Assessment-06

CRUD Operations in MongoDB

Group -05

NAME: Devi Dasari

REGD.NO:221FA04679

BRANCH: CSE

SECTION: 3C

SUBJECT: MSD

Objective: Aggregation Basics

*******Creating a databases for performing given aggregations*******

Creating databases of '**Sales','Products','Users','Blogposts','Orders','Reviews'**

```
> use myDatabase
< switched to db myDatabase
> db.sales.insertMany([
    { product: "Laptop", quantity: 10, price: 1000, date: new ISODate("2024-09-01T10:00:00Z"), city: "New York" },
    { product: "Mouse", quantity: 50, price: 25, date: new ISODate("2024-09-01T11:00:00Z"), city: "Los Angeles" },
    { product: "Keyboard", quantity: 20, price: 50, date: new ISODate("2024-09-02T14:00:00Z"), city: "New York" },
    { product: "Laptop", quantity: 5, price: 1000, date: new ISODate("2024-09-02T16:00:00Z"), city: "Chicago" }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66fbb961ea591897e66e551a'),
      '1': ObjectId('66fbb961ea591897e66e551b'),
      '2': ObjectId('66fbb961ea591897e66e551c'),
      '3': ObjectId('66fbb961ea591897e66e551d')
    }
  }
> db.users.insertMany([
    { name: "Alice", age: 25 },
    { name: "Bob", age: 30 },
    { name: "Charlie", age: 35 },
    { name: "David", age: 40 }
  ])
```

```
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66fbb9a0ea591897e66e551e'),
    '1': ObjectId('66fbb9a0ea591897e66e551f'),
    '2': ObjectId('66fbb9a0ea591897e66e5520'),
    '3': ObjectId('66fbb9a0ea591897e66e5521')
  }
}
db.products.insertMany([
  { name: "Laptop", category: "Electronics", price: 1000 },
  { name: "Mouse", category: "Electronics", price: 25 },
  { name: "Keyboard", category: "Electronics", price: 50 },
  { name: "Table", category: "Furniture", price: 150 },
  { name: "Chair", category: "Furniture", price: 100 }
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66fbb9a9ea591897e66e5522'),
    '1': ObjectId('66fbb9a9ea591897e66e5523'),
    '2': ObjectId('66fbb9a9ea591897e66e5524'),
    '3': ObjectId('66fbb9a9ea591897e66e5525'),
    '4': ObjectId('66fbb9a9ea591897e66e5526')
  }
}
```

```
> db.blogPosts.insertMany([
    { title: "MongoDB Tutorial", tags: ["database", "mongodb", "nosql"] },
    { title: "JavaScript Best Practices", tags: ["javascript", "programming", "web"] },
    { title: "Introduction to Cloud Computing", tags: ["cloud", "aws", "azure"] }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66fbb9b4ea591897e66e5527'),
      '1': ObjectId('66fbb9b4ea591897e66e5528'),
      '2': ObjectId('66fbb9b4ea591897e66e5529')
    }
  }
```

```
> db.orders.insertMany([
    { customer: "Alice", product: "Laptop", quantity: 1 },
    { customer: "Bob", product: "Mouse", quantity: 2 },
    { customer: "Charlie", product: "Keyboard", quantity: 1 },
    { customer: "Alice", product: "Table", quantity: 1 }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66fbb9c2ea591897e66e552a'),
      '1': ObjectId('66fbb9c2ea591897e66e552b'),
      '2': ObjectId('66fbb9c2ea591897e66e552c'),
      '3': ObjectId('66fbb9c2ea591897e66e552d')
    }
  }
> db.reviews.insertMany([
    { product: "Laptop", rating: 4.5 },
    { product: "Mouse", rating: 4.0 },
    { product: "Keyboard", rating: 3.5 },
    { product: "Chair", rating: 4.0 }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66fbb9e0ea591897e66e552e'),
      '1': ObjectId('66fbb9e0ea591897e66e552f'),
      '2': ObjectId('66fbb9e0ea591897e66e5530'),
      '3': ObjectId('66fbb9e0ea591897e66e5531')
    }
```

Question 1: Write an aggregation query to find the total quantity sold for each product in the sales collection.
Command: **db.sales.aggregate([**
{ $group: { _id: "$product", totalQuantitySold: { $sum: "$quantity" } } }
])

```
db.sales.aggregate([
  { $group: { _id: "$product", totalQuantitySold: { $sum: "$quantity" } } }
])
{
  _id: 'Laptop',
  totalQuantitySold: 15
}
{
  _id: 'Mouse',
  totalQuantitySold: 50
}
{
  _id: 'Keyboard',
  totalQuantitySold: 20
}
```

Question 2: Write an aggregation query to calculate the average age of users in the users collection.
Command: **db.users.aggregate([**
{ $group: { _id: null, averageAge: { $avg: "$age" } } }
])

```
> db.users.aggregate([
    { $group: { _id: null, averageAge: { $avg: "$age" } } }
  ])
< {
    _id: null,
    averageAge: 32.5
  }
```

Question 3: Write an aggregation query to find the minimum and maximum prices of products in the products collection.
Command: **db.products.aggregate([**
{ $group: { _id: null, minPrice: { $min: "$price" }, maxPrice: { $max: "$price" } } }
])

```
db.products.aggregate([
  { $group: { _id: null, minPrice: { $min: "$price" }, maxPrice: { $max: "$price" } } }
])
{
  _id: null,
  minPrice: 25,
  maxPrice: 1000
}
```

Question 4: Write an aggregation query to count the number of products in each category in the products collection.
Command**: db.products.aggregate([**
{ $group: { _id: "$category", count: { $sum: 1 } } }
])

```
> db.products.aggregate([
    { $group: { _id: "$category", count: { $sum: 1 } } }
  ])
< {
    _id: 'Electronics',
    count: 3
  }
  {
    _id: 'Furniture',
    count: 2
  }
```

Question 5: Write an aggregation query to list all unique tags used in the blogPosts collection
Command: **db.blogPosts.aggregate([**
{ $unwind: "$tags" },
{ $group: { _id: null, uniqueTags: { $addToSet: "$tags" } } }
])

```
db.blogPosts.aggregate([
  { $unwind: "$tags" },
  { $group: { _id: null, uniqueTags: { $addToSet: "$tags" } } }
])
{
  _id: null,
  uniqueTags: [
    'programming',
    'mongodb',
    'aws',
    'web',
    'cloud',
    'javascript',
    'database',
    'nosql',
    'azure'
  ]
}
```

Question 6: Write an aggregation query to find the total quantity sold per day from the sales collection
Command: **db.sales.aggregate([**
{ $group: { _id: { $dateToString: { format: "%Y-%m-%d", date: "$date" } },
totalQuantitySold: { $sum: "$quantity" } } }
])

```
> db.sales.aggregate([
    { $group: { _id: { $dateToString: { format: "%Y-%m-%d", date: "$date" } }, totalQuantitySold: { $sum: "$quantity" } } }
  ])
< {
    _id: '2024-09-01',
    totalQuantitySold: 60
  }
  {
    _id: '2024-09-02',
    totalQuantitySold: 25
  }
```

Question 7: Write an aggregation query to calculate the total revenue generated from each city in the sales collection.
Command: db.sales.aggregate([
{ $group: { _id: "$city", totalRevenue: { $sum: { $multiply: ["$price", "$quantity"] } } } }
])

```
> db.sales.aggregate([
    { $group: { _id: "$city", totalRevenue: { $sum: { $multiply: ["$price", "$quantity"] } } } }
  ])
< {
    _id: 'Los Angeles',
    totalRevenue: 1250
  }
  {
    _id: 'Chicago',
    totalRevenue: 5000
  }
  {
    _id: 'New York',
    totalRevenue: 11000
  }
```

Question 8: Write an aggregation query to find the total number of orders placed by each customer in the orders collection
Command: **db.orders.aggregate([**
{ $group: { _id: "$customer", totalOrders: { $sum: 1 } } }
])

```
> db.orders.aggregate([
    { $group: { _id: "$customer", totalOrders: { $sum: 1 } } }
  ])
< {
    _id: 'Charlie',
    totalOrders: 1
  }
  {
    _id: 'Bob',
    totalOrders: 1
  }
  {
    _id: 'Alice',
    totalOrders: 2
  }
```

Question 9: Write an aggregation query to find the average rating for each product in the reviews collection.
Command: **db.reviews.aggregate([**
{ $group: { _id: "$product", averageRating: { $avg: "$rating" } } }
])

```
> db.reviews.aggregate([
    { $group: { _id: "$product", averageRating: { $avg: "$rating" } } }
  ])
< {
    _id: 'Chair',
    averageRating: 4
  }
  {
    _id: 'Mouse',
    averageRating: 4
  }
  {
    _id: 'Laptop',
    averageRating: 4.5
  }
  {
    _id: 'Keyboard',
    averageRating: 3.5
  }
```

Question 10: Write an aggregation query to find the most popular category based on the total number of products sold in the sales collection.
Command: **db.sales.aggregate([**
{ $group: { _id: "$category", totalProductsSold: { $sum: "$quantity" } } },
{ $sort: { totalProductsSold: -1 } },
{ $limit: 1 }
])

```
db.sales.aggregate([
  { $group: { _id: "$category", totalProductsSold: { $sum: "$quantity" } } },
  { $sort: { totalProductsSold: -1 } },
  { $limit: 1 }
])
{
  _id: null,
  totalProductsSold: 85
}
```

****