**MONGODB**

**DAY-22**

* test> show dbs

admin 40.00 KiB

books 112.00 KiB

config 72.00 KiB

demo 112.00 KiB

local 72.00 KiB

* test> db

test

* test> use books

switched to db books

* books> show collections

bookList

demo

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 5,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 5,

price: 3000

}

]

* books> db.bookList.updateMany({$or:[{nopage:350},{nopage:250}]},{set:{rating:2}})

MongoInvalidArgumentError: Update document requires atomic operators

* books> db.bookList.updateMany({$or:[{nopage:350},{nopage:250}]},{$set:{rating:2}})

{

acknowledged: true,

insertedId: null,

matchedCount: 2,

modifiedCount: 2,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 2,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 2,

price: 3000

}

]

* books> db.bookList.updateMany({$and:[{name:'javaScript'},{nopage:300}]},{$set:{rating:5}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 2,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000,

rating: 5

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 2,

price: 3000

}

]

* books>db.bookList.updateMany({$and:[{name:'python'},{nopage:{$gt:200}}]},{$set:{rating:3.55}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 3.55,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000,

rating: 5

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 2,

price: 3000

}

]

* books> db.bookList.updateMany({$nor:[{name:'java'},{nopage:300}]},{$set:{rating:4.35}})

{

acknowledged: true,

insertedId: null,

matchedCount: 2,

modifiedCount: 2,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.35,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000,

rating: 5

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.35,

price: 3000

}

]

* books> db.bookList.updateMany({$nor:[{nopage:300}]},{$set:{rating:3.10}})

{

acknowledged: true,

insertedId: null,

matchedCount: 2,

modifiedCount: 2,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 3.1,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000,

rating: 5

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 3.1,

price: 3000

}

]

* books> db.bookList.updateMany({nopage:{$not:{$gt:300}}},{$set:{rating:4.6}})

{

acknowledged: true,

insertedId: null,

matchedCount: 4,

modifiedCount: 4,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 3000,

rating: 4.6

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 3000

}

]

* books> db.bookList.find({yearofPublish:{$gte:1990}});

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 3000

}

]

* books> db.bookList.find({$or:[{yearofPublish:{$gte:1991}},{yearofPublish:{$lte:1899}}]});

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 3000

}

]

* books> db.bookList.find({$and:[{yearofPublish:{$gte:1990}},{yearofPublish:{$lte:1991}}]});

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 3000

}

]

* books> db.bookList.find({name:{$in:['java','php','python']}})

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 3000

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 3000

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 3000

}

]

* books> db.bookList.updateMany({},{$set:{price:500}});

{

acknowledged: true,

insertedId: null,

matchedCount: 4,

modifiedCount: 4,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 500

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 500

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 500,

rating: 4.6

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 500

}

]

* books> db.bookList.update({name:'python'},{$set:{price:350}});

DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 500

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 350

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 500,

rating: 4.6

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 500

}

]

* books> db.bookList.updateMany({name:'javaScript'},{$mul:{price:30}});

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 500

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 350

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 15000,

rating: 4.6

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 500

}

]

* books> db.bookList.aggregate([{$match:{name:'java'}},{$project:{name:1,nopage:1,price:{$divide:["$price",2]}}}])

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

price: 250

}

]

* books> db.bookList.aggregate([{$match:{name:'java'}},{$project:{name:1,nopage:1,price:{$multiply:["$price",2]}}}])

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

price: 1000

}

]

* books> db.bookList.aggregate([{$match:{name:'java'}},{$project:{name:1,nopage:1,price:{$mod:["$price",2]}}}])

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

price: 0

}

]

* books> db.bookList.aggregate([{$match:{name:'java'}},{$project:{name:1,nopage:1,price:{$pow:["$price",2]}}}])

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

price: 250000

}

]

* books> db.bookList.aggregate([{$group:{\_id:"$name"}}])

[

{ \_id: 'php' },

{ \_id: 'javaScript' },

{ \_id: 'java' },

{ \_id: 'python' }

]

* books> db.bookList.update({'author.name':'duido van Rossum'},{$push:{author:{name:'shekar',born:1890,place:'India'}}})

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 500

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 350

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 15000,

rating: 4.6

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [ { name: 'rasmus', born: 1980, place: 'USA' } ],

yearofPublish: 1990,

rating: 4.6,

price: 500

}

]

* books> db.bookList.update({'author.name':'rasmus'},{$push:{author:{name:'shekar',born:1890,place:'India'}}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

* books> db.bookList.find()

[

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39b"),

name: 'java',

nopage: 300,

author: [ { name: 'james gosling', place: 'canada', born: 1995 } ],

rating: 4.6,

price: 500

},

{

\_id: ObjectId("63d0b30034a9cb8e4a47a39c"),

name: 'python',

nopage: 250,

author: [ { name: 'duido van Ross', born: 1995, place: 'Netherlands' } ],

yearofPublish: 1991,

rating: 4.6,

price: 350

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39d"),

name: 'javaScript',

nopage: 300,

author: [

{ name: 'CrockFord', place: 'canada', born: 1986 },

{ name: 'John Resig', place: 'USA', born: 1994 }

],

yearOfPublish: 1991,

price: 15000,

rating: 4.6

},

{

\_id: ObjectId("63d0b43134a9cb8e4a47a39e"),

name: 'php',

nopage: 250,

author: [

{ name: 'rasmus', born: 1980, place: 'USA' },

{ name: 'shekar', born: 1890, place: 'India' }

],

yearofPublish: 1990,

rating: 4.6,

price: 500

}

]