CLASS 1 :- INTRODUCTION TO MONGODB

MONGODB DOWNLOAD :-

1. Go to the MongoDB website:- Visit the official MongoDB website at https://www.mongodb.com/.

2. Navigate to the Download section: -Look for the "Products" menu on the top of the website and select "MongoDB Server". This will take you to the download page.

3. Choose the version:- MongoDB offers both the Community Server (free and open-source) and the Enterprise Server (commercial version). Select the version that suits your needs. For most users, the Community Server is sufficient.

4. Select the appropriate package:- MongoDB supports various operating systems including Windows, macOS, and Linux. Choose the package that matches your operating system.

5. Download the installer:- Click on the download button to initiate the download of the installer package.

6. Install MongoDB:- Once the download is complete, run the installer and follow the installation instructions provided.

7. Set up MongoDB: -After installation, you may need to configure MongoDB depending on your specific requirements. You can refer to the MongoDB documentation for detailed setup and configuration instructions.

WHAT IS DATABASE

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.

CLASS 2 :-ADD, UPDATE & DELETE

- Download the student csv file and import the data to the collection
- We can able to see the uploaded data in mongodb compass

Few commands to test after connection

1. Show dbs :-

If you want to see a list of databases in MongoDB, you can use the `show dbs` command. However, it's important to note that this command will only display databases that have data in them. If a database doesn't contain any collections, it might not show up in the list.

Here's how you can use it in the MongoDB shell:

```
> show dbs

admin        40.00kB

config       72.00kB

local        40.00kB

sample_db    40.001kB
```

This will display a list of databases along with their sizes. The size displayed is the total storage size of the database.

Remember that the `show dbs` command is only available in the MongoDB shell. If you're using a MongoDB client or driver in a programming language, you'll typically use methods provided by the client to list databases and perform other operations.

2. Use db :-

To switch to a specific database in MongoDB, you can use the `use` command followed by the name of the database you want to switch to.

Here's how you can do it in the MongoDB shell:

```
> use yourDatabaseName

switched to db yourDatabaseName
```

Replace `yourDatabaseName` with the name of the database you want to switch to. If the specified database doesn't exist, MongoDB will create it for you when you first write data to it.

3. Show collections:-

To list the collections in the currently selected database in MongoDB, you can use the `show collections` command in the MongoDB shell.

Here's how you can do it in MongoDB shell

```
> show collections
```

This command will display a list of collections within the currently selected database. If you haven't selected a database yet, you can first use the `use` command to switch to the desired database, and then use `show collections`.

4. insertone():-

Insert a record to collection. Create collection if not exists

Here's how you can use "insertOne()"to achieve the same result in the MongoDB shell:

```
db.foo.insert({"bar": "baz"})
```

5. insertmany():-

Insert the more then one document

Here's how you can use "insertmany()"

```
db.foo.insertMany([
{"bar": "baz1"},
{"bar": "baz2"},
// Add more documents as needed
])
```

6. find() :-

To retrieve documents from a collection in MongoDB, you can use the "find()" method. This method returns a cursor to the documents that match the query criteria.

Here's how you can use it:

```
db.collectionName.find()
```

here,collectionName means with the name of the collection you want to guery

7. Remove() :-

Used to Remove the collection table

Here how you can use it:

```
db.collectionName.remove()
```

DOCUMENTS, COLLECTION, DATABASE

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.

```
{"greeting" : "Hello, world!"}
```

COLLECTION:-

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.

A database has its own permissions, and each database is stored in separate files on disk.

DATATYPE:-

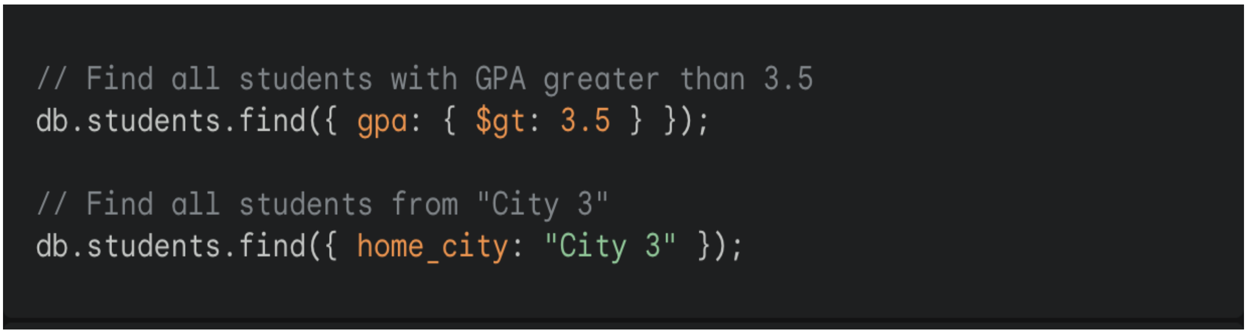
Basically each document will be in JSON format which will be as follows. Where each attributes inside can be of multiple data types

CLASS 3:Where, AND,OR & CRUD

WHERE:-

Given a collection you want to filter a subset based on a condition. That is the place WHERE is used

Here how you can do it in MongoDB shell:-



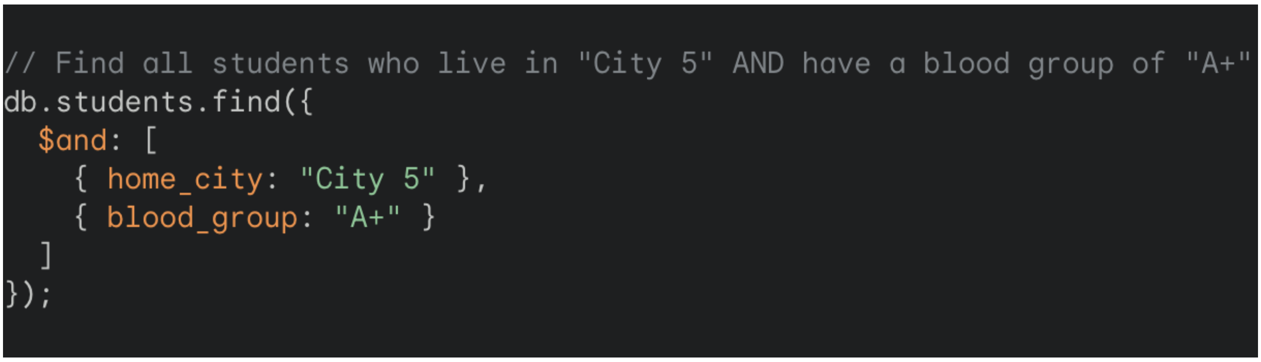
```
// Find all students with GPA greater than 3.5
db.students.find({ gpa: { $gt: 3.5 } });

// Find all students from "City 3"
db.students.find({ home_city: "City 3" });
```

MongoDB provides various comparison operators, such as `$gt` (greater than), `$lt` (less than), `$gte` (greater than or equal to), `$lte` (less than or equal to), `$ne` (not equal), and `$in` (matches any of the values specified in an array)

AND:-

Given a collection you want to FILTER a subset based on multiple conditions

Here how you can do it in MongoDB shell:-



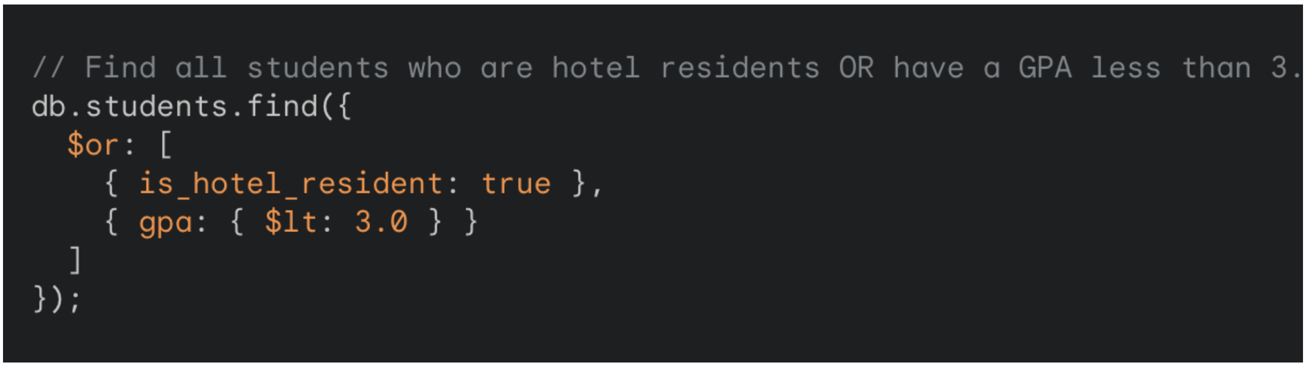
```
// Find all students who live in "City 5" AND have a blood group of "A+"
db.students.find({
  $and: [
    { home_city: "City 5" },
    { blood_group: "A+" }
  ]
});
```

The `$and` operator is not necessary in this case because the `find` method implicitly uses `$and` when you specify multiple conditions at the same level.

OR:-

Given a collection you want to filter a subset based on multiple conditions but any one is sufficient

Here how you can do it in MongoDB shell:-



```
// Find all students who are hotel residents OR have a GPA less than 3.
db.students.find({
  $or: [
    { is_hotel_resident: true },
    { gpa: { $lt: 3.0 } }
  ]
});
```

In this above example, the students database is filtered based on either “hotel_resident: true” or “gpa is less than 3.0”

If you want to use an "or" condition in MongoDB, you use the `$or` logical operator. This allows you to query documents that match at least one of the specified conditions.

CRUD:-

C:-Create/Insert

R:-Remove

U:-Update

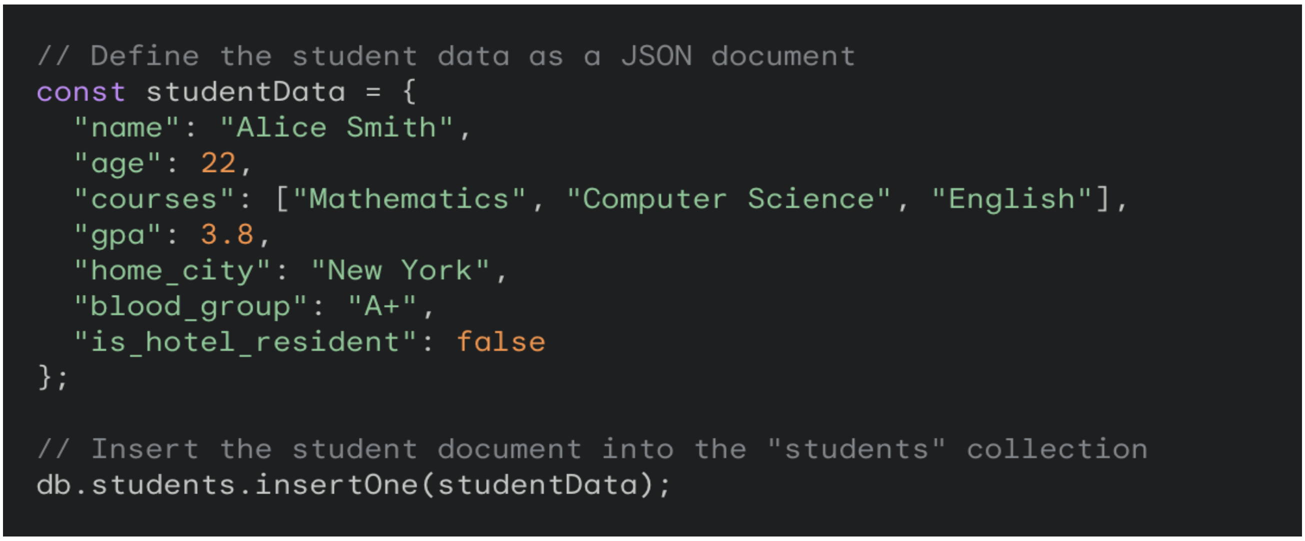
D:-Delete

This is applicable for a Collection (Table) or a Document (Row)

INSERT:-

We can insert the single document and also multiple document into a collection

Here how you can do it in MongoDB shell:-

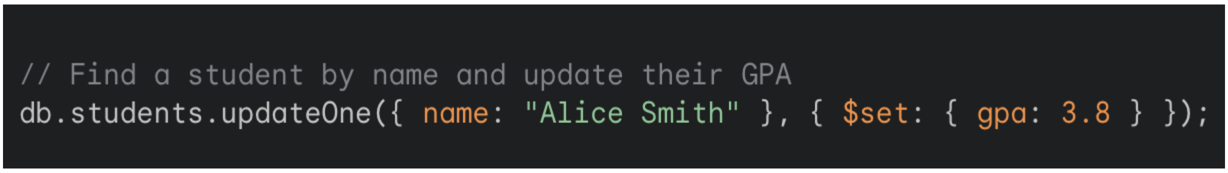


```
// Define the student data as a JSON document
const studentData = {
  "name": "Alice Smith",
  "age": 22,
  "courses": ["Mathematics", "Computer Science", "English"],
  "gpa": 3.8,
  "home_city": "New York",
  "blood_group": "A+",
  "is_hotel_resident": false
};

// Insert the student document into the "students" collection
db.students.insertOne(studentData);
```

In this above example ,single student document is insert

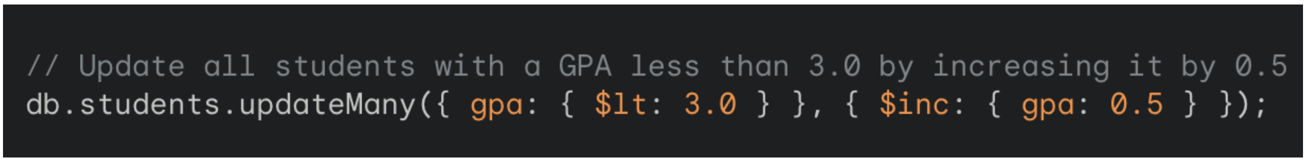
UPDATE:-

Here how you can do it in MongoDB shell:-



```
// Find a student by name and update their GPA
db.students.updateOne({ name: "Alice Smith" }, { $set: { gpa: 3.8 } });
```

In this above example we can able to “updateOne”

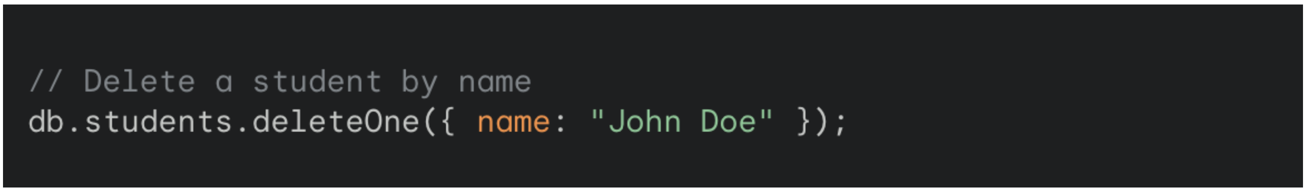


```
// Update all students with a GPA less than 3.0 by increasing it by 0.5
db.students.updateMany({ gpa: { $lt: 3.0 } }, { $inc: { gpa: 0.5 } });
```

In this above example we can able to update many time “updateMany”

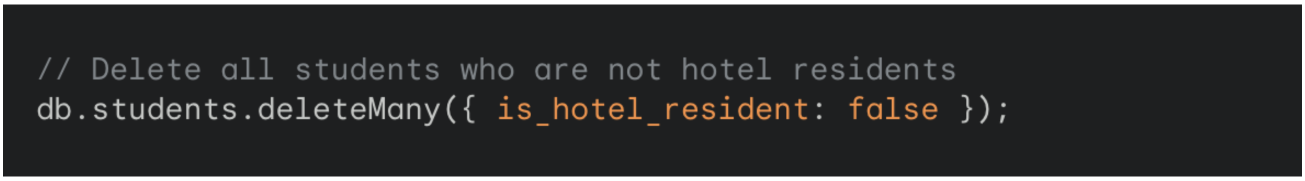
DELETE:-

Here how you can do it in MongoDB shell:-



```
// Delete a student by name
db.students.deleteOne({ name: "John Doe" });
```

In this above example we can able to “deleteonce”



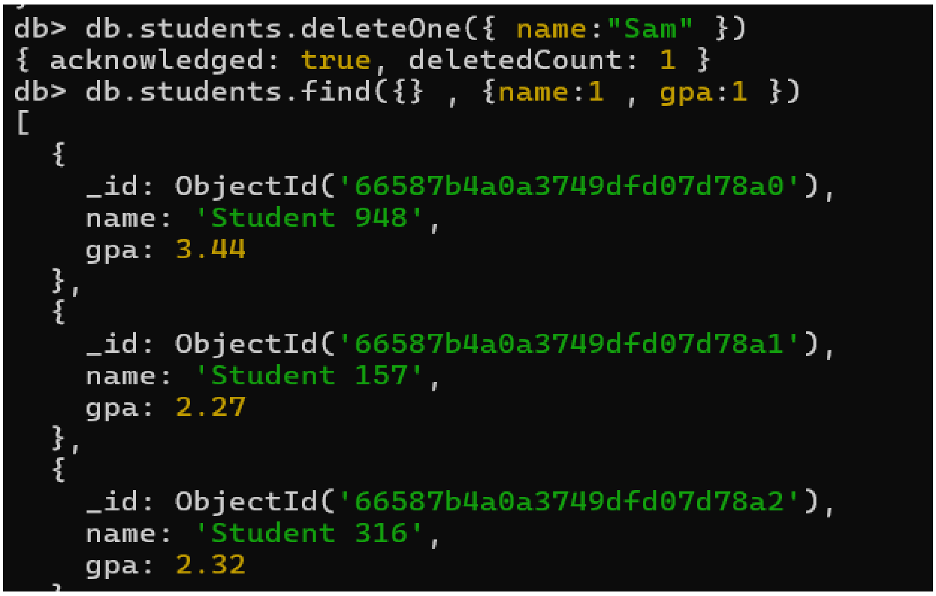
```
// Delete all students who are not hotel residents
db.students.deleteMany({ is_hotel_resident: false });
```

In this above example we can able to delete many time “deleteMany”

CLASS 4 : PROJECTION,LIMIT & SELECTORS

PROJECTION:-

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



```
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’

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.

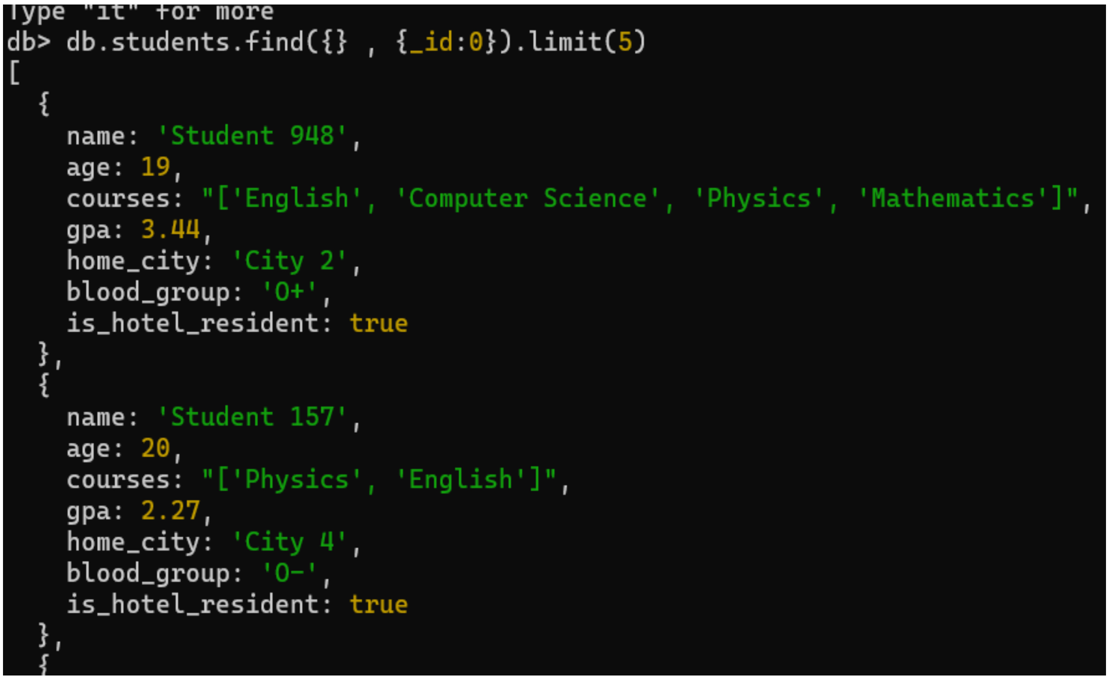
LIMIT:-

- The limit operator is used with the find method.
- It's chained after the filter criteria or any sorting operations.

  Syntax:

  db.collection.find({filter}, {projection}).limit(number)

Here how you can do it in MongoDB shell:-



```
Type "it" for more
db> db.students.find({} , {_id:0}).limit(5)
[
  {
    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
  },
  {
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
```

To get only first 5 document we use limit(5).

SELECTORS:-

- Comparison gt and lt
- AND operator
- OR operator