

Importer le fichier json ci-joint dans votre base de données.

J'ai fait l'import via MongoDBCompass

Afficher les films de l'année 2010

```
db.movies.find({year:2010}).pretty()
```

Combien de film a été réalisé en 2009.

```
> db.movies.find({year: 2009}).count()
229
> |
```

Faites une requête pour récupérer des documents des films réalisés en 2008 en excluant les champs :

_id, cast, genres.

```
> db.movies.find({year:2008},{_id:1,cast:1,genres:1}).pretty()
{
  "_id" : ObjectId("60084d65140b994b9c5da423"),
  "cast" : [
    "Steven Strait",
    "Camilla Belle",
    "Cliff Curtis"
  ],
  "genres" : [
    "Adventure"
  ]
}
{
  "_id" : ObjectId("60084d65140b994b9c5da424"),
  "cast" : [
    "Jim Sturgess",
    "Kevin Spacey",
    "Kate Bosworth",
    "Laurence Fishburne"
  ],
  "genres" : [
    "Drama"
  ]
}
{
  "_id" : ObjectId("60084d65140b994b9c5da425"),
  "cast" : [
    "Katherine Heigl",
```

Afficher tous les films commençant par la lettre A.

```
> db.movies.find({title: /^A/}).pretty()
{
  "_id" : ObjectId("60084d61140b994b9c5d3d56"),
  "title" : "After Dark in Central Park",
  "year" : 1900,
  "cast" : [ ],
  "genres" : [ ]
}
{
  "_id" : ObjectId("60084d61140b994b9c5d3d67"),
  "title" : "Acrobats in Cairo",
  "year" : 1901,
  "cast" : [ ],
  "genres" : [ ]
}
{
  "_id" : ObjectId("60084d61140b994b9c5d3d68"),
  "title" : "An Affair of Honor",
  "year" : 1901,
  "cast" : [ ],
  "genres" : [ ]
}
{
  "_id" : ObjectId("60084d61140b994b9c5d3d69"),
  "title" : "Another Job for the Undertaker",
  "year" : 1901,
  "cast" : [ ],
  "genres" : [ ]
}
```

Afficher les films où 'Timothy Gibbs' a joué en 2011

```
> db.movies.find({year:2011},{cast:"Timothy Gibbs",title:1}).pretty()
{
  "_id" : ObjectId("60084d65140b994b9c5da6a8"),
  "title" : "11-11-11",
  "cast" : "Timothy Gibbs"
}
{
  "_id" : ObjectId("60084d65140b994b9c5da6a9"),
  "title" : "30 Minutes or Less",
  "cast" : "Timothy Gibbs"
}
{
  "_id" : ObjectId("60084d65140b994b9c5da6aa"),
  "title" : "35 and Ticking",
  "cast" : "Timothy Gibbs"
}
{
  "_id" : ObjectId("60084d65140b994b9c5da6ab"),
  "title" : "50/50",
  "cast" : "Timothy Gibbs"
}
{
  "_id" : ObjectId("60084d65140b994b9c5da6ac"),
  "title" : "Abduction",
  "cast" : "Timothy Gibbs"
}
```

Afficher les films de type 'Thriller' en 2011

```
> db.movies.find({year:2011,genres:"Thriller"})
{ "_id" : ObjectId("60084d65140b994b9c5da6a8"), "title" : "11-11-11", "year" : 2011, "cast" : [ "Timothy l Landes", "Wendy Glenn", "Benjamin Cook", "Lolo Herrero", "Salome Jimenez", "Brendan Price", "Denis Raf sal", "Lluis Soler" ], "genres" : [ "Horror", "Thriller" ] }
{ "_id" : ObjectId("60084d65140b994b9c5da6ac"), "title" : "Abduction", "year" : 2011, "cast" : [ "Taylor Collins", "Alfred Molina", "Jason Isaacs", "Maria Bello", "Sigourney Weaver" ], "genres" : [ "Action", "Thriller" ] }
{ "_id" : ObjectId("60084d65140b994b9c5da6ad"), "title" : "The Adjustment Bureau", "year" : 2011, "cast" : [ "Emily Blunt", "Anthony Mackie", "Shoreh Aghdashloo", "John Slattery", "Michael Kelly", "Anthony Ruiv Stamp" ], "genres" : [ "Romance", "Science Fiction", "Thriller" ] }
{ "_id" : ObjectId("60084d65140b994b9c5da6b6"), "title" : "Atlas Shrugged: Part I", "year" : 2011, "cast" : [ "Grant Bowler", "Matthew Marsden", "Edi Gathegi", "Graham Beckel", "Jsu Garcia", "Jon Polito", " " ], "genres" : [ "Drama", "Political", "Thriller" ] }
{ "_id" : ObjectId("60084d65140b994b9c5da6ce"), "title" : "Contagion", "year" : 2011, "cast" : [ "Marion tt Damon", "Laurence Fishburne", "Jude Law", "Gwyneth Paltrow", "Kate Winslet", "Bryan Cranston", "Jenni a Lathan", "Amr Waked", "John Hawkes", "Demetri Martin" ], "genres" : [ "Action", "Thriller" ] }
```

Afficher les films de type 'Thriller' réalisés en 2016 par ordre alphabétique inverse de leurs 'title'

```

> db.movies.find({year:2016,genres:"Thriller"}).sort( { "title": -1 } ).pretty()
{
  "_id" : ObjectId("60084d65140b994b9c5dab82"),
  "title" : "Triple 9",
  "year" : 2016,
  "cast" : [
    "Casey Affleck",
    "Aaron Paul"
  ],
  "genres" : [
    "Crime",
    "Thriller"
  ]
}
{
  "_id" : ObjectId("60084d65140b994b9c5dabb7"),
  "title" : "The Shallows",
  "year" : 2016,
  "cast" : [
    "Blake Lively"
  ],
  "genres" : [
    "Horror",
    "Thriller"
  ]
}
{
  "_id" : ObjectId("60084d65140b994b9c5dabb9"),
  "title" : "The Neon Demon",
  "year" : 2016,

```

Insérer deux film de votre choix dans la base en utilisant un BulkWrite (pour le champs 'year', il doit être 2020)

```

db.movies.bulkWrite([ {insertOne: { year: 2020, title: "Café", cast: ["reMacr", "ALiQ", "SMA"], genres: ["MATA", "AKK"]}, | insertOne: { year: 2020, title: "L'ENFER", cast: ["RASS", "MAST
{
  "insertedCount": true,
  "insertedCount": 2,
  "insertedIds":
    [ "0" ObjectId("60084d65140b994b9c5dabb7"),
      "1" ObjectId("60084d65140b994b9c5dabb9") ],
  "matchedCount": 0,
  "modifiedCount": 0,
  "deletedCount": 0,
  "upsertedCount": 0,
  "upsertedIds": {}
}

```

Supprimer tous les films réalisés avant 2000

```
> db.movies.remove({"year": {$lt: 2000}})
WriteResult({ "nRemoved" : 24229 })
> db.movies.find({year: 2000}).count()
213
> db.movies.find({year: 1999}).count()
0
```

Ajouter un champs rating qui sera de type array d'objets pour tous les documents

```
> db.movies.updateMany({},{$set: {"rating": "[]"}})
{ "acknowledged" : true, "matchedCount" : 4568, "modifiedCount" : 4568 }
> |
```

Faites vous et un collaborateur, les ratings (le rating est sur 5) de deux films de votre choix, comme suit : ratings: [{ by: "moi", rating: 4 }, {by:"collaborateur", rating: 5}]

```
db.movies.updateMany({_id:ObjectId("60084d65140b994b9c5d9e01"),_id:ObjectId("60084d65140b994b9c5d9e02")},{$set:{rating:[{by: "My",rating: 2},{by:"myCollege",rating: 5}]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Créer un champ qui sera la moyenne de tous les ratings est appelé le : ar

```

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.movies.updateMany({},{$set: {"ar": "{}"}})
{ "acknowledged" : true, "matchedCount" : 4568, "modifiedCount" : 4568 }
> db.movies.updateMany({},{$set: {"ar": "{}"},{$set: {"rating": "My", "myCol":

```

```

  _id: ObjectId("60084d65140b994b9c5d9e01")
  title: "Evelyn"
  year: 2002
  cast: Array
    0: "Sophie Vavasseur"
    1: "Pierce Brosnan"
    2: "Aidan Quinn"
  genres: Array
    0: "Drama"
  rating: Array
    0: Object
      by: "My"
      rating: 2
    1: Object
      by: "myCollege"
      rating: 5

```

```

  _id: ObjectId("60084d65140b994b9c5d9e02")
  title: "Far from Heaven"
  year: 2002
  cast: Array
    0: "Julianne Moore"
    1: "Dennis Quaid"
    2: "Dennis Haysbert"
  genres: Array
    0: "Drama"
  rating: Array
    0: Object
      by: "My"
      rating: 2
    1: Object
      by: "myCollege"
      rating: 5

```

Renommer le champ créer précédemment à savoir : ar, pour devenir averageRating.

```

> db.movies.updateMany({}, {$rename: {"ar": "averageRating"} })
{ "acknowledged" : true, "matchedCount" : 4568, "modifiedCount" : 4568 }
>

```

Créer un champs views qui sera un array qui contiendra des valeurs comme ci-après

```

db.movies.updateMany({},{$set: {"views": "[123444,66855,78966]"}})
{ "acknowledged" : true, "matchedCount" : 4568, "modifiedCount" : 4568 }

```

```

  _id: ObjectId("60084d64140b994b9c5d9c0c")
  title: "The Beach"
  year: 2000
  cast: Array
    0: "Leonardo DiCaprio"
    1: "Virginie Ledoyen"
    2: "Tilda Swinton"
  genres: Array
    0: "Adventure"
  rating: 5
  averageRating: "{}"
  views: "[123444,66855,78966]"

```

‘views’:[123444, 66855,78966],

Faites une mise à jour des films que vous avez insérer en renseignant les valeurs pour le tableau views.

```
db.movies.updateMany({_id :ObjectId("600c9d107c78fb44d89e226e"),_id :ObjectId("600c9d107c78fb44d89e226f")},{ $push :{ views :[1000000,1000000,1000000]}})
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

Créer un champs totalViews qui sera la somme du tableau view

```
_id: ObjectId("600c9d107c78fb44d89e226f")
year: 2020
title: "IçERDE"
> cast: Array
> genres: Array
> views: Array
totalViews: 3000000
```

Créer un champs budget pour tous les documents et initialiser sa valeur à -1


```
> db.movies.update({},{$set:{"budget":-1}},false,true)
WriteResult({ "nMatched" : 4568, "nUpserted" : 0, "nModified" : 4568 })
>
```

Modifier les valeurs budget pour les films que vous avez ajouté.

```
> db.movies.update({_id:ObjectId("60084d64140b994b9c5d9c06")},{ $set:{"budget":1000}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.movies.update({_id:ObjectId("60084d64140b994b9c5d9c07")},{ $set:{"budget":1000}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
>
```

Faites un back up de votre base de données.

J'ai fait le BackUp via MongoDBCompass

| | | | |
|---|------------------|---------------------|----------|
|  movies.backUp | 21/01/2021 23:59 | Fichier source JSON | 1 647 Ko |
|  movies | 20/01/2021 16:18 | Fichier source JSON | 3 307 Ko |