E-Commerce Database

1. Database & Collection Setup:

o Create a new database (e.g., ecommerceDB)

o Create a collection (e.g., products)

```
C:\Users\saara>mongosh
Current Mongosh Log ID: 693c01431401d444cd1e2620
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using MongoDB:          8.2.2
Using Mongosh:          2.5.10

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

------
   The server generated these startup warnings when booting
   2025-12-10T21:11:31.733+00:00: Access control is not enabled for the database. Read and write access to data and configuration is
unrestricted
------

test> use ecommerceDB
switched to db ecommerceDB
ecommerceDB> db.createCollection("products")
{ ok: 1 }
```

2. Insert Operations:

o Insert one document into the **products** collection.

o Insert multiple documents with fields like **name, category, price**.

```
ecommerceDB> db.products.insertOne({name:"Laptop", category:"Electronics", price:10000})
{
  acknowledged: true,
  insertedId: ObjectId('693c02511401d444cd1e2621')
}
ecommerceDB> db.products.insertMany([{name:"T-Shirt", category:"Clothes", price:1000},{name:"Watch", category:"Electronics", price:50
00}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('693c05241401d444cd1e2622'),
    '1': ObjectId('693c05241401d444cd1e2623')
  }
}
```

3. Read Operations:

o Fetch all documents from the collection.

o Use filters to fetch specific data

```
ecommerceDB> db.products.find()
[
  {
    _id: ObjectId('693c02511401d444cd1e2621'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2622'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2623'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  }
]
ecommerceDB> db.products.find({name:"Watch"})
[
  {
    _id: ObjectId('693c05241401d444cd1e2623'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  }
]
```

4. Update Operations:

o Update a single document (e.g., change the price of T-Shirt).

```
ecommerceDB> db.products.updateOne({name:"T-Shirt"},{$set:{price:2000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
ecommerceDB> db.products.find()
[
  {
    _id: ObjectId('693c02511401d444cd1e2621'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2622'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 2000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2623'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  }
]
```

o Update multiple documents at once.

```
ecommerceDB> db.products.updateMany({category:"Electronics"}, {$set:{price:50000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
ecommerceDB> db.products.find()
[
  {
    _id: ObjectId('693c02511401d444cd1e2621'),
    name: 'Laptop',
    category: 'Electronics',
    price: 50000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2622'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 2000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2623'),
    name: 'Watch',
    category: 'Electronics',
    price: 50000
  }
]
```

5. Delete Operations:

o Delete one document based on a condition.

```
ecommerceDB> db.products.deleteOne({name:"T-Shirt"})
{ acknowledged: true, deletedCount: 1 }
ecommerceDB> db.products.find()
[
  {
    _id: ObjectId('693c02511401d444cd1e2621'),
    name: 'Laptop',
    category: 'Electronics',
    price: 50000
  },
  {
    _id: ObjectId('693c05241401d444cd1e2623'),
    name: 'Watch',
    category: 'Electronics',
    price: 50000
  }
]
```

o Delete multiple documents in the collection.

```
ecommerceDB> db.products.find()
[
  {
    _id: ObjectId('693c11cd1401d444cd1e2625'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c11cd1401d444cd1e2626'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  },
  {
    _id: ObjectId('693c11cd1401d444cd1e2627'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  }
]
ecommerceDB> db.products.deleteMany({category:"Electronics"})
{ acknowledged: true, deletedCount: 2 }
ecommerceDB> db.products.find()
[
  {
    _id: ObjectId('693c11cd1401d444cd1e2625'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  }
]
```

6. Query Operators:

o Practice using $gt, $lt, $in, $and, $or, $exists in different queries.

i) $gt (greater than)

```
ecommerceDB> db.products.find({price:{$gt:5000}})
[
  {
    _id: ObjectId('693c133e1401d444cd1e262a'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  }
]
```

ii) $lt (less than)

```
ecommerceDB> db.products.find({price:{$lt:5000}})
[
  {
    _id: ObjectId('693c11cd1401d444cd1e2625'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e2628'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  }
]
```

iii) $in(matches any value in array)

```
ecommerceDB> db.products.find({name:{$in:["Watch","Phone"]}})
[
  {
    _id: ObjectId('693c133e1401d444cd1e2629'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  }
]
ecommerceDB> db.products.find({name:{$in:["Watch","Laptop"]}})
[
  {
    _id: ObjectId('693c133e1401d444cd1e2629'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e262a'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  }
]
```

iv) $and

```
ecommerceDB> db.products.find({$and:[{price:{$gt:2000}},{category:"Electronics"}]})
[
  {
    _id: ObjectId('693c133e1401d444cd1e2629'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e262a'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  }
]
```

v) $or

```
ecommerceDB> db.products.find({$or:[{name:"Laptop"},{name:"T-Shirt"}]})
[
  {
    _id: ObjectId('693c11cd1401d444cd1e2625'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e2628'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e262a'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  }
]
ecommerceDB> db.products.find({$or:[{name:"Laptop"},{name:"Phone"}]})
[
  {
    _id: ObjectId('693c133e1401d444cd1e262a'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000
  }
]
```

Vi) $exists (true/false)

```
ecommerceDB>  db.products.find({brand:{$exists:true}})
[
  {
    _id: ObjectId('693c133e1401d444cd1e262a'),
    name: 'Laptop',
    category: 'Electronics',
    price: 10000,
    brand: 'Dell'
  }
]
ecommerceDB> db.products.find({brand:{$exists:false}})
[
  {
    _id: ObjectId('693c11cd1401d444cd1e2625'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e2628'),
    name: 'T-Shirt',
    category: 'Clothes',
    price: 1000
  },
  {
    _id: ObjectId('693c133e1401d444cd1e2629'),
    name: 'Watch',
    category: 'Electronics',
    price: 5000
  }
]
```