CRUD OPERATIONS

C→CREATION / INSERT

R→REMOVE

U→UPDATE

D→DELETE

CREATION / INSERT : Inserts are the basic method for adding data to MongoDB. To insert a document into a collection, use the collection's insert method.

```
test> 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
...     };

test> db.stu.insertOne(studentData);
{
  acknowledged: true,
  insertedId: ObjectId('665b529e49389824aecdcdf7')
}
test> |
```

INSERT ONE() : In MongoDB, insertOne() method inserts a document into the

collection. This method inserts only one document at a time.

```
test> db.stu.insertOne(studentData);
{
  acknowledged: true,
  insertedId: ObjectId('665b529e49389824aecdcdf7')
}
test> db.stu.insertOne({"name":"Bhagya"});
{
  acknowledged: true,
  insertedId: ObjectId('665b549349389824aecdcdf8')
}
test> db.stu.find({"name":"Bhagya"});
[ { _id: ObjectId('665b549349389824aecdcdf8'), name: 'Bhagya' } ]
test> db.stu.find().count();
505
test> |
```

INSERT MANY(): The insertMany() method inserts one or more documents in the collection. It takes array of documents to insert in the collection.

REMOVE : The remove() method removes documents from the database. It can remove one or all documents from the collection that matches the given query expression. If you pass an empty document({}) in this method, then it will remove all documents from the specified collection. It takes four parameters and returns an object that contains the status of the operation.

UPDATE : The update() method in MongoDB updates a document or multiple documents in the collection.This method can be used for a single updating of documents as well as multiple documents. By default, the db.collection.update() method updates a single document. To update all documents that match the given query. Include the option "multi: true". *Note: In MongoDB, the update() method has been deprecated in favor of using updateOne() or updateMany() depending on your specific use case.*

```
test> db.stu.updateOne({name:"Alice Smith"},{$set:{gpa:3.8}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
test> |
```

DELETE : To delete documents from a MongoDB collection, you can use the following methods

deleteOne(): deletes a single document from the collection that satisfies the given criteria.

deleteMany(): deletes multiple documents from the collection that satisfy the given criteria. **remove():** deletes one or all data that matches the

search criteria. **drop():** deletes a complete collection.

dropDatabase(): deletes an entire database.

```
test> db.stu.deleteOne({name:"John Doe"});
{ acknowledged: true, deletedCount: 0 }
test>
```

UPDATE MANY() : The updateMany() method updates all the documents in MongoDB collections that match the given query.

```
test> db.stu.updateMany({gpa:{$lt:3.0}},{$inc:{gpa:0.5}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 261,
  modifiedCount: 261,
  upsertedCount: 0
}
```

DELETE : The deleteOne() method in MongoDB deletes the first document from the collection that matches the given selection criteria. It will delete/remove a single document from the collection.

```
test> db.stu.deleteMany({is_hostel_resident:false});
{ acknowledged: true, deletedCount: 1 }
test>
```

PROJECTION : MongoDB provides a special feature that is known as Projection. It allows you to select only the necessary data rather than selecting whole data from the document.

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

```
test> db.stu.find({},{name:1,gpa:1});
```

[
 {
   _id: ObjectId('6655e91dee1dcfb73e7398a4'), name:
   'Student 948', gpa: 4
 },
 {
   _id: ObjectId('6655e91dee1dcfb73e7398a5'), name:
   'Student 157', gpa: 2.77
 },
 {
   _id: ObjectId('6655e91dee1dcfb73e7398a6'), name:
   'Student 316', gpa: 2.82

```
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398a7'), name:
      'Student 346', gpa: 3.31
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398a8'), name:
      'Student 930', gpa: 3.63
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398a9'), name:
      'Student 305',
      gpa: 3.4
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398aa'), name:
      'Student 268',
      gpa: 3.98
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398ab'),
      name: 'Student 563',
      gpa: 2.75
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398ac'), name:
      'Student 440',
      gpa: 2.56
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398ad'),
      name: 'Student 536',
      gpa: 3.37
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398ae'), name:
      'Student 256',
      gpa: 3.44
    },
    {
      _id: ObjectId('6655e91dee1dcfb73e7398af'), name:
      'Student 177',
```

```
    gpa: 3.02
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b0'),
    name: 'Student 871',
    gpa: 3.2
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b1'),
    name: 'Student 487',
    gpa: 2.6
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b2'),
    name: 'Student 213',
    gpa: 2.89
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b3'),
    name: 'Student 690',
    gpa: 2.75
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b4'),
    name: 'Student 368',
    gpa: 3.91
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b5'),
    name: 'Student 172',
    gpa: 2.96
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b6'),
    name: 'Student 647',
    gpa: 3.43
  },
  {
    _id: ObjectId('6655e91dee1dcfb73e7398b7'),
    name: 'Student 232',
    gpa: 3.04
  }
```

]
Type "it" for more test>