NAME :- Mohit B Joshi

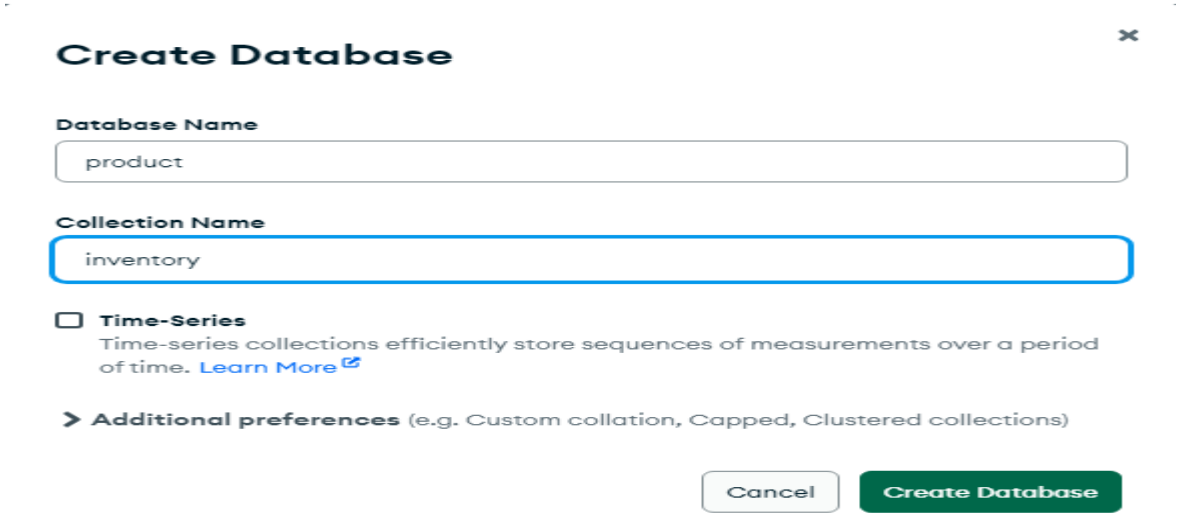
PRN :- 2122000372

ROLL NO :- C_20

ASSIGNMENT NO 7

Problem 1

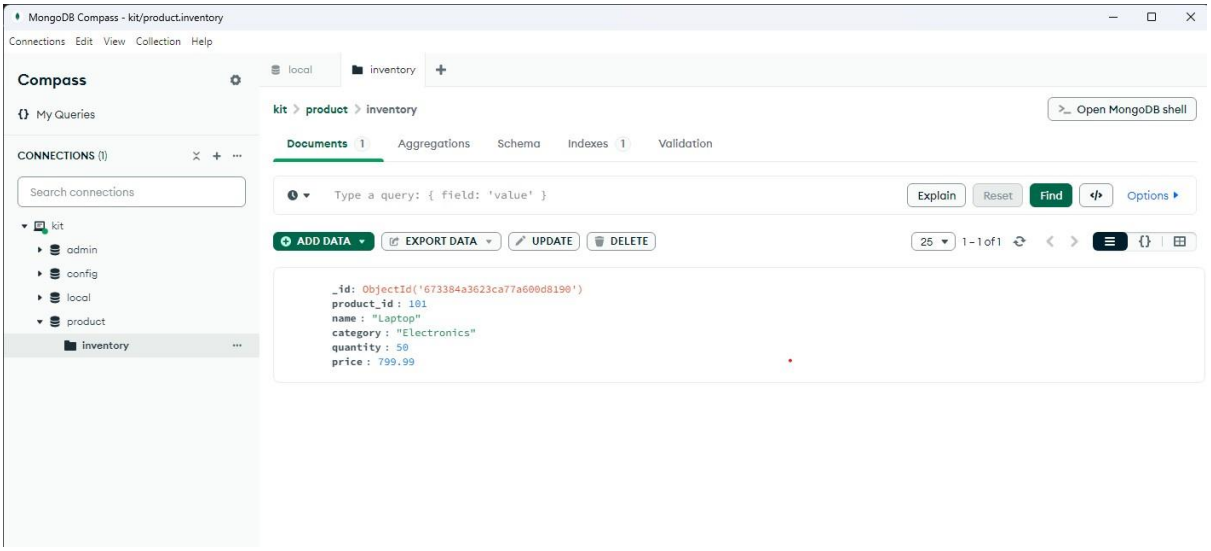
1. Create Database and collection



2. InsertOne

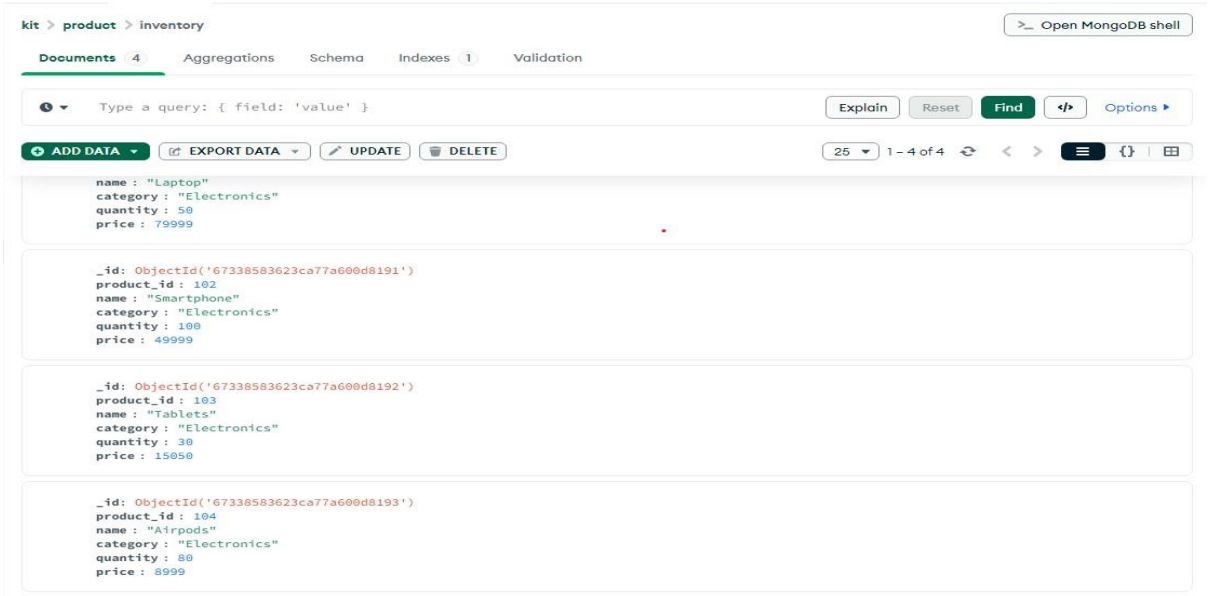
```
product> db.inventory.find()

product> db.inventory.insertOne({
...    product_id: 101,
...    name: "Laptop",
...    category: "Electronics",
...    quantity: 50,
...    price: 799.99
... })
{
  acknowledged: true,
  insertedId: ObjectId('673384a3623ca77a600d8190')
}
product>
```



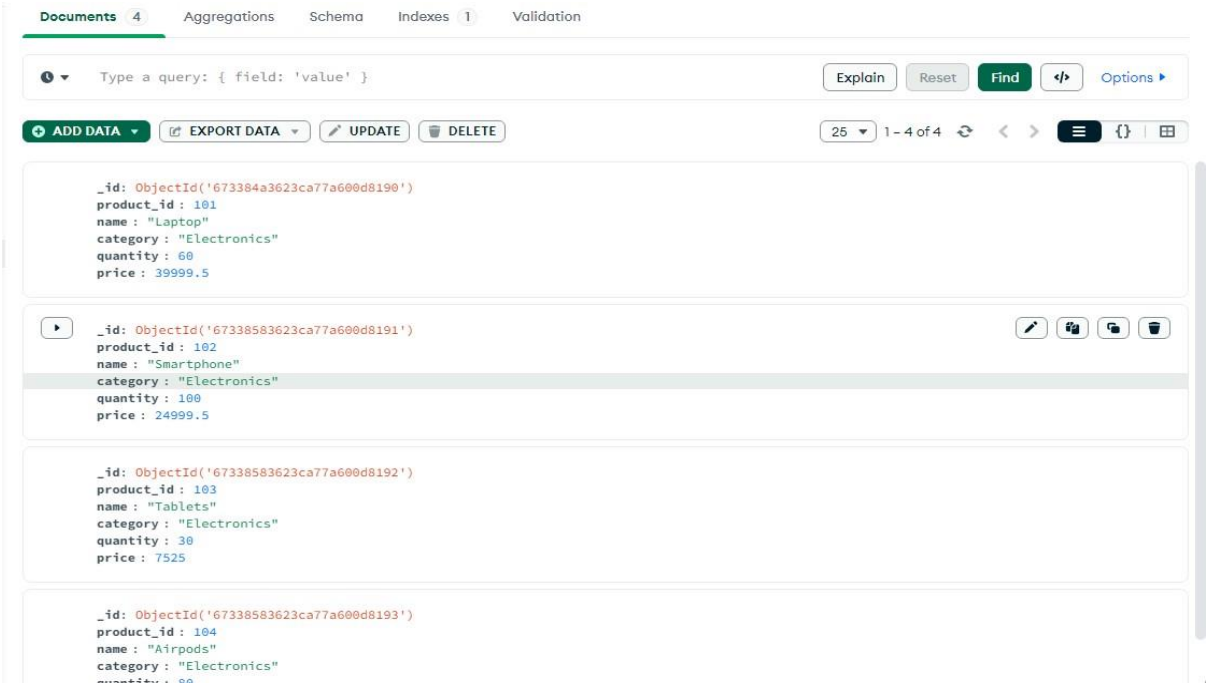
3.InsertMany

```
product> db.inventory.insertMany([
...   {
...     product_id: 102,
...     name: "Smartphone",
...     category: "Electronics",
...     quantity: 100,
...     price: 49999
...   },
...   {
...     product_id: 103,
...     name: "Tablets",
...     category: "Electronics",
...     quantity: 30,
...     price: 15050
...   },
...   {
...     product_id: 104,
...     name: "Airpods",
...     category: "Electronics",
...     quantity: 80,
...     price: 8999
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67338583623ca77a600d8191'),
    '1': ObjectId('67338583623ca77a600d8192'),
    '2': ObjectId('67338583623ca77a600d8193')
```



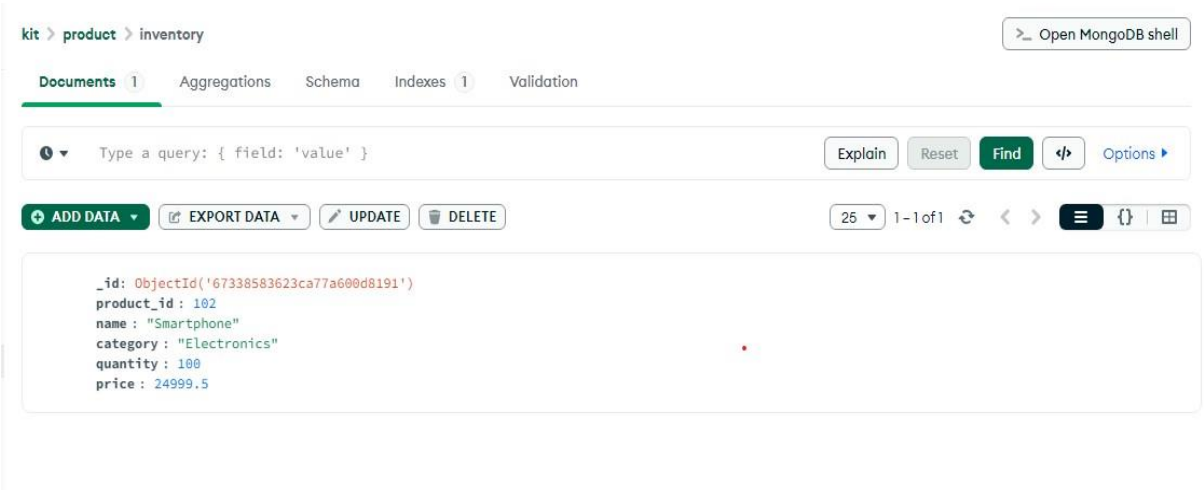
4.UpdateOne and UpdateMany

```
product> db.inventory.updateOne(
...    { product_id: 101 },  // Filter
...    { $set: { quantity: 60 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
product> db.inventory.updateMany(
...    { category: "Electronics" },
...    { $mul: { price: 0.5 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
product>
```



5.DeleteOne and DeleteMany

```
product> db.inventory.deleteOne({ product_id: 101 })
{ acknowledged: true, deletedCount: 1 }
product> db.inventory.deleteMany({ price: { $lt: 10000 } })
{ acknowledged: true, deletedCount: 2 }
product>
```



a. SELECT * FROM inventory

```
product> db.inventory.find().pretty()
[
  {
    _id: ObjectId('67338583623ca77a600d8191'),
    product_id: 102,
    name: 'Smartphone',
    category: 'Electronics',
    quantity: 100,
    price: 24999.5
  }
]
product>
```

b. SELECT * FROM inventory WHERE status = "D"

```
product> db.inventory.find({ status: "D" }).pretty()
[
  {
    _id: ObjectId('67338583623ca77a600d8191'),
    product_id: 102,
    name: 'Smartphone',
    category: 'Electronics',
    quantity: 100,
    price: 24999.5,
    status: 'D'
  },
  {
    _id: ObjectId('673387d6623ca77a600d8195'),
    product_id: 104,
    name: 'Airpods',
    category: 'Electronics',
    quantity: 80,
    price: 8999,
    status: 'D'
  }
]
```

c. SELECT * FROM inventory WHERE status in ("A", "D")

```
product> db.inventory.find({ status: { $in: ["A", "D"] } }).pretty()
[
  {
    _id: ObjectId('67338583623ca77a600d8191'),
    product_id: 102,
    name: 'Smartphone',
    category: 'Electronics',
    quantity: 100,
    price: 24999.5,
    status: 'D'
  },
  {
    _id: ObjectId('673387d6623ca77a600d8194'),
    product_id: 103,
    name: 'Tablets',
    category: 'Electronics',
    quantity: 30,
    price: 15050,
    status: 'A'
  },
  {
    _id: ObjectId('673387d6623ca77a600d8195'),
    product_id: 104,
    name: 'Airpods',
    category: 'Electronics',
    quantity: 80,
    price: 8999,
    status: 'D'
```

d. `SELECT * FROM inventory WHERE status = "A" AND qty < 30`

```
product> db.inventory.find({ status: "A", qty: { $lt: 30 } }).pretty()

product>
```

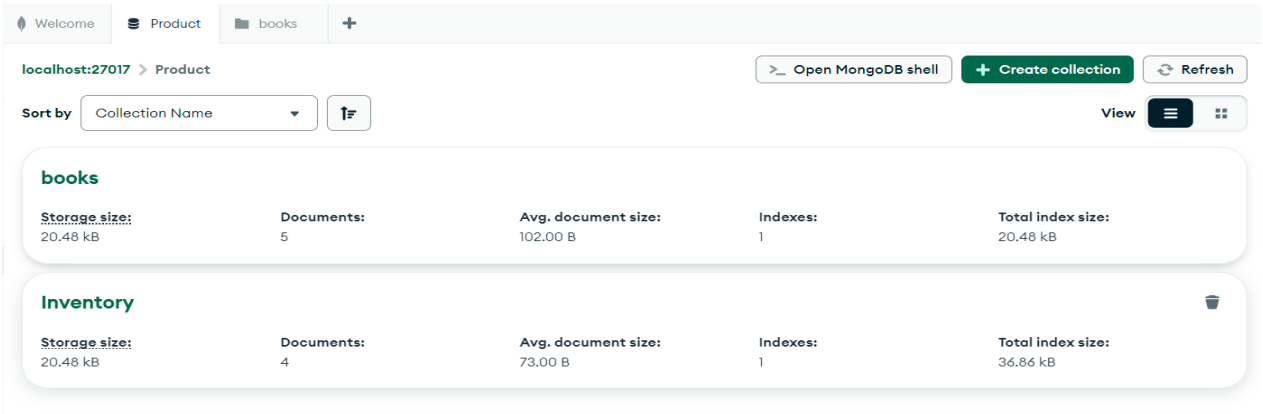
e. SELECT * FROM inventory WHERE status = "A" OR qty > 30

```
product> db.inventory.find({ $or: [ { status: "A" }, { qty: { $gt: 30 } } ] }).pretty()
[
  {
    _id: ObjectId('673387d6623ca77a600d8194'),
    product_id: 103,
    name: 'Tablets',
    category: 'Electronics',
    quantity: 30,
    price: 15050,
    status: 'A'
  }
]
```

PROBLEM 2

1. Create collection: books under product database and insertMany

```
Product> db.books.insertMany([{ "title": "1984", "author": "George Orwell", "year": 1949, "genre": "Dystopian" },
... { "title": "To Kill a Mockingbird", "author": "Harper Lee", "year": 1960, "genre": "Fiction" },
... { "title": "The Great Gatsby", "author": "F. Scott Fitzgerald", "year": 1925, "genre": "Fiction" },
... { "title": "Brave New World", "author": "Aldous Huxley", "year": 1932, "genre": "Dystopian" }])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6734ae1b7a68a58e322710be'),
    '1': ObjectId('6734ae1b7a68a58e322710bf'),
    '2': ObjectId('6734ae1b7a68a58e322710c0'),
    '3': ObjectId('6734ae1b7a68a58e322710c1')
  }
}
Product>
```



Find all books published after the year 1950

```
Product> db.books.find({ "year": { $gt: 1950 } });
[
  {
    _id: ObjectId('6734ae1b7a68a58e322710bf'),
    title: 'To Kill a Mockingbird',
    author: 'Harper Lee',
    year: 1960,
    genre: 'Fiction'
  }
]
Product>
```

3. Find all Dystopian books published before 1950.

ss

```
Product> db.books.find({ "genre": "Dystopian", "year": { $lt: 1950 } });
[
  {
    _id: ObjectId('6734adcc7a68a58e322710bd'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Dystopian'
  },
  {
    _id: ObjectId('6734ae1b7a68a58e322710be'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Dystopian'
  },
  {
    _id: ObjectId('6734ae1b7a68a58e322710c1'),
    title: 'Brave New World',
    author: 'Aldous Huxley',
    year: 1932,
    genre: 'Dystopian'
  }
]
```

4. Update the genre of "1984" to "Science Fiction".

```
Product> db.books.updateOne(
...   { "title": "1984" },
...   { $set: { "genre": "Science Fiction" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

5. Delete all books in the "Fiction" genre.

```
Product> db.books.deleteMany({ "genre": "Fiction" });
{ acknowledged: true, deletedCount: 2 }
Product>
```

6. Calculate the total number of books for each genre.

```
Product> db.books.aggregate([
...   { $group: { _id: "$genre", totalBooks: { $sum: 1 } } }
... ]);
[
  { _id: 'Dystopian', totalBooks: 2 },
  { _id: 'Science Fiction', totalBooks: 1 }
]
Product>
```

7. Create an index on the author field to improve query performance.

```
Product> db.books.createIndex({ "author": 1 });
author_1
Product>
fwd-i-search: _
```

8. Retrieve all books sorted by year in ascending order.

```
Product> db.books.find().sort({ "year": 1 });
[
  {
    _id: ObjectId('6734ae1b7a68a58e322710c1'),
    title: 'Brave New World',
    author: 'Aldous Huxley',
    year: 1932,
    genre: 'Dystopian'
  },
  {
    _id: ObjectId('6734adcc7a68a58e322710bd'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Science Fiction'
  },
  {
    _id: ObjectId('6734ae1b7a68a58e322710be'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Dystopian'
  }
]
```

9. Count the number of books written by "Harper Lee".

```
Product> db.books.countDocuments({ "author": "Harper Lee" });
0
Product>
```

a. Find books published between 1930 and 1960.

```
Product> db.books.find({}, { "title": 1, "author": 1, _id: 0 });
[
  { title: '1984', author: 'George Orwell' },
  { title: '1984', author: 'George Orwell' },
  { title: 'Brave New World', author: 'Aldous Huxley' }
]
Product>
```

b. Find all books published before 1950 and in the Fiction genre.

```
Product> db.books.find({
...   "year": { $lt: 1950 },
...   "genre": "Fiction"
... });

Product>
```

c. Find all books not written by Aldous Huxley.

```
Product> db.books.find({
...   "author": { $ne: "Aldous Huxley" }
... });
[
  {
    _id: ObjectId('6734adcc7a68a58e322710bd'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Science Fiction'
  },
  {
    _id: ObjectId('6734ae1b7a68a58e322710be'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Dystopian'
  }
]
```