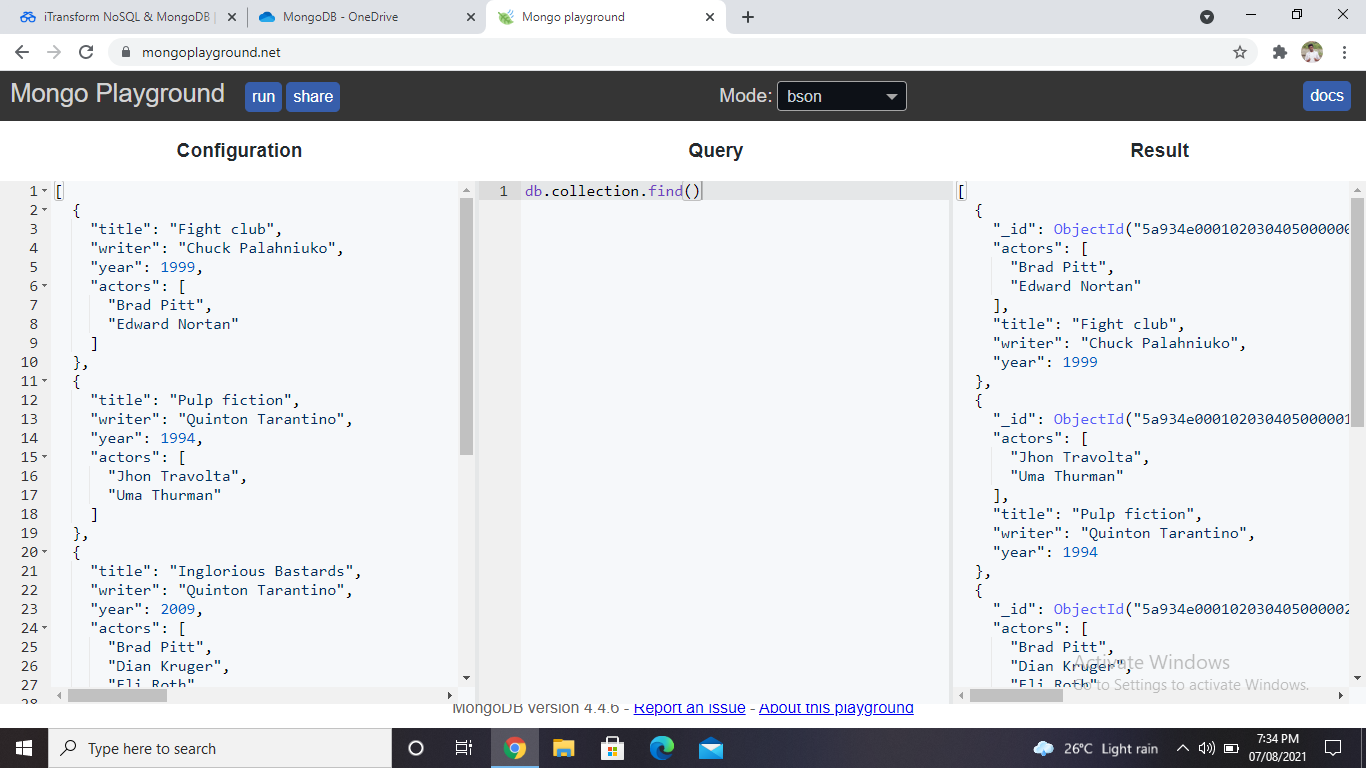
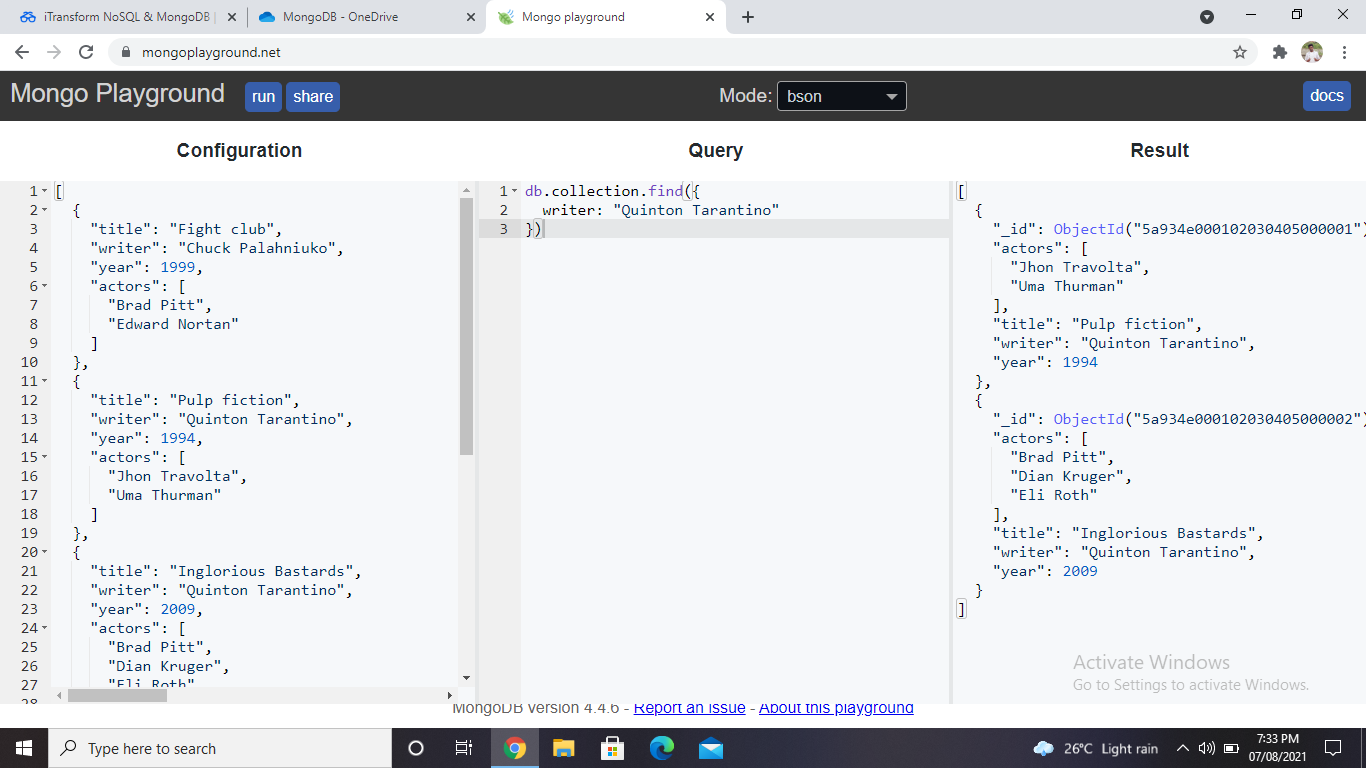
MongoDB-Assignments

Assignment-1:

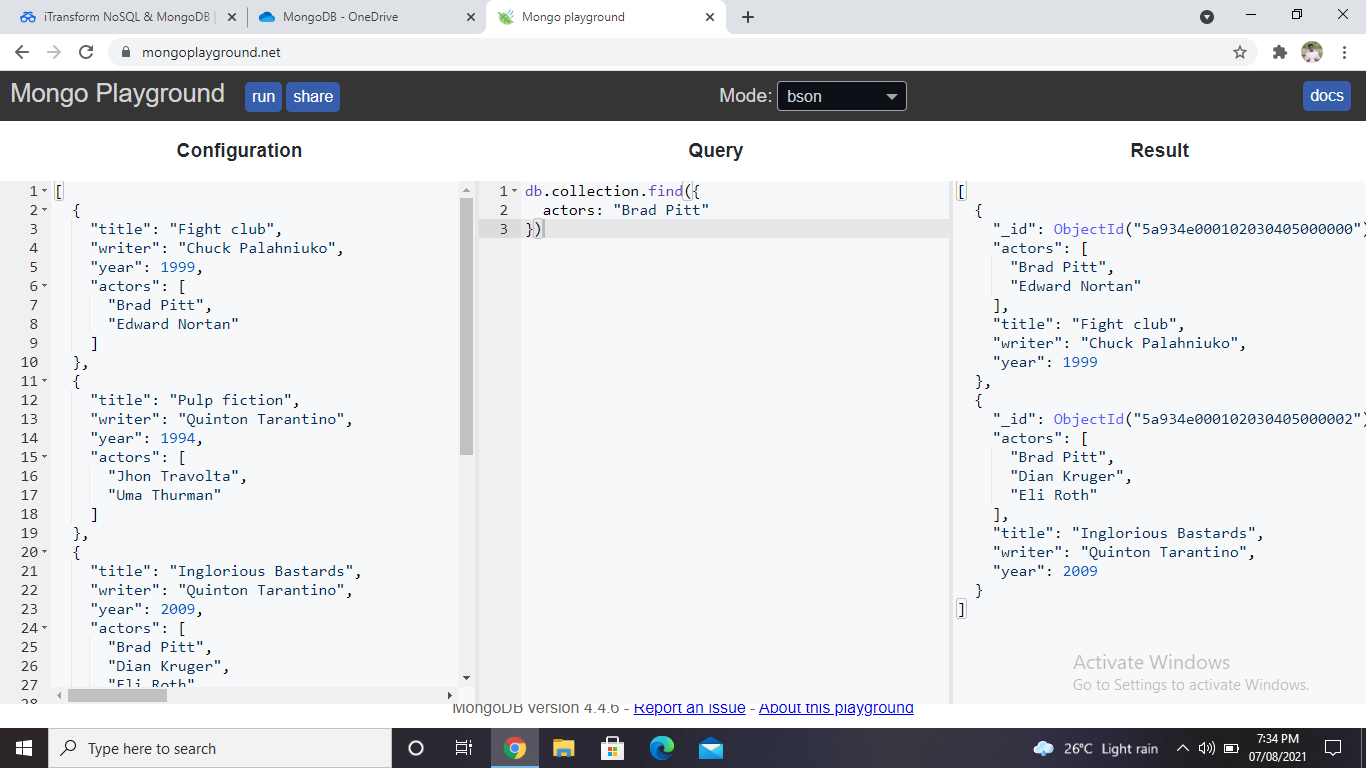
1. db.movies.find()



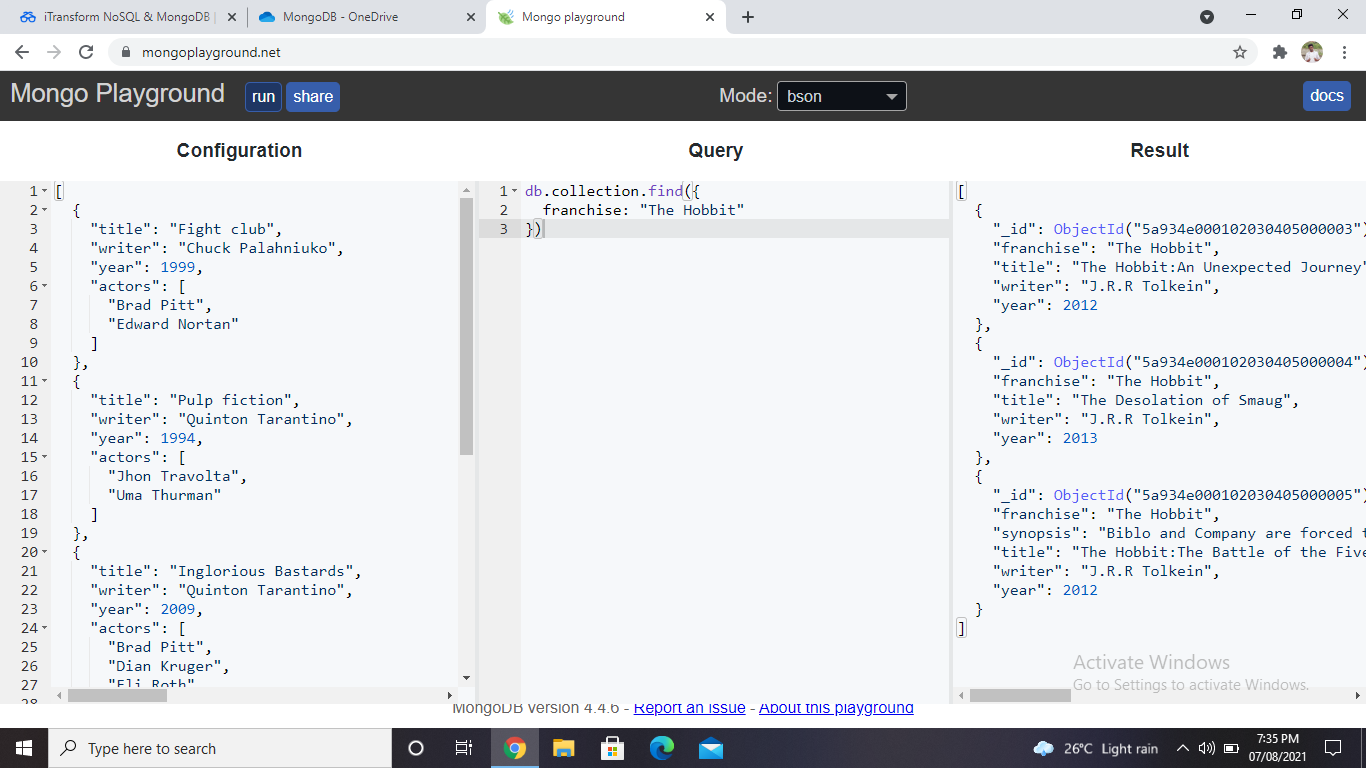
1. db.movies.find({writer: "Quentin Tarantino"})



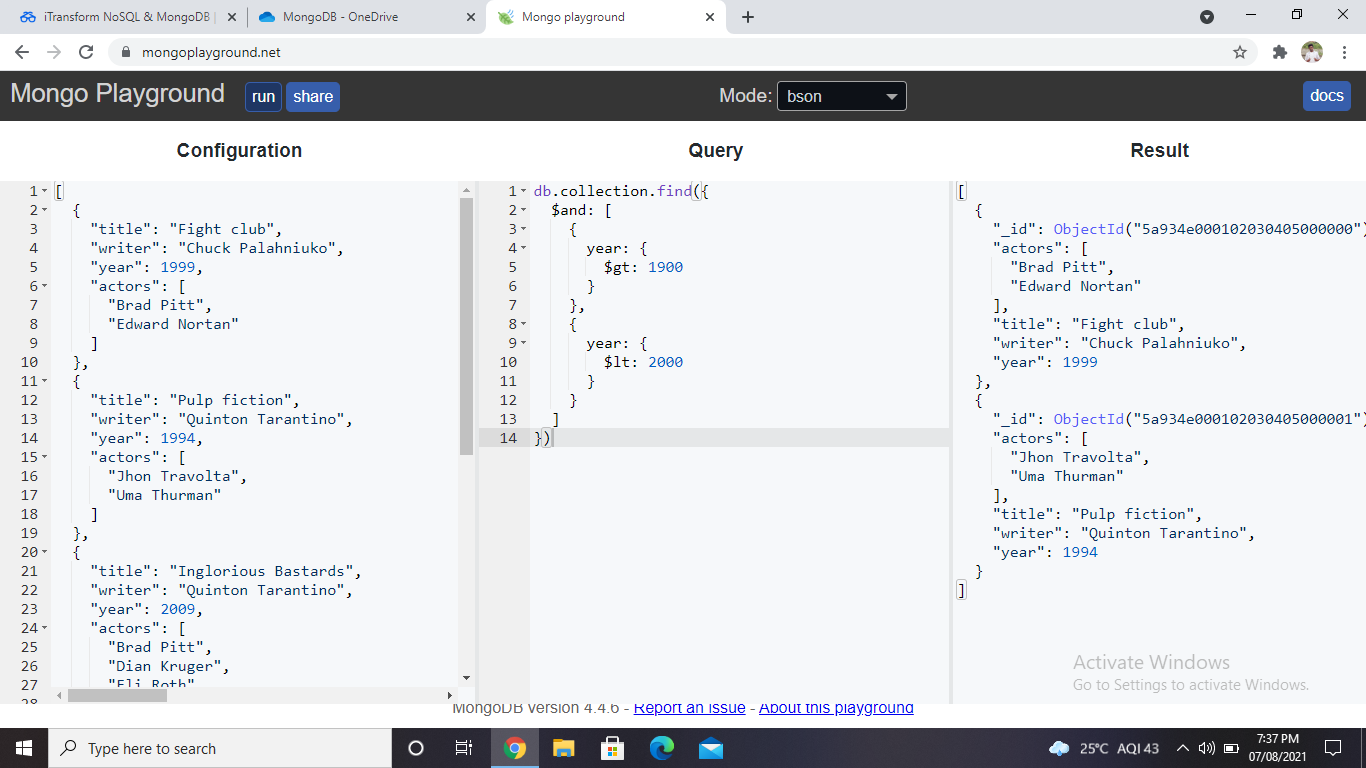
1. db.movies.find({actors: "Brad Pitt"})



1. db.movies.find({franchise: "The Hobbit"})



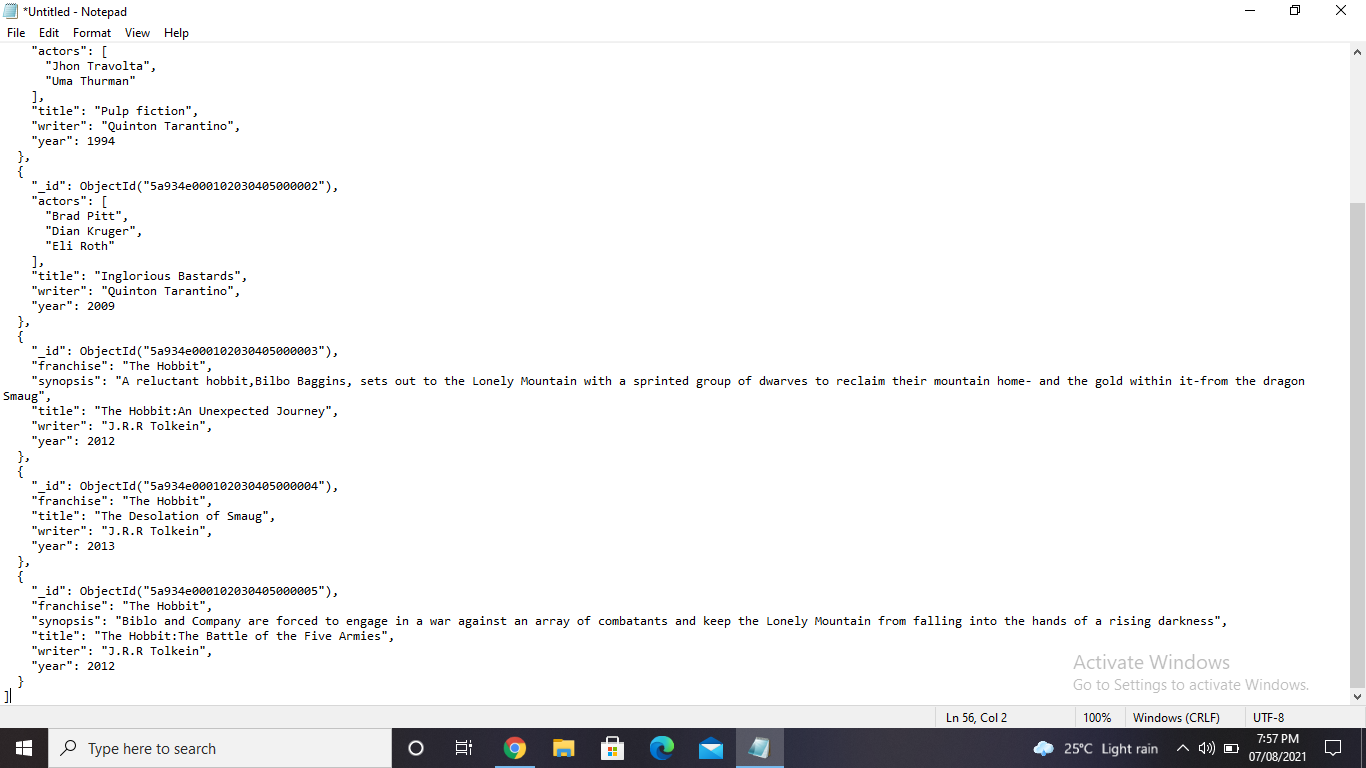
1. db.movies.find({$and: [{year: {$gt: 1900}}, {year: {$lt: 2000}}]})



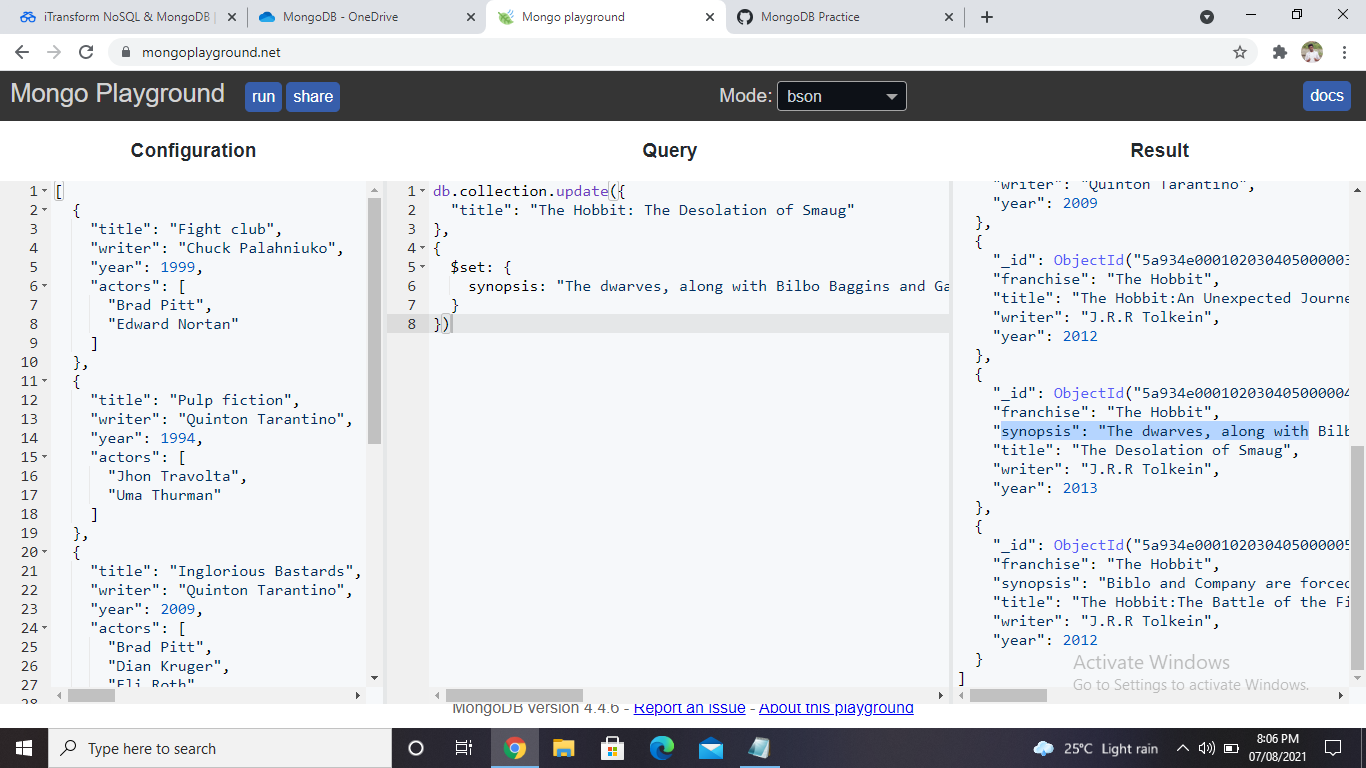
1. db.movies.find({$or: [{year: {$lt: 2000}}, {year: {$gt: 2010}}]})

UPDATE

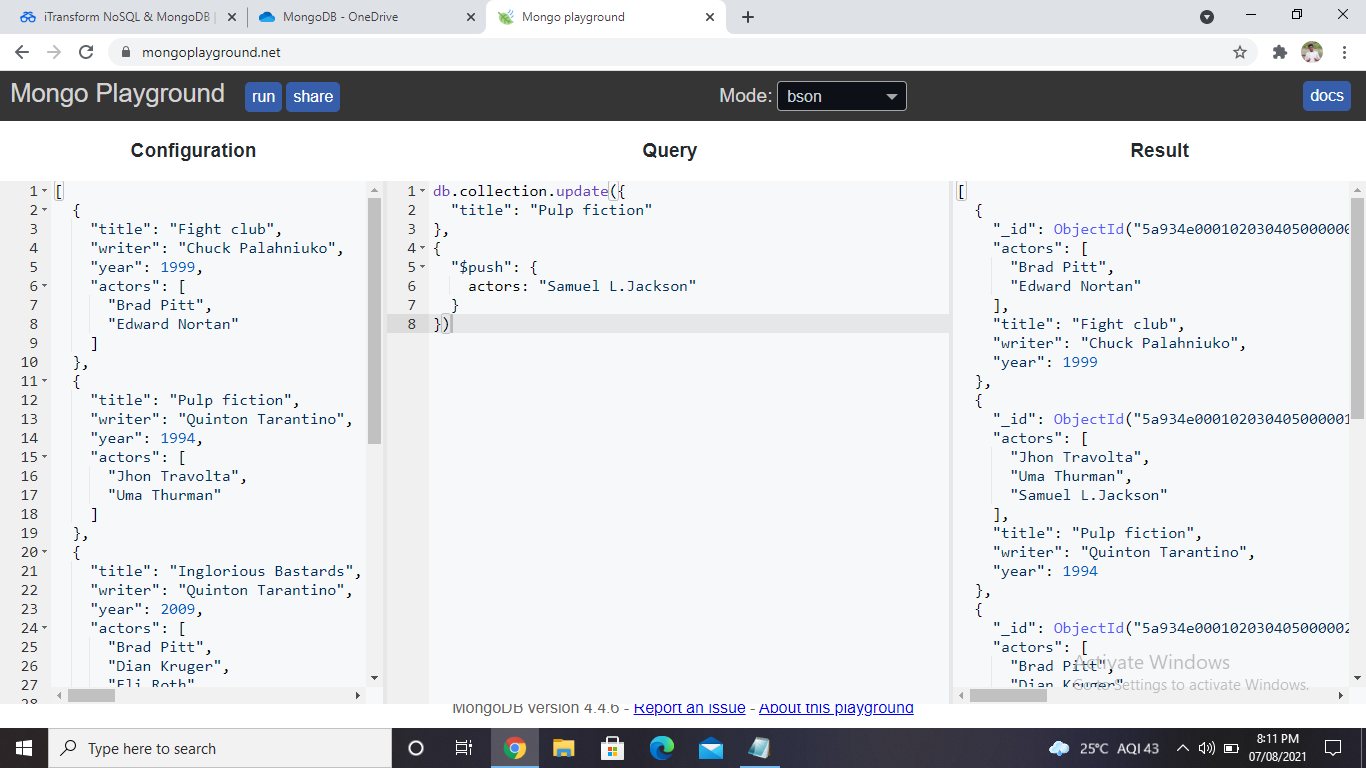
1. db.movies.update({title: "The Hobbit: An Unexpected Journey"}, {synopsis: "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."})



1. db.movies.update({title: "The Hobbit: The Desolation of Smaug"}, {synopsis: "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."})



1. db.movies.update({title: "Pulp Fiction"}, {$push: {actors: "Samuel L. Jackson"}})



TEXT SEARCH

1. db.movies.find({$text: {$search: "Bilbo"}})
2. db.movies.find({$text: {$search: "Gandalf"}})
3. db.movies.find({$text: {$search: "Bilbo -Gandalf"}})
4. db.movies.find({$text: {$search: "dwarves hobbit"}})
5. db.movies.find({$text: {$search: "gold dragon"}})

DELETE DOCUMENT

1. db.movies.remove({title: "Pee Wee Herman's Big Adventure"})
2. db.movies.remove({title: "Avatar"})

Users,Posts,Comments

1. db.users.find()
2. db.posts.find()
3. db.posts.find({username: "GoodGuyGreg"})
4. db.posts.find({username: "ScumbagSteve"})
5. db.comments.find()
6. db.comments.find({username: "GoodGuyGreg"})
7. db.comments.find({username: "ScumbagSteve"})

Assignment-2:

ATLANTA Population

1. db.zipcodes.find({city:"ATLANTA",state:"GA"})

2. db.zipcodes.aggregate([{$match: {city:"ATLANTA",state:"GA"}}])

3.db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$group:{\_id:"$city",total\_zip:{$sum:1}}}])

4.db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$group:{\_id:"$city",total:{$sum:"$pop"}}}])

Populations By State:

1. db.zipcodes.aggregate([{$group:{\_id:"$state",count:{$sum:"$pop"}}}])
2. db.zipcodes.aggregate([{$sort:{pop:-1}}])
3. db.zipcodes.aggregate([{$sort:{pop:-1}},{$limit:3}])

Population By City:

1. db.zipcodes.aggregate([{$group:{\_id:{city:"$city",state:"$state",total\_population:{$sum:"$pop"}}}}])
2. db.zipcodes.aggregate([{$sort:{pop:-1}}])
3. db.zipcodes.aggregate([{$group:{\_id:"$city"}},{$sort:{"pop":-1}},{$limit:3}])
4. db.zipcodes.aggregate([{$group:{\_id:{state:"Texas",city:"$city"}}},{$sort:{"pop":-1}},{$limit:3}])

Bonus:

1. db.zipcodes.aggregate([{$group:{\_id:{state:"$state"},avg:{$avg:"$pop"}}}])
2. db.zipcodes.aggregate([{$group:{\_id:{state:"$state"},avg:{$avg:"$pop"}}},{$sort:{pop:-1}},{$limit:3}])

Assignment-3:

Exercise Questions:

1. db.addresses.find().pretty()
2. db.addresses.aggregate([{$project:{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}}])
3. db.addresses.aggregate([{$project:{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1,"\_id":0}}])
4. db.addresses.aggregate([{$project:{"restaurant\_id":1,"name":1,"borough":1,"address.zipcode":1,"\_id":0}}])
5. db.addresses.aggregate([{$match:{borough:"Bronx"}},{$limit:5}]).pretty()
6. db.addresses.aggregate([{$match:{borough:"Bronx"}}]).pretty()
7. db.addresses.aggregate([{$match:{borough:"Bronx"}},{$skip:5},{$limit:5}]).pretty()
8. db.addresses.find({"grades.score":{$gt:90}}).pretty()
9. db.addresses.find({$and:[{"grades.score":{$gt:80}},{"grades.score":{$lt:100}}]}).pretty()
10. db.addresses.find({"address.coord.0":{$lt:-95.754168}}).pretty()
11. db.addresses.find({$and:[{"grades.score":{$gt:70}},{"address.coord.0":{$lt:-65.754168}},{cuisine:{$not:{$regex:"American"}}}]}).pretty()
12. db.addresses.find({$and:[{cuisine:{$not:{$regex:"American"}}},{"grades.grade":"A"},{borough:{$not:{$regex:"Brooklyn"}}}]}).sort({cuisine:-1}).pretty()
13. db.addresses.find({name:/^Wil/},{restaurant\_id:1,name:1,borough:1,cuisine:1})
14. db.addresses.find({name:/ces$/},{restaurant\_id:1,name:1,borough:1,cuisine:1})
15. db.addresses.find({name:/Reg/},{restaurant\_id:1,name:1,borough:1,cuisine:1})
16. db.addresses.find({borough:"Bronx",$or:[{cuisine:"American"},{cuisine:"Chinese"}]}).pretty()
17. db.addresses.find({$or:[{borough:"Staten Island"},{borough:"Queens"},{borough:"Bronxor"},{borough:"Brooklyn"}]},{restaurant\_id:1,name:1,borough:1,cuisine:1})
18. db.addresses.find({$and:[{borough:{$not:/Staten Island/}},{borough:{$not:/Queens/}},{borough:{$not:/Bronx/}},{borough:{$not:/Brooklyn/}}]},{restaurant\_id:1,name:1,borough:1,cuisine:1})
19. db.addresses.find({"grades.score":{$lt:10}},{restaurant\_id:1,name:1,borough:1,cuisine:1})
20. db.addresses.find({$or:[{name:/^Wil/},{$and:[{cuisine:{$not:{$regex:"American"}}},{cuisine:{$not:{$regex:"Chinese"}}}]}]},{restaurant\_id:1,name:1,borough:1,cuisine:1})
21. db.addresses.find({$and:[{"grades.grade":"A"},{"grades.score":11},{"grades.date":ISODate("2014-08-11T00:00:00Z")}]},{restaurant\_id:1,name:1,grades:1}).pretty()
22. db.addresses.find({$and:[{"grades.grade":"A"},{"grades.score":9},{"grades.date":ISODate("2014-08-11T00:00:00Z")}]},{restaurant\_id:1,name:1,grades:1}).pretty()
23. db.addresses.find({"address.coord.1":{$gt:42,$lte:52}},{restaurant\_id:1,name:1,address:1}).pretty()
24. db.addresses.find().sort({name:1}).pretty()
25. db.addresses.find().sort({name:-1}).pretty()
26. db.addresses.find().sort({cuisine:1,borough:-1}).pretty()
27. db.addresses.find({"address.coord":{$type:"double"}}).pretty()
28. db.addresses.find({"grades.score":{$mod:[7,0]}},{restaurant\_id:1,name:1,grades:1}).pretty()
29. db.addresses.find({name:/mon/},{name:1,borough:1,cuisine:1,"address.coord":1})
30. db.addresses.find({name:/^Mad/},{name:1,borough:1,cuisine:1,"address.coord":1})