

```
mongosh mongodbsrv://<credentials>@cluster0.bawoc.mongodb.net/myFirstDatabase
Current MongoDB ID: 6284de25471f51c94215d07
Connecting to: mongodb+srv://<credentials>@cluster0.bawoc.mongodb.net/myFirstDatabase?appName=mongosh+1.4.1
Using Mongosh: 5.0.8 (API Version 1)
Using Mongosh: 1.4.1

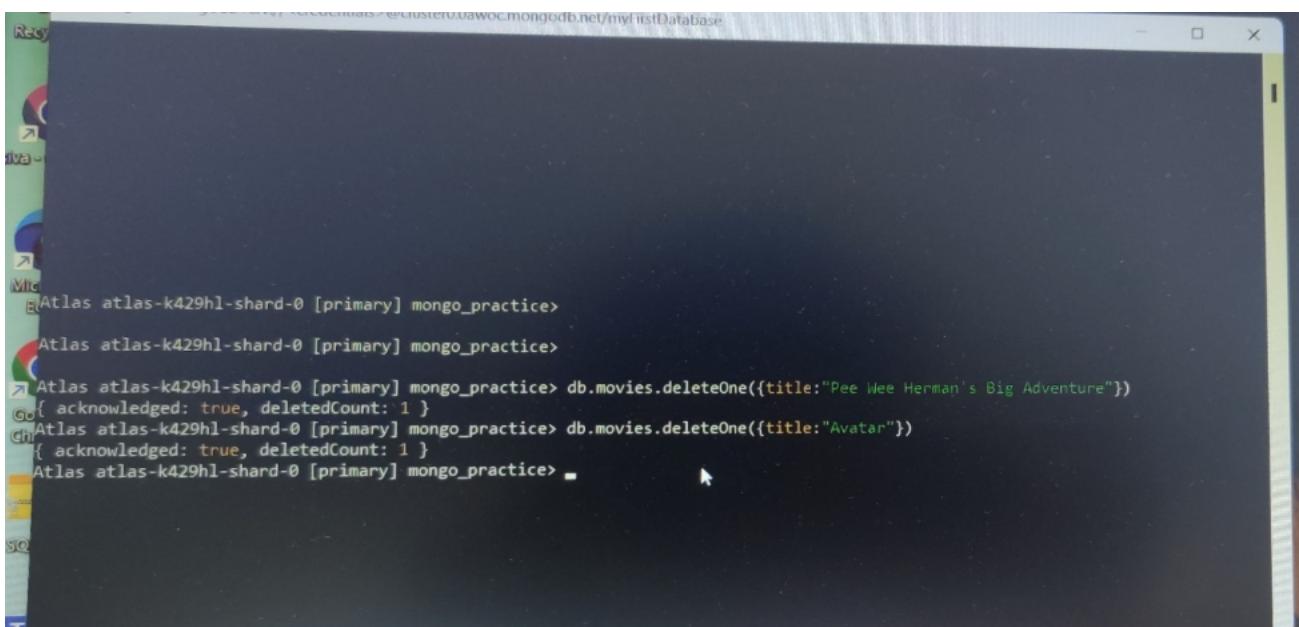
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-kd29hl-shard-0 [primary] myFirstDatabase> use mongo_practice
switched to db mongo_practice
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice> db.insertOne({title:"Fight club",writer:"Chuck Palahniuk",year:1999,actors:[{"Brad Pitt","Edward Norton"]})
MongoError: db.insertOne is not a function
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice> db.movies.insertOne({title:"Fight Club",writer:"Chuck Palahniuk",year:1999,actors:[{"Brad Pitt","Edward Norton"]})
{
  acknowledged: true,
  insertedId: ObjectId("6284e17b7db0438786c07727")
}
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice> db.movies.insertOne({title:"Pulp Fiction",writer:"Quentin Tarantino",year:1994,actors:[{"John Travolta","Uma Thurman"]})
{
  acknowledged: true,
  insertedId: ObjectId("6284e25f7db0438786c07728")
}
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice> db.movies.insertOne({title:"Inglourious Basterds",writer:"Quentin Tarantino",year:2009,actors:[{"Brad Pitt","Diane Kruger","Eli Roth"]})
{
  acknowledged: true,
  insertedId: ObjectId("6284e3247db0438786c07729")
}
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice> db.movies.insertMany([
  {
    title: "The Hobbit: An Unexpected Journey",
    writer: "J.R.R. Tolkien",
    year: 2013,
    franchise: "The Hobbit"
  },
  {
    title: "The Hobbit: The Desolation of Smaug",
    writer: "J.R.R. Tolkien",
    year: 2013,
    franchise: "The Hobbit"
  },
  {
    title: "The Hobbit: The Return of the King",
    writer: "J.R.R. Tolkien",
    year: 2014,
    franchise: "The Hobbit"
  }
])
MongoError: synopsis is required for each document in insertMany()
... synopsis:"Bilbo and company are forced to engage in a war against an array of combatants and keep the lovely mountain from falling into the hands of rising darkness."
...
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice>
```

```
mongosh mongoDB+srv://<credentials>@cluster0.basisc.mongodb.net/myFirstDatabase
... 
... 
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
... title: "The Tree of Mormon's Big Adventure"
... },
... {
... title: "Mystic"
... }
...
{
  acknowledged: true,
  insertedId: [
    0: ObjectId("6284edcb7db8438786c0712a"),
    1: ObjectId("6284edcb7db8438786c0712b"),
    2: ObjectId("6284edcb7db8438786c0712c"),
    3: ObjectId("6284edcb7db8438786c0712d"),
    4: ObjectId("6284edcb7db8438786c0712e")
  ]
}
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({writer: "Quentin Tarantino"})
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({writer: "Quentin tarantino"})
[
  {
    _id: ObjectId("6284e25f7db8438786c0712f"),
    title: "Pulp Fiction",
    writer: "Quentin tarantino",
    year: 1994,
    actors: [ 'John Travolta', 'Uma Thurman' ]
  },
  {
    _id: ObjectId("6284e3247db8438786c0712g"),
    title: "Inglourious Basterds",
    writer: "Quentin tarantino",
    year: 2009,
    actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({franchise: "The Hobbit"})
[
  {
    _id: ObjectId("6284e6cb7db8438786c0712a"),
    title: "The Hobbit: An Unexpected Journey",
    writer: 'J.R.R Tolkiens',
    year: 2012,
    franchise: "The Hobbit"
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
```

```
mongosh mongo://vcf/credentials@cluster0:27017/mongodb.net/myFirstDatabase
{
  "_id": ObjectId("6284e6cb7db8438786c0712b"),
  "title": "The Hobbit: The Desolation of Smaug",
  "writer": "J.R.R. Tolkein",
  "year": 2013,
  "franchise": "The Hobbit"
},
{
  "_id": ObjectId("6284e6cb7db8438786c0712c"),
  "title": "The Battle of Five Armies",
  "writer": "J.R.R. Tolkein",
  "year": 2014,
  "franchise": "The Hobbit",
  "synopsis": "Bilbo and company are forced to engage in a war against an array of combatants and keep the lovely mountain from falling into the hands of rising darkness."
}
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({year:{$lt:2000}})
[
  {
    "_id": ObjectId("6284e1767db8438786c07127"),
    "title": "Fight Club",
    "writer": "Chuck Palahniuk",
    "year": 1999,
    "actors": ["Brad Pitt", "Edward Norton"]
  },
  {
    "_id": ObjectId("6284e1767db8438786c07128"),
    "title": "Pulp Fiction",
    "writer": "Quentin Tarantino",
    "year": 1994,
    "actors": ["Uma Thurman", "John Travolta"]
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.update({
  ...
  "title": "The Hobbit: An Unexpected Journey",
  ...
  "$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."}
  ...
}, {
  "DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite."
  ...
})
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.updateOne({
  ...
  "title": "The Hobbit: The Desolation of Smaug",
  ...
  "$set": {"synopsis": "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious magical ring."}
})
29°C Cloudy
```

```
atlas:atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.updateOne({  
  ...  
  "title": "The Hobbit: The Desolation of Smaug",  
  ...  
  {"$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious magical ring."}}  
}  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.updateOne({  
  ...  
  "title": "Pulp Fiction",  
  ...  
  {"$set:{actors:"samuel L.Jackson"}}  
}  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
```

A screenshot of a Windows desktop environment. In the center is a terminal window titled "MongoDB Shell" with the URL "atlas-k429hl-shard-0.mongolab.com:27017/mongo_practice". The terminal window displays the following MongoDB session:

```
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Pee Wee Herman's Big Adventure"})
{
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.deleteOne({title:"Avatar"})
{
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
```

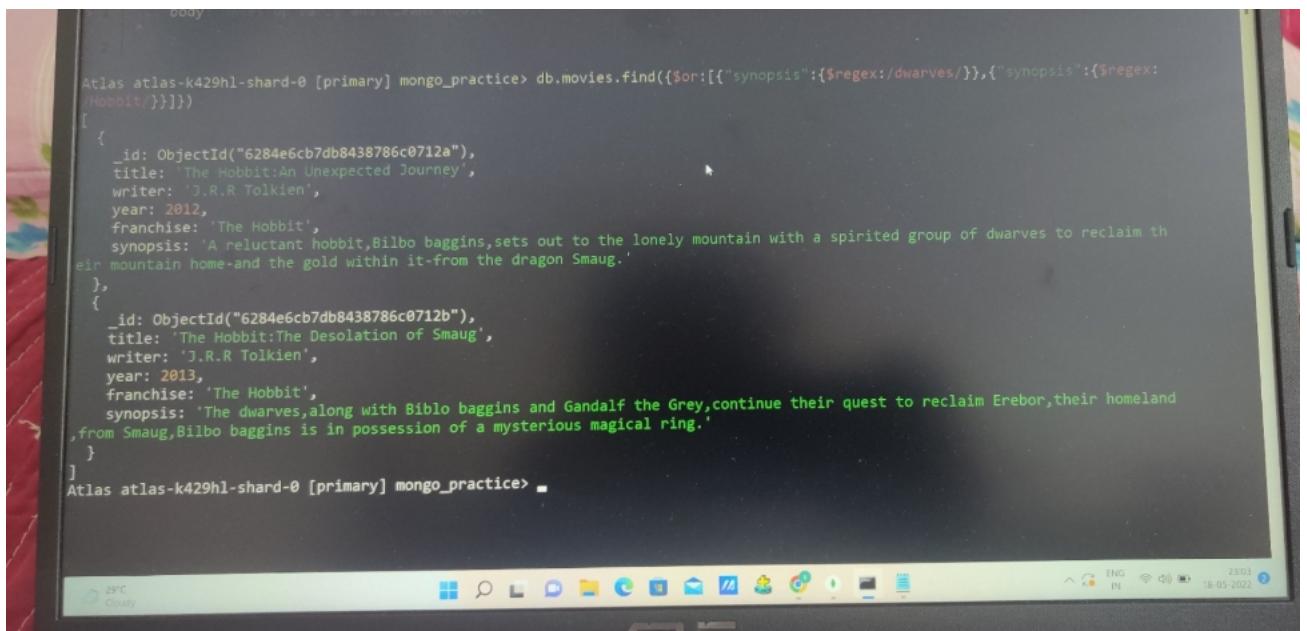
The desktop background is black, and there are several icons on the taskbar, including Recycle Bin, File Explorer, and Task View.

```
mongosh mongodb+srv://<credentials>@cluster0.0awoc.mongodb.net/myFirstDatabase
Atlas atlas-k429h1-shard-0 [primary] mongo_practice> db.movies.find({ "name": "/Bilbo" })
[
  {
    "_id": ObjectId("6284e6cb7db8438786c0712a"),
    "title": "The Hobbit: An Unexpected Journey",
    "writer": "J.R.R. Tolkien",
    "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("6284e6cb7db8438786c0712b"),
    "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 magical ring."
  },
  {
    "_id": ObjectId("6284e6cb7db8438786c0712c"),
    "title": "The Hobbit: The Desolation of Smaug",
    "writer": "J.R.R. Tolkien",
    "year": 2014,
    "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 rising darkness."
  }
]
```

```
body

Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({$or:[{"synopsis":{$regex:/dwarves/}}, {"synopsis":{$regex:/hobbit/}}]})

[
  {
    _id: ObjectId("6284e6cb7db8438786c0712a"),
    title: 'The Hobbit:An Unexpected Journey',
    writer: 'J.R.R Tolkien',
    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 th
eir mountain home-and the gold within it-from the dragon Smaug.'
  },
  {
    _id: ObjectId("6284e6cb7db8438786c0712b"),
    title: 'The Hobbit:The Desolation of Smaug',
    writer: 'J.R.R Tolkien',
    year: 2013,
    franchise: 'The Hobbit',
    synopsis: 'The dwarves,along with Biblo baggins and Gandalf the Grey,continue their quest to reclaim Erebor,their homeland
from Smaug,Bilbo baggins is in possession of a mysterious magical ring.'
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
```



```
writer: 'J.R.R Tolkien',
year: 2013,
franchise: 'The Hobbit',
synopsis: 'The dwarves,along with Biblo baggins and Gandalf the Grey,continue their quest to reclaim Erebor,their homeland,from Smaug,Bilbo baggins is in possession of a mysterious magical ring.'
}

atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({$and:[{"synopsis":{$regex:/gold/}}, {"synopsis":{$regex:/dragon/}}]})

{
  _id: ObjectId("6284e6cb7db8438786c0712a"),
  title: 'The Hobbit:An Unexpected Journey',
  writer: 'J.R.R Tolkien',
  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 mount in home-and the gold within it-from the dragon Smaug.'
}
atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({"synopsis":/Gandalf/})

{
  _id: ObjectId("6284e6cb7db8438786c0712b"),
  title: 'The Hobbit:The Desolation of Smaug',
  writer: 'J.R.R Tolkien',
  year: 2013,
  franchise: 'The Hobbit',
  synopsis: 'The dwarves,along with Biblo baggins and Gandalf the Grey,continue their quest to reclaim Erebor,their homeland,from Smaug,Bilbo baggins is in possession of a mysterious magical ring.'
}

29°C
Cloudy
```

ENG IN 23:11 18-05-2022

ASUS

```
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({$and:[{Synopsis:/Bilbo/},{Synopsis:$not:/Gandalf/}]})  
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({$and:[{synopsis:/Bilbo/g},{synopsis:$not:/Gandalf/g}]})  
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.movies.find({$and:[{synopsis:/Bilbo/},{synopsis:$not:/Gandalf/}]})  
[  
 {  
   _id: ObjectId("6284e6cb7db8438786c0712a"),  
   title: 'The Hobbit:An Unexpected Journey',  
   writer: 'J.R.R Tolkien',  
   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("6284e6cb7db8438786c0712c"),  
   title: 'The Desolation of Smaug',  
   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 lovely mountain from falling into the hands of rising darkness.'  
}
```

```
mongosh mongoDB:srcff<credentials>@cluster0.0woc.mongodb.net/myFirstDatabase
)
}
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.posts.insertMany([
... {
...   username: "GoodGuyGreg",
...   title: "Raises 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: "SucumbugSteve",
...   title: "Borrows something",
...   body: "Sells it"
... },
... {
...   username: "SucumbugSteve",
...   title: "Borrows everything",
...   body: "The end"
... },
... {
...   username: "SucumbugSteve",
...   title: "Forks your repo on github",
...   body: "Sets to private"
... }
... ])
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId("6285f8cb0a538bb8d510c6b"),
    '1': ObjectId("6285f8cb0a538bb8d510c6c"),
    '2': ObjectId("6285f8cb0a538bb8d510c6d"),
    '3': ObjectId("6285f8cb0a538bb8d510c6f"),
    '4': ObjectId("6285f8cb0a538bb8d510c70"),
    '5': ObjectId("6285f8cb0a538bb8d510c70")
  ]
}
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> =
```

```
mongosh mongodb+srv://<credentials>@cluster0.0awoc.mongodb.net/myFirstDatabase
    : ObjectId("6285f8cb0a5385b85d510c70")
}
db}
usAtlas atlas-k429hl-shard-0 [primary] mongo_practice>
ti
boAtlas atlas-k429hl-shard-0 [primary] mongo_practice> db.users.insertMany([
}, {
    username:"GoodGuyGreg",
    first_name:"Good Guy",
    last_name:"Greg"
}, {
    username:"SucumbugSteve",
    full_name:{
        fist:"Sucumbug",
        last:"Steve"
    }
}, {
    acknowledged: true,
    insertedIds: {
        '0': ObjectId("6285fb620a5385b85d510c71"),
        '1': ObjectId("6285fb620a5385b85d510c72")
    }
}
us}
tiAtlas atlas-k429hl-shard-0 [primary] mongo_practice>
bo
},
{
    username:"SucumbugSteve",
    full_name:{
```

```
mongosh mongoDB+srv://<credentials>@cluster0.owoc.mongodb.net/myFirstDatabase
atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.comments.insertMany([
...   {
...     username: "GoodGuyGreg",
...     comment: "How you got a good devil",
...     post:ObjectId("6285fb620e5385b05d510c6d")
...   },
...   {
...     username: "GoodGuyGreg",
...     comment: "What wine is yours",
...     post:ObjectId("6285fb620e5385b05d510c6d")
...   },
...   {
...     username: "GoodGuyGreg",
...     comment: "Don't violate the licensing agreement",
...     post:ObjectId("6285fb620e5385b05d510c70")
...   },
...   {
...     username: "SucumbugSteve",
...     comment: "It still isn't clean",
...     post:ObjectId("6285fb620e5385b05d510c6d")
...   },
...   {
...     username: "SucumbugSteve",
...     comment: "Denied your PR cause I found a hack",
...     post: ObjectId("6285fb620e5385b05d510c6d")
...   }
... ],
{
  acknowledged: true,
  insertedIds: [
    {
      '_id': ObjectId("6285fb620e5385b05d510c73"),
      '_t': ObjectId("6285fb620e5385b05d510c74"),
      '_v': ObjectId("6285fb620e5385b05d510c75"),
      '_r': ObjectId("6285fb620e5385b05d510c76"),
      '_d': ObjectId("6285fb620e5385b05d510c77")
    }
  ]
}
atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.users.find()
[
  {
    '_id': ObjectId("6285fb620e5385b05d510c73"),
    'username': 'GoodGuyGreg',
    'first_name': 'Good Guy',
    'last_name': 'Greg'
  },
  {
    '_id': ObjectId("6285fb620e5385b05d510c72"),
    'username': 'SucumbugSteve',
    'full_name': { first: 'Sucumbug', last: 'Steve' }
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
```

```
mongodbsrv://cadenmas-Belkin0/DemoMongoDB/testDatabase
atlas atlas-k42m1-shard-0 [primary] mongo_practice> db.users.find()
{
  "_id": ObjectId("6285fb620a5385b85d510c71"),
  "username": "GoodGuyGreg",
  "first_name": "Good Guy",
  "last_name": "Greg"
},
{
  "_id": ObjectId("6285fb620a5385b85d510c72"),
  "username": "SuccumbSteve",
  "full_name": { first: "Succumb", last: "Steve" }
}
Atlas atlas-k42m1-shard-0 [primary] mongo_practice> db.posts.find()
{
  "_id": ObjectId("6285fb8cb0a5385b85d510c60"),
  "username": "GoodGuyGreg",
  "title": "Passes out at party",
  "body": "Wakes up early and cleans house"
},
{
  "_id": ObjectId("6285fb8cb0a5385b85d510c6c"),
  "username": "GoodGuyGreg",
  "title": "Steals your identity",
  "body": "Raises your credit score"
},
{
  "_id": ObjectId("6285fb8cb0a5385b85d510c6d"),
  "username": "GoodGuyGreg",
  "title": "Reports a bug in your code",
  "body": "Sends you a pull request"
},
{
  "_id": ObjectId("6285fb8cb0a5385b85d510c6e"),
  "username": "SuccumbSteve",
  "title": "Borrows something",
  "body": "Sells it"
},
{
  "_id": ObjectId("6285fb8cb0a5385b85d510c6f"),
  "username": "SuccumbSteve",
  "title": "Borrows everything",
  "body": "The end"
},
{
  "_id": ObjectId("6285fb8cb0a5385b85d510c70"),
  "username": "SuccumbSteve",
  "title": "Forks your repo on github",
  "body": "Sets to private"
}
```

```
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.posts.find({username:"GoodGuyGreg"})
[
  {
    _id: ObjectId("6285f8cb0a5385b85d510c6b"),
    username: 'GoodGuyGreg',
    title: 'Passes out at party',
    body: 'Wakes up early and cleans house'
  },
  {
    _id: ObjectId("6285f8cb0a5385b85d510c6c"),
    username: 'GoodGuyGreg',
    title: 'Steals your identity',
    body: 'Raises your credit score'
  },
  {
    _id: ObjectId("6285f8cb0a5385b85d510c6d"),
    username: 'GoodGuyGreg',
    title: 'Reports a bug in your code',
    body: 'Sends you a pull request'
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice> db.posts.find({username:"SucumbagSteve"})
[
  {
    _id: ObjectId("6285f8cb0a5385b85d510c6e"),
    username: 'SucumbagSteve',
    title: 'Borrows something',
    body: 'Sells it'
  },
  {
    _id: ObjectId("6285f8cb0a5385b85d510c6f"),
    username: 'SucumbagSteve',
    title: 'Borrows everything',
    body: 'The end'
  },
  {
    _id: ObjectId("6285f8cb0a5385b85d510c70"),
    username: 'SucumbagSteve',
    title: 'Forks your repo on github',
    body: 'Sets to private'
  }
]
Atlas atlas-k429hl-shard-0 [primary] mongo_practice>
```

```
Atlas atlas-kd29hl-shard-0 [primary] mongo_practice> db.comments.find()
{
  _id: ObjectId("628604170a5385b85d510c73"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("6285f8cbba5385b85d510c6e")
},
{
  _id: ObjectId("628604170a5385b85d510c74"),
  username: 'GoodGuyGreg',
  comment: 'What's mine is yours!',
  post: ObjectId("6285f8cbba5385b85d510c6f")
},
{
  _id: ObjectId("628604170a5385b85d510c75"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("6285f8cbba5385b85d510c79")
},
{
  _id: ObjectId("628604170a5385b85d510c76"),
  username: 'SucumbagSteve',
  comment: "It still isn't clean",
  post: ObjectId("6285f8cbba5385b85d510c6b")
},
{
  _id: ObjectId("628604170a5385b85d510c77"),
  username: 'SucumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("6285f8cbba5385b85d510c6d")
},
{
  _id: ObjectId("628604170a5385b85d510c74"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("6285f8cbba5385b85d510c6e")
},
{
  _id: ObjectId("628604170a5385b85d510c74"),
  username: 'GoodGuyGreg',
  comment: 'What's mine is yours!',
  post: ObjectId("6285f8cbba5385b85d510c6f")
},
{
  _id: ObjectId("628604170a5385b85d510c75"),
  username: 'GoodGuyGreg',
  comment: "Don't violate the licensing agreement!",
  post: ObjectId("6285f8cbba5385b85d510c79")
}
```

```
mongosh mongo@ip-172-31-10-1:~$ cd ~/Desktop/learncodings/db-test/mongo-practice
mongosh mongo@ip-172-31-10-1:~/Desktop/learncodings/db-test/mongo-practice> db.comments.find({username:"GoodGuyGreg"})
{
  "_id": ObjectId("628604170a5385b05d510c76"),
  "username": "GoodGuyGreg",
  "comment": "How you got a good deal!",
  "post": ObjectId("6285f8cb0a5385b05d510c6b")
},
{
  "_id": ObjectId("628604170a5385b05d510c77"),
  "username": "GoodGuyGreg",
  "comment": "Dated your PP cause I found a back",
  "post": ObjectId("6285f8cb0a5385b05d510c6d")
}
Atlas atlas-k429l-shard-0 [binary] mongo_practice> db.comments.find({username:"GoodGuyGreg"})
{
  "_id": ObjectId("628604170a5385b05d510c73"),
  "username": "GoodGuyGreg",
  "comment": "Have you got a good deal!",
  "post": ObjectId("6285f8cb0a5385b05d510c6e")
},
{
  "_id": ObjectId("628604170a5385b05d510c74"),
  "username": "GoodGuyGreg",
  "comment": "What mine is yours!!",
  "post": ObjectId("6285f8cb0a5385b05d510c6f")
},
{
  "_id": ObjectId("628604170a5385b05d510c75"),
  "username": "GoodGuyGreg",
  "comment": "Don't violate the licensing agreement",
  "post": ObjectId("6285f8cb0a5385b05d510c70")
}
Atlas atlas-k429l-shard-0 [primary] mongo_practice> db.comments.find({username:"SuccumbSteve"})
[
  {
    "_id": ObjectId("628604170a5385b05d510c76"),
    "username": "SuccumbSteve",
    "comment": "It still isn't clean",
    "post": ObjectId("6285f8cb0a5385b05d510c6b")
  },
  {
    "_id": ObjectId("628604170a5385b05d510c77"),
    "username": "SuccumbSteve",
    "comment": "Dated your PP cause I found a back",
    "post": ObjectId("6285f8cb0a5385b05d510c6d")
  }
]
```

```
mongosh mongodb+srv://<credentials>-@cluster0.0awox.mongodb.net/myFirstDatabase
Atlas atlas-k429h1-shard-0 [primary] mongo_practice> db.posts.aggregate([
... {
...   $match:{title:"Reports a bug in your code"}
... },
... {
...   $lookup:{
...     from:"comments",
...     foreignField:"post",
...     as:"comments"
...   }
... }
... ])
MongoServerError: unknown argument to $lookup: foreignfield
Atlas atlas-k429h1-shard-0 [primary] mongo_practice> [
... {
...   _id:ObjectId("6285f8cb0a5305b5d510c6d"),
...   username:"GoodGuyGreg",
...   title:"Reports a bug in your code",
...   body:"Sends you a pull request",
...   comments:[]
... }]
```

PDF Created Using



Camera Scanner

Easily Scan documents & Generate PDF



<https://play.google.com/store/apps/details?id=photo.pdf.maker>