a aucun intérêt à être indexé
- Bien que l'on parle de NoSQL, le fonctionnement des indexes est similaire au monde SQL.

Insertion des documents

- Insert unitaire:
  - Exemple: coll.insert({x:1,y:2,z:"test"})
  - Commande à éviter pour des insertions en masse via un client (ordre de grandeur: > 10 000 documents)
- Bulk insert:
  - Exemple:
    var documents = [];
    for(var i = 1; i < 10 000; i++) { documents.push({x:i,y:i, z:"test"})};
    db.coll.insert(documents);

Insertion massive de documents

- Utilitaire pour charger un fichier BSON contenant des centaines de GB ou plus dans une base MongoDB: mongorestore
- Mongorestore: méthode de loin la plus efficace pour insérer des données (autres méthodes: (bulk) insert, mongoimport)
- Outil de sauvegarde/restauration d'une base MongoDB (shardée) avec mongodump

GridFS

- GridFS est une spécification pour stocker et retrouver des fichiers de plus de 16 MB.
- Les fichiers sont splittés en chunks et stockés dans différents documents.
- Deux collections sont utilisées pour stocker d'un côté les chunks et de l'autre les méta-données.