first setup:
connect to mongodb atlas and use sample_mflix

Question & Anwser
Q1. Write a MongoDB query to display the total number of documents in the collection movies.

A1. db.movies.find().count()

R1. 23539

Q2. Write a MongoDB query to display any 5 documents using pretty format in the collection movies.

A2. db.movies.find().limit(5).pretty()

Q3. Write a MongoDB query to display 5 documents sorted by “title” using pretty format in the collection movies.

A3. db.movies.find().sort({"title":1}).limit(5).pretty()

Q4. Write a MongoDB query to display 5 documents (display only title and awards) sorted by “title” using pretty format in the collection movies.

A4. db.movies.find({},{"title":2,"awards":2,_id:0}).sort({"title": 1}).limit(5).pretty()

Q5. Write a MongoDB query to display 5 documents (display only title and awards) sorted by “title” in descending order using pretty format in the collection movies.

A5. db.movies.find({},{"title":2,"awards":2,_id: 0}).sort({"title":-1}).limit(5).pretty()

Q6. Write a MongoDB query to display movies (display only title and awards) with most awards (number of awards in descending order).

A6. db.movies.find({},{"title":2,"awards":2,_id: 0}).sort({"awards.wins" :-1}).limit(1).pretty()

Q7. Write a MongoDB query to display the details of the movie that won most awards

A7. db.movies.find().sort({"awards.wins":-1}).limit(1).pretty()

--

Q8. Write a MongoDB query to display any 5 movies with both the genres: “Adventure” and “Movie” in collection movies (use $all)

A8. db.movies.find({"genres":{$all:
["Adventure","Movie"]}}).limit(5).pretty()

Q9. Write a MongoDB query to display any 5 movies with both the condition: genres “Adventure” and cast “Tom Hanks”.

A9. db.movies.find({ $and:[{"genres":{$all:["Adventure"]}}, {"cast":
{$all:["Tom Hanks"]}}] }).limit(5).pretty()

Q10. Write a MongoDB query to display average number of awards won by a movie (use aggregate function with $avg operator).

A10. db.movies.aggregate([{ $group: {_id: "", num:
{ $sum:"$awards.wins" } } }])

Q11. Write a MongoDB query to display most awards won by a movie (use aggregate function with $max operator).

A11. db.movies.aggregate([{ $group: {_id: "", num:
{ $max:"$awards.wins" } } }])

---

comments:

Q1. Write a MongoDB query to display the total number of documents in the collection comments.

A1. db.comments.find().count()

Q2. Write a MongoDB query to display the total number of users by name (use db.collection.distinct(field, query, options) and length) in the collection comments.

A2. db.comments.distinct("name").length

Q3. Write a MongoDB query to display any 5 documents using pretty format in the collection comments.

A3. db.comments.find().limit(5).pretty()

Q4. Write a MongoDB query to display 5 documents sorted by name using pretty format in the collection comments.

A4. db.comments.find().sort({name:1}).limit(5).pretty()

Q5. Write a MongoDB query to display 5 latest comments (documents sorted by date in descending order) from "Megan Richards" using pretty format in the collection comments.

A5. db.comments.find({name:"Megan Richards"}).sort({ date:-1 }).limit(5).pretty()

Q6. Write a MongoDB query to display the total number of comments posted by a user (use aggregate function on name) in the collection comments.

A6. db.comments.aggregate([{$group : {_id : "$name", num_tutorial : {$sum : 1}}}])

Q7. Write a MongoDB query to display the maximum number of comments posted by a user (use aggregate function on name with $sum operator and then use $sort) in the collection comments.

A7. db.comments.aggregate([{$group : {_id : "$name", num_tutorial : {$sum : 1}}},{ $sort:{ "num_tutorial":-1 } }])

---

join M & C

Q1. Write a MongoDB query to join comments with movies that results in embedding movie information to the comments (using movie_id of comments and _id of movies)

A1. db.comments.find().forEach(
  function(res){
    movie_id = res.movie_id;
    com_id = res._id;
    movie_info = db.movies.findOne({_id:movie_id});
    db.comments.update({_id:com_id}, {$set: {"movie_info":movie_info}});
  }

)

Q2. Write a MongoDB query to find all movies that won greater than or equal to ($gte) the 3rd most awards winning movie. This query will have two parts. The nested query will first find the number of awards for 3rd most award winning movie. The outer query will find all movies that has more or equal awards than the 3rd most awards count return by the nested query.
You can also solve in a different way. Use a var to store the 3rd most awards count as returned by a query. Use the var to find all movies that has more or equal awards than var using $gte operator.

A2. db.movies.find({"awards.wins":
{$gte:db.movies.distinct("awards.wins").sort(function(a,b){
  return parseFloat(b)-parseFloat(a);
})[2]}}).pretty()

Map Reduce
mongoimport download/Users/annettechiu/Downloads/restaurants.jason -c=restaurants

Q1.Use Map Reduce over the restaurants collection to find the total number of times each grade was used for rating all different restaurants (using “grades.grade”).
A1.

Q2.Use Map Reduce over the restaurants collection to compute the average score for each grade. To compute this task you have to emit(“grades.grade”, “grades.score”).
A2.