

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.

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
```

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 than 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 query

7. Remove() :-

Used to Remove the collection table

Here how you can use it:

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

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. For queries that cannot be done any other way, there are "\$where" clauses, which allow you to execute arbitrary JavaScript as part of your query. This allows you to do (almost) anything within a query. For security, use of "\$where" clauses should be highly restricted or eliminated. End users should never be allowed to execute arbitrary "\$where" clauses.

```
test> db.stu.find({gpa:{$gt:3.5}}).count();  
124
```

```
test> db.stu.find({home_city:"City 3"}).count();  
34
```

AND :-

Given a Collection you want to FILTER a subset based on multiple conditions. This operator is used to perform logical AND operation on the array of one or more expressions and select or retrieve only those documents that match all the given expression in the array

- This operator performs short-circuit evaluation.
- If the first expression of \$and operator evaluates to false, then MongoDB will not evaluate the remaining expressions in the array.
- You can also use AND operation implicitly with the help of comma(,).

Syntax:

The `$and` has the following syntax:

```
{ $and: [ { <expression1> }, { <expression2> }, ... , { <expressionN> } ] }
```

```

test> db.stu.find({
...   $and:[
...     {home_city:"City 5"},
...     {blood_group:"A+"}
...   ]
... });
[
  {
    _id: ObjectId('6655e91dee1dcfb73e7398db'),
    name: 'Student 142',
    age: 24,
    courses: "['History', 'English', 'Physics', 'Computer Science']",
    gpa: 3.41,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6655e91eee1dcfb73e7399fb'),
    name: 'Student 947',
    age: 20,
    courses: "['Physics', 'History', 'English', 'Computer Science']",
    gpa: 2.86,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6655e91eee1dcfb73e739a6d'),
    name: 'Student 567',
    age: 22,
    courses: "['Computer Science', 'History', 'English', 'Mathematics']",
    gpa: 2.01,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: true
  }
]

```

OR :-

Given a Collection you want to FILTER a subset based on multiple conditions but Any One is Sufficient.

- You can use this operator in methods like find(), update(), etc. according to your requirements.
- You can also use this operator with text queries, GeoSpatial queries, and sort operations.
- When MongoDB evaluating the clauses in the \$or expression, it performs a collection scan.

Syntax:

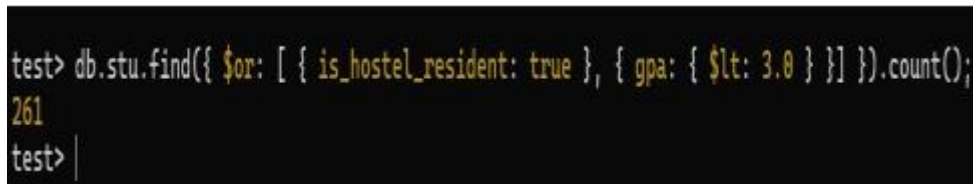
The `$or` operator has the following syntax:

```
{ $or: [ { <expression1> }, { <expression2> }, ... , { <expressionN> } ] }
```

Consider the following example:

```
db.stu.find({ $or: [{ is_hostel_resident: true }, { gpa: 3.0 } ] }).count();
```

This query will select all documents in the `inventory` collection where either the `quantity` field value is less than `20` or the `price` field value equals `10`.



```
test> db.stu.find({ $or: [ { is_hostel_resident: true }, { gpa: { $lt: 3.0 } } ] }).count();
261
test> |
```

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”.