ASSIGNMENT -1

#Query the movies collection to

---------------------------------------------------------------------

1)Get all Documents.

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find().pretty()

{

"\_id" : ObjectId("61063c401506b1ec4a887c50"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c51"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"Jhon Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c52"),

"title" : "Inglorious Baterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

{

"\_id" : ObjectId("610641661506b1ec4a887c53"),

"title" : "The Hobbit: An unexpected Journey",

"writer" : "J.R.R. Tolkein",

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641751506b1ec4a887c54"),

"title" : "The Hobbit: The Desolation of Smaug",

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."

}

{

"\_id" : ObjectId("6106418d1506b1ec4a887c56"),

"title" : "Pee Wee Herman's Big Adventures"

}

{ "\_id" : ObjectId("610641941506b1ec4a887c57"), "title" : "Avatar" }

-------------------------------------------------------------------

2)Get all documents with writer set to "Quentin tarantino"

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({writer:"Quentin Tarantino"}).pretty()

{

"\_id" : ObjectId("61063f451506b1ec4a887c51"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"Jhon Travolta",

"Uma Thurman"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c52"),

"title" : "Inglorious Baterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

--------------------------------------------------------------------------

3)Get All the docs where actors include bradd pitt.

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({actors:"Brad Pitt"}).pretty()

{

"\_id" : ObjectId("61063c401506b1ec4a887c50"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c52"),

"title" : "Inglorious Baterds",

"writer" : "Quentin Tarantino",

"year" : 2009,

"actors" : [

"Brad Pitt",

"Diane Kruger",

"Eli Roth"

]

}

-------------------------------------------------------

4)Get all the docs having fanchise as The Hobbit

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({franchise:"The Hobbit"}).pretty()

{

"\_id" : ObjectId("610641661506b1ec4a887c53"),

"title" : "The Hobbit: An unexpected Journey",

"writer" : "J.R.R. Tolkein",

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641751506b1ec4a887c54"),

"title" : "The Hobbit: The Desolation of Smaug",

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."

}

-----------------------------------------------------------

5) All movies released in 90s

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({year:{$gt:1990, $lt:2000}}).pretty()

{

"\_id" : ObjectId("61063c401506b1ec4a887c50"),

"title" : "Fight Club",

"writer" : "Chuck Palahniuko",

"year" : 1999,

"actors" : [

"Brad Pitt",

"Edward Norton"

]

}

{

"\_id" : ObjectId("61063f451506b1ec4a887c51"),

"title" : "Pulp Fiction",

"writer" : "Quentin Tarantino",

"year" : 1994,

"actors" : [

"Jhon Travolta",

"Uma Thurman"

]

}

-----------------------------------------------------------------

6)All movies before 2000 and after 2010.

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]}).pretty()

{

"\_id" : ObjectId("610641661506b1ec4a887c53"),

"title" : "The Hobbit: An unexpected Journey",

"writer" : "J.R.R. Tolkein",

"year" : "2012",

"franchise" : "The Hobbit"

}

{

"\_id" : ObjectId("610641751506b1ec4a887c54"),

"title" : "The Hobbit: The Desolation of Smaug",

"writer" : "J.R.R Tolkien",

"year" : "2013",

"franchise" : "The Hobbit"

}

---------------------------------------------------------------------------------

#Update Documents

1)add synopsis

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced to engage in a war against

an array of combatants and keep the Lonely

Mountain from falling into the hands of a rising darkness."

}

-------------------------------------------------------------------------------

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY>

db.movies.update({\_id:ObjectId("5c9fa42ae5c2dfe9b3729c03")},

{$set:{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 })

-------------------------------------------------------------------------------

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.update

({\_id:ObjectId("5c9f983ce5c2dfe9b3729bfc")}, {$push:{actors:"Samuel L. Jackson"}})

WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })

-------------------------------------------------------------------------------------

1)find all movies that have a synopsis that contains the word "Bilbo"

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({synopsis:{$regex:"Bilbo"}}).pretty()

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bibo and Company are forced to egage in a

war against an arrayof combatants and keep the Lonely Mountain from falling into

the hands of a rising darkness."

}

----------------------------------------------------------------------------------

2.find all movies that have a synopsis that contains the word "Gandalf"

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]}).pretty()

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."

}

-------------------------------------------------------------------------------------------

3.find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]}).pretty()

{

"\_id" : ObjectId("610641831506b1ec4a887c55"),

"title" : "The Hobbit: The Battle of the Five Armies",

"writer" : "J.R.R Tolkien",

"year" : "2002",

"franchise" : "The Hobbit",

"synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."

}

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]})

----------------------------------------------------------------------------------------------

find all movies that have a synopsis that contains the word "gold" and "dragon"

db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})

------------------------------------------------------------------------------------------

##Delete

delete the movie "Pee Wee Herman's Big Adventure"

db.movies.remove({\_id:ObjectId("6106418d1506b1ec4a887c56")})

-------------------------------------------------------------------------------------------

delete avatar

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.movies.remove({title:"Avatar"})

WriteResult({ "nRemoved" : 1 })

-----------------------------------------------------------------------------------------

#POSTS

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"})

WriteResult({ "nInserted" : 1 })

#CMNTS

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("5ca0b7e96435f98b5901f463")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("5ca0b9706435f98b5901f46a")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("5ca0b8766435f98b5901f467")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("5ca0b8546435f98b5901f466")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("5ca0b9256435f98b5901f469")})

WriteResult({ "nInserted" : 1 })

#USERS

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.users.insert({\_id:1,username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.users.insert({\_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> show collections

comments

posts

users

---------------------------------------------------------------

find all users

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.users.find().pretty()

{

"\_id" : 1,

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : 2,

"username" : "ScumbagSteve",

"fullname" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

--------------------------------------------------

find all posts

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.find().pretty()

{

"\_id" : ObjectId("6106a9c9c53f6e517e223323"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9d1c53f6e517e223324"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9dac53f6e517e223325"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request"

}

{

"\_id" : ObjectId("6106a9e2c53f6e517e223326"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("6106a9e8c53f6e517e223327"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("6106a9f0c53f6e517e223328"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

--------------------------------------------------------------------

Posts authored by greg

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("6106a9c9c53f6e517e223323"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9d1c53f6e517e223324"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("6106a9dac53f6e517e223325"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request"

}

---------------------------------------------------------

posts authored by scumnagsteve

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.posts.find({username:"ScumbagSteve"}).pretty()

{

"\_id" : ObjectId("6106a9e2c53f6e517e223326"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("6106a9e8c53f6e517e223327"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("6106a9f0c53f6e517e223328"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

--------------------------------------------------------

all comments

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.find().pretty()

{

"\_id" : ObjectId("6106aa9ac53f6e517e223329"),

"username" : "GoodGuyGreg",

"comment" : "Hope you got a good deal!",

"post" : ObjectId("5ca0b7e96435f98b5901f463")

}

{

"\_id" : ObjectId("6106aaabc53f6e517e22332a"),

"username" : "GoodGuyGreg",

"comment" : "What's mine is yours!",

"post" : ObjectId("5ca0b9706435f98b5901f46a")

}

{

"\_id" : ObjectId("6106aab3c53f6e517e22332b"),

"username" : "GoodGuyGreg",

"comment" : "Don't violate the licensing agreement!",

"post" : ObjectId("5ca0b8766435f98b5901f467")

}

{

"\_id" : ObjectId("6106aabcc53f6e517e22332c"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("5ca0b8546435f98b5901f466")

}

{

"\_id" : ObjectId("6106aac5c53f6e517e22332d"),

"username" : "ScumbagSteve",

"comment" : "Denied your PR cause I found a hack",

"post" : ObjectId("5ca0b9256435f98b5901f469")

}

------------------------------------------------------------------

cmnts by good guy greg

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.find({username:"GoodGuyGreg"}).pretty()

{

"\_id" : ObjectId("6106aa9ac53f6e517e223329"),

"username" : "GoodGuyGreg",

"comment" : "Hope you got a good deal!",

"post" : ObjectId("5ca0b7e96435f98b5901f463")

}

{

"\_id" : ObjectId("6106aaabc53f6e517e22332a"),

"username" : "GoodGuyGreg",

"comment" : "What's mine is yours!",

"post" : ObjectId("5ca0b9706435f98b5901f46a")

}

{

"\_id" : ObjectId("6106aab3c53f6e517e22332b"),

"username" : "GoodGuyGreg",

"comment" : "Don't violate the licensing agreement!",

"post" : ObjectId("5ca0b8766435f98b5901f467")

}

------------------------------------------------------------

cmnts by scumbag

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.comments.find({username:"ScumbagSteve"}).pretty()

{

"\_id" : ObjectId("6106aabcc53f6e517e22332c"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("5ca0b8546435f98b5901f466")

}

{

"\_id" : ObjectId("6106aac5c53f6e517e22332d"),

"username" : "ScumbagSteve",

"comment" : "Denied your PR cause I found a hack",

"post" : ObjectId("5ca0b9256435f98b5901f469")

}

db.comments.find({\_id:ObjectId("6106a9dac53f6e517e223325")}).pretty()

Assignment-2

Use db.zipcode.find() to get all with city atlanta,state GA

db.zipcode.find({city:"ATLANTA",state:"GA"}).pretty()

{

"\_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"

}............

--------------------------------------------------------------

Use aggregate for the above query

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.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" : "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" : "30312", "city" : "ATLANTA", "loc" : [ -84.378125, 33.746749 ], "pop" : 17683, "state" : "GA" }

{ "\_id" : "30313", "city" : "ATLANTA", "loc" : [ -84.39352, 33.76825 ], "pop" : 8038, "state" : "GA" }

{ "\_id" : "30314", "city" : "ATLANTA", "loc" : [ -84.425546, 33.756103 ], "pop" : 26649, "state" : "GA" }

{ "\_id" : "30315", "city" : "ATLANTA", "loc" : [ -84.380771, 33.705062 ], "pop" : 41061, "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" : "30319", "city" : "ATLANTA", "loc" : [ -84.335091, 33.868728 ], "pop" : 32138, "state" : "GA" }

{ "\_id" : "30324", "city" : "ATLANTA", "loc" : [ -84.354867, 33.820609 ], "pop" : 15044, "state" : "GA" }

{ "\_id" : "30326", "city" : "ATLANTA", "loc" : [ -84.358232, 33.848168 ], "pop" : 125, "state" : "GA" }

{ "\_id" : "30327", "city" : "ATLANTA", "loc" : [ -84.419966, 33.862723 ], "pop" : 18467, "state" : "GA" }

{ "\_id" : "30329", "city" : "ATLANTA", "loc" : [ -84.321402, 33.823555 ], "pop" : 17013, "state" : "GA" }

Type "it" for more

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> it

{ "\_id" : "30330", "city" : "ATLANTA", "loc" : [ -84.434735, 33.70645 ], "pop" : 643, "state" : "GA" }

{ "\_id" : "30331", "city" : "ATLANTA", "loc" : [ -84.520468, 33.72241 ], "pop" : 38185, "state" : "GA" }

{ "\_id" : "30334", "city" : "ATLANTA", "loc" : [ -84.388188, 33.74715 ], "pop" : 0, "state" : "GA" }

{ "\_id" : "30336", "city" : "ATLANTA", "loc" : [ -84.510028, 33.78534 ], "pop" : 1228, "state" : "GA" }

{ "\_id" : "30339", "city" : "ATLANTA", "loc" : [ -84.462879, 33.87125 ], "pop" : 11158, "state" : "GA" }

{ "\_id" : "30342", "city" : "ATLANTA", "loc" : [ -84.376091, 33.884245 ], "pop" : 19057, "state" : "GA" }

{ "\_id" : "30345", "city" : "ATLANTA", "loc" : [ -84.286961, 33.851347 ], "pop" : 19825, "state" : "GA" }

{ "\_id" : "30346", "city" : "ATLANTA", "loc" : [ -84.333354, 33.926717 ], "pop" : 18, "state" : "GA" }

{ "\_id" : "30349", "city" : "ATLANTA", "loc" : [ -84.481258, 33.605331 ], "pop" : 48116, "state" : "GA" }

{ "\_id" : "30350", "city" : "ATLANTA", "loc" : [ -84.341146, 33.979471 ], "pop" : 24573, "state" : "GA" }

{ "\_id" : "30360", "city" : "ATLANTA", "loc" : [ -84.271645, 33.937772 ], "pop" : 16023, "state" : "GA" }

------------------------------------------------

NUM OF PINCODES IN ATLANTA

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([{$match:{city:"ATLANTA"}} ,

{$group:{\_id:{pin:"$\_id",city:"$city"}}}, {$count:"Code"} ])

{ "Code" : 41 }

-------------------------------------------------------------

Population in atlanta

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([ {$match:{city:"ATLANTA"}}, {$group:{\_id:"$city",total:{$sum:"$pop"}}} ])

{ "\_id" : "ATLANTA", "total" : 630046 }

-------------------------------------------------------------------

#Pop by State

1)pop for each state

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([ {$group:{\_id:"$state",total:{$sum:"$pop"}}} ])

{ "\_id" : "MA", "total" : 6016425 }

{ "\_id" : "TN", "total" : 4876457 }

{ "\_id" : "KY", "total" : 3675484 }

{ "\_id" : "MT", "total" : 798948 }

{ "\_id" : "UT", "total" : 1722850 }

{ "\_id" : "SC", "total" : 3486703 }

{ "\_id" : "OH", "total" : 10846517 }

{ "\_id" : "DE", "total" : 666168 }

{ "\_id" : "MI", "total" : 9295297 }

{ "\_id" : "ID", "total" : 1006749 }

{ "\_id" : "IA", "total" : 2776420 }

{ "\_id" : "WI", "total" : 4891769 }

{ "\_id" : "MS", "total" : 2573216 }

{ "\_id" : "IL", "total" : 11427576 }

{ "\_id" : "TX", "total" : 16984601 }

{ "\_id" : "RI", "total" : 1003218 }

{ "\_id" : "LA", "total" : 4217595 }

{ "\_id" : "NC", "total" : 6628637 }

{ "\_id" : "MD", "total" : 4781379 }

{ "\_id" : "MN", "total" : 4372982 }

Type "it" for more-----------------------------------------------

2)pop for each state use sort highest first

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([ {$group:{\_id:"$state",total:{$sum:"$pop"}}},{$sort:{total:-1}} ])

{ "\_id" : "CA", "total" : 29754890 }

{ "\_id" : "NY", "total" : 17990402 }

{ "\_id" : "TX", "total" : 16984601 }

{ "\_id" : "FL", "total" : 12686644 }

{ "\_id" : "PA", "total" : 11881643 }

{ "\_id" : "IL", "total" : 11427576 }

{ "\_id" : "OH", "total" : 10846517 }

{ "\_id" : "MI", "total" : 9295297 }

{ "\_id" : "NJ", "total" : 7730188 }

{ "\_id" : "NC", "total" : 6628637 }

{ "\_id" : "GA", "total" : 6478216 }

{ "\_id" : "VA", "total" : 6181479 }

{ "\_id" : "MA", "total" : 6016425 }

{ "\_id" : "IN", "total" : 5544136 }

{ "\_id" : "MO", "total" : 5110648 }

{ "\_id" : "WI", "total" : 4891769 }

{ "\_id" : "TN", "total" : 4876457 }

{ "\_id" : "WA", "total" : 4866692 }

{ "\_id" : "MD", "total" : 4781379 }

{ "\_id" : "MN", "total" : 4372982 }

Type "it" for more

-------------------------------------------------------------------------------

3)limit the results for just first 3 results.What are the top 3 states in population

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([

...{$group:{\_id:"$state",total:{$sum:"$pop"}}},{$sort:{total:-1}},{$limit:3} ])

{ "\_id" : "CA", "total" : 29754890 }

{ "\_id" : "NY", "total" : 17990402 }

{ "\_id" : "TX", "total" : 16984601 }

--------------------------------------------------------------------------------------

##

POPULATION BY CITY

##

1) use aggregate to calculate the total population for each city (u have to use city/state combination)

you can use a combination for the \_id of the $group.

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([

...{$group:{\_id:{city:"$city",state:"$state"},total:{$sum:"$pop"}}} ])

{ "\_id" : { "city" : "OLMSTED FALLS", "state" : "OH" }, "total" : 15722 }

{ "\_id" : { "city" : "HUBBARD LAKE", "state" : "MI" }, "total" : 1359 }

{ "\_id" : { "city" : "HAMILTON", "state" : "GA" }, "total" : 1587 }

{ "\_id" : { "city" : "EAST ORANGE", "state" : "NJ" }, "total" : 74355 }

{ "\_id" : { "city" : "HASLET", "state" : "TX" }, "total" : 866 }

{ "\_id" : { "city" : "SPRING GROVE", "state" : "IL" }, "total" : 2783 }

{ "\_id" : { "city" : "ORLAND", "state" : "IN" }, "total" : 1326 }

{ "\_id" : { "city" : "BEAR CREEK", "state" : "AL" }, "total" : 3748 }

{ "\_id" : { "city" : "MIRAMAR", "state" : "FL" }, "total" : 54274 }

{ "\_id" : { "city" : "FAIRDALE", "state" : "KY" }, "total" : 8297 }

{ "\_id" : { "city" : "SPRING VALLEY", "state" : "IL" }, "total" : 5541 }

{ "\_id" : { "city" : "GRAVOIS MILLS", "state" : "MO" }, "total" : 5653 }

{ "\_id" : { "city" : "HESTAND", "state" : "KY" }, "total" : 386 }

{ "\_id" : { "city" : "ATTICA", "state" : "IN" }, "total" : 5523 }

{ "\_id" : { "city" : "WARNER", "state" : "NH" }, "total" : 3265 }

{ "\_id" : { "city" : "WRIGHT", "state" : "AR" }, "total" : 302 }

{ "\_id" : { "city" : "YUMA", "state" : "AZ" }, "total" : 57131 }

{ "\_id" : { "city" : "HOPE", "state" : "RI" }, "total" : 3653 }

{ "\_id" : { "city" : "NEW BRITAIN", "state" : "CT" }, "total" : 75498 }

{ "\_id" : { "city" : "FINLEYVILLE", "state" : "PA" }, "total" : 8838 }

Type "it" for more

----------------------------------------------------------------------------

sort the results by pop,highest first

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([ {$group:{\_id:{city:"$city",state:"$state"},total:{$sum:"$pop"}}},{$sort:{total:-1}} ])

{ "\_id" : { "city" : "CHICAGO", "state" : "IL" }, "total" : 2452177 }

{ "\_id" : { "city" : "BROOKLYN", "state" : "NY" }, "total" : 2300504 }

{ "\_id" : { "city" : "LOS ANGELES", "state" : "CA" }, "total" : 2102295 }

{ "\_id" : { "city" : "HOUSTON", "state" : "TX" }, "total" : 2095918 }

{ "\_id" : { "city" : "PHILADELPHIA", "state" : "PA" }, "total" : 1610956 }

{ "\_id" : { "city" : "NEW YORK", "state" : "NY" }, "total" : 1476790 }

{ "\_id" : { "city" : "BRONX", "state" : "NY" }, "total" : 1209548 }

{ "\_id" : { "city" : "SAN DIEGO", "state" : "CA" }, "total" : 1049298 }

{ "\_id" : { "city" : "DETROIT", "state" : "MI" }, "total" : 963243 }

{ "\_id" : { "city" : "DALLAS", "state" : "TX" }, "total" : 940191 }

{ "\_id" : { "city" : "PHOENIX", "state" : "AZ" }, "total" : 890853 }

{ "\_id" : { "city" : "MIAMI", "state" : "FL" }, "total" : 825232 }

{ "\_id" : { "city" : "SAN JOSE", "state" : "CA" }, "total" : 816653 }

{ "\_id" : { "city" : "SAN ANTONIO", "state" : "TX" }, "total" : 811792 }

{ "\_id" : { "city" : "BALTIMORE", "state" : "MD" }, "total" : 733081 }

{ "\_id" : { "city" : "SAN FRANCISCO", "state" : "CA" }, "total" : 723993 }

{ "\_id" : { "city" : "MEMPHIS", "state" : "TN" }, "total" : 632837 }

{ "\_id" : { "city" : "SACRAMENTO", "state" : "CA" }, "total" : 628279 }

{ "\_id" : { "city" : "JACKSONVILLE", "state" : "FL" }, "total" : 610160 }

{ "\_id" : { "city" : "ATLANTA", "state" : "GA" }, "total" : 609591 }

Type "it" for more

---------------------------------------------------------------------------------------

3) limit the list to top 3

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([

...{$group:{\_id:{city:"$city",state:"$state"},total:{$sum:"$pop"}}},{$sort:{total:-1}},{$limit:3} ])

{ "\_id" : { "city" : "CHICAGO", "state" : "IL" }, "total" : 2452177 }

{ "\_id" : { "city" : "BROOKLYN", "state" : "NY" }, "total" : 2300504 }

{ "\_id" : { "city" : "LOS ANGELES", "state" : "CA" }, "total" : 2102295 }

---------------------------------------------------------------------------------------------------

4) top 3 pop cities in texas

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([

.. {$group:{\_id:{city:"$city",state:"TX"},total:{$sum:"$pop"}}},{$sort:{total:-1}},{$limit:3} ])

{ "\_id" : { "city" : "CHICAGO", "state" : "TX" }, "total" : 2452177 }

{ "\_id" : { "city" : "BROOKLYN", "state" : "TX" }, "total" : 2341387 }

{ "\_id" : { "city" : "HOUSTON", "state" : "TX" }, "total" : 2123053 }

--------------------------------------------------------------------------------------------------------------

##BONUS##

Avg pop of state

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([

... {

... "$group":{

... "\_id":"$state",

... "total":{"$avg":"$pop"}

... }

... }

... ])

{ "\_id" : "MD", "total" : 11384.235714285714 }

{ "\_id" : "GA", "total" : 10201.914960629922 }

{ "\_id" : "WY", "total" : 3239.4857142857145 }

{ "\_id" : "VT", "total" : 2315.8765432098767 }

{ "\_id" : "CA", "total" : 19627.236147757256 }

{ "\_id" : "OR", "total" : 7401.877604166667 }

{ "\_id" : "WA", "total" : 10055.148760330578 }

{ "\_id" : "MO", "total" : 5141.496981891348 }

{ "\_id" : "CT", "total" : 12498.539923954373 }

{ "\_id" : "NY", "total" : 11279.248902821317 }

{ "\_id" : "FL", "total" : 15779.407960199005 }

{ "\_id" : "ME", "total" : 2991.8243902439026 }

{ "\_id" : "NJ", "total" : 14315.162962962962 }

{ "\_id" : "IN", "total" : 8201.384615384615 }

{ "\_id" : "ND", "total" : 1632.4092071611253 }

{ "\_id" : "SD", "total" : 1810.9296875 }

{ "\_id" : "KS", "total" : 3461.937062937063 }

{ "\_id" : "AR", "total" : 4066.998269896194 }

{ "\_id" : "DC", "total" : 25287.5 }

{ "\_id" : "AK", "total" : 2793.3230769230768 }

Type "it" for more

----------------------------------------------------------------------

2)top 3 states in terms of avg population

MongoDB Enterprise atlas-xhli14-shard-0:PRIMARY> db.zipcode.aggregate([

... { "$group":{ "\_id":"$state", "total":{"$avg":"$pop"} } },{$sort:{"total":-1}},{$limit:3} ])

{ "\_id" : "DC", "total" : 25287.5 }

{ "\_id" : "CA", "total" : 19627.236147757256 }

{ "\_id" : "FL", "total" : 15779.407960199005 }

Assignment -3

1) db.address.find().pretty()

-----------------------------------

2)db.address.aggregate([{$project:{ restaurant\_id:1, name:1, borough:1, cuisine\_id:1}}])

--------------------------------------

3)db.address.aggregate([{$project:{ restaurant\_id:1, name:1, borough:1, cuisine\_id:1,\_id:0}}])

-------------------------------------------

4) db.address.aggregate([{$project:{ restaurant\_id:1, name:1, borough:1, zipcode:1,\_id:0}}])

----------------------------------------------

5)db.address.aggregate([ {$match:{borough:"Bronx"}},{$project:{name:1}},{$limit:5} ]).pretty()

-------------------------------------------------------

6)db.address.aggregate([ {$match:{borough:"Bronx"}},{$project:{name:1}}]).pretty()

-------------------------------------------------------

7) db.address.aggregate([ {$match:{borough:"Bronx"}},{$project:{name:1}},{$skip:5},

{$limit:5} ]).pretty()

----------------------------------------------------------

8) db.address.find({grades : { $elemMatch:{"score":{$gt : 90}}}}).pretty()

-----------------------------------------------------------

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

-----------------------------------------------------------

10)db.address.find({"address.coord" : {$lt : -95.754168}});

--------------------------------------------------------------

11) 12)

db.address.find(

{$and:

[

{"cuisine" : {$ne :"American "}},

{"grades.score" : {$gt : 70}},

{"address.coord" : {$lt : -65.754168}}

]

}

);

-----------------------------------------------------------

13)db.address.find( {

"cuisine" : {$ne : "American "},

"grades.grade" :"A",

"borough": {$ne : "Brooklyn"}

}

).sort({"cuisine":-1});

----------------------------------------------------------------

14)

db.address.find( {name: /^Wil/},

{ "restaurant\_id" : 1, "name":1,"borough":1, "cuisine" :1 } );

-----------------------------------------------------------------

15)

db.address.find(

{name: /ces$/},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

-------------------------------------------------------------------

16)

db.address.find(

{"name": /.\*Reg.\*/},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

--------------------------------------------------------------------

17)

db.address.find(

{

"borough": "Bronx" ,

$or : [

{ "cuisine" : "American " },

{ "cuisine" : "Chinese" }

]

}

);

-----------------------------------------------------------------------

18)db.address.find(

{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

-------------------------------------------------------------------------

19)db.address.find(

{"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

-----------------------------------------------------------------------------

20)db.address.find(

{"grades.score" :

{ $not:

{$gt : 10}

}

},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

-------------------------------------------------------------------------------

21)db.address.find(

{$or: [

{name: /^Wil/},

{"$and": [

{"cuisine" : {$ne :"American "}},

{"cuisine" : {$ne :"Chinees"}}

]}

]}

,{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1}

);

--------------------------------------------------------------------------------

22)

db.address.find(

{

"grades.date": ISODate("2014-08-11T00:00:00Z"),

"grades.grade":"A" ,

"grades.score" : 11

},

{"restaurant\_id" : 1,"name":1,"grades":1}

);

-----------------------------------------------------------------------------------

23)

db.address.find(

{ "grades.1.date": ISODate("2014-08-11T00:00:00Z"),

"grades.1.grade":"A" ,

"grades.1.score" : 9

},

{"restaurant\_id" : 1,"name":1,"grades":1}

);-

------------------------------------------------------------------------------------------

24)

db.address.find(

{

"address.coord.1": {$gt : 42, $lte : 52}

},

{"restaurant\_id" : 1,"name":1,"address":1,"coord":1}

);

--------------------------------------------------------------------------------------------

25)db.address.find().sort({"name":1});

--------------------------------------------------------------------------------------------

26)

db.address.find().sort(

{"name":-1}

);

-----------------------------------------------------------------------------------------

27)db.address.find().sort(

{"cuisine":1,"borough" : -1,}

);

-------------------------------------------------------------------------------------------

28)

db.address.find(

{"address.street" :

{ $exists : true }

}

);

-----------------------------------------------------------------------------------------

30)

db.address.find(

{"grades.score" :

{$mod : [7,0]}

},

{"restaurant\_id" : 1,"name":1,"grades":1}

);

----------------------------------------------------------------------------------------------

29)

db.address.find(

{"address.coord" :

{$type : 1}

}

);

31)

db.address.find(

{ name :

{ $regex : "mon.\*", $options: "i" }

},

{

"name":1,

"borough":1,

"address.coord":1,

"cuisine" :1

}

);

------------------------------------------------------------------------------------

32)

db.address.find(

{ name :

{ $regex : /^Mad/i, }

},

{

"name":1,

"borough":1,

"address.coord":1,

"cuisine" :1

}

);