

Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

Лабораторна робота № 6

Дисципліна: бази даних та інформаційні системи

Тема: Практичне використання Aggregation Framework у MongoDB

Виконав: Студент групи МІТ-31

Пугач Назар

Мета: Закріпити знання про основні стадії Aggregation Framework. Навчитися будувати ефективні агрегаційні запити. Освоїти методи фільтрації, групування, сортування та обробки масивів у MongoDB. Практично працювати з \$match, \$group, \$sort, \$unwind, \$lookup, \$project. Аналізувати продуктивність агрегацій та оптимізувати запити.

Хід роботи

Завдання 1: Створити колекції orders, customers, та products.

Створення колекції orders:

```
db.orders.insertMany([
  {
    orderId: "ORD001",
    customerId: ObjectId(),
    date: new ISODate("2025-02-15T00:00:00.000Z"),
    items: [
      { product: "Laptop", quantity: 1, price: 1200 },
      { product: "Mouse", quantity: 2, price: 50 }
    ],
    status: "Completed"
  },
  {
    orderId: "ORD002",
    customerId: ObjectId(),
    date: new ISODate("2025-02-15T00:00:00.000Z"),
    items: [
      { product: "Smartphone", quantity: 1, price: 800 },
      { product: "Headphones", quantity: 1, price: 150 }
    ],
    status: "Completed"
  },
  {
    orderId: "ORD003",
    customerId: ObjectId(),
    date: new ISODate("2025-02-15T00:00:00.000Z"),
    items: [
      { product: "Monitor", quantity: 2, price: 300 },
```

```

        { product: "Mousepad", quantity: 3, price: 20 }
      ],
      status: "Completed"
    },
    {
      orderId: "ORD004",
      customerId: ObjectId(),
      date: new ISODate("2022-09-10T00:00:00Z"),
      items: [
        { product: "Desk Lamp", quantity: 1, price: 50 },
        { product: "Headphones", quantity: 1, price: 150 }
      ],
      status: "Completed"
    },
    {
      orderId: "ORD005",
      customerId: ObjectId(),
      date: new ISODate("2024-02-18T00:00:00Z"),
      items: [
        { product: "Smartphone", quantity: 2, price: 800 },
        { product: "Desk Lamp", quantity: 2, price: 50 }
      ],
      status: "Pending"
    },
    {
      orderId: "ORD006",
      customerId: ObjectId(),
      date: new ISODate("2023-07-07T00:00:00Z"),
      items: [
        { product: "Mousepad", quantity: 5, price: 20 }
      ],
      status: "Completed"
    }
  ]
})

```

Створення колекції customers:

```

db.customers.insertMany([
  {
    name: "John Doe",
    email: "john.doe@example.com",
    city: "New York",
    registeredAt: new ISODate("2021-03-15T00:00:00Z")
  },
  {
    name: "Jane Smith",
    email: "jane.smith@example.com",
    city: "Los Angeles",
    registeredAt: new ISODate("2022-06-22T00:00:00Z")
  },
  {
    name: "Alice Johnson",
    email: "alice.johnson@example.com",

```

```

        city: "Los Angeles",
        registeredAt: new ISODate("2023-08-05T00:00:00Z")
    },
    {
        name: "Bob Smith",
        email: "bob.smith@example.com",
        city: "San Francisco",
        registeredAt: new ISODate("2022-05-12T00:00:00Z")
    },
    {
        name: "Charlie Brown",
        email: "charlie.brown@example.com",
        city: "Chicago",
        registeredAt: new ISODate("2021-09-25T00:00:00Z")
    },
    {
        name: "David White",
        email: "david.white@example.com",
        city: "New York",
        registeredAt: new ISODate("2024-01-15T00:00:00Z")
    },
    {
        name: "Eva Green",
        email: "eva.green@example.com",
        city: "Miami",
        registeredAt: new ISODate("2022-11-30T00:00:00Z")
    }
]
])

```

Створення колекції products:

```

db.products.insertMany([
    {
        name: "Laptop",
        category: "Electronics",
        price: 1200,
        stock: 15
    },
    {
        name: "Mouse",
        category: "Accessories",
        price: 50,
        stock: 50
    },
    {
        name: "Keyboard",
        category: "Accessories",
        price: 100,
        stock: 30
    },
    {
        name: "Smartphone",
        category: "Electronics",
        price: 800,

```

```

    stock: 25
  },
  {
    name: "Headphones",
    category: "Accessories",
    price: 150,
    stock: 40
  },
  {
    name: "Monitor",
    category: "Electronics",
    price: 300,
    stock: 20
  },
  {
    name: "Mousepad",
    category: "Accessories",
    price: 20,
    stock: 100
  },
  {
    name: "Desk Lamp",
    category: "Furniture",
    price: 50,
    stock: 15
  }
]
])

```

Завдання 2: Базові агрегаційні операції.

1. Відфільтруйте замовлення за останні 3 місяці

```

db.orders.aggregate([
  {
    $match: {
      date: { $gte: new Date(new Date().setMonth(new Date().getMonth() - 3))
    }
  }
])

```

```
< {
  _id: ObjectId('67e3da006ab8c9ec871b74c4'),
  orderId: 'ORD001',
  customerId: ObjectId('67e3da7e6ab8c9ec871b74c5'),
  date: 2025-02-15T00:00:00.000Z,
  items: [
    {
      product: 'Laptop',
      quantity: 1,
      price: 1200
    },
    {
      product: 'Mouse',
      quantity: 2,
      price: 50
    }
  ],
  status: 'Completed'
}
```

Рисунок 1 – Результат роботи функції.

2. Групування замовлень за місяцем.

```
db.orders.aggregate([
  {
    $project: {
      month: { $dateToString: { format: "%Y-%m", date: "$date" } }, //
      // Отримуємо рік-місяць
      totalAmount: {
        $sum: {
          $map: {
            input: "$items", // Для кожного товару в масиві
            as: "item", // Елемент масиву
            in: { $multiply: ["$$item.quantity", "$$item.price"] } //
            // Множимо кількість на ціну товару
          }
        }
      }
    }
  },
  {
    $group: {
      _id: "$month", // Групуємо по місяцю
      totalOrders: { $sum: 1 }, // Підраховуємо кількість замовлень
      totalAmount: { $sum: "$totalAmount" } // Підсумовуємо загальну суму
    }
  },
  { $sort: { _id: 1 } } // Сортуюмо по місяцю
])
```

```
< {
  _id: '2022-09',
  totalOrders: 1,
  totalAmount: 200
}
{
  _id: '2023-07',
  totalOrders: 1,
  totalAmount: 100
}
{
  _id: '2024-02',
  totalOrders: 1,
  totalAmount: 1700
}
{
  _id: '2025-02',
  totalOrders: 3,
  totalAmount: 2910
}
```

Рисунок 2 – Результат роботи функції.

3. Сортування за сумою замовлення.

```
db.orders.aggregate([
  {
    $addFields: {
      totalAmount: {
        $sum: {
          $map: {
            input: "$items",
            as: "item",
            in: { $multiply: [ "$$item.price", "$$item.quantity" ] }
          }
        }
      }
    }
  },
  {
    $sort: { totalAmount: -1 } // Сортуємо за зменшенням суми замовлення
  }
])
```

```

    _id: ObjectId('67e3dfae771669cfc12d58a5'),
    orderId: 'ORD005',
    customerId: ObjectId('67e3df60771669cfc12d58a0'),
    date: 2024-02-18T00:00:00.000Z,
    items: [
      {
        product: 'Smartphone',
        quantity: 2,
        price: 800
      },
      {
        product: 'Desk Lamp',
        quantity: 2,
        price: 50
      }
    ],
    status: 'Pending',
    totalAmount: 1700
  }

```

Рисунок 3 – Результат роботи функції.

Завдання 3: Робота з масивами.

4. Розгорніть масив items у замовленнях.
5. Підрахуйте кількість проданих одиниць товарів.

```

db.orders.aggregate([
  {
    $unwind: "$items" // Розгортаємо масив items у кожному замовленні
  },
  {
    $group: {
      _id: "$items.product", // Групуємо за назвою товару
      totalSold: {
        $sum: "$items.quantity" // Підраховуємо кількість одиниць
      }
    }
  },
  {
    $sort: { totalSold: -1 } // Сортуюмо за кількістю проданих одиниць у
    порядку спадання
  }
])

```

```
{
  _id: 'Mousepad',
  totalSold: 8
}
{
  _id: 'Smartphone',
  totalSold: 3
}
{
  _id: 'Desk Lamp',
  totalSold: 3
}
```

Рисунок 4 – Результат роботи функції.

Завдання 4: З'єднання колекцій (\$lookup).

6. Отримання інформації про клієнтів у замовленнях.
7. Визначте найбільш активних клієнтів.

```
db.orders.aggregate([
  {
    $lookup: {
      from: "customers",           // З'єднуємо колекцію orders з customers
      localField: "customerId",    // Поле для з'єднання в orders
      foreignField: "_id",         // Поле для з'єднання в customers
      as: "customer_info"         // Назва нового масиву з інформацією про
      клієнта
    }
  },
  {
    $unwind: "$customer_info"      // Розгортаємо масив customer_info, щоб
    отримати доступ до даних клієнта
  },
  {
    $group: {
      _id: "$customer_info.name",  // Групуємо за іменем клієнта
      totalOrders: { $sum: 1 }     // Підраховуємо кількість замовлень
    }
  },
  {
    $sort: { totalOrders: -1 }     // Сортуюмо за кількістю замовлень у
    порядку спадання
  }
])
```



```
{
  _id: 'David White',
  totalOrders: 2
}
{
  _id: 'John Doe',
  totalOrders: 1
}
```

Рисунок 5 – Результат роботи функції.

Завдання 5: Оптимізація запитів.

8. Перевірте продуктивність запиту.

Для перевірки продуктивності можна дописати `.explain("executionStats")` після команди, так ми отримаємо всю необхідну інформацію про продуктивність.

9. Оптимізуйте агрегаційний запит.

```
db.orders.aggregate([
  { $match: { date: { $gte: new Date("2024-01-01") } } },
  { $project: { orderId: 1, date: 1, status: 1 } }, // Вибірка тільки
необхідних полів
  { $group: { _id: "$status", totalOrders: { $sum: 1 } } }
])
```

```
{
  _id: 'Completed',
  totalOrders: 3
}
{
  _id: 'Pending',
  totalOrders: 1
}
```

Рисунок 6 – Результат роботи функції.

Додаткові завдання:

10. Визначте категорії товарів із найбільшою кількістю продажів.

Використайте `$group` для підрахунку загальної кількості проданих товарів за категоріями. Відсортуйте результат за спаданням.

```

db.orders.aggregate([
  { $unwind: "$items" },
  {
    $lookup: {
      from: "products",
      localField: "items.product",
      foreignField: "name",
      as: "product_info"
    }
  },
  { $unwind: "$product_info" },
  {
    $group: {
      _id: "$product_info.category",
      totalSold: { $sum: "$items.quantity" }
    }
  },
  { $sort: { totalSold: -1 } }
])

```

```

{
  _id: 'Accessories',
  totalSold: 12
}
{
  _id: 'Electronics',
  totalSold: 6
}
{
  _id: 'Furniture',
  totalSold: 3
}

```

Рисунок 7 – Результат роботи функції.

11. Розрахуйте середню ціну товарів у кожній категорії. Використайте \$group для підрахунку середньої ціни товарів у кожній категорії.

```

db.products.aggregate([
  {
    $group: {
      _id: "$category",
      avgPrice: { $avg: "$price" }
    }
  }
])

```

```

{
  _id: 'Electronics',
  avgPrice: 766.6666666666666
}
{
  _id: 'Accessories',
  avgPrice: 80
}
{
  _id: 'Furniture',
  avgPrice: 50
}

```

Рисунок 8 – Результат роботи функції.

12. Знайдіть користувачів, які зробили більше одного замовлення.

Використайте \$group і \$match, щоб знайти клієнтів, які мали більше одного замовлення.

```

db.orders.aggregate([
  {
    $group: {
      _id: "$customerId",
      orderCount: { $sum: 1 }
    }
  },
  { $match: { orderCount: { $gt: 1 } } }
])

```

```

{
  _id: ObjectId('67e3df60771669cfc12d58a0'),
  orderCount: 2
}

```

Рисунок 9 – Результат роботи функції.

Висновок: Під час виконання лабораторної роботи я закріпив знання з розширюваності PostgreSQL. Навчився створювати користувацькі типи даних. Реалізував власну користувацьку функцію. Створив тригери для логування змін у базі даних. Оновив діаграму бази даних відповідно до виконаних завдань. Перевірив коректність роботи реалізованих об'єктів через виконання тестових SQL-запитів.