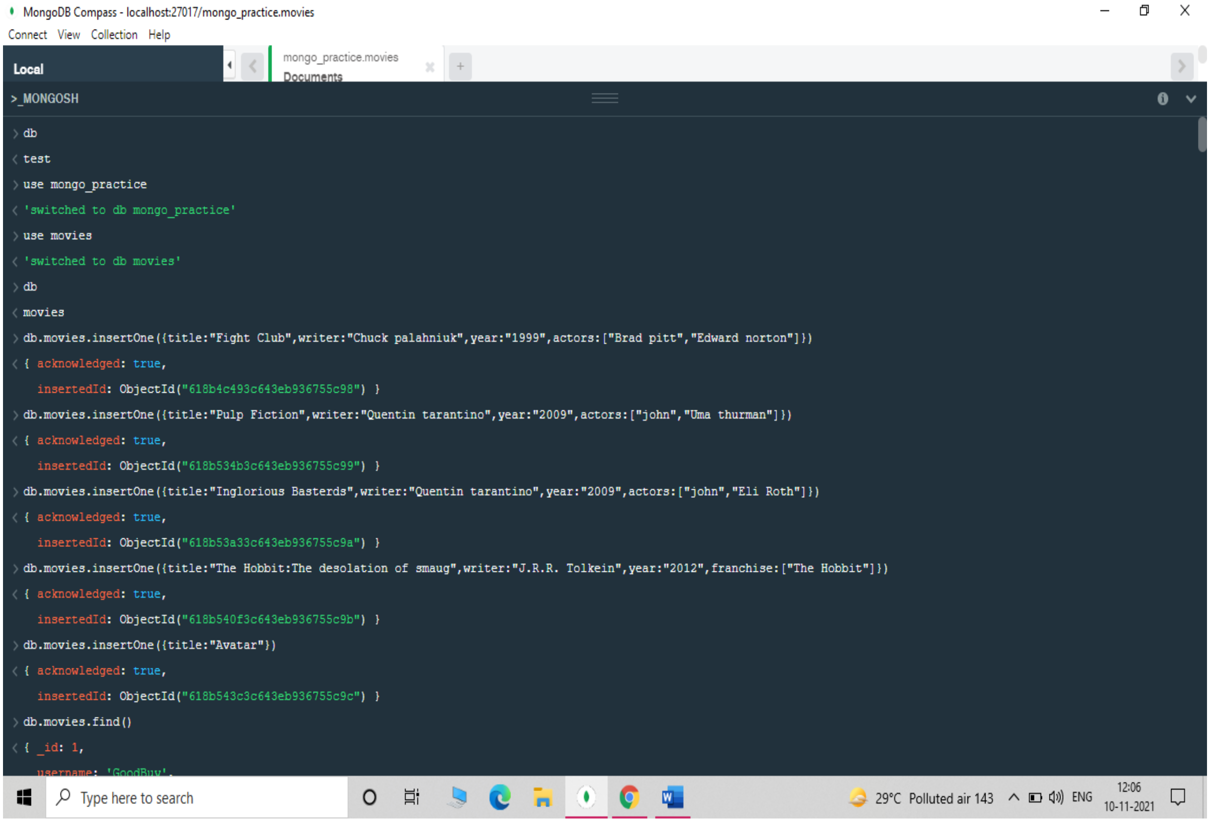
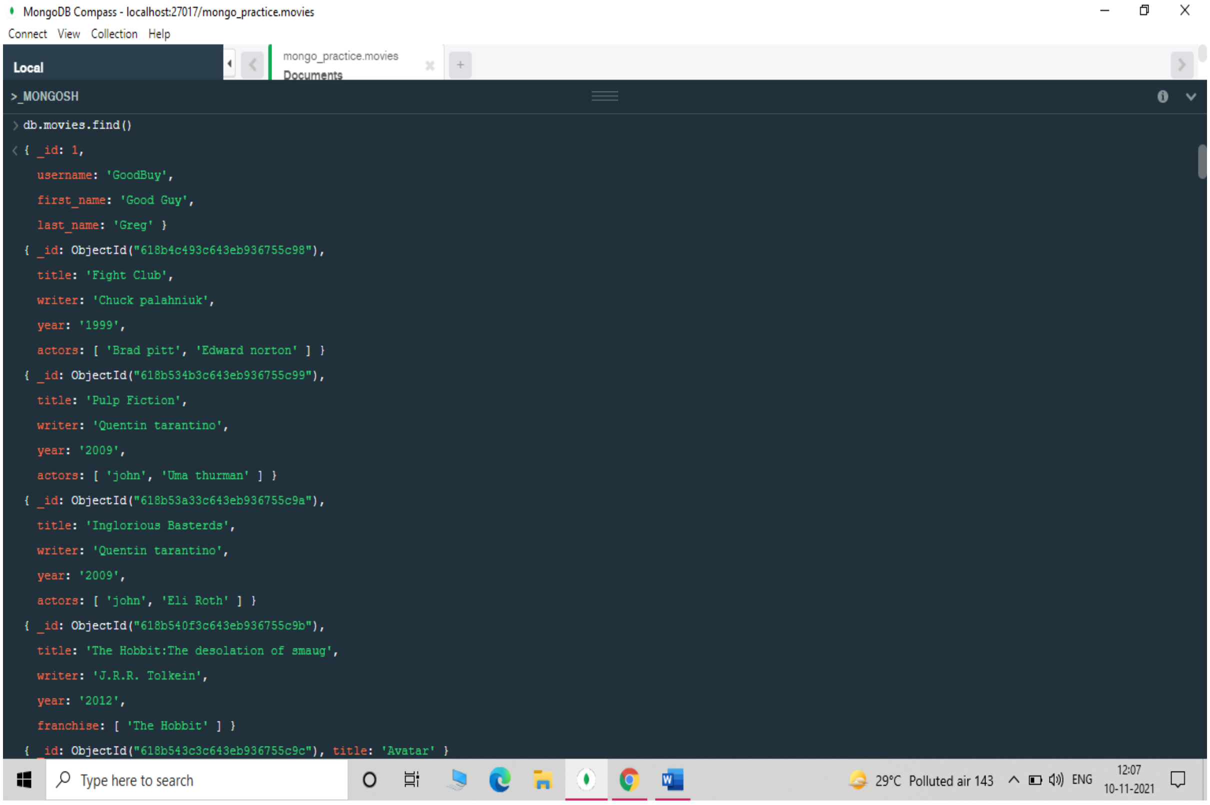
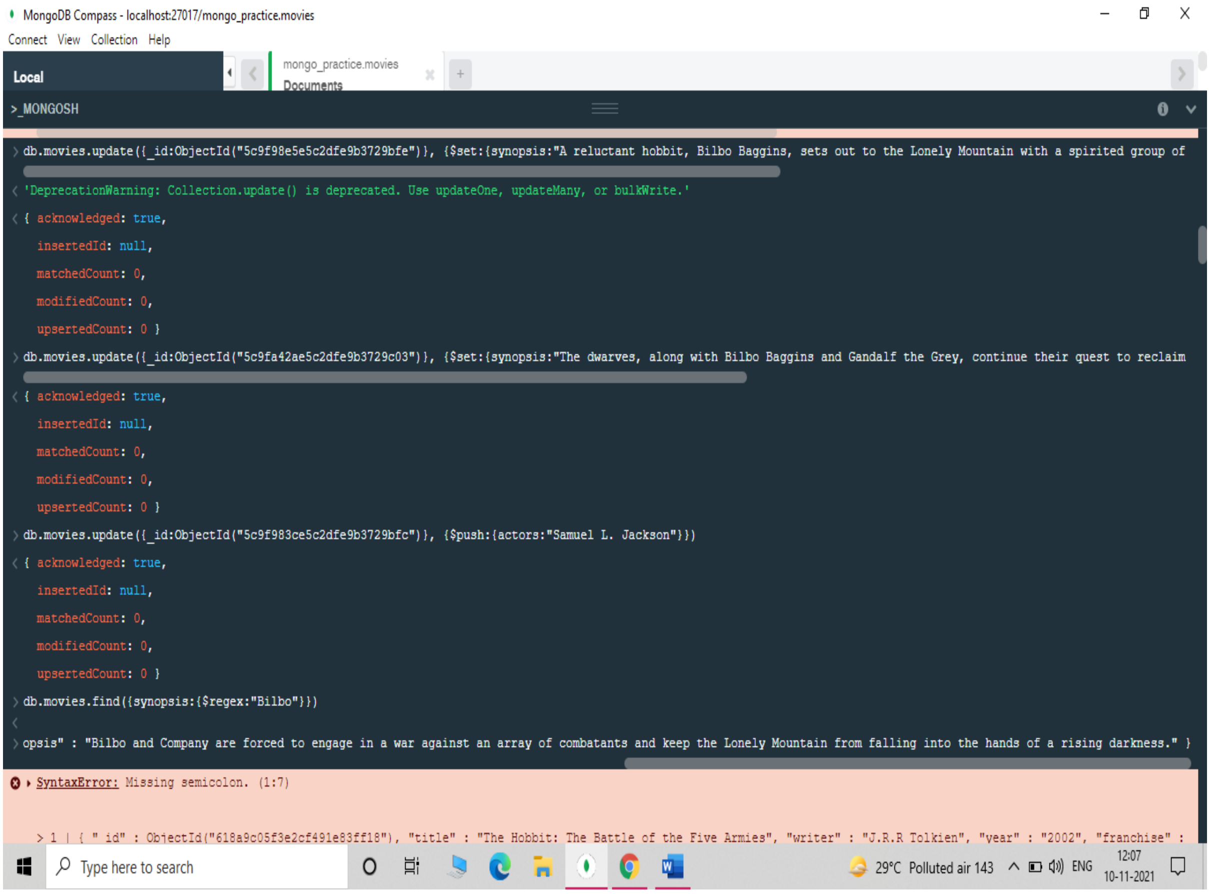
MONGODB ASSIGNMENTS



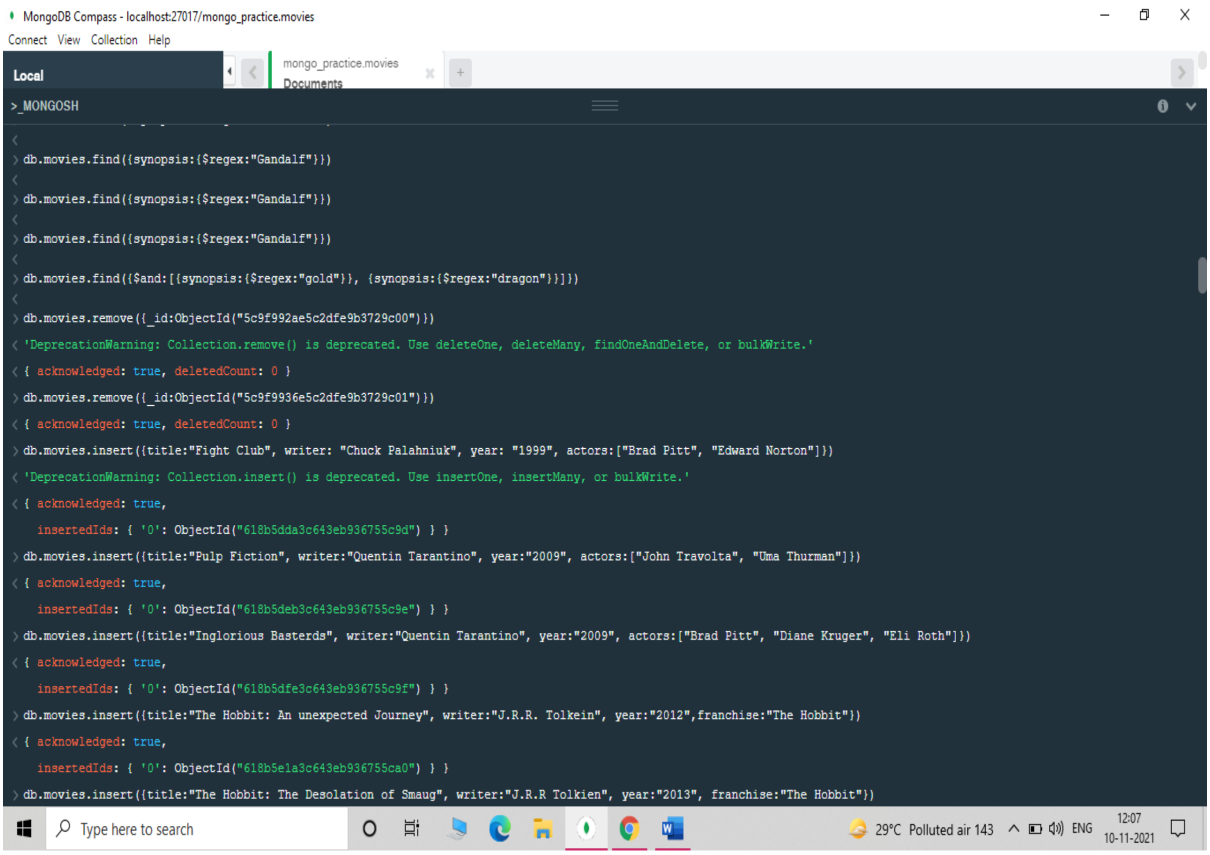
```
> db
< test
> use mongo_practice
< 'switched to db mongo_practice'
> use movies
< 'switched to db movies'
> db
< movies
> db.movies.insertOne({title:"Fight Club",writer:"Chuck palahniuk",year:"1999",actors:["Brad pitt","Edward norton"]})
< { acknowledged: true,
    insertedId: ObjectId("618b4c493c643eb936755c98") }
> db.movies.insertOne({title:"Pulp Fiction",writer:"Quentin tarantino",year:"2009",actors:["john","Uma thurman"]})
< { acknowledged: true,
    insertedId: ObjectId("618b534b3c643eb936755c99") }
> db.movies.insertOne({title:"Inglorious Basterds",writer:"Quentin tarantino",year:"2009",actors:["john","Eli Roth"]})
< { acknowledged: true,
    insertedId: ObjectId("618b53a33c643eb936755c9a") }
> db.movies.insertOne({title:"The Hobbit:The desolation of smaug",writer:"J.R.R. Tolkein",year:"2012",franchise:["The Hobbit"]})
< { acknowledged: true,
    insertedId: ObjectId("618b540f3c643eb936755c9b") }
> db.movies.insertOne({title:"Avatar"})
< { acknowledged: true,
    insertedId: ObjectId("618b543c3c643eb936755c9c") }
> db.movies.find()
< { _id: 1,
    username: 'GoodBuy',
```



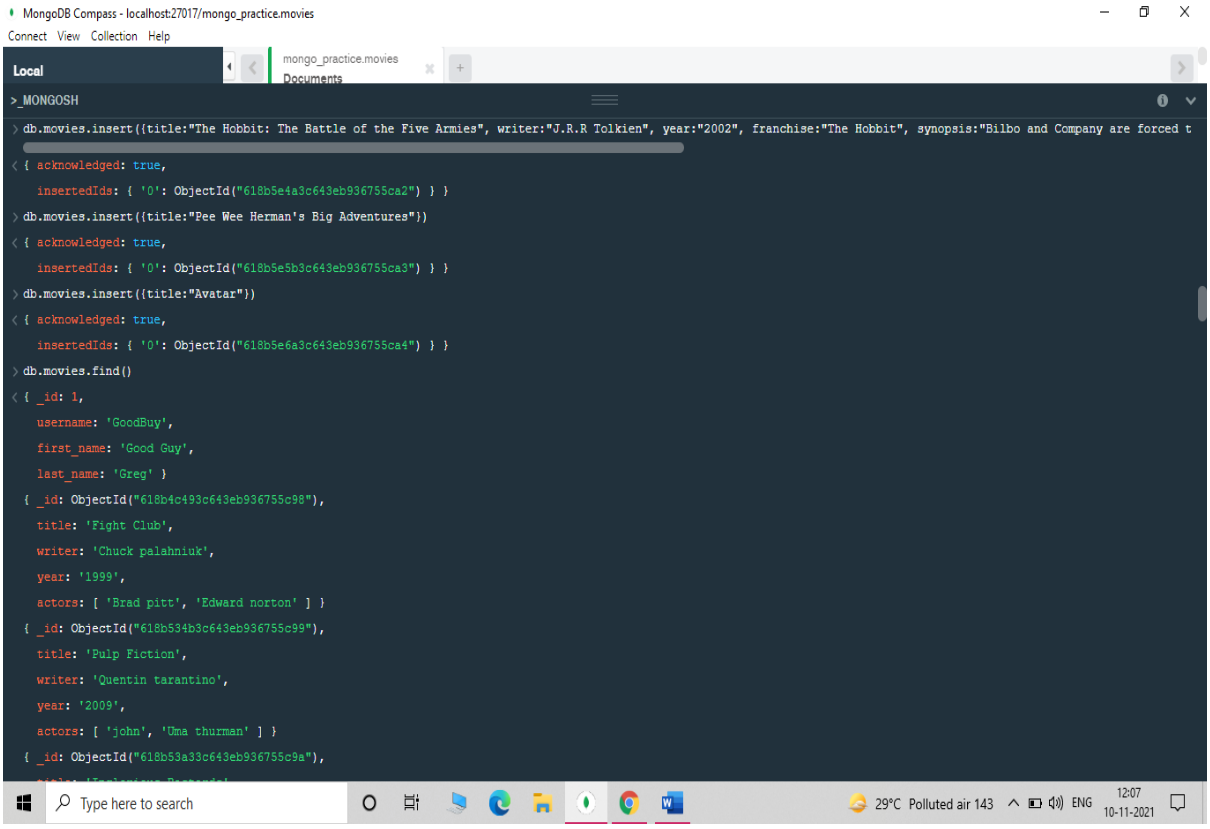
```
> db.movies.find()
< { _id: 1,
    username: 'GoodBuy',
    first_name: 'Good Guy',
    last_name: 'Greg' }
  { _id: ObjectId("618b4c493c643eb936755c98"),
    title: 'Fight Club',
    writer: 'Chuck palahniuk',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward norton' ] }
  { _id: ObjectId("618b534b3c643eb936755c99"),
    title: 'Pulp Fiction',
    writer: 'Quentin tarantino',
    year: '2009',
    actors: [ 'john', 'Uma thurman' ] }
  { _id: ObjectId("618b53a33c643eb936755c9a"),
    title: 'Inglorious Basterds',
    writer: 'Quentin tarantino',
    year: '2009',
    actors: [ 'john', 'Eli Roth' ] }
  { _id: ObjectId("618b540f3c643eb936755c9b"),
    title: 'The Hobbit:The desolation of smaug',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: [ 'The Hobbit' ] }
  { _id: ObjectId("618b543c3c643eb936755c9c"), title: 'Avatar' }
```



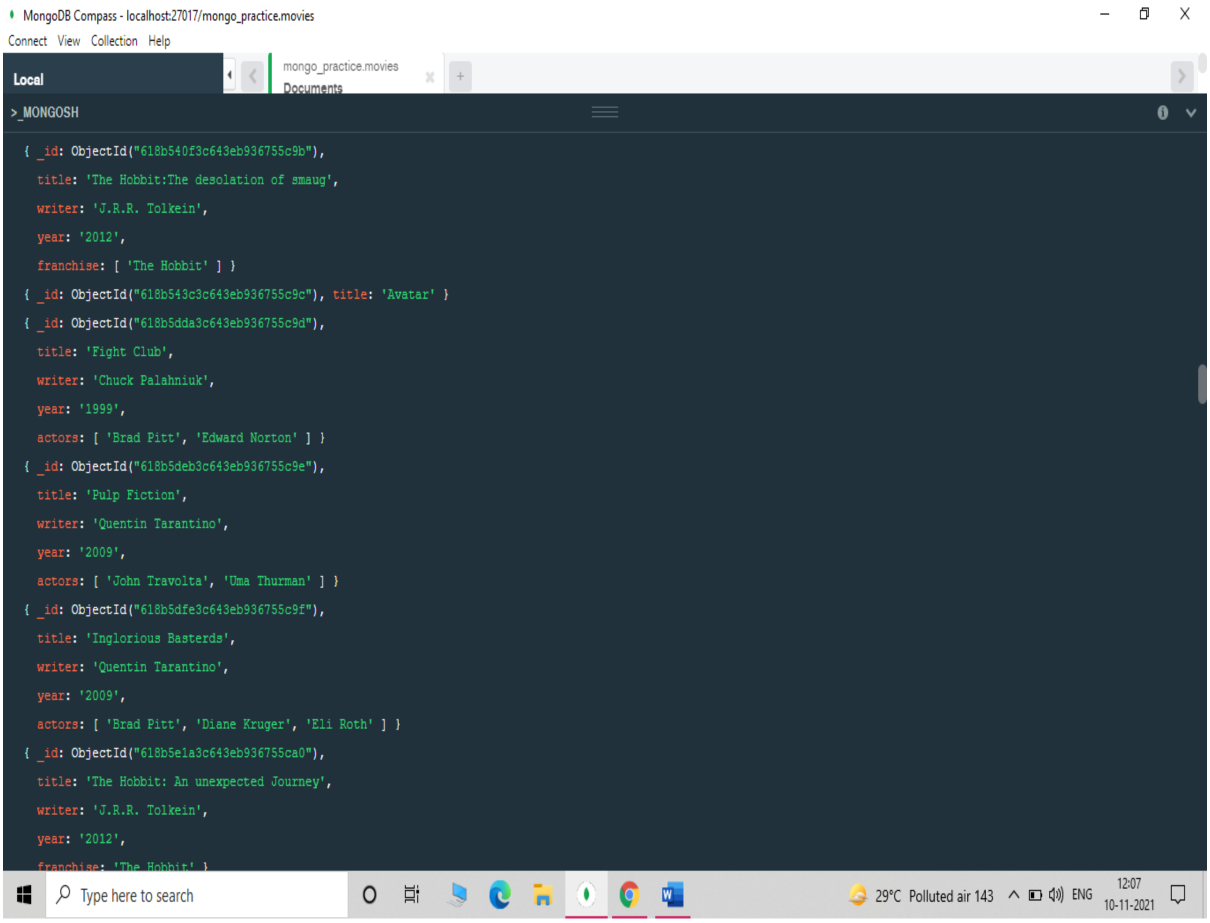
```
> db.movies.update({_id:ObjectId("5c9f98e5e5c2dfe9b3729bfe")}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of
< 'DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.'
< { acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0 }
> db.movies.update({_id:ObjectId("5c9fa42ae5c2dfe9b3729c03")}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim
< { acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0 }
> db.movies.update({_id:ObjectId("5c9f983ce5c2dfe9b3729bfc")}, {$push:{actors:"Samuel L. Jackson"}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0 }
> db.movies.find({synopsis:{$regex:"Bilbo"}})
<
> opsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }
✖ ▸ SyntaxError: Missing semicolon. (1:7)

  > 1 | { "_id" : ObjectId("618a9c05f3e2cf491e83ff18"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "year" : "2002", "franchise" :
```



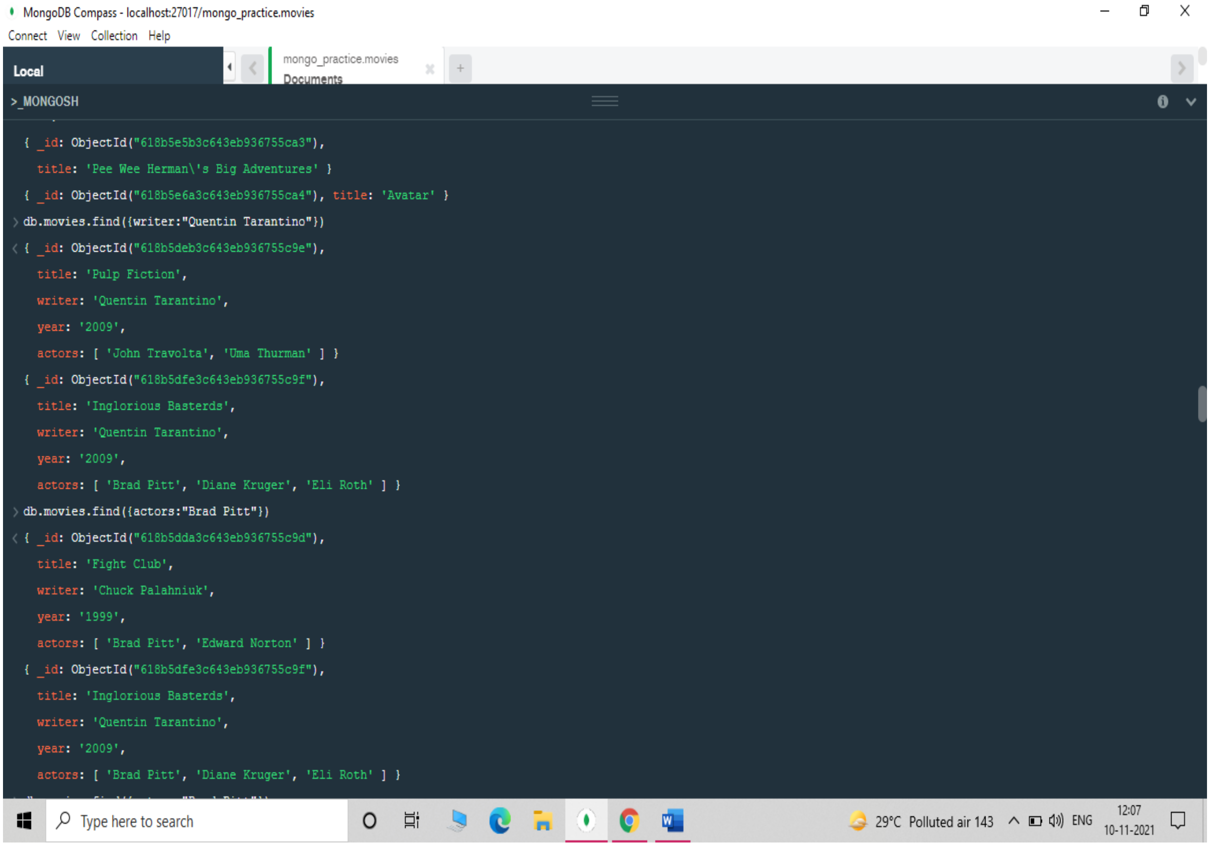
```
<
> db.movies.find({synopsis:{$regex:"Gandalf"}})
<
> db.movies.find({synopsis:{$regex:"Gandalf"}})
<
> db.movies.find({synopsis:{$regex:"Gandalf"}})
<
> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})
<
> db.movies.remove({_id:ObjectId("5c9f992ae5c2dfe9b3729c00")})
< 'DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.'
< { acknowledged: true, deletedCount: 0 }
> db.movies.remove({_id:ObjectId("5c9f9936e5c2dfe9b3729c01")})
< { acknowledged: true, deletedCount: 0 }
> db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})
< 'DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.'
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5dda3c643eb936755c9d") } }
> db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"2009", actors:["John Travolta", "Uma Thurman"]})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5deb3c643eb936755c9e") } }
> db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"2009", actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5dfe3c643eb936755c9f") } }
> db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkein", year:"2012",franchise:"The Hobbit"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5e1a3c643eb936755ca0") } }
> db.movies.insert({title:"The Hobbit: The Desolation of Smaug", writer:"J.R.R Tolkien", year:"2013", franchise:"The Hobbit"})
```



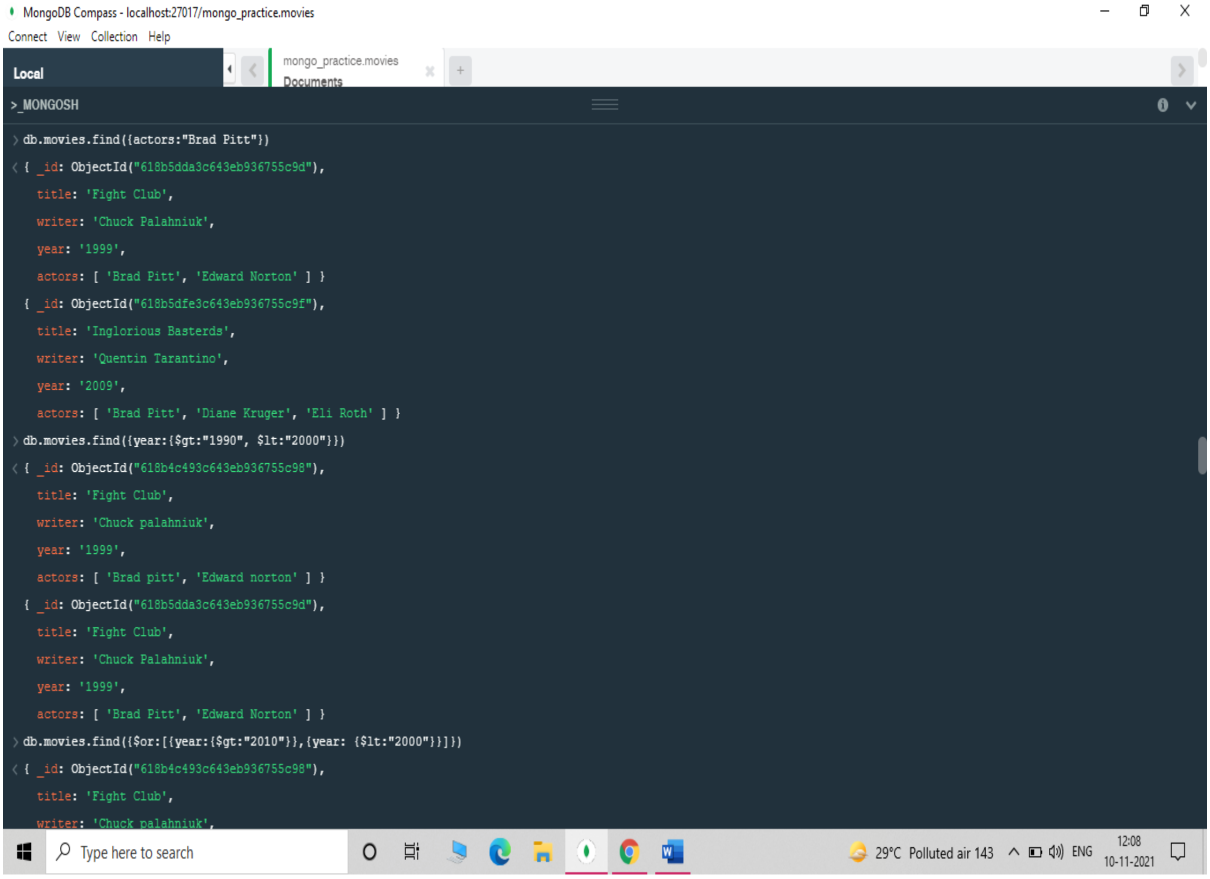
```
> db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", franchise:"The Hobbit", synopsis:"Bilbo and Company are forced t
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5e4a3c643eb936755ca2") } }
> db.movies.insert({title:"Pee Wee Herman's Big Adventures"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5e5b3c643eb936755ca3") } }
> db.movies.insert({title:"Avatar"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b5e6a3c643eb936755ca4") } }
> db.movies.find()
< { _id: 1,
    username: 'GoodBuy',
    first_name: 'Good Guy',
    last_name: 'Greg' }
  { _id: ObjectId("618b4c493c643eb936755c98"),
    title: 'Fight Club',
    writer: 'Chuck palahniuk',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward norton' ] }
  { _id: ObjectId("618b534b3c643eb936755c99"),
    title: 'Pulp Fiction',
    writer: 'Quentin tarantino',
    year: '2009',
    actors: [ 'john', 'Uma thurman' ] }
  { _id: ObjectId("618b53a33c643eb936755c9a"),
    title: 'Inglorious Basterds',
```



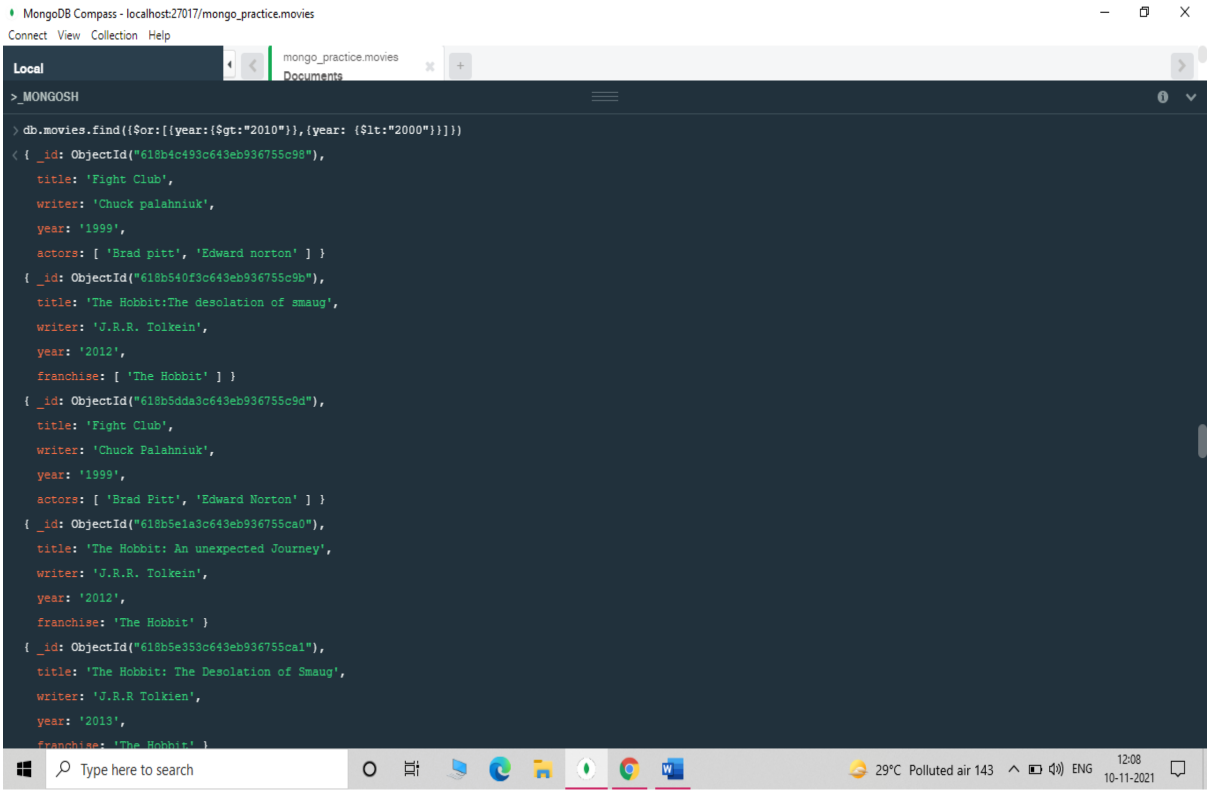
```
  { _id: ObjectId("618b540f3c643eb936755c9b"),
    title: 'The Hobbit:The desolation of smaug',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: [ 'The Hobbit' ] }
  { _id: ObjectId("618b543c3c643eb936755c9c"), title: 'Avatar' }
  { _id: ObjectId("618b5dda3c643eb936755c9d"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("618b5deb3c643eb936755c9e"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: [ 'John Travolta', 'Uma Thurman' ] }
  { _id: ObjectId("618b5dfe3c643eb936755c9f"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
  { _id: ObjectId("618b5e1a3c643eb936755ca0"),
    title: 'The Hobbit: An unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit' }
```



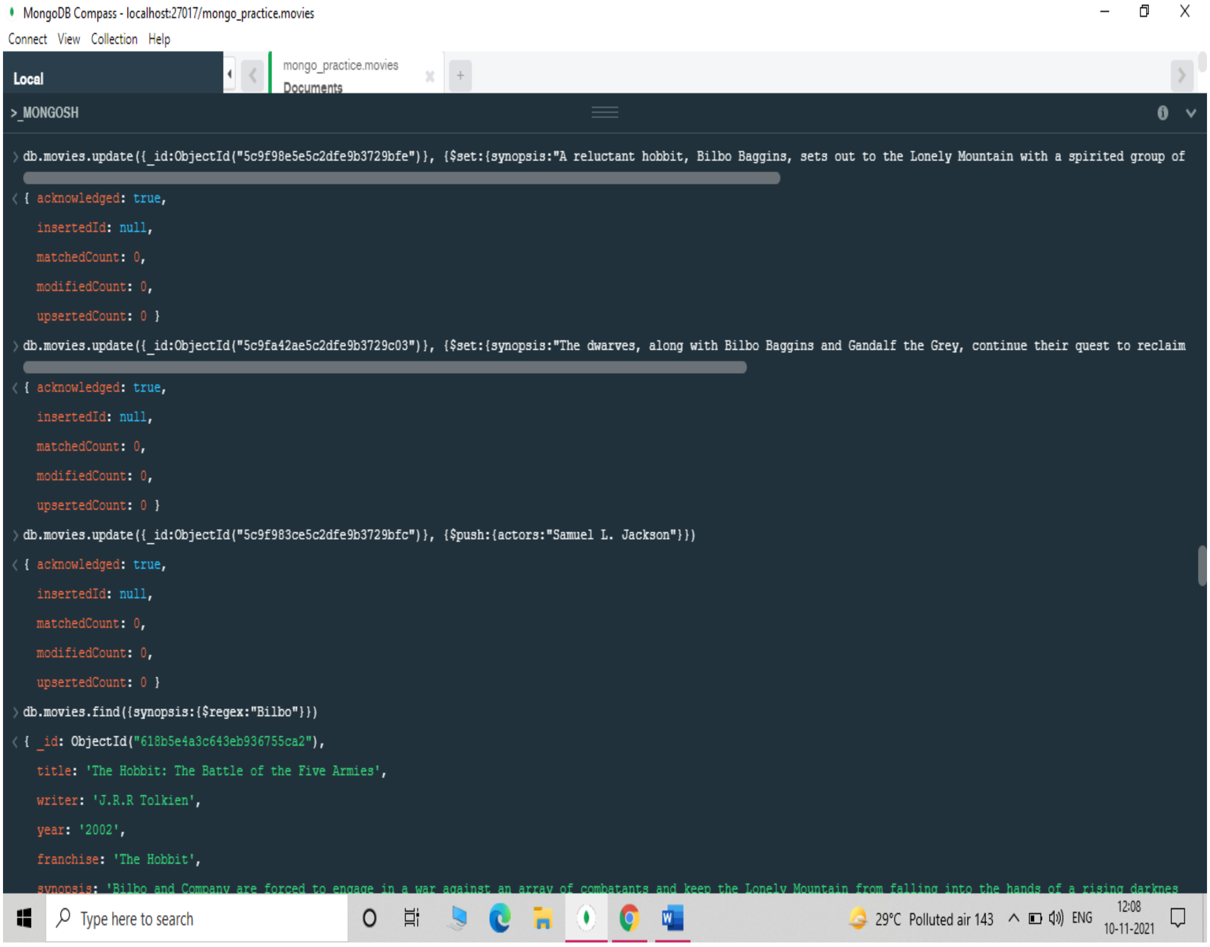
```
  { _id: ObjectId("618b5e5b3c643eb936755ca3"),
    title: 'Pee Wee Herman\'s Big Adventures' }
  { _id: ObjectId("618b5e6a3c643eb936755ca4"), title: 'Avatar' }
> db.movies.find({writer:"Quentin Tarantino"})
< { _id: ObjectId("618b5deb3c643eb936755c9e"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: [ 'John Travolta', 'Uma Thurman' ] }
  { _id: ObjectId("618b5dfe3c643eb936755c9f"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
> db.movies.find({actors:"Brad Pitt"})
< { _id: ObjectId("618b5dda3c643eb936755c9d"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("618b5dfe3c643eb936755c9f"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
```



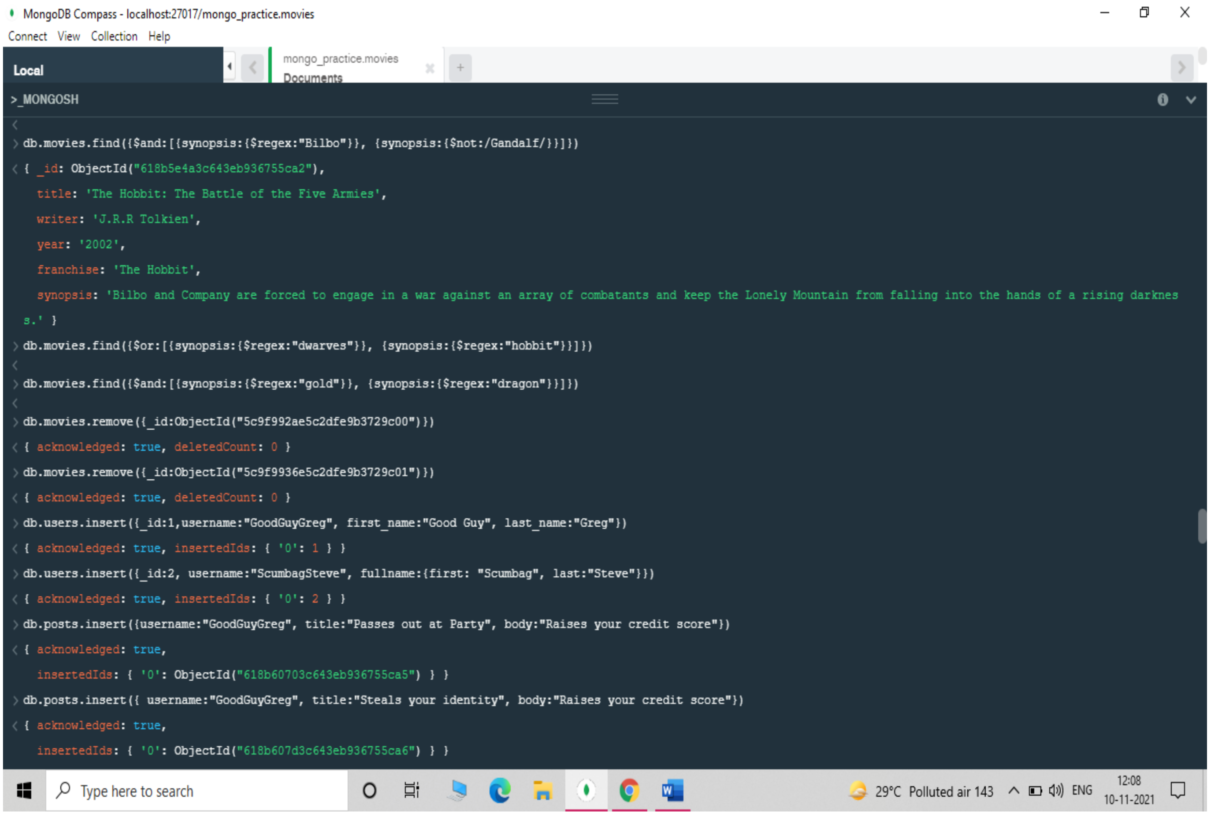
```
> db.movies.find({actors:"Brad Pitt"})
< { _id: ObjectId("618b5dda3c643eb936755c9d"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("618b5dfe3c643eb936755c9f"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
> db.movies.find({year:{$gt:"1990", $lt:"2000"}})
< { _id: ObjectId("618b4c493c643eb936755c98"),
    title: 'Fight Club',
    writer: 'Chuck palahniuk',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward norton' ] }
  { _id: ObjectId("618b5dda3c643eb936755c9d"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]})
< { _id: ObjectId("618b4c493c643eb936755c98"),
    title: 'Fight Club',
    writer: 'Chuck palahniuk',
```



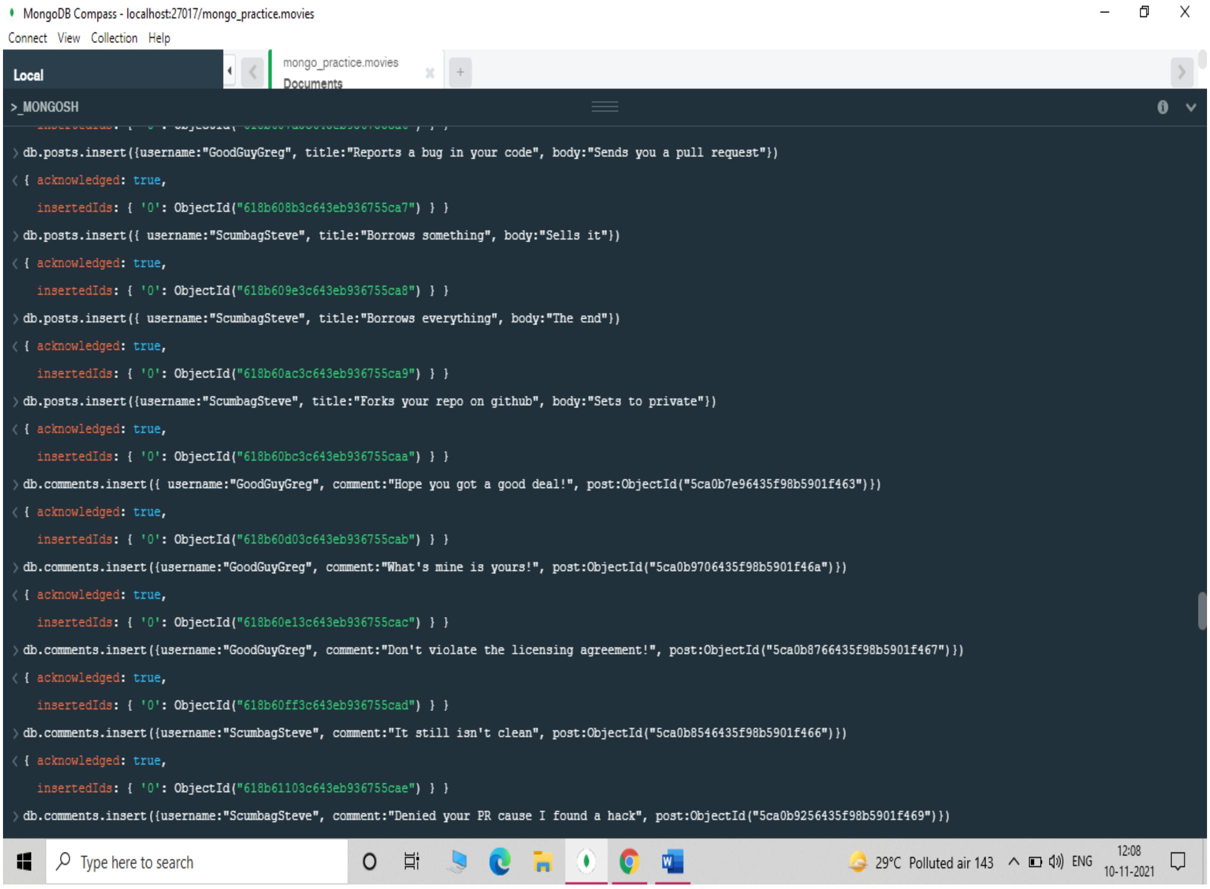
```
> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]})
< { _id: ObjectId("618b4c493c643eb936755c98"),
    title: 'Fight Club',
    writer: 'Chuck palahniuk',
    year: '1999',
    actors: [ 'Brad pitt', 'Edward norton' ] }
  { _id: ObjectId("618b540f3c643eb936755c9b"),
    title: 'The Hobbit:The desolation of smaug',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: [ 'The Hobbit' ] }
  { _id: ObjectId("618b5dda3c643eb936755c9d"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ 'Brad Pitt', 'Edward Norton' ] }
  { _id: ObjectId("618b5e1a3c643eb936755ca0"),
    title: 'The Hobbit: An unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit' }
  { _id: ObjectId("618b5e353c643eb936755ca1"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R Tolkien',
    year: '2013',
    franchise: 'The Hobbit' }
```



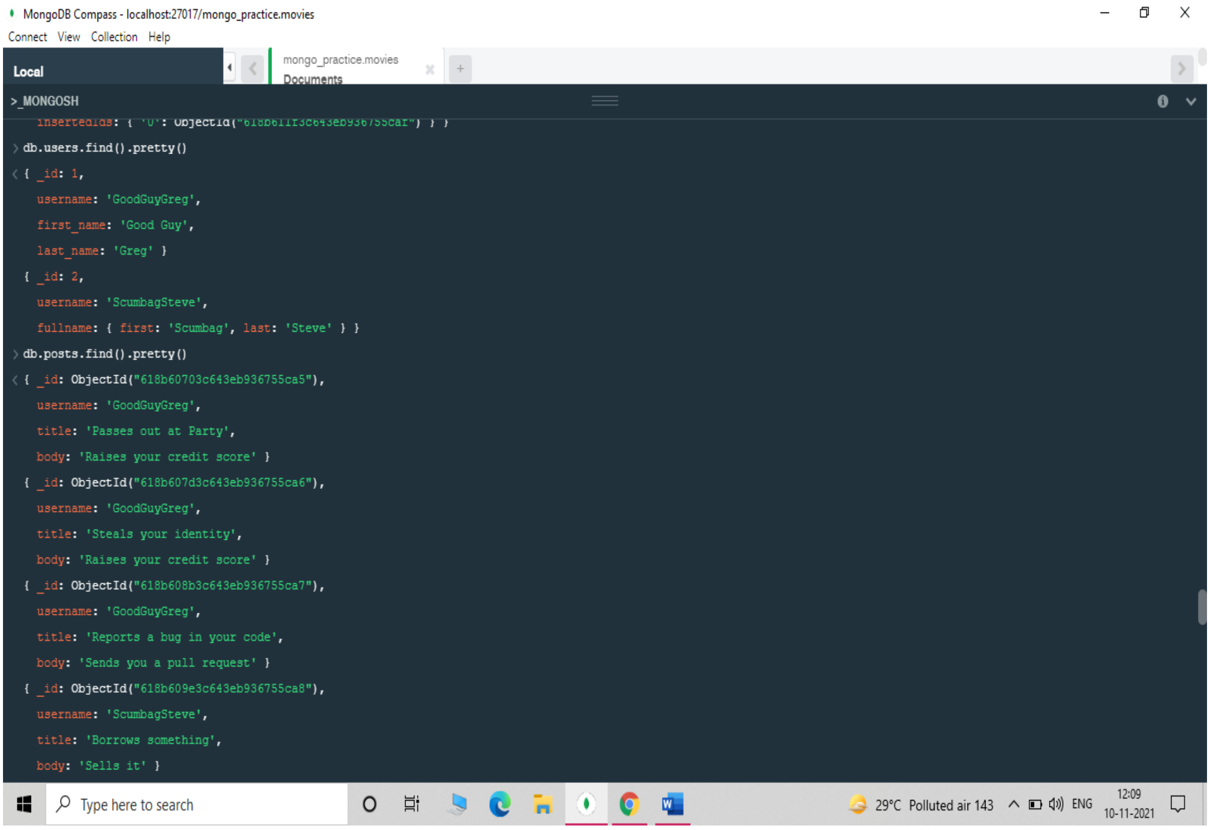
```
> db.movies.update({_id:ObjectId("5c9f98e5e5c2dfe9b3729bfe")}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of
< { acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0 }
> db.movies.update({_id:ObjectId("5c9fa42ae5c2dfe9b3729c03")}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim
< { acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0 }
> db.movies.update({_id:ObjectId("5c9f983ce5c2dfe9b3729bfc")}, {$push:{actors:"Samuel L. Jackson"}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0 }
> db.movies.find({synopsis:{$regex:"Bilbo"}})
< { _id: ObjectId("618b5e4a3c643eb936755ca2"),
    title: 'The Hobbit: The Battle of the Five Armies',
    writer: 'J.R.R Tolkien',
    year: '2002',
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darknes
```



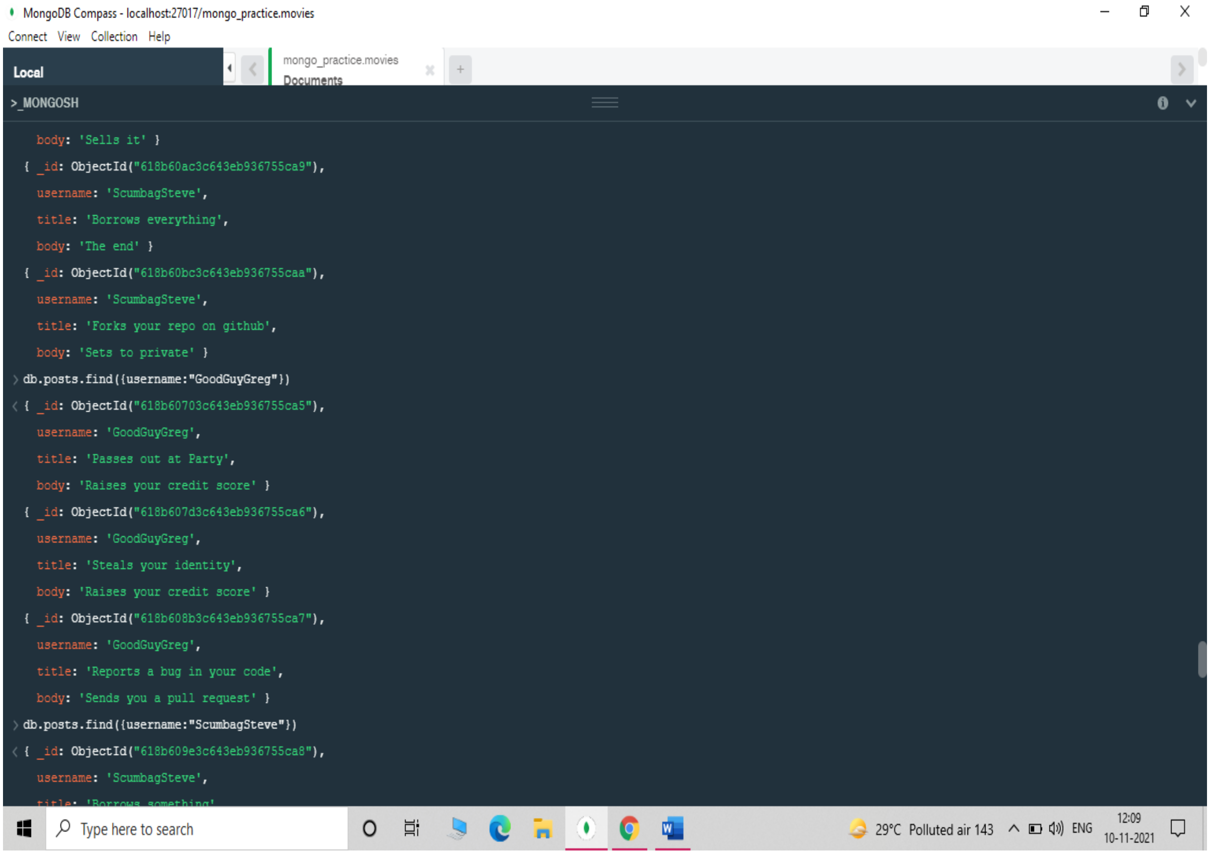
```
<
> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]})
< { _id: ObjectId("618b5e4a3c643eb936755ca2"),
    title: 'The Hobbit: The Battle of the Five Armies',
    writer: 'J.R.R Tolkien',
    year: '2002',
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darknes
  s.' }
> db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]})
<
> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})
<
> db.movies.remove({_id:ObjectId("5c9f992ae5c2dfe9b3729c00")})
< { acknowledged: true, deletedCount: 0 }
> db.movies.remove({_id:ObjectId("5c9f9936e5c2dfe9b3729c01")})
< { acknowledged: true, deletedCount: 0 }
> db.users.insert({_id:1,username:"GoodGuyGreg", first_name:"Good Guy", last_name:"Greg"})
< { acknowledged: true, insertedIds: { '0': 1 } }
> db.users.insert({_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})
< { acknowledged: true, insertedIds: { '0': 2 } }
> db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b60703c643eb936755ca5") } }
> db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b607d3c643eb936755ca6") } }
```



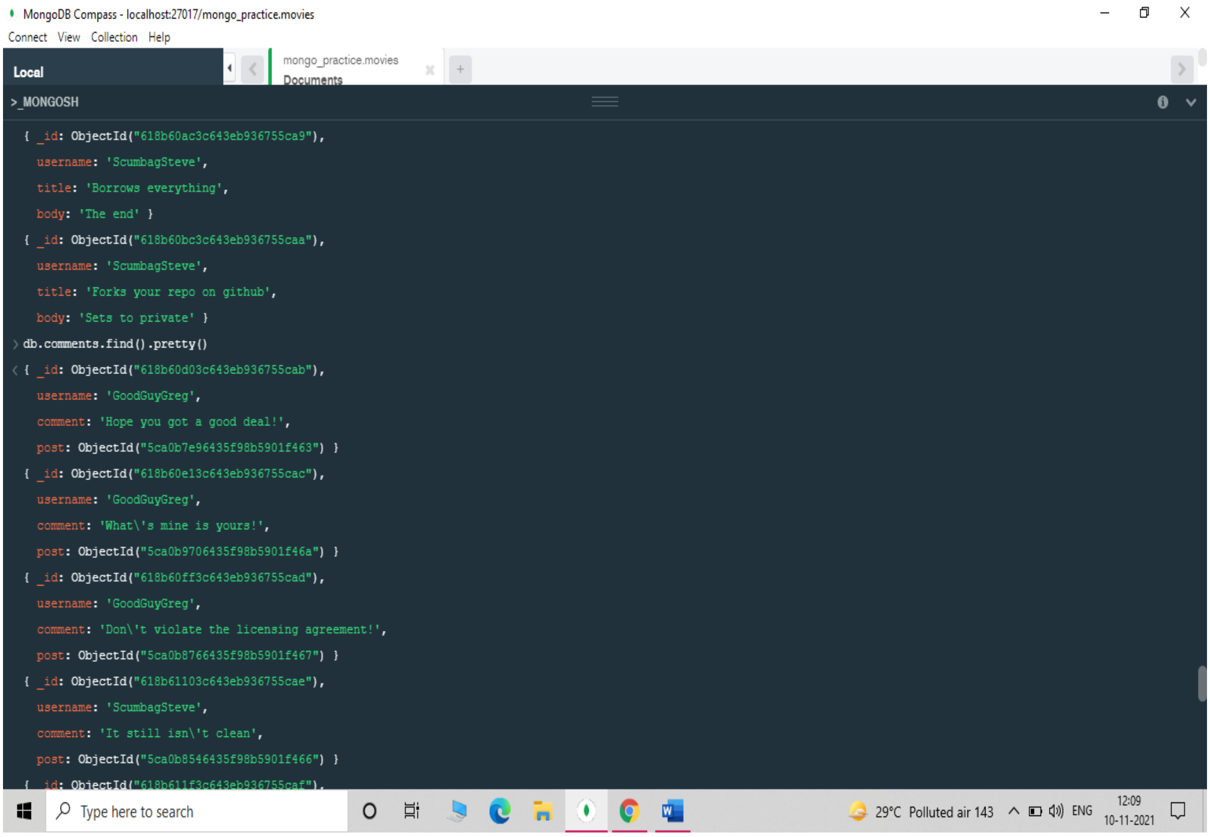
```
> db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b608b3c643eb936755ca7") } }
> db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b609e3c643eb936755ca8") } }
> db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b60ac3c643eb936755ca9") } }
> db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b60bc3c643eb936755caa") } }
> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("5ca0b7e96435f98b5901f463")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b60d03c643eb936755cab") } }
> db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("5ca0b9706435f98b5901f46a")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b60e13c643eb936755cac") } }
> db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("5ca0b8766435f98b5901f467")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b60ff3c643eb936755cad") } }
> db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("5ca0b8546435f98b5901f466")})
< { acknowledged: true,
    insertedIds: { '0': ObjectId("618b61103c643eb936755cae") } }
> db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("5ca0b9256435f98b5901f469")})
```



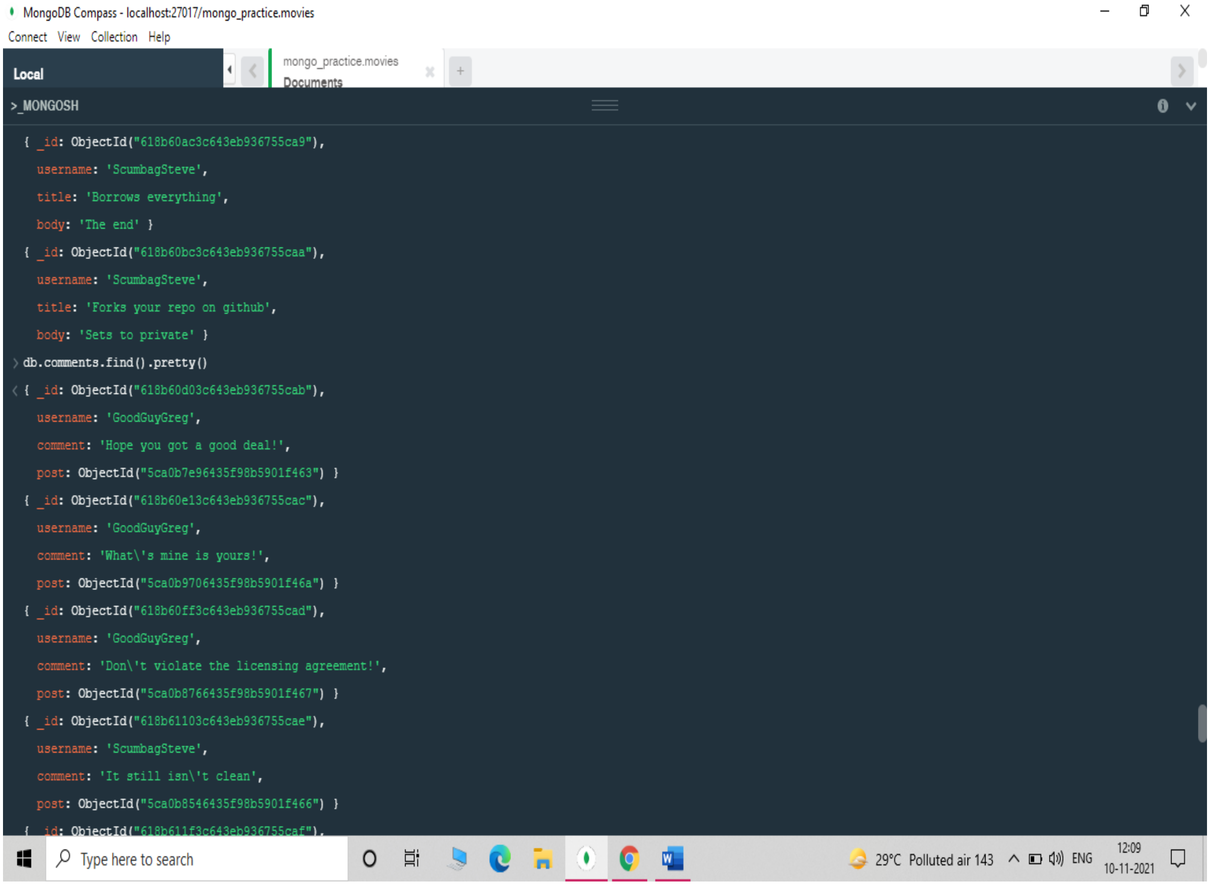
```
    insertedIds: { '0': ObjectId("618b611f3c643eb936755caf") } }
> db.users.find().pretty()
< { _id: 1,
    username: 'GoodGuyGreg',
    first_name: 'Good Guy',
    last_name: 'Greg' }
  { _id: 2,
    username: 'ScumbagSteve',
    fullname: { first: 'Scumbag', last: 'Steve' } }
> db.posts.find().pretty()
< { _id: ObjectId("618b60703c643eb936755ca5"),
    username: 'GoodGuyGreg',
    title: 'Passes out at Party',
    body: 'Raises your credit score' }
  { _id: ObjectId("618b607d3c643eb936755ca6"),
    username: 'GoodGuyGreg',
    title: 'Steals your identity',
    body: 'Raises your credit score' }
  { _id: ObjectId("618b608b3c643eb936755ca7"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request' }
  { _id: ObjectId("618b609e3c643eb936755ca8"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it' }
```



```
    body: 'Sells it' }
  { _id: ObjectId("618b60ac3c643eb936755ca9"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end' }
  { _id: ObjectId("618b60bc3c643eb936755caa"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private' }
> db.posts.find({username:"GoodGuyGreg"})
< { _id: ObjectId("618b60703c643eb936755ca5"),
    username: 'GoodGuyGreg',
    title: 'Passes out at Party',
    body: 'Raises your credit score' }
  { _id: ObjectId("618b607d3c643eb936755ca6"),
    username: 'GoodGuyGreg',
    title: 'Steals your identity',
    body: 'Raises your credit score' }
  { _id: ObjectId("618b608b3c643eb936755ca7"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request' }
> db.posts.find({username:"ScumbagSteve"})
< { _id: ObjectId("618b609e3c643eb936755ca8"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
```

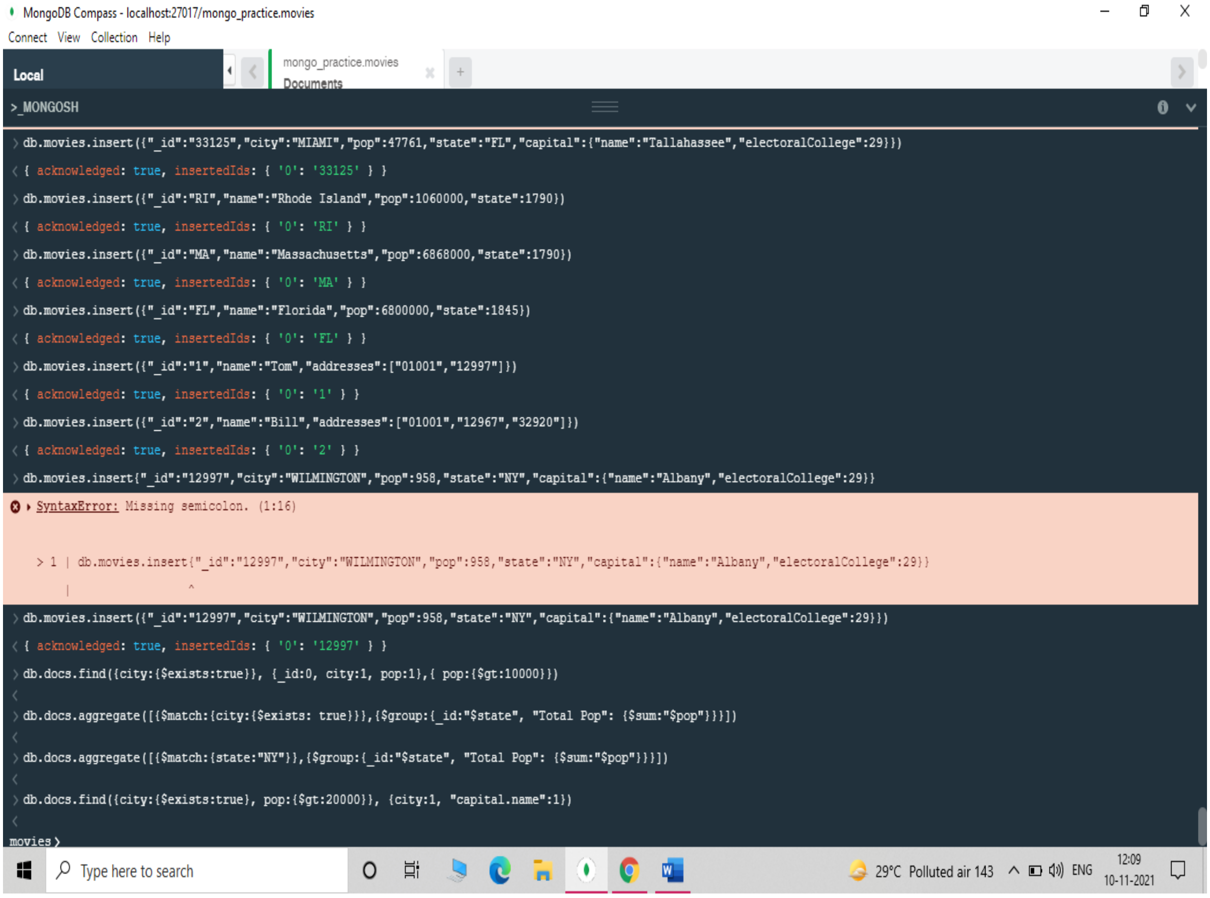


```
  { _id: ObjectId("618b60ac3c643eb936755ca9"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end' }
  { _id: ObjectId("618b60bc3c643eb936755caa"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private' }
> db.comments.find().pretty()
< { _id: ObjectId("618b60d03c643eb936755cab"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("5ca0b7e96435f98b5901f463") }
  { _id: ObjectId("618b60e13c643eb936755cac"),
    username: 'GoodGuyGreg',
    comment: 'What\'s mine is yours!',
    post: ObjectId("5ca0b9706435f98b5901f46a") }
  { _id: ObjectId("618b60ff3c643eb936755cad"),
    username: 'GoodGuyGreg',
    comment: 'Don\'t violate the licensing agreement!',
    post: ObjectId("5ca0b8766435f98b5901f467") }
  { _id: ObjectId("618b61103c643eb936755cae"),
    username: 'ScumbagSteve',
    comment: 'It still isn\'t clean',
    post: ObjectId("5ca0b8546435f98b5901f466") }
  { _id: ObjectId("618b611f3c643eb936755caf"),
```

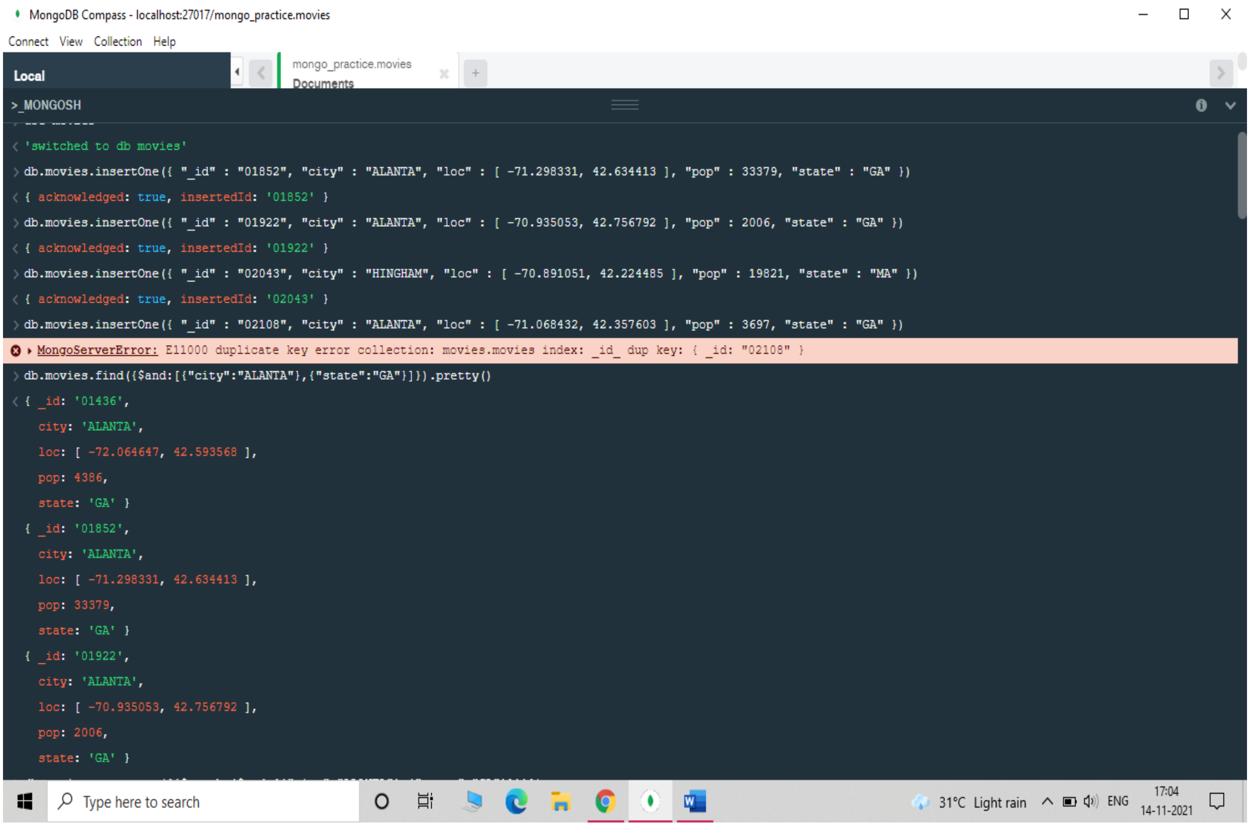


MongoDB Compass - localhost:27017/mongo_practice.movies

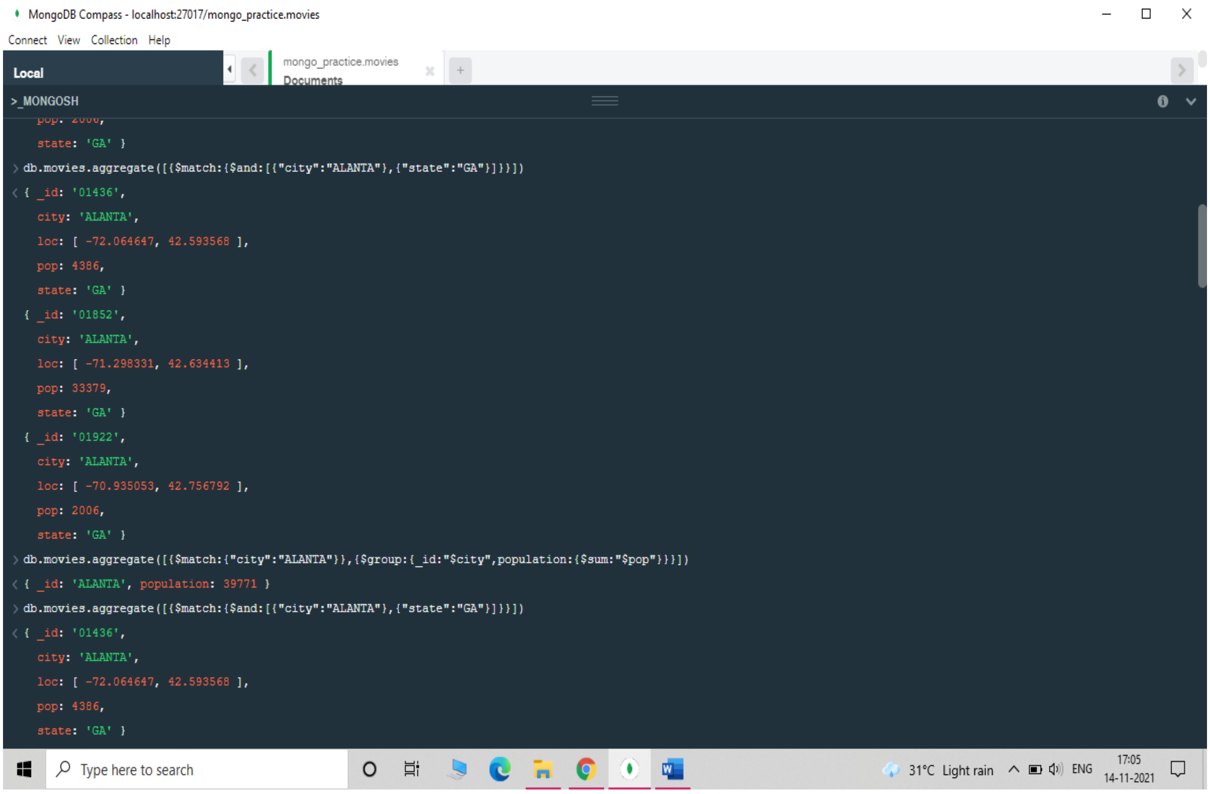
```
  { _id: ObjectId("618b60ac3c643eb936755ca9"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end' }
  { _id: ObjectId("618b60bc3c643eb936755caa"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private' }
> db.comments.find().pretty()
< { _id: ObjectId("618b60d03c643eb936755cab"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("5ca0b7e96435f98b5901f463") }
  { _id: ObjectId("618b60e13c643eb936755cac"),
    username: 'GoodGuyGreg',
    comment: 'What\'s mine is yours!',
    post: ObjectId("5ca0b9706435f98b5901f46a") }
  { _id: ObjectId("618b60ff3c643eb936755cad"),
    username: 'GoodGuyGreg',
    comment: 'Don\'t violate the licensing agreement!',
    post: ObjectId("5ca0b8766435f98b5901f467") }
  { _id: ObjectId("618b61103c643eb936755cae"),
    username: 'ScumbagSteve',
    comment: 'It still isn\'t clean',
    post: ObjectId("5ca0b8546435f98b5901f466") }
  { _id: ObjectId("618b611f3c643eb936755caf"),
```



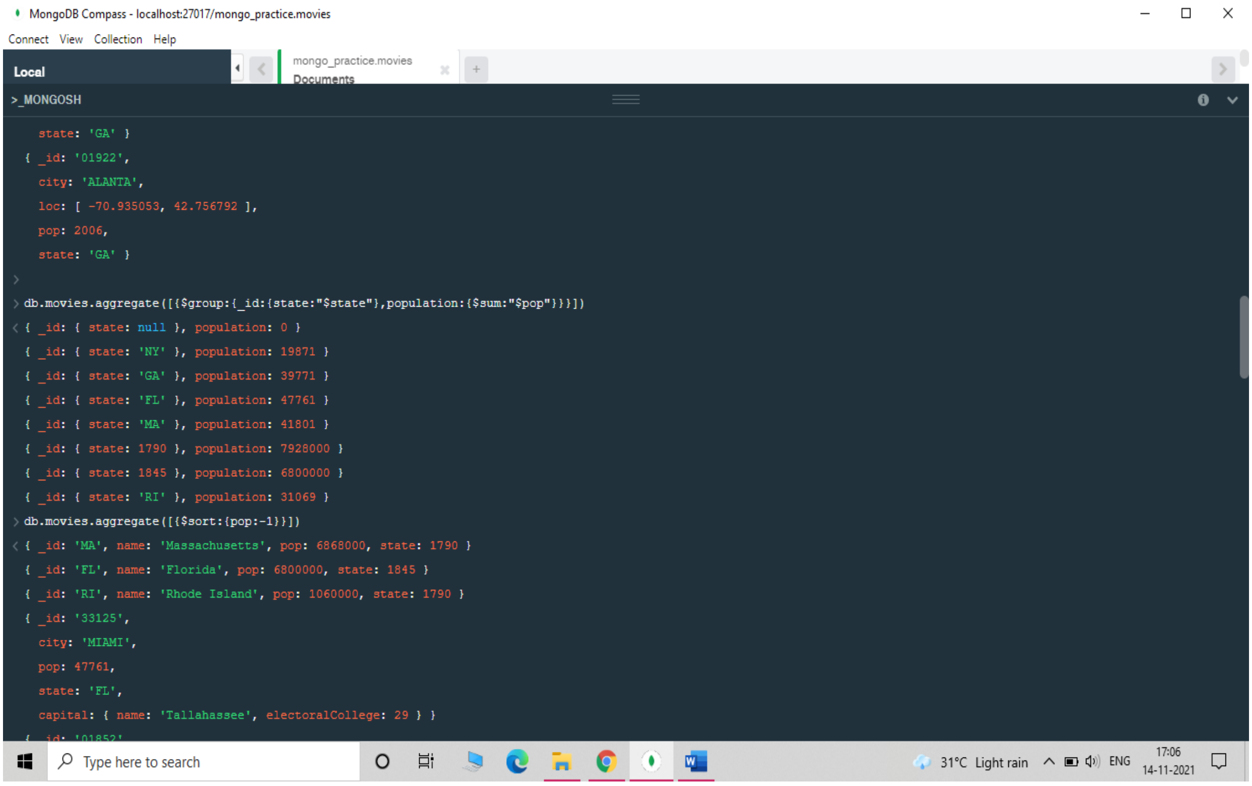
```
> db.movies.insert({"_id":"33125","city":"MIAMI","pop":47761,"state":"FL","capital":{"name":"Tallahassee","electoralCollege":29}})
< { acknowledged: true, insertedIds: { '0': '33125' } }
> db.movies.insert({"_id":"RI","name":"Rhode Island","pop":1060000,"state":1790})
< { acknowledged: true, insertedIds: { '0': 'RI' } }
> db.movies.insert({"_id":"MA","name":"Massachusetts","pop":6868000,"state":1790})
< { acknowledged: true, insertedIds: { '0': 'MA' } }
> db.movies.insert({"_id":"FL","name":"Florida","pop":6800000,"state":1845})
< { acknowledged: true, insertedIds: { '0': 'FL' } }
> db.movies.insert({"_id":"1","name":"Tom","addresses":["01001","12997"]})
< { acknowledged: true, insertedIds: { '0': '1' } }
> db.movies.insert({"_id":"2","name":"Bill","addresses":["01001","12967","32920"]})
< { acknowledged: true, insertedIds: { '0': '2' } }
> db.movies.insert{"_id":"12997","city":"WILMINGTON","pop":958,"state":"NY","capital":{"name":"Albany","electoralCollege":29}}
⊗ ▸ SyntaxError: Missing semicolon. (1:16)

    > 1 | db.movies.insert{"_id":"12997","city":"WILMINGTON","pop":958,"state":"NY","capital":{"name":"Albany","electoralCollege":29}}
        |                 ^
> db.movies.insert({"_id":"12997","city":"WILMINGTON","pop":958,"state":"NY","capital":{"name":"Albany","electoralCollege":29}})
< { acknowledged: true, insertedIds: { '0': '12997' } }
> db.docs.find({city:{$exists:true}}, {_id:0, city:1, pop:1},{ pop:{$gt:10000}})
<
> db.docs.aggregate([{$match:{city:{$exists: true}}},{$group:{_id:"$state", "Total Pop": {$sum:"$pop"}}}])
<
> db.docs.aggregate([{$match:{state:"NY"}},{$group:{_id:"$state", "Total Pop": {$sum:"$pop"}}}])
<
> db.docs.find({city:{$exists:true}, pop:{$gt:20000}}, {city:1, "capital.name":1})
<
movies >
```



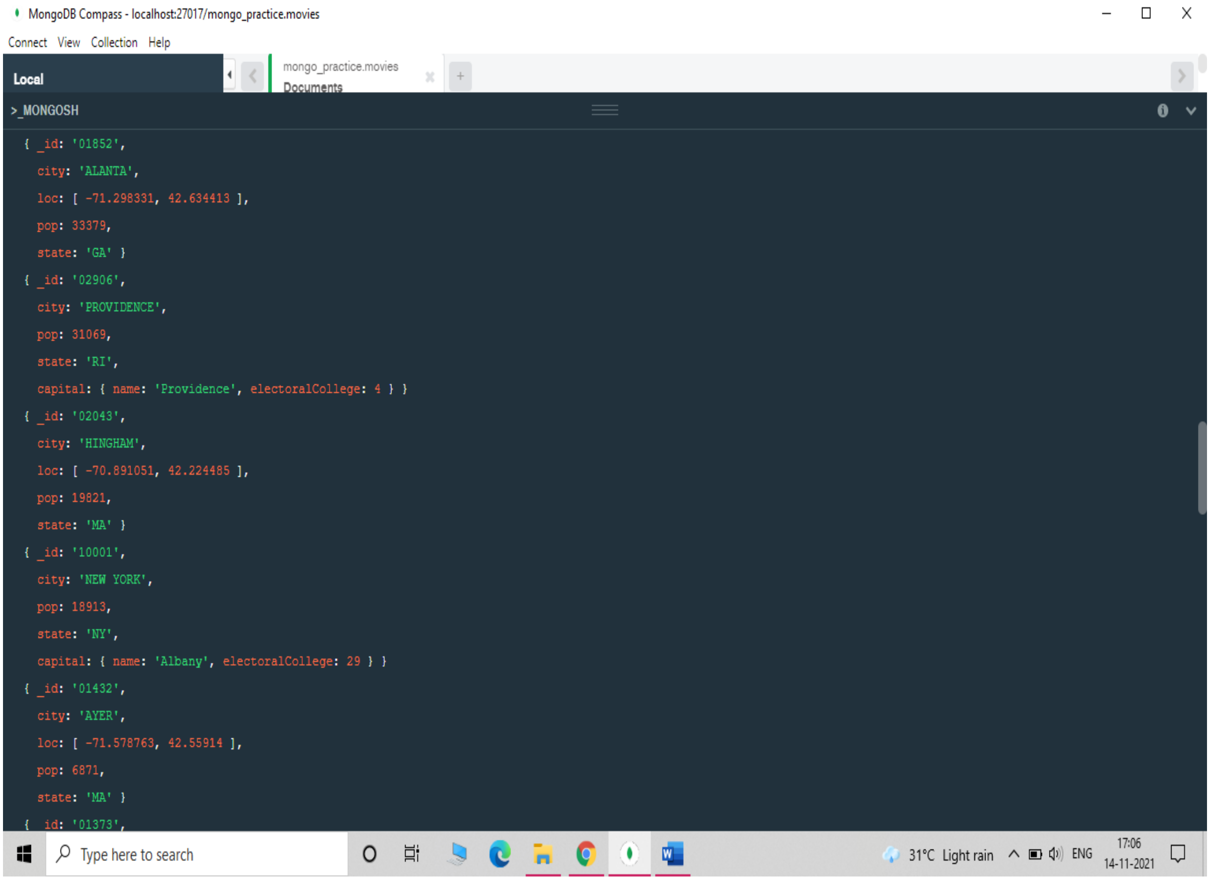
```
< 'switched to db movies'
> db.movies.insertOne({ "_id" : "01852", "city" : "ALANTA", "loc" : [ -71.298331, 42.634413 ], "pop" : 33379, "state" : "GA" })
< { acknowledged: true, insertedId: '01852' }
> db.movies.insertOne({ "_id" : "01922", "city" : "ALANTA", "loc" : [ -70.935053, 42.756792 ], "pop" : 2006, "state" : "GA" })
< { acknowledged: true, insertedId: '01922' }
> db.movies.insertOne({ "_id" : "02043", "city" : "HINGHAM", "loc" : [ -70.891051, 42.224485 ], "pop" : 19821, "state" : "MA" })
< { acknowledged: true, insertedId: '02043' }
> db.movies.insertOne({ "_id" : "02108", "city" : "ALANTA", "loc" : [ -71.068432, 42.357603 ], "pop" : 3697, "state" : "GA" })
⊗ ▸ MongoServerError: E11000 duplicate key error collection: movies.movies index: _id_ dup key: { _id: "02108" }
> db.movies.find({$and:[{"city":"ALANTA"},{"state":"GA"}]}).pretty()
< { _id: '01436',
    city: 'ALANTA',
    loc: [ -72.064647, 42.593568 ],
    pop: 4386,
    state: 'GA' }
  { _id: '01852',
    city: 'ALANTA',
    loc: [ -71.298331, 42.634413 ],
    pop: 33379,
    state: 'GA' }
  { _id: '01922',
    city: 'ALANTA',
    loc: [ -70.935053, 42.756792 ],
    pop: 2006,
    state: 'GA' }
```



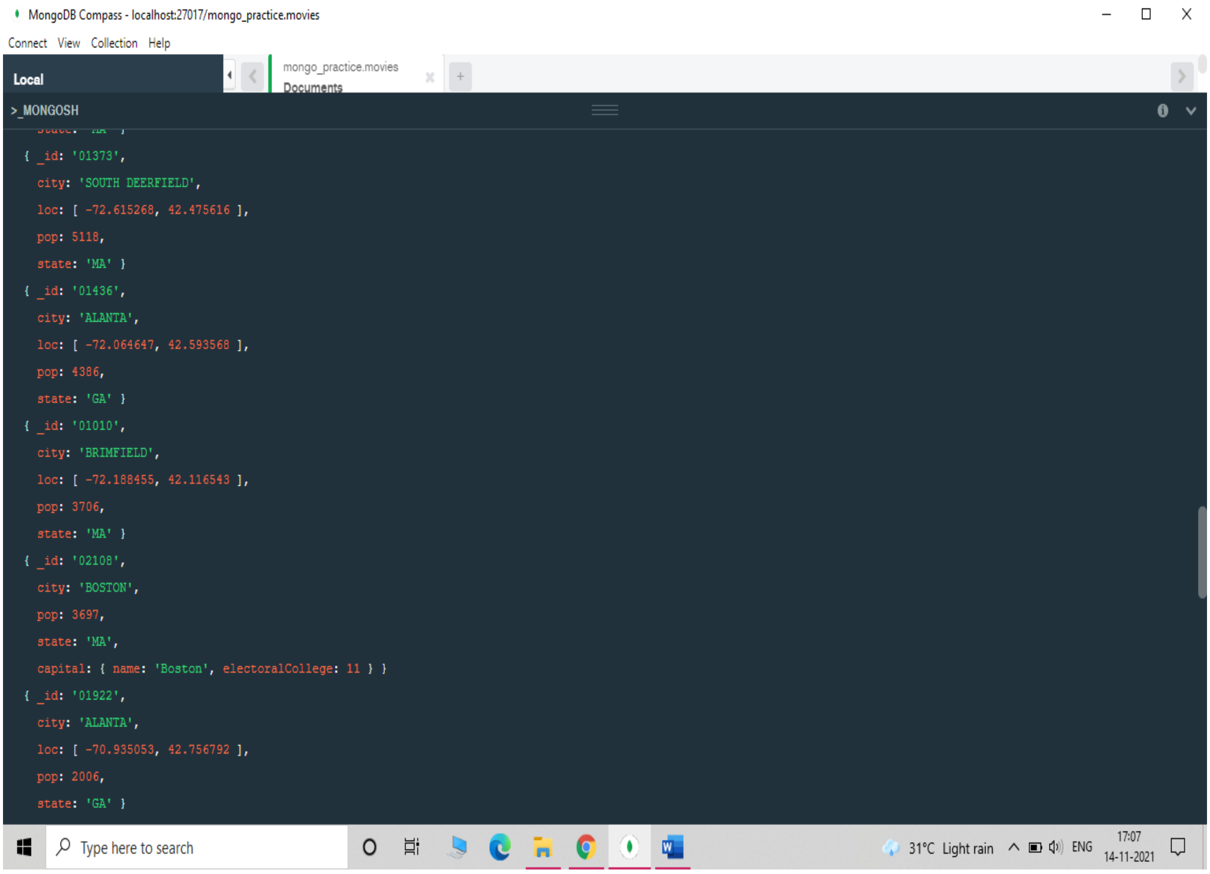
```
    state: 'GA' }
> db.movies.aggregate([{$match:{$and:[{"city":"ALANTA"},{"state":"GA"}]}}])
< { _id: '01436',
    city: 'ALANTA',
    loc: [ -72.064647, 42.593568 ],
    pop: 4386,
    state: 'GA' }
  { _id: '01852',
    city: 'ALANTA',
    loc: [ -71.298331, 42.634413 ],
    pop: 33379,
    state: 'GA' }
  { _id: '01922',
    city: 'ALANTA',
    loc: [ -70.935053, 42.756792 ],
    pop: 2006,
    state: 'GA' }
> db.movies.aggregate([{$match:{"city":"ALANTA"}},{$group:{_id:"$city",population:{$sum:"$pop"}}}])
< { _id: 'ALANTA', population: 39771 }
> db.movies.aggregate([{$match:{$and:[{"city":"ALANTA"},{"state":"GA"}]}}])
< { _id: '01436',
    city: 'ALANTA',
    loc: [ -72.064647, 42.593568 ],
    pop: 4386,
    state: 'GA' }
```



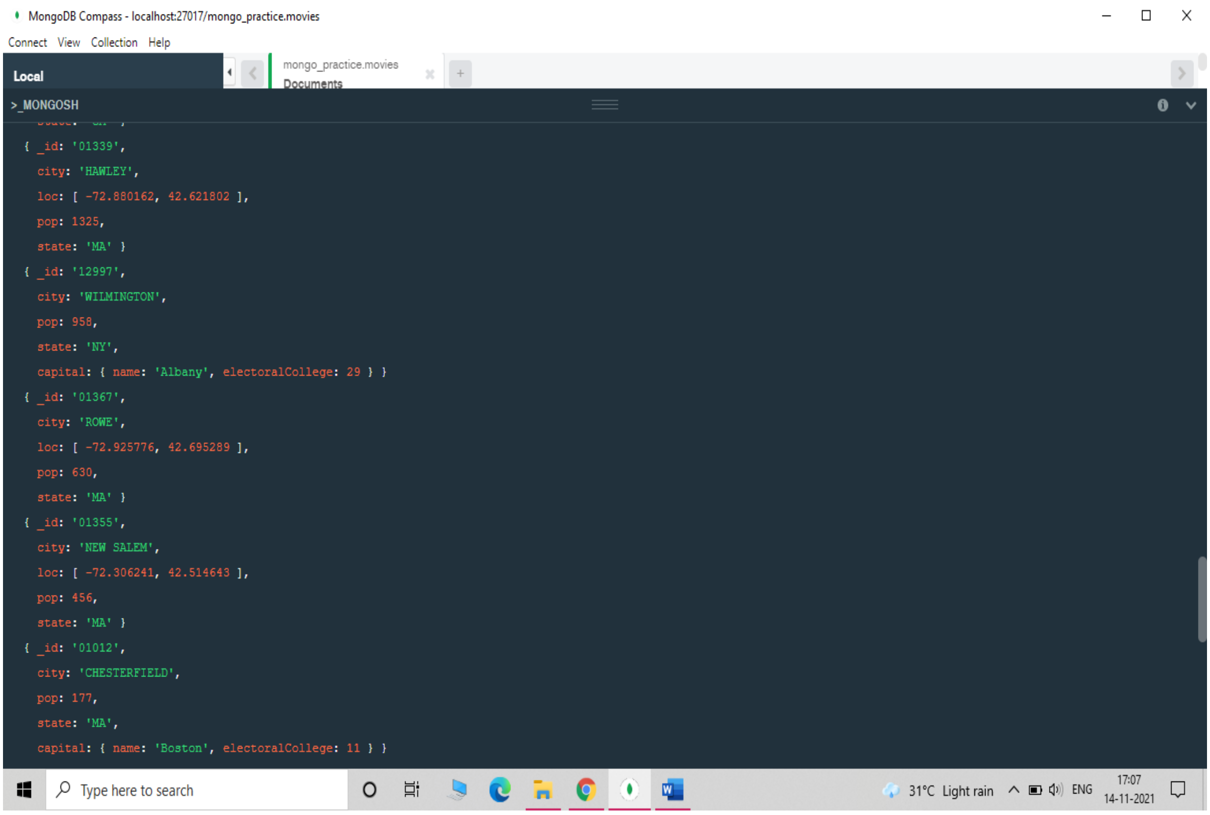
```
    state: 'GA' }
  { _id: '01922',
    city: 'ALANTA',
    loc: [ -70.935053, 42.756792 ],
    pop: 2006,
    state: 'GA' }
>
> db.movies.aggregate([{$group:{_id:{state:"$state"},population:{$sum:"$pop"}}}])
< { _id: { state: null }, population: 0 }
  { _id: { state: 'NY' }, population: 19871 }
  { _id: { state: 'GA' }, population: 39771 }
  { _id: { state: 'FL' }, population: 47761 }
  { _id: { state: 'MA' }, population: 41801 }
  { _id: { state: 1790 }, population: 7928000 }
  { _id: { state: 1845 }, population: 6800000 }
  { _id: { state: 'RI' }, population: 31069 }
> db.movies.aggregate([{$sort:{pop:-1}}])
< { _id: 'MA', name: 'Massachusetts', pop: 6868000, state: 1790 }
  { _id: 'FL', name: 'Florida', pop: 6800000, state: 1845 }
  { _id: 'RI', name: 'Rhode Island', pop: 1060000, state: 1790 }
  { _id: '33125',
    city: 'MIAMI',
    pop: 47761,
    state: 'FL',
    capital: { name: 'Tallahassee', electoralCollege: 29 } }
  { _id: '01852',
```



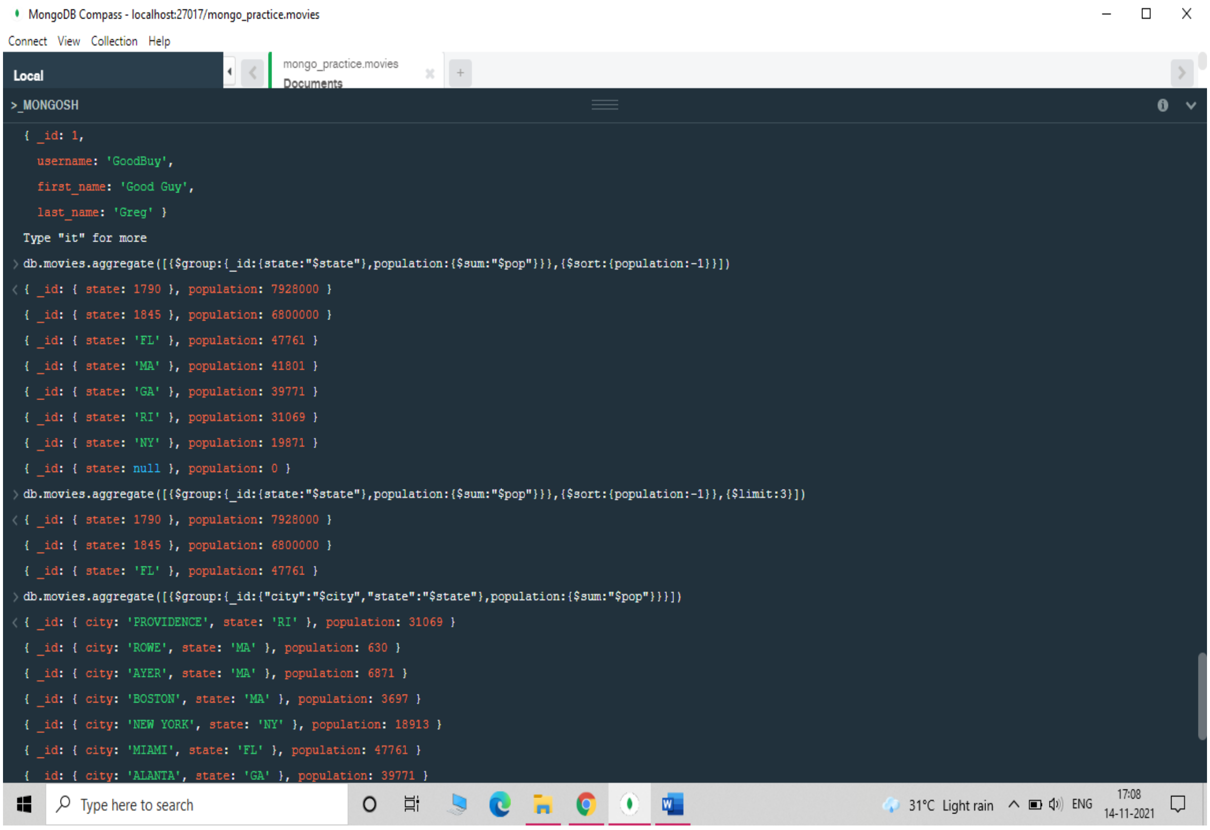
```
  { _id: '01852',
    city: 'ALANTA',
    loc: [ -71.298331, 42.634413 ],
    pop: 33379,
    state: 'GA' }
  { _id: '02906',
    city: 'PROVIDENCE',
    pop: 31069,
    state: 'RI',
    capital: { name: 'Providence', electoralCollege: 4 } }
  { _id: '02043',
    city: 'HINGHAM',
    loc: [ -70.891051, 42.224485 ],
    pop: 19821,
    state: 'MA' }
  { _id: '10001',
    city: 'NEW YORK',
    pop: 18913,
    state: 'NY',
    capital: { name: 'Albany', electoralCollege: 29 } }
  { _id: '01432',
    city: 'AYER',
    loc: [ -71.578763, 42.55914 ],
    pop: 6871,
    state: 'MA' }
  { _id: '01373',
```



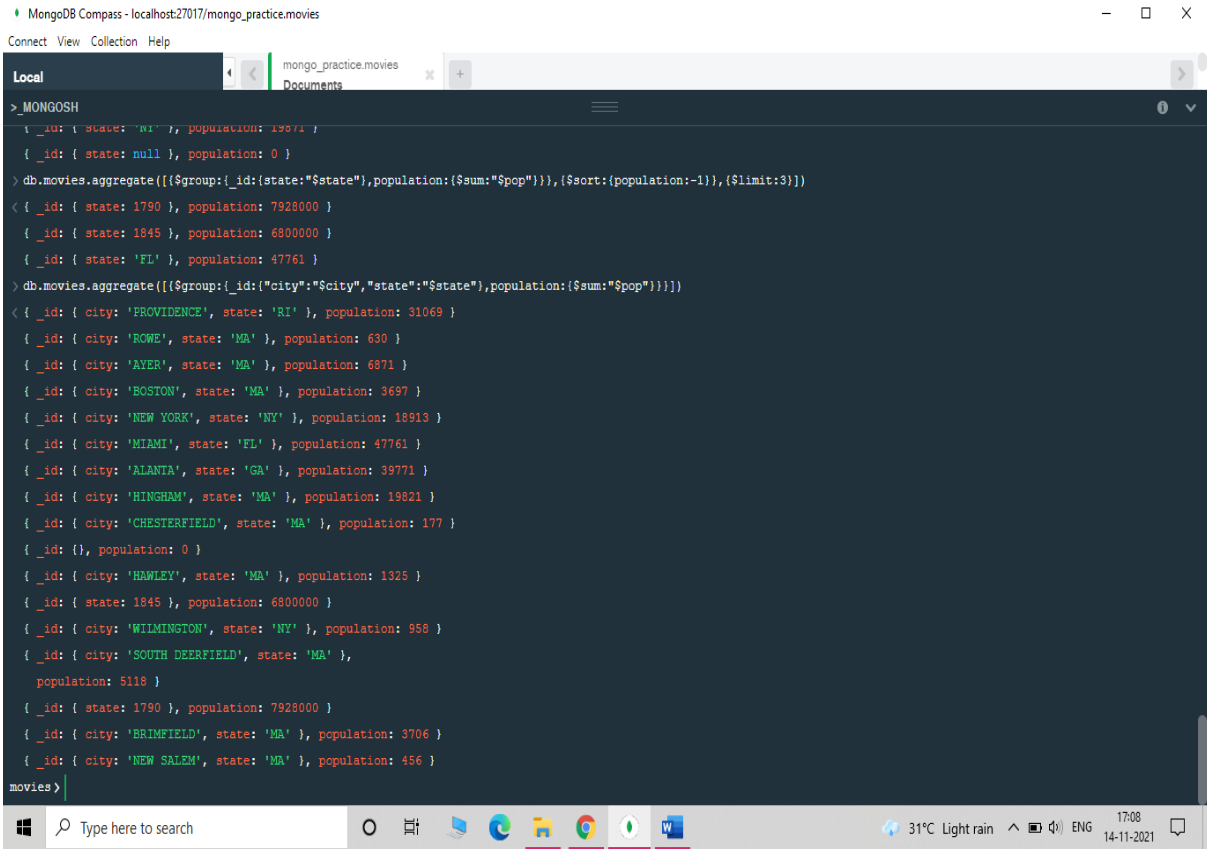
```
{ _id: '01373',
  city: 'SOUTH DEERFIELD',
  loc: [ -72.615268, 42.475616 ],
  pop: 5118,
  state: 'MA' }
{ _id: '01436',
  city: 'ALANTA',
  loc: [ -72.064647, 42.593568 ],
  pop: 4386,
  state: 'GA' }
{ _id: '01010',
  city: 'BRIMFIELD',
  loc: [ -72.188455, 42.116543 ],
  pop: 3706,
  state: 'MA' }
{ _id: '02108',
  city: 'BOSTON',
  pop: 3697,
  state: 'MA',
  capital: { name: 'Boston', electoralCollege: 11 } }
{ _id: '01922',
  city: 'ALANTA',
  loc: [ -70.935053, 42.756792 ],
  pop: 2006,
  state: 'GA' }
```



```
{ _id: '01339',
  city: 'HAWLEY',
  loc: [ -72.880162, 42.621802 ],
  pop: 1325,
  state: 'MA' }
{ _id: '12997',
  city: 'WILMINGTON',
  pop: 958,
  state: 'NY',
  capital: { name: 'Albany', electoralCollege: 29 } }
{ _id: '01367',
  city: 'ROWE',
  loc: [ -72.925776, 42.695289 ],
  pop: 630,
  state: 'MA' }
{ _id: '01355',
  city: 'NEW SALEM',
  loc: [ -72.306241, 42.514643 ],
  pop: 456,
  state: 'MA' }
{ _id: '01012',
  city: 'CHESTERFIELD',
  pop: 177,
  state: 'MA',
  capital: { name: 'Boston', electoralCollege: 11 } }
```



```
{ _id: 1,
  username: 'GoodBuy',
  first_name: 'Good Guy',
  last_name: 'Greg' }
Type "it" for more
> db.movies.aggregate([{$group:{_id:{state:"$state"},population:{$sum:"$pop"}}},{$sort:{population:-1}}])
< { _id: { state: 1790 }, population: 7928000 }
  { _id: { state: 1845 }, population: 6800000 }
  { _id: { state: 'FL' }, population: 47761 }
  { _id: { state: 'MA' }, population: 41801 }
  { _id: { state: 'GA' }, population: 39771 }
  { _id: { state: 'RI' }, population: 31069 }
  { _id: { state: 'NY' }, population: 19871 }
  { _id: { state: null }, population: 0 }
> db.movies.aggregate([{$group:{_id:{state:"$state"},population:{$sum:"$pop"}}},{$sort:{population:-1}},{$limit:3}])
< { _id: { state: 1790 }, population: 7928000 }
  { _id: { state: 1845 }, population: 6800000 }
  { _id: { state: 'FL' }, population: 47761 }
> db.movies.aggregate([{$group:{_id:{"city":"$city","state":"$state"},population:{$sum:"$pop"}}}])
< { _id: { city: 'PROVIDENCE', state: 'RI' }, population: 31069 }
  { _id: { city: 'ROWE', state: 'MA' }, population: 630 }
  { _id: { city: 'AYER', state: 'MA' }, population: 6871 }
  { _id: { city: 'BOSTON', state: 'MA' }, population: 3697 }
  { _id: { city: 'NEW YORK', state: 'NY' }, population: 18913 }
  { _id: { city: 'MIAMI', state: 'FL' }, population: 47761 }
  { _id: { city: 'ALANTA', state: 'GA' }, population: 39771 }
```



```
  { _id: { state: 'NY' }, population: 19871 }
  { _id: { state: null }, population: 0 }
> db.movies.aggregate([{$group:{_id:{state:"$state"},population:{$sum:"$pop"}}},{$sort:{population:-1}},{$limit:3}])
< { _id: { state: 1790 }, population: 7928000 }
  { _id: { state: 1845 }, population: 6800000 }
  { _id: { state: 'FL' }, population: 47761 }
> db.movies.aggregate([{$group:{_id:{"city":"$city","state":"$state"},population:{$sum:"$pop"}}}])
< { _id: { city: 'PROVIDENCE', state: 'RI' }, population: 31069 }
  { _id: { city: 'ROWE', state: 'MA' }, population: 630 }
  { _id: { city: 'AYER', state: 'MA' }, population: 6871 }
  { _id: { city: 'BOSTON', state: 'MA' }, population: 3697 }
  { _id: { city: 'NEW YORK', state: 'NY' }, population: 18913 }
  { _id: { city: 'MIAMI', state: 'FL' }, population: 47761 }
  { _id: { city: 'ALANTA', state: 'GA' }, population: 39771 }
  { _id: { city: 'HINGHAM', state: 'MA' }, population: 19821 }
  { _id: { city: 'CHESTERFIELD', state: 'MA' }, population: 177 }
  { _id: {}, population: 0 }
  { _id: { city: 'HAWLEY', state: 'MA' }, population: 1325 }
  { _id: { state: 1845 }, population: 6800000 }
  { _id: { city: 'WILMINGTON', state: 'NY' }, population: 958 }
  { _id: { city: 'SOUTH DEERFIELD', state: 'MA' },
    population: 5118 }
  { _id: { state: 1790 }, population: 7928000 }
  { _id: { city: 'BRIMFIELD', state: 'MA' }, population: 3706 }
  { _id: { city: 'NEW SALEM', state: 'MA' }, population: 456 }
movies >
```