

Learner Details

- Name: Afreen Ahmed
 - Enrollment Number: SU625MR003
 - Batch / Class: June 2025 MERN
 - Assignment: MongoDB eCommerce Queries Assignment
 - Date of Submission: 22/08/2025
-

Problem Solving Activity

1. Program Statement

Create a MongoDB database ecommerceDB with a products collection. Insert at least 5 food product documents with fields like productId, name, category, price, stock, brand, ratings, reviews, and addedOn. Perform various CRUD operations, relational queries, and aggregation queries to simulate a real-world eCommerce database.

2. Algorithm

- Start MongoDB shell and switch to ecommerceDB.
 - Create a collection products.
 - Insert 5 documents with product details (food items).
 - Perform the following queries:
 - Basic Queries: Find products by category, price, and stock conditions.
 - Update Queries: Update stock, add discount field, insert a new review.
 - Relational & Sorting Queries: Sort products by price, find the most expensive product, and filter by price range.
 - Aggregation Queries: Calculate average ratings, count products per category, and total stock value per category.
 - Observe the results and validate outputs.
-

3. Pseudocode

BEGIN

USE ecommerceDB

CREATE products collection

INSERT multiple product documents with fields:

productId, name, category, price, stock, brand, ratings[], reviews[], addedOn

PERFORM queries:

FIND products by category, price < 1000, stock > 50

UPDATE stock (reduce by 10), add discount, add new review

SORT products by price descending

FIND most expensive product

FIND products between price 500–2000

AGGREGATE to find avg ratings per product

AGGREGATE to count products per category

AGGREGATE to calculate total stock value per category

END



4. Program Code

Insert Products

```
use ecommerceDB
```

```
db.createCollection("products")
```

```
db.products.insertMany([
```

```
{
```

```
    productId: 1,
```

```
    name: "Organic Apples",
```

```
    category: "Fruits",
```

```
price: 150,  
stock: 120,  
brand: "FreshFarm",  
ratings: [5, 4, 4, 5],  
reviews: [  
    { user: "Ayesha", comment: "Very fresh and tasty", rating: 5 },  
    { user: "Rahul", comment: "Good but a little pricey", rating: 4 }  
],  
addedOn: new Date()  
,  
{  
productId: 2,  
name: "Brown Rice",  
category: "Grains",  
price: 900,  
stock: 80,  
brand: "HealthyGrain",  
ratings: [4, 5, 3],  
reviews: [{ user: "Sneha", comment: "Quality is good", rating: 4 }],  
addedOn: new Date()  
,  
{  
productId: 3,  
name: "Almond Milk",  
category: "Beverages",  
price: 250,  
stock: 60,
```

```
brand: "NutriLife",
ratings: [5, 5, 4],
reviews: [{ user: "Kiran", comment: "Tastes amazing", rating: 5 }],
addedOn: new Date()

},
{
productId: 4,
name: "Whole Wheat Bread",
category: "Bakery",
price: 50,
stock: 200,
brand: "BakeHouse",
ratings: [4, 4, 3, 5],
reviews: [{ user: "Meena", comment: "Soft and fresh", rating: 4 }],
addedOn: new Date()

},
{
productId: 5,
name: "Olive Oil",
category: "Oils",
price: 1200,
stock: 40,
brand: "OlivaPure",
ratings: [5, 4, 5, 5],
reviews: [{ user: "Ravi", comment: "High quality oil", rating: 5 }],
addedOn: new Date()

}
```

)

Sample Queries

// Basic

```
db.products.find({ category: "Fruits" })
db.products.find({ price: { $lt: 1000 } })
db.products.find({ stock: { $gt: 50 } })
```

// Update

```
db.products.updateOne({ name: "Almond Milk" }, { $inc: { stock: -10 } })
db.products.updateMany({ category: "Bakery" }, { $set: { discount: "10%" } })
db.products.updateOne(
  { name: "Brown Rice" },
  { $push: { reviews: { user: "Arjun", comment: "Very healthy option", rating: 5 } } }
)
```

// Relational & Sorting

```
db.products.find().sort({ price: -1 })
db.products.find().sort({ price: -1 }).limit(1)
db.products.find({ price: { $gte: 500, $lte: 2000 } })
```

// Aggregation

```
db.products.aggregate([
  { $project: { name: 1, avgRating: { $avg: "$ratings" } } }
])
db.products.aggregate([
  { $group: { _id: "$category", totalProducts: { $sum: 1 } } }
])
db.products.aggregate([
  { $group: { _id: "$category", totalStockValue: { $sum: { $multiply: ["$price", "$stock"] } } } }
])
```

5. Screenshots of Output

📌 Screenshots of MongoDB shell results:

- Products inserted

```
ecommerceDB> db.products.insertMany([
...   {
...     productId: 1,
...     name: "Organic Apples",
...     category: "Fruits",
...     price: 150,
...     stock: 120,
...     brand: "FreshFarm",
...     ratings: [5, 4, 4, 5],
...     reviews: [
...       { user: "Ayesha", comment: "Very fresh and tasty", rating: 5 },
...       { user: "Rahul", comment: "Good but a little pricey", rating: 4 }
...     ],
...     addedOn: new Date()
...   },
...   {
...     productId: 2,
...     name: "Brown Rice",
...     category: "Grains",
...     price: 900,
...     stock: 80,
...     brand: "HealthyGrain",
...     ratings: [4, 5, 3],
...     reviews: [
...       { user: "Sneha", comment: "Quality is good", rating: 4 }
...     ],
...     addedOn: new Date()
...   },
...   {
...     productId: 3,
...     name: "Almond Milk",
...     category: "Beverages",
...     price: 250,
...     stock: 60,
...     brand: "NutriLife",
...     ratings: [5, 5, 4],
...     reviews: [
...       { user: "Kiran", comment: "Tastes amazing", rating: 5 }
...     ],
...     addedOn: new Date()
...   }
... ])
```

- Queries for Fruits, Price < 1000, Stock > 50

```
ecommerceDB> db.products.find({ category: "Fruits" })
[ {
  _id: ObjectId('68a7d5ffb4c6f81486eec4a9'),
  productId: 1,
  name: 'Organic Apples',
  category: 'Fruits',
  price: 150,
  stock: 120,
  brand: 'FreshFarm',
  ratings: [ 5, 4, 4, 5 ],
  reviews: [
    { user: 'Ayesha', comment: 'Very fresh and tasty', rating: 5 },
    { user: 'Rahul', comment: 'Good but a little pricey', rating: 4 }
  ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
}
]
ecommerceDB> db.products.find({ price: { $lt: 1000 } })
[ {
  _id: ObjectId('68a7d5ffb4c6f81486eec4a9'),
  productId: 1,
  name: 'Organic Apples',
  category: 'Fruits',
  price: 150,
  stock: 120,
  brand: 'FreshFarm',
  ratings: [ 5, 4, 4, 5 ],
  reviews: [
    { user: 'Ayesha', comment: 'Very fresh and tasty', rating: 5 },
    { user: 'Rahul', comment: 'Good but a little pricey', rating: 4 }
  ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
},
{
  _id: ObjectId('68a7d5ffb4c6f81486eec4aa'),
  productId: 2,
  name: 'Brown Rice',
```

- Update queries

```
ecommerceDB> db.products.find({ stock: { $gt: 50 } })
[ {
  _id: ObjectId('68a7d5ffb4c6f81486eec4a9'),
  productId: 1,
  name: 'Organic Apples',
  category: 'Fruits',
  price: 150,
  stock: 120,
  brand: 'FreshFarm',
  ratings: [ 5, 4, 4, 5 ],
  reviews: [
    { user: 'Ayesha', comment: 'Very fresh and tasty', rating: 5 },
    { user: 'Rahul', comment: 'Good but a little pricey', rating: 4 }
  ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
},
{
  _id: ObjectId('68a7d5ffb4c6f81486eec4aa'),
  productId: 2,
  name: 'Brown Rice',
  category: 'Grains',
  price: 900,
  stock: 80,
  brand: 'HealthyGrain',
  ratings: [ 4, 5, 3 ],
  reviews: [ { user: 'Sneha', comment: 'Quality is good', rating: 4 } ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
},
{
  _id: ObjectId('68a7d5ffb4c6f81486eec4ab'),
  productId: 3,
  name: 'Almond Milk',
  category: 'Beverages',
  price: 250,
  stock: 60,
  brand: 'NutriLife',
  ratings: [ 5, 5, 4 ],
  reviews: [
    { user: 'Kiran', comment: 'Tastes amazing', rating: 5 }
  ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
}
```

- Sorted products by price

```
ecommerceDB> db.products.find().sort({ price: -1 })
[ {
  _id: ObjectId('68a7d5ffb4c6f81486eec4ad'),
  productId: 5,
  name: 'Olive Oil',
  category: 'Oils',
  price: 1200,
  stock: 40,
  brand: 'OlivaPure',
  ratings: [ 5, 4, 5 ],
  reviews: [ { user: 'Ravi', comment: 'High quality oil', rating: 5 } ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
},
{
  _id: ObjectId('68a7d5ffb4c6f81486eec4aa'),
  productId: 2,
  name: 'Brown Rice',
  category: 'Grains',
  price: 900,
  stock: 80,
  brand: 'HealthyGrain',
  ratings: [ 4, 5, 3 ],
  reviews: [
    { user: 'Sneha', comment: 'Quality is good', rating: 4 },
    { user: 'Arjun', comment: 'Very healthy option', rating: 5 }
  ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
},
{
  _id: ObjectId('68a7d5ffb4c6f81486eec4ab'),
  productId: 3,
  name: 'Almond Milk',
  category: 'Beverages',
  price: 250,
  stock: 50,
  brand: 'NutriLife',
  ratings: [ 5, 5, 4 ],
  reviews: [
    { user: 'Kiran', comment: 'Tastes amazing', rating: 5 }
  ],
  addedOn: ISODate('2025-08-22T02:29:19.032Z')
}
```

- Aggregation results

```

ecommerceDB> db.products.find().sort({ price: -1 }).limit(1)
[
  {
    _id: ObjectId('68a7d5ffb4c6f81486eec4ad'),
    productId: 5,
    name: 'Olive Oil',
    category: 'Oils',
    price: 1200,
    stock: 48,
    brand: 'OlivaPure',
    ratings: [ 5, 4, 5, 5 ],
    reviews: [ { user: 'Ravi', comment: 'High quality oil', rating: 5 } ],
    addedOn: ISODate('2025-08-22T02:29:19.032Z')
  }
]
ecommerceDB> db.products.find({ price: { $gte: 500, $lte: 2000 } })
[
  {
    _id: ObjectId('68a7d5ffb4c6f81486eec4aa'),
    productId: 2,
    name: 'Brown Rice',
    category: 'Grains',
    price: 900,
    stock: 88,
    brand: 'HealthyGrain',
    ratings: [ 4, 5, 3 ],
    reviews: [
      { user: 'Sneha', comment: 'Quality is good', rating: 4 },
      { user: 'Arjun', comment: 'Very healthy option', rating: 5 }
    ],
    addedOn: ISODate('2025-08-22T02:29:19.032Z')
  },
  {
    _id: ObjectId('68a7d5ffb4c6f81486eec4ad'),
    productId: 5,
    name: 'Olive Oil',
    category: 'Oils',
    price: 1200,
    stock: 48,
    brand: 'OlivaPure',
  }
]

ecommerceDB> db.products.aggregate([
...   { $group: { _id: "$category", totalProducts: { $sum: 1 } } }
... ])
[
  { _id: 'Beverages', totalProducts: 1 },
  { _id: 'Fruits', totalProducts: 1 },
  { _id: 'Grains', totalProducts: 1 },
  { _id: 'Bakery', totalProducts: 1 },
  { _id: 'Oils', totalProducts: 1 }
]
ecommerceDB> db.products.aggregate([
...   { $group: { _id: "$category", totalStockValue: { $sum: { $multiply: ["$price", "$stock"] } } } }
... ])
[
  { _id: 'Beverages', totalStockValue: 12500 },
  { _id: 'Fruits', totalStockValue: 18000 },
  { _id: 'Grains', totalStockValue: 72000 },
  { _id: 'Bakery', totalStockValue: 10000 },
  { _id: 'Oils', totalStockValue: 48000 }
]
ecommerceDB>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
ecommerceDB> |

```

6. Observation / Reflection

In this e-commerce database project, a food-based product collection was created in MongoDB with fields such as productId, name, category, price, stock, brand, ratings, reviews, and addedOn. Various queries were performed to explore and manage the data effectively