

Query/Find Documents

1. Get All Documents

```
mongosh mongodb+srv://workcluster.pclcu.mongodb.net/myFirstDatabase
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>mongosh "mongodb+srv://workcluster.pclcu.mongodb.net/myFirstDatabase" --username Manivenkat
Enter password: *****
Current Mongosh Log ID: 61eec67666457095dd883600
Connecting to:      mongodb+srv://workcluster.pclcu.mongodb.net/myFirstDatabase?appName=mongosh+1.1.9
Using MongoDB:      4.4.11
Using Mongosh:      1.1.9

For mongosh info see: https://docs.mongodb.com/mongosh-shell/
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.insertMany([{{title:'Fight Club',writer:'Quentin Tarantino',year:1999,actors:['Brad pitt','Edward Norton']},{title:'pulp Fiction',writer:'Quentin Tarantino',year:1994,actors:['John Travolta','Uma Thurman']},{title:'Inglorious basterds',writer:'Quentin Tarantino',year:2009,actors:['Brad Pitt','Diane kruger','Eli Roth']},{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'}},{title:'The Battle of the Five Armies',writer:'J.R.R.Tolkein',year:2012,Franchise:'The Hobbit',Synopsis:'Bilobo and Company are forced to engage in a war against an array of combatants and leep the Lonely Mountain from falling into the hands of raising'}},{title:'Pee Wee Hermans Big Adventure'}},{title:'Avatar'}],])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("61eed98d33095f2fbffcd45"),
    '1': ObjectId("61eed98d33095f2fbffcd46"),
    '2': ObjectId("61eed98d33095f2fbffcd47"),
    '3': ObjectId("61eed98d33095f2fbffcd48"),
    '4': ObjectId("61eed98d33095f2fbffcd49"),
    '5': ObjectId("61eed98d33095f2fbffcd4a"),
    '6': ObjectId("61eed98d33095f2fbffcd4b"),
    '7': ObjectId("61eed98d33095f2fbffcd4c")
  }
}
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

```
mongosh mongodb+srv://workcluster.pclcu.mongodb.net/myFirstDatabase
{
  _id: ObjectId("61eed98d33095f2fbffcd45"),
  title: 'Fight Club',
  writer: 'Quentin Tarantino',
  year: 1999,
  actors: [ 'Brad pitt', 'Edward Norton' ]
},
{
  _id: ObjectId("61eed98d33095f2fbffcd46"),
  title: 'pulp Fiction',
  writer: 'Quentin Tarantino',
  year: 1994,
  actors: [ 'John Travolta', 'Uma Thurman' ]
},
{
  _id: ObjectId("61eed98d33095f2fbffcd47"),
  title: 'Inglorious basterds',
  writer: 'Quentin Tarantino',
  year: 2009,
  actors: [ 'Brad Pitt', 'Diane kruger', 'Eli Roth' ]
},
{
  _id: ObjectId("61eed98d33095f2fbffcd48"),
  title: 'The Hobbit: An unexpected journey',
  writer: 'J.R.R.Tolkein',
  year: 2012,
  Franchise: 'The Hobbit'
},
{
  _id: ObjectId("61eed98d33095f2fbffcd49"),
  title: 'The hobbit: The Desolation of smaug',
  writer: 'J.R.R.Tolkein',
  year: 2013,
  Franchise: 'The Hobbit'
},
{
  _id: ObjectId("61eed98d33095f2fbffcd4a"),
  title: 'The Battle of the Five Armies',
  writer: 'J.R.R.Tolkein',
  year: 2012,
  Franchise: 'The Hobbit',
  Synopsis: 'Bilobo and Company are forced to engage in a war against an array of combatants and leep the Lonely Mountain from falling into the hands of raising'
},
{
  _id: ObjectId("61eed98d33095f2fbffcd4b"),
  title: 'Pee Wee Hermans Big Adventure'
},
{
  _id: ObjectId("61eed98d33095f2fbffcd4c"), title: 'Avatar' }
}
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

2. Get All Documents with writer set to “Quentin Tarantino”

```
mongosh mongodb+srv://workcluster.pclcu.mongodb.net/myFirstDatabase
[
  {
    _id: ObjectId("61eed98d33095f2fbffcd45"),
    title: 'Fight Club',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: [ 'Brad pitt', 'Edward Norton' ]
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd46"),
    title: 'pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd47"),
    title: 'Inglorious basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane kruger', 'Eli Roth' ]
  }
]
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> _
```

3. Get all Documents Where actors include “Brad Pitt”

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({actors:'Brad Pitt'})
[
  {
    _id: ObjectId("61eed98d33095f2fbffcd45"),
    title: 'Fight Club',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: 'Brad Pitt'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd47"),
    title: 'Inglorious basterds',
    writer: 'Quentin Tarantino',
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane kruger', 'Eli Roth' ]
  }
]
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> _
```

4. Get all documents with franchise set to “The Hobbit”

```
mongosh mongodb+srv://workcluster.pclcu.mongodb.net/myFirstDatabase
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({franchise:'The Hobbit'})
[
  {
    _id: ObjectId("61eed98d33095f2fbffcd48"),
    title: 'The Hobbit: An unexpected journey',
    writer: 'J.R.R.Tolkien',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd49"),
    title: 'The Hobbit: The Desolation of smaug',
    writer: 'J.R.R.Tolkien',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd4a"),
    title: '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 leap the Lonely Mountain from falling into the hands of raising'
  }
]
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> _
```

5. Get all movies released in the 90s

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({year:{$gt:1990,$lt:2000}})
[
  {
    _id: ObjectId("61eed98d33095f2fbffcd45"),
    title: 'Fight Club',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: 'Brad Pitt'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd46"),
    title: 'pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  }
]
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

6. Get all the movies released before the year 2000 or after 2010

```
mongosh mongodb+srv://workcluster.pclou.mongodb.net/myFirstDatabase
actors: 'Brad Pitt'
},
{
  _id: ObjectId("61eed98d33095f2fbffcd46"),
  title: 'pulp Fiction',
  writers: 'Quentin Tarantino',
  year: 1994,
  actors: [ 'John Travolta', 'Uma Thurman' ]
}
]
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$or:[{year:{$lt:2000}},{year:{$gt:2010}]}})
[
  {
    _id: ObjectId("61eed98d33095f2fbffcd45"),
    title: 'Fight Club',
    writer: 'Quentin Tarantino',
    year: 1999,
    actors: 'Brad Pitt'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd46"),
    title: 'pulp Fiction',
    writer: 'Quentin Tarantino',
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd48"),
    title: 'The Hobbit: An unexpected journey',
    writer: 'J.R.R.Tolkien',
    year: 2012,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd49"),
    title: 'The hobbit: The Desolation of smaug',
    writer: 'J.R.R.Tolkien',
    year: 2013,
    franchise: 'The Hobbit'
  },
  {
    _id: ObjectId("61eed98d33095f2fbffcd4a"),
    title: '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 leap the Lonely Mountain from falling into the hands of raising'
  }
]
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

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

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.update({title:'The Hobbit: An unexpected journey'},{$set:{synopsis:'A reluctant hobbit,sets out to the Lonely Mountain with a spirited group of d
arves to reclaim their mountain home - and the gold within it - from the dragon smaug'}},{upsert:false,multi:true})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

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."

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> 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"}},{upsert:false,multi:true})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

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

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.update({title:"pulp Fiction"},{$push:{actors:"Samuel L. Jackson"}},{upsert:false,multi:true})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

Text Search

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

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.createIndex({synopsis:'text'})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.createIndex({synopsis:'text'})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.getIndexes()
{ v: 2, key: { _id: 1, name: 'id' },
  v: 2,
  key: { _fts: 'text', _ftsx: 1 },
  name: 'synopsis_text',
  weights: { synopsis: 1 },
  default_language: 'english',
  language_override: 'language',
  textIndexVersion: 1 }
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$text:{$search:"Bilbo"}})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$text:{$search:"Bilbo"}})
MongoError: Unexpected token, expected "," (1:30)
    at db.movies.find({$text:{$search:"Bilbo"}})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$text:{$search:"Bilbo"}})
MongoError: Unexpected token, expected "," (1:30)
    at db.movies.find({$text:{$search:"Bilbo"}})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$text:{$search:"Bilbo"}})
{
  _id: ObjectId("61eed98d33095f2fbffcdad49"),
  title: "The Hobbit: The Desolation of Smaug",
  writer: "J.R.R. Tolkien",
  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"
}
```

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

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$text:{$search:"Gandalf"}})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
{
  _id: ObjectId("61eed98d33095f2fbffcdad49"),
  title: "The Hobbit: The Desolation of Smaug",
  writer: "J.R.R. Tolkien",
  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"
}
```

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

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.find({$text:{$search:"Gandalf"}})
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
{
  _id: ObjectId("61eed98d33095f2fbffcdad49"),
  title: "The Hobbit: The Desolation of Smaug",
  writer: "J.R.R. Tolkien",
  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"
}
```

4. Find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

```
{
  "_id": ObjectId("61ed0318914041b0d2ed5fcd"),
  "title": "The Hobbit: An Unexpected Journey",
  "writer": "J.R.R. Tolkien",
  "year": 2012,
  "Franchise": "The Hobbit",
  "Synopsis": "A reluctant hobbit, 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("61ed0318914041b0d2ed5fce"),
  "title": "The Hobbit: The Desolation of Smaug",
  "writer": "J.R.R. Tolkien",
  "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"
```

5. Movies that have a synopsis that contains the word "gold" and "dragon"

```
{
  "_id": ObjectId("61ed0318914041b0d2ed5fcd"),
  "title": "The Hobbit: An Unexpected Journey",
  "writer": "J.R.R. Tolkien",
  "year": 2012,
  "Franchise": "The Hobbit",
  "Synopsis": "A reluctant hobbit, 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"
}
```

Delete Documents

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

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.remove({title:"pee wee Hermans Big Adventur"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 0 }
```

2. delete the movie "Avatar"

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> db.movies.remove({title:"Avatar"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase>
```

Querying related collections

Users' collection:

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> 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:"sell 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("61eef3e833095f2fbffcd4d"),
    '1': ObjectId("61eef3e833095f2fbffcd4e"),
    '2': ObjectId("61eef3e833095f2fbffcd4f"),
    '3': ObjectId("61eef3e833095f2fbffcd50"),
    '4': ObjectId("61eef3e833095f2fbffcd51"),
    '5': ObjectId("61eef3e833095f2fbffcd52")
  }
}
```

Posts collection:

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> 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:"sell 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("61eef3e833095f2fbffcd4d"),
    '1': ObjectId("61eef3e833095f2fbffcd4e"),
    '2': ObjectId("61eef3e833095f2fbffcd4f"),
    '3': ObjectId("61eef3e833095f2fbffcd50"),
    '4': ObjectId("61eef3e833095f2fbffcd51"),
    '5': ObjectId("61eef3e833095f2fbffcd52")
  }
}
```

Comments collection:

```
Atlas atlas-zoofki-shard-0 [primary] myFirstDatabase> 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:"sell 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("61eef3e833095f2fbffcd4d"),
    '1': ObjectId("61eef3e833095f2fbffcd4de"),
    '2': ObjectId("61eef3e833095f2fbffcd4f"),
    '3': ObjectId("61eef3e833095f2fbffcd50"),
    '4': ObjectId("61eef3e833095f2fbffcd51"),
    '5': ObjectId("61eef3e833095f2fbffcd52")
  }
}
```

1. find all users

```
{
  {
    _id: ObjectId("61ed511d914041b0d2ed5fd2"),
    username: 'GoodGuyGreg',
    first_name: 'Good Guy',
    last_name: 'Greg'
  },
  {
    _id: ObjectId("61ed511d914041b0d2ed5fd3"),
    username: 'ScumbagSteve',
    full_name: { first: 'Scumbag', last: 'Steve' }
  }
}
```

2. find all posts

```
Atlas atlas-v5kpui-shard-0 [primary] TestDB> db.posts.find()
```

```
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd4"),
  username: 'GoodGuyGreg',
  title: 'Passes out at party',
  body: 'Wakes up early and cleans house'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd5"),
  username: 'GoodGuyGreg',
  title: 'Steals your identity',
  body: 'Raises your credit score'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd6"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
  body: 'Sends you a Pull Request'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd7"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd8"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd9"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
}
```

3. find all posts that was authored by "GoodGuyGreg"

```

{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdc"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("61ed54a3914041b0d2ed5fd7")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdd"),
  username: 'GoodGuyGreg',
  comment: "What's mine is yours!",
  post: ObjectId("61ed54a3914041b0d2ed5fd8")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fde"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("61ed54a3914041b0d2ed5fd9")
}

```

4. find all posts that was authored by "ScumbagSteve"

```

{
  _id: ObjectId("61ed54a3914041b0d2ed5fd7"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd8"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd9"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
}

```

5. find all comments


```

{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdc"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("61ed54a3914041b0d2ed5fd7")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdd"),
  username: 'GoodGuyGreg',
  comment: "What's mine is yours!",
  post: ObjectId("61ed54a3914041b0d2ed5fd8")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fde"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("61ed54a3914041b0d2ed5fd9")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdf"),
  username: 'ScumbagSteve',
  comment: "It still isn't clean",
  post: ObjectId("61ed54a3914041b0d2ed5fd4")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fe0"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("61ed54a3914041b0d2ed5fd6")
}
}

```

6. find all comments that was authored by "GoodGuyGreg"

```

{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdc"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("61ed54a3914041b0d2ed5fd7")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fdd"),
  username: 'GoodGuyGreg',
  comment: "What's mine is yours!",
  post: ObjectId("61ed54a3914041b0d2ed5fd8")
},
{
  _id: ObjectId("61ed5d4e914041b0d2ed5fde"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("61ed54a3914041b0d2ed5fd9")
}
}

```

7. find all comments that was authored by "ScumbagSteve"

```

{
  _id: ObjectId("61ed54a3914041b0d2ed5fd7"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd8"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end'
},
{
  _id: ObjectId("61ed54a3914041b0d2ed5fd9"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private'
}

```

8. find all comments belonging to the post "Reports a bug in your code"

```

{
  _id: ObjectId("61ed5d4e914041b0d2ed5fe0"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("61ed54a3914041b0d2ed5fd6")
}
]

```