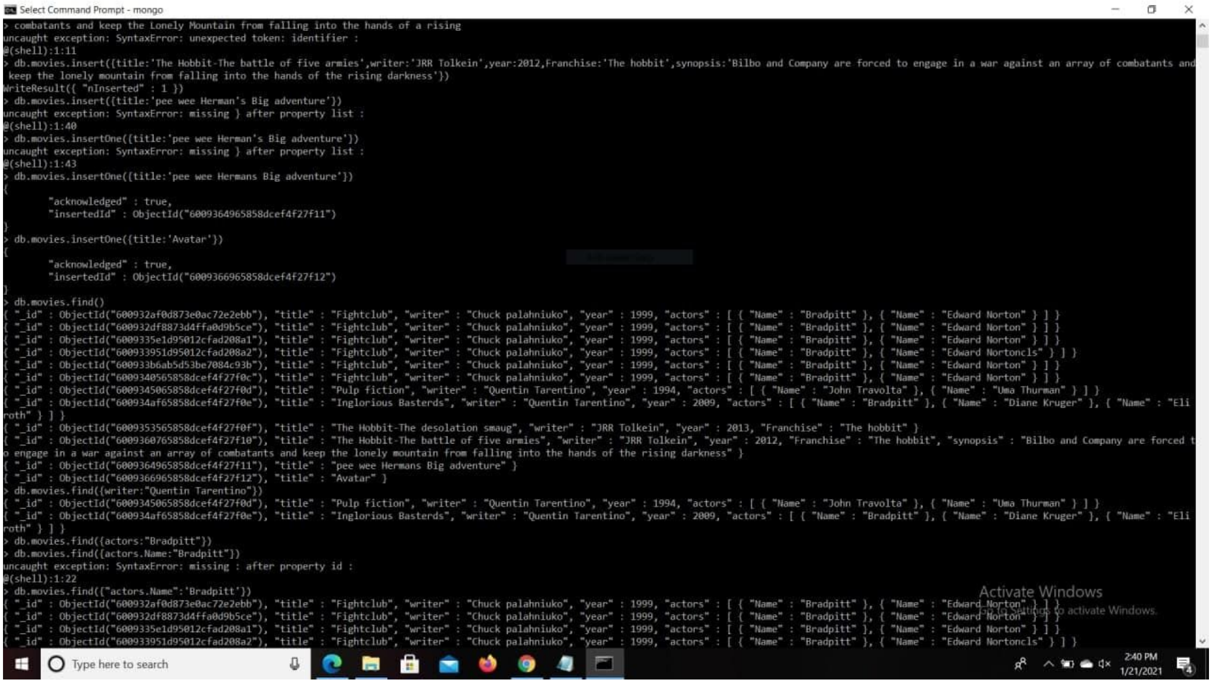
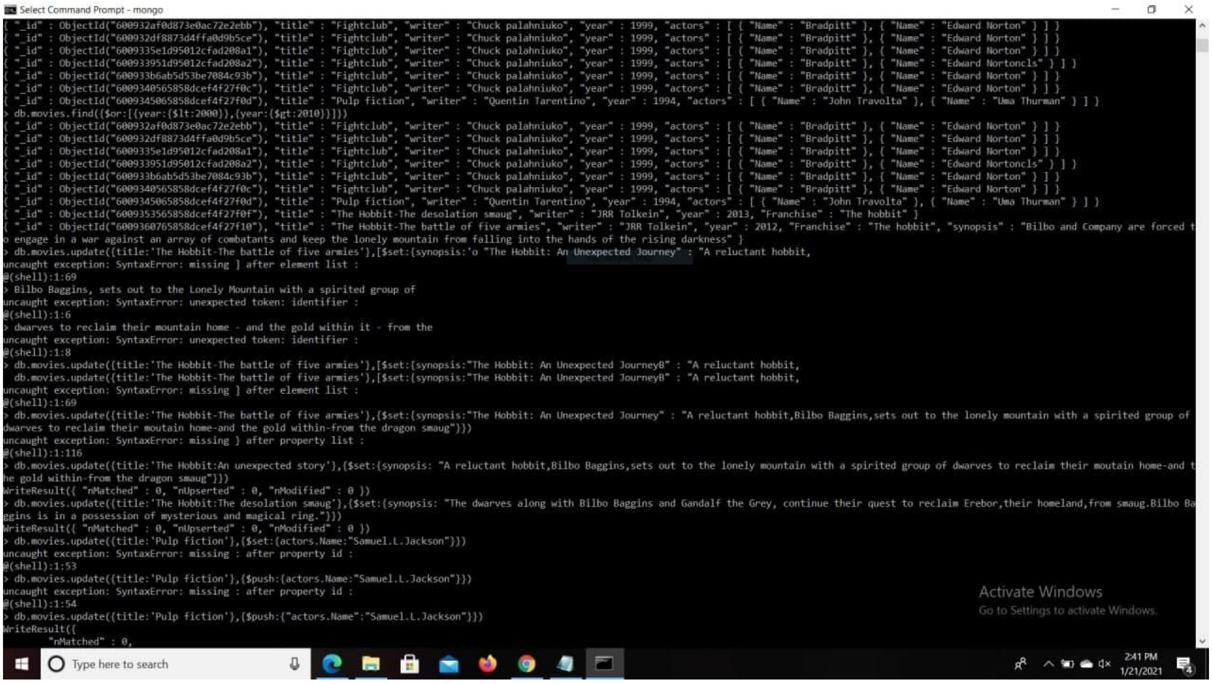
P.Monisha

MONGO DB:

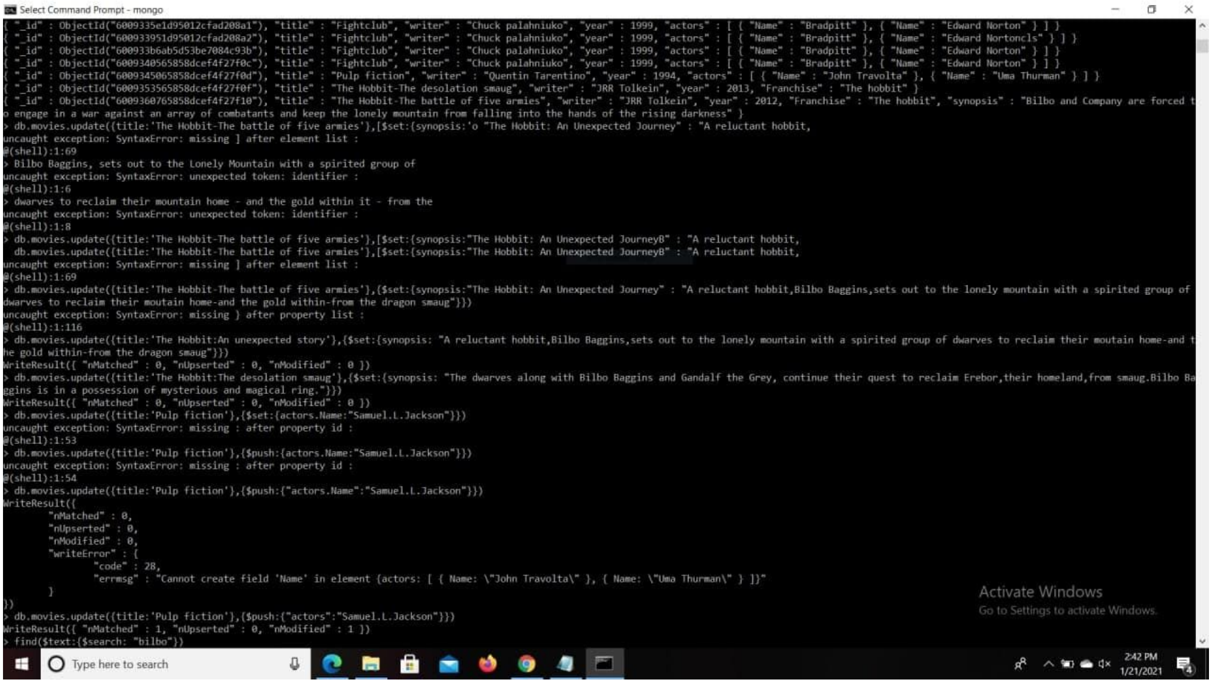

Select Command Prompt - mongo
Activate Windows
Go to Settings to activate Windows.
Type here to search
2:40 PM
1/21/2021

```
> combatants and keep the Lonely Mountain from falling into the hands of a rising
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:11
> db.movies.insert({title:'The Hobbit-The battle of five armies',writer:'JRR Tolkein',year:2012,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 the rising darkness'})
WriteResult({ "nInserted" : 1 })
> db.movies.insert({title:'pee wee Herman's Big adventure'})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:40
> db.movies.insertOne({title:'pee wee Herman's Big adventure'})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:43
> db.movies.insertOne({title:'pee wee Hermans Big adventure'})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("6009364965858dcef4f27f11")
}
> db.movies.insertOne({title:'Avatar'})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("6009366965858dcef4f27f12")
}
> db.movies.find()
{ "_id" : ObjectId("600932af0d873e0ac72e2ebb"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600932df8873d4ffa0d9b5ce"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009335e1d95012cfad208a1"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600933951d95012cfad208a2"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Nortoncls" } ] }
{ "_id" : ObjectId("600933b6ab5d53be7084c93b"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009340565858dcef4f27f0c"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009345065858dcef4f27f0d"), "title" : "Pulp fiction", "writer" : "Quentin Tarentino", "year" : 1994, "actors" : [ { "Name" : "John Travolta" }, { "Name" : "Uma Thurman" } ] }
{ "_id" : ObjectId("600934af65858dcef4f27f0e"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarentino", "year" : 2009, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Diane Kruger" }, { "Name" : "Eli
roth" } ] }
{ "_id" : ObjectId("6009353565858dcef4f27f0f"), "title" : "The Hobbit-The desolation smaug", "writer" : "JRR Tolkein", "year" : 2013, "Franchise" : "The hobbit" }
{ "_id" : ObjectId("6009360765858dcef4f27f10"), "title" : "The Hobbit-The battle of five armies", "writer" : "JRR Tolkein", "year" : 2012, "Franchise" : "The hobbit", "synopsis" : "Bilbo and Company are forced t
o engage in a war against an array of combatants and keep the lonely mountain from falling into the hands of the rising darkness" }
{ "_id" : ObjectId("6009364965858dcef4f27f11"), "title" : "pee wee Hermans Big adventure" }
{ "_id" : ObjectId("6009366965858dcef4f27f12"), "title" : "Avatar" }
> db.movies.find({writer:"Quentin Tarentino"})
{ "_id" : ObjectId("6009345065858dcef4f27f0d"), "title" : "Pulp fiction", "writer" : "Quentin Tarentino", "year" : 1994, "actors" : [ { "Name" : "John Travolta" }, { "Name" : "Uma Thurman" } ] }
{ "_id" : ObjectId("600934af65858dcef4f27f0e"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarentino", "year" : 2009, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Diane Kruger" }, { "Name" : "Eli
roth" } ] }
> db.movies.find({actors:"Bradpitt"})
> db.movies.find({actors.Name:"Bradpitt"})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:22
> db.movies.find({"actors.Name":'Bradpitt'})
{ "_id" : ObjectId("600932af0d873e0ac72e2ebb"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600932df8873d4ffa0d9b5ce"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009335e1d95012cfad208a1"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600933951d95012cfad208a2"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Nortoncls" } ] }
```

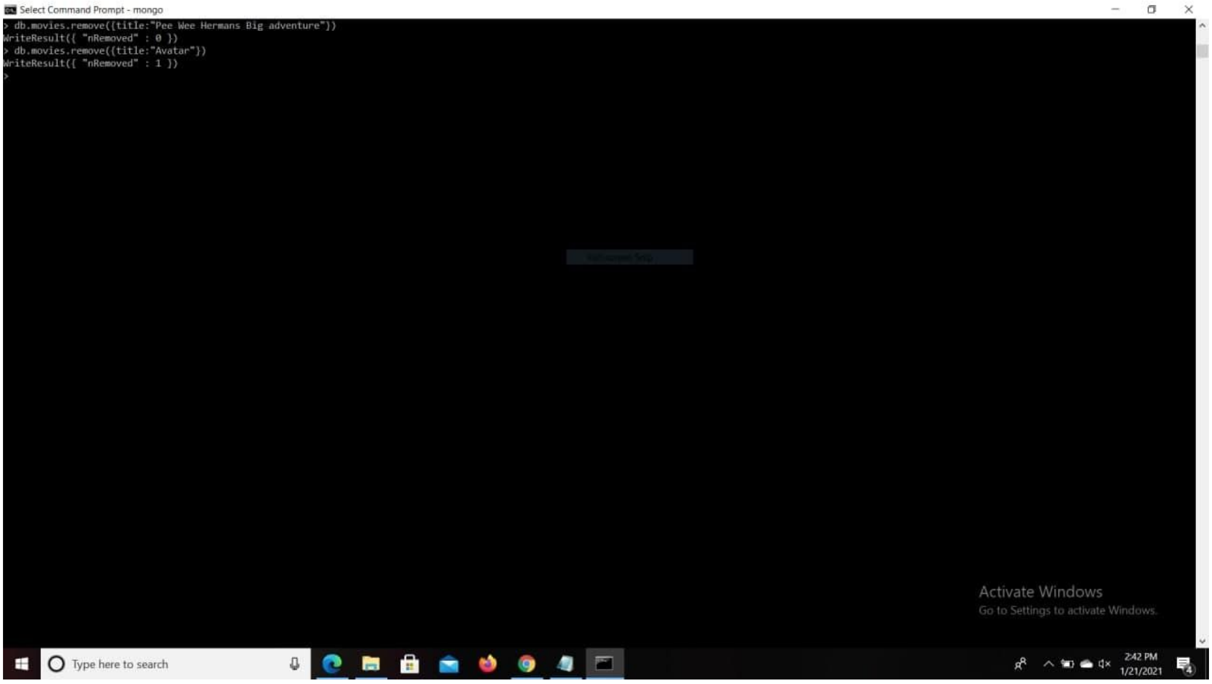

Select Command Prompt - mongo
Activate Windows
Go to Settings to activate Windows.
Type here to search
2:41 PM
1/21/2021

```
{ "_id" : ObjectId("600932af0d873e0ac72e2ebb"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600932df8873d4ffa0d9b5ce"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009335e1d95012cfad208a1"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600933951d95012cfad208a2"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Nortoncls" } ] }
{ "_id" : ObjectId("600933b6ab5d53be7084c93b"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009340565858dcef4f27f0c"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009345065858dcef4f27f0d"), "title" : "Pulp fiction", "writer" : "Quentin Tarentino", "year" : 1994, "actors" : [ { "Name" : "John Travolta" }, { "Name" : "Uma Thurman" } ] }
> db.movies.find({$or:[{year:{$lt:2000}},{year:{$gt:2010}}]})
{ "_id" : ObjectId("600932af0d873e0ac72e2ebb"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600932df8873d4ffa0d9b5ce"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009335e1d95012cfad208a1"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600933951d95012cfad208a2"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Nortoncls" } ] }
{ "_id" : ObjectId("600933b6ab5d53be7084c93b"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009340565858dcef4f27f0c"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009345065858dcef4f27f0d"), "title" : "Pulp fiction", "writer" : "Quentin Tarentino", "year" : 1994, "actors" : [ { "Name" : "John Travolta" }, { "Name" : "Uma Thurman" } ] }
{ "_id" : ObjectId("6009353565858dcef4f27f0f"), "title" : "The Hobbit-The desolation smaug", "writer" : "JRR Tolkein", "year" : 2013, "Franchise" : "The hobbit" }
{ "_id" : ObjectId("6009360765858dcef4f27f10"), "title" : "The Hobbit-The battle of five armies", "writer" : "JRR Tolkein", "year" : 2012, "Franchise" : "The hobbit", "synopsis" : "Bilbo and Company are forced t
o engage in a war against an array of combatants and keep the lonely mountain from falling into the hands of the rising darkness" }
> db.movies.update({title:'The Hobbit-The battle of five armies'},[$set:{synopsis:'o "The Hobbit: An Unexpected Journey" : "A reluctant hobbit,
uncaught exception: SyntaxError: missing ] after element list :
@(shell):1:69
> Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:6
> dwarves to reclaim their mountain home - and the gold within it - from the
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:8
> db.movies.update({title:'The Hobbit-The battle of five armies'},[$set:{synopsis:"The Hobbit: An Unexpected JourneyB" : "A reluctant hobbit,
  db.movies.update({title:'The Hobbit-The battle of five armies'},[$set:{synopsis:"The Hobbit: An Unexpected JourneyB" : "A reluctant hobbit,
uncaught exception: SyntaxError: missing ] after element list :
@(shell):1:69
> db.movies.update({title:'The Hobbit-The battle of five armies'},{$set:{synopsis:"The Hobbit: An Unexpected Journey" : "A reluctant hobbit,Bilbo Baggins,sets out to the lonely mountain with a spirited group of
dwarves to reclaim their moutain home-and the gold within-from the dragon smaug"}})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:116
> db.movies.update({title:'The Hobbit:An unexpected story'},{$set:{synopsis: "A reluctant hobbit,Bilbo Baggins,sets out to the lonely mountain with a spirited group of dwarves to reclaim their moutain home-and t
he gold within-from the dragon smaug"}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.movies.update({title:'The Hobbit:The desolation smaug'},{$set:{synopsis: "The dwarves along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor,their homeland,from smaug.Bilbo Ba
ggins is in a possession of mysterious and magical ring."}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.movies.update({title:'Pulp fiction'},{$set:{actors.Name:"Samuel.L.Jackson"}})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:53
> db.movies.update({title:'Pulp fiction'},{$push:{actors.Name:"Samuel.L.Jackson"}})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:54
> db.movies.update({title:'Pulp fiction'},{$push:{"actors.Name":"Samuel.L.Jackson"}})
WriteResult({
        "nMatched" : 0,
```

P.Monisha

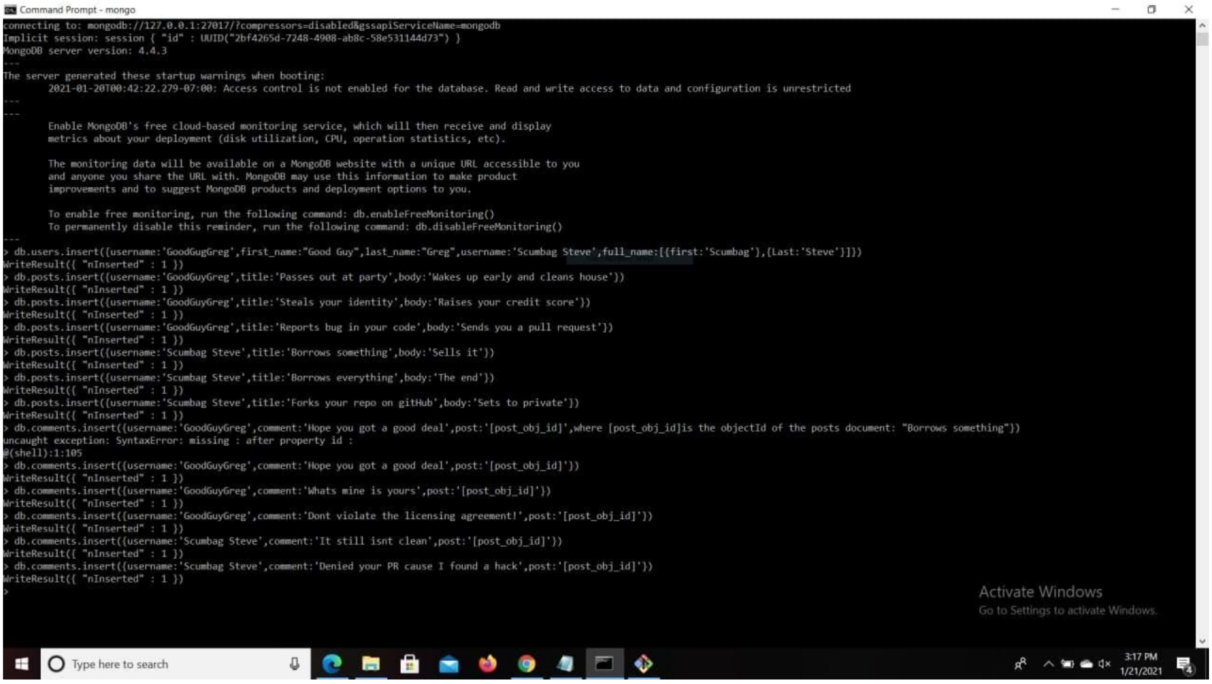


```
{ "_id" : ObjectId("6009335e1d95012cfad208a1"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("600933951d95012cfad208a2"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Nortoncls" } ] }
{ "_id" : ObjectId("600933b6ab5d53be7084c93b"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009340565858dcef4f27f0c"), "title" : "Fightclub", "writer" : "Chuck palahniuko", "year" : 1999, "actors" : [ { "Name" : "Bradpitt" }, { "Name" : "Edward Norton" } ] }
{ "_id" : ObjectId("6009345065858dcef4f27f0d"), "title" : "Pulp fiction", "writer" : "Quentin Tarentino", "year" : 1994, "actors" : [ { "Name" : "John Travolta" }, { "Name" : "Uma Thurman" } ] }
{ "_id" : ObjectId("6009353565858dcef4f27f0f"), "title" : "The Hobbit-The desolation smaug", "writer" : "JRR Tolkein", "year" : 2013, "Franchise" : "The hobbit" }
{ "_id" : ObjectId("6009360765858dcef4f27f10"), "title" : "The Hobbit-The battle of five armies", "writer" : "JRR Tolkein", "year" : 2012, "Franchise" : "The hobbit", "synopsis" : "Bilbo and Company are forced t
o engage in a war against an array of combatants and keep the lonely mountain from falling into the hands of the rising darkness" }
> db.movies.update({title:'The Hobbit-The battle of five armies'},[$set:{synopsis:'o "The Hobbit: An Unexpected Journey" : "A reluctant hobbit,
uncaught exception: SyntaxError: missing ] after element list :
@(shell):1:69
> Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:6
> dwarves to reclaim their mountain home - and the gold within it - from the
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:8
> db.movies.update({title:'The Hobbit-The battle of five armies'},[$set:{synopsis:"The Hobbit: An Unexpected JourneyB" : "A reluctant hobbit,
  db.movies.update({title:'The Hobbit-The battle of five armies'},[$set:{synopsis:"The Hobbit: An Unexpected JourneyB" : "A reluctant hobbit,
uncaught exception: SyntaxError: missing ] after element list :
@(shell):1:69
> db.movies.update({title:'The Hobbit-The battle of five armies'},{$set:{synopsis:"The Hobbit: An Unexpected Journey" : "A reluctant hobbit,Bilbo Baggins,sets out to the lonely mountain with a spirited group of
dwarves to reclaim their moutain home-and the gold within-from the dragon smaug"}})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:116
> db.movies.update({title:'The Hobbit:An unexpected story'},{$set:{synopsis: "A reluctant hobbit,Bilbo Baggins,sets out to the lonely mountain with a spirited group of dwarves to reclaim their moutain home-and t
he gold within-from the dragon smaug"}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.movies.update({title:'The Hobbit:The desolation smaug'},{$set:{synopsis: "The dwarves along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor,their homeland,from smaug.Bilbo Ba
ggins is in a possession of mysterious and magical ring."}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.movies.update({title:'Pulp fiction'},{$set:{actors.Name:"Samuel.L.Jackson"}})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:53
> db.movies.update({title:'Pulp fiction'},{$push:{actors.Name:"Samuel.L.Jackson"}})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:54
> db.movies.update({title:'Pulp fiction'},{$push:{"actors.Name":"Samuel.L.Jackson"}})
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 0,
        "nModified" : 0,
        "writeError" : {
                "code" : 28,
                "errmsg" : "Cannot create field 'Name' in element {actors: [ { Name: \"John Travolta\" }, { Name: \"Uma Thurman\" } ]}"
        }
})
> db.movies.update({title:'Pulp fiction'},{$push:{"actors":"Samuel.L.Jackson"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> find($text:{$search: "bilbo"})
```



```
> db.movies.remove({title:"Pee Wee Hermans Big adventure"})
WriteResult({ "nRemoved" : 0 })
> db.movies.remove({title:"Avatar"})
WriteResult({ "nRemoved" : 1 })
>
```

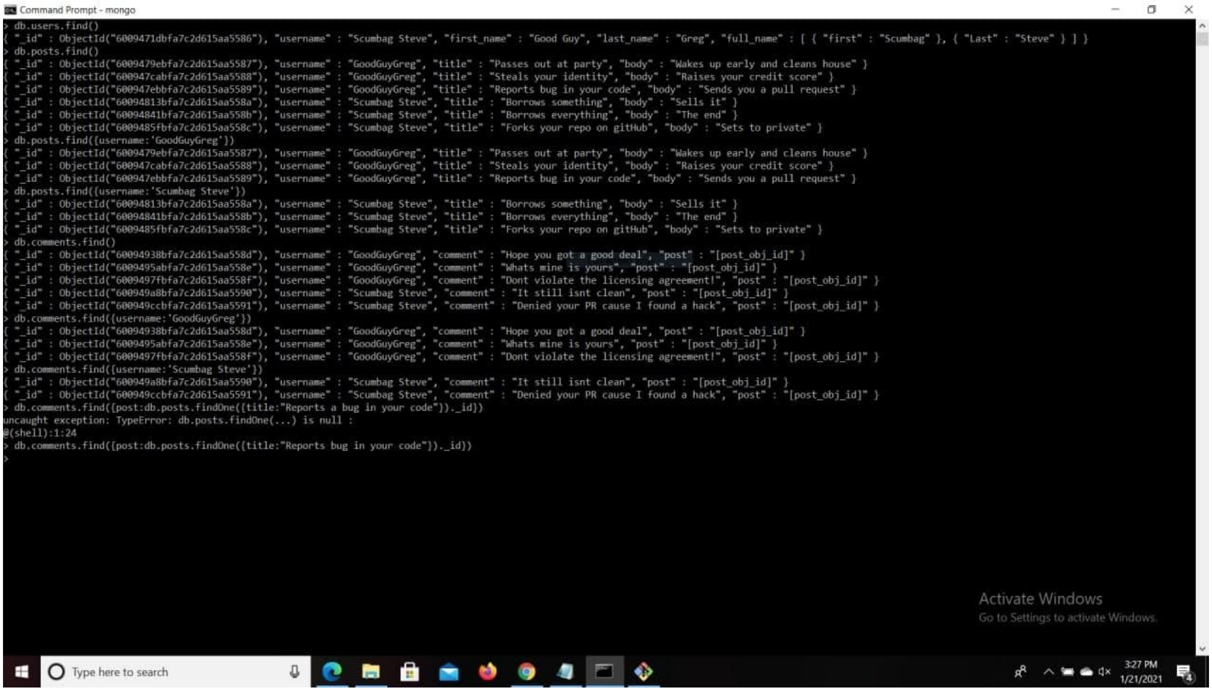
P.Monisha



```
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("2bf4265d-7248-4908-ab8c-58e531144d73") }
MongoDB server version: 4.4.3
---
The server generated these startup warnings when booting:
        2021-01-20T00:42:22.279-07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> db.users.insert({username:'GoodGugGreg',first_name:"Good Guy",last_name:"Greg",username:'Scumbag Steve',full_name:[{first:'Scumbag'},{Last:'Steve'}]})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({username:'GoodGuyGreg',title:'Passes out at party',body:'Wakes up early and cleans house'})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({username:'GoodGuyGreg',title:'Steals your identity',body:'Raises your credit score'})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({username:'GoodGuyGreg',title:'Reports bug in your code',body:'Sends you a pull request'})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({username:'Scumbag Steve',title:'Borrows something',body:'Sells it'})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({username:'Scumbag Steve',title:'Borrows everything',body:'The end'})
WriteResult({ "nInserted" : 1 })
> db.posts.insert({username:'Scumbag Steve',title:'Forks your repo on gitHub',body:'Sets to private'})
WriteResult({ "nInserted" : 1 })
> db.comments.insert({username:'GoodGuyGreg',comment:'Hope you got a good deal',post:'[post_obj_id]',where [post_obj_id]is the objectId of the posts document: "Borrows something"})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:105
> db.comments.insert({username:'GoodGuyGreg',comment:'Hope you got a good deal',post:'[post_obj_id]'})
WriteResult({ "nInserted" : 1 })
> db.comments.insert({username:'GoodGuyGreg',comment:'Whats mine is yours',post:'[post_obj_id]'})
WriteResult({ "nInserted" : 1 })
> db.comments.insert({username:'GoodGuyGreg',comment:'Dont violate the licensing agreement!',post:'[post_obj_id]'})
WriteResult({ "nInserted" : 1 })
> db.comments.insert({username:'Scumbag Steve',comment:'It still isnt clean',post:'[post_obj_id]'})
WriteResult({ "nInserted" : 1 })
> db.comments.insert({username:'Scumbag Steve',comment:'Denied your PR cause I found a hack',post:'[post_obj_id]'})
WriteResult({ "nInserted" : 1 })
>
```



```
> db.users.find()
{ "_id" : ObjectId("6009471dbfa7c2d615aa5586"), "username" : "Scumbag Steve", "first_name" : "Good Guy", "last_name" : "Greg", "full_name" : [ { "first" : "Scumbag" }, { "Last" : "Steve" } ] }
> db.posts.find()
{ "_id" : ObjectId("6009479ebfa7c2d615aa5587"), "username" : "GoodGuyGreg", "title" : "Passes out at party", "body" : "Wakes up early and cleans house" }
{ "_id" : ObjectId("600947cabfa7c2d615aa5588"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("600947ebbfa7c2d615aa5589"), "username" : "GoodGuyGreg", "title" : "Reports bug in your code", "body" : "Sends you a pull request" }
{ "_id" : ObjectId("60094813bfa7c2d615aa558a"), "username" : "Scumbag Steve", "title" : "Borrows something", "body" : "Sells it" }
{ "_id" : ObjectId("60094841bfa7c2d615aa558b"), "username" : "Scumbag Steve", "title" : "Borrows everything", "body" : "The end" }
{ "_id" : ObjectId("6009485fbfa7c2d615aa558c"), "username" : "Scumbag Steve", "title" : "Forks your repo on gitHub", "body" : "Sets to private" }
> db.posts.find({username:'GoodGuyGreg'})
{ "_id" : ObjectId("6009479ebfa7c2d615aa5587"), "username" : "GoodGuyGreg", "title" : "Passes out at party", "body" : "Wakes up early and cleans house" }
{ "_id" : ObjectId("600947cabfa7c2d615aa5588"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("600947ebbfa7c2d615aa5589"), "username" : "GoodGuyGreg", "title" : "Reports bug in your code", "body" : "Sends you a pull request" }
> db.posts.find({username:'Scumbag Steve'})
{ "_id" : ObjectId("60094813bfa7c2d615aa558a"), "username" : "Scumbag Steve", "title" : "Borrows something", "body" : "Sells it" }
{ "_id" : ObjectId("60094841bfa7c2d615aa558b"), "username" : "Scumbag Steve", "title" : "Borrows everything", "body" : "The end" }
{ "_id" : ObjectId("6009485fbfa7c2d615aa558c"), "username" : "Scumbag Steve", "title" : "Forks your repo on gitHub", "body" : "Sets to private" }
> db.comments.find()
{ "_id" : ObjectId("60094938bfa7c2d615aa558d"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("6009495abfa7c2d615aa558e"), "username" : "GoodGuyGreg", "comment" : "Whats mine is yours", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("6009497fbfa7c2d615aa558f"), "username" : "GoodGuyGreg", "comment" : "Dont violate the licensing agreement!", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("600949a8bfa7c2d615aa5590"), "username" : "Scumbag Steve", "comment" : "It still isnt clean", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("600949ccbfa7c2d615aa5591"), "username" : "Scumbag Steve", "comment" : "Denied your PR cause I found a hack", "post" : "[post_obj_id]" }
> db.comments.find({username:'GoodGuyGreg'})
{ "_id" : ObjectId("60094938bfa7c2d615aa558d"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("6009495abfa7c2d615aa558e"), "username" : "GoodGuyGreg", "comment" : "Whats mine is yours", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("6009497fbfa7c2d615aa558f"), "username" : "GoodGuyGreg", "comment" : "Dont violate the licensing agreement!", "post" : "[post_obj_id]" }
> db.comments.find({username:'Scumbag Steve'})
{ "_id" : ObjectId("600949a8bfa7c2d615aa5590"), "username" : "Scumbag Steve", "comment" : "It still isnt clean", "post" : "[post_obj_id]" }
{ "_id" : ObjectId("600949ccbfa7c2d615aa5591"), "username" : "Scumbag Steve", "comment" : "Denied your PR cause I found a hack", "post" : "[post_obj_id]" }
> db.comments.find({post:db.posts.findOne({title:"Reports a bug in your code"})._id})
uncaught exception: TypeError: db.posts.findOne(...) is null :
@(shell):1:24
> db.comments.find({post:db.posts.findOne({title:"Reports bug in your code"})._id})
>
```

MongoDB-Assignment Aggregation

```
 db.zipcodes.find({state: 'GA',city: 'ATLANTA'})
db.zipcodes. aggregate (
   [ { $match : {state: 'GA',city: 'ATLANTA'}} ]);

db.zipcodes. aggregate (
   [ { $match : {state: 'GA',city: 'ATLANTA'}}, { $group: { _id: null, count: { $sum: 1 } } } ])
```

P.Monisha

```
db.zipcodes. aggregate (
  [ { $match : {state: 'GA',city: 'ATLANTA'}}, { $group: { Total Population: { $sum: “pop” } } } ])
```

POPULATION BY STATE

```
db.zipcodes.aggregate( [{ $group: { _id: "$state", totalPop: { $sum: "$pop" } } } ])
```

```
db.zipcodes.aggregate( [{ $group: { _id: "$state", totalPop: { $sum: "$pop" } } }
{ $sort: { pop: 1 } }
```

```
db.zipcodes.aggregate( [{ $group: { _id: "$state", totalPop: { $sum: "$pop" } } }
{ $sort: { pop: 1 } },
  { $group:
    {
      _id : "$_id.state",
      Statename:  { $last: "$_id.state" },
      StatePop:   { $last: "$totalpop" },
      }
  },
{$group:
{
{ $limit : 3 }
}
}])
```

POPULATION BY CITY

```
db.zipcodes.aggregate( [   { $group:{
      _id: { state: "$state", city: "$city" },
      pop: { $sum: "$pop" }} }])
```

```
db.zipcodes.aggregate( [   { $group:{
      _id: { state: "$state", city: "$city" },
      pop: { $sum: "$pop" }} },
  { $sort: { pop: 1 } }])
```

```
db.zipcodes.aggregate( [   { $group:{
      _id: { state: "$state", city: "$city" },
      pop: { $sum: "$pop" }} },
  { $sort: { pop: 1 } }, {$group:
{
{ $limit : 3 }
}
},   { $group:
     {

       cityname:  { $last: "$_id.city" },

       }
   },
])


db.zipcodes.aggregate( [   { $group:{
       _id: { state: "TEXAS", city: "$city" },
       pop: { $sum: "$pop" }} },
   { $sort: { pop: 1 } }, {$group:
{
{ $limit : 3 }
}
},   { $group:
     {
       _id : "$_id.state",
       cityname:  { $last: "$_id.city" },

       }
   },
])
```

MongoDB complex Assignment :

```
db.restaurants.find()

db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1,"_id":0})

db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"address.zipcode" :1,"_id":0})

db.restaurants.find({"borough": "Bronx"}).limit(5)

db.restaurants.find({"borough": "Bronx"})

db.restaurants.find({"borough": "Bronx"}).skip(5).limit(5)

db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 90}}}})

db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}})

db.restaurants.find({"address.coord" : {$lt : -95.754168}})
```

P.Monisha

```
db.restaurants.find(
            {$and:
                [
                  {"cuisine" : {$ne :"American "}},
                  {"grades.score" : {$gt : 70}},
                  {"address.coord" : {$lt : -65.754168}}
                ]
             }
                )

db.restaurants.find(
                      {
                        "cuisine" : {$ne : "American "},
                        "grades.score" :{$gt: 70},
                        "address.coord" : {$lt : -65.754168}
                       }
                 )

db.restaurants.find( {
                        "cuisine" : {$ne : "American "},
                        "grades.grade" :"A",
                        "borough": {$ne : "Brooklyn"}
                   }
                ).sort({"cuisine":-1})

db.restaurants.find(
{name: /^Wil/},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
)

db.restaurants.find(
{name: /ces$/},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1});
db.restaurants.find(
{name: /ces$/},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
)
```

P.Monisha

```
db.restaurants.find(
{
"borough": "Bronx" ,
$or : [
{ "cuisine" : "American " },
{ "cuisine" : "Chinese" }
]
}
)

db.restaurants.find(
{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
)

db.restaurants.find(
{"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
)

db.restaurants.find(
{"grades.score" :
{ $not:
{$gt : 10}
}
},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
)

db.restaurants.find(
{$or: [
  {name: /^Wil/},
  {"$and": [
      {"cuisine" : {$ne :"American "}},
      {"cuisine" : {$ne :"Chinees"}}
   ]}
]}
,{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1}
)

db.restaurants.find(
              {
               "grades.date": ISODate("2014-08-11T00:00:00Z"),
               "grades.grade":"A" ,
               "grades.score" : 11
              },
              {"restaurant_id" : 1,"name":1,"grades":1}
            )

db.restaurants.find(
                   { "grades.1.date": ISODate("2014-08-11T00:00:00Z"),
                     "grades.1.grade":"A" ,
                     "grades.1.score" : 9
                   },
                    {"restaurant_id" : 1,"name":1,"grades":1}
                 )

db.restaurants.find(
                   {
                     "address.coord.1": {$gt : 42, $lte : 52}
                   },
                     {"restaurant_id" : 1,"name":1,"address":1,"coord":1}  );
db.restaurants.find().sort({"name":1});
db.restaurants.find().sort({"name":-1} );
db.restaurants.find().sort({"cuisine":1,"borough" : -1,});
db.restaurants.find( {"address.street" :
               { $exists : true }
              }
                 )

db.restaurants.find( {"address.coord" :
                    {$type : 1}
                  }
                 );
db.restaurants.find(  {"grades.score" : {$mod : [7,0]} },{"restaurant_id" :
1,"name":1,"grades":1})

db.restaurants.find(
                 { name :
                   { $regex : "mon.*", $options: "i" }
                 },
                    {
                      "name":1,
                      "borough":1,
             "address.coord":1,
             "cuisine" :1
            }
          )

db.restaurants.find(
          { name :
           { $regex : /^Mad/i, }
          },
            {
             "name":1,
             "borough":1,
             "address.coord":1,
             "cuisine" :1
            }
          )
```