# MongoDB Practice

MongoDB Exercise in mongo shell

**Lab Exercise 1:**

**Import a csv to mongodb**

{ "\_id" : "02906", "city" : "PROVIDENCE", "pop" : 31069, "state" : "RI", "capital" : { "name" : "Providence", "electoralCollege" : 4 } },

{ "\_id" : "02108", "city" : "BOSTON", "pop" : 3697, "state" : "MA", "capital" : { "name" : "Boston", "electoralCollege" : 11 } },

{ "\_id" : "10001", "city" : "NEW YORK", "pop" : 18913, "state" : "NY", "capital" : { "name" : "Albany", "electoralCollege" : 29 } },

{ "\_id" : "01012", "city" : "CHESTERFIELD", "pop" : 177, "state" : "MA", "capital" : { "name" : "Boston", "electoralCollege" : 11 } },

{ "\_id" : "32801", "city" : "ORLANDO", "pop" : 9275, "state" : "FL", "capital" : { "name" : "Tallahassee", "electoralCollege" : 29 } },

{ "\_id" : "12966", "city" : "BANGOR", "pop" : 2867, "state" : "NY", "capital" : { "name" : "Albany", "electoralCollege" : 29 } },

{ "\_id" : "32920", "city" : "CAPE CANAVERAL", "pop" : 7655, "state" : "FL", "capital" : { "name" : "Tallahassee", "electoralCollege" : 29 } },

{ "\_id" : "NY", "name" : "New York", "pop" : 28300000, "state" : 1788 },

{ "\_id" : "33125", "city" : "MIAMI", "pop" : 47761, "state" : "FL", "capital" : { "name" : "Tallahassee", "electoralCollege" : 29 } },

{ "\_id" : "RI", "name" : "Rhode Island", "pop" : 1060000, "state" : 1790 },

{ "\_id" : "MA", "name" : "Massachusetts", "pop" : 6868000, "state" : 1790 },

{ "\_id" : "FL", "name" : "Florida", "pop" : 6800000, "state" : 1845 },

{ "\_id" : "1", "name" : "Tom", "addresses" : [ "01001", "12997" ] },

{ "\_id" : "02907", "city" : "CRANSTON", "pop" : 25668, "state" : "RI", "capital" : { "name" : "Providence", "electoralCollege" : 4 } },

{ "\_id" : "2", "name" : "Bill", "addresses" : [ "01001", "12967", "32920" ] },

{ "\_id" : "3", "name" : "Mary", "addresses" : [ "32801", "32920", "33125" ] },

{ "\_id" : "12967", "city" : "NORTH LAWRENCE", "pop" : 943, "state" : "NY", "capital" : { "name" : "Albany", "electoralCollege" : 29 } },

{ "\_id" : "01001", "city" : "AGAWAM", "pop" : 15338, "state" : "MA", "capital" : { "name" : "Boston", "electoralCollege" : 11 } },

{ "\_id" : "12997", "city" : "WILMINGTON", "pop" : 958, "state" : "NY", "capital" : { "name" : "Albany", "electoralCollege" : 29 } }

To import from shell:

mongoimport --db <database\_name> --collection <collection\_name> --file

**Questions:**

1. Show name and population of the cities where the population is over 10000

Ans) db.lab1.find(

{pop :{$gt: 10000}},

{\_id:0,city:1,pop:1}

)

1. Show the name and population of the state based on the cities shown

Ans) db.lab1.find(  
{},

{\_id:0,city:1,pop:1,state:1})

1. Show the total cities in NY as 'Population'

Ans) db.lab1.aggregate([

{ $match :{state: "NY"}},

{ $group : {\_id:"NY", totalPopulation: {$sum: "$pop"}}} ])

1. Show the \_id, city, name of the capital city of each state with a popultaion greater than 20,000.

Ans) db.lab1.find(

{ "pop" : {$gt : 20000},

"capital" : { $exists: true }},

{ \_id :1,

city : 1,

"capital.name":1})

**Lab Exercise 2:**

## create database

Connect to a running mongo instance, use a database named mongo\_practice. use mongo\_practice

## Insert Documents

Insert the following documents into a movies collection.

db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})

db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"2009", actors:["John Travolta", "Uma Thurman"]})

db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"2009", actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})

db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkein", year:"2012",franchise:"The Hobbit"})

db.movies.insert({title:"The Hobbit: The Desolation of Smaug", writer:"J.R.R Tolkien", year:"2013", franchise:"The Hobbit"})

db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", franchise:"The Hobbit", synopsis:"Bilbo and Company are forced 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.insert({title:"Pee Wee Herman's Big Adventures"})

db.movies.insert({title:"Avatar"})

**Query / Find Documents**

query the movies collection to

1. get all documents

Ans) db.movies.find()

1. get all documents with writer set to "Quentin Tarantino"

Ans) db.movies.find({

writer:"Quentin Tarantino"})

1. get all documents where actors include "Brad Pitt"

Ans) db.movies.find({

actors: "Brad Pitt"})

1. get all documents with franchise set to "The Hobbit"

Ans) db.movies.find({

franchise: "The Hobbit"})

1. get all movies released in the 90s

Ans) db.movies.find(

{year : {$gte:"1990",$lte:"1999"}}

)

1. get all movies released before the year 2000 or after 2010

Ans) db.movies.find({

$or:[

{year: {$lt:"2000"}},

**Update Documents**

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

Ans) db.movies.updateOne(

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

{$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})

1. add a synopsis to "The Hobbit: The Desolation of Smaug" : "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."

Ans) db.movies.updateOne({title:"The Hobbit: The Desolation of Smaug"},

{$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."}})

1. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

Ans) db.movies.updateOne({title: "Pulp Fiction"},

{$push : {actors:"Samuel L. Jackson"}})

**Text Search**

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

db.movies.find({

synopsis: {$regex:"Bilbo",$options:"i"}})

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

ans) db.movies.find({

synopsis: {$regex:"Gandalf",$options:"i"}})

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

ans) db.movies.find({

synopsis: {$regex:"Bilbo",$options:"i"},

synopsis:{ $not: {$regex:"Gandalf",$options:"i"}}})

1. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"
2. find all movies that have a synopsis that contains the word "gold" and "dragon"

**Delete Documents**

1. delete the movie "Pee Wee Herman's Big Adventure"
2. delete the movie "Avatar"