

## Experiment No:09

### Title: MongoDB-CRUD Operations

**Aim:** Accessing MongoDB from some of the popular high-level programming Languages. Perform Create, Retrieve, and Delete or CRUD Operations.

#### Theory:

CRUD operations in MongoDB involve four basic functions: Create, Read, Update, and Delete. These operations allow interaction with MongoDB collections and documents.

Here's a breakdown of each:

#### 1. Create (Insert Operation)

In MongoDB, the `insertOne()` or `insertMany()` method is used to add new documents to a collection.

- **Syntax:**

```
db.collection.insertOne( { document } );
```

```
db.collection.insertMany( [ { document1 }, { document2 }, ... ] );
```

#### 2. Read (Query Operation)

To retrieve data from MongoDB, you use the `find()` or `findOne()` method.

- **Syntax:**

```
db.collection.find( { query } );
```

```
db.collection.findOne( { query } );
```

#### 3. Update

The `updateOne()`, `updateMany()`, or `replaceOne()` methods allow you to modify documents.

- **Syntax:**

```
db.collection.updateOne( { filter }, { $set: { field: value } } );
```

```
db.collection.updateMany( { filter }, { $set: { field: value } } );
```

#### 4. Delete

To remove documents from a collection, you use the `deleteOne()` or `deleteMany()` method.

- **Syntax:**

```
db.collection.deleteOne( { filter } );
```

```
db.collection.deleteMany( { filter } );
```

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\VAISHNAVI> **mongod --version**

db version v8.0.0

Build Info: {

  "version": "8.0.0",

  "gitVersion": "d7cd03b239ac39a3c7d63f7145e91aca36f93db6",

  "modules": [],

  "allocator": "tcmalloc-gperf",

  "environment": {

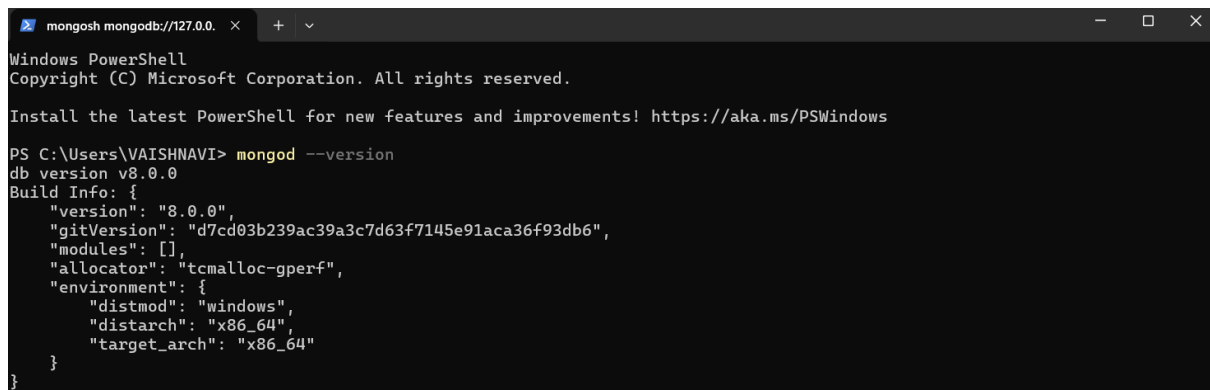
    "distmod": "windows",

    "distarch": "x86\_64",

    "target\_arch": "x86\_64"

  }

}



```
mongosh mongodb://127.0.0.1:27017
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\VAISHNAVI> mongod --version
db version v8.0.0
Build Info: {
  "version": "8.0.0",
  "gitVersion": "d7cd03b239ac39a3c7d63f7145e91aca36f93db6",
  "modules": [],
  "allocator": "tcmalloc-gperf",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

PS C:\Users\VAISHNAVI> **mongosh**

Current Mongosh Log ID: 66f3e37649202f9f6dc73bf7

Connecting to:

mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.1

Using MongoDB: 8.0.0

Using Mongosh: 2.3.1

For mongosh info see: <https://www.mongodb.com/docs/mongosh-shell>

-----

The server generated these startup warnings when booting

2024-09-25T15:33:43.437+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

```
PS C:\Users\VAISHNAVI> mongosh
Current Mongosh Log ID: 66f3e37649202f9f6dc73bf7
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.1
Using MongoDB:      8.0.0
Using Mongosh:       2.3.1

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-09-25T15:33:43.437+05:30: Access control is not enabled for the database. Read and write access to data and conf
```

test> show dbs

admin 40.00 KiB

config 48.00 KiB

local 72.00 KiB

```
test> show dbs
admin 40.00 KiB
config 48.00 KiB
local 72.00 KiB
test> use Demo
switched to db Demo
Demo> |
```

test> use Demo

switched to db Demo

```
test> use Demo
switched to db Demo
Demo> |
```

Demo> db.createCollection("Students")

{ ok: 1 }

```
test> use Demo
switched to db Demo
Demo> db.createCollection("Students")
{ ok: 1 }
Demo> |
```

Demo> db.Students.insertOne({ name:"ABC", branch:"AIML", roll\_no:04})

```
{
  acknowledged: true,
  insertedId: ObjectId('66f3e5c849202f9f6dc73bf8')
}
```

```
Demo> db.Students.insertOne({ name:"ABC", branch:"AIML", roll_no:04})
{
  acknowledged: true,
  insertedId: ObjectId('66f3e5c849202f9f6dc73bf8')
}
Demo> |
```

```
Demo> db.students.insertMany([ {name:"Vaishu",
branch:"AIML",roll_no:70}, {name:"Anish",branch:"AIML",
roll_no:05}, {name:"Ashfiya",branch:"Entc",roll_no:03} ])
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66f3e8bc49202f9f6dc73bf9'),
    '1': ObjectId('66f3e8bc49202f9f6dc73bfa'),
    '2': ObjectId('66f3e8bc49202f9f6dc73bfb')
  }
}
```

```
Demo> db.students.insertMany([ {name:"Vaishu", branch:"AIML",roll_no:70}, {name:"Anish",branch:"AIML", roll_no:05}, {name:"Ashfiya",branch:"Entc",roll_no:03} ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66f3e8bc49202f9f6dc73bf9'),
    '1': ObjectId('66f3e8bc49202f9f6dc73bfa'),
    '2': ObjectId('66f3e8bc49202f9f6dc73bfb')
  }
}
Demo> db.collection.find().pretty()
```

```
Demo> db.collection.find().pretty()
```

```
Demo> db.Students.find().pretty()
```

```
[
  {
    _id: ObjectId('66f3e5c849202f9f6dc73bf8'),
    name: 'ABC',
    branch: 'AIML',
    roll_no: 4
  }
]
```

```
Demo> db.Students.find().pretty()
[
  {
    _id: ObjectId('66f3e5c849202f9f6dc73bf8'),
    name: 'ABC',
    branch: 'AIML',
    roll_no: 4
  }
]
```

Demo> **db.students.find({roll\_no:05}).pretty()**

```
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfa'),
    name: 'Anish',
    branch: 'AIML',
    roll_no: 5
  }
]
```

```
Demo> db.students.find({roll_no:05}).pretty()
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfa'),
    name: 'Anish',
    branch: 'AIML',
    roll_no: 5
  }
]
```

Demo> **db.students.updateOne({name:"Ashfiya"},{\$set:{branch:"AIML"}})**

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
}
```

```
Demo> db.students.updateOne({name:"Ashfiya"},{$set:{branch:"AIML"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
Demo> db.students.find({roll_no:03}).pretty()
```

```
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfb'),
    name: 'Ashfiya',
    branch: 'AIML',
    roll_no: 3
  }
]
```

```
Demo> db.students.find({roll_no:03}).pretty()
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfb'),
    name: 'Ashfiya',
    branch: 'AIML',
    roll_no: 3
  }
]
```

```
Demo> db.students.deleteOne({name:"Ashfiya"})
```

```
{ acknowledged: true, deletedCount: 1 }
```

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

```
Demo> db.students.find().pretty()
```

```
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bf9'),
    name: 'Vaishu',
    branch: 'AIML',
  }
]
```

```
    roll_no: 70
  },
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfa'),
    name: 'Anish',
    branch: 'AIML',
    roll_no: 5
  }
]
```

```
Demo> db.students.find().pretty()
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bf9'),
    name: 'Vaishu',
    branch: 'AIML',
    roll_no: 70
  },
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfa'),
    name: 'Anish',
    branch: 'AIML',
    roll_no: 5
  }
]
Demo> |
```

```
Demo> db.students.find().pretty()
[
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bf9'),
    name: 'Vaishu',
    branch: 'AIML',
    roll_no: 70
  },
  {
    _id: ObjectId('66f3e8bc49202f9f6dc73bfa'),
```

```
    name: 'Anish',  
    branch: 'AIML',  
    roll_no: 5  
  }  
]
```

```
Demo> db.students.find().pretty()  
[  
  {  
    _id: ObjectId('66f3e8bc49202f9f6dc73bf9'),  
    name: 'Vaishu',  
    branch: 'AIML',  
    roll_no: 70  
  },  
  {  
    _id: ObjectId('66f3e8bc49202f9f6dc73bfa'),  
    name: 'Anish',  
    branch: 'AIML',  
    roll_no: 5  
  }  
]  
Demo> |
```