MongoDB
24/05/2025

MongoDB:
this a type of Nosql and Document base Database.

This is the way of the storing data in the local system because of security. Privacy and utilising the memory allocation so if you want access the files and data in ur understanding language for that mongo have platform like mongo compass and You can access also through the command prompt.

Layer	Format	Description
Application layer	JSON	You write/read JSON using MongoDB shell/driver
Internal conversion	BSON	MongoDB converts JSON into BSON
Storage engine	.wt files	BSON is stored inside collection-*.wt files
Recovery	Journal	Write-ahead logs to prevent data loss

#Installing Mongdb
step1 – download Mongodb latest version
step2 – set the path to environment Variable
step3 – download the mongosh
step4 – set the path to environment Variable
step4 – open cmd and type - mongod to run server
step5 – open another tab of cmd type - mongosh to work with mongodb

---

MongoDB Commands Executed Successfully

1. use firstdb
2. db.students.insertOne({name: "Avinash", age: 26 , Job: "employee"})
3. db.createCollection("teachers")
4. db.teachers.insertMany([])
5. db.teachers.find()
6. db.teachers.find().forEach(printjson)

7. db.deleteplist.insertMany([])

8. db.deleteplist.deleteOne({ product: "sheet" })

9. db.deleteplist.find({ product: "sheet" })

10. db.deleteplist.insertMany([])

creating db and creating Collections:

```
test> use db6
switched to db db6
db6> db.createCollection("teachers")
{ ok: 1 }
db6> db.teachers.insertMany([
... {
... name:"roger",
... subject:"maths"
... },
... {
... name:"anita",
... subject:"chemistry"
... },
... {
... name:"thomas",
... subject:"physics"
... },
... {
... name:"tina",
... subject:"maths"
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
.. name:"tina",
.. subject:"maths"
.. },
.. {
.. name:"jonty",
.. subject:"english"
.. }
.. ]
.. )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6833247415f90fb0496c4bd0'),
    '1': ObjectId('6833247415f90fb0496c4bd1'),
    '2': ObjectId('6833247415f90fb0496c4bd2'),
    '3': ObjectId('6833247415f90fb0496c4bd3'),
db6>
  }

db6>
```

1.find()

```
db6>
  }
{
db6> db.teachers.find()
[
  {
    _id: ObjectId('6833247415f90fb0496c4bd0'),
    name: 'roger',
    subject: 'maths'
  },
  {
    _id: ObjectId('6833247415f90fb0496c4bd1'),
    name: 'anita',
    subject: 'chemistry'
  },
  {
    _id: ObjectId('6833247415f90fb0496c4bd2'),
    name: 'thomas',
    subject: 'physics'
```

 # operations:

deleteOne() , remove() and deleteMany()

```
  }
]
db6> db.teachers.deleteOne( { name: "tina" } )
{ acknowledged: true, deletedCount: 1 }
db6> db.teachers.remove( { name: "roger" } )
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
db6> db.teachers.find()
[
  {
    _id: ObjectId('6833247415f90fb0496c4bd1'),
    name: 'anita',
    subject: 'chemistry'
  },
  {
    _id: ObjectId('6833247415f90fb0496c4bd2'),
    name: 'thomas',
    subject: 'physics'
  },
  {
```

```
]
db6> db.deleteplist.insertMany([
...     { product: "register", quantity: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
...     { product: "book", quantity: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
...     { product: "sheet", quantity: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
...     { product: "timetable", quantity: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
...     { product: "envelop", quantity: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68332fb615f90fb0496c4bd5'),
    '1': ObjectId('68332fb615f90fb0496c4bd6'),
    '2': ObjectId('68332fb615f90fb0496c4bd7'),
    '3': ObjectId('68332fb615f90fb0496c4bd8'),
    '4': ObjectId('68332fb615f90fb0496c4bd9')
```

```
db6> db.deleteplist.deleteMany( { tags: "red" } )
{ acknowledged: true, deletedCount: 3 }
db6> db.deleteplist.find()
[
  {
    _id: ObjectId('68332fb615f90fb0496c4bd9'),
    product: 'envelop',
    quantity: 45,
    tags: [ 'blue' ],
    dim_cm: [ 10, 15.25 ]
  }
```

Update()

```
    _id: ObjectId('6833247415f90fb0496c4bd4'),
    name: 'jonty',
    subject: 'english'
  }
]
db6> db.teachers.updateOne({name: "anita"} ,{$set:{ subject: "drawing"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
db6> db.teachers.find()
[
  {
    _id: ObjectId('6833247415f90fb0496c4bd1'),
    name: 'anita',
    subject: 'drawing'
  },
```

Aggregate function:

1.upsert :
2.$sort:
3.$match
4.count()
5.distinct()
6.$Group:
7.$merge:
8.$project:

Pipeline()

```
db6> db.aggregateproduct.insertMany( [
...    { product: "register", quantity: 25, size: { height: 14, weight: 21, unit: "cm" }, status: "A" },
...    { product: "book", quantity: 50, size: { height: 8.5, weight: 11, unit: "in" }, status: "A" },
...    { product: "sheet", quantity: 100, size: { height: 8.5, weight: 11, unit: "in" }, status: "D" },
...    { product: "timetable", quantity: 75, size: { height: 22.85, weight: 30, unit: "cm" }, status: "D" },
...    { product: "envelop", quantity: 45, size: { height: 10, weight: 15.25, unit: "cm" }, status: "A" },
...    { product: "register", quantity: 25, size: { height: 14, weight: 21, unit: "cm" }, status: "A" },
...    { product: "book", quantity: 50, size: { height: 8.5, weight: 11, unit: "in" }, status: "A" },
...    { product: "sheet", quantity: 100, size: { height: 8.5, weight: 11, unit: "in" }, status: "D" },
...    { product: "timetable", quantity: 75, size: { height: 22.85, weight: 30, unit: "cm" }, status: "D" },
...    { product: "envelop", quantity: 45, size: { height: 10, weight: 15.25, unit: "cm" }, status: "A" },
...    { product: "register", quantity: 25, size: { height: 14, weight: 21, unit: "cm" }, status: "A" },
...    { product: "book", quantity: 50, size: { height: 8.5, weight: 11, unit: "in" }, status: "A" }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('683332c915f90fb0496c4bda'),
    '1': ObjectId('683332c915f90fb0496c4bdb'),
    '2': ObjectId('683332c915f90fb0496c4bdc'),
```

$match & $group:

```
db6> db.aggregateproduct.aggregate([{ $match: { status: "A" } } , { $group: { _id: "$product", total: { $sum: "$quantity" } } }])
[
  { _id: 'book', total: 150 },
  { _id: 'envelop', total: 90 },
  { _id: 'register', total: 75 }
]
db6>
```

$sort:

```
db6> db.aggregateproduct.aggregate([{ $match: { status: "A" } } , { $group: { _id: "$product", total: { $sum: "$quantity" } } } , {$sort : { total : 1
 } } ])
[
  { _id: 'register', total: 75 },
  { _id: 'envelop', total: 90 },
  { _id: 'book', total: 150 }
]
db6>
```

$project:

```
db6> db.aggregateproduct.aggregate([ { $project : { product : 1,quantity : 1 } } , { $group: { _id: "$product", total: { $sum: "$quantity" } } } , {$s
ort : { total : 1 }}])
[
  { _id: 'register', total: 75 },
  { _id: 'envelop', total: 90 },
  { _id: 'timetable', total: 150 },
  { _id: 'book', total: 150 },
  { _id: 'sheet', total: 200 }
]
db6>
```

$upsert: It checks if the data is present—if it is, it updates the existing document with the new data; if not, it creates a new document in the collection.

```
db6> db.aggregateproduct.updateOne({ product: "pen" },{$set:{quantity:30,size:{height:5,weight:2,unit:"cm"},status:"A"}},{upsert:true})
{
  acknowledged: true,
  insertedId: ObjectId('6833485508ed0a546150b407'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
db6> db.aggregateproduct.find({product:"pen"})
[
  {
    _id: ObjectId('6833485508ed0a546150b407'),
    product: 'pen',
    quantity: 30,
    size: { height: 5, weight: 2, unit: 'cm' },
    status: 'A'
  }
]
db6>
```


$merge:

```
db6> db.aggregateproduct.aggregate([{$group: {_id: "$product", totalQuantity: { $sum: "$quantity" }}}, { $merge: "mergedProductSummary"}])

db6> db.aggregateproduct.find()
[
```



```
db6> db.mergedProductSummary.find().pretty()
[
  { _id: 'pen', totalQuantity: 30 },
  { _id: 'book', totalQuantity: 150 },
  { _id: 'sheet', totalQuantity: 200 },
  { _id: 'envelop', totalQuantity: 90 },
  { _id: 'register', totalQuantity: 75 },
  { _id: 'timetable', totalQuantity: 150 }
]
db6>
```


Count()

```
db6> db.aggregateproduct.countDocuments({ status: "A" })
9
db6> db.aggregateproduct.countDocuments()
13
db6>
```

Distinct()

```
test> use db6
switched to db db6
db6> db.aggregateproduct.distinct("status")
[ 'A', 'D' ]
db6> db.aggregateproduct.distinct("product")
[ 'book', 'envelop', 'pen', 'register', 'sheet', 'timetable' ]
db6>
```

limit()

```
db6> db.indexcollection.find({ status: "A" }).limit(2)
[
  {
    _id: ObjectId('68334fc648905b54866c4bd1'),
    product: 'book',
    status: 'A',
    size: { height: 8.5, weight: 11, unit: 'in' },
    instock: [ { warehouse: 'C', quantity: 5 } ],
    creationts: ISODate('2016-11-01T12:30:15.000Z')
  },
  {
    _id: ObjectId('68334fc648905b54866c4bd6'),
    product: 'register',
    status: 'A',
    size: { height: 8.5, weight: 11, unit: 'in' },
    instock: [ { warehouse: 'C', quantity: 5 } ],
    creationts: ISODate('2016-11-01T12:30:15.000Z')
  }
]
db6>
```

Indexing and Regex

```
db6> db.indexcollection.getIndexes()
db6> db.db.indexcollection.createIndex( { product: 1 } )
product_1
db6> db.indexcollection.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
db6> db.indexcollection.createIndex( { status : 1 , "size.height": 1 } )
status_1_size.height_1
db6> db.indexcollection.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { status: 1, 'size.height': 1 },
    name: 'status_1_size.height_1'
  }
]
db6> db.indexcollection.createIndex( { "$**": "text" } )
$**_text
db6> db.indexcollection.createIndex( { product: "text" , status : "text" } )
```

```
db6> db.indexcollection.find({product: { $regex: /^reg/, $options: "i"} })
[
  {
    _id: ObjectId('68334fc648905b54866c4bd0'),
    product: 'register',
    status: 'A',
    size: { height: 14, weight: 21, unit: 'cm' },
    instock: [ { warehouse: 'A', quantity: 5 } ],
    creationts: ISODate('2014-11-01T12:30:15.000Z')
  },
  {
    _id: ObjectId('68334fc648905b54866c4bd5'),
    product: 'register',
    status: 'A',
    size: { height: 14, weight: 21, unit: 'cm' },
    instock: [ { warehouse: 'A', quantity: 5 } ],
    creationts: ISODate('2014-11-01T12:30:15.000Z')
  },
```

```
  {
    _id: ObjectId('68334fc648905b54866c4bd6'),
    product: 'register',
    status: 'A',
    size: { height: 8.5, weight: 11, unit: 'in' },
    instock: [ { warehouse: 'C', quantity: 5 } ],
    creationts: ISODate('2016-11-01T12:30:15.000Z')
  },
  {
    _id: ObjectId('68334fc648905b54866c4bd7'),
    product: 'register',
    status: 'D',
    size: { height: 8.5, weight: 11, unit: 'in' },
    instock: [ { warehouse: 'A', quantity: 60 } ],
    creationts: ISODate('2017-11-01T12:30:15.000Z')
  }
]
db6>
```

```
db6> db.indexcollection.find({product: { $regex: /^book/, $options: "i" }})
[
  {
    _id: ObjectId('68334fc648905b54866c4bd1'),
    product: 'book',
    status: 'A',
    size: { height: 8.5, weight: 11, unit: 'in' },
    instock: [ { warehouse: 'C', quantity: 5 } ],
    creationts: ISODate('2016-11-01T12:30:15.000Z')
  },
  {
    _id: ObjectId('68334fc648905b54866c4bd8'),
    product: 'book',
    status: 'D',
    size: { height: 22.85, weight: 30, unit: 'cm' },
    instock: [ { warehouse: 'A', quantity: 40 } ],
    creationts: ISODate('2018-11-01T12:30:15.000Z')
  },
```

Thank You