MongoDB Questions and Answers

1. How do you limit the number of documents in the output of an aggregation pipeline?

A) Use $slice

B) Use $limit

C) Use $size

D) Use $count

b

2. Which method is used to create an index in MongoDB?

A. db.collection.addIndex()

B. db.collection.index()

C. db.collection.ensureIndex()

D. db.collection.createIndex()

d

3. Which index type is best suited for full-text search in MongoDB?

A. Hashed index

B. Text index

C. Sparse index

D. Compound index

b

4. What is a compound index?

A. An index that stores multiple collections

B. An index on multiple fields

C. An index that only supports geospatial data

D. A backup index

b

5. What does the explain() method do in MongoDB?

A. Displays syntax of a command

B. Shows the schema design

C. Provides information on how a query is executed

D. Creates a report

c

6. Which of the following statements about indexes is FALSE?

A. Indexes can improve query performance

B. Indexes can slow down write operations

C. MongoDB automatically creates indexes for every field

D. Indexes consume additional disk space

c

7. In MongoDB, what is a "collection"?

a) A group of tables

b) A group of documents

c) A single document

d) A type of index

b

8. What is the first stage in most aggregation pipelines?

A) $group

B) $sort

C) $match

D) $project

c

9. Which stage is used to reshape documents by including or excluding fields?

A) $project

B) $unwind

C) $limit

D) $skip

a

10. What does the $group stage do in a pipeline?

A) Filters documents

B) Groups documents by a specified key

C) Projects specific fields

D) Flattens arrays

B

11. What does the $unwind stage do?

A) Joins two collections

B) Splits array fields into multiple documents

C) Removes null values

D) Sorts documents

b

12. Which of the following can be used inside a $group stage?

A) $match

B) $avg

C) $skip

D) $limit

b

13. What is the use of $lookup in aggregation?

A) To sort documents

B) To perform left outer join with another collection

C) To flatten nested documents

D) To filter based on conditions

b

14. Which of the following will reduce the number of documents early in the pipeline?

A) $group

B) $match

C) $sort

D) $project

b

15. Which of the following is used to store data in MongoDB?

A. Table

B. Row

C. Document

D. Schema

C

16. Which of the following is used to compute the number of documents in a pipeline?

A) $count

B) $sum

C) $project

D) $max

a

17. What does $addFields do in aggregation?

A) Replaces all fields

B) Adds or updates fields without removing others

C) Groups documents

D) Projects only new fields

b

18. What type of join does $lookup perform by default?

A) Inner join

B) Right outer join

C) Full join

D) Left outer join

d

19. What field is used to match documents in the foreign collection during a $lookup?

A) as

B) foreignField

C) localField

D) from

b

20. What does the as field specify in a $lookup operation?

A) The database name

B) The field to sort by

C) The name of the new array field for the joined documents

D) The alias for the localField

C

21. Which of the following is a valid $lookup syntax in MongoDB?

A) { $lookup: { from: "orders", matchField: "userId", result: "userOrders" } }

B) { $lookup: { from: "orders", localField: "userId", foreignField: "user_id", as: "orders" } }

C) { $lookup: { collection: "orders", on: "userId", alias: "orders" } }

D) { $lookup: { collection: "orders", on: "userId", alias: "orders" } }

b

22. What will be the value of the as field if there is no matching document in the foreign collection?

A) null

B) undefined

C) An empty array

D) The original document is excluded

c

23. Which stage can be used after $lookup to deconstruct an array of joined documents?

A) $split

B) $project

C) $unwind

D) $merge

c

24. What aggregation feature allows more complex conditions when joining collections?

A) localField

B) mergeObjects

C) $lookup with let and pipeline

D) $group

c

25. What is a view in MongoDB?

A) A table-like object storing data permanently

B) A pre-aggregated dataset stored as a collection

C) A virtual collection defined by an aggregation pipeline

D) A stored procedure

c

26. Can you insert documents directly into a MongoDB view?

A) Yes

B) No

C) Only if it's indexed

D) Only in version 6.0+

b

27. Which command is used to create a view in MongoDB?

A) db.createView()

B) db.createCollection() with viewOn

C) db.createVirtual()

D) db.createPipeline()

a

28. Which of the following best describes a limitation of views in MongoDB?

A) They require indexes

B) They can't be queried

C) They can't contain joins

D) They are not updatable

d

29. What will db.collection.insertMany() do?

A. Insert one document

B. Insert multiple documents

C. Replace one document

D. Remove one document

b

30. In MongoDB, data is stored in:

A. Tables and Rows

B. Collections and Documents

C. Schemas and Entities

D. Tables and Columns

B

31. Which of the following best describes MongoDB’s data model?

A. Rigid and relational

B. Fixed schema

C. Flexible and schema-less

D. Object-relational

c

32. Which MongoDB operator is used to match multiple conditions where all must be true?

A. $or

B. $match

C. $and

D. $all

c

33. What does limit(5) do in a query like db.collection.find().limit(5)?

A. Skips the first 5 results

B. Limits the number of results to 5

C. Returns only documents with 5 fields

D. Returns the last 5 results

b

34. How can you sort documents in descending order by a field called score?

A. db.collection.sort({ score: 'desc' })

B. db.collection.find().sort({ score: -1 })

C. db.collection.find().orderBy({ score: -1 })

D. db.collection.sortDescending('score')

b

35. What does the $in operator do in MongoDB queries?

A. Checks if a value is not in an array

B. Checks if a value is in an array of specified values

C. Compares two fields

D. Joins two collections

B

36. Which operator is used to check if a field equals a specific value?

A. $eq

B. $equal

C. $match

D. $value

a

37. What does the following query return?

db.users.find({ age: { $gt: 25 } })

A. Users with age less than 25

B. Users with age greater than or equal to 25

C. Users with age greater than 25

D. All users

c

38. What method is used to delete a single document that matches a condition?

A. removeOne()

B. deleteOne()

C. deleteFirst()

D. removeFirst()

b

39. Which of the following is the correct syntax to find all documents in a collection?

A. db.collection.select()

B. db.collection.fetchAll()

C. db.collection.find({})

D. db.collection.readAll()

c

40. What is the main purpose of an index in MongoDB?

A. To store documents

B. To enforce relationships between collections

C. To improve the performance of queries

D. To control access to collections

c

41. What does the deleteMany() method do?

A. Deletes the database

B. Deletes all documents in a collection

C. Deletes all documents that match a condition

D. Deletes only the first matching document

c

42. Which operator is commonly used to set new values in an update operation?

A. $change

B. $update

C. $modify

D. $set

d

43. Which method is used to update a single document in MongoDB?

A. update()

B. updateMany()

C. updateOne()

D. modifyOne()

c

44. What does the find() method do in MongoDB?

A. Deletes documents

B. Updates documents

C. Inserts documents

D. Reads/fetches documents

d

45. What is the primary key field in every MongoDB document?

A. id

B. doc_id

C. _id

D. key

C

46. Which command is used to view all databases in MongoDB shell?

A. show tables

B. list dbs

C. display databases

D. show dbs

d

47. What is the default port for MongoDB?

A. 8080

B. 3306

C. 27017

D. 5432

c

48. Which of the following is true about collections in MongoDB?

A. Collections contain rows

B. Collections are like tables in RDBMS

C. A collection can have only one type of document

D. Collections must have a fixed schema

b

49. What format are MongoDB documents stored in?

A. XML

B. CSV

C. JSON

D. BSON

d

50. What type of database is MongoDB?

A. Relational Database

B. Document-Oriented Database

C. Key-Value Store

D. Graph Database

B

51. What type of index does MongoDB create by default on every collection?

A. Hashed index

B. Geospatial index

C. Compound index

D. Index on the _id field

d

52. Which of the following is NOT a good reason to use referencing instead of embedding?

A. To reduce duplication

B. To manage large arrays

C. To improve write speed

D. To ensure faster reads of related data

d

53. What is the maximum size of a MongoDB document?

A. 4 MB

B. 8 MB

C. 10 MB

D. 16 MB

d

54. When is referencing preferred over embedding in MongoDB?

A. When data is always accessed together

B. When documents grow too large

C. When updates are never required

D. When all data is static

b

55. What is embedding in MongoDB schema design?

A. Linking one document to another using foreign keys

B. Storing documents as files

C. Nesting related data inside a single document

D. Creating indexes

c

56. Which data modeling strategy is commonly used in MongoDB when related data is frequently accessed together?

A. Normalization

B. Denormalization

C. Partitioning

D. Sharding

b

57. Design a schema for a books collection where each book has a title, author, and an array of reviews. Each review should include the reviewer's name and comment.

```
{
  title: "Clean Code",
  author: "Robert C. Martin",
  reviews: [
    { reviewer: "Jane", comment: "Excellent read!" },
    { reviewer: "John", comment: "Very insightful." }
  ]
}
```

58. Connect to a local MongoDB instance and switch to the school database.

```
mongosh
use school
```

59. Create an index on the name field of the students collection.

```
db.students.createIndex({ name: 1 });
```

60. Calculate the average age of enrolled students.

```
db.students.aggregate([
  { $match: { enrolled: true } },
  { $group: { _id: null, avgAge: { $avg: "$age" } } }
]);
```

61. Count the number of students per course.

```
db.students.aggregate([
  { $unwind: "$courses" },
  { $group: { _id: "$courses", count: { $sum: 1 } } }
]);
```

62. Retrieve the names of all students, excluding the _id field.

```
db.students.find({}, { _id: 0, name: 1 });
```

63. Find all students enrolled in the "Math" course.

```
db.students.find({ courses: "Math" });
```

64. Find all students older than 21.

```
db.students.find({ age: { $gt: 21 } });
```

65. Delete all students who are not enrolled.

```
db.students.deleteMany({ enrolled: false });
```

66. Update the age of the student named "Alice Johnson" to 24.

```
db.students.updateOne(
  { name: "Alice Johnson" },
  { $set: { age: 24 } }
);
```

67. Insert the following document into a collection called students:

```
{
  "name": "Alice Johnson",
  "age": 23,
  "courses": ["Math", "Physics"],
  "enrolled": true
}
```

```
db.students.insertOne({
  name: "Alice Johnson",
  age: 23,
  courses: ["Math", "Physics"],
  enrolled: true
});
```

68. Perform bulk insert of 3 students:

```
db.students.insertMany([
  { name: "Tom", age: 22 },
  { name: "Sara", age: 24 },
  { name: "Mike", age: 21 }
]);
```

69. How do you restore a backup of the students collection?

```
mongorestore -d school -c students d:/backup/school/students.bson
```

70. What is the purpose of a replica set?

A replica set provides high availability and redundancy by having multiple MongoDB instances with one primary and multiple secondaries. If the primary fails, an automatic failover promotes a secondary.

71. How do you enable sharding on a database?

```
sh.enableSharding("mydb")

sh.shardCollection("mydb.mycollection", { shardKeyField: 1 })
```

Make sure sharding is set up with config servers and mongos.

72. Find all reviews with a rating greater than 4 in nested structure:

```
// Document example:

{

  name: "Alice",

  reviews: [

    { course: "Math", rating: 5 },

    { course: "Physics", rating: 3 }

  ]

}

db.students.find({ "reviews.rating": { $gt: 4 } });
```

73. Add a new course "Chemistry" to a student’s courses array, only if it doesn't already exist:

```
db.students.updateOne(

  { name: "Alice" },

  { $addToSet: { courses: "Chemistry" } }

);
```

74. Increment age by 1 for all enrolled students:

```
db.students.updateMany(

  { enrolled: true },

  { $inc: { age: 1 } }

);
```

75. Return only name and age for students, excluding _id:

```
db.students.find({}, { _id: 0, name: 1, age: 1 });
```

76. Skip the first 10 documents and return the next 5:

```
db.students.find().skip(10).limit(5);
```

77. What command would you use to back up the mydb database?

```
mongodump -d mydb -o d:/backup
```

78. Remove a course "Physics" from Alice’s courses:

```
db.students.updateOne(
  { name: "Alice" },
  { $pull: { courses: "Physics" } }
);
```

79. Rename the field dob to dateOfBirth in all documents:

```
db.students.updateMany(
  {},
  { $rename: { "dob": "dateOfBirth" } }
);
```

80. Create a capped collection for logging (1MB max, 1000 docs):

```
db.createCollection("logs", { capped: true, size: 1048576, max: 1000 });
```

81. Get the top 3 oldest students.

```
db.students.find().sort({ age: -1 }).limit(3);
```

82. Group students by enrollment status and get the average age per group.

```
db.students.aggregate([
  { $group: { _id: "$enrolled", avgAge: { $avg: "$age" } } }
]);
```

83. Retrieve students who are enrolled and either younger than 20 or taking "Biology"

```
db.students.find({
  enrolled: true,
  $or: [
    { age: { $lt: 20 } },
    { courses: "Biology" }
  ]
});
```

84. Find all students whose name starts with "A".

```
db.students.find({
  name: { $regex: /^A/, $options: 'i' }
});
```