**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

1. Afficher tous les films

db.movies.distinct("title")

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

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

1. Afficher les films qui durent plus que 2 heures

db.movies.find({"runtime" : {$gt :2}})

1. le nombre des films de « will smith » avant 1990 triès par catégorie

db.movies.find({"cast" : " will smith ", "year" : {$lt : 1990}}).sort({"genres" : 1})

1. Afficher la liste des différents acteurs triès par nom

db.movies.aggregate([{$project : {" **cast**" : 1 ,"\_id":0 }} , {$sort : {"cast" : 1}}])

1. Afficher la liste des pays des différents films

db.movies.aggregate([{$project : {"title" : 1 ,"countries" : 1 , "\_id":0 }}])

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

db.movies.aggregate([{$project : {"directors" : 1 ,"title" : 1,"\_id":0 }} , {$match : {"directors" : "Steven Spielberg"}}])

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

db.movies.aggregate([{$project : {"\_id" : 0 , title : 1}} , {$match : {"title" : {$regex : /avatar/i} }}])

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

db.movies.aggregate([{$project : {"title" : 1 , "imdb" : 1 , "\_id" : 0}},{$match : {"imdb.rating" : {"$gte" : 7}}} , {$count : "title"}])

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

db.movies.aggregate([{$project : {"title" : 1 ,"\_id" : 0}},{$match : {"imdb" : {"$exists" : false} , "tomatoes" : {"$exists" : false}}}])

1. la moyenne des votes imdb par genre par pays

db.movies.aggregate([{ $unwind: "$genres"},{$unwind : "$countries"}, {$group: {\_id: ["$genres" ,"$countries"], avgRating: {$avg: "$imdb.votes"}}}])

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

db.movies.aggregate([{ $unwind: "$cast"}, {$group: {\_id:"$cast", nbMovie: {$sum: 1}}}])

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

db.movies.find({"title": {"$regex" : /^The/}})

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

db.movies.aggregate([{ $unwind: "$directors"},{ $match: {"directors": {$regex: "Steven Spielberg"}}}, {$group: {\_id: "$directors", avgMovie: {$avg: "$imdb.rating"}}}])

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

db.movies.aggregate([{$unwind: "$writers"},{ $match: {"countries": {$regex: "USA"}},{$group: {\_id: "$title", total: {$sum: 1}}}])

1. Les acteurs qui ont le plus grand rating imdb

db.movies.aggregate([{$unwind : "$cast"},{$match : {"imdb.rating" : {"$gte" :0}}} , {$group : {"\_id" : "$cast" , "maxImdb" : {"$max" : "$imdb.rating"}}} ,{$sort : {"maxImdb" :-1}},{$limit:1}])

1. Le meilleur film par année

varUnwind = {$unwind:"$year"}

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

varGroup = {$group : {"\_id":{"year" : "$year"} , "Movies" : {"$push":{"title" : "$title" , "imdb\_rating" : "$imdb.rating"} }}}

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

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

1. La moyenne des durées des films par genre

varUnwind = {$unwind:"$genres"}

varGroup = {$group : {"\_id":"$genres" , "The\_average\_duration": {$avg : "$runtime"} }}

varProject = {$project : {"\_id":1 ,"The\_average\_duration" : {$round : ["$The\_average\_duration",2]}}}

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

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

varUnwind = {$unwind :"$directors"}

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

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

varSort = {$sort : {"imdb\_rating" : -1}}

varLimit = {$limit : 5}

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

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

varUnwind = {$unwind :"$year"}

varGroup = {$group : {"\_id":"$year" , "nbMovies" : {$sum : 1}}}

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

varLimit = {$limit : 1}

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