

MONGODB-ASSIGNMENT -1

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ABC>mongod
'mongod' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ABC>mongod
'mongod' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ABC>mongod
'mongod' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ABC>mongosh "mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0" --username Mrunal -password Mrunal1222
Current MongoDB Log ID: 6173bd5d5f5d217705a4f883
Connecting to: mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Using MongoDB: 4.4.10
Using Mongosh: 1.1.0

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

Atlas atlas-ro2vf9-shard-0 [primary] Cluster0> show dbs
admin 348 kB
local 3.8 GB
Atlas atlas-ro2vf9-shard-0 [primary] Cluster0> use test
switched to db test
Atlas atlas-ro2vf9-shard-0 [primary] test> db.createCollection('Demo');
{ ok: 1 }
Atlas atlas-ro2vf9-shard-0 [primary] test> db.Demo.insert({"name":"ABC"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6173c0759f0e1b53ff97f280") }
}
Atlas atlas-ro2vf9-shard-0 [primary] test> use mongo_practice
switched to db mongo_practice
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.createCollection("movies");
{ ok: 1 }
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find().pretty()
[
  {
    _id: ObjectId("6173d7919f0e1b53ff97f282"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuk',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("6173dad9f0e1b53ff97f283"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("6173db679f0e1b53ff97f284"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  },
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f285"),
    title: 'The Hobbit:An Unexpected Journey',
    writer: 'J.R.R.Tolkien',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f286"),
    title: 'The Hobbit:The Desolation of Smaug',
    writer: 'J.R.R.Tolkien',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
    title: 'The Hobbit:The Battle of the five armies',
    writer: 'J.R.R.Tolkien',
    year: 2012,
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling in to the hands of a rising dark'
  }
]
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
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 in to the hands of a rising dark
ness.'
},
{
  _id: ObjectId("6173dd9b9f0e1b53ff97f288"),
  title: 'Pee Wee Herman's Big adventure'
},
{
  _id: ObjectId("6173dd9b9f0e1b53ff97f289"), title: 'avatar' }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({"writer": "Quentin Tarantino"});
[
  {
    _id: ObjectId("6173dad9f0e1b53ff97f283"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("6173db679f0e1b53ff97f284"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({"franchise": "The Hobbit"});
[
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f285"),
    title: 'The Hobbit: An Unexpected Journey',
    writer: 'J.R.R. Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f286"),
    title: 'The Hobbit: The Desolation of Smaug',
    writer: 'J.R.R. Tolkein',
    year: 2013,
    franchise: 'The Hobbit'
  }
],
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
writer: 'J.R.R. Tolkein',
year: 2013,
franchise: 'The Hobbit'
},
{
  _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
  title: 'The Hobbit: The Battle of the five armies',
  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 in to the hands of a rising dark
ness.'
}
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({"actors": "Brad Pitt"});
[
  {
    _id: ObjectId("6173d7919f0e1b53ff97f282"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("6173db679f0e1b53ff97f284"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({"year": {$gt:1990,$lt:2000}}).pretty();
[
  {
    _id: ObjectId("6173d7919f0e1b53ff97f282"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("6173dad9f0e1b53ff97f283"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  }
]
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: 1999,
  actors: [ 'John Travolta', 'Uma Thurman' ]
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$or:[{"year":{$lt:2000}},{"year":{$gt:2010}}]}).pretty();
[
  {
    _id: ObjectId("6173d7919f0e1b53ff97f282"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("6173dad9f0e1b53ff97f283"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f285"),
    title: 'The Hobbit:An Unexpected Journey',
    writer: 'J.R.R.Tolkein',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f286"),
    title: 'The Hobbit:The Desolation of Smaug',
    writer: 'J.R.R.Tolkein',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
    title: 'The Hobbit:The Battle of the five armies',
    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 in to the hands of a rising dark
ness.'
  }
]
```

Insertion:-

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

```
db.movies.insert({"title":"Pulp Fiction","writer":"Quentin Tarantino","year":1999,"actors":["John Travolta","Uma Thurman"]});
```

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

```
db.movies.insertMany([{"title":"The Hobbit:An Unexpected Journey","writer":"J.R.R.Tolkein","year":2012,"franchise":"The Hobbit"}, {"title":"The Hobbit:The Desolation of Smaug","writer":"J.R.R.Tolkein","year":2013,"franchise":"The Hobbit"}]);
```

```
db.movies.insertMany([{"title":"The Hobbit:The Battle of the five armies","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 in to the hands of a rising darkness."}, {"title":"Pee Wee Herman's Big adventure"}, {"title":"avatar"}]);
```

Query / Find Documents:-

To get all documents:-

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

To get all documents with writer set to "Quentin Tarantino":-

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

To get all documents where actors include "Brad Pitt":-

```
db.movies.find({"actors":"Brad Pitt"});
```

To get all documents with franchise set to "The Hobbit":-

```
db.movies.find({"franchise":"The Hobbit"});
```

To get all movies released in 90s:-

```
db.movies.find({"year":{"$gt:1990,$lt:2000}}).pretty();
```

To get all movies released before the year 2000 or after 2010:-

```
db.movies.find({$or:[{"year":{"$lt:2000}},{"year":{"$gt:2010"}}]}).pretty();
```

Update Documents:-

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{
  _id: ObjectId("6173dcbd9f0e1b53ff97f286"),
  title: 'The Hobbit:The Desolation of Smaug',
  writer: 'J.R.R.Tolkein',
  year: 2013,
  franchise: 'The Hobbit'
},
{
  _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
  title: 'The Hobbit:The Battle of the five armies',
  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 in to the hands of a rising dark
ness.'
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.update({"title":"The Hobbit:An Unexpected Journey"},{$set:{"synopsis":"A reluctant hobbit,Bilbo Baggins,se
ts out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.update({"title":"The Hobbit:The Desolation of Smaug"},{$set:{"synopsis":"The dwarves, along with Bilbo Ba
ggins 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-ro2vf9-shard-0 [primary] mongo_practice> db.movies.update({"title":"Pulp Fiction"},{$push:{"actors":"Samuel L. Jackson"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice>
```

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

```
db.movies.update({"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."}});
```

2. Add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."

```
db.movies.update({"title":"The Hobbit:The Desolation of Smaug"},{$set:{"synopsis":"The dwarves, along
with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from
Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."}});
```

3.Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

```
db.movies.update({"title":"Pulp Fiction"},{$push:{"actors":"Samuel L. Jackson"}});
```

TEXT SEARCH:-

```
Select mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$synopsis:{$search:"Bilbo"}}).pretty();
MongoServerError: unknown top level operator: $synopsis
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.createIndex({"synopsis":"s"});
MongoServerError: Error in specification { key: { synopsis: "s" }, name: "synopsis_s" } :: caused by :: Unknown index plugin 's'
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.createIndex({"synopsis":"text"});
synopsis_text
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$synopsis_text:{$search:"Bilbo"}}).pretty();
MongoServerError: unknown top level operator: $synopsis_text
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$text:{$search:"Bilbo"}}).pretty();
[
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f286"),
    title: 'The Hobbit:The Desolation of Smaug',
    writer: 'J.R.R.Tolkein',
    year: 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey,continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring.'
  },
  {
    _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
    title: 'The Hobbit:The Battle of the five armies',
    writer: 'J.R.R.Tolkein',
    year: 2012,
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling in to the hands of a rising dark mass.'
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$text:{$search:"Gandalf"}}).pretty();
[
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f286"),
    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.b3io6.mongodb.net/Cluster0
{
  title: 'Pee Wee Herman's Big adventure'
},
{
  _id: ObjectId("6173dd9b9f0e1b53ff97f289"), title: 'avatar' }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$and:[{$synopsis:{$regex:"Bilbo"}}, {$synopsis:{$not:/Gandalf/}}]});
[
  {
    _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
    title: 'The Hobbit:The Battle of the five armies',
    writer: 'J.R.R.Tolkein',
    year: 2012,
    franchise: 'The Hobbit',
    synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling in to the hands of a rising dark mass.'
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$or:[{$synopsis:{$regex:"dwarves"}}, {$synopsis:{$regex:"hobbit"}}]});
[
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f285"),
    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("6173dcbd9f0e1b53ff97f286"),
    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-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find({$and:[{$synopsis:{$regex:"gold"}}, {$synopsis:{$regex:"dragon"}}]});
[
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f285"),
    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.'
  }
]
```

To find all the movies that have synopsis that contains the word "Bilbo":-

```
db.movies.createIndex({"synopsis":"text"});  
  
synopsis_text  
  
db.movies.find({$text:{$search:"Bilbo"}}).pretty();
```

To find all the movies that have synopsis that contains the word "Gandalf":-

```
db.movies.find({$text:{$search:"Gandalf"}}).pretty();
```

To find all the movies that have synopsis that contains the word "Bilbo" and not the word "Gandalf":-

```
db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]});
```

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

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

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

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

Delete Documents:-

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.remove({_id: ObjectId("6173dd9b9f0e1b53ff97f288")});
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.remove({_id: ObjectId("6173dd9b9f0e1b53ff97f289")});
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.movies.find().pretty();
[
  {
    _id: ObjectId("6173d7919f0e1b53ff97f282"),
    title: 'Fight Club',
    writer: 'Chuck Palahniuko',
    year: 1999,
    actors: [ 'Brad Pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("6173dadc9f0e1b53ff97f283"),
    title: 'Pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: [ 'John Travolta', 'Uma Thurman', 'Samuel L. Jackson' ]
  },
  {
    _id: ObjectId("6173db679f0e1b53ff97f284"),
    title: 'Inglorious Basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  },
  {
    _id: ObjectId("6173dcbd9f0e1b53ff97f285"),
    title: 'The Hobbit:An Unexpected Journey',
    writer: 'J.R.R.Tolkein',
    year: 2012,
    franchise: 'The Hobbit',
    synopsis: 'A reluctant hobbit,Bilo 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("6173dcbd9f0e1b53ff97f286"),
    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.'
  },
  {
    _id: ObjectId("6173dd9b9f0e1b53ff97f287"),
    title: 'The Hobbit:The Battle of the five armies',
    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 in to the hands of a rising dark ness.'
  }
]
```

To delete the movie "Pee Wee Herman's Big Adventure":-

```
db.movies.remove({_id: ObjectId("6173dd9b9f0e1b53ff97f288")});
```


To delete the movie "Avatar":-

```
db.movies.remove({_id: ObjectId("6173dd9b9f0e1b53ff97f289")});
```

Relationships:-

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ABC>mongosh "mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0" --username Mrunal -password Mrunal1222
Current Mongosh Log ID: 617592e775126b86110ac025
Connecting to:      mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Using MongoDB:      4.4.10
Using Mongosh:       1.1.0

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

Atlas atlas-ro2vf9-shard-0 [primary] Cluster0> use mongo_practice
switched to db mongo_practice
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> show collections
movies
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> use mongo_practice
already on db mongo_practice
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.createCollection("users");
{ ok: 1 }
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> show collections
movies
users
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.users.insertMany([{"username":"GoodGuyGreg","first_name":"Good Guy","last_name":"Greg"}, {"username":"ScumbagSteve", "full_name":{"first":"Scumbag","last":"Steve"}}]);
{
  acknowledged: true,
  insertedIds: {
    0: ObjectId("617594851c8504409021bba3"),
    1: ObjectId("617594851c8504409021bba4")
  }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> use mongo_practice
already on db mongo_practice
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.createCollection("posts");
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.posts.insertMany([{"username":"GoodGuyGreg","title":"Passes out at party","body":"Wakes up early and cleans house"}, {"username":"GoodGuyGreg","title":"Steals your identity","body":"Raises your credit score"}, {"username":"GoodGuyGreg","title":"Reports a bug in your code","body":"Sends you a Pull Request"}, {"username":"ScumbagSteve","title":"Borrows something","body":"Sells it"}, {"username":"ScumbagSteve","title":"Borrows everything","body":"The end"}, {"username":"ScumbagSteve","title":"Forks your repo on github","body":"Sets to private"}]);
{
  acknowledged: true,
  insertedIds: {
    0: ObjectId("617596a31c8504409021bba5"),
    1: ObjectId("617596a31c8504409021bba6"),
    2: ObjectId("617596a31c8504409021bba7")
  }
}
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
acknowledged: true,
insertedIds: {
  '0': ObjectId("617596a31c8504409021bba5"),
  '1': ObjectId("617596a31c8504409021bba6"),
  '2': ObjectId("617596a31c8504409021bba7"),
  '3': ObjectId("617596a31c8504409021bba8"),
  '4': ObjectId("617596a31c8504409021bba9"),
  '5': ObjectId("617596a31c8504409021bbaa")
}
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.createCollection("comments");
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.posts.find().pretty();
{
  _id: ObjectId("617596a31c8504409021bba5"),
  username: 'GoodGuyGreg',
  title: 'Passes out at party',
  body: 'Wakes up early and cleans house'
},
{
  _id: ObjectId("617596a31c8504409021bba6"),
  username: 'GoodGuyGreg',
  title: 'Steals your identity',
  body: 'Raises your credit score'
},
{
  _id: ObjectId("617596a31c8504409021bba7"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
  body: 'Sends you a Pull Request'
},
{
  _id: ObjectId("617596a31c8504409021bba8"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it'
},
{
  _id: ObjectId("617596a31c8504409021bba9"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end'
},
{
  _id: ObjectId("617596a31c8504409021bbaa"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
},
{
  _id: ObjectId("617596a31c8504409021bbab"),
  username: 'GoodGuyGreg',
  title: 'Hope you got a good deal!',
  body: 'What's mine is yours!'
},
{
  _id: ObjectId("617596a31c8504409021bbac"),
  username: 'GoodGuyGreg',
  title: 'Don't violate the licensing agreement!',
  body: ''
},
{
  _id: ObjectId("617596a31c8504409021bbad"),
  username: 'GoodGuyGreg',
  title: 'Don't violate the licensing agreement!',
  body: ''
}
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{
  _id: ObjectId("617596a31c8504409021bba7"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
  body: 'Sends you a Pull Request'
},
{
  _id: ObjectId("617596a31c8504409021bba8"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it'
},
{
  _id: ObjectId("617596a31c8504409021bba9"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end'
},
{
  _id: ObjectId("617596a31c8504409021bbaa"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
},
{
  _id: ObjectId("617596a31c8504409021bbab"),
  username: 'GoodGuyGreg',
  title: 'Hope you got a good deal!',
  body: 'What's mine is yours!'
},
{
  _id: ObjectId("617596a31c8504409021bbac"),
  username: 'GoodGuyGreg',
  title: 'Don't violate the licensing agreement!',
  body: ''
},
{
  _id: ObjectId("617596a31c8504409021bbad"),
  username: 'GoodGuyGreg',
  title: 'Don't violate the licensing agreement!',
  body: ''
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("617596a31c8504409021bba8")});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617598d81c8504409021bbab") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("617596a31c8504409021bba9")});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617599211c8504409021bbac") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("617596a31c8504409021bbaa")});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617599681c8504409021bbad") }
}
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{"_id":"617598d81c8504409021bbab"}
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617598d81c8504409021bbab") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("617596a31c8504409021bbab") });
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617599211c8504409021bbac") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("617596a31c8504409021bbab") });
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617599681c8504409021bbad") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("617596a31c8504409021bbab") });
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("617599d11c8504409021bbae") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.insert({ username:"ScumbagSteve", comment:"Denied your PR cause I found hack", post:ObjectId("617596a31c8504409021bbab") });
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("61759a3d1c8504409021bbaf") }
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> 
```

Insertions:-

```
db.createCollection("users");
```

```
db.users.insertMany([{"username":"GoodGuyGreg","first_name":"GoodGuy","last_name":"Greg"}, {"username":"ScumbagSteve","full_name":{"first":"Scumbag","last":"Steve"}}]);
```

```
db.createCollection("posts");
```

```
db.posts.insertMany([{"username":"GoodGuyGrey","title":"Passes out at party","body":"Wakes up early and cleans house"}, {"username":"GoodGuyGrey","title":"Steals your identity","body":"Raises your credit score"}, {"username":"GoodGuyGrey","title":"Reports a bug in your code","body":"Sends you a Pull Request"}, {"username":"ScumbagSteve","title":"Borrows something","body":"Sells it"}, {"username":"ScumbagSteve","title":"Borrows everything","body":"The end"}, {"username":"ScumbagSteve","title":"Forks your repo on github","body":"Sets to private"}]);
```

```
db.createCollection("comments");
```

```
db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("617596a31c8504409021bba8")});
```

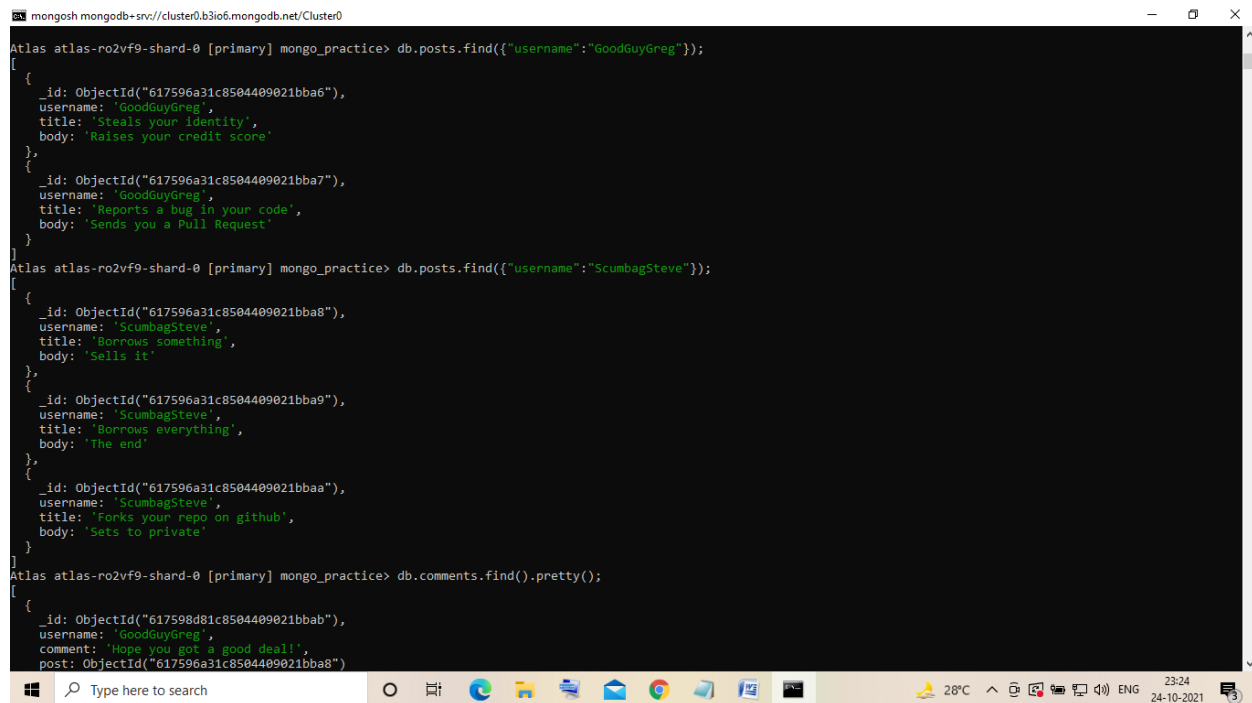
```
db.comments.insert({ username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("617596a31c8504409021bba9")});
```

```
db.comments.insert({ username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!",
post:ObjectId("617596a31c8504409021bbaa")}));
```

```
db.comments.insert({ "username":"ScumbagSteve"," comment":"It still isn't clean",
post:ObjectId("617596a31c8504409021bba5")}));
```

```
db.comments.insert({ "username":"ScumbagSteve"," comment":"Denied your PR cause I found hack",
"post":ObjectId("617596a31c8504409021bba7")}));
```

Querying related collections:-



```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.posts.find({"username":"GoodGuyGreg"});
[
  {
    _id: ObjectId("617596a31c8504409021bba6"),
    username: 'GoodGuyGreg',
    title: 'Steals your identity',
    body: 'Raises your credit score'
  },
  {
    _id: ObjectId("617596a31c8504409021bba7"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a Pull Request'
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.posts.find({"username":"ScumbagSteve"});
[
  {
    _id: ObjectId("617596a31c8504409021bba8"),
    username: 'ScumbagSteve',
    title: 'Borrows something',
    body: 'Sells it'
  },
  {
    _id: ObjectId("617596a31c8504409021bba9"),
    username: 'ScumbagSteve',
    title: 'Borrows everything',
    body: 'The end'
  },
  {
    _id: ObjectId("617596a31c8504409021bbaa"),
    username: 'ScumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private'
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find().pretty();
[
  {
    _id: ObjectId("617598d81c8504409021bbab"),
    username: 'GoodGuyGreg',
    comment: 'Hope you got a good deal!',
    post: ObjectId("617596a31c8504409021bba8")
  }
]
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{
  comment: 'Hope you got a good deal!',
  post: ObjectId("617596a31c8504409021bba8")
},
{
  _id: ObjectId("617599211c8504409021bbac"),
  username: 'GoodGuyGreg',
  comment: "What's mine is yours!",
  post: ObjectId("617596a31c8504409021bba9")
},
{
  _id: ObjectId("617599681c8504409021bbad"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("617596a31c8504409021bbaa")
},
{
  _id: ObjectId("617599d11c8504409021bbae"),
  username: 'ScumbagSteve',
  comment: "It still isn't clean",
  post: ObjectId("617596a31c8504409021bba5")
},
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find({"username":"GoodGuyGreg"});
{
  _id: ObjectId("617598d81c8504409021bbab"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("617596a31c8504409021bba8")
},
{
  _id: ObjectId("617599211c8504409021bbac"),
  username: 'GoodGuyGreg',
  comment: "What's mine is yours!",
  post: ObjectId("617596a31c8504409021bba9")
},
{
  _id: ObjectId("617599681c8504409021bbad"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("617596a31c8504409021bbaa")
},
{
  _id: ObjectId("617599d11c8504409021bbae"),
  username: 'ScumbagSteve',
  comment: "It still isn't clean",
  post: ObjectId("617596a31c8504409021bba5")
},
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
}
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find({"username":"GoodGuyGreg"});
{
  _id: ObjectId("617598d81c8504409021bbab"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("617596a31c8504409021bba8")
},
{
  _id: ObjectId("617599211c8504409021bbac"),
  username: 'GoodGuyGreg',
  comment: "What's mine is yours!",
  post: ObjectId("617596a31c8504409021bba9")
},
{
  _id: ObjectId("617599681c8504409021bbad"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("617596a31c8504409021bbaa")
},
{
  _id: ObjectId("617599d11c8504409021bbae"),
  username: 'ScumbagSteve',
  comment: "It still isn't clean",
  post: ObjectId("617596a31c8504409021bba5")
},
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find({"username":"ScumbagSteve"});
{
  _id: ObjectId("617599d11c8504409021bbae"),
  username: 'ScumbagSteve',
  comment: "It still isn't clean",
  post: ObjectId("617596a31c8504409021bba5")
},
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find({"post":ObjectId("617596a31c8504409021bba7")});
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
}
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find({"post":ObjectId("617596a31c8504409021bba7")});
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
}
```

```
mongosh mongodb+srv://cluster0.b3io6.mongodb.net/Cluster0
{
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("617596a31c8504409021bbaa")
},
{
  _id: ObjectId("617599d11c8504409021bbae"),
  username: 'ScumbagSteve',
  comment: 'It still isn't clean',
  post: ObjectId("617596a31c8504409021bba5")
},
{
  _id: ObjectId("61759a3d1c8504409021bbaf"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found hack',
  post: ObjectId("617596a31c8504409021bba7")
}
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.comments.find({post: db.posts.findOne({title: "Reports a bug in your code"})._id});
[
  {
    _id: ObjectId("61759a3d1c8504409021bbaf"),
    username: 'ScumbagSteve',
    comment: 'Denied your PR cause I found hack',
    post: ObjectId("617596a31c8504409021bba7")
  }
]
Atlas atlas-ro2vf9-shard-0 [primary] mongo_practice> db.posts.aggregate([{$match:{"title":"Reports a bug in your code"}},{$lookup:{from:"comments",localField:"_id",foreignField:"post",as:"comments"}}]).pretty();
[
  {
    _id: ObjectId("617596a31c8504409021bba7"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a Pull Request',
    comments: [
      {
        _id: ObjectId("61759a3d1c8504409021bbaf"),
        username: 'ScumbagSteve',
        comment: 'Denied your PR cause I found hack',
        post: ObjectId("617596a31c8504409021bba7")
      }
    ]
  }
]
]
```

To find all users:-

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

To find all posts:-

```
db.posts.find().pretty();
```

To find all posts that was authored by "GoodGuyGreg":-

```
db.posts.find("username":"GoodGuyGreg");
```

To find all posts that was authored by "ScumbagSteve":-

```
db.posts.find({"username":"ScumbagSteve"});
```

To find all comments:-

```
db.comments.find().pretty();
```

To find all comments that was authored by "GoodGuyGreg":-

```
db.comments.find({"username":"GoodGuyGreg"});
```

To find all comments that was authored by "ScumbagSteve":-

```
db.comments.find({"username":"ScumbagSteve"});
```

To find all comments belonging to the post "Reports a bug in your code":-

```
db.comments.find({post: db.posts.findOne({title: "Reports a bug in your code"})._id});
```

```
db.comments.find({"post":ObjectId("617596a31c8504409021bba7")});
```

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

