

**Abstract**

*Querying a pre-existing movie database through MongoDB and retrieving answers for questions given to us.*

Connor George Scrimgeour, Michael Lemetti & Brandon Sinclair.

cscu9yq – no sql databases

Assignment – Team 19

# Introduction

For this assignment we have been given a movies dataset in JSON format via the JavaScript file ‘MovieData.js’. Using MongoDB, we need to create and query this dataset according to a given brief of questions. Our job is to query the dataset to come up with answers to these questions.

This movie dataset contains information on a myriad of films created between the early 1900s and 2018. We are given the year of its release, its genre, countries associated with it, the director, actors, any awards or nominations its received, a brief synopsis and ratings from three different movie reviewing sites. These review sites are IMDB, Metacritic and Rotten Tomatoes. These sites use a review-aggregation system that takes both critic and average user reviews into two separate aggregate scores.

This assignment has been broken down into three questions, each with two objectives, that we will outline in this report. We will give our strategies on tackling the questions, what fields we are using when querying the dataset, the query and the results.

# Question 1 – General Queries

This questions given objectives are:

1. To compute the average duration of movies by genre and to list the top five genres in descending order depending on their average runtime duration.
2. List all the countries that have produced ten or more films in the UK by their name and number of films produced in collaboration with the UK.

To achieve the first objective, we used the aggregation operator ‘$unwind’. What unwind does is it deconstructs an array field and allows us to output a document for each element in the array. We unwind the genres information and then use the aggregation operator ‘$group’ to allow us to specify specific fields we want to use. Finally, we use the ‘$avg’ operator to get the average runtime for our specific group of films by genre.

## Query 1:

## *db.movies.aggregate([{$unwind:"$genres"},{$group:{\_id:"$genres",watch:{$avg:"$runtime"}}},{$sort:{watch:-1}},{$limit: 5}])*

## Result 1:

*{ "\_id" : "Musical", "watch" : 113.375 }*

*{ "\_id" : "Romance", "watch" : 107.18571428571428 }*

*{ "\_id" : "Crime", "watch" : 106.56571428571428 }*

*{ "\_id" : "Action", "watch" : 105.208 }*

*{ "\_id" : "Western", "watch" : 105.12121212121212 }*

For the second objective we start off by using the aggregate operator ‘$match’ on countries to specify that we want entries where the country associated with it is the UK. Then we use ‘$unwind’ on the countries information and use ‘$match’ and ‘$ne’ operators to find countries that are not equal to the UK. We then group the information and count it using the ‘$sum’ operator. After that we use ‘$match’, count and the ‘$gte’ by ten to make sure we return all countries that have produced at least ten or more films in the UK.

## Query 2:

## *db.movies.aggregate([{$match: {countries: "UK"}},{$unwind: "$countries"},{$match:{countries:{$ne:"UK"}}},{$group: {\_id: "$countries",count: {$sum: 1}}}, {$match: {count: {$gte: 10}}}])*

## Result 2:

*{ "\_id" : "Canada", "count" : 11 }*

*{ "\_id" : "USA", "count" : 62 }*

*{ "\_id" : "Germany", "count" : 14 }*

*{ "\_id" : "France", "count" : 16 }*

# Question 2 – Films Related to Sports

This questions given objectives are:

1. Report the number of films in the dataset that have any relation to sports whatsoever.
2. From the sport related films that we have found, list the top three that have received the highest IMDB rating by movie title, year, their rating and votes.

The first objective is relatively simple after question 1. We just unwind genres, match with the keyword ‘Sport’ and count how many ‘sportMovies’ there are.

## Query 3:

## *db.movies.aggregate([{$unwind: "$genres"}, {$match: {"genres":"Sport"}}, {$count: "sportMovies"}])*

## Result 3:

*{ "sportMovies" : 28 }*

Then for the second query we need to find the top there and list them by the movie, title and year. We do that by using using the unwind, match and sort operators and limiting our projected output to 3 based on IMDB votes.

## Query 4:

*db.movies.aggregate([{$unwind:"$genres"},{$match: {"genres":"Sport"}},{$sort:{"imdb.rating":-1}}, {$limit:3}, {$project:{\_id:0,title:1, year:1,"imdb.rating":1, "imdb.votes":1}}])*

## Result 4

*{ "title" : "Liverpool FC: Champions of Europe 2005", "year" : 2005, "imdb" : { "rating" : 9.5, "votes" : 410 } }*

*{ "title" : "WWE: DX: One Last Stand", "year" : 2011, "imdb" : { "rating" : 7.7, "votes" : 51 } }*

*{ "title" : "O sport, ty - mir!", "year" : 1981, "imdb" : { "rating" : 7.5, "votes" : 23 } }*

# Question 3 – Rating and Recommending Films

Finally, this questions given objectives are:

1. We need to create our own filed called ‘myRating’ and to create our own criteria for a single digit rating. We are allowed to tackle this our own way but it will in some form be an aggregation of the previously given ratings from review sites and our own personal tastes.
2. Using our newly created ratings system we should recommend three films we have found and give reasons as to why.

For the first part of the final question we need to create our own rating system with our own criteria. The criteria we have chosen are using the three ratings we are given but because of the increased scale that Metacritic operates by we have divided that by 10, so as to leave all ratings between 0 and 10. We then change the collection to finalMovies tolet us know the query worked.

## Query 5:

*db.movies.aggregate([{$addFields: {myRating:{$avg: ["$imdb.rating", "$tomato.rating", {$divide: ["$metacritic",10]}]}}}, {$out:"finalMovies"}])*

## Result 5:

*Collection created called finalMovies and new rating created called myRating is now useable.*

And lastly we will recommend three movies from our collection by our criteria. We have also decided to filter it via action movies because they are our favourite genre. We use the same operators as discussed earlier but also use “pretty” as this allows us to display the results in a more readable manner.

## Query 6:

*db.finalMovies.aggregate([{$unwind:"$genres"},{$match: {"genres":"Action"}},{$sort:{"myRating":-1}}, {$limit:3}]).pretty()*

## Result 6:

*{*

*"\_id" : ObjectId("603909c74c6ef4c768a05d26"),*

*"title" : "Raiders of the Lost Ark",*

*"year" : 1981,*

*"rated" : "PG",*

*"runtime" : 115,*

*"countries" : [*

*"USA"*

*],*

*"genres" : "Action",*

*"director" : "Steven Spielberg",*

*"writers" : [*

*"Lawrence Kasdan",*

*"George Lucas",*

*"Philip Kaufman"*

*],*

*"actors" : [*

*"Harrison Ford",*

*"Karen Allen",*

*"Paul Freeman",*

*"Ronald Lacey"*

*],*

*"plot" : "Archaeologist and adventurer Indiana Jones is hired by the US government to find the Ark of the Covenant before the Nazis.",*

*"poster" : "http://ia.media-imdb.com/images/M/MV5BMjA0ODEzMTc1Nl5BMl5BanBnXkFtZTcwODM2MjAxNA@@.\_V1\_SX300.jpg",*

*"imdb" : {*

*"id" : "tt0082971",*

*"rating" : 8.5,*

*"votes" : 616230*

*},*

*"tomato" : {*

*"meter" : 96,*

*"image" : "certified",*

*"rating" : 9.2,*

*"reviews" : 67,*

*"fresh" : 64,*

*"consensus" : "Featuring bravura set pieces, sly humor, and white-knuckle action, Raiders of the Lost Ark is one of the most consummately entertaining adventure pictures of all time.",*

*"userMeter" : 96,*

*"userRating" : 4.1,*

*"userReviews" : 822379*

*},*

*"metacritic" : 90,*

*"awards" : {*

*"wins" : 28,*

*"nominations" : 23,*

*"text" : "Won 4 Oscars. Another 28 wins & 23 nominations."*

*},*

*"type" : "movie",*

*"myRating" : 8.9*

*}*

*{*

*"\_id" : ObjectId("603909c74c6ef4c768a05c78"),*

*"title" : "The Adventures of Robin Hood",*

*"year" : 1938,*

*"rated" : "PG",*

*"runtime" : 102,*

*"countries" : [*

*"USA"*

*],*

*"genres" : "Action",*

*"director" : "Michael Curtiz, William Keighley",*

*"writers" : [*

*"Norman Reilly Raine",*

*"Seton I. Miller"*

*],*

*"actors" : [*

*"Errol Flynn",*

*"Olivia de Havilland",*

*"Basil Rathbone",*

*"Claude Rains"*

*],*

*"plot" : "When Prince John and the Norman Lords begin oppressing the Saxon masses in King Richard's absence, a Saxon lord fights back as the outlaw leader of a rebel guerrilla army.",*

*"poster" : "http://ia.media-imdb.com/images/M/MV5BMTUxMzE0MjI0MF5BMl5BanBnXkFtZTcwMTQ1OTM2MQ@@.\_V1\_SX300.jpg",*

*"imdb" : {*

*"id" : "tt0029843",*

*"rating" : 8,*

*"votes" : 37543*

*},*

*"tomato" : {*

*"meter" : 100,*

*"image" : "certified",*

*"rating" : 8.9,*

*"reviews" : 44,*

*"fresh" : 44,*

*"consensus" : "Errol Flynn thrills as the legendary title character, and the film embodies the type of imaginative family adventure tailor-made for the silver screen.",*

*"userMeter" : 89,*

*"userRating" : 3.7,*

*"userReviews" : 33282*

*},*

*"metacritic" : 97,*

*"awards" : {*

*"wins" : 2,*

*"nominations" : 2,*

*"text" : "Won 3 Oscars. Another 2 wins & 2 nominations."*

*},*

*"type" : "movie",*

*"myRating" : 8.866666666666667*

*}*

*{*

*"\_id" : ObjectId("603909c74c6ef4c768a05c58"),*

*"title" : "Star Wars: Episode IV - A New Hope",*

*"year" : 1977,*

*"rated" : "PG",*

*"runtime" : 121,*

*"countries" : [*

*"USA"*

*],*

*"genres" : "Action",*

*"director" : "George Lucas",*

*"writers" : [*

*"George Lucas"*

*],*

*"actors" : [*

*"Mark Hamill",*

*"Harrison Ford",*

*"Carrie Fisher",*

*"Peter Cushing"*

*],*

*"plot" : "Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a wookiee and two droids to save the universe from the Empire's world-destroying battle-station, while also attempting to rescue Princess Leia from the evil Darth Vader.",*

*"poster" : "http://ia.media-imdb.com/images/M/MV5BMTU4NTczODkwM15BMl5BanBnXkFtZTcwMzEyMTIyMw@@.\_V1\_SX300.jpg",*

*"imdb" : {*

*"id" : "tt0076759",*

*"rating" : 8.7,*

*"votes" : 822849*

*},*

*"tomato" : {*

*"meter" : 94,*

*"image" : "certified",*

*"rating" : 8.5,*

*"reviews" : 99,*

*"fresh" : 93,*

*"consensus" : "A legendarily expansive and ambitious start to the sci-fi saga, George Lucas opened our eyes to the possibilities of blockbuster filmmaking and things have never been the same.",*

*"userMeter" : 96,*

*"userRating" : 4.1,*

*"userReviews" : 851120*

*},*

*"metacritic" : 92,*

*"awards" : {*

*"wins" : 38,*

*"nominations" : 27,*

*"text" : "Won 6 Oscars. Another 38 wins & 27 nominations."*

*},*

*"type" : "movie",*

*"myRating" : 8.799999999999999*

*}*