1. **Write a MongoDB query to display at least 10 details of the documents of book id, Btitle, authors in array and price for all the documents in the collection Bookstore.**

kgisl> db.books.find({}, { \_id: 0, bookid: 1, title: 1, authors: 1, price: 1 }).limit(10)

[

{ bookid: 1, title: 'Book 1', price: 19.99 },

{ bookid: 2, title: 'Book 2', price: 14.99 },

{ bookid: 3, title: 'Book 3', price: 12.99 },

{ bookid: 4, title: 'Book 4', price: 9.99 },

{ bookid: 5, title: 'Book 5', price: 24.99 },

{ bookid: 6, title: 'Book 6', price: 17.99 },

{ bookid: 7, title: 'Book 7', price: 15.99 },

{ bookid: 8, title: 'Book 8', price: 21.99 },

{ bookid: 9, title: 'Book 9', price: 10.99 },

{ bookid: 10, title: 'Book 10', price: 29.99 }

]

1. **Write a MongoDB query to display the fields Btitle, authors and price but exclude the bookid for all the documents in the collection Bookstore.**

kgisl> db.books.find({}, { \_id: 0, Btitle: 1, authors: 1, price: 1 })

[

{ price: 19.99 },

{ price: 14.99 },

{ price: 12.99 },

{ price: 9.99 },

{ price: 24.99 },

{ price: 17.99 },

{ price: 15.99 },

{ price: 21.99 },

{ price: 10.99 },

{ price: 29.99 }

]

1. **Write a MongoDB query to display the first 3 books based on similar authors.**

kgisl> db.books.find().sort({authors:1}).limit(3)

[

{

\_id: ObjectId("653e09eaafc210868365ca4b"),

bookid: 1,

title: 'Book 1',

author: 'Author 1',

price: 19.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4c"),

bookid: 2,

title: 'Book 2',

author: 'Author 2',

price: 14.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4d"),

bookid: 3,

title: 'Book 3',

author: 'Author 3',

price: 12.99

}

]

1. **Write a MongoDB query to find the book price more than 1500.**

kgisl> db.books.find({price:{$gt:10.99}})

[

{

\_id: ObjectId("653e09eaafc210868365ca4b"),

bookid: 1,

title: 'Book 1',

author: 'Author 1',

price: 19.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4c"),

bookid: 2,

title: 'Book 2',

author: 'Author 2',

price: 14.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4d"),

bookid: 3,

title: 'Book 3',

author: 'Author 3',

price: 12.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4f"),

bookid: 5,

title: 'Book 5',

author: 'Author 5',

price: 24.99

},

{

\_id: ObjectId("653e09eaafc210868365ca50"),

bookid: 6,

title: 'Book 6',

author: 'Author 6',

price: 17.99

},

{

\_id: ObjectId("653e09eaafc210868365ca51"),

bookid: 7,

title: 'Book 7',

author: 'Author 7',

price: 15.99

},

{

\_id: ObjectId("653e09eaafc210868365ca52"),

bookid: 8,

title: 'Book 8',

author: 'Author 8',

price: 21.99

},

{

\_id: ObjectId("653e09eaafc210868365ca54"),

bookid: 10,

title: 'Book 10',

author: 'Author 10',

price: 29.99

}

]

**5. Write a MongoDB query to find the book title who's price is more than 800 but less than 2000.**

kgisl> db.books.find({price:{$gt:15.99,$lt:24.99}},{Btitle:1})

[

{ \_id: ObjectId("653e09eaafc210868365ca4b") },

{ \_id: ObjectId("653e09eaafc210868365ca50") },

{ \_id: ObjectId("653e09eaafc210868365ca52") }

]

**6. Write a MongoDB query to arrange the book name of the bookstore in ascending order along with all the columns.**

kgisl> db.books.find().sort({Btitle:1})

[

{

\_id: ObjectId("653e09eaafc210868365ca4b"),

bookid: 1,

title: 'Book 1',

author: 'Author 1',

price: 19.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4c"),

bookid: 2,

title: 'Book 2',

author: 'Author 2',

price: 14.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4d"),

bookid: 3,

title: 'Book 3',

author: 'Author 3',

price: 12.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4e"),

bookid: 4,

title: 'Book 4',

author: 'Author 4',

price: 9.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4f"),

bookid: 5,

title: 'Book 5',

author: 'Author 5',

price: 24.99

},

{

\_id: ObjectId("653e09eaafc210868365ca50"),

bookid: 6,

title: 'Book 6',

author: 'Author 6',

price: 17.99

},

{

\_id: ObjectId("653e09eaafc210868365ca51"),

bookid: 7,

title: 'Book 7',

author: 'Author 7',

price: 15.99

},

{

\_id: ObjectId("653e09eaafc210868365ca52"),

bookid: 8,

title: 'Book 8',

author: 'Author 8',

price: 21.99

},

{

\_id: ObjectId("653e09eaafc210868365ca53"),

bookid: 9,

title: 'Book 9',

author: 'Author 9',

price: 10.99

},

{

\_id: ObjectId("653e09eaafc210868365ca54"),

bookid: 10,

title: 'Book 10',

author: 'Author 10',

price: 29.99

}

]

**7.Write a MongoDB query to arrange the book name of the bookstore in ascending order along with all the columns.**

kgisl> db.books.find().sort({Btitle:1})

[

{

\_id: ObjectId("653e09eaafc210868365ca4b"),

bookid: 1,

title: 'Book 1',

author: 'Author 1',

price: 19.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4c"),

bookid: 2,

title: 'Book 2',

author: 'Author 2',

price: 14.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4d"),

bookid: 3,

title: 'Book 3',

author: 'Author 3',

price: 12.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4e"),

bookid: 4,

title: 'Book 4',

author: 'Author 4',

price: 9.99

},

{

\_id: ObjectId("653e09eaafc210868365ca4f"),

bookid: 5,

title: 'Book 5',

author: 'Author 5',

price: 24.99

},

{

\_id: ObjectId("653e09eaafc210868365ca50"),

bookid: 6,

title: 'Book 6',

author: 'Author 6',

price: 17.99

},

{

\_id: ObjectId("653e09eaafc210868365ca51"),

bookid: 7,

title: 'Book 7',

author: 'Author 7',

price: 15.99

},

{

\_id: ObjectId("653e09eaafc210868365ca52"),

bookid: 8,

title: 'Book 8',

author: 'Author 8',

price: 21.99

},

{

\_id: ObjectId("653e09eaafc210868365ca53"),

bookid: 9,

title: 'Book 9',

author: 'Author 9',

price: 10.99

},

{

\_id: ObjectId("653e09eaafc210868365ca54"),

bookid: 10,

title: 'Book 10',

author: 'Author 10',

price: 29.99

}

]

**8. Write a MongoDB query to know whether the author in the authors list.**

kgisl> db.books.find({author:"authorname"}).count()>0

false

**9. Write a MongoDB query to update & increase the book price Rs.500 in all book name as "MongoDB Practice".**

kgisl> db.books.update({Btitle:"book 8"},{$inc:{price:500}},{multi:true})

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

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

**10. Write a MongoDB query to use Update modifiers ($set, Spush, Spull, SaetOnInsert, Supsert) in bookstore.**

**1.set**

kgisl> db.books.update({Btitle:"book1"},{$set:{price:19.99}})

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

**2.push**

kgisl> db.books.update({Btitle:"book2"},{$push:{author:"new author"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

**3.pull**

kgisl> db.books.update({Btitle:"book3"},{$pull:{author:"author 1"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

**4.setoninsert**

kgisl> db.books.update({Btitle:"newbook"},{$setOnInsert:{publication\_year:2023}},{upsert:true})

{

acknowledged: true,

insertedId: ObjectId("653e114f5158bf291ca3d299"),

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 1

}

**5.upsert**

kgisl> db.books.update({Btitle:"non-existent book"},{$set:{price:10.99}},{upsert:true})

{

acknowledged: true,

insertedId: ObjectId("653e11ee5158bf291ca3d2bb"),

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 1

}