

EXERCISE 1

INSERT DOCUMENTS show

db

admin 0.000GB

config 0.000GB

local 0.000GB

mongo_practice 0.000GB > use mongo_practice switched to

dbmongo_practice >db.movies.find.pretty() uncaught exception:

TypeError: db.movies.find.pretty is not a function :

@(shell):1:1

>db.movies.find()

```
{ "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"), "title" : "fight club", "writer" : "Chuck Palahniuko", "year" : 1999, "actors" : [ "Brad Pitt", "Edward Norton" ] }
```

```
{ "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"), "title" : "pulp fiction", "writer" : "Quentin Tarantino", "year" : 1994, "actors" : [ "John Travolta", "Uma Thurman" ] }
```

```
{ "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"), "title" : "inglorious basterds", "writer" : "Quentin Tarantino", "year" : 2009, "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }
```

```
{ "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"), "title" : "The Hobbit:An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : 2012, "franchise" : "The Hobbit" }
```

>db.movies.find().pretty()

```
{
  "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"),
  "title" : "fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
```

```
{
  "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
```

```
{
  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit: An unexpected Journey",
  "writer" : "J.R.R. Tolkien",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
```

```
>db.movies.insert({title:"TheHobbit:The Desolation of Smaug", writer:"J.R.R. Tolkien", year:2013, franchise:"The Hobbit"}) WriteResult({
  "nInserted" : 1 })
```

```
>db.movies.insert({title:"TheHobbit:The Battle of five Armies", writer:"J.R.R. Tolkien", year:2012, franchise:"The Hobbit", synopsis:"Bilbo
and company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a
rising darkness"})
```

```
WriteResult({ "nInserted" : 1 })
```

```
>db.movies.insert({title:"Pee Wee Herman's Big Adventure"})
```

```
WriteResult({ "nInserted" : 1 })
```

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

```
WriteResult({ "nInserted" : 1 })
```

GET ALL DOCUMENT'S

```
db.movies.find().pretty()
```

```
{
  "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"),
  "title" : "Fight Club",
  "writer" : "Chuck Palahniuk",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
```

```

}
{
  "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"),
  "title" : "inglorious basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
{
  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6105493820ea11fa5a5dd067"),
  "title" : "The Hobbit:The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("610549df20ea11fa5a5dd068"),

```

INDRAJEET ROTE – MongoDB Assignment

```
"title" : "The Hobbit:The Battle of five Armies",
"writer" : "J.R.R. Tolkein",
"year" : 2012,
"franchise" : "The Hobbit",
"synopsis" : "Biblo 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("61054a1320ea11fa5a5dd069"),
  "title" : "Pee Wee Herman's Big Adventure"
}
{ "_id" : ObjectId("61054a1f20ea11fa5a5dd06a"), "title" : "Avatar" }
```

```
db.movies.find({writer:"Quentin Tarantino"}).pretty()
```

```
{
  "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"),
  "title" : "inglorious basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
```

```
db.movies.find({actors:{$in:["Brad Pitt"]}}).pretty()
```

```
{
  "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"),
```

```

    "title" : "fight club",
    "writer" : "Chuck Palahniuko",
    "year" : 1999,
    "actors" : [
        "Brad Pitt",
        "Edward Norton"
    ]
}
{
    "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"),
    "title" : "inglorious basterds",
    "writer" : "Quentin Tarantino",
    "year" : 2009,
    "actors" : [
        "Brad Pitt",
        "Diane Kruger",
        "Eli Roth"
    ]
}

```

```

db.movies.find({franchise:"The Hobbit"}).pretty()
{
    "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
    "title" : "The Hobbit:An unexpected Journey",
    "writer" : "J.R.R. Tolkein",
    "year" : 2012,
    "franchise" : "The Hobbit"
}
{
    "_id" : ObjectId("6105493820ea11fa5a5dd067"),
    "title" : "The Hobbit:The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2013,
    "franchise" : "The Hobbit"
}
{
    "_id" : ObjectId("610549df20ea11fa5a5dd068"),
    "title" : "The Hobbit:The Battle of five Armies",

```

INDRAJEET ROTE – MongoDB Assignment

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "Biblo 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.find({year:{<2000}}).pretty()
```

```
{
  "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"),
  "title" : "fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
```

```
{
  "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
```

```
db.movies.find({$or:[{year:{<2000}},{year:{>2010}}]}).pretty()
```

```
{
  "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"),
  "title" : "fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
```

INDRAJEET ROTE – MongoDB Assignment

```
}
{
  "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("6105493820ea11fa5a5dd067"),
  "title" : "The Hobbit:The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit"
}
{
  "_id" : ObjectId("610549df20ea11fa5a5dd068"),
  "title" : "The Hobbit:The Battle of five Armies",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "Biblo 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.update({title:"TheHobbit:An Unexpected Journey"},{$set:{synopsis:"A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted 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:"TheHobbit:The Desolation of Smaug"},{$set:{synopsis:"The dwarves along with biblobaggins and gandalf the grey, continue their quest to reclaim Erebor their homesland from SmaugBiblo Baggins is in possession of a mysterious and magical ring"}})
```

INDRAJEET ROTE – MongoDB Assignment

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 }) db.movies.update({title:"pulp fiction"},{$set:{actors:["Samuel L Jackson"]}}) WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 }) db.movies.find().pretty()
```

```
{
  "_id" : ObjectId("6104fa96e9faf8b7ac9daae4"),
  "title" : "fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
  "title" : "pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "Samuel L Jackson"
  ]
}
{
  "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"),
  "title" : "inglorious basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "Brad Pitt",
    "Diane Kruger",
    "Eli Roth"
  ]
}
{
  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
}
```


INDRAJEET ROTE – MongoDB Assignment

"synopsis" : "A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted group of dwarves to reclaim their mountain home- and the gold within it -from the dragon smaug"

}

{

"_id" : ObjectId("6105493820ea11fa5a5dd067"),

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

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

"year" : 2013,

"franchise" : "The Hobbit",

"synopsis" : "The dwarves along with biblobaggins and gandalf the grey, continue their quest to reclaim Erebor their homesland from SmaugBiblo Baggins is in possession of a mysterious and magical ring"

}

{

"_id" : ObjectId("610549df20ea11fa5a5dd068"),

"title" : "The Hobbit:The Battle of five Armies",

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "Biblo 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("61054a1320ea11fa5a5dd069"),

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

}

{ "_id" : ObjectId("61054a1f20ea11fa5a5dd06a"), "title" : "Avatar" }

(i-search)`:

INDRAJEET ROTE – MongoDB Assignment

```
db.movies.find({$regex:"Biblo"}).pretty()

  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted group of dwarves to reclaim their mountain home- and the gold within it -from the dragon smaug"

  "_id" : ObjectId("6105493820ea11fa5a5dd067"),
  "title" : "The Hobbit:The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit",
  "synopsis" : "The dwarves along with biblo baggins and gandalf the grey, continue their quest to reclaim Erebor their homesland from SmaugBiblo Baggins is in possession of a mysterious and magical ring"

  "_id" : ObjectId("610549df20ea11fa5a5dd068"),
  "title" : "The Hobbit:The Battle of five Armies",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "Biblo 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.find({$regex:"gandalf"}).pretty()

  "_id" : ObjectId("6105493820ea11fa5a5dd067"),
  "title" : "The Hobbit:The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit",
  "synopsis" : "The dwarves along with biblo baggins and gandalf the grey, continue their quest to reclaim Erebor their homesland from SmaugBiblo Baggins is in possession of a mysterious and magical ring"
```

```
db.movies.find({$and:[{$synopsis:{$regex:"Biblo"}},{synopsis:{$not:/gandalf/}}]}).pretty()

  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted group of dwarves to reclaim their mountain home- and the gold within it -from the dragon smaug"

  "_id" : ObjectId("610549df20ea11fa5a5dd068"),
  "title" : "The Hobbit:The Battle of five Armies",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "Biblo 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.find({$or:[{$synopsis:{$regex:"dwarves"}},{synopsis:{$regex:"hobbit"}}]}).pretty()

  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted group of dwarves to reclaim their mountain home- and the gold within it -from the dragon smaug"

  "_id" : ObjectId("6105493820ea11fa5a5dd067"),
  "title" : "The Hobbit:The Desolation of Smaug",
  "writer" : "J.R.R. Tolkein",
  "year" : 2013,
  "franchise" : "The Hobbit",
  "synopsis" : "The dwarves along with biblo baggins and gandalf the grey, continue their quest to reclaim Erebor their homesland from SmaugBiblo Baggins is in possession of a mysterious and magical ring"

> db.movies.find({$and:[{$synopsis:{$regex:"gold"}},{synopsis:{$regex:"dragon"}}]}).pretty()
{
  "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
  "title" : "The Hobbit:An unexpected Journey",
  "writer" : "J.R.R. Tolkein",
  "year" : 2012,
  "franchise" : "The Hobbit",
  "synopsis" : "A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted group of dwarves to reclaim their mountain home- and the gold within it -from the dragon smaug"
}
```

```
db.movies.remove({title:"Pee Wee Herman's Big Adventure"})
```

```
WriteResult({ "nRemoved" : 1 })
```

```
>db.movies.find().pretty()
```

```
{
  "_id" :ObjectId("6104fa96e9faf8b7ac9daae4"),
  "title" : "fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
```

```

        "Edward Norton"
    ]
}
{
    "_id" : ObjectId("6104fb13e9faf8b7ac9daae5"),
    "title" : "pulp fiction",
    "writer" : "Quentin Tarantino",
    "year" : 1994,
    "actors" : [
        "Samuel L Jackson"
    ]
}
{
    "_id" : ObjectId("6104fb7ae9faf8b7ac9daae6"),
    "title" : "inglorious basterds",
    "writer" : "Quentin Tarantino",
    "year" : 2009,
    "actors" : [
        "Brad Pitt",
        "Diane Kruger",
        "Eli Roth"
    ]
}
{
    "_id" : ObjectId("6104fc63e9faf8b7ac9daae7"),
    "title" : "The Hobbit:An unexpected Journey",
    "writer" : "J.R.R. Tolkein",
    "year" : 2012,
    "franchise" : "The Hobbit",
    "synopsis" : "A reluctant hobbit Biblo Baggins sets out to the lonely Mountain with a sprinted group of dwarves to reclaim their mountain home- and the gold within it -from the dragon smaug"
}
{
    "_id" : ObjectId("6105493820ea11fa5a5dd067"),
    "title" : "The Hobbit:The Desolation of Smaug",
    "writer" : "J.R.R. Tolkein",
    "year" : 2013,
    "franchise" : "The Hobbit",

```

"synopsis" : "The dwarves along with biblobaggins and gandalf the grey, continue their quest to reclaim Erebor their homeland from SmaugBiblo Baggins is in possession of a mysterious and magical ring"

}

{

"_id" : ObjectId("610549df20ea11fa5a5dd068"),

"title" : "The Hobbit:The Battle of five Armies",

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

"year" : 2012,

"franchise" : "The Hobbit",

"synopsis" : "Biblo 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("61054a1f20ea11fa5a5dd06a"), "title" : "Avatar" }

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

WriteResult({ "nRemoved" : 1 })

>db.movies.find().pretty()

```
> db.users.insert( {username: "GoodGuyGreg", first_name:"Good Guy", last_name:"Greg"})
WriteResult({ "nInserted" : 1 })
> db.users.insert( {username: "ScumbagSteve",full_name:{ first:"Scumbag", last:"Steve"}})
WriteResult({ "nInserted" : 1 })
> db.users.find().pretty()
```

```
db.users.find().pretty()

  "_id" : ObjectId("6105a672cb8f46e2f3f99f03"),
  "username" : "GoodGuyGreg",
  "first_name" : "Good Guy",
  "last_name" : "Greg"

  "_id" : ObjectId("6105a6d2cb8f46e2f3f99f04"),
  "username" : "ScumbagSteve",
  "full_name" : {
    "first" : "Scumbag",
    "last" : "Steve"
  }
}
```

```

> 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: "Sendsd 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.posts.find().pretty()
{
  "_id" : ObjectId("6105a7a2cb8f46e2f3f99f05"),
  "username" : "GoodGuyGreg",
  "title" : "Passes out at party",
  "body" : "Wakes up early and cleans house"
}
{
  "_id" : ObjectId("6105a7d1cb8f46e2f3f99f06"),
  "username" : "GoodGuyGreg",
  "title" : "Steals your identity",
  "body" : "Raises your credit score"
}
{
  "_id" : ObjectId("6105a7facb8f46e2f3f99f07"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sendsd you a Pull Request"
}
{
  "_id" : ObjectId("6105a824cb8f46e2f3f99f08"),
  "username" : "ScumbagSteve",
  "title" : "Borrows something",
  "body" : "Sells it"
}
{
  "_id" : ObjectId("6105a86acb8f46e2f3f99f09"),
  "username" : "ScumbagSteve",
  "title" : "Borrows everything",
  "body" : "The end"
}
{
  "_id" : ObjectId("6105a88dcb8f46e2f3f99f0a"),
  "username" : "ScumbagSteve",
  "title" : "Forks your repo on github",
  "body" : "Sets to private"
}
> db.comments.find({post:db.posts.findOne({title:"Reports a bug in your code"})})
> db.posts.aggregate([
... $match:{title:"Reports a bug in your code"},
... {$lookup:{from:'comments', localField:'_id', foreignField:'post', as:'comments'}}]).pretty()
{
  "_id" : ObjectId("6105a7facb8f46e2f3f99f07"),
  "username" : "GoodGuyGreg",
  "title" : "Reports a bug in your code",
  "body" : "Sendsd you a Pull Request",
  "comments" : [
    {
      "_id" : ObjectId("6105b050cb8f46e2f3f99f14"),
      "username" : "ScumbagSteve",
      "comment" : "Denied your PR cause I found a hack",
      "post" : ObjectId("6105a7facb8f46e2f3f99f07")
    }
  ]
}
>

```

```

db.comments.insert( {username: "GoodGuyGreg", comment: "Hope you get a good deal!", post:ObjectId("6105a824cb8f46e2f3f99f08")} )
iteResult({ "inserted" : 1 })
db.comments.insert( {username: "GoodGuyGreg", comment:"whats mine is yours", post:ObjectId("6105a64acb8f46e2f3f99f09")} )
iteResult({ "inserted" : 1 })
db.comments.insert( {username: "GoodGuyGreg", comment:"dont violate the license agreement!", post:ObjectId("6105a88dcb8f46e2f3f99f0a")} )
iteResult({ "inserted" : 1 })
db.comments.insert( {username: "ScumbagSteve", comment:"It still isnt clear", post:ObjectId("6105a7a2cb8f46e2f3f99f05")} )
iteResult({ "inserted" : 1 })
db.comments.insert( {username: "ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("6105a7facb8f46e2f3f99f07")} )
iteResult({ "inserted" : 1 })

> db.comments.find().pretty()
{
  "_id" : ObjectId("6105afd1cb8f46e2f3f99f10"),
  "username" : "GoodGuyGreg",
  "comment" : "Hope you get a good deal!",
  "post" : ObjectId("6105a824cb8f46e2f3f99f08")
}
{
  "_id" : ObjectId("6105affdcb8f46e2f3f99f11"),
  "username" : "GoodGuyGreg",
  "comment" : "whats mine is yours",
  "post" : ObjectId("6105a64acb8f46e2f3f99f09")
}
{
  "_id" : ObjectId("6105b018cb8f46e2f3f99f12"),
  "username" : "GoodGuyGreg",
  "comment" : "dont violate the license agreement!",
  "post" : ObjectId("6105a88dcb8f46e2f3f99f0a")
}
{
  "_id" : ObjectId("6105b037cb8f46e2f3f99f13"),
  "username" : "ScumbagSteve",
  "comment" : "It still isnt clear",
  "post" : ObjectId("6105a7a2cb8f46e2f3f99f05")
}
{
  "_id" : ObjectId("6105b050cb8f46e2f3f99f14"),
  "username" : "ScumbagSteve",
  "comment" : "Denied your PR cause I found a hack",
  "post" : ObjectId("6105a7facb8f46e2f3f99f07")
}

db.posts.find((username:"GoodGuyGreg"))
{ "_id" : ObjectId("6105a7a2cb8f46e2f3f99f05"), "username" : "GoodGuyGreg", "title" : "Passes out at party", "body" : "Wakes up early and cleans house" }
{ "_id" : ObjectId("6105a7d1cb8f46e2f3f99f06"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }
{ "_id" : ObjectId("6105a7facb8f46e2f3f99f07"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sendsd you a Pull Request" }
db.posts.find((username:"ScumbagSteve"))
{ "_id" : ObjectId("6105a802cb8f46e2f3f99f08"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }
{ "_id" : ObjectId("6105a86acb8f46e2f3f99f09"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }
{ "_id" : ObjectId("6105a88dcb8f46e2f3f99f0a"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }
db.collections.find((username:"GoodGuyGreg"))
{ "_id" : ObjectId("6105afd1cb8f46e2f3f99f10"), "username" : "GoodGuyGreg", "comment" : "Hope you get a good deal!", "post" : ObjectId("6105a824cb8f46e2f3f99f08") }
{ "_id" : ObjectId("6105affdcb8f46e2f3f99f11"), "username" : "GoodGuyGreg", "comment" : "whats mine is yours", "post" : ObjectId("6105a64acb8f46e2f3f99f09") }
{ "_id" : ObjectId("6105b018cb8f46e2f3f99f12"), "username" : "GoodGuyGreg", "comment" : "dont violate the license agreement!", "post" : ObjectId("6105a88dcb8f46e2f3f99f0a") }
db.collections.find((username:"ScumbagSteve"))
{ "_id" : ObjectId("6105b037cb8f46e2f3f99f13"), "username" : "ScumbagSteve", "comment" : "It still isnt clear", "post" : ObjectId("6105a7a2cb8f46e2f3f99f05") }
{ "_id" : ObjectId("6105b050cb8f46e2f3f99f14"), "username" : "ScumbagSteve", "comment" : "Denied your PR cause I found a hack", "post" : ObjectId("6105a7facb8f46e2f3f99f07") }

```

```

db.posts.aggregate([
{
  $match: { title: 'Reports a bug in your code' }
},
{
  $lookup: { from:
'comments',
localField: '_id',
foreignField: 'post',
as: 'comments'
}
}
]).pretty();

```


EXERCISE 2

ATLANTA POPULATION

```
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER\Downloads\mongodb-database-tools-windows-x86_64-100.4.1\mongodb-database-tools-windows-x86_64-100.4.1\bin>mongoimport --db population --collection zipcodes --file C:\Users\USER\Assignments\zip.json
2021-08-01T12:11:19.152+0530   connected to: mongodb://localhost/
2021-08-01T12:11:19.635+0530   29353 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\USER\Downloads\mongodb-database-tools-windows-x86_64-100.4.1\mongodb-database-tools-windows-x86_64-100.4.1\bin>
```

1.

```
> db.zipcodes.find({$and:[{city:"ATLANTA"},{state:"GA"}]}).pretty()
> show dbs
admin                0.000GB
config               0.000GB
local                0.000GB
mongo_practice       0.000GB
population           0.002GB
> use population
switched to db population
> db.zipcodes.find({$and:[{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"
}
{
  "_id" : "30307",
  "city" : "ATLANTA",
  "loc" : [
    -84.335957,
    33.769138
  ],

```

2.

```
> db.zipcodes.aggregate([{$match:{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" : "30317", "city" : "ATLANTA", "loc" : [ -84.31685, 33.749788 ], "pop" : 16395, "state" : "GA" }
{ "_id" : "30316", "city" : "ATLANTA", "loc" : [ -84.333913, 33.721686 ], "pop" : 34668, "state" : "GA" }
{ "_id" : "30318", "city" : "ATLANTA", "loc" : [ -84.445432, 33.786454 ], "pop" : 53894, "state" : "GA" }
{ "_id" : "30324", "city" : "ATLANTA", "loc" : [ -84.354867, 33.820609 ], "pop" : 15044, "state" : "GA" }
{ "_id" : "30319", "city" : "ATLANTA", "loc" : [ -84.335091, 33.868728 ], "pop" : 32138, "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" : "30330", "city" : "ATLANTA", "loc" : [ -84.434735, 33.70645 ], "pop" : 643, "state" : "GA" }
Type "it" for more
```

3.

```
> db.students.aggregate([{$match:{city:"ATLANTA"}},{$count:"Total number of zipcodes"}])
> db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$count:"Total number of zipcodes"}])
{ "Total number of zipcodes" : 41 }
```

4.

```
db.zipcodes.aggregate([{$sort:{pop:-1}}])
{ "_id" : "60623", "city" : "CHICAGO", "loc" : [ -87.7157, 41.849015 ], "pop" : 112047, "state" : "IL" }
{ "_id" : "11226", "city" : "BROOKLYN", "loc" : [ -73.956985, 40.646694 ], "pop" : 111396, "state" : "NY" }
{ "_id" : "10021", "city" : "NEW YORK", "loc" : [ -73.958805, 40.768476 ], "pop" : 106564, "state" : "NY" }
{ "_id" : "10025", "city" : "NEW YORK", "loc" : [ -73.968312, 40.797466 ], "pop" : 100027, "state" : "NY" }
{ "_id" : "90201", "city" : "BELL GARDENS", "loc" : [ -118.17205, 33.969177 ], "pop" : 99568, "state" : "CA" }
{ "_id" : "60617", "city" : "CHICAGO", "loc" : [ -87.556012, 41.725743 ], "pop" : 98612, "state" : "IL" }
{ "_id" : "90011", "city" : "LOS ANGELES", "loc" : [ -118.258189, 34.007856 ], "pop" : 96074, "state" : "CA" }
{ "_id" : "60647", "city" : "CHICAGO", "loc" : [ -87.704322, 41.920903 ], "pop" : 95971, "state" : "IL" }
{ "_id" : "60628", "city" : "CHICAGO", "loc" : [ -87.624277, 41.693443 ], "pop" : 94317, "state" : "IL" }
{ "_id" : "90650", "city" : "NORWALK", "loc" : [ -118.081767, 33.90564 ], "pop" : 94188, "state" : "CA" }
{ "_id" : "60620", "city" : "CHICAGO", "loc" : [ -87.654251, 41.741119 ], "pop" : 92005, "state" : "IL" }
{ "_id" : "60629", "city" : "CHICAGO", "loc" : [ -87.706936, 41.778149 ], "pop" : 91814, "state" : "IL" }
{ "_id" : "60609", "city" : "CHICAGO", "loc" : [ -87.653279, 41.809721 ], "pop" : 89762, "state" : "IL" }
{ "_id" : "60618", "city" : "CHICAGO", "loc" : [ -87.704214, 41.946401 ], "pop" : 88377, "state" : "IL" }
{ "_id" : "11373", "city" : "JACKSON HEIGHTS", "loc" : [ -73.878551, 40.740388 ], "pop" : 88241, "state" : "NY" }
{ "_id" : "91331", "city" : "ARLETA", "loc" : [ -118.420692, 34.258081 ], "pop" : 88114, "state" : "CA" }
{ "_id" : "11212", "city" : "BROOKLYN", "loc" : [ -73.914483, 40.662474 ], "pop" : 87079, "state" : "NY" }
{ "_id" : "90280", "city" : "SOUTH GATE", "loc" : [ -118.201349, 33.94617 ], "pop" : 87026, "state" : "CA" }
{ "_id" : "11385", "city" : "RIDGEWOOD", "loc" : [ -73.896122, 40.703613 ], "pop" : 85732, "state" : "NY" }
{ "_id" : "10467", "city" : "BRONX", "loc" : [ -73.871242, 40.873671 ], "pop" : 85710, "state" : "NY" }
Type "it" for more
```

POPULATION BY STATE

2.


```

db.zipcodes.aggregate([{$group:{_id:"$state"}},{ $sort:{pop:-1}}])
  "_id" : "NV" }
  "_id" : "TX" }
  "_id" : "WY" }
  "_id" : "AZ" }
  "_id" : "CO" }
  "_id" : "MO" }
  "_id" : "ME" }
  "_id" : "CA" }
  "_id" : "MA" }
  "_id" : "KY" }
  "_id" : "NJ" }
  "_id" : "MI" }
  "_id" : "ID" }
  "_id" : "DC" }
  "_id" : "CT" }
  "_id" : "NY" }
  "_id" : "MT" }
  "_id" : "UT" }
  "_id" : "NE" }
  "_id" : "GA" }
type "it" for more

```

```

1. db.zipcodes.aggregate({$match:{city:"ATLANTA"}},{ $group:{_id:'$city',zips:{ $sum:1}}}) {
  "_id" : "ATLANTA", "zips" : 41 }

db.zipcodes.aggregate({$match:{city:"ATLANTA"}},{ $group:{_id:'$city',Population:{ $sum:'$pop'}}})
{ "_id" : "ATLANTA", "Population" : 630046 } db.zipcodes.aggregate({$group:{_id:'$state',Population:{ $sum:'$pop'}}})
{ "_id" : "NV", "Population" : 1201833 }
{ "_id" : "TX", "Population" : 16984601 }
{ "_id" : "WY", "Population" : 453528 }
{ "_id" : "AZ", "Population" : 3665228 }
{ "_id" : "CO", "Population" : 3293755 }
{ "_id" : "MO", "Population" : 5110648 }
{ "_id" : "ME", "Population" : 1226648 }
{ "_id" : "CA", "Population" : 29754890 }
{ "_id" : "MA", "Population" : 6016425 }
{ "_id" : "KY", "Population" : 3675484 }
{ "_id" : "NJ", "Population" : 7730188 }
{ "_id" : "MI", "Population" : 9295297 }
{ "_id" : "ID", "Population" : 1006749 }
{ "_id" : "DC", "Population" : 606900 }
{ "_id" : "CT", "Population" : 3287116 }
{ "_id" : "NY", "Population" : 17990402 }

```

```
{ "_id" : "MT", "Population" : 798948 }
{ "_id" : "UT", "Population" : 1722850 }
{ "_id" : "NE", "Population" : 1578139 }
{ "_id" : "GA", "Population" : 6478216 }
```

Type "it" for more

```
3. db.zipcodes.aggregate({$group:{_id:{state:"$state"},population:{$sum:'$pop'}}},{sort:{population:-1}},{limit:3})
```

```
{ "_id" : { "state" : "CA" }, "population" : 29754890 }
{ "_id" : { "state" : "NY" }, "population" : 17990402 }
{ "_id" : { "state" : "TX" }, "population" : 16984601 }
```

POPULATION BY CITY

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

```
{ "_id" : { "city" : "SAINT PETERS", "state" : "MO" }, "population" : 43133 }
{ "_id" : { "city" : "PRICHARD", "state" : "WV" }, "population" : 1215 }
{ "_id" : { "city" : "RANCHITA", "state" : "CA" }, "population" : 389 }
{ "_id" : { "city" : "DARDANELLE", "state" : "AR" }, "population" : 8281 }
{ "_id" : { "city" : "TRIDELL", "state" : "UT" }, "population" : 363 }
{ "_id" : { "city" : "TURTLE CREEK", "state" : "PA" }, "population" : 8157 }
{ "_id" : { "city" : "HOUSTON", "state" : "PA" }, "population" : 6060 }
{ "_id" : { "city" : "MALONE", "state" : "TX" }, "population" : 491 }
{ "_id" : { "city" : "ROUND MOUNTAIN", "state" : "NV" }, "population" : 1302 }
{ "_id" : { "city" : "ROSSFORD", "state" : "OH" }, "population" : 5861 }
{ "_id" : { "city" : "SAN MARTIN", "state" : "CA" }, "population" : 5563 }
{ "_id" : { "city" : "FAIRBANKS", "state" : "AK" }, "population" : 31379 }
{ "_id" : { "city" : "SOUTH SHORE", "state" : "SD" }, "population" : 487 }
{ "_id" : { "city" : "HOSSTON", "state" : "LA" }, "population" : 766 }
{ "_id" : { "city" : "PEMBERTON", "state" : "MN" }, "population" : 880 }
{ "_id" : { "city" : "HOPE", "state" : "RI" }, "population" : 3653 }
{ "_id" : { "city" : "COATS", "state" : "NC" }, "population" : 4437 }
{ "_id" : { "city" : "PECK", "state" : "KS" }, "population" : 1023 }
{ "_id" : { "city" : "FORT BENTON", "state" : "MT" }, "population" : 2795 }
{ "_id" : { "city" : "LAWRENCE", "state" : "NE" }, "population" : 594 }
```

Type "it" for more

```
2. db.zipcodes.aggregate({$group:{_id:{city:"$city",state:"$state"},population:{$sum:'$pop'}}},{sort:{population:-1}})
```

```
{ "_id" : { "city" : "CHICAGO", "state" : "IL" }, "population" : 2452177 }
{ "_id" : { "city" : "BROOKLYN", "state" : "NY" }, "population" : 2300504 }
{ "_id" : { "city" : "LOS ANGELES", "state" : "CA" }, "population" : 2102295 }
```

```
{ "_id" : { "city" : "HOUSTON", "state" : "TX" }, "population" : 2095918 }
{ "_id" : { "city" : "PHILADELPHIA", "state" : "PA" }, "population" : 1610956 }
{ "_id" : { "city" : "NEW YORK", "state" : "NY" }, "population" : 1476790 }
{ "_id" : { "city" : "BRONX", "state" : "NY" }, "population" : 1209548 }
{ "_id" : { "city" : "SAN DIEGO", "state" : "CA" }, "population" : 1049298 }
{ "_id" : { "city" : "DETROIT", "state" : "MI" }, "population" : 963243 }
{ "_id" : { "city" : "DALLAS", "state" : "TX" }, "population" : 940191 }
{ "_id" : { "city" : "PHOENIX", "state" : "AZ" }, "population" : 890853 }
{ "_id" : { "city" : "MIAMI", "state" : "FL" }, "population" : 825232 }
{ "_id" : { "city" : "SAN JOSE", "state" : "CA" }, "population" : 816653 }
{ "_id" : { "city" : "SAN ANTONIO", "state" : "TX" }, "population" : 811792 }
{ "_id" : { "city" : "BALTIMORE", "state" : "MD" }, "population" : 733081 }
{ "_id" : { "city" : "SAN FRANCISCO", "state" : "CA" }, "population" : 723993 }
{ "_id" : { "city" : "MEMPHIS", "state" : "TN" }, "population" : 632837 }
{ "_id" : { "city" : "SACRAMENTO", "state" : "CA" }, "population" : 628279 }
{ "_id" : { "city" : "JACKSONVILLE", "state" : "FL" }, "population" : 610160 }
{ "_id" : { "city" : "ATLANTA", "state" : "GA" }, "population" : 609591 }
```

Type "it" for more

```
3. db.zipcodes.aggregate({$group:{_id:{city:"$city"},population:{$sum:'$pop'}}},{sort:{population:-1}},{$limit:3})
```

```
{ "_id" : { "city" : "CHICAGO" }, "population" : 2452177 }
{ "_id" : { "city" : "BROOKLYN" }, "population" : 2341387 }
{ "_id" : { "city" : "HOUSTON" }, "population" : 2123053 }
```

```
4. db.zipcodes.aggregate({$match:{state:"TX"}},{ $group:{_id:'$city',Population:{ $sum:'$pop' } }},{ $sort:{Population:-1}},{$limit:3})
```

```
{ "_id" : "HOUSTON", "Population" : 2095918 }
{ "_id" : "DALLAS", "Population" : 940191 } { "_id" : "SAN ANTONIO", "Population" : 811792 }

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

{ "_id" : "NV", "avgCityPop" : 18209.590909090908 }
{ "_id" : "TX", "avgCityPop" : 13775.02108678021 }
{ "_id" : "WY", "avgCityPop" : 3384.5373134328356 }
{ "_id" : "AZ", "avgCityPop" : 20591.16853932584 }
{ "_id" : "CO", "avgCityPop" : 9981.075757575758 }
{ "_id" : "MO", "avgCityPop" : 5672.195338512764 }
{ "_id" : "ME", "avgCityPop" : 3006.4901960784314 }
```

```
{ "_id" : "CA", "avgCityPop" : 27756.42723880597 }
{ "_id" : "MA", "avgCityPop" : 14855.37037037037 }
{ "_id" : "KY", "avgCityPop" : 4767.164721141375 }
{ "_id" : "NJ", "avgCityPop" : 15775.89387755102 }
{ "_id" : "MI", "avgCityPop" : 12087.512353706112 }
{ "_id" : "ID", "avgCityPop" : 4320.811158798283 }
{ "_id" : "DC", "avgCityPop" : 303450 }
{ "_id" : "CT", "avgCityPop" : 14674.625 }
{ "_id" : "MT", "avgCityPop" : 2593.987012987013 }
{ "_id" : "NY", "avgCityPop" : 13131.680291970803 }
{ "_id" : "UT", "avgCityPop" : 9518.508287292818 }
{ "_id" : "NE", "avgCityPop" : 3034.882692307692 }
{ "_id" : "GA", "avgCityPop" : 11547.62210338681 }
```

Type "it" for more **BONUS**

1

```
db.zipcodes.aggregate([{$sort:{avgCityPop:-1}},{$limit:3}])
{ "_id" : "01026", "city" : "CUMMINGTON", "loc" : [ -72.905767, 42.435296 ], "pop" : 1484, "state" : "MA" }
{ "_id" : "01013", "city" : "CHICOPEE", "loc" : [ -72.607962, 42.162046 ], "pop" : 23396, "state" : "MA" }
{ "_id" : "01011", "city" : "CHESTER", "loc" : [ -72.988761, 42.279421 ], "pop" : 1688, "state" : "MA" } BONUS
```

2

```
db.zipcodes.aggregate( [{ $group: { _id: { state: "$state", city: "$city" }, pop: { $sum: "$pop" } } }, { $group: { _id: "$_id.state", avgCityPop: { $avg: "$pop" } } }, {$sort:{avgCityPop:-1}},{$limit:3} ] )
{ "_id" : "DC", "avgCityPop" : 303450 }
{ "_id" : "CA", "avgCityPop" : 27756.42723880597 }
{ "_id" : "FL", "avgCityPop" : 27400.958963282937 }
```

EXERCISE 3

MONGODB COMPLEX QUERIES

C:\Users\USER\Downloads\mongodb-database-tools-windows-x86_64-100.4.1\mongodb-database-tools-windows-x86_64-100.4.1\bin>mongoimport --db restaurants --collection addresses --file C:\Users\USER\Assignments\restaurants.json

2021-08-02T01:08:39.731+0530 connected to: mongodb://localhost/

2021-08-02T01:08:39.834+0530 3772 document(s) imported successfully. 0 document(s) failed to import.

EXERCISE QNS 1.

```
use restaurants
switched to db restaurants
db.addresses.find().pretty()

  "_id" : ObjectId("6106f99585e29051a03be21d"),
  "address" : {
    "building" : "97-22",
    "coord" : [
      -73.8601152,
      40.7311739
    ],
    "street" : "63 Road",
    "zipcode" : "11374"
  },
  "borough" : "Queens",
  "cuisine" : "Jewish/Kosher",
  "grades" : [
    {
      "date" : ISODate("2014-11-24T00:00:00Z"),
      "grade" : "Z",
      "score" : 20
    },
    {
      "date" : ISODate("2013-01-17T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2012-08-02T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2011-12-15T00:00:00Z"),
      "grade" : "B",
      "score" : 25
    }
  ],
  "name" : "Tov Kosher Kitchen",
  "restaurant_id" : "40356068"

  "_id" : ObjectId("6106f99585e29051a03be21e"),
  "address" : {
```

2.

```
b.restaurant.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1,"_id":0});
b.addresses.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1,"_id":0});
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" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
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" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant_id" : "40361521" }
borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant_id" : "40361998" }
borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery", "restaurant_id" : "40362264" }
borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant_id" : "40362098" }
e "it" for more
```

3.

INDRAJEET ROTE – MongoDB Assignment

```
db.addresses.find({}, "restaurant_id", "1", "bch", "1", "borough", "1", "cuisine", "1");
```

```
{
  "_id": "ObjectID('6106f99585e29051a03be21d)", "borough": "Queens", "cuisine": "Jewish/Kosher", "name": "Tov Kosher Kitchen", "restaurant_id": "40356068" }
{
  "_id": "ObjectID('6106f99585e29051a03be21d)", "borough": "Staten Island", "cuisine": "Jewish/Kosher", "name": "Kosher Island", "restaurant_id": "40356442" }
{
  "_id": "ObjectID('6106f99585e29051a03be21f)", "borough": "Bronx", "cuisine": "Bakery", "name": "Morris Park Bake Shop", "restaurant_id": "30075445" }
{
  "_id": "ObjectID('6106f99585e29051a03be220)", "borough": "Brooklyn", "cuisine": "Chinese", "name": "May May Kitchen", "restaurant_id": "40358429" }
{
  "_id": "ObjectID('6106f99585e29051a03be221)", "borough": "Manhattan", "cuisine": "American", "name": "1 East 60th Street Kitchen", "restaurant_id": "40359480" }
{
  "_id": "ObjectID('6106f99585e29051a03be222)", "borough": "Brooklyn", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "40356649" }
{
  "_id": "ObjectID('6106f99585e29051a03be223)", "borough": "Brooklyn", "cuisine": "American", "name": "Riviera Caterer", "restaurant_id": "40356018" }
{
  "_id": "ObjectID('6106f99585e29051a03be224)", "borough": "Bronx", "cuisine": "American", "name": "Mid Asia", "restaurant_id": "4035717" }
{
  "_id": "ObjectID('6106f99585e29051a03be225)", "borough": "Brooklyn", "cuisine": "Ice Cream", "name": "Carvel Ice Cream", "restaurant_id": "40360076" }
{
  "_id": "ObjectID('6106f99585e29051a03be226)", "borough": "Queens", "cuisine": "Ice Cream", "name": "Carvel Ice Cream", "restaurant_id": "40361322" }
{
  "_id": "ObjectID('6106f99585e29051a03be227)", "borough": "Brooklyn", "cuisine": "Deli/Caterers", "name": "Nordic Delicacies", "restaurant_id": "40361390" }
{
  "_id": "ObjectID('6106f99585e29051a03be228)", "borough": "Brooklyn", "cuisine": "Jewish/Kosher", "name": "Sunda Foods", "restaurant_id": "40360045" }
{
  "_id": "ObjectID('6106f99585e29051a03be229)", "borough": "Brooklyn", "cuisine": "American", "name": "The Movable Feast", "restaurant_id": "40361606" }
{
  "_id": "ObjectID('6106f99585e29051a03be22a)", "borough": "Queens", "cuisine": "Deli/Caterers", "name": "Sal's Deli", "restaurant_id": "40361618" }
{
  "_id": "ObjectID('6106f99585e29051a03be22b)", "borough": "Manhattan", "cuisine": "Deli/Caterers", "name": "Bully's Deli", "restaurant_id": "40361708" }
{
  "_id": "ObjectID('6106f99585e29051a03be22c)", "borough": "Manhattan", "cuisine": "American", "name": "Glorious Food", "restaurant_id": "40361521" }
{
  "_id": "ObjectID('6106f99585e29051a03be22d)", "borough": "Queens", "cuisine": "Deli/Caterers", "name": "Steve Chu's Deli & Grocery", "restaurant_id": "40361998" }
{
  "_id": "ObjectID('6106f99585e29051a03be22e)", "borough": "Brooklyn", "cuisine": "American", "name": "C & C Catering Service", "restaurant_id": "40357437" }
{
  "_id": "ObjectID('6106f99585e29051a03be22f)", "borough": "Manhattan", "cuisine": "American", "name": "P & S Deli Grocery", "restaurant_id": "40362264" }
{
  "_id": "ObjectID('6106f99585e29051a03be230)", "borough": "Manhattan", "cuisine": "Chicken", "name": "Harriet's Kitchen", "restaurant_id": "40362098" }
```

4.

```

Type "it" for more
> db.addresses.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" : 1,"_id":0, "address.zipcode": 1}).pretty()
{
  "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",
  "restaurant_id" : "40356442"
}
{
  "address" : {
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "name" : "Morris Park Bake Shop",
  "restaurant_id" : "30075445"
}
{
  "address" : {
    "zipcode" : "11208"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Chinese",
  "name" : "May May Kitchen",
  "restaurant_id" : "40358429"
}
{
  "address" : {
    "zipcode" : "10065"
  },
  "borough" : "Manhattan",
  "cuisine" : "Italian",
  "name" : "Manhattan Italian",
  "restaurant_id" : "40358429"
}

```

6.

```
db.addresses.find({"borough": "Bronx"}).pretty()
{
  "_id" : ObjectId("6106f99585e29051a03be21f"),
  "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("6106f99585e29051a03be224"),
  "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" : "40357217"
}

```

```

7.
db.addresses.find({borough: "Bronx"}).limit(5).pretty()

{
  "_id" : ObjectId("6106f99585e29051a03be21f"),
  "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
    }
  ],
  "score" : 7.0
}

7.
db.addresses.find({borough: "Bronx"}).skip(5).limit(5).pretty()

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

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

{ "_id" : ObjectId("6106f99585e29051a03be41e"), "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-
0530T00:00:00Z"), "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("6106f99585e29051a03be57d"), "address": { "building": "130", "coord": [ -73.984758, 40.7457939 ], "street":
"Madison Avenue", "zipcode": "10016" }, "borough": "Manhattan", "cuisine": "Pizza/Italian", "grades": [ { "date": ISODate("2014-
1224T00:00:00Z"), "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("6106f99685e29051a03bede8"), "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-
0619T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "West 79Th Street Boat Basin Cafe", "restaurant_id": "40756344" }
```

10.

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

```
{ "_id": ObjectId("6106f99685e29051a03be863"), "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-
0517T00:00:00Z"), "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" }
```

11. db.addresses.find({\$and:

```
[
{"cuisine":{$ne:"American"}},
{"grades.score":{$gt:70}},
{"address.coord":{$lt:-65.754168}}
] }
);
```

12. db.addresses.find(

```
{
"cuisine":{$ne:"American"},
"grades.score":{$gt:70},
"address.coord":{$lt:-65.754168}
}
);
```

13. db.addresses.find({\$and:[{"cuisine": {\$ne:"American"}}, {"grades.grade": "A"}, {"borough": {\$ne
:"Brooklyn"}}]}).sort({"cuisine":-1}).pretty()

13.

```

type "it" for more
> db.addresses.find({$and:[{"cuisine" : {$ne : "American "}}, {"grades.grade" : "A"}, {"borough" : {$ne : "Brooklyn "}}]}).sort({"cuisine":-1}).pretty()
{
  "_id" : ObjectId("6106f99685e29051a03be934"),
  "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"
}

```

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

```
{ "_id" : ObjectId("6106f99585e29051a03be224"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be23e"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
```

```
{ "_id" : ObjectId("6106f99685e29051a03bf02e"), "borough" : "Bronx", "cuisine" : "Pizza", "name" : "Wilbel Pizza", "restaurant_id" : "40871979" }
```

15.

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

```
{ "_id" : ObjectId("6106f99685e29051a03be6ae"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Pieces", "restaurant_id" : "40399910" }
```

```
{ "_id" : ObjectId("6106f99685e29051a03be770"), "borough" : "Queens", "cuisine" : "American ", "name" : "S.M.R Restaurant Services", "restaurant_id" : "40403857" }
```

```
{ "_id" : ObjectId("6106f99685e29051a03be775"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Good Shepherd Services", "restaurant_id" : "40403989" }
```

```
{ "_id" : ObjectId("6106f99685e29051a03bec20"), "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "The Ice Box-Ralph'S Famous Italian Ices", "restaurant_id" : "40690899" }
```

```
{ "_id" : ObjectId("6106f99685e29051a03bee2b"), "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Alices", "restaurant_id" : "40782042" }
```

```
{ "_id" : ObjectId("6106f99685e29051a03bf046"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Re: Sources", "restaurant_id" : "40876068" }
```

16.

`db.addresses.find({name: /. *Reg* ./}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 })`

```
{ "_id" : ObjectId("6106f99585e29051a03be222"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
```

INDRAJEET ROTE – MongoDB Assignment

```
{ "_id" : ObjectId("6106f99585e29051a03be234"), "borough" : "Queens", "cuisine" : "Chinese", "name" : "Ho Mei Restaurant",
"restaurant_id" : "40362432" }

{ "_id" : ObjectId("6106f99585e29051a03be237"), "borough" : "Brooklyn", "cuisine" : "Caribbean", "name" : "Shashemene Int'L Restaura",
"restaurant_id" : "40362869" }

{ "_id" : ObjectId("6106f99585e29051a03be238"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And
Restaurant", "restaurant_id" : "30191841" }

{ "_id" : ObjectId("6106f99585e29051a03be259"), "borough" : "Manhattan", "cuisine" : "American", "name" : "Palm Restaurant",
"restaurant_id" : "40364355" }

{ "_id" : ObjectId("6106f99585e29051a03be25a"), "borough" : "Manhattan", "cuisine" : "American", "name" : "Old Town Bar &
Restaurant", "restaurant_id" : "40364389" }

{ "_id" : ObjectId("6106f99585e29051a03be25b"), "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Isle Of Capri Resturant",
"restaurant_id" : "40364373" }

{ "_id" : ObjectId("6106f99585e29051a03be260"), "borough" : "Manhattan", "cuisine" : "Latin (Cuban, Dominican, Puerto Rican, South &
Central American)", "name" : "Seville Restaurant", "restaurant_id" : "40364439" }

{ "_id" : ObjectId("6106f99585e29051a03be263"), "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Ben-Best Deli &
Restaurant", "restaurant_id" : "40364529" }

{ "_id" : ObjectId("6106f99585e29051a03be268"), "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Marchis Restaurant",
"restaurant_id" : "40364668" }

{ "_id" : ObjectId("6106f99585e29051a03be26f"), "borough" : "Queens", "cuisine" : "Pizza/Italian", "name" : "New Park Pizzeria &
Restaurant", "restaurant_id" : "40364744" }

{ "_id" : ObjectId("6106f99585e29051a03be278"), "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Forlinis Restaurant",
"restaurant_id" : "40365098" }

{ "_id" : ObjectId("6106f99585e29051a03be279"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dorrian'S Red Hand Restaurant",
"restaurant_id" : "40365239" }

{ "_id" : ObjectId("6106f99585e29051a03be27f"), "borough" : "Manhattan", "cuisine" : "American", "name" : "Keats Restaurant",
"restaurant_id" : "40365288" }

{ "_id" : ObjectId("6106f99585e29051a03be287"), "borough" : "Manhattan", "cuisine" : "Spanish", "name" : "Spain Restaurant & Bar",
"restaurant_id" : "40365472" }

{ "_id" : ObjectId("6106f99585e29051a03be288"), "borough" : "Manhattan", "cuisine" : "American", "name" : "Reynold'S Bar",
"restaurant_id" : "40365423" }

{ "_id" : ObjectId("6106f99585e29051a03be28d"), "borough" : "Manhattan", "cuisine" : "Italian", "name" : "V & T Restaurant",
"restaurant_id" : "40365577" }

{ "_id" : ObjectId("6106f99585e29051a03be28e"), "borough" : "Brooklyn", "cuisine" : "American", "name" : "Reben Luncheonette",
"restaurant_id" : "40365546" }

{ "_id" : ObjectId("6106f99585e29051a03be291"), "borough" : "Queens", "cuisine" : "Chinese", "name" : "King Yum Restaurant",
"restaurant_id" : "40365592" }

{ "_id" : ObjectId("6106f99585e29051a03be292"), "borough" : "Queens", "cuisine" : "American", "name" : "Jahn'S Restaurant",
"restaurant_id" : "40365627" }
```

17.

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

```
{ "_id" : ObjectId("6106f99585e29051a03be240"), "address" : { "building" : "1236", "coord" : [ -73.8893654, 40.81376179999999 ], "street" :
"238 Spofford Ave", "zipcode" : "10474" }, "borough" : "Bronx", "cuisine" : "Chinese", "grades" : [ { "date" : ISODate("2013-
1230T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-01-08T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" :
ISODate("2012-06-12T00:00:00Z"), "grade" : "B", "score" : 15 } ], "name" : "Happy Garden", "restaurant_id" : "40363289" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be252"), "address" : { "building" : "277", "coord" : [ -73.8941893, 40.8634684 ], "street" : "East
Kingsbridge Road", "zipcode" : "10458" }, "borough" : "Bronx", "cuisine" : "Chinese", "grades" : [ { "date" : ISODate("2014-
0303T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-09-26T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" :
```

INDRAJEET ROTE – MongoDB Assignment

```
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" }
```

```
{ "_id": ObjectId("6106f99685e29051a03be7cc"), "address": { "building": "2459", "coord": [ -73.8929629, 40.8601605 ], "street":  
"Webster Avenue", "zipcode": "10458" }, "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-  
1223T00:00:00Z"), "grade": "A", "score": 7 }, { "date": ISODate("2014-01-07T00:00:00Z"), "grade": "A", "score": 8 }, { "date":  
ISODate("2012-12-28T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2012-07-18T00:00:00Z"), "grade": "A", "score": 13 } ],  
"name": "China Wok II", "restaurant_id": "40510328" }
```

```
{ "_id": ObjectId("6106f99685e29051a03be854"), "address": { "building": "2102", "coord": [ -73.82609699999999, 40.8687177 ], "street":  
": "Bartow Ave", "zipcode": "10475" }, "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-0804T00:00:00Z"),  
"grade": "A", "score": 7 }, { "date": ISODate("2014-02-21T00:00:00Z"), "grade": "A", "score": 4 }, { "date":  
ISODate("2013-05-30T00:00:00Z"), "grade": "B", "score": 17 }, { "date": ISODate("2012-04-13T00:00:00Z"), "grade": "A", "score": 12 } ],  
"name": "Dragon City", "restaurant_id": "40529203" }
```

```
{ "_id": ObjectId("6106f99685e29051a03be902"), "address": { "building": "3511", "coord": [ -73.9100876, 40.8860444 ], "street":  
"Johnson Avenue", "zipcode": "10463" }, "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-  
0702T00:00:00Z"), "grade": "A", "score": 5 }, { "date": ISODate("2013-11-06T00:00:00Z"), "grade": "A", "score": 8 }, { "date":  
ISODate("2013-03-29T00:00:00Z"), "grade": "B", "score": 17 }, { "date": ISODate("2012-06-20T00:00:00Z"), "grade": "B", "score": 27 } ],  
"name": "Hunan Balcony", "restaurant_id": "40551996" }
```

```
{ "_id": ObjectId("6106f99685e29051a03be905"), "address": { "building": "732", "coord": [ -73.883128, 40.85353 ], "street": "East 187  
Street", "zipcode": "10458" }, "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-11-07T00:00:00Z"), "grade":  
": "A", "score": 9 }, { "date": ISODate("2014-05-15T00:00:00Z"), "grade": "A", "score": 7 }, { "date": ISODate("2013-11-18T00:00:00Z"),  
"grade": "A", "score": 7 }, { "date": ISODate("2013-06-10T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2012-  
0917T00:00:00Z"), "grade": "B", "score": 19 } ], "name": "Great Wall Restaurant", "restaurant_id": "40552226" }
```

18.

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

```
{ "_id": ObjectId("6106f99585e29051a03be21d"), "borough": "Queens", "cuisine": "Jewish/Kosher", "name": "Tov Kosher Kitchen",  
"restaurant_id": "40356068" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be21e"), "borough": "Staten Island", "cuisine": "Jewish/Kosher", "name": "Kosher Island",  
"restaurant_id": "40356442" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be21f"), "borough": "Bronx", "cuisine": "Bakery", "name": "Morris Park Bake Shop",  
"restaurant_id": "30075445" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be220"), "borough": "Brooklyn", "cuisine": "Chinese", "name": "May May Kitchen",  
"restaurant_id": "40358429" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be222"), "borough": "Brooklyn", "cuisine": "American ", "name": "Regina Caterers",  
"restaurant_id": "40356649" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be223"), "borough": "Brooklyn", "cuisine": "American ", "name": "Riviera Caterer",  
"restaurant_id": "40356018" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be224"), "borough": "Bronx", "cuisine": "American ", "name": "Wild Asia", "restaurant_id":  
"40357217" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be225"), "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel  
Ice Cream", "restaurant_id": "40360076" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be226"), "borough": "Queens", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice  
Cream", "restaurant_id": "40361322" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be227"), "borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Nordic Delicacies",  
"restaurant_id": "40361390" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be228"), "borough": "Brooklyn", "cuisine": "Jewish/Kosher", "name": "Seuda Foods",  
"restaurant_id": "40360045" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be229"), "borough": "Brooklyn", "cuisine": "American ", "name": "The Movable Feast",  
"restaurant_id": "40361606" }
```

```
{ "_id": ObjectId("6106f99585e29051a03be22a"), "borough": "Queens", "cuisine": "Delicatessen", "name": "Sal'S Deli", "restaurant_id":  
"40361618" }
```

INDRAJEET ROTE – MongoDB Assignment

```
{ "_id" : ObjectId("6106f99585e29051a03be22d"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery",  
"restaurant_id" : "40361998" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22e"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service",  
"restaurant_id" : "40357437" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be231"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste  
The Tropics Ice Cream", "restaurant_id" : "40356731" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be232"), "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard",  
"restaurant_id" : "40356151" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be233"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id"  
: "30112340" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be234"), "borough" : "Queens", "cuisine" : "Chinese", "name" : "Ho Mei Restaurant",  
"restaurant_id" : "40362432" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be236"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "White Castle",  
"restaurant_id" : "40362344" }
```

19.

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

```
{ "_id" : ObjectId("6106f99585e29051a03be221"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street  
Kitchen", "restaurant_id" : "40359480" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22b"), "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli",  
"restaurant_id" : "40361708" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22c"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food",  
"restaurant_id" : "40361521" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22f"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery",  
"restaurant_id" : "40362264" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be230"), "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen",  
"restaurant_id" : "40362098" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be235"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Angelika Film Center",  
"restaurant_id" : "40362274" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be238"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And  
Restaurant", "restaurant_id" : "30191841" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be239"), "borough" : "Manhattan", "cuisine" : "Turkish", "name" : "The Country Cafe",  
"restaurant_id" : "40362715" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be23b"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Downtown Deli",  
"restaurant_id" : "40363021" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be23f"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Cafe Metro",  
"restaurant_id" : "40363298" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be242"), "borough" : "Manhattan", "cuisine" : "Sandwiches/Salads/Mixed Buffet", "name" :  
"Lexler Deli", "restaurant_id" : "40363426" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be244"), "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Olive'S", "restaurant_id" :  
"40363151" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be245"), "borough" : "Manhattan", "cuisine" : "Continental", "name" : "Lorenzo & Maria'S",  
"restaurant_id" : "40363630" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be249"), "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza",  
"restaurant_id" : "40363644" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be24a"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Berkely", "restaurant_id" :  
"40363685" }
```

INDRAJEET ROTE – MongoDB Assignment

```
{ "_id" : ObjectId("6106f99585e29051a03be24b"), "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza",  
"restaurant_id" : "40363945" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be24c"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Spoon Bread Catering",  
"restaurant_id" : "40364179" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be254"), "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Texas Rotisserie",  
"restaurant_id" : "40364304" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be256"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "21 Club", "restaurant_id" :  
"40364362" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be259"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Palm Restaurant",  
"restaurant_id" : "40364355" }
```

20.

```
db.addresses.find( {"grades.score":{$lte:10}}, {"restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } )  
  
{ "_id" : ObjectId("6106f99585e29051a03be21e"), "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island",  
"restaurant_id" : "40356442" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be21f"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop",  
"restaurant_id" : "30075445" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be220"), "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen",  
"restaurant_id" : "40358429" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be221"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street  
Kitchen", "restaurant_id" : "40359480" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be222"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers",  
"restaurant_id" : "40356649" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be223"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer",  
"restaurant_id" : "40356018" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be224"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" :  
"40357217" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be225"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel  
Ice Cream", "restaurant_id" : "40360076" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be226"), "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice  
Cream", "restaurant_id" : "40361322" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be227"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies",  
"restaurant_id" : "40361390" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be228"), "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods",  
"restaurant_id" : "40360045" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be229"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast",  
"restaurant_id" : "40361606" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be22a"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" :  
"40361618" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be22b"), "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli",  
"restaurant_id" : "40361708" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be22c"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food",  
"restaurant_id" : "40361521" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be22d"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery",  
"restaurant_id" : "40361998" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be22e"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service",  
"restaurant_id" : "40357437" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be22f"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery",  
"restaurant_id" : "40362264" }
```

INDRAJEET ROTE – MongoDB Assignment

```
{ "_id" : ObjectId("6106f99585e29051a03be230"), "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen",  
"restaurant_id" : "40362098" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be231"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste  
The Tropics Ice Cream", "restaurant_id" : "40356731" }
```

Type "it" for more

21.

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

```
{ "_id" : ObjectId("6106f99585e29051a03be21d"), "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen",  
"restaurant_id" : "40356068" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be21e"), "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island",  
"restaurant_id" : "40356442" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be21f"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop",  
"restaurant_id" : "30075445" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be221"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street  
Kitchen", "restaurant_id" : "40359480" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be222"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers",  
"restaurant_id" : "40356649" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be223"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer",  
"restaurant_id" : "40356018" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be224"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" :  
"40357217" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be225"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel  
Ice Cream", "restaurant_id" : "40360076" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be226"), "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice  
Cream", "restaurant_id" : "40361322" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be227"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies",  
"restaurant_id" : "40361390" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be228"), "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods",  
"restaurant_id" : "40360045" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be229"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast",  
"restaurant_id" : "40361606" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22a"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" :  
"40361618" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22b"), "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli",  
"restaurant_id" : "40361708" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22c"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food",  
"restaurant_id" : "40361521" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22d"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery",  
"restaurant_id" : "40361998" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22e"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service",  
"restaurant_id" : "40357437" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be22f"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery",  
"restaurant_id" : "40362264" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be230"), "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen",  
"restaurant_id" : "40362098" }
```

```
{ "_id" : ObjectId("6106f99585e29051a03be231"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste  
The Tropics Ice Cream", "restaurant_id" : "40356731" }
```


Type "it" for more

22.

```
db.addresses.find(
...
{
...
  "grades.date": ISODate("2014-08-11T00:00:00Z"),
...
  "grades.grade": "A",
...
  "grades.score": 11
...
},
...
{"restaurant_id": 1, "name": "Neary'S Pub", "grades": 1}
...
);

{ "_id" : ObjectId("6106f99585e29051a03be29a"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 13 }, {
"date" : ISODate("2013-07-22T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-03-14T00:00:00Z"), "grade" : "A", "score" :
12 }, { "date" : ISODate("2012-07-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-02-02T00:00:00Z"), "grade" : "A",
"score" : 10 }, { "date" : ISODate("2011-08-24T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "Neary'S Pub", "restaurant_id" :
"40365871" }

{ "_id" : ObjectId("6106f99585e29051a03be377"), "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" }

{ "_id" : ObjectId("6106f99685e29051a03be638"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 6 }, {
"date" : ISODate("2013-08-29T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-25T00:00:00Z"), "grade" : "C", "score"
: 2 }, { "date" : ISODate("2012-10-23T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-04-16T00:00:00Z"), "grade" : "A",
"score" : 11 }, { "date" : ISODate("2011-11-15T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Mustang Sally'S Restaurant",
"restaurant_id" : "40397374" }

{ "_id" : ObjectId("6106f99685e29051a03be846"), "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" }

{ "_id" : ObjectId("6106f99685e29051a03be93c"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 8 }, {
"date" : ISODate("2014-01-09T00:00:00Z"), "grade" : "B", "score" : 27 }, { "date" : ISODate("2013-06-11T00:00:00Z"), "grade" : "A", "score" :
4 }, { "date" : ISODate("2012-10-09T00:00:00Z"), "grade" : "B", "score" : 19 }, { "date" : ISODate("2012-05-10T00:00:00Z"), "grade" : "A",
"score" : 11 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "A", "score" : 9 } ], "name" : "Marino'S Pizza & Restaurant",
"restaurant_id" : "40560917" }

{ "_id" : ObjectId("6106f99685e29051a03beae6"), "grades" : [ { "date" : ISODate("2015-01-05T00:00:00Z"), "grade" : "A", "score" : 9 }, {
"date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2014-03-04T00:00:00Z"), "grade" : "B", "score" :
22 }, { "date" : ISODate("2013-09-13T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-01-16T00:00:00Z"), "grade" : "A",
"score" : 11 } ], "name" : "Gene'S Coffee Shop", "restaurant_id" : "40614916" }

{ "_id" : ObjectId("6106f99685e29051a03bec52"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 10 }, {
"date" : ISODate("2014-02-20T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2013-08-29T00:00:00Z"), "grade" : "A", "score" :
7 }, { "date" : ISODate("2013-03-29T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-03-07T00:00:00Z"), "grade" : "A",
"score" : 12 }, { "date" : ISODate("2011-11-18T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-12T00:00:00Z"),
"grade" : "P", "score" : 4 } ], "name" : "Union Cafe Restaurant", "restaurant_id" : "40698823" }

{ "_id" : ObjectId("6106f99685e29051a03bf0bc"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 10 }, {
"date" : ISODate("2013-08-28T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-09-05T00:00:00Z"), "grade" : "A", "score" :
11 }, { "date" : ISODate("2011-08-25T00:00:00Z"), "grade" : "A", "score" : 7 } ], "name" : "Jojo'S Pizza", "restaurant_id" : "40892913" }
```

23.

```
db.addresses.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});
```

INDRAJEET ROTE – MongoDB Assignment

```
{ "_id" : ObjectId("6106f99685e29051a03be846"), "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" }
```

24.

```
db.addresses.find({"address.coord.1":{$gt:42, $lte:52}},{"restaurant_id" : 1,"name":1,"coord":1});  
  
{ "_id" : ObjectId("6106f99585e29051a03be4ba"), "name" : "T.G.I. Friday'S", "restaurant_id" : "40387990" }  
  
{ "_id" : ObjectId("6106f99585e29051a03be4eb"), "name" : "T.G.I. Fridays", "restaurant_id" : "40388936" }  
  
{ "_id" : ObjectId("6106f99685e29051a03be745"), "name" : "Di Luvio'S Deli", "restaurant_id" : "40402284" }  
  
{ "_id" : ObjectId("6106f99685e29051a03be97f"), "name" : "La Caridad 78", "restaurant_id" : "40568285" }  
  
{ "_id" : ObjectId("6106f99685e29051a03bf04b"), "name" : "Bijan'S", "restaurant_id" : "40876618" }  
  
{ "_id" : ObjectId("6106f99685e29051a03bf060"), "name" : "Hyatt, Ny Central/Room Service", "restaurant_id" : "40879243" } { "_id" :  
ObjectId("6106f99685e29051a03bf075"), "name" : "Sports Center At Chelsea Piers (Sushi Bar)", "restaurant_id" : "40882356" }
```

25.

```
db.addresses.find().sort({"name":1})
```

26.

```
db.addresses.find().sort({"name":-1})
```

27.

```
db.addresses.find().sort({"cuisine":1,"borough":-1})
```

28

```
db.addresses.find({"address.street": {$exists:true}});
```

29

```
db.addresses.find({"address.coord": {$type:1}})
```

30

```
db.addresses.find({"grades.score": { $mod : [7, 0]}}, {"restaurant_id" :1, "name":1, "grades": 1})
```

31 db.addresses.find(

```
    { name :  
      { $regex : "mon.*", $options: "i" }  
    },  
  
    {  
      "name":1,  
      "borough":1,  
      "address.coord":1,  
      "cuisine" :1  
    }  
  );
```

```
db.addresses.find({ name : /. *mon.*/}, {"name":1,"borough":1, "address.coord":1, "cuisine" :1});
```

32

INDRAJEET ROTE – MongoDB Assignment

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

db.addresses.find({ name : /^Mad/}, {"name":1,"borough":1, "address.coord":1, "cuisine" :1});

{ "_id" : ObjectId("6106f99685e29051a03be762"), "address" : { "coord" : [ -73.9860597, 40.7431194 ] }, "borough" : "Manhattan",
  "cuisine" : "American ", "name" : "Madison Square" }

{ "_id" : ObjectId("6106f99685e29051a03be826"), "address" : { "coord" : [ -73.98302199999999, 40.742313 ] }, "borough" : "Manhattan",
  "cuisine" : "Indian", "name" : "Madras Mahal" }

{ "_id" : ObjectId("6106f99685e29051a03bead8"), "address" : { "coord" : [ -74.000002, 40.72735 ] }, "borough" : "Manhattan", "cuisine" :
  "American ", "name" : "Madame X" }

{ "_id" : ObjectId("6106f99685e29051a03beb87"), "address" : { "coord" : [ -73.98171959999999, 40.7499406 ] }, "borough" : "Manhattan",
  "cuisine" : "French", "name" : "Madison Bistro" }

{ "_id" : ObjectId("6106f99685e29051a03bec0a"), "address" : { "coord" : [ -73.9717845, 40.6897199 ] }, "borough" : "Brooklyn", "cuisine" :
  "African", "name" : "Madiba" }

{ "_id" : ObjectId("6106f99685e29051a03bef14"), "address" : { "coord" : [ -73.9040753, 40.9069011 ] }, "borough" : "Bronx", "cuisine" :
  "Italian", "name" : "Madison'S" }

{ "_id" : ObjectId("6106f99685e29051a03bef91"), "address" : { "coord" : [ -73.9886598, 40.7565811 ] }, "borough" : "Manhattan", "cuisine"
  : "Hotdogs", "name" : "Madame Tussaud'S" }

{ "_id" : ObjectId("6106f99685e29051a03befc8"), "address" : { "coord" : [ -73.95623719999999, 40.7761697 ] }, "borough" : "Manhattan",
  "cuisine" : "American ", "name" : "Mad River Bar & Grille" }

>
```