1°} What is MongodB?
It is a database server. mongo → huge
It hase less relations.

Que 2°} Main Reason to use ?
→ 1.) Schemaless → flexibility ↑
2.) highly Scalable

Que 3°} Diff in My SQL and MongoDB

Database	Database
Table	Collection
Rows	Documents
Column	Field

Que 4°} What is JSON ?
→ Documents save in format of JSON.
BSON → Binary JSON. It convert it into Binary. It allow quick traversal as compare to JSON.

net stop Mongodb
net start MongodB

Que5 → Create DB
→ use latest_db

Create Collection

db.students / db.create collation ('student')

create document

db.students.insertOne({name: "Ank", age: 12})

JULY
AUGUST

2022 MAY

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

find() → show all the documents.

each document have unique Id

Que 6 → Nested document

→ db.students . updateOne ({ name : 'Ank' }, { $ set : { idCards : { has Pan Card : False , has Aadhar : True}}

Every document have size limit of 16 MB

Que 7 ⇒ CRUD operation

C	Insert One ({data}, {option}) InsertMany (data , option)	R	find findOne
U	Update One {filter, set} Update Many $set	D	deleteOne deleteMany json

12 SUNDAY

Que 8 → find student age < 12 ?

→ db.students . find ({ age : { $Lt : 12 }})

$ ⇒ Reserved keyword
Lte → less that equato to
gt ⇒ greater than

Que 9 → Find student age > 5 and 10 < age ?

→ db.students.find ({ age : { $ gt : 5 , $ Lt : 10 }})

db.students.find (). toArray () → give all elements

find ()	find One ()
• Cursar [ ]	• Only return One • can't apply any method.

Que 10.) Insert db. students . insert ({ name : "Ank", age : 12})
(old version)

InsertOne db. students . InserOne (—"—)
give acknowlegement

InsertMany db. students . InsertMany ([{ —"— }, {—"—}])

Que 11.) Update One

db. students . UpdateOne ({ name : "Ank" }
, { $ set : { age : 15 }}]

Update Many

db. students. update Many ({ age : 12 }, {
$ set : { age : 13 }})

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

14 TUESDAY
2022 JUNE

WEEK 25 165-200

Que 12 ⇾ Delete One

8

DeleteMany db. students. deleteMany ({ age : 13 })

9

db. students. deleteMany ({ }) JSON

10

11 Delete Database db. drop Database ( ) (all)

12 Delete Collection db. coll. name . drop ( ) (single)

1

- Id is given by default.
2 We can also give Id _id : ' '

3 - If we write {Ordered : false} in Option then it will not stop and error place
4 it print full valid doc.

5 Que 13.) What is projection ?

6 ⇾ means only want specific colum

db. student. find ({ }, { name : 1, _id:0

- Mongodb is Schema less

Datatype

```
foundedOn : new Date()  → date

Ontime : new Timestamp()  → time

typeof  db.student.findOne().name  →  string
```

Que 14.) Schema Validation

```
db.createCollection ( "student", {
        validator : {
            $ jsonSchema : {
                required : [ 'name' , 'price' ],
                properties : {
                    name : {
                        bsontype : " string ",
                        description : ' must be a str'
                    },
                    price : {
                        bsontype : 'number'
                        description : 'number must'
                    }
                }
            }
        },
        validatioAction : ' Warn'
                          'error'
```

Que 15.} Schema Validation (Change)

```
8 db.runCommand ({
        collMod : 'student',
9          validator:{
                 —"—
10 Write Concern   }
```

11 { w: <val>, j: <bool>, wtimeout: <number> }.

12 w → acknowledge   j → Done or not done   n amout of time to wait

1 Atomicity is achieved in document level.

2 Que 16.} Comparison Operator

3 $ eq $ ne   equal
$ lt $ gt   less than
4 $ lte $ gte   less than equal
$ in $ nin   in   { $ in : [5, 12] } })

5

Que 17.} logical operator

6

$ and
$ or
$ nor
$ not

JULY 2022

$ exi
$ typ
8 Que18-)
9 $ rec
10 $ ti
$ al
11 $ mo
$
12
. sor
1 . ski
2 {$ i
3 upsert
4
5
6

JULY 2022

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

FRIDAY
JUNE 2022
17

WEEK 25
168-197

$exist   find ({copy : { $exist : true}})
$type

8 Ques18.) Evaluation Operator

9 $regex   /^A/   → find start with A

10 $text: { $search: " youtube "} → give all youtube search

$all → give all sequene don't matter

11 $mod → perform modulos

$

12

.sort ({ age : 1, name : 1})

1 .skip (10)

2 {$ inc : { age : 2}} → increase age by 2

3 upsert: true → if not find then insert

4

5

6