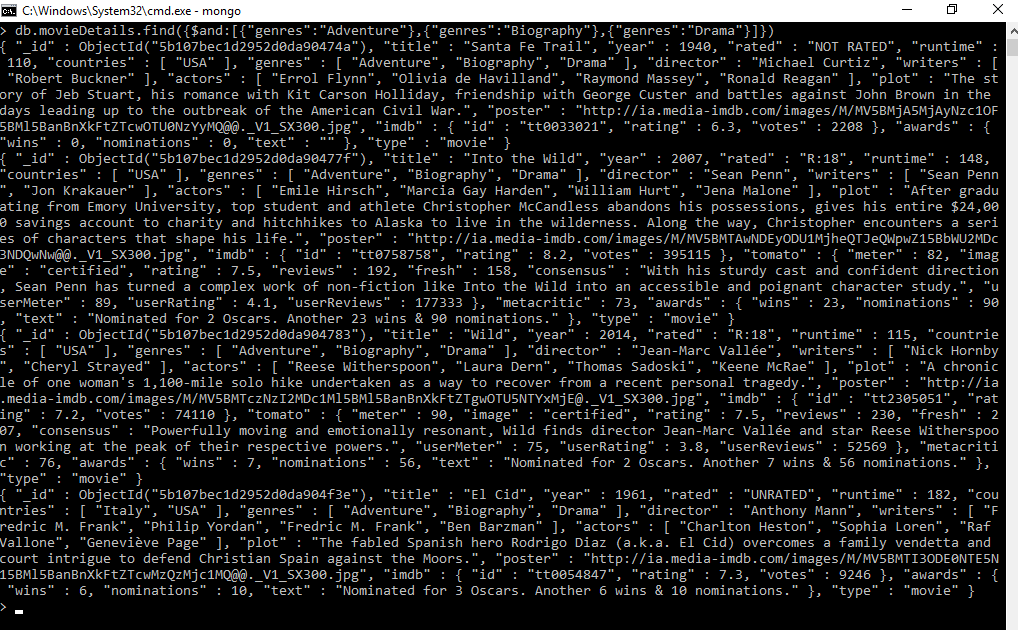
1.) Number of Movies with a runtime greater than 120mins

db.movieDetails.count({runtime:{$gt:120}})



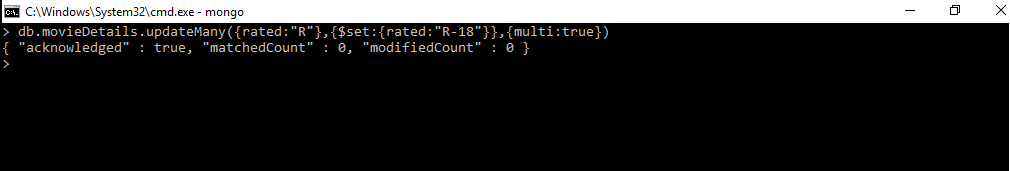
2.) Display the Movie/Movies that meets the genres Adventure, Biography, Drama

db.movieDetails.find({$and:[{“genres”:”Adventure”},{“genres”:”Biography”},{“genres”:”Drama”}]})



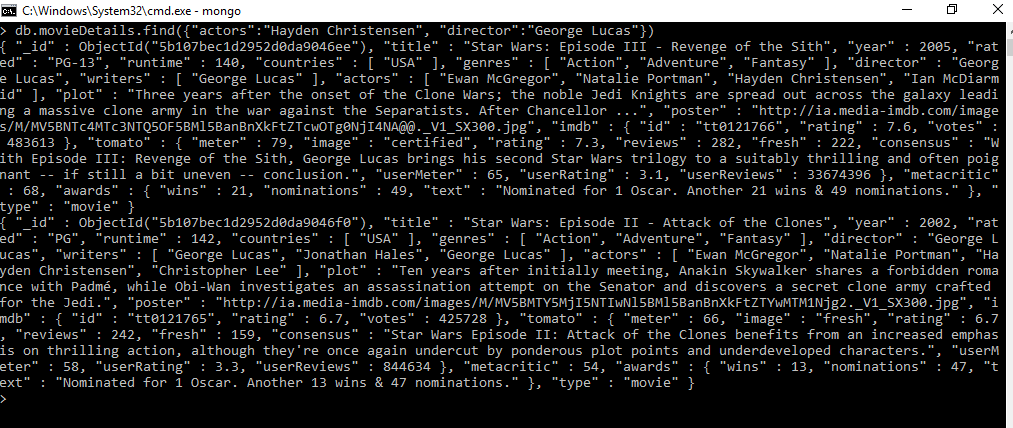
3.) Updated all the movies with a rated “R” rating to rated “R-18”

db.movieDetails.updateMany({rated:”R”},{$set:{rated:”R-18”}},{multi:true})



4.) Movies who starred actor “Hayden Christensen” and directed by “George Lucas”

db.movieDetails.find({“actors”:”Hayden Christensen”, “director”:”George Lucas”})

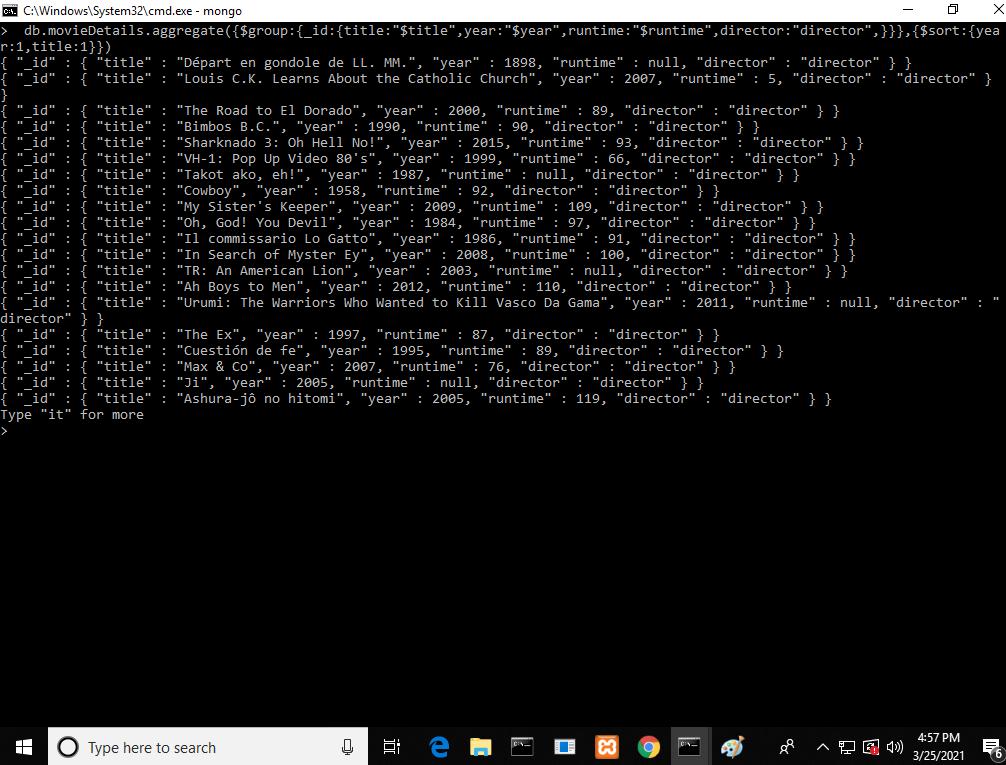


5.) Display movie name, year, runtime, and director. Order the result by year released in ascending

order.

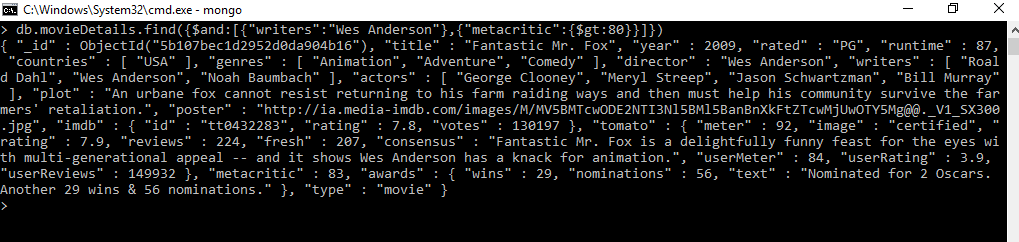
db.movieDetails.aggregate({$group:{\_id:{title:”$title”,year:”$year”,runtime:”$runtime”,director:”director”}

}},{$sort:{year:1, title:1}})



6.) Display movies that includes “Wes Anderson” as a writer with a metacritic score greater than 80

db.movieDetails.find({$and:[{“writers”:”Wes Anderson”},{“metacritic”:{$gt:80}}]})

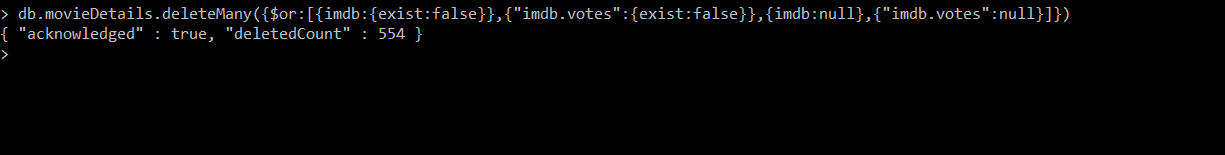


7.) Remove all movies without an imdb rating or votes or whose imdb ratings or votes values are

null

db.movieDetails.deleteMany({$or:{imdb:{exist:false}},{“imdb.votes”:{exist:false}},{imdb:null},

{“imdb.votes”:null}})



8.) Display only the movies that has all of the following:

a.

Imdb rating greater than 6

b. Imdb votes greater than 5,000

c.

a tomato meter greater than 70

d. a tomato rating greater than 5

e. a tomato userRating greater than 3

f.

a tomato userReviews greater than 1,000

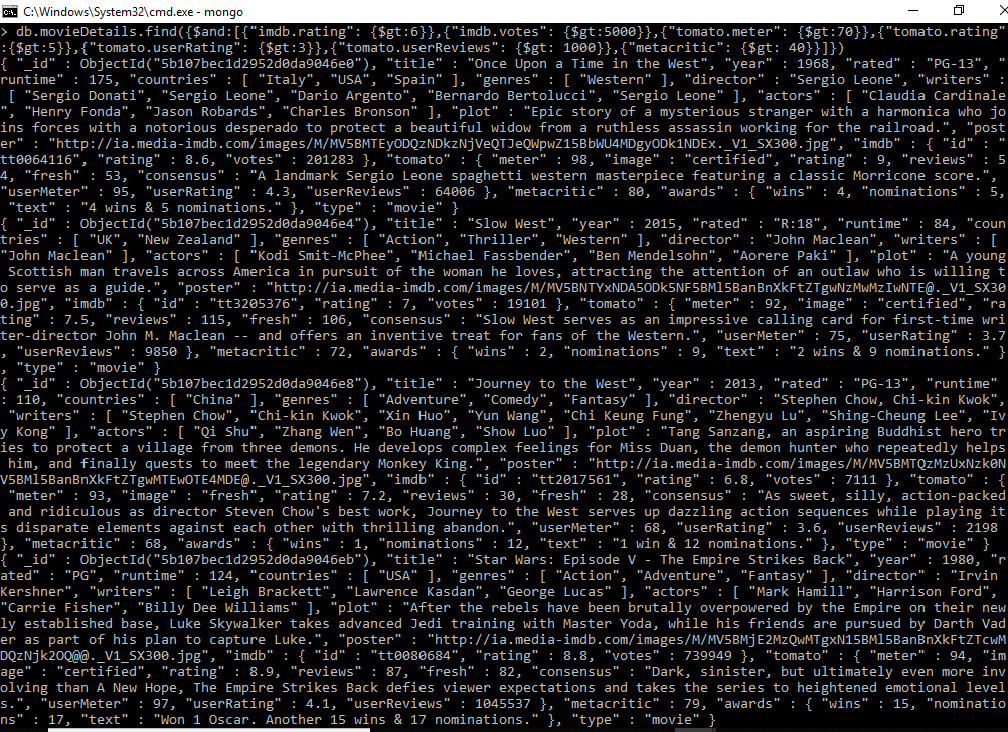
g.

a metacritic rating greater than 40

db.movieDetails.find({$and:[{“imdb.rating”:{$gt:6}},{“imdb.votes”:{$gt:5000}},

{“tomato.meter”:{$gt:70}},{“tomato.userRating”:{$gt:3}},{“tomato.userReviews”:

{$gt:1000}},{“metacritic”:{$gt:40}}]})

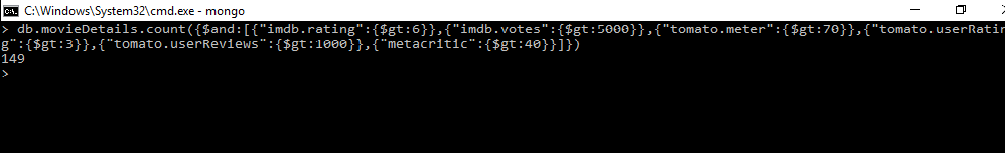


9.) Display the number of movies found on 8

db.movieDetails.count({$and:[{“imdb.rating”:{$gt:6}},{“imdb.votes”:{$gt:5000}},

{“tomato.meter”:{$gt:70}},{“tomato.userRating”:{$gt:3}},{“tomato.userReviews”:

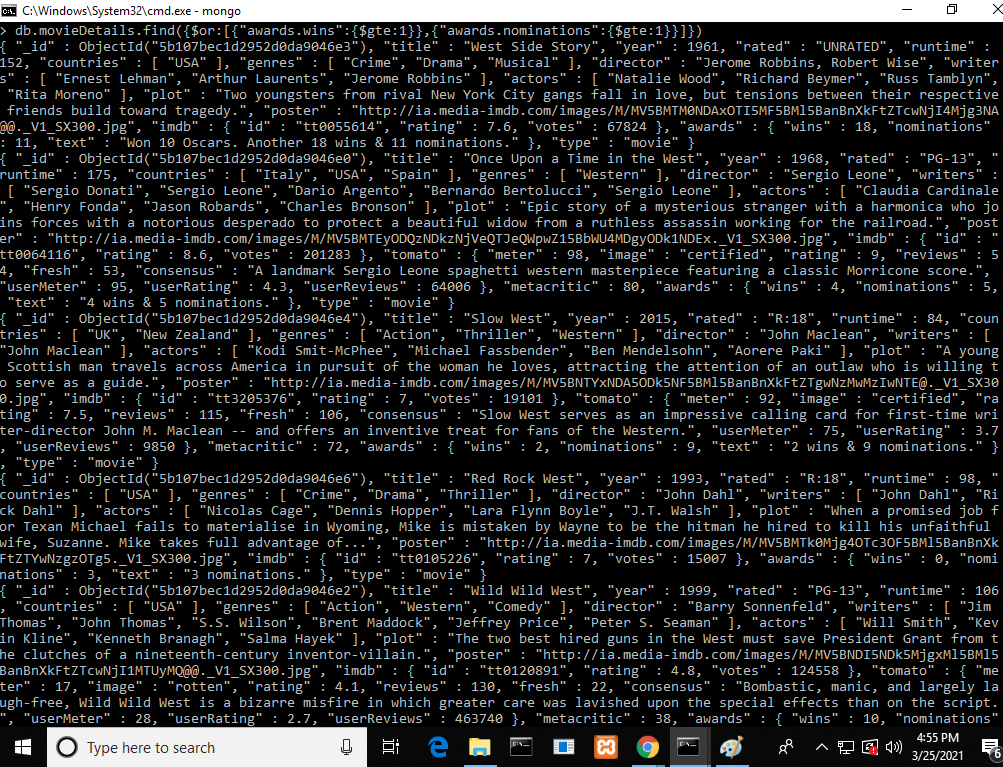
{$gt:1000}},{“metacritic”:{$gt:40}}]})



10.)Display all the movies that were either nominated for or won an Oscar. Display the movie name

,year, and number of award wins and nominations.

db.movieDetails.find({$or:[{“awards.wins”:{$gte:1}},{“awards.nominations”:{$gte:1}}]})



11.)Search for information on the 2019 movie Parasite, and add it as a document to this collection.

Required:

Title, Year, rated, runtime, countries featured in the movie, genres, director,

writers, actors, plot, and imdb{id: tt6751668 , rating:8.6, votes:491247}

If it is not required you can give it a null value.

The document added should follow the exact same format as other documents in the

collection.

db.movieDetails.insertOne({"title":"Parasite","year":2019,"rated":"R","runtime":132,"countries":

["South Korea"],"genres":["Black Comedy"],"director":"Bong Joon-ho","writers":["Bong Joon-ho",

"Han Jin-won"],"actors":["Song Kang-ho","Lee Sun-kyun","Cho Yeo-jeong","Choi Woo-shik","Park

So-dam","Lee Jung-eun","Jang Hye-jin"],"plot":"The Kim family—father Ki-taek, mother Chung-sook,

son Ki-woo and daughter Ki-jung—live in a small semi-basement apartment, The Kim family

recommends one another as unrelated and highly qualified workers to take over as servants of the

Parks. University student Min-hyuk, a friend of Ki-woo's, gives the family a scholar's rock meant to

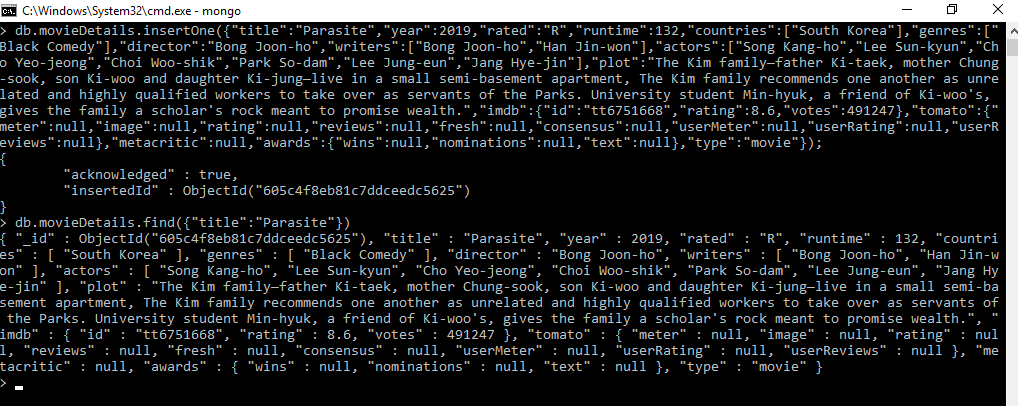
promise wealth.","imdb":{"id":"tt6751668","rating":8.6,"votes":491247},"tomato":{"meter":null,"image":

null,

"rating":null,"reviews":null,"fresh":null,"consensus":null,"userMeter":null,"userRating":null,

"userReviews":null},"metacritic":null,"awards":{"wins":null,"nominations":null,"text":null},"type":

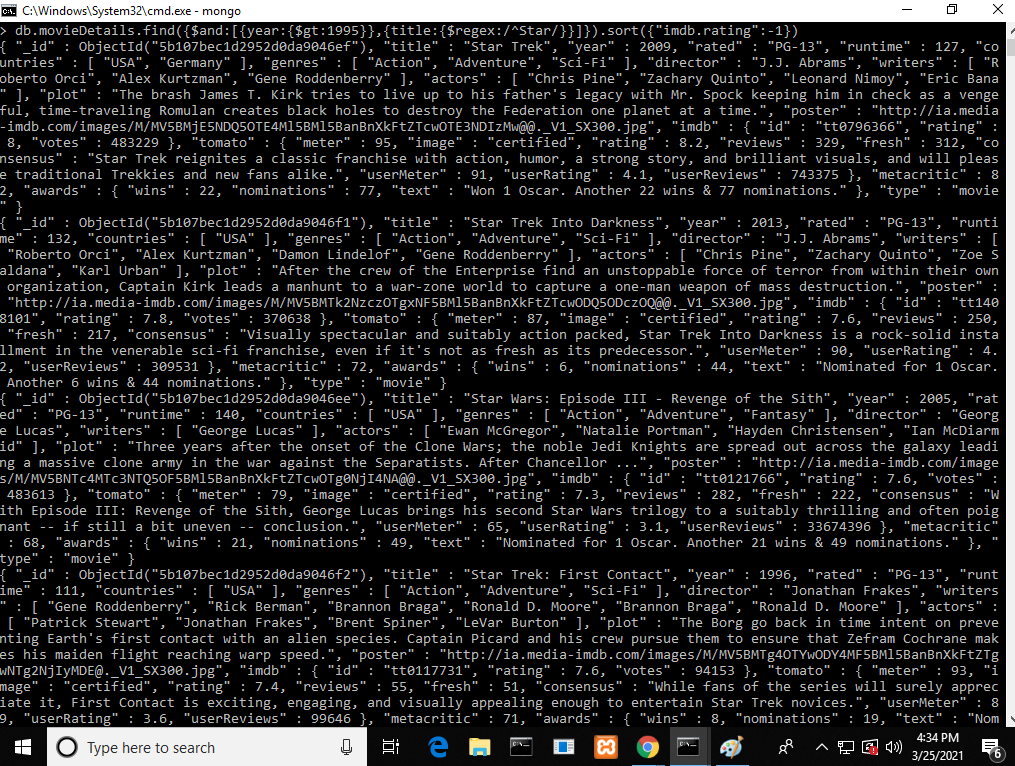
"movie"});



12.) Display documents in a readable manner that were released after 1995 and whose name starts

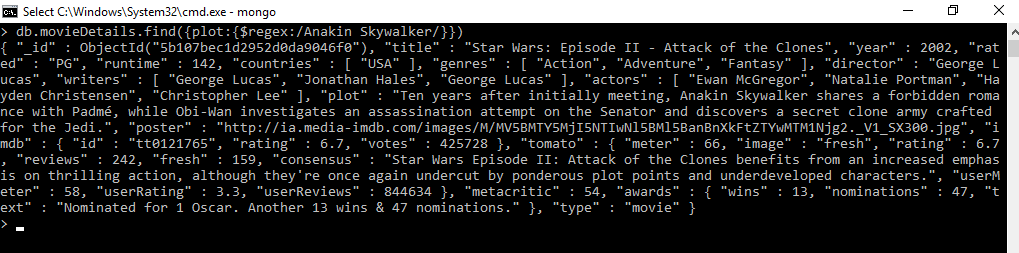
with the word “Star”. Sort by imdb ratings from descending order.

db.movieDetails.find({$and:[{year:{$gt:1995}},{title:{$regex:/^Star/}}]}).sort({“imdb.rating”:-1})



13.)Display the Movie/Movies that contain the name “Anakin Skywalker” in the plot

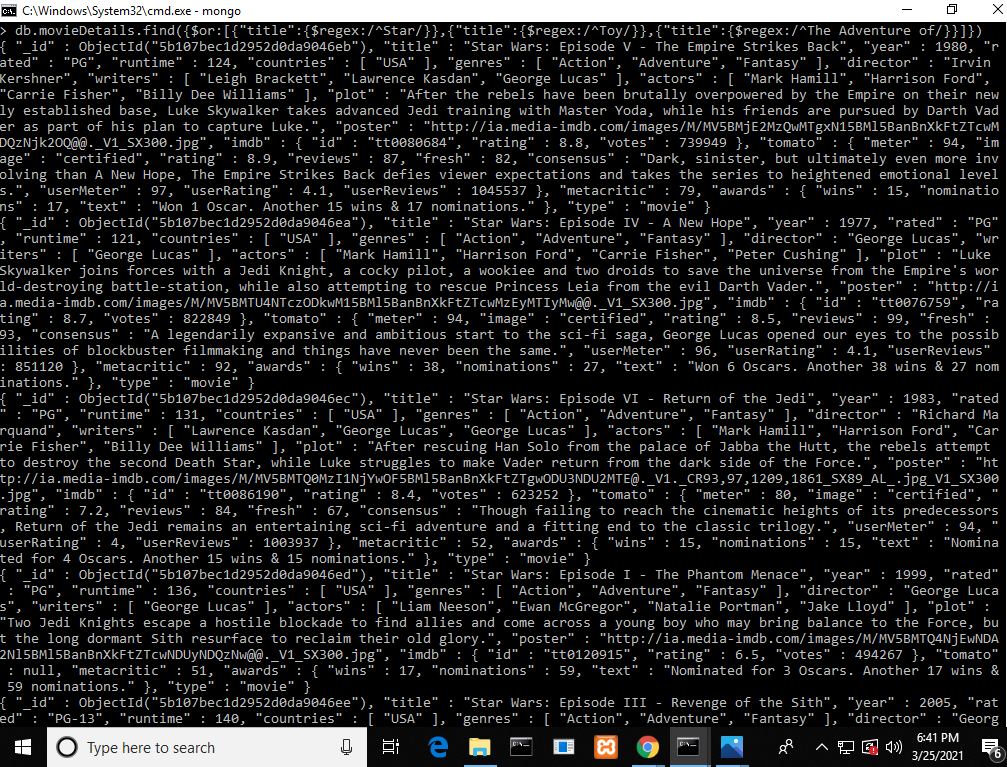
db.movieDetails.find({plot:{$regex:/Amakin Skywalker/}})



14.)Display all movies that start with the word/words “Star” , “Toy” , “The Adventure of”

db.movieDetails.find({$or:[{“title”:{$regex:/^Star/}},{“title”:{$regex:/^Toy/}},{“title”:

{$regex:/^The Adventure of/}}]})



Bonus:

Guess/Find one of my favorite movies through the use of mongoDB queries!

It is a

comedy, drama

starring

Robin Williams

that has won one

Oscar

award

db.movieDetails.find({$and:[{genres:[“Comedy”,”Drama”]},{actors:”Robin Williams”},{“awards.

text”:{$regex:/^Won 1 Oscar./}}]})

