ALL Query of MongoDB

1. **Find all books written by John Adam**
   - db.books.find({ author: { $eq: "John Adams" } })

     output:

     title: 'The Great Adventure'

2. **Find all books where the price is between 200 and 400**
   - db.books.find({price:{$gte:200,$lte:400}})

     output:

     title: 'The Great Adventure',

     title: 'Learning MongoDB'

3. **Find document that have publisher field**
   - db.books.find({publisher:{$exists:true}})

     output:

     title: 'The Great Adventure',

     title: 'Learning MongoDB'

4. **Find all books tagged with technology**
   - db.books.find({tags:{$eq:"technology"}})

     output:

     title: 'Learning MongoDB'

5. **Find all books where the price of the book equal to 299**
   - db.books.find({price:{$eq:299}})

     output:

     title: 'The Great Adventure'

6. **Retrieve all document where the price field is off the datatype double**
   - db.books.find({price:{$type:"double"}})

     output:

     title: 'Cooking with Love'

7. **Find document that don't have publisher field**
   - db.books.find({ publisher: { $exists: false } })

     output:

     title: 'Cooking with Love'

8. **Find all books tagged with both non-fiction and lifestyle**
   - db.books.find({tags:{$all:["non-fiction", "lifestyle"]}})

     output:

     title: 'Cooking with Love',

9. **Find all books that have a reviews with the rating 4.**
   - db.books.find({reviews:{$elemMatch:{rating:{$eq:4}}}})

     output:

     title: 'The Great Adventure',

     title: 'Cooking with Love'

10. **Find all books where the price have the value of either 100, 150.50, 200**
    - db.books.find({price:{$in:[100,150.50,200]}})

    output:
    title: 'Cooking with Love'

11. **Find all document tagged with both non-fiction and technology**
    - db.books.find({ tags: { $all: ["non-fiction", "technology"] } })

    output:
    title: 'Learning MongoDB'

12. **Find all books that have a reviews with a rating of 3 and comment containing 'great'.**
    - db.books.find({reviews: {$elemMatch: {rating: { $gte: 3 },comment: { $regex:"great", $options:"i" }}}})

    output:
    title: 'Cooking with Love'

13. **Find all books that has exactly 4 tags.**
    - db.books.find({ tags: { $size: 4 } })

    output:
    title: 'Cooking with Love'

14. **Count the number of book in the collection**
    - db.books.countDocuments()

    output: 3

15. **Find all books that have at least one reviews with a rating greater than or equal to 4 and the reviewer is 'Fiona'.**
    - db.books.find({reviews: {$elemMatch: {rating: { $gte: 4 },reviewer: { $eq: "Fiona" }}}})

    output: No Document.

16. **Find all books published after January 1st 2022.**
    - db.books.find({ publishedDate: { $gt: "2022-01-01" } })

    output:
    title: 'Learning MongoDB'

17. **Find all books where the title field starts with 'The' ignoring case.**
    - db.books.find({title: { $regex:/^The/, $options: "i" }})

    output:
    title: 'The Great Adventure',

18. **Find all books where the name of the publisher ends with 'press'.**
    - db.books.find({publisher: {$regex:/press$/,$options: "i"}})

    output:
    title: 'Learning MongoDB'

19. **Find all books where the reviews contains at least one element that has a rating of 5 or higher and also includes a comment.**
    - db.books.find({reviews: {$elemMatch: {rating: { $gte: 5 },comment: { $exists: true }}}})

    output:

title: 'Learning MongoDB'

20. **Find books where the title starts with 'The' and the authors name ends with dams both case insensitive.**
    - db.books.find({$and: [{title: {$regex: /^The/,$options: "i"}},{author: {$regex: /dams$/, $options: "i"}}]})

    output:
    title: 'The Great Adventure'

21. **Find books where the word MongoDB appears in the title field only.**
    - db.books. createIndex({title:"text"})
    - db.books.find({$text:{$search:"MongoDB"}})

    output:
    title: 'Learning MongoDB'

22. **Find all books where the price is divisible by 5.**
    - db.books.find({price:{$mod:[5,0]}})

    output:
    title: 'Cooking with Love',

23. **Find books that have either fiction or technology as a tags.**
    - db.books.find({tags:{$in:["fiction", "technology"]}})

    output:
    title: 'The Great Adventure',
    title: 'Learning MongoDB'

24. **Find all books where the price is greater than the discounted Price.**
    - db.books.find({discountedPrice:{$exists:true},$expr:{$gt:["$price","$discountedPrice"]}})

    output:
    title: 'The Great Adventure',
    title: 'Learning MongoDB'

25. **Find all books where the title contains the word Adventure and has more than one reviews.**
    - db.books.find({$where:function(){return this.title="Adventure" && this.reviews.length>1;}})

    output:
    title: 'The Great Adventure',
    title: 'Cooking with Love'

26. **Find books where the price is greater than 100 and the number of reviews is more than 2.**
    - db.books.find({$where:function(){return this.price>100 && this.reviews.length>2;}})

    output:
    title: 'Cooking with Love'

27. **Find all books that are currently available.**
    - db.books.find({available:{$eq:true}})

output:

title: 'The Great Adventure',

title: 'Cooking with Love'

28. Retrieve all books sorted by price in ascending order.

- db.books.find().sort({price:1})

output:

title: 'Cooking with Love',

title: 'The Great Adventure',

title: 'Learning MongoDB'

29. Retrieve all books sorted by discounted price in descending order.

- db.books.find().sort({discountedPrice:-1})

output:

title: 'Learning MongoDB',

title: 'The Great Adventure',

title: 'Cooking with Love'

30. Update the price of Learning MongoDB to 450.

- db.books. updateOne({title:"Learning MongoDB"}, {$set:{price:450}})

output:

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 0,

upsertedCount: 0

}

title: 'Learning MongoDB'

31. Count books by author.

- db.books. .aggregate([{$group:{_id:"$author", totalBook:{$sum:1}}}])

output:

{

_id: 'James Joyee',

totalBook: 1

}

{

_id: 'John Adams',

totalBook: 1

}

{

_id: 'Jane Smith',

totalBook: 1

}

32. Find the minimum discounted Price of books.

- db.books. aggregate([{$group:{_id:"null", minDiscountedPrice: {$min:"$price"}}}])

output:

{

_id: 'null',

minDiscountedPrice: 150.5

}

33. **Find the maximum price of books.**
    - db.books. aggregate([{$group:{_id:"null", minDiscountedPrice: {$min:"$price"}}}])

    output:

```
{
  _id: 'null',
  minDiscountedPrice: 150.5
}
```

34. **Get a list of all unique tags across all books tags.**
    - db.books. aggregate([{$group:{_id:null, uniqueTags: {$addToSet:"$tags"}}}])

    output:

```
{
  _id: null,
  uniqueTags: [
    [
      'fiction',
      'adventure',
      'best seller'
    ],
    [
      'non-fiction',
      'technology',
      'database'
    ],
    [
      'non-fiction',
      'cooking',
      'lifestyle',
      'best seller'
    ]
  ]
}
```

35. **Remove the books titled cooking with love from the collection.**
    - db.books.deleteOne()

    output:

```
{
  acknowledged: true,
  deletedCount: 1
}
```

```
db.books.insertMany([
  {
    title: "The Great Adventure",
    author: "Jhon Adams",
    price: 299,
    discountedPrice: 249,
    publisher: "Adventure Books",
    tags: ["fiction", "adventure", "bestseller"],
    reviews: [
      { reviewer: "Alice", rating: 4, comment: "Excellent Book!" },
```

```
      { reviewer: "Bob", rating: 3, comment: "Very enjoyable" }
    ],
    PublishedDate: ISODate("2021-05-15"),
    available: true
  },
  {
    title: "Learning MongoDB",
    author: "Jane Smith",
    price: 399,
    discountedPrice: 359,
    publisher: "Tech Press",
    tags: ["non-fiction", "technology", "database"],
    reviews: [
      { reviewer: "Charlie", rating: 5, comment: "Very informative" }
    ],
    PublishedDate: ISODate("2022-08-22"),
    available: false
  },
  {
    title: "Cooking with Love",
    author: "James Joyce",
    price: 150.50,
    tags: ["non-fiction", "cooking", "lifestyle", "bestseller"],
    reviews: [
      { reviewer: "Fiona", rating: 3, comment: "Great" },
      { reviewer: "David", rating: 2, comment: "Good" }
    ],
    PublishedDate: ISODate("2020-11-30"),
    available: true
  }
]);
```