Atlas atlas-y42dil-shard-0[primary] mongo\_practice> db.movies.insert({title:'Fight Club',writer:'Chuck Palahniuko',year:1999,actors:['Brad pitt','Edwaerd Norton']})

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

acknowledged: true,

insertedIds: { '0': ObjectId("617a92df28c0e36ac595494b") }

}

Atlasatlas-y42dil-shard-0[primary]mongo\_practice>db.movies.insert({title:'Plup Fiction',writer:'Quentin Tarantino',year:1994,actors:['Jhon Travolta','uma Thutman']})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617a947728c0e36ac595494c") }

}

Atlasatlas-y42dil-shard-0[primary]mongo\_practice>db.movies.insert({title:'Inglorious Basterds',writer:'Quentin Tarantino',year:2009,actors:['Brad Pitt','Diane Kruger','Eli Roth']})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617a991228c0e36ac595494d") }

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice>db.movies.insert({title:'The Hobbies:An Unexpected Journey',writer:'J.R.R.Tolkien',year:2012,franchise:'The Hobbit'})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617aa3e528c0e36ac595494e") }

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice>db.movies.insert({title:'The Hobbit:The Desolation of Smaug',writer:'J.R.R.Tolkien',year:2013,franchise:'The Hobbit'})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617aa4b128c0e36ac595494f") }

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.insert({title:'The Hobbit:The Battle of Five Armies',writer:'J.R.R.Tolkien',year:2012,franchise:'The HObbit',synopsis:'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617aa75228c0e36ac5954950") }

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.insert({title:'Avatar'})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617aa78f28c0e36ac5954951") }

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.insert({title:"Pee Wee Herman's Big Adventure"})

{

acknowledged: true,

insertedIds: { '0': ObjectId("617aa82c28c0e36ac5954952") }

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find()

[

{

\_id: ObjectId("617a92df28c0e36ac595494b"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: 1999,

actors: [ 'Brad pitt', 'Edwaerd Norton' ]

},

{

\_id: ObjectId("617a947728c0e36ac595494c"),

title: 'Plup Fiction',

writer: 'Quentin Tarantino',

year: 1994,

actors: [ 'Jhon Travolta', 'uma Thutman' ]

},

{

\_id: ObjectId("617a991228c0e36ac595494d"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

},

{

\_id: ObjectId("617aa3e528c0e36ac595494e"),

title: 'The Hobbies:An Unexpected Journey',

writer: 'J.R.R.Tolkien',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("617aa4b128c0e36ac595494f"),

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

writer: 'J.R.R.Tolkien',

year: 2013,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("617aa75228c0e36ac5954950"),

title: 'The Hobbit:The Battle of Five Armies',

writer: 'J.R.R.Tolkien',

year: 2012,

franchise: 'The HObbit',

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

},

{ \_id: ObjectId("617aa78f28c0e36ac5954951"), title: 'Avatar' },

{

\_id: ObjectId("617aa82c28c0e36ac5954952"),

title: "Pee Wee Herman's Big Adventure"

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({actors:"Brad Pitt"})

[

{

\_id: ObjectId("617a991228c0e36ac595494d"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({francise:"The Hobbit"})

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({franchise:"The Hobbit"})

[

{

\_id: ObjectId("617aa3e528c0e36ac595494e"),

title: 'The Hobbies:An Unexpected Journey',

writer: 'J.R.R.Tolkien',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("617aa4b128c0e36ac595494f"),

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

writer: 'J.R.R.Tolkien',

year: 2013,

franchise: 'The Hobbit'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({year:{$It:2010,$gt:2000}})

MongoServerError: unknown operator: $It

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({year:{$it:2010,$gt:2000}})

MongoServerError: unknown operator: $it

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({year:{$It:2010,$gt:2000}})

MongoServerError: unknown operator: $It

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({year:{$lt:2010,$gt:2000}})

[

{

\_id: ObjectId("617a991228c0e36ac595494d"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

}

]

Atlas atlas-y42dil-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 roup 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: 0,

modifiedCount: 0,

upsertedCount: 0

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> 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 magical ring'.}})

Uncaught:

SyntaxError: Unexpected token (1:263)

> 1 | 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 magical ring'.}})

| ^

2 |

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.update({title:'The Hobbit:The desolation of Smaug'},{$set:{synopsis:'The dwarves,along with Bilbo Baggins And Gandalf the Greyif 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: 0,

modifiedCount: 0,

upsertedCount: 0

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.update({title:'Pulp Fiction'},{$push:{$push:{actors:Samuel L. jacson}})

Uncaught:

SyntaxError: Unexpected token, expected "," (1:69)

> 1 | db.movies.update({title:'Pulp Fiction'},{$push:{$push:{actors:Samuel L. jacson}})

| ^

2 |

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.update({title:'Pulp Fiction'},{$push:{$actors:SamuelL. jackson}})

ReferenceError: SamuelL is not defined

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.update({title:'Pulp Fiction'},{$push:{$actors:Samuel L. Jackson}})

Uncaught:

SyntaxError: Unexpected token, expected "," (1:63)

> 1 | db.movies.update({title:'Pulp Fiction'},{$push:{$actors:Samuel L. Jackson}})

| ^

2 |

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.update({title:'Pulp Fiction'},{$push:{$actors:'Samuel' L. Jackson}})

Uncaught:

SyntaxError: Unexpected token, expected "," (1:65)

> 1 | db.movies.update({title:'Pulp Fiction'},{$push:{$actors:'Samuel' L. Jackson}})

| ^

2 |

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.upadte({title:'Pulp Fiction'},{$push:{$actors:'Samuel L. Jackson'}})

TypeError: db.movies.upadte is not a function

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.update({title:'Pulp Fiction'},{$push:{actor: 'Samel L. Jackson'}})

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.createindex({synopsis:'text'})

TypeError: db.movies.createindex is not a function

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.createIndex

[Function: createIndex] AsyncFunction {

apiVersions: [ 1, Infinity ],

returnsPromise: true,

serverVersions: [ '0.0.0', '999.999.999' ],

topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],

returnType: { type: 'unknown', attributes: {} },

deprecated: false,

platforms: [ 0, 1, 2 ],

isDirectShellCommand: false,

acceptsRawInput: false,

shellCommandCompleter: undefined,

help: [Function (anonymous)] Help

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.createIndex({synopsis:'text'})

synopsis\_text

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:'Bilbo'}})

[

{

\_id: ObjectId("617aa75228c0e36ac5954950"),

title: 'The Hobbit:The Battle of Five Armies',

writer: 'J.R.R.Tolkien',

year: 2012,

franchise: 'The HObbit',

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:'Gandalf'}})

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:'Gandalf'}})

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:'Bilbo -Gandalf'}})

[

{

\_id: ObjectId("617aa75228c0e36ac5954950"),

title: 'The Hobbit:The Battle of Five Armies',

writer: 'J.R.R.Tolkien',

year: 2012,

franchise: 'The HObbit',

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:'dwarves hobbit'}})

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:"'gold' 'dragon'"}})

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.find({$text:{$search:"'gold' 'dragon'"}})

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.remove({title: "Avatar"})

DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.

{ acknowledged: true, deletedCount: 1 }

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.movies.remove({title: "Pee Wee Herman's Big Adventure"})

{ acknowledged: true, deletedCount: 1 }

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.createCollection('users')

{ ok: 1 }

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.users.insertMany([

...

... { username: "GoodGuyGreg", first\_name:"Good Guy",last\_name:"Greg"},

... { username: "ScumbagSteve", full\_name:{first\_name:"Scumbag",last\_name:"Steve"} }

... ]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId("617ac3c828c0e36ac5954953"),

'1': ObjectId("617ac3c828c0e36ac5954954")

}

}

Atlas atlas-y42dil-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("617ac41728c0e36ac5954955"),

'1': ObjectId("617ac41728c0e36ac5954956"),

'2': ObjectId("617ac41728c0e36ac5954957"),

'3': ObjectId("617ac41728c0e36ac5954958"),

'4': ObjectId("617ac41728c0e36ac5954959"),

'5': ObjectId("617ac41728c0e36ac595495a")

}

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.comments.insertMany([

...

{username:"GoodGuyGreg",comment:"Hopeyougotagooddeal!",post:"607f49ca845ea1cba8e2ij18"},

{username:"GoodGuyGreg", comment:"What's mine is yours!",post:"607f49ca845ea1cba8e2ij19"},

{username:"GoodGuyGreg",comment:"Don'tviolatethelicensingagreement!",post:"607f49ca845ea1cba8e2ij20"},

{username: "ScumbagSteve", comment:"It still isn't clean",post:"607f49ca845ea1cba8e2ij21"},

{username:"ScumbagSteve",comment:"DeniedyourPRcauseIfoundahack",post:"607f49ca845ea1cba8e2ij22"}

... ]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId("617ac46a28c0e36ac595495b"),

'1': ObjectId("617ac46a28c0e36ac595495c"),

'2': ObjectId("617ac46a28c0e36ac595495d"),

'3': ObjectId("617ac46a28c0e36ac595495e"),

'4': ObjectId("617ac46a28c0e36ac595495f")

}

}

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.users.find()

[

{

\_id: ObjectId("617ac3c828c0e36ac5954953"),

username: 'GoodGuyGreg',

first\_name: 'Good Guy',

last\_name: 'Greg'

},

{

\_id: ObjectId("617ac3c828c0e36ac5954954"),

username: 'ScumbagSteve',

full\_name: { first\_name: 'Scumbag', last\_name: 'Steve' }

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.posts.find()

[

{

\_id: ObjectId("617ac41728c0e36ac5954955"),

username: 'GoodGuyGreg',

title: 'Passes out at party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("617ac41728c0e36ac5954956"),

username: 'GoodGuyGreg',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("617ac41728c0e36ac5954957"),

username: 'GoodGuyGreg',

title: 'Reports a bug in your code',

body: 'Sends you a pull request'

},

{

\_id: ObjectId("617ac41728c0e36ac5954958"),

username: 'ScumbagSteve',

title: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("617ac41728c0e36ac5954959"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("617ac41728c0e36ac595495a"),

username: 'ScumbagSteve',

title: 'Forks your repo on github',

body: 'Sets to private'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.posts.find({username:"GoodGuyGreg"})

[

{

\_id: ObjectId("617ac41728c0e36ac5954955"),

username: 'GoodGuyGreg',

title: 'Passes out at party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("617ac41728c0e36ac5954956"),

username: 'GoodGuyGreg',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("617ac41728c0e36ac5954957"),

username: 'GoodGuyGreg',

title: 'Reports a bug in your code',

body: 'Sends you a pull request'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.posts.find({username:"ScumbagSteve"})

[

{

\_id: ObjectId("617ac41728c0e36ac5954958"),

username: 'ScumbagSteve',

title: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("617ac41728c0e36ac5954959"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("617ac41728c0e36ac595495a"),

username: 'ScumbagSteve',

title: 'Forks your repo on github',

body: 'Sets to private'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.comments.find()

[

{

\_id: ObjectId("617ac46a28c0e36ac595495b"),

username: 'GoodGuyGreg',

comment: 'Hope you got a good deal!',

post: '607f49ca845ea1cba8e2ij18'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495c"),

username: 'GoodGuyGreg',

comment: "What's mine is yours!",

post: '607f49ca845ea1cba8e2ij19'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495d"),

username: 'GoodGuyGreg',

comment: "Don't violate the licensing agreement!",

post: '607f49ca845ea1cba8e2ij20'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495e"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: '607f49ca845ea1cba8e2ij21'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495f"),

username: 'ScumbagSteve',

comment: 'Denied your PR cause I found a hack',

post: '607f49ca845ea1cba8e2ij22'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.comments.find{username:"GoodGuyGreg"})

Uncaught:

SyntaxError: Missing semicolon. (1:16)

> 1 | db.comments.find{username:"GoodGuyGreg"})

| ^

2 |

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.comments.find({username:"GoodGuyGreg"})

[

{

\_id: ObjectId("617ac46a28c0e36ac595495b"),

username: 'GoodGuyGreg',

comment: 'Hope you got a good deal!',

post: '607f49ca845ea1cba8e2ij18'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495c"),

username: 'GoodGuyGreg',

comment: "What's mine is yours!",

post: '607f49ca845ea1cba8e2ij19'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495d"),

username: 'GoodGuyGreg',

comment: "Don't violate the licensing agreement!",

post: '607f49ca845ea1cba8e2ij20'

}

]

Atlas atlas-y42dil-shard-0 [primary] mongo\_practice> db.comments.find({username:"ScumbagSteve"})

[

{

\_id: ObjectId("617ac46a28c0e36ac595495e"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: '607f49ca845ea1cba8e2ij21'

},

{

\_id: ObjectId("617ac46a28c0e36ac595495f"),

username: 'ScumbagSteve',

comment: 'Denied your PR cause I found a hack',

post: '607f49ca845ea1cba8e2ij22'

}

]