

Class 1 : Introduction to MongoDB

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently. It is categorized under the NoSQL database because the storage and retrieval of data in the MongoDB are not in the form of tables.

- Structured Data:

The information is typically organized in a specific format, often using tables with rows and columns. This makes it easier to search, filter, and analyze the data.

- Database Management System (DBMS):

This is the software that acts like the filing cabinet manager. It allows you to store, retrieve, update, and manage all the data within the database.

- Data Types:

Databases can hold various kinds of information, including text, numbers, images, videos, and more.

- Applications

Web Application

Big Data

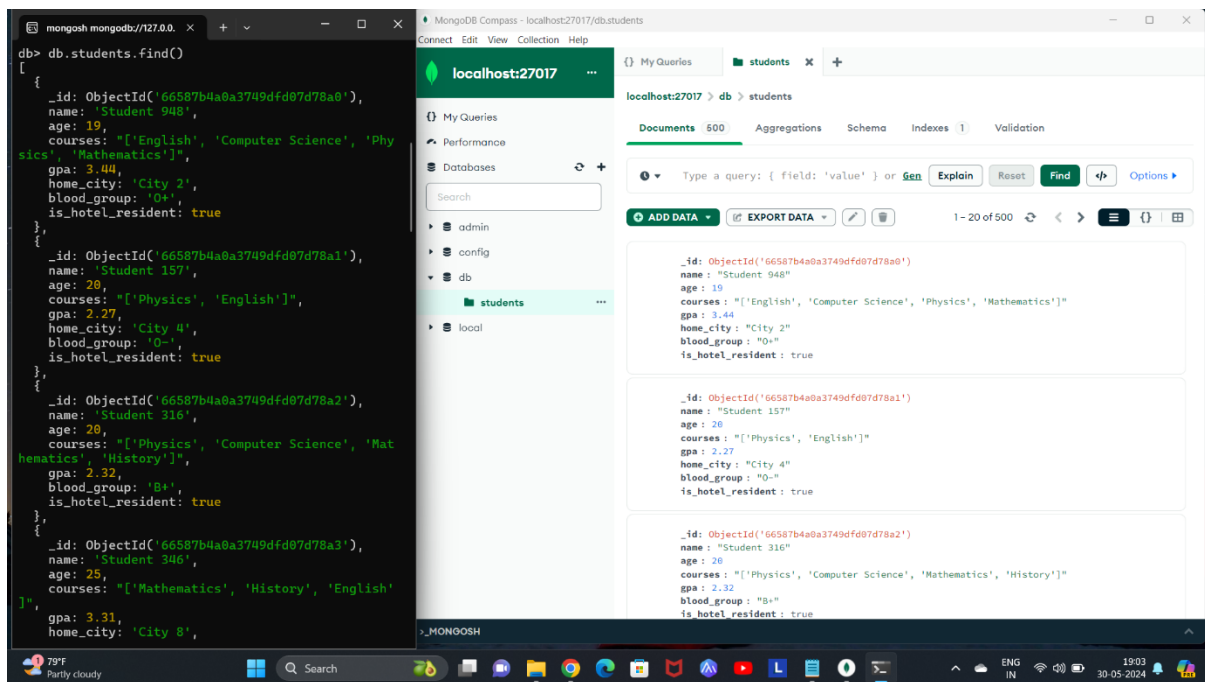
Demographic and Biometric Data

Synchronization

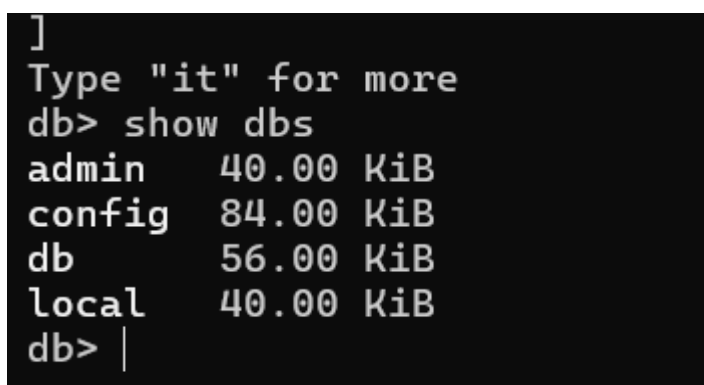
Class 2 : Add ,Update and Delete

To find the data present in the collections ,we can use the command “db.collection_name.find()” .

In this the collection name is “students”.



“show dbs” command shows all the database.



- Collections :

A collection is a group of documents.

If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

- Database:

MongoDB groups collections into databases.

A single instance of MongoDB can host several databases, each grouping together zero or more collections.

- Document:

At the heart of MongoDB is the document:

an ordered set of keys with associated values.

The representation of a document varies by programming language, but most languages have a data structure that is a natural fit, such as a map, hash, or dictionary.

Class 3 : WHERE,AND,OR & CRUD

- WHERE:

WHERE is used when we want to filter a subset based on a condition.

```
db> db.students.find({ gpa:{$gt:3}})
[
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a0'),
    name: 'Student 948',
    age: 19,
    courses: "['English', 'Computer Science', 'Physics', 'Mathematics']",
    gpa: 3.44,
    home_city: 'City 2',
    blood_group: 'O+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a3'),
    name: 'Student 346',
    age: 25,
    courses: "['Mathematics', 'History', 'English']",
    gpa: 3.31,
    home_city: 'City 8',
    blood_group: 'O-',
    is_hotel_resident: true
  },
]
```

We use the condition gpa greater than 3. The result is shown is based on this condition.

- **AND:**

AND is used when, in a given collection we want to filter a subset based on multiple conditions.

```
Type "it" for more
db> db.students.find({
... $and:[
... {home_city:"City 1"},
... {blood_group:"O-"}
... ]
... })
[
  {
    _id: ObjectId('66587b4a0a3749dfd07d78c0'),
    name: 'Student 384',
    age: 18,
    courses: "['Mathematics', 'Computer Science']"
  },
  {
    gpa: 3.9,
    home_city: 'City 1',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('66587b4a0a3749dfd07d7950'),
    name: 'Student 702',
    age: 22,
    courses: "['History', 'Mathematics', 'English']"
  },
  {
    gpa: 3.74,
    home_city: 'City 1',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {

```

home_city : City1' and 'blood_group : O-' are the conditions on which it is filtered.

- **OR:**

OR is used when , in a given collection we want to filter a subset based on multiple conditions but any one is sufficient .

```
db> db.students.find({
... $or:[
... {blood_group:"O+"},
... {gpa:{$lt:3.5}}
... ]
... })
[
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a0'),
    name: 'Student 948',
    age: 19,
    courses: "['English', 'Computer Science', 'Physics', 'Mathematics']",
    gpa: 3.44,
    home_city: 'City 2',
    blood_group: 'O+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a1'),
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
]
```

Here, the students database is filtered based on either 'blood_group : O+' or 'gpa less than 3.5'.

- **CRUD:**

C – Create / Insert

R – Remove

U – Update

D – Delete

This is applicable for a collection (table) or a document (row)

- **Update:**

‘\$set’ command is used to update the data present in the collection.

```
db> db.students.updateOne( { name:"Sam" } , { $set:{  
gpa:3} } )  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
db> |
```

- **Delete:**

It is used to delete the data present in the collection.

```
db> db.students.deleteOne({ name:"Sam" })  
{ acknowledged: true, deletedCount: 1 }  
db> |
```

- Insert:

```
db> const studentData = {  
...  "name": "Sam",  
...  "age": 22,  
...  "courses": ["Computer Science" , "Mathematics"]  
,  
...  "gpa": 3.4,  
...  "home_city": "City 3",  
...  "blood_group": "B+",  
...  "is_hotel_resident": false  
...  }  
  
db> db.students.insertOne(studentData)  
{  
  acknowledged: true,  
  insertedId: ObjectId('6658a0c70cce0c5ec1cdcdf6')  
}  
db> |
```

Here we are inserting the student details name 'Sam' and other information to the collection 'students'.the insertion is done one time.

- Projection:

This is used when we don't need all columns / attributes.

- Benefits of Projection:

- ✓ Reduced data transferred between the database and your application.
- ✓ Improves query performance by retrieving only necessary data.
- ✓ Simplifies your code by focusing on the specific information you need.


```
db> db.students.deleteOne({ name:"Sam" })
{ acknowledged: true, deletedCount: 1 }
db> db.students.find({}, {name:1 , gpa:1 })
[
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a0'),
    name: 'Student 948',
    gpa: 3.44
  },
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a1'),
    name: 'Student 157',
    gpa: 2.27
  },
  {
    _id: ObjectId('66587b4a0a3749dfd07d78a2'),
    name: 'Student 316',
    gpa: 2.32
  }
]
```

Here it only shows the name and gpa . Because the command is give as 'name:1' and 'gpa:1'.