

Assignment-1

```
C:\mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/mongo_practice
For more information on usage: https://docs.mongodb.com/mongodb-shell.
C:\Users\DAMAHESW>mongosh "mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice" --apiVersion 1 --username Mahi
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> showdb
ReferenceError: showdb is not defined
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> show dbs
admin 340 KB 5.0.8 (API Version 1)
local 5.05 GB 1.3.1
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> use mongo_practice
already on db mongo_practice/.docs.mongodb.com/mongodb-shell/
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.createCollection('movies');
{ ok: 1 }
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title:"Fight Club", writer : "Chuck Palahniuk", year : "1999", actors : "[ Brad Pitt Edward Norton ]"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627ab797354af5d4dfe8b759") }
}

Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "Pulp Fiction", writer : "Quentin Tarantino", year :" 1994", actors :" [ John Travolta Uma Thurman ]"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627ab8ff354af5d4dfe8b75a") }
}

C:\mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/mongo_practice
2
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title :"Inglourious Basterds", writer : "Quentin Tarantino", year :" 2009", actors :" [ Brad Pitt Diane Kruger Eli Roth ]"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627abb01354af5d4dfe8b75b") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> ({ title : "The Hobbit: An Unexpected Journey", writer : "J.R.R. Tolkein", year : "2012", franchise : "The Hobbit"})
{
  title: 'The Hobbit: An Unexpected Journey',
  writer: 'J.R.R. Tolkein',
  year: '2012',
  franchise: 'The Hobbit'
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({ title : "The Hobbit: An Unexpected Journey", writer : "J.R.R. Tolkein", year : "2012", franchise : "The Hobbit"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627abc2e354af5d4dfe8b75c") }
}

C:\mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/mongo_practice
SyntaxError: Missing semicolon. (1:1)
> 1 | title : The Hobbit: The Desolation of Smaug writer : J.R.R. Tolkein
> 2 |
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "The hobbit: the desolation of smaug", writer : "j.r.r. tolkein", year : "2013", franchise : "the hobbit"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627abee9354af5d4dfe8b75d") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "The Hobbit: The Desolation of Smaug", writer : "J.R.R. Tolkein", year : "2013", 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 ring darkness."})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627abf5d354af5d4dfe8b75e") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "Pee Wee Herman's Big Adventure"})
ReferenceError: insert is not defined
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "Pee Wee Herman's Big Adventure"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627abf9f354af5d4dfe8b75f") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "Avatar"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627abfc9354af5d4dfe8b760") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find
[Function: find] AsyncFunction {
  returnsPromise: true,
  apiVersions: [ 1, Infinity ],
  returnType: 'Cursor',
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplicaSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  deprecated: false,
  platforms: [ 0, 1, 2 ],
  isDirectShellCommand: false,
```

```
mongosh mongo+srv://cluster0.plbuk.mongodb.net/mongo_practice
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find().pretty()
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: 1999,
    actors: ['Brad Pitt Edward Norton']
  },
  {
    _id: ObjectId("627ab8ff354af5d4dfe8b75a"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: ['John Travolta Uma Thurman']
  },
  {
    _id: ObjectId("627abb01354af5d4dfe8b75b"),
    title: 'Inglourious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: ['Brad Pitt Diane Kruger Eli Roth']
  },
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627abee9354af5d4dfe8b75d"),
    title: 'The hobbit: the desolation of smaug',
    writer: 'J.R.R. tolkein',
    year: 2013,
    franchise: 'the hobbit'
  },
  {
    _id: ObjectId("627abf5d354af5d4dfe8b75e"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit synopsis : Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'
  },
  {
    _id: ObjectId("627abf9f354af5d4dfe8b75f"),
    title: 'Pee Wee Herman's Big Adventure'
  },
  {_id: ObjectId("627abfc9354af5d4dfe8b760"), title: 'Avatar'}
]
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.findOne({writer : "Quentin Tarantino"})
{
  _id: ObjectId("627ab8ff354af5d4dfe8b75a"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',tin Tarantino')
  year: 1994,
  actors: ['John Travolta Uma Thurman']
}
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find({ writer : "Quentin Tarantino" })
[
  {
    _id: ObjectId("627ab8ff354af5d4dfe8b75a"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: ['John Travolta Uma Thurman']
  },
  {
    _id: ObjectId("627abb01354af5d4dfe8b75b"),
    title: 'Inglourious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: ['Brad Pitt Diane Kruger Eli Roth']
  }
]
```

```
mongosh mongo+srv://cluster0.plbuk.mongodb.net/mongo_practice
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find().pretty()
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: 1999,
    actors: ['Brad Pitt Edward Norton']
  },
  {
    _id: ObjectId("627ab8ff354af5d4dfe8b75a"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: ['John Travolta Uma Thurman']
  },
  {
    _id: ObjectId("627abb01354af5d4dfe8b75b"),
    title: 'Inglourious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: ['Brad Pitt Diane Kruger Eli Roth']
  },
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627abee9354af5d4dfe8b75d"),
    title: 'The hobbit: the desolation of smaug',
    writer: 'J.R.R. tolkein',
    year: 2013,
    franchise: 'the hobbit'
  },
  {
    _id: ObjectId("627abf5d354af5d4dfe8b75e"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit synopsis : Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'
  },
  {
    _id: ObjectId("627abf9f354af5d4dfe8b75f"),
    title: 'Pee Wee Herman's Big Adventure'
  },
  {_id: ObjectId("627abfc9354af5d4dfe8b760"), title: 'Avatar'}
]
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.findOne({writer : "Quentin Tarantino"})
{
  _id: ObjectId("627ab8ff354af5d4dfe8b75a"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',tin Tarantino')
  year: 1994,
  actors: ['John Travolta Uma Thurman']
}
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find({ writer : "Quentin Tarantino" })
[
  {
    _id: ObjectId("627ab8ff354af5d4dfe8b75a"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: ['John Travolta Uma Thurman']
  },
  {
    _id: ObjectId("627abb01354af5d4dfe8b75b"),
    title: 'Inglourious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: ['Brad Pitt Diane Kruger Eli Roth']
  }
]
```

```
c:\ mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DHARMAHESW>mongosh "mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice" --apiVersion 1 --username Mahi
Enter password: *****
Current Mongosh Log ID: 627b51641fd41a3774d2fe7
Connecting to: mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice?appName=mongosh+1.3.1
Using MongoDB: 5.0.8 (API Version 1)
Using Mongosh: 1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({franchise : "The Hobbit" })
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({franchise:"The Hobbit"})
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({year : {$gt:"1990",$gt:"2000"}})
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    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("627b4fd45aa57543d054a36b"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit'
  }
]
```

```
c:\ mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({year : {$gt:"1990",$gt:"2000"}})
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'
  },
  {
    _id: ObjectId("627b4fd45aa57543d054a36b"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit'
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({ franchise: "The Hobbit" })
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({ franchise: "The Hobbit" })
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit'
  }
]
```

```
mongosh mongo+srv://cluster0.plbuk.mongodb.net/mongo_practice
{
  "_id": ObjectId("627b4ee55aa57543d054a36a"),
  "title": "The Hobbit: The Desolation of Smaug",
  "writer": "J.R.R. Tolkein",
  "year": "2013",
  "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("627b4fd45aa57543d054a36b"),
  "title": "The Hobbit: The Desolation of Smaug",
  "writer": "J.R.R. Tolkein",
  "year": "2013",
  "franchise": "The Hobbit"
}
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({year : {$gt:"1990",$gt:"2000"}})
[
  {
    "_id": ObjectId("627abc2e354af5d4dfe8b75c"),
    "title": "The Hobbit: An Unexpected Journey",
    "writer": "J.R.R. Tolkein",
    "year": "2012",
    "franchise": "The Hobbit"
  },
  {
    "_id": ObjectId("627b4ee55aa57543d054a36a"),
    "title": "The Hobbit: The Desolation of Smaug",
    "writer": "J.R.R. Tolkein",
    "year": "2013",
    "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("627b4fd45aa57543d054a36b"),
    "title": "The Hobbit: The Desolation of Smaug",
    "writer": "J.R.R. Tolkein",
    "year": "2013",
    "franchise": "The Hobbit"
  }
]
```

```
mongosh mongo+srv://cluster0.plbuk.mongodb.net/mongo_practice
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({ franchise: "The Hobbit" })
[
  {
    "_id": ObjectId("627abc2e354af5d4dfe8b75c"),
    "title": "The Hobbit: An Unexpected Journey",
    "writer": "J.R.R. Tolkein",
    "year": "2012",
    "franchise": "The Hobbit"
  },
  {
    "_id": ObjectId("627b4ee55aa57543d054a36a"),
    "title": "The Hobbit: The Desolation of Smaug",
    "writer": "J.R.R. Tolkein",
    "year": "2013",
    "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("627b4fd45aa57543d054a36b"),
    "title": "The Hobbit: The Desolation of Smaug",
    "writer": "J.R.R. Tolkein",
    "year": "2013",
    "franchise": "The Hobbit"
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({ year : { $gt:"1990",$gt:"2000" }})
[
  {
    "_id": ObjectId("627abc2e354af5d4dfe8b75c"),
    "title": "The Hobbit: An Unexpected Journey",
    "writer": "J.R.R. Tolkein",
    "year": "2012",
    "franchise": "The Hobbit"
  },
  {
    "_id": ObjectId("627b4ee55aa57543d054a36a"),
    "title": "The Hobbit: The Desolation of Smaug",
    "writer": "J.R.R. Tolkein",
    "year": "2013",
    "franchise": "The Hobbit"
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    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("627b4fd45aa57543d054a36b"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit'
  }
]
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find({ franchise: "The Hobbit" })
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
},
{
  _id: ObjectId("627b4fd45aa57543d054a36b"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkein',
  year: '2013',
  franchise: 'The Hobbit'
}
]
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find({ year : { $gt:"1990", $lt:"2000" } })
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ Brad Pitt Edward Norton ]
  }
]
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find({year : {$gt:"1990",$lt:"2000"}})
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: [ Brad Pitt Edward Norton ]
  }
]
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Pulp Fiction",writer:"Quentin Tarantino",year: " 1994",actors:" [ John Travolta Uma Thurman ]"})
{
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.insert({title:"Pulp Fiction",writer:"Quentin Tarantino",year: "1994",actors:"[ John Travolta Uma Thurman ]"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627b53b5e15fdbf4f7e9ce1f") }
}
Atlas atlas-wsaud-shard-0 [primary] mongo_practice> db.movies.find({year : {$gt:"1990",$lt:"2000"}})
[
  {
```

```
mongosh mongo:cluster0/plbk.mongodb.net/mongo_practice
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({year : {$gt:"1990",$lt:"2000"}})
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: '[ Brad Pitt Edward Norton ]'
  },
  {
    _id: ObjectId("627b53b5e15fdbf4f7e9ce1f"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: '[ John Travolta Uma Thurman ]'
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({$or:[{year:{$gt:"2010"}},{year:{$lt:"2000"}]}])
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
    actors: '[ Brad Pitt Edward Norton ]'
  },
  {
    _id: ObjectId("627abb01354af5d4dfe8b75b"),
    title: 'Inglourious Basterds',
    writer: 'Quentin Tarantino',
    year: '2009',
    actors: '[ Brad Pitt Diane Kruger Eli Roth ]'
  },
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
]
```

```
mongosh mongo:cluster0/plbk.mongodb.net/mongo_practice
},
{
  _id: ObjectId("627b4ee55aa57543d054a36a"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkein',
  year: '2013',
  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("627b4fd45aa57543d054a36b"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkein',
  year: '2013',
  franchise: 'The Hobbit'
},
{
  _id: ObjectId("627b53b5e15fdbf4f7e9ce1f"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: '1994',
  actors: '[ John Travolta Uma Thurman ]'
}
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Inglourious Basterds",writer:"Quentin Tarantino",year:" 2009",actors:"[ Brad Pitt Diane Kruger Eli Roth ]"})
{
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title:"Inglourious Basterds",writer:"Quentin Tarantino",year:"2009",actors:"[ Brad Pitt Diane Kruger Eli Roth ]"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627b550ee15fdbf4f7e9ce20") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({$or:[{year:{$gt:"2010"}},{year:{$lt:"2000"}]}])
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: '1999',
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.find({$or:[{year:$gt:'2010'}],{year:$lt: '2000'}}))
[
  {
    _id: ObjectId("627ab797354af5d4dfe8b759"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: 1999,
    actors: ['Brad Pitt Edward Norton']
  },
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("627b4fd45aa57543d054a36b"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: '1994',
    actors: ['John Travolta Uma Thurman']
  }
]
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.update({title:"The Hobbit: An Unexpected Journey"}, {synopsis: "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."})
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.updateOne({title:"The Hobbit: An Unexpected Journey"}, {$set:{synopsis: "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.updateOne({title:"The Hobbit: The Desolation of Smaug"}, {$set:{synopsis:"The dwarves, along with Bilbo Baggin 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."}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.update({title:"Pulp Fiction"},{$set:{actors:"Samuel L. Jackson"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.add({title:"Pulp Fiction"},{$set:{actors:"Samuel L. Jackson"}})
TypeError: db.movies.add is not a function
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.insert({title:"Pulp Fiction"},{$set:{actors:"John Travolta Uma Thurman"}})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627b5dfee15fdbf4f7e9ce21") }
}
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Pulp Fiction"},{$set:{actors:"John Travolta Uma Thurman"}})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Pulp Fiction"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-wsaudd-shard-0 [primary] mongo_practice> db.movies.insert({title : "Pulp Fiction", writer : "Quentin Tarantino", year : "1994", actors : "[ John Travolta Uma Thurman ]"})
{
  acknowledged: true,
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Pulp Fiction"},{$set:{actors:"John Travolta Uma Thurman"}})
{
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Pulp Fiction"})
{
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.insert({title : "Pulp Fiction", writer : "Quentin Tarantino", year : "1994", actors : "[ John Travolta Uma Thurman ]"})
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("627b5ec5e15fdbf4f7e9ce22") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.update({title:"Pulp Fiction"},{$set:{actors:"Samuel L. Jackson"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.update({title:"Pulp Fiction"},{$push:{actors:"John Travolta Uma Thurman"}})
MongoServerError: The field 'actors' must be an array but is of type string in document {_id: ObjectId('627b5ec5e15fdbf4f7e9ce22')}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.updateOne({title:"Pulp Fiction"},{$push:{actors: "John Travolta Uma Thurman"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.update({title:"Pulp Fiction"},{$set:{actors:"Samuel L. Jackson"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({$text:$search:"Bilbo"})
Uncought:
SyntaxError: Unexpected token, expected "," (1:39)
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({synopsis:{$regex:"Bilbo"}})
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'
  },
  {
    _id: ObjectId("627b4fd45aa57543d054a36b"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit',
    synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring.'
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({synopsis:{$regex:"Gandalf"}})
[
  {
    _id: ObjectId("627b4fd45aa57543d054a36b"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit',
    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.'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]}))
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.'
  },
  {
    _id: ObjectId("627b4ee55aa57543d054a36a"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit',
    synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring.'
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}]}]})
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    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.'
  },
  {
    _id: ObjectId("627b4fd45aa57543d054a36b"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: '2013',
    franchise: 'The Hobbit',
    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.'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}]}]})
[
  {
    _id: ObjectId("627abc2e354af5d4dfe8b75c"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: '2012',
    franchise: 'The Hobbit',
    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.'
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title
... :
... "Pee Wee Herman's Big Adventure"
...
... Wq
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
Uncought:
syntaxError: Unexpected token, expected "," (5:0)
  3 | "Pee Wee Herman's Big Adventure"
  4 |
> 5 | Wq
    ^
  6 |

Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title : "Pee Wee Herman's Big Adventure"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.movies.deleteOne({title : "Avatar"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.createCollection('users');
{ ok: 1 }
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.users.insert({username : GoodGuyGreg first_name : "Good Guy" last_name : "Greg" username : ScumbagSteve full_name : first : "Scumbag" last : "Steve"})
Uncought:
syntaxError: Unexpected token, expected "," (1:40)
```

```
mongosh mongodb+srv://cluster0.plblk.mongodb.net/mongo_practice
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.users.insert({username : "GoodGuyGreg", first_name : "Good Guy", last_name : "Greg", username : "ScumbagSteve", full_name : "[first : Scumbag ,last : Steve]"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b6eefe15fdbf4f7e9ce23") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.createCollection('posts');
MongoServerError: Collection already exists. NS: mongo_practice.posts
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.insert({username : "GoodGuyGreg", title :" Passes out at party", body :" Wakes up early and cleans house"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b7453e15fdbf4f7e9ce24") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.insert({username : "GoodGuyGreg", title :" Steals your identity", body :" Raises your credit score"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b7542e15fdbf4f7e9ce25") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.insert({username : "GoodGuyGreg", title :" Reports a bug in your code", body :" Sends you a Pull Request"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b75d3e15fdbf4f7e9ce26") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.insert({username : "ScumbagSteve", title : "Borrows something", body :" Sells it"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b7666e15fdbf4f7e9ce27") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.insert({username : "ScumbagSteve", title : "Borrows everything", body : "The end"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b76cae15fdbf4f7e9ce28") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({username : "ScumbagSteve", title : "Forks your repo on github", body : "Sets to private"})
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b773ce15fdbf4f7e9ce29") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.createCollection('comments');
MongoServerError: Collection already exists. NS: mongo_practice.comments
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({username : GoodGuyGreg comment : Hope you got a good deal! post : [post_obj_id]
```

```
mongosh mongodb+srv://cluster0.plblk.mongodb.net/mongo_practice
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.deleteOne({username:"GoodGuyGreg",comment:" What's mine is yours!",post:{'$objectid' :"627b7666e15fdbf4f7e9ce27"}})
{
  acknowledged: true,
  deletedCount: 0
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({ username : "GoodGuyGreg", comment : "Hope you got a good deal!", post :ObjectId("627b7666e15fdbf4f7e9ce27") })
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b8267e15fdbf4f7e9ce2e") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> (To exit, press Ctrl+C again or Ctrl+D or type .exit)
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({ username : "GoodGuyGreg", comment : "What's mine is yours!", post :ObjectId("627b76cae15fdbf4f7e9ce28") })
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b82eef15fdbf4f7e9ce2f") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({ username : "GoodGuyGreg", comment : "Don't violate the licensing agreement!", post :ObjectId("627b773ce15fdbf4f7e9ce29") })
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b8392e15fdbf4f7e9ce30") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({ username : "ScumbagSteve", comment : "It still isn't clean", post :ObjectId("627b7453e15fdbf4f7e9ce24") })
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b8421e15fdbf4f7e9ce31") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.insert({ username : "ScumbagSteve", comment : "Denied your PR cause I found a hack", post :ObjectId("627b75d3e15fdbf4f7e9ce26") })
{
  acknowledged: true,
  insertedIds: { '$0': ObjectId("627b84a0e15fdbf4f7e9ce32") }
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.users.find().pretty()
[
  {
    _id: ObjectId("627b6eefe15fdbf4f7e9ce23"),
    username: 'ScumbagSteve',
    first_name: 'Scumbag',
    last_name: 'Steve'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.users.find().pretty()
[
  {
    _id: ObjectId("627b6eef0e15fdbf4f7e9ce23"),
    username: 'ScumbagSteve',
    first_name: 'Good Guy',
    last_name: 'Greg',
    full_name: ['first : Scumbag ,last : Steve']
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.find().pretty()
[
  {
    _id: ObjectId("627b7453e15fdbf4f7e9ce24"),
    username: 'GoodGuyGreg',
    title: ' Passes out at party',
    body: ' Wakes up early and cleans house'
  },
  {
    _id: ObjectId("627b7542e15fdbf4f7e9ce25"),
    username: 'GoodGuyGreg',
    title: ' Steals your identity',
    body: ' Raises your credit score'
  },
  {
    _id: ObjectId("627b75d3e15fdbf4f7e9ce26"),
    username: 'GoodGuyGreg',
    title: ' Reports a bug in your code',
    body: ' Sends you a Pull Request'
  },
  {
    _id: ObjectId("627b7666e15fdbf4f7e9ce27"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: ' Sells it'
  },
  {
    _id: ObjectId("627b76cae15fdbf4f7e9ce28"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
{
  _id: ObjectId("627b773ce15fdbf4f7e9ce29"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
}
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.find({ username:"GoodGuyGreg" })
[
  {
    _id: ObjectId("627b7453e15fdbf4f7e9ce24"),
    username: 'GoodGuyGreg',
    title: ' Passes out at party',
    body: ' Wakes up early and cleans house'
  },
  {
    _id: ObjectId("627b7542e15fdbf4f7e9ce25"),
    username: 'GoodGuyGreg',
    title: ' Steals your identity',
    body: ' Raises your credit score'
  },
  {
    _id: ObjectId("627b75d3e15fdbf4f7e9ce26"),
    username: 'GoodGuyGreg',
    title: ' Reports a bug in your code',
    body: ' Sends you a Pull Request'
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.posts.find({ username:"ScumbagSteve" })
[
  {
    _id: ObjectId("627b7666e15fdbf4f7e9ce27"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: ' Sells it'
  },
  {
    _id: ObjectId("627b76cae15fdbf4f7e9ce28"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
{
  _id: ObjectId("627b773ce15fdbf4f7e9ce29"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
}
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find().pretty()
[
  {
    _id: ObjectId("627b8267e15fdbf4f7e9ce2e"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("627b7666e15fdbf4f7e9ce27")
  },
  {
    _id: ObjectId("627b82ef015fdbf4f7e9ce2f"),
    username: 'GoodGuyGreg',
    comment: "What's mine is yours!",
    post: ObjectId("627b76cae15fdbf4f7e9ce28")
  },
  {
    _id: ObjectId("627b8392e15fdbf4f7e9ce30"),
    username: 'GoodGuyGreg',
    comment: "Don't violate the licensing agreement!",
    post: ObjectId("627b773ce15fdbf4f7e9ce29")
  },
  {
    _id: ObjectId("627b8421e15fdbf4f7e9ce31"),
    username: 'ScumbagSteve',
    comment: "It still isn't clean",
    post: ObjectId("627b7453e15fdbf4f7e9ce24")
  },
  {
    _id: ObjectId("627b84a0e15fdbf4f7e9ce32"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("627b75d3e15fdbf4f7e9ce26")
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ username:"GoodGuyGreg" })
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ username:"GoodGuyGreg" })
[
  {
    _id: ObjectId("627b8267e15fdbf4f7e9ce2e"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("627b7666e15fdbf4f7e9ce27")
  },
  {
    _id: ObjectId("627b82ef015fdbf4f7e9ce2f"),
    username: 'GoodGuyGreg',
    comment: "What's mine is yours!",
    post: ObjectId("627b76cae15fdbf4f7e9ce28")
  },
  {
    _id: ObjectId("627b8392e15fdbf4f7e9ce30"),
    username: 'GoodGuyGreg',
    comment: "Don't violate the licensing agreement!",
    post: ObjectId("627b773ce15fdbf4f7e9ce29")
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ username:"ScumbagSteve" })
[
  {
    _id: ObjectId("627b8421e15fdbf4f7e9ce31"),
    username: 'ScumbagSteve',
    comment: "It still isn't clean",
    post: ObjectId("627b7453e15fdbf4f7e9ce24")
  },
  {
    _id: ObjectId("627b84a0e15fdbf4f7e9ce32"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("627b75d3e15fdbf4f7e9ce26")
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ post: "Reports a bug in your code" })
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ post: "Reports a bug in your code" })
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/mongo_practice
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ post: ObjectId("627b84a0e15fdbf4f7e9ce32") })
Atlas atlas-wsauud-shard-0 [primary] mongo_practice> db.comments.find({ post: ObjectId("627b75d3e15fdbf4f7e9ce26") })
[
  {
    _id: ObjectId("627b84a0e15fdbf4f7e9ce32"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found a hack',
    post: ObjectId("627b75d3e15fdbf4f7e9ce26")
  }
]
Atlas atlas-wsauud-shard-0 [primary] mongo_practice
```

Assignment-2

```
Command Prompt
C:\Users\DAMAHESW>mongoimport --host cluster0-shard-00-01.plbuk.mongodb.net:27017 --db population --collection zipcodes --type json --file C:/Users/DAMAHESW/Downloads/zip.json
2022-05-11T20:26:12.995+0530    error connecting to host: could not connect to server: server selection error: server selection timeout, current topology: { Type: Single, Servers: [ { Addr: cluster0-shard-00-01.plbuk.mongodb.net:27017, Type: Unknown, Last error: connection() error occurred during connection handshake: connection(cluster0-shard-00-01.plbuk.mongodb.net:27017[-64]) socket was unexpectedly closed: EOF }, ] }

C:\Users\DAMAHESW>mongoimport --host cluster0-shard-00-01.plbuk.mongodb.net:27017 --db population --collection zipcodes --type json --file C:/Users/DAMAHESW/Downloads/zip.json
2022-05-11T20:27:16.829+0530    error connecting to host: could not connect to server: server selection error: server selection timeout, current topology: { Type: Single, Servers: [ { Addr: cluster0-shard-00-01.plbuk.mongodb.net:27017, Type: Unknown, Last error: connection() error occurred during connection handshake: connection(cluster0-shard-00-01.plbuk.mongodb.net:27017[-64]) socket was unexpectedly closed: EOF }, ] }

C:\Users\DAMAHESW>mongoimport --host cluster0-shard-00-01.plbuk.mongodb.net:27017 --db population --collection zipcodes --type json --file C:/Users/DAMAHESW/Downloads/zip.json
--authentication database admin --ssl --username Mahi --password Mahi
2022-05-11T20:32:56.472+0530    connected to: mongod@/cluster0-shard-00-01.plbuk.mongodb.net:27017/
2022-05-11T20:32:53.479+0530    [#####.....] population.zipcodes 539KB/3.03MB (17.3%)
2022-05-11T20:32:59.473+0530    [#####[.....] population.zipcodes 859KB/3.03MB (27.6%)
2022-05-11T20:33:02.484+0530    [######[.....] population.zipcodes 1.15MB/3.03MB (37.9%)
2022-05-11T20:33:05.488+0530    [######[.....] population.zipcodes 1.46MB/3.03MB (48.2%)
2022-05-11T20:33:08.480+0530    [######[.....] population.zipcodes 1.77MB/3.03MB (58.4%)
2022-05-11T20:33:11.481+0530    [######[.....] population.zipcodes 2.08MB/3.03MB (68.5%)
2022-05-11T20:33:14.479+0530    [######[.....] population.zipcodes 2.39MB/3.03MB (78.7%)
2022-05-11T20:33:17.483+0530    [######[.....] population.zipcodes 2.70MB/3.03MB (88.8%)
2022-05-11T20:33:19.264+0530    [######[.....] population.zipcodes 3.00MB/3.03MB (98.9%)
2022-05-11T20:33:19.264+0530    29353 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\DAMAHESW>db.zipcodes.find()
'db.zipcodes.find' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\DAMAHESW>db.zipcodes.find().pretty()
'db.zipcodes.find' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\DAMAHESW>db.zipcodes.find({$where:"Atlanta"}
```

```
mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.find({city:'ATLANTA'}, {state:'GA'})
[
  { _id: '14808', state: 'GA' },
  { _id: '30303', state: 'GA' },
  { _id: '30306', state: 'GA' },
  { _id: '30305', state: 'GA' },
  { _id: '30307', state: 'GA' },
  { _id: '30308', state: 'GA' },
  { _id: '30310', state: 'GA' },
  { _id: '30309', state: 'GA' },
  { _id: '30311', state: 'GA' },
  { _id: '30312', state: 'GA' },
  { _id: '30313', state: 'GA' },
  { _id: '30314', state: 'GA' },
  { _id: '30315', state: 'GA' },
  { _id: '30317', state: 'GA' },
  { _id: '30318', state: 'GA' },
  { _id: '30316', state: 'GA' },
  { _id: '30324', state: 'GA' },
  { _id: '30319', state: 'GA' },
  { _id: '30326', state: 'GA' },
  { _id: '30327', state: 'GA' }
]
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] population> "it"
it
Atlas atlas-wsauud-shard-0 [primary] population> it
[
  { _id: '30329', state: 'GA' },
  { _id: '30330', state: 'GA' },
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsaud-shard-0 [primary] population> it
[
  { _id: '30329', state: 'GA' },
  { _id: '30330', state: 'GA' },
  { _id: '30331', state: 'GA' },
  { _id: '30336', state: 'GA' },
  { _id: '30334', state: 'GA' },
  { _id: '30339', state: 'GA' },
  { _id: '30342', state: 'GA' },
  { _id: '30345', state: 'GA' },
  { _id: '30350', state: 'GA' },
  { _id: '30349', state: 'GA' },
  { _id: '30346', state: 'GA' },
  { _id: '30360', state: 'GA' },
  { _id: '46931', state: 'GA' },
  { _id: '49709', state: 'GA' },
  { _id: '61723', state: 'GA' },
  { _id: '63530', state: 'GA' },
  { _id: '67008', state: 'GA' },
  { _id: '68923', state: 'GA' },
  { _id: '71044', state: 'GA' },
  { _id: '75551', state: 'GA' }
]
Type "it" for more
Atlas atlas-wsaud-shard-0 [primary] population> it
no cursor '83e01', state: 'GA' ]
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$group:{_id:"ATLANTA",count:{$sum:1}} }])
[ { _id: 'ATLANTA', count: 41 } ]
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$group:{_id:"ATLANTA",count:{$sum:'pop'}}}])
Atlas atlas-wsaud-shard-0 [primary] population>
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsaud-shard-0 [primary] population>
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$group:{_id:"ATLANTA",totalpopulation:{$sum:'pop'}} }])
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{city:"ATLANTA"}},{$group:{_id:"ATLANTA",totalpopulation:{$sum:'$pop'}} }])
[ { _id: 'ATLANTA', totalpopulation: 630046 } ]
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{city:"ATLANTA",state:"GA"}},{$project:{city:1,state:1}}])
[
  { _id: '30303', city: 'ATLANTA', state: 'GA' },
  { _id: '30306', city: 'ATLANTA', state: 'GA' },
  { _id: '30305', city: 'ATLANTA', state: 'GA' },
  { _id: '30307', city: 'ATLANTA', state: 'GA' },
  { _id: '30308', city: 'ATLANTA', state: 'GA' },
  { _id: '30310', city: 'ATLANTA', state: 'GA' },
  { _id: '30309', city: 'ATLANTA', state: 'GA' },
  { _id: '30311', city: 'ATLANTA', state: 'GA' },
  { _id: '30312', city: 'ATLANTA', state: 'GA' },
  { _id: '30313', city: 'ATLANTA', state: 'GA' },
  { _id: '30314', city: 'ATLANTA', state: 'GA' },
  { _id: '30315', city: 'ATLANTA', state: 'GA' },
  { _id: '30317', city: 'ATLANTA', state: 'GA' },
  { _id: '30318', city: 'ATLANTA', state: 'GA' },
  { _id: '30316', city: 'ATLANTA', state: 'GA' },
  { _id: '30324', city: 'ATLANTA', state: 'GA' },
  { _id: '30319', city: 'ATLANTA', state: 'GA' },
  { _id: '30326', city: 'ATLANTA', state: 'GA' },
  { _id: '30327', city: 'ATLANTA', state: 'GA' },
  { _id: '30329', city: 'ATLANTA', state: 'GA' }
]
Type "it" for more
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:"$state",totalpopulation:{$sum:'$pop }}}])
[
  { _id: 'ME', totalpopulation: 1226648 },
  { _id: 'NV', totalpopulation: 4372982 },
  { _id: 'TN', totalpopulation: 4876457 },
  { _id: 'IN', totalpopulation: 5544136 },
  { _id: 'PA', totalpopulation: 11881643 },
  { _id: 'SC', totalpopulation: 3486703 },
  { _id: 'NH', totalpopulation: 1109252 },
  { _id: 'DE', totalpopulation: 666168 },
  { _id: 'OH', totalpopulation: 10846517 },
  { _id: 'WI', totalpopulation: 4891769 },
  { _id: 'IA', totalpopulation: 2776428 },
  { _id: 'VT', totalpopulation: 562758 },
  { _id: 'NC', totalpopulation: 628637 },
  { _id: 'ND', totalpopulation: 638272 },
  { _id: 'MT', totalpopulation: 798948 },
  { _id: 'FL', totalpopulation: 12686644 },
  { _id: 'VA', totalpopulation: 6181479 },
  { _id: 'MS', totalpopulation: 2573216 },
  { _id: 'IL', totalpopulation: 11427576 },
  { _id: 'SD', totalpopulation: 695397 }
]
Type "it" for more
Atlas atlas-wsaud-shard-0 [primary] population> it
[
  { _id: 'KS', totalpopulation: 2475285 },
  { _id: 'LA', totalpopulation: 4217595 },
  { _id: 'AR', totalpopulation: 2350725 },
  { _id: 'OK', totalpopulation: 3145585 },
  { _id: 'TX', totalpopulation: 16984601 },
  { _id: 'UT', totalpopulation: 1722856 },
  { _id: 'NM', totalpopulation: 1515669 },
  { _id: 'AK', totalpopulation: 544698 },
  { _id: 'OR', totalpopulation: 2842321 },
  { _id: 'MA', totalpopulation: 6016425 },
  { _id: 'WV', totalpopulation: 1793146 },
  { _id: 'KY', totalpopulation: 3675484 },
  { _id: 'DC', totalpopulation: 606900 },
  { _id: 'MI', totalpopulation: 9295297 },
  { _id: 'ND', totalpopulation: 7730188 }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:"$state",totalpopulation:{$sum:'$pop }}}],{$sort:{totalpopulation:-1}})
[
  { _id: 'CA', totalpopulation: 29754898 },
  { _id: 'NY', totalpopulation: 17996482 },
  { _id: 'TX', totalpopulation: 16984601 },
  { _id: 'FL', totalpopulation: 12686644 },
  { _id: 'PA', totalpopulation: 11881543 },
  { _id: 'IL', totalpopulation: 11427576 },
  { _id: 'OH', totalpopulation: 10846517 },
  { _id: 'MI', totalpopulation: 9295297 },
  { _id: 'ND', totalpopulation: 7730188 },
  { _id: 'NV', totalpopulation: 6628637 },
  { _id: 'GA', totalpopulation: 6478216 },
  { _id: 'VA', totalpopulation: 6181479 },
  { _id: 'MA', totalpopulation: 6016425 },
  { _id: 'IN', totalpopulation: 5544136 },
  { _id: 'MO', totalpopulation: 5110648 },
  { _id: 'WI', totalpopulation: 4891769 },
  { _id: 'TN', totalpopulation: 4876457 },
  { _id: 'WA', totalpopulation: 4866692 },
  { _id: 'MD', totalpopulation: 4781379 },
  { _id: 'MN', totalpopulation: 4372982 }
]
Type "it" for more
Atlas atlas-wsaud-shard-0 [primary] population> it
[
  { _id: 'LA', totalpopulation: 4217595 },
  { _id: 'AL', totalpopulation: 4040587 },
  { _id: 'KY', totalpopulation: 3675484 },
  { _id: 'AZ', totalpopulation: 3665228 },
  { _id: 'SC', totalpopulation: 3486703 },
  { _id: 'CO', totalpopulation: 3293759 },
  { _id: 'CT', totalpopulation: 3287116 },
  { _id: 'OK', totalpopulation: 3145585 },
  { _id: 'OR', totalpopulation: 2842321 },
  { _id: 'IA', totalpopulation: 2776428 },
  { _id: 'MS', totalpopulation: 2573216 },
  { _id: 'KS', totalpopulation: 2475285 },
  { _id: 'AR', totalpopulation: 2350725 },
  { _id: 'WV', totalpopulation: 1793146 }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
{
  "_id": "WV",
  "totalpopulation": 1793146
},
{
  "_id": "UT",
  "totalpopulation": 1722850
},
{
  "_id": "NE",
  "totalpopulation": 1578139
},
{
  "_id": "NM",
  "totalpopulation": 1515069
},
{
  "_id": "ME",
  "totalpopulation": 1226648
},
{
  "_id": "NV",
  "totalpopulation": 1201833
},
{
  "_id": "NH",
  "totalpopulation": 1109252
}
]
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] population>
Atlas atlas-wsauud-shard-0 [primary] population>
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:"$state",totalpopulation:{$sum:'$pop'}}},{$sort:{totalpopulation:-1}},{$limit:3}])
[
  {
    "_id": "CA",
    "totalpopulation": 29754890
  },
  {
    "_id": "NY",
    "totalpopulation": 17990402
  },
  {
    "_id": "TX",
    "totalpopulation": 16984601
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{city: "ATLANTA", state: "GA"}}, {$project:{city:1,state:1}}])
[
  {
    "_id": "30303",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30306",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30305",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30307",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30308",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30310",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30309",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30311",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30312",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30313",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30314",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30315",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30317",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30318",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30316",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30324",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30319",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30326",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30327",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30329",
    "city": "ATLANTA",
    "state": "GA"
  }
]
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] population> it
[
  {
    "_id": "30330",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30331",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30336",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30334",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30339",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30342",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30345",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30350",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30349",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30346",
    "city": "ATLANTA",
    "state": "GA"
  },
  {
    "_id": "30360",
    "city": "ATLANTA",
    "state": "GA"
  }
]
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{id:{state:"$state",city:"$city"},totalpopulation:{$sum:'$pop'}}}])
MongoServerError: The field 'id' must be an accumulator object
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:{state:"$state",city:"$city"},totalpopulation:{$sum:'$pop'}}}])
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:{state:'$state',city:'$city'},totalpopulation:{$sum: '$pop'}}},{$sort:{totalpopulation:-1}}])
[
  { _id: { state: 'IL', city: 'CHICAGO' }, totalpopulation: 2452177 },
  { _id: { state: 'NY', city: 'BROOKLYN' }, totalpopulation: 2300504 },
  {
    _id: { state: 'CA', city: 'LOS ANGELES' },
    totalpopulation: 2102295
  },
  { _id: { state: 'TX', city: 'HOUSTON' }, totalpopulation: 2095918 },
  {
    _id: { state: 'PA', city: 'PHILADELPHIA' },
    totalpopulation: 1610956
  },
  { _id: { state: 'NY', city: 'NEW YORK' }, totalpopulation: 1476790 },
  { _id: { state: 'NY', city: 'BRONX' }, totalpopulation: 1209548 },
  { _id: { state: 'CA', city: 'SAN DIEGO' }, totalpopulation: 1049298 },
  { _id: { state: 'MD', city: 'DETROIT' }, totalpopulation: 963243 },
  { _id: { state: 'TX', city: 'DALLAS' }, totalpopulation: 940191 },
  { _id: { state: 'AZ', city: 'PHOENIX' }, totalpopulation: 890853 },
  { _id: { state: 'FL', city: 'MIAMI' }, totalpopulation: 825232 },
  { _id: { state: 'CA', city: 'SAN JOSE' }, totalpopulation: 816653 },
  {
    _id: { state: 'TX', city: 'SAN ANTONIO' },
    totalpopulation: 811792
  },
  { _id: { state: 'MD', city: 'BALTIMORE' }, totalpopulation: 733081 },
  {
    _id: { state: 'CA', city: 'SAN FRANCISCO' },
    totalpopulation: 723993
  },
  { _id: { state: 'TN', city: 'MEMPHIS' }, totalpopulation: 632837 },
  { _id: { state: 'CA', city: 'SACRAMENTO' }, totalpopulation: 628279 },
  {
    _id: { state: 'FL', city: 'JACKSONVILLE' },
    totalpopulation: 610160
  },
  { _id: { state: 'GA', city: 'ATLANTA' }, totalpopulation: 609591 }
]
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] population> it
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:{state:'$state',city:'$city"},totalpopulation:{$sum: '$pop'}}},{$sort:{totalpopulation:-1}},{$limit:3}])
[
  { _id: { state: 'IL', city: 'CHICAGO' }, totalpopulation: 2452177 },
  { _id: { state: 'NY', city: 'BROOKLYN' }, totalpopulation: 2300504 },
  {
    _id: { state: 'CA', city: 'LOS ANGELES' },
    totalpopulation: 2102295
  }
]
Atlas atlas-wsauud-shard-0 [primary] population> db.zipcodes.aggregate([{$match:{state:'TX'}},{$group:{_id:{state:'$state",city:'$city"},totalpopulation:{$sum: '$pop'}}},{$sort:{totalpopulation:-1}},{$limit:3}])
[
  { _id: { state: 'TX', city: 'HOUSTON' }, totalpopulation: 2095918 },
  { _id: { state: 'TX', city: 'DALLAS' }, totalpopulation: 940191 },
  {
    _id: { state: 'TX', city: 'SAN ANTONIO' },
    totalpopulation: 811792
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/population
]
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:{state:"$state"},state:{'$first':'$state'},count:{$sum:1},population:{$sum:"$pop"}},{$project:{_id:'$state',avg:{$divide:[{"$population","$count"]}}}}])
[
  { _id: 'WV' , avg: null },
  { _id: 'DC' , avg: null },
  { _id: 'KY' , avg: null },
Atlas atlas-wsaud-shard-0 [primary] population>
  { _id: 'MI' , avg: null },
  { _id: 'ID' , avg: null },
  { _id: 'NV' , avg: null },
  { _id: 'CA' , avg: null },
  { _id: 'WY' , avg: null },
  { _id: 'MO' , avg: null },
  { _id: 'CO' , avg: null },
  { _id: 'AZ' , avg: null },
  { _id: 'RI' , avg: null },
  { _id: 'WA' , avg: null },
  { _id: 'MD' , avg: null },
  { _id: 'GA' , avg: null },
  { _id: 'AL' , avg: null },
  { _id: 'HI' , avg: null },
  { _id: 'CT' , avg: null },
  { _id: 'NE' , avg: null }
]
Type "it" for more
Atlas atlas-wsaud-shard-0 [primary] population>
Atlas atlas-wsaud-shard-0 [primary] population>
Atlas atlas-wsaud-shard-0 [primary] population> db.zipcodes.aggregate([{$group:{_id:{state:"$state"},state:{'$first':'$state'},count:{$sum:1},population:{$sum:"$pop"}},{$project:{_id:'$state',avg:{$divide:[{"$population","$count"]}}}},{$sort:{avg:-1}},{$limit:3}])
[
  { _id: 'DC' , avg: 25287.5 },
  { _id: 'CA' , avg: 19627.236147757256 },
Atlas atlas-wsaud-shard-0 [primary] population>
]
Atlas atlas-wsaud-shard-0 [primary] population> Atlas atlas-wsaud-shard-0 [primary] population> Atlas atlas-wsaud-shard-0 [primary] population>
```

Assignment-3

```
C:\Users\DAmAHESw>mongoimport --host cluster0-shard-00-01.plbuk.mongodb.net/restaurants
Microsoft Windows [Version 10.0.19844.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DAmAHESw>mongosh "mongodb+srv://cluster0.plbuk.mongodb.net/restaurants" --apiVersion 1 --username Mahi
Enter password: *****
Current Mongosh Log ID: 627d30cce96f76dc0f48405
Connecting to: mongodb+srv://cluster0.plbuk.mongodb.net/restaurants?appName=mongosh+1.3.1
Using MongoDB: 5.0.8 (API Version 1)
Using Mongosh: 1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-wsaud-shard-0 [primary] restaurants> show dbs
mongo_practice 262 kB
population 4.1 MB
restaurants 1.27 MB
admin 340 kB
local 4.41 GB
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.getCollectionNames()
[ "addresses" ]
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find()
[
  {
    _id: ObjectId("627d307229727c87ec15c0e6"),
    address: {
      building: '351',
      coord: [ -73.98513559999999, 40.7676919 ],
      street: 'West 35 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'Irish',
    grades: [
      {
        date: ISODate("2014-09-06T00:00:00.000Z"),
        grade: 'A',
        score: 10
      }
    ],
    name: 'Di Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("627d307229727c87ec15c0e7"),
    address: {
      building: '7114',
      coord: [ -73.9868506, 40.6199034 ],
      street: 'Avenue U',
      zipcode: '11234'
    },
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    grades: [
      {
        date: ISODate("2014-05-29T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
        score: 10
      }
    ]
  }
]
```

```
C:\Users\DAmAHESw>mongo
{
  "date": ISODate("2013-07-22T00:00:00.000Z"),
  "grade": 'A',
  "score": 11
},
{
  "date": ISODate("2012-07-31T00:00:00.000Z"),
  "grade": 'A',
  "score": 12
},
{
  "date": ISODate("2011-12-29T00:00:00.000Z"),
  "grade": 'A',
  "score": 12
},
{
  "name": 'Di Reynolds Pub And Restaurant',
  "restaurant_id": '30191841'
},
{
  "address": {
    "building": '7114',
    "coord": [ -73.9868506, 40.6199034 ],
    "street": 'Avenue U',
    "zipcode": '11234'
  },
  "borough": 'Brooklyn',
  "cuisine": 'Delicatessen',
  "grades": [
    {
      "date": ISODate("2014-05-29T00:00:00.000Z"),
      "grade": 'A',
      "score": 10
    },
    {
      "date": ISODate("2014-01-14T00:00:00.000Z"),
      "grade": 'A',
      "score": 10
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "Wilken's Fine Food",
  "restaurant_id": "40356483",
  "_id": ObjectId("627d307229727c87ec15c0e8"),
  "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": ISODate("2014-03-03T00:00:00.000Z"),
      "grade": "A",
      "score": 2
    },
    {
      "date": ISODate("2013-08-03T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    },
    {
      "date": ISODate("2012-07-18T00:00:00.000Z"),
      "grade": "A",
      "score": 10
    },
    {
      "date": ISODate("2012-03-09T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2011-10-14T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30675445",
  "_id": ObjectId("627d307229727c87ec15c0e9"),
  "address": {
    "building": "2300",
    "coord": [ -73.8786113, 40.8502883 ],
    "street": "Southern Boulevard",
    "zipcode": "10460"
  },
  "borough": "Bronx",
  "cuisine": "American",
  "grades": [
    {
      "date": ISODate("2014-05-28T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2013-09-11T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2013-01-24T00:00:00.000Z"),
      "grade": "A",
      "score": 10
    },
    {
      "date": ISODate("2011-11-23T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2011-03-10T00:00:00.000Z"),
      "grade": "B",
      "score": 14
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  score: 11
},
{
  date: ISODate("2013-06-19T00:00:00.000Z"),
  grade: 'A',
  score: 4
},
{
  date: ISODate("2012-06-15T00:00:00.000Z"),
  grade: 'A',
  score: 3
},
],
name: 'Wild Asia',
restaurant_id: '40357217'
},
{
  _id: ObjectId("627d307229727c87ec15c0ea"),
  address: {
    building: '1839',
    coord: [ -73.9482609, 40.6408271 ],
    street: 'Nostrand Avenue',
    zipcode: '11226'
  },
  borough: 'Brooklyn',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  grades: [
    {
      date: ISODate("2014-07-14T00:00:00.000Z"),
      grade: 'A',
      score: 12
    },
    {
      date: ISODate("2013-07-10T00:00:00.000Z"),
      grade: 'A',
      score: 8
    },
    {
      date: ISODate("2012-07-11T00:00:00.000Z"),
      grade: 'A',
      score: 5
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  score: 5
},
{
  date: ISODate("2012-02-23T00:00:00.000Z"),
  grade: 'A',
  score: 8
},
],
name: 'Taste The Tropics Ice Cream',
restaurant_id: '40356731'
},
{
  _id: ObjectId("627d307229727c87ec15c0eb"),
  address: {
    building: '6409',
    coord: [ -74.00528899999999, 40.628886 ],
    street: '11 Avenue',
    zipcode: '11219'
  },
  borough: 'Brooklyn',
  cuisine: 'American',
  grades: [
    {
      date: ISODate("2014-07-18T00:00:00.000Z"),
      grade: 'A',
      score: 12
    },
    {
      date: ISODate("2013-07-30T00:00:00.000Z"),
      grade: 'A',
      score: 12
    },
    {
      date: ISODate("2013-02-13T00:00:00.000Z"),
      grade: 'A',
      score: 11
    },
    {
      date: ISODate("2012-08-16T00:00:00.000Z"),
      grade: 'A',
      score: 2
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "grades": ["A", "B", "C", "D", "F"],
  "score": 2,
  "date": ISODate("2011-08-17T00:00:00.000Z"),
  "grade": "A",
  "score": 11
},
{
  "name": "Regina Caterers",
  "restaurant_id": "40356649"
},
{
  "_id": ObjectId("627d307229727c87ec15c0ec"),
  "address": {
    "building": "1269",
    "coord": [-73.871194, 40.6730975],
    "street": "Sutter Avenue",
    "zipcode": "11208"
  },
  "borough": "Brooklyn",
  "cuisine": "Chinese",
  "grades": [
    {
      "date": ISODate("2014-09-16T00:00:00.000Z"),
      "grade": "B",
      "score": 21
    },
    {
      "date": ISODate("2013-08-28T00:00:00.000Z"),
      "grade": "A",
      "score": 7
    },
    {
      "date": ISODate("2013-04-02T00:00:00.000Z"),
      "grade": "C",
      "score": 56
    },
    {
      "date": ISODate("2012-08-15T00:00:00.000Z"),
      "grade": "B"
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "grades": ["B", "C", "D", "F"],
  "score": 27,
  "date": ISODate("2012-08-15T00:00:00.000Z"),
  "grade": "B",
  "score": 27
},
{
  "name": "May May Kitchen",
  "restaurant_id": "40358429"
},
{
  "_id": ObjectId("627d307229727c87ec15c0ed"),
  "address": {
    "building": "7715",
    "coord": [-73.9973325, 40.61174889999999],
    "street": "18 Avenue",
    "zipcode": "11214"
  },
  "borough": "Brooklyn",
  "cuisine": "American",
  "grades": [
    {
      "date": ISODate("2014-04-16T00:00:00.000Z"),
      "grade": "A",
      "score": 5
    },
    {
      "date": ISODate("2013-04-23T00:00:00.000Z"),
      "grade": "A",
      "score": 2
    },
    {
      "date": ISODate("2012-04-24T00:00:00.000Z"),
      "grade": "A",
      "score": 5
    },
    {
      "date": ISODate("2011-12-16T00:00:00.000Z"),
      "grade": "B"
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  {
    date: ISODate("2012-04-24T00:00:00.000Z"),
    grade: 'A',
    score: 5
  },
  {
    date: ISODate("2011-12-16T00:00:00.000Z"),
    grade: 'A',
    score: 2
  }
],
name: 'C & C Catering Service',
restaurant_id: '40357437'
},
{
  _id: ObjectId("627d307229727c87ec15c0ee"),
  address: {
    building: '1',
    coord: [ -73.96926909999999, 40.7685235 ],
    street: 'East 66 Street',
    zipcode: '10065'
  },
  borough: 'Manhattan',
  cuisine: 'American',
  grades: [
    {
      date: ISODate("2014-05-07T00:00:00.000Z"),
      grade: 'A',
      score: 3
    },
    {
      date: ISODate("2013-05-03T00:00:00.000Z"),
      grade: 'A',
      score: 4
    },
    {
      date: ISODate("2012-04-30T00:00:00.000Z"),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate("2011-12-27T00:00:00.000Z"),
      grade: 'A',
      score: 9
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  {
    date: ISODate("2011-12-27T00:00:00.000Z"),
    grade: 'A',
    score: 0
  }
],
name: '1 East 66Th Street Kitchen',
restaurant_id: '40359480'
},
{
  _id: ObjectId("627d307229727c87ec15c0ef"),
  address: {
    building: '265-15',
    coord: [ -73.7632601, 40.7386417 ],
    street: 'Hillside Avenue',
    zipcode: '11004'
  },
  borough: 'Queens',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  grades: [
    {
      date: ISODate("2014-10-28T00:00:00.000Z"),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate("2013-09-18T00:00:00.000Z"),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate("2012-09-20T00:00:00.000Z"),
      grade: 'A',
      score: 13
    }
  ],
name: 'Carvel Ice Cream',
restaurant_id: '40361322'
},
{
  _id: ObjectId("627d307229727c87ec15c0f0"),
  address: {
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "_id": ObjectId("627d307229727c87ec15c0f0"),
  "address": {
    "building": "283",
    "coord": [ -73.97822040000001, 40.6435254 ],
    "street": "Church Avenue",
    "zipcode": "11218"
  },
  "borough": "Brooklyn",
  "cuisine": "Ice Cream, Gelato, Yogurt, Ices",
  "grades": [
    {
      "date": ISODate("2014-02-10T00:00:00.000Z"),
      "grade": "A",
      "score": 2
    },
    {
      "date": ISODate("2013-01-02T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2012-01-09T00:00:00.000Z"),
      "grade": "A",
      "score": 3
    },
    {
      "date": ISODate("2011-11-07T00:00:00.000Z"),
      "grade": "B",
      "score": 12
    },
    {
      "date": ISODate("2011-07-21T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    }
  ],
  "name": "Carvel Ice Cream",
  "restaurant_id": "40360076"
},
{

```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "Carvel Ice Cream",
  "restaurant_id": "40360076"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f1"),
  "address": {
    "building": "765",
    "coord": [ -73.9653967, 40.6064339 ],
    "street": "Kings Highway",
    "zipcode": "11223"
  },
  "borough": "Brooklyn",
  "cuisine": "Jewish/Kosher",
  "grades": [
    {
      "date": ISODate("2014-11-10T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2013-10-10T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2012-10-04T00:00:00.000Z"),
      "grade": "A",
      "score": 7
    },
    {
      "date": ISODate("2012-05-21T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2011-12-30T00:00:00.000Z"),
      "grade": "B",
      "score": 19
    }
  ],
  "name": "Seuda Foods",

```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "Seuda Foods",
  "restaurant_id": "40360045",
  "_id": ObjectId("627d307229727c87ec15c0f2"),
  "address": {
    "building": "284",
    "coord": [ -73.9829239, 40.6580753 ],
    "street": "Prospect Park West",
    "zipcode": "11215"
  },
  "borough": "Brooklyn",
  "cuisine": "American",
  "grades": [
    {
      "date": ISODate("2014-11-19T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2013-11-14T00:00:00.000Z"),
      "grade": "A",
      "score": 2
    },
    {
      "date": ISODate("2012-12-05T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2012-05-17T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    }
  ],
  "name": "The Movable Feast",
  "restaurant_id": "40361606"
},
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "The Movable Feast",
  "restaurant_id": "40361606",
  "_id": ObjectId("627d307229727c87ec15c0f3"),
  "address": {
    "building": "522",
    "coord": [ -73.95171, 40.767461 ],
    "street": "East 74 Street",
    "zipcode": "10021"
  },
  "borough": "Manhattan",
  "cuisine": "American",
  "grades": [
    {
      "date": ISODate("2014-09-02T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-12-19T00:00:00.000Z"),
      "grade": "B",
      "score": 16
    },
    {
      "date": ISODate("2013-05-28T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2012-12-07T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2012-03-29T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    }
  ],
  "name": "Glorious Food",
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "Glorious Food",
  "restaurant_id": "40561521"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f4"),
  "address": {
    "building": "469",
    "coord": [-73.961704, 40.662942],
    "street": "Flatbush Avenue",
    "zipcode": "11225"
  },
  "borough": "Brooklyn",
  "cuisine": "Hamburgers",
  "grades": [
    {
      "date": ISODate("2014-12-30T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    }
  ],
  "name": "Wendy's",
  "restaurant_id": "30112340"
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "name": "Wendy's",
  "restaurant_id": "30112340"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f5"),
  "address": {
    "building": "129-09",
    "coord": [-73.839297, 40.78147],
    "street": "29 Avenue",
    "zipcode": "11356"
  },
  "borough": "Queens",
  "cuisine": "Delicatessen",
  "grades": [
    {
      "date": ISODate("2014-08-16T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-08-27T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2012-09-26T00:00:00.000Z"),
      "grade": "A",
      "score": 7
    },
    {
      "date": ISODate("2011-09-29T00:00:00.000Z"),
      "grade": "A",
      "score": 10
    }
  ],
  "name": "Sal's Deli",
  "restaurant_id": "40561618"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f6")
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "_id": ObjectId("627d307229727c87ec15c0f6"),
  "address": {
    "building": "3406",
    "coord": [ -73.9402473999999, 40.7623288 ],
    "street": "18 Street",
    "zipcode": "11106"
  },
  "borough": "Queens",
  "cuisine": "Delicatessen",
  "grades": [
    {
      "date": ISODate("2014-03-19T00:00:00.000Z"),
      "grade": "A",
      "score": 3
    },
    {
      "date": ISODate("2013-03-13T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2012-03-27T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    },
    {
      "date": ISODate("2011-04-05T00:00:00.000Z"),
      "grade": "A",
      "score": 7
    }
  ],
  "name": "Steve Chu'S Deli & Grocery",
  "restaurant_id": "40361998"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f7"),
  "address": {
    "building": "759",
    "coord": [ -73.9925306, 40.7309346 ],
    "street": "Broadway",
    "zipcode": "10003"
  },
  "borough": "Manhattan",
  "cuisine": "Delicatessen",
  "grades": [
    {
      "date": ISODate("2014-01-21T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-01-04T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-06-07T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2012-01-17T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "name": "Bully'S Deli",
  "restaurant_id": "40361708"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f8"),
  "address": {
    "building": "6909",
    "coord": [ -74.0259567, 40.6353674 ],
    "street": "3 Avenue",
    "zipcode": "11205"
  },
  "borough": "Brooklyn",
  "cuisine": "American (New)",
  "grades": [
    {
      "date": ISODate("2014-01-21T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-01-04T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-06-07T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2012-01-17T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "name": "The Corner Deli",
  "restaurant_id": "40361709"
},
{
  "_id": ObjectId("627d307229727c87ec15c0f9"),
  "address": {
    "building": "1000",
    "coord": [ -73.9925306, 40.7309346 ],
    "street": "Broadway",
    "zipcode": "10003"
  },
  "borough": "Manhattan",
  "cuisine": "American (New)",
  "grades": [
    {
      "date": ISODate("2014-01-21T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-01-04T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-06-07T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2012-01-17T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "name": "Bully'S Deli",
  "restaurant_id": "40361708"
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "building": "759",
  "coord": [ -73.9925306, 40.7309346 ],
  "street": "Broadway",
  "zipcode": "10003"
},
{
  "borough": "Manhattan",
  "cuisine": "Delicatessen",
  "grades": [
    {
      "date": ISODate("2014-01-21T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-01-04T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-06-07T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2012-01-17T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "name": "Bully'S Deli",
  "restaurant_id": "40361708"
},
{
  "building": "6909",
  "coord": [ -74.0259567, 40.6353674 ],
  "street": "3 Avenue",
  "zipcode": "11205"
},
{
  "borough": "Brooklyn",
  "cuisine": "American (New)",
  "grades": [
    {
      "date": ISODate("2014-01-21T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-01-04T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-06-07T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2012-01-17T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "name": "The Corner Deli",
  "restaurant_id": "40361709"
},
{
  "building": "1000",
  "coord": [ -73.9925306, 40.7309346 ],
  "street": "Broadway",
  "zipcode": "10003"
},
{
  "borough": "Manhattan",
  "cuisine": "American (New)",
  "grades": [
    {
      "date": ISODate("2014-01-21T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-01-04T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-06-07T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2012-01-17T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "name": "Bully'S Deli",
  "restaurant_id": "40361708"
}
```

```
mongosh mongo+srv://cluster0.plbuk.mongodb.net/restaurants
{
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  grades: [
    {
      date: ISODate("2014-08-21T00:00:00.000Z"),
      grade: 'A',
      score: 4
    },
    {
      date: ISODate("2014-03-05T00:00:00.000Z"),
      grade: 'A',
      score: 3
    },
    {
      date: ISODate("2013-01-10T00:00:00.000Z"),
      grade: 'A',
      score: 10
    }
  ],
  name: 'Nordic Delicacies',
  restaurant_id: '40361390'
},
{
  _id: ObjectId("627d307229727c87ec15c0f9"),
  address: {
    building: '730',
    coord: [ -73.9680571999999, 40.7925587 ],
    street: 'Columbus Avenue',
    zipcode: '10025'
  },
  borough: 'Manhattan',
  cuisine: 'American',
  grades: [
    {
      date: ISODate("2014-09-12T00:00:00.000Z"),
      grade: 'B',
      score: 26
    },
    {
      date: ISODate("2013-08-28T00:00:00.000Z"),
      grade: 'A',
      score: 10
    }
  ]
}
```

```
mongosh mongo+srv://cluster0.plbuk.mongodb.net/restaurants
{
  date: ISODate("2013-08-28T00:00:00.000Z"),
  grade: 'A',
  score: 9
},
{
  date: ISODate("2013-03-25T00:00:00.000Z"),
  grade: 'B',
  score: 20
},
{
  date: ISODate("2012-02-14T00:00:00.000Z"),
  grade: 'A',
  score: 12
},
],
name: 'P & S Deli Grocery',
restaurant_id: '40362264'
}
]
Type "it" for more
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.aggregate({$group:{_id:{restaurant_id:"$restaurant_id",borough:"$borough",cuisine:"$cuisine"}}})
[
  {
    _id: {
      restaurant_id: '40545598',
      borough: 'Bronx',
      cuisine: 'Caribbean'
    }
  },
  {
    _id: {
      restaurant_id: '40798867',
      borough: 'Manhattan',
      cuisine: 'American'
    }
  },
  {
    _id: {
      restaurant_id: '40750612',
      borough: 'Manhattan',
      cuisine: 'American'
    }
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "_id": {
    "restaurant_id": "40750612",
    "borough": "Manhattan",
    "cuisine": "American"
  }
},
{
  "_id": {
    "restaurant_id": "40584030",
    "borough": "Manhattan",
    "cuisine": "Soups & Sandwiches"
  }
},
{
  "_id": {
    "restaurant_id": "40424437",
    "borough": "Manhattan",
    "cuisine": "Bakery"
  }
},
{
  "_id": {
    "restaurant_id": "40716092",
    "borough": "Manhattan",
    "cuisine": "Delicatessen"
  }
},
{
  "_id": {
    "restaurant_id": "40887092",
    "borough": "Manhattan",
    "cuisine": "French"
  }
},
{
  "_id": {
    "restaurant_id": "40399784",
    "borough": "Manhattan",
    "cuisine": "Mediterranean"
  }
},
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
},
{
  "_id": {
    "restaurant_id": "40401216",
    "borough": "Manhattan",
    "cuisine": "Spanish"
  }
},
{
  "_id": {
    "restaurant_id": "40720133",
    "borough": "Queens",
    "cuisine": "Bagels/Pretzels"
  }
},
{
  "_id": {
    "restaurant_id": "40567158",
    "borough": "Manhattan",
    "cuisine": "Italian"
  }
},
{
  "_id": {
    "restaurant_id": "40849313",
    "borough": "Bronx",
    "cuisine": "Chinese"
  }
},
{
  "_id": {
    "restaurant_id": "40610212",
    "borough": "Brooklyn",
    "cuisine": "Caribbean"
  }
},
{
  "_id": {
    "restaurant_id": "40388419",
    "borough": "Manhattan",
    "cuisine": "Pizza"
  }
},
{
  "_id": {
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  "_id": {
    "restaurant_id": "40388419",
    "borough": "Manhattan",
    "cuisine": "Pizza"
  }
},
{
  "_id": {
    "restaurant_id": "40686674",
    "borough": "Manhattan",
    "cuisine": "Bakery"
  }
},
{
  "_id": {
    "restaurant_id": "40718665",
    "borough": "Manhattan",
    "cuisine": "Bakery"
  }
},
{
  "_id": {
    "restaurant_id": "40729796",
    "borough": "Queens",
    "cuisine": "American"
  }
},
{
  "_id": {
    "restaurant_id": "40400962",
    "borough": "Manhattan",
    "cuisine": "Hamburgers"
  }
},
{
  "_id": {
    "restaurant_id": "40398642",
    "borough": "Bronx",
    "cuisine": "Latin (Cuban, Dominican, Puerto Rican, South & Central American)"
  }
},
{
  "_id": {
    "restaurant_id": "40376635",
    "borough": "Brooklyn",
    "cuisine": "Mediterranean"
  }
}
]
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({}, {_id:0, name:1, borough:1, restaurant_id:1, cuisine:1})
```

```
[{"borough": "Manhattan", "cuisine": "Irish", "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841"}, {"borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Milken's Fine Food", "restaurant_id": "40356483"}, {"borough": "Bronx", "cuisine": "Bakery", "name": "Morris Park Bake Shop", "restaurant_id": "30075445"}, {"borough": "Bronx", "cuisine": "American ", "name": "Wild Asia", "restaurant_id": "40357217"}, {"borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Taste The Tropics Ice Cream", "restaurant_id": "40356731"}]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  borough: 'Brooklyn',
  cuisine: 'American',
  name: 'Regina Caterers',
  restaurant_id: '40356649'
},
{
  borough: 'Brooklyn',
  cuisine: 'Chinese',
  name: 'May May Kitchen',
  restaurant_id: '40358429'
},
{
  borough: 'Brooklyn',
  cuisine: 'American',
  name: 'C & C Catering Service',
  restaurant_id: '40357437'
},
{
  borough: 'Manhattan',
  cuisine: 'American',
  name: '1 East 66Th Street Kitchen',
  restaurant_id: '40359480'
},
{
  borough: 'Queens',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  name: 'Carvel Ice Cream',
  restaurant_id: '40361322'
},
{
  borough: 'Brooklyn',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  name: 'Carvel Ice Cream',
  restaurant_id: '40366076'
},
{
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Seuda Foods',
  restaurant_id: '40366045'
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
  restaurant_id: '40366045'
},
{
  borough: 'Brooklyn',
  cuisine: 'American',
  name: 'The Movable Feast',
  restaurant_id: '40361606'
},
{
  borough: 'Manhattan',
  cuisine: 'American',
  name: 'Glorious Food',
  restaurant_id: '40361521'
},
{
  borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: 'Wendy\'S',
  restaurant_id: '30112340'
},
{
  borough: 'Queens',
  cuisine: 'Delicatessen',
  name: "Sal'S Deli",
  restaurant_id: '40361618'
},
{
  borough: 'Queens',
  cuisine: 'Delicatessen',
  name: "Steve Chu'S Deli & Grocery",
  restaurant_id: '40361998'
},
{
  borough: 'Manhattan',
  cuisine: 'Delicatessen',
  name: "Bully'S Deli",
  restaurant_id: '40361708'
},
{
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
```

```
mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/restaurants
{
  cuisine: 'Delicatessen',
  name: 'Nordic Delicacies',
  restaurant_id: '40361390'
},
{
  borough: 'Manhattan',
  cuisine: 'American',
  name: 'P & S Deli Grocery',
  restaurant_id: '40362264'
}
]
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({ },{_id:0,name:1,cuisine:1,borough:1,restaurant_id:1,"address.zipcode":1})
[
  {
    address: { zipcode: '10019' },
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    address: { zipcode: '11234' },
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Wilken's Fine Food',
    restaurant_id: '40356483'
  },
  {
    address: { zipcode: '10462' },
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30675445'
  },
  {
    address: { zipcode: '10460' },
    borough: 'Bronx',
    cuisine: 'American',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
]
```

```
mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/restaurants
[
  {
    address: { zipcode: '11226' },
    borough: 'Brooklyn',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Taste The Tropics Ice Cream',
    restaurant_id: '40356731'
  },
  {
    address: { zipcode: '11219' },
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    address: { zipcode: '11208' },
    borough: 'Brooklyn',
    cuisine: 'Chinese',
    name: 'May May Kitchen',
    restaurant_id: '40358429'
  },
  {
    address: { zipcode: '11214' },
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'C & C Catering Service',
    restaurant_id: '40357437'
  },
  {
    address: { zipcode: '10065' },
    borough: 'Manhattan',
    cuisine: 'American',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    address: { zipcode: '11004' },
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Carvel Ice Cream',
    restaurant_id: '40361322'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
{
  borough: 'Queens',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  name: 'Carvel Ice Cream',
  restaurant_id: '40361322'
},
{
  address: { zipcode: '11218' },
  borough: 'Brooklyn',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  name: 'Carvel Ice Cream',
  restaurant_id: '40360076'
},
{
  address: { zipcode: '11223' },
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Seuda Foods',
  restaurant_id: '40360045'
},
{
  address: { zipcode: '11215' },
  borough: 'Brooklyn',
  cuisine: 'American',
  name: 'The Movable Feast',
  restaurant_id: '40361606'
},
{
  address: { zipcode: '10021' },
  borough: 'Manhattan',
  cuisine: 'American',
  name: 'Glorious Food',
  restaurant_id: '40361521'
},
{
  address: { zipcode: '11225' },
  borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: 'Mendy\'s',
  restaurant_id: '30112340'
},
{

```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({ borough: "Bronx" }).limit(5)
[
  {
    _id: ObjectId("627d307229727c87ec15c0e8"),
    address: {
      building: '1007',
      coord: [ -73.856677, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Type "it" for more
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({borough:"Bronx"}).skip(5).limit(5)
[
  {
    _id: ObjectId("627d307229727c87ec15c124"),
    address: {
      building: '277',
      coord: [ -73.8941893, 40.8634684 ],
      street: 'East Kingsbridge Road',
      zipcode: '10458'
    },
    borough: 'Bronx',
    cuisine: 'Chinese',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2013-09-26T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2013-03-19T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2012-08-29T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-08-17T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
    ],
    name: 'Happy Garden',
  }
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
]
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({"grades.score":{$gt:90}})
[
  {
    _id: ObjectId("627d307229727c87ec15c245"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West 54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American',
    grades: [
      {
        date: ISODate("2014-08-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-03-28T00:00:00.000Z"),
        grade: 'C',
        score: 131
      },
      {
        date: ISODate("2013-09-25T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-04-08T00:00:00.000Z"),
        grade: 'B',
        score: 25
      },
      {
        date: ISODate("2012-10-15T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-10-19T00:00:00.000Z"),
        grade: 'A',
        score: 11
      }
    ]
  }
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({$and : [{"grades.score":{$gt:90}}, {"grades.score":{$lt:100}}]}))
[{"_id": ObjectId("627d307229727c87ec15c245"),
  "address": {
    "building": "65",
    "coord": [-73.9782725, 40.7624022],
    "street": "West 54 Street",
    "zipcode": "10019"
  },
  "borough": "Manhattan",
  "cuisine": "American",
  "grades": [
    {
      "date": ISODate("2014-08-22T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2014-03-28T00:00:00.000Z"),
      "grade": "C",
      "score": 131
    },
    {
      "date": ISODate("2013-09-25T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2013-04-08T00:00:00.000Z"),
      "grade": "B",
      "score": 25
    },
    {
      "date": ISODate("2012-10-15T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2011-10-19T00:00:00.000Z"),
      "grade": "A"
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({"address.coord.0":{$lt:-95.754168}})
[{"_id": ObjectId("627d307229727c87ec15c72a"),
  "address": {
    "building": "3707",
    "coord": [-101.8945214, 33.5197474],
    "street": "82 Street",
    "zipcode": "11372"
  },
  "borough": "Queens",
  "cuisine": "American",
  "grades": [
    {
      "date": ISODate("2014-06-04T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-11-07T00:00:00.000Z"),
      "grade": "B",
      "score": 19
    },
    {
      "date": ISODate("2013-05-17T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-08-29T00:00:00.000Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2012-04-03T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2011-11-16T00:00:00.000Z"),
      "grade": "A"
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsauid-shard-0 [primary] restaurants> db.addresses.find({$and:[{cuisine:"American"}, {"address.coord.0":{$lt:-65.754168}}, {"grades.score":{$gt:70}}]})) ^

[{"_id": ObjectId("627d307229727c87ec15c2e5"),
  "address": {
    "building": "345",
    "coord": [-73.9864626, 40.7266739],
    "street": "East 6 Street",
    "zipcode": "10003"
  },
  "borough": "Manhattan",
  "cuisine": "Indian",
  "grades": [
    {
      "date": ISODate("2014-09-15T00:00:00.000Z"),
      "grade": "A",
      "score": 5
    },
    {
      "date": ISODate("2014-01-14T00:00:00.000Z"),
      "grade": "A",
      "score": 8
    },
    {
      "date": ISODate("2013-05-30T00:00:00.000Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2013-04-24T00:00:00.000Z"),
      "grade": "B",
      "score": 2
    },
    {
      "date": ISODate("2012-10-01T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2012-04-06T00:00:00.000Z"),
      "grade": "C",
      "score": null
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
}
])
Atlas atlas-wsauid-shard-0 [primary] restaurants> db.addresses.find({$and :[ {cuisine: {$ne : "American"}}, {"grades.grade":"A"}, {"borough : {$ne : "Brooklyn"} }]})).sort({cuisine:-1})
[{"_id": ObjectId("627d307229727c87ec15c7f2"),
  "address": {
    "building": "89",
    "coord": [-73.9995899, 40.7168015],
    "street": "Baxter Street",
    "zipcode": "10013"
  },
  "borough": "Manhattan",
  "cuisine": "Vietnamese/Cambodian/Malaysia",
  "grades": [
    {
      "date": ISODate("2014-08-21T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2013-08-31T00:00:00.000Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2013-04-11T00:00:00.000Z"),
      "grade": "C",
      "score": 3
    },
    {
      "date": ISODate("2012-10-17T00:00:00.000Z"),
      "grade": "A",
      "score": 4
    },
    {
      "date": ISODate("2012-05-15T00:00:00.000Z"),
      "grade": "A",
      "score": 10
    }
  ]
}
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find ({$name:$regex: /Wil/},{_id:0,restaurant_id:1,cuisine:1,name:1})
[
  {
    cuisine: 'Delicatessen',
    name: 'Wilken'S Fine Food',
    restaurant_id: '40356483'
  },
  {
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    cuisine: 'Pizza',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979'
  }
]
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find ({$name:$regex: /ces$/},{_id:0,restaurant_id:1,cuisine:1,name:1})
[
  {
    cuisine: 'American ',
    name: 'Pieces',
    restaurant_id: '40399910'
  },
  {
    cuisine: 'American ',
    name: 'Good Shepherd Services',
    restaurant_id: '40403989'
  },
  {
    cuisine: 'American ',
    name: 'S.M.R Restaurant Services',
    restaurant_id: '40403857'
  },
  {
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'The Ice Box-Ralph'S Famous Italian Ices',
    restaurant_id: '40690899'
  },
  {
    cuisine: 'Jewish/Kosher',
    name: 'Alices',
    restaurant_id: '40782042'
  },
  {
    cuisine: 'American ',
    name: 'Re: Sources',
    restaurant_id: '40881979'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
]
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find ({$name:$regex: /reg/},{_id:0,restaurant_id:1,cuisine:1,name:1})
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find ({$name:$regex: /Reg/},{_id:0,restaurant_id:1,cuisine:1,name:1})
[
  {
    cuisine: 'American ',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    cuisine: 'Café/Coffee/Tea',
    name: 'Caffè Reggio',
    restaurant_id: '40369418'
  },
  {
    cuisine: 'American ',
    name: 'Regency Hotel',
    restaurant_id: '40382679'
  },
  {
    cuisine: 'American ',
    name: 'Regency Whist Club',
    restaurant_id: '40402377'
  },
  {
    cuisine: 'American ',
    name: 'Rego Park Café',
    restaurant_id: '40523342'
  },
  {
    cuisine: 'Pizza',
    name: 'Regina Pizza',
    restaurant_id: '40801325'
  },
  {
    cuisine: 'American ',
    name: 'Regal Entertainment Group',
    restaurant_id: '40891782'
  }
]
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({borough:"Bronx",cuisine:{$in: ["American","Chinese"]}})
[
  {
    _id: ObjectId("627d307229727c87ec15c108"),
    ...
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
[1]Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({borough:"Bronx",cuisine:$in: ["American","Chinese"]})
[
  {
    _id: ObjectId("627d307229727c87ec15c108"),
    address: {
      building: '123E',
      coord: [ -73.8893654, 40.81376179999999 ],
      street: '238 Spofford Ave',
      zipcode: '10474'
    },
    borough: 'Bronx',
    cuisine: 'Chinese',
    grades: [
      {
        date: ISODate("2013-12-30T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2013-01-08T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2012-06-12T00:00:00.000Z"),
        grade: 'B',
        score: 15
      }
    ],
    name: 'Happy Garden',
    restaurant_id: '40363289'
  },
  {
    _id: ObjectId("627d307229727c87ec15c124"),
    address: {
      building: '277',
      coord: [ -73.8941893, 40.8634684 ],
      street: 'East Kingsbridge Road',
      zipcode: '10458'
    },
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
[1]Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({$or: [{borough:"Staten Island"},{borough:"Brooklyn"}, {"borough":"Queens"}]}, {_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Milken's Fine Food',
    restaurant_id: '40356483'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Taste The Tropics Ice Cream',
    restaurant_id: '40356731'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Chinese',
    name: 'May May Kitchen',
    restaurant_id: '40358429'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'C & C Catering Service',
    restaurant_id: '40357437'
  },
  {
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Carvel Ice Cream',
    restaurant_id: '40361322'
  },
  {
    borough: 'Queens',
    cuisine: 'American',
    name: 'The Original Bagel Company',
    restaurant_id: '40361323'
  }
]
```

```
mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/restaurants
Type "it" for more
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({$or:[{"borough:$in :["Staten Island","Queens","Bronx","Brooklyn"]},{name:1,restaurant_id:1,borough:1,cuisine:1,_id:0}]})
[{"borough": "Manhattan", "cuisine": "Irish", "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841"}, {"borough": "Manhattan", "cuisine": "American", "name": "1 East 66Th Street Kitchen", "restaurant_id": "40359480"}, {"borough": "Manhattan", "cuisine": "American", "name": "Glorious Food", "restaurant_id": "40361521"}, {"borough": "Manhattan", "cuisine": "Delicatessen", "name": "Bully'S Deli", "restaurant_id": "40361708"}, {"borough": "Manhattan", "cuisine": "American", "name": "P & S Deli Grocery", "restaurant_id": "40362264"}, {"borough": "Manhattan", "cuisine": "Chicken", "name": "Harriet'S Kitchen", "restaurant_id": "40362098"}, {"borough": "Manhattan", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "4036649"}, {"borough": "Brooklyn", "cuisine": "American", "name": "Wilken'S Fine Food", "restaurant_id": "40356483"}, {"borough": "Bronx", "cuisine": "American", "name": "Morris Park Bake Shop", "restaurant_id": "30075445"}, {"borough": "Bronx", "cuisine": "American", "name": "Wild Asia", "restaurant_id": "40357217"}, {"borough": "Brooklyn", "cuisine": "American", "name": "Taste The Tropics Ice Cream", "restaurant_id": "40356731"}, {"borough": "Brooklyn", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "40356649"}, {"borough": "Brooklyn", "cuisine": "American", "name": "May May Kitchen", "restaurant_id": "40358429"}, {"borough": "Brooklyn", "cuisine": "American", "name": "C & C Catering Service", "restaurant_id": "40357437"}, {"borough": "Manhattan", "cuisine": "American", "name": "1 East 66Th Street Kitchen", "restaurant_id": "40359480"}]
```

```
mongosh mongoDB+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find( { "grades.score": { $lt: 10 } } , { restaurant_id: 1, name: 1, borough: 1,_id:0 })
[caught:
  {taxError: Unexpected token (1:36)
  borough: 'Manhattan',
  name: 'Dj Reynolds Pub And Restaurant', $lt: 10 }, ] , { restaurant_id: 1, name: 1, borough: 1,_id:0})
  restaurant_id: '30191841' } , | {
  borough: 'Brooklyn',
  name: 'Wilken'S Fine Food',
  restaurant_id: '40356483' },
  {
  borough: 'Bronx',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445' },
  { borough: 'Bronx', name: 'Wild Asia', restaurant_id: '40357217' },
  {
  borough: 'Brooklyn',
  name: 'Taste The Tropics Ice Cream',
  restaurant_id: '40356731' },
  {
  borough: 'Brooklyn',
  name: 'Regina Caterers',
  restaurant_id: '40356649' },
  {
  borough: 'Brooklyn',
  name: 'May May Kitchen',
  restaurant_id: '40358429' },
  {
  borough: 'Brooklyn',
  name: 'C & C Catering Service',
  restaurant_id: '40357437' },
  {
  borough: 'Manhattan',
  name: '1 East 66Th Street Kitchen',
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({$nor: [{cuisine:$in: ["American", "Chinese"]}], {name:/\N\G1_\w*/}}, {_id:0,restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'DJ Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Taste The Tropics Ice Cream',
    restaurant_id: '40356731'
  },
  {
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Carvel Ice Cream',
    restaurant_id: '40361322'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Carvel Ice Cream',
    restaurant_id: '40366076'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Seuda Foods',
    restaurant_id: '40366045'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Seuda Foods',
    restaurant_id: '40366045'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":11},{restaurant_id:1,_id:0,name:1,grades:1})
[
  {
    grades: [
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2013-07-22T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2013-03-14T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2012-07-02T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2012-02-02T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-08-24T00:00:00.000Z"),
        grade: 'A',
        score: 11
      }
    ],
    name: "Neary's Pub",
    restaurant_id: '40365871'
  },
  {
    grades: [

```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({$and:[{ "grades.1.grade": "A" }, { "grades.1.score": 9 }, { "grades.1.date": ISODate("2014-08-11T00:00:00Z") }]}, {_id:0,restaurant_id:1,name:1,grades:1})
[
  {
    grades: [
      {
        date: ISODate("2015-01-12T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2013-02-07T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2012-04-30T00:00:00.000Z"),
        grade: 'A',
        score: 11
      }
    ],
    name: 'Club Macanudo (Cigar Bar)',
    restaurant_id: '40526406'
  }
]
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({$and : [{ "address.coord.1": {$lte : 52}}]}, {_id:0,restaurant_id:1,name:1,address:1})
Uncaught:
SyntaxError: Unexpected token, expected ","
(1:101)
> 1 | db.addresses.find({$and : [{ "address.coord.1": {$lte : 52}}]}, {_id:0,restaurant_id:1,name:1,address:1})
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsauud-shard-0 [primary] restaurants> db.addresses.find({$and : [{ "address.coord.1": {$lte : 52}}]}, {_id:0,restaurant_id:1,name:1,address:1})
[
  {
    address: {
      building: '351',
      coord: [ -73.98513559999999, 40.7676919 ],
      street: 'West 57 Street',
      zipcode: '10019'
    },
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    address: {
      building: '7114',
      coord: [ -73.9668506, 40.6199034 ],
      street: 'Avenue U',
      zipcode: '11234'
    },
    name: 'Wilken's Fine Food',
    restaurant_id: '40356483'
  },
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    name: 'Morris Park Bake Shop',
    restaurant_id: '30875445'
  },
  {
    address: {
      building: '2300',
      coord: [ -73.8786113, 40.8502883 ],
      street: 'Southern Boulevard',
      zipcode: '10460'
    },
    name: 'Wild Asia',
    restaurant_id: '40357217'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find().sort( {"name":1})
[
  {
    _id: ObjectId("627d307329727c87ec15cd74"),
    address: {
      building: '129',
      coord: [ -73.962943, 40.685067 ],
      street: 'Gates Avenue',
      zipcode: '11238'
    },
    borough: 'Brooklyn',
    cuisine: 'Italian',
    grades: [
      {
        date: ISODate("2014-03-06T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2013-08-29T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-03-08T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-06-27T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2011-11-17T00:00:00.000Z"),
        grade: 'A',
        score: 12
      }
    ],
    name: '(Lewis Drug Store) Locanda Vini E Olii',
    restaurant_id: '40804423'
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find().sort( {"name":-1})
[
  {
    _id: ObjectId("627d307229727c87ec15c1a3"),
    address: {
      building: '6946',
      coord: [ -73.8811834, 40.7017759 ],
      street: 'Myrtle Avenue',
      zipcode: '11385'
    },
    borough: 'Queens',
    cuisine: 'German',
    grades: [
      {
        date: ISODate("2014-09-24T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-04-17T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2013-03-12T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2012-10-02T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2012-05-09T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2011-12-28T00:00:00.000Z"),
        grade: 'B'
      }
    ]
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find().sort( {"cuisine":1,"borough":-1})
[
  {
    _id: ObjectId("627d307229727c87ec15c7d4"),
    address: {
      building: '1345',
      coord: [ -73.999249, 40.768876 ],
      street: '2 Avenue',
      zipcode: '10021'
    },
    borough: 'Manhattan',
    cuisine: 'Afghan',
    grades: [
      {
        date: ISODate("2014-10-07T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2013-10-23T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2012-10-26T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2012-04-26T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-01-12T00:00:00.000Z"),
        grade: 'P',
        score: 10
      }
    ],
    name: 'Afghan Kebab House',
    restaurant_id: '40552886'
  }
]
```

```
Select mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({"address.street":{$exists: true,$ne: null}})
[
  {
    _id: ObjectId("627d307229727c87ec15c0e6"),
    address: {
      building: '351',
      coord: [ -73.98513559999999, 40.7676919 ],
      street: 'West 57 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'Irish',
    grades: [
      {
        date: ISODate("2014-09-06T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-07-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2012-07-31T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2011-12-29T00:00:00.000Z"),
        grade: 'A',
        score: 12
      }
    ],
    name: 'DJ Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("627d307229727c87ec15c0e7"),
    address: {
      building: '7114'
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find({ "address.coord": { $type: "double" } }, { _id: 0, address: 1 })
[
  {
    address: {
      building: '351',
      coord: [ -73.98513559999999, 40.7676919 ],
      street: 'West 57 Street',
      zipcode: '10019'
    }
  },
  {
    address: {
      building: '7114',
      coord: [ -73.96068506, 40.6199034 ],
      street: 'Avenue U',
      zipcode: '11234'
    }
  },
  {
    address: {
      building: '1007',
      coord: [ -73.856677, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    }
  },
  {
    address: {
      building: '2300',
      coord: [ -73.8786113, 40.8502883 ],
      street: 'Southern Boulevard',
      zipcode: '10460'
    }
  },
  {
    address: {
      building: '1839',
      coord: [ -73.9482609, 40.6408271 ],
      street: 'Nostrand Avenue',
      zipcode: '11226'
    }
  }
]
```

```
mongosh mongodb+srv://cluster0.plbuk.mongodb.net/restaurants
Atlas atlas-wsaud-shard-0 [primary] restaurants> db.addresses.find(
... {
...   "grades.score" :
...     { $mod:[7,0]}
...   },
... { restaurant_id:1,grades:1,name:1,_id:0}
... )
[
  {
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30675445'
  },
  {
    grades: [

```