**Gestion Des données - MongoDB**

# Exercice 3

Créez une base de données qui porte le nom de votre groupe « exemple (DEV203)

Importez le fichier movies.json dans une collection qui porte le même nom.

# Les Requêtes

Réalisez les requêtes suivantes et coller le code correspondant en dessous de la question

Réalisez les requêtes suivantes et coller le code correspondant en dessous de la question

1. Afficher les titres de tous les films triès par nom croissant

db.movies.find({},{"\_id":0,"title":1}).sort({"title":1});

db.movies.distinct("title")

2. Afficher les noms des acteurs qui ont joué dans le film « avatar »

db.movies.find({"title":"Avatar"},{"\_id":0,"cast":1})

3. La liste des noms des différents acteurs qui ont joué durant l’année 2011

db.movies.distinct("cast",{"year":2011})

4. La liste des noms des différents acteurs qui ont joué durant la période 2011/2015

db.movies.distinct("cast",{"year":{$gte:2011,$lte:2015} })

db.movies.distinct("cast",{$and : [{"year":{$gte:2011}},{"year":{$lte:2015}}] })

4-B : La liste des noms des différents acteurs qui ont joué soit en 2011 ou en 2015

db.movies.distinct("cast",{$or : [{"year":2011},{"year":2015}] })

5. Les titres des films qui ont un rating imdb supérieur à 7

db.movies.find({"imdb.rating":{$gt:7}},{title:1, \_id:0})

6. Les pays des films qui ont comme director « Steven Spielberg »

db.movies.distinct("countries", {"directors":"Steven Spielberg"})

7. Les liste des différents emails qui se trouvent dans les commentaires des films

db.movies.distinct("comments.email")

8. Afficher les titres des films par Writer dans l’ordre croissant sur les noms des writers et des films.

varUnwind={$unwind:"$writers"}

varGroup={$group:{\_id:{writers:"$writers", "title":"$title"}}}

varSort={$sort:{"\_id":1}}

db.movies.aggregate([varUnwind, varGroup, varSort]);

9. Le nombre des différents genres des films

db.movies.distinct("genres").length

10. Les directeurs qui ont participé comme acteur dans leurs films

--reponse incorrect

db.movies.find({$expr:{$eq:[$cast, $directors]}},{title:1, cast:1, directors:1, \_id:0})

--reponse correct

varUnwind={$unwind:"$cast"}

varUnwind2={$unwind:"$directors"}

varMatch = {$match:{$expr:{$eq:["$cast", "$directors"]}}}

varGroup={$group:{\_id:{"title":"$title","cast":"$cast","director":"$directors"}}}

db.movies.aggregate([varUnwind,varUnwind2, varMatch , varGroup]);

11. Le nombre des films qui ont le genre « sci-fi »

db.movies.find({"genres":/sci-fi/i}).count()

12. Le nombre des films par genre

varUnwind = {$unwind : "$genres"}

varGroup = {$group : {"\_id": "$genres", "nb" : {$sum : 1}}}

db.movies.aggregate([varUnwind, varGroup])

13. La liste des films qui ont une durée entre l’année de publication

et l’année de mise a jour est supérieur à 4 ans.

varAddfield = {$addFields : {

"differe":{$subtract:[{$toInt:{$substr:["$lastupdated",0,4]}},{$toInt:{$substr:["$year",0,4]}}]}

}}

varMatch = {$match:{differe:{"$gt":4}}}

varProject = {$project:{

title:"$title",

"differe":"$differe"

}}

varSort = {$sort:{differe:1}}

db.movies.aggregate([varAddfield, varMatch, varProject, varSort])

14. La durée maximale des films « short » qui commencent par le mot « The »

db.movies.find({"genres":/Short/i, "title":{"$regex":"^The"}}, {runtime:1, \_id:0})

.sort({runtime:-1})

.limit(1)

15. Les films qui ont réalisé le plus grand nombre de votes.

varMatch={$match:{"imdb.votes":{$ne:""}}}

varGroup={$group:{"\_id":null,"maxvote":{"$max":"$imdb.votes"}}}

varProject={$project:{max:"$maxvote"}}

db.movies.aggregate([varMatch, varGroup,varProject])

varMax = db.movies.aggregate([varMatch, varGroup,varProject])

.toArray()

.map(function(e){return e.max})[0]

db.movies.find({"imdb.votes":{$eq:varMax}},{"title":1,"\_id":0})

1-6. Les acteurs qui ont le plus grand rating imdb

varMaxRating = db.movies.find({"imdb.rating":{$ne:""}}, {"imdb.rating":1, \_id:0})

.sort({"imdb.rating":-1})

.limit(1).toArray().map(function(e){return e.imdb.rating})[0]

db.movies.find({"imdb.rating":{$eq:varMaxRating}},{"cast":1,"\_id":0})

17. Le meilleur film par année « utilisez la moyenne des rating imdb »

varMatch = {$match:{"imdb.rating":{$ne:""}}}

varGroup = {$group : {"\_id":{$toInt:{$substr:["$year",0,4]}}, max:{$max:"$imdb.rating"}}}

db.movies.aggregate([varMatch, varGroup]).toArray()

.map(function(e){

return db.movies.find({"imdb.rating":{$eq:e.max}, "year":{$eq:e.\_id}}

,{year:1, title:1, \_id:0, "imdb.rating":1}).toArray()

.map(function(i){return i.title + " - " + i.year + " - " + i.imdb.rating})

});

18. Le meilleure acteur de l’année 2011 « utilisez la moyenne des rating imdb »

varMaxRating = db.movies.find({"imdb.rating":{$ne:""}, "year" : 2011}, {"imdb.rating":1, \_id:0})

.sort({"imdb.rating":-1})

.limit(1).toArray().map(function(e){return e.imdb.rating})[0]

varUnwind = {$unwind : "$cast"}

varMatch = {$match : {"imdb.rating" : varMaxRating}}

varGroup = {$group : {"\_id" : "$cast", "rating" : {$max : "$imdb.rating"}}}

db.movies.aggregate([varUnwind, varMatch, varGroup])

19. Le(s) pays qui ont le plus grand rating imdb

varMaxRating = db.movies.find({"imdb.rating":{$ne:""}}, {"imdb.rating":1, \_id:0})

.sort({"imdb.rating":-1})

.limit(1).toArray().map(function(e){return e.imdb.rating})[0]

varUnwind = {$unwind : "$countries"}

varMatch = {$match : {"imdb.rating" : varMaxRating}}

varGroup = {$group : {"\_id" : "$countries", "rating" : {$max : "$imdb.rating"}}}

db.movies.aggregate([varUnwind, varMatch, varGroup])

20. Les films qui ont un rating imdb supérieur

ou égale à la moyenne des rating de son premier genre.