**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

Cette requette permet d’afficher un nombre aléatoire inférieur à 20

db.movies.aggregate( [

{$group:{"\_id":null}},

{$set:{"a":{$floor:{$multiply:[{$rand:{}},20]}}}},

{$project:{a:1}}

])

1. Afficher tous les films

db.movies.find()

2. Afficher les films qui ont un imdb rating supérieur à 7

db.movies.find({"imdb.rating":{"$gt":7}})

3. Afficher les films qui durent plus que 2 heures

db.movies.find({"type":"movie","runtime":{"$gt":120}})

4. le nombre des films par catégorie de « will smith » avant 2000 triès par catégorie

db.movies.aggregate([

{$unwind :"$genres"},

{$match :{cast :"Will Smith" ,year:{$lt :2000}}},

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

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

])

5. Afficher la liste des différents acteurs triès par nom

db.movies.distinct("cast");

6. Afficher la liste des pays des différents films

db.movies.distinct("countries",{"type":"movie"});

7. La liste des noms des films du director « Steven Spielberg»

db.movies.find({"directors":"Steven Spielberg"},{"\_id":0,"title":1})

8. La liste des noms des films qui contiennent le mot « avatar »

db.movies.find({"title":{"$regex":/avatar/i}},{"\_id":0,"title":1})

db.movies.find({"title":/avatar/i},{"\_id":0,"title":1})

9. le nombre des films qui ont un rating imdb supérieur ou égale à 7

db.movies.find({"imdb.rating": {$gte: 7}}).count()

db.movies.count({"imdb.rating": {$gte: 7}})

db.movies.countDocuments({"imdb.rating": {$gte: 7}})

10. La liste des films qui n’ont pas de rating

db.movies.insertOne({"\_id":"007","title":"test"})

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

11. la moyenne des votes imdb par genre par pays

db.movies.aggregate([

{$unwind:"$genres"},

{$unwind:"$countries"},

{$group:{\_id:{"genre":"$genres","pays":"$countries"},"moyvotes":{$avg:"$imdb.votes"}}},

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

])

12. le nombre des films par acteurs par ordre décroissant sur le nombre

db.movies.aggregate([

{$unwind:"$cast"},

{$group:{\_id:"$cast","NombreFilm":{$sum:1}}},

{$sort:{"NombreFilm":-1}}

])

13. les films dont le titre commence par « the » « utiliser « $regex »

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

14. le rating imdb moyenne des films de « Steven Spielberg»

varMatch = {$match : {"directors":"Steven Spielberg"}}

varGroup = {$group : {"\_id":null , "imdb\_moyen\_rating": {$avg :"$imdb.rating"}}}

varProject = {$project : { "imdb\_moyen\_rating": {$round:["$imdb\_moyen\_rating",2]}}}

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

15. le nombre des ecrivains par film de « USA »

db.movies.aggregate([

{$unwind:"$writers"},

{$match:{"countries":"USA"}},

{$group:{\_id:"$writers","NombreEcr":{$sum:1}}}

])

16. Les acteurs qui ont le plus grand rating imdb

Methode 1

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

.toArray().map(function(item){return item.imdb.rating})[0]

varUnwind = {$unwind : "$cast"}

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

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

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

method2

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

.toArray().map(function(item){return item.imdb.rating})[0]

db.movies.distinct("cast",{"imdb.rating":maxRating})

17. Le meilleur film par année

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

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

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

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

.toArray().map(function(e){

return db.movies.aggregate([

{$match:{$and: [{"year":e.\_id}, {"imdb.rating":e.maxImdb}]}},

{$group:{\_id:{"year":"$year","title":"$title", "rating":"$imdb.rating"}}},

{$project:{"\_id":1}}

]).toArray().map(function(i){return i.\_id.year+ " -- "+ i.\_id.title+ " -- "+ i.\_id.rating});

});

18. La moyenne des durées des films par genre

db.movies.aggregate([

{$unwind:"$genres"},

{$group:{"\_id":"$genres","moyduree":{$avg:"$runtime"}}}])

19. Donnez les cinq meilleurs directeurs de l’année 2000

varUnwind = {$unwind: "$directors"}

varMatch = {$match : { "year" :2000 , "imdb.rating" : {$ne :""}}}

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

varSort = {$sort : {"rating" : -1}}

varLimit = {$limit:5}

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

20. L’année dans laquelle on a produit le plus grand nombre de films

varGroup = {$group:{"\_id":{$substr:[{$convert:{input:"$year",to:"string"}},0,4]},"nb":{$sum:1}}},  
varSort = {$sort:{"nb":-1}},  
varLimit = {$limit:1}  
varMatch={$match:{"nb":db.movies.aggregate([varGroup,varSort, varLimit])  
 .toArray().map(function(e){return e.nb})[0]}}  
db.movies.aggregate([varGroup,varMatch])