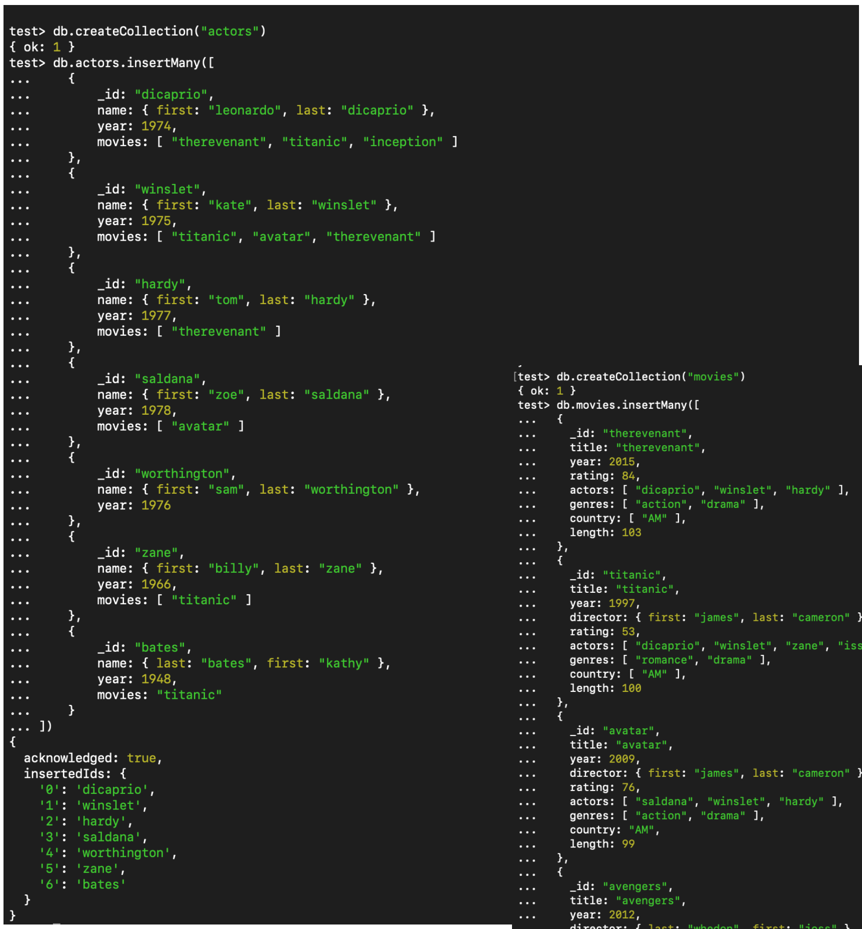
Université EUROMED de Fès
Ecole d'Ingénierie Digitale et d'Intelligence Artificielle
XML/NoSQL

Lab4
MongoDB

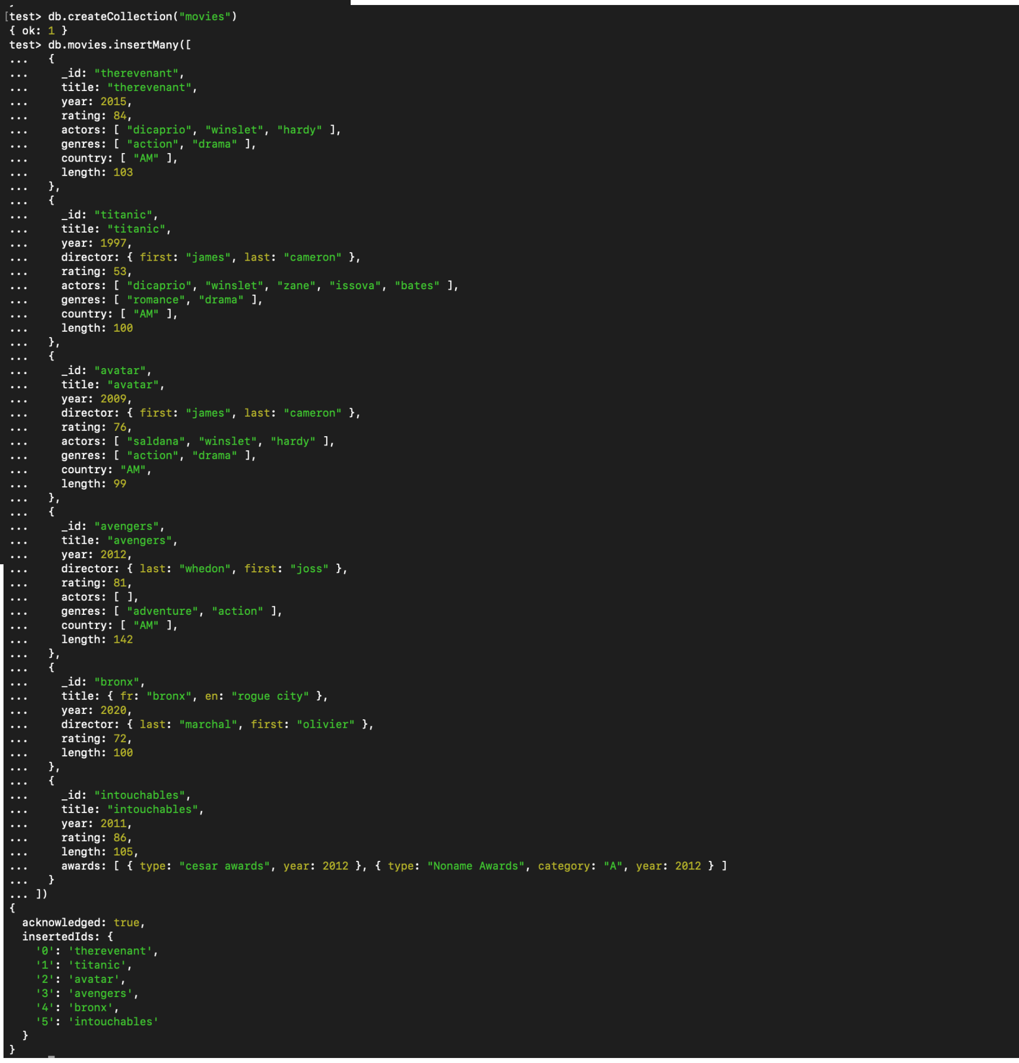
Create a database that respects the following directives:

Actors :

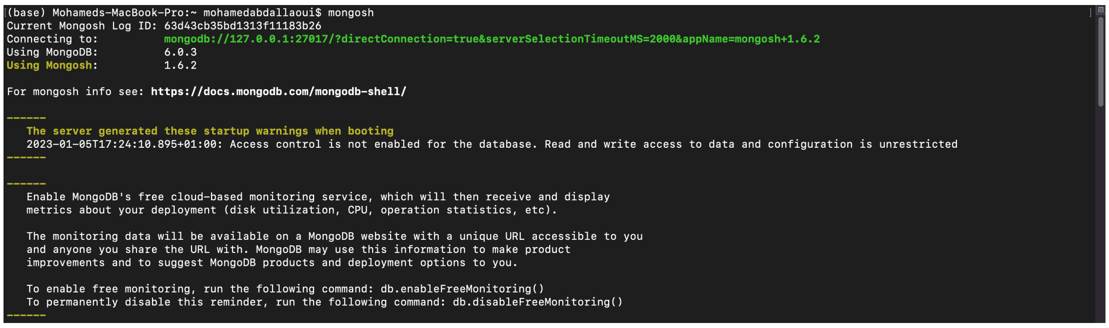


```
test> db.createCollection("actors")
{ ok: 1 }
test> db.actors.insertMany([
...     {
...         _id: "dicaprio",
...         name: { first: "leonardo", last: "dicaprio" },
...         year: 1974,
...         movies: [ "therevenant", "titanic", "inception" ]
...     },
...     {
...         _id: "winslet",
...         name: { first: "kate", last: "winslet" },
...         year: 1975,
...         movies: [ "titanic", "avatar", "therevenant" ]
...     },
...     {
...         _id: "hardy",
...         name: { first: "tom", last: "hardy" },
...         year: 1977,
...         movies: [ "therevenant" ]
...     },
...     {
...         _id: "saldana",
...         name: { first: "zoe", last: "saldana" },
...         year: 1978,
...         movies: [ "avatar" ]
...     },
...     {
...         _id: "worthington",
...         name: { first: "sam", last: "worthington" },
...         year: 1976
...     },
...     {
...         _id: "zane",
...         name: { first: "billy", last: "zane" },
...         year: 1966,
...         movies: [ "titanic" ]
...     },
...     {
...         _id: "bates",
...         name: { last: "bates", first: "kathy" },
...         year: 1948,
...         movies: "titanic"
...     }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': 'dicaprio',
    '1': 'winslet',
    '2': 'hardy',
    '3': 'saldana',
    '4': 'worthington',
    '5': 'zane',
    '6': 'bates'
  }
}
```

MOVIES :

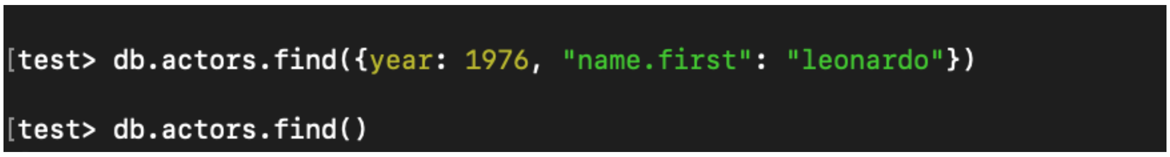


```
test> db.createCollection("movies")
{ ok: 1 }
test> db.movies.insertMany([
...   {
...     _id: "therevenant",
...     title: "therevenant",
...     year: 2015,
...     rating: 84,
...     actors: [ "dicaprio", "winslet", "hardy" ],
...     genres: [ "action", "drama" ],
...     country: [ "AM" ],
...     length: 103
...   },
...   {
...     _id: "titanic",
...     title: "titanic",
...     year: 1997,
...     director: { first: "james", last: "cameron" },
...     rating: 53,
...     actors: [ "dicaprio", "winslet", "zane", "issova", "bates" ],
...     genres: [ "romance", "drama" ],
...     country: [ "AM" ],
...     length: 100
...   },
...   {
...     _id: "avatar",
...     title: "avatar",
...     year: 2009,
...     director: { first: "james", last: "cameron" },
...     rating: 76,
...     actors: [ "saldana", "winslet", "hardy" ],
...     genres: [ "action", "drama" ],
...     country: "AM",
...     length: 99
...   },
...   {
...     _id: "avengers",
...     title: "avengers",
...     year: 2012,
...     director: { last: "whedon", first: "joss" },
...     rating: 81,
...     actors: [ ],
...     genres: [ "adventure", "action" ],
...     country: [ "AM" ],
...     length: 142
...   },
...   {
...     _id: "bronx",
...     title: { fr: "bronx", en: "rogue city" },
...     year: 2020,
...     director: { last: "marchal", first: "olivier" },
...     rating: 72,
...     length: 100
...   },
...   {
...     _id: "intouchables",
...     title: "intouchables",
...     year: 2011,
...     rating: 86,
...     length: 105,
...     awards: [ { type: "cesar awards", year: 2012 }, { type: "Noname Awards", category: "A", year: 2012 } ]
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': 'therevenant',
    '1': 'titanic',
    '2': 'avatar',
    '3': 'avengers',
    '4': 'bronx',
    '5': 'intouchables'
  }
}
```



```
(base) Mohameds-MacBook-Pro:~ mohamedabdallaoui$ mongosh
Current Mongosh Log ID: 63d43cb35bd1313f11183b26
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.2
Using MongoDB:          6.0.3
Using Mongosh:          1.6.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-01-05T17:24:10.895+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------
```

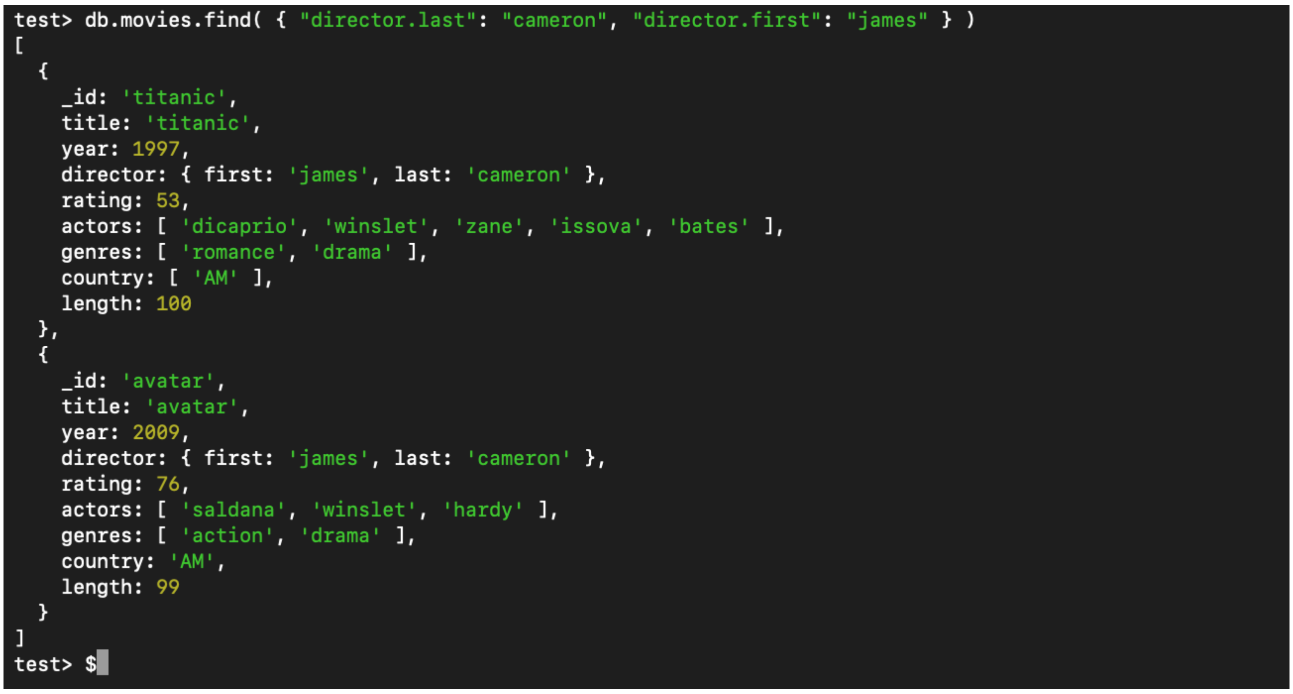
<u>START MONGO ON MAC TERMINAL</u>



```
[test> db.actors.find({year: 1976, "name.first": "leonardo"})

[test> db.actors.find()
```

<u>EXERCISE 1 ABOVE :</u>

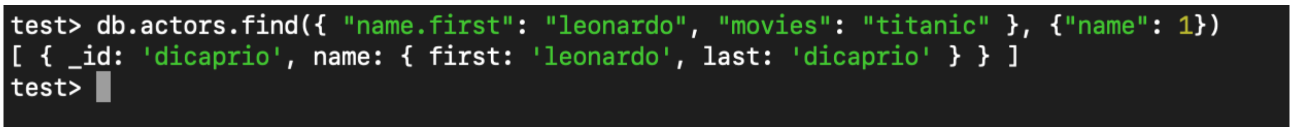
The actors born in 1976 with first name Leonardo doesn't exit since Dicaprio was born in 1974



```
test> db.movies.find( { "director.last": "cameron", "director.first": "james" } )
[
  {
    _id: 'titanic',
    title: 'titanic',
    year: 1997,
    director: { first: 'james', last: 'cameron' },
    rating: 53,
    actors: [ 'dicaprio', 'winslet', 'zane', 'issova', 'bates' ],
    genres: [ 'romance', 'drama' ],
    country: [ 'AM' ],
    length: 100
  },
  {
    _id: 'avatar',
    title: 'avatar',
    year: 2009,
    director: { first: 'james', last: 'cameron' },
    rating: 76,
    actors: [ 'saldana', 'winslet', 'hardy' ],
    genres: [ 'action', 'drama' ],
    country: 'AM',
    length: 99
  }
]
test> $
```

<u>EXERCISE 2 ABOVE :</u>

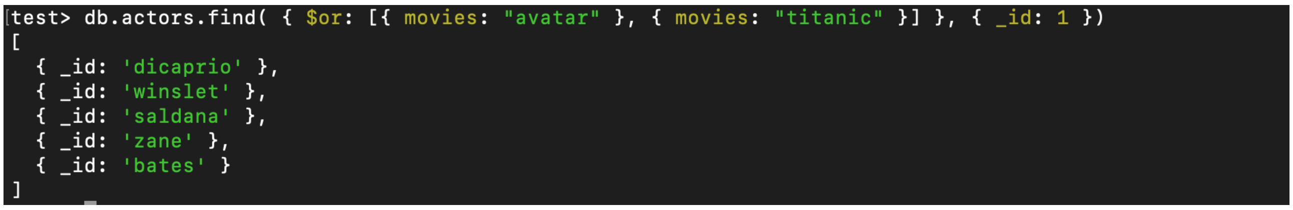
Starting with last name then first name



```
test> db.actors.find({ "name.first": "leonardo", "movies": "titanic" }, {"name": 1})
[ { _id: 'dicaprio', name: { first: 'leonardo', last: 'dicaprio' } } ]
test> 
```

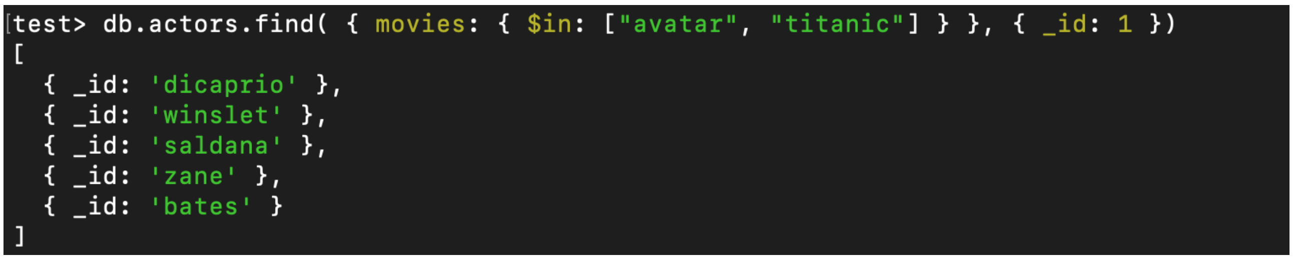
EXERCISE 3 ABOVE :

M1:



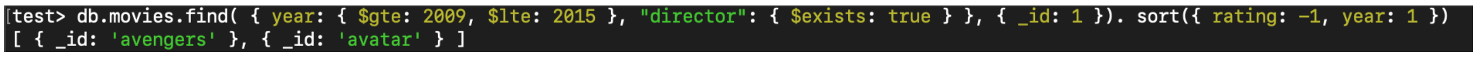
```
test> db.actors.find( { $or: [{ movies: "avatar" }, { movies: "titanic" }] }, { _id: 1 })
[
  { _id: 'dicaprio' },
  { _id: 'winslet' },
  { _id: 'saldana' },
  { _id: 'zane' },
  { _id: 'bates' }
]
```

M2 :



```
test> db.actors.find( { movies: { $in: ["avatar", "titanic"] } }, { _id: 1 })
[
  { _id: 'dicaprio' },
  { _id: 'winslet' },
  { _id: 'saldana' },
  { _id: 'zane' },
  { _id: 'bates' }
]
```

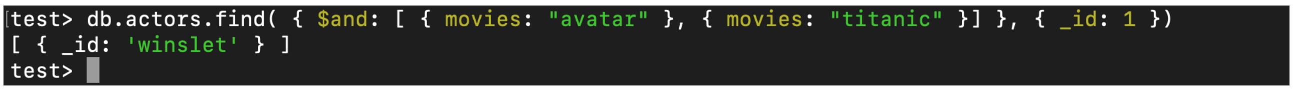
EXERCISE 5 ABOVE :



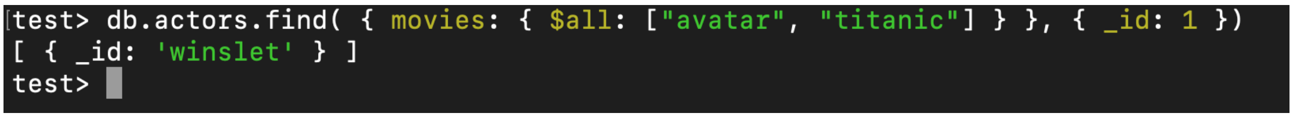
```
test> db.movies.find( { year: { $gte: 2009, $lte: 2015 }, "director": { $exists: true } }, { _id: 1 }). sort({ rating: -1, year: 1 })
[ { _id: 'avengers' }, { _id: 'avatar' } ]
```

Exercise 4 above : This query will find movies that have been shot between years 2009 and 2015 and have a director specified, return only the "_id" field of the movies, and sort the results by descending rating and ascending year.

EXERCISE 6 WITH 2 METHODS BELOW :

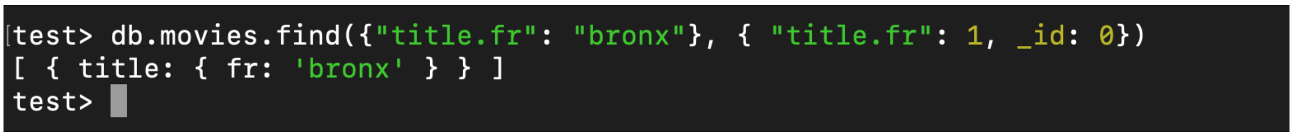


```
[test> db.actors.find( { $and: [ { movies: "avatar" }, { movies: "titanic" }] }, { _id: 1 })
[ { _id: 'winslet' } ]
test>
```



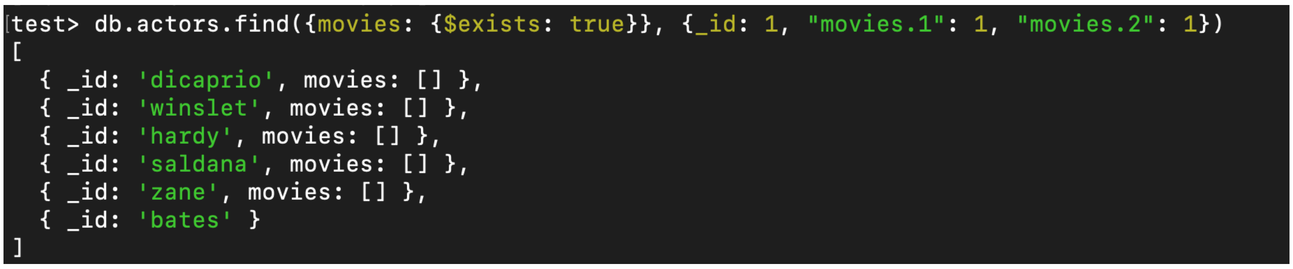
```
[test> db.actors.find( { movies: { $all: ["avatar", "titanic"] } }, { _id: 1 })
[ { _id: 'winslet' } ]
test>
```

EXERCISE 7 BELOW:



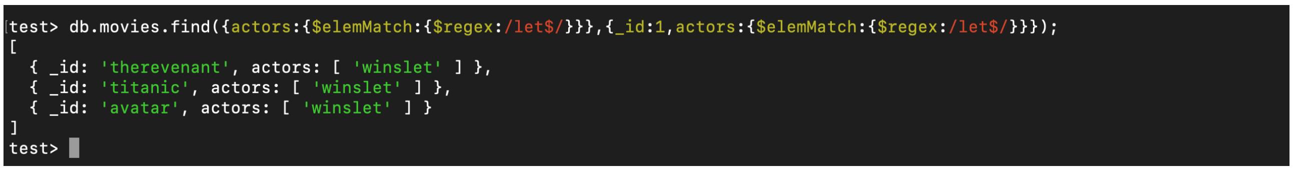
```
[test> db.movies.find({"title.fr": "bronx"}, { "title.fr": 1, _id: 0})
[ { title: { fr: 'bronx' } } ]
test>
```

EXERCISE 8 BELOW :



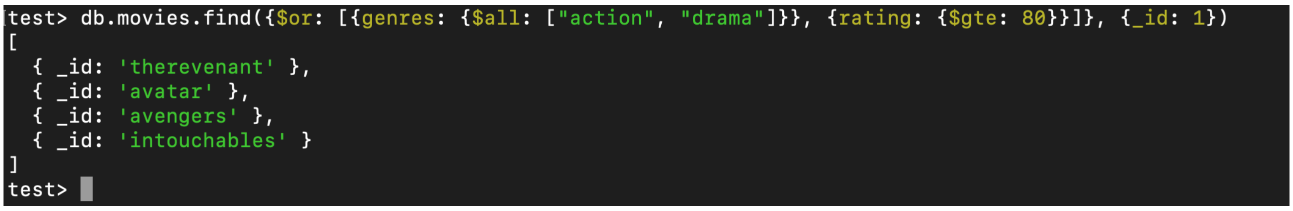
```
[test> db.actors.find({movies: {$exists: true}}, {_id: 1, "movies.1": 1, "movies.2": 1})
[
  { _id: 'dicaprio', movies: [] },
  { _id: 'winslet', movies: [] },
  { _id: 'hardy', movies: [] },
  { _id: 'saldana', movies: [] },
  { _id: 'zane', movies: [] },
  { _id: 'bates' }
]
```

EXERCISE 9 BELOW :



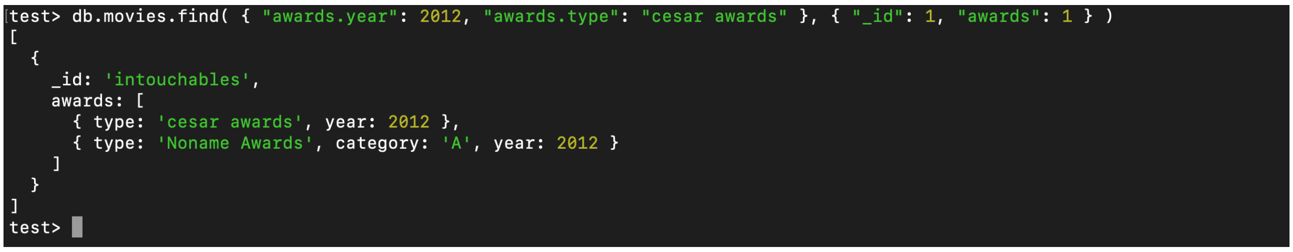
```
[test> db.movies.find({actors:{$elemMatch:{$regex:/let$/}}},{_id:1,actors:{$elemMatch:{$regex:/let$/}}});
[
  { _id: 'therevenant', actors: [ 'winslet' ] },
  { _id: 'titanic', actors: [ 'winslet' ] },
  { _id: 'avatar', actors: [ 'winslet' ] }
]
test>
```

EXERCISE 11 BELOW:



```
test> db.movies.find({$or: [{genres: {$all: ["action", "drama"]}}, {rating: {$gte: 80}}]}, {_id: 1})
[
  { _id: 'therevenant' },
  { _id: 'avatar' },
  { _id: 'avengers' },
  { _id: 'intouchables' }
]
test> 
```

EXERCISE 10 BELOW :



```
test> db.movies.find( { "awards.year": 2012, "awards.type": "cesar awards" }, { "_id": 1, "awards": 1 } )
[
  {
    _id: 'intouchables',
    awards: [
      { type: 'cesar awards', year: 2012 },
      { type: 'Noname Awards', category: 'A', year: 2012 }
    ]
  }
]
test> 
```

MADE BY : ABDALLAOUI MOHAMED.