

MONGODB

CREATE OPERATION

1. use library

- Purpose:
 - This command switches the current database context to library.
 - If the library database does not exist yet, MongoDB will create it automatically when you insert the first document.

2. db.books.insertMany()

- This command is used to insert multiple documents at once into the books collection.
- Instead of inserting documents one by one using insertOne(), insertMany() allows you to add a batch of documents in a single command, saving time.

```
test> use library
switched to db library
library> db
library
library> db.books.insertMany([
...   { title: "The Catcher in the Rye", author: "J.D. Salinger", published_year: 1951 },
...   { title: "1984", author: "George Orwell", published_year: 1949 },
...   { title: "Harry Potter and the Sorcerer's Stone", author: "J.K. Rowling", published_year: 1997 },
...   { title: "The Great Gatsby", author: "F. Scott Fitzgerald", published_year: 1925 },
...   { title: "To Kill a Mockingbird", author: "Harper Lee", published_year: 1960 },
...   { title: "The Hobbit", author: "J.R.R. Tolkien", published_year: 1937 },
...   { title: "Pride and Prejudice", author: "Jane Austen", published_year: 1813 },
...   { title: "The Da Vinci Code", author: "Dan Brown", published_year: 2003 },
...   { title: "MongoDB Basics", author: "John Doe", published_year: 2022 },
...   { title: "NoSQL Essentials", author: "Jane Smith", published_year: 2021 }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('688465bafb3d914d10eec4a9'),
    '1': ObjectId('688465bafb3d914d10eec4aa'),
    '2': ObjectId('688465bafb3d914d10eec4ab'),
    '3': ObjectId('688465bafb3d914d10eec4ac'),
    '4': ObjectId('688465bafb3d914d10eec4ad'),
    '5': ObjectId('688465bafb3d914d10eec4ae'),
    '6': ObjectId('688465bafb3d914d10eec4af'),
    '7': ObjectId('688465bafb3d914d10eec4b0'),
    '8': ObjectId('688465bafb3d914d10eec4b1'),
    '9': ObjectId('688465bafb3d914d10eec4b2')
  }
}
```

*****Read Operation:***

3. db.books.find()

- Used to retrieve (read) all documents from the books collection.
- This is part of the **Read Operation** in CRUD (Create, Read, Update, Delete).

```

library> db.books.find()
[
  {
    _id: ObjectId('688465bafb3d914d10eec4a9'),
    title: 'The Catcher in the Rye',
    author: 'J.D. Salinger',
    published_year: 1951
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4aa'),
    title: '1984',
    author: 'George Orwell',
    published_year: 1949
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ab'),
    title: 'Harry Potter and the Sorcerer's Stone',
    author: 'J.K. Rowling',
    published_year: 1997
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ac'),
    title: 'The Great Gatsby',
    author: 'F. Scott Fitzgerald',
    published_year: 1925
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ad'),
    title: 'To Kill a Mockingbird',
    author: 'Harper Lee',
    published_year: 1960
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ae'),
    title: 'The Hobbit',
    author: 'J.R.R. Tolkien',
    published_year: 1937
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4af'),
    title: 'Pride and Prejudice',
    author: 'Jane Austen',
    published_year: 1813
  },
]

```

4. db.books.find({author:'J.K. Rowling'})

- This query **retrieves only the documents** where the author field is exactly "J.K. Rowling".
- It is an example of a **filter query** in MongoDB.

```
...
library> db.books.find({author:'J.K. Rowling'})
[
  {
    _id: ObjectId('688465bafb3d914d10eec4ab'),
    title: 'Harry Potter and the Sorcerer's Stone',
    author: 'J.K. Rowling',
    published_year: 1997
  }
]
```

5. db.books.find().sort({published_year:1}).limit(2)

- Sorts documents by published_year in ascending order(1 retrieves oldest year and -1 for newest).
- Limit(2) – it retrieves 2 data from collections.

```
library> db.books.find().sort({published_year:1}).limit(2)
[
  {
    _id: ObjectId('688465bafb3d914d10eec4af'),
    title: 'Pride and Prejudice',
    author: 'Jane Austen',
    published_year: 1813
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ac'),
    title: 'The Great Gatsby',
    author: 'F. Scott Fitzgerald',
    published_year: 1925
  }
]
```

6. db.books.updateOne({title:'The Catcher in the Rye'},{\$set:{published_year:2025}})

- Finds the document with title equal to "The Catcher in the Rye".
- Updates its published_year to 2025 (current year).

```
library> db.books.updateOne({title:'The Catcher in the Rye'},{$set:{published_year:2025}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
library> db.books.find({title:'The Catcher in the Rye'})
[
  {
    _id: ObjectId('688465bafb3d914d10eec4a9'),
    title: 'The Catcher in the Rye',
    author: 'J.D. Salinger',
    published_year: 2025
  }
]
```

7. db.books.updateMany({},{\$set:{genre:"Mystery"}})

- Updates all documents ({} matches all) by adding a new field genre with the value "Mystery".

```
library> db.books.updateMany({},{$set:{genre:"Mystery"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 10,
  modifiedCount: 10,
  upsertedCount: 0
}
library> db.books.find()
[
  {
    _id: ObjectId('688465bafb3d914d10eec4a9'),
    title: 'The Catcher in the Rye',
    author: 'J.D. Salinger',
    published_year: 2025,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4aa'),
    title: '1984',
    author: 'George Orwell',
    published_year: 1949,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ab'),
    title: 'Harry Potter and the Sorcerer's Stone',
    author: 'J.K. Rowling',
    published_year: 1997,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4ac'),
    title: 'The Great Gatsby',
    author: 'F. Scott Fitzgerald',
    published_year: 1925,
    genre: 'Mystery'
  }
]
```

Delete Operation

8. `db.books.deleteOne({published_year:2003})`

- Deletes one document where the title is "1984".

9. `db.books.deleteMany({published_year:{<2000}})`

- Deletes all documents where published_year is less than (<) 2000.

```
library> db.books.deleteOne({published_year:1984})
{ acknowledged: true, deletedCount: 0 }
library> db.books.deleteOne({published_year:2003})
{ acknowledged: true, deletedCount: 1 }
library> db.books.deleteOne({title:1984})
{ acknowledged: true, deletedCount: 0 }
library> db.books.deleteOne({title:'1984'})
{ acknowledged: true, deletedCount: 1 }
library> db.books.deleteMany({published_year:{<2000}})
{ acknowledged: true, deletedCount: 5 }
library> db.books.find()
[
  {
    _id: ObjectId('688465bafb3d914d10eec4a9'),
    title: 'The Catcher in the Rye',
    author: 'J.D. Salinger',
    published_year: 2025,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4b1'),
    title: 'MongoDB Basics',
    author: 'John Doe',
    published_year: 2022,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4b2'),
    title: 'NoSQL Essentials',
    author: 'Jane Smith',
    published_year: 2021,
    genre: 'Mystery'
  }
]
```

*****Advanced Query:*****

10. db.books.find().sort({published_year:-1}).limit(3)

- Sorts documents by published_year in descending order (-1 means newest first).
- Limits results to the top 3 most recently published books.

```
library> db.books.find().sort({published_year:-1}).limit(3)
[
  {
    _id: ObjectId('688465bafb3d914d10eec4a9'),
    title: 'The Catcher in the Rye',
    author: 'J.D. Salinger',
    published_year: 2025,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4b1'),
    title: 'MongoDB Basics',
    author: 'John Doe',
    published_year: 2022,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4b2'),
    title: 'NoSQL Essentials',
    author: 'Jane Smith',
    published_year: 2021,
    genre: 'Mystery'
  }
]
```

11. db.books.find({

```
... $or: [
...   { title: /MongoDB/i },
...   { title: /NoSQL/i }
... ]
... })
```

- Uses \$or to match documents where the title contains "MongoDB" or "NoSQL".

```
library> db.books.find({
...   $or: [
...     { title: /MongoDB/i },
...     { title: /NoSQL/i }
...   ]
... })
[
  {
    _id: ObjectId('688465bafb3d914d10eec4b1'),
    title: 'MongoDB Basics',
    author: 'John Doe',
    published_year: 2022,
    genre: 'Mystery'
  },
  {
    _id: ObjectId('688465bafb3d914d10eec4b2'),
    title: 'NoSQL Essentials',
    author: 'Jane Smith',
    published_year: 2021,
    genre: 'Mystery'
  }
]
```