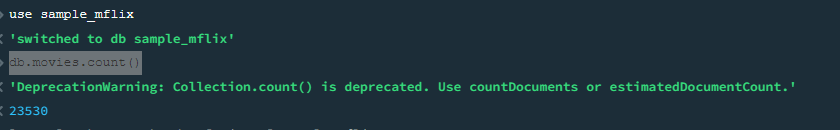
use sample\_mflix

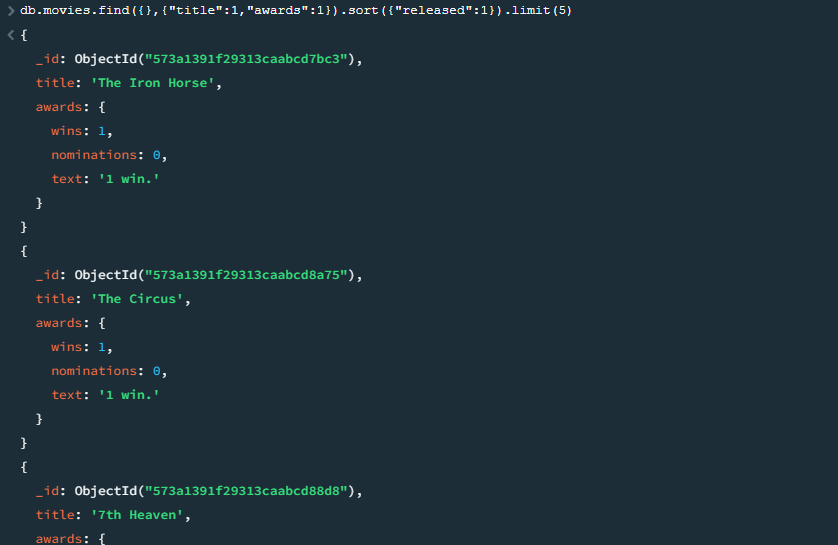
Q1. Count of movies

ANS: db.movies.count()



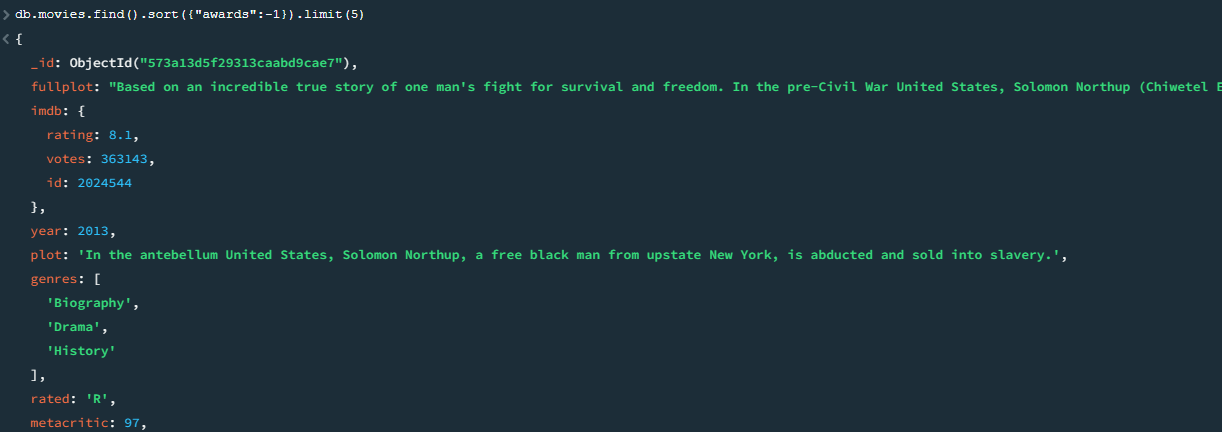
Q2. Display title and awards of the five oldest released movies

Ans : db.movies.find({},{"title":1,"awards":1}).sort({"released":1}).limit(5)



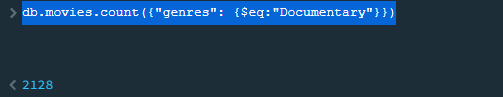
Q3. Display five movies that have won most number of awards

Ans: db.movies.find().sort({"awards":-1}).limit(5)



Q4. Display count of short documentaries

ANS: db.movies.count({"genres": {$eq:"Documentary"}})



Q5. Display all titles of movies with more than 3 awards in the adventure category.

ANS: db.movies.find({ $and:[{"genres": {$eq:"Adventure"}},{"awards.wins": {$gt: 3 }}]})

or

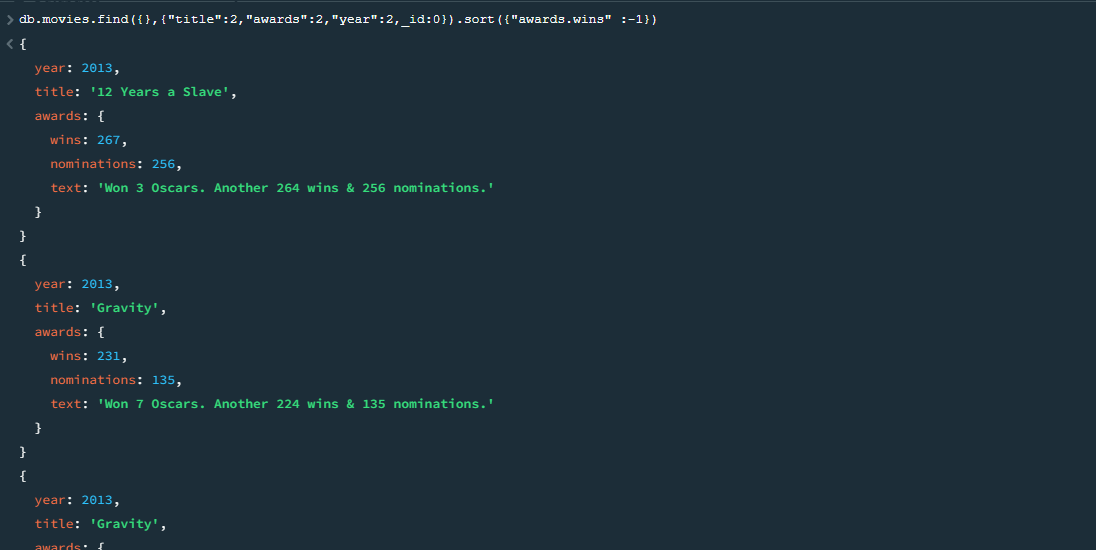
db.movies.find({"genres": {$eq:"Adventure"},"awards.wins": {$gt: 3 }})



.

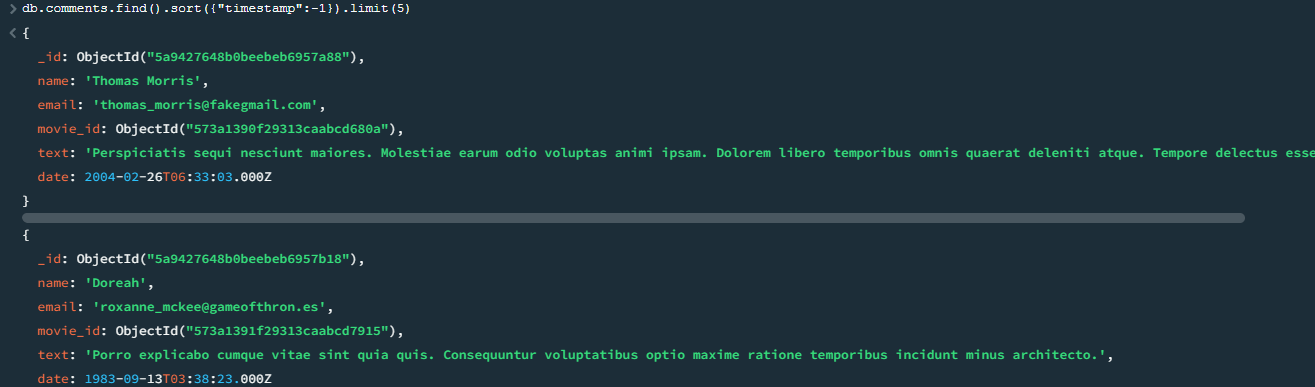
Q6. For each year, display title of the movie that has won the highest number of awards

ANS: db.movies.find({},{"title":2,"awards":2,"year":2,\_id:0}).sort({"awards.wins" :-1})



Q7. Display 5 latest comments

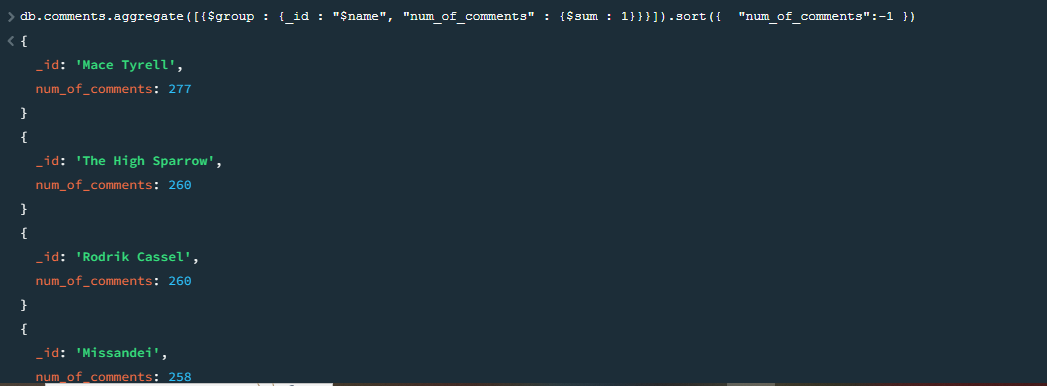
ANS: db.comments.find().sort({"timestamp":-1}).limit(5)

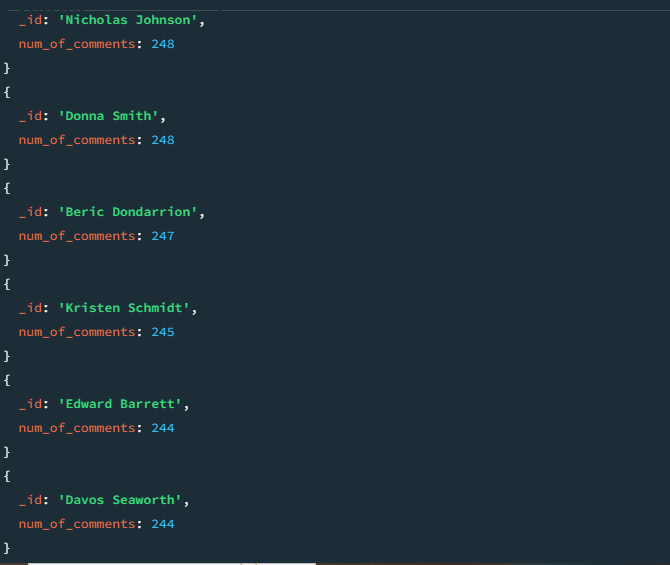


Q8. Display the name of the user with highest number of comments

ANS: FOR ALL NAME

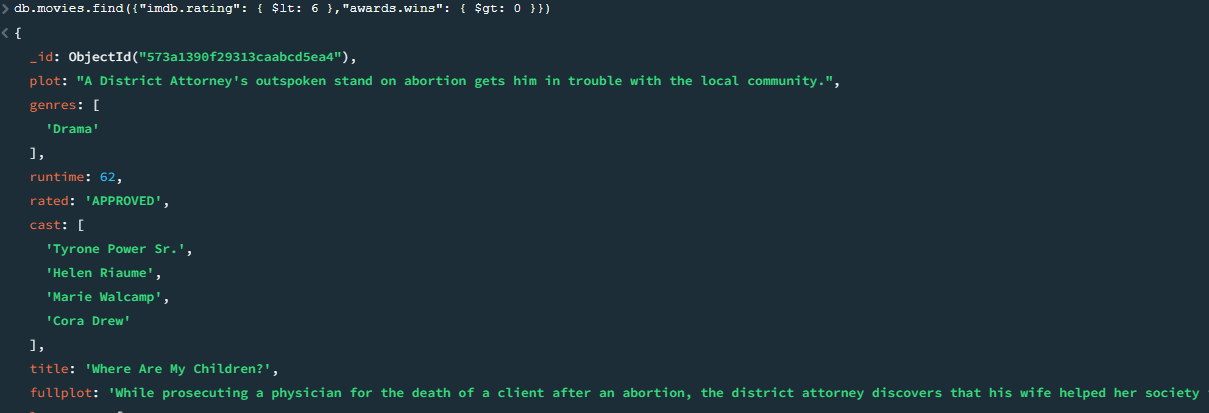
db.comments.aggregate([{$group : {\_id : "$name", "num\_of\_comments" : {$sum : 1}}}]).sort({ "num\_of\_comments":-1 })





Q9. Display movie with the IMDB rating less than 6 but has also won an award

ANS: db.movies.find({"imdb.rating": { $lt: 6 },"awards.wins": { $gt: 0 }})



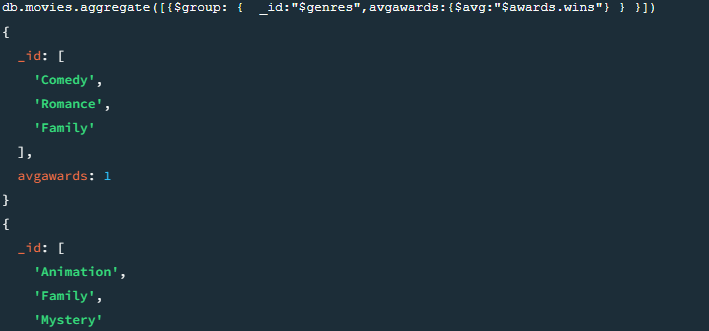
Q10. Display avg number of awards won, in each category.

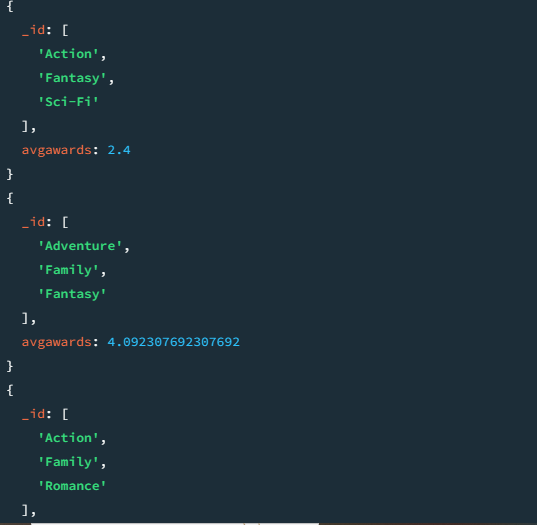
ANS :

Rough Work for checking distinct

db.movies.distinct("genres")

ANS: db.movies.aggregate([{$group: { \_id:"$genres",avgawards:{$avg:"$awards.wins"} } }])





Rough Work

db.movies.aggregate([{$unwind: "$awards"},{ $group: { \_id: "$awards.category",avgAwards: { $avg: "$awards.wins" } } },

{$project: {category: "$\_id", avgAwards: 1, \_id: 0}}])

db.movies.aggregate([{$group: { avgawards:{$avg:”$awards.wins”} } },])