

1. MongoDB क्या है?

MongoDB एक **NoSQL database** है, जो document-oriented storage प्रदान करता है। यह JSON जैसी BSON (Binary JSON) format का उपयोग करता है और high performance, scalability, और flexibility प्रदान करता है।

2. MongoDB में Document क्या है?

MongoDB में **document** JSON के समान data structure होता है, जिसमें key-value pairs होते हैं।

json

CopyEdit

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "name": "John Doe",
  "age": 30,
  "city": "New York"
}
```

3. MongoDB में Collection क्या है?

Collection MongoDB में table के समान होती है, जो एक तरह के documents को store करती है।

javascript

CopyEdit

```
db.createCollection("users");
```

4. BSON क्या है?

BSON (Binary JSON) MongoDB द्वारा उपयोग किया जाने वाला data format है, जो JSON को binary format में store करता है और high-speed data access प्रदान करता है।

5. MongoDB में Index क्या है?

Index data retrieval को तेज़ बनाते हैं।

javascript

CopyEdit

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

6. MongoDB में Sharding क्या है?

Sharding **horizontal scaling** प्रदान करता है, जिससे data को multiple servers पर distribute किया जा सकता है।

7. MongoDB में Replica Set क्या है?

Replica Set में **multiple copies** होती हैं, जिससे data redundancy और availability बढ़ती है।

javascript

CopyEdit

```
rs.initiate()
```

8. MongoDB में _id Field का उपयोग क्या है?

_id प्रत्येक document के लिए एक unique पहचानकर्ता (Primary Key) होता है।

9. Compound Index क्या है?

यह multiple fields पर index बनाने की अनुमति देता है।

javascript

CopyEdit

```
db.users.createIndex({ name: 1, age: -1 });
```

10. MongoDB में Index के प्रकार:

- Single Field Index
 - Compound Index
 - Multikey Index
 - Text Index
 - Geospatial Index (2dsphere, 2d)
-

11. MongoDB में Write Concerns क्या हैं?

Write Concerns यह निर्धारित करते हैं कि data कब safe तरीके से store हुआ है।

javascript

CopyEdit

```
db.users.insertOne({ name: "John" }, { writeConcern: { w: "majority" } });
```

12. MongoDB में Collection कैसे बनाई जाती है?

javascript

CopyEdit

```
db.createCollection("employees");
```

13. MongoDB में Projection क्या है?

Projection का उपयोग केवल कुछ specific fields को select करने के लिए किया जाता है।

javascript

CopyEdit

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

14. MongoDB में Join कैसे करें?

MongoDB में \$lookup का उपयोग करके joins किया जाता है।

javascript

CopyEdit

```
db.orders.aggregate([
  {
    $lookup: {
      from: "customers",
      localField: "customerId",
      foreignField: "_id",
      as: "customerDetails"
```

```
}  
}  
]);
```

15. Capped Collection क्या है?

Capped Collection fixed-size वाली collection होती है, जिसमें पुराना data automatic delete हो जाता है।

javascript

CopyEdit

```
db.createCollection("logs", { capped: true, size: 100000, max: 1000 });
```

16. MongoDB का Backup कैसे लें?

bash

CopyEdit

```
mongodump --out /backup/mongodb/
```

17. MongoDB में \$gt Query Operator क्या है?

यह greater than operator होता है।

javascript

CopyEdit

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

18. MongoDB Atlas क्या है?

MongoDB Atlas एक **cloud-based managed database service** है, जो automatic scaling और backups प्रदान करता है।

19. MongoDB की Limitations क्या हैं?

- Joins की कमी
- High memory consumption

- **Complex transactions का अभाव**
-

20. Write Concern क्या होता है?

Write Concern यह सुनिश्चित करता है कि data कब सफलतापूर्वक store किया गया।

javascript

CopyEdit

```
db.users.insertOne({ name: "Alice" }, { writeConcern: { w: 1 } });
```

21. MongoDB में 2dsphere Index क्या है?

GeoJSON data के लिए उपयोग किया जाता है।

javascript

CopyEdit

```
db.places.createIndex({ location: "2dsphere" });
```

22. SQL और MongoDB में अंतर:

Feature	SQL (RDBMS)	MongoDB (NoSQL)
Schema	Fixed	Dynamic
Joins	Supported	Limited
Scaling	Vertical	Horizontal
Transactions	Strong ACID	Eventual Consistency

23. MongoDB में _id क्या है?

_id MongoDB में primary key होती है, जो document को unique बनाती है।

javascript

CopyEdit

```
db.users.find({}, { _id: 1 });
```

24. MongoDB में Collection कैसे बनाएं?

javascript

CopyEdit

```
db.createCollection("products");
```

25. MongoDB High Availability कैसे सुनिश्चित करता है?

MongoDB **Replica Sets** का उपयोग करके **High Availability (HA)** सुनिश्चित करता है। इसमें एक **Primary Node** और कई **Secondary Nodes** होते हैं। यदि Primary Node डाउन हो जाता है, तो Secondary Node **Automatic Failover** द्वारा Primary बन जाता है।

javascript

CopyEdit

```
rs.initiate()
```

26. Aggregation Pipeline क्या है?

Aggregation Pipeline MongoDB में data को process करने का एक तरीका है, जिसमें data को **Stages (चरणों)** में प्रोसेस किया जाता है।

javascript

CopyEdit

```
db.orders.aggregate([
  { $match: { status: "shipped" } },
  { $group: { _id: "$customerId", total: { $sum: "$amount" } } }
]);
```

27. db.collection.updateOne() क्या करता है?

यह एक ही **document** को update करता है।

javascript

CopyEdit

```
db.users.updateOne({ name: "Alice" }, { $set: { age: 30 } });
```

28. db.collection.insertOne() क्या करता है?

यह एक **document** को collection में insert करता है।

javascript

CopyEdit

```
db.users.insertOne({ name: "John", age: 25 });
```

29. MongoDB में Document कैसे Delete करें?

- एक document delete करने के लिए:

javascript

CopyEdit

```
db.users.deleteOne({ name: "John" });
```

- सभी matching documents delete करने के लिए:

javascript

CopyEdit

```
db.users.deleteMany({ age: { $gt: 30 } });
```

30. \$match Stage क्या है?

Aggregation Pipeline में \$match stage को **फिल्टरिंग** के लिए उपयोग किया जाता है।

javascript

CopyEdit

```
db.users.aggregate([  
  { $match: { age: { $gte: 30 } } }  
]);
```

31. MongoDB में Query Results को Sort कैसे करें?

sort() method का उपयोग किया जाता है।

- Ascending Order (छोटे से बड़े):

javascript

CopyEdit

```
db.users.find().sort({ age: 1 });
```

- **Descending Order (बड़े से छोटे):**

javascript

CopyEdit

```
db.users.find().sort({ age: -1 });
```

32. Multikey Index क्या है?

Multikey Index का उपयोग array fields में indexing के लिए किया जाता है।

javascript

CopyEdit

```
db.users.createIndex({ tags: 1 });
```

33. findAndModify() Method क्या है?

यह एक document को update और return करता है।

javascript

CopyEdit

```
db.users.findAndModify({  
  query: { name: "Alice" },  
  update: { $set: { age: 28 } },  
  new: true  
});
```

34. readConcern क्या है?

readConcern यह निर्धारित करता है कि पढ़ी गई data की **Consistency Level** क्या होगी।

- **Local (default):**

javascript

CopyEdit


```
db.users.find().readConcern("local");
```

- **Majority:**

javascript

CopyEdit

```
db.users.find().readConcern("majority");
```

35. Capped Collection क्या है?

Capped Collection एक **Fixed-size Collection** होती है, जहां पुराना data automatic delete हो जाता है।

javascript

CopyEdit

```
db.createCollection("logs", { capped: true, size: 100000, max: 1000 });
```

36. Write Concern क्या होता है?

Write Concern यह सुनिश्चित करता है कि data कब सफलतापूर्वक store किया गया।

javascript

CopyEdit

```
db.users.insertOne({ name: "Alice" }, { writeConcern: { w: 1 } });
```

37. Hashed Index क्या है?

Hashed Index को **Sharding** के साथ उपयोग किया जाता है।

javascript

CopyEdit

```
db.users.createIndex({ email: "hashed" });
```

38. Compound Index कैसे बनाएं?

Compound Index multiple fields पर indexing करता है।

javascript

CopyEdit

```
db.users.createIndex({ name: 1, age: -1 });
```

39. Text Index कैसे बनाएं?

Text Index का उपयोग **full-text search** के लिए किया जाता है।

javascript

CopyEdit

```
db.users.createIndex({ bio: "text" });
```