Chapter-3 (MongoDB)

Database, Collection & Document

1. **Database:** Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.
2. **Collection:** Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.
3. **Document:** A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

The following table shows the relationship of RDBMS terminology with MongoDB.

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)

1. MongoDB Version:

- C:\Program Files\MongoDB\Server\5.0\bin>mongod --version

2. Start MongoDB:

- C:\Program Files\MongoDB\Server\5.0\bin>mongo

3. Clear Screen:

- > cls

4. View Databases:

- > show dbs

5. Create Database:

- > user pos

6. Drop Database:

- Use pos
- db.dropDatabase()

7. Create Collection:

- db.createCollection("student")

8. **Show Collections:**

- show collections / show tables

9. **View Data:**

- db.student.find()

10. **View Data in Json Format:**

- db.student.findOne()

11. **Drop Collection:**

- db.student.drop()

12. **Delete Data:**

- db.student.remove({})

13. **Single Insertion:**

- db.customer.insert({"name": "Faruq", "dept":"CSE", "cell":"01757584584" ,"email":"faruq@example.com"})

14. **Multiple Insertion:**

- db.customer.insert({"name":"Sayed", "dept":"BBA", "cell":"018125412142", "email":"sayed@example.com"}, {"name":"Rayan", "dept":"EEE", "cell":"016245124124", "email":"rayan@example.com"}, {"name":"Sadia", "dept":"Arch", "cell":"01721454241", "email":"sadia@example.com"})

15. **Embedded Insertion-1:**

- db.student.insert({"name": "Faruq", "cell": "0124521254", "email": "faruq@gmail.com", "courses":["CSE-1111","CSE-2319","CSE-4219"] })

16. **Embedded Insert Data-2:**

- db.student.insert({"name":"Raihan", "dept":"CSE", "cell":"0125412541", "email":"raihan@example.com", "address":[{"village":"Talebpur", "thana":"bissonath", "dist":"sylhet"}, {"house":"95B", "road":"7A", "thana":"dhanmondi", "dist":"sylhet"}]})

17. **View Inserted Data by ObjectId:**

- db.student.findOne("61c5d6f85e2439ea7b1735c1")

18. **Find Data Conditionally:**

- db.student.findOne({"name":"Faruq"})

```
{
        "_id" : ObjectId("61c5d6f85e2439ea7b1735c1"),
        "name" : "Raihan",
        "dept" : "CSE",
        "cell" : "0125412541",
        "email" : "raihan@example.com",
        "address" : [
            {
                "village" : "Talebpur",
                "thana" : "bissonath",
                "dist" : "sylhet"
            },
            {
                "house" : "95B",
                "road" : "7A",
                "thana" : "dhanmondi",
                "dist" : "sylhet"
            }
        ]
    }
```

19. Add Column:

- db.course.update({}, {$set: {"sid":0}},{multi: true})

20. Update Documents / Value:

- db.course.update({'sid':0},{$set:{'sid':'30'}})

21. Update all Documents / Value:

- db.course.update({"sid":0},{$set:{"sid":10}},{multi: true})

22. Join Query:

- db.student.aggregate([

```
  { $lookup: { from: "course", localField:"sid", foreignField:"sid", as:"stDetails" } }
 ])
```