MongoDB Notes & Commands

1. **show dbs**
   Shows all available databases.
   Example: show dbs

2. **use databasename**
   Switches to a database (creates it when you insert data).
   Example: use Mydata

3. **show collections**
   Displays all collections (like tables in SQL) inside the current DB.
   Example: show collections

4. **db.collection.insertOne({})**
   Inserts a single document (record).
   Example: db.Matter.insertOne({name: "Faiz", type: "emotional"})

5. **db.collection.insertMany([{},{},...])**
   Inserts multiple documents at once.
   Example:

```
db.Matter.insertMany([
 {name: "Ali", type: "Backend"},
 {name: "Sara", type: "Frontend"}
])
```

6. **db.collection.find()**
   Displays all documents in the collection.
   Example: db.Matter.find()

7. **db.collection.find().pretty()**
   Neatly displays the documents in readable format.

8. **db.createCollection("name")**
   Manually creates a collection.
   Example: db.createCollection("students")

9. **db.collection.updateOne(filter, update)**
   Updates the first matching document.
   Example: db.Matter.updateOne({name: "Faiz"}, {$set: {type: "calm"}})

10. **db.collection.updateMany(filter, update)**
    Updates all documents that match the filter.
    Example: db.Matter.updateMany({type: "Frontend"}, {$set: {type: "UI"}})

11. **db.collection.deleteOne(filter)**
    Deletes the first matching document.
    Example: db.Matter.deleteOne({name: "Faiz"})

12. **db.collection.deleteMany(filter)**
    Deletes all matching documents.
    Example: db.Matter.deleteMany({type: "UI"})

13. **db.collection.drop()**
    Deletes the entire collection.
    Example: db.Matter.drop()

14. **db.dropDatabase()**
    Deletes the current database.
    Example: db.dropDatabase()

15. **db.collection.countDocuments()**

Returns the number of documents in the collection.
Example: db.Matter.countDocuments()

16. **db.collection.distinct("field")**

Shows all unique values in a given field.
Example: db.Matter.distinct("type")

17. **db.collection.find({field: value})**

Finds documents that match the condition.
Example: db.Matter.find({type: "Frontend"})

18. **db**

Displays the current selected database name.
Example: db

19. **db.collection.find().sort({field: 1/-1})**

Sorts the results by a field.
Example:

db.Matter.find().sort({name: 1}) // A-Z

db.Matter.find().sort({name: -1}) // Z-A

20. **db.collection.findOne()**

Returns the first matching document (or first in collection if no filter).
Example: db.Matter.findOne({name: "Faiz"})

21. **db.collection.find().limit(n)**

Limits the number of results shown.

Example: db.Matter.find().limit(3)

22. **db.collection.find().skip(n)**

Skips n results — useful for pagination.

Example: db.Matter.find().skip(3).limit(3)

23. **db.collection.find({field: {$gt: value}})**

Query with greater than condition ($gt = >, $lt = <)

Example: db.students.find({marks: {$gt: 60}})

24. **db.collection.createIndex({field: 1})**

Adds an index to improve performance on queries.

Example: db.users.createIndex({name: 1})

25. **db.collection.getIndexes()**

Shows all indexes on a collection.

Example: db.users.getIndexes()

26. **db.collection.renameCollection("newName")**

Renames the collection.

Example: db.oldName.renameCollection("newName")

27. **db.stats()**

Shows stats for the current database (size, collections, etc.)
Example: db.stats()

28. **db.collection.find({field: {$in: [val1, val2]}})**

Filters data that matches any value in an array.
Example: db.users.find({type: {$in: ["Frontend", "Backend"]}})

29. **help or db.help()**

Lists useful commands and help docs.
Example: db.help()

30. **MongoDB Concepts Summary:**

SQL Concept	**MongoDB Term**
Row	Document
Table	Collection
Database	Database