****

***NOSQL Databases and Data Warehousing***

**Code: CSCUMB4**

*Assignment 2023*

Submitted by

**Stirling No: 2840080**

Submitted on

**(29/3/2023)**

1. **Table of Contents**

Introduction………………………………………………………………………………………………………………………….3

Questions 1…………………………………………………………………………………………………………………………..4

Query……………………………………………………………………………………………………………4

Explanations…………………………………………………………………………………………………4

Result…………………………………………………………………………………………………………..5

Questions 2…………………………………………………………………………………………………………………………..6

Query……………………………………………………………………………………………………………6

Explanations…………………………………………………………………………………………………6

Result…………………………………………………………………………………………………………..7

Questions 3………………………………………………………………………………………………………………………….8

Query……………………………………………………………………………………………………………8

Explanations…………………………………………………………………………………………………8

Result…………………………………………………………………………………………………………..9

Questions 4…………………………………………………………………………………………………………………………10

Query…………………………………………………………………………………………………………10

Explanations………………………………………………………………………………………………10

Result…………………………………………………………………………………………………………11

Questions 5……………………………………………………………………………………………………………………….13

Query…………………………………………………………………………………………………………13

Explanations………………………………………………………………………………………………15

Result…………………………………………………………………………………………………………15

1. **Introduction**

In this task, I got a dataset from some movies in JSON format by a JavaScript file 'movie\_data-1.js' using Mongodb, I need to create the dataset in addition to some queries about it according to the questions asked. My job is to query the dataset to find the answer to the questions attached in this the assignment. The data set contains information about films in a very large number, as they were created in the 1900s until 2018. This data also contains the type of films and their year of release, director, actors, countries, and awards received by these films, in addition to ratings from different sites. This task was divided into 5 questions, each with several objectives that will be identified in this report. I will present my solution on addressing these questions and the fields that I will use when querying.

1. **Question 1**

List all the actors who have worked with Natalie Portman in more than one movie. Indicate also in how many movies, and the years they have worked with her.

**Query:**

db.movies.aggregate([

{$match:{actors:"Natalie Portman"}},

{$unwind:"$actors"},

{$group:{\_id:"$actors",count:{$sum:1},year:{$push:"$year"}}},

{$match:{"count":{"$gt":1}}},

{$group:{\_id:0,actors:{$push:{Actor\_Name:"$\_id",Count\_of\_Movies:"$count",Years\_of\_work: "$year"}}}}

]).pretty()

**Explanation:**

* First, we match only movies with actor actors: "Natalie Portman" using Match.
* break down the movie's record into single records for each actor using $unwind.
* group only actors name and years and count how many the actor worked with Natalie Portman.
* using match, I filtered the result to show only who worked more than once with Natalie Portman
* Finally, we show the records that include the actor’s name and movie count and years.

**Results:**

{

"\_id" : 0,

"actors" : [{

"Actor\_Name" : "Hayden Christensen",

"Count\_of\_Movies" : " $count",

"Years\_of\_work" : [

2005,

2002

]},

{

"Actor\_Name" : "Natalie Portman",

"Count\_of\_Movies" : " $count",

"Years\_of\_work" : [

1999,

2005,

2002,

2009,

2000,

2008,

2013,

2011

]},

{

"Actor\_Name" : "Ewan McGregor",

"Count\_of\_Movies" : " $count",

"Years\_of\_work" : [

1999,

2005,

2002

]}

]}

1. **Question 2**

List the four actors who worked in the largest number of movie genres. Show both the genres they have worked on and the number of genres.

**Query:**

db.movies.aggregate([

{"$unwind":"$actors"},

{"$group":{"\_id":{"movieType":"$genres","actorName":"$actors"}}},

{"$group":{"\_id":"$\_id.actorName","movie":{"$push":{"genres":"$\_id.movieType","total":{"$sum":{"$size":"$\_id.movieType"}}}}}},

{$sort:{"movie.total":-1}},

{$limit:4}

]).pretty()

Explanation:

* {"$unwind":"$actors"} breaks down Actors Field to single records for each actor.
* Filter the output to show only the movie type and actor name.
* group actor name and $push Movie genres from array \_id.movieType and count it.
* using the $sort in reverse option to list the highest genres number first.
* Finally, we show only the top 4 using limit.

**Results:**

{

"\_id" : "Alex Man",

"movie" : [

{

"genres" : [

"Crime",

"Drama",

"Romance"

],

"total" : 3

}

]

}

{

"\_id" : "Don McManus",

"movie" : [

{

"genres" : [

"Comedy",

"Drama",

"Romance"

],

"total" : 3

}

]

}

{

"\_id" : "Marley Shelton",

"movie" : [

{

"genres" : [

"Action",

"Comedy",

"Horror"

],

"total" : 3

}

]

}

{

"\_id" : "John Dall",

"movie" : [

{

"genres" : [

"Crime",

"Drama",

"Film-Noir"

],

"total" : 3

}

]

}

1. **Question 3**

List the four actors who have been acting for the largest number of years. Show the number of years, as well as the first and last year they have acted.

**Query:**

db.movies.aggregate([

{"$unwind": "$actors"},

{"$group": {"\_id": "$actors", "year": {"$addToSet": "$year"}}},

{"$project": {"Actor\_Name": "$\_id","MovieYears": {"$setUnion": ["$year"]},"\_id": 0 }},

{"$group": {"\_id": "$Actor\_Name","movie\_Year": { "$push":{"years": "$MovieYears","Numbers\_of\_Years":{$size: "$MovieYears"},"Start\_Year":{$min: "$MovieYears"},"End\_Year":{$max: "$MovieYears"}}}}},

{$sort:{"movie\_Year.Numbers\_of\_Years": -1}},

{$limit:4}

]).pretty()

Explanation:

* First, I break down the movie dataset into single records using "$unwind for actors filed.
* show records with only actor name and build list using $addToSet for years and filtering out \_id Filed.
* put new Records into a new project that include Actor name and years list and count the Years. Also, I used $min for years list to show the lowest year as actor start\_Yearand $max to show the biggest year number as End\_Year.
* sort the record in reverse for movie\_Year.Numbers\_of\_Years to show the highest years number.
* Finally, I filter the top 4 records using $limit.

**Results:**

{

"\_id" : "Natalie Portman",

"movie\_Year" : [

{

"years" : [

1999,

2000,

2002,

2005,

2008,

2009,

2011,

2013

],

"Numbers\_of\_Years" : 8,

"Start\_Year" : 1999,

"End\_Year" : 2013

}

]

}

{

"\_id" : "Louis C.K.",

"movie\_Year" : [

{

"years" : [

2007,

2008,

2009,

2010,

2011,

2013,

2015

],

"Numbers\_of\_Years" : 7,

"Start\_Year" : 2007,

"End\_Year" : 2015

}

]

}

{

"\_id" : "Tom Hanks",

"movie\_Year" : [

{

"years" : [

1988,

1995,

1999,

2002,

2006,

2010,

2013

],

"Numbers\_of\_Years" : 7,

"Start\_Year" : 1988,

"End\_Year" : 2013

}

]

}

{

"\_id" : "Ewan McGregor",

"movie\_Year" : [

{

"years" : [

1999,

2002,

2003,

2005,

2009,

2014

],

"Numbers\_of\_Years" : 6,

"Start\_Year" : 1999,

"End\_Year" : 2014

}

]

}

1. **Question 4**

From the directors who have produced more than three movies, list the top four directors whose movies have the highest average Metacritic score. Show the director names, the average Metacritic score, and the number of movies they have directed.

**Query:**

db.movies.aggregate([

{ $match: {"metacritic": {"$exists": true}}},

{"$unwind": "$director"},

{"$group": {"\_id": "$director","Movies": {"$addToSet": "$title"},"Metacritic": { "$addToSet": "$metacritic"}}},

{"$project": {"Director Name": "$\_id","Number Of Movies": {$size: "$Movies"},"Average Metacritic Score": {$avg: "$Metacritic"},"\_id": 0}},

{ $match: { "Number Of Movies": { $gt: 3}}},

{ $sort: { "Average Metacritic Score": -1}},

{$limit: 4}

]).pretty()

Explanation:

* First, I filtered only records contain Metacritic score using Match with "$exists": true option.
* then break down the movie dataset into single records using "$unwind for $director filed.
* Group records including director name and adding movies titles the direct to set and the same for metacritic.
* build project from prefuse records that include Director Name, Number Of Movies by counting the size of Movies array and Average Metacritic Score using the $avg on $Metacritic value list and filter out \_id field.
* filter only director who directed more than 3 movies using $match: and $gt: 3 for field Number Of Movies.
* Sort the results based on Average Metacritic Score value in reverse order.
* filter top 4 results.

**Results:**

{

"Director Name" : "Steven Spielberg",

"Number Of Movies" : 6,

"Average Metacritic Score" : 75.33333333333333

}

{

"Director Name" : "George Lucas",

"Number Of Movies" : 4,

"Average Metacritic Score" : 66.25

}

{

"Director Name" : "Sam Raimi",

"Number Of Movies" : 5,

"Average Metacritic Score" : 65.8

}

{

"Director Name" : "Robert Rodriguez",

"Number Of Movies" : 4,

"Average Metacritic Score" : 58.25

}

1. **Question 5**

Identify the types of movies that you generally enjoy. Compute a field that uses the information of the audience popularity of movies (i.e., general viewers as opposed to specialised critics). Compute also a field that contains the sum of awards and nominations the movies have received. List the top 5 movies that you would enjoy, sorted according to the popularity information you gathered. Show the title of the movies, the year they were produced, the audience popularity and the sum of awards and nominations you computed.

**Query:**

db.movies.aggregate([

{$match:

{"genres":

{$in: ["Crime", "Adventure"]}

}},

{"$project":

{"Movie Name": "$title", "Movie Year": "$year", "viewers reviews": "$tomato.userReviews", "specialised critics reviews ": "$tomato.reviews", "Total reviews":

{"$add":[

"$tomato.userReviews","$tomato.reviews"

]},

"awards wins": "$awards.wins","awards nominations ": "$awards.nominations","Total awards":

{"$add":[

"$awards.wins", "$awards.nominations"

]},

"\_id": 0}},

{$sort: {"Total reviews": -1}},

{$limit: 5}

]).pretty()

**Explanation:**

* First, I match the genres I like using $in: operation in match stage.
* build project from results that Title of the movies, movie Years, specialised critics reviews,viewers reviews ,Total reviews(sum of specialised critics and Viewer reviews ),awards wins ,awards.nominations , sum of awards wins and nominations and filtered out \_id Filed .
* Reverse Sorts records based on Total reviews Value.
* Show only top 5 results from previous Results.

**Results:**

{

"Movie Name" : "Star Wars: Episode III - Revenge of the Sith",

"Movie Year" : 2005,

"viewers reviews" : 33674396,

"specialised critics reviews " : 282,

"Total reviews" : 33674678,

"awards wins" : 21,

"awards nominations " : 49,

"Total awards" : 70

}

{

"Movie Name" : "Mr. & Mrs. Smith",

"Movie Year" : 2005,

"viewers reviews" : 33460533,

"specialised critics reviews " : 207,

"Total reviews" : 33460740,

"awards wins" : 9,

"awards nominations " : 18,

"Total awards" : 27

}

{

"Movie Name" : "The Day After Tomorrow",

"Movie Year" : 2004,

"viewers reviews" : 32748180,

"specialised critics reviews " : 212,

"Total reviews" : 32748392,

"awards wins" : 5,

"awards nominations " : 9,

"Total awards" : 14

}

{

"Movie Name" : "Saw II",

"Movie Year" : 2005,

"viewers reviews" : 32523656,

"specialised critics reviews " : 117,

"Total reviews" : 32523773,

"awards wins" : 3,

"awards nominations " : 7,

"Total awards" : 10

}

{

"Movie Name" : "Spider-Man 3",

"Movie Year" : 2007,

"viewers reviews" : 2262185,

"specialised critics reviews " : 246,

"Total reviews" : 2262431,

"awards wins" : 3,

"awards nominations " : 31,

"Total awards" : 34

}

**End..**