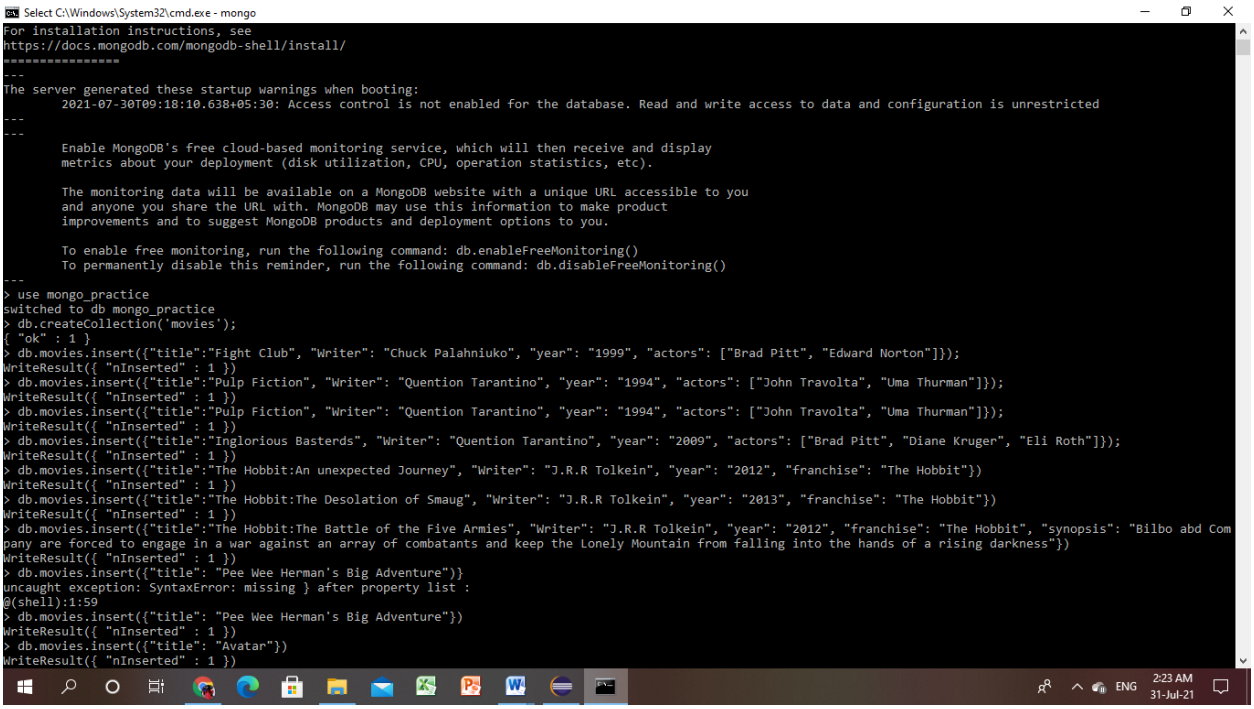
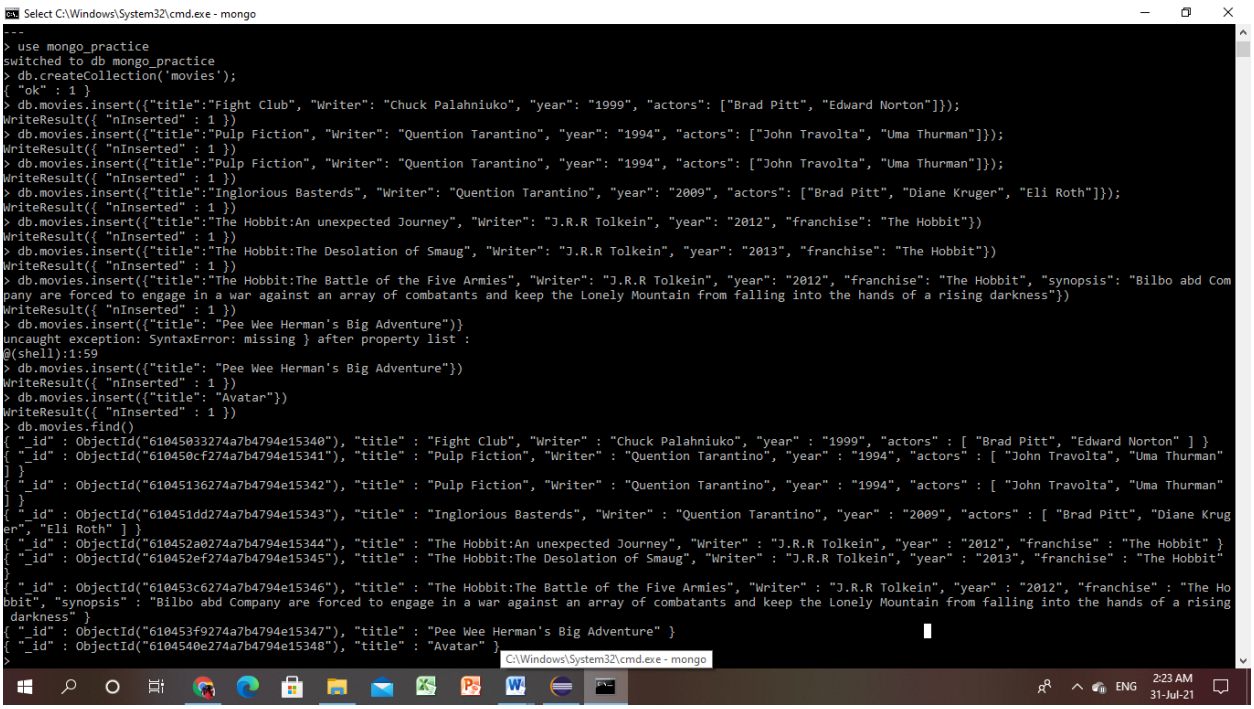
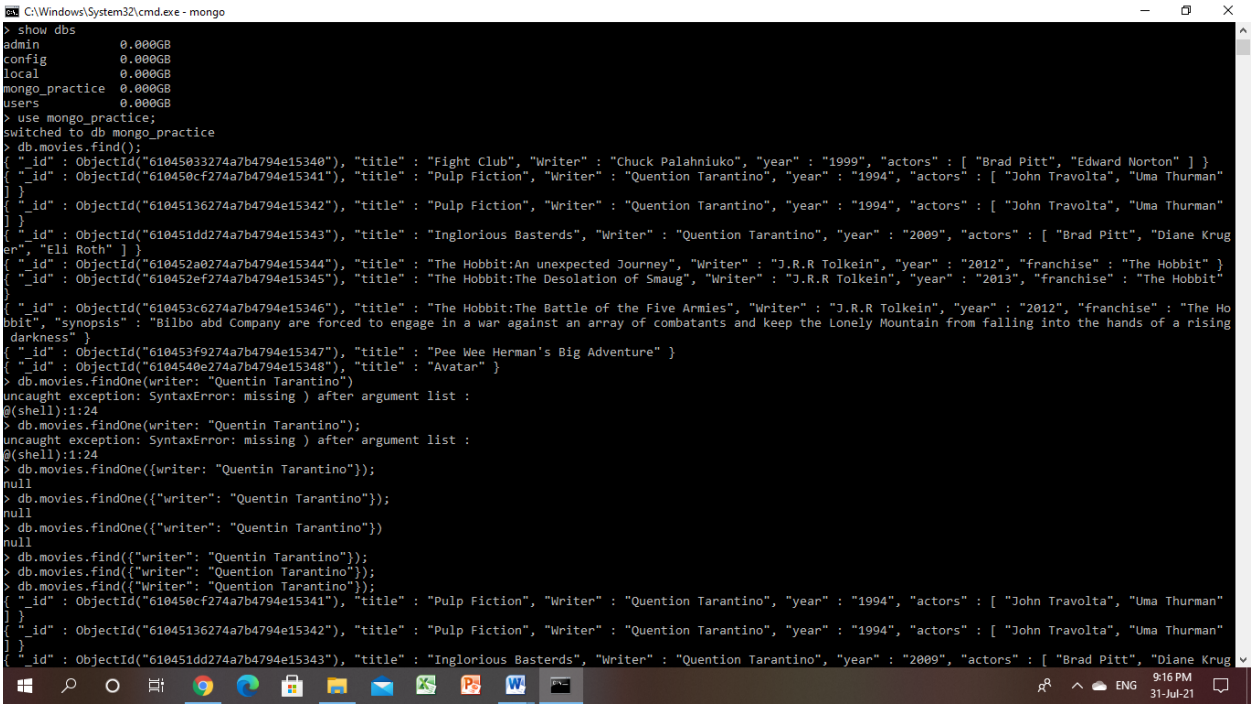
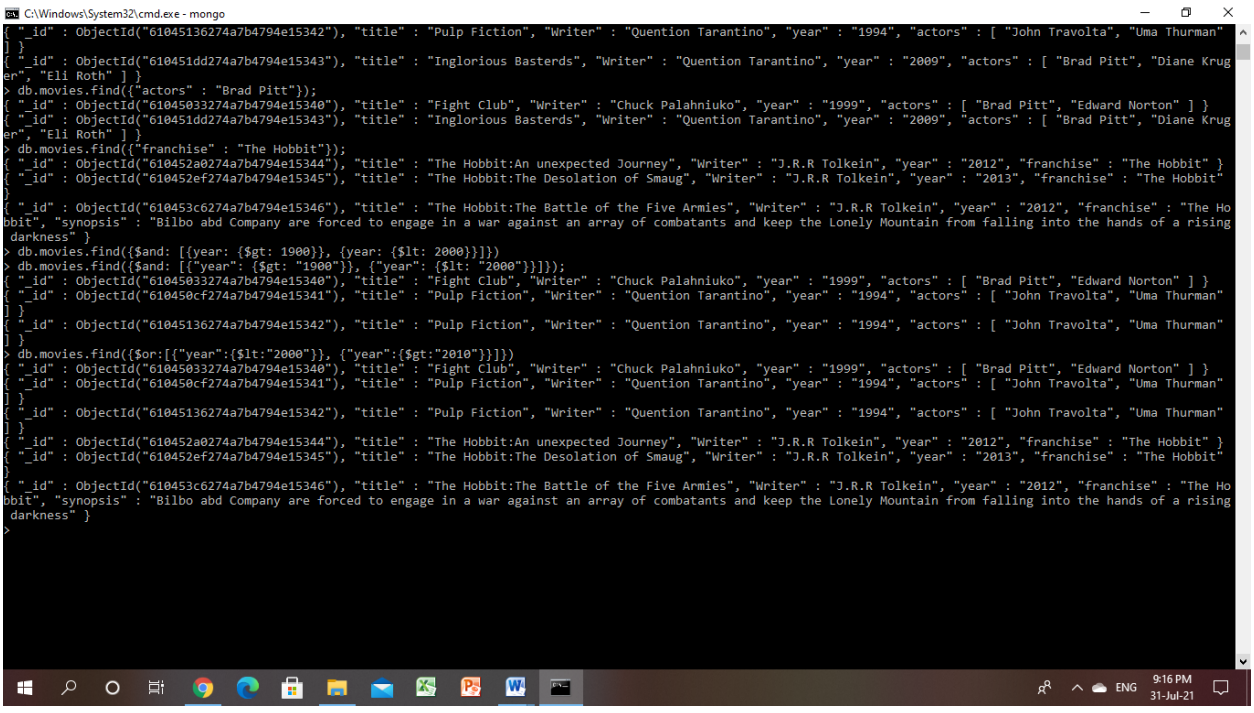
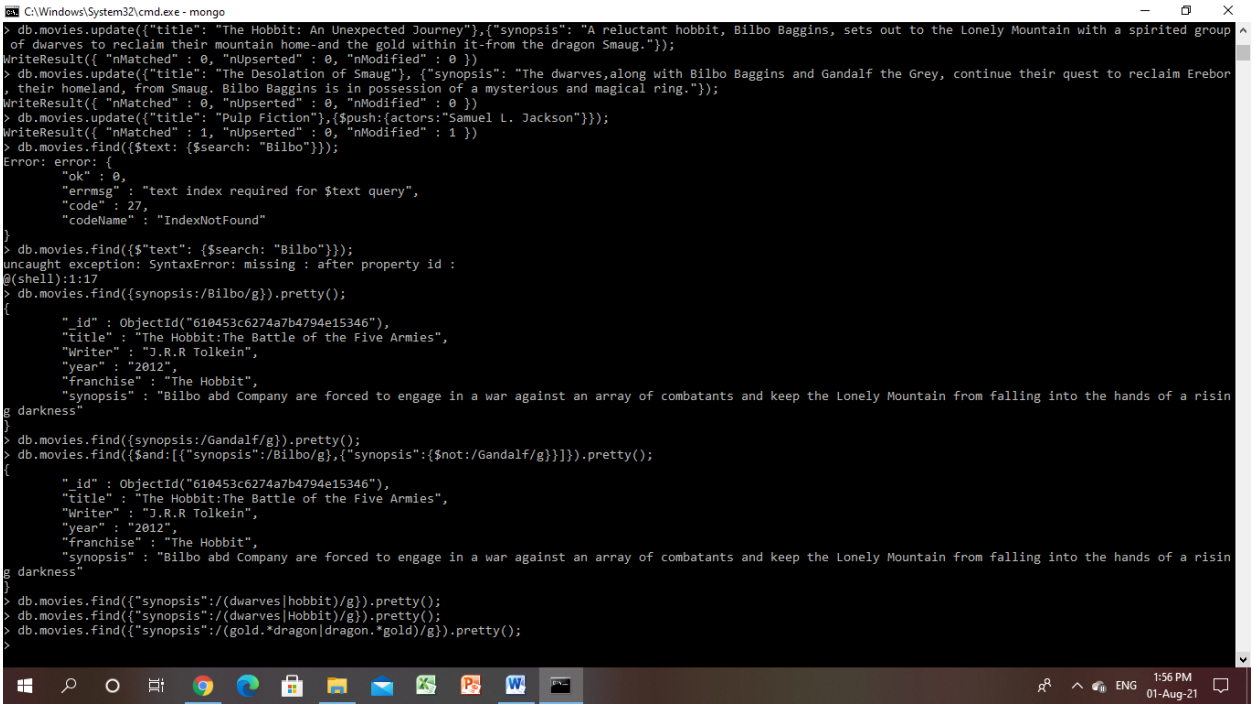
Assignment 1

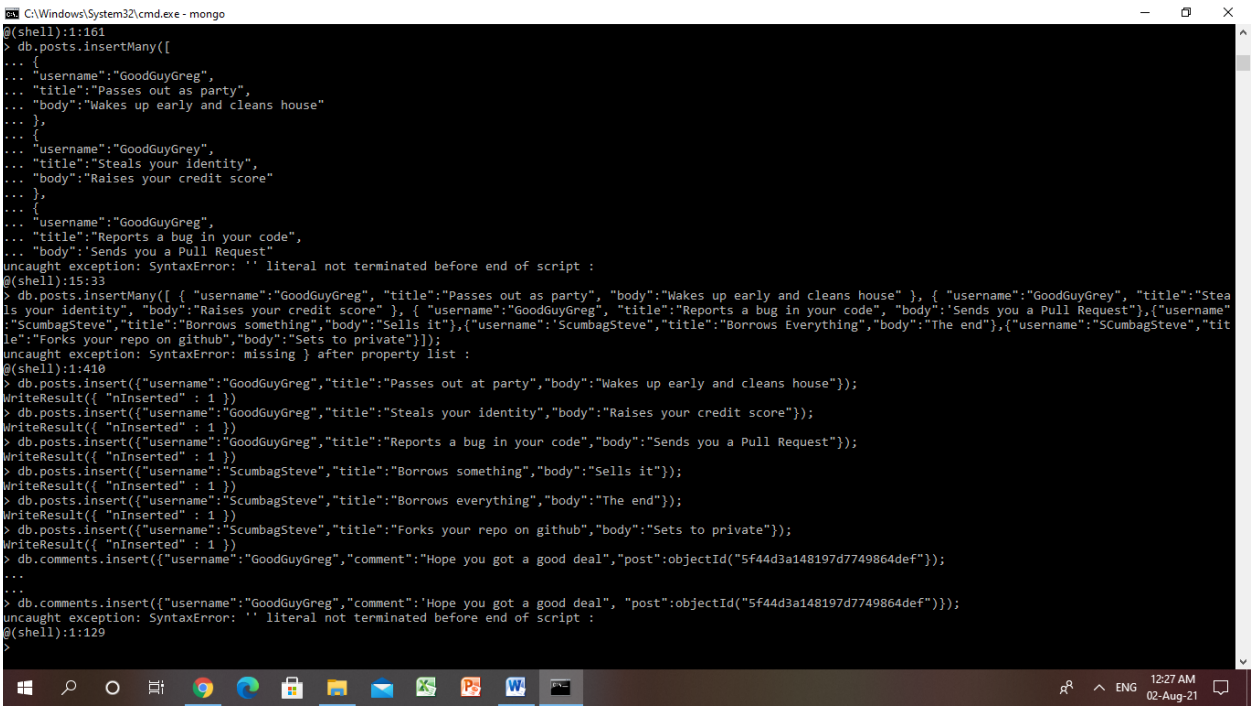


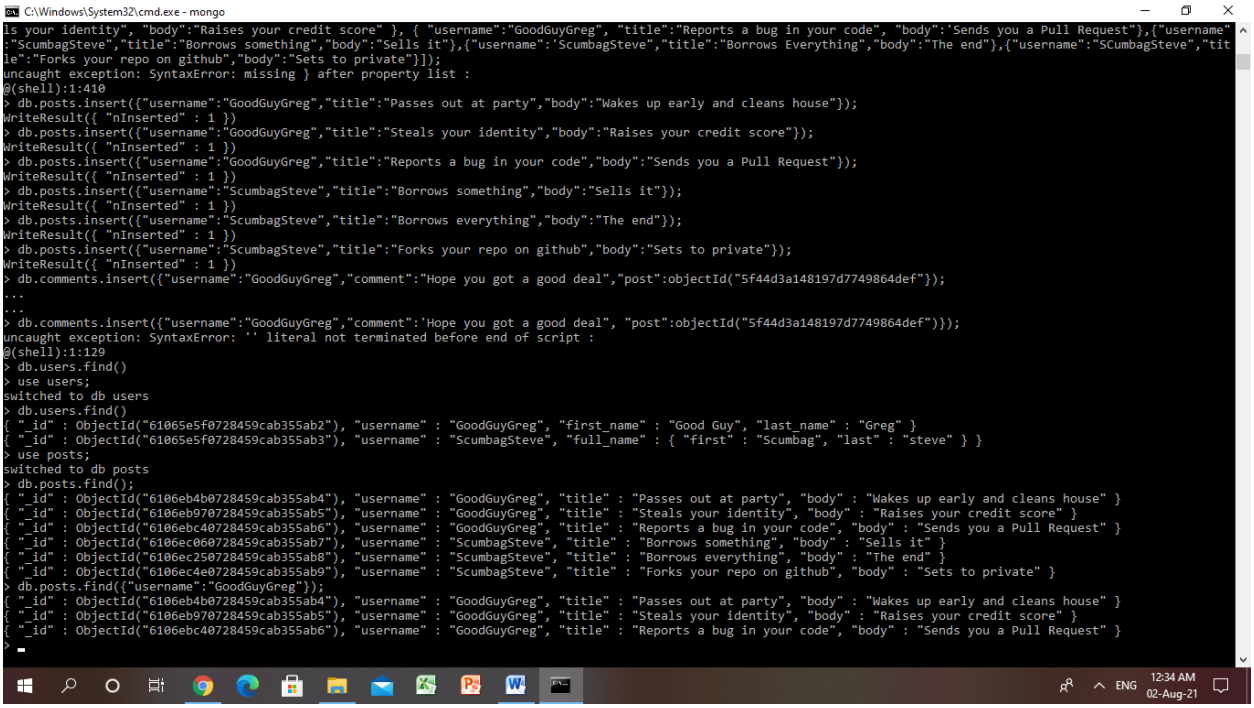


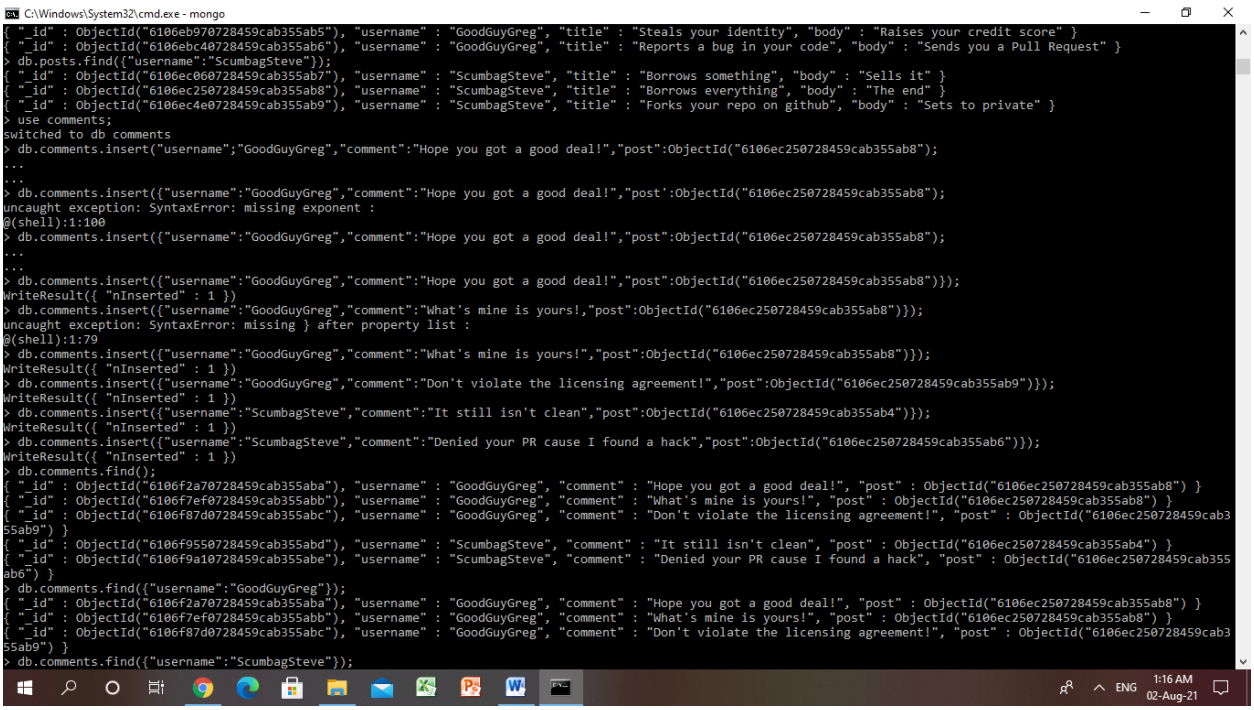

C:\Windows\System32\cmd.exe - mongo
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
mongo_practice 0.000GB
users 0.000GB
> use mongo_practice;
switched to db mongo_practice
> db.movies.find();
{ "_id" : ObjectId("61045033274a7b4794e15340"), "title" : "Fight Club", "Writer" : "Chuck Palahniuko", "year" : "1999", "actors" : ["Brad Pitt", "Edward Norton"] }
{ "_id" : ObjectId("610450cf274a7b4794e15341"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("61045136274a7b4794e15342"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("610451dd274a7b4794e15343"), "title" : "Inglorious Basterds", "Writer" : "Quention Tarantino", "year" : "2009", "actors" : ["Brad Pitt", "Diane Krug er", "Eli Roth"] }
{ "_id" : ObjectId("610452a0274a7b4794e15344"), "title" : "The Hobbit:An unexpected Journey", "Writer" : "J.R.R Tolkein", "year" : "2012", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("610452ef274a7b4794e15345"), "title" : "The Hobbit:The Desolation of Smaug", "Writer" : "J.R.R Tolkein", "year" : "2013", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("610453c6274a7b4794e15346"), "title" : "The Hobbit:The Battle of the Five Armies", "Writer" : "J.R.R Tolkein", "year" : "2012", "franchise" : "The Ho bbit", "synopsis" : "Bilbo abd 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" }
{ "_id" : ObjectId("610453f9274a7b4794e15347"), "title" : "Pee Wee Herman's Big Adventure" }
{ "_id" : ObjectId("6104540e274a7b4794e15348"), "title" : "Avatar" }
> db.movies.findOne(writer: "Quentin Tarantino")
uncaught exception: SyntaxError: missing) after argument list :
@(shell):1:24
> db.movies.findOne(writer: "Quentin Tarantino");
uncaught exception: SyntaxError: missing) after argument list :
@(shell):1:24
> db.movies.findOne({writer: "Quentin Tarantino"});
null
> db.movies.findOne({"writer": "Quentin Tarantino"});
null
> db.movies.findOne({"writer": "Quentin Tarantino"})
null
> db.movies.find({"writer": "Quentin Tarantino"});
> db.movies.find({"writer": "Quention Tarantino"});
> db.movies.find({"Writer": "Quention Tarantino"});
{ "_id" : ObjectId("610450cf274a7b4794e15341"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("61045136274a7b4794e15342"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("610451dd274a7b4794e15343"), "title" : "Inglorious Basterds", "Writer" : "Quention Tarantino", "year" : "2009", "actors" : ["Brad Pitt", "Diane Krug
ENG
9:16 PM
31-Jul-21

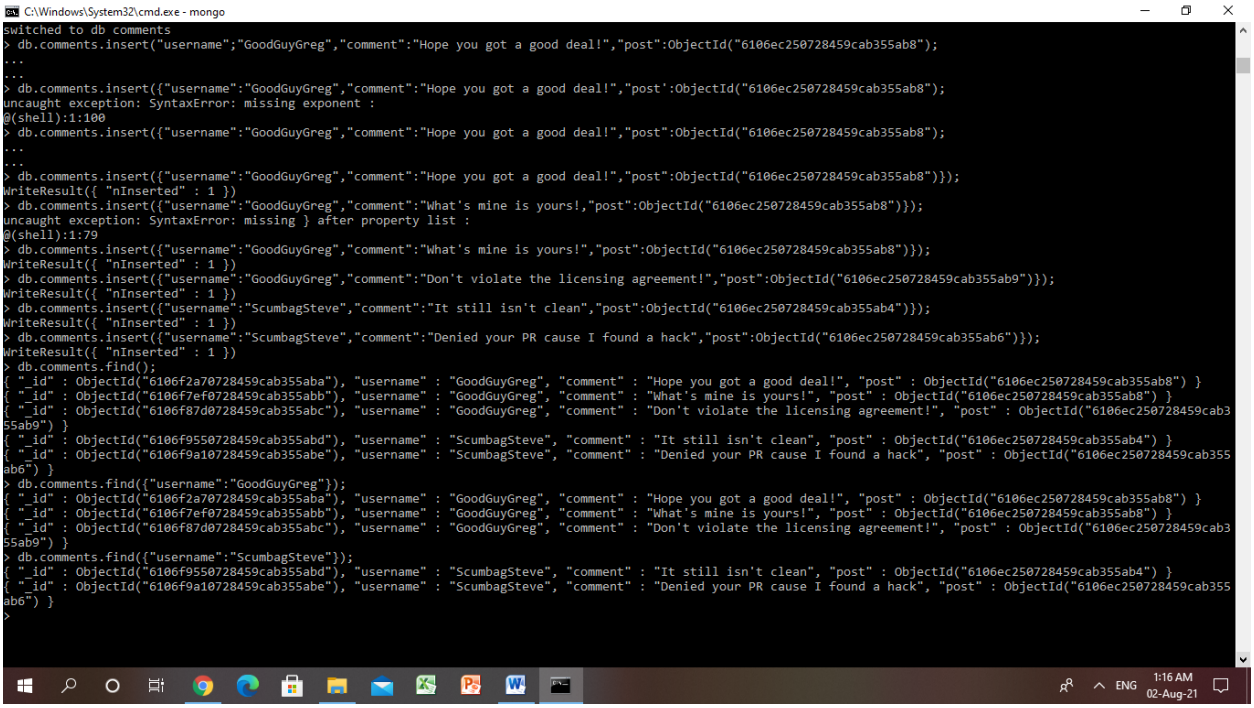

C:\Windows\System32\cmd.exe - mongo
{ "_id" : ObjectId("61045136274a7b4794e15342"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("610451dd274a7b4794e15343"), "title" : "Inglorious Basterds", "Writer" : "Quention Tarantino", "year" : "2009", "actors" : ["Brad Pitt", "Diane Krug er", "Eli Roth"] }
> db.movies.find({"actors" : "Brad Pitt"});
{ "_id" : ObjectId("61045033274a7b4794e15340"), "title" : "Fight Club", "Writer" : "Chuck Palahniuko", "year" : "1999", "actors" : ["Brad Pitt", "Edward Norton"] }
{ "_id" : ObjectId("610451dd274a7b4794e15343"), "title" : "Inglorious Basterds", "Writer" : "Quention Tarantino", "year" : "2009", "actors" : ["Brad Pitt", "Diane Krug er", "Eli Roth"] }
> db.movies.find({"franchise" : "The Hobbit"});
{ "_id" : ObjectId("610452a0274a7b4794e15344"), "title" : "The Hobbit:An unexpected Journey", "Writer" : "J.R.R Tolkein", "year" : "2012", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("610452ef274a7b4794e15345"), "title" : "The Hobbit:The Desolation of Smaug", "Writer" : "J.R.R Tolkein", "year" : "2013", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("610453c6274a7b4794e15346"), "title" : "The Hobbit:The Battle of the Five Armies", "Writer" : "J.R.R Tolkein", "year" : "2012", "franchise" : "The Ho bbit", "synopsis" : "Bilbo abd 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" }
> db.movies.find({$and: [{year: {$gt: 1900}}, {year: {$lt: 2000}}]})
> db.movies.find({$and: [{"year": {$gt: "1900"}}, {"year": {$lt: "2000"}}]});
{ "_id" : ObjectId("61045033274a7b4794e15340"), "title" : "Fight Club", "Writer" : "Chuck Palahniuko", "year" : "1999", "actors" : ["Brad Pitt", "Edward Norton"] }
{ "_id" : ObjectId("610450cf274a7b4794e15341"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("61045136274a7b4794e15342"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
> db.movies.find({$or:[{"year":{$lt:"2000"}}, {"year":{$gt:"2010"}}]})
{ "_id" : ObjectId("61045033274a7b4794e15340"), "title" : "Fight Club", "Writer" : "Chuck Palahniuko", "year" : "1999", "actors" : ["Brad Pitt", "Edward Norton"] }
{ "_id" : ObjectId("610450cf274a7b4794e15341"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("61045136274a7b4794e15342"), "title" : "Pulp Fiction", "Writer" : "Quention Tarantino", "year" : "1994", "actors" : ["John Travolta", "Uma Thurman"] }
{ "_id" : ObjectId("610452a0274a7b4794e15344"), "title" : "The Hobbit:An unexpected Journey", "Writer" : "J.R.R Tolkein", "year" : "2012", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("610452ef274a7b4794e15345"), "title" : "The Hobbit:The Desolation of Smaug", "Writer" : "J.R.R Tolkein", "year" : "2013", "franchise" : "The Hobbit" }
{ "_id" : ObjectId("610453c6274a7b4794e15346"), "title" : "The Hobbit:The Battle of the Five Armies", "Writer" : "J.R.R Tolkein", "year" : "2012", "franchise" : "The Ho bbit", "synopsis" : "Bilbo abd 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" }
>
ENG
9:16 PM
31-Jul-21


C:\Windows\System32\cmd.exe - mongo
> 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."});
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.movies.update({"title": "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."});
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.movies.update({"title": "Pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.movies.find({$text: {$search: "Bilbo"}});
Error: error: {
"ok" : 0,
"errmsg" : "text index required for $text query",
"code" : 27,
"codeName" : "IndexNotFound"
}
> db.movies.find({$"text": {$search: "Bilbo"}});
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:17
> db.movies.find({synopsis:/Bilbo/g}).pretty();
{
"_id" : ObjectId("610453c6274a7b4794e15346"),
"title" : "The Hobbit:The Battle of the Five Armies",
"Writer" : "J.R.R Tolkein",
"year" : "2012",
"franchise" : "The Hobbit",
"synopsis" : "Bilbo abd Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness"
}
> db.movies.find({synopsis:/Gandalf/g}).pretty();
> db.movies.find({$and:[{"synopsis":/Bilbo/g},{"synopsis":{$not:/Gandalf/g}}]}).pretty();
{
"_id" : ObjectId("610453c6274a7b4794e15346"),
"title" : "The Hobbit:The Battle of the Five Armies",
"Writer" : "J.R.R Tolkein",
"year" : "2012",
"franchise" : "The Hobbit",
"synopsis" : "Bilbo abd 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"
}
> db.movies.find({"synopsis":/(dwarves|hobbit)/g}).pretty();
> db.movies.find({"synopsis":/(dwarves|Hobbit)/g}).pretty();
> db.movies.find({"synopsis":/(gold.*dragon|dragon.*gold)/g}).pretty();
>
ENG
1:56 PM
01-Aug-21

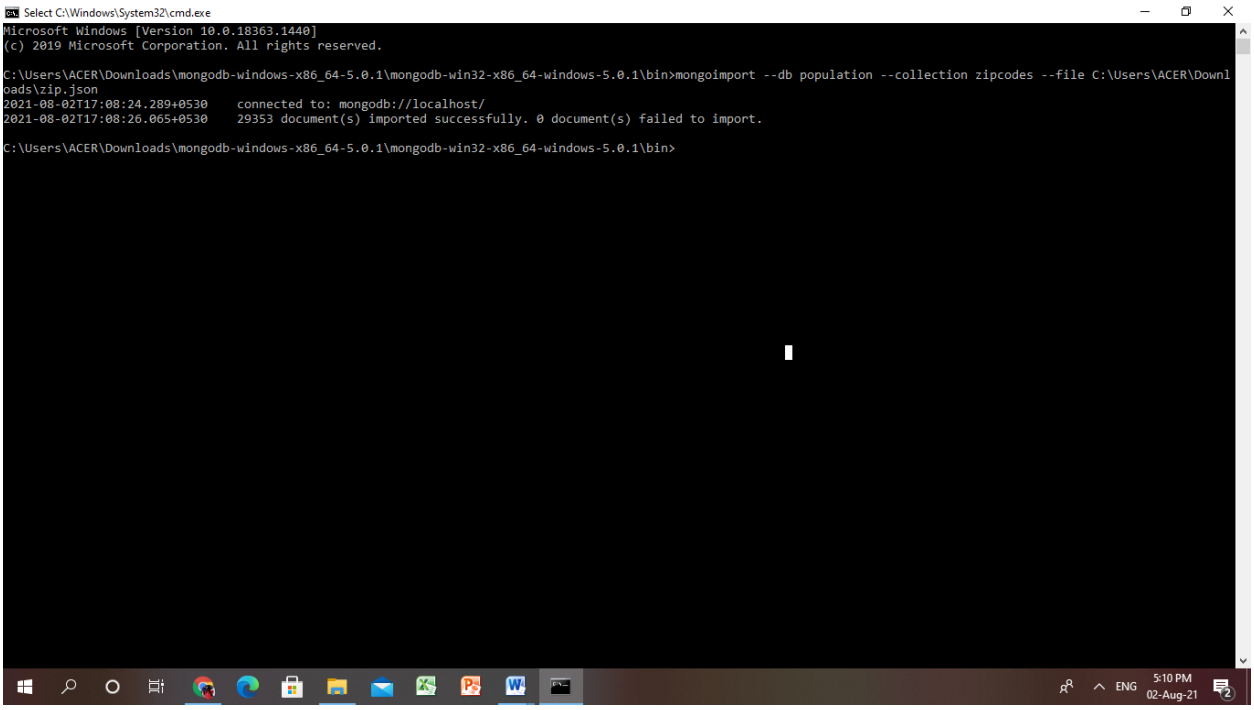

C:\Windows\System32\cmd.exe - mongo
@(shell):1:161
> db.posts.insertMany([
... {
... "username":"GoodGuyGreg",
... "title":"Passes out as party",
... "body":"Wakes up early and cleans house"
... },
... {
... "username":"GoodGuyGrey",
... "title":"Steals your identity",
... "body":"Raises your credit score"
... },
... {
... "username":"GoodGuyGreg",
... "title":"Reports a bug in your code",
... "body":'Sends you a Pull Request"
uncaught exception: SyntaxError: '' literal not terminated before end of script :
@(shell):15:33
> db.posts.insertMany([{ "username":"GoodGuyGreg", "title":"Passes out as party", "body":"Wakes up early and cleans house" }, { "username":"GoodGuyGrey", "title":"Steals your identity", "body":"Raises your credit score" }, { "username":"GoodGuyGreg", "title":"Reports a bug in your code", "body":'Sends you a Pull Request"},{"username":"ScumbagSteve","title":"Borrows something","body":"Sells it"},{"username":'ScumbagSteve","title":"Borrows Everything","body":"The end"},{"username":"SCumbagSteve","title":"Forks your repo on github","body":"Sets to private"}]);
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:410
> 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 a bug in your code","body":"Sends you a Pull Request"});
WriteResult({ "nInserted" : 1 })
> db.posts.insert({"username":"ScumbagSteve","title":"Borrows something","body":"Sells it"});
WriteResult({ "nInserted" : 1 })
> db.posts.insert({"username":"ScumbagSteve","title":"Borrows everything","body":"The end"});
WriteResult({ "nInserted" : 1 })
> db.posts.insert({"username":"ScumbagSteve","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":objectId("5f44d3a148197d7749864def"});
...
...
> db.comments.insert({"username":"GoodGuyGreg","comment":'Hope you got a good deal", "post":objectId("5f44d3a148197d7749864def")});
uncaught exception: SyntaxError: '' literal not terminated before end of script :
@(shell):1:129
>
ENG
12:27 AM
02-Aug-21


C:\Windows\System32\cmd.exe - mongo
is your identity", "body":"Raises your credit score" }, { "username":"GoodGuyGreg", "title":"Reports a bug in your code", "body":'Sends you a Pull Request"},{"username"
:"ScumbagSteve","title":"Borrows something","body":"Sells it"},{"username":'ScumbagSteve","title":"Borrows Everything","body":"The end"},{"username":"SCumbagSteve","tit
le":"Forks your repo on github","body":"Sets to private"}]);
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:410
> 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 a bug in your code","body":"Sends you a Pull Request"});
WriteResult({ "nInserted" : 1 })
> db.posts.insert({"username":"ScumbagSteve","title":"Borrows something","body":"Sells it"});
WriteResult({ "nInserted" : 1 })
> db.posts.insert({"username":"ScumbagSteve","title":"Borrows everything","body":"The end"});
WriteResult({ "nInserted" : 1 })
> db.posts.insert({"username":"ScumbagSteve","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":objectId("5f44d3a148197d7749864def"});
...
...
> db.comments.insert({"username":"GoodGuyGreg","comment":'Hope you got a good deal", "post":objectId("5f44d3a148197d7749864def")});
uncaught exception: SyntaxError: '' literal not terminated before end of script :
@(shell):1:129
> db.users.find()
> use users;
switched to db users
> db.users.find()
{ "_id" : ObjectId("61065e5f0728459cab355ab2"), "username" : "GoodGuyGreg", "first_name" : "Good Guy", "last_name" : "Greg" }
{ "_id" : ObjectId("61065e5f0728459cab355ab3"), "username" : "ScumbagSteve", "full_name" : { "first" : "Scumbag", "last" : "steve" } }
> use posts;
switched to db posts
> db.posts.find();
{ "_id" : ObjectId("6106eb4b0728459cab355ab4"), "username" : "GoodGuyGreg", "title" : "Passes out at party", "body" : "Wakes up early and cleans house" }
{ "_id" : ObjectId("6106eb970728459cab355ab5"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("6106ebc40728459cab355ab6"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a Pull Request" }
{ "_id" : ObjectId("6106ec060728459cab355ab7"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }
{ "_id" : ObjectId("6106ec250728459cab355ab8"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }
{ "_id" : ObjectId("6106ec4e0728459cab355ab9"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }
> db.posts.find({"username":"GoodGuyGreg"});
{ "_id" : ObjectId("6106eb4b0728459cab355ab4"), "username" : "GoodGuyGreg", "title" : "Passes out at party", "body" : "Wakes up early and cleans house" }
{ "_id" : ObjectId("6106eb970728459cab355ab5"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("6106ebc40728459cab355ab6"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a Pull Request" }
>
ENG
12:34 AM
02-Aug-21

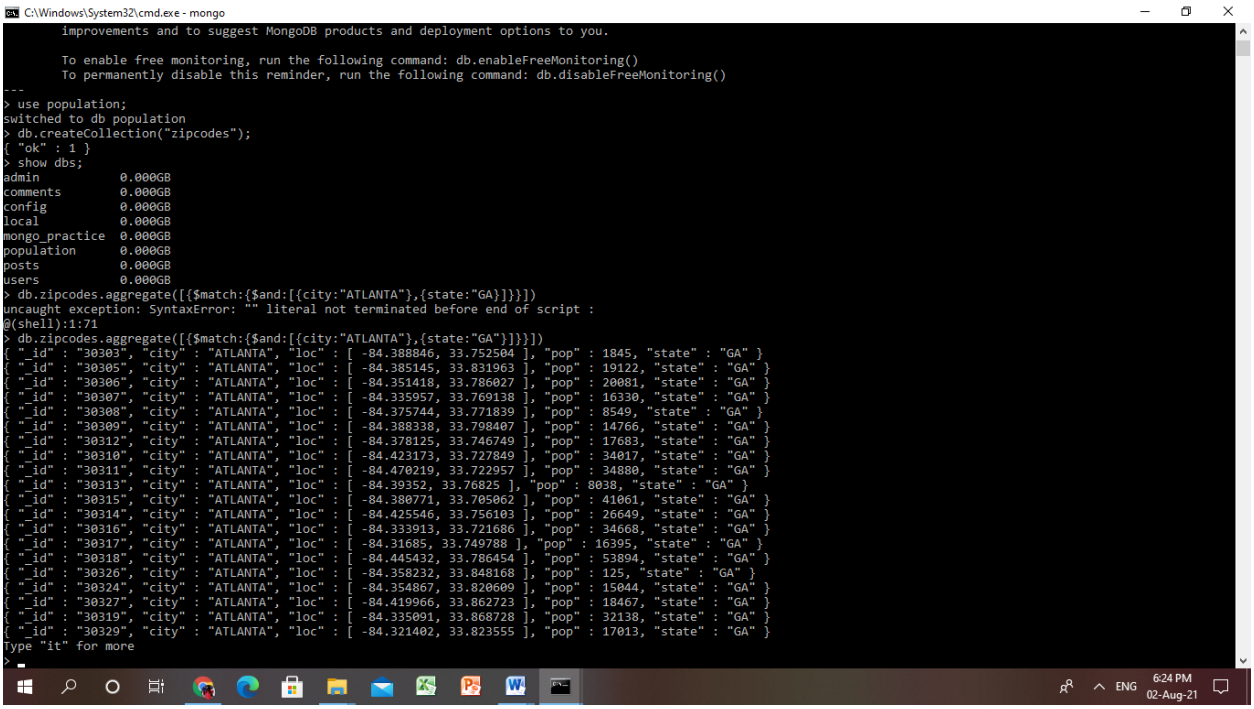

C:\Windows\System32\cmd.exe - mongo
{ "_id" : ObjectId("6106eb970728459cab355ab5"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("6106ebc40728459cab355ab6"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a Pull Request" }
> db.posts.find({"username":"ScumbagSteve"});
{ "_id" : ObjectId("6106ec060728459cab355ab7"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }
{ "_id" : ObjectId("6106ec250728459cab355ab8"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }
{ "_id" : ObjectId("6106ec4e0728459cab355ab9"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }
> use comments;
switched to db comments
> db.comments.insert("username";"GoodGuyGreg","comment":"Hope you got a good deal!","post":ObjectId("6106ec250728459cab355ab8");
...
...
> db.comments.insert({"username":"GoodGuyGreg","comment":"Hope you got a good deal!","post':ObjectId("6106ec250728459cab355ab8");
uncaught exception: SyntaxError: missing exponent :
@(shell):1:100
> db.comments.insert({"username":"GoodGuyGreg","comment":"Hope you got a good deal!","post":ObjectId("6106ec250728459cab355ab8");
...
...
> db.comments.insert({"username":"GoodGuyGreg","comment":"Hope you got a good deal!","post":ObjectId("6106ec250728459cab355ab8")});
WriteResult({ "nInserted" : 1 })
> db.comments.insert({"username":"GoodGuyGreg","comment":"What's mine is yours!,"post":ObjectId("6106ec250728459cab355ab8")});
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:79
> db.comments.insert({"username":"GoodGuyGreg","comment":"What's mine is yours!","post":ObjectId("6106ec250728459cab355ab8")});
WriteResult({ "nInserted" : 1 })
> db.comments.insert({"username":"GoodGuyGreg","comment":"Don't violate the licensing agreement!","post":ObjectId("6106ec250728459cab355ab9")});
WriteResult({ "nInserted" : 1 })
> db.comments.insert({"username":"ScumbagSteve","comment":"It still isn't clean","post":ObjectId("6106ec250728459cab355ab4")});
WriteResult({ "nInserted" : 1 })
> db.comments.insert({"username":"ScumbagSteve","comment":"Denied your PR cause I found a hack","post":ObjectId("6106ec250728459cab355ab6")});
WriteResult({ "nInserted" : 1 })
> db.comments.find();
{ "_id" : ObjectId("6106f2a70728459cab355aba"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : ObjectId("6106ec250728459cab355ab8") }
{ "_id" : ObjectId("6106f7ef0728459cab355abb"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!", "post" : ObjectId("6106ec250728459cab355ab8") }
{ "_id" : ObjectId("6106f87d0728459cab355abc"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : ObjectId("6106ec250728459cab3
55ab9") }
{ "_id" : ObjectId("6106f9550728459cab355abd"), "username" : "ScumbagSteve", "comment" : "It still isn't clean", "post" : ObjectId("6106ec250728459cab355ab4") }
{ "_id" : ObjectId("6106f9a10728459cab355abe"), "username" : "ScumbagSteve", "comment" : "Denied your PR cause I found a hack", "post" : ObjectId("6106ec250728459cab355
ab6") }
> db.comments.find({"username":"GoodGuyGreg"});
{ "_id" : ObjectId("6106f2a70728459cab355aba"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : ObjectId("6106ec250728459cab355ab8") }
{ "_id" : ObjectId("6106f7ef0728459cab355abb"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!", "post" : ObjectId("6106ec250728459cab355ab8") }
{ "_id" : ObjectId("6106f87d0728459cab355abc"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : ObjectId("6106ec250728459cab3
55ab9") }
> db.comments.find({"username":"ScumbagSteve"});
ENG
1:16 AM
02-Aug-21



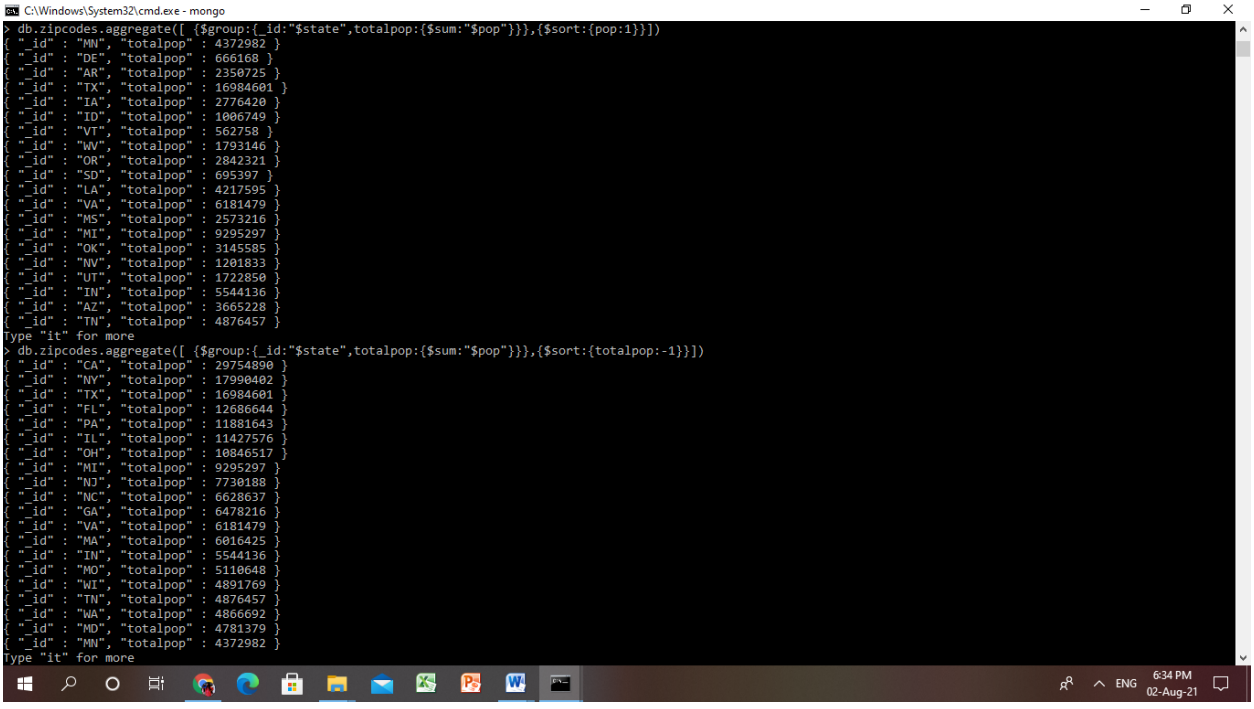
Assignment 2:



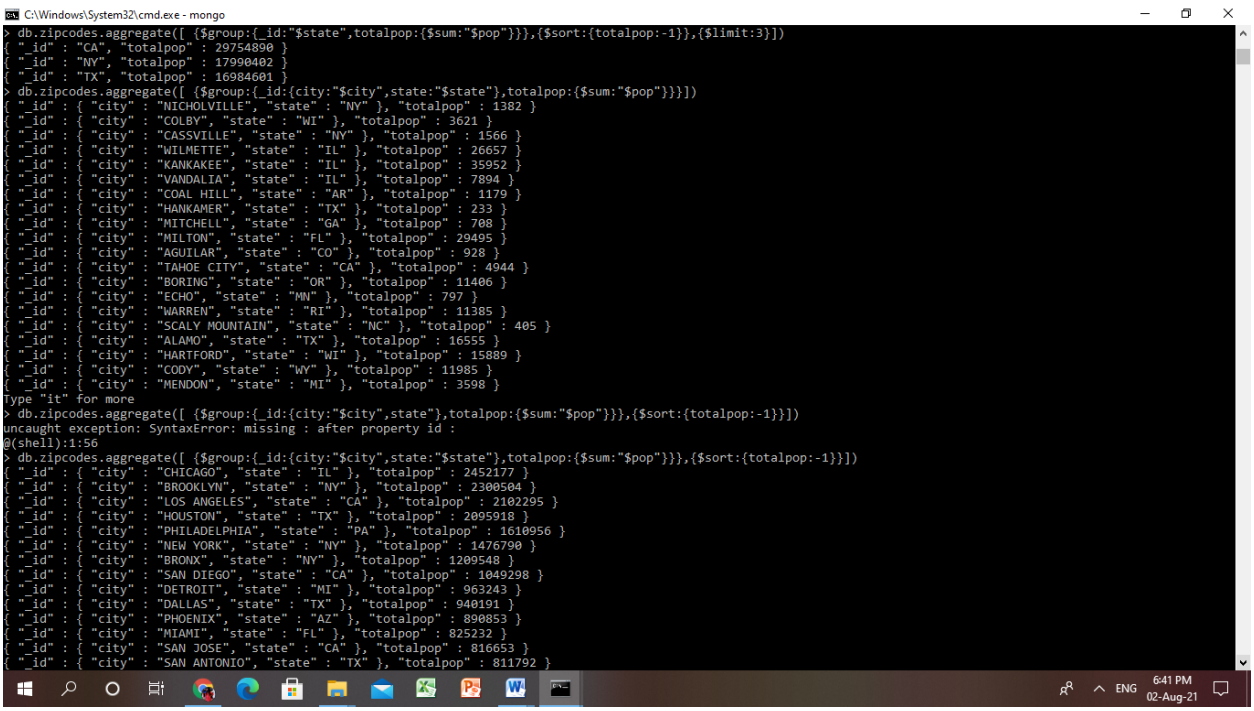
1&2)


C:\Windows\System32\cmd.exe - mongo
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()
> use population;
switched to db population
> db.createCollection("zipcodes");
{ "ok" : 1 }
> show dbs;
admin 0.000GB
comments 0.000GB
config 0.000GB
local 0.000GB
mongo_practice 0.000GB
population 0.000GB
posts 0.000GB
users 0.000GB
> db.zipcodes.aggregate([{$match:{$and:[{city:"ATLANTA"},{state:"GA}]}}])
uncaught exception: SyntaxError: "" literal not terminated before end of script :
@(shell):1:71
> db.zipcodes.aggregate([{$match:{$and:[{city:"ATLANTA"},{state:"GA"}]}}])
{ "_id" : "30303", "city" : "ATLANTA", "loc" : [-84.388846, 33.752504], "pop" : 1845, "state" : "GA" }
{ "_id" : "30305", "city" : "ATLANTA", "loc" : [-84.385145, 33.831963], "pop" : 19122, "state" : "GA" }
{ "_id" : "30306", "city" : "ATLANTA", "loc" : [-84.351418, 33.786027], "pop" : 20081, "state" : "GA" }
{ "_id" : "30307", "city" : "ATLANTA", "loc" : [-84.335957, 33.769138], "pop" : 16330, "state" : "GA" }
{ "_id" : "30308", "city" : "ATLANTA", "loc" : [-84.375744, 33.771839], "pop" : 8549, "state" : "GA" }
{ "_id" : "30309", "city" : "ATLANTA", "loc" : [-84.388338, 33.798407], "pop" : 14766, "state" : "GA" }
{ "_id" : "30312", "city" : "ATLANTA", "loc" : [-84.378125, 33.746749], "pop" : 17683, "state" : "GA" }
{ "_id" : "30310", "city" : "ATLANTA", "loc" : [-84.423173, 33.727849], "pop" : 34017, "state" : "GA" }
{ "_id" : "30311", "city" : "ATLANTA", "loc" : [-84.470219, 33.722957], "pop" : 34880, "state" : "GA" }
{ "_id" : "30313", "city" : "ATLANTA", "loc" : [-84.39352, 33.76825], "pop" : 8038, "state" : "GA" }
{ "_id" : "30315", "city" : "ATLANTA", "loc" : [-84.380771, 33.705062], "pop" : 41061, "state" : "GA" }
{ "_id" : "30314", "city" : "ATLANTA", "loc" : [-84.425546, 33.756103], "pop" : 26649, "state" : "GA" }
{ "_id" : "30316", "city" : "ATLANTA", "loc" : [-84.333913, 33.721686], "pop" : 34668, "state" : "GA" }
{ "_id" : "30317", "city" : "ATLANTA", "loc" : [-84.31685, 33.749788], "pop" : 16395, "state" : "GA" }
{ "_id" : "30318", "city" : "ATLANTA", "loc" : [-84.445432, 33.786454], "pop" : 53894, "state" : "GA" }
{ "_id" : "30326", "city" : "ATLANTA", "loc" : [-84.358232, 33.848168], "pop" : 125, "state" : "GA" }
{ "_id" : "30324", "city" : "ATLANTA", "loc" : [-84.354867, 33.820609], "pop" : 15044, "state" : "GA" }
{ "_id" : "30327", "city" : "ATLANTA", "loc" : [-84.419966, 33.862723], "pop" : 18467, "state" : "GA" }
{ "_id" : "30319", "city" : "ATLANTA", "loc" : [-84.335091, 33.868728], "pop" : 32138, "state" : "GA" }
{ "_id" : "30329", "city" : "ATLANTA", "loc" : [-84.321402, 33.823555], "pop" : 17013, "state" : "GA" }
Type "it" for more
> _
ENG 6:24 PM 02-Aug-21

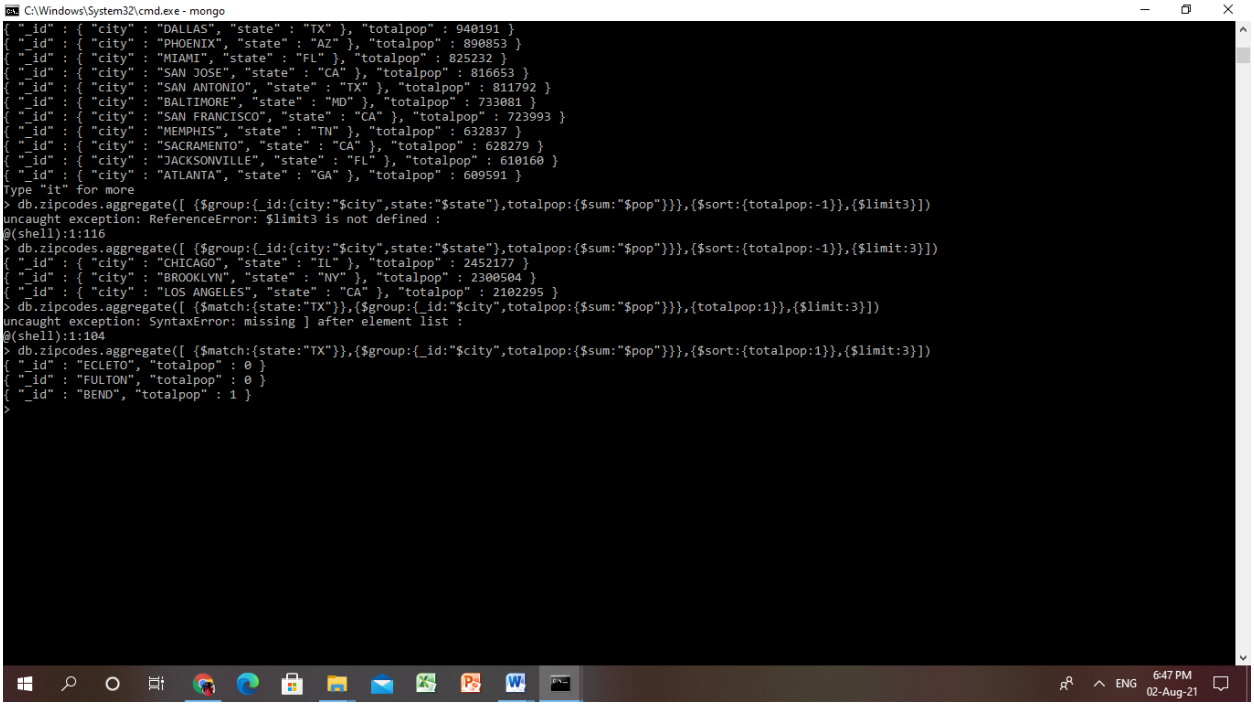
5&6)


C:\Windows\System32\cmd.exe - mongo
> db.zipcodes.aggregate([{$group:{_id:"$state",totalpop:{$sum:"$pop"}}},{$sort:{pop:1}}])
{ "_id" : "MN", "totalpop" : 4372982 }
{ "_id" : "DE", "totalpop" : 666168 }
{ "_id" : "AR", "totalpop" : 2350725 }
{ "_id" : "TX", "totalpop" : 16984601 }
{ "_id" : "IA", "totalpop" : 2776420 }
{ "_id" : "ID", "totalpop" : 1006749 }
{ "_id" : "VT", "totalpop" : 562758 }
{ "_id" : "WV", "totalpop" : 1793146 }
{ "_id" : "OR", "totalpop" : 2842321 }
{ "_id" : "SD", "totalpop" : 695397 }
{ "_id" : "LA", "totalpop" : 4217595 }
{ "_id" : "VA", "totalpop" : 6181479 }
{ "_id" : "MS", "totalpop" : 2573216 }
{ "_id" : "MI", "totalpop" : 9295297 }
{ "_id" : "OK", "totalpop" : 3145585 }
{ "_id" : "NV", "totalpop" : 1201833 }
{ "_id" : "UT", "totalpop" : 1722850 }
{ "_id" : "IN", "totalpop" : 5544136 }
{ "_id" : "AZ", "totalpop" : 3665228 }
{ "_id" : "TN", "totalpop" : 4876457 }
Type "it" for more
> db.zipcodes.aggregate([{$group:{_id:"$state",totalpop:{$sum:"$pop"}}},{$sort:{totalpop:-1}}])
{ "_id" : "CA", "totalpop" : 29754890 }
{ "_id" : "NY", "totalpop" : 17990402 }
{ "_id" : "TX", "totalpop" : 16984601 }
{ "_id" : "FL", "totalpop" : 12686644 }
{ "_id" : "PA", "totalpop" : 11881643 }
{ "_id" : "IL", "totalpop" : 11427576 }
{ "_id" : "OH", "totalpop" : 10846517 }
{ "_id" : "MI", "totalpop" : 9295297 }
{ "_id" : "NJ", "totalpop" : 7730188 }
{ "_id" : "NC", "totalpop" : 6628637 }
{ "_id" : "GA", "totalpop" : 6478216 }
{ "_id" : "VA", "totalpop" : 6181479 }
{ "_id" : "MA", "totalpop" : 6016425 }
{ "_id" : "IN", "totalpop" : 5544136 }
{ "_id" : "MO", "totalpop" : 5110648 }
{ "_id" : "WI", "totalpop" : 4891769 }
{ "_id" : "TN", "totalpop" : 4876457 }
{ "_id" : "WA", "totalpop" : 4866692 }
{ "_id" : "MD", "totalpop" : 4781379 }
{ "_id" : "MN", "totalpop" : 4372982 }
Type "it" for more
ENG 6:34 PM 02-Aug-21

7,8,9)

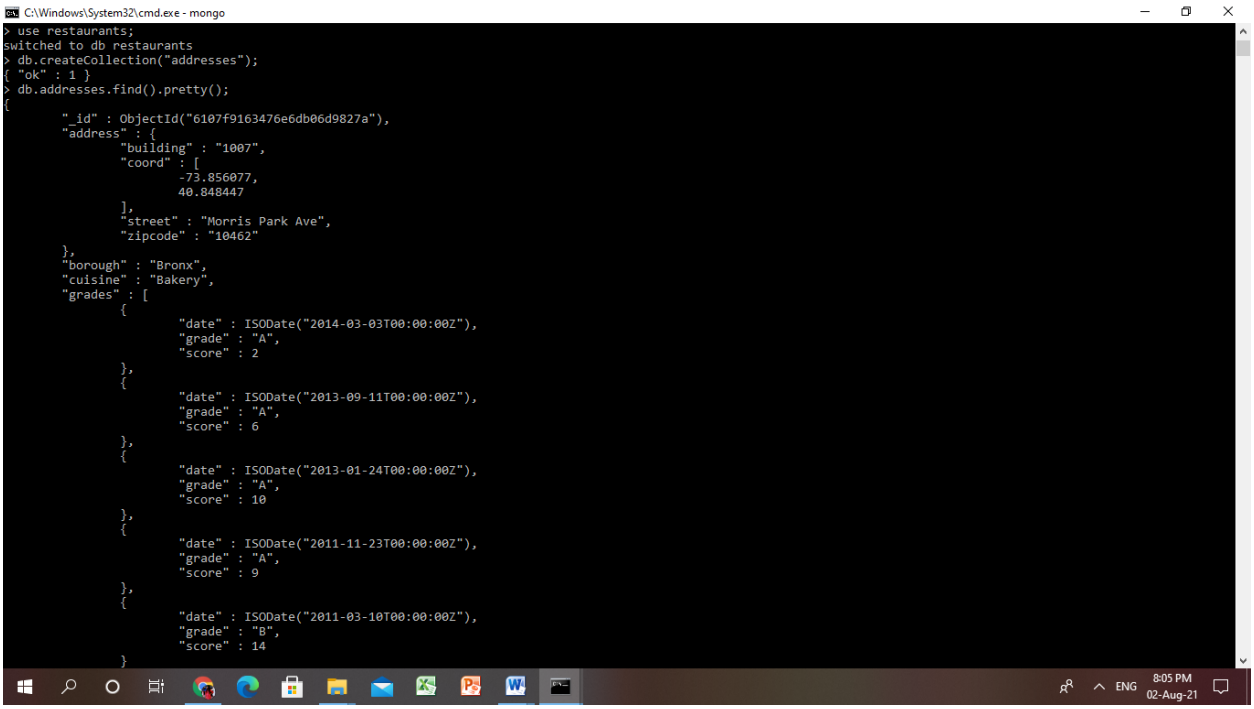

C:\Windows\System32\cmd.exe - mongo
> db.zipcodes.aggregate([{$group:{_id:"$state",totalpop:{$sum:"$pop"}}},{$sort:{totalpop:-1}},{$limit:3}])
{ "_id" : "CA", "totalpop" : 29754890 }
{ "_id" : "NY", "totalpop" : 17990402 }
{ "_id" : "TX", "totalpop" : 16984601 }
> db.zipcodes.aggregate([{$group:{_id:{city:"$city",state:"$state"},totalpop:{$sum:"$pop"}}}])
{ "_id" : { "city" : "NICHOLVILLE", "state" : "NY" }, "totalpop" : 1382 }
{ "_id" : { "city" : "COLBY", "state" : "WI" }, "totalpop" : 3621 }
{ "_id" : { "city" : "CASSVILLE", "state" : "NY" }, "totalpop" : 1566 }
{ "_id" : { "city" : "WILMETTE", "state" : "IL" }, "totalpop" : 26657 }
{ "_id" : { "city" : "KANKAKEE", "state" : "IL" }, "totalpop" : 35952 }
{ "_id" : { "city" : "VANDALIA", "state" : "IL" }, "totalpop" : 7894 }
{ "_id" : { "city" : "COAL HILL", "state" : "AR" }, "totalpop" : 1179 }
{ "_id" : { "city" : "HANKAMER", "state" : "TX" }, "totalpop" : 233 }
{ "_id" : { "city" : "MITCHELL", "state" : "GA" }, "totalpop" : 708 }
{ "_id" : { "city" : "MILTON", "state" : "FL" }, "totalpop" : 29495 }
{ "_id" : { "city" : "AGUILAR", "state" : "CO" }, "totalpop" : 928 }
{ "_id" : { "city" : "TAHOE CITY", "state" : "CA" }, "totalpop" : 4944 }
{ "_id" : { "city" : "BORING", "state" : "OR" }, "totalpop" : 11406 }
{ "_id" : { "city" : "ECHO", "state" : "MN" }, "totalpop" : 797 }
{ "_id" : { "city" : "WARREN", "state" : "RI" }, "totalpop" : 11385 }
{ "_id" : { "city" : "SCALY MOUNTAIN", "state" : "NC" }, "totalpop" : 405 }
{ "_id" : { "city" : "ALAMO", "state" : "TX" }, "totalpop" : 16555 }
{ "_id" : { "city" : "HARTFORD", "state" : "WI" }, "totalpop" : 15889 }
{ "_id" : { "city" : "CODY", "state" : "WY" }, "totalpop" : 11985 }
{ "_id" : { "city" : "MENDON", "state" : "MI" }, "totalpop" : 3598 }
Type "it" for more
> db.zipcodes.aggregate([{$group:{_id:{city:"$city",state"},totalpop:{$sum:"$pop"}}},{$sort:{totalpop:-1}}])
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:56
> db.zipcodes.aggregate([{$group:{_id:{city:"$city",state:"$state"},totalpop:{$sum:"$pop"}}},{$sort:{totalpop:-1}}])
{ "_id" : { "city" : "CHICAGO", "state" : "IL" }, "totalpop" : 2452177 }
{ "_id" : { "city" : "BROOKLYN", "state" : "NY" }, "totalpop" : 2300504 }
{ "_id" : { "city" : "LOS ANGELES", "state" : "CA" }, "totalpop" : 2102295 }
{ "_id" : { "city" : "HOUSTON", "state" : "TX" }, "totalpop" : 2095918 }
{ "_id" : { "city" : "PHILADELPHIA", "state" : "PA" }, "totalpop" : 1610956 }
{ "_id" : { "city" : "NEW YORK", "state" : "NY" }, "totalpop" : 1476790 }
{ "_id" : { "city" : "BRONX", "state" : "NY" }, "totalpop" : 1209548 }
{ "_id" : { "city" : "SAN DIEGO", "state" : "CA" }, "totalpop" : 1049298 }
{ "_id" : { "city" : "DETROIT", "state" : "MI" }, "totalpop" : 963243 }
{ "_id" : { "city" : "DALLAS", "state" : "TX" }, "totalpop" : 940191 }
{ "_id" : { "city" : "PHOENIX", "state" : "AZ" }, "totalpop" : 890853 }
{ "_id" : { "city" : "MIAMI", "state" : "FL" }, "totalpop" : 825232 }
{ "_id" : { "city" : "SAN JOSE", "state" : "CA" }, "totalpop" : 816653 }
{ "_id" : { "city" : "SAN ANTONIO", "state" : "TX" }, "totalpop" : 811792 }
ENG
6:41 PM
02-Aug-21

10&11)

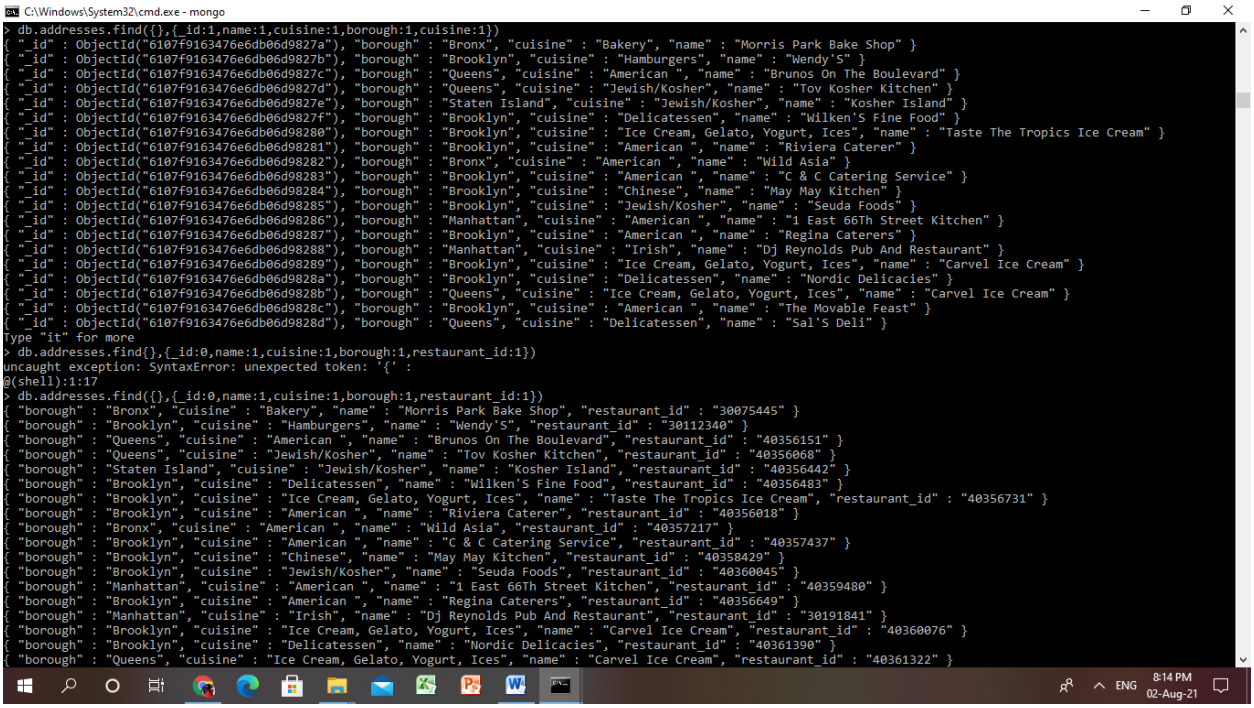

C:\Windows\System32\cmd.exe - mongo
{ "_id" : { "city" : "DALLAS", "state" : "TX" }, "totalpop" : 940191 }
{ "_id" : { "city" : "PHOENIX", "state" : "AZ" }, "totalpop" : 890853 }
{ "_id" : { "city" : "MIAMI", "state" : "FL" }, "totalpop" : 825232 }
{ "_id" : { "city" : "SAN JOSE", "state" : "CA" }, "totalpop" : 816653 }
{ "_id" : { "city" : "SAN ANTONIO", "state" : "TX" }, "totalpop" : 811792 }
{ "_id" : { "city" : "BALTIMORE", "state" : "MD" }, "totalpop" : 733081 }
{ "_id" : { "city" : "SAN FRANCISCO", "state" : "CA" }, "totalpop" : 723993 }
{ "_id" : { "city" : "MEMPHIS", "state" : "TN" }, "totalpop" : 632837 }
{ "_id" : { "city" : "SACRAMENTO", "state" : "CA" }, "totalpop" : 628279 }
{ "_id" : { "city" : "JACKSONVILLE", "state" : "FL" }, "totalpop" : 610160 }
{ "_id" : { "city" : "ATLANTA", "state" : "GA" }, "totalpop" : 609591 }
Type "it" for more
> db.zipcodes.aggregate([{$group:{_id:{city:"$city",state:"$state"},totalpop:{$sum:"$pop"}}},{$sort:{totalpop:-1}},{$limit3}])
uncaught exception: ReferenceError: $limit3 is not defined :
@(shell):1:116
> db.zipcodes.aggregate([{$group:{_id:{city:"$city",state:"$state"},totalpop:{$sum:"$pop"}}},{$sort:{totalpop:-1}},{$limit:3}])
{ "_id" : { "city" : "CHICAGO", "state" : "IL" }, "totalpop" : 2452177 }
{ "_id" : { "city" : "BROOKLYN", "state" : "NY" }, "totalpop" : 2300504 }
{ "_id" : { "city" : "LOS ANGELES", "state" : "CA" }, "totalpop" : 2102295 }
> db.zipcodes.aggregate([{$match:{state:"TX"}},{$group:{_id:"$city",totalpop:{$sum:"$pop"}}},{totalpop:1}},{$limit:3}])
uncaught exception: SyntaxError: missing] after element list :
@(shell):1:104
> db.zipcodes.aggregate([{$match:{state:"TX"}},{$group:{_id:"$city",totalpop:{$sum:"$pop"}}},{$sort:{totalpop:1}},{$limit:3}])
{ "_id" : "ECLETO", "totalpop" : 0 }
{ "_id" : "FULTON", "totalpop" : 0 }
{ "_id" : "BEND", "totalpop" : 1 }
>
ENG
6:47 PM
02-Aug-21

Assignment 3

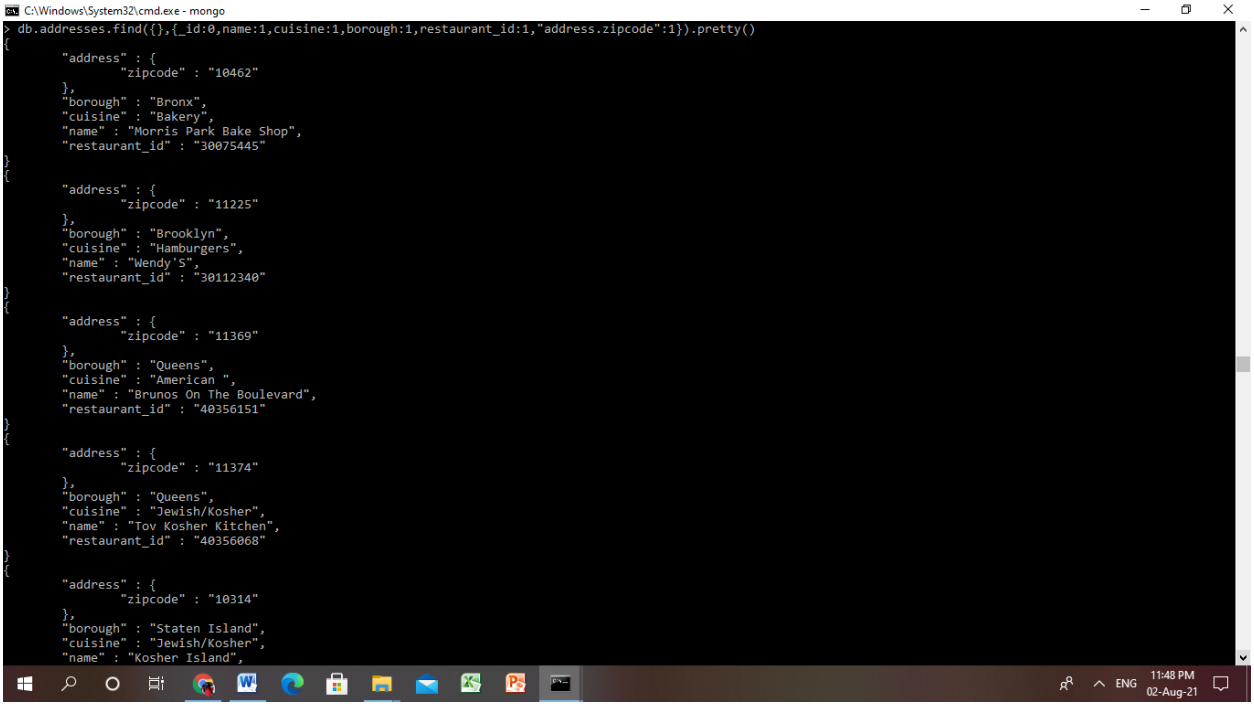
1)



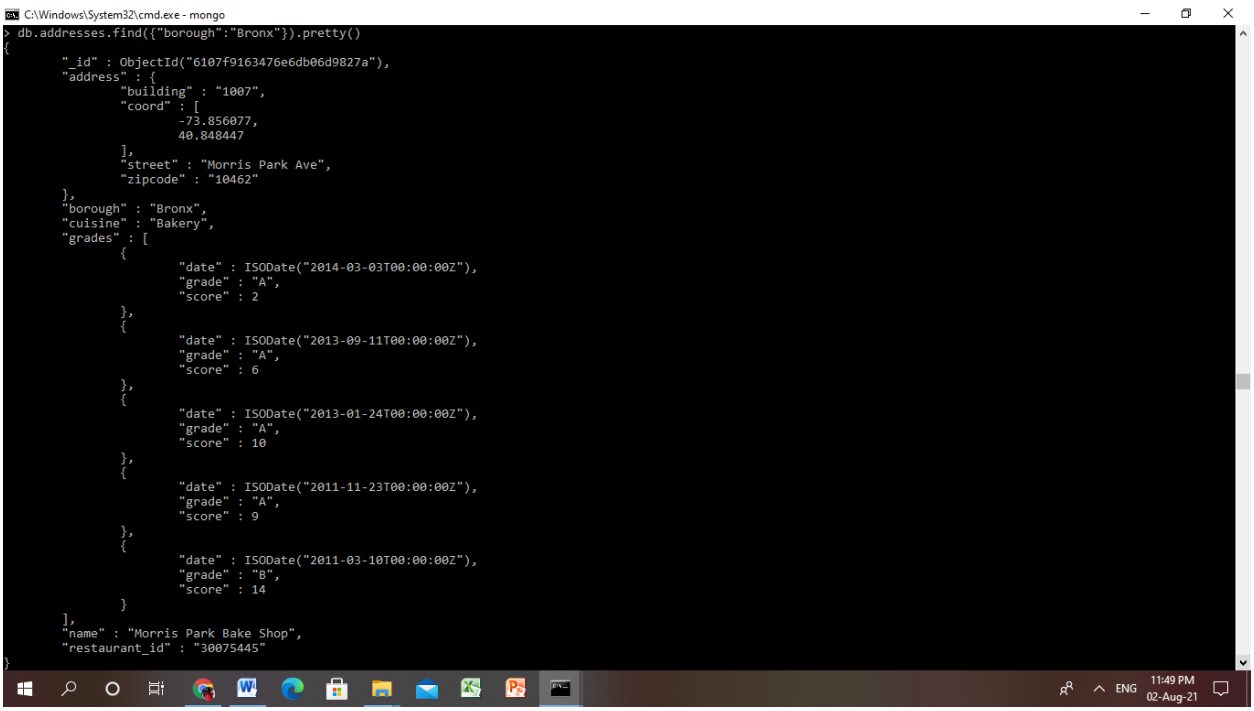
2&3)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({},{_id:1,name:1,cuisine:1,borough:1,cuisine:1})
{ "_id" : ObjectId("6107f9163476e6db06d9827a"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop" }
{ "_id" : ObjectId("6107f9163476e6db06d9827b"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S" }
{ "_id" : ObjectId("6107f9163476e6db06d9827c"), "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard" }
{ "_id" : ObjectId("6107f9163476e6db06d9827d"), "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen" }
{ "_id" : ObjectId("6107f9163476e6db06d9827e"), "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island" }
{ "_id" : ObjectId("6107f9163476e6db06d9827f"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food" }
{ "_id" : ObjectId("6107f9163476e6db06d98280"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream" }
{ "_id" : ObjectId("6107f9163476e6db06d98281"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer" }
{ "_id" : ObjectId("6107f9163476e6db06d98282"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia" }
{ "_id" : ObjectId("6107f9163476e6db06d98283"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service" }
{ "_id" : ObjectId("6107f9163476e6db06d98284"), "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen" }
{ "_id" : ObjectId("6107f9163476e6db06d98285"), "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods" }
{ "_id" : ObjectId("6107f9163476e6db06d98286"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen" }
{ "_id" : ObjectId("6107f9163476e6db06d98287"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers" }
{ "_id" : ObjectId("6107f9163476e6db06d98288"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant" }
{ "_id" : ObjectId("6107f9163476e6db06d98289"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream" }
{ "_id" : ObjectId("6107f9163476e6db06d9828a"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies" }
{ "_id" : ObjectId("6107f9163476e6db06d9828b"), "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream" }
{ "_id" : ObjectId("6107f9163476e6db06d9828c"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast" }
{ "_id" : ObjectId("6107f9163476e6db06d9828d"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli" }
Type "it" for more
> db.addresses.find({},{_id:0,name:1,cuisine:1,borough:1,restaurant_id:1})
uncaught exception: SyntaxError: unexpected token: '{' :
@(shell):1:17
> db.addresses.find({},{_id:0,name:1,cuisine:1,borough:1,restaurant_id:1})
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "40356731" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
ENG
8:14 PM
02-Aug-21

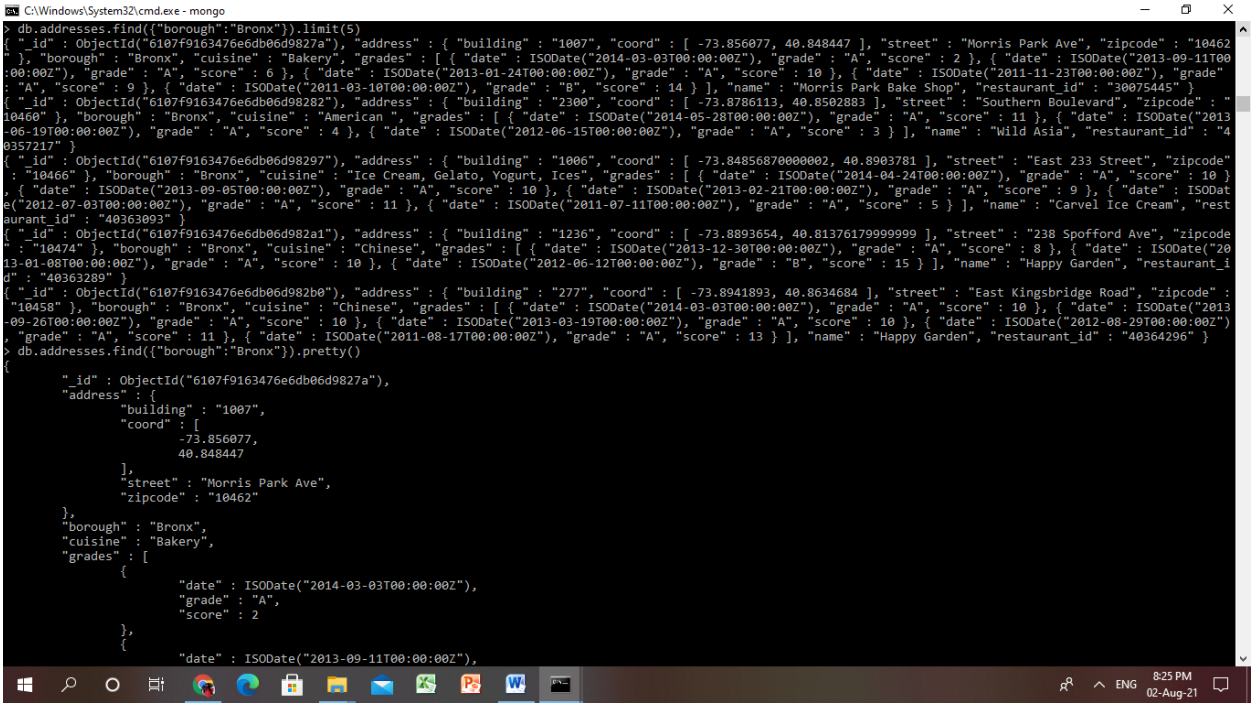
4)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({},{_id:0,name:1,cuisine:1,borough:1,restaurant_id:1,"address.zipcode":1}).pretty()
{
"address" : {
"zipcode" : "10462"
},
"borough" : "Bronx",
"cuisine" : "Bakery",
"name" : "Morris Park Bake Shop",
"restaurant_id" : "30075445"
}
{
"address" : {
"zipcode" : "11225"
},
"borough" : "Brooklyn",
"cuisine" : "Hamburgers",
"name" : "Wendy'S",
"restaurant_id" : "30112340"
}
{
"address" : {
"zipcode" : "11369"
},
"borough" : "Queens",
"cuisine" : "American ",
"name" : "Brunos On The Boulevard",
"restaurant_id" : "40356151"
}
{
"address" : {
"zipcode" : "11374"
},
"borough" : "Queens",
"cuisine" : "Jewish/Kosher",
"name" : "Tov Kosher Kitchen",
"restaurant_id" : "40356068"
}
{
"address" : {
"zipcode" : "10314"
},
"borough" : "Staten Island",
"cuisine" : "Jewish/Kosher",
"name" : "Kosher Island",
ENG
11:48 PM
02-Aug-21

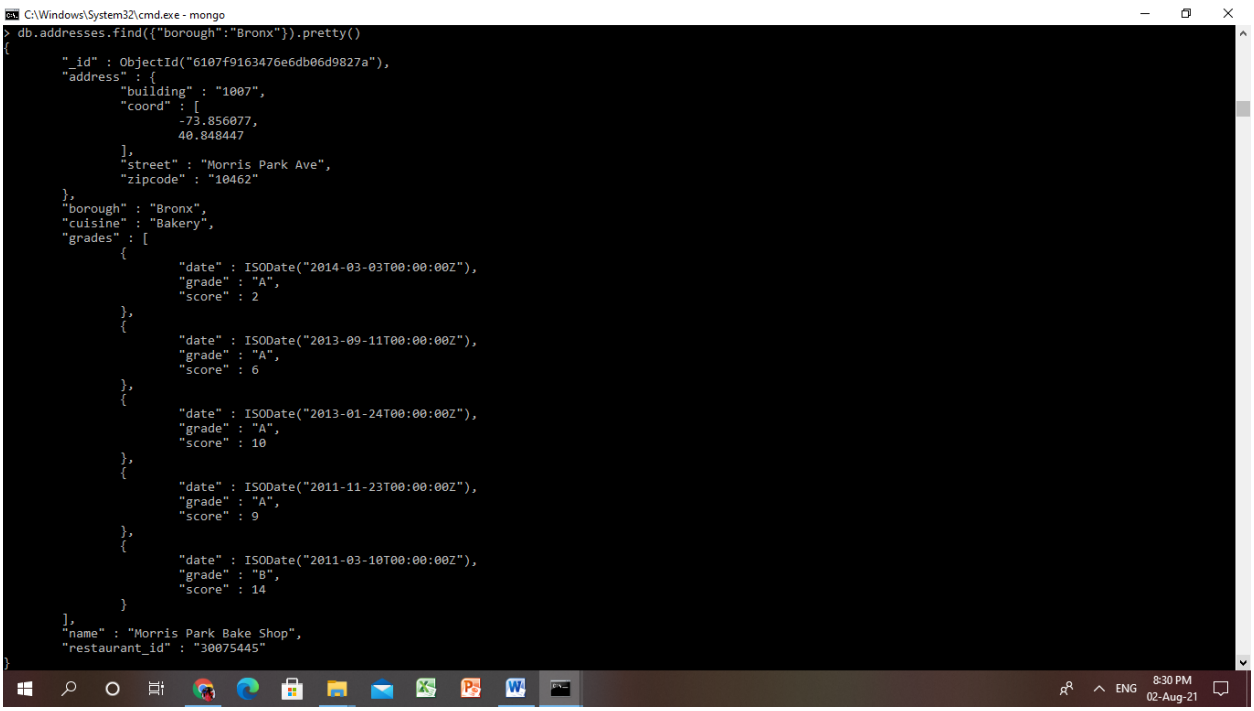
5)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"borough":"Bronx"}).pretty()
{
"_id" : ObjectId("6107f9163476e6db06d9827a"),
"address" : {
"building" : "1007",
"coord" : [
-73.856077,
40.848447
],
"street" : "Morris Park Ave",
"zipcode" : "10462"
},
"borough" : "Bronx",
"cuisine" : "Bakery",
"grades" : [
{
"date" : ISODate("2014-03-03T00:00:00Z"),
"grade" : "A",
"score" : 2
},
{
"date" : ISODate("2013-09-11T00:00:00Z"),
"grade" : "A",
"score" : 6
},
{
"date" : ISODate("2013-01-24T00:00:00Z"),
"grade" : "A",
"score" : 10
},
{
"date" : ISODate("2011-11-23T00:00:00Z"),
"grade" : "A",
"score" : 9
},
{
"date" : ISODate("2011-03-10T00:00:00Z"),
"grade" : "B",
"score" : 14
}
],
"name" : "Morris Park Bake Shop",
"restaurant_id" : "30075445"
}
ENG
11:49 PM
02-Aug-21

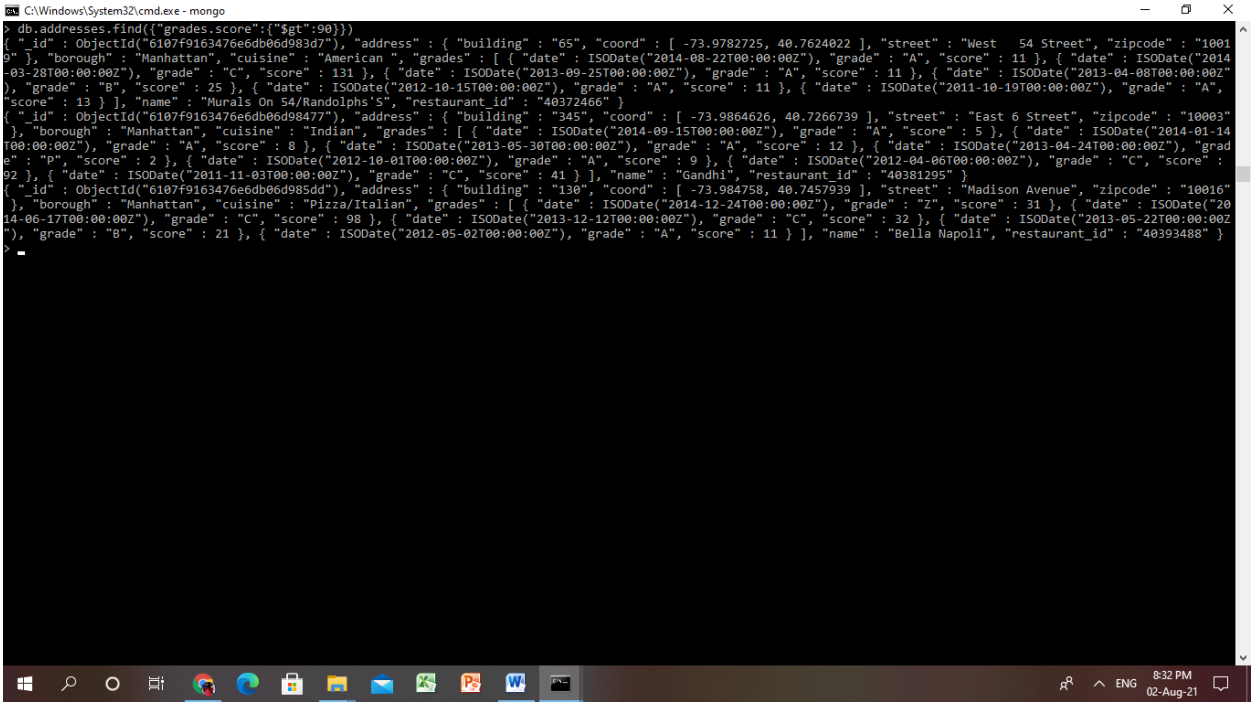
6)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"borough":"Bronx"}).limit(5)
{ "_id" : ObjectId("6107f9163476e6db06d9827a"), "address" : { "building" : "1007", "coord" : [-73.856077, 40.848447], "street" : "Morris Park Ave", "zipcode" : "10462
" }, "borough" : "Bronx", "cuisine" : "Bakery", "grades" : [{ "date" : ISODate("2014-03-03T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2013-09-11T00
:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2013-01-24T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade"
: "A", "score" : 9 }, { "date" : ISODate("2011-03-10T00:00:00Z"), "grade" : "B", "score" : 14 }], "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("6107f9163476e6db06d98282"), "address" : { "building" : "2300", "coord" : [-73.8786113, 40.8502883], "street" : "Southern Boulevard", "zipcode" : "
10460" }, "borough" : "Bronx", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-05-28T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013
-06-19T00:00:00Z"), "grade" : "A", "score" : 4 }, { "date" : ISODate("2012-06-15T00:00:00Z"), "grade" : "A", "score" : 3 }], "name" : "Wild Asia", "restaurant_id" : "4
0357217" }
{ "_id" : ObjectId("6107f9163476e6db06d98297"), "address" : { "building" : "1006", "coord" : [-73.84856870000002, 40.8903781], "street" : "East 233 Street", "zipcode"
: "10466" }, "borough" : "Bronx", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "grades" : [{ "date" : ISODate("2014-04-24T00:00:00Z"), "grade" : "A", "score" : 10 }
, { "date" : ISODate("2013-09-05T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-02-21T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODat
e("2012-07-03T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-07-11T00:00:00Z"), "grade" : "A", "score" : 5 }], "name" : "Carvel Ice Cream", "rest
aurant_id" : "40363093" }
{ "_id" : ObjectId("6107f9163476e6db06d982a1"), "address" : { "building" : "1236", "coord" : [-73.8893654, 40.81376179999999], "street" : "238 Spofford Ave", "zipcode
" : "10474" }, "borough" : "Bronx", "cuisine" : "Chinese", "grades" : [{ "date" : ISODate("2013-12-30T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("20
13-01-08T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-06-12T00:00:00Z"), "grade" : "B", "score" : 15 }], "name" : "Happy Garden", "restaurant_i
d" : "40363289" }
{ "_id" : ObjectId("6107f9163476e6db06d982b0"), "address" : { "building" : "277", "coord" : [-73.8941893, 40.8634684], "street" : "East Kingsbridge Road", "zipcode" :
"10458" }, "borough" : "Bronx", "cuisine" : "Chinese", "grades" : [{ "date" : ISODate("2014-03-03T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013
-09-26T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-03-19T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-08-29T00:00:00Z")
, "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-08-17T00:00:00Z"), "grade" : "A", "score" : 13 }], "name" : "Happy Garden", "restaurant_id" : "40364296" }
> db.addresses.find({"borough":"Bronx"}).pretty()
{
"_id" : ObjectId("6107f9163476e6db06d9827a"),
"address" : {
"building" : "1007",
"coord" : [
-73.856077,
40.848447
],
"street" : "Morris Park Ave",
"zipcode" : "10462"
},
"borough" : "Bronx",
"cuisine" : "Bakery",
"grades" : [
{
"date" : ISODate("2014-03-03T00:00:00Z"),
"grade" : "A",
"score" : 2
},
{
"date" : ISODate("2013-09-11T00:00:00Z"),
ENG
8:25 PM
02-Aug-21

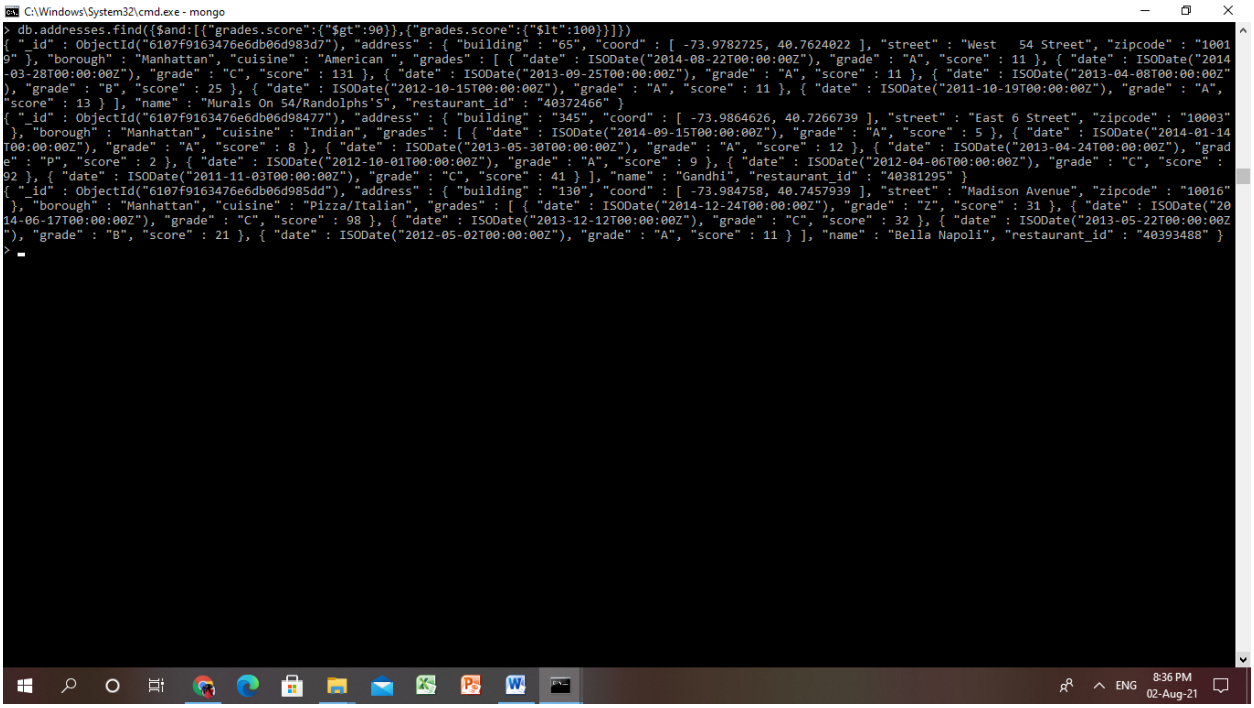
7)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"borough":"Bronx"}).pretty()
{
"_id" : ObjectId("6107f9163476e6db06d9827a"),
"address" : {
"building" : "1007",
"coord" : [
-73.856077,
40.848447
],
"street" : "Morris Park Ave",
"zipcode" : "10462"
},
"borough" : "Bronx",
"cuisine" : "Bakery",
"grades" : [
{
"date" : ISODate("2014-03-03T00:00:00Z"),
"grade" : "A",
"score" : 2
},
{
"date" : ISODate("2013-09-11T00:00:00Z"),
"grade" : "A",
"score" : 6
},
{
"date" : ISODate("2013-01-24T00:00:00Z"),
"grade" : "A",
"score" : 10
},
{
"date" : ISODate("2011-11-23T00:00:00Z"),
"grade" : "A",
"score" : 9
},
{
"date" : ISODate("2011-03-10T00:00:00Z"),
"grade" : "B",
"score" : 14
}
],
"name" : "Morris Park Bake Shop",
"restaurant_id" : "30075445"
}
ENG
8:30 PM
02-Aug-21

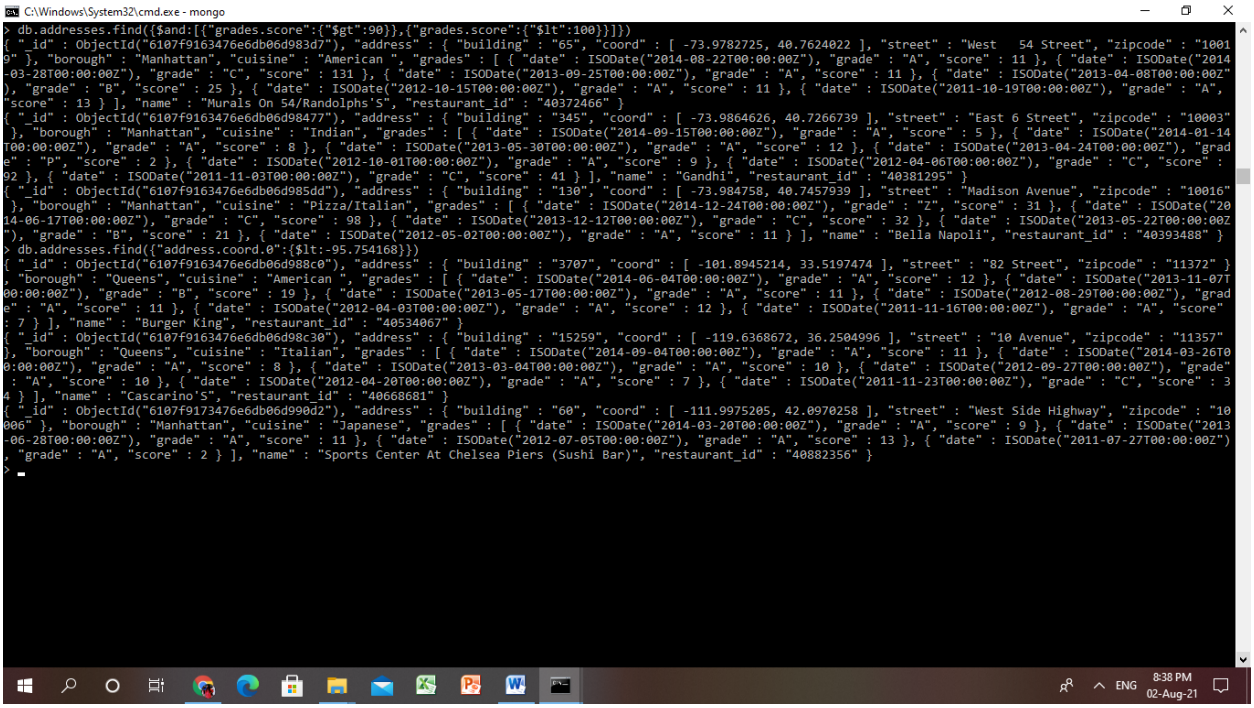
8)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"grades.score":{"$gt":90}})
{ "_id" : ObjectId("6107f9163476e6db06d983d7"), "address" : { "building" : "65", "coord" : [-73.9782725, 40.7624022], "street" : "West 54 Street", "zipcode" : "1001
9" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-08-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014
-03-28T00:00:00Z"), "grade" : "C", "score" : 131 }, { "date" : ISODate("2013-09-25T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-04-08T00:00:00Z"
}, "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-15T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A",
"score" : 13 }], "name" : "Murals On 54/Randolphs'S", "restaurant_id" : "40372466" }
{ "_id" : ObjectId("6107f9163476e6db06d98477"), "address" : { "building" : "345", "coord" : [-73.9864626, 40.7266739], "street" : "East 6 Street", "zipcode" : "10003"
}, "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [{ "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14
T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grad
e" : "P", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" :
92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 }], "name" : "Gandhi", "restaurant_id" : "40381295" }
{ "_id" : ObjectId("6107f9163476e6db06d985dd"), "address" : { "building" : "130", "coord" : [-73.984758, 40.7457939], "street" : "Madison Avenue", "zipcode" : "10016"
}, "borough" : "Manhattan", "cuisine" : "Pizza/Italian", "grades" : [{ "date" : ISODate("2014-12-24T00:00:00Z"), "grade" : "Z", "score" : 31 }, { "date" : ISODate("20
14-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z
"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }], "name" : "Bella Napoli", "restaurant_id" : "40393488" }
> _
ENG
8:32 PM
02-Aug-21

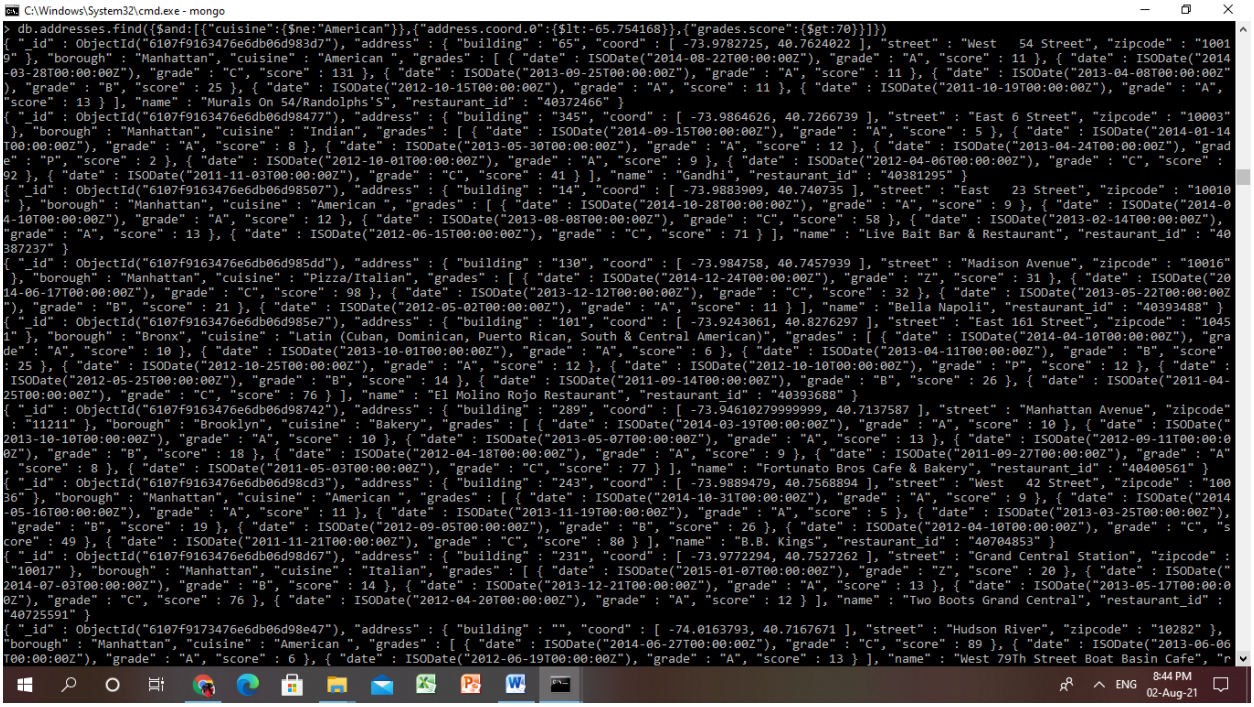
9)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({$and:[{"grades.score":{"$gt":90}},{"grades.score":{"$lt":100}}]})
{ "_id" : ObjectId("6107f9163476e6db06d983d7"), "address" : { "building" : "65", "coord" : [-73.9782725, 40.7624022], "street" : "West 54 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-08-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-03-28T00:00:00Z"), "grade" : "C", "score" : 131 }, { "date" : ISODate("2013-09-25T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-04-08T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-15T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 }], "name" : "Murals On 54/Randolphs'S", "restaurant_id" : "40372466" }
{ "_id" : ObjectId("6107f9163476e6db06d98477"), "address" : { "building" : "345", "coord" : [-73.9864626, 40.7266739], "street" : "East 6 Street", "zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [{ "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "P", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 }], "name" : "Gandhi", "restaurant_id" : "40381295" }
{ "_id" : ObjectId("6107f9163476e6db06d985dd"), "address" : { "building" : "130", "coord" : [-73.984758, 40.7457939], "street" : "Madison Avenue", "zipcode" : "10016" }, "borough" : "Manhattan", "cuisine" : "Pizza/Italian", "grades" : [{ "date" : ISODate("2014-12-24T00:00:00Z"), "grade" : "Z", "score" : 31 }, { "date" : ISODate("2014-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }], "name" : "Bella Napoli", "restaurant_id" : "40393488" }
> _
8:36 PM
02-Aug-21

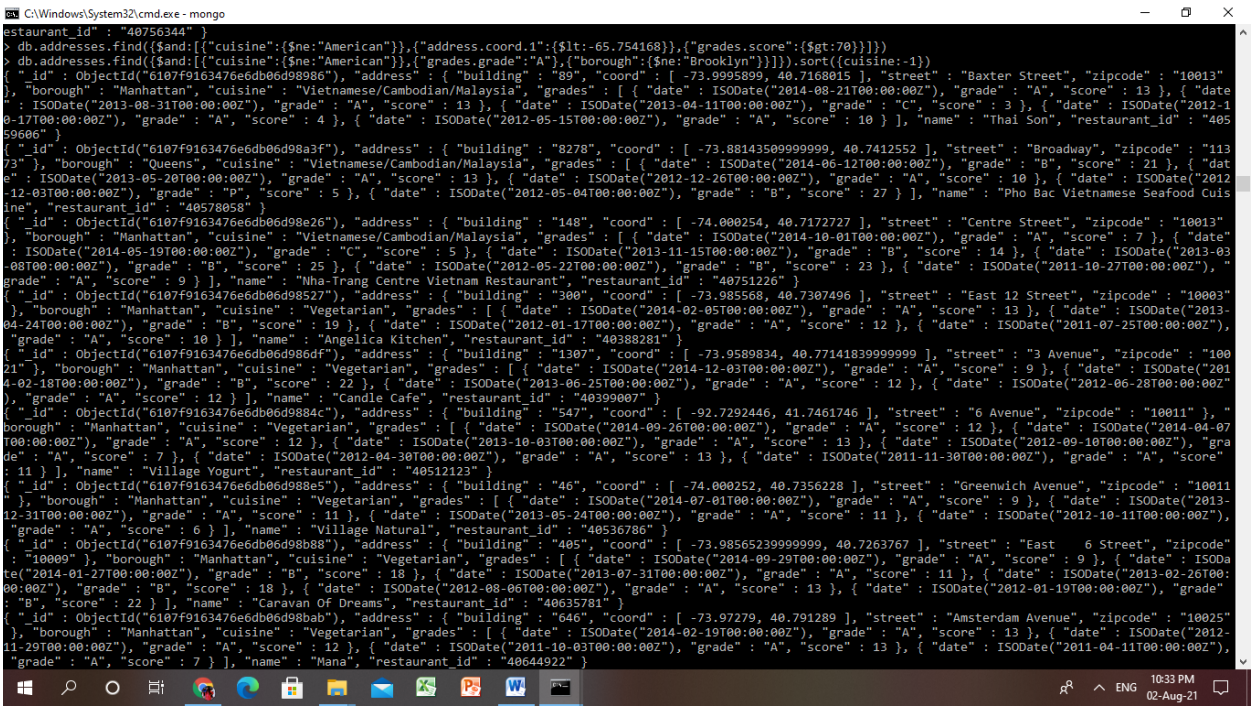
10)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({$and:[{"grades.score":{"$gt":90}},{"grades.score":{"$lt":100}}]})
{ "_id" : ObjectId("6107f9163476e6db06d983d7"), "address" : { "building" : "65", "coord" : [-73.9782725, 40.7624022], "street" : "West 54 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-08-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-03-28T00:00:00Z"), "grade" : "C", "score" : 131 }, { "date" : ISODate("2013-09-25T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-04-08T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-15T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 }], "name" : "Murals On 54/Randolphs'S", "restaurant_id" : "40372466" }
{ "_id" : ObjectId("6107f9163476e6db06d98477"), "address" : { "building" : "345", "coord" : [-73.9864626, 40.7266739], "street" : "East 6 Street", "zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [{ "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "P", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 }], "name" : "Gandhi", "restaurant_id" : "40381295" }
{ "_id" : ObjectId("6107f9163476e6db06d985dd"), "address" : { "building" : "130", "coord" : [-73.984758, 40.7457939], "street" : "Madison Avenue", "zipcode" : "10016" }, "borough" : "Manhattan", "cuisine" : "Pizza/Italian", "grades" : [{ "date" : ISODate("2014-12-24T00:00:00Z"), "grade" : "Z", "score" : 31 }, { "date" : ISODate("2014-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }], "name" : "Bella Napoli", "restaurant_id" : "40393488" }
> db.addresses.find({"address.coord.0":{$lt:-95.754168}})
{ "_id" : ObjectId("6107f9163476e6db06d988c0"), "address" : { "building" : "3707", "coord" : [-101.8945214, 33.5197474], "street" : "82 Street", "zipcode" : "11372" }, "borough" : "Queens", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-06-04T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-11-07T00:00:00Z"), "grade" : "B", "score" : 19 }, { "date" : ISODate("2013-05-17T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-08-29T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-04-03T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-11-16T00:00:00Z"), "grade" : "A", "score" : 7 }], "name" : "Burger King", "restaurant_id" : "40534067" }
{ "_id" : ObjectId("6107f9163476e6db06d98c30"), "address" : { "building" : "15259", "coord" : [-119.6368672, 36.2504996], "street" : "10 Avenue", "zipcode" : "11357" }, "borough" : "Queens", "cuisine" : "Italian", "grades" : [{ "date" : ISODate("2014-09-04T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-03-26T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-03-04T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-09-27T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-04-20T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "C", "score" : 34 }], "name" : "Cascarino'S", "restaurant_id" : "40668681" }
{ "_id" : ObjectId("6107f9173476e6db06d990d2"), "address" : { "building" : "60", "coord" : [-111.9975205, 42.0970258], "street" : "West Side Highway", "zipcode" : "10006" }, "borough" : "Manhattan", "cuisine" : "Japanese", "grades" : [{ "date" : ISODate("2014-03-20T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-06-28T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-07-05T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-07-27T00:00:00Z"), "grade" : "A", "score" : 2 }], "name" : "Sports Center At Chelsea Piers (Sushi Bar)", "restaurant_id" : "40882356" }
> _
8:38 PM
02-Aug-21

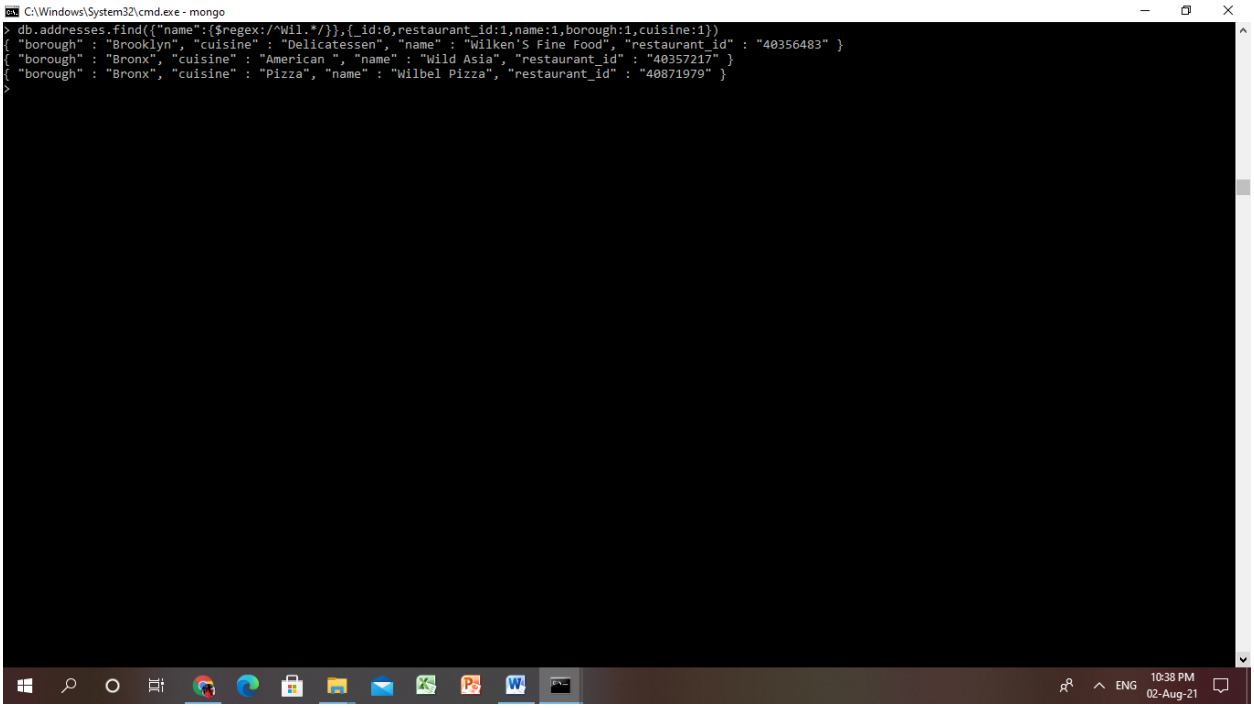
11)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({$and:[{"cuisine":{$ne:"American"}},{"address.coord.0":{$lt:-65.754168}},{"grades.score":{$gt:70}}]})
{ "_id" : ObjectId("6107f9163476e6db06d983d7"), "address" : { "building" : "65", "coord" : [-73.9782725, 40.7624022], "street" : "West 54 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-08-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-03-28T00:00:00Z"), "grade" : "C", "score" : 131 }, { "date" : ISODate("2013-09-25T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-04-08T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-15T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 }], "name" : "Murals On 54/Randolphs'S", "restaurant_id" : "40372466" }
{ "_id" : ObjectId("6107f9163476e6db06d98477"), "address" : { "building" : "345", "coord" : [-73.9864626, 40.7266739], "street" : "East 6 Street", "zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [{ "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "P", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 }], "name" : "Gandhi", "restaurant_id" : "40381295" }
{ "_id" : ObjectId("6107f9163476e6db06d98507"), "address" : { "building" : "14", "coord" : [-73.9883909, 40.740735], "street" : "East 23 Street", "zipcode" : "10010" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-10-28T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-04-10T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-08-08T00:00:00Z"), "grade" : "C", "score" : 58 }, { "date" : ISODate("2013-02-14T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-06-15T00:00:00Z"), "grade" : "C", "score" : 71 }], "name" : "Live Bait Bar & Restaurant", "restaurant_id" : "40387237" }
{ "_id" : ObjectId("6107f9163476e6db06d985dd"), "address" : { "building" : "130", "coord" : [-73.984758, 40.7457939], "street" : "Madison Avenue", "zipcode" : "10016" }, "borough" : "Manhattan", "cuisine" : "Pizza/Italian", "grades" : [{ "date" : ISODate("2014-12-24T00:00:00Z"), "grade" : "Z", "score" : 31 }, { "date" : ISODate("2014-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }], "name" : "Bella Napoli", "restaurant_id" : "40393488" }
{ "_id" : ObjectId("6107f9163476e6db06d985e7"), "address" : { "building" : "101", "coord" : [-73.9243061, 40.8276297], "street" : "East 161 Street", "zipcode" : "10451" }, "borough" : "Bronx", "cuisine" : "Latin (Cuban, Dominican, Puerto Rican, South & Central American)", "grades" : [{ "date" : ISODate("2014-04-10T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-10-01T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2013-04-11T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-25T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-10-10T00:00:00Z"), "grade" : "P", "score" : 12 }, { "date" : ISODate("2012-05-25T00:00:00Z"), "grade" : "B", "score" : 14 }, { "date" : ISODate("2011-09-14T00:00:00Z"), "grade" : "B", "score" : 26 }, { "date" : ISODate("2011-04-25T00:00:00Z"), "grade" : "C", "score" : 76 }], "name" : "El Molino Rojo Restaurant", "restaurant_id" : "40393688" }
{ "_id" : ObjectId("6107f9163476e6db06d98742"), "address" : { "building" : "289", "coord" : [-73.94610279999999, 40.7137587], "street" : "Manhattan Avenue", "zipcode" : "11211" }, "borough" : "Brooklyn", "cuisine" : "Bakery", "grades" : [{ "date" : ISODate("2014-03-19T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-10-10T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-05-07T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-09-11T00:00:00Z"), "grade" : "B", "score" : 18 }, { "date" : ISODate("2012-04-18T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2011-09-27T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2011-05-03T00:00:00Z"), "grade" : "C", "score" : 77 }], "name" : "Fortunato Bros Cafe & Bakery", "restaurant_id" : "40400561" }
{ "_id" : ObjectId("6107f9163476e6db06d98cd3"), "address" : { "building" : "243", "coord" : [-73.9889479, 40.7568894], "street" : "West 42 Street", "zipcode" : "10036" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-10-31T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-05-16T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-11-19T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-03-25T00:00:00Z"), "grade" : "B", "score" : 19 }, { "date" : ISODate("2012-09-05T00:00:00Z"), "grade" : "B", "score" : 26 }, { "date" : ISODate("2012-04-10T00:00:00Z"), "grade" : "C", "score" : 49 }, { "date" : ISODate("2011-11-21T00:00:00Z"), "grade" : "C", "score" : 80 }], "name" : "B.B. Kings", "restaurant_id" : "40704853" }
{ "_id" : ObjectId("6107f9163476e6db06d98d67"), "address" : { "building" : "231", "coord" : [-73.9772294, 40.7527262], "street" : "Grand Central Station", "zipcode" : "10017" }, "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [{ "date" : ISODate("2015-01-07T00:00:00Z"), "grade" : "Z", "score" : 20 }, { "date" : ISODate("2014-07-03T00:00:00Z"), "grade" : "B", "score" : 14 }, { "date" : ISODate("2013-12-21T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2013-05-17T00:00:00Z"), "grade" : "C", "score" : 76 }, { "date" : ISODate("2012-04-20T00:00:00Z"), "grade" : "A", "score" : 12 }], "name" : "Two Boots Grand Central", "restaurant_id" : "40725591" }
{ "_id" : ObjectId("6107f9173476e6db06d98e47"), "address" : { "building" : "", "coord" : [-74.0163793, 40.7167671], "street" : "Hudson River", "zipcode" : "10282" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [{ "date" : ISODate("2014-06-27T00:00:00Z"), "grade" : "C", "score" : 89 }, { "date" : ISODate("2013-06-06T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2012-06-19T00:00:00Z"), "grade" : "A", "score" : 13 }], "name" : "West 79Th Street Boat Basin Cafe", "r
ENG
8:44 PM
02-Aug-21

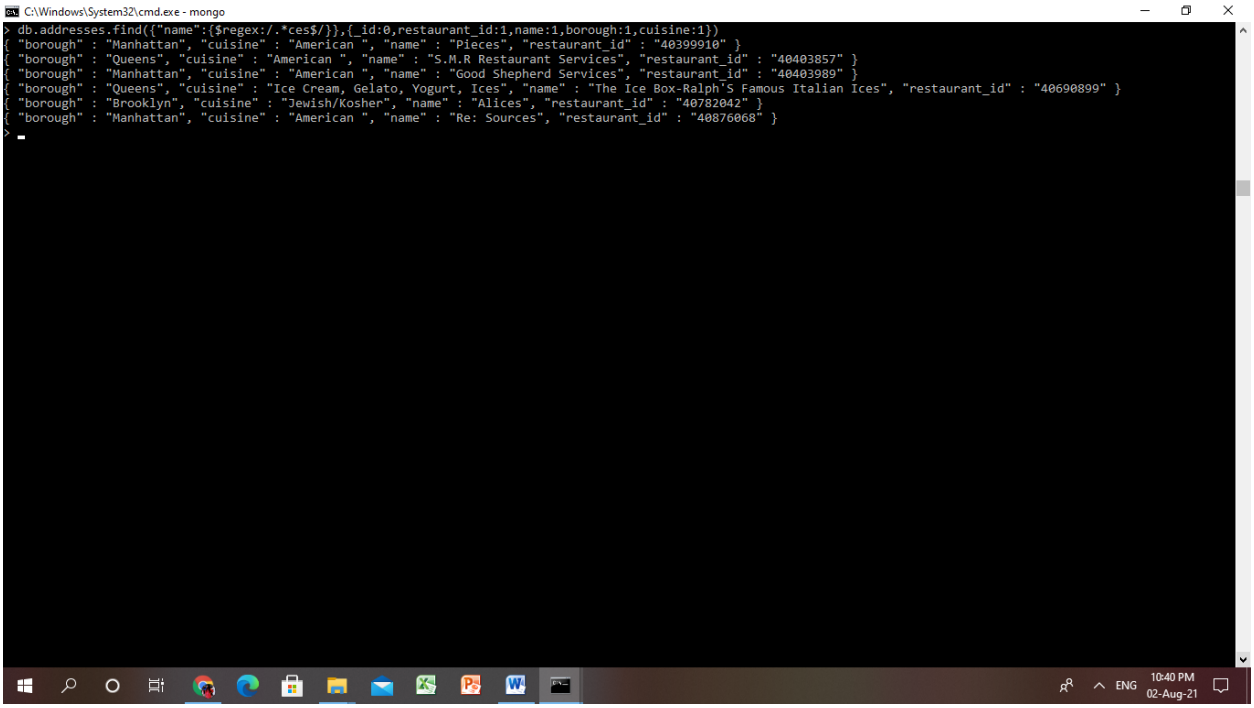
13)


C:\Windows\System32\cmd.exe - mongo
estaurant_id" : "40756344" }
> db.addresses.find({$and:[{"cuisine":{$ne:"American"}},{"address.coord.1":{$lt:-65.754168}},{"grades.score":{$gt:70}}]})
> db.addresses.find({$and:[{"cuisine":{$ne:"American"}},{"grades.grade":"A"},{"borough":{$ne:"Brooklyn"}}]}).sort({cuisine:-1})
{ "_id" : ObjectId("6107f9163476e6db06d98986"), "address" : { "building" : "89", "coord" : [-73.9995899, 40.7168015], "street" : "Baxter Street", "zipcode" : "10013" }, "borough" : "Manhattan", "cuisine" : "Vietnamese/Cambodian/Malaysia", "grades" : [{ "date" : ISODate("2014-08-21T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2013-08-31T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2013-04-11T00:00:00Z"), "grade" : "C", "score" : 3 }, { "date" : ISODate("2012-10-17T00:00:00Z"), "grade" : "A", "score" : 4 }, { "date" : ISODate("2012-05-15T00:00:00Z"), "grade" : "A", "score" : 10 }], "name" : "Thai Son", "restaurant_id" : "40559606" }
{ "_id" : ObjectId("6107f9163476e6db06d98a3f"), "address" : { "building" : "8278", "coord" : [-73.88143509999999, 40.7412552], "street" : "Broadway", "zipcode" : "11373" }, "borough" : "Queens", "cuisine" : "Vietnamese/Cambodian/Malaysia", "grades" : [{ "date" : ISODate("2014-06-12T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2013-05-20T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-12-26T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-12-03T00:00:00Z"), "grade" : "P", "score" : 5 }, { "date" : ISODate("2012-05-04T00:00:00Z"), "grade" : "B", "score" : 27 }], "name" : "Pho Bac Vietnamese Seafood Cuisine", "restaurant_id" : "40578058" }
{ "_id" : ObjectId("6107f9163476e6db06d98e26"), "address" : { "building" : "148", "coord" : [-74.000254, 40.7172727], "street" : "Centre Street", "zipcode" : "10013" }, "borough" : "Manhattan", "cuisine" : "Vietnamese/Cambodian/Malaysia", "grades" : [{ "date" : ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2014-05-19T00:00:00Z"), "grade" : "C", "score" : 5 }, { "date" : ISODate("2013-11-15T00:00:00Z"), "grade" : "B", "score" : 14 }, { "date" : ISODate("2013-03-08T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-05-22T00:00:00Z"), "grade" : "B", "score" : 23 }, { "date" : ISODate("2011-10-27T00:00:00Z"), "grade" : "A", "score" : 9 }], "name" : "Nha-Trang Centre Vietnam Restaurant", "restaurant_id" : "40751226" }
{ "_id" : ObjectId("6107f9163476e6db06d98527"), "address" : { "building" : "300", "coord" : [-73.985568, 40.7307496], "street" : "East 12 Street", "zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Vegetarian", "grades" : [{ "date" : ISODate("2014-02-05T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "B", "score" : 19 }, { "date" : ISODate("2012-01-17T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-07-25T00:00:00Z"), "grade" : "A", "score" : 10 }], "name" : "Angelica Kitchen", "restaurant_id" : "40388281" }
{ "_id" : ObjectId("6107f9163476e6db06d986df"), "address" : { "building" : "1307", "coord" : [-73.9589834, 40.77141839999999], "street" : "3 Avenue", "zipcode" : "10021" }, "borough" : "Manhattan", "cuisine" : "Vegetarian", "grades" : [{ "date" : ISODate("2014-12-03T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-02-18T00:00:00Z"), "grade" : "B", "score" : 22 }, { "date" : ISODate("2013-06-25T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-06-28T00:00:00Z"), "grade" : "A", "score" : 12 }], "name" : "Candle Cafe", "restaurant_id" : "40399007" }
{ "_id" : ObjectId("6107f9163476e6db06d9884c"), "address" : { "building" : "547", "coord" : [-92.7292446, 41.7461746], "street" : "6 Avenue", "zipcode" : "10011" }, "borough" : "Manhattan", "cuisine" : "Vegetarian", "grades" : [{ "date" : ISODate("2014-09-26T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2014-04-07T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-10-03T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-09-10T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2012-04-30T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-11-30T00:00:00Z"), "grade" : "A", "score" : 11 }], "name" : "Village Yogurt", "restaurant_id" : "40512123" }
{ "_id" : ObjectId("6107f9163476e6db06d988e5"), "address" : { "building" : "46", "coord" : [-74.000252, 40.7356228], "street" : "Greenwich Avenue", "zipcode" : "10011" }, "borough" : "Manhattan", "cuisine" : "Vegetarian", "grades" : [{ "date" : ISODate("2014-07-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-12-31T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-05-24T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-10-11T00:00:00Z"), "grade" : "A", "score" : 6 }], "name" : "Village Natural", "restaurant_id" : "40536786" }
{ "_id" : ObjectId("6107f9163476e6db06d98b88"), "address" : { "building" : "405", "coord" : [-73.98565239999999, 40.7263767], "street" : "East 6 Street", "zipcode" : "10009" }, "borough" : "Manhattan", "cuisine" : "Vegetarian", "grades" : [{ "date" : ISODate("2014-09-29T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-01-27T00:00:00Z"), "grade" : "B", "score" : 18 }, { "date" : ISODate("2013-07-31T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-02-26T00:00:00Z"), "grade" : "B", "score" : 18 }, { "date" : ISODate("2012-08-06T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-01-19T00:00:00Z"), "grade" : "B", "score" : 22 }], "name" : "Caravan Of Dreams", "restaurant_id" : "40635781" }
{ "_id" : ObjectId("6107f9163476e6db06d98bab"), "address" : { "building" : "646", "coord" : [-73.97279, 40.791289], "street" : "Amsterdam Avenue", "zipcode" : "10025" }, "borough" : "Manhattan", "cuisine" : "Vegetarian", "grades" : [{ "date" : ISODate("2014-02-19T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-11-29T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-10-03T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-04-11T00:00:00Z"), "grade" : "A", "score" : 7 }], "name" : "Mana", "restaurant_id" : "40644922" }
ENG
10:33 PM
02-Aug-21

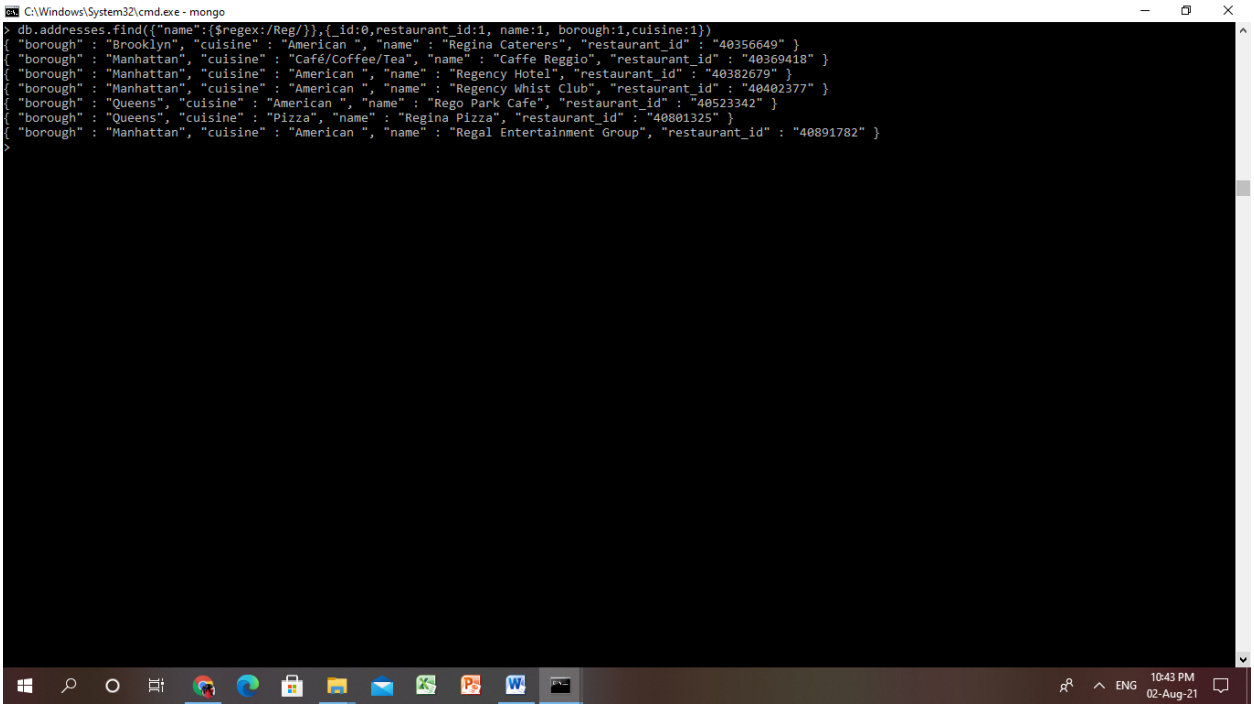
14)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"name":{$regex:/^Wil.*/}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "borough" : "Bronx", "cuisine" : "Pizza", "name" : "Wilbel Pizza", "restaurant_id" : "40871979" }
>
ENG
10:38 PM
02-Aug-21

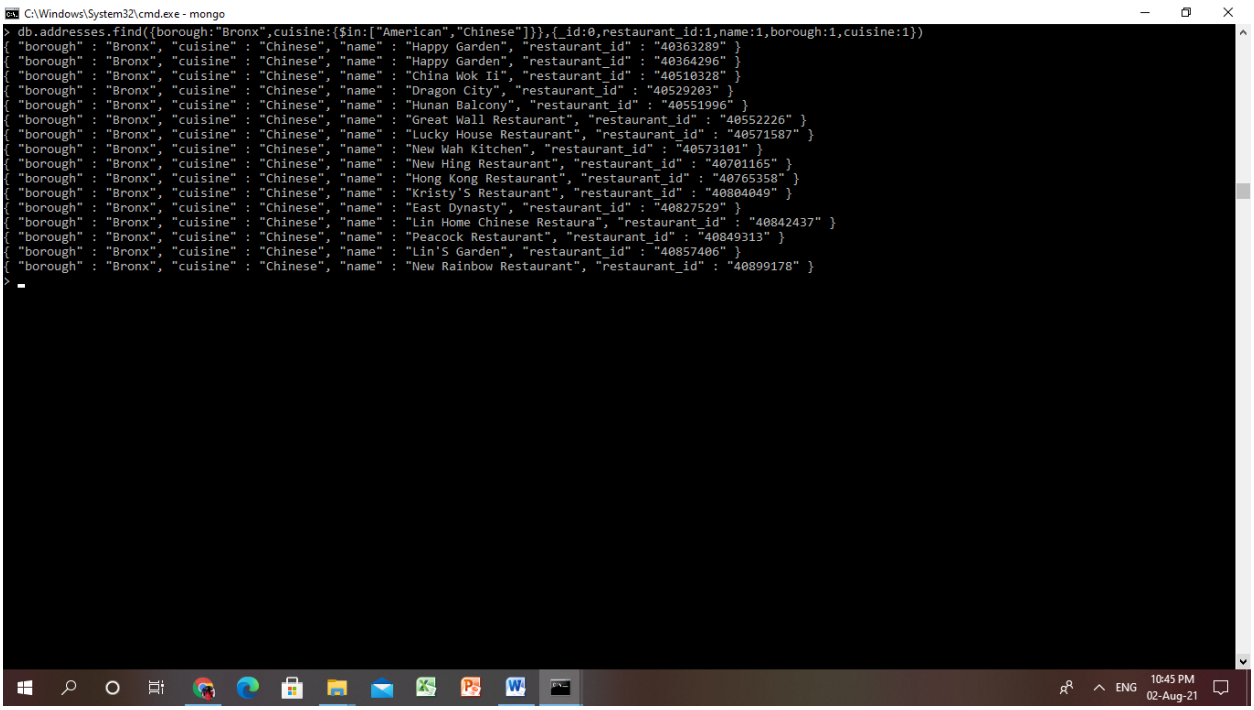
15)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"name":{$regex:/.*ces$/}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Pieces", "restaurant_id" : "40399910" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "S.M.R Restaurant Services", "restaurant_id" : "40403857" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Good Shepherd Services", "restaurant_id" : "40403989" }
{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "The Ice Box-Ralph'S Famous Italian Ices", "restaurant_id" : "40690899" }
{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Alices", "restaurant_id" : "40782042" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Re: Sources", "restaurant_id" : "40876068" }
>
ENG
10:40 PM
02-Aug-21

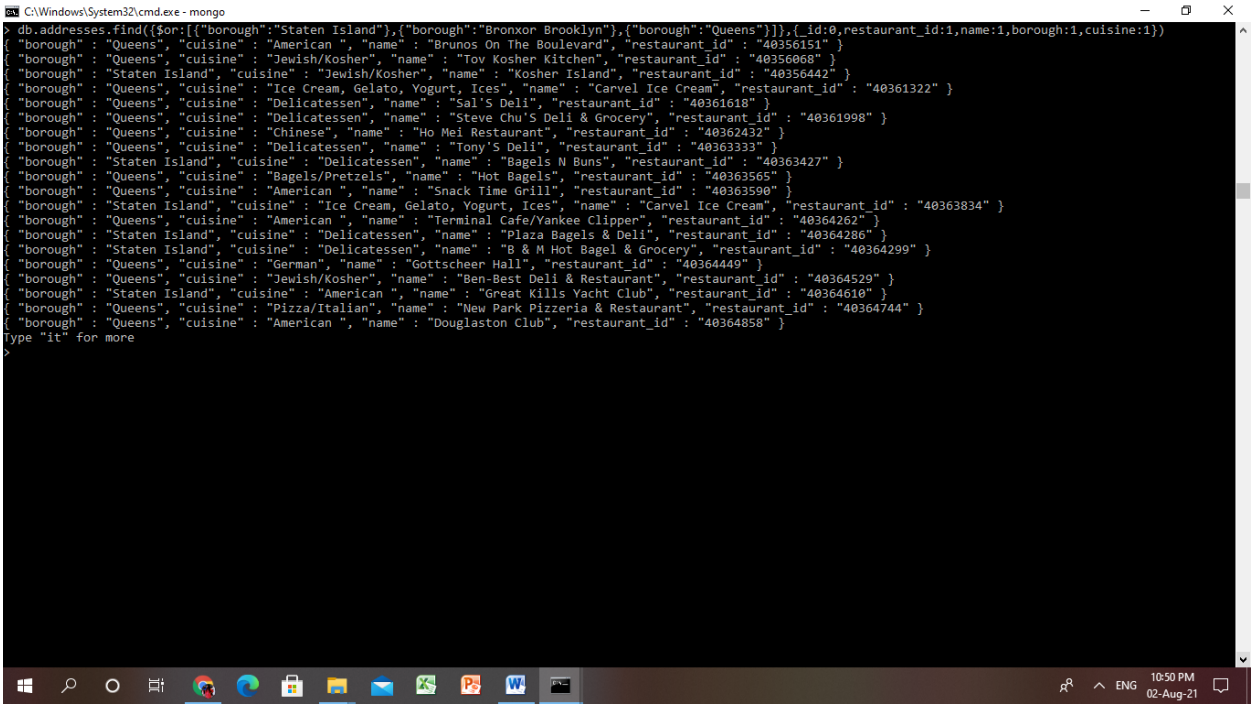
16)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"name":{$regex:/Reg/}},{_id:0,restaurant_id:1, name:1, borough:1,cuisine:1})
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "borough" : "Manhattan", "cuisine" : "Café/Coffee/Tea", "name" : "Caffe Reggio", "restaurant_id" : "40369418" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Regency Hotel", "restaurant_id" : "40382679" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Regency Whist Club", "restaurant_id" : "40402377" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Rego Park Cafe", "restaurant_id" : "40523342" }
{ "borough" : "Queens", "cuisine" : "Pizza", "name" : "Regina Pizza", "restaurant_id" : "40801325" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Regal Entertainment Group", "restaurant_id" : "40891782" }
ENG
10:43 PM
02-Aug-21

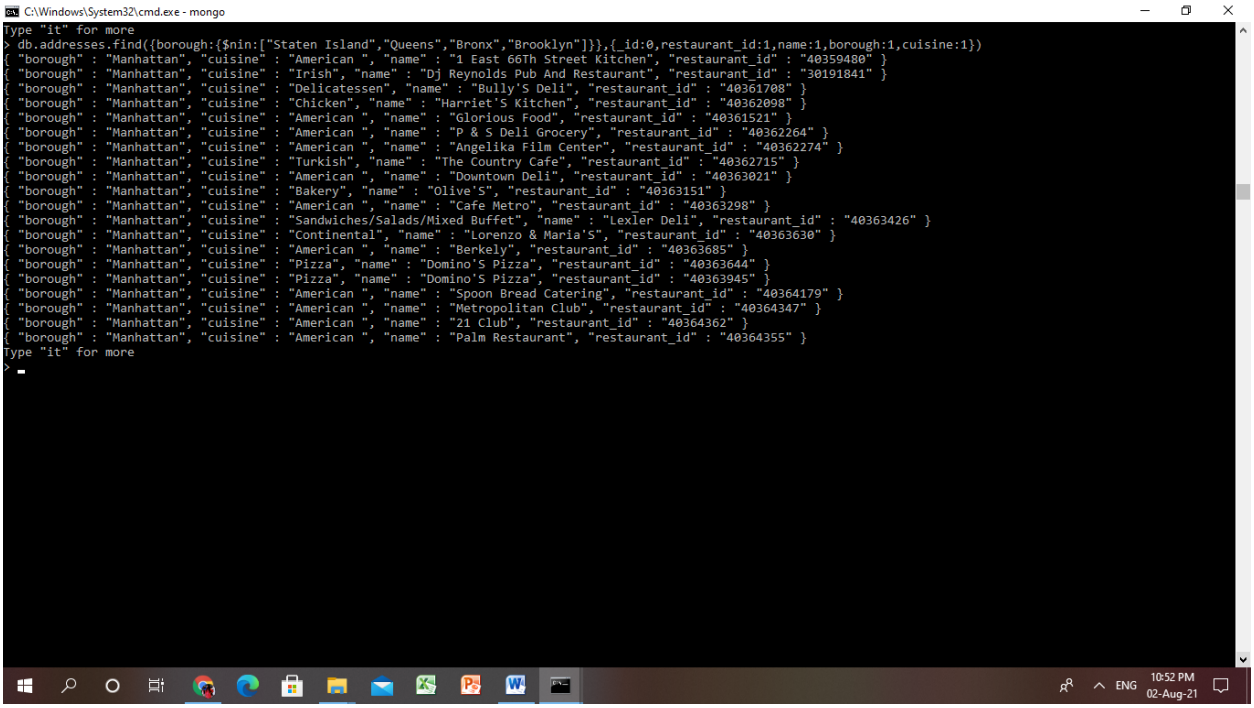
17)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({borough:"Bronx",cuisine:{$in:["American","Chinese"]}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden", "restaurant_id" : "40363289" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden", "restaurant_id" : "40364296" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "China Wok Ii", "restaurant_id" : "40510328" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Dragon City", "restaurant_id" : "40529203" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Hunan Balcony", "restaurant_id" : "40551996" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Great Wall Restaurant", "restaurant_id" : "40552226" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Lucky House Restaurant", "restaurant_id" : "40571587" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "New Wah Kitchen", "restaurant_id" : "40573101" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "New Hing Restaurant", "restaurant_id" : "40701165" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Hong Kong Restaurant", "restaurant_id" : "40765358" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Kristy'S Restaurant", "restaurant_id" : "40804049" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "East Dynasty", "restaurant_id" : "40827529" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Lin Home Chinese Restaura", "restaurant_id" : "40842437" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Peacock Restaurant", "restaurant_id" : "40849313" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Lin'S Garden", "restaurant_id" : "40857406" }
{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "New Rainbow Restaurant", "restaurant_id" : "40899178" }
ENG
10:45 PM
02-Aug-21

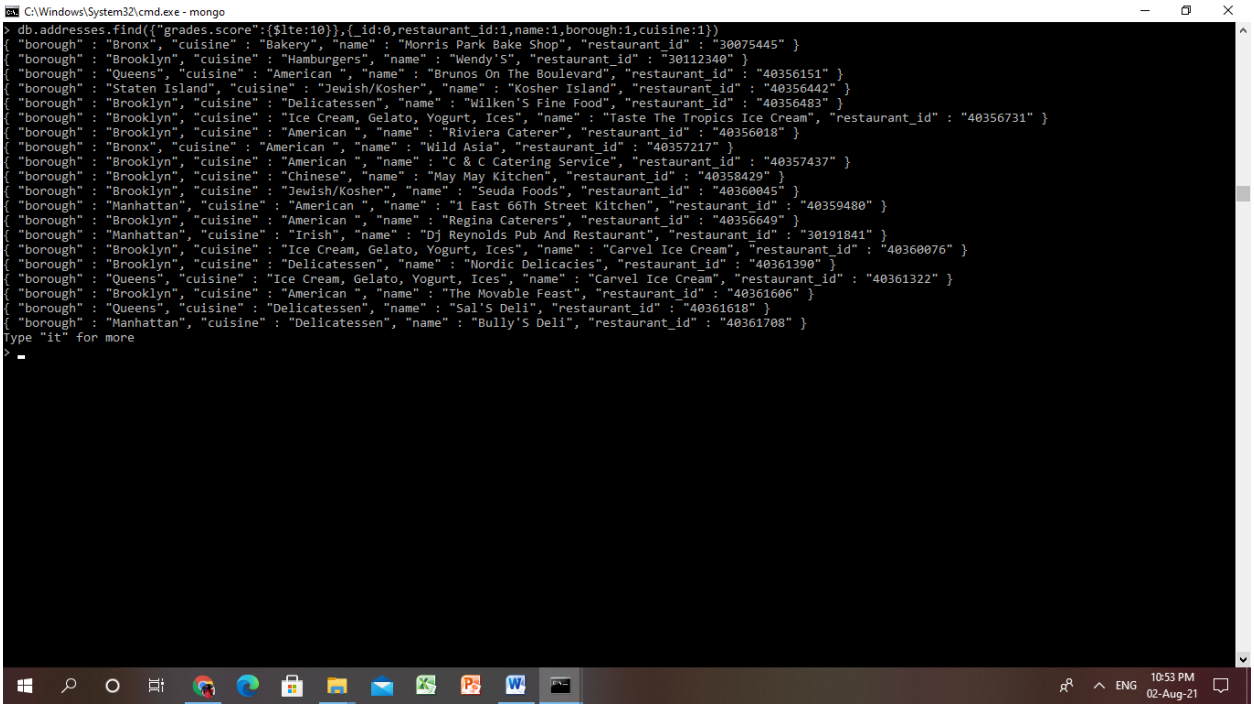
18)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({$or:[{"borough":"Staten Island"},{"borough":"Bronxor Brooklyn"},{"borough":"Queens"}]},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" : "40361618" }
{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant_id" : "40361998" }
{ "borough" : "Queens", "cuisine" : "Chinese", "name" : "Ho Mei Restaurant", "restaurant_id" : "40362432" }
{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Tony'S Deli", "restaurant_id" : "40363333" }
{ "borough" : "Staten Island", "cuisine" : "Delicatessen", "name" : "Bagels N Buns", "restaurant_id" : "40363427" }
{ "borough" : "Queens", "cuisine" : "Bagels/Pretzels", "name" : "Hot Bagels", "restaurant_id" : "40363565" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Snack Time Grill", "restaurant_id" : "40363590" }
{ "borough" : "Staten Island", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40363834" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Terminal Cafe/Yankee Clipper", "restaurant_id" : "40364262" }
{ "borough" : "Staten Island", "cuisine" : "Delicatessen", "name" : "Plaza Bagels & Deli", "restaurant_id" : "40364286" }
{ "borough" : "Staten Island", "cuisine" : "Delicatessen", "name" : "B & M Hot Bagel & Grocery", "restaurant_id" : "40364299" }
{ "borough" : "Queens", "cuisine" : "German", "name" : "Gottscheer Hall", "restaurant_id" : "40364449" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Ben-Best Deli & Restaurant", "restaurant_id" : "40364529" }
{ "borough" : "Staten Island", "cuisine" : "American ", "name" : "Great Kills Yacht Club", "restaurant_id" : "40364610" }
{ "borough" : "Queens", "cuisine" : "Pizza/Italian", "name" : "New Park Pizzeria & Restaurant", "restaurant_id" : "40364744" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Douglaston Club", "restaurant_id" : "40364858" }
Type "it" for more
>
ENG
10:50 PM
02-Aug-21

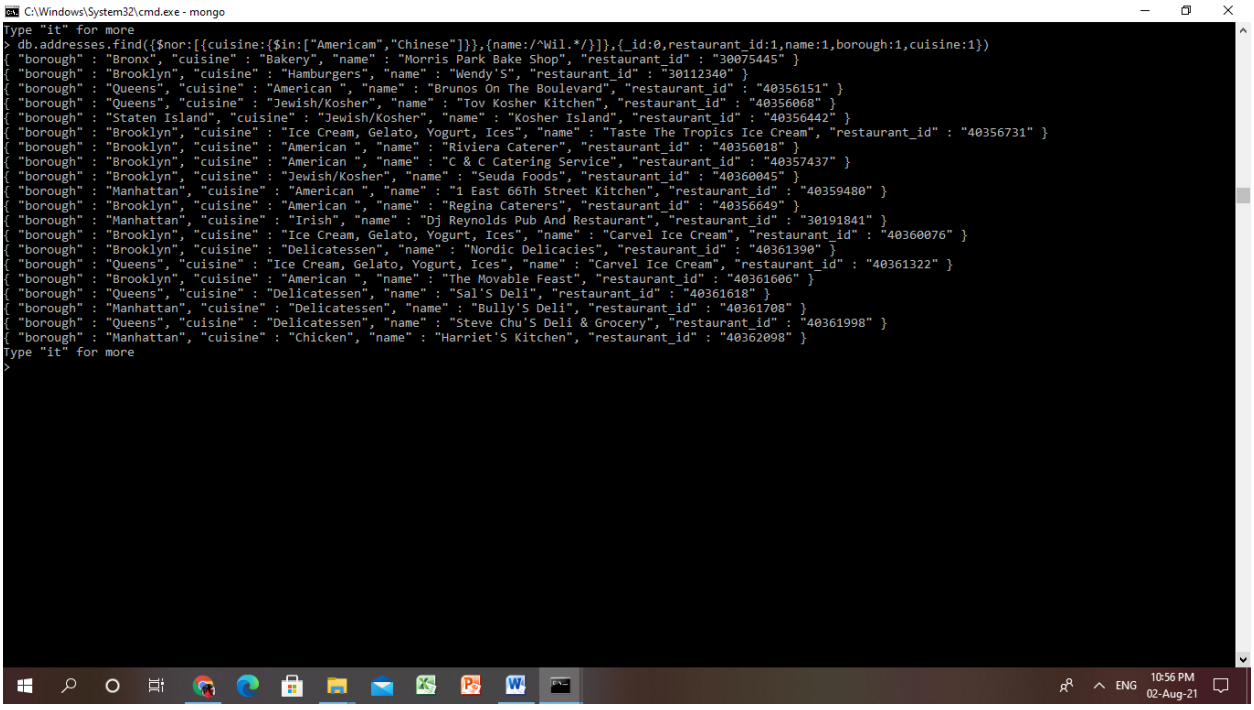
19)


C:\Windows\System32\cmd.exe - mongo
Type "it" for more
> db.addresses.find({borough:{$nin:["Staten Island","Queens","Bronx","Brooklyn"]}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant_id" : "40361708" }
{ "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant_id" : "40362098" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant_id" : "40361521" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery", "restaurant_id" : "40362264" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Angelika Film Center", "restaurant_id" : "40362274" }
{ "borough" : "Manhattan", "cuisine" : "Turkish", "name" : "The Country Cafe", "restaurant_id" : "40362715" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Downtown Deli", "restaurant_id" : "40363021" }
{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Olive'S", "restaurant_id" : "40363151" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Cafe Metro", "restaurant_id" : "40363298" }
{ "borough" : "Manhattan", "cuisine" : "Sandwiches/Salads/Mixed Buffet", "name" : "Lexler Deli", "restaurant_id" : "40363426" }
{ "borough" : "Manhattan", "cuisine" : "Continental", "name" : "Lorenzo & Maria'S", "restaurant_id" : "40363630" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Berkely", "restaurant_id" : "40363685" }
{ "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza", "restaurant_id" : "40363644" }
{ "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza", "restaurant_id" : "40363945" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Spoon Bread Catering", "restaurant_id" : "40364179" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Metropolitan Club", "restaurant_id" : "40364347" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "21 Club", "restaurant_id" : "40364362" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Palm Restaurant", "restaurant_id" : "40364355" }
Type "it" for more
>
ENG
10:52 PM
02-Aug-21

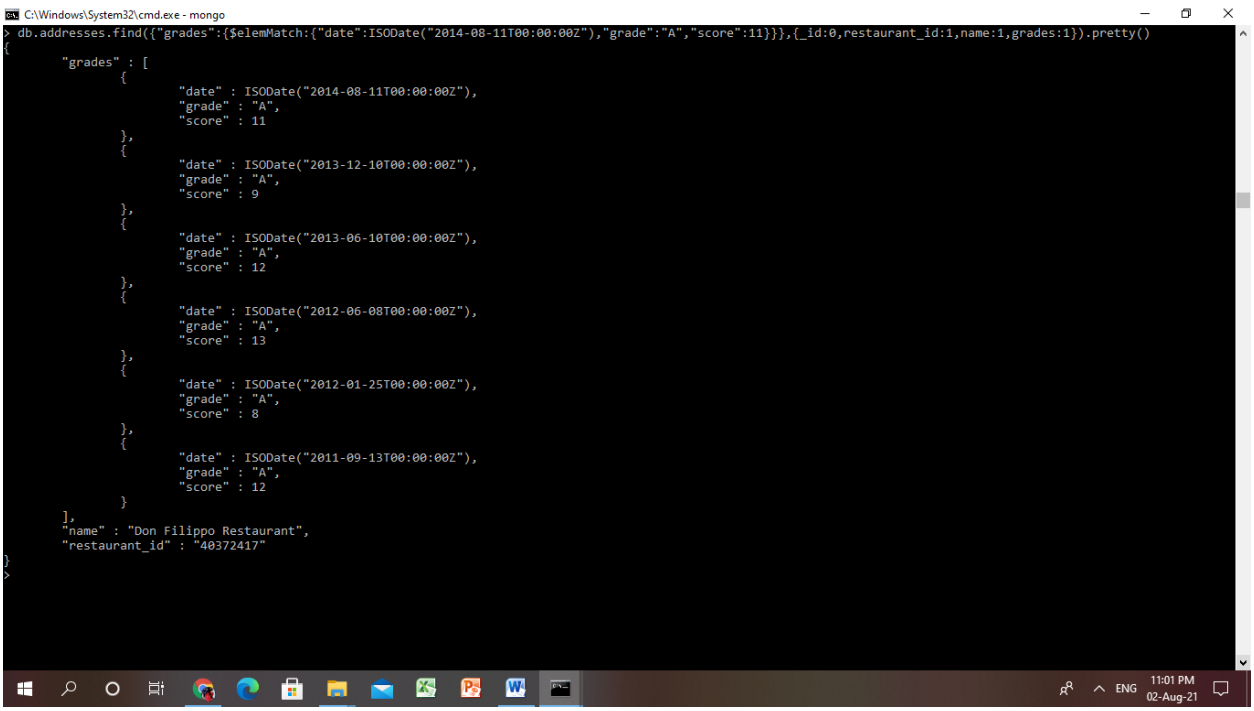
20)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"grades.score":{$lte:10}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "40356731" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant_id" : "40361606" }
{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" : "40361618" }
{ "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant_id" : "40361708" }
Type "it" for more
>
ENG
10:53 PM
02-Aug-21

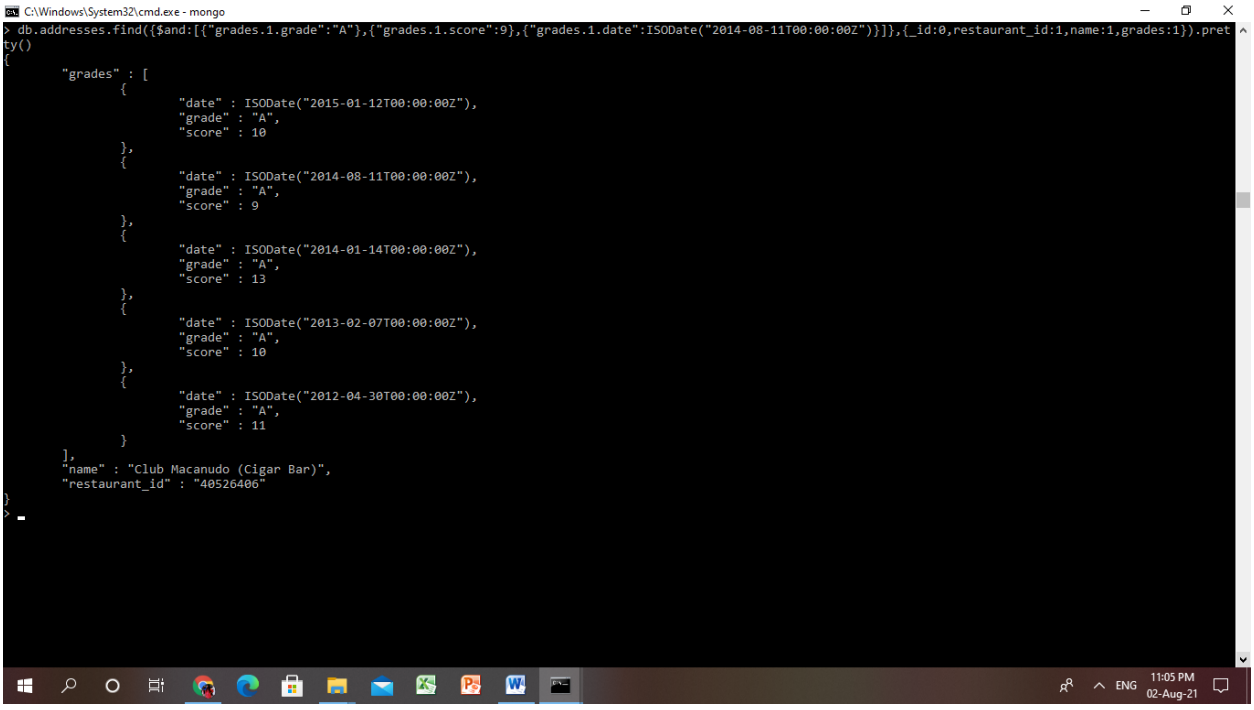
21)


C:\Windows\System32\cmd.exe - mongo
Type "it" for more
> db.addresses.find({$nor:[{cuisine:{$in:["Americam","Chinese"]}},{name:/^Wil.*/}]},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "40356731" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant_id" : "40361606" }
{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" : "40361618" }
{ "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant_id" : "40361708" }
{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant_id" : "40361998" }
{ "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant_id" : "40362098" }
Type "it" for more
>
ENG
10:56 PM
02-Aug-21

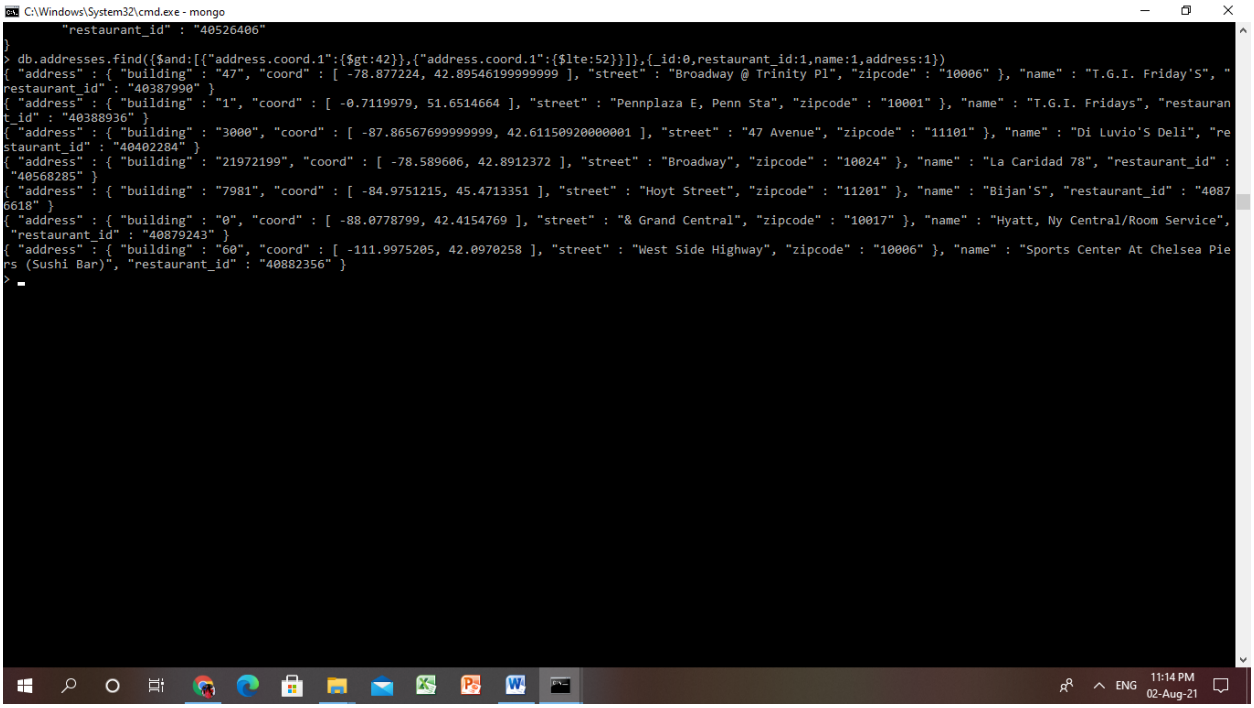
22)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"grades":{$elemMatch:{"date":ISODate("2014-08-11T00:00:00Z"),"grade":"A","score":11}}},{_id:0,restaurant_id:1,name:1,grades:1}).pretty()
{
"grades" : [
{
"date" : ISODate("2014-08-11T00:00:00Z"),
"grade" : "A",
"score" : 11
},
{
"date" : ISODate("2013-12-10T00:00:00Z"),
"grade" : "A",
"score" : 9
},
{
"date" : ISODate("2013-06-10T00:00:00Z"),
"grade" : "A",
"score" : 12
},
{
"date" : ISODate("2012-06-08T00:00:00Z"),
"grade" : "A",
"score" : 13
},
{
"date" : ISODate("2012-01-25T00:00:00Z"),
"grade" : "A",
"score" : 8
},
{
"date" : ISODate("2011-09-13T00:00:00Z"),
"grade" : "A",
"score" : 12
}
],
"name" : "Don Filippo Restaurant",
"restaurant_id" : "40372417"
}
>
ENG
11:01 PM
02-Aug-21

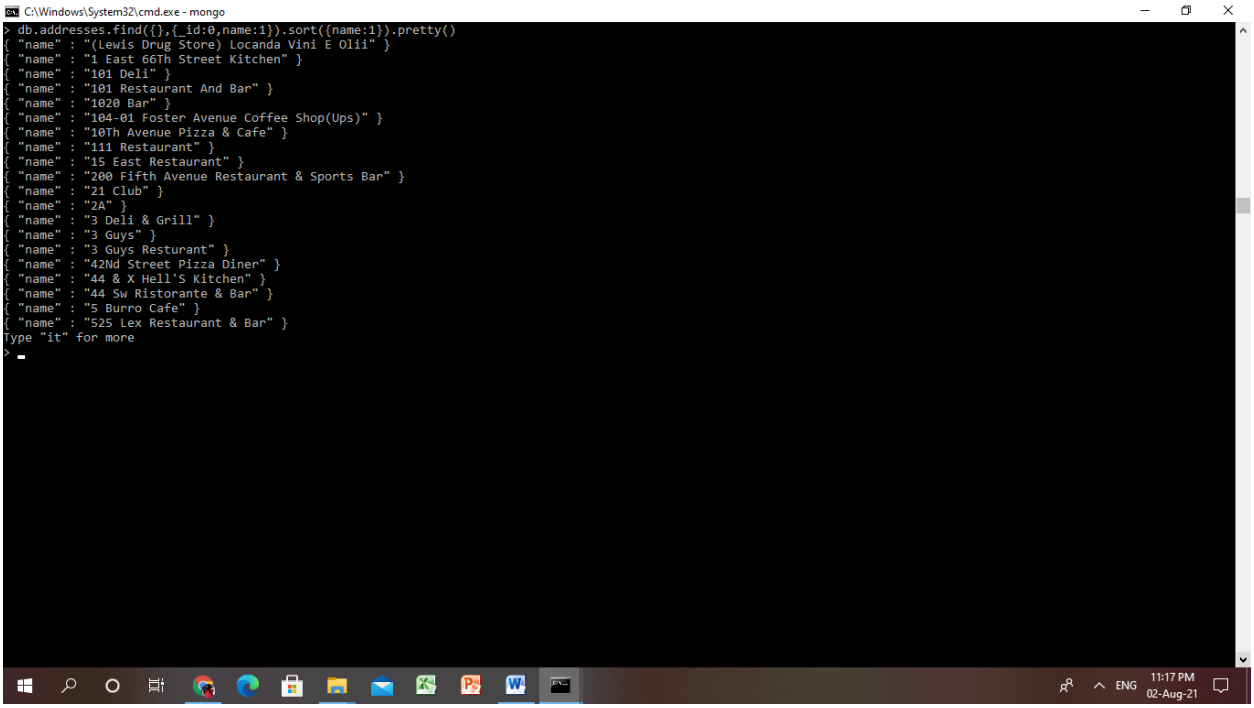
23)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({$and:[{"grades.1.grade":"A"},{"grades.1.score":9},{"grades.1.date":ISODate("2014-08-11T00:00:00Z")}]},{_id:0,restaurant_id:1,name:1,grades:1}).pretty()
{
"grades" : [
{
"date" : ISODate("2015-01-12T00:00:00Z"),
"grade" : "A",
"score" : 10
},
{
"date" : ISODate("2014-08-11T00:00:00Z"),
"grade" : "A",
"score" : 9
},
{
"date" : ISODate("2014-01-14T00:00:00Z"),
"grade" : "A",
"score" : 13
},
{
"date" : ISODate("2013-02-07T00:00:00Z"),
"grade" : "A",
"score" : 10
},
{
"date" : ISODate("2012-04-30T00:00:00Z"),
"grade" : "A",
"score" : 11
}
],
"name" : "Club Macanudo (Cigar Bar)",
"restaurant_id" : "40526406"
}
>
ENG
11:05 PM
02-Aug-21

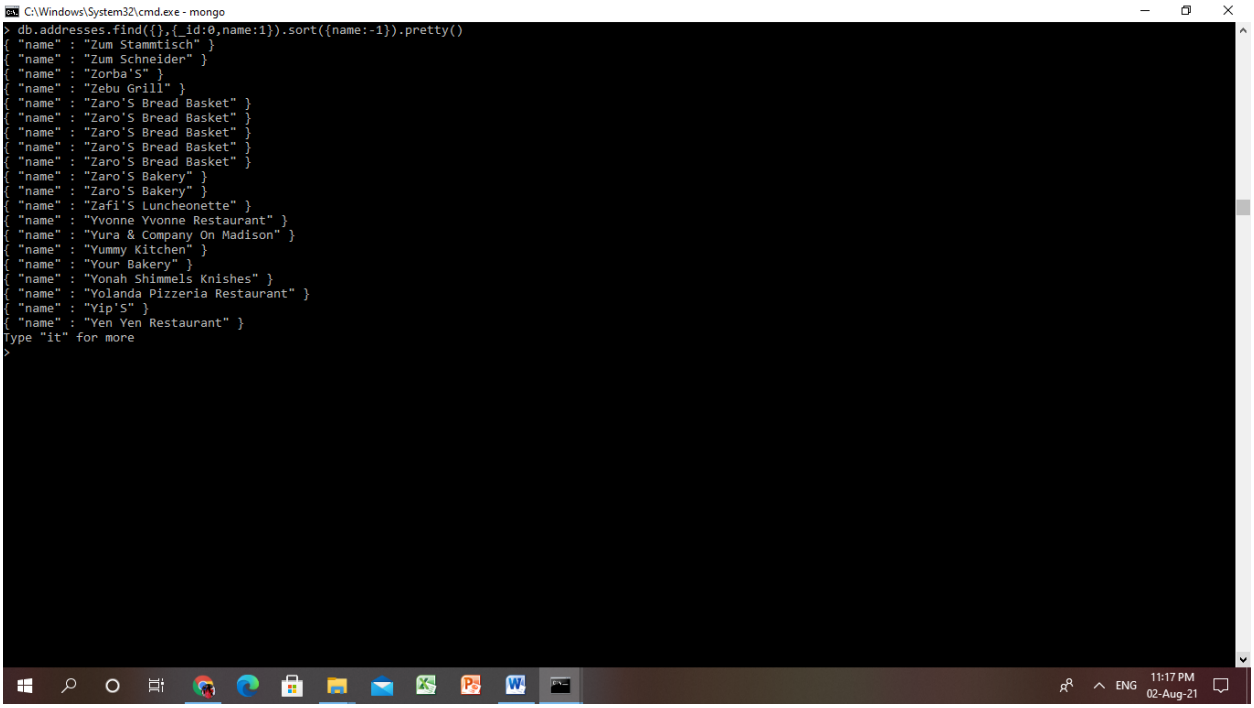
24)


C:\Windows\System32\cmd.exe - mongo
"restaurant_id" : "40526406"
}
> db.addresses.find({$and:[{"address.coord.1":{$gt:42}},{"address.coord.1":{$lte:52}}]},{_id:0,restaurant_id:1,name:1,address:1})
{ "address" : { "building" : "47", "coord" : [-78.877224, 42.89546199999999], "street" : "Broadway @ Trinity Pl", "zipcode" : "10006" }, "name" : "T.G.I. Friday'S", "restaurant_id" : "40387990" }
{ "address" : { "building" : "1", "coord" : [-0.7119979, 51.6514664], "street" : "Pennplaza E, Penn Sta", "zipcode" : "10001" }, "name" : "T.G.I. Fridays", "restaurant_id" : "40388936" }
{ "address" : { "building" : "3000", "coord" : [-87.86567699999999, 42.61150920000001], "street" : "47 Avenue", "zipcode" : "11101" }, "name" : "Di Luvio'S Deli", "restaurant_id" : "40402284" }
{ "address" : { "building" : "21972199", "coord" : [-78.589606, 42.8912372], "street" : "Broadway", "zipcode" : "10024" }, "name" : "La Caridad 78", "restaurant_id" : "40568285" }
{ "address" : { "building" : "7981", "coord" : [-84.9751215, 45.4713351], "street" : "Hoyt Street", "zipcode" : "11201" }, "name" : "Bijan'S", "restaurant_id" : "40876618" }
{ "address" : { "building" : "0", "coord" : [-88.0778799, 42.4154769], "street" : "& Grand Central", "zipcode" : "10017" }, "name" : "Hyatt, Ny Central/Room Service", "restaurant_id" : "40879243" }
{ "address" : { "building" : "60", "coord" : [-111.9975205, 42.0970258], "street" : "West Side Highway", "zipcode" : "10006" }, "name" : "Sports Center At Chelsea Piers (Sushi Bar)", "restaurant_id" : "40882356" }
> _
ENG
11:14 PM
02-Aug-21

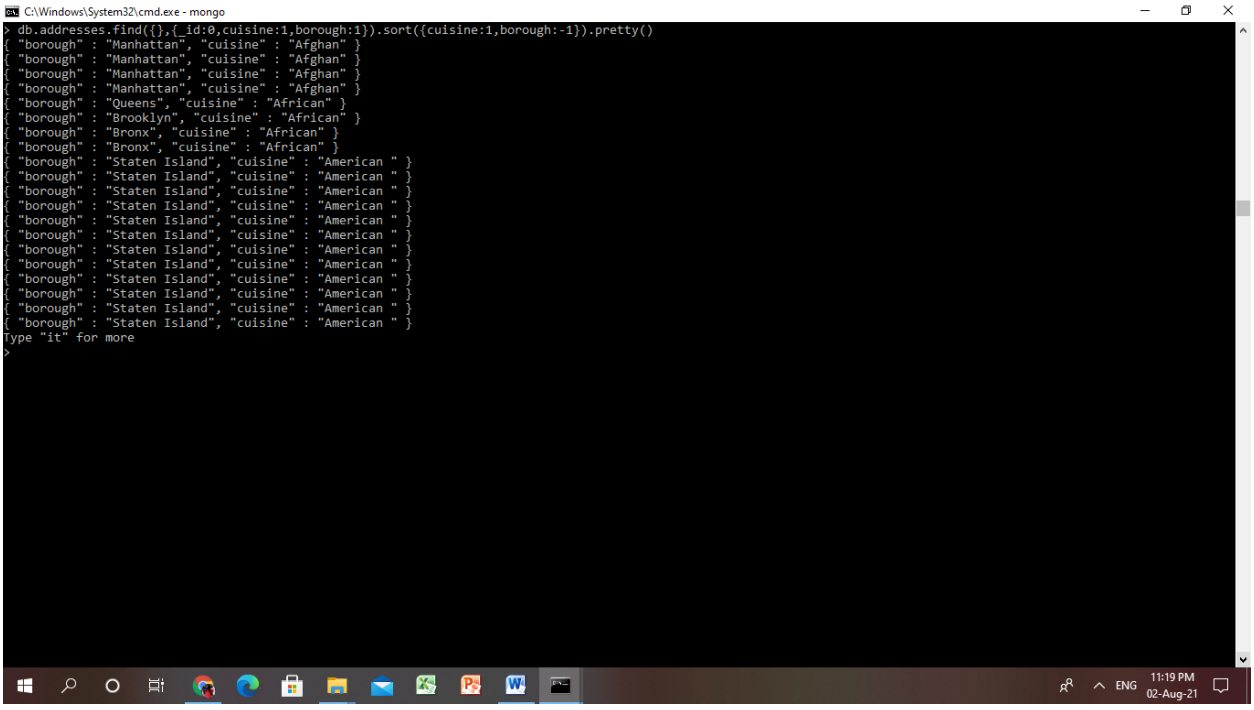
25)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({},{_id:0,name:1}).sort({name:1}).pretty()
{ "name" : "(Lewis Drug Store) Locanda Vini E Olii" }
{ "name" : "1 East 66Th Street Kitchen" }
{ "name" : "101 Deli" }
{ "name" : "101 Restaurant And Bar" }
{ "name" : "1020 Bar" }
{ "name" : "104-01 Foster Avenue Coffee Shop(Ups)" }
{ "name" : "10Th Avenue Pizza & Cafe" }
{ "name" : "111 Restaurant" }
{ "name" : "15 East Restaurant" }
{ "name" : "200 Fifth Avenue Restaurant & Sports Bar" }
{ "name" : "21 Club" }
{ "name" : "2A" }
{ "name" : "3 Deli & Grill" }
{ "name" : "3 Guys" }
{ "name" : "3 Guys Resturant" }
{ "name" : "42Nd Street Pizza Diner" }
{ "name" : "44 & X Hell'S Kitchen" }
{ "name" : "44 Sw Ristorante & Bar" }
{ "name" : "5 Burro Cafe" }
{ "name" : "525 Lex Restaurant & Bar" }
Type "it" for more
> _
ENG
11:17 PM
02-Aug-21

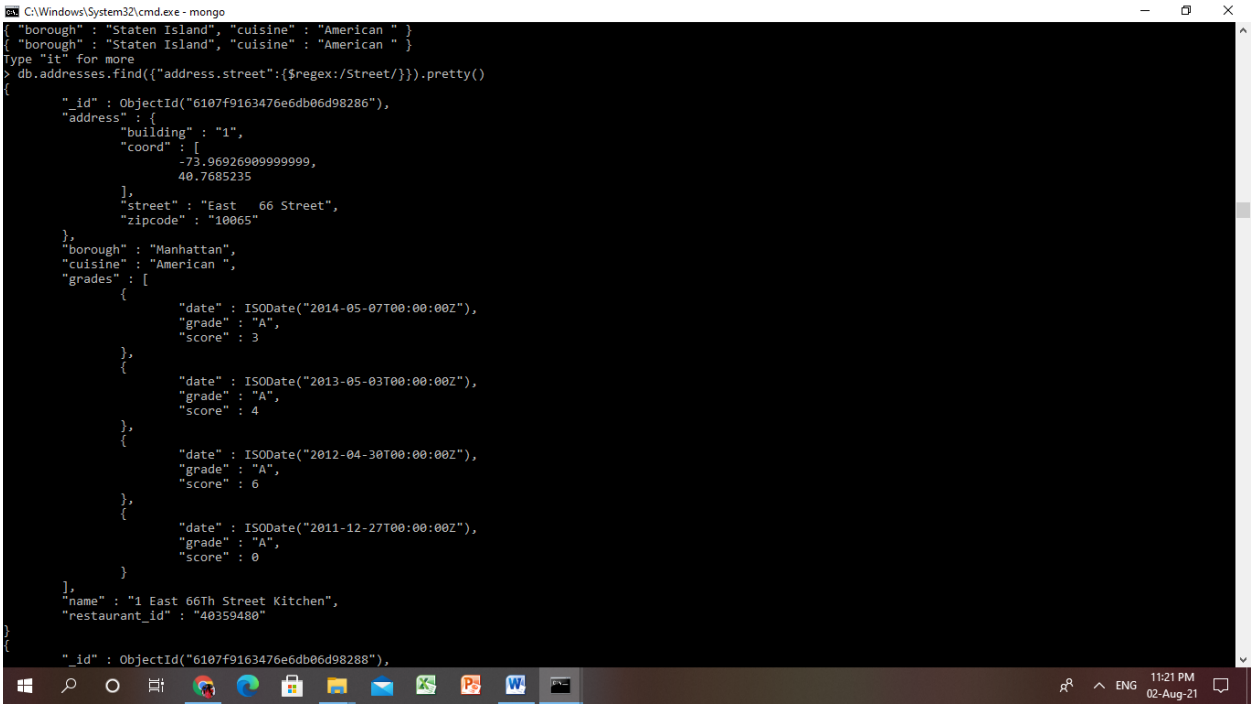
26)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({},{_id:0,name:1}).sort({name:-1}).pretty()
{ "name" : "Zum Stammtisch" }
{ "name" : "Zum Schneider" }
{ "name" : "Zorba'S" }
{ "name" : "Zebu Grill" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bakery" }
{ "name" : "Zaro'S Bakery" }
{ "name" : "Zafi'S Luncheonette" }
{ "name" : "Yvonne Yvonne Restaurant" }
{ "name" : "Yura & Company On Madison" }
{ "name" : "Yummy Kitchen" }
{ "name" : "Your Bakery" }
{ "name" : "Yonah Shimmels Knishes" }
{ "name" : "Yolanda Pizzeria Restaurant" }
{ "name" : "Yip'S" }
{ "name" : "Yen Yen Restaurant" }
Type "it" for more
>
ENG
11:17 PM
02-Aug-21

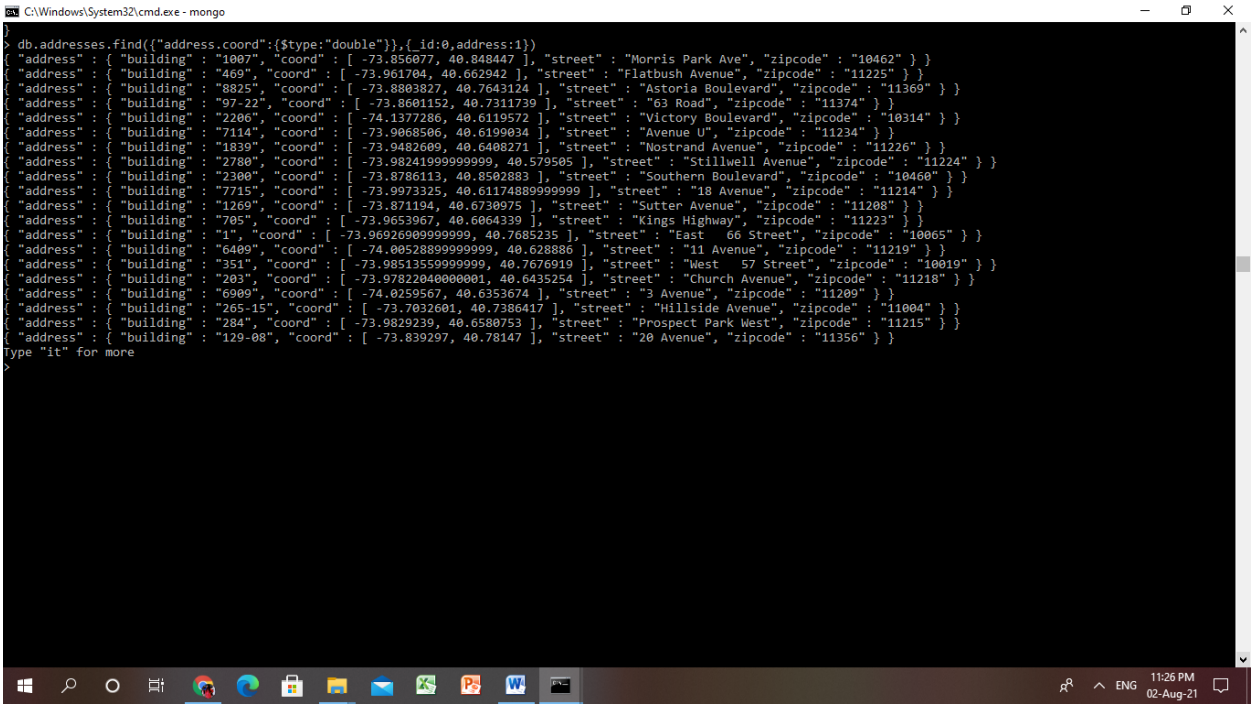
27)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({},{_id:0,cuisine:1,borough:1}).sort({cuisine:1,borough:-1}).pretty()
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Queens", "cuisine" : "African" }
{ "borough" : "Brooklyn", "cuisine" : "African" }
{ "borough" : "Bronx", "cuisine" : "African" }
{ "borough" : "Bronx", "cuisine" : "African" }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
Type "it" for more
>
ENG
11:19 PM
02-Aug-21

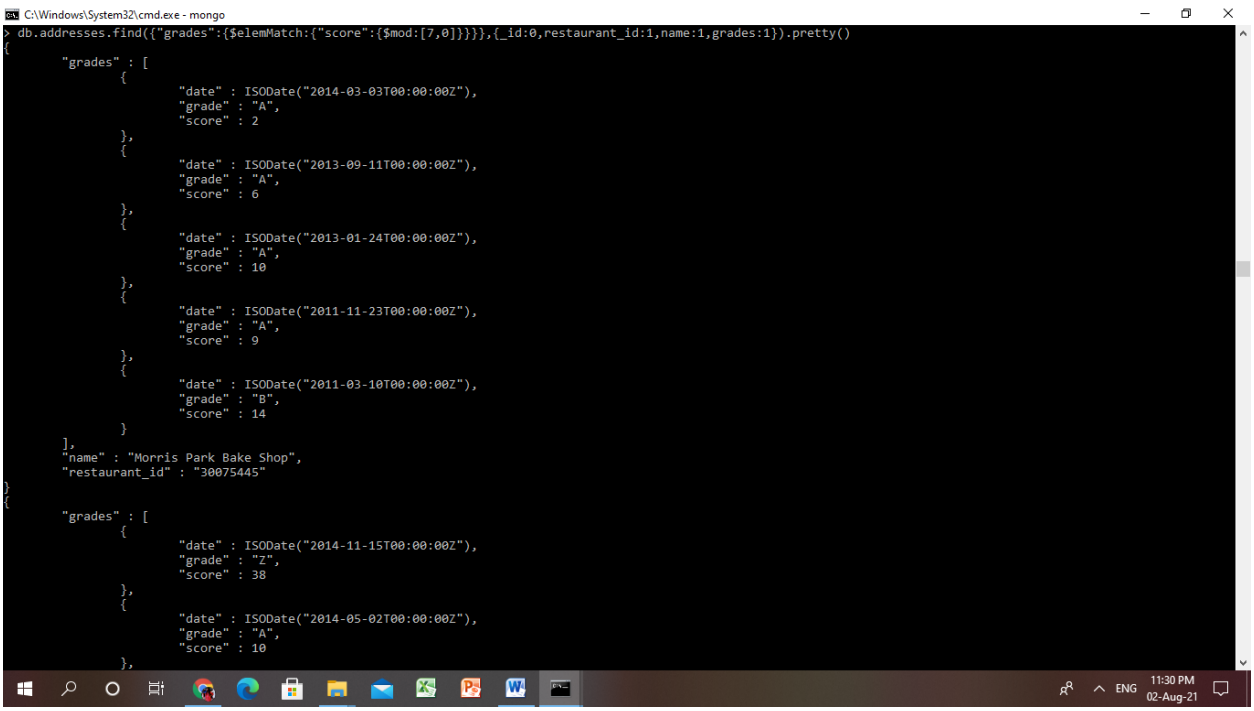
28)


C:\Windows\System32\cmd.exe - mongo
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
Type "it" for more
> db.addresses.find({"address.street":{$regex:/Street/}}).pretty()
{
"_id" : ObjectId("6107f9163476e6db06d98286"),
"address" : {
"building" : "1",
"coord" : [
-73.96926909999999,
40.7685235
],
"street" : "East 66 Street",
"zipcode" : "10065"
},
"borough" : "Manhattan",
"cuisine" : "American ",
"grades" : [
{
"date" : ISODate("2014-05-07T00:00:00Z"),
"grade" : "A",
"score" : 3
},
{
"date" : ISODate("2013-05-03T00:00:00Z"),
"grade" : "A",
"score" : 4
},
{
"date" : ISODate("2012-04-30T00:00:00Z"),
"grade" : "A",
"score" : 6
},
{
"date" : ISODate("2011-12-27T00:00:00Z"),
"grade" : "A",
"score" : 0
}
],
"name" : "1 East 66Th Street Kitchen",
"restaurant_id" : "40359480"
}
{
"_id" : ObjectId("6107f9163476e6db06d98288"),
ENG
11:21 PM
02-Aug-21

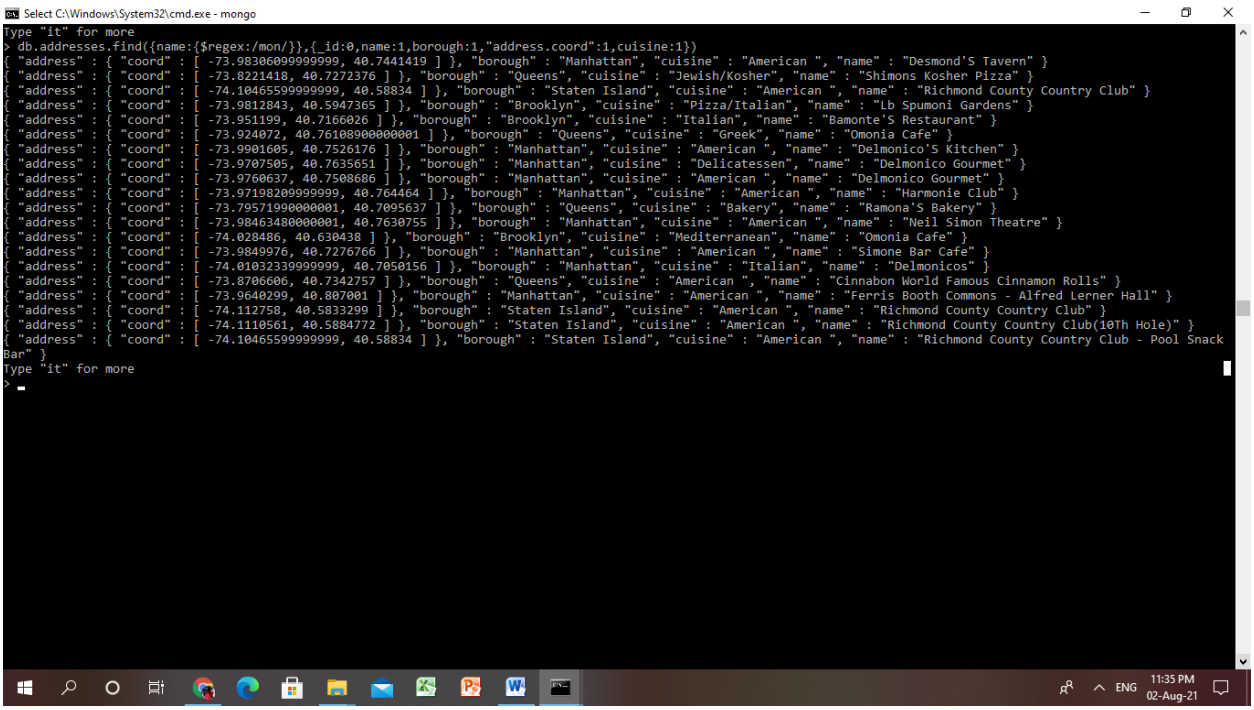
29)


C:\Windows\System32\cmd.exe - mongo
}
> db.addresses.find({"address.coord":{$type:"double"}},{_id:0,address:1})
{ "address" : { "building" : "1007", "coord" : [-73.856077, 40.848447], "street" : "Morris Park Ave", "zipcode" : "10462" } }
{ "address" : { "building" : "469", "coord" : [-73.961704, 40.662942], "street" : "Flatbush Avenue", "zipcode" : "11225" } }
{ "address" : { "building" : "8825", "coord" : [-73.8803827, 40.7643124], "street" : "Astoria Boulevard", "zipcode" : "11369" } }
{ "address" : { "building" : "97-22", "coord" : [-73.8601152, 40.7311739], "street" : "63 Road", "zipcode" : "11374" } }
{ "address" : { "building" : "2206", "coord" : [-74.1377286, 40.6119572], "street" : "Victory Boulevard", "zipcode" : "10314" } }
{ "address" : { "building" : "7114", "coord" : [-73.9068506, 40.6199034], "street" : "Avenue U", "zipcode" : "11234" } }
{ "address" : { "building" : "1839", "coord" : [-73.9482609, 40.6408271], "street" : "Nostrand Avenue", "zipcode" : "11226" } }
{ "address" : { "building" : "2780", "coord" : [-73.98241999999999, 40.579505], "street" : "Stillwell Avenue", "zipcode" : "11224" } }
{ "address" : { "building" : "2300", "coord" : [-73.8786113, 40.8502883], "street" : "Southern Boulevard", "zipcode" : "10460" } }
{ "address" : { "building" : "7715", "coord" : [-73.9973325, 40.61174889999999], "street" : "18 Avenue", "zipcode" : "11214" } }
{ "address" : { "building" : "1269", "coord" : [-73.871194, 40.6730975], "street" : "Sutter Avenue", "zipcode" : "11208" } }
{ "address" : { "building" : "705", "coord" : [-73.9653967, 40.6064339], "street" : "Kings Highway", "zipcode" : "11223" } }
{ "address" : { "building" : "1", "coord" : [-73.96926909999999, 40.7685235], "street" : "East 66 Street", "zipcode" : "10065" } }
{ "address" : { "building" : "6409", "coord" : [-74.00528899999999, 40.628886], "street" : "11 Avenue", "zipcode" : "11219" } }
{ "address" : { "building" : "351", "coord" : [-73.98513559999999, 40.7676919], "street" : "West 57 Street", "zipcode" : "10019" } }
{ "address" : { "building" : "203", "coord" : [-73.97822040000001, 40.6435254], "street" : "Church Avenue", "zipcode" : "11218" } }
{ "address" : { "building" : "6909", "coord" : [-74.0259567, 40.6353674], "street" : "3 Avenue", "zipcode" : "11209" } }
{ "address" : { "building" : "265-15", "coord" : [-73.7032601, 40.7386417], "street" : "Hillside Avenue", "zipcode" : "11004" } }
{ "address" : { "building" : "284", "coord" : [-73.9829239, 40.6580753], "street" : "Prospect Park West", "zipcode" : "11215" } }
{ "address" : { "building" : "129-08", "coord" : [-73.839297, 40.78147], "street" : "20 Avenue", "zipcode" : "11356" } }
Type "it" for more
>
ENG
11:26 PM
02-Aug-21

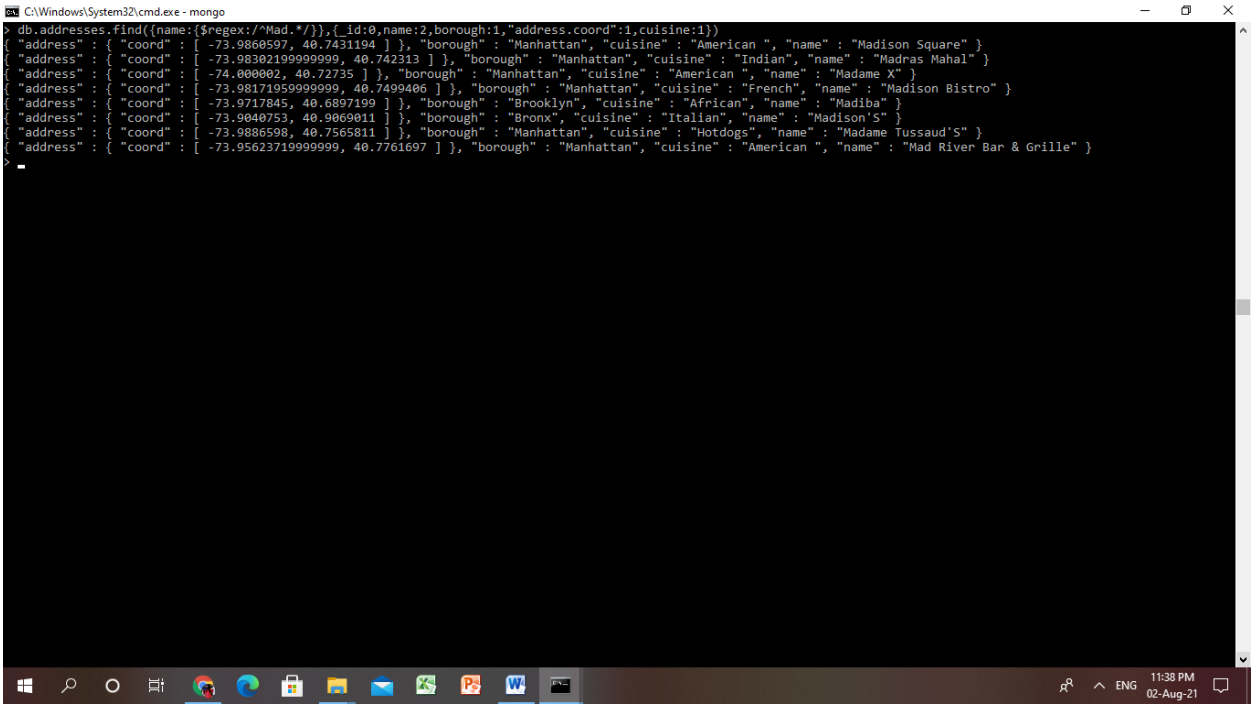
30)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({"grades":{$elemMatch:{"score":{$mod:[7,0]}}}},{_id:0,restaurant_id:1,name:1,grades:1}).pretty()
{
"grades" : [
{
"date" : ISODate("2014-03-03T00:00:00Z"),
"grade" : "A",
"score" : 2
},
{
"date" : ISODate("2013-09-11T00:00:00Z"),
"grade" : "A",
"score" : 6
},
{
"date" : ISODate("2013-01-24T00:00:00Z"),
"grade" : "A",
"score" : 10
},
{
"date" : ISODate("2011-11-23T00:00:00Z"),
"grade" : "A",
"score" : 9
},
{
"date" : ISODate("2011-03-10T00:00:00Z"),
"grade" : "B",
"score" : 14
}
],
"name" : "Morris Park Bake Shop",
"restaurant_id" : "30075445"
}
{
"grades" : [
{
"date" : ISODate("2014-11-15T00:00:00Z"),
"grade" : "Z",
"score" : 38
},
{
"date" : ISODate("2014-05-02T00:00:00Z"),
"grade" : "A",
"score" : 10
},
ENG
11:30 PM
02-Aug-21

31)


Select C:\Windows\System32\cmd.exe - mongo
Type "it" for more
> db.addresses.find({name:{$regex:/mon/}},{_id:0,name:1,borough:1,"address.coord":1,cuisine:1})
{ "address" : { "coord" : [-73.98306099999999, 40.7441419] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Desmond's Tavern" }
{ "address" : { "coord" : [-73.8221418, 40.7272376] }, "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Shimons Kosher Pizza" }
{ "address" : { "coord" : [-74.10465599999999, 40.58834] }, "borough" : "Staten Island", "cuisine" : "American ", "name" : "Richmond County Country Club" }
{ "address" : { "coord" : [-73.9812843, 40.5947365] }, "borough" : "Brooklyn", "cuisine" : "Pizza/Italian", "name" : "Lb Spumoni Gardens" }
{ "address" : { "coord" : [-73.951199, 40.7166026] }, "borough" : "Brooklyn", "cuisine" : "Italian", "name" : "Bamonte's Restaurant" }
{ "address" : { "coord" : [-73.924072, 40.76108900000001] }, "borough" : "Queens", "cuisine" : "Greek", "name" : "Omonia Cafe" }
{ "address" : { "coord" : [-73.9901605, 40.7526176] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Delmonico's Kitchen" }
{ "address" : { "coord" : [-73.9707505, 40.7635651] }, "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Delmonico Gourmet" }
{ "address" : { "coord" : [-73.9760637, 40.7508686] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Delmonico Gourmet" }
{ "address" : { "coord" : [-73.97198209999999, 40.764464] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Harmonie Club" }
{ "address" : { "coord" : [-73.79571990000001, 40.7095637] }, "borough" : "Queens", "cuisine" : "Bakery", "name" : "Ramona's Bakery" }
{ "address" : { "coord" : [-73.98463480000001, 40.7630755] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Neil Simon Theatre" }
{ "address" : { "coord" : [-74.028486, 40.630438] }, "borough" : "Brooklyn", "cuisine" : "Mediterranean", "name" : "Omonia Cafe" }
{ "address" : { "coord" : [-73.9849976, 40.7276766] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Simone Bar Cafe" }
{ "address" : { "coord" : [-74.01032339999999, 40.7050156] }, "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Delmonicos" }
{ "address" : { "coord" : [-73.8706606, 40.7342757] }, "borough" : "Queens", "cuisine" : "American ", "name" : "Cinnabon World Famous Cinnamon Rolls" }
{ "address" : { "coord" : [-73.9640299, 40.807001] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Ferris Booth Commons - Alfred Lerner Hall" }
{ "address" : { "coord" : [-74.112758, 40.5833299] }, "borough" : "Staten Island", "cuisine" : "American ", "name" : "Richmond County Country Club" }
{ "address" : { "coord" : [-74.1110561, 40.5884772] }, "borough" : "Staten Island", "cuisine" : "American ", "name" : "Richmond County Country Club(10Th Hole)" }
{ "address" : { "coord" : [-74.10465599999999, 40.58834] }, "borough" : "Staten Island", "cuisine" : "American ", "name" : "Richmond County Country Club - Pool Snack Bar" }
Type "it" for more
ENG
11:35 PM
02-Aug-21

32)


C:\Windows\System32\cmd.exe - mongo
> db.addresses.find({name:{$regex:/^Mad.*/}},{_id:0,name:2,borough:1,"address.coord":1,cuisine:1})
{ "address" : { "coord" : [-73.9860597, 40.7431194] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Madison Square" }
{ "address" : { "coord" : [-73.9830219999999, 40.742313] }, "borough" : "Manhattan", "cuisine" : "Indian", "name" : "Madras Mahal" }
{ "address" : { "coord" : [-74.000002, 40.72735] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Madame X" }
{ "address" : { "coord" : [-73.98171959999999, 40.7499406] }, "borough" : "Manhattan", "cuisine" : "French", "name" : "Madison Bistro" }
{ "address" : { "coord" : [-73.9717845, 40.6897199] }, "borough" : "Brooklyn", "cuisine" : "African", "name" : "Madiba" }
{ "address" : { "coord" : [-73.9040753, 40.9069011] }, "borough" : "Bronx", "cuisine" : "Italian", "name" : "Madison'S" }
{ "address" : { "coord" : [-73.9886598, 40.7565811] }, "borough" : "Manhattan", "cuisine" : "Hotdogs", "name" : "Madame Tussaud'S" }
{ "address" : { "coord" : [-73.95623719999999, 40.7761697] }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Mad River Bar & Grille" }
>
ENG
11:38 PM
02-Aug-21